C2110 *Operační systém UNIX a základy programování*

7. lekce / modul 2

PS/2021 Prezenční forma výuky: Rev3

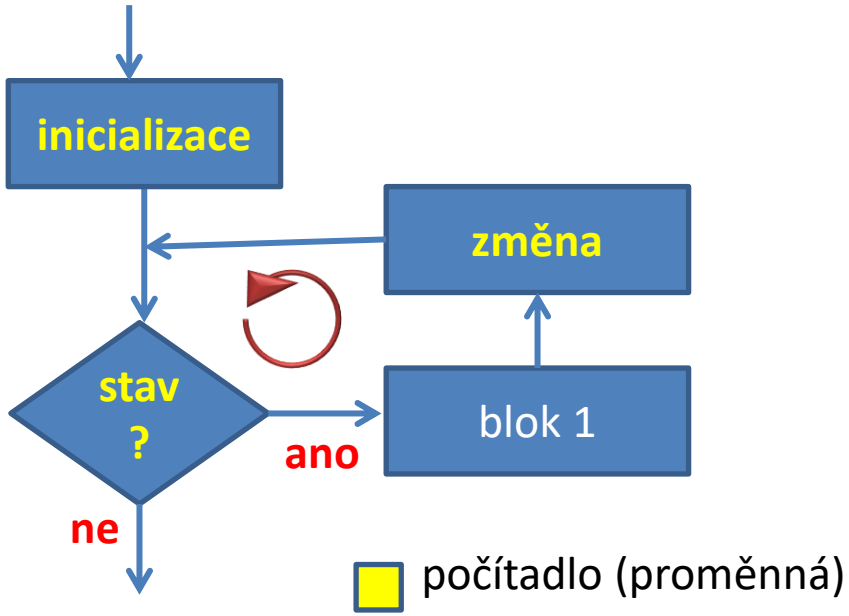
Petr Kulhánek

kulhanek@chemi.muni.cz

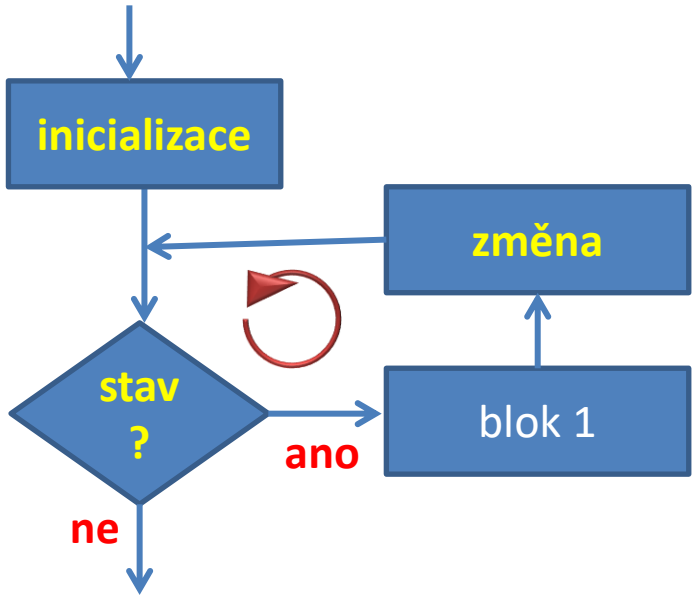
Národní centrum pro výzkum biomolekul, Přírodovědecká fakulta
Masarykova univerzita, Kamenice 5, CZ-62500 Brno

Cykly

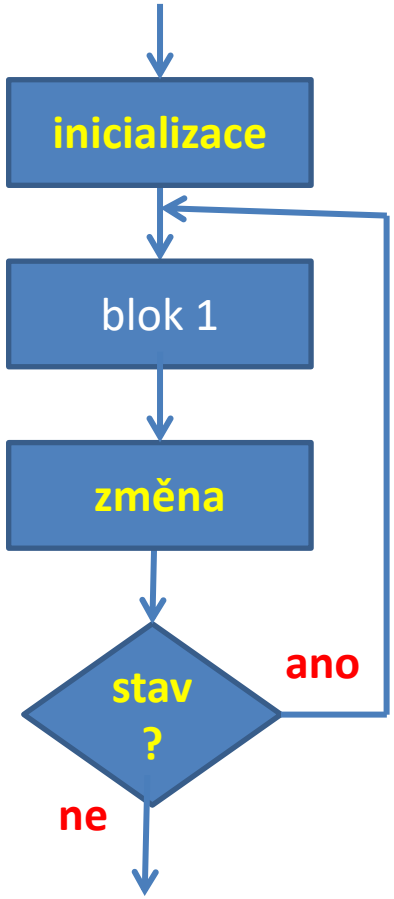
Cyklické vykonávání bloku



Cyklus pomocí while/until ...

