

Лабораторная работа-06

Инструкция поиска файлов и фильтрация текстовых данных

Световидова Полина НБИбд-04-22

Содержание

1	Цель работы	5
2	Задание	6
3	Выполнение лабораторной работы	8
4	Выводы	20
5	Контрольные вопросы	21
6	Ответы на вопросы	22

Список иллюстраций

3.1	Записал в файл названия файлов из каталога /etc	8
3.2	Расширение .conf	9
3.3	conf.txt	10
3.4	Файлы с буквы c	11
3.5	Файлы с буквы h	11
3.6	Весь каталог	12
3.7	logfile	12
3.8	Запустил редактор	13
3.9	gedit	13
3.10	ps, gedit	14
3.11	команда kill	14
3.12	kill	15
3.13	команды df and du	16
3.14	команды df and du	17
3.15	df and du	17
3.16	df and du	17
3.17	команда find	18
3.18	команда find	19

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,имеющихся в вашем домашнем каталоге.

3 Выполнение лабораторной работы

Вошла в систему, используя свой пароль. Записываю в файл file.txt названия файлов, содержащихся в каталоге /etc. Дописываю в этот же файл названия файлов, содержащихся в домашнем каталоге. (рис. [3.1])



Рис. 3.1: Записал в файл названия файлов из каталога /etc

Вывожу имена всех файлов из file.txt, имеющих расширение .conf, после чего записала их в новый текстовый файл conf.txt (рис. [3.2]) (рис. [3.3])


```
pmsvetovidova@dk8n73 ~ $ ls /etc > file.txt
pmsvetovidova@dk8n73 ~ $ ls ~ >>file.txt
pmsvetovidova@dk8n73 ~ $ grep .conf file.txt
appstream.conf
brltty.conf
ca-certificates.conf
cachefilesd.conf
cfg-update.conf
dconf
dhcpcd.conf
dispatch-conf.conf
dleyna-server-service.conf
dnsmasq.conf
e2fsck.conf
e2scrub.conf
etc-update.conf
fluidsynth.conf
fuse.conf
gai.conf
gconf
genkernel.conf
gssapi_mech.conf
host.conf
idmapd.conf
idn2.conf
idn2.conf.sample
idnalias.conf
idnalias.conf.sample
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
```

Рис. 3.2: Расширение .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
rc.conf
request-key.conf
resolv.conf
rofi-pass.conf
rsyncd.conf
rsyslog.conf
sandbox.conf
sddm.conf
```

Рис. 3.3: conf.txt

Определил, какие файлы в домашнем каталоге имеют имена, которые начинаются с символа с.Привела несколько примеров, как это сделать.(рис. [3.4])

```

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
sensors3.conf
signond.conf
smartd.conf
sudo.conf
sudo_logsrvd.conf
sysconfig
systemconfig
udhcpd.conf
updatedb.conf
vconsole.conf
whois.conf
xattr.conf
xinetd.conf
xtables.conf
pmsvetovidova@dk8n73 ~ $

```

Рис. 3.4: Файлы с буквы c

Вывела на экран имена файлов из каталога /etc, начинающиеся с символа h.(рис. [3.5]) (рис. [3.6])

```

pmsvetovidova@dk8n73 ~ $ grep .conf file.txt > conf.txt
pmsvetovidova@dk8n73 ~ $ ls -l | grep c*
-rw-r--r-- 1 pmsvetovidova studsci 1191 мар 17 19:38 conf.txt
pmsvetovidova@dk8n73 ~ $ find ~/c* -name "c*" -print
/afs/.dk.sci.pfu.edu.ru/home/p/m/pmsvetovidova/conf.txt
pmsvetovidova@dk8n73 ~ $ ls -l /etc/passwd

```

Рис. 3.5: Файлы с буквы h

```

xinetd.conf
xtables.conf
pmsvetovidova@dk8n73 - $ grep .conf file.txt > conf.txt
pmsvetovidova@dk8n73 - $ ls -l | grep c*
-rw-r--r-- 1 pmsvetovidova studsci 1191 map 17 19:38 conf.txt
pmsvetovidova@dk8n73 - $ find ~/c* -name "c*" -print
/afs/.dk.sci.pfu.edu.ru/home/p/m/pmsvetovidova/conf.txt
pmsvetovidova@dk8n73 - $ ls -l | grep h* | less

```

Рис. 3.6: Весь каталог

Запустила в фоновом режиме процесс, который будет записывать в файл ~/logfile файлы, имена которых начинаются с log. (рис. [3.7]) Удалила файл ~/logfile.

