

От переводчика lehsit:

Перевод не профессиональный делал для себя. Буду рад если полученный текст принесёт пользу в изучения темы, номера страниц кликабельны.



Red Hat Enterprise Linux 8.0 RH124

Red Hat System Administration I

Edition 1 20190507

Publication date 20190507

Authors: Fiona Allen, Marc Kesler, Saumik Paul, Snehangshu Karmakar, Victor Costea

Editor: Steven Bonneville, Ralph Rodriguez, David Sacco, Heather Charles, David O'Brien, Seth Kenlon

Авторские права © 2019 Red Hat, Inc.

Содержание этого курса, всех его модулей и сопутствующих материалов, включая раздаточные материалы для слушателей, защищено авторским правом © Red Hat, Inc., 2019.

Эта учебная программа, включая весь предоставленный здесь материал, предоставляется без каких-либо гарантий со стороны Red Hat, Inc. Red Hat, Inc. не несет ответственности за ущерб или судебные иски, возникшие в результате использования или неправильного использования содержимого или деталей, содержащихся в данном документе.

Linux® является зарегистрированным товарным знаком Линуса Торвальдса в США и других странах.

Java® является зарегистрированным товарным знаком Oracle и / или ее дочерних компаний.

XFS® является зарегистрированным товарным знаком Silicon Graphics International Corp. или ее дочерних компаний в США и / или других странах.

Знак OpenStack® Word и логотип OpenStack являются зарегистрированными товарными знаками / знаками обслуживания или товарными знаками/знаками обслуживания OpenStack Foundation в США и других странах и используются с разрешения OpenStack Foundation. Мы не связаны, не поддерживаются и не спонсируются OpenStack Foundation или сообществом OpenStack.

Все остальные товарные знаки являются собственностью соответствующих владельцев.

Авторы: Ачют Мадхусудан, Рудольф Кацл, Роб Локк

Условные обозначения в документе	9
Введение	
Системное администрирование Red Hat I	10
Ориентация на работу в классе	11
Управление вашими системами	12
Интернационализация	14
1. Начало работы с Red Hat Enterprise Linux	20
Что такое Linux?	20
Контрольный опрос: Начало работы с Red Hat Enterprise Linux	29
Резюме	31
2. Доступ к командной строке	32
Доступ к командной строке	33
Контрольный опрос: доступ к командной строке	39
Доступ к командной строке с рабочего стола	43
Упражнения с пошаговыми инструкциями: доступ к командной строке с рабочего стола	49
Выполнение команд с помощью оболочки Bash	51
Контрольный опрос: выполнение команд с помощью оболочки Bash	58
Лабораторная работа: Доступ к командной строке	62
Резюме	68
3. Управление файлами из командной строки	69
Описание концепций иерархии файловой системы Linux	70
Контрольный опрос: описание концепций иерархии файловой системы Linux	73
Указание файлов по имени	76
Контрольный опрос: указание файлов по имени	82
Управление файлами с помощью инструментов командной строки	86
Пошаговое упражнение: управление файлами с помощью инструментов командной строки	93
Создание связей между файлами	99
Упражнения с пошаговыми инструкциями: установление связей между файлами	103
Сопоставление имен файлов с расширениями оболочки	106
Контрольный опрос: сопоставление имен файлов с расширениями оболочки	112
Лабораторная работа: Управление файлами из командной строки	116

Резюме	126
4. Получение справки в Red Hat Enterprise Linux	127
Чтение страниц руководства	128
Упражнения с пошаговыми инструкциями: Чтение страниц руководства	132
Чтение документации Info	136
Упражнения с пошаговыми инструкциями: Чтение информационной документации	140
Лабораторная работа: Получение справки в Red Hat Enterprise Linux	143
Резюме	151
5. Создание, просмотр и редактирование текстовых файлов	152
Перенаправление вывода в файл или программу	153
Контрольный опрос: Перенаправление вывода в файл или программу	161
Редактирование текстовых файлов из командной строки	165
Упражнения с пошаговыми инструкциями: редактирование текстовых файлов из командной строки.	169
Изменение среды оболочки	171
Упражнения с пошаговыми инструкциями: изменение среды оболочки	178
Лабораторная работа: Создание, просмотр и редактирование текстовых файлов	181
Резюме	190
6. Управление локальными пользователями и группами	191
Описание концепций пользователей и групп	192
Контрольный опрос: Описание концепций пользователей и групп	196
Получение доступа суперпользователя	200
Упражнения с пошаговыми инструкциями: Получение доступа суперпользователя	206
Управление учетными записями локальных пользователей	212
Упражнения с пошаговыми инструкциями: Управление учетными записями локальных пользователей	215
Управление учетными записями локальной группы	219
Упражнения с пошаговыми инструкциями: Управление учетными записями локальных групп	222
Управление паролями пользователей	226
Упражнения с пошаговыми инструкциями: Управление паролями пользователей	231
Лабораторная работа: Управление локальными пользователями и группами	236

Резюме	242
7. Контроль доступа к файлам	243
Интерпретация разрешений файловой системы Linux	244
Контрольный опрос: Интерпретация разрешений файловой системы Linux	250
Управление разрешениями файловой системы из командной строки	254
Упражнения с пошаговыми инструкциями: Управление разрешениями файловой системы из командной строки	259
Управление разрешениями по умолчанию и доступом к файлам	263
Упражнения с пошаговыми инструкциями: Управление разрешениями по умолчанию и доступом к файлам	270
Лабораторная работа: Управление доступом к файлам	276
Резюме	284
8. Мониторинг и управление процессами Linux	285
Листинг Процессов	286
Контрольный опрос: Процессы листинга	292
Управление заданиями	294
Упражнения с пошаговыми инструкциями: Управление заданиями	298
Уничтожения Процессов	303
Упражнения с пошаговыми инструкциями: Уничтожения Процессов	310
Мониторинг активности процессов	315
Упражнения с пошаговыми инструкциями: Мониторинг активности процесса	320
Лабораторная работа: Мониторинг и управление процессами Linux	325
Резюме	337
9. Контролирующие службы и демоны	338
Выявление автоматически запускаемых системных процессов	339
Упражнение под руководством: определение автоматически запускаемых системных процессов	345
Управление системными службами	349
Упражнения с пошаговыми инструкциями: Управление системными услугами	354
Лабораторная работа: Службы управления и демоны	359
Резюме	364
10. Настройка и защита SSH	365

Доступ к удаленной командной строке с помощью SSH	366
Пошаговое упражнение: доступ к удаленной командной строке	370
Настройка аутентификации на основе ключа SSH	374
Упражнения с пошаговыми инструкциями: Настройка аутентификации на основе ключа SSH	379
Настройка конфигурации службы OpenSSH	385
Упражнения с пошаговыми инструкциями: Настройка конфигурации службы OpenSSH	388
Лабораторная работа: Настройка и защита SSH	395
Резюме	403
11. Анализ и хранение журналов	404
Описание архитектуры системного журнала	405
Контрольный опрос: Описание архитектуры системного журнала	407
Просмотр файлов системного журнала	411
Упражнения с пошаговыми инструкциями: Просмотр файлов системного журнала	416
Просмотр записей системного журнала	419
Упражнения с пошаговыми инструкциями: Просмотр записей системного журнала	425
Сохранение системного журнала	429
Упражнения с пошаговыми инструкциями: Сохранение системного журнала	433
Поддержание точного времени	436
Упражнения с пошаговыми инструкциями: Поддержание точного времени	440
Лабораторная работа: Анализ и хранение журналов	445
Резюме	451
12. Управление сетью	452
Описание сетевых концепций	453
Контрольный опрос: описание сетевых концепций	465
Проверка конфигурации сети	469
Упражнения с пошаговыми инструкциями: Проверка конфигурации сети	476
Настройка сети из командной строки	479
Упражнения с пошаговыми инструкциями: Настройка сети из командной строки	485
Редактирование файлов конфигурации сети	491
Упражнения с пошаговыми инструкциями: Редактирование файлов конфигурации сети	494

Настройка имен хостов и разрешения имен	500
Упражнения с пошаговыми инструкциями: Настройка имен хостов и разрешения имен	504
Лабораторная работа: Управление сетью	508
Резюме	513
13. Архивирование и передача файлов	514
Управление сжатыми архивами tar	515
Упражнения с пошаговыми инструкциями: Управление сжатыми архивами tar	522
Безопасная передача файлов между системами	525
Упражнения с пошаговыми инструкциями: Безопасная передача файлов между системами	529
Безопасная синхронизация файлов между системами	532
Упражнения с пошаговыми инструкциями: Безопасная синхронизация файлов между системами.	536
Лабораторная работа: Архивирование и передача файлов.	539
Резюме	546
14. Установка и обновление пакетов программного обеспечения	547
Регистрация систем для поддержки Red Hat	548
Контрольный опрос: Регистрация систем для поддержки Red Hat	552
Объяснение и исследование пакетов программного обеспечения RPM	554
Упражнение под руководством: объяснение и исследование пакетов программного обеспечения RPM	560
Установка и обновление пакетов программного обеспечения с помощью Yum	564
Упражнения с пошаговыми инструкциями: Установка и обновление пакетов программного обеспечения с помощью Yum	572
Включение репозиториев программного обеспечения Yum	578
Упражнения с пошаговыми инструкциями: Включение репозиториев программного обеспечения Yum	582
Управление потоками модуля пакета	593
Упражнения с пошаговыми инструкциями: Управление модулей пакета Streams	593
Лабораторная работа: Установка и обновление пакетов программного обеспечения	598
Резюме	605
15. Доступ к файловым системам Linux	606
Идентификация файловых систем и устройств	607

Контрольный опрос: Определение файловых систем и устройств	612
Монтирование и размонтирование файловых систем	614
Упражнения с пошаговыми инструкциями: Монтирование и размонтирование файловых систем	618
Поиск файлов в системе	621
Упражнения с пошаговыми инструкциями: Поиск файлов в системе	630
Лабораторная работа: Доступ к файловым системам Linux	633
Резюме	638
16. Анализ серверов и получение поддержки	639
Анализ и управление удаленными серверами	640
Упражнения с пошаговыми инструкциями: Анализ и управление удаленными серверами	657
Получение помощи на портале клиентов Red Hat	662
Упражнения с пошаговыми инструкциями: Получение помощи на портале клиентов Red Hat	673
Выявление и решение проблем с помощью Red Hat Insights	676
Контрольный опрос: Обнаружение и решение проблем с помощью Red Hat Insights	686
Резюме	688
17. Всесторонний обзор	689
Комплексный обзор	690
Лабораторная работа: Управление файлами из командной строки.	695
Лабораторная работа: Управление пользователями и группами, разрешениями и процессами	704
Лабораторная работа: Настройка и управление сервером	712
Лабораторная работа: Управление сетями	720
Лабораторная работа: Монтирование файловых систем и поиск файлов	727

СОГЛАШЕНИЯ В ДОКУМЕНТЕ



РЕКОМЕНДАЦИИ

«Рекомендации» описывают, где найти внешнюю документацию по теме.



ПРИМЕЧАНИЕ

«Примечания» - это подсказки, ярлыки или альтернативные подходы к решению поставленной задачи. Игнорирование заметки не должно иметь негативных последствий, но вы можете упустить уловку, которая облегчит вам жизнь.



ВАЖНО

В блоках «Важно» подробно описаны вещи, которые легко упустить: изменения конфигурации, которые применяются только к текущему сеансу, или службы, которые необходимо перезапустить перед применением обновления. Игнорирование поля с надписью: «Важно» не приведет к потере данных, но может вызвать раздражение и разочарование.



ПРЕДУПРЕЖДЕНИЕ

«Предупреждения» нельзя игнорировать. Игнорирование предупреждений, скорее всего, приведет к потере данных.

ВВЕДЕНИЕ

АДМИНИСТРАЦИЯ СИСТЕМЫ RED HAT I

Red Hat System Administration I (RH124) разработан для ИТ-специалистов, не имеющих опыта системного администрирования Linux. Курс предназначен для того, чтобы дать студентам «навыки выживания» в области администрирования Linux, сосредоточив внимание на основных административных задачах. Red Hat System Administration I также обеспечивает основу для студентов, планирующих стать системными администраторами Linux на полную ставку, путем ознакомления с ключевыми концепциями командной строки и инструментами корпоративного уровня. Эти концепции получили дальнейшее развитие в последующем курсе Red Hat System Administration II (RH134).

ЦЕЛИ КУРСА

- Получите достаточные навыки для выполнения основных задач системного администрирования в Red Hat Enterprise Linux.
- Развивать базовые навыки, необходимые системному администратору Red Hat Enterprise Linux, сертифицированному RHCSA.

АУДИТОРИЯ

- ИТ-специалисты в широком диапазоне дисциплин, которым необходимо выполнять основные задачи администрирования Linux, включая установку, создание сетевых подключений, управление физическим хранилищем и базовое администрирование безопасности.

НЕОБХОДИМЫЕ УСЛОВИЯ

- Для этого курса нет никаких формальных требований; однако предыдущий опыт системного администрирования в других операционных системах будет очень полезным.

ОРИЕНТАЦИЯ НА УЧЕБНУЮ СРЕДУ ДЛЯ ВЫПОЛНЕНИЯ ЗАДАНИЙ

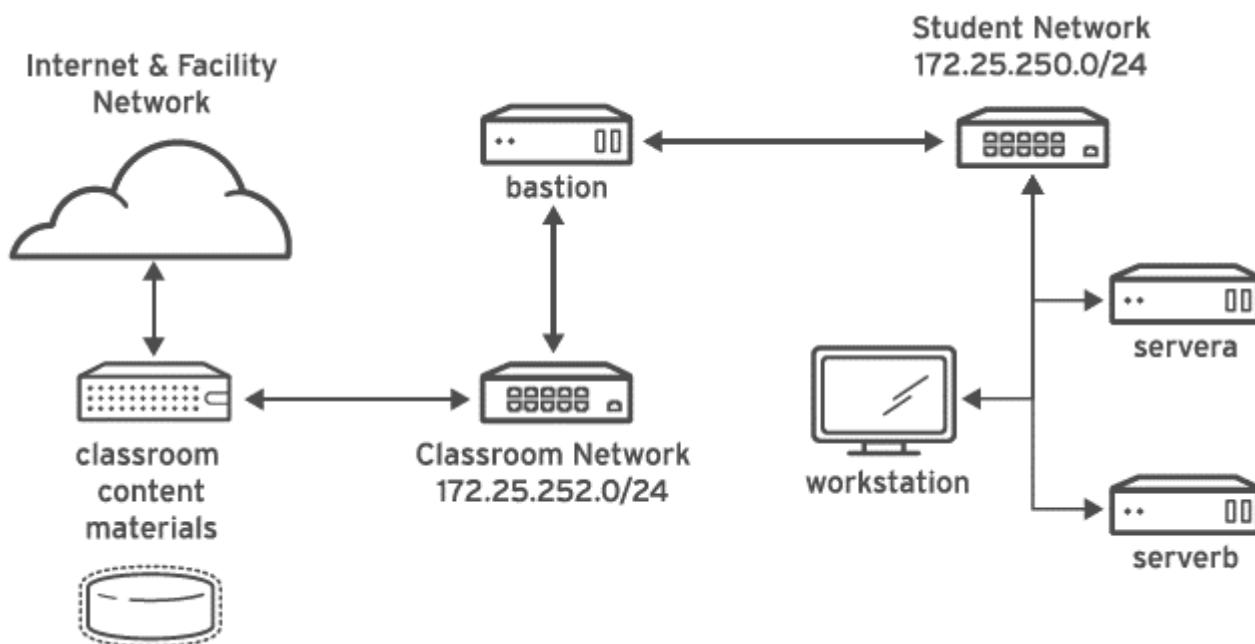


Рисунок 0.1: Среда для задач в классе

В этом курсе основной компьютерной системой, используемой для практического обучения, является рабочая станция (**workstation**). Студенты также используют две другие машины для этой деятельности: **servera**, и **serverb**. Все три из этих систем находятся в домене DNS **lab.example.com**.

Все компьютерные системы для учащихся имеют стандартную учетную запись пользователя **student** с паролем **student**. Пароль **root** во всех студенческих системах - **redhat**.

Назначение компьютеров в классной комнате

НАЗВАНИЕ МАШИНЫ	IP-АДРЕСА	РОЛЬ
bastion.lab.example.com	172.25.250.254	Система шлюза для подключения частной сети студента к классному серверу (всегда должна быть запущена)
workstation.lab.example.com	172.25.250.9	Графическая рабочая станция, используемая для системного администрирования
servera.lab.example.com	172.25.250.10	Первый сервер
serverb.lab.example.com	172.25.250.11	Второй сервер

Основная функция компьютера с именем **bastion** заключается в том, что он действует как маршрутизатор между сетью, соединяющей компьютеры учеников и сетью классной комнаты.

Если **bastion** не работает, другие компьютеры учеников смогут получить доступ к системам только в индивидуальной сети ученика.

Несколько систем в классе предоставляют вспомогательные услуги. Два сервера, **content.example.com** и **materials.example.com**, являются источниками программного обеспечения и лабораторных материалов, используемых в практических занятиях. Информация о том, как использовать эти серверы, содержится в инструкциях по этим действиям. Они предоставляются виртуальной машиной **classroom.example.com**. И **classroom**, и **bastion** должны всегда работать для правильного использования лабораторной среды.

УПРАВЛЕНИЕ ВАШИМИ СИСТЕМАМИ

Студентам назначают удаленные компьютеры в классе онлайн-обучения Red Hat. Доступ к ним осуществляется через веб-приложение, размещенное на сайте rol.redhat.com [<http://rol.redhat.com>]. Студенты должны войти на этот сайт, используя свои учетные данные пользователя Red Hat Customer Portal.

Управление виртуальными машинами

Виртуальные машины в вашем классе управляются через веб-страницу. Состояние каждой виртуальной машины в классе отображается на странице под вкладкой Онлайн-лаборатория.

Состояния машины

СОСТОЯНИЕ ВИРТУАЛЬНОЙ МАШИНЫ	ОПИСАНИЕ
STARTING	Виртуальная машина загружается.
STARTED	Виртуальная машина запущена и доступна (или, при загрузке, скоро будет).
STOPPING	Виртуальная машина завершает работу.
STOPPED	Виртуальная машина полностью выключена. При запуске виртуальная машина загружается в том же состоянии, что и при выключении (диск будет сохранен).
PUBLISHING	Выполняется первоначальное создание виртуальной машины.
WAITING_TO_START	Виртуальная машина ожидает запуска других виртуальных машин.

В зависимости от состояния машины доступны следующие действия.

Classroom/Действия машины

КНОПКА ИЛИ ДЕЙСТВИЕ	ОПИСАНИЕ
ОБЕСПЕЧЕНИЕ ЛАБОРАТОРИИ (PROVISION LAB)	Создайте класс ROL. Создает все виртуальные машины, необходимые для работы в классе, и запускает их. Это может занять несколько минут.
УДАЛИТЬ ЛАБОРАТОРИЮ (DELETE LAB)	Удалите класс ROL. Уничтожает все виртуальные машины в классе. Внимание! Любая работа, выполненная на дисках, теряется.
НАЧАТЬ ЛАБОРАТОРИЮ (START LAB)	Запустите все виртуальные машины в классе.
ЗАКРЫТИЕ ЛАБОРАТОРИИ (SHUTDOWN LAB)	Остановите все виртуальные машины в классе.
ОТКРЫТАЯ КОНСОЛЬ (OPEN CONSOLE)	Откройте новую вкладку в браузере и подключитесь к консоли виртуальной машины. Студенты могут входить прямо в виртуальную машину и запускать команды. В большинстве случаев студенты должны войти в виртуальную машину рабочей станции и использовать <code>ssh</code> для подключения к другим виртуальным машинам.
ДЕЙСТВИЕ → Старт (ACTION → Start)	Запустите (включите) виртуальную машину.
ДЕЙСТВИЕ → Завершение работы (ACTION → Shutdown)	Изящно выключите виртуальную машину, сохранив содержимое ее диска.
ДЕЙСТВИЕ → Выключение (ACTION → Power Off)	Принудительно выключите виртуальную машину, сохранив содержимое ее диска. Это эквивалентно отключению питания от физического компьютера.
ДЕЙСТВИЕ → Сброс (ACTION → Reset)	Принудительно выключите виртуальную машину и верните диск в исходное состояние. Внимание: любая работа, созданная на диске, теряется.

В начале упражнения, если есть указание на сброс одного узла виртуальной машины, щелкните **ACTION → Reset** только для конкретной виртуальной машины.

В начале упражнения, если есть указание на сброс всех виртуальных машин, нажмите **ACTION → Reset**.

Если вы хотите вернуть среду класса в исходное состояние в начале курса, вы можете нажать **DELETE LAB**, чтобы удалить всю среду класса. После удаления лаборатории вы можете щелкнуть **PROVISION LAB**, чтобы подготовить новый набор классных систем.



ПРЕДУПРЕЖДЕНИЕ

Операцию **DELETE LAB** нельзя отменить. Любая работа, которую вы к этому моменту выполнили в классе, будет потеряна.

Таймер автостопа

Регистрация в Red Hat Online Learning дает студентам определенное количество компьютерного времени. Чтобы помочь сэкономить выделенное компьютерное время, в классе ROL есть связанный таймер обратного отсчета, который отключает среду класса, когда таймер истекает.

Чтобы настроить таймер, нажмите **MODIFY**, чтобы отобразить диалоговое окно «Новое время автостопа (**New Autostop Time**)». Установите количество часов, по истечении которых класс должен автоматически остановиться. Нажмите **ADJUST TIME**, чтобы применить это изменение к настройкам таймера.

ИНТЕРНАЦИОНАЛИЗАЦИЯ

ВЫБОР ЯЗЫКА ДЛЯ ПОЛЬЗОВАТЕЛЯ

Ваши пользователи могут предпочесть использовать для среды рабочего стола язык, отличный от общесистемного по умолчанию. Они также могут захотеть использовать другую раскладку клавиатуры или другой метод ввода для своей учетной записи.

Языковые настройки

В среде рабочего стола GNOME пользователю может быть предложено установить предпочтительный язык и метод ввода при первом входе в систему. Если нет, то самый простой способ для отдельного пользователя настроить предпочтительный язык и настройки метода ввода - использовать приложение «Регион и язык (**Region & Language**)».

Вы можете запустить это приложение двумя способами. Вы можете запустить команду **gnome-control-center region** из окна терминала или на верхней панели из системного меню в правом углу, выбрать кнопку настроек (на которой есть крестовая отвертка и гаечный ключ для значка) снизу слева от меню.

В открывшемся окне выберите Регион и язык (**Region & Language**). Щелкните поле «Язык (**Language**)» и выберите предпочтительный язык из появившегося списка. Это также обновит настройку форматов до значения по умолчанию для этого языка. При следующем входе в систему эти изменения вступят в силу в полной мере.

Эти настройки влияют на среду рабочего стола GNOME и любые приложения, такие как **gnome-terminal**, которые запускаются внутри нее. Однако по умолчанию они не применяются к учетной записи, если доступ осуществляется через ssh-вход из удаленной системы или текстовый вход в виртуальную консоль (например, **tty5**).



ПРИМЕЧАНИЕ

Вы можете сделать так, чтобы в среде оболочки использовались те же настройки **LANG**, что и в графической среде, даже если вы входите в систему через текстовую виртуальную консоль или через ssh. Один из способов сделать это - разместить код, подобный приведенному ниже, в вашем файле **~/.bashrc**. В этом примере кода язык, используемый при текстовом входе в систему, будет соответствовать языку, установленному в настоящее время для среды рабочего стола GNOME пользователя:

```
i=$(grep 'Language=' /var/lib/AccountsService/users/${USER} \
| sed 's/Language=//')
if [ "$i" != "" ]; then
export LANG=$i
fi
```

Японский, корейский, китайский и другие языки с нелатинским набором символов могут некорректно отображаться на текстовых виртуальных консолях

Отдельные команды можно заставить использовать другой язык, установив переменную **LANG** в командной строке:

```
[user@host ~]$ LANG=fr_FR.utf8 date
jeu. avril 25 17:55:01 CET 2019
```

Последующие команды вернутся к использованию языка системы по умолчанию для вывода. Команду **locale** можно использовать для определения текущего значения **LANG** и других связанных переменных среды.

Настройки метода ввода

GNOME 3 в Red Hat Enterprise Linux 7 или более поздней версии автоматически использует систему выбора метода ввода **IBus**, что упрощает быстрое изменение раскладки клавиатуры и методов ввода.

Приложение «Регион и язык (**Region & Language**)» также можно использовать для включения альтернативных методов ввода. В окне приложения «Регион и язык (**Region & Language**)» поле «Источники ввода (**Input Sources**)» показывает, какие методы ввода доступны в настоящее время. По умолчанию английский (США) (**English (US)**) может быть единственным доступным способом. Выделите Английский (США) (**English (US)**) и щелкните значок клавиатуры, чтобы увидеть текущую раскладку клавиатуры.

Чтобы добавить другой метод ввода, нажмите кнопку «+» в нижнем левом углу окна «Источники ввода(**Input Sources**)». Откроется окно «Добавить источник входного сигнала (**Add**

an Input Source»). Выберите свой язык, а затем предпочтительный метод ввода или раскладку клавиатуры.

Если настроено более одного метода ввода, пользователь может быстро переключаться между ними, набирая **Super + Space** (иногда называемый **Windows + Space**). Индикатор состояния также появится на верхней панели GNOME, у которой есть две функции: он указывает, какой метод ввода активен, и действует как меню, которое можно использовать для переключения между методами ввода или выбора дополнительных функций более сложных методов ввода.

Некоторые методы отмечены шестерenkами, что указывает на то, что эти методы имеют расширенные параметры конфигурации и возможности. Например, метод ввода японский японский (Кана Кандзи (**Japanese (Kana Kanji)**)) позволяет пользователю предварительно редактировать текст на латинице и использовать клавиши со стрелками вниз и вверх для выбора правильных символов для использования.

Для носителей английского языка в США это также может оказаться полезным. Например, под **English (United States)** находится раскладка клавиатуры **English (international AltGr dead keys)**, при которой **AltGr** (или правый Alt) на 104/105-клавишной клавиатуре ПК рассматривается как клавиша-модификатор «вторичного сдвига» и неработающая клавиша. Ключ активации для набора дополнительных символов. Также доступны Дворжак (Dvorak) и другие альтернативные раскладки.



ПРИМЕЧАНИЕ

Любой символ **Unicode** можно ввести в среду рабочего стола **GNOME**, если вы знаете кодовый указатель **Unicode** символа. Введите **Ctrl+Shift+U**, а затем кодовый указатель. После нажатия **Ctrl+Shift+U** будет отображаться подчеркнутая буква **u**, указывающая на то, что система ожидает ввода кода указателя **Unicode**.

Например, строчная греческая буква лямбда имеет кодовый указатель **U+03BB**, и ее можно ввести, набрав **Ctrl+Shift+U**, затем **03BB**, и нажать клавишу **Enter**.

ОБЩИЕ СИСТЕМНЫЕ НАСТРОЙКИ ЯЗЫКА ПО УМОЛЧАНИЮ

В качестве языка системы по умолчанию используется английский (США) с использованием кодировки **Unicode UTF-8** в качестве набора символов (**en_US.utf8**), но это можно изменить во время или после установки.

Из командной строки пользователь **root** может изменить общесистемные настройки локали с помощью команды **localectl**. Если **localectl** запущен без аргументов, он отображает текущие общесистемные настройки локали.

Чтобы установить общесистемный язык по умолчанию, запустите команду **localectl set-locale LANG=locale**, где **locale** - это соответствующее значение переменной среды **LANG** из таблицы «Справочник кодов языков» в этой главе. Изменение вступит в силу для пользователей при их следующем входе в систему и хранится в **/etc/locale.conf**.

```
[root@host ~]# localectl set-locale LANG=fr_FR.utf8
```

В GNOME пользователь с правами администратора может изменить этот параметр в разделе «Регион и язык (**Region & Language**)», нажав кнопку «Экран входа в систему (**Login Screen**)» в правом верхнем углу окна. Изменение языка графического экрана входа также приведет к корректировке общесистемных языковых настроек по умолчанию, хранящихся в файле конфигурации `/etc/locale.conf`.



ВАЖНО

Текстовые виртуальные консоли, такие как `tty4`, более ограничены в шрифтах, которые они могут отображать, чем терминалы в виртуальной консоли, работающей в графической среде, или псевдотерминалы для сеансов `ssh`. Например, японские, корейские и китайские символы могут не отображаться должным образом на текстовой виртуальной консоли. По этой причине вам следует подумать об использовании английского или другого языка с набором латинских символов для общесистемного значения по умолчанию.

Точно так же виртуальные консоли на основе текста более ограничены в методах ввода, которые они поддерживают, это управляет отдельно от графической среды рабочего стола. Доступные глобальные параметры ввода можно настроить с помощью `localectl`, как для текстовых виртуальных консолей, так и для графической среды. См. Справочные страницы `localectl (1)` и `vconsole.conf (5)` для получения дополнительной информации.

ЯЗЫКОВЫЕ ПАКЕТЫ

Специальные пакеты RPM, называемые **langpacks**, устанавливают языковые пакеты, которые добавляют поддержку определенных языков. Пакеты **langpacks** используют зависимости для автоматической установки дополнительных пакетов RPM, содержащих локализации, словари и переводы для других пакетов программного обеспечения в вашей системе.

Чтобы вывести список установленных и которые могут быть установлены пакеты **langpack**, используйте `yum list langpacks-*`:

```
[root@host ~]# yum list langpacks-*
Updating Subscription Management repositories.
Updating Subscription Management repositories.
Installed Packages
langpacks-en.noarch      1.0-12.el8        @AppStream
Available Packages
langpacks-af.noarch       1.0-12.el8        rhel-8-for-x86_64-appstream-rpms
langpacks-am.noarch       1.0-12.el8        rhel-8-for-x86_64-appstream-rpms
langpacks-ar.noarch       1.0-12.el8        rhel-8-for-x86_64-appstream-rpms
langpacks-as.noarch       1.0-12.el8        rhel-8-for-x86_64-appstream-rpms
langpacks-ast.noarch      1.0-12.el8        rhel-8-for-x86_64-appstream-rpms
...output omitted...
```

Чтобы добавить языковую поддержку, установите соответствующий пакет **langpacks**. Например, следующая команда добавляет поддержку французского языка:

```
[root@host ~]# yuminstalllangpacks-fr
```

Используйте **yumrepoquery --whatsonplements**, чтобы определить, какие пакеты RPM могут быть установлены с помощью **langpack**:

```
[root@host ~]# yum repoquery --whatsonplements langpacks-fr
Updating Subscription Management repositories.
Updating Subscription Management repositories.
Last metadata expiration check: 0:01:33 ago on Wed 06 Feb 2019 10:47:24 AM CST.
glibc-langpack-fr-0:2.28-18.el8.x86_64
gnome-getting-started-docs-fr-0:3.28.2-1.el8.noarch
hunspell-fr-0:6.2-1.el8.noarch
hyphen-fr-0:3.0-1.el8.noarch
libreoffice-langpack-fr-1:6.0.6.1-9.el8.x86_64
man-pages-fr-0:3.70-16.el8.noarch
mythes-fr-0:2.3-10.el8.noarch
```



ВАЖНО

Пакеты **Langpacks** используют RPM*weak dependencies*, чтобы устанавливать дополнительные пакеты только тогда, когда также установлен основной пакет, который в этом нуждается.

Например, при установке **langpacks-fr**, как показано в предыдущих примерах, пакет **mythes-fr** будет установлен только в том случае, если тезаурус **mythes** также установлен в системе.

Если впоследствии в этой системе будет установлен **mythes**, пакет **mythes-fr** также будет установлен автоматически из-за слабой зависимости от уже установленного пакета **langpacks-fr**.



РЕКОМЕНДАЦИИ

Страницы руководства manpage: **locale (7)**, **localectl (1)**, **locale.conf (5)**, **vconsole.conf (5)**, **unicode (7)** и **utf-8 (7)**

СПРАВОЧНИК ЯЗЫКОВЫЕ КОДЫ



ПРИМЕЧАНИЕ

Эта таблица может не отражать все языковые пакеты, доступные в вашей системе. Используйте **yum info langpacks-SUFFIX**, чтобы получить дополнительную информацию о любом конкретном пакете **langpacks**.

Коды языков

ЯЗЫК	СУФФИКС ЯЗЫКОВЫХ ПАКЕТОВ	\$ LANG VALUE
English (US)	en	en_US.utf8
Assamese	as	as_IN.utf8
Bengali	bn	bn_IN.utf8
Chinese (Simplified)	zh_CN	zh_CN.utf8
Chinese (Traditional)	zh_TW	zh_TW.utf8
French	fr	fr_FR.utf8
German	de	de_DE.utf8
Gujarati	gu	gu_IN.utf8
Hindi	hi	hi_IN.utf8
Italian	it	it_IT.utf8
Japanese	ja	ja_JP.utf8
Kannada	kn	kn_IN.utf8
Korean	ko	ko_KR.utf8
Malayalam	ml	ml_IN.utf8
Marathi	mr	mr_IN.utf8
Odia	or	or_IN.utf8
Portuguese (Brazilian)	pt_BR	pt_BR.utf8
Punjabi	pa	pa_IN.utf8
Russian	ru	ru_RU.utf8
Spanish	es	es_ES.utf8

Tamil	ta	ta_IN.utf8
Telugu	te	te_IN.utf8

ГЛАВА 1

НАЧАЛО РАБОТЫ С RED HAT ENTERPRISE LINUX

ЦЕЛЬ

Описать и дать определение открытым исходным кодом, Linux, дистрибутивам Linux и Red Hat Enterprise Linux.

ЗАДАЧИ

- Определите и объясните назначение Linux, дистрибутивов Linux с открытым исходным кодом и Red Hat Enterprise Linux.

РАЗДЕЛЫ

- Что такое Linux?

КОНТРОЛЬНЫЙ ОПРОС

Началоработыс Red Hat Enterprise Linux.

ЧТО ТАКОЕ LINUX?

ЗАДАЧИ

После завершения этого раздела вы сможете определить и объяснить назначение Linux, открытого исходного кода, дистрибутивов Linux и RedHatEnterpriseLinux.

ПОЧЕМУ СЛЕДУЕТ ИЗУЧАТЬ LINUX?

Linux - важная технология, которую должны понимать ИТ-специалисты. Linux широко используется, если вы вообще пользуетесь Интернетом, вы, вероятно, уже взаимодействуете с системами Linux в своей повседневной жизни. Возможно, наиболее очевидный способ взаимодействия с системами Linux - просмотр Всемирной паутины и использование сайтов электронной коммерции для покупки и продажи продуктов.

Однако Linux используется не только для этого. Linux управляет торговыми точками и мировыми фондовыми рынками, а также поддерживает интеллектуальные телевизоры и бортовые развлекательные системы. На нем работает большинство из 500 лучших суперкомпьютеров в мире. Linux предоставляет базовые технологии, приведшие к революции в облачных технологиях,

и инструменты, используемые для создания следующего поколения контейнерных приложений микросервисов, программных технологий хранения и решений для больших данных.

В современном центре обработки данных Linux и Microsoft Windows являются основными игроками, при этом Linux - это растущий сегмент в данной области. Вот некоторые из многих причин для изучения Linux:

- Пользователь Windows должен взаимодействовать с Linux.
- При разработке приложений Linux размещает приложение или его среду выполнения.
- В облачных вычислениях экземпляры облака в частной или общедоступной облачной среде используют Linux в качестве операционной системы.
- С мобильными приложениями или Интернетом вещей (Internet of Things (IoT)) высока вероятность того, что операционная система вашего устройства использует Linux.
- Если вы ищете новые возможности в ИТ, навыки работы с Linux очень востребованы.

ЧТО ДЕЛАЕТ LINUX ВЕЛИКОЛЕПНЫМ?

Есть много разных ответов на вопрос «Что делает Linux великим?», рассмотрим три из них:

- Linux - это программное обеспечение с открытым исходным кодом.

Открытый исходный код не означает, что вы можете видеть, как работает система. Вы также можете поэкспериментировать с изменениями и бесплатно поделиться ими с другими. Модель с открытым исходным кодом означает, что улучшения легче вносить, что позволяет быстрее внедрять инновации.

- Linux обеспечивает легкий доступ к мощному интерфейсу командной строки (CLI) с поддержкой сценариев.

Linux был построен на основе базовой философии разработки, согласно которой пользователи могут выполнять все административные задачи из интерфейса командной строки. Это упрощает автоматизацию, развертывание и подготовку, а также упрощает локальное и удаленное администрирование системы. В отличие от других операционных систем, эти возможности были встроены с самого начала, и всегда предполагалось, что важные возможности задействованы.

- Linux - это модульная операционная система, которая позволяет легко заменять или удалять компоненты.

Компоненты системы можно модернизировать и обновлять по мере необходимости. Система Linux может быть универсальной рабочей станцией для разработчиков или крайне упрощенным программным устройством.

ЧТО ТАКОЕ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ С ОТКРЫТЫМ ИСТОЧНИКОМ?

Программное обеспечение с открытым исходным кодом - это программное обеспечение с исходным кодом, которое каждый может использовать, изучать, изменять и делиться.

Исходный код - это набор удобочитаемых инструкций, которые используются для создания программы. Его можно интерпретировать как сценарий или скомпилировать в двоичный исполняемый файл, который компьютер запускает напрямую. После создания исходного кода на него распространяется авторское право, и правообладатель контролирует условия, на которых программное обеспечение может быть скопировано, адаптировано и распространено. Пользователи могут использовать это программное обеспечение по лицензии на программное обеспечение.

Некоторое программное обеспечение имеет исходный код, который может видеть, изменять или распространять только человек, группа или организация, которые его создали. Это программное обеспечение иногда называют «проприетарным» или «закрытым исходным кодом». Как правило, лицензия позволяет запускать программу только конечному пользователю и не предоставляет доступа к источнику или имеет строго ограниченный доступ.

Программное обеспечение с открытым исходным кодом - другое дело. Когда правообладатель предоставляет программное обеспечение по лицензии с открытым исходным кодом, он предоставляет пользователю право запускать программу, а также просматривать, изменять, компилировать и распространять исходный код другим лицам без отчислений.

Открытый исходный код способствует сотрудничеству, совместному использованию, прозрачности и быстрым инновациям, потому что он побуждает людей, помимо первоначальных разработчиков, вносить изменения и улучшения в программное обеспечение и делиться им с другими.

Тот факт, что программное обеспечение имеет открытый исходный код, не означает, что его нельзя использовать или предоставлять в коммерческих целях. Открытый исходный код является важной частью коммерческих операций многих организаций. Некоторые лицензии с открытым исходным кодом позволяют повторно использовать код в продуктах с закрытым исходным кодом. Можно продавать открытый исходный код, но условия настоящих лицензий с открытым исходным кодом обычно позволяют заказчику распространять исходный код. Чаще всего такие поставщики, как Red Hat, предоставляют коммерческую помощь в развертывании, поддержке и расширении решений, основанных на продуктах с открытым исходным кодом.

Открытый исходный код имеет много преимуществ для пользователя:

- Контроль: посмотрите, что делает код, и измените его, чтобы улучшить.
- Обучение: учиться на реальном коде и разрабатывать более полезные приложения.
- Безопасность: проверяйте конфиденциальный код, исправляйте ошибки с помощью оригинальных разработчиков или без нее.

- Стабильность: код может пережить потерю первоначального разработчика или дистрибутора.

Суть в том, что открытый исходный код позволяет создавать более совершенное программное обеспечение с более высокой отдачей от инвестиций за счет сотрудничества.

ВИДЫ ЛИЦЕНЗИИ С ОТКРЫТЫМ ИСХОДНЫМ КОДОМ

Есть несколько способов предоставить программное обеспечение с открытым исходным кодом. Условия лицензии на программное обеспечение определяют, как исходный код может быть объединен с другим кодом или повторно использован, и существуют сотни различных лицензий с открытым исходным кодом. Однако, чтобы быть открытым исходным кодом, лицензии должны позволять пользователям свободно использовать, просматривать, изменять, компилировать и распространять код.

Есть два широких класса лицензий с открытым исходным кодом, которые особенно важны:

- **Лицензия Авторское лево**, которые предназначены для поощрения сохранения исходного кода кода.
- **Разрешительная лицензия**, предназначенные для максимального повторного использования кода.

Лицензия Авторское лево, или «одинаковые» лицензии, требуют, чтобы любой, кто распространяет исходный код, с изменениями или без них, также передавал другим право копировать, изменять и распространять код. Основное преимущество этих лицензий заключается в том, что они помогают сохранить существующий код и улучшения этого кода, открытыми и увеличивая объем доступного открытого исходного кода. Общие лицензии с авторским левом включают Стандартную общественную лицензию GNU (GPL) (*GNU General Public License (GPL)*) и Общественную лицензию ограниченного применения GNU (LGPL) (*Lesser GNU Public License (LGPL)*).

Разрешительная лицензия предназначены для максимального повторного использования исходного кода. Пользователи могут использовать исходный код для любых целей при условии сохранения заявлений об авторских правах и лицензии, включая повторное использование этого кода в рамках более ограничительных или даже проприетарных лицензий. Это очень упрощает повторное использование этого кода, но с риском поощрения только проприетарных улучшений. Несколько часто используемых разрешительных лицензий с открытым исходным кодом включают лицензию **MIT/X11**, упрощенную лицензию **BSD** и лицензию на программное обеспечение **Apache 2.0**.

КТО РАЗРАБОТАЕТ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ С ОТКРЫТЫМИ ИСХОДНЫМ КОДОМ?

Ошибка думать, что открытый исходный код разрабатывается исключительно «армией добровольцев» или даже армией отдельных лиц плюс Red Hat. Сегодня разработка с открытым исходным кодом в подавляющем большинстве случаев является профессиональной. Многим разработчикам их организации платят за работу с проектами с открытым исходным кодом, чтобы создавать и вносить улучшения, необходимые им и их клиентам.

Волонтеры и академическое сообщество играют важную роль и могут внести жизненно важный вклад, особенно в области новых технологий. Сочетание формального и неформального развития обеспечивает очень динамичную и продуктивную среду.

ЧТО ТАКОЕ RED HAT?

Red Hat - ведущий мировой поставщик программных решений с открытым исходным кодом, использующий подход с участием сообщества к надежным и высокопроизводительным облачным технологиям, Linux, промежуточному программному обеспечению, хранению данных и технологиям виртуализации. Миссия Red Hat - быть катализатором в сообществах клиентов, участников и партнеров, создающих лучшие технологии с открытым исходным кодом.

Роль Red Hat заключается в том, чтобы помочь клиентам установить связь с сообществом разработчиков ПО с открытым исходным кодом и их партнерами для эффективного использования программных решений с открытым исходным кодом. Red Hat активно участвует и поддерживает сообщество разработчиков ПО с открытым исходным кодом, и многолетний опыт убедил компанию в важности открытого исходного кода для будущего ИТ-индустрии.

Red Hat наиболее известна своим участием в сообществе Linux и дистрибутиве Red Hat Enterprise Linux. Однако Red Hat также очень активна в других сообществах с открытым исходным кодом, в том числе в проектах промежуточного программного обеспечения, сосредоточенных на сообществе разработчиков JBoss, решениях виртуализации, облачных технологиях, таких как OpenStack и OpenShift, а также в проектах хранения на основе программного обеспечения Ceph и Gluster.

ЧТО ТАКОЕ ДИСТРИБУТИВ LINUX?

Дистрибутив Linux - это устанавливаемая операционная система, созданная на основе ядра Linux и поддерживающая пользовательские программы, и библиотеки. Полная операционная система Linux разрабатывается не одной организацией, а группой независимых сообществ разработчиков с открытым исходным кодом, работающих с отдельными программными компонентами. Дистрибутив предоставляет пользователям простой способ установки и управления работающей системой Linux.

В 1991 году молодой студент-информатик по имени Линус Торвальдс разработал Unix-подобное ядро, которое он назвал **Linux**, под лицензией GPL, как программное обеспечение с открытым исходным кодом. Ядро - это основной компонент операционной системы, который управляет оборудованием, памятью и планированием запуска программ. Затем ядро Linux может быть дополнено другим программным обеспечением с открытым исходным кодом, например, утилитами и программами из проекта GNU, графическим интерфейсом из системы **X Window** от MIT и многими другими компонентами с открытым исходным кодом, такими как почтовый сервер **Sendmail** или веб-сервер **Apache** HTTP, чтобы создать полную Unix-подобную операционную систему с открытым исходным кодом.

Однако одной из проблем для пользователей Linux было собрать все эти части из множества разных источников. В самом начале своей истории разработчики Linux начали работать над предоставлением дистрибутива предварительно собранных и протестированных инструментов, которые пользователи могли бы загрузить и использовать для быстрой настройки своих систем Linux.

Существует множество различных дистрибутивов Linux с разными целями и критериями выбора и поддержки программного обеспечения, предоставляемого их дистрибутивами. Однако дистрибутивы обычно имеют много общих характеристик:

- Дистрибутивы состоят из ядра Linux и поддерживающих программ пользовательского пространства.
- Дистрибутивы могут быть небольшими и одноцелевыми или включать тысячи программ с открытым исходным кодом.
- Дистрибутивы должны обеспечивать средства установки и обновления дистрибутива и его компонентов.
- Поставщик дистрибутива должен поддерживать это программное обеспечение и, в идеале, принимать непосредственное участие в сообществе, разрабатывающем это программное обеспечение.

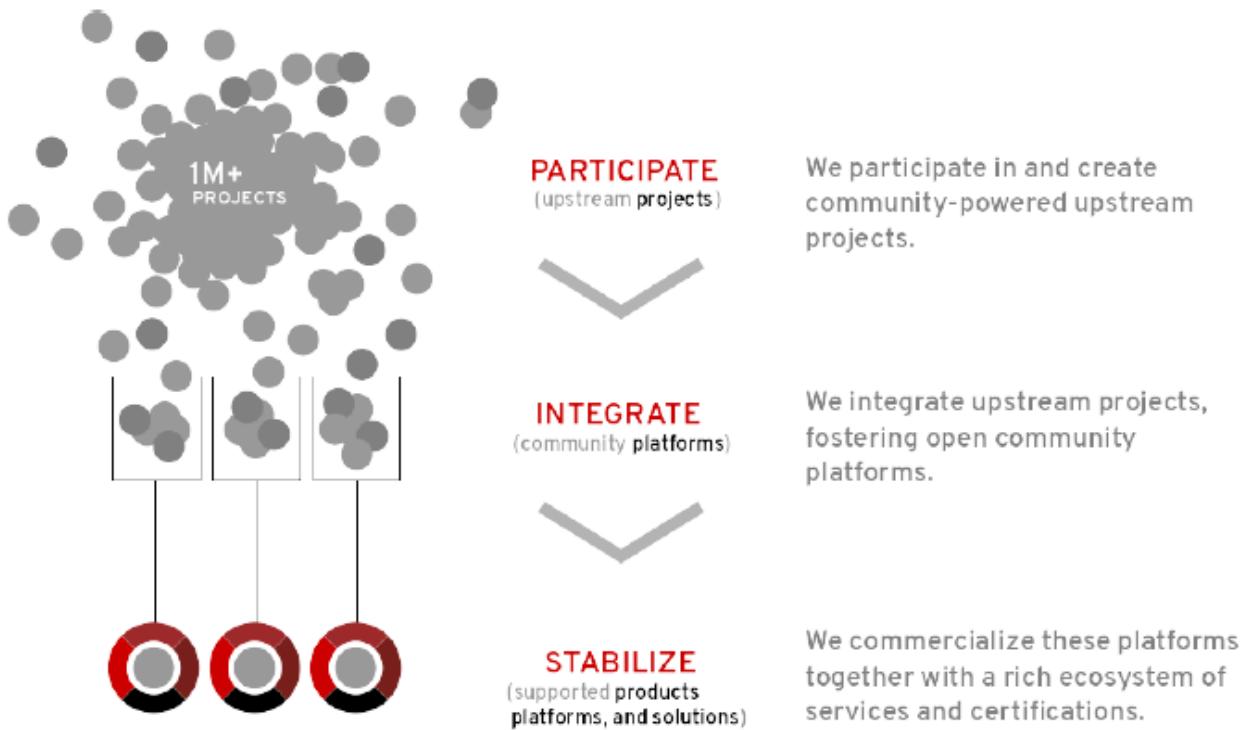
Red Hat Enterprise Linux - это коммерческий дистрибутив Linux от Red Hat.

RED HAT ENTERPRISE LINUX

Разработка Red Hat Enterprise Linux

Red Hat разрабатывает и интегрирует программное обеспечение с открытым исходным кодом в RHEL посредством многоступенчатого процесса.

- Red Hat участвует в поддержке отдельных проектов с открытым исходным кодом. Он вносит свой вклад в код, время разработчика, ресурсы и другую поддержку, часто сотрудничая с разработчиками из других дистрибутивов Linux. Это помогает улучшить общее качество программного обеспечения для всех.
- Red Hat спонсирует и интегрирует проекты с открытым исходным кодом в управляемый сообществом дистрибутив Linux, **Fedora**. Fedora предоставляет бесплатную рабочую среду, которая может служить лабораторией разработки и испытательной площадкой для функций, которые включены в их коммерческие продукты.
- Red Hat стабилизирует программное обеспечение, чтобы оно было готово к долгосрочной поддержке и стандартизации, и интегрирует его в свой корпоративный дистрибутив RHEL.



Fedora

Fedora - это проект сообщества, который производит и выпускает полную бесплатную операционную систему на базе Linux. Red Hat спонсирует сообщество и работает с представителями сообщества над интеграцией новейшего программного обеспечения из исходных продуктов в быстроразвивающийся и безопасный дистрибутив. Проект Fedora вносит свой вклад в мир бесплатных и открытых исходных кодов, и любой может принять в нем участие.

Однако **Fedora** ориентирована на инновации и совершенство, а не на долгосрочную стабильность. Новые крупные обновления происходят каждые шесть месяцев, и они могут привести к значительным изменениям. Fedora поддерживает выпуски только около года (два основных обновления), что делает ее менее подходящей для корпоративного использования.

Red Hat Enterprise Linux

Red Hat Enterprise Linux (RHEL) - коммерчески поддерживаемый дистрибутив Linux для предприятий Red Hat. Это ведущая платформа для вычислений с открытым исходным кодом, а не просто набор зрелых проектов с открытым исходным кодом. RHEL тщательно протестирован, имеет обширную поддерживающую экосистему партнеров, сертификаты оборудования и программного обеспечения, консультационные услуги, обучение, а также многолетнюю поддержку и гарантии обслуживания.

Red Hat основывает свои основные выпуски RHEL на Fedora. Однако после этого Red Hat может выбирать, какие пакеты включать, вносить дальнейшие улучшения (вносить свой вклад в вышестоящие проекты и Fedora) и принимать решения о конфигурации, которые удовлетворяют потребности клиентов. Red Hat помогает поставщикам и клиентам взаимодействовать с сообществом разработчиков ПО с открытым исходным кодом, а также работать с вышестоящей разработкой для разработки решений и устранения проблем.

Red Hat Enterprise Linux использует модель распространения на основе подписки. Потому что это открытое исходное программное обеспечение, это не плата за лицензию. Вместо этого он платит за поддержку, обслуживание, обновления, исправления безопасности, доступ к базе знаний на портале клиентов Red Hat (<http://access.redhat.com/>), сертификаты и т. д. Заказчик платит за долгосрочную поддержку, опыт, ответственное отношение и помочь, когда они в ней нуждаются. Когда основные обновления становятся доступными, клиенты могут переходить к ним в удобное для них время, не платя больше. Это упрощает управление как экономическими, так и практическими аспектами обновления системы.

CentOS

CentOS - это управляемый сообществом дистрибутив Linux, полученный из большей части кодовой базы Red Hat Enterprise Linux с открытым исходным кодом и других источников. Она бесплатна, проста в установке, укомплектована персоналом и поддерживается активным сообществом добровольцев, работающим независимо от Red Hat.

В следующей таблице перечислены некоторые ключевые различия между CentOS и Red Hat Enterprise Linux.

CENTOS	RED HAT ENTERPRISE LINUX
Только сама поддержка.	Доступно несколько уровней поддержки, включая стандартную поддержку в рабочее время, круглосуточную поддержку Premium для критических проблем и подписку на поддержку начального уровня. Различные уровни SLA могут быть смешаны и согласованы в среде.
Создание исправлений начинается, когда становится доступен официальный выпуск исправлений RHEL.	Быстрое реагирование на проблемы со стороны собственных разработчиков, оперативные исправления могут быть доступны до официального выпуска исправлений RHEL.
Обновления пакетов предоставляются для самого последнего второстепенного выпуска до конца 2 фазы технической поддержки RHEL.	Обновления доступны для более старых второстепенных выпусков в рамках программы расширенной поддержки обновлений (EUS), а также в течение многих лет после окончания технической поддержки 2 через программу расширенной поддержки жизненного цикла (ELS).
Обычно не сертифицирована поставщиками программного обеспечения, такими как SAS, SAP и Oracle, в качестве поддерживающей платформы.	Тысячи сертифицированных приложений от сотен независимых поставщиков программного обеспечения.

<p>Справочные ресурсы и документация доступны на форумах, в списках рассылки, в чате, на веб-сайте и вики-сайтах CentOS Project, а также из других источников сообщества.</p>	<p>Документация, эталонные архитектуры, тематические исследования и статьи базы знаний доступны на портале для клиентов Red Hat. Доступ к лабораториям Red Hat Customer Portal Labs, набору инструментов, которые вы можете использовать для повышения производительности, выявления проблем с безопасностью или помочь в решении любых проблем. Дополнительный упреждающий анализ системы с помощью Red Hat Insights, инструмента на основе SaaS, позволяющего в реальном времени оценивать риски, связанные с производительностью, доступностью, стабильностью и безопасностью.</p>
---	---

ПОПРОБОВАТЬ REDHAT ENTERPRISE LINUX

Есть много разных способов попробовать Red Hat Enterprise Linux. Один из способов - загрузить пробную копию с веб-сайта <https://access.redhat.com/products/red-hat-enterprise-linux/evaluation>. Эта страница содержит ссылки на дополнительную информацию.

Red Hat также предоставляет бесплатные подписки на ряд продуктов для целей разработки через Программу разработчиков Red Hat по адресу <https://developer.redhat.com>. Эти подписки позволяют разработчикам быстро разрабатывать, создавать прототипы, тестировать и демонстрировать свое программное обеспечение для развертывания на одних и тех же корпоративных продуктах.

Другой подход - развернуть экземпляр Red Hat Enterprise Linux, доступный через облачного провайдера. Например, у Red Hat есть официальные AMI, доступные для Red Hat Enterprise Linux в Amazon AWS Marketplace.

Для получения дополнительной информации посетите страницу Red Hat Enterprise Linux «Начало работы», указанную в конце этого раздела.



РЕКОМЕНДАЦИИ

Начало работы с Red Hat Enterprise Linux

<https://access.redhat.com/products/red-hat-enterprise-linux#getstarted>

Открытый исходный код

<https://opensource.com/open-source-way>

КОНТРОЛЬНЫЙ ОПРОС

НАЧАЛО РАБОТЫ С CREDHAT ENTERPRISE LINUX

Выберите правильные ответы на следующие вопросы:

- 1.** Какие два из перечисленных ниже преимуществ программного обеспечения с открытым исходным кодом для пользователя? (Выберите два.)
 - a.** Код может пережить потерю первоначального разработчика или дистрибутора.
 - b.** Чувствительные части кода защищены и доступны только исходному разработчику.
 - c.** Вы можете учиться на реальном коде и разрабатывать более эффективные приложения.
 - d.** Код остается открытым, пока он находится в общедоступном репозитории, но лицензия может измениться, если она включена в программное обеспечение с закрытым исходным кодом.
- 2.** Какие два из следующих способов используются Red Hat для разработки своих продуктов для будущего и взаимодействия с сообществом? (Выберите два.)
 - a.** Спонсируйте и интегрируйте проекты с открытым исходным кодом в проект Fedora, управляемый сообществом.
 - b.** Разработайте специальные инструменты интеграции, доступные только в дистрибутивах Red Hat.
 - c.** Участие в проектах по разработке и продвижению.
 - d.** Переупаковка и повторное лицензирование продуктов сообщества.
- 3.** Какие два утверждения описывают преимущества Linux? (Выберите два.)
 - a.** Linux полностью разрабатывается добровольцами, что делает его недорогой операционной системой.
 - b.** Linux имеет модульную структуру и может быть сконфигурирован как полноценный графический рабочий стол или небольшое устройство.
 - c.** Linux заблокирован в известном состоянии как минимум на один год для каждого выпуска, что упрощает разработку пользовательского программного обеспечения.
 - d.** Linux включает мощный интерфейс командной строки с поддержкой сценариев, упрощающий автоматизацию и подготовку.

РЕШЕНИЕ

НАЧАЛО РАБОТЫ С RED HAT ENTERPRISE LINUX

Выберите правильные ответы на следующие вопросы:

- 1.** Какие два из перечисленных ниже преимуществ программного обеспечения с открытым исходным кодом для пользователя? (Выберите два.)
 - a.** Код может пережить потерю первоначального разработчика или дистрибутора.
 - b.** Чувствительные части кода защищены и доступны только исходному разработчику.
 - c.** Вы можете учиться на реальном коде и разрабатывать более эффективные приложения.
 - d.** Код остается открытым, пока он находится в общедоступном репозитории, но лицензия может измениться, если она включена в программное обеспечение с закрытым исходным кодом.
- 2.** Какими двумя из следующих способов Red Hat разрабатывает свои продукты для будущего и взаимодействует с сообществом? (Выберите два.)
 - a.** Спонсируйте и интегрируйте проекты с открытым исходным кодом в проект Fedora, управляемый сообществом.
 - b.** Разработайте специальные инструменты интеграции, доступные только в дистрибутивах Red Hat.
 - c.** Участие в проектах по разработке и продвижению.
 - d.** Переупаковка и повторное лицензирование продуктов сообщества.
- 3.** Какие два утверждения описывают преимущества Linux? (Выберите два.)
 - a.** Linux полностью разрабатывается добровольцами, что делает его недорогой операционной системой.
 - b.** Linux имеет модульную структуру и может быть сконфигурирован как полноценный графический рабочий стол или небольшое устройство.
 - c.** Linux заблокирован в известном состоянии как минимум на один год для каждого выпуска, что упрощает разработку пользовательского программного обеспечения.
 - d.** Linux включает мощный интерфейс командной строки с поддержкой сценариев, упрощающий автоматизацию и подготовку.

РЕЗЮМЕ

В этой главе вы узнали:

- Программное обеспечение с открытым исходным кодом - это программное обеспечение с исходным кодом, которое каждый может свободно использовать, изучать, изменять и делиться.
- Дистрибутив Linux - это устанавливаемая операционная система, созданная на основе ядра Linux и поддерживающая пользовательские программы, и библиотеки.
- Red Hat участвует в поддержке и внесении кода в проекты с открытым исходным кодом, спонсирует и интегрирует программное обеспечение проекта в дистрибутивы, управляемые сообществом, а также стабилизирует программное обеспечение, чтобы предлагать его в качестве поддерживаемых продуктов, готовых к использованию на предприятиях.
- Red Hat Enterprise Linux - это коммерчески поддерживаемый дистрибутив Linux с открытым исходным кодом Red Hat, готовый для предприятий.

ГЛАВА 2

ДОСТУП К КОМАНДНОЙ СТРОКЕ

ЦЕЛЬ

Войдите в систему Linux и выполните простые команды с помощью оболочки(shell).

ЗАДАЧИ

- Войдите в систему Linux на локальной текстовой консоли и выполните простые команды с помощью оболочки.
- Войдите в систему Linux, используя среду рабочего стола GNOME 3, и выполните команды из командной строки в программе терминала.
- Экономьте время, используя ярлыки автозавершения табуляции, истории команд и редактирования команд для выполнения команд в оболочке Bash.

РАЗДЕЛЫ

- Доступ к командной строке (и контрольный опрос)
- Доступ к командной строке с помощью рабочего стола (и упражнение под руководством)
- Выполнение команд с помощью оболочки Bash (и контрольный опрос)

ЛАБОРОТОРНАЯ РАБОТА

Доступ к командной строке

ДОСТУП К КОМАНДНОЙ СТРОКЕ

ЦЕЛИ

После завершения этого раздела вы сможете войти в систему Linux и выполнять простые команды с помощью оболочки.

ВВЕДЕНИЕ В BASH SHELL

Командная строка - это текстовый интерфейс, который можно использовать для ввода инструкций в компьютерную систему. Командная строка Linux предоставляется программой, называемой **shell(оболочкой)**. За прошедшие годы были разработаны различные варианты программы **shell**, при этом у разных пользователей можно настроить использования разных вариантов shell. Однако большинство пользователей придерживаются текущего значения по умолчанию.

Оболочкой по умолчанию для пользователей Red Hat Enterprise Linux является оболочка **GNU Bourne-Again Shell (bash)**. **Bash** - это улучшенная версия одной из самых успешных оболочек, используемых в UNIX-подобных системах, **Bourne Shell (sh)**.

Когда оболочка используется в интерактивном режиме, она отображает строку, когда ожидает команды от пользователя. Это называется приглашением оболочки (**shell prompt**). Когда обычный пользователь запускает оболочку, приглашение по умолчанию заканчивается символом \$, как показано ниже.

```
[user@host ~]$
```

Символ \$ заменяется символом #, если оболочка работает от имени суперпользователя **root**. Это делает более очевидным то, что это оболочка суперпользователя, данная подсказка помогает избежать несчастных случаев и ошибок, которые могут повлиять на всю систему. Приглашение оболочки суперпользователя показано ниже.

```
[root@host ~]#
```

Использование bash для выполнения команд может быть мощным. Оболочка bash предоставляет язык сценариев, который может поддерживать автоматизацию задач. Оболочка имеет дополнительные возможности, которые могут упростить или сделать возможными операции, которые трудно эффективно выполнить с помощью графических инструментов.



ПРИМЕЧАНИЕ

Оболочка **bash** по своей концепции аналогична интерпретатору командной строки **cmd.exe** в последних версиях Microsoft Windows, хотя bash имеет более сложный язык сценариев. Он также похож на **Windows PowerShell** в Windows 7 и Windows Server 2008 R2 и более

поздних версиях. Администраторы Apple Mac, использующие утилиту **Terminal**, могут быть рады отметить, что bash является оболочкой по умолчанию в macOS.

ОСНОВЫ ИСПОЛЬЗОВАНИЯ SHELL (ОБОЛОЧКИ)

Команды, вводимые в командной строке, состоят из трех основных частей:

- **Команда** для запуска
- **Параметры** для настройки поведения команды
- **Аргументы**, которые обычно являются целями команды.

Команда - это имя запускаемой программы. За ним могут следовать одна или несколько опций, которые регулируют поведение команды или ее действия. **Параметры** обычно начинаются с одного или двух дефисов (например, **-a** или **--all**), чтобы отличать их от аргументов. За командами также может следовать один или несколько **аргументов**, которые часто указывают цель, над которой должна работать команда.

Например, команда **usermod -L user01** имеет команду (**usermod**), параметр (**-L**) и аргумент (**user01**). Результатом этой команды является блокировка пароля учетной записи пользователя **user01**.

ВХОД В ЛОКАЛЬНЫЙ КОМПЬЮТЕР

Чтобы запустить оболочку, вам необходимо авторизоваться на компьютере через терминал. **Терминал** - это текстовый интерфейс, используемый для ввода команд и вывода на печать компьютерной системы. Есть несколько способов сделать это.

Компьютер может иметь **аппаратную клавиатуру** и **дисплей** для ввода и вывода, напрямую подключенные к нему. Это физическая консоль машины Linux. Физическая консоль поддерживает несколько виртуальных консолей, **на которых могут работать отдельные терминалы**. Каждая виртуальная консоль поддерживает независимый сеанс входа в систему. Вы можете переключаться между ними, одновременно нажимая **Ctrl + Alt** и функциональную клавишу (с **F1** по **F6**). Большинство виртуальных консолей запускают терминал, предоставляющий текстовое приглашение для входа, и если вы правильно введете свое имя пользователя и пароль, вы войдете в систему и получите приглашение оболочки.

Компьютер может предоставить графическое приглашение для входа на одну из виртуальных консолей. Вы можете использовать данный вариант для входа в графическую среду. Графическая среда также работает на виртуальной консоли. Чтобы получить приглашение оболочки, вы должны запустить программу терминала в графической среде. Приглашение оболочки предоставляется в окне приложения графической программы терминала.



ПРИМЕЧАНИЕ

Многие системные администраторы предпочитают не запускать графическую среду на своих серверах. Это позволяет использовать ресурсы, предназначенные для графической среды службами сервера.

В Red Hat Enterprise Linux 8, если доступна графическая среда, экран входа в систему будет запущен на первой виртуальной консоли, называемой **tty1**. На виртуальных консолях со второй по шестую доступны пять дополнительных текстовых приглашений для входа.

Если вы входите в систему с помощью графического экрана входа в систему, ваша графическая среда запускается на первой виртуальной консоли, которая в настоящее время не используется сеансом входа в систему. Обычно ваш графический сеанс заменяет приглашение входа на вторую виртуальную консоль (**tty2**). Однако, если эта консоль используется активным сеансом текстового входа в систему (а не только приглашением входа в систему), вместо нее используется следующая свободная виртуальная консоль.

Графический экран входа в систему продолжает работать на первой виртуальной консоли (**tty1**). Если вы уже вошли в графический сеанс и входите в систему, как другой пользователь на графическом экране входа в систему или используете пункт меню «Сменить пользователя» для переключения пользователей в графической среде без выхода из системы, для этого пользователя будет запущена другая графическая среда, следующая свободная виртуальная консоль.

Когда вы выходите из графической среды, она закрывается, и физическая консоль автоматически переключается обратно на графический экран входа в систему на первой виртуальной консоли.



ПРИМЕЧАНИЕ

В Red Hat Enterprise Linux 6 и 7 графический экран входа в систему запускается на первой виртуальной консоли, но, когда вы входите в систему, ваша исходная графическая среда **заменяет** экран входа в систему на первой виртуальной консоли вместо запуска на новой виртуальной консоли.

В Red Hat Enterprise Linux 5 и ранее первые шесть виртуальных консолей всегда выдавали текстовые приглашения для входа. Если графическая среда запущена, она находится на виртуальной консоли номер **семь** (доступ к которой осуществляется через **Ctrl + Alt + F7**).

«**Безголовый**» сервер не имеет клавиатуры и постоянно подключенного к нему дисплея. Центр обработки данных может быть заполнен множеством стоек «безголовых» серверов, и отсутствие в каждой из них клавиатуры и дисплея позволяет сэкономить место и расходы. Чтобы позволить администраторам входить в систему, автономный сервер может иметь приглашение входа в систему, предоставляемое его последовательной консолью (*serial console*), работающей на последовательном порту, который подключен к сетевому консольному серверу для удаленного доступа к последовательной консоли.

Последовательная консоль обычно используется для исправления сервера, если его собственная сетевая карта была неправильно настроена и вход в систему через собственное

сетевое соединение стало невозможным. Однако в большинстве случаев доступ к безголовым серверам осуществляется через сеть другими способами.

ПОДКЛЮЧЕНИЕ ДОСТУПА К SHELL ЧЕРЕЗ СЕТЬ

Пользователям и администраторам Linux часто требуется получить доступ к оболочке удаленной системы, подключившись к ней по сети. В современной вычислительной среде многие безголовые серверы на самом деле являются виртуальными машинами или работают, как экземпляры общедоступного или частного облака. Эти системы не являются физическими и не имеют реальных аппаратных консолей. Они могут даже не предоставлять доступ к своей (смоделированной) физической консоли или последовательной консоли.

В Linux наиболее распространенным способом получения приглашения оболочки в удаленной системе является использование **Secure Shell (SSH)**. Большинство систем Linux (включая Red Hat Enterprise Linux) и macOS предоставляют для этой цели программу командной строки **OpenSSH ssh**.

В этом примере пользователь на машине с именем **host** использует **ssh** для входа на компьютер с именем **remotehost** системы Linux в качестве пользователя **remoteuser**:

```
[user@host ~]$ ssh remoteuser@remotehost  
remoteuser@remotehost's password: password  
[remoteuser@remotehost ~]$
```

Команда **ssh** шифрует соединение, чтобы защитить связь от перехвата или перехвата паролей и содержимого.

Некоторые системы (например, новые облачные экземпляры) не позволяют пользователям использовать пароль для входа в систему с помощью **ssh** для большей безопасности. Альтернативный способ аутентификации на удаленном компьютере без ввода пароля - аутентификация с открытым ключом (*public key authentication*).

При использовании этого метода аутентификации у пользователей есть специальный файл идентификации, содержащий закрытый ключ (*private key*), который эквивалентен паролю, и который они хранят в секрете. Их учетная запись на сервере настроена с соответствующим открытым ключом, который не обязательно должен быть секретным. При входе в систему пользователи могут настроить **ssh** для предоставления закрытого ключа, и, если соответствующий открытый ключ установлен в этой учетной записи на этом удаленном сервере, он выполнит вход без запроса пароля.

В следующем примере пользователь с приглашением оболочки на машине **host** входит на удаленный компьютер с именем **remotehost**, как пользователь **remoteuser** с помощью **ssh**, используя аутентификацию с открытым ключом. Параметр **-i** используется для указания файла закрытого ключа пользователя, которым является **mylab.pem**. Соответствующий открытый ключ уже настроен, как авторизованный ключ в учетной записи удаленного пользователя.

```
[user@host ~]$ ssh -i mylab.pem remoteuser@remotehost  
[remoteuser@remotehost ~]$
```

Чтобы указанный вариант подключения сработал, файл закрытого ключа должен быть доступен для чтения только пользователю, которому этот файл принадлежит. В предыдущем примере, где закрытый ключ находится в файле **mylab.pem**, для этого можно использовать команду **chmod 600 mylab.pem**. Как установить права доступа к файлам, более подробно обсуждается в следующей главе.

У пользователей также могут быть настроены **закрытые ключи**, которые проверяются автоматически, но это обсуждение выходит за рамки этого раздела. Ссылки в конце этого раздела содержат ссылки на дополнительную информацию по этой теме.



ПРИМЕЧАНИЕ

При первом входе в систему на новом компьютере вам будет предложено предупреждение от ssh о том, что он не может установить подлинность хоста:

```
[user@host ~]$ ssh -i mylab.pem remoteuser@remotehost
The authenticity of host 'remotehost (192.0.2.42)' can't be established.
ECDSA key fingerprint is 47:bf:82:cd:fa:68:06:ee:d8:83:03:1a:bb:29:14:a3.
Are you sure you want to continue connecting (yes/no)? yes
[remoteuser@remotehost ~]$
```

Каждый раз, когда вы подключаетесь к удаленному хосту с помощью **ssh**, удаленный хост отправляет **ssh** свой ключ хоста для аутентификации и помощи в настройке зашифрованной связи. Команда **ssh** сравнивает это со списком сохраненных ключей хоста, чтобы убедиться, что он не изменился. Если ключ хоста изменился, это может означать, что кто-то пытается притворяться этим хостом, чтобы захватить соединение, что также известно, как **man-in-the-middle attack**. В **SSH** ключи хоста защищают от атак типа «злоумышленник в середине», эти ключи хоста уникальны для каждого сервера, и их необходимо периодически менять и всякий раз, когда есть подозрение на компрометацию.

Вы получите это предупреждение, если на вашем локальном компьютере не сохранен ключ хоста для удаленного хоста. Если вы введете «да (**yes**)», ключ хоста, отправленный удаленным хостом, будет принят и сохранен для использования в будущем. Вход будет продолжен, и вы больше не должны видеть это сообщение при подключении к этому хосту. Если вы введете «нет (**no**)», ключ хоста будет отклонен, и соединение будет закрыто.

Если на локальном компьютере сохранен ключ хоста, и он не совпадает с ключом, фактически отправленным удаленным хостом, соединение будет автоматически закрыто с предупреждением.

ВЫХОД ИЗ СИСТЕМЫ

Когда вы закончите использовать оболочку и захотите выйти, вы можете выбрать один из нескольких способов завершить сеанс. Вы можете ввести команду **exit**, чтобы завершить текущий сеанс оболочки. Либо завершите сеанс, нажав сочетание клавиш **Ctrl+D**.

Ниже приведен пример выхода пользователя из сеанса **SSH**:

```
[remoteuser@remotehost ~]$ exit  
logout  
Connection to remotehost closed.  
[user@host ~]$
```



РЕКОМЕНДАЦИИ

« Справочные страницы **intro(1)**, **bash(1)**, **console(4)**, **pts(4)**, **ssh(1)**, **and ssh-keygen(1)**

*Примечание. Некоторые детали справочной страницы **console(4)**, включая **init(8)** и **inittab(5)**, устарели.*

Дополнительные сведения об OpenSSH и аутентификации с открытым ключом см. В главе *Использование безопасного обмена данными между двумя системами с помощью OpenSSH в Red Hat Enterprise Linux 8 Настройка и управление безопасностью на*

https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/8-beta/html-single/configuring_and_managing_security/index#assembly_usingsecure-communications-with-openssh-configuring-and-managing-security



ПРИМЕЧАНИЕ

Инструкции о том, как читать справочные страницы (**man**) и другую интерактивную справочную документацию, включены в конце следующего раздела.

КОНТРОЛЬНЫЙ ОПРОС

ДОСТУП К КОМАНДНОЙ СТРОКЕ

Выберите правильные ответы на следующие вопросы:

- 1.** Какой термин описывает интерпретатор, выполняющий команды, набранные в виде строк?
 - a. Command
 - b. Console
 - c. Shell
 - d. Terminal

- 2.** Какой термин описывает визуальную подсказку, указывающую, что интерактивная оболочка ожидает, пока пользователь наберет команду?
 - a. Argument
 - b. Command
 - c. Option
 - d. Prompt

- 3.** Какой термин описывает имя запускаемой программы?
 - a. Argument
 - b. Command
 - c. Option
 - d. Prompt

- 4.** Какой термин описывает часть командной строки, которая регулирует поведение команды?
 - a. Argument
 - b. Command
 - c. Option
 - d. Prompt

- 5.** Какой термин описывает часть командной строки, определяющую цель, с которой должна работать команда?
 - a. Argument
 - b. Command
 - c. Option
 - d. Prompt

- 6.** Какой термин описывает аппаратный дисплей и клавиатуру, используемые для взаимодействия с системой?
 - a. Физическая консоль
 - b. Виртуальная консоль
 - c. Оболочка (shell)

d. Терминал

7. Какой термин описывает одну из нескольких логических консолей, каждая из которых может поддерживать независимый сеанс входа в систему?

- a.** Физическая консоль
- b.** Виртуальная консоль
- c.** Оболочка (shell)
- d.** Терминал

8. Какой термин описывает интерфейс, который обеспечивает отображение для вывода и клавиатуру для ввода в сеанс оболочки?

- a.** Консоль (Consol)
- b.** Виртуальная консоль
- c.** Оболочка (Shell)
- d.** Терминал

РЕШЕНИЕ

ДОСТУП К КОМАНДНОЙ СТРОКЕ

Выберите правильные ответы на следующие вопросы:

1. Какой термин описывает интерпретатор, выполняющий команды, набранные в виде строк?
 - a. Command
 - b. Console
 - c. **Shell**
 - d. Terminal

2. Какой термин описывает визуальную подсказку, указывающую, что интерактивная оболочка ожидает, пока пользователь наберет команду?
 - e. Argument
 - f. Command
 - g. Option
 - a. Prompt**

3. Какой термин описывает имя запускаемой программы?
 - a. Argument
 - b. Command**
 - c. Option
 - d. Prompt

4. Какой термин описывает часть командной строки, которая регулирует поведение команды?
 - h. Argument
 - i. Command
 - a. Option**
 - j. Prompt

5. Какой термин описывает часть командной строки, определяющую цель, с которой должна работать команда?
 - a. Argument**
 - k. Command
 - l. Option
 - m. Prompt

6. Какой термин описывает аппаратный дисплей и клавиатуру, используемые для взаимодействия с системой?
 - a. Физическая консоль**
 - n. Виртуальная консоль
 - o. Оболочка (shell)

p. Терминал

7. Какой термин описывает одну из нескольких логических консолей, каждая из которых может поддерживать независимый сеанс входа в систему?

- q.** Физическая консоль
- a. Виртуальная консоль**
- r.** Оболочка (shell)
- s.** Терминал

8. Какой термин описывает интерфейс, который обеспечивает отображение для вывода и клавиатуру для ввода в сеанс оболочки?

- t.** Консоль (Consol)
- u.** Виртуальная консоль
- v.** Оболочка (Shell)
- a. Терминал**

ДОСТУП К КОМАНДНОЙ СТРОКЕ С ПОМОЩЬЮ РАБОЧЕГО СТОЛА

ЦЕЛИ

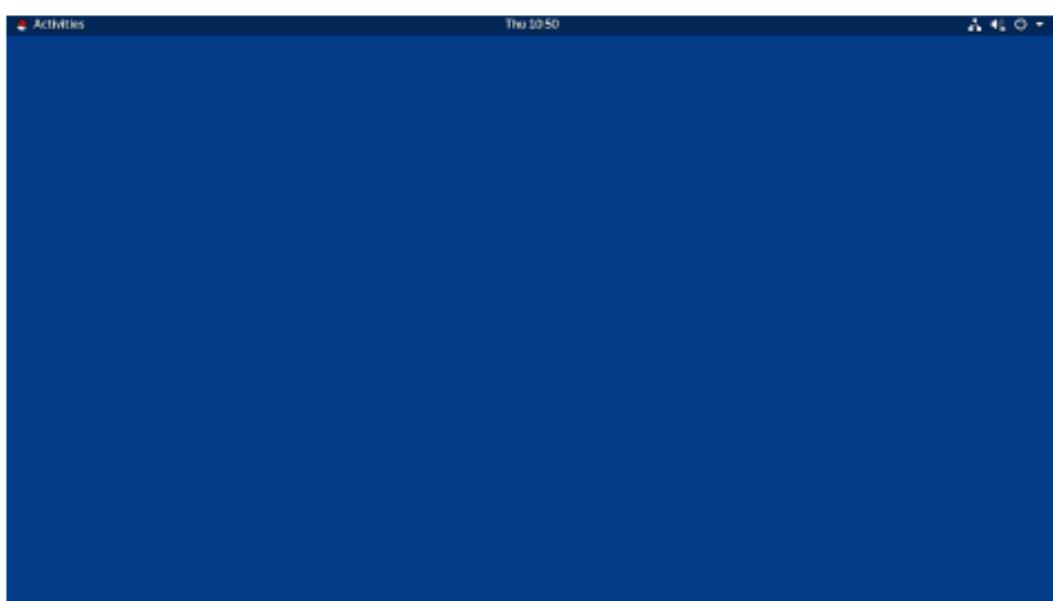
После завершения этого раздела вы сможете войти в систему Linux, используя **среду рабочего стола GNOME 3**, чтобы выполнить команды из командной строки в программе терминала.

ВВЕДЕНИЕ В РАБОЧУЮ СРЕДУ GNOME

Среда рабочего стола (*desktop environment*) - это графический пользовательский интерфейс в системе Linux. Среда рабочего стола по умолчанию в Red Hat Enterprise Linux 8 предоставляет **GNOME 3**. Она предоставляет интегрированный рабочий стол для пользователей и единую платформу разработки поверх графической инфраструктуры, предоставляемой **Wayland** (по умолчанию) или устаревшей системой **X Window**.

GNOME Shell предоставляет основные функции пользовательского интерфейса для среды рабочего стола GNOME. **Приложение GNOME Shell** легко настраивается. В Red Hat Enterprise Linux 8 по умолчанию внешний вид оболочки GNOME соответствует «стандартной (Standard)» теме, которая используется в этом разделе. Red Hat Enterprise Linux 7 по умолчанию выбрала альтернативную тему под названием «Classic», которая была ближе к стилю старых версий GNOME. Любую тему можно постоянно выбирать при входе в систему, щелкнув значок шестеренки рядом с кнопкой «Войти», которая доступна после выбора вашей учетной записи, но до ввода пароля.

Рисунок 2.1: Пустой рабочий стол GNOME 3

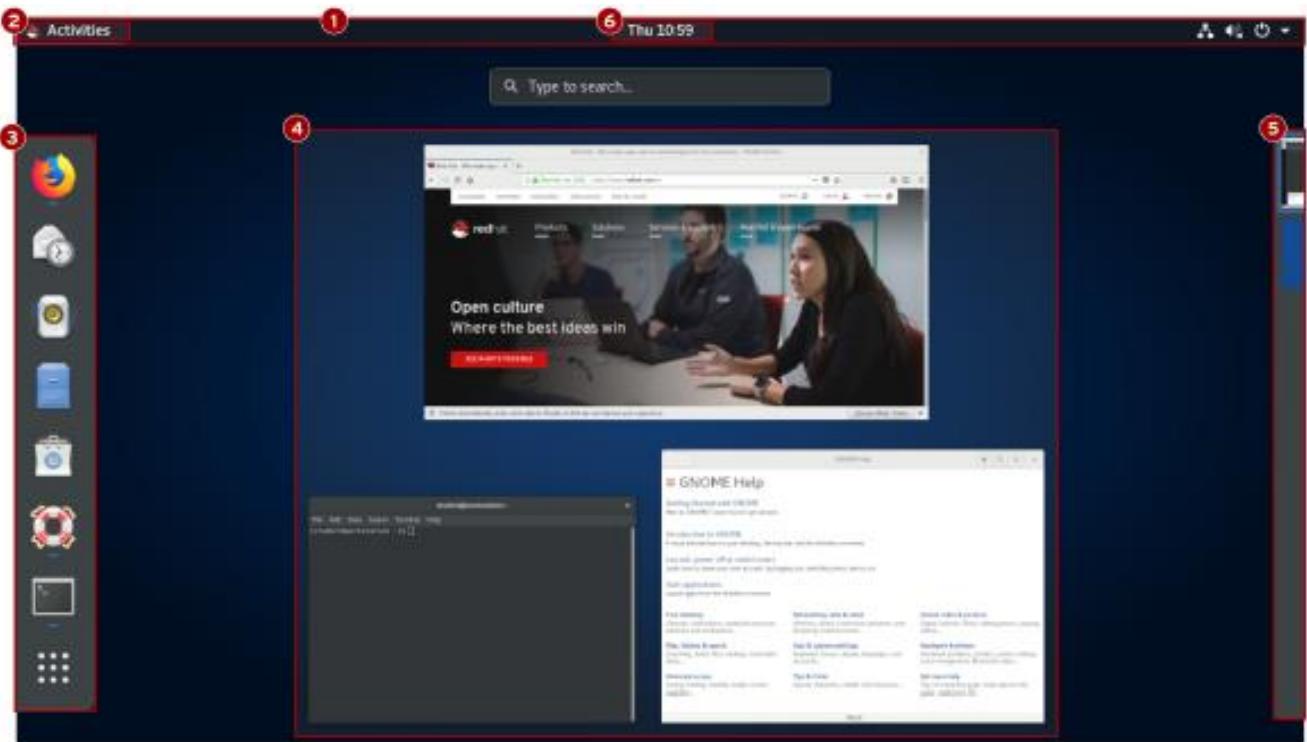


При первом входе в систему в качестве нового пользователя запускается программа начальной настройки, которая помогает настроить основные параметры учетной записи. После этого приложение справки **GNOME** запускается на экране «Начало работы с GNOME (**Getting Started with GNOME**)». Этот экран включает видео и документацию, которая поможет новым пользователям ориентироваться в среде GNOME 3. Вы можете быстро запустить справку GNOME,

щелкнув кнопку «Действия(Activities)» в левой части верхней панели, а также щелкнув значок спасательного круга на панели инструментов в левой части экрана, чтобы запустить ее.

Части оболочки GNOME

Элементы оболочки **GNOME** включают следующие части, как показано на этом снимке экрана оболочки GNOME в режиме обзора действий:

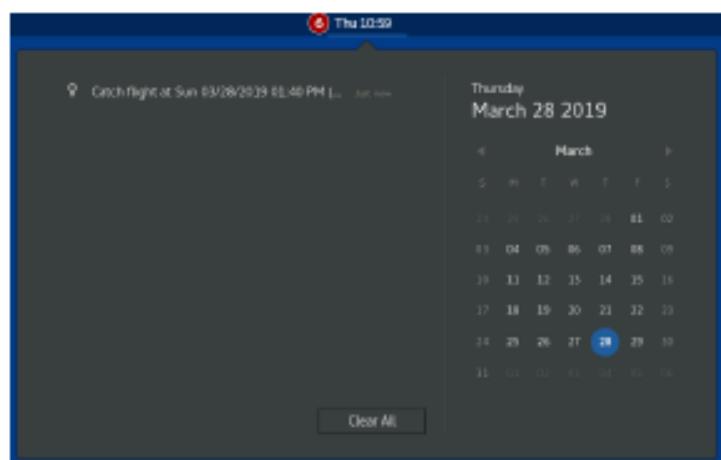


- 1. Верхняя полоса (Top bar):** полоса, которая проходит в верхней части экрана. Он отображается в обзоре действий и в рабочих областях. На верхней панели находится кнопка «Действия (Activities)» и элементы управления громкостью, сетью, доступом к календарю и переключением между методами ввода с клавиатуры (если настроено несколько). Системное меню в правом верхнем углу на верхней панели позволяет регулировать яркость экрана, а также включать и отключать сетевые соединения. В подменю для имени пользователя находятся параметры для настройки параметров учетной записи и выхода из системы. В системном меню также есть кнопки для открытия окна настроек, блокировки экрана или выключения системы.
- 2. Обзор действий (Activities overview):** это специальный режим, который помогает пользователю организовывать окна и запускать приложения. В обзор действий можно войти, нажав кнопку «Действия (Activities)» в верхнем левом углу верхней панели или нажав супер-клавишу. Клавиша **Super** (иногда называемая клавишей **Windows** или клавишей **Command**) находится в нижнем левом углу 104/105-клавишной клавиатуры IBMPC или клавиатуры Apple. Три основных области обзора действий - это **Панель значков** в левой части экрана, **обзор окон в центре экрана** и **селектор рабочего пространства** в правой части экрана. Из обзора действий можно выйти, ничего не делая, снова нажав кнопку «Действия (Activities)» на верхней панели или нажав **Super** или **Esc**.
- 3. Панель значков (Dash):** это настраиваемый список значков избранных приложений пользователя, приложений, которые в настоящее время запущены, и кнопка **решетка** в нижней части **Панели значков**, которую можно использовать для выбора произвольных приложений. Приложения можно запустить, щелкнув один из значков или используя

кнопку **решетки**, чтобы найти менее часто используемое приложение. **Панель значков** также иногда называют док-станцией.

4. **Обзор окон (Windows overview)**: область в центре обзора действий, в которой отображаются эскизы всех окон, активных в текущей рабочей области. Это позволяет легко выводить окна на передний план в загроможденном рабочем пространстве или перемещать в другое рабочее пространство.
5. **Переключатель рабочего пространства (Workspace selector)**: область справа от обзора действий, которая отображает эскизы всех активных рабочих пространств и позволяет выбирать рабочие пространства и перемещать окна из одного рабочего пространства в другое.
6. **Панель сообщений (Message tray)**: панель сообщений позволяет просматривать уведомления, отправляемые приложениями или компонентами системы в GNOME. Если происходит уведомление, обычно оно сначала появляется на короткое время в виде одной строки в верхней части экрана, а в середине верхней панели рядом с часами появляется постоянный индикатор, информирующий пользователя о недавно полученных уведомлениях. Панель сообщений можно открыть для просмотра этих уведомлений, щелкнув часы на верхней панели или нажав **Super + M**. Панель сообщений можно закрыть, щелкнув часы на верхней панели или снова нажав **Esc** или **Super + M**.

Рисунок 2.2: Крупный план открытой панели сообщений



Вы можете просматривать и редактировать сочетания клавиш **GNOME**, используемые в вашей учетной записи. Откройте системное меню в правой части верхней панели. Нажмите кнопку «Настройки (Settings)» в нижней части меню слева (**комментарии при переводе: значок гаечного ключа с отвёрткой**). В открывшемся окне приложения выберите **Устройства (Devices)** → **Клавиатура (Keyboard)** на левой панели. На правой панели отобразятся текущие настройки сочетание клавиш.



ПРИМЕЧАНИЕ

Некоторые сочетания клавиш, такие как функциональные клавиши или супер-клавиша, может быть сложно отправить на виртуальную машину. Это связано с тем, что специальные нажатия клавиш, используемые указанными сочетаниями клавиш, могут перехватываться вашей локальной операционной системой или приложением, которое вы используете для доступа к графическому рабочему столу вашей виртуальной машины.



ВАЖНО

В текущих условиях виртуального обучения Red Hat и в среде самостоятельного обучения использование клавиши **Super** может быть немного сложным. Вероятно, вы не можете просто использовать клавишу **Super** на клавиатуре, потому что она обычно не передается на виртуальную машину в среде класса вашим веб-браузером.

На панели **Ravello Systems** в верхней части окна браузера с правой стороны должен быть значок клавиатуры. Если вы нажмете на нее, откроется экранная клавиатура. Повторное нажатие закроет экранную клавиатуру.

Экранная клавиатура **Ravello** рассматривает **Super**, как клавишу-модификатор, которую часто удерживают при нажатии другой клавиши. Если вы щелкните по нему один раз, он станет желтым, указывая на то, что клавиша удерживается нажатой. Итак, чтобы ввести **Super + M** на экранной клавиатуре **Ravello**, нажмите **Super**, затем нажмите **M**.

Если вы просто хотите нажать и отпустить **Super** на экранной клавиатуре **Ravello**, вам нужно дважды щелкнуть по нему. Первый щелчок «удерживает» клавишу **Super**, а второй щелчок освобождает ее.

Другие клавиши, рассматриваемые экранной клавиатурой **Ravello** как клавиши-модификаторы (например, **Super**) - это **Shift**, **Ctrl**, **Alt** и **Caps**. Клавиши **Esc** и **Menu** обрабатываются как обычные клавиши, а не как клавиши-модификаторы.

РАБОЧИЕ ОБЛАСТИ

Рабочие области (Workspaces) - это отдельные экраны рабочего стола с разными окнами приложений. Их можно использовать для организации рабочей среды, группируя открытые окна приложений по задачам. Например, окна, используемые для выполнения определенного действия по обслуживанию системы (например, установки нового удаленного сервера), могут быть сгруппированы в одной рабочей области, а электронная почта и другие коммуникационные приложения могут быть сгруппированы в другой рабочей области.

Есть два простых метода переключения между рабочими пространствами. Один из способов, возможно, самый быстрый - нажать **Ctrl+Alt+UpArrow** or **Ctrl+Alt+DownArrow** для последовательного переключения между рабочими пространствами. Второй - переключиться на Обзор действий (**Activities**) и щелкнуть нужную рабочую область.

Преимущество использования обзора действий заключается в том, что окна можно щелкать и перетаскивать между рабочими пространствами с помощью селектора рабочего пространства в правой части экрана и обзора окон в центре экрана.



ВАЖНО

Как и в случае с клавишей **Super**, в текущих средах виртуального обучения и самостоятельного обучения Red Hat комбинации клавиш **Ctrl + Alt** обычно не передаются на виртуальную машину в учебной среде вашим веб-браузером.

Вы можете вводить эти комбинации клавиш для переключения рабочих пространств с помощью экранной клавиатуры **Ravello**. Необходимо использовать как минимум два рабочих пространства. Откройте экранную клавиатуру **Ravello** и нажмите **Ctrl, Alt**, а затем либо **UpArrow**, либо **DownArrow**.

Однако в такой учебной среде, как правило, проще избегать сочетаний клавиш и экранной клавиатуры **Ravello**. Переключите рабочие пространства, нажав кнопку «Действия», а затем в селекторе рабочих пространств справа от обзора «Действия (**Activities**)» щелкнув рабочее пространство, на которое вы хотите переключиться.



ПРИМЕЧАНИЕ ПРИ ПЕРЕВОДЕ

Чтобы добавить рабочую область, откройте приложение, перетащите его окно из существующей рабочей места в пустую рабочую область на панель селектора рабочей области. Новое рабочее пространство теперь содержит перемещённое вами окно, а под ним появится новое пустое рабочее пространство. Если на существующий ярлык в панели селектора рабочей области, наложить следующий ярлык открытого приложения, то оба приложения будут находиться в одном рабочем пространстве.

Чтобы удалить рабочее места, просто закройте все открытые на нём окна или переместите их на другие рабочие места.

ЗАПУСК ТЕРМИНАЛА

Чтобы получить приглашение оболочки в GNOME, запустите приложение графического терминала, такое, как Терминал GNOME (**GNOME Terminal**). Есть несколько способов сделать это. Два наиболее часто используемых метода перечислены ниже:

- В обзоре действий выберите Терминал на **Панель значков** (либо в области избранного, либо найдя его с помощью кнопки сетки (внутри группы «Утилиты») или поля поиска в верхней части обзора окон).
- Нажмите комбинацию клавиш **Alt+F2**, чтобы открыть. Выполните команду (**Enter a Command**) и ввести `gnome-terminal`.

Когда открывается окно терминала, отображается приглашение оболочки для пользователя, запустившего графическую программу терминала. Приглашение оболочки и строка заголовка окна терминала указывают текущее имя пользователя, имя хоста и рабочий каталог.

БЛОКИРОВКА ЭКРАНА ИЛИ ВЫХОД ИЗ СИСТЕМЫ

Блокировку экрана или полный выход из системы можно выполнить из системного меню в правом углу верхней панели.

Чтобы заблокировать экран, в системном меню в правом верхнем углу нажмите кнопку блокировки в нижней части меню или нажмите **Super+L** (что может быть проще запомнить, как **Windows+L**). Экран также блокируется, если графический сеанс простоявает в течение нескольких минут.

Появится шторка экрана блокировки, на которой отображается системное время и имя вошедшего в систему пользователя. Чтобы разблокировать экран, нажмите **Enter** или Пробел, чтобы поднять шторку экрана блокировки, затем введите пароль пользователя, указанного на экране блокировки.

Чтобы выйти из системы и завершить текущий сеанс графического входа в систему, выберите системное меню в правом верхнем углу на верхней панели и выберите (Пользователь (**User**)) → Выход (**Log Out**). Откроется окно, предлагающее возможность отменить или подтвердить действие выхода.

ВЫКЛЮЧЕНИЕ ИЛИ ПЕРЕЗАГРУЗКА СИСТЕМЫ

Чтобы выключить систему, в системном меню в правом верхнем углу нажмите кнопку питания в нижней части меню или нажмите **Ctrl+Alt+Del**. В открывшемся диалоговом окне вы можете выбрать «Выключить (**Power Off**)», «Перезагрузить (**Restart**)» устройство или «Отменить операцию(**Cancel**)». Если вы не сделаете выбор, система автоматически выключится через 60 секунд.



РЕКОМЕНДАЦИИ

Справка GNOME: выполнить команду

- **yelp**

Справка GNOME: начало работы с GNOME

- **yelp help:gnome-help/getting-started**

УПРАЖНЕНИЯ С ПОШАГОВЫМИ ИНСТРУКЦИЯМИ

ДОСТУП К КОМАНДНОЙ СТРОКЕ С ПОМОЩЬЮ РАБОЧЕГО СТОЛА

В этом упражнении вы войдете в систему через графический менеджер дисплея как обычный пользователь, чтобы ознакомиться со стандартной средой рабочего стола GNOME, предоставляемой GNOME 3.

ИТОГОВЫЙ РЕЗУЛЬТАТ

Вы должны иметь возможность входить в систему Linux, используя среду рабочего стола GNOME 3, и запускать команды из командной строки в программе терминала.

ПРЕЖДЕ ЧЕМ ВЫ НАЧНЕТЕ

Убедитесь, что виртуальная машина рабочей станции запущена. Выполните следующие задачи на рабочей станции.

1. Войдите на рабочую станцию, как **student**, используя в качестве пароля **student**.
 - 1.1. На рабочей станции на экране входа в GNOME щелкните учетную запись **student**. Введите **student**, когда будет предложено ввести пароль.
 - 1.2. Щелкните войти в систему (**Sign In**).
2. Измените пароль для пользователя **student** со **student** на **55TurnK3y**.



ВАЖНО

Сценарий завершения сбрасывает пароль для пользователя **student** на **student**. Скрипт необходимо выполнить в конце упражнения.

- 2.1. Самый простой подход - открыть Терминал и использовать команду **passwd** в приглашении оболочки.
В виртуальной среде обучения с визуальной клавиатурой дважды нажмите клавишу **Super**, чтобы перейти к обзору действий (**Activities**). Введите команду **terminal** и нажмите **Enter**, чтобы запустить терминал.
- 2.2. В открывшемся окне терминала введите **passwd** в командной строке. Измените пароль **student** с **student** на **55TurnK3y**.

```
[student@workstation ~]$ passwd
Changing password for user student.
Current password: student
New password: 55TurnK3y
Retype new password: 55TurnK3y
passwd: all authentication tokens updated successfully.
```

3. Выйдите из системы и войдите снова как **student**, используя **55TurnK3y** в качестве пароля, чтобы проверить измененный пароль.

- 3.1. Щелкните системное меню в правом верхнем углу.
- 3.2. Выберите «**Пользователь Student (Student User)**» → «**Выйти (LogOut)**».
- 3.3. Щелкните **Выход (Log Out)** в открывшемся диалоговом окне подтверждения.
- 3.4. На экране входа в GNOME щелкните учетную запись **student**. При запросе пароля введите **55TurnK3y**.
- 3.5. Щелкните: **Войти (Sign In)**.

4. Заблокируйте экран.

- 4.1. В системном меню в правом верхнем углу нажмите кнопку экрана блокировки в нижней части меню.

5. Разблокируйте экран.

- 5.1. Нажмите **Enter**, чтобы приподнять шторку экрана блокировки.
- 5.2. В поле **Пароль (Password)** введите в качестве пароля **55TurnK3y**.
- 5.3. Щелкните: **Разблокировать (Unlock)**.

6. Определите, как выключить рабочую станцию из графического интерфейса, но отмените операцию, не выключая систему.

- 6.1. В системном меню в правом верхнем углу нажмите кнопку питания в нижней части меню. Откроется диалоговое окно с возможностью **перезапуска (Restart)** или **выключения (Power Off)** устройства.
- 6.2. Щелкните **Отмена (Cancel)** в открывшемся диалоговом окне.

Завершение

На рабочей станции запустите лабораторный сценарий завершения **cli-desktop**, чтобы завершить это упражнение.

```
[student@workstation ~]$ lab cli-desktop finish
```

На этом пошаговое упражнение завершено.

ВЫПОЛНЕНИЕ КОМАНД С ПОМОЩЬЮ ОБОЛОЧКИ BASH

ЦЕЛИ

После завершения этого раздела вы сможете сэкономить время, выполняя команды из интерпретатора Bash.

ОСНОВНОЙ СИНТАКСИС КОМАНД

GNU Bourne-Again Shell (bash) - это программа, которая интерпретирует команды, введенные пользователем. Каждая строка, вводимая в оболочку, может состоять максимум из трех частей: **команды**, **параметров** (которые обычно начинаются с - или - -) и **аргументов**. Каждое слово, набранное в оболочке, отделяется друг от друга пробелами. Команды - это **названия программ**, установленных в системе. У каждой команды есть свои **параметры** и **аргументы**.

Когда вы будете готовы выполнить команду, нажмите клавишу **Enter**. Введите каждую команду в отдельной строке. Вывод команды отображается до появления следующего приглашения оболочки.

```
[user@host]$ whoami  
user  
[user@host]$
```

Если вы хотите ввести более одной команды в одной строке, используйте **точку с запятой** (;) в качестве разделителя команд. Точка с запятой является членом класса символов, называемых **метасимволами**, которые имеют особое значение в bash. В этом случае вывод обеих команд будет отображаться до появления следующего приглашения оболочки.

В следующем примере показано, как объединить две команды (**command1** и **command2**) в командной строке.

```
[user@host]$ command1;command2
```

ПРИМЕРЫ ПРОСТОЙ КОМАНДЫ

Команда **date** отображает текущую дату и время. Он также может использоваться суперпользователем для установки системных часов. Аргумент, который начинается со знака плюс (+), указывает строку формата для команды даты.

```
[user@host ~]$ date  
Sat Jan 26 08:13:50 IST 2019
```

```
[user@host ~]$ date +%R
```

08:13

```
[user@host ~]$ date +%x
```

01/26/2019

Команда **passwd** изменяет собственный пароль пользователя. Прежде чем изменение будет разрешено, необходимо указать исходный пароль для учетной записи. По умолчанию **passwd** настроен на требование надежного пароля, состоящего из строчных букв, прописных букв, цифр и символов, и не основан на словарном слове. Суперпользователь **root** может использовать команду **passwd** для изменения паролей других пользователей.

```
[user@host ~]$ passwd
Changing password for user user.
Current password: old_password
New password: new_password
Retype new password: new_password
passwd: all authentication tokens updated successfully.
```

Linux не требует расширения имен файлов для классификации файлов по типу. Команда **file** просматривает начало содержимого файла и отображает его тип. Классифицируемые файлы передаются команде в качестве аргументов.

```
[user@host ~]$ file /etc/passwd
/etc/passwd: ASCII text
[user@host ~]$ file /bin/passwd
/bin/passwd: setuid ELF 64-bit LSB shared object, x86-64, version 1
(SYSV), dynamically linked, interpreter /lib64/ld-linux-x86-64.so.2,
for GNU/Linux 3.2.0, BuildID[sha1]=a3637110e27e9a48dced9f38b4ae43388d32d0e4,
stripped
[user@host ~]$ file /home
/home: directory
```

ПРОСМОТР СОДЕРЖАНИЯ ФАЙЛОВ

Одна из самых простых и часто используемых команд в Linux - это **cat**. Команда **cat** позволяет вам создавать один или несколько файлов, просматривать содержимое файлов, объединять содержимое из нескольких файлов и перенаправлять содержимое файла на терминал или файлы. В примере показано, как просмотреть содержимое файла **/etc/passwd**.

```
[user@host ~]$ cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
```

```
adm:x:3:4:adm:/var/adm:/sbin/nologin  
...output omitted...
```

Используйте следующую команду для отображения содержимого нескольких файлов.

```
[user@host ~]$ cat file1 file2  
Hello World!!  
Introduction to Linux commands.
```

Некоторые файлы очень длинные и могут занимать больше места для отображения, чем предусмотрено терминалом. Команда **cat** не отображает содержимое файла в виде страниц. Команда **less** отображает по одной странице файла за раз и позволяет прокручивать в удобное для вас время.

Команда **less** позволяет листать вперед и назад файлы, длина которых превышает размер одного окна терминала. Используйте клавиши со **стрелкой вверх** и **вниз** для прокрутки вверх и вниз. Нажмите **Q**, чтобы выйти из команды.

Команды **head** и **tail** отображают начало и конец файла соответственно. По умолчанию эти команды отображают 10 строк файла, но обе имеют параметр **-n**, который позволяет указать разное количество строк. Файл для отображения передается этим командам в качестве аргумента.

```
[user@host ~]$ head /etc/passwd  
root:x:0:0:root:/root:/bin/bash  
bin:x:1:1:bin:/bin:/sbin/nologin  
daemon:x:2:2:daemon:/sbin:/sbin/nologin  
adm:x:3:4:adm:/var/adm:/sbin/nologin  
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin  
sync:x:5:0:sync:/sbin:/bin/sync  
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown  
halt:x:7:0:halt:/sbin:/sbin/halt  
mail:x:8:12:mail:/var/spool/mail:/sbin/nologin  
operator:x:11:0:operator:/root:/sbin/nologin  
[user@host ~]$ tail -n 3 /etc/passwd  
gdm:x:42:42::/var/lib/gdm:/sbin/nologin  
gnome-initial-setup:x:977:977::/run/gnome-initial-setup/:/sbin/nologin  
avahi:x:70:70:Avahi mDNS/DNS-SD Stack:/var/run/avahi-daemon:/sbin/nologin
```

Команда **wc** считает строки, слова и символы в файле. Требуется опция **-l**, **-w** или **-c** для отображения только количества строк, слов или символов соответственно.

```
[user@host ~]$ wc /etc/passwd  
45 102 2480 /etc/passwd  
[user@host ~]$ wc -l /etc/passwd ; wc -l /etc/group  
45 /etc/passwd
```

```
70 /etc/group  
[user@host ~]$ wc -c /etc/group /etc/hosts  
966 /etc/group  
516 /etc/hosts  
1482 total
```

ЗАВЕРШЕНИЕ КОМАНДЫ КЛАВИШЕЙ TAB

Завершение с помощью табуляции позволяет пользователю быстро заполнять команды или имена файлов после того, как они наберут достаточно текста в приглашении, чтобы сделать его уникальным. Если набранные символы не уникальны, при двойном нажатии клавиши **Tab** отображаются все команды, которые начинаются с уже набранных символов.

```
[user@host ~]$ pas1 Tab+Tab  
passwd      paste      pasusponder  
  
[user@host ~]$ pass2 Tab  
[user@host ~]$ passwd  
Changing password for user user.  
Current password:
```

1. Нажмите **Tab** дважды
2. Нажмите **Tab** один раз

Завершение табуляции может использоваться для завершения имен файлов при вводе их в качестве аргументов команд. Когда нажата клавиша **Tab**, она завершает, как можно большую часть имени файла. Повторное нажатие клавиши **Tab** приводит к тому, что оболочка (**shell**) выводит список всех файлов, соответствующих текущему шаблону. Введите дополнительные символы, пока имя не станет уникальным, затем используйте завершение табуляции для завершения команды.

```
[user@host ~]$ ls /etc/pas1 Tab  
[user@host ~]$ ls /etc/passwd2 Tab  
passwd  passwd-
```

1 и 2 Нажмите **Tab** один раз.

Аргументы и параметры можно сопоставить с автозавершением табуляции для многих команд. Команда **useradd** используется суперпользователем, **root**, для создания дополнительных пользователей в системе. У данной команды есть много параметров, которые можно использовать для управления её поведением. Завершение с помощью табуляции после частичного варианта можно использовать для завершения варианта без большого набора текста.

```
[root@host ~]# useradd --❶Tab+Tab
--base-dir      --groups        --no-log-init    --shell
--comment       --help          --non-unique     --skel
--create-home   --home-dir     --no-user-group  --system
--defaults      --inactive     --password      --uid
--expiredate   --key          --root          --user-group
--gid           --no-create-home --selinux-user
[root@host ~]# useradd --
```

❶ Нажмите **Tab** дважды.

ПРОДОЛЖЕНИЕ ДЛИННОЙ КОМАНДЫ НА ДРУГОЙ СТРОКЕ

Команды, которые поддерживают множество параметров и аргументов, могут быстро увеличиваться в размерах и автоматически прокручиваются оболочкой **Bash**. Как только курсор достигнет правого края окна, команда перейдет к следующей строке. Чтобы облегчить читаемость команды, вы можете разбить ее так, чтобы она умещалась более чем в одной строке.

Для этого добавьте символ обратной косой черты (\) в качестве последнего символа в строке. Это указывает оболочке игнорировать символ новой строки и обрабатывать следующую строку, как если бы она была частью текущей строки. Оболочка **Bash** начнет следующую строку с приглашения продолжения, обычно с символа «**больше**» (>), который указывает, что строка является продолжением предыдущей строки. Вы можете сделать это более одного раза.

```
[user@host]$ head -n 3 \
> /usr/share/dict/words \
> /usr/share/dict/linux.words
==> /usr/share/dict/words <==
1080
10-point
10th

==> /usr/share/dict/linux.words <==
1080
10-point
10th
[user@host ~]$
```

ИСТОРИЯ КОМАНД

Команда **history** отображает список ранее выполненных команд с префиксом номера команды.

Восклицательный знак (!) - это метасимвол, который используется для расширения предыдущих команд без необходимости их повторного ввода. Команда **!number** заменяется на

команду, соответствующую указанному номеру. Команда `!string` заменяется самой последней командой, которая начинается с указанной строки.

```
[user@host ~]$ history
...output omitted...
23 clear
24 who
25 pwd
26 ls /etc
27 uptime
28 ls -l
29 date
30 history
[user@host ~]$ !ls
ls -l
total 0
drwxr-xr-x. 2 user user 6 Mar 29 21:16 Desktop
...output omitted...
[user@host ~]$ !26
ls /etc
abrt           hosts          pulse
adjtime        hosts.allow    purple
aliases        hosts.deny    qemu-ga
...output omitted...
```

Клавиши со стрелками можно использовать для навигации по предыдущим командам в истории оболочки. Стрелка вверх (**UpArrow**) редактирует предыдущую команду в списке истории. Клавиша Стрелка вниз (**DownArrow**) редактирует следующую команду в списке истории. **LeftArrow** и **RightArrow** перемещают курсор влево и вправо в текущей команде из списка истории, чтобы вы могли редактировать ее перед запуском.

Вы можете использовать либо **Esc +.** или **Alt +.** комбинация клавиш для вставки последнего слова предыдущей команды в текущее положение курсора. Повторное использование комбинации клавиш заменит этот текст последним словом даже более ранних команд в истории. **Alt +.** комбинация клавиш особенно удобна, потому что вы можете удерживать **Alt** и нажать точку (.) многоократно, чтобы легко проходить все более ранние команды.

РЕДАКТИРОВАНИЕ КОМАНДНОЙ СТРОКИ

При интерактивном использовании **bash** имеет функцию редактирования из командной строки. Это позволяет пользователю использовать команды текстового редактора, чтобы перемещаться внутри и изменять текущую набираемую команду. Использование клавиш со стрелками для перемещения внутри текущей команды и перехода по истории команд было представлено ранее в этом сеансе. В следующей таблице представлены более мощные команды редактирования.

Полезные сочетания клавиш для редактирования в командной строке

СОЧЕТАНИЕ КЛАВИШ	ОПИСАНИЕ
Ctrl + A	Перейти к началу командной строки.
Ctrl + E	Перейти в конец командной строки.
Ctrl + U	Очистить от курсора до начала командной строки.
Ctrl + K	Очистить от курсора до конца командной строки.
Ctrl + стрелка влево	Перейти к началу предыдущего слова в командной строке.
Ctrl + стрелка вправо	Перейти к концу следующего слова в командной строке.
Ctrl + R	Найдите шаблон в списке истории команд.

Доступно несколько других команд редактирования командной строки, но это наиболее полезные команды для новых пользователей. Остальные команды можно найти на странице руководства **bash(1)**.



РЕКОМЕНДАЦИИ

Страницы **man bash(1)**, **date(1)**, **file(1)**, **cat(1)**, **more(1)**, **less(1)**, **head(1)**, **passwd(1)**, **tail(1)**, и **wc(1)**

КОНТРОЛЬНЫЙ ОПРОС

ВЫПОЛНЕНИЕ КОМАНД С ПОМОЩЬЮ ОБОЛОЧКИ BASH

Выберите правильные ответы на следующие вопросы:

1. Какое **сочетание клавиш** или команда **Bash** выполняет переход к началу предыдущего слова в командной строке?
 - a. Нажатие **Ctrl + LeftArrow** (стрелку влево)
 - b. Нажатие **Ctrl + K**
 - c. Нажатие **Ctrl + A**
 - d. **!string**
 - e. **!number**

2. Какое **сочетание клавиш** или команда **Bash** разделяет команды в одной строке?
 - a. Нажатие **Tab**
 - b. **history**
 - c. ;
 - d. **!string**
 - e. Нажатие **Esc +.**

3. Какое **сочетание клавиш** или команда **Bash** используется для удаления символов от курсора до конца командной строки?
 - a. Нажатие **Ctrl + LeftArrow** (стрелку влево)
 - b. Нажатие **Ctrl + K**
 - c. Нажатие **Ctrl + A**
 - d. ;
 - e. Нажатие **Esc +.**

4. Какое **сочетание клавиш** или команда **Bash** используется для повторного выполнения последней команды путем сопоставления имени команды?
 - a. Нажатие **Tab**
 - b. **!number**
 - c. **!string**
 - d. **History**
 - e. Нажатие **Esc +.**

5. Какое **сочетание клавиш** или команда **Bash** используется для завершения команд, имен файлов и параметров?
 - a. ;
 - b. **!number**
 - c. **history**

- d. Нажатие **Tab**
 - e. Нажатие **Esc+**.
6. Какое **сочетание клавиш** или команда **Bash** повторно выполняет определенную команду в списке истории?
- a. Нажатие **Tab**
 - b. **!number**
 - c. **!string**
 - d. **history**
 - e. Нажатие **Esc +.**
7. Какое **сочетание клавиш** или команда **Bash** переходит в начало командной строки?
- a. **!number**
 - b. **!string**
 - c. Нажатие **Ctrl + LeftArrow** (стрелку влево)
 - d. Нажатие **Ctrl + K**
 - e. Нажатие **Ctrl + A**
8. Какое **сочетание клавиш** или команда **Bash** отображает список предыдущих команд?
- a. Нажатие **Tab**
 - b. **!string**
 - c. **!number**
 - d. **history**
 - e. Нажатие **Esc +.**
9. Какое **сочетание клавиш** или команда **Bash** копирует последний аргумент предыдущих команд?
- a. Нажатие **Ctrl + K**
 - b. Нажатие **Ctrl + A**
 - c. **!number**
 - d. Нажатие **Esc +.**

РЕШЕНИЕ

ВЫПОЛНЕНИЕ КОМАНД С ПОМОЩЬЮ ОБОЛОЧКИ BASH

Выберите правильные ответы на следующие вопросы:

1. Какое **сочетание клавиш** или команда **Bash** выполняет переход к началу предыдущего слова в командной строке?
 - a. Нажатие **Ctrl + LeftArrow** (стрелку влево)
 - b. Нажатие **Ctrl + K**
 - c. Нажатие **Ctrl + A**
 - d. **!string**
 - e. **!number**
2. Какое **сочетание клавиш** или команда **Bash** разделяет команды в одной строке?
 - a. Нажатие Tab
 - b. history
 - c. ;
 - d. !string
 - e. Нажатие Esc +.
3. Какое **сочетание клавиш** или команда **Bash** используется для удаления символов от курсора до конца командной строки?
 - a. Нажатие **Ctrl + LeftArrow** (стрелку влево)
 - b. Нажатие **Ctrl + K**
 - c. Нажатие **Ctrl + A**
 - d. ;
 - e. Нажатие Esc +.
4. Какое **сочетание клавиш** или команда **Bash** используется для повторного выполнения последней команды путем сопоставления имени команды?
 - a. Нажатие Tab
 - b. **!number**
 - c. **!string**
 - d. History
 - e. Нажатие Esc +.
5. Какое **сочетание клавиш** или команда **Bash** используется для завершения команд, имен файлов и параметров?
 - a. ;
 - b. **!number**
 - c. history
 - d. Нажатие **Tab**

e. Нажатие Esc+.

6. Какое **сочетание клавиш** или команда **Bash** повторно выполняет определенную команду в списке истории?

- a. Нажатие Tab
- b. **!number**
- c. !string
- d. history
- e. Нажатие Esc +.

7. Какое **сочетание клавиш** или команда **Bash** переходит в начало командной строки?

- a. !number
- b. !string
- c. Нажатие Ctrl + LeftArrow (стрелку влево)
- d. Нажатие Ctrl + K
- e. Нажатие **Ctrl + A**

8. Какое **сочетание клавиш** или команда **Bash** отображает список предыдущих команд?

- a. Нажатие Tab
- b. !string
- c. !number
- d. **history**
- e. Нажатие Esc +.

9. Какое **сочетание клавиш** или команда **Bash** копирует последний аргумент предыдущих команд?

- a. Нажатие Ctrl + K
- b. Нажатие Ctrl + A
- c. !number
- d. Нажатие Esc +.

ЛАБОРАТОРНАЯ РАБОТА

ДОСТУП К КОМАНДНОЙ СТРОКЕ

КОНТРОЛЬНЫЙ СПИСОК РАБОТЫ

В этой лабораторной работе вы будете использовать оболочку **Bash** для выполнения команд.

ЦЕЛИ

- Успешно выполнять простые программы с помощью командной строки оболочки Bash.
- Выполнять команды, используемые для определения типов файлов и отображения частей текстовых файлов.
- Практикуйтесь в использовании **сочетание клавиш** или команд **Bash** для более эффективного повторения команд или их частей.

ПРЕЖДЕ ЧЕМ ВЫ НАЧНЕТЕ

Войдите на рабочую станцию, как пользователь **student**, используя в качестве пароля слово **student**. На рабочей станции запустите сценарий **lab cli-review start**, чтобы настроить чистую лабораторную среду. Сценарий также копирует файл **zcat** в домашний каталог ученика.

```
[student@workstation ~]$ labcli-reviewstart
```

1. Используйте команду **date** для отображения текущего времени и даты.
2. Отобразите текущее время в 12-часовом формате (например, 11:42:11 AM). Подсказка: строка формата, отображающая этот вывод, **%r**.
3. Что это за файл **/home/student/zcat**? Читается ли это людьми?
4. Используйте команду **wc** и **сочетание клавиш** Bash, чтобы отобразить размер **zcat**.
5. Отобразите первые 10 строк **zcat**.
6. Отобразите последние 10 строк файла **zcat**.
7. Точно повторите предыдущую команду, нажав три или меньше клавиш.
8. Повторите предыдущую команду, но используйте параметр **-n 20**, чтобы отобразить последние 20 строк в файле. Используйте редактирование в командной строке, чтобы сделать это с минимальным количеством нажатий клавиш.
9. Используйте историю оболочки, чтобы снова запустить команду **date +% r**.

ОЦЕНКА

На рабочей станции запустите сценарий **lab cli-review grade**, чтобы подтвердить успех этого упражнения.

```
[student@workstation ~]$ lab cli-review grade
```

ОКОНЧАНИЕ

На рабочей станции запустите сценарий **lab cli-review finish**, чтобы завершить лабораторную работу.

РЕШЕНИЕ

ДОСТУП К КОМАНДНОЙ СТРОКЕ

КОНТРОЛЬНЫЙ СПИСОК РАБОТЫ

В этой лабораторной работе вы будете использовать оболочку **Bash** для выполнения команд.

ЦЕЛИ

- Успешно выполнять простые программы с помощью командной строки оболочки Bash.
- Выполнять команды, используемые для определения типов файлов и отображения частей текстовых файлов.
- Практикуйтесь в использовании **сочетание клавиш** или команд **Bash** для более эффективного повторения команд или их частей.

ПРЕЖДЕ ЧЕМ ВЫ НАЧНЕТЕ

Войдите на рабочую станцию, как пользователь **student**, используя в качестве пароля слово **student**. На рабочей станции запустите сценарий **lab cli-review start**, чтобы настроить чистую лабораторную среду. Сценарий также копирует файл **zcat** в домашний каталог ученика.

```
[student@workstation ~]$ labcli-reviewstart
```

1. Используйте команду **date** для отображения текущего времени и даты.

```
[student@workstation ~]$ date  
Thu Jan 22 10:13:04 PDT 2019
```

2. Отобразите текущее время в 12-часовом формате (например, 11:42:11 AM). Подсказка: строка формата, отображающая этот вывод, **%r**.
Используйте аргумент **+%r** с командой **date**, чтобы отобразить текущее время в 12-часовом формате.

```
[student@workstation ~]$ date +%r  
10:14:07 AM
```

3. Что это за файл **/home/student/zcat**? Читается ли это людьми?
Используйте команду **file**, чтобы определить тип файла

```
[student@workstation ~]$ file zcat
zcat: POSIX shell script, ASCII text executable
```

4. Используйте команду **wc** и **сочетание клавиш**Bash, чтобы отобразить размер **zcat**.

Команду **wc** можно использовать для отображения количества строк, слов и байтов в сценарии **zcat**. Вместо того, чтобы повторно вводить имя файла, используйте сочетание клавиш истории **Bash Esc +.** (одновременно нажмите клавиши **Esc** и **.**), чтобы повторно использовать аргумент из предыдущей команды.

```
[student@workstation ~]$ wc Esc+.
[student@workstation ~]$ wc zcat
 51  299 1983 zcat
```

5. Отобразите первые 10 строк **zcat**.

Команда **head** отображает начало файла. Попробуйте использовать сочетание клавиш **Esc +.** снова.

```
[student@workstation ~]$ head Esc+.
[student@workstation ~]$ head zcat
#!/bin/sh
# Uncompress files to standard output.

# Copyright (C) 2007, 2010-2018 Free Software Foundation, Inc.

# This program is free software; you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation; either version 3 of the License, or
# (at your option) any later version.
```

6. Отобразите последние 10 строк файла **zcat**.

Используйте команду **tail** для отображения последних 10 строк файла **zcat**.

```
[student@workstation ~]$ tail Esc+.
[student@workstation ~]$ tail zcat
With no FILE, or when FILE is -, read standard input.

Report bugs to <bug-gzip@gnu.org>.

case $1 in
--help)    printf '%s\n' "$usage"    || exit 1;;
--version) printf '%s\n' "$version" || exit 1;;
esac

exec gzip -cd "$@"
```

7. Точно повторите предыдущую команду, нажав три или меньше клавиш.

Точно повторите предыдущую команду. Либо нажмите клавишу со **стрелкой вверх** один раз, чтобы прокрутить историю команд назад по одной команде, а затем нажмите **Enter** (используется два нажатия клавиш), либо введите команду быстрого доступа **!!** а затем нажмите **Enter** (используется три нажатия клавиш), чтобы выполнить самую последнюю команду в истории команд. (Попробуйте оба варианта.)

```
[student@workstation]$ !!
tail zcat
with no FILE, or when FILE is -, read standard input.

Report bugs to <bug-gzip@gnu.org>.

case $1 in
--help)   printf '%s\n' "$usage"  || exit 1;;
--version) printf '%s\n' "$version" || exit 1;;
esac

exec gzip -cd "$@"
```

8. Повторите предыдущую команду, но используйте параметр **-n 20**, чтобы отобразить последние 20 строк в файле. Используйте редактирование в командной строке, чтобы сделать это с минимальным количеством нажатий клавиш.

Стрелка вверх отображает предыдущую команду. **Ctrl+A** заставляет курсор переместиться в начало строки. **Ctrl + стрелка вправо** переходит к следующему слову, затем добавляет параметр **-n 20** и нажимает **Enter**, чтобы выполнить команду.

```
[student@workstation ~]$ tail -n 20 zcat
-1, --list      list compressed file contents
-q, --quiet     suppress all warnings
-r, --recursive  operate recursively on directories
-s, --suffix=SUF use suffix SUF on compressed files
--asynchronous synchronous output (safer if system crashes, but slower)
-t, --test       test compressed file integrity
-v, --verbose    verbose mode
--help          display this help and exit
--version       display version information and exit

With no FILE, or when FILE is -, read standard input.

Report bugs to <bug-gzip@gnu.org>.

case $1 in
--help)   printf '%s\n' "$usage"  || exit 1; exit;;
--version) printf '%s\n' "$version" || exit 1; exit;;
esac

exec gzip -cd "$@"
```

9. Используйте историю оболочки, чтобы снова запустить команду **date +%r**.

Используйте команду **history**, чтобы отобразить список предыдущих команд и определить конкретную команду **date**, которую нужно выполнить. Используйте **!number** для запуска команды, где *number* - это номер команды, который следует использовать из выходных данных команды **history**. Обратите внимание, что история вашей оболочки может отличаться от следующего примера. Определите номер команды для использования на основе результатов вашей собственной команды истории.

```
[student@workstation ~]$ history
1 date
2 date +%r
3 file zcat
4 wc zcat
5 head zcat
6 tail zcat
7 tail -n 20 zcat
8 history
[student@workstation ~]$ !2
date +%r
10:49:56 AM
```

ОЦЕНКА

На рабочей станции запустите сценарий **lab cli-review grade**, чтобы подтвердить успех этого упражнения.

```
[student@workstation ~]$ lab cli-review grade
```

ОКОНЧАНИЕ

На рабочей станции запустите сценарий **lab cli-review finish**, чтобы завершить лабораторную работу.

```
[student@workstation ~]$ lab cli-review finish
```

На этом лабораторная работа завершена.

РЕЗЮМЕ

В этой главе вы узнали:

- **Оболочка Bash** - это интерпретатор команд, который предлагает интерактивным пользователям указать команды Linux.
- Многие команды имеют параметр **--help**, отображающий сообщение об использовании или экран.
- Использование рабочих пространств упрощает организацию нескольких окон приложений.
- Кнопка «Действия (**Activities**)», расположенная в верхнем левом углу верхней панели, обеспечивает режим обзора, который помогает пользователю организовывать окна и запускать приложения.
- Команда **file** просматривает начало содержимого файла и отображает его тип.
- Команды **head** и **tail** отображают начало и конец файла соответственно.
- Вы можете использовать **Tab**(табуляция) для завершения имен файлов при вводе их в качестве аргументов команд.

ГЛАВА 3

УПРАВЛЕНИЕ ФАЙЛАМИ ИЗ КОМАНДНОЙ СТРОКИ

ЦЕЛЬ

Копируйте, перемещайте, создавайте, удаляйте и систематизируйте файлы во время работы из оболочки Bash.

ЗАДАЧИ

- Объяснение, как Linux организует файлы, назначение различных каталогов в иерархии файловой системы.
- Укажите расположение файлов относительно текущей рабочей директории и по абсолютному местоположению, определите и измените вашу рабочую директорию, перечислите содержимое директорий.
- Создавать, копировать, перемещать и удалять файлы и каталоги.
- Сделайте так, чтобы несколько имен файлов ссылались на один и тот же файл, используя жесткие ссылки и символические (или «soft») ссылки.
- Эффективно запускайте команды, влияющие на многие файлы, с помощью функций сопоставления шаблонов оболочки **Bash**.

РАЗДЕЛЫ

- Описание концепций иерархии файловой системы Linux (и контрольный опрос)
- Указание файлов по имени (и контрольный опрос)
- Управление файлами с помощью инструментов командной строки (и упражнения с пошаговыми инструкциями)
- Установление связей между файлами (и упражнения с пошаговыми инструкциями)
- Сопоставление имен файлов с помощью расширений оболочки (и контрольный опрос)

ЛАБОРОТОРНАЯ РАБОТА

Управление файлами из командной строки

ОПИСАНИЕ КОНЦЕПЦИЙ ИЕРАРХИИ ФАЙЛОВОЙ СИСТЕМЫ LINUX

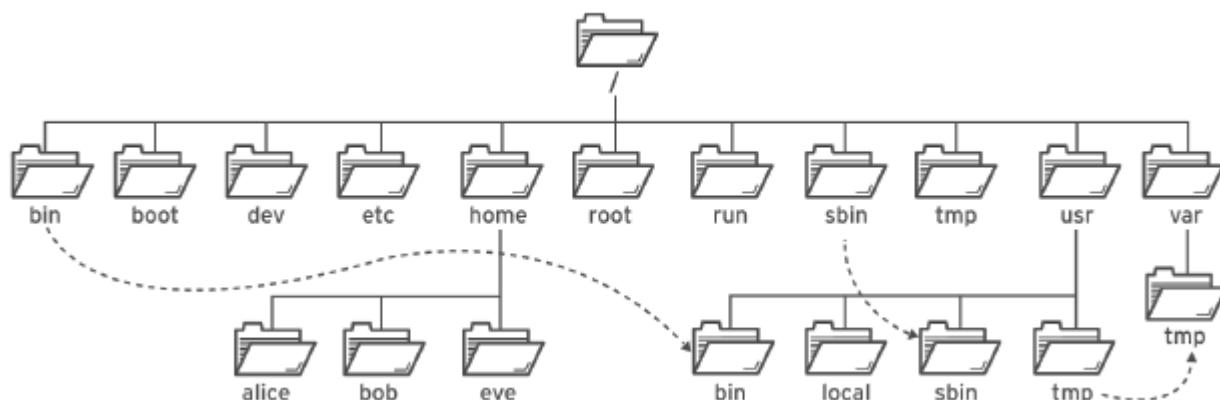
ЗАДАЧИ

После завершения этого раздела вы сможете описать, как Linux организует файлы и назначение различных каталогов в иерархии файловой системы.

ИЕРАРХИЯ ФАЙЛОВЫХ СИСТЕМ

Все файлы в системе Linux хранятся в файловых системах, которые организованы в единое перевернутое дерево каталогов, известное, как **иерархия файловой системы**. Это дерево перевернуто, потому что считается, что корень дерева находится на вершине иерархии, а ветви каталогов и подкаталогов простираются ниже корня.

Рисунок 3.1: Важные каталоги файловой системы в Red Hat Enterprise Linux 8



Каталог `/` - это корневой каталог на вершине иерархии файловой системы. Символ `/` также используется в качестве разделителя каталогов в именах файлов. Например, если `etc` является подкаталогом каталога `/`, вы можете ссылаться на этот каталог, как `/etc`. Аналогичным образом, если в каталоге `/etc` содержится файл с именем `issue`, вы можете ссылаться на этот файл как на `/etc/issue`.

Подкаталоги `/` используются в стандартных целях для организации файлов по типу и назначению. Это упрощает поиск файлов. Например, в корневом каталоге подкаталог `/boot` используется для хранения файлов, необходимых для загрузки системы.



ПРИМЕЧАНИЕ

Следующие термины помогают описать содержимое каталога файловой системы:

- Статическое (*static*) содержимое остается неизменным до тех пор, пока не будет явно отредактировано или перенастроено.
- Динамическое или переменное (*dynamic* или *variable*) содержимое может быть изменено или добавлено активными процессами.

- Постоянное (*persistent*) содержимое остается после перезагрузки, как и параметры конфигурации.
- Содержимое во время работы системы (*runtime*) - это содержимое, зависящее от процесса или системы, которое удаляется при перезагрузке.

В следующей таблице перечислены некоторые из наиболее важных каталогов в системе по имени и назначению.

Важные каталоги Red Hat Enterprise Linux

МЕСТО РАСПОЛОЖЕНИЯ	ЦЕЛЬ
/usr	Установленное программное обеспечение, общие библиотеки, включаемые файлы и данные программы только для чтения. Важные подкаталоги включают: <ul style="list-style-type: none"> • /usr/bin: пользовательские команды. • /usr/sbin: команды системного администрирования • /usr/local: программное обеспечение, настроенное локально.
/etc	Файлы конфигурации, относящиеся к этой системе.
/var	Переменные данные, относящиеся к этой системе, которые должны сохраняться между загрузками. Файлы, которые динамически изменяются, такие как базы данных, каталоги кеша, файлы журналов, буферные документы принтера и содержимое веб-сайтов, можно найти в /var.
/run	Находятся файлы текущего состояния приложений, запущенных с момента последней загрузки. Сюда, среди прочего, входят файлы идентификаторов процессов и файлы блокировки. Содержимое этого каталога воссоздается при перезагрузке. Этот каталог консолидирует /var/run и /var/lock из более ранних версий Red Hat Enterprise Linux.
/home	В домашних каталогах обычные пользователи хранят свои личные данные и файлы конфигурации.
/root	Домашний каталог для административного суперпользователя, root .
/tmp	Доступное для записи пространство для временных файлов. Файлы, которые не использовались, не изменялись или не изменялись в течение 10 дней, автоматически удаляются из этого каталога. Существует еще один временный каталог, /var/tmp, в нем файлы, которые не использовались, не изменялись или не изменялись более 30 дней, удаляются автоматически.
/boot	Файлы, необходимые для запуска процесса загрузки.
/dev	Содержит специальные файлы устройств, которые используются системой для доступа к оборудованию.



ВАЖНО

В RedHatEnterpriseLinux 7 и новее четыре старых каталога в `/` имеют идентичное содержимое с их аналогами, расположенными в `/usr`:

- `/bin` и `/usr/bin`
- `/sbin` и `/usr/sbin`
- `/lib` и `/usr/lib`
- `/lib64` и `/usr/lib64`

В более ранних версиях RedHatEnterpriseLinux это были отдельные каталоги, содержащие разные наборы файлов.

В RedHatEnterpriseLinux 7 и новее каталоги в `/` представляют собой символические ссылки на соответствующие каталоги в `/usr`.



РЕКОМЕНДАЦИИ

Справочная страница [hier\(7\)](#)

Страница функций [UsrMove \(UsrMovefeaturepage\) из Fedora 17](#)

<https://fedoraproject.org/wiki/Features/UsrMove>

КОНТРОЛЬНЫЙ ОПРОС

ОПИСАНИЕ КОНЦЕПЦИЙ ИЕРАРХИИ ФАЙЛОВОЙ СИСТЕМЫ LINUX

Выберите правильные ответы на следующие вопросы:

1. Какой каталог содержит постоянные системные данные конфигурации?

- a. /etc
- b. /root
- c. /run
- d. /usr

2. Какой каталог является верхним в иерархии файловой системы?

- a. /etc
- b. /
- c. /home/root
- d. /root

3. В каком каталоге находятся домашние каталоги пользователей?

- a. /
- b. /home
- c. /root
- d. /user

4. В каком каталоге находятся временные файлы?

- a. /tmp
- b. /trash
- c. /run
- d. /var

5. Какой каталог содержит динамические данные, например, для баз данных и веб-сайтов?

- a. /etc
- b. /run
- c. /usr
- d. /var

6. Какой каталог является домашним каталогом административного суперпользователя?

- a. /etc
- b. /
- c. /home/root
- d. /root

7. В каком каталоге находятся штатные команды и утилиты?

- a. /commands**
- b. /run**
- c. /usr/bin**
- d. /usr/sbin**

8. В каком каталоге хранятся непостоянные данные времени выполнения процесса?

- a. /tmp**
- b. /etc**
- c. /run**
- d. /var**

9. В каком каталоге находятся установленные программы и библиотеки?

- a. /etc**
- b. /lib**
- c. /usr**
- d. /var**

РЕШЕНИЕ

ОПИСАНИЕ КОНЦЕПЦИЙ ИЕРАРХИИ ФАЙЛОВОЙ СИСТЕМЫ LINUX

Выберите правильные ответы на следующие вопросы:

1. Какой каталог содержит постоянные системные данные конфигурации?

- a. /etc**
- b. /root**
- c. /run**
- d. /usr**

2. Какой каталог является верхним в иерархии файловой системы?

- a. /etc**
- b. /**
- c. /home/root**
- d. /root**

3. В каком каталоге находятся домашние каталоги пользователей?

- a. /**
- b. /home**
- c. /root**
- d. /user**

4. В каком каталоге находятся временные файлы?

- a. /tmp
- b. /trash
- c. /run
- d. /var

5. Какой каталог содержит динамические данные, например, для баз данных и веб-сайтов?

- a. /etc
- b. /run
- c. /usr
- d. /var

6. Какой каталог является домашним каталогом административного суперпользователя?

- a. /etc
- b. /
- c. /home/root
- d. /root

7. В каком каталоге находятся штатные команды и утилиты?

- a. /commands
- b. /run
- c. /usr/bin
- d. /usr/sbin

8. В каком каталоге хранятся непостоянные данные времени выполнения процесса?

- a. /tmp
- b. /etc
- c. /run
- d. /var

9. В каком каталоге находятся установленные программы и библиотеки?

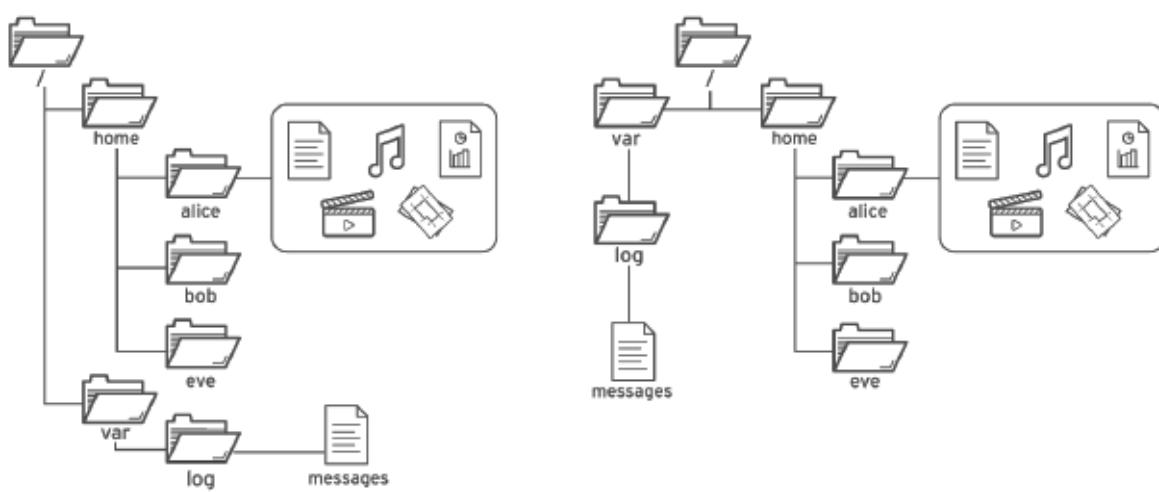
- a. /etc
- b. /lib
- c. /usr
- d. /var

УКАЗАНИЕ ФАЙЛОВ ПО ИМЕНИ

После завершения этого раздела вы сможете указать расположение файлов относительно текущего рабочего каталога и по абсолютному расположению, определить и изменить рабочий каталог, а также составить список содержимого каталогов.

АБСОЛЮТНЫЕ ПУТИ И ОТНОСИТЕЛЬНЫЕ ПУТИ

Рисунок 3.2: Обычный просмотр файлового браузера (слева) эквивалентен просмотру сверху вниз (справа).



Путь к файлу или каталогу указывает его уникальное расположение в файловой системе. После пути к файлу проходит один или несколько именованных подкаталогов, разделенных косой чертой (/), пока не будет достигнуто место назначения. **Каталоги**, также называемые **папками**, содержат другие файлы и другие подкаталоги. На них можно ссылаться так же, как на файлы.



ВАЖНО

Пробел допустим, как часть имени файла Linux. Однако оболочка также использует пробелы для разделения параметров и аргументов в командной строке. Если вы вводите команду, которая включает файл, в имени которого есть пробел, оболочка может неверно истолковать команду и предположить, что вы хотите начать с нового имени файла или другого аргумента в этом месте.

Этого можно избежать, заключив имена файлов в кавычки. Однако, если вам не нужно использовать пробелы в именах файлов, может быть проще просто избежать их использования.

Абсолютные пути

Абсолютный путь - это полное имя, указывающее точное расположение файлов в иерархии файловой системы. Он начинается с корневого каталога (/) и определяет каждый подкаталог, который необходимо пройти, чтобы добраться до конкретного файла. Каждый файл в файловой системе имеет уникальное имя абсолютного пути, распознаваемое простым правилом: **имя пути с косой чертой (/)** в качестве первого символа является абсолютным именем пути. Например, абсолютный путь к файлу журнала системных сообщений - **/var/log/messages**. Абсолютные имена путей могут быть длинными, поэтому файлы также могут располагаться относительно текущего рабочего каталога для приглашения оболочки.

Текущий рабочий каталог и относительные пути

Когда пользователь входит в систему и открывает командное окно, исходным местоположением обычно является домашний каталог пользователя. Системные процессы также имеют начальный каталог. При необходимости пользователи и процессы переходят в другие каталоги; термины «рабочий каталог (*working directory*)» или «текущий рабочий каталог (*current working directory*)» относятся к их текущему местоположению.

Как и абсолютный путь, **относительный путь** определяет уникальный файл, указывая только путь, необходимый для доступа к файлу из рабочего каталога. Распознавание имен относительных путей следует простому правилу: имя пути с любым отличным от косой черты в качестве первого символа является относительным именем пути. Пользователь в каталоге **/var** может ссылаться на файл журнала сообщений, как на **log/messages**.

Файловые системы Linux, включая, помимо прочего, ext4, XFS, GFS2 и GlusterFS, чувствительны к регистру. Создание **FileCase.txt** и **filecase.txt** в одном каталоге приводит к получению двух уникальных файлов.

Файловые системы, отличные от Linux, могут работать иначе. Например, VFAT, Microsoft NTFS и Apple HFS + сохраняют регистр. Хотя эти файловые системы не чувствительны к регистру, они отображают имена файлов с исходными заглавными буквами, которые использовались при создании файла. Следовательно, если вы попытаетесь создать файлы в предыдущем примере в файловой системе VFAT, оба имени будут обрабатываться как указывающие на один и тот же файл, а не на два разных файла.

НАВИГАЦИОННЫЕ ПУТИ

Команда **pwd** отображает полный путь к текущему рабочему каталогу для этой оболочки. Это может помочь вам определить синтаксис для доступа к файлам с использованием имен относительных путей. Команда **ls** выводит список содержимого каталога для указанного каталога или, если каталог не указан, для текущего рабочего каталога.

```
[user@host ~]$ pwd
/home/user
[user@host ~]$ ls
Desktop Documents Downloads Music Pictures Public Templates Videos
[user@host ~]$
```

Используйте команду **cd**, чтобы изменить текущий рабочий каталог вашей оболочки (**shell**). Если вы не укажете аргументы для команды, она перейдет в ваш домашний каталог.

В следующем примере сочетание абсолютных и относительных путей используется с командой **cd** для изменения текущего рабочего каталога оболочки.

```
[user@host ~]$ pwd  
/home/user  
[user@host ~]$ cd Videos  
[user@host Videos]$ pwd  
/home/user/Videos  
[user@host Videos]$ cd /home/user/Documents  
[user@host Documents]$ pwd  
/home/user/Documents  
[user@host Documents]$ cd  
[user@host ~]$ pwd  
/home/user  
[user@host ~]$
```

Как вы можете видеть в предыдущем примере, приглашение оболочки по умолчанию также отображает последний компонент абсолютного пути к текущему рабочему каталогу. Например, для **/home/user/Videos** отображаются только, как папка **Videos**. В приглашении отображается символ тильды (~), если текущий рабочий каталог является вашим домашним каталогом.

Команда **touch** обычно обновляет метку времени файла на текущую дату и время, не изменяя ее иным образом. Это полезно для создания пустых файлов, которые можно использовать на практике, потому что «прикосновение (touching)» к имени файла, которое не существует, вызывает создание файла. В следующем примере команда **touch** создает файлы практики в подкаталогах **Documents** и **Videos**.

```
[user@host ~]$ touch Videos/blockbuster1.ogg  
[user@host ~]$ touch Videos/blockbuster2.ogg  
[user@host ~]$ touch Documents/thesis_chapter1.odf  
[user@host ~]$ touch Documents/thesis_chapter2.odf  
[user@host ~]$
```

Команда **ls** имеет несколько опций для отображения атрибутов файлов. Наиболее распространенными и полезными являются **-l** (формат длинного списка), **-a** (все файлы, включая скрытые) и **-R** (рекурсивный, для включения содержимого всех подкаталогов).

```
[user@host ~]$ ls -l  
total 15  
drwxr-xr-x. 2 user user 4096 Feb 7 14:02 Desktop  
drwxr-xr-x. 2 user user 4096 Jan 9 15:00 Documents  
drwxr-xr-x. 3 user user 4096 Jan 9 15:00 Downloads
```

```
drwxr-xr-x. 2 user user 4096 Jan 9 15:00 Music
drwxr-xr-x. 2 user user 4096 Jan 9 15:00 Pictures
drwxr-xr-x. 2 user user 4096 Jan 9 15:00 Public
drwxr-xr-x. 2 user user 4096 Jan 9 15:00 Templates
drwxr-xr-x. 2 user user 4096 Jan 9 15:00 Videos
[user@host ~]$ ls -la
total 15
drwx----- 16 user user 4096 Feb 8 16:15 .
drwxr-xr-x. 6 root root 4096 Feb 8 16:13 ..
-rw----- 1 user user 22664 Feb 8 00:37 .bash_history
-rw-r--r-- 1 user user 18 Jul 9 2013 .bash_logout
-rw-r--r-- 1 user user 176 Jul 9 2013 .bash_profile
-rw-r--r-- 1 user user 124 Jul 9 2013 .bashrc
drwxr-xr-x. 4 user user 4096 Jan 20 14:02 .cache
drwxr-xr-x. 8 user user 4096 Feb 5 11:45 .config
drwxr-xr-x. 2 user user 4096 Feb 7 14:02 Desktop
drwxr-xr-x. 2 user user 4096 Jan 9 15:00 Documents
drwxr-xr-x. 3 user user 4096 Jan 25 20:48 Downloads
drwxr-xr-x. 11 user user 4096 Feb 6 13:07 .gnome2
drwx----- 2 user user 4096 Jan 20 14:02 .gnome2_private
-rw----- 1 user user 15190 Feb 8 09:49 .ICEauthority
drwxr-xr-x. 3 user user 4096 Jan 9 15:00 .local
drwxr-xr-x. 2 user user 4096 Jan 9 15:00 Music
drwxr-xr-x. 2 user user 4096 Jan 9 15:00 Pictures
drwxr-xr-x. 2 user user 4096 Jan 9 15:00 Public
drwxr-xr-x. 2 user user 4096 Jan 9 15:00 Templates
drwxr-xr-x. 2 user user 4096 Jan 9 15:00 Videos
[user@host ~]$
```

Два специальных каталога вверху списка относятся к **текущему каталогу** (.) И **родительскому каталогу** (..). Эти специальные каталоги существуют в каждом каталоге в системе. Вы обнаружите их полезность, когда начнете использовать команды управления файлами.



ВАЖНО

Имена файлов, начинающиеся с точки (.), указывают на скрытые (**hidden**) файлы; вы не можете увидеть их в обычном режиме с помощью **ls** и других команд. Это не функция безопасности. Скрытые файлы предохраняют необходимые файлы конфигурации пользователя от загромождения домашних каталогов. Многие команды обрабатывают скрытые файлы только с определенными параметрами командной строки, предотвращая случайное копирование конфигурации одного пользователя в другие каталоги или пользователей.

Для защиты содержимого файла от неправильного просмотра требуется использование прав доступа к файлу.

```
[user@host ~]$ ls -R
.:
Desktop Documents Downloads Music Pictures Public Templates Videos

./Desktop:

./Documents:
thesis_chapter1.odf thesis_chapter2.odf

./Downloads:

./Music:

./Pictures:

./Public:

./Templates:

./Videos:
blockbuster1.ogg blockbuster2.ogg
[user@host ~]$
```

У команды **cd** есть много опций. Некоторые из них настолько полезны, что их стоит практиковать с самого начала и часто использовать. Команда **cd -** переход в предыдущий каталог; где пользователь ранее находился в текущем каталоге. Следующий пример иллюстрирует это поведение при чередовании двух каталогов, что полезно при обработке серии похожих задач.

```
[user@host ~]$ cd Videos
[user@host Videos]$ pwd
/home/user/Videos
[user@host Videos]$ cd /home/user/Documents
[user@host Documents]$ pwd
/home/user/Documents
[user@host Documents]$ cd -
[user@host Videos]$ pwd
/home/user/Videos
[user@host Videos]$ cd -
[user@host Documents]$ pwd
/home/user/Documents
[user@host Documents]$ cd -
[user@host Videos]$ pwd
/home/user/Videos
[user@host Videos]$ cd
[user@host ~]$
```

Команда **cd ..** использует скрытый каталог **..** для перехода на один уровень вверх к родительскому каталогу без необходимости знать точное имя родителя. Другой скрытый каталог

(.) Определяет текущий каталог для команд, в котором текущее местоположение является аргументом источника или назначения, избегая необходимости вводить имя абсолютного пути к каталогу.

```
[user@host Videos]$ pwd  
/home/user/Videos  
[user@host Videos]$ cd .  
[user@host Videos]$ pwd  
/home/user/Videos  
[user@host Videos]$ cd ..  
[user@host ~]$ pwd  
/home/user  
[user@host ~]$ cd ..  
[user@host home]$ pwd  
/home  
[user@host home]$ cd ..  
[user@host /]$ pwd  
/  
[user@host /]$ cd  
[user@host ~]$ pwd  
/home/user  
[user@host ~]$
```



РЕКОМЕНДАЦИИ

infolibc 'разрешение имени файла' (**filenameresolution**)
(Справочно-руководство библиотеки GNU C)
• Раздел 11.2.2 Разрешение имени файла (File name resolution)
Справочные страницы **bash(1)**, **cd(1)**, **ls(1)**, **pwd(1)**, **unicode(7)** и **utf-8(7)**
UTF-8 и Unicode
<http://www.utf-8.com/>

КОНТРОЛЬНЫЙ ОПРОС

УКАЗАНИЕ ФАЙЛОВ ПО ИМЕНИ

Выберите правильные ответы на следующие вопросы:

1. Какая команда используется для возврата в домашний каталог текущего пользователя, если текущий рабочий каталог - **/tmp**, а их домашний каталог - **/home/user**?
 - a. **cd**
 - b. **cd ..**
 - c. **cd .**
 - d. **cd ***
 - e. **cd /home**
2. Какая команда отображает абсолютный путь к текущему местоположению?
 - a. **cd**
 - b. **pwd**
 - c. **ls ~**
 - d. **ls -d**
3. Какая команда всегда будет возвращать вас в рабочий каталог, который использовался до текущего рабочего каталога?
 - a. **cd -**
 - b. **cd -p**
 - c. **cd ~**
 - d. **cd ..**
4. Какая команда всегда будет изменять рабочий каталог на два уровня выше текущего?
 - a. **cd ~**
 - b. **cd ../**
 - c. **cd ../../..**
 - d. **cd -u2**
5. Какая команда выводит список файлов в текущем местоположении, используя длинный формат и включая скрытые файлы?
 - a. **llong ~**
 - b. **ls -a**
 - c. **ls -l**
 - d. **ls -al**
6. Какая команда всегда изменяет рабочий каталог на **/bin**?

- a. cd bin
- b. cd /bin
- c. cd ~bin
- d. cd -bin

7. Какая команда всегда изменяет рабочий каталог на родительский для текущего местоположения?

- a. cd ~
- b. cd ..
- c. cd ../../..
- d. cd -u1

8. Какая команда изменит рабочий каталог на /tmp, если текущий рабочий каталог - /home/student?

- a. cd tmp
- b. cd ..
- c. cd ../../tmp
- d. cd ~tmp

РЕШЕНИЕ

УКАЗАНИЕ ФАЙЛОВ ПО ИМЕНИ

Выберите правильные ответы на следующие вопросы:

1. Какая команда используется для возврата в домашний каталог текущего пользователя, если текущий рабочий каталог - **/tmp**, а их домашний каталог - **/home/user**?
 - a. **cd**
 - b. **cd ..**
 - c. **cd .**
 - d. **cd ***
 - e. **cd /home**

2. Какая команда отображает абсолютный путь к текущему местоположению?
 - a. **cd**
 - b. **pwd**
 - c. **ls ~**
 - d. **ls -d**

3. Какая команда всегда будет возвращать вас в рабочий каталог, который использовался до текущего рабочего каталога?
 - a. **cd -**
 - b. **cd -p**
 - c. **cd ~**
 - d. **cd ..**

4. Какая команда всегда будет изменять рабочий каталог на два уровня выше текущего?
 - a. **cd ~**
 - b. **cd ../**
 - c. **cd ../../..**
 - d. **cd -u2**

5. Какая команда выводит список файлов в текущем местоположении, используя длинный формат и включая скрытые файлы?
 - a. **llong ~**
 - b. **ls -a**
 - c. **ls -l**
 - d. **ls -al**

6. Какая команда всегда изменяет рабочий каталог на **/bin**?

- a. cd bin
- b. cd /bin
- c. cd ~bin
- d. cd -bin

7. Какая команда всегда изменяет рабочий каталог на родительский для текущего местоположения?

- a. cd ~
- b. cd ..
- c. cd ../../..
- d. cd -u1

8. Какая команда изменит рабочий каталог на **/tmp**, если текущий рабочий каталог - **/home/student**?

- a. cd tmp
- b. cd ..
- c. cd ../../tmp
- d. cd ~tmp

УПРАВЛЕНИЕ ФАЙЛАМИ С ПОМОЩЬЮ ИНСТРУМЕНТОВ КОМАНДНОЙ СТРОКИ

ЗАДАЧИ

После завершения этого раздела вы сможете создавать, копировать, перемещать и удалять файлы и каталоги.

УПРАВЛЕНИЕ ФАЙЛАМИ КОМАНДНОЙ СТРОКИ

Рисунок 3.2: Управление файлами с помощью инструментов командной строки

Чтобы управлять файлами, вы должны иметь возможность создавать, удалять, копировать и перемещать их. Вам также необходимо логически организовать их в каталоги, которые вы также должны иметь возможность создавать, удалять, копировать и перемещать.

В следующей таблице приведены некоторые из наиболее распространенных команд управления файлами. В оставшейся части этого раздела способы использования этих команд обсуждаются более подробно.

Общие команды управления файлами

ДЕЙСТВИЕ	СИНТАКСИС КОМАНД
Создать каталог	<code>mkdir directory</code>
Скопируйте файл	<code>cp file new-file</code>
Скопируйте каталог и его содержимое	<code>cp -r directory new-directory</code>
Переместить или переименовать файл или каталог	<code>mv file new-file</code>
Удалить файл	<code>rm file</code>
Удалить каталог, содержащий файлы	<code>rm -r directory</code>
Удалить пустой каталог	<code>rmdir directory</code>

Создание каталогов

Команда `mkdir` создает один или несколько каталогов или подкаталогов. В качестве аргументов он принимает список путей к каталогам, которые вы хотите создать.

Команда `mkdir` завершится ошибкой, если каталог уже существует, или если вы пытаетесь создать подкаталог в каталоге, который не существует. Параметр `-p` (родительский (*parent*)) создает отсутствующие родительские каталоги для запрошенного места назначения. Используйте команду `mkdir -p` с осторожностью, потому что орфографические ошибки могут создать непредусмотренные каталоги без создания сообщений об ошибках.

В следующем примере представьте, что вы пытаетесь создать каталог с именем «Watched» в каталоге «Videos», но случайно пропустили букву «s» в поле «Videos» в команде `mkdir`.

```
[user@host ~]$ mkdir Video/Watched
```

```
mkdir: cannot create directory `Video/Watched': No such file or directory
```

Команда **mkdir** завершилась ошибкой, потому что название каталога **Videos** было написано с ошибкой, а каталог **Video** не существует. Если бы вы использовали команду **mkdir** с параметром **-p**, каталог **Video** был бы создан, что не было тем, что вы планировали, и подкаталог **Watched** был бы создан в этом неправильном каталоге.

После правильного написания родительского каталога **Videos** создание подкаталога **Watched** будет успешным.

```
[user@host ~]$ mkdir Videos/Watched
[user@host ~]$ ls -R Videos
Videos/:
blockbuster1.ogg  blockbuster2.ogg  Watched

Videos/Watched:
```

В следующем примере файлы и каталоги организованы в каталоге **/home/user/Documents**. Используйте команду **mkdir** и список имен каталогов, разделенных пробелами, чтобы создать несколько каталогов.

```
[user@host ~]$ cd Documents
[user@host Documents]$ mkdir ProjectX ProjectY
[user@host Documents]$ ls
ProjectX  ProjectY
```

Используйте команду **mkdir -p** и разделенные пробелами относительные пути для каждого имени подкаталога, чтобы создать несколько родительских (parent) каталогов с подкаталогами.

```
[user@host Documents]$ mkdir -p Thesis/Chapter1 Thesis/Chapter2 Thesis/Chapter3
[user@host Documents]$ cd
[user@host ~]$ ls -R Videos Documents
Documents:
ProjectX ProjectY Thesis

Documents/ProjectX:

Documents/ProjectY:

Documents/Thesis:
Chapter1 Chapter2 Chapter3

Documents/Thesis/Chapter1:

Documents/Thesis/Chapter2:

Documents/Thesis/Chapter3:

Videos:
blockbuster1.ogg blockbuster2.ogg Watched

Videos/Watched:
```

Последняя команда **mkdir** создала три подкатаога **ChapterN** с помощью одной команды. Параметр **-p** создал отсутствующий родительский каталог **Thesis**.

Копирование файлов

Команда **cp** копирует файл, создавая новый файл либо в текущем каталоге, либо в указанном каталоге. Он также может копировать несколько файлов в каталог.



ПРЕДУПРЕЖДЕНИЕ

Если целевой файл уже существует, команда **cp** перезаписывает файл.

```
[user@host ~]$ cd Videos
[user@host Videos]$ cp blockbuster1.ogg blockbuster3.ogg
[user@host Videos]$ ls -l
total 0
-rw-rw-r--. 1 user user    0 Feb  8 16:23 blockbuster1.ogg
-rw-rw-r--. 1 user user    0 Feb  8 16:24 blockbuster2.ogg
-rw-rw-r--. 1 user user    0 Feb  8 16:34 blockbuster3.ogg
drwxrwxr-x. 2 user user 4096 Feb  8 16:05 Watched
[user@host Videos]$
```

При копировании нескольких файлов одной командой последним аргументом должен быть каталог. Скопированные файлы сохраняют свои исходные имена в новом каталоге. Если в целевом каталоге существует файл с таким же именем, существующий файл перезаписывается. По умолчанию **cp** не копирует каталоги; он их игнорирует.

В следующем примере перечислены два каталога: **Thesis** и **ProjectX**. Только последний аргумент, **ProjectX**, действителен в качестве места назначения. Каталог **Thesis** игнорируется.

```
[user@host Videos]$ cd .. /Documents
[user@host Documents]$ cp thesis_chapter1.odf thesis_chapter2.odf Thesis ProjectX
cp: omitting directory `Thesis'
[user@host Documents]$ ls Thesis ProjectX
ProjectX:
thesis_chapter1.odf  thesis_chapter2.odf

Thesis:
Chapter1  Chapter2  Chapter3
```

В первой команде **cp** не удалось скопировать каталог **Thesis**, но файлы **thesis_chapter1.odf** и **thesis_chapter2.odf** скопировались удачно.

Если вы хотите скопировать файл в текущий рабочий каталог, вы можете использовать специальный символ текущей директории **..**:

```
[user@host ~]$ cp /etc/hostname ..
[user@host ~]$ cat hostname
host.example.com
[user@host ~]$
```

Используйте команду копирования с параметром **-r** (рекурсивный), чтобы скопировать каталог **Thesis** и его содержимое в каталог **ProjectX**.

```
[user@host Documents]$ cp -r Thesis ProjectX
[user@host Documents]$ ls -R ProjectX
ProjectX:
Thesis  thesis_chapter1.odf  thesis_chapter2.odf

ProjectX/Thesis:
Chapter1  Chapter2  Chapter3

ProjectX/Thesis/Chapter1:

ProjectX/Thesis/Chapter2:
thesis_chapter2.odf

ProjectX/Thesis/Chapter3:
```

Перемещение файлов

Команда **mv** перемещает файлы из одного места в другое. Если вы думаете об абсолютном пути к файлу, как о его полном имени, перемещение файла фактически то же самое, что и переименование файла. Содержимое файла остается без изменений.

Используйте команду **mv**, чтобы переименовать файл.

```
[user@host Videos]$ cd ../Documents
[user@host Documents]$ ls -l thesis*
-rw-rw-r--. 1 user user 0 Feb  6 21:16 thesis_chapter1.odf
-rw-rw-r--. 1 user user 0 Feb  6 21:16 thesis_chapter2.odf
[user@host Documents]$ mv thesis_chapter2.odf thesis_chapter2_reviewed.odf
[user@host Documents]$ ls -l thesis*
-rw-rw-r--. 1 user user 0 Feb  6 21:16 thesis_chapter1.odf
-rw-rw-r--. 1 user user 0 Feb  6 21:16 thesis_chapter2_reviewed.odf
```

Используйте команду **mv**, чтобы *переместить* файл в другой каталог.

```
[user@host Documents]$ ls Thesis/Chapter1
[user@host Documents]$
[user@host Documents]$ mv thesis_chapter1.odf Thesis/Chapter1
[user@host Documents]$ ls Thesis/Chapter1
thesis_chapter1.odf
[user@host Documents]$ ls -l thesis*
-rw-rw-r--. 1 user user 0 Feb  6 21:16 thesis_chapter2_reviewed.odf
```

Удаление файлов и каталогов

Команда **rm** удаляет файлы. По умолчанию **rm** не удаляет каталоги, содержащие файлы, если вы не добавите параметр **-r** или **--recursive**.



