

Отчёт по лабораторной работе №6

Поиск файлов. Перенаправление ввода-вывода. Просмотр запущенных процессов

Грузинова Елизавета Константиновна; НКНбд-02-21

Содержание

1	Цель работы	4
2	Задание	5
3	Теоретическое введение	7
4	Выполнение лабораторной работы	9
5	Выводы	15
6	Контрольные вопросы	16
	Список литературы	18

Список иллюстраций

4.1	Запись файлов из /etc в file.txt	9
4.2	Просмотр содержимого файла с помощью команды cat	9
4.3	Запись файлов из домашнего каталога в file.txt	9
4.4	Выбор файлов с расширением .conf	10
4.5	Запись и просмотр файлов из file.txt с расширением .conf в conf.txt	10
4.6	Первый способ нахождения файлов, начинавшихся с символа с . .	10
4.7	Второй способ нахождения файлов, начинавшихся с символа с . .	10
4.8	Файлы, начинающиеся с символа h	11
4.9	Фоновое выполнение процесса, заданного командой	11
4.10	Удаление с помощью команды rm	11
4.11	Фоновый запуск gedit	11
4.12	Первый способ определение идентификатора процесса (через ps, конвейер и фильтр grep)	11
4.13	Второй способ определение идентификатора процесса (через фильтр rgrep)	12
4.14	Вызов команды man kill	12
4.15	Информация о команде kill	12
4.16	Завершение процесса gedit	12
4.17	Вызов команды man df	13
4.18	Информация о команде df	13
4.19	Определение занятости сегментов диска с помощью команды df -vi	13
4.20	Информация о команде du	13
4.21	Определение размера файлов в ~/Загрузки	14
4.22	Информация о поиске командой find всех директорий в домашнем каталоге	14
4.23	Использование команды find -type d	14

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 Задание

1. Запишите в файл `file.txt` названия файлов, содержащихся в каталоге `/etc`. Допишите в этот же файл названия файлов, содержащихся в вашем домашнем каталоге.
2. Выведите имена всех файлов из `file.txt`, имеющих расширение `.conf`, после чего запишите их в новый текстовый файл `conf.txt`.
3. Определите, какие файлы в вашем домашнем каталоге имеют имена, начинавшиеся с символа `c`? Предложите несколько вариантов, как это сделать.
4. Выведите на экран (по странично) имена файлов из каталога `/etc`, начинающиеся с символа `h`.
5. Запустите в фоновом режиме процесс, который будет записывать в файл `~/logfile` файлы, имена которых начинаются с `log`.
6. Удалите файл `~/logfile`.
7. Запустите из консоли в фоновом режиме редактор `gedit`.
8. Определите идентификатор процесса `gedit`, используя команду `ps`, конвейер и фильтр `grep`. Как ещё можно определить идентификатор процесса?
9. Прочтите справку (`man`) команды `kill`, после чего используйте её для завершения процесса `gedit`.
10. Выполните команды `df` и `du`, предварительно получив более подробную информацию об этих командах, с помощью команды `man`.

11. Воспользовавшись справкой команды `find`, выведите имена всех директо-
рий, имеющихсЯ в вашем домашнем каталоге.

3 Теоретическое введение

Для того, чтобы записать данные в файл или прочитать их оттуда, процессу необходимо сначала открыть этот файл (при открытии на запись, возможно, придётся предварительно создать его). При этом процесс получает дескриптор (описатель) открытого файла — уникальное для этого процесса число, которое он и будет использовать во всех операциях записи. Первый открытый файл получит дескриптор 0, второй — 1 и так далее. Закончив работу с файлом, процесс закрывает его, при этом дескриптор освобождается и может быть использован повторно. Если процесс завершается, не закрыв файлы, за него это делает система. Строго говоря, только в операции открытия дескриптора указывается, какой именно файл будет использоваться. В качестве «файла» используются и обычные файлы, и файлы-дырки (чаще всего — терминалы), и каналы, описанные в разделе Конвейер. Дальнейшие операции — чтение, запись и закрытие, работают с дескриптором, как с потоком данных, а куда именно ведёт этот поток, неважно.

