

Отчёт по лабораторной работе номер 7

Операционные системы

Нитусова Диана Денисовна

Содержание

1	Цель работы	5
2	Задание	6
3	Выполнение лабораторной работы	7
4	Контрольные вопросы	14
5	Выводы	18

List of Tables

List of Figures

3.1	Рисунок 1	7
3.2	Рисунок 2	8
3.3	Рисунок 3	8
3.4	Рисунок 4	8
3.5	Рисунок 5	9
3.6	Рисунок 6	9
3.7	Рисунок 7	9
3.8	Рисунок 8	10
3.9	Рисунок 9	10
3.10	Рисунок 10	10
3.11	Рисунок 11	11
3.12	Рисунок 12	12
3.13	Рисунок 13	12
3.14	Рисунок 14	12
3.15	Рисунок 15	13

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.

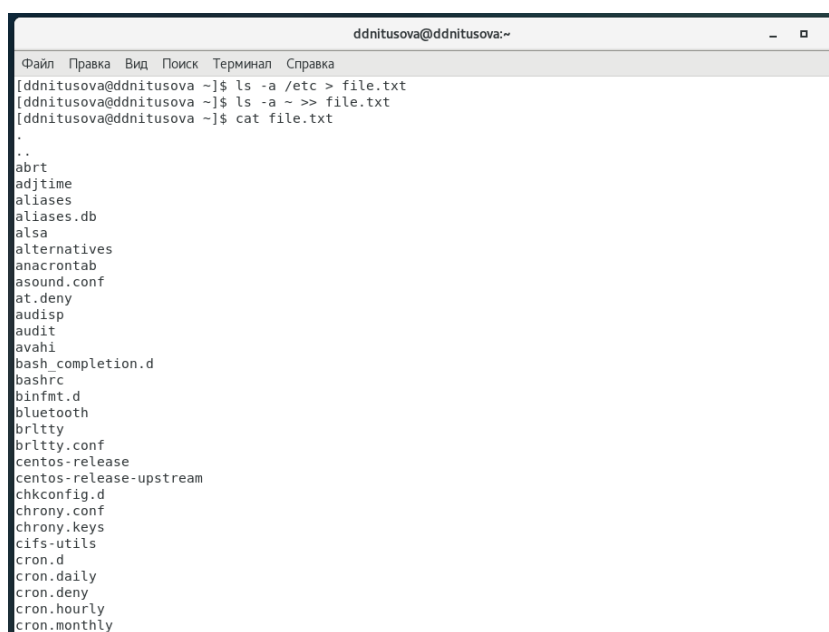
2 Задание

Ознакомиться с поиском файлов.

3 Выполнение лабораторной работы

Осуществляю вход в систему, используя свои логин и пароль.

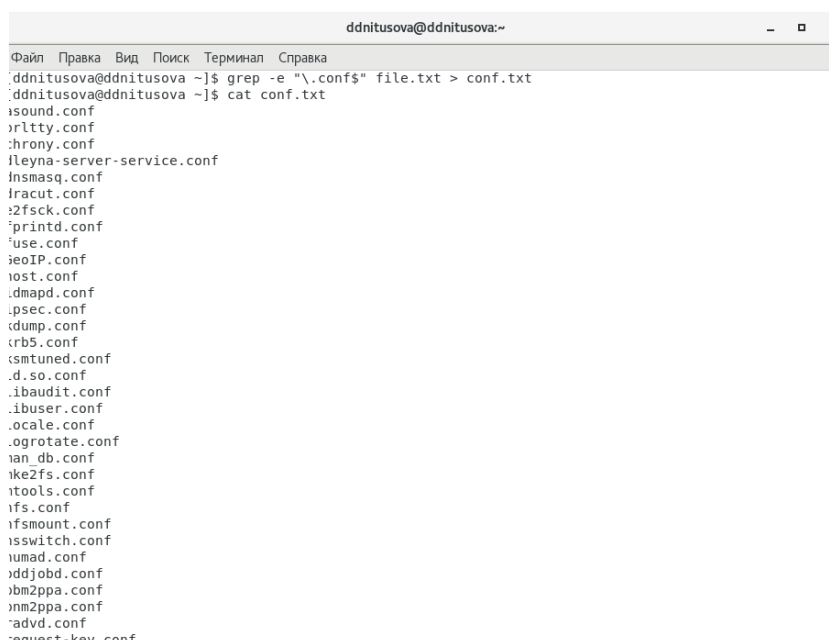
Для того, чтобы записать в файл file.txt названия файлов, содержащихся в каталоге /etc, использую команду «ls -a /etc > file.txt». Далее с помощью команды «ls -a ~ >> file.txt» дописываю в этот же файл названия файлов, содержащихся в моем домашнем каталоге. Командой «cat file.txt» просматриваю файл. (рис. 3.1)



```
ddnitusova@ddnitusova:~  
Файл  Правка  Вид  Поиск  Терминал  Справка  
[ddnitusova@ddnitusova ~]$ ls -a /etc > file.txt  
[ddnitusova@ddnitusova ~]$ ls -a ~ >> file.txt  
[ddnitusova@ddnitusova ~]$ cat file.txt  
.  
..  
abrt  
adjtime  
aliases  
aliases.db  
alsa  
alternatives  
anacrontab  
asound.conf  
at.deny  
audisp  
audit  
avahi  
bash_completion.d  
bashrc  
binfmt.d  
bluetooth  
brltty  
brltty.conf  
centos-release  
centos-release-upstream  
chkconfig.d  
chrony.conf  
chrony.keys  
cifs-utils  
cron.d  
cron.daily  
cron.deny  
cron.hourly  
cron.monthly
```

