УРОК 3. PIPE AND REDICTION

СТАНДАРТНЫЕ ПОТОКИ ВВОДА/ВЫВОДА

Рассмотрим основные виды потоков данных.

- Стандартные потоки ввода/вывода (stdin, stdout, stderr)
- Потоки данных между процессами

Стандартный поток ввода. В системе это — поток №0.
Этот поток представляет собой некую информацию, передаваемую в терминал, в частности — инструкции, переданные в оболочку для выполнения. Обычно данные в этот поток попадают в ходе ввода их пользователем с клавиатуры.

stdin (стандартный ввод): Поток данных, через который программа может принимать ввод от пользователя или из других программ.

Стандартный поток вывода (standard output), ему присвоен №1.
Это поток данных, которые оболочка выводит после выполнения каких-то действий. Обычно эти данные попадают в то же окно терминала, где была написана команда, вызвавшая их появление.
stdout (стандартный вывод): Поток данных, по которому программа выводит результат своей работы. Этот вывод может быть направлен в файл, передан другой программе или отображен в консоли.

Стандартный поток ошибок (standard error), он имеет дескриптор №2.
Этот поток похож на стандартный поток вывода, так как обычно то, что в него попадает, оказывается на экране терминала.
stderr (стандартный вывод ошибок): Поток данных, используемый для вывода сообщений об ошибках. Как и stdout, stderr может быть перенаправлен в файл или передан другой программе.

*Номера потоков называют дескрипторами.
Дескриптор — ключевое слово, характеризующее блок информации.

ОБРАБОТКА КОМАНД В LINUX

Рассмотрим, как происходит обработка команд в Linux.

Когда вы вводите команду в терминале Линукса, происходит следующий процесс обработки типичной команды, например, команды date:

Разбор команды:
- Когда вы вводите date, оболочка (shell) разбирает эту строку на имя команды (date) и ее аргументы (в данном случае, отсутствуют).

Поиск команды:
- Оболочка ищет исполняемый файл, связанный с этой командой, в соответствии с переменной PATH. Переменная PATH содержит пути к директориям, в которых операционная система будет искать исполняемые файлы.

Выполнение команды:
- Когда оболочка находит исполняемый файл для команды date, она создает новый процесс и запускает этот файл внутри нового процесса.

Выполнение команды date:
- Процесс, созданный для выполнения команды date, выполняет код команды. В случае date, это вывод текущей даты и времени в соответствии с настройками системы.

Вывод результата:
- Результат выполнения команды date выводится в стандартный вывод (stdout), который по умолчанию отображается в терминале.

Завершение процесса:
- После выполнения команды процесс завершается, и управление возвращается оболочке.

Выполняем команду date.

Смотрим содержимое переменной PATH.

При помощи команды which убеждаемся, что исполняемый файл команды находится по одному из путей, описанных в переменной PATH (через двоеточие мы видим /bin в /usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin)


localhost:~# date
Tue Jan 23 00:56:47 UTC 2024
localhost:~# echo $PATH
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
localhost:~# which date
/bin/date

ПЕРЕНАПРАВЛЕНИЕ

Перенаправление позволяет изменять потоки данных ввода/вывода для команд.

Делается это с использованием символов > и < в различных комбинациях, применение которых зависит от того, куда, в итоге, должны попасть перенаправляемые данные.
Виды перенаправлений :

- Запись, дозапись в файл,
- Вычитывание из файла,
- Комбинированное перенаправление,
- Перенаправление потока ошибок

Перенаправление вывода в файл (т.н запись и дозапись или перенаправление и двойное перенаправление).

Пример: Запишем текущую дату в файл

```
date > current_date.txt
```

Пример: Добавим (дозапишем) еще один раз дату в тот же документ (если файл существует)

```
date >>current_date.txt
```

Командой cat мы выводим содержимое файла в стандартный вывод:

```
localhost:~# date > current_date.txt
localhost:~# cat current_date.txt
Tue Jan 23 01:25:19 UTC 2024
localhost:~# date >> current_date.txt
localhost:~# cat current_date.txt
Tue Jan 23 01:25:19 UTC 2024
Tue Jan 23 01:25:41 UTC 2024
localhost:~#
```

Перенаправление ввода из файла.