ВАЖНО

Нет ни функции восстановления из командной строки, ни «корзины», из которой вы можете восстанавливать файлы, подготовленные для удаления.

Перед удалением файла или каталога рекомендуется проверить текущий рабочий каталог.

```
[user@host Documents]$ pwd
/home/student/Documents
```

Используйте команду **rm**, чтобы удалить один файл из вашего рабочего каталога.

```
[user@host Documents]$ ls -l thesis*
-rw-rw-r--. 1 user user 0 Feb  6 21:16 thesis_chapter2_reviewed.odf
[user@host Documents]$ rm thesis_chapter2_reviewed.odf
[user@host Documents]$ ls -l thesis*
ls: cannot access 'thesis*': No such file or directory
```

Если вы попытаетесь использовать команду **rm** для удаления каталога без использования параметра **-r**, команда завершится ошибкой.

```
[user@host Documents]$ rm Thesis/Chapter1
rm: cannot remove `Thesis/Chapter1': Is a directory
```

Используйте команду **rm -r**, чтобы удалить подкаталог и его содержимое.

```
[user@host Documents]$ ls -R Thesis
Thesis/:
Chapter1 Chapter2 Chapter3

Thesis/Chapter1:
thesis_chapter1.odf

Thesis/Chapter2:
thesis_chapter2.odf

Thesis/Chapter3:
[user@host Documents]$ rm -r Thesis/Chapter1
[user@host Documents]$ ls -l Thesis
total 8
drwxrwxr-x. 2 user user 4096 Feb 11 12:47 Chapter2
drwxrwxr-x. 2 user user 4096 Feb 11 12:48 Chapter3
```

Команда **rm -r** сначала просматривает каждый подкаталог, индивидуально удаляя их файлы перед удалением каждого каталога. Вы можете использовать команду **rm -ri** для интерактивного запроса подтверждения перед удалением. По сути, это противоположно использованию опции **-f**, которая вызывает принудительное удаление без запроса подтверждения у пользователя.

```
[user@host Documents]$ rm -ri Thesis
rm: descend into directory `Thesis'? y
rm: descend into directory `Thesis/Chapter2'? y
rm: remove regular empty file `Thesis/Chapter2/thesis_chapter2.odf'? y
rm: remove directory `Thesis/Chapter2'? y
rm: remove directory `Thesis/Chapter3'? y
rm: remove directory `Thesis'? y
[user@host Documents]$
```



ПРЕДУПРЕЖДЕНИЕ

Если вы укажете обе опции **-i** и **-f**, опция **-f** будет иметь приоритет, и вам не будет предложено подтвердить, прежде чем **rm** удалит файлы.

В следующем примере команда **rmdir** удаляет только пустой каталог. Как и в предыдущем примере, вы должны использовать команду **rm -r** для удаления каталога, содержащего контент.

```
[user@host Documents]$ pwd
/home/student/Documents
[user@host Documents]$ rmdir ProjectY
[user@host Documents]$ rmdir ProjectX
rmdir: failed to remove `ProjectX': Directory not empty
[user@host Documents]$ rm -r ProjectX
[user@host Documents]$ ls -lR
.:
total 0
[user@host Documents]$
```



ПРИМЕЧАНИЕ

Команда **rm** без параметров не может удалить пустой каталог. Вы должны использовать команду **rmdir**, **rm -d** (что эквивалентно **rmdir**) или **rm -r**.



РЕКОМЕНДАЦИИ

Смотрите следующие справочные страницы **man cp(1)**, **mkdir(1)**, **mv(1)**, **rm(1)**, и **rmdir(1)**

УПРАЖНЕНИЯ С ПОШАГОВЫМИ ИНСТРУКЦИЯМИ

УПРАВЛЕНИЕ ФАЙЛАМИ С ПОМОЩЬЮ ИНСТРУМЕНТОВ КОМАНДНОЙ СТРОКИ

В этом упражнении вы будете создавать, систематизировать, копировать и удалять файлы и каталоги.

РЕЗУЛЬТАТЫ

Вы должны иметь возможность создавать, систематизировать, копировать и удалять файлы и каталоги.

ПРЕЖДЕ ЧЕМ ВЫ НАЧНЕТЕ

Войдите в систему как пользователь **student** на рабочей станции, используя в качестве пароля слово **student**.

На рабочей станции запустите команду **lab files-manage start**. Эта команда запускает сценарий запуска, который определяет, доступна ли серверная машина в сети.

```
[student@workstation ~]$ lab files-manage start
```

1. Используйте команду **ssh** для входа на сервер как пользователь **-student**. Системы настроены на использование ключей SSH для аутентификации, поэтому пароль не требуется.

```
[student@workstation ~]$ ssh student@servera  
[student@servera ~]$
```

2. В домашнем каталоге пользователя ученика используйте команду **mkdir**, чтобы создать три подкатаога: **Music**, **Pictures**, и **Videos**.

```
[student@servera ~]$ mkdir Music Pictures Videos
```

3. Продолжая работу в домашнем каталоге пользователя ученика, используйте сенсорную команду для создания наборов пустых файлов практики, которые будут использоваться во время этой лабораторной работы.
 - Создайте шесть файлов с именами в формате **songX.mp3**.
 - Создайте шесть файлов с именами вида **snapX.jpg**.
 - Создайте шесть файлов с именами вида **filmX.avi**.

В каждом наборе замените **Х** цифрами от 1 до 6.

```
[student@servera ~]$ touch song1.mp3 song2.mp3 song3.mp3 song4.mp3 \
> song5.mp3 song6.mp3
[student@servera ~]$ touch snap1.jpg snap2.jpg snap3.jpg snap4.jpg \
> snap5.jpg snap6.jpg
[student@servera ~]$ touch film1.avi film2.avi film3.avi film4.avi \
> film5.avi film6.avi
[student@servera ~]$ ls -l
```

```
total 0
-rw-rw-r--. 1 student student 0 Feb  4 18:23 film1.avi
-rw-rw-r--. 1 student student 0 Feb  4 18:23 film2.avi
-rw-rw-r--. 1 student student 0 Feb  4 18:23 film3.avi
-rw-rw-r--. 1 student student 0 Feb  4 18:23 film4.avi
-rw-rw-r--. 1 student student 0 Feb  4 18:23 film5.avi
-rw-rw-r--. 1 student student 0 Feb  4 18:23 film6.avi
drwxrwxr-x. 2 student student 6 Feb  4 18:23 Music
drwxrwxr-x. 2 student student 6 Feb  4 18:23 Pictures
-rw-rw-r--. 1 student student 0 Feb  4 18:23 snap1.jpg
-rw-rw-r--. 1 student student 0 Feb  4 18:23 snap2.jpg
-rw-rw-r--. 1 student student 0 Feb  4 18:23 snap3.jpg
-rw-rw-r--. 1 student student 0 Feb  4 18:23 snap4.jpg
-rw-rw-r--. 1 student student 0 Feb  4 18:23 snap5.jpg
-rw-rw-r--. 1 student student 0 Feb  4 18:23 snap6.jpg
-rw-rw-r--. 1 student student 0 Feb  4 18:23 song1.mp3
-rw-rw-r--. 1 student student 0 Feb  4 18:23 song2.mp3
-rw-rw-r--. 1 student student 0 Feb  4 18:23 song3.mp3
-rw-rw-r--. 1 student student 0 Feb  4 18:23 song4.mp3
-rw-rw-r--. 1 student student 0 Feb  4 18:23 song5.mp3
-rw-rw-r--. 1 student student 0 Feb  4 18:23 song6.mp3
drwxrwxr-x. 2 student student 6 Feb  4 18:23 Videos
```

4. Продолжая работу в домашнем каталоге пользователя **student**, переместите файлы песен в подкаталог «**Music**», файлы моментальных снимков - в подкаталог «**Pictures**», а файлы фильмов - в подкаталог «**Videos**».

При распространении файлов из одного места во многие места сначала перейдите в каталог, содержащий исходные файлы. Используйте простейший синтаксис пути, абсолютный или относительный, чтобы добраться до места назначения для каждой задачи управления файлами.

```
[student@servera ~]$ mv song1.mp3 song2.mp3 song3.mp3 song4.mp3 \
> song5.mp3 song6.mp3 Music
[student@servera ~]$ mv snap1.jpg snap2.jpg snap3.jpg snap4.jpg \
> snap5.jpg snap6.jpg Pictures
[student@servera ~]$ mv film1.avi film2.avi film3.avi film4.avi \
> film5.avi film6.avi Videos
[student@servera ~]$ ls -l Music Pictures Videos
Music:
total 0
-rw-rw-r--. 1 student student 0 Feb 4 18:23 song1.mp3
-rw-rw-r--. 1 student student 0 Feb 4 18:23 song2.mp3
-rw-rw-r--. 1 student student 0 Feb 4 18:23 song3.mp3
```

```
-rw-rw-r--. 1 student student 0 Feb 4 18:23 song4.mp3  
-rw-rw-r--. 1 student student 0 Feb 4 18:23 song5.mp3  
-rw-rw-r--. 1 student student 0 Feb 4 18:23 song6.mp3
```

Pictures:

total 0

```
-rw-rw-r--. 1 student student 0 Feb 4 18:23 snap1.jpg  
-rw-rw-r--. 1 student student 0 Feb 4 18:23 snap2.jpg  
-rw-rw-r--. 1 student student 0 Feb 4 18:23 snap3.jpg  
-rw-rw-r--. 1 student student 0 Feb 4 18:23 snap4.jpg  
-rw-rw-r--. 1 student student 0 Feb 4 18:23 snap5.jpg  
-rw-rw-r--. 1 student student 0 Feb 4 18:23 snap6.jpg
```

Videos:

total 0

```
-rw-rw-r--. 1 student student 0 Feb 4 18:23 film1.avi  
-rw-rw-r--. 1 student student 0 Feb 4 18:23 film2.avi  
-rw-rw-r--. 1 student student 0 Feb 4 18:23 film3.avi  
-rw-rw-r--. 1 student student 0 Feb 4 18:23 film4.avi  
-rw-rw-r--. 1 student student 0 Feb 4 18:23 film5.avi  
-rw-rw-r--. 1 student student 0 Feb 4 18:23 film6.avi
```

5. Продолжая работу в домашнем каталоге пользователя **student**, создайте три подкаталога для организации ваших файлов в проекты. Назовите подкаталоги **friends**, **family**, **iwork**. Используйте одну команду, чтобы создать все три подкаталога одновременно.

Вы будете использовать эти каталоги для преобразования файлов в проекты.

```
[student@servera ~]$ mkdir friends family work  
[student@servera ~]$ ls -l  
total 0  
drwxrwxr-x. 2 student student 6 Feb 4 18:38 family  
drwxrwxr-x. 2 student student 6 Feb 4 18:38 friends  
drwxrwxr-x. 2 student student 108 Feb 4 18:36 Music  
drwxrwxr-x. 2 student student 108 Feb 4 18:36 Pictures  
drwxrwxr-x. 2 student student 108 Feb 4 18:36 Videos  
drwxrwxr-x. 2 student student 6 Feb 4 18:38 work
```

6. Скопируйте выбранные новые файлы в каталоги проекта **family** и **friends**. Используйте столько команд, сколько необходимо. Вам не обязательно использовать только одну команду, как в примере. Для каждого проекта сначала перейдите в каталог проекта, затем скопируйте исходные файлы в этот каталог. Помните, что вы делаете копии, поэтому исходные файлы останутся на своих местах после того, как файлы будут скопированы в каталоги проекта.

- Скопируйте файлы (всех типов), содержащие числа 1 и 2, в подкаталог **friends**.
- Скопируйте файлы (всех типов), содержащие числа 3 и 4, в подкаталог **family**.

При копировании файлов из нескольких мест в одно место Red Hat рекомендует перед копированием файлов перейти в целевой каталог. Используйте простейший синтаксис пути, абсолютный или относительный, чтобы достичь источника для каждой задачи управления файлами.

```
[student@servera ~]$ cd friends
[student@servera friends]$ cp ~/Music/song1.mp3 ~/Music/song2.mp3 \
> ~/Pictures/snap1.jpg ~/Pictures/snap2.jpg ~/Videos/film1.avi \
> ~/Videos/film2.avi .
[student@servera friends]$ ls -l
total 0
-rw-rw-r--. 1 student student 0 Feb 4 18:42 film1.avi
-rw-rw-r--. 1 student student 0 Feb 4 18:42 film2.avi
-rw-rw-r--. 1 student student 0 Feb 4 18:42 snap1.jpg
-rw-rw-r--. 1 student student 0 Feb 4 18:42 snap2.jpg
-rw-rw-r--. 1 student student 0 Feb 4 18:42 song1.mp3
-rw-rw-r--. 1 student student 0 Feb 4 18:42 song2.mp3
[student@servera friends]$ cd ../family
[student@servera family]$ cp ~/Music/song3.mp3 ~/Music/song4.mp3 \
> ~/Pictures/snap3.jpg ~/Pictures/snap4.jpg ~/Videos/film3.avi \
> ~/Videos/film4.avi .
[student@servera family]$ ls -l
total 0
-rw-rw-r--. 1 student student 0 Feb 4 18:44 film3.avi
-rw-rw-r--. 1 student student 0 Feb 4 18:44 film4.avi
-rw-rw-r--. 1 student student 0 Feb 4 18:44 snap3.jpg
-rw-rw-r--. 1 student student 0 Feb 4 18:44 snap4.jpg
-rw-rw-r--. 1 student student 0 Feb 4 18:44 song3.mp3
-rw-rw-r--. 1 student student 0 Feb 4 18:44 song4.mp3
```

7. Для рабочего проекта создайте дополнительные копии.

```
[student@servera family]$ cd ../work
[student@servera work]$ cp ~/Music/song5.mp3 ~/Music/song6.mp3 \
> ~/Pictures/snap5.jpg ~/Pictures/snap6.jpg \
> ~/Videos/film5.avi ~/Videos/film6.avi .
[student@servera work]$ ls -l
total 0
-rw-rw-r--. 1 student student 0 Feb 4 18:48 film5.avi
-rw-rw-r--. 1 student student 0 Feb 4 18:48 film6.avi
-rw-rw-r--. 1 student student 0 Feb 4 18:48 snap5.jpg
-rw-rw-r--. 1 student student 0 Feb 4 18:48 snap6.jpg
-rw-rw-r--. 1 student student 0 Feb 4 18:48 song5.mp3
-rw-rw-r--. 1 student student 0 Feb 4 18:48 song6.mp3
```

8. Задачи вашего проекта завершены, и пора навести порядок в проектах. Перейдите в домашний каталог пользователя **student**. Попытайтесь удалить каталоги проекта **family** и **friends** с помощью одной команды **rmdir**.

```
[student@servera work]$ cd  
[student@servera ~]$ rmdir family friends  
rmdir: failed to remove 'family': Directory not empty  
rmdir: failed to remove 'friends': Directory not empty
```

Использование команды **rmdir** должно завершиться ошибкой, потому что оба подкаталога содержат файлы.

9. Используйте команду **rm -r**, чтобы рекурсивно удалить подкаталоги **family** и **friends**, а также их содержимое.

```
[student@servera ~]$ rm -r family friends  
[student@servera ~]$ ls -l  
total 0  
drwxrwxr-x. 2 student student 108 Feb 4 18:36 Music  
drwxrwxr-x. 2 student student 108 Feb 4 18:36 Pictures  
drwxrwxr-x. 2 student student 108 Feb 4 18:36 Videos  
drwxrwxr-x. 2 student student 108 Feb 4 18:48 work
```

10. Удалите все файлы в рабочем проекте, но не удаляйте рабочий каталог.

```
[student@servera ~]$ cd work  
[student@servera work]$ rm song5.mp3 song6.mp3 snap5.jpg snap6.jpg \  
> film5.avi film6.avi  
[student@servera work]$ ls -l  
total 0
```

11. Наконец, из домашнего каталога пользователя **student** удалите каталог **work** с помощью команды **rmdir**. Теперь, когда она пуста, команда должна завершиться успешно.

```
[student@servera work]$ cd  
[student@servera ~]$ rmdir work  
[student@servera ~]$ ls -l  
total 0  
drwxrwxr-x. 2 student student 108 Feb 4 18:36 Music  
drwxrwxr-x. 2 student student 108 Feb 4 18:36 Pictures  
drwxrwxr-x. 2 student student 108 Feb 4 18:36 Videos
```

12. Выход из сервера.

```
[student@servera ~]$ exit  
logout  
Connection to servera closed.
```

```
[student@workstation ~]$
```

ОКОНЧАНИЕ

На рабочей станции запустите сценарий **lab files-manage finish**, чтобы завершить это упражнение. Сценарий удалит все каталоги и файлы, созданные во время этого упражнения.

```
[student@workstation ~]$ lab files-manage finish
```

На этом пошаговое упражнение завершено.

СОЗДАНИЕ ССЫЛКИ МЕЖДУ ФАЙЛАМИ

ЦЕЛИ

После завершения этого раздела вы сможете сделать так, чтобы несколько имен файлов ссылались на один и тот же файл, используя жесткие ссылки и символические (или «soft») ссылки.

УПРАВЛЕНИЕ ССЫЛКАМИ МЕЖДУ ФАЙЛАМИ

Жесткие ссылки и символические ссылки

Можно создать несколько имен, указывающих на один и тот же файл. Это можно сделать двумя способами: создать **жесткую** ссылку на файл или создать **символической(soft)** ссылку (иногда называемую символической ссылкой) на файл. У каждого есть свои преимущества и недостатки.

Создание жестких ссылок

Каждый файл начинается с одной жесткой ссылки, от его начального имени до данных в файловой системе. Когда вы создаете новую жесткую ссылку на файл, вы создаете другое имя, указывающее на те же данные. Новая жесткая ссылка действует точно так же, как и исходное имя файла. После создания вы не сможете отличить новую жесткую ссылку от исходного имени файла.

Вы можете узнать, есть ли у файла несколько жестких ссылок, с помощью команды **ls -l**. Одним из пунктов, которые она сообщает, - это *количество ссылок* на каждый файл, количество жестких ссылок, которые имеет файл.

```
[user@host ~]$ pwd  
/home/user  
[user@host ~]$ ls -l newfile.txt  
-rw-r--r--. 1 user user 0 Mar 11 19:19 newfile.txt
```

В предыдущем примере счетчик ссылок в **newfile.txt** равен **1**. Он имеет ровно один абсолютный путь, то есть **/home/user/newfile.txt**.

Вы можете использовать команду **ln**, чтобы создать новую жесткую ссылку (другое имя), указывающую на существующий файл. Команде требуется как минимум два аргумента: **путь к существующему файлу** и **путь к жесткой ссылке**, которую вы хотите создать.

В следующем примере создается жесткая ссылка с именем **newfile-link2.txt** для существующего файла **newfile.txt** в каталоге **/tmp**.

```
[user@host ~]$ ln newfile.txt /tmp/newfile-hlink2.txt  
[user@host ~]$ ls -l newfile.txt /tmp/newfile-hlink2.txt  
-rw-rw-r--. 2 user user 12 Mar 11 19:19 newfile.txt  
-rw-rw-r--. 2 user user 12 Mar 11 19:19 /tmp/newfile-hlink2.txt
```

Если вы хотите узнать, являются ли два файла жесткими связями друг с другом, один из способов узнать это - использовать параметр **-i** с командой **ls** для вывода списка номеров **inode** файлов. Если файлы находятся в одной файловой системе (обсуждается ниже) и их номера **inode** совпадают, файлы представляют собой жесткие ссылки, указывающие на одни и те же данные (файл).

```
[user@host ~]$ ls -il newfile.txt /tmp/newfile-hlink2.txt  
8924107 -rw-rw-r--. 2 user user 12 Mar 11 19:19 newfile.txt  
8924107 -rw-rw-r--. 2 user user 12 Mar 11 19:19 /tmp/newfile-hlink2.txt
```



ВАЖНО

Все жесткие ссылки, которые ссылаются на один и тот же файл, будут иметь одинаковое количество ссылок, права доступа, права собственности пользователей и групп, отметки времени и содержимое файла. Если какая-либо из этой информации была изменена с помощью одной жесткой ссылки, все другие жесткие ссылки, указывающие на тот же файл, также будут отображать новую информацию. Это связано с тем, что каждая жесткая ссылка указывает на одни и те же данные на устройстве хранения.

Даже если исходный файл будет удален, его содержимое останется доступным, если существует хотя бы одна жесткая ссылка. Данные удаляются из хранилища только при удалении последней жесткой ссылки.

```
[user@host ~]$ rm -f newfile.txt  
[user@host ~]$ ls -l /tmp/newfile-hlink2.txt  
-rw-rw-r--. 1 user user 12 Mar 11 19:19 /tmp/newfile-hlink2.txt  
[user@host ~]$ cat /tmp/newfile-hlink2.txt  
Hello World
```

Ограничения жестких ссылок

Жесткие ссылки имеют некоторые ограничения. Во-первых, жесткие ссылки можно использовать только с обычными файлами. Вы не можете использовать **ln** для создания жесткой ссылки на каталог или специальный файл.

Во-вторых, жесткие ссылки можно использовать только в том случае, если оба файла находятся в одной файловой системе. Иерархия файловой системы может состоять из нескольких устройств хранения. В зависимости от конфигурации вашей системы, когда вы переходите в новый каталог, этот каталог и его содержимое могут храниться в другой файловой системе.

Вы можете использовать команду **df** для вывода списка каталогов, находящихся в разных файловых системах. Например, вы можете увидеть следующий результат:

```
[user@host ~]$ df
Filesystem      1K-blocks   Used Available Use% Mounted on
/devtmpfs          886788     0   886788  0% /dev
tmpfs             902108     0   902108  0% /dev/shm
tmpfs             902108   8696   893412  1% /run
tmpfs             902108     0   902108  0% /sys/fs/cgroup
/dev/mapper/rhel_rhel8--root 10258432 1630460  8627972 16% /
/dev/sda1         1038336 167128   871208 17% /boot
tmpfs            180420     0   180420  0% /run/user/1000
[user@host ~]$
```

Файлы в двух разных каталогах «Смонтированы (**Mounted on**)» и их подкаталоги находятся в разных файловых системах. (Побеждает наиболее конкретное совпадение.) Таким образом, в системе в этом примере вы можете создать жесткую связь между `/var/tmp/link1` и `/home/user/file`, потому что они оба являются подкаталогами `/`, но не любым другим каталогом из списка. Но вы не можете создать жесткую связь между файлами `/boot/test/badlink` и `/home/user/file`, потому что первый файл находится в подкаталоге `/boot` (в колонке Смонтированы (**Mounted on**)), а второй файл - нет.

Создание символических ссылок

Команда `ln -s` создает символическую ссылку. Символическая ссылка - это не обычный файл, а файл особого типа, указывающий на существующий файл или каталог.

Символические ссылки имеют некоторые преимущества перед жесткими ссылками:

- Они могут связывать два файла в разных файловых системах.
- Они могут указывать на каталог или специальный файл, а не только на обычный файл.

В следующем примере команда `ln -s` используется для создания новой символической ссылки для существующего файла `/home/user/newfile-link2.txt` с именем `/tmp/newfile-symlink.txt`.

```
[user@host ~]$ ln -s /home/user/newfile-link2.txt /tmp/newfile-symlink.txt
[user@host ~]$ ls -l newfile-link2.txt /tmp/newfile-symlink.txt
-rw-rw-r--. 1 user user 12 Mar 11 19:19 newfile-link2.txt
lrwxrwxrwx. 1 user user 11 Mar 11 20:59 /tmp/newfile-symlink.txt -> /home/user/
newfile-link2.txt
[user@host ~]$ cat /tmp/newfile-symlink.txt
Soft Hello World
```

В предыдущем примере первым символом длинного списка для `/tmp/newfile-symlink.txt` является `l` вместо `-`. Это означает, что файл является программной ссылкой, а не обычным файлом. (`d` означает, что файл является каталогом.)

Когда исходный обычный файл будет удален, символическая ссылка по-прежнему будет указывать на файл, но цель исчезнет. Символическая ссылка, указывающая на отсутствующий файл, называется «зависшей символьской ссылкой (dangling soft link)».

```
[user@host ~]$ rm -f newfile-link2.txt
[user@host ~]$ ls -l /tmp/newfile-symlink.txt
lrwxrwxrwx. 1 user user 11 Mar 11 20:59 /tmp/newfile-symlink.txt -> /home/user/
newfile-link2.txt
[user@host ~]$ cat /tmp/newfile-symlink.txt
cat: /tmp/newfile-symlink.txt: No such file or directory
```



ВАЖНО

Одним из побочных эффектов зависшей символьической ссылки в предыдущем примере является то, что если позже вы создадите новый файл с тем же именем, что и удаленный файл (**/home/user/newfilelink2.txt**), программная ссылка больше не будет «зависшей» и укажет на новый файл.

Жесткие ссылки так не работают. Если вы удалите жесткую ссылку, а затем используете обычные инструменты (а не **ln**) для создания нового файла с тем же именем, новый файл не будет связан со старым файлом.

Один из способов сравнить жесткие ссылки и программные ссылки, который может помочь вам понять, как они работают:

- Жесткая ссылка указывает имя на данные на устройстве хранения.
- Символическая ссылка указывает имя на другое имя, которое указывает на данные на устройстве хранения.

Символическая ссылка может указывать на каталог. В этом случае символическая ссылка действует как каталог. Переход по символическим ссылкам с помощью команды **cd** сделает текущий рабочий каталог связанным каталогом. Некоторые инструменты могут отслеживать тот факт, что вы перешли по символической ссылке, чтобы попасть туда. Например, по умолчанию **cd** обновит ваш текущий рабочий каталог, используя имя символьской ссылки, а не имя фактического каталога (есть опция **-P**, которая вместо этого обновит его именем фактического каталога.)

В следующем примере создается программная ссылка с именем **/home/user/configfiles**, указывающая на каталог **/etc**.

```
[user@host ~]$ ln -s /etc /home/user/configfiles
[user@host ~]$ cd /home/user/configfiles
[user@host configfiles]$ pwd
/home/user/configfiles
```



РЕКОМЕНДАЦИИ

Справочные страницы **man ln(1)**
infoln ('**ln**': создавать ссылки между файлами)

УПРАЖНЕНИЯ С ПОШАГОВЫМИ ИНСТРУКЦИЯМИ

СОЗДАНИЕ СВЯЗЕЙ (links) МЕЖДУ ФАЙЛАМИ

В этом упражнении вы создадите жесткие и символические ссылки и сравните результаты.

РЕЗУЛЬТАТЫ

Вы будете способны создавать жесткие и символические ссылки между файлами.

ПРЕЖДЕ ЧЕМ ВЫ НАЧНЕТЕ

Войдите на рабочую станцию, как пользователь **student**, используя в качестве пароля слово **student**.

На рабочей станции запустите команду **lab files-make start**. Выполниться сценарий запуска, который определяет, доступен ли хост сервера в сети, и создает файлы и рабочие каталоги на сервере.

```
[student@workstation ~]$ lab files-make start
```

1. Используйте команду **ssh** для входа на сервер как пользователь **student**. Системы настроены на использование ключей **SSH** для аутентификации, поэтому пароль не требуется.

```
[student@workstation ~]$ ssh student@servera  
[student@servera ~]$
```

2. Создайте жесткую ссылку с именем **/home/student/backups/source.backup** для существующего файла **/home/student/files/source.file**
.
- 2.1. Просмотрите количество ссылок для файла **/home/student/files/source.file**.

```
[student@servera ~]$ ls -l files/source.file  
total 4  
-rw-r--r--. 1 student student 11 Mar 5 21:19 source.file
```

- 2.2. Создайте жесткую ссылку с именем **/home/student/backups/source.backup**.
Свяжите его с файлом **/home/student/files/source.file**.

```
[student@servera ~]$ ln /home/student/files/source.file \  
> /home/student/backups/source.backup
```

- 2.3. Проверьте количество ссылок для исходного файла **/home/student/files/source.file** и нового файла **/home/student/backups/source.backup**. Количество ссылок должно быть **2** для обоих файлов.

```
[student@servera ~]$ ls -l /home/student/files/  
-rw-r--r--. 2 student student 11 Mar 5 21:19 source.file  
[student@servera ~]$ ls -l /home/student/backups/  
-rw-r--r--. 2 student student 11 Mar 5 21:19 source.backup
```

3. Создайте символьическую ссылку с именем **/home/student/tempdir**, которая указывает на каталог **/tmp** на сервере.

- 3.1. Создайте символьическую ссылку с именем **/home/student/tempdir** и свяжите ее с **/tmp**.

```
[student@servera ~]$ ln -s /tmp /home/student/tempdir
```

- 3.2. Используйте команду **ls -l**, чтобы проверить вновь созданную программную ссылку.

```
[student@servera ~]$ ls -l /home/student/tempdir  
lrwxrwxrwx. 1 student student 4 Mar 5 22:04 /home/student/tempdir -> /tmp
```

4. Выход из сервера.

```
[student@servera ~]$ exit  
logout  
Connection to servera closed.  
[student@workstation ~]$
```

ОКОНЧАНИЕ

На рабочей станции запустите сценарий **lab files-make finish**, чтобы завершить это упражнение. Этот скрипт удаляет все файлы и каталоги, созданные на сервере во время упражнения.

```
[student@workstation ~]$ lab files-make finish
```

На этом пошаговое упражнение завершено.

СОПОСТАВЛЕНИЕ ИМЕН ФАЙЛОВ В ОБОЛОЧКИ (SHELL)

ЗАДАЧИ

После завершения этого раздела вы сможете эффективно запускать команды, влияющие на многие файлы, с помощью функций сопоставления шаблонов оболочки **Bash**.

РАСШИРЕНИЯ КОМАНДНОЙ СТРОКИ



ПРИМЕЧАНИЕ ПРИ ПЕРЕВОДЕ

Впервые столкнулся при переводе с понятием **РАСШИРЕНИЕ** всегда считал, что речь идёт о **подстановки символов**. Нашёл интересный комментарий на сайте <https://zalinux.ru/?p=1293#1> Текст привожу ниже, и далее буду переводить, как есть в оригинальном.

Термин оболочки «expansion» – раскрытие или подстановка?

В русскоязычной литературе «expansion» из документации **Bash** часто переводят как подстановка. Внешне этот процесс выглядит именно так – вместо одних символов в командной строке появляются другие символы или новые слова. Но если заглянуть глубже, то имеет место не механическая подстановка («найти символ звёздочка и заменить её на набор слов bin Видео Документы Загрузки Изображения Музыка Общедоступные Рабочий стол Шаблоны»), а именно расширение, раскрытие значения. Значение, которое может иметь * в командной строке, зависит от того, в какой директории пользователь запустил эту команду и что содержится в этой директории.

Тем не менее, использование слов «раскрытие» или «подстановка», в первую очередь, обусловлено привычкой, поскольку, как уже отмечено выше, визуально, интуитивно, происходит именно подстановка…

Оболочка **Bash** имеет несколько способов расширения командной строки, включая сопоставление с образцом, расширение домашнего каталога, расширение строки и подстановку переменных. Возможно, наиболее мощным из них является возможность сопоставления имен путей, исторически называемая подстановкой. Функция подстановки **Bash**, иногда называемая «подстановочными знаками», упрощает управление большим количеством файлов. Используя метасимволы, которые «расширяются», чтобы соответствовать искомым именам файлов и путей, команды выполняются одновременно с определенным набором файлов.

Соответствие шаблону

Универсализация (Globbing) - это операция синтаксического анализа команд оболочки, которая расширяет возможности работы в командной строке при помощи шаблона подстановки в списке совпадающих имен путей. Перед выполнением команды метасимволы командной строки заменяются списком совпадений. Шаблоны, которые не возвращают совпадений, отображают исходный запрос шаблона как буквальный текст. Ниже приведены общие метасимволы и классы шаблонов.

Таблица метасимволов и соответствующие им совпадения

Подстановочные символы и кассы символов	Значение
*	Любая строка из нуля или более символов.
?	Любой одиночный символ.
[abc ...]	Любой символ в закрытом классе (в квадратных скобках).
[! abc ...]	Любой один символ, не входящий в закрытый класс.
[^ abc ...]	Любой один символ, не входящий в закрытый класс.
[:alpha:]	Любой буквенный символ.
[:lower:]	Любой символ нижнего регистра.
[:upper:]	Любой символ верхнего регистра
[:alnum:]	Любой буквенный символ или цифра.
[:punct:]	Любой печатный символ, кроме пробела или буквенно-цифрового символа.
[:digit:]	Любая отдельная цифра от 0 до 9.
[:space:]	Любой одиночный символ пробела. Это может быть табуляция, перевод строки, возврат каретки, поля формы или пробелы.

Для следующих нескольких примеров представьте, что вы выполнили следующие команды, чтобы создать несколько файлов для примеров.

```
[user@host ~]$ mkdir glob; cd glob
[user@host glob]$ touch alfa bravo charlie delta echo able baker cast dog easy
[user@host glob]$ ls
able alfa baker bravo cast charlie delta dog easy echo
[user@host glob]$
```

В первом примере будет использоваться простое сопоставление шаблонов с символами звездочки (*) и вопросительного знака (?), а также класс символов для сопоставления некоторых из этих имен файлов.

```
[user@host glob]$ ls a*
able alfa
[user@host glob]$ ls *a*
able alfa baker bravo cast charlie delta easy
[user@host glob]$ ls [ac]*
able alfa cast charlie
```

```
[user@host glob]$ ls ****  
able alfa cast easy echo  
[user@host glob]$ ls *****  
baker bravo delta  
[user@hostglob]$
```

Расширение тильда (~)

Символ тильды (~) соответствует домашнему каталогу текущего пользователя. Если он начинает строку символов, отличных от косой черты (/), оболочка интерпретирует строку до этой косой черты как имя пользователя, если оно совпадает, и заменяет строку абсолютным путем к домашнему каталогу этого пользователя. Если имя пользователя не совпадает, вместо него будет использоваться фактическая тильда, за которой следует строка символов.

В следующем примере команда **echo** используется для отображения значения символа тильды. Команду **echo** также можно использовать для отображения значений скобок, символов раскрытия переменных и других.

```
[user@host glob]$ ls ~root  
/root  
[user@host glob]$ ls ~user  
/home/user  
[user@host glob]$ ls ~/glob  
able alfa baker bravo cast charlie delta dog easy echo  
[user@host glob]$ echo ~/glob  
/home/user/glob  
[user@host glob]$
```

Расширение скобки

Раскрытие скобки используется для создания произвольных строк символов. Фигурные скобки содержат список строк, разделенных запятыми, или выражение последовательности. Результат включает текст, предшествующий или следующий за определением скобки. Раскладки скобок могут быть вложены одно в другое. Также синтаксис с двумя точками (..) расширяется до последовательности, так что {m..p} расширяется до m n o p.

```
[user@host glob]$ echo {Sunday,Monday,Tuesday,Wednesday}.log  
Sunday.log Monday.log Tuesday.log Wednesday.log  
[user@host glob]$ echo file{1..3}.txt  
file1.txt file2.txt file3.txt  
[user@host glob]$ echo file{a..c}.txt  
filea.txt fileb.txt filec.txt  
[user@host glob]$ echo file{a,b}{1,2}.txt  
filea1.txt filea2.txt fileb1.txt fileb2.txt  
[user@host glob]$ echo file{a{1,2},b,c}.txt
```

```
filea1.txt filea2.txt fileb.txt filec.txt  
[user@host glob]$
```

Практическое использование расширения скобок позволяет быстро создать ряд файлов или каталогов.

```
[user@host glob]$ mkdir ../RHEL{6,7,8}  
[user@host glob]$ ls ../RHEL*  
RHEL6 RHEL7 RHEL8  
[user@hostglob]$
```

Расширение переменные

Переменная действует как именованный контейнер, который может хранить значение в памяти. Переменные упрощают доступ к сохраненным данным и их изменение из командной строки или в сценарии оболочки.

Вы можете присвоить данные как значение переменной, используя следующий синтаксис:

```
[user@host ~]$ VARIABLENAME=value
```

Вы можете использовать расширение переменная, чтобы преобразовать имя переменной в ее значение в командной строке. Если строка начинается со знака доллара (\$), оболочка попытается использовать оставшуюся часть этой строки в качестве имени переменной и заменить ее значением переменной.

```
[user@host ~]$ USERNAME=operator  
[user@host ~]$ echo $USERNAME  
operator
```

Чтобы избежать ошибок из-за других расширений оболочки, вы можете заключить имя переменной в фигурные скобки, например \${VARIABLENAME}.

```
[user@host ~]$ USERNAME=operator  
[user@host ~]$ echo ${USERNAME}  
operator
```

Переменные оболочки и способы их использования будут рассмотрены более подробно позже в этом курсе.

Подстановка команд

Подстановка команд позволяет выводом команды заменить саму команду в командной строке. Подстановка команд происходит, когда команда заключена в круглые скобки и ей предшествует знак доллара (\$). Форма **\$(command)** может содержать несколько расширений команд друг в друге.

```
[user@host glob]$ echo Today is $(date +%A).
Today is Wednesday.
[user@host glob]$ echo The time is $(date +%M) minutes past $(date +%l%p).
The time is 26 minutes past 11AM.
[user@host glob]$
```



ПРИМЕЧАНИЕ

В более старых формах подстановки команд используются обратные кавычки: `command`
К недостаткам формы обратных кавычек можно отнести: 1) обратные кавычки легко визуально спутать с одинарными кавычками и 2) обратные кавычки не могут быть вложенными.

Захист аргументов от расширения (выполнение подмены)

Многие символы имеют особое значение в оболочке **Bash**. Чтобы оболочка не выполняла расширения для частей командной строки, вы можете заключать в **кавычки и экранировать** символы и строки.

Обратная косая черта (\) - это символ экранирования в оболочке **Bash**. Он защитит символы, следующего сразу за ним, от выполнения расширения.

```
[user@host glob]$ echo The value of $HOME is your home directory.
The value of /home/user is your home directory.
[user@host glob]$ echo The value of \$HOME is your home directory.
The value of $HOME is your home directory.
[user@host glob]$
```

В предыдущем примере защита знака **доллара** от раскрытия заставила **Bash** рассматривать его как обычный символ и не выполняла раскрытие переменной для **\$HOME**.

Чтобы защитить более длинные символьные строки, для заключения строк используются одинарные кавычки ('') или двойные кавычки (""). Они имеют несколько иной эффект. Одиночные кавычки останавливают все раскрытие оболочки. Двойные кавычки останавливают большинство расширений оболочки.

Используйте двойные кавычки, чтобы запретить подстановку символов и раскрытие оболочки, но все же разрешите подстановку команд и переменных.

```
[user@host glob]$ myhost=$(hostname -s); echo $myhost
host
[user@host glob]$ echo "***** hostname is ${myhost} *****"
***** hostname is host *****
[user@host glob]$
```

Используйте одинарные кавычки для буквального толкования всего текста.

```
[user@host glob]$ echo "Will variable $myhost evaluate to $(hostname -s)?"
Will variable myhost evaluate to host?
[user@host glob]$ echo 'Will variable $myhost evaluate to $(hostname -s)?'
Will variable $myhost evaluate to $(hostname -s)?
[user@host glob]$
```



ВАЖНО

Одиночные кавычки (`) и обратную кавычку подстановки команд (`) легко перепутать как на экране, так и на клавиатуре. Использование одного, когда вы собираетесь использовать другой, приведет к неожиданному поведению оболочки.



РЕКОМЕНДАЦИИ

Справочные страницы **namdash(1)**, **cd(1)**, **glob(7)**, **isalpha(3)**, **ls(1)**, **path_resolution(7)**, **ipwd(1)**

КОНТРОЛЬНЫЙ ОПРОС

СОГЛАСОВАНИЕ ИМЕНИ ФАЙЛОВ С РАСШИРЕНИЯМИ ОБОЛОЧКИ

Выберите правильные ответы на следующие вопросы:

1. Какой шаблон будет соответствовать только именам файлов, заканчивающимся на "b"?

- a. b*
- b. *b
- c. *b*
- d. [!b]*

2. Какой шаблон будет соответствовать только именам файлов, начинающимся с "b"?

- a. b*
- b. *b
- c. *b*
- d. [!b]*

3. Какой шаблон будет соответствовать только именам файлов, в которых первый символ не "b"?

- a. b*
- b. *b
- c. *b*
- d. [!b]*

4. Какой шаблон будет соответствовать всем именам файлов, содержащим букву "b"?

- a. b*
- b. *b
- c. *b*
- d. [!b]*

5. Какой шаблон будет соответствовать только именам файлов, которые содержат число?

- a. *#*
- b. *[[digit:]]*
- c. *[digit]*
- d. [0-9]

6. Какой шаблон будет соответствовать только именам файлов, начинающимся с заглавной буквы?

- a. ^?*
- b. ^*
- c. [upper]*

- d. [:upper:]^{*}
- e. [[CAP]]^{*}

7. Какой шаблон будет соответствовать только именам файлов длиной не менее трех символов?

- a. ???*
- b. ???
- c. \3*
- d. +++*
- e. ...*

РЕШЕНИЕ

СОГЛАСОВАНИЕ ИМЕНИ ФАЙЛОВ С РАСШИРЕНИЯМИ ОБОЛОЧКИ

Выберите правильные ответы на следующие вопросы:

1. Какой шаблон будет соответствовать только именам файлов, заканчивающимся на "b"?

- a. b*
- b. *b
- c. *b*
- d. [!b]*

2. Какой шаблон будет соответствовать только именам файлов, начинающимся с "b"?

- a. b*
- b. *b
- c. *b*
- d. [!b]*

3. Какой шаблон будет соответствовать только именам файлов, в которых первый символ не "b"?

- a. b*
- b. *b
- c. *b*
- d. [!b]*

4. Какой шаблон будет соответствовать всем именам файлов, содержащим букву "b"?

- a. b*
- b. *b
- c. *b*
- d. [!b]*

5. Какой шаблон будет соответствовать только именам файлов, которые содержат число?

- a. *#*
- b. *[[digit:]]*
- c. *[digit]*
- d. [0-9]

6. Какой шаблон будет соответствовать только именам файлов, начинающимся с заглавной буквы?

- a. ^?*
- b. ^*
- c. [upper]*

- d. [:upper:]^{*}
- e. [[CAP]]^{*}

7. Какой шаблон будет соответствовать только именам файлов длиной не менее трех символов?

- a. ???*
- b. ???
- c. \3*
- d. +++*
- e. ...*

ЛАБОРАТОРНАЯ РАБОТА

УПРАВЛЕНИЕ ФАЙЛАМИ ИЗ КОМАНДНОЙ СТРОКИ

КОНТРОЛЬНЫЙ СПИСОК РАБОТЫ

В этой лабораторной работе вы будете эффективно создавать, перемещать и удалять файлы и каталоги с помощью команд оболочки `bash` и различных методов сопоставления имен файлов.

РЕЗУЛЬТАТЫ

Вы должен быть способны:

- Используйте подстановочные символы для поиска файлов и управления ими.

ПРЕЖДЕ ЧЕМ ВЫ НАЧНЕТЕ

Войдите на рабочую станцию как пользователь `student`, используя в качестве пароля слово `student`.

На рабочей станции выполните команду `lab files-review start`. Команда запускает сценарий запуска, который определяет, доступен ли сервер `serverb` в сети.

```
[student@workstation ~]$ lab files-review start
```

1. Используйте команду `ssh` для входа на `serverb` как пользователь `student`. Системы настроены на использование ключей SSH для аутентификации, поэтому пароль не требуется.

```
[student@workstation ~]$ ssh student@serverb  
[student@serverb ~]$
```

2. Перед созданием файлов проекта используйте команду `mkdir` с расширением фигурных скобок, чтобы создать пустые документы планирования проекта в каталоге `/home/student/Documents/project_plans`. (Подсказка: если каталога `~/Documents` не существует, параметр `-p` для команды `mkdir` создаст его.)
Создайте два пустых файла в каталоге `~/Documents/project_plans`:
`season1_project_plan.odf`
`season2_project_plan.odf`.
3. Создайте наборы пустых файлов практики для использования в этой лабораторной работе. Если вы не сразу поймёте, как создать синтаксическое выражение для подстановки в оболочки, используйте **решение** для обучения и практики. Используйте клавиши табуляции, чтобы легко завершать набор имени пути к файлам.
Создайте в общей сложности 12 файлов с именами `tv_seasonX_episodeY.ogg`. Замените **X** номером сезона, а **Y** - эпизодом этого сезона, чтобы получить два сезона по шесть серий в каждом.

4. Как автор успешной серии детективных романов, главы вашего следующего бестселлера редактируются для публикации. Создайте в общей сложности восемь файлов с именами **mystery_chapterX.odf**. Замените X цифрами от **1** до **8**.
5. С помощью одной команды создайте два подкаталога с именами **Season1** и **Season2** в каталоге **Videos**, чтобы организовать сериалы.
6. Переместите соответствующие сериалы в подкаталоги сезонов. Используйте только две команды, указав места назначения с использованием относительного синтаксиса.
7. Создайте двухуровневую иерархию каталогов с помощью одной команды, чтобы организовать главы загадочной книги. Создайте **my_bestseller** в каталоге **Documents** и **chapters** новом каталоге **my_bestseller**.
8. Создайте еще три подкаталога непосредственно в каталоге **my_bestseller** с помощью одной команды. Назовите эти подкаталоги **editor**, **changes** и **vacation**. Параметр **-p** (создать родителей (**parents**)) не требуется, поскольку родительский каталог **my_bestseller** уже существует.
9. Перейдите в каталог **chapters**. Используя синтаксическое расширение для обозначения домашнего каталога тильды (~) для указания исходных файлов, переместите все главы книги в каталог **chapters**, который теперь является вашим текущим каталогом. Каков самый простой синтаксис для указания каталога назначения?
10. Вы отправили первые две главы на проверку редактору. Переместите только эти две главы в каталог **editor**, чтобы не изменять их во время рецензирования. Начиная с подкаталога **chapters**, используйте расширение скобок с диапазоном, чтобы указать имена файлов глав, которые нужно переместить, и относительный путь для каталога назначения.
11. Находясь в отпуске, вы собираетесь написать главы 7 и 8. Используйте одну команду, чтобы переместить файлы из каталога **chapters** в каталог **vacation**. Укажите имена файлов глав с помощью расширения скобок со списком строк и без использования подстановочных знаков
12. Измените рабочий каталог на **~/Videos/season2**, а затем скопируйте первую серию сезона в каталог **vacation**.
13. Используйте одну команду **cd**, чтобы перейти из рабочего каталога в каталог **~/Documents/my_bestseller/vacation**. Перечислите его файлы. Используйте предыдущий аргумент рабочего каталога, чтобы вернуться в каталог **season2**. (Действие будет успешным, если последнее изменение каталога с помощью команды **cd** было выполнено с помощью одной команды, а не нескольких команд **cd**.) Из каталога **season2** скопируйте файл **episode 2** в каталог **vacation**. Снова используйте синтаксическое выражение для подстановки, чтобы вернуться в каталог **vacation**.
14. Авторы глав 5 и 6 хотят поэкспериментировать с возможными изменениями. Скопируйте оба файла из каталога **~/Documents/my_bestseller/chapters** в каталог **~/Documents/my_bestseller/changes**, чтобы предотвратить изменение исходных файлов при редактировании глав. Перейдите в каталог **~/Documents/my_bestseller**. Используйте сопоставление с образцом в квадратных скобках, чтобы указать, какие номера глав должны совпадать в аргументе имени файла команды **cp**.
15. Измените текущий каталог на каталог **changes**.
Используйте команду **date +%F** с заменой, чтобы скопировать **mystery_chapter5.odf** в новый файл, который включает полную дату. Имя должно иметь вид **mystery_chapter5_YYYY-MM-DD.odf**.
Сделайте еще одну копию **mystery_chapter5.odf**, добавив текущую отметку времени (как количество секунд с начала **1970-01-01 00:00 UTC**), чтобы гарантировать уникальное имя файла. Для этого используйте подстановку команд с командой **date +%s**.
16. После дополнительного просмотра вы решаете, что в изменении сюжета нет необходимости. Удалите каталог **changes**.

При необходимости перейдите в каталог **changes** и удалите все файлы в этом каталоге. Вы не можете удалить каталог, пока он является текущим рабочим каталогом. Перейдите в родительский каталог каталога **changes**. Попробуйте удалить пустой каталог с помощью команды **rm** без рекурсивной опции **-r**. Эта попытка должна потерпеть неудачу. Наконец, используйте команду **rmdir** для удаления пустого каталога, что будет успешным.

17. По окончании отпуска каталог **vacation** больше не нужен. Удалите его с помощью команды **rm** с опцией рекурсии. По завершении вернитесь в домашний каталог пользователя **student**.
18. Создайте жесткую ссылку на файл **~/Documents/project_plans/season2_project_plan.odf** с именем **~/Documents/backups/season2_project_plan.odf.back**. Жесткая ссылка защитит от случайного удаления исходного файла и позволит обновлять файл резервной копии по мере внесения изменений в оригинал.
19. Выйти с **serverb**.

Оценка

На рабочей станции запустите сценарий **lab files-review grade**, чтобы подтвердить успех этой лабораторной работы.

```
[student@workstation ~]$ lab files-review grade
```

Завершение

На рабочей станции запустите сценарий **lab files-review finish**, чтобы завершить лабораторную работу. Данный сценарий удаляет все файлы и каталоги, созданные на **serverb** во время выполнения работы.

```
[student @ workstation ~] $ lab files-review finish
```

На этом лабораторная работа завершена.

РЕШЕНИЕ

УПРАВЛЕНИЕ ФАЙЛАМИ ИЗ КОМАНДНОЙ СТРОКИ

КОНТРОЛЬНЫЙ СПИСОК РАБОТЫ

В этой лабораторной работе вы будете эффективно создавать, перемещать и удалять файлы и каталоги с помощью команд оболочки `bash` и различных методов сопоставления имен файлов.

РЕЗУЛЬТАТЫ

Вы должен быть способны:

- Используйте подстановочные символы для поиска файлов и управления ими.

ПРЕЖДЕ ЧЕМ ВЫ НАЧНЕТЕ

Войдите на рабочую станцию как пользователь **student**, используя в качестве пароля слово **student**.

На рабочей станции выполните команду **lab files-review start**. Команда запускает сценарий запуска, который определяет, доступен ли сервер **serverb** в сети.

```
[student@workstation ~]$ lab files-review start
```

1. Используйте команду **ssh** для входа на **serverb** как пользователь **student**. Системы настроены на использование ключей SSH для аутентификации, поэтому пароль не требуется.

```
[student@workstation ~]$ ssh student@serverb  
[student@serverb ~]$
```

2. Перед созданием файлов проекта используйте команду **mkdir** с расширением фигурных скобок, чтобы создать пустые документы планирования проекта в каталоге **/home/student/Documents/project_plans**. (Подсказка: если каталога **~/Documents** не существует, параметр **-p** для команды **mkdir** создаст его.)
Создайте два пустых файла в каталоге **~/Documents/project_plans**:
season1_project_plan.odf и **season2_project_plan.odf**.

```
[student@serverb ~]$ mkdir -p ~/Documents/project_plans  
[student@serverb ~]$ touch ~/Documents/project_plans/{season1,season2}_project_plan.odf  
[student@serverb ~]$ ls -lR Documents/  
Documents:/  
total 0
```

```
drwxrwxr-x. 2 student student 70 Jan 31 18:20 project_plans
Documents/project_plans:
total 0
-rw-rw-r--. 1 student student 0 Jan 31 18:20 season1_project_plan.odf
-rw-rw-r--. 1 student student 0 Jan 31 18:20 season2_project_plan.odf
```

3. Создайте наборы пустых файлов практики для использования в этой лабораторной работе. Если вы не сразу поймёте, как создать синтаксическое выражение для подстановки в оболочки, используйте **решение** для обучения и практики. Используйте клавишу табуляции, чтобы легко завершать набор имени пути к файлам.
- Создайте в общей сложности 12 файлов с именами **tv_seasonX_episodeY.ogg**. Замените **X** номером сезона, а **Y** - эпизодом этого сезона, чтобы получить два сезона по шесть серий в каждом.

```
[student@serverb ~]$ touch tv_season{1..2}_episode{1..6}.ogg
[student@serverb ~]$ ls tv*
tv_season1_episode1.ogg tv_season1_episode5.ogg tv_season2_episode3.ogg
tv_season1_episode2.ogg tv_season1_episode6.ogg tv_season2_episode4.ogg
tv_season1_episode3.ogg tv_season2_episode1.ogg tv_season2_episode5.ogg
tv_season1_episode4.ogg tv_season2_episode2.ogg tv_season2_episode6.ogg
```

4. Как автор успешной серии детективных романов, главы вашего следующего бестселлера редактируются для публикации. Создайте в общей сложности восемь файлов с именами **mystery_chapterX.odf**. Замените **X** цифрами от **1** до **8**.

```
[student@serverb ~]$ touch mystery_chapter{1..8}.odf
[student@serverb ~]$ ls mys*
mystery_chapter1.odf mystery_chapter4.odf mystery_chapter7.odf
mystery_chapter2.odf mystery_chapter5.odf mystery_chapter8.odf
mystery_chapter3.odf mystery_chapter6.odf
```

5. С помощью одной команды создайте два подкатаога с именами **Season1** и **Season2** в каталоге **Videos**, чтобы организовать сериалы.

```
[student@serverb ~]$ mkdir -p Videos/season{1..2}
[student@serverb ~]$ ls Videos
season1 season2
```

6. Переместите соответствующие сериалы в подкаталоги сезонов. Используйте только две команды, указав места назначения с использованием относительного синтаксиса.

```
[student@serverb ~]$ mv tv_season1* Videos/season1
[student@serverb ~]$ mv tv_season2* Videos/season2
[student@serverb ~]$ ls -R Videos
```

Videos:

season1 season2

Videos/season1:

tv_season1_episode1.ogg tv_season1_episode3.ogg tv_season1_episode5.ogg

tv_season1_episode2.ogg tv_season1_episode4.ogg tv_season1_episode6.ogg

Videos/season2:

tv_season2_episode1.ogg tv_season2_episode3.ogg tv_season2_episode5.ogg

tv_season2_episode2.ogg tv_season2_episode4.ogg tv_season2_episode6.ogg

7. Создайте двухуровневую иерархию каталогов с помощью одной команды, чтобы организовать главы загадочной книги. Создайте **my_bestseller** в каталоге **Documents** и **chapters** в новом каталоге **my_bestseller**.

```
[student@serverb ~]$ mkdir -p Documents/my_bestseller/chapters
```

```
[student@serverb ~]$ ls -R Documents
```

Documents:

my_bestseller project_plans

Documents/my_bestseller:

chapters

Documents/my_bestseller/chapters:

Documents/project_plans:

season1_project_plan.odf season2_project_plan.odf

8. Создайте еще три подкатаога непосредственно в каталоге **my_bestseller** с помощью одной команды. Назовите эти подкаталоги **editor**, **changes** и **vacation**. Параметр **-p** (создать родителей (**parents**)) не требуется, поскольку родительский каталог **my_bestseller** уже существует.

```
[student@serverb ~]$ mkdir Documents/my_bestseller/{editor,changes,vacation}
```

```
[student@serverb ~]$ ls -R Documents
```

Documents:

my_bestseller project_plans

Documents/my_bestseller:

changes chapters editor vacation

Documents/my_bestseller/changes:

Documents/my_bestseller/chapters:

Documents/my_bestseller/editor:

Documents/my_bestseller/vacation:

Documents/project_plans:

season1_project_plan.odf season2_project_plan.odf

9. Перейдите в каталог **chapters**. Используя синтаксическое расширение для обозначения домашнего каталога тильды (~) для указания исходных файлов, переместите все главы книги в каталог **chapters**, который теперь является вашим текущим каталогом. Каков самый простой синтаксис для указания каталога назначения?

```
[student@serverb ~]$ cd Documents/my_bestseller/chapters  
[student@serverb chapters]$ mv ~/mystery_chapter*.  
[student@serverb chapters]$ ls  
mystery_chapter1.odf mystery_chapter4.odf mystery_chapter7.odf  
mystery_chapter2.odf mystery_chapter5.odf mystery_chapter8.odf  
mystery_chapter3.odf mystery_chapter6.odf
```

10. Вы отправили первые две главы на проверку редактору. Переместите только эти две главы в каталог **editor**, чтобы не изменять их во время рецензирования. Начиная с подкаталога **chapters**, используйте расширение скобок с диапазоном, чтобы указать имена файлов глав, которые нужно переместить, и относительный путь для каталога назначения.

```
[student@serverb chapters]$ mv mystery_chapter{1..2}.odf ../editor  
[student@serverb chapters]$ ls  
mystery_chapter3.odf mystery_chapter5.odf mystery_chapter7.odf  
mystery_chapter4.odf mystery_chapter6.odf mystery_chapter8.odf  
[student@serverb chapters]$ ls ../editor  
mystery_chapter1.odf mystery_chapter2.odf
```

11. Находясь в отпуске, вы собираетесь написать главы 7 и 8. Используйте одну команду, чтобы переместить файлы из каталога **chapters** в каталог **vacation**. Укажите имена файлов глав с помощью расширения скобок со списком строк и без использования подстановочных знаков

```
[student@serverb chapters]$ mv mystery_chapter{7,8}.odf ../vacation  
[student@serverb chapters]$ ls  
mystery_chapter3.odf mystery_chapter5.odf  
mystery_chapter4.odf mystery_chapter6.odf  
[student@serverb chapters]$ ls ../vacation  
mystery_chapter7.odf mystery_chapter8.odf
```

12. Измените рабочий каталог на **~/Videos/season2**, а затем скопируйте первую серию сезона в каталог **vacation**.

```
[student@serverb chapters]$ cd ~/Videos/season2  
[student@serverb season2]$ cp *episode1.ogg ~/Documents/my_bestseller/vacation
```

13. Используйте одну команду **cd**, чтобы перейти из рабочего каталога в каталог **~/Documents/my_bestseller/vacation**. Перечислите его файлы. Используйте предыдущий аргумент рабочего каталога, чтобы вернуться в каталог **season2**. (Действие будет успешным, если последнее изменение каталога с помощью команды **cd** было выполнено с помощью одной команды, а не нескольких команд **cd**.) Из каталога **season2** скопируйте файл **episode 2** в каталог **vacation**. Снова используйте синтаксическое выражение для подстановки, чтобы вернуться в каталог **vacation**.

14. Авторы глав 5 и 6 хотят поэкспериментировать с возможными изменениями. Скопируйте оба файла из каталога `~/Documents/my_bestseller/chapters` в каталог `~/Documents/my_bestseller/changes`, чтобы предотвратить изменение исходных файлов при редактировании глав. Перейдите в каталог `~/Documents/my_bestseller`. Используйте сопоставление с образцом в квадратных скобках, чтобы указать, какие номера глав должны совпадать в аргументе имени файла команды `cp`.

```
[student@serverb vacation]$ cd ~/Documents/my_bestseller  
[student@serverb my_bestseller]$ cp chapters/mystery_chapter[56].odf changes  
[student@serverb my_bestseller]$ ls chapters  
mystery_chapter3.odf mystery_chapter5.odf  
mystery_chapter4.odf mystery_chapter6.odf  
[student@serverb my_bestseller]$ ls changes  
mystery_chapter5.odf mystery_chapter6.odf
```

15. Измените текущий каталог на каталог `changes`.

Используйте команду `date +%F` с заменой, чтобы скопировать `mystery_chapter5.odf` в новый файл, который включает полную дату. Имя должно иметь вид `mystery_chapter5_YYYY-MM-DD.odf`.

Сделайте еще одну копию `mystery_chapter5.odf`, добавив текущую отметку времени (как количество секунд с начала `1970-01-01 00:00 UTC`), чтобы гарантировать уникальное имя файла. Для этого используйте подстановку команд с командой `date +%s`.

```
[student@serverb my_bestseller]$ cd changes  
[student@serverb changes]$ cp mystery_chapter5.odf  
> mystery_chapter5_$(date +%F).odf  
[student@serverb changes]$ cp mystery_chapter5.odf \  
> mystery_chapter5_$(date +%s).odf  
[student@serverb changes]$ ls  
mystery_chapter5_1492545076.odf mystery_chapter5.odf  
mystery_chapter5_2017-04-18.odf mystery_chapter6.odf
```

16. После дополнительного просмотра вы решаете, что в изменении сюжета нет необходимости. Удалите каталог `changes`.

При необходимости перейдите в каталог `changes` и удалите все файлы в этом каталоге. Вы не можете удалить каталог, пока он является текущим рабочим каталогом. Перейдите в родительский каталог каталога `changes`. Попробуйте удалить пустой каталог с помощью команды `rm` без рекурсивной опции `-r`. Эта попытка должна потерпеть неудачу. Наконец, используйте команду `rmdir` для удаления пустого каталога, что будет успешным.

```
[student@serverb changes]$ rm mystery*  
[student@serverb changes]$ cd ..  
[student@serverb my_bestseller]$ rm changes  
rm: cannot remove 'changes': Is a directory  
[student@serverb my_bestseller]$ rmdir changes  
[student@serverb my_bestseller]$ ls  
chapters editor vacation
```

17. По окончании отпуска каталог **vacation** больше не нужен. Удалите его с помощью команды **rm** с опцией рекурсии. По завершении вернитесь в домашний каталог пользователя **student**.

```
[student@serverb my_bestseller]$ rm -r vacation
[student@serverb my_bestseller]$ ls
chapters editor
[student@serverb my_bestseller]$ cd
[student@serverb ~]$
```

18. Создайте жесткую ссылку на файл **~/Documents/project_plans/season2_project_plan.odf** симе нем **~/Documents/backups/season2_project_plan.odf.back**.

Жесткая ссылка защитит от случайного удаления исходного файла и позволит обновлять файл из зерновой копии,而不ревнесения изменений в оригинал.

```
[student@serverb ~]$ mkdir ~/Documents/backups
[student@serverb ~]$ ln ~/Documents/project_plans/season2_project_plan.odf \
> ~/Documents/backups/season2_project_plan.odf.back
[student@serverb ~]$ ls -lR ~/Documents/
/home/student/Documents/:
total 0
drwxrwxr-x. 2 student student 43 Jan 31 18:59 backups
drwxrwxr-x. 4 student student 36 Jan 31 19:42 my_bestseller
drwxrwxr-x. 2 student student 70 Jan 31 18:20 project_plans
/home/student/Documents/backups:
total 4
-rw-rw-r--. 2 student student 0 Jan 31 19:05 season2_project_plan.odf.back

/home/student/Documents/my_bestseller:
total 0
drwxrwxr-x. 2 student student 118 Jan 31 19:39 chapters
drwxrwxr-x. 2 student student 62 Jan 31 19:38 editor

/home/student/Documents/my_bestseller/chapters:
total 0
-rw-rw-r--. 1 student student 0 Jan 31 19:18 mystery_chapter3.odf
-rw-rw-r--. 1 student student 0 Jan 31 19:18 mystery_chapter4.odf
-rw-rw-r--. 1 student student 0 Jan 31 19:18 mystery_chapter5.odf
-rw-rw-r--. 1 student student 0 Jan 31 19:18 mystery_chapter6.odf

/home/student/Documents/my_bestseller/editor:
total 0
-rw-rw-r--. 1 student student 0 Jan 31 19:18 mystery_chapter1.odf
-rw-rw-r--. 1 student student 0 Jan 31 19:18 mystery_chapter2.odf
/home/student/Documents/project_plans:
total 4
-rw-rw-r--. 1 student student 0 Jan 31 18:20 season1_project_plan.odf
-rw-rw-r--. 2 student student 0 Jan 31 19:05 season2_project_plan.odf
```

19. Выйти с **serverb**.

```
[student@serverb ~]$ exit  
logout  
Connection to serverb closed.  
[student@workstation ~]$
```

Оценка

На рабочей станции запустите сценарий **lab files-review grade**, чтобы подтвердить успех этой лабораторной работы.

```
[student@workstation ~]$ lab files-review grade
```

Завершение

На рабочей станции запустите сценарий **lab files-review finish**, чтобы завершить лабораторную работу. Данный сценарий удаляет все файлы и каталоги, созданные на **serverb** во время выполнения работы.

```
[student @ workstation ~] $ lab files-review finish
```

На этом лабораторная работа завершена.

РЕЗЮМЕ

В этой главе вы узнали:

- Файлы в системе Linux организованы в единое перевернутое дерево каталогов, известное как иерархия файловой системы.
- Абсолютные пути начинаются с символа `/` и указывают расположение файла в иерархии файловой системы.
- Относительные пути не начинаются с символа `/` и указывают расположение файла относительно текущего рабочего каталога.
- Для управления файлами используются пять ключевых команд: `mkdir`, `rmdir`, `cp`, `mv` и `rm`.
- Жесткие и символьические ссылки - это разные способы, чтобы несколько имен файлов указывали на одни и те же данные.
- Оболочка `Bash` предоставляет функции сопоставления с образцом, расширения и замены, которые помогут вам эффективно выполнять команды.

ГЛАВА 4

ПОЛУЧЕНИЕ ПОМОЩИ В RED HAT ENTERPRISE LINUX

ЦЕЛЬ

Решайте проблемы с помощью локальной справочной системы.

ЗАДАЧИ

- Найдите информацию на страницах руководства по локальной системе Linux.
- Найдите информацию в местной документации в GNU Info.

РАЗДЕЛЫ

- Чтение страниц руководства (и упражнения с пошаговыми инструкциями)
- Чтение информационной документации (и упражнения с пошаговыми инструкциями)

ЛАБОРОТОРНАЯ РАБОТА

Получение помощи в Red Hat Enterprise Linux

ЧТЕНИЕ СТРАНИЦ РУКОВОДСТВА

ВВЕДЕНИЕ ЧЕЛОВЕЧЕСКОЙ КОМАНДЫ

Одним из источников документации, обычно доступной в локальной системе, являются системные справочные страницы или страницы *man*. Эти страницы поставляются как часть программных пакетов, для которых они предоставляют документацию, и к ним можно получить доступ из командной строки с помощью команды **man**.

Историческое Руководство программиста Linux, из которого берут начало страницы руководства, было достаточно большим, чтобы вместить несколько печатных разделов. Каждый раздел содержит информацию по определенной теме.

Общие разделы руководства по Linux

РАЗДЕЛ	ТИП СОДЕРЖИМОГО
1	Пользовательские команды (как исполняемые программы, так и программы оболочки)
2	Системные вызовы (процедуры ядра, вызываемые из пользовательского пространства)
3	Библиотечные функции (предоставляются программными библиотеками)
4	Специальные файлы (например, файлы устройств)
5	Форматы файлов (для многих конфигурационных файлов и структур)
6	Игры (исторический раздел забавных программ)
7	Конвенции, стандарты и прочее (протоколы, файловые системы)
8	Системное администрирование и привилегированные команды (задачи обслуживания)
9	API ядра Linux (внутренние вызовы ядра)

Чтобы различать идентичные названия тем в разных разделах, ссылки на страницы руководства включают номер раздела в круглых скобках после темы. Например, **passwd(1)** описывает команду для изменения паролей, а **passwd(5)** объясняет формат файла **/etc/passwd** для хранения локальных учетных записей пользователей.

Чтобы прочитать определенные страницы руководства, используйте формат команды **man topic (имя_страницы)**. Содержание отображается по одному экрану. Команда **man** выполняет поиск разделов руководства в алфавитно-цифровом порядке. Например, **man passwd** по умолчанию отображает **passwd(1)**. Чтобы отобразить тему справочной страницы из определенного раздела, включите аргумент номера раздела: **man 5 passwd** отображает **passwd(5)**.

НАВИГАЦИЯ И ПОИСК ПО СТРАНИЦАМ

Способность эффективно искать имя страницы и перемещаться по страницам руководства **man** является критически важным навыком администратора. Инструменты графического интерфейса пользователя упрощают настройку общих системных ресурсов, но использование интерфейса командной строки по-прежнему более эффективно. Чтобы продуктивно перемещаться по командной строке, вы должны уметь находить нужную информацию на страницах руководства.

В следующей таблице перечислены основные команды навигации при просмотре страниц руководства **man**:

Навигация по страницам руководства **man**

КОМАНДА	РЕЗУЛЬТАТ
Пробел	Прокрутка вперед (вниз) на один экран
PageDown	Прокрутка вперед (вниз) на один экран
PageUp	Прокрутка назад (вверх) на один экран
Стрелка вниз	Прокрутите вперед (вниз) на одну строку
Стрелка вверх	Прокрутка назад (вверх) на одну строку
D	Прокрутка вперед (вниз) на половину экрана
U	Прокрутка назад (вверх) на половину экрана
/string	Искать вперед (вниз) строку на странице руководства
N	Повторить предыдущий поиск вперед (вниз) на странице руководства
Shift + N	Повторить предыдущий поиск назад (вверх) на странице руководства
G	Перейти к началу страницы руководства.
Shift + G	Перейти в конец страницы руководства.
Q	Выходите из man и вернитесь в командную строку.



ВАЖНО

При выполнении поиска **/string** допускает синтаксис регулярных выражений. В то время как простой текст (например, **passwd**) работает должным образом, в регулярных выражениях используются метасимволы (такие как **\$**, *****, **.** и **^**) для более сложных сопоставлений с образцом. Следовательно, поиск по строкам, содержащим метасимволы программного выражения, например, **make \$\$\$**, может привести к неожиданным результатам.

Регулярные выражения и синтаксис обсуждаются в книге *Red Hat System Administration II* и в название страницы **man regex(7)**.

Чтение страниц руководства

Каждая тема разделена на несколько частей. Большинство тем имеют одинаковые заголовки и представлены в одном порядке. Обычно тема содержит не все заголовки, поскольку не все заголовки применимы ко всем темам.

Общие заголовки:

ЗАГОЛОВОК	ОПИСАНИЕ
НАЗВАНИЕ (NAME)	Имя субъекта. Обычно это имя команды или файла. Очень краткое описание.
ОБЗОР (SYNOPSIS)	Краткое описание синтаксиса команды.
ОПИСАНИЕ (DESCRIPTION)	Подробное описание, обеспечивающее базовое понимание темы.
ПАРАМЕТРЫ (OPTIONS)	Объяснение вариантов выполнения команды.
ПРИМЕРЫ (EXAMPLES)	Примеры использования команды, функции или файла.
ФАЙЛЫ (FILES)	Список файлов и каталогов, относящихся к странице руководства.
СМОТРИТЕ ТАКЖЕ (SEE ALSO)	Связанная информация, обычно это другие разделы справочной страницы.
ОШИБКИ (BUGS)	Известные ошибки в программном обеспечении.
АВТОР (AUTHOR)	Информация о том, кто способствовал развитию темы.

ПОИСК MAN-СТРАНИЦ ПО КЛЮЧЕВОМУ СЛОВУ

Поиск, по ключевым словам, по страницам руководства выполняется с помощью команды **man -k keyword**, которая отображает список тем справочной страницы, соответствующих ключевым словам, с указанием номеров разделов.

```
[student@desktopX ~]$ man -k passwd
checkPasswdAccess (3) - query the SELinux policy database in the kernel.
chpasswd (8)          - update| passwords in batch mode
ckpasswd (8)          - nnrpd password authenticator
fgetpwent_r (3)       - get passwd file entry reentrantly
getpwent_r (3)        - get passwd file entry reentrantly
...
passwd (1)            - update user's authentication tokens
sslpasswd (1ssl)      - compute password hashes
passwd (5)            - password file
passwd.nntp (5)       - Passwords for connecting to remote NNTP servers
passwd2des (3)        - RFS password encryption
...
```

Популярные темы системного администрирования находятся в разделах 1 (пользовательские команды), 5 (форматы файлов) и 8 (административные команды). Администраторы, использующие определенные инструменты для устранения неполадок, также используют раздел 2 (системные вызовы). Остальные разделы предназначены, как правило, для справки программистам или для расширенного администрирования.



ПРИМЕЧАНИЕ

Поиск, по ключевым словам, основывается на индексе, сгенерированном командой **mandb** (8), которая должна запускаться от имени пользователя **root**. Команда запускается ежедневно через **cron.daily** или через **anacrontab** в течение часа после загрузки, если она устарела.



ВАЖНО

Параметр команды **man -K** (верхний регистр) выполняет полнотекстовый поиск страницы, а не только заголовки и описания, такие как параметр **-k**. Полнотекстовый поиск использует больше системных ресурсов и занимает больше времени.



РЕКОМЕНДАЦИИ

Страницы руководства **man man(1)**, **mandb(8)**, **man-pages(7)**, **less(1)**, **intro(1)**, **intro(2)**, **intro(5)**, **intro(7)**, **intro(8)**

УПРАЖНЕНИЯ С ПОШАГОВЫМИ ИНСТРУКЦИЯМИ

В этом упражнении вы попрактикуетесь в поиске нужной информации, используя опции и аргументы **man**.

РЕЗУЛЬТАТЫ

Вы должны уметь использовать систему **manLinux**, находить полезную информацию с помощью поиска и просмотра.

ПРЕЖДЕ ЧЕМ ВЫ НАЧНЕТЕ

Войдите на рабочую станцию как **student**, используя в качестве пароля слово **student**.

На рабочей станции запустите команду **lab help-manual start**. Скрипт при выполнение создаёт файл с именем **manual**.

```
[student @ workstation ~] $ lab help-manual start
```

1. На рабочей станции просмотрите справочную страницу **gedit**. Просмотрите параметры редактирования определенного файла с помощью **gedit** из командной строки. Используйте один из вариантов на странице руководства **gedit**, чтобы открыть файл **/home/student/manual** с помощью команды **gedit** с курсором в конце файла.

1.1. Просмотрите справочную страницу **gedit**.

```
[студент @ рабочая станция ~] $ mangedit
```

```
GEDIT(1)      General Commands Manual      GEDIT(1)
NAME
       gedit - text editor for the GNOME Desktop

SYNOPSIS
       gedit [OPTION...] [FILE...] [+LINE[:COLUMN]]
       gedit [OPTION...] -
...output omitted...
```

- 1.2. На странице руководства **gedit** изучите параметры редактирования определенного файла из командной строки.

```
...output omitted...
       FILE  Specifies the file  to open when gedit starts.
...output omitted...
       +LINE  For the first file, go to the line specified by LINE (do not insert
a space between the "+" sign and the number). If LINE is missing, go to the last
line.
```

```
...output omitted...
```

Нажмите **q**, чтобы закрыть страницу руководства.

1.3. Используйте команду **gedit +**, чтобы открыть файл **manual**. Отсутствующий номер строки рядом с опцией + открывает файл, переданный в качестве аргумента, с курсором в конце последней строки.

```
[student@workstation ~]$ gedit + manual
```

the quick brown fox just came over to greet the lazy poodle!

Убедитесь, что файл открыт с курсором в конце последней строки в файле. Нажмите **Ctrl + Q**, чтобы закрыть приложение.

2. Прочтите man страницу su(1).

Обратите внимание: когда пользователь не указан, команда su предполагает, что пользователь является пользователем **root**. Если за командой **su** следует тире (-), запускается дочерняя оболочка входа в систему. Без тире создается дочерняя оболочка без входа в систему, которая соответствует текущей среде пользователя.

```
[student@workstation ~]$ man 1 su
```

```
SU(1)           User Commands           SU(1)
NAME
    su - run a command with substitute user and group ID

SYNOPSIS
    su [options] [-] [user [argument...]]

DESCRIPTION
    su allows to run commands with a substitute user and group ID.
    When called without arguments, su defaults to running an interactive
    shell as root.
...output omitted...
OPTIONS
...output omitted...
-, -l, --login
    Start the shell as a login shell with an environment similar to a real login
...output omitted...
```



ПРИМЕЧАНИЕ

Обратите внимание, что параметры, разделенные запятыми в одной строке, такие как **-**, **-l** и **--login**, приводят к одинаковому поведению.

Нажмите **q**, чтобы закрыть справочную страницу.

- Команда **man** также имеет свои собственные справочные страницы.

```
[student@workstation ~]$ man man
```

MAN(1)	Manual pager utils	MAN(1)
NAME		
man - an interface to the on-line reference manuals		
...output omitted...		
DESCRIPTION		
man is the system's manual pager. Each page argument given to man is normally the name of a program, utility or function. The manual page associated with each of these arguments is then found and displayed. A section, if provided, will direct man to look only in that section of the manual.		
...output omitted...		

Нажмите **q**, чтобы закрыть справочную страницу.

- Все страницы руководства расположены в **/usr/share/man**. Найдите двоичный файл, исходный код и страницы руководства, расположенные в каталоге **/usr/share/man**, с помощью команды **whereis**.

```
[student@workstation ~]$ whereis passwd
```

```
passwd: /usr/bin/passwd /etc/passwd /usr/share/man/man1/passwd.1.gz /usr/share/
man/man5/passwd.5.gz
```

- Используйте команду **man -k zip** для вывода подробной информации о ZIP-архиве.

```
[student@workstation ~]$ man -k zip
```

```
...output omitted...
zipinfo (1)          - list detailed information about a ZIP archive
zipnote (1)          - write the comments in zipfile to stdout, edit comments and
                      rename files in zipfile
zipsplit (1)         - split a zipfile into smaller zipfiles
```

6. Используйте команду `man -k boot`, чтобы отобразить страницу руководства, содержащую список параметров, которые могут быть переданы ядру во время загрузки.

```
[студент @ рабочая станция ~] $ man -k boot
```

```
...output omitted...
bootctl (1)           - Control the firmware and boot manager settings
bootparam (7)          - introduction to boot time parameters of the Linux kernel
bootup (7)             - System bootup process
...output omitted...
```

7. Используйте команду `man -k ext4`, чтобы найти команду, используемую для настройки параметров файловой системы `ext4`.

```
[студент @ рабочая станция ~] $ man -k ext4
```

```
...output omitted...
resize2fs (8)          - ext2/ext3/ext4 file system resizer
tune2fs (8)            - adjust tunable filesystem parameters on ext2/ext3/ext4
filesystems
```

ЗАВЕРШЕНИЕ

На рабочей станции запустите сценарий завершения лабораторной работы `lab help-manual finish`, чтобы завершить упражнение.

```
[student @ workstation ~] $ lab help-manual завершение
```

На этом пошаговое упражнение завершено.

ЧТЕНИЕ ДОКУМЕНТАЦИИINFO

ЗАДАЧИ

После завершения этого раздела студенты смогут найти информацию в локальной документации в **GNU Info**.

ПРЕДСТАВЛЕНИЕ GNU INFO

Страницы руководства имеют формат, полезный в качестве справочника по командам, но менее полезный в качестве общей документации. Для таких документов проект **GNU** разработал другую систему онлайн-документации, известную как **GNU Info**. Информационные документы являются важным ресурсом в системе Red Hat Enterprise Linux, поскольку многие фундаментальные компоненты и утилиты, такие как пакет **coreutils** и стандартные библиотеки **libc**, либо разрабатываются проектом **GNU**, либо используют систему документов **Info**.



ВАЖНО

Вы можете задаться вопросом, почему существуют две локальные системы документации, справочные страницы и информационные документы. Некоторые из причин для этого имеют практический характер, а некоторые связаны с тем, как Linux и его приложения разрабатывались различными сообществами с открытым исходным кодом на протяжении многих лет.

Страницы руководства имеют гораздо более формальный формат и обычно документируют определенную команду или функцию из программного пакета и структурированы как отдельные текстовые файлы. Информационные документы обычно охватывают отдельные программные пакеты в целом, обычно содержат более практические примеры того, как использовать программное обеспечение, и структурированы как гипертекстовые документы.

Вы должны быть знакомы с обеими системами, чтобы максимально использовать информацию, доступную вам из системы.

Чтение info документации

Чтобы запустить программу просмотра документов **info**, используйте команду **pinfo**. **pinfo** открывается в начальном каталоге.

Рисунок 4.1: Программа просмотра документов **info** **pinfo**, верхний каталог

The screenshot shows a terminal window titled "File: dir" and "Node: Top". The title bar also says "This is the top of the INFO tree". The main content area displays a menu of major topics:

- This (the Directory node) gives a menu of major topics.
- Typing "q" exits, "?" lists all Info commands, "d" returns here,
- "h" gives a primer for first-timers,
- "mEmacs<Return>" visits the Emacs topic, etc.

Below this, instructions for Emacs users are given:

In Emacs, you can click mouse button 2 on a menu item or cross reference to select it.

* Menu:

Archiving

- * Cpio: ([cpio](#)). Copy-in-copy-out archiver to tape or disk.
- * Tar: ([tar](#)). Making tape (or disk) archives.

Basics

- * Common options: ([coreutils](#))[Common options](#). Common options.
- * Coreutils: ([coreutils](#)). Core GNU (file, text, shell) utilities.
- * Date input formats: ([coreutils](#))[Date input formats](#).
- * File permissions: ([coreutils](#))[File permissions](#). Access modes.
- * Finding files: ([find](#)). Operating on files matching certain criteria.
- * ed: ([ed](#)). The GNU Line Editor.

Viewing line 25/2002, 1%

Информационная документация исчерпывающая и снабжена гиперссылками. Информационные страницы можно выводить в несколько форматов. Напротив, страницы руководства оптимизированы для вывода на печать. Формат **info** более гибкий, чем справочные страницы **man**, что позволяет подробно обсуждать сложные команды и концепции. Как и страницы руководства, информационные узлычитываются из командной строки с помощью команды **pinfo**.

Типичная страница руководства имеет небольшой объем контента, посвященный одной конкретной теме, команде, инструменту или файлу. Информационная документация представляет собой исчерпывающий документ. **info** предоставляет следующие улучшения:

- Единый документ для большой системы, содержащий всю необходимую информацию для этой системы.
- Гиперссылки
- Полный указатель документов с возможностью просмотра
- Полнотекстовый поиск по всему документу

Некоторые команды и утилиты имеют как справочные страницы, так и документацию **info**; обычно документация **info** более подробна. Сравните различия в документации **tar**, используя **man** и **pinfo**:

```
[user@host ~]$ mantar
[user@host ~]$ pinfotar
```

Команда **pinfo** более продвинута, чем исходная команда **info**. Чтобы просмотреть конкретный вопрос (topic), используйте команду **pinfotopic**. Команда **pinfo** без аргументов открывает верхний каталог. Новая документация становится доступной в **pinfo** после установки пакетов программного обеспечения.



ПРИМЕЧАНИЕ

Если для конкретной запрошенной вами информации в системе нет раздела info, info будет искать соответствующую справочную **man** страницу и отобразит её содержание.

СРАВНЕНИЕ НАВИГАЦИИ В GNU INFO И ПО СТРАНИЦАМ MAN

Команда **pinfo** и команда **man** используют несколько разные навигационные нажатия клавиш. В следующей таблице сравниваются нажатия клавиш для обеих команд:

Сравнение привязки действия к клавишам при навигации в **pinfo** и **man**

НАВИГАЦИЯ	PINFO	MAN
Прокрутка вперед (вниз) на один экран	PageDown или пробел	PageDown или пробел
Прокрутка назад (вверх) на один экран	PageUp или b	PageUp или b
Показать каталог тем	D	-
Прокрутка вперед (вниз) на половину экрана	-	D
Показать родительский узел темы	U	-
Отображать начало (вверху) темы	HOME	G
Прокрутка назад (вверх) на половину экрана	-	U
Прокрутите вперед (вниз) до следующей гиперссылки	Стрелка вниз	-
Открыть тему в месте нахождения курсора	Enter	-
Прокрутите вперед (вниз) на одну строку	-	Стрелка вниз или Enter
Прокрутка назад (вверх) к предыдущей гиперссылке	Стрелка вверх	-
Прокрутка назад (вверх) на одну строку	-	Стрелка вверх
Искать по шаблону	<i>/string</i>	<i>/string</i>
Показать следующий узел (главу) в теме	N	-
Повторить предыдущий поиск вперед (вниз)	<i>/</i> затем Enter	n
Показать предыдущий узел (главу) в теме	n	-

Повторить предыдущий поиск назад (вверх)	-	Shift+N
Выйти из программы	q	q



РЕКОМЕНДАЦИИ

pinfoinfo (*Info: Введение*)

pinfopinfo (*Документация для pinfo*)

The GNU Project

<http://www.gnu.org/gnu/thegnuproject.html>

Страницы руководства **manpinfo(1)** и **info(1)**

УПРАЖНЕНИЯ С ПОШАГОВЫМИ ИНСТРУКЦИЯМИ

ЧТЕНИЕ ДОКУМЕНТАЦИИINFO

В этом упражнении вы будете искать информацию, хранящуюся в документах **GNU Info**, перемещаясь по этим документам с помощью инструментов командной строки.

РЕЗУЛЬТАТЫ

Вы должны иметь возможность перемещаться по документации **GNU Info** с помощью инструментов командной строки.

ПРЕЖДЕ ЧЕМ ВЫ НАЧНЕТЕ

Войдите на рабочую станцию как пользователь **student**, используя в качестве пароля слово **student**.

На рабочей станции запустите команду **lab help-info start**.

```
[student@workstation ~]$ lab help-info start
```

1. На рабочей станции без аргументов выполните команду **pinfo**.

```
[student@workstation ~]$ pinfo
```

2. Перейдите к разделу Общие параметры (**Common options**).

Используйте стрелку вверх (**UpArrow**) или стрелку вниз (**DownArrow**), пока не будут выделены (**coreutils**)**Common options**.

Рисунок 4.2: Документация Bash

The screenshot shows a terminal window with the command `lab help-info start` entered. The output displays the `pinfo` command's documentation. A specific section, `* Common options: (coreutils)Common options.`, is highlighted in blue, indicating it has been selected with the Up or Down Arrow keys. The title of the page is `Bash: (bash).` and the subtitle is `The GNU Bourne-Again SHeLL.`

3. Нажмите **Enter**, чтобы просмотреть эту тему.

Рисунок 4.3: Тема с информацией об Общих параметрах (Common options)

```

File: coreutils.info,  Node: Common options,  Next: Output of entire files,  Prev: Introduction,  Up: Top

2 Common options
*****  

Certain options are available in all of these programs. Rather than
writing identical descriptions for each of the programs, they are
described here. (In fact, every GNU program accepts (or should accept)
these options.)  

Normally options and operands can appear in any order, and programs
act as if all the options appear before any operands. For example,
'sort -r passwd -t :' acts like 'sort -r -t : passwd', since ':' is an
option-argument of '-t'. However, if the 'POSIXLY_CORRECT' environment
variable is set, options must appear before operands, unless otherwise
specified for a particular command.

```

4. Просмотрите этот информационный раздел. Узнайте, можно ли сокращать варианты длинного стиля.

Используйте **PageUp** и **PageDown** для навигации по теме. Да, многие программы позволяют сокращать длинные параметры.

5. Определите, что обозначают символы `--` при использовании в качестве аргумента команды.

Символы `--` обозначают конец параметров команды и начало аргументов команды в сложных командах, где синтаксический анализатор командной строки оболочки может неправильно различить их.

6. Не выходя из **pinfo**, перейдите к узлу **GNUCoreutils**.

Нажмите **U**, чтобы перейти к верхнему узлу темы.

7. Вернитесь к теме верхнего уровня.

Снова нажмите **U**. Обратите внимание, что при размещении вверху узла темы перемещение вверх возвращается в каталог тем. Или же нажатие **D** с любого уровня или темы позволяет перейти непосредственно в каталог тем.

8. Найдите шаблон **coreutils** и выберите эту тему. Нажмите `/`, а затем введите шаблон поиска «**coreutils**». Выделив тему, нажмите **Enter**.

Рисунок 4.4: Результат поиска

```
* Coreutils: (coreutils).          Core GNU (file, text, shell) utilities.
```

9. В меню вверху найдите и выберите «Вывод файлов целиком (**Output of entire files**)», нажав **N**.

```
* Output of entire files::      cat tac nl od base32 base64
```

Просмотрите тему.

Используйте **Enter**, чтобы выбрать **cat invocation**. Используйте клавиши со стрелками для просмотра темы.

```
* cat invocation::           Concatenate and write files.
```

- 10.** Перейдите на два уровня вверх, чтобы вернуться в **GNUCoreutils**. Найдите и выделите пункт **Summarizing files**.

Нажмите **Enter**, чтобы выбрать тему, затем просмотрите ее.

- 11.** Нажмите **Q**, чтобы выйти из **pinfo**.

- 12.** Снова используйте команду **pinfo**, указав **coreutils** в качестве целевой темы в командной строке.

```
[student@workstation ~]$ pinfo coreutils
```

- 13.** Выберите тему Использование диска (**Disk usage**).

Нажмите стрелку вниз, чтобы выделить Использование диска **Disk usage**, затем нажмите **Enter**, чтобы выбрать эту тему.

- 14.** Прочтите подтемы **df invocation** и **du invocation**.

Используйте клавиши со стрелками, чтобы выделить тему, **PageUp** и **PageDown**, чтобы просмотреть текст, затем нажмите **U**, чтобы перейти на один уровень вверх. По завершении нажмите **Q**, чтобы выйти.

ЗАВЕРШЕНИЕ

На рабочей станции запустите сценарий **lab help-info finish**, чтобы завершить упражнение.

```
[student@workstation ~]$ labhelp-info finish
```

На этом пошаговое упражнение завершено.

ЛАБОРАТОРНАЯ РАБОТА

ПОЛУЧЕНИЕ ПОМОЩИ В RED HAT ENTERPRISE LINUX

КОНТРОЛЬНЫЙ СПИСОК РАБОТЫ

В этой лабораторной работе вы будете искать информацию, которая поможет вам выполнять задачи на страницах руководств **man** и документах **GNU Info**.

В РЕЗУЛЬТАТЕ

Вы должны быть способны:

- Находить соответствующие команды, выполнив поиск по страницам руководства **man** и узлам **info**.
- Изучите новые параметры для часто используемых команд документации.
- Используйте соответствующие инструменты для просмотра и печати документации и других файлов в нетекстовом формате.

ПРЕЖДЕ ЧЕМ ВЫ НАЧНЕТЕ

Войдите на рабочую станцию как пользователь **student**, используя в качестве пароля слово **student**.

На рабочей станции запустите команду **lab help-review start**.

```
[student@workstation ~]$ lab help-review start
```

1. На рабочей станции определите, как подготовить справочную страницу к печати. В частности, найдите, какой формат или язык визуализации используется для печати.
2. Создайте форматированный выходной файл страницы руководства **passwd**. Назовите файл **passwd.ps**. Определите формат содержимого файла. Проверьте содержимое файла **passwd.ps**.



ПРИМЕЧАНИЕ

Создайте форматированный вывод страницы руководства **passwd**, используя следующую команду:

```
[student@workstation ~]$ man -t passwd > passwd.ps
```

Символ **>** *перенаправляет* содержимое справочной страницы в файл **passwd.ps**. Эта команда рассматривается более подробно в следующей главе.

- Используя **man**, изучите команды, используемые для просмотра и печати файлов **PostScript**.
- Узнайте, как использовать программу просмотра **evince(1)** в режиме предварительного просмотра. Кроме того, определите, как открыть документ, начиная с определенной страницы.
- Просмотрите свой файл **PostScript**, используя различные изученные вами варианты **evince**. Закройте файл документа, когда закончите.
- Используя команду **man**, исследуйте **lp(1)**, чтобы определить, как распечатать любой документ, начиная с определенной страницы. Фактически не вводя никаких команд (потому что нет принтеров), изучите синтаксис в одной команде, чтобы печатать только страницы 2 и 3 вашего файла **PostScript**.
- Используя **pinfo**, найдите документацию **GNU Info** о программе просмотра **evince**.
- Используя **Firefox**, откройте каталог документации пакета системы и перейдите в подкаталог пакета **man-db**. Просмотрите предоставленные руководства.
- С помощью браузера **Firefox** найдите и перейдите в подкаталог пакета **initscripts**. Просмотрите файл **sysconfig.txt**, в котором описаны важные параметры конфигурации системы, хранящиеся в каталоге **/etc/sysconfig**.

Оценка

На рабочей станции запустите **lab help-review grade**, чтобы подтвердить успешность выполнения упражнения.

```
[student@workstation ~]$ lab help-review grade
```

Завершение

На рабочей станции запустите сценарий **lab help-review finish**, чтобы завершить это упражнение.

```
[student@workstation ~]$ lab help-review finish
```

На этом лабораторная работа завершена.

РЕШЕНИЕ

ПОЛУЧЕНИЕ ПОМОЩИ В REDHAT ENTERPRISE LINUX

КОНТРОЛЬНЫЙ СПИСОК РАБОТЫ

В этой лабораторной работе вы будете искать информацию, которая поможет вам выполнять задачи на страницах руководств **man** и документах **GNU Info**.

В РЕЗУЛЬТАТЕ

Вы должны быть способны:

- Находить соответствующие команды, выполнив поиск по страницам руководства **man** и узлам **info**.
- Изучите новые параметры для часто используемых команд документации.
- Используйте соответствующие инструменты для просмотра и печати документации и других файлов в нетекстовом формате.

ПРЕЖДЕ ЧЕМ ВЫ НАЧНЕТЕ

Войдите на рабочую станцию как пользователь **student**, используя в качестве пароля слово **student**.

На рабочей станции запустите команду **lab help-review start**.

```
[student@workstation ~]$ lab help-review start
```

1. На рабочей станции определите, как подготовить справочную страницу к печати. В частности, найдите, какой формат или язык визуализации используется для печати.
 - 1.1. Используйте команду **man man**, чтобы определить, как подготовить справочную страницу к печати.

```
[student@worksation ~]$ man man
```



ПРИМЕЧАНИЕ

Для подготовки страницы руководствак печати с помощью PostScriptиспользует **man-t**.

2. Создайте форматированный выходной файл страницы руководства **passwd**. Назовите файл **passwd.ps**. Определите формат содержимого файла. Проверьте содержимое файла **passwd.ps**.



ПРИМЕЧАНИЕ

Создайте форматированный вывод страницы руководства **passwd**, используя следующую команду:

```
[student@workstation $]$ man -t passwd > passwd.ps
```

Символ **>** *перенаправляет* содержимое справочной страницы в файл **passwd.ps**. Эта команда рассматривается более подробно в следующей главе.

- 2.1. Используйте команду **man -t** для создания отформатированного файла страницы руководства **passwd**.

```
[student@workstation ~]$ man -t passwd > passwd.ps  
[student@workstation ~]$ ls -al  
...output omitted...  
-rw-rw-r--. 1 student student 19947 Feb 26 11:14 passwd.ps  
...output omitted...
```

- 2.2. Используйте команду **file**, чтобы определить формат содержимого файла.

```
[student@workstation ~]$ file /home/student/passwd.ps  
passwd.ps: PostScript document text conforming DSC level 3.0
```

- 2.3. Используйте команду **less** для просмотра файла **/home/student/passwd.ps**.

```
[student@workstation ~]$ less /home/student/passwd.ps  
% !PS-Adobe-3.0  
%%Creator: groff version 1.22.3  
%%CreationDate: Tue Feb 26 11:14:40 2019  
%%DocumentNeededResources: font Times-Roman  
%%+ font Times-Bold  
%%+ font Times-Italic  
%%+ font Symbol  
%%DocumentSuppliedResources: procset grops 1.22 3  
...output omitted...
```



ПРИМЕЧАНИЕ

Вывод утилиты **file** утверждает, что файл имеет формат PostScript, и вы подтвердили это, просмотрев его содержимое. Обратите внимание на строки заголовка информации PostScript. Используйте **Q**, чтобы выйти из команды **less**.

3. Используя **man**, изучите команды, используемые для просмотра и печати файлов PostScript.

- 3.1. Используя команду **man** изучите команды, используемые для просмотра и печати файлов PostScript.

```
[student@workstation ~]# man -k postscript viewer
```

```
evince (1)           - GNOME document viewer
evince-previewer (1) - show a printing preview of PostScript and PDF documents
evince-thumbnailer (1) - create png thumbnails from PostScript and PDF documents
gcm-viewer (1)        - GNOME Color Manager Profile Viewer Tool
gnome-logs (1)        - log viewer for the systemd journal
grops (1)            - PostScript driver for groff
pango-view (1)        - Pango text viewer
pluginviewer (8)      - list loadable SASL plugins and their properties
```



ПРИМЕЧАНИЕ

Использование нескольких слов с параметром **-k** позволяет находить страницы руководства, соответствующие любому слову; те, в описании которых есть «*postscript*» или «*viewer*». Обратите внимание на команды **evince(1)** в выводе.

4. Узнайте, как использовать программу просмотра **evince(1)** в режиме предварительного просмотра. Кроме того, определите, как открыть документ, начиная с определенной страницы.

- 4.1. Используйте команду **man evince**, чтобы узнать, как использовать программу просмотра в режиме предварительного просмотра.



ПРИМЕЧАНИЕ

Параметр **-w** (или **--preview**) открывает просмотр в режиме предварительного просмотра. Параметр **-i** используется для указания начальной страницы.

(Примечание при переводе: Предполагается, что вы работаете в графической оболочке)

5. Просмотрите свой файл **PostScript**, используя различные изученные вами варианты **evince**. Закройте файл документа, когда закончите.

5.1. Используйте команду **evince**, чтобы открыть **/home/student/passwd.ps**

```
[student@workstation ~]$ evince /home/student/passwd.ps
```

5.2. Используйте команду **evince -w /home/student/passwd.ps**, чтобы открыть файл в режиме предварительного просмотра.

```
[student@workstation ~]$ evince -w /home/student/passwd.ps
```

5.3. Используйте команду **evince -i 3 /home/student/passwd.ps**, чтобы открыть файл на странице 3.

```
[student@workstation ~]$ evince -i 3 /home/student/passwd.ps
```



ПРИМЕЧАНИЕ

В то время как обычный режим **evince** позволяет просматривать в полноэкранном режиме и в стиле презентации, режим предварительного просмотра **evince** полезен для быстрого просмотра и печати. Обратите внимание на значок печати вверху.

6. Используйте команду **man**, исследуйте **lp(1)**, чтобы определить, как распечатать любой документ, начиная с определенной страницы. Фактически не вводя никаких команд (потому что нет принтеров), изучите синтаксис в одной команде, чтобы печатать только страницы 2 и 3 вашего файла **PostScript**.

6.1. Используйте **man**, чтобы определить, как печатать определенные страницы документа.

```
[student@serverX ~]$ man lp
```



ПРИМЕЧАНИЕ

Из справочной информации **lp(1)** вы узнаете, что указывает страницы, опция **-P**. Команда **lp** загружается на принтер по умолчанию, отправляя только диапазон страниц, начиная с 2 и заканчивая 3. Таким образом, допустимый ответ будет **lp passwd.ps -P 2-3**.

7. Используя **pinfo**, найдите документацию **GNU Info** о программе просмотра **evince**.

7.1. Используйте команду **pinfo**, чтобы найти документацию **GNU Info** о программе просмотра **evince**.

```
[student@workstation ~]$ pinfo evince
```



ПРИМЕЧАНИЕ

Обратите внимание, что вместо этого отображается **man** справочная страница **evince(1)**. Средство просмотра документов **pinfo** ищет соответствующую страницу руководства, когда для запрошенной темы не существует подходящего узла документации **GNU**. Нажмите **Q**, чтобы выйти.

8. Используя **Firefox**, откройте каталог документации пакета системы и перейдите в подкаталог пакета **man-db**. Просмотрите предоставленные руководства.

8.1. Используйте **firefox /usr/share/doc** для просмотра системной документации. Перейдите в подкаталог **man-db**. Щелкните руководства, чтобы просмотреть их.

```
[student@workstation ~]$ firefox /usr/share/doc
```



ПРИМЕЧАНИЕ

Закладки можно делать для любого часто используемого каталога. Перейдя в каталог **man-db**, щелкните, чтобы открыть и просмотреть текстовую версию руководства, затем закройте ее. Щелкните, чтобы открыть версию PostScript. Как отмечалось ранее, **evince** - это программа для просмотра документов PostScript и PDF по умолчанию. Возможно, вы захотите вернуться к этим документам позже, чтобы больше узнать о **man**. Когда закончите, закройте программу просмотра **evince**.

Index of file:///usr/share/doc/man-db/			
Up to higher level directory			
Name	Size	Last Modified	
File: ChangeLog	51 KB	12/12/16	6:44:30 PM GMT+6
File: NEWS	60 KB	5/11/19	8:56:46 PM GMT+6
File: README	12 KB	12/11/16	5:44:45 AM GMT+6
 man-db-manual.ps	129 KB	5/11/19	8:57:50 PM GMT+6
 man-db-manual.txt	70 KB	5/11/19	8:57:47 PM GMT+6

9. С помощью браузера **Firefox** найдите и перейдите в подкаталог пакета **initscripts**. Просмотрите файл **sysconfig.txt**, в котором описаны важные параметры конфигурации системы, хранящиеся в каталоге **/etc/sysconfig**.

9.1. В браузере **Firefox** найдите подкаталог пакета **initscripts**.

Обратите внимание, насколько полезен браузер для поиска и просмотра документации по локальной системе. По завершении закройте документ и **Firefox**.

Оценка

На рабочей станции запустите **lab help-review grade**, чтобы подтвердить успешность выполнения упражнения.

```
[student@workstation ~]$ lab help-review grade
```

Завершение

На рабочей станции запустите сценарий **lab help-review finish**, чтобы завершить это упражнение.

```
[student@workstation ~]$ lab help-review finish
```

На этом лабораторная работа завершена.

РЕЗЮМЕ

В этой главе вы узнали:

- Страницы руководства просматриваются с помощью команды **man** и предоставляют информацию о компонентах системы Linux, таких как файлы, команды и функции.
- По соглашению, при обращении к странице руководства после названия страницы следует номер ее раздела в круглых скобках.
- Документы GNUInfo просматриваются с помощью команды **pinfo** и состоят из набора гипертекстовых узлов, предоставляющих информацию о программных пакетах в целом.
- Клавиши навигации, используемые **man** и **pinfo**, немного отличаются.

ГЛАВА 5

СОЗДАНИЕ, ПРОСМОТР И РЕДАКТИРОВАНИЕ ТЕКСТОВЫХ ФАЙЛОВ

ЦЕЛЬ

Создавайте, просматривайте и редактируйте текстовые файлы из вывода команд или в текстовом редакторе.

ЗАДАЧИ

- Сохранять выходные данные команд или ошибки в файл с перенаправлением оболочки и обрабатывать выходные данные команд с помощью нескольких программ командной строки с использованием конвейера (pipe).
- Создавать и изменять текстовые файлы с помощью редактора **vim**.
- Используйте переменные оболочки для облегчения выполнения команд и редактируйте сценарии запуска Bash, чтобы установить переменные оболочки и среды для изменения поведения оболочки и программ, запускаемых из оболочки.

РАЗДЕЛЫ

- Перенаправление вывода в файл или программу (и контрольный опрос)
- Редактирование текстовых файлов из командной строки (и упражнения с пошаговыми инструкциями)
- Изменение среды оболочки (и упражнения с пошаговыми инструкциями)

ЛАБОРОТОРНАЯ РАБОТА

Создание, просмотр и редактирование текстовых файлов

ПЕРЕНАПРАВЛЕНИЕ ВЫВОДА В ФАЙЛ ИЛИ ПРОГРАММУ

ЗАДАЧИ

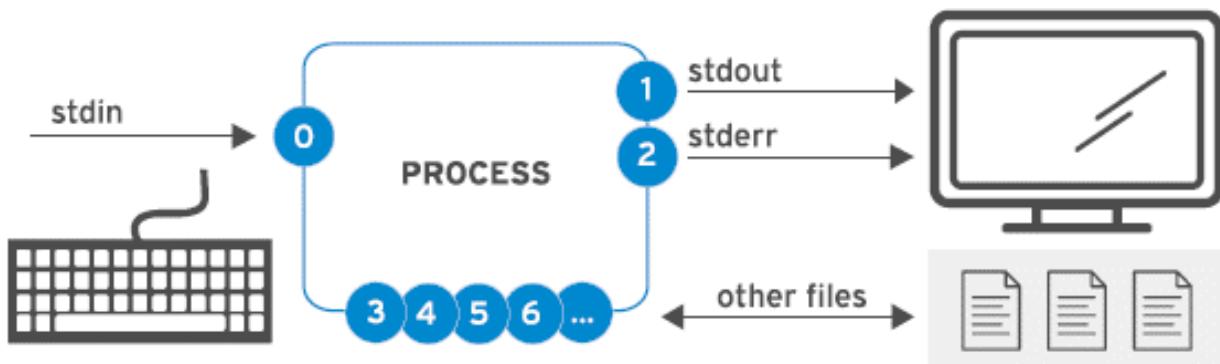
После завершения этого раздела вы сможете сохранять выходные данные или ошибки в файл перенаправлением shell и обрабатывать выходные данные команд с помощью нескольких программ командной строки при помощи переправления потока конвейерами.

СТАНДАРТНЫЙ ВХОД, СТАНДАРТНЫЙ ВЫХОД И СТАНДАРТНЫЙ ВЫВОД ОШИБОК

Запущенной программе или процессу необходимо откуда-то читать ввод и куда-то записывать вывод. Команда, запускаемая из приглашения оболочки, обычно считывает ввод с клавиатуры и отправляет вывод в окно терминала.

Процесс использует пронумерованные каналы, называемые *файловыми дескрипторами*, для получения ввода и отправки вывода. Все процессы начинаются как минимум с трех файловых дескрипторов. *Стандартный ввод (Standard input)* (канал 0) считывает ввод с клавиатуры. *Стандартный вывод (Standard output)* (канал 1) отправляет на терминал обычный вывод. *Стандартная ошибка (Standard error)* (канал 2) отправляет на терминал сообщения об ошибках. Если программа открывает отдельные подключения к другим файлам, она может использовать файловые дескрипторы с более высокими номерами.

Рисунок 5.1: Каналы ввода-вывода процесса (файловые дескрипторы)



Каналы (дескрипторы файлов)

НОМЕР	НАЗВАНИЕ КАНАЛА	ОПИСАНИЕ	СОЕДИНЕНИЕ ПО УМОЛЧАНИЮ	ПРИМЕНЕНИЕ
0	stdin	Стандартный ввод	Клавиатура	только чтение
1	stdout	Стандартный вывод	Терминал	только писать

2	<code>stderr</code>	Стандартный вывод ошибок	Терминал	только писать
3+	<code>filename</code>	Другие файлы	никто	читать и/или писать

ПЕРЕНАПРАВЛЕНИЕ ВЫВОДА В ФАЙЛ

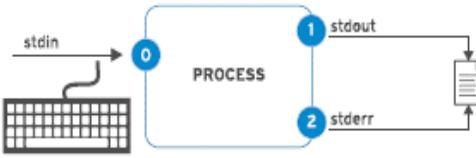
Перенаправление ввода-вывода изменяет способ получения процессом ввода или вывода. Вместо получения ввода с клавиатуры или отправки вывода и ошибок на терминал процесс читает или записывает в файлы. Перенаправление позволяет сохранять сообщения в файл, который обычно отправляется в окно терминала. В качестве альтернативы вы можете использовать перенаправление, чтобы отбросить вывод или ошибки, чтобы они не отображались на терминале и не сохранялись.

Перенаправление `stdout` отбрасывает отображение вывода процесса на терминале. Как видно из следующей таблицы, перенаправление только `stdout` не отклоняет отображение сообщений об ошибках `stderr` на терминале. Если файл не существует, он будет создан. Если файл существует, а перенаправление не является тем, что добавляется к файлу, содержимое файла будет перезаписано.

Если вы хотите отбросить сообщения, в специальный файл `/dev/null` незаметно отбрасывает перенаправленный на него вывод канала и всегда остается пустым файлом.

Операторы перенаправления вывода

ПРИМЕНЕНИЕ	ОБЪЯСНЕНИЕ	НАГЛЯДНОЕ ПОСОБИЕ
<code>>файл</code>	перенаправить <code>stdout</code> для перезаписи файла	
<code>>>файл</code>	перенаправить <code>stdout</code> для добавления в файл	
<code>2>файл</code>	перенаправить <code>stderr</code> для перезаписи файла	
<code>2>/dev/null</code>	Отменить вывод об ошибках <code>stderr</code> , перенаправив на <code>/dev/null</code>	

>файл 2>&1	перенаправить stdout и stderr для перезаписи одного и того же файла	
>>файл 2>&1	перенаправить stdout и stderr для добавления в один и тот же файл	



ВАЖНО

ВАЖНЫЙ

Порядок операций перенаправления важен. Следующая последовательность перенаправляет стандартный вывод в **file**, а затем перенаправляет стандартный вывод ошибок в то же место, что и стандартный вывод (**file**).

```
> file 2>&1
```

Однако следующая последовательность выполняет перенаправление в обратном порядке. Это перенаправляет стандартную ошибку в место по умолчанию для стандартного вывода (окно терминала, поэтому никаких изменений), а затем перенаправляет только стандартный вывод в файл.

```
2>&1 > file
```

Из-за этого некоторые люди предпочитают использовать операторы перенаправления слияния:

```
&>file instead >file 2>&1  
of
```

```
&>>file instead >>file 2>&1 (in Bash 4 / RHEL 6 and later)  
of
```

Однако другие системные администраторы и программисты, которые также используют различные оболочки, связанные с bash (известные как Bourne-совместимые оболочки) для команд сценариев, считают, что следует избегать новых операторов перенаправления слияния, поскольку они не стандартизированы и не реализованы во всех этих оболочках и есть другие ограничения.

Авторы этого курса занимают нейтральную позицию по этой теме, и оба синтаксиса, вероятно, будут встречаться в этой области.

Примеры перенаправления вывода

Многие рутинные задачи администрирования упрощаются за счет перенаправления. Используйте предыдущую таблицу для помощи при рассмотрении следующих примеров:

- Сохраните отметку времени для дальнейшего использования.

```
[user@host ~]$ date > /tmp/saved-timestamp
```

- Скопируйте последние 100 строк из файла журнала в другой файл.

```
[user@host ~]$ tail -n 100 /var/log/dmesg > /tmp/last-100-boot-messages
```

- Объедините четыре файла в один.

```
[user@host ~]$ cat file1 file2 file3 file4 > /tmp/all-four-in-one
```

- Перечислите в файл скрытые и обычные имена файлов домашнего каталога.

```
[user@host ~]$ ls -a > /tmp/my-file-names
```

- Добавить вывод в существующий файл.

```
[user@host ~]$ echo "new line of information" >> /tmp/many-lines-of-information  
[user@host ~]$ diff previous-file current-file >> /tmp/tracking-changes-made
```

- Следующие несколько команд генерируют сообщения об ошибках, потому что некоторые системные каталоги недоступны для обычных пользователей. Наблюдайте, как перенаправляются сообщения об ошибках. Перенаправляйте ошибки в файл при просмотре обычного вывода команд на терминале.

```
[user@host ~]$ find /etc -name passwd 2> /tmp/errors
```

- Сохраняйте выходные данные процесса и сообщения об ошибках в отдельные файлы.

```
[user@host ~]$ find /etc -name passwd > /tmp/output 2> /tmp/errors
```

- Игнорируйте и отбрасывайте сообщения об ошибках.

```
[user@host ~]$ find /etc -name passwd > /tmp/output 2> /dev/null
```

- Сохраните вывод и сгенерированные ошибки вместе.

```
[user@host ~]$ find /etc -name passwd &> /tmp/save-both
```

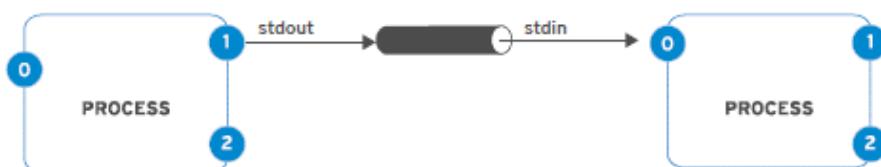
- Добавить вывод и сгенерированные ошибки в существующий файл.

```
[user@host ~]$ find /etc -name passwd >> /tmp/save-both 2>&1
```

КОНСТРУИРОВАНИЕ КОНВЕЕРОВ

Конвейер- это последовательность из одной или нескольких команд, разделенных вертикальной чертой (**pipe**) (|). Канал соединяет стандартный вывод первой команды со стандартным вводом следующей команды.

Рисунок 5.8: Процесс ввода-вывода (I/O) через конвейер (pipe)



Конвейеры позволяют другим процессам управлять и форматировать вывод процесса перед его выводом на терминал. Один полезный мысленный образ - представить, что данные «текут» по конвейеру от одного процесса к другому, слегка изменяясь каждой командой в конвейере, по которому они текут.



ПРИМЕЧАНИЕ

Конвейеры и перенаправление ввода-вывода управляют стандартным выводом и стандартным вводом. Перенаправление отправляет стандартный вывод в файлы или

получает стандартный ввод из файлов. Каналы отправляют стандартный вывод одного процесса на стандартный ввод другого процесса.

Примеры конвейера

В этом примере вывод команды **ls** используется для вывода на терминал одного экрана за раз.

```
[user@host ~]$ ls -l /usr/bin | less
```

Вывод команды **ls** передается по конвейеру в **wc -l**, который подсчитывает количество строк, полученных от **ls**, и выводит его на терминал.

```
[user@host ~]$ ls | wc -l
```

В этом конвейере команда **head** выведет первые 10 строк вывода из **ls -t**, а конечный результат будет перенаправлен в файл.

```
[user@host ~]$ ls -t | head -n 10 > /tmp/ten-last-changed-files
```

Конвейеры, перенаправление и команда **tee**

Когда перенаправление комбинируется с конвейером, оболочка сначала настраивает весь конвейер, а затем перенаправляет ввод/вывод. Если перенаправление вывода используется в середине конвейера, вывод будет идти в файл, а не к следующей команде в конвейере.

В этом примере вывод команды **ls** попадает в файл, а **less** ничего не отображает на терминале.

```
[user@host ~]$ ls > /tmp/saved-output | less
```

Команда **tee** преодолевает данное ограничение. В конвейере **tee** копирует свой стандартный ввод в стандартный вывод, а также перенаправляет свой стандартный вывод в файлы, названные в качестве аргументов команды. Если представить данные как воду, текущую по трубопроводу, действие команды **tee** можно представить, как тройник в трубе, который направляет выход в двух направлениях.

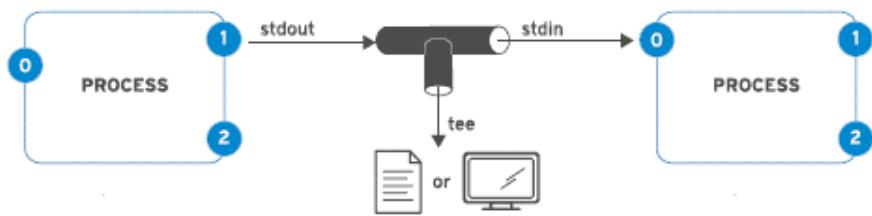


Рисунок 5.9: Процесс ввода/вывода через конвейер с использованием команды tee

Примеры конвейеров с использованием команды **tee** в этом примере вывод команды **ls** перенаправляется в файл и передается в **less** для отображения на терминале по одному экрану за раз.

```
[user@host ~]$ ls -l | tee /tmp/saved-output | less
```

Если команда **tee** используется в конце конвейера, то окончательный вывод команды может быть сохранен и выведен на терминал одновременно.

```
[user@host ~]$ ls -t | head -n 10 | tee /tmp/ten-last-changed-files
```



ВАЖНО

Стандартная ошибка может быть перенаправлена через конвейер, но операторы перенаправления слияния (**&>** и **&>>**) не могут быть использованы для этого.

Ниже приведен правильный способ перенаправления стандартного вывода и стандартной ошибки через конвейер:

```
[user@host ~]$ find -name /passwd 2>&1 | less
```



РЕКОМЕНДАЦИИ

infobash(*Справочное руководство GNU Bash*)

- Раздел 3.2.2: Конвейеры (Pipelines)
- Раздел 3.6: Перенаправления (Redirections)

info coreutils 'tee invocation' (*Руководство GNU coreutils*)

- Раздел 17.1: Перенаправление вывода на несколько файлов или процессов

Справочные страницы **manbash(1)**, **cat(1)**, **head(1)**, **less(1)**, **mail(1)**, **tee(1)**, **tty(1)**, **wc(1)**

КОНТРОЛЬНЫЙ ОПРОС

ПЕРЕНАПРАВЛЕНИЕ ВЫВОДА В ФАЙЛ ИЛИ ПРОГРАММУ

Выберите правильный ответ на следующие вопросы:

1. Какой ответ отображает вывод на терминал и игнорирует все ошибки?

 - a. &>file
 - b. 2>&>file
 - c. 2>/dev/null
 - d. 1>/dev/null

2. Какой ответ отправляет выходные данные в файл и отправляет ошибки в другой файл?

 - a. >file 2>file2
 - b. >file 1>file2
 - c. >file &2>file2
 - d. | tee file

3. Какой ответ отправляет как результат, так и ошибки в файл, создавая его или перезаписывая его.

 - a. | tee file
 - b. 2 &>file
 - c. 1 &>file
 - d. &>file

4. Какой ответ отправляет вывод и ошибки в тот же файл, гарантируя, что существующее содержимое файла

 - a. >file 2>file2
 - b. &>file
 - c. >>file 2>&1
 - d. >>file 1>&1

5. Какой ответ отбрасывает все сообщения, обычно отправляемые на терминал?

 - a. >file 2>file2
 - b. &/>/dev/null
 - c. &/>/dev/null 2>file
 - d. &>file

6. Какой ответ отправляет вывод одновременно на экран и в файл?

 - a. &/>/dev/null
 - b. >file 2>file2
 - c. | tee file
 - d. | < file

7. Какой ответ сохраняет вывод в файл и отбрасывает сообщения об ошибках?

 - a. &>file

- b.** | tee file 2> /dev/null
- c.** > file 1> /dev/null
- d.** > file 2> /dev/null

РЕШЕНИЕ

ПЕРЕНАПРАВЛЕНИЕ ВЫВОДА В ФАЙЛ ИЛИ ПРОГРАММУ

Выберите правильный ответ на следующие вопросы:

1. Какой ответ отображает вывод на терминал и игнорирует все ошибки?
 - a. &>file
 - b. 2>&>file
 - c. **2>/dev/null**
 - d. 1>/dev/null
2. Какой ответ отправляет выходные данные в файл и отправляет ошибки в другой файл?
 - a. >file 2>file2
 - b. >file 1>file2
 - c. >file &2>file2
 - d. | tee file
3. Какой ответ отправляет как результат, так и ошибки в файл, создавая его или перезаписывая его.
 - a. | tee file
 - b. 2 &>file
 - c. 1 &>file
 - d. **&>file**
4. Какой ответ отправляет вывод и ошибки в тот же файл, гарантируя, что существующее содержимое файла
 - a. >file 2>file2
 - b. &>file
 - c. **>>file 2>&1**
 - d. >>file 1>&1
5. Какой ответ отбрасывает все сообщения, обычно отправляемые на терминал?
 - a. >file 2>file2
 - b. **&>/dev/null**
 - c. &>/dev/null 2>file
 - d. &>file
6. Какой ответ отправляет вывод одновременно на экран и в файл?
 - a. &>/dev/null
 - b. >file 2>file2
 - c. | **tee file**
 - d. | < file
7. Какой ответ сохраняет вывод в файл и отбрасывает сообщения об ошибках?

- a.** &>file
- b.** | tee file 2> /dev/null
- c.** > file 1> /dev/null
- d.** > file 2> /dev/null

РЕДАКТИРОВАНИЕ ТЕКСТОВЫХ ФАЙЛОВ С ПОМОЩЬЮ ОБОЛОЧКИ SHELL

ЗАДАЧИ

После завершения этого раздела вы сможете создавать и редактировать текстовые файлы из командной строки с помощью редактора **vim**.

РЕДАКТИРОВАНИЕ ФАЙЛОВ С ПОМОЩЬЮ VIM

Ключевым принципом разработки Linux является то, что информация и параметры конфигурации обычно хранятся в текстовых файлах. Эти файлы могут быть структурированы различными способами: в виде списков настроек, в INI-подобных форматах, в виде структурированного XML или YAML и т. д. Однако преимущество текстовых файлов в том, что их можно просматривать и редактировать с помощью любого простого текстового редактора.

Vim - это улучшенная версия редактора **vi**, распространяемого с системами Linux и UNIX. **Vim** легко настраивается и эффективен для опытных пользователей, включая такие функции, как редактирование разделенного экрана, цветовое форматирование и выделение для редактирования текста.

Зачем изучать Vim?

Вы должны знать, как использовать хотя бы один текстовый редактор, который можно использовать только из текстового приглашения оболочки. Вы можете редактировать текстовые файлы конфигурации из окна терминала или удаленного входа через ssh или веб-консоль. Тогда вам не потребуется доступ к графическому рабочему столу для редактирования файлов на сервере, на самом деле этому серверу может вообще не потребоваться запускать графическую среду рабочего стола.

Но тогда зачем изучать Vim вместо других возможных вариантов? Основная причина в том, что Vim почти всегда устанавливается на сервере, если присутствует текстовый редактор. Это связано с тем, что **vi** был определен стандартом POSIX, которому Linux и многие другие UNIX-подобные операционные системы в значительной степени соответствуют.

Кроме того, Vim часто используется как реализация **vi** в других распространенных операционных системах или дистрибутивах. Например, в настоящее время **macOS** по умолчанию включает облегченную установку Vim. Таким образом, навыки работы с Vim для Linux могут также помочь вам сделать что-то в другом месте.

Запуск Vim

Vim можно установить в RedHatEnterpriseLinux двумя способами. Это может повлиять на доступные вам функции и команды Vim.

На вашем сервере может быть установлен только пакет **vim-minimal**. Это очень легкая установка, которая включает только основной набор функций и базовую команду **vi**. В этом случае вы можете открыть файл для редактирования командой **vi filename**, и вам будут доступны все основные функции, обсуждаемые в этом разделе.

В качестве альтернативы на вашем сервере может быть установлен пакет с расширенным vim (**vim-enhanced**). Это обеспечивает гораздо более полный набор функций, интерактивную

справочную систему и обучающую программу. Чтобы запустить Vim в этом расширенном режиме, вы используете команду **vim**.

```
[user@host ~]$ vim filename
```

В любом случае основные функции, которые мы обсудим в этом разделе, будут работать с обеими командами.



ПРИМЕЧАНИЕ

Если установлена расширенная версия **vim**, у обычных пользователей будет установлен псевдоним оболочки, поэтому при запуске команды **vi** они автоматически получат команду **vim**. Это не относится к пользователям **root** и другим пользователям с **UID** ниже 200 (которые используются системными службами).

Если вы редактируете файлы как пользователь **root** и ожидаете, что **vi** будет работать в расширенном режиме, это может быть сюрпризом. Точно так же, если установлен **vim-enhanced** и обычному пользователю по какой-то причине нужен простой **vi**, ему может потребоваться использовать команду **\vi** для временной отмены действия псевдонима.

Опытные пользователи могут использовать **\vi --version** и **vim --version** для сравнения наборов функций этих двух команд.

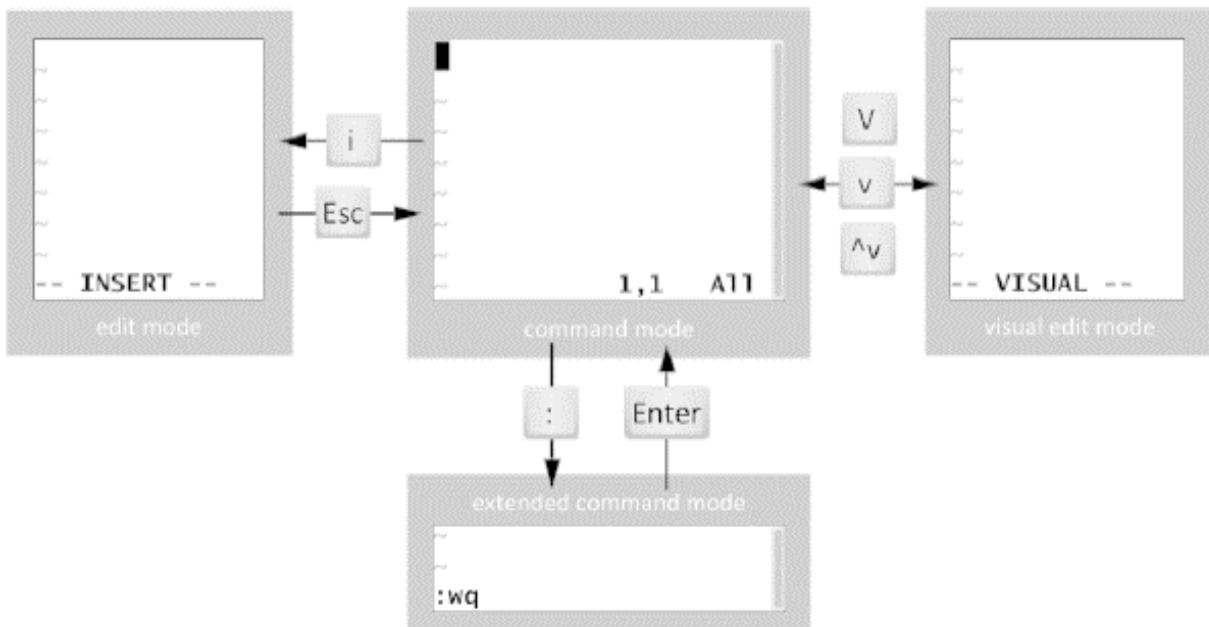
Режимы работы Vim

Необычной характеристикой **Vim** является то, что он имеет несколько режимов работы, включая командный режим, расширенный командный режим, режим редактирования и визуальный режим. В зависимости от режима вы можете вводить команды, редактировать текст или работать с блоками текста. Как новый пользователь **Vim**, вы всегда должны знать о своем текущем режиме, так как нажатия клавиш имеют разный эффект в разных режимах.

Когда вы впервые открываете **Vim**, он запускается в командном режиме, который используется для навигации, вырезания и вставки и других манипуляций с текстом. Переходите между режимами с помощью одно символьных нажатий клавиш, чтобы получить доступ к определенным функциям редактирования:

- Нажатие клавиши **i** переходит в режим вставки, в котором весь набранный текст становится содержимым файла. Нажатие **Esc** возвращает в командный режим.
- Нажатие клавиши **v** переходит в визуальный режим, в котором для работы с текстом можно выбрать несколько символов. Используйте **Shift+V** для многострочного и **Ctrl+V** для выбора блока. Для выхода используется то же нажатие клавиши, что и для входа в визуальный режим (**v**, **Shift+V** или **Ctrl+V**).
- Нажатие клавиши **(:)** запускает расширенный командный режим для таких задач, как запись файла (для его сохранения) и выход из редактора **Vim**.

Рисунок 5.10: Переход между режимами Vim



ПРИМЕЧАНИЕ

Если вы не уверены, в каком режиме находится Vim, вы можете попробовать несколько раз нажать Esc, чтобы вернуться в командный режим. Нажатие Esc в командном режиме безвредно, поэтому допустимо несколько дополнительных нажатий клавиш.

Минимальный, базовый рабочий процесс Vim

В Vim есть эффективные, скоординированные нажатия клавиш для сложных задач редактирования. Хотя возможности Vim считаются полезными, на практике, они могут ошеломить новых пользователей.

Клавиша **i** переводит Vim в режим вставки. Весь текст, введенный после этого, рассматривается как содержимое файла, пока вы не выйдете из данного режима. Клавиша Esc выходит из режима вставки и возвращает Vim в командный режим. Клавиша **u** отменяет самое последнее изменение. Нажмите кнопку **x**, чтобы удалить один символ. Команда:**w** записывает (сохраняет) файл и остается в командном режиме для дальнейшего редактирования. Команда:**wq** записывает (сохраняет) файл и закрывает Vim. Символы в командном режиме **:q!** завершает работу Vim, отменяя все изменения файла с момента последней записи. Пользователь Vim должен выучить эти команды для выполнения любой задачи редактирования.

Изменение существующего текста

В Vim копирование и вставка известны, как «*yank*(вырвать) и *put* (вставить)» с использованием командных символов **Y** и **P**. Начните с размещения курсора на первом выбранном символе, а затем войдите в визуальный режим. Используйте клавиши со стрелками, чтобы расширить визуальный выбор. Когда будете готовы, нажмите **u**, чтобы сохранить выбор в памяти. Поместите курсор в новое место, а затем нажмите **p**, чтобы выделить курсор.

Визуальный режим в Vim

Визуальный режим - отличный способ выделить текст и управлять им. Есть три нажатия клавиш:

- Режим выделения символа: **v**
- Режим выделения линии: **Shift+V**
- Режим выделения блока: **Ctrl+V**

Символьный режим выделяет текст посимвольно. Слово **VISUAL** появится внизу экрана. Нажмите **v**, чтобы войти в режим визуальных символов. **Shift+V** переводит редактор в режим выделения линии. Внизу экрана появится **VISUAL LINE**.

Режим выделения блока идеально подходит для работы с файлами данных. Ставите курсор в начало блока, который хотите выделить, нажмите **Ctrl+V**, чтобы войти в визуальный блок. Внизу экрана появится **VISUAL BLOCK**. Используйте клавиши со стрелками, чтобы выделить раздел, который нужно изменить.



ПРИМЕЧАНИЕ

У Vim много возможностей, но сначала вам следует освоить базовый рабочий процесс. Вам не нужно быстро разбираться во всем редакторе и его возможностях. Освойте основы на практике, а затем вы сможете расширить свой словарный запас Vim, изучив дополнительные команды Vim (нажатия клавиш).

Упражнение для этого раздела познакомит вас с командой **vimtutor**. Это руководство, которое поставляется с расширенным vim, является отличным способом изучить основные функции Vim.



РЕКОМЕНДАЦИИ

Справочная страница man vim(1)

Команда :help в vim (если установлен расширенный пакет vim).

Редактор vim

<http://www.vim.org/>

Начало работы с визуальным режимом Vim

<https://opensource.com/article/19/2/getting-started-vim-visual-mode>

УПРАЖНЕНИЯ С ПОШАГОВЫМИ ИНСТРУКЦИЯМИ

РЕДАКТИРОВАНИЕ ТЕКСТОВЫХ ФАЙЛОВ С ПОМОЩЬЮ ОБОЛОЧКИ SHELL

В этом упражнении вы будете использовать **vimtutor**, чтобы попрактиковаться в основных приемах редактирования в редакторе vim.

РЕЗУЛЬТАТЫ

Вы должны быть способны:

- Редактировать файлы с помощью **Vim**.
- Получите знания в **Vim** с помощью **vimtutor**.

ПРЕЖДЕ ЧЕМ ВЫ НАЧНЕТЕ

Войдите на рабочую станцию как пользователь **student** используя в качестве пароля слово **student**.

На рабочей станции запустите команду **lab edit-vim start**.

```
[student@workstation ~]$ lab edit-vim start
```

1. Используйте команду **ssh** для входа на сервер.

```
[student@workstation ~]$ ssh servera
Activate the web console with: systemctl enable --now cockpit.socket
Last login: Mon Jan 14 15:36:15 2019 from 172.25.250.254
[student@servera ~]$
```

2. Откройте **vimtutor**. Прочтите экран приветствия и выполните *Урок 1.1*.

```
[student@servera ~]$ vimtutor
```

В презентации для навигации используются клавиши со стрелками на клавиатуре. Когда **vi** был впервые разработан, пользователи не могли полагаться на наличие клавиш со стрелками или работающих сопоставлений клавиатуры для клавиш со стрелками для перемещения курсора. Поэтому **vi** изначально был разработан для перемещения курсора с помощью команд с использованием стандартных символьных клавиш, таких как удобно сгруппированные **H**, **J**, **K** и **L**.

Вот один из способов запомнить их:

hangназад, jumpвниз, kickвверх, leapвперед.

3. В окне **vimtutor выполните Урок 1.2.**

Этот урок научит пользователей, как выйти, не сохраняя нежелательных изменений. Все изменения будут потеряны. Иногда это предпочтительнее оставлять критический файл в неправильном состоянии.

4. В окне **vimtutor выполните Урок 1.3.**

В Vim есть быстрые и эффективные нажатия клавиш для удаления точного количества слов, строк, предложений и абзацев. Однако любое задание по редактированию можно выполнить, используя **х** для удаления одного символа.

5. В окне **vimtutor выполните Урок 1.4.**

Для большинства задач редактирования первая нажатая клавиша - это **i**.

6. В окне **vimtutor выполните Урок 1.5.**

В лекции только команда **i** (вставьте **insert**) была обучена как нажатие клавиши для входа в режим редактирования. В этом уроке **vimtutor** демонстрируются другие доступные нажатия клавиш для изменения положения курсора при входе в режим вставки. В режиме вставки весь набранный текст является содержимым файла.

7. В окне **vimtutor выполните Урок 1.6.**

Введите: **wq**, чтобы сохранить файл и выйти из редактора.

8. В окне **vimtutor прочтите Урок 1 Summary**

Команда **vimtutor** включает в себя еще шесть многошаговых уроков. Эти уроки не входят в состав этого курса, но вы можете изучить их самостоятельно, чтобы узнать больше.

9. Выходите из сервера.

```
[student@server ~]$ logout  
Connection to servera closed.  
[student@workstation ~]$
```

Завершение

На рабочей станции запустите скрипт **lab edit-vim finish**, чтобы завершить это упражнение.

```
[student@workstation ~]$ lab edit-vim finish
```

На этом пошаговое упражнение завершено.

ИЗМЕНЕНИЕ СРЕДЫ ОБОЛОЧКИ SHELL

ЗАДАЧИ

После завершения этого раздела вы сможете установить переменные оболочки, которые помогут запускать команды, и отредактировать сценарии запуска Bash, чтобы установить переменные оболочки и среды для изменения поведения оболочки и программ, запускаемых из оболочки.

ИСПОЛЬЗОВАНИЕ ПЕРЕМЕННЫХ ОБОЛОЧКИ

Оболочка Bash позволяет вам устанавливать переменные оболочки, которые можно использовать для выполнения команд или для изменения поведения оболочки. Вы также можете экспортировать переменные оболочки, как переменные среды, которые автоматически копируются в программы, запускаемые из этой оболочки, при их запуске. Вы можете использовать переменные, чтобы упростить выполнение команды с длинным аргументом или применить общие настройки к командам, запускаемым из этой оболочки.

Переменные оболочки уникальны для конкретного сеанса оболочки. Если у вас открыто два окна терминала или два независимых сеанса входа на один и тот же удаленный сервер, вы запускаете две оболочки. Каждая оболочка имеет свой собственный набор значений.

Присвоение значений переменным

Присвойте значение переменной оболочки, используя следующий синтаксис:

VARIABLENAME=value

Имена переменных могут содержать прописные или строчные буквы, цифры и символ подчеркивания (_). Например, следующие команды устанавливают переменные оболочки:

```
[user@host ~]$ COUNT=40
[user@host ~]$ first_name=John
[user@host ~]$ file1=/tmp/abc
[user@host ~]$ _ID=RH123
```

Помните, что это изменение влияет только на оболочку, в которой вы запускаете команду, а не на любые другие оболочки, которые вы можете запускать на этом сервере.

Вы можете использовать команду **set** для вывода списка всех переменных оболочки, которые установлены в данный момент. (в нем также перечислены все функции оболочки, которые вы можете игнорировать.) Этот список достаточно длинный, чтобы вы могли направить вывод в команду **less** используя конвейер, чтобы вы могли просматривать его по одной странице за раз.

```
[user@host ~]$ set | less
BASH=/usr/bin/bash
BASHOPTS=checkwinsize:cmdhist:complete_fullquote:expand_aliases:extglob:extquote:
force_fignore:histappend:interactive_comments:progcomp:promptvars:sourcepath
BASHRCSSOURCED=Y
...output omitted...
```

Получение значений с расширением переменной

Вы можете использовать расширение переменной для ссылки на значение установленной вами переменной. Для этого перед именем переменной поставьте знак доллара (\$). В следующем примере команда **echo** распечатывает оставшуюся часть введенной командной строки, но после выполнения расширения переменной.

Например, следующая команда устанавливает для переменной **COUNT** значение **40**.

```
[user@host ~]$ COUNT=40
```

Если вы введете команду **echo COUNT**, она распечатает строку **COUNT**.

```
[user@host ~]$ echo COUNT
COUNT
```

Но если вы введете команду **echo \$COUNT**, она распечатает значение переменной **COUNT**.

```
[user@host ~]$ echo $COUNT
40
```

Более практическим примером может быть использование переменной для ссылки на длинное имя файла для нескольких команд.

```
[user@host ~]$ file1=/tmp/tmp.z9pXW0HqcC
[user@host ~]$ ls -l $file1
-rw----- 1 student student 1452 Jan 22 14:39 /tmp/tmp.z9pXW0HqcC
[user@host ~]$ rm $file1
[user@host ~]$ ls -l $file1
total 0
```



ВАЖНО

Если рядом с именем переменной есть завершающие символы, вам может потребоваться защитить имя переменной фигурными скобками. Вы всегда можете использовать фигурные скобки при раскрытии переменных, но вы также увидите много примеров, в которых они не нужны и опускаются.

В следующем примере первая команда **echo** пытается расширить несуществующую переменную **COUNTx**, что не вызывает ошибки, а вместо этого ничего не возвращает.

```
[user@host ~]$ echo Repeat $COUNTx  
Repeat  
[user@host ~]$ echo Repeat ${COUNT}x  
Repeat 40x
```

Настройка Bash с помощью переменных оболочки

Некоторые переменные оболочки устанавливаются при запуске **Bash**, но их можно изменить, чтобы настроить поведение оболочки.

Например, две переменные оболочки, которые влияют на историю оболочки и команду **history**, - это **HISTFILE** и **HISTFILESIZE**. Если установлен **HISTFILE**, он указывает расположение файла, в котором будет сохраняться история оболочки при выходе. По умолчанию это пользовательский файл **~/.bash_history**. Переменная **HISTFILESIZE** указывает, сколько команд должно быть сохранено в этом файле из истории команд.

Другой пример - **PS1**, переменная оболочки, которая управляет внешним видом приглашения оболочки. Если вы измените это значение, это изменит внешний вид приглашения вашей оболочки. Ряд расширений специальных символов, поддерживаемых приглашением, перечислены в разделе «**PROMPTING**» на странице руководства **bash(1)**.

```
[user@host ~]$ PS1="bash\$ "  
bash$ PS1="[\u@\h \W]\$ "  
[user@host ~]$
```

Два момента, на которые следует обратить внимание в отношении приведенного выше примера: во-первых, поскольку значение, установленное **PS1**, является подсказкой, практически всегда желательно заканчивать подсказку конечным пробелом. Во-вторых, всякий раз, когда значение переменной содержит какую-либо форму пробела, включая пробел, табуляцию или возврат, значение должно быть заключено в кавычки, одинарные или двойные; это не обязательно. Если кавычки опущены, будут получены неожиданные результаты. Изучите приведенный выше пример **PS1** и обратите внимание, что он соответствует как рекомендации (конечный пробел), так и правилу (кавычки).

КОНФИГУРИРОВАНИЕ ПРОГРАММ С ПОМОЩЬЮ ПЕРЕМЕННЫХ СРЕДЫ

Оболочка предоставляет среду для программ, которые вы запускаете из этой оболочки. Помимо прочего, эта среда включает информацию о текущем рабочем каталоге в файловой системе, параметрах командной строки, переданных программе, и значениях переменных среды. Программы могут использовать эти переменные среды для изменения своего поведения или настроек по умолчанию.

Переменные оболочки, не являющиеся переменными среды, могут использоваться только оболочкой. Переменные среды могут использоваться оболочкой и программами, запускаемыми из этой оболочки.



ПРИМЕЧАНИЕ

HISTFILE, **HISTFILESIZE** и **PS1**, изученные в предыдущем разделе, не нужно экспортировать как переменные среды, потому что они используются только самой оболочкой, а не программами, которые вы запускаете из оболочки.

Вы можете превратить любую переменную, определенную в оболочке, в переменную среды, пометив ее для экспорта с помощью команды **export**.

```
[user@host ~]$ EDITOR=vim  
[user@host ~]$ export EDITOR
```

Вы можете установить и экспортить переменную за один шаг:

```
[user@host ~]$ exportEDITOR=vim
```

Приложения и сеансы используют эти переменные для определения своего поведения. Например, оболочка автоматически устанавливает переменную **HOME** в имя файла домашнего каталога пользователя при запуске. Это можно использовать, чтобы помочь программам определить, где сохранять файлы.

Другой пример - переменная **LANG**, который устанавливает локаль. Это регулирует предпочтительный язык для вывода программы; набор символов; форматирование дат, чисел и валюты; и порядок сортировки программ. Если установлено значение **en_US.UTF-8**, языковой стандарт будет использовать английский (США) с кодировкой символов **UTF-8 Unicode**. Если установлено другое значение, например, **fr_FR.UTF-8**, будет использоваться французская кодировка **Unicode UTF-8**.

```
[user@host ~]$ date  
Tue Jan 22 16:37:45 CST 2019
```

```
[user@host ~]$ export LANG=fr_FR.UTF-8  
[user@host ~]$ date  
mar. janv. 22 16:38:14 CST 2019
```

Еще одна важная переменная среды - **PATH**. Переменная **PATH** содержит список разделенных двоеточиями каталогов, содержащих программы:

```
[user@host ~]$ echo $PATH  
/home/user/.local/bin:/home/user/bin:/usr/share/Modules/bin:/usr/local/bin:/usr/  
bin:/usr/local/sbin:/usr/sbin
```

Когда вы запускаете такую команду, как **ls**, оболочка ищет исполняемый файл **ls** в каждом из этих каталогов по порядку и запускает первый найденный соответствующий файл. (В типичной системе это **/usr/bin/ls**.)

Вы можете легко добавить дополнительные каталоги в конец **PATH**. Например, возможно, у вас есть исполняемые программы или сценарии, которые вы хотите запускать как обычные команды в **/home/user/sbin**. Вы можете добавить **/home/user/sbin** в конец **PATH** для текущего сеанса следующим образом:

```
[user@host ~]$ export PATH=${PATH}:/home/user/sbin
```

Чтобы вывести список всех переменных среды для конкретной оболочки, выполните команду **env**:

```
[user@host ~]$ env  
...output omitted...  
LANG=en_US.UTF-8  
HISTCONTROL=ignoredups  
HOSTNAME=host.example.com  
XDG_SESSION_ID=4  
...output omitted...
```

Настройка текстового редактора по умолчанию

Переменная среды **EDITOR** определяет программу, которую вы хотите использовать в качестве текстового редактора по умолчанию для программ командной строки. Многие программы используют **vi** или **vim**, если они не указаны, но при необходимости вы можете изменить это предпочтение:

```
[user@host ~]$ export EDITOR=nano
```



ВАЖНО

По соглашению, переменные среды и переменные оболочки, которые автоматически устанавливаются оболочкой, имеют имена, в которых используются все символы верхнего регистра. Если вы устанавливаете свои собственные переменные, вы можете использовать имена, состоящие из символов нижнего регистра, чтобы избежать конфликтов имен.

УСТАНОВКА ПЕРЕМЕННЫХ АВТОМАТИЧЕСКИ

Если вы хотите автоматически устанавливать переменные оболочки или среды при запуске оболочки shell, вы можете отредактировать сценарии запуска Bash. При запуске Bash запускаются несколько текстовых файлов, содержащих команды оболочки, которые инициализируют её среду.

Точные запускаемые сценарии зависят от того, как была запущена оболочка, будь то интерактивная оболочка входа в систему, интерактивная оболочка без входа в систему или сценарий оболочки.

Предполагая, что используются файлы по умолчанию **/etc/profile**, **/etc/bashrc** и **~/.bash_profile**, если вы хотите внести изменения в свою учетную запись, которые влияют на все запросы интерактивной оболочки при запуске, отредактируйте файл **~/.bashrc**. Например, вы можете установить редактор **nano** по умолчанию для этой учетной записи, отредактировав файл следующим образом:

```
# .bashrc

# Source global definitions
if [ -f /etc/bashrc ]; then
    . /etc/bashrc
fi

# User specific environment
PATH="$HOME/.local/bin:$HOME/bin:$PATH"
export PATH

# User specific aliases and functions
export EDITOR=nano
```



ПРИМЕЧАНИЕ

Лучший способ настроить параметры, влияющие на все учетные записи пользователей, - это добавить файл с именем, заканчивающимся на **.sh**, содержащий изменения, в каталог **/etc/profile.d**. Для этого вам необходимо войти в систему как пользователь **root**.

ОТМЕНА УСТАНОВКИ И НЕЭКСПОРТА ПЕРЕМЕННЫХ

Чтобы полностью отключить переменную и не экспортовать ее, используйте команду **unset**:

```
[user@host ~]$ echo $file1  
/tmp/tmp.z9pXW0HqcC  
[user@host ~]$ unset file1  
[user@host ~]$ echo $file1  
  
[user@host ~]$
```

Чтобы вывести переменную из экспорта, не снимая ее настройки, используйте команду **export -n**:

```
[user@host ~]$ export -n PS1
```



РЕКОМЕНДАЦИИ

Справочные страницы **manbash(1)**, **env(1)**, и **builtins(1)**

УПРАЖНЕНИЯ С ПОШАГОВЫМИ ИНСТРУКЦИЯМИ

ИЗМЕНЕНИЕ ПЕРЕМЕННЫХ ОБОЛОЧКИ

В этом упражнении вы будете использовать переменные оболочки и расширение переменных для выполнения команд и установить переменную среды для настройки редактора по умолчанию для новых оболочек.

РЕЗУЛЬТАТЫ:

Вы должны быть способны:

- Редактировать профиль пользователя.
- Создавать переменные оболочки shell.
- Создавать переменные среды.

ПРЕЖДЕ ЧЕМ ВЫ НАЧНЕТЕ

Войдите на рабочую станцию как пользователь **student**, используя в качестве пароля слово **student**.

На рабочей станции запустите команду **lab edit-shell start**.

```
[student@workstation ~]$ lab edit-shell start
```

1. Измените переменную оболочки PS1 пользователя ученика на **[\u@\h \t \w]\$** (не забудьте заключить значение **PS1** в кавычки и поставить конечный пробел после знака доллара). Это добавит время печати приглашения к каждому приглашению оболочки.

1.1. На рабочей станции используйте команду **ssh** для входа на сервер.

```
[student@workstation ~]$ ssh servera
Last login: Wed Jan 23 11:25:05 2019 from 172.25.250.250
[student@servera ~]$
```

1.2. Используйте Vim для редактирования файла конфигурации **~/.bashrc**.

```
[student@servera ~]$ vim ~/.bashrc
```

- 1.3. Добавьте переменную оболочки PS1 и ее значение в файл **~/.bashrc**. Не забудьте добавить конечный пробел в конце установленного вами значения и заключить все значение в кавычки, включая конечный пробел.

```
...output omitted...
# User specific environment and startup programs
PATH=$PATH:$HOME/.local/bin:$HOME/bin
PS1='[\u@\h \t \w]$ '
export PATH
```

- 1.4. Выйдите из сервера и войдите снова, используя команду **ssh**, чтобы обновить приглашение в командной строке.

```
[student@servera ~]$ exit
logout
Connection to servera closed.
[student@workstation ~]$ ssh servera
Last login: Wed Jan 23 14:42:15 2019 from 172.25.250.254
[student@servera 14:45:05 ~]$
```

2. Присвойте значение локальной переменной оболочки. Имена переменных могут содержать прописные или строчные буквы, цифры и символ подчеркивания. Получить значение переменной.

- 2.1.** Создайте новую переменную с именем **file** со значением **tmp.zdkei083**.

```
[student@servera 14:45:05 ~]$ file=tmp.zdkei083
```

- 2.2.** Получить значение файловой переменной.

```
[student@servera 14:45:35 ~]$ echo $file
tmp.zdkei083
```

- 2.3.** Используйте файл с именем переменной и команду **ls -l**, чтобы просмотреть файл **tmp.zdkei083**. Используйте команду **rm** и имя переменной файла, чтобы удалить файл **tmp.zdkei083**.

Подтвердите, что он был удален.

```
[student@servera 14:46:07 ~]$ ls -l $file
-rw-rw-r--. 1 student student 0 Jan 23 14:59 tmp.zdkei083
[student@servera 14:46:10 ~]$ rm $file
[student@servera 14:46:15 ~]$ ls -l $file
ls: cannot access 'tmp.zdkei083': No such file or directory
```

3. Присвойте значение переменной редактора. Используйте одну команду, чтобы сделать переменную переменной среды.

```
[student@servera 14:46:40 ~]$ export EDITOR=vim  
[student@servera 14:46:55 ~]$ echo $EDITOR  
Vim
```

4. Выйдите из сервера.

```
[student@servera 14:47:11 ~]$ logout  
logout  
Connection to servera closed.  
[student@workstation ~]$
```

ЗАВЕРШЕНИЕ

На рабочей станции запустите сценарий **lab edit-shell finish**, чтобы завершить это упражнение.

```
[student@workstation ~]$ lab edit-shell finish
```

На этом упражнения с пошаговыми инструкциями завершено.

ЛАБОРАТОРНАЯ РАБОТА

СОЗДАНИЕ, ПРОСМОТР И РЕДАКТИРОВАНИЕ ТЕКСТОВЫХ ФАЙЛОВ

КОНТРОЛЬНЫЙ СПИСОК РАБОТЫ

В этой лабораторной работе вы отредактируете текстовый файл с помощью редактора **vim**.

РЕЗУЛЬТАТЫ

Вы должны быть способны:

- Используйте **Vim** для редактирования файлов.
- Используйте визуальный режим, чтобы упростить редактирование файлов.

ПРЕЖДЕ ЧЕМ ВЫ НАЧНЕТЕ

Войдите на рабочую станцию как пользователь **student**, используя в качестве пароля слово **student**.

На рабочей станции запустите команду **lab edit-review start**

```
[student@workstation ~]$ lab edit-review start
```

1. Перенаправьте длинный список всего содержимого в домашнем каталоге учащегося, включая скрытые каталоги и файлы, в файл с именем **edit_final_lab.txt**.
2. Отредактируйте файл с помощью редактора **Vim**.
3. Удалите первые три строки. Войдите в линейный визуальный режим с большой буквой **V**.
4. Удалите столбцы в первой строке. Войдите в визуальный режим с нижним регистром **v**. Нижний регистр **v** выбирает символы только в одной строке. Столбцы после **-rw-** следует удалить.
5. Удалите столбцы и последующие точки («.») В оставшихся строках. Используйте режим визуального блока. Войдите в визуальный блок с помощью управляющей последовательности **Ctrl+V**. Используйте эту последовательность клавиш для выбора блока символов в нескольких строках. Столбцы после **-rw-** следует удалить.
6. Используйте режим визуальной блокировки, чтобы удалить четвертый столбец.
7. Используйте режим визуальной блокировки, чтобы удалить столбец времени, оставив месяц и день во всех строках.
8. Удалите строки **Desktop** и **Public**. Войдите в режим визуальной линии с прописной буквой **V**.
9. Используйте команду: **wq** для сохранения и выхода из файла. Сделайте резервную копию, используя дату (в секундах) для создания уникального имени файла.
10. Добавьте в файл пунктирную линию. Пунктирная линия должна содержать не менее 12 тире.

11. Добавьте список каталогов в каталог **Documents**. Выведите список каталогов на терминале и отправьте его в файл **edit_final_lab.txt** с помощью одной командной строки.
12. Убедитесь, что список каталогов находится в нижней части лабораторного файла.

Оценка

На рабочей станции запустите команду **lab edit-review grade**, чтобы подтвердить успех этого упражнения.

```
[student@workstation ~]$ lab edit-review grade
```

Завершение

На рабочей станции запустите сценарий **lab edit-review finish**, чтобы завершить упражнение.

```
[student@workstation ~]$ lab edit-review finish
```

На этом лабораторная работа завершена.

РЕШЕНИЕ

СОЗДАНИЕ, ПРОСМОТР И РЕДАКТИРОВАНИЕ ТЕКСТОВЫХ ФАЙЛОВ

КОНТРОЛЬНЫЙ СПИСОК РАБОТЫ

В этой лабораторной работе вы отредактируете текстовый файл с помощью редактора **vim**.

РЕЗУЛЬТАТЫ

Вы должны быть способны:

- Используйте **Vim** для редактирования файлов.
- Используйте визуальный режим, чтобы упростить редактирование файлов.

ПРЕЖДЕ ЧЕМ ВЫ НАЧНЕТЕ

Войдите на рабочую станцию как пользователь **student**, используя в качестве пароля слово **student**.

На рабочей станции запустите команду **lab edit-review start**

```
[student@workstation ~]$ lab edit-review start
```

1. Перенаправьте длинный список всего содержимого в домашнем каталоге учащегося, включая скрытые каталоги и файлы, в файл с именем **edit_final_lab.txt**.



ПРИМЕЧАНИЕ

Результат может не точно соответствовать показанным примерам.

На рабочей станции используйте команду **cd**, чтобы вернуться в домашний каталог **student**. Используйте команду **ls -al**, чтобы перенаправить длинный список всего содержимого в файл с именем **edit_final_lab.txt**.

```
[student@workstation ~]$ cd  
[student@workstation ~]$ ls -al > editing_final_lab.txt
```

2. Отредактируйте файл с помощью редактора **Vim**.

```
[student@workstation ~]$ vim editing_final_lab.txt
```

3. Удалите первые три строки. Войдите в линейный визуальный режим с большой буквой **V**.

С помощью клавиш со стрелками поместите курсор на первый символ в первой строке. Войдите в линейный визуальный режим с помощью **Shift+V**. Переместитесь вниз, используя клавишу со стрелкой вниз дважды, чтобы выбрать первые три строки. Удалите строки используя символ **x**.

```
student@workstation:~  
File Edit View Search Terminal Help  
total 32  
drwx----- 15 student student 4096 Jan 22 10:31 .  
drwxr-xr-x 3 root root 21 Jan 9 18:57 ..  
-rw----- 1 student student 318 Jan 21 15:44 .bash_history  
-rw-r--r-- 1 student student 18 Oct 12 23:56 .bash_logout  
-rw-r--r-- 1 student student 141 Oct 12 23:56 .bash_profile  
-rw-r--r-- 1 student student 312 Oct 12 23:56 .bashrc  
drwxr-xr-x 10 student student 203 Jan 14 14:16 .config  
drwxr-xr-x 2 student student 6 Jan 14 14:16 Desktop  
drwxr-xr-x 2 student student 6 Jan 14 14:16 Documents  
drwxr-xr-x 2 student student 6 Jan 14 14:16 Downloads  
-rw-rw-r-- 1 student student 0 Jan 22 10:45 editing_final_lab.txt  
-rw----- 1 student student 16 Jan 14 14:15 .esd_auth  
-rw----- 1 student student 318 Jan 14 14:16 .ICEauthority  
drwx----- 3 student student 19 Jan 14 14:16 .local  
drwxr-xr-x 2 student student 6 Jan 14 14:16 Music  
drwxr-xr-x 2 student student 6 Jan 14 14:16 Pictures  
drwxrw---- 3 student student 19 Jan 14 14:16 .pki  
drwxr-xr-x 2 student student 6 Jan 14 14:16 Public  
drwx----- 2 student student 73 Jan 14 15:35 .ssh  
drwxr-xr-x 2 student student 6 Jan 14 14:16 Templates  
drwxr-xr-x 2 student student 6 Jan 14 14:16 Videos  
-rw----- 1 student student 1095 Jan 14 15:34 .viminfo  
~  
-- VISUAL LINE -- 3 3,1 All
```

4. Удалите столбцы в первой строке. Войдите в визуальный режим с нижним регистром **v**. Нижний регистр **v** выбирает символы только в одной строке. Столбцы после **-rw-** следует удалить.

С помощью клавиш со стрелками установите курсор на первый символ. Войдите в визуальный режим, используя строчные буквы **v**. С помощью клавиш со стрелками установите курсор на последний символ. Удалите выделение с помощью **x**

```
student@workstation:~  
File Edit View Search Terminal Help  
-rw-----| 1 student student 310 Jan 21 15:44 .bash_history  
-rw-r--r--| 1 student student 18 Oct 12 23:56 .bash_logout  
-rw-r--r--| 1 student student 141 Oct 12 23:56 .bash_profile  
-rw-r--r--| 1 student student 312 Oct 12 23:56 .bashrc  
drwx-----| 8 student student 201 Jan 14 14:16 .cache  
drwxr-xr-x.| 10 student student 203 Jan 14 14:16 .config  
drwxr-xr-x.| 2 student student 6 Jan 14 14:16 Desktop  
drwxr-xr-x.| 2 student student 6 Jan 14 14:16 Documents  
drwxr-xr-x.| 2 student student 6 Jan 14 14:16 Downloads  
-rw-rw-r--| 1 student student 0 Jan 22 18:45 editing_final_lab.txt  
-rw-----| 1 student student 16 Jan 14 14:15 .esd_auth  
-rw-----| 1 student student 310 Jan 14 14:16 .ICEauthority  
drwx-----| 3 student student 19 Jan 14 14:16 .local  
drwxr-xr-x.| 2 student student 6 Jan 14 14:16 Music  
drwxr-xr-x.| 2 student student 6 Jan 14 14:16 Pictures  
drwxrwx---| 3 student student 19 Jan 14 14:16 .pkic  
drwxr-xr-x.| 2 student student 6 Jan 14 14:16 Public  
drwx-----| 2 student student 73 Jan 14 15:35 .ssh  
drwxr-xr-x.| 2 student student 6 Jan 14 14:16 Templates  
drwxr-xr-x.| 2 student student 6 Jan 14 14:16 Videos  
-rw-----| 1 student student 1095 Jan 14 15:34 .viminfo  
-  
~  
~  
~  
~  
-- VISUAL -- 7 1,11 All
```

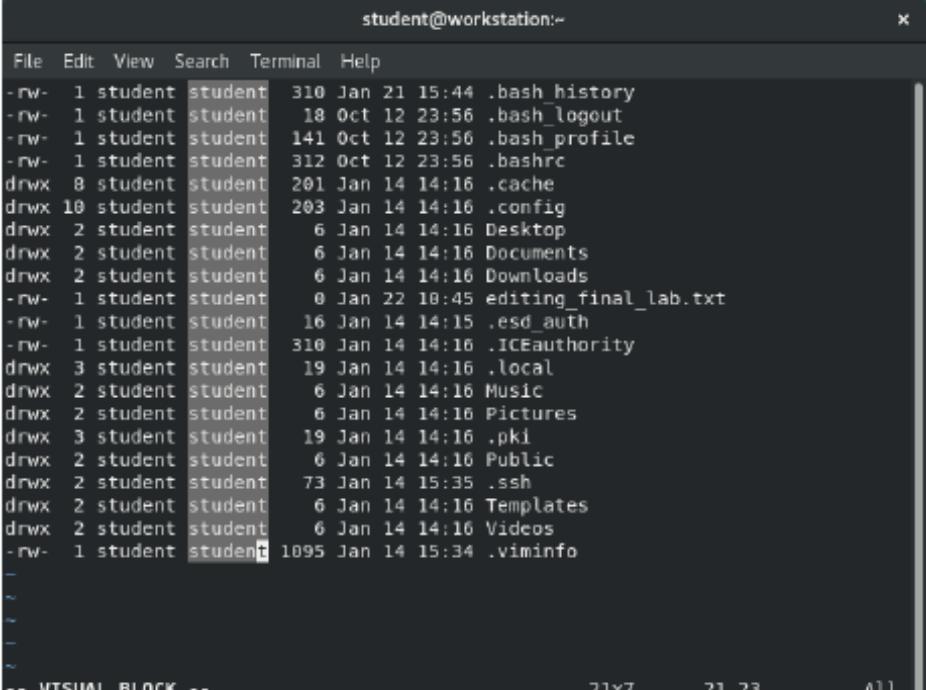
5. Удалите столбцы и последние точки («.») В оставшихся строках. Используйте режим визуального блока. Войдите в визуальный блок с помощью управляющей последовательности **Ctrl+V**. Используйте эту последовательность клавиш для выбора блока символов в нескольких строках. Столбцы после **-rw-** следует удалить.

