

# **Отчёт по лабораторной работе №7**

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

Калистратова Ксения Евгеньевна

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>4</b>
<b>2</b>	<b>Задачи лабораторной работы</b>	<b>5</b>
<b>3</b>	<b>Выполнение лабораторной работы</b>	<b>6</b>
<b>4</b>	<b>Выводы</b>	<b>16</b>
<b>5</b>	<b>Ответы на контрольные вопросы</b>	<b>17</b>

# List of Figures

3.1	Использую команды <code>grep -e '.conf\$' file.txt &gt; conf.txt, cat conf.txt . .</code>	8
3.2	Использую команды <code>find ~ -maxdepth 1 -name "c" -print, ls ~/c, ls -a ~   grep c*</code> . . . . .	9
3.3	Вывод (постранично) имён файлов из каталога <code>/etc</code> , начинающиеся с символа <code>h</code> . . . . .	9
3.4	Использую команду <code>find/ -name "log*" &gt; logfile &amp;</code> . . . . .	10
3.5	Использую команду <code>cat logfile</code> . . . . .	10
3.6	Использую команду <code>rm logfile</code> . . . . .	10
3.7	Использую команду <code>gedit &amp;</code> . . . . .	10
3.8	Использую команду <code>man kill</code> и открывается памятка о данной команде	11
3.9	Использую команду <code>kill 6386</code> для завершения процесса <code>gedit</code> . . .	12
3.10	Использую команды <code>man df</code> и <code>man du</code> . . . . .	12
3.11	Памятка о команде <code>df</code> . . . . .	13
3.12	Памятка о команде <code>du</code> . . . . .	13
3.13	Использую команды <code>df</code> и <code>du</code> . . . . .	14
3.14	Использую команду <code>man find</code> и открывается памятка о данной команде . . . . .	15
3.15	Использую команду <code>find ~ -type d</code> . . . . .	15

# 1 Цель работы

Целью данной работы является 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.

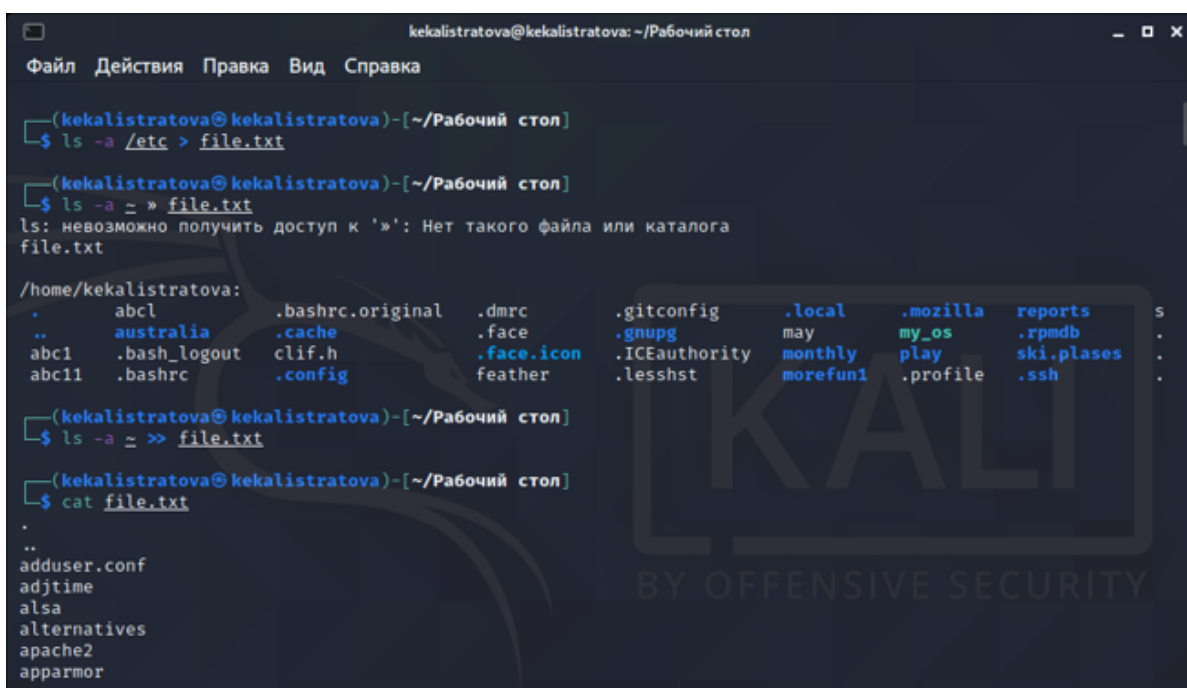
## 2 Задачи лабораторной работы

Задачи:

