

РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и естественных наук

Кафедра прикладной информатики и теории вероятностей

ОТЧЕТ

ПО ЛАБОРАТОРНОЙ РАБОТЕ № 4

дисциплина: Операционные системы

Студент:

Леонтьева Ксения Андреевна

Группа:

НПМбд-02-19

МОСКВА

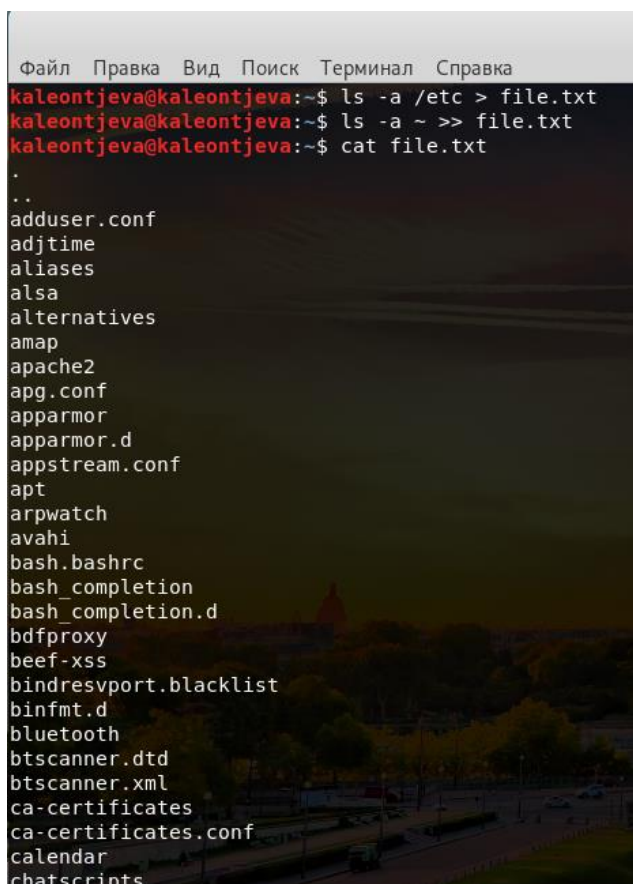
2020 г.

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. Ход работы:

- 1) Осуществляю вход в систему, используя свои 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 (Рисунки 1, 2).



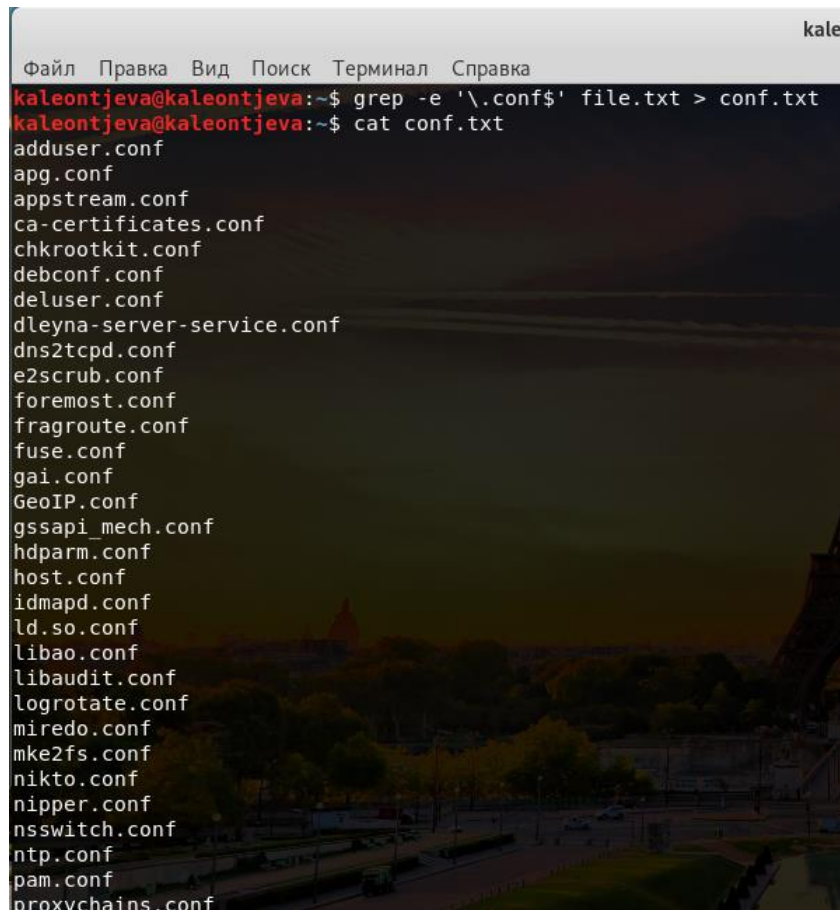
(Рисунок 1)



