

# Учебный курс

## **Основы администрирования ОС Linux. Практикум L-adm-1**

Автор: Лесковец В.В.

г. Екатеринбург

2026

# Оглавление

Глава 1. Установка GNU/Linux.....	3
Глава 2. Начало работы в GNU/Linux.....	4
Глава 3. Основы командной строки.....	5
Глава 4. Использование графического интерфейса.....	9
Глава 5. Помощь и документация.....	10
Глава 6. Управление программным обеспечением.....	12
Глава 7. Управление пользователями.....	15
Глава 8. Логическая структура файловой системы. Работа с файлами и каталогами.....	19
Глава 9. Текстовые файлы и потоки.....	23
Глава 10. Текстовые редакторы.....	33
Глава 11. Работа с дисками и файловыми системами.....	34
Глава 12. Файловая система Linux.....	39
Глава 13. Процессы.....	46

# Глава 1. Установка GNU/Linux.

## 1.1. Установка Rocky Linux.

1. Создайте виртуальную машину со следующими характеристиками:
  1. CPU — 2 ядра;
  2. Память — 2Гб;
  3. Диск — 40Гб.
  4. Чипсет — UEFI.
2. Запустите виртуальную машину с установочного носителя.
3. Выберите русский язык для программы установки.
4. Выполните ручное разбиение диска со следующей схемой:
  1. Раздел ESP — 600Мб;
  2. /boot — 1Гб, ФС — ext4;
  3. swap — 2Гб;
  4. /home — 20Гб, ФС — xfs
  5. / - все что осталось, ФС — ext4.
5. Создайте пользователя sa, с паролем 12345678. Сделайте пользователя администратором. Полное имя Sys Admin.
6. Задайте пароль суперпользователю — 12345678.
7. Установите имя машины — rocky.
8. Отключить kdump и применение политик безопасности.
9. Установку производить с ближайшего зеркала.
10. Установить вариант: «Сервер с GUI».

## 1.2. Установка Debian.

1. Создайте виртуальную машину со следующими характеристиками:
  1. CPU — 2 ядра;
  2. Память — 2Гб;
  3. Диск — 40Гб.
  4. Чипсет — BIOS.
2. Запустите виртуальную машину с установочного носителя.
3. Выберите русский язык для программы установки.
4. Установите имя машины — debian.
5. Создайте пользователя sa, с паролем 12345678. Полное имя Sys Admin.
6. Задайте пароль суперпользователю — 12345678.
7. Выполните ручное разбиение диска со следующей схемой:
  1. /boot — 1Гб, ФС — ext4;
  2. swap — 2Гб;
  3. /home — 20Гб, ФС — btrfs
  4. / - все что осталось, ФС — ext4.
8. Установить дополнительное ПО: «Окружение рабочего стола Debian», «Стандартные системные утилиты».

## Глава 2. Начало работы в GNU/Linux.

### 2.1. Вход в систему

1. Получите у преподавателя данные для входа и войдите в сеанс.
2. Выйдите из сеанса.
3. Попробуйте войти в сеанс, используя неверное имя пользователя. Выдается ли при этом приглашение ввести пароль?

**Ответ: Да**

4. Попробуйте войти в сеанс, используя неверный пароль. Какое сообщение выдается при этом?

**Ответ: login incorrect**

5. Войдите в сеанс и выполните в нем команду `set -o ignoreeof`. Удастся ли после этого покинуть сеанс с помощью сочетания клавиш `C^D`.

**Ответ: Нет**

6. Попробуйте ввести команду `WHO` или `Who`. Что при этом происходит?

**Ответ: Выдается ошибка `command not found`**

7. Узнайте свои UID и GID.

**Ответ:**

```
$ id
```

8. К каким группам Вы принадлежите?

**Ответ:**

```
$ id
```

9. Какие идентификаторы имеет пользователь `root`?

**Ответ:**

```
$ id root
```

10. Как получить только UID пользователя с помощью команды `id`? Подсказка: используйте команду `id --help` для получения помощи по команде.

**Ответ:**

```
$ id -u
```

11. Используйте команду `who` с опцией `-H`. Удобнее ли она?

**Ответ: Да**

12. Определите время, прошедшее с момента загрузки Вашей системы.

**Ответ:**

```
$ uptime
```

## Глава 3. Основы командной строки.

### 3.1. Написание команд.

1. Определите имя исполняемого файла оболочки, запускаемой при входе в сеанс с вашим учетным именем.

**Ответ:**

**\$ echo \$SHELL**

2. Выполните команду `ls -l ~`, выводящую содержимое Вашего домашнего каталога в подробном формате.
3. Опишите структуру командной строки предыдущей команды: где в командной строке имя команды, опции и аргументы?

**Ответ: имя ls опция -l аргумент ~**

4. Посмотрите на приведенный выше пример использования команды `ls -d`. Проверьте, можно ли менять части командной строки местами.

**Ответ: нет**

5. Получите список системных команд, файлы которых находятся в каталоге `/bin`. Есть ли среди них знакомые Вам команды?

**Ответ:**

**\$ ls /bin**

6. Обычно помимо встроенной команды `pwd` имеется ее системный двойник в каталоге `/bin`. Попробуйте вызвать встроенную и системную команды `pwd` с опцией `--help`. Есть ли разница в работе встроенной и системной команд?

**Ответ: встроенная — pwd и системная — /bin/pwd**

7. Напечатайте в командной строке `sl -F /etc /tmp /opt`. Используя команды, приведенные в таблице в данной главе исправьте команду на `ls`. Выполните команду.

**Ответ:**

**\$ Ctrl-P Ctrl-A Alt-D ls**

8. С помощью механизма замены строк замените опции команды на `-ld` и выполните команду.

**Ответ:**

**\$ ^-F^-ld**

9. Введите эту же команду так, чтобы имя команды, ее опции и все аргументы занимали отдельную строку. То есть каждый элемент командной строки должен быть введен в отдельной строке.

**Ответ:**

**\$ ls \**

**> -ld \**

```
> /etc \  
> /tmp \  
> /opt
```

10. Получите, используя переменные окружения, имена текущего каталога и домашнего каталога.

**Ответ:**

```
$ echo $HOME
```

```
$ echo $PWD
```

11. Создайте новую переменную NEWVAR со значением 1982 и проверьте, доступна ли она в порожденной оболочке.

**Ответ: нет**

```
$ NEWVAR=1982
```

```
$ echo $NEWVAR
```

```
1982
```

```
$ bash
```

```
$ echo $NEWVAR
```

**Ничего не выводится**

```
$ exit
```

12. Экспортируйте переменную NEWVAR и снова проверьте, доступна ли она в порожденной оболочке.

**Ответ: да**

```
$ export NEWVAR
```

```
$ bash
```

```
$ echo $NEWVAR
```

```
1982
```

```
$ exit
```

13. Получите списки переменных оболочки и переменных окружения с их значениями. Какой список содержит больше переменных?

**Ответ: список переменных оболочки - set и переменных окружения — env. список переменных оболочки больше, т. к. все переменные окружения также и переменные оболочки.**

### 3.2. История команд.

1. Выполните последнюю введенную команду заново.

**Ответ:**

```
$ !!
```

2. Пользуясь клавишами управления курсором, найдите в файле истории команду echo и выполните ее.

**Ответ: используется клавиша «стрелка вверх».**

3. Вызовите последнюю введенную команду echo по первым двум буквам ее имени.

**Ответ:**

**\$ !ec**

4. Вызовите команду, содержащую подстроку ho.

**Ответ: !?ho**

5. Выведите в текстовый редактор последнюю исполненную команду.

**Ответ:**

**\$ fc echo**

6. Выведите полный список команд в файле истории.

**Ответ:**

**\$ history**

7. Введите команду ls -ld /, а затем получите список всех возможных продолжений командной строки. Подставьте все возможные продолжения в командную строку и выполните команду.

**Ответ:**

**\$ ls -ld /Alt-? Alt-\***

8. Получите значение переменной окружения HISTFILESIZE, пользуясь механизмом дополнения имен с автоматическим определением контекста дополнения.

**Ответ: используйте клавишу «tab» после ввода echo \$HI**

9. Получите список всех возможных подстановок имен переменных окружения, начинающихся с символа H.

