

Лабораторная работа №8

**Поиск файлов. Перенаправление ввода-вывода. Просмотр
запущенных процессов**

Дашкина Анита

Содержание

1	Цель работы	5
2	Задание	6
3	Теоретическое введение	8
4	Выполнение лабораторной работы	10
5	Выводы	19
6	Контрольные вопросы	20
	Список литературы	21

Список иллюстраций

4.1	выполнение команды	10
4.2	вывод файлов	11
4.3	запись файлов в conf.txt	12
4.4	два варианта	13
4.5	выполнение команды	14
4.6	запуск gedit	14
4.7	выполнение команды	14
4.8	выполнение команды	14
4.9	выполнение команды	15
4.10	выполнение команды	15
4.11	man df	17
4.12	man du	17
4.13	выполнение команды df	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 директо-

рий, имею- щихся в вашем домашнем каталоге

3 Теоретическое введение

- Перенаправление ввода-вывода

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

Конвейер (`pipe`) служит для объединения простых команд или утилит в цепочки, в которых результат работы предыдущей команды передаётся последующей. Конвейеры можно группировать в цепочки и выводить с помощью перенаправления в файл

- Поиск файла

Команда `find` используется для поиска и отображения на экран имён файлов, соответствующих заданной строке символов.

- Фильтрация текста

Найти в текстовом файле указанную строку символов позволяет команда `grep`. Кроме того, команда `grep` способна обрабатывать стандартный вывод других команд (любой текст). Для этого следует использовать конвейер, связав вывод команды с вводом `grep`.

- Проверка использования диска

Команда `df` показывает размер каждого смонтированного раздела диска. Команда `du` показывает число килобайт, используемое каждым файлом или каталогом. Управление задачами Любую выполняющуюся в консоли команду или внешнюю программу можно запустить в фоновом режиме. Для этого следует в конце имени команды указать знак амперсанда `&`.

- Управление процессами

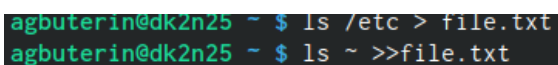
Любой команде, выполняемой в системе, присваивается идентификатор процесса (process ID). Получить информацию о процессе и управлять им, пользуясь идентификатором процесса, можно из любого окна командного интерпретатора

- Получение информации о процессах

Команда `ps` используется для получения информации о процессах

4 Выполнение лабораторной работы

1. Осуществите вход в систему, используя соответствующее имя пользователя.
2. Запишем в файл file.txt названия файлов, содержащихся в каталоге /etc. Допишите в этот же файл названия файлов, содержащихся в вашем домашнем каталоге

A screenshot of a terminal window showing two commands being executed. The first command is 'ls /etc > file.txt' and the second is 'ls ~ >>file.txt'. Both commands are preceded by the prompt 'agbuterin@dk2n25 ~ \$'.

```
agbuterin@dk2n25 ~ $ ls /etc > file.txt
agbuterin@dk2n25 ~ $ ls ~ >>file.txt
```

Рис. 4.1: выполнение команды

3. Выведем имена всех файлов из file.txt, имеющих расширение .conf, после чего запиши их в новый текстовый файл conf.txt.

```
agbuterin@dk2n25 ~ $ grep .conf file.txt
appstream.conf
brltty.conf
ca-certificates.conf
cachefilesd.conf
cfg-update.conf
chrome-flags.conf
chrony.conf
dconf
dhcpcd.conf
dispatch-conf.conf
dley-na-renderer-service.conf
dley-na-server-service.conf
dnsmasq.conf
dracut.conf
e2fsck.conf
e2scrub.conf
etc-update.conf
fluidsynth.conf
fuse.conf
gai.conf
gconf
genkernel.conf
gssapi_mech.conf
host.conf
i3blocks.conf
i3status.conf
idmapd.conf
idn2.conf
idn2.conf.sample
idnalias.conf
idnalias.conf.sample
ipsec.conf
java-config-2
krb5.conf
krb5.conf.example
ldap.conf
ldap.conf.sudo
ld.so.conf
ld.so.conf.d
libaudit.conf
lightdm.conf
locale.conf
logrotate.conf
```