(Рисунок 2)

- 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

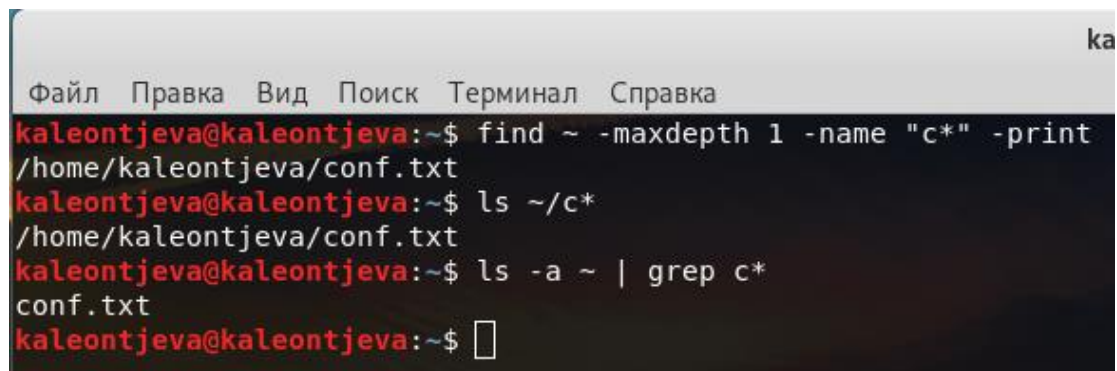
правильность выполненных действий (Рисунок 3).



```
Файл  Правка  Вид  Поиск  Терминал  Справка
kaleontjeva@kaleontjeva:~$ grep -e '\.conf$' file.txt > conf.txt
kaleontjeva@kaleontjeva:~$ cat conf.txt
adduser.conf
apg.conf
appstream.conf
ca-certificates.conf
chkrootkit.conf
debconf.conf
deluser.conf
dleyna-server-service.conf
dns2tcpd.conf
e2scrub.conf
foremost.conf
fragroute.conf
fuse.conf
gai.conf
GeoIP.conf
gssapi_mech.conf
hdparm.conf
host.conf
idmapd.conf
ld.so.conf
libao.conf
libaudit.conf
logrotate.conf
miredo.conf
mke2fs.conf
nikto.conf
nipper.conf
nsswitch.conf
ntp.conf
pam.conf
proxvchains.conf
```

(Рисунок 3)

- 4) Определить, какие файлы в моем 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 только в домашнем каталоге (не в его подкаталогах)), «ls ~/с*» и «ls -a ~ | grep с*» (Рисунок 4).

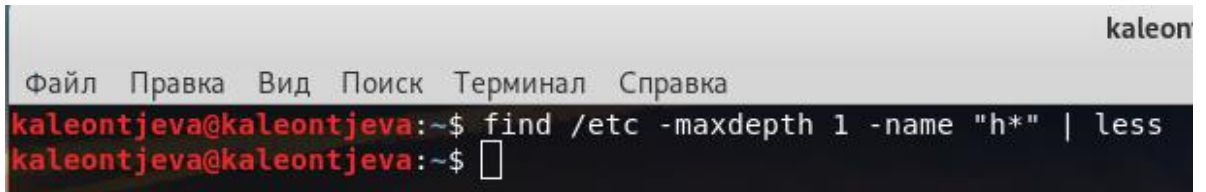


```
Файл  Правка  Вид  Поиск  Терминал  Справка
kaleontjeva@kaleontjeva:~$ find ~ -maxdepth 1 -name "с*" -print
/home/kaleontjeva/conf.txt
kaleontjeva@kaleontjeva:~$ ls ~/с*
/home/kaleontjeva/conf.txt
kaleontjeva@kaleontjeva:~$ ls -a ~ | grep с*
conf.txt
kaleontjeva@kaleontjeva:~$
```

(Рисунок 4)

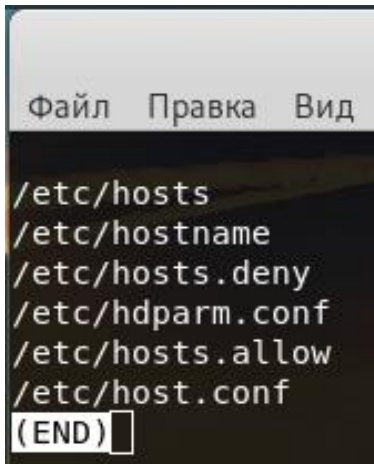
- 5) Чтобы вывести на экран (по странично) имена файлов из каталога /etc, начинающиеся с символа h, воспользуемся командой «find /etc –

maxdepth 1 -name "h*" | less» (Рисунки 5, 6).