С помощью клавиш со стрелками установите курсор на первый символ. Войдите в визуальный режим, используя управляющую последовательность **Ctrl+V**. С помощью клавиш со стрелками установите курсор на последний символ столбца в последней строке. Удалите выделение с помощью **X**.

```
student@workstation:~  
File Edit View Search Terminal Help  
-rw-----| 1 student student 310 Jan 21 15:44 .bash_history  
-rw-r--r--| 1 student student 18 Oct 12 23:56 .bash_logout  
-rw-r--r--| 1 student student 141 Oct 12 23:56 .bash_profile  
-rw-r--r--| 1 student student 312 Oct 12 23:56 .bashrc  
drwx-----| 8 student student 201 Jan 14 14:16 .cache  
drwxr-xr-x.| 10 student student 203 Jan 14 14:16 .config  
drwxr-xr-x.| 2 student student 6 Jan 14 14:16 Desktop  
drwxr-xr-x.| 2 student student 6 Jan 14 14:16 Documents  
drwxr-xr-x.| 2 student student 6 Jan 14 14:16 Downloads  
-rw-rw-r--| 1 student student 0 Jan 22 18:45 editing_final_lab.txt  
-rw-----| 1 student student 16 Jan 14 14:15 .esd_auth  
-rw-----| 1 student student 310 Jan 14 14:16 .ICEauthority  
drwx-----| 3 student student 19 Jan 14 14:16 .local  
drwxr-xr-x.| 2 student student 6 Jan 14 14:16 Music  
drwxr-xr-x.| 2 student student 6 Jan 14 14:16 Pictures  
drwxrwx---| 3 student student 19 Jan 14 14:16 .pkic  
drwxr-xr-x.| 2 student student 6 Jan 14 14:16 Public  
drwx-----| 2 student student 73 Jan 14 15:35 .ssh  
drwxr-xr-x.| 2 student student 6 Jan 14 14:16 Templates  
drwxr-xr-x.| 2 student student 6 Jan 14 14:16 Videos  
-rw-----| 1 student student 1095 Jan 14 15:34 .viminfo  
-  
~  
~  
~  
~  
-- VISUAL BLOCK -- 20x7 21,11 All
```

6. Используйте визуальный режим выделения блока, чтобы удалить четвертый столбец.

С помощью клавиш со стрелками установите курсор на первый символ. Войдите в режим визуального блока, используя **Ctrl+V**. С помощью клавиш со стрелками установите курсор на последний символ и строку столбца времени. Удалите выделение с помощью **x**.



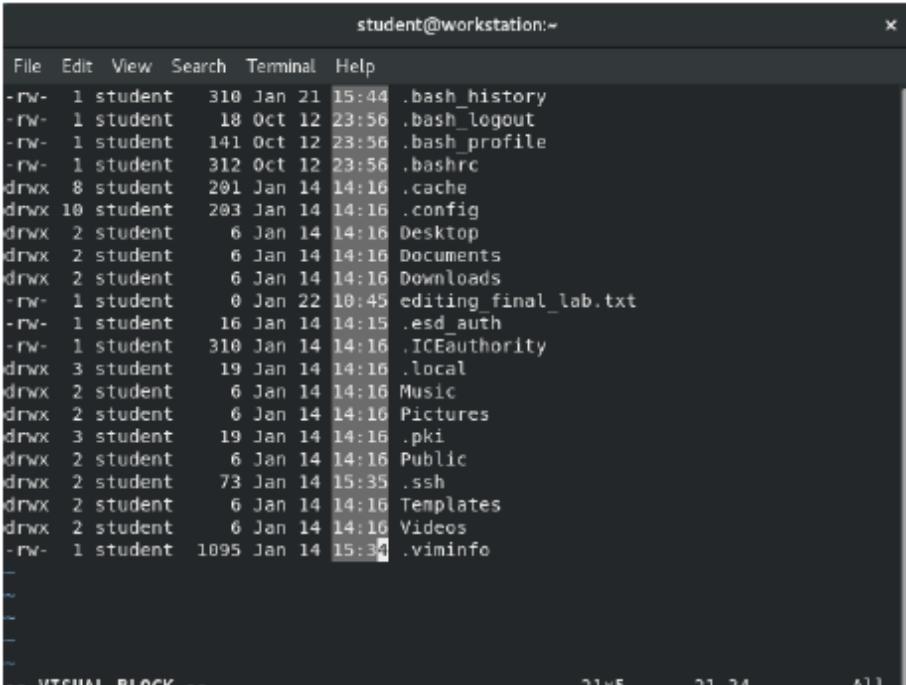
```
student@workstation:~
```

```
File Edit View Search Terminal Help
-rw- 1 student student 310 Jan 21 15:44 .bash_history
-rw- 1 student student 18 Oct 12 23:56 .bash_logout
-rw- 1 student student 141 Oct 12 23:56 .bash_profile
-rw- 1 student student 312 Oct 12 23:56 .bashrc
drwx 8 student student 201 Jan 14 14:16 .cache
drwx 10 student student 203 Jan 14 14:16 .config
drwx 2 student student 6 Jan 14 14:16 Desktop
drwx 2 student student 6 Jan 14 14:16 Documents
drwx 2 student student 6 Jan 14 14:16 Downloads
-rw- 1 student student 0 Jan 22 10:45 editing_final_lab.txt
-rw- 1 student student 16 Jan 14 14:15 .esd_auth
-rw- 1 student student 310 Jan 14 14:16 .ICEauthority
drwx 3 student student 19 Jan 14 14:16 .local
drwx 2 student student 6 Jan 14 14:16 Music
drwx 2 student student 6 Jan 14 14:16 Pictures
drwx 3 student student 19 Jan 14 14:16 .pki
drwx 2 student student 6 Jan 14 14:16 Public
drwx 2 student student 73 Jan 14 15:35 .ssh
drwx 2 student student 6 Jan 14 14:16 Templates
drwx 2 student student 6 Jan 14 14:16 Videos
-rw- 1 student student 1095 Jan 14 15:34 .viminfo
-
~
~
~
~
-- VISIBLE BLOCK --
```

21x7 21,23 All

7. Используйте режим визуальной блокировки, чтобы удалить столбец времени, оставив месяц и день во всех строках.

С помощью клавиш со стрелками установите курсор на первый символ. Войдите в режим визуального блока, используя **Ctrl + V**. С помощью клавиш со стрелками установите курсор на последний символ и строку столбца времени. Удалите выделение с помощью **x**.



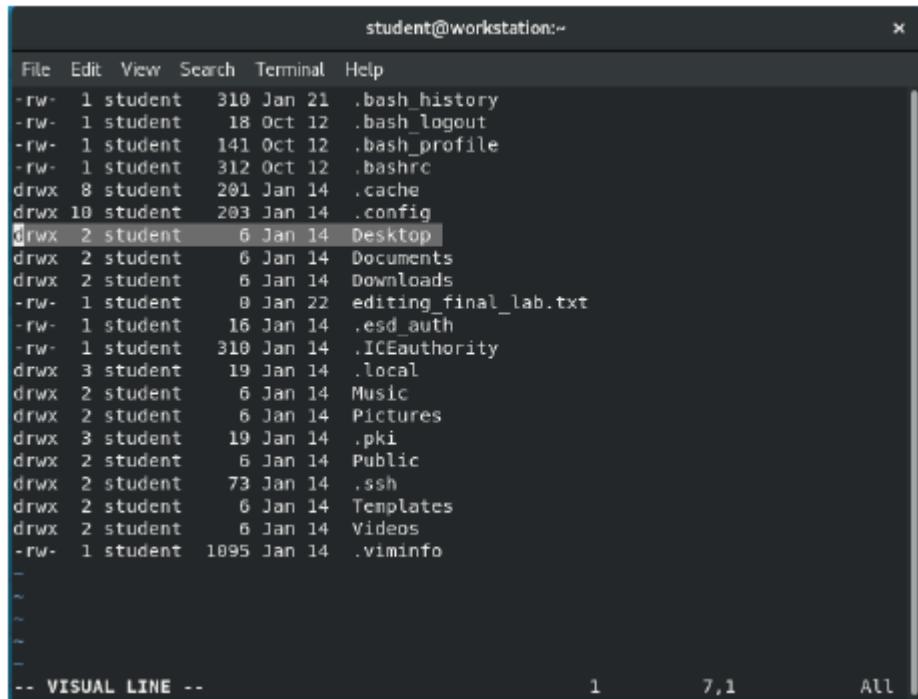
```
student@workstation:~
```

```
File Edit View Search Terminal Help
-rw- 1 student 310 Jan 21 15:44 .bash_history
-rw- 1 student 18 Oct 12 23:56 .bash_logout
-rw- 1 student 141 Oct 12 23:56 .bash_profile
-rw- 1 student 312 Oct 12 23:56 .bashrc
drwx 8 student 201 Jan 14 14:16 .cache
drwx 10 student 203 Jan 14 14:16 .config
drwx 2 student 6 Jan 14 14:16 Desktop
drwx 2 student 6 Jan 14 14:16 Documents
drwx 2 student 6 Jan 14 14:16 Downloads
-rw- 1 student 0 Jan 22 10:45 editing_final_lab.txt
-rw- 1 student 16 Jan 14 14:15 .esd_auth
-rw- 1 student 310 Jan 14 14:16 .ICEauthority
drwx 3 student 19 Jan 14 14:16 .local
drwx 2 student 6 Jan 14 14:16 Music
drwx 2 student 6 Jan 14 14:16 Pictures
drwx 3 student 19 Jan 14 14:16 .pki
drwx 2 student 6 Jan 14 14:16 Public
drwx 2 student 73 Jan 14 15:35 .ssh
drwx 2 student 6 Jan 14 14:16 Templates
drwx 2 student 6 Jan 14 14:16 Videos
-rw- 1 student 1095 Jan 14 15:34 .viminfo
-
~
~
~
~
-- VISIBLE BLOCK --
```

21x5 21,34 All

8. Удалите строки **Desktop** и **Public**. Войдите в режим визуальной линии с прописной буквой **V**.

С помощью клавиш со стрелками поместите курсор на любой символ в **Desktop**. Войдите в визуальный режим с большой буквой **V**. Выделена вся строка. Удалите выделение с помощью **x**. Повторите для строки **Public**.



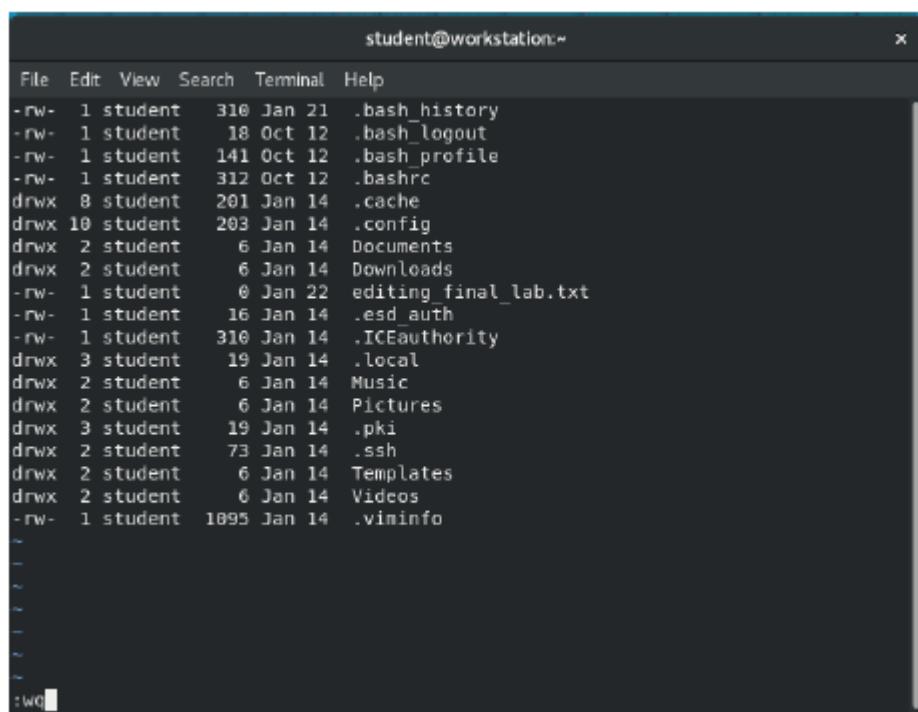
```
student@workstation:~
```

```
File Edit View Search Terminal Help
```

```
-rw- 1 student 310 Jan 21 .bash_history
-rw- 1 student 18 Oct 12 .bash_logout
-rw- 1 student 141 Oct 12 .bash_profile
-rw- 1 student 312 Oct 12 .bashrc
drwx 8 student 201 Jan 14 .cache
drwx 10 student 203 Jan 14 .config
drwx 2 student 6 Jan 14 Desktop
drwx 2 student 6 Jan 14 Documents
drwx 2 student 6 Jan 14 Downloads
-rw- 1 student 0 Jan 22 editing_final_lab.txt
-rw- 1 student 16 Jan 14 .esd_auth
-rw- 1 student 310 Jan 14 .ICEauthority
drwx 3 student 19 Jan 14 .local
drwx 2 student 6 Jan 14 Music
drwx 2 student 6 Jan 14 Pictures
drwx 3 student 19 Jan 14 .pki
drwx 2 student 6 Jan 14 Public
drwx 2 student 73 Jan 14 .ssh
drwx 2 student 6 Jan 14 Templates
drwx 2 student 6 Jan 14 Videos
-rw- 1 student 1095 Jan 14 .viminfo
-
~
~
~
-
-- VISUAL LINE --
```

```
1 7,1 All
```

9. Используйте команду :wq для сохранения и выхода из файла. Сделайте резервную копию, используя дату (в секундах) для создания уникального имени файла.



```
student@workstation:~
```

```
File Edit View Search Terminal Help
```

```
-rw- 1 student 310 Jan 21 .bash_history
-rw- 1 student 18 Oct 12 .bash_logout
-rw- 1 student 141 Oct 12 .bash_profile
-rw- 1 student 312 Oct 12 .bashrc
drwx 8 student 201 Jan 14 .cache
drwx 10 student 203 Jan 14 .config
drwx 2 student 6 Jan 14 Desktop
drwx 2 student 6 Jan 14 Documents
drwx 2 student 6 Jan 14 Downloads
-rw- 1 student 0 Jan 22 editing_final_lab.txt
-rw- 1 student 16 Jan 14 .esd_auth
-rw- 1 student 310 Jan 14 .ICEauthority
drwx 3 student 19 Jan 14 .local
drwx 2 student 6 Jan 14 Music
drwx 2 student 6 Jan 14 Pictures
drwx 3 student 19 Jan 14 .pki
drwx 2 student 73 Jan 14 .ssh
drwx 2 student 6 Jan 14 Templates
drwx 2 student 6 Jan 14 Videos
-rw- 1 student 1095 Jan 14 .viminfo
-
~
~
~
-
-
-
-
-
-
:wq
```

```
[student@workstation ~]$ cp editing_final_lab.txt \
editing_final_lab_$(date +%s).txt
```

10. Добавьте в файл пунктирную линию. Пунктирная линия должна содержать не менее 12 тире.

```
[student@workstation ~]$ echo "-----" \
>> editing_final_lab.txt
```

11. Добавьте список каталогов в каталог **Documents**. Выведите список каталогов на терминале и отправьте его в файл **edit_final_lab.txt** с помощью одной командной строки.

```
[student@workstation ~]$ ls Documents/ | tee -a editing_final_lab.txt
lab_review.txt
```

12. Убедитесь, что список каталогов находится в нижней части лабораторного файла.

```
[student@workstation ~]$ cat editing_final_lab.txt
-rw- 1 student 310 Jan 21 .bash_history
-rw- 1 student 18 Oct 12 .bash_logout
-rw- 1 student 141 Oct 12 .bash_profile
-rw- 1 student 312 Oct 12 .bashrc
drwx 8 student 201 Jan 14 .cache
drwx 10 student 203 Jan 14 .config
drwx 2 student 6 Jan 14 Documents
drwx 2 student 6 Jan 14 Downloads
-rw- 1 student 0 Jan 22 editing_final_lab.txt
-rw- 1 student 16 Jan 14 .esd_auth
-rw- 1 student 310 Jan 14 .ICEauthority
drwx 3 student 19 Jan 14 .local
drwx 2 student 6 Jan 14 Music
drwx 2 student 6 Jan 14 Pictures
drwx 3 student 19 Jan 14 .pki
drwx 2 student 73 Jan 14 .ssh
drwx 2 student 6 Jan 14 Templates
drwx 2 student 6 Jan 14 Videos
-rw- 1 student 1095 Jan 14 .viminfo
-----
lab_review.txt
```

Оценка

На рабочей станции запустите команду **lab edit-review grade**, чтобы подтвердить успех этого упражнения.

```
[student@workstation ~]$ lab edit-review grade
```

Завершение

На рабочей станции запустите сценарий **lab edit-review finish**, чтобы завершить упражнение.

```
[student@workstation ~]$ lab edit-review finish
```

На этом лабораторная работа завершена.

РЕЗЮМЕ

В этой главе вы узнали:

- У запущенных программ или процессов есть три стандартных канала связи: стандартный ввод, стандартный вывод и стандартная вывод ошибки.
- Вы можете использовать перенаправление ввода-вывода для чтения стандартного ввода из файла или записи вывода, или ошибок из процесса в файл.
- Конвейеры могут использоваться для соединения стандартного вывода одного процесса со стандартным вводом другого процесса, а также могут использоваться для форматирования вывода или построения сложных команд.
- Вы должны знать, как использовать хотя бы один текстовый редактор командной строки. **Vim** обычно установлен в операционной системе.
- Переменные оболочки могут помочь вам запускать команды и уникальны для конкретного сеанса оболочки shell.
- Переменные среды могут помочь вам настроить поведение оболочки или запускаемых ею процессов.

ГЛАВА 6

УПРАВЛЕНИЕ ЛОКАЛЬНЫМИ ПОЛЬЗОВАТЕЛЯМИ И ГРУППАМИ

ЦЕЛЬ

Создание, управление и удаление локальных пользователей и групп, а также администрирование локальных политик паролей.

ЗАДАЧИ

- Опишите назначение пользователей и групп в системе Linux.
- Переключитесь на учетную запись суперпользователя для управления системой Linux и предоставьте другим пользователям доступ суперпользователя с помощью команды **sudo**.
- Создавать, изменять и удалять локально определенные учетные записи пользователей.
- Создание, изменение и удаление локально определенных групповых учетных записей.
- Установите политику управления паролями для пользователей, а также вручную блокируйте и разблокируйте учетные записи пользователей.

РАЗДЕЛЫ

- Описание концепций пользователей и групп (и контрольный опрос)
- Получение доступа суперпользователя (и упражнения с пошаговыми инструкциями)
- Управление учетными записями локальных пользователей (и упражнения с пошаговыми инструкциями)
- Управление учетными записями локальных групп (и упражнения с пошаговыми инструкциями)
- Управление паролями пользователей (и упражнения с пошаговыми инструкциями)

ЛАБОРОТОРНАЯ РАБОТА

Управление локальными пользователями и группами Linux.

ОПИСАНИЕ КОНЦЕПЦИИ ПОЛЬЗОВАТЕЛЯ И ГРУППЫ

ЗАДАЧИ

После завершения этого раздела вы будете способны описать назначение пользователей и групп в системе Linux.

ЧТО ТАКОЕ ПОЛЬЗОВАТЕЛЬ?

Учетная запись пользователя используется для обеспечения границ безопасности между разными людьми и программами, которые могут выполнять команды.

У пользователей есть имена пользователей (*user names*), чтобы идентифицировать их для пользователей-людей и облегчить работу с ними. Внутренне система различает учетные записи пользователей по присвоенному им уникальному идентификационному номеру, идентификатору пользователя (*user ID*) или **UID**. Если учетная запись пользователя используется людьми, ей обычно назначается секретный пароль, который пользователь будет использовать, чтобы доказать, что он является действительным авторизованным пользователем при входе в систему.

Учетные записи пользователей имеют фундаментальное значение для безопасности системы. Каждый процесс (запущенная программа) в системе запускается от имени конкретного пользователя. Владелец каждого файла - конкретный пользователь. Владение файлом помогает системе обеспечить контроль доступа пользователей к файлам. Пользователь, связанный с запущенным процессом, определяет файлы и каталоги, доступные этому процессу.

Существует три основных типа учетных записей пользователей: суперпользователь (*superuser*), системные пользователи (*system users*) и обычные пользователи (*regular users*).

- Учетная запись суперпользователя предназначена для администрирования системы. Имя суперпользователя - **root**, а **UID** учетной записи **0**. Суперпользователь имеет полный доступ к системе.
- В системе есть системные учетные записи пользователей, которые используются процессами, предоставляющими вспомогательные услуги. Эти процессы или демоны (*daemons*) обычно не нужно запускать от имени суперпользователя. Им назначаются непривилегированные учетные записи, которые позволяют им защищать свои файлы и другие ресурсы друг от друга и от обычных пользователей в системе. Пользователи не входят в систему в интерактивном режиме, используя системную учетную запись.
- У большинства пользователей есть обычные учетные записи, которые они используют в повседневной работе. Как и пользователи системы, обычные пользователи имеют ограниченный доступ к системе.

Вы можете использовать команду **id**, чтобы показать информацию о текущем вошедшем в систему пользователе.

```
[user01@host ~]$ id  
uid=1000(user01) gid=1000(user01) groups=1000(user01)  
context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
```

Чтобы просмотреть основную информацию о другом пользователе, передайте имя пользователя команде **id** в качестве аргумента.

```
[user01@host]$ id user02  
uid=1002(user02) gid=1001(user02) groups=1001(user02)  
context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
```

Чтобы просмотреть владельца файла, используйте команду **ls -l**. Чтобы просмотреть владельца каталога, используйте команду **ls -ld**. В следующем выводе данных третий столбец показывает имя пользователя.

```
[user01@host ~]$ ls -lfile1  
-rw-rw-r--. 1 user01 user01 0 Feb 5 11:10 file1  
[user01@host]$ ls -ld dir1  
drwxrwxr-x. 2 user01 user01 6 Feb 5 11:10 dir1
```

Чтобы просмотреть информацию о процессе, используйте команду **ps**. По умолчанию отображаются только процессы в текущей оболочке shell. Добавьте возможность просмотра всех процессов с помощью терминала. Чтобы просмотреть пользователя, связанного с процессом, включите параметр **u**. В следующем выводе данных имя пользователя отображается в первом столбце.

```
[user01@host]$ ps -au  
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START  TIME COMMAND  
root      777  0.0  0.0 225752  1496 tty1      Ss+  11:03  0:00 /sbin/agetty -o -  
p -- \u --noclear tty1 linux  
root      780  0.0  0.1 225392  2064 ttyS0     Ss+  11:03  0:00 /sbin/agetty -o -  
p -- \u --keep-baud 115200,38400,9600  
user01    1207  0.0  0.2 234044  5104 pts/0      Ss   11:09  0:00 -bash  
user01    1319  0.0  0.2 266904  3876 pts/0      R+   11:33  0:00 ps au
```

В выходных данных предыдущей команды пользователи отображаются по именам, но операционная система внутри себя для отслеживания пользователей использует идентификаторы **UID**. Сопоставление имен пользователей с **UID** определяется в базах данных учетной записи. По умолчанию системы используют файл **/etc/passwd** для хранения информации о локальных пользователях.

Каждая строка в файле **/etc/passwd** содержит информацию об одном пользователе. Он разделен на семь полей, разделенных двоеточиями. Далее приведён пример строки из **/etc/passwd**:

❶user01:❷x:❸1000:❹1000:❺User One:❻/home/user01:❻/bin/bash

- ❶ Имя пользователя для этого пользователя (**user01**).

2. Раньше здесь хранился пароль пользователя в зашифрованном виде. Теперь пароль был перемещен в файл `/etc/shadow`, о котором будет рассказано позже. Данное поле всегда должно иметь значениех.
3. Номер **UID** для этой учетной записи пользователя (**1000**).
4. Номер **GID** для основной группы этой учетной записи пользователя (**1000**). Группы будут обсуждаться позже в этом разделе.
5. Настоящее имя этого пользователя (**User One**).
6. Домашний каталог для этого пользователя (`/home/user01`). Это начальный рабочий каталог при запуске оболочки`shell`, содержащий данные пользователя и параметры конфигурации.
7. Программа оболочки`shell` по умолчанию для этого пользователя, которая запускается при входе в систему (`/bin/bash`). Для обычного пользователя это обычно программа, которая предоставляет подсказку командной строки пользователя. Системный пользователь может использовать `/sbin/nologin`, если интерактивный вход для этого пользователя запрещен.

ЧТО ТАКОЕ ГРУППА?

Группа - это совокупность пользователей, которым необходимо предоставить общий доступ к файлам и другим системным ресурсам. Группы могут использоваться для предоставления доступа к файлам группе пользователей, а не одному пользователю.

Как и у пользователей, у групп есть названия, чтобы с ними было легче работать. Внутри системы различают группы по присвоенному им уникальному идентификационному номеру, идентификатору группы или **GID**.

Сопоставление имен групп с **GID** определяется в базах данных информации об учетных записях групп. По умолчанию системы используют файл `/etc/group` для хранения информации о локальных группах.

Каждая строка в файле `/etc/group` содержит информацию об одной группе. Каждая запись группы разделена на четыре поля, разделенных двоеточиями. Вот пример строки из `/etc/group`:

```
❶ group01:❷ x:❸ 10000:❹ user01,user02,user03
```

1. Имя группы для этой группы (**group01**).
2. Поле устаревшего пароля группы. Это поле всегда должно быть **x**.
3. Номер **GID** для этой группы (**10000**).
4. Список пользователей, которые входят в эту группу, как дополнительную группу (**user01**, **user02**, **user03**). Основные (или стандартные) и дополнительные группы обсуждаются позже в этом разделе.

Основные группы и дополнительные группы

У каждого пользователя есть ровно одна основная группа. Для локальных пользователей это группа, указанная по номеру **GID** в файле `/etc/passwd`. По умолчанию это группа, которой будут принадлежать новые файлы, созданные пользователем.

Обычно, когда вы создаете нового обычного пользователя, создается новая группа с тем же именем, что и у этого пользователя. Данная группа используется, как основная группа для нового пользователя, и этот пользователь является единственным членом этой частной группы пользователей. Оказывается, это помогает упростить управление правами доступа к файлам, о чем мы поговорим позже в этом курсе.

У пользователей также могут быть дополнительные группы. Членство в дополнительных группах определяется файлом **/etc/group**. Пользователям предоставляется доступ к файлам в зависимости от того, есть ли доступ у какой-либо из их групп. Не имеет значения, являются ли группа или группы, у которых есть доступ, первичными или дополнительными для пользователя.

Например, если у пользователя **user01** есть основная группа **user01** и дополнительные группы **wheel** и **webadmin**, то этот пользователь может читать файлы, доступные для чтения любой из этих трех групп.

Команду **id** также можно использовать, чтобы узнать о членстве пользователя в группе.

```
[user03@host ~]$ id  
uid=1003(user03) gid=1003(user03) groups=1003(user03),10(wheel),10000(group01)  
context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
```

В предыдущем примере **user03** имеет группу **user03** в качестве своей основной группы (**gid**). В элементе **groups** перечислены все группы для этого пользователя, кроме основной группы **user03**, у пользователя есть группы **wheel** и **group01** в качестве дополнительных групп.



РЕКОМЕНДАЦИИ

Справочные страницы **manid(1)**, **passwd(5)** и **group(5)**

info libc (Справочное руководство по библиотеке GNU C)

- Раздел 30: Пользователи и группы

(Обратите внимание, что для того, чтобы этот информационный узел был доступен, должен быть установлен пакет **glibc-devel**.)

КОНТРОЛЬНЫЙ ОПРОС

ОПИСАНИЕ ПОЛЬЗОВАТЕЛЕЙ И КОНЦЕПЦИЙ ГРУППЫ

Выберите правильный ответ на следующие вопросы:

1. Какой элемент представляет собой число, которое идентифицирует пользователя на самом фундаментальном уровне?
 - a. primary user
 - b. UID
 - c. GID
 - d. Username

2. Какой элемент представляет программу, которая предоставляет пользователю командную строку?
 - a. Первичная оболочка
 - b. Домашний каталог
 - c. Оболочка входа
 - d. Имя команды

3. В каком элементе или файле содержится информация о локальной группе?
 - a. home directory
 - b. /etc/passwd
 - c. /etc/GID
 - d. /etc/group

4. Какой элемент или файл представляет собой расположение личных файлов пользователя?
 - a. Домашний каталог
 - b. Оболочка входа в систему shell
 - c. /etc/passwd
 - d. /etc/group

5. Какой элемент представляет собой число, которое идентифицирует группу на самом фундаментальном уровне?
 - a. Основная (primary) группа
 - b. UID
 - c. GID
 - d. groupid

6. В каком элементе или файле содержится информация об учетной записи локального пользователя?
 - a. Домашний каталог
 - b. /etc/passwd
 - c. /etc/UID

d. /etc/group

7. Что определяет четвертое поле в файле **/etc/passwd**?

- a.** Домашний (home) каталог
- b.** **UID**
- c.** Оболочку входа в систему
- d.** Основную (primary) группу

РЕШЕНИЕ

ОПИСАНИЕ ПОЛЬЗОВАТЕЛЕЙ И КОНЦЕПЦИЙ ГРУППЫ

Выберите правильный ответ на следующие вопросы:

1. Какой элемент представляет собой число, которое идентифицирует пользователя на самом фундаментальном уровне?
 - a. primary user
 - b. **UID**
 - c. GID
 - d. Username
2. Какой элемент представляет программу, которая предоставляет пользователю командную строку?
 - a. Первичная оболочка shell
 - b. Домашний (home) каталог
 - c. **Оболочка входа (login shell)**
 - d. Имя команды
3. В каком элементе или файле содержится информация о локальной группе?
 - a. home directory
 - b. /etc/passwd
 - c. /etc/GID
 - d. **/etc/group**
4. Какой элемент или файл представляет собой расположение личных файлов пользователя?
 - a. **Домашний (home) каталог**
 - b. Оболочка входа в систему(loginshell)
 - c. /etc/passwd
 - d. /etc/group
5. Какой элемент представляет собой число, которое идентифицирует группу на самом фундаментальном уровне?
 - a. Основная (primary) группа
 - b. UID
 - c. **GID**
 - d. groupid
6. В каком элементе или файле содержится информация об учетной записи локального пользователя?
 - a. Домашний каталог
 - b. **/etc/passwd**

- c. /etc/UID
- d. /etc/group

7. Что определяет четвертое поле в файле **/etc/passwd**?

- a. Домашний (home) каталог
- b. UID
- c. Оболочку входа в систему
- d. Основную (primary) группу**

ПОЛУЧЕНИЕ ДОСТУПА ДЛЯ СУПЕРПОЛЬЗОВАТЕЛЯ

ЗАДАЧИ

После завершения этого раздела вы сможете переключиться на учетную запись суперпользователя для управления системой Linux и предоставить другим пользователям доступ суперпользователя с помощью команды **sudo**.

СУПЕРПОЛЬЗОВАТЕЛЬ

В большинстве операционных систем есть своего рода суперпользователь - пользователь, обладающий всей властью над системой. В Red Hat Enterprise Linux это пользователь **root**. Он имеет право отменять обычные привилегии в файловой системе и используется для управления и администрирования системы. Для выполнения таких задач, как установка или удаление программного обеспечения, а также для управления системными файлами и каталогами, пользователи должны повысить свои права до пользователя **root**.

Пользователь **root** только среди обычных пользователей может управлять большинством устройств, но есть несколько исключений. Например, обычные пользователи могут управлять съемными устройствами, такими как USB-устройства. Таким образом, обычные пользователи могут добавлять и удалять файлы и иным образом управлять съемным устройством, но только **root** может управлять «фиксированными» жесткими дисками по умолчанию.

Однако эта неограниченная привилегия влечет за собой ответственность. Пользователь **root** имеет неограниченные возможности повредить систему: удалять файлы и каталоги, удалять учетные записи пользователей, добавлять лазейки и т. д. Если учетная запись пользователя **root** будет скомпрометирована, кто-то другой будет иметь административный контроль над системой. На протяжении всего курса администраторам рекомендуется входить в систему как обычный пользователь и повышать привилегии до **root** только при необходимости.

Учетная запись **root** в Linux примерно эквивалентна учетной записи локального администратора в Microsoft Windows. В Linux большинство системных администраторов входят в систему как непривилегированный пользователь и используют различные инструменты для временного получения привилегий **root**.



ПРЕДУПРЕЖДЕНИЕ

Одна из распространенных практик в Microsoft Windows в прошлом заключалась в том, что локальный пользователь-администратор входил в систему напрямую для выполнения обязанностей системного администратора. Хотя это возможно в Linux, Red Hat рекомендует системным администраторам не входить в систему напрямую как **root**. Вместо этого системные администраторы должны войти в систему как обычный пользователь и использовать другие механизмы (например, **su**, **sudo** или **PolicyKit**) для временного получения привилегий суперпользователя.

При входе в систему как суперпользователь вся среда рабочего стола без необходимости запускается с административными привилегиями. В этой ситуации любая уязвимость системы безопасности, которая обычно ставит под угрозу только учетную запись пользователя, может поставить под угрозу всю систему.

ПЕРЕКЛЮЧЕНИЕ ПОЛЬЗОВАТЕЛЕЙ

Команда **su** позволяет пользователям переключаться на другую учетную запись пользователя. Если вы запустите **su** из учетной записи обычного пользователя, вам будет предложено ввести пароль учетной записи, на которую вы хотите переключиться. Когда **root** запускает **su**, вводить пароль пользователя не нужно.

```
[user01@host ~]$ su - user02  
Password:  
[user02@host ~]$
```

Если вы опустите имя пользователя, команда **su** или (**su -**) попытается переключиться на учётную запись **root** по умолчанию.

```
[user01@host ~]$ su -  
Password:  
[root@host ~]#
```

Команда **su** запускает оболочку *shell* (*non-login shell*) без входа в систему, а команда **su -** (с опцией тире) запускает оболочку *shell* (*login shell*) входа в систему. Основное различие между этими двумя командами состоит в том, что **su -** устанавливает среду оболочки, как если бы это был новый логин от имени этого пользователя, в то время как **su** просто запускает оболочку от имени этого пользователя, но использует настройки среды исходного пользователя.

В большинстве случаев администраторам следует запустить **su -**, чтобы получить оболочку с обычными настройками среды целевого пользователя. Для получения дополнительной информации см. Справочную страницу **bash(1)**.



ПРИМЕЧАНИЕ

Команда **su** чаще всего используется для получения интерфейса командной строки (приглашения оболочки), который работает от имени другого пользователя, обычно **root**. Однако с параметром **-c** его можно использовать, как **runas** утилит Windows, для запуска произвольной программы от имени другого пользователя. Запустите **info su**, чтобы просмотреть дополнительные сведения.

КОМАНДЫ РАБОТЫ С SUDO

В некоторых случаях учетная запись пользователя **root** может вообще не иметь действительного пароля по соображениям безопасности. В этом случае пользователи не могут войти в систему как **root** напрямую с паролем, а **su** нельзя использовать для получения

интерактивной оболочки. Одним из инструментов, который можно использовать для получения **root**-доступа в этом случае, является команда **sudo**.

В отличие от **su**, **sudo** обычно требует, чтобы пользователи вводили собственный пароль для аутентификации, а не пароль учетной записи пользователя, к которой они пытаются получить доступ. То есть пользователям, которые используют **sudo** для запуска команд от имени **root**, не нужно знать пароль **root**. Вместо этого они используют свои собственные пароли для аутентификации доступа.

Кроме того, **sudo** можно настроить так, чтобы определенные пользователи могли запускать любую команду от имени другого пользователя или только некоторые команды от имени этого пользователя.

Например, когда **sudo** настроен так, чтобы позволить пользователю **user01** запускать команду **usermod** от имени пользователя **root**, **user01** может запустить следующую команду, чтобы заблокировать или разблокировать учетную запись пользователя:

```
[user01@host ~]$ sudo usermod -L user02
[sudo] password for user01:
[user01@host ~]$ su - user02
Password:
su: Authenticationfailure
[user01@host ~]$
```

Если пользователь пытается запустить команду от имени другого пользователя, а конфигурация **sudo** не позволяет этого, команда будет заблокирована, попытка будет зарегистрирована, и по умолчанию пользователю **root** будет отправлено электронное письмо.

```
[user02@host ~]$ sudo tail /var/log/secure
[sudo] password for user02:
user02 is not in the sudoers file. This incident will be reported.
[user02@host ~]$
```

Еще одно преимущество использования **sudo** заключается в том, что все выполняемые команды по умолчанию записываются в **/var/log/secure**.

```
[user01@host ~]$ sudo tail /var/log/secure
...output omitted...
Feb  6 20:45:46 host sudo[2577]:  user01 : TTY=pts/0 ; PWD=/home/user01 ;
USER=root ; COMMAND=/sbin/usermod -L user02
...output omitted...
```

В Red Hat Enterprise Linux 7 и Red Hat Enterprise Linux 8 все члены группы **wheel** могут использовать **sudo** для выполнения команд от имени любого пользователя, включая **root**. Пользователю предлагается ввести собственный пароль. Это изменение по сравнению с Red Hat

Enterprise Linux 6 и более ранними версиями, где пользователи, входившие в группу **wheel**, по умолчанию не получали административный доступ.



ПРЕДУПРЕЖДЕНИЕ

RHEL 6 по умолчанию не предоставлял группе **wheel** никаких особых привилегий. Пользователи, которые использовали эту группу для нестандартных целей, могут быть удивлены, когда RHEL 7 и RHEL 8 автоматически предоставляют всем участникам **wheel** полные привилегии **sudo**. Это может привести к тому, что неавторизованные пользователи получат административный доступ к системам RHEL 7 и RHEL 8.

Исторически сложилось так, что в UNIX-подобных системах членство в группе **wheel** используется для предоставления или управления доступом суперпользователя.

Получение интерактивной оболочки **root** с помощью **sudo**

Если в системе есть неадминистративная учетная запись пользователя, которая может использовать **sudo** для запуска команды **su**, вы можете выполнить **sudo su** -от этой учетной записи, чтобы получить интерактивную оболочку пользователя **root**. Это работает, потому что **sudo** будет запускать **su** - как **root**, и **root** не должен вводить пароль для использования **su**.

Другой способ получить доступ к учетной записи **root** с помощью **sudo** - использовать команду **sudo -i**. После выполнения команда акаут пользователя переключится на учетную запись **root** и запустит оболочку этого пользователя по умолчанию (обычно bash) и связанные сценарии входа в оболочку. Если вы просто хотите запустить оболочку **shell**, вы можете использовать команду **sudo -s**.

Например, администратор может получить интерактивную оболочку от имени пользователя **root** на экземпляре AWS EC2, используя аутентификацию с открытым ключом **SSH** для входа в систему как обычный пользователь **ec2-user**, а затем запустив **sudo -i** для получения оболочки пользователя **root**.

```
[ec2-user@host ~]$ sudo -i  
[sudo] password for ec2-user:  
[root@host ~]#
```

Команды **sudo su** - и команда **sudo -i** ведут себя по-разному. Об этом будет кратко сказано в конце раздела.

```
%wheel      ALL=(ALL)      ALL
```

В этой строке **%wheel** - это пользователь или группа, к которым применяется правило, % указывает, что это группа **wheel**. Сочетание **ALL=(ALL)** определяет, что на любом хосте, где может находиться этот файл, члены группы **wheel** могут запускать любую команду. Последний **ALL** указывает, что **wheel** может запускать эти команды от имени любого пользователя в системе.

По умолчанию конфигурационном файле `/etc/sudoers` также включено содержимое любых файлов в каталоге `/etc/sudoers.d` как часть этого файла. Это позволяет администратору добавить доступ `sudo` для пользователя, просто поместив соответствующий файл в этот каталог.



ПРИМЕЧАНИЕ

Использование дополнительных файлов в каталоге `/etc/sudoers.d` удобно и просто. Вы можете включить или отключить доступ `sudo`, просто скопировав файл в каталог или удалив его из каталога.

В этом курсе вы будете создавать и удалять файлы в каталоге `/etc/sudoers.d`, чтобы настроить доступ `sudo` для пользователей и групп.

Чтобы разрешить полный доступ `sudo` для пользователя `user01`, вы можете создать `/etc/sudoers.d/user01` со следующим содержимым:

```
user01    ALL=(ALL)      ALL
```

Чтобы включить полный доступ sudo для группы `group01`, вы можете создать `/etc/sudoers.d/group01` со следующим содержимым:

```
%group01  ALL=(ALL)      ALL
```

Также можно настроить `sudo`, чтобы пользователь мог запускать команды от имени другого пользователя без ввода своего пароля:

```
ansible ALL=(ALL)      NOPASSWD: ALL
```

Хотя есть очевидные риски безопасности при предоставлении такого уровня доступа пользователю или группе, он часто используется с облачными экземплярами, виртуальными машинами и системами подготовки для помощи в настройке серверов. Учетная запись с таким доступом должна быть тщательно защищена и может потребовать аутентификации с открытым ключом `SSH`, чтобы пользователь удаленной системы вообще мог получить к ней доступ.

Например, официальный **AMI** для Red Hat Enterprise Linux на Amazon Web Services Marketplace поставляется с заблокированными паролями пользователей `root` и пользователей `ec2`. Учетная запись пользователя `ec2` настроена так, чтобы разрешить удаленный интерактивный доступ через аутентификацию с открытым ключом `SSH`. Пользователь `ec2-user` также может запускать любую команду от имени пользователя `root` без пароля, потому что последняя строка файла `AMI/etc/sudoers` настроена следующим образом:

```
ec2-user ALL=(ALL) NOPASSWD: ALL
```

Требование ввести пароль для sudo может быть повторно включено, или могут быть внесены другие изменения для усиления безопасности как часть процесса настройки системы.



ПРИМЕЧАНИЕ

В этом курсе вы часто будете видеть, что **sudosu** - используется вместо **sudo -i**. Обе команды работают, но между ними есть некоторые тонкие различия.

Команда **sudosu** - устанавливает среду **root** точно так же, как при обычном входе в систему, поскольку команда **su** игнорирует настройки, сделанные **sudo**, и настраивает среду с нуля.

Конфигурация по умолчанию команды **sudo -i** фактически устанавливает некоторые детали среды пользователя **root** иначе, чем при обычном входе в систему. Например, он немного по-другому устанавливает переменную среды **PATH**. Это влияет на то, где оболочка будет искать выполняемые команды.

Вы можете сделать **sudo -i** более похожим на **su** - отредактировав **/etc/sudoers** с помощью **visudo**. Найдите следующую линию в содержание файла:

```
Defaults secure_path = /sbin:/bin:/usr/sbin:/usr/bin
```

и замените его следующими двумя строками:

```
Defaults secure_path = /usr/local/bin:/usr/bin  
Defaults>root secure_path = /usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin
```

Для большинства целей это несущественная разница. Однако для согласованности настроек **PATH** в системах с файлом по умолчанию **/etc/sudoers** авторы этого курса в большинстве случаев используют **sudo su** - в примерах.



РЕКОМЕНДАЦИИ

Справочные страницы **man su(1)**, **sudo(8)**, **visudo(8)** и **sudoers(5)**

info libc persona (Справочное руководство по библиотеке GNU C)

- Раздел 30.2: Сущность процесса (The Persona of a Process)

(Обратите внимание, что для того, чтобы этот информационный узел был доступен, должен быть установлен пакет **glibc-devel**.)

УПРАЖНЕНИЯ С ПОШАГОВЫМИ ИНСТРУКЦИЯМИ

ПОЛУЧЕНИЕ ДОСТУПА ДЛЯ СУПЕРПОЛЬЗОВАТЕЛЯ

В этом упражнении вы попрактикуетесь в переключении на учетную запись **root** и выполнении команд от имени пользователя **root**.

РЕЗУЛЬТАТЫ

Вы должны быть способны:

- Используйте **sudo**, чтобы переключиться в акаунт **root** и получить доступ к интерактивной оболочке как **root**, не зная пароля суперпользователя.
- Объяснить, как **su** и **su -** могут влиять на среду оболочки, выполняя или не выполняя сценарии входа в систему.
- Используйте **sudo** для запуска других команд от имени пользователя **root**.

ПРЕЖДЕ ЧЕМ ВЫ НАЧНЕТЕ

Войдите на рабочую станцию как пользователь **student**, используя в качестве пароля слово **student**.

На рабочей станции запустите команду **lab users-sudo start**, чтобы начать упражнение. Этот сценарий создает необходимые учетные записи пользователей и файлы для правильной настройки среды.

```
[student@workstation ~]$ lab users-sudo start
```

1. С рабочей станции откройте сеанс SSH на сервере как студент.

```
[student@workstation ~]$ ssh student@servera  
...output omitted...  
[student@servera ~]$
```

2. Изучите среду оболочки студента. Просмотр информации о текущем пользователе и группе и отображение текущего рабочего каталога. Также просмотрите переменные среды, которые определяют домашний каталог пользователя и расположение исполняемых файлов пользователя.

- 2.1. Выполните команду **id**, чтобы просмотреть информацию о текущем пользователе и группе.

```
[student@servera ~]$ id  
uid=1000(student) gid=1000(student) groups=1000(student),10(wheel)  
context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
```

- 2.2. Запустите **pwd**, чтобы отобразить текущий рабочий каталог.

```
[student@servera ~]$ pwd  
/home/student
```

- 2.3. Выведите значения переменных **HOME** и **PATH**, чтобы определить путь к домашнему каталогу и исполняемым файлам пользователя соответственно.

```
[student@servera ~]$ echo $HOME  
/home/student  
[student@servera ~]$ echo $PATH  
/home/student/.local/bin:/home/student/bin:/usr/local/bin:/usr/bin:/usr/local/  
sbin:/usr/sbin
```

3. Переключитесь напользователя **root** в оболочке без входа в систему и исследуйте новую среду оболочки.

- 3.1. Запустите **sudo su** в командной строке, чтобы стать пользователем **root**.

```
[student@servera ~]$ sudo su  
[sudo] password for student: student  
[root@servera student]#
```

- 3.2. Выполните команду **id**, чтобы просмотреть информацию о текущем пользователе и группе.

```
[root@servera student]# id  
uid=0(root) gid=0(root) groups=0(root)  
context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
```

- 3.3. Запустите **pwd**, чтобы отобразить текущий рабочий каталог.

```
[root@servera student]# pwd
```

```
/home/student
```

- 3.4. Выведите значения переменных **HOME** и **PATH**, чтобы определить путь к домашнему каталогу и исполняемым файлам пользователя соответственно.

```
[root@servera student]# echo $HOME  
/root  
[root@servera student]# echo $PATH  
/sbin:/bin:/usr/sbin:/usr/bin:/usr/local/sbin:/usr/local/bin
```

Если у вас уже есть некоторый опыт работы с Linux и командой **su**, вы, возможно, ожидали, что использование **su** без опции тире (-) для перехода в режим **root** приведет к тому, что вы сохраните текущие значение **PATH** пользователя **student**. Этого не произошло. Как вы увидите на следующем шаге, это тоже не обычное значение **PATH** для пользователя **root**.

Что случилось? Разница в том, что вы не запускали **su** напрямую. Вместо этого вы запускали **su** от имени пользователя **root** с помощью **sudo**, потому что у вас не было пароля суперпользователя. Команда **sudo** изначально переопределяет переменную **PATH** из исходной среды по соображениям безопасности. Любая команда, выполняемая после первоначального переопределения, может по-прежнему обновлять переменную **PATH**, как вы увидите в следующих шагах.

- 3.5. Выйдите из оболочки пользователя **root**, чтобы вернуться в оболочку пользователя **student**.

```
[root@servera student]# exit  
exit
```

4. Переключитесь на **root** в оболочке входа в систему и изучите новую среду оболочки.

- 4.1. Запустите **sudo su** - в приглашении оболочки, чтобы стать пользователем **root**.

```
[student@servera ~]$ sudo su -  
[root@servera ~]#
```

Обратите внимание на разницу в приглашении оболочки по сравнению с запросом **sudo su** на предыдущем шаге.

sudo может запрашивать или не запрашивать пароль учащегося, в зависимости от периода ожидания **sudo**. По умолчанию период ожидания составляет пять минут. Если вы прошли аутентификацию в **sudo** в течение последних пяти минут, **sudo** не будет запрашивать пароль. Если с момента аутентификации в **sudo** прошло более пяти минут, вам необходимо ввести **student** в качестве пароля для аутентификации в **sudo**.

- 4.2. Выполните команду **id**, чтобы просмотреть информацию о текущем пользователе и группе.

```
[root@servera ~]# id  
uid=0(root) gid=0(root) groups=0(root)  
context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
```

- 4.3. Запустите **pwd**, чтобы отобразить текущий рабочий каталог.

```
[root@servera ~]# pwd  
/root
```

- 4.4. Выведите значения переменных **HOME** и **PATH**, чтобы определить домашний каталог и путь к пользовательским исполняемым файлам соответственно.

```
[root@servera ~]# echo $HOME  
/root  
[root@servera ~]# echo $PATH  
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/root/bin
```

Как и на предыдущем шаге, после того, как **sudo** сбросил переменную **PATH** из настроек в среде оболочки пользователя **student**, команда **su** запустила сценарии входа в оболочку для **root** и установила для переменной **PATH** еще одно значение. Команда **su** без опции тире (-) этого не делает.

- 4.5. Выйдите из оболочки пользователя **root**, чтобы вернуться в оболочку пользователя **student**.

```
[root@servera ~]# exit  
logout  
[student@servera ~]$
```

5. Убедитесь, что пользователь **operator1** настроен для выполнения любой команды от имени любого пользователя, использующего **sudo**.

```
[student@servera ~]$ sudo cat /etc/sudoers.d/operator1  
operator1 ALL=(ALL) ALL
```

6. Станьте **operator1** и просмотрите содержимое **/var/log/messages**. Скопируйте **/etc/motd** в **/etc/motdOLD** и удалите его (**/etc/motdOLD**). Для этих операций требуются права администратора, поэтому используйте **sudo** для выполнения этих команд от имени суперпользователя. Не переключайтесь на **root** с помощью **sudo su** или **sudosu -**.

6.1. Переключитесь на **operator1**.

```
[student@servera ~]$ su - operator1  
Password: redhat  
[operator1@servera ~]$
```

6.2. Попытайтесь просмотреть последние пять строк **/var/log/messages** без использования **sudo**. Вы должно потерпеть неудачу.

```
[operator1@servera ~]$ tail -5 /var/log/messages  
tail: cannot open '/var/log/messages' for reading: Permission denied
```

6.3. Попытайтесь просмотреть последние пять строк **/var/log/messages** с помощью **sudo**. Команда должна выполниться успешно.

```
[operator1@servera ~]$ sudo tail -5 /var/log/messages  
[sudo] password for operator1: redhat  
Jan 23 15:53:36 servera su[2304]: FAILED SU (to operator1) student on pts/1  
Jan 23 15:53:51 servera su[2307]: FAILED SU (to operator1) student on pts/1  
Jan 23 15:53:58 servera su[2310]: FAILED SU (to operator1) student on pts/1  
Jan 23 15:54:12 servera su[2322]: (to operator1) student on pts/1  
Jan 23 15:54:25 servera su[2353]: (to operator1) student on pts/1
```



ПРИМЕЧАНИЕ

Предыдущий вывод может отличаться в вашей системе.

6.4. Попытка сделать копию **/etc/motd** как **/etc/motdOLD** без использования **sudo**. Вы должны потерпеть неудачу.

```
[operator1@servera ~]$ cp /etc/motd /etc/motdOLD  
cp: cannot create regular file '/etc/motdOLD': Permission denied
```

6.5. Попытайтесь сделать копию **/etc/motd** как **/etc/motdOLD** с помощью **sudo**. Команда должна выполниться успешно.

```
[operator1@servera ~]$ sudo cp /etc/motd /etc/motdOLD  
[operator1@servera ~]$
```

6.6. Попытка удалить **/etc/motdOLD** без использования **sudo**. Вы потерпели неудачу.

```
[operator1@servera ~]$ sudo rm /etc/motdOLD  
[operator1@servera ~]$
```

6.7. Попытка удалить **/etc/motdOLD** с помощью **sudo**. Будет успешной.

```
[operator1@servera ~]$ sudo rm /etc/motdOLD  
[operator1@servera ~]$
```

6.8. Выйдите из оболочки пользователя **operator1**, чтобы вернуться в оболочку пользователя **student**.

```
[operator1@servera ~]$ exit  
logout  
[student@servera ~]$
```

6.9. Выйти из сервера.

```
[student@servera ~]$ exit  
logout  
Connection to servera closed.  
[student@workstation ~]$
```

Завершение

На рабочей станции выполните **lab users-sudo finish**, чтобы завершить упражнение. Данный сценарий удаляет учетные записи пользователей и файлы, созданные в начале упражнения, чтобы обеспечить чистоту среды.

```
[student@workstation ~]$ lab users-sudo finish
```

УПРАВЛЕНИЕ ЛОКАЛЬНЫМИ УЧЕТАМИ ПОЛЬЗОВАТЕЛЕЙ

ЗАДАЧИ

После завершения этого раздела вы сможете создавать, изменять и удалять локальные учетные записи пользователей.

УПРАВЛЕНИЕ ЛОКАЛЬНЫМИ ПОЛЬЗОВАТЕЛЯМИ

Для управления локальными учетными записями пользователей можно использовать ряд инструментов командной строки.

Создание пользователей из командной строки

Команда **useradd** *username* создает нового пользователя с именем *username*. Команда устанавливает домашний каталог пользователя и информацию об учетной записи, а также создает личную группу для пользователя с именем *username*. На этом этапе для учетной записи не установлен действующий пароль, и пользователь не может войти в систему, пока он не будет установлен.

Команда **useradd --help** отображает основные параметры, которые можно использовать для отмены значений по умолчанию. В большинстве случаев те же параметры можно использовать с командой **usermod** для изменения существующего пользователя.

Некоторые значения по умолчанию, такие как диапазон допустимых номеров **UID** и правила устаревания пароля по умолчанию,читываются из файла **/etc/login.defs**. Значения в этом файле используются только при создании новых пользователей. Изменение этого файла не влияет на существующих пользователей.

Изменение существующих пользователей из командной строки

Команда **usermod --help** отображает основные параметры, которые можно использовать для изменения учетной записи. Вот некоторые распространенные варианты:

ОПЦИИ usermod:	ПРИМЕНЕНИЕ
-c, --comment COMMENT	Добавьте настоящее имя пользователя в поле комментария.
-g, --gid GROUP	Укажите основную группу для учетной записи пользователя.
-G, --groups GROUPS	Укажите разделенный запятыми список дополнительных групп для учетной записи пользователя.
-a, --append	Используется с параметром -G для добавления дополнительных групп к текущему набору членства пользователя в группах вместо замены набора дополнительных групп новым набором.
-d, --home HOME_DIR	Укажите конкретный домашний каталог для учетной записи пользователя.
-m, --move-home	Переместите домашний каталог пользователя в новое место. Должен использоваться с опцией -d .
-s, --shell SHELL	Укажите конкретную оболочку входа в систему для учетной записи пользователя.
-L, --lock	Заблокируйте учетную запись пользователя.
-U, --unlock	Разблокируйте учетную запись пользователя.

Удаление пользователей из командной строки

- Команда **userdel *username*** удаляет детали пользователя *username* из каталога /etc/passwd, но оставляет домашний каталог пользователя нетронутым.
- Команда **userdel -r *username*** удаляет детали пользователя *username* из каталога /etc/passwd, а также удаляет домашний каталог пользователя.



ПРЕДУПРЕЖДЕНИЕ

Когда пользователь удаляется с помощью **userdel** без указанной опции **-r**, в системе будут файлы, которые принадлежат неназначенному **UID**. Это также может произойти, когда файл, владельцем которого является удаленный пользователь, существует за пределами домашнего каталога этого пользователя. Эта ситуация может привести к утечке информации и другим проблемам с безопасностью.

В Red Hat Enterprise Linux 7 и Red Hat Enterprise Linux 8 команда **useradd** назначает новым пользователям первый свободный **UID**, больший или равный 1000, если вы явно не укажете его с помощью параметра **-u**.

Вот как может происходить утечка информации. Если первый свободный **UID** был ранее назначен учетной записи пользователя, которая с тех пор была удалена из системы, **UID** старого пользователя будет переназначен новому пользователю, давая новому пользователю право владения оставшимися файлами старого пользователя.

Следующий сценарий демонстрирует эту ситуацию.

```
[root@host ~]# useradd user01
[root@host ~]# ls -l /home
drwx----- 3 user01 user01 74 Feb 4 15:22 user01
[root@host ~]# userdel user01
[root@host ~]# ls -l /home
drwx----- 3 1000 1000 74 Feb 4 15:22 user01
[root@host ~]# useradd user02
[root@host ~]# ls -l /home
drwx----- 3 user02 user02 74 Feb 4 15:23 user02
drwx----- 3 user02 user02 74 Feb 4 15:22 user01
```

Обратите внимание, что **user02** теперь владеет всеми файлами, которые ранее принадлежали **user01**.

В зависимости от ситуации одним из решений этой проблемы является удаление всех файлов, не принадлежащих владельцу, из системы при удалении пользователя, создавшего их. Другое решение - вручную назначить незарегистрированные файлы другому пользователю. Пользователь **root** может использовать команду **find / -nouser -o -nogroup**, чтобы найти все незарегистрированные файлы и каталоги.

Установка паролей из командной строки

- Команда **passwd username** устанавливает начальный пароль или изменяет существующий пароль пользователя.
- Пользователь **root** может установить пароль на любое значение. Если пароль не соответствует минимальным рекомендуемым критериям, отображается сообщение, но за ним следует запрос на повторный ввод нового пароля, и все токены успешно обновляются.

```
[root@host ~]# passwd user01
Changing password for user user01.
New password: redhat
BAD PASSWORD: The password fails the dictionary check - it is based on a
dictionary word
Retype new password: redhat
passwd: all authentication tokens updated successfully.
[root@host ~]#
```

- Обычный пользователь должен выбрать пароль длиной не менее восьми символов и также не основанный на словарном слове, имени пользователя или предыдущем пароле.

Диапазоны UID

Определенные номера **UID** и диапазоны номеров используются Red Hat Enterprise Linux для определенных целей.

- UID 0** всегда присваивается учетной записи суперпользователя **root**.
- UID 1-200** - это диапазон «системных пользователей», статически назначаемых системным процессам Red Hat.
- UID 201-999** - это диапазон «системных пользователей», используемых системными процессами, которые не владеют файлами в файловой системе. Обычно они назначаются динамически из доступного пула при установке необходимого в них программного обеспечения. Программы запускаются от имени этих «непривилегированных» системных пользователей, чтобы ограничить их доступ только теми ресурсами, которые им необходимы для работы.
- UID 1000+** - это диапазон, доступный для назначения обычным пользователям.



ПРИМЕЧАНИЕ

До RHEL 7 соглашение заключалось в том, что **UID 1-499** использовался для системных пользователей и **UID 500+** для обычных пользователей. Диапазоны по умолчанию, используемые **useradd** и **groupadd**, можно изменить в файле **/etc/login.defs**.



РЕКОМЕНДАЦИИ

Страницы руководства **manuseradd(8)**, **usermod(8)**, **userdel(8)**

УПРАЖНЕНИЯ С ПОШАГОВЫМИ ИНСТРУКЦИЯМИ

УПРАВЛЕНИЕ ЛОКАЛЬНЫМИ УЧЕТАМИ ПОЛЬЗОВАТЕЛЕЙ

В этом упражнении вы создадите несколько пользователей в своей системе и зададите для них пароли.

РЕЗУЛЬТАТЫ

Вы должны иметь возможность настроить систему Linux с дополнительными учетными записями пользователей.

ПРЕЖДЕ ЧЕМ ВЫ НАЧНЕТЕ

Войдите на рабочую станцию как пользователь **student**, используя в качестве пароля слово **student**.

На рабочей станции запустите команду **lab users-manage start**, чтобы начать упражнение. Данный сценарий обеспечивает правильную настройку среды для выполнения упражнения.

```
[student@workstation ~]$ labusers-managestart
```

1. С рабочей станции откройте сеанс **SSH** на **servera** используя учётную запись **student**.

```
[student@workstation ~]$ ssh student@servera  
...output omitted...  
[student@servera ~]$
```

2. На сервере переключитесь на аккаунт **root** с помощью **sudo**, преобразовав в среду оболочки пользователя **root**.

```
[student@servera ~]$ sudo su -  
[sudo] password for student:  
[root@servera ~]#
```

3. Создайте пользователя **operator1** и убедитесь, что он существует в системе.

```
[root@servera ~]# useradd operator1  
[root@servera ~]# tail /etc/passwd  
...output omitted...  
operator1:x:1001:1001::/home/operator1:/bin/bash
```

4. Установите для **operator1** пароль **redhat**.

```
[root@servera ~]# passwd operator1
Changing password for user operator1.
New password: redhat
BAD PASSWORD: The password is shorter than 8 characters
Retype new password: redhat
passwd: all authentication tokens updated successfully.
```

5. Создайте дополнительных пользователей **operator2** и **operator3**. Определите им паролем слово **redhat**.

5.1. Добавьте пользователя **operator2**. Установите пароль для **operator2** слово **redhat**.

```
[root@servera ~]# useradd operator2
[root@servera ~]# passwd operator2
Changing password for user operator2.
New password: redhat
BAD PASSWORD: The password is shorter than 8 characters
Retype new password: redhat
passwd: all authentication tokens updated successfully.
```

5.2. Добавьте пользователя **operator3**. Установите пароль для пользователя **operator3** слово **redhat**.

```
[root@servera ~]# useradd operator3
[root@servera ~]# passwd operator3
Changing password for user operator3.
New password: redhat
BAD PASSWORD: The password is shorter than 8 characters
Retype new password: redhat
passwd: all authentication tokens updated successfully.
```

6. Обновите учетные записи пользователей **operator1** и **operator2**, включив в них комментарии **Operator One** и **Operator Two** соответственно. Убедитесь, что комментарии добавлены успешно.

6.1. Выполните **usermod -c**, чтобы обновить комментарии учетной записи пользователя **operator1**.

```
[root@servera ~]# usermod -c "Operator One" operator1
```

6.2. Запустите **usermod -c**, чтобы обновить комментарии учетной записи пользователя **operator2**.

```
[root@servera ~]# usermod -c "Operator Two" operator2
```

- 6.3. Убедитесь, что комментарии для каждого из пользователей **operator1** и **operator2** отражены в пользовательских записях.

```
[root@servera ~]# tail /etc/passwd  
...output omitted...  
operator1:x:1001:1001:Operator One:/home/operator1:/bin/bash  
operator2:x:1002:1002:Operator Two:/home/operator2:/bin/bash  
operator3:x:1003:1003::/home/operator3:/bin/bash
```

7. Удалите пользователя **operator3** вместе с любыми личными данными пользователя. Подтвердите, что пользователь успешно удален.

- 7.1. Удалите пользователя **operator3** из системы.

```
[root@servera ~]# userdel -r operator3
```

- 7.2. Подтвердите, что **operator3** успешно удален.

```
[root@servera ~]# tail /etc/passwd  
...output omitted...  
operator1:x:1001:1001:Operator One:/home/operator1:/bin/bash  
operator2:x:1002:1002:Operator Two:/home/operator2:/bin/bash
```

- 7.3. Выдите из оболочки пользователя **root**, чтобы вернуться в оболочку пользователя **student**.

```
[root@servera ~]# exit  
logout  
[student@servera ~]$
```

- 7.4. Выйти из сервера.

```
[student@servera ~]$ exit  
logout  
Connection to servera closed.  
[student@workstation ~]$
```

Завершение

На рабочей станции запустите **lab users-manage finish**, чтобы завершить это упражнение.
Данные сценарий обеспечивает очистку среды.

```
[student@workstation ~]$ labusers-managefinish
```

На этом пошаговое упражнение завершено.

УПРАВЛЕНИЕ УЧЕТНЫМИ ЗАПИСЯМИ ЛОКАЛЬНЫХ ГРУПП

ЗАДАЧИ

После завершения этого раздела учащиеся должны иметь возможность создавать, изменять и удалять учетные записи локальных групп.

УПРАВЛЕНИЕ ЛОКАЛЬНЫМИ ГРУППАМИ

Группа должна существовать, прежде чем пользователь может быть в неё добавлен. Для управления учетными записями локальных групп используются несколько инструментов командной строки.

Создание групп из командной строки

- Команда **groupadd** создает группы. Без параметров команда **groupadd** использует следующий доступный **GID** из диапазона, указанного в файле **/etc/login.defs**, при создании групп.
- Параметр **-g** указывает конкретный **GID** для использования группой.

```
[user01@host ~]$ sudo groupadd -g 10000 group01  
[user01@host ~]$ tail /etc/group  
...output omitted...  
group01:x:10000:
```



ПРИМЕЧАНИЕ

Учитывая автоматическое создание частных групп пользователей (GID 1000+), обычно рекомендуется выделить ряд GID, которые будут использоваться для дополнительных групп. Более высокий диапазон позволит избежать столкновения с системной группой (GID 0-999).

- Параметр **-r** создает системную группу с использованием **GID** из диапазона допустимых системных **GID**, перечисленных в файле **/etc/login.defs**. Элементы конфигурации **SYS_GID_MIN** и **SYS_GID_MAX** в **/etc/login.defs** определяют диапазон системных GID.

```
[user01@host ~]$ sudo groupadd -r group02  
[user01@host ~]$ tail /etc/group  
...output omitted...  
group01:x:10000:  
group02:x:988:
```

Изменение существующих групп из командной строки

- Команда **groupmod** изменяет свойства существующей группы. Параметр **-n** указывает новое имя для группы.

```
[user01@host ~]$ sudo groupmod -n group0022 group02
[user01@host ~]$ tail /etc/group
...output omitted...
group0022:x:988:
```

Обратите внимание, что имя группы обновлено до **group0022** из **group02**.

- Параметр **-g** указывает новый **GID**.

```
[user01@host ~]$ sudo groupmod -g 20000 group0022
[user01@host ~]$ tail /etc/group
...output omitted...
group0022:x:20000:
```

Обратите внимание, что GID обновлен до 20000 с 988.

Удаление групп из командной строки

- Команда **groupdel** удаляет группы.



ПРИМЕЧАНИЕ

Вы не можете удалить группу, если это основная группа любого существующего пользователя. Как и в случае с **userdel**, проверьте все файловые системы, чтобы убедиться, что в системе не осталось файлов, принадлежащих группе.

Изменение членства в группе из командной строки

- Членство в группе контролируется с помощью управления пользователями. Используйте команду **usermod -g**, чтобы изменить основную группу пользователя.

```
[user01@host ~]$ id user02
uid=1006(user02) gid=1008(user02) groups=1008(user02)
[user01@host ~]$ sudo usermod -g group01 user02
[user01@host ~]$ id user02
uid=1006(user02) gid=10000(group01) groups=10000(group01)
```

- Используйте команду **usermod -aG**, чтобы добавить пользователя в дополнительную группу.

```
[user01@host ~]$ id user03  
uid=1007(user03) gid=1009(user03) groups=1009(user03)  
[user01@host ~]$ sudo usermod -aG group01 user03  
[user01@host ~]$ id user03  
uid=1007(user03) gid=1009(user03) groups=1009(user03),10000(group01)
```



ВАЖНО

Использование опции **-a** переводит функцию **usermod** в режим добавления. Без **-a** пользователь будет удален из любой из его текущих дополнительных групп, которые не включены в список опции **-G**.



РЕКОМЕНДАЦИИ

Страницы руководствам **group(5)**, **groupadd(8)**, **groupdel(8)**, и **usermod(8)**

УПРАЖНЕНИЯ С ПОШАГОВЫМИ ИНСТРУКЦИЯМИ

УПРАВЛЕНИЕ УЧЕТНЫМИ ЗАПИСЯМИ ЛОКАЛЬНЫХ ГРУПП

В этом упражнении вы создадите группы, будете использовать их в качестве дополнительных групп для некоторых пользователей, не изменяя основные группы этих пользователей, и настроите одну из групп с доступом **sudo** для выполнения команд от имени пользователя **root**.

В РЕЗУЛЬТАТЕ

Вы будете способны:

- Создавайте группы и используйте их как дополнительные группы.
- Настроить доступ **sudo** для группы.

ПРЕЖДЕ ЧЕМ ВЫ НАЧНЕТЕ

Войдите на рабочую станцию как пользователь **student**, используя в качестве пароля слово **student**.

На рабочей станции запустите команду **lab users-group-manage start**, чтобы начать упражнение. Данный сценарий обеспечивает правильную настройку среды для выполнения упражнения.

```
[student@workstation ~]$ labusers-group-managestart
```

1. С рабочей станции откройте сеанс **SSH** на сервере как пользователь **student**.

```
[student@workstation ~]$ ssh student@servera  
...output omitted...  
[student@servera ~]$
```

2. На сервере переключитесь на **root** с помощью **sudo**, унаследовав полное окружение пользователя **root**.

```
[student@servera ~]$ sudo su -  
[sudo] password for student:  
[root@servera ~]#
```

3. Создайте дополнительную группу операторов с **GID30000**.

```
[root@servera ~]# groupadd -g 30000 operators
```

4. Создайте дополнительную группу **admin**.

```
[root@servera ~]# groupadd admin
```

5. Убедитесь, что существуют группы **operators**, и дополнительная группа **admin**.

```
[root@servera ~]# tail /etc/group
...output omitted...
operators:x:30000:
admin:x:30001:
```

6. Убедитесь, что пользователи **operator1**, **operator2** и **operator3** принадлежат к группе **operators**.

6.1. Добавьтек группе **operators** учётные записи **operator1**, **operator2** и **operator3**.

```
[root@servera ~]# usermod -aG operators operator1
[root@servera ~]# usermod -aG operators operator2
[root@servera ~]# usermod -aG operators operator3
```

6.2. Подтвердите, что пользователи успешно добавлены в группу.

```
[root@servera ~]# id operator1
uid=1001(operator1) gid=1001(operator1) groups=1001(operator1),30000(operators)
[root@servera ~]# id operator2
uid=1002(operator2) gid=1002(operator2) groups=1002(operator2),30000(operators)
[root@servera ~]# id operator3
uid=1003(operator3) gid=1003(operator3) groups=1003(operator3),30000(operators)
```

7. Убедитесь, что пользователи **sysadmin1**, **sysadmin2** и **sysadmin3** принадлежат к группе **admin**. Включите права администратора для всех членов группы **admin**. Убедитесь, что любой член администратора может запускать административные команды.

7.1. Добавьте **sysadmin1**, **sysadmin2** и **sysadmin3** к группе **admim**.

```
[root@servera ~]# usermod -aG admin sysadmin1
[root@servera ~]# usermod -aG admin sysadmin2
[root@servera ~]# usermod -aG admin sysadmin3
```

7.2. Подтвердите, что пользователи успешно добавлены в группу.

```
[root@servera ~]# id sysadmin1
uid=1004(sysadmin1) gid=1004(sysadmin1) groups=1004(sysadmin1),30001(admin)
```

```
[root@servera ~]# id sysadmin2  
uid=1005(sysadmin2) gid=1005(sysadmin2) groups=1005(sysadmin2),30001(admin)  
[root@servera ~]# id sysadmin3  
uid=1006(sysadmin3) gid=1006(sysadmin3) groups=1006(sysadmin3),30001(admin)
```

7.3. Изучите **/etc/group**, чтобы проверить членство в дополнительных группах.

```
[root@servera ~]# tail /etc/group  
...output omitted...  
operators:x:30000:operator1,operator2,operator3  
admin:x:30001:sysadmin1,sysadmin2,sysadmin3
```

7.4. Создайте файл **/etc/sudoers.d/admin**, чтобы члены **admin** имели полные административные привилегии.

```
[root@servera ~]# echo "%admin ALL=(ALL) ALL" >> /etc/sudoers.d/admin
```

7.5. Переключитесь на учётную запись **sysadmin1** (член **admin**) и убедитесь, что вы можете запустить команду **sudo** как пользователь **sysadmin1**.

```
[root@servera ~]# su - sysadmin1  
[sysadmin1@servera ~]$ sudo cat /etc/sudoers.d/admin  
[sudo] password for sysadmin1:redhat  
%admin ALL=(ALL) ALL
```

7.6. Выйдите из оболочки пользователя **sysadmin1**, чтобы вернуться в оболочку пользователя **root**.

```
[sysadmin1@servera ~]$ exit  
logout  
[root@servera ~]#
```

7.7. Выйдите из оболочки пользователя **root**, чтобы вернуться в оболочку пользователя **student**.

```
[root@servera ~]# exit  
logout  
[student@servera ~]$
```

7.8. Выйти из сервера.

```
[student@servera ~]$ exit  
logout  
Connection to servera closed.  
[student@workstation ~]$
```

Завершение

На рабочей станции запустите **lab users-group-manage finish**, чтобы завершить это упражнение. Данные сценарий обеспечивает очистку среды.

```
[student@workstation ~]$ labusers-group-managefinish
```

На этом пошаговое упражнение завершено.

УПРАВЛЕНИЕ ПАРОЛЯМИ ПОЛЬЗОВАТЕЛЕЙ

ЗАДАЧИ

После завершения этого раздела вы сможете установить политику управления паролями для пользователей, а также вручную блокировать и разблокировать учетные записи пользователей.

ТЕНЕВЫЕ ПАРОЛИ И ПАРОЛЬНАЯ ПОЛИТИКА

Когда-то зашифрованные пароли хранились в легко читаемом файле `/etc/passwd`. Это считалось достаточно безопасным, пока словарные атаки на зашифрованные пароли не стали обычным явлением.

На этом этапе зашифрованные пароли были перемещены в отдельный файл `/etc/shadow`, доступный для чтения только `root`. Новый файл также позволил реализовать функции устаревания и истечения срока действия пароля.

Как и `/etc/passwd`, у каждого пользователя есть строка в файле `/etc/shadow`. Пример строки из `/etc/shadow` с девятью полями, разделенными двоеточиями, показан ниже.

```
❶ user03:❷ $6$CSsX...output omitted...:❸ 17933:❹ 0:❺ 99999:❻ 7:❼ 2:❽ 18113:❾
```

- ❶ Имя пользователя учетной записи, которой принадлежит этот пароль.
- ❷ Зашифрованный пароль пользователя. Формат зашифрованных паролей обсуждается далее в этом разделе.
- ❸ День последней смены пароля. Он устанавливается в днях с 01.01.1970 и рассчитывается в часовом поясе UTC.
- ❹ Минимальное количество дней, которое должно пройти с момента последней смены пароля, прежде чем пользователь сможет изменить его снова.
- ❺ Максимальное количество дней, которое может пройти без смены пароля до истечения срока действия пароля. Пустое поле означает, что срок его действия не истекает в зависимости от времени, прошедшего с момента последнего изменения.
- ❻ Период предупреждения. Пользователь будет предупрежден об истечении срока действия пароля, когда он войдет в систему за это количество дней до крайнего срока.
- ❼ Период бездействия. Когда срок действия пароля истечет, он все равно будет приниматься для входа в систему в течение этого количества дней. По истечении этого периода учетная запись будет заблокирована.
- ❽ День истечения срока действия пароля. Он устанавливается в днях с 01.01.1970 и рассчитывается в часовом поясе UTC. Пустое поле означает, что срок его действия не истекает в конкретную дату.
- ❾ Последнее поле обычно пустое и зарезервировано для использования в будущем.

Формат зашифрованного пароля

В поле зашифрованного пароля хранятся три части информации: используемый **алгоритм хеширования** (*hashing algorithm*), **модификатор входа хэш-функции** (*salt*) и зашифрованный хеш (*encrypted hash*). Каждый фрагмент информации отделяется знаком `$`.

```
$① 6$② CSsXcYG1L/4ZfHr/$③ 2W6evvJahUfzfHpc9X.45Jc6H30E...output omitted...
```

1. Алгоритм хеширования, используемый для этого пароля. Число **6** указывает на то, что это хэш **SHA-512**, который используется по умолчанию в Red Hat Enterprise Linux 8.1 означает **MD5**, а **5** - **SHA-256**.
2. **salt**, используемая для шифрования пароля. Изначально он выбирается случайным образом.
3. Зашифрованный хеш пароля пользователя. **salt** и незашифрованный пароль объединяются и зашифровываются для генерации зашифрованного хэша пароля.

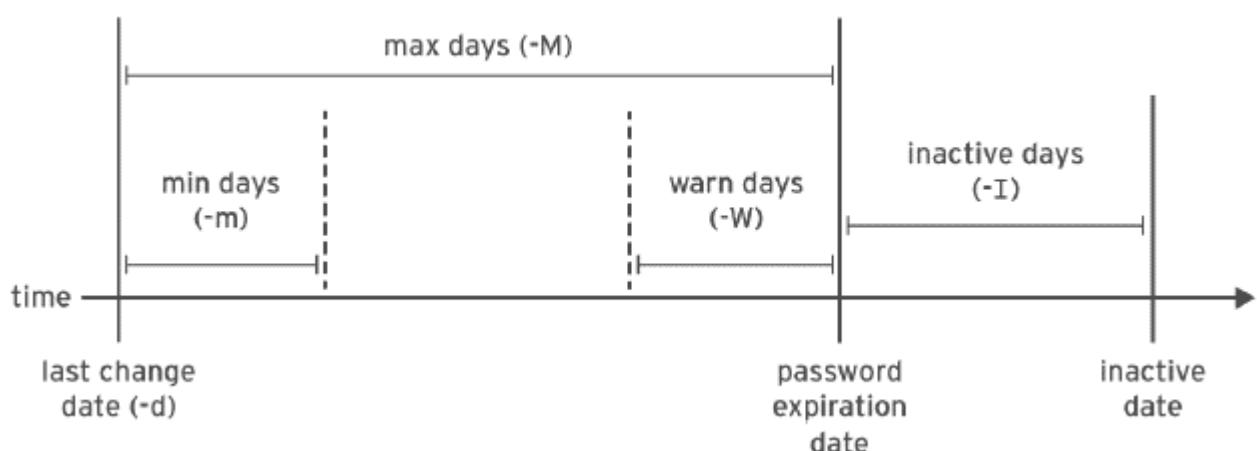
Использование соли (**модификатора входа хэш-функции**) не позволяет двум пользователям с одинаковым паролем иметь одинаковые записи в файле **/etc/shadow**. Например, даже если **user01** и **user02** оба используют **словоoredhat** в качестве своих паролей, их зашифрованные пароли в **/etc/shadow** будут разными, если их соли будут разными.

Подтверждение пароля

Когда пользователь пытается войти в систему, система ищет запись для пользователя в файле **/etc/shadow**, объединяет соль для пользователя с введенным незашифрованным паролем и шифрует их, используя указанный алгоритм хеширования. Если результат совпадает с зашифрованным хешем, пользователь ввел правильный пароль. Если результат не совпадает с зашифрованным хешем, пользователь ввел неправильный пароль и попытка входа в систему не удалась. Этот метод позволяет системе определить, ввел ли пользователь правильный пароль, не сохраняя этот пароль в форме, пригодной для входа в систему.

НАСТРОЙКА СТАРЕНИЯ ПАРОЛЯ

На следующей диаграмме показаны соответствующие параметры устаревания пароля, которые можно настроить с помощью команды **chage** для реализации политики устаревания пароля.



```
[user01@host ~]$ sudo chage -m 0 -M 90 -W 7 -I 14 user03
```

Предыдущая команда **chage** использует параметры **-m**, **-M**, **-W** и **-I** для установки минимального срока действия, максимального срока действия, периода предупреждения и периода бездействия пароля пользователя соответственно.

Команда **chage -d 0 user03** заставляет пользователя **user03** обновить свой пароль при следующем входе в систему.

Команда **chage -l user03** отображает сведения об устаревании пароля пользователя **user03**.

Команда **chage -E 2019-08-05 user03** приводит к истечению срока действия учетной записи пользователя **user03 05.08.2019** (в формате ГГГГ-ММ-ДД).



ПРИМЕЧАНИЕ

Команду **date** можно использовать для вычисления даты в будущем. Параметр **-u** сообщает время в формате UTC.

```
[user01@host ~]$ date -d "+45 days" -u  
Thu May 23 17:01:20 UTC 2019
```

Отредактируйте элементы конфигурации устаревания пароля в файле **/etc/login.defs**, чтобы установить политики устаревания пароля по умолчанию. **PASS_MAX_DAYS** устанавливает максимальный срок действия пароля по умолчанию. **PASS_MIN_DAYS** устанавливает минимальный возраст пароля по умолчанию. **PASS_WARN_AGE** устанавливает период предупреждения пароля по умолчанию. Любое изменение политики устаревания пароля по умолчанию будет действовать только для новых пользователей. Существующие пользователи продолжат использовать старые настройки устаревания пароля, а не новые.

ОГРАНИЧЕНИЕ ДОСТУПА

Вы можете использовать команду **chage** для установки даты истечения срока действия учетной записи. По достижении этой даты пользователь не может войти в систему в интерактивном режиме. Команда **usermod** может заблокировать учетную запись с помощью опции **-L**.

```
[user01@host ~]$ sudo usermod -L user03  
[user01@host ~]$ su - user03  
Password: redhat  
su: Authentication failure
```

Если пользователь покидает компанию, администратор может заблокировать и удалить учетную запись с помощью одной команды **usermod**. Дата должна быть указана в виде количества дней с **01.01.1970** или в формате ГГГГ-ММ-ДД (**YYYY-MM-DD**).

```
[user01@host ~]$ sudo usermod -L -e 2019-10-05 user03
```

Предыдущая команда **usermod** использует параметр **-e**, чтобы установить дату истечения срока действия учетной записи для данной учетной записи пользователя. Параметр **-L** блокирует пароль пользователя.

Блокировка учетной записи предотвращает аутентификацию пользователя с паролем в системе. Это рекомендуемый метод предотвращения доступа к учетной записи сотрудника, уволившегося из компании. Если сотрудник вернется, учетную запись позже можно будет разблокировать с помощью **usermod -U**. Если срок действия учетной записи также истек, не забудьте также изменить дату истечения срока действия.

nologin Shell

Оболочка **nologin** Shell действует как замещающая оболочка для учетных записей пользователей, не предназначенных для интерактивного входа в систему. С точки зрения безопасности разумно отключить учетную запись пользователя от входа в систему, когда учетная запись пользователя выполняет обязанности, не требующие от пользователя входа в систему. Например, почтовый сервер может потребовать учетную запись для хранения почты и пароль для аутентификации пользователя в почтовом клиенте, используемом для получения почты. Этому пользователю не нужно входить в систему напрямую.

Распространенным решением этой ситуации является установка оболочки входа пользователя в **/sbin/nologin**. Если пользователь пытается войти в систему напрямую, оболочка **nologin** закрывает соединение.

```
[user01@host ~]$ usermod -s /sbin/nologin user03
```

```
[user01@host ~]$ su - user03
```

```
Last login: Wed Feb 6 17:03:06 IST 2019 on pts/0
```

```
This account is currently not available.
```



ВАЖНО

Оболочка **nologin** предотвращает интерактивное использование системы, но не предотвращает любой доступ. Пользователи могут иметь возможность аутентифицировать и загружать или извлекать файлы через такие приложения, как веб-приложения, программы передачи файлов или программы чтения почты, если они используют пароль пользователя для аутентификации.



РЕКОМЕНДАЦИИ

Справочные страницы **mchange(1)**, **usermod(8)**, **shadow(5)**, **crypt(3)**.

УПРАЖНЕНИЯ С ПОШАГОВЫМИ ИНСТРУКЦИЯМИ

В этом упражнении вы установите политики паролей для нескольких пользователей.

В РЕЗУЛЬТАТЕ

Вы будете способны:

- Принудительно сменить пароль при первом входе пользователя в систему.
- Принудительно менять пароль каждые 90 дней.
- Установите срок действия учетной записи на 180 дней с текущего дня.

ПРЕЖДЕ ЧЕМ ВЫ НАЧНЕТЕ

Войдите на рабочую станцию как пользователь **student**, используя в качестве пароля слово **student**.

На рабочей станции запустите команду **lab users-pw-manage start** чтобы начать упражнение. Данный сценарий обеспечивает правильную настройку среды для выполнения упражнения.

```
[student@workstation ~]$ labusers-pw-managestart
```

1. С рабочей станции откройте сеанс **SSH** на сервере как пользователь **student**.

```
[student@workstation ~]$ ssh student@servera  
...output omitted...  
[student@servera ~]$
```

2. На сервере изучите блокировку и разблокировку учетных записей пользователей в качестве пользователя **student**.

- 2.1. Будучи пользователем **student**, заблокируйте учетную запись **operator1** с правами администратора.

```
[student@servera ~]$ sudo usermod -L operator1  
[sudo] password for student
```

- 2.2. Попытка войти в систему как **operator1**. Попытка должна завершиться неудачей.

```
[student@servera ~]$ su - operator1  
Password: redhat  
su: Authentication failure
```

2.3. Разблокируйте аккаунт **operator1**.

```
[student@servera ~]$ sudo usermod -U operator1
```

2.4. Попытайтесь снова войти в систему как **operator1**. Действие будет успешным.

```
[student@servera ~]$ su - operator1  
Password: redhat  
...output omitted...  
[operator1@servera ~]$
```

2.5. Выходите из оболочки пользователя **operator1**, чтобы вернуться в оболочку пользователя **student**.

```
[operator1@servera ~]$ exit  
Logout
```

3. Измените политику паролей для **operator1**, чтобы он запрашивал новый пароль каждые 90 дней. Подтвердите, что срок действия пароля установлен успешно.

3.1. Установите максимальный срок действия пароля пользователя **operator1** на 90 дней.

```
[student@servera ~]$ sudo chage -M 90 operator1
```

3.2. Убедитесь, что срок действия пароля пользователя **operator1** истекает через 90 дней после его изменения.

```
[student@servera ~]$ sudo chage -l operator1  
Last password change      : Jan 25, 2019  
Password expires          : Apr 25, 2019  
Password inactive         : never  
Account expires           : never  
Minimum number of days between password change : 0  
Maximum number of days between password change : 90  
Number of days of warning before password expires : 7
```

4. Установить принудительную смену пароля при первом входе в учетную запись **operator1**.

```
[student@servera ~]$ sudo chage -d 0 operator1
```

5. Войдите в систему как **operator1** и измените пароль на **forsooth123**. После установки пароля вернитесь в оболочку пользователя **student**.

5.1. Войдите в систему как **operator1** и при появлении запроса измените пароль на **forsooth123**.

```
[student@servera ~]$ su - operator1
Password: redhat
You are required to change your password immediately (administrator enforced)
Current password: redhat
New password: forsooth123
Retype new password: forsooth123
...output omitted...
[operator1@servera ~]$
```

5.2. Выйдите из оболочки пользователя operator1, чтобы вернуться в оболочку пользователя **student**.

```
[operator1@servera ~]$ exit
Logout
```

6. Установите срок действия учетной записи **operator1** на **180** дней с текущего дня. Подсказка: **date -d "+180 days"**дает вам дату и время через 180 дней от текущей даты и времени.

6.1. Определите дату на 180 дней в будущем. Используйте формат **%F** с командой **date**, чтобы получить точное значение.

```
[student@servera ~]$ date -d "+180 days" +%F
2019-07-24
```

Вы можете получить другое значение для использования на следующем шаге в зависимости от текущей даты и времени в вашей системе.

6.2. Установите срок действия учетной записи на дату, указанную на предыдущем шаге.

```
[student@servera ~]$ sudo chage -E 2019-07-24 operator1
```

6.3. Убедитесь, что дата истечения срока действия учетной записи успешно установлена.

```
[student@servera ~]$ sudo chage -l operator1
```

```
Last password change      : Jan 25, 2019
Password expires          : Apr 25, 2019
Password inactive         : never
Account expires           : Jul 24, 2019
Minimum number of days between password change   : 0
Maximum number of days between password change  : 90
Number of days of warning before password expires : 7
```

7. Установите срок действия паролей на 180 дней с текущей даты для всех пользователей. Используйте права администратора для редактирования файла конфигурации.

- 7.1. Установите **PASS_MAX_DAYS** на **180** в **/etc/login.defs**. Используйте права администратора при открытии файла в текстовом редакторе. Вы можете использовать команду **sudo vim /etc/login.defs** для выполнения этого шага.

```
...output omitted...
# Password aging controls:
#
#      PASS_MAX_DAYS    Maximum number of days a password may be
#      used.
#      PASS_MIN_DAYS    Minimum number of days allowed between
#      password changes.
#      PASS_MIN_LEN     Minimum acceptable password length.
#      PASS_WARN_AGE    Number of days warning given before a
#      password expires.
#
PASS_MAX_DAYS    180
PASS_MIN_DAYS    0
PASS_MIN_LEN     5|
PASS_WARN_AGE    7
...output omitted...
```



ВАЖНО

Настройки пароля и срока действия учетной записи по умолчанию будут действовать для новых пользователей, но не для существующих.

- 7.2. Выйти из сервера.

```
[student@servera ~]$ exit
logout
Connection to servera closed.
[student@workstation ~]$
```

Завершение

На рабочей станции запустите **lab users-pw-manage finish**, чтобы завершить это упражнение. Данные сценарий обеспечивает очистку среды.

```
[student@workstation ~]$ labusers-pw-managefinish
```

На этом пошаговое упражнение завершено.

ЛАБОРАТОРНАЯ РАБОТА

УПРАВЛЕНИЕ ЛОКАЛЬНЫМИ ПОЛЬЗОВАТЕЛЯМИ И ГРУППАМИ

КОНТРОЛЬНЫЙ СПИСОК РАБОТЫ

В этой лабораторной работе вы установите локальную политику паролей по умолчанию, создадите дополнительную группу для трех пользователей, разрешите этой группе использовать **sudo** для выполнения команд от имени пользователя **root** и измените политику паролей для одного пользователя.

В РЕЗУЛЬТАТЕ

Вы будете способны:

- Установить политику устаревания пароля по умолчанию для пароля локального пользователя.
- Создать группу и используйте ее как дополнительную группу для новых пользователей.
- Создать трех новых пользователей с новой группой в качестве дополнительной группы.
- Настройте членов дополнительной группы для выполнения любой команды от имени любого пользователя, использующего **sudo**.
- Установите политику устаревания пароля для конкретного пользователя.

ПРЕЖДЕ ЧЕМ ВЫ НАЧНЕТЕ

Войдите на рабочую станцию как пользователь **student**, используя в качестве пароля слово **student**.

На рабочей станции запустите команду **lab users-review start**, чтобы начать упражнение. Данный сценарий обеспечивает правильную настройку среды для выполнения упражнения.

```
[student@workstation ~]$ labusers-reviewstart
```

1. С рабочей станции откройте сеанс SSH на **serverb** как пользователь **student**.
2. На **serverb** убедитесь, что у вновь созданных пользователей есть пароли, которые необходимо менять каждые 30 дней.
3. Создайте группу **consultants** с **GID 40000**.
4. Настройте административные права для всех членов группы **consultants**, чтобы они могли выполнять любую команду от имени любого пользователя.
5. Создайте пользователей **consultant1**, **consultant2**, и **consultant3** в качестве их дополнительной группы задайте группу **consultants**.
6. Установите для учетных записей **consultant1**, **consultant2**, и **consultant3** срок действия до 90 дней с текущего дня.
7. Измените политику паролей для учетной записи **consultant2**, чтобы новый пароль запрашивался каждые 15 дней.

8. Кроме того, принудительно заставьте пользователей к **consultant1**, **consultant2**, и **consultant3** изменить свои пароли при первом входе в систему.

Оценка

На рабочей станции выполните команду **labusers-reviewgrade**, чтобы подтвердить успех этого упражнения.

```
[student@workstation ~]$ lab users-review grade
```

Завершение

На рабочей станции запустите **lab users-review finish**, чтобы завершить лабораторную работу. Данный сценарий удаляет учетные записи пользователей и файлы, созданные в лабораторной работе, чтобы обеспечить чистоту среды.

```
[student@workstation ~]$ lab users-review finish
```

На этом лабораторная работа завершена.

РЕШЕНИЕ

УПРАВЛЕНИЕ ЛОКАЛЬНЫМИ ПОЛЬЗОВАТЕЛЯМИ И ГРУППАМИ

КОНТРОЛЬНЫЙ СПИСОК РАБОТЫ

В этой лабораторной работе вы установите локальную политику паролей по умолчанию, создадите дополнительную группу для трех пользователей, разрешите этой группе использовать **sudo** для выполнения команд от имени пользователя **root** и измените политику паролей для одного пользователя.

В РЕЗУЛЬТАТЕ

Вы будете способны:

- Установить политику устаревания пароля по умолчанию для пароля локального пользователя.
- Создать группу и используйте ее как дополнительную группу для новых пользователей.
- Создать трех новых пользователей с новой группой в качестве дополнительной группы.
- Настройте членов дополнительной группы для выполнения любой команды от имени любого пользователя, использующего **sudo**.
- Установите политику устаревания пароля для конкретного пользователя.

ПРЕЖДЕ ЧЕМ ВЫ НАЧНЕТЕ

Войдите на рабочую станцию как пользователь **student**, используя в качестве пароля слово **student**.

На рабочей станции запустите команду **lab users-review start**, чтобы начать упражнение. Данный сценарий обеспечивает правильную настройку среды для выполнения упражнения.

```
[student@workstation ~]$ labusers-reviewstart
```

1. С рабочей станции откройте сеанс **SSH** на **serverb** как пользователь **student**.

```
[student@workstation ~]$ ssh student@serverb  
...output omitted...  
[student@serverb ~]$
```

2. На **serverb** убедитесь, что у вновь созданных пользователей есть пароли, которые необходимо менять каждые 30 дней.

- 2.1. Установите для **PASS_MAX_DAYS** значение **30** в **/etc/login.defs**. Используйте права администратора при открытии файла в текстовом редакторе. Вы можете использовать команду **sudo vim /etc/login.defs** для выполнения этого шага. Используйте слово **student** в качестве пароля, когда **sudo** предлагает вам ввести пароль пользователя **student**.

```
...output omitted...
# Password aging controls:
#
#      PASS_MAX_DAYS    Maximum number of days a password may be
#      used.
#      PASS_MIN_DAYS    Minimum number of days allowed between
#      password changes.
#      PASS_MIN_LEN     Minimum acceptable password length.
#      PASS_WARN_AGE    Number of days warning given before a
#      password expires.
#
PASS_MAX_DAYS    30
PASS_MIN_DAYS    0
PASS_MIN_LEN     5
PASS_WARN_AGE    7
...output omitted...
```

3. Создайте группу **consultants** с **GID 40000**.

```
[student@serverb ~]$ sudo groupadd -g 40000 consultants
```

4. Настройте административные права для всех членов группы **consultants**, чтобы они могли выполнять любую команду от имени любого пользователя.

- 4.1. Создайте новый файл **/etc/sudoers.d/consultants** и добавьте в него следующее содержимое. Вы можете использовать команду **sudo vim /etc/sudoers.d/consultants** для выполнения этого шага.

```
%consultants ALL=(ALL) ALL
```

5. Создайте пользователей **consultant1**, **consultant2**, и **consultant3** в качестве их дополнительной группы задайте группу **consultants**.

```
[student@serverb ~]$ sudo useradd -G consultants consultant1
[student@serverb ~]$ sudo useradd -G consultants consultant2
[student@serverb ~]$ sudo useradd -G consultants consultant3
```

6. Установите для учетных записей **consultant1**, **consultant2**, и **consultant3** срок действия до 90 дней с текущего дня.

- 6.1. Определите дату на 90 дней вперед. Вы можете получить другое значение по сравнению со следующим выводом в зависимости от текущей даты и времени вашей системы.

```
[student@serverb ~]$ date -d "+90 days" +%F  
2019-04-28
```

- 6.2. Установите дату истечения срока действия учетной записи **consultant1**, **consultant2**, и **consultant3** на такое же значение, как определено на предыдущем шаге.

```
[student@serverb ~]$ sudo chage -E 2019-04-28 consultant1  
...output omitted...  
[student@serverb ~]$ sudo chage -E 2019-04-28 consultant2  
...output omitted...  
[student@serverb ~]$ sudo chage -E 2019-04-28 consultant3  
...output omitted...
```

7. Измените политику паролей для учетной записи **consultant2**, чтобы новый пароль запрашивался каждые 15 дней.

```
[student@serverb ~]$ sudo chage -M 15 consultant2  
...output omitted...
```

8. Кроме того, принудительно заставьте пользователей к **consultant1**, **consultant2**, и **consultant3** изменить свои пароли при первом входе в систему.

- 8.1. Установите последний день смены пароля равным **0**, чтобы пользователи были вынуждены менять пароль всякий раз, когда они входят в систему в первый раз.

```
[student@serverb ~]$ sudo chage -d 0 consultant1  
...output omitted...  
[student@serverb ~]$ sudo chage -d 0 consultant2  
...output omitted...  
[student@serverb ~]$ sudo chage -d 0 consultant3  
...output omitted...
```

- 8.2. Выйти с **serverb**.

```
[student@serverb ~]$ exit  
logout  
Connection to serverb closed.
```

Оценка

На рабочей станции выполните команду **labusers-reviewgrade**, чтобы подтвердить успех этого упражнения.

```
[student@workstation ~]$ lab users-review grade
```

Завершение

На рабочей станции запустите **lab users-review finish**, чтобы завершить лабораторную работу. Данный сценарий удаляет учетные записи пользователей и файлы, созданные в лабораторной работе, чтобы обеспечить чистоту среды.

```
[student@workstation ~]$ lab users-review finish
```

На этом лабораторная работа завершена.

РЕЗЮМЕ

В этой главе вы узнали:

- Существует три основных типа учетных записей пользователей: суперпользователь, системные пользователи и обычные пользователи.
- Пользователь должен иметь основную группу и может быть членом одной или нескольких дополнительных групп.
- Три важных файла, содержащих информацию о пользователях и группах, - это **/etc/passwd**, **/etc/group**, и **/etc/shadow**.
- Команды **su** и **sudo** могут использоваться для запуска команд от имени суперпользователя.
- Команды **useradd**, **usermod** и **userdel** могут использоваться для управления пользователями.
- Команды **groupadd**, **groupmod** и **groupdel** могут использоваться для управления группами.
- Команду **chage** можно использовать для настройки и просмотра параметров истечения срока действия пароля для пользователей.

ГЛАВА 7

КОНТРОЛЬ ДОСТУПА К ФАЙЛАМ

ЦЕЛЬ

Установите разрешения файловой системы Linux для файлов и интерпретируйте влияние различных настроек разрешений на безопасность.

ЗАДАЧИ

- Составьте список разрешений файловой системы для файлов и каталогов и интерпретируйте влияние этих разрешений на доступ пользователей и групп.
- Измените права доступа и владельца файлов с помощью инструментов командной строки.
- Управляйте разрешениями по умолчанию для новых файлов, созданных пользователями, объясните действие специальных разрешений и используйте особые разрешения и разрешения по умолчанию, чтобы установить владельца группы файлов, созданных в определенном каталоге.

РАЗДЕЛЫ

- Интерпретация разрешений файловой системы Linux (иконтрольный опрос)
- Управление разрешениями файловой системы из командной строки (и упражнения с пошаговыми инструкциями)
- Управление разрешениями по умолчанию и доступом к файлам (и упражнения с пошаговыми инструкциями)

ЛАБОРОТОРНАЯ РАБОТА

Контроль доступа к файлам

ИНТЕРПРЕТАЦИЯ РАЗРЕШЕНИЙ ФАЙЛОВОЙ СИСТЕМЫ LINUX

ЗАДАЧИ

После завершения этого раздела вы сможете составить список разрешений файловой системы для файлов и каталогов и интерпретировать влияние этих разрешений на доступ пользователей и групп.

РАЗРЕШЕНИЯ ДЛЯ ФАЙЛОВОЙ СИСТЕМЫ LINUX

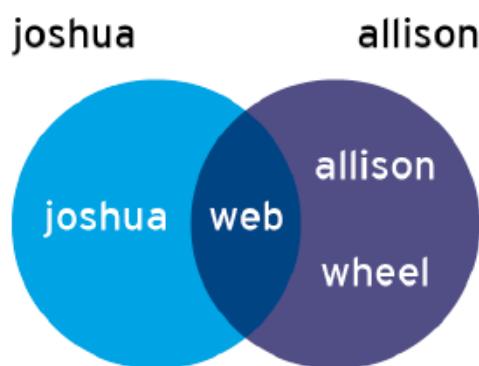
Права доступа к файлам (*File permissions*) контролируют доступ к файлам. Система разрешений для файлов Linux проста, но гибка, что упрощает ее понимание и применение, но при этом позволяет легко обрабатывать большинство обычных разрешений.

У файлов есть три категории пользователей, к которым применяются разрешения. Файл принадлежит пользователю (**user**), обычно тому, кто его создал. Файл также принадлежит одной группе (**group**), обычно основной группе пользователя, создавшего файл, но это можно изменить. Для пользователя-владельца, группы-владельца и для всех других (**other**) пользователей в системе, которые не являются пользователем или членом группы-владельца, могут быть установлены разные разрешения.

Конкретные разрешения имеют разный приоритет. Разрешения пользователей (**user**) имеют приоритет над разрешениями группы (**group**), которые имеют приоритет над категорией другими (**other**) разрешениями.

На рис. 7.1 **joshua** является членом групп **joshua** и **web**, а **allison** - членом групп **allison**, **wheel** и **web**. Когда Джошуа и Эллисон должны сотрудничать, файлы должны быть связаны группой **web**, разрешения которой дают желаемый доступ.

Рисунок 7.1: Пример членства в группе для облегчения совместной работы



Применяются три категории разрешений: чтение, запись и выполнение. В следующей таблице объясняется, как эти разрешения влияют на доступ к файлам и каталогам.

Влияние разрешений на файлы и каталоги

РАЗРЕШЕНИЕ	ВЛИЯНИЕ НА ФАЙЛЫ	ВЛИЯНИЕ НА КАТАЛОГИ
------------	------------------	---------------------

г (read)	Можно прочитать содержимое файла.	Можно перечислить содержимое каталога (имена файлов).
w (write)	Содержимое файла можно изменить.	Любой файл в каталоге можно создать или удалить.
х (execute)	Файлы могут выполняться как команды.	Доступ к содержимому каталога. (Вы можете перейти в каталог, прочитать информацию о его файлах и получить доступ к его файлам, если разрешения файлов позволяют это.)

Обратите внимание, что пользователи обычно имеют разрешения на чтение и выполнение в каталогах только для чтения, так что они могут перечислить содержимое каталога и иметь полный доступ только для чтения к его содержимому. Если у пользователя есть доступ только для чтения к каталогу, имена файлов в нем могут быть перечислены, но никакая другая информация, включая разрешения или временные метки, недоступна, и к ним нельзя получить доступ. Если у пользователя есть доступ только на выполнение к каталогу, он не может получить список имён файлов в каталоге, но если он уже знает имя файла, на чтение которого у него есть разрешение, то он может получить доступ к содержимому этого файла с помощью явное указание имени файла.

Файл может быть удален любым, у кого есть разрешение на запись в каталог, в котором он находится, независимо от владельца или разрешений на сам файл. Это можно изменить с помощью специального разрешения, липкого (*sticky*) бита, который обсуждается далее в этой главе.



ПРИМЕЧАНИЕ

Права доступа к файлам Linux работают иначе, чем система разрешений, используемая файловой системой NTFS для Microsoft Windows.

В Linux разрешения применяются только к файлу или каталогу, в котором они установлены. То есть разрешения на каталог не наследуются автоматически подкаталогами и файлами в нем. Однако разрешения для каталога могут блокировать доступ к содержимому каталога в зависимости от того, насколько они ограничены.

Разрешение на чтение для каталога в Linux примерно эквивалентно списку содержимого папки в Windows.

Разрешение на запись в каталог в Linux эквивалентно изменить в Windows; он подразумевает возможность удалять файлы и подкаталоги. В Linux, если для каталога установлены и запись, *isticky* бит, то только пользователь, владеющий файлом или подкаталогом в каталоге, может удалить его, что близко к поведению разрешения Windows на запись.

Пользователь **root** в Linux имеет эквивалент разрешения Windows Full Control для всех файлов. Однако **root** может по-прежнему иметь доступ, ограниченный системной политикой **SELinux** и контекстом безопасности рассматриваемого процесса и файлов. **SELinux** будет обсуждаться позже.

ПРОСМОТР РАЗРЕШЕНИЙ И ВЛАДЕНИЯ НА ФАЙЛ И КАТАЛОГ

Параметр **-l** команды **ls** показывает более подробную информацию о разрешениях и владельцах файлов:

```
[user@host~]$ ls -l test
-rw-rw-r--. 1 student student 0 Feb 8 17:36 test
```

Вы можете использовать опцию **-d**, чтобы показать подробную информацию о самом каталоге, а не о его содержимом.

```
[user@host ~]$ ls -ld /home
drwxr-xr-x. 5 root root 4096 Jan 31 22:00 /home
```

Первый символ в длинном списке - это тип файла. Вы интерпретируете это так:

- **-** это обычный файл.
- **d** - это каталог.
- **l** - это программная ссылка.
- Остальные символы обозначают аппаратные устройства (**b** и **c**) или другие файлы специального назначения (**p** и **s**).

Следующие девять символов - это права доступа к файлу. Они представлены тремя наборами по три символа: разрешения, которые применяются к пользователю, которому принадлежит файл, группе, которой принадлежит файл, и всем остальным пользователям. Если набор показывает **rwx**, эта категория имеет все три разрешения: чтение, запись и выполнение. Если буква была заменена на **(-)**, то эта категория не имеет разрешения.

После подсчета ссылок первое имя указывает владельца, которому принадлежит файл, а второе имя - группу, владеющую файлом.

Итак, в приведенном выше примере разрешения для пользователя **student** указаны первым набором из трех символов. Пользователь **student** может читать и изменять файл **test**, но не выполнил его.

Группа **student** определяется вторым набором из трех символов: он также имеет права на чтение и запись в файл **test**, но не на его выполнение.

Разрешения для (**other**) любого другого пользователя определяются третьим набором из трех символов: у них есть только разрешение на чтение для файла **test**.

Применяется наиболее конкретный набор разрешений. Таким образом, если пользователь **student** имеет права, отличные от разрешений группы **student**, и пользователь **student** также является членом этой группы, тогда применимы будут разрешения пользователя.

ПРИМЕРЫ РАЗРЕШЕНИЯ ДЕЙСТВИЯ

Следующие примеры помогут проиллюстрировать, как взаимодействуют права доступа к файлам. Для этих примеров у нас есть четыре пользователя со следующим членством в группах:

ПОЛЬЗОВАТЕЛЬ	ЧЛЕНСТВО В ГРУППЕ
operator1	operator1, consultant1
database1	database1, consultant1
database2	database2, operator2
contractor1	contractor1, operator2

Эти пользователи будут работать с файлами в каталоге **dir**. Далее показан полный список файлов в этом каталоге:

```
[database1@host dir]$ ls -la
total 24
drwxrwxr-x.  2 database1 consultant1  4096 Apr  4 10:23 .
drwxr-xr-x. 10 root      root       4096 Apr  1 17:34 ..
-rw-rw-r--.  1 operator1 operator1   1024 Apr  4 11:02 lfile1
-rw-r--rw-.  1 operator1 consultant1 3144 Apr  4 11:02 lfile2
-rw-rw-r--.  1 database1 consultant1 10234 Apr  4 10:14 rfile1
-rw-r-----.  1 database1 consultant1  2048 Apr  4 10:18 rfile2
```

Параметр **-a** показывает разрешения для скрытых файлов, включая специальные файлы, используемые для представления каталога и его родительского элемента. В этом примере. отражает разрешения самого каталога и .. разрешения его родительского каталога.

Какие разрешения у файла **rfile1**? Пользователь, владеющий файлом (**database1**), может читать и изменять содержимое файла, но не может его выполнять. Группа, которой принадлежит файл (**consultant1**), имеет права чтения и записи, но не на выполнение. Все остальные пользователи могут прочитать файл, но не редактировать (записывать) или выполнять его.

В следующей таблице рассматриваются некоторые эффекты этого набора разрешений для указанных пользователей:

ЭФФЕКТ	ПОЧЕМУ ЭТО ИСТИНА?

Пользователь operator1 может изменять содержимое файла lfile1 .	Пользователь operator1 является членом группы consultant1 , и эта группа имеет права на чтение и запись для файла lfile1 .
Пользователь database1 может просматривать и изменять содержимое файла rfile2 .	Пользователь database1 владеет файлом и имеет права на чтение и запись в файл rfile2 .
Пользователь operator1 может просматривать, но не изменять содержимое файла lfile2 (без его удаления и повторного создания).	Пользователь operator1 является членом группы consultant1 , и эта группа имеет доступ только для чтения к файлу lfile2 .
Пользователь database2 и contractor1 не имеют доступа к содержимому файла lfile2 .	Другие (other) разрешения применяются к пользователям database2 и contractor1 , и эти разрешения не включают разрешение на чтение или запись.
operator1 - единственный пользователь, который может изменять содержимое файла lfile1 (без его удаления и повторного создания).	Пользователь и группа operator1 имеют разрешение на запись в файл, другие пользователи - нет. Единственный член группы operator1 - это сам пользователь operator1 .
Пользователь database2 может изменять содержимое файла lfile2 .	Пользователь database2 не является владельцем файла и не входит в группу consultant1 , поэтому применяются разрешения назначенные третьей группе other . Они предоставляют разрешение на запись.
Пользователь database1 может просматривать содержимое файла lfile2 , но не может изменять его содержимое(без его удаления и повторного создания).	Пользователь database1 является членом группы consultant1 , и эта группа имеет разрешения только на чтение для файла lfile2 . Даже если у другого (other) есть разрешение на запись, разрешения группы имеют приоритет.
Пользователь database1 может удалять файлы lfile1 и lfile2 .	Пользователь database1 имеет права на запись в каталог, содержащий оба файла (обозначен символом (.)), и поэтому может удалить любой файл в этом каталоге. Это верно, даже если database1 не имеет разрешения на запись в сам файл.



РЕКОМЕНДАЦИИ

справочная страница `man ls(1)`

infocoreutils(*GNUCoreutils*)

- Раздел 13: Изменение атрибутов файла (Changing file attributes).

КОНТРОЛЬНЫЙ ОПРОС

ИНТЕРПРЕТАЦИЯ РАЗРЕШЕНИЙ ФАЙЛОВОЙ СИСТЕМЫ LINUX

Просмотрите следующую информацию и используйте ее, чтобы ответить на вопросы викторины.

В системе четыре пользователя включены в следующие группы:

- Пользователь **consultant1** входит в группы **consultant1** и **database1**.
- Пользователь **operator1** находится в группах **operator1** и **database1**.
- Пользователь **contractor1** входит в группы **contractor1** и **contractor3**.
- Пользователь **оператор2** находится в группах **оператор2** и **contractor3**.

Текущий каталог (.). Содержит четыре файла со следующей информацией о правах доступа:

```
drwxrwxr-x. operator1 database1 .
-rw-rw-r--. consultant1 consultant1 lfile1
-rw-r--rw-. consultant1 database1 lfile2
-rw-rw-r--. operator1 database1 rfile1
-rw-r-----. operator1 database1 rfile2
```

1. Какой обычный файл принадлежит **operator1** и доступен для чтения всем пользователям?
 - a. lfile1
 - b. lfile2
 - c. rfile1
 - d. rfile2
2. Какой файл может быть изменен пользователем **contractor1**?
 - a. lfile1
 - b. lfile2
 - c. rfile1
 - d. rfile2
3. Какой файл не может прочитать пользователь **operator2**?
 - a. lfile1
 - b. lfile2
 - c. rfile1
 - d. rfile2
4. Какой файл принадлежит группе **consultant1**?
 - a. lfile1

- b. lfile2
- c. rfile1
- d. rfile2

5. Какие файлы может удалить пользователь**operator1**?

- a. rfile1
- b. rfile2
- c. All of the above.
- d. None of the above.

6. Какие файлы может удалить пользователь**operator2**?

- a. lfile1
- b. lfile2
- c. All of the above.
- d. None of the above.

РЕШЕНИЕ

ИНТЕРПРЕТАЦИЯ РАЗРЕШЕНИЙ ФАЙЛОВОЙ СИСТЕМЫ LINUX

Просмотрите следующую информацию и используйте ее, чтобы ответить на вопросы викторины.

В системе четыре пользователя включены в следующие группы:

- Пользователь **consultant1** входит в группы **consultant1** и **database1**.
- Пользователь **operator1** находится в группах **operator1** и **database1**.
- Пользователь **contractor1** входит в группы **contractor1** и **contractor3**.
- Пользователь **оператор2** находится в группах **оператор2** и **contractor3**.

Текущий каталог (.). Содержит четыре файла со следующей информацией о правах доступа:

```
drwxrwxr-x. operator1 database1 .
-rw-rw-r--. consultant1 consultant1 lfile1
-rw-r--rw-. consultant1 database1 lfile2
-rw-rw-r--. operator1 database1 rfile1
-rw-r-----. operator1 database1 rfile2
```

- Какой обычный файл принадлежит **operator1** и доступен для чтения всем пользователям?
 - lfile1**
 - lfile2**
 - rfile1**
 - rfile2**
- Какой файл может быть изменен пользователем **contractor1**?
 - lfile1**
 - lfile2**
 - rfile1**
 - rfile2**
- Какой файл не может прочитать пользователь **operator2**?
 - lfile1**
 - lfile2**
 - rfile1**
 - rfile2**
- Какой файл принадлежит группе **consultant1**?
 - lfile1**

- b. lfile2
- c. rfile1
- d. rfile2

5. Какие файлы может удалить пользователь**operator1**?

- e. rfile1
- f. rfile2
- g. All of the above.
- h. None of the above.

6. Какие файлы может удалить пользователь**operator2**?

- e. lfile1
- f. lfile2
- g. All of the above.
- h. None of the above.

УПРАВЛЕНИЕ РАЗРЕШЕНИЯМИ ФАЙЛОВОЙ СИСТЕМЫ ИЗ КОМАНДНОЙ СТРОКИ

ЗАДАЧИ

После завершения этого раздела вы сможете изменить разрешения и права собственности на файлы с помощью инструментов командной строки.

ИЗМЕНЕНИЕ РАЗРЕШЕНИЙ НА ФАЙЛ И КАТАЛОГ

Для изменения разрешений из командной строки используется команда **chmod**, что означает «режим изменения (changemode)» (разрешения также называются режимом файла). Команда **chmod** принимает инструкцию разрешения, за которой следует список файлов или каталогов, которые нужно изменить. Инструкция разрешения может быть выдана либо символически (символьный метод), либо численно (числовой метод).

Изменение разрешений символьным методом

chmod Who What Which file/directory

- Кто **u, g, o, a** (для пользователя, группы, другого, всех)
- Что делает **+, -, =** (для добавления, удаления, точной установки)
- Что определяет **w, x** (для чтения, записи, выполнения)

В символическом методе изменения прав доступа к файлам используются буквы для обозначения различных групп разрешений: **u(user)** для пользователя, **g** для группы (**group**), **o** для прочих (**other**) и **a** для всех (**all**).

При использовании символьного метода нет необходимости устанавливать полностью новую группу разрешений. Вместо этого вы можете изменить одно или несколько существующих разрешений. Используйте **+** или **-**, чтобы добавить или удалить разрешения, соответственно, или используйте **=**, чтобы заменить весь набор для группы разрешений.

Сами разрешения представлены одной буквой: **r** для чтения (**read**), **w** для записи (**write**) и **x** для выполнения (**execute**). При использовании команды **chmod** для изменения разрешений с помощью символьского метода использование заглавной буквы **X** в качестве флага разрешения добавит разрешение на выполнение только в том случае, если файл является каталогом или уже имеет набор выполнения для пользователя, группы или другого.



ПРИМЕЧАНИЕ

Команда **chmod** поддерживает параметр **-R** для рекурсивной установки разрешений для файлов во всем дереве каталогов. При использовании параметра **-R** может быть полезно установить разрешения символически с помощью параметра **X**. Это позволяет установить разрешение на выполнение (поиск) для каталогов, чтобы можно было получить доступ к

их содержимому, без изменения разрешений для большинства файлов. Однако будьте осторожны с опцией **X**, потому что, если для файла установлено какое-либо разрешение на выполнение, **X** также установит указанное разрешение на выполнение для этого файла. Например, следующая команда рекурсивно устанавливает доступ для чтения и записи для **demodir** и всех его дочерних элементов для их владельца группы, но применяет разрешения на выполнение группы только к каталогам и файлам, которые уже имеют набор выполнения для пользователя, группы или другого.

```
[root@host opt]# chmod -R g+rwX demodir
```

Примеры

- Удалить права на чтение и запись для группы и других файлов в **file1**:

```
[user@host ~]$ chmod go-rw file1
```

- Добавить разрешение на выполнение для всех в **file2**:

```
[user@host ~]$ chmod a+x file2
```

Изменение разрешений с помощью числового метода

В приведенном ниже примере символ **#** представляет собой цифру.

```
chmod ### file/directory
```

- Каждая цифра представляет разрешения для уровня доступа: пользователь, группа, другой.
- Цифра рассчитывается путем сложения чисел для каждого разрешения, которое вы хотите добавить: 4 для чтения, 2 для записи и 1 для выполнения.

При использовании числового метода разрешения представляются трехзначным (или четырехзначным, при настройке расширенных разрешений) восьмеричным числом. Одна восьмеричная цифра может представлять любое отдельное значение от 0 до 7.

В трехзначном восьмеричном (числовом) представлении разрешений каждая цифра обозначает один уровень доступа слева направо: пользователь, группа и другие. Чтобы определить каждую цифру:

1. Начните с 0.
2. Если для этого уровня доступа должно быть разрешение на чтение, добавьте 4.
3. Если должно быть разрешение на запись, добавьте 2.
4. Если должно быть разрешение на выполнение, добавьте 1.

Изучите разрешения **-rwxr-x ---**. Для пользователя **rwx** рассчитывается как **4 + 2 + 1 = 7**. Для группы **r-x** рассчитывается как **4 + 0 + 1 = 5**, а для других пользователей **---** представляется с **0**. Сложив эти три вместе, числовое представление этих разрешений будет **750**.

Этот расчет может быть выполнен и в обратном направлении. Посмотрите на разрешения **640**. Для прав пользователя **6** представляет чтение (**4**) и запись (**2**), что отображается как **rw-**. Для групповой части **4** включает только чтение (**4**) и отображается как **r--**. Значение **0** для других не дает разрешений (**---**), а последний набор символьических разрешений для этого файла - **-rw-r-----**.

Опытные администраторы часто используют числовые разрешения, потому что они короче для ввода и произношения, но при этом дают полный контроль над всеми разрешениями.

Примеры

- Установите разрешения на чтение и запись для пользователя, разрешение на чтение для группы и другие для файла **samplefile**:

```
[user @ host ~] $ chmod 644 samplefile
```

- Установите разрешения на чтение, запись и выполнение для пользователя, права на чтение и выполнение для группы и отсутствие разрешений для других в **sampledir**:

```
[user@host ~]$ chmod 750 sampledir
```

ИЗМЕНЕНИЕ ФАЙЛА И КАТАЛОГА ВЛАДЕЛЬСТВА ПОЛЬЗОВАТЕЛЯ ИЛИ ГРУППЫ

Вновь созданный файл принадлежит пользователю, создавшему этот файл. По умолчанию новые файлы принадлежат группе, которая является основной группой пользователя, создавшего файл. В Red Hat Enterprise Linux основная группа пользователя обычно является частной группой, в которую входит только этот пользователь. Чтобы предоставить доступ к файлу на основе членства в группе, может потребоваться изменить группу, владеющую файлом.

Только **root** может изменить пользователя, владеющего файлом. Однако владение группой может быть установлено пользователем **root** или **владельцем файла**. **root** может

предоставить право собственности на файл любой группе, но обычные пользователи могут сделать группу владельцем файла, только если они являются членами этой группы.

Владение файлом можно изменить с помощью команды **chown** (сменить владельца (**change owner**)). Например, чтобы предоставить право собственности на файл **test_file** пользователю-**student**, используйте следующую команду:

```
[root@host ~]# chown student test_file
```

chown можно использовать с параметром **-R** для рекурсивного изменения владения всем деревом каталогов. Следующая команда предоставляет студенту право владения **test_dir** и всеми файлами и подкаталогами в нем:

```
[root@host ~]# chown -R student test_dir
```

Команду **chown** также можно использовать для изменения группового владения файлом, поставив перед именем группы двоеточие (:). Например, следующая команда изменяет группу **test_dir** на **admin**:

```
[root@host ~]# chown :admin test_dir
```

Команду **chown** также можно использовать для одновременной смены владельца и группы с помощью синтаксиса **owner:group**. Например, чтобы изменить владельца файла **test_dir** на **visitor** и группу на **guests**, используйте следующую команду:

```
[root@host ~]# chown visitor:guests test_dir
```

Вместо использования **chown** некоторые пользователи меняют владельца группы с помощью команды **chggrp**. Эта команда работает так же, как **chown**, за исключением того, что она используется только для изменения владельца группы, а двоеточие (:) перед именем группы не требуется.



ВАЖНО

Вы можете встретить примеры команд **chown**, использующих альтернативный синтаксис, который разделяет владельца и группу точкой вместо двоеточия:

```
[root@host ~]# chown owner.group filename
```

Вы не должны использовать этот синтаксис. Всегда используйте двоеточие. Точка является допустимым символом в имени пользователя, а двоеточие - нет. Если пользователь **enoch.root**, пользователь **enoch** и группа **root** существуют в системе, результатом **chown enoch.root filename** будет то, что имя файла будет принадлежать пользователю **enoch.root**. Возможно, вы пытались установить владение файлом для пользователя **enoch** и группы **root**. Это может сбивать с толку.

Если вы всегда используете синтаксис **chown** двоеточия при одновременной настройке пользователя и группы, результаты всегда легко предсказать.



РЕКОМЕНДАЦИИ

Справочные страницы **man ls(1)**, **chmod(1)**, **chown(1)**, и **chgrp(1)**

УПРАЖНЕНИЯ С ПОШАГОВЫМИ ИНСТРУКЦИЯМИ

УПРАВЛЕНИЕ РАЗРЕШЕНИЯМИ ФАЙЛОВОЙ СИСТЕМЫ ИЗ КОМАНДНОЙ СТРОКИ

В этом упражнении вы будете использовать разрешения файловой системы для создания каталога, в котором все члены определенной группы могут добавлять и удалять файлы.

В РЕЗУЛЬТАТЕ

У вас должна быть возможность создать общий каталог, доступный для всех членов определенной группы.

ПРЕЖДЕ ЧЕМ ВЫ НАЧНЕТЕ

Войдите на рабочую станцию как пользователь **student**, используя в качестве пароля слово **student**.

На рабочей станции запустите команду **lab perms-cli start**, чтобы начать упражнение. Сценарий запуска создает группу с именем **consultants** и двух пользователей с именами **consultant1** и **consultant2**.

```
[student@workstation ~]$ lab perms-cli start
```

1. С рабочей станции используйте команду **ssh** для входа в сервер в качестве пользователя **student**.

```
[student@workstation ~]$ ssh student@servera  
[student@serverb ~]$
```

2. Переключитесь на пользователя **root**, используя слово **redhat** в качестве пароля.

```
[student@servera ~]$ su -  
Password: redhat  
[root@servera ~]#
```

3. Используйте команду **mkdir**, чтобы создать каталог **/home/consultants**.

```
[root@servera ~]# mkdir /home/consultants
```

4. Используйте команду **chown**, чтобы изменить группу, владеющую каталогом **consultants**, на **consultants**.

```
[root@servera ~]# chown :consultants /home/consultants
```

5. Убедитесь, что разрешения группы позволяют членам группы создавать и удалять файлы. Разрешения должны запрещать другим (**others**) доступ к файлам.

- 5.1. Используйте команду **ls**, чтобы убедиться, что разрешения группы позволяют членам группы создавать и удалять файлы в каталоге **/home/consultants**.

```
[root@servera ~]# ls -ld /home/consultants
drwxr-xr-x. 2 root     consultants       6 Feb  1 12:08 /home/consultants
```

Обратите внимание, что группа **consultants** в настоящее время не имеет разрешения на запись.

- 5.2. Используйте команду **chmod**, чтобы добавить разрешение на запись группе **consultants**.

```
[root@servera ~]# chmod g+w /home/consultants
[root@servera ~]# ls -ld /home/consultants
drwxrwxr-x. 2 root     consultants       6 Feb  1 13:21 /home/consultants
```

- 5.3. Используйте команду **chmod**, чтобы запретить прочим(**others**) доступ к файлам в каталоге **/home/consultants**.

```
[root@servera ~]# chmod 770 /home/consultants
[root@servera ~]# ls -ld /home/consultants
drwxrwx---. 2 root     consultants       6 Feb  1 12:08 /home/consultants/
```

6. Выдите из оболочки пользователя **root** и переключитесь на пользователя **consultant1**. Пароль - **redhat**.

```
[root@servera ~]# exit
logout
[student@servera ~]$
[student@servera ~]$ su - consultant1
Password: redhat
```

7. Перейдите в каталог **/home/consultants** и создайте файл с именем **consultant1.txt**.

7.1. Используйте команду **cd**, чтобы перейти в каталог **/home/consultants**.

```
[consultant1@servera ~]$ cd /home/consultants
```

7.2. Используйте команду **touch**, чтобы создать пустой файл с именем **consultant1.txt**.

```
[consultant1@servera consultants]$ touch consultant1.txt
```

8. Используйте команду **ls -l**, чтобы вывести список пользователей и групп по умолчанию, владеющих новым файлом, а также их разрешения.

```
[consultant1@servera consultants]$ ls -l consultant1.txt  
-rw-rw-r--. 1 consultant1 consultant1 0 Feb 1 12:53 consultant1.txt
```

9. Убедитесь, что все члены группы **consultants** могут редактировать файл **consultant1.txt**. Измените групповое владение файла **consultant1.txt** на **consultants**.

9.1. Используйте команду **chown**, чтобы изменить групповое владение файлом **consultant1.txt** на **consultants**.

```
[consultant1@servera consultants]$ chown :consultants consultant1.txt
```

9.2. Используйте команду **ls** с параметром **-l**, чтобы вывести список нового владельца файла **consultant1.txt**.

```
[consultant1@servera consultants]$ ls -l consultant1.txt  
-rw-rw-r--. 1 consultant1 consultants 0 Feb 1 12:53 consultant1.txt
```

10. Выйдите из оболочки и переключитесь на пользователя **consultant2**. Пароль - **redhat**.

```
[consultant1@servera consultants]$ exit  
logout  
[student@servera ~]$ su - consultant2  
Password: redhat  
[consultant2@servera ~]$
```

11. Перейдите в каталог **/home/consultants**. Убедитесь, что пользователь **consultant2** может добавлять содержимое в файл **consultant1.txt**. Выход из оболочки.

11.1. Используйте команду **cd**, чтобы перейти в каталог **/home/consultants**. Используйте команду **echo**, чтобы добавить текст в файл **consultant1.txt**.

```
[consultant2@servera ~]$ cd /home/consultants/  
[consultant2@servera consultants]$ echo "text" >> consultant1.txt  
[consultant2@servera consultants]$
```

11.2. Используйте команду **cat**, чтобы убедиться, что текст был добавлен в файл **consultant1.txt**.

```
[consultant2@servera consultants]$ cat consultant1.txt  
text  
[consultant2@servera consultants]$
```

11.3. Выйти из оболочки.

```
[consultant2@servera consultants]$ exit  
logout  
[student@servera ~]$
```

12. Выдите из сервера.

```
[student@servera ~]$ exit  
logout  
Connection to servera closed.  
[student@workstation ~]$
```

Завершение

На рабочей станции запустит **elab perms-cli finish**, чтобы завершить это упражнение. Данные сценарий обеспечивает очистку среды.

```
[student@workstation ~]$ labperms-clifinish
```

На этом пошаговое упражнение завершено.

УПРАВЛЕНИЕ РАЗРЕШЕНИЯМИ ПО УМОЛЧАНИЮ И ДОСТУПОМ К ФАЙЛАМ

ЗАДАЧИ

По завершении этого раздела учащиеся должны уметь:

- Управляйте разрешениями по умолчанию для новых файлов, созданных пользователями.
- Объясните эффект специальных разрешений.
- Используйте специальные разрешения и разрешения по умолчанию, чтобы установить владельца группы файлов, созданных в определенном каталоге.

СПЕЦИАЛЬНЫЕ РАЗРЕШЕНИЯ

Специальные разрешения составляют четвертый тип разрешений в дополнение к базовым типам пользователей, групп и другим типам. Как следует из названия, эти разрешения предоставляют дополнительные функции, связанные с доступом, помимо основных типов разрешений. В этом разделе подробно описывается влияние специальных разрешений, которые кратко описаны в таблице ниже.

Влияние специальных разрешений на файлы и каталоги

СПЕЦИАЛЬНОЕ РАЗРЕШЕНИЕ	ВЛИЯНИЕ НА ФАЙЛЫ	ВЛИЯНИЕ НА КАТАЛОГИ
u+s (suid)	Файл выполняется от имени пользователя, владеющего файлом, а не от имени пользователя, запустившего файл.	Нет эффекта.
g+s (sgid)	Файл выполняется как группа, владеющая файлом.	Для файлов, вновь созданных в каталоге, устанавливается владелец группы, совпадающий с владельцем группы в каталоге.
o+t (sticky)	Нет эффекта.	Пользователи с правом записи в каталог могут удалять только те файлы, которые им принадлежат; они не могут удалить или принудительно сохранить файлы, принадлежащие другим пользователям.

Разрешение **setuid** для исполняемого файла означает, что команды выполняются от имени пользователя, владеющего файлом, а не от пользователя, выполнившего команду. Один из примеров команды **passwd**:

```
[user@host ~]$ ls -l /usr/bin/passwd  
-rwsr-xr-x. 1 root root 35504 Jul 16 2010 /usr/bin/passwd
```

В длинном списке вы можете идентифицировать разрешения **setuid** с помощью строчной буквы **s**, где обычно должен быть **x** (права на выполнение владельцем). Если у владельца нет прав на выполнение, он заменяется заглавной **S**.

Специальное разрешение **setgid** для каталога означает, что файлы, созданные в каталоге, наследуют свое групповое владение от каталога, а не от создавшего пользователя. Это обычно используется в групповых каталогах для совместной работы, чтобы автоматически изменить файл из частной группы по умолчанию в общую группу, или если файлы в каталоге всегда должны принадлежать определенной группе. Примером этого является каталог **/run/log/journal**:

```
[user@host ~]$ ls -ld /run/log/journal  
drwxr-sr-x. 3 root systemd-journal 60 May 18 09:15 /run/log/journal
```

Если **setgid** установлен для исполняемого файла, команды выполняются от имени группы, которой принадлежит этот файл, а не от пользователя, запустившего команду, аналогично **setuid**. Одним из примеров является команда **locate**:

```
[user@host ~]$ ls -ld /usr/bin/locate  
-rwx--s--x. 1 root slocate 47128 Aug 12 17:17 /usr/bin/locate
```

В выводе содержимого каталога длинным списком вы можете идентифицировать разрешения **setgid** с помощью строчной буквы **s**, где обычно должен быть **x** (разрешения на выполнение группы). Если у группы нет разрешений на выполнение, они заменяются заглавными буквами **S**.

Наконец, липкий бит (*sticky bit*) для каталога устанавливает специальное ограничение на удаление файлов. Только владелец файла (и **root**) может удалять файлы в каталоге. Пример каталог **/tmp**:

```
[user@host ~]$ ls -ld /tmp  
drwxrwxrwt. 39 rootroot 4096 Feb 8 20:52 /tmp
```

В выводе содержимого каталога длинным списком вы можете идентифицировать закрепленные разрешения по букве **t** в нижнем регистре, где обычно должен быть **x** (разрешения для прочих (**other**) на выполнение). Если прочие (**other**) не имеет разрешений на выполнение, он заменяется заглавной буквой **T**.

Установка специальных разрешений

- Символически: setuid = **u+s**; setgid = **g+s**; sticky = **o+t**
- Численно (четвертая предыдущая цифра): setuid = 4; setgid = 2; sticky = 1

Примеры

- Добавьте бит **setgid** в каталог:

```
[user@host ~]# chmod g+s directory
```

- Установите бит **setgid** и добавьте разрешения на чтение/запись/выполнение для пользователя и группы без доступа для прочих в каталоге:

```
[user@host ~]# chmod 2770 directory
```

РАЗРЕШЕНИЯ НА ФАЙЛ ПО УМОЛЧАНИЮ

Когда вы создаете новый файл или каталог, ему назначаются начальные разрешения. На эти начальные разрешения влияют две вещи. Первый - создаете ли вы обычный файл или каталог. Вторая - текущая **umask**.

Если вы создаете новый каталог, операционная система запускается с присвоения ему восьмеричных прав доступа **0777 (drwxrwxrwx)**. Если вы создаете новый обычный файл, операционная система назначает ему восьмеричные права доступа **0666 (-rw-rw-rw-)**. Вам всегда

нужно явно добавлять разрешение на выполнение для обычного файла. Это усложняет злоумышленнику возможность взломать сетевую службу, чтобы она создавала новый файл и немедленно запускала его как программу.

Однако сеанс оболочки также установит **umask** для дальнейшего ограничения изначально установленных разрешений. Это восьмеричная битовая маска, используемая для очистки разрешений новых файлов и каталогов, созданных процессом. Если в **umask** установлен бит, то соответствующее разрешение сбрасывается для новых файлов. Например, **umask 0002** очищает бит записи для других пользователей. Начальные нули указывают на то, что особые, пользовательские и групповые разрешения не удалены. **Umask 0077** очищает все групповые и другие разрешения для вновь созданных файлов.

Команда **umask** без аргументов отобразит текущее значение **umask** оболочки:

```
[user@host ~]$ umask  
0002
```

Используйте команду **umask** с одним числовым аргументом, чтобы изменить **umask** текущей оболочки. Числовой аргумент должен быть восьмеричным значением, соответствующим новому значению **umask**. Вы можете опустить любые ведущие нули в **umask**.

Системные значения **umask** по умолчанию для пользователей оболочки **Bash** определены в файлах **/etc/profile** и **/etc/bashrc**. Пользователи могут переопределить системные значения по умолчанию в файлах **.bash_profile** и **.bashrc** в своих домашних каталогах.

Пример **umask**

В следующем примере объясняется, как **umask** влияет на разрешения файлов и каталогов. Посмотрите на разрешения **umask** по умолчанию для файлов и каталогов в текущей оболочке. Владелец и группа имеют права на чтение и запись файлов, а остальные настроены на чтение. Владелец и группа имеют разрешения на чтение, запись и выполнение в каталогах. Для прочих единственное разрешение для чтения.

```
[user@host ~]$ umask  
0002  
  
[user@host ~]$ touchdefault.txt  
  
[user@host ~]$ ls -l default.txt  
-rw-rw-r--. 1 user user 0 May 9 01:54 default.txt  
  
[user@host ~]$ mkdir default  
  
[user@host ~]$ ls -ld default  
drwxrwxr-x. 2 user user 0 May 9 01:54 default
```

Если установить значение **umask** равным **0**, права доступа к файлам для других файлов изменятся с чтения на чтение и запись. Разрешения каталога для других изменений от чтения и выполнения до чтения, записи и выполнения.

```
[user@host ~]$ umask 0
[user@host ~]$ touch zero.txt
[user@host ~]$ ls -l zero.txt
-rw-rw-rw-. 1 user user 0 May 9 01:54 zero.txt
[user@host ~]$ mkdir zero
[user@host ~]$ ls -ld zero
drwxrwxrwx. 2 user user 0 May 9 01:54 zero
```

Чтобы применить маску на все права доступа к файлам и каталогам для прочих, установите значение **umask** на **007**.

```
[user@host ~]$ umask 007
[user@host ~]$ touch seven.txt
[user@host ~]$ ls -l seven.txt
-rw-rw----. 1 user user 0 May 9 01:55 seven.txt
[user@host ~]$ mkdir seven
[user@host ~]$ ls -ld seven
drwxrwx---. 2 user user 0 May 9 01:54 seven
```

Umask 027 гарантирует, что новые файлы будут иметь права на чтение и запись для пользователя и права на чтение для группы. Новые каталоги имеют права чтения и записи для группы и не имеют прав для прочих.

```
[user@host ~]$ umask 027
[user@host ~]$ touch two-seven.txt
[user@host ~]$ ls -l two-seven.txt
-rw-r-----. 1 user user 0 May 9 01:55 two-seven.txt
[user@host ~]$ mkdir two-seven
```

```
[user@host ~]$ ls -ld two-seven
drwxr-x---. 2 user user 0 May 9 01:54 two-seven
```

По умолчанию **umask** для пользователей устанавливается сценариями запуска оболочки. По умолчанию, если **UID** вашей учетной записи равен **200** или больше, а ваше имя пользователя и имя основной группы совпадают, вам будет назначена **umask 002**. В противном случае ваша **umask** будет **022**.

Как **root**, вы можете изменить это, добавив сценарий запуска оболочки с именем **/etc/profile.d/localumask.sh**, который выглядит примерно так, как в этом примере:

```
[root@host ~]# cat /etc/profile.d/local-umask.sh
# Overrides default umask configuration
if [ $UID -gt 199 ] && [ "`id -gn`" = "`id -un`" ]; then
    umask 007
else
    umask 022
fi
```

В предыдущем примере для **umask** будет установлено значение **007** для пользователей с **UID**, превышающим **199** и совпадающими с именем пользователя и именем основной группы, и **022** для всех остальных. Если вы просто хотите установить **umask** для всех равным **022**, вы можете создать этот файл со следующим содержимым:

```
# Overrides default umask configuration
umask 022
```

Чтобы гарантировать, что глобальные изменения **umask** вступят в силу, вы должны выйти из оболочки и снова войти в нее. До этого времени **umask**, настроенный в текущей оболочке, все еще действует.



ПРИМЕЧАНИЕ ПРИ ПЕРЕВОДЕ

Авторы не упомянули на мой взгляд, два важных пункта для понимания о **umask**:

- **umask** - это пользовательская маска (**user mask**), которая используется для определения конечных прав доступа.

Как посчитать (определить) права файла для маски

- Для файла изначальное значение **666** из него вычитается **umask**, например, **(-)027**, получаем права на файл **640** или в символьном значении **-rw-r----**.
- Для каталога изначальное значение **777** из него вычитается **umask** например **(-)027**, получаем права на файл **750** или в символьном значении **drwxr-x---**.



РЕКОМЕНДАЦИИ

Страницы руководства **man bash(1)**, **ls(1)**, **chmod(1)**, и **umask(1)**

УПРАЖНЕНИЯ С ПОШАГОВЫМИ ИНСТРУКЦИЯМИ

УПРАВЛЕНИЕ РАЗРЕШЕНИЯМИ ПО УМОЛЧАНИЮ И ДОСТУПОМ К ФАЙЛАМ

В этом упражнении вы будете управлять разрешениями на новые файлы, созданные в каталоге, с помощью настроек **umask** и разрешения **setgid**.

В РЕЗУЛЬТАТЕ

Вы должны быть способны:

- Создать общий каталог, в котором новые файлы автоматически принадлежат группе **operators**.
- Поэкспериментируйте с различными настройками **umask**.
- Настроить разрешения по умолчанию для определенных пользователей.
- Подтвердите правильность настройки.

ПРЕЖДЕ ЧЕМ ВЫ НАЧНЕТЕ

Войдите на рабочую станцию **workstation** как пользователь **student**, используя в качестве пароля слово **student**.

На рабочей станции запустите команду **lab perms-default start**, чтобы начать упражнение. Команда выполняет сценарий запуска, который определяет, доступен ли **servera** сети. Скрипт также создает группу **operators**и пользователя **operator1** на сервере **servera**.

```
[student@workstation ~]$ lab perms-default start
```

1. Используйте команду **ssh** для входа на сервер как пользователь **student**.

```
[student@workstation ~]$ ssh student@servera  
[student@servera ~]$
```

2. Используйте команду **su**, чтобы переключиться на пользователя **operator1** с паролем **redhat**.

```
[student@servera ~]$ su - operator1  
Password: redhat  
[operator1@servera ~]$
```

3. Используйте команду **umask** для вывода значения **umask** по умолчанию для пользователя **operator1**.

```
[operator1@servera ~]$ umask  
0002
```

4. Создайте новый каталог с именем **/tmp/shared**. В каталоге **/tmp/shared** создайте файл с именем **defaults**. Посмотрите разрешения по умолчанию.

- 4.1. Используйте команду **mkdir** для создания каталога **/tmp/shared**. Используйте команду **ls -l**, чтобы вывести список разрешений для нового каталога.

```
[operator1@servera ~]$ mkdir /tmp/shared  
[operator1@servera ~]$ ls -l /tmp/shared  
drwxrwxr-x. 2 operator1 operator1 6 Feb 4 14:06 /tmp/shared
```

- 4.2. Используйте команду **touch**, чтобы создать файл с именем **defaults** в каталоге **/tmp/shared**.

```
[operator1@servera ~]$ touch /tmp/shared/defaults
```

- 4.3. Используйте команду **ls -l**, чтобы вывести список разрешений для нового файла.

```
[operator1@servera ~]$ ls -l /tmp/shared/defaults  
-rw-rw-r--. 1 operator1 operator1 0 Feb 4 14:09 /tmp/shared/defaults
```

5. Измените владение правами для группы **/tmp/shared** на группу **operators**. Подтвердите новое право владение и разрешения.

- 5.1. Используйте команду **chown**, чтобы изменить групповое владение каталогом **/tmp/shared** на группу **operators**.

```
[operator1@servera ~]$ chown :operators /tmp/shared
```

- 5.2. Используйте команду **ls -l**, чтобы вывести список разрешений для каталога **/tmp/shared**.

```
[operator1@servera ~]$ ls -l /tmp/shared  
drwxrwxr-x. 2 operator1 operators 22 Feb 4 14:09 /tmp/shared
```

- 5.3. Используйте команду **touch** для создания файла с именем **group** в каталоге **/tmp/shared**. Используйте команду **ls -l**, чтобы вывести список разрешений для файлов.

```
[operator1@servera ~]$ touch /tmp/shared/group  
[operator1@servera ~]$ ls -l /tmp/shared/group  
-rw-rw-r--. 1 operator1 operator1 0 Feb 4 17:00 /tmp/shared/group
```



ПРИМЕЧАНИЕ

Владелец группы файла **/tmp/shared/group** - это не **operators**, а **operator1**.

6. Убедитесь, что файлы, созданные в каталоге **/tmp/shared**, принадлежат группе **operators**.

- 6.1. Используйте команду **chmod**, чтобы установить идентификатор группы для группы **operators** для каталога **/tmp/shared**.

```
[operator1@servera ~]$ chmod g+s /tmp/shared
```

- 6.2. С помощью команды **touch** создайте новый файл с именем **operations_database.txt** в каталоге **/tmp/shared**.

```
[operator1@servera ~]$ touch /tmp/shared/operations_database.txt
```

- 6.3. Используйте команду **ls -l**, чтобы убедиться, что группа **operators** является владельцем группы для нового файла.

```
[operator1@servera ~]$ ls -l /tmp/shared/operations_database.txt  
-rw-rw-r--. 1 operator1 operators 0 Feb 4 16:11 /tmp/shared/  
operations_database.txt
```

7. Создайте новый файл в каталоге **/tmp/shared** с именем **operations_network.txt**. Запишите право собственности и разрешения. Измените **umask** для **operator1**. Создайте новый файл с именем **operations_production.txt**. Запишите владельца и права доступа к файлу **operations_production.txt**.

- 7.1. Используйте команду **echo** для создания файла с именем **operations_network.txt** в каталоге **/tmp/shared**.

```
[operator1@servera ~]$ echo text >> /tmp/shared/operations_network.txt
```

7.2. Используйте команду **ls -l**, чтобы вывести список разрешений для файла **operations_network.txt**.

```
[operator1@servera ~]$ ls -l /tmp/shared/operations_network.txt  
-rw-rw-r--. 1 operator1 operators 5 Feb 4 15:43 /tmp/shared/  
operations_network.txt
```

7.3. Используйте команду **umask**, чтобы изменить **umask** для пользователя **operator1** на **027**. Используйте команду **umask**, чтобы подтвердить изменение.

```
[operator1@servera ~]$ umask 027  
[operator1@servera ~]$ umask  
0027
```

7.4. С помощью команды **touch** создайте новый файл с именем **operations_production.txt** в каталоге **/tmp/shared/**. Используйте команду **ls -l**, чтобы убедиться, что вновь созданные файлы создаются с доступом только для чтения для группы операторов и без доступа для других пользователей.

```
[operator1@servera ~]$ touch /tmp/shared/operations_production.txt  
[operator1@servera ~]$ ls -l /tmp/shared/operations_production.txt  
-rw-r-----. 1 operator1 operators 0 Feb 4 15:56 /tmp/shared/  
operations_production.txt
```

8. Откройте новое окно терминала и войдите на сервер **servera**, как **operator1**.

```
[student@workstation ~]$ ssh operator1@servera  
[operator1@servera ~]$
```

9. Посмотрите значение **umask** для **operator1**.

```
[operator1@servera ~]$ umask  
0002
```

10. Измените **umask** по умолчанию для пользователя **operator1**. Новая маска **umask** запрещает любой доступ для пользователей, не входящих в их группу. Подтвердите, что **umask** была изменена.

10.1. Используйте команду **echo**, чтобы изменить **umask** по умолчанию для пользователя **operator1** на **007**.

```
[operator1@servera ~]$ echo "umask 007" >> ~/.bashrc
[operator1@servera ~]$ cat ~/.bashrc

# .bashrc

# Source global definitions
if [ -f /etc/bashrc ]; then
    . /etc/bashrc
fi

# Uncomment the following line if you don't like systemctl's auto-paging feature:
# export SYSTEMD_PAGER=

# User specific aliases and functions
umask 007
```

10.2. Выйдите из системы и снова войдите в систему как пользователь **operator1**. Используйте команду **umask**, чтобы подтвердить, что изменение является постоянным.

```
[operator1@servera ~]$ exit
logout
Connection to servera closed.
[student@workstation ~]$ ssh operator1@servera
[operator1@servera ~]$ umask
0007
```

11. На сервере выйдите из всех пользовательских оболочек **operator1** и **student**.



ПРИМЕЧАНИЕ

Выход из всех оболочек, открытых **operator1**. Несспособность выйти из всех оболочек приведет к сбою сценария завершения.

```
[operator1@servera ~]$ exit
logout
Connection to servera closed.
[student@workstation ~]$
```

Завершение

На рабочей станции **workstation**, запустите **lab perms-default finish**, чтобы завершить это упражнение. Данные сценарий обеспечивает очистку среды.

```
[student@workstation ~]$ labperms-defaultfinish
```

На этом пошаговое упражнение завершено.

ЛАБОРАТОРНАЯ РАБОТА

КОНТРОЛЬ ДОСТУПА К ФАЙЛАМ

КОНТРОЛЬНЫЙ СПИСОК РАБОТЫ

В этой лабораторной работе вы настроите разрешения для файлов и создадите каталог, который пользователи в определенной группе могут использовать для удобного обмена файлами в локальной файловой системе.

В РЕЗУЛЬТАТЕ

Вы должны быть способны:

- Создавать каталог, в котором пользователи могут совместно работать с файлами.
- Создавать файлы, которым автоматически назначается групповое владение.
- Создавать файлы, недоступные вне группы.

ПРЕЖДЕ ЧЕМ ВЫ НАЧНЕТЕ

Войдите на рабочую станцию как пользователь **student**, используя в качестве пароля слово **student**.

На рабочей станции запустите команду **lab perms-review start**, чтобы начать упражнение. Команда выполняет сценарий запуска, который определяет, доступен ли **serverb** в сети. Сценарий также создает группу **techdocs** и трех пользователей с именами **tech1**, **tech2** и **database1**.

```
[student@workstation ~]$ lab perms-review start
```

1. Используйте команду **ssh** для входа на **serverb** как пользователь **student**. Переключитесь на **root** на **serverb**, используя **redhat** в качестве пароля.
2. Создайте каталог с именем **/home/techdocs**.
3. Измените групповое владение каталогом **/home/techdocs** на группу **techdocs**.
4. Убедитесь, что пользователи в группе **techdocs** могут создавать и редактировать файлы в каталоге **/home/techdocs**.
5. Установите разрешения для каталога **/home/techdocs**. В каталоге **/home/techdocs** настройте **setgid(2)**, права на чтение/запись/выполнение (7) для владельца/пользователя и группы и отсутствие разрешений (0) для других пользователей.
6. Убедитесь, что разрешения установлены правильно.

- Убедитесь, что пользователи группы **techdocs** теперь могут создавать и редактировать файлы в каталоге **/home/techdocs**. Пользователи, не входящие в группу **techdocs**, не могут редактировать или создавать файлы в каталоге **/home/techdocs**. Пользователи **tech1** и **tech2** входят в группу **techdocs**. Пользователь **database1** не входит в эту группу.
- Измените сценарии глобального входа в систему. Обычные пользователи должны иметь настройку **umask**, которая запрещает другим пользователям просматривать или изменять новые файлы и каталоги.
- Выходите из сервера **serverb**.

```
[student@serverb ~]$ exit  
logout  
Connection to serverb closed.
```

Оценка

На рабочей станции запустите сценарий **lab perms-review grade**, чтобы подтвердить успех этого упражнения.

```
[student@workstation ~]$ lab perms-review grade
```

Окончание

На рабочей станции запустите сценарий **lab perms-review finish**, чтобы завершить лабораторную работу.

```
[student@workstation ~]$ lab perms-review finish
```

На этом лабораторная работа завершена.

РЕШЕНИЕ

КОНТРОЛЬ ДОСТУПА К ФАЙЛАМ

КОНТРОЛЬНЫЙ СПИСОК РАБОТЫ

В этой лабораторной работе вы настроите разрешения для файлов и создадите каталог, который пользователи в определенной группе могут использовать для удобного обмена файлами в локальной файловой системе.

В РЕЗУЛЬТАТЕ

Вы должны быть способны:

- Создавать каталог, в котором пользователи могут совместно работать с файлами.
- Создавать файлы, которым автоматически назначается групповое владение.
- Создавать файлы, недоступные вне группы.

ПРЕЖДЕ ЧЕМ ВЫ НАЧНЕТЕ

Войдите на рабочую станцию как пользователь **student**, используя в качестве пароля слово **student**.

На рабочей станции запустите команду **lab perms-review start**, чтобы начать упражнение. Команда выполняет сценарий запуска, который определяет, доступен ли **serverb** в сети. Сценарий также создает группу **techdocs** и трех пользователей с именами **tech1**, **tech2** и **database1**.

```
[student@workstation ~]$ lab perms-review start
```

1. Используйте команду **ssh** для входа на **serverb** как пользователь **student**. Переключитесь на **root** на **serverb**, используя **redhat** в качестве пароля.

```
[student@workstation ~]$ ssh student@serverb
[student@serverb ~]$ su -
Password: redhat
[root@serverb ~]#
```

2. Создайте каталог с именем **/home/techdocs**.

2.1. Используйте команду **mkdir**, чтобы создать каталог с именем **/home/techdocs**.

```
[root@serverb ~]# mkdir /home/techdocs
```

3. Измените групповое владение каталогом **/home/techdocs** на группу **techdocs**.

3.1. Используйте команду **chown**, чтобы изменить групповое владение каталогом **/home/techdocs** на группу **techdocs**.

```
[root@serverb ~]# chown :techdocs /home/techdocs
```

4. Убедитесь, что пользователи в группе **techdocs** могут создавать и редактировать файлы в каталоге **/home/techdocs**.

4.1. Используйте команду **su**, чтобы переключиться на пользователя **tech1**.

```
[root@serverb ~]# su - tech1  
[tech1@serverb ~]$
```

4.2. С помощью команды **touch** создайте файл с именем **techdoc1.txt** в каталоге **/home/techdocs**.

```
[tech1@serverb ~]$ touch /home/techdocs/techdoc1.txt  
touch: cannot touch '/home/techdocs/techdoc1.txt': Permission denied
```



ПРИМЕЧАНИЕ

Обратите внимание, что даже несмотря на то, что каталог **/home/techdocs** принадлежит **techdocs**, а **tech1** является частью группы **techdocs**, создать новый файл в этом каталоге невозможно. Это потому, что у группы **techdocs** нет разрешения на запись. Используйте команду **ls -ld**, чтобы показать разрешения.

```
[tech1@serverb ~]$ ls -ld /home/techdocs/  
drwxr-xr-x. 2 root techdocs 6 Feb 5 16:05 /home/techdocs/
```

5. Установите разрешения для каталога **/home/techdocs**. В каталоге **/home/techdocs** настройте **setgid(2)**, права на чтение/запись/выполнение (**7**) для владельца/пользователя и группы и отсутствие разрешений (**0**) для других пользователей.

5.1. Выйдите из оболочки пользователя **tech1**.

```
[tech1@serverb ~]$ exit  
logout  
[root@serverb ~]#
```

5.2. Используйте команду **chmod**, чтобы установить разрешение для группы на каталог **/home/techdocs**. В каталоге **/home/techdocs** настройте **setgid (2)**, права на чтение/запись/выполнение (**7**) для владельца/пользователя и группы и отсутствие разрешений (**0**) для прочих пользователей.

```
[root@serverb ~]# chmod 2770 /home/techdocs
```

6. Убедитесь, что разрешения установлены правильно.

```
[root@serverb ~]# ls -ld /home/techdocs  
drwxrws---. 2 root techdocs 6 Feb 4 18:12 /home/techdocs/
```

7. Убедитесь, что пользователи группы **techdocs** теперь могут создавать и редактировать файлы в каталоге **/home/techdocs**. Пользователи, не входящие в группу **techdocs**, не могут редактировать или создавать файлы в каталоге **/home/techdocs**. Пользователи **tech1** и **tech2** входят в группу **techdocs**. Пользователь **database1** не входит в эту группу.

7.1. Переключитесь на пользователя **tech1**. С помощью команды **touch** создайте файл с именем **techdoc1.txt** в каталоге **/home/techdocs**. Выполните выход из оболочки пользователя **tech1**.

```
[root@serverb ~]# su - tech1  
[tech1@serverb ~]$ touch /home/techdocs/techdoc1.txt  
[tech1@serverb ~]$ ls -l /home/techdocs/techdoc1.txt  
-rw-rw-r--. 1 tech1 techdocs 0 Feb 5 16:42 /home/techdocs/techdoc1.txt  
[tech1@serverb ~]$ exit  
logout  
[root@serverb ~]#
```

7.2. Переключитесь на пользователя **tech2**. Используйте команду **echo**, чтобы добавить содержимое в файл **/home/techdocs/techdoc1.txt**. Выйдите из оболочки пользователя **tech2**.

```
[root@serverb ~]# su - tech2  
[tech2@serverb ~]$ echo "This is the first tech doc." > /home/techdocs/techdoc1.txt  
[tech2@serverb ~]$ exit  
logout  
[root@serverb ~]#
```

7.3. Переключитесь на пользователя **database1**. Используйте команду **echo**, чтобы добавить содержимое в файл **/home/techdocs/techdoc1.txt**. Обратите внимание, что вы получите сообщение **Permission Denied**. Используйте команду **ls -l**, чтобы убедиться, что **database1** не имеет доступа к файлу. Выйдите из оболочки пользователя **database1**.

```
[root@serverb ~]# su - database1
[database1@serverb ~]$ echo "This is the first tech doc." >> /home/techdocs/techdoc1.txt
-bash: /home/techdocs/techdoc1.txt: Permission denied
[database1@serverb ~]$ ls -l /home/techdocs/techdoc1.txt
ls: cannot access '/home/techdocs/techdoc1.txt': Permission denied
[database1@serverb ~]$ exit
logout
[root@serverb ~]#
```

8. Измените сценарии глобального входа в систему. Обычные пользователи должны иметь настройку **umask**, которая запрещает другим пользователям просматривать или изменять новые файлы и каталоги.

8.1. Определите **umask** пользователя **student**. Используйте команду **su - student**, чтобы переключиться на оболочку входа в систему для пользователя **student**. По завершении выйдите из оболочки.

```
[root@serverb ~]# su - student
[student@serverb ~]$ umask
0002
[student@serverb ~]$ exit
Logout
```

8.2. Создайте файл **/etc/profile.d/local-umask.sh** со следующим содержимым, чтобы установить **umask** на **007** для пользователей с **UID** больше **199** и с соответствующими именем пользователя и именем основной группы, а также на **022** для всех остальных:

```
[root@serverb ~]# cat /etc/profile.d/local-umask.sh

# Overrides default umask configuration
if [ $UID -gt 199 ] && [ "`id -gn`" = "`id -un`" ]; then
    umask 007
else
    umask 022
fi
...output omitted...
```

8.3. Выйдите из оболочки и войдите снова как пользователь **student**, чтобы убедиться, что глобальная **umask** изменилась на **007**.

```
[root@serverb ~]# exit  
logout  
[student@serverb ~]$ exit  
logout  
Connection to serverb closed.  
[student@workstation ~]$ ssh student@serverb  
[student@serverb ~]$ umask  
0007
```

9. Выйдите из сервера **serverb**.

```
[student@serverb ~]$ exit  
logout  
Connection to serverb closed.
```

Оценка

На рабочей станции запустите сценарий **lab perms-review grade**, чтобы подтвердить успех этого упражнения.

```
[student@workstation ~]$ lab perms-review grade
```

Окончание

На рабочей станции запустите сценарий **lab perms-review finish**, чтобы завершить лабораторную работу.

```
[student@workstation ~]$ lab perms-review finish
```

На этом лабораторная работа завершена.

РЕЗЮМЕ

В этой главе вы узнали:

- Файлы имеют три категории, к которым применяются разрешения. Файл принадлежит пользователю, отдельной группе и прочим пользователям. Применяется самое конкретное разрешение. Разрешения пользователей имеют приоритет над разрешениями группы, а разрешения группы имеют приоритет над прочими разрешениями.
- Команда **ls** с параметром **-l** выводит расширенный список файлов, включая права доступа к файлам и владельцев.
- Команда **chmod** изменяет права доступа к файлам из командной строки. Есть два метода представления разрешений: символьный (буквы) и числовой (цифры).
- Команда **chown** изменяет владельца файла. Параметр **-R** рекурсивно изменяет владельца дерева каталогов.
- Команда **umask** без аргументов отображает текущее значение **umask** оболочки. У каждого процесса в системе есть маска. Значения **umask** по умолчанию для **Bash** определены в файлах **/etc/profile** и **/etc/bashrc**.

ГЛАВА 8

УПРАВЛЕНИЕ ЛОКАЛЬНЫМИ ПОЛЬЗОВАТЕЛЯМИ И ГРУППАМИ

ЦЕЛЬ

Оценивать и контролировать процессы, запущенные в системе Red Hat Enterprise Linux.

