

# **Отчёт по лабораторной работе №8**

**Дисциплина: Операционные системы**

Верниковская Екатерина Андреевна

# Содержание

1	Цель работы	5
2	Задание	6
3	Выполнение лабораторной работы	7
4	Ответы на контрольные вопросы	17
5	Выводы	20
6	Список литературы	21

## Список иллюстраций

3.1	/etc в file.txt . . . . .	7
3.2	Создался file.txt . . . . .	7
3.3	Содержимое файла file.txt . . . . .	7
3.4	~/ в file.txt . . . . .	8
3.5	.conf в file.txt . . . . .	8
3.6	.conf в conf.txt . . . . .	8
3.7	Создался conf.txt . . . . .	8
3.8	Содержимое файла conf.txt . . . . .	9
3.9	Файлы в ~/ начинающиеся на с . . . . .	9
3.10	Другой вариант команды (1) . . . . .	9
3.11	Другой вариант команды (2) . . . . .	9
3.12	Другой вариант команды (3) . . . . .	10
3.13	Файлы в /etc начинающиеся на h . . . . .	10
3.14	Работа с logfile . . . . .	10
3.15	Создался logfile . . . . .	10
3.16	Содержимое файла logfile . . . . .	11
3.17	Удаление logfile . . . . .	11
3.18	Запуск gedit . . . . .	11
3.19	gedit в фоновом режиме . . . . .	12
3.20	Идентификатор процесса gedit . . . . .	12
3.21	Справка по команде kill . . . . .	13
3.22	Завершение процесса gedit . . . . .	13
3.23	Использование команды man для df и du . . . . .	13
3.24	Справка по команде df . . . . .	14
3.25	Справка по команде du . . . . .	14
3.26	Выполнение команды df . . . . .	14
3.27	Выполнение команды du . . . . .	15
3.28	Использование команды man для find . . . . .	15
3.29	Справка по команде find . . . . .	15
3.30	Использование команды find . . . . .	15
3.31	Имена всех директорий в ~/ . . . . .	16

## Список таблиц

# 1 Цель работы

Ознакомиться с инструментами поиска файлов и фильтрации текстовых данных. Приобрести практические навыки: по управлению процессами (и заданиями), по проверке использования диска и обслуживанию файловых систем.

## 2 Задание

1. Записать в файл `file.txt` названия файлов, содержащихся в каталоге `/etc`. Дописать в этот же файл названия файлов, содержащихся в нашем домашнем каталоге.
2. Вывести имена всех файлов из `file.txt`, имеющих расширение `.conf`, после чего записать их в новый текстовый файл `conf.txt`.
3. Определить, какие файлы в нашем домашнем каталоге имеют имена, начинавшиеся с символа `s`. Предложить несколько вариантов, как это сделать.
4. Вывести на экран (по странично) имена файлов из каталога `/etc`, начинающиеся с символа `h`.
5. Запустить в фоновом режиме процесс, который будет записывать в файл `~/logfile` файлы, имена которых начинаются с `log`.
6. Удалить файл `~/logfile`.
7. Запустить из консоли в фоновом режиме редактор `gedit`.
8. Определить идентификатор процесса `gedit`, используя команду `ps`, конвейер и фильтр `grep`.
9. Прочитать справку (`man`) команды `kill`, после чего использовать её для завершения процесса `gedit`.
10. Выполнить команды `df` и `du`, предварительно получив более подробную информацию об этих командах, с помощью команды `man`.
11. Воспользовавшись справкой команды `find`, высести имена всех директорий, имеющихся в нашем домашнем каталоге.

