

Отчет по лабораторной работе №6

Дисциплина: Операционные системы

Астраханцева А. А.

Содержание

1	Цель работы	4
2	Задание	5
3	Теоретическое введение	6
4	Выполнение лабораторной работы	8
5	Выводы	19
6	Ответы на контрольные вопросы	20
	Список литературы	23

Список иллюстраций

4.1	Запись в file.txt названия файлов, содержащихся в каталоге /etc. .	9
4.2	Запись в file.txt названия файлов, содержащихся в домашнем каталоге	9
4.3	Демонстрация того, что файлы из домашнего каталога были записаны	10
4.4	Вывод имен всех файлов из file.txt, имеющих расширение .conf . .	11
4.5	Запись в файл conf.txt имен всех файлов из file.txt, имеющих расширение .conf	11
4.6	Поиск файлов, имена которых начинаются с символа с в домашнем каталоге	12
4.7	Поиск файлов, имена которых начинаются с символа с в домашнем каталоге (grep)	12
4.8	Вывод имен файлов из каталога /etc, начинающиеся с символа h. .	13
4.9	Запись имен файлов, которые начинаются с log в файл в фоновом режиме	13
4.10	Запись имен файлов, которые начинаются с log в файл в фоновом режиме	13
4.11	Запуск gedit в фоновом режиме	14
4.12	Определение идентификатора	14
4.13	man kill	14
4.14	Завершение процесса gedit	15
4.15	man df	15
4.16	df -vi	16
4.17	man du	16
4.18	du	17
4.19	du	17
4.20	Имена всех директорий, имеющих в домашнем каталоге	18

1 Цель работы

Ознакомление с инструментами поиска файлов и фильтрации текстовых данных. Приобретение практических навыков: по управлению процессами (и заданиями), по проверке использования диска и обслуживанию файловых систем.

2 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

Процесс в Linux (как и в UNIX) - это программа, которая выполняется в отдельном виртуальном адресном пространстве. Когда пользователь регистрируется в системе, автоматически создается процесс, в котором выполняется оболочка (shell), например, /bin/bash.

В Linux поддерживается классическая схема мультипрограммирования. Linux поддерживает параллельное (или квазипараллельного при наличии только одного процессора) выполнение процессов пользователя. Каждый процесс выполняется в собственном виртуальном адресном пространстве, т.е. процессы защищены друг от друга и крах одного процесса никак не повлияет на другие выполняющиеся процессы и на всю систему в целом. Один процесс не может прочитать что-либо из памяти (или записать в нее) другого процесса без “разрешения” на то другого процесса. Санкционированные взаимодействия между процессами допускаются системой.

Проверка и восстановление файловой системы Linux выполняется программой fsck. Перед проверкой файловая система должна быть смонтирована в режиме «только чтение».

Программа fsck умеет проверять и другие типы файловых систем, но для исправления ошибок лучше использовать родные для этой файловой системы программы, предназначенные для проверки и исправления ошибок. Программа fsck автоматически проверяет файловые системы при загрузке Linux в соответствии с указаниями, содержащимися в файле etc/fstab. Для программы fsck можно использовать опции

Перенаправление ввода - вывода

системе по умолчанию открыто три специальных потока:

- `stdin` — стандартный поток ввода (по умолчанию: клавиатура), файловый дескриптор 0;
- `stdout` — стандартный поток вывода (по умолчанию: консоль), файловый дескриптор 1;
- `stderr` — стандартный поток вывод сообщений об ошибках (по умолчанию: консоль), файловый дескриптор 2.

Большинство используемых в консоли команд и программ записывают результаты своей работы в стандартный поток вывода `stdout`. Например, команда `ls` выводит в стандартный поток вывода (консоль) список файлов в текущей директории. Потоки вывода и ввода можно перенаправлять на другие файлы или устройства. Проще всего это делается с помощью символов `>`, `»`, `<`, `«`.

