

# **ОТЧЕТ О ВЫПОЛНЕНИИ ЛАБОРАТОРНОЙ РАБОТЫ №6**

*дисциплина: Операционные системы*

Лихтенштейн Алина Алексеевна

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>4</b>
<b>2</b>	<b>Задачи</b>	<b>5</b>
<b>3</b>	<b>Теоретическое введение</b>	<b>6</b>
3.1	Перенаправление ввода-вывода . . . . .	6
3.2	Конвейер . . . . .	7
3.3	Поиск файла . . . . .	8
3.4	Фильтрация текста . . . . .	8
3.5	Проверка использования диска . . . . .	9
3.6	Управление задачами . . . . .	10
3.7	Управление процессами . . . . .	10
3.8	Получение информации о процессах . . . . .	10
<b>4</b>	<b>Выполнение лабораторной работы</b>	<b>12</b>
<b>5</b>	<b>Выводы</b>	<b>22</b>

## Список иллюстраций

4.1	вход в систему . . . . .	12
4.2	запись названий файлов в файл . . . . .	13
4.3	вывод имен файлов с расширением .conf . . . . .	14
4.4	запись в новый текстовый файл . . . . .	14
4.5	использование команд <code>ls ~/C, find ~/ -maxdepth 1 -name "c"</code> . . . . .	15
4.6	команда . . . . .	15
4.7	вывод команды . . . . .	15
4.8	запись файлов, начинающихся с log в файл ~/logfile . . . . .	16
4.9	удаление файла . . . . .	16
4.10	запуск gedit . . . . .	16
4.11	запуск gedit . . . . .	16
4.12	команда <code>man kill</code> . . . . .	17
4.13	команда <code>kill</code> . . . . .	17
4.14	информация о команде <code>df</code> . . . . .	18
4.15	информация о команде <code>du</code> . . . . .	19
4.16	выполнение команды <code>df</code> . . . . .	19
4.17	выполнение команды <code>du</code> . . . . .	20
4.18	команда . . . . .	20
4.19	вывод команды . . . . .	21

# 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 Задачи

1. Осуществим вход в систему, используя соответствующее имя пользователя.
2. Запишем в файл `file.txt` названия файлов, содержащихся в каталоге `/etc`. Допишем в этот же файл названия файлов, содержащихся в вашем домашнем каталоге.
3. Выведем имена всех файлов из `file.txt`, имеющих расширение `.conf`, после чего запишем их в новый текстовый файл `conf.txt`.
4. Определим, какие файлы в вашем домашнем каталоге имеют имена, начинавшиеся с символа `s`. Предложим несколько вариантов, как это сделать.
5. Выведем на экран (по странично) имена файлов из каталога `/etc`, начинающиеся с символа `h`.
6. Запустим в фоновом режиме процесс, который будет записывать в файл `~/logfile` файлы, имена которых начинаются с `log`.
7. Удалим файл `~/logfile`.
8. Запустим из консоли в фоновом режиме редактор `gedit`.
9. Определим идентификатор процесса `gedit`, используя команду `ps`, конвейер и фильтр `grep`. Узнаем, как ещё можно определить идентификатор процесса.
10. Прочтем справку (`man`) команды `kill`, после чего используем её для завершения процесса `gedit`.
11. Выполним команды `df` и `du`, предварительно получив более подробную информацию об этих командах, с помощью команды `man`.
12. Воспользовавшись справкой команды `find`, выведем 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 `>`, `>>`, `<`, `<<`. Рассмотрим пример.

```
# Перенаправление stdout (вывода) в файл.  
# Если файл отсутствовал, то он создаётся,  
# иначе -- перезаписывается.  
  
# Создаёт файл, содержащий список дерева каталогов.  
ls -lR > dir-tree.list  
1>filename  
# Перенаправление вывода (stdout) в файл "filename".  
1>>filename  
# Перенаправление вывода (stdout) в файл "filename",
```

```
# файл открывается в режиме добавления.  
2>filename  
# Перенаправление stderr в файл "filename".  
2>>filename  
# Перенаправление stderr в файл "filename",  
# файл открывается в режиме добавления.  
&>filename  
# Перенаправление stdout и stderr в файл "filename".
```

## 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:

```
ls -la |sort > sortilg_list
```

вывод команды `ls -la` передаётся команде сортировки `sort\verb`, которая пишет результат в файл `sorting_list\verb`. Чаще всего скрипты на Bash используются в качестве автоматизации каких-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 файла

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.

