

# **Лабораторная работа № 8**

**Поиск файлов. Перенаправление ввода-вывода. Просмотр  
запущенных процессов**

Павличенко Родион Андреевич

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>5</b>
<b>2</b>	<b>Выполнение лабораторной работы</b>	<b>6</b>
<b>3</b>	<b>Контрольный вопросы</b>	<b>9</b>
<b>4</b>	<b>Выводы</b>	<b>11</b>

# Список иллюстраций

2.1	Пункты 1 и 2 . . . . .	6
2.2	Пункты 1 и 2 . . . . .	6
2.3	Вывод файлов . . . . .	6
2.4	Запись файлов . . . . .	7
2.5	Поиск файлов . . . . .	7
2.6	Поиск файлов . . . . .	7
2.7	Работа с файлом logfile . . . . .	7
2.8	Завершение процесса . . . . .	7
2.9	Выполнение команд df и du . . . . .	8
2.10	Выводим имена всех директорий . . . . .	8

## **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

Осуществим вход в систему, используя соответствующее имя пользователя. Запишем в файл file.txt названия файлов, содержащихся в каталоге /etc. Допишем в этот же файл названия файлов, содержащихся в домашнем каталоге.

```
rapavlichenko@rapavlichenko:~$ ls /etc > file.txt
rapavlichenko@rapavlichenko:~$ cat file.txt
abrt
adjtime
```

Рис. 2.1: Пункты 1 и 2

```
rapavlichenko@rapavlichenko:~$ ls >> file.txt
rapavlichenko@rapavlichenko:~$ tac file.txt
Шаблоны
Вспомогательные
```

Рис. 2.2: Пункты 1 и 2

Вывели имена всех файлов из file.txt, имеющих расширение .conf, после чего записали их в новый текстовый файл conf.txt.

```
rapavlichenko@rapavlichenko:~$ grep .conf file.txt
anthy-unicode.conf
asound.conf
brltty.conf
chkconfig.d
chrony.conf
dconf
dley-na-server-service.conf
dnsmasq.conf
dracut.conf
```

Рис. 2.3: Вывод файлов

```
rapavlichenko@rapavlichenko:~$ grep "\.conf" file.txt > conf.txt
rapavlichenko@rapavlichenko:~$
```

Рис. 2.4: Запись файлов

Найдем файлы, начинающиеся с “с”

```
rapavlichenko@rapavlichenko:~$ ls -r | grep c
conf.txt
```

Рис. 2.5: Поиск файлов

Найдем файлы, начинающиеся с “h”

```
rapavlichenko@rapavlichenko:~$ find ~ name "h*" -print | less
rapavlichenko@rapavlichenko:~$
```

Рис. 2.6: Поиск файлов

Запустили в фоновом режиме процесс, который будет записывать в файл ~/logfile файлы, имена которых начинаются с log. Удалили файл ~/logfile.

```
rapavlichenko@rapavlichenko:~$ rm logfile
```

Рис. 2.7: Работа с файлом logfile

Запустили из консоли в фоновом режиме редактор gedit и определение его id.

```
rapavlichenko@rapavlichenko:~$ gedit &
[3] 14603
```

```
rapavlichenko@rapavlichenko:~$ ps | gre
14603 pts/0    00:00:00 gedit
rapavlichenko@rapavlichenko:~$ pidof ge
14603
```

Прочли справку (man) команды kill, после чего использовали её для завершения процесса gedit.

```
rapavlichenko@rapavlichenko:~$ man kill
rapavlichenko@rapavlichenko:~$ kill 14603
rapavlichenko@rapavlichenko:~$
```

