

Отчёт по лабораторной работе № 6

Королёв Иван Андреевич

Содержание

| | | |
|----------|---|-----------|
| 1 | Цель работы | 5 |
| 2 | Задание | 6 |
| 3 | Теоретическое введение | 7 |
| 4 | Выполнение лабораторной работы | 8 |
| 4.1 | File.txt | 8 |
| 4.2 | File.txt in conf.txt | 9 |
| 4.3 | Файлы начинающиеся с символа “с” | 11 |
| 4.4 | Имена файлов начинающиеся с символа “h” | 11 |
| 4.5 | Запуск в фоновом режиме ~/logfile. Удаление | 12 |
| 4.6 | Gedit в фоновом режиме | 12 |
| 4.7 | Определение идентификатора процесса gedit | 13 |
| 4.8 | Man command kill | 13 |
| 4.9 | Выполнение команды df и du | 14 |
| 4.10 | Find -type d | 16 |
| 5 | Выводы | 18 |
| 6 | Ответы на контрольные вопросы | 19 |

Список иллюстраций

| | | |
|------|---|----|
| 4.1 | file.txt | 8 |
| 4.2 | file.txt | 8 |
| 4.3 | file.txt | 9 |
| 4.4 | file.txt | 10 |
| 4.5 | conf.txt | 10 |
| 4.6 | Файлы начинающиеся с символа “с” | 11 |
| 4.7 | Файлы начинающиеся с символа “с” | 11 |
| 4.8 | Файлы начинающиеся с символа “h” | 12 |
| 4.9 | Запуск в фоновом режиме ~/logfile. Удаление | 12 |
| 4.10 | Gedit в фоновом режиме | 13 |
| 4.11 | Определение идентификатора процесса gedit | 13 |
| 4.12 | Определение идентификатора процесса gedit | 13 |
| 4.13 | Man command kill | 14 |
| 4.14 | Man command kill | 14 |
| 4.15 | Выполнение команды df и du | 15 |
| 4.16 | Выполнение команды df и du | 15 |
| 4.17 | Выполнение команды df и du | 16 |
| 4.18 | Выполнение команды df и du | 16 |
| 4.19 | Find -type d | 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 команды `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 Теоретическое введение

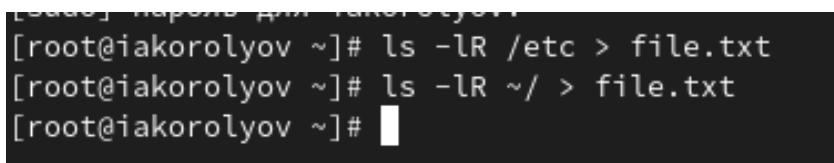
В системе по умолчанию открыто три специальных потока: – `stdin`—стандартный поток ввода (по умолчанию: клавиатура), файловый дескриптор 0; – `stdout` — стандартный поток вывода (по умолчанию: консоль), файловый дескриптор 1; – `stderr` — стандартный поток вывод сообщений об ошибках (по умолчанию: консоль), 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 `>`, `>>`, `<`, `<<`.

Конвейер (`pipe`) служит для объединения простых команд или утилит в цепочки, в которых результат работы предыдущей команды передаётся последующей. Синтаксис следующий: команда 1 | команда 2

4 Выполнение лабораторной работы

4.1 File.txt

1. Запишите в файл file.txt названия файлов, содержащихся в каталоге /etc. Допишите в этот же файл названия файлов, содержащихся в вашем домашнем каталоге. (рис. 4.1), (рис. 4.2), (рис. 4.3).



```
[root@iakorolyov ~]# ls -lR /etc > file.txt
[root@iakorolyov ~]# ls -lR ~/ > file.txt
[root@iakorolyov ~]#
```

