

Отчёт по лабораторной работе №6

Поиск файлов. Перенаправление ввода-вывода. Просмотр запущенных процессов

Самигуллин Эмиль Артурович

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 Теоретическое введение

2.1 1. Перенаправление ввода-вывода

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 `>`, `>>`, `<`, `<<`. Рассмотрим пример.

```
# Перенаправление stdout (вывода) в файл.
```

```
# Если файл отсутствовал, то он создаётся,
```

```
# иначе -- перезаписывается.
```

```
# Создаёт файл, содержащий список дерева каталогов.
```

```
$ ls -lR > dir-tree.list
```

```
$ 1>filename
```

```
# Перенаправление вывода (stdout) в файл "filename".
```

```
$ 1>>filename
```

```
# Перенаправление вывода (stdout) в файл "filename",
# файл открывается в режиме добавления.
$ 2>filename
# Перенаправление stderr в файл "filename".
$ 2>>filename
# Перенаправление stderr в файл "filename",
# файл открывается в режиме добавления.
$ &>filename
# Перенаправление stdout и stderr в файл "filename".
```

2.2 2. Конвейер

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
# означает, что вывод команды 1 передаётся на ввод команде 2
```

Конвейеры можно группировать в цепочки и выводить с помощью перенаправления в файл, например:

```
$ ls -la |sort > sortilg_list
```

вывод команды `ls -la` передаётся команде сортировки `sort\verb`, которая пишет результат в файл `sorting_list\verb`. Чаще всего скрипты на Bash используются в качестве автоматизации каких-то рутинных операций в консоли, отсюда иногда возникает необходимость в обработке stdout одной команды и передача на stdin другой команде, при этом результат выполнения команды должен обработан. ## 3. Поиск файла Команда `find` используется для поиска и отображения на экран имён файлов, соответствующих заданной строке символов. Формат команды:

`$ find путь [-опции]`

Путь определяет каталог, начиная с которого по всем подкаталогам будет вестись поиск. Примеры:

1. Вывести на экран имена файлов из вашего домашнего каталога и его подкаталогов, начинающихся на f:

```
$ find ~ -name "f*" -print
```

Здесь `~` — обозначение вашего домашнего каталога, `-name` — после этой опции указывается имя файла, который нужно найти, `"f*"` — строка символов, определяющая имя файла, `-print` — опция, задающая вывод результатов поиска на экран.

2. Вывести на экран имена файлов в каталоге `/etc`, начинающихся с символа `p`:

```
$ find /etc -name "p*" -print
```

3. Найти в Вашем домашнем каталоге файлы, имена которых заканчиваются символом `~` и удалить их:

```
$ find ~ -name "*~" -exec rm "{}" \;
```

Здесь опция `-exec rm "{}"` ; задаёт применение команды `rm` ко всем файлам, имена которых соответствуют указанной после опции `-name` строке символов. Для просмотра опций команды `find` воспользуйтесь командой `man`.

2.3 4. Фильтрация текста

Найти в текстовом файле указанную строку символов позволяет команда `grep`.
Формат команды:

```
$ grep строка имя_файла
```

Кроме того, команда `grep` способна обрабатывать стандартный вывод других команд (любой текст). Для этого следует использовать конвейер, связав вывод команды с вводом `grep`. Примеры: 1. Показать строки во всех файлах в вашем домашнем каталоге с именами, начинающимися на `f`, в которых есть слово `begin`:

```
$ grep begin f*
```

2. Найти в текущем каталоге все файлы, содержащих в имени «лаб»:

```
$ ls -l | grep лаб
```

2.4 5. Проверка использования диска

Команда `df` показывает размер каждого смонтированного раздела диска. Формат команды:

```
$ df [-опции] [файловая_система]
```

Пример:

```
$ df -vi
```

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

```
$ du [-опции] [имя_файла...]
```

Пример.

