

# ***C2110 Operační systém UNIX a základy programování***

## **8. lekce**

**PS/2021 Prezenční forma výuky: Rev4**

**Petr Kulhánek**

[kulhanek@chemi.muni.cz](mailto:kulhanek@chemi.muni.cz)

Národní centrum pro výzkum biomolekul, Přírodovědecká fakulta  
Masarykova univerzita, Kamenice 5, CZ-62500 Brno

# Opakování

---

➤ **Bash**

# Vstup/výstup

## Vstup

- Argumenty skriptu
  - Dostupné ve speciálních proměnných (0, 1, 2, ..., 9, #)
- Příkaz **read** slouží ke **čtení textu** ze standardního vstupu a jeho uložení do proměnných.

```
read A      # celý řádek se uloží do proměnné A
```

```
read A B    # první slovo se uloží do proměnné A  
# zbytek řádku do proměnné B
```

## Výstup

- Příkaz **echo** slouží k **neformátovanému** výpisu do standardního výstupního proudu.

```
echo [volby] [retezec1] [retezec2] ...
```

- Příkaz **printf** slouží k vypisování **formátovaných** textů a čísel do standardního výstupního proudu.

```
printf [format] [hodnota1] [hodnota2] ...
```

# Aritmetické operace

- Aritmetické operace s celými čísly lze vykonat v bloku `(( ... ))`.
- Znaky se v bloku interpretují jako názvy proměnných. Proto není nutné používat operátor `$` pro získávání jejich hodnoty.
- Hodnoty proměnných se interpretují jako celá čísla. Pokud konverze selže, tak se použije nulová hodnota.


## Možné zápisy:

```
(( I = I + 1 ))
```

```
(( I++ ))
```

## Aritmetická operace se získáním výsledku:

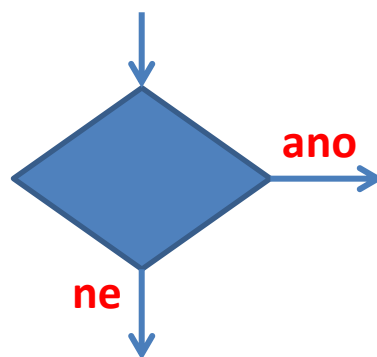
```
echo "Hodnota I zvetsena o jedna : $( ( I + 1 ) )"
```

 hodnotu výsledku se vypíše do místa zápisu

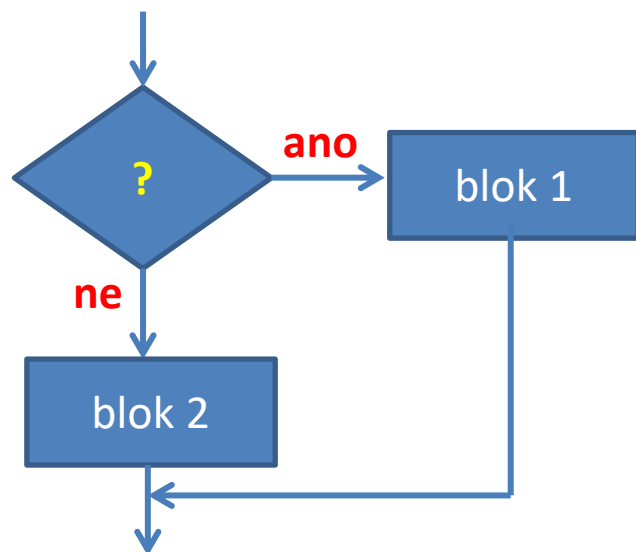
Další informace: `man bash`

# Blok rozhodování

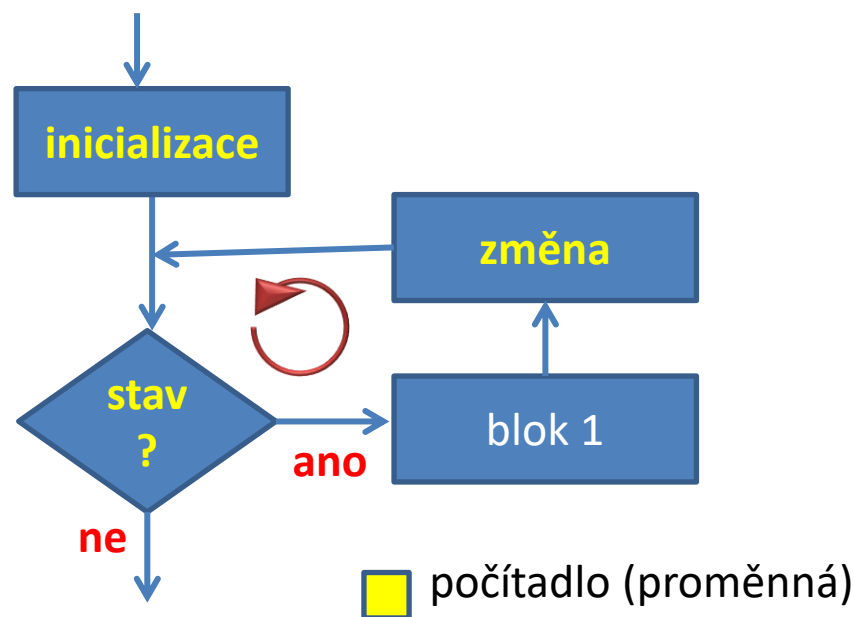
Typické použití bloku rozhodování



Podmíněné vykonání bloku (podmínky)



Cyklické vykonávání bloku (cykly)



# Podmínky

```
if prikaz1; then
    prikaz2
    ...
else
    prikaz3
    ...
fi
```

Je možné testovat návratovou hodnotu příkazu nebo výsledek logického výrazu.

```
if ! mkdir adresar 2> /dev/null; then
    echo "Nemohu vytvorit adresar!"
    exit 1
fi
```

Logické výrazy:

```
[[ cislo1 operator cislo2 ]]
[[ operator retezec1 ]]
```

Složitější logické výrazy:

```
||    logické nebo
&&   logické a
!     negace
```

Příklad:

```
[[ (I -ge 5) && (I -le 10) ]]
```

# Nová látka

---

➤ **Bash**

## ➤ Hledání chyb

- syntaktické a logické chyby, zvýrazňování syntaxe, statická analýza, ladění

## ➤ Cykly

- while/until
- vnořování