

Отчет по лабораторной работе №7

Дисциплина

Филиппова Анна Дмитриевна

Содержание

1	Цель работы	4
2	Выполнение лабораторной работы	5
3	Выводы	13
4	Контрольные вопросы	14

List of Figures

2.1	Запись в файл названий каталогов	5
2.2	Содержимое файла	6
2.3	Вывод файлов с определенным расширением и запись их в другой файл	6
2.4	Команды для поиска определенных файлов	7
2.5	Использование команды find	7
2.6	Использование команды find	7
2.7	Запуск процесса в фоновом режиме	8
2.8	Проверка действий	8
2.9	Удаление файла	8
2.10	Запуск gedit	9
2.11	Команда man	9
2.12	Справка о команде kill	9
2.13	Завершение процесса gedit	9
2.14	Команда man	10
2.15	Справка df	10
2.16	Справка du	11
2.17	Использовние команды du	11
2.18	Использовние команды df	11
2.19	Команда man	12
2.20	Справка о find	12
2.21	Выполнение команды	12

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 Выполнение лабораторной работы

1. Осуществляем вход в систему, используя свои логин и пароль.
2. Для того, чтобы записать в файл file.txt названия файлов, содержащихся в каталоге /etc, используем команду «ls -a /etc > file.txt». С помощью команды «ls -a ~ >> file.txt» дописываем в этот файл названия файлов, содержащихся в моем домашнем каталоге. Командой «cat file.txt» просматриваем файл, чтобы убедиться в правильности действий. (рис. 2.1) (рис. 2.2)

```
[adfilippova@adfilippova ~]$ ls -a /etc > file.txt
[adfilippova@adfilippova ~]$ ls -a ~ >> file.txt
[adfilippova@adfilippova ~]$ cat file.txt
.
..
abrt
adjtime
aliases
aliases.db
alsa
alternatives
anacrontab
asound.conf
at.deny
audisp
audit
avahi
bash_completion.d
bashrc
binfmt.d
bluetooth
brltty
brltty.conf
centos-release
centos-release-upstream
chkconfig.d
```

Figure 2.1: Запись в файл названий каталогов

```
.mozilla
my_os
new
os-intro1
pandoc-2.5
pandoc-2.5-linux.tar.gz
pandoc-2.9.2.1-1-amd64.deb
pandoc-2.9.2.1-1-amd64.deb.1
pandoc-2.9.2.1-1-amd64.deb.2
pandoc-crossref-Linux.tar.xz
.pki
play
reports
ski.places
.ssh
usr
Видео
Документы
Загрузки
Изображения
Музыка
Общедоступные
Рабочий стол
Шаблоны
[adfilippova@adfilippova ~]$ █
```

Figure 2.2: Содержимое файла

3. Выводим имена всех файлов из file.txt, имеющих расширение .conf и записываем их в новый текстовый файл conf.txt с помощью команды «grep -e ‘.conf\$’ file.txt > conf.txt». Командой «cat conf.txt» проверяем правильность выполненных действий. (рис. 2.3)

```
[adfilippova@adfilippova ~]$ grep -e '\.conf$' file.txt > conf.txt
[adfilippova@adfilippova ~]$ cat conf.txt
asound.conf
brltty.conf
chrony.conf
dleyna-server-service.conf
dnsmasq.conf
dracut.conf
e2fsck.conf
fprintd.conf
fuse.conf
GeoIP.conf
host.conf
idmapd.conf
ipsec.conf
kdump.conf
krb5.conf
ksmtuned.conf
ld.so.conf
libaudit.conf
libuser.conf
locale.conf
logrotate.conf
man db.conf
```

Figure 2.3: Вывод файлов с определенным расширением и запись их в другой файл

4. Определяем, какие файлы в моем домашнем каталоге имеют имена, начинающиеся с символа `c`, можно несколькими командами: «`ls ~/c*`», «`find ~ -maxdepth 1 -name "c*" -print`» (опция `maxdepth 1` необходима для того, чтобы файлы находились только в домашнем каталоге) и «`ls -a ~ | grep c*`». (рис. 2.4)

