**[Шпаргалка по логическим операциям в bash](http://kyrych.ru/linux/bash/30-shpargalka-po-logicheskim-operatsiyam-v-bash)**

* [**linux**](http://kyrych.ru/component/tags/tag/3-linux)

* [**bash/sh и т.д.**](http://kyrych.ru/component/tags/tag/5-bash-sh-i-t-d)

[](http://kyrych.ru/linux/28-bash/30-shpargalka-po-logicheskim-operatsiyam-v-bash)

В bash есть много операторов для описания условных выражений и логических операций и их синтаксис не всегда привычен (по крайней мере на первый взгляд), так что не лишним будет иметь шпаргалку.

**Логические операции:**

* !  – отрицание;
* –a или && – логическое И;
* –o или || – логическое ИЛИ.

**Элементарные условные выражения для файлов:**

* -e — файл существует;
* -f — обычный файл (не каталог и не файл устройства);
* -s — ненулевой размер файла;
* -d — файл является каталогом;
* -b — файл является блочным устройством (floppy, cdrom и т.п.);
* -c — файл является символьным устройством (клавиатура, модем, звуковая карта и т.п.);
* -p — файл является каналом;
* -h — файл является символической ссылкой;
* -L — файл является символической ссылкой;
* -S — файл является сокетом;
* -t — файл связан с терминальным устройством;
* -r — файл доступен для чтения (пользователю, запустившему сценарий);
* -w — файл доступен для записи (пользователю, запустившему сценарий);
* -x — файл доступен для исполнения (пользователю, запустившему сценарий);
* -g — (sgid) флаг для файла или каталога установлен;
* -u — (suid) флаг для файла установлен;
* -k — флаг sticky bit установлен;
* -O — вы являетесь владельцем файла (пользователь запустивший сценарий);
* -G — вы принадлежите к той же группе, что и файл (пользователь запустивший сценарий);
* -N — файл был модифицирован с момента последнего чтения;
* file1 -nt file2 – file1 более новый, чем file2;
* file1 -ot file2 – file1 более старый, чем file2;
* file1 -ef file2 – file1 и file2 являются «жесткими» ссылками на один и тот же файл.

**Элементарные условные выражение для сравнения строк:**

* -z str – длина строки str равна 0;
* -n str – длина строки str не равно 0;
* str1 == str2 или str1 = str2 – строки совпадают;
* str1 !== str2 или str1 != str2 – строки не совпадают;
* str1 < str2 – str1 предшествует str2 в лексикографическом порядке;
* str1 > str2 – str1 следует за str2 в лексикографическом порядке.

**Арифметическое условное выражение имеет формат:** arg1 операция arg2, где аргументами являются целые числа, и допустимы следующие операции:

* -eq – равно;
* -ne – не равно;
* -lt – меньше;
* -le – меньше или равно;
* -gt – больше;
* -ge – больше или равно;
* < — меньше (внутри двойных круглых скобок);
* <= — меньше или равно (внутри двойных круглых скобок);
* > — больше (внутри двойных круглых скобок);
* >= — больше или равно (внутри двойных круглых скобок);

**Битовые операции:**

* << - сдвигает на 1 бит влево (умножение на 2);
* <<= - сдвиг-влево-равно; Например let "x <<= 2" значение переменной x сдвигается влево на 2 бита (умножается на 4);
* >> - сдвиг вправо на 1 бит (деление на 2);
* >>= - сдвиг-вправо-равно (имеет смысл обратный <<=);
* & - по-битовое И (AND);
* &= - по-битовое И-равно;
* | - по-битовое ИЛИ (OR);
* |= - по-битовое ИЛИ-равно;
* ~ - по-битовая инверсия;
* ! - По-битовое отрицание;
* ^ - по-битовое исключающее или (XOR);
* ^= - по-битовое исключающее-или-равно;

**Команды возвращающие код возврата (звучит странно, но команды нужные):**

* test - используется для логического сравнения (например test -f /home/kyrych/readme вернет истину если readme - обычный файл);
* [ ] - является специальной встроенной командой воспринимающей свои аргументы как выражение сравнения или файловую проверку [.....]. Вызывает ошибку при попытке использования внутри || (логическое или) и & (логическое и);
* [[ ]] - расширенная версия "[" (начиная с версии 2.02), внутри которой могут быть использованы || (логическое или), & (логическое и). Должна иметь закрывающую скобку "]]" (является зарезервированным словом, а не командой, его bash выполняет как один элемент с кодом возврата)
* (( )) - математическое сравнение.