Рис. 4.1: file.txt



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

Рис. 4.2: file.txt


```

[iakorolyov@iakorolyov ~]$ cat file.txt
/etc:
итого 1536
drwxr-xr-x. 1 root root      126 мая  5  2022 abrt
-rw-r--r--. 1 root root       16 окт 13 12:35 adjtime
-rw-r--r--. 1 root root    1529 июл 20  2022 aliases
drwxr-xr-x. 1 root root       70 окт 24 18:43 alsa
drwxr-xr-x. 1 root root    1554 фев 20 14:23 alternatives
drwxr-xr-x. 1 root root       56 мая 19  2022 anaconda
-rw-r--r--. 1 root root     269 мая  6  2022 anthy-unicode.conf
-rw-r--r--. 1 root root     833 авг 22  2022 appstream.conf
-rw-r--r--. 1 root root       55 окт 24 12:24 asound.conf
drwxr-xr-x. 1 root root     108 фев 12 22:46 audit
drwxr-xr-x. 1 root root     232 фев 12 23:01 authselect
drwxr-xr-x. 1 root root       66 ноя  7 20:30 avahi
drwxr-xr-x. 1 root root     144 фев 12 22:45 bash_completion.d
-rw-r--r--. 1 root root    2638 июл 20  2022 bashrc
-rw-r--r--. 1 root root     535 авг  8  2022 bindresvport.blackli
drwxr-xr-x. 1 root root        0 дек 20 21:56 binfmt.d
drwxr-xr-x. 1 root root       18 фев 12 22:18 bluetooth
-rw-r-----. 1 root brlapi    33 мая  5  2022 brlapi.key
drwxr-xr-x. 1 root root       76 июл  6  2022 brltty
-rw-r--r--. 1 root root   29842 июл  6  2022 brltty.conf
drwxr-xr-x. 1 root root        0 июл 23  2022 ceph

```

Рис. 4.3: file.txt

4.2 File.txt in conf.txt

2. Выведите имена всех файлов из file.txt, имеющих расширение .conf, после чего запишите их в новый текстовый файл conf.txt.(рис. 4.4), (рис. 4.5).

```

[iakorolyov@iakorolyov ~]$ grep .conf file.txt
-rw-r--r--. 1 root root      269 мая  6  2022 anthy-unicode.conf
-rw-r--r--. 1 root root      833 авг 22  2022 appstream.conf
-rw-r--r--. 1 root root       55 окт 24 12:24 asound.conf
-rw-r--r--. 1 root root    29842 июл  6  2022 brltty.conf
drwxr-xr-x. 1 root root       0 окт  5 12:38 chkconfig.d
-rw-r--r--. 1 root root     1371 авг 29  2022 chrony.conf
drwxr-xr-x. 1 root root      18 янв 20  2022 dconf
-rw-r--r--. 1 root root     1280 янв 20  2022 dleyna-renderer-service.conf
-rw-r--r--. 1 root root     1174 янв 20  2022 dleyna-server-service.conf
-rw-r--r--. 1 root root    27981 июл  8  2022 dnsmasq.conf
-rw-r--r--. 1 root root      117 сен 26 17:47 dracut.conf
drwxr-xr-x. 1 root root       0 сен 26 17:47 dracut.conf.d
lrwxrwxrwx. 1 root root      30 янв 27  2022 extlinux.conf -> ../boot/ext
extlinux.conf
-rw-r--r--. 1 root root      20 фев 24  2022 fprintd.conf
-rw-r--r--. 1 root root      38 янв 20  2022 fuse.conf
-rw-r--r--. 1 root root       9 июл 20  2022 host.conf
-rw-r--r--. 1 root root     5799 авг  9  2022 idmapd.conf
-rw-r--r--. 1 root root    78055 янв 20  2022 jwhois.conf
-rw-r--r--. 1 root root     8892 фев 12 22:24 kdump.conf
-rw-r--r--. 1 root root      880 ноя 16 20:16 krb5.conf