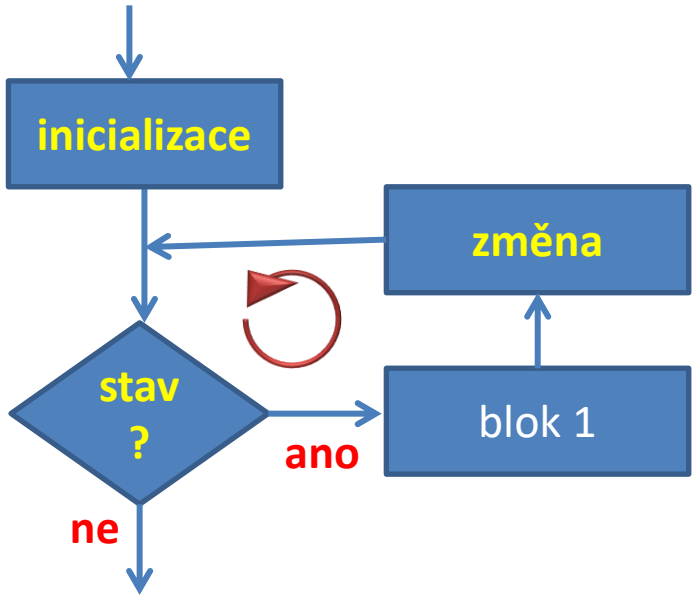


vyhodnocení podmínky
na začátku cyklu

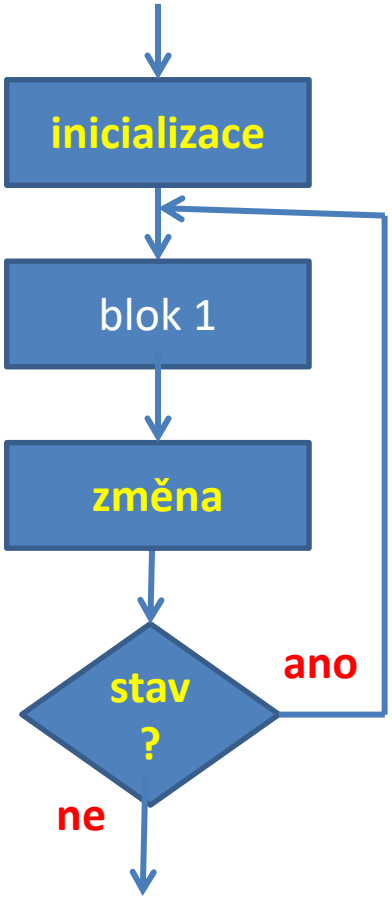


vyhodnocení podmínky
na konci cyklu

Cyklus pomocí while/until ...



vyhodnocení podmínky
na začátku cyklu

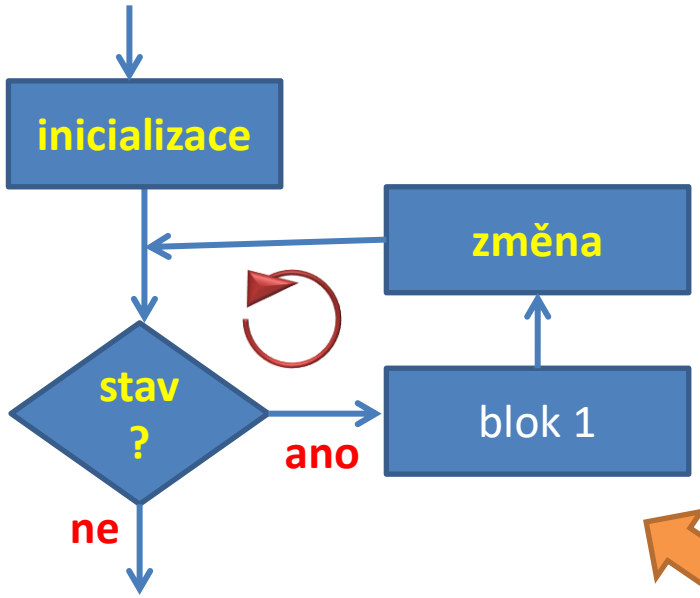


vyhodnocení podmínky
na konci cyklu

komplikovaná implementace v bash

Tento algoritmus nemá přímou podporu v řídích strukturách jazyka bash, jeho přepis je možný, ale za cenu horší čitelnosti výsledného kódu.

Cyklus pomocí while/until ...



snadná transformace

vyhodnocení podmínky
na začátku cyklu

vyhodnocení podmínky
na konci cyklu

snadná implementace v bash

Tento algoritmus nemá přímou podporu v řídích strukturách jazyka bash, jeho přepis je možný, ale za cenu horší čitelnosti výsledného kódu.

