

# **Лабораторная работа №7**

Дисциплина: Операционные системы

Кочуров Арсений Владимирович

# Содержание

|   |                                |    |
|---|--------------------------------|----|
| 1 | Цель работы                    | 5  |
| 2 | Задание                        | 6  |
| 3 | Выполнение лабораторной работы | 7  |
| 4 | Выводы                         | 23 |

## List of Tables

# List of Figures

|      |  |    |
|------|--|----|
| 3.1  | Запись файлов . . . . .                                    | 7  |
| 3.2  | Файлы в файле file.txt . . . . .                           | 8  |
| 3.3  | Вывожу имена файлов . . . . .                              | 8  |
| 3.4  | Файлы с расширением .conf . . . . .                        | 9  |
| 3.5  | Создание файлов . . . . .                                  | 9  |
| 3.6  | команда «find /etc –maxdepth1 –name “h *   less» . . . . . | 9  |
| 3.7  | список файлов .....  | 10 |
| 3.8  | Фоновый режим процесса .....                               | 11 |
| 3.9  | Фоновый режим процесса .....                               | 11 |
| 3.10 | Удаление файла.....  | 12 |
| 3.11 | Удаление файла с помощью команды rm .....                  | 12 |
| 3.12 | Редактор gedit .....                                       | 12 |
| 3.13 | команда «ps  grep-i“gedit”» .....                          | 13 |
| 3.14 | команда kill.....  | 13 |
| 3.15 | информация по команде kill .....                           | 13 |
| 3.16 | завершение процесса gedit.....                             | 14 |
| 3.17 | команды df, du .....                                       | 15 |
| 3.18 | информация команды df .....                                | 15 |
| 3.19 | информация команды du.....                                 | 16 |
| 3.20 | команда df в консоли .....                                 | 16 |
| 3.21 | команда du в консоли.....                                  | 17 |
| 3.22 | Информация команды find.....                               | 17 |
| 3.23 | имена всех директорий.....                                 | 18 |

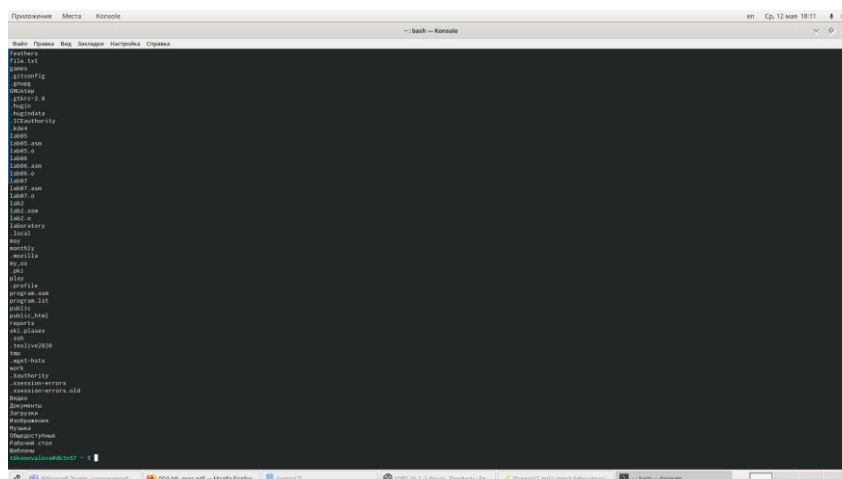
# 1 Цель работы

Цель данной лабораторной работы — Ознакомление с инструментами поиска файлов и фильтрации текстовых данных. Приобретение практических навыков: по управлению процессами (и задания-ми), по проверке использования диска и обслуживанию файловых систем.

## 2 Задание

1. Сделать отчёт по лабораторной работе №7 в формате Markdown.
2. Ознакомление с инструментами поиска файлов и фильтрации текстовых данных.





3). Вывел имена всех файлов из file.txt, имеющих расширение .conf и записал их в новый текстовый файл conf.txt, используя команду «grep-e‘.conf\$’file.txt>conf.txt». С помощью команды «catconf.txt» проверяю правильность выполненных действий (иллюстрация на рис. 3.3 , 3.4).

