

Отчет по выполнению лабораторной работы №8

Операционные системы

София Андреевна Кудякова

Содержание

1	Цель работы	5
2	Задание	6
3	Теоретическое введение	8
4	Выполнение лабораторной работы	9
5	Выводы	15
6	Контрольные вопросы	16
	Список литературы	19

Список иллюстраций

4.1	Запись в файл	9
4.2	Вывод содержимого файла	9
4.3	Добавление данных в файл	9
4.4	Поиск файлов определенного расширения	10
4.5	Запись в файл	10
4.6	Поиск файлов, начинающихся с определенного элемента	11
4.7	Поиск файлов, начинающихся с определенного элемента	11
4.8	Поиск файлов, начинающихся с определенного элемента	11
4.9	Создание фонового процесса	11
4.10	Удаление файла	11
4.11	Создание фонового процесса	12
4.12	Поиск идентификатора процесса	12
4.13	Чтение документации	12
4.14	Удаление процесса	13
4.15	Чтение документации	13
4.16	Утилита df	13
4.17	Утилита du	13
4.18	Чтение документации	14
4.19	Название рисунка	14

Список таблиц

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`. 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 директо-

рий, 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). То же 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. 4.1).

```
sakudyakova@dk2n25 ~ $ ls -lR /etc > file.txt
```

Рис. 4.1: Запись в файл

Проверила, что в файл записались нужные значения с помощью утилиты head, она выводит первые 10 строк файла на экран (рис. 4.2).

```
sakudyakova@dk2n25 ~ $ head file.txt
/etc:
итого 3256
drwxr-xr-x  2 root   root   4096 дек 25 00:12 a2ps
drwxr-xr-x  4 root   root   4096 ноя 20 2022 acpi
-rw-r--r--  1 root   root    47 авг 28 2019 adjtime
-rw-----  1 root   root   494 мая 17 2007 afs.keytab
drwxr-xr-x  3 root   root   4096 июн  4 2019 alsa
drwxr-xr-x  4 root   root   4096 авг 28 2023 apache2
drwxr-xr-x  2 root   root   4096 авг 29 2023 apache2.d
drwxr-xr-x  2 root   root   4096 ноя 26 22:07 apparmor.d
```

Рис. 4.2: Вывод содержимого файла

Добавила в созданный файл имена файлов из домашнего каталога, используя перенаправление “>>” в режиме добавления (рис. 4.3).

```
sakudyakova@dk2n25 ~ $ ls -lR ~/ >> file.txt
```

Рис. 4.3: Добавление данных в файл

Вывела на экран имена всех файлов, имеющих расширение “.conf” с помощью утилиты grep (рис. 4.4).

```
sakudyakova@dk2n25 ~ $ grep .conf file.txt
-rw-r--r-- 1 root root 833 ноя 10 23:28 appstream.conf
-rw-r--r-- 1 root root 30597 ноя 30 15:59 brltty.conf
-rw-r--r-- 1 root root 5961 фев 3 19:08 ca-certificates.conf
-rw-r--r-- 1 root root 756 июл 19 2022 cachefilesd.conf
-rw-r--r-- 1 root root 10883 июл 19 2022 cfg-update.conf
-rw-r--r-- 1 root root 254 фев 8 12:45 chrome-flags.conf
lrwxrwxrwx 1 root root 18 янв 8 19:03 chrony.conf -> chrony/chrony.conf
drwxr-xr-x 2 root root 4096 янв 31 18:06 conf.d
drwxr-xr-x 4 root root 4096 июн 4 2019 dconf
-rw-r--r-- 1 root root 1429 фев 3 19:17 dhcpcd.conf
-rw-r--r-- 1 root root 2314 янв 5 11:21 dispatch-conf.conf
-rw-r--r-- 1 root root 1280 авг 31 2023 dlevna-renderer-service.conf
```

Рис. 4.4: Поиск файлов определенного расширения

Добавила вывод прошлой команды в новый файл conf.txt с помощью перенаправления “>” (файл создается при выполнении этой команды) (рис. 4.5).

```
sakudyakova@dk2n25 ~ $ grep .conf file.txt > conf.txt
sakudyakova@dk2n25 ~ $ head conf.txt
-rw-r--r-- 1 root root 833 ноя 10 23:28 appstream.conf
-rw-r--r-- 1 root root 30597 ноя 30 15:59 brltty.conf
-rw-r--r-- 1 root root 5961 фев 3 19:08 ca-certificates.conf
-rw-r--r-- 1 root root 756 июл 19 2022 cachefilesd.conf
-rw-r--r-- 1 root root 10883 июл 19 2022 cfg-update.conf
-rw-r--r-- 1 root root 254 фев 8 12:45 chrome-flags.conf
lrwxrwxrwx 1 root root 18 янв 8 19:03 chrony.conf -> chrony/chrony.conf
drwxr-xr-x 2 root root 4096 янв 31 18:06 conf.d
drwxr-xr-x 4 root root 4096 июн 4 2019 dconf
-rw-r--r-- 1 root root 1429 фев 3 19:17 dhcpcd.conf
sakudyakova@dk2n25 ~ $
```