ЗАДАЧИ

- Получите информацию о программах, запущенных в системе, чтобы вы могли определить статус, использование ресурсов и владельцев, чтобы вы могли контролировать их.
- Используйте управление заданиями Bash для управления несколькими процессами, запущенными из одного сеанса терминала.
- Контроль и завершение процессов, которые не связаны с вашей оболочкой, а также принудительное завершение сеансов и процессов конечных пользователей.
- Опишите, какова средняя нагрузка, и определите процессы, ответственные за высокое использование ресурсов на сервере.

РАЗДЕЛЫ

- Список процессов (и контрольный опрос)
- Управление заданиями (и упражнения с пошаговыми инструкциями)
- Уничтожение процессов (и упражнения с пошаговыми инструкциями)
- Мониторинг активности процесса (и упражнения с пошаговыми инструкциями)

ЛАБОРОТОРНАЯ РАБОТА

Мониторинг и управление процессами Linux.

ОТСЛЕЖИВАНИЕ ПРОЦЕСОВ

ЗАДАЧИ

После завершения этого раздела вы сможете получить информацию о программах, запущенных в системе, для определения статуса, использования ресурсов и владения, чтобы вы могли контролировать их.

ОПРЕДЕЛЕНИЕ ПРОЦЕССА

Процесс - это запущенный экземпляр исполняемой программы. Процесс состоит из:

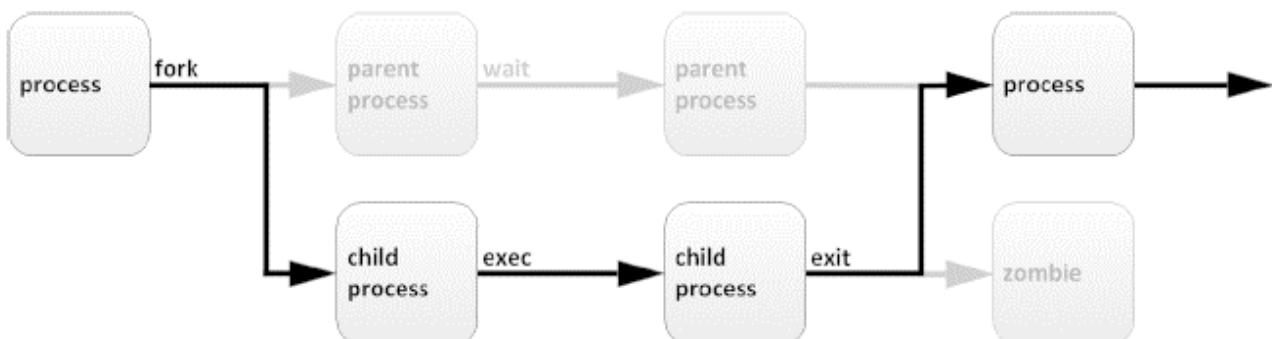
- Адресное пространство выделенной памяти
- Свойства безопасности, включая учетные данные и права владения
- Один или несколько потоков выполнения программного кода
- Состояние процесса

Среда процесса включает:

- Локальные и глобальные переменные
- Текущий контекст планирования
- Выделенные системные ресурсы, такие как файловые дескрипторы и сетевые порты.

Существующий (родительский (*parent*)) процесс дублирует свое собственное адресное пространство (вилку (*fork*)) для создания новой (дочерней (*child*)) структуры процесса. Каждому новому процессу присваивается уникальный идентификатор процесса (*process ID*) (**PID**) для отслеживания и безопасности. **PID** и идентификатор родительского процесса (**PPID**) являются элементами новой среды процесса. Любой процесс может создать дочерний процесс. Все процессы являются потомками первого системного процесса, **systemd** в системе Red Hat Enterprise Linux 8).

Рисунок 8.1: Жизненный цикл процесса



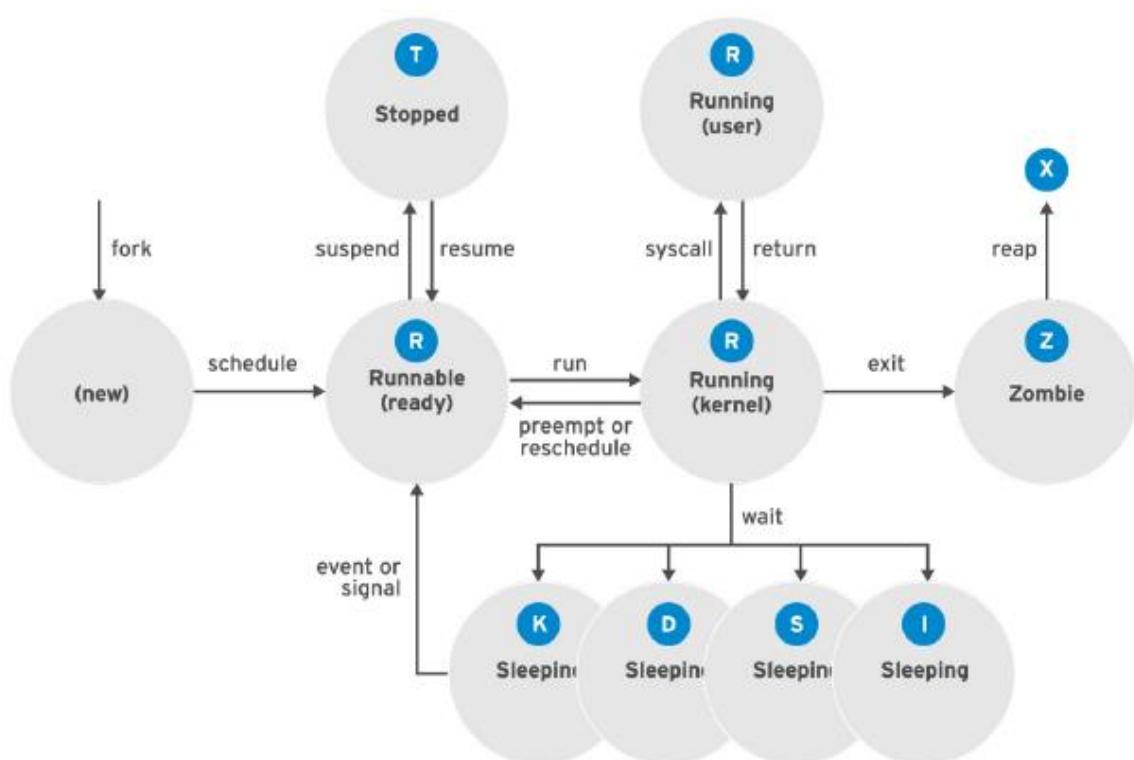
Посредством процедуры **fork** дочерний процесс наследует идентификаторы безопасности, предыдущие и текущие файловые дескрипторы, права доступа к портам и ресурсам, переменные среды и программный код. Затем дочерний процесс может выполнить (*exec*) свой собственный программный код. Обычно родительский процесс спит (*sleeps*), пока выполняется дочерний

процесс, устанавливая запрос (ожидание(*wait*)), который будет сигнализироваться, когда дочерний процесс завершится. При выходе дочерний процесс уже закрыл или отказался от своих ресурсов и среды. Единственный оставшийся ресурс, называемый зомби (*zombie*), - это запись в таблице процессов. Родитель, проснувшийся при выходе из дочернего процесса, очищает таблицу процессов от записи дочернего процесса, освобождая таким образом последний ресурс дочернего процесса. Затем родительский процесс продолжает выполнение собственного программного кода.

ОПИСАНИЕ СОСТОЯНИЯ ПРОЦЕССА

В многозадачной операционной системе каждый ЦП (или ядро ЦП) может работать над одним процессом в один момент времени. По мере выполнения процесса меняются его непосредственные потребности в процессоре и распределении ресурсов. Процессам присваивается состояние (*state*), которое меняется в зависимости от обстоятельств.

Рисунок 8.2: Состояния процессов Linux



Состояния процессов Linux проиллюстрированы на предыдущей диаграмме и описаны в следующей таблице:

НАЗВАНИЕ	ФЛАГ	ОПРЕДЕЛЕННОЕ ЯДРОМ СОСТОЯНИЯ	НАЗВАНИЕ И ОПИСАНИЕ
----------	------	------------------------------	---------------------

Running	R	TASK_RUNNING: процесс либо выполняется на ЦП, либо ожидает запуска. Процесс может выполнять пользовательские подпрограммы или подпрограммы ядра (системные вызовы) или находится в очереди и быть готовым в состоянии «Выполняется (<i>Running</i>)» (или «Выполняется (<i>Runnable</i>)»).
Sleeping	S	TASK_INTERRUPTIBLE: процесс ожидает некоторого условия: аппаратного запроса, доступа к системным ресурсам или сигнала. Когда событие или сигнал удовлетворяют условию, процесс возвращается к выполнению (<i>Running</i>).
	D	TASK_UNINTERRUPTIBLE: этот процесс также находится в спящем режиме (<i>Sleeping</i>), но, в отличие от состояния S , не отвечает на сигналы. Используется только тогда, когда прерывание процесса может вызвать непредсказуемое состояние устройства.
	K	TASK_KILLABLE: идентично непрерывному состоянию D , но изменено, чтобы позволить ожидающей задаче реагировать на сигнал о том, что она должна быть уничтожена (полностью выйти). Утилиты часто отображают убиваемые (<i>Killable</i>) процессы как состояние D
	I	TASK_REPORT_IDLE: подмножество состояния D . Ядро не учитывает эти процессы при вычислении средней нагрузки. Используется для потоков ядра. Установлены флаги TASK_UNINTERRUPTABLE и TASK_NOLOAD . Подобно TASK_KILLABLE , также является подмножеством состояния D . Он принимает фатальные сигналы.
Stopped	T	TASK_STOPPED: процесс был остановлен (приостановлен), обычно по сигналу пользователя или другого процесса. Процесс может быть продолжен (возобновлен) другим сигналом возврата в состояние «Выполнение». Остановлен
	T	TASK_TRACED: отлаживаемый процесс также временно остановлен и использует тот же флаг состояния T
Zombie	Z	EXIT_ZOMBIE: дочерний процесс сигнализирует своему родительскому при выходе. Все ресурсы, кроме идентификатора процесса (PID), освобождаются.
	X	EXIT_DEAD: когда родитель очищает (перебирает) оставшуюся структуру дочернего процесса, процесс теперь полностью освобождается. Это состояние никогда не будет соблюдаться в утилитах, перечисляющих процессы.

Почему так важны состояния процесса

При устраниении неполадок в системе важно понимать, как ядро взаимодействует с процессами и как процессы взаимодействуют друг с другом. При создании процесса система присваивает процессу состояние. Столбец **S** команды **top** или столбец **STAT** команды **ps** показывают состояние каждого процесса. В системе с одним ЦП одновременно может выполняться только один процесс. Можно увидеть несколько процессов с состоянием **R**. Однако не все из них будут выполняться последовательно, некоторые из них будут находиться в состоянии ожидания.

```
[user@host ~]$ top
PID USER PR NI VIRT RES SHR S %CPU %MEM TIME+ COMMAND
 1 root 20 0 244344 13684 9024 S 0.0 0.7 0:02.46 systemd
 2 root 20 0 0 0 0 S 0.0 0.0 0:00.00 kthreadd
...output omitted...
```

```
[user@host ~]$ ps aux
USER PID %CPU %MEM VSZ RSS TTY STAT START TIME COMMAND
...output omitted...
root 2 0.0 0.0 0 0 ? S 11:57 0:00 [kthreadd]
student 3448 0.0 0.2 266904 3836 pts/0 R+ 18:07 0:00 ps aux
...output omitted...
```

Процесс может быть приостановлен, остановлен, возобновлен, завершен или прерван с помощью сигналов. Сигналы обсуждаются более подробно позже в этой главе. Сигналы могут использоваться другими процессами, самим ядром или пользователями, вошедшими в систему.

ВЫВОД СПИСКА ПРОЦЕССОВ

Команда **ps** используется для вывода списка текущих процессов. Она может предоставить подробную информацию о процессе, включая:

- Идентификатор пользователя (**UID**), который определяет привилегии процесса
- Уникальный идентификатор процесса (**PID**)
- Затраченное процессорное и реальное время
- Сколько памяти процесс выделил в различных местах
- Местоположение **stdout** процесса, известное как управляющий терминал
- Текущее состояние процесса



ВАЖНО

Версия **ps** для **Linux** поддерживает три формата опций:

Параметры **UNIX** (**POSIX**), которые могут быть сгруппированы и должны начинаться с тире.

Опции BSD, которые могут быть сгруппированы и не должны использоваться с тире.

Длинные параметры GNU, которым предшествуют два дефиса

Например, **ps -aux** - это не то же самое, что **ps aux**.

Возможно, наиболее распространенный набор опций, **aux**, отображает все процессы, включая процессы без управляющего терминала. Длинный список (опции **lax**) предоставляет более подробную техническую информацию, но может отображаться быстрее, если не нужно искать имя пользователя. Аналогичный синтаксис UNIX использует параметры **-ef** для отображения всех процессов.

```
[user@host ~]$ ps aux
USER        PID %CPU %MEM      VSZ      RSS TTY      STAT START   TIME COMMAND
root          1  0.1  0.1  51648  7504 ?          Ss  17:45  0:03 /usr/lib/systemd/
sys
root          2  0.0  0.0      0      0 ?          S   17:45  0:00 [kthreadd]
root          3  0.0  0.0      0      0 ?          S   17:45  0:00 [ksoftirqd/0]
root          5  0.0  0.0      0      0 ?          S<  17:45  0:00 [kworker/0:0H]
root          7  0.0  0.0      0      0 ?          S   17:45  0:00 [migration/0]
...output omitted...
[user@host ~]$ ps lax
F   UID    PID  PPID PRI  NI      VSZ      RSS WCHAN  STAT TTY      TIME COMMAND
4   0      1      0  20   0  51648  7504 ep_pol Ss   ?      0:03 /usr/lib/
systemd/
1   0      2      0  20   0      0      0 kthrea S   ?      0:00 [kthreadd]
1   0      3      2  20   0      0      0 smpboo S   ?      0:00 [ksoftirqd/0]
1   0      5      2  0  -20      0      0 worker  S<  ?      0:00 [kworker/0:0H]
1   0      7      2 -100  -      0      0 smpboo S   ?      0:00 [migration/0]
...output omitted...
[user@host ~]$ ps -ef
UID        PID  PPID  C STIME TTY      TIME CMD
root          1      0  0 17:45 ?
switched-ro
root          2      0  0 17:45 ?      00:00:00 [kthreadd]
root          3      2  0 17:45 ?      00:00:00 [ksoftirqd/0]
root          5      2  0 17:45 ?      00:00:00 [kworker/0:0H]
root          7      2  0 17:45 ?      00:00:00 [migration/0]
...output omitted...
```

По умолчанию **ps** без параметров выбирает все процессы с тем же эффективным идентификатором пользователя (**EUID**), что и текущий пользователь, и которые связаны с тем же терминалом, на котором была вызвана **ps**.

- Процессы в скобках (обычно вверху списка) являются запланированными потоками ядра.
- Зомби перечислены как выбывшие (**exiting**) или прекратившие существование (**defunct**).
- Вывод **ps** отображается один раз. Используйте команду **top** для отображения процесса, который динамически обновляется.

- **ps** может отображаться в виде дерева, чтобы вы могли просматривать отношения между родительскими и дочерними процессами.
- Выходные данные по умолчанию сортируются по идентификатору процесса. На первый взгляд это может показаться хронологическим порядком. Однако ядро повторно использует идентификаторы процессов, поэтому порядок менее структурирован, чем кажется. Для сортировки используйте параметры **-O** или **--sort**. Порядок отображения совпадает с порядком отображения системной таблицы процессов, в которой строки таблицы повторно используются по мере того, как процессы умирают и создаются новые. Вывод может отображаться в хронологическом порядке, но это не гарантируется, если не используются явные параметры **-O** или **--sort**.



РЕКОМЕНДАЦИИ

info libc signal (GNU C Library Reference Manual)

- Section 24: Signal Handling

info libc processes (GNU C Library Reference Manual)

- Section 26: Processes

Справочные страницы **man ps(1)** и **signal(7)**

КОНТРОЛЬНЫЙ ОПРОС

СПИСОК ПРОЦЕСОВ

Выберите правильные ответы на следующие вопросы:

1. Какое состояние соответствует остановленному или приостановленному процессу?
 - a. D
 - b. R
 - c. S
 - d. T
 - e. Z

2. Какое состояние представляет процесс, который освободил все свои ресурсы, кроме своего PID?
 - a. D
 - b. R
 - c. S
 - d. T
 - e. Z

3. Какой процесс родительский использует для дублирования для создания нового дочернего процесса?
 - a. exec
 - b. fork
 - c. zombie
 - d. syscall
 - e. reap

4. Какое состояние представляет процесс, который находится в спящем режиме, пока не будет выполнено какое-либо условие?
 - a. D
 - b. R
 - c. S
 - d. T
 - e. Z

РЕШЕНИЕ

СПИСОК ПРОЦЕСОВ

Выберите правильные ответы на следующие вопросы:

- 1.** Какое состояние соответствует остановленному или подвисшему процессу?
 - a. D
 - b. R
 - c. S
 - d. T
 - e. Z

- 2.** Какое состояние представляет процесс, который освободил все свои ресурсы, кроме своего PID?
 - a. D
 - b. R
 - c. S
 - d. T
 - e. Z

- 3.** Какой процесс родительский использует для дублирования для создания нового дочернего процесса?
 - a. exec
 - b. fork**
 - c. zombie
 - d. syscall
 - e. reap

- 4.** Какое состояние представляет процесс, который находится в спящем режиме, пока не будет выполнено какое-либо условие?
 - a. D
 - b. R
 - c. S
 - d. T
 - e. Z

УПРАВЛЕНИЕ ЗАДАНИЯМИ

ЗАДАЧИ

После завершения этого раздела вы сможете использовать управление заданиями **Bash** для управления несколькими процессами, запущенными из одного сеанса терминала.

ОПИСАНИЕ ЗАДАНИЙ И СЕССИЙ

Управление заданиями (*Job control*) - это функция оболочки, которая позволяет одному экземпляру оболочки запускать и управлять несколькими командами.

Задание связано с каждым конвейером (**pipeline**), введенным в приглашении оболочки. Все процессы в этом конвейере являются частью задания и являются членами одной группы процессов. Если в приглашении оболочки вводится только одна команда, это можно рассматривать как минимальный «конвейер» из одной команды, создавая задание только с одним членом.

Только одно задание может одновременно считывать ввод и сигналы, генерируемые клавиатурой из определенного окна терминала. Процессы, которые являются частью этой работы, являются процессами переднего плана (*foreground*) этого управляющего терминала.

Фоновый (*background*) процесс управляющего терминала является членом любого другого задания, связанного с этим терминалом. Фоновые процессы терминала не могут читать ввод или получать прерывания, генерируемые клавиатурой, от терминала, но могут иметь возможность писать в терминал. Задание в фоновом режиме может быть остановлено (приостановлено) или может быть запущено. Если запущенное фоновое задание пытается прочитать с терминала, оно будет автоматически приостановлено.

Каждый терминал является отдельным сеансом и может иметь процесс переднего плана и любое количество независимых фоновых процессов. Задание является частью только одного сеанса: сеанса, принадлежащего его управляющему терминалу.

Команда **ps** показывает имя устройства управляющего терминала процесса в столбце **TTY**. Некоторые процессы, такие как системные демоны (*system daemons*), запускаются системой, а не из командной строки. У этих процессов нет управляющего терминала, они не являются членами задания и не могут быть выведены на передний план. Команда **ps** отображает вопросительный знак (?) в столбце **TTY** для этих процессов.

ВЫПОЛНЕНИЕ ЗАДАНИЙ В ФОНОВОМ РЕЖИМЕ (BACKGROUND)

Любую команду или конвейер можно запустить в фоновом режиме, добавив амперсанд (&) в конец командной строки. Оболочка **Bash** отображает номер задания (уникальный для сеанса) и **PID** нового дочернего процесса. Оболочка не ждет завершения дочернего процесса, а отображает приглашение оболочки.

```
[user@host ~]$ sleep 10000 &
[1] 5947
[user@host ~]$
```



ПРИМЕЧАНИЕ

Когда командная строка, содержащая канал, отправляется в фоновый режим с использованием амперсанда, в качестве вывода используется **PID** последней команды в конвейере. Все процессы в конвейере по-прежнему участвуют в этой работе.

```
[user@host ~]$ example_command | sort | mail -s "Sort output" &
[1] 5998
```

Вы можете отобразить список заданий, которые **Bash** отслеживает для определенного сеанса, с помощью команды **jobs**.

```
[user@host ~]$ jobs
[1]+ Running sleep 10000 &
[user@host ~]$
```

Фоновое задание можно вывести на передний план с помощью команды **fg** с его идентификатором задания (**% номерзадания**).

```
[user@host ~]$ fg %1
sleep 10000
```

В предыдущем примере команда **sleep** теперь выполняется на переднем плане на управляющем терминале. Сама оболочка снова спит, ожидая выхода этого дочернего процесса.

Чтобы отправить процесс переднего плана на задний план, сначала нажмите на клавиатуре запрос на приостановку (**Ctrl + Z**) в терминале.

```
sleep 10000
^Z
[1]+ Stopped sleep 10000
[user@host ~]$
```

Работа немедленно переводится в фоновый режим и приостанавливается.

Команда **ps j** отображает информацию о заданиях. **PID** - это уникальный идентификатор процесса. **PPID** - это **PID** родительского процесса (*parent process*) этого процесса, процесса, который его запустил (разветвил (**forked**))). **PGID** - это **PID** лидера группы процессов (*process group leader*), обычно первого процесса в конвейере задания. **SID** - это **PID** лидера сеанса, который (для задания) обычно является интерактивной оболочкой, работающей на его управляющем

терминале. Поскольку пример команды `sleep` настоящее время приостановлен, состояние процесса **T**.

```
[user@host ~]$ ps j
PPID  PID  PGID  SID TTY      TPGID STAT   UID    TIME COMMAND
2764  2768  2768  2768 pts/0      6377 Ss    1000   0:00 /bin/bash
2768  5947  5947  2768 pts/0      6377 T     1000   0:00 sleep 10000
2768  6377  6377  2768 pts/0      6377 R+    1000   0:00 ps j
```

Чтобы запустить приостановленный процесс в фоновом режиме, используйте команду **bg** с тем же идентификатором задания.

```
[user@host ~]$ bg %1
[1]+ sleep 10000 &
```

Оболочка предупредит пользователя, при попытке выйти из окна терминала (сеанса) с приостановленными заданиями. Если пользователь снова попытается выйти сразу же, приостановленные задания будут уничтожены.



ПРИМЕЧАНИЕ

Обратите внимание на знак + после [1] в приведенных выше примерах. Знак + указывает, что это задание является текущим заданием по умолчанию. То есть, если используется команда, которая ожидает аргумент **% номера задания**, а номер задания не указан, то действие выполняется для задания с индикатором +.



РЕКОМЕНДАЦИИ

Информационная страница **Bash** (Справочное руководство GNU Bash)
<https://www.gnu.org/software/bash/manual>

- Раздел 7: Управление заданиями

Информационная страница **libc** (Справочное руководство по библиотеке GNU C)
<https://www.gnu.org/software/libc/manual/pdf/libc.pdf/>

- Раздел 24: Обработка сигналов
- Раздел 26: Процессы

справочные страницы **man bash(1)**, **builtins(1)**, **ps(1)**, **sleep(1)**

Дополнительные сведения см. В главе «Инструменты мониторинга системы» в Руководстве системного администратора Red Hat Enterprise Linux 7 по адресу https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/7/html-single/ch-system_monitoring_tools.

УПРАЖНЕНИЯ С ПОШАГОВЫМИ ИНСТРУКЦИЯМИ

УПРАВЛЕНИЕ ЗАДАНИЯМИ

В этом упражнении вы будете запускать, приостанавливать, запускать несколько процессов в фоновом режиме и на переднем плане, используя управление заданиями.

В РЕЗУЛЬТАТЕ

Вы должны иметь возможность использовать управление заданиями для приостановки и перезапуска пользовательских процессов.

ПРЕЖДЕ ЧЕМ ВЫ НАЧНЕТЕ

Войдите на рабочую станцию, как пользователь **student**, используя в качестве пароля слово **student**.

На рабочей станции запустите сценарий **lab processes-control start**. Этот сценарий обеспечивает доступность сервера.

```
[student@workstation ~]$ labprocesses-controlstart
```

1. На рабочей станции откройтесь в два окна терминала. В этом разделе эти два терминала называются **левым** и **правым**. В каждом терминале используйте команду **ssh** для входа на сервер в качестве пользователя **student**.

```
[student@workstation ~]$ ssh student@servera  
Activate the web console with: systemctl enable --now cockpit.socket
```

```
Last login: Tue Feb 5 16:24:14 2019 from 172.25.250.254  
[student@servera ~]$
```

2. В левом окне создайте новый каталог с именем **/home/student/bin**. В новом каталоге создайте сценарий оболочки с именем **control**. Сделайте скрипт исполняемым.

- 2.1. Используйте команду **mkdir**, чтобы создать новый каталог с именем **/home/student/bin**.

```
[student@servera ~]$ mkdir /home/student/bin
```

- 2.2. Используйте команду **vim**, чтобы создать сценарий под названием **control** в **/home/student/bin**. Чтобы войти в интерактивный режим **Vim**, нажмите клавишу **I**. Используйте команду **:wq**, чтобы сохранить файл.

```
[student@servera ~]$ vim /home/student/bin/control
#!/bin/bash
while true; do
    echo -n "$@" >> ~/control_outfile
    sleep 1
done
```



ПРИМЕЧАНИЕ

Сценарий **control** выполняется до завершения. Он добавляет аргументы командной строки к файлу **~/control_outfile** один раз в секунду.

- 2.3. Используйте команду **chmod** с опцией **+x**, чтобы сделать файл **control** исполняемым.

```
[student@servera ~]$ chmod +x /home/student/bin/control
```

3. Запустите скрипт **control**. Сценарий непрерывно добавляет слово «**technical**» и пробел к файлу **~/control_outfile** с интервалом в одну секунду.



ПРИМЕЧАНИЕ

Вы можете выполнить свой управляющий скрипт, потому что он находится в вашем PATH и стал исполняемым.

```
[student@servera ~]$ control technical
```

4. В оболочке правого терминала используйте команду **tail** с параметром **-f**, чтобы подтвердить, что новый процесс выполняет запись в файл.

```
[student@servera ~]$ tail -f ~/control_outfile
technical technical technical technical
...output omitted...
```

5. В оболочке левого терминала нажмите **Ctrl+Z**, чтобы приостановить запущенный процесс. Оболочка возвращает идентификатор задания в квадратных скобках. В правом окне подтвердите, что вывод процесса остановлен.

```
^Z
[1]+  Stopped                  control technical
[student@servera ~]$
```

```
technical technical technical technical
...no further output...
```

6. В оболочке левого терминала просмотрите список заданий. Помните, что знак + указывает на задание по умолчанию. Перезапустите работу в фоновом режиме. В shellправого терминала убедитесь, что вывод процесса снова активен.

6.1. Просмотрите список заданий с помощью команды **jobs**.

```
[student@servera ~]$ jobs
[1]+  Stopped                  control technical
```

6.2. Используя команду **bg**, перезапустите контрольное задание в фоновом режиме.

```
[student@servera ~]$ bg
[1]+ control technical &
```

6.3. Используйте команду **jobs**, чтобы убедиться, что задание **control** снова выполняется.

```
[student@servera ~]$ jobs
[1]+  Running                  control technical &
```

6.4. В оболочке правого терминала убедитесь, что команда **tail** выдаёт нужный результат.

```
...output omitted...
technical technical technical technical technical technical technical technical
```

7. В оболочке левого терминала запустите еще два процесса выполнив скрипт **control**, которые будут добавлены в файл **~/output**. Используйте амперсанд (&), чтобы процессы запустились в фоновом режиме. Замените документ **technicaldocuments**, а затем **database**. Замена аргументов помогает различать три процесса.

```
[student@servera ~]$ control documents &
[2] 6579
[student@servera ~]$
[student@servera ~]$ control database &
[3] 6654
```



ПРИМЕЧАНИЕ

Номер задания каждого нового процесса указан в квадратных скобках. Второе число - это уникальный общесистемный идентификационный номер процесса (PID) для процесса.

8. В оболочке левого терминала используйте команду **jobs**, чтобы просмотреть три запущенных процесса. В оболочке правого терминала убедитесь, что все три процесса добавляются к файлу.

```
[student@servera ~]$ jobs
[1]  Running                  control technical &
[2]- Running                  control documents &
[3]+ Running|                control database &

...output omitted...
technical documents database technical documents database technical documents
database technical documents database
...output omitted...
```

9. Приостановить **control technical**. Подтвердите, что он был приостановлен. Прервите процесс **control documents** и подтвердите, что он был прекращен.

- 9.1. В оболочке левого терминала используйте команду **fg** с идентификатором задания, чтобы вывести на передний план процесс **control technical**. Нажмите **Ctrl+Z**, чтобы приостановить процесс. Используйте команду **jobs**, чтобы подтвердить, что процесс приостановлен.

```
[student@servera ~]$ fg %1
control technical
^Z
[1]+ Stopped                  control technical
[student@servera ~]$ jobs
[1]+ Stopped                  control technical
[2]  Running                  control documents &
[3]- Running|                control database &
```

- 9.2. В оболочке правого терминала убедитесь, что процесс **control technical** больше не отправляет выходные данные.

```
database documents database documents database
...no further output...
```

- 9.3. В оболочке левого терминала используйте команду **fg** с идентификатором задания, чтобы вывести процесс **control documents** на передний план. Нажмите **Ctrl+C**, чтобы завершить процесс. Используйте команду **jobs**, чтобы подтвердить, что процесс завершен.

```
[student@servera ~]$ fg %2
control documents
^C
[student@servera ~]$ jobs
[1]+  Stopped                  control technical
[3]-  Running                  control database &
```

- 9.4. В оболочке правого терминала убедитесь, что процесс **control documents** больше не отправляет выходные данные.

```
...output omitted...
database database database database database database database
...no further output...
```

10. В левом окне используйте команду **ps** с параметром **jT**, чтобы просмотреть оставшиеся задания. Приостановленные задания имеют состояние **T**. Остальные фоновые задания находятся в спящем состоянии (**S**).

```
[student@servera ~]$ ps jT
  PPID  PID  PGID  SID TTY      TPGID STAT   UID    TIME COMMAND
 27277 27278 27278 27278 pts/1    28702 Ss   1000   0:00 -bash
 27278 28234 28234 27278 pts/1    28702 T    1000   0:00 /bin/bash /home/student/
bin/control technical
 27278 28251 28251 27278 pts/1    28702 S    1000   0:00 /bin/bash /home/student/
bin/control database
 28234 28316 28234 27278 pts/1    28702 T    1000   0:00 sleep 1
 28251 28701 28251 27278 pts/1    28702 S    1000   0:00 sleep 1
 27278 28702 28702 27278 pts/1    28702 R+   1000   0:00 ps jT
```

11. Выходите из сервера на обоих терминалах.

```
[student@servera ~]$ exit
logout
Connection to servera closed.
```

```
[student@servera ~]$ exit
logout
Connection to servera closed.
```

Завершение

На рабочей станции запустите сценарий **lab processes-control finish**, чтобы завершить данное упражнение.

На этом пошаговое упражнение завершено.

УНИЧТОЖЕНИЕ ПРОЦЕССОВ

ЗАДАЧИ

После завершения этого раздела вы сможете:

- Используйте команды для уничтожения процессов и взаимодействия с ними.
- Определение характеристик процесса демона.
- Сессии и процессы конечных пользователей.

УПРАВЛЕНИЕ ПРОЦЕССОМ С ПОМОЩЬЮ СИГНАЛОВ

Сигнал - это программное прерывание, доставленное процессу. Сигналы сообщают о событиях исполняемой программе. События, генерирующие сигнал, могут быть ошибкой, внешним событием (например, запросом ввода-вывода или истекшим таймером) или явным запросом (использование команды отправки сигнала или последовательности клавиатуры).

В следующей таблице перечислены основные сигналы, используемые системными администраторами для повседневного управления процессами. Обращайтесь к сигналам по их короткому (HUP) или собственному (SIGHUP) имени.

Основные сигналы управления процессами

НОМЕР СИГНАЛА	КОРОТКОЕ ИМЯ	ОПРЕДЕЛЕНИЕ	ЦЕЛЬ
1	HUP	Зависание, отбой	Используется для сообщения о завершении процесса управления терминалом. Также используется для запроса повторной инициализации процесса (перезагрузки конфигурации) без завершения.
2	INT	Прерывание клавиатуры	Вызывает завершение программы. Может быть заблокирован или обработан. Отправляется нажатием комбинации клавиш INTR (Ctrl+C).
3	QUIT	Выйти клавиатуры	Подобно SIGINT, но также производит дамп процесса при завершении. Отправляется нажатием комбинации клавиш QUIT (Ctrl + \).
9	KILL	Убить, неблокируемый	Вызывает внезапное завершение программы. Не могут быть заблокированы, проигнорированы или обработаны; всегда фатально.

15 <i>default</i>	TERM	Прекратить	Вызывает завершение программы. В отличие от SIGKILL , его можно заблокировать, проигнорировать или обработать. «Вежливый» способ попросить программу прекратить работу; позволяет само очистку.
18	CONT	Продолжать	Отправлено процессу для возобновления, если он остановлен. Не может быть заблокирован. Даже если обработать, всегда возобновляет процесс.
19	STOP	Стоп, неблокируемый	Приостанавливает процесс. Не может быть заблокирован или обработан.
20	TSTP	Остановка клавиатуры	В отличие от SIGSTOP , его можно заблокировать, проигнорировать или обработать. Отправляется нажатием комбинации клавиш SUSP (Ctrl+Z).



ПРИМЕЧАНИЕ

Номера сигналов различаются на разных аппаратных платформах Linux, но названия и значения сигналов стандартизированы. При использовании команд рекомендуется использовать имена сигналов вместо чисел. Числа, обсуждаемые в этом разделе, относятся к системам `x86_64`.

Каждый сигнал имеет действие по умолчанию, обычно одно из следующих:

Term- вызывает немедленное завершение (выход) программы.

Core- заставляет программу сохранять образ памяти (дамп ядра), а затем завершать работу.

Stop- заставляет программу прекратить выполнение (приостановить) и дождаться продолжения (возобновления).

Программы могут быть подготовлены к реагированию на ожидаемые сигналы событий путем реализации процедур-обработчиков для игнорирования, замены или расширения действия сигнала по умолчанию.

Команды для отправки сигналов явным запросом

Вы сигнализируете их текущему процессу переднего плана, нажимая последовательность управления с клавиатуры, чтобы **приостановить** (**Ctrl + Z**), **убить** (**Ctrl + C**) или **сделать дамп ядра** (**Ctrl + I**) процесса. Однако вы будете использовать команды отправки сигналов для отправки сигналов фоновому процессу или процессам в другом сеансе.

Сигналы могут быть указаны в качестве параметров либо по имени (например, **-HUP** или **-SIGHUP**), либо по номеру (соответствующий **-1**). Пользователи могут уничтожать свои собственные процессы, но для уничтожения процессов, принадлежащих другим, требуется **привилегия root**.

Команда **kill** отправляет сигнал процессу по номеру **PID**. Несмотря на название, команду **kill** можно использовать для отправки любого сигнала, а не только для завершения программ. Вы можете использовать команду **kill -l**, чтобы перечислить имена и номера всех доступных сигналов.

```
[user@host ~]$ kill -l
 1) SIGHUP      2) SIGINT      3) SIGQUIT      4) SIGILL      5) SIGTRAP
 6) SIGABRT     7) SIGBUS      8) SIGFPE       9) SIGKILL     10) SIGUSR1
11) SIGSEGV     12) SIGUSR2     13) SIGPIPE     14) SIGALRM     15) SIGTERM
16) SIGSTKFLT   17) SIGCHLD     18) SIGCONT     19) SIGSTOP     20) SIGTSTP
...output omitted...
[user@host ~]$ ps aux | grep job
5194  0.0  0.1 222448  2980 pts/1    S    16:39   0:00 /bin/bash /home/user/bin/
control job1
5199  0.0  0.1 222448  3132 pts/1    S    16:39   0:00 /bin/bash /home/user/bin/
control job2
5205  0.0  0.1 222448  3124 pts/1    S    16:39   0:00 /bin/bash /home/user/bin/
control job3
5430  0.0  0.0 221860  1096 pts/1    S+   16:41   0:00 grep --color=auto job
```

```
[user@host ~]$ kill 5194
[user@host ~]$ ps aux | grep job
user  5199  0.0  0.1 222448  3132 pts/1    S    16:39   0:00 /bin/bash /home/
user/bin/control job2
user  5205  0.0  0.1 222448  3124 pts/1    S    16:39   0:00 /bin/bash /home/
user/bin/control job3
user  5783  0.0  0.0 221860   964 pts/1    S+   16:43   0:00 grep --color=auto
job
[1]  Terminated                  control job1
[user@host ~]$ kill -9 5199
[user@host ~]$ ps aux | grep job
user  5205  0.0  0.1 222448  3124 pts/1    S    16:39   0:00 /bin/bash /home/
user/bin/control job3
user  5930  0.0  0.0 221860  1048 pts/1    S+   16:44   0:00 grep --color=auto
job
[2]- Killed                      control job2
[user@host ~]$ kill -SIGTERM 5205
user  5986  0.0  0.0 221860  1048 pts/1    S+   16:45   0:00 grep --color=auto
job
[3]+ Terminated                  control job3
```

Команда **killall** может сигнализировать нескольким процессам в зависимости от их имени.

```
[user@host ~]$ ps aux | grep job
5194 0.0 0.1 222448 2980 pts/1 S 16:39 0:00 /bin/bash /home/user/bin/
control job1
5199 0.0 0.1 222448 3132 pts/1 S 16:39 0:00 /bin/bash /home/user/bin/
control job2
5205 0.0 0.1 222448 3124 pts/1 S 16:39 0:00 /bin/bash /home/user/bin/
control job3
5430 0.0 0.0 221860 1096 pts/1 S+ 16:41 0:00 grep --color=auto job
[user@host ~]$ killall control
[1] Terminated control job1
[2]- Terminated control job2
[3]+ Terminated control job3
[user@host ~]$
```

Используйте **pkill** для отправки сигнала одному или нескольким процессам, которые соответствуют критериям выбора. Критериями выбора могут быть имя команды, процессы, принадлежащие конкретному пользователю, или все общесистемные процессы. Команда **pkill** включает расширенные критерии выбора:

- Команда (**Command**) - процессы с именем команды, совпадающим с шаблоном.
- **UID** - процессы, принадлежащие учетной записи пользователя Linux, действующей или реальной.
- **GID** - процессы, принадлежащие учетной записи группы Linux, эффективными или реальными.
- Родительский (**Parent**) - дочерние процессы определенного родительского процесса.
- Терминал (**Terminal**) - процессы, выполняемые на определенном управляющем терминале.

```
[user@host ~]$ ps aux |grep pkill
user 5992 0.0 0.1 222448 3040 pts/1 S 16:59 0:00 /bin/bash /home/
user/bin/control pkill1
user 5996 0.0 0.1 222448 3048 pts/1 S 16:59 0:00 /bin/bash /home/
user/bin/control pkill2
user 6004 0.0 0.1 222448 3048 pts/1 S 16:59 0:00 /bin/bash /home/
user/bin/control pkill3
```

```
[user@host ~]$ pkill control
[1] Terminated control pkill1
[2]- Terminated control pkill2
```

```
[user@host ~]$ ps aux | grep pkill
```

```

user 6219 0.0 0.0 221860 1052 pts/1 S+ 17:00 0:00 grep --color=auto
pkill
[3]+ Terminated control pkill3

[user@host ~]$ ps aux | grep test

user 6281 0.0 0.1 222448 3012 pts/1 S 17:04 0:00 /bin/bash /home/
user/bin/control test1
user 6285 0.0 0.1 222448 3128 pts/1 S 17:04 0:00 /bin/bash /home/
user/bin/control test2
user 6292 0.0 0.1 222448 3064 pts/1 S 17:04 0:00 /bin/bash /home/
user/bin/control test3
user 6318 0.0 0.0 221860 1080 pts/1 S+ 17:04 0:00 grep --color=auto
test

[user@host ~]$ pkill -U user
[user@host ~]$ ps aux | grep test

user 6870 0.0 0.0 221860 1048 pts/0 S+ 17:07 0:00 grep --color=auto
test

```

АДМИНИСТРАТИВНЫЙ ВЫВОД ПОЛЬЗОВАТЕЛЕЙ ИЗ СИСТЕМЫ

Вам может потребоваться вывести из системы других пользователей по разным причинам. Назовём несколько из большого множества: пользователь совершил нарушение безопасности; пользователь мог чрезмерно использовать ресурсы; у пользователя может быть не отвечающая система; или пользователь имеет ненадлежащий доступ к материалам. В этих случаях вам может потребоваться административное завершение их сеанса с помощью сигналов.

Чтобы выйти из системы, сначала определите сеанс входа в систему, который необходимо завершить. Используйте команду `w` для вывода списка учетных записей пользователей и текущих запущенных процессов. Обратите внимание на столбцы **TTY** и **FROM**, чтобы определить сеансы, которые нужно закрыть.

Все сеансы входа в систему связаны с терминальным устройством (**TTY**). Если имя устройства имеет форму **pts/N**, это псевдотерминал, связанный с окном графического терминала или сеансом удаленного входа в систему. Если он имеет форму **ttyN**, пользователь находится на системной консоли, альтернативной консоли или другом напрямую подключенном терминальном устройстве.

```

[user@host ~]$ w
12:43:06 up 27 min, 5 users, load average: 0.03, 0.17, 0.66
USER   TTY    FROM          LOGIN@    IDLE   JCPU   PCPU WHAT
root   tty2          12:26   14:58  0.04s  0.04s -bash
bob    tty3          12:28   14:42  0.02s  0.02s -bash
user   pts/1  desk.example.com 12:41   2.00s  0.03s  0.03s w

```

Узнайте, как долго пользователь находится в системе, просмотрев время входа в сеанс. Для каждого сеанса ресурсы ЦП, потребляемые текущими заданиями, включая фоновые задачи и дочерние процессы, указаны в столбце **JCPU**. Текущее потребление ЦП процессом переднего плана указано в столбце **PCPU**.

Процессы и сеансы могут сигнализироваться индивидуально или коллективно. Чтобы завершить все процессы для одного пользователя, используйте команду **pkill**. Поскольку начальный процесс в сеансе входа в систему (лидер сеанса) предназначен для обработки запросов на завершение сеанса и игнорирования непреднамеренных сигналов клавиатуры, для уничтожения всех пользовательских процессов и оболочек входа в систему требуется использование сигнала **SIGKILL**.



ВАЖНО

Администраторы часто используют **SIGKILL** слишком поспешно.

Поскольку сигнал **SIGKILL** нельзя обработать или проигнорировать, он всегда фатален. Однако он принудительно завершает работу, не позволяя убитому процессу запускать процедуры самоочистки. Рекомендуется сначала отправить **SIGTERM**, затем попробовать **SIGINT** и только в том случае, если оба не удастся повторить попытку с **SIGKILL**.

Сначала определите номера **PID**, которые должны быть уничтожены, с помощью **pgrep**, который работает так же, как **pkill**, включая использование тех же параметров, за исключением того, что **pgrep** перечисляет процессы, а не уничтожает их.

```
[root@host ~]# pgrep -l -u bob
```

```
6964 bash
6998 sleep
6999 sleep
```

```
[root@host ~]# pkill -SIGKILL -u bob
```

```
[root@host ~]# pgrep -l -u bob
```

```
[root@host ~]#
```

Когда процессы, требующие внимания, находятся в одном сеансе входа в систему, нет необходимости убивать все процессы пользователя. Определите управляющий терминал для сеанса с помощью команды **w**, затем уничтожьте только процессы, ссылающиеся на тот же идентификатор терминала. Если не указан **SIGKILL**, лидер сеанса (здесь оболочка входа в **Bash**) успешно обрабатывает запрос на завершение и выдерживает его, но все остальные процессы сеанса завершаются.

```
[root@host ~]# pgrep -l -u bob
```

```
7391 bash
7426 sleep
7427 sleep
```

```
7428 sleep
[root@host ~]# w -h -u bob
bob    tty3  18:37  5:04  0.03s  0.03s  -bash
[root@host ~]# pkill -t tty3
[root@host ~]# pgrep -l -u bob
7391 bash
[root@host ~]# pkill -SIGKILL -t tty3
[root@host ~]# pgrep -l -u bob
[root@host ~]#
```

Такое же выборочное завершение процесса может быть применено с использованием отношений родительского и дочернего процессов. Используйте команду **pstree**, чтобы просмотреть дерево процессов для системы или отдельного пользователя. Используйте **PID** родительского процесса, чтобы убить всех созданных ими дочерних процессов. На этот раз родительская оболочка ввода в Bash выживает, потому что сигнал направляется только ее дочерним процессам.

```
[root@host ~]# pstree -p bob
bash(8391)─sleep(8425)
             ├sleep(8426)
             └sleep(8427)

[root@host ~]# pkill -P 8391
[root@host ~]# pgrep -l -u bob
bash(8391)
[root@host ~]# pkill -SIGKILL -P 8391
[root@host ~]# pgrep -l -u bob
bash(8391)
[root@host ~]#
```



РЕКОМЕНДАЦИИ

info libc signal (Справочное руководство по библиотеке GNU C)

Раздел 24: Обработка сигналов

info libc processes (Справочное руководство по библиотеке GNU C)

- Раздел 26: Процессы

Справочные страницы **mckill(1)**, **killall(1)**, **pgrep(1)**, **pkill(1)**, **pstree(1)**, **signal(7)** и **w(1)**

УПРАЖНЕНИЯ С ПОШАГОВЫМИ ИНСТРУКЦИЯМИ

УНИЧТОЖЕНИЕ ПРОЦЕССОВ

В этом упражнении вы будете использовать сигналы для управления процессами и их остановки.

В РЕЗУЛЬТАТЕ

Вы должны иметь возможность запускать и останавливать несколько процессов оболочки.

ПРЕЖДЕ ЧЕМ ВЫ НАЧНЕТЕ

Войдите на рабочую станцию, как пользователь **student**, используя в качестве пароля слово **student**.

На рабочей станции запустите **lab processes-kill start**. Команда запускает сценарий, который определяет, доступен ли хост, сервер, в сети.

```
[student@workstation ~]$ lab processes-kill start
```

1. На рабочей станции откройте рядом два окна терминала. В этом разделе соответственно будут называться **левым** и **правым**. В каждом терминале используйте команду **ssh** для входа на сервер **servera** в качестве пользователя - **student**.

```
[student@workstation ~]$ ssh student@servera
Activate the web console with: systemctl enable --now cockpit.socket
Last login: Fri Feb  8 12:04:52 2019 from 172.25.250.254
[student@servera ~]$
```

2. В левом окне создайте новый каталог с именем **/home/student/bin**. В новом каталоге создайте сценарий оболочки под названием **killing**. Сделайте скрипт исполняемым.

- 2.1. Используйте команду **mkdir**, чтобы создать новый каталог с именем **/home/student/bin**.

```
[student@servera ~]$ mkdir /home/student/bin
```

- 2.2. Используйте команду **vim**, чтобы создать сценарий под названием **killing** в каталоге **/home/student/bin**. Нажмите клавишу **I**, чтобы войти в интерактивный режим **Vim**. Используйте команду **:wq**, чтобы сохранить файл.

```
[student@servera ~]$ vim /home/student/bin/killing
#!/bin/bash
```

```
while true; do
    echo -n "$@" >> ~/killing_outfile
    sleep 5
done
```



ПРИМЕЧАНИЕ

Сценарий **killing** выполняется до остановки. Он добавляет аргументы командной строки в **~/killing_outfile** каждые 5 секунд.

- 2.3. Используйте команду **chmod** с опцией **+x**, чтобы сделать файл **killing** исполняемым.

```
[student@servera ~]$ chmod +x /home/student/bin/killing
```

3. В оболочке левого терминала используйте команду **cd**, чтобы перейти в каталог **/home/student/bin/**. Запустите три процесса **killing** с аргументами **network**, **interface** и **connection** соответственно. Запустите три процесса: **network**, **interface** и **connection**. Используйте амперсанд (**&**), чтобы запускать процессы в фоновом режиме.

```
[student@servera ~]$ cd /home/student/bin
[student@servera bin]$ killing network &
[1] 3460
[student@servera bin]$ killing interface &
[2] 3482
[student@servera bin]$ killing connection &
[3] 3516
```

У ваших процессов будут разные номера **PID**.

4. В оболочке правого терминала используйте команду **tail** с параметром **-f**, чтобы убедиться, что все три процесса добавляются к файлу.

```
[student@servera ~]$ tail -f ~/killing_outfile
network interface network connection interface network connection interface
network
...output omitted...
```

5. В оболочке левого терминала используйте команду **jobs** для вывода списка процессов.

```
[student@servera bin]$ jobs
[1] Running      killing network &
[2]- Running     killing interface &
[3]+ Running     killing connection &
```

6. Используйте сигналы, чтобы приостановить процесс **network**. Убедитесь, что процесс **network** остановлен. В оболочке правого терминала убедитесь, что сетевой процесс больше не добавляет вывод в **~/killing_output**.

- 6.1. Используйте **kill** с опцией **-SIGSTOP**, чтобы остановить процесс **network**. Выполните **jobs**, чтобы убедиться, что процессы остановлены.

```
[student@servera bin]$ kill -SIGSTOP %1
[1]+ Stopped killing network
[student@servera bin]$ jobs
[1]+ Stopped killing network
[2] Running killing interface &
[3]- Running killing connection &
```

- 6.2. В оболочке правого терминала посмотрите на вывод команды **tail**. Убедитесь, что слово «**network**» больше не добавляется к файлу **~/killing_outfile**.

```
...output omitted...
interface connection interface connection interface connection interface
```

7. В оболочке левого терминала завершите процесс **interface** с помощью сигналов. Подтвердите, что процесс **interface** исчез. В оболочке правого терминала убедитесь, что выходные данные процесса **interface** больше не добавляются в файл **~/killing_outfile**.

- 7.1. Используйте команду **kill** с параметром **-SIGTERM**, чтобы завершить процесс **interface**. Запустите команду **jobs**, чтобы убедиться, что она была прекращена.

```
[student@servera bin]$ kill -SIGTERM %2
[student@servera bin]$ jobs
[1]+ Stopped killing network
[2] Terminated killing interface
[3]- Running killing connection &
```

- 7.2. В оболочке правого терминала посмотрите на вывод команды **tail**. Убедитесь, что слово **interface** больше не добавляется к файлу **~/killing_outfile**.

```
...output omitted...
connection connection connection connection connection connection connection
connection
```

8. В оболочке левого терминала с помощью сигналов возобновите процесс **network**. Убедитесь, что процесс **network** запущен. В правом окне убедитесь, что выходные данные процесса **network** добавляются в файл **~/killing_outfile**.

- 8.1.** Используйте команду **kill** с **-SIGCONT**, чтобы возобновить процесс **network**. Запустите команду **jobs**, чтобы убедиться, что процесс выполняется (**Running**).

```
[student@servera bin]$ kill -SIGCONT %1
[student@servera bin]$ jobs
[1]+  Running                  killing network &
[3]-  Running                  killing connection &
```

- 8.2.** В оболочке правого терминала посмотрите на вывод команды **tail**. Убедитесь, что слово «**network**» добавляется к файлу **~/killing_outfile**.

```
...output omitted...
network connection network connection network connection network connection
network connection
```

- 9.** В оболочке левого терминала завершите оставшиеся два задания. Убедитесь, что никаких заданий не осталось и что вывод остановлен.

- 9.1.** Используйте команду **kill** с параметром **-SIGTERM**, чтобы завершить процесс **network**. Используйте ту же команду, чтобы завершить процесс **connection**.

```
[student@servera bin]$ kill -SIGTERM %1
[student@servera bin]$ kill -SIGTERM %3
[1]+  Terminated                killing network
[student@servera bin]$ jobs
[3]+  Terminated                killing connection
```

- 10.** В оболочке левого терминала перечислите процессы **tail**, запущенные во всех открытых терминальных оболочках. Завершить выполнение команд **tail**. Убедитесь, что процесс больше не запущен.

- 10.1.** Используйте команду **ps** с параметром **-ef**, чтобы вывести список всех запущенных процессов **tail**. Уточните поиск с помощью команды **grep**.

```
[student@servera bin]$ ps -ef | grep tail
student  4581 31358  0 10:02 pts/0    00:00:00 tail -f killing_outfile
student  4869  2252  0 10:33 pts/1    00:00:00 grep --color=auto tail
```

- 10.2.** Используйте команду **pkill** с параметром **-SIGTERM**, чтобы убить процесс **tail**. Используйте **ps**, чтобы подтвердить, что его больше нет.

```
[student@servera bin]$ pkill -SIGTERM tail
[student@servera bin]$ ps -ef | grep tail
```

```
student 4874 2252 0 10:36 pts/1 00:00:00 grep --color=auto tail
```

- 10.3. В оболочке правого терминала убедитесь, что команда **tail** больше не выводит данные.

```
...output omitted...
network connection network connection network connection Terminated
[student@servera ~]$
```

11. Выдите из терминалов в обоих окнах. Несспособность выйти из всех сеансов приводит к сбою сценария завершения.

```
[student@servera bin]$ exit
logout
Connection to servera closed.
[student@workstation ~]$
```

```
[student@servera ~]$ exit
logout
Connection to servera closed.
[student@workstation ~]$
```

Завершение

На рабочей станции запустите сценарий **lab processes-kill finish**, чтобы завершить упражнение.

```
[student@workstation ~]$ lab processes-kill finish
```

На этом пошаговое упражнение завершено.

МОНИТОРИНГ АКТИВНОСТИ ПРОЦЕССОВ

ЗАДАЧИ

После завершения этого раздела вы сможете описать, какая средняя нагрузка, и определить процессы, ответственные за высокое использование ресурсов на сервере.

ОПИСАНИЕ СРЕДНЕЙ НАГРУЗКИ

Средняя загрузка - это измерение, предоставляемое ядром Linux, которое представляет собой простой способ представить воспринимаемую загрузку системы с течением времени. Его можно использовать в качестве приблизительного показателя того, сколько запросов системных ресурсов ожидает обработки, а также для определения того, увеличивается или уменьшается нагрузка на систему с течением времени. Каждые пять секунд ядро собирает текущие показатели загрузки в зависимости от количества процессов, находящихся в работоспособном и бесперебойном состояниях. Это число накапливается и отображается, как экспоненциальная скользящая средняя за последние 1, 5 и 15 минут.

Понимание расчета средней нагрузки Linux

Средняя нагрузка представляет собой воспринимаемую нагрузку на систему за период времени. Linux определяет это, сообщая, сколько процессов готовы к запуску на ЦП и сколько процессов ожидают завершения дискового или сетевого ввода-вывода.

- Число загрузки, по существу, основано на количестве процессов, которые готовы к запуску (в состоянии процесса **R**) и ожидают завершения ввода-вывода (в состоянии процесса **D**).
- Некоторые системы UNIX учитывают только загрузку ЦП или длину очереди выполнения для обозначения загрузки системы. Linux также включает использование диска или сети, потому что это может иметь такое же значительное влияние на производительность системы, как и загрузка процессора. Если вы испытываете среднюю высокую нагрузку при минимальной активности ЦП, проверьте активность диска и сети.

Средняя нагрузка - это приблизительное измерение того, сколько процессов в настоящее время ожидают завершения запроса, прежде чем они смогут сделать что-либо еще. Может потребоваться процессорное время для запуска процесса. В качестве альтернативы запрос может быть на завершение критической операции ввода-вывода диска, и процесс не может быть запущен на ЦП до завершения запроса, даже если ЦП находится в режиме ожидания. В любом случае это влияет на загрузку системы, и кажется, что система работает медленнее, потому что процессы ожидают запуска.

Интерпретация отображаемых средних значений нагрузки

Команда **uptime** - это один из способов отобразить текущую среднюю нагрузку. Он печатает текущее время, как долго машина была в рабочем состоянии, сколько пользовательских сеансов запущено, а также текущую среднюю нагрузку.

```
[user@host ~]$ uptime  
15:29:03 up 14 min, 2 users, load average: 2.92, 4.48, 5.20
```

Три значения средней нагрузки представляют нагрузку за последние 1, 5 и 15 минут. Быстрый взгляд показывает, увеличивается или уменьшается нагрузка на систему.

Если основной вклад в среднюю нагрузку вносят процессы, ожидающие ЦП, можно рассчитать приблизительное значение нагрузки на ЦП, чтобы определить, испытывает ли система значительное ожидание.

Команда **lscpu** может помочь вам определить, сколько процессоров имеет система.

В следующем примере система представляет собой двухъядерную однопроцессорную систему с двумя гиперпотоками на ядро. Грубо говоря, Linux будет рассматривать это как систему с четырьмя процессорами для целей планирования.

```
[user@host ~]$ lscpu  
Architecture:           x86_64  
CPU op-mode(s):        32-bit, 64-bit  
Byte Order:            Little Endian  
CPU(s):                4  
On-line CPU(s) list:  0-3  
Thread(s) per core:   2  
Core(s) per socket:   2  
Socket(s):             1  
NUMA node(s):          1  
...output omitted...
```

На мгновение представьте, что единственный вклад в число загрузки вносят процессы, которым требуется процессорное время. Затем вы можете разделить отображаемые средние значения нагрузки на количество логических процессоров в системе. Значение ниже 1 указывает на удовлетворительное использование ресурсов и минимальное время ожидания. Значение выше 1 указывает на насыщение ресурсов и некоторую задержку обработки.

```
# From lscpu, the system has four logical CPUs, so divide by 4:  
#                                     load average: 2.92, 4.48, 5.20  
#         divide by number of logical CPUs:    4    4    4  
#                                         -----  
#                                     per-CPU load average: 0.73  1.12  1.30  
#  
# This system's load average appears to be decreasing.  
# With a load average of 2.92 on four CPUs, all CPUs were in use ~73% of the time.  
# During the last 5 minutes, the system was overloaded by ~12%.  
# During the last 15 minutes, the system was overloaded by ~30%.
```

Очередь простоявшего ЦП имеет номер загрузки 0. Каждый процесс, ожидающий ЦП, добавляет счетчик 1 к числу загрузки. Если на ЦП выполняется один процесс, номер загрузки равен единице, ресурс (ЦП) используется, но нет ожидающих запросов. Если этот процесс выполняется целую минуту, его вклад к минутной средней нагрузке будет 1.

Однако процессы, непрерывно спящие для критических операций ввода-вывода из-за загруженного диска или сетевого ресурса, также включаются в счет и увеличивают среднюю нагрузку. Хотя это и не является показателем использования ЦП, эти процессы добавляются в счетчик очереди, потому что они ожидают ресурсов и не могут выполняться на ЦП, пока не получат их. Это по-прежнему нагрузка на систему из-за ограничений ресурсов, из-за которых процессы не запускаются.

До насыщения ресурсов средняя загрузка остается ниже 1, поскольку задачи редко находятся в очереди. Средняя нагрузка увеличивается только тогда, когда из-за насыщения ресурсов запросы остаются в очереди и подсчитываются программой расчета нагрузки. Когда использование ресурсов приближается к 100%, каждый дополнительный запрос начинает испытывать время ожидания службы. Ряд дополнительных инструментов сообщает о средней загрузке, включая **w** и **top**.

МОНИТОРИНГ ПРОЦЕССОВ В РЕАЛЬНОМ ВРЕМЕНИ

Программа **top** представляет собой динамическое представление процессов системы, отображающее сводный заголовок, за которым следует список процессов или потоков, аналогичный информации **ps**. В отличие от статического вывода **ps**, **top** постоянно обновляется с настраиваемым интервалом и предоставляет возможности для изменения порядка столбцов, сортировки и выделения. Пользовательские конфигурации можно сохранять и делать постоянными.

Столбцы вывода по умолчанию можно узнать из других инструментов ресурсов:

- Идентификатор процесса (**PID**).
- Имя пользователя (**USER**) - владелец процесса.
- Виртуальная память (**VIRT**) - это вся память, которую использует процесс, включая резидентный набор, разделяемые библиотеки и любые отображаемые или заменяемые страницы памяти. (Обозначается **VSZ** в команде **ps**.)
- Резидентная память (**RES**) - это физическая память, используемая процессом, включая любые резидентные общие объекты. (Обозначается **RSS** в команде **ps**.)
- Состояние процесса (**S**) отображается как:
 - **D** = Бесперебойное спящее состояние
 - **R** = Работающий или запускаемый
 - **S** = Спящий
 - **T** = Остановлен или отложен
 - **Z** = Зомби

- Процессорное (**CPU**) время (**TIME**) - общее время обработки с момента запуска процесса. Может быть переключено, чтобы включать суммарное время всех предыдущих дочерних процессов.
- Имя команды процесса (**COMMAND**).

Основные значения при нажатии клавиш во время выполнения команды top

КЛЮЧ	ЦЕЛЬ
? или H	Помощь при интерактивном нажатии клавиш.
L, T, M	Переключает для строк заголовка нагрузки, потоков и памяти.
1	Переключить отображение отдельных ЦП или сводки по всем ЦП в заголовке.
S (1)	Измените частоту обновления (экрана) в десятичных секундах (например, 0,5, 1, 5).
B	Включить обратное выделение для запущенных процессов; по умолчанию выделено только жирным шрифтом.
B	Позволяет использовать полужирный шрифт при отображении, в заголовке и для запущенных процессов.
Shift + H	Переключить темы; показать сводку процесса или отдельные потоки.
U, Shift + U	Фильтр по любому имени пользователя (действующему, реальному).
Shift + M	Сортирует список процессов по использованию памяти в порядке убывания.
Shift + P	Сортирует список процессов по загрузке процессора в порядке убывания.
K (1)	Убить процесс. При появлении запроса введите PID, затем подайте сигнал.
R (1)	Возродите процесс. При появлении запроса введите PID, затем nice_value.
Shift + W	Запишите (сохраните) текущую конфигурацию дисплея для использования при следующем перезапуске сверху.
Q	Покинуть
Примечание:	(1) Недоступно, если верх был запущен в безопасном режиме. См. Вверху (1).



РЕКОМЕНДАЦИИ

Справочные страницы **manps(1)**, **top(1)**, **uptime(1)**, **iw(1)**

УПРАЖНЕНИЯ С ПОШАГОВЫМИ ИНСТРУКЦИЯМИ

МОНИТОРИНГ АКТИВНОСТИ ПРОЦЕССОВ

В этом упражнении вы будете использовать команду **top** для динамического изучения запущенных процессов и управления ими.

В РЕЗУЛЬТАТЕ

Вы должны иметь возможность управлять процессами в режиме реального времени.

ПРЕЖДЕ ЧЕМ ВЫ НАЧНЕТЕ

Войдите на рабочую станцию, как пользователь **student**, используя в качестве пароля слово **student**.

На рабочей станции запустите сценарий **lab processes-monitor start**. Команда запускает сценарий, который определяет, доступен ли хост, **servera**, в сети.

```
[student@workstation ~]$ lab processes-monitor start
```

1. На рабочей станции откройте рядом два окна терминала. Терминалы соответственно называются **левым** и **правым**. На каждом терминале используйте команду **ssh** для входа на **servera** в качестве пользователя - **student**.

```
[student@workstation ~]$ ssh student@servera
Activate the web console with: systemctl enable --now cockpit.socket

Last login: Fri Feb  8 12:04:52 2019 from 172.25.250.254
[student@servera ~]$
```

2. В левом окне создайте новый каталог с именем **/home/student/bin**. В новом каталоге создайте сценарий оболочки с именем **monitor**. Убедитесь, что сценарий исполняемый.

- 2.1. Используйте команду **mkdir**, чтобы создать новый каталог с именем **/home/student/bin**.

```
[student@servera ~]$ mkdir /home/student/bin
```

- 2.2. Используйте команду **vim**, чтобы создать сценарий с именем **monitor** в каталоге **/home/student/bin**. Нажмите клавишу **I**, чтобы войти в интерактивный режим **Vim**. Используйте команду **:wq**, чтобы сохранить файл.

```
[student@servera ~]$ vim /home/student/bin/monitor
#!/bin/bash
while true; do
    var=1
    while [[ var -lt 50000 ]]; do
        var=$((var+1))
    done
    sleep 1
done
```



ПРИМЕЧАНИЕ

Сценарий **monitor** выполняется до завершения. Он создает искусственную нагрузку на ЦП, выполняя пятьдесят тысяч задач сложения. Затем он засыпает на одну секунду, сбрасывает переменную и повторяет.

2.3. Используйте команду **chmod** с опцией **a+x**, чтобы сделать файл **monitor** исполняемым.

```
[student@servera ~]$ chmod a+x /home/student/bin/monitor
```

3. В оболочке правого терминала запустите утилиту **top**. Размер окна должен быть как можно более высоким.

```
[student@servera ~]$ top
top - 12:13:03 up 11 days, 58 min, 3 users, load average: 0.00, 0.00, 0.00
Tasks: 113 total, 2 running, 111 sleeping, 0 stopped, 0 zombie
%CPU(s): 0.2 us, 0.0 sy, 0.0 ni, 99.8 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
MiB Mem : 1829.4 total, 1377.3 free, 193.9 used, 258.2 buff/cache
MiB Swap: 1024.0 total, 1024.0 free, 0.0 used. 1476.1 avail Mem

PID USER      PR  NI      VIRT      RES      SHR S %CPU %MEM     TIME+ COMMAND
5861 root      20   0          0          0          0 I  0.3    0.0  0:00.71 kworker/1:3-
events
6068 student   20   0  273564  4300  3688 R  0.3    0.2  0:00.01 top
  1 root      20   0  178680 13424  8924 S  0.0    0.7  0:04.03 systemd
  2 root      20   0          0          0          0 S  0.0    0.0  0:00.03 kthreadd
  3 root      0 -20          0          0          0 I  0.0    0.0  0:00.00 rcu_gp
...output omitted...
```

4. В оболочке левого терминала используйте команду **lscpu**, чтобы определить количество логических процессоров на этой виртуальной машине.

```
[student@servera ~]$ lscpu
Architecture: x86_64
```

```
CPU op-mode(s): 32-bit, 64-bit  
Byte Order: Little Endian  
CPU(s): 2  
...output omitted...
```

5. В оболочке левого терминала запустите единственный экземпляр исполняемого файла **monitor**. Используйте амперсанд (&), чтобы запустить процесс в фоновом режиме.

```
[student@servera ~]$ monitor &  
[1] 6071
```

6. В оболочке правого терминала обратите внимание на вывод команды **top**. Используйте одиночные нажатия клавиш **L**, **T** и **M** для переключения строк заголовка нагрузки, потоков и памяти. После наблюдения за этим поведением убедитесь, что отображаются все заголовки.
7. Запишите идентификатор процесса (**PID**) исполняемого файла **monitor**. Просмотрите процентную долю ЦП для процесса, которая, как ожидается, колеблется от 15% до 20%.

```
[student@servera ~]$ top  
PID USER      PR  NI    VIRT    RES    SHR S %CPU %MEM     TIME+ COMMAND  
6071 student    20   0 222448   2964   2716 S 18.7  0.2  0:27.35 monitor  
...output omitted...
```

Просмотрите средние значения нагрузки. Средняя загрузка за одну минуту в настоящее время меньше значения 1. На наблюдаемое значение может влиять конкуренция за ресурсы со стороны другой виртуальной машины или виртуального хоста.

```
top - 12:23:45 up 11 days,  1:09,  3 users, load average: 0.21, 0.14, 0.05
```

8. В оболочке левого терминала запустите второй экземпляр исполняемого файла **monitor**. Используйте амперсанд (&), чтобы запустить процесс в фоновом режиме.

```
[student@servera ~]$ monitor &  
[2] 6498
```

9. В оболочке правого терминала запишите идентификатор процесса (**PID**) для второго процесса монитора. Просмотрите процентную долю ЦП для процесса, которая также может колебаться от 15% до 20%.

```
[student@servera ~]$ top  
PID USER      PR  NI    VIRT    RES    SHR S %CPU %MEM     TIME+ COMMAND  
6071 student    20   0 222448   2964   2716 S 19.0  0.2  1:36.53 monitor  
6498 student    20   0 222448   2996   2748 R 15.7  0.2  0:16.34 monitor  
...output omitted...
```

Снова просмотрите среднюю нагрузку за одну минуту, которая по-прежнему меньше 1. Важно подождать хотя бы одну минуту, чтобы расчет адаптировался к новой рабочей нагрузке.

10. В оболочке левого терминала запустите третий экземпляр исполняемого файла **monitor**. Используйте амперсанд (&), чтобы запустить процесс в фоновом режиме.

```
[student@servera ~]$ monitor &
[3] 6881
```

11. В оболочке правого терминала запишите идентификатор процесса (**PID**) для третьего процесса монитора. Просмотрите процентную долю ЦП для процесса, опять же ожидается, что она колеблется от 15% до 20%.

```
[student@servera ~]$ top
  PID USER      PR  NI    VIRT    RES    SHR S %CPU %MEM     TIME+ COMMAND
 6881 student    20   0 222448  3032  2784 S 18.6  0.2  0:11.48 monitor
 6498 student    20   0 222448  2996  2748 S 15.6  0.2  0:47.86 monitor
 6071 student    20   0 222448  2964  2716 S 18.1  0.2  2:07.86 monitor
```

Чтобы повысить среднюю нагрузку выше 1, необходимо запустить больше процессов мониторинга. Классная установка имеет 2 процессора, поэтому только трех процессов недостаточно, чтобы нагружать ее. Запустите еще три процесса **monitor**. Снова просмотрите среднюю нагрузку за одну минуту, которая теперь должна быть выше 1. Важно подождать не менее одной минуты, чтобы расчет адаптировался к новой рабочей нагрузке.

```
[student@servera ~]$ monitor &
[4] 10708
[student@servera ~]$ monitor &
[5] 11122
[student@servera ~]$ monitor &
[6] 11338
```

```
top - 12:42:32 up 11 days,  1:28,  3 users,  load average: 1.23, 2.50, 1.54
```

12. Закончив наблюдение за средними значениями нагрузки, завершите каждый из процессов монитора изнутри.

- 12.1. В оболочке правого терминала нажмите **K**. Обратите внимание на подсказку под заголовками и над столбцами.

```
...output omitted...
PID to signal/kill [default pid = 11338]
```

- 12.2. Приглашение (prompt) выбрало процессы мониторинга вверху списка. Нажмите **Enter**, чтобы остановить процесс.

```
...output omitted...
Send pid 11338 signal [15/sigterm]
```

12.3. Снова нажмите **Enter**, чтобы подтвердить сигнал **SIGTERM 15**.

Подтвердите, что выбранный процесс больше не наблюдается вверху. Если **PID** все еще остается, повторите завершающие шаги, заменив сигнал **SIGKILL 9** при появлении запроса.

PID	User	State	PPID	PCNT	CPUTIME	TIME	COMMAND	
6498	student	20	0	222448	2996	2748 R	22.9 0.2	5:31.47 monitor
6881	student	20	0	222448	3032	2784 R	21.3 0.2	4:54.47 monitor
11122	student	20	0	222448	2984	2736 R	15.3 0.2	2:32.48 monitor
6071	student	20	0	222448	2964	2716 S	15.0 0.2	6:50.90 monitor
10708	student	20	0	222448	3032	2784 S	14.6 0.2	2:53.46 monitor

13. Повторите предыдущий шаг для каждого оставшегося экземпляра процесса **monitor**. Убедитесь, что в выводе команды **top**не осталось процессов **monitor**.

14. В оболочке правого терминала нажмите **Q**, чтобы прекратить выполнение команды **top**. Выйти из **servera** в обоих окнах терминала.

```
[student@servera bin]$ exit
logout
Connection to servera closed.
[student@workstation ~]$
```

```
[student@servera ~]$ exit
logout
Connection to servera closed.
[student@workstation ~]$
```

Окончание

На рабочей станции запустите сценарий **lab processes-monitor finish**, чтобы завершить упражнение.

```
[student@workstation ~]$ lab processes-monitor finish
```

На этом пошаговое упражнение завершено.

ЛАБОРАТОРНАЯ РАБОТА

МОНИТОРИНГ И УПРАВЛЕНИЕ ПРОЦЕССАМИ LINUX

КОНТРОЛЬНЫЙ СПИСОК РАБОТЫ

В этой лабораторной работе вы будете находить и управлять процессами, которые используют больше всего ресурсов в системе. системе.

В РЕЗУЛЬТАТЕ

Вы должны уметь управлять процессами, используя команду **top**, как инструмент управления процессами.

ПРЕЖДЕ ЧЕМ ВЫ НАЧНЕТЕ

Войдите на рабочую станцию, как пользователь **student**, используя в качестве пароля слово **student**.

На рабочей станции запустите сценарий **lab processes-review start**. Команда запускает сценарий, чтобы определить, доступен ли хост **serverb** в сети.

```
[student@workstation ~]$ lab processes-review start
```

1. На рабочей станции откройте два окна терминала рядом друг с другом. В данном разделе эти терминалы будут называться левый и правый. В каждом окне терминала войдите на **серверb** как пользователь **student**.

Создайте сценарий с именем **process101**, который будет генерировать искусственную нагрузку на ЦП. Создайте сценарий в каталоге **/home/student/bin**.

```
#!/bin/bash
while true; do
    var=1
    while [[ var -lt 50000 ]]; do
        var=$((var+1))
    done
    sleep 1
done
```

2. В правом окне запускаем утилиту **top**.
3. В оболочке левого терминала определите количество логических процессоров на виртуальной машине. Запустите сценарий **process101** в фоновом режиме.

4. В оболочке правого терминала обратите внимание на отображение **top**. Переключение между нагрузкой, потоками и памятью. Обратите внимание на идентификатор процесса (**PID**) для **process101**. Просмотрите процентное соотношение ЦП. Он должен колебаться от 10% до 15%. После просмотра нагрузки, потоков и памяти убедитесь, что **top** показывает использование ЦП.
5. Отключите использование жирного шрифта на дисплее. Сохраните эту конфигурацию для повторного использования при перезапуске **top**. Подтвердите, что изменения сохранены.
6. Скопируйте сценарий **process101** в новый файл с именем **process102**. Отредактируйте сценарий, чтобы создать более искусственную нагрузку на ЦП. Увеличьте нагрузку с пятидесяти тысяч до ста тысяч. Запустите процесс **process102** в фоновом режиме.
7. В оболочке правого терминала убедитесь, что процесс запущен и использует больше всего ресурсов ЦП. Нагрузка должна колебаться между 25% и 35%.
8. Средняя загрузка все еще ниже 1. Скопируйте **process101** в новый сценарий с именем **process103**. Увеличьте счетчик сложения до восьмисот тысяч. Запустить **process103** в фоновом режиме. Убедитесь, что средняя нагрузка выше 1. Изменение средней нагрузки может занять несколько минут.
9. В оболочке левого терминала получите права пользователя **root**. Приостановите процесс **process101**. Перечислите оставшиеся задания. Обратите внимание, что состояние процесса для **process101** теперь равно **T**.
10. Возобновите процесс **process101**.
11. Завершите **process101**, **process102** и **process103** с помощью командной строки. Убедитесь, что процессы больше не отображаются в выводе команды **top**.
12. В оболочке левого терминала выйдите из оболочки пользователя **root**. В оболочке правого терминала остановите выполнение команды **top**. Выйти из **serverb** в обоих окнах.

Оценка

На рабочей станции запустите сценарий **labprocesses-reviewgrade**, чтобы подтвердить успех выполнения упражнения.

```
[student@workstation ~]$ lab processes-review grade
```

Завершение

На рабочей станции запустите сценарий **lab processes-review finish**, чтобы завершить лабораторную работу.

```
[student@workstation ~]$ lab processes-review finish
```

На этом лабораторная работа завершена.

РЕШЕНИЕ

МОНИТОРИНГ И УПРАВЛЕНИЕ ПРОЦЕССАМИ LINUX

КОНТРОЛЬНЫЙ СПИСОК РАБОТЫ

В этой лабораторной работе вы будете находить и управлять процессами, которые используют больше всего ресурсов в системе. системе.

В РЕЗУЛЬТАТЕ

Вы должны уметь управлять процессами, используя команду **top**, как инструмент управления процессами.

ПРЕЖДЕ ЧЕМ ВЫ НАЧНЕТЕ

Войдите на рабочую станцию, как пользователь **student**, используя в качестве пароля слово **student**.

На рабочей станции запустите сценарий **lab processes-review start**. Команда запускает сценарий, чтобы определить, доступен ли хост **serverb** в сети.

```
[student@workstation ~]$ lab processes-review start
```

1. На рабочей станции откройте два окна терминала рядом друг с другом. В данном разделе эти терминалы будут называться левый и правый. В каждом окне терминала войдите на **серверb** как пользователь **student**.

Создайте сценарий с именем **process101**, который будет генерировать искусственную нагрузку на ЦП. Создайте сценарий в каталоге **/home/student/bin**.

```
#!/bin/bash
while true; do
    var=1
    while [[ var -lt 50000 ]]; do
        var=$((var+1))
    done
    sleep 1
done
```

- 1.1. На рабочей станции откройте рядом два окна терминала. В каждом терминале используйте команду **ssh** для входа на **serverb** как пользователь **student**.

```
[student@workstation ~]$ ssh student@serverb  
Activate the web console with: systemctl enable --now cockpit.socket
```

```
Last login: Wed Feb 13 12:00:37 2019 from 172.25.250.254  
[student@serverb ~]$
```

- 1.2. В оболочке левого терминала используйте команду **mkdir** для создания каталога **/home/student/bin**.

```
[student@serverb ~]$ mkdir /home/student/bin
```

- 1.3. В оболочке левого терминала используйте команду **vim** для создания сценария **process101**. Нажмите клавишу **I**, чтобы войти в интерактивный режим. Введите:**wq**, чтобы сохранить файл.

```
[student@serverb ~]$ vim /home/student/bin/process101
```

```
#!/bin/bash  
while true; do  
    var=1  
    while [[ var -lt 50000 ]]; do  
        var=$((var+1))  
    done  
    sleep 1  
done
```

- 1.4. Сделайте скрипт **process101** исполняемым с помощью команды **chmod** и параметра **+x**.

```
[student@serverb ~]$ chmod +x /home/student/bin/process101
```

2. В правом окне запускаем утилиту **top**.

- 2.1. В правом окне запускаем утилиту **top**. Размер окна должен быть как можно более высоким.

```
[student@serverb ~]$ top  
top - 13:47:06 up 19 min, 2 users, load average: 0.00, 0.00, 0.00  
Tasks: 110 total, 1 running, 109 sleeping, 0 stopped, 0 zombie  
%Cpu(s): 0.0 us, 3.1 sy, 0.0 ni, 96.9 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st  
MiB Mem : 1829.4 total, 1439.1 free, 171.9 used, 218.4 buff/cache  
MiB Swap: 1024.0 total, 1024.0 free, 0.0 used. 1499.6 avail Mem  
  
 PID USER      PR  NI    VIRT    RES    SHR S %CPU %MEM     TIME   COMMAND  
  1 root      20   0 178536 13488 8996 S  0.0  0.7  0:01.15 systemd  
  2 root      20   0      0      0      0 S  0.0  0.0  0:00.00 kthreadd  
  3 root      0 -20      0      0      0 I  0.0  0.0  0:00.00 rcu_gp  
  4 root      0 -20      0      0      0 I  0.0  0.0  0:00.00 rcu_par_gp  
  6 root      0 -20      0      0      0 I  0.0  0.0  0:00.00 kworker/0:0H-  
kblockd  
...output omitted...
```

3. В оболочке левого терминала определите количество логических процессоров на виртуальной машине. Запустите сценарий **process101** в фоновом режиме.

3.1. Используйте команду **grep**, чтобы определить количество логических процессоров.

```
[student@serverb ~]$ grep "model name" /proc/cpuinfo | wc -l  
2
```

3.2. Используйте команду **cd**, чтобы перейти в каталог **/home/student/bin**. Запустите сценарий **process101** в фоновом режиме.

```
[student@serverb ~]$ cd /home/student/bin  
[student@serverb bin]$ process101 &  
[1] 1180
```

4. В оболочке правого терминала обратите внимание на отображение **top**. Переключение между нагрузкой, потоками и памятью. Обратите внимание на идентификатор процесса (**PID**) для **process101**. Просмотрите процентное соотношение ЦП. Он должен колебаться от 10% до 15%. После просмотра нагрузки, потоков и памяти убедитесь, что **top** показывает использование ЦП.

4.1. Нажмите **Shift + M**.

```
top - 13:56:24 up 28 min,  2 users,  load average: 0.21, 0.08, 0.02  
Tasks: 112 total,   2 running, 110 sleeping,   0 stopped,   0 zombie  
%Cpu(s):  5.8 us,  1.3 sy,  0.0 ni, 92.8 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st  
MiB Mem : 1829.4 total, 1438.1 free,   172.7 used,   218.6 buff/cache  
MiB Swap: 1024.0 total, 1024.0 free,     0.0 used. 1498.7 avail Mem  
  
 PID USER      PR  NI    VIRT    RES    SHR S %CPU %MEM     TIME+ COMMAND  
 705 root      20   0 409956  34880  33620 S  0.0  1.9  0:00.04 sssd_nss  
 706 root      20   0 454304  34472  14304 S  0.0  1.8  0:00.62 firewalld  
 725 root      20   0 611348  28244  14076 S  0.0  1.5  0:00.27 tuned  
 663 polkitd   20   0 1907312  23876  16040 S  0.0  1.3  0:00.04 polkitd  
 718 root      20   0 600316  17176  14832 S  0.0  0.9  0:00.06 NetworkManager  
...output omitted...
```



ПРИМЕЧАНИЕ

Обратите внимание, что, когда **top** переключается в режим фильтрации по использованию памяти, **process101** больше не является первым процессом. Вы можете нажать **Shift + P**, чтобы вернуться к использованию фильтрации по использованию ЦП.

4.2. Нажмите Shift + L.

```
top - 09:32:52 up 20:05,  2 users,  load average: 0.18, 0.10, 0.03
Tasks: 112 total,   2 running, 110 sleeping,   0 stopped,   0 zombie
%Cpu(s):  7.8/1.5      9[|||||]                                ]
MiB Mem : 18.3/1829.4 [||||||||||||||||||]                      ]
MiB Swap:  0.0/1024.0 [                                         ]
PID USER      PR  NI    VIRT     RES     SHR S %CPU %MEM     TIME+ COMMAND
 705 root      20   0  409956  34880  33620 S  0.0  1.9  0:00.04 sssd_nss
 706 root      20   0  454304  34472  14304 S  0.0  1.8  0:00.62 firewalld
 725 root      20   0  611348  28244  14076 S  0.0  1.5  0:00.30 tuned
 663 polkitd   20   0 1907312  23876  16040 S  0.0  1.3  0:00.04 polkitd
 718 root      20   0  600316  17176  14832 S  0.0  0.9  0:00.07 NetworkManager
...output omitted...
```

4.3. Нажмите T.

```
Tasks: 113 total,   2 running, 111 sleeping,   0 stopped,   0 zombie
%Cpu(s):  7.8/1.5      9[|||||]                                ]
MiB Mem : 1829.4 total,   1436.7 free,    173.7 used,   219.0 buff/cache
MiB Swap:  1024.0 total,   1024.0 free,     0.0 used.  1497.7 avail Mem

PID USER      PR  NI    VIRT     RES     SHR S %CPU %MEM     TIME+ COMMAND
1180 student   20   0  222448  3056  2808 S 12.0  0.2  1:59.94 process101
 705 root      20   0  409956  34880  33620 S  0.0  1.9  0:00.04 sssd_nss
 706 root      20   0  454304  34472  14304 S  0.0  1.8  0:00.62 firewalld
 725 root      20   0  611348  28244  14076 S  0.0  1.5  0:00.30 tuned
 663 polkitd   20   0 1907312  23876  16040 S  0.0  1.3  0:00.04 polkitd
 718 root      20   0  600316  17176  14832 S  0.0  0.9  0:00.07 NetworkManager
...output omitted...
```

4.4. Нажмите Shift+P.

```
top - 09:35:48 up 20:08,  2 users,  load average: 0.10, 0.10, 0.04
Tasks: 110 total,   4 running, 106 sleeping,   0 stopped,   0 zombie
%Cpu(s):  6.8 us,  1.0 sy,  0.0 ni, 92.2 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
MiB Mem : 18.3/1829.4 [||||||||||||||||||]                      ]
MiB Swap:  0.0/1024.0 [                                         ]

PID USER      PR  NI    VIRT     RES     SHR S %CPU %MEM     TIME+ COMMAND
27179 student   20   0  222448  3060  2812 R 15.6  0.2  1:13.47 process101
...output omitted...
```

- Отключите использование жирного шрифта на дисплее. Сохраните эту конфигурацию для повторного использования при перезапуске top. Подтвердите, что изменения сохранены.

5.1. Нажмите Shift+B, чтобы отключить использование полужирного шрифта.

```
top - 19:40:30 up 6:12, 2 users, load average: 0.11, 0.12, 0.09
Tasks: 112 total, 1 running, 111 sleeping, 0 stopped, 0 zombie
%Cpu(s): 7.6/1.5      9[|||||||||]
                                ]
MiB Mem : 18.2/1829.4 [|||||||||||||||||]
                                ]
MiB Swap: 0.0/1024.0 [
```

- 5.2.** Нажмите **Shift+W**, чтобы сохранить конфигурацию. Конфигурация по умолчанию хранится в **.toprc** в каталоге **/home/student/.config/procps**. В оболочке левого терминала убедитесь, что файл **toprc** существует.

```
[student@serverb bin]$ ls -l /home/student/.config/procps/toprc
-rw-rw-r--. 1 student student 966 Feb 18 19:45 /home/student/.config/procps/toprc
```

- 5.3.** В оболочке правого терминала выйдите в **top**, а затем перезапустите его. Убедитесь, что новый дисплей использует сохраненную конфигурацию.

```
top - 09:41:30 up 20:13, 2 users, load average: 0.22, 0.16, 0.08
Tasks: 112 total, 2 running, 110 sleeping, 0 stopped, 0 zombie
%Cpu(s): 6.8 us, 1.3 sy, 0.0 ni, 91.7 id, 0.0 wa, 0.2 hi, 0.0 si, 0.0 st
MiB Mem : 1829.4 total, 1432.0 free, 175.8 used, 221.7 buff/cache
MiB Swap: 1024.0 total, 1024.0 free, 0.0 used. 1495.3 avail Mem
```

- 6.** Скопируйте сценарий **process101** в новый файл с именем **process102**. Отредактируйте сценарий, чтобы создать более искусственную нагрузку на ЦП. Увеличьте нагрузку с пятидесяти тысяч до ста тысяч. Запустите процесс **process102** в фоновом режиме.

- 6.1.** В оболочке левого терминала используйте команду **cp**, чтобы скопировать **process101** в **process102**.

```
[student@serverb bin]$ cp process101 process102
```

- 6.2.** Используйте команду **vim** для редактирования сценария **process102**. Увеличьте количество задач сложения с пятидесяти тысяч до ста тысяч. Войдите в интерактивный режим с помощью **I**. Введите: **wq**, чтобы сохранить файл.

```
[student@serverb bin]$ vim process102
#!/bin/bash
while true; do
    var=1
    while [[ var -lt 100000 ]]; do
        var=$((var+1))
    done
    sleep 1
```

done

6.3. Запустите процесс **process102** в фоновом режиме.

```
[student@serverb bin]$ process102 &
[2] 20723
```

6.4. Используйте команду **jobs**, чтобы убедиться, что оба процесса работают в фоновом режиме.

```
[student@serverb bin]$ jobs
[1]- Running process101 &
[2]+ Running process102 &
```

7. В оболочке правого терминала убедитесь, что процесс запущен и использует больше всего ресурсов ЦП. Нагрузка должна колебаться между 25% и 35%.

7.1. В оболочке правого терминала убедитесь, что процесс запущен и использует больше всего ресурсов ЦП. Нагрузка должна колебаться между 25% и 35%.

```
top - 20:14:16 up 6:46, 2 users, load average: 0.58, 0.34, 0.18
Tasks: 112 total, 2 running, 110 sleeping, 0 stopped, 0 zombie
499 %Cpu(s): 0.0 us, 0.0 sy, 0.0 ni, 100.0 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
500 MiB Mem : 1829.4 total, 1428.7 free, 179.0 used, 221.8 buff/cache
501 MiB Swap: 1024.0 total, 1024.0 free, 0.0 used. 1492.1 avail Mem

PID USER PR NI VIRT RES SHR S %CPU %MEM TIME+ COMMAND
20723 student 20 0 222448 3016 2764 S 24.7 0.2 0:53.28 process102
1180 student 20 0 222448 3056 2808 S 12.0 0.2 58:01.56 process101
...output omitted...
```



ПРИМЕЧАНИЕ

Если вы не видите **process101** и **process102** в списке процессов вывода команды **top**, нажмите **Shift+P**, чтобы убедиться, что **top** отсортировано по загрузке ЦП.

8. Средняя загрузка все еще ниже 1. Скопируйте **process101** в новый сценарий с именем **process103**. Увеличьте счетчик сложения до восьмисот тысяч. Запустите **process103** в фоновом режиме. Убедитесь, что средняя нагрузка выше 1. Изменение средней нагрузки может занять несколько минут.

8.1. В оболочке правого терминала убедитесь, что средняя нагрузка выше 1.

```
top - 20:24:13 up 6:56, 2 users, load average: 0.43, 0.41, 0.29
...output omitted...
```

- 8.2. В оболочке левого терминала используйте команду **cp**, чтобы скопировать **process101** в новый сценарий с именем **process103**.

```
[student@serverb bin]$ cp process101 process103
```

- 8.3. В оболочке левого терминала используйте команду **vim** для редактирования сценария **process103**. Увеличьте счетчик сложения до восьмисот тысяч. Войдите в интерактивный режим с помощью клавиши **I**. Введите:**wq**, чтобы сохранить файл.

```
[student@serverb ~]$ vim /home/student/bin/process103
#!/bin/bash
while true; do
    var=1
    while [[ var -lt 800000]]; do
        var=$((var+1))
    done
    sleep 1
done
```

- 8.4. Запустить **process103** в фоновом режиме. Использование ЦП колеблется от 60% до 85%.

```
[student@serverb bin]$ process103 &
[3] 22751
```

- 8.5. Убедитесь, что все три задания выполняются в фоновом режиме.

```
[student@serverb bin]$ jobs
[1]  Running                  process101 &
[2]- Running                  process102 &
[3]+ Running                  process103 &
```

- 8.6. В правом окне терминала подтвердите, что средняя загрузка выше 1.

```
top - 20:45:34 up 7:17, 2 users, load average: 1.10, 0.90, 0.64
```

9. В оболочке левого терминала получите права пользователя **root**. Приостановите процесс **process101**. Перечислите оставшиеся задания. Обратите внимание, что состояние процесса для **process101** теперь равно **T**.

9.1. Используйте команду **su -**, чтобы стать пользователем **root**. Пароль - **redhat**.

```
[student@serverb bin]$ su -  
Password: redhat
```

9.2. Используйте команду **pkill** с параметром **-SIGSTOP**, чтобы приостановить процесс **process101**.

```
[root@serverb ~]# pkill -SIGSTOP process101
```

9.3. В оболочке правого терминала подтвердите, что **process101** больше не запущен.

```
top - 20:52:01 up 7:24, 2 users, load average: 1.19, 1.19, 0.89  
Tasks: 112 total, 2 running, 110 sleeping, 0 stopped, 0 zombie  
499 %Cpu(s): 0.0 us, 0.0 sy, 0.0 ni, 100.0 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0  
st  
500 MiB Mem : 1829.4 total, 1428.7 free, 179.0 used, 221.8 buff/cache  
501 MiB Swap: 1024.0 total, 1024.0 free, 0.0 used. 1492.1 avail Mem  
  
 PID USER PR NI VIRT RES SHR S %CPU %MEM TIME+ COMMAND  
24043 student 20 0 222448 2992 2744 R 66.1 0.2 6:59.50 process103  
20723 student 20 0 222448 3016 2764 R 29.9 0.2 11:04.84 process102  
...output omitted...
```

9.4. В оболочке левого терминала запустите команду **ps** с **jT**, чтобы просмотреть оставшиеся задания. Обратите внимание, что **process101** имеет статус **T**. Это означает, что процесс в настоящее время приостановлен.

```
[root@serverb ~]# ps jT  
PPID PID PGID SID TTY TPGID STAT UID TIME COMMAND  
...output omitted...  
27138 1180 1180 27138 pts/0 28558 T 1000 3:06 /bin/bash /home/student/  
bin/process101  
27138 20723 20723 27138 pts/0 28558 R 1000 1:23 /bin/bash /home/student/  
bin/process102  
27138 24043 24043 27138 pts/0 28558 R 1000 2:35 /bin/bash /home/student/  
bin/process103  
...output omitted...
```

10. Возобновите процесс **process101**.

- 10.1.** В оболочке левого терминала используйте команду **pkill** с параметром **-SIGCONT**, чтобы возобновить процесс **process101**.

```
[root@serverb ~]# pkill -SIGCONT process101
```

- 10.2.** В оболочке правого терминала подтвердите, что процесс снова запущен.

```
top - 20:57:02 up 7:29, 2 users, load average: 1.14, 1.20, 0.99
Tasks: 112 total, 2 running, 110 sleeping, 0 stopped, 0 zombie
%Cpu(s): 0.0 us, 0.0 sy, 0.0 ni, 100.0 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
MiB Mem : 1829.4 total, 1428.7 free, 179.0 used, 221.8 buff/cache
MiB Swap: 1024.0 total, 1024.0 free, 0.0 used. 1492.1 avail Mem

PID USER PR NI VIRT RES SHR S %CPU %MEM TIME+ COMMAND
24043 student 20 0 222448 2992 2744 R 66.8 0.2 10:40.61 process103
20723 student 20 0 222448 3016 2764 S 24.9 0.2 12:25.10 process102
1180 student 20 0 222448 3056 2808 S 17.9 0.2 64:07.99 process101
```

- 11.** Завершите **process101**, **process102** и **process103** с помощью командной строки. Убедитесь, что процессы больше не отображаются в выводе команды **top**.

- 11.1.** В оболочке левого терминала используйте команду **pkill**, чтобы завершить **process101**, **process102** и **process103**.

```
[root@serverb ~]# pkill process101
[root@serverb ~]# pkill process102
[root@serverb ~]# pkill process103
```

- 11.2.** В оболочке правого терминала убедитесь, что процессы больше не отображаются в выводе команды **top**.

```
top - 21:05:06 up 7:37, 2 users, load average: 1.26, 1.29, 1.12
Tasks: 112 total, 2 running, 110 sleeping, 0 stopped, 0 zombie
499 %Cpu(s): 0.0 us, 0.0 sy, 0.0 ni, 100.0 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
500 MiB Mem : 1829.4 total, 1428.7 free, 179.0 used, 221.8 buff/cache
501 MiB Swap: 1024.0 total, 1024.0 free, 0.0 used. 1492.1 avail Mem

PID USER PR NI VIRT RES SHR S %CPU %MEM TIME+ COMMAND
1 root 20 0 178536 13488 8996 S 0.0 0.7 0:01.21 systemd
2 root 20 0 0 0 0 S 0.0 0.0 0:00.00 kthreadd
3 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 rcu_gp
...output omitted...
```

- 12.** В оболочке левого терминала выйдите из оболочки пользователя **root**. В оболочке правого терминала остановите выполнение команды **top**. Выйти из **serverb** в обоих окнах.

12.1. Используйте команду **exit**, чтобы выйти из системы пользователя **root**.

```
[root@serverb ~]# exit  
logout  
[1]  Terminated      process101  
[2]  Terminated      process102  
[3]- Terminated      process103
```

12.2. Выйти из всех окон терминала.

```
[student@serverb bin]$ exit  
[student@workstation ~]$
```

12.3. В оболочке правого терминала используйте команду **Ctrl+C** для выхода.
Используйте команду **exit** для выхода.

```
[student@serverb ~]$ exit  
logout  
Connection to serverb closed.  
[student@workstation ~]$
```

Оценка

На рабочей станции запустите сценарий **labprocesses-reviewgrade**, чтобы подтвердить успех выполнения упражнения.

```
[student@workstation ~]$ lab processes-review grade
```

Завершение

На рабочей станции запустите сценарий **lab processes-review finish**, чтобы завершить лабораторную работу.

```
[student@workstation ~]$ lab processes-review finish
```

На этом лабораторная работа завершена.

РЕЗЮМЕ

В этой главе вы узнали:

- **Процесс** - это запущенный экземпляр исполняемой программы. Процессам присваивается состояние, которое может быть запущенным, спящим, остановленным или зомби (running, sleeping, stopped, или zombie). Команда **ps** используется для вывода списка процессов.
- **Каждый терминал** является отдельным сеансом и может иметь процесс переднего плана и независимые фоновые процессы. Команда **jobs** отображает процессы в рамках сеанса терминала.
- **Сигнал** - это программное прерывание, которое сообщает о событиях исполняющейся программе. Команды **kill**, **pkill** и **killall** используют сигналы для управления процессами.
- **Средняя нагрузка** - это оценка загруженности системы. Чтобы отобразить средние значения нагрузки, вы можете использовать команду **top**, **uptime** или **w**.

ГЛАВА 9

УПРАВЛЕНИЕ СЕРВИСАМИ И ДЕМОНАМИ

ЦЕЛЬ

Управляйте и отслеживайте сетевые службы и системные демоны с помощью **Systemd**.

ЗАДАЧИ

- Список системных демонов и сетевых служб, запущенных службой **systemd** и юнитами сокетов.
- Управление демонами системы и сетевыми службами с помощью **systemctl**.

РАЗДЕЛЫ

- Выявление автоматически запускаемых системных процессов (и упражнения с пошаговыми инструкциями)
- Управление системными службами (и упражнения с пошаговыми инструкциями)

ЛАБОРОТОРНАЯ РАБОТА

Управление сервисами и демонами.

ИДЕНТИФИКАЦИЯ АВТОМАТИЧЕСКИ ЗАПУСКАЕМЫХ СИСТЕМНЫХ ПРОЦЕССОВ

ЗАДАЧИ

После завершения этого раздела вы будете способны перечислить системные демоны и сетевые службы, запущенные службой **systemd** и юнитами сокетов.

ВВЕДЕНИЕ В **systemd**

Демон **systemd** управляет запуском Linux, включая запуск служб и управление службами в целом. Он активирует системные ресурсы, серверные демоны и другие процессы, как во время загрузки, так и в работающей системе.

Демоны - это процессы, которые либо ждут, либо работают в фоновом режиме, выполняя различные задачи. Обычно демоны запускаются автоматически во время загрузки и продолжают работать до завершения работы или до тех пор, пока они не будут остановлены вручную. По соглашению имена многих программ-демонов заканчиваются на букву **d**.

Служба **systemd** часто относится к одному или нескольким демонам, но запуск или остановка службы может вместо этого привести к однократному изменению состояния системы, которое не включает оставление процесса демона, запущенного после этого (так называемый **oneshot**).

В Red Hat Enterprise Linux первым запускаемым процессом (PID 1) является **systemd**. Некоторые из функций, предоставляемых **systemd**, включают:

- Возможности распараллеливания (одновременный запуск нескольких служб), которые увеличивают скорость загрузки системы.
- Запуск демонов по требованию без необходимости отдельной службы.
- Автоматическое управление зависимостями служб, которое может предотвратить длительные тайм-ауты. Например, зависящая от сети служба не будет пытаться запустить, пока сеть не станет доступной.
- Метод совместного отслеживания связанных процессов с помощью групп управления Linux.

ОПИСАНИЕ ЮНИТОВ СЕРВИСОВ

systemd использует юниты для управления различными типами объектов. Некоторые распространенные типы юнитов перечислены ниже:

- Сервисные юниты имеют расширение **.service** и представляют системные службы. Этот тип юнитов используется для запуска часто используемых демонов, таких как веб-сервер.
- Юниты **socket** имеют расширение **.socket** и представляют сокеты меж процессного взаимодействия (IPC), которые **systemd** должен отслеживать. Если клиент подключается к сокету, **systemd** запускает демон и передает ему соединение. Юниты сокетов используются для задержки запуска службы во время загрузки и для запуска менее часто используемых служб по запросу.

- Юниты пути (path) имеют расширение **.path** и используются для задержки активации службы до тех пор, пока не произойдет определенное изменение файловой системы. Их обычно используется для служб, которые используют каталоги **spool**, таких как система печати.

Команда **systemctl** используется для управления юнитами. Например, отобразите доступные типы юнитов с помощью команды **systemctl -t help**.



ВАЖНО

При использовании **systemctl** вы можете сокращать имена юнитов, записи дерева процессов и описания юнитов.

ПЕРЕЧЕНЬ СЕРВИСНЫХ ЮНИТОВ

Используйте команду **systemctl** для изучения текущего состояния системы. Например, следующая команда выводит список всех загруженных сервисных юнитов, команда **less** производит разбивку вывода на страницы для лучшего обзора.

```
[root@host ~]# systemctl list-units --type=service
UNIT                                     LOAD   ACTIVE SUB      DESCRIPTION
atd.service                               loaded active running Job spooling tools
auditd.service                            loaded active running Security Auditing Service
chronyd.service                           loaded active running NTP client/server
crond.service                            loaded active running Command Scheduler
dbus.service                             loaded active running D-Bus System Message Bus
...output omitted...
```

Приведенный выше вывод ограничивает тип единиц, перечисленных до обслуживающих единиц с параметром **--type=service**. На выходе представлены следующие столбцы:

UNIT

Название сервисного юнита.

LOAD (НАГРУЗКА)

Правильно ли **systemd** проанализировал конфигурацию юнитов и загрузил его в память.

ACTIVE(АКТИВНЫЙ)

Состояние активации высокого уровня юнита. Эта информация указывает, успешно ли запустился юнит.

SUB

Низкоуровневое состояние активации устройства. Эта информация указывает на более подробную информацию о юните. Информация различается в зависимости от типа, состояния и того, как юнит работает.

DESCRIPTION(ОПИСАНИЕ)

Краткое описание юнита.

По умолчанию команда **systemctl list-units --type=service** перечисляет только служебные юниты в активные состояния. Параметр **--all** перечисляет все служебные юниты независимо от состояния активации. Используйте параметр **--state=** для фильтрации по значениям поля **LOAD**, **ACTIVE** или **SUB**.

```
[root@host ~]# systemctl list-units --type=service --all
UNIT                      LOAD   ACTIVE   SUB     DESCRIPTION
atd.service                loaded  active   running Job spooling tools
audited.service             loaded  active   running Security Auditing ...
auth-rpcgss-module.service loaded  inactive dead    Kernel Module ...
chronyd.service             loaded  active   running NTP client/server
cpupower.service            loaded  inactive dead    Configure CPU power ...
crond.service               loaded  active   running Command Scheduler
dbus.service                loaded  active   running D-Bus System Message Bus
● display-manager.service   not-found inactive dead    display-manager.service
...output omitted...
```

Команда **systemctl** без аргументов выводит список как загруженных, так и активных юнитов.

```
[root@host ~]# systemctl
UNIT                      LOAD   ACTIVE   SUB     DESCRIPTION
proc-sys-fs-binfmt_misc.automount  loaded  active waiting Arbitrary...
sys-devices-....device          loaded  active plugged Virtio network...
sys-subsystem-net-devices-ens3.device loaded  active plugged Virtio network...
...
-.mount                     loaded  active mounted Root Mount
boot.mount                  loaded  active mounted /boot
...
systemd-ask-password-plymouth.path loaded  active waiting Forward Password...
systemd-ask-password-wall.path  loaded  active waiting Forward Password...
init.scope                  loaded  active running System and Servi...
session-1.scope              loaded  active running Session 1 of...
atd.service                 loaded  active running Job spooling tools
auditd.service               loaded  active running Security Auditing...
chronyd.service              loaded  active running NTP client/server
crond.service                loaded  active running Command Scheduler
...output omitted...
```

Команда **systemctl list-units** отображает юниты, которые служба **systemd** пытается проанализировать и загрузить в память; он не отображает установленные, но не включенные службы. Чтобы увидеть состояние всех установленных файлов модулей, используйте команду **systemctl list-unit-files**. Например:

```
[root@host ~]# systemctl list-unit-files --type=service
```

UNIT FILE	STATE
arp-ethers.service	disabled
atd.service	enabled
auditd.service	enabled
auth-rpcgss-module.service	static
autovt@.service	enabled
blk-availability.service	disabled
...output omitted...	

Вывод команды **systemctl list-units-files --type=service** допустимые записи для поля **STATE**: **enabled** (включены), **disabled** (отключены), **static** (статичны) и **masked** (замаскированы).

ПРОСМОТР СОСТОЯНИЯ СЕРВИСОВ

Просмотрите статус конкретного юнита с помощью команды **systemctl status name.type**. Если тип юнита не указан, **systemctl** покажет статус юнита сервиса, если таковой существует.

```
[root@host ~]# systemctl status sshd.service
```

```
● sshd.service - OpenSSH server daemon
   Loaded: loaded (/usr/lib/systemd/system/sshd.service; enabled; vendor preset: enabled)
     Active: active (running) since Thu 2019-02-14 12:07:45 IST; 7h ago
       Main PID: 1073 (sshd)
          CGroup: /system.slice/sshd.service
                  └─1073 /usr/sbin/sshd -D ...

Feb 14 11:51:39 host.example.com systemd[1]: Started OpenSSH server daemon.
Feb 14 11:51:39 host.example.com sshd[1073]: Could not load host key: /et...y
Feb 14 11:51:39 host.example.com sshd[1073]: Server listening on 0.0.0.0 ....
Feb 14 11:51:39 host.example.com sshd[1073]: Server listening on :: port 22.
Feb 14 11:53:21 host.example.com sshd[1270]: error: Could not load host k...y
Feb 14 11:53:22 host.example.com sshd[1270]: Accepted password for root f...2
...output omitted...
```

Эта команда отображает текущий статус службы. Значения полей:

Информация о сервисном центре

ПОЛЕ	ОПИСАНИЕ
Loaded (Загружено)	Загружен ли служебный юнит в память.
Active (Активный)	Работает ли сервисный юнит, и если да, то как долго он работает.
Main PID	ID основного процесса службы, включая имя команды.
Status (Статус)	Дополнительная информация о сервисе.

В выводе статуса можно найти несколько ключевых слов, указывающих состояние службы:

Состояния сервисов в выводе systemctl

КЛЮЧЕВОЕ СЛОВО	ОПИСАНИЕ
loaded (загружен)	Файл конфигурации объекта обработан.
active (активный) running (выполняется)	Запуск с одним или несколькими продолжающимися процессами.
active (активен) exited (вышел)	Успешно завершена разовая конфигурация.
active (активный) waiting (ожидающий)	Выполняется, но ждёт события.
inactive (неактивный)	Не выполняется.
enabled (включено)	Запускается во время загрузки.
disabled (отключен)	Не настроен на запуск во время загрузки.
static (статический)	Не может быть включен, но может быть запущен включенным устройством автоматически.



ПРИМЕЧАНИЕ

Команды `systemctl status NAME` заменяет команду `service NAME status`, используемую в Red Hat Enterprise Linux 6 и ранее.

ПРОВЕРКА СТАТУСА СЕРВИСОВ

Команда `systemctl` предоставляет методы для проверки конкретных состояний службы. Например, используйте следующую команду, чтобы убедиться, что сервисный юнит в настоящее время **active (running)** (активен (работает)):

```
[root@host ~]# systemctl is-active sshd.service  
active
```

Команда возвращает состояние юнита сервиса, которая обычно активен или неактивен.

Выполните следующую команду, чтобы проверить, разрешен ли автоматический запуск служебного юнита во время загрузки системы:

```
[root@host ~]# systemctl is-enabled sshd.service  
enabled
```

Команда возвращает, разрешен ли запуск служебного модуля во время загрузки, что обычно **enabled** (включено) или **disabled** (отключено).

Чтобы проверить, не вышел ли юнит из строя во время запуска, выполните следующую команду:

```
[root@host ~]# systemctl is-failed sshd.service  
active
```

Команда либо возвращает **active** (активную), если она работает правильно, либо **failed** (не работает), если во время запуска произошла ошибка. В случае, если юнит был остановлен, возвращается значение **unknown** (неизвестно) или **inactive** (неактивно).

Чтобы вывести список всех отказавших юнитов, выполните команду **systemctl --failed --type=service**.



РЕКОМЕНДАЦИИ

Справочные страницы **man systemd(1)**,
systemd.unit(5),
systemd.socket(5), и **systemctl(1)**

Дополнительные сведения см. В главе «Управление службами с помощью **systemd**» в бета-версии Red Hat Enterprise Linux 8.0 Настройка основных параметров системы по адресу https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/8-beta/html-single/configuring_basic_system_settings/managing-services-withsystemd

УПРАЖНЕНИЯ С ПОШАГОВЫМИ ИНСТРУКЦИЯМИ

ИДЕНТИФИКАЦИЯ АВТОМАТИЧЕСКИ ЗАПУСКАЕМЫХ СИСТЕМНЫХ ПРОЦЕССОВ

В этом упражнении вы составите список установленных сервисных юнитов и определите, какие сервисы в настоящее время включены и активны на сервере.

РЕЗУЛЬТАТЫ

Вы должны иметь возможность составить список установленных сервисных юнитов и идентифицировать активные и включенные сервисы в системе.

ПРЕЖДЕ ЧЕМ ВЫ НАЧНЕТЕ

Войдите на рабочую станцию как пользователь **student**, используя в качестве пароля слово **student**.

С рабочей (**workstation**) станции выполните команду **lab services-identify start**. Команда запускает сценарий запуска, который определяет, доступен ли **хост**, **servera**, в сети.

```
[student@workstation ~]$ lab services-identify start
```

- Используйте команду **ssh** для входа на **servera** как пользователь **student**. Системы настроены на использование ключей **SSH** для аутентификации, поэтому пароль для входа на сервер не требуется.

```
[student@workstation ~]$ ssh student@servera  
[student@servera ~]$
```

- Вывести список всех сервисных юнитов, установленных на **servera**.

```
[student@servera ~]$ systemctl list-units --type=service  
UNIT           LOAD   ACTIVE SUB   DESCRIPTION  
atd.service    loaded  active  running Job spooling tools  
auditd.service loaded  active  running Security Auditing Service  
chronyd.service loaded  active  running NTP client/server  
crond.service  loaded  active  running Command Scheduler  
dbus.service    loaded  active  running D-Bus System Message Bus  
...output omitted...
```

Нажмите **Q**, чтобы выйти из команды.

- Перечислите все юниты сокетов, **active** (активные) и **inactive** (неактивные), на сервере **servera**.

```
[student@servera ~]$ systemctl list-units --type=socket --all
UNIT                  LOAD   ACTIVE    SUB      DESCRIPTION
dbus.socket           loaded  active   running  D-Bus System Message Bus Socket
dm-event.socket       loaded  active   listening Device-mapper event daemon FIFOs
lvm2-lvmpolld.socket loaded  active   listening LVM2 poll daemon socket
...output omitted...
systemd-udevd-control.socket  loaded  active   running  udev Control Socket
systemd-udevd-kernel.socket   loaded  active   running  udev Kernel Socket

LOAD   = Reflects whether the unit definition was properly loaded.
ACTIVE = The high-level unit activation state, i.e. generalization of SUB.
SUB    = The low-level unit activation state, values depend on unit type.

12 loaded units listed.

To show all installed unit files use 'systemctl list-unit-files'.
```

4. Изучите статус службы **chrony**. Эта услуга используется для синхронизации времени сети (NTP).

4.1. Показать статус службы **chrony**. Запишите ID процесса всех активных демонов.

```
[student@servera ~]$ systemctl status chronyd
● chronyd.service - NTP client/server
  Loaded: loaded (/usr/lib/systemd/system/chronyd.service; enabled; vendor
  preset: enabled)
  Active: active (running) since Wed 2019-02-06 12:46:57 IST; 4h 7min ago
    Docs: man:chronyd(8)
          man:chrony.conf(5)
  Process: 684 ExecStartPost=/usr/libexec/chrony-helper update-daemon
  (code=exited, status=0/SUCCESS)
  Process: 673 ExecStart=/usr/sbin/chronyd $OPTIONS (code=exited, status=0/
  SUCCESS)
 Main PID: 680 (chronyd)
   Tasks: 1 (limit: 11406)
  Memory: 1.5M
  CGroup: /system.slice/chronyd.service
          └─680 /usr/sbin/chronyd

... jegui.ilt.example.com systemd[1]: Starting NTP client/server...
...output omitted...
... jegui.ilt.example.com systemd[1]: Started NTP client/server.
... servera.lab.example.com chronyd[680]: Source 172.25.254.254 offline
... servera.lab.example.com chronyd[680]: Source 172.25.254.254 online
... servera.lab.example.com chronyd[680]: Selected source 172.25.254.254
```

- 4.2.** Убедитесь, что перечисленные демоны работают. В приведенной выше команде вывод идентификатора процесса, связанного со службой **chrony**, равен **680**. Идентификатор процесса может отличаться в вашей системе.

```
[student@servera ~]$ ps -p 680
 PID TTY      TIME CMD
 680 ?        00:00:00 chrony
```

- 5.** Изучите статус службы **sshd**. Эта услуга используется для безопасной зашифрованной связи между системами.

- 5.1.** Определите, разрешен ли запуск службы **sshd** при загрузке системы.

```
[student@servera ~]$ systemctl is-enabled sshd
Enabled
```

- 5.2.** Определите, активна ли служба **sshd**, не отображая всю информацию о состоянии.

```
[student@servera ~]$ systemctl is-active sshd
active
```

- 5.3.** Показать статус службы **sshd**.

```
[student@servera ~]$ systemctl status sshd
● sshd.service - OpenSSH server daemon
  Loaded: loaded (/usr/lib/systemd/system/sshd.service; enabled; vendor preset: enabled)
  Active: active (running) since Wed 2019-02-06 12:46:58 IST; 4h 21min ago
    Docs: man:sshd(8)
          man:sshd_config(5)
  Main PID: 720 (sshd)
     Tasks: 1 (limit: 11406)
   Memory: 5.8M
      CGroup: /system.slice/sshd.service
           └─720 /usr/sbin/sshd -D -oCiphers=aes256-gcm@openssh.com,
              chacha20-poly1305@openssh.com,aes256-ctr,
              aes256-cbc,aes128-gcm@openssh.com,aes128-ctr,
              aes128-cbc -oMACs=hmac-sha2-256-etm@openssh.com,hmac-sha>

... jegui.ilt.example.com systemd[1]: Starting OpenSSH server daemon...
... servera.lab.example.com sshd[720]: Server listening on 0.0.0.0 port 22.
... servera.lab.example.com systemd[1]: Started OpenSSH server daemon.
... servera.lab.example.com sshd[720]: Server listening on :: port 22.
... output omitted...
... servera.lab.example.com sshd[1380]: pam_unix(sshd:session): session opened for
user student by (uid=0)
```

Нажмите **Q**, чтобы выйти из команды.

6. Перечислите включенные или отключенные состояния всех сервисных единиц.

```
[student@servera ~]$ systemctl list-unit-files --type=service
UNIT FILE                      STATE
arp-ethers.service              disabled
atd.service                     enabled
auditd.service                  enabled
auth-rpcgss-module.service     static
autovt@.service                 enabled
blk-availability.service        disabled
chrony-dnssrv@.service         static
chrony-wait.service             disabled
chronyd.service                 enabled
...output omitted...
```

Нажмите **Q**, чтобы выйти из команды.

7. Выход из сервера **servera**.

ЗАВЕРШЕНИЕ

На рабочей станции (**workstation**) запустите сценарий завершения **lab services-identify finish**, чтобы завершить это упражнение.

```
[student@workstation ~]$ lab services-identify finish
```

На этом пошаговое упражнение завершено.

УПРАВЛЕНИЕ СИСТЕМНЫМИ СЕРВИСАМИ

ЗАДАЧИ

После завершения этого раздела вы сможете управлять системными демонами и сетевыми службами с помощью **systemctl**.

ЗАПУСК И ОСТАНОВКА СЕРВИСОВ

ЗАДАЧИ

После завершения этого раздела вы сможете управлять системными демонами и сетевыми службами с помощью **systemctl**.

ЗАПУСК И ОСТАНОВКА СЕРВИСОВ

Службы необходимо останавливать или запускать вручную по ряду причин: возможно, службу необходимо обновить; файл конфигурации может потребоваться изменить; или может потребоваться удалить службу; или администратор может вручную запустить редко используемую службу.

Чтобы запустить службу, сначала убедитесь, что она не запущена, с помощью команды **systemctl status**. Затем используйте команду **systemctl start** от имени пользователя **root** (при необходимости с помощью **sudo**). В приведенном ниже примере показано, как запустить службу **sshd.service**:

```
[root@host ~] # systemctl start sshd.service
```

Служба **systemd** ищет файлы **.service** для управления службами в командах при отсутствии типа службы с именем службы. Таким образом, указанная выше команда может быть выполнена как:

```
[root@host ~] # systemctl start sshd
```

Чтобы остановить работающую в данный момент службу, используйте аргумент **stop** с командой **systemctl**. В приведенном ниже примере показано, как остановить службу **sshd.service**:

```
[root@host ~]# systemctl stop sshd.service
```

ПЕРЕЗАПУСК И ПЕРЕЗАГРУЗКА СЕРВИСОВ

Во время перезапуска работающей службы служба останавливается, а затем запускается. При перезапуске службы идентификатор процесса изменяется, и во время запуска связывается новый идентификатор процесса. Чтобы перезапустить работующую службу, используйте аргумент **restart** с командой **systemctl**. В приведенном ниже примере показано, как перезапустить службу **sshd.service**:

```
[root@host ~]# systemctl restart sshd.service
```

Некоторые службы могут перезагружать свои файлы конфигурации **без перезапуска**. Этот процесс **называется перезагрузкой службы**. При перезагрузке службы не изменяется идентификатор процесса, связанный с различными процессами службы. Чтобы перезагрузить работующую службу, используйте аргумент **reload** с командой **systemctl**. В приведенном ниже примере показано, как перезагрузить службу **sshd.service** после изменения конфигурации:

```
root@host ~]# systemctl reload sshd.service
```

Если вы не уверены, есть ли у службы возможность перезагрузки изменений файла конфигурации, используйте аргумент **reload-or-restart** с командой **systemctl**. Команда перезагружает изменения конфигурации, если доступна функция перезагрузки. В противном случае команда перезапускает службу для реализации новых изменений конфигурации:

```
[root@host ~]# systemctl reload-or-restart sshd.service
```

СПИСОК ЗАВИСИМОСТЕЙ ЮНИТОВ

Некоторые службы требуют, до своего запуска, запуск других служб, что создает зависимости от других служб. Другие службы запускаются не во время загрузки, а только по запросу. В обоих случаях **systemd** и **systemctl** запускают службы по мере необходимости, чтобы разрешить зависимость или запустить редко используемую службу. Например, если служба печати CUPS не запущена, а файл помещен в каталог очереди печати, то система запустит связанные с CUPS демоны или команды для удовлетворения запроса на печать.

```
[root@host ~]# systemctl stop cups.service
Warning: Stopping cups, but it can still be activated by:
  cups.path
  cups.socket
```

Чтобы полностью остановить службы печати в системе, остановите все три юнита. Отключение службы отключает зависимости.

Команда **systemctl list-dependencies UNIT** отображает иерархическое отображение зависимостей для запуска юнита службы. Чтобы перечислить обратные зависимости (юниты, которые зависят от указанного юнита), используйте командой опцию **--reverse**.

```
[root@host ~]# systemctl list-dependencies sshd.service
```

УСЛУГИ МАСКИРОВАНИЯ И РАСКРЫТИЯ МАСКИРОВКИ

Иногда в системе могут быть установлены разные службы, которые конфликтуют друг с другом. Например, существует несколько методов управления почтовыми серверами (например, **postfix** и **sendmail**). Маскирование службы предотвращает случайный запуск администратором службы, конфликтующей с другими. Маскирование создает ссылку в каталогах конфигурации на файл **/dev/null**, что предотвращает запуск службы.

```
[root@host ~]# systemctl mask sendmail.service  
Created symlink /etc/systemd/system/sendmail.service → /dev/null.
```

```
[root@host ~]# systemctl list-unit-files --type=service
```

UNIT FILE	STATE
...output omitted...	
sendmail.service	masked
...output omitted...	

Попытка запустить замаскированный сервисный юнит завершается ошибкой со следующим выводом:

```
[root@host ~]# systemctl start sendmail.service  
Failed to start sendmail.service: Unit sendmail.service is masked.
```

Используйте команду **systemctl unmask**, чтобы демаскировать служебный юнит.

```
[root@host ~]# systemctl unmask sendmail  
Removed /etc/systemd/system/sendmail.service.
```



ВАЖНО

Отключенная служба может быть запущена вручную или с помощью других файлов юнитов, но она не запускается автоматически при загрузке. Маскированная служба **не запускается вручную или автоматически**.

РАЗРЕШЕНИЕ СЕРВИСОВ ДЛЯ ЗАПУСКА ИЛИ ОСТАНОВКИ ПРИ ЗАГРУЗКЕ СИСТЕМЫ

Запуск службы в работающей системе не гарантирует, что служба автоматически запускается при перезагрузке системы. Точно так же остановка службы в работающей системе не препятствует ее повторному запуску при перезагрузке системы. Создание ссылок в каталогах конфигурации **systemd** позволяет службе запускаться при загрузке. Команды **systemctl** создают и удаляют эти ссылки.

Чтобы служба запустилась при загрузке, используйте команду **systemctl enableservice**.

```
[root@root ~]# systemctl enable sshd.service
Created symlink /etc/systemd/system/multi-user.target.wants/sshd.service → /usr/
lib/systemd/system/sshd.service.
```

Приведенная выше команда создает символическую ссылку из файла служебного юнита, обычно в каталоге **/usr/lib/systemd/system**, на место на диске, где **systemd** ищет файлы, то есть в каталоге **/etc/systemd/system/TARGETNAME.target.wants**. Включение службы не запускает службу в текущем сеансе. Чтобы запустить службу и включить ее автоматический запуск во время загрузки, выполните команды **systemctl start** и **systemctlenable**.

Чтобы отключить автоматический запуск службы, используйте следующую команду, которая удаляет символическую ссылку, созданную при включении службы. Обратите внимание, что отключение службы не останавливает ее.

```
[root@host ~]# systemctl disable sshd.service
Removed /etc/systemd/system/multi-user.target.wants/sshd.service.
```

Чтобы проверить, включена или отключена служба, используйте команду **systemctlis-enabled**.

РЕЗЮМЕКОМАНД **systemctl**

Службы можно запускать и останавливать в работающей системе, а также включать или отключать автоматический запуск во время загрузки.

Полезные команды управления сервисами

ЗАДАЧА	КОМАНДА
Просмотр подробной информации о состоянии юнита.	systemctl status <i>UNIT</i>
Остановите службу в работающей системе.	systemctl stop <i>UNIT</i>
Запустите службу в работающей системе.	systemctl start <i>UNIT</i>
Перезапустите службу в работающей системе.	systemctl restart <i>UNIT</i>
Перезагрузите файл конфигурации работающей службы.	systemctl reload <i>UNIT</i>
Полностью отключите запуск службы как вручную, так и при загрузке.	systemctl mask <i>UNIT</i>
Сделайте скрытый сервис доступным.	systemctl unmask <i>UNIT</i>
Настройте службу для запуска во время загрузки.	systemctl включить БЛОК
Отключить запуск службы во время загрузки.	systemctl enable <i>UNIT</i>
Перечислить юниты, необходимые и желаемые для выполнения необходимого юнита.	systemctl list-dependencies <i>UNIT</i>



РЕКОМЕНДАЦИИ

Справочные страницы **man systemd(1)**, **systemd.unit(5)**, **systemd.service(5)**, **systemd.socket(5)**, **isystemctl(1)**

Дополнительные сведения см. В главе «Управление службами с помощью systemd» в бета-версии Red Hat Enterprise Linux 8.0 «Настройка основных параметров системы» по адресу

https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/8-beta/html-single/configuring_basic_system_settings/managing-services-with-systemd

УПРАЖНЕНИЯ С ПОШАГОВЫМИ ИНСТРУКЦИЯМИ

В этом упражнении вы будете использовать **systemctl** для остановки, запуска, перезапуска, перезагрузки, включения и отключения службы, управляемой **systemd**.

В РЕЗУЛЬТАТЕ

Вы должны иметь возможность использовать команду **systemctl** для управления службами **systemd**.

ПРЕЖДЕ ЧЕМ ВЫ НАЧНЕТЕ

Войдите на рабочую станцию, как пользователь **student**, используя в качестве пароля слово **student**.

С рабочей станции (**workstation**) выполните команду **lab services-control start**. Команда запускает сценарий, который определяет, доступен ли хост, **servera**, в сети. Сценарий также гарантирует, что службы **sshd** и **chrony** работают на сервере **servera**

```
[student@workstation ~]$ lab services-control start
```

- Используйте команду **ssh** для входа на сервер **servera**, как пользователь **student**. Системы настроены на использование ключей **SSH** для аутентификации, поэтому пароль не требуется.

```
[student@workstation ~]$ ssh student@servera  
[student@servera ~]$
```

- Выполните команды **systemctl restart** и **systemctl reload** для службы **sshd**. Обратите внимание на разные результаты выполнения этих команд.
 - Показать статус службы **sshd**. Обратите внимание на идентификатор процесса демона **sshd**.

```
[student@servera ~]$ systemctl status sshd  
● sshd.service - OpenSSH server daemon  
   Loaded: loaded (/usr/lib/systemd/system/sshd.service; enabled; vendor preset:  
           enabled)  
     Active: active (running) since Wed 2019-02-06 23:50:42 EST; 9min ago  
       Docs: man:sshd(8)  
             man:sshd_config(5)  
 Main PID: 759 (sshd)  
    Tasks: 1 (limit: 11407)  
   Memory: 5.9M  
...output omitted...
```

Нажмите **Q**, чтобы выйти из команды.

- 2.2. Перезапустите службу **sshd** и просмотрите статус. Идентификатор процесса демона должен измениться. В выходных данных выше идентификатор процесса изменился с **759** на **1132** (в вашей системе числа, вероятно, будут другими).

```
[student@servera ~]$ sudo systemctl restart sshd
[sudo] password for student: student
[student@servera ~]$ systemctl status sshd
● sshd.service - OpenSSH server daemon
  Loaded: loaded (/usr/lib/systemd/system/sshd.service; enabled; vendor preset: enabled)
  Active: active (running) since Wed 2019-02-06 23:50:42 EST; 9min ago
    Docs: man:sshd(8)
          man:sshd_config(5)
  Main PID: 1132 (sshd)
    Tasks: 1 (limit: 11407)
   Memory: 5.9M
...output omitted...
```

Нажмите **Q**, чтобы выйти из команды.

- 2.3. Перезагрузите службу **sshd** и просмотрите статус. Идентификатор процесса демона не должен изменяться, и соединения не прерываются.

```
[student@servera ~]$ sudo systemctl reload sshd
[student@servera ~]$ systemctl status sshd
● sshd.service - OpenSSH server daemon
  Loaded: loaded (/usr/lib/systemd/system/sshd.service; enabled; vendor preset: enabled)
  Active: active (running) since Wed 2019-02-06 23:50:42 EST; 9min ago
    Docs: man:sshd(8)
          man:sshd_config(5)
  Main PID: 1132 (sshd)
    Tasks: 1 (limit: 11407)
   Memory: 5.9M
...output omitted...
```

Нажмите **Q**, чтобы выйти из команды.

3. Убедитесь, что служба **chrony** запущена.

```
[student@servera ~]$ systemctl status chronyd
```

```
● chronyd.service - NTP client/server
  Loaded: loaded (/usr/lib/systemd/system/chronyd.service; enabled; vendor
  preset: enabled)
    Active: active (running) since Wed 2019-02-06 23:50:38 EST; 1h 25min ago
      ...output omitted...
   ... servera.lab.example.com systemd[1]: Started NTP client/server.
   ... servera.lab.example.com chronyd[710]: Selected source 172.25.254.254
   ... servera.lab.example.com chronyd[710]: System clock wrong by 1.349113 seconds,
adjustment started
```

4. Остановите службу **chronyd** и просмотрите статус.

```
[student@servera ~]$ sudo systemctl stop chronyd
[sudo] password for student: student
[student@servera ~]$ systemctl status chronyd
● chronyd.service - NTP client/server
  Loaded: loaded (/usr/lib/systemd/system/chronyd.service; enabled; vendor
  preset: enabled)
    Active: inactive (dead) since Thu 2019-02-07 01:20:34 EST; 44s ago
      ...output omitted...
   ... servera.lab.example.com chronyd[710]: System clock wrong by 1.349113 seconds,
adjustment started
   ... servera.lab.example.com systemd[1]: Stopping NTP client/server...
   ... servera.lab.example.com systemd[1]: Stopped NTP client/server.
```

5. Определите, разрешен ли запуск службы **chronyd** при загрузке системы.

```
[student@server ~]$ systemctl is-enabled chronyd
enabled
```

6. Перезагрузите сервер **servera**, затем просмотрите статус службы **chronyd**.

```
[student@servera ~]$ sudo reboot
[sudo] password for student: student
Connection to servera closed by remote host.
Connection to servera closed.
[student@workstation ~]$
```

Войдите в систему как пользователь **student** на сервере **servera** и просмотрите состояние сервера **chronyd**.

```
[student@workstation ~]$ ssh student@servera
[student@servera ~]$ systemctl status chronyd
```

```
● chronyd.service - NTP client/server
  Loaded: loaded (/usr/lib/systemd/system/chronyd.service; enabled; vendor
  preset: enabled)
    Active: active (running) since Thu 2019-02-07 01:48:26 EST; 5min ago
      ...output omitted...
```

7. Отключите службу **chronyd**, чтобы она не запускалась при загрузке системы, затем просмотрите состояние службы.

```
[student@servera ~]$ sudo systemctl disable chronyd
[sudo] password for student: student
Removed /etc/systemd/system/multi-user.target.wants/chronyd.service.
[student@servera ~]$ systemctl status chronyd
● chronyd.service - NTP client/server
  Loaded: loaded (/usr/lib/systemd/system/chronyd.service; disabled; vendor
  preset: enabled)
    Active: active (running) since Thu 2019-02-07 01:48:26 EST; 5min ago
      ...output omitted...
```

8. Перезагрузите сервер **servera**, затем просмотрите статус службы **chronyd**.

```
[student@servera ~]$ sudo reboot
[sudo] password for student: student
Connection to servera closed by remote host.
Connection to servera closed.
[student@workstation ~]$
```

Войдите в систему как пользователь **student** на сервере **servera** и просмотрите состояние сервера **chronyd**.

```
[student@workstation ~]$ ssh student@servera
[student@servera ~]$ systemctl status chronyd
● chronyd.service - NTP client/server
  Loaded: loaded (/usr/lib/systemd/system/chronyd.service; disabled; vendor
  preset: enabled)
    Active: inactive (dead)
  Docs: man:chronyd(8)
        man:chrony.conf(5)
```

9. Выход из сервера **servera**.

```
[student@servera ~]$ exit
logout
Connection to servera closed.
[student@workstation]$
```

Завершение

На рабочей станции (**workstation**) запустите сценарий **lab services-control finish**, чтобы завершить это упражнение.

```
[student@workstation ~]$ lab services-control finish
```

На этом пошаговое упражнение завершено.

ЛАБОРАТОРНАЯ РАБОТА

УПРАВЛЕНИЕ СЛУЖБАМИ И ДЕМОНАМИ

КОНТРОЛЬНЫЙ СПИСОК РАБОТЫ

В этой лабораторной работе вы настроите несколько служб, которые будут включены или отключены, запущены или остановлены, в зависимости от предоставленной вам спецификации.

В РЕЗУЛЬТАТЕ

У вас должна быть возможность включать, отключать, запускать и останавливать службы.

ПРЕЖДЕ ЧЕМ ВЫ НАЧНЕТЕ

Войдите на рабочую станцию, как пользователь **student**, используя в качестве пароля слово **student**.

С рабочей станции (**workstation**) выполните команду **lab services-review start**. Команда запускает сценарий, который определяет, доступен ли хост **serverb** в сети. Сценарий также обеспечивает правильную настройку служб **psacct** и **rsyslog** на **serverb**.

```
[student@workstation ~]$ lab services-review start
```

1. На **serverb** запустите сервис **psacct**.
2. Настройте службу **psacct** для запуска при загрузке системы.
3. Остановите службу **rsyslog**.
4. Настройте службу **rsyslog** так, чтобы она не запускалась при загрузке системы.
5. Перезагрузите **serverb** перед оценкой лабораторной работы.

Оценка

На рабочей станции **workstation** запустите сценарий **lab services-review grade**, чтобы подтвердить успех выполнения лабораторной работы.

```
[student@workstation ~]$ lab services-review grade
```

Окончание

На рабочей станции **workstation** запустите сценарий **lab services-review finish**, чтобы завершить лабораторную работу.

```
[student@workstation ~]$ lab services-review finish
```

На этом лабораторная работа завершена.

РЕШЕНИЕ

УПРАВЛЕНИЕ СЛУЖБАМИ И ДЕМОНАМИ

КОНТРОЛЬНЫЙ СПИСОК РАБОТЫ

В этой лабораторной работе вы настроите несколько служб, которые будут включены или отключены, запущены или остановлены, в зависимости от предоставленной вам спецификации.

В РЕЗУЛЬТАТЕ

У вас должна быть возможность включать, отключать, запускать и останавливать службы.

ПРЕЖДЕ ЧЕМ ВЫ НАЧНЕТЕ

Войдите на рабочую станцию, как пользователь **student**, используя в качестве пароля слово **student**.

С рабочей станции (**workstation**) выполните команду **lab services-review start**. Команда запускает сценарий, который определяет, доступен ли хост **serverb** в сети. Сценарий также обеспечивает правильную настройку служб **psacct** и **rsyslog** на **serverb**.

```
[student@workstation ~]$ lab services-review start
```

1. На **serverb** запустите сервис **psacct**.

1.1. Используйте команду **ssh** для входа на **serverb** как пользователь **student**.

```
[student@workstation ~]$ ssh student@serverb  
[student@serverb ~]$
```

1.2. Используйте команду **systemctl**, чтобы проверить состояние службы **psacct**. Обратите внимание, что **psacct** остановлен и отключен для запуска во время загрузки.

```
[student@serverb ~]$ systemctl status psacct  
● psacct.service - Kernel process accounting  
  Loaded: loaded (/usr/lib/systemd/system/psacct.service; disabled; vendor  
  preset: disabled)  
    Active: inactive (dead)
```

1.3. Запустите службу **psacct**.

```
[student@serverb ~]$ sudo systemctl start psacct  
[sudo] password for student: student  
[student@serverb ~]$
```

1.4. Убедитесь, что служба **psacct** запущена.

```
[student@serverb ~]$ systemctl is-active psacct  
active
```

2. Настройте службу **psacct** для запуска при загрузке системы.

2.1. Включите запуск службы **psacct** при загрузке системы.

```
[student@serverb ~]$ sudo systemctl enable psacct  
Created symlink /etc/systemd/system/multi-user.target.wants/psacct.service → /usr/  
lib/systemd/system/psacct.service.
```

2.2. Убедитесь, что служба **psacct** может запускаться при загрузке системы.

```
[student@serverb ~]$ systemctl is-enabled psacct  
enabled
```

3. Остановите службу **rsyslog**.

3.1. Используйте команду **systemctl**, чтобы проверить состояние службы **rsyslog**. Обратите внимание, что **rsyslog** запущен и разрешен для запуска во время загрузки.

```
[student@serverb ~]$ systemctl status rsyslog  
● rsyslog.service - System Logging Service  
  Loaded: loaded (/usr/lib/systemd/system/rsyslog.service; enabled; vendor  
  preset: enabled)  
  Active: active (running) since Fri 2019-02-08 10:16:00 IST; 2h 34min ago  
    ...output omitted...
```

3.2. Остановите службу **rsyslog**.

```
[student@serverb ~]$ sudo systemctl stop rsyslog  
[sudo] password for student: student  
[student@serverb ~]$
```

3.3. Убедитесь, что служба **rsyslog** остановлена.

```
[student@serverb ~]$ systemctl is-active rsyslog  
Inactive
```

4. Настройте службу **rsyslog** так, чтобы она не запускалась при загрузке системы.

4.1. Отключите службу **rsyslog**, чтобы она не запускалась при загрузке системы.

```
[student@serverb ~]$ sudo systemctl disable rsyslog
[sudo] password for student:
Removed /etc/systemd/system/syslog.service.
Removed /etc/systemd/system/multi-user.target.wants/rsyslog.service.
```

4.2. Убедитесь, что **rsyslog** отключен для запуска при загрузке системы.

```
[student@serverb ~]$ systemctl is-enabled rsyslog
disabled
```

5. Перезагрузите **serverb** перед оценкой лабораторной работы.

```
[student@serverb ~]$ sudo reboot
[sudo] password for student:
Connection to serverb closed by remote host.
Connection to serverb closed.
[student@workstation ~]$
```

Оценка

На рабочей станции **workstation** запустите сценарий **lab services-review grade**, чтобы подтвердить успех выполнения лабораторной работы.

```
[student@workstation ~]$ lab services-review grade
```

Окончание

На рабочей станции **workstation** запустите сценарий **lab services-review finish**, чтобы завершить лабораторную работу.

```
[student@workstation ~]$ lab services-review finish
```

На этом лабораторная работа завершена.

РЕЗЮМЕ

В этой главе вы узнали:

- **systemd** предоставляет метод активации системных ресурсов, серверных демонов и других процессов как во время загрузки, так и в работающей системе.
- Используйте **systemctl** для запуска, остановки, перезагрузки, включения и отключения служб.
- Используйте команду **systemctl status**, чтобы определить состояние системных демонов и сетевых служб, запущенных **systemd**.
- Команда **systemctl list-dependencies** перечисляет все служебные юниты, от которых зависит конкретная служба.
- **systemd** может маскировать служебный **юнит**, чтобы он не запускался даже для удовлетворения зависимостей.

ГЛАВА 10

НАСТРОЙКА И ЗАЩИТА SSH

ЦЕЛЬ

Настройте безопасную службу командной строки в удаленных системах с помощью OpenSSH.

ЗАДАЧИ

- Войдите в удаленную систему и выполните команды с помощью **ssh**.
- Настройте аутентификацию на основе ключей для учетной записи пользователя, чтобы безопасно входить в удаленные системы без пароля.
- Ограничьте прямой вход в систему с правами **root** и отключите аутентификацию на основе пароля для службы **OpenSSH**.

РАЗДЕЛЫ

- Доступ к командной строке с помощью **SSH** (и упражнения с пошаговыми инструкциями)
- Настройка аутентификации на основе ключей **SSH** (и упражнения с пошаговыми инструкциями)
- Настройка конфигурации службы **OpenSSH** (и упражнения с пошаговыми инструкциями)

ЛАБОРОТОРНАЯ РАБОТА

Настройка и защита SSH

ДОСТУП К КОМАНДНОЙ СТРОКЕ УДАЛЁННОГО КОМПЬЮТЕРА С ПОМОЩЬЮ SSH

ЗАДАЧИ

После завершения этого раздела вы сможете войти в удаленную систему и запускать команды с помощью **ssh**.

ЧТО ТАКОЕ OPENSSH?

OpenSSH реализует протокол **SecureShell** или **SSH** в системах **Red Hat Enterprise Linux**. Протокол **SSH** позволяет системам обмениваться данными в зашифрованном и безопасном режиме по незащищенной сети.

Вы можете использовать команду **ssh** для создания безопасного соединения с удаленной системой, аутентификации как определенного пользователя и получения интерактивного сеанса оболочки в удаленной системе от имени этого пользователя. Вы также можете использовать команду **ssh** для запуска отдельной команды в удаленной системе без запуска интерактивной оболочки.

ПРИМЕРЫ БЕЗОПАСНОЙ ОБОЛОЧКИ (SHELL)

Следующая команда **ssh** выполнит вход на удаленный сервер **remotehost**, используя то же имя пользователя, что и текущий локальный пользователь. В этом примере удаленная система предлагает вам пройти аутентификацию с помощью пароля этого пользователя.

```
[user01@host ~]$ ssh remotehost  
user01@remotehost's password: redhat  
...output omitted...  
[user01@remotehost ~]$
```

Вы можете выполнить команду **exit** для выхода с удаленной системы.

```
[user01@remotehost ~]$ exit  
logout  
Connection to remotehost closed.  
[user01@host ~]$
```

Следующая команда **ssh** войдет в систему на удаленном сервере **remotehost**, используя имя пользователя **user02**. И снова удаленная система предложит вам пройти аутентификацию с использованием пароля данного пользователя.

```
[user01@host ~]$ ssh user02@remotehost
```

```
user02@remotehost's password: shadowman
...output omitted...
[user02@remotehost ~]$
```

Команда **ssh** выполнит команду **hostname** в удаленной системе **remotehost** от имени пользователя **user02** без доступа к удаленной интерактивной оболочке.

```
[user01@host ~]$ ssh user02@remotehost hostname
user02@remotehost's password: shadowman
remotehost.lab.example.com
[user01@host ~]$
```

Обратите внимание, что предыдущая команда отображала вывод в терминале локальной системы.

ИДЕНТИФИКАЦИЯ УДАЛЕННЫХ ПОЛЬЗОВАТЕЛЕЙ

Команда **w** отображает список пользователей, которые в данный момент вошли в систему. Это особенно полезно, когда необходимо выяснить, какие пользователи вошли в систему с помощью **ssh** из каких удаленных мест и что они делают.

```
[user01@host ~]$ ssh user01@remotehost
user01@remotehost's password: redhat
[user01@remotehost ~]$ w
16:13:38 up 36 min, 1 user, load average: 0.00, 0.00, 0.00
USER   TTY     FROM             LOGIN@   IDLE   JCPU   PCPU WHAT
user02  pts/0   172.25.250.10  16:13    7:30   0.01s  0.01s -bash
user01  pts/1   172.25.250.10  16:24    3.00s  0.01s  0.00s w
[user02@remotehost ~]$
```

Предыдущие данные показывают, что пользователь **user02** вошел в систему на псевдотерминале 0 (pts/0) в **16:13** сегодня с хоста с IP-адресом 172.25.250.10 и бездействовал в приглашении оболочки в течение семи минут и тридцати секунд. Предыдущие данные также показывают, что пользователь **user01** вошел в систему на псевдотерминале 1 (pts/1) и бездействует с последних трех секунд после выполнения команды **w**.

КЛЮЧИ SSH HOST

SSH защищает связь с помощью шифрования с открытым ключом. Когда **SSH-клиент** подключается к **SSH-серверу**, сервер отправляет копию своего открытого ключа клиенту до того, как клиент входит в систему. Это используется для настройки безопасного шифрования для канала связи и для аутентификации сервера для клиента.

Когда пользователь использует команду `ssh` для подключения к серверу SSH, команда проверяет, есть ли у него копия открытого ключа для этого сервера в его локальных файлах известных хостов. Системный администратор может предварительно настроить его в `/etc/ssh/ssh_known_hosts`, или у пользователя может быть файл `~/.ssh/known_hosts` в своем домашнем каталоге, который содержит ключ.

Если у клиента есть копия ключа, `ssh` сравнивает ключ из файлов известных хостов для этого сервера с тем, который он получил. Если ключи не совпадают, клиент предполагает, что сетевой трафик к серверу может быть перехвачен или что сервер был скомпрометирован, и запрашивает у пользователя подтверждение того, продолжать ли соединение.



ПРИМЕЧАНИЕ

Установите для параметра `StrictHostKeyChecking` значение `yes` в пользовательском файле `~/.ssh/config` или в общем файле настроек `/etc/ssh/ssh_config`, чтобы команда `ssh` всегда прерывала соединение SSH, если открытые ключи не совпадают.

Если у клиента нет копии открытого ключа в файлах известных хостов, команда `ssh` спросит вас, хотите ли вы войти в систему в любом случае. Если вы это сделаете, копия открытого ключа будет сохранена в вашем файле `~/.ssh/known_hosts`, так что идентификация сервера может быть автоматически подтверждена в будущем.

```
[user01@host ~]$ ssh newhost
The authenticity of host 'remotehost (172.25.250.12)' can't be established.
ECDSA key fingerprint is SHA256:qaS0PToLrq1C02XGk1A0iY7CaP7aPKimerDoaUkv720.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'newhost,172.25.250.12' (ECDSA) to the list of known
hosts.
user01@newhost's password: redhat
...output omitted...
[user01@newhost ~]$
```

Управление ключами известных хостов SSH

Если открытый ключ сервера был изменен из-за того, что ключ был утерян из-за сбоя жесткого диска или был заменен по какой-либо законной причине, вам нужно будет отредактировать файлы известных хостов, чтобы убедиться, что запись для старого открытого ключа заменена записью с новым открытый ключом для безопасного входа в систему.

Открытые ключи хранятся в `/etc/ssh/ssh_known_hosts` и `~/.ssh/known_hosts` каждого пользователя на клиенте SSH. Каждый ключ находится в одной строке. Первое поле - список имен хостов и IP-адресов, которые используют этот открытый ключ. Второе поле - алгоритм шифрования ключа. Последнее поле - сам ключ.

```
[user01@host ~]$ cat ~/.ssh/known_hosts
remotehost,172.25.250.11 ecdsa-sha2-nistp256
```

AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAIBmlzdHAyNTYAAABBOsEi0e+FlaNT6jul8Ag
+RViZl0yE2w6iYUr+1fPtOIF0EaOgFZ1LXM37VFTxdgFxHS3D5WhnIfb+68zf8+w=

Каждый удаленный **сервер SSH**,
к которому вы подключаетесь,
хранит свой открытый ключ в каталоге **/etc/ssh** в файлах с расширением **.pub**.

```
[user01@remotehost ~]$ ls /etc/ssh/*key.pub
/etc/ssh/ssh_host_ecdsa_key.pub /etc/ssh/ssh_host_ed25519_key.pub /etc/ssh/
ssh_host_rsa_key.pub
```



ПРИМЕЧАНИЕ

Рекомендуется добавлять записи, соответствующие файлам **ssh_host_* key.pub** сервера, в ваш файл **~/.ssh/known_hosts** или в общесистемный файл **/etc/ssh/ssh_known_hosts**.



РЕКОМЕНДАЦИИ

Справочные страницы **man ssh(1)**, **w(1)**, и **hostname(1)**

Дополнительные сведения см. В главе «Использование безопасного обмена данными между двумя системами с помощью OpenSSH» в Руководстве по настройке и управлению безопасностью Red Hat Enterprise Linux 8.0 BETA по адресу

https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/8-beta/html/configuring_and_managing_security/assembly_using-securecommunications-with-openssh-configuring-and-managing-security

УПРАЖНЕНИЯ С ПОШАГОВЫМИ ИНСТРУКЦИЯМИ

ДОСТУП К УДАЛЕННОЙ КОМАНДНОЙ СТРОКЕ

В этом упражнении вы войдете в удаленную систему как разные пользователи и будете выполнять команды.

РЕЗУЛЬТАТЫ

Вы должны быть способны:

- Войдите в удаленную систему.
- Выполнить команды с помощью защищенной оболочки **OpenSSH**.

ПРЕЖДЕ ЧЕМ ВЫ НАЧНЕТЕ

Войдите на рабочую станцию, как пользователь **student**, используя в качестве пароля слово **student**.

На рабочей станции (workstation) выполните **lab ssh-access start**, чтобы начать упражнение. Этот сценарий обеспечивает правильную настройку среды.

```
[student@workstation ~]$ labssh-accessstart
```

1. С рабочей станции (**workstation**) откройте сеанс **SSH** на сервер **servera** как пользователь **student**.

```
[student@workstation ~]$ ssh student@servera  
...output omitted...  
[student@servera ~]$
```

2. Откройте сеанс **SSH** на **serverb**, как пользователь **student**. Получите ключ хоста. Используйте слово **student** в качестве пароля, когда вам будет предложено ввести пароль пользователя **student** на **serverb**.

```
[student@servera ~]$ ssh student@serverb  
The authenticity of host 'serverb (172.25.250.11)' can't be established.  
ECDSA key fingerprint is SHA256:ERTdjooOIrIwVSZQnqD5or+JbXfidg0udb3DXBuHWzA.  
Are you sure you want to continue connecting (yes/no)? yes  
Warning: Permanently added 'serverb,172.25.250.11' (ECDSA) to the list of known  
hosts.  
student@serverb's password: student  
...output omitted...  
[student@serverb ~]$
```

Ключ хоста записывается в файле **/home/student/.ssh/known_hosts** на сервере **servera**, чтобы идентифицировать **serverb**, потому что пользователь **student** инициировал SSH-соединение с **servera**. Если файл **/home/student/.ssh/known_hosts** еще не существует, он появляется как новый файл вместе с новой записью в нем. Команда **ssh** не может выполняться должным образом, если удаленный хост имеет другой ключ, чем записанный.

- Запустите команду **w**, чтобы отобразить пользователей, которые в настоящее время вошли в систему на **serverb**.

```
[student@serverb ~]$ w
18:49:29 up 2:55, 1 user,  load average: 0.00, 0.00, 0.00
USER   TTY      FROM          LOGIN@    IDLE   JCPU   PCPU WHAT
student pts/0    172.25.250.10    18:33    0.00s  0.01s  0.00s w
```

Предыдущие выходные данные показывают, что пользователь **student** вошел в систему с хоста с IP-адресом 172.25.250.10, который является сервером **servera** в сети класса.



ПРИМЕЧАНИЕ

IP-адрес системы идентифицирует систему в сети. Вы узнаете об IP-адресах в следующей главе.

- Выходите из оболочки пользователя **student** на **serverb**.

```
[student@serverb ~]$ exit
logout
Connection to serverb closed.
[student@servera ~]$
```

- Откройте сеанс SSH для **serverb** от имени пользователя **root**.

```
[student@servera ~]$ ssh root@serverb
root@serverb's password: redhat
...output omitted...
[root@serverb ~]#
```

Обратите внимание, что предыдущая команда **ssh** не просила вас принять ключ хоста, потому что он был найден среди известных хостов. Если идентификатор **serverb** изменится в любое время, **OpenSSH** предложит вам подтвердить и принять новый ключ хоста.

- Запустите команду **w**, чтобы отобразить пользователей, которые в данный момент вошли на **serverb**.

```
[root@serverb ~]# w
19:10:28 up 3:16, 1 user, load average: 0.00, 0.00, 0.00
USER     TTY      FROM          LOGIN@    IDLE   JCPU   PCPU WHAT
root     pts/0    172.25.250.10  19:09    1.00s  0.01s  0.00s w
```

Предыдущие выходные данные показывают, что пользователь **root** вошел в систему с хоста с IP-адресом 172.25.250.10, который является сервером **serverb** в сети класса.

7. Закройте оболочку пользователя **root** на **serverb**.

```
[root@serverb ~]# exit
logout
Connection to serverb closed.
[student@servera ~]$
```

8. Удалите файл **/home/student/.ssh/known_hosts** с **servera**. Это приводит к тому, что **ssh** теряет записанные идентификаторы удаленных систем.

```
[student@servera ~]$ rm /home/student/.ssh/known_hosts
```

Ключи хоста могут измениться по законным причинам: возможно, удаленная машина была заменена из-за аппаратного сбоя, или, возможно, удаленная машина была переустановлена. Обычно рекомендуется удалить только ключевую запись для конкретного хоста в файле **known_hosts**. Поскольку этот конкретный файл **known_hosts** имеет только одну запись, вы можете удалить весь файл.

9. Откройте сеанс **SSH** на **serverb** как пользователь **student**. Примите ключ хоста, если его попросят. Используйте слово **student** в качестве пароля, когда вам будет предложено ввести пароль пользователя **student** на **serverb**.

```
[student@servera ~]$ ssh student@serverb
The authenticity of host 'serverb (172.25.250.11)' can't be established.
ECDSA key fingerprint is SHA256:ERTdjoo0IrIwVSZQnqD5or+JbXfidg0udb3DXBuHWzA.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'serverb,172.25.250.11' (ECDSA) to the list of known
hosts.
student@serverb's password: student
...output omitted...
[student@serverb ~]$
```

Обратите внимание, что команда **ssh** запросила ваше подтверждение принятия или отклонения ключа хоста, потому что не смогла найти ключ для удаленного хоста.

10. Выйдите из оболочки пользователя **student** на **serverb** и убедитесь, что на сервере существует новый экземпляр **known_hosts**.

```
[student@serverb ~]$ exit  
logout  
Connection to serverb closed.  
[student@servera ~]$ ls -l /home/student/.ssh/known_hosts  
-rw-r--r--. 1 student student 183 Feb 1 20:26 /home/student/.ssh/known_hosts
```

11. Убедитесь, что новый экземпляр файла **known_hosts** имеет ключ хоста **serverb**.

```
[student@servera ~]$ cat /home/student/.ssh/known_hosts  
serverb,172.25.250.11 ecdsa-sha2-nistp256  
AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAlbmlzdHAyNTYAAABBI9LEYEhwmU1rN  
qnbBPukH2Ba0/QBAu9WbS4m03B3MIhhXWKFFNa/UINjY8NDpEM+hkJe/GmnkcEYMLbCfd  
9nMA=
```

Фактический результат может отличаться.

12. Запустите **hostname** удаленно на **serverb** без доступа к интерактивной оболочке.

```
[student@servera ~]$ ssh student@serverb hostname  
student@serverb's password: student  
serverb.lab.example.com
```

Предыдущая команда отображала полное имя хоста удаленного системного **serverb**.

13. Выйдите из оболочки пользователя **student** на **servera**.

```
[student@servera ~]$ exit  
logout  
Connection to servera closed.
```

Завершение

На рабочей станции (workstation) выполните скрипт **lab ssh-access finish**, чтобы завершить это упражнение.

```
[student@workstation ~]$ lab ssh-access finish
```

На этом пошаговое упражнение завершено.

НАСТРОЙКА АУТЕНТИФИКАЦИИ НА ОСНОВЕ SSH

ЗАДАЧИ

После завершения этого раздела вы сможете настроить учетную запись пользователя для использования аутентификации на основе ключей для безопасного входа в удаленные системы без пароля.

АУТЕНТИФИКАЦИЯ НА ОСНОВЕ SSH-КЛЮЧЕЙ

Вы можете настроить SSH-сервер, так чтобы вы могли проходить аутентификацию без пароля, используя аутентификацию на основе ключей. Данная возможность основана на схеме закрытого и открытого ключа.

Для этого вы создаете соответствующую пару файлов криптографических ключей. Один из них - закрытый ключ, другой - соответствующий открытый ключ. Файл закрытого ключа используется в качестве учетных данных для аутентификации и, как и пароль, должен храниться в секрете. Открытый ключ копируется в системы, к которым пользователь хочет подключиться, и используется для проверки закрытого ключа. Открытый ключ не обязательно должен быть секретным.

Вы помещаете копию открытого ключа в свою учетную запись на сервере. Когда вы пытаетесь войти в систему, SSH-сервер может использовать открытый ключ для выдачи запроса, на который можно правильно ответить только с помощью закрытого ключа. В результате ваш ssh-клиент может автоматически аутентифицировать ваш логин на сервере с вашей уникальной копией закрытого ключа. Это позволяет вам получить безопасный доступ к системам, не требуя ввода пароля каждый раз в интерактивном режиме.

Генерация ключей SSH

Чтобы создать закрытый ключ и соответствующий открытый ключ для аутентификации, используйте команду **ssh-keygen**. По умолчанию ваши закрытый и открытый ключи сохраняются в файлах `~/.ssh/id_rsa` и `~/.ssh/id_rsa.pub` соответственно.

Если вы не укажете кодовую фразу при появлении запроса **ssh-keygen**, сгенерированный закрытый ключ не будет защищен. В этом случае любой, у кого есть ваш файл закрытого ключа, может использовать его для аутентификации. Если вы установите парольную фразу, вам нужно будет ввести эту парольную фразу при использовании закрытого ключа для аутентификации. (Следовательно, для аутентификации вы будете использовать парольную фразу закрытого ключа, а не свой пароль на удаленном хосте.)

Вы можете запустить вспомогательную программу **ssh-agent**, которая может временно кэшировать кодовую фразу вашего закрытого ключа в памяти в начале сеанса, чтобы получить настоящую аутентификацию без пароля. Это будет обсуждаться позже в этом разделе.

```
[user@host ~]$ ssh-keygen
```

```
Generating public/private rsa key pair.
Enter file in which to save the key (/home/user/.ssh/id_rsa): Enter
Created directory '/home/user/.ssh'.
Enter passphrase (empty for no passphrase): Enter
Enter same passphrase again: Enter
Your identification has been saved in /home/user/.ssh/id_rsa.
Your public key has been saved in /home/user/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:vxutUNPio3QDCyvkYm1oIx35hmMrHpPKWFdIYu3HV+w user@host.lab.example.com
The key's randomart image is:
+---[RSA 2048]---+
| . . .
| o o o
| . = o o .
| o + = S E .
| ..o o + * +
| .% o . + B .
|= *o o . + *
|++.
|++.
+---[SHA256]---+
```

В следующем примере команды **ssh-keygen** показано создание закрытого ключа, защищенного парольной фразой, вместе с открытым ключом.

```
[user@host ~]$ ssh-keygen -f .ssh/key-with-pass
Generating public/private rsa key pair.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in .ssh/key-with-pass.
Your public key has been saved in .ssh/key-with-pass.pub.
The key fingerprint is:
SHA256:w3GGB7EyHUr4aOcNPKmhNKS7dl1YsMVLvFZJ77VxAo user@host.lab.example.com
The key's randomart image is:
+---[RSA 2048]---+
| . + = o ...
| = B XEo o.
| . o o X =.....
| = = = B = o.
|= + * * S .
|.+= o + .
| + .
|
|
+---[SHA256]---+
```

Параметр **-f** команды **ssh-keygen** определяет файлы, в которых сохраняются ключи. В предыдущем примере закрытый открытый ключ сохранен в файлах **/home/user/.ssh/key-with-pass** и **/home/user/.ssh/key-with-pass.pub** соответственно.



ПРЕДУПРЕЖДЕНИЕ

Во время дальнейшего создания пары ключей **SSH**, если вы не укажете уникальное имя файла, вам будет предложено разрешение на перезапись существующих файлов **id_rsa** и **id_rsa.pub**. Если вы перезаписываете существующие файлы **id_rsa** и **id_rsa.pub**, вы должны заменить старый открытый ключ новым на всех серверах **SSH**, на которых есть ваш старый открытый ключ.

После создания ключей **SSH** они по умолчанию сохраняются в каталоге **.ssh/** домашнего каталога пользователя. Режимы разрешений должны быть **600** для закрытого ключа и **644** для открытого ключа.