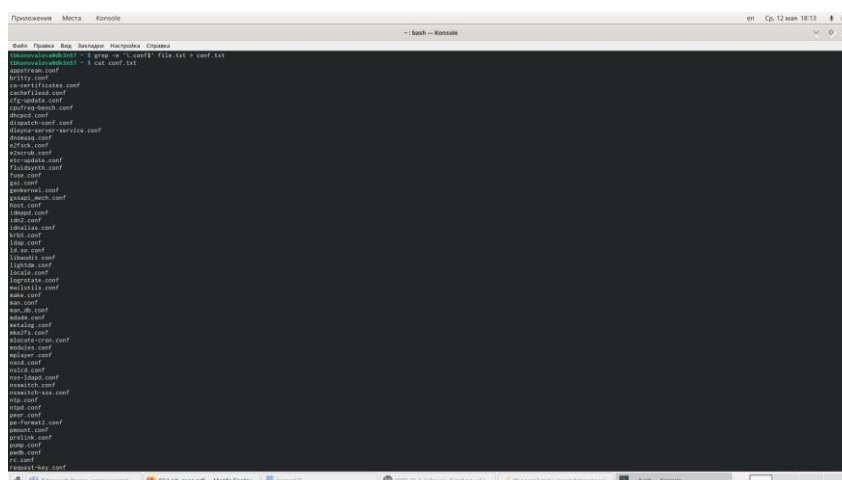


Figure 3.3: Вывожу имена файлов



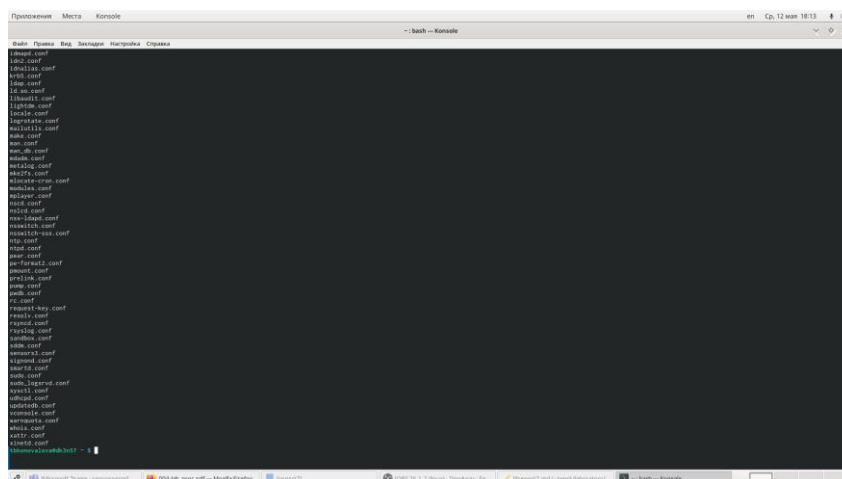


Figure 3.4: Файлы с расширением .conf

4). Определить, какие файлы в домашнем каталоге имеют имена, начинающиеся с символа с, можно несколькими командами: - «find ~ -maxdepth 1 -name "с" -print» (опция *maxdepth 1* необходима для того, чтобы файлы находились строго только в домашнем каталоге); - «ls~/с»; - «ls-a ~ | grep с\*» (см. рис. 3.5).

```
tbkonovalova@dk3n57 ~ $ find ~ -maxdepth 1 -name "с*" -print
find: Для -maxdepth ожидалось целое положительное значение аргумента, а получен '1'
tbkonovalova@dk3n57 ~ $ find ~ -maxdepth 1 -name "с*" -print
/afs/.dk.sci.pfu.edu.ru/home/t/b/tbkonovalova/conf.txt
tbkonovalova@dk3n57 ~ $ ls ~/с*
/afs/.dk.sci.pfu.edu.ru/home/t/b/tbkonovalova/conf.txt
tbkonovalova@dk3n57 ~ $ ls -a ~ | grep с*
conf.txt
tbkonovalova@dk3n57 ~ $
```

Figure 3.5: Создание файлов

5). Чтобы вывести на экран (постранично) имена файлов из каталога /etc, начинающиеся с символа h, я использовала команду «find /etc -maxdepth 1 -name "h\*" | less» (Скриншоты 3.6 , 3.7).

```
tbkonovalova@dk3n57 ~ $ find /etc -maxdepth 1 -name "h*" | less
tbkonovalova@dk3n57 ~ $
```

Figure 3.6: команда «find /etc -maxdepth 1 -name "h\*" | less»