```

Рис. 4.4: file.txt

```

[iakorolyov@iakorolyov ~]$ grep '\.conf' file.txt > conf.txt
[iakorolyov@iakorolyov ~]$ cat conf.txt
-rw-r--r--. 1 root root      269 мая  6  2022 anthy-unicode.conf
-rw-r--r--. 1 root root      833 авг 22  2022 appstream.conf
-rw-r--r--. 1 root root       55 окт 24 12:24 asound.conf
-rw-r--r--. 1 root root    29842 июл  6  2022 brltty.conf
-rw-r--r--. 1 root root     1371 авг 29  2022 chrony.conf
-rw-r--r--. 1 root root     1280 янв 20  2022 dleyna-renderer-service.conf
-rw-r--r--. 1 root root     1174 янв 20  2022 dleyna-server-service.conf
-rw-r--r--. 1 root root    27981 июл  8  2022 dnsmasq.conf
-rw-r--r--. 1 root root      117 сен 26 17:47 dracut.conf
drwxr-xr-x. 1 root root       0 сен 26 17:47 dracut.conf.d
lrwxrwxrwx. 1 root root      30 янв 27  2022 extlinux.conf -> ../boot/ext
extlinux.conf
-rw-r--r--. 1 root root      20 фев 24  2022 fprintd.conf
-rw-r--r--. 1 root root      38 янв 20  2022 fuse.conf
-rw-r--r--. 1 root root       9 июл 20  2022 host.conf
-rw-r--r--. 1 root root     5799 авг  9  2022 idmapd.conf
-rw-r--r--. 1 root root    78055 янв 20  2022 jwhois.conf
-rw-r--r--. 1 root root     8892 фев 12 22:24 kdump.conf
-rw-r--r--. 1 root root      880 ноя 16 20:16 krb5.conf
drwxr-xr-x. 1 root root     136 фев 12 22:46 krb5.conf.d
-rw-r--r--. 1 root root       28 янв 11 11:46 ld.so.conf
drwxr-xr-x. 1 root root     152 фев 13 03:50 ld.so.conf.d

```

Рис. 4.5: conf.txt

4.3 Файлы начинающиеся с символа “с”

3. Определите, какие файлы в вашем домашнем каталоге имеют имена, начинавшиеся с символа с? Предложите несколько вариантов, как это сделать. (рис. 4.6), (рис. 4.7).

```
[iakorolyov@iakorolyov ~]$ find ~ -name "c*" -print
/home/iakorolyov/.mozilla/firefox/zleiph5u.default-release/crashes
/home/iakorolyov/.mozilla/firefox/zleiph5u.default-release/compatibility.i
/home/iakorolyov/.mozilla/firefox/zleiph5u.default-release/cookies.sqlite
/home/iakorolyov/.mozilla/firefox/zleiph5u.default-release/cert9.db
/home/iakorolyov/.mozilla/firefox/zleiph5u.default-release/storage/permane
ome
/home/iakorolyov/.mozilla/firefox/zleiph5u.default-release/containers.json
/home/iakorolyov/.mozilla/firefox/zleiph5u.default-release/content-prefs.s
/home/iakorolyov/.cache/mesa_shader_cache/0a/cb68d651efce4c90485a1b89a23ab
1dc
/home/iakorolyov/.cache/mesa_shader_cache/25/c91e1101b327104cb38c56d70c762
d0a
/home/iakorolyov/.cache/mesa_shader_cache/c7
/home/iakorolyov/.cache/mesa_shader_cache/bc/cd478565bb11fac4e88c431783503
c6d
/home/iakorolyov/.cache/mesa_shader_cache/40/cade9405a2ad22e1cd6d109312227
9da
/home/iakorolyov/.cache/mesa_shader_cache/44/c97a1bf68fe1467be380f121798e2
```

Рис. 4.6: Файлы начинающиеся с символа “с”

```
[iakorolyov@iakorolyov ~]$ ls -l | grep c*
-rw-r--r--. 1 iakorolyov iakorolyov 41303 map 12 15:46 conf.txt
[iakorolyov@iakorolyov ~]$
```

Рис. 4.7: Файлы начинающиеся с символа “с”

4.4 Имена файлов начинающиеся с символа “h”

4. Выведите на экран (по странично) имена файлов из каталога /etc, начинающиеся с символа h.(рис. 4.8).

```
[root@iakorolyov ~]# find /etc -name "h*" -print
/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/libibverbs.d/hfi1verbs.driver
/etc/libibverbs.d/hns.driver
/etc/logrotate.d/httpd
```

Рис. 4.8: Файлы начинающиеся с символа “h”

4.5 Запуск в фоновом режиме ~/logfile. Удаление

5. Запустите в фоновом режиме процесс, который будет записывать в файл ~/logfile файлы, имена которых начинаются с log. Удалите файл ~/logfile. (рис. 4.9).

```
/etc/mercurial/hgrc.d
[iakorolyov@iakorolyov ~]$ find ~ -name 'log' -print > ~/logfile &
[2] 3755
[iakorolyov@iakorolyov ~]$ rm logfile
[2]-  Завершён      find ~ -name 'log' -print > ~/logfile
[iakorolyov@iakorolyov ~]$
```

