

# Лабораторная работа №6

Поиск файлов. Перенаправление ввода-вывода. Просмотр запущенных процессов

Мальков Роман

# Содержание

Цель работы	3
Задание	4
Теоретическое введение	6
Ход работы	8
Выводы	19

# 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.

# 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.
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.

13.

# 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

# означает, что вывод команды 1 передаётся на ввод команде 2

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

find путь [-опции]

- Найти в текстовом файле указанную строку символов позволяет команда grep.

Формат команды:

grep строка имя\_файла

Команда df показывает размер каждого смонтированного раздела диска. Формат команды:

df [-опции] [файловая\_система]

Команда du показывает число килобайт, используемое каждым файлом или каталогом. Формат команды:

du [-опции] [имя\_файла...]

- Любую выполняющуюся в консоли команду или внешнюю программу можно запустить в фоновом режиме. Для этого следует в конце имени команды указать знак амперсанд &. Например:

gedit &

\*Запущенные фоном 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:

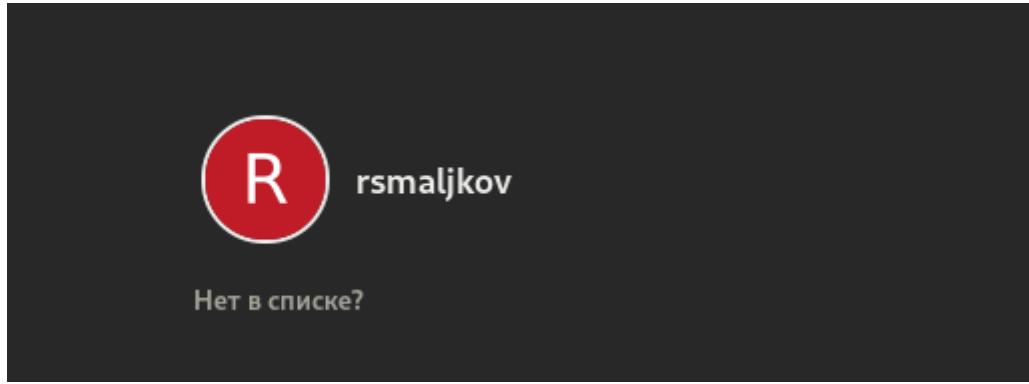
kill %номер задачи

Команда ps используется для получения информации о процессах. Формат команды:

ps [-опции]

# Ход работы

Осуществляем вход в систему, используя соответствующее имя и пароль(Скриншот 1).



(Скриншот 1)

Записываем в файл file.txt названия файлов, содержащихся в каталоге /etc. Дописываем в этот же файл названия файлов, содержащихся в вашем домашнем каталоге(Скриншот 2).

The screenshot shows a terminal window with the following session:

```
[rsmaljkov@rsmaljkov /]$ ls -a /etc|sort>file.txt
bash: file.txt: Permission denied
[rsmaljkov@rsmaljkov /]$ sudo -s
[sudo] password for rsmaljkov:
Sorry, try again.
[sudo] password for rsmaljkov:
[root@rsmaljkov /]$ ls -a /etc|sort>file.txt
[root@rsmaljkov /]# ls -a /home/rsmaljkov|sort>>file.txt
[root@rsmaljkov /]# cat file.txt
.
..
abrt
adjtime
aliases
alsa
alternatives
anaconda
anthy-unicode.conf
appstream.conf
asound.conf
audit
authselect
avahi
bash_completion.d
```

(Скриншот 1)

Выводим имена всех файлов из file.txt, имеющих расширение .conf, после чего записываем их в новый текстовой файл conf.txt (Скриншот 3).

