

# **Отчёт по лабораторной работе №12**

*дисциплина:Операционные системы*

Бондаренко Елизавета Валентиновна

# Содержание

1	Цель работы	5
2	Задание	6
3	Ход выполнения лабораторной работы	7
4	Контрольные вопросы	15
5	Выводы	20
	Список литературы	21

## Список иллюстраций

3.1	Запись в файл file.txt названия файлов, содержащихся в каталоге /etc	7
3.2	содержание . . . . .	8
3.3	Выводим имена всех файлов из file.txt, имеющих расширение .conf	8
3.4	Сами файлы . . . . .	9
3.5	Имена файлов начинающихся с символа с . . . . .	10
3.6	Имена файлов начинающихся с символа h в каталоге etc . . . . .	10
3.7	Имена файлов начинающихся с символа h . . . . .	10
3.8	Запуск в фоновом режиме . . . . .	11
3.9	Удаление . . . . .	11
3.10	Запуск в фоновом режиме . . . . .	11
3.11	Идентификатор процесса gedit . . . . .	11
3.12	Справка man kill . . . . .	12
3.13	Команды df,du . . . . .	12
3.14	Команда df . . . . .	13
3.15	Команда du . . . . .	13
3.16	Справка команды find . . . . .	13
3.17	Справка команды find . . . . .	14

## Список таблиц

# 1 Цель работы

В ходе выполнения лабораторной работы я должна ознакомиться с инструментами поиска файлов и фильтрации текстовых данных. Приобрести практические навыки по управлению процессами (и заданиями), по проверке использования диска и обслуживанию файловых систем.

## 2 Задание

1. Запишем в файл `file.txt` названия файлов, содержащихся в каталоге `/etc`.  
Допишем в этот же файл названия файлов, содержащихся в домашнем каталоге.
2. Выведем имена всех файлов из `file.txt`, имеющих расширение `.conf`, после чего запишем их в новый текстовый файл `conf.txt`.
3. Определим, какие файлы в вашем домашнем каталоге имеют имена, начинающиеся с символа `s`?
4. Выведем на экран (по странично) имена файлов из каталога `/etc`, начинающиеся с символа `h`.
5. Запустим в фоновом режиме процесс, который будет записывать в файл `~/logfile` файлы, имена которых начинаются с `log`.
6. Удалим файл `~/logfile`.
7. Запустим из консоли в фоновом режиме редактор `gedit`.
8. Определим идентификатор процесса `gedit`, используя команду `ps`, конвейер и фильтр `grep`.
9. Прочтем справку (`man`) команды `kill`, после чего используйте её для завершения процесса `gedit`.
10. Выполним команды `df` и `du`, предварительно получив более подробную информацию об этих командах, с помощью команды `man`.
11. Воспользовавшись справкой команды `find`, выведем имена всех директорий, имеющихся в домашнем каталоге.

### 3 Ход выполнения лабораторной работы

1. Запишем в файл file.txt названия файлов, содержащихся в каталоге /etc. Допишем в этот же файл названия файлов, содержащихся в домашнем каталоге.(рис. 3.1; 3.2)

```
evbondarenko@dk3n60 ~ $ ls -a /etc >file.txt
evbondarenko@dk3n60 ~ $ ls -a ~ - >>file.txt
ls: невозможно получить доступ к '-': Нет такого файла или каталога
evbondarenko@dk3n60 ~ $ ls -a ~ >>file.txt
evbondarenko@dk3n60 ~ $ cat file.txt
```

Рис. 3.1: Запись в файл file.txt названия файлов, содержащихся в каталоге /etc

```
.p7zip
.pki
play
.profile
public
public_html
reports
ski.places
.ssh
.texlive2021
tmp
.viminfo
.Xauthority
.xsession-errors
.xsession-errors.old
Видео
Документы
Загрузки
Изображения
Музыка
Общедоступные
Рабочий стол
Шаблоны
```

Рис. 3.2: содержание

2. Выведем имена всех файлов из file.txt, имеющих расширение .conf, после чего запишем их в новый текстовый файл conf.txt. Рис.3.3;3.4

```
evbondarenko@dk3n60 ~ $ grep -e '\.conf$' file.txt > conf.txt
evbondarenko@dk3n60 ~ $ cat conf.txt
```