Рис. 4.5: 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. 4.6). Но таким образом я получаю информацию даже о файлах из подкаталогов домашнего каталога.

```
sakudyakova@dk2n25 ~ $ find ~ -name "c*" -print
/afs/.dk.sci.pfu.edu.ru/home/s/a/sakudyakova/.mozilla/firefox/twx70tha.default-release/c
lity.ini
/afs/.dk.sci.pfu.edu.ru/home/s/a/sakudyakova/.mozilla/firefox/twx70tha.default-release/c
qlite
/afs/.dk.sci.pfu.edu.ru/home/s/a/sakudyakova/.mozilla/firefox/twx70tha.default-release/c
e
/afs/.dk.sci.pfu.edu.ru/home/s/a/sakudyakova/.mozilla/firefox/twx70tha.default-release/st
ermanent/chrome
/afs/.dk.sci.pfu.edu.ru/home/s/a/sakudyakova/.mozilla/firefox/twx70tha.default-release/st
efault/https+++www.youtube.com*partitionKey=%28https%2Cgoogle.com%29/cache
/afs/.dk.sci.pfu.edu.ru/home/s/a/sakudyakova/.mozilla/firefox/twx70tha.default-release/st
efault/https+++www.youtube.com*partitionKey=%28https%2Cgoogle.com%29/cache/caches.sqlite
/afs/.dk.sci.pfu.edu.ru/home/s/a/sakudyakova/.mozilla/firefox/twx70tha.default-release/st
```

Рис. 4.6: Поиск файлов, начинающихся с определенного элемента

Второй способ использовать утилиту `ls -lR` и использовать `grep`, чтобы найти элементы с первым символом `c`. Однако этот способ не работает для поиска файлов из подкаталогов каталога (рис. 4.7).

```
sakudyakova@dk2n25 ~ $ ls -lR | grep c*
-rw-r--r-- 1 sakudyakova studsci 139689 map 28 10:53 conf.txt
```

Рис. 4.7: Поиск файлов, начинающихся с определенного элемента

С помощью метода `find`, чьи опции я расписала ранее, ищу все файлы, начинающиеся с буквы `h` (рис. 4.8).

```
sakudyakova@dk2n25 ~ $ sudo find /etc -name "h*" -print
```

Рис. 4.8: 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. 4.9).

```
sakudyakova@dk2n25 ~ $ find ~ -name "log*" -print > logfile &
[1] 5252
```

Рис. 4.9: Создание фонового процесса

Проверяю, что файл создан, удаляю его, проверяю, что файл удален (рис. 4.10).

```
sakudyakova@dk2n25 ~ $ ls
'2024-03-28 11-16-11.mp4'  conf.txt  fun      monthly  public  ski.places  Документы  Музыка  Шаблоны
abc1                      file.txt  logfile  my_os    public_html  work      Загрузки  Общедоступные
australia                 file.txt  may      play     reports   Видео      Изображения  'Рабочий стол'
```

```
sakudyakova@dk2n25 ~ $ rm logfile
sakudyakova@dk2n25 ~ $ ls
'2024-03-28 11-16-11.mp4'  conf.txt  fun      my_os    public_html  work      Загрузки  Общедоступные
abc1                      file.txt  may      play     reports   Видео      Изображения  'Рабочий стол'
```

```
sakudyakova@dk2n25 ~ $
```

Рис. 4.10: 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. 4.11).

```
sakudyakova@dk2n25 ~ $ find ~ -name "log*" -print > logfile &
[1] 5252
sakudyakova@dk2n25 ~ $ gedit &
[2] 5340
[1]   Завершён                  find ~ -name "log*" -print > logfile
```