The screenshot shows a terminal window with the following content:

```
root@rsmaljkov:/  
.vboxclient-clipboard.pid  
.vboxclient-display-svga-xli.pid  
.vboxclient-draganddrop.pid  
.vboxclient-seamless.pid  
Videos  
.wget-hsts  
work  
[root@rsmaljkov /]# grep .conf file.txt|sort>conf.txt  
[root@rsmaljkov /]# cat conf.txt  
anthy-unicode.conf  
appstream.conf  
asound.conf  
brltty.conf  
chkconfig.d  
chrony.conf  
.config  
dconf  
dleyna-renderer-service.conf  
dleyna-server-service.conf  
dnsmasq.conf  
dracut.conf  
dracut.conf.d  
extlinux.conf  
fprintd.conf
```

(Скриншот 3)

Определяем, какие файлы в нашем домашнем каталоге имеют имена, начинающиеся с символа с. Выполняем несколькими способами (Скриншоты 4 - 5).

The screenshot shows a terminal window with a dark theme. The title bar reads "root@rsmaljkov:/". The window contains a list of configuration files in the root directory, followed by the output of a "find" command. The "find" command was run from the root directory to search for files named "c\*" in the user's home directory, "/home/rsmaljkov".

```
resolv.conf
rsyncd.conf
rygel.conf
sestatus.conf
sudo.conf
swtpm-localca.conf
swtpm_setup.conf
sysconfig
sysctl.conf
tcsd.conf
Trolltech.conf
updatedb.conf
uresourced.conf
usb_modeswitch.conf
vconsole.conf
xattr.conf
[root@rsmaljkov /]# find /home/rsmaljkov -name "c*" -print
/home/rsmaljkov/.mozilla/firefox/y3cp5brv.default-release/crashes
/home/rsmaljkov/.mozilla/firefox/y3cp5brv.default-release/compatibility.ini
/home/rsmaljkov/.mozilla/firefox/y3cp5brv.default-release/cookies.sqlite
/home/rsmaljkov/.mozilla/firefox/y3cp5brv.default-release/storage/permanent/chrome
/home/rsmaljkov/.mozilla/firefox/y3cp5brv.default-release/cert9.db
/home/rsmaljkov/.mozilla/firefox/y3cp5brv.default-release/content-prefs.sqlite
```

(Скриншот 4)

```
```
```
placeimg_800_600_tech.jpg

/home/rsmaljkov/work/study/2021-2022/OperatingSystems/study_2021-2022_os-intro/t
emplate/report/report/pandoc:
```
```
csl

/home/rsmaljkov/work/study/2021-2022/OperatingSystems/study_2021-2022_os-intro/t
emplate/report/report/pandoc/csl:
```
```
gost-r-7-0-5-2008-numeric.csl

/home/rsmaljkov/work/study/2021-2022/OperatingSystems/study_2021-2022_os-intro/t
emplate/report/scripts:
```
```
image-report
mpv-shot
[root@rsmaljkov /]# ls -aR /home/rsmaljkov|grep "*c"
```

(Скриншот 5)

Выводим на экран имена файлов из каталога /etc, начинающиеся с символа h(Скриншот 6).

```
[root@rsmaljkov /]# find /etc -name "*h" -print
/etc/X11/xinit/xinitrc.d/50-systemd-user.sh
/etc/X11/xinit/xinitrc.d/localuser.sh
/etc/X11/xinit/xinitrc.d/00-start-message=bus.sh
/etc/X11/xinit/xinitrc.d/10-qt5-check-opengl2.sh
/etc/X11/xinit/xinitrc.d/98vboxadd-xclient.sh
/etc/authselect/system-auth
/etc/authselect/password-auth
/etc/authselect/fingerprint-auth
/etc/authselect/smartcard-auth
/etc/bash_completion.d/authselect-completion.sh
/etc/bluetooth
/etc/ceph
/etc/dhcp/dhclient.d/chrony.sh
/etc/libssh
```

(Скриншот 6)

