

Лабораторная работа №6

Поиск файлов. Перенаправление ввода-вывода. Просмотр запущенных процессов

Латыпова Диана. НФИбд-02-21

Содержание

1	Цель работы	5
2	Задание	6
3	Выполнение лабораторной работы	7
4	Контрольные вопросы	18
5	Выводы	22

Список иллюстраций

3.1	Запись названия файлов в file.txt	7
3.2	Названия файлов, содержащихся в домашнем каталоге	8
3.3	Файлы с расширением .conf	9
3.4	Файлы, каталоги, подкаталоги, имеющие имена, начинающиеся с символа с (1)	9
3.5	Файлы, каталоги, подкаталоги, имеющие имена, начинающиеся с символа с (1)	9
3.6	Вывод файла, начинающегося с символа с(1)	10
3.7	Вывод файла, начинающегося с символа с(2)	10
3.8	Вывод файлов, начинающиеся с символа h из каталога /etc	10
3.9	Файлы, имена которых начинаются с log	11
3.10	Содержимое logfile	11
3.11	Удаление logfile	11
3.12	Запуск gedit	12
3.13	Определение идентификатора процесса gedit(1)	12
3.14	Определение идентификатора процесса gedit(2)	13
3.15	man kill	13
3.16	Справка kill	13
3.17	Завершение процесса gedit	14
3.18	man df	14
3.19	man du	15
3.20	df	15
3.21	df -h	16
3.22	du	16
3.23	du -h	16
3.24	Справка find	17
3.25	Вывод имен всех директорий, имеющиеся в домашнем каталоге .	17

Список таблиц

1 Цель работы

Ознакомление с инструментами поиска файлов и фильтрации текстовых данных. Приобретение практических навыков: по управлению процессами (и заданиями), по проверке использования диска и обслуживанию файловых систем.

2 Задание

1. Осуществите вход в систему, используя соответствующее имя пользователя.
2. Запишите в файл `file.txt` названия файлов, содержащихся в каталоге `/etc`. Допишите в этот же файл названия файлов, содержащихся в вашем домашнем каталоге.
3. Выведите имена всех файлов из `file.txt`, имеющих расширение `.conf`, после чего запишите их в новый текстовый файл `conf.txt`.
4. Определите, какие файлы в вашем домашнем каталоге имеют имена, начинающиеся с символа `s`? Предложите несколько вариантов, как это сделать.
5. Выведите на экран (по странично) имена файлов из каталога `/etc`, начинающиеся с символа `h`.
6. Запустите в фоновом режиме процесс, который будет записывать в файл `~/logfile` файлы, имена которых начинаются с `log`.
7. Удалите файл `~/logfile`.
8. Запустите из консоли в фоновом режиме редактор `gedit`.
9. Определите идентификатор процесса `gedit`, используя команду `ps`, конвейер и фильтр `grep`. Как ещё можно определить идентификатор процесса?
10. Прочтите справку (`man`) команды `kill`, после чего используйте её для завершения процесса `gedit`.
11. Выполните команды `df` и `du`, предварительно получив более подробную информацию об этих командах, с помощью команды `man`.
12. Воспользовавшись справкой команды `find`, выведите имена всех директорий, имеющих в вашем домашнем каталоге.

3 Выполнение лабораторной работы

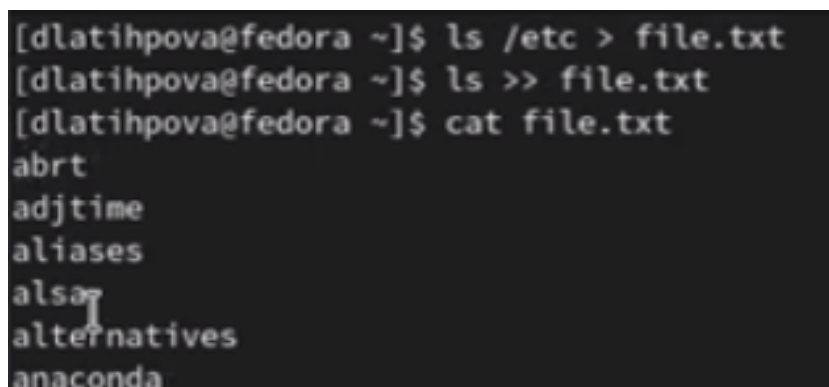
Первым заданием было войти в систему, мы уже осуществили вход в нее, поэтому я сразу перешла к выполнению второго задания.

