

# **Отчёт по лабораторной работе №8**

**Операционные системы**

Ирина Васильевна Панявкина

# 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>7</b>
<b>4</b>	<b>Выполнение лабораторной работы</b>	<b>8</b>
<b>5</b>	<b>Выводы</b>	<b>15</b>
<b>6</b>	<b>Ответы на контрольные вопросы</b>	<b>16</b>
	<b>Список литературы</b>	<b>19</b>

# Список иллюстраций

4.1	Запись в файл . . . . .	8
4.2	Вывод содержимого файла . . . . .	8
4.3	Добавление данных в файл . . . . .	9
4.4	Поиск файлов определенного расширения . . . . .	9
4.5	Запись в файл . . . . .	9
4.6	Поиск файлов, начинающихся с определенного элемента . . . . .	10
4.7	Поиск файлов, начинающихся с определенного элемента . . . . .	10
4.8	Поиск файлов, начинающихся с определенного элемента . . . . .	11
4.9	Создание фонового процесса . . . . .	11
4.10	Удаление файла . . . . .	11
4.11	Создание фонового процесса . . . . .	12
4.12	Поиск идентификатора процесса . . . . .	12
4.13	Чтение документации . . . . .	12
4.14	Удаление процесса . . . . .	12
4.15	Чтение документации . . . . .	13
4.16	Утилита df . . . . .	13
4.17	Утилита du . . . . .	13
4.18	Чтение документации . . . . .	14
4.19	Вывод имен всех директорий . . . . .	14

## **Список таблиц**

# 1 Цель работы

Цель данной лабораторной работы - ознакомление с инструментами поиска файлов и фильтрации текстовых данных, а также приобретение практических навыков по управлению процессами (и заданиями), по проверке использования диска и по 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, 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

Я вошла в систему под соответствующим именем пользователя, открыла терминал, записала в файл file.txt названия файлов из каталога /etc с помощью перенаправления “>” (и файл создала, и записала в него то, что могло быть выведено ls -lR /etc). В файл я добавила также все файлы из подкаталогов (рис. 4.1).

```
[root@irinapanyavkina ~]# ls -lR /etc > file.txt
```

Рис. 4.1: Запись в файл

Проверила, что в файл записались нужные значения с помощью утилиты head, она выводит первые 10 строк файла на экран (рис. 4.2).

```
[root@irinapanyavkina ~]# head file.txt
/etc:
итого 1372
drwxr-xr-x. 1 root root    126 ноя 20 03:00 abrt
-rw-r--r--. 1 root root    16 мар  2 15:36 adjtime
-rw-r--r--. 1 root root  1529 дек 27 03:00 aliases
drwxr-xr-x. 1 root root    70 ноя 14 03:00 alsa
drwxr-xr-x. 1 root root  1386 мар  2 22:25 alternatives
drwxr-xr-x. 1 root root    56 окт 24 17:50 anaconda
-rw-r--r--. 1 root root    541 июл 17 2024 anacrontab
-rw-r--r--. 1 root root    55 ноя 15 03:00 asound.conf
```

Рис. 4.2: Вывод содержимого файла

Добавила в созданный файл имена файлов из домашнего каталога, используя перенаправление “>” в режиме добавления (рис. 4.3).



```
[root@irinapanyavkina ~]# ls -lR ~/ >> file.txt
[root@irinapanyavkina ~]#
```

