МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

НИЖЕГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ

УНИВЕРСИТЕТ им. Р.Е.АЛЕКСЕЕВА

Институт радиоэлектроники и информационных технологий

ОТЧЕТ

по лабораторной работе №3

«Работа пользователя в LINUX»

по дисциплине

«Эксплуатация современных операционных систем»

РУКОВОДИТЕЛЬ:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Кочешков А. А

(подпись) (фамилия, и.,о.)

СТУДЕНТ:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Сухоруков В. А

(подпись) (фамилия, и.,о.)

19-В-2\_\_\_\_\_

(шифр группы)

Работа защищена «\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_

С оценкой \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Нижний Новгород 2022

Оглавление

[Цель работы 3](#_Toc105533368)

[Ход работы 3](#_Toc105533369)

[1. Регистрация в системе. 3](#_Toc105533370)

[2. Основные информационные команды 4](#_Toc105533371)

[1) Руководство man 4](#_Toc105533372)

[2) Команда apropos 5](#_Toc105533373)

[3) Файлы HOWTO 5](#_Toc105533374)

[3. Управление учетными записями пользователей. 6](#_Toc105533375)

[1) Получение информации о пользователях 6](#_Toc105533376)

[Команда who 6](#_Toc105533377)

[Команда w 6](#_Toc105533378)

[Команда id 6](#_Toc105533379)

[Команда logname 7](#_Toc105533380)

[Команда finger 7](#_Toc105533381)

[2) Создание новой учетной записи 7](#_Toc105533382)

[3) Файлы /etc/passwd и /etc/shadows 8](#_Toc105533383)

[4) Псевдопользователи 9](#_Toc105533384)

[4. Работа с устройствами и файловой системой. 10](#_Toc105533385)

[1) Специальные файлы устройств 10](#_Toc105533386)

[2) Использование файлов дисковых устройств в командах управления 10](#_Toc105533387)

[3) Аппаратная конфигурация компьютера 11](#_Toc105533388)

[4) Распределение аппаратных ресурсов 12](#_Toc105533389)

[5. Управление файлами 15](#_Toc105533390)

[6. Управление доступом к файловой системе. 16](#_Toc105533391)

[1) Состав индексного дескриптора файла. 16](#_Toc105533392)

[2) Определение владельца-пользователя и владельца группы 16](#_Toc105533393)

[3) Режимы доступа для файла и каталога 17](#_Toc105533394)

[7. Управление заданиями 19](#_Toc105533395)

[8. Базовая файловая структура Linux 20](#_Toc105533396)

[Вывод 22](#_Toc105533397)

# Цель работы

Изучить устройство и возможности операционной системы Linux. Получить опыт использования типовых команд.

# Ход работы

## Регистрация в системе.

Для регистрации в системе необходимо ввести имя пользователя, как аргумент команды login, и пароль, как аргумент команды passwd (Рис 1).

После того, как были правильно введены имя пользователя и пароль, система выводит на экран приглашение командной строки, это означает, что произошла идентификация пользователя

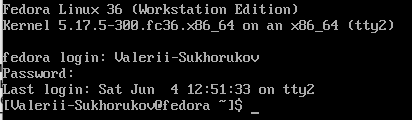


Рис 1. Регистрация в системе

Командой passwd без параметров можно изменить пароль текущей учетной записи, или другой учетной записи, указав имя пользователя.

С помощью команды logout можно выйти из системы и зайти под другой учетной записью (Рис 2).

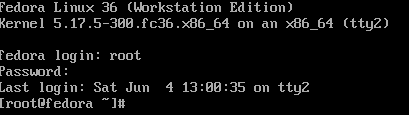


Рис 2. Вход под учетной записью root

С помощью команды exit можно завершить текущий процесс. Если никаких процессов не запущено, то произойдет выход из системы.

Перезапустить компьютер можно командой reboot без параметров, или командой shutdown с ключом –r и параметром now. Командой shutdown –h можно выключить компьютер в данный момент – параметр now, или в определенное время – параметром является значение часов и минут, например, 13:20.

Команда halt тоже выключает компьютер. Она не выполняет никаких подготовительных действий перед выключением, а просто отключает питание. Использование halt может повредить систему.

## Основные информационные команды

### Руководство man

Команда man позволяет получить доступ к общей базе справки по команде, функции или программе. Обычно для просмотра справки программе надо передать название команды или другого объекта в системе.

Получим справку о файле паролей passwd. Для этого используем команду man 5 passwd.

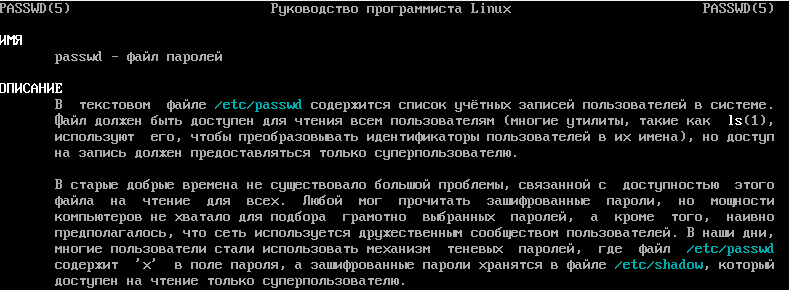


Рис 3. Справка о файле passwd

Цифра в команде – номер раздела. Таблица с номерами разделов и их назначением:

|  |  |
| --- | --- |
| **Номер раздела** | **Описание** |
| 1 | Основные команды. |
| 2 | Системные вызовы. |
| 3 | Библиотечные функции, включая стандартную библиотеку языка C. |
| 4 | Специальные файлы (обычно устройства из /dev) и драйвера. |
| 5 | Форматы файлов и соглашения. |
| 6 | Игры. |
| 7 | Разное. |
| 8 | Команды системного администрирования и демоны. |

Теперь получим справку о команде passwd, указав в качестве номер раздела 1: man 1 passwd.