**Примеры:** 1. Вывести на экран имена файлов из вашего домашнего каталога и его подкаталогов, начинающихся на `f`:

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

Здесь `~` — обозначение вашего домашнего каталога, `-name` — после этой опции указывается имя файла, который нужно найти, `"f*"` — строка символов, определяющая имя файла, `-print` — опция, задающая вывод результатов поиска на экран.

2. Вывести на экран имена файлов в каталоге `/etc`, начинающихся с символа `p`:

```
find /etc -name "p*" -print
```

3. Найти в Вашем домашнем каталоге файлы, имена которых заканчиваются символом `~` и удалить их:

```
find ~ -name "*~" -exec rm "{}" \;
```

Здесь опция `-exec rm "{}"` ; задаёт применение команды `rm` ко всем файлам, имена которых соответствуют указанной после опции `-name` строке символов. Для просмотра опций команды `find` воспользуйтесь командой `man`.

### 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`. **Примеры:** 1. Показать строки во всех файлах в вашем домашнем каталоге с именами, начинающимися на `f`, в которых есть слово `begin`:

```
grep begin f*
```

2. Найти в текущем каталоге все файлы, содержащих в имени «лаб»:

```
ls -l | grep лаб
```

## 3.5 Проверка использования диска

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

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

**Пример:**

```
df -vi
```

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

```
du [-опции] [имя_файла...]
```

**Пример.**

```
du -a ~/
```

На `afs` можно посмотреть использованное пространство командой

```
fs quota
```

## 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. Осуществим вход в систему, используя соответствующее имя пользователя (рис. [4.1])

