

Отчет по выполнению лабораторной работы №8

Операционные системы

Зарицкая Марина Петровна

Содержание

1	Цель работы	5
2	Задание	6
3	Теоретическое введение	7
4	Выполнение лабораторной работы	8
5	Выводы	16
6	Ответы на контрольные вопросы	17

Список иллюстраций

4.1	Запись в файл	8
4.2	Вывод содержимого файла	8
4.3	Добавление данных в файл	9
4.4	Поиск файлов определенного расширения	9
4.5	Запись в файл	10
4.6	Поиск файлов, начинающихся с определенного элемента	10
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	12
4.11	Создание фонового процесса	12
4.12	Поиск идентификатора процесса	12
4.13	Чтение документации	13
4.14	Удаление процесса	13
4.15	Чтение документации	14
4.16	Утилита df	14
4.17	Утилита du	14
4.18	Чтение документации	15
4.19	Название рисунка	15

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 директорий, имеющихся 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. 4.1).

```
mpzarickaya@mpzarickaya:~$ sudo ls -lR /etc > file.txt
[sudo] password for mpzarickaya:
mpzarickaya@mpzarickaya:~$ sudo ls -lR /etc > file.txt
mpzarickaya@mpzarickaya:~$ head file.txt
/etc:
```

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).

```
mpzarickaya@mpzarickaya:~$ head file.txt
/etc:
total 1364
drwxr-xr-x. 1 root root 126 Nov 1 2023 abrt
-rw-r--r--. 1 root root 16 Mar 15 20:48 adjtime
-rw-r--r--. 1 root root 1529 Jul 25 2023 aliases
drwxr-xr-x. 1 root root 70 Jan 29 03:00 alsa
drwxr-xr-x. 1 root root 1416 May 18 14:38 alternatives
drwxr-xr-x. 1 root root 56 Nov 1 2023 anaconda
-rw-r--r--. 1 root root 541 Jul 19 2023 anacrontab
-rw-r--r--. 1 root root 269 Jul 19 2023 anthy-unicode.conf
```

Рис. 4.2: Вывод содержимого файла

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

```
mpzarickaya@mpzarickaya:~$ ls -lR ~/etc >> file.txt
ls: cannot access '/home/mpzarickaya/etc': No such file or directory
mpzarickaya@mpzarickaya:~$ ls -lR ~/ >> file.txt
mpzarickaya@mpzarickaya:~$ grep .conf 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).

```
mpzarickaya@mpzarickaya:~$ grep .conf file.txt
-rw-r--r--. 1 root root 269 Jul 19 2023 anthy-unicode.conf
-rw-r--r--. 1 root root 833 Feb 10 2023 appstream.conf
-rw-r--r--. 1 root root 55 Jan 29 03:00 asound.conf
-rw-r--r--. 1 root root 1372 Dec 5 2023 chrony.conf
drwxr-xr-x. 1 root root 18 Nov 1 2023 dconf
-rw-r--r--. 1 root root 28602 Feb 13 03:00 dnsmasq.conf
-rw-r--r--. 1 root root 117 Nov 16 2023 dracut.conf
drwxr-xr-x. 1 root root 0 Nov 16 2023 dracut.conf.d
-rw-r--r--. 1 root root 20 Feb 24 2022 fprintd.conf
-rw-r--r--. 1 root root 38 Aug 9 2023 fuse.conf
-rw-r--r--. 1 root root 9 Jul 25 2023 host.conf
-rw-r--r--. 1 root root 1001 Nov 9 2021 i3status.conf
-rw-r--r--. 1 root root 5799 Dec 9 2023 idmapd.conf
-rw-r--r--. 1 root root 8979 Mar 15 21:10 kdump.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).

```
mpzarickaya@mpzarickaya:~$ grep .conf file.txt > conf.txt
mpzarickaya@mpzarickaya:~$ head conf.txt
-rw-r--r--. 1 root root 269 Jul 19 2023 anthy-unicode.conf
-rw-r--r--. 1 root root 833 Feb 10 2023 appstream.conf
-rw-r--r--. 1 root root 55 Jan 29 03:00 asound.conf
-rw-r--r--. 1 root root 1372 Dec 5 2023 chrony.conf
drwxr-xr-x. 1 root root 18 Nov 1 2023 dconf
-rw-r--r--. 1 root root 28602 Feb 13 03:00 dnsmasq.conf
-rw-r--r--. 1 root root 117 Nov 16 2023 dracut.conf
drwxr-xr-x. 1 root root 0 Nov 16 2023 dracut.conf.d
-rw-r--r--. 1 root root 20 Feb 24 2022 fprintd.conf
-rw-r--r--. 1 root root 38 Aug 9 2023 fuse.conf
```

