

Операционные системы

Лабораторная работа №6

Матюшкин Денис Владимирович (НПИбд-02-21)

Содержание

1	Цель работы	3
2	Ход работы	4
3	Контрольные вопросы	9
4	Вывод:	12

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 Ход работы

1. Осуществим вход в систему, используя соответствующее имя пользователя.
2. Запишем в файл *file.txt* названия файлов, содержащихся в каталоге */etc*. Допишем в этот же файл названия файлов, содержащихся в вашем домашнем каталоге (рис. 2.1).

```
[dvmatyushkin@dvmatyushkin ~]$ ls /etc >> file.txt
[dvmatyushkin@dvmatyushkin ~]$ ls >> file.txt
[dvmatyushkin@dvmatyushkin ~]$ cat file.txt
abrt
adjtime
aliases
alsa
alternatives
anaconda
anthy-unicode.conf
appstream.conf
```

Рис. 2.1: Создание и заполнение текстового файла

3. Выведем имена всех файлов из *file.txt*, имеющих расширение *.conf*, после чего запишем их в новый текстовый файл *conf.txt* (рис. 2.2).

```
[dvmatyushkin@dvmatyushkin ~]$ grep .conf$ file.txt >> conf.txt
[dvmatyushkin@dvmatyushkin ~]$ cat conf.txt
anthy-unicode.conf
appstream.conf
asound.conf
brltty.conf
chrony.conf
dconf
dleyna-renderer-service.conf
dleyna-server-service.conf
dnsmasq.conf
```

Рис. 2.2: Создание и заполнение нового текстового файла

4. Определим двумя методами, какие файлы в вашем домашнем каталоге имеют имена, начинавшиеся с символа *c* (рис. 2.3):

- `ls | grep c`
- `find c*`

```
[dvmatyushkin@dvmatyushkin ~]$ ls | grep c*
conf.txt
[dvmatyushkin@dvmatyushkin ~]$ find c*
conf.txt
[dvmatyushkin@dvmatyushkin ~]$ ls
bin      file.txt  resources  themes  Видео      Загрузки  Музыка      'Рабочий стол'
conf.txt  opt       snap       work    Документы  Изображения  Общедоступные  Шаблоны
[dvmatyushkin@dvmatyushkin ~]$
```

Рис. 2.3: Поиск файлов в каталоге

5. Выведем на экран (по странично) имена файлов из каталога */etc*, начинающиеся с символа *h* (рис. 2.4).

```
[dvmatyushkin@dvmatyushkin ~]$ find /etc -name "h*" -print
find: '/etc/audit': Отказано в доступе
/etc/avahi/hosts
/etc/brltty/Contraction/ha.ctb
/etc/brltty/Input/bm/horizontal.kti
/etc/brltty/Input/hd
/etc/brltty/Input/hm
/etc/brltty/Input/ht
/etc/brltty/Input/hw
/etc/brltty/Text/he.ttb
/etc/brltty/Text/hi.ttb
/etc/brltty/Text/hr.ttb
```

Рис. 2.4: Вывод список файлов из каталога

6. Запустим в фоновом режиме процесс, который будет записывать в файл *~/logfile* файлы, имена которых начинаются с *log* (рис. 2.5).

```
[dvmatyushkin@dvmatyushkin ~]$ find / -name "log*" -print >> logfile &
[2] 6995
find: '/boot/lost+found': Отказано в доступе
find: '/boot/grub2': Отказано в доступе
find: '/boot/efi': Отказано в доступе
[dvmatyushkin@dvmatyushkin ~]$ find: '/boot/loader/entries': Отказано в доступе
find: '/home/dvmatyushkin/.local/share/Trash/files/monthly/play': Отказано в доступе
find: '/proc/tty/driver': Отказано в доступе
find: '/proc/1/task/1/fd': Отказано в доступе
find: '/proc/1/task/1/ns': Отказано в доступе
find: '/proc/1/fd': Отказано в доступе
```

