

Отчёт по лабораторной работе №6

Операционные системы

Екатерина Павловна Канева

Содержание

1	Цель работы	5
2	Задание	6
3	Теоретическое введение	7
4	Выполнение лабораторной работы	8
5	Контрольные вопросы	14
6	Выводы	17

Список иллюстраций

4.1	Запись файлов домашнего каталога и каталога <code>/etc</code>	8
4.2	Вывод файлов с расширением <code>.conf</code>	9
4.3	Запись названий нужных файлов в файл <code>conf.txt</code>	9
4.4	Способ 1.	9
4.5	Способ 2.	9
4.6	Файлы, начинающиеся с <code>h</code>	10
4.7	Запись имён нужных файлов в фоновом режиме.	10
4.8	Удаление файла <code>logfile</code>	10
4.9	Запуск <code>gedit</code> в фоновом режиме.	10
4.10	Мануал по команде <code>kill</code>	11
4.11	Завершение процесса <code>gedit</code>	11
4.12	Мануал по команде <code>df</code>	11
4.13	Мануал по команде <code>du</code>	12
4.14	Команда <code>df</code>	12
4.15	Команда <code>du</code>	12
4.16	Мануал по команде <code>find</code>	12
4.17	Вывод имён директорий.	13

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 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, начинающиеся с символа `s`? Предложите несколько вариантов, как это сделать.
4. Выведите на экран 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 директорий, имеющих в вашем домашнем каталоге.

3 Теоретическое введение

В интерфейсе командной строки есть очень полезная возможность перенаправления (переадресации) ввода и вывода (англ. термин I/O Redirection). Как мы уже заметили, многие программы выводят данные на экран. А ввод данных в терминале осуществляется с клавиатуры. С помощью специальных обозначений можно перенаправить вывод многих команд в файлы или иные устройства вывода (например, распечатать на принтере). Тоже самое и со вводом информации, вместо ввода данных с клавиатуры, для многих программ можно задать считывание символов их файла. Кроме того, можно даже вывод одной программы передать на ввод другой программе.

К каждой программе, запускаемой в командной строке, по умолчанию подключено три потока данных:

STDIN (0) — стандартный поток ввода (данные, загружаемые в программу). STDOUT (1) — стандартный поток вывода (данные, которые выводит программа). По умолчанию — терминал. STDERR (2) — стандартный поток вывода диагностических и отладочных сообщений (например, сообщениях об ошибках). По умолчанию — терминал.

Pipe (конвейер) – это однонаправленный канал межпроцессного взаимодействия. Термин был придуман Дугласом Макилроем для командной оболочки Unix и назван по аналогии с трубопроводом. Конвейеры чаще всего используются в shell-скриптах для связи нескольких команд путем перенаправления вывода одной команды (stdout) на вход (stdin) последующей, используя символ конвейера '|'.

4 Выполнение лабораторной работы

- Записали в файл `file.txt` названия файлов, содержащихся в каталоге `/etc`, дописали в этот же файл названия файлов, содержащихся в вашем домашнем каталоге (рис. 4.1):

```
[epkaneva@epkaneva ~]$ ls -R /etc > file.txt
ls: cannot open directory '/etc/audit': Permission denied
ls: cannot open directory '/etc/cups/ssl': Permission denied
ls: cannot open directory '/etc/dhcp': Permission denied
ls: cannot open directory '/etc/firewalld': Permission denied
ls: cannot open directory '/etc/grub.d': Permission denied
ls: cannot open directory '/etc/libvirt': Permission denied
ls: cannot open directory '/etc/lvm/archive': Permission denied
ls: cannot open directory '/etc/lvm/backup': Permission denied
ls: cannot open directory '/etc/lvm/cache': Permission denied
ls: cannot open directory '/etc/nftables': Permission denied
ls: cannot open directory '/etc/openvpn/client': Permission denied
ls: cannot open directory '/etc/openvpn/server': Permission denied
ls: cannot open directory '/etc/polkit-1/localauthority': Permission denied
ls: cannot open directory '/etc/polkit-1/rules.d': Permission denied
ls: cannot open directory '/etc/sos/cleaner': Permission denied
ls: cannot open directory '/etc/ssh/ssh_config.d': Permission denied
ls: cannot open directory '/etc/ssh/sshd_config.d': Permission denied
ls: cannot open directory '/etc/ssh/sshd_config.d': Permission denied
ls: cannot open directory '/etc/sudoers.d': Permission denied
[epkaneva@epkaneva ~]$ ls -R ~/ >> file.txt
```

Рис. 4.1: Запись файлов домашнего каталога и каталога `/etc`.

