

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №6

Дисциплина: Операционные системы

Обрезкова Анастасия Владимировна

Содержание

1	Цель работы	5
2	Задание	6
3	Теоретическое введение	7
3.1	Перенаправление ввода-вывода	7
3.2	Конвейер	7
3.3	Поиск файла	8
4	Выполнение лабораторной работы	9
5	Выводы	23
	Список литературы	24

Список иллюстраций

4.1	Запись файлов	9
4.2	Вывод файлов	10
4.3	Нахождение файлов по символу	10
4.4	Команда для вывода файлов	10
4.5	Найденные файлы	11
4.6	Нахождение файлов по символам	11
4.7	Удаление	12
4.8	Редактор	12
4.9	определение идентификатора процессора	12
4.10	Справка команды kill	13
4.11	Завершение процесса	13
4.12	Опции команды df	14
4.13	Опции команды du	14
4.14	Команды df	15
4.15	Команда du	15
4.16	Опции команды find	16
4.17	Выполнение команды	16

Список таблиц

1 Цель работы

Ознакомиться 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. Ознакомиться с инструментами поиска файлов и фильтрации текстовых данных.
2. Приобрести практических навыков: по управлению процессами

3 Теоретическое введение

3.1 Перенаправление ввода-вывода

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`. 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 `>`, `»`, `<`, `«`.

3.2 Конвейер

Конвейер (`pipe`) служит для объединения простых команд или утилит в цепочки, в которых результат работы предыдущей команды передаётся последующей.

Конвейеры можно группировать в цепочки и выводить с помощью перенаправления в файл.

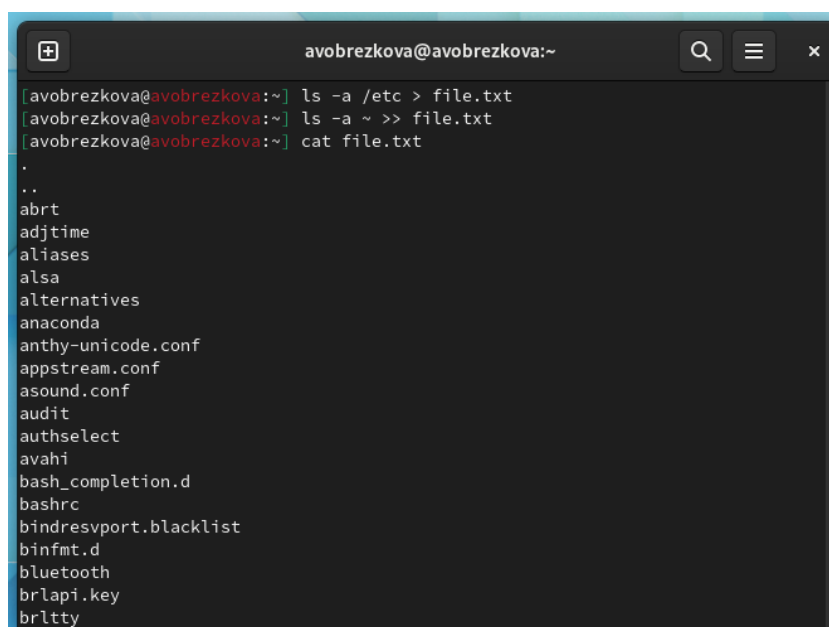
Чаще всего скрипты на Bash используются в качестве автоматизации каких-то рутинных операций в консоли, отсюда иногда возникает необходимость в обработке stdout одной команды и передача на stdin другой команде, при этом результат выполнения команды должен обработан.

