

Отчет по выполнению лабораторной работы №6

Операционные системы

Hanjiangtao

Содержание

1 Цель работы

Цель данной лабораторной работы - ознакомление с инструментами поиска файлов и фильтрации текстовых данных, а также приобретение практических навыков по управлению процессами (и заданиями), по проверке использования диска и по 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. Допи- шите 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, начинавшиеся с символа с? 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, имею- щихся в вашем домашнем каталоге.

3 Теоретическое введение

В интерфейсе командной строки есть очень полезная возможность перенаправления (переадресации) ввода и вывода (англ. термин I/O Redirection). Как мы уже заметили, многие программы выводят данные на экран. А ввод данных в терминале осуществляется с клавиатуры. С помощью специальных обозначений можно перенаправить вывод многих команд в файлы или иные устройства вывода (например, распечатать на принтере). Тоже самое и со вводом информации, вместо ввода данных с клавиатуры, для многих программ можно задать считывание символов их файла. Кроме того, можно даже вывод одной программы передать на ввод другой программе.

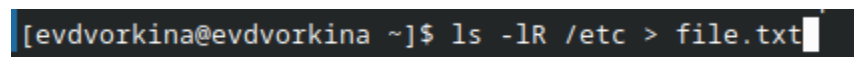
К каждой программе, запускаемой в командной строке, по умолчанию подключено три потока данных:

STDIN (0) — стандартный поток ввода (данные, загружаемые в программу). STDOUT (1) — стандартный поток вывода (данные, которые выводит программа). По умолчанию — терминал. STDERR (2) — стандартный поток вывода диагностических и отладочных сообщений (например, сообщениях об ошибках). По умолчанию — терминал.

Pipe (конвейер) – это однонаправленный канал межпроцессного взаимодействия. Термин был придуман Дугласом Макилроем для командной оболочки Unix и назван по аналогии с трубопроводом. Конвейеры чаще всего используются в shell-скриптах для связи нескольких команд путем перенаправления вывода одной команды (stdout) на вход (stdin) последующей, используя символ конвейера '|’.

4 Выполнение лабораторной работы

Я вошла в систему под соответствующим именем пользователя, открыла терминал, записала в файл file.txt названия файлов из каталога /etc с помощью перенаправления “>” (и файл создала, и записала в него то, что могло быть выведено ls -lR /etc). В файл я добавила также все файлы из подкаталогов (рис. 1).



```
[evdvorkina@evdvorkina ~]$ ls -lR /etc > file.txt
```

Figure 1: Запись в файл

Проверила, что в файл записались нужные значения с помощью утилиты head, она выводит первые 10 строк файла на экран (рис. 2).

```

-rw-r--r--. 1 root root 1344 янв 29 10:28 fedora-updates-testing.re
[evdvorkina@evdvorkina ~]$ head file.txt
/etc:
итого 1416
drwxr-xr-x. 1 root root      126 ноя  5 11:18 abrt
-rw-r--r--. 1 root root       16 фев 12 14:05 adjtime
-rw-r--r--. 1 root root    1529 июл 20  2022 aliases
drwxr-xr-x. 1 root root       70 окт 24 18:29 alsa
drwxr-xr-x. 1 root root    1258 фев 25 05:32 alternatives
drwxr-xr-x. 1 root root       56 ноя  5 11:18 anaconda
-rw-r--r--. 1 root root     541 июл 21  2022 anacrontab
-rw-r--r--. 1 root root     269 июл 20  2022 anthy-unicode.conf

```

Figure 2: Вывод содержимого файла

Добавила в созданный файл имена файлов из домашнего каталога, используя перенаправление ">>" в режиме добавления (рис. 3).

```

[evdvorkina@evdvorkina ~]$ ls -lR ~/ >> file.txt
[evdvorkina@evdvorkina ~]$

