

# **Отчет по лабораторной работе №8**

**Операционные системы**

Ермакова Анастасия Алексеевна

# 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>7</b>
<b>4</b>	<b>Выполнение лабораторной работы</b>	<b>8</b>
<b>5</b>	<b>Выводы</b>	<b>15</b>
<b>6</b>	<b>Ответы на контрольные вопросы</b>	<b>16</b>
<b>7</b>	<b>Список литературы</b>	<b>18</b>

# Список иллюстраций

4.1	Вход в систему . . . . .	8
4.2	Запись в файл . . . . .	8
4.3	Вывод содержимого файла . . . . .	9
4.4	Добавление данных в файл . . . . .	9
4.5	Поиск файлов определенного расширения . . . . .	9
4.6	Запись в файл . . . . .	10
4.7	Поиск файлов, начинающихся с определенного элемента . . . . .	10
4.8	Поиск файлов, начинающихся с определенного элемента . . . . .	10
4.9	Поиск файлов, начинающихся с определенного элемента . . . . .	11
4.10	Создание фонового процесса . . . . .	11
4.11	Удаление файла . . . . .	11
4.12	Создание фонового процесса . . . . .	12
4.13	Поиск идентификатора процесса . . . . .	12
4.14	Чтение документации . . . . .	12
4.15	Удаление процесса . . . . .	13
4.16	Чтение документации . . . . .	13
4.17	Утилита df . . . . .	13
4.18	Утилита du . . . . .	13
4.19	Чтение документации . . . . .	14
4.20	Использование команды find . . . . .	14

## **Список таблиц**

# 1 Цель работы

Цель данной лабораторной работы - ознакомление с инструментами поиска файлов и фильтрации текстовых данных. Приобретение практических навыков: по управлению процессами (и заданиями), по проверке использования диска и 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

Вошла в систему под соответствующим именем пользователя (рис. 4.1).

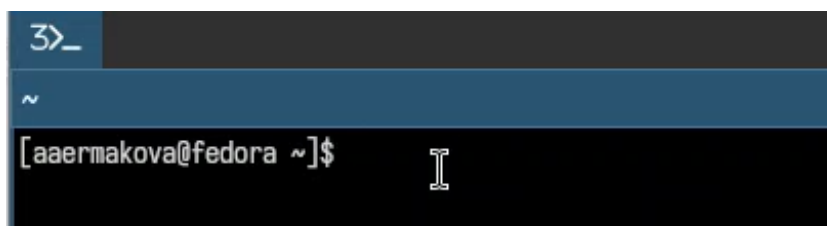


Рис. 4.1: Вход в систему

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. 4.2).

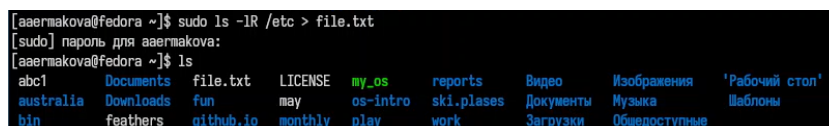


Рис. 4.2: Запись в файл

Проверила, что в файл записались нужные значения с помощью утилиты head, она выводит первые 10 строк файла на экран (рис. 4.3).



```
[aaermakova@fedora ~]$ head file.txt
/etc:
итого 1360
drwxr-xr-x. 1 root root    126 ноя 20 03:00 abrt
-rw-r--r--. 1 root root     16 мар  3 15:04 adjtime
-rw-r--r--. 1 root root  1529 дек 27 03:00 aliases
drwxr-xr-x. 1 root root     70 ноя 14 03:00 alsa
drwxr-xr-x. 1 root root   1400 мар  8 14:05 alternatives
drwxr-xr-x. 1 root root     56 окт 24 17:50 anaconda
-rw-r--r--. 1 root root    541 июл 17 2024 anacrontab
-rw-r--r--. 1 root root     55 ноя 15 03:00 asound.conf
[aaermakova@fedora ~]$
```

Рис. 4.3: Вывод содержимого файла

Добавила в созданный файл имена файлов из домашнего каталога, используя перенаправление “>” в режиме добавления (рис. 4.4).

```
[aaermakova@fedora ~]$ ls -lR ~/ >> file.txt
```