```
$ du -a ~/
```

На `afs` можно посмотреть использованное пространство командой

```
$ fs quota
```

2.5 6. Управление задачами

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

```
$ gedit &
```

Будет запущен текстовый редактор `gedit` в фоновом режиме. Консоль при этом не будет заблокирована. Запущенные фоном программы называются задачами (`jobs`). Ими можно управлять с помощью команды `jobs`, которая выводит список запущенных в данный момент задач. Для завершения задачи необходимо выполнить команду

```
$ kill %номер задачи
```

2.6 7. Управление процессами

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.

2.7 8. Получение информации о процессах

Команда `ps` используется для получения информации о процессах. Формат команды:

```
$ ps [-опции]
```

Для получения информации о процессах, управляемых вами и запущенных (работающих или остановленных) на вашем терминале, используйте опцию `aux`. Пример:

```
$ ps aux
```

Для запуска команды в фоновом режиме необходимо в конце командной строки указать знак `&` (амперсанд). Пример работы, требующей много машинного времени для выполнения, и которую целесообразно запустить в фоновом режиме:

```
$ find /var/log -name "*.log" -print > l.log &
```


3 Выполнение лабораторной работы

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

```
[edsmirnovmaljce@edsmirnovmaljce ~]$ ls /etc >> file.txt
```

рис. 1

Дописал в этот же файл имена объектов домашнего каталога.(рис. 2)

```
[edsmirnovmaljce@edsmirnovmaljce ~]$ ls ~ >> file.txt
```

рис. 2

2. Вывел имена всех файлов из file.txt, имеющих расширение .conf и записал их в conf.txt.(рис.3)

```
[edsmirnovmaljce@edsmirnovmaljce ~]$ cat file.txt | grep .*\.conf >> conf.txt
```

рис. 3

3. Двумя способами вывел файлы домашнего каталога, начинающиеся на с.(рис. 4)

```
[edsmirnovmaljce@edsmirnovmaljce ~]$ ls | grep c.*
conf.txt
[edsmirnovmaljce@edsmirnovmaljce ~]$ find ~ -name c* -print
/home/edsmirnovmaljce/conf.txt
```

рис. 4

4. Вывел на экран имена файлов каталога /etc, начинающиеся с символа h. (рис. 5)

```
[edsmirnovmaljce@edsmirnovmaljce ~]$ find /etc -name "h*" -print | less
```

рис. 5

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

```
[edsmirnovmaljce@edsmirnovmaljce ~]$ find /home -name "log*" -print >> logfile &  
[1] 3879
```

рис. 6

6. Удалил logfile.(рис. 7)

```
[edsmirnovmaljce@edsmirnovmaljce ~]$ rm logfile  
[1]+  Завершён          find /home -name "log*" -print >> logfile
```

рис. 7

7. Запустил в фоновом режиме gedit.(рис. 8)

```
[edsmirnovmaljce@edsmirnovmaljce ~]$ gedit &  
[1] 4017
```

рис. 8

8. Определил идентификатор процесса gedit 2 способами.(рис. 9.1, 9.2)

```
[edsmirnovmaljce@edsmirnovmaljce ~]$ ps | grep gedit  
4017 pts/0    00:00:01 gedit
```

рис. 9.1

```
[edsmirnovmaljce@edsmirnovmaljce ~]$ pidof gedit  
4017
```

рис. 9.2

9. Завершил процесс gedit.(рис. 10)

```
[edsmirnovmaljce@edsmirnovmaljce ~]$ kill 4017  
[1]+  Завершено        gedit
```

рис. 10

10. С помощью команды `df` узнал список подключенных устройств, информацию о занятом месте (в килобайтах) и точку монтирования.(рис. 11)