Рис. 2.8: Завершение процесса

Выполнили команды `df` и `du`, предварительно получив более подробную информацию об этих командах, с помощью команды `man`.

```
rapavlichenko@rapavlichenko:~$ man df
rapavlichenko@rapavlichenko:~$ man du
rapavlichenko@rapavlichenko:~$ df
Файловая система 1K-блоков  Использовано  Доступно  Использовано%  Смонтир
овано в
/dev/sda3          82834432      15147000  66853192          19% /
devtmpfs           4096          0        4096            0% /dev
tmpfs              5561392       96       5561296          1% /dev/sh
m
tmpfs              2224560       1372     2223188          1% /run
tmpfs              1024          0        1024            0% /run/cr
edentials/systemd-journald.service
tmpfs              1024          0        1024            0% /run/cr
edentials/systemd-network-generator.service
tmpfs              1024          0        1024            0% /run/cr
edentials/systemd-udev-load-credentials.service
tmpfs              1024          0        1024            0% /run/cr
edentials/systemd-sysctl.service
tmpfs              1024          0        1024            0% /run/cr
edentials/systemd-tmpfiles-setup-dev-early.service
tmpfs              1024          0        1024            0% /run/cr
edentials/systemd-tmpfiles-setup-dev.service
tmpfs              5561392       16       5561376          1% /tmp
/dev/sda3          82834432      15147000  66853192          19% /home
tmpfs              1024          0        1024            0% /run/cr
edentials/systemd-vconsole-setup.service
/dev/sda2          996780       333740     594228          36% /boot
tmpfs              1024          0        1024            0% /run/cr
edentials/systemd-tmpfiles-setup.service
tmpfs              1024          0        1024            0% /run/cr
edentials/systemd-resolved.service
tmpfs              1112276       208     1112068          1% /run/us
er/1000
/dev/sr0           52250        52250         0          100% /run/me
dia/rapavlichenko/VBox_GAs_7.0.20
rapavlichenko@rapavlichenko:~$ du
8      ../mozilla/extensions/{ec8030f7-c20a-464f-9b0e-13a3a9e97384}
8      ../mozilla/extensions
```

Рис. 2.9: Выполнение команд `df` и `du`

Выводим имена всех директорий

Выводим имена всех директорий

Рис. 2.10: Выводим имена всех директорий



### 3 Контрольный вопросы

1. Существуют три основных потока ввода-вывода: стандартный ввод (stdin, дескриптор 0), стандартный вывод (stdout, дескриптор 1) и стандартный поток ошибок (stderr, дескриптор 2).
2. Операция > используется для перезаписи содержимого файла, тогда как >> добавляет данные в конец файла, не удаляя его текущее содержимое.
3. Конвейер (pipeline) – это способ передачи вывода одной команды в качестве ввода для другой с использованием символа |, что позволяет объединять несколько команд в цепочку обработки данных.
4. Процесс – это запущенный экземпляр программы с собственным адресным пространством и состоянием выполнения. Программа – это просто файл на диске, содержащий исполняемый код, а процесс является активным выполнением этой программы в памяти.
5. PID (Process ID) – уникальный идентификатор процесса в системе. GID (Group ID) – идентификатор группы пользователей, который определяет права доступа в многопользовательской системе.
6. Задачи – это процессы, выполняющиеся в фоновом или приостановленном режиме. Управление задачами осуществляется с помощью команды jobs, а также команд fg, bg и kill для возобновления или завершения задач.
7. Утилита top отображает динамическую информацию о запущенных процессах, загрузке процессора и потреблении памяти. htop – это более удобная и

наглядная альтернатива `top`, предоставляющая цветовую разметку и возможность управления процессами с помощью клавиш.

8. Для поиска файлов используется команда `find`. Она позволяет искать файлы по имени, типу, размеру и другим параметрам. Примеры: `find /home -name "file.txt"`, `find /var -size +100M`, `find /etc -type d` (поиск директорий).
9. Найти файл по его содержимому можно с помощью команды `grep`. Например, `grep "поиск" /home/user/*.txt` ищет слово “поиск” во всех `.txt` файлах в указанной директории.
10. Определить объем свободного места на диске можно с помощью команды `df -h`, которая показывает информацию в удобном для чтения формате.
11. Определить объем домашнего каталога можно с помощью команды `du -sh ~/`, которая выводит общий размер каталога в человекочитаемом формате.
12. Удалить зависший процесс можно с помощью команды `kill PID`, где `PID` – идентификатор процесса. Если процесс не завершается, можно использовать `kill -9 PID` для принудительного завершения.

## **4 Выводы**

Мы 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.