**Ответ:**

**\$ echo \$H Alt-\$ Alt-\$**

10. Введите команду ping -c3 и первую букву имени хоста localhost. Получите список всех возможных продолжений имен хостов. Выберите из них localhost и выполните команду.

**Ответ:**

**\$ ping -c3 locCtrl-x @ Alt-@**

### 3.3. Псевдонимы, шаблоны и командная подстановка.

1. Выполните последнюю введенную команду заново.

**Ответ:**

**\$ !!**

2. Пользуясь клавишами управления курсором, найдите в файле истории команду echo и выполните ее.

**Ответ: используется клавиша «стрелка вверх».**

3. Вызовите последнюю введенную команду echo по первым двум буквам ее имени.

**Ответ:**

**\$ !ec**

4. Вызовите команду, содержащую подстроку ho.

**Ответ: !?ho**

5. Выведите в текстовый редактор последнюю исполненную команду.

**Ответ:**

**\$ fc echo**

6. Выведите полный список команд в файле истории.

**Ответ:**

**\$ history**

7. Введите команду ls -ld /, а затем получите список всех возможных продолжений командной строки. Подставьте все возможные продолжения в командную строку и выполните команду.

**Ответ:**

**\$ ls -ld /Alt-? Alt-\***

8. Получите значение переменной окружения HISTFILESIZE, пользуясь механизмом дополнения имен с автоматическим определением контекста дополнения.

**Ответ: используйте клавишу «tab» после ввода echo \$HI**

9. Получите список всех возможных подстановок имен переменных окружения, начинающихся с символа H.

**Ответ:**

**\$ echo \$H Alt-\$ Alt-\$**

10. Введите команду ping -c3 и первую букву имени хоста localhost. Получите список всех возможных продолжений имен хостов. Выберите из них localhost и выполните команду.

**Ответ:**

**\$ ping -c3 locCtrl-x @ Alt-@**

# Глава 4. Использование графического интерфейса.

## 4.1 Установка графического интерфейса

1. Установите tasksel.

**Ответ:**

```
$ sudo apt tasksel
```

2. Установите окружение рабочего стола Cinnamon.

**Ответ: выбор опции с помощью пробела, и подтверждение Enter**

3. Перейдите в систему, выбрав новое окружение.
4. Вернитесь обратно в окружение рабочего стола GNOME.

## Глава 5. Помощь и документация.

### 5.1. Работа со справкой

1. Проверьте, работает ли опция `--help` с командой `id`.

**Ответ: работает**

2. Попробуйте вместо `--help` использовать `-h` с командой `id`.

**Ответ: выдает ошибку**

3. Подсчитайте количество строк в файле `/etc/hosts` с помощью команды `wc`, изучите базовую подсказку по этой команде и используйте нужную опцию для подсчета строк.

**Ответ:**

```
$ wc --help
```

```
...
```

```
-l, --lines print the newline counts
```

```
...
```

```
$ wc -l /etc/hosts
```

4. Является ли команда `cd` встроенной командой оболочки?

**Ответ:**

```
$ help cd.
```

**cd не может быть не встроенной.**

5. Получите помощь по команде `alias`.

**Ответ:**

```
$ help alias
```

6. Для чего нужен ключ `-P` команды `man`?

**Ответ:**

```
$ man --help
```

**Если выдается сообщение команда не найдена, то установите пакет `man-db`: `sudo yum install man-db`**

7. Выведите все имеющиеся данные о системе `man`.

**Ответ:**

```
$ man man
```

8. Получите все страницы, называющиеся `groff`.

**Ответ:**

```
$ man -f groff
```

9. Найдите все файлы страниц `man`, касающиеся любых объектов, называющихся `exit`.

**Ответ:**

**\$ man -k exit**

10. Вы собираетесь опубликовать игру для X Window, называющуюся xzombie. Как следует назвать файл страницы man по этой игре?

**Ответ: xzombie.6**

11. В каком каталоге следует установить эту страницу, если она используется на данной системе локально и не связана с системой установки пакетов?

**Ответ: /usr/local/share/man/man6**

12. Вы собираетесь опубликовать обсуждаемую игру в виде пакета, устанавливаемого с помощью системы установки программного обеспечения RPM (Red Hat Package Manager). В какой каталог в таком случае следует установить страницу помощи?

**Ответ: /usr/share/man/man6**

13. Получите помощь по команде who, пользуясь системой info.

**Ответ:**

**\$ info who**

14. Найдите раздел документации info, относящийся к теме переходов по гипертекстовым ссылкам, перейдите в родительский узел.

**Ответ:**

**\$ info info**

**Команда u**

15. Перейдите в начало страницы, найдите строку scroll на этой странице.

**Ответ: /scroll / - команда поиска**

16. Выйдите из info.

**Ответ: q**

17. Проверьте, какая документация для оболочки Bash имеется в вашей системе.

**Ответ:**

**\$ ls /usr/share/doc/bash\***

18. Имеется ли дополнительная информация о системе man?

**Ответ:**

**\$ ls -d /usr/share/doc/man\***

19. В вашей системе может быть установлена документация HOWTO. Проверьте ее наличие в каталоге с /usr/share/doc.

**Ответ:**

**\$ ls -d /usr/share/doc/[hH]\***

# Глава 6. Управление программным обеспечением.

## 6.1. Управление пакетами rpm.

1. Получите список установленных пакетов.

**Ответ:**

```
# rpm -qa
```

2. Как в RPM увеличить информативность получаемых данных, например, при проверке целостности установленных пакетов.

**Ответ: опция -v - выводить более детальные сообщения**

3. Определите, каким образом можно получить список всех возможных ключей запроса к базе данных RPM?

**Ответ:**

```
$ rpm --help
```

```
$ man rpm
```

```
rpm --querytags
```

4. Выведите информацию о полном имени, группе и размере пакета bash.

**Ответ:**

```
$ rpm -qi bash
```

5. Получите список включенных и доступных репозиториев.

**Ответ:**

```
# dnf repolist
```

6. Установите репозиторий epel. (Для этого имеется стандартный пакет).

**Ответ:**

```
# dnf install epel-release
```

7. 8. Установите пакеты bash-completion и vim-enhanced.

**Ответ:**

```
# dnf install bash-completion vim-enhanced
```

8. Найдите и установите пакет, который предоставляет команду ifconfig.

**Ответ:**

```
# dnf provides ifconfig
```

9. Обновите ОС.

**Ответ:**

```
# dnf upgrade
```

10. Получите список доступных групп пакетов.

**Ответ:**

**# dnf group list**

11. Установите группу пакетов для разработчика (Development Tools)

**Ответ:**

**# dnf group install "Development Tools"**

12. Узнайте список пакетов входящих в группу «Графические средства администрирования».

**Ответ:**

**# dnf group info "Графические средства администрирования"**

## 6.2. Пакеты Debian.

1. Получите список установленных пакетов.

**Ответ:**

**# dpkg -l**

2. Определите из какого пакета установлен файл /usr/bin/passwd.

**Ответ:**

**# dpkg -S /usr/bin/passwd**

3. Установите пакет apt-file и с помощью команды apt-file узнайте какой пакет содержит команду ifconfig. Установите этот пакет.

**Ответ:**

**# apt install apt-file**

**# apt-file update**

**# apt-file search ifconfig**

4. Отключите репозитории deb-src.

**Ответ:**

**# apt edit-sources**

**Закомментируйте строку с deb-src добавив в начало строки #**

5. Добавьте в репозиториях компоненты contrib и non-free.

**Ответ:**

**# apt edit-sources**

**Допишите contrib и non-free в конец строк**

6. Обновите локальный кеш репозиториев.

**Ответ:**

**# apt update**

7. Обновите пакеты, если имеются обновления.

**Ответ:**

**# apt upgrade**

8. Установите с помощью taskset задачу «SSH-сервер».

**Ответ:**

**# taskset install «SSH-сервер»**

## Глава 7. Управление пользователями.

### 7.1. Управление пользователями.

1. Просмотрите файл `/etc/shadow` (с правами `root`). У всех ли пользователей содержимое второго поля выглядит приблизительно одинаково или же у некоторых пользователей имеются существенные отличия от записей для других пользователей? Подсказка: запустить оболочку с правами `root` можно при помощи команды `sudo -i`

**Ответ:**

```
$sudo -i
```

```
#cat /etc/shadow
```

Есть отличия для пользователей, у которых нет пароля, во втором поле установлено `*` или `!!`. У пользователей с паролями там хэш пароля.