1. Изучить потоки ввода и вывода.
2. Изучить конвейер.
3. Изучить команду поиска файлов.
4. Изучить команду, позволяющую найти указанную строку символов.
5. Изучить команды по проверке использования диска.
6. В ходе работы использовать эти команды и интерпретировать их вывод.

### 3 Выполнение лабораторной работы

1. Входим в систему, используя свой логин и пароль.
2. Для того, чтобы записать в файл file.txt названия файлов, содержащихся в каталоге /etc, использую команду «ls -a /etc > file.txt». Далее с помощью команды «ls -a ~ > file.txt» дописываю в этот же файл названия файлов, содержащихся в моем домашнем каталоге. Командой «cat file.txt» просматриваю файл, чтобы убедиться в правильности действий (рис. ??).



```
kekalistratova@kekalistratova: ~/Рабочий стол
Файл Действия Правка Вид Справка

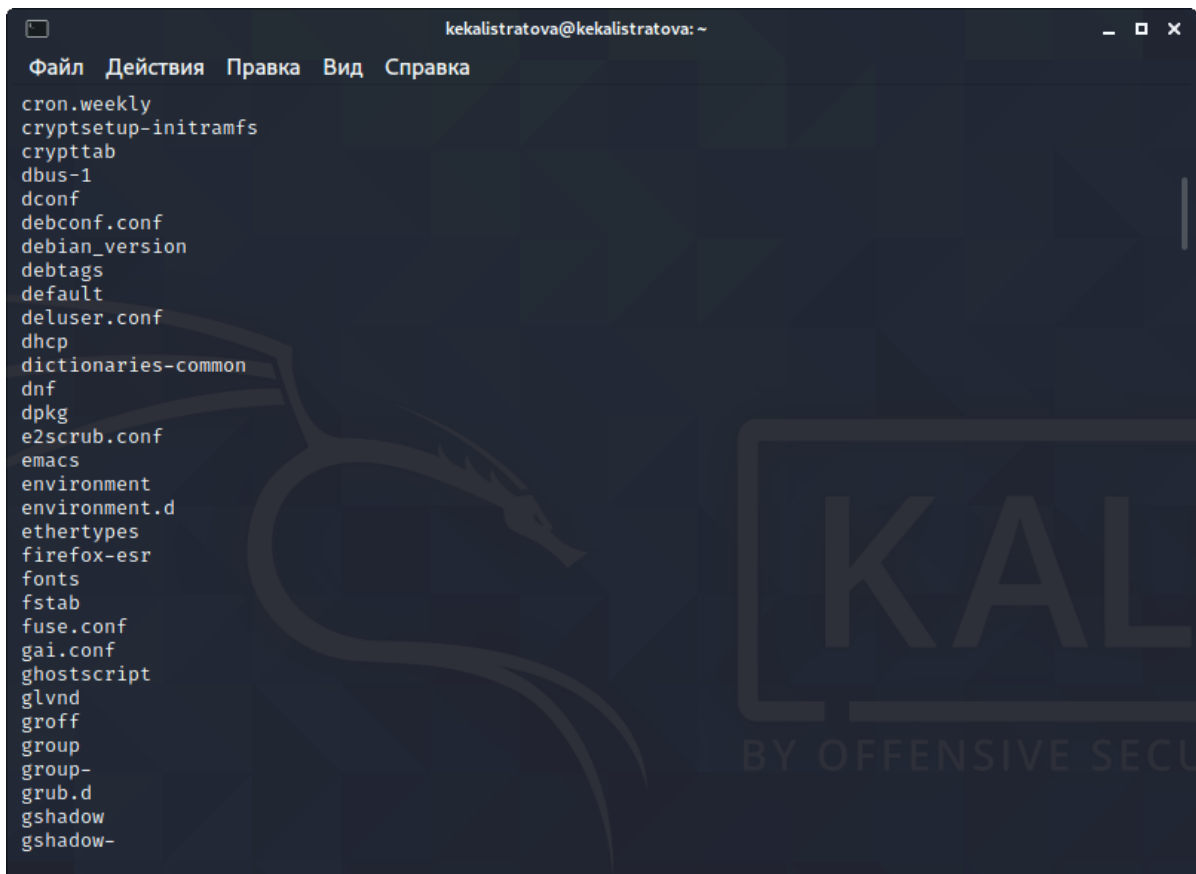
(kekalistratova@kekalistratova)-[~/Рабочий стол]
$ ls -a /etc > file.txt

(kekalistratova@kekalistratova)-[~/Рабочий стол]
$ ls -a ~ > file.txt
ls: невозможно получить доступ к '~': Нет такого файла или каталога
file.txt

/home/kekalistratova:
.      abcl      .bashrc.original  .dmrc      .gitconfig    .local      .mozilla  reports  s
..     australia .cache            .face      .gnupg        may         my_os     .rpmdb   .
abc1   .bash_logout  clif.h          .face.icon .ICEauthority monthly    play      ski.plases .
abc11  .bashrc       .config         feather    .lessht      morefun1   .profile  .ssh     .

(kekalistratova@kekalistratova)-[~/Рабочий стол]
$ ls -a ~ >> file.txt

(kekalistratova@kekalistratova)-[~/Рабочий стол]
$ cat file.txt
.
..
adduser.conf
adjtime
alsa
alternatives
apache2
apparmor
```



