

# **Отчет по выполнению лабораторной работы №8**

**Операционные системы**

Гомазкова Алина

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>5</b>
<b>2</b>	<b>Задание</b>	<b>6</b>
<b>3</b>	<b>Теоретическое введение</b>	<b>7</b>
<b>4</b>	<b>Выполнение лабораторной работы</b>	<b>8</b>
<b>5</b>	<b>Выводы</b>	<b>16</b>
<b>6</b>	<b>Ответы на контрольные вопросы</b>	<b>17</b>

## Список иллюстраций

4.1	Запись в файл . . . . .	8
4.2	Вывод содержимого файла . . . . .	8
4.3	Добавление данных в файл . . . . .	9
4.4	Поиск файлов определенного расширения . . . . .	9
4.5	Запись в файл . . . . .	9
4.6	Поиск файлов, начинающихся с определенного элемента . . . . .	10
4.7	Поиск файлов, начинающихся с определенного элемента . . . . .	10
4.8	Поиск файлов, начинающихся с определенного элемента . . . . .	11
4.9	Создание фонового процесса . . . . .	11
4.10	Удаление файла . . . . .	12
4.11	Создание фонового процесса . . . . .	12
4.12	Поиск идентификатора процесса . . . . .	12
4.13	Чтение документации . . . . .	13
4.14	Удаление процесса . . . . .	13
4.15	Чтение документации . . . . .	13
4.16	Утилита df . . . . .	14
4.17	Утилита du . . . . .	14
4.18	Чтение документации . . . . .	14
4.19	Название рисунка . . . . .	15

## 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`. Определите, какие файлы в вашем домашнем каталоге имеют имена, начинавшиеся с символа `c`? Предложите несколько вариантов, как это сделать.
4. Выведите на экран (по странично) имена файлов из каталога `/etc`, начинающиеся с символа `h`.
5. Запустите в фоновом режиме процесс, который будет записывать в файл `~/logfile` файлы, имена которых начинаются с `log`.
6. Удалите файл `~/logfile`.
7. Запустите из консоли в фоновом режиме редактор `gedit`.
8. Определите идентификатор процесса `gedit`, используя команду `ps`, конвейер и фильтр `grep`. Как ещё можно определить идентификатор процесса?
9. Прочтите справку (`man`) команды `kill`, после чего используйте её для завершения процесса `gedit`.
10. Выполните команды `df` и `du`, предварительно получив более подробную информацию об этих командах, с помощью команды `man`.
11. Воспользовавшись справкой команды `find`, выведите имена всех директорий, имеющих в вашем домашнем каталоге.

### 3 Теоретическое введение

В системе по умолчанию открыто три специальных потока: – `stdin` — стандартный поток ввода (по умолчанию: клавиатура), файловый дескриптор 0; – `stdout` — стандартный поток вывода (по умолчанию: консоль), файловый дескриптор 1; – `stderr` — стандартный поток вывод сообщений об ошибках (по умолчанию: консоль), файловый дескриптор 2. Большинство используемых в консоли команд и программ записывают результаты своей работы в стандартный поток вывода `stdout`. Например, команда `ls` выводит в стандартный поток вывода (консоль) список файлов в текущей директории. Потоки вывода и ввода можно перенаправлять на другие файлы или устройства. Проще всего это делается с помощью символов `>`, `>>`, `<`, `<<`.