```

Figure 3: Добавление данных в файл

Вывела на экран имена всех файлов, имеющих расширение ".conf" с помощью утилиты grep (рис. 4).

```

[evdvorkina@evdvorkina ~]$ grep .conf file.txt
-rw-r--r--. 1 root root     269 июл 20  2022 anthy-unicode.conf
-rw-r--r--. 1 root root     833 авг 22  2022 appstream.conf
-rw-r--r--. 1 root root      55 дек  7 17:18 asound.conf
drwxr-xr-x. 1 root root       0 окт  5 12:39 chkconfig.d
-rw-r--r--. 1 root root    1371 авг 29  2022 chrony.conf
drwxr-xr-x. 1 root root       18 ноя  5 11:17 dconf

```

Figure 4: Поиск файлов определенного расширения

Добавила вывод прошлой команды в новый файл conf.txt с помощью перенаправления ">" (файл создается при выполнении этой команды) (рис. 5).

```
[evdvorkina@evdvorkina ~]$ grep .conf file.txt > conf.txt
[evdvorkina@evdvorkina ~]$ head conf.txt
-rw-r--r--. 1 root root      269 июл 20  2022 anthy-unicode.conf
-rw-r--r--. 1 root root      833 авг 22  2022 appstream.conf
-rw-r--r--. 1 root root       55 дек  7 17:18 asound.conf
drwxr-xr-x. 1 root root       0 окт  5 12:39 chkconfig.d
-rw-r--r--. 1 root root     1371 авг 29  2022 chrony.conf
drwxr-xr-x. 1 root root       18 ноя  5 11:17 dconf
-rw-r--r--. 1 root root    28442 дек  8 16:04 dnsmasq.conf
-rw-r--r--. 1 root root      117 ноя 16 21:00 dracut.conf
drwxr-xr-x. 1 root root       0 ноя 16 21:00 dracut.conf.d
-rw-r--r--. 1 root root       20 фев 24  2022 fprintd.conf
```

Figure 5: Запись в файл

Определяю, какие файлы в домашнем каталоге начинаются с символа “с” с помощью утилиты find, прописываю ей в аргументах домашнюю директорию (тогда вывод относительно корневого каталога, а не домашнего будет), выбираю опцию -name (ищем по имени), и пишу маску, по которой будем искать имя, где * - любое кол-во любых символов, добавляю опцию -print, чтобы мне вывелся результат (рис. 6). Но таким образом я получаю информацию даже о файлах из подкаталогов домашнего каталога.

```
[evdvorkina@evdvorkina ~]$ find ~ -name "с*" -print
/home/evdvorkina/.mozilla/firefox/67pfd4e.default-release/compatibility.ini
/home/evdvorkina/.mozilla/firefox/67pfd4e.default-release/cookies.sqlite
/home/evdvorkina/.mozilla/firefox/67pfd4e.default-release/cert9.db
/home/evdvorkina/.mozilla/firefox/67pfd4e.default-release/storage/permanent/chrome
/home/evdvorkina/.mozilla/firefox/67pfd4e.default-release/storage/default/https++vk.com/cache
/home/evdvorkina/.mozilla/firefox/67pfd4e.default-release/storage/default/https++vk.com/cache/caches.sqlite
/home/evdvorkina/.mozilla/firefox/67pfd4e.default-release/storage/default/https++vk.com/cache/caches.sqlite
```

Figure 6: Поиск файлов, начинающихся с определенного элемента

Второй способ использовать утилиту ls -lR и использовать grep, чтобы найти элементы с первым символом с. Однако этот способ не работает для поиска файлов из подкаталогов каталога (рис. 7).

```
[evdvorkina@evdvorkina ~]$ ls -lR | grep с*
-rw-r--r--. 1 evdvorkina evdvorkina 45801 мар 15 02:13 conf.txt
```