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

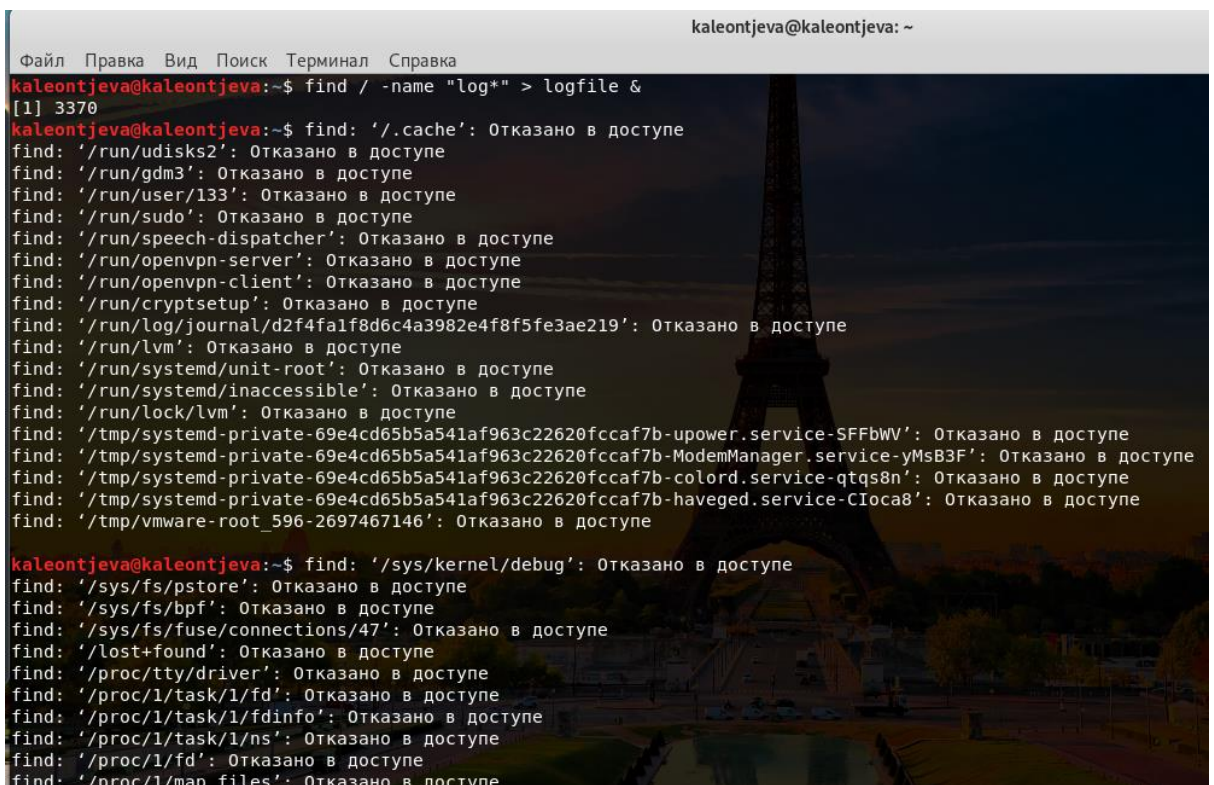
(Рисунок 5)



```
Файл Правка Вид
/etc/hosts
/etc/hostname
/etc/hosts.deny
/etc/hdparm.conf
/etc/hosts.allow
/etc/host.conf
(END)
```

(Рисунок 6)