4 Выполнение лабораторной работы

2. Запись в файл file.txt названия файлов, содержащихся в каталоге /etc. (Результат команды `ls /etc` записывается в файл file.txt с помощью “>”)(рис. 4.1).


```

[aaastrakhantseva@aaastrakhantseva ~]$ ls /etc > file.txt
[aaastrakhantseva@aaastrakhantseva ~]$ cat file.txt
abrt
adjtime
aliases
alsa
alternatives
anaconda
anthy-unicode.conf
appstream.conf
asound.conf
audit
authselect
avahi
bash_completion.d
bashrc
bindresvport.blacklist
binfmt.d
bluetooth
brlapi.key
brltty
brltty.conf
ceph
chkconfig.d
chromium
chrony.conf
chrony.keys
cifs-utils
containers
crypto-policies
crypttab
csh.cshrc
csh.login
cups
cupshelpers
dbus-1
dconf
debuginfod
default
depmod.d

```

Рис. 4.1: Запись в file.txt названия файлов, содержащихся в каталоге /etc.

После этого дописываю в этот же файл названия файлов, содержащихся в домашнем каталоге (рис. 4.2 - 4.3).

```

zls=rose
[aaastrakhantseva@aaastrakhantseva ~]$ ls ~/ >> file.txt
[aaastrakhantseva@aaastrakhantseva ~]$ cat file.txt
abrt

```

Рис. 4.2: Запись в file.txt названия файлов, содержащихся в домашнем каталоге

```
xattr.conf
xdg
xml
yum.repos.d
zfs-fuse
australia
bin
equiplist2
feathers
file.txt
may
monthly
my_os
newdir
os-intro
play
reports
ski.places
work
Видео
Документы
Загрузки
Изображения
Музыка
Общедоступные
Рабочий стол
Шаблоны
[aaastrakhantseva@aaastrakhantseva ~]$
```

Рис. 4.3: Демонстрация того, что файлы из домашнего катлога были записаны

3. С помощью команды `grep` вывожу имена всех файлов из `file.txt`, имеющих расширение `.conf` (рис. 4.4)

```
[aaastrakhantseva@aaastrakhantseva ~]$ grep .conf file.txt
anthy-unicode.conf
appstream.conf
asound.conf
brltty.conf
chkconfig.d
chrony.conf
dconf
dley-na-renderer-service.conf
dley-na-server-service.conf
dnsmasq.conf
dracut.conf
dracut.conf.d
extlinux.conf
fprintd.conf
fuse.conf
host.conf
idmapd.conf
jwhois.conf
kdump.conf
krb5.conf
krb5.conf.d
```

Рис. 4.4: Вывод имен всех файлов из file.txt, имеющих расширение .conf

После чего записываю их в новый текстовый файл conf.txt. с помощью команды grep и символа “>” (рис. 4.5)

```
[aaastrakhantseva@aaastrakhantseva ~]$ grep "\.conf" file.txt > conf.txt
[aaastrakhantseva@aaastrakhantseva ~]$ cat conf.txt
anthy-unicode.conf
appstream.conf
asound.conf
brltty.conf
chrony.conf
dley-na-renderer-service.conf
dley-na-server-service.conf
dnsmasq.conf
dracut.conf
dracut.conf.d
extlinux.conf
fprintd.conf
fuse.conf
host.conf
idmapd.conf
jwhois.conf
kdump.conf
krb5.conf
krb5.conf.d
```

Рис. 4.5: Запись в файл conf.txt имен всех файлов из file.txt, имеющих расширение .conf

4. Определите, какие файлы в вашем домашнем каталоге имеют имена, начинавшиеся с символа `c`? Предложите несколько вариантов, как это сделать.