```

итого 47
-rw-rw-r-- 1 pmsvetovidova studsci 0 map 3 16:23 abc1
drwxr--r-- 2 pmsvetovidova studsci 2048 map 3 16:37 australia
drwxr-xr-x 2 pmsvetovidova studsci 2048 map 17 11:21 bin
-rw-r--r-- 1 pmsvetovidova studsci 1191 map 17 19:38 conf.txt
-rw-r--r-- 1 pmsvetovidova studsci 0 map 3 16:41 feathers
-rw-r--r-- 1 pmsvetovidova studsci 4035 map 17 19:36 file.txt
drwxr-xr-x 3 pmsvetovidova studsci 2048 map 2 17:54 GNUstep
-rw-r--r-- 1 pmsvetovidova studsci 0 окт 6 15:13 lab1.cpp
-rwxr--r-- 1 pmsvetovidova studsci 0 map 3 16:21 may
drwxr-xr-x 2 pmsvetovidova studsci 2048 map 1 13:04 mkdir
drwxr-xr-x 2 pmsvetovidova studsci 2048 map 3 16:18 monthly
dr-xr--r-x 2 pmsvetovidova studsci 2048 map 3 16:40 my_os
drw-r-xr-x 3 pmsvetovidova studsci 2048 map 3 16:44 play
drwxr-xr-x 3 pmsvetovidova root 2048 сен 2 2022 public
lrwxr-xr-x 1 pmsvetovidova root 18 map 3 00:18 public_html -> public/public_html
drwxr-xr-x 3 pmsvetovidova studsci 2048 map 3 16:20 reports
drwxr-xr-x 4 pmsvetovidova studsci 2048 map 3 16:33 ski.places
drwxr-xr-x 2 pmsvetovidova studsci 2048 map 2 17:54 tmp
drwxr-xr-x 5 pmsvetovidova studsci 2048 map 17 11:30 work
drwxr-xr-x 2 pmsvetovidova studsci 2048 сен 7 2022 Видео
drwxr-xr-x 2 pmsvetovidova studsci 2048 сен 7 2022 Документы
drwxr-xr-x 2 pmsvetovidova studsci 2048 map 17 18:37 Загрузки
drwxr-xr-x 3 pmsvetovidova studsci 2048 map 17 18:38 Изображения
drwxr-xr-x 2 pmsvetovidova studsci 2048 сен 7 2022 Музыка
drwxr-xr-x 2 pmsvetovidova studsci 2048 сен 7 2022 Общедоступные
drwxr-xr-x 2 pmsvetovidova studsci 2048 сен 7 2022 Рабочий стол
drwxr-xr-x 2 pmsvetovidova studsci 2048 сен 7 2022 Шаблоны
lines 1-28/28 (END)

```

Рис. 3.7: logfile

Запустила из консоли в фоновом режиме редактор gedit.(рис. [3.8]) (рис. [3.9])

```
pmsvetovidova@dk8n73 ~ $ find ~ -name "log*" -print > ~/logfile &
[1] 92170
pmsvetovidova@dk8n73 ~ $ rm -r logfile
[1]+  Завершён      find ~ -name "log*" -print > ~/logfile
pmsvetovidova@dk8n73 ~ $
```

Рис. 3.8: Запустил редактор