Запускаем в фоновом режиме процесс, который будет записывать в файл `~/logfile` файлы, имена которых начинаются с `log`(Скриншот 7).

```
[rsmaljkov@rsmaljkov ~]$ find / -name "log*" -print > logfile &
[1] 5608
find: '/boot/lost+found': Permission denied
find: '/boot/loader/entries': Permission denied
find: '/boot/efi': Permission denied
find: '/boot/grub2': Permission denied
[rsmaljkov@rsmaljkov ~]$ find: '/home/rsmaljkov/play/games/play': Permission denied
find: '/proc/tty/driver': Permission denied
find: '/proc/1/task/1/fd': Permission denied
find: '/proc/1/task/1/ns': Permission denied
find: '/proc/1/fd': Permission denied
find: '/proc/1/map_files': Permission denied
find: '/proc/1/ns': Permission denied
find: '/proc/2/task/2/fd': Permission denied
find: '/proc/2/task/2/ns': Permission denied
find: '/proc/2/fd': Permission denied
find: '/proc/2/map_files': Permission denied
find: '/proc/2/ns': Permission denied
find: '/proc/3/task/3/fd': Permission denied
find: '/proc/3/task/3/ns': Permission denied
find: '/proc/3/fd': Permission denied
find: '/proc/3/map_files': Permission denied
find: '/proc/3/ns': Permission denied
```

(Скриншот 7)

Удаляем файл `~/logfile`(Скриншот 8).

```
[rsmaljkov@rsmaljkov ~]$ rm -f logfile
[rsmaljkov@rsmaljkov ~]$
```

(Скриншот 8)

Запускаем из консоли в фоновом режиме редактор gedit. Определяем идентификатор процесса gedit, используя команду ps, конвейер и фильтр grep.(Скриншот 9).

```
[rsmaljkov@rsmaljkov ~]$ gedit &
[1] 5864
[rsmaljkov@rsmaljkov ~]$ ps aux|grep gedit
rsmaljk+ 5864 0.5 3.0 777892 60708 pts/1    Sl 12:48 0:00 gedit
rsmaljk+ 5945 0.0 0.1 221668 2300 pts/1    S+ 12:49 0:00 grep --color=
auto gedit
[rsmaljkov@rsmaljkov ~]$
```

(Скриншот 9)

Читаем справку (man) команды kill, после чего используем её для завершения

процесса gedit(скриншот 10 - 11).

The screenshot shows a terminal window with the title bar "rsmaljkov@rsmaljkov:~ — man kill". The content of the window is the man page for the kill command. It includes sections for options like -0, -1, -n, and --name, and descriptions for each. It also includes sections for OPTIONS, -s, --signal signal, and -l, --list [number]. The bottom of the window shows the footer "Manual page kill(1) line 43 (press h for help or q to quit)".

```
0
      All processes in the current process group are signaled.

-1
      All processes with a PID larger than 1 are signaled.

-n
      where n is larger than 1. All processes in process group n are
      signaled. When an argument of the form "-n" is given, and it is
      meant to denote a process group, either a signal must be
      specified first, or the argument must be preceded by a '--'
      option, otherwise it will be taken as the signal to send.

name
      All processes invoked using this name will be signaled.

OPTIONS
-s, --signal signal
      The signal to send. It may be given as a name or a number.

-l, --list [number]
      Print a list of signal names, or convert the given signal number to
```

Manual page kill(1) line 43 (press h for help or q to quit)

(Скриншот 10)

The screenshot shows a terminal window with the title bar "[rsmaljkov@rsmaljkov ~]\$". The user runs several commands: gedit & (background gedit), ps aux | grep gedit (finds the process ID 5864), kill 5945 (tries to kill process 5945, which does not exist), kill 5864 (kills the gedit process). The output is in red.

```
[rsmaljkov@rsmaljkov ~]$ gedit &
[1] 5864
[rsmaljkov@rsmaljkov ~]$ ps aux|grep gedit
rsmaljk+ 5864 0.5 3.0 777892 60768 pts/1    Sl 12:48 0:00 gedit
rsmaljk+ 5945 0.0 0.1 221668 2300 pts/1    S+ 12:49 0:00 grep --color=auto gedit
[rsmaljkov@rsmaljkov ~]$ man kill
[rsmaljkov@rsmaljkov ~]$ kill 5945
bash: kill: (5945) - No such process
[rsmaljkov@rsmaljkov ~]$ kill 5864
[1]+  Terminated          gedit
[rsmaljkov@rsmaljkov ~]$
```

(Скриншот 11)

Выполняем команды df и du, предварительно получив более подробную информацию об этих командах, с помощью команды man(Скриншоты 12 - 15).

rsmaljkov@rsmaljkov:~ — man df

**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 containing each file name argument. If no file name is given, the space available on all currently mounted file systems is shown. Disk 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 disk 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 non-portable intimate 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.