3.3 Поиск файла

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

4 Выполнение лабораторной работы

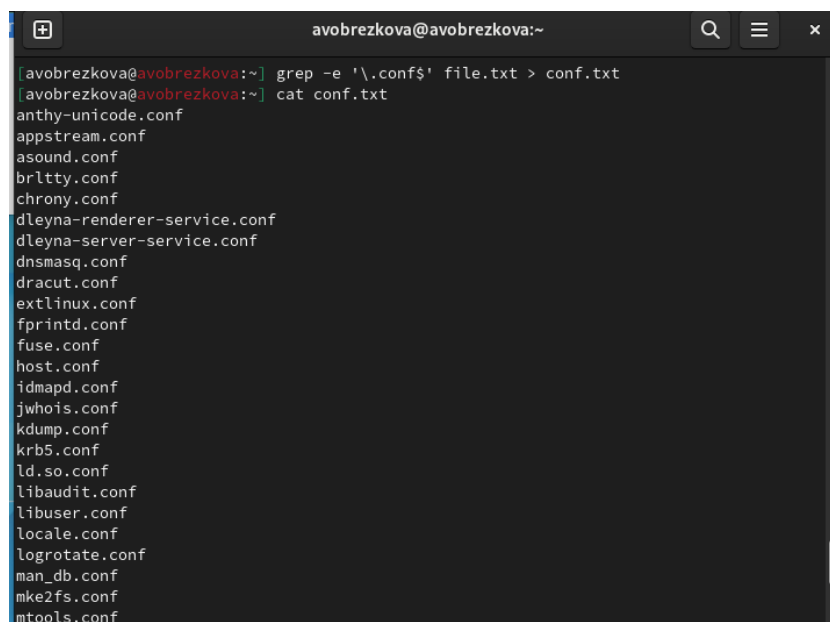
1. Осуществила вход в систему, используя свое имя пользователя.
2. Записала в файл file.txt названия файлов, содержащихся в каталоге /etc. Дописала в этот же файл названия файлов, содержащихся в моем домашнем каталоге. (рис. [4.1])



```
avobrezkova@avobrezkova:~  
[avobrezkova@avobrezkova:~] ls -a /etc > file.txt  
[avobrezkova@avobrezkova:~] ls -a ~ >> file.txt  
[avobrezkova@avobrezkova:~] 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
```

Рис. 4.1: Запись файлов

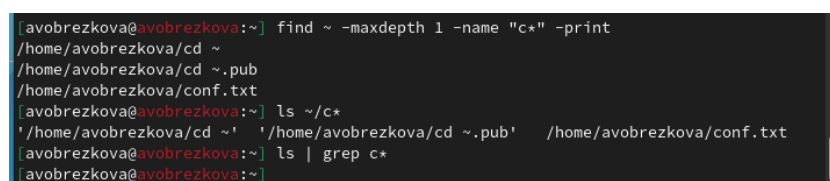
3. Вывела имена всех файлов из file.txt, имеющих расширение .conf, после чего записала их в новый текстовый файл conf.txt. (рис. [4.2])



```
avobrezkova@avobrezkova:~  
[avobrezkova@avobrezkova:~] grep -e '\.conf$' file.txt > conf.txt  
[avobrezkova@avobrezkova:~] cat conf.txt  
anthy-unicode.conf  
appstream.conf  
asound.conf  
brltty.conf  
chrony.conf  
dleyna-renderer-service.conf  
dleyna-server-service.conf  
dnsmasq.conf  
dracut.conf  
extlinux.conf  
fprintd.conf  
fuse.conf  
host.conf  
idmapd.conf  
jwhois.conf  
kdump.conf  
krb5.conf  
ld.so.conf  
libaudit.conf  
libuser.conf  
locale.conf  
logrotate.conf  
man_db.conf  
mke2fs.conf  
mtools.conf
```

Рис. 4.2: Вывод файлов

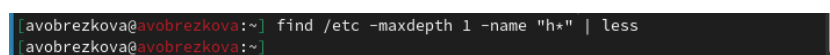
4. Определила, какие файлы в моем домашнем каталоге имеют имена, начинавшиеся с символа с. Найти эти файлы можно с помощью команд, указанных на рисунке. (рис. [4.3])



```
[avobrezkova@avobrezkova:~] find ~ -maxdepth 1 -name "c*" -print  
/home/avobrezkova/cd ~  
/home/avobrezkova/cd ~/.pub  
/home/avobrezkova/conf.txt  
[avobrezkova@avobrezkova:~] ls ~/c*  
'/home/avobrezkova/cd ~' '/home/avobrezkova/cd ~/.pub' /home/avobrezkova/conf.txt  
[avobrezkova@avobrezkova:~] ls | grep c*  
[avobrezkova@avobrezkova:~]
```

