

# **Отчет по выполнению лабораторной работы №6**

**Операционные системы**

Дзаки Рафли Зайдан

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>5</b>
<b>2</b>	<b>Задание</b>	<b>6</b>
<b>3</b>	<b>Теоретическое введение</b>	<b>8</b>
<b>4</b>	<b>Выполнение лабораторной работы</b>	<b>9</b>
<b>5</b>	<b>Выводы</b>	<b>16</b>
<b>6</b>	<b>Ответы на контрольные вопросы</b>	<b>17</b>

## Список иллюстраций

4.1	Запись в файл . . . . .	9
4.2	Вывод содержимого файла . . . . .	9
4.3	Добавление данных в файл . . . . .	10
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 . . . . .	12
4.9	Создание фонового процесса . . . . .	12
4.10	Удаление файла . . . . .	12
4.11	Создание фонового процесса . . . . .	13
4.12	Поиск идентификатора процесса . . . . .	13
4.13	Чтение документации . . . . .	13
4.14	Удаление процесса . . . . .	14
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, начинавшиеся 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:

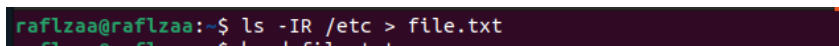
STDIN (0) — стандартный поток ввода (данные, загружаемые в программу). STDOUT (1) — стандартный поток вывода (данные, которые выводит программа). По умолчанию — терминал. STDERR (2) — стандартный поток вывода диагностических и отладочных сообщений (например, сообщениях об ошибках). По умолчанию — терминал.

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

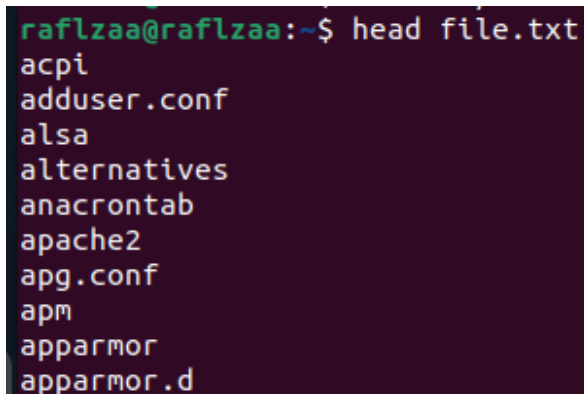
Я вошла в систему под соответствующим именем пользователя, открыла терминал, записала в файл file.txt названия файлов из каталога /etc с помощью перенаправления “>” (и файл создала, и записала в него то, что могло быть выведено ls -lR /etc). В файл я добавила также все файлы из подкаталогов (рис. fig. 4.1).



```
raflzaa@raflzaa:~$ 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. fig. 4.2).



```
raflzaa@raflzaa:~$ head file.txt
acpi
adduser.conf
alsa
alternatives
anacrontab
apache2
apc.conf
apm
apparmor
apparmor.d
```

Рис. 4.2: Вывод содержимого файла

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

```
raflzaa@raflzaa:~$ ls -IR ~/ >> file.txt
```

Рис. 4.3: Добавление данных в файл

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

```
raflzaa@raflzaa:~$ 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
gai.conf
```

Рис. 4.4: Поиск файлов определенного расширения

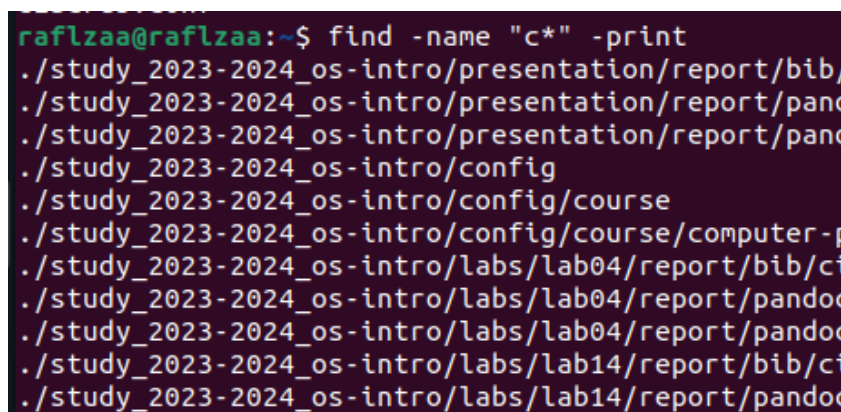
Добавила вывод прошлой команды в новый файл conf.txt с помощью перенаправления “>” (файл создается при выполнении этой команды) (рис. fig. 4.5).

```
raflzaa@raflzaa:~$ grep .conf file.txt > conf.txt
raflzaa@raflzaa:~$ head conf.txt
adduser.conf
apg.conf
appstream.conf
brltty.conf
ca-certificates.conf
ca-certificates.conf.dpkg-old
dconf
debconf.conf
deluser.conf
e2scrub.conf
```

Рис. 4.5: Запись в файл

Определяю, какие файлы в домашнем каталоге начинаются с символа “с” с

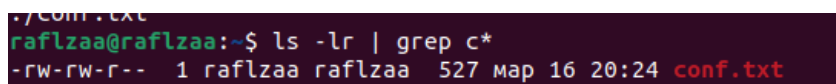
помощью утилиты `find`, прописываю ей в аргументах домашнюю директорию (тогда вывод относительно корневого каталога, а не домашнего будет), выбираю опцию `-name` (ищем по имени), и пишу маску, по которой будем искать имя, где `*` - любое кол-во любых символов, добавляю опцию `-print`, чтобы мне вывелся результат (рис. fig. 4.6). Но таким образом я получаю информацию даже о файлах из подкаталогов домашнего каталога.



