

# **Лабораторная работа 6**

**Поиск файлов. Перенаправление ввода-вывода. Просмотр запущенных процессов**

Гисматуллин Артём Вадимович НПИбд-01-22

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>5</b>
<b>2</b>	<b>Задание</b>	<b>6</b>
<b>3</b>	<b>Теоретическое введение</b>	<b>7</b>
3.1	Перенаправление ввода-вывода . . . . .	7
3.2	Конвейер . . . . .	7
3.3	Фильтрация текста . . . . .	8
3.4	Проверка использования диска . . . . .	8
3.5	Управление задачами . . . . .	9
3.6	Управление процессами . . . . .	9
3.7	Получение информации о процессах . . . . .	9
<b>4</b>	<b>Выполнение лабораторной работы</b>	<b>11</b>
<b>5</b>	<b>Контрольные вопросы</b>	<b>24</b>
<b>6</b>	<b>Выводы</b>	<b>28</b>
	<b>Список литературы</b>	<b>29</b>

## Список иллюстраций

4.1	Командная строка. Заполнение файла file.txt . . . . .	11
4.2	Командная строка. Заполнение файла conf.txt . . . . .	12
4.3	Командная строка. Вывод всех файлов на с* . . . . .	13
4.4	Командная строка. Вывод всех файлов на h* . . . . .	14
4.5	Командная строка. Фоновая запись в logfile . . . . .	15
4.6	Командная строка. Удаление logfile . . . . .	16
4.7	Командная строка. Справка о kill . . . . .	17
4.8	Командная строка. Прекращение процесса . . . . .	17
4.9	Командная строка. man df . . . . .	18
4.10	Командная строка. man du . . . . .	19
4.11	Командная строка. Выполнение команды df . . . . .	20
4.12	Командная строка. Выполнение команды du . . . . .	21
4.13	Командная строка. Справочная информация find . . . . .	22
4.14	Командная строка. Результат поиска директорий . . . . .	23

## Список таблиц

# 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, имеющихся в вашем домашнем каталоге.

## 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, например:

```
ls -la |sort > sortilg_list
```

вывод команды `ls -la` передаётся команде сортировки `sort\verb`, которая пишет результат в файл `sorting_list\verb`.

Чаще всего скрипты на Bash используются в качестве автоматизации каких-то рутин-ных 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.

Команда `find` используется для поиска и отображения на экран имён файлов, соответствующих заданной строке символов. Формат команды: `find путь [-опции]` Путь определяет каталог, начиная с которого по всем подкаталогам будет вестись поиск.

### 3.3 Фильтрация текста

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

`grep строка имя_файла`

Кроме того, команда `grep` способна обрабатывать стандартный вывод других команд (любой текст). Для этого следует использовать конвейер, связав вывод команды с вводом `grep`.

### 3.4 Проверка использования диска

Команда `df` показывает размер каждого смонтированного раздела диска. Формат команды:

`df [-опции] [файловая_система]`

Команда `du` показывает число килобайт, используемое каждым файлом или каталогом. Формат команды:

`du [-опции] [имя_файла...]`



## 3.5 Управление задачами

Любую выполняющуюся в консоли команду или внешнюю программу можно запустить в фоновом режиме. Для этого следует в конце имени команды указать знак амперсанда &. Например:

```
gedit &
```

Будет запущен текстовый редактор gedit в фоновом режиме. Консоль при этом не будет заблокирована. Запущенные фоном программы называются задачами (jobs). Ими можно управлять с помощью команды jobs, которая выводит список запущенных в данный момент задач. Для завершения задачи необходимо выполнить команду kill %номер задачи

## 3.6 Управление процессами

Любой команде, выполняемой в системе, присваивается идентификатор процесса (process ID). Получить информацию о процессе и управлять им, пользуясь идентификатором процесса, можно из любого окна командного интерпретатора.