2. Какие символы могут содержаться зашифрованной строке пароля во втором поле файла `/etc/shadow` ?

**Ответ: Цифры, спецсимволы, буквы в обоих регистрах**

3. Зарегистрируйте пользователя `test1`, для которого запрещен вход в сеанс, имеющего домашний каталог `/var/spool/mail`, и являющегося членом групп `users` и `mail`. Пользователь должен иметь `UID=1100`.

**Ответ:**

```
# useradd -s /sbin/nologin -d /var/spool/mail -G users,mail -u 1100 test1
```

4. Создайте учетную запись для пользователя `test2` с настройками по умолчанию, но без создания приватной группы. Проверьте, создан ли домашний каталог пользователя, наполнен ли он файлами, и кому он принадлежит.

**Ответ:**

```
# useradd -n test2
```

```
# ls -l /home
```

```
# ls /home -a /test2
```

5. Измените имя пользователя `test2` на `test3`.

**Ответ:**

```
# usermod -l test3 test2
```

6. Удалите пользователя `test3` вместе с его домашним каталогом.

**Ответ:**

```
# userdel -r test3
```

7. Помимо файла `/etc/default/useradd` имеется еще один конфигурационный файл, влияющий на поведение команды `useradd` (см. `man useradd`). Найдите его и изучите его содержание. Какая настройка позволяет изменять минимальный `UID` для новых пользователей?

**Ответ:** Файл `/etc/login.defs` и параметр в нем `UID_MIN`

```
# grep ^UID_MIN /etc/login.defs
```

```
UID_MIN          1000
```

8. Зарегистрируйте пользователя `test4` с настройками по умолчанию и установите для него пароль. Изучите содержимое соответствующей пользователю записи в файле `/etc/shadow`.

**Ответ:**

```
# useradd test4
```

```
# passwd test4
```

```
# cat /etc/shadow
```

9. Установите дату устаревания пароля (`Password expires`) для пользователя `test4` на 31 декабря текущего года. Проверьте, что изменилось в `/etc/shadow`.

**Ответ:**

```
# chage -E 2026-12-31 test4
```

10. Удалите пароль пользователя и проверьте изменения в `/etc/shadow`.

**Ответ:**

```
# passwd -d test4
```

```
# cat /etc/shadow
```

11. Заблокируйте учетную запись `test4`.

**Ответ:**

```
# usermod -e 1 test4
```

**Команды `passwd -l` или `usermod -l` не всегда могут подойти, так как блокируют только вход в систему с аутентификацией по паролю. Но если останутся другие способы входа в систему (например при помощи ключей `ssh`), то вход в учетную запись будет разрешён при использовании этих методов. Команда `usermod -e 1` - делает устаревшей саму учетную запись, поэтому другие методы входа в систему тоже не будут работать.**

12. Создайте группу пользователей `xusers` с GID 1010.

**Ответ:**

```
# groupadd -g 1010 xusers
```

13. Зарегистрируйте себя (пользуясь правами `root`) в качестве участника группы `xusers`.

**Ответ:**

```
# gpasswd -a sa xusers
```

14. Попробуйте от имени своей учетной записи добавьте пользователя `test4` в группу `xusers`.

**Ответ:**

**`$ gpasswd -a test4 xusers` Команда проваливается. Не смотря на то что ваша учетная запись входит в данную группу - это ещё не дает вам права добавлять в неё других**

пользователей. Добавлять пользователя в группу могут только root, администраторы с использованием sudo или администраторы самой группы.

15. Найдите утилиту, позволяющую изменять имена и GID групп. Измените имя группы на yusers.

**Ответ:**

**# groupmod -n yusers xusers**

## 7.2. Управление средой пользователя.

1. Измените значение umask на 027 для всех пользователей системы, оболочкой по умолчанию для которых является Bash.

**Ответ:**

**# vi /etc/profile**

2. Установите в собственном профиле оболочки псевдоним l для команды ls -l.

**Ответ:**

**#vi .bash\_profile**

**alias l='ls -l'**

3. Каким образом сделать так, чтобы этот же псевдоним устанавливался и для всех вновь регистрируемых пользователей в системе, для которых оболочкой по умолчанию будет Bash?

**Ответ: Добавить alias l='ls -l' в /etc/profile**

4. В каком файле удобнее всего добавить к переменной окружения PATH путь к каталогу bin, находящемуся в домашнем каталоге обычного пользователя? Проверьте может такая настройка уже есть.

**Ответ:**

**# cat ~/.bash\_profile там есть строчка PATH=\$PATH:\$HOME/bin**

5. Измените вид вторичного приглашения командной строки на -> так, чтобы при каждом входе в сеанс Bash значение этой переменной устанавливалось автоматически.

**Ответ:**

**#vim ~/.bashrc**

**PS2="->"**

## 7.3. Мониторинг пользовательской активности.

1. Определите, когда была последний раз загружена система.

**Ответ:**

**# last reboot**

2. С помощью опции -a команды who получите подробную информацию о пользователях и статусе системы.

**Ответ:**

**# who -aq**

3. С помощью команды `who` получите список пользователей, входивших в сеанс ранее.

**# who**

4. Сравните предыдущий список со списком, выводимым командой `last`.

**Ответ:**

**# last**

5. Получите отчет по входам в сеанс суперпользователя с помощью `lastlog`.

**Ответ:**

**# lastlog | grep root**

6. Кто входил в сеанс за последние пять дней?

**Ответ:**

**# lastlog -t 5**

7. Откройте еще один сеанс и с помощью `loginctl` получите описание и состояние сеанса. Завершите этот сеанс.

**Ответ:**

**# loginctl show-session 3**

**# loginctl terminate-session 3**

# Глава 8. Логическая структура файловой системы. Работа с файлами и каталогами.

## 8.1. Структура файловой системы

1. Имя домашнего каталога ~. Выведите его содержимое.

**Ответ:**

```
$ ls ~
```

2. Выведите содержимое домашнего каталога, включая скрытые файлы.

**Ответ:**

```
$ ls -a ~
```

3. Можно ли использовать с командой ls опцию --all?

**Ответ:**

```
$ ls -all да
```

4. Выведите содержимое домашнего каталога в обратном (реверсивном) формате.

**Ответ:**

```
$ ls -r
```

5. Узнайте какая опция команды ls позволяет вывести файлы в порядке, отсортированном по времени изменения этих файлов.

**Ответ:**

```
$ ls -t
```

6. Можно ли получить список файлов, отсортированный по их размеру?

**Ответ:**

```
$ ls -S или ls -Sr
```

7. Получите список не сортированный файлов в каталоге /tmp.

**Ответ:**

```
$ ls -U /tmp
```

8. Перейдите в каталог /tmp.

**Ответ:**

```
$ cd /tmp
```

9. С помощью одной команды перейдите в подкаталог local/bin каталога /usr.

**Ответ:**

```
$ cd /usr/local/bin
```

10. Перейдите в домашний каталог, проверьте, какой каталог является текущим.

**Ответ:**

```
$ cd; pwd
```

11. С помощью одной команды создайте цепочку каталогов dir1/dir2/dir3/dir4 в домашнем каталоге. Проверьте создались ли они.

**Ответ:**

```
$ mkdir -p dir1/dir2/dir3/dir4; ls -R dir1
```

12. Создайте в каталоге dir1/dir2 файл Bubuka.

**Ответ:**

```
$ > dir1/dir2/Bubuka
```

**или**

```
$ touch dir1/dir2/Bubuka
```

13. Попробуйте удалить цепочку dir1/dir2/dir3/dir4 . Проверьте, какие каталоги удалены.

**Ответ:**

```
$ rmdir -p dir1/dir2/dir3/dir4
```

**проверка:**

```
$ ls -R
```

14. Удалите каталог dir1 со всем содержимым.

**Ответ:**

```
$ rm -rf dir1
```

## 8.2. Управление файлами

1. Создайте каталог Toppler, содержащий два файла high11 и low11. Скопируйте оба эти файла в каталог /tmp.

**Ответ:**

```
$ mkdir Toppler
```

```
$ cd Toppler
```

```
$ touch high11 low11
```

```
$ cp high11 low11 /tmp
```

2. Переместите эти два файла из каталога /tmp в домашний каталог, используя символы подстановки.