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

Figure 2.4: Команды для поиска определенных файлов

5. Воспользуемся командой «`find /etc -maxdepth 1 -name "h*" | less`», чтобы вывести на экран (по странично) имена файлов из каталога `/etc`, начинающиеся с символа `h`. (рис. 2.5) (рис. 2.6)

```
[adfilippova@adfilippova ~]$ find /etc -maxdepth 1 -name "h*" | less
```

Figure 2.5: Использование команды `find`

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

Figure 2.6: Использование команды `find`

6. Запускаем в фоновом режиме процесс, который будет записывать в файл `~/logfile` файлы, имена которых начинаются с `log`, используя команду «`find -name "log*" > logfile &`». Командой «`cat logfile`» проверяем выполненные действия. (рис. 2.7) (рис. 2.8)

```
[adfilippova@adfilippova ~]$ find / -name "log*" > logfile &
[1] 3292
find: '/boot/efi': Отказано в доступе
find: '/boot/grub2': Отказано в доступе
[adfilippova@adfilippova ~]$ 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
```

Figure 2.7: Запуск процесса в фоновом режиме

```
[adfilippova@adfilippova ~]$ cat logfile
/dev/log
/proc/sys/dev/scsi/logging_level
/proc/sys/net/ipv4/conf/all/log_martians
/proc/sys/net/ipv4/conf/default/log_martians
/proc/sys/net/ipv4/conf/enp0s3/log_martians
/proc/sys/net/ipv4/conf/lo/log_martians
/proc/sys/net/ipv4/conf/virbr0/log_martians
/proc/sys/net/ipv4/conf/virbr0-nic/log_martians
/proc/1/task/1/loginuid
/proc/1/loginuid
/proc/2/task/2/loginuid
/proc/2/loginuid
/proc/3/task/3/loginuid
/proc/3/loginuid
/proc/4/task/4/loginuid
/proc/4/loginuid
/proc/5/task/5/loginuid
/proc/5/loginuid
/proc/6/task/6/loginuid
/proc/6/loginuid
/proc/7/task/7/loginuid
/proc/7/loginuid
/proc/8/task/8/loginuid
/proc/8/loginuid
```

Figure 2.8: Проверка действий

7. Удаляем файл ~/logfile командой «rm logfile». (рис. 2.9)

```
[adfilippova@adfilippova ~]$ rm logfile
```

Figure 2.9: Удаление файла

8. Запускаем редактор gedit в фоновом режиме командой «gedit &», на экране появляется окно редактора. Чтобы определить идентификатор процесса

gedit, используем команду «ps | grep -i "gedit"». Процесс имеет PID 3683. Узнать идентификатор процесса можно также, используя команду «pgrep gedit» или «pidof gedit». (рис. 2.10)

```
[adfilippova@adfilippova ~]$ gedit &
[1] 3683
[adfilippova@adfilippova ~]$ ps | grep -i "gedit"
3683 pts/0    00:00:01 gedit
[adfilippova@adfilippova ~]$
```

Figure 2.10: Запуск gedit

9. см. пункт 8

10. Прочитав информацию о команде kill с помощью команды «man kill», используем её для завершения процесса gedit (команда «kill 3683»). (рис. 2.11) (рис. 2.12) (рис. 2.13)

```
[adfilippova@adfilippova ~]$ man kill
```

Figure 2.11: Команда man

```
KILL(1)                                User Commands                                KILL(1)

NAME
    kill - terminate a process

SYNOPSIS
    kill [-s signal|-p] [-q sigval] [-a] [--] pid...
    kill -l [signal]

DESCRIPTION
    The command kill sends the specified signal to the specified process or process group. If no signal is specified, the TERM signal is sent. The TERM signal will kill processes which do not catch this signal. For other processes, it may be necessary to use the KILL (9) signal, since this signal cannot be caught.

    Most modern shells have a builtin kill function, with a usage rather similar to that of the command described here. The '-a' and '-p' options, and the possibility to specify processes by command name are a local extension.

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

OPTIONS
    pid... Specify the list of processes that kill should signal. Each pid can
    Manual page kill(1) line 1 (press h for help or q to quit)
```