Рис. 2.5: Запуск процесса в фоновом режиме

7. Удалим файл `~/logfile` (рис. 2.6).

```
[dvmatyushkin@dvmatyushkin ~]$ rm logfile
[dvmatyushkin@dvmatyushkin ~]$
```

Рис. 2.6: Удаление файла

8. Запустим из консоли в фоновом режиме редактор `gedit` (рис. 2.7).

```
[dvmatyushkin@dvmatyushkin ~]$ gedit &
[2] 7239
[dvmatyushkin@dvmatyushkin ~]$
```

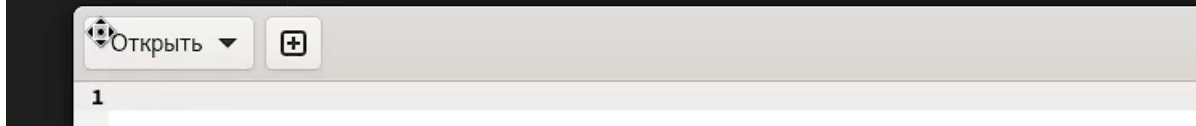


Рис. 2.7: Запуск редактора в фоновом режиме

9. Определим идентификатор процесса `gedit`, используя команду `ps`, конвейер и фильтр `grep`. Это можно сделать командой `ps aux | grep gedit`. После чего командой `kill` завершим процесс `gedit` (рис. 2.8).

```
[dvmatyushkin@dvmatyushkin ~]$ ps aux | grep gedit
dvmatyu+   7239  1.7  1.9 795732 79108 pts/0    Sl   15:34   0:01 gedit
dvmatyu+   7305  0.0  0.0 221692  2488 pts/0    S+   15:35   0:00 grep --color=auto gedit
[dvmatyushkin@dvmatyushkin ~]$ man kill
[dvmatyushkin@dvmatyushkin ~]$ kill 7239
[2]-  Завершено      gedit
[dvmatyushkin@dvmatyushkin ~]$
```

Рис. 2.8: Определение идентификатора процесса gedit

10. Выполним команды *df* и *du*, предварительно получив более подробную информацию об этих командах, с помощью команды *man* (рис. 2.9).

```
[dvmatyushkin@dvmatyushkin ~]$ df -vi
Файловая система  Инодов  ИИспользовано  ИСвободно  ИИспользовано%  Смонтировано в
devtmpfs          131072         461      130611           1% /dev
tmpfs             501985         10      501975           1% /dev/shm
tmpfs             819200        925      818275           1% /run
/dev/sda2          0              0          0             - /
tmpfs            1048576        52     1048524           1% /tmp
/dev/sda2          0              0          0             - /home
/dev/loop1        12866         12866         0          100% /var/lib/snapd/snap/go/9605
/dev/loop0        11778         11778         0          100% /var/lib/snapd/snap/core20/1405
/dev/loop2         484           484          0          100% /var/lib/snapd/snap/snapd/15534
/dev/sda1         65536         456       65080           1% /boot
OC               1000         -999000     1000000         - /media/sf_
tmpfs            100397        155      100242           1% /run/user/1000
/dev/sr0          0              0          0             - /run/media/dvmatyushkin/VBox_GAs_6.1.34
[dvmatyushkin@dvmatyushkin ~]$ du -a ~/
4      /home/dvmatyushkin/.mozilla/extensions/{ec8030f7-c20a-464f-9b0e-13a3a9e97384}/langpack-ru@firefox.mozilla.org.xpi
4      /home/dvmatyushkin/.mozilla/extensions/{ec8030f7-c20a-464f-9b0e-13a3a9e97384}/.fedora-langpack-install
8      /home/dvmatyushkin/.mozilla/extensions/{ec8030f7-c20a-464f-9b0e-13a3a9e97384}
8      /home/dvmatyushkin/.mozilla/extensions
0      /home/dvmatyushkin/.mozilla/plugins
0      /home/dvmatyushkin/.mozilla/firefox/Crash Reports/events
4      /home/dvmatyushkin/.mozilla/firefox/Crash Reports/InstallTime20211007232822
4      /home/dvmatyushkin/.mozilla/firefox/Crash Reports/InstallTime20220413162341
8      /home/dvmatyushkin/.mozilla/firefox/Crash Reports
0      /home/dvmatyushkin/.mozilla/firefox/Pending Pings
```