3. Вывожу имена всех файлов из file.txt, имеющих расширение .conf и записываю их в новый текстовый файл conf.txt с помощью команды «grep -e '.conf\$' file.txt > conf.txt». Командой «cat conf.txt» проверяю правильность выполненных действий (рис. ??).

```
kekalistratova@kekalistratova: ~  
Файл Действия Правка Вид Справка  
(kekalistratova@kekalistratova)~[~/Рабочий стол]  
$ grep -e '\.conf$' file.txt > conf.txt  
  
(kekalistratova@kekalistratova)~[~/Рабочий стол]  
$ cat conf.txt  
adduser.conf  
ca-certificates.conf  
debconf.conf  
deluser.conf  
e2scrub.conf  
fuse.conf  
gai.conf  
host.conf  
ipsec.conf  
kernel-img.conf  
ld.so.conf  
libaudit.conf  
logrotate.conf  
mke2fs.conf  
nftables.conf  
nsswitch.conf  
pam.conf  
resolv.conf  
rsyslog.conf  
sensors3.conf  
strongswan.conf  
sudo.conf  
sudo_logsrvd.conf  
sysctl.conf  
ucf.conf  
updatedb.conf
```

Figure 3.1: Используя команды `grep -e '.conf$' file.txt > conf.txt`, `cat conf.txt`

4. Определить, какие файлы в моем домашнем каталоге имеют имена, начинающиеся с символа `c`, можно несколькими командами: `find ~ -maxdepth 1 -name "c" -print` (опция `maxdepth 1` необходима для того, чтобы файлы находились только в домашнем каталоге (не в его подкаталогах)), `ls ~/c` и `ls -a ~ | grep c*` (рис. ??).

