

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
Факультет прикладной математики и информатики  
Кафедра технологий программирования

**Лабораторная работа №1**

По курсу “Программирование мобильных и встраиваемых систем”

**Базовые команды для работы с ОС LINUX для мобильных и  
встраиваемых платформ**

Методические указания по выполнению лабораторной работы

**Подготовила:** Давидовская М.И.,  
Ст. преподаватель кафедры ТП

Минск, 2022 г.

# Содержание

Цель работы.....	3
Задачи работы.....	3
Краткие теоретические сведения.....	3
Задания.....	4
Методические указания.....	4
Критерии оценивания.....	4
Содержание отчета.....	5
Задание 1. Развертывание виртуальных машин под управлением ОС Raspberry PI и ОС Ubuntu.....	5
Задание 2. Настройка ssh-соединений между хостовой системой и гостевыми машинами.....	6
Задание 3. Использование команды script для протоколирования команд.....	6
Задание 4. Безопасное копирование файлов и публикация в репозиторий.....	7
Задание 5. Управление файлами и папками.....	8
Задание 6. Управление пользователями и права доступа.....	10
Задание 7. Базовые команды для управления процессами.....	10
Задание 8. Сжатие и архивация файлов.....	11
Задание 9. Синхронизация в Linux с помощью утилиты rsync.....	12
Задание 10. Подключение по ssh к контейнеру Docker.....	13
Контрольные вопросы.....	13
Приложение 1.....	15
1. Синтаксис основных команд.....	15
2. Учетные записи в ОС Linux.....	22
3. Управление пользователями и группами.....	26
3.1 Добавление учетной записи пользователя.....	26
3.2 Установка пароля пользователя.....	27
3.3 Изменение информации о времени действия пароля пользователя.....	27
3.4 Изменение личной информации о пользователе.....	27
3.5 Модификация учетной записи пользователя.....	27
3.6 Добавление группы.....	28
3.7 Удаление учетной записи пользователя и группы.....	28
4. Управление файлами, каталогами и права доступа.....	28
4.1. Создание и просмотр файлов.....	28
4.2 Создание и просмотр каталогов.....	29
4.3 Копирование и перемещение файлов.....	29
4.4 Работа со ссылками.....	29
4.5 Управление разрешениями на доступ к файлу.....	30
4.6 Удаление файлов и каталогов.....	32
5. Архивация файлов и каталогов в ОС Linux.....	32
5.1 Утилиты zip и unzip.....	32
5.2 Утилиты gzip и gunzip.....	33
5.3 Утилита tar.....	34

## Цель работы

Ознакомление с операционной системой ОС Unix/Linux и основными командами работы с файлами и каталогами, правами доступа и процессами ОС Unix/Linux.

## Задачи работы

- 1.1. Изучить процедуры входа в систему, выполнения основных команд, получения подсказки, создания и редактирования текстовых файлов.
- 1.2. Изучить основные принципы работы по протоколу ssh.
- 1.3. Изучить основные команды для создания каталогов, файлов, жестких ссылок и символических ссылок, включая архивацию и синхронизацию данных.
- 1.4. Изучить команды создания управления учетными записями, команды изменения прав доступа и владения `chmod` и `chown`.
- 1.5. Изучить работу и ключи команд `ps`, `pidof`, `fuser` — получения сведений о процессах в операционной системе.

## Краткие теоретические сведения

Лабораторная работа выполняется в среде Ubuntu и Raspberry PI ОС (ранее Raspbian ОС), исполняемых в виртуальных машинах под управлением Virtualbox. Доступ к Ubuntu и Raspberry PI ОС осуществляется с рабочего места, функционирующего в среде ОС Windows или macOS. При выполнении лабораторной работы Вы получаете доступ к учетной записи администратора и создаете учетные записи пользователей для выполнения лабораторной работы.

По умолчанию приглашением в Unix/Linux является символ '#' или '\$', в ответ на которое Вы можете вводить команды. Если в примере приводится команда и в качестве символа приглашения указан символ '#', то это означает, что это пример командной строки для суперпользователя `root`, и команду мы вводим под учетной записью `root`. Если же в примере указан символ '\$', то это означает, что команда вводится по учетной записью обычного пользователя. Левее символа командной строки в квадратных скобках указана Ваша учетная запись и имя устройства с ОС Linux, а также текущий каталог.

В сеансе работы с Linux Вашим текущим (домашним) каталогом является каталог: `/home/имя`, где `имя` — Ваше сетевое имя пользователя. К этому каталогу вам предоставлен полный доступ, т.е. вы имеете права чтения, записи и выполнения. Вы не имеете права записи к каталогам, не являющимся подкаталогами вашего домашнего каталога, если их владельцы (или системный администратор) не дали вам соответствующих прав. Не забывайте, что в Unix/Linux символ "слэш" — разделитель имен каталогов наклонен вправо: `'/'`!

Для получения справки по командам и программам служат инструкции `man` и `info`, а также другие команды. Некоторые разделы справки даны на английском языке. Краткое описание применяемых в лабораторной работе команд и программ на русском языке представлено в Приложение 1, стр. 15.

Для окончания сеанса работы с командной оболочкой Linux используем команды `exit` или `logout`.

## Задания

### Методические указания

- 1.1. Вводите команды ОС только при наличии приглашения оболочки.
- 1.2. В используемой оболочке имеется буфер команд (история команд), перемещение по которому производится стрелками управления курсором (вверх и вниз). Для редактирования команды из буфера используются клавиши `<-` и `>`, `<BackSpace>` и `<Delete>`. Кроме того для просмотра истории команд используется команда `history`.
- 1.3. Для окончания работы инструкций `man` и `info` используйте символ `q`.
- 1.4. Для протоколирования результатов выполнения команд используйте команду `script`, которая позволяет создать протокол работы пользователя в командной оболочке. Примеры использования команды `script` описаны в руководстве [script\\_command\\_guide.pdf](#). Требуется логгировать не только сами команды, но временные метки. Как это сделать, рассмотрено в 4-м примере указанного выше руководства.
- 1.5. Все результаты лабораторной работы должны быть опубликованы в git-репозитории, ссылка на который доступна в курсе «[Программирование мобильных и встраиваемых систем](#)».
- 1.6. Возможно открыть новый терминал в пределах той же сессии, используя клавиши `<Alt><F2>` – `<F6>`. В виртуальной машине переключение может не работать.
- 1.7. При работе в файловом менеджере `Midnight Commander` для вызова команд можно применять сочетания клавиш с `<Ctrl>` или `<Alt>`. Если указано сочетание трех клавиш, то первые две нажимаются одновременно, третья — отдельно. Если символ после служебной клавиши расположен в верхнем регистре, то одновременно нажимаются все три клавиши, включая `<Shift>`. Если `Midnight Commander` не установлен, установите.
- 1.8. Для доступа к файлам, созданным в виртуальной машине `Ubuntu` или `Raspberry PI` ОС, применяйте протокол `ssh`. Для копирования файлов на хостовую машину (ваше окружение с `Windows` или `macOS`) используйте протокол `ftp` или команду `scp`. Протокол `ftp` доступен в менеджере файлов `Far Manager`. При этом учетная запись и пароль для `ftp` те же, что и при подключении по протоколу `ssh`.

### Критерии оценивания

Оценка 4

Выполнены задания 1-7. Файлы протоколов команд и меток времени с недочетами. Лабораторная работа сдана с задержкой в 2 недели.

#### **Оценка 5-6**

Выполнены задания 1-8, 10. Представлен отчет, файлы протоколов и меток времени в git-репозитории. Отчет, файлы протоколов команд и меток времени могут содержать ошибки. Лабораторная работа сдана с задержкой в 1 неделю.

#### **Оценка 7-8**

Выполнены задания 1-10 на хорошем уровне (есть недочеты). Представлен отчет, ответы на контрольные вопросы, файлы протоколов и меток времени в git-репозитории. Отчет, файлы протоколов команд и меток времени могут содержать незначительные ошибки. Лабораторная работа сдана с задержкой в 1 неделю.

#### **Оценка 9**

Выполнены задания 1-10 на отличном уровне. Представлен отчет, ответы на контрольные вопросы и файлы протоколов и меток времени в git-репозитории. Отчет, файлы протоколов команд и меток времени не содержат ошибок. Лабораторная работа сдана в срок.

### **Содержание отчета**

1. Цель работы.
2. Вариант задания.
3. Протоколы выполненных действий.
4. Ответы на контрольные вопросы.

Отчет должен быть опубликован в git-репозитории.

### **Задание 1. Развертывание виртуальных машин под управлением ОС Raspberry PI и ОС Ubuntu**

1.1. Изучить материалы учебного курса [Raspbian x86 on VirtualBox on a Windows PC](#) и [Как установить Ubuntu в VirtualBox](#).

1.2. Установить ОС Raspberry PI Desktop. Учесть, что в процессе установки создается учетная запись с именем pi и паролем raspberry. Пароль после установки рекомендуется сменить. Созданный пользователь обладает правами администратора. Продемонстрировать процедуру установки скриншотами в отчете.

1. Установить ОС Ubuntu Mate или Kali Linux. В процессе установки создать пользователя с именем по следующему шаблону ФамилияN, где Фамилия — заменить на вашу фамилию латиницей и строчными буквами, N — номер группы, например 12 или 13. Созданный пользователь обладает правами администратора. Продемонстрировать процедуру установки скриншотами в отчете.

## Задание 2. Настройка ssh-соединений между хостовой системой и гостевыми машинами

2.1. Изучить материалы из учебного курса по настройке ssh в ОС Raspberry PI и ОС Ubuntu:

- [Как настроить ssh-соединение в гостевой ОС в Virtualbox](#)
- [Как настроить ssh в ОС Raspbian](#)
- [Как настроить ssh ОС UBuntu](#)
- [Как пользоваться SSH? А также установка и настройка \(Linux, Windows, macOS, Android, iOS\)](#)

2.2. Настроить ssh-соединение в гостевой виртуальной ОС Raspberry PI и Ubuntu. Основные этапы настройки продемонстрировать скриншотами в отчете.

