



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:

(подпись)

Кочешков А. А
(фамилия, и.,о.)

СТУДЕНТ:

(подпись)

Сухоруков В. А
(фамилия, и.,о.)

19-В-2
(шифр группы)

Работа защищена «__» _____
С оценкой _____

Нижний Новгород 2022

Оглавление

Цель работы	3
Ход работы.....	3
1. Регистрация в системе.....	3
2. Основные информационные команды.....	4
1) Руководство man	4
2) Команда apropos	5
3) Файлы HOWTO	5
3. Управление учетными записями пользователей.	6
1) Получение информации о пользователях	6
Команда who	6
Команда w	6
Команда id.....	6
Команда logname	7
Команда finger	7
2) Создание новой учетной записи	7
3) Файлы /etc/passwd и /etc/shadows.....	8
4) Псевдопользователи.....	9
4. Работа с устройствами и файловой системой.	10
1) Специальные файлы устройств	10
2) Использование файлов дисковых устройств в командах управления	10
3) Аппаратная конфигурация компьютера.....	11
4) Распределение аппаратных ресурсов	12
5. Управление файлами	15
6. Управление доступом к файловой системе.....	16
1) Состав индексного дескриптора файла.....	16
2) Определение владельца-пользователя и владельца группы	16
3) Режимы доступа для файла и каталога	17
7. Управление заданиями	19
8. Базовая файловая структура Linux	20
Вывод.....	22

Цель работы

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

Ход работы

1. 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

```
Fedora Linux 36 (Workstation Edition)
Kernel 5.17.5-300.fc36.x86_64 on an x86_64 (tty2)

fedora login: Valerii-Sukhorukov
Password:
Last login: Sat Jun  4 12:51:33 on tty2
[Valerii-Sukhorukov@fedora ~]$ _
```

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).

```
Fedora Linux 36 (Workstation Edition)
Kernel 5.17.5-300.fc36.x86_64 on an x86_64 (tty2)

fedora login: root
Password:
Last login: Sat Jun  4 13:00:35 on tty2
[root@fedora ~]#
```

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.

2. Основные информационные команды

1) 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`.

```
PASSWD(5)                                Руководство программиста Linux                                PASSWD(5)

ИМЯ
    passwd - файл паролей

ОПИСАНИЕ
    В текстовом файле /etc/passwd содержится список учётных записей пользователей в системе.
    Файл должен быть доступен для чтения всем пользователям (многие утилиты, такие как ls(1),
    используют его, чтобы преобразовывать идентификаторы пользователей в их имена), но доступ
    на запись должен предоставляться только суперпользователю.

    В старые добрые времена не существовало большой проблемы, связанной с доступностью этого
    файла на чтение для всех. Любой мог прочитать зашифрованные пароли, но мощности
    компьютеров не хватало для подбора грамотно выбранных паролей, а кроме того, наивно
    предполагалось, что сеть используется дружественным сообществом пользователей. В наши дни,
    многие пользователи стали использовать механизм теневых паролей, где файл /etc/passwd
    содержит 'x' в поле пароля, а зашифрованные пароли хранятся в файле /etc/shadow, который
    доступен на чтение только суперпользователю.
```

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 <code>/dev</code>) 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`.

```
PASSWD(1)                                User utilities                                PASSWD(1)

NAME
    passwd - update user's authentication tokens

SYNOPSIS
    passwd [-k] [-l] [-u [-f]] [-d] [-e] [-n mindays] [-x maxdays] [-w warndays] [-i inactive-
    days] [-S] [--stdin] [-?] [--usage] [username]

DESCRIPTION
    The passwd utility is used to update user's authentication token(s).

    This task is achieved through calls to the Linux-PAM and Libuser API. Essentially, it
    initializes itself as a "passwd" service with Linux-PAM and utilizes configured password
    modules to authenticate and then update a user's password.
```