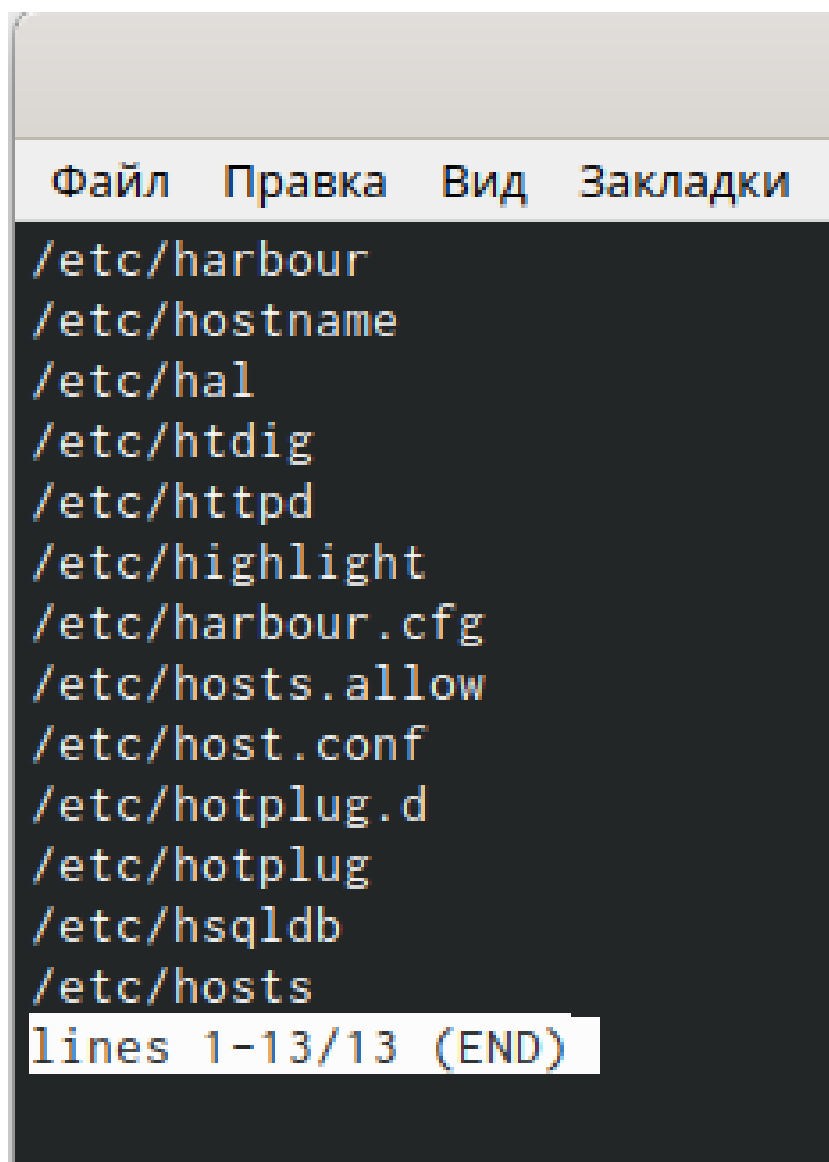


Figure 3.7: список файлов

6). Запускаю в фоновом режиме процесс, который будет записывать в файл ~/logfile файлы, имена которых начинаются с log, используя команду «find/ -name“log\*” > logfile&» (Скриншоты 3.8 , 3.9). Так как в фоновом режиме у меня запустился непрерывный процесс записывания файла, я сделал скриншоты некоторых частей работы процесса. Командой «catlogfile» проверяю выполненные действия (Рисунок 3.10). Далее удалила файл ~/logfile с помощью команды «rm logfile», у меня данный файл уже был удалён, поэтому программа вывела

сообщение об этом (Рисунок 3.11).

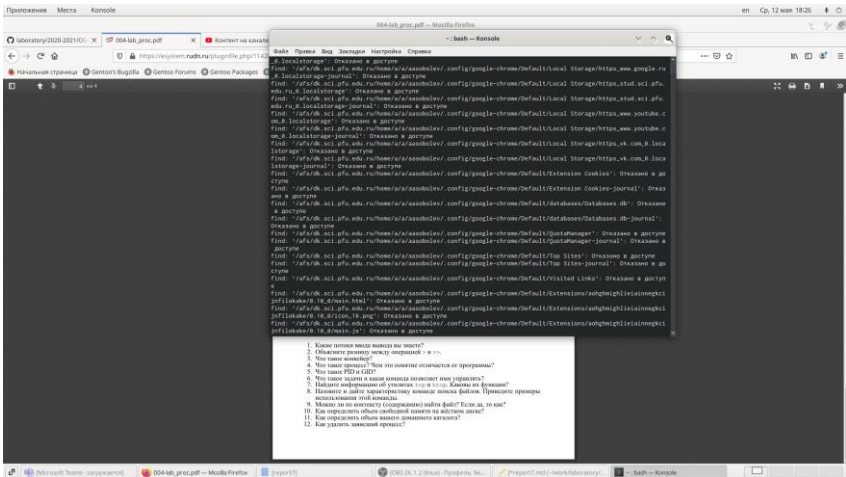


Figure 3.8: Фоновый режим процесса

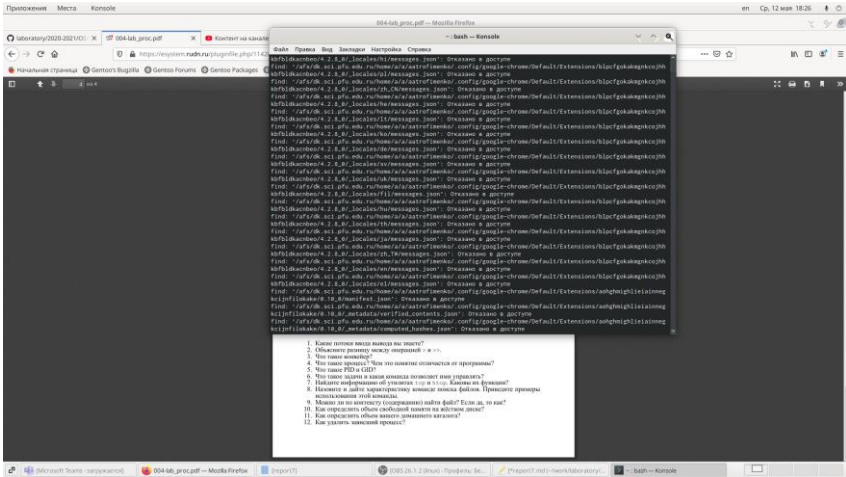
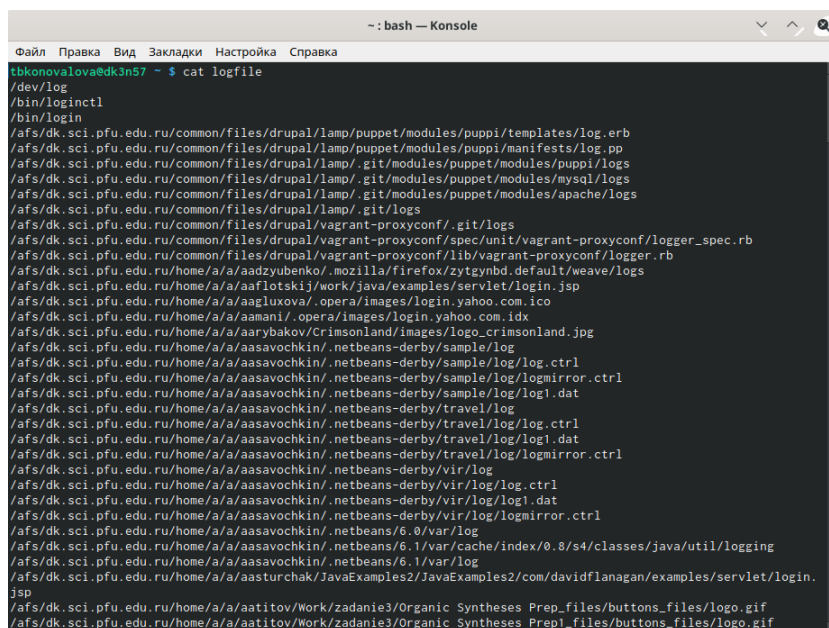
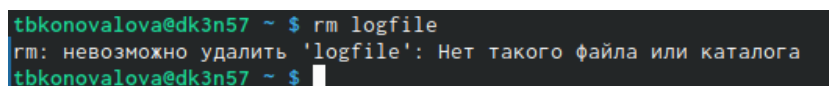