Figure 7: Поиск файлов, начинающихся с определенного элемента

С помощью метода find, чьи опции я расписала ранее, ищу все файлы, начинающиеся с буквы “h” (рис. 8).

```

[evdvorkina@evdvorkina ~]$ sudo find /etc -name "h*" -print
/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
/etc/hp
/etc/hp/hpkip.conf
/etc/sane.d/dll.d/hpaio
/etc/sane.d/hp.conf
/etc/sane.d/hp3900.conf
/etc/sane.d/hp4200.conf
/etc/sane.d/hp5400.conf
/etc/sane.d/hpsj5s.conf
/etc/sane.d/hs2p.conf
/etc/mercurial/hgrc.d

```

Figure 8: Поиск файлов, начинающихся с определенного элемента

Запускаю в фоновом режиме (на это указывает символ &) процесс, который будет записывать в файл logfile (с помощью перенаправления >) файлы, имена которых начинаются с log (рис. 9).

```

[evdvorkina@evdvorkina ~]$ find ~ -name "log*" -print > logfile &
[1] 3822

```

Figure 9: Создание фонового процесса

Проверяю, что файл создан, удаляю его, проверяю, что файл удален (рис. 10).

```

[evdvorkina@evdvorkina ~]$ ls
bin  conf.txt  file.txt  logfile  work  Видео  Документы
[1]+  Завершён      find ~ -name "log*" -print > logfile
[evdvorkina@evdvorkina ~]$ rm logfile
[evdvorkina@evdvorkina ~]$ ls
bin  conf.txt  file.txt  work  Видео  Документы  Загрузки

```

Figure 10: Удаление файла

Запускаю в консоли в фоновом режиме (с помощью символа &) редактор mousepad, потому что редактора gedit у меня, к сожалению, но работают они идентично (рис. 11).

```
[evdvorkina@evdvorkina ~]$ mousepad &  
[2] 3873
```

Figure 11: Создание фонового процесса

С помощью утилиты `ps` определяю идентификатор процесса `mousepad`, его значение 3913 (рис. 12). Также мы можем определить идентификатор с помощью `pgrep`.

```
[evdvorkina@evdvorkina ~]$ ps aux | grep mousepad  
evdvork+ 3913 1.4 1.3 762584 54996 pts/1 S1 02:24 0:00 mousepad  
evdvork+ 3942 0.0 0.0 222180 2340 pts/1 S+ 02:25 0:00 grep --color=auto mousepad  
[evdvorkina@evdvorkina ~]$ pgrep mousepad  
3913  
[evdvorkina@evdvorkina ~]$ ps aux | grep mousepad | grep -v grep  
evdvork+ 3913 0.6 1.2 759292 51832 pts/1 S1 02:24 0:00 mousepad
```

Figure 12: Поиск идентификатора процесса

Прочитала справку команды `kill` (рис. 13).

```
Kill(1)                                User Commands                                Kill(1)  
NAME  
kill - terminate a process  
SYNOPSIS  
kill [-signal|-s signal]-p [-q value] [-a] [--timeout milliseconds signal] [-- pidname ...  
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.
```

Figure 13: Чтение документации

Использую команду `kill` и идентификатор процесса, чтобы его удалить (рис. 14). Заметила, как у меня закрылась программа `mousepad`.

```
[evdvorkina@evdvorkina ~]$ man kill  
[evdvorkina@evdvorkina ~]$ kill 3913  
[evdvorkina@evdvorkina ~]$
```

Figure 14: Удаление процесса

Прочитала документацию про функции `df` и `du` (рис. 15).

```
[evdvorkina@evdvorkina ~]$ man df  
[3]- Завершён mousepad  
[evdvorkina@evdvorkina ~]$ man du  
[evdvorkina@evdvorkina ~]$
```

Figure 15: Чтение документации

Использую утилиту `df` опции `-iv` позволяют увидеть информацию об инодах и сделать вывод читаемым, игнорируя сообщения системы о нем (рис. 16). Эта

