

Отчет по лабораторной работе №7

Дисциплина: Операционные системы

Морозова Анастасия Владимировна

Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
3	Выводы	17

List of Tables

List of Figures

2.1	Запись названий файлов в file.txt	6
2.2	Вывод имен всех файлов из file.txt	7
2.3	Файлы, начинающиеся с “с”	7
2.4	Файлы, начинающиеся с “h”	8
2.5	Список файлов	8
2.6	Запись файлов в фоновом процессе	8
2.7	Удаление файла logfile	9
2.8	Запуск редактора gedit	9
2.9	Редактор gedit	9
2.10	Идентификатор процесса gedit	9
2.11	Команда kill	10
2.12	man kill	10
2.13	команды man df и man du	10
2.14	man df	11
2.15	man du	11
2.16	Команда df	11
2.17	Команда du	12
2.18	Команда find ~ -typed	12
2.19	Команда man find	13

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. Осуществляю вход в систему, используя свои логин и пароль.
2. Записываю в файл file.txt названия файлов, содержащихся в каталоге /etc (команда «ls-a/etc > file.txt»). Дописываю в этот же файл названия файлов, содержащихся в домашнем каталоге(команда ls-a~ » file.txt).Командой cat file.txt просматриваю файл, чтобы убедиться в правильности действий (рис. 2.1)

```
avmorofova@avmorofova:~$ ls -a /etc > file.txt
avmorofova@avmorofova:~$ ls -a ~ >> file.txt
avmorofova@avmorofova:~$ cat file.txt
.
..
acpi
adduser.conf
alsa
alternatives          I
anacrontab
apache2
apq.conf
apm
apparmor
apparmor.d
appport
appstream.conf
apt
avahi
bash.bashrc
```

Figure 2.1: Запись названий файлов в file.txt

3. Вывожу имена всех файлов из file.txt, имеющих расширение .conf и записываю их в новый текстовый файл conf.txt (команда grep -e '.conf\$' file.txt

> conf.txt). Проверяю правильность выполненных действий (команда cat conf.txt) (рис. 2.2)

```
avmorofova@avmorofova:~$ grep -e '\.conf$' file.txt > conf.txt
avmorofova@avmorofova:~$ cat cont.txt
cat: cont.txt: No such file or directory
avmorofova@avmorofova:~$ cat conf.txt
adduser.conf
apg.conf
appstream.conf
britty.conf
ca-certificates.conf
debconf.conf
deluser.conf
e2scrub.conf
fprindd.conf
fuse.conf
gat.conf
hdparm.conf
host.conf
kernel-lmg.conf
kerneloops.conf
ld.so.conf
```

Figure 2.2: Вывод имен всех файлов из file.txt

4. Определяю, какие файлы в домашнем каталоге имеют имена, начинающиеся с символа c, несколькими командами:

- find ~ -maxdepth 1 -name "c*" -print (опция maxdepth 1 необходима для того, чтобы файлы находились только в домашнем каталоге, а не в его подкаталогах)
- ls ~ /c*
- ls -a ~ | grep c* (рис. 2.3)

```
avmorofova@avmorofova:~$ find ~ -maxdepth 1 -name "c*" -print
/home/avmorofova/conf.txt
avmorofova@avmorofova:~$ ls ~/c*
/home/avmorofova/conf.txt
avmorofova@avmorofova:~$ ls -a ~ | grep c*
conf.txt
```

Figure 2.3: Файлы, начинающиеся с "c"

5. Вывожу на экран (постранично) имена файлов из каталога /etc, начинающиеся с символа h (команда find /etc -maxdepth 1 -name "h*" | less) (рис. 2.4) (рис. 2.5)

```
avmorozova@avmorozova:~$ find /etc -maxdepth 1 -name "h*" | less
```

Figure 2.4: Файлы, начинающиеся с “h”

```
/etc/hosts  
/etc/hdparm.conf  
/etc/host.conf  
/etc/hosts.allow  
/etc/hosts.deny  
/etc/hp  
/etc/hostname  
/etc/hostid  
(END)
```

Figure 2.5: Список файлов

6. Запускаю в фоновом режиме процесс, который будет записывать в файл ~/logfile файлы, имена которых начинаются с log (команда `find / -name "log*" > logfile &`). Проверяю выполненные действия (команда `cat logfile`). (рис. 2.6)

```
avmorozova@avmorozova:~$ find / -name "log*" > logfile &
```