Можно сделать это с помощью команды `find`, выглядеть это будет так: (рис. 4.6)

```
[aaastrakhantseva@aaastrakhantseva ~]$ find ~ -name "c*" -print
/home/aaastrakhantseva/.mozilla/firefox/82o4k1zk.default-release/crashes
/home/aaastrakhantseva/.mozilla/firefox/82o4k1zk.default-release/compatibility.ini
/home/aaastrakhantseva/.mozilla/firefox/82o4k1zk.default-release/cookies.sqlite
/home/aaastrakhantseva/.mozilla/firefox/82o4k1zk.default-release/cert9.db
/home/aaastrakhantseva/.mozilla/firefox/82o4k1zk.default-release/storage/permanent/chrome
/home/aaastrakhantseva/.mozilla/firefox/82o4k1zk.default-release/storage/default/https+++www.office.com/cache
/home/aaastrakhantseva/.mozilla/firefox/82o4k1zk.default-release/storage/default/https+++www.office.com/cache/caches.sqlite
/home/aaastrakhantseva/.mozilla/firefox/82o4k1zk.default-release/storage/default/https+++www.youtube.com*partitionKey=%28
/home/aaastrakhantseva/.mozilla/firefox/82o4k1zk.default-release/storage/default/https+++www.youtube.com/cache
/home/aaastrakhantseva/.mozilla/firefox/82o4k1zk.default-release/storage/default/https+++vk.com/cache/caches.sqlite
/home/aaastrakhantseva/.mozilla/firefox/82o4k1zk.default-release/storage/default/https+++www.youtube.com/cache
/home/aaastrakhantseva/.mozilla/firefox/82o4k1zk.default-release/storage/default/https+++www.youtube.com/cache/caches.sqlite
/home/aaastrakhantseva/.mozilla/firefox/82o4k1zk.default-release/storage/default/https+++rutube.ru/cache
/home/aaastrakhantseva/.mozilla/firefox/82o4k1zk.default-release/storage/default/https+++rutube.ru/cache/caches.sqlite-wa
/home/aaastrakhantseva/.mozilla/firefox/82o4k1zk.default-release/storage/default/https+++rutube.ru/cache/context_open.mar
/home/aaastrakhantseva/.mozilla/firefox/82o4k1zk.default-release/storage/default/https+++habr.com/cache
/home/aaastrakhantseva/.mozilla/firefox/82o4k1zk.default-release/storage/default/https+++habr.com/cache/caches.sqlite
```

Рис. 4.6: Поиск файлов, имена которых начинаются с символа `c` в домашнем каталоге

Можно использовать `grep`: (рис. 4.7)

```
[aaastrakhantseva@aaastrakhantseva ~]$ ls | grep c
conf.txt
ski.places
[aaastrakhantseva@aaastrakhantseva ~]$ ls -a | grep c
.bashrc
.cache
.config
conf.txt
.gitconfig
.local
ski.places
.vboxclient-clipboard.pid
.vboxclient-draganddrop.pid
.vboxclient-seamless.pid
[aaastrakhantseva@aaastrakhantseva ~]$
```

Рис. 4.7: Поиск файлов, имена которых начинаются с символа `c` в домашнем каталоге (`grep`)

5. Выведите на экран (по странично) имена файлов из каталога `/etc`, начинающиеся с символа `h`.

Сделать это можно с помощью команды `find` (рис. 4.8)

```
[aaastrakhantseva@aaastrakhantseva report]$ find /etc -name "h*" -print
find: '/etc/audit': Отказано в доступе
/etc/avahi/hosts
/etc/brltty/Contraction/ha.ctb
/etc/brltty/Input/bm/horizontal.kti
/etc/brltty/Input/hd
/etc/brltty/Input/hm
/etc/brltty/Input/ht
/etc/brltty/Input/hw
/etc/brltty/Text/he.ttb
/etc/brltty/Text/hi.ttb
/etc/brltty/Text/hr.ttb
/etc/brltty/Text/hu.ttb
/etc/brltty/Text/hy.ttb
/etc/containers/oci/hooks.d
```

Рис. 4.8: Вывод имен 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.