Рис 4. Справка о команде `passwd`

2) Команда apropos

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

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

```
(Valerii-Sukhorukov@fedora ~)$ apropos -s 1 install
abrt-action-analyze-vmcore (1) - Extracts the oops message from the kernel core and install the k
abrt-action-install-debuginfo (1) - Install debuginfos for build-ids
desktop-file-edit (1) - Installation and edition of desktop files
desktop-file-install (1) - Installation and edition of desktop files
extlinux (1)
  - install the SYSLINUX bootloader on a ext2/ext3 filesystem
flatpak (1)
  - Build, install and run applications and runtimes
flatpak-info (1)
  - Show information about an installed application or runtime
flatpak-install (1)
  - Install an application or runtime
flatpak-list (1)
  - List installed applications and/or runtimes
flatpak-mask (1)
  - Mask out updates and automatic installation
flatpak-repair (1)
  - Repair a flatpak installation
flatpak-uninstall (1)
  - Uninstall an application or runtime
gnome-software (1)
  - Install applications
install (1)
  - copy files and set attributes
```

Рис 5. Поиск информации с помощью команды `apropos`

Посмотрим подробную справку о команде `flatpak` командой `man 1 flatpak`:

```
FLATPAK(1)                                flatpak                                FLATPAK(1)
NAME
  flatpak - Build, install and run applications and runtimes
SYNOPSIS
  flatpak [OPTION...] {COMMAND}
DESCRIPTION
  Flatpak is a tool for managing applications and the runtimes they use. In the Flatpak
  model, applications can be built and distributed independently from the host system they
  are used on, and they are isolated from the host system ('sandboxed') to some degree, at
  runtime.
```

Рис 6. Справка команды `flatpak`

3) 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>

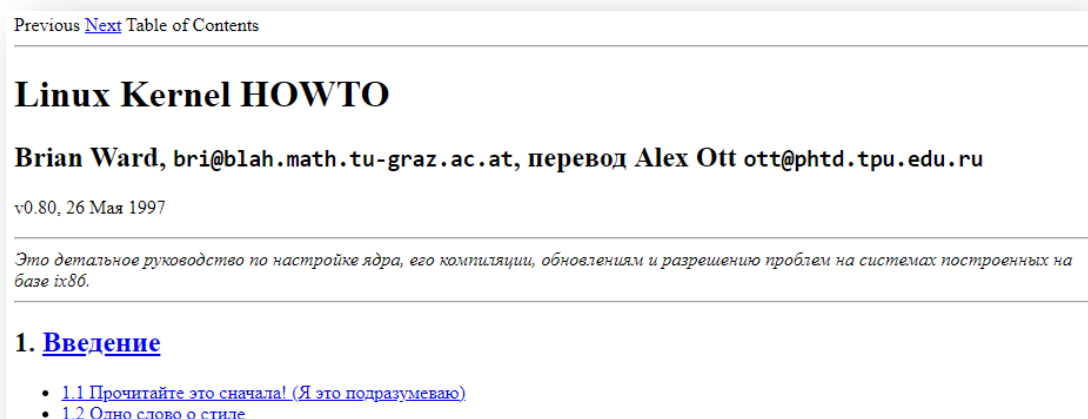


Рис 7. Пример HOWTO файла

3. Управление учетными записями пользователей.

1) 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.

```
[Valerii-Sukhorukov@fedora ~]$ who -H
ИМЯ  ЛИНИЯ  ВРЕМЯ      КОММЕНТАРИЙ
Valerii-Sukhorukov tty2      2022-06-04 13:51
```

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.

```
[Valerii-Sukhorukov@fedora ~]$ w
12:21:49 up 1:22, 1 user, load average: 0,02, 0,06, 0,03
USER      TTY      LOGIN@  IDLE   JCPU   PCPU WHAT
Valerii-  tty2          C613    1.00s  0.13s  0.00s w
```

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.

```
[Valerii-Sukhorukov@fedora ~]$ id
uid=1000(Valerii-Sukhorukov) gid=1000(Valerii-Sukhorukov) группы=1000(Valerii-Sukhorukov),10(wheel)
```

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.

```
[Valerii-Sukhorukov@fedora ~]$ logname
Valerii-Sukhorukov
```

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.

```
[Valerii-Sukhorukov@fedora ~]$ finger -s
Login      Name      Tty      Idle   Login Time   Office      Office Phone   Host
Valerii-Sukhorukov Valerii-Sukh tty2                Jun  4 13:51
```

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.

```
[Valerii-Sukhorukov@fedora ~]$ finger -l
Login: Valerii-Sukhorukov                      Name: Valerii-Sukhorukov
Directory: /home/Valerii-Sukhorukov           Shell: /bin/bash
On since Сб июн  4 13:51 (MSK) on tty2      2 seconds idle
No mail.
No Plan.
```

Рис 13. Использование команды `finger -l`

2) 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:

```
[root@fedora ~]# adduser БАЖИЕПУУ
[root@fedora ~]# mc
```

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.