2.Используя команду `ls` и перенаправления ввода-вывода, я записала в файл `file.txt` названия файлов, содержащихся в каталоге `/etc`, а также следующей командой `ls` и `»` дописала в этот же файл названия файлов, содержащихся в нашем домашнем каталоге. С помощью команды `cat` просмотрела содержимое файла `file.txt` (рис. 3.1)(рис. 3.2)

1 `ls /etc > file.txt`

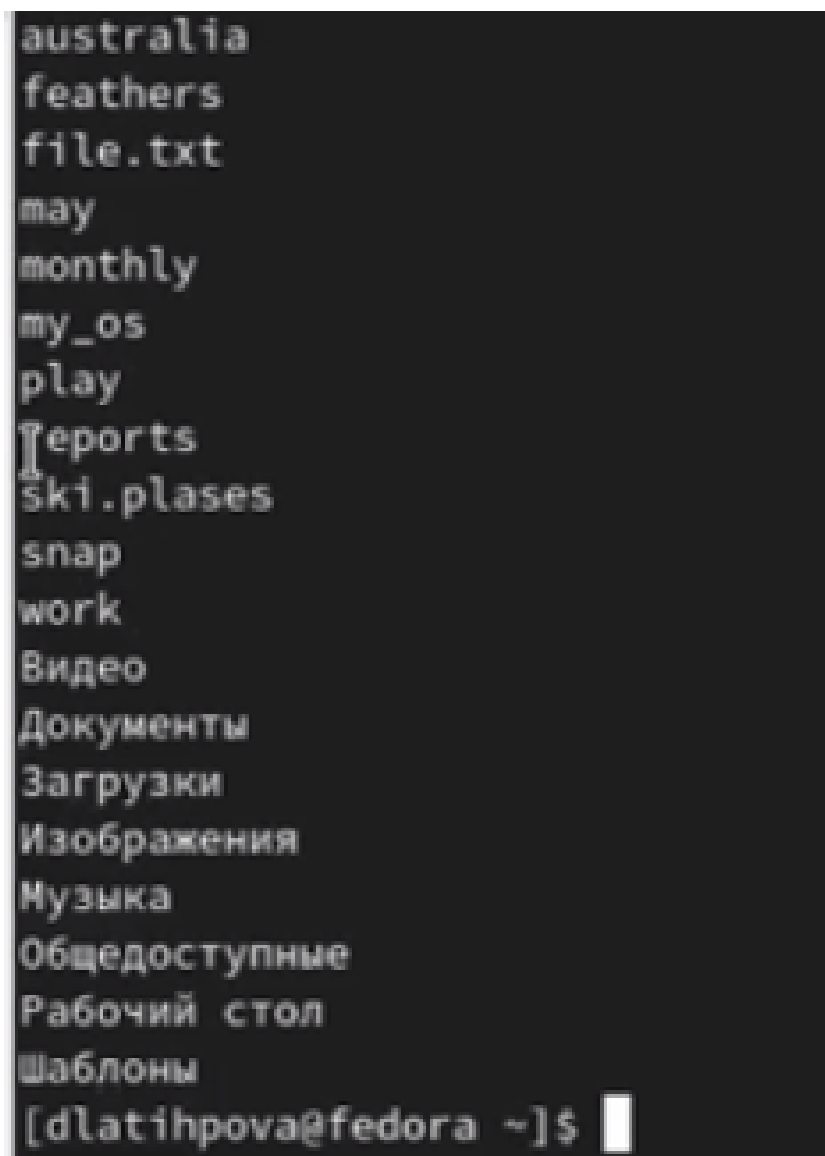
2 `ls » file.txt`

3 `cat file.txt`



```
[dlatihpova@fedora ~]$ ls /etc > file.txt
[dlatihpova@fedora ~]$ ls >> file.txt
[dlatihpova@fedora ~]$ cat file.txt
abrt
adjtime
aliases
alsa
alternatives
anaconda
```

