

# **Лабораторная работа № 6**

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

Шулуужук Айраана Вячеславовна НПИбд-02-22

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>5</b>
<b>2</b>	<b>Задание</b>	<b>6</b>
<b>3</b>	<b>Теоретическое введение</b>	<b>8</b>
3.1	Перенаправление ввода-вывода . . . . .	8
3.2	Конвейер . . . . .	8
3.3	Поиск файлов . . . . .	9
3.4	Фильтрация текста . . . . .	9
3.5	Проверка использования диска . . . . .	10
3.6	Управление задачами . . . . .	10
3.7	Управление процессами . . . . .	10
3.8	Получение информации о процессах . . . . .	11
<b>4</b>	<b>Выполнение лабораторной работы</b>	<b>12</b>
<b>5</b>	<b>Контрольные вопросы</b>	<b>17</b>
<b>6</b>	<b>Выводы</b>	<b>20</b>

## Список иллюстраций

4.1	запись файлов в file.txt . . . . .	12
4.2	запись файлов с расширением .conf в новый файл conf.txt . . . .	12
4.3	вывод файлов, начинающие с с- . . . . .	13
4.4	вывод имен файлов из каталога /etc, начинающие с h . . . . .	13
4.5	процесс записи файлов, начинающие с log, в файл ~/logfile . . . .	13
4.6	запуск редактора в фоновом режиме . . . . .	13
4.7	определение идентификатора процесса gedit . . . . .	14
4.8	завершение процесса gedit командой kill . . . . .	14
4.9	команда df . . . . .	15
4.10	команда du . . . . .	15
4.11	вывод имен всех директорий в домашнем каталоге . . . . .	16

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

11. Выполните команды `df` и `du`, предварительно получив более подробную информацию об этих командах, с помощью команды `man`.
12. Воспользовавшись справкой команды `find`, выведите имена всех директорий, имеющихся в вашем домашнем каталоге.

## 3 Теоретическое введение

### 3.1 Перенаправление ввода-вывода

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`. Например, команда `ls` выводит в стандартный поток вывода (консоль) список файлов в текущей директории. Потоки вывода и ввода можно перенаправлять на другие файлы или устройства. Проще всего это делается с помощью символов `>`, `»`, `<`, `«`.

### 3.2 Конвейер

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

команда 1 | команда 2

# означает, что вывод команды 1 передаётся на ввод команде 2



Конвейеры можно группировать в цепочки и выводить с помощью перенаправления в файл, например:

```
1 ls -la |sort > sortlg_list
```

вывод команды `ls -la` передаётся команде сортировки `sort` которая пишет результат в файл. В то рутинных операций в консоли отсюда иногда возникает необходимость в обработке `stdout` одной команды и передача на `stdin` другой команде, при этом результат выполнения команды должен обработан

### 3.3 Поиск файлов

Команда `find` используется для поиска и отображения на экран имён файлов, соответствующих заданной строке символов. Формат команды:

```
find путь [-опции]
```

Путь определяет каталог, начиная с которого по всем подкаталогам будет вестись поиск

### 3.4 Фильтрация текста

Найти в текстовом файле указанную строку символов позволяет команда `grep`. Формат команды:

```
grep строка имя_файла
```

Кроме того, команда `grep` способна обрабатывать стандартный вывод других команд (любой текст). Для этого следует использовать конвейер, связав вывод команды с вводом `grep`.