2.3. Подключиться по ssh из ОС Raspberry PI в Ubuntu и продемонстрировать скриншотами в отчете.

2.4. В зависимости от вашей основной операционной системы изучите рекомендации:

- [Как пользоваться SSH? А также установка и настройка \(Linux, Windows, macOS, Android, iOS\)](#)
- **Для Windows:** «[Как активировать ssh-клиент в Windows 10](#)» или установите клиент Putty для создания терминальных сессий, используя инструкцию «[Как подключиться по ssh из Windows](#)».
- **Для macOS:** «[Как использовать встроенный ssh-клиент в macOS](#)»

2.5. Открыть ssh-клиент в вашей хостовой системе Windows 10 или macOS. Подключитесь из хостовой машины в гостевые виртуальные машины под управлением ОС Raspberry PI и Ubuntu. Процесс подключения продемонстрировать скриншотами в отчете.

## Задание 3. Использование команды script для протоколирования команд

3.1. Изучить руководство по команде script в материале [Руководство по команде script](#) из учебного курса. Обратите внимание на пример команды, которая позволяет протолировать команды и вести журнал меток времени (пример 4).

3.2. Подключиться из хостовой ОС по ssh под пользователем с правами администратора в Ubuntu с именем ФамилияN, где Фамилия — заменить на вашу фамилию латиницей и строчными буквами, N — номер группы, например 12 или 13.

3.3. Включить ведение протокола командой script с журналом меток времени. **Протокол** назвать по следующему шаблону — taskXФамилияNM, где X — номер выполняемого задания, Фамилия — заменить на вашу фамилию латиницей и строчными буквами, N — номер группы, например 12 или 13, M — **r** — для Raspberry PI, **u** — для Ubuntu. **Журнал меток** назвать по следующему шаблону — timelogXФамилияNM, где X — номер выполняемого задания, Фамилия — заменить на вашу фамилию латиницей и строчными буквами, N — номер группы, например 12 или 13, M — **r** — для Raspberry PI, **u** — для Ubuntu.

3.4. Выполните команды help, ls, cd, pwd, mkdir, rmdir, rm, history. Если требуется изучите описания команд с помощью man и info.

3.5. Завершить протоколирование команд в Ubuntu, т. е. ввести команду `exit`.

3.6. Не выключая Ubuntu, включите гостевую ОС Raspberry PI. Подключитесь по `ssh` к гостевой виртуальной ОС Raspberry PI под учетной записью пользователя `pi`.

3.7. Включить ведение протокола командой `script` с журналом меток времени.

**Протокол** назвать по следующему шаблону — `taskXФамилияNM`, где `X` — номер выполняемого задания, `Фамилия` — заменить на вашу фамилию латиницей и строчными буквами, `N` — номер группы, например 12 или 13, `M` — `r` — для Raspberry PI, `u` — для Ubuntu. **Журнал меток** назвать по следующему шаблону — `timelogXФамилияNM`, где `X` — номер выполняемого задания, `Фамилия` — заменить на вашу фамилию латиницей и строчными буквами, `N` — номер группы, например 12 или 13, `M` — `r` — для Raspberry PI, `u` — для Ubuntu.

3.8. Создайте каталог, задав в качестве имени вашу фамилию латиницей строчными буквами. Данные каталог использовать для размещения результатов лабораторной работы в ОС Raspberry PI и сохраняйте в него результаты работы.

3.9. Выполнить команды `cal`, `date`, `who`, `whoami`.

3.10. Создать текстовый файл в ОС Raspberry PI с произвольным содержанием при помощи команды `tee`. Просмотреть содержимое файла при помощи команды `cat` и исправить в нем ошибки при помощи текстового редактора `vi`.

3.11. Изучить справку на команды `cp`, `unlink`.

3.12. Создать копию файла, созданного в п. 3.10, командой `cp`, удалить ее командой `unlink`.

3.13. Завершить протоколирование команд в Raspberry PI, т. е. ввести команду `exit`.

## Задание 4. Безопасное копирование файлов и публикация в репозиторий

4.1. Если время действия `ssh`-сессии к гостевой ОС Ubuntu истекло, подключиться по `ssh` к ней.

4.2. Включить ведение протокола командой `script` с журналом меток времени.

**Протокол** назвать по следующему шаблону — `taskXФамилияNM`, где `X` — номер выполняемого задания, `Фамилия` — заменить на вашу фамилию латиницей и строчными буквами, `N` — номер группы, например 12 или 13, `M` — `r` — для Raspberry PI, `u` — для Ubuntu. **Журнал меток** назвать по следующему шаблону — `timelogXФамилияNM`, где `X` — номер выполняемого задания, `Фамилия` — заменить на вашу фамилию латиницей и строчными буквами, `N` — номер группы, например 12 или 13, `M` — `r` — для Raspberry PI, `u` — для Ubuntu.

4.3. Создайте каталог, задав в качестве имени вашу фамилию латиницей строчными буквами. Данные каталог использовать для размещения результатов лабораторной работы в ОС Ubuntu.

4.4. Установить `git`, если требуется.

4.5. Перейдите в каталог, созданный в п. 4.3. Склонируйте репозиторий из `github`.

4.6. Скопируйте в каталог, созданный в п. 4.3, результаты работы в Ubuntu, полученные в ходе выполнения задания 3.

4.7. Скопируйте с помощью `scp` из Raspberry PI сохранённые протоколы работы и добавьте в git-репозиторий, созданный в ОС Ubuntu в п.4.5.

4.8. Завершите протоколирование команд с помощью `script`, т. е. введите команду `exit`.

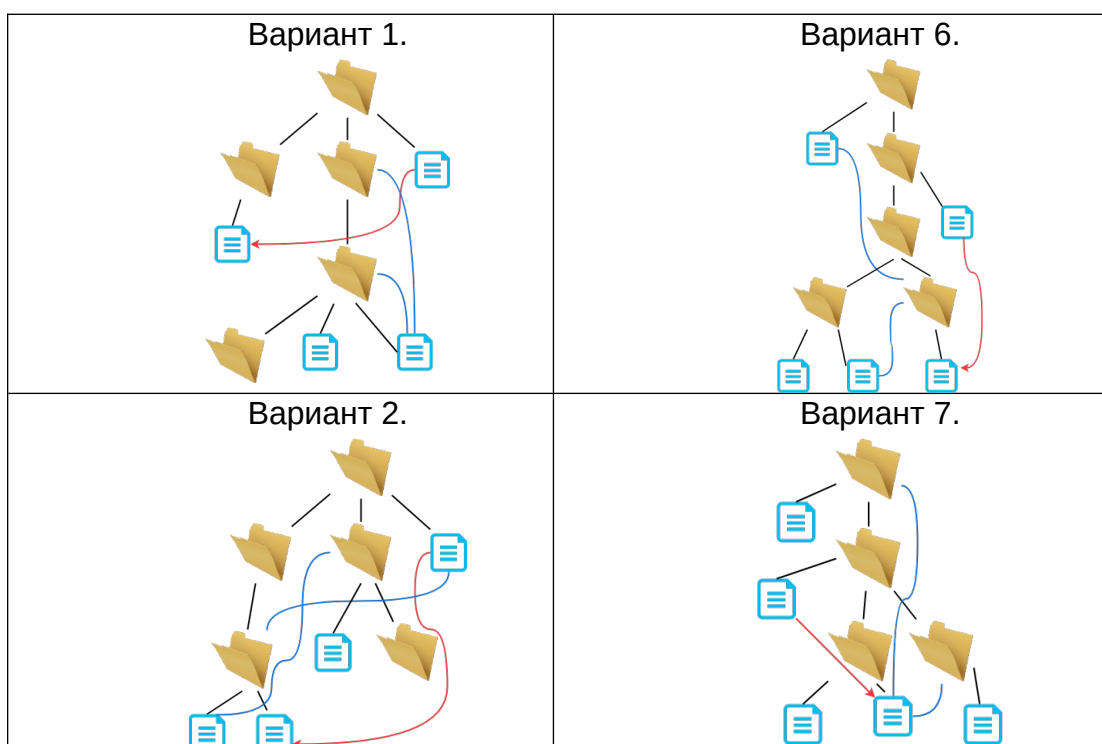
4.9. Опубликуйте изменения из локального репозитория, т. е. результаты выполнения заданий 3 и 4, в удаленный.