## 3.7 Получение информации о процессах

Команда ps используется для получения информации о процессах. Формат команды:

```
ps [-опции]
```

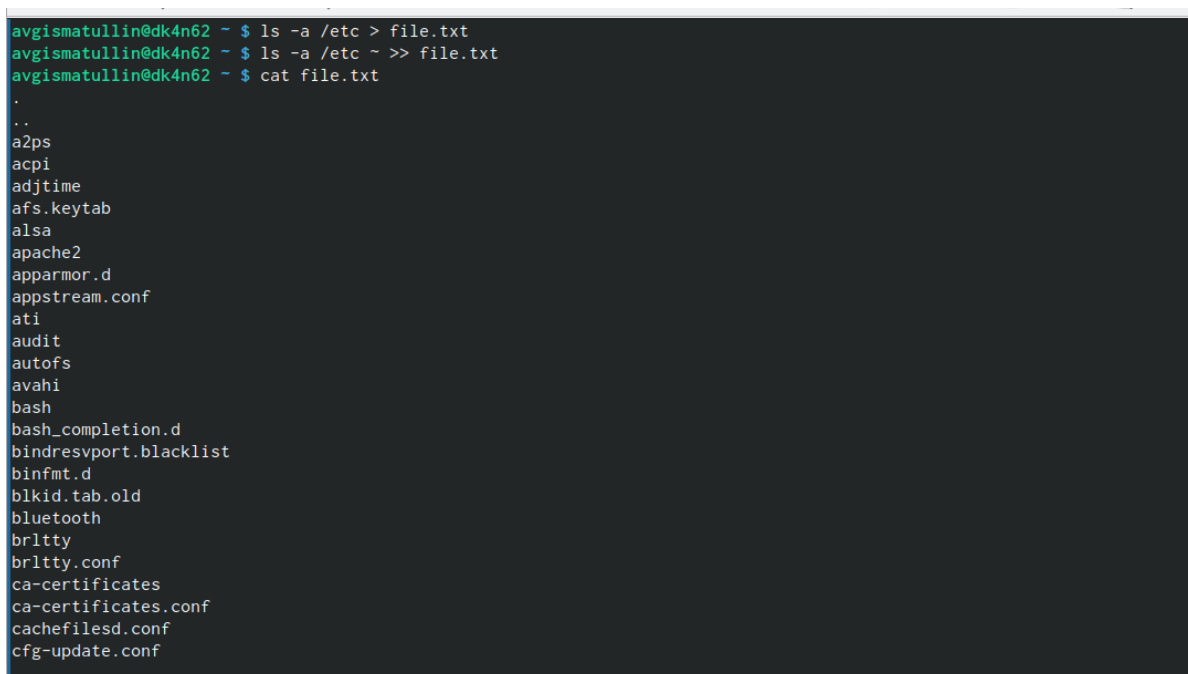
Для получения информации о процессах, управляемых вами и запущенных (работающих или остановленных) на вашем терминале, используйте опцию aux. Пример:

```
ps aux
```

Для запуска команды в фоновом режиме необходимо в конце командной строки указать знак & (амперсанд).

## 4 Выполнение лабораторной работы

1. Первым делом запишем в file.txt все названия файлов каталога /etc и домашнего каталога. Это делается с помощью стандартного потока ввода > (рис. 4.1).



```
avgismatullin@dk4n62 ~ $ ls -a /etc > file.txt
avgismatullin@dk4n62 ~ $ ls -a /etc ~ >> file.txt
avgismatullin@dk4n62 ~ $ cat file.txt
.
..
a2ps
acpi
adjtime
afs.keytab
alsa
apache2
apparmor.d
appstream.conf
ati
audit
autofs
avahi
bash
bash_completion.d
bindresvport.blacklist
binfmt.d
blkid.tab.old
bluetooth
brltty
brltty.conf
ca-certificates
ca-certificates.conf
cachefilesd.conf
cfg-update.conf
```