Пример: использование файла current_date.txt в качестве ввода для команды cat :

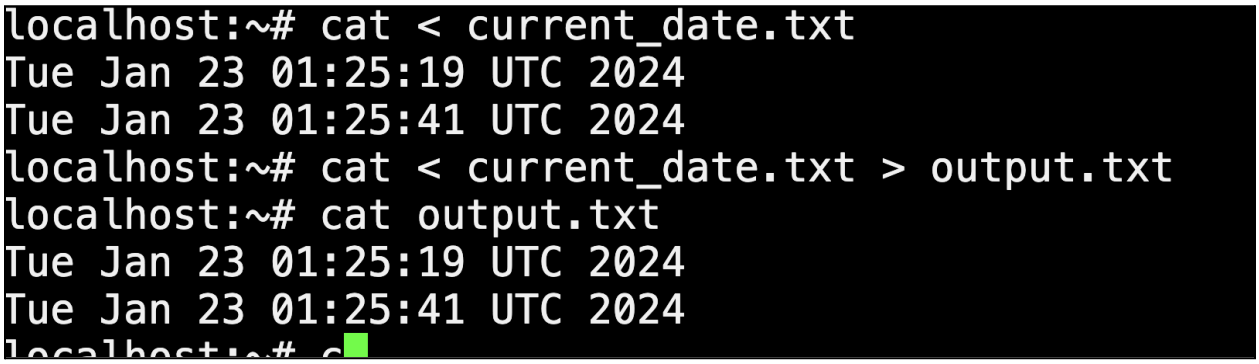
```
cat < current_date.txt
```

Комбинированное перенаправление.

Пример: считывание ввода из файла current_date.txt и запись вывода в файл output.txt

```
cat < current_date.txt > output.txt
```

Проверяем:



Перенаправление стандартных ошибок.

Стандартные ошибки (stderr) можно перенаправлять отдельно от стандартного вывода.

```
command > output_error.txt 2>&1
```

В этом примере 2>&1 означает, что стандартная ошибка перенаправляется так, как стандартный вывод.

ЗАДАНИЕ ДЛЯ ЗАКРЕПЛЕНИЯ

1. Как вы поняли, чем отличается запись от дозаписи?
2. Какие бывают потоки ввода-вывода данных?

PIPING

Программы могут обмениваться данными через потоки при помощи механизмов, таких как каналы (pipes). Каналы позволяют передавать вывод одной программы в качестве ввода другой программе.
Это обеспечивает эффективное взаимодействие между программами, так как данные могут передаваться напрямую, без сохранения на диске.

Символ для создания канала в командной строке Линукса - это вертикальная черта **|**

Используя пайпинг можно создавать сложных цепочки команд, где каждая последующая команда обрабатывает вывод предыдущей. Это позволяет строить мощные и гибкие конвейеры для обработки данных в командной строке.

Команды , которые помогут нам понять перенаправления и пайпинг:

- cat - вывести файл в стандартный вывод
- wc - подсчет количества строк, слов и байт в текстовом файле или выводе текстовой команды

Некоторые из основных опций команды wc:

  - -l: Выводит только количество строк.
  - -w: Выводит только количество слов.
  - -c: Выводит только количество байт.

- head - вывод определенного количество строк с начала документа
  По умолчанию - 10 строк, количество можно менять через ключ -n

- tail - вывод определенного количество строк с конца документа

- echo - вывод текста или значение переменной. Аналог print из любого языка программирования.

- history - вывод истории команд

Команда	Пояснение
cat /etc/group	Выводит на экран все, что есть в файле /etc/group
cat /etc/group | wc -l	Посчитает количество строк в этом файле. Тут воспользуемся pipe, который перенаправит результат вывода всех данных в файле на новую команду, которая посчитает количество всех строк.
echo “Hello world”	Вернет Hello world в стандартный вывод

Перенаправление можно использовать и в обратном направлении, но чаще используют первый вариант.
wc -l < /etc/group

Команда	Пояснение
cat /etc/group | head	Выведет на экран первые 10 строк из файла /etc/group. По умолчанию - 10 строк, но количество можно менять через ключ -n
cat /etc/group | tail	tail также по умолчанию показывает 10 строк, но из конца файла
history | tail -3	Покажет последние три команды из истории.
history | tail -3 >> /tmp/file	Дозаписали в файл эти три строки из истории, используя >>