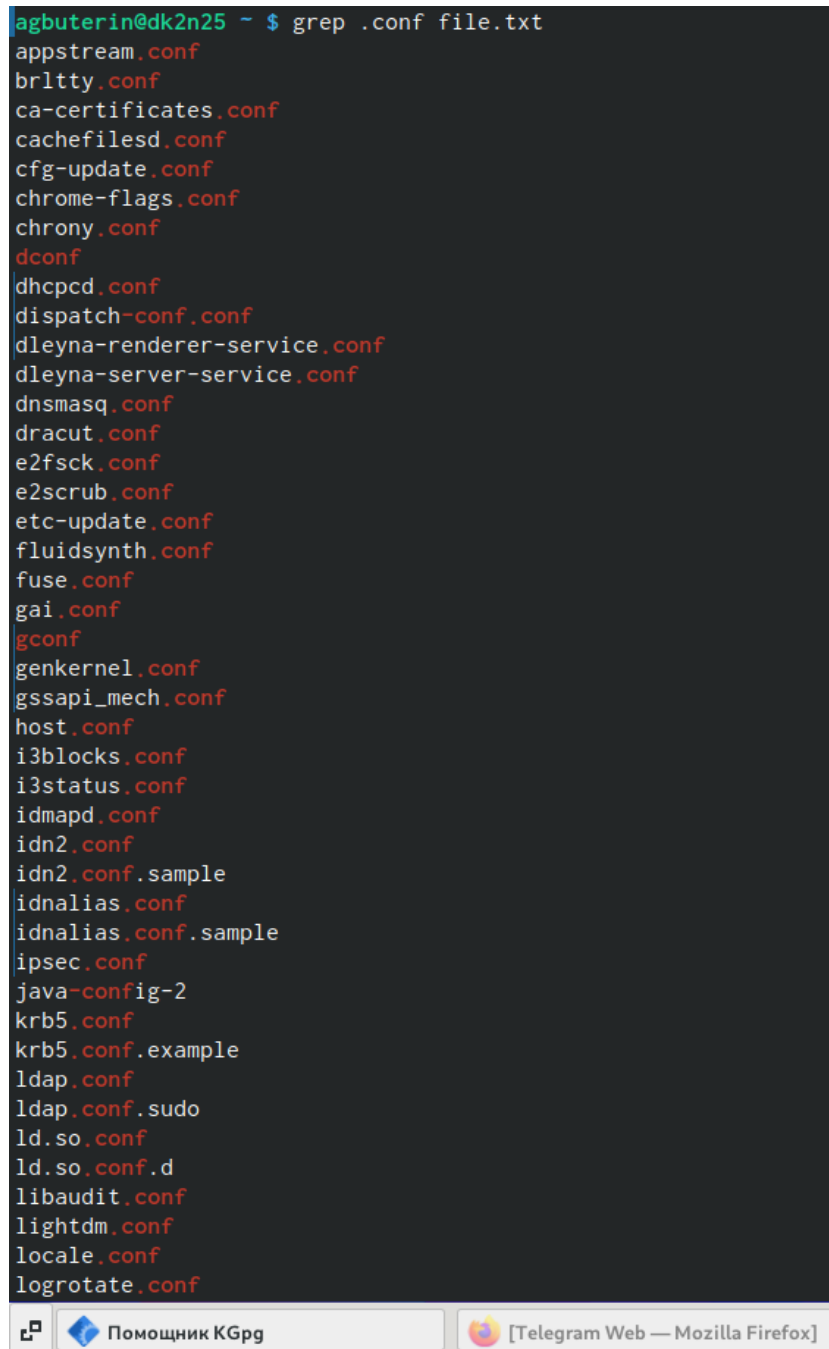


Рис. 4.2: вывод файлов

```
idnalias.conf
idnalias.conf.sample
ipsec.conf
java-config-2
krb5.conf
krb5.conf.example
ldap.conf
ldap.conf.sudo
ld.so.conf
ld.so.conf.d
libaudit.conf
lightdm.conf
locale.conf
logrotate.conf
mailutils.conf
make.conf
man.conf
man_db.conf
metalog.conf
mke2fs.conf
mlocate-cron.conf
modprobe.conf.1100
modprobe.conf.old
modules.conf
modules.conf.old
mplayer.conf
netconfig
nscd.conf
nslcd.conf
nss-ldapd.conf
nsswitch.conf
nsswitch.conf-
nsswitch.conf.20060126
nsswitch.conf.orig
nsswitch-sss.conf
ntp.conf
ntp.conf-eth0.sv
ntpd.conf
pear.conf
pe-format2.conf
pmount.conf
prelink.conf.d
pump.conf
pwdb.conf
```