Figure 3.9: Фоновый режим процесса



```
tbkonovalova@dk3n57 ~ $ cat logfile
/dev/log
/bin/loginctl
/bin/login
/afs/dk.sci.pfu.edu.ru/common/files/drupal/lamp/puppet/modules/puppi/templates/log.erb
/afs/dk.sci.pfu.edu.ru/common/files/drupal/lamp/puppet/modules/puppi/manifests/log.pp
/afs/dk.sci.pfu.edu.ru/common/files/drupal/lamp/.git/modules/puppet/modules/puppi/logs
/afs/dk.sci.pfu.edu.ru/common/files/drupal/lamp/.git/modules/puppet/modules/mysql/logs
/afs/dk.sci.pfu.edu.ru/common/files/drupal/lamp/.git/modules/puppet/modules/apache/logs
/afs/dk.sci.pfu.edu.ru/common/files/drupal/lamp/.git/logs
/afs/dk.sci.pfu.edu.ru/common/files/drupal/vagrant-proxyconf/.git/logs
/afs/dk.sci.pfu.edu.ru/common/files/drupal/vagrant-proxyconf/spec/unit/vagrant-proxyconf/logger_spec.rb
/afs/dk.sci.pfu.edu.ru/common/files/drupal/vagrant-proxyconf/lib/vagrant-proxyconf/logger.rb
/afs/dk.sci.pfu.edu.ru/home/a/a/aadzyubenko/.mozilla/firefox/zytzynd.default/weave/logs
/afs/dk.sci.pfu.edu.ru/home/a/a/aaflotskij/work/java/examples/servlet/login.jsp
/afs/dk.sci.pfu.edu.ru/home/a/a/aagluxova/.opera/images/login.yahoo.com.ico
/afs/dk.sci.pfu.edu.ru/home/a/a/aamani/.opera/images/login.yahoo.com.idx
/afs/dk.sci.pfu.edu.ru/home/a/a/aarybakov/Crimsonland/images/logo_crimsonland.jpg
/afs/dk.sci.pfu.edu.ru/home/a/a/aasavochkin/.netbeans-derby/sample/log
/afs/dk.sci.pfu.edu.ru/home/a/a/aasavochkin/.netbeans-derby/sample/log/log.ctrl
/afs/dk.sci.pfu.edu.ru/home/a/a/aasavochkin/.netbeans-derby/sample/log/logmirror.ctrl
/afs/dk.sci.pfu.edu.ru/home/a/a/aasavochkin/.netbeans-derby/sample/log/log1.dat
/afs/dk.sci.pfu.edu.ru/home/a/a/aasavochkin/.netbeans-derby/travel/log
/afs/dk.sci.pfu.edu.ru/home/a/a/aasavochkin/.netbeans-derby/travel/log/log.ctrl
/afs/dk.sci.pfu.edu.ru/home/a/a/aasavochkin/.netbeans-derby/travel/log/log1.dat
/afs/dk.sci.pfu.edu.ru/home/a/a/aasavochkin/.netbeans-derby/travel/log/logmirror.ctrl
/afs/dk.sci.pfu.edu.ru/home/a/a/aasavochkin/.netbeans-derby/vir/log
/afs/dk.sci.pfu.edu.ru/home/a/a/aasavochkin/.netbeans-derby/vir/log/log.ctrl
/afs/dk.sci.pfu.edu.ru/home/a/a/aasavochkin/.netbeans-derby/vir/log/log1.dat
/afs/dk.sci.pfu.edu.ru/home/a/a/aasavochkin/.netbeans-derby/vir/log/logmirror.ctrl
/afs/dk.sci.pfu.edu.ru/home/a/a/aasavochkin/.netbeans/6.0/var/log
/afs/dk.sci.pfu.edu.ru/home/a/a/aasavochkin/.netbeans/6.1/var/cache/index/0.8/s4/classes/java/util/logging
/afs/dk.sci.pfu.edu.ru/home/a/a/aasavochkin/.netbeans/6.1/var/log
/afs/dk.sci.pfu.edu.ru/home/a/a/aasturchak/JavaExamples2/JavaExamples2/com/davidflanagan/examples/servlet/login.jsp
/afs/dk.sci.pfu.edu.ru/home/a/a/aatitov/Work/zadanie3/Organic Syntheses Prep_files/buttons_files/logo.gif
/afs/dk.sci.pfu.edu.ru/home/a/a/aatitov/Work/zadanie3/Organic Syntheses Prep1_files/buttons_files/logo.gif
```