Cyklus pomocí while/until

Cyklus (smyčka) je řídicí struktura, která opakovaně provádí posloupnost příkazů. Opakování i ukončení cyklu je řízeno podmínkou.

```
while prikaz1
      do
            prikaz2
            ...
      done
```

cyklus probíhá **zatímco** prikaz1 **vrací** v návratové hodnotě 0 (bez chyby)

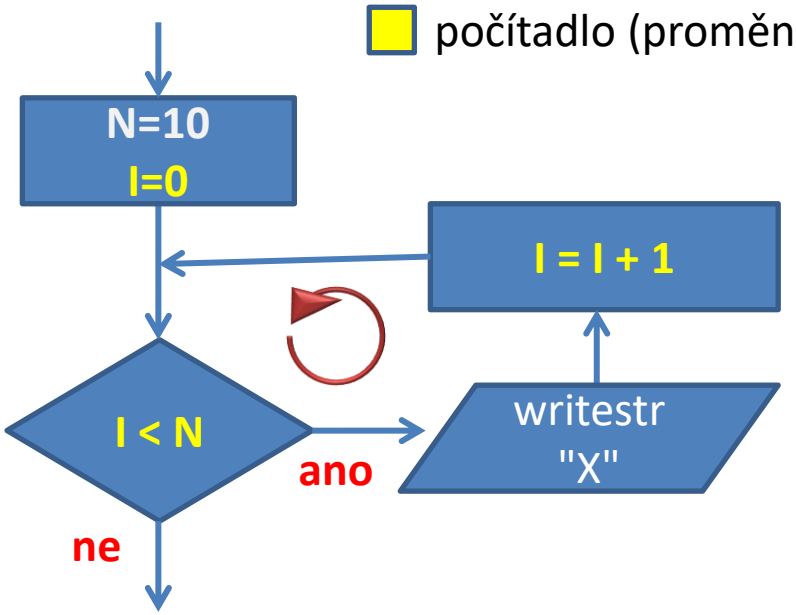
cyklus probíhá **dokud** prikaz1 **nevrátí** v návratové hodnotě 0

Kompaktní zápis:

```
while prikaz1; do
      prikaz2
      ...
done
```

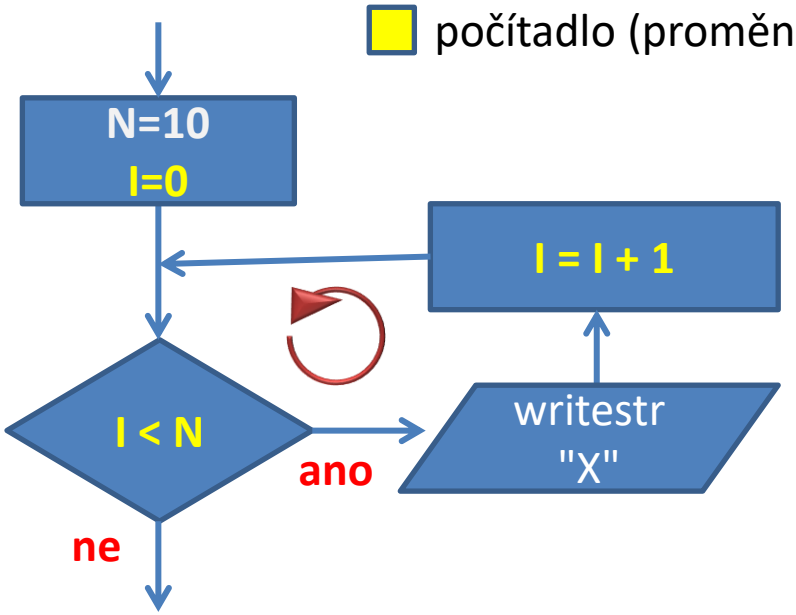
```
until prikaz1; do
      prikaz2
      ...
done
```

Praktický příklad - cyklus




```
N=10
I=0
while [[ I -lt N ]]; do
      echo "X"
      ((I = I + 1))
done
```

Praktický příklad - cyklus



nutno použít $

```
N=10
I=0
while test "$I" -lt "$N"; do
      echo "X"
      ((I = I + 1))
done
```

```
N=10
I=0
while [[ I -lt N ]]; do
      echo "X"
      ((I = I + 1))
done
```

```
N=10
I=0
while [[ "$I" -lt "$N" ]]; do
      echo "X"
      ((I = I + 1))
done
```