Рис. 4.1: Командная строка. Заполнение файла file.txt

Далее отберем с помощью фильтра `grep` все файлы с форматом `.conf` из `file.txt` и добавим их в `conf.txt` (рис. 4.2).

```
avgismatullin@dk4n62 ~ $ grep -e '\.conf$' file.txt > conf.txt
avgismatullin@dk4n62 ~ $ cat conf.txt
appstream.conf
brltty.conf
ca-certificates.conf
cachefilesd.conf
cfg-update.conf
dhcpcd.conf
dispatch-conf.conf
dleyna-server-service.conf
dnsmasq.conf
e2fsck.conf
e2scrub.conf
etc-update.conf
fluidsynth.conf
fuse.conf
gai.conf
genkernel.conf
gssapi_mech.conf
host.conf
idmapd.conf
idn2.conf
idnalias.conf
krb5.conf
ldap.conf
```

Рис. 4.2: Командная строка. Заполнение файла conf.txt

- Следующим шагом будет изучение возможностей поиска по названиям. Это мы можем сделать двумя способами: фильтрация команды `ls` или с помощью команды `find` с определенными опциями. Два способа реализованы следующим образом (рис. 4.3):

```
avgismatullin@dk4n62 ~ $ ls -a ~ | grep c*
conf.txt
avgismatullin@dk4n62 ~ $ find ~ -maxdepth 1 -name "c*" -print
/afs/.dk.sci.pfu.edu.ru/home/a/v/avgismatullin/conf.txt
avgismatullin@dk4n62 ~ $
```

Рис. 4.3: Командная строка. Вывод всех файлов на с\*

Отбор всех файлов на h\* из каталога /etc с помощью find (рис. 4.4):

```
avgismatullin@dk4n62 ~ $ find /etc -maxdepth 1 -name "h*" -print
/etc/hotplug.d
/etc/highlight
/etc/host.conf
/etc/hosts
/etc/hostname
/etc/hal
/etc/htdig
/etc/hsqldb
/etc/hotplug
/etc/httpd
/etc/hosts.allow
/etc/harbour.cfg
/etc/harbour
avgismatullin@dk4n62 ~ $
```

Рис. 4.4: Командная строка. Вывод всех файлов на h\*

3. Далее мы запускаем фоновые процессы. Первым делом запишем все файлы log. в ~/logfile. Для этого применим стандартный поток ввода, команду find для отбора файлов с определенным названием и апперсанд для запуска процесса в фоновом режиме (рис. 4.5):

```
avgismatullin@dk4n62 ~ $ find / -name "log*" -print > logfile &
[2] 18556
avgismatullin@dk4n62 ~ $ find: '/lost+found': Отказано в доступе
find: '/root': Отказано в доступе
find: '/run/udisks2': Отказано в доступе
find: '/run/lightdm': Отказано в доступе
find: '/run/cups/certs': Отказано в доступе
find: '/run/user/4864/systemd/inaccessible/dir': Отказано в доступе
find: '/run/sudo': Отказано в доступе
find: '/run/multipath': Отказано в доступе
find: '/run/lvm': Отказано в доступе
find: '/run/lock/lvm': Отказано в доступе
find: '/run/cryptsetup': Отказано в доступе
find: '/run/chrony': Отказано в доступе
find: '/run/brltty': Отказано в доступе
find: '/run/apache2': Отказано в доступе
find: '/run/rpcbind': Отказано в доступе
find: '/run/log/journal/bce4714beaaa098e9fbe221a00000052': Отказано в доступе
find: '/run/credentials/systemd-tmpfiles-setup.service': Отказано в доступе
find: '/run/credentials/systemd-tmpfiles-setup-dev.service': Отказано в доступе
find: '/run/credentials/systemd-sysctl.service': Отказано в доступе
find: '/run/systemd/propagate': Отказано в доступе
find: '/run/systemd/unit-root': Отказано в доступе
find: '/run/systemd/inaccessible/dir': Отказано в доступе
find: '/run/initramfs': Отказано в доступе
```

Рис. 4.5: Командная строка. Фоновая запись в logfile

Затем удалим logfile (рис. 4.6):

```
avgismatullin@dk4n62 ~ $ rm logfile
avgismatullin@dk4n62 ~ $ ls
bin      GNUstep      public_html  work      Загрузки    Общедоступные
conf.txt pandoc-2.18  PycharmProjects  Видео     Изображения  'Рабочий стол'
file.txt public       tmp          Документы Музыка      Шаблоны
avgismatullin@dk4n62 ~ $
```