## 3.5 Проверка использования диска

Команда `df` показывает размер каждого смонтированного раздела диска. Формат команды:

```
df [-опции] [файловая_система]
```

Команда `du` показывает число килобайт, используемое каждым файлом или каталогом. Формат команды:

```
du [-опции] [имя_файла...]
```

## 3.6 Управление задачами

Любую выполняющуюся в консоли команду или внешнюю программу можно запустить в фоновом режиме. Для этого следует в конце имени команды указать знак амперсанда `&`. Например:

```
gedit &
```

Будет запущен текстовый редактор `gedit` в фоновом режиме. Консоль при этом не будет заблокирована. Запущенные фоновые 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
```

## 3.7 Управление процессами

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.

## 3.8 Получение информации о процессах

Команда `ps` используется для получения информации о процессах. Формат команды:

```
ps [-опции]
```

Для получения информации о процессах, управляемых вами и запущенных (работающих или остановленных) на вашем терминале, используйте опцию `aux`.

Пример:

```
ps aux
```

Для запуска команды в фоновом режиме необходимо в конце командной строки указать знак `&` (амперсанд). Пример работы, требующей много машинного времени для выполнения, и которую целесообразно запустить в фоновом режиме:

```
find /var/log -name "*.log" -print > l.log &
```

## 4 Выполнение лабораторной работы

1. Запишем файл file.txt названия файлов, содержащихся в каталоге/etc. Допишем в этот же файл названия файлов, содержащихся в домашнем каталоге.(рис. 4.1)

```
[avshuluuzhuk@fedora ~]$ ls -a /etc > file.txt  
[avshuluuzhuk@fedora ~]$ ls -a ~ >> file.txt  
[avshuluuzhuk@fedora ~]$
```

Рис. 4.1: запись файлов в file.txt

2. Выводим имена всех файлов из file.txt, имеющих расширение .conf, после чего запишем их в новый текстовый файл conf.txt. (рис. 4.2)

```
xattr.conf  
[avshuluuzhuk@fedora ~]$ cat file.txt | grep .conf$ >> conf.txt  
[avshuluuzhuk@fedora ~]$ cat file.txt | grep .conf$  
anthy-unicode.conf  
appstream.conf  
asound.conf  
brltty.conf  
chrony.conf  
dconf  
dleyna-renderer-service.conf  
dleyna-server-service.conf  
dnsmasq.conf  
dracut.conf  
fontconfig.conf
```

Рис. 4.2: запись файлов с расширением .conf в новый файл conf.txt

3. Определим файлы в домашнем каталоге, начинающие с символа с (рис. 4.3)

```

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

```

Рис. 4.3: вывод файлов, начинающие с с-

4. Выводим на экран имена файлов из каталога /etc, начинающие с символа h (рис. 4.4)

```

[avshuluuzhuk@fedora ~]$ find ~ -maxdepth 1 -name "c*" -print
/home/avshuluuzhuk/conf.txt
[avshuluuzhuk@fedora ~]$ find /etc -maxdepth 1 -name h*
/etc/hp
/etc/httpd
/etc/host.conf
/etc/hosts
/etc/hostname
[avshuluuzhuk@fedora ~]$ \

```

Рис. 4.4: вывод имен файлов из каталога /etc, начинающие с h

5. Запустим в фоновом режиме процесс, который будет записывать в файл ~/logfile файлы, имена которых начинаются с log (рис. 4.5). Далее удаляем этот файл

```

[avshuluuzhuk@fedora ~]$ find /etc -maxdepth 1 -name "log*" >> ~/logfile &
[1] 7229
[1]+  Завершён      find /etc -maxdepth 1 -name "log*" >> ~/logfile
[avshuluuzhuk@fedora ~]$ rm logfile
[avshuluuzhuk@fedora ~]$

```

Рис. 4.5: процесс записи файлов, начинающие с log, в файл ~/logfile

6. Запустим из консоли в фоновом режиме редактор gedit. (рис. 4.6).