- б) Запускаю в фоновом режиме процесс, который будет записывать в файл ~/logfile файлы, имена которых начинаются с log, используя команду «find / -name "log*" > logfile &» (Рисунки 7, 8). Командой «cat logfile» проверяю выполненные действия (Рисунок 9). Далее удаляю файл ~/logfile командой «rm logfile» (Рисунки 10).



```
kaleontjeva@kaleontjeva:~$ find / -name "log*" > logfile &
[1] 3370
kaleontjeva@kaleontjeva:~$ find: '/.cache': Отказано в доступе
find: '/run/udisks2': Отказано в доступе
find: '/run/gdm3': Отказано в доступе
find: '/run/user/133': Отказано в доступе
find: '/run/sudo': Отказано в доступе
find: '/run/speech-dispatcher': Отказано в доступе
find: '/run/openvpn-server': Отказано в доступе
find: '/run/openvpn-client': Отказано в доступе
find: '/run/cryptsetup': Отказано в доступе
find: '/run/log/journal/d2f4falf8d6c4a3982e4f8f5fe3ae219': Отказано в доступе
find: '/run/lvm': Отказано в доступе
find: '/run/systemd/unit-root': Отказано в доступе
find: '/run/systemd/inaccessible': Отказано в доступе
find: '/run/lock/lvm': Отказано в доступе
find: '/tmp/systemd-private-69e4cd65b5a541af963c22620fccaf7b-upower.service-SFFbWV': Отказано в доступе
find: '/tmp/systemd-private-69e4cd65b5a541af963c22620fccaf7b-ModemManager.service-yMsB3F': Отказано в доступе
find: '/tmp/systemd-private-69e4cd65b5a541af963c22620fccaf7b-color.service-qtqs8n': Отказано в доступе
find: '/tmp/systemd-private-69e4cd65b5a541af963c22620fccaf7b-haveged.service-CIoca8': Отказано в доступе
find: '/tmp/vmware-root_596-2697467146': Отказано в доступе

kaleontjeva@kaleontjeva:~$ find: '/sys/kernel/debug': Отказано в доступе
find: '/sys/fs/pstore': Отказано в доступе
find: '/sys/fs/bpf': Отказано в доступе
find: '/sys/fs/fuse/connections/47': Отказано в доступе
find: '/lost+found': Отказано в доступе
find: '/proc/tty/driver': Отказано в доступе
find: '/proc/1/task/1/fd': Отказано в доступе
find: '/proc/1/task/1/fdinfo': Отказано в доступе
find: '/proc/1/task/1/ns': Отказано в доступе
find: '/proc/1/fd': Отказано в доступе
find: '/proc/1/map_files': Отказано в доступе
```


(Рисунок 7)

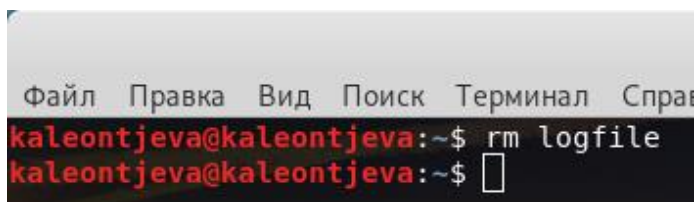
```
kaleontje
Файл Правка Вид Поиск Терминал Справка
find: '/var/lib/gdm3/.cache/libgweather': Отказано в доступе
find: '/var/lib/gdm3/.gnupg': Отказано в доступе
find: '/var/lib/gdm3/.local/share/gnome-shell': Отказано в доступе
find: '/var/lib/gdm3/.local/share/sounds': Отказано в доступе
find: '/var/lib/gdm3/.config/ibus': Отказано в доступе
find: '/var/lib/gdm3/.config/pulse': Отказано в доступе
find: '/var/lib/gdm3/.config/gnome-session': Отказано в доступе
find: '/var/lib/samba/usershares': Отказано в доступе
find: '/var/lib/samba/private/msg.sock': Отказано в доступе
find: '/var/lib/udisks2': Отказано в доступе
find: '/var/lib/private': Отказано в доступе
find: '/var/lib/mysql/performance_schema': Отказано в доступе
find: '/var/lib/mysql/mysql': Отказано в доступе
find: '/var/lib/color/.cache': Отказано в доступе
find: '/var/lib/apt/lists/partial': Отказано в доступе
find: '/var/lib/php/sessions': Отказано в доступе
find: '/var/log/apache2': Отказано в доступе
find: '/var/log/exim4': Отказано в доступе
find: '/var/log/gdm3': Отказано в доступе
find: '/var/log/samba': Отказано в доступе
find: '/var/log/unattended-upgrades': Отказано в доступе
find: '/var/log/inetd': Отказано в доступе
find: '/var/log/private': Отказано в доступе
find: '/var/log/mysql': Отказано в доступе
find: '/var/log/speech-dispatcher': Отказано в доступе
find: '/var/cache/ldconfig': Отказано в доступе
find: '/var/cache/private': Отказано в доступе
find: '/var/cache/apt/archives/partial': Отказано в доступе
find: '/media/root': Отказано в доступе

[1]+ Выход 1          find / -name "log*" > logfile
kaleontjeva@kaleontjeva:~$
```

