

# Командата test

**check file types and  
compare values**

# Synopsis

test EXPRESSION

test

[ EXPRESSION ]

[ ]

[ OPTION

- *Интервалите около квадратните скоби са задължителни*

# Options

`--help`

`--version`

- *използват се само с* [ `OPTION`

# Празен test

- `test` или `[ ]`
- празният израз се оценява като *false*, командата завършва неуспешно

```
mara@OVNI:~$ touch junkfile
```

```
mara@OVNI:~$ test
```

```
mara@OVNI:~$ echo $?
```

```
1
```

```
mara@OVNI:~$ test -f junkfile
```

```
mara@OVNI:~$ echo $?
```

```
0
```

```
mara@OVNI:~$
```

# Логически операции в EXPRESSION

- **( EXPRESSION )**  
*тества се дали изразът има оценка true*
- **! EXPRESSION**  
*тества се дали изразът има оценка false*
- **EXPRESSION1 -a EXPRESSION2**  
*тества се дали и двата израза се оценяват като true*
- **EXPRESSION1 -o EXPRESSION2**  
*тества се дали поне един от двата израза се оценява като true*

# EXPRESSION

## за СИМВОЛНИ НИЗОВЕ

- **STRING**

*тества се дали дължината на низа е ненулева*

- **-n STRING**

*тества се дали дължината на низа е ненулева, низът трябва да е видим от test*

- **-z STRING**

*тества се дали дължината на низа е нулева, низът трябва да е видим от test*

- **STRING1 = STRING2**

*тества дали двата низа съвпадат (работи и с ==)*

- **STRING1 != STRING2**

*тества дали двата низа са различни*

# EXPRESSION

## за цели числа

- **INTEGER1 -eq INTEGER2**
- **INTEGER1 -ne INTEGER2**
- **INTEGER1 -gt INTEGER2**
- **INTEGER1 -ge INTEGER2**
- **INTEGER1 -lt INTEGER2**
- **INTEGER1 -le INTEGER2**
- *ако вместо цяло число, запишем **-l STRING**, това ще се оцени като дължината на низа*

# EXPRESSION

## за съществуване и вид на файл

- **-e FILE**
- **-f FILE**
- **-d FILE**
- **-L FILE, -h FILE**
- **-p FILE**
- **-b FILE**
- **-c FILE**
- **-S FILE**



## EXPRESSION

### за код на защита на файл

- `-r FILE, -w FILE, -x FILE`
- `-u FILE, -g FILE`
- `-k FILE`

## Други възможности

- `FILE1 -ef FILE2`
- `FILE1 -nt FILE2, FILE1 -ot FILE2`
- `-O FILE, -G FILE`

# Следване на символните връзки

*При оценяването на изразите за файлове символните връзки се следват, с изключение на тестовете **-h FILE** и **-l FILE***

# Командата ((

- **SYNOPSIS**

**(( ARITHMETIC\_EXPRESSION ))**

- Изчислява аритметичния израз и завършва с нулев код, ако оценката на израза е ненулева, в противен случай завършва с положителен код. За разлика от *expr* **не** извежда на стандартния изход оценката на израза.
- Синтаксисът на изразите е както в езика C, с всички възможности (сравнения, побитови и логически операции и т.н.)

# (( примери

```
mara@OVNI:~$ (( 2+3 ))
```

```
mara@OVNI:~$ echo $?
```

```
0
```

```
mara@OVNI:~$ (( 2-2 )) ; echo $?
```

```
1
```

```
mara@OVNI:~$ (( 2<1 )) ; echo $?
```

```
1
```

```
mara@OVNI:~$ (( 2+3 > 7-3 )) ; echo $?
```

```
0
```

```
mara@OVNI:~$ (( 2+3 >= 7-2 )) ; echo $?
```

```
0
```

```
mara@OVNI:~$
```

# Командата [ [

- **SYNOPSIS**

[ [ **EXPRESSION** ] ]

- разширява *test*, добавяйки възможности от езици за програмиране като C

# [[ и логически операции с test-изрази

- В `[[ ]]` може да поставяме изрази от *test*, оградени в `( )`, а между тях да има логическите знаци `&&` и `||`

```
mara@OVNI:~$ [[ ( -e /home/dvd ) &&  
( -e /home/mara ) ]] ; echo $?
```

```
0
```

```
mara@OVNI:~$ [[ ( -f /home/dvd ) ||  
( -f /home/mara ) ]] ; echo $?
```

```
1
```

```
mara@OVNI:~$
```

# [[ и СИМВОЛНИ НИЗОВЕ

Освен сравняване дали два низа съвпадат, чрез знаците = (==) и != между низовете [[ може да прави и pattern matching (базиран на filename expansion):

```
mara@OVNI:~$ [[ "abc" == ab.* ]] ; echo $?
```

```
1
```

```
mara@OVNI:~$ [[ "abc" == ab* ]] ; echo $?
```

```
0
```

```
mara@OVNI:~$ [[ "abc" = ab[^d] ]] ; echo  
$?
```

```
0
```

```
mara@OVNI:~$ [[ "abc" == ab[^d]? ]] ; echo  
$?
```

```
1
```

```
mara@OVNI:~$ [[ "abc" == ab? ]] ; echo $?
```

```
0
```

```
mara@OVNI:~$
```

# [[ и аритметични изрази

В [[ може да има и аритметични изрази като пресмятания и сравнения. Но е важно да се знае, че сравненията с < и > са не аритметични, а лексикографски!

```
mara@OVNI:~$ [[ 3+2 -lt 4+3 ]] ; echo $?  
0
```

```
mara@OVNI:~$ [[ 3 < 2 ]] ; echo $?  
1
```

```
mara@OVNI:~$ [[ 123 < 3 ]] ; echo $?  
0
```

```
mara@OVNI:~$ [[ 3+2 <= 4+3 ]]; echo $?  
bash: syntax error in conditional expression  
bash: syntax error near `4+3'
```

```
mara@OVNI:~$ [[ 3+2 < 4+3 ]]; echo $?  
0
```

```
mara@OVNI:~$ [[ 3+2 < a ]]; echo $?  
0
```

```
mara@OVNI:~$
```



# Край

Автор: Мария Николова

Източници: **man test** и

<http://www.ibm.com/developerworks/linux/library/l-bash-test.html>