### 3 Выполнение лабораторной работы

Записываем в файл `file.txt` названия файлов, содержащихся в каталоге `/etc` с помощью команды `ls -lR /etc » file.txt` (рис. 3.1), (рис. 3.2), (рис. 3.3)

```
[eavernikovskaya@eavernikovskaya ~]$ sudo ls -lR /etc >> file.txt
[sudo] пароль для eavernikovskaya:
[eavernikovskaya@eavernikovskaya ~]$
```

Рис. 3.1: `/etc` в `file.txt`

```
[eavernikovskaya@eavernikovskaya ~]$ ls -l
итого 204
-rw-r--r--. 1 eavernikovskaya eavernikovskaya      0 map 18 17:56 abc1
drwxr--r--. 1 eavernikovskaya eavernikovskaya      0 map 18 18:02 australia
-rw-rw-r--. 1 eavernikovskaya eavernikovskaya      0 map 18 18:40 feathers
-rw-r--r--. 1 eavernikovskaya eavernikovskaya 186459 map 26 13:12 file.txt
```

Рис. 3.2: Создался `file.txt`

```
[eavernikovskaya@eavernikovskaya ~]$ head file.txt
/etc:
итого 1360
drwxr-xr-x. 1 root root    126 ноя  1 04:07 abrt
-rw-r--r--. 1 root root     16 фев 20 19:03 adjtime
-rw-r--r--. 1 root root   1529 июл 25  2023 aliases
drwxr-xr-x. 1 root root     70 янв 29 03:00 alsa
drwxr-xr-x. 1 root root   1386 фев 20 20:36 alternatives
drwxr-xr-x. 1 root root     56 ноя  1 04:07 anaconda
-rw-r--r--. 1 root root    541 июл 19  2023 anacrontab
-rw-r--r--. 1 root root    833 фев 10  2023 appstream.conf
[eavernikovskaya@eavernikovskaya ~]$
```

Рис. 3.3: Содержимое файла `file.txt`

Далее дописываем в файл `file.txt` названия файлов, содержащихся в нашем домашнем каталоге с помощью команды `ls -lR ~/ » file.txt` (рис. 3.4)

```
[eavernikovskaya@eavernikovskaya ~]$ sudo ls -lR ~/ >> file.txt
[eavernikovskaya@eavernikovskaya ~]$
```

Рис. 3.4: ~/ в file.txt

Выводим имена всех файлов из file.txt, имеющих расширение .conf с помощью команды *grep .conf file.txt* (рис. 3.5)

```
[eavernikovskaya@eavernikovskaya ~]$ grep .conf file.txt
-rw-r--r--. 1 root root 833 фев 10 2023 appstream.conf
-rw-r--r--. 1 root root 55 янв 29 03:00 asound.conf
-rw-r--r--. 1 root root 1372 дек 5 03:00 chrony.conf
drwxr-xr-x. 1 root root 18 ноя 1 04:05 dconf
-rw-r--r--. 1 root root 28602 фев 13 03:00 dnsmasq.conf
-rw-r--r--. 1 root root 117 ноя 16 03:00 dracut.conf
drwxr-xr-x. 1 root root 0 ноя 16 03:00 dracut.conf.d
-rw-r--r--. 1 root root 20 фев 24 2022 fprintd.conf
-rw-r--r--. 1 root root 38 авг 9 2023 fuse.conf
-rw-r--r--. 1 root root 9 июл 25 2023 host.conf
```

Рис. 3.5: .conf в file.txt

После чего записываем их в новый текстовый файл conf.txt введя команду *grep .conf file.txt > conf.txt* (рис. 3.6), (рис. 3.7), (рис. 3.8)

```
[eavernikovskaya@eavernikovskaya ~]$ grep .conf file.txt > conf.txt
[eavernikovskaya@eavernikovskaya ~]$
```

Рис. 3.6: .conf в conf.txt

```
[eavernikovskaya@eavernikovskaya ~]$ ls -l
итого 132
-rw-r--r--. 1 eavernikovskaya eavernikovskaya 0 мар 18 17:56 abc1
drwxr--r--. 1 eavernikovskaya eavernikovskaya 0 мар 18 18:02 australia
-rw-r--r--. 1 eavernikovskaya eavernikovskaya 587 мар 26 13:17 conf.txt
```

