

Отчёт по лабораторной работе №7
Дисциплина: Операционные системы

Студент:Джеффри Родригес Сантос

Группа: НПМбд-02-20

МОСКВА 2021 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. 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, начинавшиеся с символа `s`? Предложите несколько вариантов, как это сделать.
5. Выведите на экран (постранично) 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`, конвейерный фильтр `grep`. Можно ли определить этот идентификатор более простым способом?
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 директо-

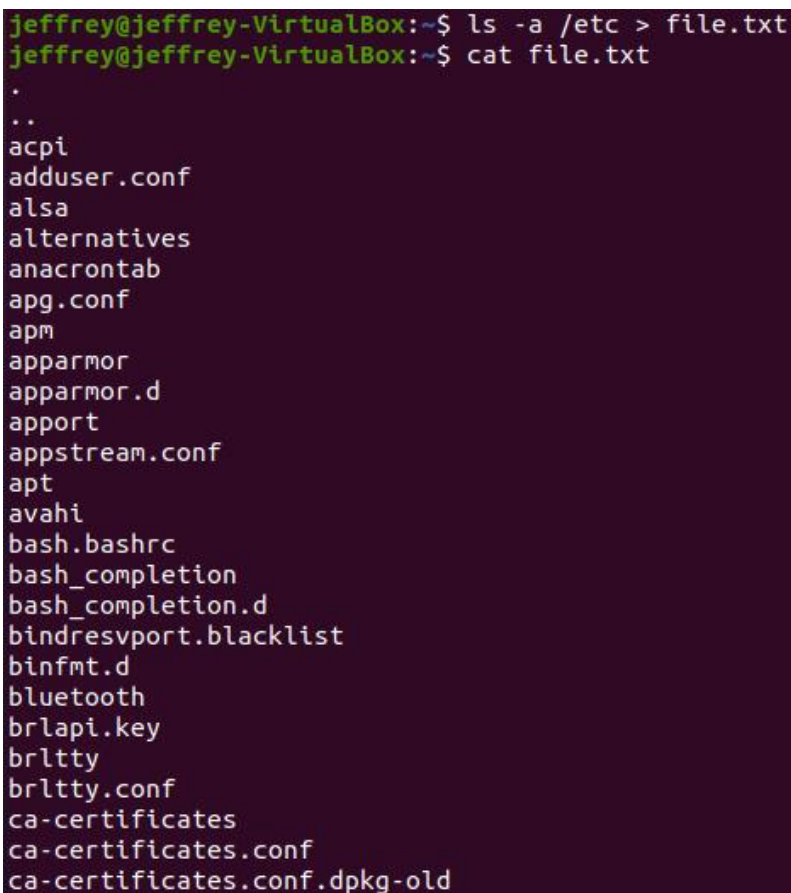
рий, имеющихся в вашем домашнем каталоге.

3. Выполнение лабораторной работы

1.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» (рис. -fig. 3.1).

2.Далее с помощью команды «ls -a ~» file.txt» дописываю в этот же файл названия файлов, содержащихся в моем домашнем каталоге (рис. -fig. 3.2). Командой«cat file.txt» просматриваю файл, чтобы убедиться в правильности действий (рис. -fig. 3.3).



```
jeffrey@jeffrey-VirtualBox:~$ ls -a /etc > file.txt
jeffrey@jeffrey-VirtualBox:~$ cat file.txt
.
..
acpi
adduser.conf
alsa
alternatives
anacrontab
apg.conf
apm
apparmor
apparmor.d
appport
appstream.conf
apt
avahi
bash.bashrc
bash_completion
bash_completion.d
bindresvport.blacklist
binfmt.d
bluetooth
brlapi.key
brltty
brltty.conf
ca-certificates
ca-certificates.conf
ca-certificates.conf.dpkg-old
```