Рис. 4.6: Командная строка. Удаление logfile

4. Изучив команду `kill` (рис. 4.7), запустим `gedit` в фоновом режиме и с помощью этой команды прекратим процесс (рис. 4.8):



```
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. 4.7: Командная строка. Справка о kill

```
avgismatullin@dk4n62 ~ $ gedit &
[3] 18947
avgismatullin@dk4n62 ~ $ ps
  PID TTY          TIME CMD
 15145 pts/0    00:00:00 bash
 18283 pts/0    00:00:17 find
 18556 pts/0    00:00:13 find
 18947 pts/0    00:00:01 gedit
 19137 pts/0    00:00:00 ps
avgismatullin@dk4n62 ~ $ ps | grep -i "gedit"
 18947 pts/0    00:00:01 gedit
avgismatullin@dk4n62 ~ $ man kill
avgismatullin@dk4n62 ~ $ kill 18947
avgismatullin@dk4n62 ~ $ ps
  PID TTY          TIME CMD
 15145 pts/0    00:00:00 bash
 18283 pts/0    00:00:22 find
 18556 pts/0    00:00:19 find
 19499 pts/0    00:00:00 ps
[3]+  Завершено      gedit
avgismatullin@dk4n62 ~ $
```

Рис. 4.8: Командная строка. Прекращение процесса

Можно заметить, что идентификатор процесса мы определили посредством команды `ps`.

5. Выполним команды `df` и `du` (рис. [4.11]), (рис. [4.12]) предварительно получив более подробную информацию об этих командах, с помощью команды `man`.

Команда `df` показывает размер каждого смонтированного раздела диска (рис. [4.9]).

```
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 cannot show the space avail-
    able 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 sys-
    tems by default.

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

Рис. 4.9: Командная строка. `man df`

Команда `du` показывает число килобайт, используемое каждым файлом или каталогом (рис. [4.10]).

```
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

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

Рис. 4.10: Командная строка. man du

```
avgismatullin@dk4n62 ~ $ df
Файловая система      1К-блоков  Использовано  Доступно  Использовано%
Смонтировано в
none                   3999704      16808      3982896      1%
/run
udev                   10240        0        10240        0%
/dev
tmpfs                  3999704        0      3999704        0%
/dev/shm
/dev/sda8              484939832    79022576    381210200     18%
/
tmpfs                  3999708      628860      3370848     16%
/tmp
/dev/sda6              50090536     12648     47500992      1%
/var/cache/openafs
AFS                    2147483647        0    2147483647      0%
/afs
tmpfs                  799940       200       799740        1%
/run/user/4864
mark.sci.pfu.edu.ru:/usr/portage  18350080    6449408    10426112     39%
/usr/portage
mark.sci.pfu.edu.ru:/usr/local/share/portage  18350080    6449408    10426112     39%
/usr/local/share/portage
avgismatullin@dk4n62 ~ $
```

Рис. 4.11: Командная строка. Выполнение команды df

```
avgismatullin@dk4n62 ~ $ du
2    ./public/public_html
4    ./public
5    ../local/share/keyrings
4    ../local/share/gnome-shell
2    ../local/share/evolution/addressbook/trash
2    ../local/share/evolution/addressbook/system/photos
88   ../local/share/evolution/addressbook/system
92   ../local/share/evolution/addressbook
2    ../local/share/evolution/calendar/trash
3    ../local/share/evolution/calendar/system
7    ../local/share/evolution/calendar
2    ../local/share/evolution/mail/trash
2    ../local/share/evolution/mail/local/tmp
2    ../local/share/evolution/mail/local/cur
2    ../local/share/evolution/mail/local/new
2    ../local/share/evolution/mail/local/.Drafts/tmp
2    ../local/share/evolution/mail/local/.Drafts/cur
2    ../local/share/evolution/mail/local/.Drafts/new
8    ../local/share/evolution/mail/local/.Drafts
2    ../local/share/evolution/mail/local/.Outbox/tmp
2    ../local/share/evolution/mail/local/.Outbox/cur
2    ../local/share/evolution/mail/local/.Outbox/new
8    ../local/share/evolution/mail/local/.Outbox
2    ../local/share/evolution/mail/local/.Sent/tmp
2    ../local/share/evolution/mail/local/.Sent/cur
```

Рис. 4.12: Командная строка. Выполнение команды du

6. Также изучив справочную информацию о find (рис. [??]), найдем следующее: как отобразить название директорий? Для этого есть опция -type с ключом d. Результат получается следующий (рис. [4.13]).

