

# **Лабораторная работа №6**

**Дисциплина - Операционные системы**

Оширова Юлия Николаевна, НКАбд-02-22

# **Содержание**

<b>1 Цель работы</b>	<b>5</b>
<b>2 Задание</b>	<b>6</b>
<b>3 Теоретическое введение</b>	<b>8</b>
<b>4 Выполнение лабораторной работы</b>	<b>9</b>
<b>5 Выводы</b>	<b>16</b>
<b>6 Контрольные вопросы</b>	<b>17</b>
<b>Список литературы</b>	<b>20</b>

# Список иллюстраций

4.1	Запись названий файлов из каталога /etc в файл file.txt . . . . .	9
4.2	Запись названий файлов из домашнего каталога в файл file.txt . . . . .	9
4.3	Имена файлов с расширением .conf . . . . .	9
4.4	Запись этих имен в файл conf.txt . . . . .	10
4.5	Команда find . . . . .	10
4.6	Команда grep . . . . .	11
4.7	Имена файлов из каталога /etc, начинающиеся с символа h . . . . .	11
4.8	Запуск процесса в фоновом режиме . . . . .	11
4.9	Удалите файла ~logfile . . . . .	12
4.10	Запуск редактора в фоновом режиме . . . . .	12
4.11	Идентификатор процесса gedit . . . . .	12
4.12	Справка man kill . . . . .	13
4.13	Команда kill . . . . .	13
4.14	man df . . . . .	13
4.15	man du . . . . .	14
4.16	Команда df -vi . . . . .	14
4.17	Команда du -a . . . . .	14
4.18	Man find . . . . .	15
4.19	Команда find -type d . . . . .	15

# **Список таблиц**

# **1 Цель работы**

Ознакомиться с инструментами поиска файлов и фильтрации текстовых данных. Приобрести практические навыки: 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 текстовой файл conf.txt.
4. Определите, какие файлы в вашем домашнем каталоге имеют имена, начинавшиеся с символа c? 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, имеющихся в вашем домашнем каталоге.

## 3 Теоретическое введение

В интерфейсе командной строки есть очень полезная возможность перенаправления (переадресации) ввода и вывода (англ. термин I/O Redirection). Как мы уже заметили, многие программы выводят данные на экран. А ввод данных в терминале осуществляется с клавиатуры. С помощью специальных обозначений можно перенаправить вывод многих команд в файлы или иные устройства вывода (например, распечатать на принтере). Тоже самое и со вводом информации, вместо ввода данных с клавиатуры, для многих программ можно задать считывание символов из файла. Кроме того, можно даже вывод одной программы передать на ввод другой программе. [1]

К каждой программе, запускаемой в командной строке, по умолчанию подключено три потока данных:

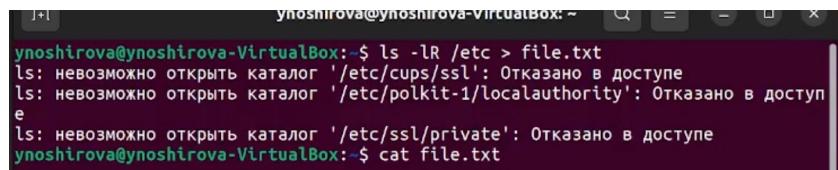
STDIN (0) — стандартный поток ввода (данные, загружаемые в программу).  
STDOUT (1) — стандартный поток вывода (данные, которые выводит программа).  
По умолчанию — терминал. STDERR (2) — стандартный поток вывода диагностических и отладочных сообщений (например, сообщениях об ошибках). По умолчанию — терминал. [2]

Pipe (конвеер) – это односторонний канал межпроцессного взаимодействия. Термин был придуман Дугласом Макилроем для командной оболочки Unix и назван по аналогии с трубопроводом. Конвейеры чаще всего используются в shell-скриптах для связи нескольких команд путем перенаправления вывода одной команды (stdout) на вход (stdin) последующей, используя символ конвеера '|'. [3]

## 4 Выполнение лабораторной работы

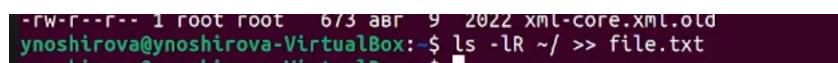
*Хочу сказать, что первое задание - это устно, нумерация заданий в отчете идет со второго задания.*