Левая панель	Файл	Команда	Настройки
<code>< /home/BAJIEPUU . [^] ></code>			
<code>.и</code>	Имя	Размер	Время правки
<code>/..</code>		-ВВЕРХ-	июн 5 12:55
<code>/.mozilla</code>		34	янв 20 23:37
<code>.bash_logout</code>		18	янв 20 01:12
<code>.bash_profile</code>		141	янв 20 01:12
<code>.bashrc</code>		492	янв 20 01:12

Рис 15. Домашний каталог пользователя

Установим пароль для учетной записи командой `passwd`:

```
root@fedora home1# passwd BAJIEPUU
Изменение пароля пользователя BAJIEPUU.
Новый пароль:
Повторите ввод нового пароля:
passwd: данные аутентификации успешно обновлены.
```

Рис 16.

3) 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;
- ❖ **x**: 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.


```
flatpak:x:986:983:User for flatpak system helper:/:/sbin/nologin
gdm:x:42:42:GNOME Display Manager:/var/lib/gdm:/sbin/nologin
gnome-initial-setup:x:985:982:/:run/gnome-initial-setup:/sbin/nologin
vboxadd:x:984:1:/:var/run/vboxadd:/sbin/nologin
sshd:x:74:74:Privilege-separated SSH:/usr/share/empty.sshd:/sbin/nologin
tcpdump:x:72:72:/:/sbin/nologin
Valerii-Sukhorukov:x:1000:1000:Valerii-Sukhorukov:/home/Valerii-Sukhorukov:/bin/bash
BAJIEPUU:x:1001:1001:/:home/BAJIEPUU:/bin/bash
```

Рис 17. Фрагмент файла passwd

Файл `/etc/shadow` 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

```
flatpak:!!:19116:!!!!:
gdm:!!:19116:!!!!:
gnome-initial-setup:!!:19116:!!!!:
vboxadd:!!:19116:!!!!:
sshd:!!:19116:!!!!:
tcpdump:!!:19116:!!!!:
Valerii-Sukhorukov:$y$j9T$5hbhx.nHmL0taLRoKyGjW/$Jv.J.t5m1jU3fRaaqdAlp0Povd0Hnidnp3s1AmW/f.:19147:0:99999:7:::
BAJIEPUU:$y$j9T$7zKhkeRJY14XZxdMya/3r.$1yuxRiXJtY64z4WogJ0.jAQkgmf3t1SN3C3buG4U6ay6:19148:0:99999:7:::
```

Рис 18. Фрагмент файла shadow

4) 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.

4. Работа с устройствами и файловой системой.

1) 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.

2) 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

```
[root@fedora ~]# fdisk -l /dev/sda
Диск /dev/sda: 18,24 GiB, 19589496832 байт, 38260736 секторов
Disk model: VBOX HARDDISK
Единицы: секторов по 1 * 512 = 512 байт
Размер сектора (логический/физический): 512 байт / 512 байт
Размер I/O (минимальный/оптимальный): 512 байт / 512 байт
Тип метки диска: dos
Идентификатор диска: 0x6911b7d1

Устр-во    Загрузочный  начало      Конеч      Секторы    Размер    Идентификатор  Тип
/dev/sda1  *            2048        2099199    2097152     1G        83 Linux
/dev/sda2                                2099200    38260735    36161536    17,2G     83 Linux
```

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 /mnt. 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).

```
Диск /dev/sdb: 3,77 GiB, 4051697664 байт, 7913472 секторов
Disk model: USB Flash Drive
Единицы: секторов по 1 * 512 = 512 байт
Размер сектора (логический/физический): 512 байт / 512 байт
Размер I/O (минимальный/оптимальный): 512 байт / 512 байт
Тип метки диска: dos
Идентификатор диска: 0xcad4ebee

Устр-во    Загрузочный  начало      Конеч      Секторы    Размер    Идентификатор  Тип
/dev/sdb4  *            256         7913471    7913216     3,8G      c W95 FAT32 (LBA)

[root@fedora ~]#
```