## 4 Выполнение лабораторной работы

Я вошла в систему под соответствующим именем пользователя, открыла терминал, записала в файл file.txt названия файлов из каталога /etc с помощью перенаправления “>” (и файл создала, и записала в него то, что могло быть выведено ls -lR /etc). В файл я добавила также все файлы из подкаталогов (рис. fig. 4.1).

```
[alinagomazkova@10 ~]$ ls -lR /etc > file.txt
ls: невозможно открыть каталог '/etc/audit': Отка
```

Рис. 4.1: Запись в файл

Проверила, что в файл записались нужные значения с помощью утилиты head, она выводит первые 10 строк файла на экран (рис. fig. 4.2).

```
ls: невозможно открыть каталог '/etc/audit': Отказано в досту
[alinagomazkova@10 ~]$ head file.txt
/etc:
итого 1452
drwxr-xr-x. 1 root root    126 июн 30  2023 abrt
-rw-r--r--. 1 root root     16 сен  8  2023 adjtime
-rw-r--r--. 1 root root   1529 ноя 27  2022 aliases
drwxr-xr-x. 1 root root     70 сен  4  2023 alsa
drwxr-xr-x. 1 root root   1496 сен  8  2023 alternatives
drwxr-xr-x. 1 root root     56 апр 14  2023 anaconda
-rw-r--r--. 1 root root    269 янв 18  2023 anthy-unicode.conf
-rw-r--r--. 1 root root    833 фев 10  2023 appstream.conf
```

Рис. 4.2: Вывод содержимого файла

Добавила в созданный файл имена файлов из домашнего каталога, используя перенаправление “>” в режиме добавления (рис. fig. 4.3).



```
[alinagomazkova@10 ~]$ ls -lR ~/ >> file.txt
[alinagomazkova@10 ~]$
```

Рис. 4.3: Добавление данных в файл

Вывела на экран имена всех файлов, имеющих расширение “.conf” с помощью утилиты `grep` (рис. fig. 4.4).

```
[alinagomazkova@10 ~]$ grep .conf file.txt
-rw-r--r--. 1 root root      269 янв 18  2023 anthy-unicode.conf
-rw-r--r--. 1 root root      833 фев 10  2023 appstream.conf
-rw-r--r--. 1 root root       55 сен  4  2023 asound.conf
-rw-r--r--. 1 root root    29842 янв 24  2023 brltty.conf
drwxr-xr-x. 1 root root       0 авг  2  2023 chkconfig.d
-rw-r--r--. 1 root root     1372 авг  9  2023 chrony.conf
drwxr-xr-x. 1 root root      18 апр 14  2023 dconf
-rw-r--r--. 1 root root     1280 янв 19  2023 dleyna-renderer-service.conf
-rw-r--r--. 1 root root     1174 янв 19  2023 dleyna-server-service.conf
-rw-r--r--. 1 root root    28485 апр 27  2023 dnsmasq.conf
-rw-r--r--. 1 root root      117 апр 27  2023 dracut.conf
drwxr-xr-x. 1 root root       0 апр 27  2023 dracut.conf.d
-rw-r--r--. 1 root root       20 фев 24  2022 fprintd.conf
-rw-r--r--. 1 root root       38 апр  3  2023 fuse.conf
```

Рис. 4.4: Поиск файлов определенного расширения

Добавила вывод прошлой команды в новый файл `conf.txt` с помощью перенаправления “>” (файл создается при выполнении этой команды) (рис. fig. 4.5).

```
[alinagomazkova@10 ~]$ grep .conf file.txt > conf.txt
[alinagomazkova@10 ~]$ head conf.txt
-rw-r--r--. 1 root root      269 янв 18  2023 anthy-unicode.conf
-rw-r--r--. 1 root root      833 фев 10  2023 appstream.conf
-rw-r--r--. 1 root root       55 сен  4  2023 asound.conf
-rw-r--r--. 1 root root    29842 янв 24  2023 brltty.conf
drwxr-xr-x. 1 root root       0 авг  2  2023 chkconfig.d
-rw-r--r--. 1 root root     1372 авг  9  2023 chrony.conf
drwxr-xr-x. 1 root root      18 апр 14  2023 dconf
-rw-r--r--. 1 root root     1280 янв 19  2023 dleyna-renderer-service.conf
-rw-r--r--. 1 root root     1174 янв 19  2023 dleyna-server-service.conf
-rw-r--r--. 1 root root    28485 апр 27  2023 dnsmasq.conf
[alinagomazkova@10 ~]$
```