1. Запишем в файл file.txt названия файлов, содержащихся в каталоге /etc. Допишим в этот же файл названия файлов, содержащихся в вашем домашнем каталоге:



```
ynoshirova@ynoshirova-VirtualBox: ~$ ls -lR /etc > file.txt
ls: невозможно открыть каталог '/etc/cups/ssl': Отказано в доступе
ls: невозможно открыть каталог '/etc/polkit-1/localauthority': Отказано в доступе
ls: невозможно открыть каталог '/etc/ssl/private': Отказано в доступе
ynoshirova@ynoshirova-VirtualBox: ~$ cat file.txt
```

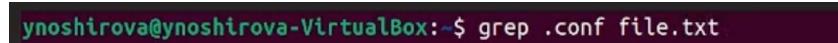
Рис. 4.1: Запись названий файлов из каталога /etc в файл file.txt



```
-r--r--r-- 1 root root 673 авг  9 2022 xml-core.xml.old
ynoshirova@ynoshirova-VirtualBox: ~$ ls -lR ~/ >> file.txt
ynoshirova@ynoshirova-VirtualBox: ~$ cat file.txt
```

Рис. 4.2: Запись названий файлов из домашнего каталога в файл file.txt

2. Выведите имена всех файлов из file.txt, имеющих расширение .conf, после чего запишите их в новый текстовой файл conf.txt:



```
ynoshirova@ynoshirova-VirtualBox: ~$ grep .conf file.txt
```

Рис. 4.3: Имена файлов с расширением .conf

```

/home/ynoshirova/work/study/2022-2023/Архитектура компьютера/arch-pc/config/course:
/home/ynoshirova/work/study/2022-2023/Архитектура компьютера/arch-pc/config/script:
drwxrwxr-x 4 ynoshirova ynoshirova 4096 фев 16 13:09 config
/home/ynoshirova/work/study/2022-2023/Операционные системы/os-intro/config:
/home/ynoshirova/work/study/2022-2023/Операционные системы/os-intro/config/course:
/home/ynoshirova/work/study/2022-2023/Операционные системы/os-intro/config/script:
ynoshirova@ynoshirova-VirtualBox:~$ grep '\.conf' file.txt > conf.txt
ynoshirova@ynoshirova-VirtualBox:~$ cat conf.txt
-rw-r--r-- 1 root root 3028 авг 9 2022 adduser.conf
-rw-r--r-- 1 root root 433 мар 23 2022 apg.conf
-rw-r--r-- 1 root root 769 фев 22 2022 appstream.conf
-rw-r--r-- 1 root root 29219 июн 28 2022 brltty.conf
-rw-r--r-- 1 root root 5532 фев 15 19:12 ca-certificates.conf
-rw-r--r-- 1 root root 5529 авг 9 2022 ca-certificates.conf.dpkg-old
-rw-r--r-- 1 root root 2969 фев 20 2022 debconf.conf
-rw-r--r-- 1 root root 604 сен 16 2018 deluser.conf
-rw-r--r-- 1 root root 685 янв 8 2022 e2scrub.conf
-rw-r--r-- 1 root root 20 фев 24 2022 fprintd.conf
-rw-r--r-- 1 root root 694 мар 23 2022 fuse.conf
-rw-r--r-- 1 root root 2584 фев 3 2022 gai.conf
-rw-r--r-- 1 root root 4436 дек 16 2020 hdparm.conf
-rw-r--r-- 1 root root 92 окт 15 2021 host.conf
drwxr-xr-x 2 root root 4096 авг 9 2022 insserv.conf.d
-rw-r--r-- 1 root root 110 ноя 5 22:17 kernel-img.conf
-rw-r--r-- 1 root root 1308 мар 24 2022 kerneloops.conf
-rw-r--r-- 1 root root 34 дек 16 2020 ld.so.conf
drwxr-xr-x 2 root root 4096 фев 15 13:33 ld.so.conf.d
-rw-r--r-- 1 root root 27 мар 13 2022 libao.conf
-rw-r--r-- 1 root root 191 мар 17 2022 libaudit.conf
-rw-r--r-- 1 root root 592 янв 24 2022 logrotate.conf
-rw-r--r-- 1 root root 5217 мар 17 2022 manpath.config
-rw-r--r-- 1 root root 744 янв 8 2022 mke2fs.conf
-rwxr-xr-x 1 root root 228 мар 23 2022 nftables.conf
-rw-r--r-- 1 root root 542 ноя 5 22:19 nsswitch.conf
-rw-r--r-- 1 root root 552 авг 12 2020 pam.conf

```

Рис. 4.4: Запись этих имен в файл conf.txt

3. Определите, какие файлы в вашем домашнем каталоге имеют имена, начинаяющиеся с символа с? Предложите несколько вариантов, как это сделать.

```

ynoshirova@ynoshirova-VirtualBox:~$ find ~ -name "c*" -print
/home/ynoshirova/.cache/thunderbird/hyhr4v5m.default-release/cache2
/home/ynoshirova/.cache/pip/http/c
/home/ynoshirova/.cache/pip/http/c/b/c
/home/ynoshirova/.cache/pip/http/c/b/c/9/4/cbc9492d2a4616aad83530ad74970874936540fe00b6ed22d96d3e:
/home/ynoshirova/.cache/pip/http/2/0/a/e/c
/home/ynoshirova/.cache/thumbna...