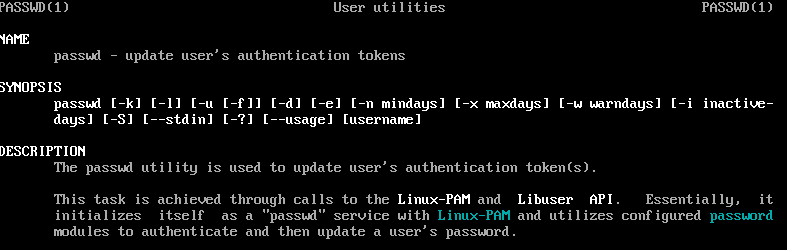


Рис 4. Справка о команде passwd

### Команда apropos

Утилита apropos выполняет поиск ключевого слова в первых строчках man-страниц и выводит те стоки, которые содержат указанное ключевое слово. Для того чтобы вывести информацию только из man-страниц определенного раздела, существует опция -s, за которой указывается номер man-страницы.

Среди основных программ (категория 1) найдем те, которые содержат в названии, или в описании слово install (Рис 5).

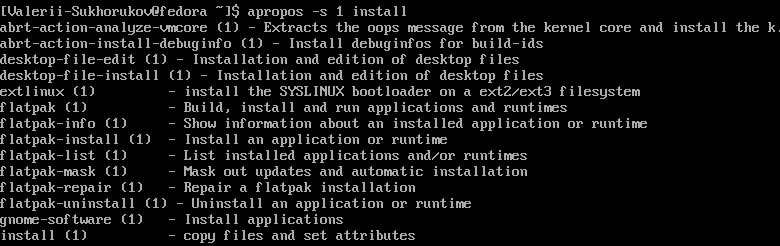


Рис 5. Поиск информации с помощью команды apropos

Посмотрим подробную справку о команде flatpak командой man 1 flatpak:

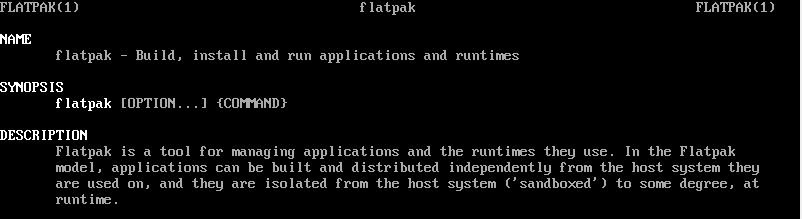


Рис 6. Справка команды flatpak

### Файлы HOWTO

HOWTO – (how to - как сделать). Эти документы описывают, как делать что-либо. Если установлен пакет с коллекцией HOWTO, то их можно найти в каталоге /usr/doc/Linux-HOWTOs. Эти файлы очень полезны в случае, если нужно найти информацию о том, как сделать какое-то определённое действие.

Русскоязычные файлы можно найти на сайте <http://rus-linux.net/lib.php?name=/MyLDP/HOWTO-ru/index.html>



Рис 7. Пример HOWTO файла

## Управление учетными записями пользователей.

### Получение информации о пользователях

#### Команда who

Команда who служит для получения информации о пользователях, которые подключены к системе, в том числе и об терминальных сессиях, через которые происходит подключение. При выполнении команды используются данные из файла /etc/utmp.

Синтаксис: who опции имя\_файла аргументы

Опции WHO:

* -a (--all) — включает в себя все основные опции.
* - b (--boot) — показывает время загрузки операционной системы.
* -d (--dead) — выводит перечень зомби-процессов.
* - H (--heading) — добавляет колонкам заголовки
* -m — показать пользователя, который сейчас работает в терминале.
* -r — вывести текущий уровень запуска (runinit);
* -t — показать последнее изменение системных часов;
* -s — вывести только имя, терминальную сессию и время.
* -q — вывести количество авторизованных пользователей.
* -T — данные о терминальной сессии.
* -u — показать активных пользователей.



Рис 8. Использование команды who

#### Команда w

Используя команду w, можно узнать имена активных учетных записей, время их регистрации в системе и бездействия, названия терминалов, соединенных со стандартным вводом (tty), а также JCPU (время, использованное всеми процессами под управлением tty) и PCPU (время, использованное текущим процессом). Текущие процессы, которыми занята каждая из учетных записей, также будут отображены в выводе команды.

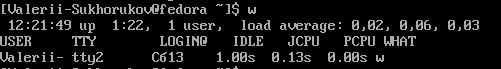


Рис 9. Использование команды w

#### Команда id

При вызове без какой-либо опции id печатает реальный идентификатор пользователя ( uid ), реальный идентификатор основной группы пользователя ( gid ) и реальные идентификаторы дополнительных групп ( groups ), к которым принадлежит пользователь.



Рис 10. Использование команды id

#### Команда logname

Вы можете отобразить имя текущего пользователя с помощью команды logname. Эта команда читает файл var/run/utmp или /etc/utmp для отображения имени текущего пользователя.



Рис 11. Использование команды logname

#### Команда finger

С помощью опции -s finger отображается имя пользователя, его реальное имя, имя терминала и статус записи, время простоя, время входа в систему, местоположение офиса и номер телефона офиса.



Рис 12. Использование команды finger –s

Опция -l выполняет команду finger, создавая многострочный формат, отображающий всю информацию, описанную для опции -s, а также домашний каталог пользователя, домашний номер телефона, оболочку входа, статус почты и содержимое файлов «.plan», «.project», «.pgpkey» и «.forward» из домашнего каталога пользователя.

