

# Презентация по лабораторной работе №6

Операционные системы

---

Попова Елизавета Сергеевна, НММбд-03-22

18 марта 2023

Российский университет дружбы народов, Москва, Россия

## Цель работы

---

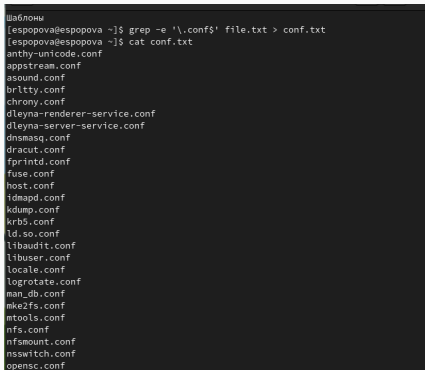
Ознакомиться с инструментами поиска файлов и фильтрации текстовых данных. Приобрести практические навыки: по управлению процессами (и заданиями), по проверке использования диска и обслуживанию файловых систем.

# Выполнение лабораторной работы

1. Осуществили вход в систему, используя наше имя.
2. Далее запишем в файл *file.txt* названия файлов, содержащихся в каталоге */etc*. Для этого используем команду *ls -a /etc >file.txt*. С помощью команды *ls -a ~ - > file.txt* дописываем в этот же файл названия файлов, содержащихся в домашнем каталоге. Для проверки действий используем команду *cat file.txt*.

```
[espopova@espopova ~]$ ls -a /etc > file.txt
[espopova@espopova ~]$ ls -a ~ >> file.txt
[espopova@espopova ~]$ cat file.txt
.
..
abrt
adjtime
aliases
alsa
alternatives
anaconda
anthy-unicode.conf
appstream.conf
asound.conf
audit
authselect
avahi
bash_completion.d
bashrc
bindresvport.blacklist
binfmt.d
bluetooth
brlapi.key
brltty
brltty.conf
ceph
chkconfig.d
chromium
```

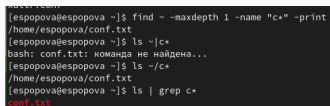
3. Нужно вывести имена всех файлов из *file.txt*, которые имеют расширение *.conf* и записать их в новый текстовый файл *conf.txt*. Для этого используем команду *grep -e '.conf\$' file.txt > conf.txt*. Проверяем выполнение действий.



```
Шаблоны
[espopova@espopova ~]$ grep -e '\.conf$' file.txt > conf.txt
[espopova@espopova ~]$ cat conf.txt
anthy-unicode.conf
appstream.conf
asound.conf
brltty.conf
chrony.conf
dley-na-renderer-service.conf
dley-na-server-service.conf
dnsmasq.conf
dracut.conf
fprintd.conf
fuse.conf
host.conf
idmapd.conf
kdump.conf
krb5.conf
ld.so.conf
libaudit.conf
libuser.conf
locale.conf
logrotate.conf
man_db.conf
mke2fs.conf
mttools.conf
nfs.conf
nfsmount.conf
nsswitch.conf
opensc.conf
```

Рис. 2: Вывод файлов

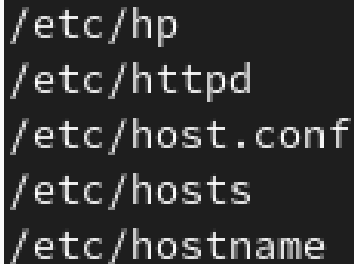
4. Затем найдём файлы в домашнем каталоге, которые начинаются на с. Это можно сделать несколькими командами, которые представлены на рисунке.

A terminal window with a dark background. The user 'espopova@espopova' is in the home directory '~'. The first command is 'find ~ -maxdepth 1 -name "с\*" -print', which outputs '/home/espopova/conf.txt'. The second command is 'ls ~/с\*', which results in a 'bash: conf.txt: команда не найдена...' error. The third command is 'ls ~/с\*', which outputs '/home/espopova/conf.txt'. The fourth command is 'ls | grep с\*', which outputs 'conf.txt' in red text.

```
[espopova@espopova ~]$ find ~ -maxdepth 1 -name "с*" -print
/home/espopova/conf.txt
[espopova@espopova ~]$ ls ~/с*
bash: conf.txt: команда не найдена...
[espopova@espopova ~]$ ls ~/с*
/home/espopova/conf.txt
[espopova@espopova ~]$ ls | grep с*
conf.txt
```

Рис. 3: Нахождение файлов по символу

5. После этого выведем на экран (по странично) имена файлов из каталога */etc*, которые начинаются с символа *h*. Для этого я использовала команду `* find /etc -maxdepth1 -name "h" | less`.