Рис. 4.3: Добавление данных в файл

Вывела на экран имена всех файлов, имеющих расширение “.conf” с помощью утилиты grep (рис. 4.4).

```
[root@irinapanyavkina ~]# grep .conf file.txt
-rw-r--r--. 1 root root    55 ноя 15 03:00 asound.conf
-rw-r--r--. 1 root root  1381 окт  8 03:00 chrony.conf
drwxr-xr-x. 1 root root    18 окт 24 17:49 dconf
-rw-r--r--. 1 root root 28602 дек 27 03:00 dnsmasq.conf
-rw-r--r--. 1 root root   117 ноя 27 03:00 dracut.conf
drwxr-xr-x. 1 root root     0 ноя 27 03:00 dracut.conf.d
-rw-r--r--. 1 root root    20 сен  5 2024 fprintd.conf
-rw-r--r--. 1 root root    38 июл 17 2024 fuse.conf
-rw-r--r--. 1 root root     9 дек 27 03:00 host.conf
-rw-r--r--. 1 root root  5799 фев  4 03:00 idmapd.conf
-rw-r--r--. 1 root root   989 фев 21 2024 imv_config
-rw-r--r--. 1 root root  1670 июл 18 2024 ipsec.conf
-rw-r--r--. 1 root root  8782 мар  2 17:26 kdump.conf
-rw-r--r--. 1 root root   880 фев 11 03:00 krb5.conf
```

Рис. 4.4: Поиск файлов определенного расширения

Добавила вывод прошлой команды в новый файл conf.txt с помощью перенаправления “>” (файл создается при выполнении этой команды) (рис. 4.5).

```
[root@irinapanyavkina ~]# grep .conf file.txt > conf.txt
```

Рис. 4.5: Запись в файл

Определяю, какие файлы в домашнем каталоге начинаются с символа “с” с помощью утилиты find, прописываю ей в аргументах домашнюю директорию (тогда вывод относительно корневого каталога, а не домашнего будет), выбираю опцию -name (ищем по имени), и пишу маску, по которой будем искать имя, где \* - любое кол-во любых символов, добавляю опцию -print, чтобы мне вывелся

результат (рис. 4.6). Но таким образом я получаю информацию даже о файлах из подкаталогов домашнего каталога.

```
[root@irinapanyavkina ~]# find ~ -name "c*" -print
/root/.cache/pip/http-v2/4/c
/root/.cache/pip/http-v2/d/c
/root/.cache/pip/http-v2/1/2/8/c
/root/.cache/pip/http-v2/c
/root/.cache/pip/http-v2/c/3/c
/root/.cache/pip/http-v2/c/3/c/c
/root/.cache/pip/http-v2/c/3/c/c/2/c3cc2d1866f223b42419def375d9ba1574137ebe41f06199530ead86
/root/.cache/pip/http-v2/c/3/c/c/2/c3cc2d1866f223b42419def375d9ba1574137ebe41f06199530ead86.body
/root/.cache/pip/http-v2/c/1/4/2/c
/root/.cache/pip/http-v2/c/1/4/2/c/c142c005888845cbebed2978119a034310bc68a893cf35461eb22589
/root/.cache/pip/http-v2/c/1/4/2/c/c142c005888845cbebed2978119a034310bc68a893cf35461eb22589.body
/root/.cache/pip/http-v2/a/1/3/e/c
/root/.local/lib/python3.13/site-packages/pandocxnos/core.py
/root/.local/lib/python3.13/site-packages/pandocxnos/__pycache__/core.cpython-313.pyc
/root/snap/telegram-desktop/common
/root/snap/telegram-desktop/current
/root/conf.txt
```

Рис. 4.6: Поиск файлов, начинающихся с определенного элемента

Второй способ использовать утилиту `ls -lR` и использовать `grep`, чтобы найти элементы с первым символом `c`. Однако этот способ не работает для поиска файлов из подкаталогов каталога (рис. 4.7).

```
[root@irinapanyavkina ~]# ls -lR | grep c*
-rw-r--r-- 1 root root 42440 апр 3 19:48 conf.txt
```

Рис. 4.7: Поиск файлов, начинающихся с определенного элемента

С помощью метода `find`, чьи опции я расписала ранее, ищу все файлы, начинающиеся с буквы `h` (рис. 4.8).

```
[root@irinapanyavkina ~]# find /etc -name "h*" -print
/etc/avahi/hosts
/etc/firewalld/helpers
/etc/libibverbs.d/hfiverbs.driver
/etc/libibverbs.d/hns.driver
/etc/nvme/hostnqn
/etc/nvme/hostid
/etc/udev/hwdb.d
/etc/udev/hwdb.bin
/etc/host.conf
/etc/hosts
/etc/hostname
```

