

Лабораторная работа №6

Дисциплина: Операционные системы

Алиева Милена Арифовна

Содержание

1	Цель работы	5
2	Задание	6
3	Теоретическое введение	7
4	Выполнение лабораторной работы	8
5	Выводы	16

Список иллюстраций

4.1	Запись в файл file.txt	8
4.2	Запись в файл file.txt	8
4.3	Вывод файлов с расширением .conf	8
4.4	Копирование файла io.h	9
4.5	Поиск файлов, чьи имена начинаются с символа с	9
4.6	Поиск файлов, чьи имена начинаются с символа с	9
4.7	Вывод файлов, чьи имена начинаются с символа h	9
4.8	Работа с файлом logfile	10
4.9	Запуск редактора gedit и определение идентификатора	10
4.10	Определение идентификатора	10
4.11	Завершение процесса gedit	11
4.12	Справка по команде df	11
4.13	Справка по команде du	11
4.14	df с опцией -vi	12
4.15	Команда du	12
4.16	Вывод имен всех директорий в домашнем каталоге	12

Список таблиц

1 Цель 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, имеющих в вашем домашнем каталоге

3 Теоретическое введение

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. Большинство используемых в консоли команд и программ записывают результаты своей работы в стандартный поток вывода `stdout`. Например, команда `ls` выводит в стандартный поток вывода (консоль) список файлов в текущей директории. Потоки вывода и ввода можно перенаправлять на другие файлы или устройства. Проще всего это делается с помощью символов `>`, `>>`, `<`, `<<`

Конвейер (`pipe`) служит для объединения простых команд или утилит в цепочки, в которых результат работы предыдущей команды передаётся последующей.

Команда `find` используется для поиска и отображения на экран имён файлов, соответствующих заданной строке символов.

Команда `df` показывает размер каждого смонтированного раздела диска.

Команда `du` показывает число килобайт, используемое каждым файлом или каталогом.

Любую выполняющуюся в консоли команду или внешнюю программу можно запустить в фоновом режиме. Для этого следует в конце имени команды указать знак амперсанда `&`.

4 Выполнение лабораторной работы

1. Записали в файл file.txt названия файлов, содержащихся в каталоге /etc. С помощью команды cat проверили корректность выполнения (рис. [4.1])

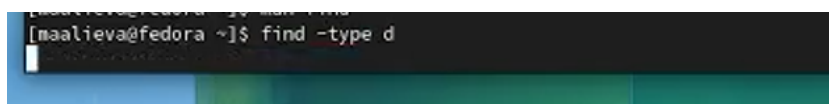


Рис. 4.1: Запись в файл file.txt

Затем дописали в этот же файл названия файлов, содержащихся в домашнем каталоге (рис. [4.2])

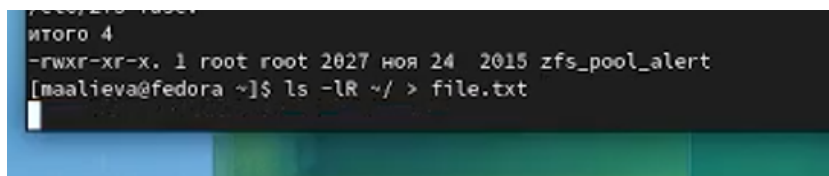


Рис. 4.2: Запись в файл file.txt

2. Вывели имена всех файлов из file.txt, имеющих расширение .conf (рис. [4.3])

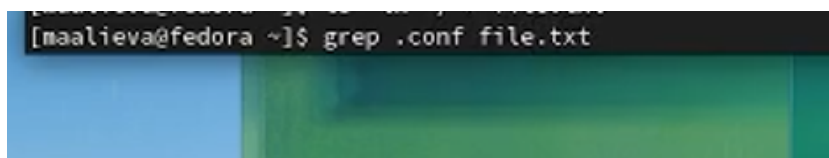
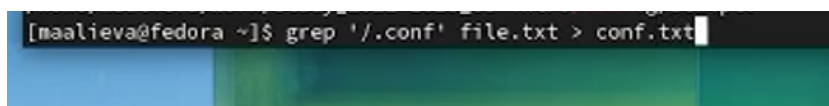


Рис. 4.3: Вывод файлов с расширением .conf

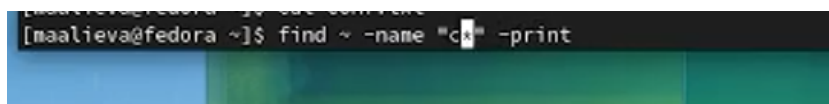
После чего записали их в новый текстовый файл conf.txt (рис. [4.4])



```
[maalieva@fedora ~]$ grep '/.conf' file.txt > conf.txt
```