**Ответ:**

```
$ mv /tmp/*11 ~
```

3. Рекурсивно скопируйте каталог Toppler в /tmp.

**Ответ:**

```
$ cp -r Toppler /tmp
```

4. Переименуйте в /tmp этот каталог в Roller.

**Ответ:**

```
$ mv /tmp/Toppler /tmp/Roller
```

5. Переместите полученный каталог со всем содержимым в домашний каталог так, чтобы на экране отображалась подробная информация об этом процессе.

**Ответ:**

```
$ mv -v /tmp/Roller ~
```

6. Найдите в домашнем каталоге все пустые файлы.

**Ответ:**

```
$ find ~ -empty
```

7. Найдите в домашнем каталоге все файлы, больше 1Кб, но меньше 10Кб.

**Ответ:**

```
$ find ~ -size +1k -size -10k
```

8. В каталоге /dev проверьте нет ли файлов, принадлежащих вам.

**Ответ:**

```
$ find /dev -user $(whoami)
```

**Обратите внимание на командную подстановку. Такая команда будет универсальной при любом имени пользователя.**

9. Произведите быстрый поиск всех файлов, содержащих в имени строку user.

**Ответ:**

```
$ locate user
```

**Возможно для выполнения задания будет необходимо установить пакет mlocate (sudo yum install mlocate) и создать индексную базу данных с именами файлов.**

10. Создайте новый файл filemy. Попытайтесь найти его с помощью locate.

**Ответ:**

```
$ > filemy
```

```
$ locate filemy
```

11. Войдите в сеанс как суперпользователь, проиндексируйте базу данных поиска и вновь повторите поиск от имени простого пользователя.

**Ответ:**

```
$ su -
```

```
# updatedb
```

**или**

```
$ sudo updatedb
```

12. Определите типы файлов текущего каталога, команды /bin/bash и файла ~/.bashrc

**Ответ:**

```
$ file /bin/bash
```

```
$ file ~/.bashrc
```

13. Узнайте где находится файл с магическими числами.

**Ответ:**

**man file**

**см. секцию FILES (/usr/share/misc/magic)**

14. Просмотрите его структуру. Как команда file определяет тип файла?

**Ответ:**

**\$ less /usr/share/misc/magic**

**В этом файле содержится таблица где в первой колонке описывается где искать, во второй какого типа данные, в третьей что должно быть в этом месте и в четвертой что это такое.**

15. Можно ли создать собственный тип файла?

**Ответ: да**

## Глава 9. Текстовые файлы и потоки.

### 9.1. Перенаправление и вывод текстового потока

1. Выполните команду поиска всех обычных файлов в каталоге /usr так, чтобы найденные имена файлов были записаны в файл SysComm в домашнем каталоге, а поток ошибок был записан в нуль-устройство /dev/null.

**Ответ:**

```
$ find /usr -type f > SysComm 2> /dev/null
```

2. Выполните ту же команду, но так, чтобы потоки вывода и ошибок были записаны в файл SysComm.

**Ответ:**

```
$ find /usr -type f >& SysComm
```

3. Допишите в конец файла SysComm информацию о текущей дате и времени, выводимую командой date.

**Ответ:**

```
$ date >> SysComm
```

4. Выведите текущую дату в формате: YYYY-MM-DD hh:mm:ss

**Ответ:**

```
$ date +"%Y-%m-%d %T"
```

5. Установите опцию noclobber и сотрите содержимое файла ps.txt с помощью перенаправления вывода. Снимите опцию noclobber .

**Ответ:**

```
$ set -o noclobber
```

```
$ >| ps.txt
```

```
$ set +o noclobber
```

6. Проведите поиск файлов символьных ссылок в каталоге /usr/share/doc так, чтобы их список был выведен в отсортированном виде с помощью фильтра sort.

**Ответ:**

```
$ find /usr/share/doc -type l | sort
```

7. Выведите сортированный список пользователей, вошедших в сеанс, в системе в файл seans.txt и на экран одновременно.

**Ответ:**

```
$ who | sort | tee seans.txt
```

8. Опция -n команды echo подавляет добавление перевода строки. С помощью man определите, как это можно сделать иначе.

**Ответ: из man echo: \c produce no further output**

```
$ echo -e abc\\c
```

## 9.2. Вывод содержимого файла

1. Команда `man` обладает опцией для явного указания пэйджера, в котором будет отображаться найденная страница помощи. Попробуйте ее действие.

**Ответ:**

```
$ man man
```

```
$ man -P more man
```

2. Команда `less` позволяет открыть несколько файлов, указав их в качестве аргументов.

**Ответ:**

```
$ less f1 f2
```

**переключиться на следующий файл командой :n**

**Это можно узнать из встроенной справки (команда h).**

3. Команда `cat` обладает опцией, устанавливающей нумерацию выводимых строк. Определите, какая она.

**Ответ:**

```
$ cat -n
```

4. Найдите опцию `cat`, позволяющую этой команде удалять последовательно повторяющиеся переводы строки, оставляя лишь один перевод строки.

**Ответ:**

```
$ cat -s
```

5. Выведите содержимое файла `/etc/passwd` в обратном порядке следования строк.

**Ответ:**

```
$ tac /etc/passwd
```

6. Для чего предназначена опция `-b` команды `tac`?

**Ответ: присоединяет разделитель к началу строки**

7. \*Получите имена трех наиболее объемных файлов в каталоге `/usr/bin`.

**Ответ:**

```
$ ls -Ss /usr/bin
```

```
total 47940
```

```
5232 ld.gold
```

```
3104 dwp
```

```
2284 vim
```

```
1524 systemd-analyze
```

```
984 ld.bfd
```

```
944 bash
```

```
...
```

опция `-S` сортирует вывод по размеру. Опция `-s` показывает размер файлов. Общая сводка по размерам всех файлов нам не нужна, поэтому нужно выводить только со 2 строки (команда `tail -n +2`) а, затем, получить только 3 первых строки:

```
$ ls -Ss /usr/bin | tail -n +2 | head -3
```

или другой вариант

```
$ ls -Ss /usr/bin | head -4 | tail -3
```

### 9.3.Регулярные выражения

1. Составьте регулярное выражение для пустой строки.

**Ответ:** `^$`

2. Найдите опцию, позволяющую `grep` выводить имена файлов перед найденными строками.

**Ответ:**

```
$ grep --help | grep filename
```

**Опция `-H, --with-filename` print the file name for each match**

3. \*С помощью `grep` получите список всех пользователей системы, с `10<GID<100`.

**Ответ:** В регулярных выражениях нельзя сравнивать числа как числа. Для регулярных выражений числа это тоже строки. Кроме того в данном примере нужно правильно определить позицию где находится GID в файле `/etc/passwd`.

```
$ grep '^.*:.*:[0-9]*:[1-9][0-9]:' /etc/passwd
```

4. \*Отфильтруйте все сообщения, находящиеся в файле `/var/log/messages` так, чтобы были выведены сообщения о событиях, происшедших вчера в диапазоне с 8.30 утра до 12.30.

**Ответ:**

Перед выполнением задания следует ознакомиться со структурой записей в журнале:

```
$ sudo tail -1 /var/log/messages
```

```
Jan 14 07:59:25 lin00 systemd: Started Session c1 of user user.
```

Обратите внимание как задается время и на какой оно позиции находится.

Здесь так же как и с числами: в регулярных выражениях нет возможности анализировать время напрямую. Чтобы решить эту задачу, надо для начала определить интервалы времени, которые нас интересуют:

```
8:30-8:59
```

```
9:00-9:59
```

```
10:00-10:59
```

```
11:00-11:59
```

```
12:00-12:29
```

Т.о. у нас есть 2 интервала по полчаса и 3 по часу. Интервалы 10 и 11 часов мы можем определить как одно рег. выражение:

```
1[01]:[0-5][0-9]
```

Теперь этот интервал в 2 часа можем объединить с интервалом 9 часов:

```
(09|1[01]):[0-5][0-9]
```

Остались 2 интервала:

```
08:[3-5][0-9] и 12:[0-2][0-9]
```

Теперь все это объединяем в общее регулярное выражение:

```
(08:[3-5][0-9]|(09|1[01]):[0-5][0-9]|12:[0-2][0-9]):
```

Теперь нужно получить вчерашнюю дату

```
$ LANG=C date -d '1 day ago'
```

Мы получим что-то подобное: Wed Jan 13 08:09:06 UTC 2021

LANG=C нужно, чтобы получить нужный язык. Из всего вывода нам нужны только месяц и день. см. date --help

```
$ LANG=C date -d '1 day ago' +%b %_d'
```

```
Jan 13
```

Наконец команда будет следующая:

```
$ sudo egrep "^$(LANG=C date -d '1 day ago' +%b %_d') (08:[3-5][0-9]|(09|1[01]):[0-5][0-9]|12:[0-2][0-9]):" /var/log/messages
```

5. Найдите в каталоге /etc/rc.d и его подкаталогах все файлы, содержащие слово fsck . В выводе должны присутствовать имена найденных файлов.