## Задание 5. Управление файлами и папками

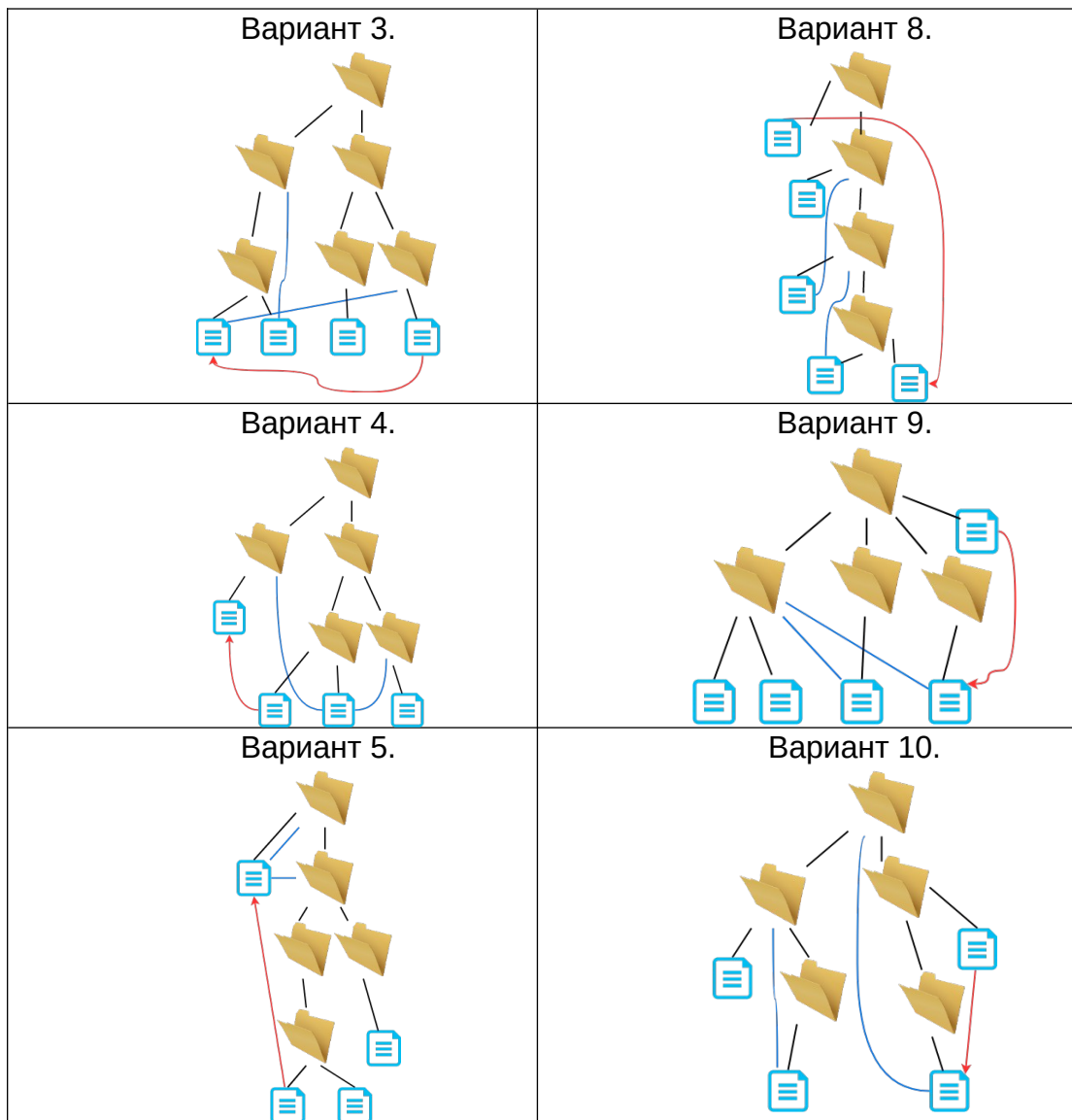
5.1. Задание выполнить в ОС Ubuntu и ОС Raspberry PI. Для каждой ОС вести протокол командой `script` с журналом меток времени. **Протокол** назвать по следующему шаблону — `taskXФамилияНКМ`, где `X` — номер выполняемого задания, `Фамилия` — заменить на вашу фамилию латиницей и строчными буквами, `N` — номер группы, например 12 или 13, `K` — номер варианта, `M` — `r` — для Raspberry PI, `u` — для Ubuntu. **Журнал меток** назвать по следующему шаблону — `timelogXФамилияНКМ`, где `X` — номер выполняемого задания, `Фамилия` — заменить на вашу фамилию латиницей и строчными буквами, `N` — номер группы, например 12 или 13, `K` — номер варианта, `M` — `r` — для Raspberry PI, `u` — для Ubuntu.

5.2. Создайте структуру каталогов в соответствии с вариантом в ОС Ubuntu. Черными линиями представлена вложенность файлов/подкаталогов в каталоги. Синими линиями представлены жесткие ссылки. Красными линиями — символические ссылки. Стрелка на красной линии указывает на целевой файл ссылки. Файлы создаются копированием ранее созданного файла командой `cp` с внесением в копии некоторых изменений. Ссылки создаются командой `ln`, символические ссылки — ей же, но с ключом `-s`:

Таблица 1.







5.3. Для всех вариантов выполнить следующие действия:

- Создать ссылки (синие линии).
- Создать символические ссылки (красные линии).
- Провести ряд экспериментов, иллюстрирующих доступ к файлам по основным именам, по ссылкам и по символическим ссылкам. Для доступа использовать команду `cat` или редактор `vi`.
- Провести ряд экспериментов, иллюстрирующих реакцию системы на удаление файла, на который имеются ссылки, и файла, на который имеются символические ссылки. Проверять результаты командой `ls -la`.
- Уничтожить созданные подкаталоги и файлы в них, используя команды `rmdir` и `unlink`, сохранив, однако, файл, созданный в пункте 5.2 и одну его рабочую копию в домашнем каталоге.

5.4. Завершите протоколирование команд с помощью `script`, т. е. введите команду `exit`.

5.5. Подключитесь по `ssh` из Ubuntu в ОС Raspberry PI и выполните шаги 5.1-5.4.

5.6. Скопируйте с помощью `scp` из Raspberry PI сохранённые протоколы работы и добавьте в `git`-репозиторий, созданный в ОС Ubuntu в п.4.5.

5.7. Опубликуйте изменения из локального репозитория, т. е. результаты выполнения задания 5, в удаленный.

## Задание 6. Управление пользователями и права доступа

6.1. Изучить справку к командам `useradd`, `groupadd`, `chmod` и `chown`.

6.2. Задание выполнить в ОС Ubuntu и ОС Raspberry PI. Для каждой ОС вести протокол командой `script` с журналом меток времени. **Протокол** назвать по следующему шаблону — `taskXФамилияNM`, где *X* — номер выполняемого задания, *Фамилия* — заменить на вашу фамилию латиницей и строчными буквами, *N* — номер группы, например 12 или 13, *M* — **π** — для Raspberry PI, **u** — для Ubuntu. **Журнал меток** назвать по следующему шаблону — `timelogXФамилияNM`, где *X* — номер выполняемого задания, *Фамилия* — заменить на вашу фамилию латиницей и строчными буквами, *N* — номер группы, например 12 или 13, *M* — **π** — для Raspberry PI, **u** — для Ubuntu.

6.3. Добавить группу `studentNM` в ОС Ubuntu, где *N* — номер группы, например 12 или 13, *M* — **π** — для Raspberry PI, **u** — для Ubuntu.

6.4. Добавить своего пользователя для каждой ОС в созданную группу.

6.5. Открыть для своей основной группы доступ к своему домашнему каталогу — для поиска в каталоге и к рабочей копии файла в домашнем каталоге — для чтения и записи.

6.6. Создать пользователя `studentNM` в ОС Ubuntu, где *N* — номер группы, например 12 или 13, *M* — **π** — для Raspberry PI, **u** — для Ubuntu. Открыть доступ на чтение и запись к одному из своих каталогов и одному из своих файлов для созданного пользователя.

6.7. Подключиться по `ssh` с созданной учетной записью или авторизоваться в гостевой ОС. И проверить права доступа на свой каталог и файл, внеся в них изменения.

6.8. Переключиться в свою учетную запись. Проверить изменения.

6.9. Закрыть доступ к своему домашнему каталогу.

6.10. Завершите протоколирование команд с помощью `script`, т. е. введите команду `exit`.

6.11. Подключитесь по `ssh` из Ubuntu в ОС Raspberry PI и выполните шаги 6.2-6.10.

6.12. Скопируйте с помощью `scp` из Raspberry PI сохранённые протоколы работы и добавить в `git`-репозиторий, созданный в ОС Ubuntu в п.4.5.

6.13. Опубликуйте изменения из локального репозитория, т. е. результаты выполнения задания 6, в удаленный.

## Задание 7. Базовые команды для управления процессами

7.1. Задание выполнить в ОС Ubuntu и ОС Raspberry PI. Для каждой ОС вести протокол командой `script` с журналом меток времени. **Протокол** назвать по следующему шаблону — `taskXФамилияNM`, где *X* — номер выполняемого задания, *Фамилия* — заменить на вашу фамилию латиницей и строчными буквами, *N* — номер группы, например 12 или 13, *M* — **π** — для Raspberry PI, **u** — для Ubuntu. **Журнал**

**меток** назвать по следующему шаблону — `timelogXФамилияNM`, где `X` — номер выполняемого задания, `Фамилия` — заменить на вашу фамилию латиницей и строчными буквами, `N` — номер группы, например 12 или 13, `M` — `r` — для Raspberry PI, `u` — для Ubuntu.

7.2. Изучите справку к команде `ps` и статью <https://www.tecmint.com/ps-command-examples-for-linux-process-monitoring/>, выполнить ее с ключами `-a`, `-e`, `a`, `x`, `ax` в ОС Raspberry PI, записать результаты в файл, например: `ps -e > ps.log`.

7.3. Авторизуйтесь в ОС Ubuntu по `ssh`. Изучите справку к команде `pidof` и статью <https://www.tecmint.com/find-process-name-pid-number-linux/>. Запустите команду `top`. А в другом терминале, используя `pidof` определите ее идентификатор процесса. Завершите выполнение команды `top`.

7.4. Авторизуйтесь в ОС Ubuntu по `ssh`. Изучите справку к команде `fuser` и статью <https://www.tecmint.com/learn-how-to-use-fuser-command-with-examples-in-linux/>. Откройте любой текстовый файл на редактирование. Определите, какие процессы используют ваш файл.

7.5. Авторизуйтесь в ОС Ubuntu по `ssh`. Изучите справку к команде `vmstat`. Результаты вывода команды запишите в файл с указанием в названии файла вашу фамилию и имя и название используемой команды.

7.6. Авторизуйтесь в ОС Ubuntu по `ssh`. Изучите справку к команде `lsof`. Результаты вывода команды запишите в файл с указанием в названии файла вашу фамилию и имя и название используемой команды.