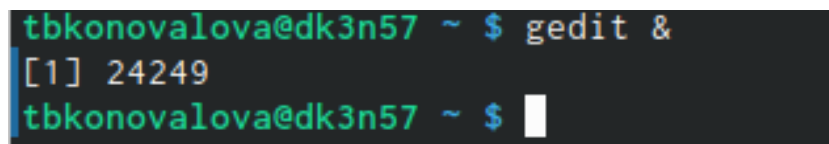
Figure 3.10: Удаление файла



```
tbkonovalova@dk3n57 ~ $ rm logfile
rm: невозможно удалить 'logfile': Нет такого файла или каталога
tbkonovalova@dk3n57 ~ $
```

Figure 3.11: Удаление файла с помощью команды rm

- 7). Смотрите пункт 6.
- 8). Запускаю редактор gedit в фоновом режиме командой «gedit &» (Рисунок 3.12). После этого на экране появляется окно редактора.



```
tbkonovalova@dk3n57 ~ $ gedit &
[1] 24249
tbkonovalova@dk3n57 ~ $
```

Figure 3.12: Редактор gedit

- 9). Чтобы определить идентификатор процесса gedit, использую команду «ps | grep -i "gedit"» (Скриншот 3.13). Из рисунка видно, что наш процесс имеет PID 24249. Узнать идентификатор процесса можно также, используя команду «pgrep gedit» или «pidof gedit».

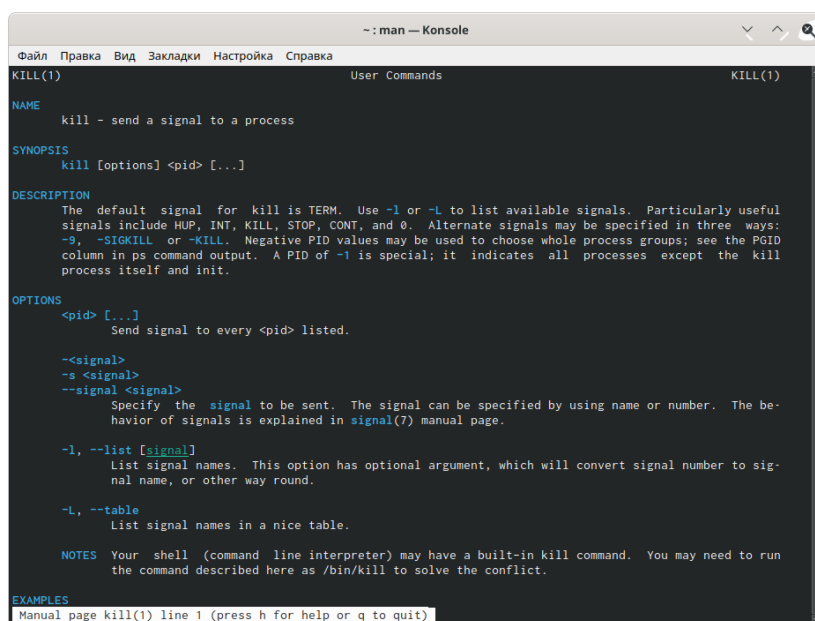
```
tbkonovalova@dk3n57 ~ $ gedit &
[1] 24249
tbkonovalova@dk3n57 ~ $ ps | grep -i "gedit"
[1]+  Завершён      gedit
tbkonovalova@dk3n57 ~ $
```

Figure 3.13: команда «ps| grep-i“gedit”»

10). Прочитав информацию о команде kill с помощью команды «man kill», использую её для завершения процесса gedit (команда «kill 24249») (Алгоритм действий представлен на рис. 3.14 , 3.15 , 3.16).

```
tbkonovalova@dk3n57 ~ $ man kill
tbkonovalova@dk3n57 ~ $
```

Figure 3.14: команда kill



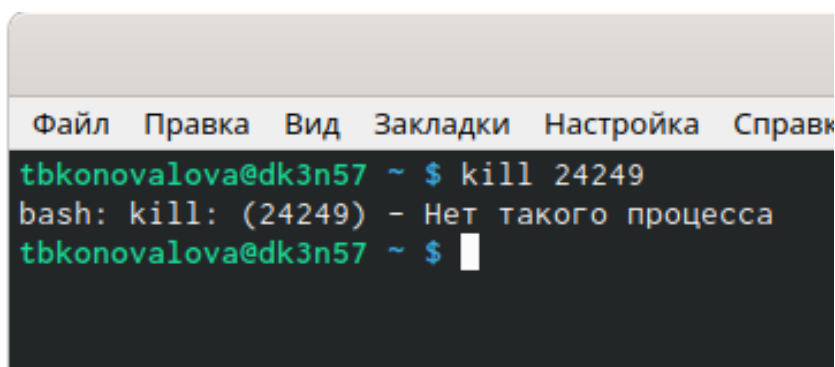
```
~: man — Konsole
Файл  Правка  Вид  Закладки  Настройка  Справка
KILL(1)                                User Commands                                KILL(1)
NAME
    kill - send a signal to a process
SYNOPSIS
    kill [options] <pid> [...]
DESCRIPTION
    The default signal for kill is TERM. Use -l or -L to list available signals. Particularly useful
    signals include HUP, INT, KILL, STOP, CONT, and 0. Alternate signals may be specified in three ways:
    -9, -SIGKILL or -KILL. Negative PID values may be used to choose whole process groups; see the PGID
    column in ps command output. A PID of -1 is special; it indicates all processes except the kill
    process itself and init.
OPTIONS
    <pid> [...]
        Send signal to every <pid> listed.

    -<signal>
    -s <signal>
    --signal <signal>
        Specify the signal to be sent. The signal can be specified by using name or number. The be-
        havior of signals is explained in signal(7) manual page.

    -l, --list [signal]
        List signal names. This option has optional argument, which will convert signal number to sig-
        nal name, or other way round.

    -L, --table
        List signal names in a nice table.
NOTES
    Your shell (command line interpreter) may have a built-in kill command. You may need to run
    the command described here as /bin/kill to solve the conflict.
EXAMPLES
    Manual page kill(1) line 1 (press h for help or q to quit)
```

