

Лабораторная работа №7

Журавлев Георгий Иванович

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.

Выполнение лабораторной работы

1. Запишем в файл file.txt названия файлов, содержащихся в каталоге /etc и домашнем каталоге



```
g_zhuravlev@g ~$ touch file.txt
g_zhuravlev@g ~$ ls /etc > file.txt
g_zhuravlev@g ~$ ls ~ >> file.txt
g_zhuravlev@g ~$
```

✓	615	14:57:25
✓	616	14:59:45
✓	617	15:00:15
✓	618	15:00:35

Рис. 1: Запись названий в файл

```
g_zhuravlev@g ~$ grep .conf file.txt
adduser.conf
apg.conf
appstream.conf
brltty.conf
ca-certificates.conf
ca-certificates.conf.dpkg-old
dconf
debconf.conf
deluser.conf
e2scrub.conf
fprintd.conf
fuse.conf
gal.conf
gconf
hdparm.conf
host.conf
insserv.conf.d
kernel-img.conf
kerneloops.conf
ld.so.conf
ld.so.conf.d
libao.conf
libaudit.conf
logrotate.conf
ltrace.conf
manpath.config
mke2fs.conf
mtools.conf
nsswitch.conf
pan.conf
pnm2ppa.conf
popularity-contest.conf
preload.conf
resolv.conf
rsyslog.conf
rygel.conf
sensors3.conf
sysctl.conf
ucf.conf
usb_modeswitch.conf
vconsole.conf
xattr.conf
g_zhuravlev@g ~$ grep .conf file.txt >> conf.txt
g_zhuravlev@g ~$
```

Рис. 2: Вывод имён с расширением .conf

```
g_zhuravlev@g ~$ ls -l | grep c*
conf.txt:ca-certificates.conf
conf.txt:ca-certificates.conf.dpkg-old
grep: cpp-projects: Is a directory
```

Рис. 3: 1 способ определения

```
g_zhuravlev@g ~$ find ~ -type f -name "c*" -print
/home/g_zhuravlev/Desktop/routine/reading/comments stats.txt
/home/g_zhuravlev/Desktop/routine/music routine/ear training/ноты/audio/c tonality.flac
/home/g_zhuravlev/Desktop/routine/music routine/ear training/Diktanti/var 14.03 cde/cde.mp3
```

Рис. 4: 2 способ определения

```
g_zhuravlev@g ~$ ls -l | grep ^* | less
g_zhuravlev@g ~$
```

Рис. 5: постраничный вывод



Рис. 6: результат

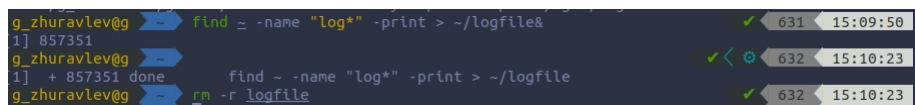


Рис. 7: Выполнение действий

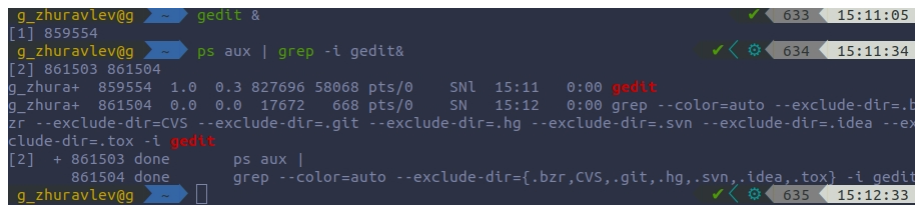


Рис. 8: запуск gedit



Рис. 9: определение PID

```

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 speci-
    fied 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 indi-
    cates 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.

    -l, --list [signal]
        List signal names. This option has optional argument, which will convert signal
        number to signal name, or other way round.

    -L, --table
        List signal names in a nice table.

NOTES
    Your shell (command line interpreter) may have a built-in kill command. You may
    need to run the command described here as /bin/kill to solve the conflict.

EXAMPLES
    kill -9 -1
        Kill all processes you can kill.

    kill -l 11
        Translate number 11 into a signal name.

    kill -L
        List the available signal choices in a nice table.

    kill 123 543 2341 3453
        Send the default signal, SIGTERM, to all those processes.

SEE ALSO
    kill(2), killall(1), nice(1), pkill(1), renice(1), signal(7), skill(1)

STANDARDS
    This command meets appropriate standards. The -L flag is Linux-specific.
    Manual page kill(1) line 1 (press h for help or q to quit)