Рис. 4.3: Нахождение файлов по символу

5. Вывела на экран (по странично) имена файлов из каталога /etc, начинающиеся с символа h. (рис. [4.4], рис. [4.5])



```
[avobrezkova@avobrezkova:~] find /etc -maxdepth 1 -name "h*" | less  
[avobrezkova@avobrezkova:~]
```

Рис. 4.4: Команда для вывода файлов

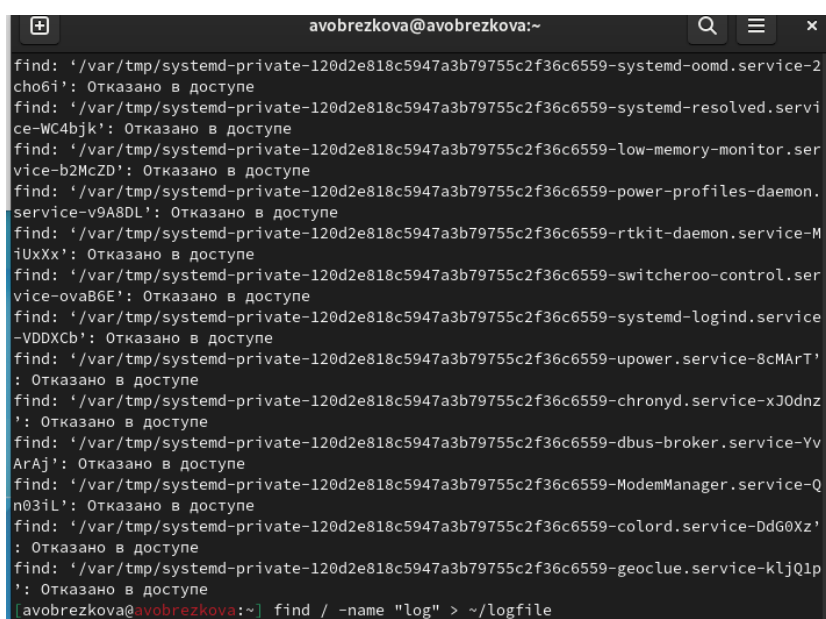
```

/etc/hp
/etc/httpd
/etc/host.conf
/etc/hosts
/etc/hostname
(END)

```

Рис. 4.5: Найденные файлы

6. Запустила в фоновом режиме процесс, который будет записывать в файл ~/logfile файлы, имена которых начинаются с log с помощью команды `find / -name "log" > ~/logfile`. (рис. [4.6])



```

avobrezkova@avobrezkova:~
find: '/var/tmp/systemd-private-120d2e818c5947a3b79755c2f36c6559-systemd-oomd.service-2cho6i': Отказано в доступе
find: '/var/tmp/systemd-private-120d2e818c5947a3b79755c2f36c6559-systemd-resolved.service-WC4bjk': Отказано в доступе
find: '/var/tmp/systemd-private-120d2e818c5947a3b79755c2f36c6559-low-memory-monitor.service-b2McZD': Отказано в доступе
find: '/var/tmp/systemd-private-120d2e818c5947a3b79755c2f36c6559-power-profiles-daemon.service-v9A8DL': Отказано в доступе
find: '/var/tmp/systemd-private-120d2e818c5947a3b79755c2f36c6559-rtkit-daemon.service-MiUxXx': Отказано в доступе
find: '/var/tmp/systemd-private-120d2e818c5947a3b79755c2f36c6559-switcheroo-control.service-ovaB6E': Отказано в доступе
find: '/var/tmp/systemd-private-120d2e818c5947a3b79755c2f36c6559-systemd-logind.service-VDDXCb': Отказано в доступе
find: '/var/tmp/systemd-private-120d2e818c5947a3b79755c2f36c6559-upower.service-8cMarT': Отказано в доступе
find: '/var/tmp/systemd-private-120d2e818c5947a3b79755c2f36c6559-chronyd.service-xJ0dnz': Отказано в доступе
find: '/var/tmp/systemd-private-120d2e818c5947a3b79755c2f36c6559-dbus-broker.service-YvArAj': Отказано в доступе
find: '/var/tmp/systemd-private-120d2e818c5947a3b79755c2f36c6559-ModemManager.service-Qn03iL': Отказано в доступе
find: '/var/tmp/systemd-private-120d2e818c5947a3b79755c2f36c6559-colord.service-DdG0Xz': Отказано в доступе
find: '/var/tmp/systemd-private-120d2e818c5947a3b79755c2f36c6559-geoclue.service-kljQ1p': Отказано в доступе
[avobrezkova@avobrezkova:~] find / -name "log" > ~/logfile