Рис. 3.7: Создался conf.txt



```
[eavernikovskaya@eavernikovskaya ~]$ head conf.txt
-rw-r--r--. 1 root root 833 фев 10 2023 appstream.conf
-rw-r--r--. 1 root root 55 янв 29 03:00 asound.conf
-rw-r--r--. 1 root root 1372 дек 5 03:00 chrony.conf
drwxr-xr-x. 1 root root 18 ноя 1 04:05 dconf
-rw-r--r--. 1 root root 28602 фев 13 03:00 dnsmasq.conf
-rw-r--r--. 1 root root 117 ноя 16 03:00 dracut.conf
drwxr-xr-x. 1 root root 0 ноя 16 03:00 dracut.conf.d
-rw-r--r--. 1 root root 20 фев 24 2022 fprintd.conf
-rw-r--r--. 1 root root 38 авг 9 2023 fuse.conf
-rw-r--r--. 1 root root 9 июл 25 2023 host.conf
[eavernikovskaya@eavernikovskaya ~]$
```

Рис. 3.8: Содержимое файла conf.txt

Определяем, какие файлы в нашем домашнем каталоге имеют имена, начинавшиеся с символа `c` введя команду `find ~ -name "c*" -print` (рис. 3.9), (рис. 3.10), (рис. 3.11), (рис. 3.12)

```
[eavernikovskaya@eavernikovskaya ~]$ find ~ -name "c*" -print
/home/eavernikovskaya/.mozilla/firefox/rv87tpji.default-release/crashes
/home/eavernikovskaya/.mozilla/firefox/rv87tpji.default-release/compatibility.ini
/home/eavernikovskaya/.mozilla/firefox/rv87tpji.default-release/cookies.sqlite
/home/eavernikovskaya/.mozilla/firefox/rv87tpji.default-release/cert9.db
/home/eavernikovskaya/.mozilla/firefox/rv87tpji.default-release/storage/permanent/chrome
/home/eavernikovskaya/.mozilla/firefox/rv87tpji.default-release/content-prefs.sqlite
/home/eavernikovskaya/.mozilla/firefox/rv87tpji.default-release/cookies.sqlite-wal
/home/eavernikovskaya/.mozilla/firefox/rv87tpji.default-release/containers.json
/home/eavernikovskaya/.cache/fontconfig/c4be257954870c0bf6972134c1de66d5-1e64.cache-8
/home/eavernikovskaya/.cache/fontconfig/cd93a7c10a59c5398bfa30047da1f86f-1e64.cache-8
/home/eavernikovskaya/.cache/fontconfig/c50efc045d30b64cc7bf686518957ef8-1e64.cache-8
/home/eavernikovskaya/.cache/fontconfig/c793051652af0a082fa4b8a7ceb4a8c7-1e64.cache-8
/home/eavernikovskaya/.cache/fontconfig/ce0d969af2176489fbefad4fe7aa6451-1e64.cache-8
```

Рис. 3.9: Файлы в ~/ начинающиеся на c

```
[eavernikovskaya@eavernikovskaya ~]$ ls ~/c*
/home/eavernikovskaya/conf.txt
```

Рис. 3.10: Другой вариант команды (1)

```
[eavernikovskaya@eavernikovskaya ~]$ ls -a ~ | grep "^c"
conf.txt
[eavernikovskaya@eavernikovskaya ~]$
```

Рис. 3.11: Другой вариант команды (2)

```
[eavernikovskaya@eavernikovskaya ~]$ ls -lR | grep c*
-rw-r--r--. 1 eavernikovskaya eavernikovskaya 42808 мар 26 19:44 conf.txt
```

Рис. 3.12: Другой вариант команды (3)

Выводим на экран (по странично) имена файлов из каталога /etc, начинающиеся с символа h введя `find /etc -name "h*" -print` (рис. 3.13)

```
[eavernikovskaya@eavernikovskaya ~]$ sudo find /etc -name "h*" -print
[sudo] пароль для eavernikovskaya:
/etc/avahi/hosts
/etc/firewalld/helpers
/etc/libibverbs.d/hfi1verbs.driver
/etc/libibverbs.d/hns.driver
/etc/systemd/homed.conf
/etc/udev/hwdb.d
/etc/udev/hwdb.bin
/etc/host.conf
/etc/hosts
/etc/hostname
[eavernikovskaya@eavernikovskaya ~]$
```