Рис. 3.1: Запись названия файлов в `file.txt`

A terminal window with a dark background. It displays a list of files and directories in the home directory. The files are: australia, feathers, file.txt, may, monthly, my_os, play, reports, ski.places, snap, work, Видео, Документы, Загрузки, Изображения, Музыка, Общедоступные, Рабочий стол, and Шаблоны. At the bottom, the prompt is [dlatihpova@fedora ~]\$ followed by a cursor.

```
australia
feathers
file.txt
may
monthly
my_os
play
reports
ski.places
snap
work
Видео
Документы
Загрузки
Изображения
Музыка
Общедоступные
Рабочий стол
Шаблоны
[dlatihpova@fedora ~]$
```

Рис. 3.2: Названия файлов, содержащихся в домашнем каталоге

3.Далее с помощью команд `cat`, `grep` и конвейера вывела имена всех файлов из `file.txt`, имеющих расширение `.conf` и записала их в новый текстовый файл `conf.txt`(рис. 3.3).

```
1 cat file.txt | grep .conf > conf.txt
```

```
2 cat conf.txt
```



```
[dlatihpova@fedora ~]$ cat file.txt | grep .conf > conf.txt
[dlatihpova@fedora ~]$ cat conf.txt
anthy-unicode.conf
appstream.conf
asound.conf
brltty.conf
chkconfig.d
chrony.conf
dconf
dleyna-renderer-service.conf
dleyna-server-service.conf
dnsmasq.conf
```

Рис. 3.3: Файлы с расширением .conf

4. Следующим заданием нужно было определить, какие файлы в моем домашнем каталоге имеют имена, начинавшиеся с символа с. И предложить несколько вариантов, как это сделать.

Сначала я попробовала команду:

```
**find ~ -name "с*" -print**
```

Однако, консоль вывела не только все файлы, каталоги, но и подкаталоги, имеющие имена, начинавшиеся с символа с (рис. 3.4) (рис. 3.5).

```
[dlatihpova@fedora ~]$ find ~ -name "с*" -print
/home/dlatihpova/.mozilla/firefox/gw87ofbz.default-release/crashes
/home/dlatihpova/.mozilla/firefox/gw87ofbz.default-release/compatibility.ini
/home/dlatihpova/.mozilla/firefox/gw87ofbz.default-release/cookies.sqlite
/home/dlatihpova/.mozilla/firefox/gw87ofbz.default-release/cert9.db
/home/dlatihpova/.mozilla/firefox/gw87ofbz.default-release/storage/permanent/chrome
/home/dlatihpova/.mozilla/firefox/gw87ofbz.default-release/storage/default/https+++
www.youtube.com/cache
/home/dlatihpova/.mozilla/firefox/gw87ofbz.default-release/storage/default/https+++
www.youtube.com/cache/caches.sqlite
```

Рис. 3.4: Файлы, каталоги, подкаталоги, имеющие имена, начинающиеся с символа с (1)

```
/home/dlatihpova/work/didi/public/css
/home/dlatihpova/work/didi/public/media/icons/brands/courser
/home/dlatihpova/work/didi/public/categories
/home/dlatihpova/work/didi/public/category
/home/dlatihpova/work/dlatypova.github.io/.git/hooks/commit-
/home/dlatihpova/work/dlatypova.github.io/.git/config
/home/dlatihpova/conf.txt
```

Рис. 3.5: Файлы, каталоги, подкаталоги, имеющие имена, начинающиеся с символа с (1)

Поэтому я немного изменила команду, добавив maxdepth 1:

find ~ -maxdepth 1 -name "с*" -print

После чего нам вывелся лишь один файл conf.txt(рис. 3.6).

```
[dlatihpova@fedora ~]$ find ~ -maxdepth 1 -name "с*" -print  
/home/dlatihpova/conf.txt
```

Рис. 3.6: Вывод файла, начинающегося с символа с(1)

Далее я попробовала другой способ, с помощью команд ls и grep, вывести файлы, начинающиеся с символа с(рис. 3.7):

ls | grep с*

```
[dlatihpova@fedora ~]$ ls |grep с*  
conf.txt
```

Рис. 3.7: Вывод файла, начинающегося с символа с(2)

5.Далее с помощью похожей выше команды я вывела на экран (по странично) имена файлов, начинающиеся с символа h, только уже из определенного каталога /etc(рис. 3.8).

find /etc -maxdepth 1 -name "h*" - print

```
[dlatihpova@fedora ~]$ find /etc -maxdepth 1 -name "h*" -print  
/etc/hp  
/etc/httpd  
/etc/host.conf  
/etc/hosts  
/etc/hostname
```