Рис. 4.9: Запуск в фоновом режиме ~/logfile. Удаление

4.6 Gedit в фоновом режиме

6. Запустите из консоли в фоновом режиме редактор gedit. (рис. 4.10).

```
[iakorolyov@iakorolyov ~]$ ps aux | grep gedit
iakorol+  3811  0.8  3.2 858812 77060 pts/2    Sl   16:04   0:03 gedit
iakorol+  3914  0.0  0.1 222192  2388 pts/2    S+   16:11   0:00 grep --color
=auto gedit
```

Рис. 4.10: Gedit в фоновом режиме

4.7 Определение идентификатора процесса gedit

7. Определите идентификатор процесса gedit, используя команду ps, конвейер и фильтр grep. (рис. 4.11), (рис. 4.12).

```
[iakorolyov@iakorolyov ~]$ ps aux | grep gedit | grep -v grep
iakorol+  3811  0.7  3.2 859240 77324 pts/2    Sl   16:04   0:03 gedit
```

Рис. 4.11: Определение идентификатора процесса gedit

```
[iakorolyov@iakorolyov ~]$ pgrep gedit
3811
```

Рис. 4.12: Определение идентификатора процесса gedit

4.8 Man command kill

8. Прочтите справку (man) команды kill, после чего используйте её для завершения процесса gedit. (рис. 4.13), (рис. 4.14).

```
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

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

Рис. 4.13: Man command kill

```
3811
[iakorolyov@iakorolyov ~]$ man kill

[3]+  Остановлен   man kill
[iakorolyov@iakorolyov ~]$ kill 3811
[2]   Завершено    gedit
[iakorolyov@iakorolyov ~]$
```

Рис. 4.14: Man command kill

4.9 Выполнение команды df и du

9. Выполните команды **df** и **du**, предварительно получив более подробную информацию об этих командах, с помощью команды **man**. (рис. 4.15), (рис. 4.16), (рис. 4.17), (рис. 4.18).

```

DF(1)                                User Commands                                DF(1)

NAME
    df - report file system space usage

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

DESCRIPTION
    This manual page documents the GNU version of df. df displays the
    amount of space available on the file system containing each file name
    argument. If no file name is given, the space available on all cur-
    rently mounted file systems is shown. 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 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 ver-
    sion of df cannot show the space available on unmounted file systems,
    because on most kinds of systems doing so requires very nonportable
    intimate knowledge of file system structures.

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

```

Рис. 4.15: Выполнение команды df и du

```

DU(1)                                User Commands                                DU(1)

NAME
    du - estimate file space usage

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

DESCRIPTION
    Summarize device usage of the set of FILES, recursively for directo-
    ries.

    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

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

```

Рис. 4.16: Выполнение команды df и du

```

[iakorolyov@iakorolyov ~]$ df -vi
Файловая система  Инодов  ИИспользовано  ИСвободно  ИИспользовано%  Смонтировано в
devtmpfs          1048576      459    1048117          1% /dev
tmpfs             298393        1    298392          1% /dev/shm
tmpfs            819200       914    818286          1% /run
/dev/sda2          0            0        0          - /
tmpfs            1048576       49    1048527          1% /tmp
/dev/sda1          65536       456    65080          1% /boot
/dev/sda2          0            0        0          - /home
doc               1000      -999000    1000000          - /media/sf_doc
tmpfs            59678       154    59524          1% /run/user/1000
/dev/sr0          0            0        0          - /run/media/iako
roryov/VBox_GAs_7.0.0
[iakorolyov@iakorolyov ~]$

```

Рис. 4.17: Выполнение команды df и du