Совместное использование открытого ключа

Прежде чем можно будет использовать аутентификацию на основе ключей, открытый ключ необходимо скопировать в целевую систему. Команда **ssh-copy-id** копирует открытый ключ пары ключей **SSH** в целевую систему. Если вы опустите путь к файлу открытого ключа при запуске **ssh-copy-id**, он будет использовать файл по умолчанию **/home/user/.ssh/id_rsa.pub**.

```
[user@host ~]$ ssh-copy-id -i .ssh/key-with-pass.pub user@remotehost
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/user/.ssh/
id_rsa.pub"
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter
out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted
now it is to install the new keys
user@remotehost's password: redhat
Number of key(s) added: 1

Now try logging into the machine, with: "ssh 'user@remotehost'"
and check to make sure that only the key(s) you wanted were added.
```

После успешной передачи открытого ключа в удаленную систему вы можете пройти аутентификацию в удаленной системе с помощью соответствующего закрытого ключа при входе в удаленную систему через **SSH**. Если вы не укажете путь к файлу закрытого ключа при запуске команды **ssh**, будет использован файл по умолчанию **/home/user/.ssh/id_rsa**.

```
[user@host ~]$ ssh -i .ssh/key-with-pass user@remotehost
Enter passphrase for key '.ssh/key-with-pass': redhatpass
...output omitted...
```

```
[user@remotehost ~]$ exit  
logout  
Connection to remotehost closed.  
[user@host ~]$
```

Использование ssh-agent для неинтерактивной аутентификации

Если ваш закрытый ключ **SSH** защищен парольной фразой, вам обычно необходимо её ввести, чтобы использовать закрытый ключ для аутентификации. Однако вы можете использовать программу **ssh-agent** для временного кэширования ключевой фразы в памяти. Затем каждый раз, когда вы используете **SSH** для входа в другую систему с закрытым ключом, **ssh-agent** автоматически предоставит вам парольную фразу. Это удобно и может повысить безопасность, поскольку у кого-то, «занимающегося просмотром через плечо»^{"shoulder surfing"}, меньше возможностей увидеть, как вы вводите кодовую фразу.

В зависимости от конфигурации вашей локальной системы, если вы изначально входите в графическую среду рабочего стола **GNOOME**, программа **ssh-agent** может быть автоматически запущена и настроена для вас.

Если вы входите в систему с помощью текстовой консоли, входите в систему с помощью **ssh** или **sudo** или **su**, вам, вероятно, потребуется вручную запустить **ssh-agent** для этого сеанса. Вы можете сделать это с помощью следующей команды:

```
[user@host ~]$ eval $(ssh-agent)  
Agent pid 10155  
[user@host ~]$
```



ПРИМЕЧАНИЕ

Когда вы запускаете **ssh-agent**, он выводит некоторые команды оболочки. Вам необходимо запустить эти команды, чтобы установить переменные среды, используемые такими программами, как **ssh-add**, для связи с ним. Команда **eval \$(ssh-agent)** запускает **ssh-agent** и выполняет эти команды для автоматической установки этих переменных среды для этого сеанса оболочки. Он также отображает **PID** процесса **ssh-agent**.

После запуска **ssh-agent** вам необходимо сообщить ему кодовую фразу для вашего закрытого ключа или ключей. Вы можете сделать это с помощью команды **ssh-add**.

Следующие команды **ssh-add** добавляют закрытые ключи из файлов **/home/user/.ssh/id_rsa** (по умолчанию) и **/home/user/.ssh/key-with-pass** соответственно.

```
[user@host ~]$ ssh-add  
Identity added: /home/user/.ssh/id_rsa (user@host.lab.example.com)  
[user@host ~]$ ssh-add .ssh/key-with-pass  
Enter passphrase for .ssh/key-with-pass: redhatpass
```

```
Identity added: .ssh/key-with-pass (user@host.lab.example.com)
```

После успешного добавления закрытых ключей в процесс **ssh-agent** вы можете вызвать SSH-соединение с помощью команды **ssh**. Если вы используете какой-либо файл закрытого ключа, отличный от файла по умолчанию **/home/user/.ssh/id_rsa**, вы должны использовать параметр **-i** с командой **ssh**, чтобы указать путь к файлу закрытого ключа.

В следующем примере команды **ssh** используется файл закрытого ключа по умолчанию для аутентификации на сервере SSH.

```
[user@host ~]$ ssh user@remotehost
Last login: Fri Apr 5 10:53:50 2019 from host.example.com
[user@remotehost ~]$
```

В следующем примере команды **ssh** используется файл закрытого ключа **/home/user/.ssh/key-with-pass** (не по умолчанию) для аутентификации на сервере SSH. Закрытый ключ в следующем примере уже был расшифрован и добавлен в его родительский процесс **ssh-agent**, поэтому команда **ssh** не предлагает вам расшифровать закрытый ключ путем интерактивного ввода его парольной фразы.

```
[user@host ~]$ ssh -i .ssh/key-with-pass user@remotehost
Last login: Mon Apr 8 09:44:20 2019 from host.example.com
[user@remotehost ~]$
```

Когда вы выйдете из сеанса, в котором запущен **ssh-agent**, процесс завершится, и ваши парольные фразы для ваших закрытых ключей будут удалены из памяти.



РЕКОМЕНДАЦИИ

Справочные страницы **man ssh-keygen(1)**, **ssh-copy-id(1)**, **ssh-agent(1)**, **ssh-add(1)**

УПРАЖНЕНИЯ С ПОШАГОВЫМИ ИНСТРУКЦИЯМИ

НАСТРОЙКА АУТЕНТИФИКАЦИИ НА ОСНОВЕ SSH

В этом упражнении вы настроите пользователя для использования аутентификации на основе ключей для SSH.

В РЕЗУЛЬТАТЕ

Вы должны быть способны:

- Сгенерируйте пару ключей SSH без защиты парольной фразой.
- Создайте пару ключей SSH с защитой парольной фразы.
- Аутентификация с использованием ключей SSH без парольной фразы и защищенных парольной фразой.

ПРЕЖДЕ ЧЕМ ВЫ НАЧНЕТЕ

Войдите на рабочую станцию как пользователь **student**, используя в качестве пароля слово **student**.

На рабочей станции запустите сценарий **labssh-configurestart**, чтобы начать упражнение. Этот скрипт создает необходимые учетные записи пользователей.

```
[student@workstation ~]$ labssh-configurestart
```

1. С рабочей станции **workstation**, откройте сеанс SSH на **serverb** как пользователь **student**.

```
[student@workstation ~]$ ssh student@serverb  
...output omitted...  
[student@serverb ~]$
```

2. Используйте команду **su**, чтобы переключиться на пользователя **operator1** на **serverb**.

```
[student@serverb ~]$ su - operator1  
Password: redhat  
[operator1@serverb ~]$
```

3. Используйте команду **ssh-keygen** для генерации ключей SSH. Не вводите кодовую фразу.

```
[operator1@serverb ~]$ ssh-keygen  
Generating public/private rsa key pair.
```

```
Enter file in which to save the key (/home/operator1/.ssh/id_rsa): Enter  
Created directory '/home/operator1/.ssh'.
```

```
Enter passphrase (empty for no passphrase): Enter
```

```
Enter same passphrase again: Enter
```

```
Your identification has been saved in /home/operator1/.ssh/id_rsa.
```

```
Your public key has been saved in /home/operator1/.ssh/id_rsa.pub.
```

```
The key fingerprint is:
```

```
SHA256:JainiQdnRosC+xXhOqsJQQLzBNULdb+jByrCZQBERI
```

```
operator1@serverb.lab.example.com
```

```
The key's randomart image is:
```

```
+---[RSA 2048]---+  
|E+*+ooo . . . |  
| . = o . o . . |  
|o.. = . . o . . |  
|+. + * . o . . |  
|+ = X . S + . . |  
| + @ + = . . . |  
| . + = o . . . |  
| .o . . . . . . |  
|o . . . . . . . |  
+---[SHA256]---+
```

4. Используйте команду **ssh-copy-id** для отправки открытого ключа пары ключей **SSHoperator1** на **servera**.

```
[operator1@serverb ~]$ ssh-copy-id operator1@servera  
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/  
operator1/.ssh/id_rsa.pub"  
The authenticity of host 'servera (172.25.250.10)' can't be established.  
ECDSA key fingerprint is SHA256:ERTdjoo0IrIwVSZQnqD5or+JbXfidg0udb3DXBuHWzA.  
Are you sure you want to continue connecting (yes/no)? yes  
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter  
out any that are already installed  
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted  
now it is to install the new keys  
operator1@servera's password: redhat  
Number of key(s) added: 1  
  
Now try logging into the machine, with: "ssh 'operator1@servera'"  
and check to make sure that only the key(s) you wanted were added.
```

5. Выполните команду **hostname** на **servera** удаленно, используя **SSH**, без доступа к удаленной интерактивной оболочке.

```
[operator1@serverb ~]$ ssh operator1@servera hostname  
servera.lab.example.com
```

Обратите внимание, что предыдущая команда `ssh` не запрашивала пароль, потому что она использовала закрытый ключ без парольной фразы для экспортированного открытого ключа для аутентификации как **operator1** на **servera**. Этот подход небезопасен, потому что любой, у кого есть доступ к файлу закрытого ключа, может войти на **servera** как **operator1**. Безопасная альтернатива - защитить закрытый ключ парольной фразой, что является следующим шагом.

6. Используйте команду `ssh-keygen`, чтобы сгенерировать еще один набор ключей **SSH** с защитой парольной фразой. Сохраните ключ как `/home/operator1/.ssh/key2`.



ПРЕДУПРЕЖДЕНИЕ

Если вы не укажете файл, в котором будет сохранен ключ, будет использован файл по умолчанию (`/home/user/.ssh/id_rsa`). Вы уже использовали имя файла по умолчанию при генерации ключей **SSH** на предыдущем шаге, поэтому очень важно указать файл не по умолчанию, иначе существующие ключи **SSH** будут перезаписаны.

```
[operator1@serverb ~]$ ssh-keygen -f .ssh/key2
Generating public/private rsa key pair.
Enter passphrase (empty for no passphrase): redhatpass
Enter same passphrase again: redhatpass
Your identification has been saved in .ssh/key2.
Your public key has been saved in .ssh/key2.pub.
The key fingerprint is:
SHA256:OCtCjfPm5QrbPBgqbEIWCcw5AI4oSIMEbgLrBQ1HWKI
operator1@serverb.lab.example.com
The key's randomart image is:
+--- [RSA 2048] ---+
|0=X*           |
|OB=.          |
|E*o.          |
|Booo .        |
|..= . o S     |
|+.o  o        |
|+.oo+ o       |
|+o.o.+        |
|+ . =o.       |
+--- [SHA256] ---+
```

7. Используйте команду `ssh-copy-id`, чтобы отправить открытый ключ пары ключей, защищенной парольной фразой, **operator1** на **servera**.

```
[operator1@serverb ~]$ ssh-copy-id -i .ssh/key2.pub operator1@servera
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: ".ssh/key2.pub"
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter
out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted
now it is to install the new keys
```

```
Number of key(s) added: 1
```

Now try logging into the machine, with: "ssh 'operator1@servera'"
and check to make sure that only the key(s) you wanted were added.

Обратите внимание, что предыдущая команда **ssh-copy-id** не запрашивала пароль, поскольку использовала открытый ключ, который вы экспортировали в сервер на предыдущем шаге.

8. Выполните команду **hostname** на **servera** удаленно с помощью **SSH** без доступа к удаленной интерактивной оболочке. Используйте **/home/operator1/.ssh/key2** в качестве файла идентификации. Укажите **redhatpass** в качестве ключевой фразы, которую вы установили для закрытого ключа на предыдущем шаге.

```
[operator1@serverb ~]$ ssh -i .ssh/key2 operator1@servera hostname  
Enter passphrase for key '.ssh/key2': redhatpass  
servera.lab.example.com
```

Обратите внимание, что предыдущая команда **ssh** запросила у вас парольную фразу, которую вы использовали для защиты закрытого ключа пары ключей **SSH**. Эта кодовая фраза защищает закрытый ключ. Если злоумышленник получит доступ к закрытому ключу, он не сможет использовать его для доступа к другим системам, поскольку сам закрытый ключ защищен парольной фразой. Команда **ssh** использует парольную фразу, отличную от той, что используется для **operator1** на **servera**, и требует, чтобы пользователи знали и то, и другое. Вы можете использовать **ssh-agent**, как на следующем шаге, чтобы избежать интерактивного ввода парольной фразы при входе в систему с помощью **SSH**. Использование **ssh-agent** удобнее и безопаснее в ситуациях, когда администраторы регулярно входят в удаленные системы.

9. Запустите **ssh-agent** в оболочке **Bash** и добавьте защищенный парольной фразой закрытый ключ (**/home/operator1/.ssh/key2**) пары ключей **SSH** в сеанс оболочки.

```
[operator1@serverb ~]$ eval $(ssh-agent)  
Agent pid 21032  
[operator1@serverb ~]$ ssh-add .ssh/key2  
Enter passphrase for .ssh/key2: redhatpass  
Identity added: .ssh/key2 (operator1@serverb.lab.example.com)
```

Предыдущая команда **eval** запустила **ssh-agent** и настроила этот сеанс оболочки для его использования. Затем вы использовали **ssh-add** для предоставления разблокированного закрытого ключа **ssh-agent**.

10. Выполните команду **hostname** на **servera** удаленно, не обращаясь к удаленной интерактивной оболочке. Используйте **/home/operator1/.ssh/key2** в качестве файла идентификации.

```
[operator1@serverb ~]$ ssh -i .ssh/key2 operator1@servera hostname  
servera.lab.example.com
```

Обратите внимание, что предыдущая команда **ssh** не предлагала вам ввести парольную фразу в интерактивном режиме.

- 11.** Откройте другой терминал на рабочей станции**workstation** и откройте сеанс **SSH** на **serverb** как пользователь**student**.

```
[student@workstation ~]$ ssh student@serverb  
...output omitted...  
[student@serverb ~]$
```

- 12.** На **serverb** используйте команду **su**, чтобы переключиться на **operator1** и вызвать **SSH**-соединение с сервером**servera**. Используйте **/home/operator1/.ssh/key2** в качестве файла идентификации для аутентификации с использованием ключей **SSH**.

- 12.1.** Используйте команду **su**, чтобы переключиться на **operator1**.

```
[student@serverb ~]$ su - operator1  
Password: redhat  
[operator1@serverb ~]$
```

- 12.2.** Откройте сеанс **SSH** для сервера**servera** как **operator1**.

```
[operator1@serverb ~]$ ssh -i .ssh/key2 operator1@servera  
Enter passphrase for key '.ssh/key2': redhatpass  
...output omitted...  
[operator1@servera ~]$
```

Обратите внимание, что предыдущая команда **ssh** предложила вам ввести парольную фразу в интерактивном режиме, потому что вы не вызывали соединение **SSH** из оболочки, которую вы использовали для запуска **ssh-agent**.

- 13.** Закройте все используемые оболочки во втором терминале.

- 13.1.** Выйти из сервера.

```
[operator1@servera ~]$ exit  
logout  
Connection to servera closed.  
[operator1@serverb ~]$
```

- 13.2.** Закройте оболочки **operator1** и **student** на **serverb**, чтобы вернуться в оболочку пользователя **student** на рабочей станции.

- 13.3.** Закройте второй терминал на рабочей станции**workstation**.

```
[operator1@serverb ~]$ exit
```

```
logout  
[student@serverb ~]$ exit  
logout  
Connection to serverb closed.  
[student@workstation ~]$
```

14. Выдите из **serverb** на первом терминале и завершите это упражнение.

14.1. В первом терминале выдите из оболочки пользователя **operator1** на **серверb**

```
[operator1@serverb ~]$ exit  
Logout  
[student@serverb ~]$
```

Команда **exit** заставила вас выйти из оболочки пользователя **operator1**, завершив сеанс оболочки, в котором был активен **ssh-agent**, и вернуться в оболочку пользователя **student** на **serverb**.

14.2. Выдите из оболочки пользователя **student**на **серверb**, чтобы вернуться в оболочку пользователя **student**на рабочей станции**workstation**.

```
[student@serverb ~]$ exit  
logout  
Connection to serverb closed.  
[student@workstation ~]$
```

Завершение

На рабочей станции запустите скрипт **lab ssh-configure finish**, чтобы завершить это упражнение.

```
[student@workstation ~]$ lab ssh-configure finish
```

На этом пошаговое упражнение завершено.

НАСТРОЙКА КОНФИГУРАЦИИ СЕРВИСА OPENSSH

ЗАДАЧИ

После завершения этого раздела вы сможете ограничить прямой вход в систему с правами **root** и отключить аутентификацию на основе пароля для службы **OpenSSH**.

НАСТРОЙКА СЕРВЕРА OPENSSH

Сервис **OpenSSH** предоставляется демоном **sshd**. Его основной файл конфигурации - **/etc/ssh/sshd_config**.

Конфигурация сервера **OpenSSH** по умолчанию работает хорошо. Однако вы можете внести некоторые изменения, чтобы усилить безопасность вашей системы. Есть два общих изменения, которые вы, возможно, захотите внести. Возможно, вы захотите запретить прямой удаленный вход в учетную запись **root**, и вы можете запретить аутентификацию на основе пароля (в пользу аутентификации с закрытым ключом **SSH**).

ЗАПРЕТИТЕ СУПЕРПОЛЬЗОВАТЕЛЮ ВХОДИТЬ В СИСТЕМУ С ПОМОЩЬЮ SSH

Рекомендуется запретить прямой вход в учетную запись пользователя **root** из удаленных систем. Некоторые из рисков, связанных с прямым входом в систему с правами **root**, включают:

- Имя пользователя **root** по умолчанию существует в каждой системе Linux, поэтому потенциальному злоумышленнику нужно только угадать пароль, а не действительную комбинацию имени пользователя и пароля. Это снижает сложность для злоумышленника.
- Пользователь **root** имеет неограниченные привилегии, поэтому его компрометация может привести к максимальному ущербу для системы.
- С точки зрения аудита может быть сложно отследить, какой авторизованный пользователь вошел в систему как **root** и внес изменения. Если пользователям необходимо войти в систему как обычный пользователь и переключиться на учетную запись **root**, это создает событие в журнале, которое можно использовать для отслеживания использования учётной записи **root**.

Сервер **OpenSSH** использует параметр конфигурации **PermitRootLogin** в файле конфигурации **/etc/ssh/sshd_config**, чтобы разрешить или запретить пользователям входить в систему как **root**.

```
PermitRootLogin yes
```

Если для параметра **PermitRootLogin** установлено значение **yes**, как и по умолчанию, людям разрешается входить в систему как **root**. Чтобы этого не произошло, установите значение **no**. В качестве альтернативы, чтобы предотвратить аутентификацию на основе пароля, но разрешить аутентификацию на основе закрытого ключа для **root**, установите для параметра **PermitRootLogin** значение **without-password** (без пароля).

Чтобы изменения вступили в силу, необходимо **перезагрузить SSH-сервер (sshd)**.

```
[root@host ~]# systemctl reload sshd
```



ВАЖНО

Преимущество использования команды **systemctl reload sshd** заключается в том, что она сообщает **sshd** перечитать файл конфигурации, а не полностью перезапускать службу. Команда **systemctl restart sshd** также применит изменения, но также остановит и запустит службу, разорвав все активные SSH-соединения с этим хостом.

ЗАПРЕТ АУТЕНТИФИКАЦИИ ПО ПАРОЛЯМ ДЛЯ SSH

Разрешение входа в удаленную командную строку только на основе закрытого ключа имеет ряд преимуществ:

- Злоумышленники не могут использовать атаки по подбору пароля для удаленного взлома известных учетных записей в системе.
- При использовании закрытых ключей, защищенных парольной фразой, злоумышленнику требуется как кодовая фраза, так и копия закрытого ключа. С паролями злоумышленнику просто нужен пароль.
- При использовании закрытых ключей, защищенных парольной фразой, в сочетании с **ssh-agent**, парольная фраза раскрывается реже, поскольку она вводится реже, и вход в систему более удобен для пользователя.

Сервер **OpenSSH** использует параметр **PasswordAuthentication** в файле конфигурации **/etc/ssh/sshd_config**, чтобы контролировать, могут ли пользователи использовать аутентификацию на основе пароля для входа в систему.

```
PasswordAuthentication yes
```

Значение по умолчанию **yes** для параметра **PasswordAuthentication** в файле конфигурации **/etc/ssh/sshd_config** заставляет **SSH-сервер** разрешать пользователям использовать аутентификацию на основе пароля при входе в систему. Значение **no** для **PasswordAuthentication** запрещает пользователям использовать аутентификацию на основе пароля.

Помните, что всякий раз, когда вы изменяете файл **/etc/ssh/sshd_config**, вы должны перезагрузить службу **sshd**, чтобы изменения вступили в силу.



ВАЖНО

Помните, что, если вы отключите аутентификацию на основе пароля для **ssh**, у вас должен быть способ гарантировать, что файл **~/.ssh/authorized_keys** пользователя на удаленном сервере заполнен их открытым ключом, чтобы они могли войти в систему.



РЕКОМЕНДАЦИИ

Справочные страницы **manssh(1)**, **sshd_config(5)**.

УПРАЖНЕНИЯ С ПОШАГОВЫМИ ИНСТРУКЦИЯМИ

НАСТРОЙКА КОНФИГУРАЦИИ СЕРВИСА OPENSSH

В этом упражнении вы отключите прямой вход в систему для пользователя **root** и аутентификацию на основе пароля для службы **OpenSSH** на одном из ваших серверов.

В РЕЗУЛЬТАТЕ

Вы должны быть способны:

- Отключить прямой вход в систему как **root** через **ssh**.
- Отключить аутентификацию на основе пароля для удаленных пользователей для подключения через **SSH**.

ПРЕЖДЕ ЧЕМ ВЫ НАЧНЕТЕ

Войдите на рабочую станцию, как пользователь **student**, используя в качестве пароля слово **student**.

На рабочей станции **workstation** запустите сценарий **lab ssh-customize start**, чтобы начать упражнение. Этот сценарий создает необходимые учетные записи пользователей и файлы.

```
[student@workstation ~]$ lab ssh-customize start
```

1. С рабочей станции **workstation** откройте сеанс **SSH** на **serverb** как пользователь **student**.

```
[student@workstation ~]$ ssh student@serverb  
...output omitted...  
[student@serverb ~]$
```

2. Используйте команду **su** для переключения на пользователя **operator2** на **сервере b**.

```
[student@serverb ~]$ su - operator2  
Password: redhat  
[operator2@serverb ~]$
```

3. Используйте команду **ssh-keygen** для генерации ключей **SSH**. Не вводите парольную фразу для ключей.

```
[operator2@serverb ~]$ ssh-keygen  
Generating public/private rsa key pair.  
Enter file in which to save the key (/home/operator2/.ssh/id_rsa): Enter
```

```
Created directory '/home/operator2/.ssh'.
Enter passphrase (empty for no passphrase): Enter
Enter same passphrase again: Enter
Your identification has been saved in /home/operator2/.ssh/id_rsa.
Your public key has been saved in /home/operator2/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:JainiQdnRosC+xXhOqsJQQLzBNUldb+jJbyrCZQBERI
operator1@serverb.lab.example.com
The key's randomart image is:
+--- [RSA 2048] ---+
|E+*+ooo . |
|.= o . o . |
|o.. = . . o |
|+. + * . o . |
|+ = X . S + |
| + @ + = . |
| . + = o |
| .o . . . |
|o o.. |
+--- [SHA256] ---+
```

4. Используйте команду **ssh-copy-id** для отправки открытого ключа пары ключей SSHпользователю **operator2**на сервер**servera**.

```
[operator2@serverb ~]$ ssh-copy-id operator2@servera
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/
operator1/.ssh/id_rsa.pub"
The authenticity of host 'servera (172.25.250.10)' can't be established.
ECDSA key fingerprint is SHA256:ERTdjooOIrIwVSZQnqD5or+JbXfidg0udb3DXBuHWzA.
Are you sure you want to continue connecting (yes/no)? yes
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter
out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted
now it is to install the new keys
operator2@servera's password: redhat
Number of key(s) added: 1
```

Now try logging into the machine, with: "ssh 'operator2@servera'"
and check to make sure that only the key(s) you wanted were added.

5. Подтвердите, что вы можете успешно войти на сервер как пользователь**operator2**с помощью ключей **SSH**.

5.1. Откройте сеанс **SSH** для сервера**servera**как пользователь**operator2**.

```
[operator2@serverb ~]$ ssh operator2@servera
...output omitted...
[operator2@servera ~]$
```

Обратите внимание, что предыдущая команда **ssh** использовала ключи **SSH** для аутентификации.

5.2. Выйти из сервера.

```
[operator2@servera ~]$ exit  
logout  
Connection to servera closed.
```

- Подтвердите, что вы можете успешно войти на сервер **servera** как пользователь **root**, используя пароль суперпользователя.

6.1. Откройте сеанс **SSH** для сервера **servera** как **root**, используя пароль **redhat**.

```
[operator2@serverb ~]$ ssh root@servera  
root@servera's password: redhat  
...outputomitted...  
[root@servera ~]#
```

Обратите внимание, что предыдущая команда **ssh** использовала пароль суперпользователя для аутентификации, поскольку **SSH-ключи** не существуют для суперпользователя.

6.2. Выйти из сервера.

```
[root@servera ~]# exit  
logout  
Connection to servera closed.  
[operator2@serverb ~]$
```

- Подтвердите, что вы можете успешно войти в сервер **servera** как **operator3**, используя пароль **redhat**.

7.1. Откройте сеанс **SSH** для сервера **servera** как **operator3**, используя пароль **redhat**.

```
[operator2@serverb ~]$ ssh operator3@servera  
operator3@servera's password: redhat  
...outputomitted...  
[operator3@servera ~]$
```

Обратите внимание, что предыдущая команда **ssh** использовала пароль **operator3** для аутентификации, поскольку ключи **SSH** не существуют для **operator3**.

7.2. Выйти из сервера.

```
[operator3@servera ~]$ exit  
logout  
Connection to servera closed.  
[operator2@serverb ~]$
```

8. Настройте **sshd** на сервере **servera**, чтобы пользователи не входили в систему как **root**. При необходимости используйте **redhat** в качестве пароля суперпользователя.

8.1. Откройте сеанс **SSH** для сервера **servera** как пользователь **operator2**, используя ключи **SSH**.

```
[operator2@serverb ~]$ ssh operator2@servera  
...output omitted...  
[operator2@servera ~]$
```

8.2. На сервере **servera** переключитесь на учётную запись **root**.

```
[operator2@servera ~]$ su -  
Password: redhat  
[root@servera ~]#
```

8.3. Установите для **PermitRootLogin** значение **no** в **/etc/ssh/sshd_config** и перезагрузите **sshd**.

Вы можете использовать **vim /etc/ssh/sshd_config** для редактирования файла конфигурации **sshd**.

```
...output omitted...  
PermitRootLogin no  
...output omitted...  
[root@servera ~]# systemctl reload sshd
```

8.4. Откройте другой терминал на рабочей станции **workstation** и откройте сеанс **SSH** на сервере **serverb** как **operator2**. С сервера **serverb** попробуйте войти на сервер **servera**, как пользователь **root**. Это не должно произойти, потому что вы отключили вход пользователя **root** через **SSH** на предыдущем шаге.



ПРИМЕЧАНИЕ

Для вашего удобства вход без пароля уже настроен между рабочей станцией **workstation** и сервером **serverb** на рабочем стенде класса.

```
[student@workstation ~]$ ssh operator2@serverb  
...output omitted...  
[operator2@serverb ~]$ ssh root@servera  
root@servera's password: redhat
```

```
Permission denied, please try again.  
root@servera's password: redhat  
Permission denied, please try again.  
root@servera's password: redhat  
root@servera: Permission denied (publickey,gssapi-keyex,gssapi-with-mic,password).
```

По умолчанию команда **ssh** сначала пытается пройти аутентификацию с использованием аутентификации на основе ключей, а затем, если это не удается, аутентификации на основе пароля.

9. Настройте **sshd** на сервере **servera**, чтобы разрешить пользователям аутентифицироваться только с использованием ключей, а не паролей.

9.1. Вернитесь к первому терминалу, на котором на сервере **servera**, где активна оболочка пользователя **root**. Установите для **PasswordAuthentication** значение **no** в **/etc/ssh/sshd_config** и перезагрузите **sshd**. Вы можете использовать **vim /etc/ssh/sshd_config** для редактирования файла конфигурации **sshd**.

```
...output omitted...  
PasswordAuthentication no  
...output omitted...  
[root@servera ~]# systemctl reload sshd
```

9.2. Перейдите ко второму терминалу, на котором активна оболочка пользователя **operator2** на **serverb**, и попробуйте войти на сервере **servera**, как **operator3**. Это должно завершиться ошибкой, потому что ключи **SSH** не настроены для **operator3**, а служба **sshd** на сервере **servera** не позволяет использовать пароли для аутентификации.

```
[operator2@serverb ~]$ ssh operator3@servera  
operator3@servera: Permission denied (publickey,gssapi-keyex,gssapi-with-mic).
```



ПРИМЕЧАНИЕ

Для большей детализации вы можете использовать явные параметры **-o PubkeyAuthentication = no** и **-o PasswordAuthentication = yes** с командой **ssh**. Это позволяет вам переопределить значения по умолчанию для команды **ssh** и с уверенностью определить, что предыдущая команда завершилась ошибкой, на основе настроек, которые вы изменили в **/etc/ssh/sshd_config** на предыдущем шаге.

- 9.3. Вернитесь к первому терминалу, в активную оболочку пользователя **root** на сервере **servera**. Убедитесь, что **PubkeyAuthentication** включен в **/etc/ssh/sshd_config**. Вы можете использовать **vim /etc/ssh/sshd_config** для просмотра файла конфигурации **sshd**.

```
...output omitted...
```

```
#PubkeyAuthentication yes  
...output omitted...
```

Обратите внимание, что строка **PubkeyAuthentication** закомментирована. Любая Закомментированная строка в этом файле использует значение по умолчанию. Закомментированные строки указывают значения параметра по умолчанию. Аутентификация с открытым ключом **SSH** активна по умолчанию, как указывает строка с комментариями.

9.4. Вернитесь ко второму терминалу, на котором активна оболочка пользователя **operator2** на **serverb**, и попробуйте войти на сервер **servera** как **operator2**. Действие должно быть успешным, потому что ключи **SSH** настроены для **operator2** для входа в сервер **servera** с **serverb**.

```
[operator2@serverb ~]$ ssh operator2@servera  
...output omitted...  
[operator2@servera ~]$
```

9.5. Из второго терминала выйдите из оболочки пользователя **operator2** как на сервере **servera**, так и на сервере **serverb**.

```
[operator2@servera ~]$ exit  
logout  
Connection to servera closed.  
[operator2@serverb ~]$ exit  
logout  
Connection to serverb closed.  
[student@workstation ~]$
```

9.6. Закройте второй терминал на рабочей станции **workstation**.

```
[student@workstation ~]$ exit
```

9.7. Из первого терминала выйдите из оболочки пользователя **root** на сервере **servera**.

```
[root@servera ~]# exit  
Logout
```

9.8. Из первого терминала выйдите из оболочки пользователя **operator2**, как на сервере **servera**, так и на сервере **serverb**.

```
[operator2@servera ~]$ exit  
logout  
Connection to servera closed.  
[operator2@serverb ~]$ exit
```

```
logout  
[student@serverb ~]$
```

9.9. Выдите из **serverb** и вернитесь в оболочку пользователя**student**на рабочей станции**workstation**.

```
[student@serverb ~]$ exit  
logout  
Connection to serverb closed.  
[student@workstation ~]$
```

Завершение

На рабочей станции**workstation**, выполните скрипт**lab ssh-customize finish**, чтобы завершить это упражнение.

```
[student@workstation ~]$ lab ssh-customize finish
```

На этом пошаговое упражнение завершено.

ЛАБОРОТОРНАЯ РАБОТА

НАСТРОЙКА И ЗАЩИТА SSH

КОНТРОЛЬНЫЙ СПИСОК РАБОТЫ

В этой лабораторной работе вы настроите аутентификацию пользователей на основе ключей и отключите прямой вход в систему с правами **root** и аутентификацию по паролю для всех пользователей службы **OpenSSH** на одном из ваших серверов.

В РЕЗУЛЬТАТЕ

Вы должны быть способны:

- Выполните аутентификацию с помощью ключей **SSH**.
- Запретить пользователям напрямую входить в систему как **root** через **ssh**.
- Запретить пользователям входить в систему с помощью аутентификации на основе пароля **SSH**.

ПРЕЖДЕ ЧЕМ ВЫ НАЧНЕТЕ

Войдите на рабочую станцию **workstation**, как пользователь **student**, используя в качестве пароля слово **student**.

На рабочей станции **workstation** запустите скрипт **lab ssh-review start**, чтобы начать упражнение. Этот сценарий создает необходимые учетные записи пользователей и файлы.

```
[student@workstation ~]$ lab ssh-review start
```

1. С рабочей станции **workstation** откройте сеанс **SSH** на сервере **servera**, как пользователь **student**.
2. Используйте команду **su**, чтобы переключиться на **production1** на сервере **servera**.
3. Используйте команду **ssh-keygen** для генерации ключей **SSH** без парольной фразы для **production1** на сервере **servera**.
4. Используйте команду **ssh-copy-id**, чтобы отправить открытый ключ пары ключей **SSH** на **production1** на **serverb**.
5. Подтвердите, что пользователь **production1** может успешно войти на **serverb** с помощью ключей **SSH**.
6. Настройте **sshd** на **serverb**, чтобы пользователи не могли входить в систему как суперпользователь **root**. Используйте **redhat** в качестве пароля суперпользователя.
7. Настройте **sshd** на **serverb**, чтобы разрешить пользователям аутентифицироваться только с использованием ключей **SSH**, а не паролей.

Оценка

На рабочей станции запустите команду **lab ssh-review grade**, чтобы подтвердить успех этого упражнения.

```
[student@workstation ~]$ lab ssh-review grade
```

Завершение

На рабочей станции **workstation**, выполните скрипт **lab ssh-review finish**, чтобы завершить лабораторную работу.

```
[student@workstation ~]$ lab ssh-review finish
```

На этом лабораторная работа завершена.

РЕШЕНИЕ

НАСТРОЙКА И ЗАЩИТА SSH

КОНТРОЛЬНЫЙ СПИСОК РАБОТЫ

В этой лабораторной работе вы настроите аутентификацию пользователей на основе ключей и отключите прямой вход в систему с правами **root** и аутентификацию по паролю для всех пользователей службы **OpenSSH** на одном из ваших серверов.

В РЕЗУЛЬТАТЕ

Вы должны быть способны:

- Выполните аутентификацию с помощью ключей **SSH**.
- Запретить пользователям напрямую входить в систему как **root** через **ssh**.
- Запретить пользователям входить в систему с помощью аутентификации на основе пароля **SSH**.

ПРЕЖДЕ ЧЕМ ВЫ НАЧНЕТЕ

Войдите на рабочую станцию **workstation**, как пользователь **student**, используя в качестве пароля слово **student**.

На рабочей станции **workstation** запустите скрипт **lab ssh-review start**, чтобы начать упражнение. Этот сценарий создает необходимые учетные записи пользователей и файлы.

```
[student@workstation ~]$ lab ssh-review start
```

1. С рабочей станции **workstation** откройте сеанс **SSH** на сервере **servera**, как пользователь **student**.

```
[student@workstation ~]$ ssh student@servera  
...output omitted...  
[student@servera ~]$
```

2. Используйте команду **su**, чтобы переключиться на **production1** на сервере **servera**.

```
[student@servera ~]$ su - production1  
Password: redhat  
[production1@servera ~]$
```

3. Используйте команду **ssh-keygen** для генерации ключей **SSH** без парольной фразы для **production1** на сервере **servera**.

```
[production1@servera ~]$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/production1/.ssh/id_rsa): Enter
Created directory '/home/production1/.ssh'.
Enter passphrase (empty for no passphrase): Enter
Enter same passphrase again: Enter
Your identification has been saved in /home/production1/.ssh/id_rsa.
Your public key has been saved in /home/production1/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:CsWCAmW0g5qaJujLzIAcengNj3u21kbrPP4Ys13PXCA
    production1@servera.lab.example.com
The key's randomart image is:
+---[RSA 2048]---+
| ..o          |
| o+ . .       |
| = o . o     |
| .+  o        |
| o... . E .   |
| *o.= ... .   |
| Xo+ +oo... . |
| oo .+==+ +   |
| *o+o=*o. +   |
+---[SHA256]---
```

- Используйте команду **ssh-copy-id**, чтобы отправить открытый ключ пары ключей SSH на **production1** на **serverb**.

```
[production1@servera ~]$ ssh-copy-id production1@serverb
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/
production1/.ssh/id_rsa.pub"
The authenticity of host 'serverb (172.25.250.11)' can't be established.
ECDSA key fingerprint is SHA256:ERTdjooOIrIwVSZQnqD5or+JbXfidg0udb3DXBuHWzA.
Are you sure you want to continue connecting (yes/no)? yes
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter
out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted
now it is to install the new keys
production1@serverb's password: redhat
Number of key(s) added: 1
```

Now try logging into the machine, with: "ssh 'production1@serverb'"
and check to make sure that only the key(s) you wanted were added.

- Подтвердите, что пользователь **production1** может успешно войти на **serverb** с помощью ключей **SSH**.

```
[production1@servera ~]$ ssh production1@serverb
...output omitted...
[production1@serverb ~]$
```

6. Настройте **sshd** на **serverb**, чтобы пользователи не могли входить в систему как суперпользователь **root**. Используйте **redhat** в качестве пароля суперпользователя.

6.1. Используйте команду **su -** для переключения на **root** на **serverb**.

```
[production1@serverb ~]$ su -  
Password: redhat  
[root@serverb ~]#
```

6.2. Установите для **PermitRootLogin** значение **no** в **/etc/ssh/sshd_config** и перезагрузите **sshd**. Вы можете использовать **vim /etc/ssh/sshd_config** для редактирования файла конфигурации **sshd**.

```
...output omitted...  
PermitRootLogin no  
...output omitted...  
[root@serverb ~]# systemctl reload sshd.service
```

6.3. Откройте другой терминал на рабочей станции **workstation** и откройте сеанс **SSH** для сервера **servera** как пользователь **production1**. С сервера **servera** попробуйте войти на **serverb** как пользователь **root**. Это не должно произойти, потому что вы отключили вход для пользователя **root** через **SSH** на предыдущем шаге.



ПРИМЕЧАНИЕ

Для вашего удобства вход без пароля уже настроен между рабочей станцией **workstation** и сервером **servera** в среде класса.

```
[student@workstation ~]$ ssh production1@servera  
root@serverb's password: redhat  
Permission denied, please try again.  
root@serverb's password: redhat  
Permission denied, please try again.  
root@serverb's password: redhat  
root@serverb: Permission denied (publickey,gssapi-keyex,gssapi-with-mic,password).  
[production1@servera ~]$
```

Предыдущая команда **ssh** вернулась после трех неудачных попыток войти на сервер как **root**. По умолчанию команда **ssh** предпочитает использовать ключи **SSH** для аутентификации, но, если она не находит необходимые ключи пользователя, она запрашивает пароль пользователя для аутентификации.

7. Настройте **sshd** на **serverb**, чтобы разрешить пользователям аутентифицироваться только с использованием ключей **SSH**, а не паролей.

- 7.1. Вернитесь к первому терминалу, на котором активна оболочка пользователя **root** на **serverb**. Установите для **PasswordAuthentication** значение **no** в **/etc/ssh/sshd_config** и перезагрузите **sshd**. Вы можете использовать **vim/etc/ssh/sshd_config** для редактирования файла конфигурации **sshd**.

```
...output omitted...
PasswordAuthentication no
...output omitted...
[root@serverb ~]# systemctl reload sshd
```

- 7.2. Перейдите на второй терминал, на котором активна оболочка пользователя **production1** на сервере **servera**, и попробуйте войти на **serverb** как **production2**. Это должно завершиться ошибкой, потому что ключи **SSH** не настроены для пользователя **production2**, а служба **sshd** на **serverb** не позволяет использовать пароли для аутентификации.

```
[production1@servera ~]$ ssh production2@serverb
production2@serverb: Permission denied (publickey,gssapi-keyex,gssapi-with-mic).
```



ПРИМЕЧАНИЕ

Для большей детализации вы можете использовать явные параметры **-oPubkeyAuthentication=no** и **-o PasswordAuthentication = yes** с командой **ssh**. Это позволяет вам переопределить значения по умолчанию для команды **ssh** и с уверенностью установить, действительно ли предыдущая команда не сработает, на основе настроек, которые вы изменили в **/etc/ssh/sshd_config** на предыдущем шаге.

- 7.3. Вернитесь к первому терминалу, на котором активна оболочка пользователя **root** на **serverb**. Убедитесь, что **PubkeyAuthentication** включен в **/etc/ssh/sshd_config**. Вы можете использовать **vim/etc/ssh/sshd_config** для просмотра файла конфигурации **sshd**.

```
...output omitted...
#PubkeyAuthentication yes
...output omitted...
```

Обратите внимание, что строка **PubkeyAuthentication** закомментирована. Любая закомментированная строка в этом файле использует значение по умолчанию. Закомментированные строки указывают значения параметра по умолчанию. Аутентификация с открытым ключом **SSH** активна по умолчанию, как указывает строка с комментариями.

- 7.4. Вернитесь ко второму терминалу, на котором активна оболочка пользователя **production1** на сервере **servera**, и попробуйте войти на **serverb** как пользователь **production1**. Действие должно быть успешным, потому что ключи **SSH** настроены для **production1** для входа на **serverb** с сервера **servera**.

```
[production1@servera ~]$ ssh production1@serverb  
...output omitted...  
[production1@serverb ~]$
```

7.5. Из второго терминала выйдите из оболочки пользователя **production1** как на **serverb**, так и на сервере **servera**.

```
[production1@serverb ~]$ exit  
logout  
Connection to serverb closed.  
[production1@servera ~]$ exit  
logout  
[student@workstation ~]$
```

7.6. Закройте второй терминал на рабочей станции **workstation**.

```
[student@workstation ~]$ exit
```

7.7. Из первого терминала выйдите из оболочки пользователя **root** на **serverb**.

```
[root@serverb ~]# exit  
logout
```

7.8. Из первого терминала выйдите из оболочки пользователя **production1** как на **serverb**, так и на сервере **servera**.

```
[production1@serverb ~]$ exit  
logout  
Connection to serverb closed.  
[production1@servera ~]$ exit  
logout  
[student@servera ~]$
```

7.9. Выйдите из сервера **servera** и вернитесь в оболочку пользователя **student** на рабочей станции **workstation**.

```
[student@servera ~]$ exit  
logout  
Connection to servera closed.  
[student@workstation ~]$
```

Оценка

На рабочей станции запустите команду **lab ssh-review grade**, чтобы подтвердить успех этого упражнения.

```
[student@workstation ~]$ lab ssh-review grade
```

Завершение

На рабочей станции **workstation**, выполните скрипт **lab ssh-review finish**, чтобы завершить лабораторную работу.

```
[student@workstation ~]$ lab ssh-review finish
```

На этом лабораторная работа завершена.

РЕЗЮМЕ

В этой главе вы узнали:

- Команда **ssh** позволяет пользователям безопасно получать доступ к удаленным системам с помощью протокола **SSH**.
- Клиентская система хранит идентификаторы удаленных серверов в `~/.ssh/known_hosts` и `/etc/ssh/`
- **ssh_known_hosts**.
- **SSH** поддерживает аутентификацию на основе паролей и ключей.
- Команда **ssh-keygen** генерирует пару ключей **SSH** для аутентификации. Команда **ssh-copy-id** экспортирует открытый ключ в удаленные системы.
- Служба **sshd** реализует протокол **SSH** в системах Red Hat Enterprise Linux.
- Рекомендуется настроить **sshd** для отключения удаленного входа в систему с правами **root** и для требования аутентификации с открытым ключом, а не аутентификации на основе пароля.

ГЛАВА 11

АНАЛИЗ И ХРАНЕНИЕ ЖУРНАЛОВ

ЦЕЛЬ

Найдите и точно интерпретируйте журналы системных событий для устранения неполадок.

ЗАДАЧИ

- Описать базовую архитектуру журналирования, используемую RedHatEnterpriseLinux для записи событий.
- Интерпретируйте события в соответствующих файлах системного журнала для устранения проблем или проверки состояния системы.
- Находите и интерпретируйте записи в системном журнале для устранения проблем или проверки состояния системы.
- Настройте системный журнал для сохранения записи событий при перезагрузке сервера.
- Поддерживайте точную синхронизацию времени с **помощью NTP** и настраивайте часовой пояс, чтобы обеспечить правильные отметки времени для событий, записываемых в системный журнал и журналы.

РАЗДЕЛЫ

- Описание архитектуры системного журнала (и опрос)
- Просмотр файлов системного журнала (и упражнения с пошаговыми инструкциями)
- Просмотр записей системного журнала (и упражнения с пошаговыми инструкциями)
- Сохранение системного журнала (и упражнения с пошаговыми инструкциями)
- Поддержание точного времени (и упражнения с пошаговыми инструкциями)

ЛАБОРОТОРНАЯ РАБОТА

Анализ и хранение журналов.

ОПИСАНИЕ АРХИТЕКТУРЫ СИСТЕМНЫХ ЖУРНАЛОВ

ЗАДАЧИ

После завершения этого раздела вы сможете описать базовую архитектуру журнализации, используемую Red Hat Enterprise Linux для записи событий.

ЖУРНАЛ СИСТЕМЫ

Процессы и ядро операционной системы записывают журнал происходящих событий. Эти журналы используются для аудита системы и устранения проблем.

Многие системы записывают журналы событий в текстовые файлы, которые хранятся в каталоге **/var/log**. Данные журналы можно просмотреть с помощью обычных текстовых утилит, таких как **less** и **tail**.

Стандартная система ведения журнала, основанная на протоколе **Syslog**, встроена в **Red Hat Enterprise Linux**. Многие программы используют эту систему для записи событий и их систематизации в файлы журналов. Службы **systemd-journald** и **rsyslog** обрабатывают сообщения системного журнала в Red Hat Enterprise Linux 8.

Служба **systemd-journald** лежит в основе архитектуры регистрации событий операционной системы. Он собирает сообщения о событиях из многих источников, включая ядро, вывод на ранних этапах процесса загрузки, стандартный вывод и стандартные сообщения об ошибках от демонов при их запуске и запуске, а также события системного журнала. Затем он реструктурирует их в стандартный формат и записывает в структурированный индексированный системный журнал. По умолчанию этот журнал хранится в файловой системе, которая не сохраняется после перезагрузки. Однако служба **rsyslog** считывает сообщения системного журнала, полученные **systemd-journald**, из журнала по мере их поступления. Затем служба обрабатывает события системного журнала, записывая их в свои файлы журналов или пересыпая их другим службам в соответствии со своей собственной конфигурацией.

Служба **rsyslog** сортирует и записывает сообщения системного журнала в файлы журнала, которые сохраняются после перезагрузки в **/var/log**. Служба **rsyslog** сортирует сообщения журнала в определенные файлы журнала в зависимости от типа программы, отправившей каждое сообщение или средство, и приоритета каждого сообщения системного журнала.

Помимо файлов сообщений системного журнала, каталог **/var/log** содержит файлы журналов других служб системы. В следующей таблице перечислены некоторые полезные файлы в каталоге **/var/log**.

Выбранные файлы системного журнала

ЖУРНАЛ	ТИП СОХРАНЕННЫХ СООБЩЕНИЙ
/var/log/messages	Здесь регистрируется большинство сообщений системного журнала. Исключения включают сообщения, связанные с аутентификацией и обработкой электронной почты, выполнение заданий по расписанию, а также сообщения, относящиеся исключительно к отладке.

/var/log/secure	Сообщения системного журнала, относящиеся к событиям безопасности и аутентификации.
/var/log/maillog	Сообщения системного журнала, относящиеся к почтовому серверу.
/var/log/cron	Сообщения системного журнала, связанные с выполнением запланированного задания.
/var/log/boot.log	Консольные сообщения, не относящиеся к системному журналу, связанные с запуском системы.



ПРИМЕЧАНИЕ

Некоторые приложения не используют системный журнал для управления своими сообщениями журнала, хотя обычно они помещают свои файлы журнала в подкаталог **/var/log**. Например, веб-сервер **Apache** сохраняет сообщения журнала в файлы в подкаталоге каталога **/var/log**.



РЕКОМЕНДАЦИИ

Справочные страницы **man** **systemd-journald.service(8)**, **rsyslogd(8)**, и **rsyslog.conf(5)**

For more information refer to the *Using the log files to troubleshoot problems* section in the *Red Hat Enterprise Linux 8.0 BETA Configuring basic system settings Guide* at https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/8-beta/html/configuring_basic_system_settings/getting-started-with-systemadministration#Troubleshoot-log-files

КОНТРОЛЬНЫЙ ОПРОС

ОПИСАНИЕ АРХИТЕКТУРЫ ЖУРНАЛА СИСТЕМЫ

Выберите правильный ответ на следующие вопросы:

1. В каком из этих файлов журнала хранится большинство сообщений системного журнала, за исключением тех, которые связаны с аутентификацией, почтой, запланированными заданиями и отладкой?
 - a. /var/log/maillog
 - b. /var/log/boot.log
 - c. /var/log/messages
 - d. /var/log/secure
2. В каком файле журнала хранятся сообщения системного журнала, относящиеся к операциям безопасности и аутентификации в системе?
 - a. /var/log/maillog
 - b. /var/log/boot.log
 - c. /var/log/messages
 - d. /var/log/secure
3. Какая служба сортирует и организует сообщения системного журнала в файлы в / var / log
 - a. rsyslog
 - b. systemd-journald
 - c. auditd
 - d. tuned
4. В каком каталоге находятся файлы системного журнала, удобные для чтения?
 - a. /sys/kernel/debug
 - b. /var/log/journal
 - c. /run/log/journal
 - d. /var/log
5. В каком файле хранятся сообщения системного журнала, относящиеся к почтовому серверу?
 - a. /var/log/lastlog
 - b. /var/log/maillog
 - c. /var/log/tallylog
 - d. /var/log/boot.log
6. В каком файле хранятся сообщения системного журнала, относящиеся к запланированным заданиям?

- a. /var/log/cron
- b. /var/log/tallylog
- c. /var/log/spooler
- d. /var/log/secure

7. В каком файле хранятся сообщения консоли, связанные с запуском системы?

- a. /var/log/messages
- b. /var/log/cron
- c. /var/log/boot.log
- d. /var/log/secure

РЕШЕНИЕ

ОПИСАНИЕ АРХИТЕКТУРЫ ЖУРНАЛА СИСТЕМЫ

Выберите правильный ответ на следующие вопросы:

1. В каком из этих файлов журнала хранится большинство сообщений системного журнала, за исключением тех, которые связаны с аутентификацией, почтой, запланированными заданиями и отладкой?
 - a. /var/log/maillog
 - b. /var/log/boot.log
 - c. /var/log/messages
 - d. /var/log/secure
2. В каком файле журнала хранятся сообщения системного журнала, относящиеся к операциям безопасности и аутентификации в системе?
 - a. /var/log/maillog
 - b. /var/log/boot.log
 - c. /var/log/messages
 - d. /var/log/secure
3. Какая служба сортирует и организует сообщения системного журнала в файлы в / var / log
 - a. rsyslog
 - b. systemd-journald
 - c. auditd
 - d. tuned
4. В каком каталоге находятся файлы системного журнала, удобные для чтения?
 - a. /sys/kernel/debug
 - b. /var/log/journal
 - c. /run/log/journal
 - d. /var/log
5. В каком файле хранятся сообщения системного журнала, относящиеся к почтовому серверу?
 - a. /var/log/lastlog
 - b. /var/log/maillog
 - c. /var/log/tallylog
 - d. /var/log/boot.log
6. В каком файле хранятся сообщения системного журнала, относящиеся к запланированным заданиям?

- a. /var/log/cron**
- b. /var/log/tallylog
- c. /var/log/spooler
- d. /var/log/secure

7. В каком файле хранятся сообщения консоли, связанные с запуском системы?

- a. /var/log/messages
- b. /var/log/cron
- c. /var/log/boot.log**
- d. /var/log/secure

ПРОСМОТР ФАЙЛОВ SYSLOG

ЗАДАЧИ

После завершения этого раздела вы сможете интерпретировать события в соответствующих файлах системного журнала для устранения проблем или проверки состояния системы.

ЗАПИСЬ СОБЫТИЙ В СИСТЕМУ

Многие программы используют протокол системного журнала для регистрации событий в системе. Каждое сообщение журнала классифицируется по возможностям (типу сообщения) и приоритету (серьезности сообщения). Доступные средства описаны на странице руководства **rsyslog.conf** (5).

В следующей таблице перечислены восемь стандартных приоритетов системного журнала от самого высокого до самого низкого.

Обзор приоритетов системного журнала

КОД	ПРИОРИТЕТ	СТРОГОСТЬ
0	emerg	Система непригодна для использования
1	alert	Действия должны быть предприняты немедленно
2	crit	Критическое состояние
3	err	Состояние некритической ошибки
4	warning	Состояние предупреждения
5	notice	Нормальное, но знаменательное событие
6	info	Информационное мероприятие
7	debug	Сообщение уровня отладки

Служба **rsyslog** использует возможности и приоритет сообщений журнала, чтобы определить, как их обрабатывать. Это настраивается правилами в файле **/etc/rsyslog.conf** и в любом файле в каталоге **/etc/rsyslog.d** с расширением имени файла **.conf**. Программные пакеты могут легко добавлять правила, установив соответствующий файл в каталог **/etc/rsyslog.d**.

Каждое правило, управляющее сортировкой сообщений системного журнала, представляет собой строку в одном из файлов конфигурации. Левая сторона каждой строки указывает на возможность и серьезность сообщений системного журнала, которым соответствует правило. Правая сторона каждой строки указывает, в каком файле сохранить сообщение журнала (или куда еще доставить сообщение). Звездочка (*) - это подстановочный знак, соответствующий всем значениям.

Например, следующая строка будет записывать сообщения, отправленные в средство **authpriv** с любым приоритетом в файл **/var/log/secure**:

authpriv.*	/var/log/secure
------------	-----------------

Сообщения журнала иногда соответствуют нескольким правилам в **rsyslog.conf**. В таких случаях одно сообщение сохраняется более чем в одном файле журнала. Чтобы ограничить сохраняемые сообщения, ключевое слово **none** в поле приоритета указывает, что в данном файле не должны храниться сообщения для указанного средства.

Вместо того, чтобы записывать сообщения системного журнала в файл, их также можно выводить на терминалах всех вошедших в систему пользователей. В файле **rsyslog.conf** есть настройка для вывода всех сообщений системного журнала с приоритетом **emerg** на терминалы всех вошедших в систему пользователей.

ПРИМЕРЫ ПРАВИЛ RSYSLOG

```
##### RULES #####
# Log all kernel messages to the console.
# Logging much else clutters up the screen.
#kern.*                                     /dev/console

# Log anything (except mail) of level info or higher.
# Don't log private authentication messages!
*.info;mail.none;authpriv.none;cron.none      /var/log/messages

# The authpriv file has restricted access.
authpriv.*                                    /var/log/secure

# Log all the mail messages in one place.
mail.*                                         -/var/log/maillog

# Log cron stuff
cron.*                                         /var/log/cron

# Everybody gets emergency messages
*.emerg                                         :omusrmsg:*

# Save news errors of level crit and higher in a special file.
uucp,news.crit                                  /var/log/spooler

# Save boot messages also to boot.log
local7.*                                        /var/log/boot.log
```



ПРИМЕЧАНИЕ

Подсистема **syslog** имеет множество других функций, выходящих за рамки этого курса. Для тех, кто хочет продолжить изучение, обратитесь к странице руководства **rsyslog.conf(5)** и обширной документации HTML в **/usr/share/doc/rsyslog/html/index.html**, содержащейся в пакете **rsyslog-doc**, доступном из **AppStream** репозиторий в Red Hat Enterprise Linux 8.

РОТАЦИЯ ФАЙЛОВ ЖУРНАЛОВ

Инструмент **logrotate** сменяет файлы журналов, чтобы они не занимали слишком много места в файловой системе, содержащей каталог **/var/log**. Когда файл журнала сменяется, он переименовывается с расширением, указывающим дату, когда была выполнена ротация. Например, старый файл **/var/log/messages** может превратиться в **/var/log/messages-20190130**, если он будет заменен **30.01.2019**. После ротации старого файла журнала создается новый файл журнала, и служба, выполняющая запись в него, получает уведомление.

После определенного количества ротаций, обычно через четыре недели, самый старый файл журнала удаляется, чтобы освободить место на диске. Планировщик заданий запускает программу **logrotate** ежедневно, чтобы увидеть, нужно ли выполнить ротацию каких-либо журналов. Большинство файлов журналов меняются еженедельно, но **logrotate** сменяет быстрее или медленнее, или когда они достигают определенного размера.

В этом курсе не рассматривается настройка **logrotate**. Для получения дополнительной информации см. Справочную страницу **logrotate(8)**.

АНАЛИЗ ЗАПИСИ SYSLOG

Сообщения журнала начинаются с самого старого сообщения вверху и самого нового сообщения в конце файла журнала. Служба **rsyslog** использует стандартный формат при записи записей в файлы журналов. В следующем примере объясняется анатомия сообщения журнала в файле журнала **/var/log/secure**.

```
1 Feb 11 20:11:48 2 localhost 3 sshd[1433]: 4 Failed password for student from  
172.25.0.10 port 59344 ssh2
```

1. Отметка времени, когда запись в журнале была записана.
2. Хост, с которого было отправлено сообщение журнала
3. Имя программы или процесса и номер PID, отправившего сообщение журнала.
4. Фактическое отправленное сообщение.

ЖУРНАЛЫ МОНИТОРИНГА

Мониторинг одного или нескольких файлов журнала для событий помогает воспроизвести проблемы и проблемы. Команда **tail -f /path/to/file** выводит последние 10 строк указанного файла и продолжает выводить новые строки в файле по мере их записи.

Например, чтобы отслеживать неудачные попытки входа в систему, запустите команду **tail** в одном терминале, а затем в другом терминале, запустите команду **ssh** от имени пользователя **root**, пока пользователь пытается войти в систему.

В первом терминале запустите команду **tail**:

```
[root@host ~]# tail -f /var/log/secure
```

Во втором терминале запустите следующую команду **ssh**:

```
[root@host ~]# ssh root@localhost
root@localhost's password: redhat
...output omitted...
[root@host ~]#
```

Вернитесь к первому терминалу и просмотрите журналы.

```
...output omitted...
Feb 10 09:01:13 host sshd[2712]: Accepted password for root from 172.25.254.254
port 56801 ssh2
Feb 10 09:01:13 host sshd[2712]: pam_unix(sshd:session): session opened for user
rootby (uid=0)
```

ОТПРАВКА СООБЩЕНИЙ SYSLOG ВРУЧНУЮ

Команда **logger** может отправлять сообщения в службу **rsyslog**. По умолчанию он отправляет сообщение в средство пользователя с приоритетом уведомления (**user.notice**), если иное не указано с параметром **-p**. Полезно проверить любые изменения в конфигурации службы **rsyslog**.

Чтобы отправить сообщение службе **rsyslog**, которое записывается в файл журнала **/var/log/boot.log**, выполните следующую команду **logger**:

```
[root@host ~]# logger -p local7.notice "Log entry created on host"
```



РЕКОМЕНДАЦИИ

Справочные страницы **manlogger(1)**, **tail(1)**, **rsyslog.conf(5)**, и **logrotate(8)**
Руководство **barsyslog**

- **/usr/share/doc/rsyslog/html/index.html** предоставляется пакетом **rsyslog-doc**

Дополнительные сведения см. В разделе «Использование файлов журнала для устранения неполадок» в Руководстве по настройке основных параметров системы Red Hat Enterprise Linux 8.0 BETA по адресу

https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/8-beta/html/configuring_basic_system_settings/getting-started-with-system-administration#Troubleshoot-log-files

УПРАЖНЕНИЯ С ПОШАГОВЫМИ ИНСТРУКЦИЯМИ

ПРОСМОТР ФАЙЛОВ SYSLOG

В этом упражнении вы измените конфигурацию **rsyslog** для записи определенных сообщений журнала в новый файл.

В РЕЗУЛЬТАТЕ

У вас должна быть возможность настроить службу **rsyslog** для записи всех сообщений журнала с приоритетом**debug**(отладки) в файл журнала **/var/log/messages-debug**.

ПРЕЖДЕ ЧЕМ ВЫ НАЧНЕТЕ

Войдите на рабочую станцию, как пользователь **student**, используя в качестве пароля слово **student**.

На рабочей станции запустите сценарий **lab log-configure start**, чтобы начать упражнение. Этот сценарий обеспечивает правильную настройку среды.

```
[student@workstation ~]$ lablog-configurestart
```

1. С рабочей станции **workstation**, откройте сеанс SSH на сервере **servera**, как пользователь**student**.

```
[student@workstation ~]$ ssh student@servera  
...output omitted...  
[student@servera ~]$
```

2. Настройте **rsyslog** на сервере **servera** для записи всех сообщений с приоритетом **debug**или выше для любой службы в новый файл журнала **/var/log/messages-debug**, добавив файл конфигурации **rsyslogb/etc/rsyslog.d/debug.conf** .

- 2.1. Используйте команду **sudo -i**, чтобы переключиться на пользователя **root**. Используйтесь **student** в качестве пароля для пользователя **student**, если его спросят при выполнении команды **sudo -i**.

```
[student@servera ~]$ sudo -i  
[sudo] password for student: student  
[root@servera ~]#
```

- 2.2. Создайте файл **/etc/rsyslog.d/debug.conf** с необходимыми записями для перенаправления всех сообщений журнала, имеющих приоритет **debug**(отладки), в **/var/log/messages-debug**. Вы можете использовать команду **vim /etc/rsyslog.d/debug.conf** для создания файла со следующим содержимым.

```
*.debug /var/log/messages-debug
```

Предыдущая строка конфигурации перехватывает сообщения системного журнала, имеющие какое-либо средство с уровнем приоритета **debug** отладки или выше, и заставляет службу **rsyslog** записывать эти сообщения системного журнала в файл **/var/log/messages-debug**. Подстановочный знак (*) в поле объекта или в поле приоритета строки конфигурации указывает любое средство или любой приоритет.

2.3. Перезапустите службу **rsyslog**.

```
[root@servera ~]# systemctl restart rsyslog
```

- Убедитесь, что все сообщения журнала с приоритетом отладки отображаются в файле **/var/log/messages-debug**.

3.1. Используйте команду **logger** с параметром **-p**, чтобы сгенерировать сообщение журнала с указанием возможностей **user** и приоритетом **debug**(отладки.)

```
[root@servera ~]# logger -p user.debug "Debug Message Test"
```

3.2. Используйте команду **tail** для просмотра последних десяти сообщений журнала из файла **/var/log/messages-debug** и убедитесь, что вы видите сообщение **DebugMessageTest** среди других сообщений журнала.

```
[root@servera ~]# tail /var/log/messages-debug
Feb 13 18:22:38 servera systemd[1]: Stopping System Logging Service...
Feb 13 18:22:38 servera rsyslogd[25176]: [origin software="rsyslogd"
  swVersion="8.37.0-9.el8" x-pid="25176" x-info="http://www.rsyslog.com"] exiting
  on signal 15.
Feb 13 18:22:38 servera systemd[1]: Stopped System Logging Service.
Feb 13 18:22:38 servera systemd[1]: Starting System Logging Service...
Feb 13 18:22:38 servera rsyslogd[25410]: environment variable TZ is not set, auto
  correcting this to TZ=/etc/localtime [v8.37.0-9.el8 try http://www.rsyslog.com/
  e/2442 ]
Feb 13 18:22:38 servera systemd[1]: Started System Logging Service.
Feb 13 18:22:38 servera rsyslogd[25410]: [origin software="rsyslogd"
  swVersion="8.37.0-9.el8" x-pid="25410" x-info="http://www.rsyslog.com"] start
Feb 13 18:27:58 servera student[25416]: Debug Message Test
```

- 3.3. Выйдите из оболочки пользователя **root** и пользователя **student** на сервере **servera**, чтобы вернуться в оболочку пользователя **student** на рабочей станции **workstation**.

```
[root@servera ~]# exit
```

```
logout  
[student@servera ~]$ exit  
logout  
Connection to servera closed.  
[student@workstation ~]$
```

Завершение

На рабочей станции**workstation** запустите скрипт**lab log-configure finish**, чтобы завершить упражнение. Скрипт гарантирует, что среда будет восстановлена до чистого состояния.

```
[student@workstation ~]$ lablog-configurefinish
```

На этом пошаговое упражнение завершено.

ПРОВЕРКА ЗАПИСЕЙ СИСТЕМНОГО ЖУРНАЛА

ЗАДАЧИ

После завершения этого раздела вы сможете находить и интерпретировать записи в системном журнале для устранения проблем или проверки состояния (**troubleshoot**) системы.

ПОИСК СОБЫТИЙ

Служба **systemd-journald** хранит данные журнала в структурированном индексированном двоичном файле, который называется **journal**. Эти данные включают дополнительную информацию о событиях. Например, для журнала **syslog** включает средство и приоритет исходного сообщения.



ВАЖНО

В Red Hat Enterprise Linux 8 в каталоге **/run/log** по умолчанию хранится системный журнал. После перезагрузки содержимое каталога **/run/log** очищается. Вы можете изменить этот параметр, как это сделать, обсуждается далее в этой главе.

Чтобы получить сообщения из журнала, используйте команду **journalctl**. Вы можете использовать эту команду для просмотра всех сообщений в журнале или для поиска определенных событий на основе широкого диапазона параметров и критериев. Если вы запустите команду от имени пользователя **root**, у вас будет полный доступ к журналу. Обычные пользователи также могут использовать эту команду, но им может быть запрещено просматривать определенные сообщения.

```
[root@host ~]# journalctl
...output omitted...
Feb 21 17:46:25 host.lab.example.com systemd[24263]: Stopped target Sockets.
Feb 21 17:46:25 host.lab.example.com systemd[24263]: Closed D-Bus User Message Bus
Socket.
Feb 21 17:46:25 host.lab.example.com systemd[24263]: Closed Multimedia System.
Feb 21 17:46:25 host.lab.example.com systemd[24263]: Reached target Shutdown.
Feb 21 17:46:25 host.lab.example.com systemd[24263]: Starting Exit the Session...
Feb 21 17:46:25 host.lab.example.com systemd[24268]: pam_unix(systemduser:
session): session c>
Feb 21 17:46:25 host.lab.example.com systemd[1]: Stopped User Manager for UID
1001.
Feb 21 17:46:25 host.lab.example.com systemd[1]: user-runtime-dir@1001.service:
Unit not needed
Feb 21 17:46:25 host.lab.example.com systemd[1]: Stopping /run/user/1001 mount
wrapper...
Feb 21 17:46:25 host.lab.example.com systemd[1]: Removed slice User Slice of UID
1001.
Feb 21 17:46:25 host.lab.example.com systemd[1]: Stopped /run/user/1001 mount
wrapper.
```

```
Feb 21 17:46:36 host.lab.example.com sshd[24434]: Accepted publickey for root from
172.25.250.>
Feb 21 17:46:37 host.lab.example.com systemd[1]: Started Session 20 of user root.
Feb 21 17:46:37 host.lab.example.com systemd-logind[708]: New session 20 of user
root.
Feb 21 17:46:37 host.lab.example.com sshd[24434]: pam_unix(sshd:session): session
opened for u>
Feb 21 18:01:01 host.lab.example.com CROND[24468]: (root) CMD (run-parts /etc/
cron.hourly)
Feb 21 18:01:01 host.lab.example.com run-parts[24471]: (/etc/cron.hourly) starting
0anacron
Feb 21 18:01:01 host.lab.example.com run-parts[24477]: (/etc/cron.hourly) finished
0anacron
lines 1464-1487/1487 (END) q
```

Команда **journalctl** выделяет важные сообщения журнала: сообщения с приоритетом уведомления (**notice**) или предупреждения (**warning**) выделяются **жирным** шрифтом, а сообщения с приоритетом ошибки (**error**) или выше - **красным**.

Ключом к успешному использованию журнала для устранения неполадок и аудита является ограничение поиска по журналу для отображения только релевантных выходных данных.

По умолчанию **journalctl -n** показывает последние 10 записей журнала. Вы можете настроить это с помощью необязательного аргумента, который указывает, сколько записей журнала нужно отображать. Для последних пяти записей журнала выполните следующую команду **journalctl**:

```
[root@host ~]# journalctl -n 5
-- Logs begin at Wed 2019-02-20 16:01:17 +07, end at Thu 2019-02-21 18:01:01 +07.
--
...output omitted...
Feb 21 17:46:37 host.lab.example.com systemd-logind[708]: New session 20 of user
root.
Feb 21 17:46:37 host.lab.example.com sshd[24434]: pam_unix(sshd:session): session
opened for u>
Feb 21 18:01:01 host.lab.example.com CROND[24468]: (root) CMD (run-parts /etc/
cron.hourly)
Feb 21 18:01:01 host.lab.example.com run-parts[24471]: (/etc/cron.hourly) starting
0anacron
Feb 21 18:01:01 host.lab.example.com run-parts[24477]: (/etc/cron.hourly) finished
0anacron
lines 1-6/6 (END) q
```

Подобно команде **tail -f**, команда **journalctl -f** выводит последние 10 строк системного журнала и продолжает вывод новых записей журнала по мере их поступления в журнал. Чтобы выйти из процесса **journalctl -f**, используйте комбинацию клавиш **Ctrl + C**.

```
[root@host ~]# journalctl -f
-- Logs begin at Wed 2019-02-20 16:01:17 +07. --
...output omitted...
Feb 21 18:01:01 host.lab.example.com run-parts[24477]: (/etc/cron.hourly) finished
0anacron
Feb 21 18:22:42 host.lab.example.com sshd[24437]: Received disconnect from
172.25.250.250 port 48710:11: disconnected by user
Feb 21 18:22:42 host.lab.example.com sshd[24437]: Disconnected from user root
172.25.250.250 port 48710
Feb 21 18:22:42 host.lab.example.com sshd[24434]: pam_unix(sshd:session): session
closed for user root
Feb 21 18:22:42 host.lab.example.com systemd-logind[708]: Session 20 logged out.
Waiting for processes to exit.
Feb 21 18:22:42 host.lab.example.com systemd-logind[708]: Removed session 20.
Feb 21 18:22:43 host.lab.example.com sshd[24499]: Accepted
publickey for root from 172.25.250.250 port 48714 ssh2: RSA
SHA256:1UGybTe52L2jzEJa1HLVKn9QUCKrTv3ZzxMJo1Fro
Feb 21 18:22:44 host.lab.example.com systemd-logind[708]: New session 21 of user
root.
Feb 21 18:22:44 host.lab.example.com systemd[1]: Started Session 21 of user root.
Feb 21 18:22:44 host.lab.example.com sshd[24499]: pam_unix(sshd:session): session
opened for user root by (uid=0)
^C
```

```
[root@host ~]#
```

Чтобы помочь в устранении неполадок, вы можете отфильтровать выходные данные журнала на основе приоритета записей журнала. **journalctl -p** принимает имя или номер уровня приоритета и показывает записи журнала для записей с этим приоритетом и выше. Команда **journalctl** понимает уровни приоритета **debug, info, notice, warning, err, crit, alert, и emerg**.

Выполните следующую команду **journalctl**, чтобы вывести список записей журнала с приоритетом **err** или выше:

```
[root@host ~]# journalctl -p err
-- Logs begin at Wed 2019-02-20 16:01:17 +07, end at Thu 2019-02-21 18:01:01 +07.
--
...output omitted...
Feb 20 16:01:17 host.lab.example.com kernel: Detected CPU family 6 model 13
stepping 3
Feb 20 16:01:17 host.lab.example.com kernel: Warning: Intel Processor - this
hardware has not undergone testing by Red Hat and might not be certif>
Feb 20 16:01:20 host.lab.example.com smartd[669]: DEVICESCAN failed: glob(3)
aborted matching pattern /dev/disks/disc*
Feb 20 16:01:20 host.lab.example.com smartd[669]: In the system's table of devices
NO devices found to scan
lines 1-5/5 (END) q
```

При поиске конкретных событий вы можете ограничить вывод определенным периодом времени. У команды **journalctl** есть две опции для ограничения вывода определенным временным диапазоном: опции **--since** и **--until**. Обе опции принимают аргумент времени в формате «ГГГГ-ММ-ДД чч:мм:сс» (двойные кавычки необходимы для сохранения места в опции). Если дата опущена, команда принимает текущий день, а если время опущено, команда принимает весь день, начиная с 00:00:00. Оба варианта принимают вчера, сегодня и завтра (**yesterday**, **today**, **tomorrow**), как допустимые аргументы в дополнение к полям даты и времени.

Выполните следующую команду **journalctl**, чтобы вывести список всех записей журнала из сегодняшних записей.

```
root@host ~]# journalctl --since today
-- Logs begin at Wed 2019-02-20 16:01:17 +07, end at Thu 2019-02-21 18:31:14 +07.
--
...output omitted...
Feb 21 18:22:44 host.lab.example.com systemd-logind[708]: New session 21 of user
root.
Feb 21 18:22:44 host.lab.example.com systemd[1]: Started Session 21 of user root.
Feb 21 18:22:44 host.lab.example.com sshd[24499]: pam_unix(sshd:session): session
opened for user root by (uid=0)
Feb 21 18:31:13 host.lab.example.com systemd[1]: Starting dnf makecache...
Feb 21 18:31:14 host.lab.example.com dnf[24533]: Red Hat Enterprise Linux 8.0
AppStream (dvd) 637 kB/s | 2.8 kB 00:00
Feb 21 18:31:14 host.lab.example.com dnf[24533]: Red Hat Enterprise Linux 8.0
BaseOS (dvd) 795 kB/s | 2.7 kB 00:00
Feb 21 18:31:14 host.lab.example.com dnf[24533]: Metadata cache created.
Feb 21 18:31:14 host.lab.example.com systemd[1]: Started dnf makecache.
lines 533-569/569 (END) q
```

Выполните следующую команду **journalctl**, так чтобы вывести список всех записей журнала с 10 февраля 2019 года 20:30:00 до 13 февраля 2019 года 12:00:00.

```
[root@host ~]# journalctl --since "2014-02-10 20:30:00" --until "2014-02-13 12:00:00"
...output omitted...
```

Вы также можете указать все записи с момента времени относительно настоящего. Например, чтобы указать все записи за последний час, вы можете использовать следующую команду:

```
[root@host ~]# journalctl --since "-1 hour"
...output omitted...
```



ПРИМЕЧАНИЕ

Вы можете использовать другие, более сложные спецификации времени с параметрами **--since** и **--until**. Для некоторых примеров см. Справочную страницу **systemd.time** (7).

Помимо видимого содержимого журнала, к записям журнала прикреплены поля, которые можно увидеть, только если включен подробный вывод. Любое отображаемое дополнительное поле можно использовать для фильтрации вывода запроса журнала. Это полезно для уменьшения вывода сложных поисков определенных событий в журнале.

```
[root@host ~]# journalctl -o verbose
-- Logs begin at Wed 2019-02-20 16:01:17 +07, end at Thu 2019-02-21 18:31:14 +07.
--
...output omitted...
Thu 2019-02-21 18:31:14.509128 +07...
    _UID=0
    _GID=0
    CODE_FILE=../src/core/job.c
    CODE_LINE=826
    CODE_FUNC=job_log_status_message
    JOB_TYPE=start
    JOB_RESULT=done
    MESSAGE_ID=39f53479d3a045ac8e11786248231fbf
    _TRANSPORT=journal
    _PID=1
    _COMM=systemd
    _EXE=/usr/lib/systemd/systemd
    _CMDLINE=/usr/lib/systemd/systemd --switched-root --system --deserialize 18
    _CAP_EFFECTIVE=3fffffff
    _SELINUX_CONTEXT=system_u:system_r:init_t:s0
    _SYSTEMD_CGROUP=/init.scope
    _SYSTEMD_UNIT=init.scope
    _SYSTEMD_SLICE=--slice
    UNIT=dnf-makecache.service
    MESSAGE=Started dnf makecache.
    _HOSTNAME=host.lab.example.com
    INVOCATION_ID=d6f90184663f4309835a3e8ab647cb0e
    _SOURCE_REALTIME_TIMESTAMP=1550748674509128
lines 32239-32275/32275 (END) q
```

В следующем списке приведены общие поля системного журнала, которые можно использовать для поиска строк, относящихся к определенному процессу или событию.

- **_COMM** Имя команды.
- **_EXE** Путь к исполняемому файлу для процесса.
- **_PID** PID процесса.
- **_UID** - UID пользователя, запускающего процесс.
- **_SYSTEMD_UNIT** - модуль **systemd**, запустивший процесс.

С помощью команды **journalctl** можно объединить несколько полей системного журнала для формирования детального поискового запроса. Например, следующая команда **journalctl** показывает все записи журнала, относящиеся к модулю **sshd.service systemd** из процесса с **PID1182**.

```
[root@host ~]# journalctl _SYSTEMD_UNIT=sshd.service _PID=1182
Apr 03 19:34:27 host.lab.example.com sshd[1182]: Accepted password for root
from ::1 port 52778 ssh2
Apr 03 19:34:28 host.lab.example.com sshd[1182]: pam_unix(sshd:session): session
opened for user root by (uid=0)
...output omitted...
```



ПРИМЕЧАНИЕ

Список часто используемых полей журнала см. На странице руководства **mansystemd.journalfields (7)**.



РЕКОМЕНДАЦИИ

Справочные страницы **manjournalctl(1)**, **systemd.journal-fields(7)**, и **systemd.time(7)**

Дополнительные сведения см. В разделе «Использование файлов журнала для устранения неполадок» в Руководстве по настройке основных параметров системы (*Configuring basic system settings Guide*) Red Hat Enterprise Linux 8.0 BETA по адресу https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/8-beta/html/configuring_basic_system_settings/getting-started-with-systemadministration#Troubleshoot-log-files

УПРАЖНЕНИЯ С ПОШАГОВЫМИ ИНСТРУКЦИЯМИ

ПРОВЕРКА ЗАПИСЕЙ В ЖУРНАЛЕ СИСТЕМЫ

В этом упражнении вы будете искать в системном журнале записи о событиях, которые соответствуют определенным критериям.

В РЕЗУЛЬТАТЕ

Вы должны иметь возможность искать в системном журнале записи о событиях, основанные на различных критериях.

ПРЕЖДЕ ЧЕМ ВЫ НАЧНЕТЕ

Войдите на рабочую станцию **workstation** как пользователь **student**, используя в качестве пароля слово **student**.

На рабочей станции **workstation** выполните скрипт **lab log-query start**, чтобы начать упражнение. Этот сценарий обеспечивает правильную настройку среды.

```
[student@workstation ~]$ lablog-querystart
```

1. С рабочей станции **workstation** откройте сеанс **SSH** на сервер **servera**, как пользователь **student**.

```
[student@workstation ~]$ ssh student@servera  
...output omitted...  
[student@servera ~]$
```

2. Используйте совпадение **_PID=1** с командой **journalctl**, чтобы отображать только события журнала, происходящие из процесса **systemd**, запущенного с идентификатором процесса 1 на сервер **servera**. Чтобы выйти из **journalctl**, нажмите **Q**.

```
[student@servera ~]$ journalctl _PID=1  
...output omitted...  
Feb 13 13:21:08 localhost systemd[1]: Found device /dev/disk/by-uuid/  
cdf61ded-534c-4bd6-b458-cab18b1a72ea.  
Feb 13 13:21:08 localhost systemd[1]: Started dracut initqueue hook.  
Feb 13 13:21:08 localhost systemd[1]: Found device /dev/disk/byuuid/  
44330f15-2f9d-4745-ae2e-20844f22762d.  
Feb 13 13:21:08 localhost systemd[1]: Reached target Initrd Root Device.  
lines 1-5/5 (END) q  
[student@servera ~]$
```



ПРИМЕЧАНИЕ

Команда **journalctl** может дать другой результат в вашей системе.

- Используйте совпадение **_UID=81** с командой **journalctl**, чтобы отобразить все события журнала, происходящие из системной службы, запущенной с идентификатором пользователя 81 на сервере **servera**. Чтобы выйти из журнала, нажмите **Q**.

```
[student@servera ~]$ journalctl _UID=81
...output omitted...
Feb 22 01:29:09 servera.lab.example.com dbus-daemon[672]: [system] Activating via
systemd: service name='org.freedesktop.nm_dispatcher'
Feb 22 01:29:09 servera.lab.example.com dbus-daemon[672]: [system] Successfully
activated service 'org.freedesktop.nm_dispatcher'
lines 1-5/5 (END) q
[student@servera ~]$
```

- Используйте параметр **-pwarning** с командой **journalctl**, чтобы отображать события журнала с приоритетным **warning** (предупреждением) и выше на сервере **servera**. Чтобы выйти из журнала, нажмите **Q**.

```
student@servera ~]$ journalctl -p warning
...output omitted...
Feb 13 13:21:07 localhost kernel: Detected CPU family 6 model 13 stepping 3
Feb 13 13:21:07 localhost kernel: Warning: Intel Processor - this hardware has not
undergone testing by Red Hat and might not >
Feb 13 13:21:07 localhost kernel: acpi PNP0A03:00: fail to add MMCONFIG
information, can't access extended PCI configuration s>
Feb 13 13:21:07 localhost rpc.statd[288]: Running as root. chown /var/lib/nfs/
statd to choose different user
Feb 13 13:21:07 localhost rpc.idmapd[293]: Setting log level to 0
...output omitted...
Feb 13 13:21:13 servera.lab.example.com rsyslogd[1172]: environment variable TZ is
not set, auto correcting this to TZ=/etc/lo>
Feb 13 14:51:42 servera.lab.example.com systemd[1]: cgroup compatibility
translation between legacy and unified hierarchy sett>
Feb 13 17:15:37 servera.lab.example.com rsyslogd[25176]: environment variable TZ
is not set, auto correcting this to TZ=/etc/l>
Feb 13 18:22:38 servera.lab.example.com rsyslogd[25410]: environment variable TZ
is not set, auto correcting this to TZ=/etc/l>
Feb 13 18:47:55 servera.lab.example.com rsyslogd[25731]: environment variable TZ
is not set, auto correcting this to TZ=/etc/l>
lines 1-17/17 (END) q
[student@servera ~]$
```

5. Отобразите все события журнала, записанные за последние 10 минут с текущего времени на сервере **servera**.

5.1. Используйте параметр **--since** с командой **journalctl**, чтобы отобразить все события журнала, записанные на сервере **servera** за последние 10 минут. Чтобы выйти из журнала, нажмите **Q**.

```
[student@servera ~]$ journalctl --since "-10min"
...output omitted...
Feb 13 22:31:01 servera.lab.example.com CROND[25890]: (root) CMD (run-parts /etc/
cron.hourly)
Feb 13 22:31:01 servera.lab.example.com run-parts[25893]: (/etc/cron.hourly)
starting 0anacron
Feb 13 22:31:01 servera.lab.example.com run-parts[25899]: (/etc/cron.hourly)
finished 0anacron
Feb 13 22:31:41 servera.lab.example.com sshd[25901]: Bad protocol version
identification 'brain' from 172.25.250.254 port 37450
Feb 13 22:31:42 servera.lab.example.com sshd[25902]: Accepted publickey for root
from 172.25.250.254 port 37452 ssh2: RSA SHA2>
Feb 13 22:31:42 servera.lab.example.com systemd[1]: Started /run/user/0 mount
wrapper.
Feb 13 22:31:42 servera.lab.example.com systemd[1]: Created slice User Slice of
UID 0.
Feb 13 22:31:42 servera.lab.example.com systemd[1]: Starting User Manager for UID
0...
Feb 13 22:31:42 servera.lab.example.com systemd[1]: Started Session 118 of user
root.
Feb 13 22:31:42 servera.lab.example.com systemd-logind[712]: New session 118 of
user root.
Feb 13 22:31:42 servera.lab.example.com systemd[25906]: pam_unix(systemduser:
session): session opened for user root by (uid=0)
...output omitted...
lines 1-32/84 39% q
[student@servera ~]$
```

6. Используйте параметр **--since** и совпадение **_SYSTEMD_UNIT="sshd.service"** с командой **journalctl**, чтобы отобразить все события журнала, происходящие из службы **sshd**, записанные с **09:00:00** сегодня утром на сервере **servera**. Чтобы выйти из журнала, нажмите **Q**.

```
[student@servera ~]$ journalctl --since 09:00:00 _SYSTEMD_UNIT="sshd.service"
...output omitted...
Feb 13 13:21:12 servera.lab.example.com sshd[727]: Server listening on 0.0.0.0
port 22.
Feb 13 13:21:12 servera.lab.example.com sshd[727]: Server listening on :: port 22.
Feb 13 13:22:07 servera.lab.example.com sshd[1238]: Accepted publickey for student
from 172.25.250.250 port 50590 ssh2: RSA SH>
Feb 13 13:22:07 servera.lab.example.com sshd[1238]: pam_unix(sshd:session):
session opened for user student by (uid=0)
```

```
Feb 13 13:22:08 servera.lab.example.com sshd[1238]: pam_unix(sshd:session):  
session closed for user student  
Feb 13 13:25:47 servera.lab.example.com sshd[1289]: Accepted publickey for root  
from 172.25.250.254 port 37194 ssh2: RSA SHA25>  
Feb 13 13:25:47 servera.lab.example.com sshd[1289]: pam_unix(sshd:session):  
session opened for user root by (uid=0)  
Feb 13 13:25:47 servera.lab.example.com sshd[1289]: pam_unix(sshd:session):  
session closed for user root  
Feb 13 13:25:48 servera.lab.example.com sshd[1316]: Accepted publickey for root  
from 172.25.250.254 port 37196 ssh2: RSA SHA25>  
Feb 13 13:25:48 servera.lab.example.com sshd[1316]: pam_unix(sshd:session):  
session opened for user root by (uid=0)  
Feb 13 13:25:48 servera.lab.example.com sshd[1316]: pam_unix(sshd:session):  
session closed for user root  
Feb 13 13:26:07 servera.lab.example.com sshd[1355]: Accepted publickey for student  
from 172.25.250.254 port 37198 ssh2: RSA SH>  
Feb 13 13:26:07 servera.lab.example.com sshd[1355]: pam_unix(sshd:session):  
session opened for user student by (uid=0)  
Feb 13 13:52:28 servera.lab.example.com sshd[1473]: Accepted publickey for root  
from 172.25.250.254 port 37218 ssh2: RSA SHA25>  
Feb 13 13:52:28 servera.lab.example.com sshd[1473]: pam_unix(sshd:session):  
session opened for user root by (uid=0)  
...output omitted...  
lines 1-32 q  
[student@servera ~]$
```

7. Выдите из сервера.

```
[student@servera ~]$ exit  
logout  
Connection to servera closed.  
[student@workstation ~]$
```

Завершение

На рабочей станции запустите скрипт **lab log-query finish**, чтобы завершить это упражнение. Этот сценарий гарантирует, что среда будет восстановлена до чистого состояния.

```
[student@workstation ~]$ lab log-query finish
```

На этом пошаговое упражнение завершено.

СОХРАНЕНИЕ СИСТЕМНОГО ЖУРНАЛА

ЗАДАЧИ

После завершения этого раздела вы сможете настроить системный журнал для сохранения записей о событиях при перезагрузке сервера.

ПОСТОЯННОЕ ХРАНЕНИЕ СИСТЕМНОГО ЖУРНАЛА

По умолчанию системные журналы хранятся в каталоге `/run/log/journal`, это значит, что журналы очищаются при перезагрузке системы. Вы можете изменить параметры конфигурации службы `systemd-journald` в файле `/etc/systemd/journald.conf`, чтобы журналы сохранялись при перезагрузке.

Параметр `Storage` в файле `/etc/systemd/journald.conf` определяет, следует ли хранить системные журналы непостоянно или постоянно при перезагрузке. Установите для этого параметра значение `persistent`, `volatile` или `auto` следующим образом:

- **persistent** (постоянный): сохраняет журналы в каталоге `/var/log/journal`, который сохраняется после перезагрузки.

Если каталог `/var/log/journal` не существует, его создает служба `systemd-journald`.

- **volatile**: хранит журналы `volatile` в каталоге `/run/log/journal`.

Поскольку файловая система `/run` является временной и существует только в оперативной памяти, данные, хранящиеся в ней, включая системные журналы, не сохраняются при перезагрузке.

- **auto**: `rsyslog` определяет, использовать ли постоянное или энергозависимое хранилище. Если каталог `/var/log/journal` существует, тогда `rsyslog` использует постоянное хранилище, в противном случае - энергозависимое хранилище.

Это действие по умолчанию, если параметр `Storage` не установлен.

Преимущество постоянных системных журналов состоит в том, что данные доступны сразу при загрузке. Однако даже при наличии постоянного журнала не все данные хранятся вечно. Журнал имеет встроенный механизм ротации журналов, который запускается ежемесячно. Кроме того, по умолчанию журналам не разрешается занимать более 10% файловой системы, в которой они находятся, или оставлять менее 15% файловой системы свободными. Эти значения можно настроить как для журналов временного выполнения, так и для постоянных журналов в `/etc/systemd/journald.conf`. Текущие ограничения на размер журнала регистрируются при запуске процесса `systemd-journald`. Следующий вывод команды показывает записи журнала, которые отражают текущие ограничения размера:

```
[user@host ~]$ journalctl |grep -E 'Runtime|System journal'
```

```
Feb 25 13:01:46 localhost systemd-journald[147]: Runtime journal (/run/log/
journal/ae06db7da89142138408d77efea9229c) is 8.0M, max 91.4M, 83.4M free.
Feb 25 13:01:48 remotehost.lab.example.com systemd-journald[548]: Runtime journal
(/run/log/journal/73ab164e278e48be9bf80e80714a8cd5) is 8.0M, max 91.4M, 83.4M
free.
Feb 25 13:01:48 remotehost.lab.example.com systemd-journald[548]: System journal
(/var/log/journal/73ab164e278e48be9bf80e80714a8cd5) is 8.0M, max 3.7G, 3.7G free.
Feb 25 13:01:48 remotehost.lab.example.com systemd[1]: Starting Tell Plymouth To
Write Out Runtime Data...
Feb 25 13:01:48 remotehost.lab.example.com systemd[1]: Started Tell Plymouth To
Write Out Runtime Data.
```



ПРИМЕЧАНИЕ

В приведенном выше **grep** символ вертикальной черты (|) действует как индикатор **or**. То есть **grep** соответствует любой строке, содержащей либо строку времени выполнения, либо строку **System** из вывода **journalctl**. Это извлекает текущие ограничения на размер для хранилища непостоянных (время выполнения) журналов, а также для постоянного (системного) хранилища журналов.

Настройка постоянных системных журналов

Чтобы настроить службу **systemd-journald** на постоянное сохранение системных журналов при перезагрузке, установите для **Storage** значение **persistent** в файле **/etc/systemd/journald.conf**. Запустите любой текстовый редактор от имени суперпользователя, чтобы отредактировать файл **/etc/systemd/journald.conf**.

```
[Journal]
Storage=persistent
...output omitted...
```

После редактирования файла конфигурации перезапустите службу **systemd-journald**, чтобы изменения конфигурации вступили в силу.

```
[root@host ~]# systemctl restart systemd-journald
```

Если служба **systemd-journald** успешно перезапустится, вы увидите, что каталог **/var/log/journal** создан и содержит один или несколько подкаталогов. Эти подкаталоги содержат шестнадцатеричные символы в своих длинных именах и содержат файлы ***.journal**. Файлы ***.journal** - это двоичные файлы, в которых хранятся структурированные и проиндексированные записи журнала.

```
[root@host ~]# ls /var/log/journal  
73ab164e278e48be9bf80e80714a8cd5  
[root@host ~]# ls /var/log/journal/73ab164e278e48be9bf80e80714a8cd5  
system.journaluser-1000.journal
```

Несмотря на то, что системные журналы сохраняются при перезагрузке, вы получаете большое количество записей в выводе команды **journalctl**, которая включает записи из текущей загрузки системы, а также из предыдущих. Чтобы ограничить вывод определенной загрузкой системы, используйте параметр **-b** с командой **journalctl**. Следующая команда **journalctl** извлекает записи, ограниченные первой загрузкой системы:

```
[root@host ~]# journalctl -b 1  
...output omitted...
```

Следующая команда **journalctl** извлекает записи, ограниченные второй загрузкой системы. Следующий аргумент имеет смысл только в том случае, если система перезагружалась более двух раз:

```
[root@host ~]# journalctl -b 2
```

Следующая команда **journalctl** извлекает записи, ограниченные текущей загрузкой системы:

```
[root@host ~]# journalctl -b
```



ПРИМЕЧАНИЕ

При отладке сбоя системы с помощью постоянного журнала обычно требуется ограничить запрос журнала перезагрузкой до того, как произошел сбой. Параметр **-b** может сопровождаться отрицательным числом, указывающим, сколько предыдущих загрузок системы должен включать вывод. Например, **journalctl -b -1** ограничивает вывод только предыдущей загрузкой.



РЕКОМЕНДАЦИИ

Справочные страницы [man systemd-journald.conf\(5\)](#), [man systemd-journald\(8\)](#)

Дополнительные сведения см. В разделе «Использование файлов журнала для устранения неполадок (*Using the log files to troubleshoot problems*)» в Руководстве по настройке основных параметров системы RedHatEnterpriseLinux 8.0 BETA по адресу:

https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/8-beta/html/configuring_basic_system_settings/getting-started-with-systemadministration#Troubleshoot-log-files

УПРАЖНЕНИЯ С ПОШАГОВЫМИ ИНСТРУКЦИЯМИ

СОХРАНЕНИЕ СИСТЕМНОГО ЖУРНАЛА

В этом упражнении вы настроите системный журнал для сохранения данных после перезагрузки.

В РЕЗУЛЬТАТЕ

У вас должна быть возможность настроить системный журнал для сохранения данных после перезагрузки.

ПРЕЖДЕ ЧЕМ ВЫ НАЧНЕТЕ

Войдите на рабочую станцию **workstation** как пользователь **student**, используя в качестве пароля слово **student**.

На рабочей станции **workstation** выполните скрипт **lablog-preservestart**, чтобы начать упражнение. Этот сценарий обеспечивает правильную настройку среды.

```
[student@workstation ~]$ lablog-preservestart
```

1. С рабочей станции **workstation** откройте сеанс **SSH** на сервере **servera** как пользователь **student**.

```
[student@workstation ~]$ ssh student@servera  
...output omitted...  
[student@servera ~]$
```

2. В качестве суперпользователя убедитесь, что каталог **/var/log/journal** не существует. Используйте команду **ls**, чтобы просмотреть содержимое каталога **/var/log/journal**. Используйте **sudo** для повышения привилегий пользователя **student**. Если потребуется, используйте в качестве пароля слово **student**.

```
[student@servera ~]$ sudo ls /var/log/journal  
[sudo] password for student: student  
ls: cannot access '/var/log/journal': No such file or directory
```

Каталог **/var/log/journal** не существует, поэтому очевидно, что служба **systemd-journald** в настоящее время не сохраняет свои журналы.

3. Настройте службу **systemd-journald** на сервере для сохранения журналов, чтобы журналы сохранялись после перезагрузки.

- 3.1. Раскомментируйте строку **Storage = auto** в файле **/etc/systemd/journald.conf** и установите для **Storage** значение **persistent**. Вы можете использовать команду **sudovim/etc/systemd/journald.conf** для редактирования файла конфигурации. Введите /

Storage=auto в командном режиме в **vim**, чтобы найти строку **Storage=auto** в файле **/etc/systemd/journald.conf**. Введите:**wq!** из командного режима в **vim** для сохранения и выхода.

```
...output omitted...
[Journal]
Storage=persistent
...output omitted...
```

3.2. Используйте команду **systemctl**, для перезапуска службы **systemd-journald**, для того чтобы изменения конфигурации вступили в силу.

```
[student@servera ~]$ sudo systemctl restart systemd-journald.service
```

4. Убедитесь, что служба **systemd-journald** на сервере сохраняет свои журналы таким образом, чтобы журналы сохранялись при перезагрузках.

4.1. Используйте команду **reboot** для перезапуска сервера **servera**.

```
[student@servera ~]$ sudo reboot
Connection to servera closed by remote host.
Connection to servera closed.
[student@workstation ~]$
```

Обратите внимание, что соединение **SSH** было прервано, как только вы перезапустили систему сервера **servera**.

4.2. Снова откройте сеанс **SSH** для сервера **servera**.

```
[student@workstation ~]$ ssh student@servera
...output omitted...
[student@servera ~]$
```

4.3. Используйте команду **ls**, чтобы убедиться, что каталог **/var/log/journal** существует. Каталог **/var/log/journal** содержит подкаталог с длинной строкой шестнадцатеричных символов в качестве имени. Внутри этого каталога вы найдете журналы. Вы можете встретить каталог с другим именем, отличным от того, который показан в следующих выходных данных.

```
[student@servera ~]$ sudo ls /var/log/journal
[sudo] password for student:
73ab164e278e48be9bf80e80714a8cd5
[student@servera ~]$ sudo ls /var/log/journal/73ab164e278e48be9bf80e80714a8cd5
system.journal user-1000.journal
```

4.4. Выйти из сервера.

```
[student@servera ~]$ exit  
logout  
Connection to servera closed.
```

Завершение

На рабочей станции **workstation** запустите скрипт **lab log-preserve finish**, чтобы завершить упражнение. Данный сценарий гарантирует, что среда будет восстановлена до чистого состояния.

```
[student@workstation ~]$ lab log-preserve finish
```

На этом пошаговое упражнение завершено.

ПОДДЕРЖАНИЕ ТОЧНОГО ВРЕМЕНИ

ЗАДАЧИ

После завершения этого раздела вы сможете поддерживать точную синхронизацию времени с помощью **NTP** и настраивать часовой пояс, чтобы обеспечить правильные отметки времени для событий, записываемых системным журналом и логов.

УСТАНОВКА МЕСТНЫХ ЧАСОВ И ЧАСОВЫХ ПОЯСОВ

Правильное синхронизированное системное время имеет решающее значение для анализа файлов журнала в нескольких системах. Сетевой протокол времени (*Network Time Protocol/NTP*) - это стандартный способ для компьютеров предоставлять и получать точную информацию о времени в Интернете. Машина может получать точную информацию о времени из общедоступных служб **NTP** в Интернете, таких как проект, NTP Pool. Другой вариант - высококачественные аппаратные часы, которые показывают точное время местным клиентам.

Команда **timedatectl** показывает обзор текущих настроек системы, связанных со временем, включая текущее время, часовой пояс и настройки синхронизации **NTP** системы.

```
[user@host ~]$ timedatectl
          Local time: Fri 2019-04-05 16:10:29 CDT
          Universal time: Fri 2019-04-05 21:10:29 UTC
                  RTC time: Fri 2019-04-05 21:10:29
                  Time zone: America/Chicago (CDT, -0500)
System clock synchronized: yes
          NTP service: active
      RTC in local TZ: no
```

Доступна база данных часовых поясов, которую можно просмотреть с помощью команды **timedatectl list-timezones**.

```
[user@host ~]$ timedatectl list-timezones
Africa/Abidjan
Africa/Accra
Africa/Addis_Ababa
Africa/Algiers
Africa/Asmara
Africa/Bamako
...
```

Имена часовых поясов основаны на общедоступной базе данных часовых поясов, которую ведет **IANA**. Часовые пояса называются в зависимости от континента или океана, а затем обычно,

но не всегда, крупнейшего города в регионе часового пояса. Например, большая часть часового пояса горных районов США - **America/Denver**.

Выбор правильного имени может быть не интуитивно понятным в случаях, когда для населенных пунктов внутри часового пояса действуют другие правила перехода на летнее время. Например, в США большая часть штата Аризона (горное время США) вообще не имеет перехода на летнее время и находится в часовом поясе **America/ Phoenix**.

Команда **tzselect** полезна для определения правильных имен часовых поясов **zoneinfo**. Она интерактивно предлагает пользователю вопросы о местоположении системы и выводит название правильного часового пояса. Это не меняет настройки часового пояса системы.

Суперпользователь может изменить настройки системы для обновления текущего часового пояса с помощью команды **timedatectl set-timezone**. Следующая команда **timedatectl** обновляет текущий часовой пояс на **America/ Phoenix**.

```
[root@host ~]# timedatectl set-timezone America/Phoenix
[root@host ~]# timedatectl
          Local time: Fri 2019-04-05 14:12:39 MST
          Universal time: Fri 2019-04-05 21:12:39 UTC
                  RTC time: Fri 2019-04-05 21:12:39
                 Time zone: America/Phoenix (MST, -0700)
System clock synchronized: yes
          NTP service: active
     RTC in local TZ: no
```



ПРИМЕЧАНИЕ

Если вам нужно использовать универсальное координированное время (UTC) на определенном сервере, установите его часовой пояс на UTC. Команда **tzselect** не включает название часового пояса UTC. Используйте команду **timedatectl set-timezone UTC**, чтобы установить текущий часовой пояс системы на UTC.

Используйте команду **timedatectl set-time**, чтобы изменить текущее время системы. Время указывается в формате «ГГГГ-ММ-ДД чч:мм:сс»("YYYY-MM-DD hh:mm:ss"), где дату или время можно не указывать. Следующая команда **timedatectl** изменяет время на 09:00:00.

```
[root@host ~]# timedatectl set-time 9:00:00
[root@serverX ~]$ timedatectl
          Local time: Fri 2019-04-05 09:00:27 MST
          Universal time: Fri 2019-04-05 16:00:27 UTC
                  RTC time: Fri 2019-04-05 16:00:27
                 Time zone: America/Phoenix (MST, -0700)
System clock synchronized: yes
          NTP service: active
     RTC in local TZ: no
```

Команда **timedatectl set-ntp** включает или отключает синхронизацию **NTP** для автоматической корректировки времени. Для включения или выключения этой опции требуется аргумент «**true**» или «**false**». Следующая команда **timedatectl** включает синхронизацию **NTP**.

```
[root@host ~]# timedatectl set-ntp true
```



ПРИМЕЧАНИЕ

ПРИМЕЧАНИЕ

В Red Hat Enterprise Linux 8 команда **timedatectl set-ntp** определяет, работает ли служба **chrony** и **NTP**. Другие дистрибутивы Linux могут использовать этот параметр для настройки другой службы **NTP** или **SNTP**.

Включение или отключение **NTP** с помощью других утилит в Red Hat Enterprise Linux, например, в графическом приложении настроек **GNOME**, также обновляет этот параметр.

НАСТРОЙКА И МОНИТОРИНГ CHRONYD

Служба **chrony** отслеживает обычно неточные локальные аппаратные часы (RTC), синхронизируя их с настроенными серверами **NTP**. Если сетевое соединение недоступно, **chrony** вычисляет дрейф часов **RTC**, который записывается в **driftfile** файл, указанный в файле конфигурации **/etc/chrony.conf**.

По умолчанию служба **chrony** использует серверы из проекта пула **NTP** для синхронизации времени и не требует дополнительной настройки. Может быть полезно изменить серверы **NTP**, когда рассматриваемая машина находится в изолированной сети.

Уровень (**stratum**) источника времени **NTP** определяет его качество. **stratum** определяет количество прыжков, на которое машина удалена от высокопроизводительных опорных часов. Эталонные часы являются источником времени **stratum 0**. **NTP-сервер**, напрямую подключенный к нему, является **stratum 1**, в то время как машина, синхронизирующая время с **NTP-сервером**, является источником времени **stratum 2**.

Сервер и одноранговый узел (**peer**) - это две категории источников времени, которые вы можете указать в файле конфигурации **/etc/chrony.conf**. Сервер находится на один уровень выше локального сервера **NTP**, а партнер находится на том же уровне. Можно указать более одного сервера и более одного однорангового узла, по одному в каждой строке.

Первый аргумент строки **server** - это IP-адрес или DNS-имя **NTP-сервера**. После IP-адреса или имени сервера может быть указан ряд параметров для сервера. Рекомендуется использовать опцию **iburst**, потому что после запуска службы выполняется четыре измерения за короткий период времени для более точной начальной синхронизации часов.

Следующая строка **iburst** сервера **classroom.example.com** в файле **/etc/chrony.conf** заставляет службу **chrony** использовать источник времени **NTP classroom.example.com**.

```
# Use public servers from the pool.ntp.org project.  
...output omitted...  
server classroom.example.com iburst
```

...output omitted...

После указания **chrony** на источник местного времени **classroom.example.com** следует перезапустить службу.

```
[root@host ~]# systemctl restart chronyd
```

Команда **chronyc** действует как **клиент** службы **chrony**. После настройки синхронизации **NTP** вы должны убедиться, что локальная система беспрепятственно использует **сервер NTP** для синхронизации системных часов с помощью команды **chrony sources**. Для более подробного вывода с дополнительными пояснениями о выводе используйте команду **chronyc sources -v**.

```
[root@host ~]# chronyc sources -v
```

210 Number of sources = 1

```
.-- Source mode '^' = server, '=' = peer, '#' = local clock.  
/ .- Source state '*' = current synced, '+' = combined , '-' = not combined,  
| / '?' = unreachable, 'x' = time may be in error, '~' = time too variable.  
|| .- xxxx [ yyyy ] +/- zzzz  
|| / xxxx = adjusted offset,  
|| Log2(Polling interval) -. | yyyy = measured offset,  
|| \ | zzzz = estimated error.  
|| | |  
MS Name/IP address Stratum Poll Reach LastRx Last sample  
=====
```

MS Name/IP address	Stratum	Poll	Reach	LastRx	Last sample
^* classroom.example.com	8	6	17	23	-497ns[-7000ns] +/- 956us

Символ * в поле **S** (Исходное состояние (**Source state**)) указывает, что сервер **classroom.example.com** использовался в качестве источника времени и является **сервером NTP**, с которым в настоящее время синхронизируется машина.



РЕКОМЕНДАЦИИ

Справочные страницы **man timedatectl(1)**, **tzselect(8)**, **chrony(8)**, **chrony.conf(5)**, и **chronyc(1)**

NTP Pool Project

<http://www.pool.ntp.org/>

Time Zone Database

<http://www.iana.org/time-zones>

УПРАЖНЕНИЯ С ПОШАГОВЫМИ ИНСТРУКЦИЯМИ

ПОДДЕРЖАНИЕ ТОЧНОГО ВРЕМЕНИ

В этом упражнении вы настроите часовой пояс на сервере и убедитесь, что его системные часы синхронизированы с источником времени **NTP**.

ВРЕЗУЛЬТАТЕ

Вы должны быть способны:

- Изменить часовой пояс на сервере.
- Настройте сервер на синхронизацию своего времени с источником времени **NTP**.

ПРЕЖДЕ ЧЕМ ВЫ НАЧНЕТЕ

Войдите на рабочую станцию **workstation**, как пользователь **student**, используя в качестве пароля слово **student**.

На рабочей станции **workstation** выполните скрипт **lab log-maintenance start**, чтобы начать упражнение. Этот сценарий гарантирует, что синхронизация времени отключена в системе сервера, чтобы предоставить вам возможность вручную обновить настройки в системе и включить синхронизацию времени.

```
[student @ workstation ~] $ lab log-maintenance start
```

1. С рабочей станции **workstation** откройте сеанс SSH на сервере **servera** как студент пользователь **student**.

```
[student@workstation ~]$ ssh student@servera  
...output omitted...  
[student@servera ~]$
```

2. В рамках данного задания представьте, что система **servera** переехала на Гаити, и поэтому вам нужно соответствующим образом обновить часовой пояс. Используйте **sudo** для повышения привилегий пользователя **student** при выполнении команды **timedatectl** что бы обновить часовой пояс. Если потребуется, используйте в качестве пароля слово **student**.

- 2.1. Используйте команду **tzselect**, чтобы определить подходящий часовой пояс для Гаити.

```
[student@servera ~]$ tzselect  
Please identify a location so that time zone rules can be set correctly.  
Please select a continent, ocean, "coord", or "TZ".  
1) Africa  
2) Americas  
3) Antarctica  
4) Asia
```

- 5) Atlantic Ocean
 - 6) Australia
 - 7) Europe
 - 8) Indian Ocean
 - 9) Pacific Ocean
- 10) coord - I want to use geographical coordinates.
11) TZ - I want to specify the time zone using the Posix TZ format.

#? 2

Please select a country whose clocks agree with yours.

Please select a country whose clocks agree with yours.

1) Anguilla	19) Dominican Republic	37) Peru
2) Antigua & Barbuda	20) Ecuador	38) Puerto Rico
3) Argentina	21) El Salvador	39) St Barthelemy
4) Aruba	22) French Guiana	40) St Kitts & Nevis
5) Bahamas	23) Greenland	41) St Lucia
6) Barbados	24) Grenada	42) St Maarten (Dutch)
7) Belize	25) Guadeloupe	43) St Martin (French)
8) Bolivia	26) Guatemala	44) St Pierre & Miquelon
9) Brazil	27) Guyana	45) St Vincent
10) Canada	28) Haiti	46) Suriname
11) Caribbean NL	29) Honduras	47) Trinidad & Tobago
12) Cayman Islands	30) Jamaica	48) Turks & Caicos Is
13) Chile	31) Martinique	49) United States
14) Colombia	32) Mexico	50) Uruguay
15) Costa Rica	33) Montserrat	51) Venezuela
16) Cuba	34) Nicaragua	52) Virgin Islands (UK)
17) Curaçao	35) Panama	53) Virgin Islands (US)
18) Dominica	36) Paraguay	

#? 28

The following information has been given:

Haiti

Therefore TZ='America/Port-au-Prince' will be used.

Selected time is now: Tue Feb 19 00:51:05 EST 2019.

Universal Time is now: Tue Feb 19 05:51:05 UTC 2019.

Is the above information OK?

1) Yes

2) No

#? 1

You can make this change permanent for yourself by appending the line
TZ='America/Port-au-Prince'; export TZ
to the file '.profile' in your home directory; then log out and log in again.

Here is that TZ value again, this time on standard output so that you
can use the /usr/bin/tzselect command in shell scripts:

America/Port-au-Prince

Обратите внимание, что предыдущая команда **tzselect** отображала соответствующий часовой пояс для Гаити (Haiti).

- 2.2. Используйте команду **timedatectl**, чтобы обновить часовой пояс на сервере до Америки/Порт-о-Пренса (America/Port-au-Prince).

```
[student@servera ~]$ sudo timedatectl set-timezone America/Port-au-Prince  
[sudo] password for student: student
```

Обратите внимание, что предыдущая команда **tzselect** отображала соответствующий часовой пояс для Гаити.

- 2.3. Используйте команду **timedatectl**, чтобы убедиться, что часовой пояс был обновлен до Америка America/Port-au-Prince.

```
[student@servera ~]$ timedatectl  
  Local time: Tue 2019-02-19 01:16:29 EST  
  Universal time: Tue 2019-02-19 06:16:29 UTC  
    RTC time: Tue 2019-02-19 06:16:29  
  Time zone: America/Port-au-Prince (EST, -0500)  
System clock synchronized: no  
      NTP service: inactive  
     RTC in local TZ: no
```

3. Настройте службу chronyd на сервере **servera** для синхронизации системного времени с источником времени NTP **classroom.example.com**.

- 3.1. Отредактируйте файл **/etc/chrony.conf**, указав сервер **classroom.example.com** в качестве источника времени **NTP**. Вы можете использовать команду **sudo vim /etc/chrony.conf** для редактирования файла конфигурации. Следующий вывод показывает строку конфигурации, которую необходимо добавить в файл конфигурации:

```
...output omitted...  
server classroom.example.com iburst  
...output omitted...
```

Предыдущая строка в файле конфигурации **/etc/chrony.conf** включает параметр **iburst** для ускорения начальной синхронизации времени.

- 3.2. Используйте команду **timedatectl**, чтобы включить синхронизацию времени на сервере.

```
[student@servera ~]$ sudo timedatectl set-ntp yes
```

Предыдущая команда **timedatectl** активирует сервер **NTP** с измененным настройки в файле конфигурации **/etc/chrony.conf**. Предыдущая команда **timedatectl** может активировать службу **chronyd** или **ntpd** в зависимости от того, что в настоящее время установлено в системе.

4. Убедитесь, что настройки времени на сервере **servera** в настоящее время настроены для синхронизации с источником времени **classroom.example.com** в среде класса.
- 4.1. Используйте команду **timedatectl**, чтобы убедиться, что на сервере в настоящее время включена синхронизация времени.

```
[student@servera ~]$ timedatectl
    Local time: Tue 2019-02-19 01:52:17 EST
    Universal time: Tue 2019-02-19 06:52:17 UTC
          RTC time: Tue 2019-02-19 06:52:17
         Time zone: America/Port-au-Prince (EST, -0500)
System clock synchronized: yes
          NTP service: active
RTC in local TZ: no
```



ПРИМЕЧАНИЕ

Если предыдущие выходные данные показывают, что часы не синхронизированы, подождите две секунды и повторно запустите команду **timedatectl**. На то, чтобы успешно синхронизировать настройки времени с источником времени.

- 4.2. Используйте команду **chronyc**, чтобы убедиться, что серверная система в настоящее время синхронизирует свои настройки времени с источником времени **classroom.example.com**.

```
[student@servera ~]$ chronyc sources -v
210 Number of sources = 1

    .-- Source mode  '^' = server, '=' = peer, '#' = local clock.
    / .- Source state '*' = current synced, '+' = combined , '-' = not combined,
    | /  '?' = unreachable, 'x' = time may be in error, '~' = time too variable.
    ||                               .- xxxx [ yyyy ] +/- zzzz
    ||       Reachability register (octal) -.           |   xxxx = adjusted offset,
    ||       Log2(Polling interval) --.      |           |   yyyy = measured offset,
    ||                           \     |           |   zzzz = estimated error.
    ||                           |     |           \
MS Name/IP address          Stratum Poll Reach LastRx Last sample
=====
^* classroom.example.com        2    6   377    62    +105us[ +143us] +/-   14ms
```

Обратите внимание, что предыдущие выходные данные показывают звездочку (*) в поле исходного состояния (**S**) для источника времени **NTP classroom.example.com**. Звездочка указывает, что локальное системное время в настоящее время успешно синхронизируется с источником времени **NTP**.

4.3. Выйти из сервера.

```
[student@servera ~]$ exit  
logout  
Connection to servera closed.  
[student@workstation ~]$
```

Завершение

На рабочей станции **workstation** запустите сценарий **lab log-maintain finish**, чтобы завершить упражнение. Этот сценарий гарантирует, что исходный часовой пояс будет восстановлен вместе со всеми исходными настройками времени на сервере.

```
[student @ workstation ~] $ lab lab log-maintain finish
```

На этом пошаговое упражнение завершено.

ЛАБОРАТОРНАЯ РАБОТА

АНАЛИЗ И ХРАНЕНИЕ ЖУРНАЛОВ

КОНТРОЛЬНЫЙ СПИСОК РАБОТЫ

В этой лабораторной работе вы измените часовой пояс на существующем сервере и настройте новый файл журнала для всех событий, связанных с ошибками аутентификации.

В РЕЗУЛЬТАТЕ

Вы должен быть способен:

- Обновите часовой пояс на существующем сервере.
- Настройте новый файл журнала для хранения всех сообщений, связанных с ошибками аутентификации.

ПРЕЖДЕ ЧЕМ ВЫ НАЧНЕТЕ

Войдите на рабочую станцию **workstation** как пользователь **student**, используя в качестве пароля слово **student**.

На рабочей станции **workstation** выполните скрипт **lab log-review start**, чтобы начать упражнение. Этот сценарий записывает текущий часовой пояс системы **serverb** и обеспечивает правильную настройку среды.

```
[student@workstation ~]$ lab log-review start
```

1. С рабочей станции **workstation** откройте сеанс **SSH** на **serverb** как пользователь **student**.
2. Для этого упражнения представьте, что система **serverb** была перенесена на Ямайку, и вы должны соответствующим образом обновить часовой пояс. Используйте **sudo** для повышения привилегий пользователя **student** при выполнении команды **timedatectl** для обновления часового пояса. Если потребуется, используйте в качестве пароля слово **student**.
3. Отобразите все события журнала, записанные за предыдущие 30 минут с текущего времени на **сервереb**.
4. Создайте файл **/etc/rsyslog.d/auth-errors.conf** с необходимыми строками, чтобы помочь службе **rsyslog** записывать сообщения, связанные с аутентификацией и проблемами безопасности, в новый файл **/var/log/auth-errors**. Используйте средство **authpriv** и приоритет предупреждений в файле конфигурации.

Оценка

На рабочей станции запустите команду **lab log-review grade**, чтобы подтвердить успешное выполнение данного упражнения.

Завершение

На рабочей станции **workstation** запустите скрипт **lab log-review finish**, чтобы завершить лабораторную работу. Данный сценарий гарантирует, что исходный часовой пояс будет восстановлен вместе со всеми исходными настройками времени на **serverb**.

```
[student@workstation ~]$ lab log-review finish
```

На этом пошаговое упражнение завершено.

РЕШЕНИЕ

АНАЛИЗ И ХРАНЕНИЕ ЖУРНАЛОВ

КОНТРОЛЬНЫЙ СПИСОК РАБОТЫ

В этой лабораторной работе вы измените часовой пояс на существующем сервере и настройте новый файл журнала для всех событий, связанных с ошибками аутентификации.

В РЕЗУЛЬТАТЕ

Вы должен быть способен:

- Обновите часовой пояс на существующем сервере.
- Настройте новый файл журнала для хранения всех сообщений, связанных с ошибками аутентификации.

ПРЕЖДЕ ЧЕМ ВЫ НАЧНЕТЕ

Войдите на рабочую станцию **workstation** как пользователь **student**, используя в качестве пароля слово **student**.

На рабочей станции **workstation** выполните скрипт **lab log-review start**, чтобы начать упражнение. Этот сценарий записывает текущий часовой пояс системы **serverb** и обеспечивает правильную настройку среды.

```
[student@workstation ~]$ lab log-review start
```

1. С рабочей станции **workstation** откройте сеанс **SSH** на **serverb** как пользователь **student**.

```
[student@workstation ~]$ ssh student@serverb  
...output omitted...  
[student@serverb ~]$
```

2. Для этого упражнения представьте, что система **serverb** была перенесена на Ямайку, и вы должны соответствующим образом обновить часовой пояс. Используйте **sudo** для повышения привилегий пользователя **student** при выполнении команды **timedatectl** для обновления часового пояса. Если потребуется, используйте в качестве пароля слово **student**.

- 2.1. Используйте команду **timedatectl**, чтобы просмотреть список доступных часовых поясов и определить соответствующий часовой пояс для Ямайки (**Jamaica**).

```
[student@serverb ~]$ timedatectl list-timezones | grep Jamaica  
America/Jamaica
```

- 2.2. Используйте команду **timedatectl**, чтобы установить часовой пояс на системе **serverb** на **America/Jamaica**.

```
[student@serverb ~]$ sudo timedatectl set-timezone America/Jamaica  
[sudo] password for student:
```

- 2.3. Используйте команду **timedatectl**, чтобы убедиться, что часовой пояс успешно установлен на **America/Jamaica**.

```
[student@serverb ~]$ timedatectl  
    Local time: Tue 2019-02-19 11:12:46 EST  
Universal time: Tue 2019-02-19 16:12:46 UTC  
      RTC time: Tue 2019-02-19 16:12:45  
    Time zone: America/Jamaica (EST, -0500)  
System clock synchronized: yes  
      NTP service: active  
RTC in local TZ: no
```

3. Отобразите все события журнала, записанные за предыдущие 30 минут с текущего времени на **сервере**.

- 3.1. Используйте команду **date**, чтобы определить период времени, за который вы хотите просмотреть записи журнала.

```
[student@serverb ~]$ date  
Fri Feb 22 07:31:05 EST 2019  
[student@serverb ~]$ date -d "-30 minutes"  
Fri Feb 22 07:01:31 EST 2019
```

- 3.2. Используйте параметры **--since** и **--until** с командой **journalctl**, чтобы отобразить все события журнала, записанные на сервере за предыдущие 30 минут на системе **serverb**. Чтобы выйти из **journalctl**, нажмите **Q**.

```
[student@serverb ~]$ journalctl --since 07:01:00 --until 07:31:00  
...output omitted...  
Feb 22 07:24:28 serverb.lab.example.com systemd[1138]: Reached target Timers.  
Feb 22 07:24:28 serverb.lab.example.com systemd[1138]: Reached target Paths.  
Feb 22 07:24:28 serverb.lab.example.com systemd[1138]: Starting D-Bus User Message  
Bus Socket.  
Feb 22 07:24:28 serverb.lab.example.com systemd[1138]: Listening on D-Bus User  
Message Bus Socket.  
Feb 22 07:24:28 serverb.lab.example.com systemd[1138]: Reached target Sockets.  
Feb 22 07:24:28 serverb.lab.example.com systemd[1138]: Reached target Basic  
System.  
Feb 22 07:24:28 serverb.lab.example.com systemd[1138]: Reached target Default.
```

```
Feb 22 07:24:28 serverb.lab.example.com systemd[1138]: Startup finished in 123ms.  
Feb 22 07:24:28 serverb.lab.example.com systemd[1]: Started User Manager for UID  
1000.  
Feb 22 07:24:28 serverb.lab.example.com sshd[1134]: pam_unix(sshd:session):  
session opened for user student by (uid=0)  
Feb 22 07:26:56 serverb.lab.example.com systemd[1138]: Starting Mark boot as  
successful...  
Feb 22 07:26:56 serverb.lab.example.com systemd[1138]: Started Mark boot as  
successful.  
lines 1-36/36 (END) q  
[student@serverb ~]$
```

4. Создайте **файл /etc/rsyslog.d/auth-errors.conf** с необходимыми строками, чтобы помочь службе **rsyslog** записывать сообщения, связанные с аутентификацией и проблемами безопасности, в новый файл **/var/log/auth-errors**. Используйте средство **authpriv** и приоритет предупреждений в файле конфигурации.

- 4.1. Создайте файл **/etc/rsyslog.d/auth-errors.conf**, чтобы указать новый файл **/var/log/auth-errors** в качестве места назначения для сообщений, связанных с проблемами аутентификации и безопасности. Вы можете использовать команду **sudo vim /etc/rsyslog.d/auth-errors.conf** для создания файла конфигурации.

```
authpriv.alert /var/log/auth-errors
```

- 4.2. Перезапустите службу **rsyslog**, чтобы изменения в файле конфигурации вступили в силу.

```
[student@serverb ~]$ sudo systemctl restart rsyslog
```

- 4.3. Используйте команду **logger**, чтобы записать новое сообщение журнала в файл **/var/log/auth-errors**. Примените параметр **-p authpriv.alert** с командой **logger**, чтобы сгенерировать сообщение журнала, относящееся к проблемам аутентификации и безопасности.

```
[student@serverb ~]$ logger -p authpriv.alert "Logging test authpriv.alert"
```

- 4.4. Используйте команду **tail**, чтобы убедиться, что файл **/var/log/auth-errors** заполнен записью журнала, содержащей сообщение **Logging test authpriv.alert**.

```
[student@serverb ~]$ sudo tail /var/log/auth-errors  
Feb 19 11:56:07 serverb student[6038]: Logging test authpriv.alert
```

- 4.5. Выйти из сервера **serverb**.

```
[student@serverb ~]$ exit  
logout  
Connection to serverb closed.  
[student@workstation ~]$
```

Оценка

На рабочей станции запустите команду **lab log-review grade**, чтобы подтвердить успешное выполнение данного упражнения.

Завершение

На рабочей станции **workstation** запустите скрипт **lab log-review finish**, чтобы завершить лабораторную работу. Данный сценарий гарантирует, что исходный часовой пояс будет восстановлен вместе со всеми исходными настройками времени на **serverb**.

```
[student@workstation ~]$ lab log-review finish
```

На этом пошаговое упражнение завершено.

РЕЗЮМЕ

В этой главе вы узнали:

- Сервисы **systemd-journald** и **rsyslog** собирают и записывают сообщения журнала в соответствующие файлы.
- Каталог **/var/log** содержит файлы журналов.
- Периодическая ротация файлов журнала предотвращает заполнение ими пространства файловой системы.
- Журналы **systemd** являются временными и не сохраняются после перезагрузки.
- Сервис **chrony** помогает синхронизировать настройки времени с источником времени.
- Часовой пояс сервера может быть обновлен в зависимости от его местоположения.

ГЛАВА 12

УПРАВЛЕНИЕ СЕТЬЮ

ЦЕЛЬ

Настройте сетевые интерфейсы и параметры на серверах Red Hat Enterprise Linux.

ЗАДАЧИ

- Опишите фундаментальные концепции сетевой адресации и маршрутизации для сервера.
- Протестируйте и проверьте текущую конфигурацию сети с помощью утилит командной строки.
- Управляйте настройками сети и устройствами с помощью nmcli.
- Измените настройки сети, отредактировав файлы конфигурации.
- Настройте статическое имя хоста сервера и его разрешение имен и проверьте результаты.

РАЗДЕЛЫ

- Описание сетевых концепций (и контрольный опрос)
- Проверка конфигурации сети (и упражнения с пошаговыми инструкциями)
- Настройка сети из командной строки (и упражнения с пошаговыми инструкциями)
- Редактирование файлов конфигурации сети (и упражнения с пошаговыми инструкциями)
- Настройка имен хостов и разрешения имен (и упражнения с пошаговыми инструкциями)

ЛАБОРОТОРНАЯ РАБОТА

Управление сетью

ОПИСАНИЕ КОНЦЕПЦИИ СЕТИ

ЗАДАЧИ

После завершения этого раздела вы сможете описать фундаментальные концепции сетевой адресации и маршрутизации для сервера.

МОДЕЛЬ СЕТИ TCP/IP

Сетевая модель **TCP/IP** представляет собой упрощенный четырехуровневый набор абстракций, который описывает, как различные протоколы взаимодействуют друг с другом, чтобы компьютеры могли отправлять трафик с одной машины на другую через Интернет. Это определено RFC 1122, Требования к Интернет-хостам - Уровни связи (*Requirements for Internet Hosts – Communication Layers*). Четыре уровня:

- **Приложение (Application)**

Каждое приложение имеет спецификации для связи, чтобы клиенты и серверы могли взаимодействовать между платформами. Общие протоколы включают **SSH** (удаленный вход), **HTTPS** (безопасный Интернет), **NFS** или **CIFS** (обмен файлами) и **SMTP** (доставка электронной почты).

- **Транспорт (Transport)**

Транспортные протоколы - **TCP** и **UDP**. **TCP** - это надежная связь с установлением соединения, а **UDP** - протокол дейтаграмм без установления соединения. Протоколы приложений используют порты **TCP** или **UDP**. Список известных и зарегистрированных портов можно найти в файле **/etc/services**. Когда пакет отправляется по сети, комбинация служебного порта и **IP-адреса** образует сокет. У каждого пакета есть сокет источника и сокет назначения. Эта информация может использоваться при мониторинге и фильтрации.

- **Интернет (Internet)**

Интернет, или сетевой уровень, переносит данные от хоста источника к хосту назначения. Протоколы **IPv4** и **IPv6** являются протоколами уровня Интернета. У каждого хоста есть **IP-адрес** и префикс, используемый для определения сетевых адресов. Маршрутизаторы используются для соединения сетей.

- **Link**

Уровень связи или доступа к среде передачи обеспечивает соединение с физическим носителем. Наиболее распространенными типами сетей являются проводной **Ethernet (802.3)** и беспроводной **WLAN (802.11)**. Каждое физическое устройство имеет аппаратный адрес (**MAC**), который используется для определения места назначения пакетов в сегменте локальной сети.

ОПИСАНИЕ НАЗВАНИЙ СЕТЕВОГО ИНТЕРФЕЙСА

У каждого сетевого порта в системе есть имя, которое вы используете для его настройки и идентификации.

В более старых версиях Red Hat Enterprise Linux для каждого сетевого интерфейса использовались такие имена, как **eth0**, **eth1** и **eth2**. Имя **eth0** было первым сетевым портом, обнаруженым операционной системой, **eth1** - вторым и так далее. Однако по мере того, как устройства добавляются и удаляются, механизм обнаружения устройств и присвоения им имен может изменить, какой интерфейс получает какое имя. Кроме того, стандарт **PCIe** не гарантирует порядок, в котором устройства **PCIe** будут обнаруживаться при загрузке, что может неожиданно изменить название устройства из-за изменений во время запуска устройства или системы.

В более новых версиях Red Hat Enterprise Linux используется другая система имен. Вместо того, чтобы основываться на порядке обнаружения, имена сетевых интерфейсов назначаются на основе информации из **прошивки**, **топологии шины PCI** и **типа сетевого устройства**.

Имена сетевых интерфейсов начинаются с типа интерфейса:

- Интерфейсы **Ethernet** начинаются с **en**
- Интерфейсы **WLAN** начинаются с **wl**
- Интерфейсы **WWAN** начинаются с **ww**

Остальная часть имени интерфейса после типа будет основана на информации, предоставляемой микропрограммой сервера, или определяться местоположением устройства в топологии PCI.

- **oN** указывает, что это встроенное устройство, и встроенное ПО сервера предоставило устройству порядковый номер **N**. Итак, **eno1** - это встроенное устройство **Ethernet 1**. Многие серверы не предоставляют эту информацию.
- **sN** указывает, что это устройство находится в слоте горячей замены **PCI N**. Таким образом, **Ens3** - это **карта Ethernet** в слоте горячей замены **PCI 3**.
- **pMsN** указывает, что это устройство **PCI** нашине **M** в слоте **N**. Таким образом, **wlp4s0** - это карта **WLAN** нашине **PCI 4** в слоте **0**. Если карта является многофункциональным устройством (возможно с картой Ethernet с несколькими портами или устройством, которые имеют Ethernet и некоторые другие функции), вы можете увидеть, что к имени устройства добавляется **fN**. Таким образом, **enp0s1f0** - это функция **0** карты **Ethernet** нашине **0** в слоте **1**. Также может быть второй интерфейс с именем **enp0s1f1**, который является функцией **1** того же устройства.

Постоянное именование означает, что, если вы знаете, какое имя для сетевого интерфейса в системе, вы также знаете, что оно не изменится позже. Компромисс заключается в том, что вы не можете предположить, что система с одним интерфейсом назовет этот интерфейс **eth0**.

СЕТЬ IPV4

IPv4 - это основной сетевой протокол, используемый сегодня в Интернете. Вы должны иметь хотя бы базовое представление о сети **IPv4**, чтобы управлять сетевой связью для ваших серверов.

Адреса IPv4

Адрес IPv4 - это 32-битное число, обычно выражаемое в десятичном виде в виде четырех **8-битных октетов** в диапазоне значений от **0 до 255**, разделенных точками. Адрес разделен на две части: **сетевая часть** и **часть хоста**. Все хосты в одной подсети, которые могут общаться друг с другом напрямую без маршрутизатора, имеют одну и ту же сетевую часть; сетевая часть идентифицирует подсеть. Никакие два хоста в одной подсети не могут иметь одну и ту же часть хоста; часть хоста идентифицирует конкретный хост в подсести.

В современном Интернете размер подсети **IPv4** варьируется. Чтобы знать, какая часть **IPv4**-адреса является **сетевой**, а какая - **хост-частью**, администратор должен знать **сетевую маску**, которая назначается подсети. Сетевая маска указывает, сколько битов IPv4-адреса принадлежит подсети. Чем больше битов доступно для узла, тем больше узлов может быть в подсети.

Наименьший возможный адрес в подсети (часть хоста - это все нули в двоичном формате) иногда называют сетевым адресом. Максимально возможный адрес в подсети (часть узла - это все единицы в двоичном формате) используется для широковещательных сообщений в IPv4 и называется широковещательным адресом.

Маски сети выражаются в двух формах. **Старый синтаксис маски сети** использует **24 бита** для сетевой части и читается как **255.255.255.0**. Новый синтаксис, называемый нотацией **CIDR**, определяет **сетевой префикс /24**. Обе формы содержат одинаковую информацию; а именно, сколько ведущих битов IP-адреса вносят вклад в его сетевой адрес.

В следующих примерах показано, как связаны IP-адрес, префикс (сетевая маска), сетевая часть и часть хоста.

IP Address:

172.17.5.3 = **10101100.00010001.00000101.00000011**

Prefix: /16

Netmask:

255.255.0.0 = **11111111.11111111.00000000.00000000**

10101100.00010001.00000101.00000011

Network

Host

IP Address:

192.168.5.3 = **11000000.10101000.00000101.00000011**

Prefix: /24

Netmask:

255.255.255.0 = **11111111.11111111.11111111.00000000**

11000000.10101000.00000101.00000011

Network

Host

Figure 12.1: IPv4 addresses and netmasks

Расчет сетевого адреса для 192.168.1.107/24

Адрес хоста	192.168.1.107	11000000.10101000.00000001.01101011
Префикс сети	/24 (255.255.255.0)	11111111.11111111.11111111.00000000
Сетевой адрес	192.168.1.0	11000000.10101000.00000001.00000000
Адрес трансляции	192.168.1.255	11000000.10101000.00000001.11111111

Расчет сетевого адреса для 10.1.1.18/8

Адрес хоста	10.1.1.18	00001010.00000001.00000001.00010010
Префикс сети	/8 (255.0.0.0)	11111111.00000000.00000000.00000000
Сетевой адрес	10.0.0.0	00001010.00000000.00000000.00000000
Адрес трансляции	10.255.255.255	00001010.11111111.11111111.11111111

Расчет сетевого адреса для 172.16.181.23/19

Адрес хоста	172.168.181.23	10101100.10101000.10110101.00010111
Префикс сети	/19 (255.255.224.0)	11111111.11111111.11100000.00000000
Сетевой адрес	172.168.160.0	10101100.10101000.10100000.00000000
Адрес трансляции	172.168.191.255	10101100.10101000.10111111.11111111

Специальный адрес 127.0.0.1 всегда указывает на локальную систему («**localhost**»), а сеть **127.0.0.0/8** принадлежит локальной системе, так что она может разговаривать сама с собой, используя сетевые протоколы.

Маршрутизация IPv4

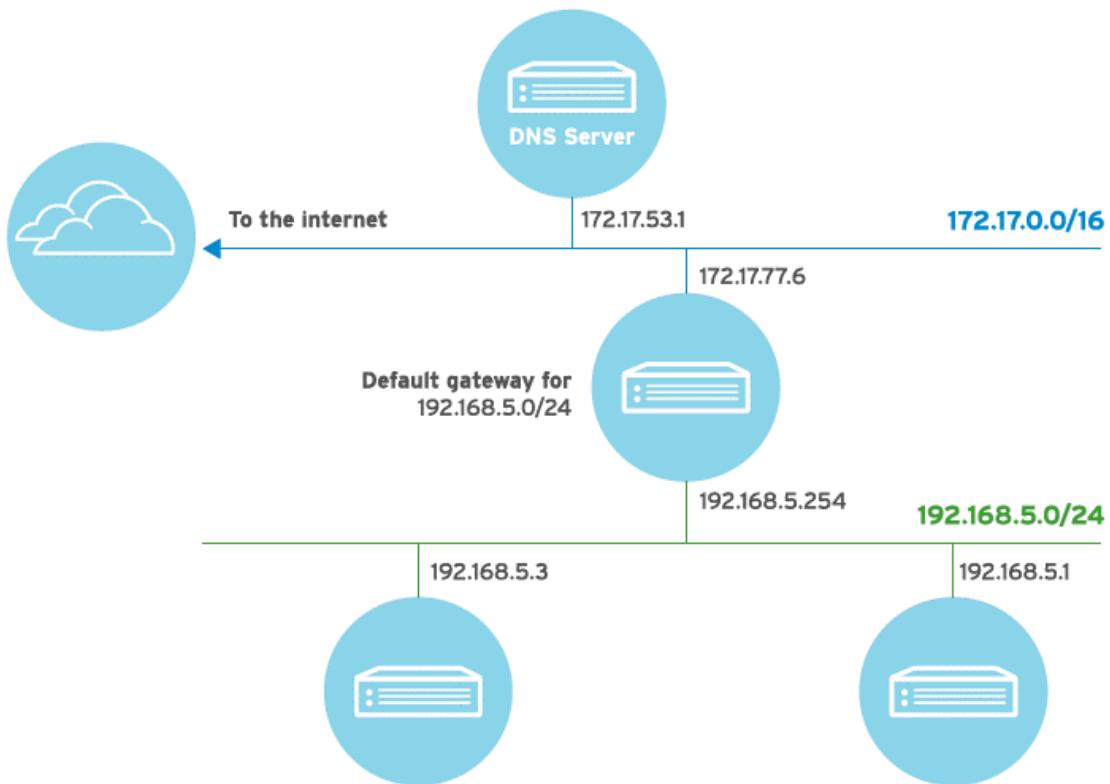
Независимо от того, используется ли IPv4 или IPv6, сетевой трафик должен перемещаться от хоста к хосту и из сети в сеть. У каждого хоста есть таблица маршрутизации (*routing table*), в которой указано, как маршрутизировать трафик для определенных сетей. В записи таблицы маршрутизации указывается сеть назначения, какой интерфейс использовать при отправке трафика, и IP-адрес любого промежуточного маршрутизатора, необходимого для ретрансляции сообщения в его конечный пункт назначения. Запись в таблице маршрутизации, соответствующая получателю сетевого трафика, используется для его маршрутизации. Если совпадают две записи, используется одна с самым длинным префиксом.

Если сетевой трафик не соответствует более конкретному маршруту, в таблице маршрутизации обычно есть запись для маршрута (*default route*) по умолчанию для всего Интернета IPv4: 0.0.0.0/0. Этот маршрут по умолчанию указывает на маршрутизатор (*router*) в доступной подсети (то есть в подсети, у которой есть более конкретный маршрут в таблице маршрутизации хоста).

Если маршрутизатор получает трафик, который не адресован ему вместо того, чтобы игнорировать его, как обычный хост, он пересыпает трафик на основе своей собственной таблицы маршрутизации. Маршрутизатор может отправить трафик непосредственно на хост назначения (если маршрутизатор находится в подсети назначения) или он может быть перенаправлен на

другой маршрутизатор. Этот процесс пересылки продолжается до тех пор, пока трафик не достигнет своего конечного пункта назначения.

Рисунок 12.2: Пример топологии сети



Пример таблицы маршрутизации

ПУНКТ НАЗНАЧЕНИЯ	ИНТЕРФЕЙС	МАРШРУТИЗАТОР (ПРИ НЕОБХОДИМОСТИ)
192.0.2.0/24	wlo1	
192.168.5.0/24	enp3s0	
0.0.0.0/0 (по умолчанию)	enp3s0	192.168.5.254

В этом примере трафик, направляемый на IP-адрес 192.0.2.102 от этого хоста, передается непосредственно в это место назначения через беспроводной интерфейс wlo1, поскольку он наиболее точно соответствует маршруту 192.0.2.0/24. Трафик для IP-адреса 192.168.5.3 передается непосредственно в этот пункт назначения через интерфейс Ethernet enp3s0, поскольку он наиболее точно соответствует маршруту 192.168.5.0/24.

Трафик на IP-адрес 10.2.24.1 передается через интерфейс enp3s0 Ethernet на маршрутизатор по адресу 192.168.5.254, который пересыпает этот трафик в его конечный пункт назначения. Этот трафик наиболее точно соответствует маршруту 0.0.0.0/0, поскольку в таблице маршрутизации

этого хоста нет более конкретного маршрута. Маршрутизатор использует свою собственную таблицу маршрутизации, чтобы определить, куда направить этот трафик в следующий раз.

IPv4-адрес и настройка маршрута

Сервер может автоматически настраивать свои сетевые параметры IPv4 во время загрузки с DHCP-сервера. Демон локального клиента запрашивает ссылку на параметры сервера и сети и получает аренду на использование этих параметров в течение определенного периода времени. Если клиент периодически не запрашивает продление аренды, он может потерять настройки конфигурации сети.

В качестве альтернативы вы можете настроить сервер на использование статической сетевой конфигурации. В этом случае сетевые настройкичитываются из локальных файлов конфигурации. Вы должны получить правильные настройки у своего сетевого администратора и обновить их вручную по мере необходимости, чтобы избежать конфликтов с другими серверами.

СЕТЬ IPV6

IPv6 предназначен в качестве возможной замены сетевого протокола **IPv4**. Вам нужно будет понять, как это работает, поскольку все большее количество производственных систем используют адресацию **IPv6**. Например, многие интернет-провайдеры уже используют **IPv6** для внутренней связи и сетей управления устройствами, чтобы сохранить скучные адреса **IPv4** для нужд клиентов.

IPv6 также может использоваться параллельно с **IPv4** в модели с двумя стеками. В этой конфигурации сетевой интерфейс может иметь адрес или адреса **IPv6**, а также адреса **IPv4**. Red Hat Enterprise Linux по умолчанию работает в режиме двойного стека.

Адреса IPv6

Адрес **IPv6** - это 128-битное число, обычно выражаемое в виде восьми разделенных двоеточиями групп по четыре шестнадцатеричных полубайта (полубайта). Каждый полубайт представляет четыре бита адреса IPv6, поэтому каждая группа представляет 16 бит адреса IPv6.

```
2001:0db8:0000:0010:0000:0000:0000:0001
```

Чтобы упростить запись адресов **IPv6**, нет необходимости записывать начальные нули в группе, разделенной двоеточиями. Однако, по крайней мере, одна шестнадцатеричная цифра должна быть записана в каждой группе, разделенной двоеточиями.

```
2001:db8:0:10:0:0:0:1
```

Поскольку адреса с длинными строками нулей являются общими, одна или несколько последовательных групп нулей могут быть объединены ровно с одним *exactly one :: block*.

```
2001:db8:0:10::1
```

Обратите внимание, что в соответствии с этими правилами **2001:db8::0010:0:0:1** будет еще одним менее удобным способом записи адреса в качестве примера. Но это действительное представление того же адреса, и это может сбить с толку администраторов, плохо знакомых с IPv6. Несколько советов по написанию последовательно читаемых адресов:

- Подавить начальные нули в группе.
- Используйте ::, чтобы сократить как можно больше.
- Если адрес содержит две последовательные группы нулей одинаковой длины, предпочтительно сократить крайние левые группы нулей до :: и крайние правые группы до :0: для каждой группы.
- Хотя это разрешено, не используйте :: для сокращения одной группы нулей. Используйте вместо этого :0: и сохраняйте запись :: для последовательных групп нулей.
- Всегда используйте строчные буквы для шестнадцатеричных чисел от a до f.



ВАЖНО

При включении сетевого порта TCP или UDP после адреса IPv6 всегда заключайте его в квадратные скобки, чтобы порт выглядел как часть адреса IPv6. IPv6-адрес в квадратных скобках, чтобы порт не выглядел как часть адреса.

[2001:db8:0:10::1]:80

Подсети IPv6

Обычный одноадресный адрес IPv6 делится на две части: префикс сети и идентификатор интерфейса. Префикс сети определяет подсеть. Никакие два сетевых интерфейса в одной подсети не могут иметь одинаковый идентификатор интерфейса; идентификатор интерфейса определяет конкретный интерфейс в подсети.

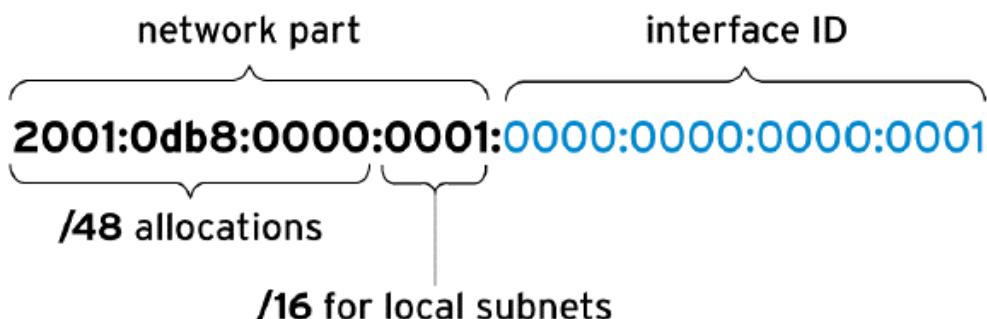
В отличие от **IPv4**, **IPv6** имеет стандартную маску подсети, которая используется почти для всех обычных адресов, /64. В этом случае половина адреса является префиксом сети, а половина - идентификатором интерфейса. Это означает, что одна подсеть может содержать столько хостов, сколько необходимо.

Обычно сетевой провайдер выделяет организации более короткий префикс, например /48. Это оставляет остальную часть сети для назначения подсетей (всегда длиной /64) из этого выделенного префикса. Для распределения /48 остается 16 бит для подсетей (до 65536 подсетей).

Рисунок 12.3: Части IPv6-адреса и подсети

IPv6 address is **2001:db8:0:1::1/64**

Allocation from provider is **2001:db8::/48**



Общие адреса IPv6 и сети

АДРЕС IPV6 ИЛИ СЕТЬ	ЦЕЛЬ	ОПИСАНИЕ
::1/128	localhost	IPv6 эквивалент 127.0.0.1/8 , установленный на интерфейсе обратной связи.
::	Неопределённый адрес	IPv6 эквивалент 0.0.0.0 . Для сетевой службы это может означать, что она прослушивает все настроенные IP-адреса.
::/0	Маршрут по умолчанию (Интернет IPv6)	IPv6 эквивалент 0.0.0.0/0 . Маршрут по умолчанию в таблице маршрутизации соответствует этой сети; маршрутизатор для этой сети — это то место, куда отправляется весь трафик, для которого нет лучшего маршрута.
2000::/3	Глобальные одноадресные адреса	«Обычные» адреса IPv6 в настоящее время выделяются из этого пространства IANA. Это эквивалентно всем сетям от 2000::/16 до 3fff::/16 .
fd00::/8	Уникальные локальные адреса (RFC 4193)	IPv6 не имеет прямого эквивалента частного адресного пространства RFC 1918, хотя это близко. Сайт может использовать их для самостоятельного выделения пространства частных маршрутизуемых IP-адресов внутри организации, но эти сети нельзя использовать в глобальной сети Интернет. Сайт должен случайным образом выбрать /48 из этого пространства, но обычно он может подсеть выделение в сети /64 .

fe80::/10	Адреса локальных соединений	Каждый интерфейс IPv6 автоматически настраивает индивидуальный адрес локальной связи, который работает только на локальном канале в сети fe80::/64 . Однако весь диапазон fe80::/10 зарезервирован для будущего использования локальной связью. Подробнее об этом мы поговорим позже.
ff00::/8	Многоадресная рассылка	IPv6 эквивалент 224.0.0.0/4 . Многоадресная передача используется для одновременной передачи на несколько хостов и особенно важна в IPv6, поскольку у нее нет широковещательных адресов.



ВАЖНО

В таблице выше перечислены распределения сетевых адресов, которые зарезервированы для определенных целей. Эти распределения могут состоять из множества различных сетей. Помните, что сети IPv6, выделенные из пространств глобальной одноадресной и локальной одноадресной рассылки, имеют стандартную маску подсети /64.

Локальный адрес в IPv6 - это не маршрутизуемый адрес, используемый только для связи с хостами по определенному сетевому соединению. Каждый сетевой интерфейс в системе автоматически конфигурируется с локальным адресом соединения в сети **fe80::/64**. Чтобы гарантировать его уникальность, идентификатор интерфейса локального адреса канала создается на основе **аппаратного адреса Ethernet** сетевого интерфейса. Обычная процедура преобразования **48-битного MAC-адреса в 64-битный идентификатор** интерфейса - это инвертировать 7-й бит MAC-адреса и вставить **ff:fe** между двумя его средними байтами.

- **Префикс сети:** fe80::/64
- **MAC-адрес:** 00:11:22:aa:bb:cc
- **Локальный адрес соединения:** fe80::211:22ff:fea:bbcc/64

Адреса локальных ссылок других машин могут использоваться как обычные адреса другими хостами в той же ссылке. Поскольку каждая ссылка имеет сеть **fe80::/64**, таблицу маршрутизации нельзя использовать для правильного выбора исходящего интерфейса. Линк для использования при «разговоре» с локальным адресом соединения должна быть указана с идентификатором области в конце адреса. Идентификатор области состоит из символа **%**, за которым следует имя сетевого интерфейса.

Например, чтобы использовать **ping6** для проверки связи с локальным адресом ссылки **fe80::211:22ff:fea:bbcc**, используя канал, подключенный к сетевому интерфейсу **ens3**, правильный синтаксис команды будет следующим:

```
[user@host ~]$ ping6 fe80::211:22ff:feaa:bbcc%ens3
```



ПРИМЕЧАНИЕ

Идентификаторы области необходимы только при обращении к адресам, имеющим область действия «ссылка “[link](#)”». Обычные глобальные адреса используются так же, как и в IPv4, и их исходящие интерфейсы выбираются из таблицы маршрутизации.

Многоадресная рассылка позволяет одной системе отправлять трафик на специальный IP-адрес, который получают несколько систем. Он отличается от широковещательной передачи, поскольку трафик получают только определенные системы в сети. Он также отличается от широковещательной рассылки в IPv4, поскольку некоторый многоадресный трафик может быть направлен в другие подсети в зависимости от конфигурации ваших сетевых маршрутизаторов и систем.

Многоадресная рассылка играет большую роль в **IPv6**, чем в **IPv4**, поскольку в **IPv6** нет широковещательного адреса. Одним из ключевых адресов многоадресной рассылки в **IPv6** является **ff02::1**, локальный адрес для всех узлов. Пинг этого адреса отправляет трафик на все узлы в ссылке. Адреса многоадресной рассылки в области связи (начиная с **ff02::/8**) должны быть указаны с идентификатором области, как и локальный адрес ссылки.

```
[user@host ~]$ ping6 ff02::1%ens3
PING ff02::1%ens3(ff02::1) 56 data bytes
64 bytes from fe80::211:22ff:feaa:bbcc: icmp_seq=1 ttl=64 time=0.072 ms
64 bytes from fe80::200:aaff:fe33:2211: icmp_seq=1 ttl=64 time=102 ms (DUP!)
64 bytes from fe80::bcd:efff:fea1:b2c3: icmp_seq=1 ttl=64 time=103 ms (DUP!)
64 bytes from fe80::211:22ff:feaa:bbcc: icmp_seq=2 ttl=64 time=0.079 ms
...output omitted...
```

Конфигурация IPv6-адреса

IPv4 имеет два способа настройки адресов на сетевых интерфейсах. Сетевые адреса могут быть настроены на интерфейсах вручную администратором или динамически из сети с помощью **DHCP**. **IPv6** также поддерживает ручную настройку и два метода динамической настройки, один из которых - **DHCPv6**.

Идентификаторы интерфейсов для статических адресов **IPv6** могут быть выбраны по желанию, как и **IPv4**. В **IPv4** есть два адреса в сети, которые нельзя использовать: самый низкий адрес в подсети и самый высокий адрес в подсети. В **IPv6** следующие идентификаторы интерфейсов зарезервированы и не могут использоваться для обычного сетевого адреса на хосте.

- Идентификатор из нулей **0000:0000:0000:0000** («произвольный маршрутизатор подсети»), используемый всеми маршрутизаторами в канале. (Для сети **2001:db8::/64** это будет адрес **2001:db8::**)
- Идентификаторы с **fdff:ffff:ffff:ff80** до **fdff:ffff:ffff:ffff**.

DHCPv6 работает иначе, чем **DHCP** для **IPv4**, потому что нет широковещательного адреса. По сути, хост отправляет запрос **DHCPv6** со своего локального адреса канала на порт **547/UDP** на **ff02::1:2**, группе локальной многоадресной рассылки **all-dhcp-servers**. Затем сервер **DHCPv6** обычно отправляет ответ с соответствующей информацией на порт **546/UDP** на локальном адресе канала клиента.

Пакет **dhcp** в Red Hat Enterprise Linux 8 обеспечивает поддержку сервера **DHCPv6**.

Помимо **DHCPv6**, **IPv6** также поддерживает второй метод динамической конфигурации, называемый автоконфигурацией адреса без сохранения состояния (**Stateless Address Autoconfiguration SLAAC**). Используя **SLAAC**, хост обычно вызывает свой интерфейс с адресом **linklocal fe80::/64**. Затем он отправляет «запрос маршрутизатора (**router solicitation**)» к **ff02::2**, группе многоадресной рассылки **allrouters**, работающей по локальному каналу. Маршрутизатор **IPv6** на локальном канале отвечает на локальный адрес узла с префиксом сети и, возможно, другой информацией. Затем хост использует этот сетевой префикс с идентификатором интерфейса, который он обычно конструирует так же, как конструируются локальные адреса ссылки. Маршрутизатор периодически отправляет многоадресные обновления («объявления маршрутизатора» (**router advertisements**)) для подтверждения или обновления предоставленной информации.

Пакет **radvd** в Red Hat Enterprise Linux 8 позволяет маршрутизатору **IPv6** на базе Red Hat Enterprise Linux предоставлять **SLAAC** через объявления маршрутизатора.



ВАЖНО

Типичный компьютер Red Hat Enterprise Linux 8, настроенный для получения адресов **IPv4** через **DHCP**, обычно также настроен на использование **SLAAC** для получения адресов **IPv6**. Это может привести к тому, что машины неожиданно получат адреса **IPv6** при добавлении маршрутизатора **IPv6** в сеть.

Некоторые развертывания **IPv6** сочетают **SLAAC** и **DHCPv6**, используя **SLAAC** только для предоставления информации о сетевом адресе и **DHCPv6** для предоставления другой информации, например, какие **DNS**-серверы и поисковые домены нужно настроить.

Имена хостов и IP-адреса

Было бы неудобно, если бы вам всегда приходилось использовать IP-адреса для связи с вашими серверами. Люди обычно предпочитают работать с именами, а не с длинными и трудно запоминаемыми числами. Итак, в Linux есть несколько механизмов для сопоставления имени хоста с IP-адресом, которые в совокупности называются разрешением имен.

Один из способов - установить статическую запись для каждого имени в файле **/etc/hosts** в каждой системе. Это требует, чтобы вы вручную обновляли копию файла на каждом сервере.

Для большинства хостов вы можете найти адрес по имени хоста (или имя хоста из адреса) в сетевой службе, называемой системой доменных имен (**DNS**). **DNS** - это распределенная сеть серверов, обеспечивающая сопоставление имен хостов с **IP-адресами**. Чтобы служба имен работала, хост должен быть указан на сервере имен. Этот сервер имен не обязательно должен находиться в одной подсети; он просто должен быть доступен хосту. Обычно это настраивается с помощью **DHCP** или статической настройки в файле с именем **/etc/resolv.conf**. В следующих разделах этой главы мы обсудим, как настроить разрешение имен.



РЕКОМЕНДАЦИИ

Справочные страницы **man services(5)**, **ping(8)**, **biosdevname(1)**, и **udev(7)**

Дополнительная информация может быть доступна в главах о настройке сети и согласованном именовании сетевых устройств в Руководстве по работе с сетью Red Hat Enterprise Linux для Red Hat Enterprise Linux 8, которое можно найти по адресу

<https://access.redhat.com/documentation/>

Понимание предсказуемых имен сетевых устройств systemd

<https://major.io/2015/08/21/understanding-systemds-predictable-networkdevice-names/>

Избранные ссылки IETF RFC:

RFC 2460: Internet Protocol, Version 6 (IPv6) Specification

<http://tools.ietf.org/html/rfc2460>

RFC 4291: IP Version 6 Addressing Architecture

<http://tools.ietf.org/html/rfc4291>

RFC 5952: A Recommendation For IPv6 Address Text Representation

<http://tools.ietf.org/html/rfc5952>

RFC 4862: IPv6 Stateless Address Autoconfiguration

<http://tools.ietf.org/html/rfc4862>

RFC 3315: Dynamic Host Configuration Protocol for IPv6 (DHCPv6)

<http://tools.ietf.org/html/rfc3315>

RFC 3736: Stateless Dynamic Host Configuration Protocol (DHCP) Service for IPv6

<http://tools.ietf.org/html/rfc3736>

RFC 4193: Unique Local IPv6 Unicast Addresses

<http://tools.ietf.org/html/rfc4193>

КОНТРОЛЬНЫЙ ОПРОС

ОПИСАНИЕ КОНЦЕПЦИИ СЕТИ

Выберите правильный ответ на следующие вопросы:

1. Какое число представляет собой размер IPv4-адреса в битах?
 - a. 4
 - b. 8
 - c. 16
 - d. 32
 - e. 64
 - f. 128
2. Какой термин определяет, сколько начальных битов IP-адреса вносят вклад в его сетевой адрес?
 - a. netscope
 - b. netmask
 - c. subnet
 - d. multicast
 - e. netaddr
 - f. network
3. Какой адрес представляет собой действительный IP-адрес хоста IPv4?
 - a. 192.168.1.188
 - b. 192.168.1.0
 - c. 192.168.1.255
 - d. 192.168.1.256
4. Какое число представляет собой размер IPv6-адреса в битах?
 - a. 4
 - b. 8
 - c. 16
 - d. 32
 - e. 64
 - f. 128
5. Какой адрес не соответствует действительному IPv6-адресу?
 - a. 2000:0000:0000:0000:0000:0000:0001
 - b. 2::1
 - c. ::
 - d. ff02::1:0:0
 - e. 2001:3::7:0:2
 - f. 2001:db8::7::2

g. 2000::1

6. Какой термин позволяет одной системе отправлять трафик на специальный IP-адрес, который получает несколько систем?

- a.** netscope
- b.** netmask
- c.** subnet
- d.** multicast
- e.** netaddr
- f.** network

КОНТРОЛЬНЫЙ ОПРОС

ОПИСАНИЕ КОНЦЕПЦИИ СЕТИ

Выберите правильный ответ на следующие вопросы:

- 1.** Какое число представляет собой размер IPv4-адреса в битах?
 - a. 4
 - b. 8
 - c. 16
 - d. 32
 - e. 64
 - f. 128

- 2.** Какой термин определяет, сколько начальных битов IP-адреса вносят вклад в его сетевой адрес?
 - a. netscope
 - b. netmask
 - c. subnet
 - d. multicast
 - e. netaddr
 - f. network

- 3.** Какой адрес представляет собой действительный IP-адрес хоста IPv4?
 - a. 192.168.1.188
 - b. 192.168.1.0
 - c. 192.168.1.255
 - d. 192.168.1.256

- 4.** Какое число представляет собой размер IPv6-адреса в битах?
 - a. 4
 - b. 8
 - c. 16
 - d. 32
 - e. 64
 - f. 128

- 5.** Какой адрес не соответствует действительному IPv6-адресу?
 - a. 2000:0000:0000:0000:0000:0000:0001
 - b. 2::1
 - c. ::
 - d. ff02::1:0:0
 - e. 2001:3::7:0:2
 - f. 2001:db8::7::2

g. 2000::1

6. Какой термин позволяет одной системе отправлять трафик на специальный IP-адрес, который получает несколько систем?

- a. netscope
- b. netmask
- c. subnet
- d. multicast**
- e. netaddr
- f. network

ПРОВЕРКА КОНФИГУРАЦИИ СЕТИ

ЗАДАЧИ

После завершения этого раздела вы сможете протестировать и проверить текущую конфигурацию сети с помощью утилит командной строки.

СБОР ИНФОРМАЦИИ О СЕТЕВОМ ИНТЕРФЕЙСЕ

Определение сетевых интерфейсов

Команда **ip link** выведет список всех сетевых интерфейсов, доступных в вашей системе:

```
[user@host ~]$ ip link show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN mode DEFAULT
    group default qlen 1000
        link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
2: ens3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP mode DEFAULT
    group default qlen 1000
        link/ether 52:54:00:00:00:0a brd ff:ff:ff:ff:ff:ff
3: ens4: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP mode DEFAULT
    group default qlen 1000
        link/ether 52:54:00:00:00:1e brd ff:ff:ff:ff:ff:ff
```

В предыдущем примере сервер имеет три сетевых интерфейса: **lo**, он является устройством обратной связи, подключенным к самому серверу, и два интерфейса **Ethernet**, **ens3** и **ens4**.