С помощью команды `find` ищем имена файлов, которые начинаются с `log` и записываем их в файл `logfile` с помощью символа `>`. Чтобы процесс выполнялся в фоновом режиме нужно добавить `&` (рис. 4.9)

```
[aaastrakhantseva@aaastrakhantseva ~]$ find -name "lof*" - print > ~/logfile &
[1] 4266
find: paths must precede expression: `&'
```

Рис. 4.9: Запись имен файлов, которые начинаются с `log` в файл в фоновом режиме

7. Удалите файл `~/logfile`. (рис. 4.9)

Делаю это с помощью команды `rm` (рис. 4.10).

```
[aaastrakhantseva@aaastrakhantseva ~]$ rm ~/logfile
[aaastrakhantseva@aaastrakhantseva ~]$
```

Рис. 4.10: Запись имен файлов, которые начинаются с `log` в файл в фоновом режиме

8. Запустите из консоли в фоновом режиме редактор `gedit`.

Нужно просто написать: `gedit &`. В следующей строке выведен идентификатор процесса `gedit`.(рис. 4.11).

```
[aaastrakhantseva@aaastrakhantseva ~]$ gedit &
[1] 4297
[aaastrakhantseva@aaastrakhantseva ~]$
```

Рис. 4.11: Запуск `gedit` в фоновом режиме

9. Определите идентификатор процесса `gedit`, используя команду `ps`, конвейер и фильтр `grep`. Как ещё можно определить идентификатор процесса?

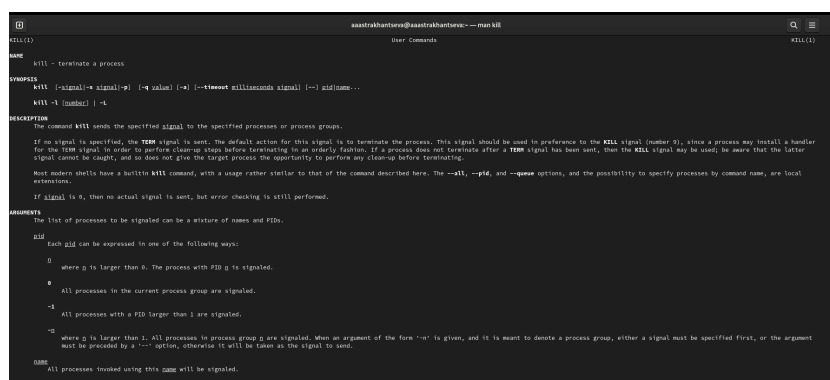
Использование команды `ps`, конвейера и фильтра `grep`. Ещё есть вариант использовать только команду `pgrep` (рис. 4.12).

```
[aaastrakhantseva@aaastrakhantseva ~]$ ps aux | grep gedit
aaastra+  4297  0.3  1.9 859400 77952 pts/0    Sl   15:13   0:00  gedit
aaastra+  4341  0.0  0.0 222192 2304 pts/0    S+   15:16   0:00  grep --color=auto gedit
[aaastrakhantseva@aaastrakhantseva ~]$ pgrep gedit
4297
```

Рис. 4.12: Определение идентификатора

10. Прочтите справку (`man`) команды `kill`, после чего используйте её для завершения процесса `gedit`.

Команда `man kill` (рис. 4.13)



```
kill(1)
User Commands
kill(1)

NAME
    kill - terminate a process

SYNOPSIS
    kill [-signal=-s signal] [-q value] [-w] [--timeout milliseconds signal] [-r] pidname...
    kill -l [number] | -t

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 used in preference to the KILL signal (number 9), since a process may install a handler for the TERM signal in order to perform cleanup steps before terminating in an orderly fashion. If a process does not terminate after a TERM signal has been sent, then the KILL signal may be used; be aware that the latter signal cannot be caught, and so does not give the target process the opportunity to perform any cleanup 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 --signal options, and the possibility to specify processes by command name, are local extensions.

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

ARGUMENTS
    The list of processes to be signaled can be a mixture of names and PIDs.

    pid
        Each pid can be expressed in one of the following ways:
        0       where p is larger than 0. The process with PID p is signaled.
        p       All processes in the current process group are signaled.
        -p       All processes with a PID larger than 1 are signaled.
        -s       where p is larger than 2. All processes in process group p are signaled. When an argument of the form '-' is given, and it is meant to denote a process group, either a signal must be specified first, or the argument must be preceded by a '-' option, otherwise it will be taken as the signal to send.

    name
        All processes invoked using this name will be signaled.
```