```
[edsmirnovmaljce@edsmirnovmaljce ~]$ df
Файловая система 1К-блоков  Использовано  Доступно  Использовано%  Смонтировано в
devtmpfs           4096          0      4096           0% /dev
tmpfs              1010444        0    1010444          0% /dev/shm
tmpfs              404180       1380    402800           1% /run
/dev/sda2          82836480    13158816  68691104          17% /
tmpfs              1010444        328    1010116           1% /tmp
/dev/sda2          82836480    13158816  68691104          17% /home
/dev/loop1         29184        29184         0          100% /var/lib/snapd/snap/marp-cli-carroarmato0/33
/dev/loop0         56960        56960         0          100% /var/lib/snapd/snap/core18/2344
/dev/loop2         45824        45824         0          100% /var/lib/snapd/snap/snapd/15534
/dev/sda1          996780       223104    704864           25% /boot
tmpfs              202088        124    201964           1% /run/user/1000
/dev/sr0           60096        60096         0          100% /run/media/edsmirnovmaljce/VBox_GAs_6.1.34
```

рис. 11

11. С помощью команды `du` вывел все папки постранично.(рис. 12)

```

12 ./mozilla/firefox/mngi6i5l.default-release/storage/default/https+++tex-stackexchange-com.translate.google.com/ls
16 ./mozilla/firefox/mngi6i5l.default-release/storage/default/https+++tex-stackexchange-com.translate.google.com/ls
12 ./mozilla/firefox/mngi6i5l.default-release/storage/default/https+++forum.ubuntu.ru/ls
16 ./mozilla/firefox/mngi6i5l.default-release/storage/default/https+++forum.ubuntu.ru
12 ./mozilla/firefox/mngi6i5l.default-release/storage/default/https+++www.linux.org.ru/ls
16 ./mozilla/firefox/mngi6i5l.default-release/storage/default/https+++www.linux.org.ru
12 ./mozilla/firefox/mngi6i5l.default-release/storage/default/https+++mc.yandex.md^partitionKey=%28https%2Clinux.org.ru%29/ls
16 ./mozilla/firefox/mngi6i5l.default-release/storage/default/https+++mc.yandex.md^partitionKey=%28https%2Clinux.org.ru%29
12 ./mozilla/firefox/mngi6i5l.default-release/storage/default/https+++yastatic.net^partitionKey=%28https%2Clinux.org.ru%29/ls
16 ./mozilla/firefox/mngi6i5l.default-release/storage/default/https+++yastatic.net^partitionKey=%28https%2Clinux.org.ru%29
16 ./mozilla/firefox/mngi6i5l.default-release/storage/default/https+++www.cyberforum.ru/ls
20 ./mozilla/firefox/mngi6i5l.default-release/storage/default/https+++www.cyberforum.ru
12 ./mozilla/firefox/mngi6i5l.default-release/storage/default/https+++serieslife.online^partitionKey=%28https%2Ccyberforum.ru%29/ls
16 ./mozilla/firefox/mngi6i5l.default-release/storage/default/https+++serieslife.online^partitionKey=%28https%2Ccyberforum.ru%29
16 ./mozilla/firefox/mngi6i5l.default-release/storage/default/https+++kata.academy/ls
20 ./mozilla/firefox/mngi6i5l.default-release/storage/default/https+++kata.academy
12 ./mozilla/firefox/mngi6i5l.default-release/storage/default/https+++yastatic.net^partitionKey=%28https%2Ccyberforum.ru%29/ls
16 ./mozilla/firefox/mngi6i5l.default-release/storage/default/https+++yastatic.net^partitionKey=%28https%2Ccyberforum.ru%29
0 ./mozilla/firefox/mngi6i5l.default-release/storage/default/https+++www.youtube.com^partitionKey=%28https%2Ckata.academy%29/idb/2171031483YattIedMb.files
44 ./mozilla/firefox/mngi6i5l.default-release/storage/default/https+++www.youtube.com^partitionKey=%28https%2Ckata.academy%29/idb
12 ./mozilla/firefox/mngi6i5l.default-release/storage/default/https+++www.youtube.com^partitionKey=%28https%2Ckata.academy%29/ls
60 ./mozilla/firefox/mngi6i5l.default-release/storage/default/https+++www.youtube.com^partitionKey=%28https%2Ckata.academy%29
36 ./mozilla/firefox/mngi6i5l.default-release/storage/default/https+++moe.video^partitionKey=%28https%2Ccyberforum.ru%29/ls
40 ./mozilla/firefox/mngi6i5l.default-release/storage/default/https+++moe.video^partitionKey=%28https%2Ccyberforum.ru%29
28 ./mozilla/firefox/mngi6i5l.default-release/storage/default/https+++playreplay.me^partitionKey=%28https%2Ccyberforum.ru%29/ls
32 ./mozilla/firefox/mngi6i5l.default-release/storage/default/https+++playreplay.me^partitionKey=%28https%2Ccyberforum.ru%29
12 ./mozilla/firefox/mngi6i5l.default-release/storage/default/https+++mc.yandex.md^partitionKey=%28https%2Ckata.academy%29/ls
16 ./mozilla/firefox/mngi6i5l.default-release/storage/default/https+++mc.yandex.md^partitionKey=%28https%2Ckata.academy%29
12 ./mozilla/firefox/mngi6i5l.default-release/storage/default/https+++thesame.tv^partitionKey=%28https%2Ccyberforum.ru%29/ls

```

рис. 12

12. Вывел имена всех директорий домашнего каталога с помощью команды find.
(рис. 13)