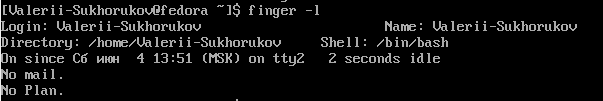


Рис 13. Использование команды finger –l

### Создание новой учетной записи

Для создания пользователей из командной строки используют утилиты **adduser** или **useradd**.

**useradd** — это низкоуровневая утилита для создания пользователей в Linux.

**adduser** — представляет собой более простое решение для создания пользователей и по факту является надстройкой над useradd, groupadd и usermod.

Утилита adduser доступна не во всех дистрибутивах Linux. Реализация adduser также может отличаться. Если в дистрибутиве присутствует утилита adduser, то для создания пользователей рекомендуется использовать именно ее.

Чтобы создать нового пользователя, выполним команду adduser и укажем имя пользователя:



Рис 14. Создание нового пользователя

В результате выполнения команды adduser будут выполнены следующие действия:

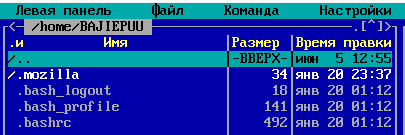
* Создается новый пользователь с именем, которое было указано при выполнении команды.
* Создается группа с тем же именем.
* Создается домашний каталог пользователя в директории /home/имяпользователя
* В домашний каталог копируются файлы из директории /etc/skel. В данной директории хранятся файлы, которые копируются в домашний каталог всех новых пользователей.

Рис 15. Домашний каталог пользователя

Установим пароль для учетной записи командной passwd:

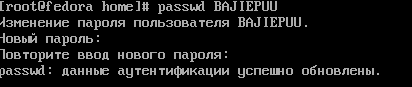


Рис 16.

### Файлы /etc/passwd и /etc/shadows

В файле passwd определены все пользователи, которые существуют в системе. Каждой строкой в **/etc/passwd** определяется аккаунт пользователя. Когда-то этот файл хранил хешированные пароли всех пользователей в системе. По соображениям безопасности позже эта ответственность была перенесена в отдельный файл. Этот файл доступен для непривилегированных пользователей. То есть, любой пользователь системы может читать данный файл. Поэтому пароли были перенесены в другой файл.

Поля информации разделяются двоеточием (:). Каждая строка типичного Linux-файла «/etc/passwd» содержит 7 полей:

* **BAJIEPUU**: имя пользователя;
* **х**: место для информации о паролях; пароль можно найти в файле «/etc/shadow».
* **1001**: ID пользователя. Каждый пользователь имеет уникальный идентификатор, благодаря которому система распознает его. ID root-пользователя всегда 0;
* **1001**: ID группы. Каждая группа имеет уникальный идентификатор. По умолчанию у каждого пользователя есть главная группа. Опять же, ID root-группы всегда 0;
* Поле для примечаний. Данное поле можно использовать для описания пользователя или его функций. Оно может содержать что угодно, начиная от контактной информации пользователя и заканчивая описанием сервисов, для которых была создана учетная запись;
* **/home/BAJIEPUU**: домашний каталог. Для обычных пользователей домашним каталогом является «/home/username», для root-пользователя это «/root»;
* /**bin/bash**: оболочка пользователя. Данное поле содержит оболочку, которая будет создана, или команды, которые будут выполняться при входе пользователя в систему.

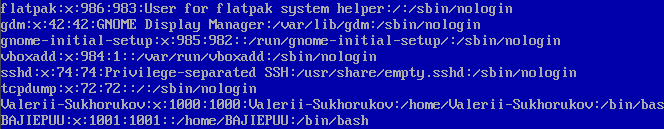


Рис 17. Фрагмент файла passwd

Файл **/etc/shadows** доступен для чтения только суперпользователю и содержит зашифрованную информацию о паролях.

Каждая строка определяет информацию о пароле конкретного аккаунта, поля в ней разделены знаком ":".

* Первое поле определяет конкретный пользовательский аккаунт, которому соответствует данная «теневая» запись.
* Поле 2— содержит зашифрованный пароль.
* Поле 3 — количество дней с 01.01.1970 до момента, когда пароль был изменен
* Поле 4 — количество дней до того, как будет разрешено сменить пароль («0» — «менять в любое время»)
* Поле 5 — количество дней до того, как система заставит пользователя сменить пароль ("-1" — «никогда»)
* Поле 6 — количество дней до истечения срока действия пароля, когда пользователь получит предупреждение об этом ("-1" — «не предупреждать»)
* Поле 7 — количество дней после истечения срока действия пароля, по прошествии которых аккаунт будет автоматически отключен системой ("-1" — «не отключать»)
* Поле 8 — количество дней, прошедшее с момента отключения этого аккаунта ("-1" — «этот аккаунт включен»)
* Поле 9 — зарезервировано для будущего использования

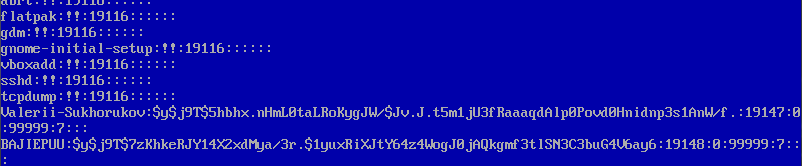


Рис 18. Фрагмент файла shadows

### Псевдопользователи

Для каждой версии UNIX существуют записи файла паролей для нескольких псевдопользователей. Эти записи не подлежат редактированию. Такие пользователи без права входа в систему располагают соответствующими процессами для каждого аспекта, связанного с принадлежностью системы.

Список наиболее распространенных псевдопользователей:

* **daemon** — Используется серверными процессами системы.
* **bin** —Владеет исполняемыми файлами пользователя.

Файлы и процессы, которые являются частью операционной системы, но не должны принадлежать пользователю root, иногда передаются во владение пользователям bin или daemon. Считается, что это поможет избежать риска, связанного с действиями от имени суперпользователя.

