

C2110 Operační systém UNIX a základy programování

8. lekce / modul 1

PS/2020 Distanční forma výuky: Rev2

Petr Kulhánek

kulhanek@chemi.muni.cz

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

Cyklus

Cyklus pomocí for

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

provede se před spuštěním cyklu
(inicializace počítadla)

pokud je podmínka splněna, vykonají se
příkazy prikaz1 a další

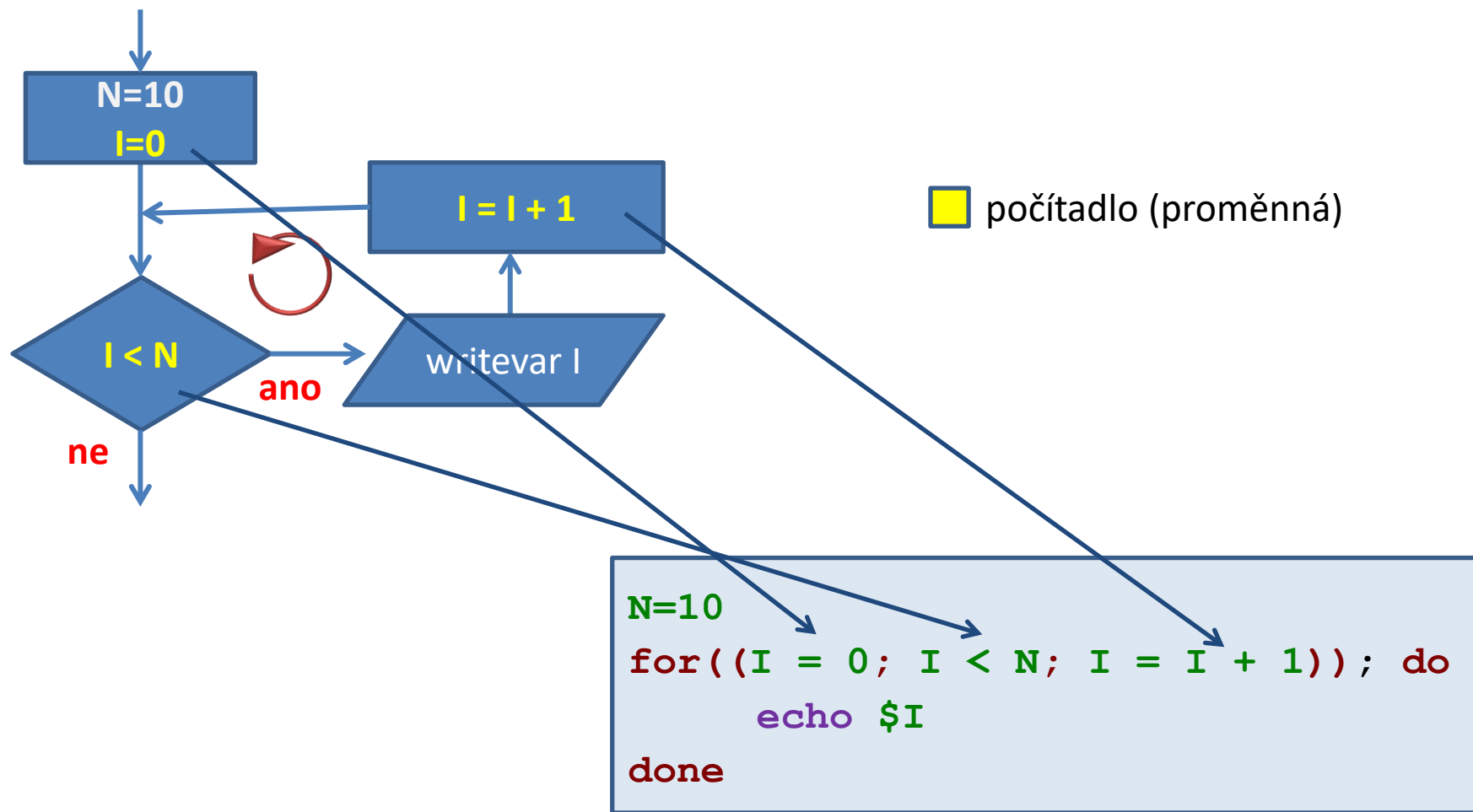
```
for( (inicializace; podminka; zmena) )
do
    prikaz1
    ...
done
```

aktualizace počítadla po
vykonání příkazů

Kompaktní zápis:

```
for( (inicializace; podminka; zmena) ); do
    prikaz1
    ...
done
```