Figure 3.1: Рисунок 1

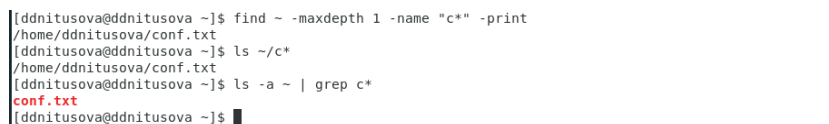
Вывожу имена всех файлов из file.txt, имеющих расширение .conf и записываю их в новый текстовый файл conf.txt с помощью команды «grep -e '.conf\$' file.txt > conf.txt». (рис. 3.2)



```
ddnitusova@ddnitusova:~$ grep -e "\.conf$" file.txt > conf.txt
ddnitusova@ddnitusova ~]$ cat conf.txt
isound.conf
rltty.conf
hrony.conf
fleya-server-service.conf
fmsmasq.conf
iracut.conf
ifscck.conf
iprintd.conf
iuse.conf
ieoIP.conf
iost.conf
idmapd.conf
ipsec.conf
idump.conf
crb5.conf
csmuned.conf
.d.so.conf
.ibaudit.conf
.ibuser.conf
.ocale.conf
.ogrotate.conf
ian_db.conf
ike2fs.conf
itools.conf
ifs.conf
ifsmount.conf
isswitch.conf
iumad.conf
iddjobd.conf
ibm2ppa.conf
inm2ppa.conf
iadvd.conf
request-key.conf
```

Figure 3.2: Рисунок 2

Определить, какие файлы в моем домашнем каталоге имеют имена, начинающиеся с символа с, можно несколькими командами: «find ~ -maxdepth 1 -name “с*” -print», «ls ~/с *» и «ls -a ~ | grep с*». (рис. 3.3)



```
[ddnitusova@ddnitusova ~]$ find ~ -maxdepth 1 -name "с*" -print
/home/ddnitusova/conf.txt
[ddnitusova@ddnitusova ~]$ ls ~/с*
/home/ddnitusova/conf.txt
[ddnitusova@ddnitusova ~]$ ls -a ~ | grep с*
conf.txt
[ddnitusova@ddnitusova ~]$
```

Figure 3.3: Рисунок 3

Чтобы вывести на экран (по странично) имена файлов из каталога /etc, начинающиеся с символа h, воспользуемся командой «find /etc-maxdepth 1 -name “h*” | less». (рис. 3.4) (рис. 3.5)



```
/etc/host.conf
/etc/hosts
/etc/hosts.allow
/etc/hosts.deny
/etc/hp
/etc/hostname
(END)
```