Рис 20. Список разделов на /dev/sdb

```
[root@fedora ~]# mount -v /dev/sdb4 /mnt/
mount: /dev/sdb4 mounted on /mnt.
[root@fedora ~]# _
```

Рис 21. Монтирование диска

```
/dev/sdb4 on /mnt type vfat (rw,relatime,fmask=0022,dmask=0022,codepage=437,ioccharset=ascii,shortname=mixed,errors=remount-ro)
```

Рис 22. Свойства смонтированного диска

3) 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:

```
[root@fedora ~]# lsblk
NAME MAJ:MIN RM SIZE RO TYPE MOUNTPOINTS
sda   8:0    0 18,2G 0 disk
├─sda1 8:1    0    1G 0 part /boot
└─sda2 8:2    0 17,2G 0 part /home
sdc   8:32   1  3,8G 0 disk
└─sdc4 8:36   1  3,8G 0 part
zram0 252:0   0  3,8G 0 disk [SWAP]
```

Рис 23. Подключенные дисковые устройства

Команда lscpu – выводит информацию о CPU:

```
[Valerii-Sukhorukov@fedora ~]$ lscpu
Архитектура:          x86_64
  CPU op-mode(s):      32-bit, 64-bit
  Address sizes:        48 bits physical, 48 bits virtual
  Порядок байт:        Little Endian
CPU(s):               3
  On-line CPU(s) list:  0-2
ID производителя:     AuthenticAMD
  Имя модели:           AMD Ryzen 5 1600 Six-Core Processor
  Семейство ЦПУ:        23
  Модель:               1
  Thread(s) per core:   1
  Ядер на сокет:        3
  Сокетов:              1
  Стейпинг:             1
  BogomIPS:             7186.50
```

Рис 24. Информация о процессоре

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

```
[Valerii-Sukhorukov@fedora ~]$ lspci
00:00.0 Host bridge: Intel Corporation 440FX - 82441FX PMC [Natoma] (rev 02)
00:01.0 ISA bridge: Intel Corporation 82371SB PIIX3 ISA [Natoma/Triton II]
00:01.1 IDE interface: Intel Corporation 82371AB/EB/MB PIIX4 IDE (rev 01)
00:02.0 VGA compatible controller: VMware SVGA II Adapter
00:03.0 Ethernet controller: Intel Corporation 82540EM Gigabit Ethernet Controller
00:04.0 System peripheral: InnoTek Systemberatung GmbH VirtualBox Guest Service
00:05.0 Multimedia audio controller: Intel Corporation 82801AA AC'97 Audio Controller
00:06.0 USB controller: Apple Inc. KeyLargo/Intrepid USB
00:07.0 Bridge: Intel Corporation 82371AB/EB/MB PIIX4 ACPI (rev 08)
00:0d.0 SATA controller: Intel Corporation 82801HM/HEM (ICH8M/ICH8M-E) SATA Controller
```

Рис 25. Информация о подключенных устройствах

4) Распределение аппаратных ресурсов

Файлы iports (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.

<i>Адреса портов ввода/вывода (Виртуальная машина Windows)</i>	
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

<i>Адреса портов ввода/вывода (Виртуальная машина Linux)</i>	
<i>Выделенные адреса</i>	<i>Устройство</i>
[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 (Intel 8281aa - ICH)
[D020-D027]	Эмулируемый сетевой гигабитный контроллер (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 - [3C0-3DF].

```

/proc/ioprots
0060-0060 : keyboard
0064-0064 : keyboard
0070-0071 : rtc_cmos
0070-0071 : rtc0
0080-008f : dma page reg
00a0-00a1 : pic2
00c0-00df : dma2
00f0-00ff : fpu
0170-0177 : 0000:00:01.1
0170-0177 : ata_piix
01f0-01f7 : 0000:00:01.1
01f0-01f7 : ata_piix
0376-0376 : 0000:00:01.1
0376-0376 : ata_piix
03c0-03df : vga+
03f6-03f6 : 0000:00:01.1