* **sys** —Владеет системными файлами (образами памяти ядра), которые хранятся в каталоге /dev. Доступ к этим файлам имеют лишь немногие программы, и все они изменяют текущий идентификатор пользователя на sys. Иногда вместо пользователя sys создается группа kmem или sys.
* **nobody** —Используется для NFS(Network File System). Чтобы лишить суперпользователей их исключительных прав, излишних в данном контексте, NFS должна на время сеанса удаленного доступа заменить нулевой идентификатор чем-то другим. Этой цели как раз и служит учетная запись nobody. В среде сервера NFSv4 эта учетная запись также применяется для удаленных пользователей, которые не имеют допустимые локальные учетные записи. Пользователю nobody не нужны специальные права доступа, и он не должен владеть никакими файлами. Если бы ему принадлежали какие-то файлы, то над ними получили бы контроль суперпользователи, регистрирующиеся в удаленных системах.

## Работа с устройствами и файловой системой.

### Специальные файлы устройств

Для Linux нет разницы между устройством и файлом. Все устройства системы представлены в корневой файловой системе в виде обычных файлов. Файлы устройств хранятся в каталоге /dev.

В Unix-подобных операционных системах файл устройства или специальный файл - это интерфейс к драйверу устройства, который появляется в файловой системе, как если бы это был обычный файл. Эти специальные файлы позволяют прикладной программе взаимодействовать с устройством, используя его драйвер устройства через стандартные системные вызовы ввода / вывода. Использование стандартных системных вызовов упрощает многие задачи программирования и приводит к согласованным механизмам ввода-вывода пользовательского пространства независимо от особенностей и функций устройства.

Файлы устройств обычно предоставляют простые интерфейсы для стандартных устройств (таких как принтеры и последовательные порты), но также могут использоваться для доступа к определенным уникальным ресурсам на этих устройствах, таким как разделы диска. Кроме того, файлы устройств полезны для доступа к системным ресурсам, которые не связаны с каким-либо фактическим устройством, таким как приемники данных и генераторы случайных чисел.

Все драйверы ядра пронумерованы главными (мажорными, major) числами , а аппаратные устройства, находящиеся под их управлением, — дополнительными (минорными, minor) числами. Основной характеристикой специальных файлов устройств является пара чисел major, minor (иногда называемых характеристическими числами), привязывающая их к конкретному драйверу и управляемому им устройству.

### Использование файлов дисковых устройств в командах управления

Специальными файлами дисковых устройств пользуются программы, управляющие структурами самого носителя, например таблицами разделов fdisk и parted или файловыми системами разделов носителя mount, fsck, mkfs и пр.

**fdisk** — это программа для создания таблиц разделов и управления ими. Она понимает таблицы разделов GPT, MBR, Sun, SGI и BSD.

Выведем список разделов на диске /dev/sda. Для этого используем ключ –l (Рис 19). На диске находится два раздела – загрузочный, на который «ссылается» файл /dev/sda1, и основной – /dev/sda2

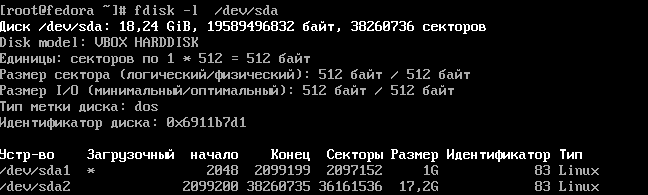


Рис 19. Список разделов на диске

С помощью команды mount можно подключить сетевой диск, раздел жесткого диска или USB-накопитель. Смонтируем USB накопитель в каталог /mount. Для этого выполним команду fdisk –l (Рис 20), чтобы отобразить список подключенных устройств, и найдем название USB-флэш-накопителя. После этого выполним команду mount –v /dev/sdb4 /mnt/ (Ключ v позволяет отображать состояние процесса) (Рис 21). Просмотрим список смонтированных устройств командой mount без параметров (Рис 22).

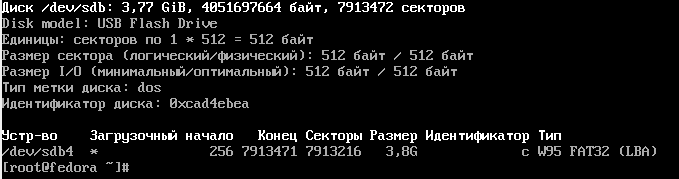


Рис 20. Список разделов на /dev/sdb



Рис 21. Монтирование диска



Рис 22. Свойства смонтированного диска

### Аппаратная конфигурация компьютера

С помощью команд lsblk, lspci, lscpu можно узнать аппаратную конфигурацию.

**Команда lsblk** – выводит на экран информацию обо всех блочных устройствах, таких как жесткие диски, приводы DVD и прочее:

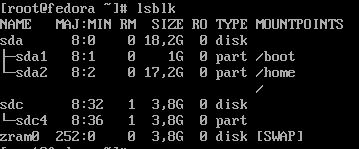


Рис 23. Подключенные дисковые устройства

**Команда lscpu** – выводит информацию о CPU:

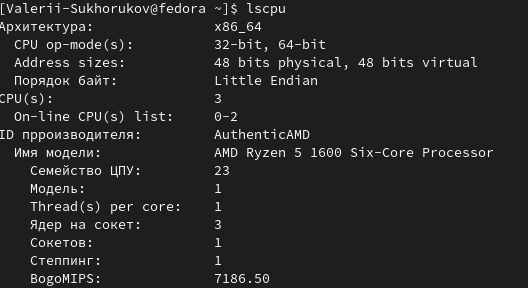


Рис 24. Информация о процессоре

**Команда lspci** – выводит информацию обо всех устройствах, присоединённых через pci -интерфейс:

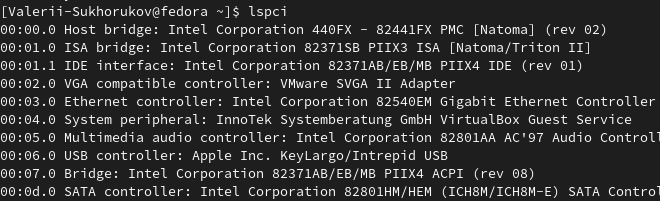


Рис 25. Информация о подключенных устройствах

### Распределение аппаратных ресурсов

Файлы ioports (Рис 26), interrupts (Рис 27), iomem (Рис 28), расположенные в каталоге /proc, хранят информация и распределении соответствующих аппаратных ресурсов. Составим таблицы распределения аппаратных ресурсов и сопоставим с аналогичными таблицами из второй лабораторной работы.