A terminal window with a dark background and light-colored text. It displays the output of a command, showing five file paths: /etc/hp, /etc/httpd, /etc/host.conf, /etc/hosts, and /etc/hostname, each on a new line.

```
/etc/hp
/etc/httpd
/etc/host.conf
/etc/hosts
/etc/hostname
```

Рис. 4: Нахождение файлов по символу

6. Запустим в фоновом режиме процесс, который будет записывать в файл `~/logfile`, файлы, которые начинаются с `log` с помощью команды `find / -name "log" > logfile&»*`. Запустился непрерывный процесс записывания файла.

```
lespopova@lespopova ~]$ find/-name "log">logfile&>>
bash: *: неоднозначное перенаправление
lespopova@lespopova ~]$ find / -name "log" > ~/logfile
find: '/boot/loader/entries': Отказано в доступе
find: '/boot/grub2': Отказано в доступе
find: '/boot/lost+found': Отказано в доступе
find: '/boot/efi': Отказано в доступе
find: '/proc/tty/driver': Отказано в доступе
find: '/proc/1/task/1/fd': Отказано в доступе
find: '/proc/1/task/1/fdinfo': Отказано в доступе
find: '/proc/1/task/1/ns': Отказано в доступе
find: '/proc/1/fd': Отказано в доступе
find: '/proc/1/map_files': Отказано в доступе
find: '/proc/1/fdinfo': Отказано в доступе
find: '/proc/1/ns': Отказано в доступе
find: '/proc/2/task/2/fd': Отказано в доступе
find: '/proc/2/task/2/fdinfo': Отказано в доступе
find: '/proc/2/task/2/ns': Отказано в доступе
find: '/proc/2/fd': Отказано в доступе
find: '/proc/2/map_files': Отказано в доступе
find: '/proc/2/fdinfo': Отказано в доступе
find: '/proc/2/ns': Отказано в доступе
find: '/proc/3/task/3/fd': Отказано в доступе
find: '/proc/3/task/3/fdinfo': Отказано в доступе
find: '/proc/3/task/3/ns': Отказано в доступе
find: '/proc/3/fd': Отказано в доступе
find: '/proc/3/map_files': Отказано в доступе
find: '/proc/3/fdinfo': Отказано в доступе
find: '/proc/3/ns': Отказано в доступе
find: '/proc/4/task/4/fd': Отказано в доступе
find: '/proc/4/task/4/fdinfo': Отказано в доступе
find: '/proc/4/task/4/ns': Отказано в доступе
find: '/proc/4/fd': Отказано в доступе
find: '/proc/4/map_files': Отказано в доступе
find: '/proc/4/fdinfo': Отказано в доступе
```

Рис. 5: Нахождение файлов по символам




7. Проверим наличие файла *logfile*, а затем с помощью команды *rm logfile* удалим его.

```
[espopova@espopova ~]$ rm logfile  
[espopova@espopova ~]$ rm logfile  
rm: невозможно удалить 'logfile': Нет такого файла или каталога  
[espopova@espopova ~]$
```

Рис. 6: Удаление файла

8. Запускаем в консоли в фоновом режиме редактор *gedit*. После ввода команды *gedit &* появляется окно редактора.



```
[espopova@espopova ~]$ gedit &  
[1] 4013
```

Рис. 7: Редактор gedit

9. Для определения идентификатора процесса *gedit* используем команду *ps | grep -i "gedit"*.  
Из рисунка видно, что наш процесс имеет PID 4507.

```
[espopova@espopova ~]$ gedit &
[1] 4013
[espopova@espopova ~]$ ps | grep -i "gedit"
[1]+  Завершён          gedit
[espopova@espopova ~]$ pgrep gedit
[espopova@espopova ~]$ pidof gedit
[espopova@espopova ~]$
```

