

Введение

Стандартные потоки ввода и вывода в Linux являются наиболее распространенным средством обмена информацией для процессов. С помощью перенаправления ввода/вывода можно перенаправить вывод одной программы в файл или стандартный ввод другой программы. К примеру, можно перенаправить список всех процессов (`ps -ax`) команде `grep`, которая найдет в списке интересующий вас процесс.

Потоки

Стандартный ввод при работе пользователя в терминале передается с использованием клавиатуры.

Стандартный вывод и стандартная ошибка отображаются на дисплее терминала пользователя в виде текста.

Ввод и вывод разделен на 3 стандартных потока:

- `stdin` - стандартный ввод (клавиатура)
- `stdout` - стандартный вывод (экран)
- `stderr` - стандартная ошибка (вывод ошибок на экран)

Потоки пронумерованы:

- `stdin` - 0
- `stdout` - 1
- `stderr` - 2

Из стандартного ввода команда может только считывать данные, а два других потока могут использоваться только для записи. Данные выводятся на экран и считываются с клавиатуры, так как стандартные потоки по умолчанию ассоциированы с терминалом пользователя. Потоки можно подключать к файлам, программам и устройствам. В командном интерпретаторе такая операция называется перенаправлением:

- **< file** – Использовать файл как источник данных для стандартного потока ввода
- **> file** - Направить стандартный поток вывода в файл. Если файл не существует, он будет создан, если существует — перезаписан
- **2> file** - Направить стандартный поток ошибок в файл. Если файл не существует, он будет создан, если существует — перезаписан
- **>>file** - Направить стандартный поток вывода в файл. Если файл не существует, он будет создан, если существует — данные будут дописаны в конец файла
- **2>>file** - Направить стандартный поток ошибок в файл. Если файл не существует, он будет создан, если существует — данные будут дописаны в конец файла
- **&>file** или **>&file** - Направить стандартный поток вывода и стандартный поток ошибок в файл. Другая форма записи: **>file 2>&1**

Стандартный ввод

Стандартный входной поток обычно переносит данные от пользователя к программе. Стандартный ввод прекращается по достижении конца файла (EOF), который указывает на то, что данных для чтения больше нет.

EOF вводится нажатием сочетания клавиш Ctrl+D.

Рассмотрим работу со стандартным выводом на примере команды **cat**.

Cat отправляет полученные входные данные на дисплей терминала в качестве стандартного вывода и останавливается после того как получает EOF.

Пример

Введите команду **cat** без аргументов:

```
cat
```

В открывшейся строке введите, например, “a” и нажмите клавишу Enter. На дисплей выводится “a”. Введите “b” и нажмите клавишу Enter. На дисплей выводится “b”.

Дисплей терминала выглядит следующим образом:

```
vbox@user:~/example$ cat
a
a
b
b
```

Для завершения ввода данных следует нажать сочетание клавиш Ctrl + D.

Стандартный вывод

Стандартный вывод записывает данные, сгенерированные программой.

При использовании без каких-либо дополнительных опций, команда **echo** выводит на экран любой аргумент, который передается ему в командной строке:

```
echo Test
```

Аргументом является то, что получено программой, в результате на дисплей терминала будет выведено:

```
Test
```

При выполнении **echo** без каких-либо аргументов, возвращается пустая строка.

Пример

Команда объединяет три файла: file1, file2 и file3 в один файл complete_file:

```
cat file1 file2 file3 > complete_file
```

Команда cat по очереди выводит содержимое файлов, перечисленных в качестве параметров на стандартный поток вывода. Стандартный поток вывода перенаправляется в файл complete_file.

Стандартная ошибка

Стандартная ошибка записывает ошибки, возникающие в ходе исполнения программы.

Рассмотрим пример стандартной ошибки с помощью команды **ls**, которая выводит список содержимого директорий.

При запуске без аргументов **ls** выводит содержимое текущей директории.