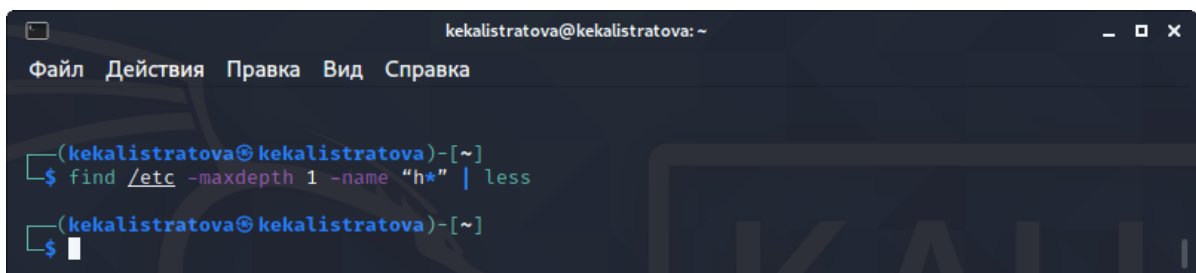


A terminal window titled 'kekalistratova@kekalistratova: ~' with a menu bar (Файл, Действия, Правка, Вид, Справка). The user enters the command 'find ~ -name "c\*.txt"'. The output shows the file path '/home/kekalistratova/Рабочий стол/conf.txt'. Then, the user enters 'ls -a ~ | grep c\*'. The output lists files and directories in the home directory, with those containing 'c' highlighted in red: '.', '..', 'abc1', 'abc11', 'abc1', 'australia', '.bash\_logout', '.bashrc', '.bashrc.original', '.cache', '.config', '.dmrc', '.face', '.face.icon', 'feather', '.gitconfig', '.gnupg', '.ICEauthority', '.lessshst', '.local', 'may', and 'monthly'.

```
kekalistratova@kekalistratova: ~  
Файл Действия Правка Вид Справка  
(kekalistratova@kekalistratova)~  
$ find ~ -name "c*.txt"  
/home/kekalistratova/Рабочий стол/conf.txt  
(kekalistratova@kekalistratova)~  
$ ls -a ~ | grep c*  
.  
..  
abc1  
abc11  
abc1  
australia  
.bash_logout  
.bashrc  
.bashrc.original  
.cache  
.config  
.dmrc  
.face  
.face.icon  
feather  
.gitconfig  
.gnupg  
.ICEauthority  
.lessshst  
.local  
may  
monthly
```

Figure 3.2: Используя команды `find ~ -maxdepth 1 -name "c" -print`, `ls ~/c`, `ls -a ~ | grep c*`

- Чтобы вывести на экран (постранично) имена файлов из каталога `/etc`, начинающиеся с символа `h`, воспользуемся командой «`find /etc -maxdepth 1 -name "h*" | less`» (рис. ??, ??).

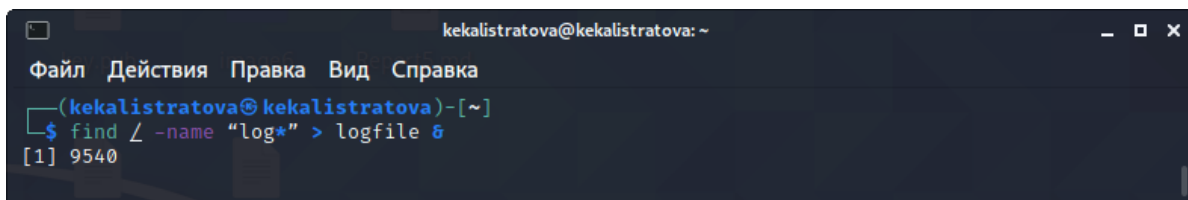
A terminal window titled 'kekalistratova@kekalistratova: ~' with a menu bar (Файл, Действия, Правка, Вид, Справка). The user enters the command 'find /etc -maxdepth 1 -name "h\*" | less'. The prompt returns to the user, and the cursor is on a new line.

```
kekalistratova@kekalistratova: ~  
Файл Действия Правка Вид Справка  
(kekalistratova@kekalistratova)~  
$ find /etc -maxdepth 1 -name "h*" | less  
(kekalistratova@kekalistratova)~  
$
```

Figure 3.3: Вывод (постранично) имён файлов из каталога `/etc`, начинающиеся с символа `h`