Чтобы правильно настроить каждый сетевой интерфейс, вам необходимо знать, какой из них подключен к какой сети. Во многих случаях вы будете знать MAC-адрес интерфейса, подключенного к каждой сети, либо потому, что он физически напечатан на карте или сервере, либо потому, что это виртуальная машина, и вы знаете, как она настроена. MAC-адрес устройства указан после **link/ether** для каждого интерфейса. Итак, вы знаете, что сетевая карта с **MAC-адресом 52:54:00:00:00:0a** является сетевым интерфейсом **ens3**.

Отображение IP-адресов

Используйте команду **ip** для просмотра информации об устройстве и адресе. Один сетевой интерфейс может иметь несколько адресов **IPv4** или **IPv6**.

```
[user@host ~]$ ip addr show ens3
```

```

2: ens3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP
    qlen 1000
        ② link/ether 52:54:00:00:00:0b brd ff:ff:ff:ff:ff:ff
        ③ inet 192.0.2.2/24 brd 192.0.2.255 scope global ens3
            valid_lft forever preferred_lft forever
        ④ inet6 2001:db8:0:1:5054:ff:fe00:b/64 scope global
            valid_lft forever preferred_lft forever
        ⑤ inet6 fe80::5054:ff:fe00:b/64 scope link
            valid_lft forever preferred_lft forever

```

1. Активный интерфейс - **UP**.
2. Стока **link/ether** указывает аппаратный (**MAC**) адрес устройства.
3. Стока **inet** показывает **IPv4**-адрес, длину префикса сети и область действия.
4. Стока **inet6** показывает **IPv6**-адрес, длину его сетевого префикса и область действия. Этот адрес имеет глобальный (*global*) характер и обычно используется.
5. Стока **inet6** показывает, что интерфейс имеет **IPv6**-адрес области связи, который может использоваться только для связи по локальному каналу **Ethernet**.

Отображение статистики производительности

Команду **ip** также можно использовать для отображения статистики производительности сети. Счетчики для каждого сетевого интерфейса могут использоваться для определения наличия сетевых проблем. Счетчики записывают статистику таких вещей, как количество принятых (**RX**) и переданных (**TX**) пакетов, ошибок пакетов и пакетов, которые были отброшены.

```

[user@host ~]$ ip -s link show ens3
2: ens3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP qlen
    1000
        link/ether 52:54:00:00:00:0a brd ff:ff:ff:ff:ff:ff

        RX: bytes   packets   errors   dropped overrun mcast
        269850      2931       0        0        0        0
        TX: bytes   packets   errors   dropped carrier collsns
        300556      3250       0        0        0        0

```

ПРОВЕРКА ПОДКЛЮЧЕНИЯ МЕЖДУ ХОСТАМИ

Команда **ping** используется для проверки возможности подключения. Команда продолжает выполняться до тех пор, пока не будет нажата комбинация клавиш **Ctrl+C**, если не указаны параметры, ограничивающие количество отправляемых пакетов.

```

[user@host ~]$ ping -c3 192.0.2.254
PING 192.0.2.1 (192.0.2.254) 56(84) bytes of data.

```

```
64 bytes from 192.0.2.254: icmp_seq=1 ttl=64 time=4.33 ms
64 bytes from 192.0.2.254: icmp_seq=2 ttl=64 time=3.48 ms
64 bytes from 192.0.2.254: icmp_seq=3 ttl=64 time=6.83 ms
```

```
--- 192.0.2.254 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2003ms
rtt min/avg/max/mdev = 3.485/4.885/6.837/1.424 ms
```

Команда **ping6** - это версия ping для **IPv6** в Red Hat Enterprise Linux. Он обменивается данными по **IPv6** и принимает адреса **IPv6**, но в остальном работает, как предыдущие команда **ping**.

```
[user@host ~]$ ping6 2001:db8:0:1::1
PING 2001:db8:0:1::1(2001:db8:0:1::1) 56 data bytes
64 bytes from 2001:db8:0:1::1: icmp_seq=1 ttl=64 time=18.4 ms
64 bytes from 2001:db8:0:1::1: icmp_seq=2 ttl=64 time=0.178 ms
64 bytes from 2001:db8:0:1::1: icmp_seq=3 ttl=64 time=0.180 ms
^C
```

```
--- 2001:db8:0:1::1 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2001ms
rtt min/avg/max/mdev = 0.178/6.272/18.458/8.616 ms
[user@host ~]$
```

Когда вы проверяете связь с локальными адресами и групповой многоадресной рассылкой всех узлов (**ff02::1**), используемый сетевой интерфейс должен быть явно указан с идентификатором зоны действия (например, **ff02::1%ens3**). Если это не указано, отображается ошибка подключения: недопустимый аргумент (*connect: Invalid argument*).

Пинг **ff02::1** может быть полезен для поиска других узлов **IPv6** в локальной сети.

```
[user@host ~]$ ping6 ff02::1%ens4
PING ff02::1%ens4(ff02::1) 56 data bytes
64 bytes from fe80::78cf:7fff:fed2:f97b: icmp_seq=1 ttl=64 time=22.7 ms
64 bytes from fe80::f482:dbff:fe25:6a9f: icmp_seq=1 ttl=64 time=30.1 ms (DUP!)
64 bytes from fe80::78cf:7fff:fed2:f97b: icmp_seq=2 ttl=64 time=0.183 ms
64 bytes from fe80::f482:dbff:fe25:6a9f: icmp_seq=2 ttl=64 time=0.231 ms (DUP!)
^C
--- ff02::1%ens4 ping statistics ---
2 packets transmitted, 2 received, +2 duplicates, 0% packet loss, time 1001ms
rtt min/avg/max/mdev = 0.183/13.320/30.158/13.374 ms
[user@host ~]$ ping6 -c 1 fe80::f482:dbff:fe25:6a9f%ens4
PING fe80::f482:dbff:fe25:6a9f%ens4(fe80::f482:dbff:fe25:6a9f) 56 data bytes
64 bytes from fe80::f482:dbff:fe25:6a9f: icmp_seq=1 ttl=64 time=22.9 ms

--- fe80::f482:dbff:fe25:6a9f%ens4 ping statistics ---
```

```
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 22.903/22.903/22.903/0.000 ms
```

Помните, что локальные ссылки адреса **IPv6** могут использоваться другими хостами в той же ссылке, как и обычные адреса.

```
[user@host ~]$ ssh fe80::f482:dbff:fe25:6a9f%ens4
user@fe80::f482:dbff:fe25:6a9f%ens4's password:
Last login: Thu Jun 5 15:20:10 2014 from host.example.com
[user@server ~]$
```

УСТРАНЕНИЕ НЕПОЛАДОК МАРШРУТИЗАЦИИ ПАКЕТОВ

Сетевая маршрутизация сложна, и иногда трафик ведет себя не так, как вы ожидали. Далее несколько полезных инструментов диагностики.

Отображение таблицы маршрутизации

Используйте команду **ip** с параметром **route**, чтобы отобразить информацию о маршрутизации.

```
[user@host ~]$ ip route
default via 192.0.2.254 dev ens3 proto static metric 1024
192.0.2.0/24 dev ens3 proto kernel scope link src 192.0.2.2
10.0.0.0/8 dev ens4 proto kernel scope link src 10.0.0.11
```

Приведённый пример показывает таблицу маршрутизации **IPv4**. Все пакеты, предназначенные для **сети 10.0.0.0/8**, отправляются прямо в пункт назначения через устройство **ens4**. Все пакеты, предназначенные для **сети 192.0.2.0/24**, отправляются прямо в пункт назначения через устройство **ens3**. Все остальные пакеты отправляются на маршрутизатор по умолчанию, расположенный по адресу **192.0.2.254**, а также через устройство **ens3**.

Добавьте параметр **-6**, чтобы отобразить таблицу маршрутизации **IPv6**:

```
[user@host ~]$ ip -6 route
unreachable ::/96 dev lo metric 1024 error -101
unreachable ::ffff:0.0.0.0/96 dev lo metric 1024 error -101
2001:db8:0:1::/64 dev ens3 proto kernel metric 256
unreachable 2002:a00::/24 dev lo metric 1024 error -101
unreachable 2002:7f00::/24 dev lo metric 1024 error -101
unreachable 2002:a9fe::/32 dev lo metric 1024 error -101
unreachable 2002:ac10::/28 dev lo metric 1024 error -101
unreachable 2002:c0a8::/32 dev lo metric 1024 error -101
unreachable 2002:e000::/19 dev lo metric 1024 error -101
```

```
unreachable 3ffe:ffff:/32 dev lo metric 1024 error -101
fe80::/64 dev ens3 proto kernel metric 256
default via 2001:db8:0:1::ffff dev ens3 proto static metric 1024
```

В этом примере игнорируйте недоступные маршруты, которые указывают на неиспользуемые сети. Остается три пути:

1. Сеть **2001:db8:0:1::/64**, использующая интерфейс **ens3** (который предположительно имеет адрес в этой сети).
2. Сеть **fe80::/64**, использующая интерфейс **ens3**, для локального адреса. В системе с несколькими интерфейсами существует маршрут к **fe80::/64** из каждого интерфейса для каждого локального адреса.
3. Маршрут по умолчанию ко всем сетям в Интернете IPv6 (сеть **::/0**), которые не имеют более конкретного маршрута в системе, через маршрутизатор по адресу **2001:db8:0:1::ffff**, доступный с помощью **ens3** устройство.

Отслеживание маршрута, пройденных трафиком

Чтобы отследить путь, по которому сетевой трафик достигает удаленного хоста через несколько маршрутизаторов, используйте команды **traceroute** или **tracepath**. Команды позволяют определить, связана ли проблема с одним из ваших маршрутизаторов или с промежуточным. Обе команды по умолчанию используют UDP-пакеты для отслеживания пути; однако многие сети блокируют трафик UDP и ICMP. Команда **traceroute** позволяет отслеживать путь с помощью пакетов UDP (по умолчанию), ICMP (-I) или TCP (-T). Однако обычно команда **traceroute** не устанавливается по умолчанию.

```
[user@host ~]$ tracepath access.redhat.com
...output omitted...
 4: 71-32-28-145.rcmt.qwest.net          48.853ms asymm  5
 5: dcp-brdr-04.inet.qwest.net           100.732ms asymm  7
 6: 206.111.0.153.ptr.us.xo.net          96.245ms asymm  7
 7: 207.88.14.162.ptr.us.xo.net          85.270ms asymm  8
 8: ae1d0.cir1.atlanta6-ga.us.xo.net     64.160ms asymm  7
 9: 216.156.108.98.ptr.us.xo.net         108.652ms
10: bu-ether13.atlngamq46w-bcr00.tbone.rr.com 107.286ms asymm 12
...output omitted...
```

Каждая строка в выходных данных **tracepath** представляет маршрутизатор или переход, через который проходит пакет между источником и конечным пунктом назначения. Дополнительная информация предоставляется по мере доступности, включая время приема-передачи (**round trip timing (RTT)**) и любые изменения размера максимального блока передачи (**maximum transmission unit (MTU)**). Индикация асимметрии означает, что трафик достиг этого маршрутизатора и вернулся от этого маршрутизатора с использованием других (асимметричных) маршрутов. Показанные маршрутизаторы используются для исходящего трафика, а не для обратного трафика.

Команды **tracepath6** и **traceroute -6** эквивалентны командам **tracepath** и **traceroute** для IPv6.

```
[user@host ~]$ tracepath6 2001:db8:0:2::451
1?: [LOCALHOST]          0.091ms pmtu 1500
1: 2001:db8:0:1::ba    0.214ms
2: 2001:db8:0:1::1    0.512ms
3: 2001:db8:0:2::451  0.559ms reached
Resume: pmtu 1500 hops 3 back 3
```

УСТРАНЕНИЯ НЕИСПРАВНОСТЕЙ СВЯЗАННЫЕ С ПОРТАМИ И СЕРВИСАМИ

Службы TCP используют сокеты в качестве конечных точек для связи и состоят из IP-адреса, протокола и номера порта. Службы обычно прослушивают стандартные порты, в то время как клиенты используют случайный доступный порт. Общеизвестные имена стандартных портов перечислены в файле **/etc/services**.

Команда **ss** используется для отображения статистики сокетов. Команда **ss** предназначена для замены старого инструмента **netstat**, входящего в пакет **net-tools**, который может быть более знаком некоторым системным администраторам, но не всегда устанавливается.

```
[user@host ~]$ ss -ta
State      Recv-Q Send-Q      Local Address:Port      Peer Address:Port
LISTEN      0      128          *:sunrpc           *:*
LISTEN      0      128          ① *:ssh             *:*
LISTEN      0      100          ② 127.0.0.1:smtp     *:*
LISTEN      0      128          *:36889            *:*
ESTAB       0      0            ③ 172.25.250.10:ssh   172.25.254.254:59392
LISTEN      0      128          :::sunrpc          :::*
LISTEN      0      128          ④ :::ssh            :::*
LISTEN      0      100          ⑤ :::1:smtp          :::*
LISTEN      0      128          :::34946            :::*
```

- Порт, используемый для SSH, прослушивает все адреса IPv4. Знак «*» используется для обозначения «всего» при обращении к IPv4-адресам или портам.
- Порт, используемый для SMTP, прослушивает интерфейс обратной связи IPv4 127.0.0.1.
- Установленное соединение SSH находится на интерфейсе 172.25.250.10 и исходит из системы с адресом 172.25.254.254.
- Порт, используемый для SSH, прослушивает все адреса IPv6. Синтаксис «::» используется для представления всех интерфейсов IPv6.
- Порт, используемый для SMTP, прослушивает интерфейс обратной петли ::1 IPv6.

Параметры для команд ss и netstat

ОПЦИИ	ОПИСАНИЕ
-n	Показывать числа вместо имен интерфейсов и портов.
-t	Показать сокеты TCP .
-u	Показать сокеты UDP .
-l	Показывать только прослушивающие сокеты.
-a	Показать все сокеты (прослушивающие и установленные).
-p	Покажите процесс с помощью сокетов.
-A inet	Отображение активных подключений (но не прослушивающих сокетов) для семейства адресов inet . То есть игнорировать локальные сокеты домена UNIX . Для ss отображаются подключения как IPv4 , так и IPv6 . Для netstat отображаются только соединения IPv4 . (netstat -A inet6 отображает соединения IPv6 , а netstat -46 одновременно отображает IPv4 и IPv6 .)



РЕКОМЕНДАЦИИ

Справочные страницы **man ip-link(8)**, **ip-address(8)**, **ip-route(8)**, **ip(8)**, **ping(8)**, **tracepath(8)**, **traceroute(8)**, **ss(8)**, и **netstat(8)**

Дополнительная информация доступна в главе о настройке сети в Руководстве по настройке сети Red Hat Enterprise Linux для Red Hat Enterprise Linux 8, которое можно найти по адресу

<https://access.redhat.com/documentation/>

УПРАЖНЕНИЯ С ПОШАГОВЫМИ ИНСТРУКЦИЯМИ

ПРОВЕРКА КОНФИГУРАЦИИ СЕТИ

В этом упражнении вы проверите сетевую конфигурацию одного из ваших серверов.

В РЕЗУЛЬТАТЕ

Определите текущие сетевые интерфейсы и основные сетевые адреса.

ПРЕЖДЕ ЧЕМ ВЫ НАЧНЕТЕ

Войдите на рабочую станцию **workstation** как пользователь **student**, используя в качестве пароля слово **student**.

На рабочей станции **workstation** выполните скрипт **lab net-validate start**. Команда запускает сценарий, который определяет, доступен ли **host, servera** в сети.

```
[student@workstation ~]$ lab net-validate start
```

- Используйте команду **ssh** для входа на сервер **servera**, как пользователь **student**. Системы настроены на использование ключей **SSH** для аутентификации, поэтому пароль для входа на сервер не требуется.

```
[student@workstation ~]$ ssh student@servera  
[student@servera ~]$
```

- Определите, какой сетевой интерфейс имеет **MAC**-адрес **52:54:00:00:fa:0a**, и проверьте имя интерфейса. В этих инструкциях предполагается, что это **ens3**. В оставшейся части этого упражнения замените это имя любым именем сетевого интерфейса, фактически отображаемым вашей системой.

```
[student@servera ~]$ ip link  
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default  
    qlen 1000  
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00  
2: ens3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group  
    default qlen 1000  
    link/ether 52:54:00:00:fa:0a brd ff:ff:ff:ff:ff:ff
```

- Отобразите текущий **IP**-адрес и сетевую маску для всех интерфейсов.

```
[student@servera ~]$ ip addr
```

```
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default
  qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
      inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
      inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: ens3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group
  default qlen 1000
    link/ether 52:54:00:00:fa:0a brd ff:ff:ff:ff:ff:ff
      inet 172.25.250.10/24 brd 172.25.250.255 scope global noprefixroute ens3
        valid_lft forever preferred_lft forever
      inet6 fe80::3059:5462:198:58b2/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
```

4. Отобразите статистику интерфейса **ens3**.

```
[student@servera ~]$ ip -s link show ens3
2: ens3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP mode
  DEFAULT group default qlen 1000
    link/ether 52:54:00:00:fa:0a brd ff:ff:ff:ff:ff:ff
    RX: bytes  packets errors dropped overrun mcast
      89014225   168251     0     154418     0       0
    TX: bytes  packets errors dropped carrier collsns
      608808     6090     0       0       0       0
```

5. Отобразите информацию о маршруте.

```
[student@servera ~]$ ip route
default via 172.25.250.254 dev ens3 proto static metric 100
172.25.250.0/24 dev ens3 proto kernel scope link src 172.25.250.10 metric 100
```

6. Убедитесь, что маршрутизатор доступен.

```
[student@servera ~]$ ping -c3 172.25.250.254
PING 172.25.250.254 (172.25.250.254) 56(84) bytes of data.
64 bytes from 172.25.250.254: icmp_seq=1 ttl=64 time=0.196 ms
64 bytes from 172.25.250.254: icmp_seq=2 ttl=64 time=0.436 ms
64 bytes from 172.25.250.254: icmp_seq=3 ttl=64 time=0.361 ms

--- 172.25.250.254 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 49ms
rtt min/avg/max/mdev = 0.196/0.331/0.436/0.100 ms
```

7. Покажите все переходы между локальной системой и **classroom.example.com**.

```
[student@servera ~]$ tracepath classroom.example.com
1?: [LOCALHOST]                                pmtu 1500
1: workstation.lab.example.com                0.270ms
1: workstation.lab.example.com                0.167ms
2: classroom.example.com                     0.473ms reached
Resume: pmtu 1500 hops 2 back 2
```

8. Отобразите прослушивающие сокеты **TCP** в локальной системе.

```
[student@servera ~]$ ss -lt
State      Recv-Q Send-Q     Local Address:Port      Peer Address:Port
LISTEN      0      128          0.0.0.0:sunrpc      0.0.0.0:*
LISTEN      0      128          0.0.0.0:ssh       0.0.0.0:*
LISTEN      0      128          [::]:sunrpc        [::]:*
LISTEN      0      128          [::]:ssh           [::]:*
```

9. Выход из сервера **servera**.

```
[student@servera ~]$ exit
logout
Connection to servera closed.
[student@workstation ~]$
```

Завершение

На рабочей станции **workstation**, запустите сценарий **lab net-validate finish** для проверки сети, для завершения упражнения.

```
[student@workstation ~]$ lab net-validate finish
```

На этом пошаговое упражнение завершено.

НАСТРОЙКА СЕТИ ИЗ КОМАНДНОЙ СТРОКИ

ЗАДАЧИ

После завершения этого раздела вы сможете управлять сетевыми настройками и устройствами с помощью команды **nmcli**.

ОПИСАНИЕ КОНЦЕПЦИИ NETWORKMANAGER

NetworkManager - это демон, который контролирует сетевые настройки и управляет ими. В дополнение к демону существует апплет области уведомлений **GNOME**, предоставляющий информацию о состоянии сети. Командная строка и графические инструменты взаимодействуют с **NetworkManager** и сохраняют файлы конфигурации в каталоге **/etc/sysconfig/network-scripts**.

- **Устройство**(*device*) - это сетевой интерфейс.
- **Подключение**(*connection*) - это набор параметров, которые можно настроить для устройства.
- Только одно соединение может быть активным для любого устройства одновременно. Может существовать несколько соединений для использования разными устройствами или для изменения конфигурации одного и того же устройства. Если вам нужно временно изменить параметры сети, вместо изменения конфигурации подключения вы можете изменить активное подключение для устройства. Например, устройство для беспроводного сетевого интерфейса на портативном компьютере может использовать разные соединения для беспроводной сети на рабочем месте и для беспроводной сети дома.
- Каждое соединение имеет **имя** (*name*) или **идентификатор** (**ID**), который его идентифицирует.
- Утилита **nmcli** используется для создания и редактирования файлов подключения из командной строки.

ПРОСМОТР СЕТЕВОЙ ИНФОРМАЦИИ

Команда **nmcli dev status** отображает состояние всех сетевых устройств:

```
[user@host ~]$ nmcli dev status
DEVICE  TYPE      STATE       CONNECTION
eno1    ethernet  connected   eno1
ens3    ethernet  connected   static-ens3
eno2    ethernet  disconnected --
lo     loopback  unmanaged   --
```

Команда **nmcli con show** отображает список всех подключений. Чтобы вывести список только активных подключений, добавьте параметр **--active**.

```
[user@host ~]$ nmcli con show
NAME      UUID                                  TYPE      DEVICE
eno2      ff9f7d69-db83-4fed-9f32-939f8b5f81cd 802-3-ethernet --
static-ens3 72ca57a2-f780-40da-b146-99f71c431e2b 802-3-ethernet ens3
eno1      87b53c56-1f5d-4a29-a869-8a7bdaf56dfa 802-3-ethernet eno1
[user@host ~]$ nmcli con show --active
NAME      UUID                                  TYPE      DEVICE
static-ens3 72ca57a2-f780-40da-b146-99f71c431e2b 802-3-ethernet ens3
eno1      87b53c56-1f5d-4a29-a869-8a7bdaf56dfa 802-3-ethernet eno1
```

ДОБАВЛЕНИЕ СЕТЕВОГО СОЕДИНЕНИЯ

Команда **nmcli con add** используется для добавления новых сетевых подключений. В следующем примере команды **nmcli con add** предполагают, что имя добавляемого сетевого подключения еще не используется.

Следующая команда добавляет новое соединение с именем **eno2** для интерфейса **eno2**, которое получает сетевую информацию **IPv4** с помощью **DHCP** и автоматически подключается при запуске. Он также получает сетевые настройки **IPv6**, прослушивая объявления маршрутизатора на локальном канале. Имя файла конфигурации основано на значении параметра **con-name**, **eno2**, и сохраняется в файле **/etc/sysconfig/network-scripts/ifcfg-eno2**.

```
[root@host ~]# nmcli con add con-name eno2 type ethernet ifname eno2
```

В следующем примере создается соединение **eno2** для устройства **eno2** со статическим **IPv4**-адресом с использованием **IPv4**-адреса и сетевого префикса **192.168.0.5/24** и шлюза по умолчанию **192.168.0.254**, но по-прежнему автоматически подключается при запуске и сохраняет свою конфигурацию в том же файле. Из-за ограничений по размеру экрана завершите первую строку оболочки знаком \ переход и завершите команду на следующей строке.

```
[root@host ~]# nmcli con add con-name eno2 type ethernet ifname eno2 \
> ip4 192.168.0.5/24 gw4 192.168.0.254
```

В этом примере создается соединение **eno2** для устройства **eno2** со статическими адресами **IPv6** и **IPv4** с использованием адреса **IPv6** и сетевого префикса **2001:db8:0:1::c000:207/64** и шлюза **IPv6** по умолчанию **2001:db8:0:1::1**, а также **IPv4**-адрес и сетевой префикс **192.0.2.7/24** и шлюз **IPv4** по умолчанию **192.0.2.1**, но по-прежнему автоматически подключается при запуске и сохраняет свою конфигурацию в **/etc/sysconfig/network-scripts/ifcfg-eno2**. Из-за ограничений по размеру экрана завершите первую строку оболочки знаком \ переход и завершите команду на следующей строке.

```
[root@host ~]# nmcli con add con-name eno2 type ethernet ifname eno2 \
> ip6 2001:db8:0:1::c000:207/64 gw6 2001:db8:0:1::1 ip4 192.0.2.7/24 gw4 192.0.2.1
```

УПРАВЛЕНИЕ СЕТЕВЫМИ СОЕДИНЕНИЯМИ

Команда **nmcli con up name** активирует имя соединения на сетевом интерфейсе, к которому оно привязано. Обратите внимание, что команда принимает имя соединения, а не имя сетевого интерфейса. Помните, что команда **nmcli con show** отображает имена всех доступных подключений.

```
[root@host ~]# nmcli con up static-ens3
```

Команда **nmcli dev disconnect device** отключает устройство сетевого интерфейса и отключает его. Эта команда может быть сокращена до **nmcli dev dis device**:

```
[root@host ~]# nmcli dev dis ens3
```



ВАЖНО

Используйте команду **nmcli dev dis device**, чтобы отключить сетевой интерфейс.

Команда **nmcli con down name** обычно не лучший способ деактивировать сетевой интерфейс, потому что она обрывает соединение. Однако по умолчанию большинство проводных системных подключений настроено с автоматическим подключением. Соединение активируется, как только станет доступен его сетевой интерфейс. Поскольку сетевой интерфейс соединения по-прежнему доступен, **nmcli con down name** отключает интерфейс, но затем **NetworkManager** немедленно его снова вызывает, если соединение не будет полностью отключено от интерфейса.

ИЗМЕНЕНИЕ ПАРАМЕТРОВ СЕТЕВОГО СОЕДИНЕНИЯ

Подключения **NetworkManager** имеют два типа настроек. Существуют статические свойства подключения, настраиваемые администратором и хранящиеся в файлах конфигурации в **/etc/sysconfig/network-scripts/ifcfg-***. Также могут быть активные данные соединения, которые соединение получает от **DHCP-сервера** и которые не хранятся постоянно.

Чтобы вывести список текущих настроек соединения, запустите команду **nmcli con show name**, где **name** - это имя соединения. Настройки в нижнем регистре - это статические свойства, которые администратор может изменять. Настройки, выделенные заглавными буквами, являются активными настройками во временном использовании для данного экземпляра соединения.

```
[root@host ~]# nmcli con show static-ens3
```

```

connection.id:                      static-ens3
connection.uuid:                    87b53c56-1f5d-4a29-a869-8a7bdaf56dfa
connection.interface-name:          --
connection.type:                   802-3-ethernet
connection.autoconnect:            yes
connection.timestamp:              1401803453
connection.read-only:              no
connection.permissions:            --
connection.zone:                  --
connection.master:                 --
connection.slave-type:             --
connection.secondaries:            --
connection.gateway-ping-timeout:   0
802-3-ethernet.port:               --
802-3-ethernet.speed:              0
802-3-ethernet.duplex:             --
802-3-ethernet.auto-negotiate:    yes
802-3-ethernet.mac-address:        CA:9D:E9:2A:CE:F0
802-3-ethernet.cloned-mac-address: --
802-3-ethernet.mac-address-blacklist: --
802-3-ethernet.mtu:                auto
802-3-ethernet.s390-subchannels:   --
802-3-ethernet.s390-nettype:       --
802-3-ethernet.s390-options:      --
ipv4.method:                       manual
ipv4.dns:                          192.168.0.254
ipv4.dns-search:                  example.com
ipv4.addresses:                   { ip = 192.168.0.2/24, gw =
                                   192.168.0.254 }
ipv4.routes:                       --
ipv4.ignore-auto-routes:           no
ipv4.ignore-auto-dns:              no
ipv4.dhcp-client-id:              --
ipv4.dhcp-send-hostname:           yes
ipv4.dhcp-hostname:                --
ipv4.never-default:                no
ipv4.may-fail:                     yes
ipv6.method:                       manual
ipv6.dns:                          2001:4860:4860::8888
ipv6.dns-search:                  example.com
ipv6.addresses:                   { ip = 2001:db8:0:1::7/64, gw =
                                   2001:db8:0:1::1 }
ipv6.routes:                       --
ipv6.ignore-auto-routes:           no
ipv6.ignore-auto-dns:              no
ipv6.never-default:                no
ipv6.may-fail:                     yes
ipv6.ip6-privacy:                 -1 (unknown)
ipv6.dhcp-hostname:                --
...output omitted...

```

Команда **nmcli con mod name** используется для изменения настроек соединения. Эти изменения также сохраняются в файле **/etc/sysconfig/network-scripts/ifcfg-name** для подключения. Доступные настройки задокументированы на странице руководства **nm-settings(5)**.

Чтобы установить **IPv4-адрес 192.0.2.2/24** и шлюз по умолчанию **192.0.2.254** для подключения **static-ens3**:

```
[root@host ~]# nmcli con mod static-ens3 ipv4.addresses "192.0.2.2/24 192.0.2.254"
```

Чтобы установить адрес IPv6 на **2001:db8:0:1::a00:1/64** и шлюз по умолчанию на **2001:db8:0:1::1** для подключения **static-ens3**:

```
[root@host ~]# nmcli con mod static-ens3 ipv6.address "2001:db8:0:1::a00:1/64  
2001:db8:0:1::1"
```



ВАЖНО

Если соединение, которое получает информацию о **IPv4** от сервера **DHCPv4**, изменяется для получения ее только из файлов статической конфигурации, параметр **ipv4.method** также должен быть изменен с автоматического на ручной.

Аналогичным образом, если соединение, которое получает информацию об **IPv6** с помощью **SLAAC** или **DHCPv6-сервера**, изменяется для получения ее только из статических файлов конфигурации, параметр **ipv6.method** также должен быть изменен с **auto** или **dhcp** на **manual**.

В противном случае соединение может зависнуть или не завершиться успешно при его активации, или оно может получить адрес **IPv4** от **DHCP** или адрес **IPv6** от **DHCPv6** или **SLAAC** в дополнение к статическому адресу.

Ряд настроек может иметь несколько значений. Определенное значение можно добавить в список или удалить из списка для параметра, добавив символ + или - в начало имени параметра.

УДАЛЕНИЕ СЕТЕВОГО СОЕДИНЕНИЯ

Команда **nmcli con del name** удаляет соединение с именем **name** из системы, отключает его от устройства и удаляет файл **/etc/sysconfig/network-scripts/ifcfg-name**.

```
[root@host ~]# nmcli con del static-ens3
```

КТО МОЖЕТ ИЗМЕНЯТЬ НАСТРОЙКИ СЕТИ?

Пользователь **root** может вносить любые необходимые изменения в конфигурацию сети с помощью **nmcli**.

Однако обычные пользователи, которые вошли в систему с локальной консолью, также могут вносить в систему множество изменений конфигурации сети. Они должны войти в систему с клавиатуры системы либо в текстовую виртуальную консоль, либо в среду графического рабочего стола, чтобы получить этот контроль. Логика этого заключается в том, что, если кто-то физически присутствует у консоли компьютера, он, вероятно, используется в качестве рабочей станции или ноутбука, и им может потребоваться настроить, активировать и деактивировать беспроводные или проводные сетевые интерфейсы по своему желанию. Напротив, если система является сервером в центре обработки данных, обычно единственными пользователями, входящими в систему локально на самой машине, должны быть администраторы.

Обычные пользователи, которые входят в систему с помощью **ssh**, не имеют доступа для изменения сетевых разрешений, не став **root**.

Вы можете использовать команду **nmcli gen permissions**, чтобы узнать, какие у вас текущие разрешения.

РЕЗЮМЕ КОМАНД

В следующей таблице представлен список основных команд **nmcli**, обсуждаемых в этом разделе.

КОМАНДА	ЦЕЛЬ
nmcli dev status	Показать статус NetworkManager всех сетевых интерфейсов.
nmcli con show	Список всех подключений.
nmcli con show name	Список текущих настроек для подключения <i>name</i> .
nmcli con add con-name name	Добавьте новое соединение с именем <i>name</i> .
nmcli con mod name	Измените подключения <i>name</i> .
nmcli con reload	Перезагрузите файлы конфигурации (полезно после того, как они были отредактированы вручную).
nmcli con up name	Активируйте подключение <i>name</i> .
nmcli dev dis dev	Деактивировать и отключить текущее соединение сетевого интерфейса <i>dev</i> .
nmcli con del name	Удалите подключение <i>name</i> и его файл конфигурации.



РЕКОМЕНДАЦИИ

Справочные страницы **man NetworkManager(8)**, **nmcli(1)**, **nmcli-examples(5)**, **nm-settings(5)**, **hostnamectl(1)**, **resolv.conf(5)**, **hostname(5)**, **ip(8)**, и **ip-address(8)**

УПРАЖНЕНИЯ С ПОШАГОВЫМИ ИНСТРУКЦИЯМИ

НАСТРОЙКА СЕТИ ИЗ КОМАНДНОЙ СТРОКИ

В этом упражнении вы настроите параметры сети с помощью **nmcli**.

В РЕЗУЛЬТАТЕ

Вы должны иметь возможность преобразовать систему из DHCP в статическую конфигурацию.

ПРЕЖДЕ ЧЕМ ВЫ НАЧНЕТЕ

Войдите на рабочую станцию **workstation** как пользователь **student**, используя в качестве пароля слово **student**.

На рабочей станции **workstation** выполните скрипт **lab net-configure start**. Команда запускает сценарий, который определяет, доступен ли **host**, сервер **servera**, в сети.

```
[student@workstation ~]$ lab net-configure start
```

1. Используйте команду **ssh** для входа на сервер **servera** как пользователь **student**. Системы настроены на использование ключей SSH для аутентификации, поэтому пароль для входа на сервер не требуется.

```
[student@workstation ~]$ ssh student@servera  
[student@servera ~]$
```

2. Эти инструкции написаны исходя из предположения, что ваш сетевой интерфейс с MAC-адресом **52:54:00:00:fa:0a** называется **ens3**. Используйте команду **ip link**, чтобы проверить это, и, если ваша система имеет другое имя интерфейса, используйте его для остальной части упражнения.

```
[student@servera ~]$ ip link  
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default  
    qlen 1000  
        link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00  
2: ens3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group  
    default qlen 1000  
        link/ether 52:54:00:00:fa:0a brd ff:ff:ff:ff:ff:ff
```

3. Просмотрите настройки сети с помощью **nmcli**.

3.1. Показать все соединения.

```
[student@servera ~]$ nmcli con show
NAME           UUID                                  TYPE      DEVICE
ens3           03da038a-3257-4722-a478-53055cc90128  ethernet  ens3
Wired connection 1  91d515af-5367-384b-a012-6dbd2e390f66  ethernet  --
```

3.2. Отображать только активное соединение.

Имя вашего сетевого интерфейса должно появиться в разделе УСТРОЙСТВО, а имя подключения, активного для этого устройства, указано в той же строке в разделе ИМЯ. В этом упражнении предполагается, что активное соединение - ens3, соответствующее имени сетевого интерфейса.

Если имя активного соединения отличается, используйте его вместо Ens3 для остальной части этого упражнения.

```
[student@servera ~]$ nmcli con show --active
NAME           UUID                                  TYPE      DEVICE
ens3           03da038a-3257-4722-a478-53055cc90128  ethernet  ens3
```

3.3. Отобразите все параметры конфигурации для активного соединения.

```
[student@servera ~]$ nmcli con show "ens3"
connection.id:          ens3
connection.uuid:        03da038a-3257-4722-a478-53055cc90128
connection.stable-id:   --
connection.type:        802-3-ethernet
connection.interface-name:  ens3
connection.autoconnect: yes
...output omitted...
ipv4.method:            manual
ipv4.dns:               172.25.250.254
ipv4.dns-search:        lab.example.com,example.com
ipv4.dns-options:       ""
ipv4.dns-priority:      0
ipv4.addresses:         172.25.250.10/24
ipv4.gateway:           172.25.250.254
...output omitted...
GENERAL.NAME:           ens3
GENERAL.UUID:           03da038a-3257-4722-a478-53055cc90128
GENERAL.DEVICES:        ens3
GENERAL.STATE:          activated
GENERAL.DEFAULT:        yes
GENERAL.DEFAULT6:       no
GENERAL.SPEC-OBJECT:    --
GENERAL.VPN:            no
GENERAL.DBUS-PATH:      /org/freedesktop/NetworkManager/ActiveConnection/2
GENERAL.CON-PATH:        /org/freedesktop/NetworkManager/Settings/2
```

```
GENERAL.ZONE:          --
GENERAL.MASTER-PATH:   --
IP4.ADDRESS[1]:        172.25.250.10/24
IP4.GATEWAY:           172.25.250.254
IP4.ROUTE[1]:          dst = 172.25.250.0/24, nh = 0.0.0.0, mt = 100
IP4.ROUTE[2]:          dst = 0.0.0.0/0, nh = 172.25.250.254, mt = 100
IP4.DNS[1]:             172.25.250.254
IP6.ADDRESS[1]:         fe80::3059:5462:198:58b2/64
IP6.GATEWAY:            --
IP6.ROUTE[1]:           dst = fe80::/64, nh = ::, mt = 100
IP6.ROUTE[2]:           dst = ff00::/8, nh = ::, mt = 256, table=255
```

Нажмите **Q**, чтобы выйти из команды.

3.4. Показать статус устройства.

```
[student@servera ~]$ nmcli dev status
DEVICE  TYPE      STATE      CONNECTION
ens3    ethernet  connected  ens3
lo     loopback  unmanaged  --
```

3.5. Отобразите настройки для устройства **ens3**.

```
[student@servera ~]$ nmcli dev show ens3
GENERAL.DEVICE:          ens3
GENERAL.TYPE:            ethernet
GENERAL.HWADDR:          52:54:00:00:FA:0A
GENERAL.MTU:              1500
GENERAL.STATE:            100 (connected)
GENERAL.CONNECTION:       ens3
GENERAL.CON-PATH:         /org/freedesktop/NetworkManager/ActiveConnection/2
WIRED-PROPERTIES.CARRIER: on
IP4.ADDRESS[1]:           172.25.250.10/24
IP4.GATEWAY:              172.25.250.254
IP4.ROUTE[1]:              dst = 172.25.250.0/24, nh = 0.0.0.0, mt = 100
IP4.ROUTE[2]:              dst = 0.0.0.0/0, nh = 172.25.250.254, mt = 100
IP4.DNS[1]:                172.25.250.254
IP6.ADDRESS[1]:            fe80::3059:5462:198:58b2/64
IP6.GATEWAY:               --
IP6.ROUTE[1]:              dst = fe80::/64, nh = ::, mt = 100
IP6.ROUTE[2]:              dst = ff00::/8, nh = ::, mt = 256, table=255
```

4. Создайте статическое соединение с тем же адресом IPv4, префиксом сети и шлюзом по умолчанию. Назовите новое соединение **static-ens3**.



ПРЕДУПРЕЖДЕНИЕ

Поскольку доступ к вашему компьютеру предоставляется через основное сетевое соединение, установка неверных значений во время настройки сети может сделать вашу машину недоступной. В этом случае используйте кнопку «Reset», расположенную над тем, что раньше было графическим дисплеем вашего устройства, и повторите попытку.

```
[student@servera ~]$ sudo nmcli con add con-name "static-ens3" ifname ens3 \
> type ethernet ip4 172.25.250.10/24 gw4 172.25.250.254
Connection 'static-ens3' (15aa3901-555d-40cb-94c6-cea6f9151634) successfully
added.
```

- Измените новое соединение, чтобы добавить настройку **DNS**.

```
[student@servera ~]$ sudo nmcli con mod "static-ens3" ipv4.dns 172.25.250.254
```

- Отобразите и активируйте новое соединение.

6.1. Просмотреть все подключения.

```
[student@servera ~]$ nmcli con show
NAME           UUID                                  TYPE      DEVICE
ens3           88a449d0-694c-4e5c-a672-01b0826a966e  ethernet  ens3
static-ens3    15aa3901-555d-40cb-94c6-cea6f9151634  ethernet  --
Wired connection 1 9b531978-64e7-3d2c-a574-38d7323c78dc  ethernet  --
```

6.2. Просмотрите активное соединение.

```
[student@servera ~]$ nmcli con show --active
ens3   88a449d0-694c-4e5c-a672-01b0826a966e  ethernet  ens3
```

6.3. Активируйте новое соединение static-ens3.

```
[student@servera ~]$ sudo nmcli con up "static-ens3"
Connection successfully activated (D-Bus active path: /org/freedesktop/
NetworkManager/ActiveConnection/3)
```

6.4. Проверьте новое активное соединение.

```
[student@servera ~]$ nmcli con show --active
```

```
NAME      UUID           TYPE      DEVICE
static-ens3 15aa3901-555d-40cb-94c6-cea6f9151634  ethernet  ens3
```

7. Проверьте возможность подключения, используя новые сетевые адреса.

7.1. Проверьте IP-адрес.

```
[student@servera ~]$ ip addr show ens3
2: ens3: >BROADCAST,MULTICAST,UP,LOWER_UP< mtu 1500 qdisc fq_codel state UP group
default qlen 1000
    link/ether 52:54:00:00:fa:0a brd ff:ff:ff:ff:ff:ff
    inet 172.25.250.10/24 brd 172.25.250.255 scope global noprefixroute ens3
        valid_lft forever preferred_lft forever
    inet6 fe80::6556:cdd9:ce15:1484/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
```

7.2. Проверьте шлюз по умолчанию.

```
[student@servera ~]$ ip route
default via 172.25.250.254 dev ens3 proto static metric 100
172.25.250.0/24 dev ens3 proto kernel scope link src 172.25.250.10 metric 100
```

7.3. Пингуйте DNS-адрес.

```
[student@servera ~]$ ping -c3 172.25.250.254
PING 172.25.250.254 (172.25.250.254) 56(84) bytes of data.
64 bytes from 172.25.250.254: icmp_seq=1 ttl=64 time=0.225 ms
64 bytes from 172.25.250.254: icmp_seq=2 ttl=64 time=0.314 ms
64 bytes from 172.25.250.254: icmp_seq=3 ttl=64 time=0.472 ms

--- 172.25.250.254 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 46ms
rtt min/avg/max/mdev = 0.225/0.337/0.472/0.102 ms
```

8. Настройте исходное соединение так, чтобы оно не запускалось при загрузке, и убедитесь, что статическое соединение используется при перезагрузке системы.

8.1. Отключите исходное соединение от автозапуска при загрузке.

```
[student@servera ~]$ sudo nmcli con mod "ens3" connection.autoconnect no
```

8.2. Перезагрузите систему.

```
[student@servera ~]$ sudo reboot  
Connection to servera closed by remote host.  
Connection to servera closed.  
[student@workstation ~]$
```

8.3. Просмотрите активное соединение.

```
[student@workstation ~]$ ssh student@servera  
[student@servera ~]$ nmcli con show --active  
NAME           UUID                                  TYPE      DEVICE  
static-ens3    15aa3901-555d-40cb-94c6-cea6f9151634  ethernet  ens3
```

8.4. Выйти с сервера.

```
[student@servera ~]$ exit  
logout  
Connection to servera closed.  
[student@workstation ~]$
```

Завершение

На рабочей станции запустите сценарий **lab net-configure finish**, чтобы завершить упражнение.

```
[student@workstation ~]$ lab net-configure finish
```

На этом пошаговое упражнение завершено.

РЕДАКТИРОВАНИЕ ФАЙЛОВ КОНФИГУРАЦИИ СЕТИ

ЗАДАЧИ

После завершения этого раздела вы сможете изменить конфигурацию сети, отредактировав файлы конфигурации.

ОПИСАНИЕ ФАЙЛОВ КОНФИГУРАЦИИ ДЛЯ ПОДКЛЮЧЕНИЯ

По умолчанию изменения, сделанные с помощью **nmcli con mod name**, автоматически сохраняются в **/etc/sysconfig/network-scripts/ifcfg-name**. Этот файл также можно редактировать вручную с помощью текстового редактора. После этого запустите **nmcli con reload**, чтобы **NetworkManager** прочитал изменения конфигурации.

По причинам обратной совместимости директивы, сохраненные в этом файле, имеют другие имена и синтаксис, чем имена **nm-settings(5)**. В следующей таблице некоторые имена ключевых параметров сопоставляются с директивами **ifcfg-***.

Сравнение nm-settings и директив ifcfg-*

NMCLI CON MOD	IFCFG-* FILE	ЭФФЕКТ
ipv4.method manual	BOOTPROTO=none	Адреса IPv4 настроены статически.
ipv4.method auto	BOOTPROTO=dhcp	Ищет параметры конфигурации с сервера DHCPv4. Если также установлены статические адреса, они не будут отображаться, пока мы не получим информацию от DHCPv4.
ipv4.addresses "192.0.2.1/24 192.0.2.254"	IPADDR0=192.0.2.1 PREFIX0=24 GATEWAY0=192.0.2.254	Устанавливает статический IPv4-адрес , сетевой префикс и шлюз по умолчанию. Если для подключения установлено более одного, то вместо 0 директивы ifcfg-* заканчиваются цифрами 1, 2, 3 и так далее.
ipv4.dns 8.8.8.8	DNS0=8.8.8.8	Измените /etc/resolv.conf , чтобы использовать этот сервер имен (nameserver).
ipv4.dns-search example.com	DOMAIN=example.com	Измените /etc/resolv.conf , чтобы использовать этот домен в директиве search .
ipv4.ignore-auto-dns true	PEERDNS=no	Игнорировать информацию DNS-сервера от DHCP-сервера .

ipv6.method manual	IPV6_AUTOCONF=no	Адреса IPv6 настроены статически.
ipv6.method auto	IPV6_AUTOCONF=yes	Настраивает параметры сети с помощью SLAAC из объявлений маршрутизатора.
ipv6.method dhcp	IPV6_AUTOCONF=no DHCPV6C=yes	Настраивает параметры сети с помощью DHCPv6 , но не SLAAC .
ipv6.addresses "2001:db8::a/64 2001:db8::1"	IPV6ADDR=2001:db8::a/64 IPV6_DEFAULTGW=2001:db8::	Устанавливает статический IPv6 -адрес, сетевой префикс и шлюз по умолчанию. Если для подключения установлено более одного адреса, IPV6_SECONDARIES берет список с ограничениями в двойных кавычках определения адресов/префиксов.
ipv6.dns ...	DNS0= ...	Измените /etc/resolv.conf , чтобы использовать этот сервер имен. Точно так же, как IPv4 .
ipv6.dns-search example.com	DOMAIN=example.com	Измените /etc/resolv.conf , чтобы использовать этот домен в директиве search . Точно так же, как IPv4 .
ipv6.ignore-auto-dns true	IPV6_PEERDNS=no	Игнорировать информацию DNS -сервера от DHCP -сервера.
connection.autoconnect yes	ONBOOT=yes	Автоматически активировать это соединение при загрузке.
connection.id ens3	NAME=ens3	Имя этого соединения.
connection.interface-name ens3	DEVICE=ens3	Подключение привязано к сетевому интерфейсу с этим именем.
802-3-ethernet.mac-address ...	HWADDR= ...	Соединение привязано к сетевому интерфейсу с этим MAC -адресом.

ИЗМЕНЕНИЕ КОНФИГУРАЦИИ СЕТИ

Также возможно настроить сеть, напрямую отредактировав файлы конфигурации подключения. Файлы конфигурации подключения управляют программными интерфейсами для отдельных сетевых устройств. Эти файлы обычно называются **/etc/sysconfig/network-scripts/ifcfg-name**, где **name** относится к имени устройства или соединения, которым управляет файл конфигурации. Ниже приведены стандартные переменные, содержащиеся в файле, используемом для статической или динамической конфигурации **IPv4**.

Параметры конфигурации IPv4 для файла ifcfg

СТАТИЧЕСКАЯ	ДИНАМИЧЕСКАЯ	ТАКЖЕ
BOOTPROTO=none IPADDR0=172.25.250.10 PREFIX0=24 GATEWAY0=172.25.250.254 DEFROUTE=yes DNS1=172.25.254.254	BOOTPROTO=dhcp	DEVICE=ens3 NAME="static-ens3" ONBOOT=yes UUID=f3e8(...)ad3e USERCTL=yes

В статических настройках переменные для IP-адреса, префикса и шлюза имеют номер в конце. Это позволяет назначать интерфейсу несколько наборов значений. Переменная DNS также имеет номер, используемый для указания порядка поиска при указании нескольких серверов.

После изменения файлов конфигурации запустите **nmcli con reload**, чтобы **NetworkManager** прочитал изменения конфигурации. Интерфейс по-прежнему необходимо перезапустить, чтобы изменения вступили в силу.

```
[root@host ~]# nmcli con reload
[root@host ~]# nmcli con down "static-ens3"
[root@host ~]# nmcli con up "static-ens3"
```



РЕКОМЕНДАЦИИ

Справочные страницы **man nmcli(1)**

Дополнительная информация может быть доступна в главе о настройке сети в Руководстве по работе с сетью Red Hat Enterprise Linux для Red Hat Enterprise Linux 8, которое можно найти по адресу

<https://access.redhat.com/documentation/>

УПРАЖНЕНИЯ С ПОШАГОВЫМИ ИНСТРУКЦИЯМИ

РЕДАКТИРОВАНИЕ ФАЙЛОВ КОНФИГУРАЦИИ СЕТИ

В этом упражнении вы вручную измените файлы конфигурации сети и убедитесь, что новые настройки вступят в силу.

В РЕЗУЛЬТАТЕ

У вас должна быть возможность добавить дополнительный сетевой адрес в каждую систему.

ПРЕЖДЕ ЧЕМ ВЫ НАЧНЕТЕ

Войдите на рабочую станцию **workstation** как пользователь **student**, используя в качестве пароля слово **student**.

На рабочей станции **workstation** выполните скрипт **lab net-edit start**. Команда запускает сценарий, который определяет, достижимы ли, сервера **servera** и **serverb**, в сети.

```
[student@workstation ~]$ lab net-edit start
```

1. Используйте команду **ssh** для входа на сервер **servera**, как пользователь **student**. Системы настроены на использование ключей SSH для аутентификации, поэтому пароль для входа на сервер не требуется.

```
[student@workstation ~]$ ssh student@servera  
...output omitted...  
[student@servera ~]$
```

2. Используйте команду **sudo -i**, чтобы переключиться на пользователя **root**. Если будет предложено, используйте в качестве пароля слово **student**.

```
[student@servera ~]$ sudo -i  
[sudo] password for student:  
[root@servera ~]#
```

3. В этом упражнении предполагается, что сетевое устройство с MAC-адресом **52:54:00:00:fa:0a** и называется **ens3**, его активное соединение также называется **ens3** (и, следовательно, управляет файлом **/etc/sysconfig/network-scripts/ifcfg-ens3**). Если вы выполняли предыдущие упражнения в этой главе, и это верно для вашей системы, то это должно быть верно и для этого упражнения.

Прежде чем продолжить, убедитесь, что это так, или определите имена для вашей системы.

```
[student@servera ~]$ ip link
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default
    qlen 1000
        link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
2: ens3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default
    qlen 1000
        link/ether 52:54:00:00:fa:0a brd ff:ff:ff:ff:ff:ff
[student@servera ~]$ nmcli con show --active
NAME           UUID                                  TYPE      DEVICE
ens3          03da038a-3257-4722-a478-53055cc90128  ethernet  ens3
[student@servera ~]$ ls /etc/sysconfig/network-scripts/ifcfg-ens3
/etc/sysconfig/network-scripts/ifcfg-ens3
```

4. Как пользователь **root** отредактируйте файл **/etc/sysconfig/network-scripts/ifcfg-ens3** на сервере **servera**, чтобы добавить дополнительный адрес **10.0.1.1/24**.

4.1. Добавьте запись в файл, чтобы указать адрес **IPv4**.

```
[root@servera ~]# echo "IPADDR1=10.0.1.1" >> \
> /etc/sysconfig/network-scripts/ifcfg-ens3
```

4.2. Добавьте в файл запись для указания префикса сети.

```
[root@servera ~]# echo "PREFIX1=24" >> /etc/sysconfig/network-scripts/ifcfg-ens3
```

5. Активируйте новый адрес.

5.1. Перезагрузите изменения конфигурации.

```
[root@servera ~]# nmcli con reload
```

5.2. Перезапустите соединение с новыми настройками.

```
[root@servera ~]# nmcli con up "ens3"
Connection successfully activated (D-Bus active path: /org/freedesktop/
NetworkManager/ActiveConnection/3)
```

5.3. Проверьте новый IP-адрес.

```
[root@servera ~]# ip addr show ens3
```

```
2: ens3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 52:54:00:00:fa:0a brd ff:ff:ff:ff:ff:ff
    inet 172.25.250.10/24 brd 172.25.250.255 scope global noprefixroute ens3
        valid_lft forever preferred_lft forever
    inet 10.0.1.1/24 brd 10.0.1.255 scope global noprefixroute ens3
        valid_lft forever preferred_lft forever
    inet6 fe80::4bf3:e1d9:3076:f8d7/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
```

- 5.4. Выдите из сервера **servera**, чтобы вернуться на рабочую станцию **workstation** в качестве пользователя **student**.

```
[root@servera ~]# exit
logout
[student@servera ~]$ exit
logout
Connection to servera closed.
[student@workstation ~]$
```

6. Как пользователь **root** на сервере **serverb** отредактируйте файл **/etc/sysconfig/network-scripts/ifcfg-ens3**, чтобы добавить дополнительный адрес **10.0.1.2/24**, затем загрузите новую конфигурацию.

- 6.1. С рабочей станции **workstation** используйте команду **ssh** для входа на **serverb** как пользователь **student**.

```
[student@workstation ~]$ ssh student@serverb
...output omitted...
[student@serverb ~]$
```

- 6.2. Используйте команду **sudo -i**, чтобы переключиться на пользователя **root**. Если будет предложено, используйте слово **student** в качестве пароля.

```
[student@serverb ~]$ sudo -i
[sudo] password for student: student
[root@serverb ~]#
```

- 6.3. Измените файл **ifcfg-ens3**, чтобы добавить второй адрес **IPv4** и сетевой префикс.

```
[root@serverb ~]$ echo "IPADDR2=10.0.1.2" >> \
> /etc/sysconfig/network-scripts/ifcfg-ens3
```

```
[root@serverb ~]# echo "PREFIX2=24" >> /etc/sysconfig/network-scripts/ifcfg-ens3
```

6.4. Перезагрузите изменения конфигурации.

```
[root@serverb ~]# nmcli con reload
```

6.5. Установите соединение с новыми настройками.

```
[root@serverb ~]# nmcli con up "ens3"
```

```
Connection successfully activated (D-Bus active path: /org/freedesktop/NetworkManager/ActiveConnection/4)
```

6.6. Проверьте новый IP-адрес.

```
[root@serverb ~]# ip addr show ens3
```

```
2: ens3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 52:54:00:00:fa:0b brd ff:ff:ff:ff:ff:ff
        inet 172.25.250.11/24 brd 172.25.250.255 scope global noprefixroute ens3
            valid_lft forever preferred_lft forever
        inet 10.0.1.2/24 brd 10.0.1.255 scope global noprefixroute ens3
            valid_lft forever preferred_lft forever
        inet6 fe80::74c:3476:4113:463f/64 scope link noprefixroute
            valid_lft forever preferred_lft forever
```

7. Проверьте возможность подключения, используя новые сетевые адреса.

7.1. С **serverb** выполните команду **ping** на новый адрес сервера **servera**.

```
[root@serverb ~]# ping -c3 10.0.1.1
```

```
PING 10.0.1.1 (10.0.1.1) 56(84) bytes of data.
64 bytes from 10.0.1.1: icmp_seq=1 ttl=64 time=0.342 ms
64 bytes from 10.0.1.1: icmp_seq=2 ttl=64 time=0.188 ms
64 bytes from 10.0.1.1: icmp_seq=3 ttl=64 time=0.317 ms
```

```
--- 10.0.1.1 ping statistics ---
```

```
3 packets transmitted, 3 received, 0% packet loss, time 35ms
rtt min/avg/max/mdev = 0.188/0.282/0.342/0.068 ms
```

7.2. Выйдите с сервера **serverb**, чтобы вернуться на рабочую станцию **workstation** в качестве пользователя **student**.

```
[root@serverb ~]# exit  
logout  
[student@serverb ~]$ exit  
logout  
Connection to serverb closed.  
[student@workstation ~]$
```

7.3. С рабочей станции **workstation** используйте команду **ssh**, чтобы войти в систему **servera** как пользователь **student**.

```
[student@workstation ~]$ ssh student@servera  
...output omitted...  
[student@servera ~]$
```

7.4. Используйте команду **sudo -i**, чтобы переключиться на пользователя **root**. Если будет предложено, используйте в качестве пароля слово **student**.

```
[student@servera ~]$ sudo -i  
[sudo] password for student: student  
[root@servera ~]#
```

7.5. С сервера **servera**, пропингуйте новый адрес системы **serverb**.

```
[root@servera ~]# ping -c3 10.0.1.2  
PING 10.0.1.2 (10.0.1.2) 56(84) bytes of data.  
64 bytes from 10.0.1.2: icmp_seq=1 ttl=64 time=0.269 ms  
64 bytes from 10.0.1.2: icmp_seq=2 ttl=64 time=0.338 ms  
64 bytes from 10.0.1.2: icmp_seq=3 ttl=64 time=0.361 ms  
  
--- 10.0.1.2 ping statistics ---  
3 packets transmitted, 3 received, 0% packet loss, time 48ms  
rtt min/avg/max/mdev = 0.269/0.322/0.361/0.044 ms
```

7.6. Выйдите с сервера **servera**.

```
[root@servera ~]# exit  
logout  
[student@servera ~]$ exit  
logout  
Connection to servera closed.  
[student@workstation ~]$
```

Завершение

На рабочей станции **workstation** запустите сценарий **lab net-edit finish**, чтобы завершить упражнение.

```
[student@workstation ~]$ lab net-edit finish
```

На этом пошаговое упражнение завершено.

НАСТРОЙКА ИМЕНИ ХОСТОВ И РАЗРЕШЕНИЯ ИМЕНИ

ЗАДАЧИ

После завершения этого раздела вы сможете настроить статическое имя хоста сервера и разрешение его имени, а также протестировать результаты.

ИЗМЕНЕНИЕ ИМЕНИ ХОСТА СИСТЕМЫ

Команда **hostname** отображает или временно изменяет полное имя хоста системы.

```
[root@host ~]# hostname  
host@example.com
```

В файле **/etc/hostname** можно указать статическое имя хоста. Команда **hostnamectl** используется для изменения этого файла и может применяться для просмотра состояния полного имени хоста системы. Если этот файл не существует, имя хоста устанавливается обратным DNS-запросом после того, как интерфейсу назначен IP-адрес.

```
[root@host ~]# hostnamectl set-hostname host@example.com  
[root@host ~]# hostnamectl status  
  Static hostname: host.example.com  
    Icon name: computer-vm  
      Chassis: vm  
    Machine ID: 73ab164e278e48be9bf80e80714a8cd5  
      Boot ID: 6b1cbc4177164ef58c0e9ed4adb2904f  
Virtualization: kvm  
Operating System: Red Hat Enterprise Linux 8.0 beta (Ootpa)  
    CPE OS Name: cpe:/o:redhat:enterprise_linux:8.0:beta  
      Kernel: Linux 4.18.0-60.el8.x86_64  
Architecture: x86-64  
[root@host ~]# cat /etc/hostname  
host@example.com
```



ВАЖНО

В Red Hat Enterprise Linux 7 и новее статическое имя хоста хранится в **/etc/hostname**. Red Hat Enterprise Linux 6 и более ранних версий сохраняет имя хоста как переменную в файле **/etc/sysconfig/network**.

НАСТРОЙКА РАЗРЕШЕНИЯ ИМЕНИ

stub resolver используется для преобразования имен хостов в IP-адреса или наоборот. Он определяет, где искать, на основе конфигурации файла `/etc/nsswitch.conf`. По умолчанию сначала проверяется содержимое файла `/etc/hosts`.

```
[root@host ~]# cat /etc/hosts
127.0.0.1      localhost localhost.localdomain localhost4 localhost4.localdomain4
::1            localhost localhost.localdomain localhost6 localhost6.localdomain6

172.25.254.254 classroom.example.com
172.25.254.254 content.example.com
```

Команду `getent hosts hostname` можно использовать для проверки разрешения имени хоста с помощью файла `/etc/hosts`.

Если запись не найдена в файле `/etc/hosts`, по умолчанию **stub resolver** пытается найти имя хоста с помощью сервера имен DNS. Файл `/etc/resolv.conf` контролирует выполнение этого запроса:

- **search**: список доменных имен, которые можно попробовать с коротким именем хоста. Причем, домены не должны быть заданы в одном файле; если это так, то выигрывает последний экземпляр. Подробности см. В `resolv.conf(5)`.
- **nameserver**: IP-адрес сервера имен для запроса. Для резервного копирования, если одна из них не работает, можно указать до трех директив сервера имен.

```
[root@host ~]# cat /etc/resolv.conf
# Generated by NetworkManager
domain example.com
search example.com
nameserver 172.25.254.254
```

NetworkManager обновляет файл `/etc/resolv.conf`, используя настройки **DNS** в файлах конфигурации подключения. Используйте `nmcli` для изменения подключений.

```
[root@host ~]# nmcli con mod ID ipv4.dns IP
[root@host ~]# nmcli con down ID
[root@host ~]# nmcli con up ID
[root@host ~]# cat /etc/sysconfig/network-scripts/ifcfg-ID
...output omitted...
DNS1=8.8.8.8
...output omitted...
```

Поведение `nmcli con mod ID ipv4.dns IP` по умолчанию заключается в замене всех предыдущих настроек **DNS** новым списком **IP**. Символ + или - перед аргументом `ipv4.dns` добавляет или удаляет отдельную запись.

```
[root@host ~]# nmcli con mod ID +ipv4.dns IP
```

Чтобы добавить DNS-сервер с IP-адресом **IPv6 001:4860:4860::8888** в список серверов имен для использования с подключением **static-ens3**:

```
[root@host ~]# nmcli con mod static-ens3 +ipv6.dns 2001:4860:4860::8888
```



ПРИМЕЧАНИЕ

Все настройки статического **DNS IPv4** и **IPv6** заканчиваются как директивы сервера имен в **/etc/resolv.conf**. Вы должны убедиться, что в списке указан как минимум **IPv4**-доступный сервер имен (при условии, что система с двумя стеками). Лучше иметь хотя бы один сервер имен, использующий **IPv4**, а второй - **IPv6**, на случай, если у вас есть сетевые проблемы с сетью **IPv4** или **IPv6**.

Проверка разрешения имен DNS

Команду **host HOSTNAME** можно использовать для проверки подключения к **DNS**-серверу.

```
[root@host ~]# host classroom.example.com
classroom.example.com has address 172.25.254.254
[root@host ~]# host 172.25.254.254
254.254.25.172.in-addr.arpa domain name pointer classroom.example.com.
```



ВАЖНО

DHCP автоматически перезаписывает файл **/etc/resolv.conf** при запуске интерфейсов, если вы не укажете **PEERDNS=no** в соответствующих файлах конфигурации интерфейса. Установите это с помощью команды **nmcli**.

```
[root@host ~]# nmcli con mod "static-ens3" ipv4.ignore-auto-dns yes
```



РЕКОМЕНДАЦИИ

Справочные страницы **man nmcli(1)**, **hostnamectl(1)**, **hosts(5)**, **getent(1)**, **host(1)**, и **resolv.conf(5)**

Дополнительная информация может быть доступна в главе о настройке имен хостов в Руководстве по работе с сетью Red Hat Enterprise Linux для Red Hat Enterprise Linux 8, которое можно найти по адресу

<https://access.redhat.com/documentation/>

УПРАЖНЕНИЯ С ПОШАГОВЫМИ ИНСТРУКЦИЯМИ

НАСТРОЙКА ИМЕНИ ХОСТОВ И РАЗРЕШЕНИЯ ИМЕНИ

В этом упражнении вы вручную настроите статическое имя хоста системы, файл `/etc/hosts` и преобразователь имен **DNS**.

В РЕЗУЛЬТАТЕ

У вас должна быть возможность установить собственное имя хоста и настроить параметры разрешения имен.

ПРЕЖДЕ ЧЕМ ВЫ НАЧНЕТЕ

Войдите на рабочую станцию **workstation** как пользователь **student**, используя в качестве пароля слово **student**.

На рабочей станции **workstation** выполните скрипт **lab net-hostnames start**. Команда запускает сценарий, который определяет, доступен ли хост, **servera**, в сети.

```
[student@workstation ~]$ lab net-hostnames start
```

1. Используйте команду **ssh** для входа на сервер **servera** как пользователь **student**. Системы настроены на использование ключей SSH для аутентификации, поэтому пароль для входа на сервер не требуется.

```
[student@workstation ~]$ ssh student@servera  
[student@servera ~]$
```

2. Просмотрите текущие настройки имени хоста.

2.1. Показать текущее имя хоста.

```
[student@servera ~]$ hostname  
servera.lab.example.com
```

2.2. Отображение статуса имени хоста.

```
[student@servera ~]$ hostnamectl status
```

```
Static hostname: n/a
Transient hostname: servera.lab.example.com
Icon name: computer-vm
Chassis: vm
Machine ID: 73ab164e278e48be9bf80e80714a8cd5
Boot ID: 76b13a300c944ab49445af778cb8f749
Virtualization: kvm
Operating System: Red Hat Enterprise Linux 8.0 Beta (Ootpa)
CPE OS Name: cpe:/o:redhat:enterprise_linux:8.0:beta
Kernel: Linux 4.18.0-60.el8.x86_64
Architecture: x86-64
```

3. Задайте статическое имя хоста, соответствующее текущему временному имени хоста.

3.1. Измените имя хоста и файл конфигурации имени хоста.

```
[student@servera ~]$ sudo hostnamectl set-hostname servera.lab.example.com
[sudo] password for student: student
[student@servera ~]$
```

3.2. Просмотрите файл конфигурации, в котором указано имя хоста при запуске сети.

```
[student@servera ~]$ cat /etc/hostname
servera.lab.example.com
```

3.3. Отображение статуса имени хоста.

```
[student@servera ~]$ hostnamectl status
Static hostname: servera.lab.example.com
Icon name: computer-vm
Chassis: vm
Machine ID: 73ab164e278e48be9bf80e80714a8cd5
Boot ID: 76b13a300c944ab49445af778cb8f749
Virtualization: kvm
Operating System: Red Hat Enterprise Linux 8.0 Beta (Ootpa)
CPE OS Name: cpe:/o:redhat:enterprise_linux:8.0:beta
Kernel: Linux 4.18.0-60.el8.x86_64
Architecture: x86-64
```

4. Временно измените имя хоста.

4.1. Измените имя хоста.

```
[student@servera ~]$ sudo hostname testname
```

4.2. Показать текущее имя хоста.

```
[student@servera ~]$ hostname  
testname
```

4.3. Просмотрите файл конфигурации, в котором указано имя хоста при запуске сети.

```
[student@servera ~]$ cat /etc/hostname  
servera.lab.example.com
```

4.4. Перезагрузите систему.

```
[student@servera ~]$ sudo reboot  
Connection to servera closed by remote host  
Connection to servera closed.  
[student@workstation ~]$
```

4.5. С рабочей станции **workstation** войдите в систему **servera** как пользователь **student**.

```
[student@workstation ~]$ ssh student@servera  
[student@servera ~]$
```

4.6. Показать текущее имя хоста.

```
[student@servera ~]$ hostname  
servera.lab.example.com
```

5. Добавьте локальный псевдоним (**nickname**) для сервера класса.

5.1. Найдите IP-адрес **classroom.example.com**.

```
[student@servera ~]$ host classroom.example.com  
classroom.example.com has address 172.25.254.254
```

5.2. Измените **/etc/hosts** так, чтобы имени **class** соответствовал IP-адрес **172.25.254.254** и мог использоваться для связи с **classroom.example.com**.

```
[student@servera ~]$ sudo vim /etc/hosts  
[student@servera ~]$ cat /etc/hosts  
127.0.0.1      localhost localhost.localdomain localhost4 localhost4.localdomain4
```

```
::1           localhost localhost.localdomain localhost6 localhost6.localdomain6
172.25.254.254 classroom.example.com class
172.25.254.254 content.example.com
...content omitted...
```

5.3. Найдите IP-адрес **class**.

```
[student@servera ~]$ host class
Host class not found: 3(NXDOMAIN)
[student@servera ~]$ getent hosts class
172.25.254.254 classroom.example.com class
```

5.4. Ping **class**.

```
[student@servera ~]$ ping -c3 class
PING classroom.example.com (172.25.254.254) 56(84) bytes of data.
64 bytes from classroom.example.com (172.25.254.254): icmp_seq=1 ttl=64 time=0.397ms
64 bytes from classroom.example.com (172.25.254.254): icmp_seq=2 ttl=64 time=0.447ms
64 bytes from classroom.example.com (172.25.254.254): icmp_seq=3 ttl=64 time=0.470ms
--- classroom.example.com ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2000ms
rtt min/avg/max/mdev = 0.397/0.438/0.470/0.030 ms
```

5.5. Выход из сервера **servera**.

```
[student@servera ~]$ exit
logout
Connection to servera closed.
[student@workstation ~]$
```

Завершение

На рабочей станции **workstation**, запустите **lab net-hostnames finish**, чтобы завершить упражнение.

```
[student@workstation ~]$ lab net-hostnames finish
```

На этом пошаговое упражнение завершено.

ЛАБОРАТОРНАЯ РАБОТА

УПРАВЛЕНИЕ СЕТЬЮ

КОНТРОЛЬНЫЙ СПИСОК РАБОТЫ

В лабораторной работе вы настроите сетевые параметры на сервере Red Hat Enterprise Linux.

В РЕЗУЛЬТАТЕ

У вас должна быть возможность настроить два статических IPv4-адреса для основного сетевого интерфейса.

ПРЕЖДЕ ЧЕМ ВЫ НАЧНЕТЕ

Войдите на рабочую станцию **workstation** как пользователь **student**, используя в качестве пароля слово **student**.

На рабочей станции **workstation** выполните скрипт **lab net-review start**. Команда запускает сценарий, который определяет, доступен ли хост **serverb** в сети.

```
[student@workstation ~]$ lab net-review start
```

- Используйте команду **ssh** для входа на **serverb** как пользователь **student**. Системы настроены на использование ключей SSH для аутентификации, поэтому пароль не требуется для входа на **serverb**.
- Используйте команду **sudo -i**, чтобы переключиться на пользователя **root**. Если будет предложено, используйте слово **student** в качестве пароля.
- Создайте новое соединение со статическим сетевым соединением, используя настройки в таблице.

ПАРАМЕТР	ЗНАЧЕНИЯ
Название соединения	lab
Имя интерфейса	Ens3 (может отличаться, используйте интерфейс с MAC-адресом 52:54:00:00:fa:0b)
IP адрес	172.25.250.11/24
Адрес шлюза	172.25.250.254
Адрес DNS	172.25.250.254

- Настройте новое соединение для автоматического запуска. Другие подключения не должны запускаться автоматически.
- Измените новое соединение так, чтобы оно также использовало адрес 10.0.1.1/24.
- Настройте файл **hosts** так, чтобы на 10.0.1.1 можно было ссылаться как на частный(**private**).
- Перезагрузите систему.

Оценка

На рабочей станции **workstation**, запустите сценарий **lab net-review grade**, чтобы подтвердить успех лабораторной работы.

```
[student@workstation ~]$ lab net-review grade
```

Завершение

На рабочей станции **workstation** запустите сценарий **lab net-review finish**, чтобы завершить лабораторную работу.

```
[student@workstation ~]$ lab net-review finish
```

На этом лабораторная работа завершена.

РЕШЕНИЕ

УПРАВЛЕНИЕ СЕТЬЮ

КОНТРОЛЬНЫЙ СПИСОК РАБОТЫ

В лабораторной работе вы настроите сетевые параметры на сервере Red Hat Enterprise Linux.

В РЕЗУЛЬТАТЕ

У вас должна быть возможность настроить два статических IPv4-адреса для основного сетевого интерфейса.

ПРЕЖДЕ ЧЕМ ВЫ НАЧНЕТЕ

Войдите на рабочую станцию **workstation** как пользователь **student**, используя в качестве пароля слово **student**.

На рабочей станции **workstation** выполните скрипт **lab net-review start**. Команда запускает сценарий, который определяет, доступен ли хост **serverb** в сети.

```
[student@workstation ~]$ lab net-review start
```

1. Используйте команду **ssh** для входа на **serverb** как пользователь **student**. Системы настроены на использование ключей SSH для аутентификации, поэтому пароль не требуется для входа на **serverb**.

```
[student@workstation ~]$ ssh student@serverb
[student@serverb ~]$
```

2. Используйте команду **sudo -i**, чтобы переключиться на пользователя **root**. Если будет предложено, используйте слово **student** в качестве пароля.

```
[student@serverb ~]$ sudo -i
[sudo] password for student: student
[root@serverb ~]#
```

3. Создайте новое соединение со статическим сетевым соединением, используя настройки в таблице.

ПАРАМЕТР	ЗНАЧЕНИЯ
Название соединения	lab

Имя интерфейса	Ens3 (может отличаться, используйте интерфейс с MAC-адресом 52:54:00:00:fa:0b)
IP адрес	172.25.250.11/24
Адрес шлюза	172.25.250.254
Адрес DNS	172.25.250.254

Определите имя интерфейса и имя текущего активного соединения. В решении предполагается, что имя интерфейса - **ens3**, а имя подключения - Проводное соединение (**Wired connection**) 1.

```
[root@serverb ~]# ip link
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default
    qlen 1000
        link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
2: ens3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default
    qlen 1000
        link/ether 52:54:00:00:fa:0b brd ff:ff:ff:ff:ff:ff
[root@serverb ~]# nmcli con show --active
NAME           UUID
Wired connection 1 03da038a-3257-4722-a478-53055cc90128  ethernet  ens3
```

Создайте новый профиль подключения к **lab** на основе информации в таблице, описанной в инструкциях. Свяжите профиль с именем вашего сетевого интерфейса, указанным в выходных данных предыдущей команды **ip link**.

```
[root@serverb ~]# nmcli con add con-name lab ifname ens3 type ethernet \
> ip4 172.25.250.11/24 gw4 172.25.250.254
[root@serverb ~]# nmcli con mod "lab" ipv4.dns 172.25.250.254
```

4. Настройте новое соединение для автоматического запуска. Другие подключения не должны запускаться автоматически.

```
[root@serverb ~]# nmcli con mod "lab" connection.autoconnect yes
[root@serverb ~]# nmcli con mod "Wired connection 1" connection.autoconnect no
```

5. Измените новое соединение так, чтобы оно также использовало адрес **10.0.1.1/24**.

```
[root@serverb ~]# nmcli con mod "lab" +ipv4.addresses 10.0.1.1/24
```

Или поочередно:

```
[root@serverb ~]# echo "IPADDR1=10.0.1.1" >> \
> /etc/sysconfig/network-scripts/ifcfg-lab
[root@serverb ~]# echo "PREFIX1=24" >> /etc/sysconfig/network-scripts/ifcfg-lab
```

6. Настройте файл **hosts** так, чтобы на **10.0.1.1** можно было ссылаться как на частный(**private**).

```
[root@serverb ~]# echo "10.0.1.1 private" >> /etc/hosts
```

7. Перезагрузите систему.

```
[root@serverb ~]# reboot
Connection to serverb closed by remote host.
Connection to serverb closed.
[student@workstation ~]$
```

Оценка

На рабочей станции **workstation**, запустите сценарий **lab net-review grade**, чтобы подтвердить успех лабораторной работы.

```
[student@workstation ~]$ lab net-review grade
```

Завершение

На рабочей станции **workstation** запустите сценарий **lab net-review finish**, чтобы завершить лабораторную работу.

```
[student@workstation ~]$ lab net-review finish
```

На этом лабораторная работа завершена.

РЕЗЮМЕ

В этой главе вы узнали:

- Сетевая модель **TCP/IP** представляет собой упрощенный четырехуровневый набор абстракций, который описывает, как различные протоколы взаимодействуют друг с другом, чтобы компьютеры могли отправлять трафик с одной машины на другую через Интернет.
- **IPv4** - основной сетевой протокол, используемый сегодня в Интернете. **IPv6** предназначен в качестве возможной замены сетевого протокола **IPv4**. По умолчанию Red Hat Enterprise Linux работает в режиме двойного стека, используя оба протокола параллельно.
- **NetworkManager** - демон, который отслеживает и управляет конфигурацией сети.
- Команда **nmcli** - инструмент командной строки для настройки сетевых параметров с помощью **NetworkManager**.
- **Статическое имя** хоста системы хранится в файле **/etc/hostname**. Команда **hostnamectl** используется для изменения или просмотра статуса имени хоста системы и связанных настроек. Команда **hostname** отображает или временно изменяет имя хоста системы.

ГЛАВА 13

АРХИВИРОВАНИЕ И ПЕРЕДАЧА ФАЙЛОВ

ЦЕЛЬ

Архивируйте и копируйте файлы из одной системы в другую.

ЗАДАЧИ

- Архивируйте файлы и каталоги в сжатый файл с помощью **tar** и извлекайте содержимое существующего архива **tar**.
- Безопасная передача файлов в удаленную систему или из нее с помощью **SSH**.
- Синхронизировать содержимое локального файла или каталога с копией на удаленном сервере.

РАЗДЕЛЫ

- Управление сжатыми архивами **Tar** (и упражнения с пошаговыми инструкциями)
- Безопасная передача файлов между системами (и упражнения с пошаговыми инструкциями)
- Безопасная синхронизация файлов между системами (и упражнения с пошаговыми инструкциями)

ЛАБОРОТОРНАЯ РАБОТА

Архивирование и передача файлов.

УПРАВЛЕНИЕ АРХИВОМ СЖАТОГО TAR

ЗАДАЧИ

После завершения этого раздела вы сможете архивировать файлы и каталоги в сжатый файл с помощью **tar** и извлекать содержимое из существующего архива **tar**.

КОМАНДА tar

Архивирование и сжатие файлов полезно при создании резервных копий и передаче данных по сети. Одна из старейших и наиболее распространенных команд для создания резервных архивов и работы с ними - это команда **tar**.

С помощью **tar** пользователи могут собирать большие наборы файлов в один файл (архив). Архив **tar** - это структурированная последовательность файловых данных, смешанных с метаданными о каждом файле и индексом, чтобы можно было извлечь отдельные файлы. Архив можно сжать, используя **gzip**, **bzip2** или **xz**.

Команда **tar** может отображать содержимое архивов или извлекать их файлы в текущую систему.

ВЫБРАННЫЕ ОПЦИИ tar

Параметры команды **tar** делятся на операции (действия, которые вы хотите предпринять): общие параметры и параметры сжатия. В таблице ниже показаны общие параметры, длинная версия параметров и их описание:

Обзор операций с tar

ОПЦИИ	ОПИСАНИЕ
-c, --create	Создайте новый архив.
-x, --extract	Извлечь из существующего архива.
-t, --list	Вывести оглавление архива.

Выбранные общие параметры tar

ВАРИАНТ	ОПИСАНИЕ
-v, --verbose	Подробный. Показывает, какие файлы заархивированы или извлечены.
-f, --file =	Имя файла. За этой опцией должно следовать имя файла архива, который нужно использовать или создать.
-p, --preserve-permissions	Сохраните права доступа к файлам и каталогам при распаковке архива без вычитания umask .

Обзор параметров сжатия tar

ВАРИАНТ	ОПИСАНИЕ
-z, --gzip	Используйте сжатие gzip (.tar.gz).
-j, --bzip2	Используйте сжатие bzip2 (.tar.bz2). bzip2 обычно обеспечивает лучшую степень сжатия, чем gzip .
-J, --xz	Используйте сжатие xz (.tar.xz). Сжатие xz обычно обеспечивает лучшую степень сжатия, чем bzip2 .

СПИСОК ВАРИАНТОВ КОМАНДЫ tar

Команда **tar** ожидает один из трех следующих вариантов:

- Используйте параметр **-c** или **--create** для создания архива.
- Используйте параметр **-t** или **--list** для вывода списка содержимого архива.
- Используйте параметр **-x** или **--extract** для извлечения архива.

Другие часто используемые варианты:

- Используйте параметр **-f** или **--file=** с именем файла в качестве аргумента архива для работы.
- Используйте опцию **-v** или **--verbose** для детализации; полезно видеть, какие файлы добавляются или извлекаются из архива.



ПРИМЕЧАНИЕ

На самом деле команда **tar** поддерживает третий, старый стиль параметров, в котором используются стандартные однобуквенные параметры без начальных символов **-**. Он по-прежнему часто встречается, и вы можете столкнуться с этим синтаксисом при работе с инструкциями или командами других людей. Команда **info tar 'old options'** подробно описывает, чем это отличается от обычных коротких опций.

Вы можете пока игнорировать старые параметры и сосредоточиться на стандартном синтаксисе коротких и длинных параметров.

АРХИВИРОВАНИЕ ФАЙЛОВ И КАТАЛОГОВ

Первый вариант, который следует использовать при создании нового архива, - это параметр **-c**, за ним следует параметр **-f**, затем один пробел, затем имя файла архива, который будет создан, и, наконец, список файлов и каталогов, которые должны быть добавлены в архив. Архив создается в текущем каталоге, если не указано иное.



ПРЕДУПРЕЖДЕНИЕ

Перед созданием **tar-архива** убедитесь, что в каталоге нет другого архива с тем же именем, что и новый архив, который нужно создать. Команда **tar** перезаписывает существующий архив без предупреждения.

Следующая команда создает архив с именем **archive.tar** с содержимым файлов **file1**, **file2** и **file3** в домашнем каталоге пользователя.

```
[user@host ~]$ tar -cf archive.tar file1 file2 file3  
[user@host ~]$ ls archive.tar  
archive.tar
```

Вышеупомянутая команда **tar** также может быть выполнена с использованием параметров длинной версии.

```
[user@host ~]$ tar --file=archive.tar --create file1 file2 file3
```



ПРИМЕЧАНИЕ

При архивации файлов по абсолютным путям, начальная **/** в пути удаляется из имени файла по умолчанию. Удаление **/** пути помогает пользователям избежать перезаписи важных файлов при распаковке архива. Команда **tar** извлекает файлы относительно текущего рабочего каталога.

Чтобы **tar** мог архивировать выбранные файлы, необходимо, чтобы пользователь, выполняющий команду **tar**, мог читать файлы. Например, для создания нового архива папки **/etc** и всего ее содержимого требуется привилегии суперпользователя, поскольку только пользователю **root** разрешено читать все файлы, находящиеся в каталоге **/etc**. Непrivилегированный пользователь может создать архив каталога **/etc**, но в архиве отсутствуют файлы, которые не включают разрешение на чтение для пользователя, а также каталоги, которые не включают в себя права на чтение и выполнение для пользователя.

Чтобы создать **tar-архив** с именем **/root/etc.tar** с каталогом **/etc** в качестве содержимого от имени пользователя **root**:

```
[root@host ~]# tar -cf /root/etc.tar /etc  
tar: Removing leading `/' from member names  
[root@host ~]#
```



ВАЖНО

Некоторые расширенные разрешения, которые мы не рассмотрели в этом курсе, такие как **ACL** и контексты **SELinux**, не сохраняются автоматически в **tar-архиве**. Используйте параметр **--xattrs** при создании архива для хранения этих расширенных атрибутов в архиве **tar**.

ПЕРЕЧЕНЬ СОДЕРЖИМОГО АРХИВА

Параметр **t** указывает **tar** на вывод содержимого (оглавление (**table of contents**), отсюда буква, **t**) архива. Используйте параметр **f** с именем запрашиваемого архива. Например:

```
[root@host ~]# tar -tf /root/etc.tar
etc/
etc/fstab
etc/crypttab
etc/mtab
...output omitted...
```

ИЗВЛЕЧЕНИЕ ФАЙЛОВ ИЗ АРХИВА

Архив **tar** обычно следует извлекать в пустой каталог, чтобы гарантировать, что он не перезаписывает существующие файлы. Когда **root** извлекает архив, команда **tar** сохраняет исходного пользователя и пользователя, извлекающего файлы из архива.

Чтобы восстановить файлы из архива **/root/etc.tar** в каталог **/root/etcbackup**, выполните:

```
[root@host ~]# mkdir /root/etcbackup
[root@host ~]# cd /root/etcbackup
[root@host etcbackup]# tar -tf /root/etc.tar
etc/
etc/fstab
etc/crypttab
etc/mtab
...output omitted...
[root@host etcbackup]# tar -xf /root/etc.tar
```

По умолчанию, когда файлы извлекаются из архива, **umask** вычитается из разрешений содержимого архива. Чтобы сохранить права доступа к архивному файлу, используется опция **-p** при распаковке архива.

В этом примере архив с именем **/root/myscripts.tar** извлекается в каталог **/root/scripts** с сохранением прав для извлеченных файлов:

```
[root@host ~]# mkdir /root/scripts  
[root@host ~]# cd /root/scripts  
[root@host scripts]# tar -xpf /root/myscripts.tar
```

СОЗДАНИЕ СЖАТЫХ АРХИВОВ

Команда **tar** поддерживает три различных метода сжатия. Сжатие **gzip** является самым быстрым и старейшим и наиболее широко доступно в разных дистрибутивах и даже на разных платформах. Сжатие **bzip2** создает меньшие архивные файлы по сравнению с **gzip**, но менее широко доступно, чем **gzip**, в то время как метод сжатия **xz** является относительно новым, но обычно предлагает лучший коэффициент сжатия из доступных методов.



ПРИМЕЧАНИЕ

Эффективность любого алгоритма сжатия зависит от типа сжимаемых данных. Файлы данных, которые уже сжаты, такие как сжатые форматы изображений или файлы RPM, обычно имеют низкий коэффициент сжатия.

Рекомендуется использовать один каталог верхнего уровня, который может содержать другие каталоги и файлы, чтобы упростить извлечение файлов организованным способом.

Используйте один из следующих вариантов для создания сжатого tar-архива:

- **-z** или **--gzip** для сжатия **gzip** (filename.tar.gz или filename.tgz)
- **-j** или **--bzip2** для сжатия **bzip2** (filename.tar.bz2)
- **-J** или **-xz** для сжатия **xz** (filename.tar.xz)

Чтобы создать сжатый с помощью **gzip** архив с именем **/root/etcbackup.tar.gz** с содержимым из каталога **/etc** на **host**:

```
[root@host ~]# tar -czf /root/etcbackup.tar.gz /etc  
tar: Removing leading `/' from member names
```

Чтобы создать сжатый архив **bzip2** с именем **/root/logbackup.tar.bz2**, с содержимым из каталога **/var/log** на **host**:

```
[root@host ~]$ tar -cjf /root/logbackup.tar.bz2 /var/log  
tar: Removing leading `/' from member names
```

Чтобы создать сжатый архив **xz** с именем **/root/sshconfig.tar.xz** с содержимым из каталога **/etc/ssh** на **host**:

```
[root@host ~]$ tar -cJf /root/sshconfig.tar.xz /etc/ssh  
tar: Removing leading `/' from member names
```

После создания архива проверьте его содержимое с помощью параметров **tf**. Не обязательно использовать опцию для агента сжатия при перечислении содержимого сжатого архивного файла. Например, чтобы просмотреть содержимое, заархивированное в файле **/root/etcbackup.tar.gz**, в котором используется сжатие **gzip**, используйте следующую команду:

```
[root@host ~]# tar -tf /root/etcbackup.tar.gz /etc  
etc/  
etc/fstab  
etc/crypttab  
etc/mtab  
...output omitted...
```

ИЗВЛЕЧЕНИЕ СЖАТОГО АРХИВА

Первый шаг при распаковке сжатого **tar-архива** - определить, куда следует извлечь заархивированные файлы, а затем создать и перейти в целевой каталог. Команда **tar** определяет, какое сжатие использовалось, и обычно нет необходимости использовать тот же параметр сжатия, который использовался при создании архива. Допустимо добавить метод распаковки к команде **tar**. Если кто-то решит это сделать, необходимо использовать правильный вариант типа декомпрессии; в противном случае **tar** выдает ошибку о типе распаковки, указанном в опциях, не соответствующем типу распаковки файла.

Чтобы извлечь содержимое архива, сжатого с помощью **gzip**, с именем **/root/etcbackup.tar.gz** в каталог **/tmp/etcbackup**:

```
[root@host ~]# mkdir /tmp/etcbackup  
[root@host ~]# cd /tmp/etcbackup  
[root@host etcbackup]# tar -tf /root/etcbackup.tar.gz  
etc/  
etc/fstab  
etc/crypttab  
etc/mtab  
...output omitted...  
[root@host etcbackup]# tar -xzf /root/etcbackup.tar.gz
```

Чтобы извлечь содержимое сжатого архива **bzip2** с именем **/root/logbackup.tar.bz2** в каталог **/tmp/logbackup**:

```
[root@host ~]# mkdir /tmp/logbackup
```

```
[root@host ~]# cd /tmp/logbackup
[root@host logbackup]# tar -tf /root/logbackup.tar.bz2
var/log/
var/log/lastlog
var/log/README
var/log/private/
var/log/wtmp
var/log/btmp
...output omitted...
[root@host logbackup]# tar -xjf /root/logbackup.tar.bz2
```

Чтобы извлечь содержимое сжатого архива **xz** с именем **/root/sshbackup.tar.xz** в каталог **/tmp/sshbackup**:

```
[root@host ~]$ mkdir /tmp/sshbackup
[root@host ~]# cd /tmp/sshbackup
[root@host logbackup]# tar -tf /root/sshbackup.tar.xz
etc/ssh/
etc/ssh/moduli
etc/ssh/ssh_config
etc/ssh/ssh_config.d/
etc/ssh/ssh_config.d/05-redhat.conf
etc/ssh/sshd_config
...output omitted...
[root@host sshbackup]# tar -xJf /root/sshbackup.tar.xz
```

Список сжатого **tar-архива** работает так же, как и список несжатого **tar-архива**.



ПРИМЕЧАНИЕ

Кроме того, **gzip**, **bzip2** и **xz** можно использовать независимо для сжатия отдельных файлов. Например, команда **gzip etc.tar** приводит к сжатому файлу **etc.tar.gz**, команда **bzip2 abc.tar** - к сжатому файлу **abc.tar.bz2**, а команда **xz myarchive.tar** - к **myarchive.tar.xz** сжатый файл **tar.xz**.

Соответствующие команды для распаковки - это **gunzip**, **bunzip2** и **unxz**. Например, команда **gunzip /tmp/etc.tar.gz** возвращает несжатый **tar-файл etc.tar**, а команда **bunzip2 abc.tar.bz2** возвращает несжатый **tar-файл abc.tar** и **unxz myarchive.tar**. Команда **xz** приводит к созданию несжатого **tar-файла myarchive.tar**.



РЕКОМЕНДАЦИИ

Справочные страницы **man tar(1)**, **gzip(1)**, **gunzip(1)**, **bzip2(1)**, **bunzip2(1)**, **xz(1)**, **unxz(1)**

УПРАЖНЕНИЯ С ПОШАГОВЫМИ ИНСТРУКЦИЯМИ

УПРАВЛЕНИЕ АРХИВОМ СЖАТОГО TAR

В этом упражнении вы создадите архивные файлы и извлечете их содержимое с помощью команды **tar**.

В РЕЗУЛЬТАТЕ

У вас должна быть возможность заархивировать дерево каталогов и извлечь содержимое архива в другое место.

ПРЕЖДЕ ЧЕМ ВЫ НАЧНЕТЕ

Войдите на рабочую станцию **workstation** как пользователь **student**, используя в качестве пароля слово **student**.

На рабочей станции **workstation** выполните скрипт **lab archive-manage start**. Команда запускает сценарий, который определяет, доступен ли хост, **servera**, в сети. Сценарий также гарантирует, что файл и каталог, которые будут созданы в упражнении, не существуют на сервере.

```
[student@workstation ~]$ lab archive-manage start
```

1. Используйте команду **ssh** для входа на сервер **servera** как пользователь **student**.

```
[student@workstation ~]$ ssh student@servera  
[student@servera ~]$
```

2. Переключитесь на пользователя **root**, поскольку только пользователь **root** может получить доступ ко всему содержимому каталога **/etc**.

```
[student@servera ~]$ su -  
Password: redhat  
[root@servera ~]#
```

3. Используйте команду **tar** с параметрами **-czf**, чтобы создать архив каталога **/etc** с использованием сжатия **gzip**. Сохраните архивный файл как **/tmp/etc.tar.gz**.

```
[root@servera ~]# tar -czf /tmp/etc.tar.gz /etc  
tar: Removing leading `/' from member names  
[root@servera ~]#
```

4. Используйте команду **tar** с параметрами **-tf**, чтобы убедиться, что архив **etc.tar.gz** содержит файлы из каталога **/etc**.

```
[root@servera ~]# tar -tf /tmp/etc.tar.gz
etc/
etc/mtab
etc/fstab
etc/crypttab
etc/resolv.conf
...output omitted...
```

5. На сервере создайте каталог с именем **/backuptest**. Убедитесь, что файл резервной копии **etc.tar.gz** является допустимым архивом, распаковав файл в каталог **/backuptest**.

5.1. Создайте каталог **/backuptest**.

```
[root@servera ~]# mkdir /backuptest
```

5.2. Перейдите в каталог **/backuptest**.

```
[root@servera ~]# cd /backuptest
[root@servera backuptest]#
```

5.3. Перечислите содержимое архива **etc.tar.gz** перед распаковкой.

```
[root@servera backuptest]# tar -tf /tmp/etc.tar.gz
etc/
etc/mtab
etc/fstab
etc/crypttab
etc/resolv.conf
...output omitted...
```

5.4. Распакуйте архив **/tmp/etc.tar.gz** в каталог **/backuptest**.

```
[root@servera backuptest]# tar -xzf /tmp/etc.tar.gz
[root@servera backuptest]#
```

5.5. Вывести список содержимого каталога **/backuptest**. Убедитесь, что каталог содержит файлы из каталога **/etc**.

```
[root@servera backuptest]# ls -l
```

```
total 12
drwxr-xr-x. 95 root root 8192 Feb 8 10:16 etc
[root@servera backuptest]# cd etc
[root@servera etc]# ls -l
total 1204
-rw-r--r--. 1 root root 16 Jan 16 23:41 adjtime
-rw-r--r--. 1 root root 1518 Sep 10 17:21 aliases
drwxr-xr-x. 2 root root 169 Feb 4 21:58 alternatives
-rw-r--r--. 1 root root 541 Oct 2 21:01 anacrontab
...output omitted...
```

6. Выход из сервера.

```
[root@servera backuptest]# exit
logout
[student@servera ~]$ exit
logout
Connection to servera closed.
[student@workstation]$
```

Завершение

На рабочей станции **workstation** запустите сценарий **lab archive-manage finish**, чтобы завершить это упражнение.

```
[student@workstation ~]$ lab archive-manage finish
```

На этом пошаговое упражнение завершено.

БЕЗОПАСНАЯ ПЕРЕДАЧА ФАЙЛОВ МЕЖДУ СИСТЕМАМИ

ЗАДАЧИ

После завершения этого раздела вы сможете безопасно передавать файлы в удаленную систему или из нее, используя **SSH**.

ПЕРЕДАЧА ФАЙЛОВ С ПОМОЩЬЮ ЗАЩИТНОЙ КОПИИ

OpenSSH полезен для безопасного выполнения команд оболочки в удаленных системах. Команда **Secure Copy**, **scp**, которая является частью пакета **OpenSSH**, копирует файлы из удаленной системы в локальную или из локальной системы в удаленную. Команда использует **SSH-сервер** для аутентификации и шифрует данные при их передаче.

Вы можете указать удаленное расположение для источника или назначения копируемых файлов. Формат удаленного местоположения должен быть в виде **[user@]host:/path**. Часть аргумента **user@** необязательна. Если он отсутствует, будет использоваться ваше текущее локальное имя пользователя. Когда вы запускаете команду, ваш **scp-клиент** будет аутентифицироваться на удаленном **SSH-сервере** точно так же, как **ssh**, используя аутентификацию на основе ключей или запрашивая у вас пароль.

В следующем примере показано, как скопировать локальные файлы **/etc/yum.conf** и **/etc/hosts** на компьютере **host** в домашний каталог удаленного пользователя (**remoteuser**) на удаленной системе **remotehost**:

```
[user@host ~]$ scp /etc/yum.conf /etc/hosts remoteuser@remotehost:/home/remoteuser
remoteuser@remotehost's password: password
      yum.conf          100%   813      0.8KB/s  00:00
      hosts            100%   227      0.2KB/s  00:00
```

Вы также можете скопировать файл в другом направлении, из удаленной системы в локальную файловую систему. В этом примере файл **/etc/hostname** на удаленном хосте (**remotehost**) копируется в локальный каталог **/home/user**. Команда **scp** аутентифицируется на удаленном хосте как удаленный пользователь(**remoteuser**).

```
[user@host ~]$ scp remoteuser@remotehost:/etc/hostname /home/user
remoteuser@remotehost's password: password
      hostname        100%     22          0.0KB/s  00:00
```

Чтобы рекурсивно скопировать все дерево каталогов, используйте параметр **-r**. В следующем примере удаленный каталог **/var/log** на удаленном хосте рекурсивно копируется в локальный каталог **/tmp** на хосте. Вы должны подключиться к удаленной системе как **root**, чтобы убедиться, что вы можете читать все файлы в удаленном каталоге **/var/log**.

```
[user@host ~]$ scp -r root@remoteuser:/var/log /tmp  
root@remotehost's password: password  
...output omitted...
```

ПЕРЕДАЧА ФАЙЛОВ С ПОМОЩЬЮ ПРОГРАММЫ БЕЗОПАСНОЙ ПЕРЕДАЧИ ФАЙЛОВ

Чтобы в интерактивном режиме загружать или скачивать файлы с **SSH-сервера**, используйте программу безопасной передачи файлов **sftp**. Сеанс с командой **sftp** использует механизм безопасной аутентификации и передачу зашифрованных данных на сервер **SSH** и с него.

Как и команда **scp**, команда **sftp** использует **[user@]host** для определения целевой системы и имени пользователя. Если вы не укажете пользователя, команда попытается войти в систему, используя ваше локальное имя пользователя в качестве имени удаленного пользователя. Затем вам будет предложено **sftp>** приглашение.

```
[user@host ~]$ sftp remoteuser@remotehost
remoteuser@remotehost's password: password
Connected to remotehost.
sftp>
```

Интерактивный сеанс **sftp** принимает различные команды, которые работают в удаленной файловой системе так же, как и в локальной файловой системе, например **ls**, **cd**, **mkdir**, **rmdir** и **pwd**. Команда **put** загружает файл в удаленную систему. Команда **get** загружает файл из удаленной системы. Команда **exit** завершает сеанс **sftp**.

Чтобы загрузить файл **/etc/hosts** в локальной системе во вновь созданный каталог **/home/remoteuser/hostbackup** на удаленном хосте (**remotehost**). Сеанс **sftp** всегда предполагает, что за командой **put** следует файл в локальной файловой системе и запускается в домашнем каталоге подключающегося пользователя; в этом случае **/home/remoteuser**:

```
sftp> mkdir hostbackup
sftp> cd hostbackup
sftp> put /etc/hosts
Uploading /etc/hosts to /home/remoteuser/hostbackup/hosts
/etc/hosts                                         100%   227      0.2KB/s   00:00
sftp>
```

Чтобы загрузить **/etc/yum.conf** с удаленного хоста в текущий каталог в локальной системе, выполните команду **get /etc/yum.conf** и выйдите из сеанса **sftp** с помощью команды **exit**.

```
sftp> get /etc/yum.conf
Fetching /etc/yum.conf to yum.conf
/etc/yum.conf                                         100%     813      0.8KB/s   00:00
sftp> exit
[user@host ~]$
```



РЕКОМЕНДАЦИИ

Справочные страницы **man scp(1)** и **sftp(1)**

УПРАЖНЕНИЯ С ПОШАГОВЫМИ ИНСТРУКЦИЯМИ

В этом упражнении вы скопируете файлы из удаленной системы в локальный каталог с помощью команды **scp**.

В РЕЗУЛЬТАТЕ

У вас должна быть возможность копировать файлы с удаленного хоста в каталог на локальном компьютере.

ПРЕЖДЕ ЧЕМ ВЫ НАЧНЕТЕ

Войдите на рабочую станцию **workstation** как пользователь **student**, используя в качестве пароля слово **student**.

На рабочей станции **workstation** выполните скрипт **lab archive-transfer start**. Команда запускает сценарий, который определяет, достижимы ли хосты **servera** и **serverb** в сети. Сценарий также гарантирует, что файл и каталог, которые будут созданы в упражнении, не существуют на сервере.

```
[student@workstation ~]$ lab archive-transfer start
```

1. Используйте команду **ssh** для входа на сервер **servera** как пользователь **student**.

```
[student@workstation ~]$ ssh student@servera  
[student@servera ~]$
```

2. Используйте команду **scp**, чтобы скопировать каталог **/etc/ssh** с системы **serverb** в каталог **/home/student/serverbackup** на сервер **servera**.

- 2.1. На сервере **servera** создайте каталог с именем **/home/student/serverbackup**

```
[student@servera ~]$ mkdir ~/serverbackup
```

- 2.2. Используйте команду **scp**, чтобы рекурсивно скопировать каталог **/etc/ssh** с **serverb** в каталог **/home/student/serverbackup** на сервер **servera**. При появлении запроса введите в качестве пароля **redhat**. Обратите внимание, что только пользователь **root** может читать все содержимое каталога **/etc/ssh**.

```
[student@servera ~]$ scp -r root@serverb:/etc/ssh ~/serverbackup  
The authenticity of host 'serverb (172.25.250.11)' can't be established.  
ECDSA key fingerprint is SHA256:qaS0PToLrqlCO2XGklA0iY7CaP7aPKimerDoeUkv720.  
Are you sure you want to continue connecting (yes/no)? yes
```

```
Warning: Permanently added 'serverb,172.25.250.11' (ECDSA) to the list of known hosts.
```

```
root@serverb's password: redhat
```

ssh_config	100%	1727	1.4MB/s	00:00
05-redhat.conf	100%	690	1.6MB/s	00:00
sshd_config	100%	4469	9.5MB/s	00:00
ssh_host_ed25519_key	100%	387	1.2MB/s	00:00
ssh_host_ed25519_key.pub	100%	82	268.1KB/s	00:00
ssh_host_ecdsa_key	100%	492	1.5MB/s	00:00
ssh_host_ecdsa_key.pub	100%	162	538.7KB/s	00:00
ssh_host_rsa_key	100%	1799	4.9MB/s	00:00
ssh_host_rsa_key.pub	100%	382	1.2MB/s	00:00

2.3. Убедитесь, что каталог **/etc/ssh** с **serverb** скопирован в каталог **/home/student/serverbackup** на сервер **servera**.

```
[student@servera ~]$ ls -lR ~/serverbackup
/home/student/serverbackup:
total 0
drwxr-xr-x. 3 student student 245 Feb 11 18:35 ssh

/home/student/serverbackup/ssh:
total 588
-rw-r--r--. 1 student student 563386 Feb 11 18:35 moduli
-rw-r--r--. 1 student student 1727 Feb 11 18:35 ssh_config
drwxr-xr-x. 2 student student 28 Feb 11 18:35 ssh_config.d
-rw-----. 1 student student 4469 Feb 11 18:35 sshd_config
-rw-r-----. 1 student student 492 Feb 11 18:35 ssh_host_ecdsa_key
-rw-r--r--. 1 student student 162 Feb 11 18:35 ssh_host_ecdsa_key.pub
-rw-r-----. 1 student student 387 Feb 11 18:35 ssh_host_ed25519_key
-rw-r--r--. 1 student student 82 Feb 11 18:35 ssh_host_ed25519_key.pub
-rw-r-----. 1 student student 1799 Feb 11 18:35 ssh_host_rsa_key
-rw-r--r--. 1 student student 382 Feb 11 18:35 ssh_host_rsa_key.pub

/home/student/serverbackup/ssh/ssh_config.d:
total 4
-rw-r--r--. 1 student student 690 Feb 11 18:35 05-redhat.conf
```

3. Выход из системы **servera**.

```
[student@servera ~]$ exit
logout
Connection to servera closed.
[student@workstation]$
```

Завершение

На рабочей станции **workstation**, запустите сценарий **lab archive-transfer finish**, чтобы выполнить это упражнение.

```
[student@workstation ~]$ lab archive-transfer finish
```

На этом пошаговое упражнение завершено.

БЕЗОПАСНАЯ СИНХРОНИЗАЦИЯ ФАЙЛОВ МЕЖДУ СИСТЕМАМИ

ЗАДАЧИ

После завершения этого раздела вы сможете эффективно и безопасно синхронизировать содержимое локального файла или каталога с копией на удаленном сервере.

СИНХРОНИЗИРУЙТЕ ФАЙЛЫ И КАТАЛОГИ С ПОМОЩЬЮ КОМАНДЫ rsync

Команда **rsync** - еще один способ безопасного копирования файлов из одной системы в другую. Инструмент использует алгоритм, который сводит к минимуму объем копируемых данных, синхронизируя только те части файлов, которые были изменены. Он отличается от **scp** тем, что, если два файла или каталога на двух серверах похожи, **rsync** копирует различия между файловыми системами на двух серверах, тогда как **scp** нужно будет копировать всё.

Одним из преимуществ **rsync** является то, что он может безопасно и эффективно копировать файлы между локальной системой и удаленной системой. Хотя первоначальная синхронизация каталога занимает примерно то же время, что и его копирование, любая последующая синхронизация требует только копирования различий по сети, что, возможно, существенно ускоряет обновления.

Одна из наиболее важных опций **rsync** - опция **-n** для пробного запуска. Пробный прогон - это имитация того, что происходит при выполнении команды. Пробный прогон показывает изменения, которые **rsync** выполнит, если команда будет запущена без опции пробного запуска. Вы должны выполнить пробный запуск перед выполнением операции **rsync**, чтобы убедиться, что важные файлы не будут перезаписаны или удалены.

Двумя наиболее распространенными параметрами при синхронизации файлов и каталогов с помощью **rsync** являются параметры **-v** и **-a**.

Параметр **-v** или **--verbose** обеспечивает более подробный вывод по мере выполнения синхронизации. Это полезно для устранения неполадок и помогает увидеть прогресс.

Параметр **-a** или **--archive** включает «режим архивации». Это быстрый способ включить рекурсивное копирование и включить большое количество полезных опций для сохранения большинства характеристик файлов. Режим архивации использует следующие параметры:

Параметры, доступные с помощью rsync -a (режим архива)

ОПЦИИ	ОПИСАНИЕ
-r, --recursive	рекурсивно синхронизировать все дерево каталогов
-l, --links	синхронизировать символические ссылки
-p, --perms	сохранить разрешения
-t, - раз	сохранить отметки времени
-g, --group	сохранить групповую собственность
-o, --owner	сохранить владельца файлов
-D, --devices	синхронизировать файл устройства

В режиме архива жесткие ссылки не сохраняются, поскольку это может значительно увеличить время синхронизации. Если вы хотите сохранить жесткие ссылки, добавьте параметр **-H**.



ПРИМЕЧАНИЕ

Если вы используете расширенные разрешения, вам может потребоваться добавить два дополнительных параметра:

- **-A** для сохранения списков **ACL**
- **-X** для сохранения контекстов **SELinux**

Вы можете использовать **rsync** для синхронизации содержимого локального файла или каталога с файлом или каталогом на удаленном компьютере, используя любую машину в качестве источника. Вы также можете синхронизировать содержимое двух локальных файлов или каталогов.

Например, чтобы синхронизировать содержимое каталога **/var/log** с каталогом **/tmp**:

```
[user@host ~]$ su -
Password: password
[root@host ~]# rsync -av /var/log /tmp
receiving incremental file list
log/
log/README
log/boot.log
...output omitted...
log/tuned/tuned.log

sent 11,592,423 bytes    received 779 bytes  23,186,404.00 bytes/sec
total size is 11,586,755   speedup is 1.00
[root@host ~]$ ls /tmp
log  ssh-RLjDdarkKiW1
[root@host ~]$
```

Завершающая косая черта в конце исходного каталога синхронизирует содержимое каталога без создания нового подкаталога в целевом каталоге. В этом примере каталог **log** не создается в каталоге **/tmp**. Только содержимое каталога **/var/log/** синхронизируется с каталогом **/tmp**.

```
[root@host ~]# rsync -av /var/log/ /tmp
sending incremental file list
./
README
boot.log
```

```
...output omitted...
tuned/tuned.log
sent 11,592,389 bytes received 778 bytes 23,186,334.00 bytes/sec
total size is 11,586,755 speedup is 1.00
[root@host ~]# ls /tmp
anaconda          dnf.rpm.log-20190318  private
audit             dnf.rpm.log-20190324  qemu-ga
boot.log          dnf.rpm.log-20190331  README
...output omitted...
```



ВАЖНО

При вводе исходного каталога для команды **rsync** важно помнить, что наличие завершающей косой черты в имени каталога имеет значение. Он определяет, синхронизируется ли каталог или только его содержимое с целью.

Завершение команд **Bash** с помощью табуляции автоматически добавляет завершающую косую черту в конце имен каталогов.

Как и команды **scp** и **sftp**, для **rsync** вы указываете удаленные местоположения в формате **[user@]host:/path**. Удаленным расположением может быть либо исходная система, либо система назначения, но одна из двух машин должна быть локальной.

Чтобы сохранить право собственности на файл, вам необходимо иметь **root-права** в целевой системе. Если место назначения удалённое, авторизуйтесь как **root**. Если место назначения является локальным, вы должны запустить **rsync** от имени пользователя **root**.

В этом примере синхронизируйте локальный каталог **/var/log** с каталогом **/tmp** в системе **remotehost**:

```
[root@host ~]# rsync -av /var/log remotehost:/tmp
root@remotehost's password: password
receiving incremental file list
log/
log/README
log/boot.log
...output omitted...
Sent 9,783 bytes received 290,576 bytes    85,816.86 bytes/sec
total size is 11,585,690      speedup is 38.57
```

Таким же образом удаленный каталог **/var/log** на хосте **remotehost** можно синхронизировать с локальным каталогом **/tmp** на хосте **host**:

```
[root@host ~]# rsync -av remotehost:/var/log /tmp
root@remotehost's password: password
```

```
receiving incremental file list
```

```
log/boot.log
```

```
log/dnf.librepo.log
```

```
log/dnf.log
```

```
...output omitted...
```

```
sent 9,783 bytes      received 290,576 bytes      85,816.86 bytes/sec
```

```
total size is 11,585,690      speedup is 38.57
```



РЕКОМЕНДАЦИИ

Справочные страницы **man rsync(1)**

УПРАЖНЕНИЯ С ПОШАГОВЫМИ ИНСТРУКЦИЯМИ

В этом упражнении вы синхронизируете содержимое локального каталога с копией на удаленном сервере с помощью **rsync** через SSH.

В РЕЗУЛЬТАТЕ

Вы должны использовать команду **rsync** для синхронизации содержимого локального каталога с копией на удаленном сервере.

ПРЕЖДЕ ЧЕМ ВЫ НАЧНЕТЕ

Войдите на рабочую станцию **workstation** как пользователь **student**, используя в качестве пароля слово **student**.

На рабочей станции **workstation** выполните команду **lab archive-sync start**. Команда запускает сценарий, который определяет, доступны ли в сети хосты, **servera** и **serverb**. Сценарий также гарантирует, что файл и каталог, которые будут созданы в упражнении, не существуют на сервере **servera**.

```
[student@workstation ~]$ lab archive-sync start
```

1. Используйте команду **ssh** для входа на сервер **servera**, как пользователь **student**.

```
[student@workstation ~]$ ssh student@servera  
[student@servera ~]$
```

2. Создайте на сервере **servera** каталог с именем **/home/student/serverlogs**. Используйте команду **rsync** для безопасного создания начальной копии дерева каталогов **/var/log** с системы **serverb** в каталог **/home/student/serverlogs** на сервер **servera**.
 - 2.1. На сервере **servera** создайте целевой каталог с именем **/home/student/serverlogs** для хранения **log** файлов, синхронизированных с **serverb**.

```
[student@servera ~]$ mkdir ~/serverlogs
```

- 2.2. Используйте команду **rsync** для синхронизации дерева каталогов **/var/log** на сервере **serverb** с каталогом **/home/student/serverlogs** на сервере **servera**. Обратите внимание, что только пользователь **root** может читать все содержимое каталога **/var/log** на **serverb**. Все файлы передаются при начальной синхронизации.

```
[student@servera ~]$ rsync -av root@serverb:/var/log ~/serverlogs  
root@serverb's password: redhat
```

```
receiving incremental file list
log/
log/README
log/boot.log
...output omitted...
log/tuned/tuned.log

sent 992 bytes    received 13,775,064 bytes   2,119,393.23 bytes/sec
total size is 13,768,109  speedup is 1.00
```

3. В качестве пользователя **root** на **serverb** выполните команду **logger "Log files synchronized"**, чтобы получить новую запись в файле журнала **/var/log/messages**, отражающую дату последней синхронизации.

```
[student@servera ~]$ ssh root@serverb 'logger "Log files synchronized"'
Password: redhat
[student@servera ~]$
```

4. Используйте команду **rsync** для безопасной синхронизации из дерева каталогов **/var/log** на сервере **serverb** с каталогом **/home/student/serverlogs** на сервере **servera**. Обратите внимание, что на этот раз передаются только измененные файлы журналов.

```
[student@servera ~]$ rsync -av root@serverb:/var/log ~/serverlogs
root@serverb's password: redhat
receiving incremental file list
log/messages
log/secure
log/audit/audit.log

sent 3,496 bytes  received 27,243 bytes      8,782.57 bytes/sec
total size is 11,502,695  speedup is 374.21
```

5. Выход из сервера **servera**.

```
[student@servera ~]$ exit
logout
Connection to servera closed.
[student@workstation]$
```

Завершение

На рабочей станции **workstation**, запустите сценарий **lab archive-sync finish**, чтобы завершить упражнение.

```
[student@workstation ~]$ lab archive-sync finish
```

На этом пошаговое упражнение завершено.

ЛАБОРАТОРИНАЯ РАБОТА

АРХИВИРОВАНИЕ И ПЕРЕДАЧА ФАЙЛОВ

КОНТРОЛЬНЫЙ СПИСОК РАБОТЫ

В этой лабораторной работе вы будете использовать команды **tar**, **rsync** и **scp** для архивации и резервного копирования содержимого каталогов.

В РЕЗУЛЬТАТЕ

Вы должны быть способны:

- Синхронизируйте удаленный каталог с локальным каталогом.
- Создать архив содержимого синхронизированного каталога.
- Надежно скопируйте архив на удаленный хост.
- Распаковать архив.

ПРЕЖДЕ ЧЕМ ВЫ НАЧНЕТЕ

Войдите на рабочую станцию **workstation** как пользователь **student**, используя в качестве пароля слово **student**.

На рабочей станции **workstation** выполните команду **lab archive-review start**. Команда запускает сценарий, который определяет, достижимы ли хосты **servera** и **serverb** в сети. Сценарий также гарантирует, что файлы и каталоги, которые будут созданы в лабораторной работе, не существуют на **serverb** и рабочей станции **workstation**.

```
[student@workstation ~]$ lab archive-review start
```

1. На **serverb** синхронизируйте дерево каталогов **/etc** с сервера **servera** в каталог **/configsync**.
2. Используйте сжатие **gzip**, чтобы создать архив с именем **configfile-backup-servera.tar.gz** с содержимым каталога **/configsync**.
3. Надежно скопируйте архивный файл **/root/configfile-backup-servera.tar.gz** с **serverb** в каталог **/home/student** на рабочей станции **workstation** в качестве пользователя **student**, используя в качестве пароля слово **student**.
4. На рабочей станции **workstation** извлеките содержимое архива **/home/student/configfile-backup-servera.tar.gz** в каталог **/tmp/savedconfig/**.
5. На рабочей станции **workstation** вернитесь в домашний каталог пользователя **student**.

```
[student@workstation savedconfig]$ cd
```

Оценка

На рабочей станции **workstation** запустите сценарий **lab archive-review grade**, чтобы подтвердить успешное выполнение лабораторной работы.

```
[student@workstation ~]$ lab archive-review grade
```

Завершение

На рабочей станции **workstation** запустите сценарий **lab archive-review finish**, чтобы завершить упражнение.

```
[student@workstation ~]$ lab archive-review finish
```

На этом лабораторная работа завершена.

РЕШЕНИЕ

АРХИВИРОВАНИЕ И ПЕРЕДАЧА ФАЙЛОВ

КОНТРОЛЬНЫЙ СПИСОК РАБОТЫ

В этой лабораторной работе вы будете использовать команды **tar**, **rsync** и **scp** для архивации и резервного копирования содержимого каталогов.

В РЕЗУЛЬТАТЕ

Вы должны быть способны:

- Синхронизируйте удаленный каталог с локальным каталогом.
- Создать архив содержимого синхронизированного каталога.
- Надежно скопируйте архив на удаленный хост.
- Распаковать архив.

ПРЕЖДЕ ЧЕМ ВЫ НАЧНЕТЕ

Войдите на рабочую станцию **workstation** как пользователь **student**, используя в качестве пароля слово **student**.

На рабочей станции **workstation** выполните команду **lab archive-review start**. Команда запускает сценарий, который определяет, достижимы ли хосты **servera** и **serverb** в сети. Сценарий также гарантирует, что файлы и каталоги, которые будут созданы в лабораторной работе, не существуют на **serverb** и рабочей станции **workstation**.

```
[student@workstation ~]$ lab archive-review start
```

1. На **serverb** синхронизируйте дерево каталогов **/etc** с сервера **servera** в каталог **/configsync**.

1.1. Используйте команду **ssh** для входа на **serverb** как пользователь **student**.

```
[student@workstation ~]$ ssh student@serverb  
[student@serverb ~]$
```

1.2. Используйте команду **su**, чтобы переключиться на пользователя **root**, так как для создания каталога **/configsync** требуются привилегии суперпользователя. На более поздних этапах вы заархивируете файлы, присутствующие в дереве каталогов **/etc**, которые принадлежат пользователю **root**, это также требует привилегий суперпользователя.

```
[student@serverb ~]$ su -  
Password: redhat  
[root@serverb ~]#
```

- 1.3.** Создайте каталог **/configsync** для хранения синхронизированных файлов конфигурации с сервера.

```
[root@serverb ~]# mkdir /configsync
```

- 1.4.** Используйте команду **rsync** для синхронизации дерева каталогов **/etc** с системы **servera** в каталог **/configsync** на сервер **serverb**.

Имейте в виду, что только пользователь **root** может читать все содержимое каталога **/etc** на сервере **servera**.

```
[root@serverb ~]# rsync -av root@servera:/etc /configsync  
root@servera's password: redhat  
receiving incremental file list  
etc/  
etc/.pwd.lock  
...output omitted...  
etc/yum/protected.d -> ../dnf/protected.d  
etc/yum/vars -> ../dnf/vars  
  
sent 10,958 bytes      received 21,665,987 bytes   3,334,914.62 bytes/sec  
total size is 21,615,767  speedup is 1.00
```

- 2.** Используйте сжатие **gzip**, чтобы создать архив с именем **configfile-backup-servera.tar.gz** с содержимым каталога **/configsync**.

- 2.1.** Используйте команду **tar** с параметрами **-czf**, чтобы создать архив, сжатый **gzip**.

```
[root@serverb ~]# tar -czf configfile-backup-servera.tar.gz /configsync  
tar: Removing leading `/' from member names  
[root@serverb ~]#
```

- 2.2.** Используйте команду **tar** с параметрами **-tf**, чтобы просмотреть содержимое архива **configfilebackup-servera.tar.gz**.

```
[root@serverb ~]# tar -tf configfile-backup-servera.tar.gz  
...output omitted...  
configsync/etc/vimrc  
configsync/etc/wgetrc
```

```
configsync/etc/xattr.conf
```

3. Безопасно скопируйте архивный файл **/root/configfile-backup-servera.tar.gz** с **serverb** в каталог **/home/student** на рабочей станции **workstation** в качестве пользователя **student**, используя в качестве пароля слово **student**.

```
[root@serverb ~]# scp ~/configfile-backup-servera.tar.gz \
> student@workstation:/home/student
...output omitted...
student@workstation's password: student
configfile-backup-servera.tar.gz      100% 5110KB      64.5MB/s   00:00
```

4. На рабочей станции **workstation** извлеките содержимое архива **/home/student/configfile-backup-servera.tar.gz** в каталог **/tmp/savedconfig/**.

4.1. Выйти с сервера **serverb**.

```
[root@serverb ~]# exit
logout
[student@serverb ~]$ exit
logout
Connection to serverb closed.
[student@workstation]$
```

4.2. Создайте каталог **/tmp/savedconfig**, куда будет извлекаться содержимое архива **/home/student/configfile-backup-servera.tar.gz**.

```
[student@workstation ~]$ mkdir /tmp/savedconfig
```

4.3. Перейдите в каталог **/tmp/savedconfig**.

```
[student@workstation ~]$ cd /tmp/savedconfig
[student@workstation savedconfig]$
```

4.4. Используйте команду **tar** с параметрами **-tfz**, чтобы просмотреть содержимое архива **configfilebackup-servera.tar.gz**

```
[student@workstation savedconfig]$ tar -tfz ~/configfile-backup-servera.tar.gz
...output omitted...
configsync/etc/vimrc
```

```
configsSync/etc/wgetrc  
configsSync/etc/xattr.conf
```

- 4.5.** Используйте команду **tar** с параметрами **-xzf**, чтобы извлечь содержимое архива **/home/student/configfile-backup-servera.tar.gz** в каталог **/tmp/savedconfig/**.

```
[student@workstation savedconfig]$ tar -xzf ~/configfile-backup-servera.tar.gz  
[student@workstation savedconfig]$
```

- 4.6.** Просмотрите дерево каталогов, чтобы убедиться, что директория содержит файлы из каталога **/etc**.

```
[student@workstation savedconfig]$ ls -lR  
.:  
total 0  
drwxr-xr-x. 3 student student 17 Feb 13 10:13 configsSync  
  
.configsSync:  
total 12  
drwxr-xr-x. 95 student student 8192 Feb 13 09:41 etc  
  
.configsSync/etc:  
total 1212  
-rw-r--r--. 1 student student 16 Jan 16 23:41 adjtime  
-rw-r--r--. 1 student student 1518 Sep 10 17:21 aliases  
drwxr-xr-x. 2 student student 169 Feb 4 21:58 alternatives  
...output omitted...
```

- 5.** На рабочей станции **workstation** вернитесь в домашний каталог пользователя **student**.

```
[student@workstation savedconfig]$ cd
```

Оценка

На рабочей станции **workstation** запустите сценарий **lab archive-review grade**, чтобы подтвердить успешное выполнение лабораторной работы.

```
[student@workstation ~]$ lab archive-review grade
```

Завершение

На рабочей станции **workstation** запустите сценарий **lab archive-review finish**, чтобы завершить упражнение.

```
[student@workstation ~]$ lab archive-review finish
```

На этом лабораторная работа завершена.

РЕЗЮМЕ

В этой главе вы узнали:

- Команда **tar** создает файл архива из набора файлов и каталогов, извлекает файлы из архива и выводит список содержимого архива.
- Команда **tar** предоставляет набор различных методов сжатия для уменьшения размера архива.
- Помимо обеспечения безопасной удаленной оболочки, служба **SSH** также предоставляет команды **scp** и **sftp** в качестве безопасных способов передачи файлов с и на удаленную систему, на которой запущен сервер **SSH**.
- Команда **rsync** надежно и эффективно синхронизирует файлы между двумя каталогами, любой из которых может находиться в удаленной системе.

ГЛАВА 14

УСТАНОВКА И ОБНОВЛЕНИЕ ПАКЕТОВ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

ЦЕЛЬ

Загружайте, устанавливайте, обновляйте и управляйте пакетами программного обеспечения из репозиториев пакетов **Red Hat** и **Yum**.

ЗАДАЧИ

- Зарегистрируйте систему в своей учетной записи Red Hat и назначьте ей права на обновления программного обеспечения и услуги поддержки с помощью управления подпиской (**Subscription Management**) Red Hat.
- Объясните, как программное обеспечение предоставляется в виде пакетов **RPM**, и исследуйте пакеты, установленные в системе с помощью **Yum** и **RPM**.
- Найдите, установите и обновите пакеты программного обеспечения с помощью команды **yum**.
- Включение и отключение использования сервером Red Hat или сторонних репозиториев **Yum**.
- Объясните, как модули позволяют устанавливать определенные версии программного обеспечения, перечислять, включать и переключать потоки модулей, а также устанавливать и обновлять пакеты из модуля.

РАЗДЕЛЫ

- Регистрация систем для поддержки Red Hat (и контрольный опрос)
- Объяснение и исследование пакетов программного обеспечения **RPM** (и контрольный опрос)
- Установка и обновление пакетов программного обеспечения с помощью **Yum** (и упражнения с пошаговыми инструкциями)
- Включение репозиториев программного обеспечения **Yum** (и упражнения с пошаговыми инструкциями)
- Управление потоками (**Streams**) модулей пакета (и упражнения с пошаговыми инструкциями)

ЛАБОРОТОРНАЯ РАБОТА

Установка и обновление пакетов программного обеспечения.

РЕГИСТРАЦИОННЫЕ СИСТЕМЫ ДЛЯ ПОДДЕРЖКИ RED HAT

ЗАДАЧИ

После завершения этого раздела вы сможете зарегистрировать систему в своей учетной записи Red Hat и назначить ей права на обновления программного обеспечения и службы поддержки с помощью управления подпиской Red Hat.

УПРАВЛЕНИЕ ПОДПИСКОЙ НА RED HAT

Red Hat Subscription Management предоставляет инструменты, которые можно использовать для предоставления машинам права подписки на продукты, позволяя администраторам получать обновления пакетов программного обеспечения и отслеживать информацию о контрактах на поддержку и подписках, используемых системами. Стандартные инструменты, такие как **PackageKit** и **yum**, могут получать пакеты программного обеспечения и обновления через сеть распространения контента, предоставляемую Red Hat.

Инструменты управления подпиской Red Hat выполняют четыре основных задачи:

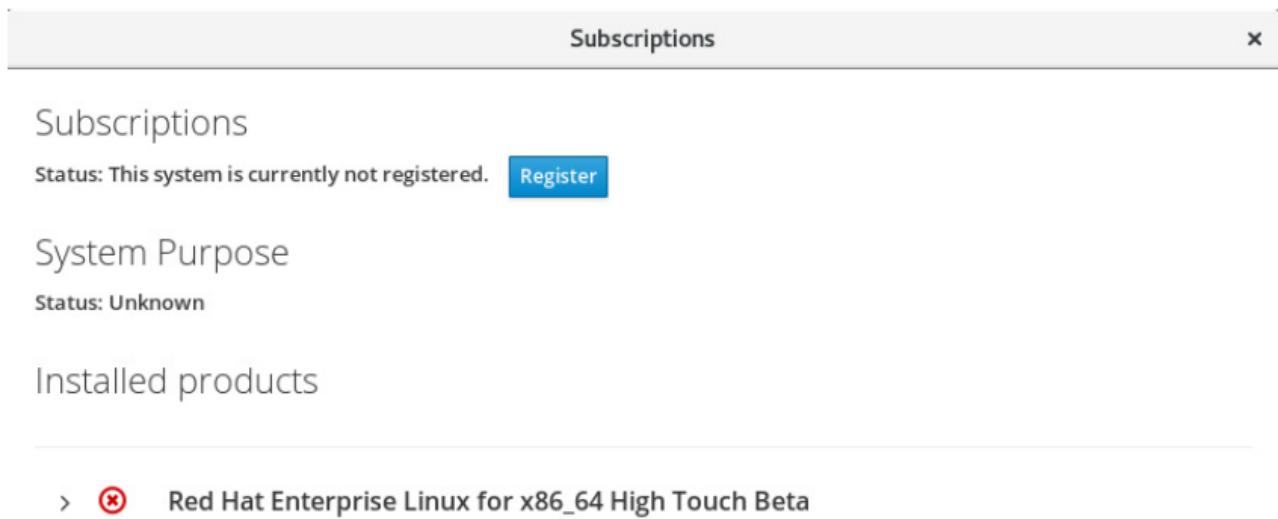
- Зарегистрируйте (**Register**) систему, чтобы связать эту систему с учетной записью Red Hat. Это позволяет Менеджеру подписки проводить уникальную инвентаризацию системы. Когда система больше не используется, ее можно отменить.
- Подпишитесь (**Subscribe**) на систему, чтобы получить право на получение обновлений для выбранных продуктов Red Hat. Подписки имеют определенные уровни поддержки, даты истечения срока действия и репозитории по умолчанию. Эти инструменты можно использовать либо для автоматического присоединения, либо для выбора определенного права. При изменении потребностей подписки могут быть удалены.
- Разрешить репозиториям (**Enable repositories**) предоставлять пакеты программного обеспечения. Несколько репозиториев включены по умолчанию с каждой подпиской, но другие репозитории, такие как обновления или исходный код, могут быть включены или отключены по мере необходимости.
- Просматривайте и отслеживайте (**Review** и **track**) доступные или использованные права. Информацию о подписке можно просмотреть локально в конкретной системе или, для учетной записи, на странице подписок клиентского портала Red Hat или в диспетчере активов подписки (**Subscription Asset Manager SAM**).

Регистрация системы

Существует несколько способов зарегистрировать систему на портале клиентов Red Hat. Существует графический интерфейс, к которому вы можете получить доступ с помощью приложения **GNOME** или через службу **веб-консоли**, и есть инструмент **командной строки**.

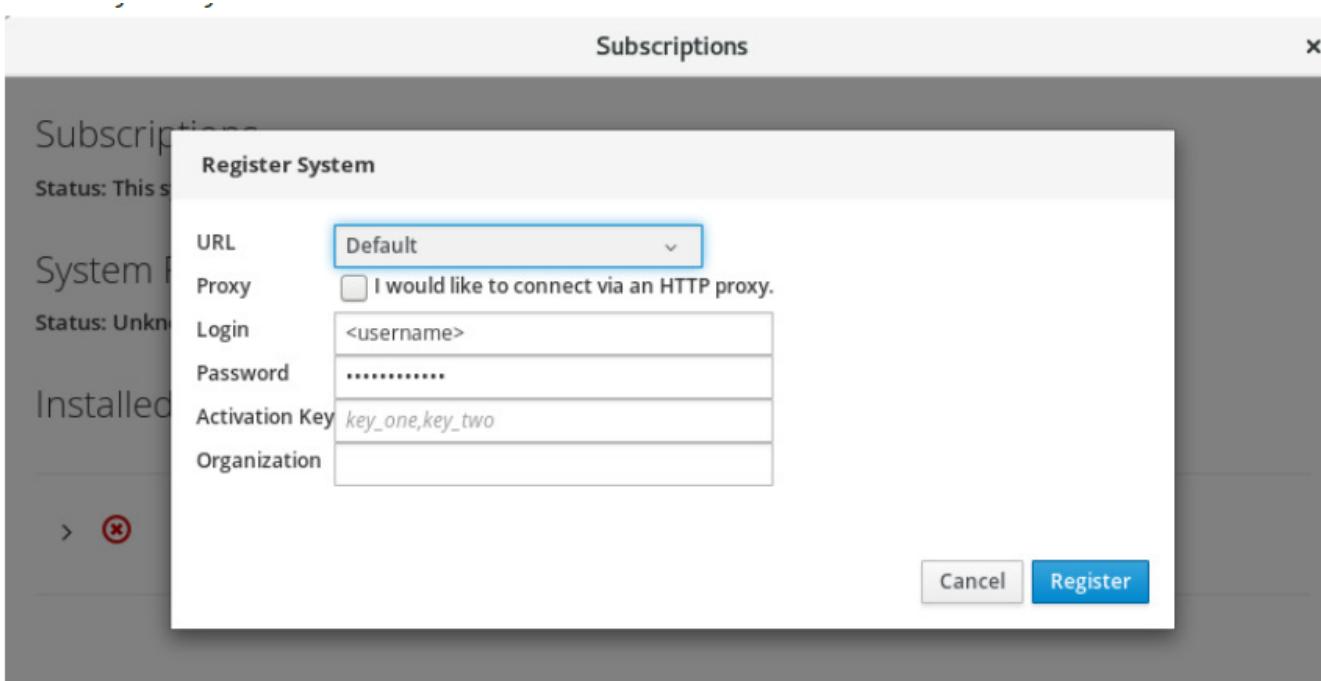
Чтобы зарегистрировать систему в приложении **GNOME**, запустите **Red Hat Subscription Manager**, выбрав Действия (**Activities**). Введите подписку (*subscription*) в поле Тип для поиска (*Type to search*) и щелкните Диспетчер подписок Red Hat (**Red Hat Subscription Manager**). При появлении запроса на аутентификацию введите соответствующий пароль. Откроется следующее окно подписок:

Рисунок 14.1: Главное окно Red Hat Subscription Manager



Чтобы зарегистрировать систему, нажмите кнопку «Зарегистрироваться (Register)» в окне «Подписки (Subscriptions)». Появится следующий диалог:

Рисунок 14.2: Диалог расположения службы и информации об учетной записи в Red Hat Subscription Manager



Это диалоговое окно регистрирует систему на сервере подписки. По умолчанию он регистрирует сервер на портале клиентов Red Hat. Введите логин (Login) и пароль (Password) для учетной записи **Red Hat Customer Portal**, в которой должна быть зарегистрирована система, и нажмите кнопку «Зарегистрироваться (Register)».

При регистрации к системе автоматически прикрепляется подписка, если она доступна.

После регистрации системы и назначения подписки закройте окно Подписки (**Subscriptions**). Теперь система правильно подписана и готова получать обновления или устанавливать новое программное обеспечение от Red Hat.

РЕГИСТРАЦИЯ ИЗ КОМАНДНОЙ СТРОКИ

Используйте диспетчер подписки **subscription-manager(8)** для регистрации системы без использования графической среды. Команда **subscription-manager** может автоматически присоединить систему к наиболее подходящим совместимым подпискам для системы.

- Зарегистрируйте систему в учетной записи Red Hat:

```
[user@host ~]$ subscription-manager register --username=yourusername --password=yourpassword
```

- Просмотреть доступные подписки:

```
[user@host ~]$ subscription-manager list --available | less
```

- Прикрепить подписку автоматически:

```
[user@host ~]$ subscription-manager attach -auto
```

- Либо прикрепите подписку с определенного пула из списка доступных подписок:

```
[user@host ~]$ subscription-manager attach --pool=poolID
```

- Просмотр использованных подписок:

```
[user@host ~]$ subscription-manager list --consumed
```

- Отменить регистрацию системы:

```
[user@host ~]$ subscription-manager unregister
```



ПРИМЕЧАНИЕ

subscription-manager также может использоваться в сочетании с ключами активации, позволяя регистрировать и назначать предопределенные подписки без использования имени **пользователя** или **пароля**. Этот метод регистрации может быть очень полезен для автоматической установки и развертывания. Ключи активации часто выдаются локальной службой управления подпиской, такой как **Subscription Asset Manager** или **Red Hat Satellite**, и в этом курсе подробно не рассматриваются.

ПРАВА (ENTITLEMENT) СЕРТИФИКАТОВ

Право (**entitlement**) - это подпись, прикрепленная к системе. Цифровые сертификаты используются для хранения текущей информации о правах в локальной системе. После регистрации сертификаты полномочий хранятся в директории **/etc/pki** и его подкаталогах.

- **/etc/pki/product** содержит сертификаты, указывающие, какие продукты Red Hat установлены в системе.
- **/etc/pki/consumer** содержит сертификаты, идентифицирующие учетную запись Red Hat, под которой зарегистрирована система.
- **/etc/pki/grantlement** содержит сертификаты, указывающие, какие подписи подключены к системе.

Сертификаты можно проверить напрямую с помощью утилиты **rct**, но инструменты диспетчера подписок **subscription-manager** предоставляют более простые способы проверки подписок, подключенных к системе.



ПРИМЕЧАНИЕ

Справочные страницы **man subscription-manager(8)** и **rct(8)**

Начать работу с Red Hat Subscription Management

<https://access.redhat.com/site/articles/433903>

КОНТРОЛЬНЫЙ ОПРОС

РЕГИСТРАЦИОННЫЕ СИСТЕМЫ ДЛЯ ПОДДЕРЖКИ RED HAT

Выберите правильный ответ на следующие вопросы:

- 1.** Какая команда используется для регистрации системы без использования графической среды?
 - a. **rct**
 - b. **subscription-manager**
 - c. **rpm**
 - d. **yum**

- 2.** Какой графический интерфейс используется для регистрации и подписки системы?
 - a. **PackageKit**
 - b. **gpk-application**
 - c. **Red Hat Subscription Manager**
 - d. **gnome-software**

- 3.** Какие задачи можно выполнять с помощью инструментов управления подпиской Red Hat?
 - a. Зарегистрируйте систему.
 - b. Подпишитесь на систему.
 - c. Включить репозитории.
 - d. Просматривайте и отслеживайте права.
 - e. Все вышеперечисленное.

РЕШЕНИЕ

РЕГИСТРАЦИОННЫЕ СИСТЕМЫ ДЛЯ ПОДДЕРЖКИ RED HAT

Выберите правильный ответ на следующие вопросы:

1. Какая команда используется для регистрации системы без использования графической среды?
 - a. `rct`
 - b. `subscription-manager`**
 - c. `rpm`
 - d. `yum`

2. Какой графический интерфейс используется для регистрации и подписки системы?
 - a. `PackageKit`
 - b. `gpk-application`
 - c. Red Hat Subscription Manager**
 - d. `gnome-software`

3. Какие задачи можно выполнять с помощью инструментов управления подпиской Red Hat?
 - a. Зарегистрируйте систему.
 - b. Подпишитесь на систему.
 - c. Включить репозитории.
 - d. Просматривайте и отслеживайте права.
 - e. Все вышеперечисленное.

ОБЪЯСНЕНИЕ И ИЗУЧЕНИЕ ПАКЕТОВ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ RPM

ЗАДАЧИ

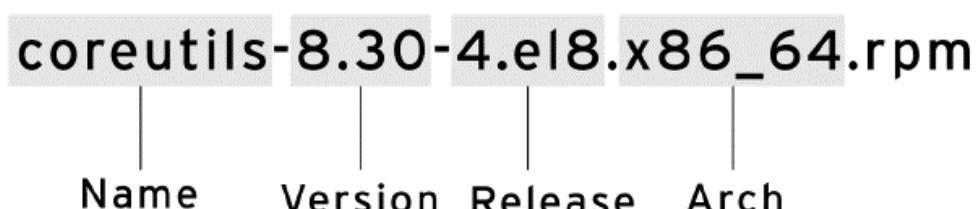
После завершения этого раздела вы сможете объяснить, как программное обеспечение предоставляется в виде пакетов **RPM**, и изучить пакеты, установленные в системе с помощью **Yum** и **RPM**.

ПАКЕТЫ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ И RPM

Менеджер пакетов **RPM**, первоначально разработанный Red Hat, предоставляет стандартный способ упаковки программного обеспечения для распространения. Управлять программным обеспечением в виде пакетов **RPM** намного проще, чем работать с программным обеспечением, которое было просто извлечено в файловую систему из архива. Он позволяет администраторам отслеживать, какие файлы были установлены программным пакетом, а какие необходимо удалить, если он был удален, и проверять наличие поддерживающих пакетов при его установке. Информация об установленных пакетах хранится в локальной базе данных **RPM** каждой системы. Все программное обеспечение, предоставляемое Red Hat для Red Hat Enterprise Linux, предоставляется в виде пакета **RPM**.

Имена файлов пакетов **RPM** состоят из четырех элементов (плюс суффикс **.rpm**): **name-version-release.architecture**:

Рисунок 14.3: Элементы имени файла RPM



- **NAME** - одно или несколько слов, описывающих содержимое (**coreutils**).
- **ВЕРСИЯ** - номер версии исходного программного обеспечения (8.30).
- **RELEASE** - это номер выпуска пакета, основанный на этой версии, и устанавливается упаковщиком, который может не быть исходным разработчиком программного обеспечения (4.el8).
- **ARCH** - это архитектура процессора, для работы на которой был скомпилирован пакет. **noarch** указывает, что содержимое этого пакета не зависит от архитектуры (в отличие от **x86_64** для 64-разрядной версии, **aarch64** для 64-разрядной **ARM** и т. д.).

Для установки пакетов из репозиториев требуется только имя пакета. Если существует несколько версий, устанавливается пакет с более высоким номером версии. Если существует несколько выпусков одной версии, устанавливается пакет с более высоким номером выпуска.

Каждый пакет **RPM** - это особый архив, состоящий из трех компонентов:

Файлы, установленные пакетом.

Информация о пакете (метаданные), такая как имя, версия, выпуск и архитектура; краткое содержание и описание пакета; требуется ли установка других пакетов; лицензирование; журнал изменений пакета; и другие подробности.

Сценарии, которые могут запускаться при установке, обновлении или удалении этого пакета или запускаться при установке, обновлении или удалении других пакетов.

Обычно поставщики программного обеспечения **подписывают пакеты RPM цифровой подписью** с помощью ключей **GPG** (Red Hat подписывает цифровой подписью все выпускаемые пакеты). Система **RPM** проверяет целостность пакета, подтверждая, что пакет был подписан соответствующим ключом **GPG**. Система **RPM** отказывается устанавливать пакет, если подпись GPG не совпадает.

ИЗУЧЕНИЕ ПАКЕТОВ RPM

Утилита **rpm** - это инструмент низкого уровня, который может получать информацию о содержимом файлов пакетов и установленных пакетах. По умолчанию он получает информацию из локальной базы данных установленных пакетов. Однако вы можете использовать опцию **-r**, чтобы указать, что вы хотите получить информацию о загруженном файле пакета. Возможно, вы захотите сделать это, чтобы проверить содержимое файла пакета перед его установкой.

Общая форма запроса:

- **rpm -q [select-options] [query-options]**

RPM queries (запросы): Общая информация об установленных пакетах

- **rpm -qa**: Список всех установленных пакетов
- **rpm -qf FILENAME**: Узнайте, какой пакет предоставляет *FILENAME*

```
[user@host ~]$ rpm -qf /etc/yum.repos.d  
redhat-release-8.0-0.39.el8.x86_64
```

RPM queries: Информация о конкретных пакетах

- **rpm -q**: Укажите, какая версия пакета установлена в данный момент.

```
[user@host ~]$ rpm -q yum  
yum-4.0.9.2-4.el8.noarch
```

- **rpm -qi**: Получить подробную информацию о пакете
- **rpm -ql**: Список файлов, установленных пакетом

```
[user@host ~]$ rpm -ql yum
```

```
/etc/yum.conf  
/etc/yum/pluginconf.d  
/etc/yum/protected.d  
/etc/yum/vars  
/usr/bin/yum  
/usr/share/man/man1/yum-aliases.1.gz  
/usr/share/man/man5/yum.conf.5.gz  
/usr/share/man/man8/yum-shell.8.gz  
/usr/share/man/man8/yum.8.gz
```

- **rpm -qc:** Перечислить только файлы конфигурации, установленные пакетом

```
[user@host ~]$ rpm -qc openssh-clients  
/etc/ssh/ssh_config  
/etc/ssh/ssh_config.d/05-redhat.conf
```

- **rpm -qd:** Перечислить только файлы документации, установленные пакетом

```
[user@host ~]$ rpm -qd openssh-clients  
/usr/share/man/man1/scp.1.gz  
/usr/share/man/man1/sftp.1.gz  
/usr/share/man/man1/ssh-add.1.gz  
/usr/share/man/man1/ssh-agent.1.gz  
/usr/share/man/man1/ssh-copy-id.1.gz  
/usr/share/man/man1/ssh-keyscan.1.gz  
/usr/share/man/man1/ssh.1.gz  
/usr/share/man/man5/ssh_config.5.gz  
/usr/share/man/man8/ssh-pkcs11-helper.8.gz
```

- **rpm -q --scripts:** список сценариев оболочки, которые запускаются до или после установки или удаления пакета.

```
[user@host ~]$ rpm -q --scripts openssh-server  
preinstall scriptlet (using /bin/sh):  
getent group sshd >/dev/null || groupadd -g 74 -r sshd || :  
getent passwd sshd >/dev/null || \  
    useradd -c "Privilege-separated SSH" -u 74 -g sshd \  
    -s /sbin/nologin -r -d /var/empty/sshd sshd 2>/dev/null || :  
postinstall scriptlet (using /bin/sh):  
  
if [ $1 -eq 1 ] ; then  
    # Initial installation  
    /usr/bin/systemctl preset sshd.service sshd.socket >/dev/null 2>&1 || :  
fi
```

```
preuninstall scriptlet (using /bin/sh):
```

```
if [ $1 -eq 0 ] ; then
    # Package removal, not upgrade
    /usr/bin/systemctl --no-reload disable sshd.service sshd.socket > /dev/
null 2>&1 || :
    /usr/bin/systemctl stop sshd.service sshd.socket > /dev/null 2>&1 || :
fi
```

```
postuninstall scriptlet (using /bin/sh):
```

```
/usr/bin/systemctl daemon-reload >/dev/null 2>&1 || :
if [ $1 -ge 1 ] ; then
    # Package upgrade, not uninstall
    /usr/bin/systemctl try-restart sshd.service >/dev/null 2>&1 || :
fi
```

- **rpm -q --changelog:** вывести список изменений для пакета

```
[user@host ~]$ rpm -q --changelog audit
```

```
* Wed Jan 09 2019 Steve Grubb <sgrubb@redhat.com> 3.0-0.10.20180831git0047a6c
resolves: rhbz#1655270] Message "audit: backlog limit exceeded" reported
- Fix annobin failure
```

```
* Fri Dec 07 2018 Steve Grubb <sgrubb@redhat.com> 3.0-0.8.20180831git0047a6c
resolves: rhbz#1639745 - build requires go-toolset-7 which is not available
resolves: rhbz#1643567 - service auditd stop exits prematurely
resolves: rhbz#1616428 - Update git snapshot of audit package
- Remove static libs subpackage
...output omitted...
```

Запрос локальных файлов пакета:

```
[user@host ~]$ ls -l wonderwidgets-1.0-4.x86_64.rpm
```

```
-rw-rw-r--. 1 user user 257 Mar 13 20:06 wonderwidgets-1.0-4.x86_64.rpm
```

```
[user@host ~]$ rpm -qlp wonderwidgets-1.0-4.x86_64.rpm
```

```
/etc/wonderwidgets.conf
```

```
/usr/bin/wonderwidgets
```

```
/usr/share/doc/wonderwidgets-1.0
```

```
/usr/share/doc/wonderwidgets-1.0/README.txt
```

УСТАНОВКА ПАКЕТОВ RPM

Команду **rpm** также можно использовать для установки пакета **RPM**, который вы скачали в свой локальный каталог.

```
[root@host ~]# rpm -ivh wonderwidgets-1.0-4.x86_64.rpm
Verifying...          ###### [100%]
Preparing...          ###### [100%]
Updating / installing...
    1:wonderwidgets-1.0-4  ###### [100%]
[root@host ~]#
```

Однако в следующем разделе этой главы мы обсудим более мощный инструмент для управления установкой **RPM** и обновлениями из командной строки, команду **yum**.



ПРЕДУПРЕЖДЕНИЕ

Будьте осторожны при установке пакетов от третьих лиц, не только из-за программного обеспечения, которое они могут установить, но потому, что пакет RPM может включать произвольные сценарии, которые запускаются от имени пользователя root как часть процесса установки.



ПРИМЕЧАНИЕ

Вы можете извлекать файлы из файла пакета **RPM** без установки пакета. Утилита **rpm2cpio** может передавать содержимое **RPM** в специальный инструмент архивирования, называемый **cpio**, который может извлекать все файлы или отдельные файлы.

Перенаправте вывод **rpm2cpio PACKAGEFILE.rpm** в **cpio -id**, чтобы извлечь все файлы, хранящиеся в пакете **RPM**. Деревья подкаталогов создаются по мере необходимости относительно текущего рабочего каталога.

```
[user@host tmp-extract]$ rpm2cpio wonderwidgets-1.0-4.x86_64.rpm | cpio -id
```

Отдельные файлы извлекаются путем указания пути к файлу:

```
[user@host ~]$ rpm2cpio wonderwidgets-1.0-4.x86_64.rpm | cpio -id "*txt"
11 blocks
[user@host ~]$ ls -l usr/share/doc/wonderwidgets-1.0/
total 4
-rw-r--r--. 1 user user 76 Feb 13 19:27 README.txt
```

СВОДКА КОМАНД ЗАПРОСА RPM

Установленные пакеты можно запросить напрямую с помощью команды **rpm**. Добавьте параметр **-q** для запроса файла пакета перед установкой.

КОМАНДА	ЗАДАЧА
rpm -qa	Список всех установленных пакетов RPM
rpm -q NAME	Отображение версии NAME , установленной в системе
rpm -qi NAME	Показать подробную информацию о пакете
rpm -ql NAME	Список всех файлов, включенных в пакет
rpm -qc NAME	Список файлов конфигурации, включенных в пакет
rpm -qd NAME	Список файлов документации, включенных в пакет
rpm -q --changelog NAME	Покажите краткое изложение причины выпуска нового пакета
rpm -q --scripts NAME	Отображение сценариев оболочки, запускаемых при установке, обновлении или удалении пакета



РЕКОМЕНДАЦИИ

Справочные страницы **man rpm(8)**, **rpm2cpio(8)**, **cpio(1)**, и **rpmkeys(8)**

УПРАЖНЕНИЯ С ПОШАГОВЫМИ ИНСТРУКЦИЯМИ

ОБЪЯСНЕНИЕ И ИЗУЧЕНИЕ ПАКЕТОВ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ RPM

В этом упражнении вы соберете информацию о пакете от третьей стороны, извлечете из него файлы для проверки, а затем установите его на сервер.

В РЕЗУЛЬТАТЕ

У вас должна быть возможность установить на сервере пакет, который не предоставляется репозиториями программного обеспечения.

ПРЕЖДЕ ЧЕМ ВЫ НАЧНЕТЕ

Войдите на рабочую станцию **workstation** как пользователь **student**, используя в качестве пароля слово **student**.

На рабочей станции **workstation** выполните скрипт **lab software-rpm start**. Сценарий, определяет, доступен ли хост, **servera**, в сети. Он также загружает пакет **rhcsa-script-1.0.0-1.noarch.rpm** в каталог **/home/student** на сервер **servera**.

```
[student@workstation ~]$ lab software-rpm start
```

1. Используйте команду **ssh** для входа на сервер **servera** как пользователь **student**.

```
[student@workstation ~]$ ssh student@servera  
[student@servera ~]$
```

2. Просмотрите информацию о пакете и список файлов в пакете **rhcsa-script-1.0.0-1.noarch.rpm**. Также просмотрите сценарий, который запускается при установке или удалении пакета.

- 2.1. Просмотрите информацию о пакете **rhcsa-script-1.0.0-1.noarch.rpm**.

```
[student@servera ~]$ rpm -q -p rhcsa-script-1.0.0-1.noarch.rpm -i  
Name           : rhcsa-script  
Version        : 1.0.0  
Release        : 1  
Architecture   : noarch  
Install Date   : (not installed)  
Group          : System  
Size           : 1056  
License         : GPL  
Signature       : (none)  
Source RPM     : rhcsa-script-1.0.0-1.src.rpm
```

```
Build Date      : Wed 06 Mar 2019 03:59:46 PM IST
Build Host      : foundation0.ilt.example.com
Relocations     : (not relocatable)
Packager        : Snehangshu Karmakar
URL             : http://example.com
Summary          : RHCSA Practice Script
Description      :
A RHCSA practice script.
The package changes the motd.
```

2.2. Перечислить файлы в пакете **rhcsa-script-1.0.0-1.noarch.rpm**.

```
[student@servera ~]$ rpm -q -p rhcsa-script-1.0.0-1.noarch.rpm -l
/opt/rhcsa-script/mymotd
```

2.3. Просмотрите сценарий, который запускается при установке или удалении **пакета rhcsa-script-1.0.0-1.noarch.rpm**.

```
[student@servera ~]$ rpm -q -p rhcsa-script-1.0.0-1.noarch.rpm --scripts
preinstall scriptlet (using /bin/sh):
if [ "$1" == "2" ]; then
    if [ -e /etc/motd.orig ]; then
        mv -f /etc/motd.orig /etc/motd
    fi
fi
postinstall scriptlet (using /bin/sh):
...output omitted...
```

3. Извлеките содержимое пакета **rhcsa-script-1.0.0-1.noarch.rpm** в каталог **/home/student**.

3.1. Используйте команды **rpm2cpio** и **cpio -tv** для вывода списка файлов в пакете **rhcsa-script-1.0.0-1.noarch.rpm**.

```
[student@servera ~]$ rpm2cpio rhcsa-script-1.0.0-1.noarch.rpm | cpio -tv
-rw-r--r--      1 root      root  1056 Mar  6 15:59      ./opt/rhcsa-script/mymotd
3 blocks
```

3.2. Извлеките все файлы из пакета **rhcsa-script-1.0.0-1.noarch.rpm** в каталог **/home/student**. Используйте команды **rpm2cpio** и **cpio -idv** для извлечения файлов и создания ведущих каталогов там, где это необходимо, в подробном режиме.

```
[student@servera ~]$ rpm2cpio rhcsa-script-1.0.0-1.noarch.rpm | cpio -idv
./opt/rhcsa-script/mymotd
3 blocks
```

3.3. Посмотреть список для проверки извлеченных файлов в каталоге /home/student/opt.

```
[student@servera ~]$ ls -lR opt
opt:
total 0
drwxrwxr-x. 2 student student 20 Mar 7 14:44 rhcsa-script

opt/rhcsa-script:
total 4
-rw-r--r--. 1 student student 1056 Mar 7 14:44 mymotd
```

- Установите пакет **rhcsa-script-1.0.0-1.noarch.rpm**. Используйте команду **sudo**, чтобы получить права суперпользователя для установки пакета.

4.1. Используйте команду **sudo rpm -ivh**, чтобы установить RPM-пакет **rhcsa-script-1.0.0-1.noarch.rpm**.

```
[student@servera ~]$ sudo rpm -ivh rhcsa-script-1.0.0-1.noarch.rpm
[sudo] password for student: student
Verifying...                                #####
Preparing...                                 #####
Updating / installing...
 1:rhcsa-script-1.0.0-1                      #####
[student@servera ~]$
```

4.2. Используйте команду **rpm**, чтобы убедиться, что пакет установлен.

```
[student@servera ~]$ rpm -q rhcsa-script
rhcsa-script-1.0.0-1.noarch
```

- Выход из сервера.

```
[student@servera ~]$ exit
logout
Connection to servera closed.
[student@workstation ~]$
```

Завершение

На рабочей станции **workstation**, запустите сценарий **lab software-rpm finish**, чтобы завершить данное упражнение. Этот скрипт удаляет все пакеты, установленные на сервере **servera** во время упражнения.

```
[student@workstation ~]$ lab software-rpm finish
```

На этом пошаговое упражнение завершено.

УСТАНОВКА И ОБНОВЛЕНИЕ ПАКЕТОВ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ С ПОМОЩЬЮ УТИЛИТЫ YUM

ЗАДАЧИ

После завершения этого раздела вы сможете находить, устанавливать и обновлять пакеты программного обеспечения с помощью команды **yum**.

УПРАВЛЕНИЕ ПАКЕТАМИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ С ПОМОЩЬЮ YUM

Команду **rpm** низкого уровня можно использовать для установки пакетов, но она не предназначена для работы с репозиториями пакетов или автоматического разрешения зависимостей из нескольких источников.

Yum разработан как лучшая система для управления установкой и обновлениями программного обеспечения на основе RPM. Команда **yum** позволяет устанавливать, обновлять, удалять и получать информацию о программных пакетах и их зависимостях. Вы можете получить историю выполненных транзакций и работать с несколькими репозиториями Red Hat и стороннего программного обеспечения.

Поиск программного обеспечения с помощью Yum

- **yum help** отображает информацию об использовании.
- **yum list** отображает установленные и доступные пакеты

```
[user@host ~]$ yum list 'http*'
Available Packages
http-parser.i686          2.8.0-2.el8           rhel8-appstream
http-parser.x86_64          2.8.0-2.el8           rhel8-appstream
httpcomponents-client.noarch 4.5.5-4.module+el8+2452+b359bfcd rhel8-appstream
httpcomponents-core.noarch   4.4.10-3.module+el8+2452+b359bfcd rhel8-appstream
httpd.x86_64                2.4.37-7.module+el8+2443+605475b7 rhel8-appstream
httpd-devel.x86_64          2.4.37-7.module+el8+2443+605475b7 rhel8-appstream
httpd-filesystem.noarch     2.4.37-7.module+el8+2443+605475b7 rhel8-appstream
httpd-manual.noarch         2.4.37-7.module+el8+2443+605475b7 rhel8-appstream
httpd-tools.x86_64          2.4.37-7.module+el8+2443+605475b7 rhel8-appstream
```

- **yum search KEYWORD** перечисляет пакеты по ключевым словам, найденным в полях имени и сводки.

Для поиска пакетов, которые содержат «**web server**» в полях имени, сводки и описания, используйте команду **search all**:

```
[user@host ~]$ yum search all 'web server'
```

```
===== Summary & Description Matched: web server =====
pcp-pmda-weblog.x86_64 : Performance Co-Pilot (PCP) metrics from web server logs
nginx.x86_64 : A high performance web server and reverse proxy server
===== Summary Matched: web server =====
libcurl.x86_64 : A library for getting files from web servers
libcurl.i686 : A library for getting files from web servers
libcurl.x86_64 : A library for getting files from web servers
===== Description Matched: web server =====
httpd.x86_64 : Apache HTTP Server
git-instaweb.x86_64 : Repository browser in gitweb
...output omitted...
```

- **yum info PACKAGE NAME** возвращает подробную информацию о пакете, включая дисковое пространство, необходимое для установки.

Чтобы получить информацию о HTTP-сервере Apache:

```
[user@host ~]$ yum info httpd
Available Packages
Name        : httpd
Version     : 2.4.37
Release     : 7.module+el8+2443+605475b7
Arch        : x86_64
Size        : 1.4 M
Source      : httpd-2.4.37-7.module+el8+2443+605475b7.src.rpm
Repo        : rhel8-appstream
Summary     : Apache HTTP Server
URL         : https://httpd.apache.org/
License      : ASL 2.0
Description  : The Apache HTTP Server is a powerful, efficient, and extensible
               : web server.
```

- **yum provides PATHNAME** отображает пакеты, соответствующие указанному имени пути (которые часто включают подстановочные знаки).

