

Отчёта по лабораторной работе 8

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

Дельгадильо Валерия

Содержание

Цель работы	5
Лабораторной работы	6
Контрольные вопросы	13
Выводы	15
Список литературы	16

Список иллюстраций

1	Запишем в файл file.txt названия файлов, содержащихся в каталоге /etc	6
2	Вывод файлов №1	6
3	Допишем в этот же файл названия файлов, содержащихся в домашнем каталоге	6
4	Вывод файлов №2	7
5	С помощью команды cat проверим, что в файле содержатся названия файлов как каталога /etc, так и домашнего каталога	8
6	Выведем имена всех файлов из file.txt, имеющих расширение .conf . .	8
7	Запишем их в новый текстовый файл conf.txt	8
8	Определим, какие файлы в домашнем каталоге имеют имена, начинающиеся с символа h	8
9	Запустим в фоновом режиме процесс, который будет записывать в файл ~/logfile файлы, имена которых начинаются с log	8
10	С помощью команды jobs проверим, что процесс работает в фоновом режиме	9
11	С помощью команды cat проверим, что в файле содержатся названия файлов, начинающихся на log	9
12	Проверим, что созданный файл logfile находится в домашнем каталоге	9
13	После удаления файла ~/logfile с помощью команды jobs увидим, что процесс всё ещё запущен	9
14	Определим идентификатор процесса gedit, используя команду ps, конвейер и фильтр grep	10
15	Прочтем справку (man) команд df	10
16	Прочтем справку (man) команд du	11
17	Прочтем справку (man) команд du	11
18	Завершим процесс с помощью команды kill, посылая сигнал SIGKILL, имеющий номер 9, процессу 3439	11
19	Выполним команду df	12
20	Выполним команду du	12

Список таблиц

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.

Лабораторной работы

```
[yvdeljgadiljo@fedora ~]$ ls /etc > file.txt  
[yvdeljgadiljo@fedora ~]$ cat file.txt
```

Рис. 1: Запишем в файл file.txt названия файлов, содержащихся в каталоге /etc

```
xdg  
xinetd.conf  
xinetd.d  
xml  
xrootd  
xtables.conf  
zsh
```

Рис. 2: Вывод файлов №1

```
[yvdeljgadiljo@fedora ~]$ ls ~ >> file.txt  
[yvdeljgadiljo@fedora ~]$ cat file.txt  
abrt
```

Рис. 3: Допишем в этот же файл названия файлов, содержащихся в домашнем каталоге

```
my_os
play
public
public_html
reports
ski.plases
tmp
work
Видео
Документы
Загрузки
Изображения
Музыка
Общедоступные
Рабочий стол
Шаблоны
```

Рис. 4: Вывод файлов №2

```
xrootd
xtables.conf
zsh
abc1
australia
bin
feathers
file.txt
GNUstep
```

Рис. 5: С помощью команды `cat` проверим, что в файле содержатся названия файлов как каталога `/etc`, так и домашнего каталога

```
[yvdeljgatiljo@fedora ~]$ grep .conf file.txt
```

Рис. 6: Выведем имена всех файлов из `file.txt`, имеющих расширение `.conf`

```
[yvdeljgatiljo@fedora ~]$ grep .conf file.txt > conf.txt
[yvdeljgatiljo@fedora ~]$ cat conf.txt
```

Рис. 7: Запишем их в новый текстовый файл `conf.txt`

```
[yvdeljgatiljo@fedora ~]$ find /etc -name "h*" -print
```

Рис. 8: Определим, какие файлы в домашнем каталоге имеют имена, начинающиеся с символа `h`

```
[yvdeljgatiljo@fedora ~]$ find / -name "log*" > ~/logfile &
```

Рис. 9: Запустим в фоновом режиме процесс, который будет записывать в файл `~/logfile` файлы, имена которых начинаются с `log`


```
[yvdeljgatiljo@fedora ~]$ jobs
```