Figure 2.12: Справка о команде kill

```
[adfilippova@adfilippova ~]$ kill 3683
[1]+  Завершено      gedit
```

Figure 2.13: Завершение процесса gedit

11. С помощью команд «man df» и «man du» узнаем информацию по необходимым командам и далее использую их. df – утилита, показывающая список всех файловых систем по именам устройств, сообщает их размер, занятое и свободное пространство и точки монтирования. Синтаксис: df [опции] устройство 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. 2.14) (рис. 2.15) (рис. 2.16) (рис. 2.17) (рис. 2.18)

```
[adfilippova@adfilippova ~]$ man df
[adfilippova@adfilippova ~]$ man du
```

Figure 2.14: Команда man

```
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 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 can-
    not show the space available on unmounted file systems, because on most kinds
    of systems doing so requires very nonportable intimate knowledge of file sys-
    tem structures.

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

Figure 2.15: Справка 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 each FILE, recursively for directories.

    Mandatory arguments to long options are mandatory for short options too.

    -0, --null
        end each output line with 0 byte rather than newline

    -a, --all
        write counts for all files, not just directories

    --apparent-size
        print apparent sizes, rather than disk usage; although the apparent
        size is usually smaller, it may be larger due to holes in ('sparse')
        files, internal fragmentation, indirect blocks, and the like
Manual page du(1) line 1 (press h for help or q to quit)

```

Figure 2.16: Справка du

```

[adfilippova@adfilippova ~]$ df
Файловая система    1K-блоков  Использовано  Доступно  Использовано%  Смонтировано в
devtmpfs             490496      0            490496         0% /dev
tmpfs                507372      0            507372         0% /dev/shm
tmpfs                507372      776         499596         2% /run
tmpfs                507372      0            507372         0% /sys/fs/cgroup
/dev/mapper/centos-root 40825368    7746396    33078972        19% /
/dev/sda1            1038336    216704     821632         21% /boot
tmpfs                101476      36          101440         1% /run/user/1000
/dev/sr0              59588      59588        0             100% /run/media/adfili
ppova/VBox__GAs_6.1.20
/dev/sr1              4600876    4600876      0             100% /run/media/adfili
ppova/CentOS 7 x86_64

```

Figure 2.17: Использование команды du

```

[adfilippova@adfilippova ~]$ du
4      ./mozilla/extensions/{ec8030f7-c20a-464f-9b0e-13a3a9e97384}
4      ./mozilla/extensions
0      ./mozilla/plugins
50960  ./mozilla/firefox/czfg3j26.default-default/extensions
8      ./mozilla/firefox/czfg3j26.default-default/bookmarkbackups
0      ./mozilla/firefox/czfg3j26.default-default/storage/permanent/chrome/idb/387011
2724rsegmnoittet-es.files
0      ./mozilla/firefox/czfg3j26.default-default/storage/permanent/chrome/idb/145131
8868ntouromlalnody--epcr.files
0      ./mozilla/firefox/czfg3j26.default-default/storage/permanent/chrome/idb/165711
4595Amcateirvtisty.files
0      ./mozilla/firefox/czfg3j26.default-default/storage/permanent/chrome/idb/356128
8849sdhlie.files
0      ./mozilla/firefox/czfg3j26.default-default/storage/permanent/chrome/idb/282331
8777ntouromlalnody--naod.files
0      ./mozilla/firefox/czfg3j26.default-default/storage/permanent/chrome/idb/291806
3365piupsah.files
14584  ./mozilla/firefox/czfg3j26.default-default/storage/permanent/chrome/idb
14588  ./mozilla/firefox/czfg3j26.default-default/storage/permanent/chrome
14588  ./mozilla/firefox/czfg3j26.default-default/storage/permanent
0      ./mozilla/firefox/czfg3j26.default-default/storage/temporary
0      ./mozilla/firefox/czfg3j26.default-default/storage/default/moz-extension+++5f3
0b7f2-ab88-4d19-b51b-34acec35cbc5^userContextId=4294967295/idb/3647222921wleabcEoxlt-ee
nqsairo.files