```

Рис 26. Фрагмент файла ioprots

```

/proc/iomem
00000000-00000fff : Reserved
00001000-0009fbff : System RAM
0009fc00-0009ffff : Reserved
000a0000-000bffff : PCI Bus 0000:00
000c0000-000c7fff : Video ROM
000e2000-000ef3ff : Adapter ROM
000f0000-000ffffff : Reserved
000f0000-000ffffff : System ROM
00100000-dfffffff : System RAM
9b000000-9c00247f : Kernel code
9c200000-9cce8fff : Kernel rodata
9ce00000-9d191c3f : Kernel data
9d9f5000-9dffffff : Kernel bss
dffff000-dfffffff : ACPI Tables
e0000000-fdffffff : PCI Bus 0000:00
e0000000-e1ffffff : 0000:00:02.0

```

Рис 28. Фрагмент файла iomem

```

/proc/interrupts
2048/2048
CPU0      CPU1      CPU2
0:         106          0          0 IO-APIC 2-edge timer
1:          0      2253          0 IO-APIC 1-edge i8042
8:          0          0          0 IO-APIC 8-edge rtc0
9:          0          0          0 IO-APIC 9-fastetoi acpi
12:         524          0          0 IO-APIC 12-edge i8042
14:          0          0          0 IO-APIC 14-edge ata_piix
15:          0          0          0 IO-APIC 15-edge ata_piix
18:          0      21030          0 IO-APIC 18-fastetoi vmwgfx
19:          0          0      8512 IO-APIC 19-fastetoi enp0s3
20:        24173          0          0 IO-APIC 20-fastetoi vboxguest
21:          0          0     25999 IO-APIC 21-fastetoi ahci[0000:00:0d.0], snd_intel18x0
22:          0          0      9138 IO-APIC 22-fastetoi ohci_hcd:usb1

```

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- F01FFFFFF]	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-F07FFFFFF]	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.

5. 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 одноклоночного или многоклоночного режима вывода могут использоваться опции `-l` и `-C`, соответственно.

```
[Valerii-Sukhorukov@fedora ~]$ ls -l
итого 0
drwxr-xr-x. 1 Valerii-Sukhorukov Valerii-Sukhorukov 0 июн  4 12:37 Видео
drwxr-xr-x. 1 Valerii-Sukhorukov Valerii-Sukhorukov 0 июн  4 12:37 Документы
drwxr-xr-x. 1 Valerii-Sukhorukov Valerii-Sukhorukov 0 июн  4 12:37 Загрузки
drwxr-xr-x. 1 Valerii-Sukhorukov Valerii-Sukhorukov 0 июн  4 12:37 Изображения
drwxr-xr-x. 1 Valerii-Sukhorukov Valerii-Sukhorukov 0 июн  4 12:37 Музыка
drwxr-xr-x. 1 Valerii-Sukhorukov Valerii-Sukhorukov 0 июн  4 12:37 Общедоступные
drwxr-xr-x. 1 Valerii-Sukhorukov Valerii-Sukhorukov 0 июн  4 12:37 'Рабочий стол'
drwxr-xr-x. 1 Valerii-Sukhorukov Valerii-Sukhorukov 0 июн  4 12:37 Шаблоны
```

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.

```
[Valerii-Sukhorukov@fedora ~]$ pwd
/home/Valerii-Sukhorukov
```

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 символической ссылки на него.

```
[Valerii-Sukhorukov@fedora ~]$ echo "Тестовый текст для файла source" > source
[Valerii-Sukhorukov@fedora ~]$ cat source
Тестовый текст для файла source
[Valerii-Sukhorukov@fedora ~]$ ln -s source symlink
[Valerii-Sukhorukov@fedora ~]$ cat symlink
Тестовый текст для файла source
[Valerii-Sukhorukov@fedora ~]$
```

Рис 31. Пример использования команд работы с файлами

6. Управление доступом к файловой системе.

1) 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.

