

# Návratový kód

- Každý program vrací návratový kód (exit status)
- Možné hodnoty:
  - 0 – úspěšné dokončení příkazu
  - 1 – program pracoval, ale nebyl úspěšný
  - 2 – program nepracoval
  - 127 – příkaz nebyl vykonán (nastaveno shellem)
  - 127 + n – program ukončen abnormálně (n – číslo signálu)

```
$ cat > a
```

```
obsah_a
```

```
$ ls
```

```
a
```

```
$ echo $?
```

```
0
```

```
$ ls 2
```

```
ls: 2: No such file or directory
```

```
$ echo $?
```

```
1
```

```
$ lss a
```

```
bash: lss: command not found
```

```
$ echo $?
```

```
127
```

# Příkaz `test`

- vyhodnocuje příkaz v závislosti na jeho pravdivostní hodnotě
- návratový kód:

0 – pravda, úspěch

1 – nepravda, chyba

`test podmínka`

- Spouštění příkazu je možné i pomocí levé hranaté závorky

```
if [ podmínka ]
```

```
then
```

```
    příkaz
```

```
fi
```

[www.dedalebeda.wz.cz/36unx/14.pdf](http://www.dedalebeda.wz.cz/36unx/14.pdf)

- r** *soubor* soubor je uživateli přístupný pro čtení?
- w** *soubor* soubor je uživateli přístupný pro zápis?
- x** *soubor* soubor je uživateli přístupný pro spuštění?
- f** *soubor* soubor je obyčejný?
- d** *soubor* je to adresářový soubor?
- c** *soubor* jedná se o speciální znakový soubor?
- b** *soubor* blokový speciální soubor?
- p** *soubor* soubor je pojmenovanou rourou?
- u** *soubor* má nastavený set-user-ID bit?
- g** *soubor* má nastavený set-group-ID bit?
- k** *soubor* má nastavený sticky bit?

**-z řetězec** délka řetězce je nulová?

**-n řetězec** délka řetězce je nenulová?

**řetězec1 = řetězec2** řetězce jsou identické?

**řetězec1 != řetězec2** řetězce jsou různé?

**číslo1 -eq číslo2** (equal)

**číslo1 -ne číslo2** (not equal)

**-lt** (less than)                      **-le** (less or equal)

**-gt** (greater than)                      **-ge** (greater or equal)

**výraz1 -a výraz2** (AND)

**výraz1 -o výraz2** (OR)

```
$ A=12
```

```
$ B=12
```

```
$ test $A -eq $B && echo " $A je rovno $B "
```

```
$ 12 je rovno 12
```

# Příkaz `if`

- patří mezi složené příkazy (`case`, `until`, `while` ...)
- Syntaxe:

```
if seznam příkazů
then seznam příkazů
[ elif seznam příkazů
then seznam příkazů ] ...
[ else seznam příkazů ]
fi
```

```
$ if test -d bin      #test existence adresáře
> then
> mv pgm bin/pgm
> else
> echo "Adresar neexistuje"
> fi
$ Adresar neexistuje
```



```
$ if [ "$USER" == "root" ] then
> echo "Ahoj admine"
> elif [ "$USER" == "deda" ] then
> echo "Ahoj Dedo"
> else
> echo "Ahoj uzivатели"
> fi
```

Ahoj uzivатели