7.7. Скопируйте с помощью `scp` из Raspberry PI сохранённые протоколы работы и добавить в `git`-репозиторий, созданный в ОС Ubuntu в п.4.5.

7.8. Опубликуйте изменения из локального репозитория, т. е. результаты выполнения задания 7, в удаленный.

## Задание 8. Сжатие и архивация файлов

8.1. Задание выполнить в ОС Ubuntu и ОС Raspberry PI. Для каждой ОС вести протокол командой `script` с журналом меток времени. **Протокол** назвать по следующему шаблону — `taskXФамилияNM`, где `X` — номер выполняемого задания, `Фамилия` — заменить на вашу фамилию латиницей и строчными буквами, `N` — номер группы, например 12 или 13, `M` — `r` — для Raspberry PI, `u` — для Ubuntu. **Журнал меток** назвать по следующему шаблону — `timelogXФамилияNM`, где `X` — номер выполняемого задания, `Фамилия` — заменить на вашу фамилию латиницей и строчными буквами, `N` — номер группы, например 12 или 13, `M` — `r` — для Raspberry PI, `u` — для Ubuntu.

### Утилиты `gzip` и `gunzip`

8.2. Включить запись протокола команд и меток времени командой `script`.

8.3. Подключиться по `ssh` со своей учетной записью к ОС Ubuntu или авторизоваться в виртуальной машине. Используя программу `gzip`, сжать файл любой текстовый файл в созданном каталоге согласно варианту из задания 5.

8.4. Используя программу `gzip`, сжать файл любой текстовый файл в созданном каталоге согласно варианту из задания 5, чтобы исходный файл остался в текущем каталоге.

8.5. Проверить целостность сжатого файла.

8.6. Скопировать полученный архив с помощью `scp` в домашний каталог своего пользователя в ОС Raspberry PI. Распаковать файл архивы `*.gz` в домашний каталог в ОС Raspberry PI.

8.7. Завершить протоколирование команд.

8.8. Проверить содержимое домашнего каталога в ОС Raspberry PI.

### Утилита `tar`

8.9. Включить в ОС Ubuntu запись протокола команд и меток времени командой `script`.

8.10. Создать в домашнем каталоге архив формата `tar`, включающий каталог и файлы.

8.11. Сжать полученный архив.

8.12. Скопировать полученный архив с помощью `scp` в домашний каталог своего пользователя в ОС Raspberry PI.

8.13. Завершить протоколирование команд.

8.14. Подключиться к ОС Raspberry PI и распаковать архив.

8.15. Скопируйте сохранённые протоколы работы и добавить в `git`-репозиторий, созданный в ОС Ubuntu в п.4.5.

8.16. Опубликуйте изменения из локального репозитория, т. е. результаты выполнения задания 8, в удаленный.

## Задание 9. Синхронизация в Linux с помощью утилиты `rsync`

9.1. Задание выполнить в ОС Ubuntu и ОС Raspberry PI. Для каждой ОС вести протокол командой `script` с журналом меток времени. **Протокол** назвать по следующему шаблону — `taskXФамилияNM`, где `X` — номер выполняемого задания, `Фамилия` — заменить на вашу фамилию латиницей и строчными буквами, `N` — номер группы, например 12 или 13, `M` — `r` — для Raspberry PI, `u` — для Ubuntu. **Журнал меток** назвать по следующему шаблону — `timelogXФамилияNM`, где `X` — номер выполняемого задания, `Фамилия` — заменить на вашу фамилию латиницей и строчными буквами, `N` — номер группы, например 12 или 13, `M` — `r` — для Raspberry PI, `u` — для Ubuntu.

9.2. Создать в домашнем каталоге каталог `rsync-backup` в ОС Ubuntu и в ОС Raspberry PI.

9.3. Скопировать несколько файлов в каталог `rsync-backup` в ОС Ubuntu.

9.4. Используя `rsync`, скопировать содержимое каталога `rsync-backup` из ОС Ubuntu в каталог `rsync-backup` в ОС Raspberry PI.

9.5. Внести изменения в любой из файлов в каталоге `rsync-backup` в ОС Ubuntu и синхронизировать изменения с каталогом в ОС Raspberry PI.

9.6. Скопируйте сохранённые протоколы работы и добавить в git-репозиторий, созданный в ОС Ubuntu в п.4.5.

9.7. Опубликуйте изменения из локального репозитория, т. е. результаты выполнения задания 9, в удаленный.

## Задание 10. Подключение по ssh к контейнеру Docker

10.1. Задание выполнить в **хостовой** системе. Для Windows и macOS найти документацию по установке контейнера Docker.

10.2. Просмотреть видео <https://youtu.be/V7ITLVzsK5U> с установкой docker на ОС Ubuntu 18.04.

10.3. Выполнить установку docker согласно документации, например для ОС Ubuntu <https://docs.docker.com/engine/install/ubuntu/>.

10.4. Установить docker и развернуть контейнер для ubuntu.

10.5. Изучить основные команды для работы с docker <https://youtu.be/XtO3hUWLKF8>.

10.6. Установить Docker Compose согласно документации <https://docs.docker.com/compose/install/>.

10.7. Изучить статью <https://phoenixnap.com/kb/how-to-ssh-into-docker-container> и <https://andreyex.ru/linux/kak-podklyuchitsya-po-ssh-k-kontejneru-docker/>.

10.8. Для контейнера ubuntu настроить подключение по ssh, скопировать ключ ssh с хостовой системы и подключиться к контейнеру docker ubuntu по ssh. Продемонстрировать подключение по ключу ssh (без применения пароля) и по нестандартному порту для ssh (стандартный пор для ssh — 22).

10.9. Изучить статьи для создания контейнера Docker для тестирования сборки для Android:

<https://andresand.medium.com/android-emulator-on-docker-container-f20c49b129ef>

<https://betterprogramming.pub/build-a-lightweight-docker-container-for-android-testing-2aa6bdaea422>

<https://andresand.medium.com/building-android-with-docker-8dbf717f54d4>

10.10. Настроить контейнер docker для Android и продемонстрировать запуск эмулятора.

## Контрольные вопросы.

1. Какие ключи команды `ls` Вы знаете? Какой ключ используется для рекурсивного обхода каталогов?

2. Чем отличаются `man` и `info`? Как с ними работать? Какая команда позволяет найти документацию по ключевому слову в каждом случае?
3. Команда `script` — назначение и применение.
4. Какая команда используется для генерации ssh-ключа?
5. Для каких задач применяется команда `scp`?
6. Команды `tee` и `cat`. Назначение и применение. Чем `cat` отличается от `more` и `less`?
7. Основные команды редактора `vi`.
8. Жесткие ссылки и символические ссылки. Создание и применение.
9. Создание и копирование файлов и папок в Linux.
10. Перемещение и удаление файлов и папок в Linux.
11. Команды `useradd`, `adduser`, `userdel`, `deluser`, `groupadd`, `addgroup`, `groupdel`, `delgroup`. Назначение и применение.
12. Команды `chmod` и `chown`. Назначение и применение.
13. Какие права доступа Вы имеете к своему домашнему каталогу, каталогам `/home` и `/`?
14. Как вы считаете, почему в Linux разделены команды для сжатия и команды архивирования? Приведите их примеры.
15. Какой алгоритм сжатия использует `gzip`?
16. В каком году был создан архиватор `xz`? Для каких задач применяется? Какой алгоритм сжатия использует?
17. Какой архиватор, созданный в 2015 году, поддерживает 22 уровня сжатия.
18. Команда `ps`. Назначение и применение. Ключи команды.
19. Команда `pidof`. Назначение и применение. Ключи команды.
20. Команда `fuser`. Назначение и применение. Ключи команды.

# Приложение 1

## 1. Синтаксис основных команд

**ls** — вывод содержимого каталога. **Синтаксис:** `ls [опции] [файл...]`

**Описание:** Команда `ls` сначала выводит список всех файлов (не каталогов), перечисленных в командной строке, а затем выводит список всех файлов, находящихся в каталогах, перечисленных в командной строке. Если не указано ни одного файла, то по умолчанию аргументом назначается '.' (текущий каталог). Опция `-d` заставляет `ls` не считать аргументы-каталоги каталогами. Будут отображаться только файлы, которые не начинаются с '.' или все файлы, если задана опция `-a`.

Результаты печатаются на стандартный вывод, по одному файлу на строку, если с помощью опции `-C` не задан многоколоночный вывод.

Каждый список файлов (для файлов, которые не являются каталогами и для каждого каталога, содержащего список файлов) сортируется отдельно в алфавитной последовательности.

### Опции:

`-l` В дополнение к имени каждого файла, выводятся тип файла, права доступа к файлу, количество ссылок на файл, имя владельца, имя группы, размер файла в байтах и временной штамп (время последней модификации файла, если не задано другое).

`-a` Выдавать все файлы в каталогах, включая все файлы и подкаталоги, имена которых начинаются с '.'.

`-d` Выдавать имена каталогов, как будто они обычные файлы, вместо того, чтобы показывать их содержимое.

`-L` Выдавать информацию о файлах, на которые указывают символические ссылки, вместо информации о самих символических ссылках.

`-R` Рекурсивно выдавать список содержимого всех каталогов.

`-h` Добавлять к каждому размеру файла букву размера, например, М (мегабайт).

`-X` Производить сортировку в алфавитном порядке по расширениям файлов (символы после последней '.'); файлы без расширений будут показаны первыми.

`-S` Производить сортировку по размеру файла, вместо сортировки по алфавиту. Таким образом, наибольшие файлы будут показаны сначала.

`-c` Сортировать содержимое каталога в соответствии с временем изменения состояния файла. Если с помощью опции `-l` задан длинный формат, то выдавать время изменения состояния файла вместо времени его модификации.

`-t` Сортировать по времени последней модификации вместо того, чтобы производить сортировку по алфавиту. Самые свежие файлы будут отображаться первыми.