Рис. 3.8: Вывод файлов, начинающиеся с символа h из каталога /etc

6.Далее с помощью следующей команды, добавив в конце амперсанду, я запустила в фоновом режиме процесс, который записывал в файл logfile файлы, имена которых начинаются с log(рис. 3.9).

find / -name "log*" -print >logfile &

8.Запустила в фоновом режиме с помощью амперсанды редактор gedit(рис. 3.12).

gedit &

```
[dlatihpova@fedora ~]$ gedit &
[1] 5911
[dlatihpova@fedora ~]$ ps
  PID TTY          TIME CMD
 5641 pts/1        00:00:00 bash
 5911 pts/1        00:00:00 gedit
 5936 pts/1        00:00:00 ps
```

Рис. 3.12: Запуск gedit

9.Далее я определила идентификатор данного процесса gedit, используя команду ps, конвейер и фильтр grep(рис. 3.12). Кроме них, использовала команды pidof и pstree(рис. 3.13)(рис. 3.14).

1 **ps**

2 **pgrep gedit**

3 **pidof gedit**

4 **pstree -p | grep gedit**

Идентификатор процесса gedit - **5911**

```
[dlatihpova@fedora ~]$ ps
  PID TTY          TIME CMD
 5641 pts/1        00:00:00 bash
 5911 pts/1        00:00:00 gedit
 5936 pts/1        00:00:00 ps
[dlatihpova@fedora ~]$ pgrep gedit
5911
```

Рис. 3.13: Определение идентификатора процесса gedit(1)

```
[dlatihpova@fedora ~]$ pidof gedit
5911
[dlatihpova@fedora ~]$ pstree -p | grep gedit
      |               | -gnome-terminal-(4138) --bash(5641) --gedit(5911) --{gedit}
(5916) |               |               |               |               |
(5917) |               |               |               |               |
(5919) |               |               |               |               |
```

Рис. 3.14: Определение идентификатора процесса gedit(2)

10.С помощью команды `man` прочла справку команды `kill`(рис. 3.15)(рис. 3.16):
man kill

```
[dlatihpova@fedora ~]$ man kill
```

Рис. 3.15: man kill

```
kill -l [number] | -L

DESCRIPTION
The command kill sends the specified signal to the specified processes or
process groups.

If no signal is specified, the TERM signal is sent. The default action for
this signal is to terminate the process. This signal should be used in
preference to the KILL signal (number 9), since a process may install a
handler for the TERM signal in order to perform clean-up steps before
terminating in an orderly fashion. If a process does not terminate after a
TERM signal has been sent, then the KILL signal may be used; be aware that
the latter signal cannot be caught, and so does not give the target process
the opportunity to perform any clean-up before terminating.

Most modern shells have a builtin kill command, with a usage rather similar
to that of the command described here. The --all, --pid, and --queue options,
and the possibility to specify processes by command name, are local
extensions.
```

Рис. 3.16: Справка kill

Я запустила gedit несколько раз, поэтому его идентификатор изменился на 5060

После чего завершила процесс gedit(рис. 3.17):

