

Отчёт по лабораторной работе №8

Операционные системы

Лисенков Е.Р.

Содержание

Цель работы	1
Задачи.....	2
Теоретическое введение	2
Выполнение лабораторной работы	3
Выводы	7
Ответы на контрольные вопросы	7
1	7
2	7
3	7
4	7
5	7
6	8
7	8
8	8
9	8
10	8
11	9
12	9

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.

Задачи

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.

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 '|’.

Выполнение лабораторной работы

Осуществлю вход в систему, используя соответствующее имя пользователя. Запишу в файл file.txt названия файлов, содержащихся в каталоге /etc. Допишу в этот же файл названия файлов, содержащихся в вашем домашнем каталоге.(рис.1).

```
egorlisenkov@fedora:~$ ls -lR > dir-tree.list
egorlisenkov@fedora:~$ ls -lR /etc > file.txt
```

Выведу имена всех файлов из file.txt, имеющих расширение .conf, после чего запишу их в новый текстовый файл conf.txt. (рис.2) (рис. 3)

```
egorlisenkov@fedora:~$ cat file
```

```
-rw-r--r--. 1 root root 376 map 12 20:05 _copr:copr.fedorainfracloud.org
:phracek:PyCharm.repo
-rw-r--r--. 1 root root 728 окт 6 03:00 fedora-cisco-openh264.repo
-rw-r--r--. 1 root root 1239 окт 6 03:00 fedora.repo
-rw-r--r--. 1 root root 1286 окт 6 03:00 fedora-updates.repo
-rw-r--r--. 1 root root 1344 окт 6 03:00 fedora-updates-testing.repo
-rw-r--r--. 1 root root 199 map 12 20:05 google-chrome.repo
-rw-r--r--. 1 root root 1487 map 12 20:05 rpmpfusion-nonfree-nvidia-driver
.repo
-rw-r--r--. 1 root root 1391 map 12 20:05 rpmpfusion-nonfree-steam.repo

/etc/zfs-fuse:
итого 4
-rwxr-xr-x. 1 root root 2027 ноя 24 2015 zfs_pool_alert
egorlisenkov@fedora:~$
```

(рис.4)

```
итого 4
-rwxr-xr-x. 1 root root 2027 ноя 24 2015 zfs_pool_alert
egorlisenkov@fedora:~$ ls -lR ~/ >> dir-tree.list
egorlisenkov@fedora:~$ grep .conf
```

(рис.5).

```
lrwxrwxrwx. 1 root root 35 ноя 1 04:06 default.conf -> /etc/alternative
/qtchooser-default
egorlisenkov@fedora:~$ grep '\.conf' file.txt > conf.txt
egorlisenkov@fedora:~$ cat conf.txt
```

(рис.6)

```
user-5
lrwxrwxrwx. 1 root root 35 ноя 1 04:06 default.conf -> /etc/alternative
/qtchooser-default
egorlisenkov@fedora:~$ find ~ -name "c*" -print
```

Определяю, какие файлы в вашем домашнем каталоге имеют имена, начинавшиеся с символа c. (рис. 7)

```
/home/egorlisenkov/conf.txt
egorlisenkov@fedora:~$ ls -l | grep c*
-rw-r--r--. 1 egorlisenkov egorlisenkov 46587 мар 26 20:22 conf.txt
```

Выведу на экран (по странично) имена файлов из каталога /etc, начинающиеся с символа h. (рис.8).

```
egorlisenkov@fedora:~$ find /etc -name "h*"
egorlisenkov@fedora:~$ find /etc -name "h*" -print
```

1. Запустите в фоновом режиме процесс, который будет записывать в файл ~/logfile файлы, имена которых начинаются с log.
2. Удалите файл ~/logfile.
3. Запустите из консоли в фоновом режиме редактор gedit. (рис.9)

```
egorlisenkov@fedora:~$ find ~ -name "log*" -print > logfile &
[1] 5691
egorlisenkov@fedora:~$ rm logfile
[1]+  Завершён      find ~ -name "log*" -print > logfile
egorlisenkov@fedora:~$ gedit &
[1] 5778
egorlisenkov@fedora:~$
```

Определяю идентификатор процесса gedit, используя команду ps, конвейер и фильтр grep.(рис. 10)

```
egorlisenkov@fedora:~$ ps aux | grep gedit
egorlis+  5778  0.6  0.8 925476 64724 pts/1    Sl   20:27   0:00 gedit
egorlis+  6001  0.0  0.0 222456 2432 pts/1      S+   20:28   0:00 grep --color=auto
egorlisenkov@fedora:~$
```

(рис. 11)

```

egorlisenkov@fedora:~$ pgrep gedit
5778
egorlisenkov@fedora:~$ ps aux | grep gedit | grep -v grep
egorlis+  5778  0.3  0.8 925476 64724 pts/1    Sl   20:27   0:00 gedit
egorlisenkov@fedora:~$

```

Прочту справку (man) команды kill, после чего использую её для завершения процесса gedit. (рис. 12)

```

egorlis+  5778  0.3  0.8 925476 64724 pts/1    Sl   20:27   0:00 gedit
egorlisenkov@fedora:~$ man kill

```

(рис 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.

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

```

(рис. 14)

```

egorlisenkov@fedora:~$ man kill
egorlisenkov@fedora:~$ kill 5778
[1]+  Завершено      gedit

```

(рис. 15)

```

[1]+  Завершено      gedit
egorlisenkov@fedora:~$ man df

```



```

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
    Manual page df(1) line 10 (press h for help or q to quit)

```

Выполню команды **df** и **du**, предварительно получив более подробную информацию об этих командах, с помощью команды **man** (рис. 17)

```

egorlisenkov@fedora:~$ man du

```

(рис. 18)

```

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.

-0, --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 appar
ent size is usually smaller, it may be larger due to holes in
('sparse') files, internal fragmentation, indirect blocks, and the
Manual page du(1) line 1 (press h for help or q to quit)

```

(рис. 19)

```
egorlisenkov@fedora: ~$ df -vi
```

Воспользовавшись справкой команды `find`, выведу имена всех директорий, имеющих в вашем домашнем каталоге. (рис. 20)

```
egorlisenkov@fedora: ~$ du -a /home/egorlisenkov/work/study/2022-2023/Операционные системы/os-3ntro/Labs/Lab02
```

Выводы

Я усвоил материал и готов к дальнейшему изучению линукс!

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.