(Рисунок 8)

```
kaleontjeva@kaleontjeva: ~
Файл Правка Вид Поиск Терминал Справка
kaleontjeva@kaleontjeva:~$ cat logfile
/run/log
/usr/src/linux-headers-5.2.0-kali2-common/include/linux/logic_pio.h
/usr/src/linux-headers-5.2.0-kali2-common/include/linux/log2.h
/usr/src/linux-headers-5.2.0-kali2-amd64/include/config/printk/safe/log
/usr/src/linux-headers-5.2.0-kali2-amd64/include/config/logirumblepad2
/usr/src/linux-headers-5.2.0-kali2-amd64/include/config/logig940
/usr/src/linux-headers-5.2.0-kali2-amd64/include/config/log
/usr/src/linux-headers-5.2.0-kali2-amd64/include/config/nf/log
/usr/src/linux-headers-5.2.0-kali2-amd64/include/config/logitech
/usr/src/linux-headers-5.2.0-kali2-amd64/include/config/security/tomoyo/max/audit/log.h
/usr/src/linux-headers-5.2.0-kali2-amd64/include/config/nft/log.h
/usr/src/linux-headers-5.2.0-kali2-amd64/include/config/scsi/logging.h
/usr/src/linux-headers-5.2.0-kali2-amd64/include/config/logiwheels
/usr/src/linux-headers-5.2.0-kali2-amd64/include/config/dm/log
/usr/src/linux-headers-5.2.0-kali2-amd64/include/config/hid/logitech.h
/usr/src/linux-headers-5.2.0-kali2-amd64/include/config/hid/logitech
/usr/src/linux-headers-5.2.0-kali2-amd64/include/config/sfc/mcdi/logging.h
/usr/src/linux-headers-5.2.0-kali2-amd64/include/config/xen/mce/log.h
/usr/src/linux-headers-5.2.0-kali2-amd64/include/config/mouse/ps2/logips2pp.h
/usr/src/linux-headers-5.2.0-kali2-amd64/include/config/message/loglevel
/usr/src/linux-headers-5.2.0-kali2-amd64/include/config/console/loglevel
/usr/src/linux-headers-5.2.0-kali2-amd64/include/config/bridge/ebt/log.h
/usr/src/linux-headers-5.2.0-kali2-amd64/include/config/netfilter/netlink/log.h
/usr/src/linux-headers-5.2.0-kali2-amd64/include/config/netfilter/xt/target/log.h
/usr/lib/python3/dist-packages/pathod/log.py
/usr/lib/python3/dist-packages/c1219/access/log.py
/usr/lib/python3/dist-packages/pyqtgraph/examples/logAxis.py
/usr/lib/python3/dist-packages/sqlalchemy/log.py
/usr/lib/python3/dist-packages/scipy/io/matlab/tests/data/logical_sparse.mat
/usr/lib/python3/dist-packages/scipy/special/_precompute/loggamma.py
/usr/lib/python3/dist-packages/traitlets/log.py
/usr/lib/python3/dist-packages/orca/logger.py
```

(Рисунок 9)



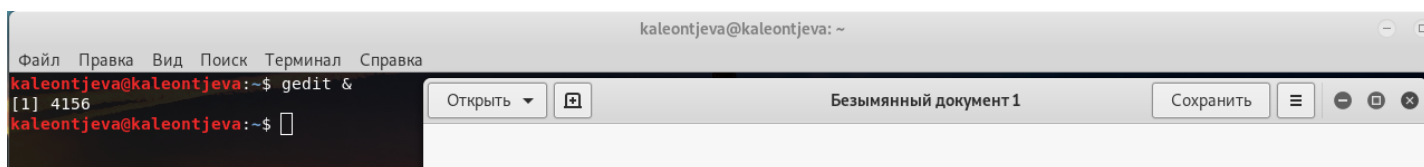
```
Файл Правка Вид Поиск Терминал Справка
kaleontjeva@kaleontjeva:~$ rm logfile
kaleontjeva@kaleontjeva:~$
```

(Рисунок 10)

7) См. пункт выше.