Ответ:

```
$ fgrep -r fsck /etc/rc.d/
```

6. \*С помощью команд ps, grep, sort, uniq подсчитайте сколько процессов в настоящий момент связано с каждым виртуальным терминалом (tty) в вашей системе.

Ответ:

```
$ ps -eo tty | fgrep -v '?' | fgrep -v TT | sort | uniq -c
```

7. \*Получите список всех учетных записей пользователей вашей системы, у которых имя домашнего каталога (последнего каталога в пути) совпадает с именем пользователя.

Ответ: Посмотрим на структуру текста в passwd

```
$ getent passwd user
```

```
user:x:1000:1000:~/home/user:/bin/bash
```

Нам нужно искать совпадения в первом и шестом полях, поэтому:

```
$ getent passwd | egrep '(.+):.*:[0-9]+:[0-9]+:.*:/\1:'
```

или

```
$ getent passwd | grep '\(..*\):.*:[0-9][0-9]*:[0-9][0-9]*:.*:/\1:'
```

8. Проверьте, имеются ли в файле `/var/log/messages` (файл системного журнала) сообщения, у которых во времени совпадают часы и минуты (например, `09:09`).

**Ответ:**

```
sudo egrep '^... .. ([0-9][0-9]):\1:' /var/log/messages
```

#### 9.4. Поточковые редакторы

1. Получите столбец из имен пользователей, находящийся в данный момент в сеансе.

**Ответ:**

```
$ who | cut -f1 -d" "
```

2. Получите столбец, составленный из символов строк – имен файлов в текущем каталоге таким образом, чтобы были выведены символы с пятого до последнего.

**Ответ:**

```
$ ls | cut -c5-
```

#### 9.5. Работа со строками

1. Введите команду, сравнивающую содержимое вашего домашнего каталога с домашним каталогом суперпользователя. (Предварительно зайдите за учётную запись `root`'а с помощью команды `sudo -i` и введите пароль от своей учётной записи администратора.)

**Ответ:**

```
# diff ~root ~user
```

2. Определите, с помощью какой опции `diff` можно рекурсивно сравнить эти каталоги.

**Ответ:**

```
# diff -r ~root ~user
```

3. Поочередно запишите результат выполнения команды `ps` в файлы `ps1.txt` и `ps2.txt`. Получите файл `patch.txt` с отличиями файлов `ps1.txt` и `ps2.txt` в контекстном формате. Удалите файл `ps2.txt` и восстановите его содержимое с помощью файлов `ps1.txt` и `patch.txt`.

**Ответ:**

```
$ ps > ps1.txt
```

```
$ ps > ps2.txt
```

```
$ diff -c ps1.txt ps2.txt > patch.txt
```

```
$ rm ps2.txt
```

```
$ patch -o ps2.txt ps1.txt patch.txt
```

4. Получите список групп пользователей в системе, отсортированный по GID в обратном числовом порядке.

**Ответ:**

```
$ cat /etc/passwd | sort -t: -nk3 -r
```

5. С помощью утилит find, head и sort получите список из десяти файлов в домашнем каталоге, занимающих наибольшее дисковое пространство.

**Ответ:**

```
$ find ~ -type f -exec ls -sh {} \; | sort -hr | head
```

6. \*Выведите список тех файлов текущего каталога, для которых можно найти в этом каталоге файлы, первые три символа в именах которых совпадают не менее, чем у двух файлов.

**Решение:**

**Для начала получим список из первых 3 букв имен файлов:**

```
$ ls | cut -c1-3
```

**Wel**

**Wel**

**bam**

**bam**

**div**

**div**

**emp**

**emp**

**fir**

**msd**

**...**

**Из этого списка нужны только повторяющиеся строки. Дополнительно выведем количество повторений и отсортируем по количеству повторений:**

```
$ ls | cut -c1-3 | sort | uniq -dc | sort -nr
```

**6 tex**

**5 tab**

**2 sam**

**2 sal**

**2 emp**

**2 div**

**2 bam**

**2 Wel**

**Мы увидели что повторения есть, но пока не видим имен файлов.**

**Чтобы получить имена мы сконструируем команды для получения этих имен:**

```
$ ls | cut -c1-3 | sort | uniq -dc | sort -nr | awk '{print "echo "$0"*"}'
```

```
echo 6 tex*
```

```
echo 5 tab*
echo 2 sam*
echo 2 sal*
echo 2 emp*
echo 2 div*
echo 2 bam*
echo 2 Wel*
```

Теперь остается только передать команды на выполнение оболочке. Итоговый ответ:

```
$ ls | cut -c1-3 | sort | uniq -dc | sort -nr | awk '{print "echo "$0"*"}' | bash
6 text.txt text1 text1.txt text2 text_h.txt text_t.txt
5 tab_class.txt tab_dept tab_dept.dat tab_dept.txt tab_staff.txt
2 sample sample.txt
2 salary.txt salary_2.txt
2 empty1 empty2
2 div.txt div_4.txt
2 bamboo bamboo.txt
2 Wells Wells.txt
```

Важное замечание. Иногда необходимо «налету» сконструировать скрипт и тут же его выполнить.

Другой ответ с использованием цикла. Циклы часто бывают удобнее, но не в этом случае:

```
$( { for f in $(ls | cut -c1-3 | sort | uniq -d); do echo -n "$(ls ${f}* | wc -l)" "; echo ${f}*; done } | sort -nr
6 text.txt text1 text1.txt text2 text_h.txt text_t.txt
5 tab_class.txt tab_dept tab_dept.dat tab_dept.txt tab_staff.txt
2 sample sample.txt
2 salary.txt salary_2.txt
2 empty1 empty2
2 div.txt div_4.txt
2 bamboo bamboo.txt
2 Wells Wells.txt
```

7. \*Имеется два каталога без подкаталогов. Как с помощью join получить список файлов, которые имеются в обоих каталогах?

Ответ:

```
# ls ~root | sort > /tmp/root.ls
```

```
# ls ~admin | sort > /tmp/admin.ls
```

```
# join /tmp/root.ls /tmp/admin.ls
```

8. \*Как сделать то же самое для файлов, которые имеются только в одном из каталогов и отсутствуют в другом?

**Ответ:**

```
# join -v1 /tmp/root.ls /tmp/admin.ls
```

```
# join -v2 /tmp/root.ls /tmp/admin.ls
```

9. \*Как получить список файлов кроме тех, которые имеются в обоих каталогах?

**Ответ:**

```
# join -v1 v2 /tmp/root.ls /tmp/admin.ls
```

## 9.6. Продвинутая работа со строками

1. Определите, используя файл `/etc/passwd`, содержащий данные об учетных записях пользователей, сколько пользователей зарегистрировано в системе.

**Ответ:**

```
$ cat /etc/passwd | wc -l
```

2. Определите для скольких пользователей оболочкой по умолчанию является `bash`.

**Ответ:**

```
$ awk -F':' '$7~/bash$/' /etc/passwd | wc -l
```

3. Сколько имеется пользователей, UID которых больше 100?

**Ответ:**

```
$ awk -F':' '$3>100' /etc/passwd | wc -l
```

4. Выведите пронумерованный список файлов в текущем каталоге.

**Ответ:**

```
$ ls | nl
```

5. Сделайте то же, но с шагом нумерации равным двум.

**Ответ:**

```
$ ls | nl -i 2
```

6. Выведите последние три строки файла `/etc/passwd`, заменив разделители – двоеточия на разделители – вертикальные черты.