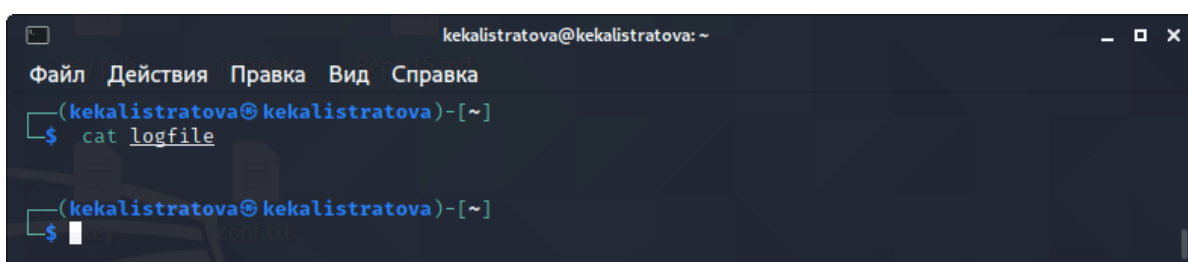
- Запускаю в фоновом режиме процесс, который будет записывать в файл `~/logfile` файлы, имена которых начинаются с `log`, используя команду «`find/`

-name "log\*" > logfile &»(рис. ??). Командой «cat logfile» проверяю выполненные действия (рис. ??, ??). Далее удаляю файл ~/logfile командой «rm logfile» (рис. ??).



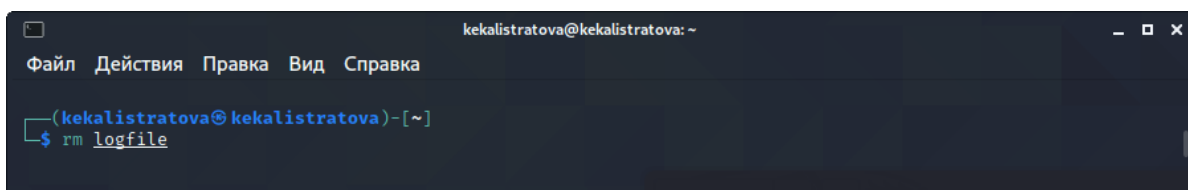
```
kekalistratova@kekalistratova: ~  
Файл Действия Правка Вид Справка  
(kekalistratova@kekalistratova)-[~]  
$ find / -name "log*" > logfile &  
[1] 9540
```

Figure 3.4: Использую команду find/ -name "log\*" > logfile &



```
kekalistratova@kekalistratova: ~  
Файл Действия Правка Вид Справка  
(kekalistratova@kekalistratova)-[~]  
$ cat logfile  
  
(kekalistratova@kekalistratova)-[~]  
$
```

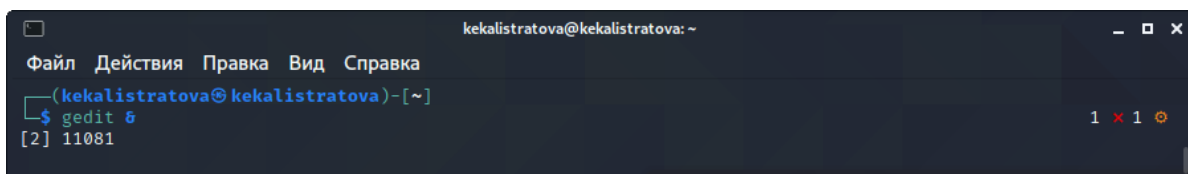
Figure 3.5: Использую команду cat logfile



```
kekalistratova@kekalistratova: ~  
Файл Действия Правка Вид Справка  
(kekalistratova@kekalistratova)-[~]  
$ rm logfile
```