Каждый процесс Linux получает при старте три «файла», открытых для него системой. Первый из них (дескриптор 0) открыт на чтение, это стандартный ввод процесса. Именно со стандартным вводом работают все операции чтения, если в них не указан дескриптор файла. Второй (дескриптор 1) — открыт на запись, это стандартный вывод процесса. С ним работают все операции записи, если дескриптор файла не указан в них явно. Наконец, третий поток данных (дескриптор 2) предназначается для вывода диагностических сообщений, он называется стандартный вывод ошибок. Поскольку эти три дескриптора уже открыты к моменту запуска процесса, первый файл, открытый самим процессом,

будет, скорее всего, иметь дескриптор 3.[1]

4 Выполнение лабораторной работы

1. Запишите в файл file.txt названия файлов, содержащихся в каталоге /etc. Допишите в этот же файл названия файлов, содержащихся в вашем домашнем каталоге. (рис. 4.1, 4.2, 4.3)

```
[ekgruzinova@fedora ~]$ ls -lR /etc > file.txt
ls: невозможно открыть каталог '/etc/audit': Отказано в доступе
ls: невозможно открыть каталог '/etc/cups/ssl': Отказано в доступе
ls: невозможно открыть каталог '/etc/dhcp': Отказано в доступе
```

Рис. 4.1: Запись файлов из /etc в file.txt

```
[ekgruzinova@fedora ~]$ cat file.txt
/etc:
итого 1528
drwxr-xr-x. 1 root root      126 окт 26  2021 abrt
-rw-r--r--. 1 root root       16 апр 30  15:19 adjtime
-rw-r--r--. 1 root root    1529 июл 16  2021 aliases
drwxr-xr-x. 1 root root       70 дек  6  21:03 alsa
```

Рис. 4.2: Просмотр содержимого файла с помощью команды cat

```
[ekgruzinova@fedora ~]$ ls -lR ~/ >> file.txt
ls: невозможно получить доступ к '/home/ekgruzinova/work/a_programms': Отказано в доступе
ls: невозможно получить доступ к '/home/ekgruzinova/work/a_programms.zip': Отказано в доступе
ls: невозможно получить доступ к '/home/ekgruzinova/work/australia': Отказано в доступе
```

Рис. 4.3: Запись файлов из домашнего каталога в file.txt

2. Выведите имена всех файлов из file.txt, имеющих расширение .conf, после чего запишите их в новый текстовый файл conf.txt. (рис. 4.4, 4.5)

```
[ekgruzinova@fedora ~]$ grep .conf file.txt
-rw-r--r--. 1 root root      269 дек 24 16:29 anthy-unicode.conf
-rw-r--r--. 1 root root      769 окт  8 2021 appstream.conf
-rw-r--r--. 1 root root       55 дек 17 21:18 asound.conf
-rw-r--r--. 1 root root    28974 июл 21 2021 brltty.conf
drwxr-xr-x. 1 root root       0 июл 23 2021 chkconfig.d
-rw-r--r--. 1 root root     1371 дек 16 15:17 chrony.conf
drwxr-xr-x. 1 root root       18 авг  4 2021 dconf
-rw-r--r--. 1 root root     1280 авг 20 2021 dleyna-renderer-service.conf
```

Рис. 4.4: Выбор файлов с расширением .conf

```
[ekgruzinova@fedora ~]$ grep '\.conf' file.txt > conf.txt
[ekgruzinova@fedora ~]$ cat conf.txt
-rw-r--r--. 1 root root      269 дек 24 16:29 anthy-unicode.conf
-rw-r--r--. 1 root root      769 окт  8 2021 appstream.conf
-rw-r--r--. 1 root root       55 дек 17 21:18 asound.conf
-rw-r--r--. 1 root root    28974 июл 21 2021 brltty.conf
-rw-r--r--. 1 root root     1371 дек 16 15:17 chrony.conf
-rw-r--r--. 1 root root     1280 авг 20 2021 dleyna-renderer-service.conf
-rw-r--r--. 1 root root     1174 авг 20 2021 dleyna-server-service.conf
```

Рис. 4.5: Запись и просмотр файлов из file.txt с расширением .conf в conf.txt

3. Определите, какие файлы в вашем домашнем каталоге имеют имена, начинавшиеся с символа с? Предложите несколько вариантов, как это сделать. (рис. 4.6, 4.7)

```
[ekgruzinova@fedora ~]$ find ~ -name "с*" -print
/home/ekgruzinova/.mozilla/firefox/cu8u2w1d.default-release
/home/ekgruzinova/.mozilla/firefox/cu8u2w1d.default-release/crashes
/home/ekgruzinova/.mozilla/firefox/cu8u2w1d.default-release/compatibility.ini
/home/ekgruzinova/.mozilla/firefox/cu8u2w1d.default-release/cookies.sqlite
/home/ekgruzinova/.mozilla/firefox/cu8u2w1d.default-release/cert9.db
/home/ekgruzinova/.mozilla/firefox/cu8u2w1d.default-release/storage/permanent/chrome
```