```

Рис. 4.6: Нахождение файлов по символам

7. Удалила файл ~/logfile. (рис. [4.7])

```
[avobrezkova@avobrezkova:~] cat ~/logfile
/dev/log
/home/avobrezkova/.config/user-share/log
/home/avobrezkova/.miktex/texmfs/data/miktex/log
/run/log
/run/initramfs/log
/sys/module/devres/parameters/log
/sys/module/vboxguest/parameters/log
/opt/VBoxGuestAdditions-6.1.38/src/vboxguest-6.1.38/vboxguest/common/log
/usr/lib/golang/src/log
/usr/lib/golang/pkg/linux_amd64/log
/usr/share/doc/zfs-fuse/contrib/solaris/onnv-gate/log
/var/log
[avobrezkova@avobrezkova:~] rm logfile
[avobrezkova@avobrezkova:~] rm logfile
rm: невозможно удалить 'logfile': Нет такого файла или каталога
[avobrezkova@avobrezkova:~]
```

Рис. 4.7: Удаление

8. Запустила из консоли в фоновом режиме редактор gedit. после ввода команды gedit & появляется окно редактора. (рис. [4.8])

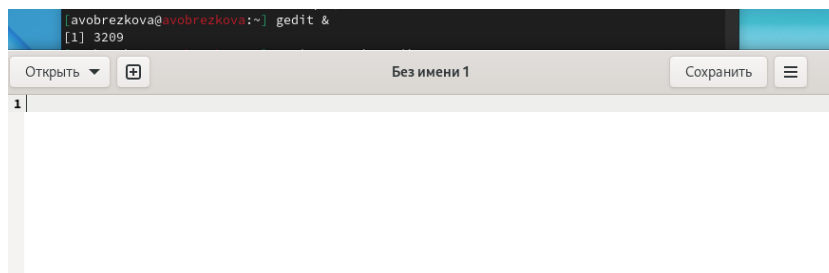


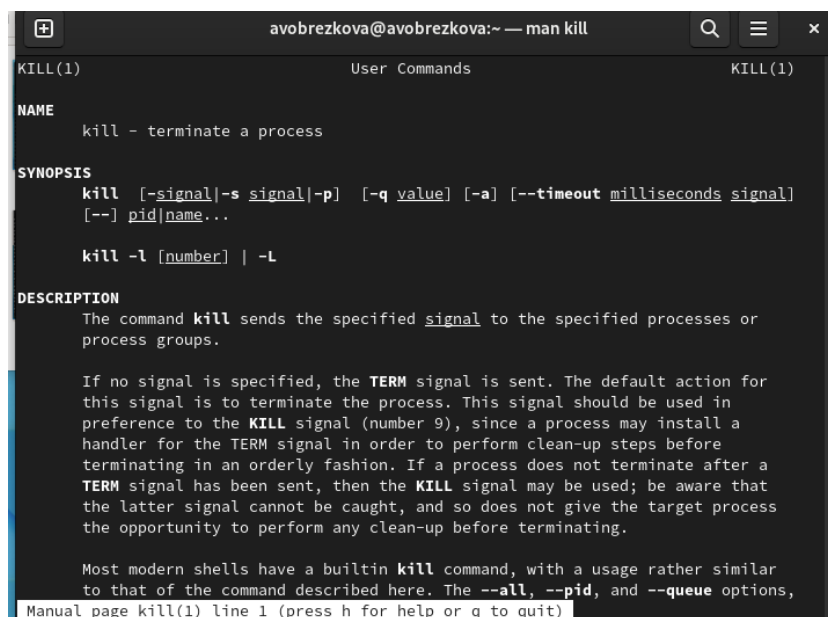
Рис. 4.8: Редактор

9. Определила идентификатор процесса gedit, используя команду ps, конвейер и фильтр grep, с помощью команды ps | grep -i "gedit". (рис. [4.9])

```
[avobrezkova@avobrezkova:~] gedit &
[1] 3209
[avobrezkova@avobrezkova:~] ps | grep -i "gedit"
 3209 pts/0    00:00:00 gedit
[avobrezkova@avobrezkova:~] pgrep gedit
3209
[avobrezkova@avobrezkova:~] pidof gedit
3209
[avobrezkova@avobrezkova:~]
```

Рис. 4.9: определение идентификатора процессора

10. Прочитала справку (man) команды kill, после чего использовала её для завершения процесса gedit. (рис. [4.10]; рис. [4.11])

A screenshot of a terminal window showing the man page for the 'kill' command. The window title is 'avobrezkova@avobrezkova:~ — man kill'. The content is divided into sections: NAME, SYNOPSIS, and DESCRIPTION. The NAME section says 'kill - terminate a process'. The SYNOPSIS section shows the command syntax: 'kill [-signal|-s signal|-p] [-q value] [-a] [--timeout milliseconds signal] [-- pid|name...'. The DESCRIPTION section explains that the command sends a signal to specified processes or process groups, with the default being the TERM signal (number 9). It also mentions that modern shells have a builtin kill command.