Чтобы найти пакеты, которые предоставляют каталог **/var/www/html**, используйте:

```
[user@host ~]$ yum provides /var/www/html
httpd-filesystem-2.4.37-7.module+el8+2443+605475b7.noarch : The basic directory
layout for the Apache HTTP server
Repo        : rhel8-appstream
Matched from:
Filename   : /var/www/html
```

Установка и удаление программного обеспечения с помощью yum

- **yum install *PACKAGENAME*** получает и устанавливает пакет программного обеспечения, включая все зависимости.

```
[user@host ~]$ yum install httpd
Dependencies resolved.

=====
Package           Arch    Version      Repository      Size
=====
Installing:
httpd            x86_64  2.4.37-7.module...
Installing dependencies:
apr               x86_64  1.6.3-8.el8        rhel8-appstream 125 k
apr-util          x86_64  1.6.1-6.el8        rhel8-appstream 105 k
...output omitted...
Transaction Summary
=====
Install 9 Packages

Total download size: 2.0 M
Installed size: 5.4 M
Is this ok [y/N]: y
Downloading Packages:
(1/9): apr-util-bdb-1.6.1-6.el8.x86_64.rpm      464 kB/s | 25 kB   00:00
(2/9): apr-1.6.3-8.el8.x86_64.rpm                1.9 MB/s | 125 kB  00:00
(3/9): apr-util-1.6.1-6.el8.x86_64.rpm            1.3 MB/s | 105 kB  00:00
...output omitted...
Total                                         8.6 MB/s | 2.0 MB  00:00

Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing          :                                1/1
  Installing         : apr-1.6.3-8.el8.x86_64          1/9
  Running scriptlet: apr-1.6.3-8.el8.x86_64          1/9
  Installing         : apr-util-bdb-1.6.1-6.el8.x86_64 2/9
...output omitted...
Installed:
  httpd-2.4.37-7.module+el8+2443+605475b7.x86_64 apr-util-bdb-1.6.1-6.el8.x86_64
  apr-util-openssl-1.6.1-6.el8.x86_64                 apr-1.6.3-8.el8.x86_64
...output omitted...
Complete!
```

- **yum update *PACKAGENAME*** получает и устанавливает более новую версию указанного пакета, включая все зависимости. Обычно процесс пытается сохранить файлы конфигурации на месте, но в некоторых случаях они могут быть переименованы, если упаковщик считает, что старые установки не будут работать после обновления. Если **PACKAGENAME** не указан, команда устанавливает все соответствующие обновления.

```
[user@host ~]$ sudo yum update
```

Поскольку новое ядро можно протестировать только при загрузке с этим ядром, пакет специально разработан таким образом, чтобы можно было установить сразу несколько версий. Если новое ядро не загружается, старое ядро все еще доступно. Использование **yum update kernel** фактически установит новое ядро. Файлы конфигурации содержат список пакетов, которые нужно всегда устанавливать, даже если администратор запрашивает обновление.



ПРИМЕЧАНИЕ

Используйте **yum list kernel**, чтобы вывести список всех установленных и доступных ядер. Чтобы просмотреть текущее работающее ядро, используйте команду **uname**. Параметр **-r** показывает только версию и выпуск ядра, а параметр **-a** показывает выпуск ядра и дополнительную информацию.

```
[user@host ~]$ yum list kernel
```

Installed Packages

kernel.x86_64	4.18.0-60.el8	@anaconda
kernel.x86_64	4.18.0-67.el8	@rhel-8-for-x86_64-baseos-htb-rpms

```
[user@host ~]$ uname -r
4.18.0-60.el8.x86_64
[user@host ~]$ uname -a
Linux host.lab.example.com 4.18.0-60.el8.x86_64 #1 SMP Fri Jan 11 19:08:11 UTC
2019 x86_64 x86_64 x86_64 GNU/Linux
```

- **yum remove PACKAGE NAME** удаляет установленный программный пакет, включая все поддерживаемые пакеты.

```
[user@host ~]$ sudo yum remove httpd
```



ПРЕДУПРЕЖДЕНИЕ

Команда **yum remove** удаляет перечисленные пакеты и любой пакет, который требует удаления пакетов (и пакеты, которым требуются эти пакеты, и т.д.). Это может привести к неожиданному удалению пакетов, поэтому внимательно просмотрите список удаляемых пакетов.

Установка и удаление групп программного обеспечения с помощью yum

- В **yum** также есть концепция групп (*groups*), которые представляют собой наборы связанного программного обеспечения, установленного вместе для определенной цели. В Red Hat Enterprise Linux 8 есть два типа групп. Обычные группы - это наборы пакетов. Группы окружения (*Environment groups*) - это наборы обычных групп. Пакеты или группы, предоставляемые группой, могут быть обязательными (они должны быть установлены, если группа установлена), по умолчанию (обычно устанавливается, если группа установлена) или необязательными (не устанавливаются при установке группы, если специально не запрашивается).

Как и **yum list**, команда **yum group list** показывает имена установленных и доступных групп.

```
[user@host ~]$ yum group list
Available Environment Groups:
  Server with GUI
  Minimal Install
  Server
...output omitted...
Available Groups:
  Container Management
  .NET Core Development
  RPM Development Tools
...output omitted...
```

Некоторые группы обычно устанавливаются через группы среды и по умолчанию скрыты. Перечислите эти скрытые группы с помощью команды **yum group list hidden**.

- **yum group info** отображает информацию о группе. Он включает список обязательных, стандартных и дополнительных имен пакетов.

```
[user@host ~]$ yum group info "RPM Development Tools"
Group: RPM Development Tools
Description: These tools include core development tools such rpmbuild.
Mandatory Packages:
  redhat-rpm-config
  rpm-build
Default Packages:
  rpmdevtools
Optional Packages:
  rpmlint
```

- **yum group install** устанавливает группу, которая устанавливает свои обязательные пакеты и пакеты по умолчанию, а также пакеты, от которых они зависят.

```
[user@host ~]$ sudo yum group install "RPM Development Tools"
```

...output omitted...

Installing Groups:

```
RPM Development Tools
```

```
Transaction Summary
```

```
=====  
Install 64 Packages
```

```
Total download size: 21 M
```

```
Installed size: 62 M
```

```
Is this ok [y/N]: y
```

...output omitted...



ВАЖНО

Поведение групп **Yum** изменилось, начиная с Red Hat Enterprise Linux 7. В RHEL 7 и более поздних версиях группы рассматриваются как объекты и отслеживаются системой. Если установленная группа обновляется, и новые обязательные пакеты или пакеты по умолчанию были добавлены в группу репозиторием Yum, эти новые пакеты устанавливаются при обновлении.

RHEL 6 и более ранние версии считают группу установленной, если все ее обязательные пакеты были установлены, или если у нее нет обязательных пакетов, или если в группе установлены какие-либо пакеты по умолчанию или дополнительные пакеты. Начиная с RHEL 7, группа считается установленной, только если для ее установки использовалась команда **yum group install**. Команду **yum group mark install GROUPNAME** можно использовать, чтобы пометить группу как установленную, и любые отсутствующие пакеты и их зависимости будут установлены при следующем обновлении.

Наконец, в RHEL 6 и более ранних версиях команды **yum group** не было двух слов. Другими словами, в RHEL 6 существовал список команд **yum grouplist**, но не было эквивалентных команд **yum group list** RHEL 7 и RHEL 8.

Просмотр истории транзакций

- Все транзакции установки и удаления регистрируются в **/var/log/dnf.rpm.log**.

```
[user@host ~]$ tail -5 /var/log/dnf.rpm.log
```

```
2019-02-26T18:27:00Z SUBDEBUG Installed: rpm-build-4.14.2-9.el8.x86_64
```

```
2019-02-26T18:27:01Z SUBDEBUG Installed: rpm-build-4.14.2-9.el8.x86_64
```

```
2019-02-26T18:27:01Z SUBDEBUG Installed: rpmdevtools-8.10-7.el8.noarch
```

```
2019-02-26T18:27:01Z SUBDEBUG Installed: rpmdevtools-8.10-7.el8.noarch
```

2019-02-26T18:38:40Z INFO --- logging initialized ---

- **yum history** отображает сводку по транзакциям установки и удаления.

[user@host ~]\$ sudo yum history

ID	Command line	Date and time	Action(s)	Altered
7	group install RPM Develo	2019-02-26 13:26	Install	65
6	update kernel	2019-02-26 11:41	Install	4
5	install httpd	2019-02-25 14:31	Install	9
4	-y install @base firewal	2019-02-04 11:27	Install	127 EE
3	-C -y remove firewalld -	2019-01-16 13:12	Removed	11 EE
2	-C -y remove linux-firmw	2019-01-16 13:12	Removed	1
1		2019-01-16 13:05	Install	447 EE

- **history undo** отменяет транзакцию.

[user@host ~]\$ sudo yum history undo 5

Undoing transaction 7, from Tue 26 Feb 2019 10:40:32 AM EST

```
Install apr-1.6.3-8.el8.x86_64          @rhel8-appstream
Install apr-util-1.6.1-6.el8.x86_64      @rhel8-appstream
Install apr-util-bdb-1.6.1-6.el8.x86_64  @rhel8-appstream
Install apr-util-openssl-1.6.1-6.el8.x86_64 @rhel8-appstream
Install httpd-2.4.37-7.module+el8+2443+605475b7.x86_64 @rhel8-appstream
...output omitted...
```

РЕЗЮМЕ КОМАНД YUM

Пакеты могут быть найдены, установлены, обновлены и удалены по имени или по группам пакетов.

ЗАДАЧА:	КОМАНДА:
Список установленных и доступных пакетов по имени	yum list [NAME-PATTERN]
Список установленных и доступных групп	yum group list
Искать пакет по ключевому слову	yum search KEYWORD
Показать детали пакета	yum info PACKAGE NAME
Установить пакет	yum install PACKAGE NAME
Установить группу пакетов	yum group install GROUP NAME
Обновить все пакеты	yum update

Удалить пакет	yum remove PACKAGENAME
Показать историю транзакций	yum history



РЕКОМЕНДАЦИИ

Справочные страницы **man yum(1)** и **yum.conf(5)**

Примечания к выпуску Red Hat Enterprise Linux 8.0 Beta

https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/8-beta/html-single/8.0_beta_release_notes/index

УПРАЖНЕНИЯ С ПОШАГОВЫМИ ИНСТРУКЦИЯМИ

УСТАНОВКА И ОБНОВЛЕНИЕ ПАКЕТОВ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ С ПОМОЩЬЮ КОМАНДЫ YUM

В этом упражнении вы установите и удалите пакеты и группы пакетов.

В РЕЗУЛЬТАТЕ

У вас должна быть возможность устанавливать и удалять пакеты с зависимостями.

ПРЕЖДЕ ЧЕМ ВЫ НАЧНЕТЕ

Войдите на рабочую станцию **workstation** как пользователь **student**, используя в качестве пароля слово **student**.

На рабочей станции **workstation** выполните скрипт **lab software-yum start**. Команда запускает сценарий, который определяет, доступен ли хост, **servera**, в сети.

```
[student@workstation ~]$ lab software-yum start
```

1. Используйте команду **ssh** для входа на сервер **servera**, как пользователь **student**. Системы настроены на использование ключей **SSH** для аутентификации, поэтому пароль для входа на сервер не требуется.

```
[student@workstation ~]$ ssh student@servera  
[student@servera ~]$
```

2. Используйте **su -** для переключения на пользователя **root** в приглашении оболочки.

```
[student@servera ~]$ su -  
Password: redhat  
[root@servera ~]#
```

3. Найдите определённый пакет.

- 3.1. При попытке выполнить команду **guile**. Вы должны обнаружить, что он не установлен.

```
[root@servera ~]# guile  
-bash: guile: command not found
```

3.2. Используйте команду **yum search** для поиска пакетов, в названии или кратном описание которых есть слово **guile**.

```
[root@servera ~]# yum search guile
=====
Name Exactly Matched: guile
guile.i686 : A GNU implementation of Scheme for application extensibility
guile.x86_64 : A GNU implementation of Scheme for application extensibility
```

3.3. Используйте команду **yum info**, чтобы узнать больше о пакете **guile**.

```
[root@servera ~]# yum info guile
Available Packages
Name           : guile
Epoch          : 5
Version        : 2.0.14
Release        : 7.el8
...output omitted...
```

4. Используйте команду **yum install -y**, чтобы установить пакет **guile**.

```
[root@servera ~]# yum install -y guile
...output omitted...
Dependencies resolved.

=====
Package      Arch    Version       Repository      Size
=====
Installing:
  guile       x86_64  5:2.0.14-7.el8   rhel-8.0-for-x86_64-appstream-rpms 3.5 M
Installing dependencies:
  gc          x86_64  7.6.4-3.el8     rhel-8.0-for-x86_64-appstream-rpms 109 k
  libatomic_ops x86_64  7.6.2-3.el8     rhel-8.0-for-x86_64-appstream-rpms 38 k
  libtool-ltdl x86_64  2.4.6-25.el8    rhel-8.0-for-x86_64-baseos-rpms    58 k

Transaction Summary
=====
...output omitted...
```

5. Удалить пакеты.

5.1. Используйте команду **yum remove**, чтобы удалить пакет **guile**, но ответьте «нет(no)» на запрос. Сколько пакетов будет удалено?

```
[root@servera ~]# yum remove guile
...output omitted...
Dependencies resolved.
```

Package	Arch	Version	Repository	Size
===== Removing:				
guile	x86_64	5:2.0.14-7.el8	@rhel-8.0-for-x86_64-appstream-rpms	12 M
Removing unused dependencies:				
gc	x86_64	7.6.4-3.el8	@rhel-8.0-for-x86_64-appstream-rpms	221 k
libatomic_ops	x86_64	7.6.2-3.el8	@rhel-8.0-for-x86_64-appstream-rpms	75 k
libtool-ltdl	x86_64	2.4.6-25.el8	@rhel-8.0-for-x86_64-baseos-rpms	69 k
Transaction Summary				
Remove 4 Packages				
Freed space: 12 M				
Is this ok [y/N]: n				

- 5.2.** Используйте команду **yum remove**, чтобы удалить пакет **gc**, но при появлении запроса ответьте **нет(no)**. Сколько пакетов будет удалено?

```
[root@servera ~]# yum remove gc
...output omitted...
Dependencies resolved.

=====
      Package           Arch   Version        Repository      Size
=====
Removing:
  gc                  x86_64  7.6.4-3.el8    @rhel-8.0-for-x86_64-appstream-rpms 221 k
Removing dependent packages:
  guile              x86_64  5:2.0.14-7.el8  @rhel-8.0-for-x86_64-appstream-rpms 12 M
Removing unused dependencies:
  libatomic_ops       x86_64  7.6.2-3.el8    @rhel-8.0-for-x86_64-appstream-rpms 75 k
  libtool-ltdl        x86_64  2.4.6-25.el8   @rhel-8.0-for-x86_64-baseos-rpms   69 k

Transaction Summary

=====
Remove 4 Packages

Freed space: 12 M
Is this ok [y/N]: n
```

- 6.** Соберите информацию о группе компонентов «Инструменты разработки RPM (**RPM Development Tools**)» и установите ее на сервере.

- 6.1.** Используйте команду **yum group list**, чтобы вывести список всех доступных групп компонентов.

```
[root@servera ~]# yum group list
```

6.2. Используйте команду **yum group info**, чтобы узнать больше о группе компонентов **RPM Development Tools**, включая список включенных пакетов.

```
[root@servera ~]# yum group info "RPM Development Tools"
Group: RPM Development Tools
Description: These tools include core development tools such rpmbuild.
Mandatory Packages:
  redhat-rpm-config
  rpm-build
Default Packages:
  rpmdevtools
Optional Packages:
  rpmlint
```

6.3. Используйте команду **yum group install**, чтобы установить группу компонентов **RPM Development Tools**.

```
[root@servera ~]# yum group install "RPM Development Tools"
Dependencies resolved.

=====
Package           Arch    Version        Repository      Size
=====
Installing group/module packages:
  redhat-rpm-config
    noarch 115-1.el8      rhel-8.0-for-x86_64-appstream-rpms 82 k
  rpm-build       x86_64 4.14.2-9.el8   rhel-8.0-for-x86_64-appstream-rpms 166 k
Installing dependencies:
  dwz             x86_64 0.12-9.el8   rhel-8.0-for-x86_64-appstream-rpm 109 k
  efi-srpm-macros noarch 3-2.el8     rhel-8.0-for-x86_64-appstream-rpm 22 k
...output omitted...

Transaction Summary
=====
Install 60 Packages

Total download size: 17 M
Installed size: 50 M
Is this ok [y/N]: y
...output omitted...
Installing      : perl-Exporter-5.72-396.el8.noarch          1/60
Installing      : perl-libs-4:5.26.3-416.el8.x86_64          2/60
Installing      : perl-Carp-1.42-396.el8.noarch          3/60
...output omitted...
Verifying       : dwz-0.12-9.el8.x86_64                  1/60
Verifying       : efi-srpm-macros-3-2.el8.noarch          2/60
Verifying       : gdb-headless-8.2-5.el8.x86_64          3/60
...output omitted...
Installed:
  redhat-rpm-config-115-1.el8.noarch
  rpm-build-4.14.2-9.el8.x86_64
  rpmdevtools-8.10-7.el8.noarch
```

...output omitted...

Complete!

7. Изучите историю и варианты отмены команды **yum**.

7.1. Используйте команду **yum history**, чтобы отобразить недавнюю историю выполнения команды **yum**.

```
[root@servera ~]# yum history
ID      | Command line           | Date and time   | Action(s)    | Altered
-----
 6 | group install RPM Develo | 2019-02-26 17:11 | Install      | 61
 5 | install -y guile       | 2019-02-26 17:05 | Install      | 4
 4 | -y install @base firewal | 2019-02-04 11:27 | Install      | 127 EE
 3 | -C -y remove firewalld - | 2019-01-16 13:12 | Removed      | 11 EE
 2 | -C -y remove linux-firmw | 2019-01-16 13:12 | Removed      | 1
 1 |                         | 2019-01-16 13:05 | Install      | 447 EE
```

7.2. Используйте команду **yum history info**, чтобы подтвердить, что последней транзакцией является установка группы.

```
[root@servera ~]# yum history info 6
Transaction ID : 6
Begin time      : Tue 26 Feb 2019 05:11:25 PM EST
Begin rpmbdb    : 563:bf48c46156982a78e290795400482694072f5ebb
End time        : Tue 26 Feb 2019 05:11:33 PM EST (8 seconds)
End rpmbdb      : 623:bf25b424ccf451dd0a6e674fb48e497e66636203
User            : Student User <student>
Return-Code     : Success
Releasever     : 8
Command Line    : group install RPM Development Tools
Packages Altered:
  Install dwz-0.12-9.el8.x86_64          @rhel-8.0-for-x86_64-appstream-rpms
  Install efi-srpm-macros-3-2.el8.noarch @rhel-8.0-for-x86_64-appstream-rpms
...output omitted...
```

7.3. Используйте команду **yum history undo**, чтобы удалить набор пакетов, которые были установлены при установке пакета **guile**.

```
[root@servera ~]# yum history undo 5
```

8. Выходите из системы **servera**.

```
[root@servera ~]# exit
logout
[student@servera ~]$ exit
Connection to servera closed.
[student@workstation ~]$
```

Завершение

На рабочей станции **workstation**, запустите скрипт **lab software-yum finish**, чтобы завершить упражнение.

```
[student@workstation ~]$ lab software-yum finish
```

На этом пошаговое упражнение завершено.

ВКЛЮЧЕНИЕ РЕПОЗИТОРИЙ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ YUM

ЗАДАЧИ

После завершения этого раздела вы сможете включать и отключать использование сервером Red Hat или сторонних репозиториев **Yum**.

ВКЛЮЧЕНИЕ РЕПОЗИТОРИЙ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ RED HAT

При регистрации системы в службе управления подписками автоматически настраивается доступ к репозиториям программного обеспечения на основе прикрепленных подписок. Чтобы просмотреть все доступные репозитории выполните команду:

```
[user@host ~]$ yum repolist all
Loaded plugins: langpacks
repo id                                repo name
      status
rhel-8-server-debug-rpms/8Server/x86_64    Red Hat Enterprise Linux 8 Server
(Debug RPMs)  disabled
rhel-8-server-rpms/8Server/x86_64           Red Hat Enterprise Linux 8 Server
(RPMs)        enabled: 5,071
rhel-8-server-source-rpms/8Server/x86_64    Red Hat Enterprise Linux 8 Server
(Source RPMs)  disabled
repolist: 5,071
```

Команду **yum-config-manager** можно использовать для включения или отключения репозиториев. Чтобы включить репозиторий, команда устанавливает для параметра **enabled значение 1**. Например, следующая команда включает репозиторий **rhel-8-server-debug-rpms**:

```
[user@host ~]$ yum-config-manager --enable rhel-8-server-debug-rpms
Loaded plugins: langpacks
=====
[repo: rhel-8-server-debug-rpms]
=====
[rhel-8-server-debug-rpms]
async = True
bandwidth = 0
base_persistdir = /var/lib/yum/repos/x86_64/8Server
baseurl = https://cdn.redhat.com/content/dist/rhel/server/8/8Server/x86_64/debug
cache = 0
cachedir = /var/cache/yum/x86_64/8Server/rhel-8-server-debug-rpms
check_config_file_age = True
cost = 1000
deltarpm_percentage =
enabled = 1
...output omitted...
```

Источники, не относящиеся к Red Hat, предоставляют программное обеспечение через сторонние репозитории, к которым можно получить доступ с помощью команды **yum** с веб-сайта, **FTP-сервера** или **локальной файловой системы**. Например, Adobe предоставляет часть своего программного обеспечения для Linux через репозиторий Yum. В классе Red Hat на сервере класса **content.example.com** размещаются **репозитории Yum**.

Чтобы включить поддержку нового стороннего репозитория, создайте файл в каталоге **/etc/yum.repos.d/**. Файлы конфигурации репозитория должны иметь расширение **.repo**. Определение репозитория содержит **URL-адрес репозитория, имя, использование GPG** для проверки подписей пакетов и, если да, **URL-адрес, указывающий на доверенный ключ GPG**.

Создание репозиториев Yum

Создайте репозитории **Yum** с помощью команды **yum-config-manager**. Следующая команда создает файл с именем **/etc/yum.repos.d/dl.fedoraproject.org_pub_epel_8_x86_64.repo** с показанными выходными данными.

```
[user@host ~]$ yum-config-manager --add-repo="http://dl.fedoraproject.org/pub/epel/8/x86_64/"
Loaded plugins: langpacks
adding repo from: http://dl.fedoraproject.org/pub/epel/8/x86_64/

[dl.fedoraproject.org_pub_epel_8_x86_64_]
name=added from: http://dl.fedoraproject.org/pub/epel/8/x86_64/
baseurl=http://dl.fedoraproject.org/pub/epel/8/x86_64/
enabled=1
```

Измените этот файл, чтобы указать настраиваемые значения и расположение ключа GPG. Ключи хранятся в различных местах на сайте удаленного репозитория, например, **http://dl.fedoraproject.org/pub/epel/RPM-GPG-KEY-EPEL-8**. Администраторам следует загрузить ключ в локальный файл, а не разрешать **yum** извлекать ключ из внешнего источника. Например:

```
[EPEL]
name=EPEL 8
baseurl=http://dl.fedoraproject.org/pub/epel/8/x86_64/
enabled=1
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-EPEL-8
```

Пакеты конфигурации RPM для локальных репозиториев

Некоторые репозитории предоставляют **файл конфигурации и открытый ключ GPG** как **часть пакета RPM**, который можно загрузить и установить с помощью команды **yum localinstall**. Например, добровольный проект под названием **Extra Packages for Enterprise Linux (EPEL)**

предоставляет программное обеспечение, не поддерживаемое Red Hat, но совместимое с Red Hat Enterprise Linux.

Следующая команда устанавливает репозиторий **Red Hat Enterprise Linux 8 EPEL**:

```
[user@host ~]$ rpm --import http://dl.fedoraproject.org/pub/epel/RPM-GPG-KEYEPEL-8
[user@host ~]$ yum install http://dl.fedoraproject.org/pub/epel/8/x86\_64/e/epelrelease-8-2.noarch.rpm
```

Файлы конфигурации часто содержат несколько ссылок на репозиторий в одном файле. Каждая ссылка на репозиторий начинается с имени из одного слова в квадратных скобках.

```
[user@host ~]$ cat /etc/yum.repos.d/epel.repo
[epel]
name=Extra Packages for Enterprise Linux 8 - $basearch
#baseurl=http://download.fedoraproject.org/pub/epel/8/$basearch
mirrorlist=https://mirrors.fedoraproject.org/metalink?repo=epel-8&arch=$basearch
failovermethod=priority
enabled=1
gpgcheck=0
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-EPEL-8

[epel-debuginfo]
name=Extra Packages for Enterprise Linux 8 - $basearch - Debug
#baseurl=http://download.fedoraproject.org/pub/epel/8/$basearch/debug
mirrorlist=https://mirrors.fedoraproject.org/metalink?repo=epel-debug-8&arch=
$basearch
failovermethod=priority
enabled=0
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-EPEL-8
gpgcheck=1

[epel-source]
name=Extra Packages for Enterprise Linux 8 - $basearch - Source
#baseurl=http://download.fedoraproject.org/pub/epel/8/SRPMS
mirrorlist=https://mirrors.fedoraproject.org/metalink?repo=epel-source-8&arch=
$basearch
failovermethod=priority
enabled=0
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-EPEL-8
gpgcheck=1
```

Чтобы определить репозиторий, но не искать в нем по умолчанию, вставьте параметр **enabled=0**. Репозитории можно постоянно включать и отключать с помощью команды **yum-config-manager** или временно с помощью параметров команды **yum --enablerepo=PATTERN** и **--disablerepo=PATTERN**.



ПРЕДУПРЕЖДЕНИЕ

Перед установкой подпсанных пакетов **установите ключ RPM GPG**. Он подтверждает, что пакеты принадлежат ключу, который был импортирован. В противном случае команда **yum** завершится ошибкой из-за отсутствия ключа. Параметр **--nogpgcheck** можно использовать для игнорирования **отсутствующих ключей GPG**, но это может привести к установке поддельных или небезопасных пакетов в системе, что потенциально может поставить под угрозу ее безопасность.



РЕКОМЕНДАЦИИ

Дополнительная информация может быть доступна в разделе о настройке репозиториев Yum и Yum в Руководстве системного администратора Red Hat Enterprise Linux для Red Hat Enterprise Linux 8, которое можно найти по адресу

<https://access.redhat.com/documentation/>

Справочные страницы **man yum(1)**, **yum.conf(5)**, и **yum-config-manager(1)**

УПРАЖНЕНИЯ С ПОШАГОВЫМИ ИНСТРУКЦИЯМИ

В этом упражнении вы настроите свой сервер для получения пакетов из удаленного репозитория Yum, а затем обновите или установите пакет из этого репозитория.

В РЕЗУЛЬТАТЕ

У вас должна быть возможность настроить систему для получения обновлений программного обеспечения с сервера класса и обновить систему для использования последних пакетов.

ПРЕЖДЕ ЧЕМ ВЫ НАЧНЕТЕ

Войдите на рабочую станцию **workstation** как пользователь **student**, используя в качестве пароля слово **student**.

На рабочей станции **workstation** выполните скрипт **lab software-repo start**. Команда запускает сценарий, который определяет, доступен ли хост, сервер **servera**, в сети. Сценарий также обеспечивает установку пакета **yum**.

```
[student@workstation ~]$ lab software-repo start
```

1. Используйте команду **ssh** для входа на сервер **servera**, как пользователь **student**.

```
[student@workstation ~]$ ssh student@servera  
[student@servera ~]$
```

2. Используйте команду **su -** для переключения на **root** в приглашении оболочки.

```
[student@servera ~]$ su -  
Password: redhat  
[root@servera ~]#
```

3. Настройте репозитории программного обеспечения на сервере для получения пользовательских пакетов и обновлений по **следующему URL-адресу**:

- Пользовательские пакеты, представленные на http://content.example.com/rhel8.0/x86_64/rhcsa-practice/rht
- Обновления пользовательских пакетов, представленных на http://content.example.com/rhel8.0/x86_64/rhcsa-practice/errata.

- 3.1.** Используйте **yum-config-manager**, чтобы добавить репозиторий пользовательских пакетов.

```
[root@servera ~]# yum-config-manager \
> --add-repo "http://content.example.com/rhel8.0/x86_64/rhcsa-practice/rht"
Adding repo from: http://content.example.com/rhel8.0/x86_64/rhcsa-practice/rht
```

- 3.2.** Изучите файл репозитория программного обеспечения, созданный предыдущей командой, в каталоге **/etc/yum.repos.d**. Используйте команду **vim** для редактирования файла и добавьте параметр **gpgcheck=0**, чтобы отключить проверку ключа GPG для репозитория.

```
[root@servera ~]# vim \
> /etc/yum.repos.d/content.example.com_rhel8.0_x86_64_rhcsa-practice_rht.repo
[content.example.com_rhel8.0_x86_64_rhcsa-practice_rht]
name=created by dnf config-manager from http://content.example.com/rhel8.0/x86_64/
rhcsa-practice/rht
baseurl=http://content.example.com/rhel8.0/x86_64/rhcsa-practice/rht
enabled=1
gpgcheck=0
```

- 3.3.** Создайте файл **/etc/yum.repos.d/errata.repo**, чтобы включить репозиторий обновлений со следующим содержимым:

```
[rht-updates]
name=rht updates
baseurl=http://content.example.com/rhel8.0/x86_64/rhcsa-practice/errata
enabled=1
gpgcheck=0
```

- 3.4.** Используйте команду **yum repolist all**, чтобы вывести список всех репозиториев в системе:

```
[root@servera ~]# yum repolist all
repo id                                repo name      status
content.example.com_rhel8.0_x86_64_rhcsa-practice_rht created by .... enabled: 2
rht-updates                            rht updates    enabled: 2
...output omitted...
```

- 4.** Отключите репозиторий программного обеспечения **rht-updates** и установите пакет **rht-system**.

- 4.1.** Используйте **yum-config-manager --disable**, чтобы отключить репозиторий **rht-updates**.

```
[root@servera ~]# yum-config-manager --disable rht-updates
```

4.2. Перечислите, а затем установите пакет **rht-system**:

```
[root@servera ~]# yum list rht-system
Available Packages
rht-system.noarch 1.0.0-1 content.example.com_rhel8.0_x86_64_rhcsa-practice_rht
[root@servera ~]# yum install rht-system
Dependencies resolved.
```

Package	Arch	Version	Repository	Size
<hr/>				
Installing:				
rht-system	noarch	1.0.0-1	content..._rht	3.7 k
...output omitted...				
Is this ok [y/N]: y				
...output omitted...				
Installed:				
rht-system-1.0.0-1.noarch				
Complete!				

4.3. Убедитесь, что пакет **rht-system** установлен, и запишите номер версии пакета.

```
[root@servera ~]# yum list rht-system
Installed Packages
rht-system.noarch      1.0.0-1 @content.example.com_rhel8.0_x86_64_rhcsa-practice_rht
```

5. Включите репозиторий программного обеспечения **rht-updates** и обновите все соответствующие пакеты программного обеспечения.

5.1. Используйте **yum-config-manager --enable**, чтобы включить репозиторий **rht-updates**.

```
[root@servera ~]# yum-config-manager --enable rht-updates
```

5.2. Используйте команду **yum update**, чтобы обновить все программные пакеты на сервере.

```
[root@servera ~]# yum update
Dependencies resolved.
```

```
=====
 Package          Arch      Version       Repository      Size
 =====
Upgrading:
 rht-system      x86_64    1.0.0-2.el7   rht-updates    3.9 k
...output omitted...
Is this ok [y/N]: y
...output omitted...
Complete!
```

5.3. Убедитесь, что пакет **rht-system** обновлен, и запишите номер версии пакета.

```
[root@servera ~]# yum list rht-system
Installed Packages
rht-system.noarch      1.0.0-2.el7      @rht-updates
```

6. Выход из сервера.

```
[root@servera ~]# exit
logout
[student@servera ~]$ exit
logout
Connection to servera closed.
[student@workstation ~]$
```

Завершение

На рабочей станции **workstation**, запустите сценарий **lab software-repo finish**, чтобы завершить упражнение. Этот сценарий удаляет все программные репозитории и пакеты, установленные на сервере во время упражнения.

```
[student@workstation ~]$ lab software-repo finish
```

На этом пошаговое упражнение завершено.

УПРАВЛЕНИЕ PACKAGE MODULE STREAMS

ЗАДАЧИ

После завершения этого раздела вы сможете:

- Объяснить, как модули позволяют устанавливать определенные версии программного обеспечения.
- Как составлять список, включать и переключать модули stream.
- Устанавливать и обновлять пакеты из модуля.

ВВЕДЕНИЕ В APPLICATION STREAM

Red Hat Enterprise Linux 8.0 представляет концепцию **потоков приложений** (**Application Streams**). **Несколько версий компонентов** пользовательского пространства, поставляемых с дистрибутивом, **теперь поставляются одновременно**. Они могут обновляться чаще, чем пакеты основной операционной системы. Это обеспечивает большую гибкость для настройки Red Hat Enterprise Linux без ущерба для базовой стабильности платформы или конкретных развертываний.

Традиционно управление альтернативными версиями программного пакета приложения и связанных с ним пакетов означало поддержание разных репозиториев для каждой версии. Для разработчиков, которым нужна была последняя версия приложения, и администраторов, которым нужна была наиболее стабильная версия приложения, это создавало ситуацию, управлять которой было утомительно. В Red Hat Enterprise Linux 8 этот процесс упрощен с использованием новой технологии, называемой модульностью. Модульность позволяет в одном репозитории размещать несколько версий пакета приложения и его зависимостей.

Контент Red Hat Enterprise Linux 8 распространяется через два основных репозитория программного обеспечения: **BaseOS** и **Application Stream (AppStream)**.

BaseOS

Репозиторий **BaseOS** предоставляет основное содержимое операционной системы для Red Hat Enterprise Linux в виде пакетов **RPM**. Компоненты **BaseOS** имеют жизненный цикл, идентичный жизненному циклу содержимого в предыдущих выпусках Red Hat Enterprise Linux.

Application Stream

Репозиторий **Application Stream** предоставляет контент с различным жизненным циклом как в модулях, так и в традиционных пакетах. **Application Stream** содержит необходимые части системы, а также широкий спектр приложений, ранее доступных в составе **Red Hat Software Collections** и других продуктов, и программ.



ВАЖНО

И **BaseOS**, и **AppStream** являются необходимой частью системы Red Hat Enterprise Linux 8.

Репозиторий **Application Stream** содержит два типа контента: **модули и традиционные пакеты RPM**. Модуль описывает набор принадлежащих друг другу пакетов **RPM**. Модули могут содержать несколько потоков, чтобы сделать несколько версий приложений доступными для установки. Включение потока модулей дает системе доступ к пакетам **RPM** в этом потоке модулей.

МОДУЛИ

Модуль - это набор пакетов **RPM**, которые представляют собой согласованный набор, принадлежащий друг другу. Обычно это организовано вокруг конкретной версии программного приложения или языка программирования. Типичный модуль может содержать пакеты с приложением, пакеты со специфическими библиотеками зависимостей приложения, пакеты с документацией для приложения и пакеты со вспомогательными утилитами.

Модуль Streams

Каждый модуль может иметь один или несколько модульных потоков, которые содержат разные версии контента. Каждый из потоков получает обновления независимо. Подумайте о модульном потоке как о виртуальном репозитории в физическом репозитории **Application Stream**.

Для каждого модуля можно включить только один из его потоков и предоставить свои пакеты.

Профили модуля

Каждый модуль может иметь один или несколько профилей. **Профиль** - это список определенных пакетов, которые должны быть установлены вместе для конкретного варианта использования, такого как сервер, клиент, разработка, минимальная установка или другое.

Установка определенного профиля модуля просто устанавливает определенный набор пакетов из потока модулей. Впоследствии вы можете установить или удалить пакеты обычным образом. Если вы не укажете профиль, модуль установит свой профиль по умолчанию.

УПРАВЛЕНИЕ МОДУЛЯМИ С ПОМОЩЬЮ YUM

Yum версии 4, новое в Red Hat Enterprise Linux 8, добавляет поддержку новых модульных функций Application Stream.

Для обработки содержимого модулей добавлена команда **yum module**. В остальном **yum** работает с модулями так же, как и с обычными пакетами.

```
[user@host ~]$ yum module list
Red Hat Enterprise Linux 8.0 AppStream - AppStream HTB (RPMs)
Name           Stream      Profiles   Summary
389-ds          1.4        default    389 Directory Server (base)
ant             1.10 [d]    common    Java build tool
container-tools 1.0 [d]    common    Common tools and dependencies
for container runtimes
...output omitted...
Hint: [d]efault, [e]nabled, [x]disabled, [i]nstalled
```



ПРИМЕЧАНИЕ

Используйте *подсказку (Hint)* в конце вывода, чтобы определить, какие потоки и профили включены, отключены, установлены, а также какие из них являются значениями по умолчанию.

Чтобы перечислить потоки модулей для определенного модуля и получить их статус используйте команду:

```
[user@host ~]$ yum module list perl
Red Hat Enterprise Linux 8 for x86_64 - AppStream HTB (RPMs)
Name Stream Profiles Summary
perl 5.24 common [d], minimal Practical Extraction and Report Language
perl 5.26 [d] common [d], minimal Practical Extraction and Report Language
```

Чтобы вывести список пакетов, установленных потоком модулей по умолчанию используйте команду:

```
[user@host ~]$ yum module info perl
Name : perl
Stream : 5.26 [d][a]
Version : 820181219174508
Context : 9edba152
Profiles : common, minimal
Default profiles : default
Repo : rhel-8-for-x86_64-appstream-htb-rpms
Summary : Practical Extraction and Report Language
...output omitted...
Artifacts : perl-4:5.24.4-401.module+el8+2464+d274aed1.x86_64
            : perl-Algorithm-Diff-0:1.1903-9.module+el8+2464+d274aed1.noarch
            : perl-Archive-Tar-0:2.30-1.module+el8+2464+d274aed1.noarch
...output omitted...
```



ПРИМЕЧАНИЕ

Без указания потока модуля **yum module info** показывает детали потока модуля по умолчанию. Используйте формат **module-name:stream** для просмотра определенного потока модуля. Добавьте параметр **--profile** для отображения информации о пакетах, установленных каждым из профилей модуля. Например:

```
[user@host ~]$ yum module info --profile perl:5.24
```

Включение потоков модулей (Module Streams) и установка модулей

Потоки модулей должны быть включены (enabled), чтобы установить их модуль. Чтобы упростить этот процесс, когда модуль установлен, он включает поток модуля, если это необходимо. Потоки модулей можно включить вручную, используя **yum module enable** и указав имя потока модуля.



ВАЖНО

Для данного модуля может быть включен только один поток модуля. Включение дополнительного потока модуля отключит исходный поток модуля.

Установите модуль, используя поток и профили по умолчанию:

```
[user@host ~]$ sudo yum module install -y perl
Dependencies resolved.

=====
Package      Arch    Version       Repository          Size
=====
Installing group/module packages:
perl         x86_64  4:5.26.3-416.el8      rhel-8-for-x86_64-appstream-htb-rpms 72 k
Installing dependencies:
...output omitted...
Running transaction
Preparing           : 1/1
Installing perl-Exporter-5.72-396.el8.noarch      1/155
Installing perl-Carp-1.42-396.el8.noarch          2/155
...output omitted...
Installed:
perl-4:5.26.3-416.el8.x86_64
perl-Encode-Locale-1.05-9.el8.noarch
...output omitted...
Complete!
```



ПРИМЕЧАНИЕ

Те же результаты можно было бы получить, запустив **yum install @perl**. Обозначение **@** сообщает yum, что аргумент - это имя модуля, а не имя пакета.

Чтобы проверить состояние потока модуля и установленного профиля:

```
[user@host ~]$ yum module list perl
Red Hat Enterprise Linux 8.0 AppStream - AppStream HTB (RPMs)
Name Stream Profiles Summary
perl 5.24 common, minimal Practical Extraction and Report Language
perl 5.26 [d][e] common [i], minimal Practical Extraction and Report Language

Hint: [d]efault, [e]nabled, [x]disabled, [i]nstalled
```

Удаление модулей и отключение потоков модулей

При удалении модуля удаляются все пакеты, установленные профилями текущего включенного потока модулей, а также любые другие пакеты и модули, которые зависят от них. Пакеты, установленные из этого модульного потока, не указанные ни в одном из его профилей, остаются установленными в системе и могут быть удалены вручную.



ПРЕДУПРЕЖДЕНИЕ

Удаление модулей и переключение потоков модулей может быть немного сложным. Переключение потока, включенного для модуля, эквивалентно сбросу текущего потока и включению нового потока. Он не изменяет автоматически установленные пакеты. Это нужно делать вручную.

Непосредственная установка потока модулей, отличных от того, который установлен в настоящее время, не рекомендуется, поскольку во время установки могут выполняться сценарии обновления, которые нарушают работу исходного потока модулей. Это может привести к потере данных или другим проблемам с конфигурацией.

Действуйте с осторожностью.

Чтобы удалить установленный модуль:

```
[user@host ~]$ sudo yum module remove -y perl
Dependencies resolved.

=====
Package           ArchVersion      Repository      Size
=====
Removing:
perl            x86_64:5.26.3-416.el8      @rhel-8-for-x86_64-appstream-htb-rpms 0
Removing unused dependencies:
...output omitted...
Running transaction
Preparing       :                                1/1
Erasing        : perl-4:5.26.3-416.el8.x86_64    1/155
```

```
Erasing      : perl-CPAN-2.18-397.el8.noarch
...output omitted...
Removed:
perl-4:5.26.3-416.el8.x86_64
dwz-0.12-9.el8.x86_64
...output omitted...
Complete!
```

2/155

После удаления модуля поток модуля все еще включен. Чтобы убедиться, что поток модуля все еще включен:

```
[user@host ~]$ yum module list perl
Red Hat Enterprise Linux 8.0 AppStream - AppStream HTB (RPMs)
Name   Stream      Profiles          Summary
perl   5.24        common [d], minimal Practical Extraction and Report Language
perl   5.26 [d][e]   common [d], minimal Practical Extraction and Report Language

Hint: [d]efault, [e]nabled, [x]disabled, [i]nstalled
```

Чтобы отключить поток модуля:

```
[user@host ~]$ sudo yum module disable perl
...output omitted...
Dependencies resolved.
=====
Package           Arch       Version      Repository      Size
=====
Disabling module streams:
perl              5.26
Is this ok [y/N]: y
Complete!
```

Переключение потоков модуля

Переключение потоков модуля обычно требует обновления или понижения версии содержимого до другой.

Чтобы обеспечить чистый коммутатор, сначала следует удалить модули, предоставленные потоком модулей. Это приведет к удалению всех пакетов, установленных профилями модуля, а также всех модулей и пакетов, от которых эти пакеты зависят.

Чтобы включить другой поток модуля и установить модуль:

```
[user@host ~]$ sudo yum module install perl:5.24
```

Новый поток модуля будет включен, а текущий поток отключен. Может потребоваться обновить или понизить версию пакетов из предыдущего потока модулей, которые не указаны в новом профиле. При необходимости используйте **yum distro-sync** для выполнения этой задачи. Также могут быть пакеты, которые остались установленными из предыдущего потока модулей. Удалите те, которые используют **yum remove**.



РЕКОМЕНДАЦИИ

Использование потока приложений (**Application Stream**)

https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/8-beta/html/using_application_stream/index

Модульность

<https://docs.fedoraproject.org/en-US/modularity/>

УПРАЖНЕНИЯ С ПОШАГОВЫМИ ИНСТРУКЦИЯМИ

УПРАВЛЕНИЕ ПОТОКАМИ MODULE STREAMS

В этом упражнении вы перечислите доступные модули, включите определенный поток модулей и установите пакеты из этого потока.

В РЕЗУЛЬТАТЕ

Вы должны быть способны:

- Перечислить установленные модули и изучить информацию о модуле.
- Включить и установить модуль из потока.
- Переключиться на конкретный поток модуля.
- Удалить и отключить модуль.

ПРЕЖДЕ ЧЕМ ВЫ НАЧНЕТЕ

Войдите на рабочую станцию **workstation** как пользователь **student**, используя в качестве пароля слово **student**.

На рабочей станции **workstation** выполните скрипт **lab software-module start**. Команда запускает сценарий, который определяет, доступен ли хост, **servera** по сети. Сценарий также обеспечивает доступность необходимых репозиториев программного обеспечения и устанавливает модуль **postgresql:9.6**.

```
[student@workstation ~]$ lab software-module start
```

1. Используйте команду **ssh** для входа на сервер **servera** как пользователь **student**.

```
[student@workstation ~]$ ssh student@servera  
[student@servera ~]$
```

2. Переключитесь на **root** в приглашении оболочки.

```
[student@servera ~]$ su -  
Password: redhat  
[root@servera ~]#
```

3. Перечислите доступные модули, потоки и установленные модули. Изучите информацию о модуле **python36**.

- 3.1. Используйте команду **yum module list** для вывода списка доступных модулей и потоков.

```
[root@servera ~]# yum module list
Red Hat Enterprise Linux 8.0 AppStream (dvd)
Name           Stream      Profiles          Summary
...
python27       2.7 [d]     common [d]        Python programming ..., version 2.7
python36       3.6 [d]     common [d], build  Python programming ..., version 3.6
...
Hint: [d]efault, [e]nabled, [x]disabled, [i]nstalled
```

- 3.2. Используйте команду **yum module list --installed** для вывода списка установленных модулей и потоков.

```
[root@servera ~]# yum module list --installed
Red Hat Enterprise Linux 8.0 AppStream (dvd)
Name           Stream      Profiles          Summary
...
postgresql    9.6 [e]     client, server [d] [i]  PostgreSQL server and client ...
...
Hint: [d]efault, [e]nabled, [x]disabled, [i]nstalled
```

- 3.3. Используйте команду **yum module info**, чтобы изучить детали модуля *python36*.

```
[root@servera ~]# yum module info python36
Name          : python36
Stream        : 3.6 [d]
Version       : 820190123171828
Context       : 17efdbc7
Profiles      : common [d], build
Default profiles: common
Repo          : rhel-8.0-for-x86_64-appstream-rpms
Summary       : Python programming language, version 3.6
...output omitted...
Hint: [d]efault, [e]nabled, [x]disabled, [i]nstalled, [a]ctive]
```

4. Установите модуль *python36* из потока 3.6 и общего профиля. Проверьте текущий статус модуля.

- 4.1. Используйте команду **yum module install**, чтобы установить модуль *python36*. Используйте синтаксис **name:stream/profile** для установки модуля *python36* из потока 3.6 и общего профиля.



ПРИМЕЧАНИЕ

Вы можете опустить **/profile**, чтобы использовать профиль по умолчанию, и **:stream**, чтобы использовать поток по умолчанию.

```
[root@servera ~]# yum module install python36:3.6/common  
...output omitted...  
Is this ok [y/N]: y  
...output omitted...  
Complete!
```

В качестве альтернативы вы можете сначала включить модуль, чтобы его пакеты были доступны системе, не устанавливая их. Затем используйте команду **yum install @python36** для установки пакетов в модуле.

```
[root@servera ~]# yum module enable python36:3.6/common  
[root@servera ~]# yum install @python36  
...output omitted...  
Is this ok [y/N]: y  
...output omitted...  
Complete!
```

4.2. Изучите текущий статус модуля *python36*.

```
[root@servera ~]# yum module list python36  
Red Hat Enterprise Linux 8.0 AppStream (dvd)  
Name      Stream      Profiles          Summary  
python36  3.6 [d][e]  common [d] [i], build  Python programming ..., version 3.6  
  
Hint: [d]efault, [e]nabled, [x]disabled, [i]nstalled
```

5. Переключите модуль **postgresql** в профиле сервера на использование потока (stream) 10.

5.1. Используйте команду **yum module list**, чтобы вывести список модуля **postgresql** и потока. Обратите внимание, что поток модуля **postgresql:9.6** в настоящее время установлен.

```
[root@servera ~]# yum module list postgresql  
Red Hat Enterprise Linux 8.0 AppStream (dvd)  
Name      Stream      Profiles          Summary  
postgresql  10 [d]    client, server [d]  PostgreSQL server and client ...  
postgresql  9.6 [e]    client, server [d] [i]  PostgreSQL server and client ...  
  
Hint: [d]efault, [e]nabled, [x]disabled, [i]nstalled
```

5.2. Используйте команду **yum module install**, чтобы переключиться на поток модуля **postgresql:10**.

```
[root@servera ~]# yum module install postgresql:10
...output omitted...
Is this ok [y/N]: y
...output omitted...
Complete!
```

5.3. Убедитесь, что модуль **postgresql** переключен на **поток (stream) 10**.

```
[root@servera ~]# yum module list postgresql
Red Hat Enterprise Linux 8.0 AppStream (dvd)
      Name      Stream     Profiles          Summary
  postgresql    10 [d] [e] client, server [d] [i] PostgreSQL server and client ...
  postgresql     9.6        client, server [d] PostgreSQL server and client ...

Hint: [d]efault, [e]nabled, [x]disabled, [i]nstalled
```

6. Удалите и отключите поток модуля **postgresql** вместе со всеми пакетами, установленными профилем.

6.1. Используйте команду **yum remove module**, чтобы удалить модуль **postgresql**. Команда также удаляет все пакеты, установленные из этого модуля.

```
[root@servera ~]# yum module remove postgresql
...output omitted...
Is this ok [y/N]: y
...output omitted...
Complete!
```

6.2. Отключите поток модуля **postgresql**.

```
[root@servera ~]# yum module disable postgresql
...output omitted...
Is this ok [y/N]: y
...output omitted...
Complete!
```

6.3. Убедитесь, что поток модуля **postgresql** удален и отключен.

```
[root@servera ~]# yum module list postgresql
Red Hat Enterprise Linux 8.0 AppStream (dvd)
```

```
Name      Stream   Profiles          Summary
postgresql 10 [d][x] client, server [d] PostgreSQL server and client ...
postgresql 9.6 [x]    client, server [d] PostgreSQL server and client ...

Hint: [d]efault, [e]nabled, [x]disabled, [i]nstalled
```

7. Выход из сервера **servera**.

```
[root@servera ~]# exit
logout
[student@servera ~]$ exit
logout
Connection to servera closed.
[student@workstation ~]$
```

Завершение

На рабочей станции **workstation**, запустите сценарий **lab software-module finish**, чтобы завершить упражнение. Этот скрипт удаляет все модули, установленные на сервере во время упражнения.

```
[student@workstation ~]$ lab software-module finish
```

На этом пошаговое упражнение завершено.

ЛАБОРАТОРНАЯ РАБОТА

УСТАНОВКА И ОБНОВЛЕНИЕ ПАКЕТОВ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

КОНТРОЛЬНЫЙ СПИСОК РАБОТЫ

В этой лабораторной работе вы будете управлять репозиториями программного обеспечения и потоками модулей, а также устанавливать и обновлять пакеты из этих репозиториев и потоков.

В РЕЗУЛЬТАТЕ

Вы должны быть способны:

- Управление репозиториями программного обеспечения и потоками модулей.
- Устанавливать и обновлять пакеты из репозиториев и потоков.
- Установите пакет **RPM**.

ПРЕЖДЕ ЧЕМ ВЫ НАЧНЕТЕ

Войдите на рабочую станцию **workstation** как пользователь **student**, используя в качестве пароля слово **student**.

На рабочей станции **workstation** выполните скрипт з **lab software-review start**. Сценарий гарантирует, что **serverb** доступен. Он также загружает все пакеты, необходимые для лабораторного упражнения.

```
[student@workstation ~]$ lab software-review start
```

1. На **serverb** настройте репозиторий программного обеспечения для получения обновлений. Репозиторий должен называться **/etc/yum.repos.d/errata.repo**. Должен быть доступ по адресу **http://content.example.com/rhel8.0/x86_64/rhcsa-practice/errata**. Не проверяйте подписи GPG.
2. На **serverb** установите новый пакет **xsane-gimp** и модуль **Apache HTTP Server** из потока 2.4 и общего профиля.
3. По соображениям безопасности **serverb** не должен иметь возможность отправлять что-либо на печать. Для этого удалите пакет **cups**. Выйти из учетной записи **root**.
4. Убедитесь, что пакет **rhcsa-script-1.0.0-1.noarch.rpm** доступен на **serverb**. Установите пакет. Для установки пакета вам потребуется получить права суперпользователя. Убедитесь, что пакет установлен. Выйти с сервера **serverb**.

Оценка

На рабочей станции **workstation**, запустите сценарий **lab software-review grade**, чтобы подтвердить успех в этой лабораторной работе.

```
[student@workstation ~]$ lab software-review grade
```

Завершение

На рабочей станции **workstation** запустите сценарий **lab software-review finish**, чтобы завершить данное упражнение. Этот сценарий удаляет репозиторий и пакеты, созданные во время этого упражнения.

```
[student@workstation ~]$ lab software-review finish
```

На этом лабораторная работа завершена.

РЕШЕНИЕ

УСТАНОВКА И ОБНОВЛЕНИЕ ПАКЕТОВ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

КОНТРОЛЬНЫЙ СПИСОК РАБОТЫ

В этой лабораторной работе вы будете управлять репозиториями программного обеспечения и потоками модулей, а также устанавливать и обновлять пакеты из этих репозиториев и потоков.

В РЕЗУЛЬТАТЕ

Вы должны быть способны:

- Управление репозиториями программного обеспечения и потоками модулей.
- Устанавливать и обновлять пакеты из репозиториев и потоков.
- Установите пакет **RPM**.

ПРЕЖДЕ ЧЕМ ВЫ НАЧНЕТЕ

Войдите на рабочую станцию **workstation** как пользователь **student**, используя в качестве пароля слово **student**.

На рабочей станции **workstation** выполните скрипт з **lab software-review start**. Сценарий гарантирует, что **serverb** доступен. Он также загружает все пакеты, необходимые для лабораторного упражнения.

```
[student@workstation ~]$ lab software-review start
```

1. На **serverb** настройте репозиторий программного обеспечения для получения обновлений. Репозиторий должен называться **/etc/yum.repos.d/errata.repo**. Должен быть доступ по адресу http://content.example.com/rhel8.0/x86_64/rhcsa-practice/errata. Не проверяйте подписи GPG.

1.1. С рабочей станции используйте команду **ssh** для входа на **serverb** как пользователь **student**.

```
[student@workstation ~]$ ssh student@serverb
Warning: Permanently added 'serverb,172.25.250.11' (ECDSA) to the list of known
hosts.
Activate the web console with: systemctl enable --now cockpit.socket

Last login: Thu Mar  7 15:17:22 2019
[student@serverb ~]$
```

1.2. Используйте команду **su**, чтобы переключиться на пользователя **root**. Пароль - **redhat**.

```
[student@serverb ~]$ su -  
Password: redhat  
[root@serverb ~]#
```

1.3. Создайте файл **/etc/yum.repos.d/errata.repo** со следующего содержанием:

```
[updates]  
name=Red Hat Updates  
baseurl=http://content.example.com/rhel8.0/x86_64/rhcsa-practice/errata  
enabled=1  
gpgcheck=0
```

2. На **serverb** установите новый пакет **xsane-gimp** и модуль **Apache HTTP Server** из потока 2.4 и общего профиля.

2.1. Используйте команду **yum list**, чтобы вывести список доступных пакетов для **xsane-gimp**.

```
[root@serverb ~]# yum list xsane-gimp  
Last metadata expiration check: 0:24:30 ago on Thu 07 Mar 2019 03:50:55 PM CET.  
Available Packages  
xsane-gimp.x86_64      0.999-30.el8    rhel-8.0-for-x86_64-appstream-rpms
```

2.2. Установите последнюю версию пакета **xsane-gimp** с помощью команды **yum install**.

```
[root@serverb ~]# yum install xsane-gimp  
...output omitted...  
Install 57 Packages  
  
Total download size: 51 M  
Installed size: 205 M  
Is this ok [y/N]: y  
...output omitted...  
Complete!  
[root@serverb ~]#
```

2.3. Список доступных модулей и потоков. Найдите модуль **httpd**. Используйте команду **yum install**, чтобы установить модуль **httpd** с потоком **2.4** и общим профилем.

```
[student@serverb ~]$ yum module list  
Name      Stream      Profiles      Summary  
...output omitted...  
httpd 2.4 [d] common [d], devel, minimal Apache HTTP Server
```

```
...output omitted...
[root@serverb ~]# yum module install httpd:2.4/common
Install      10 Packages

Total download size: 2.1 M
Installed size: 5.7 M
Is this ok [y/N]: y
...output omitted...
Complete!
[root@serverb ~]#
```

3. По соображениям безопасности **serverb** не должен иметь возможность отправлять что-либо на печать. Для этого удалите пакет **cups**. Выйти из учетной записи **root**.

3.1. Используйте команду **yum list**, чтобы вывести список установленных пакетов **cups**.

```
[root@serverb ~]# yum list cups
Installed Packages
cups.x86_64      1:2.2.6-25.el8          @rhel-8.0-for-x86_64-appstream-rpms
[root@serverb ~]#
```

3.2. Используйте команду **yum remove**, чтобы удалить пакет **cups**.

```
[root@serverb ~]# yum remove cups.x86_64
...output omitted...
Remove 9 Packages

Freed space: 11 M
Is this ok [y/N]:
...output omitted...
Complete!
```

3.3. Выйти из учетной записи **root**.

```
[root@serverb ~]# exit
[student@serverb ~]$
```

4. Убедитесь, что пакет **rhcsa-script-1.0.0-1.noarch.rpm** доступен на **serverb**. Установите пакет. Для установки пакета вам потребуется получить права суперпользователя. Убедитесь, что пакет установлен. Выйти с сервера **serverb**.

4.1. Используйте команду **rpm**, чтобы убедиться, что пакет **rhcsa-script-1.0.0-1.noarch.rpm** доступен на **serverb**, просмотрев информацию о пакете.

```
[student@serverb ~]$ rpm -q -p rhcsa-script-1.0.0-1.noarch.rpm -i
Name        : rhcsa-script
Version     : 1.0.0
Release     : 1
Architecture: noarch
Install Date: (not installed)
Group       : System
Size        : 1056
License      : GPL
Signature    : (none)
Source RPM  : rhcsa-script-1.0.0-1.src.rpm
Build Date  : Wed 06 Mar 2019 11:29:46 AM CET
Build Host   : foundation0.ilt.example.com
Relocations  : (not relocatable)
Packager    : Snehangshu Karmakar
URL         : http://example.com
Summary      : RHCSA Practice Script
Description  :
A RHCSA practice script.
```

The package changes the motd.

4.2. Используйте команду **sudo yum localinstall** для установки пакета **rhcsa-script-1.0.0-1.noarch.rpm**. Используйте пароль **student**.

```
[student@serverb ~]$ sudo yum localinstall rhcsa-script-1.0.0-1.noarch.rpm
```

```
[sudo] password for student: student
```

```
Last metadata expiration check: 1:31:22 ago on Thu 07 Mar 2019 03:50:55 PM CET.
```

```
Dependencies resolved.
```

Package	Arch	Version	Repository	Size
rhcsa-script	noarch	1.0.0-1	@commandline	7.6 k

```
Transaction Summary
```

```
Install 1 Package
```

```
Total size: 7.6 k
```

```
Installed size: 1.0 k
```

```
Is this ok [y/N]: y
```

```
Downloading Packages:
```

```
Running transaction check
```

```
Transaction check succeeded.
```

```
Running transaction test
```

```
Transaction test succeeded.
```

```
Running transaction
```

```
Preparing : 1/1
```

```
Running scriptlet: rhcsa-script-1.0.0-1.noarch 1/1
```

```
Installing : rhcsa-script-1.0.0-1.noarch 1/1
```

```
Running scriptlet: rhcsa-script-1.0.0-1.noarch 1/1
Verifying : rhcsa-script-1.0.0-1.noarch 1/1
```

Installed:
rhcsa-script-1.0.0-1.noarch

Complete!

4.3. Используйте команду **rpm**, чтобы убедиться, что пакет установлен.

```
[student@serverb ~]$ rpm -q rhcsa-script
rhcsa-script-1.0.0-1.noarch
[student@serverb ~]$
```

4.4. Выход с сервера **serverb**.

```
[student@serverb ~]$ exit
logout
Connection to serverb closed.
[student@workstation ~]$
```

Оценка

На рабочей станции **workstation**, запустите сценарий **lab software-review grade**, чтобы подтвердить успех в этой лабораторной работе.

```
[student@workstation ~]$ lab software-review grade
```

Завершение

На рабочей станции **workstation** запустите сценарий **lab software-review finish**, чтобы завершить данное упражнение. Этот сценарий удаляет репозиторий и пакеты, созданные во время этого упражнения.

```
[student@workstation ~]$ lab software-review finish
```

РЕЗЮМЕ

В этой главе вы узнали:

- Red Hat Subscription Management предоставляет инструменты для предоставления компьютерам права подписки на продукты, получения обновлений для пакетов программного обеспечения и отслеживания информации о контрактах на поддержку и подписках, используемых системами.
- Программное обеспечение предоставляется в виде пакетов **RPM**, что упрощает установку, обновление и удаление программного обеспечения из системы.
- Команду **rpm** можно использовать для запроса локальной базы данных, чтобы предоставить информацию о содержимом установленных пакетов и установить загруженные файлы пакетов.
- **yum** - это мощный инструмент командной строки, который можно использовать для установки, обновления, удаления и запроса пакетов программного обеспечения.
- **Red Hat Enterprise Linux 8** использует потоки приложений для обеспечения единого репозитория для размещения нескольких версий пакетов приложения и его зависимостей.

ГЛАВА 15

ДОСТУП К ФАЙЛОВЫМ СИСТЕМАМ LINUX

ЦЕЛЬ

Доступ, проверка и использование существующих файловых систем в хранилище, подключенном к серверу Linux.

ЗАДАЧИ

- Объясните, что такое блочное устройство, интерпретируйте имена файлов запоминающих устройств и определите запоминающее устройство, используемое файловой системой для конкретного каталога или файла.
- Доступ к файловым системам, прикрепляя их к каталогу в иерархии файловой системы.
- Поиск файлов в смонтированных файловых системах с помощью команд **find** и **locate**.

РАЗДЕЛЫ

- Определение файловых систем и устройств (и контрольный опрос)
- Монтирование и размонтиrovание файловых систем (и упражнения с пошаговыми инструкциями)
- Поиск файлов в системе (и упражнения с пошаговыми инструкциями)

ЛАБОРОТОРНАЯ РАБОТА

Доступ к файловым системам Linux.

ИДЕНТИФИКАЦИЯ ФАЙЛОВЫХ СИСТЕМ И УСТРОЙСТВ

ЗАДАЧИ

После завершения этого раздела вы сможете определить каталог в иерархии файловой системы и на каком устройстве он хранится.

КОНЦЕПЦИИ УПРАВЛЕНИЯ ХРАНЕНИЕМ

Доступ к файлам на сервере Linux осуществляется через иерархию файловой системы, единое перевернутое дерево каталогов. Эта иерархия файловой системы собирается из файловых систем, предоставляемых устройствами хранения, доступными вашей системе. Каждая файловая система - это запоминающее устройство, отформатированное для хранения файлов.

В некотором смысле иерархия файловой системы Linux представляет собой набор файловых систем на отдельных устройствах хранения, как если бы это был один набор файлов на одном гигантском устройстве хранения, по которому вы можете перемещаться. В большинстве случаев вам не нужно знать, на каком устройстве хранения находится конкретный файл, вам просто нужно знать каталог, в котором находится файл.

Однако иногда это может быть важно. Возможно, вам потребуется определить, насколько заполнено устройство хранения и какие каталоги в иерархии файловой системы затронуты. В журналах с устройства хранения могут быть ошибки, и вам необходимо знать, какие файловые системы подвергаются риску. Вы можете просто создать жесткую связь между двумя файлами, и вам нужно знать, находятся ли они в одной файловой системе, чтобы определить, возможно ли это.

Файловые системы и точки монтирования

Чтобы содержимое файловой системы стало доступным в иерархии файловой системы, оно должно быть смонтировано в пустой каталог. Этот каталог называется точкой монтирования. После монтирования, если вы используете `ls` для вывода списка этого каталога, вы увидите содержимое смонтированной файловой системы, и вы сможете получить доступ к этим файлам и использовать их в обычном режиме. Многие файловые системы автоматически монтируются в процессе загрузки.

Если вы работали только с буквами дисков Microsoft Windows, это принципиально другая концепция. Это несколько похоже на функцию смонтированных папок **NTFS**.

Файловые системы, хранилища и блочные устройства

Низкоуровневый доступ к устройствам хранения в Linux обеспечивается файлом особого типа, называемым **блочным устройством** (*block device*). Блочные устройства должны быть отформатированы в файловой системе, прежде чем их можно будет смонтировать.

Файлы блочных устройств хранятся в каталоге `/dev` вместе с другими файлами устройств. Файлы устройств создаются операционной системой автоматически. В Red Hat Enterprise Linux первый обнаруженный жесткий диск **SATA/PATA, SAS, SCSI** или **USB** называется `/dev/sda`, второй - `/dev/sdb` и т. д. Эти имена представляют собой весь жесткий диск.

Другие типы хранилищ будут иметь другие формы именования.

Присвоение имен блочным устройствам

ТИП УСТРОЙСТВА	ШАБЛОН НАЗВАНИЯ УСТРОЙСТВА
SATA/SAS/USB-накопитель	/dev/sda, /dev/sdb ...
паравиртуализированное хранилище virtio-blk (некоторые виртуальные машины)	/dev/vda, /dev/vdb ...
Хранилище, подключенное к NVMe (многие твердотельные накопители SSD)	/dev/nvme0, /dev/nvme1 ...
SD/MMC/eMMC хранилище (SD -карты)	/dev/mmcblk0, /dev/mmcblk1 ...



ПРИМЕЧАНИЕ

Многие виртуальные машины используют новое паравиртуализированное хранилище **virtio-scsi** с именами в стиле **/dev/sd***.

Разделы диска

Обычно вы не превращаете все устройство хранения в одну файловую систему. Устройства хранения обычно делятся на более мелкие части, называемые разделами.

Разделы позволяют разделить диск на разделы: различные разделы могут быть отформатированы в разных файловых системах или использоваться для разных целей. Например, один раздел может содержать домашние каталоги пользователей, а другой может содержать системные данные и журналы. Если пользователь заполняет раздел домашнего каталога данными, в системном разделе все еще может быть свободное место.

Разделы сами по себе являются блочными устройствами. В хранилище, подключенном к **SATA**, первый раздел на первом диске будет именоваться - **/dev/sda1**. Третий раздел на втором диске - **/dev/sdb3** и так далее.

Паравиртуализированные устройства хранения имеют аналогичную систему именования. Устройство **SSD**, подключенное к **NVMe**, называет свои разделы по-другому. В этом случае первый раздел на первом диске будет называться - **/dev/nvme0p1**. Третий раздел на втором диске - **/dev/nvme1p3** и так далее. Карты **SD** или **MMC** имеют аналогичную систему именования.

Длинный список файла устройства **/dev/sda1** на хосте показывает его специальный тип файла как **b**, что означает блочное устройство:

```
[user@host ~]$ ls -l /dev/sda1
brw-rw----. 1 root disk 8, 1 Feb 22 08:00 /dev/sda1
```

Логические объемы

Другой способ организации дисков и разделов - управление логическими томами (*logical volume management LVM*). С помощью **LVM** одно или несколько блочных устройств могут быть объединены в пул хранения, называемый группой томов (*volume group*). Затем дисковое пространство в группе томов распределяется по одному или нескольким логическим томам (*logical volumes*), которые являются функциональным эквивалентом раздела, находящегося на физическом диске.

Система **LVM** присваивает имена группам томов и логическим томам при создании. **LVM** создает каталог в **/dev**, который соответствует имени группы, а затем создает символическую ссылку в этом новом каталоге с тем же именем, что и логический том. После этого файл логического тома можно будет смонтировать. Например, если группа томов называется **myvg**, а логический том в ней называется **mylv**, то полный путь к файлу устройства логического тома будет **/dev/myvg/mylv**.



ПРИМЕЧАНИЕ

Упомянутая выше форма имени устройства логического тома фактически реализована как символьическая ссылка на фактический файл устройства, используемый для доступа к нему, который может варьироваться в зависимости от загрузки. Существует еще одна форма имени устройства логического тома, связанная с файлами в **/dev/mapper**, которые часто используются, а также являются символьическими ссылками на фактический файл устройства.

ИЗУЧЕНИЕ ФАЙЛОВЫХ СИСТЕМ

Чтобы получить обзор устройств с локальной и удаленной файловой системой, а также доступного свободного места, выполните команду **df**. Когда команда **df** запускается без аргументов, она сообщает об общем дисковом пространстве, используемом дисковом пространстве, свободном дисковом пространстве и процентах от общего дискового пространства, используемого во всех смонтированных обычных файловых системах. Он сообщает как о локальной, так и о удаленной файловой системе.

В следующем примере показаны файловые системы и точки монтирования на хосте.

```
[user@host ~]$ df
Filesystem      1K-blocks   Used Available Use% Mounted on
/devtmpfs        912584     0    912584   0% /dev
tmpfs           936516     0    936516   0% /dev/shm
tmpfs           936516  16812    919704   2% /run
tmpfs           936516     0    936516   0% /sys/fs/cgroup
/dev/vda3       8377344 1411332   6966012  17% /
/dev/vda1       1038336 169896   868440  17% /boot
tmpfs          187300     0   187300   0% /run/user/1000
```

Разбиение на хост-систему показывает две физические файловые системы, смонтированные в **/** и **/boot**. Это обычное дело для виртуальных машин. Устройства **tmpfs** и

devtmpfs представляют собой файловые системы в системной памяти. Все файлы, записанные в **tmpfs** или **devtmpfs**, исчезают после перезагрузки системы.

Чтобы улучшить читаемость размеров вывода, есть два разных удобочитаемых параметра: **-h** или **-H**. Разница между этими двумя параметрами заключается в том, что **-h** сообщает в **KiB** (210), **MiB** (220) или **GiB** (230), а опция **-H** сообщает в единицах СИ: **KB** (103), **MB** (106) или **GB** (109). Производители жестких дисков обычно используют единицы СИ при рекламе своей продукции.

Показать отчет о файловых системах в хост-системе со всеми модулями, преобразованными в удобочитаемый формат:

```
[user@host ~]$ df -h
Filesystem      Size  Used Avail Use% Mounted on
devtmpfs        892M    0  892M   0% /dev
tmpfs          915M    0  915M   0% /dev/shm
tmpfs          915M   17M  899M   2% /run
tmpfs          915M    0  915M   0% /sys/fs/cgroup
/dev/vda3       8.0G  1.4G  6.7G  17% /
/dev/vda1     1014M  166M  849M  17% /boot
tmpfs         183M    0  183M   0% /run/user/1000
```

Для получения более подробной информации о пространстве, используемом определенным деревом каталогов, используйте команду **du**. Команда **du** имеет параметры **-h** и **-H** для преобразования вывода в удобочитаемый формат. Команда **du** рекурсивно показывает размер всех файлов в текущем дереве каталогов.

Показать отчет об использовании диска для каталога **/usr/share** на хосте:

```
[root@host ~]# du /usr/share
...output omitted...
176  /usr/share/smartmontools
184  /usr/share/nano
8   /usr/share/cmake/bash-completion
8   /usr/share/cmake
356676 /usr/share
```

Показать отчет об использовании диска в удобочитаемом формате для каталога **/var/log** на хосте:

```
[root@host ~]# du -h /var/log
...output omitted...
176K /usr/share/smartmontools
184K /usr/share/nano
```

```
8.0K /usr/share/cmake/bash-completion  
8.0K /usr/share/cmake  
369M /usr/share
```



РЕКОМЕНДАЦИИ

Справочная страница **man df(1)** и **du(1)**

КОНТРОЛЬНЫЙ ОПРОС

ИДЕНТИФИКАЦИЯ ФАЙЛОВЫХ СИСТЕМ И УСТРОЙСТВ

Выберите правильные ответы на следующие вопросы:

1. Как называется файл устройства всего жесткого диска **SATA** в каталоге **/dev**?
 - a. **/dev/vda**
 - b. **/dev/sda1**
 - c. **/dev/sda**
 - d. **/dev/vg_install/lv_home**
2. Выберите имя файла устройства третьего раздела на втором жестком диске **SATA**.
 - a. **/dev/vda2**
 - b. **/dev/sda3**
 - c. **/dev/sdb2**
 - d. **/dev/sdb3**
3. Как называется файл устройства для всего второго диска **virtio-blk**, подключенного к виртуальной машине?
 - a. **/dev/vda2**
 - b. **/dev/sda2**
 - c. **/dev/vdb2**
 - d. **/dev/vdb**
4. Выберите правильное имя файла устройства для третьего раздела на втором диске **virtio-blk**, подключенном к виртуальной машине?
 - a. **/dev/vda3**
 - b. **/dev/sda3**
 - c. **/dev/vdb3**
 - d. **/dev/vda3**
5. Какая команда обеспечивает обзор точек монтирования файловой системы и количества доступного свободного места в единицах **СИ**?
 - a. **df**
 - b. **df -H**
 - c. **df -h**
 - d. **du -h**

РЕШЕНИЕ

ИДЕНТИФИКАЦИЯ ФАЙЛОВЫХ СИСТЕМ И УСТРОЙСТВ

Выберите правильные ответы на следующие вопросы:

1. Как называется файл устройства всего жесткого диска **SATA** в каталоге **/dev**?
 - a. /dev/vda
 - b. /dev/sda1
 - c. **/dev/sda**
 - d. /dev/vg_install/lv_home

2. Выберите имя файла устройства третьего раздела на втором жестком диске **SATA**.
 - a. /dev/vda2
 - b. /dev/sda3
 - c. /dev/sdb2
 - d. **/dev/sdb3**

3. Как называется файл устройства для всего второго диска **virtio-blk**, подключенного к виртуальной машине?
 - a. /dev/vda2
 - b. /dev/sda2
 - c. /dev/vdb2
 - d. **/dev/vdb**

4. Выберите правильное имя файла устройства для третьего раздела на втором диске **virtio-blk**, подключенном к виртуальной машине?
 - a. /dev/vda3
 - b. /dev/sda3
 - c. **/dev/vdb3**
 - d. /dev/vda3

5. Какая команда обеспечивает обзор точек монтирования файловой системы и количества доступного свободного места в единицах **СИ**?
 - a. df
 - b. **df -H**
 - c. df -h
 - d. du -h

МОНТИРОВАНИЕ И ДЕМОНТИРОВАНИЕ ФАЙЛОВЫХ СИСТЕМ

ЗАДАЧИ

После завершения этого раздела вы сможете получить доступ к содержимому файловых систем, добавляя и удаляя файловые системы из иерархии файловой системы.

УСТАНОВКА ФАЙЛОВЫХ СИСТЕМ ВРУЧНУЮ

Чтобы получить к нему доступ, необходимо смонтировать файловую систему на съемном запоминающем устройстве. Команда **mount** позволяет пользователю **root** вручную смонтировать файловую систему. Первый аргумент команды **mount** указывает монтируемую **файловую систему**. Второй аргумент указывает **каталог**, который будет использоваться в качестве точки монтирования в иерархии файловой системы.

Есть два распространенных способа указать файловую систему на разделе диска для команды **mount**:

- С именем файла устройства в **/dev**, содержащего файловую систему.
- С **UUID**, записанным в файловую систему, универсальный уникальный идентификатор.

Установить устройство относительно просто. Вам необходимо определить устройство, которое вы хотите смонтировать, убедиться, что точка монтирования существует, и смонтировать устройство в точке монтирования.

Идентификация блочного устройства

Запоминающее устройство с возможностью горячей замены, будь то жесткий диск (**HDD**) или твердотельное устройство (**SSD**) в серверной тележке или запоминающее устройство **USB**, может подключаться к другому порту каждый раз, когда они подключаются к системе.

Используйте команду **lsblk**, чтобы вывести список деталей указанного блочного устройства или всех доступных устройств.

```
[root@host ~]# lsblk
NAME      MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
vda        253:0   0   12G  0 disk
└─vda1     253:1   0    1G  0 part /boot
└─vda2     253:2   0    1G  0 part [SWAP]
└─vda3     253:3   0   11G  0 part /
vdb        253:16  0   64G  0 disk
└─vdb1     253:17  0   64G  0 part
```

Если вы знаете, что только что добавили устройство хранения на 64 ГБ с одним разделом, то из предыдущего вывода вы можете догадаться, что **/dev/vdb1** - это раздел, который вы хотите смонтировать.

Монтаж по имени блочного устройства

В следующем примере файловая система монтируется в разделе **/dev/vdb1** в каталоге **/mnt/data**.

```
[root@host ~]# mount /dev/vdb1 /mnt/data
```

Чтобы смонтировать файловую систему, целевой каталог должен уже существовать. Каталог **/mnt** существует по умолчанию и предназначен для использования в качестве временной точки монтирования.

Вы можете использовать каталог **/mnt** или, еще лучше, создать подкаталог **/mnt** для использования в качестве временной точки монтирования, если у вас нет веских причин для монтирования его в определенном месте в иерархии файловой системы.



ВАЖНО

Если каталог, выступающий в качестве точки монтирования, не пуст, любые файлы, скопированные в этот каталог до монтирования файловой системы, будут недоступны до тех пор, пока файловая система не будет снова размонтирована.

Этот подход отлично работает в краткосрочной перспективе. Однако порядок, в котором операционная система обнаруживает диски, может измениться, если устройства добавляются в систему или удаляются из нее. Это изменит имя устройства, связанное с этим устройством хранения. Лучшим подходом было бы монтирование по какой-то характеристике, встроенной в файловую систему.

Монтирование по UUID файловой системы

Одним из стабильных идентификаторов, связанных с файловой системой, является ее **UUID**, очень длинное шестнадцатеричное число, которое действует как универсальный уникальный идентификатор. Этот **UUID** является частью файловой системы и остается тем же самым до тех пор, пока файловая система не воссоздается.

Команда **lsblk -fp** выводит полный путь к устройству вместе с **UUID** и точками монтирования, а также тип файловой системы в разделе. Если файловая система не смонтирована, точка монтирования будет пустой.

```
[root@host ~]# lsblk -fp
```

NAME	FSTYPE	LABEL	UUID	MOUNTPOINT
/dev/vda				
└─/dev/vda1	xfs		23ea8803-a396-494a-8e95-1538a53b821c	/boot
└─/dev/vda2	swap		cdf61ded-534c-4bd6-b458-cab18b1a72ea	[SWAP]
└─/dev/vda3	xfs		44330f15-2f9d-4745-ae2e-20844f22762d	/
/dev/vdb				
└─/dev/vdb1	xfs		46f543fd-78c9-4526-a857-244811be2d88	

Смонтируйте файловую систему по UUID файловой системы.

```
[root@host ~]# mount UUID="46f543fd-78c9-4526-a857-244811be2d88" /mnt/data
```

АВТОМАТИЧЕСКИЙ МОНТАЖ СЪЕМНЫХ УСТРОЙСТВ ХРАНЕНИЯ

Если вы вошли в систему и используете графическую среду рабочего стола, она автоматически подключит все съемные носители, когда они будут вставлены.

Съемное запоминающее устройство монтируется в **/run/media/USERNAME/LABEL**, где **USERNAME** - это имя пользователя, вошедшего в графическую среду, а **LABEL** - это идентификатор, часто это имя, присвоенное файловой системе при ее создании, если таковая имеется.

Перед извлечением устройства его следует размонтировать вручную.

РАЗМОНТИРОВАНИЕ ФАЙЛОВЫХ СИСТЕМ

Процедуры выключения и перезагрузки автоматически размонтируют все файловые системы. В рамках этого процесса все данные файловой системы, кэшированные в памяти, сбрасываются на запоминающее устройство, таким образом гарантируя, что файловая система не будет повреждена.



ПРЕДУПРЕЖДЕНИЕ

Данные файловой системы часто кэшируются в памяти. Поэтому, чтобы не повредить данные на диске, важно отключать съемные диски перед их отключением. Процедура размонтирования синхронизирует данные перед освобождением диска, обеспечивая целостность данных.

Чтобы размонтировать файловую систему, команда **umount** ожидает точку монтирования в качестве аргумента.

```
[root@host ~]# umount /mnt/data
```

Размонтирование невозможно, если смонтированная файловая система используется. Для успешного выполнения команды **umount** все процессы должны прекратить доступ к данным в точке монтирования.

В приведенном ниже примере происходит сбой выполнения команды **umount** из-за того, что файловая система используется (оболочка использует **/mnt/data** в качестве текущего рабочего каталога), генерируя сообщение об ошибке.

```
[root@host ~]# cd /mnt/data  
[root@host data]# umount /mnt/data  
umount: /mnt/data: target is busy.
```

Команда **lsof** выводит список всех открытых файлов и процессов, обращающихся к ним, в указанном каталоге. Полезно определить, какие процессы в настоящее время препятствуют успешному размонтированию файловой системы.

```
[root@host data]# lsof /mnt/data  
COMMAND  PID USER   FD   TYPE DEVICE SIZE/OFF NODE NAME  
bash    1593 root cwd    DIR 253,17          6  128 /mnt/data  
lsof    2532 root cwd    DIR 253,17         19  128 /mnt/data  
lsof    2533 root cwd    DIR 253,17         19  128 /mnt/data
```

После того, как процессы идентифицированы, может быть предпринято действие, например ожидание завершения процесса или отправка процессу сигнала **SIGTERM** или **SIGKILL**. В этом случае достаточно изменить текущий рабочий каталог на каталог за пределами точки монтирования.

```
[root@host data]# cd  
[root@host ~]# umount /mnt/data
```



ПРИМЕЧАНИЕ

Распространенная причина, по которой файловые системы не могут отключиться, заключается в том, что оболочка **Bash** использует точку монтирования или подкаталог в качестве текущего рабочего каталога. Используйте команду **cd**, чтобы выйти из файловой системы, чтобы решить эту проблему.



РЕКОМЕНДАЦИИ

Справочные системы **man lsblk(8)**, **mount(8)**, **umount(8)**, и **lsof(8)**

УПРАЖНЕНИЯ С ПОШАГОВЫМИ ИНСТРУКЦИЯМИ

МОНТИРОВАНИЕ И ДЕМОНТИРОВАНИЕ ФАЙЛОВЫХ СИСТЕМ

В этом упражнении вы попрактикуетесь в монтировании и размонтировании файловых систем.

В РЕЗУЛЬТАТЕ

Вы должны иметь возможность идентифицировать и смонтировать новую файловую систему в указанной точке монтирования, а затем размонтировать ее.

ПРЕЖДЕ ЧЕМ ВЫ НАЧНЕТЕ

Войдите на рабочую станцию **workstation** как пользователь **student**, используя в качестве пароля слово **student**.

На рабочей станции **workstation** выполните скрипт **lab fs-mount start**. Команда запускает сценарий, который определяет, доступен ли хост, сервер **servera**, в сети.

Скрипт также создает раздел на втором диске, подключенном к серверу.

```
[student@workstation ~]$ lab fs-mount start
```

1. Используйте команду **ssh** для входа на сервер **servera** как пользователь **student**.

```
[student@workstation ~]$ ssh student@servera  
[student@servera ~]$
```

2. На второй диск (**/dev/vdb**) на сервере **servera** добавлен новый раздел с файловой системой. Смонтируйте новый доступный раздел по **UUID** во вновь созданной точке монтирования **/mnt/newspace**.

- 2.1. Используйте команду **su** - для переключения на пользователя **root**, поскольку только данный пользователь может вручную смонтировать устройство.

```
[student@servera ~]$ su -  
Password: redhat  
[root@servera ~]#
```

- 2.2. Создайте каталог **/mnt/newspace**.

```
[root@servera ~]# mkdir /mnt/newspace
```

2.3. Используйте команду **lsblk** с параметром **-fp**, чтобы узнать **UUID** устройства, **/dev/vdb1**.

```
[root@servera ~]# lsblk -fp /dev/vdb
NAME      FSTYPE   LABEL     UUID
/dev/vdb
└─/dev/vdb1  xfs          a04c511a-b805-4ec2-981f-42d190fc9a65
```

2.4. Смонтируйте файловую систему, используя **UUID** в каталоге **/mnt/newspace**. Замените на идентификатор диска **UUID** имя **/dev/vdb1** из предыдущих выходных данных команды.

```
[root@servera ~]# mount UUID="a04c511a-b805-4ec2-981f-42d190fc9a65" /mnt/newspace
```

2.5. Убедитесь, что устройство **/dev/vdb1** смонтировано в каталоге **/mnt/newspace**.

```
[root@servera ~]# lsblk -fp /dev/vdb
NAME      FSTYPE   LABEL     UUID                                     MOUNTPOINT
/dev/vdb
└─/dev/vdb1  xfs          a04c511a-b805-4ec2-981f-42d190fc9a65  /mnt/newspace
```

3. Перейдите в каталог **/mnt/newspace** и создайте новый каталог, **/mnt/newspace/newdir**, с пустым файлом **/mnt/newspace/newdir/newfile**.

3.1. Перейдите в каталог **/mnt/newspace**.

```
[root@servera ~]# cd /mnt/newspace
```

3.2. Создайте новый каталог **/mnt/newspace/newdir**.

```
[root@servera newspace]# mkdir newdir
```

3.3. Создайте новый пустой файл **/mnt/newspace/newdir/newfile**.

```
[root@servera newspace]# touch newdir/newfile
```

4. Размонтируйте файловую систему, установленную в каталоге **/mnt/newspace**.

4.1. Используйте команду **umount**, чтобы размонтировать **/mnt/newspace**, в то время как текущий каталог в оболочке по-прежнему **/mnt/newspace**. Команде **umount** не удается размонтировать устройство.

```
[root@servera newspace]# umount /mnt/newspace
umount: /mnt/newspace: target is busy.
```

4.2. Измените текущий каталог в оболочке на **/root**.

```
[root@servera newspace]# cd
[root@servera ~]#
```

4.3. Теперь успешно размонтируйте **/mnt/newspace**.

```
[root@servera newspace]# umount /mnt/newspace
umount: /mnt/newspace: target is busy.
```

5. Выход из сервера **servera**.

```
[root@servera ~]# exit
logout
[student@servera ~]$ exit
logout
Connection to servera closed.
[student@workstation]$
```

Завершение

На рабочей станции запустите сценарий **lab fs-mount finish**, чтобы завершить это упражнение.

```
[student@workstation ~]$ lab fs-mount finish
```

На этом пошаговое упражнение завершено.

ПОИСК ФАЙЛОВ В СИСТЕМЕ

ЗАДАЧИ

После завершения этого раздела вы сможете искать файлы в смонтированных файловых системах с помощью команды **find** и **locate**.

ПОИСК ФАЙЛОВ

Системному администратору нужны инструменты для поиска файлов, соответствующих определенным критериям, в файловой системе. В этом разделе обсуждаются две команды, которые могут искать файлы в иерархии файловой системы.

- Команда **locate** ищет в предварительно сгенерированном индексе имена файлов или пути к файлам и мгновенно возвращает результаты.
- Команда **find** выполняет поиск файлов в реальном времени путем обхода иерархии файловой системы.

ПОИСК ФАЙЛОВ ПО ИМЕНИ

Команда **locate** находит файлы по имени или пути к файлу. Это быстро, потому что он ищет эту информацию в базе данных местоположения. Однако эта база данных не обновляется в реальном времени, и ее необходимо часто обновлять, чтобы результаты были точными. Это также означает, что **locate** не найдет файлы, созданные с момента последнего обновления базы данных.

База данных местоположения автоматически обновляется каждый день. Однако в любой момент пользователь **root** может выполнить команду **updatedb**, чтобы выполнить немедленное обновление.

```
[root@host ~]# updatedb
```

Команда **locate** ограничивает результаты для непrivилегированных пользователей. Чтобы увидеть получившееся имя файла, пользователь должен иметь разрешение на поиск в каталоге, в котором находится файл.

Искать файлы с **passwd** в имени или пути в деревьях каталогов, читаемых пользователем на хосте.

```
[user@host ~]$ locate passwd
/etc/passwd
/etc/passwd-
/etc/pam.d/passwd
/etc/security/opasswd
/usr/bin/gpasswd
/usr/bin/grub2-mkpasswd-pbkdf2
/usr/bin/lppasswd
/usr/bin/passwd
```

```
...output omitted...
```

Результаты возвращаются, даже если имя файла или путь лишь частично совпадают с поисковым запросом.

```
[root@host ~]# locate image
/etc/selinux/targeted-contexts/virtual_image_context
/usr/bin/grub2-mkimage
/usr/lib/sysimage
/usr/lib/dracut/dracut.conf.d/02-generic-image.conf
/usr/lib/firewalld/services/ovirt-imageio.xml
/usr/lib/grub/i386-pc/lnxboot.image
...output omitted...
```

Параметр **-i** выполняет поиск без учета регистра. С этой опцией все возможные комбинации прописных и строчных букв соответствуют поиску.

```
[user@host ~]$ locate -i messages
...output omitted...
/usr/share/vim/vim80/lang/zh_TW/LC_MESSAGES
/usr/share/vim/vim80/lang/zh_TW/LC_MESSAGES/vim.mo
/usr/share/vim/vim80/lang/zh_TW.UTF-8/LC_MESSAGES
/usr/share/vim/vim80/lang/zh_TW.UTF-8/LC_MESSAGES/vim.mo
/usr/share/vim/vim80/syntax/messages.vim
/usr/share/vim/vim80/syntax/msmessages.vim
/var/log/messages
```

Параметр **-n** ограничивает количество результатов поиска, возвращаемых командой **locate**. В следующем примере результаты поиска, возвращаемые командой **locate**, ограничиваются первыми пятью совпадениями:

```
[user@host ~]$ locate -n 5 snow.png
/usr/share/icons/HighContrast/16x16/status/weather-snow.png
/usr/share/icons/HighContrast/22x22/status/weather-snow.png
/usr/share/icons/HighContrast/24x24/status/weather-snow.png
/usr/share/icons/HighContrast/256x256/status/weather-snow.png
/usr/share/icons/HighContrast/32x32/status/weather-snow.png
```

ПОИСК ФАЙЛОВ В РЕАЛЬНОМ ВРЕМЕНИ

Команда **find** находит файлы, выполняя поиск в реальном времени в иерархии файловой системы. Это медленнее, чем определение местоположения, но более точное. Он также может искать файлы на основе критериев, отличных от имени файла, таких как права доступа к файлу, тип файла, его размер или время его изменения.

Команда **find** просматривает файлы в файловой системе, используя учетную запись пользователя, выполнившего поиск. Пользователь,зывающий команду **find**, должен иметь разрешение на чтение и выполнение в каталоге для проверки его содержимого.

Первый аргумент команды **find** - это каталог для поиска. Если аргумент каталога опущен, **find** запускает поиск в текущем каталоге и ищет совпадения в любом подкаталоге.

Для поиска файлов по имени используйте параметр **-name FILENAME**. С этой опцией **find** возвращает путь к файлам, точно совпадающим с **FILENAME**. Например, чтобы найти файлы с именем **sshd_config**, начиная с каталога **/**, выполните следующую команду:

```
[root@host ~]# find / -name sshd_config  
/etc/ssh/sshd_config
```



ПРИМЕЧАНИЕ

В команде **find** для полных параметров слова используется один дефис, а параметры следуют за аргументом имени пути, в отличие от большинства других команд Linux.

Подстановочные знаки доступны для поиска имени файла и возврата всех результатов, которые частично совпадают. При использовании подстановочных знаков важно заключить в кавычки имя файла, который нужно искать, чтобы терминал не интерпретировал подстановочный знак.

В следующем примере ищите файлы, начинающиеся в каталоге **/** и заканчивающиеся на **.txt**:

```
[root@host ~]# find / -name '*.txt'  
/etc/pki/nssdb/pkcs11.txt  
/etc/brltty/brl-lt-all.txt  
/etc/brltty/brl-mb-all.txt  
/etc/brltty/brl-md-all.txt  
/etc/brltty/brl-mn-all.txt  
...output omitted...
```

Чтобы найти файлы в каталоге **/etc/**, содержащие слово **pass**, в любом месте своего имени на хосте, выполните следующую команду:

```
[root@host ~]# find /etc -name '*pass*'  
/etc/security/opasswd  
/etc/pam.d/passwd  
/etc/pam.d/password-auth  
/etc/passwd-  
/etc/passwd  
/etc/authselect/password-auth
```

Чтобы выполнить поиск по заданному имени файла без учета регистра, используйте параметр **-iname**, за которым следует имя файла для поиска. Для поиска файлов с нечувствительным к регистру текстом, сообщениями в их именах в каталоге **/** на хосте, выполните следующую команду:

```
[root@host ~]# find / -iname '*messages*'  
...output omitted...  
/usr/share/vim/vim80/lang/zh_CN.UTF-8/LC_MESSAGES  
/usr/share/vim/vim80/lang/zh_CN.cp936/LC_MESSAGES  
/usr/share/vim/vim80/lang/zh_TW/LC_MESSAGES  
/usr/share/vim/vim80/lang/zh_TW.UTF-8/LC_MESSAGES  
/usr/share/vim/vim80/syntax/messages.vim  
/usr/share/vim/vim80/syntax/msmessages.vim
```

Поиск файлов на основе владельца файла или разрешения прав для файла

Команда **find** может искать файлы на основе их прав собственности или разрешений. При поиске по владельцу полезными параметрами являются **-user** и **-group**, которые ищут по имени, и **-uid** и **-gid**, которые ищут по **ИД**.

Найдите файлы, принадлежащие пользователю **user**, в каталоге **/home/user** на хосте.

```
[user@host ~]$ find -user user  
. ./.bash_logout  
./.bash_profile  
./.bashrc  
./.bash_history
```

Найдите файлы, принадлежащие пользователю группы **user**, в каталоге **/home/user** на хосте.

```
[user@host ~]$ find -group user  
. ./.bash_logout  
./.bash_profile
```

```
./.bashrc  
./.bash_history
```

Найдите файлы, принадлежащие пользователю с **ID 1000**, в каталоге **/home/user** на хосте.

```
[user@host ~]$ find -uid 1000  
. .  
./.bash_logout  
./.bash_profile  
./.bashrc  
./.bash_history
```

Найдите файлы, принадлежащие группе с идентификатором **1000**, в каталоге **/home/user** на хосте.

```
[user@host ~]$ find -gid 1000  
. .  
./.bash_logout  
./.bash_profile  
./.bashrc  
./.bash_history
```

Параметры **-user** и **-group** можно использовать вместе для поиска файлов, в которых владелец файла и владелец группы различаются. В приведенном ниже примере перечислены файлы, которые принадлежат пользователю **root** и связаны с групповой **mail**.

```
[root@host ~]# find / -user root -group mail  
/var/spool/mail  
...output omitted...
```

Параметр **-perm** используется для поиска файлов с определенным набором разрешений. Разрешения можно описать как восьмеричные значения с некоторой комбинацией **4, 2** и **1** для чтения, записи и выполнения (**read, write, и execute**). Разрешениям может предшествовать знак **/** или **-**.

Числовое разрешение, которому предшествует **/** соответствует файлам, которые имеют хотя бы один бит пользователя, группы или другого для этого набора разрешений. Файл с разрешениями **r--r--r--** не соответствует **/222**, но файл с **rw-r--r--** соответствует. Знак **-** перед разрешением означает, что все три экземпляра этого бита должны быть включены, поэтому ни один из предыдущих примеров не будет соответствовать, но что-то вроде **rw-rw-rw->** будет.

Чтобы использовать более сложный пример, следующая команда соответствует любому файлу, для которого пользователь имеет разрешения на чтение, запись и выполнение, члены группы имеют разрешения на чтение и запись, а другие имеют доступ только для чтения:

```
[root@host ~]# find /home -perm 764
```

Чтобы сопоставить файлы, для которых у пользователя есть как минимум права на запись и выполнение, а у группы есть как минимум права на запись, а у других есть как минимум права на чтение:

```
[root@host ~]# find /home -perm -324
```

Чтобы сопоставить файлы, для которых у пользователя есть разрешения на чтение, или у группы есть хотя бы разрешения на чтение, или у других есть хотя бы доступ на запись:

```
[root@host ~]# find /home -perm /442
```

При использовании с / или - значение **0** работает как подстановочный знак, поскольку означает разрешение как минимум ничего (*permission of at least nothing*).

Чтобы сопоставить любой файл в каталоге **/home/user**, к которому у других есть хотя бы доступ для чтения на хосте, выполните:

```
[user@host ~]$ find -perm -004
```

Найдите все файлы в каталоге **/home/user**, где у прочих(*other*) есть разрешения на запись (**write**) на хосте.

```
[user@host ~]$ find -perm -002
```

Поиск файлов по размеру

Команда **find** может искать файлы, размер которых соответствует размеру, указанному в параметре **-size**, за которым следует числовое значение и единица измерения. Используйте следующий список в качестве единиц с параметром **-size**:

- **k**, для килобайта
- **M**, для мегабайта

- **G**, для гигабайта

В приведенном ниже примере показано, как искать файлы размером 10 мегабайт с округлением в большую сторону.

```
[user@host ~]$ find -size 10M
```

Для поиска файлов размером *более* 10 гигабайт выполните команду:

```
[user@host ~]$ find -size +10G
```

Вывести список всех файлов размером менее 10 килобайт.

```
[user@host ~]$ find -size -10k
```



ВАЖНО

Опции единиц **-size** округляют все до единичных единиц. Например, команда **find -size 1M** показывает файлы размером менее **1 МБ**, поскольку округляет все файлы до **1 МБ**.

Поиск файлов по времени модификации

Параметр **-mmin**, за которым следует время в минутах, выполняет поиск всех файлов, содержимое которых было изменено **n** минут назад в прошлом. Отметка времени файла всегда округляется в меньшую сторону. Он также поддерживает дробные значения при использовании с диапазонами (**+n** и **-n**).

Чтобы найти все файлы, содержимое которых было изменено **120 минут назад** на хосте, выполните команду:

```
[root@host ~]# find / -mmin 120
```

Модификатор **+** перед количеством минут ищет все файлы в **/**, которые были изменены **более n минут назад**. В этом примере перечислены файлы, которые были изменены **более 200 минут назад**.

```
[root@host ~]# find / -mmin +200
```

Модификатор **-** изменяет поиск для поиска всех файлов в каталоге **/**, которые были изменены менее **n** минут назад. В этом примере перечислены файлы, которые были изменены **менее 150 минут назад**.

```
[root@host ~]# find / -mmin -150
```

Поиск файлов по их типу

Параметр **-type** в команде **find** ограничивает область поиска заданным типом файла. Используйте следующий список, чтобы передать необходимые флаги, чтобы ограничить область поиска:

- **f**, для обычного файла
- **d**, для справочника
- **l**, для символьской-ссылки
- **b**, для блочного устройства

Найдите все каталоги в каталоге **/etc** на хосте.

```
[root@host ~]# find /etc -type d  
/etc  
/etc/tmpfiles.d  
/etc/systemd  
/etc/systemd/system  
/etc/systemd/system/getty.target.wants  
...output omitted...
```

Найдите все символические ссылки на хосте.

```
[root@host ~]# find / -type l
```

Сгенерируйте список всех блочных устройств в каталоге **/dev** на хосте:

```
[root@host ~]# find /dev -type b  
/dev/vda1  
/dev/vda
```

Параметр **-links**, за которым следует число, выполняет поиск всех файлов с определенным количеством жестких ссылок. Перед числом может стоять модификатор **+** для поиска файлов с

числом выше заданного числа жестких ссылок. Если перед числом стоит модификатор **-**, поиск ограничивается всеми файлами с числом жестких ссылок меньше заданного числа.

Искать все обычные файлы с более чем одной жесткой ссылкой на хосте:

```
[root@host ~]# find / -type f -links +1
```



РЕКОМЕНДАЦИИ

Справочные страницы **man locate(1)**, **updatedb(8)**, и **find(1)**

УПРАЖНЕНИЯ С ПОШАГОВЫМИ ИНСТРУКЦИЯМИ

ПОИСК ФАЙЛОВ В СИСТЕМЕ

В этом упражнении вы найдете определенные файлы в смонтированных файловых системах с помощью команд **find** и **locate**.

В РЕЗУЛЬТАТЕ

У вас должна быть возможность искать файлы с помощью команд **find** и **locate**.

ПРЕЖДЕ ЧЕМ ВЫ НАЧНЕТЕ

Войдите на рабочую станцию **workstation** как пользователь **student**, используя в качестве пароля слово **student**.

На рабочей станции **workstation** выполните скрипт **lab fs-locate start**. Команда запускает сценарий, который определяет, доступен ли хост, сервер **servera**, в сети.

```
[student@workstation ~]$ lab fs-locate start
```

1. Используйте команду **ssh** для входа на сервер **servera**, как пользователь **student**.

```
[student@workstation ~]$ ssh student@servera  
[student@servera ~]$
```

2. Используйте команду **locate** для поиска файлов на сервере.

2.1. Несмотря на то, что база данных местоположения обновляется автоматически каждый день, убедитесь, что база данных актуальна, вручную запустив обновление на сервере. Используйте команду **sudo updatedb**, чтобы обновить базу данных, используемую командой **locate**.

```
[student@servera ~]$ sudo updatedb  
[sudo] password for student: student  
[student@servera ~]$
```

- 2.2. Найдите файл конфигурации **logrotate.conf**.

```
[student@servera ~]$ locate logrotate.conf  
/etc/logrotate.conf  
/usr/share/man/man5/logrotate.conf.5.gz
```

- 2.3. Найдите файл конфигурации **networkmanager.conf**, игнорируя регистр.

```
[student@servera ~]$ locate -i networkmanager.conf  
/etc/NetworkManager/NetworkManager.conf  
/etc/dbus-1/system.d/org.freedesktop.NetworkManager.conf  
/usr/share/man/man5/NetworkManager.conf.5.gz
```

3. Используйте команду **find** для выполнения поиска в реальном времени на сервере в соответствии со следующими требованиями:

- Искать все файлы в каталоге **/var/lib**, принадлежащие пользователю **chrony**.
- Перечислить все файлы в каталоге **/var**, которые принадлежат пользователю **root**, а владельцем группы является **mail**.
- Список всех файлов в каталоге **/usr/bin**, размер которых превышает **50 КБ**.
- Искать все файлы в каталоге **/home/student**, которые не были изменены **за последние 120 минут**.
- Список всех файлов блочных устройств в каталоге **/dev**.

- 3.1. Используйте команду **find** для поиска всех файлов в каталоге **/var/lib**, которые принадлежат пользователю **chrony**. Используйте команду **sudo**, поскольку файлы в каталоге **/var/lib** принадлежат пользователю **root**.

```
[student@servera ~]$ sudo find /var/lib -user chrony  
[sudo] password for student: student  
/var/lib/chrony  
/var/lib/chrony/drift
```

- 3.2. Перечислите все файлы в каталоге **/var**, которые принадлежат пользователю **root** и связаны с группой **mail**.

```
[student@servera ~]$ sudo find /var -user root -group mail  
/var/spool/mail
```

- 3.3. Перечислите все файлы в каталоге **/usr/bin** с размером файла **более 50 КБ**.

```
[student@servera ~]$ find /usr/bin -size +50k  
/usr/bin/iconv  
/usr/bin/locale  
/usr/bin/localedef  
/usr/bin/cmp  
...output omitted...
```

- 3.4. Найдите в каталоге **/home/student** все файлы, которые не менялись **за последние 120 минут**.

```
[student@servera ~]$ find /home/student -mmin +120  
/home/student/.bash_logout
```

```
/home/student/.bash_profile  
/home/student/.bashrc  
...output omitted...
```

3.5. Список всех файлов блочных устройств в каталоге **/dev**.

```
[student@servera ~]$ find /dev -type b  
/dev/vdb  
/dev/vda3  
/dev/vda2  
/dev/vda1  
/dev/vda
```

4. Выход из сервера **servera**.

```
[student@servera ~]$ exit  
logout  
Connection to servera closed.  
[student@workstation]$
```

Завершение

На рабочей станции запустите сценарий **lab fs-locate finish**, чтобы завершить это упражнение.

```
[student@workstation ~]$ lab fs-locate finish
```

На этом пошаговое упражнение завершено.

ЛАБОРАТОРНАЯ РАБОТА

ДОСТУП К ФАЙЛОВЫМ СИСТЕМАМ LINUX

КОНТРОЛЬНЫЙ СПИСОК РАБОТЫ

В этой лабораторной работе вы смонтируете локальную файловую систему и найдете в ней определенные файлы.

В РЕЗУЛЬТАТЕ

Вы должны быть способны:

- Смонтировать файловую систему.
- Создать отчет об использовании диска.
- Искать файлы в локальной файловой системе.

ПРЕЖДЕ ЧЕМ ВЫ НАЧНЕТЕ

Войдите на рабочую станцию **workstation** как пользователь **student**, используя в качестве пароля слово **student**.

На рабочей станции **workstation** выполните скрипт **lab fs-review start**. Команда запускает сценарий, который определяет, доступен ли хост, сервер **serverb**, в сети.

Сценарий также создает раздел на втором диске, подключенном к **serverb**.

```
[student@workstation ~]$ lab fs-review start
```

1. На сервере **serverb** как **root** определите **UUID** для **/dev/vdb1** и смонтируйте **/dev/vdb1** по его **UUID** в каталоге **/mnt/freespace**.
2. Создайте отчет об использовании диска в каталоге **/usr/share** и сохраните результат в файле **/mnt/freespace/results.txt**.
3. Используйте команду **locate**, чтобы найти все файлы конфигурации **rsyslog.conf** и сохранить результат в файле **/mnt/freespace/search1.txt**.
4. Сохраните результаты поиска всех файлов в каталоге **/usr/share** размером **более 50 МБ** и **менее 100 МБ** в файле **/mnt/freespace/search2.txt**.
5. Выйти с сервера **serverb**.

Оценка

На рабочей станции **workstation** запустите сценарий **lab fs-review grade**, чтобы подтвердить успех выполнения лабораторной работы.

```
[student@workstation ~]$ lab fs-review grade
```

Завершение

На рабочей станции **workstation** запустите сценарий **lab fs-review finish**, чтобы завершить упражнение.

```
[student@workstation ~]$ lab fs-review finish
```

На этом лабораторная работа завершена.

РЕШЕНИЕ

ДОСТУП К ФАЙЛОВЫМ СИСТЕМАМ LINUX

КОНТРОЛЬНЫЙ СПИСОК РАБОТЫ

В этой лабораторной работе вы смонтируете локальную файловую систему и найдете в ней определенные файлы.

В РЕЗУЛЬТАТЕ

Вы должны быть способны:

- Смонтировать файловую систему.
- Создать отчет об использовании диска.
- Искать файлы в локальной файловой системе.

ПРЕЖДЕ ЧЕМ ВЫ НАЧНЕТЕ

Войдите на рабочую станцию **workstation** как пользователь **student**, используя в качестве пароля слово **student**.

На рабочей станции **workstation** выполните скрипт **lab fs-review start**. Команда запускает сценарий, который определяет, доступен ли хост, сервер **serverb**, в сети.

Сценарий также создает раздел на втором диске, подключенном к **serverb**.

```
[student@workstation ~]$ lab fs-review start
```

1. На сервере **serverb** как **root** определите **UUID** для **/dev/vdb1** и смонтируйте **/dev/vdb1** по его **UUID** в каталоге **/mnt/freespace**.

1.1. Используйте команду **ssh** для входа на сервер **serverb**, как пользователь **student**.

```
[student@workstation ~]$ ssh student@serverb  
[student@serverb ~]$
```

1.2. Используйте команду **su**, чтобы переключиться на пользователя **root**.

```
[student@serverb ~]$ su -  
Password: redhat  
[root@serverb ~]#
```

- 1.3. Используйте команду **lsblk**, чтобы определить **UUID** устройства **/dev/vdb1**.

```
[root@serverb ~]# lsblk -fp /dev/vdb
NAME      FSTYPE LABEL UUID                                     MOUNTPOINT
/dev/vdb
└─/dev/vdb1  xfs   a04c511a-b805-4ec2-981f-42d190fc9a65
```

- 1.4. Создайте каталог **/mnt/freespace**.

```
[root@serverb ~]# mkdir /mnt/freespace
```

- 1.5. Смонтируйте устройство **/dev/vdb1**, используя **UUID** в каталоге **/mnt/freespace**.

```
[root@serverb ~]# mount UUID="a04c511a-b805-4ec2-981f-42d190fc9a65" /mnt/freespace
```

- 1.6. Убедитесь, что устройство **/dev/vdb1** смонтировано в каталоге **/mnt/freespace**.

```
[root@serverb ~]# lsblk -fp /dev/vdb1
NAME      FSTYPE LABEL UUID                                     MOUNTPOINT
/dev/vdb
└─/dev/vdb1  xfs   a04c511a-b805-4ec2-981f-42d190fc9a65 /mnt/freespace
```

2. Создайте отчет об использовании диска в каталоге **/usr/share** и сохраните результат в файле **/mnt/freespace/results.txt**.

```
[root@serverb ~]# du /usr/share > /mnt/freespace/results.txt
```

3. Используйте команду **locate**, чтобы найти все файлы конфигурации **rsyslog.conf** и сохранить результат в файле **/mnt/freespace/search1.txt**.

- 3.1. Используйте команду **updatedb**, чтобы обновить базу данных, используемую **locate**.

```
[root@serverb ~]# updatedb
```

- 3.2. Найдите файлы конфигурации **rsyslog.conf** и сохраните результат в файле **/mnt/freespace/search1.txt**.

```
[root@serverb ~]# locate rsyslog.conf > /mnt/freespace/search1.txt
```

- Сохраните результаты поиска всех файлов в каталоге **/usr/share** размером **более 50 МБ и менее 100 МБ** в файле **/mnt/freespace/search2.txt**.

```
[root@serverb ~]# find /usr/share -size +50M -size 100M > \  
> /mnt/freespace/search2.txt
```

- Выйти с сервера **serverb**.

```
[root@serverb ~]$ exit  
logout  
[student@serverb ~]$ exit  
logout  
Connection to serverb closed.  
[student@workstation]$
```

Оценка

На рабочей станции **workstation** запустите сценарий **lab fs-review grade**, чтобы подтвердить успех выполнения лабораторной работы.

```
[student@workstation ~]$ lab fs-review grade
```

Завершение

На рабочей станции **workstation** запустите сценарий **lab fs-review finish**, чтобы завершить упражнение.

```
[student@workstation ~]$ lab fs-review finish
```

На этом лабораторная работа завершена.

РЕЗЮМЕ

В этой главе вы узнали:

- Устройства хранения представлены особым типом файлов, называемым блочным устройством.
- Команда **df** сообщает об общем дисковом пространстве, используемом дисковом пространстве и свободном дисковом пространстве во всех смонтированных обычных файловых системах.
- Команда **mount** позволяет пользователю **root** вручную монтировать файловую систему.
- Все процессы должны прекратить доступ к точке монтирования, чтобы успешно размонтировать устройство.
- Съемные запоминающие устройства при использовании графической среды монтируются в каталоге **/run/media**.
- Команда **find** выполняет поиск в реальном времени в локальных файловых системах для поиска файлов на основе критериев поиска.

ГЛАВА 16

АНАЛИЗ СЕРВЕРОВ И ПОЛУЧЕНИЕ ПОДДЕРЖКИ

ЦЕЛЬ

Исследуйте и решайте проблемы в веб-интерфейсе управления, пользуясь поддержкой Red Hat для решения проблем.

ЗАДАЧИ

- Активируйте интерфейс управления веб-консолью для удаленного управления и мониторинга производительности сервера Red Hat Enterprise Linux.
- Опишите ключевые ресурсы, доступные на портале для клиентов Red Hat, и найдите информацию в документации Red Hat и базе знаний.
- Анализируйте серверы на наличие проблем, исправляйте или устраняйте их и подтверждайте решение с помощью Red Hat Insights.

РАЗДЕЛЫ

- Анализ и управление удаленными серверами (и упражнения с пошаговыми инструкциями)
- Получение помощи на портале для клиентов Red Hat (и упражнения с пошаговыми инструкциями)
- Выявление и решение проблем с помощью Red Hat Insights (и контрольный опрос)

АНАЛИЗ И УПРАВЛЕНИЕ УДАЛЕННЫМИ СЕРВЕРАМИ

ЗАДАЧИ

После завершения этого раздела вы сможете активировать интерфейс управления веб-консолью для удаленного управления и мониторинга производительности сервера Red Hat Enterprise Linux.

ОПИСАНИЕ ВЕБ-КОНСОЛИ

Веб-консоль - это веб-интерфейс управления Red Hat Enterprise Linux 8, предназначенный для управления и мониторинга ваших серверов. Он основан на сервисе **Cockpit** с открытым исходным кодом.

Вы можете использовать веб-консоль для отслеживания системных журналов и просмотра графиков производительности системы.

Кроме того, вы можете использовать свой веб-браузер для изменения настроек с помощью графических инструментов в интерфейсе веб-консоли, включая полнофункциональный интерактивный сеанс терминала.

ВКЛЮЧЕНИЕ ВЕБ-КОНСОЛИ

Red Hat Enterprise Linux 8 устанавливает веб-консоль по умолчанию во всех вариантах установки, кроме минимальной.

Включите (Enable) и запустите (start) службу **cockpit.socket**, которая запускает веб-сервер. Этот шаг необходим, если вам нужно подключиться к системе через веб-интерфейс.

```
[user@host ~]$ sudo systemctl enable --now cockpit.socket
Created symlink /etc/systemd/system/sockets.target.wants/cockpit.socket -> /usr/
lib/systemd/system/cockpit.socket.
```

Если вы используете настраиваемый профиль брандмауэра, вам необходимо добавить службу cockpit в **firewalld**, чтобы открыть порт **9090** в брандмауэре:

```
[user@host ~]$ sudo firewall-cmd --add-service=cockpit --permanent
success
[user@host ~]$ sudo firewall-cmd --reload
Success
```

ВХОД В ВЕБ-КОНСОЛЬ

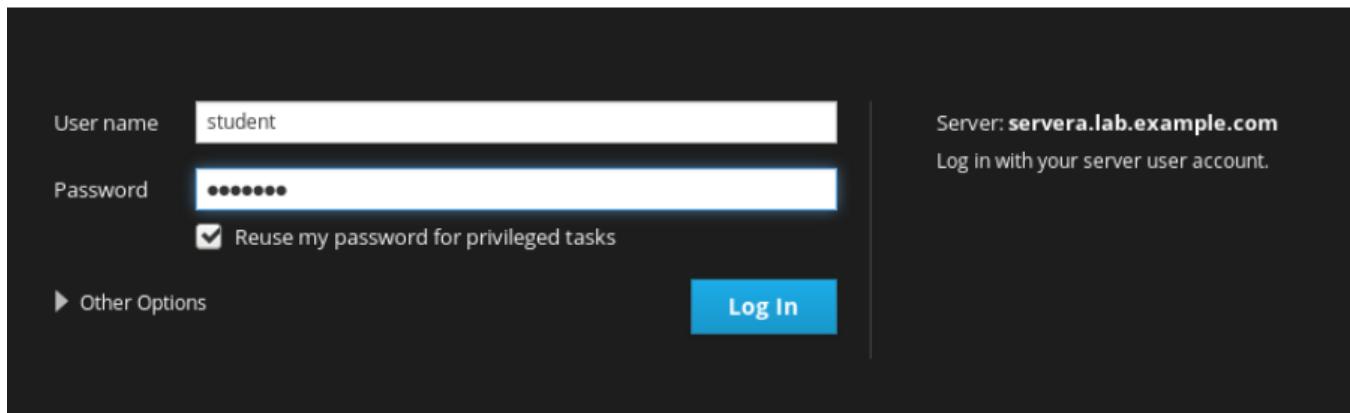
Веб-консоль предоставляет собственный веб-сервер. Запустите Firefox, чтобы войти в веб-консоль. Вы можете войти в систему, используя имя пользователя и пароль любой локальной учетной записи в системе, включая пользователя root.

Откройте **https://servername:9090** в своем веб-браузере, где *servername* - это имя хоста или **IP-адрес** вашего сервера. Соединение будет защищено сеансом TLS. По умолчанию система будет установлена с самозаверяющим сертификатом TLS, и при первом подключении ваш веб-браузер, вероятно, отобразит предупреждение системы безопасности. На странице руководства **cockpit**-

ws(8) приведены инструкции по замене сертификата TLS на сертификат, подписанный должным образом.

Введите свое имя пользователя и пароль на экране входа в систему.

Рисунок 16.1: Экран входа в веб-консоль



При желании выберите параметр **повторно (Reuse)** использовать мой пароль для привилегированных задач. Это позволяет вам выполнять команды с привилегиями **sudo**, появляется возможность выполнять такие задачи, как изменение системной информации или настройка новых учетных записей.

Щелкните **Войти (Log In)**.

Веб-консоль отображает имя пользователя в правой части строки заголовка. Если вы выберете параметр «**Повторно использовать мой пароль для привилегированных задач**», значок «**Привилегированные**» отобразится слева от имени пользователя.

Рисунок 16.2: Стока заголовка привилегированного пользователя



Если вы вошли в систему как непривилегированный пользователь, значок «**Привилегированные (Privileged)**» не отображается.

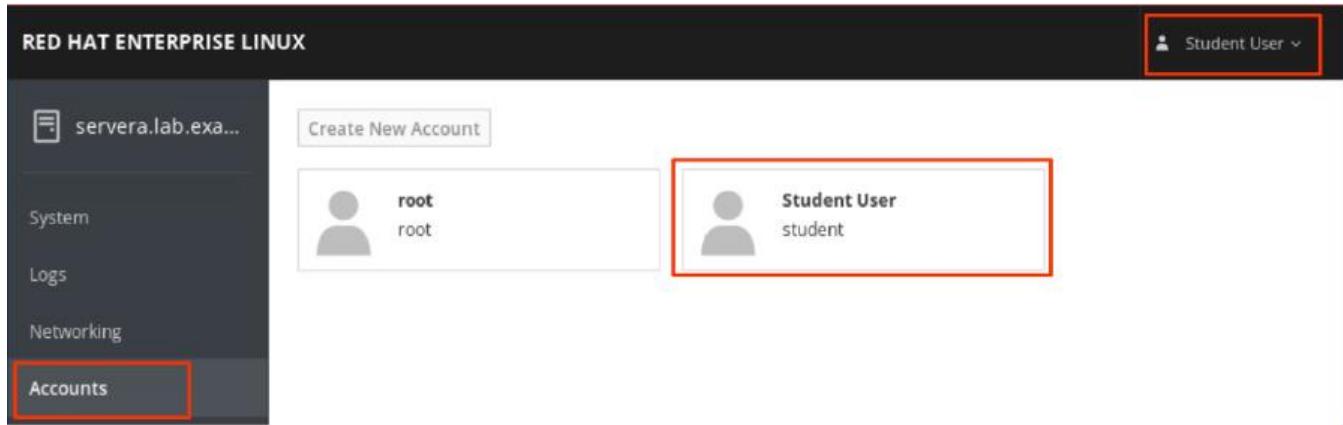
Рисунок 16.3: Стока заголовка непривилегированного пользователя



ИЗМЕНЕНИЕ ПАРОЛЕЙ

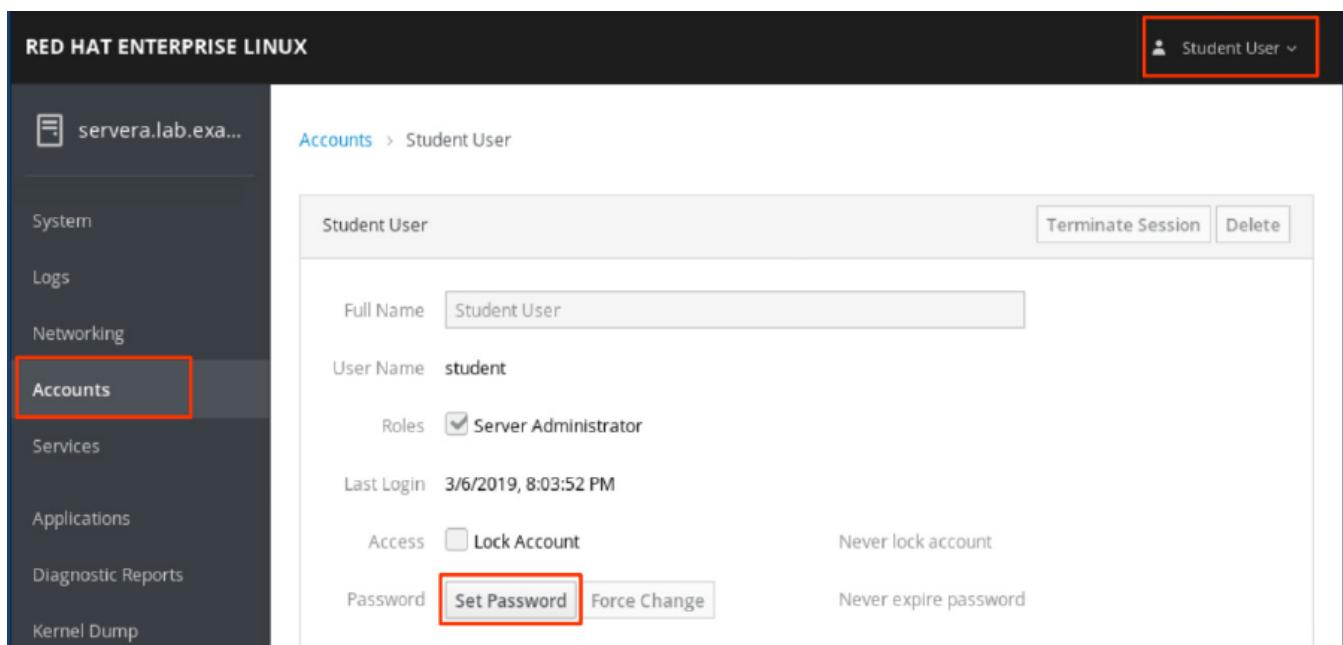
Привилегированные и непривилегированные пользователи могут изменять свои пароли во время входа в веб-консоль. Щелкните Учетные записи (**Accounts**) на левой панели навигации. Нажмите на ярлык своей учетной записи, чтобы открыть страницу сведений об учетной записи.

Рисунок 16.4: Отображение учетных записей пользователей



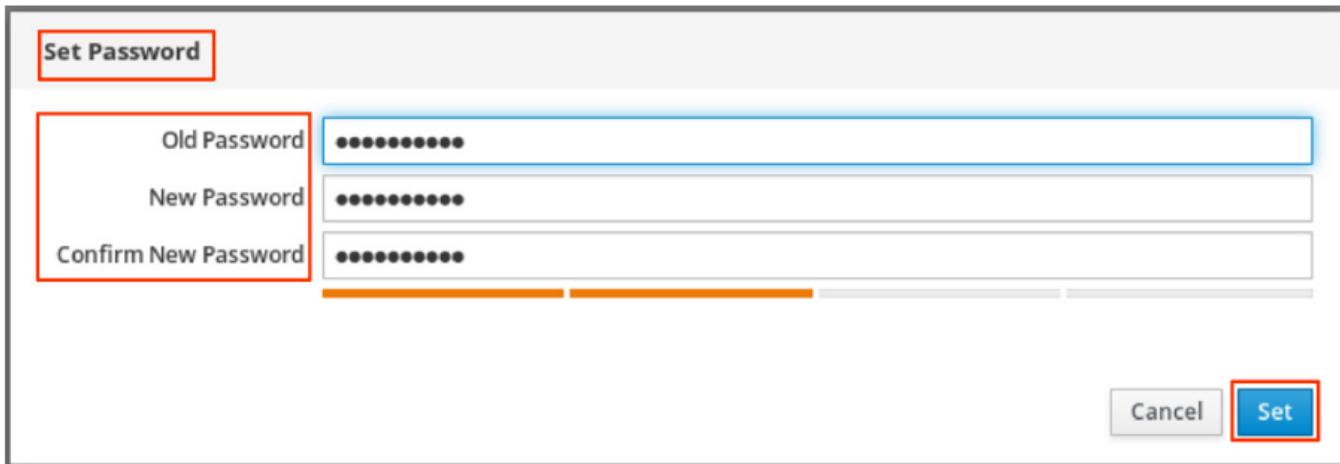
Как непривилегированный пользователь, вы можете устанавливать или сбрасывать пароль и управлять общедоступными ключами SSH. Чтобы установить или сбросить пароль, нажмите кнопку «Установить пароль (Set Password)».

Рисунок 16.5: Детали учетной записи пользователя



Введите свою информацию в поля Старый пароль (**Old Password**), Новый пароль (**New Password**) и Подтвердите новый пароль (**Confirm New Password**). Щелкните Установить (**Set**), чтобы активировать новый пароль.

Рисунок 16.6: Установка или сброс паролей



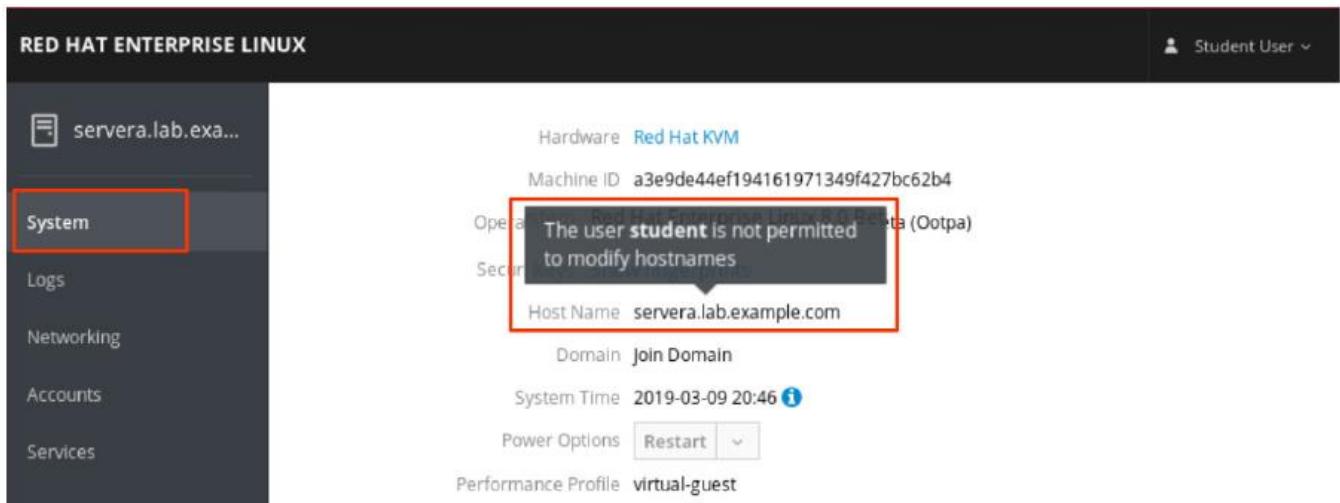
УСТРАНЕНИЕ НЕПОЛАДОК С ВЕБ-КОНСОЛЬ

Веб-консоль - мощный инструмент для устранения неполадок. Вы можете отслеживать основную системную статистику в режиме реального времени, проверять системные журналы и быстро переключаться на сеанс терминала в веб-консоли для сбора дополнительной информации из интерфейса командной строки.

Мониторинг системной статистики в реальном времени

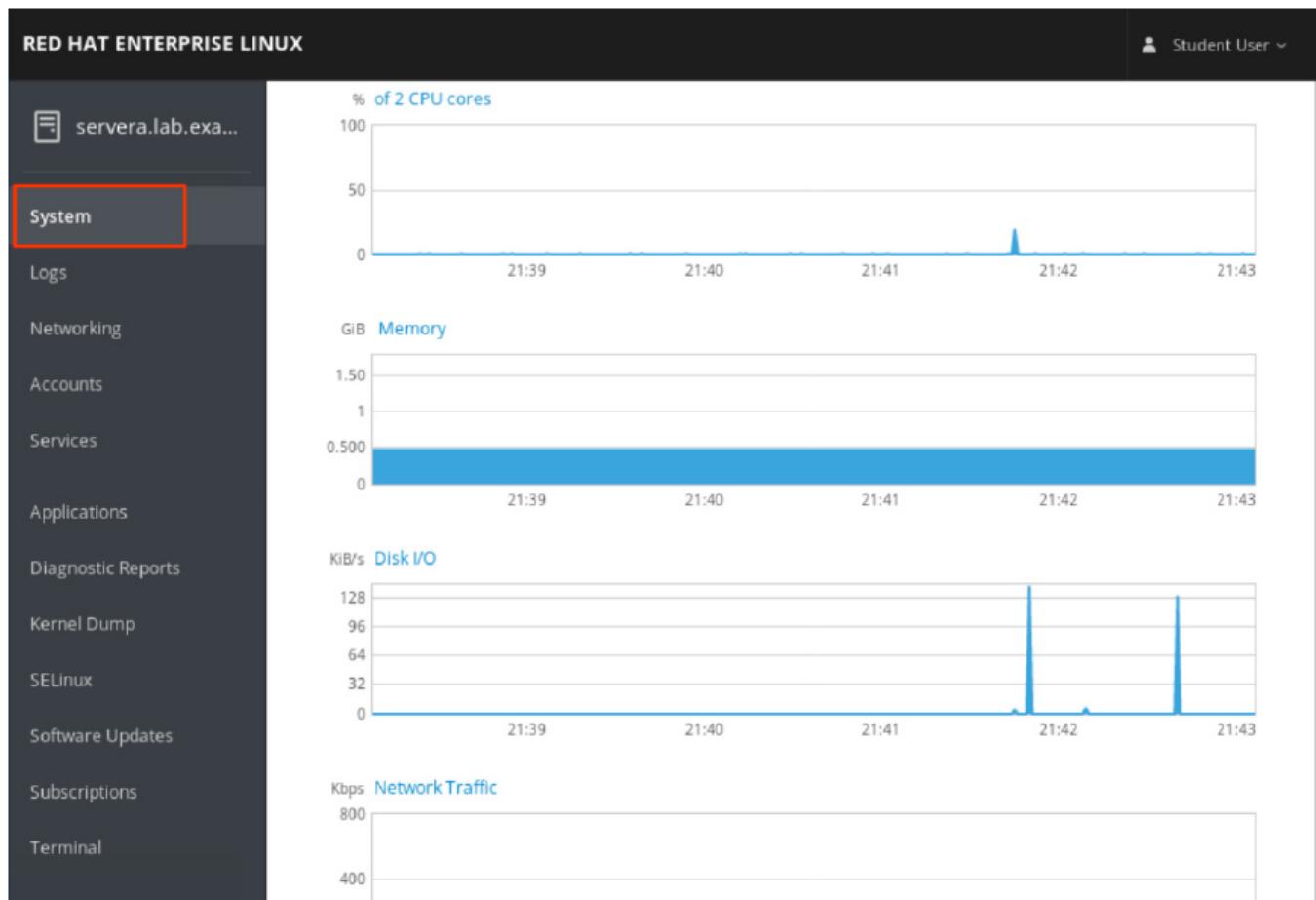
Щелкните Система (**System**) на левой панели навигации, чтобы просмотреть информацию о системе, такую как тип оборудования, операционная система, имя хоста и т.д. Обратите внимание, что, если вы вошли в систему как непrivилегированный пользователь, вы видите всю информацию, но не имеете права изменять значения. На следующем изображении показана верхняя часть страницы параметров системного меню.

Рисунок 16.7: Страница системной информации непrivилегированного пользователя



Прокрутите страницу «Информация о системе (System information)» вниз, чтобы просмотреть графики текущей производительности системы для работы ЦП, использования памяти, дискового ввода-вывода и использования сети.

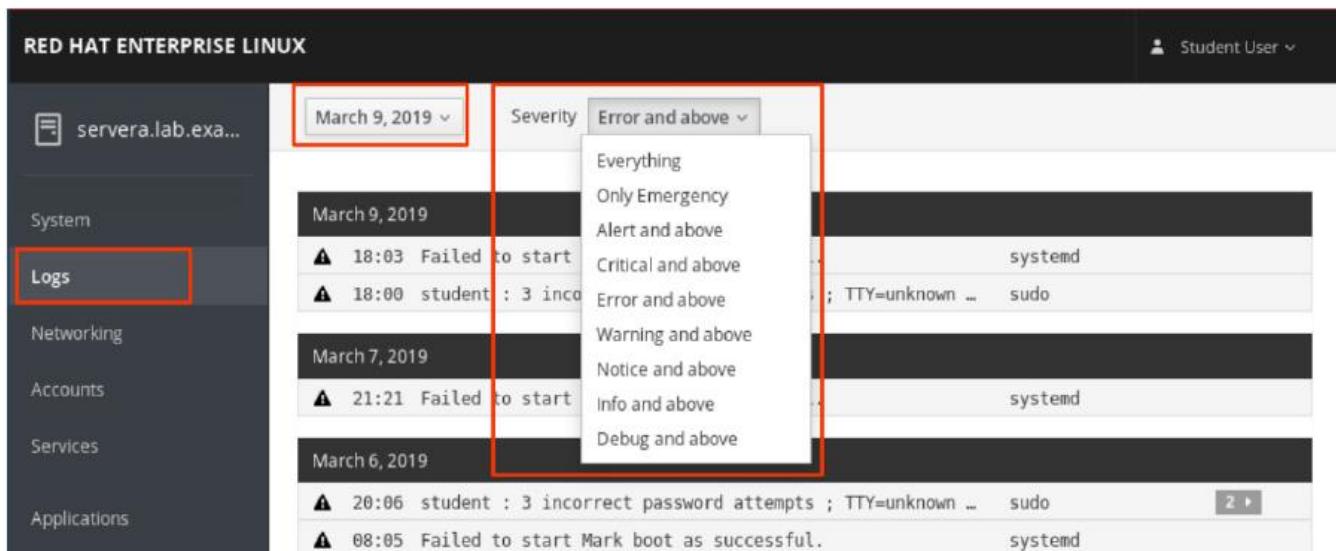
Рисунок 16.8: Показатели производительности системы непrivилегированного пользователя



Проверка и фильтрация событий системного журнала

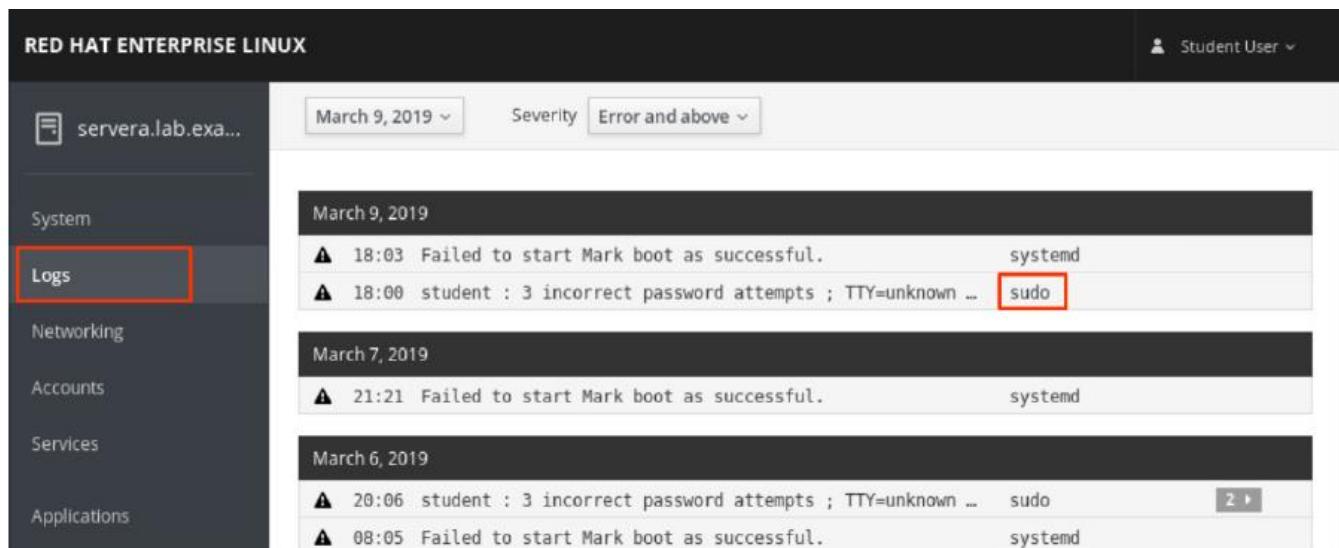
Журналы на левой панели навигации обеспечивают доступ к инструментам анализа системных журналов. Вы можете использовать экран для фильтрации сообщений журнала на основе диапазона дат регистрации, уровня серьезности или того и другого. Веб-консоль использует текущую дату по умолчанию, но вы можете изменить ее, указав любой диапазон дат, щелкнув раскрывающееся меню даты и выбрав соответствующий диапазон. Точно так же в раскрывающемся меню «Приоритет (Severity)» представлены варианты от «Все (Everything)» до более конкретных условий серьезности, таких как «Тревога (Alert)» и выше, «Отладка (Debug)» и выше и т.д.

Рисунок 16.9: Выбор важности журнала



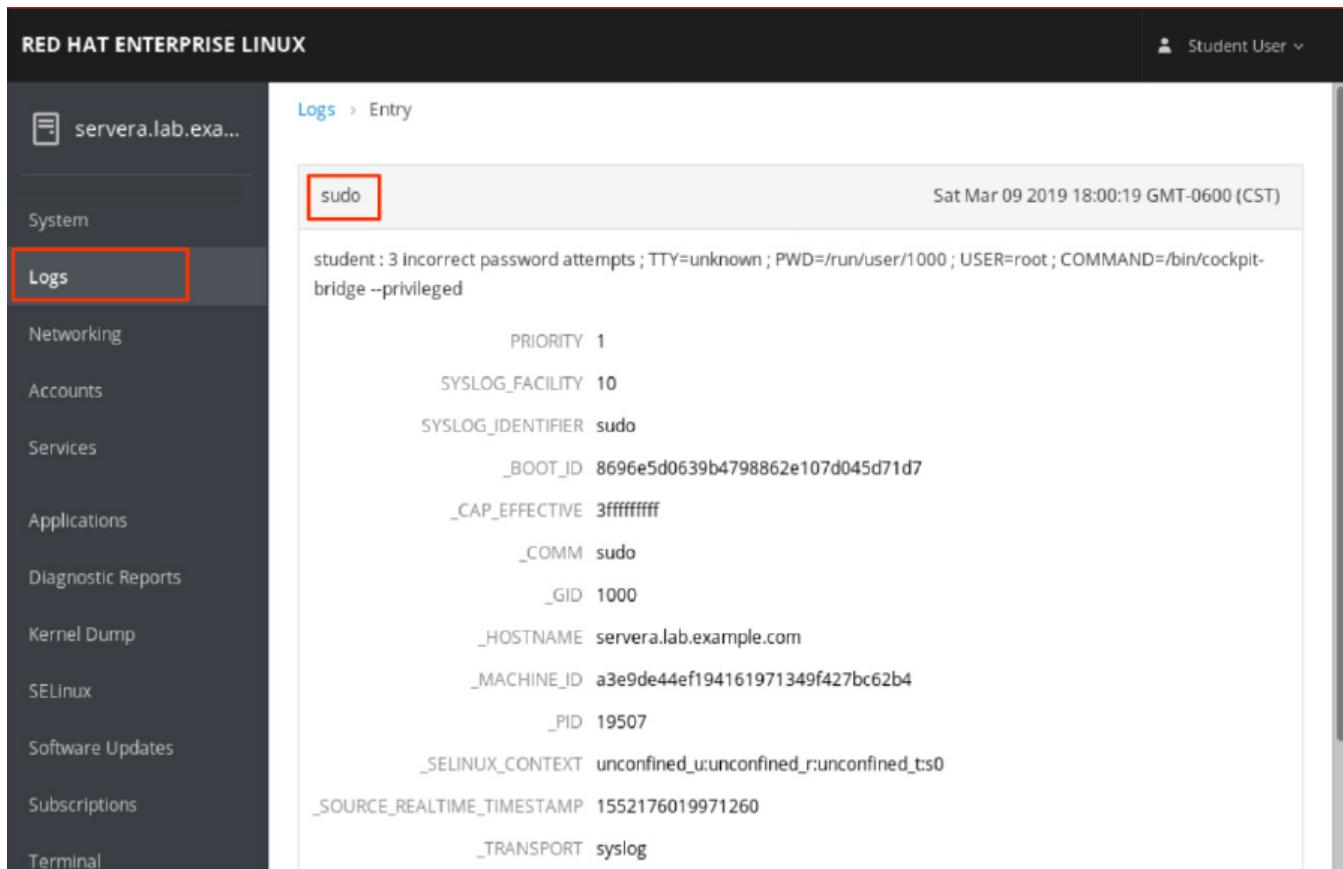
Щелкните строку, чтобы просмотреть подробные сведения об отчете журнала. В приведенном ниже примере обратите внимание на первую строку, которая сообщает о сообщении журнала **sudo**.

Рисунок 16.10: Выбор записи журнала



В приведенном ниже примере показаны подробности, отображаемые при нажатии на строку **sudo**. Подробности отчета включают выбранную запись журнала (**sudo**), дату, время, приоритет и средство системного журнала записи журнала, имя хоста системы, которая сообщила сообщение журнала, и многое другое.

Рисунок 16.11: Детали записи журнала



Запуск команд из терминального сеанса

Терминал на левой панели навигации обеспечивает доступ к полнофункциональному сеансу терминала в интерфейсе веб-консоли. Это позволяет запускать произвольные команды для управления системой и работы с ней, а также выполнять задачи, не поддерживаемые другими инструментами, предоставляемыми веб-консолью.

На следующем изображении показаны примеры общих команд, используемых для сбора дополнительной информации. Перечисление содержимого каталога `/var/log` обеспечивает напоминания о файлах журналов, в которые содержится ценная информация.

Команда `id` предоставляет быструю информацию, например о членстве в группе, которая может помочь в устранении неполадок, связанных с ограничениями доступа к файлам. Команда `ps au` обеспечивает быстрый просмотр процессов, запущенных в терминале, и пользователя, связанного с процессом.

Рисунок 16.12: Устранение неполадок через терминал с правами непrivилегированного пользователя.

The screenshot shows the Red Hat Enterprise Linux web console interface. On the left, a sidebar lists navigation options: System, Logs, Networking, Accounts, Services, Applications, Diagnostic Reports, Kernel Dump, SELinux, Software Updates, Subscriptions, and Terminal. The 'Terminal' option is highlighted with a red box. The main area is a terminal window titled 'servera.lab.exa...'. It displays a command-line session:

```
[student@servera ~]$ ls /var/log
anaconda      cron-20190310      hawkey.log-20190310  rhsm
audit         dnf.librepo.log    lastlog             samba
boot.log      dnf.librepo.log-20190310  maillog            secure
boot.log-20190306  dnf.log        maillog-20190310   secure-20190310
btmp          dnf.log-20190310    messages           spooler
chrony        dnf.rpm.log      messages-20190310  spooler-20190310
cloud-init.log dnf.rpm.log-20190310 private            sssd
cloud-init-output.log  firewalld     qemu-ga            tuned
cron          hawkey.log      README              wtmp

[student@servera ~]$
[student@servera ~]$ id
uid=1000(student) gid=1000(student) groups=1000(student),10(wheel) context=unconfined_u:unconfined_r:un
confined_t:s0
[student@servera ~]$
[student@servera ~]$ ps au
USER      PID %CPU %MEM VSZ RSS TTY      STAT START  TIME COMMAND
root      1255  0.0  0.1 225392 1996 ttyS0  Ss+ Mar05 0:00 /sbin/agetty -o -p -- \u --keep-baud 1
root      2048  0.0  0.2 236164 5068 ttym1  Ss+ Mar05 0:00 -bash
student  19484  0.0  0.2 233932 4932 pts/0   Ss  Mar09 0:00 /bin/bash -i
student  21243  0.0  0.2 266920 3748 pts/0   R+  08:31 0:00 ps au

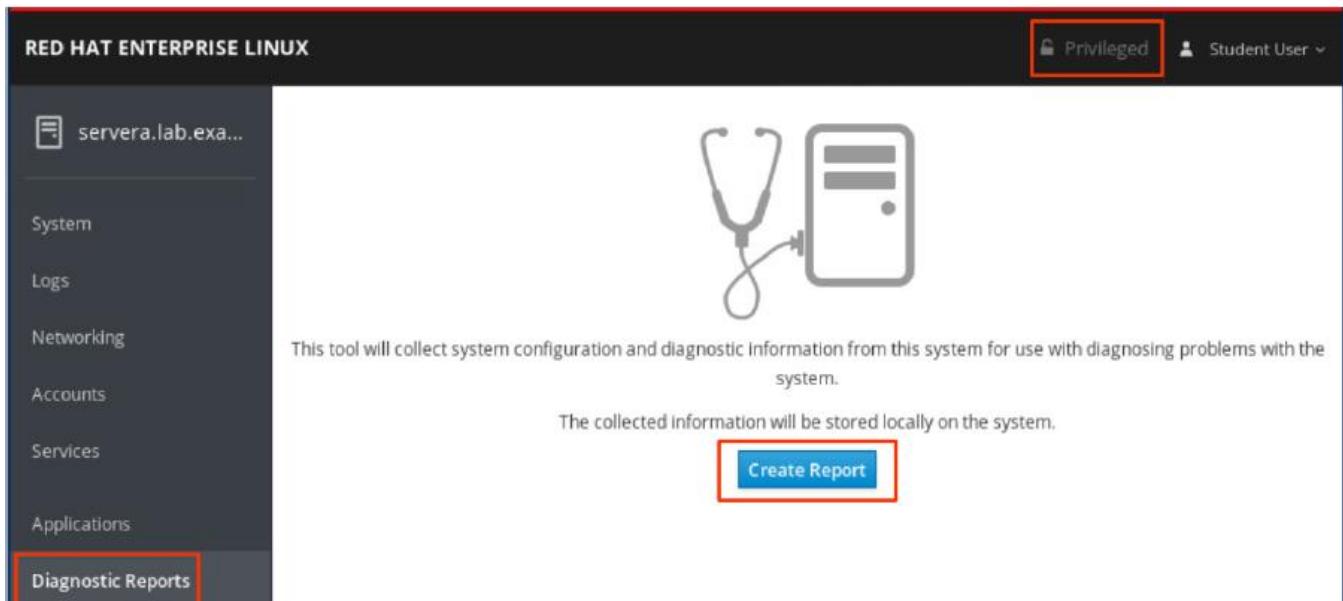
[student@servera ~]$
```

Создание диагностических отчетов

Диагностический отчет - набор сведений о конфигурации, системной информации и диагностической информации из системы Red Hat Enterprise Linux. Данные, собранные в завершенном отчете, включают системные журналы и отладочную информацию, которая может использоваться для устранения проблем.

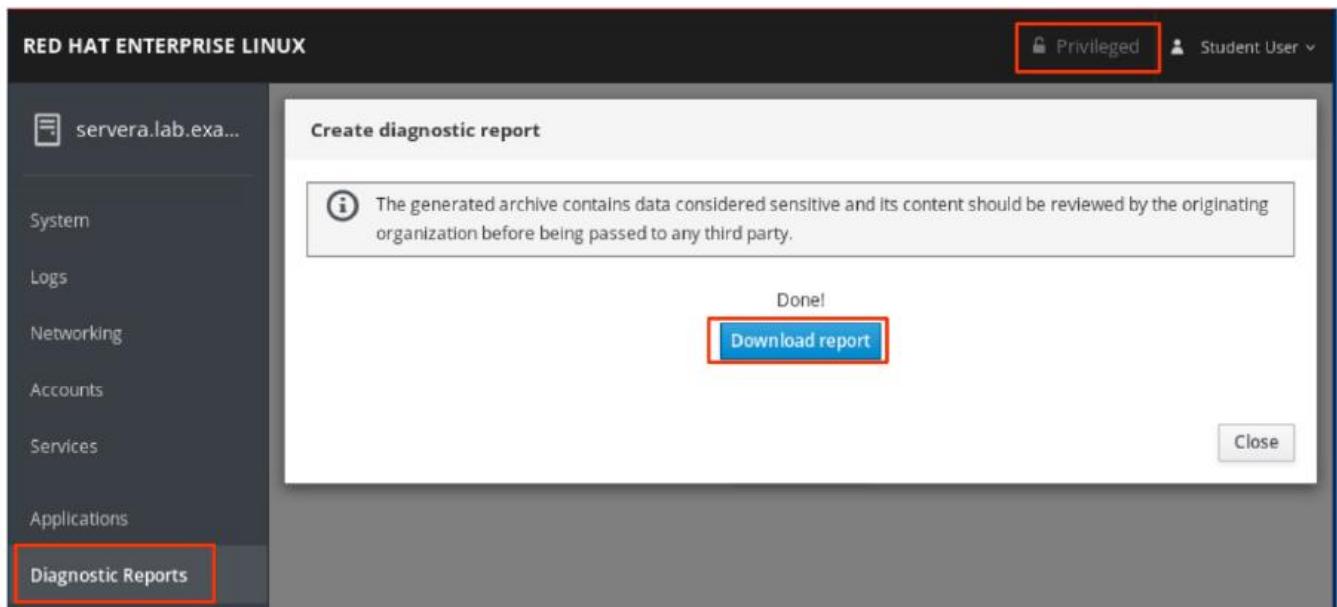
Войдите в веб-консоль как привилегированный пользователь. Щелкните Диагностические отчеты (**Diagnostic Reports**) на левой панели навигации, чтобы открыть страницу, на которой создаются эти отчеты. Нажмите кнопку «Создать отчет (**Create Report**)», чтобы создать новый диагностический отчет.

Рисунок 16.13: Создание диагностического отчета



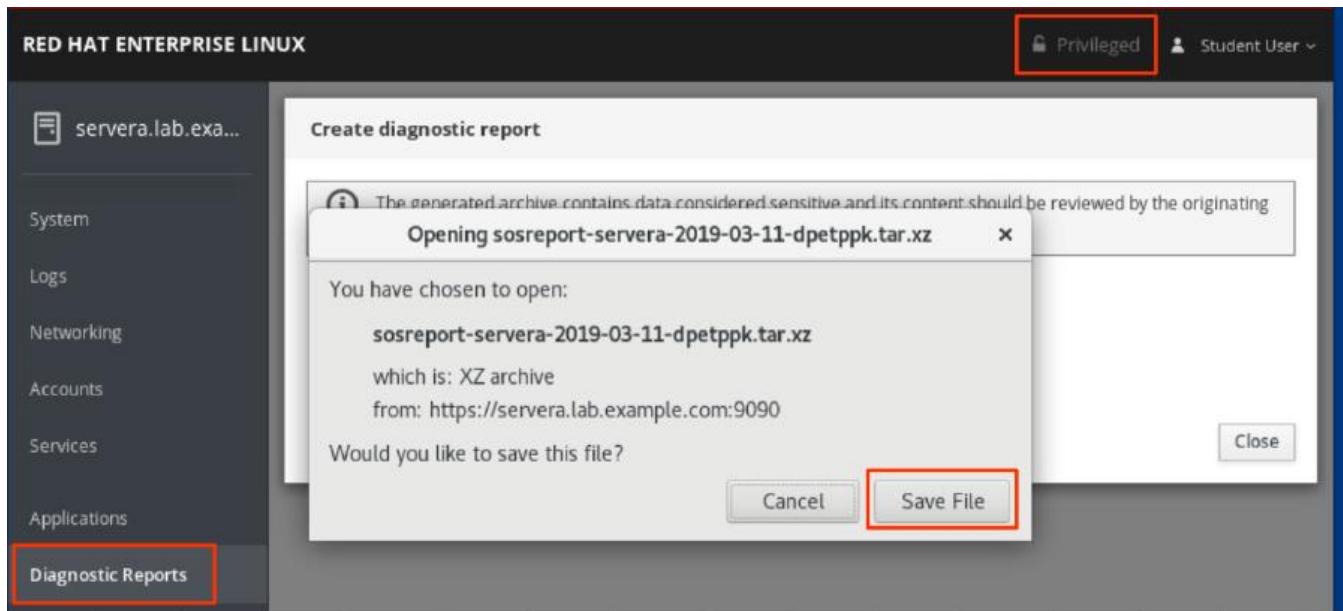
В интерфейсе отобразиться Готово! (**Done!**) когда отчет будет готов. Нажмите кнопку Загрузить отчет (**Download report**), чтобы сохранить отчет.

Рисунок 16.14: Загрузить готовый отчет



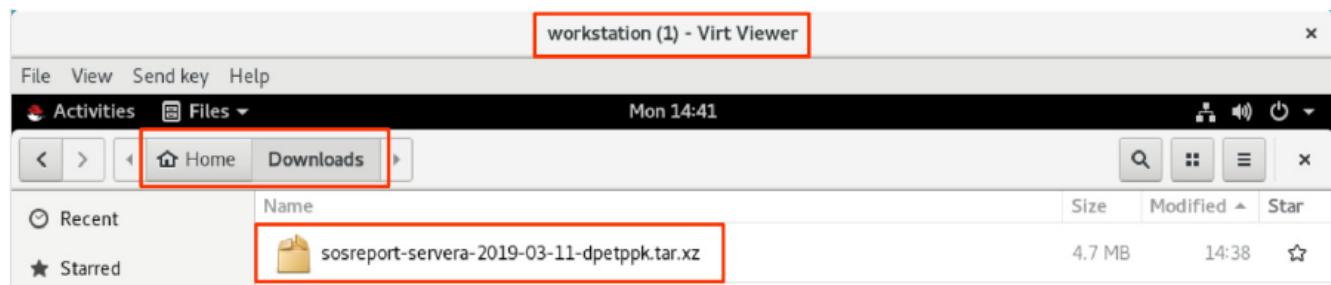
Щелкните Сохранить файл (**Save File**), чтобы завершить процесс.

Рисунок 16.15: Сохранить диагностический отчет



Заполненный отчет сохраняется в каталоге загрузок (**Downloads**) в системе, в которой размещен веб-браузер, используемый для доступа к веб-консоли. В этом примере хост - **workstation**.

Рисунок 16.16: Доступ к завершенному отчету



УПРАВЛЕНИЕ СИСТЕМНЫМИ СЕРВИСАМИ С ВЕБ-КОНСОЛИ

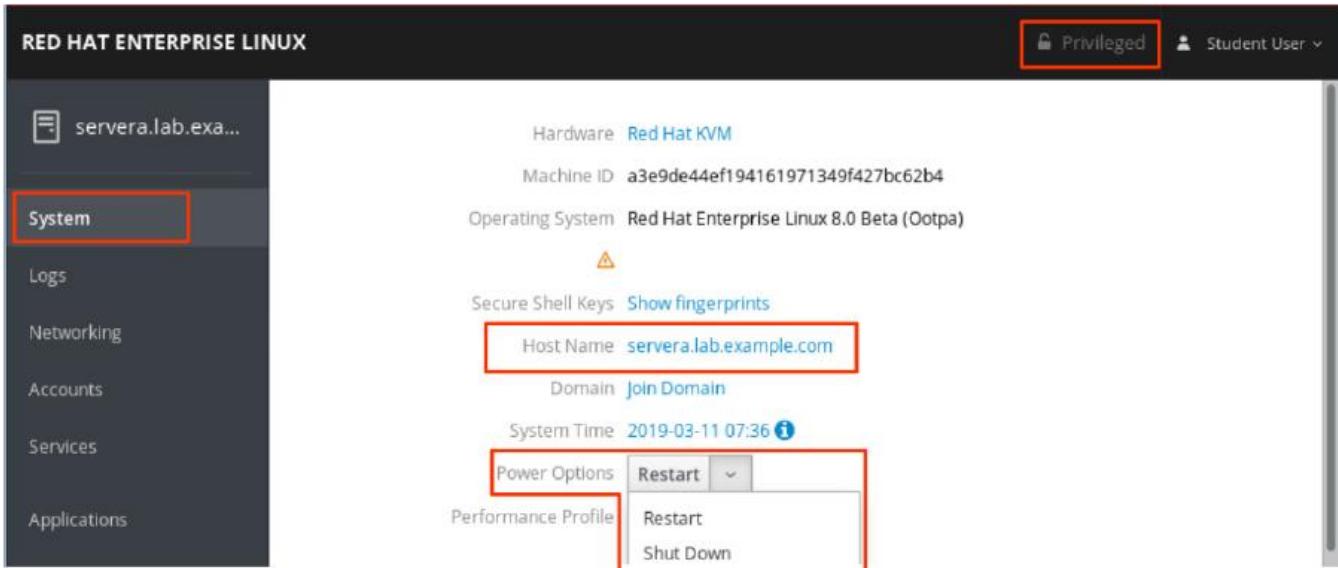
В качестве привилегированного пользователя веб-консоли вы можете останавливать, запускать, включать и перезапускать системные службы. Кроме того, вы можете настраивать сетевые интерфейсы, настраивать службы брандмауэра, администрировать учетные записи пользователей и многое другое. На следующих изображениях показаны распространенные примеры использования инструментов управления веб-консоли.

Параметры питания системы

Веб-консоль позволяет перезапустить или выключить систему. Войдите в веб-консоль как привилегированный пользователь. Щелкните Система (**System**) на левой панели навигации, чтобы получить доступ к параметрам питания системы.

Выберите нужный вариант в раскрывающемся меню «Электропитание (**Power Options**)», чтобы либо перезапустить (**Restart**), либо выключить (**Shut Down**) систему.

Рисунок 16.17: Варианты питания системы

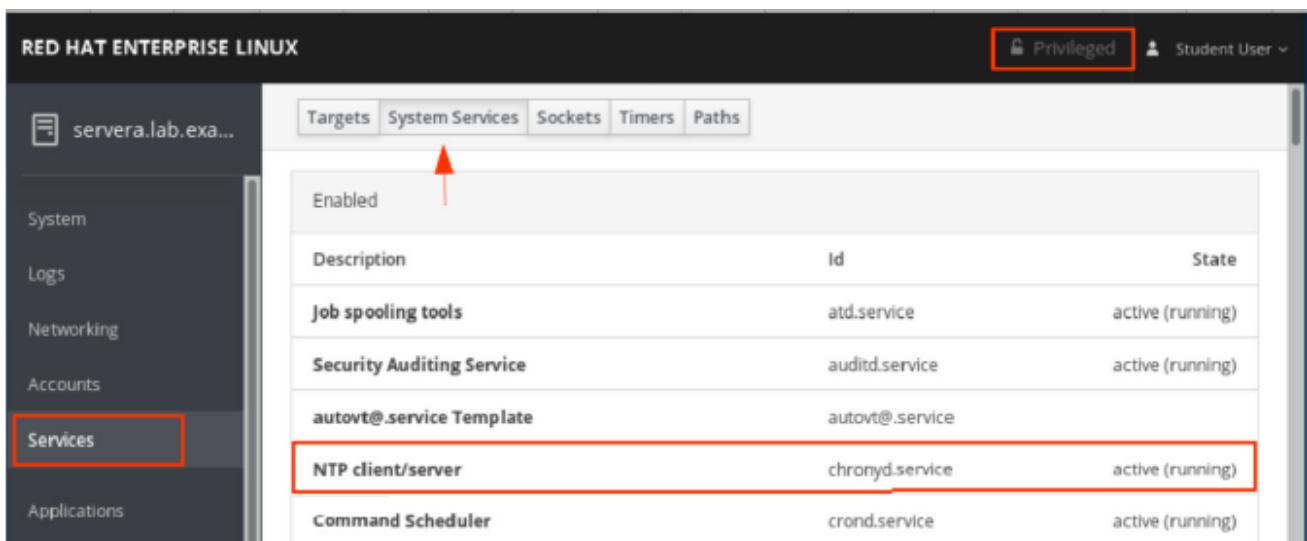


Управление запущенными системными службами

Вы можете запускать, включать, отключать и останавливать службы с помощью графических инструментов в веб-консоли. Щелкните «Службы (Services)» на левой панели навигации, чтобы перейти к начальной странице служб веб-консоли. Для управления службами нажмите кнопку «Системные службы (System Services)», расположенную в верхней правой части начальной страницы службы. Службы отображаются в разделах, помеченных как «Включено (Enabled)», «Отключено (Disabled)» и «Статический (Static)». Прокрутите страницу, чтобы выбрать службу, которой хотите управлять.