```

[avshuluuzhuk@fedora ~]$ gedit &
[1] 7302

```

Рис. 4.6: запуск редактора в фоновом режиме

7. Определим идентификатор процесса gedit, используя команду ps, конвейер и фильтр grep. (рис. 4.7)

```
[avshuluuzhuk@fedora ~]$ gedit &
[1] 7302
[avshuluuzhuk@fedora ~]$ ps
  PID TTY          TIME CMD
 6698 pts/0    00:00:00 bash
  7302 pts/0    00:00:00 gedit
  7326 pts/0    00:00:00 ps
[avshuluuzhuk@fedora ~]$ ps | grep "gedit"
  7302 pts/0    00:00:00 gedit
[avshuluuzhuk@fedora ~]$
```

Рис. 4.7: определение идентификатора процесса gedit

8. Просмотрим описание команды kill, и после чего используем ее для завершения процесса gedit (рис. 4.8)

```
[avshuluuzhuk@fedora ~]$ ps | grep "gedit"
  7302 pts/0    00:00:00 gedit
[avshuluuzhuk@fedora ~]$ man kill
[avshuluuzhuk@fedora ~]$ kill 7302
[1]+  Завершено      gedit
[avshuluuzhuk@fedora ~]$
```

Рис. 4.8: завершение процесса gedit командой kill

9. Выполним команды df и du, предварительно получив более подробную информацию об этих командах, с помощью команды man (рис. 4.9) (рис. 4.10)

Команда df показывает размер каждого смонтированного раздела диска

Команда du показывает число килобайт, используемых каждым файлом или каталогом

```
[avshuluuzhuk@fedora ~]$ df
Файловая система 1K-блоков  Использовано  Доступно  Использовано%  Смонтировано в
devtmpfs           4096             0         4096             0% /dev
tmpfs              2006236         10924      1995312             1% /dev/shm
tmpfs              802496          1420       801076             1% /run
/dev/sda3          30405632        15144652    14329252           52% /
tmpfs              2006240          16        2006224             1% /tmp
/dev/sda2          996780         246040      681928             27% /boot
/dev/sda3          30405632        15144652    14329252           52% /home
tmpfs              401244          172        401072             1% /run/user/1000
/dev/sr0           51806          51806         0            100% /run/media/avshuluuzh
```

Рис. 4.9: команда df

```
[avshuluuzhuk@fedora ~]$ du
8      ./mozilla/extensions/{ec8030f7-c20a-464f-9b0e-13a3a9e97384}
8      ./mozilla/extensions
0      ./mozilla/plugins
0      ./mozilla/firefox/Crash Reports/events
4      ./mozilla/firefox/Crash Reports
0      ./mozilla/firefox/Pending Pings
0      ./mozilla/firefox/8nv4gckb.default-release/minidumps
0      ./mozilla/firefox/8nv4gckb.default-release/crashes/events
4      ./mozilla/firefox/8nv4gckb.default-release/crashes
2680   ./mozilla/firefox/8nv4gckb.default-release/security_state
0      ./mozilla/firefox/8nv4gckb.default-release/storage/permanent/chrome/idb/38
```

Рис. 4.10: команда du

10. Воспользовавшись справкой команды find, выводим имена всех директорий, имеющихсх домашнем каталоге (рис. 4.11)

