

Отчёт по лабораторной работе №6

дисциплина: Операционные системы

Мишонков Максим Александрович

Содержание

1	Цель работы	4
2	Теоретическое введение	5
3	Выполнение лабораторной работы	6
4	Выводы	17

Список иллюстраций

3.1	Запись файлов	6
3.2	Запись файлов	6
3.3	Вывод имён файлов	7
3.4	Определение файлов	7
3.5	Вывод названий файлов	7
3.6	Названия файлов	8
3.7	Запуск процесса	8
3.8	Удаление файла	8
3.9	Запуск редактора	8
3.10	Определение идентификатора процесса	9
3.11	Справка команды kill	9
3.12	Команда man kill	9
3.13	Справка команды du	10
3.14	Команда du	10
3.15	Справка команды df	11
3.16	Команда df	11
3.17	Команда find	12

1 Цель работы

Целью данной лабораторной работы является 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 Теоретическое введение

Перенаправление ввода-вывода - возможность командной оболочки ряда операционных систем перенаправлять стандартные потоки в определённое пользователем место.

Команда `find` используется для поиска и отображения на экран имён файлов, соответствующих заданной строке символов. Формат команды: `find путь [-опции]`

Любой команде, выполняемой в системе, присваивается идентификатор процесса (process ID). Получить информацию о процессе и управлять им, пользуясь идентификатором процесса, можно из любого окна командного интерпретатора.

3 Выполнение лабораторной работы

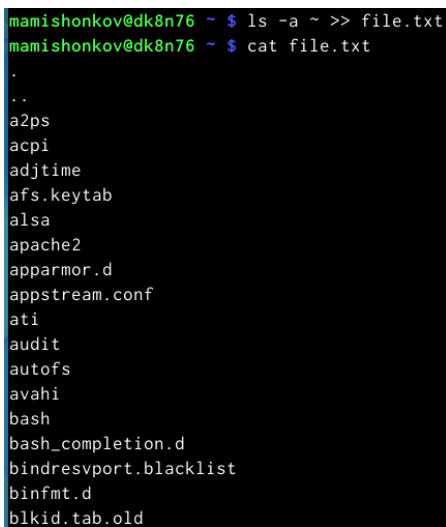
1. Записал в файл file.txt названия файлов, содержащихся в каталоге /etc. (рис. [3.1])



```
mamishonkov@dk8n76 ~ $ ls -a /etc >file.txt
```

Рис. 3.1: Запись файлов

2. Дописал в этот же файл названия файлов, содержащихся в домашнем каталоге. (рис. [3.2])



```
mamishonkov@dk8n76 ~ $ ls -a ~ >> file.txt
mamishonkov@dk8n76 ~ $ cat file.txt
.
..
a2ps
acpi
adjtime
afs.keytab
alsa
apache2
apparmor.d
appstream.conf
ati
audit
autofs
avahi
bash
bash_completion.d
bindresvport.blacklist
binfmt.d
blkid.tab.old
```

Рис. 3.2: Запись файлов

3. Вывел имена всех файлов из text.txt, имеющих расширение .conf, после чего записал их в новый текстовый файл conf.txt. (рис. [3.3])

```

mamishonkov@dk8n76 ~ $ grep -e '\.conf$' file.txt > conf.txt
mamishonkov@dk8n76 ~ $ cat conf.txt
appstream.conf
brltty.conf
ca-certificates.conf
cachefilesd.conf
cfg-update.conf
dhcpcd.conf
dispatch-conf.conf
dleyna-server-service.conf
dnsmasq.conf
e2fsck.conf
e2scrub.conf
etc-update.conf
fluidsynth.conf
fuse.conf
gai.conf
genkernel.conf
gssapi_mech.conf
host.conf
idmapd.conf
idn2.conf
idnalias.conf