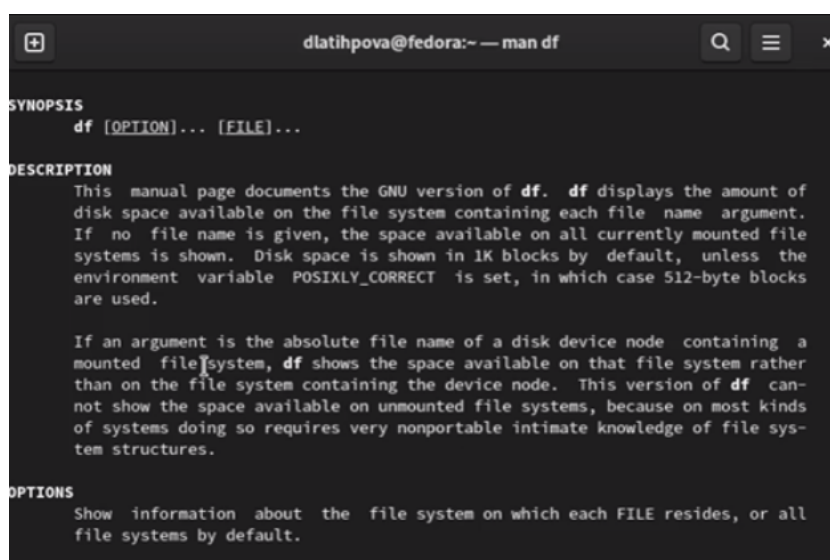
1 **kill 5060**

2 **ps**

```
[dlatihpova@fedora ~]$ kill 5060
[dlatihpova@fedora ~]$ ps
  PID TTY          TIME CMD
 4656 pts/0        00:00:00 bash
 5117 pts/0        00:00:00 ps
[1]+  Завершено      gedit
```

Рис. 3.17: Завершение процесса gedit

11. Далее с помощью man просмотрела информацию о командах df и du(рис. 3.18)(рис. 3.19).



```
SYNOPSIS
  df [OPTION]... [FILE]...

DESCRIPTION
  This manual page documents the GNU version of df. df displays the amount of
  disk space available on the file system containing each file name argument.
  If no file name is given, the space available on all currently mounted file
  systems is shown. Disk space is shown in 1K blocks by default, unless the
  environment variable POSIXLY_CORRECT is set, in which case 512-byte blocks
  are used.

  If an argument is the absolute file name of a disk device node containing a
  mounted file system, df shows the space available on that file system rather
  than on the file system containing the device node. This version of df can-
  not show the space available on unmounted file systems, because on most kinds
  of systems doing so requires very nonportable intimate knowledge of file sys-
  tem structures.

OPTIONS
  Show information about the file system on which each FILE resides, or all
  file systems by default.
```

Рис. 3.18: man df

```

dlatihpova@fedora:~ -- man du
DU(1)                                User Commands                                DU(1)

NAME
    du - estimate file space usage

SYNOPSIS
    du [OPTION]... [FILE]...
    du [OPTION]... --files0-from=F

DESCRIPTION
    Summarize disk usage of the set of FILES, recursively for directories.
    Mandatory arguments to long options are mandatory for short options too.

    -0, --null
        end each output line with NUL, not newline

    -a, --all
        write counts for all files, not just directories

    --apparent-size
        print apparent sizes, rather than disk usage; although the apparent

```

Рис. 3.19: man du

Поочередно выполнила команды df и du.

df используется для получения полной сводной информации об использовании доступного и используемого дискового пространства файловой системы в системе Linux(рис. 3.20).

```

[dlatihpova@fedora ~]$ df
Файловая система 1К-блоков  Использовано  Доступно  Использовано%  Смонтировано в
devtmpfs           4096             0    4096             0% /dev
tmpfs              1892704          172   1892532           1% /dev/shm
tmpfs              757084          1892   755192           1% /run
/dev/sda6          101350400      18203832   81631352        19% /
tmpfs              1892704          116   1892588           1% /tmp
/dev/sda6          101350400      18203832   81631352        19% /home
/dev/loop4         409088          409088           0        100% /var/lib/snapd/snap/obs-
studio/1284
/dev/loop0          128             128           0        100% /var/lib/snapd/snap/bare
/5
/dev/loop2         63488           63488           0        100% /var/lib/snapd/snap/core
20/1405
/dev/loop3         63488           63488           0        100% /var/lib/snapd/snap/core
20/1434
/dev/loop1        66816           66816           0        100% /var/lib/snapd/snap/etk-

```

Рис. 3.20: df

С помощью опции “-h” я вывела статистику дискового пространства файловой системы в удобном формате для чтения человеком, то есть размеры представлены в байтах, килобайтах, мегабайтах, гигабайтах и т.д. (рис. 3.21).

```
[dlatihpova@fedora ~]$ df -h
Файловая система  Размер  Использовано  Дост  Использовано%  Смонтировано в
devtmpfs          4,0M      0  4,0M           0% /dev
tmpfs             1,9G     172K  1,9G           1% /dev/shm
tmpfs            740M     1,9M  738M           1% /run
/dev/sda6         97G      18G   78G          19% /
tmpfs            1,9G     84K  1,9G           1% /tmp
/dev/sda6         97G      18G   78G          19% /home
/dev/loop4       400M     400M    0          100% /var/lib/snapd/snap/obs-studio
/1284
/dev/loop0       128K     128K    0          100% /var/lib/snapd/snap/bare/5
/dev/loop2       62M      62M    0          100% /var/lib/snapd/snap/core20/140
5
```