Рис. 3.1: Записываем названия файлов, содержащихся в каталоге /etc

```

jeffrey@jeffrey-VirtualBox:~$ ls -a ~ >> file.txt
jeffrey@jeffrey-VirtualBox:~$ ls -a ~
.          Documents      .local      ski.plases
..         Downloads      may         snap
abc1       feathers        monthly     .ssh
australia  file.txt             .mozilla   .sudo_as_admin_successful
.bash_history .gitconfig      Music       Templates
.bash_logout .gnupg          my_os       Videos
.bashrc     helloworld      Pictures    .wget-hsts
.cache      initial-commit  play
.config     laboratory      .profile
Desktop     laboratory2     Public
jeffrey@jeffrey-VirtualBox:~$ cat file.txt
.
..
acpi
adduser.conf
alsa
alternatives
anacrontab
apg.conf
apm
apparmor
apparmor.d
appport
appstream.conf
apt
avahi

```

Рис. 3.2: Записываем названия файлов, содержащихся в домашнем каталоге.

```

.bashrc
.cache
.config
Desktop
Documents
Downloads
file.txt
.gitconfig
.gnupg
helloworld
initial-commit
laboratory
laboratory2
.local
may
monthly
.mozilla
Music
newdir
Pictures
.profile
Public
snap
.ssh
.sudo_as_admin_successful
Templates
Videos
.wget-hsts

```

Рис. 3.3: Просматриваем файл

4. 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. -fig. 3.4).

```
jeffrey@jeffrey-VirtualBox:~$ cat conf.txt
.
..
abc1
australia
.bash_history
.bash_logout
.bashrc
.cache
.config
conf.txt
Desktop
Documents
Downloads
feathers
file.txt
.gitconfig
.gnupg
helloworld
initial-commit
laboratory
laboratory2
.local
may
monthly
.mozilla
Music
my_os
Pictures
-
.wget-hsts
jeffrey@jeffrey-VirtualBox:~$ ls
abc1      Documents  helloworld  may      Pictures  snap
australia Downloads  initial-commit  monthly  play      Templates
conf.txt  feathers   laboratory   Music    Public    Videos
Desktop   file.txt   laboratory2  my_os    ski.places
jeffrey@jeffrey-VirtualBox:~$
```

```

jeffrey@jeffrey-VirtualBox:~$ man grep
jeffrey@jeffrey-VirtualBox:~$ grep -e '\.conf$' file.txt > conf.txt
jeffrey@jeffrey-VirtualBox:~$ cat conf.txt
adduser.conf
apg.conf
appstream.conf
brltty.conf
ca-certificates.conf
debconf.conf
deluser.conf
e2scrub.conf
fprind.conf
fuse.conf
gai.conf
hdparm.conf
host.conf
kernel-img.conf
kerneloops.conf
ld.so.conf
libao.conf
libaudit.conf
logrotate.conf
ltrace.conf
mke2fs.conf
mtools.conf
nsswitch.conf
pam.conf
pnm2ppa.conf
popularity-contest.conf

```

Рис. 3.4: Вывожу имена файлов, имеющих расширение .conf

13. 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 » (рис. -fig. 3.5).

```

jeffrey@jeffrey-VirtualBox:~$ find ~ -maxdepth 1 -name "с*" -print
/home/jeffrey/conf.txt
jeffrey@jeffrey-VirtualBox:~$ ls ~/с*
/home/jeffrey/conf.txt
jeffrey@jeffrey-VirtualBox:~$ ls | grep с*
conf.txt
jeffrey@jeffrey-VirtualBox:~$

```

Рис. 3.5: Определим, какие файлы начинаются с символа с

14. Чтобы вывести на экран (постранично) имена файлов из каталога /etc, начинающиеся с символа h, воспользуемся командой «find /etc -maxdepth 1 -name "h" | less» (рис. -fig. 3.6).