```

Рис. 4.5: Команда find

```
ynoshirova@ynoshirova-VirtualBox:~$ ls -l | grep c*
-rw-rw-r-- 1 ynoshirova ynoshirova 36404 мар 17 15:13 conf.txt
ynoshirova@ynoshirova-VirtualBox:~$
```

Рис. 4.6: Команда grep

4. Выведите на экран (по странично) имена файлов из каталога /etc, начинающиеся с символа h.

```
ynoshirova@ynoshirova-VirtualBox:~$ find /etc -name "h*" -print
find: '/etc/ssl/private': Отказано в доступе
/etc/hosts.deny
/etc/initramfs-tools/hooks
/etc/hosts.allow
/etc/hostname
/etc/udev/hwdb.d
/etc/X11/cursors/handhelds.theme
/etc/hdparm.conf
/etc/hosts
/etc/hp
/etc/hp/hplip.conf
/etc/hostid
find: '/etc/cups/ssl': Отказано в доступе
/etc/sane.d/hp5400.conf
/etc/sane.d/hs2p.conf
/etc/sane.d/hp4200.conf
/etc/sane.d/hp3900.conf
/etc/sane.d/hp.conf
/etc/sane.d/hpsj5s.conf
/etc/sane.d/dll.d/hplip
find: '/etc/polkit-1/localauthority': Отказано в доступе
/etc/ubuntu Advantage/help_data.yaml
/etc/brltty/Contraction/ha.ctb
/etc/brltty/Input/hd
/etc/brltty/Input/hw
```

Рис. 4.7: 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.

```
ynoshirova@ynoshirova-VirtualBox:~$ find ~ -name "log*" -print > logfile &
[1] 29451
```

Рис. 4.8: Запуск процесса в фоновом режиме

6. Удалите файл ~/logfile.

```
[1] 29451
ynoshirova@ynoshirova-VirtualBox:$ rm logfile
[1]+  Завершён          find ~ -name "log*" -print > logfile
```

Рис. 4.9: Удалите файла ~/logfile

7. Запустите из консоли в фоновом режиме редактор gedit.

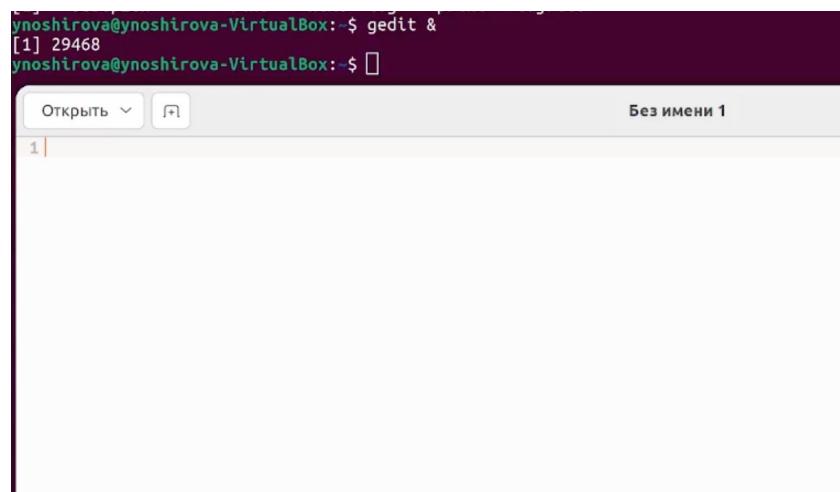


Рис. 4.10: Запуск редактора в фоновом режиме

8. Определите идентификатор процесса gedit, используя команду ps, конвейер и фильтр grep. Как ещё можно определить идентификатор процесса?

```
[1]+  завершён          gedit
ynoshirova@ynoshirova-VirtualBox:$ ps aux | grep gedit
ynoshir+ 29505 0.0 0.0 17756 2432 pts/0    S+   15:32   0:00 grep --color=auto gedit
ynoshirova@ynoshirova-VirtualBox:$
```