Рис. 3.21: df -h

du используется для показа суммарной информации об использовании диска файлами и директориями(рис. 3.22).

```
[dlatihpova@fedora ~]$ du
8      ./mozilla/extensions/{ec8030f7-c20a-464f-9b0e-13a3a9e97384}
8      ./mozilla/extensions
0      ./mozilla/plugins
0      ./mozilla/firefox/Crash Reports/events
8      ./mozilla/firefox/Crash Reports
0      ./mozilla/firefox/Pending Pings
0      ./mozilla/firefox/gw87ofbz.default-release/minidumps
0      ./mozilla/firefox/gw87ofbz.default-release/crashes/events
8      ./mozilla/firefox/gw87ofbz.default-release/crashes
2688   ./mozilla/firefox/gw87ofbz.default-release/security_state
0      ./mozilla/firefox/gw87ofbz.default-release/storage/permanent/chrome/idb/387011
2724rsegmnoittet-es.files/journals
```

Рис. 3.22: du

И также использовала опцию “-h”(рис. 3.23)

```
[dlatihpova@fedora ~]$ du -h
8,0K   ./mozilla/extensions/{ec8030f7-c20a-464f-9b0e-13a3a9e97384}
8,0K   ./mozilla/extensions
0      ./mozilla/plugins
0      ./mozilla/firefox/Crash Reports/events
8,0K   ./mozilla/firefox/Crash Reports
0      ./mozilla/firefox/Pending Pings
0      ./mozilla/firefox/gw87ofbz.default-release/minidumps
0      ./mozilla/firefox/gw87ofbz.default-release/crashes/events
8,0K   ./mozilla/firefox/gw87ofbz.default-release/crashes
2,7M   ./mozilla/firefox/gw87ofbz.default-release/security_state
0      ./mozilla/firefox/gw87ofbz.default-release/storage/permanent/chrome/idb/387011
2724rsegmnoittet-es.files/journals
```

Рис. 3.23: du -h

12.И наконец, я просмотрела справку команды find(рис. 3.24)


```
[dlatihpova@fedora ~]$ find --help
Usage: find [-H] [-L] [-P] [-Olevel] [-D debugopts] [path...] [expression]

путём по умолчанию является текущий подкаталог; выражение по умолчанию: -print
выражение может состоять из операторов, параметров, тестов и действий:
операторы (в нисходящем порядке; -and берется по умолчанию, если не дано других):
    ( EXPR )  ! EXPR  -not EXPR  EXPR1 -a EXPR2  EXPR1 -and EXPR2
    EXPR1 -o EXPR2  EXPR1 -or EXPR2  EXPR1 , EXPR2
positional options (always true): -daystart -follow -regextype

normal options (always true, specified before other expressions):
    -depth --help -maxdepth LEVELS -mindepth LEVELS -mount -noleaf
```

Рис. 3.24: Справка find

Введя следующую команду, вывела имена всех директорий, имеющихя в моем домашнем каталоге(рис. 3.25):

find / -maxdepth 1

```
[dlatihpova@fedora ~]$ find . -maxdepth 1
.
./.mozilla
./.bash_logout
./.bash_profile
./.bashrc
./.cache
./.config
./.local
./Рабочий стол
./Загрузки
./Шаблоны
./Общедоступные
./Документы
```

Рис. 3.25: Вывод имен всех директорий, имеющиеся в домашнем каталоге

4 Контрольные вопросы

1. Какие потоки ввода вывода вы знаете?

Ввод и вывод распределяется между тремя стандартными потоками:

`stdin` – стандартный ввод (клавиатура),

`stdout` – стандартный вывод (экран),

`stderr` – стандартная ошибка (вывод ошибок на экран).

Потоки также пронумерованы:

`stdin` – 0,

`stdout` – 1,

`stderr` – 2.

2. Объясните разницу между операцией `>` и `>>`.

`>>` – это оператор вывода, который перезаписывает существующий файл, а `>` также является оператором вывода, но добавляет данные в уже существующий файл.

3. Что такое конвейер?

Конвейер — некоторое множество процессов, для которых выполнено следующее перенаправление ввода-вывода: то, что выводит на поток стандартного вывода предыдущий процесс, попадает в поток стандартного ввода следующего процесса.