Рис. 4.4: Копирование файла io.h

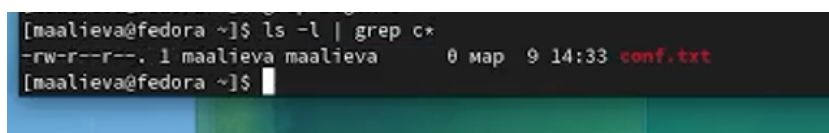
3. Определили, какие файлы в домашнем каталоге имеют имена, начинающиеся с символа c. Сделали это двумя разными способами, представленными на двух скриншотах ниже. Первый способ: (рис. [4.5])



```
[maalieva@fedora ~]$ find ~ -name "c*" -print
```

Рис. 4.5: Поиск файлов, чьи имена начинаются с символа c

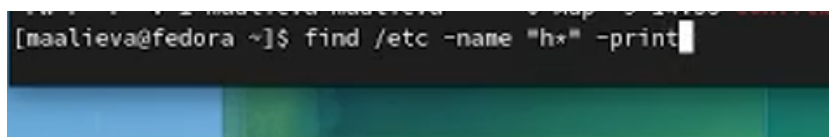
Второй способ: (рис. [4.6])



```
[maalieva@fedora ~]$ ls -l | grep c*  
-rw-r--r--. 1 maalieva maalieva    0 map  9 14:33 conf.txt  
[maalieva@fedora ~]$
```

Рис. 4.6: Поиск файлов, чьи имена начинаются с символа c

4. Вывели на экран имена файлов из каталога /etc, начинающиеся с символа h (рис. [4.7])



```
[maalieva@fedora ~]$ find /etc -name "h*" -print
```

Рис. 4.7: Вывод файлов, чьи имена начинаются с символа h

5. Запустили в фоновом режиме процесс, который будет записывать в файл ~/logfile файлы, имена которых начинаются с log. Затем удалили файл ~/logfile (рис. [4.8])

```
[maalieva@fedora ~]$ find ~/ -name "log*" -print > logfile &
[1] 2572
[maalieva@fedora ~]$ rm logfile
[1]+  Завершён      find ~/ -name "log*" -print > logfile
[maalieva@fedora ~]$
```

Рис. 4.8: Работа с файлом logfile

6. Запустили из консоли в фоновом режиме редактор gedit. Также используя команду ps, конвейер и фильтр grep определили идентификатор процесса (рис. [4.9])

```
[maalieva@fedora ~]$ gedit &
[1] 2802
[maalieva@fedora ~]$ ps aux | grep gedit
maalieva  2802  4.0  3.8 787528 60936 pts/1    Sl   14:59   0:00 gedit
maalieva  2827  0.0  0.1 222192  2240 pts/1    S+   14:59   0:00 grep --color=auto gedit
[maalieva@fedora ~]$
```

Рис. 4.9: Запуск редактора gedit и определение идентификатора

Второй способ определить идентификатор процесса: (рис. [4.10])

```
[maalieva@fedora ~]$ pgrep gedit
2802
[maalieva@fedora ~]$
```

Рис. 4.10: Определение идентификатора

7. После прочтения man kill использовали эту команду и идентификатор процесса gedit для завершения процесса gedit (рис. [4.11])

```
2802
[maalieva@fedora ~]$ kill 2802
[1]+  Завершено      gedit
[maalieva@fedora ~]$
```

Рис. 4.11: Завершение процесса gedit

8. Прочитали man по командам df и du. Команда df: (рис. [4.12])

```
DF(1)                                User Commands                                DF(1)
NAME
df - report file system space usage

SYNOPSIS
df [OPTION]... [FILE]...

DESCRIPTION
This manual page documents the GNU version of df. df displays the amount of space available on the file system containing each file name argument. If no file name is given, the space available on all currently mounted file systems is shown. Space is shown in 1K blocks by default, unless the environment variable POSIXLY_CORRECT is set, in which case 512-byte blocks are used.

If an argument is the absolute file name of a device node containing a mounted file system, df shows the space available on that file system rather than on the file system containing the device node. This version of df cannot show the space available on unmounted file systems, because on most kinds of systems doing so requires very nonportable intimate knowledge of file system structures.

OPTIONS
Show information about the file system on which each FILE resides, or all file systems by default.

Mandatory arguments to long options are mandatory for short options too.

-a, --all
    include pseudo, duplicate, inaccessible file systems

-B, --block-size=SIZE
    scale sizes by SIZE before printing them; e.g., '-BM' prints sizes in units of 1,048,576 bytes; see SIZE format below

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

Рис. 4.12: Справка по команде df

Команда du: (рис. [4.13])

```
DU(1)                                User Commands                                DU(1)
NAME
du - estimate file space usage

SYNOPSIS
du [OPTION]... [FILE]...
du [OPTION]... --files0-from=F

DESCRIPTION
Summarize device usage of the set of FILES, recursively for directories.
Mandatory arguments to long options are mandatory for short options too.