Manual page df(1) line 7 (press h for help or q to quit)

(Скриншот 12)

```
[rsmaljkov@rsmaljkov ~]$ df
Filesystem      1K-blocks    Used Available Use% Mounted on
devtmpfs          4096        0     4096   0% /dev
tmpfs            1010444       0  1010444   0% /dev/shm
tmpfs            404180     1364    402816   1% /run
/dev/sda2        82836480  5532760  75639800   7% /
tmpfs            1010444     452   1009992   1% /tmp
/dev/sda2        82836480  5532760  75639800   7% /home
/dev/sdal         996780    222952   705016  25% /boot
files           153561084 17496648 136064436  12% /media/sf_files
tmpfs            202088     148    201940   1% /run/user/1000
[rsmaljkov@rsmaljkov ~]$
```

(Скриншот 13)

```
DU(1) User Commands DU(1)

NAME
    du - estimate file space usage

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

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

(Скриншот 14)

```
[rsmaljkov@rsmaljkov ~]$ du
0      ./mozilla/extensions/(ec8030f7-c28a-464f-9b0e-13a3a9e97384)
0      ./mozilla/extensions
0      ./mozilla/plugins
0      ./mozilla/firefox/Crash Reports/events
0      ./mozilla/firefox/Crash Reports
0      ./mozilla/firefox/Pending Pings
0      ./mozilla/firefox/y3cp5brv.default-release/minidumps
0      ./mozilla/firefox/y3cp5brv.default-release/crashes/events
4      ./mozilla/firefox/y3cp5brv.default-release/crashes
```

(Скриншот 15)

Воспользовавшись справкой команды find, выводим имена всех директорий, имеющихся в вашем домашнем каталоге(Скриншоты 16 - 17).

```
rsmaljkov@rsmaljkov:~ — man find

File's status was last changed less than, more than or exactly n
minutes ago.

-cnewer reference
        Time of the last status change of the current file is more re-
        cent than that of the last data modification of the reference
        file. If reference is a symbolic link and the -H option or the
        -L option is in effect, then the time of the last data modifica-
        tion of the file it points to is always used.

-ctime n
        File's status was last changed less than, more than or exactly
        n*24 hours ago. See the comments for -atime to understand how
        rounding affects the interpretation of file status change times.