Введем команду **ls** с директорией % в качестве аргумента:

```
ls %
```

В результате должно выводиться содержимое соответствующей директории. Но так как директории % не существует, на дисплей терминала будет выведен следующий текст стандартной ошибки:

```
ls: cannot access %: No such file or directory
```

Перенаправление потока

Команды со знаками **>** или **<** означают перезапись существующего содержимого файла:

- **>** - стандартный вывод
- **<** - стандартный ввод
- **2>** - стандартная ошибка

Команды со знаками **>>** или **<<** не перезаписывают существующее содержимое файла, а присоединяют данные к нему:

- * **>>** - стандартный вывод
- * **<<** - стандартный ввод
- * **2>>** - стандартная ошибка

В приведенном примере команда **cat** используется для записи в файл **file**:

```
cat > file
a
b
c
```

Для завершения записи нажмите сочетание клавиш **Ctrl + D**.

Если файла **file** не существует, то в текущем каталоге создается новый файл с таким именем.

Для просмотра содержимого файла **file** введите команду:

```
cat file
```

В результате на дисплей терминала, должно быть выведено следующее:

```
a
b
c
```

Для перезаписи содержимого файла введите следующее:

```
cat > file  
a  
б  
в
```

Для завершения цикла нажмите сочетание клавиш Ctrl + D.

В результате на дисплей терминала, должно быть выведено следующее:

```
a  
б  
в
```

Предыдущего текста в текущем файле больше не существует, так как содержимое файла было переписано командой **>**.

Для добавления нового текста к уже существующему в файлу (**>>**) выполните команду:

```
cat >> file  
a  
b  
c
```

Для завершения цикла нажмите сочетание клавиш Ctrl + D.

Откройте file снова и в результате на дисплее монитора должно быть отражено следующее:

```
a  
б  
в  
a  
b  
c
```

Каналы

Каналы – способ подключения вывода одной программы на вход другой, без каких бы то ни было временных файлов; а конвейер – это соединение двух или более программ посредством каналов. Данные первой программы, которые получает вторая программа, не будут отображаться. На дисплей терминала будут выведены только отфильтрованные данные, возвращаемые второй командой.

Пример

Введите команду:

```
ls | less
```

В результате каждый файл текущей директории будет размещен на новой строке:

```
f1  
f2
```

Перенаправлять данные с помощью каналов можно как из одной команды в другую, так и из одного файла к другому, а перенаправление с помощью `>` и `>>` возможно только для перенаправления данных в файлах.

В конвейер можно включить любое количество программ

```
who | grep mary | wc -l
```

Данный конвейер подсчитывает, сколько раз Мэри зарегистрирована в системе.

В действительности все программы конвейера выполняются одновременно, а не одна за другой. Это означает, что программа конвейера может быть и интерактивной; ядро следит за тем, как правильно обеспечить синхронизацию и диспетчеризацию, чтобы все работало.

Созданием каналов занимается оболочка; отдельные программы не имеют понятия о перенаправлении ввода-вывода. Конечно, для того чтобы иметь возможность совместной работы, программы должны придерживаться определенных соглашений. Большинство команд имеет одинаковый дизайн, так что они подходят для любой позиции в конвейере. Обычно вызов программы выглядит следующим образом:

команда необязательные-аргументы необязательные-имена-файлов

Если имена файлов не указаны, то команда читает свой стандартный ввод, который по умолчанию является терминалом, но может перенаправляться и поступать из файла или канала. Что же касается вывода, большая часть команд направляет его на стандартный вывод, который по умолчанию связан с терминалом, но и он также может быть перенаправлен в файл или канал.

А вот сообщения об ошибках исполнения команд должны обрабатываться по-другому, иначе они могут исчезнуть в файле или канале. Поэтому у каждой команды есть еще и стандартный вывод ошибок, который обычно направляется на терминал.

Задание

1. Есть файл, example.txt, содержащий следующие слова:

Bat
Goat
Apple
Dog
First
Eat
Hide

Необходимо вывести на экран строки содержащие символ "a", но в обратном порядке.

2. Опишите, что делает команда: `ls -y >> logfile 2>&1`
3. Напишите сценарий позволяющий вывести N строк файла начиная с заданной строки. Номер заданной строки и их количество являются входными параметрами. Задачу следует решать, используя стандартные утилиты и перенаправление ввода вывода.
4. Дана команда: `ls file1 file2`. Файл file1 существует, файл file2 не существует. Перенаправьте **только** стандартную ошибку в конвейер, одновременно записывая номера строк и перенаправьте вывод в файл.