

Отчёт по лабораторной работе №6

Операционные системы

Федоров Андрей Андреевич

Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
3	Вопросы для самопроверки	11
4	Выводы	14
	Список литературы	15

Список иллюстраций

2.1	консоль	6
2.2	консоль (file.txt)	7
2.3	консоль	8
2.4	kill	8
2.5	du	9
2.6	df	9
2.7	find	10
2.8	консоль	10

Список таблиц

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. Я осуществил вход в систему, используя соответствующее имя пользователя. Записал в файл file.txt названия файлов, содержащихся в каталоге /etc. Дописал в этот же файл названия файлов, содержащихся в моём домашнем каталоге



```
aafedorov@dk5n51 ~ $ ls /etc > file.txt
aafedorov@dk5n51 ~ $ ls ~ >>file.txt
```

Рис. 2.1: консоль

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

```
aafedorov@dk5n51 ~ $ grep .conf file.txt
appstream.conf
brltty.conf
ca-certificates.conf
cachefilesd.conf
cfg-update.conf
dconf
dhcpcd.conf
dispatch-conf.conf
dleyna-server-service.conf
dnsmasq.conf
e2fsck.conf
e2scrub.conf
etc-update.conf
fluidsynth.conf
fuse.conf
gai.conf
gconf
genkernel.conf
gssapi_mech.conf
host.conf
idmapd.conf
idn2.conf
idn2.conf.sample
idnalias.conf
idnalias.conf.sample
java-config-2
krb5.conf
krb5.conf.example
ldap.conf
ldap.conf.sudo
ld.so.conf
ld.so.conf.d
libaudit.conf
lightdm.conf
locale.conf
logrotate.conf
mailutils.conf
make.conf
man.conf
man_db.conf
metalog.conf
mke2fs.conf
mlocate-cron.conf
modprobe.conf.1100
modprobe.conf.old
modules.conf
```

Рис. 2.2: консоль (file.txt)

3. Определил, какие файлы в вашем домашнем каталоге имеют имена, начинающиеся с символа с? Предложил несколько вариантов, как это сделать. Вывел на экран (по странично) имена файлов из каталога /etc, начинающиеся с символа h. Запустил в фоновом режиме процесс, который будет

записывать в файл ~/logfile файлы, имена которых начинаются с log. Удалил файл ~/logfile

```
kill -r-- 1 aafedorov studsci 1191 мар 16 12:57 conf.txt
drwxr-xr-x 2 aafedorov studsci 2048 мар 16 12:14 cpp
-rw-r--r-- 1 aafedorov studsci 3957 мар 16 12:49 file.txt
drwxr-xr-x 3 aafedorov studsci 2048 дек 9 15:13 GNUstep
-rw----- 1 aafedorov studsci 2622 окт 13 15:57 lab02
drwxr-xr-x 3 aafedorov root 2048 фев 4 18:37 public
lrwxr-xr-x 1 aafedorov root 18 мар 2 21:03 public_html -> public/public_html
drwxr-xr-x 2 aafedorov studsci 2048 дек 9 15:12 tmp
drwxr-xr-x 2 aafedorov studsci 2048 фев 16 15:47 video
drwxr-xr-x 5 aafedorov studsci 2048 мар 16 12:28 work
drwxr-xr-x 2 aafedorov studsci 2048 сен 14 2022 Видео
drwxr-xr-x 2 aafedorov studsci 2048 фев 16 13:15 Документы
drwxr-xr-x 3 aafedorov studsci 2048 мар 16 12:24 Загрузки
drwxr-xr-x 3 aafedorov studsci 2048 сен 28 13:06 Изображения
drwxr-xr-x 2 aafedorov studsci 2048 сен 14 2022 Музыка
drwxr-xr-x 2 aafedorov studsci 2048 сен 14 2022 Общедоступные
drwxr-xr-x 2 aafedorov studsci 2048 дек 21 14:30 Рабочий стол
drwxr-xr-x 2 aafedorov studsci 2048 сен 14 2022 Шаблоны
```