утилита нам нужна, чтобы выяснить, сколько свободного места есть у нашей системы.

```
[evdvorkina@evdvorkina ~]$ df -vi
Файловая система  Инодов  ИИспользовано  ИСвободно  ИИспользовано%  Смонтировано в
devtmpfs          1048576      476    1048100          1% /dev
tmpfs              501496        9     501487          1% /dev/shm
tmpfs              819200       876    818324          1% /run
/dev/sda3           0            0         0          - /
tmpfs              1048576       34    1048542          1% /tmp
/dev/sda3           0            0         0          - /home
/dev/sda2           65536        394     65142          1% /boot
tmpfs              100299       125    100174          1% /run/user/1000
[evdvorkina@evdvorkina ~]$
```

Figure 16: Утилита *df*

Использую утилиту *du*. Она нужна чтобы просмотреть, сколько места занимают файлы в определенной директории и найти самые большие из них (рис. 17).

```
[evdvorkina@evdvorkina ~]$ du -a work/study/2022-2023/Операционные\ системы/os-intro/labs/lab03/presentation/
4   work/study/2022-2023/Операционные системы/os-intro/labs/lab03/presentation/Makefile
48  work/study/2022-2023/Операционные системы/os-intro/labs/lab03/presentation/image/kulyabov.jpg
12  work/study/2022-2023/Операционные системы/os-intro/labs/lab03/presentation/image/1.png
16  work/study/2022-2023/Операционные системы/os-intro/labs/lab03/presentation/image/2.png
44  work/study/2022-2023/Операционные системы/os-intro/labs/lab03/presentation/image/3.png
64  work/study/2022-2023/Операционные системы/os-intro/labs/lab03/presentation/image/4.png
20  work/study/2022-2023/Операционные системы/os-intro/labs/lab03/presentation/image/5.png
32  work/study/2022-2023/Операционные системы/os-intro/labs/lab03/presentation/image/6.png
32  work/study/2022-2023/Операционные системы/os-intro/labs/lab03/presentation/image/7.png
268 work/study/2022-2023/Операционные системы/os-intro/labs/lab03/presentation/image
8   work/study/2022-2023/Операционные системы/os-intro/labs/lab03/presentation/presentation.md
4   work/study/2022-2023/Операционные системы/os-intro/labs/lab03/presentation/Л03_Дворкина_презентация.md
112 work/study/2022-2023/Операционные системы/os-intro/labs/lab03/presentation/presentation.pdf
172 work/study/2022-2023/Операционные системы/os-intro/labs/lab03/presentation/Л03_Дворкина_презентация.pdf
804 work/study/2022-2023/Операционные системы/os-intro/labs/lab03/presentation/presentation.html
928 work/study/2022-2023/Операционные системы/os-intro/labs/lab03/presentation/Л03_Дворкина_презентация.html
2300 work/study/2022-2023/Операционные системы/os-intro/labs/lab03/presentation/
```

Figure 17: Утилита *du*

Прочитала документацию о команде *find* (рис. 18).

```
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 specified, . is assumed.

  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.
```

Figure 18: Чтение документации

Вывела имена всех директорий, имеющихя в моем домашнем каталоге, используя аргумент *d* у утилиты *find* опции *-type*, то есть указываю тип файлов, который мне нужен и этот тип Директория (рис. 19). Утилита *-a* позволит увидеть размер всех файлов, а не только диреткорий.

```
./texlive2022/texmf-var/luatex-cache
./texlive2022/texmf-var/luatex-cache/generic
./texlive2022/texmf-var/luatex-cache/generic/names
./texlive2022/texmf-var/luatex-cache/generic/fonts
./texlive2022/texmf-var/luatex-cache/generic/fonts/otl
./bin
[evdvorkina@evdvorkina ~]$ find -type d
```

Figure 19: Название рисунка

5 Выводы

В результате данной лабораторной работы я ознакомилась с инструментами поиска файлов и фильтрации текстовых данных, а также приобрела практические навыки по управлению процессами (и заданиями), по проверке использования диска и по обслуживанию файловых систем.

6 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 ». Этот знак > - перенаправление ввода/вывода, а » - перенаправление в режиме добавления.
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 "p*" -print`

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