`-u` Сортировать по времени последнего доступа к файлу, вместо времени последней модификации.

Типы файлов (первая буква в строке при задании опции `-l`) могут принимать следующие значения: — для обычного файла, `d` для каталога, `b` для блочного устройства, `c` для символьного устройства, `l` для символической ссылки, `r` для FIFO и `s` для гнезда (socket)

Права доступа составляют 9 символов и делятся на три группы по три символа: права доступа владельца, других пользователей из его группы, всех прочих пользователей. Права обозначаются следующим образом:

`r` Право на чтение.

`w` Право на запись.

`x` Право на выполнение (поиск в каталоге).

– Данное право доступа отсутствует.

Для каталога под правом на выполнение подразумевается право на просмотр в поисках требуемого файла.

**cd** — смена текущего каталога. **Синтаксис:** cd [каталог]

**Описание:** cd изменяет текущий каталог на каталог. Имя каталог может задаваться абсолютным (от корневого каталога) — в этом случае оно начинается с символа '/' — или относительным (от текущего каталога) — в этом случае оно начинается с символов './' или '../'. Если каталог не указан, текущим становится "домашний" каталог пользователя, определяемый значением переменной окружения \$HOME.

**pwd** — выдача имени текущего каталога. **Синтаксис:** pwd

**Описание:** Команда pwd выдает имя текущего (рабочего) каталога.

**mkdir** — создание каталога. **Синтаксис:** mkdir [опции] каталог...

**Описание:** Команда mkdir создает каталоги с заданными именами.

По умолчанию права доступа к каталогам устанавливаются в 0777 ('a+rwX').

**Опции:**

-m права Устанавливает права доступа к создаваемым каталогам. Эти права могут быть заданы либо в символьном виде, либо в виде восьмеричного числа, как описано в

**rmdir** — удаление пустых каталогов. **Синтаксис** rmdir [опции] каталог...

**Описание.** Команда rmdir удаляет пустые каталоги. Если какой-либо из аргументов каталог не указывает на существующий пустой каталог, то будет выдано сообщение об ошибке.

**Опции:**

-r Если каталог включает более, чем один компонент пути, то удаляется каталог, затем убирается последний компонент пути и удаляется получившийся каталог и т.д. до тех пор, пока все компоненты не будут удалены. Таким образом, команда rmdir -r a/b/c эквивалентна rmdir a/b/c; rmdir a/b; rmdir a.

**man** — форматирование и отображение онлайн-справочных страниц.

**Синтаксис:** man [раздел] имя...

**Описание:** Команда man выполняет форматирование и отображение онлайн-справочных страниц Unix. Если задан раздел, то man ищет только в заданном разделе руководства. имя — это обычно имя страницы руководства, которое, как правило, является именем команды, функции или файла.

Имеющиеся справочные страницы разбиты на несколько разделов. Важнейшими являются разделы:

1 — команды Unix;

2, 3 — системные вызовы Unix;

4, 5 — форматы файлов Unix.

Когда команда отображает страницу подсказки, в нижней строке экрана выводится приглашение man — символ ':'. После приглашения можно вводить внутренние команды man. В кратком руководстве следует упомянуть только две внутренние команды man:

h получение подробной информации о внутренних командах man;

q выход из man или переход к следующей странице, если команда man была введена с указанием нескольких имен команд.

Двигаться по отображаемой странице можно при помощи клавиш управления курсором.

Для получения более подробной информации о команде man введите: man man

**info** — отображение онлайн-справочных страниц. **Синтаксис:** info имя...

**Описание:** Команда info выполняет форматирование и отображение онлайн-справочных страниц Linux.



В текстах, отображаемых командой, могут быть наборы строк, озаглавленные "\*" Menu", каждая строка такого набора начинается с символа "\*". Выбрав курсором пункт меню и нажав клавишу Enter, можно получить страницу подсказки по этому пункту.

Независимо от положения курсора, после приглашения можно вводить внутренние команды info. В кратком руководстве следует упомянуть только две внутренние команды info:

h получение подробной информации о внутренних командах info;

q выход из info или переход к следующей странице, если команда info была введена с указанием нескольких имен команд.

Двигаться по отображаемой странице можно при помощи клавиш управления курсором.

**script** — протоколирование сеанса. **Синтаксис:** script [-a] файл

**Описание:** Команда script начинает "вложенный" сеанс и протоколирует весь терминальный ввод и вывод в заданном файле. Завершение вложенного сеанса и выполнения команды script происходит по нажатию комбинации клавиш Ctrl+D.

**Опции:**

-a добавление протокола нового сеанса к содержимому файла, если эта опция не задана, то файл создается заново.

**who** — кто в системе? **Синтаксис:** who [опции]

**Описание:** Команда who сообщает имя пользователя, имя терминальной линии, астрономическое время начала сеанса, продолжительность бездействия терминальной линии с момента последнего обмена, идентификатор процесса для каждого из пользователей, работающих в системе.

Сообщения команды who имеют следующий формат:

NAME STATE LINE TIME IDLE PID COMMENT

где NAME — входное имя пользователя; LINE — имя терминальной линии, под которым она фигурирует в каталоге /dev; TIME — время начала сеанса; IDLE — время (часы и минуты), протекшее с последнего момента активизации данной линии. Точка (.) свидетельствует о том, что это действующий терминал. PID — идентификатор процесса интерпретатора shell, обслуживающего данного пользователя; COMMENT — комментарий, характеризующий данную линию.

**Опции:**

-H отображение заголовков столбцов в выводимой информации

-i отображается поле IDLE

-q отображение только имен и количества пользователей, работающих в системе в данный момент; все прочие опции при этом игнорируются

-T аналогично -s, но при этом отображается также поле STATE, как:

+ терминал, на который можно передавать сообщения

- терминал, на который нельзя передавать сообщения

? терминал неисправен

-s выводятся только поля NAME, LINE и TIME; это опция по умолчанию.

**write** — передача сообщения другому пользователю. **Синтаксис:** write *адресат*

**Описание:** *Адресат* задается как сетевое имя пользователя. После запуска команда write устанавливает связь с адресатом и переходит в режим ожидания ввода. В момент установки связи на терминал адресата выводится сообщение:

Message from *отправитель* ...

Отправитель вводит любой текст, который отображается на терминале получателя.

Отправитель заканчивает сообщение нажатием комбинации клавиш Ctrl+D в начале строки. У адресата окончание сообщения индицируется строкой: EOF

Получатель может заблокировать/разблокировать вывод сообщений на свой экран при помощи команды `mesg`. При попытке передать сообщение на заблокированный терминал отправитель получает диагностику:  
`write: адресат has messages disabled`

**tee** — ответвление канала. **Синтаксис:** `tee [опции]... [файл]...`

**Описание:** Команда `tee` переписывает стандартный ввод на стандартный вывод и делает копии в файлах. Признаком окончания ввода является комбинация клавиш `Ctrl+D`.

**Опции:**

-а добавлять выводимую информацию в файлы, а не переписывать их с начала.

**cat** — слияние и вывод файлов. **Синтаксис:** `cat [-опции] файл ...`

**Описание:** Команда `cat` по очереди читает указанные файлы и выдает их содержимое на стандартный вывод. Так, например, `cat f` распечатывает содержимое файла `f`, а `cat f1 f2 > f3` сливает первые два файла и помещает результат в третий. Чтобы добавить файл `f1` к файлу `f2`, надо выполнить команду `cat f1 >> f2`. Если не указан ни один файл или среди аргументов встретился `-`, команда `cat` читает данные со стандартного ввода.

**Опции:**

-b Нумеруются непустые строки файла.

-s Нумеруются все строки файла. (Поле номера отделяется от текста символом табуляции).

-v Визуализация непечатных символов. Управляющие символы изображаются в виде `^X` (`CTRL+X`); символ `DEL` (восьмеричное `0177`) — в виде `^?`. Символы, не входящие в набор `ASCII` (то есть с восьмым битом, установленным в 1) выдаются в виде `M-x`, где `x` — определяемый младшими семью битами символ.

С опцией `-v` можно использовать следующие опции:

-t Визуализация символов табуляции в виде `^I`.

-e Визуализация символов перевода строки в виде `$` (строка при этом все же переводится).

Если опция `-v` не указана, то опции `-t` и `-e` игнорируются.

**vi** — текстовый редактор. **Синтаксис:** `vi имя_файла`

**Описание:** Редактор `vi` имеет три режима:

1. Командный — в этом режиме можно перемещаться по файлу и выполнять редактирующие команды над текстом. Команды вызываются ОБЫЧНЫМИ ЛАТИНСКИМИ БУКВАМИ.
2. Ввода текста — в этом режиме обычные латинские буквы будут вставляться в текст.
3. Режим строчного редактора `vi` используется для управления файлами (типа сохранить файл, зачитать файл и т.д.)

**VI в КОМАНДНОМ РЕЖИМЕ.**

**ЧТОБЫ ВЫЙТИ ИЗ ФАЙЛА БЕЗ СОХРАНЕНИЯ, нажмите:**

`ESC : q ! Enter`

**чтобы выйти из файла, сохранив изменения, нажмите:**

`ESC : w ! Enter`

`ESC : q Enter`

**выйти из файла с сохранением, одной командой:**

`ESC : wq Enter`

**для перехода В РЕЖИМ ВВОДА нужно нажать команды типа:**

`i` вставлять здесь

`A` вставлять с конца строки



заданного каталога, включая и заданный каталог (без ограничения на глубину дерева).

#### Опции:

-f Игнорировать несуществующие файлы и никогда не запрашивать подтверждение на удаления.

-i Выдавать запрос на удаление каждого файла. (Принята по умолчанию).

-r Рекурсивно удалять содержимое каталогов.

-v Выдавать имя каждого файла перед его удалением.