Рис. 4.13: `man kill`

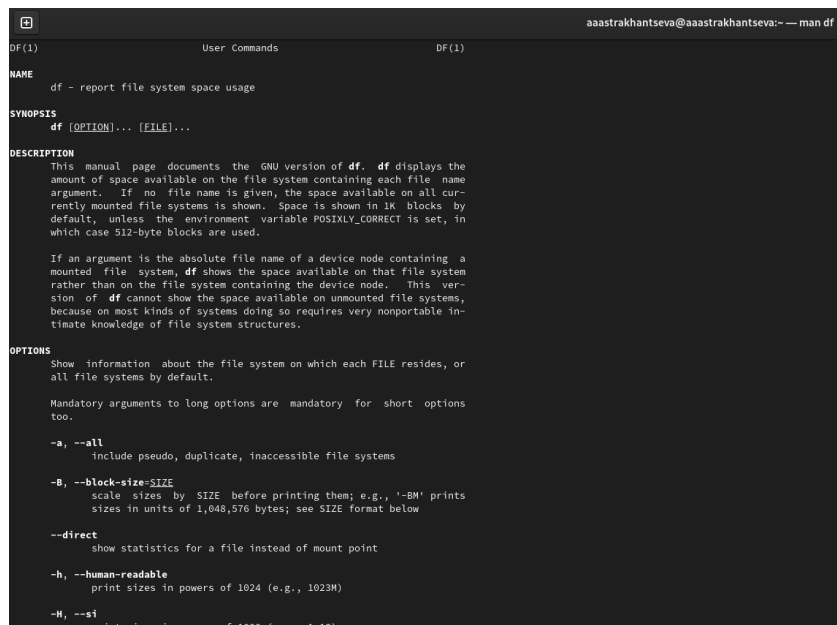
Завершение процесса gedit (рис. 4.14)

```
[aaastrakhantseva@aaastrakhantseva ~]$ man kill
[aaastrakhantseva@aaastrakhantseva ~]$ kill 4296
bash: kill: (4296) - Операция не позволена
[aaastrakhantseva@aaastrakhantseva ~]$ kill 4297
[aaastrakhantseva@aaastrakhantseva ~]$
```

Рис. 4.14: Завершение процесса gedit

11. Выполните команды `df` и `du`, предварительно получив более подробную информацию об этих командах, с помощью команды `man`.

Команда `man df` (рис. 4.15)



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

NAME
  df - report file system space usage

SYNOPSIS
  df [OPTION]... [FILE]...

DESCRIPTION
  This manual page documents the GNU version of df. df displays the
  amount of space available on the file system containing each file name
  argument. If no file name is given, the space available on all cur-
  rently mounted file systems is shown. Space is shown in 1K blocks by
  default, unless the environment variable POSIXLY_CORRECT is set, in
  which case 512-byte blocks are used.

  If an argument is the absolute file name of a 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 ver-
  sion of df cannot show the space available on unmounted file systems,
  because on most kinds of systems doing so requires very nonportable in-
  timate 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

  --direct
      show statistics for a file instead of mount point

  -h, --human-readable
      print sizes in powers of 1024 (e.g., 1023M)

  -H, --si
      print sizes in powers of 1000 (e.g., 1.1G)
```

Рис. 4.15: `man df`

Выполнение команды `df` (рис. 4.16)

```
[aaastrakhantseva@aaastrakhantseva ~]$ df -vi
Файловая система    Инодов  ИИспользовано  ИСвободно  ИИспользовано%  Смонтировано в
devtmpfs             1048576        469    1048107            1% /dev
tmpfs                 501462         15    501447            1% /dev/shm
tmpfs                 819200        890    818310            1% /run
/dev/sda2              0              0          0             - /
tmpfs                 1048576        45    1048531            1% /tmp
/dev/sda2              0              0          0             - /home
/dev/sda1              65536         456    65080            1% /boot
tmpfs                 100292        180    100112            1% /run/user/1000
[aaastrakhantseva@aaastrakhantseva ~]$
```

Рис. 4.16: df -vi

Команда man du (рис. 4.17)

```
aaastrakhantseva@aaastrakhantseva ~$ man du
NAME
    du - estimate file space usage

SYNOPSIS
    du [OPTION]... [FILE]...
    du [OPTION]... --files-from=F

DESCRIPTION
    Summarize device usage of the set of FILES, recursively for directories.
    Mandatory arguments to long options are mandatory for short options too.