```

[iakorolyov@iakorolyov ~]$ du -a ~/
4      /home/iakorolyov/.mozilla/extensions/{ec8030f7-c20a-464f-9b0e-13a3a9e7
384}/langpack-ru@firefox.mozilla.org.xpi
4      /home/iakorolyov/.mozilla/extensions/{ec8030f7-c20a-464f-9b0e-13a3a9e7
384}/.fedora-langpack-install
8      /home/iakorolyov/.mozilla/extensions/{ec8030f7-c20a-464f-9b0e-13a3a9e7
384}
8      /home/iakorolyov/.mozilla/extensions
0      /home/iakorolyov/.mozilla/plugins
0      /home/iakorolyov/.mozilla/firefox/Crash Reports/events
4      /home/iakorolyov/.mozilla/firefox/Crash Reports/InstallTime202205031033
50
4      /home/iakorolyov/.mozilla/firefox/Crash Reports
0      /home/iakorolyov/.mozilla/firefox/Pending Pings
4      /home/iakorolyov/.mozilla/firefox/zleiph5u.default-release/times.json
0      /home/iakorolyov/.mozilla/firefox/zleiph5u.default-release/.parentlock
0      /home/iakorolyov/.mozilla/firefox/zleiph5u.default-release/minidumps
0      /home/iakorolyov/.mozilla/firefox/zleiph5u.default-release/crashes/even
ts
4      /home/iakorolyov/.mozilla/firefox/zleiph5u.default-release/crashes/stor
e.json.mozlz4
4      /home/iakorolyov/.mozilla/firefox/zleiph5u.default-release/crashes
4      /home/iakorolyov/.mozilla/firefox/zleiph5u.default-release/compatibilit
v.ini

```

Рис. 4.18: Выполнение команды df и du

4.10 Find -type d

10. Воспользовавшись справкой команды find, выведите имена всех директо-
рий, имеющихсх в вашем домашнем каталоге.(рис. 4.19).


```
d[iakorolyov@iakorolyov ~]$ find -type d
0.
1./.mozilla
2./.mozilla/extensions
3./.mozilla/extensions/{ec8030f7-c20a-464f-9b0e-13a3a9e97384}
4./.mozilla/plugins
5./.mozilla/firefox
6./.mozilla/firefox/Crash Reports
7./.mozilla/firefox/Crash Reports/events
8./.mozilla/firefox/Pending Pings
9./.mozilla/firefox/zleiph5u.default-release
10./.mozilla/firefox/zleiph5u.default-release/minidumps
11./.mozilla/firefox/zleiph5u.default-release/crashes
12./.mozilla/firefox/zleiph5u.default-release/crashes/events
13./.mozilla/firefox/zleiph5u.default-release/security_state
14./.mozilla/firefox/zleiph5u.default-release/storage
15./.mozilla/firefox/zleiph5u.default-release/storage/permanent
16./.mozilla/firefox/zleiph5u.default-release/storage/permanent/chrome
17./.mozilla/firefox/zleiph5u.default-release/storage/permanent/chrome/idb
18./.mozilla/firefox/zleiph5u.default-release/storage/permanent/chrome/idb/3870112
19.724rsegmnoittet-es.files
20./.mozilla/firefox/zleiph5u.default-release/storage/permanent/chrome/idb/3870112
21.724rsegmnoittet-es.files/journals
```

Рис. 4.19: Find -type d

5 Выводы

Я ознакомился с инструментами поиска файлов и фильтрации текстовых данных. Приобрел практически навыки: по управлению процессами (и заданиями), по проверке использования диска и обслуживанию файловых систем.

6 Ответы на контрольные вопросы

1.
 - `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.
 - Перенаправление (какого-то) в файл (`file.txt`). `>` - Перенаправление (какого-то) в файл (`file.txt`). Файл открывается в режиме добавления.
3. Конвейер (`pipe`) служит для объединения простых команд или утилит в цепочки, в которых результат работы предыдущей команды передаётся последующей.
4. Между процессом и программой существует отношение многие-к-одному, что означает, что одна программа может вызывать несколько процессов или, другими словами, несколько процессов могут быть частью одной и той же программы. Сравнительная таблица. Основа для сравнения. программа. Процесс. основной. Программа представляет собой набор инструкций.
5. Если два процесса находятся в одном пространстве имен, тогда `ID-вне-ns` интерпретируется как `UID (GID)` в родительском `user namespace` идентификатора (`PID`) процесса.
6. Запущенные фоновыми программы называются задачами (`jobs`). Ими можно

управлять с помощью команды `jobs`, которая выводит список запущенных в данный момент задач.

7. 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. 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. `find / -type f -exec grep -H 'текстДляПоиска' {} ;`
10. С помощью команды `df -h`.
11. С помощью команды `du -s`.
12. С помощью команды `kill%` номер задачи.