Рис. 3.13: Файлы в /etc начинающиеся на h

Запускаем в фоновом режиме процесс, который будет записывать в файл ~/logfile файлы, имена которых начинаются с log с помощью команды `find ~ -name "log*" -print > logfile &` (рис. 3.14), (рис. 3.15), (рис. 3.16)

```
[eavernikovskaya@eavernikovskaya ~]$ find ~ -name "log*" -print > logfile &
[1] 1829
[eavernikovskaya@eavernikovskaya ~]$
```

Рис. 3.14: Работа с logfile

```
[eavernikovskaya@eavernikovskaya ~]$ ls -l
итого 136
-rw-r--r--. 1 eavernikovskaya eavernikovskaya      0 мар 18 17:56 abc1
drwxr--r--. 1 eavernikovskaya eavernikovskaya      0 мар 18 18:02 australia
-rw-r--r--. 1 eavernikovskaya eavernikovskaya    587 мар 26 13:17 conf.txt
-rw-rw-r--. 1 eavernikovskaya eavernikovskaya      0 мар 18 18:40 feathers
-rw-r--r--. 1 eavernikovskaya eavernikovskaya 108062 мар 26 13:14 file.txt
drwxr-xr-x. 1 eavernikovskaya eavernikovskaya    132 мар  3 21:29 git-extended
drwxr-xr-x. 1 eavernikovskaya eavernikovskaya      0 мар 12 19:21 katerok
-rw-r--r--. 1 eavernikovskaya eavernikovskaya   18657 мар 12 14:17 LICENSE
-rw-r--r--. 1 eavernikovskaya eavernikovskaya    914 мар 26 13:24 logfile
```

Рис. 3.15: Создался logfile

```

[eavernikovskaya@eavernikovskaya ~]$ cat logfile
/home/eavernikovskaya/.mozilla/firefox/rv87tpji.default-release/logins.json
/home/eavernikovskaya/.local/share/keyrings/login.keyring
/home/eavernikovskaya/.local/share/chezmoi/.git/logs
/home/eavernikovskaya/git-extended/.git/logs
/home/eavernikovskaya/git-extended/node_modules/handlebars/lib/handlebars/helpers/log.js
/home/eavernikovskaya/git-extended/node_modules/handlebars/lib/handlebars/logger.js
/home/eavernikovskaya/git-extended/node_modules/handlebars/dist/amd/handlebars/helpers/log.js
/home/eavernikovskaya/git-extended/node_modules/handlebars/dist/amd/handlebars/logger.js
/home/eavernikovskaya/git-extended/node_modules/handlebars/dist/cjs/handlebars/helpers/log.js
/home/eavernikovskaya/git-extended/node_modules/handlebars/dist/cjs/handlebars/logger.js
/home/eavernikovskaya/git-extended/node_modules/neo-async/log.js
/home/eavernikovskaya/.password-store/.git/logs
/home/eavernikovskaya/logfile
[1]+  Завершён      find ~ -name "log*" -print > logfile
[eavernikovskaya@eavernikovskaya ~]$

