

Лабораторная работа. Перенаправление стандартных потоков ввода/вывода.

Задачи

1. Получение навыков перенаправления потоков стандартного ввода/вывода и потока ошибок
2. Объединение команд с применением конвейера

Описание

Ниже приведены команды для работы в ОС Linux в консольном режиме. Изучите приведенные команды и отработайте их в командной строке Linux. При выполнении работы проявите творчество и поэкспериментируйте с командами.

Перенаправление потока вывода STDOUT и потока ошибок STDERR

Используя символ > вы можете перенаправить стандартный вывод STDOUT.

```
username@linux-pc:~$ echo "Hello World"
username@linux-pc:~$ echo "Hello World" > mymessage
username@linux-pc:~$ cat mymessage
```

При этом информация в файле перезаписывается.

```
username@linux-pc:~$ echo "Greetings" > mymessage
username@linux-pc:~$ cat mymessage
```

Вы можете избежать затирания файла используя >> вместо >

```
username@linux-pc:~$ cat mymessage
username@linux-pc:~$ echo "How are you?" >> mymessage
username@linux-pc:~$ cat mymessage
```

Команда find, запускаемая от прав непривилегированного пользователя, хороший пример для демонстрации перенаправления потока ошибок STDERR.

```
username@linux-pc:~$ find /etc -name hosts
```

Собрать все ошибки в отдельный файл и не выводить их на экран поможет команда:

```
username@linux-pc:~$ find /etc -name hosts 2> err.txt
username@linux-pc:~$ cat err.txt
```

Вы можете перенаправить STDOUT и STDERR в разные файлы

```
username@linux-pc:~$ find /etc -name hosts > std.out 2> std.err
username@linux-pc:~$ cat std.err
username@linux-pc:~$ cat std.out
```

Для перенаправления STDOUT и STDERR в один файл используйте сначала символ > , а затем 2>&1

```
username@linux-pc:~$ find /etc -name hosts >find.out 2>&1
```

Или просто &>

```
username@linux-pc:~$ find /etc -name hosts &> find.out
```

```
username@linux-pc:~$ cat find.out
```

Перенаправление потока ввода STDIN

Для демонстрации перенаправления потока ввода STDIN мы будем использовать команду tr (трансформации символов).

Команда tr позволяет преобразовать одни символы в другие. Для выхода из программы необходимо нажать Ctrl+D

```
username@linux-pc:~$ tr a-z A-Z
```

this is interesting

how do I stop this?

^D

Вы можете использовать перенаправление вывода STDOUT для получения файла с маленькими символами.

```
username@linux-pc:~$ tr A-Z a-z > myfile
```

Wow, I SEE NOW

This WORKS!

Для преобразования имеющегося файла можно использовать перенаправление ввода STDIN.

```
username@linux-pc:~$ cat myfile
```

```
username@linux-pc:~$ tr a-z A-Z < myfile
```

Конвейер команд

Другим видом частого перенаправления является ситуация, когда вывод одной команды перенаправляется в качестве входных данных для другой.

Так, например, большой вывод команды ls может быть просмотрен постранично командой more.

```
username@linux-pc:~$ ls -l /etc | more
```

Команда cut используется для вывода части файла, используя разделители.

Например, файл passwd содержит символ : для разделения полей. Мы можем, используя команду cut, вывести только одно поле из всего файла.

Например, выведем список всех пользователей из файла /etc/passwd

```
username@linux-pc:~$ cut -d: -f1 /etc/passwd
```

Опция -d задает разделитель, а опция -f указывает номер поля для вывода.

Как мы видим, список пользователей выведен по мере их добавления в файл (в соответствии с их идентификаторами).

Для упорядочивания списка можно перенаправить вывод команды `cut` на вход команды `sort`.

```
username@linux-pc:~$ cut -d: -f1 /etc/passwd | sort
```

Можно использовать несколько перенаправлений подряд. Например, предыдущий вывод команды `cut`, отсортированный командой `sort` можно вывести постранично командой `more` или `less`.

```
username@linux-pc:~$ cut -d: -f1 /etc/passwd | sort | more
```