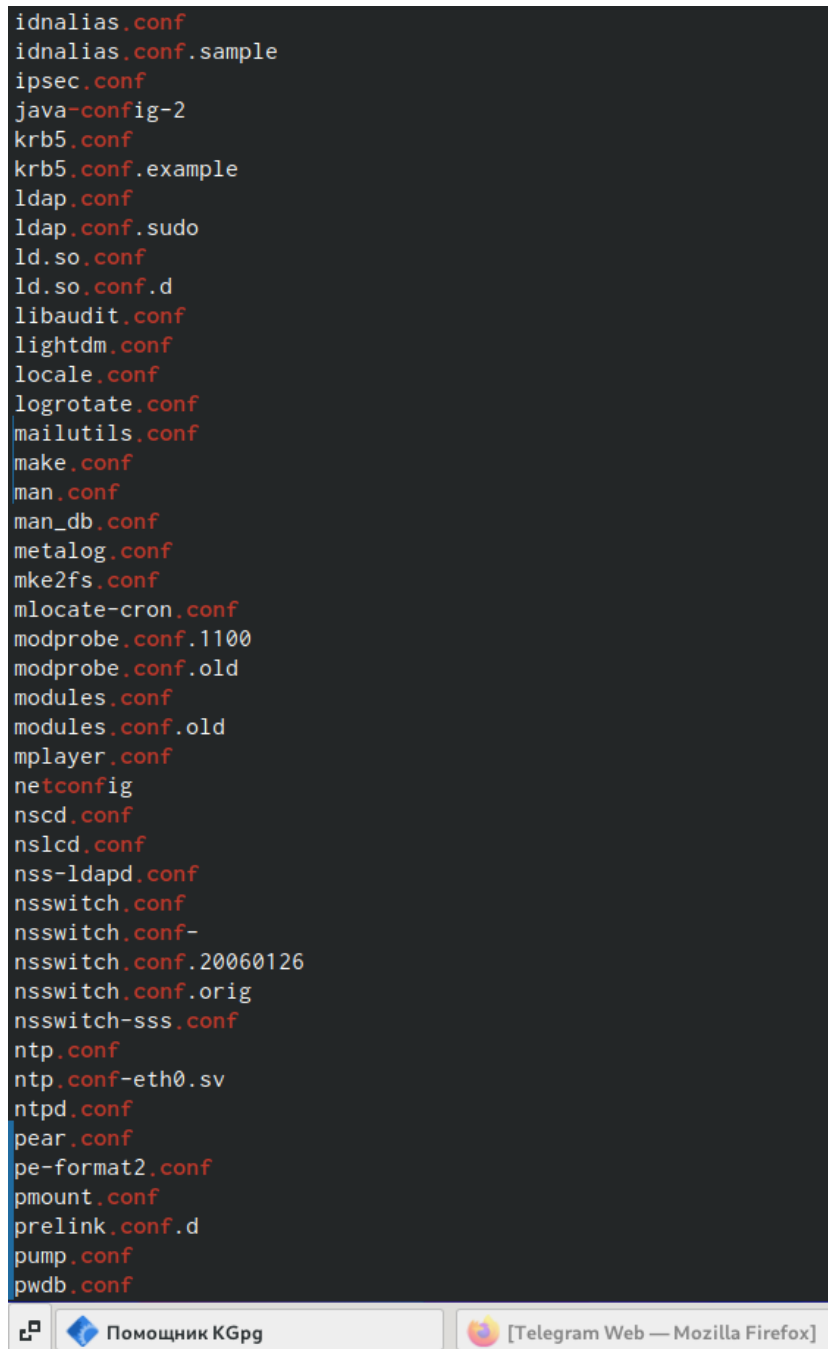


Рис. 4.3: запись файлов в conf.txt

4. Определим какие файлы в домашнем каталоге начинаются с символа с.

```
netconfig
nscd.conf
nslcd.conf
nss-ldapd.conf
nsswitch.conf
nsswitch.conf-
nsswitch.conf.20060126
nsswitch.conf.orig
nsswitch-sss.conf
ntp.conf
ntp.conf-eth0.sv
ntpd.conf
pear.conf
pe-format2.conf
pmount.conf
prelink.conf.d
pump.conf
pwdb.conf
rc.conf
request-key.conf
resolv.conf
rofi-pass.conf
rsyncd.conf
rsyslog.conf
sandbox.conf
sddm.conf
sddm.conf.d
sensors3.conf
signond.conf
smartd.conf
strongswan.conf
sudo.conf
sudo_logsrvd.conf
swtpm-localca.conf
swtpm_setup.conf
sysconfig
systemconfig
udhcpd.conf
updatedb.conf
vconsole.conf
whois.conf
xattr.conf
xinetd.conf
agbuterin@edk2n25 ~ $
```