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

Рис. 3.6: Вывод на экран (постранично) файлы, начинающиеся с символа h

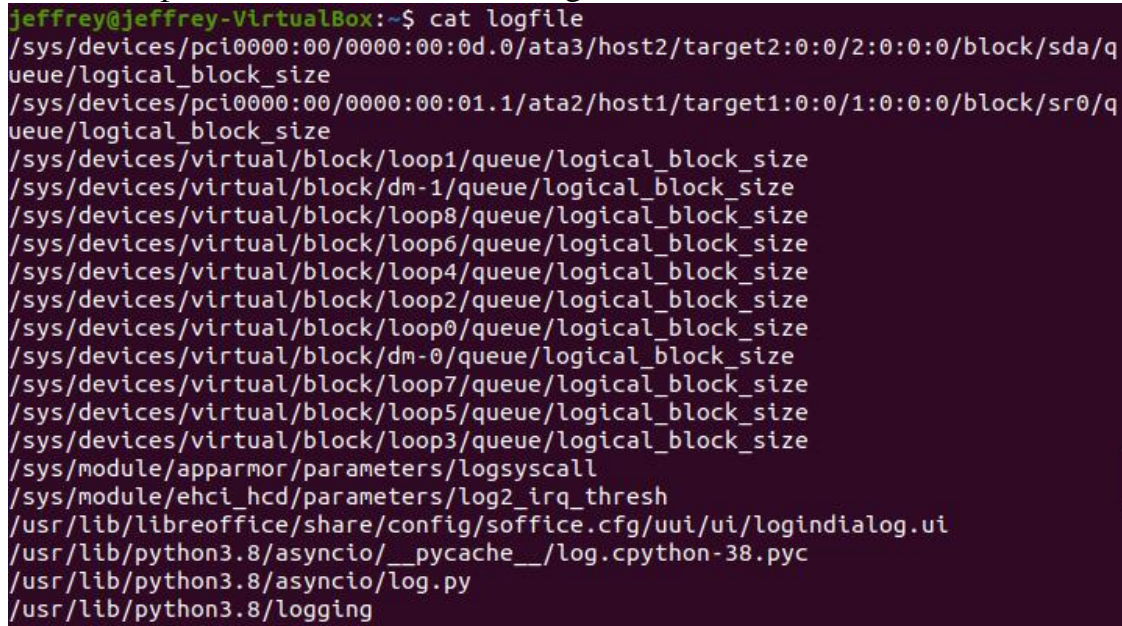
15. Запускаю в * фоновом режиме процесс, который будет записывать в файл ~/logfile файлы, имена которых начинаются с log, используя команду «find / -name "log" >log file &» (рис. -fig. 3.7). Командой «cat logfile» проверяю выполненные действия (рис. -fig. 3.8).
16. Удаляю файл ~/logfile командой «rm logfile».

```
jeffrey@jeffrey-VirtualBox:~$ find / -name "log*" > logfile &
[2] 5789
[1] Done find ~ -name "log" > ~/logfile
jeffrey@jeffrey-VirtualBox:~$ find: '/sys/kernel/tracing': Permission denied
find: '/sys/kernel/debug': Permission denied
find: '/sys/fs/pstore': Permission denied
find: '/sys/fs/bpf': Permission denied

jeffrey@jeffrey-VirtualBox:~$ find: '/boot/efi': Permission denied
find: '/tmp/systemd-private-4e02afe8aaab4e0d80b26fead1044294-ModemManager.service-oj3Cjg': Permission denied
find: '/tmp/systemd-private-4e02afe8aaab4e0d80b26fead1044294-systemd-logind.service-s0eDkf': Permission denied
find: '/tmp/systemd-private-4e02afe8aaab4e0d80b26fead1044294-geoclue.service-G4V72i': Permission denied
find: '/tmp/systemd-private-4e02afe8aaab4e0d80b26fead1044294-switcheroo-control.service-Je8ULg': Permission denied
find: '/tmp/systemd-private-4e02afe8aaab4e0d80b26fead1044294-upower.service-tjBwbl': Permission denied
find: '/tmp/systemd-private-4e02afe8aaab4e0d80b26fead1044294-systemd-resolved.service-cjl90i': Permission denied
find: '/tmp/systemd-private-4e02afe8aaab4e0d80b26fead1044294-systemd-timesyncd.service-wtv5pi': Permission denied
find: '/tmp/systemd-private-4e02afe8aaab4e0d80b26fead1044294-fwupd.service-Teggkh': Permission denied
```