Figure 3.15: информация по команде kill

A screenshot of a terminal window with a menu bar at the top containing 'Файл', 'Правка', 'Вид', 'Закладки', 'Настройка', and 'Справка'. The terminal shows a user prompt 'tbkonovalova@dk3n57 ~ \$' followed by the command 'kill 24249'. The next line shows the system response: 'bash: kill: (24249) - Нет такого процесса'. The prompt returns to 'tbkonovalova@dk3n57 ~ \$' with a cursor.

```

Файл  Правка  Вид  Закладки  Настройка  Справка
tbkonovalova@dk3n57 ~ $ kill 24249
bash: kill: (24249) - Нет такого процесса
tbkonovalova@dk3n57 ~ $

```

Figure 3.16: завершение процесса gedit

11). С помощью команд «mandf» и «mandu» узнаю информацию по необходимым командам и далее использую их (Рисунки 3.17 , 3.18 , 3.19 , 3.20 , 3.21). df – утилита, показывающая список всех файловых систем по именам устройств, сообщает их размер, занятое и свободное пространство и точки монтирования. Синтаксис: df[опции] устройств. du – утилита, предназначенная для вывода информации об объеме дискового пространства, занятого файлами и директориями. Она принимает путь к элементу файловой системы и выводит информацию о количестве байт дискового пространства или блоков диска, задействованных для его хранения. Синтаксис: du [опции] каталог\_или\_файл

```
~: bash — Konsole
Файл Правка Вид Закладки Настройка Справка
/afs/.dk.sci.pfu.edu.ru/home/t/b/tbkonovalova/laboratory/.git/objects/54
/afs/.dk.sci.pfu.edu.ru/home/t/b/tbkonovalova/laboratory/.git/objects/90
/afs/.dk.sci.pfu.edu.ru/home/t/b/tbkonovalova/laboratory/.git/objects/5b
/afs/.dk.sci.pfu.edu.ru/home/t/b/tbkonovalova/laboratory/.git/objects/1b
/afs/.dk.sci.pfu.edu.ru/home/t/b/tbkonovalova/laboratory/.git/logs
/afs/.dk.sci.pfu.edu.ru/home/t/b/tbkonovalova/laboratory/.git/logs/refs
/afs/.dk.sci.pfu.edu.ru/home/t/b/tbkonovalova/laboratory/.git/logs/refs/remotes
/afs/.dk.sci.pfu.edu.ru/home/t/b/tbkonovalova/laboratory/.git/logs/refs/remotes/origin
/afs/.dk.sci.pfu.edu.ru/home/t/b/tbkonovalova/laboratory/.git/logs/refs/remotes/origin/release
/afs/.dk.sci.pfu.edu.ru/home/t/b/tbkonovalova/laboratory/.git/logs/refs/heads
/afs/.dk.sci.pfu.edu.ru/home/t/b/tbkonovalova/laboratory/.git/logs/refs/heads/release
/afs/.dk.sci.pfu.edu.ru/home/t/b/tbkonovalova/laboratory/2020-2021
/afs/.dk.sci.pfu.edu.ru/home/t/b/tbkonovalova/laboratory/2020-2021/05
/afs/.dk.sci.pfu.edu.ru/home/t/b/tbkonovalova/laboratory/2020-2021/05/lab02
/afs/.dk.sci.pfu.edu.ru/home/t/b/tbkonovalova/.texlive2020
/afs/.dk.sci.pfu.edu.ru/home/t/b/tbkonovalova/.texlive2020/texmf-var
/afs/.dk.sci.pfu.edu.ru/home/t/b/tbkonovalova/.texlive2020/texmf-var/luatex-cache
/afs/.dk.sci.pfu.edu.ru/home/t/b/tbkonovalova/.texlive2020/texmf-var/luatex-cache/generic
/afs/.dk.sci.pfu.edu.ru/home/t/b/tbkonovalova/.texlive2020/texmf-var/luatex-cache/generic/names
/afs/.dk.sci.pfu.edu.ru/home/t/b/tbkonovalova/.texlive2020/texmf-var/luatex-cache/generic/fonts
/afs/.dk.sci.pfu.edu.ru/home/t/b/tbkonovalova/.texlive2020/texmf-var/luatex-cache/generic/fonts/otl
/afs/.dk.sci.pfu.edu.ru/home/t/b/tbkonovalova/.emacs.d
/afs/.dk.sci.pfu.edu.ru/home/t/b/tbkonovalova/tmp
/afs/.dk.sci.pfu.edu.ru/home/t/b/tbkonovalova/.pk1
/afs/.dk.sci.pfu.edu.ru/home/t/b/tbkonovalova/.pk1/nssdb
tbkonovalova@dk3n57 ~ $ rm logfile
rm: невозможно удалить 'logfile': Нет такого файла или каталога
tbkonovalova@dk3n57 ~ $ gedit &
[1] 24249
tbkonovalova@dk3n57 ~ $ ps | grep -i "gedit"
[1]+  Завершён      gedit
tbkonovalova@dk3n57 ~ $ man kill
tbkonovalova@dk3n57 ~ $ kill 4156
bash: kill: (4156) - Нет такого процесса
tbkonovalova@dk3n57 ~ $ man df
tbkonovalova@dk3n57 ~ $ man du
tbkonovalova@dk3n57 ~ $
```

