

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

*дисциплина: Операционные системы*

Шилоносков Данил Вячеславович

# 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>19</b>

## Список иллюстраций

4.1	№1	12
4.2	№2	13
4.3	№3	13
4.4	№4	14
4.5	№5	14
4.6	№6	15
4.7	№7	15
4.8	№8	15
4.9	№9	15
4.10	№10	16
4.11	№11	17
4.12	№12	18

# 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`, которая пишет результат в файл `sorting_list`. 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])

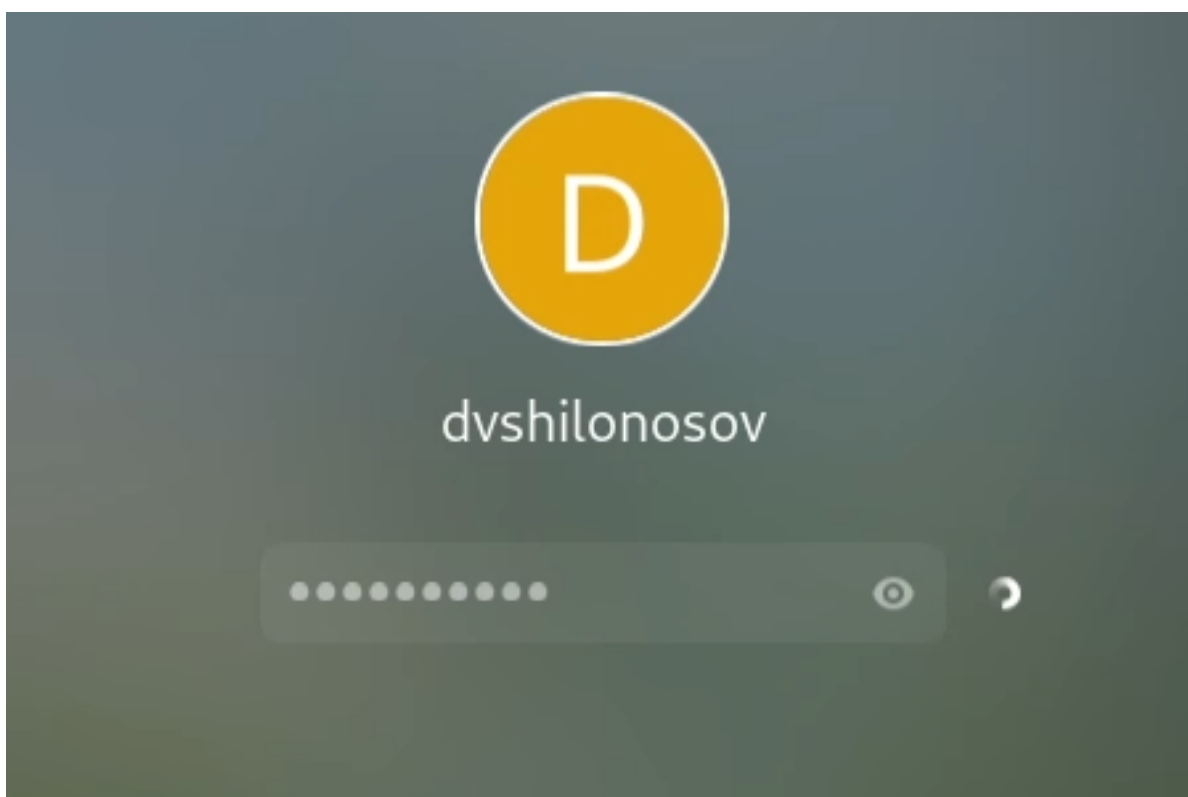


Рис. 4.1: №1

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])