Рис. 4.8: Поиск файлов, начинающихся с определенного элемента

Запускаю в фоновом режиме (на это указывает символ &) процесс, который будет записывать в файл logfile (с помощью перенаправления >) файлы, имена которых начинаются с log (рис. 4.9).

```
[root@irinapanyavkina ~]# find ~ -name "log*" -print > logfile &
[1] 60427
```

Рис. 4.9: Создание фонового процесса

Проверяю, что файл создан, удаляю его, проверяю, что файл удален (рис. 4.10).

```
[root@irinapanyavkina ~]# ls
anaconda-ks.cfg  conf.txt  file.txt  logfile  snap
[1]+  Завершён      find ~ -name "log*" -print > logfile
[root@irinapanyavkina ~]# rm logfile
rm: удалить обычный файл 'logfile'? y
[root@irinapanyavkina ~]# ls
anaconda-ks.cfg  conf.txt  file.txt  snap
```

Рис. 4.10: Удаление файла

Запускаю в консоли в фоновом режиме (с помощью символа &) редактор mousepad, потому что редактора gedit у меня, к сожалению нет, но работают они идентично (рис. 4.11).

```
[irinapanyavkina@irinapanyavkina ~]$ mousepad &
[1] 60765
```

Рис. 4.11: Создание фонового процесса

С помощью утилиты ps определяю идентификатор процесса mousepad, его значение 60765 (рис. 4.12). Также мы можем определить идентификатор с помощью pgrep.

```
[irinapanyavkina@irinapanyavkina ~]$ ps aux | grep mousepad
irinapa+  60765  2.3  0.8 765148 53508 pts/2    Sl   20:15   0:01 mousepad
irinapa+  60787  0.0  0.0 230496 2464 pts/2    S+   20:16   0:00 grep --color=auto mousepad
[irinapanyavkina@irinapanyavkina ~]$ pgrep mousepad
60765
[irinapanyavkina@irinapanyavkina ~]$ ps aux | grep mousepad | grep -v grep
irinapa+  60765  1.2  0.8 765148 53508 pts/2    Sl   20:15   0:01 mousepad
```

Рис. 4.12: Поиск идентификатора процесса

Прочитала справку команды kill (рис. 4.13).

```
kill(1) User Commands
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 performed clean-up steps before terminating in an orderly fashion. If a process does not terminate after a TERM signal has been sent, then 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

    If signal is 0, then no actual signal is sent, but error checking is still performed.
```

Рис. 4.13: Чтение документации

Использую команду kill и идентификатор процесса, чтобы его удалить (рис. 4.14). Заметила, как у меня закрылась программа mousepad.

```
[irinapanyavkina@irinapanyavkina ~]$ man kill
[irinapanyavkina@irinapanyavkina ~]$ kill 60765
[irinapanyavkina@irinapanyavkina ~]$
```

Рис. 4.14: Удаление процесса

Прочитала документацию про функции df и du (рис. 4.15).

```
[irinapanyavkina@irinapanyavkina ~]$ man df
[1]+  Завершён      mousepad
[irinapanyavkina@irinapanyavkina ~]$ man du
```