```
pmsvetovidova@dk8n73 ~ $ find ~ -name "log*" -print > ~/logfile &
[1] 92170
pmsvetovidova@dk8n73 ~ $ rm -r logfile
[1]+  Завершён      find ~ -name "log*" -print > ~/logfile
pmsvetovidova@dk8n73 ~ $ gedit &
[1] 92264
pmsvetovidova@dk8n73 ~ $
```

Рис. 3.9: gedit

Определила идентификатор процесса gedit Используя команду ps, конвейер

и фильтр grep. Более простым способом определить этот идентификатор не получилось.(рис. [3.10])

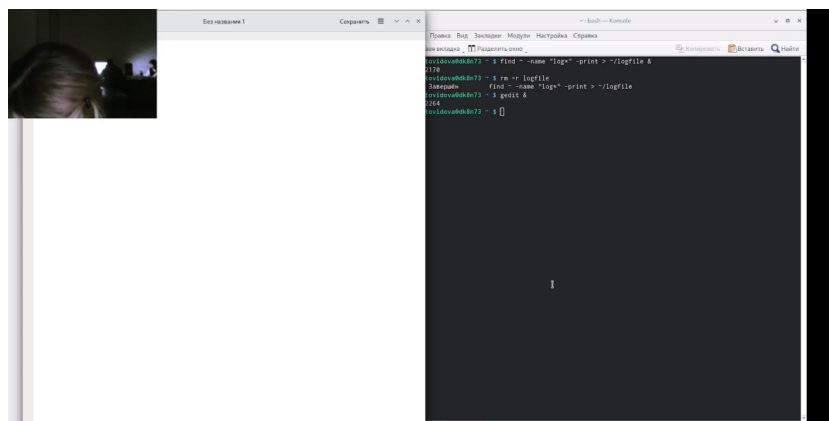


Рис. 3.10: ps, gedit

Прочла справку (man) команды kill, после чего использовала её для завершения процесса gedit.(рис. [3.11]) (рис. [3.12])

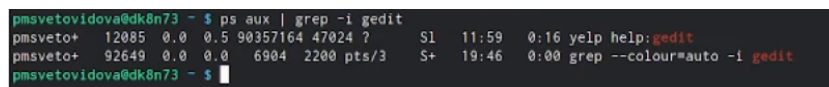


Рис. 3.11: команда kill

```
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
```

Рис. 3.12: kill

Выполнила команды df и du Предварительно получив более подробную информацию об этих командах, с помощью команды man.(рис. [3.13]) (рис. [3.14]) (рис. [3.15]) (рис. [3.16])

```

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 currently mounted file systems is shown. Space is shown in 1K blocks by de-
    fault, 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 very nonportable intimate knowledge of file
    system structures.

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

    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

```

Рис. 3.13: команды df and 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 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

```

Рис. 3.14: команды df and du

```

pmsvetovidova@dk8n73 ~ $ df
Файловая система 1K-блоков  Использовано    Доступно  Использовано%  Смонтировано в
none              3999704      65984      3933720        2% /run
udev              10240       0          10240         0% /dev
tmpfs             3999704       0      3999704         0% /dev/shm
/dev/sda8         484939832    75839044    384393732      17% /
tmpfs             3999708     838356     3161352        21% /tmp
/dev/sda6         50090536     11560     47502080        1% /var/cache/openafs
AFS               2147483647   0      2147483647     0% /afs
tmpfs             799940       216       799724         1% /run/user/4955
pmsvetovidova@dk8n73 ~ $ du

```

Рис. 3.15: df and du

```

2 ./ski.places/equipment
2 ./ski.places/plans
6 ./ski.places
2 ./play/games
4 ./play
2 ./my_os
54875 ./bin
580909 .
pmsvetovidova@dk8n73 ~ $ man fibd
Нет справочной страницы для fibd
pmsvetovidova@dk8n73 ~ $ man find

```

Рис. 3.16: df and du

Воспользовавшись справкой команды `find`, вывела имена всех директорий, имеющих в домашнем каталоге. (рис. [3.17]) (рис. [3.18]) (рис. [??])

```
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).

    This manual page talks about 'options' within the expression list. These options control the behaviour of find but are 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 any remaining arguments are not options, but this does not really work due to the way find determines the end of the following path arguments: it does that by reading until an expression argument comes (which also starts with a -). Now, if a path argument would start with a -, then find would treat it as expression argument instead. Thus, to ensure that all start points are taken as such, and especially to prevent that wildcard patterns expanded by the calling shell are not mistakenly treated as expression arguments, it is generally safer to prefix wildcards or dubious path names with either ./ or to use absolute path names starting with /. Alternatively, it is generally safe though non-portable to use the GNU option -files0-from to pass arbitrary starting points to find.

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

Рис. 3.17: команда `find`