Рис. 4.5: Запись в файл

Определяю, какие файлы в домашнем каталоге начинаются с символа “с” с помощью утилиты `find`, прописываю ей в аргументах домашнюю директорию (тогда вывод относительно корневого каталога, а не домашнего будет), выбираю опцию `-name` (ищем по имени), и пишу маску, по которой будем искать имя, где \* - любое кол-во любых символов, добавляю опцию `-print`, чтобы мне вывелся

результат (рис. fig. 4.6). Но таким образом я получаю информацию даже о файлах из подкаталогов домашнего каталога.

```
[alinagomazkova@10 ~]$ find - -name "*" -print
/home/alinagomazkova/.mozilla/firefox/gl0coq29.default-release/compatibility.ini
/home/alinagomazkova/.mozilla/firefox/gl0coq29.default-release/cookies.sqlite
/home/alinagomazkova/.mozilla/firefox/gl0coq29.default-release/cert9.db
/home/alinagomazkova/.mozilla/firefox/gl0coq29.default-release/storage/permanent/chrome
/home/alinagomazkova/.mozilla/firefox/gl0coq29.default-release/storage/default/https+++www.youtube.com*partitionKey=%28https
%2Cgoogle.com%29/cache
/home/alinagomazkova/.mozilla/firefox/gl0coq29.default-release/storage/default/https+++www.youtube.com*partitionKey=%28https
%2Cgoogle.com%29/cache/caches.sqlite
/home/alinagomazkova/.mozilla/firefox/gl0coq29.default-release/storage/default/https+++www.zdnet.com/cache
/home/alinagomazkova/.mozilla/firefox/gl0coq29.default-release/storage/default/https+++www.zdnet.com/cache/caches.sqlite
/home/alinagomazkova/.mozilla/firefox/gl0coq29.default-release/storage/default/https+++mail.google.com/cache
/home/alinagomazkova/.mozilla/firefox/gl0coq29.default-release/storage/default/https+++mail.google.com/cache/caches.sqlite
/home/alinagomazkova/.mozilla/firefox/gl0coq29.default-release/storage/default/https+++frontend-stuff.com/cache
/home/alinagomazkova/.mozilla/firefox/gl0coq29.default-release/storage/default/https+++frontend-stuff.com/cache/caches.sqlit
e
/home/alinagomazkova/.mozilla/firefox/gl0coq29.default-release/storage/default/https+++www.youtube.com*partitionKey=%28https
%2Cfedoraproject.org%29/cache
```

Рис. 4.6: Поиск файлов, начинающихся с определенного элемента

Второй способ использовать утилиту `ls -lR` и использовать `grep`, чтобы найти элементы с первым символом `c`. Однако этот способ не работает для поиска файлов из подкаталогов каталога (рис. fig. 4.7).

```
/home/alinagomazkova/.mozilla/firefox/gl0coq29.default-release/storage/default/https+++www.youtube.com*partitionKey=%28https%2Cfedoraproject.org%29/cache
[alinagomazkova@10 ~]$ ls -lr | grep c*
-rw-r--r--. 1 alinagomazkova alinagomazkova 49076 map 22 23:56 conf.txt
[alinagomazkova@10 ~]$
```

Рис. 4.7: Поиск файлов, начинающихся с определенного элемента

С помощью метода `find`, чьи опции я расписала ранее, ищу все файлы, начинающиеся с буквы `h` (рис. fig. 4.8).

```

[alinagomazkova@10 ~]$ sudo find /etc -name "h*" -print
[sudo] пароль для alinagomazkova:
/etc/avahi/hosts
/etc/brltty/Contraction/ha.ctb
/etc/brltty/Input/bm/horizontal.kti
/etc/brltty/Input/hd
/etc/brltty/Input/hm
/etc/brltty/Input/ht
/etc/brltty/Input/hw
/etc/brltty/Text/he.ttb
/etc/brltty/Text/hi.ttb
/etc/brltty/Text/hr.ttb
/etc/brltty/Text/hu.ttb
/etc/brltty/Text/hy.ttb
/etc/containers/oci/hooks.d
/etc/firewalld/helpers
/etc/hp
/etc/hp/hplip.conf
/etc/httpd
/etc/httpd/conf/httpd.conf
/etc/liblverbs.d/hfiverbs.driver