```
KILL(1) User Commands KILL(1)

NAME
    kill - terminate a process

SYNOPSIS
    kill [-signal|-s signal|-p] [-q value] [-a] [--timeout milliseconds signal]
    [-- pid|name...

    kill -l [number] | -L

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 clean-up 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 clean-up 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 --queue options,
    Manual page kill(1) line 1 (press h for help or q to quit)
```

Рис. 4.10: Справка команды kill

A screenshot of a terminal window showing the execution of the 'kill' command. The user enters 'man kill' and then 'kill 3209'. The terminal shows the command being executed and the process being terminated.

```
[avobrezkova@avobrezkova:~] man kill
[avobrezkova@avobrezkova:~] kill 3209
[1]+  Завершено      gedit
[avobrezkova@avobrezkova:~]
```

Рис. 4.11: Завершение процесса

11. Выполнила команды `df` и `du`, предварительно получив более подробную информацию об этих командах, с помощью команды `man`. (рис. [4.12]; рис. [4.13]; рис. [4.14]; рис. [4.15])
- `df` - утилита, показывающая список всех файловых систем по именам устройств, сообщает их размер, занятое и свободное пространство и точки монтирования.
 - `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.

```
avobrezkova@avobrezkova:~ — man df
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 currently 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 version of df can-
  not show the space available on unmounted file systems, because on most kinds
  of systems doing so requires very nonportable intimate knowledge of file sys-
  tem structures.

OPTIONS
  Show information about the file system on which each FILE resides, or all
  file systems by default.

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

Рис. 4.12: Опции команды df

```
avobrezkova@avobrezkova:~ — man du
DU(1)                                User Commands                                DU(1)

NAME
  du - estimate file space usage

SYNOPSIS
  du [OPTION]... [FILE]...
  du [OPTION]... --files0-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.