|  |  |
| --- | --- |
| ***Адреса портов ввода/вывода (Виртуальная машина Windows)*** | |
| ***Выделенные адреса*** | ***Устройство*** |
| [00000000-0000000F] | Контроллер прямого доступа к памяти №1 |
| [00000000-00000CF7] | Шина PCI |
| [00000020-00000021] | Программируемый контроллер прерываний №1 |
| [00000040-00000043] | Системный таймер |
| [00000060-00000060] | Стандартная клавиатура PS/2 |
| [000003B0-000003BB] | Virtual Box Graphic Adapter |
| [0000D000-0000D00F] | Intel (R) PCI BUS Master IDE-контроллер |
| [0000D020-0000D027] | Intel (R) PRO/1000 MT |

|  |  |
| --- | --- |
| ***Адреса портов ввода/вывода (Виртуальная машина Linux)*** | |
| ***Выделенные адреса*** | ***Устройство*** |
| [0000-00F1] | Контроллер прямого доступа к памяти №1 (dma1) |
| [0020-0021] | Программируемый контроллер прерываний №1(pic1) |
| [0060-0060] | Клавиатура (keyboard) |
| [0070-0071] | Часы реального времени (rtc\_cmos) |
| [03C0-03DF] | Графический адаптер (vga+) |
| [0D000-D00F] | Контроллер IDE (ata\_piix) |
| [D100- D1FF] | Контроллер ввода-вывода (Itel 8281aa - ICH) |
| [D020-D027] | Эмулируемый сетевой гигабитный контролер (e1000) |

На двух виртуальных машинах совпадают адреса контроллера прямого доступа к памяти №1 [0-F], контроллера прерываний №1 [20-21], клавиатуры [60], Контроллера дисков [0D000-D00F], сетевого адаптера [D020-D027]. Различаются адреса системного таймера windows - [40-43], Linux - [70-71]; графического адаптера windows - [3B0-3BB],Linux -[3С0-3DF].

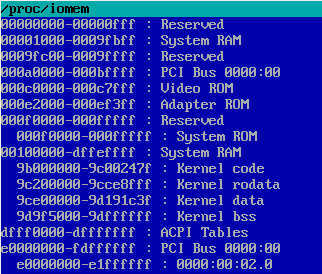
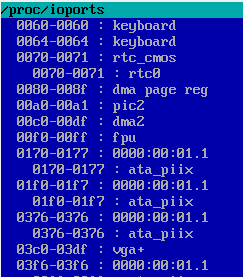


Рис 26. Фрагмент файла ioports Рис 28. Фрагмент файла iomem

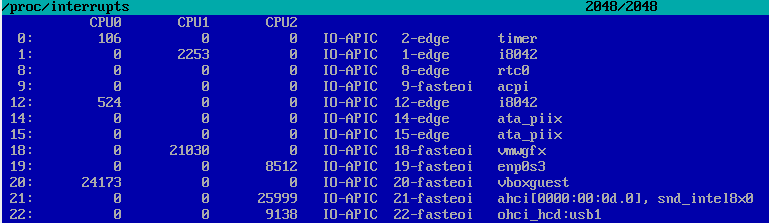


Рис 27. Фрагмент файла interrupts

|  |  |
| --- | --- |
| ***Номера запросов на прерывание (Виртуальная машина Windows )*** | |
| ***Номер прерывания*** | ***Устройство*** |
| 0 | Системный таймер |
| 1 | Стандартная клавиатура PS/2 |
| C h (12d) | Microsoft PS/2 мышь |
| E h (14d) | ATA Chanel 0 |
| 12 h (18 d) | Virtual Box Graphic Adapter |
| 13 h (19 d) | Intel (R) PRO/1000 MT |
| 15 h (21d) | Контроллер High Definition Audio (Microsoft) |
| 16 h (22d) | USB хост- контроллер |

|  |  |
| --- | --- |
| ***Номера запросов на прерывание (Виртуальная машина Linux )*** | |
| ***Номер прерывания*** | ***Устройство*** |
| 0d | Системный таймер |
| 1d | Драйвер PS/2 (i8042) |
| 12d | Драйвер PS/2 (i8042) |
| 14d | Контроллер дисков (ata\_piix) |
| 18d | Графический адаптер (vmgfx) |
| 19d | Сетевой интерфейс (enp0s3) |
| 21d | Звуковой контроллер (snd\_intel8x0) |
| 22d | USB контроллер (ochi\_hcd:usb1) |

На двух виртуальных машинах полностью совпадают номера прерываний от внешних устройств.

|  |  |
| --- | --- |
| ***Диапазоны адресов памяти (Виртуальная машина Windows)*** | |
| ***Диапазон адресов*** | ***Устройство*** |
| [0000A000- F01FFFFF] | Virtual Box Graphic Adapter |
| [F0200000-F021FFFF] | Intel (R) PRO/1000 MT |
| [F0840000-F087FFFF] | Контроллер High Definition Audio (Microsoft) |
| [F0808000-F0808FFF] | USB хост- контроллер |
| [F080A000-F080BFFF] | Стандартный контроллер SATA AHCI |

|  |  |
| --- | --- |
| ***Диапазоны адресов памяти (Виртуальная машина Linux)*** | |
| ***Диапазон адресов*** | ***Устройство*** |
| [C0000- C7FFF] | Video ROM |
| [F0200000-F021FFFF] | Эмулируемый сетевой гигабитный контролер (e1000) |
| [F0400000-F07FFFFF] | Звуковой контроллер (vboxguest) |
| [F0804000-F0804FFF] | USB контроллер (ochi\_hcd) |
| [F0806000-F0807FFF] | Контроллер SATA AHCI |

Порядок распределения памяти для внешних устройств одинаков на двух машинах, у некоторых устройств (USB контроллер, SATA контроллер) отличается диапазон адресов.