```
raflzaa@raflzaa:~$ find -name "c*" -print
./study_2023-2024_os-intro/presentation/report/bib/
./study_2023-2024_os-intro/presentation/report/pandoc
./study_2023-2024_os-intro/presentation/report/pandoc
./study_2023-2024_os-intro/config
./study_2023-2024_os-intro/config/course
./study_2023-2024_os-intro/config/course/computer-g
./study_2023-2024_os-intro/labs/lab04/report/bib/ci
./study_2023-2024_os-intro/labs/lab04/report/pandoc
./study_2023-2024_os-intro/labs/lab04/report/pandoc
./study_2023-2024_os-intro/labs/lab14/report/bib/ci
./study_2023-2024_os-intro/labs/lab14/report/pandoc
```

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. fig. 4.7).



```
raflzaa@raflzaa:~$ ls -lr | grep c*
-rw-rw-r-- 1 raflzaa raflzaa 527 map 16 20:24 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. fig. 4.8).

```

raflzaa@raflzaa:~$ sudo find /etc -name "h*" -print
[sudo] password for raflzaa:
/etc/apparmor.d/tunables/home
/etc/apparmor.d/tunables/home.d
/etc/apparmor.d/abstractions/hosts_access
/etc/brltty/Input/ht
/etc/brltty/Input/hd
/etc/brltty/Input/bm/horizontal.kti
/etc/brltty/Input/hw
/etc/brltty/Input/hm
/etc/brltty/Contraction/ha.ctb
/etc/brltty/Text/hy.ttb
/etc/brltty/Text/hi.ttb
/etc/brltty/Text/hr.ttb
/etc/brltty/Text/he.ttb
/etc/brltty/Text/hu.ttb
/etc/hdparm.conf
/etc/sane.d/hp4200.conf
/etc/sane.d/hp5400.conf
/etc/sane.d/hpsj5s.conf
/etc/sane.d/hp.conf
/etc/sane.d/hs2p.conf
/etc/sane.d/dll.d/hplip
/etc/sane.d/hp3900.conf
/etc/initramfs-tools/hooks

```

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. fig. 4.9).

```

raflzaa@raflzaa:~$ find ~ -name "log*" -print > logfile &
[1] 9021
raflzaa@raflzaa:~$ ls

```

Рис. 4.9: Создание фонового процесса

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

```

raflzaa@raflzaa:~$ find ~ -name "log*" -print > logfile &
[1] 9021
raflzaa@raflzaa:~$ ls

```

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. fig. 4.11).

```
raflzaa@raflzaa:~$ mousepad &  
[1] 9038
```

Рис. 4.11: Создание фонового процесса

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

```
raflzaa@raflzaa:~$ ps aux | grep mousepad  
raflzaa  9055  0.0  0.0  9536 2688 pts/4    S+   20:28   0:00 grep --color=auto mousepad
```

Рис. 4.12: Поиск идентификатора процесса

Прочитала справку команды kill (рис. fig. 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>  
        Specify the signal to be sent. The signal can be  
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. fig. 4.14). Заметила, как у меня закрылась программа mousepad.

```

raflzaa@raflzaa:~$ man kill
raflzaa@raflzaa:~$ kill 9038

```

Рис. 4.14: Удаление процесса

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

```

raflzaa@raflzaa:~$ man df
raflzaa@raflzaa:~$ man du

```

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. fig. 4.16). Эта утилита нам нужна, чтобы выяснить, сколько свободного места есть у нашей системы.

```

raflzaa@raflzaa:~$ df -vi

```

Filesystem	Inodes	IUsed	IFree	IUse%	Mounted on
tmpfs	1215928	1021	1214907	1%	/run
/dev/sda3	2011296	566885	1444411	29%	/
tmpfs	1215928	1	1215927	1%	/dev/shm
tmpfs	1215928	4	1215924	1%	/run/lock
/dev/sda2	0	0	0	-	/boot/efi
tmpfs	243185	171	243014	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. fig. 4.17).

```

raflzaa@raflzaa:~$ du -a work/study/2023-2024//home/Операционные\ системы/os-intro/labs/lab03/presentation

```

Рис. 4.17: Утилита du

Прочитала документацию о команде find (рис. fig. 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 start-
    ing-point by evaluating the given expression from left to
    right, according to the rules of precedence (see section OP-
    ERATORS), 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 direc-
    tories that are writable by other users), you should read
    the 'Security Considerations' chapter of the findutils docu-
    mentation, 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
    Manual page find(1) line 1 (press h for help or q to quit)

```

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. fig. 4.19). Утилита **-a** позволит увидеть размер всех файлов, а не только диреткорий.

```

SEARCHING

/pattern      * Search forward for (N-th) matching line.
?pattern      * Search backward for (N-th) matching line.
n             * Repeat previous search (for N-th occurrence).
N             * Repeat previous search in reverse direction.
ESC-n         * Repeat previous search, spanning files.
ESC-N         * Repeat previous search, reverse dir. & spanning files.
ESC-u         Undo (toggle) search highlighting.
ESC-U         Clear search highlighting.
&pattern      * Display only matching lines.
-----
A search pattern may begin with one or more of:
^N or !       Search for NON-matching lines.
^E or *       Search multiple files (pass thru END OF FILE).
^F or @       Start search at FIRST file (for /) or last file (for ?).
^K           Highlight matches, but don't move (KEEP position).
^R           Don't use REGULAR EXPRESSIONS.
^W           WRAP search if no match found.
-----

```

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.