```

Рис. 10: мануал команды kill

```

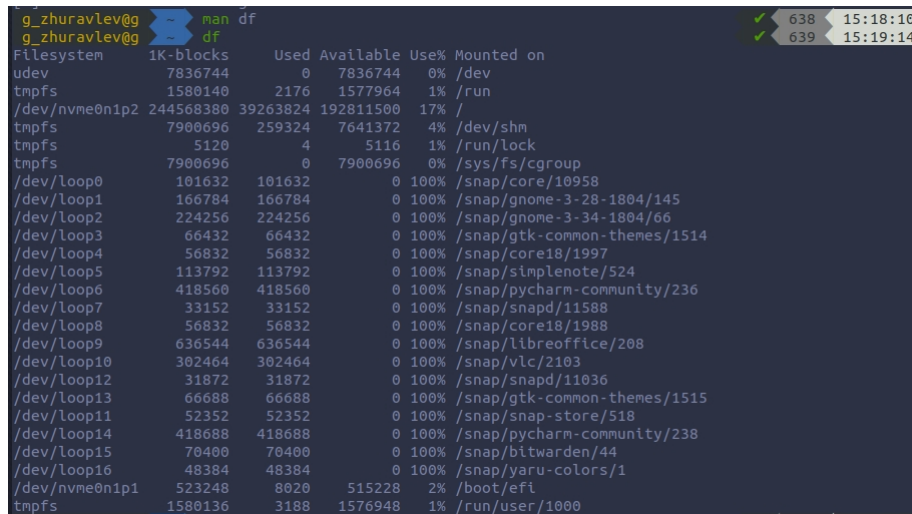
g_zhuravlev@g ➜ kill -9 859554 637 15:17:42
** (gedit:859554): WARNING **: 15:18:10.220: Error querying file info: Error when getting information
for file "/home/g_zhuravlev/.zsh_history.LOCK": No such file or directory
[1] + 859554 killed gedit
g_zhuravlev@g ➜ 638 15:18:10

```

Рис. 11: Завершение процесса по PID

2. Выводим имена всех файлов из file.txt, имеющих расширение .conf, далее запишем их в новый текстовый файл conf.txt.
3. Определим, какие файлы в нашем домашнем каталоге имеют имена, начинающиеся с символа “с”. 2 типа.
4. Выведем на экран (по-странично) имена файлов из каталога /etc, начинающиеся с символа h.
5. Запустим в фоновом режиме процесс, который будет записывать в файл ~/logfile файлы, имена которых начинаются с log и удалим файл ~/logfile.
6. Запустим из консоли в фоновом режиме редактор gedit.
7. Определим идентификатор процесса gedit, используя команду ps, конвейер и фильтр grep. (PID = 859554)
8. Прочтём справку (man) команды kill, после чего воспользуемся ею для завершения процесса gedit.
9. Выполним команды df и du, предварительно получив более подробную информацию об этих командах, с помощью команды man.

9.1. df



```

g_zhuravlev@g ~$ man df
g_zhuravlev@g ~$ df
Filesystem      1K-blocks    Used Available Use% Mounted on
udev            7836744         0   7836744    0% /dev
tmpfs           1580140         2176   1577964    1% /run
/dev/nvme0n1p2 244568380 39263824 192811500   17% /
tmpfs           7900696 259324 7641372    4% /dev/shm
tmpfs           5120         4    5116    1% /run/lock
tmpfs           7900696         0   7900696    0% /sys/fs/cgroup
/dev/loop0      101632 101632    0 100% /snap/core/10958
/dev/loop1      166784 166784    0 100% /snap/gnome-3-28-1804/145
/dev/loop2      224256 224256    0 100% /snap/gnome-3-34-1804/66
/dev/loop3       66432  66432    0 100% /snap/gtk-common-themes/1514
/dev/loop4       56832  56832    0 100% /snap/core18/1997
/dev/loop5      113792 113792    0 100% /snap/simplenote/524
/dev/loop6      418560 418560    0 100% /snap/pycharm-community/236
/dev/loop7       33152  33152    0 100% /snap/snapd/11588
/dev/loop8       56832  56832    0 100% /snap/core18/1988
/dev/loop9      636544 636544    0 100% /snap/libreoffice/208
/dev/loop10     302464 302464    0 100% /snap/vlc/2103
/dev/loop12      31872  31872    0 100% /snap/snapd/11036
/dev/loop13      66688  66688    0 100% /snap/gtk-common-themes/1515
/dev/loop11      52352  52352    0 100% /snap/snap-store/518
/dev/loop14      418688 418688    0 100% /snap/pycharm-community/238
/dev/loop15      70400  70400    0 100% /snap/bitwarden/44
/dev/loop16      48384  48384    0 100% /snap/yaru-colors/1
/dev/nvme0n1p1   523248    8020   515228    2% /boot/efi
tmpfs           1580136         3188   1576948    1% /run/user/1000