Figure 3.4: Рисунок 4


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

Figure 3.5: Рисунок 5

Запускаю в фоновом режиме процесс, который будет записывать в файл ~/logfile файлы, имена которых начинаются с log, используя команду «find / -name “log*” > logfile &» (рис. 3.6). Далее удаляю файл ~/logfile командой «rm logfile» (рис. 3.7).

```
ddnitusova@ddnitusova:~$ find / -name "log*" > logfile &
1) 6096
ind: '/boot/efi': Отказано в доступе
ddnitusova@ddnitusova ~]$ find: '/boot/grub2': Отказано в доступе
ind: '/proc/tty/driver': Отказано в доступе
ind: '/proc/1/task/1/fd': Отказано в доступе
ind: '/proc/1/task/1/fdinfo': Отказано в доступе
ind: '/proc/1/task/1/ns': Отказано в доступе
ind: '/proc/1/fd': Отказано в доступе
ind: '/proc/1/map_files': Отказано в доступе
ind: '/proc/1/fdinfo': Отказано в доступе
ind: '/proc/1/ns': Отказано в доступе
ind: '/proc/2/task/2/fd': Отказано в доступе
ind: '/proc/2/task/2/fdinfo': Отказано в доступе
ind: '/proc/2/task/2/ns': Отказано в доступе
ind: '/proc/2/fd': Отказано в доступе
ind: '/proc/2/map_files': Отказано в доступе
ind: '/proc/2/fdinfo': Отказано в доступе
ind: '/proc/2/ns': Отказано в доступе
ind: '/proc/4/task/4/fd': Отказано в доступе
ind: '/proc/4/task/4/fdinfo': Отказано в доступе
ind: '/proc/4/task/4/ns': Отказано в доступе
ind: '/proc/4/fd': Отказано в доступе
ind: '/proc/4/map_files': Отказано в доступе
ind: '/proc/4/fdinfo': Отказано в доступе
ind: '/proc/4/ns': Отказано в доступе
ind: '/proc/6/task/6/fd': Отказано в доступе
ind: '/proc/6/task/6/fdinfo': Отказано в доступе
ind: '/proc/6/task/6/ns': Отказано в доступе
ind: '/proc/6/fd': Отказано в доступе
ind: '/proc/6/map_files': Отказано в доступе
ind: '/proc/6/fdinfo': Отказано в доступе
ind: '/proc/6/ns': Отказано в доступе
ind: '/proc/7/task/7/fd': Отказано в доступе
ind: '/proc/7/task/7/fdinfo': Отказано в доступе
```

Figure 3.6: Рисунок 6

```
/opt/VBoxGuestAdditions-6.1.20/src/VBoxGuest-6.1.20/VBoxGuest/Common/log/log.c
/opt/VBoxGuestAdditions-6.1.20/src/VBoxGuest-6.1.20/VBoxGuest/common/log/logfor
/opt/VBoxGuestAdditions-6.1.20/src/VBoxGuest-6.1.20/VBoxsf/include/VBox/log.h
/opt/VBoxGuestAdditions-6.1.20/src/VBoxGuest-6.1.20/VBoxsf/include/iprt/log.h
ddnitusova@ddnitusova ~]$ rm logfile
ddnitusova@ddnitusova ~]$
```

Figure 3.7: Рисунок 7

Запускаю редактор gedit в фоновом режиме командой «gedit &». После этого на экране появляется окно редактора.

Чтобы определить идентификатор процесса `gedit`, использую команду «`ps | grep -i "gedit"`». Из рисунка видно, что наш процесс имеет PID 6490. (рис. 3.8)

```
[ddnitusova@ddnitusova ~]$ ps | grep -i "gedit"
[1]+  Done                  gedit
[ddnitusova@ddnitusova ~]$ pgrep gedit
[ddnitusova@ddnitusova ~]$ gedit &
[1] 6490
[ddnitusova@ddnitusova ~]$ ps | grep -i "gedit"
 6490 pts/0    00:00:00 gedit
[ddnitusova@ddnitusova ~]$ pgrep gedit
6490
[ddnitusova@ddnitusova ~]$ pidof gedit
6490
[ddnitusova@ddnitusova ~]$
```




Figure 3.8: Рисунок 8

Прочитав информацию о команде `kill` с помощью команды «`man kill`», использую её для завершения процесса `gedit` (команда «`kill 6490`»). (рис. 3.9) (рис. 3.10)