```

/afs/.dk.sci.pfu.edu.ru/home/p/m/pmsvetovidova/work/pmsvetovidova.github.io/.git/objects
/afs/.dk.sci.pfu.edu.ru/home/p/m/pmsvetovidova/work/pmsvetovidova.github.io/.git/objects
/afs/.dk.sci.pfu.edu.ru/home/p/m/pmsvetovidova/work/pmsvetovidova.github.io/.git/objects
/afs/.dk.sci.pfu.edu.ru/home/p/m/pmsvetovidova/work/pmsvetovidova.github.io/.git/objects
/afs/.dk.sci.pfu.edu.ru/home/p/m/pmsvetovidova/work/pmsvetovidova.github.io/.git/logs/re
/afs/.dk.sci.pfu.edu.ru/home/p/m/pmsvetovidova/work/pmsvetovidova.github.io/.git/logs/re
ads
/afs/.dk.sci.pfu.edu.ru/home/p/m/pmsvetovidova/work/pmsvetovidova.github.io/.git/logs/re
motes
/afs/.dk.sci.pfu.edu.ru/home/p/m/pmsvetovidova/work/pmsvetovidova.github.io/.git/logs/re
motes/origin
/afs/.dk.sci.pfu.edu.ru/home/p/m/pmsvetovidova/.texlive2022
/afs/.dk.sci.pfu.edu.ru/home/p/m/pmsvetovidova/.texlive2022/texmf-var
/afs/.dk.sci.pfu.edu.ru/home/p/m/pmsvetovidova/.texlive2022/texmf-var/luatex-cache
/afs/.dk.sci.pfu.edu.ru/home/p/m/pmsvetovidova/.texlive2022/texmf-var/luatex-cache/gener
/afs/.dk.sci.pfu.edu.ru/home/p/m/pmsvetovidova/.texlive2022/texmf-var/luatex-cache/gener
mes
/afs/.dk.sci.pfu.edu.ru/home/p/m/pmsvetovidova/.texlive2022/texmf-var/luatex-cache/gener
nts
/afs/.dk.sci.pfu.edu.ru/home/p/m/pmsvetovidova/.texlive2022/texmf-var/luatex-cache/gener
nts/otl
/afs/.dk.sci.pfu.edu.ru/home/p/m/pmsvetovidova/GNUstep
/afs/.dk.sci.pfu.edu.ru/home/p/m/pmsvetovidova/GNUstep/Library
/afs/.dk.sci.pfu.edu.ru/home/p/m/pmsvetovidova/GNUstep/Library/Services
/afs/.dk.sci.pfu.edu.ru/home/p/m/pmsvetovidova/mkdir
/afs/.dk.sci.pfu.edu.ru/home/p/m/pmsvetovidova/australia
/afs/.dk.sci.pfu.edu.ru/home/p/m/pmsvetovidova/monthly
/afs/.dk.sci.pfu.edu.ru/home/p/m/pmsvetovidova/reports
/afs/.dk.sci.pfu.edu.ru/home/p/m/pmsvetovidova/reports/monthly
/afs/.dk.sci.pfu.edu.ru/home/p/m/pmsvetovidova/reports/monthly/monthly
/afs/.dk.sci.pfu.edu.ru/home/p/m/pmsvetovidova/reports/monthly/monthly.01
/afs/.dk.sci.pfu.edu.ru/home/p/m/pmsvetovidova/reports/monthly/monthly.01/monthly
/afs/.dk.sci.pfu.edu.ru/home/p/m/pmsvetovidova/ski.places
/afs/.dk.sci.pfu.edu.ru/home/p/m/pmsvetovidova/ski.places/equipment
/afs/.dk.sci.pfu.edu.ru/home/p/m/pmsvetovidova/ski.places/plans
/afs/.dk.sci.pfu.edu.ru/home/p/m/pmsvetovidova/play
/afs/.dk.sci.pfu.edu.ru/home/p/m/pmsvetovidova/play/games
/afs/.dk.sci.pfu.edu.ru/home/p/m/pmsvetovidova/my_os
/afs/.dk.sci.pfu.edu.ru/home/p/m/pmsvetovidova/bin
pmsvetovidova@dk8n73 ~ $ find ~ -type d -print

```

Рис. 3.18: команда find

4 Выводы

Ознакомилась с инструментами поиска файлов и фильтрацией текстовых данных, приобрела практические навыки, как по управлению процессами, так и по проверке использования диска и обслуживанию файловых систем.

5 Контрольные вопросы

1. Какие потоки ввода вывода вы знаете?
2. Объясните разницу между операцией > и ».
3. Что такое конвейер?
4. Что такое процесс? Чем это понятие отличается от программы?
5. Что такое PID и GID?
6. Что такое задачи и какая команда позволяет ими управлять?
7. Найдите информацию об утилитах top и htop. Каковы их функции?
8. Назовите и дайте характеристику команде поиска файлов. Приведите примеры использования этой команды.