Cyklus pomocí for a vývojový diagram



Cyklus pomocí for versus while

```
for( (I=1; I <= 10; I++) ); do
    echo $I
done
```

provede se před spuštěním cyklu
(inicializace počítadla)

Kromě změny počítadla na konci cyklu, je možné provádět i další změny v těle cyklu. Toto se však **NEDOPORUČUJE**, protože to snižuje čitelnost kódu.

pokud je podmínka splněna, vykonají se příkazy v bloku do/done

```
I=1
while [[ I -le 10 ]]; do
    echo $I
    (( I = I + 1 ))
done
```

aktualizace počítadla po vykonání příkazů

Změnu počítadla je možné provádět **kdekoliv** v těle cyklu (i na více místech).

Cyklus pomocí for, použití

Vypíše čísla 1 až 10

```
for((I=1;I <= 10;I++)); do
    echo $I
done
```

Proměnná **I** má roli **počítadla**.

Inicializace se řídí volnými pravidly, jelikož je výraz uveden v (()) bloku.

Změna:

Lze použít libovolný výraz, který je možné interpretovat v (()) bloku, např.

- ++ hodnotu proměnné zvýší o jedničku
- hodnotu proměnné sníží o jedničku
- další

Vypíše čísla 10 až 1

```
for((I=10;I >= 1;I--)); do
    echo $I
done
```

Podmínka:

Lze použít následující porovnávací operátory:

!=	nerovná se
==	rovná se
<	menší
<=	menší nebo rovno
>	větší
>=	větší nebo rovno

Lze použít pouze na celá čísla v (()) .

Cyklus pomocí for, změna počítadla

Pokud lze proměnnou interpretovat jako celé číslo, lze použít následující aritmetické operátory:

++ hodnotu proměnné zvýší o jedničku

A++

-- hodnotu proměnné sníží o jedničku

A--

+ sečte dvě hodnoty

A = 5 + 6

A = A + 1

- odečte dvě hodnoty

A = 5 - 6

A = A - 1

***** vynásobí dvě hodnoty

A = 5 * 6

A = A * 1

/ vydělí dvě hodnoty (celočíslné dělení)

A = 5 / 6

A = A / 1

A=A+3

+= k proměnné přičte hodnotu

A += 3

A += B

-= od proměnné odečte hodnotu

A -= 3

A -= B

***=** proměnnou vynásobí hodnotou

A *= 3

A *= B

/= proměnnou podělí hodnotou

A /= 3

A /= B



Vnořování cyklů

Řídící skupiny cyklů lze do sebe libovolně vnořovat.

```
for((I=1;I <= 10;I++)); do
  for((J=1;J <= 10;J++)); do
    echo "$I $J"
  done
done
```

vnější cyklus

vnitřní cyklus

```
for((I=1;I <= 10;I++)); do
  for((J=1;J <= I;J++)); do
    echo "$I $J"
  done
done
```

počítadlo vnějšího cyklu může ovlivňovat chování vnitřního cyklu

Počet vnoření není omezen. Lze kombinovat s jinými cykly (while, until, for in) nebo podmínkami.

Cvičení 1

1. Napište skripty v jazyce bash řešící Úkol 1, 2 a 3. Místo cyklu pomocí while použijte cyklus pomocí for. Rozměr vykreslovaného obrazce nechť uživatel zadá jako první argument skriptu. Skript otestuje, zda-li je zadán správný počet argumentů a zda-li je první argument celé číslo větší než nula.
2. Upravte řešení Úkolu 1 tak, aby se vykresloval obdélník. Rozměry obdélníku bude uživatel zadávat interaktivně po spuštění skriptu.
3. Napište skripty v jazyce bash řešící Úkol 4 a 5.

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. Popř. použijte "X " - X a mezera.

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
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
```

Úkol 4

Do terminálu vytiskněte obrys čtverce 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
```

Vhodným řešením je využít kostru z Úkolu 1 a vypisování znaků kontrolovat pomocí vhodně zvolené podmínky.

Úkol 5

Do terminálu vytiskněte obrys čtverce a jeho uhlopříčky 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
```

Vhodným řešením je využít kostru z Úkolu 1 a vypisování znaků kontrolovat pomocí vhodně zvolené podmínky.