

## 1. Вставьте последний параметр предыдущей строки

Начнем команды `bash` с простого. `alt-.`  – самая полезная комбинация: попробуйте и убедитесь в этом. По каким-то причинам мало кто ней не знает. Используйте ее, чтобы выбрать более старые параметры. Она просто незаменима если вы хотите сделать что-то другое с тем, что использовали недавно.

## 2. Продвинутые команды bash: работа после выхода из системы

Если вы хотите, чтобы процесс продолжался даже после выхода из системы, используйте это:

```
1 disown -h
```

Полезная встроенная функция. В отличие от [nohup](#), вы можете использовать `disown` в уже запущенном процессе. Прекратите работу с помощью `control-Z`, получите `pid` из `ps` (или используйте `echo $!`), используйте `bg` для отправки в фоновый режим, затем используйте `disown` с флагом `-h`. Не забудьте о работе в фоне, иначе все прекратится после выхода.

## 3. Почти все перечисленное в секции EXPANSION

В частности расширение параметра:

```
1
2 $ I=foobar $ echo ${I/oo/aa} #replacement faabar $ echo ${I:1:2} #substring oo $
3 echo ${I%bar} #trailing substitution foo $ echo ${I#foo} #leading substitution bar
```

## 4. Перезапуск

Команды `bash` также включают в себя перезапуск предыдущей команды:

```
1
2 sudo !!
3
```

## 5. Больше волшебных комбинаций:

- `Ctrl + r` начинает обратный инкрементный поиск по истории введенных команд. После того, как вы продолжите ввод, она выдаст последнюю команду, содержащую то, что вы ввели.
- `Tab` закончит введенное слово, если оно однозначно.
- `Tab Tab` выведет все возможные варианты конца введенного слова.
- `Alt + *` вставит все возможные окончания, которые могут быть полезны, например, если вы только что ввели деструктивную команду с символами подстановки.

```
1
2 rm -r source/d*.c Alt + *
3 rm -r source/delete_me.c source/do_not_delete_me.c
4
```

- `Ctrl + Alt + e` выполняет `alias`, историю и расширение консоли в текущей строке. Иными словами текущая строка будет отображена так же, как если бы она была обработана консолью.

```
1  
2 ls $HOME/tmp Ctrl Alt + e  
3 ls -N --color=tty -T 0 /home/cramey  
4
```

## 6. Возврат к истории команд и аргументов

Существует возможность выборочного доступа к введенным командам и аргументам с помощью оператора `!`. Это очень полезно при работе с длинными путями

Проверить свои последние команды можно через `history`.

Вы можете использовать предыдущие команды с `!`, где `n` индекс команды в `history`, отрицательные числа считаются в обратном порядке, начиная с последней команды в истории.

```
1 ls -l foo bar touch foo bar !-2
```

Вы можете использовать предыдущие аргументы с помощью `!:`, `0` это команда, `>= 1` это аргументы.

```
1 ls -l foo touch !:2 cp !:1 bar
```

Вы можете сочетать обе, используя `!:`

```
1 touch foo bar ls -l !:1 !:2 rm !-2:1 !-2:2 !-2
```

Вы также можете использовать диапазон аргументов `!:-`:

```
1 touch boo far ls -l !:1-2
```

## Другие специальные модификаторы `!` :

`*` для всех аргументов

```
1 ls -l foo bar ls !*
```

`^` для первого аргумента (`!:1 == !^`)

`$` для последнего аргумента

```
1 ls -l foo bar cat !$ &gt; /dev/null
```

## 7. Состояние скрипта

Использование `-x` позволяет увидеть что происходит в скрипте:

```
1 bash -x script.sh
```

## 8. SECONDS

Каждый раз при ссылке на этот параметр отображается количество секунд прошедших с момента вызова консоли. Если `SECONDS` было присвоено значение, то возвращаемое число будет состоять из секунд, прошедших со времени присвоения, и из числа, которое было присвоено.

```
1
2 SECONDS=0; sleep 5 ; echo "that took approximately $SECONDS seconds"
3
```

## 9. Генерирование переменной

Случайное генерирование специальной переменной:

```
1  
2 if [[ $((($RANDOM % 6)) = 0 ]]  
3 then echo "BANG"  
4 else echo "Try again" fi  
5
```

## 10. Регистр

Эта команда позволяет вводу перестать быть чувствительным к регистру. Это очень вам поможет во время быстрого набора директорий путей, особенно на Мас, где файловая система изначально не чувствительна к регистру. Например, можно поместить `.inputrc` в мою домашнюю папку.

```
1  
2 set completion-ignore-case on  
3
```