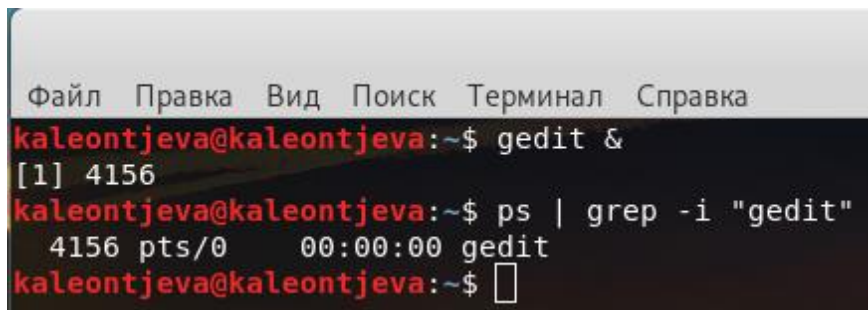
8) Запускаю редактор gedit в фоновом режиме командой «gedit &»

(Рисунок 11). После этого на экране появляется окно редактора.



(Рисунок 11)

9) Чтобы определить идентификатор процесса gedit, использую команду «ps | grep -i “gedit”» (Рисунок 12). Из рисунка видно, что наш процесс имеет PID 4156. Узнать идентификатор процесса можно также, используя команду «pgrep gedit» или «pidof gedit».

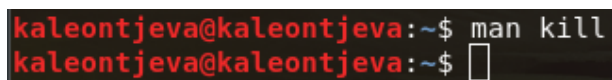


```
Файл Правка Вид Поиск Терминал Справка
kaleontjeva@kaleontjeva:~$ gedit &
[1] 4156
kaleontjeva@kaleontjeva:~$ ps | grep -i "gedit"
 4156 pts/0    00:00:00 gedit
kaleontjeva@kaleontjeva:~$
```

(Рисунок 12)

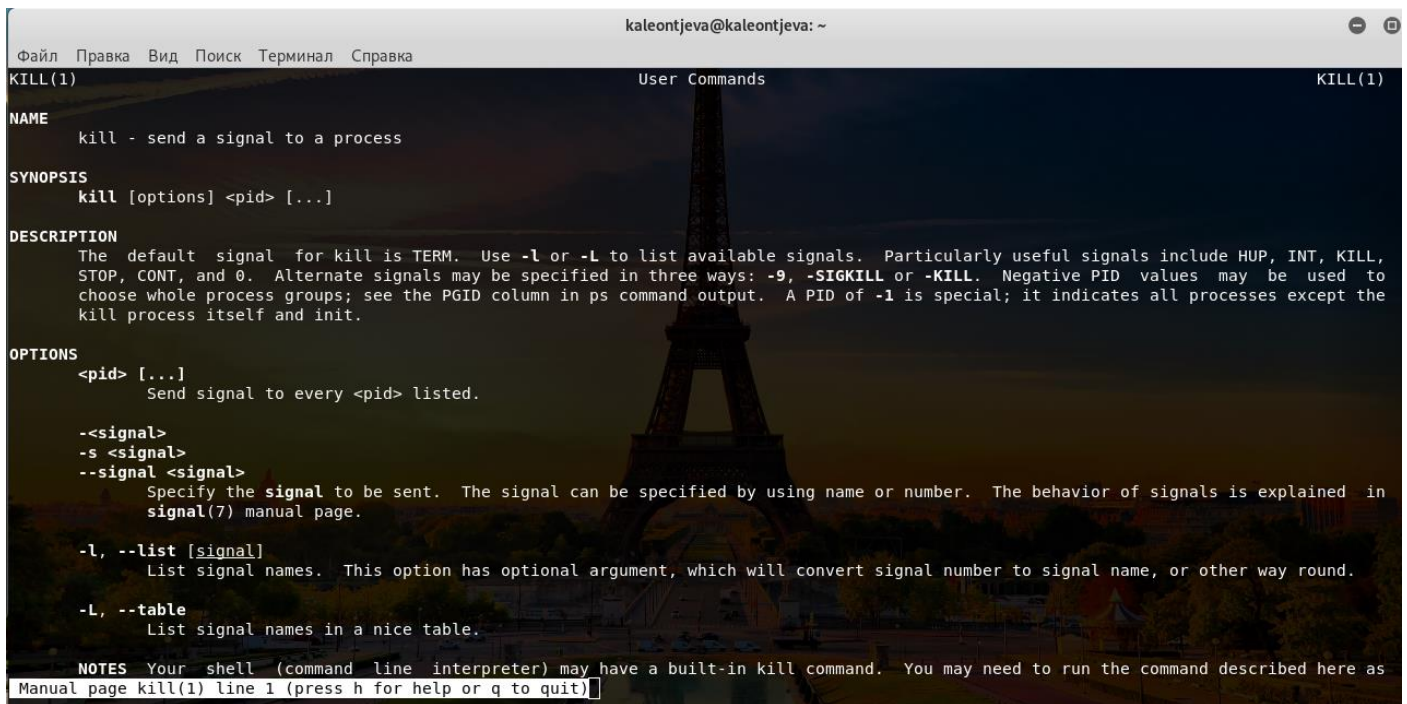
10) Прочитав информацию о команде kill с помощью команды «man kill», использую её для завершения процесса gedit (команда «kill 4156»)