**ln** — создание ссылки на файл. **Синтаксис:** ln [-f] файл1 [файл2 ...] целевой\_файл

**Описание:** Команда ln делает целевой\_файл ссылкой на файл1. Файл1 не должен совпадать с целевым\_файлом. Если целевой\_файл является каталогом, то в нем создаются ссылки на файл1, файл2,... с теми же именами. Только в этом случае можно указывать несколько исходных файлов.

Если целевой\_файл существует и не является каталогом, его старое содержимое теряется.

#### Опции:

-f удаление существующего целевого файла

-s создание символической ссылки (по умолчанию создается жесткая ссылка)

**chmod** — изменение режима доступа к файлам. **Синтаксис:** chmod режим файл...

**Описание:** Команда chmod изменяет права доступа к указанным файлам (среди которых могут быть каталоги) в соответствии с указанным режимом. Режим может быть задан в абсолютном или символьном виде.

*Абсолютный вид* — восьмеричное число, являющееся поразрядным ИЛИ следующих режимов (названы не все режимы):

00400 Доступен для чтения владельцем.

00200 Доступен для записи владельцем.

00100 Доступен для выполнения (в случае каталога — для просмотра) владельцем.

00040 Доступен для чтения членами группы.

00020 Доступен для записи членами группы.

00010 Доступен для выполнения (просмотра) членами группы.

00004 Доступен для чтения прочими пользователями.

00002 Доступен для записи прочими пользователями.

00001 Доступен для выполнения (просмотра) прочими пользователями.

*Символьный вид* основан на однобуквенных обозначениях, которые определяют класс доступа и права доступа для членов данного класса. Права доступа к файлу зависят от идентификатора пользователя и идентификатора группы, в которую он входит. Режим в целом описывается в терминах трех последовательностей, по три буквы в каждой:

Владелец	Группа	Прочие
(u)	(g)	(o)
rwx	rwx	rwx

Для задания режима доступа в символьном виде используется синтаксис:

[кому] операция права

Часть кому есть комбинация букв u, g и o (владелец, члены группы и прочие пользователи соответственно). Если часть кому опущена или указано a, то это эквивалентно ugo.

Операция может быть: + (добавить право), — (лишить права), = (в пределах данного класса присвоить права абсолютно, то есть добавить указанные права и отнять неуказанные).

Права — любая осмысленная комбинация следующих букв (не все):

r Право на чтение.

w Право на запись.

x Право на выполнение (поиск в каталоге).

Опустить часть права можно только если операция есть = (для лишения всех прав).

Если надо сделать более одного указания об изменении прав, то при использовании символьного вида в правах не должно быть пробелов, а указания должны разделяться запятыми. Например, команда

```
chmod u+w,go+x f1
```

добавит для владельца право писать в файл f1, а для членов группы и прочих пользователей — право выполнять файл. Права устанавливаются в указанном порядке.

Изменить режим доступа к файлу может только его владелец или суперпользователь.

Для просмотра прав доступа и контроля при их изменении используется команда ls с флагом -l.

**chown** — изменение владельца и группы файлов.

**Синтаксис:** chown [опции] пользователь[:группа] файл...

**Описание:** Команда chown изменяет владельца и/или группу для каждого заданного файла. В качестве имени владельца/группы берется первый аргумент, не являющийся опцией. Если задано только имя пользователя (или числовой идентификатор пользователя), то данный пользователь становится владельцем каждого из указанных файлов, а группа этих файлов не изменяется. Если за именем пользователя через двоеточие следует имя группы (или числовой идентификатор группы), без пробелов между ними, то изменяется также и группа файла.

**Опции:**

-R Рекурсивное изменение владельца для каталогов и их содержимого.

**ps** — вывод информации о состоянии процессов. **Синтаксис:** ps [опции]

**Описание:** Команда ps выводит в стандартный вывод информацию о текущем состоянии процессов.

**Опции:**

-a все процессы, кроме лидеров групп и процессов, не ассоциированных с терминалом.

-d все процессы, кроме лидеров групп.

-e все процессы.

-g*список* выбирать процессы по *списку* лидеров групп.

-r*список* выбирать процессы по *списку* идентификаторов процессов.

-t*список* выбирать процессы по *списку* терминалов.

-u*список* выбирать процессы по *списку* идентификаторов пользователей.

-f генерировать полный листинг.

-l генерировать листинг в длинном формате.

**Результат команды ps:**

Ниже приводятся заголовки колонок выдачи команды ps и объясняется смысл их содержимого. Буквы l или f означают, что эта колонка появляется соответственно при длинном или полном формате выдачи; отсутствие букв означает, что данная колонка выводится всегда.

F l Флаги (шестнадцатеричные), логическая сумма которых дает следующие сведения о процессе:

00 процесс терминирован; элемент таблицы процессов свободен;

01 системный процесс: всегда в основной памяти;

02 процесс трассируется родительским процессом;

04 родительский трассировочный сигнал остановил процесс; родительский процесс ждет;

08 процесс не может быть разбужен сигналом;

10 процесс в основной памяти;

20 процесс в основной памяти; блокирован до завершения события;

40 идет сигнал к удаленной системе;  
80 процесс в очереди на ввод/вывод.

S I статус процесса:

- O активный: обрабатывается процессором;
- S спящий: ожидает завершения события;
- R готов: стоит в очереди на выполнение;
- I рождающийся: процесс создается;
- Z состояние "зомби": процесс завершен, но родительский процесс не ждет этого;
- T трассируемый: процесс остановлен сигналом, так как родительский процесс трассирует его;
- X растущий: процесс ожидает получения большего объема основной памяти.

UID f,I идентификатор владельца процесса; при указании опции -f выдается входное имя пользователя.

PID идентификатор процесса .

PPID f,I идентификатор родительского процесса.

C f,I доля выделенного планировщиком времени ЦП.

STIME f время запуска процесса (часы:минуты:секунды); если процесс запущен более чем 24 часа назад, выдается месяц и день запуска.

PRI I приоритет процесса.

NI I поправка к приоритету.

ADDR I адрес процесса в памяти.

SZ I размер (в блоках по 512 байт) образа процесса в памяти.

WCHAN I адрес события, которого ожидает процесс.

TTY I управляющий терминал.

TIME I использованное процессом время.

COMMAND I имя программы; если указана опция -f, то выводится полное имя команды и ее аргументы.

## 2. Учетные записи в ОС Linux

1. Linux, как и любая unix-подобная система, является не только многозадачной, но и многопользовательской, т.е эта операционная система позволяет одновременно нескольким пользователям работать с ней. Но система должна как-то узнавать, какой или какие из пользователей работают в данный момент. Именно для этих целей в Linux существует два понятия – *учетные записи* и *аутентификация*, которые являются частями одного механизма.

**Учетная запись пользователя** – это необходимая для системы информация о пользователе, хранящаяся в специальных файлах. Информация используется Linux для аутентификации пользователя и назначения ему прав доступа.

**Аутентификация** – системная процедура, позволяющая Linux определить, какой именно пользователь осуществляет вход.

Вся информация о пользователе обычно храниться в файлах `/etc/passwd` и `/etc/group`.

**/etc/passwd** – этот файл содержит информацию о пользователях. Запись для каждого пользователя занимает одну строку:

The diagram shows a shell prompt: `root:$1$OvdTRu2w$Cm5nk.6w6nmNdUierh5pu:0:0:root:/root:/bin/bash`. Brackets and labels identify the parts of the prompt:

- `root:` is labeled "имя пользователя" (username).
- `$1$OvdTRu2w$Cm5nk.6w6nmNdUierh5pu:` is labeled "зашифрованный пароль" (encrypted password).
- The first `0` is labeled "ID".
- The second `0` is labeled "ID".
- `root:` is labeled "омашний каталог" (home directory).
- `/root/` is labeled "ящее имя" (current directory).
- `/bin/bash` is labeled "олочка" (shell).

**имя пользователя** – имя, используемое пользователем на все приглашения типа login при аутентификации в системе.

**зашифрованный пароль** – обычно хешированный по необратимому алгоритму MD5 пароль пользователя или символ '!', в случаях, когда интерактивных вход пользователя в систему запрещен.

**UID** – числовой идентификатор пользователя. Система использует его для распределения прав файлам и процессам.

**GID** – числовой идентификатор группы. Имена групп расположены в файле /etc/group. Система использует его для распределения прав файлам и процессам.

**Настоящее имя пользователя** – используется в административных целях, а также командами типа finger (получение информации о пользователе через сеть).

**Домашний каталог** – полный путь к домашнему каталогу пользователя.

**Оболочка** – командная оболочка, которую использует пользователь при сеансе. Для нормальной работы она должна быть указана в файле регистрации оболочек /etc/shells.

**/etc/group** – этот файл содержит информацию о группах, к которым принадлежат пользователи:

project: \$1\$QydTRu2w\$Cm5gk.6w6nmNdUjerh5pu:100:root,bin,daemon

Имя группы                      Шифрованный пароль                      G ID                      Пользователи, включенные в несколько групп

**Имя группы** – имя, используемое для удобства использования таких программ, как newgrp.

**Шифрованный пароль** – используется при смене группы командой newgrp. Пароль для групп может отсутствовать.

**GID** – числовой идентификатор группы. Система использует его для распределения прав файлам и процессам.

**Пользователи, включенные в несколько групп** – В этом поле через запятую отображаются те пользователи, у которых по умолчанию (в файле /etc/passwd) назначена другая группа.

На сегодняшний день хранение паролей в файлах passwd и group считается ненадежным. В новых версиях Linux применяются так называемые теневые файлы паролей – shadow и gshadow. Права на них назначены таким образом, что даже чтение этих файлов без прав суперпользователя невозможно. Нужно учесть, что нормальное функционирование системы при использовании теневых файлов подразумевает одновременно и наличие файлов passwd и group. При использовании теневых паролей в /etc/passwd и /etc/group вместо самого пароля устанавливается символ 'x', что и является указанием на хранение пароля в /etc/shadow или /etc/gshadow.