Figure 2.6: Запись файлов в фоновом процессе

7. Удаляю файл ~/logfile (команда `rm logfile`) (рис. 2.7)


```
avmorofova@avmorofova:~$ rm logfile
```

Figure 2.7: Удаление файла logfile

8. Запускаю редактор gedit в фоновом режиме (команда `gedit &`). На экране появляется окно редактора. (рис. 2.8) (рис. 2.9)

```
avmorofova@avmorofova:~$ gedit &
[1] 9650
```

Figure 2.8: Запуск редактора gedit

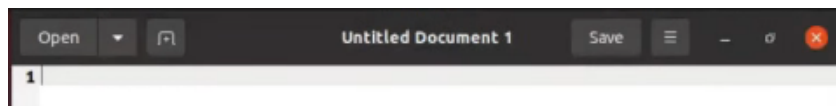


Figure 2.9: Редактор gedit

9. Определяю идентификатор процесса gedit (команда «`ps | grep -i "gedit"`»). Из рисунка видно, что наш процесс имеет PID 9650. Узнать идентификатор процесса можно также, используя команду `pgrep gedit` или `pidof gedit`. (рис. 2.10)

```
avmorofova@avmorofova:~$ gedit &
[1] 9650
avmorofova@avmorofova:~$ ps | grep -i "gedit"
  9650 pts/0    00:00:00 gedit
avmorofova@avmorofova:~$ pgrep gedit
9650
avmorofova@avmorofova:~$ pidof gedit
9650
```

Figure 2.10: Идентификатор процесса gedit

10. Читаю информацию о команде kill (команда `man kill`), использую её для завершения процесса gedit (команда `kill 9650`) (рис. 2.11)(рис. 2.12)

```
avmorofova@avmorofova:~$ man kill
avmorofova@avmorofova:~$ kill 9650
```

Figure 2.11: 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.
```

Figure 2.12: man kill

11. С помощью команд man df и man du узнаю информацию по необходимым командам и далее использую их:(рис. 2.13)

- df – утилита, показывающая список всех файловых систем по именам устройств, сообщает их размер, занятое и свободное пространство и точки монтирования. Синтаксис: df [опции] устройство (рис. 2.14)
- 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. Синтаксис: du [опции] каталог_ или_ файл (рис. 2.15)

```
avmorofova@avmorofova:~$ man df
[2]+  Terminated                  gedit
avmorofova@avmorofova:~$ man du
```

Figure 2.13: команды man df и man du

```

DF(1)                                User Commands                                DF(1)

NAME
    df - report file system disk space usage

SYNOPSIS
    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 con-

```

Figure 2.14: man df

```

DU(1)                                User Commands                                DU(1)

NAME
    du - estimate file space usage

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

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

```

Figure 2.15: man du

- Команда df (рис. 2.16)

```

avmorozova@avmorozova:~$ df
Filesystem      1K-blocks    Used Available Use% Mounted on
udev            1985628         0   1985628   0% /dev
tmpfs           403036       1380    401656   1% /run
/dev/sda5       19992176 11031152   7922432  59% /
tmpfs           2015172         0   2015172   0% /dev/shm
tmpfs           5120          4     5116   1% /run/lock
tmpfs           2015172         0   2015172   0% /sys/fs/cgroup
/dev/loop0       9344         9344         0 100% /snap/canonical-livepatch/98
/dev/loop2       56832        56832         0 100% /snap/core18/1988
/dev/loop1       101632       101632         0 100% /snap/core/10958
/dev/loop3       101376       101376         0 100% /snap/core/11081
/dev/loop4       56832        56832         0 100% /snap/core18/1997
/dev/loop5       224256       224256         0 100% /snap/gnome-3-34-1804/66
/dev/loop6       52352        52352         0 100% /snap/snap-store/518
/dev/loop7       33152        33152         0 100% /snap/snapd/11588
/dev/loop8       32896        32896         0 100% /snap/snapd/11841
/dev/loop9       66432        66432         0 100% /snap/gtk-common-themes/1514
/dev/loop10      66688        66688         0 100% /snap/gtk-common-themes/1515
/dev/sda1        523248         4     523244   1% /boot/efi
shared          125829116 78151580  47677536  63% /media/sf_shared
для_лаб         125829116 78151580  47677536  63% /media/sf_
tmpfs           403032         48    402984   1% /run/user/1000
/dev/sr0         75542        75542         0 100% /media/avmorozova/VBox_GAs_6.0
.24

```

Figure 2.16: Команда df

- Команда du (рис. 2.17)