(Рисунки 13, 14, 15).

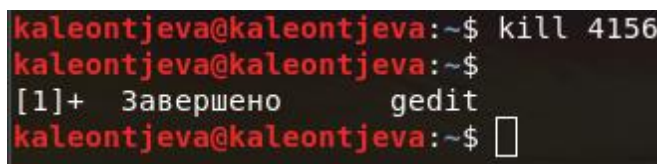


```
kaleontjeva@kaleontjeva:~$ man kill
kaleontjeva@kaleontjeva:~$
```

(Рисунок 13)



(Рисунок 14)



(Рисунок 15)

- 11) С помощью команд «man df» и «man du» узнаю информацию по необходимым командам и далее использую их (Рисунки 16, 17, 18, 19, 20).

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

Синтаксис: df [опции] устройство

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


```
Файл Правка Вид Поиск Терминал
kaleontjeva@kaleontjeva:~$ man df
kaleontjeva@kaleontjeva:~$ man du
kaleontjeva@kaleontjeva:~$
```

(Рисунок 16)

```
kaleontjeva@kaleontjeva: ~
Файл Правка Вид Поиск Терминал 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)

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

(Рисунок 17)

```
kaleontjeva@kaleontjeva: ~
Файл Правка Вид Поиск Терминал 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
      display the total disk usage of all files and directories of the command

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

(Рисунок 18)


```
kaleontjeva@kaleontjeva:~$ df
Файловая система 1К-блоков  Использовано  Доступно  Использовано%  Смонтировано в
udev                973076            0    973076            0% /dev
tmpfs               201396           9084    192312            5% /run
/dev/sda1          18447056        11242916    6244040           65% /
tmpfs               1006972            0    1006972            0% /dev/shm
tmpfs                5120              0     5120            0% /run/lock
tmpfs               1006972            0    1006972            0% /sys/fs/cgroup
tmpfs               201392            12    201380            1% /run/user/133
tmpfs               201392            56    201336            1% /run/user/1000
/dev/sr0            3627628          3627628            0          100% /media/cdrom0
kaleontjeva@kaleontjeva:~$
```

(Рисунок 19)

```
kaleontjeva@kaleontjeva:~$ du
4      ../cache/gnome-control-center/backgrounds
8      ../cache/gnome-control-center
4      ../cache/evolution/mail/trash
8      ../cache/evolution/mail
4      ../cache/evolution/sources/trash
8      ../cache/evolution/sources
4      ../cache/evolution/addressbook/trash
8      ../cache/evolution/addressbook
4      ../cache/evolution/tasks/trash
8      ../cache/evolution/tasks
4      ../cache/evolution/memos/trash
8      ../cache/evolution/memos
4      ../cache/evolution/calendar/trash
8      ../cache/evolution/calendar
52     ../cache/evolution
8632   ../cache/tracker
260    ../cache/mozilla/firefox/ou6m2n91.default/OfflineCache
80     ../cache/mozilla/firefox/ou6m2n91.default/thumbnails
10736  ../cache/mozilla/firefox/ou6m2n91.default/startupCache
4      ../cache/mozilla/firefox/ou6m2n91.default/safebrowsing/google4
728    ../cache/mozilla/firefox/ou6m2n91.default/safebrowsing
115284 ../cache/mozilla/firefox/ou6m2n91.default/cache2/entries
4      ../cache/mozilla/firefox/ou6m2n91.default/cache2/doomed
115452 ../cache/mozilla/firefox/ou6m2n91.default/cache2
127356 ../cache/mozilla/firefox/ou6m2n91.default
127360 ../cache/mozilla/firefox
127364 ../cache/mozilla
14076  ../cache/thumbnails/large
780    ../cache/thumbnails/normal
60     ../cache/thumbnails/fail/gnome-thumbnail-factory
64     ../cache/thumbnails/fail
14924  /cache/thumbnails
```