## Управление файлами

* **Ls** - утилита Unix, которая печатает в стандартный вывод содержимое каталогов. 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. Для задания одноколоночного или многоколоночного режима вывода могут использоваться опции -1 и -C, соответственно.

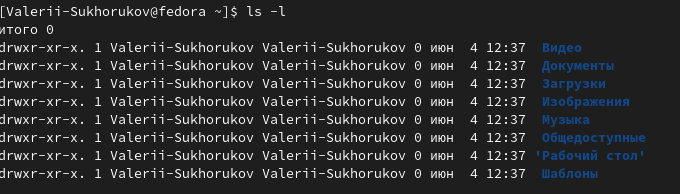


Рис 29. Использование утилиты ls

* **Cd** - команда командной строки для изменения текущего рабочего каталога в UNIX, DOS и других операционных системах. Она также доступна для использования в скриптах командного интерпретатора или в пакетных файлах.
* **Cp** - команда Unix в составе GNU Coreutils, предназначенная для копирования файлов из одного в другие каталоги (возможно, с другой файловой системой). Исходный файл остаётся неизменным, имя созданного файла может быть таким же, как у исходного, или изменится.
* **echo** – выводит строку в терминал.
* **Pwd** — консольная утилита в UNIX-подобных системах, которая выводит полный путь от корневого каталога к текущему рабочему каталогу: в контексте которого (по умолчанию) будут исполняться вводимые команды.



Рис 30. Использование утилиты pwd

* **Mv** - утилита в UNIX и UNIX-подобных системах, используется для перемещения или переименования файлов.

Если в качестве аргументов заданы имена двух файлов, то имя первого файла будет изменено на имя второго.

Если последний аргумент является именем существующего каталога, то mv перемещает все заданные файлы в этот каталог (mv file ./dir/ )

* + **Rm** - утилита в UNIX и UNIX-подобных системах, используемая для удаления файлов из файловой системы
  + **cat** — одна из наиболее часто используемых команд в Linux. Она считывает данные из файлов и выводит их содержимое.
* **Ln** - команда UNIX, устанавливающая связь между файлом и именем файла.

ln файл имя\_ссылки # создаётся «жёсткая» ссылка (hard link)

ln -s файл имя\_ссылки # создаётся «символьная» ссылка (symbolic link).

На рисунке 31 представлен пример использования команд для создания текстового файла, и символьной ссылки на него.

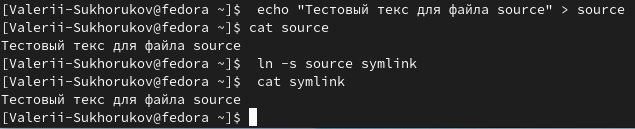


Рис 31. Пример использования команд работы с файлами

## Управление доступом к файловой системе.

### Состав индексного дескриптора файла.

Каждому файлу на диске соответствует один и только один индексный дескриптор файла, который идентифицируется своим порядковым номером - индексом файла. Это означает, что число файлов, которые могут быть созданы в файловой системе, ограничено числом индексных дескрипторов, которое либо явно задается при создании файловой системы, либо вычисляется исходя из физического объема дискового раздела. Индексный дескриптор файла имеет следующее строение:

|  |  |
| --- | --- |
| Название поля | Описание |
| di\_mode | Тип и права доступа к данному файлу |
| di\_uid | Идентификатор владельца файла (Owner UID) |
| di\_size | Размер файла в байтах |
| di\_atime | Время последнего обращения к файлу (Access time) |
| di\_ctime | Время создания файла |
| di\_mtime | Время последней модификации файла |
| di\_gid | Идентификатор группы (GID) |

Как видно из таблицы, индексный дескриптор каждого файла содержит два важных поля: идентификатор владельца файла (di\_uid) и идентификатор группы владельцев файлов (di\_gid). Каждый пользователь или группа пользователей имеют определенные права по отношению к файлу или каталогу, владельцы файла, как правило, имеют более широкие права. Владельцами файла обычно становятся пользователи, которые создают данный файл.

### Определение владельца-пользователя и владельца группы

Существуют специальные команды, позволяющие явно определить владельца-пользователя или владельца-группу для файла:

* **chown** - команда, позволяющая изменить владельца-пользователя для файла: sudo chown -v user file.txt - смена владельца-пользователя для файла file.txt на пользователя user с выводом диагностического сообщения, которое подтвердило смену пользователя и указало имя старого пользователя.



Рис 32. Пример использования команды chow

* **chgrp** - команда, позволяющая изменить владельца-группу для файла: sudo chgrp -R –v group ex - смена группы-владельца на группу group для директории ex с рекурсивной сменой группы-владельца для всего содержимого директории ex.



Рис 33. Пример использования команды chgrp

### Режимы доступа для файла и каталога

В операционных системах Linux используются 3 базовых права доступа – на чтение (read), запись (write) и исполнение (execute). Соответственно, права назначаются пользователю (user), группе (group) и всем остальным (world).