Figure 3.6: Использую команду rm logfile

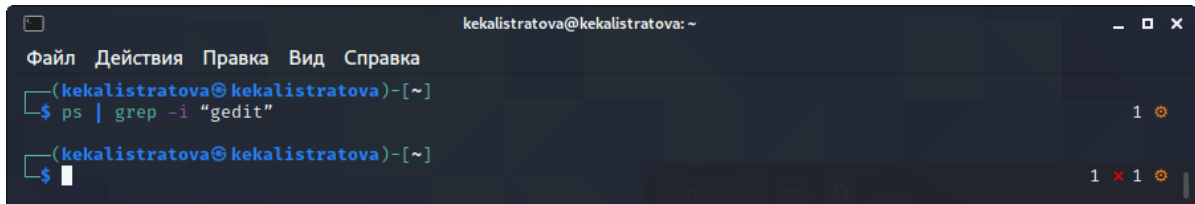
7. Запускаю редактор gedit в фоновом режиме командой «gedit &» (рис. ??). После этого на экране появляется окно редактора.



```
kekalistratova@kekalistratova: ~  
Файл Действия Правка Вид Справка  
(kekalistratova@kekalistratova)-[~]  
$ gedit &  
[2] 11081
```

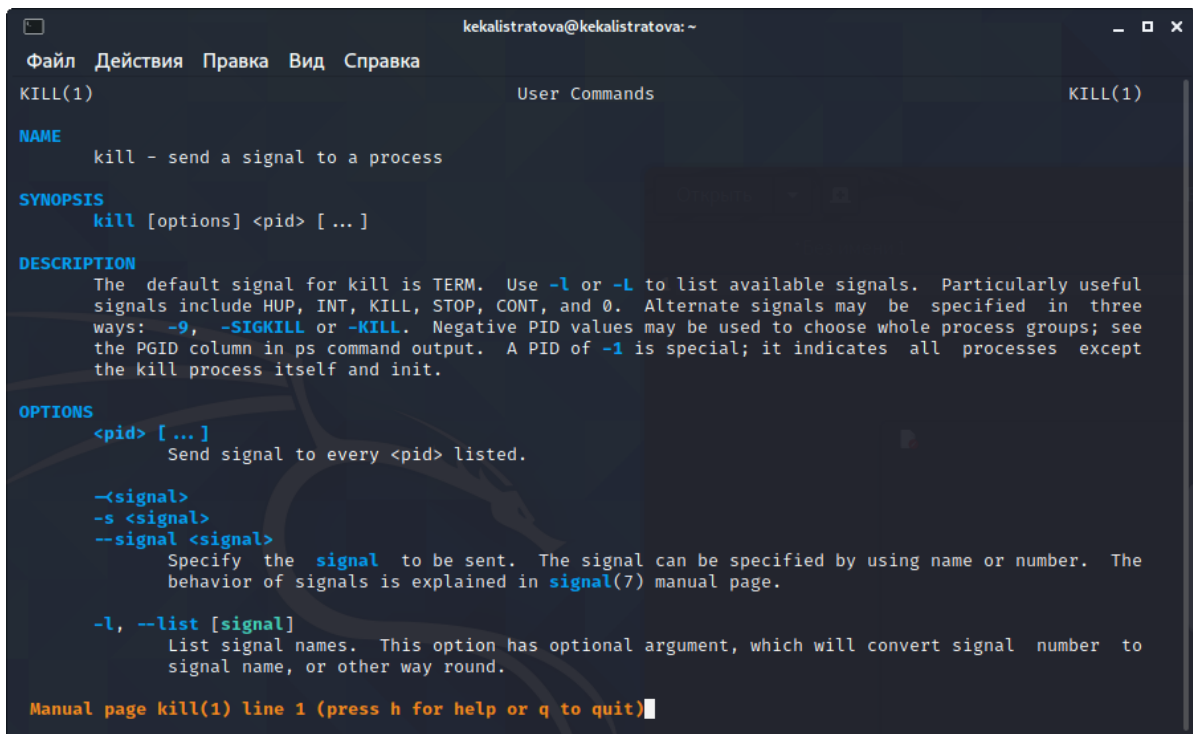
Figure 3.7: Использую команду gedit &

8. Чтобы определить идентификатор процесса `gedit`, использую команду «`ps | grep -i "gedit"`» (рис. ??). Из рисунка видно, что наш процесс имеет PID6028. Узнать идентификатор процесса можно также, используя команду «`pidof gedit`» или «`pgrep gedit`»