Figure 3.17: команды df, du

```
~: man — Konsole
Файл Правка Вид Закладки Настройка Справка
DF(1) User Commands DF(1)
NAME
df - report file system disk space usage
SYNOPSIS
df [OPTION]... [FILE]...
DESCRIPTION
This manual page documents the GNU version of df. df displays the amount of disk 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. Disk 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 disk 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 version of df cannot show the space available on unmounted file systems, because on most kinds of systems doing so requires very nonportable intimate knowledge of file system structures.
OPTIONS
Show information about the file system on which each FILE resides, or all file systems by default.
Mandatory arguments to long options are mandatory for short options too.
-a, --all
include pseudo, duplicate, inaccessible file systems
-B, --block-size=SIZE
scale sizes by SIZE before printing them; e.g., '-BM' prints sizes in units of 1,048,576 bytes; see SIZE format below
-h, --human-readable
print sizes in powers of 1024 (e.g., 1023M)
-H, --si
print sizes in powers of 1000 (e.g., 1.1G)
Manual page df(1) line 1 (press h for help or q to quit)
```

Figure 3.18: информация команды df

```
~: man — Konsole
Файл Правка Вид Закладки Настройка Справка
DU(1) User Commands DU(1)
NAME
    du - estimate file space usage
SYNOPSIS
    du [OPTION]... [FILE]...
    du [OPTION]... --files0-from=F
DESCRIPTION
    Summarize disk 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
        print apparent sizes, rather than disk usage; although the apparent size is usually smaller, it
        may be larger due to holes in ('sparse') files, internal fragmentation, indirect blocks, and
        the like
    -B, --block-size=SIZE
        scale sizes by SIZE before printing them; e.g., '-BM' prints sizes in units of 1,048,576 bytes;
        see SIZE format below
    -b, --bytes
        equivalent to '--apparent-size --block-size=1'
    -c, --total
        produce a grand total
    -D, --dereference-args
Manual page du(1) line 1 (press h for help or q to quit)
```

Figure 3.19: информация команды du

```
tbkonovalova@dk3n57 ~ $ df
Файловая система      1K-блоков  Использовано  Доступно  Использовано%  Смонтировано в
none                  4000160      28424        3971736         1% /run
udev                 3924436         0        3924436         0% /dev
tmpfs                 4000160      170696       3829464         5% /dev/shm
/dev/sda8             491812356    59561872    407244740        13% /
tmpfs                  4096         0           4096         0% /sys/fs/cgroup
tmpfs                 4000164      174940       3825224         5% /tmp
/dev/sda6             91557952     811188       86072820         1% /var/cache/openafs
mark.sci.pfu.edu.ru:/usr/portage 8388608    4353792     1804032        71% /usr/portage
mark.sci.pfu.edu.ru:/usr/local/share/portage 8388608    4353792     1804032        71% /usr/local/share/p
ortage
mark.sci.pfu.edu.ru:/com/lib/portage 733747200  115401984   618345216        16% /com/lib/portage
AFS                  2147483647         0    2147483647         0% /afs
tmpfs                 800032         320        799712         1% /run/user/3713
tbkonovalova@dk3n57 ~ $
```

Figure 3.20: команда df в консоли





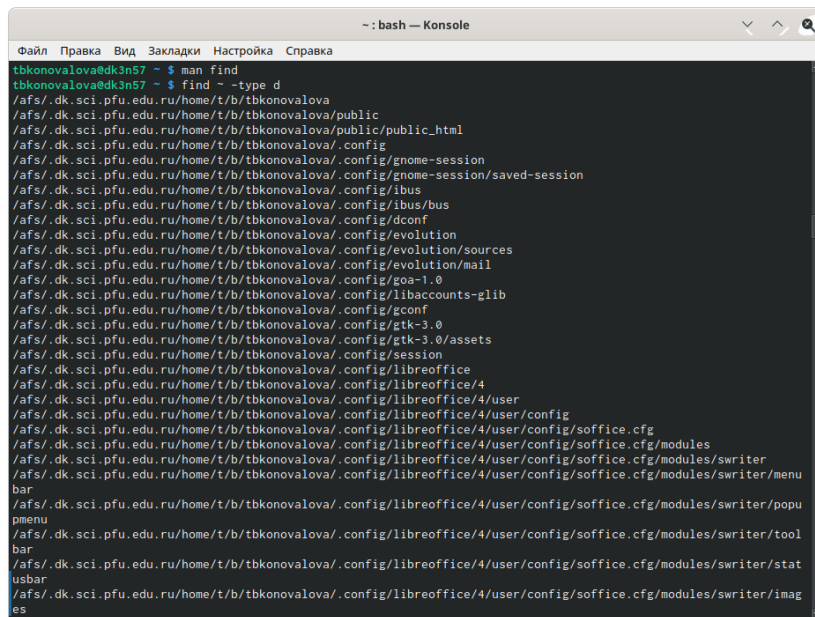


Figure 3.23: имена всех директорий

Ответы на контрольные вопросы:

1). 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.

2). ‘>’ Перенаправление вывода в файл

‘>>’ Перенаправление вывода в файл и открытие файла в режиме добавления (данные добавляются в конец файла)/

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

Синтаксис следующий:

команда1|команда2 (это означает, что вывод команды 1 передается на ввод команде 2)

4). Процесс рассматривается операционной системой как заявка на потребление всех видов ресурсов, кроме одного – процессорного времени. Этот последний важнейший ресурс распределяется операционной системой между другими единицами работы – потоками, которые и получили свое название благодаря тому, что они представляют собой последовательности (потоки выполнения) команд.