Рис. 4.11: Создание фонового процесса

С помощью утилиты ps определяю идентификатор процесса mousepad, его значение 3913 (рис. 4.12). Также мы можем определить идентификатор с помощью pgrep.

```
sakudyakova@dk2n25 ~ $ ps aux | grep gedit
sakudya+  8616  1.9  1.5 889952 124200 ?        S1   11:32   0:09 /usr/bin/gedit --gapplication-service
sakudya+  9295  0.0  0.0   6932  2340 pts/2    S+   11:41   0:00 grep --colour=auto gedit
sakudyakova@dk2n25 ~ $ pgrep gedit
8616
sakudyakova@dk2n25 ~ $ ps aux | grep gedit | grep -v grep
sakudya+  8616  1.9  1.5 889952 124200 ?        S1   11:32   0:09 /usr/bin/gedit --gapplication-service
sakudyakova@dk2n25 ~ $
```

Рис. 4.12: Поиск идентификатора процесса

Прочитала справку команды kill (рис. 4.13).

```
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>

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

Рис. 4.13: Чтение документации

Использую команду kill и идентификатор процесса, чтобы его удалить (рис. 4.14). Заметила, как у меня закрылась программа mousepad.

```
sakudyakova@dk2n25 ~ $ kill 5340
sakudyakova@dk2n25 ~ $
```

Рис. 4.14: Удаление процесса

Прочитала документацию про функции df и du (рис. 4.15).

```
sakudyakova@dk1n22 ~ $ man df
sakudyakova@dk1n22 ~ $ man du
sakudyakova@dk1n22 ~ $
```

Рис. 4.15: Чтение документации

Используя утилиту df опции -iv позволяют увидеть информацию об инодах и сделать вывод читаемым, игнорируя сообщения системы о нем (рис. 4.16). Эта утилита нам нужна, чтобы выяснить, сколько свободного места есть у нашей системы.

```
sakudyakova@dk2n25 ~ $ df -vi
Файловая система    Инодов  ИИспользовано  ИСвободно  ИИспользовано%  Смонтировано в
/dev/sda8             33447936      1779402      31668534           6% /
devtmpfs              986741        1628        985113           1% /dev
tmpfs                 999906         1        999905           1% /dev/shm
tmpfs                 819200        1899        817301           1% /run
tmpfs                1048576       1233       1047343           1% /tmp
AFS                  2147483647         0  2147483647           0% /afs
tmpfs                199981        188        199793           1% /run/user/5260
[2]+  Завершено      gedit
sakudyakova@dk2n25 ~ $
```

Рис. 4.16: Утилита df

Используя утилиту du. Она нужна чтобы просмотреть, сколько места занимают файлы в определенной директории и найти самые большие из них (рис. 4.17).

```
sakudyakova@dk2n25 ~ $ du -a work/study/2023-2024/"Операционные системы"/os-intro/labs/lab08/presentation/
47  work/study/2023-2024/Операционные системы/os-intro/labs/lab08/presentation/image/kulyabov.jpg
49  work/study/2023-2024/Операционные системы/os-intro/labs/lab08/presentation/image
1   work/study/2023-2024/Операционные системы/os-intro/labs/lab08/presentation/Makefile
8   work/study/2023-2024/Операционные системы/os-intro/labs/lab08/presentation/presentation.md
60  work/study/2023-2024/Операционные системы/os-intro/labs/lab08/presentation/
sakudyakova@dk2n25 ~ $
```

Рис. 4.17: Утилита du

Прочитала документацию о команде find (рис. 4.18).

```
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.

OPTIONS
    The -H, -L and -P options control the treatment of symbolic links. Command-line arguments following these are taken to be names of files or directories to be examined, up to the first argument that begins with '-', or the argument '(' or '!'. That argument and any following arguments are taken to be the expression describing what is to be searched for. If no paths are given, the current directory is used. If no expression is given, the expression -print is used (but you should probably

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

Рис. 4.18: Чтение документации

Вывела имена всех директорий, имеющихсх 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. 4.19). Утилита `-a` позволит увидеть размер всех файлов, а не только диреткорий.

```
.. sakudyakova@dk2n25 ~ $ man find
sakudyakova@dk2n25 ~ $ find -type d
.
./public
./public/public_html
./mozilla
./mozilla/firefox
./mozilla/firefox/twx70tha.default-release
./mozilla/firefox/twx70tha.default-release/security_state
./mozilla/firefox/twx70tha.default-release/storage
./mozilla/firefox/twx70tha.default-release/storage/permanent
./mozilla/firefox/twx70tha.default-release/storage/permanent/chrome
```

Рис. 4.19: Название рисунка

5 Выводы

В ходе 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 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.

Список литературы

Архитектура ЭВМ