Рис. 4.4: Добавление данных в файл

Вывела на экран имена всех файлов, имеющих расширение “.conf” с помощью утилиты grep (рис. 4.5).

```
[aaermakova@fedora ~]$ grep .conf file.txt
-rw-r--r--. 1 root root    55 ноя 15 03:00 asound.conf
-rw-r--r--. 1 root root  1381 окт  8 03:00 chrony.conf
drwxr-xr-x. 1 root root    18 окт 24 17:49 dconf
-rw-r--r--. 1 root root 28602 дек 27 03:00 dnsmasq.conf
-rw-r--r--. 1 root root   117 ноя 27 03:00 dracut.conf
drwxr-xr-x. 1 root root     0 ноя 27 03:00 dracut.conf.d
-rw-r--r--. 1 root root    20 сен  5 2024 fprintd.conf
```

Рис. 4.5: Поиск файлов определенного расширения

Добавила вывод прошлой команды в новый файл conf.txt с помощью перенаправления “>” (файл создается при выполнении этой команды) (рис. 4.6).

```
[aaermakova@fedora ~]$ grep .conf file.txt > conf.txt
[aaermakova@fedora ~]$ head conf.txt
-rw-r--r--. 1 root root    55 ноя 15 03:00 asound.conf
-rw-r--r--. 1 root root  1381 окт  8 03:00 chrony.conf
drwxr-xr-x. 1 root root   18 окт 24 17:49 dconf
-rw-r--r--. 1 root root 28602 дек 27 03:00 dnsmasq.conf
-rw-r--r--. 1 root root   117 ноя 27 03:00 dracut.conf
drwxr-xr-x. 1 root root    0 ноя 27 03:00 dracut.conf.d
-rw-r--r--. 1 root root   20 сен  5 2024 fprintd.conf
-rw-r--r--. 1 root root   38 июл 17 2024 fuse.conf
-rw-r--r--. 1 root root    9 дек 27 03:00 host.conf
-rw-r--r--. 1 root root  5799 фев  4 03:00 idmapd.conf
```

Рис. 4.6: 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. Но таким образом я получаю информацию даже о файлах из подкаталогов домашнего каталога. (рис. 4.7).

```
[aaermakova@fedora ~]$ find ~ -name "c*" -print
/home/aaermakova/.mozilla/firefox/zv53mqsk.default-release/crashes
/home/aaermakova/.mozilla/firefox/zv53mqsk.default-release/compatibility.ini
/home/aaermakova/.mozilla/firefox/zv53mqsk.default-release/cookies.sqlite
/home/aaermakova/.mozilla/firefox/zv53mqsk.default-release/cert9.db
/home/aaermakova/.mozilla/firefox/zv53mqsk.default-release/security_state/crlite.filter
/home/aaermakova/.mozilla/firefox/zv53mqsk.default-release/security_state/crlite.coverage
/home/aaermakova/.mozilla/firefox/zv53mqsk.default-release/security_state/crlite.enrollment
/home/aaermakova/.mozilla/firefox/zv53mqsk.default-release/storage/permanent/chrome
```

Рис. 4.7: 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. 4.8).

```
[aaermakova@fedora ~]$ ls -lR | grep c*
-rw-r--r--. 1 aaermakova aaermakova 43704 апр  3 15:42 conf.txt
```

Рис. 4.8: Поиск файлов, начинающихся с определенного элемента

С помощью метода find, чьи опции я расписала ранее, ищу все файлы, начинающиеся с буквы “h” (рис. 4.9).

```
[aaermakova@fedora ~]$ sudo find /etc -name "h*" -print
[sudo] пароль для aaermakova:
/etc/avahi/hosts
/etc/firewalld/helpers
/etc/libibverbs.d/hfi1verbs.driver
/etc/libibverbs.d/hns.driver
/etc/nvme/hostnqn
/etc/nvme/hostid
/etc/udev/hwdb.d
/etc/udev/hwdb.bin
/etc/host.conf
```