Рис. 3.7: Запускаем в фоновом режиме процесс, который запишет

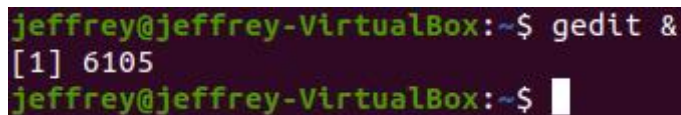
файлы, начинающиеся с log



```
jeffrey@jeffrey-VirtualBox:~$ cat logfile
/sys/devices/pci0000:00/0000:00:0d.0/ata3/host2/target2:0:0/2:0:0:0/block/sda/queue/logical_block_size
/sys/devices/pci0000:00/0000:00:01.1/ata2/host1/target1:0:0/1:0:0:0/block/sr0/queue/logical_block_size
/sys/devices/virtual/block/loop1/queue/logical_block_size
/sys/devices/virtual/block/dm-1/queue/logical_block_size
/sys/devices/virtual/block/loop8/queue/logical_block_size
/sys/devices/virtual/block/loop6/queue/logical_block_size
/sys/devices/virtual/block/loop4/queue/logical_block_size
/sys/devices/virtual/block/loop2/queue/logical_block_size
/sys/devices/virtual/block/loop0/queue/logical_block_size
/sys/devices/virtual/block/dm-0/queue/logical_block_size
/sys/devices/virtual/block/loop7/queue/logical_block_size
/sys/devices/virtual/block/loop5/queue/logical_block_size
/sys/devices/virtual/block/loop3/queue/logical_block_size
/sys/module/apparmor/parameters/logsyscall
/sys/module/ehci_hcd/parameters/log2_irq_thresh
/usr/lib/libreoffice/share/config/soffice.cfg/uui/ui/logindialog.ui
/usr/lib/python3.8/asyncio/__pycache__/log.cpython-38.pyc
/usr/lib/python3.8/asyncio/log.py
/usr/lib/python3.8/logging
```

Рис. 3.8: проверяем выполненные действия

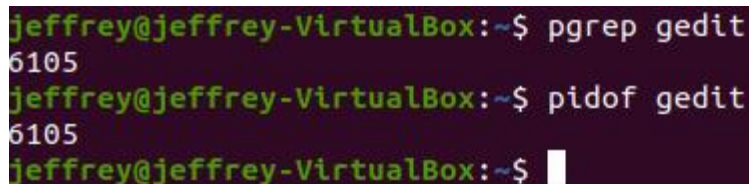
17.Запускаю редактор gedit в фоновом режиме командой «gedit &» (рис.-fig. [3.9](#)). После этого на экране появляется окно редактора.



```
jeffrey@jeffrey-VirtualBox:~$ gedit &
[1] 6105
jeffrey@jeffrey-VirtualBox:~$
```

Рис. 3.9: запускаю редактор gedit в фоновом режиме

18.Чтобы определить идентификатор процесса gedit, использую команду «ps | grep -i “gedit”». Наш процесс имеет PID 518. Узнать идентификатор про-цесса можно также, используя команду «pgrep gedit» или «pidof gedit» (рис.-fig. [3.10](#)).



```
jeffrey@jeffrey-VirtualBox:~$ pgrep gedit
6105
jeffrey@jeffrey-VirtualBox:~$ pidof gedit
6105
jeffrey@jeffrey-VirtualBox:~$
```

Рис. 3.10: определяем идентификатор процесса gedit

19.Прочитав информацию о команде kill с помощью команды «man kill», используя её для завершения процесса gedit (команда «kill 518») (рис.-fig. [3.11](#)) (рис.-fig. [3.12](#)).

```
jeffrey@jeffrey-VirtualBox:~$ man kill
jeffrey@jeffrey-VirtualBox:~$ kill 6105
jeffrey@jeffrey-VirtualBox:~$ man df
[1]+  Terminated                  gedit
jeffrey@jeffrey-VirtualBox:~$ man du
```

Рис. 3.11: используем kill для завершения процесса gedit.

```
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.

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

Рис. 3.12: Информация о команде kill

20.С помощью команд «man df» (рис. -fig. 3.13) и «man du» (рис. -fig. 3.14) узнаю информацию по необходимым командам и далее использую их (рис.-fig. 3.15).

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.