```

Рис. 4.8: Поиск файлов, начинающихся с определенного элемента

Запускаю в фоновом режиме (на это указывает символ &) процесс, который будет записывать в файл logfile (с помощью перенаправления >) файлы, имена которых начинаются с log (рис. fig. 4.9).

```

/etc/nostname
[alinagomazkova@10 ~]$ find ~ -name "log*" -print > logfile &
[1] 4279
[alinagomazkova@10 ~]$

```

Рис. 4.9: Создание фонового процесса

Проверяю, что файл создан, удаляю его, проверяю, что файл удален (рис. fig. 4.10).

```
[alinagomazkova@10 ~]$ ls
abc1      install-tl-20231012  pandoc-3.1.11.1-linux-amd64.tar.gz  reports  Музыка
ASD       install-tl-unx.tar.gz  pandoc-3.1.12.2                      ski.plases  Общедоступные
australia LICENSE              pandoc-3.1.8-linux-amd64.tar.gz      test       'Рабочий стол'
bin       logfile              pandoc-crossref                       work       Шаблоны
conf.txt  may                 pandoc-crossref.1                     Видео
feathers  monthly            pandoc-crossref-Linux.tar.xz          Документы
file.txt  my_os              Pictures                              Загрузки
fun       pandoc-3.1.11.1     play                                 Изображения

[1]+  Завершён      find ~ -name "log*" -print > logfile
[alinagomazkova@10 ~]$ rm logfile
[alinagomazkova@10 ~]$ ls
abc1      fun                pandoc-3.1.11.1  Pictures  Документы
ASD       install-tl-20231012  pandoc-3.1.11.1-linux-amd64.tar.gz  play      Загрузки
australia install-tl-unx.tar.gz  pandoc-3.1.12.2  reports  Изображения
bin       LICENSE              pandoc-3.1.8-linux-amd64.tar.gz      ski.plases Музыка
conf.txt  may                 pandoc-crossref  test      Общедоступные
feathers  monthly            pandoc-crossref.1 work      'Рабочий стол'
file.txt  my_os              pandoc-crossref-Linux.tar.xz  Видео     Шаблоны
```

Рис. 4.10: Удаление файла

Запускаю в консоли в фоновом режиме (с помощью символа &) редактор mousepad, потому что редактора gedit у меня, к сожалению, но работают они идентично (рис. fig. 4.11).

```
[alinagomazkova@10 ~]$ mousepad &
[3] 4679
[2] Завершён      mousepad
```

Рис. 4.11: Создание фонового процесса

С помощью утилиты ps определяю идентификатор процесса mousepad, его значение 3913 (рис. fig. 4.12). Также мы можем определить идентификатор с помощью pgrep.

```
[alinagomazkova@10 ~]$ ps aux | grep mousepad
alinago+  4679  0.1  1.2 905636 56336 pts/0    Sl  19:43   0:01 mousepad
alinago+  4978  0.0  0.0 222564 2560 pts/0    S+   20:02   0:00 grep --color=auto mousepad
[alinagomazkova@10 ~]$ pgrep mousepad
4679
[alinagomazkova@10 ~]$ ps aux | grep mousepad | grep -v grep
alinago+  4679  0.1  1.2 905636 56336 pts/0    Sl  19:43   0:01 mousepad
```