- Вывели имена всех файлов из `file.txt`, имеющих расширение `.conf`, после чего записали их в новый текстовый файл `conf.txt` (рис. 4.2 и 4.3):


```
[epkaneva@epkaneva ~]$ grep .conf file.txt
anthy-unicode.conf
appstream.conf
asound.conf
brltty.conf
chkconfig.d
chrony.conf
dconf
dleyna-renderer-service.conf
dleyna-server-service.conf
dnsmasq.conf
dracut.conf
dracut.conf.d
extlinux.conf
fprintd.conf
```

Рис. 4.2: Вывод файлов с расширением .conf.

```
[epkaneva@epkaneva ~]$ grep '\.conf' file.txt > conf.txt
```

Рис. 4.3: Запись названий нужных файлов в файл conf.txt.

- Определили, какие файлы в домашнем каталоге имеют имена, начинавши-
еся с символа с (рис. 4.4 и 4.5):

```
[epkaneva@epkaneva ~]$ find ~/ -name "c*" -print
/home/epkaneva/.mozilla/firefox/zeirw5vq.default-release/crashes
/home/epkaneva/.mozilla/firefox/zeirw5vq.default-release/compatibility.ini
/home/epkaneva/.mozilla/firefox/zeirw5vq.default-release/cookies.sqlite
/home/epkaneva/.mozilla/firefox/zeirw5vq.default-release/storage/permanent/chrome
/home/epkaneva/.mozilla/firefox/zeirw5vq.default-release/storage/default/https+++web.telegram.org/cache
/home/epkaneva/.mozilla/firefox/zeirw5vq.default-release/storage/default/https+++web.telegram.org/cache/caches.sqlite
```

Рис. 4.4: Способ 1.

```
[epkaneva@epkaneva ~]$ ls -l | grep c*
-rw-r--r--. 1 epkaneva epkaneva 9746 Mar 18 13:50 conf.txt
```

Рис. 4.5: Способ 2.

- Вывели на экран имена файлов из каталога /etc, начинающиеся с символа h (рис. 4.6):

```
[epkaneva@epkaneva ~]$ find /etc -name "h*" -print
find: '/etc/audit': Permission denied
/etc/avahi/hosts
/etc/brltty/Contraction/ha.ctb
/etc/brltty/Input/bm/horizontal.kti
/etc/brltty/Input/hd
```

Рис. 4.6: Файлы, начинающиеся с h.

- Запустили в фоновом режиме процесс, который будет записывать в файл ~/logfile файлы, имена которых начинаются с log (рис. 4.7). Как видим, идентификатор процесса - 4853.

```
[epkaneva@epkaneva ~]$ find ~ -name "log*" -print > logfile &
[1] 4853
```

Рис. 4.7: Запись имён нужных файлов в фоновом режиме.

- Удалили файл ~/logfile (рис. 4.8):

```
[epkaneva@epkaneva ~]$ rm logfile
[1]+  Done                  find ~ -name "log*" -print > logfile
```

Рис. 4.8: Удаление файла logfile.