```
avmorozova@avmorozova:~$ du
8      ./mozilla/firefox/4pin4k5w.default
4      ./mozilla/firefox/Crash Reports/events
12     ./mozilla/firefox/Crash Reports
4      ./mozilla/firefox/Pending Pings
1360   ./mozilla/firefox/psugcvua.default-release/gmp-gmpopenh264/1.8.1.1
1364   ./mozilla/firefox/psugcvua.default-release/gmp-gmpopenh264
4      ./mozilla/firefox/psugcvua.default-release/extensions
4628   ./mozilla/firefox/psugcvua.default-release/security.state
8      ./mozilla/firefox/psugcvua.default-release/bookmarkbackups
4      ./mozilla/firefox/psugcvua.default-release/saved-telemetry-pings
616    ./mozilla/firefox/psugcvua.default-release/sessionstore-backups
4      ./mozilla/firefox/psugcvua.default-release/minidumps
4      ./mozilla/firefox/psugcvua.default-release/crashes/events
12     ./mozilla/firefox/psugcvua.default-release/crashes
532    ./mozilla/firefox/psugcvua.default-release/datareporting/archived/2021
-05
156    ./mozilla/firefox/psugcvua.default-release/datareporting/archived/2021
-04
692    ./mozilla/firefox/psugcvua.default-release/datareporting/archived
4      ./mozilla/firefox/psugcvua.default-release/datareporting/glean/tmp
4      ./mozilla/firefox/psugcvua.default-release/datareporting/glean/events
4      ./mozilla/firefox/psugcvua.default-release/datareporting/glean/pending
_pings
8      ./mozilla/firefox/psugcvua.default-release/datareporting/glean/db
24     ./mozilla/firefox/psugcvua.default-release/datareporting/glean
```

Figure 2.17: Команда du

12. Вывожу имена всех директорий, имеющихя в домашнем каталоге (команда `find ~ -type d`), предварительно получив информацию с помощью команды `man find` (рис. 2.18) (рис. 2.19)

```
avmorozova@avmorozova:~$ find ~ -type d
/home/avmorozova
/home/avmorozova/.mozilla
/home/avmorozova/.mozilla/firefox
/home/avmorozova/.mozilla/firefox/4pin4k5w.default
/home/avmorozova/.mozilla/firefox/Crash Reports
/home/avmorozova/.mozilla/firefox/Crash Reports/events
/home/avmorozova/.mozilla/firefox/Pending Pings
/home/avmorozova/.mozilla/firefox/psugcvua.default-release
/home/avmorozova/.mozilla/firefox/psugcvua.default-release/gmp-
```

Figure 2.18: Команда `find ~ -type d`

```
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...] [ex-
    pression]