Рис. 4.12: Поиск идентификатора процесса

Прочитала справку команды kill (рис. fig. 4.13).

```
KILL(1)                                User Commands                                KILL(1)

NAME
    kill - terminate a process

SYNOPSIS
    kill [-signal|-s signal|-p] [-q value] [-a] [--timeout milliseconds signal] [--] pid/name...

    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.

    If signal is 0, then no actual signal is sent, but error checking is still performed.
```

Рис. 4.13: Чтение документации

Использую команду **kill** и идентификатор процесса, чтобы его удалить (рис. fig. 4.14). Заметила, как у меня закрылась программа **mousepad**.

```
[alinagomazkova@10 ~]$ man kill
[alinagomazkova@10 ~]$ kill 4679
```

Рис. 4.14: Удаление процесса

Прочитала документацию про функции **df** и **du** (рис. fig. 4.15).

```
[alinagomazkova@10 ~]$ man df
[3]+  Завершён          mousepad
[alinagomazkova@10 ~]$ man du
[alinagomazkova@10 ~]$
```

Рис. 4.15: Чтение документации

Использую утилиту **df** опции **-iv** позволяют увидеть информацию об инодах и сделать вывод читаемым, игнорируя сообщения системы о нем (рис. fig. 4.16). Эта утилита нам нужна, чтобы выяснить, сколько свободного места есть у нашей системы.

```
[alinagomazkova@10 ~]$ df -vi
Файловая система      Инодов  Ииспользовано  ICсвободно  IIиспользовано%  Cмонтировано в
devtmpfs               543568        464      543104          1% /dev
tmpfs                  548495         1      548494          1% /dev/shm
tmpfs                  819200        866      818334          1% /run
/dev/sda3               0           0           0           - /
tmpfs                 1048576        39     1048537          1% /tmp
/dev/sda2              65536        394       65142          1% /boot
/dev/sda3               0           0           0           - /home
tmpfs                 109699        147     109552          1% /run/user/1000
/dev/sr0                0           0           0           - /run/media/alinagomazkova/VBox_GAs_7.0.14
```

Рис. 4.16: Утилита df

Использую утилиту du. Она нужна чтобы просмотреть, сколько места занимают файлы в определенной директории и найти самые большие из них (рис. fig. 4.17).

```
[alinagomazkova@10 ~]$ du -a work/study/2023-2024/Операционные\ системы/os-intro/labs/lab02/presentation/
48 work/study/2023-2024/Операционные системы/os-intro/labs/lab02/presentation/image/kulyabov.jpg
44 work/study/2023-2024/Операционные системы/os-intro/labs/lab02/presentation/image/1.png
16 work/study/2023-2024/Операционные системы/os-intro/labs/lab02/presentation/image/2.png
8 work/study/2023-2024/Операционные системы/os-intro/labs/lab02/presentation/image/3.png
8 work/study/2023-2024/Операционные системы/os-intro/labs/lab02/presentation/image/4.png
12 work/study/2023-2024/Операционные системы/os-intro/labs/lab02/presentation/image/5.png
56 work/study/2023-2024/Операционные системы/os-intro/labs/lab02/presentation/image/6.png
52 work/study/2023-2024/Операционные системы/os-intro/labs/lab02/presentation/image/7.png
16 work/study/2023-2024/Операционные системы/os-intro/labs/lab02/presentation/image/8.png
80 work/study/2023-2024/Операционные системы/os-intro/labs/lab02/presentation/image/9.png
60 work/study/2023-2024/Операционные системы/os-intro/labs/lab02/presentation/image/10.png
32 work/study/2023-2024/Операционные системы/os-intro/labs/lab02/presentation/image/11.png
8 work/study/2023-2024/Операционные системы/os-intro/labs/lab02/presentation/image/12.png
16 work/study/2023-2024/Операционные системы/os-intro/labs/lab02/presentation/image/13.png
80 work/study/2023-2024/Операционные системы/os-intro/labs/lab02/presentation/image/14.png
32 work/study/2023-2024/Операционные системы/os-intro/labs/lab02/presentation/image/15.png
16 work/study/2023-2024/Операционные системы/os-intro/labs/lab02/presentation/image/16.png
24 work/study/2023-2024/Операционные системы/os-intro/labs/lab02/presentation/image/17.png
20 work/study/2023-2024/Операционные системы/os-intro/labs/lab02/presentation/image/18.png
8 work/study/2023-2024/Операционные системы/os-intro/labs/lab02/presentation/image/19.png
116 work/study/2023-2024/Операционные системы/os-intro/labs/lab02/presentation/image/20.png
16 work/study/2023-2024/Операционные системы/os-intro/labs/lab02/presentation/image/21.png
12 work/study/2023-2024/Операционные системы/os-intro/labs/lab02/presentation/image/22.png
36 work/study/2023-2024/Операционные системы/os-intro/labs/lab02/presentation/image/23.png
36 work/study/2023-2024/Операционные системы/os-intro/labs/lab02/presentation/image/24.png
852 work/study/2023-2024/Операционные системы/os-intro/labs/lab02/presentation/image
4 work/study/2023-2024/Операционные системы/os-intro/labs/lab02/presentation/Makefile
12 work/study/2023-2024/Операционные системы/os-intro/labs/lab02/presentation/presentation.md
868 work/study/2023-2024/Операционные системы/os-intro/labs/lab02/presentation/
```

