



Практика в консоли ч.4



Редирект

- Включаем VM, логинимся, создаем копию файла passwd:

```
cp /etc/passwd users.txt
```

- Выводим файл

```
cat users.txt
```

- Альтернативно можно использовать stdin

```
cat < users.txt
```



Редирект

- Часто мы перенаправляем вывод в файл:

```
ls -l > dir-list.txt
```

- Проверим что там получилось:

```
cat dir-list.txt
```

- Окажется, что наш файл также является частью списка, но с размером 0 байт. Объяснение простое – перед выполнением команды выполняется перенаправление.
- Если мы выполним приведенную выше команду несколько раз, мы заметим, что каждый раз файл создается заново, а существующее содержимое теряется. Это не всегда то, чего мы хотим.



Редирект

- Изменим команду в:

```
ls -l >> dir-list.txt
```

- Проверим что там получилось:

```
cat dir-list.txt
```

- Лучше! Но есть более удачный способ избежать пересоздания файла:

```
set -o noclobber
```

```
ls -l > dir-list.txt
```

- И... нам не дают перезапустить файл!



Редирект

- Вернем поведение по умолчанию:

```
set +o noclobber
```

- Мы можем редиректить и ошибки, к примеру дает ошибку:

```
ls non-existing-file.txt
```

- Перенаправим ее проверим:

```
ls non-existing-file.txt 2> ls-error.txt
```

```
cat ls-error.txt
```



Редирект

- К примеру мы можем редиректить как stdout так и stderr в разные файлы

```
ls -l ~ /missing > ls-stdout.txt 2> ls-stderr.txt
```

- Проверим:

```
cat ls-stdout.txt
```

```
cat ls-stderr.txt
```



echo

- Создавать файлы с редиректом можно и через echo:

```
echo 'Hello World!' > hello.txt
```

- Проверим:

```
cat hello.txt
```

- К примеру можно сохранить результат команды и фразу

```
echo "Current date is" date > date.txt
```

```
cat date.txt
```



echo

- Попались! Все дело в том что команду необходимо написать так:

```
echo "Current date is" $(date) > date.txt
```

- Проверим:

```
cat date.txt
```

- Альтернативно можно так:

```
echo "Current date is" `date` > date.txt
```

```
cat date.txt
```




echo

- С кавычками очень внимательно, подставляя команды, к примеру:

```
echo "Current date is `date`" > date.txt
```

- Проверим:

```
cat date.txt
```

- Но, если так (одиночные), подставление не работает:

```
echo 'Current date is `date`' > date.txt
```

```
cat date.txt
```



Серия команд

Как мы уже знаем, мы можем выполнить серию команд за одну итерацию. Один из возможных способов выполнить эту задачу - разделить их точкой с запятой. Таким образом, каждая команда будет выполнена, независимо от того, как закончилась предыдущая – успешно или с ошибкой:

```
ls > ls-seq.txt ; cat missing-file.txt ; cat ls-seq.txt
```



Серия команд

Изменим последовательность команд таким образом, чтобы каждая следующая команда выполнялась только в том случае, если предыдущая была успешно выполнена:

```
ls ls* && ls users.txt && ls missing.txt && echo "Not executed"
```



Серия команд

Изменим поведение, чтобы каждая следующая команда выполнялась только в том случае, если предыдущая привела к ошибке:

```
ls ls* || ls users.txt || ls missing.txt || echo "Not executed"
```



Есть вопросы?

miitqa@outlook.com