```

Рис. 3.16: Содержимое файла logfile

После удаляем файл ~/logfile (рис. 3.17)

```

[eavernikovskaya@eavernikovskaya ~]$ rm logfile
[eavernikovskaya@eavernikovskaya ~]$ ls -l
итого 132
-rw-r--r--. 1 eavernikovskaya eavernikovskaya      0 мар 18 17:56 abc1
drwxr--r--. 1 eavernikovskaya eavernikovskaya      0 мар 18 18:02 australia
-rw-r--r--. 1 eavernikovskaya eavernikovskaya    587 мар 26 13:17 conf.txt
-rw-rw-r--. 1 eavernikovskaya eavernikovskaya      0 мар 18 18:40 feathers
-rw-r--r--. 1 eavernikovskaya eavernikovskaya 108862 мар 26 13:14 file.txt
drwxr-xr-x. 1 eavernikovskaya eavernikovskaya     132 мар  3 21:29 git-extended
drwxr-xr-x. 1 eavernikovskaya eavernikovskaya      0 мар 12 19:21 katerok
-rw-r--r--. 1 eavernikovskaya eavernikovskaya   18657 мар 12 14:17 LICENSE
-rw-r--r--. 1 eavernikovskaya eavernikovskaya      0 мар 18 17:24 may
drwx--x--x. 1 eavernikovskaya eavernikovskaya      0 мар 18 17:35 monthly
-r-xr--r--. 1 eavernikovskaya eavernikovskaya      0 мар 18 18:05 my_os
drwx--x--x. 1 eavernikovskaya eavernikovskaya     26 мар 18 18:47 play
drwxr-xr-x. 1 eavernikovskaya eavernikovskaya     14 мар 18 17:21 reports
drwxr-xr-x. 1 eavernikovskaya eavernikovskaya     28 мар 18 18:00 ski.places
drwxr-xr-x. 1 eavernikovskaya eavernikovskaya      0 фев 20 19:06 Видео
drwxr-xr-x. 1 eavernikovskaya eavernikovskaya      0 фев 20 19:06 Документы
drwxr-xr-x. 1 eavernikovskaya eavernikovskaya     90 фев 20 20:11 Загрузки
drwxr-xr-x. 1 eavernikovskaya eavernikovskaya    156 мар 12 13:47 Изображения
drwxr-xr-x. 1 eavernikovskaya eavernikovskaya      0 фев 20 19:06 Музыка
drwxr-xr-x. 1 eavernikovskaya eavernikovskaya      0 фев 20 19:06 Общедоступные
drwxr-xr-x. 1 eavernikovskaya eavernikovskaya      0 фев 20 19:06 'Рабочий стол'
drwxr-xr-x. 1 eavernikovskaya eavernikovskaya      0 фев 20 19:06 Шаблоны
[eavernikovskaya@eavernikovskaya ~]$

```

Рис. 3.17: Удаление logfile

Запустить из консоли в фоновом режиме редактор gedit введя *gedit &* (рис. 3.18), (рис. 3.19)

```

[eavernikovskaya@eavernikovskaya ~]$ gedit &
[1] 2017
[eavernikovskaya@eavernikovskaya ~]$