  -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 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
    print sizes in SIZE blocks

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

Рис. 4.13: Опции команды du

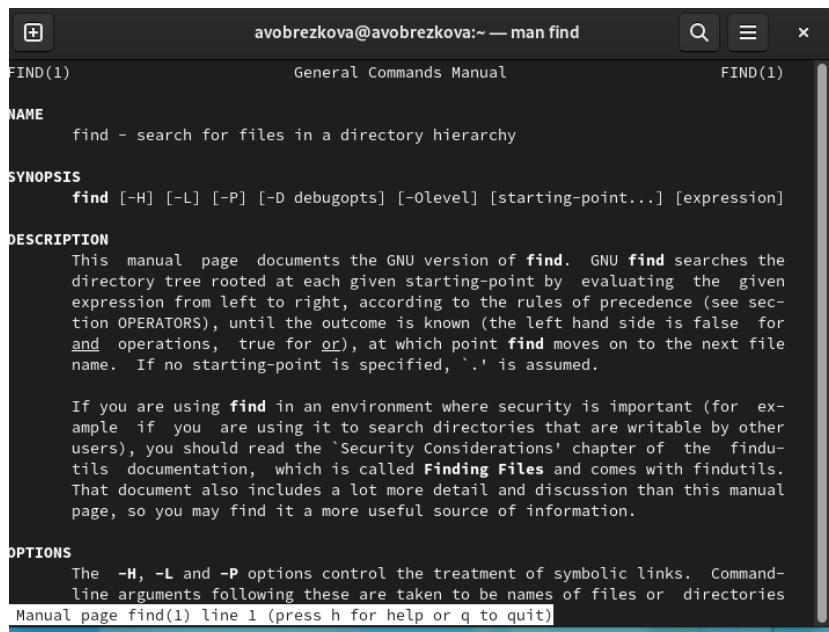
```
[avobrezkova@avobrezkova:~] df
Файловая система 1К-блоков  Использовано  Доступно  Использовано%  Смонтировано в
devtmpfs           4096             0      4096             0% /dev
tmpfs              1637988          0    1637988             0% /dev/shm
tmpfs              655196           1384    653812             1% /run
/dev/sda2          41783296       25193136    15774768          62% /
tmpfs              1637988          24    1637964             1% /tmp
/dev/sda1          996780         230068    697900            25% /boot
/dev/sda2          41783296       25193136    15774768          62% /home
tmpfs              327596          508    327088             1% /run/user/1000
/dev/sr0           62390           62390          0          100% /run/media/avobrezkova/V
Box_GAs_6.1.38
[avobrezkova@avobrezkova:~]
```

Рис. 4.14: Команды df

```
28 ./website/public/tags
4 ./website/public/tag/academic/page/1
4 ./website/public/tag/academic/page
32 ./website/public/tag/academic
4 ./website/public/tag/deep-learning/page/1
4 ./website/public/tag/deep-learning/page
28 ./website/public/tag/deep-learning
4 ./website/public/tag/demo/page/1
4 ./website/public/tag/demo/page
24 ./website/public/tag/demo
4 ./website/public/tag/source-themes/page/1
4 ./website/public/tag/source-themes/page
28 ./website/public/tag/source-themes
4 ./website/public/tag/开源/page/1
4 ./website/public/tag/开源/page
32 ./website/public/tag/开源
144 ./website/public/tag
8900 ./website/public
30692 ./website
8 ./ski.plases/equipment
0 ./ski.plases/plans
8 ./ski.plases
du: невозможно получить доступ к './play/games': Отказано в доступе
0 ./play
1792488 .
[avobrezkova@avobrezkova:~]
```