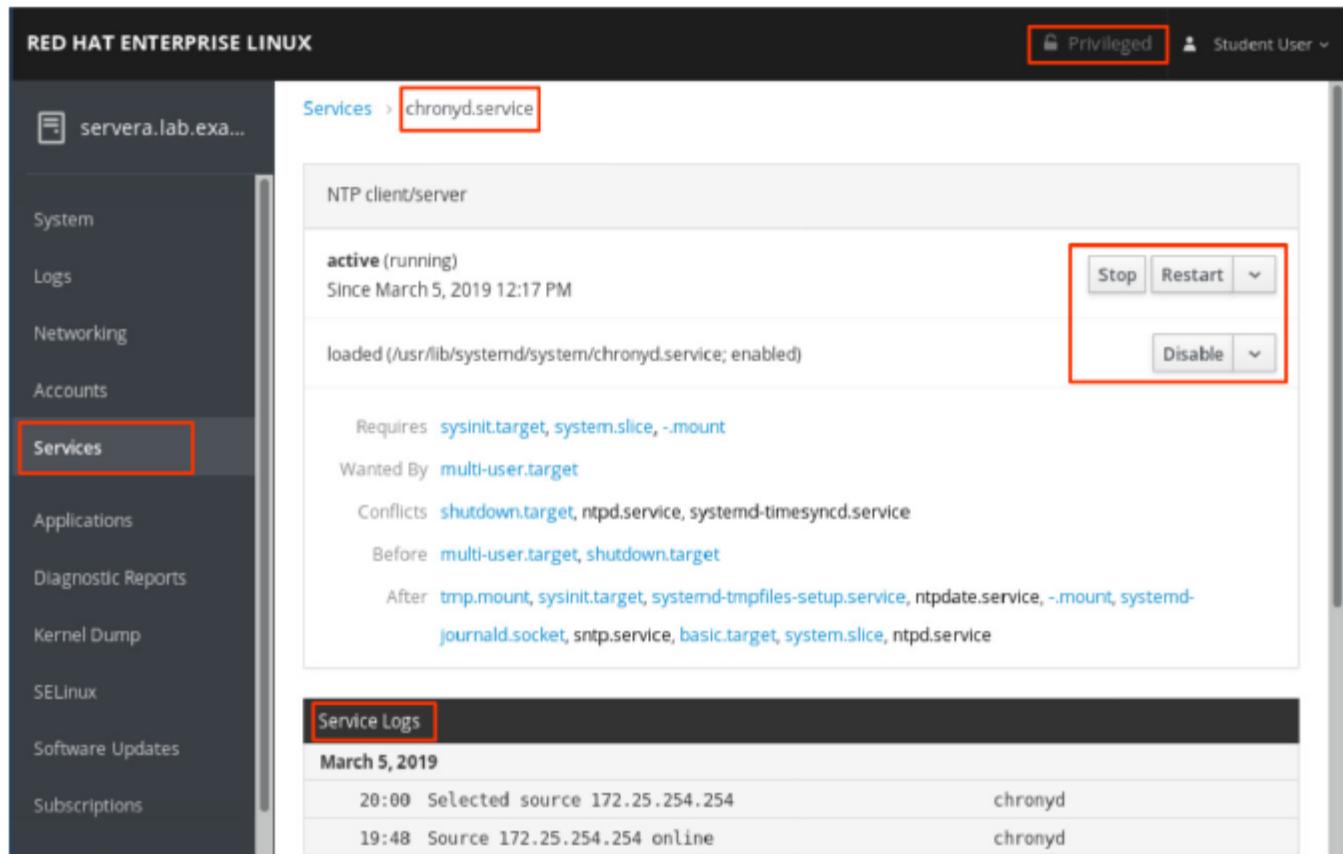
В приведенном ниже примере выберите строку **chronyd.service**, чтобы открыть страницу управления службой.

Рисунок 16.18: Службы: Первоначальный вид



Нажмите нужную кнопку в правой части интерфейса управления службой, чтобы остановить (**stop**), перезапустить (**restart**) или отключить (**disable**) службу. В этом представлении служба уже запущена, поэтому кнопка запуска (**start**) не отображается. Дополнительную информацию, относящуюся к сервисам, можно получить, щелкнув любую из выделенных ссылок или прокрутив Журналы сервисов (**Service Logs**), отображаемые под разделом управления сервисами.

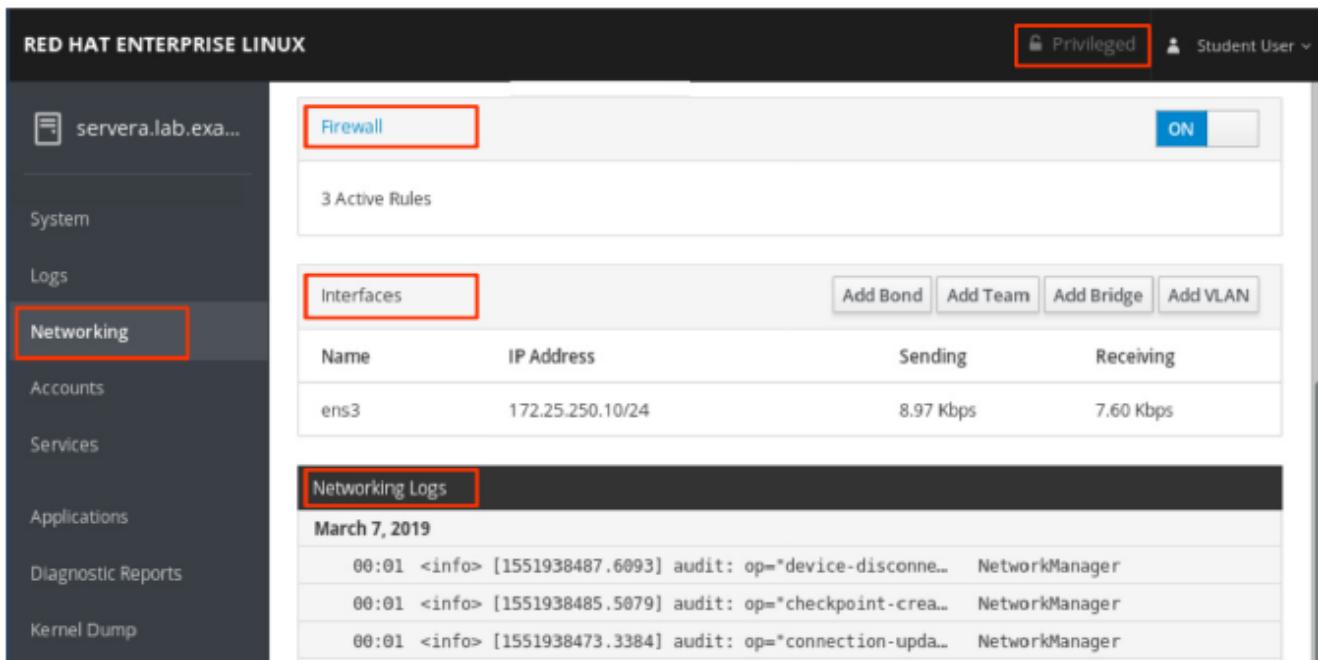
Рисунок 16.19: Сервисы: подробные сведения о сервисе и интерфейс управления



Настройка сетевых интерфейсов и Firewalld

Чтобы управлять правилами брандмауэра и сетевыми интерфейсами, щелкните Сеть (**Networking**) на левой панели навигации. В следующем примере показано, как собирать информацию о сетевых интерфейсах и как ими управлять.

Рисунок 16.20: Сеть: Первоначальный вид



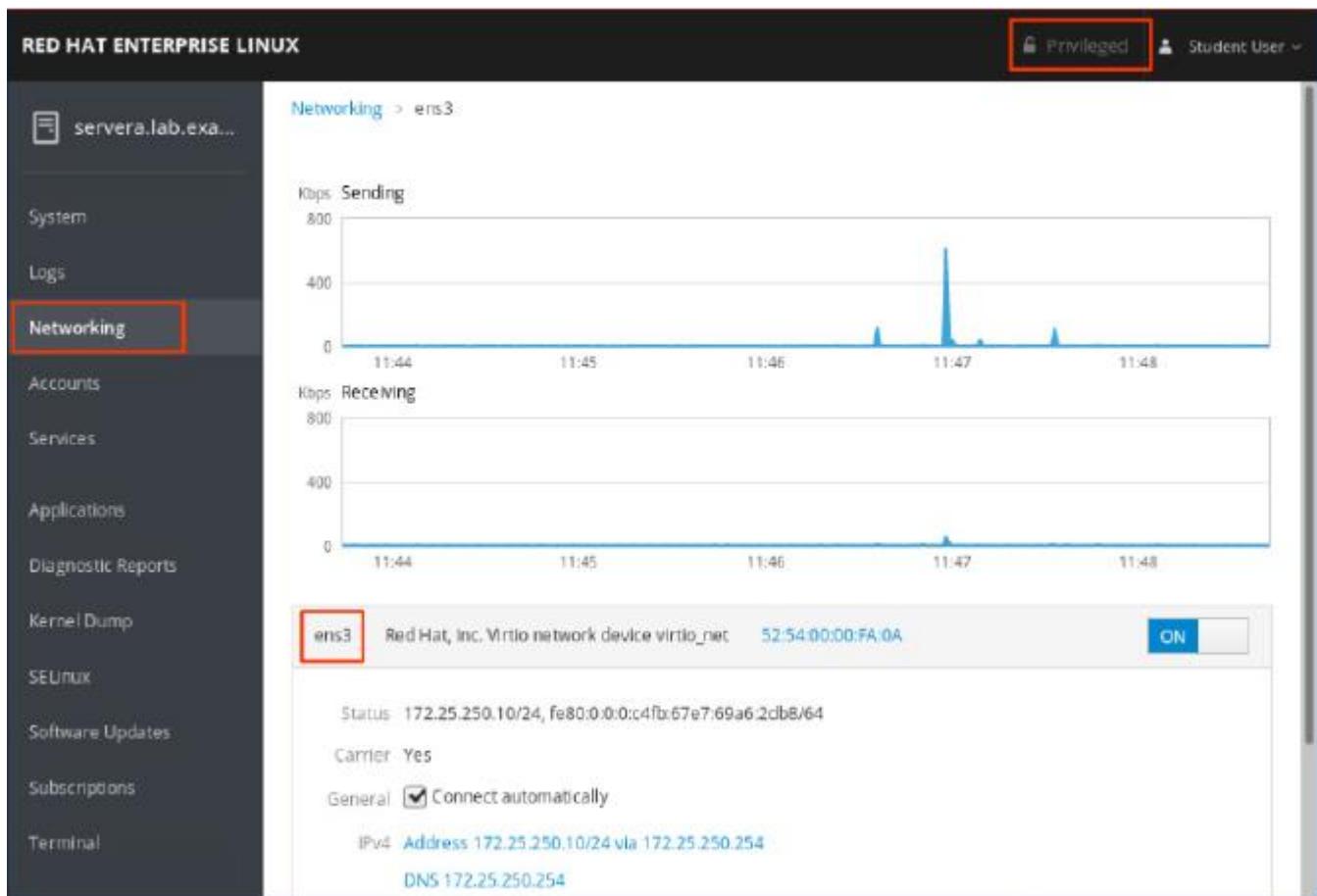
Щелкните имя нужного интерфейса в разделе «Интерфейсы (Interfaces)», чтобы перейти на страницу управления. В этом примере выбран интерфейс **ens3**.

Рисунок 16.21: Сеть: интерфейсы

Interfaces				Add Bond	Add Team	Add Bridge	Add VLAN
Name	IP Address	Sending	Receiving				
ens3	172.25.250.10/24	8.48 Kbps	4.75 Kbps				

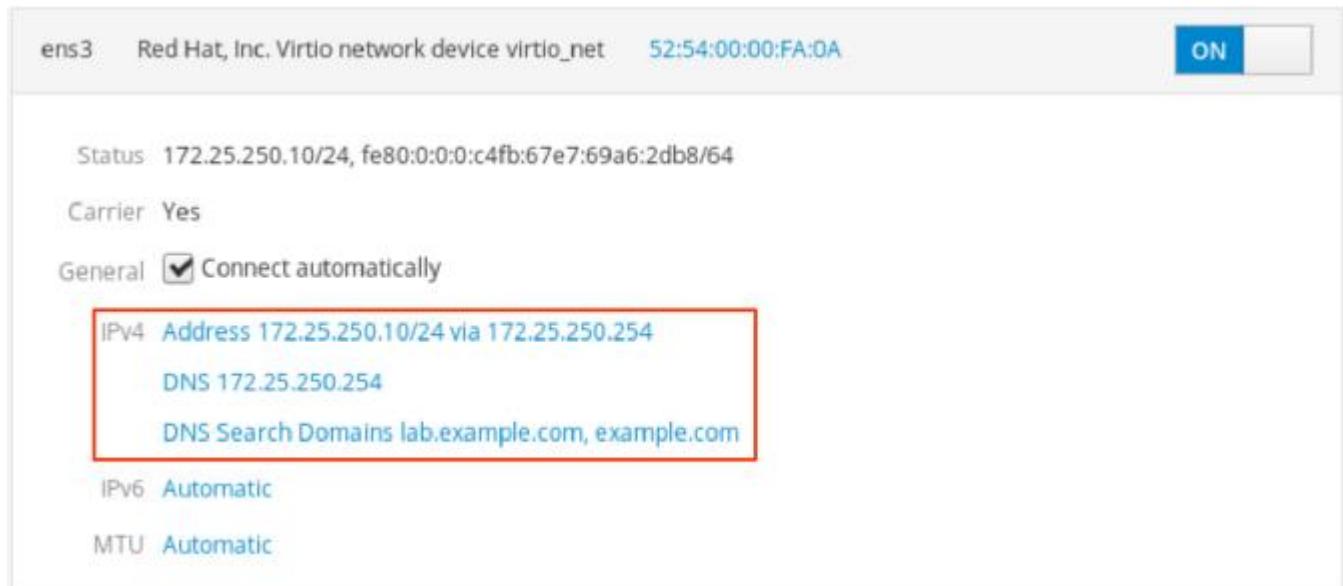
В верхней части страницы управления отображается активность сетевого трафика для выбранного устройства. Прокрутите вниз, чтобы просмотреть параметры конфигурации и параметры управления.

Рисунок 16.22: Сеть: детали интерфейса



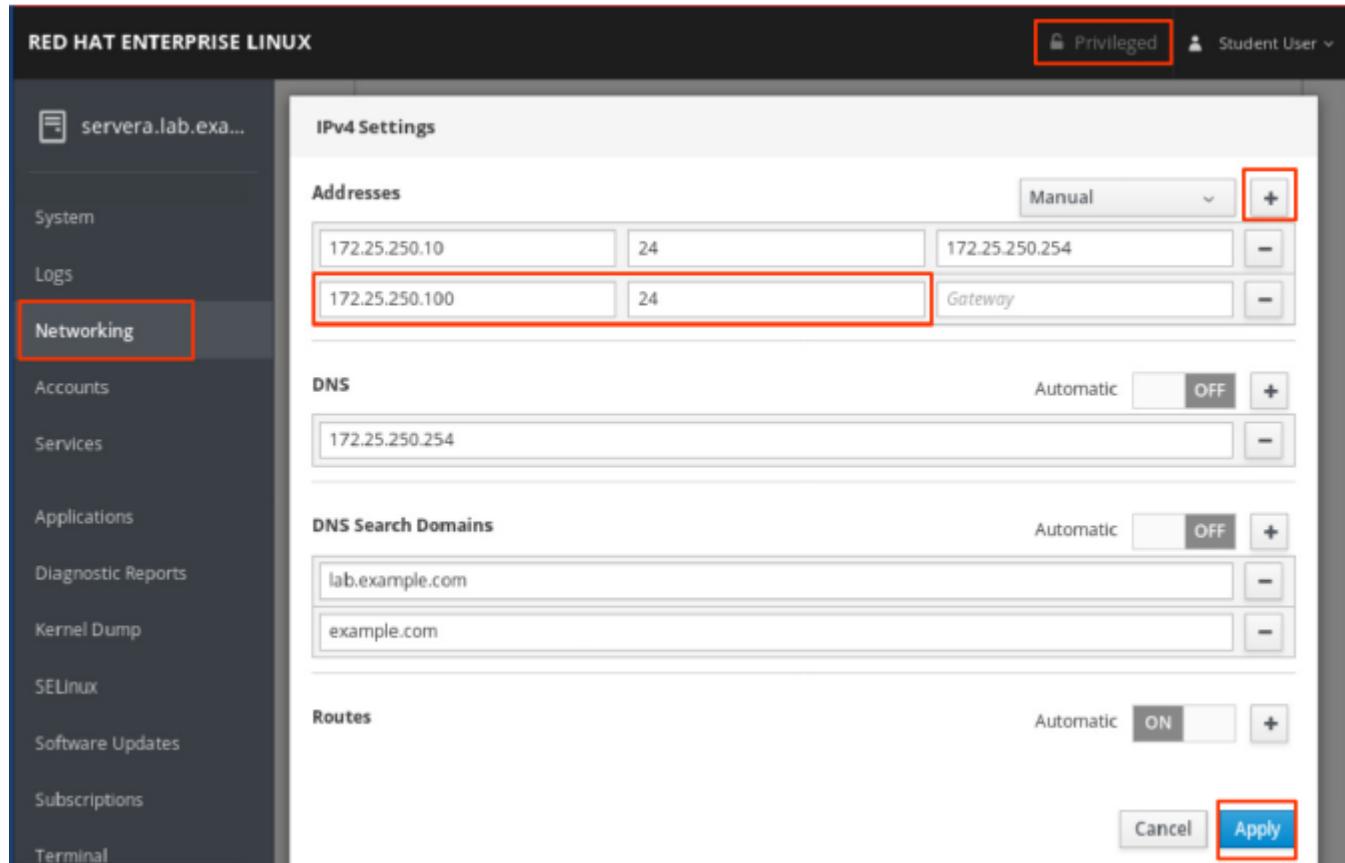
Чтобы изменить или добавить параметры конфигурации в интерфейс, щелкните выделенные ссылки для нужной конфигурации. В этом примере ссылка **IPv4** показывает один IP-адрес и сетевую маску, **172.25.250.10/24** для сетевого интерфейса **ens3**. Чтобы добавить дополнительный IP-адрес к сетевому интерфейсу ens3, щелкните выделенную ссылку.

Рисунок 16.23: Сеть: раздел конфигурации ens3



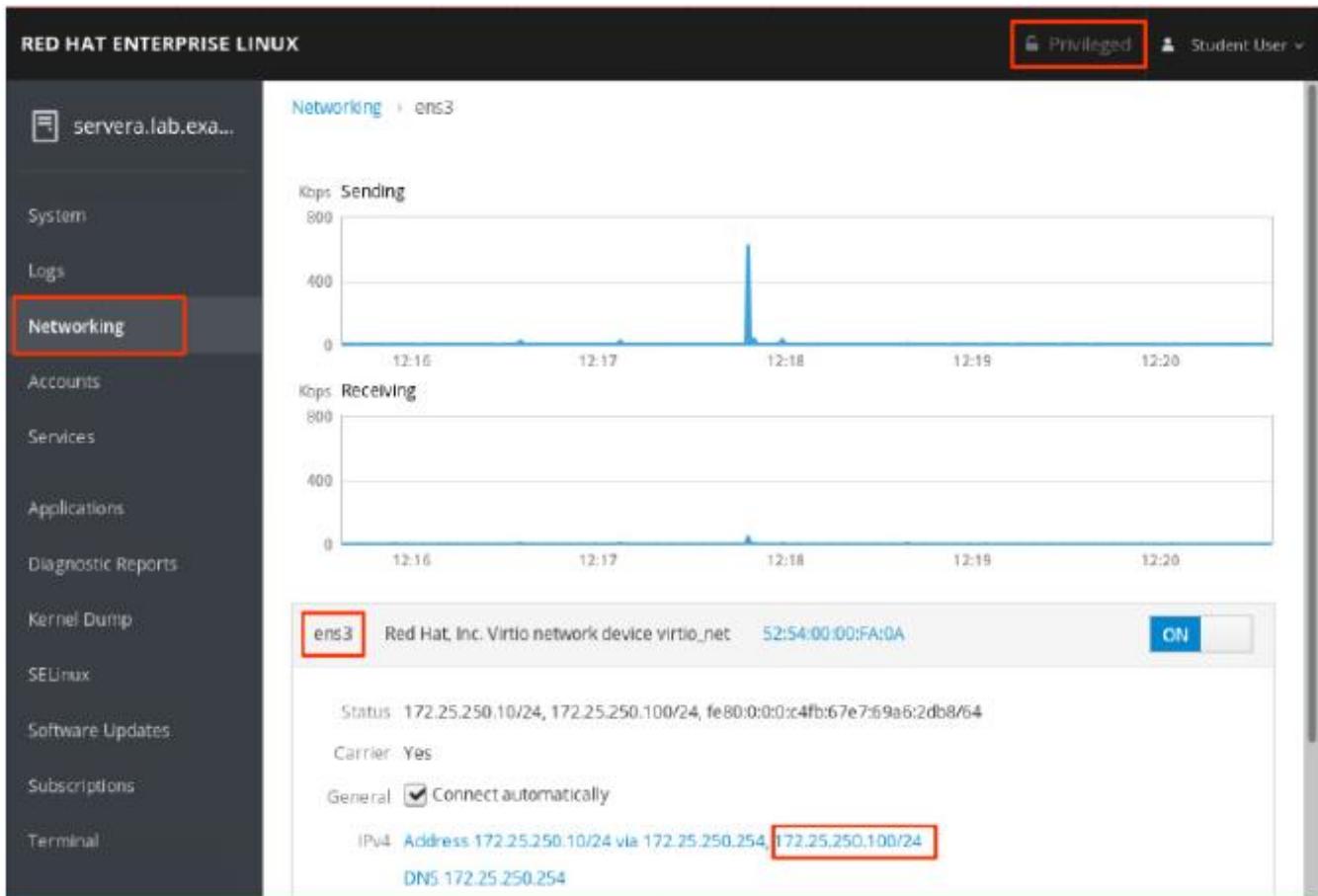
Щелкните + в правой части раскрывающегося списка «Вручную (Manual)», чтобы добавить дополнительный IP-адрес. Введите IP-адрес и маску сети в соответствующие поля. Нажмите «Применить (Apply)», чтобы активировать новые настройки.

Рисунок 16.24: Добавление IP-адреса к существующему интерфейсу



Дисплей автоматически переключается обратно на страницу управления интерфейсом, где вы можете подтвердить новый IP-адрес.

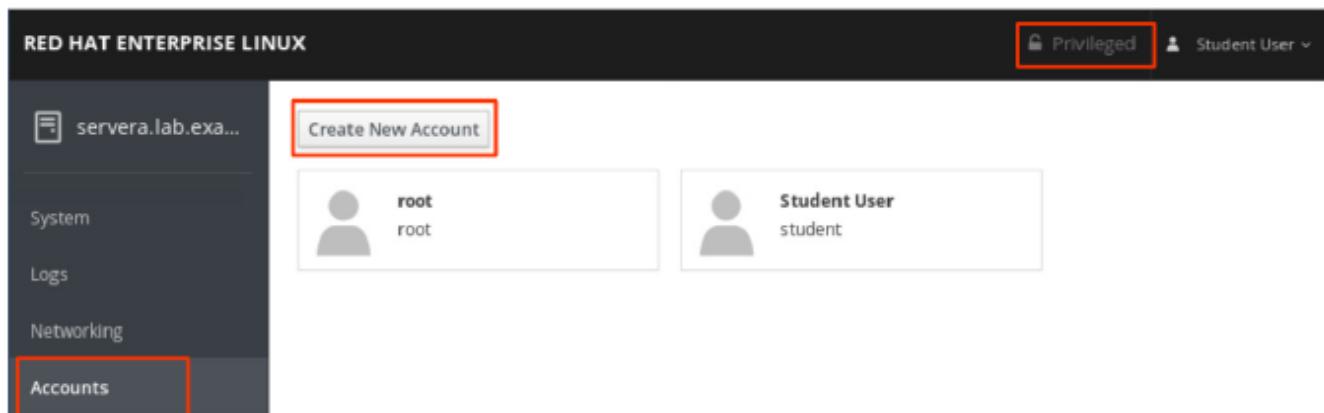
Рисунок 16.25: Подтвердите новый IP-адрес



Администрирование учетных записей пользователей

Как привилегированный пользователь вы можете создавать новые учетные записи пользователей в веб-консоли. Щелкните «Учетные записи (Accounts)» на левой панели навигации, чтобы просмотреть существующие учетные записи. Нажмите «Создать новую учетную запись (Create New Account)», чтобы открыть страницу управления учетной записью.

Рисунок 16.26: Существующие учетные записи пользователей



Введите информацию для новой учетной записи, затем нажмите «Создать», чтобы завершить действие.

Рисунок 16.27: Интерфейс создания новой учетной записи

The screenshot shows a 'Create New Account' dialog box. It contains the following fields:

- Full Name: New User
- User Name: nuser
- Password: [REDACTED]
- Confirm: [REDACTED]
- Access: Lock Account

At the bottom right are two buttons: 'Cancel' and a blue 'Create' button.

Дисплей автоматически вернется на страницу управления учетной записью, где вы можете подтвердить новую учетную запись пользователя.

Рисунок 16.28: Страница управления учетной записью

The screenshot shows the Red Hat Enterprise Linux Cockpit web interface. The left sidebar has a red box around the 'Accounts' item. The main area shows a list of users:

- New User (highlighted with a red box)
- root
- Student User

At the top right, there are 'Privileged' and 'Student User' status indicators.



РЕКОМЕНДАЦИИ

Справочные страницы **man cockpit(1)**, **cockpit-ws(8)**, и **cockpit.conf(5)**

Дополнительные сведения см. В разделе «Управление системами с помощью веб-интерфейса Cockpit (*Managing systems using the Cockpit web interface*)» в руководстве по использованию Cockpit для управления системами в Red Hat Enterprise Linux 8.0 Beta по адресу:

https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/8-beta/html/managing_systems_using_the_cockpit_web_interface/

УПРАЖНЕНИЯ С ПОШАГОВЫМИ ИНСТРУКЦИЯМИ

АНАЛИЗ И УПРАВЛЕНИЕ УДАЛЕННЫМИ СЕРВЕРАМИ

В этом упражнении вы включите веб-консоль на сервере и получите доступ к ней для управления ему, а также для диагностики и решения проблем.

В РЕЗУЛЬТАТЕ

Вы должны иметь возможность использовать веб-консоль для мониторинга основных функций системы, проверки файлов журналов, создания учетных записей пользователей и доступа к терминалу.

ПРЕЖДЕ ЧЕМ ВЫ НАЧНЕТЕ

Войдите на рабочую станцию **workstation** как пользователь **student**, используя в качестве пароля слово **student**.

С рабочей станции **workstation** выполните скрипт **lab support-cockpit start**. Команда запускает сценарий, чтобы определить, доступны ли в сети хосты **servera** и **serverb**.

```
[student@workstation ~]$ lab support-cockpit start
```

1. Используйте команду **ssh** для входа на сервер **servera** как пользователь **student**. Системы настроены на использование ключей SSH для аутентификации, поэтому пароль для входа на сервер не требуется.

```
[student@workstation ~]$ ssh student@servera
Activate the web console with: systemctl enable --now cockpit.socket
[student@servera ~]$
```

2. Веб-консоль уже установлена в системе, но не активирована. Включите и запустите службу кабинки.

- 2.1. Используйте команду **systemctl enable --now cockpit.socket**, чтобы включить службу веб-консоли.

```
[student@servera ~]$ sudo systemctl enable --now cockpit.socket
[sudo] password for student: student
Created symlink /etc/systemd/system/sockets.target.wants/cockpit.socket -> /usr/
lib/systemd/system/cockpit.socket.
```

- На рабочей станции **workstation**, откройте Firefox и войдите в интерфейс веб-консоли, работающий в системе **servera.lab.example.com**, как пользователь **student** с паролем **student**.

3.1. Откройте Firefox и перейдите по адресу <https://servera.lab.example.com:9090>.

3.2. Примите самозаверяющий сертификат, добавив его в качестве исключения.

3.3. Войдите в систему как пользователь **student** с паролем **student**.

Теперь вы вошли в систему как обычный пользователь с минимальными привилегиями.

- Проверьте текущую авторизацию в интерфейсе веб-консоли.

4.1. Щелкните Терминал (**Terminal**) на левой панели навигации, чтобы получить доступ к терминалу.

Открывается сеанс терминала, в котором пользователь **student** уже вошел в систему. Используйте команду **id**, чтобы убедиться, что выполнение команды работает во встроенным терминале.

```
[student@servera ~]$ id  
uid=1000(student) gid=1000(student) groups=1000(student),10(wheel)  
context=unconfined_u:unconfined_r:unconfined_t:s0
```

4.2. Щелкните «Учетные записи (**Accounts**)» на левой панели навигации, чтобы управлять пользователями.

Наведите указатель мыши на кнопку «Создать новую учетную запись», расположенную в верхнем левом углу. Обратите внимание, что пользователю-студенту не разрешено создавать новые учетные записи.

4.3. Щелкните ссылку "Пользователь-студент (**Student User**)".

На странице сведений об учетной записи пользователя ученика обратите внимание, что пользователю разрешено только установить новый пароль или добавить авторизованные открытые ключи SSH.

4.4. Щелкните раскрывающееся меню «Пользователь-студент (**Student User**)» в правом верхнем углу. В раскрывающемся меню выберите «Выйти (**Log Out**)».

- Войдите в веб-консоль с правами администратора.

5.1. Снова войдите в интерфейс веб-консоли как пользователь **student** с паролем **student**, но на этот раз отметьте флажок «Повторно использовать мой пароль для привилегированных задач (**Reuse my password for privileged tasks**)».

5.2. Чтобы проверить административный доступ, убедитесь, что метка «Привилегированный (**Privileged**)» отображается рядом с именем учетной записи «пользователя-ученика (**Student User**)» в верхней правой части интерфейса веб-консоли.

- Чтобы изучить статистику системы, щелкните Система на левой панели навигации.

На этой странице отображается различная базовая статистика операционной системы, например: текущая нагрузка, использование диска, дисковый ввод-вывод и сетевой трафик.

7. Чтобы просмотреть системные журналы, щелкните «Журналы (Logs)» на левой панели навигации.

На этой странице отображаются системные журналы **systemd**. Используйте кнопки, расположенные в верхнем левом углу, чтобы изменить способ отображения записей журнала в зависимости от даты и типа **приоритета (severity)** журналов.

- 7.1. Щелкните раскрывающийся список «Приоритета (Severity)» и выберите «Все (Everything)».

- 7.2. В зависимости от текущего дня месяца щелкните любую запись журнала из списка.

Откроется страница сведений о записи журнала с дополнительной информацией о событии, такой как имя хоста, контекст **SELinux** или номер **PID** процесса, которому соответствует запись.

8. Добавьте второй IP-адрес к существующему устройству сетевого интерфейса.

- 8.1. Щелкните «Сеть (Networking)» на левой панели навигации.

На этой странице отображаются сведения о текущей конфигурации сети для сервера, а также сетевая статистика в реальном времени, конфигурация брандмауэра и записи журнала, связанные с сетью.

- 8.2. Прокрутите вниз до раздела «Интерфейсы (Interfaces)» и щелкните строку с именем сетевого интерфейса **ens3**.

На странице сведений в реальном времени отображается сетевая статистика, а также текущая конфигурация этого сетевого интерфейса.

- 8.3. Щелкните ссылку **Address 172.25.250.10/24 via 172.25.250.254**.

Откроется окно настроек **IPv4**, в котором вы можете изменить конфигурацию сетевого интерфейса.

- 8.4. В окне «Настройки IPv4 (IPv4 Settings)» нажмите «+» рядом с «Вручную (Manual)».

- 8.5. В текстовом поле «Адрес (Address)» введите 172.25.250.99 в качестве второго IP-адреса.

- 8.6. В текстовом поле «префикс (Prefix length)» или «Сетевая маска (Netmask)» введите значения сетевой маски 24.

- 8.7. Нажмите «Применить (Apply)», чтобы сохранить новую конфигурацию сети.

Обратите внимание, что новая конфигурация применяется немедленно. Новый IP-адрес отображается в строке IPv4.

9. Создайте новую учетную запись пользователя.

- 9.1.** Щелкните «Учетные записи (Accounts)» на левой панели навигации.
- 9.2.** Щелкните «Создать новую учетную запись (Create New Account)».
- 9.3.** В окне «Создать новую учетную запись (Create New Account)» добавьте следующие данные:

ПОЛЕ	ЗЕНИТЬ
Полное имя (Full Name)	manager1
Имя пользователя (User Name)	manager1
Пароль (Password)	redh@t123
Подтвердить (Confirm)	redh@t123

- 9.4.** Щелкните «Создать (Create)».
- 10.** Войдите в сеанс терминала (**terminal**) в веб-консоли, чтобы добавить пользователя **manager1** в группу **wheel**.
- 10.1.** Щелкните «Терминал (Terminal)» на левой панели навигации.
- 10.2.** Используйте команду **id manager1** для просмотра членства в группе пользователя **manager1**.

```
[student@servera ~]$ sudo id manager1
uid=1001(manager1) gid=1001(manager1) groups=1001(manager1)
```

- 10.3.** Используйте команду **sudo usermod -aG wheel manager1**, чтобы добавить **manager1** в группу **wheel**.

```
[student@servera ~]$ sudo usermod -aG wheel manager1
[sudo] password for student: student
[student@servera ~]$
```

- 10.4.** Используйте команду **id manager1**, чтобы убедиться, что **manager1** является членом группы **wheel**.

```
[student@servera ~]$ id manager1
uid=1001(manager1) gid=1001(manager1) groups=1001(manager1),10(wheel)
```

- 11.** Включите и запустите службу «учета процессов ядра (Kernel process accounting)» (**psacct**).
- 11.1.** Щелкните «Службы (Services)» на левой панели навигации.

- 11.2.** Прокрутите список «системных служб (**System Services**)» до раздела «Отключено (**Disabled**)». Найдите и щелкните ссылку «Учет процесса ядра (**Kernel process accounting**)».
- 11.3.** Щелкните «Включить (**Enable**)».
- 11.4.** Щелкните "Пуск (**Start**)".

Теперь служба включена и запущена.

12. Выдите из интерфейса веб-консоли.

13. Выдите из сервера.

```
[student@servera ~]$ exit  
[student@workstation ~]$
```

Завершение

На рабочей станции запустите **lab support-cockpit finish**, чтобы завершить это упражнение.

```
[student@workstation ~]$ lab support-cockpit finish
```

На этом пошаговое упражнение завершено.

ПОЛУЧЕНИЕ ПОМОЩИ НА ПОРТАЛЕ КЛИЕНТОВ RED HAT

ЗАДАЧИ

По завершении этого раздела вы сможете описать ключевые ресурсы, доступные на портале клиентов Red Hat, и использовать их для поиска информации из документации Red Hat и базы знаний.

ДОСТУП К РЕСУРСАМ ПОДДЕРЖКИ НА ПОРТАЛЕ КЛИЕНТОВ RED HAT

Портал для клиентов Red Hat (<https://access.redhat.com>) предоставляет клиентам доступ к документации, загружаемым материалам, инструментам и техническим знаниям. Клиенты могут искать решения, ответы на часто задаваемые вопросы и статьи в базе знаний. На портале для клиентов вы можете:

- Доступ к официальной документации по продукту.
- Отправляйте заявки в службу поддержки и управляйте ими.
- Управление подписками на программное обеспечение и правами.
- Получать загружаемое программное обеспечение, обновления и оценки.
- Проконсультируйтесь с инструментами, которые помогут вам оптимизировать конфигурацию ваших систем.

Некоторые части сайта доступны всем, а другие доступны только клиентам с активными подписками. Получите помощь в доступе к клиентскому порталу по адресу <https://access.redhat.com/help/>.

ОРИЕНТИРОВАНИЕ НА ПОРТАЛ КЛИЕНТОВ

Клиенты могут получить доступ к клиентскому порталу Red Hat через веб-браузер. В этом разделе представлен обзор клиентского портала. Тур можно найти по адресу <https://access.redhat.com/start>.

Тур - очень полезный инструмент для знакомства со всем, что может предложить портал, и для того, чтобы получить максимальную отдачу от подписки Red Hat. После входа на клиентский портал Red Hat нажмите «Обзор клиентского портала».

ОРИЕНТИРОВАНИЕ НА ПОРТАЛ КЛИЕНТОВ

Клиенты могут получить доступ к клиентскому порталу Red Hat через веб-браузер. В этом разделе представлен обзор клиентского портала. Тур можно найти по адресу <https://access.redhat.com/start>.

Тур - очень полезный инструмент для знакомства со всем, что может предложить портал, и для того, чтобы получить максимальную отдачу от подписки Red Hat. После входа на клиентский портал Red Hat нажмите «Обзор клиентского портала (**Customer Portal**)».

Рисунок 16.29: Обзор клиентского портала



Tour the Customer Portal

ДОБРО ПОЖАЛОВАТЬ НА ПОРТАЛ КЛИЕНТОВ RED HAT! (**WELCOME TO THE RED HAT CUSTOMER PORTAL!**) окно открывается с двумя вариантами: ЗАКРЫТЬ (**CLOSE**) и ДАЛЕЕ (**NEXT**). Нажатие кнопки «ДАЛЕЕ (**NEXT**)» запустит обзор интерфейса клиентского портала. Это первое из последовательности окон, в которых выделяются различные части интерфейса.

Верхняя панель навигации

Первые три остановки в обзоре клиентского портала можно найти на верхней панели навигации веб-сайта клиентского портала Red Hat:

Рисунок 16.30: Верхняя панель навигации

SUBSCRIPTIONS DOWNLOADS CONTAINERS SUPPORT CASES

Подписки (**Subscriptions**) открывают новую страницу, на которой вы можете управлять своими зарегистрированными системами, а также своими подписками и использованием прав. В нем приводится информация о применяемых ошибках и позволяет создавать ключи активации (*activation keys*), которые можно использовать при регистрации систем, чтобы гарантировать, что они получают права по правильным подпискам. Обратите внимание, что, если вы являетесь частью организации, администратор вашей организации может ограничить ваш доступ к этой странице.

Загрузки (**Downloads**) открывает новую страницу, которая дает вам доступ к загрузкам вашего продукта и для запроса разрешений на оценку для продуктов, на которые у вас нет прав.

Обращения в службу поддержки (**Support Cases**) открывает новую страницу, которая обеспечивает доступ для создания, отслеживания и управления вашими обращениями в службу поддержки через систему управления обращениями при условии, что ваша организация санкционировала этот уровень доступа.

Ваше имя - это заголовок меню пользователя (**User Menu**), которое позволяет вам управлять своей учетной записью, учетными записями, для которых вы являетесь администратором организации, своим личным профилем и опциями для уведомлений по электронной почте о новом доступном содержимом.

Значок глобуса открывает меню «Выбор языка (**Select Your Language**)», в котором можно указать языковые предпочтения для клиентского портала.

Темы Меню

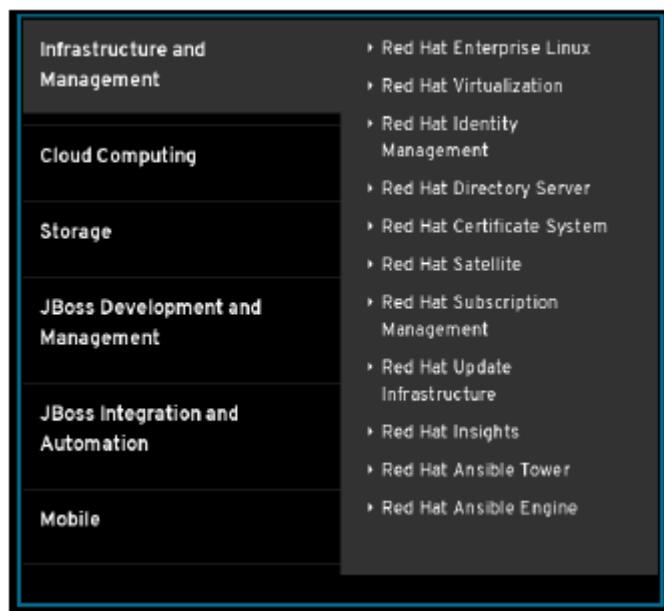
Под верхней панелью навигации на главной странице клиентского портала находится меню, которое можно использовать для перехода к четырем основным категориям ресурсов, доступных на сайте.

Рисунок 16.31: Меню ресурсов



Продукты и услуги (**Products & Services**) предоставляют доступ к центрам продуктов (**Product Hubs**), страницам, которые предоставляют доступ к оценкам конкретных продуктов, обзорам, руководствам по началу работы и другой информации о поддержке продуктов. Вы также можете получить доступ к документации по продуктам Red Hat, прямым ссылкам на Базу знаний статей поддержки, а также к информации о политиках поддержки и о том, как связаться со службой поддержки Red Hat.

Рисунок 16.32: Продукты и услуги



Инструменты (**Tools**) предоставляют вам ссылки на инструменты, которые помогут вам добиться успеха с продуктами Red Hat.

Механизм решений (**Solution Engine**) предоставляет вам эффективный способ быстрого поиска решений ваших проблем по продукту и открытия заявки в службу поддержки, если вы не нашли удовлетворительного решения.

Лаборатории клиентского портала (**Labs**) предоставляют набор веб-приложений и инструментов, которые помогут вам повысить производительность, диагностировать проблемы, выявить проблемы безопасности и оптимизировать ваши конфигурации. Например, Средство проверки жизненного цикла продукта (**Product Life Cycle Checker**) позволяет выбрать конкретный продукт и просмотреть график его жизненного цикла поддержки. Другой инструмент, **Rescue Mode Assistant**, помогает сбросить пароль **root** системы, создавать диагностические отчеты или устранять проблемы с файловыми системами во время загрузки. Но на этом сайте доступно множество других инструментов.

Рисунок 16.33: Инструменты



Безопасность (**Security**) обеспечивает доступ к Центру безопасности продуктов (**Red Hat Product Security Center**) Red Hat по адресу <https://access.redhat.com/security/>. В этом разделе также содержится информация о серьезных проблемах безопасности, доступе к базе данных Red Hat CVE, каналу безопасности в блоге Red Hat, а также ресурсы о процессе реагирования Red Hat, а также о том, как мы оцениваем проблемы и решаем их.

Наконец, меню «Сообщество (**Community**)» - это место, где эксперты, клиенты и партнеры Red Hat могут общаться и сотрудничать. Здесь можно найти дискуссионные форумы, блоги и информацию о предстоящих событиях в вашем регионе.



ПРИМЕЧАНИЕ

Чтобы получить полный рассказ о портале для клиентов. Для доступа к этой странице вам потребуется хотя бы одна активная подписка в вашей учетной записи на портале клиентов.

ПОИСК БАЗЫ ЗНАНИЙ С ИНСТРУМЕНТОМ ПОДДЕРЖКИ RED HAT

Утилита Red Hat Support Tool, **redhat-support-tool**, предоставляет текстовый интерфейс, который позволяет вам искать статьи базы знаний и файлы поддержки на портале для клиентов из командной строки вашей системы. У этого инструмента нет графического интерфейса, и, поскольку он взаимодействует с клиентским порталом Red Hat, ему требуется доступ в Интернет. Запустите команду **redhat-support-tool**, используя любой терминал или соединение **SSH**.

Команду **redhat-support-tool** можно использовать в интерактивном режиме или вызывать как команду с параметрами и аргументами. Синтаксис инструмента одинаков для обоих методов. По умолчанию программа запускается в интерактивном режиме. Используйте подкоманду **help**, чтобы увидеть все доступные команды. Интерактивный режим поддерживает завершение табуляции и возможность вызова программ в родительской оболочке.

```
[user@host ~]$ redhat-support-tool
Welcome to the Red Hat Support Tool.
Command (? for help):
```

При первом запуске **redhat-support-tool** запрашивает информацию для входа в систему подписчика Red Hat Customer Portal. Чтобы избежать повторного предоставления этой информации, инструмент просит сохранить информацию об учетной записи в домашнем каталоге пользователя (**~/.redhat-support-tool/redhat-supporttool.conf**). Если все проблемы регистрируются через конкретную учетную запись Red Hat Customer Portal, опция **--global** может сохранить информацию учетной записи в **/etc/redhat-support-tool.conf** вместе с другой

общесистемной конфигурацией. Команда **config** инструмента изменяет настройки конфигурации инструмента.

Команда **redhat-support-tool** позволяет подписчикам искать и отображать содержимое базы знаний на портале клиентов Red Hat. База знаний разрешает поиск, по ключевым словам, аналогично команде **man**. Пользователи могут вводить коды ошибок, синтаксис из файлов журнала или любое сочетание ключевых слов для создания списка соответствующих документов решения.

Ниже представлена начальная конфигурация и демонстрация базового поиска:

```
[user@host ~]$ redhat-support-tool
Welcome to the Red Hat Support Tool.
Command (? for help): search How to manage system entitlements with subscription-manager
Please enter your RHN user ID: subscriber
Save the user ID in /home/student/.redhat-support-tool/redhat-support-tool.conf
(y/n): y
Please enter the password for subscriber: password
Save the password for subscriber in /home/student/.redhat-support-tool/redhatsupport-
tool.conf (y/n): y
```

После запроса у пользователя требуемой конфигурации пользователя инструмент продолжает исходный поисковый запрос:

```
Type the number of the solution to view or 'e' to return to the previous menu.
1 [ 253273:VER] How to register and subscribe a system to the Red Hat Customer
    Portal using Red Hat Subscription-Manager
2 [ 265523:VER] Enabling or disabling a repository using Red Hat Subscription
    Management
3 [ 100423:VER] Why does subscription-manager list return: "No Installed
    Products found" ?
...output omitted...
Select a Solution: 1
```

Выберите статью **номер 1**, как указано выше, и вам будет предложено выбрать раздел документа для чтения. Наконец, используйте клавишу **Q**, чтобы выйти из раздела, в котором вы находитесь, или используйте ее несколько раз, чтобы выйти из команды **redhat-support-tool**.

Select a Solution: 1

Type the number of the section to view or 'e' to return to the previous menu.

```
1 Title
2 Issue
3 Environment
4 Resolution
5 Display all sections
End of options.
Section: 1

Title
=====
How to register and subscribe a system to the Red Hat Customer Portal using Red
Hat Subscription-Manager
URL:      https://access.redhat.com/solutions/253273
Created On:  None
Modified On: 2017-11-29T15:33:51Z

(END) q
Section:
Section: q

Select a Solution: q

Command (? for help): q
[user@hosts ~]#
```

Доступ к статьям базы знаний по идентификатору документа

Находите онлайн-статьи напрямую, используя команду **kb** инструмента с идентификатором документа базы знаний. Возвращенный документ прокручивается на экране без разбивки на страницы, но вы можете перенаправить его в файл, чтобы сохранить его, и использовать команду **less**, чтобы прокручивать его по экрану за раз.

```
[user@host ~]$ redhat-support-tool kb 253273

Title
=====
How to register and subscribe a system to the Red Hat Customer Portal using Red
Hat Subscription-Manager
URL:      https://access.redhat.com/solutions/253273
Created On:  None
Modified On: 2017-11-29T15:33:51Z

Issue
=====
*  How to register a new `Red Hat Enterprise Linux` system to the Customer Portal
   using `Red Hat Subscription-Manager`
...output omitted...
```

УПРАВЛЕНИЕ СЛУЧАЯМИ ПОДДЕРЖКИ С ПОМОЩЬЮ RED HAT SUPPORT TOOL

Одним из преимуществ подписки на продукт является доступ к технической поддержке через портал клиентов Red Hat. В зависимости от уровня поддержки подписки системы, с Red Hat можно связаться через онлайн-инструменты или по телефону. См. https://access.redhat.com/site/support/policy/support_process для получения подробной информации.

Подготовка отчета об ошибке

Прежде чем обращаться в службу поддержки Red Hat, важно собрать соответствующую информацию для отчета об ошибке.

Определите проблему. Уметь четко обозначить проблему и ее симптомы. Будьте как можно более конкретными. Подробно опишите шаги, которые позволяют воспроизвести проблему.

Сберите справочную информацию. Какой продукт и версия затронуты? Будьте готовы предоставить актуальную диагностическую информацию. Это может включать вывод **sosreport**, который обсуждается далее в этом разделе. В случае проблем с ядром это может включать аварийный дамп системы **kdump** или цифровую фотографию трассировки ядра, отображаемую на мониторе аварийной системы.

Определите уровень серьезности. Red Hat использует четыре уровня серьезности для классификации проблем. После отчетов о срочных и серьезных проблемах следует позвонить в соответствующий местный центр поддержки (см. <https://access.redhat.com/site/support/contact/technicalSupport>).

СТРОГОСТЬ	ОПИСАНИЕ
Срочно (<i>Urgent</i>) (уровень серьезности 1)	Проблема, котораяserьезно влияет на использование программного обеспечения в производственной среде. Это включает потерю производственных данных или неисправные производственные системы. Ситуация останавливает ваши бизнес-операции, и нет никаких процедурных решений.
Высокая (<i>High</i>) (уровень серьезности 2)	Проблема, при которой программное обеспечение функционирует, но его использование в производственной среде значительно сокращается. Ситуация сильно влияет на ваши бизнес-операции, и процедурного обходного пути не существует.
Средний (<i>Medium</i>) (уровень серьезности 3)	Проблема, которая включает частичную некритическую потерю возможности использования программного обеспечения в производственной среде или среде разработки. В производственной среде влияние на ваш бизнес от среднего до низкого. Бизнес продолжает функционировать, используя процедурный обходной путь. Для сред разработки такая ситуация вызывает проблемы при переносе проекта в производственную среду.

Низкий (Low) (уровень серьезности 4)	Общий вопрос об использовании, сообщение об ошибке в документации или рекомендация по усовершенствованию или модификации продукта в будущем. Для производственных сред это практически не влияет на ваш бизнес, производительность или функциональность вашей системы. В случае сред разработки влияние на ваш бизнес от среднего до низкого, но ваш бизнес продолжает функционировать, используя процедурный обходной путь.
---	--

Управление отчетом об ошибке с помощью redhat-support-tool

Подписчики могут создавать, просматривать, изменять и закрывать обращения в службу поддержки Red Hat с помощью **redhat-support-tool**. Когда заявки в службу поддержки находятся в открытом (**opened**) или поддерживаемом (**maintained**) состояниях, пользователи могут прикреплять файлы или документацию, например, диагностические отчеты (**sosreport**). Инструмент загружает и прикрепляет файлы к делам.

Подробности обращения, включая продукт, версию, сводку, описание, серьезность (*product, version, summary, description, severity*) и группу случаев (*case group*), могут быть назначены с помощью параметров команды или позволяя инструменту запрашивать необходимую информацию. В следующем примере открывается новое дело. Указаны параметры **--product** и **--version**.

```
[user@host ~]$ redhat-support-tool
Welcome to the Red Hat Support Tool.
Command (? for help): opencase --product="Red Hat Enterprise Linux" --
version="7.0"
Please enter a summary (or 'q' to exit): System fails to run without power
Please enter a description (Ctrl-D on an empty line when complete):
When the server is unplugged, the operating system fails to continue.

1  Urgent
2  High
3  Normal
4  Low
Please select a severity (or 'q' to exit): 4
Would you like to assign a case group to this case (y/N)? N
Would see if there is a solution to this problem before opening a support case?
(y/N) N
-----
Support case 01034421 has successfully been opened.
```

Если параметры **--product** и **--version** не указаны, **redhat-support-tool** предоставляет список вариантов для этих параметров.

```
[user@host ~]$ redhat-support-tool
Welcome to the Red Hat Support Tool.
```

```
Command (? for help): opencase
```

```
Do you want to use the default product - "Red Hat Enterprise Linux" (y/N)?: y  
...output omitted...
```

```
29 7.4
```

```
30 7.5
```

```
31 7.6
```

```
32 8.0 Beta
```

```
Please select a version (or 'q' to exit): 32
```

```
Please enter a summary (or 'q' to exit): yum fails to install apache
```

```
Please enter a description (Ctrl-D on an empty line when complete):
```

```
yum cannot find correct repo
```

```
1 Urgent
```

```
2 High
```

```
3 Normal
```

```
4 Low
```

```
Please select a severity (or 'q' to exit): 4
```

```
Would you like to use the default (Ungrouped Case) Case Group (y/N)? : y
```

```
Would you like to see if there's a solution to this problem before opening a  
support case? (y/N) N
```

```
-----  
Support case 010355678 has successfully been opened.
```

Прикрепление диагностической информации к запросу в службу поддержки

Включение диагностической информации может привести к более быстрому решению. При открытии кейса прикрепите **sosreport**. Команда **sosreport** создает сжатый tar-архив диагностической информации, собранной из работающей системы. **Redhat-support-tool** предлагает включить его, если архив был создан ранее:

```
Please attach a SoS report to support case 01034421. Create a SoS report as  
the root user and execute the following command to attach the SoS report  
directly to the case:
```

```
redhat-support-tool addattachment -c 01034421 path to sosreport
```

```
Would you like to attach a file to 01034421 at this time? (y/N) N
```

```
Command (? for help):
```

Если текущий отчет **SoS** не существует, администратор может сгенерировать и прикрепить его позже. Используйте команду **redhat-support-tool addattachment**, чтобы прикрепить отчет.

Подписчик также может просматривать, изменять и закрывать обращения в службу поддержки:

```
Command (? for help): listcases
```

```
Type the number of the case to view or 'e' to return to the previous menu.
```

```
1 [Waiting on Red Hat] System fails to run without power
```

```
No more cases to display
```

```
Select a Case: 1
```

```
Type the number of the section to view or 'e' to return to the previous menu.
```

```
1 Case Details  
2 Modify Case  
3 Description  
4 Recommendations  
5 Get Attachment  
6 Add Attachment  
7 Add Comment  
End of options.
```

```
Option: q
```

```
Select a Case: q
```

```
Command (? for help):q
```

```
[user@host ~]$ redhat-support-tool modifycase --status=Closed 01034421  
Successfully updated case 01034421  
[user@host ~]$
```

Red Hat Support Tool имеет расширенные возможности диагностики и анализа приложений. Используя файлы ядра аварийного дампа ядра, **redhat-support-tool** может создавать и извлекать обратную трассировку. Основной файл аварийного дампа ядра создается с помощью команды **kdump**. Обратная трассировка — это отчет об активных кадрах стека на момент аварийного дампа и обеспечивает диагностику на месте. Один из вариантов инструмента **redhat-support-tool** — открыть обращение в службу поддержки.

Инструмент также обеспечивает анализ файла журнала. С помощью команды **analyze** файлы журналов многих типов, включая операционную систему, JBoss, Python, Tomcat и oVirt, могут быть проанализированы для выявления симптомов проблемы. Файлы журнала можно просматривать и диагностировать индивидуально. Предоставление предварительно обработанного анализа, в отличие от необработанных данных, таких как аварийный дамп или файлы журнала, позволяет быстрее открывать обращения в службу поддержки и предоставлять их инженерам.

ПРИСОЕДИНЯЕМСЯ К РАЗРАБОТЧИКАМ RED HAT

Еще один полезный ресурс, доступный в Red Hat, - это Red Hat Developer. Эта программа, размещенная на <https://developer.redhat.com>, предоставляет права подписки на программное обеспечение Red Hat для целей разработки, документации и дополнительных книг от наших экспертов по микросервисам, бессерверным вычислениям, Kubernetes и Linux. Также доступны блог, ссылки на информацию о предстоящих мероприятиях и обучении и другие справочные ресурсы, а также ссылки на портал для клиентов Red Hat.

Регистрация бесплатна и может быть завершена на странице <https://developer.redhat.com/register>.



РЕКОМЕНДАЦИИ

Справочные страницы **man sosreport(1)**

Red Hat Access: инструмент поддержки Red Hat

<https://access.redhat.com/site/articles/445443>

Инструмент поддержки Red Hat Первое использование

<https://access.redhat.com/site/videos/534293>

Обращение в службу технической поддержки Red Hat

https://access.redhat.com/site/support/policy/support_process/

Справка - Портал для клиентов Red Hat

<https://access.redhat.com/site/help/.>

УПРАЖНЕНИЯ С ПОШАГОВЫМИ ИНСТРУКЦИЯМИ

ПОЛУЧЕНИЕ ПОМОЩИ НА ПОРТАЛЕ КЛИЕНТОВ RED HAT

В этом упражнении вы создадите диагностический отчет с помощью веб-консоли.

В РЕЗУЛЬТАТЕ

Вы должны иметь возможность сгенерировать диагностический отчет с помощью веб-консоли, который можно будет отправить на портал клиентов Red Hat как часть обращения в службу поддержки.

ПРЕЖДЕ ЧЕМ ВЫ НАЧНЕТЕ

Войдите на рабочую станцию **workstation** как пользователь **student**, используя в качестве пароля слово **student**.

На рабочей станции **workstation** выполните скрипт **lab support-portal start**. Команда запускает сценарий, который определяет, доступен ли хост, сервер **servera**, в сети. Он также запускает и включает веб-консоль на сервере.

```
[student@workstation ~]$ lab support-portal start
```

1. С рабочей станции **workstation** используйте команду **ssh** для входа в сервер **servera**, как пользователь **student**.

```
[student@workstation ~]$ ssh student@servera  
Web console: https://servera.lab.example.com:9090/ or https://172.25.250.10:9090/
```

```
Last login: Mon Mar 4 10:42:20 2019 from 172.25.250.254  
[student@servera ~]$
```

2. Используйте команду **systemctl**, чтобы убедиться, что служба **cockpit** запущена. При появлении запроса введите пароль **student**.

```
[student@servera ~]$ sudo systemctl status cockpit.service  
[sudo] password for student: student  
● cockpit.service - Cockpit Web Service
```

```
Loaded: loaded (/usr/lib/systemd/system/cockpit.service; static; vendor preset: disabled)
Active: active (running) since Mon 2019-03-04 10:52:42 CET; 12min ago
          Docs: man:cockpit-ws(8)
Process: 2151 ExecStartPre=/usr/sbin/remotectl certificate --ensure --user=root --group=cockpit-ws --selinux-type=etc_t (code=exited, sta>
Main PID: 2155 (cockpit-ws)
      Tasks: 3 (limit: 11406)
     Memory: 8.0M
        CGroup: /system.slice/cockpit.service
                  ├─2155 /usr/libexec/cockpit-ws
                  └─3604 /usr/bin/ssh-agent
...output omitted...
```

3. Выйти из сервера **servera**.

```
[student@servera ~]$ exit
[student@workstation ~]$
```

4. На рабочей станции **workstation** откройте **Firefox** и войдите в интерфейс веб-консоли на сервере **servera.lab.example.com** как пользователь **root** с паролем **redhat**.

- 4.1. Откройте **Firefox** и перейдите по адресу <https://servera.lab.example.com:9090>.
 - 4.2. Если будет предложено, примите самозаверяющий сертификат, добавив его в качестве исключения.
 - 4.3. Войдите в систему как пользователь **root** с паролем **redhat**. Теперь вы вошли в систему как привилегированный пользователь, что необходимо для создания диагностического отчета.
 - 4.4. Щелкните «Диагностический отчет (**Diagnostic Report**)» на левой панели навигации. Щелкните "Создать отчет (**Create Report**)". Создание отчета занимает несколько минут.
5. Когда отчет будет готов, нажмите «Загрузить отчет (**Download Report**)». Сохраните файл.
 - 5.1. Нажмите кнопку «Загрузить отчет (**Download report**)», а затем кнопку «Сохранить файл (**Save File**)».
 - 5.2. Нажмите кнопку "Закрыть (**Close**)".

Завершение

На рабочей станции **workstation**, запустите сценарий **lab support-portal finish**, чтобы завершить это упражнение.

```
[student@workstation ~]$ lab support-portal finish
```

На этом пошаговое упражнение завершено.

ОБНАРУЖЕНИЕ И РЕШЕНИЕ ПРОБЛЕМ В RED HAT INSIGHTS

ЗАДАЧИ

После завершения этого раздела вы сможете использовать Red Hat Insights для анализа серверов на наличие проблем, их исправления или разрешения и подтверждения работоспособности решения.

ВВИДЕНИЕ В RED HAT INSIGHTS

Red Hat Insights - это инструмент прогнозной аналитики, который поможет вам выявлять и устранять угрозы безопасности, производительности, доступности и стабильности систем, в которых используются продукты Red Hat в вашей инфраструктуре. Red Hat Insights поставляется в виде продукта «программное обеспечение как услуга (Software-as-a-Service) » (SaaS), поэтому вы можете быстро развернуть и масштабировать его без дополнительных требований к инфраструктуре. Кроме того, это означает, что вы можете немедленно воспользоваться последними рекомендациями и обновлениями Red Hat для ваших развернутых систем.

Red Hat регулярно обновляет базу знаний, используемую Red Hat Insights, на основе общих рисков поддержки, уязвимостей безопасности, заведомо плохих конфигураций и других проблем, выявленных Red Hat. Действия по устранению или устраниению этих проблем подтверждаются и проверяются Red Hat. Это позволяет вам заранее выявлять, расставлять приоритеты и решать проблемы, прежде чем они станут более серьезными.

Для каждой обнаруженной проблемы Red Hat Insights предоставляет оценки представленного риска и рекомендации по смягчению или устраниению проблемы. Эти рекомендации могут содержать материалы, такие как Ansible Playbooks или удобочитаемые пошаговые инструкции, которые помогут вам решить проблему.

Рекомендации Red Hat Insights адаптированы к каждой системе, зарегистрированной в службе. Вы устанавливаете каждую клиентскую систему с агентом, который собирает метаданные о конфигурации времени выполнения системы. Эти данные являются частью того, что вы можете предоставить службе поддержки Red Hat с помощью **sosreport** для разрешения обращения в службу поддержки. Вы можете ограничить или скрыть данные, которые отправляют ваши клиенты. Это блокирует работу некоторых аналитических правил, в зависимости от того, что вы ограничиваете.

Практически сразу после того, как вы зарегистрируете сервер, и он завершит начальную синхронизацию метаданных системы, вы сможете увидеть свой сервер и любые рекомендации по нему в интерфейсе **Red Hat Insights** в своей учетной записи на портале клиентов Red Hat.

Red Hat Insights в настоящее время предоставляет прогнозную аналитику и рекомендации для следующих продуктов Red Hat:

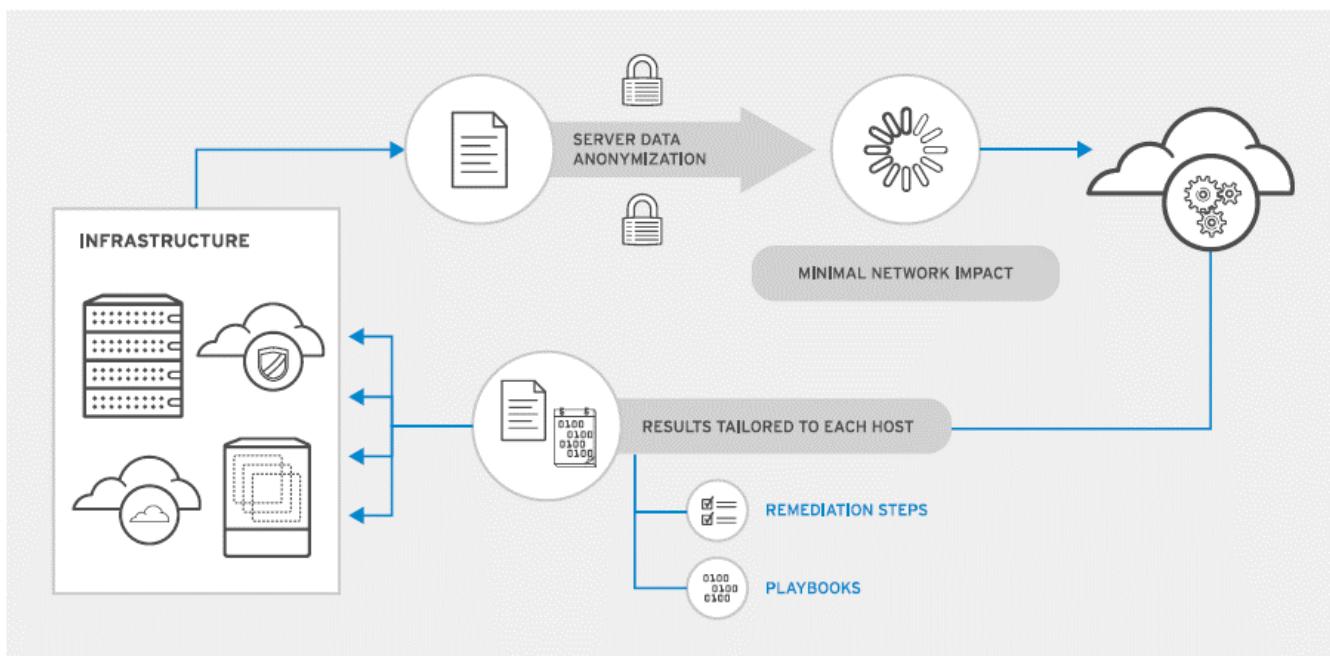
- Red Hat Enterprise Linux 6.4 and later
- Red Hat Virtualization 4 and later
- Red Hat OpenShift Container Platform
- Red Hat Cloud Suite 6 and Red Hat Cloud Infrastructure 6 and later

Описание архитектуры Red Hat Insights

Вы можете зарегистрировать систему в Red Hat Insights через портал для клиентов Red Hat. Когда вы регистрируете систему, она предоставляет Red Hat Insights метаданные о своей текущей конфигурации. Эти данные отправляются в Red Hat Insights с использованием шифрования TLS для защиты при передаче. Он также анонимизируется перед отправкой.

На основе рекомендаций, предоставленных механизмом правил Red Hat Insights, результаты анализа отображаются в интерфейсе Red Hat Insights в вашей учетной записи на портале клиентов Red Hat.

Рисунок 16.34: Архитектура высокого уровня Red Hat Insights



УСТАНОВКА КЛИЕНТОВ RED HAT INSIGHTS

Чтобы настроить Red Hat Insights для серверов Red Hat Enterprise Linux, установите в системе пакет **insights-client**.



ВАЖНО

Пакет **insights-client** заменяет старый пакет **redhat-access-insights**, начиная с Red Hat Enterprise Linux 7.5.

Если ваша система зарегистрирована для получения прав на программное обеспечение через службу управления подпиской на портале клиентов, вы можете активировать Red Hat Insights с помощью одной команды. Используйте команду **insights-client --register** для регистрации системы.

```
[root@demo ~]# insights-client --register
```

Клиент **Insights** периодически обновляет метаданные, предоставляемые Red Hat Insights. Используйте команду **insights-client** для обновления метаданных клиента в любое время.

```
[root@demo ~]# insights-client
Starting to collect Insights data for demo.lab.example.com
Uploading Insights data.
Successfully uploaded report from 773b351b-dfb1-4393-afa8-915cc2875e06 to
account XXXXX.
```

Регистрация системы RHEL в Red Hat Insights на портале для клиентов

Чтобы зарегистрировать сервер RHEL в Red Hat Insights на клиентском портале Red Hat, общий процесс выглядит следующим образом:

1. Зарегистрируйте систему в интерактивном режиме в службе управления подпиской Red Hat.

```
[root@demo ~]# subscription-manager register --auto-attach
```

К системе должно быть прикреплено действующее разрешение на Red Hat Insights, которое вы можете получить как часть подписки Smart Management. Также должны быть включены стандартные каналы пакетов Red Hat Enterprise Linux.

2. Убедитесь, что в системе установлен пакет **insights-client**. В RHEL 7 этот пакет находится в канале **rhel-7-server-rpms**.

```
[root@demo ~]# yum install insights-client
```

3. Используйте команду **insights-client --register**, чтобы зарегистрировать систему в службе Red Hat Insights и загрузить исходные метаданные системы.

```
[root@demo ~]# insights-client --register
```

4. Убедитесь, что система видна по адресу <https://access.redhat.com/insights>.

Рисунок 16.35: Обзор Red Hat Insights на портале для клиентов

The screenshot shows the Red Hat Insights web interface. The left sidebar contains navigation links: Overview, Actions, Inventory, Planner, Rules, Executive Report, Configuration, Customer Portal, and Logout. The main content area is divided into several sections: 'Latest' (news about blog posts), 'Actions Summary' (with a bar chart for Low, Medium, High, and Critical issues), 'Plan Summary' (listing newest plans like MyRemediation, LeapSecondFix, and blueborne), and an 'Optimize Your Experience' banner. The banner displays system registration statistics (1/100 registered, 99 more available), an evaluation expiration date (Sat Aug 11 2018), and two call-to-action buttons: 'Protect yourself!' (13 results) and 'Increase your uptime!' (1 result).

ПРОСМОТР ОТЧЕТОВ RED HAT INSIGHTS

Отчет Red Hat Insights показывает состояние системы с течением времени. С помощью этих отчетов вы можете легко просматривать текущие оценки рисков и определять исторические тенденции, чтобы улучшить принятие решений.

Интерфейс Red Hat Insights предоставляет вам информацию, которая включает в себя:

- Текущая общая оценка риска на основе ваших зарегистрированных систем.
- Рекомендуемые действия для ваших систем, которые далее разбиты на категории и уровни серьезности.
- Информация о том, когда системы последний раз регистрировались с помощью Red Hat Insights.
- Проблемы, которым необходимо определить приоритетность в зависимости от их воздействия.

Определение вкладок интерфейса Red Hat Insights на клиентском портале

Веб-интерфейс Red Hat Insights на клиентском портале содержит следующие вкладки:

Обзор (Overview)

Вкладка «Обзор (Overview)» представляет собой инструментальную панель высокого уровня, отображающую состояние ваших систем.

В разделе «Последние новости» представлены обновления и объявления Red Hat. В разделе «Новые системы (Newest Systems)» показаны названия систем, которые вы недавно зарегистрировали в Red Hat Insights, со ссылками на информацию о них. Сводка действий содержит сводку и ссылки на рекомендуемые действия, которые вы должны предпринять в своих системах, с разбивкой по категориям: низкая, средняя, высокая и критическая (**Low, Medium, High, и Critical**). Сводка плана и сводка правил содержат информацию о планах обслуживания, которые вы добавили, и правилах анализа, которые Red Hat добавила за последние 15 дней.

Действия (Actions)

На вкладке «Действия (Actions)» представлен более подробный обзор рекомендуемых действий.

На диаграмма «Проблемы (Issues)» отображается общее количество проблем, влияющих на ваши системы, и процент проблем, относящихся к разным категориям: доступность, стабильность, производительность и безопасность (Availability, Stability, Performance, и Security). Вы можете щелкать по различным частям диаграммы, чтобы получить более подробную информацию.

Она также отображает гистограмму сводной информации о рисках, в которой проблемы классифицируются по степени риска, как показано на вкладке «Обзор (Overview)». Вы также можете щелкнуть столбцы на этом графике для получения более подробной информации.

Рисунок 16.36: Страница действия (Actions) в Red Hat Insights на клиентском портале



Инвентарь (Inventory)

На вкладке «Инвентарь (Inventory)» представлен список систем, которые вы зарегистрировали в Red Hat Insights.

Вы можете легко отфильтровать список для определенных систем, групп или поддерживаемых платформ. В столбце Last Check In отображается время последнего обновления метаданных для каждой системы. В столбце «Состояние (Status)» указано количество ожидающих действий для каждой системы, связанных с информацией об этих действиях для этой системы.

Планировщик (Planner)

На вкладке «Планировщик (Planner)» перечислены ваши текущие планы обслуживания и вы можете создавать, редактировать и удалять их. План обслуживания позволяет вам выбирать проблемы, которые влияют на определенные системы, из отчетов Red Hat Insights, выбирать между различными вариантами их решения и создавать Ansible Playbook для их смягчения или устранения.

Правила (Rules)

На вкладке «Правила (Rules)» представлен список правил, которые Red Hat Insights может использовать при анализе проблем в вашей системе.

Эти правила можно фильтровать по возрасту, общему риску, категории, вероятности и влиянию (age, total risk, category, likelihood, и impact) на зарегистрированные системы.

В интерфейсе клиентского портала вы заметите, что некоторые из этих проблем отмечены логотипом **Ansible**. Это указывает на то, что для проблемы доступен сценарий исправления **Ansible**. Проблемы с затемненным перечеркнутым логотипом Ansible не имеют руководства по устранению проблем с помощью Ansible, но могут содержать инструкции по устранению или исправлению вручную в деталях проблемы.

Исполнительный отчет (Executive Report)

На вкладке **Executive Report** отображается текущая общая оценка ваших систем на основе исторических данных. Вы можете экспортить исполнительный отчет в документ PDF, чтобы легко им поделиться.

Рисунок 16.37: Страница Executive Report в Insights на клиентском портале



ВАЖНО

Для создания отчета для руководителей у вас должно быть не менее 10 систем, зарегистрированных в **Red Hat Insights**.

Конфигурация (Configuration)

Вкладка «Конфигурация (Configuration)» управляет параметрами конфигурации для службы Red Hat Insights на клиентском портале. Например, вы можете настроить его для отправки вам еженедельного сводного отчета по электронной почте, настроить отображение систем, зарегистрированных через Red Hat Satellite, на портале клиентов и т. д.

Просмотр проблем, о которых сообщает Red Hat Insights

Чтобы просмотреть проблемы, о которых сообщает Red Hat Insights, общий процесс выглядит следующим образом:

1. Войдите на портал клиентов Red Hat и перейдите на страницу Red Hat Insights по адресу <https://access.redhat.com/insights>.

2. На портале перейдите на вкладку Обзор (**Overview**). Просмотрите все ваши зарегистрированные системы в разделе Новейшие системы. В разделе «Сводка действий (**Actions Summary**)» приводится процентное соотношение необходимых действий в зависимости от степени серьезности: низкой, средней и высокой.
3. Перейдите на вкладку Действия (**Actions**), на которой представлена дополнительная информация о любых проблемах безопасности, стабильности или критических проблемах. Щелкните график, представляющий процентное соотношение, чтобы увидеть более подробную информацию об этих проблемах.
4. Перейдите на вкладку «Инвентарь (**Inventory**)». На странице перечислены все зарегистрированные системы.
5. Щелкните зарегистрированную систему. На странице перечислены все проблемы, влияющие на систему.

ИНТЕРПРЕТИРОВАНИЕ ОТЧЕТОВ RED HAT INSIGHTS

В Red Hat Insights его правила определяют проблемы, которые он ищет в ваших системах. Red Hat часто добавляет новые правила в Red Hat Insights для проверки новых обнаруженных проблем. Правила могут искать инциденты, которые произошли в вашей системе, которые указывают на проблему, или они могут заранее предвидеть проблемы на основе текущей конфигурации вашей системы.

Когда правило соответствует вашей системе, что указывает на наличие проблемы, вместе с правилом предоставляется дополнительная информация, которая поможет вам понять проблему, расставить приоритеты для работы по ее устранению, определить, какое смягчение или исправление доступно, а также помочь вам автоматизировать ее решение.

Каждое правило классифицируется по типу, имеет краткое название и более подробное описание, объясняющее, в чем заключается проблема. Правила обычно связаны со статьями базы знаний на портале для клиентов с дополнительной информацией. Статья базы знаний может предоставить информацию о различных способах смягчения или устранения проблемы, а правила могут предоставлять Ansible Playbook или другие материалы, чтобы помочь автоматизировать смягчение и исправление.

Некоторые проблемы сложно решить, и для полного исправления может потребоваться перезагрузка или простой. В этом случае могут быть варианты смягчения проблемы путем снижения риска в качестве временной меры. Правило дает оценку риска, связанного с проблемой, по нескольким различным категориям.

Например, рассмотрим проблему безопасности, которая требует обновления пакетов ядра и перезагрузки для исправления, но которую также может быть очень сложно использовать из-за определенных временных изменений конфигурации. Вы можете сразу применить временные изменения и отложить перезагрузку до тех пор, пока не сможете запланировать окно экстренного обслуживания.

Red Hat Insights делит риск, который проблема представляет для вашей системы, на четыре категории. Он оценивает уровень риска, используя следующие уровни: очень низкий, низкий, средний и высокий (**Very Low, Low, Moderate, и High**).

Категории «Вероятность (**Likelihood**)», «Воздействие (**Impact**)», «Общий риск (**Total Risk**)» и «Риск изменения (**Risk of change**)» позволяют прогнозировать факторы риска проблемы, обнаруженной в ваших системах с подпиской.

Влияние (**Impact**)

Указывает прогнозируемый уровень воздействия этой проблемы на систему.

Вероятность (Likelihood)

Указывает вероятность того, что данная проблема повлияет на систему.

Общий риск (Total Risk)

Комбинация вероятности и воздействия устанавливает общий балл риска.

Риск изменения (Risk of change)

Указывает на риск того, что рекомендованное действие по исправлению может вызвать сбой в работе системы.

Чтобы просмотреть факторы риска, прогнозируемые различными правилами в Red Hat Insights, перейдите к Действиям (Actions) или на вкладке Правила (Rules). Каждое правило показывает значок «Вероятность (Likelihood)», «Влияние (Impact)», «Общий риск (Total Risk)» и «Риск изменения (Risk of change)».

Рисунок 16.38: Правила Red Hat Insights, применяемые к хосту, отображаемые на портале клиентов

The screenshot shows the Red Hat Insights client interface. At the top, there's a summary card for the host 'servere.lab.example.com' with details like Registration Date (2 minutes ago), Operating System (RHEL Server release 7.3 (Maipo)), and Last Check-in (2 minutes ago). Below this, there are tabs for System and Network, with Network selected. A 'RULES' section lists four security vulnerabilities:

- Security > Dnsmasq vulnerable to remote code execution via crafted DNS requests (CVE-2017-14491)
Impact: High, Likelihood: Low, Total Risk: Medium, Risk of change: Very Low
- Security > Remote code execution vulnerability in NSS via crafted base64 data (CVE-2017-5461)
Impact: High, Likelihood: Low, Total Risk: Medium, Risk of change: High
- Security > sudo vulnerable to local privilege escalation via process TTY name parsing (CVE-2017-1000368) impact: Local Privilege Escalation
Impact: Low, Likelihood: Low, Total Risk: Low, Risk of change: Very Low
- Security > NetworkManager DHCP potentially vulnerable to remote code execution (CVE-2018-1111)
Impact: Low, Likelihood: Low, Total Risk: Low, Risk of change: Very Low

Each rule entry includes an 'Expand All' link and a small circular icon with a letter 'A'.

Когда вы определили проблемы, которые хотите решить, вы можете решить их вручную или автоматически. После того, как проблема будет решена и ваш клиент Red Hat Insights загрузит новые метаданные, правило больше не должно соответствовать этой системе, и проблема должна исчезнуть из списка рекомендуемых действий.

Устранение проблемы, о которой сообщает Red Hat Insights, вручную

1. Войдите на портал клиентов Red Hat и перейдите на страницу Red Hat Insights по адресу <https://access.redhat.com/insights>.

Перейдите на вкладку "Инвентарь". Щелкните зарегестрированную систему. На отображаемой странице будут перечислены все известные проблемы, влияющие на систему.

2. В пользовательском веб-интерфейсе разверните элемент, связанный с проблемой безопасности, которую вы планируете исправить. В разделе «Шаги по устранению (Steps to resolve)» представлена информация о том, как решить проблему.
3. Выполните шаги в уязвимой системе, чтобы решить проблему.
4. После выполнения действий по исправлению запустите следующую команду от имени пользователя **root** в системе, чтобы сообщить об изменениях в Red Hat Insights:

```
[root@demo ~]# insights-client
Starting to collect Insights data for demo.lab.example.com
Uploading Insights data.
Successfully uploaded report from 773b351b-dfb1-4393-afa8-915cc2875e06 to
account xxxxxx.
```

5. В веб-интерфейсе пользователя Red Hat Insights перейдите на вкладку Inventory. Щелкните имя системы в столбце «Имя системы» и убедитесь, что проблема больше не отображается в списке проблем.



РЕКОМЕНДАЦИИ

Страницы руководства **man insights-client(8)** и **insights-client.conf(5)**.

Дополнительные сведения см. в главе **GET STARTED в Руководстве по началу работы с Red Hat Insights 1.0 по адресу**

<https://access.redhat.com/products/red-hat-insights/#getstarted>

Дополнительную информацию об обновлениях функций для Red Hat Insights можно найти по адресу

https://access.redhat.com/documentation/en-us/red_hat_insights/1.0/htmlsingle/release_notes/#release_information

Информация о данных, собранных Red Hat Insights, доступна по адресу

System Information Collected by Red Hat Insights

<https://access.redhat.com/articles/1598863>

Информация о том, как исключить данные, собранные Red Hat Insights, доступна по адресу

Opting Out of Sending Metadata from Red Hat Insights Client

<https://access.redhat.com/articles/2025273>

КОНТРОЛЬНЫЙ ОПРОС

ОБНАРУЖЕНИЕ И РЕШЕНИЕ ПРОБЛЕМ В RED HAT INSIGHTS

Выберите правильные ответы на следующие вопросы:

1. В каком порядке происходят следующие события при управлении системой Red Hat Enterprise Linux с помощью Red Hat Insights?
 1. Red Hat Insights анализирует метаданные системы, чтобы определить, какие проблемы и рекомендации применимы.
 2. Клиент Insights загружает системные метаданные в службу Red Hat Insights.
 3. Администратор просматривает рекомендуемые действия на клиентском портале Red Hat Insights.
 4. Red Hat Insights собирает системные метаданные в системе Red Hat Enterprise Linux.
- a. 1, 2, 3, 4
b. 4, 2, 1, 3
c. 4, 2, 3, 1
d. 4, 1, 2, 3
5. Какая команда используется для регистрации клиента в Red Hat Insights?
 - a. **insights-client --register**
 - b. **insights-client --no-upload**
 - c. **subscription-manager register**
 - d. **insights-client –unregister**
6. На какой вкладке клиентского портала **Red Hat Insights** вы можете отображать список зарегистрированных хостов с использованием фильтров, основанных на состоянии системы?
 - a. Actions
 - b. Executive Report
 - c. Planner
 - d. Inventory
 - e. Rules

РЕШЕНИЕ

ОБНАРУЖЕНИЕ И РЕШЕНИЕ ПРОБЛЕМ В RED HAT INSIGHTS

Выберите правильные ответы на следующие вопросы:

1. В каком порядке происходят следующие события при управлении системой Red Hat Enterprise Linux с помощью Red Hat Insights?
 1. Red Hat Insights анализирует метаданные системы, чтобы определить, какие проблемы и рекомендации применимы.
 2. Клиент Insights загружает системные метаданные в службу Red Hat Insights.
 3. Администратор просматривает рекомендуемые действия на клиентском портале Red Hat Insights.
 4. Red Hat Insights собирает системные метаданные в системе Red Hat Enterprise Linux.

f. 1, 2, 3, 4
g. 4, 2, 1, 3
h. 4, 2, 3, 1
i. 4, 1, 2, 3
2. Какая команда используется для регистрации клиента в Red Hat Insights?
 - a. **insights-client --register**
 - j. insights-client --no-upload
 - k. subscription-manager register
 - l. insights-client –unregister
3. На какой вкладке клиентского портала **Red Hat Insights** вы можете отображать список зарегистрированных хостов с использованием фильтров, основанных на состоянии системы?
 - m. Actions
 - n. Executive Report
 - o. Planner
 - a. Inventory
 - p. Rules

РЕЗЮМЕ

В этой главе вы узнали:

- **Веб-консоль** - это веб-интерфейс управления вашим сервером на основе службы **Cockpit** с открытым исходным кодом.
- **Веб-консоль** предоставляет графики производительности системы, графические инструменты для управления конфигурацией системы и проверки журналов, а также интерактивные интерфейсы терминала.
- **Портал для клиентов Red Hat** предоставляет вам доступ к документации, загружаемым материалам, инструментам оптимизации, управлению обращениями в службу поддержки, а также управлению подписками и правами для ваших продуктов Red Hat.
- **redhat-support-tool** - это инструмент командной строки для запроса базы знаний и работы с обращениями в службу поддержки из командной строки сервера.
- **Red Hat Insights** - это инструмент прогнозной аналитики на основе SaaS, который поможет вам выявлять и устранять угрозы безопасности, производительности, доступности и стабильности ваших систем.

ГЛАВА 17

ВСЕСТОРОННИЙ ОБЗОР

ЦЕЛЬ

Обзор задач **Red Hat System Administration I.**

ЗАДАЧИ

- Обзор задач **Red Hat System Administration I.**

РАЗДЕЛЫ

- Всесторонний обзор

ЛАБОРАТОРНЫЕ РАБОТЫ

- Лабораторная работа: Управление файлами из командной строки.
- Лабораторная работа: Управление пользователями и группами, разрешениями и процессами.
- Лабораторная работа: Настройка и управление сервером.
- Лабораторная работа: Управление сетями.
- Лабораторная работа: Монтирование файловых систем и поиск файлов.

ВСЕСТОРОННИЙ ОБЗОР

ЗАДАЧИ

После завершения этого раздела студенты должны проверить и освежить знания и навыки, полученные в **Red Hat System Administration I**.

ОБЗОР СИСТЕМНОГО АДМИНИСТРИРОВАНИЯ RED HAT I

Прежде чем приступить к всестороннему обзору этого курса, студенты должны хорошо разбираться в темах, затронутых в каждой главе.

Студенты могут обратиться к более ранним разделам учебника для дополнительного изучения.

Глава 1, Начало работы с Red Hat Enterprise Linux.

Опишите и определите открытый исходный код, Linux, дистрибутивы Linux и Red Hat Enterprise Linux.

- Определите и объясните назначение Linux, дистрибутивов Linux с открытым исходным кодом и Red Hat Enterprise Linux.

Глава 2, Доступ к командной строке.

Войдите в систему Linux и выполните простые команды с помощью оболочки.

- Войдите в систему Linux на локальной текстовой консоли и выполните простые команды с её помощью.
- Войдите в систему Linux, используя среду рабочего стола GNOME 3, и выполните команды из командной строки в программе терминала.
- Экономьте время, используя для авто завершения клавишу табуляции, историю команд и редактирования команд для выполнения команд в оболочке **Bash**.

Глава 3, Управление файлами из командной строки

Копируйте, перемещайте, создавайте, удаляйте и систематизируйте файлы во время работы из оболочки **Bash**.

- Опишите, как Linux организует файлы и назначение различных каталогов в иерархии файловой системы.
- Укажите расположение файлов относительно текущего рабочего каталога и по абсолютному расположению, определите и измените рабочий каталог и составьте список содержимого каталогов.
- Создавать, копировать, перемещать и удалять файлы и каталоги.
- Сделайте так, чтобы несколько имен файлов ссылались на один и тот же файл, используя жесткие ссылки и символические (или «программные») ссылки.
- Эффективно запускайте команды, влияющие на многие файлы, с помощью функций сопоставления шаблонов оболочки Bash.

Глава 4, Получение справки в Red Hat Enterprise Linux

Решайте проблемы с помощью местных справочных систем.

- Найдите информацию на страницах руководства **man** по локальной системе Linux.
- Найдите информацию в местной документации в GNU Info.

Глава 5, Создание, просмотр и редактирование текстовых файлов

Создавайте, просматривайте и редактируйте текстовые файлы из вывода команд или в текстовом редакторе.

- Сохранять выходные данные команд или ошибки в файл с перенаправлением оболочки и обрабатывать выходные данные команд с помощью нескольких программ командной строки с конвейерами.
- Создавать и редактировать текстовые файлы с помощью редактора **vim**.
- Используйте переменные оболочки для облегчения выполнения команд и редактируйте сценарии запуска **Bash**, чтобы установить переменные оболочки и среды для изменения поведения оболочки и программ, запускаемых из оболочки.

Глава 6, Управление локальными пользователями и группами

Создание, управление и удаление локальных пользователей и групп, а также администрирование локальных политик паролей.

- Опишите назначение пользователей и групп в системе Linux.
- Переключитесь на учетную запись суперпользователя для управления системой Linux и предоставьте другим пользователям доступ суперпользователя с помощью команды **sudo**.
- Создавать, изменять и удалять локально определенные учетные записи пользователей.
- Создание, изменение и удаление локально определенных групповых учетных записей.
- Установите политику управления паролями для пользователей, а также вручную блокируйте и разблокируйте учетные записи пользователей.

Глава 7, Управление доступом к файлам

Установите разрешения файловой системы Linux для файлов и интерпретируйте влияние различных настроек разрешений на безопасность.

- Составьте список разрешений файловой системы для файлов и каталогов и интерпретируйте влияние этих разрешений на доступ пользователей и групп.
- Измените права доступа и права собственности на файлы с помощью инструментов командной строки.
- Управляйте разрешениями по умолчанию для новых файлов, созданных пользователями, объясните действие специальных разрешений и используйте особые разрешения и разрешения по умолчанию, чтобы установить владельца группы файлов, созданных в определенном каталоге.

Глава 8, Мониторинг и управление процессами Linux.

Оценка и управление процессами, выполняемыми в системе Red Hat Enterprise Linux.

- Получите информацию о программах, запущенных в системе, чтобы вы могли определить статус, использование ресурсов и владельцев, для их контроля.
- Используйте управление заданиями Bash для управления несколькими процессами, запущенными из одного терминального сеанса.
- Контролируйте и завершайте процессы, не связанные с вашей оболочкой, а также принудительно завершать пользовательские сеансы и процессы.
- Опишите, что такая средняя нагрузка, и определите процессы, ответственные за высокое использование ресурсов на сервере.

Глава 9, Контроль служб и демонов

Управляйте и отслеживайте сетевые службы и системные демоны с помощью **Systemd**.

- Список системных демонов и сетевых служб, запущенных службой **systemd** и сокетами юнитов.
- Управление демонами системы и сетевыми службами с помощью **systemctl**.

Глава 10, Настройка и защита SSH

Настройте безопасную службу командной строки в удаленных системах с помощью **OpenSSH**.

- Войдите в удаленную систему и запускайте команды с помощью ssh.
- Настройте аутентификацию на основе ключей для учетной записи пользователя, чтобы безопасно входить в удаленные системы без пароля.
- Ограничьте прямой вход в систему как пользователь **root** и отключите аутентификацию на основе пароля для службы **OpenSSH**.

Глава 11, Анализ и хранение журналов

Найдите и точно интерпретируйте журналы системных событий для устранения неполадок.

- Опишите базовую архитектуру ведения журнала, используемую Red Hat Enterprise Linux для записи событий.
- Интерпретируйте события в соответствующих файлах системного журнала для устранения проблем или проверки состояния системы.
- Найдите и интерпретируйте записи в системном журнале для устранения проблем или проверки состояния системы.
- Настройте системный журнал, чтобы сохранять запись событий при перезагрузке сервера.
- Поддерживайте точную синхронизацию времени с помощью NTP и настраивайте часовой пояс, чтобы обеспечить правильные отметки времени для событий, записываемых в системный журнал и журналы логов.

Глава 12, Управление сетью

Настройте сетевые интерфейсы и параметры на серверах Red Hat Enterprise Linux.

- Опишите фундаментальные концепции сетевой адресации и маршрутизации для сервера.
- Протестируйте и проверьте текущую конфигурацию сети с помощью утилит командной строки.
- Управляйте настройками сети и устройствами с помощью **nmcli**.
- Измените настройки сети, отредактировав файлы конфигурации.
- Настройте статическое имя хоста сервера и его разрешение имен и проверьте результаты.

Глава 13, Архивирование и передача файлов

Архивируйте и копируйте файлы из одной системы в другую.

- Архивируйте файлы и каталоги в сжатый файл с помощью утилиты **tar**, и извлекайте содержимое существующего архива **tar**.
- Безопасная передача файлов в удаленную систему или из нее с помощью **SSH**.
- Синхронизируйте содержимое локального файла или каталога с копией на удаленном сервере.

Глава 14, Установка и обновление пакетов программного обеспечения

Загружайте, устанавливайте, обновляйте и управляйте пакетами программного обеспечения из репозиториев пакетов Red Hat и Yum.

- Зарегистрируйте систему в своей учетной записи Red Hat и назначьте ей права на обновления программного обеспечения и услуги поддержки с помощью управления подпиской Red Hat.
- Объясните, как программное обеспечение предоставляется в виде пакетов **RPM**, и исследуйте пакеты, установленные в системе с помощью **Yum** и **RPM**.
- Найдите, установите и обновите пакеты программного обеспечения с помощью команды **yum**.
- Включение и отключение использования сервером Red Hat или сторонних репозиториев **Yum**.
- Объясните, как модули позволяют устанавливать определенные версии программного обеспечения, перечислять, включать и переключать потоки модулей, а также устанавливать и обновлять пакеты из модуля.

Глава 15, Доступ к файловым системам Linux

Доступ, проверка и использование существующих файловых систем в хранилище, подключенном к серверу Linux.

- Объясните, что такое блочное устройство, интерпретируйте имена файлов запоминающих устройств и определите запоминающее устройство, используемое файловой системой для конкретного каталога или файла.
- Получите доступ к файловым системам, прикрепив их к каталогу в иерархии файловой системы.
- Найдите файлы в смонтированных файловых системах с помощью команд **find** и **locate**.

Глава 16. Анализ серверов и получение поддержки

Исследуйте и решайте проблемы в веб-интерфейсе управления, пользуясь поддержкой Red Hat для решения проблем.

- Активируйте интерфейс управления веб-консолью для удаленного управления и мониторинга производительности сервера Red Hat Enterprise Linux.
- Опишите ключевые ресурсы, доступные на портале для клиентов Red Hat, и найдите информацию в документации Red Hat и в базе знаний.
- Анализируйте серверы на наличие проблем, устраняйте или разрешайте их и подтверждайте решение с помощью **Red Hat Insights**.

ЛАБОРАТОРНАЯ РАБОТА

УПРАВЛЕНИЕ ФАЙЛАМИ ИЗ КОМАНДНОЙ СТРОКИ

В этом обзоре вы будете управлять файлами, перенаправлять определенный набор строк из текстового файла в другой файл и редактировать текстовые файлы.

В РЕЗУЛЬТАТЕ

Вы должны быть способны:

- Управлять файлами из командной строки.
- Отображать определенное количество строк из текстовых файлов и перенаправлять вывод в другой файл.
- Редактировать текстовые файлы.

ПРЕЖДЕ ЧЕМ ВЫ НАЧНЕТЕ

Войдите на рабочую станцию **workstation** как пользователь **student**, используя в качестве пароля слово **student**.

На рабочей станции **workstation** выполните скрипт **lab rhcsa-rh124-review1 start**, чтобы начать всесторонний обзор. Этот сценарий создает необходимые файлы для правильной настройки среды.

```
[student@workstation ~]$ lab rhcsa-rh124-review1 start
```

ИНСТРУКЦИИ

Выполните следующие задачи на сервере **serverb**, чтобы выполнить упражнение.

- Создайте новый каталог с именем **/home/student/grading**.
- Создайте три пустых файла в каталоге **/home/student/grading** с именами **grade1**, **grade2** и **grade3**.
- Сохраните первые пять строк файла **/home/student/bin/manage-files** в файле **/home/student/grading/manage-files.txt**.
- Добавьте последние три строки **/home/student/bin/manage-files** в файл **/home/student/grading/manage-files.txt**. Вы не должны перезаписывать какой-либо текст, уже находящийся в файле **/home/student/grading/manage-files.txt**.
- Скопируйте **/home/student/grading/manage-files.txt** в **/home/student/grading/manage-files-copy.txt**.
- Отредактируйте файл **/home/student/grading/manage-files-copy.txt** так, чтобы в нем были две последовательные строки текста для чтения **Test JJ**.
- Отредактируйте файл **/home/student/grading/manage-files-copy.txt** так, чтобы текстовая строка **Test HH** не существовала в файле.

- Отредактируйте файл **/home/student/grading/manage-files-copy.txt** так, чтобы между строкой, читающей **Test BB**, и строкой, читающей **Test CC**, существовала строка **A new line**.
- Создайте жесткую ссылку с именем **/home/student/hardlink** на файл **/home/student/grading/grade1**. Вам нужно будет сделать это после создания пустого файла **/home/student/grading/grade1**, как указано выше.
- Создайте символьическую ссылку с именем **/home/student/softlink** на файл **/home/student/grading/grade2**.
- Сохраните вывод команды, которая перечисляет содержимое каталога **/boot**, в файл **/home/student/grading/longlisting.txt**. Результатом должен быть «длинный список (long listing)», включающий права доступа к файлу, владельца и владельца группы, размер и дату изменения каждого файла.

Оценка

На рабочей станции **workstation** запустите команду **lab rhcsa-rh124-review1 grade**, чтобы подтвердить успех выполнения упражнения.

```
[student@workstation ~]$ lab rhcsa-rh124-review1 grade
```

Завершение

На рабочей **workstation** станции запустите **lab rhcsa-rh124-review1 finish**, чтобы завершить всесторонний обзор. Сценарий удаляет файлы и каталоги, созданные в начале всестороннего обзора, и обеспечивает чистоту среды на **serverb**.

```
[student@workstation ~]$ lab rhcsa-rh124-review1 finish
```

На этом всеобъемлющий обзор завершен.

РЕШЕНИЕ

УПРАВЛЕНИЕ ФАЙЛАМИ ИЗ КОМАНДНОЙ СТРОКИ

В этом обзоре вы будете управлять файлами, перенаправлять определенный набор строк из текстового файла в другой файл и редактировать текстовые файлы.

В РЕЗУЛЬТАТЕ

Вы должны быть способны:

- Управляйте файлами из командной строки.
- Отображать определенное количество строк из текстовых файлов и перенаправлять вывод в другой файл.
- Редактировать текстовые файлы.

ПРЕЖДЕ ЧЕМ ВЫ НАЧНЕТЕ

Войдите на рабочую станцию **workstation** как пользователь **student**, используя в качестве пароля слово **student**.

На рабочей станции **workstation** выполните скрипт **lab rhcsa-rh124-review1 start**, чтобы начать всесторонний обзор. Этот сценарий создает необходимые файлы для правильной настройки среды.

```
[student@workstation ~]$ lab rhcsa-rh124-review1 start
```

ИНСТРУКЦИИ

Выполните следующие задачи на сервере **serverb**, чтобы выполнить упражнение.

- Создайте новый каталог с именем **/home/student/grading**.
- Создайте три пустых файла в каталоге **/home/student/grading** с именами **grade1**, **grade2** и **grade3**.
- Сохраните первые пять строк файла **/home/student/bin/manage-files** в файле **/home/student/grading/manage-files.txt**.
- Добавьте последние три строки **/home/student/bin/manage-files** в файл **/home/student/grading/manage-files.txt**. Вы не должны перезаписывать какой-либо текст, уже находящийся в файле **/home/student/grading/manage-files.txt**.
- Скопируйте **/home/student/grading/manage-files.txt** в **/home/student/grading/manage-files-copy.txt**.
- Отредактируйте файл **/home/student/grading/manage-files-copy.txt** так, чтобы в нем были две последовательные строки текста для чтения **Test JJ**.
- Отредактируйте файл **/home/student/grading/manage-files-copy.txt** так, чтобы текстовая строка **Test HH** не существовала в файле.

- Отредактируйте файл **/home/student/grading/manage-files-copy.txt** так, чтобы между строкой, читающей **Test BB**, и строкой, читающей **Test CC**, существовала строка **A new line**.
- Создайте жесткую ссылку с именем **/home/student/hardlink** на файл **/home/student/grading/grade1**. Вам нужно будет сделать это после создания пустого файла **/home/student/grading/grade1**, как указано выше.
- Создайте символьическую ссылку с именем **/home/student/softlink** на файл **/home/student/grading/grade2**.
- Сохраните вывод команды, которая перечисляет содержимое каталога **/boot**, в файл **/home/student/grading/longlisting.txt**. Результатом должен быть «длинный список (long listing)», включающий права доступа к файлу, владельца и владельца группы, размер и дату изменения каждого файла.

1. Создайте новый каталог с именем **/home/student/grading**.

1.1. С рабочей станции **workstation**, откройте сеанс SSH на **serverb** как пользователь **student**.

```
[student@workstation ~]$ ssh student@serverb
...output omitted...
[student@serverb ~]$
```

1.2. Используйте команду **mkdir**, чтобы создать каталог **/home/student/grading**.

```
[student@serverb ~]$ mkdir grading
```

Поскольку вы выполнили предыдущую команду, находясь в домашнем каталоге пользователя **student**, вы не указали абсолютный путь к каталогу **grading** при его создании.

2. Создайте три пустых файла в каталоге **/home/student/grading** с именами **grade1**, **grade2** и **grade3**.

2.1. С помощью команды **touch** создайте пустые файлы с именами **grade1**, **grade2** и **grade3** в каталоге **/home/student/grading**. Примените функцию оболочки расширения скобок, чтобы создать все три файла с помощью одной команды **touch**.

```
[student@serverb ~]$ touch grading/grade{1,2,3}
```

2.2. Используйте команду **ls**, чтобы убедиться, что файлы **grade1**, **grade2** и **grade3** существуют в каталоге **/home/student/grading**.

```
[student@serverb ~]$ ls grading/
grade1 grade2 grade3
```

3. Сохраните первые пять строк файла **/home/student/bin/manage-file** в файле **/home/student/grading/manage-files.txt**.

- 3.1.** Используйте команду **head**, чтобы просмотреть первые пять строк файла **/home/student/bin/manage-file** и перенаправить вывод в файл **/home/student/grading/manage-files.txt**.

```
[student@serverb ~]$ head -5 bin/manage-files > grading/manage-files.txt
```

В предыдущей команде используется единственный символ перенаправления (**>**) для сохранения вывода команды в **/home/student/grading/manage-files.txt**, чтобы любое существующее содержимое в файле было перезаписано.

- 3.2.** Убедитесь, что файл **/home/student/grading/manage-files.txt** содержит следующий текст.

```
Test AA  
Test BB  
Test CC  
Test DD  
Test EE
```

- 4.** Добавьте последние три строки **/home/student/bin/manage-file** в **/home/student/grading/manage-files.txt**. Вы не должны перезаписывать какой-либо текст, уже находящийся в файл **/home/student/grading/manage-files.txt**.

- 4.1.** Используйте команду **tail** для просмотра последних трех строк файла **/home/student/bin/manage-file** и добавьте результат в **/home/student/grading/managefiles.txt**.

```
[student@serverb ~]$ tail -3 bin/manage-files >> grading/manage-files.txt
```

Предыдущая команда использует символ двойного перенаправления (**>>**) для добавления вывода в **/home/student/grading/manage-files.txt**, чтобы сохранить существующее содержимое в файле.

- 4.2.** Убедитесь, что файл **/home/student/grading/manage-files.txt** содержит следующий текст.

```
Test AA  
Test BB  
Test CC  
Test DD  
Test EE  
Test HH  
Test II  
Test JJ
```

- 5.** Скопируйте файл **/home/student/grading/manage-files.txt** в **/home/student/grading/manage-files-copy.txt**.

- 5.1.** Используйте команду **cd**, чтобы перейти в каталог **/home/student/grading**.

```
[student@serverb ~]$ cd grading/  
[student@serverb grading]$
```

- 5.2.** Используйте команду **cp**, чтобы скопировать файл **/home/student/grading/manage-files.txt** в **/home/student/grading/manage-files-copy.txt**.

```
[student@serverb grading]$ cp manage-files.txt manage-files-copy.txt
```

- 5.3.** Вернитесь в домашний каталог пользователя **student**.

```
[student@serverb grading]$ cd  
[student@serverb ~]$
```

- 6.** Отредактируйте файл **/home/student/grading/manage-files-copy.txt** так, чтобы в нем были две последовательные строки текста для чтения **Test JJ**.

- 6.1.** Используйте текстовый редактор **vim**, чтобы открыть файл **/home/student/grading/manage-files-copy.txt**.

```
student@serverb ~]$ vim grading/manage-files-copy.txt
```

- 6.2.** В командном режиме в **vim** прокрутите вниз до строки, содержащей строку текста **Test JJ**. Дважды нажмите клавишу **u** на клавиатуре, чтобы скопировать строку текста, и нажмите клавишу **p**, чтобы вставить ее под курсором. Введите **:wq**, чтобы сохранить изменения и выйти из **vim**. Убедитесь, что файл **/home/student/grading/manage-files-copy.txt** содержит следующий текст.

```
Test AA  
Test BB  
Test CC  
Test DD  
Test EE  
Test HH  
Test II  
Test JJ  
Test JJ
```

Обратите внимание, что предыдущее содержимое включает две копии строки текста **Test JJ**.

- 7.** Отредактируйте файл **/home/student/grading/manage-files-copy.txt** так, чтобы текстовая строка **Test HH** не существовала в файле.

- 7.1.** Используйте текстовый редактор **vim**, чтобы открыть файл **/home/student/grading/manage-files-copy.txt**.

```
[student@serverb ~]$ vim grading/manage-files-copy.txt
```

- 7.2.** Из командного режима в **vim** прокрутите вниз до строки, в которой есть строка текста **Test HH**. Дважды нажмите клавишу **d** на клавиатуре, чтобы удалить строку текста. Введите **:wq**, чтобы сохранить изменения и выйти из **vim**. Убедитесь, что файл **/home/student/grading/managefiles-copy.txt** содержит следующий текст.

```
Test AA  
Test BB  
Test CC  
Test DD  
Test EE  
Test II  
Test JJ  
Test JJ
```

Обратите внимание, что предыдущее содержимое не включает строку текста **Test HH**.

- 8.** Отредактируйте файл **/home/student/grading/manage-files-copy.txt** так, чтобы между строкой **Test BB** и строкой **Test CC** существовала строка **new line**.

- 8.1.** Используйте текстовый редактор **vim**, чтобы открыть файл **/home/student/grading/manage-files-copy.txt**.

```
[student@serverb ~]$ vim grading/manage-files-copy.txt
```

- 8.2.** Из командного режима в **vim** прокрутите вниз до строки, в которой есть строка текста **Test CC**. Нажмите клавишу **i** на клавиатуре, чтобы переключиться в режим вставки, удерживая курсор в начале строки текста **Test CC**. В режиме вставки нажмите клавишу **Enter** на клавиатуре, чтобы создать пустую строку над курсором. Используйте стрелку вверх, чтобы перейти к пустой строке и создать новую строку текста. Нажмите клавишу **Esc** на клавиатуре, чтобы вернуться в командный режим. Введите **:wq**, чтобы сохранить изменения и выйти из **vim**. Убедитесь, что файл **/home/student/grading/manage-files-copy.txt** содержит следующий текст.

```
Test AA  
Test BB  
A new line  
Test CC  
Test DD  
Test EE  
Test II  
Test JJ  
Test JJ
```

Обратите внимание, что предыдущее содержимое включает текст **A new line**.

9. Создайте жесткую ссылку с именем **/home/student/hardlink** на файл **/home/student/grading/grade1**.

9.1. Используйте команду **ln**, чтобы создать жесткую ссылку с именем **/home/student/hardlink** на файл **/home/student/grading/grade1**. Вам нужно будет сделать это после создания пустого файла **/home/student/grading/grade1**, как указано выше.

```
[student@serverb ~]$ ln grading/grade1 hardlink
```

9.2. Используйте команду **ls -l**, чтобы просмотреть количество ссылок в файле **/home/student/grading/grade1**.

```
[student@serverb ~]$ ls -l grading/grade1
-rw-rw-r--. 2 student student 0 Mar 6 16:45 grading/grade1
```

10. Создайте символьную ссылку с именем **/home/student/softlink** на файл **/home/student/grading/grade2**.

10.1. Используйте команду **ln -s**, чтобы создать символьную ссылку с именем **/home/student/softlink** на файл **/home/student/grading/grade2**.

10.2. Используйте команду **ls -l** для просмотра свойств символьной ссылки **/home/student/softlink**.

```
[student@serverb ~]$ ls -l softlink
lrwxrwxrwx. 1 student student 14 Mar 6 17:58 softlink -> grading/grade2
```

11. Сохраните вывод команды, которая перечисляет содержимое каталога **/boot**, в файл **/home/student/grading/longlisting.txt**. Результатом должен быть «**long listing**», включающий права доступа к файлу, владельца и владельца группы, размер и дату изменения каждого файла.

11.1. Используйте команду **ls -l**, чтобы просмотреть содержимое каталога **/boot** в формате «**длинного списка**» и перенаправить вывод в файл **/home/student/grading/longlisting.txt**.

```
[student@serverb ~]$ ls -l /boot > grading/longlisting.txt
```

11.2. Выйти из сервера **serverb**.

```
[student@serverb ~]$ exit
```

```
logout  
Connection to serverb closed.
```

Оценка

На рабочей станции **workstation** запустите команду **lab rhcsa-rh124-review1 grade**, чтобы подтвердить успех выполнения упражнения.

```
[student@workstation ~]$ lab rhcsa-rh124-review1 grade
```

Завершение

На рабочей **workstation** станции запустите **lab rhcsa-rh124-review1 finish**, чтобы завершить всесторонний обзор. Сценарий удаляет файлы и каталоги, созданные в начале всестороннего обзора, и обеспечивает чистоту среды на **serverb**.

```
[student@workstation ~]$ lab rhcsa-rh124-review1 finish
```

На этом всеобъемлющий обзор завершен.

ЛАБОРОТОРНАЯ РАБОТА

УПРАВЛЕНИЕ ПОЛЬЗОВАТЕЛЯМИ И ГРУППАМИ, РАЗРЕШЕНИЯМИ И ПРОЦЕССАМИ

В этом обзоре вы будете управлять учетными записями пользователей и групп, устанавливать разрешения для файлов и каталогов и управлять процессами.

В РЕЗУЛЬТАТЕ

Вы будете способны:

- Управляйте пользователями и группами.
- Установите разрешения для файлов и каталогов.
- Удалять процессы, которые потребляют слишком много ресурсов ЦП.

ПРЕЖДЕ ЧЕМ ВЫ НАЧНЕТЕ

Войдите на рабочую станцию **workstation** как пользователь **student**, используя в качестве пароля слово **student**.

На рабочей станции **workstation** выполните скрипт **lab rhcsa-rh124-review2 start**, чтобы начать всесторонний обзор. Данный сценарий запускает процесс, который потребляет максимум ресурсов ЦП и создает файлы, необходимые для правильной настройки среды.

```
[student@workstation ~]$ lab rhcsa-rh124-review2 start
```

ИНСТРУКЦИИ

Выполните следующие задачи на хосте **serverb**, чтобы выполнить упражнение.

- Завершите процесс, который в настоящее время использует больше всего процессорного времени.
- Создайте новую группу под названием **database** с **GID 5000**.
- Создайте нового пользователя с именем **dbuser1**, который будет использовать группу **database** в качестве одной из своих вторичных групп. Первоначальный пароль **dbuser1** должен быть установлен **redhat**. Настройте пользователя **dbuser1** на принудительную смену пароля при первом входе в систему. Пользователь **dbuser1** должен иметь возможность изменить свой пароль через **10** дней со дня смены пароля. Срок действия пароля **dbuser1** должен истечь через **30** дней с последнего дня смены пароля.
- Настройте пользователя **dbuser1** на использование **sudo** для выполнения любой команды от имени суперпользователя.
- Настройте для пользователя **dbuser1** значение **umask** по умолчанию **007**.
- Разрешения на **/home/student/grading/review2** должны позволять членам группы **database** и пользователю **student** получать доступ к каталогу и создавать в нем содержимое. Все остальные пользователи должны иметь разрешения на чтение и выполнение в каталоге.

Оценка

На рабочей станции **workstation** запустите команду **lab rhcsa-rh124-review2 grade**, чтобы подтвердить успех выполнения упражнения.

```
[student@workstation ~]$ lab rhcsa-rh124-review2 grade
```

Завершение

На рабочей станции **workstation** запустите **lab rhcsa-rh124-review2 finish**, чтобы завершить всесторонний обзор. Этот сценарий завершает процесс и удаляет файлы и каталоги, созданные во время начала комплексной проверки, и обеспечивает чистоту среды на хосте **serverb**.

```
[student@workstation ~]$ lab rhcsa-rh124-review2 finish
```

На этом всеобъемлющий обзор завершен.

РЕШЕНИЕ

УПРАВЛЕНИЕ ПОЛЬЗОВАТЕЛЯМИ И ГРУППАМИ, РАЗРЕШЕНИЯМИ И ПРОЦЕССАМИ

В этом обзоре вы будете управлять учетными записями пользователей и групп, устанавливать разрешения для файлов и каталогов и управлять процессами.

В РЕЗУЛЬТАТЕ

Вы будете способны:

- Управляйте пользователями и группами.
- Установите разрешения для файлов и каталогов.
- Удалять процессы, которые потребляют слишком много ресурсов ЦП.

ПРЕЖДЕ ЧЕМ ВЫ НАЧНЕТЕ

Войдите на рабочую станцию **workstation** как пользователь **student**, используя в качестве пароля слово **student**.

На рабочей станции **workstation** выполните скрипт **lab rhcsa-rh124-review2 start**, чтобы начать всесторонний обзор. Данный сценарий запускает процесс, который потребляет максимум ресурсов ЦП и создает файлы, необходимые для правильной настройки среды.

```
[student@workstation ~]$ lab rhcsa-rh124-review2 start
```

ИНСТРУКЦИИ

Выполните следующие задачи на хосте **serverb**, чтобы выполнить упражнение.

- Завершите процесс, который в настоящее время использует больше всего процессорного времени.
- Создайте новую группу под названием **database** с **GID 5000**.
- Создайте нового пользователя с именем **dbuser1**, который будет использовать группу **database** в качестве одной из своих вторичных групп. Первоначальный пароль **dbuser1** должен быть установлен **redhat**. Настройте пользователя **dbuser1** на принудительную смену пароля при первом входе в систему. Пользователь **dbuser1** должен иметь возможность изменить свой пароль через **10** дней со дня смены пароля. Срок действия пароля **dbuser1** должен истечь через **30** дней с последнего дня смены пароля.
- Настройте пользователя **dbuser1** на использование **sudo** для выполнения любой команды от имени суперпользователя.
- Настройте для пользователя **dbuser1** значение **umask** по умолчанию **007**.
- Разрешения на **/home/student/grading/review2** должны позволять членам группы **database** и пользователю **student** получать доступ к каталогу и создавать в нем содержимое. Все остальные пользователи должны иметь разрешения на чтение и выполнение в каталоге.

Кроме того, убедитесь, что пользователям разрешено удалять только принадлежащие им файлы `/home/student/grading/review2`, а не файлы, принадлежащие другим лицам.

1. Завершите процесс, который в настоящее время использует больше всего процессорного времени.

- 1.1. С рабочей станции **workstation** откройте сеанс **SSH** на хост **serverb** как пользователь **student**.

```
[student@workstation ~]$ ssh student@serverb
...output omitted...
[student@serverb ~]$
```

- 1.2. Используйте команду **top** для просмотра состояния системы в реальном времени.

```
[student@serverb ~]$ top
```

- 1.3. В интерактивном просмотре вывода команды **top** обратите внимание на столбец **%CPU** и убедитесь, что существует процесс с именем **dd**, потребляющий больше всего ресурсов ЦП.

```
...output omitted...
 PID USER      PR  NI    VIRT    RES    SHR S %CPU %MEM     TIME+ COMMAND
 2303 student    20   0  217048   944    876 R 99.7  0.1 100:11.64 dd
 ...output omitted...
```

Обратите внимание на процесс **dd** с **PID 2303** в предыдущем выводе, который потребляет большую часть ресурсов ЦП на 99,7%. PID и процент потребления ресурсов ЦП могут отличаться в вашей системе.

- 1.4. В интерактивном интерфейсе команды **top** введите **k**, чтобы завершить (убить(**kill**)) процесс **dd** с PID 2303, как вы определили на предыдущем шаге. Если PID по умолчанию, показанный в приглашении, совпадает с идентификатором процесса, потребляющего большую часть ресурсов ЦП, нажмите клавишу **Enter** на клавиатуре. Если он не совпадает, укажите PID в интерактивном режиме.

```
...output omitted...
PID to signal/kill [default pid = 2303] Enter
...output omitted...
```

- 1.5. Используйте сигнал по умолчанию **SIGTERM**, чтобы завершить процесс.

```
...output omitted...
Send pid 2833 signal [15/sigterm] Enter
...output omitted...
```

- 1.6. В интерактивном интерфейсе команды **top** нажмите клавишу **q** на клавиатуре, чтобы завершить выполнение команды.
2. Создайте новую группу с именем **database** с **GID 50000**.

2.1. Переключитесь на пользователя **root**.

```
[student@serverb ~]$ sudo su -  
[sudo] password for student: student  
[root@serverb ~]#
```

2.2. Используйте команду **groupadd**, чтобы создать новую группу с именем **database** с **GID 50000**.

```
[root@serverb ~]# groupadd -g 50000 database
```

3. Создайте нового пользователя с именем **dbuser1** с групповой **database** в качестве одной из вторичных групп. Установите начальный пароль **dbuser1** как **redhat**. Настройте пользователя **dbuser1** на принудительную смену пароля при первом входе в систему. Пользователь **dbuser1** должен иметь возможность изменить свой пароль через **10** дней с последнего дня смены пароля. Срок действия пароля **dbuser1** должен истечь через **30** дней с последнего дня смены пароля.

3.1. Используйте команду **useradd**, чтобы создать нового пользователя с именем **dbuser1**, который использует **database** в качестве одной из своих вторичных групп.

```
[root@serverb ~]# useradd -G database dbuser1
```

3.2. Используйте команду **passwd**, чтобы установить для **dbuser1** пароль **redhat**.

```
[root@serverb ~]# passwd dbuser1  
Changing password for user dbuser1.  
New password: redhat  
BAD PASSWORD: The password is shorter than 8 characters  
Retype new password: redhat  
passwd: all authentication tokens updated successfully.
```

3.3. Используйте команду **chage**, чтобы заставить **dbuser1** изменить свой пароль при первом входе в систему.

```
[root@serverb ~]# chage -d 0 dbuser1
```

3.4. Используйте команду **chage**, чтобы установить минимальный возраст пароля **dbuser1** до **10** дней.

```
[root@serverb ~]# chage -m 10 dbuser1
```

- 3.5. Используйте команду **chage**, чтобы установить максимальный срок действия пароля **dbuser1** до **30** дней.

```
[root@serverb ~]# chage -M 30 dbuser1
```

4. Создайте файл **/etc/sudoers.d/dbuser1** для настройки **dbuser1**, чтобы пользователь мог использовать **sudo** для выполнения любой команды в качестве **суперпользователя**. Вы можете использовать команду **vim** **/etc/sudoers.d/dbuser1** для создания файла. **/etc/sudoers.d/dbuser1** должен содержать следующие строки.

```
dbuser1 ALL=(ALL) ALL
```

5. Настройте для пользователя **dbuser1** значение **umask** по умолчанию **007**.

- 5.1. Переключитесь на пользователя **dbuser1**.

```
[root@serverb ~]# su - dbuser1  
[dbuser1@serverb ~]$
```

- 5.2. Добавьте строку **umask 007** в файлы **/home/dbuser1/.bash_profile** и **/home/dbuser1/.bashrc**.

```
[dbuser1@serverb ~]$ echo "umask 007" >> .bash_profile  
[dbuser1@serverb ~]$ echo "umask 007" >> .bashrc
```

- 5.3. Выйдите из оболочки пользователя **dbuser1**.

6. Создайте новый каталог с именем **/home/student/grading/review2**, в котором пользователь **student** и группа **database** будут его владельцем и группой соответственно. Настройте разрешения в этом каталоге так, чтобы любой новый файл в нем наследовал **database** в качестве группы-владельца независимо от создающего пользователя. Разрешения на **/home/student/grading/review2** должны позволять членам группы **database** и пользователю **student** получать доступ к каталогу и создавать в нем содержимое. Все остальные пользователи должны иметь разрешения на чтение и выполнение в каталоге. Также убедитесь, что пользователям разрешено удалять только принадлежащие им файлы из **/home/student/grading/review2**, а не файлы других лиц.

- 6.1. Используйте команду **mkdir**, чтобы создать **/home/student/grading/review2**.

```
[root@serverb ~]# mkdir /home/student/grading/review2
```

6.2. В **/home/student/grading/review2** используйте команду **chown**, чтобы установить **student** и **database** в качестве пользователя-владельца и группы соответственно.

```
[root@serverb ~]# chown student:database /home/student/grading/review2
```

6.3. Используйте команду **chmod**, чтобы применить специальное разрешение **SetGID** для **/home/student/grading/review2**.

```
[root@serverb ~]# chmod g+s /home/student/grading/review2
```

6.4. Используйте команду **chmod**, чтобы применить режим разрешений **775** для **/home/student/grading/review2**.

```
[root@serverb ~]# chmod 775 /home/student/grading/review2
```

6.5. Используйте команду **chmod**, чтобы применить специальное разрешение **stickybit** к **/home/student/grading/review2**.

```
[root@serverb ~]# chmod o+t /home/student/grading/review2
```

6.6. Выйдите из оболочки пользователя **root**.

```
[root@serverb ~]# exit  
logout  
[student@serverb ~]$
```

Оценка

На рабочей станции **workstation** запустите команду **lab rhcsa-rh124-review2 grade**, чтобы подтвердить успех выполнения упражнения.

```
[student@workstation ~]$ lab rhcsa-rh124-review2 grade
```

Завершение

На рабочей станции **workstation** запустите **lab rhcsa-rh124-review2 finish**, чтобы завершить всесторонний обзор. Этот сценарий завершает процесс и удаляет файлы и каталоги, созданные во время начала комплексной проверки, и обеспечивает чистоту среды на хосте **serverb**.

```
[student@workstation ~]$ lab rhcsa-rh124-review2 finish
```

На этом всеобъемлющий обзор завершен.

ЛАБОРАТОРНАЯ РАБОТА

НАСТРОЙКА И УПРАВЛЕНИЕ СЕРВЕРА

В этом обзоре вы будете настраивать, защищать и использовать службу **SSH** для доступа к удаленному компьютеру, настраивать службу **rsyslog**, архивировать локальные файлы, передавать локальные файлы на удаленную машину и управлять пакетами с помощью команды **yum**.

В РЕЗУЛЬТАТЕ

Вы должны быть способны:

- Создать новую пару ключей **SSH**.
- Отключите вход по **SSH** от имени пользователя **root**.
- Отключить вход по **SSH** с помощью пароля.
- Обновить часовой пояс сервера.
- Установите пакеты и модули пакетов с помощью команды **yum**.
- Архивировать локальные файлы для резервного копирования.
- Перенести локальные файлы на удаленную машину.

ПРЕЖДЕ ЧЕМ ВЫ НАЧНЕТЕ

Войдите на рабочую станцию **workstation** как пользователь **student**, используя в качестве пароля слово **student**.

На рабочей станции **workstation** выполните скрипт **lab rhcsa-rh124-review3 start**, чтобы начать всесторонний обзор. Этот сценарий создает необходимые файлы для правильной настройки среды.

ИНСТРУКЦИИ

Выполните следующие задачи на хосте **serverb**, чтобы завершить упражнение.

- Сгенерируйте ключи **SSH** для пользователя **student** на **serverb**. Не защищайте закрытый ключ парольной фразой.
- На сервере настройте пользователя **student**, чтобы он принимал входы в систему, **аутентифицированные парой ключей SSH**, которую вы создали для пользователя **student** на **serverb**. Пользователь **student** на **serverb** должен иметь возможность входить в систему с помощью **SSH** без ввода пароля.
- На **serverb** настройте службу **sshd**, чтобы пользователи не могли входить в систему как **root** через **SSH**.
- На **serverb** настройте службу **sshd**, чтобы пользователи не могли использовать свои пароли для входа в систему. Пользователи по-прежнему должны иметь возможность аутентифицировать вход с помощью пары ключей **SSH**.
- Создайте **tar**-архив с именем **/tmp/log.tar**, содержащий содержимое **/var/log** на **serverb**. Удаленно передайте **tar**-архив в каталог **/tmp** на сервере, аутентифицируясь как пользователь **student**, используя закрытый ключ пользователя **student** из пары ключей **SSH**.
- Настройте службу **rsyslog** на **serverb**, чтобы все полученные сообщения с уровнем приоритета отладки или выше регистрировались в файле **/var/log/grading-debug**. Эта конфигурация должна быть установлена в файле **/etc/rsyslog.d/grading-debug.conf**, который вам необходимо создать.

- Установите пакет **zsh**, доступный в репозитории **BaseOS**, на **serverb**.
- Включите **stream** модули по умолчанию для модуля **python36** и установите все предоставленные пакеты из этого потока на **serverb**.
- Установите часовой пояс сервера **serverb** на **Asia/Kolkata**.

Оценка

На рабочей станции **workstation** запустите команду **lab rhcsa-rh124-review3 grade**, чтобы подтвердить успех выполнения упражнения.

```
[student@workstation ~]$ lab rhcsa-rh124-review3 grade
```

Завершение

На рабочей станции запустите скрипт **lab rhcsa-rh124-review3 finish**, чтобы завершить всесторонний обзор. Сценарий удаляет файлы и каталоги, созданные в начале всестороннего обзора, и обеспечивает чистоту среды на хосте **serverb**.

```
[student@workstation ~]$ lab rhcsa-rh124-review3 finish
```

На этом всеобъемлющий обзор завершен.

РЕШЕНИЕ

НАСТРОЙКА И УПРАВЛЕНИЕ СЕРВЕРА

В этом обзоре вы будете настраивать, защищать и использовать службу **SSH** для доступа к удаленному компьютеру, настраивать службу **rsyslog**, архивировать локальные файлы, передавать локальные файлы на удаленную машину и управлять пакетами с помощью команды **yum**.

В РЕЗУЛЬТАТЕ

Вы должны быть способны:

- Создать новую пару ключей **SSH**.
- Отключите вход по **SSH** от имени пользователя **root**.
- Отключить вход по **SSH** с помощью пароля.
- Обновить часовой пояс сервера.
- Установите пакеты и модули пакетов с помощью команды **yum**.
- Архивировать локальные файлы для резервного копирования.
- Перенести локальные файлы на удаленную машину.

ПРЕЖДЕ ЧЕМ ВЫ НАЧНЕТЕ

Войдите на рабочую станцию **workstation** как пользователь **student**, используя в качестве пароля слово **student**.

На рабочей станции **workstation** выполните скрипт **lab rhcsa-rh124-review3 start**, чтобы начать всесторонний обзор. Этот сценарий создает необходимые файлы для правильной настройки среды.

ИНСТРУКЦИИ

Выполните следующие задачи на хосте **serverb**, чтобы завершить упражнение.

- Сгенерируйте ключи **SSH** для пользователя **student** на **serverb**. Не защищайте закрытый ключ парольной фразой.
- На сервере настройте пользователя **student**, чтобы он принимал входы в систему, **аутентифицированные парой ключей SSH**, которую вы создали для пользователя **student** на **serverb**. Пользователь **student** на **serverb** должен иметь возможность входить в систему с помощью **SSH** без ввода пароля.
- На **serverb** настройте службу **sshd**, чтобы пользователи не могли входить в систему как **root** через **SSH**.
- На **serverb** настройте службу **sshd**, чтобы пользователи не могли использовать свои пароли для входа в систему. Пользователи по-прежнему должны иметь возможность аутентифицировать вход с помощью пары ключей **SSH**.
- Создайте **tar**-архив с именем **/tmp/log.tar**, содержащий содержимое **/var/log** на **serverb**. Удаленно передайте **tar**-архив в каталог **/tmp** на сервере, аутентифицируясь как пользователь **student**, используя закрытый ключ пользователя **student** из пары ключей **SSH**.
- Настройте службу **rsyslog** на **serverb**, чтобы все полученные сообщения с уровнем приоритета отладки или выше регистрировались в файле **/var/log/grading-debug**. Эта конфигурация должна быть установлена в файле **/etc/rsyslog.d/grading-debug.conf**, который вам необходимо создать.

- Установите пакет **zsh**, доступный в репозитории **BaseOS**, на **serverb**.
 - Включите **stream** модули по умолчанию для модуля **python36** и установите все предоставленные пакеты из этого потока на **serverb**.
 - Установите часовой пояс сервера **serverb** на **Asia/Kolkata**.
- 1.** Сгенерируйте ключи SSH для пользователя **student** на сервере **serverb**. Не защищайте закрытый ключ парольной фразой.
- 1.1.** С рабочей станции **workstation**, откройте сеанс SSH на **serverb** как пользователь **student**.
- ```
[student@workstation ~]$ ssh student@serverb
...output omitted...
[student@serverb ~]$
```
- 1.2.** Используйте команду **ssh-keygen**, чтобы сгенерировать ключи SSH для пользователя **student**. Файлы закрытого и открытого ключей должны называться **/home/student/.ssh/review3\_key** и **/home/student/.ssh/review3\_key.pub** соответственно.
- ```
[student@serverb ~]$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/student/.ssh/id_rsa): /home/
student/.ssh/review3_key
Enter passphrase (empty for no passphrase): Enter
Enter same passphrase again: Enter
Your identification has been saved in /home/student/.ssh/review3_key.
Your public key has been saved in /home/student/.ssh/review3_key.pub.
The key fingerprint is:
SHA256:Uqefehw+vRfm94fQZDoz/6IfNYSLK/0piQ4n6lrlKIbY student@serverb.lab.example.com
The key's randomart image is:
+--- [RSA 2048] ---+
| |
| . |
| . . . |
| . o . = |
| . S . * ..|
| . . .B +..|
| .o . o . =o+ 0.0 |
| + = . + ..X o * .o|
| Eoo .o.+.+o=.+=|
+--- [SHA256] ---+
```
- 2.** На сервере **servera**, настройте пользователя **student**, чтобы он принимал вход в систему, аутентифицированные парой ключей SSH, которую вы создали для пользователя **student** на **serverb**. Пользователь **student** на **serverb** должен иметь возможность входить в систему с помощью SSH без ввода пароля.
- 2.1.** Используйте команду **ssh-copy-id** для экспорта открытого ключа **/home/student/.ssh/review3_key.pub** с сервера **servera** на сервер **serverb**.

```
[student@serverb ~]$ ssh-copy-id -i .ssh/review3_key.pub student@servera  
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: ".ssh/review3.pub"  
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter  
out any that are already installed  
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted  
now it is to install the new keys  
student@servera's password: student
```

Number of key(s) added: 1

Now try logging into the machine, with: "ssh 'student@servera'"
and check to make sure that only the key(s) you wanted were added.

- 2.2.** Используйте команду **ssh**, чтобы подтвердить, что вы можете войти на сервер с **serverb** как пользователь **student**, используя закрытый ключ **SSH /home/student/.ssh/review3_key** без запроса пароля.

```
[student@serverb ~]$ ssh -i .ssh/review3_key student@servera  
...output omitted...  
[student@servera ~]$
```

- 2.3.** Выйти из сервера **servera**.

```
[student@servera ~]$ exit  
logout  
Connection to servera closed.  
[student@serverb ~]$
```

- 3.** На **serverb** настройте службу **sshd**, чтобы пользователи не могли входить в систему как пользователь **root** с помощью **SSH**.

- 3.1.** Установите для параметра **PermitRootLogin** значение **no** в файле **/etc/ssh/sshd_config**. Вы можете использовать команду **sudo vim /etc/ssh/sshd_config** для редактирования файла конфигурации.

- 3.2.** Перезагрузите службу **sshd**.

```
[student@serverb ~]$ sudo systemctl reload sshd.service
```

- 4.** На **serverb** настройте службу **sshd**, чтобы пользователи не могли использовать свои пароли для входа в систему. Пользователи по-прежнему должны иметь возможность аутентифицировать логины, используя свой закрытый ключ из пары ключей **SSH**.

- 4.1.** Установите для параметра **PasswordAuthentication** значение **no** в файле **/etc/ssh/sshd_config**. Вы можете использовать команду **sudo vim /etc/ssh/sshd_config** для редактирования файла конфигурации.

4.2. Используйте команду **sudo systemctl**, чтобы перезагрузить службу **sshd**.

```
[student@serverb ~]$ sudo systemctl reload sshd.service
```

5. Создайте **tar**-архив с именем **/tmp/log.tar**, включающий содержимое **/var/log** на **serverb**. Удаленно перенесите **tar**-архив в каталог **/tmp** на сервер **servera**, аутентифицируясь как пользователь **student**, используя **/home/student/.ssh/review3_key** в качестве закрытого ключа пользователя **student** из пары ключей **SSH** для аутентификации.

5.1. Используйте команду **sudo tar** для создания архива с именем **/tmp/log.tar** в качестве суперпользователя, включающего содержимое файла **/var/log**.

```
[student@serverb ~]$ sudo tar -cvf /tmp/log.tar /var/log  
[sudo] password for student:  
...output omitted...
```

5.2. Используйте команду **scp** для удаленной передачи файла архива **/tmp/log.tar** в каталог **/tmp** на сервер **servera**. Укажите **/home/student/.ssh/review3_key** в качестве закрытого ключа пары ключей **SSH**.

```
[student@serverb ~]$ scp -i .ssh/review3_key /tmp/log.tar student@servera:/tmp  
log.tar 100% 14MB 57.4MB/s 00:00
```

6. Настройте службу **rsyslog** на **serverb**, чтобы все полученные сообщения с уровнем приоритета отладки или выше регистрировались в файле **/var/log/grading-debug**. Эта конфигурация должна быть установлена в файле **/etc/rsyslog.d/grading-debug.conf**, который вы должны создать.

6.1. Создайте файл **/etc/rsyslog.d/grading-debug.conf** со следующим содержимым. Вы можете использовать **sudo vim /etc/rsyslog.d/grading-debug.conf** для создания файла.

```
*.debug /var/log/grading-debug
```

6.2. Используйте команду **sudo systemctl**, чтобы перезапустить службу **rsyslog**.

```
[student@serverb ~]$ sudo systemctl restart rsyslog.service
```

6.3. Используйте команду **logger** для создания сообщения журнала **Debug Testing**, имеющего приоритет отладки (**debug**).

```
[student@serverb ~]$ logger -p debug Debug Testing
```

- 6.4.** Убедитесь, что сообщение журнала «**Debug Testing**» сохранено в файле **/var/log/grading-debug**.

```
[student@serverb ~]$ sudo tail /var/log/grading-debug  
...output omitted...  
Mar 12 09:55:23 serverb student[32383]: Debug Testing
```

- 7.** Используйте команду **sudo yum** для установки пакета **zsh**, доступного в репозитории **BaseOS**, на **serverb**.

```
[student@serverb ~]$ sudo yum install zsh  
...output omitted...  
Is this ok [y/N]: y  
...output omitted...  
Installed:  
  zsh-5.5.1-6.el8.x86_64  
Complete!
```

- 8.** Используйте команду **yum**, чтобы включить stream модуля по умолчанию для модуля **python36** и установить все предоставленные пакеты из этого потока на **serverb**.

```
[student@serverb ~]$ sudo yum module install python36  
...output omitted...  
Is this ok [y/N]: y  
...output omitted...  
Installed:  
  python36-3.6.6-18.module+el8+2339+1a6691f8.x86_64          python3-  
  pip-9.0.3-13.el8.noarch  
  
Complete!
```

- 9.** Установите часовой пояс **serverb** на **Asia/Kolkata**.

- 9.1.** Используйте команду **sudo timedatectl**, чтобы установить часовой пояс **serverb** на **Asia/Kolkata**.

```
[student@serverb ~]$ sudo timedatectl set-timezone Asia/Kolkata
```

- 9.2.** Выйти из сервера **serverb**.

```
[student@serverb ~]$ exit  
logout  
Connection to serverb closed.
```

```
[student@workstation ~]$
```

Оценка

На рабочей станции **workstation** запустите команду **lab rhcsa-rh124-review3 grade**, чтобы подтвердить успех выполнения упражнения.

```
[student@workstation ~]$ lab rhcsa-rh124-review3 grade
```

Завершение

На рабочей станции запустите скрипт **lab rhcsa-rh124-review3 finish**, чтобы завершить всесторонний обзор. Сценарий удаляет файлы и каталоги, созданные в начале всестороннего обзора, и обеспечивает чистоту среды на хосте **serverb**.

```
[student@workstation ~]$ lab rhcsa-rh124-review3 finish
```

На этом всеобъемлющий обзор завершен.

ЛАБОРАТОРНАЯ РАБОТА

УПРАВЛЕНИЕ СЕТЬЮ

В этом обзоре вы настроите и протестируете сетевое подключение.

В РЕЗУЛЬТАТЕ

Вы должны быть способны:

- Настройте параметры сети.
- Проверьте подключение к сети.
- Установите статическое имя хоста для системы.
- Используйте локально разрешаемые канонические имена хостов для подключения к системам.

ПРЕЖДЕ ЧЕМ ВЫ НАЧНЕТЕ

Войдите на рабочую станцию **workstation** как пользователь **student**, используя в качестве пароля слово **student**.

На рабочей станции **workstation** выполните скрипт **lab rhcsa-rh124-review4 start**, чтобы начать всесторонний обзор. Этот сценарий создает необходимые файлы для правильной настройки среды.

```
[student@workstation ~]$ lab rhcsa-rh124-review4 start
```

ИНСТРУКЦИИ

Выполните следующие задачи на сервере **serverb**, чтобы выполнить упражнение.

Определите имя интерфейса **Ethernet** и его активный профиль подключения на сервере **serverb**.

- На **serverb** создайте новый профиль подключения, называемый **static**, для доступного интерфейса **Ethernet**, который статически устанавливает параметры сети и не использует **DHCP**. Это должно быть основано на настройках, указанных в следующей таблице.

IPv4 address	172.25.250.111
Netmask	255.255.255.0
Gateway	172.25.250.254
DNS server	172.25.250.254

Настройте интерфейс **Ethernet** сервера на использование обновленных сетевых настроек, показанных в таблице выше.

- Убедитесь, что имя хоста **serverb** статически установлено как **server-review4.lab4.example.com**.
- На **serverb** установите **client-review4** в качестве канонического имени хоста для **IPv4**-адреса **172.25.250.10** хоста **servera.lab.example.com**.
- Настройте дополнительный **IPv4**-адрес **172.25.250.211** с сетевой маской **255.255.255.0** на том же интерфейсе **serverb**, который имеет существующие статические сетевые настройки. Не удаляйте существующий адрес **IPv4**. Убедитесь, что **serverb** отвечает на все адреса, соединения, которое вы статически настроили на его интерфейсе, активно.
- На **serverb** восстановите исходные параметры сети, активировав исходное сетевое соединение и деактивировав статическое сетевое соединение, созданное вручную.

Оценка

На рабочей станции запустите команду **lab rhcsa-rh124-review4 grade**, чтобы подтвердить успех этого упражнения.

```
[student@workstation ~]$ lab rhcsa-rh124-review4 grade
```

Завершение

На рабочей станции запустите скрипт **lab rhcsa-rh124-review4 finish**, чтобы завершить всесторонний обзор. Этот сценарий удаляет файлы и каталоги, созданные в начале всестороннего обзора, и обеспечивает чистоту среды на **serverb**.

```
student@workstation ~]$ lab rhcsa-rh124-review4 finish
```

На этом всеобъемлющий обзор завершен.

РЕШЕНИЕ

УПРАВЛЕНИЕ СЕТЬЮ

В этом обзоре вы настроите и протестируете сетевое подключение.

В РЕЗУЛЬТАТЕ

Вы должны быть способны:

- Настройте параметры сети.
- Проверьте подключение к сети.
- Установите статическое имя хоста для системы.
- Используйте локально разрешаемые канонические имена хостов для подключения к системам.

ПРЕЖДЕ ЧЕМ ВЫ НАЧНЕТЕ

Войдите на рабочую станцию **workstation** как пользователь **student**, используя в качестве пароля слово **student**.

На рабочей станции **workstation** выполните скрипт **lab rhcsa-rh124-review4 start**, чтобы начать всесторонний обзор. Этот сценарий создает необходимые файлы для правильной настройки среды.

```
[student@workstation ~]$ lab rhcsa-rh124-review4 start
```

ИНСТРУКЦИИ

Выполните следующие задачи на сервере **serverb**, чтобы выполнить упражнение.

Определите имя интерфейса **Ethernet** и его активный профиль подключения на сервере **serverb**.

- На **serverb** создайте новый профиль подключения, называемый **static**, для доступного интерфейса **Ethernet**, который статически устанавливает параметры сети и не использует **DHCP**. Это должно быть основано на настройках, указанных в следующей таблице.

IPv4 address	172.25.250.111
Netmask	255.255.255.0
Gateway	172.25.250.254
DNS server	172.25.250.254

Настройте интерфейс **Ethernet** сервера на использование обновленных сетевых настроек, показанных в таблице выше.

- Убедитесь, что имя хоста **serverb** статически установлено как **server-review4.lab4.example.com**.
- На **serverb** установите **client-review4** в качестве канонического имени хоста для **IPv4**-адреса **172.25.250.10** хоста **servera.lab.example.com**.
- Настройте дополнительный **IPv4**-адрес **172.25.250.211** с сетевой маской **255.255.255.0** на том же интерфейсе **serverb**, который имеет существующие статические сетевые настройки. Не удаляйте существующий адрес **IPv4**. Убедитесь, что **serverb** отвечает на все адреса, соединения, которое вы статически настроили на его интерфейсе, активно.
- На **serverb** восстановите исходные параметры сети, активировав исходное сетевое соединение и деактивировав статическое сетевое соединение, созданное вручную.

1. Войдите в "физическую" консоль **serverb** как пользователь **student**:

На веб-странице, которая управляет вашей лабораторной средой, нажмите кнопку **ОТКРЫТЬ КОНСОЛЬ (OPEN CONSOLE)** для **serverb**. В вашем браузере откроется вкладка с сеансом консоли для **serverb**. Войдите в систему как пользователь **student** в ответ на приглашение.

Рекомендуется вносить сетевые изменения с консоли сервера, будь то локально, с помощью специального оборудования для удаленного доступа к консоли или удаленно доступной последовательной консоли. Если вы используете **ssh** для настройки сетевых параметров, вы можете легко повесить сеанс или заблокировать удаленный доступ к серверу при изменении профилей подключения.

2. Определите имя интерфейса **Ethernet** на сервере **serverb** и имя профиля подключения, который он использует.

```
[student@serverb ~]$ nmcli device status
```

В этом примере **ens3** - это имя интерфейса **Ethernet** в системе. Это интерфейс, для которого вы создадите статический профиль подключения.

Имя сетевого интерфейса и первоначального профиля подключения могут отличаться на вашем сервере **serverb**. В этом случае используйте имя, показываемое вашей системой на остальных этапах этого решения, вместо имени, используемого в этом решении.

3. На **serverb** создайте новый профиль подключения с именем **static** для доступного интерфейса **Ethernet**. Установите параметры сети статически, чтобы **не использовать DHCP**. Настройте параметры на основе следующей таблицы:

IPv4 address	172.25.250.111
Netmask	255.255.255.0
Gateway	172.25.250.254
DNS server	172.25.250.254

Интерфейс Ethernet сервера **serverb** должен использовать обновленные сетевые настройки, как указано в предыдущей таблице.

- 3.1.** Используйте **nmcli** для создания статического соединения с заданными сетевыми настройками.

```
[student@serverb ~]$ sudo nmcli connection add con-name static type ethernet \
ifname ens3 ipv4.addresses '172.25.250.111/24' ipv4.gateway '172.25.250.254' \
ipv4.dns '172.25.250.254' ipv4.method manual
[sudo] password for student:
Connection 'static' (ac8620e6-b77e-499f-9931-118b8b015807) successfully added.
```

- 3.2.** Используйте команду **nmcli**, чтобы активировать новые настройки подключения.

```
[student@serverb ~]$ sudo nmcli connection up static
```

- 4.** Используйте команду **hostnamectl** для статической установки имени хоста **serverb** на **serverreview4.lab4.example.com** и проверьте новое имя хоста с помощью команды **hostname**.

```
[student@serverb ~]$ sudo hostnamectl set-hostname server-review4.lab4.example.com
[sudo] password for student:
[student@serverb ~]$ hostname
server-review4.lab4.example.com
```

- 5.** На **serverb** отредактируйте файл **/etc/hosts**, чтобы установить **client-review4** в качестве канонического имени хоста для IPv4-адреса **172.25.250.10** для хоста **servera.lab.example.com**.

- 5.1.** Отредактируйте файл **/etc/hosts**, чтобы добавить **client-review4** в качестве имени IPv4-адреса **172.25.250.10**. Вы можете использовать команду **sudo vim /etc/hosts** для редактирования файла.

```
172.25.250.10 servera.lab.example.com servera client-review4
```

- 5.2.** Используйте команду **ping**, чтобы убедиться, что вы можете достичь **172.25.250.10**, используя каноническое имя хоста **client-review4**.

```
[student@serverb ~]$ ping -c2 client-review4
```

```
PING servera.lab.example.com (172.25.250.10) 56(84) bytes of data.  
64 bytes from servera.lab.example.com (172.25.250.10): icmp_seq=1 ttl=64  
time=0.259 ms  
64 bytes from servera.lab.example.com (172.25.250.10): icmp_seq=2 ttl=64  
time=0.391 ms  
  
--- servera.lab.example.com ping statistics ---  
2 packets transmitted, 2 received, 0% packet loss, time 33ms  
rtt min/avg/max/mdev = 0.259/0.325/0.391/0.066 ms
```

6. Настройте профиль **static** сетевого подключения, чтобы настроить дополнительный **IPv4**-адрес **172.25.250.211** с сетевой маской **255.255.255.0** на том же интерфейсе **serverb**, который имеет существующие статические сетевые настройки. Не удаляйте существующий адрес **IPv4**. Убедитесь, что **serverb** отвечает на все адреса, когда соединение, которое вы статически настроили на его интерфейсе, активно.

6.1. Используйте команду **nmcli**, чтобы добавить новый IP-адрес.

```
[student@serverb ~]$ sudo nmcli connection modify static \  
+ipv4.addresses '172.25.250.211/24'
```

6.2. Используйте команду **nmcli**, чтобы активировать новый IP-адрес.

```
[student@serverb ~]$ sudo nmcli connection up static  
...output omitted...
```

6.3. С рабочей станции **workstation**, используйте команду **ping**, чтобы убедиться, что вы можете связаться с хостом с **IPv4**-адресом **172.25.250.211**.

```
[student@workstation ~]$ ping -c2 172.25.250.211  
PING 172.25.250.211 (172.25.250.211) 56(84) bytes of data.  
64 bytes from 172.25.250.211: icmp_seq=1 ttl=64 time=0.246 ms  
64 bytes from 172.25.250.211: icmp_seq=2 ttl=64 time=0.296 ms  
  
--- 172.25.250.211 ping statistics ---  
2 packets transmitted, 2 received, 0% packet loss, time 50ms  
rtt min/avg/max/mdev = 0.246/0.271/0.296/0.025 ms
```

7. На **serverb** восстановите исходные настройки сети, активировав исходное сетевое соединение.

7.1. Вернитесь в консоль и используйте команду **nmcli**, чтобы активировать исходный сетевой профиль.

```
[student@serverb ~]$ sudo nmcli connection up "Wired connection 1"  
...output omitted...
```

Имя исходного профиля подключения может отличаться на вашем **serverb**. Если это так, замените имя, показанное в этом решении, на имя, которое вы действительно видели в своей системе. Если вы забыли, вы сможете найти его с помощью команды **nmcli connection show**.

7.2. С рабочей станции откройте сеанс **SSH** на **serverb** как пользователь **student**, чтобы убедиться, что исходные сетевые настройки успешно активированы.

```
[student@workstation ~]$ ssh serverb  
...output omitted...  
[student@server-review4 ~]$
```

7.3. Выйдите с **serverb** и выйдите из всех терминалов, кроме одного, на рабочей станции **workstation**.

```
[student@server-review4 ~]$ exit  
logout  
Connection to serverb closed.  
[student@workstation ~]$
```

Оценка

На рабочей станции запустите команду **lab rhcsa-rh124-review4 grade**, чтобы подтвердить успех этого упражнения.

```
[student@workstation ~]$ lab rhcsa-rh124-review4 grade
```

Завершение

На рабочей станции запустите скрипт **lab rhcsa-rh124-review4 finish**, чтобы завершить всесторонний обзор. Этот сценарий удаляет файлы и каталоги, созданные в начале всестороннего обзора, и обеспечивает чистоту среды на **serverb**.

```
student@workstation ~]$ lab rhcsa-rh124-review4 finish
```

На этом всеобъемлющий обзор завершен.

ЛАБОРАТОРНАЯ РАБОТА

МОНТАЖ ФАЙЛОВЫХ СИСТЕМ И ПОИСК ФАЙЛОВ

В этом обзоре вы смонтируете файловую систему и найдете файлы по разным критериям.

В РЕЗУЛЬТАТЕ

Вы должны быть способны:

- Смонтировать существующую файловую систему.
- Находить файлы по их имени, разрешениям и размеру.

ПРЕЖДЕ ЧЕМ ВЫ НАЧНЕТЕ

Войдите на рабочую станцию **workstation** как пользователь **student**, используя в качестве пароля слово **student**.

На рабочей станции **workstation** выполните скрипт **lab rhcsa-rh124-review5 start**, чтобы начать всесторонний обзор. Этот сценарий создает необходимую файловую систему, учетные записи пользователей и групповые учетные записи.

```
[student@workstation ~]$ lab rhcsa-rh124-review5 start
```

ИНСТРУКЦИИ

Выполните следующие задачи на **serverb**, чтобы выполнить упражнение.

- На **serverb** существует блочное устройство, содержащее файловую систему **XFS**, но оно еще не смонтировано. Определите блочное устройство и смонтируйте его в каталог **/review5-disk**. При необходимости создайте каталог **/review5-disk**.
- На **serverb** найдите файл с именем **review5-path**. Создайте файл с именем **/review5-disk/review5.txt**, содержащий единственную строку, абсолютный путь к файлу **review5**.
- На **serverb** найдите все файлы, в которых в качестве пользователя и группы указаны **contractor1** и **contractor** соответственно. Файлы также должны иметь восьмеричное разрешение **640**. Сохраните список этих файлов в **/review5-disk/review5-perms.txt**.
- На **serverb** найдите все файлы размером 100 байт. Сохраните абсолютные пути к этим файлам в **/review5-disk/review5-size.txt**.

Оценка

На рабочей станции запустите команду **lab rhcsa-rh124-review5 grade**, чтобы подтвердить успех выполнения этого упражнения.

```
[student@workstation ~]$ lab rhcsa-rh124-review5 grade
```

Завершение

На рабочей станции **workstation**, запустите скрипт **lab rhcsa-rh124-review5 finish**, чтобы завершить всесторонний обзор. Этот сценарий удаляет файловую систему, учетные записи пользователей и групповые учетные записи, созданные во время начала комплексной проверки, и обеспечивает чистоту среды на сервере **serverb**.

```
[student@workstation ~]$ lab rhcsa-rh124-review5 finish
```

На этом всеобъемлющий обзор завершен.

РЕШЕНИЕ

МОНТАЖ ФАЙЛОВЫХ СИСТЕМ И ПОИСК ФАЙЛОВ

В этом обзоре вы смонтируете файловую систему и найдете файлы по разным критериям.

В РЕЗУЛЬТАТЕ

Вы должны быть способны:

- Смонтировать существующую файловую систему.
- Находить файлы по их имени, разрешениям и размеру.

ПРЕЖДЕ ЧЕМ ВЫ НАЧНЕТЕ

Войдите на рабочую станцию **workstation** как пользователь **student**, используя в качестве пароля слово **student**.

На рабочей станции **workstation** выполните скрипт **lab rhcsa-rh124-review5 start**, чтобы начать всесторонний обзор. Этот сценарий создает необходимую файловую систему, учетные записи пользователей и групповые учетные записи.

```
[student@workstation ~]$ lab rhcsa-rh124-review5 start
```

ИНСТРУКЦИИ

Выполните следующие задачи на **serverb**, чтобы выполнить упражнение.

- На **serverb** существует блочное устройство, содержащее файловую систему **XFS**, но оно еще не смонтировано. Определите блочное устройство и смонтируйте его в каталог **/review5-disk**. При необходимости создайте каталог **/review5-disk**.
 - На **serverb** найдите файл с именем **review5-path**. Создайте файл с именем **/review5-disk/review5.txt**, содержащий единственную строку, абсолютный путь к файлу **review5**.
 - На **serverb** найдите все файлы, в которых в качестве пользователя и группы указаны **contractor1** и **contractor** соответственно. Файлы также должны иметь восьмеричное разрешение **640**. Сохраните список этих файлов в **/review5-disk/review5-perms.txt**.
 - На **serverb** найдите все файлы размером 100 байт. Сохраните абсолютные пути к этим файлам в **/review5-disk/review5-size.txt**.
1. На **serverb** смонтируйте бездействующее блочное устройство, содержащее файловую систему **XFS**, в каталог **/review5-disk**.
 - 1.1. С рабочей станции **workstation**, откройте сеанс **SSH** на **serverb** как пользователь **student**.

```
[student@workstation ~]$ ssh student@serverb  
...output omitted...  
[student@serverb ~]$
```

- 1.2.** Используйте команду **lsblk -fs**, чтобы определить бездействующее блочное устройство, содержащее файловую систему **XFS**.

```
[student@serverb ~]$ lsblk -fs  
NAME FSTYPE LABEL UUID MOUNTPOINT  
...output omitted...  
vdb1 xfs          3d97c5ef-23e7-4c1c-a9be-d5c475b3d0d5  
└─vdb  
...output omitted...
```

Из предыдущего вывода обратите внимание, что блочное устройство **vdb1** содержит файловую систему **XFS**, которая не смонтирована ни в одном каталоге.

- 1.3.** Используйте команду **sudo mkdir**, чтобы создать каталог **/review5-disk** в качестве суперпользователя. Когда команда **sudo** запрашивает пароль, введите пароль **student**.

```
[student@serverb ~]$ sudo mkdir /review5-disk  
[sudo] password for student: student
```

- 1.4.** Используйте команду **sudo mount**, чтобы смонтировать блочное устройство **vdb1** используя точку монтирования каталог **/review5-disk** в качестве суперпользователя.

```
[student@serverb ~]$ sudo mount /dev/vdb1 /review5-disk
```

- 1.5.** Убедитесь, что блочное устройство **vdb1** успешно смонтировано в каталоге **/review5-disk**.

```
[student@serverb ~]$ df -Th
```

- 2.** На **serverb** найдите файл с именем **review5-path**. Запишите его абсолютный путь в текстовый файл **/review5-disk/review5-path.txt**.

- 2.1.** Используйте команду **find**, чтобы найти файл с именем **review5-path**. Перенаправьте все ошибки команды **find** в **/dev/null**. Это перенаправление позволяет отбросить любые ошибки в выводе команды **find**.

```
[student@serverb ~]$ find / -iname review5-path 2>/dev/null
```

```
/var/tmp/review5-path
```

Обратите внимание на абсолютный путь к файлу **review5-path** из предыдущего вывода.

- 2.2.** Создайте текстовый файл **/review5-disk/review5-path.txt**. Запишите абсолютный путь к файлу **review5-path**, как определено на предыдущем шаге, в текстовом файле **/review5-disk/review5-path.txt**. Вы можете использовать команду **sudo vim /review5-disk/review5-path.txt** для создания текстового файла. Введите **:wq!** из командного режима в **vim**, чтобы сохранить изменения и выйти из файла. Следующий вывод показывает содержимое текстового файла **/review5-disk/review5-path.txt**.

```
/var/tmp/review5-path
```

- 3.** На **serverb** найдите все файлы, в которых в качестве пользователя и группы указаны **contractor1** и **contractor** соответственно. Файлы также должны иметь восьмеричное разрешение **640**. Запишите абсолютные пути ко всем этим файлам в текстовом файле **/review5-disk/review5-perms.txt**.

- 3.1.** Используйте параметры **-user**, **-group**, **-perm** с командой **find**, чтобы найти все файлы, у которых есть права владельца, группа-владелец и восьмеричные права доступа **contractor1**, **contractor** и **640** соответственно. Перенаправьте все ошибки команды **find** в **/dev/null**.

```
[student@serverb ~]$ find / -user contractor1 \  
> -group contractor \  
> -perm 640 2>/dev/null \  
/usr/share/review5-perms
```

Обратите внимание на абсолютный путь к файлу **review5-perms** из предыдущего вывода. Файл **/usr/share/review5-perms** единственный, который соответствует критериям предыдущей команды **find**.

- 3.2.** Создайте текстовый файл **/review5-disk/review5-perms.txt**. Запишите абсолютный путь к единственному файлу (**review5-perms**), у которого есть пользователь-владелец, группа-владелец и восьмеричные разрешения **Contractor1**, **Contractor** и **640**, соответственно, как определено на предыдущем шаге, в текстовом файле **/review5-disk/review5-perms.txt**. Вы можете использовать команду **sudo vim /review5-disk/review5-perms.txt** для создания текстового файла. Введите **:wq!** из командного режима в **vim**, чтобы сохранить изменения и выйти из файла. Следующий вывод показывает содержимое текстового файла **/review5-disk/review5-perms.txt**

```
/usr/share/review5-perms
```

4. На **serverb** найдите все файлы **размером 100 байт**. Запишите абсолютные пути ко всем этим файлам в **/review5-disk/review5-size.txt**.

- 4.1. Используйте параметр **-size** с командой **find**, чтобы найти все файлы размером **100 байт**. Перенаправьте все ошибки команды **find** в **/dev/null**.

```
[student@serverb ~]$ find / -size 100c 2>/dev/null
/dev/disk
/run/initramfs
/etc/lvm
/etc/audit
/etc/sos.conf
/usr/lib/python3.6/site-packages/dnf/conf
/usr/lib/python3.6/site-packages/ptyprocess
/usr/share/licenses/ethtool/LICENSE
/usr/share/doc/libuser
/usr/share/doc/python3-cryptography/docs/x509
/usr/share/doc/python3-jinja2/ext
/usr/share/doc/plymouth/AUTHORS
/usr/share/vim/vim80/macros/maze/main.aap
/usr/libexec/plymouth
/opt/review5-size
```

Предыдущий вывод может отличаться в вашей системе в зависимости от количества файлов размером **100** байт в вашей системе. Обратите внимание на абсолютные пути ко всем файлам из предыдущего вывода.

- 4.2. Создайте текстовый файл **/review5-disk/review5-size.txt**. Запишите абсолютные пути ко всем файлам размером **100** байт, определенным на предыдущем шаге, в текстовом файле **/review5-disk/review5-size.txt**. Вы можете использовать команду **sudo vim /review5-disk/review5-size.txt** для создания текстового файла. Введите **:wq!** из командного режима в **vim**, чтобы сохранить изменения и выйти из файла. Текстовый файл **/review5-disk/review5-size.txt** должен содержать абсолютный путь к файлу размера **review5** среди других путей.

```
...output omitted...
/opt/review5-size
...output omitted...
```

- 4.3. Выйти из сервера **serverb**.

```
[student@serverb ~]$ exit
logout
Connection to serverb closed.
[student@workstation ~]$
```

Оценка

На рабочей станции запустите команду **lab rhcsa-rh124-review5 grade**, чтобы подтвердить успех выполнения этого упражнения.

```
[student@workstation ~]$ lab rhcsa-rh124-review5 grade
```

Завершение

На рабочей станции **workstation**, запустите скрипт **lab rhcsa-rh124-review5 finish**, чтобы завершить всесторонний обзор. Этот сценарий удаляет файловую систему, учетные записи пользователей и групповые учетные записи, созданные во время начала комплексной проверки, и обеспечивает чистоту среды на сервере **serverB**.

```
[student@workstation ~]$ lab rhcsa-rh124-review5 finish
```

На этом всеобъемлющий обзор завершен.