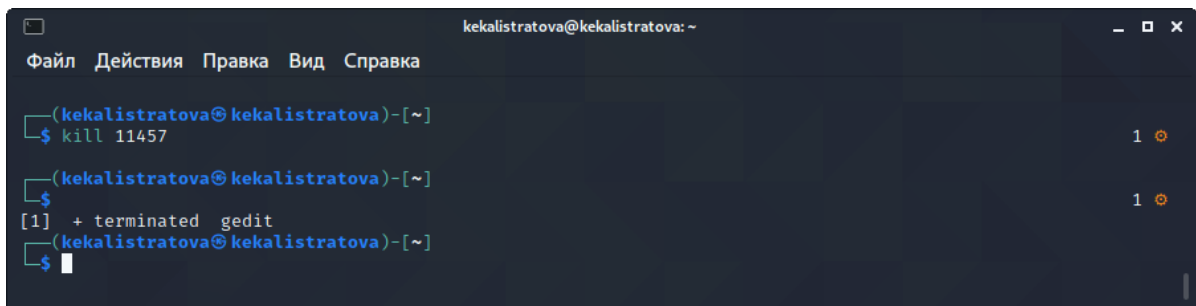
```
kekalistratova@kekalistratova: ~  
Файл Действия Правка Вид Справка  
(kekalistratova@kekalistratova)~  
$ ps | grep -i "gedit"  
(kekalistratova@kekalistratova)~  
$
```

9. Прочитав информацию о команде `kill` с помощью команды «`man kill`», используя её для завершения процесса `gedit` (команда «`kill6386`») (рис. ??, ??)



```
kekalistratova@kekalistratova: ~  
Файл Действия Правка Вид 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 behavior 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 signal name, or other way round.  
Manual page kill(1) line 1 (press h for help or q to quit)
```

Figure 3.8: Использу команду `man kill` и открывается памятка о данной команде



```
kekalistratova@kekalistratova: ~  
Файл Действия Правка Вид Справка  
(kekalistratova@kekalistratova)~  
$ kill 11457  
(kekalistratova@kekalistratova)~  
$  
[1] + terminated gedit  
(kekalistratova@kekalistratova)~  
$
```

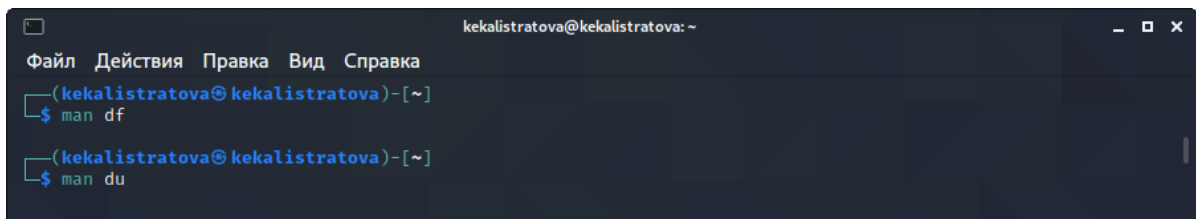
Figure 3.9: Использу команду kill6386 для завершения процесса gedit

10. С помощью команд «man df»и «man du» узнаю информацию по необходи-  
мым командам и далее использую их (рис. ??, ??, ??, ??).

df – утилита,показывающаясписок всех файловых систем по именам устройств,  
сообщает их размер, занятое и свободное пространство и точки монтирования.

Синтаксис: df [опции]у устройство

du – утилита, предназначенная для вывода информации об объеме дискового  
пространства, занятого файлами и директориями. Она принимает путь к эле-  
менту файловой системы и выводит информацию о количестве байт дискового  
пространства или блоков диска, задействованных для его хранения.Синтаксис:  
du [опции] каталог\_или\_файл



```
kekalistratova@kekalistratova: ~  
Файл Действия Правка Вид Справка  
(kekalistratova@kekalistratova)~  
$ man df  
(kekalistratova@kekalistratova)~  
$ man du
```

Figure 3.10: Использу команды man df и man du

```
kekalistratova@kekalistratova: ~  
Файл Действия Правка Вид 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.  
  
Manual page df(1) line 1 (press h for help or q to quit)
```