```

Figure 2.18: Использование команды df

12. Выводим имена всех директорий, имеющихя в моем домашнем каталоге с помощью команды «find ~ -type d», предварительно получив информацию

с помощью команды «man find». (рис. 2.19) (рис. 2.20) (рис. 2.21)

```
[adfilippova@adfilippova ~]$ man find
```

Figure 2.19: Команда man

```
FIND(1)                                General Commands Manual                                FIND(1)

NAME
  find - search for files in a directory hierarchy

SYNOPSIS
  find [-H] [-L] [-P] [-D debugopts] [-Olevel] [path...] [expression]

DESCRIPTION
  This manual page documents the GNU version of find. GNU find searches the
  directory tree rooted at each given file name by evaluating the given expres-
  sion 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 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 findu-
  tils 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-
  Manual page find(1) line 1 (press h for help or q to quit)
```

Figure 2.20: Справка о find

```
[adfilippova@adfilippova ~]$ find ~ -type d
/home/adfilippova
/home/adfilippova/.mozilla
/home/adfilippova/.mozilla/extensions
/home/adfilippova/.mozilla/extensions/{ec8030f7-c20a-464f-9b0e-13a3a9e97384}
/home/adfilippova/.mozilla/plugins
/home/adfilippova/.mozilla/firefox
/home/adfilippova/.mozilla/firefox/czfg3j26.default-default
/home/adfilippova/.mozilla/firefox/czfg3j26.default-default/extensions
/home/adfilippova/.mozilla/firefox/czfg3j26.default-default/bookmarkbackups
/home/adfilippova/.mozilla/firefox/czfg3j26.default-default/storage
/home/adfilippova/.mozilla/firefox/czfg3j26.default-default/storage/permanent
/home/adfilippova/.mozilla/firefox/czfg3j26.default-default/storage/permanent/chrome
/home/adfilippova/.mozilla/firefox/czfg3j26.default-default/storage/permanent/chrome/id
b
/home/adfilippova/.mozilla/firefox/czfg3j26.default-default/storage/permanent/chrome/id
b/3870112724rsegmnoittet-es.files
/home/adfilippova/.mozilla/firefox/czfg3j26.default-default/storage/permanent/chrome/id
b/1451318868ntouromlalnodry--epcr.files
/home/adfilippova/.mozilla/firefox/czfg3j26.default-default/storage/permanent/chrome/id
b/1657114595AmcateirvtiSty.files
/home/adfilippova/.mozilla/firefox/czfg3j26.default-default/storage/permanent/chrome/id
b/3561288849sdhlie.files
/home/adfilippova/.mozilla/firefox/czfg3j26.default-default/storage/permanent/chrome/id
```

Figure 2.21: Выполнение команды

3 Выводы

Я изучила инструменты поиска файлов и фильтрации текстовых данных, а также приобрела практические навыки: по управлению процессами (и заданиями), по проверке использования диска и обслуживанию файловых систем.

4 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. Большинство используемых в консоли команд и программ записывают результаты своей работы в стандартный поток вывода `stdout`.
2. “>” Перенаправление вывода в файл “>>” Перенаправление вывода в файл и открытие файла в режиме добавления (данные добавляются в конец файла).
3. 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).
4. 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.

5. `pid`: идентификатор процесса (PID) процесса (process ID), к которому вызывают метод `gid`: идентификатор группы UNIX, в котором работает программа.
6. Любую выполняющуюся в консоли команду или внешнюю программу можно запустить в фоновом режиме. Для этого следует в конце имени команды указать знак амперсанда `&`. Запущенные фоном программы называются задачами. Ими можно управлять с помощью команды `jobs`, которая выводит список запущенных в данный момент задач.
7. `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.
8. `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

9. Файл по его содержимому можно найти с помощью команды grep: «grep -r "слово/выражение, которое нужно найти"».
10. Утилита df, позволяет проанализировать свободное пространство на всех подключенных к системе разделах.
11. При выполнении команды du (без указания папки и опции) можно получить все файлы и папки текущей директории с их размерами. Для домашнего каталога: du ~/
12. 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+;

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.