Рис. 3.3: Выводим имена всех файлов из file.txt, имеющих расширение .conf



```
prelink.conf  
pump.conf  
pwdb.conf  
rc.conf  
request-key.conf  
resolv.conf  
rsyncd.conf  
rsyslog.conf  
sandbox.conf  
sddm.conf  
sensors3.conf  
signond.conf  
smartd.conf  
sudo.conf  
sudo_logsrvd.conf  
sysctl.conf  
udhcpd.conf  
updatedb.conf  
vconsole.conf  
warnquota.conf  
whois.conf  
xattr.conf  
xinetd.conf
```

Рис. 3.4: Сами файлы

3. Определим, какие файлы в вашем домашнем каталоге имеют имена, начинающиеся с символа с? Рис.3.5

```

evbondarenko@dk3n60 ~ $ find ~ -name "c*" -print
/afs/.dk.sci.pfu.edu.ru/home/e/v/evbondarenko/.config/gtk-3.0/colors.css
/afs/.dk.sci.pfu.edu.ru/home/e/v/evbondarenko/.config/gtk-3.0/assets/close-normal.svg
/afs/.dk.sci.pfu.edu.ru/home/e/v/evbondarenko/.config/gtk-3.0/assets/close-active.svg
/afs/.dk.sci.pfu.edu.ru/home/e/v/evbondarenko/.config/gtk-3.0/assets/close-hover.svg
/afs/.dk.sci.pfu.edu.ru/home/e/v/evbondarenko/.config/gtk-3.0/assets/close-backdrop-normal.svg
/afs/.dk.sci.pfu.edu.ru/home/e/v/evbondarenko/.config/gtk-3.0/assets/close-backdrop-active.svg
/afs/.dk.sci.pfu.edu.ru/home/e/v/evbondarenko/.config/gtk-3.0/assets/close-backdrop-hover.svg
/afs/.dk.sci.pfu.edu.ru/home/e/v/evbondarenko/.config/kdeconnect/certificate.p
/afs/.dk.sci.pfu.edu.ru/home/e/v/evbondarenko/.config/kdeconnect/config

```

Рис. 3.5: Имена файлов начинающихся с символа c

4. Выведем на экран (по странично) имена файлов из каталога /etc, начинающиеся с символа h. Рис. 3.6; 3.7

```

evbondarenko@dk3n60 ~ $ find /etc -name "h*" | less

```

Рис. 3.6: Имена файлов начинающихся с символа h в каталоге etc

```

/etc/conf.d/hostname
/etc/conf.d/hddtemp
/etc/conf.d/hdparm
/etc/conf.d/hwclock
/etc/conf.d/hsqldb
/etc/hotplug
find: '/etc/munge': Отказано в доступе
find: '/etc/sudoers.d': Отказано в доступе
/etc/mercurial/hgrc.d
/etc/gconf/gconf.xml.defaults/desktop/gnome/applications/help_viewer
/etc/gconf/gconf.xml.defaults/desktop/gnome/url-handlers/h323
/etc/gconf/gconf.xml.defaults/desktop/gnome/url-handlers/https
/etc/gconf/gconf.xml.defaults/desktop/gnome/url-handlers/http
/etc/gconf/gconf.xml.defaults/schemas/desktop/gnome/applications/help_viewer
lines 1-23find: '/etc/cron.daily': Отказано в доступе
find: '/etc/audit/plugins.d': Отказано в доступе

```

Рис. 3.7: Имена файлов начинающихся с символа h

5. Запустим в фоновом режиме процесс, который будет записывать в файл ~/logfile файлы, имена которых начинаются с log. Рис. 3.8

```

evbondarenko@dk3n60 ~ $ find /var/log -name "log*" -print > logfile
find: '/var/log/crond': Отказано в доступе
find: '/var/log/cron': Отказано в доступе
find: '/var/log/everything': Отказано в доступе
find: '/var/log/hal': Отказано в доступе
find: '/var/log/portage': Отказано в доступе
find: '/var/log/munge': Отказано в доступе
find: '/var/log/critical': Отказано в доступе
find: '/var/log/kernel': Отказано в доступе
find: '/var/log/private': Отказано в доступе
find: '/var/log/audit': Отказано в доступе
find: '/var/log/mysql': Отказано в доступе
find: '/var/log/telnet': Отказано в доступе
find: '/var/log/pwdfail': Отказано в доступе
find: '/var/log/mail': Отказано в доступе
find: '/var/log/gdm': Отказано в доступе
find: '/var/log/sshd': Отказано в доступе
find: '/var/log/apache2': Отказано в доступе
find: '/var/log/sandbox': Отказано в доступе