```
FIND(1)                                General Commands Manual                                FIND(1)

NAME
    find - search for files in a directory hierarchy

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

DESCRIPTION
    This manual page documents the GNU version of find.  GNU find searches the directory
    tree rooted at each given starting-point by evaluating the given expression from
    left to right, according to the rules of precedence (see section OPERATORS), until
    the outcome is known (the left hand side is false for and operations, true for or),
    at which point find moves on to the next file name.  If no starting-point is speci-
    fied, find assumes the current directory.

    If you are using find in an environment where security is important (for example if
    you are using it to search directories that are writable by other users), you should
    read the 'Security Considerations' chapter of the findutils documentation, which is
    called Finding Files and comes with findutils.  That document also includes a lot
    more detail and discussion than this manual page, so you may find it a more useful
    source of information.

OPTIONS
    The -H, -L and -P options control the treatment of symbolic links.  Command-line ar-
    Manual page find(1) line 1 (press h for help or q to quit)
```

Рис. 4.13: Командная строка. Справочная информация find

```
avgismatullin@dk4n62 ~ $ find ~ -maxdepth 1 -type d -print
/afs/.dk.sci.pfu.edu.ru/home/a/v/avgismatullin
/afs/.dk.sci.pfu.edu.ru/home/a/v/avgismatullin/public
/afs/.dk.sci.pfu.edu.ru/home/a/v/avgismatullin/.local
/afs/.dk.sci.pfu.edu.ru/home/a/v/avgismatullin/Рабочий стол
/afs/.dk.sci.pfu.edu.ru/home/a/v/avgismatullin/Загрузки
/afs/.dk.sci.pfu.edu.ru/home/a/v/avgismatullin/Шаблоны
/afs/.dk.sci.pfu.edu.ru/home/a/v/avgismatullin/Общедоступные
/afs/.dk.sci.pfu.edu.ru/home/a/v/avgismatullin/Документы
/afs/.dk.sci.pfu.edu.ru/home/a/v/avgismatullin/Музыка
/afs/.dk.sci.pfu.edu.ru/home/a/v/avgismatullin/Изображения
/afs/.dk.sci.pfu.edu.ru/home/a/v/avgismatullin/Видео
/afs/.dk.sci.pfu.edu.ru/home/a/v/avgismatullin/.config
/afs/.dk.sci.pfu.edu.ru/home/a/v/avgismatullin/.gnupg
/afs/.dk.sci.pfu.edu.ru/home/a/v/avgismatullin/.mozilla
/afs/.dk.sci.pfu.edu.ru/home/a/v/avgismatullin/.pki
/afs/.dk.sci.pfu.edu.ru/home/a/v/avgismatullin/.java
/afs/.dk.sci.pfu.edu.ru/home/a/v/avgismatullin/PycharmProjects
/afs/.dk.sci.pfu.edu.ru/home/a/v/avgismatullin/GNUstep
/afs/.dk.sci.pfu.edu.ru/home/a/v/avgismatullin/.ssh
/afs/.dk.sci.pfu.edu.ru/home/a/v/avgismatullin/work
/afs/.dk.sci.pfu.edu.ru/home/a/v/avgismatullin/.texlive2022
/afs/.dk.sci.pfu.edu.ru/home/a/v/avgismatullin/pandoc-2.18
/afs/.dk.sci.pfu.edu.ru/home/a/v/avgismatullin/tmp
/afs/.dk.sci.pfu.edu.ru/home/a/v/avgismatullin/bin
/afs/.dk.sci.pfu.edu.ru/home/a/v/avgismatullin/.gphoto
```

Рис. 4.14: Командная строка. Результат поиска директорий

## 5 Контрольные вопросы

1. Какие потоки ввода вывода вы знаете?

Ввод и вывод распределяется между тремя стандартными потоками: - `stdin` — стандартный ввод (клавиатура), - `stdout` — стандартный вывод (экран), - `stderr` — стандартная ошибка (вывод ошибок на экран).

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

Основное отличие: `>` : Перезаписывает существующий файл или создает файл, если файл с указанным именем отсутствует в каталоге. `»` : добавляет существующий файл или создает файл, если файл с указанным именем отсутствует в каталоге.

3. Что такое конвейер?

Конвейер (англ. `pipeline`) в терминологии операционных систем семейства Unix — некоторое множество процессов, для которых выполнено следующее перенаправление ввода-вывода: то, что выводит на поток стандартного вывода предыдущий процесс, попадает в поток стандартного ввода следующего процесса.