    -B, --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 device 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'

    -e, --total
        produce a grand total

    -D, --dereference-args
        dereference only symlinks that are listed on the command line

    -d, --max-depth=N
        print the total for a directory (or file, with --all) only if it is N or fewer levels below the command line argument; --max-depth=0 is the same as --summary

    --files-from=F
        summarize device usage of the NUL-terminated file names specified in file F; if F is -, then read names from standard input

    -H
        equivalent to --dereference-args (-D)

    -h, --human-readable
        print sizes in human readable format (e.g., 1.23M, 2.1G)

Manual page du(1) line 8, version 3.25, 2015-09-16, 14.01.2016
```

Рис. 4.17: man du

Выполнение команды du. На скриншоте выше приведен вывод команды (рис. 4.18)


```

0      ./ski.places/equipment
0      ./ski.places/plans
8      ./ski.places
0      ./equiplist2
0      ./australia
0      ./play/file.old
0      ./play/games/play/file.old
0      ./play/games/play
0      ./play/games
0      ./play
0      ./my_os
0      ./feathers
4      ./file.txt
4      ./conf.txt
4      ./lessht
1900628 .
[aaastrakhantseva@aaastrakhantseva ~]$ du

```

Рис. 4.18: du

12. Воспользовавшись справкой команды `find`, выведите имена всех директорий, имеющих- щихся в вашем домашнем каталоге.

Использую `man find` (рис. 4.19)

```

'0' for gbytes (010, units of 1024 * 1024 * 1024 = 1073741824 bytes)

The size is simply the st_size member of the struct stat populated by the stat (or stat64) system call, rounded up as shown above. In other words, it's consistent with the result you get for 'ls -l'. Bear in mind that the 'm' and 'M' format specifiers of printf handle sparse files differently. The 'b' suffix always denotes 512-byte blocks and never 1024-byte blocks, which is different to the behaviour of 'du'.

The '+' and '-' prefixes signify greater than and less than, as usual; i.e., an exact size of g units does not match. Bear in mind that the size is rounded up to the next unit. Therefore -size -100 is not equivalent to -size -10485760. The former only matches empty files, the latter matches files from 0 to 1,048,575 bytes.

-true Always true.

-type t
File is of type t:
b  block (buffered) special
c  character (unbuffered) special
d  directory
p  named pipe (fifo)
f  regular file
l  symbolic link; this is never true if the -L option or the -follow option is in effect, unless the symbolic link is broken. If you want to search for symbolic links when -L is in effect, use -xtype.
s  socket
D  door (Solaris)

```

Рис. 4.19: du

С помощью опции `-type d` вывожу имена всех директорий, имеющих- щихся в домашнем каталоге (рис. 4.20)