4. Что такое процесс? Чем это понятие отличается от программы?

Процесс - это: программа на стадии выполнения; “объект”, которому выделено процессорное время; асинхронная работа. Процесс состоит из адресного пространства памяти и набора структур данных. По сути, процесс это запущенная программа или служба.

5. Что такое PID и GID?

Идентификатор процесса (англ. Process IDentifier, PID) — уникальный номер (идентификатор) процесса в многозадачной операционной системе (ОС).

Кроме идентификационного номера пользователя с учётной записью связан идентификатор группы. Группы пользователей применяются для организации доступа нескольких пользователей к некоторым ресурсам. У группы, так же, как и у пользователя, есть имя и идентификационный номер — GID (Group ID).

6. Что такое задачи и какая команда позволяет ими управлять?

Задача - это истощить чьи-то ресурсы или поручить кому-то выполнение определенной работы.

Команда `kill` используется для отправки сигнала процессам. Наиболее часто используемым сигналом является `SIGKILL` или `-9`, который завершает данные процессы. Ее можно использовать для принудительного завершения работы приложений.

7. Найдите информацию об утилитах `top` и `htop`. Каковы их функции?

`top` (Table Of Processes) — каждые 2 секунды обновляет активные процессы. Колонка PID указывает на ID процесса, USER на пользователя и т.д. Удобно. Если процесс активен постоянно, но уследить за постоянно обновляющимся списком не всегда получается.

Более продвинутые утилиты командной строки Linux — `atop`, `htop`. Отображают не только активные процессы с обновлением в две секунды, но и контроль

над загрузкой и т.д. Утилиты полезны для системных администраторов, но для обычного пользователя все-таки перебор.

8. Назовите и дайте характеристику команде поиска файлов. Приведите примеры использования этой команды.

Команда `find` - это одна из наиболее важных и часто используемых утилит системы Linux. Это команда для поиска файлов и каталогов на основе специальных условий. Ее можно использовать в различных обстоятельствах, например, для поиска файлов по разрешениям, владельцам, группам, типу, размеру и другим подобным критериям.

-P - никогда не открывать символические ссылки.

-L - получает информацию о файлах по символическим ссылкам. Важно для дальнейшей обработки, чтобы обрабатывалась не ссылка, а сам файл.

-maxdepth - максимальная глубина поиска по подкаталогам, для поиска только в текущем каталоге установите 1.

-depth - искать сначала в текущем каталоге, а потом в подкаталогах.

-mount искать файлы только в этой файловой системе.

-version - показать версию утилиты `find`.

-print - выводить полные имена файлов.

9. Можно ли по контексту (содержанию) найти файл? Если да, то как?

Для поиска файла по содержимому проще всего воспользоваться командой `grep` (вместо `find`).

Пример:

`grep -r строка_поиска каталог`

10. Как определить объем свободной памяти на жёстком диске?

Проверка свободного места на диске в Linux с помощью команды `df` (и понимание ее вывода) Когда вы используете команду `df` для проверки места на диске, она

покажет набор «файловых систем» с их размером, используемым пространством и свободным пространством.

11. Как определить объем вашего домашнего каталога?

В операционных системах на базе Linux посмотреть размер папки (директории) можно с помощью команды `du`. Эта команда, выполняемая в консоли, позволяет оценить используемый объем места на жестком диске отдельно по папкам и файлам, просуммировать результат, узнать общий размер папки.

12. Как удалить зависший процесс?

Когда известен PID процесса, мы можем убить его командой `kill`. Команда `kill` принимает в качестве параметра PID процесса.

Вообще команда `kill` предназначена для отправки сигнала процессу. По умолчанию, если мы не указываем какой сигнал посылать, посылается сигнал `SIGTERM` (от слова *termination* — завершение). `SIGTERM` указывает процессу на то, что необходимо завершиться.

Команда `killall` в Linux предназначена для «убийства» всех процессов, имеющих одно и то же имя. Это удобно, так как нам не нужно знать PID процесса.

5 Выводы

Я ознакомилась с инструментами поиска файлов и фильтрации текстовых данных. А также я приобрела практические навыки: по управлению процессами (и заданиями), по проверке использования диска и обслуживанию файловых систем.