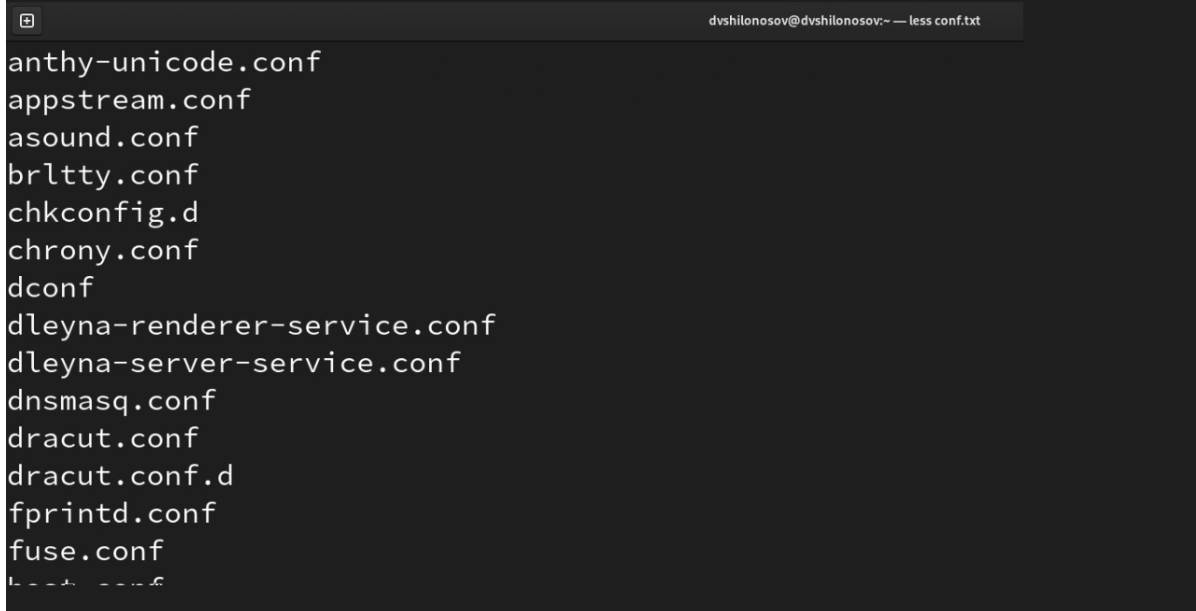
```
[dvshilonosov@dvshilonosov ~]$ ls /etc > file.txt
[dvshilonosov@dvshilonosov ~]$ ls ~ >> file.txt
```

Рис. 4.2: №2

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

```
[dvshilonosov@dvshilonosov ~]$ grep ".conf" file.txt
anthy-unicode.conf
appstream.conf
asound.conf
brltty.conf
chkconfig.d
chrony.conf
dconf
dley-na-renderer-service.conf

[dvshilonosov@dvshilonosov ~]$ grep ".conf" file.txt > conf.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
i-bus.conf
```

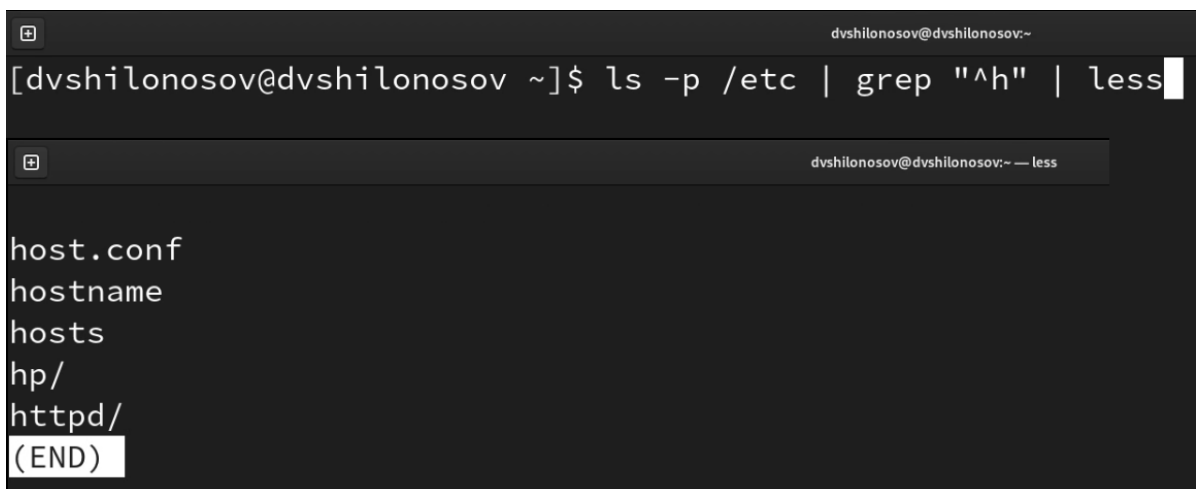
Рис. 4.3: №3

4. Определим, какие файлы в вашем домашнем каталоге имеют имена, начинавшиеся с символа c. Предложим несколько вариантов, как это сделать (рис. [4.4])