Рис. 4.9: 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. 4.10).

```
[aaermakova@fedora ~]$ find ~ -name "log*" -print > logfile &
[1] 5943
```

Рис. 4.10: Создание фонового процесса

Проверяю, что файл создан, удаляю его, проверяю, что файл удален (рис. 4.11).

```
[aaermakova@fedora ~]$ ls
abc1  conf.txt  feathers  github.io  may  os-intro  ski.plases  Документы  Музыка  Шаблоны
australia  Documents  file.txt  LICENSE  monthly  play  work  Загрузки  Общедоступные
bin  Downloads  fun  logfile  my_os  reports  Видео  Изображения  'Рабочий стол'
[1]+  Завершён      find ~ -name "log*" -print > logfile
[aaermakova@fedora ~]$ rm logfile
[aaermakova@fedora ~]$ ls
abc1  conf.txt  feathers  github.io  monthly  play  work  Загрузки  Общедоступные
australia  Documents  file.txt  LICENSE  my_os  reports  Видео  Изображения  'Рабочий стол'
bin  Downloads  fun  may  os-intro  ski.plases  Документы  Музыка  Шаблоны
```

Рис. 4.11: Удаление файла

Запускаю в консоли в фоновом режиме (с помощью символа &) редактор mousepad (рис. 4.12).

```
[aaermakova@fedora ~]$ mousepad &  
[1] 6942
```

Рис. 4.12: Создание фонового процесса

С помощью утилиты ps определяю идентификатор процесса mousepad, его значение 6942. Также мы можем определить идентификатор с помощью pgrep. (рис. 4.13).

```
[aaermakova@fedora ~]$ ps aux | grep mousepad  
aaermak+ 6942 1.8 1.3 767984 55352 pts/0 S1 15:50 0:00 mousepad  
aaermak+ 7088 0.0 0.0 238496 2444 pts/0 S+ 15:51 0:00 grep --color=auto mousepad  
[aaermakova@fedora ~]$ pgrep mousepad  
6942  
[aaermakova@fedora ~]$ ps aux | grep mousepad | grep -v grep  
aaermak+ 6942 0.9 1.3 767984 55488 pts/0 S1 15:50 0:00 mousepad
```

Рис. 4.13: Поиск идентификатора процесса

Прочитала справку команды kill (рис. 4.14).

```
man kill  
KILL(1) User Commands  
  
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, used in preference to the KILL signal (number 9), since a process may install a handler for the TERM signal to perform steps before terminating in an orderly fashion. If a process does not terminate after a TERM signal has been sent, it may be used; be aware that the latter signal cannot be caught, and so does not give the target process a chance to 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 --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.  
  
ARGUMENTS
```

Рис. 4.14: Чтение документации

Использую команду kill и идентификатор процесса, чтобы его удалить (рис. 4.15).

```
[aaermakova@fedora ~]$ kill 6942
```

Рис. 4.15: Удаление процесса

Прочитала документацию про функции df и du (рис. 4.16).

```
[aaermakova@fedora ~]$ man df
[1]+  Завершён      mousepad
[aaermakova@fedora ~]$ man du
[aaermakova@fedora ~]$
```

Рис. 4.16: Чтение документации

Используя утилиту df опции -iv позволяют увидеть информацию об инодах и сделать вывод читаемым, игнорируя сообщения системы о нем. Эта утилита нам нужна, чтобы выяснить, сколько свободного места есть у нашей системы. (рис. 4.17).

```
[aaermakova@fedora ~]$ df -vi
Файловая система  Инодов  Использовано  ISвободно  IИспользовано%  Смонтировано в
/dev/sda3          0         0         0          1% /dev
devtmpfs          495506     523    494983         1% /dev
tmpfs             499868      9    499859         1% /dev/shm
tmpfs             819200     886    818314         1% /run
tmpfs             1024        2     1022         1% /run/credentials/systemd-network-generator.service
tmpfs             1024        2     1022         1% /run/credentials/systemd-journald.service
tmpfs             1024        2     1022         1% /run/credentials/systemd-udev-load-credentials.service
tmpfs             1024        2     1022         1% /run/credentials/systemd-tmpfiles-setup-dev-early.service
tmpfs             1024        2     1022         1% /run/credentials/systemd-sysctl.service
tmpfs             1024        2     1022         1% /run/credentials/systemd-tmpfiles-setup-dev.service
tmpfs            1048576     34   1048542         1% /tmp
/dev/sda3          0         0         0          1% /home
/dev/sda2          65536     398    65138         1% /boot
tmpfs             1024        2     1022         1% /run/credentials/systemd-vconsole-setup.service
tmpfs             1024        2     1022         1% /run/credentials/systemd-tmpfiles-setup.service
tmpfs             1024        2     1022         1% /run/credentials/systemd-resolved.service
tmpfs            99973      106    99867         1% /run/user/1000
```