Рис. 2.3: консоль

4. Посмотрел команды kill du df find с помощью команды man

```
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.
    -s <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.
    -q, --queue <value>
        Use sigqueue(3) rather than kill(2) and the value argument is used to specify an integer to be sent with the signal. If the receiving process has installed a handler for this signal using the SA_SIGINFO flag to sigaction(2), then it can obtain this data via the si_value field of the siginfo_t structure.
    -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 5431
        Send the default signal, SIGTERM, to all those processes.
SEE ALSO
    kill(2), killall(1), nice(1), pkill(1), renice(1), signal(7), sigqueue(3), skill(1)
STANDARDS
    Manual page kill(1) line 1/59 82% (press h for help or q to quit)
```

Рис. 2.4: kill


```
df(1) User Commands df(1)
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 name argument. If no file name is given, the space available on all currently mounted file systems 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 used.
  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 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 nonportable 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.
  -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 units of 1,048,576 bytes; see SIZE format below
  -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 if FIELD_LIST is omitted.
  -P, --portability
    use the POSIX output format
  --sync
    invoke sync before getting usage info
Manual page df(1) line 1/94 528 (press h for help or q to quit)
```

Рис. 2.5: du

```
du(1) User Commands du(1)
NAME
  du - estimate file space usage
SYNOPSIS
  du [OPTION]... [FILE]...
  du [OPTION]... --files-from=F
DESCRIPTION
  Summarize device usage of the set of FILES, recursively for directories.
  Mandatory arguments to long options are mandatory for short options too.
  -a, --all
    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 device 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
  -B, --block-size=SIZE
    scale sizes by SIZE before printing them; e.g., "-BM" 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-depth=0 is the same as --summarize
  --files-from=F
    summarize device usage of the NUL-terminated file names specified in file F; if F is -, then read names from standard input
  -H, --dereference-args
    equivalent to --dereference-args (-D)
  -h, --human-readable
    print sizes in human readable format (e.g., 1K 234M 2G)
  --inodes
    list inode usage information instead of block usage
  -k, --block-size=K
    like --block-size=1K
Manual page du(1) line 1/125 408 (press h for help or q to quit)
```

Рис. 2.6: df

3 Вопросы для самопроверки

1. Какие потоки ввода вывода вы знаете?
2. Объясните разницу между операцией > и ».
3. Что такое конвейер?
4. Что такое процесс? Чем это понятие отличается от программы?
5. Что такое PID и GID?
6. Что такое задачи и какая команда позволяет ими управлять?
7. Найдите информацию об утилитах top и htop. Каковы их функции?
8. Назовите и дайте характеристику команде поиска файлов. Приведите примеры использования этой команды.
9. Можно ли по контексту (содержанию) найти файл? Если да, то как?
10. Как определить объем свободной памяти на жёстком диске?
11. Как определить объем вашего домашнего каталога?
12. Как удалить зависший процесс?
13. 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.

14. • 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.
15. Конвейер (pipe) служит для объединения простых команд или утилит в цепочки, в которых результат работы предыдущей команды передаётся последующей.
16. 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.
17. PID — уникальный номер (идентификатор) процесса в многозадачной ОС. GID – идентификатор группы. 18
18. 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.
19. 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

13 часто применяется в тех случаях, когда информации даваемой утилитой `top` недостаточно, например при поиске утечек памяти в процессах.

20. 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 `i` и удалить их: `find ~ -name "*i" -exec rm {} ;`
21. 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* 19` • найти в текущем каталоге все файлы, в имени которых есть буквосочетание «лаб»: `ls -l | grep лаб`
22. Определить объем свободной памяти на жёстком диске позволяет команда `df`.
23. Определить объем домашнего каталога позволяет команда `df /home/`
24. Удалить зависший процесс можно командой `kill %номер задачи`.

4 Выводы

Я познакомился с инструментами поиска файлов и фильтрации текстовых данных и приобрёл практических навыков: по управлению процессами (и заданиями), по проверке использования диска и обслуживанию файловых систем.

Список литературы