Для того, чтобы позволить обычным пользователям выполнять программы от имени суперпользователя без знания его пароля была придумана такая вещь, как SUID и SGID биты. Рассмотрим эти полномочия:

* **SUID** - если этот бит установлен, то при выполнении программы, id пользователя, от которого она запущена заменяется на id владельца файла. Фактически, это позволяет обычным пользователям запускать программы от имени суперпользователя;
* **SGID** - этот флаг работает аналогичным образом, только разница в том, что пользователь считается членом группы, с которой связан файл, а не групп, к которым он действительно принадлежит. Если SGID флаг установлен на каталог, все файлы, созданные в нем, будут связаны с группой каталога, а не пользователя. Такое поведение используется для организации общих папок;
* **Sticky-bit** - этот бит тоже используется для создания общих папок. Если он установлен, то пользователи могут только создавать, читать и выполнять файлы, но не могут удалять файлы, принадлежащие другим пользователям.

Для управления правами доступа к файлам и папкам используется команда **chmod**, которая позволяет задать права доступа к тому или иному файлу или папке для пользователя-владельца, группы-владельца и остальных пользователей. В качестве аргумента данная команда может принимать список прав доступа в символьном виде или в виде числовых эквивалентов:

* sudo chmod -v 755 1.txt - данная команда устанавливает для пользователя-владельца права 7 (rwx), для группы-владельца - 5(r-х), для остальных пользователей - 5(r-х).



Рис 34. Пример использования команды chmod

Рассмотрим подробнее, что значат условные значения флагов прав:

* **---** - нет прав, совсем;
* **--x** - разрешено только выполнение файла, как программы, но не изменение и не чтение;
* **-w-** - разрешена только запись и изменение файла;
* **-wx** - разрешено изменение и выполнение, но в случае с каталогом, вы не можете посмотреть его содержимое;
* **r--** - права только на чтение;
* **r-x** - только чтение и выполнение, без права на запись;
* **rw-** - права на чтение и запись, но без выполнения;
* **rwx** - все права;
* **--s** - установлен SUID или SGID бит, первый отображается в поле для владельца, второй для группы;
* **--t** - установлен sticky-bit, а значит, пользователи не могут удалить этот файл.

Чтобы узнать права доступа выполним команду ls –l:

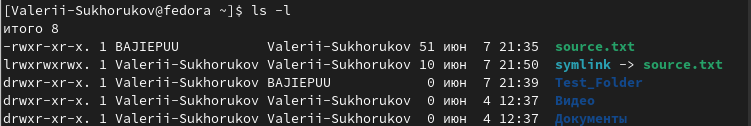


Рис 35. Просмотр свойств файлов и каталогов

Назначим всем пользователям полный доступ к Test\_Folder, для файла source.txt установим SUID:

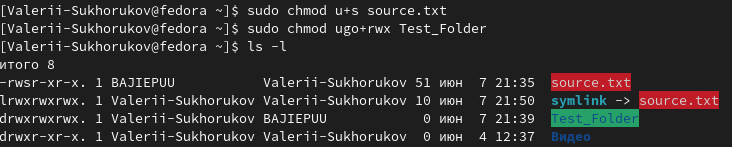


Рис 36. Изменение доступа к файлу и папке

Изменение прав доступа к самой символической ссылке не имеет никакого смысла. Манипуляции с правами доступа к символической ссылке «прозрачно транслируются» на объект файловой системы, на который она ссылается.

Можем сделать вывод, что дополнительные флаги SUID, SGID, StikyBit для файла и каталога позволяют изменить права. С помощью команды «ls -l» можно увидеть права на файл или каталог, с использованием флагов прав. Также были изучены состав и свойства индексного дескриптора файла, команды для изменения владельцев файлов, варианты доступа к файлам и папкам, была применена команда назначения доступа.

## Управление заданиями

Всякая выполняющаяся в Linux программа называется процессом. Linux как многозадачная система характеризуется тем, что одновременно может выполняться множество процессов, принадлежащих одному или нескольким пользователям.

Задания могут быть либо на переднем плане (foreground), либо фоновыми (background). На переднем плане в любой момент времени может быть только одно задание. Задание на переднем плане — это то задание, с которым вы взаимодействуете; оно получает ввод с клавиатуры и посылает вывод на экран (если, разумеется, вы не перенаправили ввод или вывод куда-либо ещё).

Между заданиями в фоновом режиме и приостановленными заданиями есть большая разница. Приостановленное задание не работает — на него не тратятся вычислительные мощности процессора. Это задание не выполняет никаких действий. Приостановленное задание занимает некоторый объем оперативной памяти компьютера, через некоторое время ядро откачает эту часть памяти на жёсткий диск «до востребования». Напротив, задание в фоновом режиме выполняется, использует память и совершает некоторые действия, которые, возможно, вам требуются, но вы в это время можете работать с другими программами.

* jobs - команда, которая выводит список фоновых процессов, запущенных в данном терминале
* ps - команда, позволяющая вывести список всех запущенных процессов:

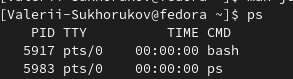


Рис 37. Использование команды ps

* stop, - команда, позволяющая приостановить работу процесса.
* Ctrl+Z - сочетание клавиш, позволяющее остановить выполнение процесса. Данное сочетание клавиш часто необходимо в случае запуска длительных по времени выполнения программ в оперативном режиме, так как в отличие от команды stop, это сочетание клавиш может прервать выполнение программы, не дожидаясь освобождения терминала.
* nice - команда позволяющая запустить процесс с заданным приоритетом. Приоритет задается целыми числами в диапазоне: -20 (высокий) - 19 (низкий).
* at, - команда, позволяющая запустить определенную задачу в указанное время.
* top - команда, выводящая на экран информацию о всех процессах, запущенных в системе: идентификатор процесса, пользователь, запустивший процесс, использование процессом системных ресурсов, время работы процесса, команда, запустившая процесс.
* fg,bg - команды, позволяющие осуществлять переключение между фоновым и оперативным режимами выполнения программы прямо во время выполнения.
* kill - команда, позволяющая принудительно завершить выполнение процесса.