```
[edsmirnovmaljce@edsmirnovmaljce ~]$ find ~ -maxdepth 1 -type d
/home/edsmirnovmaljce
/home/edsmirnovmaljce/.mozilla
/home/edsmirnovmaljce/.cache
/home/edsmirnovmaljce/.config
/home/edsmirnovmaljce/.local
/home/edsmirnovmaljce/Рабочий стол
/home/edsmirnovmaljce/Загрузки
/home/edsmirnovmaljce/Шаблоны
/home/edsmirnovmaljce/Общедоступные
/home/edsmirnovmaljce/Документы
/home/edsmirnovmaljce/Музыка
/home/edsmirnovmaljce/Изображения
/home/edsmirnovmaljce/Видео
/home/edsmirnovmaljce/.ssh
/home/edsmirnovmaljce/.gnupg
/home/edsmirnovmaljce/work
/home/edsmirnovmaljce/.texlive2021
/home/edsmirnovmaljce/.grip
/home/edsmirnovmaljce/.npm
/home/edsmirnovmaljce/.texlive2022
/home/edsmirnovmaljce/go_home
/home/edsmirnovmaljce/bin
/home/edsmirnovmaljce/getting-started
/home/edsmirnovmaljce/monthly
/home/edsmirnovmaljce/reports
/home/edsmirnovmaljce/ski.plases
/home/edsmirnovmaljce/australia
/home/edsmirnovmaljce/play
/home/edsmirnovmaljce/3
```

рис. 13

4 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.
2. `>` - создает новый файл; `>>` - дописывает старый.
3. Конвейер - последовательность команд, в которой следующая команда получает на вход вывод предыдущей команды.
4. Программа - это определенная группа упорядоченных операций, которые должны быть выполнены. С другой стороны, экземпляр выполняемой программы является процессом.
5. PID - это идентификатор процесса. GID — идентификационный номер основной группы пользователя.
6. Задачи - запущенные фоном программы. Ими можно управлять с помощью команды `jobs`.
7. Команды `top` и `htop` показывают информацию о процессах в реальном времени, выводят данные о потреблении системных ресурсов и позволяют искать, останавливать и управлять процессами.
8. Поиск файлов производится с помощью команды `find`. Пример:

```
$ find ~ -name "f*" -print
```

9. Найти файл по контексту можно с помощью следующей команды:

```
$ grep -rn '<путь к папке>' -e "шаблон"
```

10. Узнать свободное место на жестком диске можно с помощью следующей команды:

```
$ df /
```

11. Объем домашнего каталога можно определить с помощью следующей команды:

```
$ df ~
```

12. Удалить зависший процесс можно с помощью следующей команды:

```
$ kill <PID процесса>
```

5 Выводы

- Я изучил инструменты поиска файлов и фильтрации текстовых данных.
- Я научился управлять процессами.
- Я научился проверять использование диска и обслуживать файловые системы.