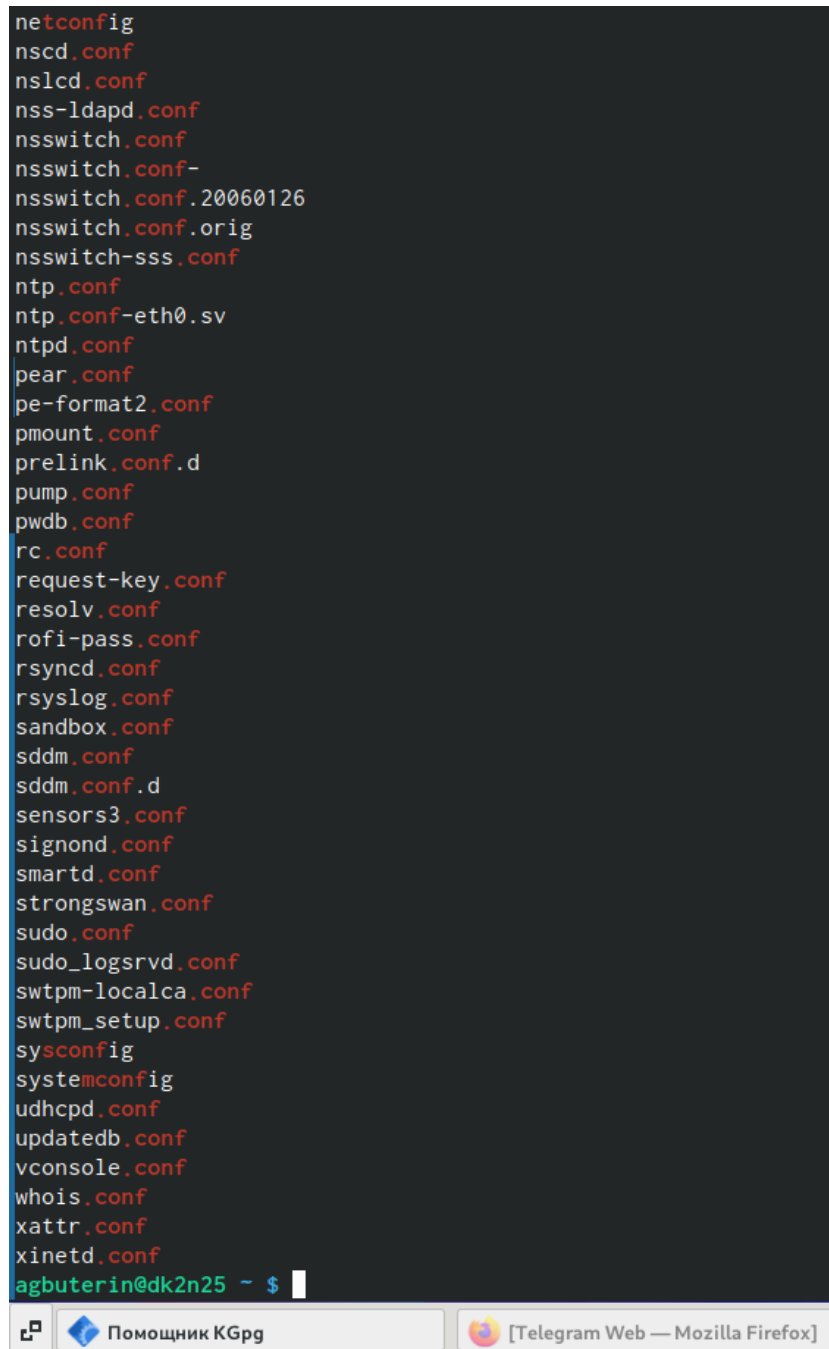


Рис. 4.4: два варианта

5. Выведем на экран имена файлов из каталога /etc, начинающиеся с символа h

```

agbuterin@dk2n25 ~ $ grep .conf file.txt > conf.txt
agbuterin@dk2n25 ~ $ ls -l | grep c*
-rw-r--r-- 1 agbuterin studsci 1352 map 29 16:08 conf.txt
agbuterin@dk2n25 ~ $ find ~/c* -name "c*" -print
/afs/.dk.sci.pfu.edu.ru/home/a/g/agbuterin/conf.txt
agbuterin@dk2n25 ~ $

```

6. Запустим в фоновом режиме процесс, который будет записывать в файл ~/logfile файлы, имена которых начинаются с log, удалим logfile.

```

find ~ -name "log*" -print > ~/logfile &
[2] 17189
[1] Выход 1
agbuterin@dk2n25 ~ $ rm -r logfile
[2]+ Завершён
agbuterin@dk2n25 ~ $

```

Рис. 4.5: выполнение команды

7. Запустим из консоли в фоновом режиме gedit.

```

agbuterin@dk2n25 ~ $ gedit &
[1] 17303
agbuterin@dk2n25 ~ $

```

Рис. 4.6: запуск gedit

8. Определим идентификатор процесса с помощью команды ps

```

agbuterin@dk2n25 ~ $ ps aux | grep -l gedit
(стандартный ввод)
[1]+ Завершён
agbuterin@dk2n25 ~ $

```

Рис. 4.7: выполнение команды

9. Определим идентификатор процесса с помощью команды ps

```

agbuterin@dk2n25 ~ $ ps aux | grep -l gedit
agbuter+ 17545  0.0  0.0  6932 2332 pts/2    S+   16:14   0:00 grep --colour=auto -i gedit
agbuterin@dk2n25 ~ $

```

Рис. 4.8: выполнение команды

10. Изучим команду kill и с помощью неё прекратим gedit

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

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 include HUP, INT, KILL, STOP, CONT, and 0. Alternate signals may be specified
    in three ways: -9, -SIGKILL or -KILL. Negative PID values may be used to choose whole process
    groups; see the PGID column in ps command output. A PID of -1 is special; it 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 signals is 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 integer
        to be sent with the signal. If the receiving process has installed a handler for
        this signal using the SA_SIGINFO flag to sigaction(2), 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 number
        to signal name, or other way round.

    -L, --table
        List signal names in a nice table.

NOTES
    Your shell (command line interpreter) may have a built-in kill command. You may need
    to run the command described here as /bin/kill to solve the conflict.

EXAMPLES
    kill -9 -1
        Kill all processes you can kill.