Рис. 2.9: Выполнение команд *df* и *du*

11. Воспользовавшись справкой команды *find*, выведем имена всех директорий, имеющих в вашем домашнем каталоге (рис. 2.10).

```
[dvmatyushkin@dvmatyushkin ~]$ find ~/ -type d
/home/dvmatyushkin/
/home/dvmatyushkin/.mozilla
/home/dvmatyushkin/.mozilla/extensions
/home/dvmatyushkin/.mozilla/extensions/{ec8030f7-c20a-464f-9b0e-13a3a9e97384}
/home/dvmatyushkin/.mozilla/plugins
/home/dvmatyushkin/.mozilla/firefox
/home/dvmatyushkin/.mozilla/firefox/Crash Reports
/home/dvmatyushkin/.mozilla/firefox/Crash Reports/events
/home/dvmatyushkin/.mozilla/firefox/Pending Pings
```

Рис. 2.10: Вывод имени всех директорий командой find

3 Контрольные вопросы

1. Какие потоки ввода вывода вы знаете?

- `stdin` — стандартный поток ввода (по умолчанию: клавиатура), файловый дескриптор 0.
- `stdout` — стандартный поток вывода (по умолчанию: консоль), файловый дескриптор 1.
- `stderr` — стандартный поток вывода сообщений об ошибках (по умолчанию: консоль) файловый дескриптор 2.

2. Объясните разницу между операцией `>` и `»`.

- `>` - перезапись информации в файл
- `»` - добавление информации, а не ее перезапись

3. Что такое конвейер?

- Конвейер (`pipe`) служит для объединения простых команд или утилит в цепочки, в которых результат работы предыдущей команды передаётся последующей. Пример: `ls -la | sort > sortilg_list`

4. Что такое процесс? Чем это понятие отличается от программы?

- Программа - это сценарий, хранящийся на диске или, по-видимому, предыдущий этап процесса. Наоборот, процесс является событием программы в процессе выполнения.

5. Что такое PID и GID?

- PID - идентификатор процесса.
- GID - идентификационный номер группы данного процесса.

6. Что такое задачи и какая команда позволяет ими управлять?

- Запущенные фоном программы называются задачами. Ими можно управлять с помощью команды `jobs`, которая выводит список запущенных в данный момент задач. Для завершения задачи необходимо выполнить команду `kill`

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

- `top` — Консольная команда, которая выводит список работающих в системе процессов и информацию о них. По умолчанию она в реальном времени сортирует их по нагрузке на процессор.
- `htop` — Продвинутый монитор процессов. Он был задуман заменить стандартную программу `top`. `Htop` показывает динамический список системных процессов, список обычно выравнивается по использованию ЦПУ. В отличие от `top`, `htop` показывает все процессы в системе. Также показывает время непрерывной работы, использование процессоров и памяти.

8. Назовите и дайте характеристику команде поиска файлов. Приведите примеры использования этой команды.

- `find` - 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.
- Пример: `find / -name "c*" -print`

9. Можно ли по контексту (содержанию) найти файл? Если да, то как?

- Для поиска файла по содержимому проще всего воспользоваться командой `grep`.

- Пример: `grep -lir 'Цель работы' /home/dvmatyushkin/work/study/2021-2022/Операционные системы/os-intro/labs/lab06/report`

10. Как определить объем свободной памяти на жёстком диске?

- Командой: `df -vi`

11. Как определить объем вашего домашнего каталога?

- Командой: `du -sh ~/`

12. Как удалить зависший процесс?

- Командой: `kill`

4 Вывод:

- В ходе этой лабораторной работы мы ознакомились с инструментами поиска файлов и фильтрации текстовых данных. Приобрели практические навыки: по управлению процессами (и заданиями), по проверке использования диска и обслуживанию файловых систем.