Рис. 4.15: Чтение документации

Используя утилиту `df` опции `-iv` позволяют увидеть информацию об инодах и сделать вывод читаемым, игнорируя сообщения системы о нем (рис. 4.16). Эта утилита нам нужна, чтобы выяснить, сколько свободного места есть у нашей системы.

```
[irinapanyavkina@irinapanyavkina ~]$ df -iv
Файловая система  Инодов  ИИспользовано  ИСвободно  ИИспользовано%  Смонтировано в
/dev/sda3          0          0          0          0%              /
devtmpfs           752715      566      752149      1% /dev
tmpfs              757058      367      756691      1% /dev/shm
tmpfs             819200     1219     817981      1% /run
tmpfs              1024         2       1022      1% /run/credentials/systemd-journald.service
tmpfs              1024         2       1022      1% /run/credentials/systemd-udev-load-credentials.service
tmpfs              1024         2       1022      1% /run/credentials/systemd-network-generator.service
tmpfs              1024         2       1022      1% /run/credentials/systemd-sysctl.service
tmpfs              1024         2       1022      1% /run/credentials/systemd-tmpfiles-setup-dev-early.service
tmpfs              1024         2       1022      1% /run/credentials/systemd-tmpfiles-setup-dev.service
tmpfs            1048576         54    1048522      1% /tmp
/dev/sda3          0          0          0          0%              /home
tmpfs              1024         2       1022      1% /run/credentials/systemd-vconsole-setup.service
/dev/sda2          65536       398     65138      1% /boot
tmpfs              1024         2       1022      1% /run/credentials/systemd-tmpfiles-setup.service
tmpfs              1024         2       1022      1% /run/credentials/systemd-resolved.service
tmpfs            151411       166     151245      1% /run/user/1000
/dev/sr0           0          0          0          0%              - /run/media/irinapanyavkina/VBox_GAs_7.1.2
Fedora_sway_shared 1000     -999000    1000000      - /mnt/shared
/dev/loop0         608         608         0     100% /var/lib/napd/snap/snapd/23545
/dev/loop1        10615       10615         0     100% /var/lib/napd/snap/core24/739
/dev/loop2         29          29          0     100% /var/lib/napd/snap/bare/5
/dev/loop3        76208       76208         0     100% /var/lib/napd/snap/gtk-common-themes/1535
/dev/loop4        28964       28964         0     100% /var/lib/napd/snap/telegram-desktop/6529
/dev/loop5        28903       28903         0     100% /var/lib/napd/snap/telegram-desktop/6597
[irinapanyavkina@irinapanyavkina ~]$
```

Рис. 4.16: Утилита `df`

Используя утилиту `du`. Она нужна чтобы просмотреть, сколько места занимают файлы в определенной директории и найти самые большие из них (рис. 4.17). Утилита `-a` позволит увидеть размер всех файлов, а не только директорий.

```
[irinapanyavkina@irinapanyavkina ~]$ du -a work/study/2024-2025/Операционные системы/os-intro/labs/lab03/presentation/
0  work/study/2024-2025/Операционные системы/os-intro/labs/lab03/presentation/.projectile
0  work/study/2024-2025/Операционные системы/os-intro/labs/lab03/presentation/.texlabroot
4  work/study/2024-2025/Операционные системы/os-intro/labs/lab03/presentation/Makefile
24 work/study/2024-2025/Операционные системы/os-intro/labs/lab03/presentation/image/lab3.1.png
24 work/study/2024-2025/Операционные системы/os-intro/labs/lab03/presentation/image/lab3.2.png
156 work/study/2024-2025/Операционные системы/os-intro/labs/lab03/presentation/image/lab3.3.png
96 work/study/2024-2025/Операционные системы/os-intro/labs/lab03/presentation/image/lab3.4.png
48 work/study/2024-2025/Операционные системы/os-intro/labs/lab03/presentation/image/lab3.5.png
24 work/study/2024-2025/Операционные системы/os-intro/labs/lab03/presentation/image/lab3.6.png
60 work/study/2024-2025/Операционные системы/os-intro/labs/lab03/presentation/image/lab3.7.png
76 work/study/2024-2025/Операционные системы/os-intro/labs/lab03/presentation/image/lab3.8.png
508 work/study/2024-2025/Операционные системы/os-intro/labs/lab03/presentation/image
8  work/study/2024-2025/Операционные системы/os-intro/labs/lab03/presentation/presentation.md
364 work/study/2024-2025/Операционные системы/os-intro/labs/lab03/presentation/presentation.pdf
1396 work/study/2024-2025/Операционные системы/os-intro/labs/lab03/presentation/presentation.html
2280 work/study/2024-2025/Операционные системы/os-intro/labs/lab03/presentation/
```

Рис. 4.17: Утилита `du`

Прочитала документацию о команде `find` (рис. 4.18).