Figure 3.11: Памятка о команде df

```
kekalistratova@kekalistratova: ~  
Файл Действия Правка Вид 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  
  
Manual page du(1) line 1 (press h for help or q to quit)
```

Figure 3.12: Памятка о команде du

```
kekalistratova@kekalistratova: ~  
Файл Действия Правка Вид Справка  
~(kekalistratova@ kekalistratova)-[~]  
$ df  
Файловая система      1К-блоков  Использовано  Доступно  Использовано%  Смонтировано в  
udev                  2502196      0      2502196      0% /dev  
tmpfs                  505384      944      504440      1% /run  
/dev/mapper/kekalistratova--vg-root 6661460 3561904 2739928 57% /  
tmpfs                  2526916    24904    2502012      1% /dev/shm  
tmpfs                   5120         0       5120      0% /run/lock  
/dev/sda1              480618      61316    394368     14% /boot  
kali_Linux_            249032700 218587892 30444808    88% /media/sf_kali_Linux_  
tmpfs                   505380         68     505312      1% /run/user/1000  
  
~(kekalistratova@ kekalistratova)-[~]  
$ du  
4      ./Документы  
4      ./monthly  
4      ./play/games/fun  
8      ./play/games  
12     ./play  
500    ./rpmdb  
4      ./gnupg/private-keys-v1.d  
12     ./gnupg  
4      ./australia  
4      ./Видео  
4      ./reports/monthly/monthly  
8      ./reports/monthly  
12     ./reports  
16     ./ssh  
4      ./Шаблоны  
4      ./ski.plases/equipment  
4      ./ski.plases/plans  
12     ./ski.plases  
60     ./cache/fontconfig  
4      ./cache/sessions  
796    ./cache/gstreamer-1.0
```

Figure 3.13: Используя команды df и du

11. Вывожу имена всех директорий, имеющих в моем домашнем каталоге с помощью команды «find ~ -type d», предварительно получив информацию с помощью команды «man find» (рис. ??, ??)



```
kekalistratova@kekalistratova: ~
Файл Действия Правка Вид Справка
FIND(1) General Commands Manual FIND(1)

NAME
  find - search for files in a directory hierarchy

SYNOPSIS
  find [-H] [-L] [-P] [-D debugopts] [-Olevel] [starting-point ...] [expression]

DESCRIPTION
  This manual page documents the GNU version of find. GNU find searches the directory tree rooted at each given starting-point by evaluating the given expression from left to right, according to the rules of precedence (see section OPERATORS), until the outcome is known (the left hand side is false for and operations, true for or), at which point find moves on to the next file name. If no starting-point is specified, find assumes .

  If you are using find in an environment where security is important (for example if you are using it to search directories that are writable by other users), you should read the 'Security Considerations' chapter of the findutils documentation, which is called Finding Files and comes with findutils. That document also includes a lot more detail and discussion than this manual page, so you may find it a more useful source of information.

Manual page find(1) line 1 (press h for help or q to quit)
```

Figure 3.14: Использу команду `man find` и открывается памятка о данной команде



```
(kekalistratova@kekalistratova)-[~]
$ find ~ -type d
/home/kekalistratova
/home/kekalistratova/Документы
/home/kekalistratova/monthly
/home/kekalistratova/play
/home/kekalistratova/play/games
/home/kekalistratova/play/games/fun
/home/kekalistratova/.rpmdb
/home/kekalistratova/.gnupg
/home/kekalistratova/.gnupg/private-keys-v1.d
/home/kekalistratova/australia
/home/kekalistratova/Видео
/home/kekalistratova/reports
/home/kekalistratova/reports/monthly
/home/kekalistratova/reports/monthly/monthly
/home/kekalistratova/.ssh
/home/kekalistratova/Шаблоны
```

Figure 3.15: Использу команду `find ~ -type d`

## 4 Выводы

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



## 5 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 передастся 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 `stop`, 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+;`;

`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 дочерни епроцессы 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 вышеперечисленных 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. Утилитар kill–это оболочка для kill, она ведет себя точно также, и имеет тот же синтаксис, только в качестве идентификатора процесса ей нужно передать его имя. kill all 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.