```

Рис. 3.18: Запуск gedit

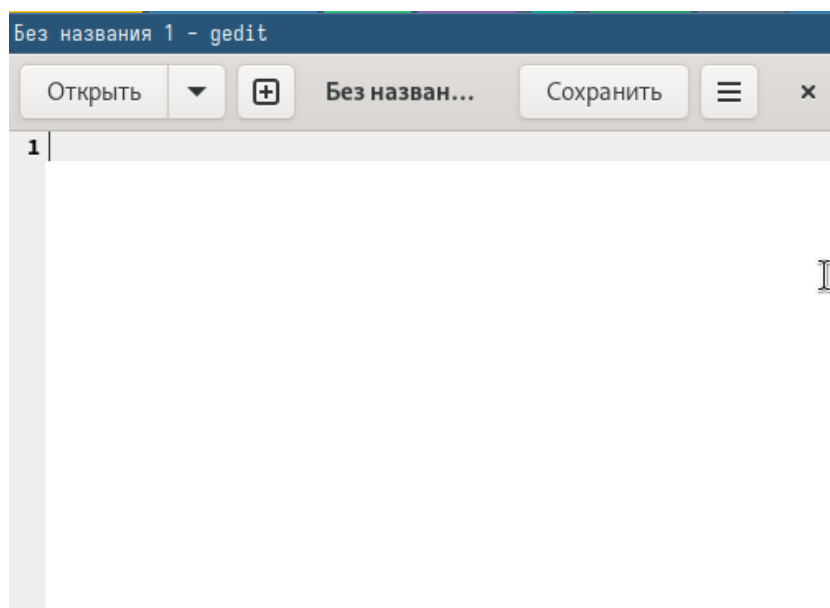


Рис. 3.19: gedit в фоновом режиме

Определяем идентификатор процесса gedit, используя команду ps, конвейер и фильтр grep. Варианты команд: 1. *ps aux | grep gedit* 2. *pgrep gedit* 3. *ps aux | grep gedit | grep -v grep* (рис. 3.20)

```
[eavernikovskaya@eavernikovskaya ~]$ ps aux | grep gedit
eaverni+  2017  2.7  3.0 851012 61524 pts/0    Sl   13:27  0
:02 gedit
eaverni+  2039  0.0  0.1 222456  2432 pts/0    S+   13:29  0
:00 grep --color=auto gedit
[eavernikovskaya@eavernikovskaya ~]$ pgrep gedit
2017
[eavernikovskaya@eavernikovskaya ~]$ ps aux | grep gedit | grep
-v grep
eaverni+  2017  2.6  3.1 851492 62036 pts/0    Sl   13:27  0
:04 gedit
[eavernikovskaya@eavernikovskaya ~]$
```

Рис. 3.20: Идентификатор процесса gedit

Читаем справку (man) команды kill, после чего используем её для завершения процесса gedit. Вводим *kill* (рис. 3.21), (рис. 3.22)

```
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, and the possibility to specify processes by command
    name, are local extensions.

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

Рис. 3.21: Справка по команде kill

```
[eavernikovskaya@eavernikovskaya ~]$ kill 2017
[eavernikovskaya@eavernikovskaya ~]$
```

Рис. 3.22: Завершение процесса gedit

Получаем более подробную информацию о командах df и du, с помощью команды man (рис. 3.23), (рис. 3.24), (рис. 3.25)

```
[eavernikovskaya@eavernikovskaya ~]$ man df
[eavernikovskaya@eavernikovskaya ~]$ man du
[eavernikovskaya@eavernikovskaya ~]$
```

Рис. 3.23: Использование команды man для df и du

```
DE(1) User Commands DE(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 available on unmounted file systems, because on most kinds of systems doing so requires very non-portable intimate knowledge of file system structures.
```

Рис. 3.24: Справка по 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 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
```

Рис. 3.25: Справка по команде du

Выполняем команду df и du. Первая показывает размер каждого смонтированного раздела диска, вторая и показывает число килобайт, используемое каждым файлом или каталогом (рис. 3.26), (рис. 3.27)

```
[eavernikovskaya@eavernikovskaya ~]$ df -vi
Файловая система  Инодов  ИИспользовано  ИСвободно  ИИспользовано%  Смонтировано в
/dev/sda3          0          0          0             1% /dev
devtmpfs           243963      524      243439        1% /dev
tmpfs              249211        8      249203        1% /dev/shm
efivarfs           0           0           0             - /sys/firmware/efi/efivars
tmpfs              819200      875      818325        1% /run
tmpfs              1048576     30      1048546        1% /tmp
/dev/sda3          0           0           0             - /home
/dev/sda2          65536       38       65498         1% /boot
/dev/sda1          0           0           0             - /boot/efi
work               1000      -999000     1000000        - /media/sf_work
tmpfs              49842      106       49736         1% /run/user/1000
```

Рис. 3.26: Выполнение команды df

```
[eavernikovskaya@eavernikovskaya ~]$ du -a ~/file.txt
108    /home/eavernikovskaya/file.txt
[eavernikovskaya@eavernikovskaya ~]$ du -a ~/conf.txt
4      /home/eavernikovskaya/conf.txt
[eavernikovskaya@eavernikovskaya ~]$ du -a ~/Изображения/
112    /home/eavernikovskaya/Изображения/2024-03-12T13:47:18,161506051+03:00.png
112    /home/eavernikovskaya/Изображения/2024-03-12T13:47:18,309654188+03:00.png
224    /home/eavernikovskaya/Изображения/
[eavernikovskaya@eavernikovskaya ~]$
```

Рис. 3.27: Выполнение команды `du`

Далее смотрим справку по команде `find` (рис. 3.28), (рис. 3.29)

```
[eavernikovskaya@eavernikovskaya ~]$ man find
[eavernikovskaya@eavernikovskaya ~]$
```