Рис. 4.15: Команда du

12. Воспользовавшись справкой команды find, вывела имена всех директорий, имеющих в моем домашнем каталоге. (рис. [4.16]; рис. [4.17])



```
avobrezkova@avobrezkova:~ — man find
FIND(1)                                General Commands Manual                                FIND(1)

NAME
    find - search for files in a directory hierarchy

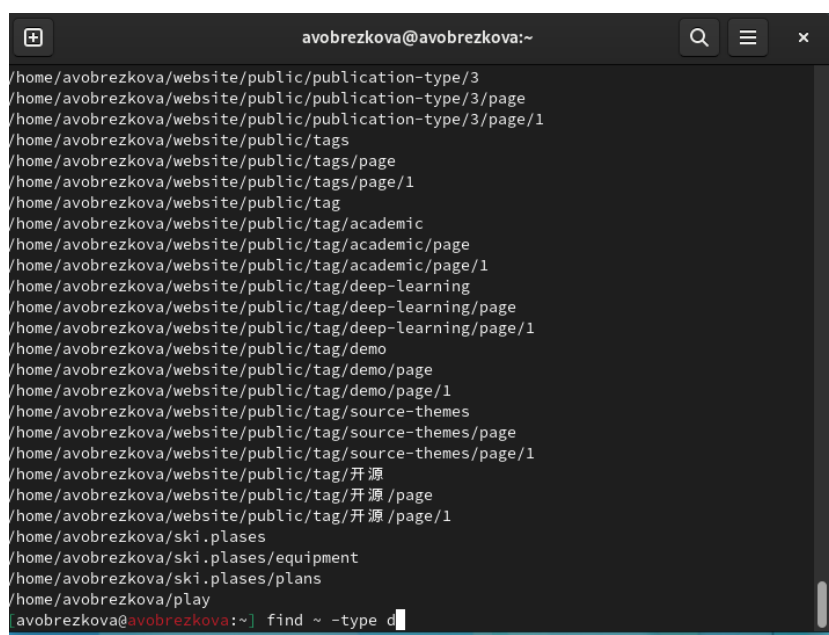
SYNOPSIS
    find [-H] [-L] [-P] [-D debugopts] [-Olevel] [starting-point...] [expression]

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 sec-
    tion 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 ex-
    ample if you are using it to search directories that are writable by other
    users), you should read the 'Security Considerations' chapter of the findu-
    tils 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
    Manual page find(1) line 1 (press h for help or q to quit)
```

Рис. 4.16: Опции команды find



```
avobrezkova@avobrezkova:~
/home/avobrezkova/website/public/publication-type/3
/home/avobrezkova/website/public/publication-type/3/page
/home/avobrezkova/website/public/publication-type/3/page/1
/home/avobrezkova/website/public/tags
/home/avobrezkova/website/public/tags/page
/home/avobrezkova/website/public/tags/page/1
/home/avobrezkova/website/public/tag
/home/avobrezkova/website/public/tag/academic
/home/avobrezkova/website/public/tag/academic/page
/home/avobrezkova/website/public/tag/academic/page/1
/home/avobrezkova/website/public/tag/deep-learning
/home/avobrezkova/website/public/tag/deep-learning/page
/home/avobrezkova/website/public/tag/deep-learning/page/1
/home/avobrezkova/website/public/tag/demo
/home/avobrezkova/website/public/tag/demo/page
/home/avobrezkova/website/public/tag/demo/page/1
/home/avobrezkova/website/public/tag/source-themes
/home/avobrezkova/website/public/tag/source-themes/page
/home/avobrezkova/website/public/tag/source-themes/page/1
/home/avobrezkova/website/public/tag/开源
/home/avobrezkova/website/public/tag/开源/page
/home/avobrezkova/website/public/tag/开源/page/1
/home/avobrezkova/ski.plases
/home/avobrezkova/ski.plases/equipment
/home/avobrezkova/ski.plases/plans
/home/avobrezkova/play
avobrezkova@avobrezkova:~$ find ~ -type d
```

Рис. 4.17: Выполнение команды

Данные изменения можно проверить по ссылке: https://github.com/avobrezkova/study_2022-2023_os-intro/tree/master/labs/lab06/report

#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.

Большинство используемых в консоли команд и программ записывают результаты своей работы в стандартный поток вывода `stdout`.