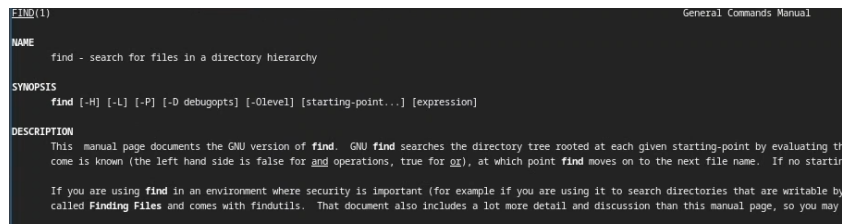


Рис. 4.18: Чтение документации

Вывела имена всех директорий, имеющиххся в моем домашнем каталоге, используя аргумент `d` у утилиты `find` опции `-type`, то есть указываю тип файлов, который мне нужен и этот тип Директория (рис. 4.19).

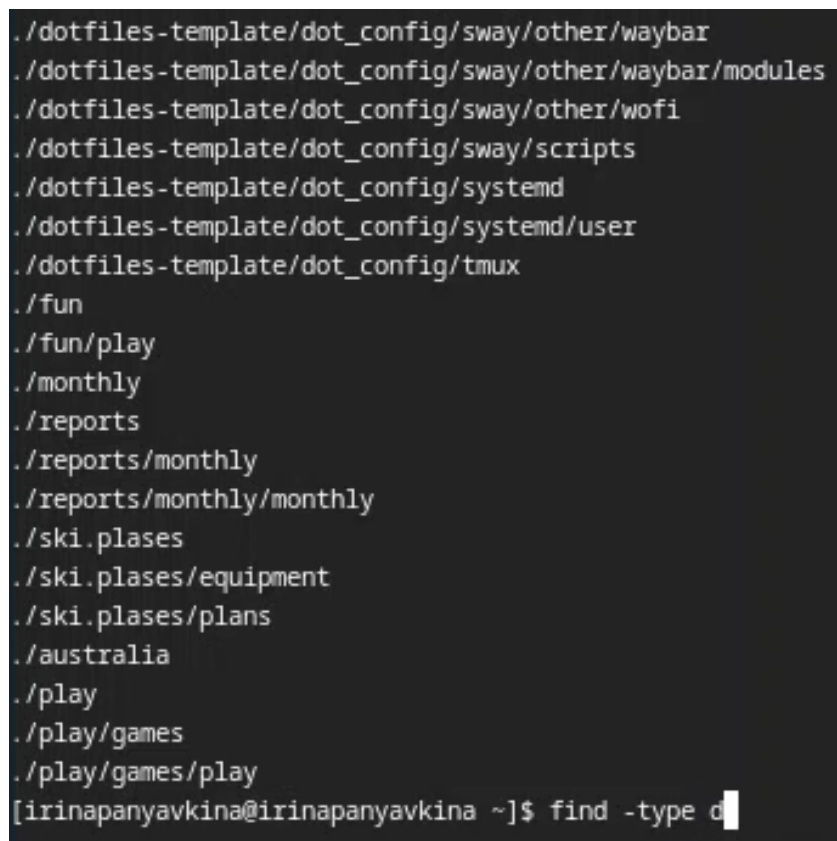


Рис. 4.19: Вывод имен всех директорий

## **5 Выводы**

В результате данной лабораторной работы я ознакомилась с инструментами поиска файлов и фильтрации текстовых данных, а также приобрела практические навыки по управлению процессами (и заданиями), по проверке использования диска и по обслуживанию файловых систем.

## 6 Ответы на контрольные вопросы

1. Какие потоки ввода вывода вы знаете? 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.
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: find /etc -name "p\*" -print

9. Можно ли по контексту (содержанию) найти файл? Если да, то как? `find / -type f -exec grep -H 'текстДляПоиска' {} ;`
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`.

## Список литературы

1. Лабораторная работа №8 [Электронный ресурс] URL: <https://esystem.rudn.ru/mod/resource/>