Рис. 3.28: Использование команды `man` для `find`

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

NAME
    find - search for files in a directory hierarchy

SYNOPSIS
    find [-H] [-L] [-P] [-D debugopts] [-O level] [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 example if you are using it to search di-
    rectories 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 de-
    tail and discussion than this manual page, so you may find it a more useful source of information.
```

Рис. 3.29: Справка по команде `find`

Используя команду `find` выводим имена всех директорий, имеющихя в на-  
шем домашнем каталоге. Вводим команду `find ~ -type d` (рис. 3.30), (рис. 3.31)

```
[eavernikovskaya@eavernikovskaya ~]$ find ~ -type d
```

Рис. 3.30: Использование команды `find`



```
/home/eavernikovskaya/.password-store/bin  
/home/eavernikovskaya/.bashrc.d  
/home/eavernikovskaya/.cofig  
/home/eavernikovskaya/.cofig/sway  
/home/eavernikovskaya/.cofig/sway/config.d  
/home/eavernikovskaya/katerok  
/home/eavernikovskaya/reports  
/home/eavernikovskaya/reports/monthly  
/home/eavernikovskaya/reports/monthly/monthly  
/home/eavernikovskaya/monthly  
/home/eavernikovskaya/ski.plases  
/home/eavernikovskaya/ski.plases/equipment  
/home/eavernikovskaya/ski.plases/plans  
/home/eavernikovskaya/australia  
/home/eavernikovskaya/play  
/home/eavernikovskaya/play/games
```

Рис. 3.31: Имена всех директорий в ~/



## 4 Ответы на контрольные вопросы

### 1. Какие потоки ввода вывода вы знаете?

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

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

- Символ `>` используется для перенаправления вывода команды в файл. Если файл уже существует, его содержимое будет полностью перезаписано.
- Символ `»` также используется для перенаправления вывода команды в файл, но с дополнением данных в конец файла, не перезаписывая существующее содержимое.

### 3. Что такое конвейер?

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

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

Главное отличие между программой и процессом заключается в том, что программа - это набор инструкций, который позволяет ЦПУ выполнять определенную задачу, в то время как процесс - это исполняемая программа.

## 5. Что такое PID и GID?

PPID - (parent process ID) идентификатор родительского процесса. Процесс может порождать и другие процессы. UID, GID - реальные идентификаторы пользователя и его группы, запустившего данный процесс.

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

Запущенные фоном программы называются задачами (jobs). Ими можно управлять с помощью команды `jobs`, которая выводит список запущенных в данный момент задач.

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

Команда `htop` похожа на команду `top` по выполняемой функции: они обе показывают информацию о процессах в реальном времени, выводят данные о потреблении системных ресурсов и позволяют искать, останавливать и управлять процессами. У обеих команд есть свои преимущества. Например, в программе `htop` реализован очень удобный поиск по процессам, а также их фильтрация. В команде `top` это не так удобно — нужно знать кнопку для вывода функции поиска. Зато в `top` можно разделять область окна и выводить информацию о процессах в соответствии с разными настройками. В целом `top` намного более гибкая в настройке отображения процессов.

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

Команда `find` - это одна из наиболее важных и часто используемых утилит системы Linux. 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. Утилита `find` предустановлена по умолчанию во всех

Linux дистрибутивах, поэтому вам не нужно будет устанавливать никаких дополнительных пакетов. Это очень важная находка для тех, кто хочет использовать командную строку наиболее эффективно. Команда `find` имеет такой синтаксис: `find [папка] [параметры] критерий шаблон [действие]` Пример: `find /etc -name "h*" -print`

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

Да, можно. Синтаксис команды: `find ~ -type f -exec grep -H 'текстДляПоиска' {}` ;

10. Как определить объем свободной памяти на жёстком диске?

С помощью команды `df -h`.

11. Как определить объем вашего домашнего каталога?

С помощью команды `du -s`.

12. Как удалить зависший процесс?

С помощью команды `kill% номер задачи`

## 5 Выводы

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

## **6 Список литературы**

Не пользовалась сайтами.