Рис. 10: С помощью команды jobs проверим, что процесс работает в фоновом режиме

```
[yvdeljgatiljo@fedora ~]$ cat logfile
/home/yvdeljgatiljo/.mozilla/firefox/nw555gll.default-release/logins-backup.json
/home/yvdeljgatiljo/.mozilla/firefox/nw555gll.default-release/logins.json
/home/yvdeljgatiljo/.cache/pnpm/metadata/registry.npmjs.org/log-symbols.json
/home/yvdeljgatiljo/.local/share/keyrings/login.keyring
/home/yvdeljgatiljo/.local/share/Trash/files/os-intro.3/.git/logs
/home/yvdeljgatiljo/.local/share/Trash/files/os-intro.3/.git/modules/template/presentation/logs
/home/yvdeljgatiljo/.local/share/Trash/files/os-intro.3/.git/modules/template/report/logs
/home/yvdeljgatiljo/.local/share/Trash/files/yvdeljgatiljo.github.io/.git/logs
/home/yvdeljgatiljo/.local/share/Trash/files/blog.2/.git/logs
/home/yvdeljgatiljo/.local/share/Trash/files/blog.3/.git/logs
/home/yvdeljgatiljo/.local/share/Trash/files/yvdeljgatiljo.2.github.io/.git/logs
/home/yvdeljgatiljo/.local/share/pnpm/global/5/.pnpm/commitizen@4.3.0_@types+node@20.11.24_typescript@5.3.3/node_modules/commitizen/dist/git/log.js
/home/yvdeljgatiljo/.local/share/pnpm/global/5/.pnpm/commitizen@4.3.0_@types+node@20.11.24_typescript@5.3.3/node_modules/commitizen/logo
/home/yvdeljgatiljo/.local/share/pnpm/global/5/.pnpm/log-symbols@4.1.0
/home/yvdeljgatiljo/.local/share/pnpm/global/5/.pnpm/log-symbols@4.1.0/node_modules/log-symbols
/home/yvdeljgatiljo/.local/share/pnpm/global/5/.pnpm/ora@5.4.1/node_modules/log-symbols
/home/yvdeljgatiljo/.local/share/pnpm/global/5/.pnpm/node_modules/log-symbols
/home/yvdeljgatiljo/.local/share/pnpm/global/5/.pnpm/handlebars@4.7.8/node_modules/handlebars/lib/handlebars/helpers/log.js
```

Рис. 11: С помощью команды cat проверим, что в файле содержатся названия файлов, начинающихся на log

```
[yvdeljgatiljo@fedora ~]$ ls
abcl    conf.txt  feathers  LICENSE  monthly  reports  Видео  Изображения  'Рабочий стол'
australia Desktop  feathers1 logfile  my_os    ski.places  Документы  Музыка  Шаблоны
bin     Downloads file.txt  may      play     work      Загрузки  Общедоступные
```

Рис. 12: Проверим, что созданный файл logfile находится в домашнем каталоге

```
[yvdeljgatiljo@fedora ~]$ rm logfile
[yvdeljgatiljo@fedora ~]$ ls
abcl    conf.txt  feathers  LICENSE  my_os    ski.places  Документы  Музыка  Шаблоны
australia Desktop  feathers1 may      play     work      Загрузки  Общедоступные
bin     Downloads file.txt  monthly  reports  Видео  Изображения  'Рабочий стол'
```

Рис. 13: После удаления файла ~/logfile с помощью команды jobs увидим, что процесс всё ещё запущен

```
[yvdeljgatiljo@fedora ~]$ gedit &
[3] 2960
[yvdeljgatiljo@fedora ~]$ jobs
[1]- Остановлен    gedit
[2]+ Остановлен    gedit
[3]  Запущен      find ~ -name "log*" > ~/logfile &
```

```
[yvdeljgadiljo@fedora ~]$ ps aux | grep gedit
yvdeljg+  2883  0.0  0.1 596252  6656 pts/0    Tl   15:12   0:00 /usr/libexec/pk-command-not-found gedit
yvdeljg+  2893  0.0  0.1 596252  6656 pts/0    Tl   15:15   0:00 /usr/libexec/pk-command-not-found gedit
yvdeljg+  2961  0.1  0.1 596252  6656 pts/0    Sl   15:26   0:00 /usr/libexec/pk-command-not-found gedit
yvdeljg+  2973  1.0  0.0 222436  2432 pts/0    S+   15:27   0:00 grep --color=auto gedit
```

Рис. 14: Определим идентификатор процесса gedit, используя команду ps, конвейер и фильтр grep

```
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 cur-
    rently 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 ver-
    sion of df cannot show the space available on unmounted file systems,
    because on most kinds of systems doing so requires very nonportable in-
    timate knowledge of file system structures.
```

Рис. 15: Прочтем справку (man) команд df

```

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

```

Рис. 16: Прочтем справку (man) команд du

```
[yvdeljgatiljo@fedora ~]$ kill -9 3439
```

Рис. 17: Прочтем справку (man) команд du

```
[yvdeljgatiljo@fedora ~]$ df
```

Файловая система	1K-блоков	Использовано	Доступно	Использовано%	Смонтировано в
devtmpfs	4096	0	4096	0%	/dev
tmpfs	2002656	0	2002656	0%	/dev/shm
tmpfs	801064	1264	799800	1%	/run
/dev/sda3	51377152	18357704	32681352	36%	/
tmpfs	2002660	48	2002612	1%	/tmp
/dev/sda2	996780	263444	664524	29%	/boot
/dev/sda3	51377152	18357704	32681352	36%	/home
tmpfs	400528	112	400416	1%	/run/user/1000