- Запустили из консоли в фоновом режиме редактор gedit, определили идентификатор процесса gedit, используя команду ps, конвейер и фильтр grep (рис. 4.9). Идентификатор также указывался сразу после запуска команды:

```
[epkaneva@epkaneva ~]$ gedit &
[2] 4900
[1] Done gedit
[epkaneva@epkaneva ~]$ ps aux | grep gedit
epkaneva  4900  1.7  1.5 780420 62720 pts/1    Sl   13:59   0:00 gedit
epkaneva  4924  0.0  0.0 222168 2232 pts/1    S+   13:59   0:00 grep --color=auto gedit
[epkaneva@epkaneva ~]$ pgrep gedit
4900
[epkaneva@epkaneva ~]$ ps aux | grep gedit | grep -v grep
epkaneva  4900  0.7  1.5 780420 62720 pts/1    Sl   13:59   0:00 gedit
```

Рис. 4.9: Запуск gedit в фоновом режиме.

- Прочитали справку (man) команды `kill` (рис. 4.10), после чего использовали её для завершения процесса `gedit` (рис. 4.11):

```
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.
```

Рис. 4.10: Мануал по команде `kill`.

```
[epkaneva@epkaneva ~]$ kill 4900
[2]+  Terminated                  gedit
```

Рис. 4.11: Завершение процесса `gedit`.

- Получив подробную информацию о командах `df` (рис. 4.12) и `du` (рис. 4.13), выполнили их (рис. 4.14 и 4.15):

```
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.
```

Рис. 4.12: Мануал по команде `df`.

```

DU(1)                                     User Commands                                     DU(1)

NAME
    du - estimate file space usage

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

DESCRIPTION
    Summarize device usage of the set of FILES, recursively for directories.

```

Рис. 4.13: Мануал по команде du.

```

[epkaneva@epkaneva ~]$ df -vi
Filesystem      Inodes  IUsed   IFree IUse% Mounted on
devtmpfs        1048576  454    1048122   1% /dev
tmpfs           501559    4    501555   1% /dev/shm
tmpfs           819200   881   818319   1% /run
/dev/sda2        0         0         0    - /
tmpfs           1048576  43   1048533   1% /tmp
/dev/sda1        65536   456   65080    1% /boot
/dev/sda2        0         0         0    - /home
tmpfs           100311  168   100143   1% /run/user/1000

```

Рис. 4.14: Команда df.

```

[epkaneva@epkaneva ~]$ du ~/work/study
0      /home/epkaneva/work/study/2022-2023/Архитектура компьютера/arch-pc/.git/branches
60     /home/epkaneva/work/study/2022-2023/Архитектура компьютера/arch-pc/.git/hooks
4      /home/epkaneva/work/study/2022-2023/Архитектура компьютера/arch-pc/.git/info
4      /home/epkaneva/work/study/2022-2023/Архитектура компьютера/arch-pc/.git/refs/heads
0      /home/epkaneva/work/study/2022-2023/Архитектура компьютера/arch-pc/.git/refs/tags
8      /home/epkaneva/work/study/2022-2023/Архитектура компьютера/arch-pc/.git/refs/remotes/origin
8      /home/epkaneva/work/study/2022-2023/Архитектура компьютера/arch-pc/.git/refs/remotes
12     /home/epkaneva/work/study/2022-2023/Архитектура компьютера/arch-pc/.git/refs
24     /home/epkaneva/work/study/2022-2023/Архитектура компьютера/arch-pc/.git/objects/pack
0      /home/epkaneva/work/study/2022-2023/Архитектура компьютера/arch-pc/.git/objects/info
8      /home/epkaneva/work/study/2022-2023/Архитектура компьютера/arch-pc/.git/objects/36
4      /home/epkaneva/work/study/2022-2023/Архитектура компьютера/arch-pc/.git/objects/29
52     /home/epkaneva/work/study/2022-2023/Архитектура компьютера/arch-pc/.git/objects/65

```

Рис. 4.15: Команда du.

- Воспользовавшись справкой команды find (рис. 4.16), вывели имена всех директорий, имеющиххся в домашнем каталоге (рис. 4.17):

```

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 prece-
    dence (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.