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.

```
/home/mpzarickaya/Downloads/devicon-master/icons/cucumber/cucumber-plain.svg
/home/mpzarickaya/Downloads/devicon-master/icons/cypressio
/home/mpzarickaya/Downloads/devicon-master/icons/cypressio/cypressio-line-wordmark.svg
/home/mpzarickaya/Downloads/devicon-master/icons/cypressio/cypressio-line.svg
/home/mpzarickaya/Downloads/devicon-master/icons/cypressio/cypressio-original-wordmark.svg
/home/mpzarickaya/Downloads/devicon-master/icons/cypressio/cypressio-original.svg
/home/mpzarickaya/Downloads/devicon-master/icons/cypressio/cypressio-plain-wordmark.svg
/home/mpzarickaya/Downloads/devicon-master/icons/cypressio/cypressio-plain.svg
/home/mpzarickaya/Downloads/FastQC/Configuration/contaminant_list.txt
/home/mpzarickaya/Downloads/FastQC/org/apache/commons
/home/mpzarickaya/conf.txt
mpzarickaya@mpzarickaya:~$ find ~ -name "c*" -print
```

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).

```
mpzarickaya@mpzarickaya:~$ ls -lr | grep c*
-rw-r--r--. 1 mpzarickaya mpzarickaya 41853 Jun 17 19:55 conf.txt
mpzarickaya@mpzarickaya:~$
```

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).

```
mpzarickaya@mpzarickaya:~$ sudo find /etc -name "h*" -print
[sudo] password for mpzarickaya:
/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/mercurial/hgrc.d
```

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).

```
mpzarickaya@mpzarickaya:~$ find ~ -name "log*" -print > logfile &
[1] 3816
mpzarickaya@mpzarickaya:~$
```

Рис. 4.9: Создание фонового процесса

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

```

mpzarickaya@mpzarickaya:~$ ls
abcl    conf.txt  file.txt  logfile  monthly  play      work      Изображения  Шаблоны
australia Desktop  fun       may      montlhy.00 reports  Видео      Музыка
'~bin'  Downloads letters  memos    montly.00 ski.plases Документы  Общедоступные
bin     feathers  LICENSE   misk     my_os    tutorial  Загрузки   'Рабочий стол'
[1]+  Done                  find ~ -name "log*" -print > logfile
mpzarickaya@mpzarickaya:~$ rm logfile
mpzarickaya@mpzarickaya:~$ ls
abcl    conf.txt  file.txt  may      montlhy.00 reports  Видео      Музыка
australia Desktop  fun       memos    montly.00 ski.plases Документы  Общедоступные
'~bin'  Downloads letters  misk     my_os    tutorial  Загрузки   'Рабочий стол'
bin     feathers  LICENSE   monthly  play      work      Изображения  Шаблоны

```

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).

```

mpzarickaya@mpzarickaya:~$ mousepad &
[1] 3836
mpzarickaya@mpzarickaya:~$

```

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.

```

mpzarickaya@mpzarickaya:~$ ps aux | grep mousepad
mpzaric+  3836  5.4  2.5 848224 51816 pts/0    Sl   20:15   0:02 mousepad
mpzaric+  3880  0.0  0.1 222432  2304 pts/0    S+   20:16   0:00 grep --color=auto mousepad
mpzarickaya@mpzarickaya:~$ pgrep mousepad
3836
mpzarickaya@mpzarickaya:~$ ps aux | grep mousepad | grep -v grep
mpzaric+  3836  2.7  2.5 848224 51816 pts/0    Sl   20:15   0:02 mousepad