```
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 con-
    taining 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 non-
    portable 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
Manual page df(1) line 1 (press h for help or q to quit)
```

Рис. 3.13: Информация о команде df


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

Рис. 3.14: Информация о команде du

```
jeffrey@jeffrey-VirtualBox:~$ df
Filesystem            1K-blocks    Used Available Use% Mounted on
udev                  470552         0   470552  0% /dev
tmpfs                 100432      1440    98992  2% /run
/dev/mapper/vgubuntu-root 101915616 7410796  89284676  8% /
tmpfs                 502152         0    502152  0% /dev/shm
tmpfs                  5120          4     5116  1% /run/lock
tmpfs                 502152         0    502152  0% /sys/fs/cgroup
/dev/loop0             56832    56832         0 100% /snap/core18/1988
/dev/loop2             224256   224256         0 100% /snap/gnome-3-34-180
4/66
/dev/loop1             56832    56832         0 100% /snap/core18/1997
/dev/loop3             66432    66432         0 100% /snap/gtk-common-the
mes/1514
/dev/loop4             52352    52352         0 100% /snap/snap-store/518
/dev/loop5             66688    66688         0 100% /snap/gtk-common-the
mes/1515
/dev/loop6             32896    32896         0 100% /snap/snapd/11841
/dev/loop7             33152    33152         0 100% /snap/snapd/11588
/dev/sda1              523248         4   523244  1% /boot/efi
tmpfs                 100428        40   100388  1% /run/user/1000
jeffrey@jeffrey-VirtualBox:~$
```

Рис. 3.15: Используем df и du

21.Получаем информацию с помощью команды «man find» (рис. -fig. [3.16](#))и выводим имена всех директорий, имеющихся в домашнем каталоге с помощью команды «find ~ -type d» (рис. -fig. [3.17](#)).

```
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...] [ex-
    pression]

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 Considera-
    tions' 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.
    Manual page find(1) line 1 (press h for help or q to quit)
```

Рис. 3.16: Информация о команде find

```
jeffrey@jeffrey-VirtualBox:~$ man find
jeffrey@jeffrey-VirtualBox:~$ find ~ -type d
/home/jeffrey
/home/jeffrey/laboratory2
/home/jeffrey/laboratory2/.git
/home/jeffrey/laboratory2/.git/objects
/home/jeffrey/laboratory2/.git/objects/62
/home/jeffrey/laboratory2/.git/objects/ed
/home/jeffrey/laboratory2/.git/objects/b6
/home/jeffrey/laboratory2/.git/objects/34
/home/jeffrey/laboratory2/.git/objects/41
/home/jeffrey/laboratory2/.git/objects/da
/home/jeffrey/laboratory2/.git/objects/a1
/home/jeffrey/laboratory2/.git/objects/7e
/home/jeffrey/laboratory2/.git/objects/a8
/home/jeffrey/laboratory2/.git/objects/fa
/home/jeffrey/laboratory2/.git/objects/50
/home/jeffrey/laboratory2/.git/objects/fb
/home/jeffrey/laboratory2/.git/objects/81
/home/jeffrey/laboratory2/.git/objects/3c
/home/jeffrey/laboratory2/.git/objects/b5
/home/jeffrey/laboratory2/.git/objects/3e
/home/jeffrey/laboratory2/.git/objects/15
/home/jeffrey/laboratory2/.git/objects/e6
/home/jeffrey/laboratory2/.git/objects/1a
/home/jeffrey/laboratory2/.git/objects/de
/home/jeffrey/laboratory2/.git/objects/95
/home/jeffrey/laboratory2/.git/objects/b4
```

Рис. 3.17: Вывод имен всех директорий, имеющихя в домашнем каталоге

22. Контрольные вопросы

23. 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`.

24. “>” Перенаправление вывода в файл

“>>” Перенаправление вывода в файл и открытие файла в режиме добавления (данные добавляются в конец файла).

25. 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)

26. 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.

27. `pid`: идентификатор процесса (PID) процесса (process ID), к которому вызывают метод

`gid`: идентификатор группы UNIX, в котором работает программа.

28. 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.

29. `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.

30. `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 "*.x"` поиск скрытых файлов в домашнем каталоге

31. Файл по его содержимому можно найти с помощью команды `grep`: «`grep -r "слово/выражение, которое нужно найти"`».
32. Утилита `df`, позволяет проанализировать свободное пространство на всех подключенных к системе разделах.
33. При выполнении команды `du` (без указания папки и опции) можно получить все файлы и папки текущей директории с их размерами. Для домашнего каталога: `du ~/`
34. 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 синтернетом;
 - `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 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`.

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.

35. Выводы

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