```

Рис. 3.8: Запуск в фоновом режиме

6. Удалим файл ~/logfile.Рис.3.9

```

evbondarenko@dk3n60 ~ $ rm logfile

```

Рис. 3.9: Удаление

7. Запустим из консоли в фоновом режиме редактор geditРис.3.10.

```

evbondarenko@dk3n60 ~ $ gedit &

```

Рис. 3.10: Запуск в фоновом режиме

8. Определим идентификатор процесса gedit, используя команду ps, конвейер и фильтр.Рис.3.11 grep.

```

evbondarenko@dk3n60 ~ $ ps | grep -i "gedit"
59432 pts/3    00:00:01 gedit

```

Рис. 3.11: Идентификатор процесса gedit

9. Прочтем справку (man) команды kill, после чего используйте её для завершения процесса gedit. Рис.3.12

```
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>
```

Рис. 3.12: Справка man kill

10. Выполним команды df и du, предварительно получив более подробную информацию об этих командах, с помощью команды man.Рис.3.13;3.14;3.15

```
evbondarenko@dk3n60 ~ $ man df
evbondarenko@dk3n60 ~ $ man du
```

Рис. 3.13: Команды df,du

```
DF(1)                                User Commands                                DF(1)

NAME
    df - report file system disk space usage

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

DESCRIPTION
    This manual page documents the GNU version of df. df displays the
    amount of disk 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. Disk 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 disk device node contain-
    ing 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 sys-
    tems, because on most kinds of systems doing so requires very non-
    portable intimate knowledge of file system structures.
```

Рис. 3.14: Команда df

```
DU(1)                                User Commands                                DU(1)

NAME
    du - estimate file space usage

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

DESCRIPTION
    Summarize disk 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
        I
    -a, --all
        write counts for all files, not just directories
```

Рис. 3.15: Команда du

11. Воспользовавшись справкой команды **find**, выведем имена всех директорий, имеющих в домашнем каталоге. Рис. 3.16; 3.17

```
evbondarenko@dk3n60 ~ $ man find
```

Рис. 3.16: Справка команды find

```
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...] [ex-
    pression]

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.
```

Рис. 3.17: Справка команды find

## 4 Контрольные вопросы

- 1) 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`.
- 2)       Перенаправление вывода в файл > Перенаправление вывода в файл и открытие файла в режиме добавления (данные добавляются в конец файла).
- 3) Конвейер (`pipe`) служит для объединения простых команд или утилит в цепочки, в которых результат работы предыдущей команды передаётся последующей. Синтаксис следующий: команда 1 | команда 2 (это означает, что вывод команды 1 передаётся на ввод команде 2)
- 4) Процесс рассматривается операционной системой как заявка на потребление всех видов ресурсов, кроме одного – процессорного времени. Этот последний важнейший ресурс распределяется операционной системой между другими единицами работы – потоками, которые и получили свое название благодаря тому, что они представляют собой последовательности (потоки выполнения) команд. Процесс – это выполнение программы. Он считается активной сущностью и реализует действия, указанные в программе. Про-

грамма представляет собой статический набор команд, а процесс это набор ресурсов и данных, использующихся при выполнении программы.

- 5) `pid`: идентификатор процесса (PID) процесса (process ID), к которому вызывают метод `gid`: идентификатор группы UNIX, в котором работает программа.
- 6) Любую выполняющуюся в консоли команду или внешнюю программу можно запустить в фоновом режиме. Для этого следует в конце имени команды указать знак амперсанда `&`. Запущенные фоном программы называются задачами (`jobs`). Ими можно управлять с помощью команды `jobs`, которая выводит список запущенных в данный момент задач.
- 7) `top` – это консольная программа, которая показывает список работающих процессов в системе. Программа в реальном времени отсортирует запущенные процессы по их нагрузке на процессор. `htop` – это продвинутый консольный мониторинг процессов. Утилита выводит постоянно меняющийся список системных процессов, который сортируется в зависимости от нагрузки на ЦПУ. Если делать сравнение с `top`, то `htop` показывает абсолютно все процессы в системе, время их непрерывного использования, загрузку процессоров и расход оперативной памяти.
- 8) `find` – 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. Команда `find` имеет такой синтаксис: `find [папка] [параметры] критерий шаблон [действие]` Папка – каталог в котором будем искать Параметры – дополнительные параметры, например, глубина поиска, и т.д. Критерий – по какому критерию будем искать: имя, дата создания, права, владелец и т.д. Шаблон – непосредственно значение по которому будем отбирать файлы. Основные параметры:



- -P никогда не открывать символические ссылки
  - -L - получает информацию о файлах по символическим ссылкам. Важно для дальнейшей обработки, чтобы обрабатывалась не ссылка, а сам файл.
  - -maxdepth - максимальная глубина поиска по подкаталогам, для поиска только в текущем каталоге установите 1.
  - -depth - искать сначала в текущем каталоге, а потом в подкаталогах
  - -mount искать файлы только в этой файловой системе.
  - -version - показать версию утилиты find
  - -print - выводить полные имена файлов
  - -type f - искать только файлы
  - -type d - поиск папки в Linux
- Основные критерии:
- -name - поиск файлов по имени
  - -perm - поиск файлов в Linux по режиму доступа
  - -user - поиск файлов по владельцу
  - -group - поиск по группе
  - -mtime - поиск по времени модификации файла
  - -atime - поиск файлов по дате последнего чтения
  - -nogroup - поиск файлов, не принадлежащих ни одной группе
  - -nouser - поиск файлов без владельцев
  - -newer - найти файлы новее чем указанный
  - -size - поиск файлов в Linux по их размеру
- Примеры: `find ~ -type d` поиск директорий в домашнем каталоге  
`find ~ -type f -name ".*"` поиск скрытых файлов в домашнем каталоге
- 9) Файл по его содержимому можно найти с помощью команды `grep`: «`grep -r "слово/выражение, которое нужно найти"`».
  - 10) Утилита `df`, позволяет проанализировать свободное пространство на всех подключенных к системе разделах.
  - 11) При выполнении команды `du` (без указания папки и опции) можно получить все файлы и папки текущей директории с их размерами. Для домашнего

каталога: `du ~/`

12) Основные сигналы (каждый сигнал имеет свой номер), которые используются для завершения процесса:

- **SIGINT** – самый безобидный сигнал завершения, означает Interrupt. Он отправляется процессу, запущенному из терминала с помощью сочетания клавиш Ctrl+C. Процесс правильно завершает все свои действия и возвращает управление;
- **SIGQUIT** – это еще один сигнал, который отправляется с помощью сочетания клавиш, программе, запущенной в терминале. Он сообщает ей что нужно завершиться и программа может выполнить корректное завершение или проигнорировать сигнал. В отличие от предыдущего, она генерирует дампы памяти. Сочетание клавиш Ctrl+Q;
- **SIGHUP** – сообщает процессу, что соединение с управляющим терминалом разорвано, отправляется, в основном, системой при разрыве соединения с интернетом;
- **SIGTERM** – немедленно завершает процесс, но обрабатывается программой, поэтому позволяет ей завершить дочерние процессы и освободить все ресурсы;
- **SIGKILL** – тоже немедленно завершает процесс, но, в отличие от предыдущего варианта, он не передается самому процессу, а обрабатывается ядром. Поэтому ресурсы и дочерние процессы остаются запущенными. Также для передачи сигналов процессам в Linux используется утилита `kill`, её синтаксис: `kill [-сигнал] [pid_процесса]` (PID – уникальный идентификатор процесса). Сигнал представляет собой один из выше перечисленных сигналов для завершения процесса. Перед тем, как выполнить остановку процесса, нужно определить его PID. Для этого используют команды `ps` и `grep`. Команда `ps` предназначена для вывода списка активных процессов в системе и информации о них. Команда `grep` запускается одновременно с `ps` (в канале) и будет выполнять поиск по результатам команды `ps`. Утилита

`pskill` – это оболочка для `kill`, она ведет себя точно так же, и имеет тот же синтаксис, только в качестве идентификатора процесса ей нужно передать его имя. `killall` работает аналогично двум предыдущим утилитам. Она тоже принимает имя процесса в качестве параметра и ищет его PID в директории `/proc`. Но эта утилита обнаружит все процессы с таким именем и завершит их.

## 5 Выводы

В ходе выполнения 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.

## **Список литературы**