**Ответ:**

```
$ tail -n 3 /etc/passwd | tr ':' '|'
```

7. \*Подсчитайте с помощью команды `wc` сколько в получившемся потоке имеется вертикальных черт.

**Ответ:**

```
$ tail -n 3 /etc/passwd | tr ':' '|' | tr -dc '|' | wc -m
```

8. \*Сколько символов – десятичных цифр содержится в файле `/etc/sysctl.conf` ?

**Ответ:**

```
$ cat /etc/sysctl.conf | tr -dc 0-9 | wc -m
```

9. \*Получите список всех PID процессов, связанных с терминалами, так, чтобы весь список был выведен в одну строку, а после каждого PID процесса был указан соответствующий ему терминал. Подсказка: выделите нужные поля с помощью awk из списка всех процессов в системе.

**Ответ:**

```
$ ps -eo pid,TTY | awk '$2!="?" && $2!="TT" {printf $0"\t"}'
```

10. Получите пронумерованный список всех пользователей (имена получите из /etc/passwd) в системе так, чтобы он выводился в виде одной строки. Для номера должно отводиться три позиции (разряда).

**Ответ:**

```
$ cut -f1 -d: /etc/passwd | nl -w3 | awk '{printf $0"\t"}'
```

11. Измените предыдущую команду, добавив в нее sed и tr, так, чтобы перед каждым номером строки были вставлены строки «+++++» и после самих строк учетных записей были вставлены строки «-----».

**Ответ:**

```
$ cut -f1 -d: /etc/passwd | nl -w3 | awk '{printf $0"\t"}' | tr '\t' '\n' | sed -e '/^..[0-9]$/i+++++' -e '/^[a-zA-Z]/a-----'
```

## 9.7. Создание команд из потока

1. \* В домашнем каталоге найдите все созданные в течение последних трех дней не скрытые пустые каталоги и удалите их, используя команду xargs. Поиск должен осуществляться не с родительских каталогов к дочерним, а наоборот.

**Решение:**

**Предварительно создадим несколько цепочек каталогов:**

```
$ mkdir -p .d1/d2/d3/d4 d1/d2/d3/.d4 d1/d2/d3/d4 d2/d2/d3/d4 d3/d2/d3/d4 d4/d2/d3/d4
```

**Теперь найдем нужные каталоги:**

```
$ find ~/* -depth -type d -empty -name '[!.]*'
```

не скрытые ~/\* и -name '[!.]\*'

**Поиск должен осуществляться не с родительских каталогов к дочерним, а наоборот: -depth**

**пустые -empty**

**каталоги -type d**

**Попробуем удалить из с помощью xargs и проверим результат:**

```
$ find ~/* -depth -type d -empty -name '[!.]*' | xargs rmdir
```

```
$ find ~/* -depth -type d -empty -name '[!.]*'
```

**Результат:**

/home/user/d2/d2/d3

/home/user/d3/d2/d3

/home/user/d4/d2/d3

Снова есть такие каталоги. Это пример, когда команда `xargs` не уместна.

Правильный путь:

```
$ find ~/* -depth -type d -empty -name '[!.]*' -delete
```

или

```
$ find ~/* -depth -type d -empty -name '[!.]*' -exec rmdir {} \;
```

2. \*С помощью какой команды можно было бы сравнить быстродействие обоих вариантов команды `find`, приведенной выше?

Ответ:

`time`

Пример:

```
$ time find ~ -type f | xargs ls -sh | tail -1
```

```
4.0K /home/user/time.txt
```

```
real    0m0.006s
```

```
user    0m0.002s
```

```
sys     0m0.007s
```

```
$ time find ~ -type f -exec ls -sh {} \; | tail -1
```

```
2.0M /home/user/r2.bin
```

```
real    0m0.077s
```

```
user    0m0.016s
```

```
sys     0m0.064s
```

Видим что второй вариант почти в 13 раз медленней. Т.к. в первом случае команда `ls` запускается 1 раз, а во втором столько раз сколько файлов в домашнем каталоге.

## Глава 10. Текстовые редакторы.

1. Запустите vimtutor и выполните в нем задания.

Ответ:

`$ vimtutor`

# Глава 11. Работа с дисками и файловыми системами

## 11.1. Управление разделами дисков.

Предполагается что вы выполняете задание на виртуальной машине, в которую вы предварительно добавили два новых диска. Размер добавляемых дисков не менее 8Гб.

1. Проверьте текущее разбиение дисков на разделы.

**Ответ:**

```
# fdisk -l
```

2. Выполните команду `sfdisk -s`. Узнайте, что она выдает.

**Ответ: Кол-во блоков на дисках**

3. На каком разделе вашего диска имеется область подкачки `swap`?

**Ответ:**

```
# fdisk -l
```

4. В интерактивном режиме команды `fdisk` получите информацию о таблице разделов на одном из ваших новых дисков.

**Ответ:**

```
# fdisk /dev/sdb
```

```
> p
```

5. С помощью команды `fdisk` создайте на одном из дисков таблицу разделов типа DOS и:

раздел размером 2Гб с типом «Linux filesystem»;

раздел размером 2Гб с типом «Linux swap».

**Ответ:**

```
> n
```

```
> t
```

```
> 83 - Linux filesystem
```

```
> n
```

```
> t
```

```
> 2
```

```
> 82 - Linux swap
```

```
> w
```

6. На втором диске создайте таблицу разделов GPT. Затем два раздела по 2Гб типа «Linux filesystem» и «Linux swap».

**Ответ:**

```
# fdisk /dev/sdc
```

```
> g
> n
> t
> 83
> n
> t
> 2
> 82
> w
```

7. Определите код типа раздела для физического тома системы Linux LVM (Logical Volume Manager). Для таблицы разделов типа DOS и типа GPT.

**Ответ:**

```
# fdisk /dev/sdb
> t
> L
```

Внимание! При ошибочных действиях все данные на диске могут быть утрачены! При создании раздела на диске, где размещена корневая файловая система, возможно потребуется перезагрузка после создания раздела.

## 11.2. Управление файловыми системами.

1. Создайте на вновь созданном разделе диска файловую систему XFS.

**Ответ:**

```
# mkfs -t xfs /dev/sdb1
```

2. Создайте на другом разделе файловую систему EXT2. Отличается ли информация, выдаваемая командой создания файловой системы, от той, которую выдавала предыдущая команда?

**Ответ:**

```
# mkfs -t ext2 /dev/sdc1
```

3. Создайте на том же разделе, где ранее создавали EXT2 файловую систему EXT4 с предварительной проверкой поверхности диска на плохие блоки. Файловая система должна иметь размер блока 2048 байт.

**Ответ:**

```
# mkfs -t ext4 /dev/sdc1 -c
```

4. Определите, где находится первая копия суперблока в только что созданной файловой системе.

**Ответ:**

```
# dumpe2fs /dev/sdc1
```

5. Проверьте целостность только что файловой системы.

**Ответ:**

```
# e2fsck /dev/sdc1
```

6. Выполните команду проверки этой же файловой системы так, чтобы была выполнена ее полная проверка.

**Ответ:**

```
# e2fsck -f /dev/sdc1
```

### 11.3. Монтирование файловых систем.

1. Для файловой системы EXT4, созданной в предыдущих параграфах, создайте каталог - точку монтирования /storage1 и смонтируйте эту файловую систему.

**Ответ:**

```
# mkdir /storage1
```

```
# mount /dev/sdc1 /storage1
```

2. Какой каталог находится в /storage1?
3. Для файловой системы XFS создайте каталог - точку монтирования /storage2 и смонтируйте эту файловую систему

**Ответ:**

```
# mkdir /storage2
```

```
# mount /dev/sdb1 /storage2
```

4. Имеется ли в ФС /storage2 какие либо файлы или каталоги?

**Ответ: . . .**

5. Перейдите в каталог - точку монтирования. Попробуйте размонтировать файловую систему. Что при этом происходит?

**Ответ:**

```
# cd /storage1
```

```
# umount
```

**Провал, тк ФС используется**

6. Какая команда позволяет узнать, какой процесс использует файл устройства?

**Ответ: fuser**

7. Что надо сделать для размонтирования устройства?

**Ответ: Перейти в каталог вне смонтированной ФС**

8. Создайте файл подкачки /.swap1 размером 64Мб.