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

Рис. 4.4: №4

5. Выведем на экран (по странично) имена файлов из каталога /etc, начинающиеся с символа h (рис. [4.5])



```
dvshilonosov@dvshilonosov:~  
[dvshilonosov@dvshilonosov ~]$ ls -p /etc | grep "^h" | less  
host.conf  
hostname  
hosts  
hp/  
httpd/  
(END)
```

Рис. 4.5: №5

6. Запустим в фоновом режиме процесс, который будет записывать в файл ~/logfile файлы, имена которых начинаются с log (рис. [4.6])

```
[dvshilonosov@dvshilonosov ~]$ sudo find / -name "log*" > logfile.txt &  
[1] 6836
```

Рис. 4.6: №6

7. Удалим файл ~/logfile (рис. [4.7])

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

Рис. 4.7: №7

8. Запустим из консоли в фоновом режиме редактор gedit (рис. [4.8])

```
[dvshilonosov@dvshilonosov ~]$ gedit &  
[2] 6976
```

Рис. 4.8: №8

9. Определим идентификатор процесса gedit, используя команду ps, конвейер и фильтр grep. Узнаем, как ещё можно определить идентификатор процесса (рис. [4.9])

```
[dvshilonosov@dvshilonosov ~]$ ps | grep gedit  
6924 pts/0 00:00:00 gedit  
[2]+  Завершён gedit  
[dvshilonosov@dvshilonosov ~]$ pidof gedit  
6924
```

Рис. 4.9: №9

10. Прочтем справку (man) команды `kill`, после чего используем её для завершения процесса `gedit` (рис. [4.10])

```
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.

[dvshilonosov@dvshilonosov ~]$ ps aux | grep "gedit"
dvshilo+   7097  2.1  1.8 858108 74836 pts/1    Sl   17:43   0:00 gedit
dvshilo+   7123  0.0  0.0 222044  2376 pts/1    S+   17:43   0:00 grep --color=auto gedit
[dvshilonosov@dvshilonosov ~]$ man kill
[dvshilonosov@dvshilonosov ~]$ kill 7097
[dvshilonosov@dvshilonosov ~]$
```

Рис. 4.10: №10

11. Выполним команды `df` и `du`, предварительно получив более подробную информацию об этих командах, с помощью команды `man` (рис. [4.11])



```
DF(1)                                User Commands                                DF(1)

NAME
  I  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.

[dvshilonosov@dvshilonosov ~]$ df
Файловая система 1K-блоков  Использовано  Доступно  Использовано%  Смонтировано в
devtmpfs           4096          0      4096          0% /dev
tmpfs              2005856       0    2005856          0% /dev/shm
tmpfs              802344      1392    800952          1% /run
/dev/sda3          82834432    11490008  68313960         15% /
tmpfs              2005860       16    2005844          1% /tmp
/dev/sda3          82834432    11490008  68313960         15% /home
/dev/sda2          996780      296092    631876         32% /boot
tmpfs              401168       132    401036          1% /run/user/1000
/dev/sr0           62448      62448          0        100% /run/media/dvshilonosov/VBox_GAs
_6.1.42

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.

[dvshilonosov@dvshilonosov ~]$ du
16      ./work/dvshilonosov.github.io/.git/objects
4       ./work/dvshilonosov.github.io/.git/logs/refs/heads
4       ./work/dvshilonosov.github.io/.git/logs/refs/remotes/origin
4       ./work/dvshilonosov.github.io/.git/logs/refs/remotes
8       ./work/dvshilonosov.github.io/.git/logs/refs
12      ./work/dvshilonosov.github.io/.git/logs
120     ./work/dvshilonosov.github.io/.git
120     ./work/dvshilonosov.github.io
196404  ./work
54516   ./bin
3300    ./texlive2021/texmf-var/web2c/luahbtex
3300    ./texlive2021/texmf-var/web2c
1552    ./texlive2021/texmf-var/luatex-cache/generic/names
0       ./texlive2021/texmf-var/luatex-cache/generic/fonts/otl
0       ./texlive2021/texmf-var/luatex-cache/generic/fonts
1552    ./texlive2021/texmf-var/luatex-cache/generic
```

Рис. 4.11: №11

12. Воспользовавшись справкой команды `find`, выведем имена всех директорий, имеющихся в вашем домашнем каталоге (рис. [4.12])

```
[dvshilonosov@dvshilonosov ~]$ find ~/ -type d -printf "%f\n"
demo
page
1
dvshilonosov.github.io
.git
branches
hooks
info
refs
heads
tags
remotes
origin
objects
pack
info
```

Рис. 4.12: №12

## 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.