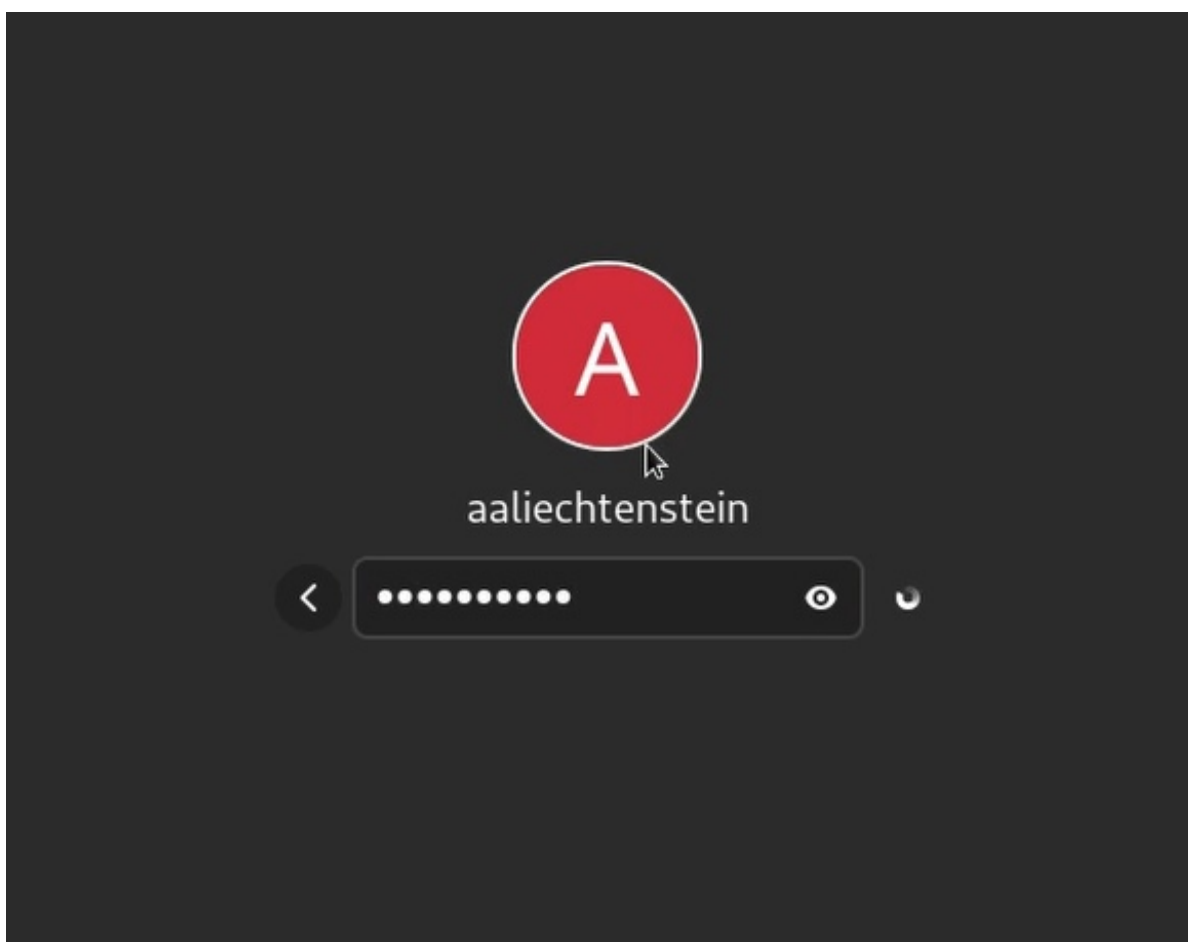


Рис. 4.1: вход в систему

2. Запишем в файл `file.txt` названия файлов, содержащихся в каталоге `/etc`. Допишем в этот же файл названия файлов, содержащихся в вашем домашнем каталоге (рис. [4.2])

```
[aaliechtenstein@aaliechtenstein ~]$ ls /etc > file.txt  
[aaliechtenstein@aaliechtenstein ~]$ ls ~ >> file.txt
```

Рис. 4.2: запись названий файлов в файл

3. Выведем имена всех файлов из `file.txt`, имеющих расширение `.conf`, после чего запишем их в новый текстовый файл `conf.txt` (рис. [4.3]) (рис. [4.4])

```
[aaliechtenstein@aaliechtenstein ~]$ grep ".conf" file.txt
anthy-unicode.conf
appstream.conf
asound.conf
brltty.conf
chkconfig.d
chrony.conf
dconf
dley-na-renderer-service.conf
dley-na-server-service.conf
dnsmasq.conf
dracut.conf
dracut.conf.d
fprintd.conf
fuse.conf
host.conf
idmapd.conf
kdump.conf
krb5.conf
krb5.conf.d
ld.so.conf
ld.so.conf.d
libaudit.conf
libuser.conf
```

Рис. 4.3: вывод имен файлов с расширением .conf

```
[aaliechtenstein@aaliechtenstein ~]$ grep ".conf" file.txt > conf.txt
[aaliechtenstein@aaliechtenstein ~]$
```

Рис. 4.4: запись в новый текстовый файл

4. Определим, какие файлы в вашем домашнем каталоге имеют имена, начинающиеся с символа с. Предложим несколько вариантов, как это сделать (рис. [4.5])

```
[aaliechtenstein@aaliechtenstein ~]$ ls ~/c*  
/home/aaliechtenstein/conf.txt  
[aaliechtenstein@aaliechtenstein ~]$ find ~/ -maxdepth 1 -name "c*"  
/home/aaliechtenstein/conf.txt  
[aaliechtenstein@aaliechtenstein ~]$
```

Рис. 4.5: использование команд `ls ~/C`, `find ~/ -maxdepth 1 -name "c"`

5. Выведем на экран (по странично) имена файлов из каталога `/etc`, начинающиеся с символа `h` (рис. [4.6]) (рис. [4.7])

```
[aaliechtenstein@aaliechtenstein ~]$ ls /etc | grep "^h" | less
```

Рис. 4.6: команда



```
aaliechtenstein@aaliechtenstein:~ — less  
host.conf  
hostname  
hosts  
hp  
httpd  
(END)
```

Рис. 4.7: вывод команды

6. Запустим в фоновом режиме процесс, который будет записывать в файл `~/logfile` файлы, имена которых начинаются с `log` (рис. [4.8])

```
[aaliechtenstein@aaliechtenstein ~]$ sudo find / -name "log*" > logfile.txt &  
[1] 6691
```

Рис. 4.8: запись файлов, начинающихся с log в файл ~/logfile

7. Удалим файл ~/logfile (рис. [4.9])

```
[aaliechtenstein@aaliechtenstein ~]$ rm logfile.txt
```