Файл shadow хранит защищенную информацию о пользователях, а также обеспечивает механизмы устаревания паролей и учетных записей. Вот структура файла shadow:

cisco:\$1\$0AJZcVg0\$EGORy8Mh3swT1RfJeX.UR0:13770:10:99999:7:30:99999:

**а) имя пользователя**

**б) шифрованный пароль** – применяются алгоритмы хеширования, как правило MD5 или символ '!', в случаях, когда интерактивный вход пользователя в систему запрещен.

**в)** число дней последнего изменения пароля, начиная с 1 января 1970 года, последнего изменения пароля

**г)** число дней, перед тем как пароль может быть изменён

**д)** число дней, после которых пароль должен быть изменён

**е)** число дней, за сколько пользователя начнут предупреждать, что пароль устаревает

**ж)** число дней, после устаревания пароля для блокировки учётной записи

**з)** дней, отсчитывая с 1 января 1970 года, когда учётная запись будет заблокирована

**и)** зарезервированное поле

Файл gshadow так же накладывает дополнительную функциональность, в комплекте с защищенным хранением паролей групп. Он имеет следующую структуру:

root:\$1\$QydTRu2w\$Cm5gk.6w6nmNdUjrh5pu:root:cisco,oem

**Имя группы** – имя, используемое для удобства использования таких программ, как newgrp.

**Шифрованный пароль** – используется при смене группы командой newgrp. Пароль для групп может отсутствовать.

**Администратор группы** – пользователь, имеющий право изменять пароль с помощью grasswd.

**Список пользователей** – В этом поле через запятую отображаются те пользователи, у которых по умолчанию (в файле /etc/passwd) назначена другая группа.

2. В Linux, кроме обычных пользователей, существует один (и только один) пользователь с неограниченными правами. Идентификаторы UID и GID такого пользователя всегда 0. Его имя, как правило, root, однако оно может быть легко изменено (или создано несколько символьных имен с одинаковым GID и UID), так как значение для применения неограниченных прав доступа имеет только GID 0. Для пользователя root права доступа к файлам и процессам не проверяются системой. При работе с использованием учетной записи root необходимо быть предельно осторожным, т.к. всегда существует возможность уничтожить систему.

3. В Linux используется развитая система распределения прав пользователям. Но для точного опознания пользователя одного имени недостаточно с точки зрения безопасности. Именно поэтому используется и *пароль* – произвольный набор символов произвольной длины, обычно ограниченной лишь используемыми методами шифрования.

Сегодня в большинстве версий Linux пароли шифруются по алгоритмам 3DES и MD5. Когда алгоритм 3DES является обратимым, то есть такой пароль можно



расшифровать, MD5 – это необратимое преобразование. Пароли, зашифрованные по алгоритму 3DES не применяются при использовании теневых файлов для хранения паролей.

При аутентификации, пароль, введенный пользователем, шифруется тем же методом, что и исходный, а потом сравниваются уже зашифрованные копии. Если они одинаковые, то аутентификация считается успешной.

Учитывая ежедневно увеличивающиеся требования к безопасности, в Linux есть возможность использовать скрытые пароли. Файлы `/etc/passwd` и `/etc/group` доступны для чтения всем пользователям, что является довольно большой брешью в безопасности системы. Именно поэтому в современных версиях Linux предпочтительнее использовать скрытые пароли. Такие пароли располагаются в файлах `/etc/shadow` и `/etc/gshadow`, для паролей пользователей и групп соответственно.

4. Команда *login* запускает сеанс интерактивной работы в системе. Она проверяет правильность ввода имени и пароля пользователя, меняет каталог на домашний, выстраивает окружение и запускает командный интерпретатор. Команду *login* как правило не запускают из командной строки — это обычно за пользователя делает менеджеры консоли — например *getty* или *mgetty*.

Команда *su* (*switch user*) позволяет сменить идентификатор пользователя уже в процессе сеанса. Синтаксис ее прост: *su username*, где *username* – имя пользователя, которое будет использоваться. После этого программа запросит пароль. При правильно введенном пароле, *su* запустит новый командный интерпретатор с правами пользователя, указанного *su* и присвоит сеансу его идентификаторы. Если имя пользователя опущено, то команда *su* использует имя *root*.

```
[student@ns student]$ su root
Password:
[root@ns student]#_
```

При использовании команды *su* пользователем *root* она, как правило, не запрашивает пароль.

Команда *newgrp* аналогична по своим возможностям *su* с той разницей, что происходит смена группы. Пользователь должен быть включен в группу, которая указывается в командной строке *newgrp*. При использовании команды *newgrp* пользователем *root* она никогда не запрашивает пароль. Синтаксис команды аналогичен синтаксису команды *su*: *newgrp groupname*, где *groupname* – имя группы, на которую пользователь меняет текущую.

Команда *passwd* является инструментом для смены пароля в Linux. Для смены своего пароля достаточно набрать в командной строке *passwd*:

```
[student@ns student]$ passwd
Changing password for student
(current) UNIX password:
New password:
Retype new password:
passwd: all authentication tokens updated successfully
[student@ns student]$_
```

Для смены пароля группы и управления группой используется команда *grpasswd*. Для смены пароля достаточно набрать в командной строке *grpasswd GROUPNAME*. Сменить пароль вам удастся только если Вы являетесь администратором группы. Если пароль не пустой, то для членов группы вызов *newgrp* пароля не требует, а не члены группы должны ввести пароль. Администратор группы может добавлять и удалять пользователей с помощью параметров *-a* и *-d* соответственно. Администраторы могут использовать параметр *-r* для удаления пароля группы. Если пароль не задан, то только члены группы с помощью команды *newgrp* могут войти в группу. Указав параметр *-R* можно запретить доступ в группу по паролю с помощью команды *newgrp* (однако на членов группы это не распространяется). Системный администратор (*root*) может использовать параметр *-A*, чтобы назначить группе администратора.

Команда *chage* управляет информацией об устаревании пароля и учетной записи. Обычный пользователь (не *root*) может использовать команду только для просмотра своих параметров устаревания пароля:

```
gserg@ADM:/$ chage -l gserg
Last password change                : Май 03, 2007
Password expires                    : never
Password inactive                   : never
Account expires                     : never
Minimum number of days between password change : 0
Maximum number of days between password change : 99999
Number of days of warning before password expires : 7
```

Суперпользователь же может использовать также иные параметры, такие как:  
*-d* дата (в формате системной даты, например ДД.ММ.ГГГГ) – устанавливает дату последней смены пароля пользователем.

*-E* дата – установить дату устаревания учетной записи пользователя

*-I* N – установить количество дней неактивности N с момента устаревания пароля перед тем как учетная запись будет заблокирована

*-m* N – задает минимальное количество дней (N) между сменами пароля

*-M* N – задает максимальное количество дней (N) между сменами пароля

*-W* N – задает количество дней, за которые будет выдаваться предупреждение об устаревании пароля.

### 3. Управление пользователями и группами

#### 3.1 Добавление учетной записи пользователя

1. Войдите в систему под именем *root*. Создайте пользователя с именем *temp*, действие учетной записи которого истекает 31 декабря 2016 года, выполнив следующую команду:

```
# useradd -e 2016-12-31 temp
```

2. Создайте пользователя с именем *user*, учетная запись которого становится недействительной через 14 дней после истечения действия пароля, выполнив следующую команду:

```
# useradd -f 14 user
```

## 3.2 Установка пароля пользователя

1. Установите пароль temppass для учетной записи пользователя temp, выполнив команду

```
# passwd temp
```

Установите пароль userpass для учетной записи user, выполнив команду

```
# passwd user
```

Проверьте создание паролей, войдя в командную оболочку под именами пользователей temp и user.

2. Войдите в командную оболочку как пользователь root и заблокируйте пароль пользователя temp, выполнив команду

```
# passwd -l temp
```

Проверьте, что пароль пользователя temp заблокирован. Войдите в командную оболочку как пользователь root и разблокируйте пароль пользователя temp, выполнив команду

```
# passwd -u temp
```

Проверьте, что пароль пользователя temp разблокирован.

## 3.3 Изменение информации о времени действия пароля пользователя

1. Просмотрите информацию о пользователе temp, выполнив команду времени действия пароля

```
# chage -l temp
```

2. Измените дату истечения действия пароля на 14 января 2017 года

```
# chage -E 2017-01-14 temp
```

## 3.4 Изменение личной информации о пользователе

1. Задайте следующую личную информацию о пользователе temp:

- настоящее имя: Nick;

- номер офиса: 12;

- номер офисного телефона 209-52-73;

- номер домашнего телефона: 217-52-73;

выполнив следующую команду:

```
# chfn -f Nick -o 12 -p 209-52-73 -h 217-52-73 temp
```

## 3.5 Модификация учетной записи пользователя

1. Измените имя пользователя temp на Nick, выполнив команду

```
# usermod -l Nick temp
```

2. Измените домашний каталог пользователя Nick на /home/Nick, выполнив команду

```
# usermod -d /home/Nick Nick
```

### 3.6 Добавление группы

1. Создайте группу с именем agroup, выполнив команду

```
# groupadd agroup
```

2. Назначьте пользователя Nick администратором группы agroup

```
# gpasswd -A Nick agroup
```

3. Войдите в систему под именем Nick и включить пользователя user в группу agroup

```
$ gpasswd -a user agroup
```

4. Переименуйте группу agroup на группу bgroup

```
# groupmod -n bgroup agroup
```

### 3.7 Удаление учетной записи пользователя и группы

1. Удалите группу bgroup, выполнив команду

```
# groupdel users
```

2. Удалите учетные записи пользователей Nick и user, выполнив команды

```
# userdel -r Nick
```

```
# userdel -r user
```