Рис. 8: Определение идентификатора процесса

10. Далее ознакомимся со справкой команды *kill* и используем её для завершения процесса *gedit*.

```
kill(1)                                User Commands
kill - terminate a process

SYNOPSIS
kill [-signal=-s signal][-s] [-n value] [-w] [-timeout milliseconds signal] [-o pidname...]
kill -l {name} | -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 for the SIGKILL signal to allow 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 signal cannot be caught, and so does not give the target process the opportunity to perform any clean-up before terminating.

Most modern shells have a built-in kill command, with a syntax rather similar to that of the command described here. The --all, --pid, and --queue options, and the --posix extension, are not supported.

If signal is 0, then no actual signal is sent, but error checking is still performed.

ARGUMENTS
The list of processes to be signaled can be a mixture of names and PIDs.

PID
Each pid can be expressed in one of the following ways:
- A
  where a is larger than 0. The process with PID a is signaled.
- #
  All processes in the current process group are signaled.
- -#
  All processes with a PID larger than # are signaled.
- %g
  where g is larger than 1. All processes in process group g are signaled. When an argument of the form "-g" is given, and % is meant to denote a process group, must be preceded by a "+" option, otherwise it will be taken as the signal to send.

NAME
This document describes using this tool.

BASIC USAGE
kill [signal] [-s signal] [-w] [-o pidname] [-o pidname]...
```

Рис. 9: Опции команды kill

```
ch@pop-os:~$ ps -ef | grep gedit
ch@pop-os:~$ kill -9 4011
ch@pop-os:~$ ps -ef | grep gedit
```

Рис. 10: Завершение процесса

11. Далее получим более подробную информацию о командах *df* и *du*.

- *df*– утилита, показывающая список всех файловых систем по именам устройств, сообщает их размер, занятое и свободное пространство и точки монтирования.
- *du* – утилита, предназначенная для вывода информации об объеме дискового пространства, занятого файлами и директориями. Она принимает путь к элементу файловой системы и выводит информацию о количестве байт дискового пространства или блоков диска, задействованных для его хранения

```
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 or system 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 shown. 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. If no argument is given, df shows the space available on all file systems. df does not show the space available on unmounted file systems, because on most kinds of systems doing so requires very nonportable intimate knowledge of the file system.

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., "-BK" prints sizes in units of 1,048,576 bytes; see SIZE format below
    --direct
        show statistics for a file instead of mount point
    -h, --human-readable
        print sizes in powers of 1024 (e.g., 102M)
    -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
    -l, --local
        limit listing to local file systems
```

```

[111]                                     User Commands
NAME
    du - estimate file space usage

SYNOPSIS
    du [OPTION]... [FILE]...
    du [OPTION]... --file-segment

DESCRIPTION
    Summarize device usage of the set of FILES, recursively for directories.
    Mandatory arguments to long options are mandatory for short options too.

    -D, --null
        end each output line with NUL, not newline

    -s, --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 ('

    -B, --block-size=SIZE
        scale sizes by SIZE before printing them; e.g., '-B8' 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-dep

    --file-segment=Z
        summarize device usage of the NUL-terminated file names specified in file F; if F is -, then read names from standard input

    -H
        equivalent to --dereference-args (-D)

```

Рис. 12: Опции команды du

```

#df -h
Filesystem      1K-Blocks  Used Avail Use% Mounted on
devtmpfs        4896        0  4896   0% /dev
tmpfs           2005856    0 2005856   0% /dev/shm
tmpfs           972344    1420  969344   1% /run
/dev/sda3       82034432 14541068 65227332 18% /
tmpfs          2005856    0 2005856   0% /tmp
/dev/sda3       82034432 14541068 65227332 18% /home
/dev/sda2       998768    245812  682156  27% /boot
tmpfs          401368     148  401220   1% /run/user/1000
tmpfs          51716      0  51716   0% /run/media/espovna/86ox_Oas_T_d.2
#df -h --output=du

```

Рис. 13: Команда df

```
12662 ./webc
1360 ./texlive2021/texmf-var/webc/lsu00tex
1360 ./texlive2021/texmf-var/webc
1360 ./texlive2021/texmf-var/luatex-cache/generic/rares
15736 ./texlive2021/texmf-var/luatex-cache/generic/fonts/otl
15736 ./texlive2021/texmf-var/luatex-cache/generic/fonts
17292 ./texlive2021/texmf-var/luatex-cache/generic
17292 ./texlive2021/texmf-var/luatex-cache
16592 ./texlive2021/texmf-var
16592 ./texlive2021
14516 ./bin
1 ./bin
```

Рис. 14: Команда du

12. Выведем имена всех директорий, которые имеются в домашнем каталоге, предварительно узнаем опции команды *find*.

[illegible]

Рис. 15: Опции команды find

[illegible]



## Выводы

---

В ходе выполнения данной лабораторной работы я ознакомилась с инструментами поиска файлов и фильтрации текстовых данных, а также приобрела практические навыки по управлению процессами, по проверке использования диска и обслуживанию файловых систем.