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

Процесс - это: - программа на стадии выполнения - “объект”, которому выделено процессорное время - асинхронная работа



## 5. Что такое PID и GID?

Идентификатор процесса (PID). Каждому новому процессу ядро присваивает уникальный идентификационный номер. В любой момент времени идентификатор процесса является уникальным, хотя после завершения процесса он может использоваться снова для другого процесса. Некоторые идентификаторы зарезервированы системой для особых процессов. Так, процесс с идентификатором 1 - это процесс инициализации `init`, являющийся предком всех других процессов в системе.

Идентификатор группы GID и эффективный идентификатор группы (EGID) GID - это идентификационный номер группы данного процесса. EGID связан с GID также, как EUID с UID.

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

Принудительное завершение процесса и изменение его приоритета) можно выполнить и без команды `top`. Процессы в Linux имеют возможность обмениваться так называемыми “сигналами” с ядром и другими процессами. При получении сигнала процессом, управление передается подпрограмме его обработки или ядру, если такой подпрограммы не существует. В Linux имеется команда `kill`, которая позволяет послать заданному процессу любой сигнал.

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

`top` - интерактивный просмотрщик процессов. `htop` аналог `top`. Программа `top` динамически выводит в режиме реального времени информации о работающей системе, т.е. о фактической активности процессов. По умолчанию она выдает задачи, наиболее загружающие процессор сервера, и обновляет список каждые две секунды.

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

`find` : Для поиска файлов из командной строки вы можете использовать команду “`find`”. У этой команды следующий синтаксис:

`find path criteria action` - “`path`” - Секция для указания директории поиска. Если ничего не указано поиск идет по текущей директории. - “`criteria`” - Опции поиска. - “`action`” - Опции, которые влияют на состояние поиска или контролируют его, например, - “`-print`”

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

Для поиска файла по содержимому проще всего воспользоваться командой `grep` (вместо `find`).

Пример: `grep -r строка_поиска каталог`

10. Как определить объем свободной памяти на жёстком диске?

Самый простой способ найти свободное место на диске в Linux - это используйте команду `df`. Команда `df` означает «свободное от диска» и, очевидно, показывает вам свободное и доступное дисковое пространство в системах Linux. Работы С Нами -`h` вариант, он показывает дисковое пространство в удобочитаемом формате (МБ и ГБ).

11. Как определить объем вашего домашнего каталога?

В операционных системах на базе Linux посмотреть размер папки (директории) можно с помощью команды `du`. Эта команда, выполняемая в консоли, позволяет оценить используемый объем места на жестком диске отдельно по папкам и файлам, просуммировать результат, узнать общий размер папки.

12. Как удалить зависший процесс?

Убиваем процессы в Linux — команды `ps`, `kill` и `killall`

- Находим PID зависшего процесса Каждый процесс в Linux имеет свой идентификатор, называемый PID.

- «Убиваем» процесс командой kill. Когда известен PID процесса, мы можем убить его командой kill.
- Убиваем процессы командой killall.
- Заключение

## 6 Выводы

В ходе выполнения лабораторной работы мы ознакомились с инструментами поиска файлов и фильтрации текстовых данных. Приобрели практические навыки: по управлению процессами (и заданиями), по проверке использования диска и обслуживанию файловых систем.

## **Список литературы**