Процесс – это выполнение программы. Он считается активной сущностью и реализует действия, указанные в программе.

Программа представляет собой статический набор команд, а процесс это набор ресурсов и данных, использующихся при выполнении программы.

5). pid: идентификатор процесса (PID) процесса (processID), к которому вызывают метод

gid: идентификатор группы UNIX, в котором работает программа.

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

Запущенные фоном программы называются задачами (jobs). Ими можно управлять с помощью команды jobs, которая выводит список запущенных в данный момент задач.

7). top – это консольная программа, которая показывает список работающих процессов в системе. Программа в реальном времени отсортирует запущенные процессы по их нагрузке на процессор.

htop – это продвинутый консольный мониторинг процессов. Утилита выводит постоянно меняющийся список системных процессов, который сортируется в зависимости от нагрузки на ЦПУ. Если делать сравнение с top, то htop показывает абсолютно все процессы в системе, время их непрерывного использования, загрузку процессоров и расход оперативной памяти.

8). `find` – 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.

Команда `find` имеет такой синтаксис:

`find[папка][параметры] критерий шаблон [действие]`

Папка – каталог в котором будем искать

Параметры – дополнительные параметры, например, глубина поиска, и т.д.

Критерий – по какому критерию будем искать: имя, дата создания, права, владелец и т.д.

Шаблон – непосредственно значение по которому будем отбирать файлы.

Основные параметры:

- P никогда не открывать символические ссылки
- L - получает информацию о файлах по символическим ссылкам. Важно для дальнейшей обработки, чтобы обрабатывалась не ссылка, а сам файл.

- maxdepth - максимальная глубина поиска по подкаталогам, для поиска только в текущем каталоге установите 1.

- depth - искать сначала в текущем каталоге, а потом в подкаталогах

- mount искать файлы только в этой файловой системе.

- version - показать версию утилиты `find`

- print - выводить полные имена файлов

- typef - искать только файлы

- typed - поиск папки в Linux

Основные критерии:

- name - поиск файлов по имени

- perm - поиск файлов в Linux по режиму доступа

- user - поиск файлов по владельцу

- group - поиск по группе

- mtime - поиск по времени модификации файла

- atime - поиск файлов по дате последнего чтения
- nogroup - поиск файлов, не принадлежащих ни одной группе
- nouser - поиск файлов без владельцев
- newer - найти файлы новее чем указанный
- size - поиск файлов в Linux по их размеру

Примеры:

find~ -type d поиск директорий в домашнем каталоге

find~ -type f -name ".\*" поиск скрытых файлов в домашнем каталоге

9). Файл по его содержимому можно найти с помощью команды grep: «grep -r» слово/выражение, которое нужно найти»».

10). Утилита df, позволяет проанализировать свободное пространство на всех подключенных к системе разделах.

11). При выполнении команды du (без указания папки и опции) можно получить все файлы и папки текущей директории с их размерами. Для домашнего каталога: du ~/

12). Основные сигналы (каждый сигнал имеет свой номер), которые используются для завершения процесса:

- SIGINT—самый безобидный сигнал завершения, означает Interrupt. Он отправляется процессу, запущенному из терминала с помощью сочетания клавиш Ctrl+C. Процесс правильно завершает все свои действия и возвращает управление;
- SIGQUIT—это еще один сигнал, который отправляется с помощью сочетания клавиш, программе, запущенной в терминале. Он сообщает ей что нужно завершиться и программа может выполнить корректное завершение или проигнорировать сигнал. В отличие от предыдущего, она генерирует дампы памяти. Сочетание клавиш Ctrl+Q;
- SIGHUP—сообщает процессу, что соединение с управляющим терминалом разорвано, отправляется, в основном, системой при разрыве соединения с

интернетом;

- **SIGTERM**—немедленно завершает процесс, но обрабатывается программой, поэтому позволяет ей завершить дочерние процессы и освободить все ресурсы;
- **SIGKILL**—тоже немедленно завершает процесс, но, в отличие от предыдущего варианта, он не передается самому процессу, а обрабатывается ядром. Поэтому ресурсы и дочерние процессы остаются запущенными.

Также для передачи сигналов процессам в Linux используется утилита `kill`, её синтаксис: `kill [-сигнал] [pid_процесса]` (PID – уникальный идентификатор процесса). Сигнал представляет собой один из выше перечисленных сигналов для завершения процесса.

Перед тем, как выполнить остановку процесса, нужно определить его PID. Для этого используют команды `ps` и `grep`. Команда `ps` предназначена для вывода списка активных процессов в системе и информации о них. Команда `grep` запускается одновременно с `ps` (в канале) и будет выполнять поиск по результатам команды `ps`.

Утилита `pkill` – это оболочка для `kill`, она ведет себя точно так же, и имеет тот же синтаксис, только в качестве идентификатора процесса ей нужно передать его имя.

`killall` работает аналогично двум предыдущим утилитам. Она тоже принимает имя процесса в качестве параметра и ищет его PID в директории `/proc`. Но эта утилита обнаружит все процессы с таким именем и завершит их.

## 4 Выводы

В ходе выполнения данной лабораторной работы я изучила инструменты поиска файлови фильтрации текстовых данных, а также приобрела практические навыки: по управлению процессами (и заданиями), по проверке использования диска и обслуживанию файловых систем.