Рис. 4.17: Утилита du

Прочитала документацию о команде find (рис. fig. 4.18).

```
FIND(1)                                     General Commands Manual                 FIND(1)

NAME
  find - search for files in a directory hierarchy

SYNOPSIS
  find [-H] [-L] [-P] [-D debugopts] [-Olevel] [starting-point...] [expression]

DESCRIPTION
  This manual page documents the GNU version of find. GNU find searches the directory tree rooted at each given starting-point by evaluating the given expression from left to right, according to the rules of precedence (see section OPERATORS), until the outcome is known (the left hand side is false for and operations, true for or), at which point find moves on to the next file name. If no starting-point is specified, . is assumed.

  If you are using find in an environment where security is important (for example if you are using it to search directories that are writable by other users), you should read the 'Security Considerations' chapter of the findutils documentation, which is called Finding Files and comes with findutils. That document also includes a lot more detail and discussion than this manual page, so you may find it a more useful source of information.

OPTIONS
  The -H, -L and -P options control the treatment of symbolic links. Command-line arguments following these are taken to be names of files or directories to be examined, up to the first argument that begins with '-', or the argument '(' or '!'. That argument and any following arguments are taken to be the expression describing what is to be searched for. If no paths are given, the current directory is used. If no expression is given, the expression -print is used (but you should probably consider using -print0 instead, anyway).
```

Рис. 4.18: Чтение документации

Вывела имена всех директорий, имеющихся в моем домашнем каталоге, используя аргумент `d` у утилиты `find` опции `-type`, то есть указываю тип файлов, который мне нужен и этот тип Директория (рис. fig. 4.19). Утилита `-a` позволит увидеть размер всех файлов, а не только директорий.

```
[alinagomazkova@10 ~]$ find -type d
.
./mozilla
./mozilla/extensions
./mozilla/extensions/{ec8030f7-c20a-464f-9b0e-13a3a9e97384}
./mozilla/plugins
./mozilla/firefox
./mozilla/firefox/gl0coq29.default-release
./mozilla/firefox/gl0coq29.default-release/security_state
./mozilla/firefox/gl0coq29.default-release/storage
./mozilla/firefox/gl0coq29.default-release/storage/permanent
./mozilla/firefox/gl0coq29.default-release/storage/permanent/chrome
./mozilla/firefox/gl0coq29.default-release/storage/permanent/chrome/ldb
./mozilla/firefox/gl0coq29.default-release/storage/permanent/chrome/ldb/3870112724rsegmnoittet-es.files
./mozilla/firefox/gl0coq29.default-release/storage/permanent/chrome/ldb/3870112724rsegmnoittet-es.files/journals
./mozilla/firefox/gl0coq29.default-release/storage/permanent/chrome/ldb/356128884sdhlie.files
./mozilla/firefox/gl0coq29.default-release/storage/permanent/chrome/ldb/1451318868ntouromlalnodyr--epcr.files
./mozilla/firefox/gl0coq29.default-release/storage/permanent/chrome/ldb/1657114595Amcateirvtistt.files
./mozilla/firefox/gl0coq29.default-release/storage/permanent/chrome/ldb/2823318777ntouromlalnodyr--naod.files
./mozilla/firefox/gl0coq29.default-release/storage/permanent/chrome/ldb/2918063365piupsah.files
./mozilla/firefox/gl0coq29.default-release/storage/temporary
./mozilla/firefox/gl0coq29.default-release/storage/default
```