**Ответ:**

```
# dd if=/dev/zero of=/.swap1 count=1 bs=64M
```

```
# mkswap /.swap1
```

9. Подключите созданный файл подкачки.

**Ответ:**

```
# swapon /.swap1
```

10. Проверьте, используется ли он.

Ответ:

```
# swapon -s
```

11. Настройте автоматическое монтирование каталога /storage1 в /etc/fstab.

Ответ:

```
# blkid /dev/sdc1
```

```
/dev/sdc1: UUID="457968bb-fef4-4030-95b3-84a18a3b2e6c" BLOCK_SIZE="4096"  
TYPE="ext4" PARTUUID="b51c04b0-63d9-2c49-8a0d-60eb3db66cc5"
```

```
# vi /etc/fstab
```

```
UUID=457968bb-fef4-4030-95b3-84a18a3b2e6c /storage1 ext4 defaults 0  
0
```

12. Для каталога /storage2 опишите автоматическое монтирование посредством юнита systemd.

Ответ:

```
# blkid /dev/sdb2
```

```
/dev/sdb2: UUID="afd4f037-c9d3-47c0-8bbc-c8b468d74777" BLOCK_SIZE="512"  
TYPE="xfs" PARTUUID="50f6f004-5436-a449-9057-c7a97795f7ef"
```

```
# cat /etc/systemd/system/storage2.mount:
```

```
[Unit]
```

```
Description=filestorage
```

```
[Mount]
```

```
What=/dev/disk/by-uuid/afd4f037-c9d3-47c0-8bbc-c8b468d74777
```

```
Where=/storage2
```

```
Type=xfs
```

```
Options=default
```

```
[Install]
```

```
WantedBy=graphical.target
```

13. Опишите подключение созданных вами ранее разделов подкачки в /etc/fstab.

Ответ:

```
# mkswap /dev/sdb2
```

```
# mkswap /dev/sdc2
```

```
# blkid | grep swap
```

```
B /etc/fstab
```

```
UUID="a8c885e0-b3ad-44cb-97eb-686c3b6eddf3" none swap defaults 0 0
UUID="34bd69c5-864e-4a1c-9c16-1bf497f31185" none swap defaults 0 0
```

14. Перезагрузите систему и убедитесь, что:

- ОС загружается.
- Монтируются каталоги /storage1 и /storage2
- Подключаются разделы подкачки.

## Глава 12. Файловая система Linux.

### 12.1. Изучение inode.

1. Определите номер inode родительского каталога корневого каталога.

**Ответ:**

```
$ ls -id /..
```

2. Сравните номер inode вашего домашнего каталога в /home и имени точка (.) в нем самом.

**Ответ:**

```
$ ls -id ~
```

```
$ ls -id ~/.
```

3. Определите количество имен у файла /bin/gzip.

**Ответ:**

```
# ls -l /bin/gzip
```

4. Получите подробную информацию об inode этого файла.

**Ответ:**

```
# stat /bin/gzip
```

5. Какая группа пользователей установлена для вашего домашнего каталога?

**Ответ:**

```
$ ls -ld ~
```

6. Какая группа пользователей установлена на файл /bin/ls и кто является его владельцем?

**Ответ:**

```
# ls -l /bin/ls
```

7. Получите список файлов в домашнем каталоге в подробном формате. Какие права доступа установлены на них?

**Ответ:**

```
# ls -l ~
```

### 12.2. Изучение прав доступа.

1. Переведите из восьмеричной формы записи прав доступа в символьную: 641.

**Ответ:**

```
rw-r----x
```

2. Переведите запись прав доступа rw-r----- в восьмеричную форму.

**Ответ: 640**

3. Определите, кто является владельцем домашнего каталога пользователя mail, какова группа пользователей.

**Ответ:**

**\$ ls -ld ~mail**

4. Имеется файл скрипта Bash script.sh с правами 750, владелец root, группа sys. Может ли пользователь user2, являющийся членом групп user2 и mail, что-либо изменить в тексте файла?

**Ответ: Нет**

5. Может ли пользователь user2 запустить на исполнение этот файл?

**Ответ: Нет**

6. Какие права должны быть добавлены на файл, чтобы user2 мог читать и исполнять этот файл?

**Ответ: Чтение и исполнение в последней триаде**

7. Определите какие права установлены на ваш домашний каталог.

**Ответ:**

**\$ ls -ld ~**

8. Какие права установлены на домашний каталог пользователя mail?

**Ответ:**

**\$ ls -ld ~mail**

9. На каталог d1 установлены права drwx--x---. Владелец каталога adm, группа sys. Может ли пользователь user1, принадлежащий группе sys, переименовать файл f1 в этом каталоге?

**Ответ: Нет**

10. Мог бы пользователь user1 узнать, какие права установлены на файл f1 ?

**Ответ: Нет**

11. Предположим, что на файл f1 установлены права 640, владелец adm, группа sys. Сможет ли user1 просмотреть содержимое файла? Сможет ли он запустить на исполнение этот файл?

**Ответ: Да; нет**

12. Если бы на этот каталог были бы установлены права 751, то смог бы пользователь user1 переименовать файл f1 ?

**Ответ: Нет**

13. Какие минимальные права и на какие объекты должны быть установлены, чтобы user1 смог удалить файл f1 из каталога d1 ?

**Ответ: 030 на d1**

14. В вашем домашнем каталоге находится файл enigma с правами 600, владельцем root и группой root. Можете ли вы удалить этот файл?

**Ответ: Да**

### 12.3. Изменение прав доступа.

1. Находясь в сеансе обычного пользователя, попытайтесь изменить права владения на любой файл в вашем домашнем каталоге. Что происходит?

**Ответ:**

```
$ chown root f1
```

```
chown: changing ownership of 'f1': Operation not permitted
```

2. Перейдите в сеанс суперпользователя и, находясь в домашнем каталоге обычного пользователя (с именем которого вы осуществляете повседневную работу), измените права владения на любой файл так, чтобы он принадлежал пользователю daemon.

**Ответ:**

```
$ sudo chown daemon f1
```

3. Измените группу того же файла на sys.

**Ответ:**

```
$ sudo chgrp sys f1
```

4. С помощью одной команды измените владельца и группу пользователей этого файла на исходные в POSIX нотации.

**Ответ:**

```
$ sudo chown admuser:admuser f1
```

5. Создайте цепочку каталогов d1/d2/d3, а в них - файлы d1/f1, d1/d2/f2, d1/d2/d3/f3 . Установите на файл d1/f1 права 400. Можете ли вы изменить этот файл?

**Ответ:**

```
$ mkdir -p d1/d2/d3
```

```
$ touch d1/f1
```

```
$ touch d1/d2/f2
```

```
$ touch d1/d2/d3/f3
```

```
$ chmod 400 d1/f1
```

```
$ echo 123 > d1/f1
```

```
-bash: d1/f1: Permission denied
```

6. Рекурсивно добавьте права на чтение и запись для каталога d1. Можете вы теперь изменить файл f1?

**Ответ:**

```
$ chmod -R u+rw d1
```

```
$ echo 123 > d1/f1
```

7. Отнимите у каталога d1 права на чтение и запись. Можно ли теперь получить информацию о файлах в этом каталоге?

**Ответ:**

```
$ chmod u-rw d1
```

```
$ ls -l d1
```

```
ls: cannot open directory 'd1': Permission denied
```

8. С помощью команды `find` при условии установки `-exec` измените права на каталоги и все подкаталоги `d1` на `750`.

**Ответ:**

```
$ find d1 -type d -exec chmod 750 {} \;
```

9. Командами `find`, `xargs` и `chmod` установите права на все обычные файлы в каталоге `d1` равными `-rw-r--rw-`.

**Ответ:**

```
$ find d1 -type f | xargs chmod 640
```

#### 12.4. `umask` и специальные биты.

1. Установите значение `umask 000`. Проверьте с какими правами создаются новые каталоги и файлы.

**Ответ:**

```
# umask 000
```

```
# > test3
```

```
# mkdir d8
```

```
# ls -l
```

2. В символической форме установите такое значение `umask`, чтобы вновь создаваемые файлы имели права доступа `644`.