Рис. 18: Завершим процесс с помощью команды kill, посылая сигнал SIGKILL, имеющий номер 9, процессу 3439

```
[yvdeljgatiljo@fedora ~]$ du  
8 /usr/lib/mozilla/extensions/
```

Рис. 19: Выполним команду df

```
[yvdeljgatiljo@fedora ~]$ find . -maxdepth 1 -type d
```

Рис. 20: Выполним команду du

Воспользовавшись справкой команды `find`, выведем имена всех директорий, имеющих в домашнем каталоге

С помощью `type d` мы попросили команду `find` искать только каталоги. С помощью `maxdepth 1` мы попросили команду `find` сохранить поиск только на текущем уровне (и не заходить в подкаталоги). Введённая команда также показывает скрытые каталоги.

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. Операция > создаёт операция » дополняет
3. Конвейер (pipe) служит для объединения простых команд или утилит в цепочки, в которых результат работы предыдущей команды передаётся последующей.
4. Компьютерная программа сама по себе – лишь пассивная последовательность инструкций. В то время как процесс – непосредственное выполнение этих инструкций
5. PID - идентификатор процесса, уникальный номер процесса в многозадачной операционной системе. GID - идентификатор группы
6. Запущенные фоном программы называются задачами (jobs). Ими можно управлять с помощью команды jobs, которая выводит список запущенных в данный момент задач
7. Top (table of processes) – консольная команда, которая выводит список работающих в системе процессов и информацию о них. Htop – хорошо известная утилита для мониторинга, аналог top
8. Команда find используется для поиска и отображения на экран имён файлов, соответствующих заданной строке символов. Формат команды: find (путь) (опции)
Пример: Вывести на экран имена файлов из вашего домашнего каталога и его

- подкаталогов, начинающихся на f: `find ~ -name "f*" -print`
9. Файл можно найти по контексту. Показать строки во всех файлах, в которых есть слово begin: `grep begin`
 10. Определить объем свободной памяти на жёстком диске можно с помощью команды `df`
 11. Команда `du` показывает число килобайт, используемое каждым файлом или каталогом. Чтобы найти объём домашнего каталога надо ввести команду `du ~` в терминал
 12. Зависший процесс можно завершить с помощью команды `kill`, указав опцию `-9` и номер процесса

Выводы

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

Список литературы

- GDB: The GNU Project Debugger. — URL: <https://www.gnu.org/software/gdb/>.
- GNU Bash Manual. — 2016. — URL: <https://www.gnu.org/software/bash/manual/>.
- Midnight Commander Development Center. — 2021. — URL: <https://midnight-commander.org/>.
- NASM Assembly Language Tutorials. — 2021. — URL: <https://asmtutor.com/>.
- Newham C. Learning the bash Shell: Unix Shell Programming. — O'Reilly Media, 2005. — 354 с. — (In a Nutshell). — ISBN 0596009658. — URL: <http://www.amazon.com/Learningbash-Shell-Programming-Nutshell/dp/0596009658>.
- Robbins A. Bash Pocket Reference. — O'Reilly Media, 2016. — 156 с. — ISBN 978-1491941591.
- The NASM documentation. — 2021. — URL: <https://www.nasm.us/docs.php>.
- Zarrelli G. Mastering Bash. — Packt Publishing, 2017. — 502 с. — ISBN 9781784396879.
- Колдаев В. Д., Lupin С. А. Архитектура ЭВМ. — М. : Форум, 2018.
- Куляс О. Л., Никитин К. А. Курс программирования на ASSEMBLER. — М. : Солон-Пресс, 2017.
- Новожилов О. П. Архитектура ЭВМ и систем. — М. : Юрайт, 2016.
- Расширенный ассемблер: NASM. — 2021. — URL: <https://www.opennet.ru/docs/RUS/nasm/>.

- Робачевский А., Немнюгин С., Стесик О. Операционная система UNIX. — 2-е изд. — БХВПетербург, 2010. — 656 с. — ISBN 978-5-94157-538-1.
- Столяров А. Программирование на языке ассемблера NASM для ОС Unix. — 2-е изд. — М. : МАКС Пресс, 2011. — URL: http://www.stolyarov.info/books/asm_unix.
- Таненбаум Э. Архитектура компьютера. — 6-е изд. — СПб. : Питер, 2013. — 874 с. — (Классика Computer Science).
- Таненбаум Э., Бос Х. Современные операционн