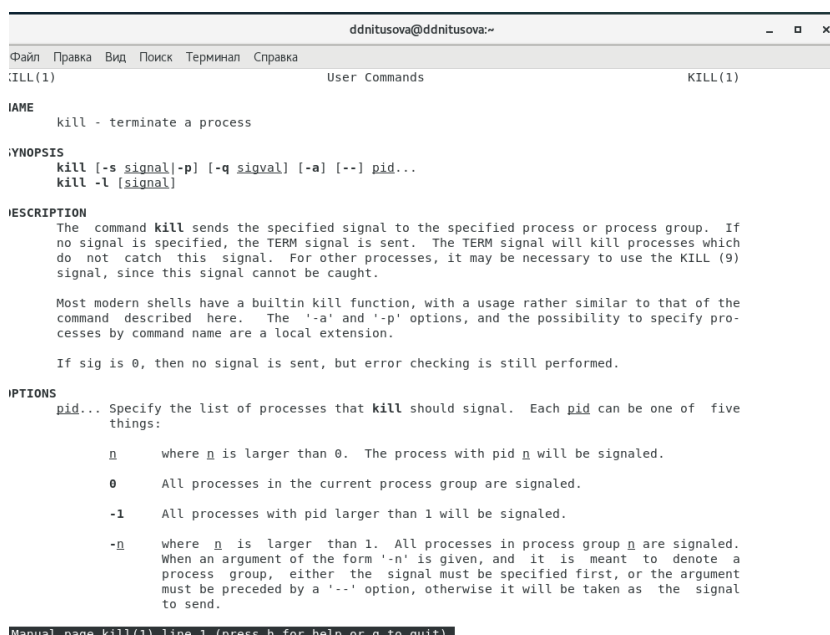


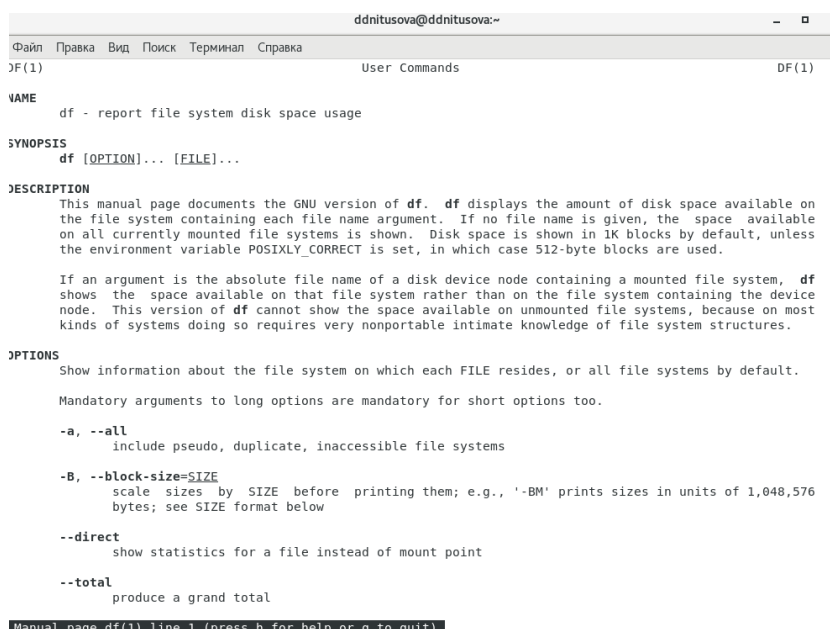
Figure 3.9: Рисунок 9

```
[ddnitusova@ddnitusova ~]$ man kill
[ddnitusova@ddnitusova ~]$ kill 6490
[1]+  Завершено          gedit
[ddnitusova@ddnitusova ~]$
```

Figure 3.10: Рисунок 10

С помощью команд «man df» и «man du» узнаю информацию по необходимым командам и далее использую их. **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

(рис. 3.11) (рис. 3.12) (рис. 3.13) (рис. 3.14)



```

ddnitusova@ddnitusova:~
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

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

    --total
        produce a grand total

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