```

Рис. 4.16: Мануал по команде find.

```
[epkaneva@epkaneva ~]$ find -type d
.
./.mozilla
./.mozilla/extensions
./.mozilla/extensions/{ec8030ff-c20a-464f-9b0e-13a3a9e97384}
./.mozilla/plugins
./.mozilla/firefox
./.mozilla/firefox/Crash Reports
./.mozilla/firefox/Crash Reports/events
./.mozilla/firefox/Pending Pings
./.mozilla/firefox/zeirw5vq.default-release
./.mozilla/firefox/zeirw5vq.default-release/minidumps
./.mozilla/firefox/zeirw5vq.default-release/crashes
./.mozilla/firefox/zeirw5vq.default-release/crashes/events
./.mozilla/firefox/zeirw5vq.default-release/storage
./.mozilla/firefox/zeirw5vq.default-release/storage/permanent
./.mozilla/firefox/zeirw5vq.default-release/storage/permanent/chrome
./.mozilla/firefox/zeirw5vq.default-release/storage/permanent/chrome/idb
./.mozilla/firefox/zeirw5vq.default-release/storage/permanent/chrome/idb/3870112724rsegmnoittet-es.files
./.mozilla/firefox/zeirw5vq.default-release/storage/permanent/chrome/idb/3870112724rsegmnoittet-es.files/journals
./.mozilla/firefox/zeirw5vq.default-release/storage/permanent/chrome/idb/3561288849sdhlie.files
./.mozilla/firefox/zeirw5vq.default-release/storage/permanent/chrome/idb/1451318868ntouomlalnody--epcr.files
```

Рис. 4.17: Вывод имён директорий.

5 Контрольные вопросы

1. Какие потоки ввода вывода вы знаете?

- `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. Объясните разницу между операцией `>` и `>>`.

Этот знак `>` - перенаправление ввода/вывода, а `>>` - перенаправление в режиме добавления.

3. Что такое конвейер?

Конвейер (`pipe`) служит для объединения простых команд или утилит в цепочки, в которых результат работы предыдущей команды передаётся последующей.

4. Что такое процесс? Чем это понятие отличается от программы?

Главное отличие между программой и процессом заключается в том, что программа - это набор инструкций, который позволяет ЦПУ выполнять определенную задачу, в то время как процесс - это исполняемая программа.

5. Что такое PID и GID?

PPID - (parent process ID) идентификатор родительского процесса. Процесс может порождать и другие процессы. UID, GID - реальные идентификаторы пользователя и его группы, запустившего данный процесс.

6. Что такое задачи и какая команда позволяет ими управлять?

Запущенные фоном программы называются задачами (jobs). Ими можно управлять с помощью команды `jobs`, которая выводит список запущенных в данный момент задач.

7. Найдите информацию об утилитах `top` и `htop`. Каковы их функции?

Команда `htop` похожа на команду `top` по выполняемой функции: они обе показывают информацию о процессах в реальном времени, выводят данные о потреблении системных ресурсов и позволяют искать, останавливать и управлять процессами.

У обеих команд есть свои преимущества. Например, в программе `htop` реализован очень удобный поиск по процессам, а также их фильтрация. В команде `top` это не так удобно — нужно знать кнопку для вывода функции поиска.

Зато в `top` можно разделять область окна и выводить информацию о процессах в соответствии с разными настройками. В целом `top` намного более гибкая в настройке отображения процессов.

8. Назовите и дайте характеристику команде поиска файлов. Приведите примеры использования этой команды.

Команда `find` - это одна из наиболее важных и часто используемых утилит системы Linux. 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.

Утилита `find` предустановлена по умолчанию во всех Linux дистрибутивах, поэтому вам не нужно будет устанавливать никаких дополнительных пакетов.

Это очень важная находка для тех, кто хочет использовать командную строку наиболее эффективно.

Команда `find` имеет такой синтаксис:

```
find [папка] [параметры] критерий шаблон [действие]
```

Пример: см. рис. 4.17.

9. Можно ли по контексту (содержанию) найти файл? Если да, то как?

```
find / -type f -exec grep -H 'text_here' {} \
```

10. Как определить объем свободной памяти на жёстком диске?

С помощью команды `df -h`.

11. Как определить объем вашего домашнего каталога?

С помощью команды `du -s`.

12. Как удалить зависший процесс?

С помощью команды `kill` номер задачи.

6 Выводы

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