```

Рис. 3.3: Вывод имён файлов

4. Определил, какие файлы в домашнем каталоге имеют имена, начинающиеся с символа с разными способами. (рис. [3.4])

```

mamishonkov@dk8n76 ~ $ find ~ -maxdepth 1 -name "c*" -print
/afs/.dk.sci.pfu.edu.ru/home/m/a/mamishonkov/conf.txt
mamishonkov@dk8n76 ~ $ ls ~/c*
/afs/.dk.sci.pfu.edu.ru/home/m/a/mamishonkov/conf.txt
mamishonkov@dk8n76 ~ $ ls -a ~ | grep c*
conf.txt
mamishonkov@dk8n76 ~ $

```

Рис. 3.4: Определение файлов

5. Вывел на экран имена файлов из каталога /etc, начинающиеся с символа h. (рис. [3.5], [3.6])

```

mamishonkov@dk8n76 ~ $ find /etc -maxdepth 1 -name "h*" | less
[1]+ Остановлен find /etc -maxdepth 1 -name "h*" | less
mamishonkov@dk8n76 ~ $

```

Рис. 3.5: Вывод названий файлов

```

/etc/hal
/etc/hostname
/etc/hosts.allow
/etc/hosts.allow
/etc/harbour.cfg
/etc/highlight
/etc/htdig
/etc/hotplug
/etc/httpd
/etc/hosts
/etc/hotplug.d
/etc/hsqldb
/etc/harbour
/etc/host.conf
lines 1-13/13 (END)

```

Рис. 3.6: Названия файлов

6. Запустил в фоновом режиме процесс, который будет записывать в файл ~/logfile файлы, имена которых начинаются с log. (рис. [3.7])

```

mamishonkov@dk8n76 ~ $ find /etc -maxdepth 1 -name "log*" >> ~/logfile $
find: paths must precede expression: '$'
mamishonkov@dk8n76 ~ $

```

Рис. 3.7: Запуск процесса

7. Удалил файл ~/logfile. (рис. [3.8])

```

mamishonkov@dk8n76 ~ $ rm ~/logfile
mamishonkov@dk8n76 ~ $ ls
abc1      games      public_html  Изображения
australia may        reports      Общедоступные
bin        monthly    ski.places   'Отчёт 4 Markdown и скриншоты'
conf.txt   my_os      work         'Рабочий стол'
feathers   play       Документы    Шаблоны
file.txt   public     Загрузки
mamishonkov@dk8n76 ~ $

```

Рис. 3.8: Удаление файла

8. Запустил в консоли в фоновом режиме редактор gedit. (рис. [3.9])

```

mamishonkov@dk8n76 ~ $ gedit &
[3] 7246
mamishonkov@dk8n76 ~ $

```

Рис. 3.9: Запуск редактора

9. Определил идентификатор процесса gedit, используя команду ps, конвейер и фильтр grep. Узнать идентификатор можно также, используя команду “pgrep gedit” или “pidof gedit”. (рис. [3.10])

```
mamishonkov@dk8n76 ~ $ ps | grep -i "gedit"
[3]  Завершён      gedit
mamishonkov@dk8n76 ~ $
```

Рис. 3.10: Определение идентификатора процесса

10. Прочёл справку команды kill, после чего использовал её для завершения процесса gedit. (рис. [3.11], [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>

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

Рис. 3.11: Справка команды kill

```
mamishonkov@dk8n76 ~ $ man kill
```

Рис. 3.12: Команда man kill

11. Выполнил команду `du`, предварительно прочитав справку об этой команде.
(рис. [3.13], [3.14])

```
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

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

    --apparent-size
        print apparent sizes, rather than disk usage; although the ap-
Manual page du(1) line 1 (press h for help or q to quit)
```