```

Figure 3.11: Рисунок 11

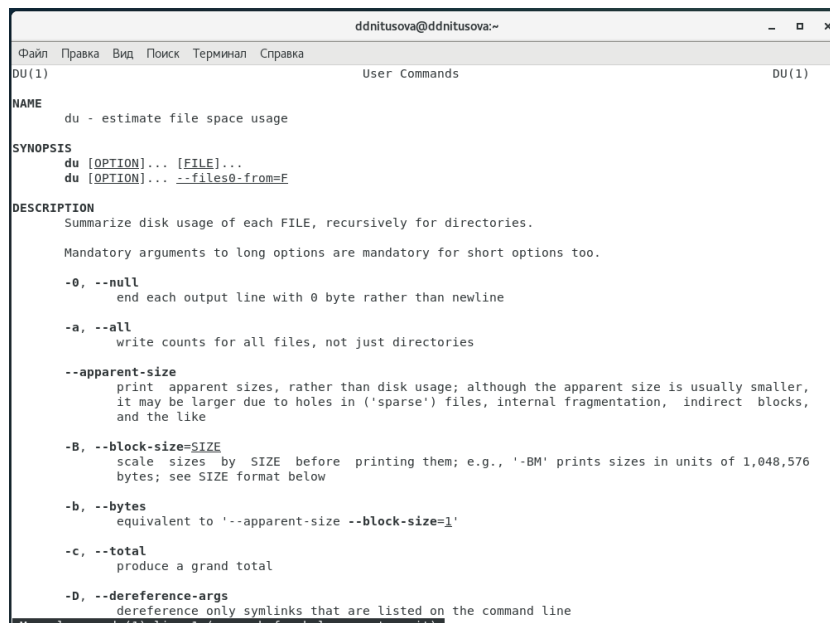


Figure 3.12: Рисунок 12

```

ddnitusova@ddnitusova ~]$ df
Файловая система      1K-блоков  Использовано  Доступно  Использовано%  Смонтировано в
devtmpfs                490560           0    490560           0% /dev
tmpfs                   507376           0    507376           0% /dev/shm
tmpfs                   507376       7788    499588           2% /run
tmpfs                   507376           0    507376           0% /sys/fs/cgroup
/dev/mapper/centos-root 17811456   5252416  12559040          30% /
/dev/sda1               1038336   175100    863236          17% /boot
tmpfs                   101476           48    101428           1% /run/user/1000
/dev/sr0                 59588       59588           0         100% /run/media/ddnitusova/VBox_GAs_6.1.20
/dev/sr1                 59588       59588           0         100% /run/media/ddnitusova/VBox_GAs_6.1.20
ddnitusova@ddnitusova ~]$
  
```

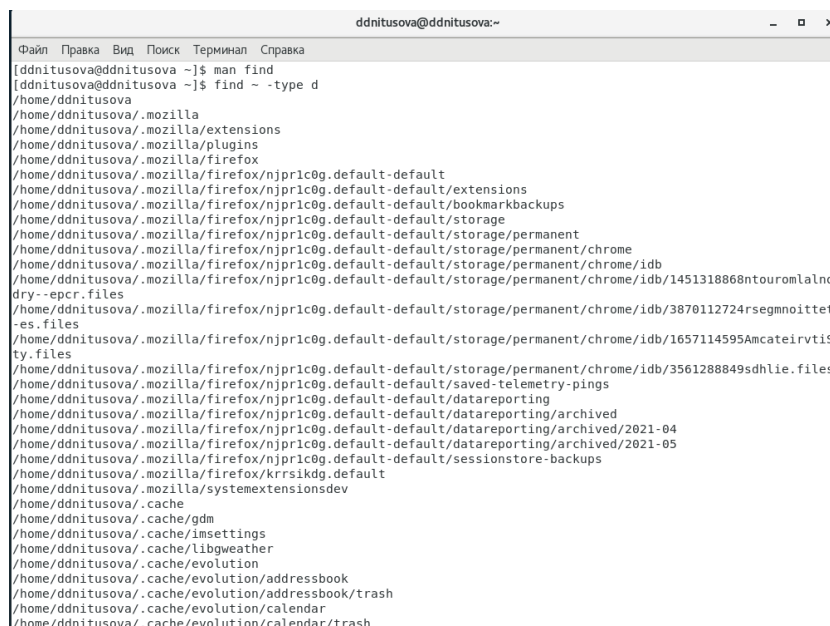
Figure 3.13: Рисунок 13

```

ddnitusova@ddnitusova ~]$ du
.          4096 ./mozilla/extensions
.          4096 ./mozilla/plugins
16964      ./mozilla/firefox/njprlc0g.default-default/extensions
1984       ./mozilla/firefox/njprlc0g.default-default/bookmarkbackups
1984       ./mozilla/firefox/njprlc0g.default-default/storage/permanent/chrome/idb/1451318868ntouromlalnodyr--ep
./r.files
.          4096 ./mozilla/firefox/njprlc0g.default-default/storage/permanent/chrome/idb/3870112724rsegmnoittet-es.fil
./s
.          4096 ./mozilla/firefox/njprlc0g.default-default/storage/permanent/chrome/idb/1657114595AmcateirvtiSty.file
.
.          4096 ./mozilla/firefox/njprlc0g.default-default/storage/permanent/chrome/idb/3561288849sdhlie.files
1980       ./mozilla/firefox/njprlc0g.default-default/storage/permanent/chrome/idb
1984       ./mozilla/firefox/njprlc0g.default-default/storage/permanent/chrome
1984       ./mozilla/firefox/njprlc0g.default-default/storage/permanent
1984       ./mozilla/firefox/njprlc0g.default-default/storage
16         ./mozilla/firefox/njprlc0g.default-default/saved-telemetry-pings
16         ./mozilla/firefox/njprlc0g.default-default/datareporting/archived/2021-04
16         ./mozilla/firefox/njprlc0g.default-default/datareporting/archived/2021-05
16         ./mozilla/firefox/njprlc0g.default-default/datareporting/archived
16         ./mozilla/firefox/njprlc0g.default-default/datareporting
16         ./mozilla/firefox/njprlc0g.default-default/sessionstore-backups
16368      ./mozilla/firefox/njprlc0g.default-default
16368      ./mozilla/firefox/krrsikdg.default
16380      ./mozilla/firefox
16380      ./mozilla/systemextensionsdev
16380      ./mozilla
.          4096 ./cache/gdm
.          4096 ./cache/imsettings
.          4096 ./cache/libgweather
.          4096 ./cache/evolution/addressbook/trash
.          4096 ./cache/evolution/addressbook
.          4096 ./cache/evolution/calendar/trash
.          4096 ./cache/evolution/calendar
.          4096 ./cache/evolution/last-trash
  
