

Потоки ввода/вывода (или потоки `stdin`, `stdout`, `stderr`) представляют собой концепцию в UNIX-подобных операционных системах, таких как Linux. Эти потоки являются абстракциями для обмена данными между процессами и средой выполнения программы.

1. **Стандартный ввод (`stdin` - File Descriptor 0):**

- **Описание:** Поток, через который программа получает вводные данные.
- **Пример:** Ввод с клавиатуры, данные из файла.

2. **Стандартный вывод (`stdout` - File Descriptor 1):**

- **Описание:** Поток, через который программа отправляет выводные данные.
- **Пример:** Вывод на экран (консоль), запись в файл.

3. **Стандартный поток ошибок (`stderr` - File Descriptor 2):**

- **Описание:** Поток, через который программа отправляет сообщения об ошибках.
- **Пример:** Вывод сообщений об ошибках на экран или в файл.

Пример использования потоков в командной строке Linux:

• **Поток ввода (`stdin`)**

передача содержимого файла **input.txt** в качестве входных данных (`stdin`) для команды **command1**

```
command1 < input.txt
```

• **Поток вывода (`stdout`)**

вывод команды **command2** не выводится на экран, а направляется в файл **output.txt**

```
command2 > output.txt
```

• **Поток ошибок (`stderr`)**

перенаправление стандартного потока ошибок (`stderr`) команды **command3** в файл с именем **error.txt**

```
command3 2> error.txt
```

• **Перенаправление ввода/вывода**

command4 читает данные из файла **input.txt** в качестве входных данных. Вывод команды **command4** записывается в файл **output.txt**. Сообщения об ошибках, которые генерирует **command4**, записываются в файл **error.txt**

```
command4 < input.txt > output.txt 2> error.txt
```

• **Потоки в фоновом режиме:**

- `command5 &` # выполнять команду в фоновом режиме

Команды для работы с потоками:

|: Этот символ (вертикальная черта) представляет собой оператор конвейера (pipeline), который передает вывод одной команды в качестве ввода для другой.

- `cat file.txt | grep "pattern"`

Потоки обеспечивают мощный механизм для взаимодействия процессов и управления вводом/выводом данных в командной строке. Эти концепции широко используются при написании сценариев командной строки, скриптов и в программировании в целом