

Условия

С записью условий в Bash связано огромное количество тонкостей и подводных камней. В курсе я рассмотрю только основы, а в дебри Вы уже сможете залезть самостоятельно (при желании).

Начнём с того, что в Bash нет логического (булева) типа данных. Проверка условий основана на кодах завершения команд. Так, если команда вернула код завершения 0, то это означает, что ошибок не было, и условие считается выполненным. Рассмотрим пример:

```
#!/bin/bash

file="some_file"

if rm "$file" 2>/dev/null
then
    echo "Файл $file успешно удалён."
else
    echo "Не удалось удалить файл $file. Код завершения команды rm: $?"
fi
```

Здесь выполнение ветви `then` или `else` зависит от того, удалось ли удалить файл. Запись `2>/dev/null` просто убирает отображение сообщений об ошибках команды `rm`, чтобы они не мешали в нашем примере.

В системе Linux присутствуют две интересные команды с именами `true` и `false`. Первая всегда завершается с нулевым кодом возврата (то есть успешно), а вторая — с ненулевым. Как Вы уже вероятно догадались, их можно использовать аналогично булевым значениям `true` и `false` в других языках программирования. Пример:

```
#!/bin/bash

if true
then
    echo "Это будет напечатано."
else
    echo "А это напечатано не будет."
fi
```

А как же тогда записать, например, сравнение чисел или строк? Об этом далее.