Рис. 4.19: Название рисунка

## 5 Выводы

По итогу лабораторной работы я ознакомилась с инструментами поиска файлов и фильтрации текстовых данных. Приобрела практические навыки: по управлению процессами (и заданиями), по проверке использования диска и обслуживанию файловых систем.



## 6 Ответы на контрольные вопросы

1. Какие потоки ввода вывода вы знаете? В системе по умолчанию открыто три специальных потока: – `stdin` — стандартный поток ввода (по умолчанию: клавиатура), файловый дескриптор 0; – `stdout` — стандартный поток вывода (по умолчанию: консоль), файловый дескриптор 1; – `stderr` — стандартный поток вывод сообщений об ошибках (по умолчанию: консоль), файловый дескриптор 2.
2. Объясните разницу между операцией `>` и `»`. Этот знак `>` - перенаправление ввода/вывода, а `»` - перенаправление в режиме добавления.
3. Что такое конвейер? Конвейер (`pipe`) служит для объединения простых команд или утилит в цепочки, в которых результат работы предыдущей команды передаётся последующей.
4. Что такое процесс? Чем это понятие отличается от программы? Главное отличие между программой и процессом заключается в том, что программа - это набор инструкций, который позволяет ЦПУ выполнять определенную задачу, в то время как процесс - это исполняемая программа.
5. Что такое PID и GID? PPID - (`parent process ID`) идентификатор родительского процесса. Процесс может порождать и другие процессы. UID, GID - реальные идентификаторы пользователя и его группы, запустившего данный процесс.
6. Что такое задачи и какая команда позволяет ими управлять? Запущенные

фоном программы называются задачами (jobs). Ими можно управлять с помощью команды jobs, которая выводит список запущенных в данный момент задач.

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

Команда htop похожа на команду top по выполняемой функции: они обе показывают информацию о процессах в реальном времени, выводят данные о потреблении системных ресурсов и позволяют искать, останавливать и управлять процессами.

У обеих команд есть свои преимущества. Например, в программе htop реализован очень удобный поиск по процессам, а также их фильтрация. В команде top это не так удобно — нужно знать кнопку для вывода функции поиска.

Зато в top можно разделять область окна и выводить информацию о процессах в соответствии с разными настройками. В целом top намного более гибкая в настройке отображения процессов.

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.

Утилита find предустановлена по умолчанию во всех Linux дистрибутивах, поэтому вам не нужно будет устанавливать никаких дополнительных пакетов. Это очень важная находка для тех, кто хочет использовать командную строку наиболее эффективно.

Команда find имеет такой синтаксис: find [папка] [параметры] критерий шаблон [действие] Пример: find /etc -name "p\*" -print

9. Можно ли по контексту (содержанию) найти файл? Если да, то как? `find / -type f -exec grep -H 'текстДляПоиска' {} ;`
10. Как определить объем свободной памяти на жёстком диске? С помощью команды `df -h`.
11. Как определить объем вашего домашнего каталога? С помощью команды `du -s`.
12. Как удалить зависший процесс? С помощью команды `kill%` номер задачи.