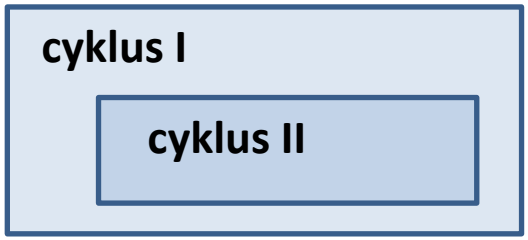
volitelné $, pokud je použit blok [[ ]] nebo (( ))

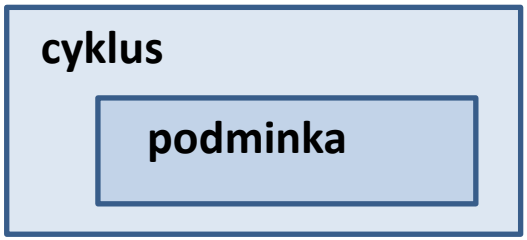
Cvičení I

1. Napište skript v jazyce bash, který do terminálu vypíše N znaků "X ". Počet znaků uživatel zadá jako první argument skriptu. Skript vypíše chybové hlášení v případě, že zadaný počet znaků nebude větší než dva.

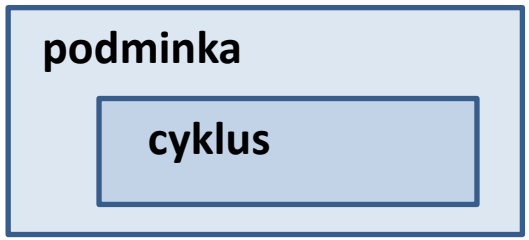
Složitější konstrukce - vnořování

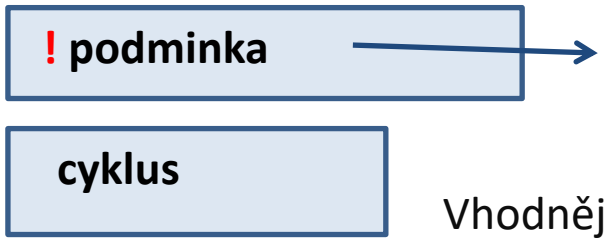
Jazyk bash **nemá návěstí a příkaz goto**, či jeho obdobu. Komplexnějších konstrukcí lze tedy dosáhnout jen zanořováním cyklů a podmínek vzájemně do sebe. Úroveň zanoření není omezena.





Při návrhu algoritmu/skriptu se však snažíme o zamezení zbytečného vnořovaní (převážně z důvodu snadnější orientace ve skriptu).





Vhodnější uspořádání např. pro testovaní vstupních dat od uživatelů.

Vnořování cyklů - příklad

```
N=10
I=0
while [[ I -lt N ]]; do
      J=0
      while [[ J -lt I ]]; do
            echo -n "X"
            ((J = J + 1))
      done
      echo ""
      ((I = I + 1))
done
```

počítadlo vnějšího cyklu může ovlivňovat chování vnitřního cyklu

U zanořených konstrukcí dbáme na **odsazovaní textových bloků**, které zvyšuje **přehlednost a čitelnost kódu**. V textových editorech je integrována podpora, která odsazovaní usnadňuje, např. v editoru **gedit**, lze odsazení označeného textového bloku dosáhnout klávesou TAB či Shift+TAB.

Cvičení II

1. Napište skripty v jazyce bash pro následující úlohy. Rozměr vykreslovaného obrazce nechť uživatel zadá interaktivně po spuštění skriptu. Při práci se opírejte o vytvořený algoritmus z domácí úlohy.

Úkol 1

Do terminálu vytiskněte čtverec se znaků **X**. Délku strany čtverce zadá uživatel.

```
x x x x x x x x x x
x x x x x x x x x x
x x x x x x x x x x
x x x x x x x x x x
x x x x x x x x x x
x x x x x x x x x x
x x x x x x x x x x
x x x x x x x x x x
x x x x x x x x x x
x x x x x x x x x x
```

To, že se nejedná vzhledově o čtverec, ignorujte. Počet znaků **X** na řádku a počet řádků však musí být stejný.

Úkol 2

Do terminálu vytiskněte pravoúhlý trojúhelník se znaků **X**, tak aby jedna odvěsna byla umístěna nahoře a druhá na levé straně. Délku odvěsny zadá uživatel.

```
x x x x x x x x x x
x x x x x x x x x
x x x x x x x x
x x x x x x x
x x x x x x
x x x x x
x x x x
x x x
x x
x
```

Úkol 3

Do terminálu vytiskněte pravoúhlý trojúhelník se znaků **X**, tak aby jedna odvěsna byla umístěna dole a druhá na levé straně. Délku odvěsny zadá uživatel.

```
x
x x
x x x
x x x x
x x x x x
x x x x x x
x x x x x x x
x x x x x x x x
x x x x x x x x x
x x x x x x x x x x
```