2. Объясните разницу между операцией `>` и `>>`.

`'>'` Перенаправление вывода в файл

`'>>'` Перенаправление вывода в файл и открытие файла в режиме добавления (данные добавляются в конец файла)/

3. Что такое конвейер?

Конвейер (`pipe`) служит для объединения простых команд или утилит в цепочки, в которых результат работы предыдущей команды передаётся последующей.

Синтаксис следующий:

`команда1|команда2` (это означает, что вывод команды 1 передаётся на ввод команде 2)

4. Что такое процесс? Чем это понятие отличается от программы?

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. Что такое PID и GID?

- `pid`: идентификатор процесса (PID) процесса (`processID`), к которому вызывают метод
- `gid`: идентификатор группы UNIX, в котором работает программа.

6. Что такое задачи и какая команда позволяет ими управлять?

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. Найдите информацию об утилитах `top` и `htop`. Каковы их функции?

- `top` – это консольная программа, которая показывает список работающих процессов в системе. Программа в реальном времени отсортирует запущенные процессы по их нагрузке на процессор.
- `htop` – это продвинутый консольный мониторинг процессов. Утилита выводит постоянно меняющийся список системных процессов, который сортируется в зависимости от нагрузки на ЦПУ. Если делать сравнение `stop`, то

htop показывает абсолютно все процессы в системе, время их непрерывного использования, загрузку процессоров и расход оперативной памяти.

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

find – это команда для поиска файлов и каталогов на основе специальных условий. Ее можно использовать в различных обстоятельствах, например, для поиска файлов по разрешениям, владельцам, группам, типу, размеру и другим подобным критериям.

Команда find имеет такой синтаксис:

find[папка][параметры] критерий шаблон [действие]

Папка – каталог в котором будем искать

Параметры – дополнительные параметры, например, глубина поиска, и т д.

Критерий – по какому критерию будем искать: имя, дата создания, права, владелец и т д.

Шаблон – непосредственно значение по которому будем отбирать файлы.

Основные параметры:

- -P никогда не открывать символические ссылки
- -L - получает информацию о файлах по символическим ссылкам. Важно для дальнейшей обработки, чтобы обрабатывалась не ссылка, а сам файл.
- -maxdepth - максимальная глубина поиска по подкаталогам, для поиска только в текущем каталоге установите 1.
- -depth - искать сначала в текущем каталоге, а потом в подкаталогах
- -mount искать файлы только в этой файловой системе.
- -version - показать версию утилиты find
- -print - выводить полные имена файлов

- -typef - искать только файлы
- -typed - поиск папки в Linux

Основные критерии:

- -name - поиск файлов по имени
- -perm - поиск файлов в Linux по режиму доступа
- -user - поиск файлов по владельцу
- -group - поиск по группе
- -mtime - поиск по времени модификации файла
- -atime - поиск файлов по дате последнего чтения
- -nogroup - поиск файлов, не принадлежащих ни одной группе
- -nouser - поиск файлов без владельцев
- -newer - найти файлы новее чем указанный
- -size - поиск файлов в Linux по их размеру

Примеры:

`find~ -type d` поиск директорий в домашнем каталоге

`find~ -type f -name ".*"` поиск скрытых файлов в домашнем каталоге

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

Файл по его содержимому можно найти с помощью команды `grep`:

«`grep -r`”слово/выражение, которое нужно найти”».

10. Как определить объем свободной памяти на жёстком диске?

Утилита `df`, позволяет проанализировать свободное пространство на всех подключенных к системе разделах.

11. Как определить объем вашего домашнего каталога?

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. Как удалить зависший процесс?

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 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 (вканале) 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 ознакомилась с инструментами поиска файлов и фильтрации текстовых данных, а также приобрела практические навыки по управлению процессами, по проверке использования диска и обслуживанию файловых систем.

Список литературы

1. https://esystem.rudn.ru/pluginfile.php/1976037/mod_resource/content/4/006-lab_proc.pdf