```

Figure 3.14: Рисунок 14

Вывожу имена всех директорий, имеющихся в моем домашнем каталоге с помощью команды «`find ~ -type d`». (рис. 3.15)



```
ddnitusova@ddnitusova:~  
[ddnitusova@ddnitusova ~]$ man find  
[ddnitusova@ddnitusova ~]$ find ~ -type d  
/home/ddnitusova  
/home/ddnitusova/.mozilla  
/home/ddnitusova/.mozilla/extensions  
/home/ddnitusova/.mozilla/plugins  
/home/ddnitusova/.mozilla/firefox  
/home/ddnitusova/.mozilla/firefox/njpric0g.default-default  
/home/ddnitusova/.mozilla/firefox/njpric0g.default-default/extensions  
/home/ddnitusova/.mozilla/firefox/njpric0g.default-default/bookmarkbackups  
/home/ddnitusova/.mozilla/firefox/njpric0g.default-default/storage  
/home/ddnitusova/.mozilla/firefox/njpric0g.default-default/storage/permanent  
/home/ddnitusova/.mozilla/firefox/njpric0g.default-default/storage/permanent/chrome  
/home/ddnitusova/.mozilla/firefox/njpric0g.default-default/storage/permanent/chrome/ldb  
/home/ddnitusova/.mozilla/firefox/njpric0g.default-default/storage/permanent/chrome/ldb/1451318868ntouromlalno  
dry-epcr.files  
/home/ddnitusova/.mozilla/firefox/njpric0g.default-default/storage/permanent/chrome/ldb/3870112724rsegmnoittet  
-es.files  
/home/ddnitusova/.mozilla/firefox/njpric0g.default-default/storage/permanent/chrome/ldb/1657114595AmcateirvtiS  
ty.files  
/home/ddnitusova/.mozilla/firefox/njpric0g.default-default/storage/permanent/chrome/ldb/3561288849sdhlie.files  
/home/ddnitusova/.mozilla/firefox/njpric0g.default-default/saved-telemetry-pings  
/home/ddnitusova/.mozilla/firefox/njpric0g.default-default/datareporting  
/home/ddnitusova/.mozilla/firefox/njpric0g.default-default/datareporting/archived  
/home/ddnitusova/.mozilla/firefox/njpric0g.default-default/datareporting/archived/2021-04  
/home/ddnitusova/.mozilla/firefox/njpric0g.default-default/datareporting/archived/2021-05  
/home/ddnitusova/.mozilla/firefox/njpric0g.default-default/sessionstore-backups  
/home/ddnitusova/.mozilla/firefox/krrsikd.default  
/home/ddnitusova/.mozilla/systemextensionsdev  
/home/ddnitusova/.cache  
/home/ddnitusova/.cache/gdm  
/home/ddnitusova/.cache/insettings  
/home/ddnitusova/.cache/libgweather  
/home/ddnitusova/.cache/evolution  
/home/ddnitusova/.cache/evolution/addressbook  
/home/ddnitusova/.cache/evolution/addressbook/trash  
/home/ddnitusova/.cache/evolution/calendar  
/home/ddnitusova/.cache/evolution/calendar/trash
```

Figure 3.15: Рисунок 15

4 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 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.

5 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.