Рис. 4.6: Первый способ нахождения файлов, начинавшихся с символа с

```
[ekgruzinova@fedora ~]$ ls -l | grep с*
-rw-rw-r--. 1 ekgruzinova ekgruzinova 39794 мая  7 14:23 conf.txt
```

Рис. 4.7: Второй способ нахождения файлов, начинавшихся с символа с

4. Выведите на экран (по странично) имена файлов из каталога /etc, начинающиеся с символа h. (рис. 4.8)

```
[ekgruzinova@fedora ~]$ find /etc -name "h*" -print
find: '/etc/audit': Отказано в доступе
/etc/avahi/hosts
/etc/brltty/Contraction/ha.ctb
/etc/brltty/Input/bm/horizontal.kti
```

Рис. 4.8: Файлы, начинающиеся с символа h

5. Запустите в фоновом режиме процесс, который будет записывать в файл ~/logfile файлы, имена которых начинаются с log. (рис. 4.9)

```
[ekgruzinova@fedora ~]$ find ~ -name "log*" -print > ~/logfile &
[1] 5655
```

Рис. 4.9: Фоновое выполнение процесса, заданного командой

6. Удалите файл ~/logfile. (рис. 4.10)

```
[ekgruzinova@fedora ~]$ rm ~/logfile
[1]+  Завершён      find ~ -name "log*" -print > ~/logfile
```

Рис. 4.10: Удаление с помощью команды rm

7. Запустите из консоли в фоновом режиме редактор gedit. (рис. 4.11)

```
[ekgruzinova@fedora ~]$ gedit &
[1] 5676
```

Рис. 4.11: Фоновый запуск gedit

8. Определите идентификатор процесса gedit, используя команду ps, конвейер и фильтр grep. Как ещё можно определить идентификатор процесса? (рис. 4.12, 4.13)

```
[ekgruzinova@fedora ~]$ ps aux | grep gedit
ekgruzi+  5676  1.7  3.9 859540 79052 pts/0    Sl  14:33   0:01 gedit
ekgruzi+  5706  0.0  0.1 221692  2312 pts/0    S+  14:34   0:00 grep --color=auto gedit
```

Рис. 4.12: Первый способ определение индетификатора процесса (через ps, конвейер и фильтр grep)

```
[ekgruzinova@fedora ~]$ pgrep gedit
5676
```

Рис. 4.13: Второй способ определение идентификатора процесса (через фильтр pgrep)

9. Прочтите справку (man) команды kill, после чего используйте её для завершения процесса gedit. (рис. 4.14, 4.15, 4.16)

```
[ekgruzinova@fedora ~]$ man kill
```

Рис. 4.14: Вызов команды man kill

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

SYNOPSIS
    kill [-signal|-s signal|-p] [-q value] [-a] [--timeout milliseconds signal] [--] pid|name...

    kill -l [number] | -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 signal (number 9),
    since a process may install a handler for the TERM signal in order 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 the KILL signal may be used; be aware that the latter 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 builtin kill command, with a usage rather similar to that of the
    command described here. The --all, --pid, and --queue options, and the possibility to specify
    processes by command name, are local extensions.

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

ARGUMENTS
```

Рис. 4.15: Информация о команде kill

```
[ekgruzinova@fedora ~]$ kill 5676
[ekgruzinova@fedora ~]$ ps aux | grep gedit
ekgruzi+  5676  0.0  0.1 221692  2256 pts/0    S+   14:36   0:00 grep --color=auto gedit
[1]+  Завершено      gedit
```

Рис. 4.16: Завершение процесса gedit

10. Выполните команды df и du, предварительно получив более подробную информацию об этих командах, с помощью команды man. (рис. 4.17, 4.18, 4.19, 4.20, 4.21)

```
[ekgruzinova@fedora ~]$ man df
```

Рис. 4.17: Вызов команды man df

```
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 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.
```

Рис. 4.18: Информация о команде df

```
[ekgruzinova@fedora ~]$ df -vi
Файловая система  Инодов  ИИспользовано  ИСвободно  ИИспользовано%  Смонтировано в
devtmpfs          131072         469      130603          1% /dev
tmpfs              252519          6      252513          1% /dev/shm
tmpfs              819200        988      818212          1% /run
/dev/sda2           0             0           0           - /
tmpfs             1048576         49     1048527          1% /tmp
/dev/sda2           0             0           0           - /home
/dev/sda1           65536         456      65080          1% /boot
tmpfs              50503         144      50359          1% /run/user/1000
```

Рис. 4.19: Определение занятости сегментов диска с помощью команды df -vi