```
[avshuluuzhuk@fedora ~]$ man find
[avshuluuzhuk@fedora ~]$ find ~/ -maxdepth 1 -type d
/home/avshuluuzhuk/
/home/avshuluuzhuk/.mozilla
/home/avshuluuzhuk/.cache
/home/avshuluuzhuk/.config
/home/avshuluuzhuk/.local
/home/avshuluuzhuk/Рабочий стол
/home/avshuluuzhuk/Загрузки
/home/avshuluuzhuk/Шаблоны
/home/avshuluuzhuk/Общедоступные
/home/avshuluuzhuk/Документы
/home/avshuluuzhuk/Музыка
/home/avshuluuzhuk/Изображения
/home/avshuluuzhuk/Видео
/home/avshuluuzhuk/.texlive2021
/home/avshuluuzhuk/.ssh
/home/avshuluuzhuk/work
/home/avshuluuzhuk/.gnupg
/home/avshuluuzhuk/bin
```

Рис. 4.11: вывод имен всех директорий в домашнем каталоге



## 5 Контрольные вопросы

1. Какие потоки ввода вывода вы знаете?

Ввод и вывод распределяется между тремя стандартными потоками: - `stdin` — стандартный ввод (клавиатура), - `stdout` — стандартный вывод (экран), - `stderr` — стандартная ошибка (вывод ошибок на экран).

2. Объясните разницу между операцией `>` и `»`.

Основное отличие: `>` : Перезаписывает существующий файл или создает файл, если файл с указанным именем отсутствует в каталоге. `»` : добавляет существующий файл или создает файл, если файл с указанным именем отсутствует в каталоге.

3. Что такое конвейер?

Конвейер (англ. `pipeline`) в терминологии операционных систем семейства Unix — некоторое множество процессов, для которых выполнено следующее перенаправление ввода-вывода: то, что выводит на поток стандартного вывода предыдущий процесс, попадает в поток стандартного ввода следующего процесса.

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

Процесс - это программа на стадии выполнения - “объект”, которому выделено процессорное время - асинхронная работа

## 5. Что такое PID и GID?

Идентификатор процесса (PID). Каждому новому процессу ядро присваивает уникальный идентификационный номер. В любой момент времени идентификатор процесса является уникальным, хотя после завершения процесса он может использоваться снова для другого процесса. Некоторые идентификаторы зарезервированы системой для особых процессов. Так, процесс с идентификатором 1 - это процесс инициализации `init`, являющийся предком всех других процессов в системе. Идентификатор группы GID и эффективный идентификатор группы (EGID) GID - это идентификационный номер группы данного процесса. EGID связан с GID также, как EUID с UID.

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

Принудительное завершение процесса и изменение его приоритета) можно выполнить и без команды `top`. Процессы в Linux имеют возможность обмениваться так называемыми “сигналами” с ядром и другими процессами. При получении сигнала процессом, управление передается подпрограмме его обработки или ядру, если такой подпрограммы не существует. В Linux имеется команда `kill`, которая позволяет послать заданному процессу любой сигнал.

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

`top` - интерактивный просмотрщик процессов. `htop` аналог `top`. Программа `top` динамически выводит в режиме реального времени информации о работающей системе, т.е. о фактической активности процессов. По умолчанию она выдает задачи, наиболее загружающие процессор сервера, и обновляет список каждые две секунды.

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

`find` : Для поиска файлов из командной строки вы можете использовать команду “`find`”. У этой команды следующий синтаксис: `find path criteria action - “path”` - Секция для указания директории поиска. Если ничего не указано поиск идет по текущей директории. - “`criteria`” - Опции поиска. - “`action`” - Опции, которые влияют на состояние поиска или контролируют его, например, - “`-print`”

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

Для поиска файла по содержимому проще всего воспользоваться командой `grep` (вместо `find`). Пример: `grep -r строка_поиска каталог`

10. Как определить объем свободной памяти на жёстком диске?

Самый простой способ найти свободное место на диске в Linux - это используйте команду `df`. Команда `df` означает «свободное от диска» и, очевидно, показывает вам свободное и доступное дисковое пространство в системах Linux. Работы С Нами -h вариант, он показывает дисковое пространство в удобочитаемом формате (МБ и ГБ).

11. Как определить объем вашего домашнего каталога?

В операционных системах на базе Linux посмотреть размер папки (директории) можно с помощью команды `du`. Эта команда, выполняемая в консоли, позволяет оценить используемый объем места на жестком диске отдельно по папкам и файлам, просуммировать результат, узнать общий размер папки.

12. Как удалить зависший процесс?

Убиваем процессы в Linux — команды `ps`, `kill` и `killall`

- Находим PID зависшего процесса Каждый процесс в Linux имеет свой идентификатор, называемый PID.

- «Убиваем» процесс командой `kill`. Когда известен PID процесса, мы можем убить его командой `kill`.

- Убиваем процессы командой `killall`.

- Заключение

## 6 Выводы

В ходе выполнения работы 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