Рис. 4.11: Идентификатор процесса gedit

9. Прочтите справку (man) команды kill, после чего используйте её для завершения процесса gedit.

```

yoshirova@ynoshirova-VirtualBox: ~
User Commands

KILL(1)                                         User Commands
NAME   kill - send a signal to a process
SYNOPSIS kill [options] <pid> [...]
DESCRIPTION
The default signal for kill is TERM. Use -l or -L to list available signals. Particularly useful signals
INT, KILL, STOP, CONT, and 0. Alternate signals may be specified in three ways: -9, -SIGKILL or -KILL. N
values may be used to choose whole process groups; see the PGID column in ps command output. A PID of -1 is
indicates all processes except the kill process itself and init.
OPTIONS
<pid> [...]
    Send signal to every <pid> listed.

-<signal>
-s <signal>
--signal <signal>
    Specify the signal to be sent. The signal can be specified by using name or number. The behavior of
explained in signal(7) manual page.

-q, --queue value
    Use sigqueue(3) rather than kill(2) and the value argument is used to specify an in
nal. If the receiving process has installed a handler for this signal using the SA_S
then it can obtain this data via the si_value field of the siginfo_t structure.

-l, --list [signal]
    List signal names. This option has optional argument, which will convert signal n
way round.

```

Рис. 4.12: Справка man kill

```

yoshirova@ynoshirova-VirtualBox: ~$ man kill
yoshirova@ynoshirova-VirtualBox: ~$ kill 29505
bash: kill: (29505) - Нет такого процесса

```

Рис. 4.13: Команда kill

10. Выполните команды `df` и `du`, предварительно получив более подробную информацию об этих командах, с помощью команды `man`:

```

DF(1)                                         User Commands
NAME   df - report file system disk space usage
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 co
taining each file name argument. If no file name is given, the space available on all currently mounted file systems
shown. Disk space is shown in 1K blocks by default, unless the environment variable POSIXLY_CORRECT is set, in whi
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 spa
available on that file system rather than on the file system containing the device node. This version of df cannot sh
the space available on unmounted file systems, because on most kinds of systems doing so requires very nonportable int
imate knowledge of file system structures.

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

Mandatory arguments to long options are mandatory for short options too.

-a, --all
    include pseudo, duplicate, inaccessible file systems

-B, --block-size=SIZE
    scale sizes by SIZE before printing them; e.g., '-BM' prints sizes in units of 1,
    below

```

Рис. 4.14: man df

```

DU(1)                               User Commands
NAME
    du - estimate file space usage
SYNOPSIS
    du [OPTION]... [FILE]...
    du [OPTION]... -f[ile]0...-f[ile]n...-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 size is usually smaller, it may be
        to holes in ('sparse') files, internal fragmentation, indirect blocks, and the like
    -B, --block-size=SIZE
        scale sizes by SIZE before printing them; e.g., '-BM' prints sizes in units of 1,048
        below
    -b, --bytes
        equivalent to '--apparent-size --block-size=1'
    -c, --total

```



Рис. 4.15: man du

```

ynoshirova@ynoshirova-VirtualBox: ~ $ df -vi
Файл.система  ИNodeов  ИИспользовано  ИСвободно  ИИспользовано%  Смонтировано в
tmpfs          502238      977      501261      1% /run
/dev/sda3       5251072     484415     4766657     10% /
tmpfs          502238       1      502237      1% /dev/shm
tmpfs          502238       4      502234      1% /run/lock
/dev/sda2        0          0          0          - /boot/efi
tmpfs          100447      145      100302      1% /run/user/1000