Рис. 3.13: Справка команды `du`

```
1010  ./work/study/2022-2023/Операционные системы/os-intro/.git/objects/24
36    ./work/study/2022-2023/Операционные системы/os-intro/.git/objects/49
28    ./work/study/2022-2023/Операционные системы/os-intro/.git/objects/14
63    ./work/study/2022-2023/Операционные системы/os-intro/.git/objects/51
114   ./work/study/2022-2023/Операционные системы/os-intro/.git/objects/7c
35    ./work/study/2022-2023/Операционные системы/os-intro/.git/objects/3a
233   ./work/study/2022-2023/Операционные системы/os-intro/.git/objects/00
783   ./work/study/2022-2023/Операционные системы/os-intro/.git/objects/ff
97    ./work/study/2022-2023/Операционные системы/os-intro/.git/objects/a3
59    ./work/study/2022-2023/Операционные системы/os-intro/.git/objects/8b
1381  ./work/study/2022-2023/Операционные системы/os-intro/.git/objects/d7
1047  ./work/study/2022-2023/Операционные системы/os-intro/.git/objects/36
3     ./work/study/2022-2023/Операционные системы/os-intro/.git/objects/db
3     ./work/study/2022-2023/Операционные системы/os-intro/.git/objects/f1
3     ./work/study/2022-2023/Операционные системы/os-intro/.git/objects/60
10    ./work/study/2022-2023/Операционные системы/os-intro/.git/objects/1c
5     ./work/study/2022-2023/Операционные системы/os-intro/.git/objects/b8
1371  ./work/study/2022-2023/Операционные системы/os-intro/.git/objects/eb
624   ./work/study/2022-2023/Операционные системы/os-intro/.git/objects/ef
3     ./work/study/2022-2023/Операционные системы/os-intro/.git/objects/56
4     ./work/study/2022-2023/Операционные системы/os-intro/.git/objects/ce
3     ./work/study/2022-2023/Операционные системы/os-intro/.git/objects/f0
369   ./work/study/2022-2023/Операционные системы/os-intro/.git/objects/4f
47    ./work/study/2022-2023/Операционные системы/os-intro/.git/objects/e6
```

Рис. 3.14: Команда `du`

12. Выполнил команду `df`, предварительно прочитав справку об этой команде. (рис. [3.15], [3.16])

```
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.

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

Рис. 3.15: Справка команды `df`

```
mamishonkov@dk8n76 ~ $ man df
[7]+  Остановлен   man df
mamishonkov@dk8n76 ~ $ df
df: /run/user/4672/doc: Операция не позволена
Файловая система  1K-блоков  Использовано  Доступно  Использовано%  Смонтировано в
none              3999712      17188      3982524      1% /run
udev              10240         0          10240       0% /dev
tmpfs             3999712         0      3999712       0% /dev/shm
/dev/sda8         484939832    69118692    391114084    16% /
tmpfs             3999716      85272      3914444       3% /tmp
AFS               2147483647     0    2147483647     0% /afs
tmpfs             799940        192       799748       1% /run/user/4672
mamishonkov@dk8n76 ~ $
```

Рис. 3.16: Команда `df`

13. Воспользовавшись справкой команды `find`, вывел имена всех директорий, имеющих в домашнем каталоге. (рис. [3.17])

```
mamishonkov@dk8n76 ~ $ man find
[8]+ Остановлен      man find
mamishonkov@dk8n76 ~ $ find -maxdepth 1 -type d
.
./.ssh
./public
./.mozilla
./.local
./Рабочий стол
./Загрузки
./Шаблоны
./Общедоступные
./Документы
./Изображения
./config
./gnupg
./work
./.texlive2022
./Отчёт 4 Markdown и скриншоты
./pki
./.fr-KH4001
```

Рис. 3.17: Команда find

Ответы на 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:

команда1|команда2 (это означает, что вывод команды 1 передается 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.

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

5). `pid`: идентификатор процесса (PID) процесса (`processID`), 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 `stop`, 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

- typef - искать только файлы

- typed - 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 пр

SIGQUIT–это еще один сигнал, который отправляется с помощью сочетания клавиш, пр

SIGHUP–сообщает процессу, что соединение с управляющим терминалом разорвано, отпр

SIGTERM–немедленно завершает процесс, но обрабатывается программой, поэтому позво

SIGKILL–тоже немедленно завершает процесс, но, в отличие от предыдущего варианта,

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 `kill` – 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.

4 Выводы

В ходе выполнения данной лабораторной работы я изучил инструменты поиска файлов и фильтрации текстовых данных, а также приобрёл практические навыки по управлению процессами и заданиями, по проверке использования диска и обслуживанию файловых систем.