(Рисунок 20)

- 12) Вывожу имена всех директорий, имеющихя в моем домашнем каталоге с помощью команды «find ~ -type d», предварительно получив информацию с помощью команды «man find» (Рисунки 21, 22).

```
kaleontjeva@kaleontjeva: ~  
Файл Правка Вид Поиск Терминал 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, . is assumed.  
  
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.  
  
OPTIONS  
The -H, -L and -P options control the treatment of symbolic links. Command-line arguments following these are taken to be names of files or directories to be examined, up to the first argument that begins with .-, or the argument '.' or '!'. That argument and any following arguments are taken to be the expression describing what is to be searched for. If no paths are given, the current directory is used. If no expression is given, the expression -print is used (but you should probably consider using -print0 instead, anyway).  
  
This manual page talks about 'options' within the expression list. These options control the behaviour of find but are specified immediately after the last path name. The five 'real' options -H, -L, -P, -D and -O must appear before the first path name, if at all. A double dash -- can also be used to signal that any remaining arguments are not options (though ensuring that all start points begin with either ./ or / is generally safer if you use wildcards in the list of start points).  
  
Manual page find(1) line 1 (press h for help or q to quit)
```

(Рисунок 21)

```
kaleontjeva@kaleontjeva: ~  
Файл Правка Вид Поиск Терминал Справка  
kaleontjeva@kaleontjeva:~$ man find  
kaleontjeva@kaleontjeva:~$ find ~ -type d  
/home/kaleontjeva  
/home/kaleontjeva/.cache  
/home/kaleontjeva/.cache/gnome-control-center  
/home/kaleontjeva/.cache/gnome-control-center/backgrounds  
/home/kaleontjeva/.cache/evolution  
/home/kaleontjeva/.cache/evolution/mail  
/home/kaleontjeva/.cache/evolution/mail/trash  
/home/kaleontjeva/.cache/evolution/sources  
/home/kaleontjeva/.cache/evolution/sources/trash  
/home/kaleontjeva/.cache/evolution/addressbook  
/home/kaleontjeva/.cache/evolution/addressbook/trash  
/home/kaleontjeva/.cache/evolution/tasks  
/home/kaleontjeva/.cache/evolution/tasks/trash  
/home/kaleontjeva/.cache/evolution/memos  
/home/kaleontjeva/.cache/evolution/memos/trash  
/home/kaleontjeva/.cache/evolution/calendar  
/home/kaleontjeva/.cache/evolution/calendar/trash  
/home/kaleontjeva/.cache/tracker  
/home/kaleontjeva/.cache/mozilla  
/home/kaleontjeva/.cache/mozilla/firefox  
/home/kaleontjeva/.cache/mozilla/firefox/ou6m2n91.default  
/home/kaleontjeva/.cache/mozilla/firefox/ou6m2n91.default/OfflineCache  
/home/kaleontjeva/.cache/mozilla/firefox/ou6m2n91.default/thumbnails  
/home/kaleontjeva/.cache/mozilla/firefox/ou6m2n91.default/startupCache  
/home/kaleontjeva/.cache/mozilla/firefox/ou6m2n91.default/safebrowsing  
/home/kaleontjeva/.cache/mozilla/firefox/ou6m2n91.default/safebrowsing/google4  
/home/kaleontjeva/.cache/mozilla/firefox/ou6m2n91.default/cache2  
/home/kaleontjeva/.cache/mozilla/firefox/ou6m2n91.default/cache2/entries  
/home/kaleontjeva/.cache/mozilla/firefox/ou6m2n91.default/cache2/doomed  
/home/kaleontjeva/.cache/thumbnails  
/home/kaleontjeva/.cache/thumbnails/large
```

(Рисунок 22)

3. 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:

команда 1 | команда 2 (это означает, что вывод команды 1 передаётся 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 (`process ID`), 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
- -type f - искать только файлы
- -type d - 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 дамп 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 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 `pgrep` – 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. Вывод:** 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.