```

Рис. 12: выполнение команды df

9.2. du

```
g_zhuravlev@g ~$ man du
g_zhuravlev@g ~$ du
4      ./test/dir2
4      ./test/dir1
4      ./test/dir3
16     ./test
4      ./gconf
20     ./texlive2019/texmf-var/fonts/source/lh/lh-t2a
24     ./texlive2019/texmf-var/fonts/source/lh
28     ./texlive2019/texmf-var/fonts/source
128    ./texlive2019/texmf-var/fonts/pk/ljfour/lh/lh-t2a
132    ./texlive2019/texmf-var/fonts/pk/ljfour/lh
136    ./texlive2019/texmf-var/fonts/pk/ljfour
140    ./texlive2019/texmf-var/fonts/pk
20     ./texlive2019/texmf-var/fonts/tfm/lh/lh-t2a
24     ./texlive2019/texmf-var/fonts/tfm/lh
28     ./texlive2019/texmf-var/fonts/tfm
200    ./texlive2019/texmf-var/fonts
204    ./texlive2019/texmf-var
208    ./texlive2019
8      ./python-projects/.idea/inspectionProfiles
28     ./python-projects/.idea
8      ./python-projects/1st.project/.idea/inspectionProfiles
32     ./python-projects/1st.project/.idea
28     ./python-projects/1st.project/venv/bin
4      ./python-projects/1st.project/venv/include
4      ./python-projects/1st.project/venv/lib/python3.8/site-packages
8      ./python-projects/1st.project/venv/lib/python3.8
12     ./python-projects/1st.project/venv/lib
56     ./python-projects/1st.project/venv
92     ./python-projects/1st.project
124    ./python-projects
```

Рис. 13: выполнение команды du

```
17048844
g_zhuravlev@g ~$ man find
g_zhuravlev@g ~$ find . -type d
.
./test
./test/dir2
./test/dir1
./test/dir3
./gconf
./texlive2019
./texlive2019/texmf-var
./texlive2019/texmf-var/fonts
./texlive2019/texmf-var/fonts/source
./texlive2019/texmf-var/fonts/source/lh
./texlive2019/texmf-var/fonts/source/lh/lh-t2a
./texlive2019/texmf-var/fonts/pk
./texlive2019/texmf-var/fonts/pk/ljfour
./texlive2019/texmf-var/fonts/pk/ljfour/lh
./texlive2019/texmf-var/fonts/pk/ljfour/lh/lh-t2a
./texlive2019/texmf-var/fonts/tfm
./texlive2019/texmf-var/fonts/tfm/lh
./texlive2019/texmf-var/fonts/tfm/lh/lh-t2a
```

Рис. 14: вывод всех директорий

10. Воспользовавшись справкой команды `find`, выведем имена всех директорий, имеющихся в нашем домашнем каталоге.

Вывод

Благодаря этой лабораторной работе, я научился: пользоваться инструментами поиска файлов и фильтрацией текстовых данных. Также я приобрел навыки: проверки диска; обслуживания файловых систем; управления процессами.

Контрольные вопросы

1. В системе по умолчанию открыто три специальных потока:
 - `stdin` — стандартный поток ввода (по умолчанию: клавиатура), файловый дескриптор 0;
 - `stdout` — стандартный поток вывода (по умолчанию: консоль), файловый дескриптор 1;
 - `stderr` — стандартный поток вывод сообщений об ошибках (по умолчанию: консоль), файловый дескриптор 2.
2. `>` - перенаправление `stdout` в файл. Если файл отсутствовал, то он создаётся, иначе - перезаписывается. `>>` - перенаправление `stdout` в файл. Если файл отсутствовал, то он создаётся, иначе - добавляется.
3. Конвейер (`pipe`)- инструмент для объединения простых команд или утилит в цепочки.
4. Процесс это - совокупность программного кода и данных, загруженных в память ЭВМ. Любой командой можно управлять. Программа - единица выполняемого процесса.
5. `PID` — `personal identifier` (идентификатор) процесса в многозадачной ОС. `GID` – `group identifier` - идентификатор группы.
6. Запущенные фоном программы называются задачами (`jobs`). Ими можно управлять с помощью команд `fg` и `bg`.
7. Команда `htop` — продвинутый монитор процессов, показывает динамический список системных процессов. Команда `top` в Linux системах позволяет вывести в виде таблицы перечень запущенных процессов и оценить, какой объем ресурсов они потребляют.
8. Команда `find` используется для поиска и отображения имён файлов. Формат команды: `find путь [-опции]`

Примеры:

- вывести на экран имена файлов из вашего домашнего каталога и его подкаталогов, начинающихся на `f`:
`find ~ -name "f*" -print`
- вывести на экран имена файлов в каталоге `/etc`, начинающихся с символа `p`:
`find /etc -name "p*" -print`

```
find /etc -name "p*" -print
```

9. Найти файл по контексту позволяет команда `grep`.

Формат команды: `grep строка имя_файла`

Примеры:

- показать строки во всех файлах в вашем домашнем каталоге с именами, начинающимися на `f`, в которых

- найти в текущем каталоге все файлы, в имени которых есть буквосочетание «`car`»: `ls -l | grep car`

10. `Df` - позволяет определить объем свободной памяти на жёстком диске позволяет.

11. `df /home/<user_name>` - позволяет определить объем домашнего каталога позволяет команда `df /ho`

12. Удалить зависший процесс можно командой `kill %номер задачи`.