2) 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.

```
[Valerii-Sukhorukov@fedora ~]$ sudo chown -v BAJIEPUU source.txt
изменён владелец 'source.txt' с Valerii-Sukhorukov на BAJIEPUU
```

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.

```
[Valerii-Sukhorukov@fedora ~]$ sudo chgrp -R -v BAJIEPUU Test_Folder
изменена группа 'Test_Folder' с Valerii-Sukhorukov на BAJIEPUU
```

Рис 33. Пример использования команды chgrp

3) 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-x), для остальных пользователей - 5(r-x).

```
[Valerii-Sukhorukov@fedora ~]$ sudo chmod -v 755 source.txt
[sudo] пароль для Valerii-Sukhorukov:
права доступа 'source.txt' изменены с 0664 (rw-rw-r--) на 0755 (rwxr-xr-x)
```

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`:

```
[Valerii-Sukhorukov@fedora ~]$ ls -l
итого 8
-rwxr-xr-x. 1 BAJIEPUU          Valerii-Sukhorukov 51 июн  7 21:35 source.txt
lrwxrwxrwx. 1 Valerii-Sukhorukov Valerii-Sukhorukov 10 июн  7 21:50 symlink -> source.txt
drwxr-xr-x. 1 Valerii-Sukhorukov BAJIEPUU           0 июн  7 21:39 Test_Folder
drwxr-xr-x. 1 Valerii-Sukhorukov Valerii-Sukhorukov  0 июн  4 12:37 Видео
drwxr-xr-x. 1 Valerii-Sukhorukov Valerii-Sukhorukov  0 июн  4 12:37 Документы
```

Рис 35. Просмотр свойств файлов и каталогов

Назначим всем пользователям полный доступ к `Test_Folder`, для файла `source.txt` установим SUID:

```
[Valerii-Sukhorukov@fedora ~]$ sudo chmod u+s source.txt
[Valerii-Sukhorukov@fedora ~]$ sudo chmod ugo+rwX Test_Folder
[Valerii-Sukhorukov@fedora ~]$ ls -l
итого 8
-rwsr-xr-x. 1 BAJIEPUU          Valerii-Sukhorukov 51 июн  7 21:35 source.txt
lrwxrwxrwx. 1 Valerii-Sukhorukov Valerii-Sukhorukov 10 июн  7 21:50 symlink -> source.txt
drwxrwxrwx. 1 Valerii-Sukhorukov BAJIEPUU           0 июн  7 21:39 Test_Folder
drwxr-xr-x. 1 Valerii-Sukhorukov Valerii-Sukhorukov  0 июн  4 12:37 Видео
```

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.

7. Управление заданиями

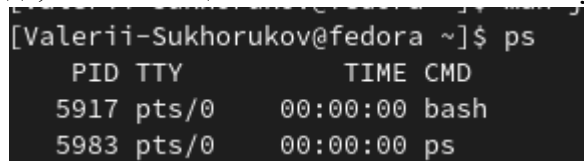
Всякая выполняющаяся в Linux программа называется процессом. Linux как многозадачная система характеризуется тем, что одновременно может выполняться множество процессов, принадлежащих одному или нескольким пользователям.

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:



PID	TTY	TIME	CMD
5917	pts/0	00:00:00	bash
5983	pts/0	00:00:00	ps

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.

```
[root@fedora ~]# sleep 100
```

Рис 37.

```
top - 22:15:21 up 3:24, 2 users, load average: 0,13, 0,18, 0,08
Tasks: 300 total, 1 running, 295 sleeping, 4 stopped, 0 zombie
%Cpu(s): 6,2 us, 0,9 sy, 0,0 ni, 92,4 id, 0,0 wa, 0,5 hi, 0,0 si, 0,0 st
MiB Mem : 3918,9 total, 728,1 free, 1485,3 used, 1705,6 buff/cache
MiB Swap: 3918,0 total, 3918,0 free, 0,0 used. 2179,5 avail Mem
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
1981	Valerii+	20	0	4586868	354412	131564	S	26,2	8,8	1:38.49	gnome-shell
5887	Valerii+	20	0	770460	57216	42648	S	1,7	1,4	0:07.06	gnome-terminal-
2487	Valerii+	20	0	366492	2780	2420	S	0,3	0,1	0:06.90	VBoxClient
6761	root	20	0	0	0	0	I	0,3	0,0	0:00.03	kworker/u6:4-events_unbound
6850	Valerii+	20	0	226424	4432	3532	R	0,3	0,1	0:00.25	top

Рис 38.

```
[Valerii-Sukhorukov@fedora ~]$ sudo kill 6761
[Valerii-Sukhorukov@fedora ~]$
```

Рис 39.

8. 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 символичные 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.