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.
```

Figure 2.19: Команда man find

13. Контрольные вопросы:
14. 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**.
15. “>” Перенаправление вывода в файл “>>” Перенаправление вывода в файл и открытие файла в режиме добавления (данные добавляются в конец файла).
16. Конвейер служит для объединения простых команд или утилит в цепочки, в которых результат работы предыдущей команды передаётся последующей. Синтаксис следующий: команда 1 | команда 2 (это означает, что вывод команды 1 передаётся на ввод команде 2).
17. Процесс рассматривается операционной системой как заявка на потребление всех видов ресурсов, кроме одного – процессорного времени. Этот последний важнейший ресурс распределяется операционной системой между другими единицами работы – потоками, которые и получили свое название благодаря тому, что они представляют собой последовательности (потоки

выполнения) команд. Процесс – это выполнение программы. Он считается активной сущностью и реализует действия, указанные в программе. Программа представляет собой статический набор команд, а процесс это набор ресурсов и данных, использующихся при выполнении программы.

18. `pid`: идентификатор процесса (PID) процесса (process ID), к которому вызывают метод `gid`: идентификатор группы UNIX, в котором работает программа.
19. Любую выполняющуюся в консоли команду или внешнюю программу можно запустить в фоновом режиме. Для этого следует в конце имени команды указать знак амперсанда `&`. Запущенные фоновые программы называются задачами. Ими можно управлять с помощью команды `jobs`, которая выводит список запущенных в данный момент задач.
20. `top` – это консольная программа, которая показывает список работающих процессов в системе. Программа в реальном времени отсортирует запущенные процессы по их нагрузке на процессор. `htop` – это продвинутый консольный мониторинг процессов. Утилита выводит постоянно меняющийся список системных процессов, который сортируется в зависимости от нагрузки на ЦПУ. Если делать сравнение с `top`, то `htop` показывает абсолютно все процессы в системе, время их непрерывного использования, загрузку процессоров и расход оперативной памяти.
21. `find` – это команда для поиска файлов и каталогов на основе специальных условий. Ее можно использовать в различных обстоятельствах, например, для поиска файлов по разрешениям, владельцам, группам, типу, размеру и другим подобным критериям. Команда `find` имеет такой синтаксис: `find [папка] [параметры] критерий шаблон [действие]` Папка – каталог в котором будем искать Параметры – дополнительные параметры, например, глубина поиска, и т.д. Критерий – по какому критерию будем искать: имя, дата создания, права, владелец и т.д. Шаблон – непосредственно значение по которому

будем отбирать файлы. Основные параметры: -P никогда не открывать символические ссылки -L - получает информацию о файлах по символическим ссылкам. Важно для дальнейшей обработки, чтобы обрабатывалась не ссылка, а сам файл. -maxdepth - максимальная глубина поиска по подкаталогам, для поиска только в текущем каталоге установите 1. -depth - искать сначала в текущем каталоге, а потом в подкаталогах -mount искать файлы только в этой файловой системе. -version - показать версию утилиты find -print - выводить полные имена файлов -type f - искать только файлы -type d - поиск папки в Linux Основные критерии: -name - поиск файлов по имени -perm - поиск файлов в Linux по режиму доступа -user - поиск файлов по владельцу -group - поиск по группе -mtime - поиск по времени модификации файла -atime - поиск файлов по дате последнего чтения -nogroup - поиск файлов, не принадлежащих ни одной группе -nouser - поиск файлов без владельцев -newer - найти файлы новее чем указанный -size - поиск файлов в Linux по их размеру Примеры: find ~ -type d поиск директорий в домашнем каталоге find ~ -type f -name ".*" поиск скрытых файлов в домашнем каталоге

22. Файл по его содержимому можно найти с помощью команды grep: «grep -r "слово/выражение, которое нужно найти"».
23. Утилита df, позволяет проанализировать свободное пространство на всех подключенных к системе разделах.
24. При выполнении команды du (без указания папки и опции) можно получить все файлы и папки текущей директории с их размерами. Для домашнего каталога: du ~/
25. Основные сигналы (каждый сигнал имеет свой номер), которые используются для завершения процесса:

SIGINT – самый безобидный сигнал завершения, означает Interrupt. Он отправляется процессу, запущенному из терминала с помощью сочетания клавиш Ctrl+C.

Процесс правильно завершает все свои действия и возвращает управление;

SIGQUIT – это еще один сигнал, который отправляется с помощью сочетания клавиш, программе, запущенной в терминале. Он сообщает ей что нужно завершиться и программа может выполнить корректное завершение или проигнорировать сигнал. В отличие от предыдущего, она генерирует дамп памяти. Сочетание клавиш Ctrl+Q;

SIGHUP – сообщает процессу, что соединение с управляющим терминалом разорвано, отправляется, в основном, системой при разрыве соединения с интернетом;

SIGTERM – немедленно завершает процесс, но обрабатывается программой, поэтому позволяет ей завершить дочерние процессы и освободить все ресурсы;

SIGKILL – тоже немедленно завершает процесс, но, в отличие от предыдущего варианта, он не передается самому процессу, а обрабатывается ядром. Поэтому ресурсы и дочерние процессы остаются запущенными.

Также для передачи сигналов процессам в Linux используется утилита kill, её синтаксис: kill [-сигнал] pid_процесса. Сигнал представляет собой один из выше перечисленных сигналов для завершения процесса. Перед тем, как выполнить остановку процесса, нужно определить его PID. Для этого используют команды ps и grep. Команда ps предназначена для вывода списка активных процессов в системе и информации о них. Команда grep запускается одновременно с ps (в канале) и будет выполнять поиск по результатам команды ps.

Утилита pkill – это оболочка для kill, она ведет себя точно так же, и имеет тот же синтаксис, только в качестве идентификатора процесса ей нужно передать его имя.

killall работает аналогично двум предыдущим утилитам. Она тоже принимает имя процесса в качестве параметра и ищет его PID в директории /proc. Но эта утилита обнаружит все процессы с таким именем и завершит их.

3 Выводы

В ходе выполнения лабораторной работы я приобрела практические навыки взаимодействия пользователя с системой посредством командной строки.