Создадим процесс sleep 100 (Рис 37). Используем команду top (Рис 38), для получения PID процесса. Воспользуемся утилитой kill (Рис 39) для завершения процесса.



Рис 37.

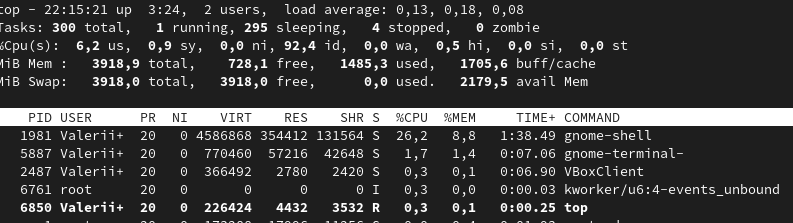


Рис 38.



Рис 39.

## Базовая файловая структура Linux

В разных дистрибутивах Linux каталоги тоже могут быть разные: в некоторых будут дополнительные файлы/каталоги конфигурации, а некоторые файлы конфигурации, вполне возможно, будут называться как-то по-другому. Приведем названия каталогов, которые входят в дистрибутив Fedora.

Основные каталоги корневой файловой системы:

* / — корневой каталог
* /bin — содержит стандартные утилиты Linux, основные исполняемые файлы, доступные всем пользователям, а также содержит символьные ссылки на исполняемые файлы.
* /boot — содержит конфигурационные файлы загрузчика в каталоге grub, образы ядра, файлы Initrd.
* /dev — содержит файлы устройств. Большинство устройств в Linux представляют из себя файлы в особой файловой системе. И вот эти файлы хранятся в каталоге /dev, куда к ним обращается система для выполнения задач, связанных с вводом/выводом.
* /etc — содержит конфигурационные файлы операционной системы и всех сетевых служб. В Linux общесистемные настройки хранятся в разных конфигурационных файлах, которые можно редактировать обычным текстовым редактором.
* /home — домашний каталог пользователей. Домашний каталог пользователя обозначается /home/Имя\_Пользователя.
* /lib — здесь находятся различные библиотеки и модули ядра. В процессе установки различных программ устанавливаются зависимости для корректной работы программы. Вот эти зависимости — набор собранных особым образом файлов, которые подключаются во время установки к устанавливаемой программе.
* /mnt и /media — обычно в этих каталогах содержатся точки монтирования. В современных дистрибутивах Linux этот процесс обычно происходит автоматически. При этом в каталогах /mnt или /media создается подкаталог, имя которого совпадает с именем монтируемого тома.
* /opt — здесь обычно размещаются установленные программы, имеющие большой дисковый объем, или вспомогательные пакеты.
* /proc — это каталог псевдофайловой системы procfs, которая используется для предоставления информации о процессах. В системе Linux присутствует виртуальный файловый объект, именуемый каталогом /proc. Он существует только во время работы системы в оперативной памяти компьютера. Каталог представляет интерес и с точки зрения безопасности. Многие из утилит, выводящие информацию о системе (например, команда ps), берут свои исходные данные именно из этого каталога.
* /root — каталог пользователя root. Для доступа нужно предварительно пройти аутентификацию.
* /run — это совершенно новый каталог для хранения данных, которые были запущены приложениями, требующимися в процессе работы.
* /sbin — набор утилит для системного администрирования, содержит исполняемые файлы, необходимые для загрузки системы и ее восстановления. Запускать эти утилиты имеет право только root.
* /tmp — каталог, в котором хранятся временные файлы. Linux, в отличие от Windows, следит за чистотой и регулярно очищает этот каталог.
* /usr — содержит пользовательские программы, документацию, исходные коды программ и ядра. По размеру это один из самых больших каталогов файловой системы. В этот каталог устанавливаются практически все программы.
* /var — содержит файлы, которые подвергаются наиболее частому изменению.

Файловая система /proc является механизмом для ядра и его модулей, позволяющим посылать информацию процессам (отсюда и название /proc). С помощью этой виртуальной файловой системы Мы можем работать с внутренними структурами ядра, получать полезную информацию о процессах и изменять установки (меняя параметры ядра).

* /proc/cpuinfo - информация о процессоре (модель, семейство, размер кэша и т.д.)
* /proc/meminfo - информация о RAM, размере свопа и т.д.
* /proc/mounts - список подмонтированных файловых систем.
* /proc/devices - список устройств.
* /proc/filesystems - поддерживаемые файловые системы.
* /proc/modules - список загружаемых модулей.
* /proc/version - версия ядра.
* /proc/cmdline - список параметров, передаваемых ядру при загрузке.
* /proc/self подкаталог - с его помощью приложение найдет информацию о себе. /proc/self является символической ссылкой на каталог процесса обращающегося к /proc.
* В каталоге /proc/sys/kernel находится информация общего плана для ядра. Соответственно в /proc/sys/kernel/{domainname, hostname} находится информация о доменном имени и host имени, которую пользователь может изменить.

# Вывод

В данной лабораторной работе мы установили ОС linux, зарегистрировалась в системе. Научились пользоваться диалоговым руководством. Ознакомились со справочной системой в формате HOW-TO. Научились управлять учетными записями пользователя на данной ОС. Совершили работу с устройствами и файловой системой. Управляли некоторыми файлами, совершили операции над группами файлов, совершили работу с командами вывода, поиска и фильтра. Описали и применили функции управления доступом к файловой системе. Научились управлять заданиями. Описали базовую структуру Linux.