Рис. 4.17: Утилита df

Используя утилиту du. Она нужна чтобы просмотреть, сколько места занимают файлы в определенной директории и найти самые большие из них (рис. 4.18).

```
[aaermakova@fedora ~]$ du -a work/study/2024-2025/"Операционные системы"/os-intro/labs/lab07/report/
4  work/study/2024-2025/Операционные системы/os-intro/labs/lab07/report/Makefile
8  work/study/2024-2025/Операционные системы/os-intro/labs/lab07/report/bib/cite.bib
8  work/study/2024-2025/Операционные системы/os-intro/labs/lab07/report/bib
316 work/study/2024-2025/Операционные системы/os-intro/labs/lab07/report/image/1.png
284 work/study/2024-2025/Операционные системы/os-intro/labs/lab07/report/image/2.png
136 work/study/2024-2025/Операционные системы/os-intro/labs/lab07/report/image/3.png
```

Рис. 4.18: Утилита du

Прочитала документацию о команде find (рис. 4.19).

```
man find
FIND(1)                                General Commands Manual

NAME
    find - search for files in a directory hierarchy

SYNOPSIS
    find [-H] [-L] [-P] [-D debugopts] [-O level] [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, testing each file in the tree against the given expression from left to right, according to the rules of precedence (see section OPERATORS), until the expression is found to be true.  If 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 expression 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 for files writable by other users), you should read the 'Security Considerations' chapter of the findutils documentation, which is included in the findutils package.  That document also includes a lot more detail and discussion than this manual page.

OPTIONS
```

Рис. 4.19: 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. Утилита `-a` позволит увидеть размер всех файлов, а не только директорий. (рис. 4.20).

```
[aaermakova@fedora ~]$ find -type d
.
./.mozilla
./.mozilla/extensions
./.mozilla/extensions/{ec8030f7-c20a-464f-9b0e-13a3a9e97384}
./.mozilla/plugins
./.mozilla/firefox
./.mozilla/firefox/Crash Reports
./.mozilla/firefox/Crash Reports/events
./.mozilla/firefox/Pending Pings
./.mozilla/firefox/zv53mqsk.default-release
./.mozilla/firefox/zv53mqsk.default-release/minidumps
./.mozilla/firefox/zv53mqsk.default-release/crashes
./.mozilla/firefox/zv53mqsk.default-release/crashes/events
./.mozilla/firefox/zv53mqsk.default-release/security state
```

Рис. 4.20: Использование команды find

## **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

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.

Объясните разницу между операцией `>` и `>>`. Этот знак `>` - перенаправление ввода/вывода, а `>>` - перенаправление в режиме добавления.

Что такое конвейер? Конвейер (`pipe`) служит для объединения простых команд или утилит в цепочки, в которых результат работы предыдущей команды передается последующей.

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.

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.

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.

Найдите информацию об утилитах `top` и `htop`. Каковы их функции?

Команда `htop` похожа на команду `top` по выполняемой функции: они обе по-



казывают 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.

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`

Можно ли по контексту (содержанию) найти файл? Если да, то как? `find / -type f -exec grep -H 'текстДляПоиска' {} ;`

Как определить объем свободной памяти на жёстком диске? С помощью команды `df -h`.

Как определить объем вашего домашнего каталога? С помощью команды `du -s`.

Как удалить зависший процесс? С помощью команды `kill %номер задачи`.

## **7 Список литературы**