## 4. Управление файлами, каталогами и права доступа

### 4.1. Создание и просмотр файлов

1. Входим в систему по имени student и создаем в текущем

каталоге пустой файл с именем report.txt:

```
$ touch report.txt
```

2. Открываем файл текстовым редактором vim или nano и набираем в нем произвольный текст

```
$ vim report.txt
```

3. Выводим содержимое файла report.txt на консоль

```
$ cat report.txt
```

4. Выводим на консоль первых десять строк файла report.txt

```
$ head report.txt
```

5. Выводим на консоль первых две строки файла report.txt

```
$ head -n 2 report.txt
```

6. Выводим на консоль последних десять строк файла report.txt

```
$ tail report.txt
```

7. Выводим на консоль последних две строки файла report.txt

```
$ tail -n 2 report.txt
```

## **4.2 Создание и просмотр каталогов**

1. Создаем в текущем каталоге новый каталог с именем documents:

```
$ mkdir documents
```

2. Просматриваем содержимое текущего каталога

```
$ ls -l
```

3. Просматриваем содержимое текущего каталога, включая скрытые файлы

```
$ ls -a
```

4. Просматриваем файлы текущего каталога в порядке возрастания времени их последней модификации:

```
$ ls -ltr
```

## **4.3 Копирование и перемещение файлов**

1. Копируем файл report.txt из текущего каталога в подкаталог documents текущего каталога:

```
$ cp report.txt documents
```

2. Переименовываем файл report.txt в файл report.bak

```
$ mv report.txt report.bak
```

3. Перемещаем файл report.bak в каталог documents

```
$ mv report.bak documents
```

4. Просматриваем содержимое каталога documents

```
$ ls -l documents
```

## **4.4 Работа со ссылками**

1. Создаем в текущем каталоге жесткую ссылку на файл report.txt, который находится в каталоге documents:

```
$ ln ~/documents/report.txt
```

2. Создаем в текущем каталоге жесткую ссылку с именем rep.txt на файл report.txt, который находится в каталоге documents:

```
$ ln ~/documents/report.txt rep.txt
```

3. Создаем в текущем каталоге символическую ссылку на файл report.bak, который находится в каталоге documents:

```
$ ln -s ~/documents/report.bak
```

4. Создаем в текущем каталоге мягкую ссылку doc на каталог documents:

```
$ ln -s ~/documents/ doc
```

5. Просматриваем содержимое мягкой ссылки doc:

```
$ readlink doc
```

6. Открываем в текстовом редакторе vim файл report.txt через жесткую ссылку, созданную в текущем каталоге:

```
$ vim report.txt
```

Вносим изменения в файл и закрываем его.

7. Открываем в текстовом редакторе vim файл rep.txt через мягкую ссылку на каталог documents, созданную в текущем каталоге:

```
$ vim ~/doc/report.txt
```

и просматриваем изменения в этом файле.

8. Удаляем созданные мягкие и жесткие ссылки:

```
$ unlink report.txt
```

```
$ unlink report.back
```

9. Просматриваем содержимое каталога documents через ссылку:

```
$ ln -l doc
```

10 Удаляем созданную мягкую ссылку на каталог documents:

```
$ unlink doc
```

## **4.5 Управление разрешениями на доступ к файлу**

1. Входим в систему под именем root. Создаем в текущем каталоге файл message.txt.

```
# touch message.txt
```

2. Просматриваем разрешения на доступ к файлам текущего каталога:

```
# ls -l
```

3. Устанавливает пользователя с именем student владельцем файла message.txt:

```
# chown student message.txt
```

4. Устанавливаем пользователя с именем student владельцем каталога documents и всех файлов, находящихся в этом каталоге:

```
# chown -R student ~/documents
```

5. Создаем группу с именем students

```
# groupadd students
```

6. Устанавливаем группу с именем students группой файла

message.txt:

```
# chgrp students message.txt
```

7. Устанавливаем группу с именем students группой каталога

documents и всех файлов этого каталога

```
# chgrp -R students ~/documents
```

Просматриваем разрешения на доступ к файлам текущего каталога:

```
# ls -l
```

8. Для того чтобы разрешить владельцу файла message.txt читать и писать в этот файл, а членам группы файла и всем остальным пользователям разрешить только читать этот файл (-rw-r--r--), нужно выполнить следующую команду:

```
# chmod 644 message.txt
```

```
# ls -l
```

9. Для того чтобы сбросить все разрешения на доступ к файлу message.txt для владельца файла и установить для него только разрешение на запись в файл, нужно выполнить следующую команду:

```
# chmod u=w message.txt
```

```
# ls -l
```

10. Просматриваем текущую маску разрешений на доступ к файлу в числовом и символьном формате:

```
# umask
```

```
# umask -S
```

11. По умолчанию для создаваемых файлов и каталогов установлены следующие системные маски разрешений на доступ: 666 -для файлов, 777 - для каталогов, а маска разрешений на доступ к файлу команды umask установлена на 022. Если выполнена команда

```
# umask 200
```

то при создании файлов и каталогов будут использоваться следующие маски разрешений на доступ: 466 -для файлов, 577 - для каталогов, что эквивалентно следующим символьным обозначениям:

- r-rw-rw- для файлов;

- r-xrwxrwx для каталогов.

12. Создаем новый файл и проверяем его маску разрешений на доступ:

```
# touch testmask.txt
```

```
# ls -l
```

## 4.6 Удаление файлов и каталогов

1. Удаляем файлы message.txt и testmask.txt:

```
# rm message.txt testmask.txt
```

2. Входим в систему под именем student и удаляем файл report.txt из каталога documents:

```
$ rm rep.txt
```

3. Удаляем файл report.bak из каталога documents с запросом на подтверждение удаления:

```
$ rm -i ~/documents/report.bak
```

Проверяем, что каталог documents пустой.

4. Удаляем пустой каталог documents из текущего каталога:

```
$ rmdir documents
```

## 5. Архивация файлов и каталогов в ОС Linux

### 5.1 Утилиты zip и unzip

1. Создаем в текущем каталоге файлы report.txt и message.txt

```
$ touch report.txt
```

```
$ touch message.txt
```

С помощью текстового редактора nano вводим в созданные файлы произвольный текст

```
$ nano report.txt
```

```
$ nano message.txt
```

2. Создаем zip архив arch1 с файлами report.txt и message.txt

```
$ zip arch1 report.txt message.txt
```

3. Создаем каталог documents и перемещаем в него файлы report.txt и message.txt

```
$ mkdir documents
```

```
$ mv report.txt message.txt documents
```

4. Сжимаем каталог documents в архив arch2

```
$ zip arch2 documents
```

5. Создаем в текущем каталоге файл temp.txt и вводим в него произвольный текст.

```
$ touch temp.txt
```

```
$ nano temp.txt
```

6. Добавляем файл temp.txt в архивы arch1 и arch2

```
$ zip arch1 temp.txt
```

```
$ zip arch2 temp.txt
```



7. Создаем zip архив arch3, в который помещаем все файлы из текущего каталога

```
$ zip arch3 *
```

8. Просматриваем содержимое архивов arch1 и arch2

```
$ unzip -l arch1
```

```
$ unzip -l arch2
```

9. Проверяем целостность архива arch3

```
$ unzip -t arch3
```

10. Удаляем текстовые файлы и каталог documents из текущего каталога

```
$ rm *.txt
```

```
$ rm -r documents
```

11. Распаковываем архив arch1 и arch2 в текущий каталог

```
$ unzip arch1
```

```
$ unzip arch2
```

12. Проверяем содержимое текущего каталога

```
$ ls -l
```

## 5.2 Утилиты gzip и gunzip

1. Сжимаем файл report.txt в текущем каталоге

```
$ gzip report.txt
```

Просматриваем текущий каталог

```
$ ls -l
```

Как видим, команда gzip удалила исходный файл report.txt из текущего каталога.

2. Сжимаем файл temp.txt в текущем каталоге

```
$ gzip -c temp.txt > temp.txt.gz
```

Просматриваем текущий каталог

```
$ ls -l
```

Как видим, в этом случае в текущем каталоге сохраняется файл temp.txt.

3. Проверяем целостность сжатого файла report.txt.gz

```
$ gunzip -t report.txt.gz
```

4. Распаковываем файл report.txt.gz в текущий каталог

```
$ gunzip report.txt.gz
```

Просматриваем текущий каталог

```
$ ls -l
```

Как видим, команда gunzip удалила исходный сжатый файл report.txt.gz из текущего каталога.

5. Распаковываем файл temp.txt.gz в текущий каталог

```
$ gunzip -c temp.txt.gz > test.txt
```

Просматриваем текущий каталог

```
$ ls -l
```

Как видим, команда распаковала сжатый файл test.txt.gz остался в текущем каталоге.

## 5.3 Утилита tar

1. Создаем в текущем каталоге tar архив atar1, включающий файлы report.txt, message.txt и temp.txt, сжатых программой gzip

```
$ tar -czf atar1.tar.gz report.txt message.txt temp.txt
```

2. Проверяем архив atar1, сжатый программой gzip

```
$ tar -tvzf atar1.tar.gz
```

3. Распаковываем в текущий каталог архив atar1, сжатый программой gzip

```
$ tar -xvzf atar1.tar.gz
```

## 5.4 Утилита rsync

Программа *Rsync*, используется для удаленного копирования, резервного копирования или синхронизации файлов и каталогов, с минимальными затратами трафика. Рассмотрим примеры:

1. Создаем копию папки /home/student в папке /tmp

```
# rsync -a /home/student/ /tmp/student/
```

2. Синхронизируем две папки на локальном компьютере

```
#rsync -zvr /usr/local/ /home/student/temp
```

3. Синхронизируем две папки на локальном компьютере с сохранением временных меток

```
#rsync -avr /usr/local/ /home/student/temp
```

4. Синхронизация одного файла

```
#rsync -v /home/student/test.txt /home/student/temp/
```