

Лабораторная работа № 7.

Кальсин Захар Алексеевич

Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6

List of Tables

List of Figures

1 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.

2 Выполнение лабораторной работы

1. Осуществил вход в систему, используя соответствующее имя пользователя.
2. Записал в файл file.txt названия файлов, содержащихся в каталоге /etc. Дописал в этот же файл названия файлов, содержащихся в домашнем каталоге.

```
zakaljsin@dk3n58 ~ $ ls /etc > file.txt
zakaljsin@dk3n58 ~ $ ls ~ >> file.txt
```

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

```
zakaljsin@dk3n58 ~ $ grep .conf file.txt
appstream.conf
brltty.conf
ca-certificates.conf
cachefilesd.conf
cfg-update.conf
cpufreq-bench.conf
dconf
dhcpcd.conf
dispatch-conf.conf
dley-na-server-service.conf
dnsmasq.conf
e2fsck.conf
e2scrub.conf
etc-update.conf
fluidsynth.conf
fuse.conf
gai.conf
zakaljsin@dk3n58 ~ $ grep .conf
```

4. Определил, какие файлы в домашнем каталоге имеют имена, начинающиеся с символа с. Несколько вариантов, как это сделать:

```
zakaljsin@dk3n58 ~ $ ls -l | grep c*
-rw-r--r-- 1 zakaljsin studsci      0 мая 15 11:46 conf.txt
zakaljsin@dk3n58 ~ $ find ~/c* -name "c*" -print
/afs/.dk.sci.pfu.edu.ru/home/z/a/zakaljsin/conf.txt
```

5. Вывел на экран (по странично) имена файлов из каталога /etc, начинающиеся с символа h.

```
zakaljsin@dk3n58 ~/etc $ ls -l | grep h* | less
итого 0
lines 1-1/1 (END)
```

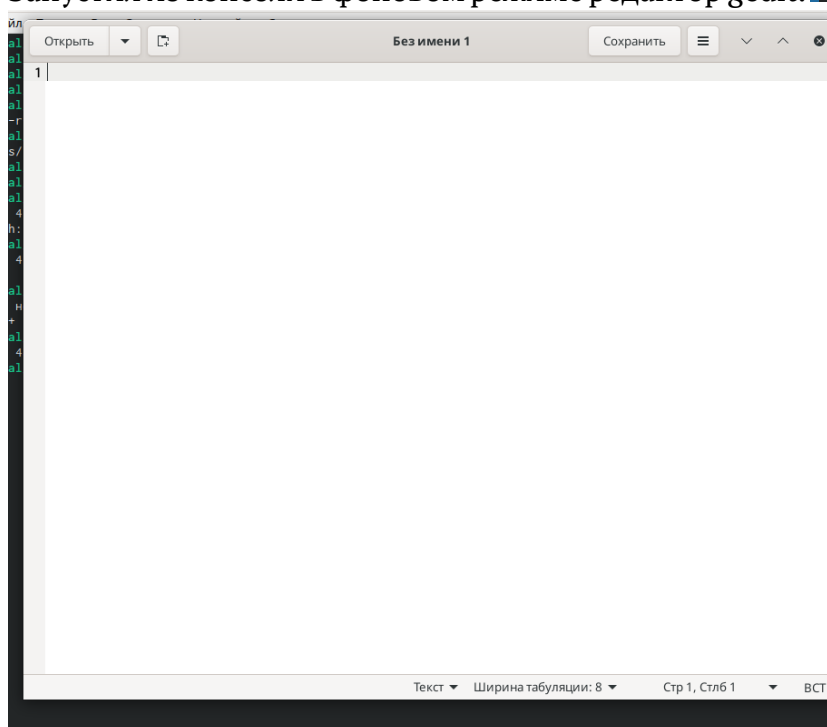
6. Запустил в фоновом режиме процесс, который будет записывать в файл ~/logfile файлы, имена которых начинаются с log.

```
zakaljsin@dk3n58 ~/etc $ find ~ -name "log*" -print > ~/logfile &
[2] 4779
```

7. Удалил файл ~/logfile.

```
zakaljsin@dk3n58 ~/etc $ rm ~/logfile
[1] 4846
```

8. Запустил из консоли в фоновом режиме редактор gedit.