**Ответ:**

```
# umask u=rw, g=r, o=r
```

3. Требуется, чтобы владелец создаваемого каталога мог создавать, читать и записывать файлы каталог, а также переходить в него. Члены группы владельцев должны иметь права на создание и удаление файлов в этом каталоге. Все остальные никаких прав иметь не должны. Какое должно быть значение маски для того, чтобы удовлетворить перечисленным требованиям?

**Ответ: 027**

4. Проверьте, какие биты прав доступа установлены на исполняемый файл команды `passwd`.

**Ответ:**

```
$ ls -l /etc/passwd
```

5. С помощью `awk` получите список всех процессов в системе, для которых `RUID` не равен `EUID`.

**Ответ:**

```
$ ps -eo ruid,euid,cmd | awk '$1!=$2{print $0}'
```

6. На какой-либо виртуальной консоли запустите команду `passwd` от имени обычного пользователя. Выполните ту же команду, что и в предыдущем пункте. Имеется ли в требуемом списке процесс `passwd`?

**Ответ:** Да имеются. В списке должен появиться процесс `passwd`.

7. Используя `find` запишите в файл `SUGID.txt` все имена файлов, на которые установлены биты `SUID` или `SGID`.

**Ответ:**

```
# find -type f -perm /6000 > SUGID.txt
```

8. Найдите в системе все каталоги с установленными битами `SGID` или `Sticky bit`.

**Ответ:**

```
# find -type d -perm /3000
```

9. Создайте каталог и (будучи `root`) установите на этот каталог группу `users`. Установите бит `SGID` на этот каталог. В сеансе обычного пользователя создайте в этом каталоге файл. Проверьте, кому он принадлежит и какая группа пользователей у этого файла.

**Ответ:**

```
# mkdir d1
```

```
# chgrp users d1
```

```
# chmod g+s d1
```

```
# touch ./d1/f1
```

```
# ls -l ./d1
```

## 12.5. Типы файлов.

1. \*Найдите каталог, в котором находится наибольшее число файлов - разделяемых библиотек.

**Ответ:**

```
# find / -type f -name '*.so*' | xargs file | awk -F/ '{NF=""; OFS="/"; print $0}' | sort | uniq -dc | sort -nk1 | tail -1
```

2. Какой каталог является основным местом хранения системных заголовочных файлов?

**Ответ:** `/usr/include`

3. Определите, какие данные находятся в файле (файлах), имеющем имя вида `/boot/init*.img`.

**Ответ:**

```
# file /boot/init*.img
```

4. Найдите в системе все файлы именованных каналов и сокетов.

**Ответ:**

```
# find -type p,s
```

5. Какой размер имеют файлы именованных каналов и сокетов?

**Ответ: 0**

6. Имеются ли в системе файлы устройств, принадлежащие вам?

**Ответ:**

**# find -type b,c -user sa**

## 12.6. Ссылки.

1. Создайте в текущем каталоге файл file1 и жесткую связь с файлом hlink1.

**Ответ:**

**\$ touch file1**

**\$ ln file1 hlink1**

**\$ ls -il file1 hlink1**

2. Можете ли вы перенести этот файл в каталог /tmp и почему?

**Ответ:**

**\$ mv hlink1 /tmp \$ ls -il /tmp/hlink1**

3. Попробуйте создать жесткую связь командой ln с домашним каталогом. Удастся ли это?

**Ответ:**

**\$ ln ~ hhome1**

4. Как можно создать жесткую связь с именем домашнего каталога? Какое может быть имя у этой связи и с помощью какой команды она создается?

**Ответ:**

**\$ mkdir d3**

**\$ ls -id ~ d3/..**

5. Попробуйте создать в домашнем каталоге жесткую связь с файлом устройства /dev/null.

**Ответ:**

**\$ ln /dev/null hnull1**

6. Создайте символическую ссылку на каталог /usr/share/doc в вашем домашнем каталоге. Попробуйте сделать команду cd ~/doc.

**Ответ:**

**\$ ln -s /usr/share/doc doc**

**\$ cd ~/doc**

7. Получите список файлов - символических ссылок, находящихся в каталоге /usr.

**Ответ:**

**# find /usr -type l**

8. Создайте в домашнем каталоге символическую ссылку на исполняемый файл /bin/ls. Попробуйте воспользоваться новой ссылкой.

**Ответ:**

```
$ ln -s /bin/ls l
```

```
$ l
```

## Глава 13. Процессы.

1. Найдите пустые файлы в домашнем каталоге в фоновом режиме.

**Ответ:**

```
$ find ~ -empty &
```

2. Запустите в фоновом режиме два задания: sleep 200 и sleep 2000, выведите информацию о состоянии заданий.

**Ответ:**

```
$ sleep 200 &
```

```
$ sleep 2000 &
```

```
$ jobs
```

3. Снимите с выполнение 2-е задание, выведите информацию о заданиях.

**Ответ:**

```
$ kill %2
```

```
$ jobs
```

4. Как изменяется общая картина жизненного цикла процесса в системе по сравнению с описанным, если процесс запускается в фоновом режиме.

**Ответ: не выполняется системный вызов wait()**

5. Выполните команду exes ls. Изучите ее поведение и обсудите отличия жизненного цикла процесса, возникающего в таком случае, от описанного выше.

**Ответ: не было системного вызова fork() и соответственно не запускался дочерний процесс оболочки, а сразу произошло замещение exes() кода процесса оболочки на программу ls, которая завершив работу выполнила exit() и сеанс закрылся**

6. Получите информацию о процессах в обычном и подробном форматах.

**Ответ:**

```
$ ps
```

```
$ ps -l
```

7. Выведите список всех процессов, запущенных всеми пользователями, вошедшими в сеанс, в системе.

**Ответ:**

```
$ ps -u
```

8. Получите список процессов, не связанные с терминалами.

**Ответ:**

```
$ ps -t -
```

9. Проследите в течение минуты поведение процессов, использующих наибольшее процессорное время, используя наиболее подходящую для этого утилиту.

**Ответ:**

**\$ top**

10. Выведите PID процессов, использующих файл блочного устройства /dev/sda1.

**Ответ:**

**\$ sudo fuser -m /dev/sda1**

**без sudo не работает в сеансе пользователя или делать в сеансе root.**

**Еще может и не быть такого диска. Или диск может вообще не присутствовать, например если линукс запущен в контейнере.**

11. Получите иерархический список процессов с помощью команды ps.

**Ответ:**

**\$ ps auxf**

**или**

**\$ ps -eH**

12. Запустите порожденную оболочку Bash. Исследуйте, посылая родительской оболочке Bash сигналы TERM, INT, QUIT и HUP, что при этом происходит.

**Ответ:**

**\$ bash**

**\$ kill \$\$**

**\$ kill -2 \$\$**

**\$ kill -3 \$\$**

**\$ kill -1 \$\$**

13. От имени обычного!!! пользователя пошлите сигнал KILL любому процессу, запущенному от имени другого пользователя. Что произойдет?

**Ответ:**

**\$ ps aux**

**\$ kill -9 4965**

**должно быть permission denied**

14. Запустите в фоновом режиме команду sleep 1000. Проверьте, на какие сигналы из следующих: TERM, INT, QUIT и HUP, реагирует эта команда.

**Ответ:**

**\$ sleep 1000 &**

**\$ kill %%**

**Terminated процесс завершен**

**\$ sleep 1000 &**

**\$ kill -2 %%**

**Interrupt процесс завершен**

**\$ sleep 1000 &**

```
$ kill -3 %%
```

**Quit процесс завершен**

```
$ sleep 1000 &
```

```
$ kill -1 %%
```

**Hangup процесс завершен**

15. Запрограммируйте оболочку так, чтобы при получении ей сигнала TERM создавался файл pwd.txt, содержащий информацию о текущем каталоге.

**Ответ:**

```
$ trap "pwd > ~/pwd.txt" TERM
```

```
$ kill $$
```

```
$ cat ~/pwd.txt
```

16. Запустите порожденную оболочку. Работает ли в ней созданный обработчик?

**Ответ: нет**

17. От имени обычного пользователя попытайтесь запустить оболочку Bash со значением nice number равным -1. Какое сообщение выводится?

**Ответ:**

```
$ nice -n -1 bash
```

**Получим Permission denied**

18. Будучи суперпользователем измените приоритет для всех процессов пользователя user на +5.

**Ответ:**

```
# renice 5 -u user
```