```

Рис. 4.12: Поиск идентификатора процесса

Прочитала справку команды kill (рис. 4.13).

```
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. 4.13: Чтение документации

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

```
mpzarickaya@mpzarickaya:~$ man kill
[1]+  Done mousepad
mpzarickaya@mpzarickaya:~$ kill 3836
bash: kill: (3836) - No such process
mpzarickaya@mpzarickaya:~$ mousepad &
[1] 3910
mpzarickaya@mpzarickaya:~$ kill 3910
mpzarickaya@mpzarickaya:~$
```

Рис. 4.14: Удаление процесса

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

```
mpzarickaya@mpzarickaya:~$ man df
[1]+  Done                  mousepad
mpzarickaya@mpzarickaya:~$ man du
mpzarickaya@mpzarickaya:~$
```

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.

```
mpzarickaya@mpzarickaya:~$ df -vi
Filesystem      Inodes IUsed   IFree IUse% Mounted on
/dev/sda3         0      0      0    - /
devtmpfs         246056   523  245533    1% /dev
tmpfs            251256    2  251254    1% /dev/shm
tmpfs            819200   876  818324    1% /run
tmpfs           1048576    33 1048543    1% /tmp
/dev/sda2         65536   395   65141    1% /boot
/dev/sda3         0      0      0    - /home
tmpfs            50251    91   50160    1% /run/user/1000
```

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).

```
mpzarickaya@mpzarickaya:~$ du -a work/study/2022-2023/Операционные системы/os-intro/labs/lab07/report/report
8 work/study/2022-2023/Операционные системы/os-intro/labs/lab07/report/report/bib/cite.bib
8 work/study/2022-2023/Операционные системы/os-intro/labs/lab07/report/report/bib
16 work/study/2022-2023/Операционные системы/os-intro/labs/lab07/report/report/pandoc/csl/gost-r-7-0-5-2008-num
ric.csl
16 work/study/2022-2023/Операционные системы/os-intro/labs/lab07/report/report/pandoc/csl
28 work/study/2022-2023/Операционные системы/os-intro/labs/lab07/report/report/pandoc/filters/pandoc_eqnos.py
28 work/study/2022-2023/Операционные системы/os-intro/labs/lab07/report/report/pandoc/filters/pandoc_fignos.py
16 work/study/2022-2023/Операционные системы/os-intro/labs/lab07/report/report/pandoc/filters/pandoc_secnos.py
28 work/study/2022-2023/Операционные системы/os-intro/labs/lab07/report/report/pandoc/filters/pandoc_tablenos.py
4 work/study/2022-2023/Операционные системы/os-intro/labs/lab07/report/report/pandoc/filters/pandocxnos/__init
__.py
48 work/study/2022-2023/Операционные системы/os-intro/labs/lab07/report/report/pandoc/filters/pandocxnos/core.py
4 work/study/2022-2023/Операционные системы/os-intro/labs/lab07/report/report/pandoc/filters/pandocxnos/main.py
8 work/study/2022-2023/Операционные системы/os-intro/labs/lab07/report/report/pandoc/filters/pandocxnos/pandoc
ttributes.py
4 work/study/2022-2023/Операционные системы/os-intro/labs/lab07/report/report/pandoc/filters/pandocxnos/__pyca
he_/_init_.cpython-312.pyc
44 work/study/2022-2023/Операционные системы/os-intro/labs/lab07/report/report/pandoc/filters/pandocxnos/__pyca
he_/_core.cpython-312.pyc
```

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.

```

Рис. 4.18: Чтение документации

Вывела имена всех директорий, имеющихся 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 директорий.

```

./texlive2023
./texlive2023/texmf-var
./texlive2023/texmf-var/luatex-cache
./texlive2023/texmf-var/luatex-cache/generic
./texlive2023/texmf-var/luatex-cache/generic/names
./texlive2023/texmf-var/luatex-cache/generic/fonts
./texlive2023/texmf-var/luatex-cache/generic/fonts/otl
mpzarickaya@mpzarickaya:~$ find -type d

```

Рис. 4.19: 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.