9. Определил идентификатор процесса gedit, используя команду ps, конвейер и

фильтр grep. Более простым способом определить этот идентификатор не получилось.

```
zakaljsin@dk3n58 ~/etc $ ps aux | grep -i gedit
zakaljs+  4936  0.0  0.0 14320  912 pts/0    S+   11:57   0:00 grep --colour=auto -i gedit
[1]+  Завершён      gedit
```

10.

Прочла справку (man) команды kill, после чего использовал её для завершения процесса gedit.

```
zakaljsin@dk3n58 ~/etc $ kill -9 3325
```

```
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.

  -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.

  kill -l 11
    Translate number 11 into a signal name.

  kill -L
    List the available signal choices in a nice table.

  kill 123 543 2341 3453
    Send the default signal, SIGTERM, to all those processes.
SEE ALSO
  kill(2), killall(1), nice(1), pkill(1), renice(1), signal(7), skill(1)
STANDARDS
  This command meets appropriate standards. The -L flag is Linux-specific.
AUTHOR
  Albert Cahalan (albert@users.sf.net) wrote kill in 1999 to replace a bsdutils one that was not standards
Manual page kill(1) line 1 (press h for help or q to quit)
```

11. Выполнил команды df и du, предварительно получив более подробную информацию об этих командах, с помощью команды man.


```

DF (1)                                     User Commands

NAME
    df - report file system disk space usage

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

DESCRIPTION
    This manual page documents the GNU version of df. df displays
    file system containing each file name argument. If no file
    is specified, the file system on which the shell is currently
    mounted is shown. Disk space is shown in 1024-byte blocks
    unless the POSIXLY_CORRECT environment variable is set, in which
    case 512-byte blocks are used.

    If an argument is the absolute file name of a disk device
    rather than a file system name, the space available on that
    device is shown. The version of df cannot show the space
    available on unmounted file systems, so doing so requires
    very nonportable intimate knowledge of the file system.

OPTIONS
    Show information about the file system on which each FILE
    resides.

    Mandatory arguments to long options are mandatory for short
    options.

    -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 powers of 1024 (e.g., 1023M)

    -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, --block-size=K
        like --block-size=1K

    -l, --local
        limit listing to local file systems

    --no-sync
        do not invoke sync before getting usage info (default)

    --output=FIELD_LIST
        use the output format defined by FIELD_LIST, or print
        all fields by default

    -P, --portability
        use the POSIX output format

    --sync
        invoke sync before getting usage info

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

```

```
zaka1jsin@dk3n58 ~/e
2 .
```

9

```

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 operator 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 them 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 -- can also be used to signal that all remaining arguments are not options (though ensuring that all start points begin with either './' or '/' is generally safer if you use wildcards in the list of start points).

    -P      Never follow symbolic links. This is the default behaviour. When find examines or prints information about a file, and the file is a symbolic link, the information used shall be taken from the properties of the symbolic link itself.

    -L      Follow symbolic links. When find examines or prints information about files, the information used shall be taken from the properties of the file to which the link points, not from the link itself (unless it is a broken symbolic link or find is unable to examine the file to which the link points). Use of this option implies -noleaf. If you later use the -P option, -noleaf will still be in effect. If -L is in effect and find discovers a symbolic link to a subdirectory during its search, the subdirectory pointed to by the symbolic link will be searched.

    When the -L option is in effect, the -type predicate will always match against the type of the file to which a symbolic link points rather than the link itself (unless the symbolic link is broken). Actions that can cause symbolic links to become broken while find is executing (for example -delete) can give rise to confusing behaviour. Using -L causes the -lname and -ilname predicates always to return false.

    -H      Do not follow symbolic links, except while processing the command line arguments. When find examines or prints information about files, the information used shall be taken from the properties of the symbolic link itself. The only exception to this behaviour is when a file specified on the command line is a symbolic link.

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

```

имеющихся в домашнем каталоге.

Вывод

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

Ответы на контрольные 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 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.

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`.