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

Рис. 4.9: выполнение команды

```
agbuterin@dk2n25 ~ $ kill 37641
bash: kill: (37641) - Нет такого процесса
agbuterin@dk2n25 ~ $
```

Рис. 4.10: выполнение команды

11. Изучим и выполним команды df и du

```
df(1)                                User Commands

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 currently 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 version of df cannot show the space available on unmounted file systems because on most kinds of systems doing so requires non-portable intimate knowledge of the 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 1,048,576 bytes; see SIZE format below

-h, --human-readable
    print sizes in powers of 1024 (e.g., 1023M)

-H, --si
    print sizes in powers of 1000 (e.g., 1.1G)

-i, --inodes
    list inode information instead of block usage

-k      like --block-size=1K

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



```
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 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 device usage; although the apparent size is usually
        smaller, it may be larger due to holes in ('sparse') files, internal fragmentation, in-
        direct blocks, and the like

    -B, --block-size=SIZE
        scale sizes by SIZE before printing them; e.g., '-BM' prints sizes in units of
        1,048,576 bytes; see SIZE format below

    -b, --bytes
        equivalent to '--apparent-size --block-size=1'

    -c, --total
        produce a grand total

    -D, --dereference-args
        dereference only symlinks that are listed on the command line

    -d, --max-depth=N
        print the total for a directory (or file, with --all) only if it is N or fewer levels
        below the command line argument; --max-depth=0 is the same as --summarize

    --files0-from=F
        summarize device usage of the NUL-terminated file names specified in file F; if F is -,
        then read names from standard input
Manual page du(1) line 1 (press h for help or q to quit)
```

Рис. 4.11: man df

```
agbuterin@dk2n25 ~ $ man df
agbuterin@dk2n25 ~ $ man du
agbuterin@dk2n25 ~ $
```

Рис. 4.12: man du

```
agbuterin@dk2n25 ~ $ df
Файловая система 1K-блоков  Использовано  Доступно  Использовано%  Смонтировано в
/dev/sda8          525626596    125880888    372971916        26% /
devtmpfs           4096         0            4096            0% /dev
tmpfs              3999624      0            3999624          0% /dev/shm
tmpfs              1599852      20244        1579608          2% /run
tmpfs              3999628      243956       3755672          7% /tmp
AFS                2147483647   0            2147483647       0% /afs
tmpfs              799924       208          799716           1% /run/user/5429
agbuterin@dk2n25 ~ $
```

Рис. 4.13: выполнение команды df

```

49      ./blog/resources/_gen/images/project/exte
880    ./blog/resources/_gen/images/project
49      ./blog/resources/_gen/images/publication
126    ./blog/resources/_gen/images/publication
63      ./blog/resources/_gen/images/publication
240    ./blog/resources/_gen/images/publication
90      ./blog/resources/_gen/images/event/exampl
92      ./blog/resources/_gen/images/event
2524   ./blog/resources/_gen/images
3180   ./blog/resources/_gen
3182   ./blog/resources
34      ./blog/AGButerin.github.io/.git/hooks
3      ./blog/AGButerin.github.io/.git/info
3      ./blog/AGButerin.github.io/.git/refs/heads
2      ./blog/AGButerin.github.io/.git/refs/tags
3      ./blog/AGButerin.github.io/.git/refs/remotes
5      ./blog/AGButerin.github.io/.git/refs/remotes
12     ./blog/AGButerin.github.io/.git/refs
6      ./blog/AGButerin.github.io/.git/objects/pack
2      ./blog/AGButerin.github.io/.git/objects/pack
10     ./blog/AGButerin.github.io/.git/objects/pack
3      ./blog/AGButerin.github.io/.git/logs/refs/heads
5      ./blog/AGButerin.github.io/.git/logs/refs/heads
3      ./blog/AGButerin.github.io/.git/logs/refs/tags
10     ./blog/AGButerin.github.io/.git/logs/refs/tags
13     ./blog/AGButerin.github.io/.git/logs
79     ./blog/AGButerin.github.io/.git
82     ./blog/AGButerin.github.io
35496  ./blog
59     ./pki/nssdb
61     ./pki
2      ./letters
2      ./nenos
2      ./nisk
192    ./Снимки экрана
2      ./feathers
2      ./ski.plases/plans
7      ./ski.plases/equipment
11     ./ski.plases
2      ./play/games
9      ./play
7      ./equiplist2
2      ./australia

```

12. С помощью команды `find` выведем имена всех директорий

5 Выводы

Мы ознакомились с инструментами поиска файлов и фильтрации текстовых данных. Приобрели практические навыки: по управлению процессами (и заданиями), по проверке использования диска и обслуживанию файловых систем.

6 Контрольные вопросы

Список литературы