-B, --null
    end each output line with NUL, not newline

-a, --all
    write counts for all files, not just directories

--apparent-size
    print apparent sizes rather than device usage; although the apparent size is usually smaller, it may be larger due to holes in ('sparse') files, internal fragmentation, indirect blocks, and the like

-B, --block-size=SIZE
    scale sizes by SIZE before printing them; e.g., '-BM' prints sizes in units of 1,048,576 bytes; see SIZE format below

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

Рис. 4.13: Справка по команде du

9. Выполнили команду df с опцией -vi (рис. [4.14])

```

[maalieva@fedora ~]$ df -vi
Файловая система  Инодов  ИИспользовано  ИСвободно  ИИспользовано%  Смонтировано в
devtmpfs          1048576      459      1048117          1% /dev
tmpfs             195611        1      195610          1% /dev/shm
tmpfs             819200       872      818328          1% /run
/dev/sda2          0            0            0           - /
tmpfs             1048576       43      1048533          1% /tmp
/dev/sda1          65536        456      65080          1% /boot
/dev/sda2          0            0            0           - /home
tmpfs             39122        154      38968          1% /run/user/1000
[maalieva@fedora ~]$

```

Рис. 4.14: df с опцией -vi

10. Выполнили команду du (рис. [4.15])

```

2624 /home/maalieva/.texlive2022/texmf-var/luatex-cache/generic/fonts/otl/firasans-lightitalic.lua
1760 /home/maalieva/.texlive2022/texmf-var/luatex-cache/generic/fonts/otl/firasans-lightitalic.luc
2736 /home/maalieva/.texlive2022/texmf-var/luatex-cache/generic/fonts/otl/firasans-regular.lua
1716 /home/maalieva/.texlive2022/texmf-var/luatex-cache/generic/fonts/otl/firasans-regular.luc
2624 /home/maalieva/.texlive2022/texmf-var/luatex-cache/generic/fonts/otl/firasans-italic.lua
1680 /home/maalieva/.texlive2022/texmf-var/luatex-cache/generic/fonts/otl/firasans-italic.luc
292 /home/maalieva/.texlive2022/texmf-var/luatex-cache/generic/fonts/otl/firamon-regular.lua
180 /home/maalieva/.texlive2022/texmf-var/luatex-cache/generic/fonts/otl/firamon-regular.luc
292 /home/maalieva/.texlive2022/texmf-var/luatex-cache/generic/fonts/otl/firamon-bold.lua
180 /home/maalieva/.texlive2022/texmf-var/luatex-cache/generic/fonts/otl/firamon-bold.luc
292 /home/maalieva/.texlive2022/texmf-var/luatex-cache/generic/fonts/otl/firamon-medium.lua
180 /home/maalieva/.texlive2022/texmf-var/luatex-cache/generic/fonts/otl/firamon-medium.luc
292 /home/maalieva/.texlive2022/texmf-var/luatex-cache/generic/fonts/otl/firamon-boldoblique.lua
180 /home/maalieva/.texlive2022/texmf-var/luatex-cache/generic/fonts/otl/firamon-boldoblique.luc
292 /home/maalieva/.texlive2022/texmf-var/luatex-cache/generic/fonts/otl/firamon-oblique.lua
180 /home/maalieva/.texlive2022/texmf-var/luatex-cache/generic/fonts/otl/firamon-oblique.luc
27144 /home/maalieva/.texlive2022/texmf-var/luatex-cache/generic/fonts/otl
27144 /home/maalieva/.texlive2022/texmf-var/luatex-cache/generic/fonts
27672 /home/maalieva/.texlive2022/texmf-var/luatex-cache/generic
27672 /home/maalieva/.texlive2022/texmf-var/luatex-cache
27672 /home/maalieva/.texlive2022/texmf-var
27672 /home/maalieva/.texlive2022
4 /home/maalieva/.gitconfig
136 /home/maalieva/.file.txt
0 /home/maalieva/.conf.txt
4 /home/maalieva/.lessht
559516 /home/maalieva/
[maalieva@fedora ~]$

```

Рис. 4.15: Команда du

11. Воспользовавшись справкой команды find, вывели имена всех директорий, имеющих в домашнем каталоге (рис. [4.16])

```

[maalieva@fedora ~]$ find -type d

```

Рис. 4.16: Вывод имен всех директорий в домашнем каталоге

#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 `>>`.

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?

PID - (`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. 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 'текст для поиска' {} ;
```

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}`.

5 Выводы

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