Рис. 4.9: удаление файла

8. Запустим из консоли в фоновом режиме редактор gedit (рис. [4.10])

```
[aaliechtenstein@aaliechtenstein ~]$ gedit &  
[1] 6806
```

Рис. 4.10: запуск gedit

9. Определим идентификатор процесса gedit, используя команду ps, конвейер и фильтр grep. Узнаем, как ещё можно определить идентификатор процесса (рис. [4.11])

```
[aaliechtenstein@aaliechtenstein ~]$ ps | grep gedit  
6806 pts/0    00:00:00 gedit  
[aaliechtenstein@aaliechtenstein ~]$ pidof gedit  
6806  
[aaliechtenstein@aaliechtenstein ~]$
```

Рис. 4.11: запуск gedit



10. Прочтем справку (man) команды kill, после чего используем её для завершения процесса gedit (рис. [4.12]) (рис. [4.13])

```
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
```

Рис. 4.12: команда man kill

```
[aaliechtenstein@aaliechtenstein ~]$ man kill
[aaliechtenstein@aaliechtenstein ~]$ kill 7019
[aaliechtenstein@aaliechtenstein ~]$ ps aux | grep "gedit"
aaliech+   7076   0.0   0.1 222044   2440 pts/0    S+   21:54   0:00 grep --color=
auto gedit
[2]+  Завершено      gedit
[aaliechtenstein@aaliechtenstein ~]$
```

Рис. 4.13: команда kill

11. Выполним команды df и du, предварительно получив более подробную информацию об этих командах, с помощью команды man (рис. [4.14]) (рис. [4.15]) (рис. [4.16]) (рис. [4.17])

```
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 cur-
  rently 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 ver-
  sion of df cannot show the space available on unmounted file systems,
  because on most kinds of systems doing so requires very nonportable in-
  timate knowledge of file system structures.
```

Рис. 4.14: информация о 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.

    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

    --apparent-size
```

Рис. 4.15: информация о команде du

```
[aaliechtenstein@aaliechtenstein ~]$ df
Файловая система 1K-блоков  Использовано  Доступно  Использовано%  Смонтировано в
devtmpfs          4096          0      4096          0% /dev
tmpfs             1008616        0  1008616          0% /dev/shm
tmpfs             403448       1384   402064          1% /run
/dev/sda3         82834432    12022572  67801764         16% /
tmpfs             1008620        16   1008604          1% /tmp
/dev/sda3         82834432    12022572  67801764         16% /home
/dev/sda2         996780       296080   631888          32% /boot
/dev/sr0          62386       62386         0        100% /run/media/aaliec
htenstein/VBox_GAs_6.1.40
tmpfs             201720       132   201588          1% /run/user/1000
[aaliechtenstein@aaliechtenstein ~]$
```

Рис. 4.16: выполнение команды df

```
[aaliechtenstein@aaliechtenstein ~]$ du
8      ./mozilla/extensions/{ec8030f7-c20a-464f-9b0e-13a3a9e97384}
8      ./mozilla/extensions
0      ./mozilla/plugins
2680   ./mozilla/firefox/dq48xt6q.default-release/security_state
0      ./mozilla/firefox/dq48xt6q.default-release/storage/permanent/chrome/idb
/3870112724rsegmnoittet-es.files
0      ./mozilla/firefox/dq48xt6q.default-release/storage/permanent/chrome/idb
/3561288849sdhlie.files
0      ./mozilla/firefox/dq48xt6q.default-release/storage/permanent/chrome/idb
```

Рис. 4.17: выполнение команды du

12. Воспользовавшись справкой команды find, выведем имена всех директорий, имеющих в вашем домашнем каталоге (рис. [4.18]) (рис. [4.19])

```
[aaliechtenstein@aaliechtenstein ~]$ find ~/ -type d -printf "%f\n"
```

Рис. 4.18: команда

```
heads
remotes
origin
.texlive2021
texmf-var
web2c
luahbtex
luatex-cache
generic
names
bin
.gnupg
reports
monthly
monthly
monthly
ski.places
equipment
plans
ex.3
australia
play
games
[aaliechtenstein@aaliechtenstein ~]$
```

Рис. 4.19: вывод команды

## 5 Выводы

В процессе работы мы ознакомились с инструментами поиска файлов и фильтрации текстовых данных. Были приобретены практические навыки по управлению процессами (и заданиями), по проверке использования диска и обслуживанию файловых систем.