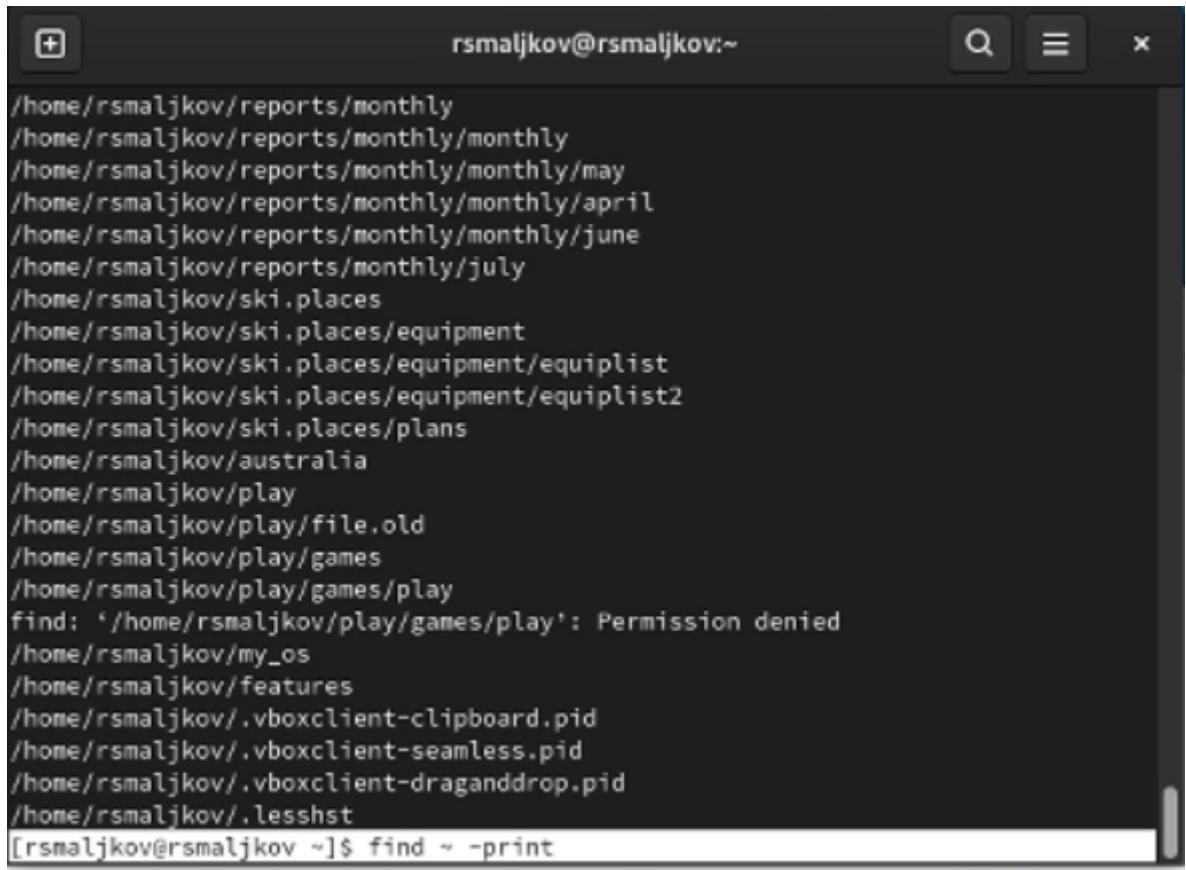
-empty File is empty and is either a regular file or a directory.

[

-executable
        Matches files which are executable and directories which are
        searchable (in a file name resolution sense) by the current
        user. This takes into account access control lists and other
        permissions artefacts which the -perm test ignores. This test
        makes use of the access(2) system call, and so can be fooled by

Manual page find(1) line 400 (press h for help or q to quit)
```

(Скриншот 16)



The screenshot shows a terminal window with the following content:

```
rsmaljkov@rsmaljkov:~
```

```
/home/rsmaljkov/reports/monthly
/home/rsmaljkov/reports/monthly/monthly
/home/rsmaljkov/reports/monthly/monthly/may
/home/rsmaljkov/reports/monthly/monthly/april
/home/rsmaljkov/reports/monthly/monthly/june
/home/rsmaljkov/reports/monthly/july
/home/rsmaljkov/ski.places
/home/rsmaljkov/ski.places/equipment
/home/rsmaljkov/ski.places/equipment/equiplist
/home/rsmaljkov/ski.places/equipment/equiplist2
/home/rsmaljkov/ski.places/plans
/home/rsmaljkov/australia
/home/rsmaljkov/play
/home/rsmaljkov/play/file.old
/home/rsmaljkov/play/games
/home/rsmaljkov/play/games/play
find: '/home/rsmaljkov/play/games/play': Permission denied
/home/rsmaljkov/my_os
/home/rsmaljkov/features
/home/rsmaljkov/.vboxclient-clipboard.pid
/home/rsmaljkov/.vboxclient-seamless.pid
/home/rsmaljkov/.vboxclient-draganddrop.pid
/home/rsmaljkov/.lessht
[rsmaljkov@rsmaljkov ~]$ find ~ -print
```

(Скриншот 17)

## 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.