9. Можно ли по контексту (содержанию) найти файл? Если да, то как?
10. Как определить объем свободной памяти на жестком диске?
11. Как определить объем вашего домашнего каталога?
12. Как удалить зависший процесс?

6 Ответы на вопросы

1. 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.
 - перенаправление `stdout` (вывода) в файл. Если файл отсутствовал, то он создаётся, иначе - перезаписывается.
 - перенаправление `stdout` (вывода) в файл. Если файл отсутствовал, то он создаётся, иначе - добавляется.
3. Конвейер (`pipe`) служит для объединения простых команд или утилит в цепочки, в которых результат работы предыдущей команды передаётся последующей.
4. Процесс это - совокупность программного кода и данных, загруженных в память ЭВМ. 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. Процессом называют выполняющуюся программу и все её элементы: адресное пространство, глобальные переменные, регистры, стек, открытые файлы и так далее.
5. `PID` — уникальный номер (идентификатор) процесса в многозадачной ОС. `GID` – идентификатор группы.

6. Любую выполняющуюся в консоли команду или внешнюю программу можно запустить в фоновом режиме. Запущенные фоновые программы называются задачами (jobs). Ими можно управлять с помощью команды `jobs`, которая выводит список запущенных в данный момент задач. Для завершения задачи необходимо выполнить команду `kill %номер задачи`.
7. Команда `top` в Linux системах позволяет вывести в виде таблицы перечень запущенных процессов и оценить, какой объем ресурсов они потребляют, т.е., какую нагрузку создают на сервер и дисковую подсистему. Команда `htop` — продвинутый монитор процессов, показывает динамический список системных процессов, список обычно выравнивается по использованию ЦПУ. В отличие от `top`, `htop` показывает все процессы в системе. Также показывает время непрерывной работы, использование процессоров и памяти. `htop` часто применяется в тех случаях, когда информации, даваемой утилитой `top` недостаточно, например при поиске утечек памяти в процессах.
8. Команда `find` используется для поиска и отображения имён файлов, соответствующих заданной строке символов. Формат команды: `find путь [-опции]` Путь определяет каталог, начиная с которого по всем подкаталогам будет вестись поиск. Примеры:
- вывести на экран имена файлов из вашего домашнего каталога и его подкаталогов, начинающихся на `f`: `find ~ -name "f*" -print`
 - вывести на экран имена файлов в каталоге `/etc`, начинающихся с символа `p`: `find /etc -name "p*" -print`
 - найти в вашем домашнем каталоге файлы, имена которых заканчиваются символом `~` и удалить их: `find ~ -name "*~" -exec rm "{}" ;`
9. Найти файл по контексту (содержанию) позволяет команда `grep`. Формат команды: `grep строка имя_файла` Примеры:
- показать строки во всех файлах в вашем домашнем каталоге с именами, начинающимися на `f`, в которых есть слово `begin`: `grep begin f*`

- найти в текущем каталоге все файлы, в имени которых есть буквосочетание «лаб»: `ls -l | grep лаб`
10. Определить объем свободной памяти на жёстком диске позволяет команда `df`.
 11. Определить объем домашнего каталога позволяет команда `df /home/`
 12. Удалить зависший процесс можно командой `kill %номер задачи`.