

# Перенаправление ввода-вывода Linux

**Выполнил/ ла:**

**Студент/ка группы:** НПИбд-02-21

**Студенческий билет :**1032205421

**Студент /ка:**Стелина Петрити

## Потоки

Ввод и вывод в среде Linux распределяются по трем потокам. Эти потоки являются:

- **Стандартный ввод**
- **Стандартный Вывод**
- **Стандартная ошибка**

Потоки также пронумерованы:

- `stdin` (0)
- стандартный вывод (1)
- `stderr` (2)

о время стандартных взаимодействий между пользователем и терминалом стандартный ввод передается через клавиатуру пользователя. Стандартный вывод и стандартная ошибка отображаются на терминале пользователя в виде текста. В совокупности эти три потока называются стандартными потоками.

### • **Стандартный ввод**

Стандартный входной поток обычно передает данные от пользователя в программу.

Программы, которые ожидают стандартного ввода, обычно получают ввод с устройства, такого как клавиатура. Стандартный ввод завершается достижением EOF (end-of-file). Как следует из его названия, EOF указывает, что больше нет данных для чтения.

Чтобы увидеть стандартный ввод в действии, запустите программу `cat`. `cat` расшифровывается как `concatenate`, что означает связывать или объединять что-либо. Обычно он используется для объединения содержимого двух файлов. При самостоятельном запуске `cat` открывает циклическую подсказку.

```
$ cat
```

После открытия `cat` введите серию цифр во время его выполнения.

```
1
```

```
2
```

```
3
```

```
ctrl -d
```

Когда вы вводите число и нажимаете `enter`, вы отправляете стандартный ввод в запущенную программу `cat`, которая ожидает указанный ввод. В свою очередь, программа `cat` отправляет ваш ввод обратно на дисплей терминала в качестве стандартного вывода.

EOF может быть введен пользователем нажатием `ctrl-d`. После того, как программа `cat` получит EOF, она останавливается

## • Стандартный Вывод

Стандартный вывод записывает данные, сгенерированные программой. Когда стандартный выходной поток не перенаправляется, он выводит текст на терминал. *echo standard output*  
При использовании без каких-либо дополнительных опций команда `echo` отображает любой аргумент, который передается ей в командной строке. Аргумент - это то, что принимается программой.

```
$ echo
```

он вернет пустую строку, так как в ней нет аргументов.

## • Стандартная ошибка

Стандартная ошибка записывает ошибки, сгенерированные программой, которая завершилась сбоем в какой-то момент ее выполнения. Как и стандартный вывод, назначением по умолчанию для этого потока является дисплей терминала.

Когда стандартный поток ошибок программы передается по каналу во вторую программу, передаваемые по каналу данные (состоящие из программных ошибок) также одновременно отправляются на терминал.

Давайте рассмотрим базовый пример стандартной ошибки с помощью команды `ls`. `ls` перечисляет содержимое каталога.

При запуске без аргумента `ls` выводит список содержимого в текущем каталоге. Если `ls` запускается с каталогом в качестве аргумента, он отобразит содержимое предоставленного каталога.

```
$ ls %
```

Поскольку `%` не является существующим каталогом, это приведет к отправке следующего текста в стандартную ошибку:

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

Перенаправление потока Linux включает в себя команды перенаправления для каждого потока. Эти команды записывают стандартный вывод в файл. Если выбран несуществующий файл (с помощью команды с одинарными или двойными скобками), перед записью будет создан новый файл с таким именем.

Команды с одной скобкой перезаписывают существующее содержимое назначения.

Переписывать

> - стандартный выход

< - стандартный ввод

2> - стандартная ошибка

Команды с двойными скобками не перезаписывают существующее содержимое назначения.

Добавлять

>> - стандартный вывод

<< - стандартный ввод

2>> - стандартная ошибка

Давайте посмотрим на пример:

```
*$ cat > write.txt
```

*a*

*b*

*c*

Здесь cat используется для записи в файл, который создается в результате цикла.

Просмотреть содержимое write.txt использование *cat:cat write.txt*

Он должен иметь следующее содержание:*a b c*

Перенаправить его на write.txt еще раз и введите три цифры.

1

2

3

Когда вы используете cat для просмотра write.txt , вы увидите следующее:

1

2

3

Предыдущее содержимое больше не существует, так как файл был перезаписан командой с одной скобкой.

Выполняем еще одно перенаправление cat, на этот раз с использованием двойных скобок

```
:cat >> write.txt
```

*a*

*b*

*c*

ctrl -d

Открыть write.txt еще раз, и вы увидите это

1

2

З

а

б

с

Файл теперь содержит текст из обоих вариантов использования cat, так как второй вариант не переопределял первый

## Вывод

Теперь мы узнали, что одной из самых важных и интересных тем в администрировании Linux является перенаправление ввода-вывода. Изучение того, как использовать возможности перенаправления, встроенные в командную строку Linux, может быть немного сложным. Теперь, когда мы ознакомились с основами работы перенаправлений, нам будет проще продолжить изучение других аспектов файловой системы.

## Источник

<https://tldp.org/LDP/abs/html/io-redirection.html>

<https://www.digitalocean.com/community/tutorials/an-introduction-to-linux-i-o-redirection>

[https://www.tecmint.com/linux-io-input-output-redirection-](https://www.tecmint.com/linux-io-input-output-redirection-operators/#:~:text=One%20of%20the%20most%20important,as%20a%20%E2%80%9Ccommand%20pipeline%E2%80%9D.)

[operators/#:~:text=One%20of%20the%20most%20important,as%20a%20%E2%80%9Ccommand%20pipeline%E2%80%9D.](https://www.tecmint.com/linux-io-input-output-redirection-operators/#:~:text=One%20of%20the%20most%20important,as%20a%20%E2%80%9Ccommand%20pipeline%E2%80%9D.)