


Слова...

 Эта статья нуждается в обзоре, она охватывает две темы (разделение командной строки и разделение слов) и слишком сильно смешивает обе. Но в целом, это все еще можно использовать, чтобы помочь понять это поведение, это "неправильно, но не неправильно".

Одним из фундаментальных принципов Bash является распознавание слов, введенных в командной строке или при других обстоятельствах, таких как расширение переменной.

Разделение командной строки

Bash сканирует командную строку и разбивает ее на слова, обычно для того, чтобы поместить параметры, которые вы вводите для команды, в нужную C-память (argv вектор), чтобы позже правильно вызвать команду. Эти слова распознаются путем разделения командной строки на позиции специального символа, **пробела** или **табуляции** (в руководстве они определены как **пробелы**). Для примера возьмем программу echo. Он отображает все свои параметры, разделенные пробелом. Когда вы вводите команду echo в командной строке Bash, Bash будет искать эти специальные символы и использовать их для разделения параметров.

Ты не понимаешь, о чем я говорю? Я говорю об этом:

```
$ echo Привет, маленький мир
Привет, маленький мир
```

Другими словами, то, что вы делаете (и Bash делает) каждый день. Символы, в которых Bash разделяет командную строку (ПРОБЕЛ, табуляция, т.е. Пробелы), распознаются как разделители. При наличии 2 или более пробелов в командной строке аргумент null не генерируется. **Последовательность из большего количества пустых символов обрабатывается как один пробел.** Вот пример:

```
$ echo Привет, маленький мир
Привет, маленький мир
```

Bash разбивает командную строку на пробелы на слова, затем вызывает echo с **каждым словом в качестве аргумента**. В этом примере echo вызывается с тремя аргументами: "hello ", " little " и " world "!

Означает ли это, что мы не можем повторять более одного пробела? Конечно, нет! Bash обрабатывает пробелы как специальные символы, но есть два способа сказать Bash, чтобы он не относился к ним по-особому: **экранирование** и **цитирование**.

После того, как командная строка была разделена на слова, Bash выполнит расширение, если это необходимо - переменные, которые встречаются в командной строке, должны быть расширены (заменены их значением), например. Здесь возникает второй тип разделения слов - несколько расширений подвергаются **разделению слов** (но другие этого не делают).

Представьте, что у вас есть имя файла, хранящееся в переменной:

```
MYFILE="тестовый файл"
```

Когда используется эта переменная, ее значение будет заменено ее содержимым.

```
$ cat $MYFILE
cat: тест: нет такого файла или каталога
cat: file: нет такого файла или каталога
```

Хотя это еще один шаг, на котором пробелы усложняют задачу, **цитирование** используется для решения проблемы. Кавычки также влияют на разделение слов:

```
$ cat "$MYFILE"
m00!
```

Пример

Давайте выполним команду без кавычек, выполнив следующие шаги, предполагая, что переменная установлена:

```
MYFILE="THE FILE.TXT "
```

и первый обзор:

```
echo Файл называется $MYFILE
```

Анализатор будет сканировать пробелы и отмечать соответствующие слова ("разделение командной строки"):

Начальное разделение командной строки:

Слово 1	Слово 2	Слово 3	Слово 4	Слово 5	Слово 6
echo	The	file	is	named	\$MYFILE

Расширение параметра / переменной является частью этой командной строки, Bash выполнит замену и разбиение слов на результаты:

Разделение слов после замены:

Слово 1	Слово 2	Слово 3	Слово 4	Слово 5	Слово 6	Слово 7
echo	The	file	is	named	THE	FILE.TXT

Теперь давайте представим, что мы процитировали \$MYFILE , командная строка теперь выглядит так:

```
echo файл называется "$MYFILE"
```

Разделение слов после замены (в кавычках!):

Слово 1	Слово 2	Слово 3	Слово 4	Слово 5	Слово 6
echo	The	file	is	named	THE FILE.TXT

Жирный текст 72 я люблю этот мир

Смотрите также

- Внутренние: цитирование и экранирование символов
- Внутренний: разделение слов
- Внутренний: введение в расширения и замены
- Внешнее: Grymore: Цитирование шелла (<http://www.grymoire.com/Unix/Quote.html>)

Обсуждение

кобу, [07.02.2012 21:25 \(\)](#)

пробел, табуляция И перевод строки ... не забывайте перевод строки

Rptx, [2014/06/05 03:39 \(\)](#)

Разве расширение параметров не происходит перед разделением слов?
Последняя часть неверна. \$MYFILE будет расширен перед разделением слов.

Ян Шампера, [2015/06/18 04:25 \(\)](#)

Но кавычки там, чтобы сохранить границы слов (которые точно описывают работу с цитированием)

This site is supported by Performing Databases - your experts for database administration

Bash Hackers Wiki



Except where otherwise noted, content on this wiki is licensed under the following license:
GNU Free Documentation License 1.3