```

[aaastrakhantseva@aaastrakhantseva ~]$ find -type d
./
./mozilla
./mozilla/extensions
./mozilla/extensions/{ec8030f7-c20a-464f-9b0e-13a3a9e97384}
./mozilla/plugins
./mozilla/firefox
./mozilla/firefox/Crash Reports
./mozilla/firefox/Crash Reports/events
./mozilla/firefox/Pending Pings
./mozilla/firefox/8204k1zk.default-release
./mozilla/firefox/8204k1zk.default-release/minidumps
./mozilla/firefox/8204k1zk.default-release/crashes
./mozilla/firefox/8204k1zk.default-release/crashes/events
./mozilla/firefox/8204k1zk.default-release/security_state
./mozilla/firefox/8204k1zk.default-release/storage
./mozilla/firefox/8204k1zk.default-release/storage/permanent
./mozilla/firefox/8204k1zk.default-release/storage/permanent/chrome
./mozilla/firefox/8204k1zk.default-release/storage/permanent/chrome/ldb
./mozilla/firefox/8204k1zk.default-release/storage/permanent/chrome/ldb/3870112724rsegmnoittet-es.files
./mozilla/firefox/8204k1zk.default-release/storage/permanent/chrome/ldb/3870112724rsegmnoittet-es.files/journals
./mozilla/firefox/8204k1zk.default-release/storage/permanent/chrome/ldb/3561288849sdhlie.files
./mozilla/firefox/8204k1zk.default-release/storage/permanent/chrome/ldb/1451318868ntouromlalnodry--epcr.files
./mozilla/firefox/8204k1zk.default-release/storage/permanent/chrome/ldb/2823318777ntouromlalnodry--naod.files
./mozilla/firefox/8204k1zk.default-release/storage/permanent/chrome/ldb/1657114595AmcateirvtiSty.files
./mozilla/firefox/8204k1zk.default-release/storage/permanent/chrome/ldb/2918063365piupsah.files
./mozilla/firefox/8204k1zk.default-release/storage/temporary

```

Рис. 4.20: Имена всех директорий, имеющих в домашнем каталоге

5 Выводы

В ходе выполнения лабораторной работы я ознакомилась с инструментами поиска файлов и фильтрации текстовых данных. Приобрела практические навыки: по управлению процессами (и заданиями), по проверке использования диска и 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 `»`.

Операция `>` используется для перенаправления ввода/вывода, а `»` используется как 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?

Выполняющаяся программа называется в Linux процессом. Все процессы система регистрирует в таблице процессов, присваивая каждому уникальный номер — идентификатор процесса.

Компьютерная программа сама по себе — лишь пассивная последовательность инструкций. В то время как процесс — непосредственное выполнение этих инструкций. Также, процессом называют выполняющуюся программу и все её элементы: адресное пространство, глобальные переменные, регистры, стек, открытые файлы и так далее.

5. Что такое PID и GID?

Каждому процессу Linux или Unix или выполняемой программе, автоматически присваивается идентификационный номер уникального процесса (PID). PID автоматически присваивает номер для каждого процесса в системе.

GID— это идентификационный номер группы данного процесса. Допустимые идентификаторы групп указываются в файле/ etc / group и в поле GID файла/etc / passwd . Когда процесс запускается, его GID устанавливается равным GID родительского процесса.

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?

top - интерактивный просмотрщик процессов. htop аналог top.

Топ показывает все запущенные процессы и выводит важную информацию о нагрузке на производительные мощности компьютера или сервера. Также эта утилита даёт возможность завершить ненужные процессы.

Htop – просмотрщик процессов подобный top, но позволяющий прокручивать список процессов вертикально и горизонтально, чтобы видеть их полные параметры запуска. Управление процессами (остановка, изменение приоритета) может выполняться без ручного ввода их идентификаторов.

8. Назовите и дайте характеристику команде поиска файлов. Приведите примеры использования этой команды.

Команда `find` используется для поиска и отображения на экран имён файлов, соответствующих заданной строке символов. Формат команды:

```
find путь [-опции]
```

Для примера: Вывести на экран имена файлов из вашего домашнего каталога и его подкаталогов, начинающихся на `f`:

```
find ~ -name "f*" -print
```

Здесь `~` — обозначение вашего домашнего каталога, `-name` — после этой опции указывается имя файла, который нужно найти, `"f*"` — строка символов, определяющая имя файла, `-print` — опция, задающая вывод результатов поиска на экран.

9. Можно ли по контексту (содержанию) найти файл? Если да, то как?

Найти в текстовом файле указанную строку символов позволяет команда `grep`. Формат команды:

```
grep строка имя_файла
```

Пример:

Показать строки во всех файлах в вашем домашнем каталоге с именами, начинающимися на `f`, в которых есть слово `begin`: `grep begin f*`

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?

`kill%` номер задачи

Список литературы

- [illegible]