```
[ekgruzinova@fedora ~]$ man du
[ekgruzinova@fedora ~]$ du -a ~/Загрузки
17848 /home/ekgruzinova/Загрузки/hugo_extended_0.98.0_Linux-64bit.tar.gz
12 /home/ekgruzinova/Загрузки/hugo_extended_0.98.0_Linux-64bit/LICENSE
12 /home/ekgruzinova/Загрузки/hugo_extended_0.98.0_Linux-64bit/README.md
24 /home/ekgruzinova/Загрузки/hugo_extended_0.98.0_Linux-64bit
17848 /home/ekgruzinova/Загрузки/hugo_extended_0.98.0_Linux-64bit(1).tar.gz
12 /home/ekgruzinova/Загрузки/hugo_extended_0.98.0_Linux-64bit(1)/LICENSE
12 /home/ekgruzinova/Загрузки/hugo_extended_0.98.0_Linux-64bit(1)/README.md
54940 /home/ekgruzinova/Загрузки/hugo_extended_0.98.0_Linux-64bit(1)/hugo
54964 /home/ekgruzinova/Загрузки/hugo_extended_0.98.0_Linux-64bit(1)
```

Рис. 4.20: Информация о команде du

```
DU(1)                                User Commands

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
        write counts for all files, not just directories
```

Рис. 4.21: Определение размера файлов в ~/Загрузки

11. Воспользовавшись справкой команды `find`, выведите имена всех директо-
рий, имеющиххся в вашем домашнем каталоге. (рис. 4.22, 4.23)

```
-type c
File is of type c:

b      block (buffered) special
c      character (unbuffered) special
d      directory
```

Рис. 4.22: Информация о поиске командой `find` всех директорий в домашнем каталоге

```
[ekgruzinova@fedora ~]$ find -type d
.
./.mozilla
./.mozilla/extensions
./.mozilla/extensions/{ec8030f7-c20a-464f-9b0e-13a3a9e97384}
./.mozilla/plugins
./.mozilla/firefox
./.mozilla/firefox/Crash Reports
./.mozilla/firefox/Crash Reports/events
./.mozilla/firefox/Pending Pings
./.mozilla/firefox/cu8u2w1d.default-release
./.mozilla/firefox/cu8u2w1d.default-release/minidumps
```

Рис. 4.23: Использование команды `find -type d`

5 Выводы

При выполнении лабораторной работы я ознакомилась с инструментами поиска файлов и фильтрации текстовых данных, приобрела практические навыки: по управлению процессами (и заданиями), по проверке использования диска и обслуживанию файловых систем.

6 Контрольные вопросы

1. `stdin`, `stdout`, `stderr`
2. При “»” файл также открывается в режиме добавления, а при “>” - нет.
3. Объединение простых команд или утилит в цепочки, в которых результат работы предыдущей команды передаётся в следующую.
4. Компьютерная программа сама по себе — это только пассивная совокупность инструкций, в то время как процесс — это непосредственное выполнение этих инструкций.
5. `gid` - идентификатор группы UNIX, в котором работает программа.

При запуске процесса ему присваивается уникальный номер, называемый `process ID (PID)`, который идентифицирует этот процесс для системы.

6. Запущенные из консоли с помощью амперсанда команды, работают в фоновом режиме, называются задачами. С помощью команды `top` можно управлять ими.
7. `top` - интерактивный просмотрщик процессов. `Htop` – основанный на `ncurses` просмотрщик процессов подобный `top`, `htop`, `atop` интерактивные просмотрщики процессов, но позволяющий прокручивать список процессов вертикально и горизонтально, чтобы видеть их полные параметры запуска. Управление процессами (остановка, изменение приоритета) может выполняться без ручного ввода их идентификаторов.

8. Команда `find` представляет собой универсальный инструмент поиска: она позволяет искать файлы и каталоги, просматривать все каталоги в системе или только текущий каталог.

Команда

```
find ~ -name "t*" -print
```

выведет на экран имена всех файлов из домашнего каталога, начинающиеся с символа `t`.

9. Да, возможно. Для этого необходимо использовать команду

```
grep -rnw '/ПУТЬ/ДО/ПАПКИ/' -e 'ШАБЛОН'
```

10. С помощью команды

```
df -h (имя диска)
```

11. С помощью команды

```
du ~/
```

12. С помощью команды `kill%номер процесса`

Список литературы

1. Перенаправление ввода и вывода [Электронный ресурс]. URL: <http://heaper.althlinux.org/kirill/LinuxIntro/Pipeline.html>.