```

Рис. 4.16: Команда df -vi

```

ynoshirova@ynoshirova-VirtualBox: ~ $ du -a /home/ynoshirova/work/study/2022-2023/Операционные системы/os-intro/labs/lab03
4   /home/ynoshirova/work/study/2022-2023/Операционные системы/os-intro/labs/lab03/presentation/Makefile
4   /home/ynoshirova/work/study/2022-2023/Операционные системы/os-intro/labs/lab03/presentation/presentation.md
736  /home/ynoshirova/work/study/2022-2023/Операционные системы/os-intro/labs/lab03/presentation/presentation.html
32   /home/ynoshirova/work/study/2022-2023/Операционные системы/os-intro/labs/lab03/presentation/presentation.pdf
48   /home/ynoshirova/work/study/2022-2023/Операционные системы/os-intro/labs/lab03/presentation/image/kulyabov.jpg
52   /home/ynoshirova/work/study/2022-2023/Операционные системы/os-intro/labs/lab03/presentation/image
832  /home/ynoshirova/work/study/2022-2023/Операционные системы/os-intro/labs/lab03/report
4   /home/ynoshirova/work/study/2022-2023/Операционные системы/os-intro/labs/lab03/report/Makefile
16   /home/ynoshirova/work/study/2022-2023/Операционные системы/os-intro/labs/lab03/report/pandoc/csl/gost-r-7-0-5-2008-numeric.
51.
29   /home/ynoshirova/work/study/2022-2023/Операционные системы/os-intro/labs/lab03/report/pandoc/csl
16   /home/ynoshirova/work/study/2022-2023/Операционные системы/os-intro/labs/lab03/report/pandoc/filters/pandoc_secnos.py
28   /home/ynoshirova/work/study/2022-2023/Операционные системы/os-intro/labs/lab03/report/pandoc/filters/pandoc_tablenos.py
4   /home/ynoshirova/work/study/2022-2023/Операционные системы/os-intro/labs/lab03/report/pandoc/filters/pandocxnos/_pycache_
_init__.cpython-310.pyc
28   /home/ynoshirova/work/study/2022-2023/Операционные системы/os-intro/labs/lab03/report/pandoc/core.cpython-310.pyc
core.cpython-310.pyc
4   /home/ynoshirova/work/study/2022-2023/Операционные системы/os-intro/labs/lab03/report/pandoc/main.cpython-310.pyc
8   /home/ynoshirova/work/study/2022-2023/Операционные системы/os-intro/labs/lab03/report/pandocattributes.cpython-310.pyc
48   /home/ynoshirova/work/study/2022-2023/Операционные системы/os-intro/labs/lab03/report/pandocfilters.cpython-310.pyc
4   /home/ynoshirova/work/study/2022-2023/Операционные системы/os-intro/labs/lab03/report/pandocsecnos.cpython-310.pyc

```



Рис. 4.17: Команда du -a

11. Воспользовавшись справкой команды find, выведите имена всех директорий, имеющихся в вашем домашнем каталоге:



```

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 point by evaluating the given expression from left to right, according to the rules of precedence (see section PRECEDENCE), until the outcome is known (the left hand side is false for and operations, true for or), at which point the search 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 for files 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 information 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 names of files or directories to be examined, up to the first argument that begins with '.', or the argument '!'. That argument and any following arguments are taken to be the expression describing what is to be searched. If no paths are given, the current directory is used. If no expression is given, the expression -print is used. You should probably consider using -print0 instead, anyway.

    This manual page talks about 'options' within the expression list. These options control the behaviour of find when specified immediately after the last path name. The five 'real' options -H, -L, -P, -D and -O must appear before the first path name, if at all. A double dash -- could theoretically be used to signal that a real option follows, but this does not really work due to the way find determines the end of the following expression. Instead, that by reading until an expression argument comes (which also starts with a '-'). Now, if there is a '-', then find would treat it as expression argument instead. Thus, to ensure that such options are treated as options, and especially to prevent that wildcard patterns expanded by the calling shell are treated as expression arguments, it is generally safer to prefix wildcards or dubious path names with either '-' or '--'.

Manual page find(1) line 1 (press h for help or q to quit)

```

Рис. 4.18: Man find

```

ynoshirova@ynoshirova-VirtualBox: $ find -type d
.
./australia
./play
./play/games
./play/games/play
./play/games/play/file.old
./play/games/file.old
./play/file.old
./.cache
./.cache/thunderbird
./.cache/thunderbird/5yb2ck8a.default
./.cache/thunderbird/hyhr4v5m.default-release
./.cache/thunderbird/hyhr4v5m.default-release/startupCache
./.cache/thunderbird/hyhr4v5m.default-release/cache2
./.cache/thunderbird/hyhr4v5m.default-release/cache2/entries
./.cache/thunderbird/hyhr4v5m.default-release/cache2/doomed
./.cache/thunderbird/hyhr4v5m.default-release/safebrowsing
./.cache/thunderbird/hyhr4v5m.default-release/safebrowsing/google4
./.cache/ibus
./.cache/ibus/bus
./.cache/pip
./.cache/pip/http
./.cache/pip/http/
./.cache/pip/http/c
./.cache/pip/http/c/b
./.cache/pip/http/c/b/c
./.cache/pip/http/c/b/c/9
./.cache/pip/http/c/b/c/9/4
./.cache/pip/http/7
./.cache/pip/http/7/4
./.cache/oin/http/7/4/b

```

Рис. 4.19: Команда find -type d

## **5 Выводы**

В процессе выполнения лабораторной работы 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 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.

# **Список литературы**

1. Перенаправление ввода и вывода
2. Конвейеры и перенаправление ввода-вывода в Linux
3. Linux pipes tips & tricks