



KATEDRA
INFORMATIKY
UNIVERZITA PALACKÉHO V OLMOUCI

Jazyk C

Řízení běhu programu

Mgr. Markéta Trnečková, Ph.D.

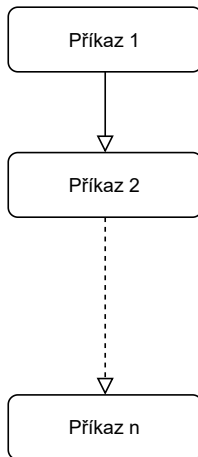
Jazyk C

„Computers are good at following instructions but not at reading your mind.“

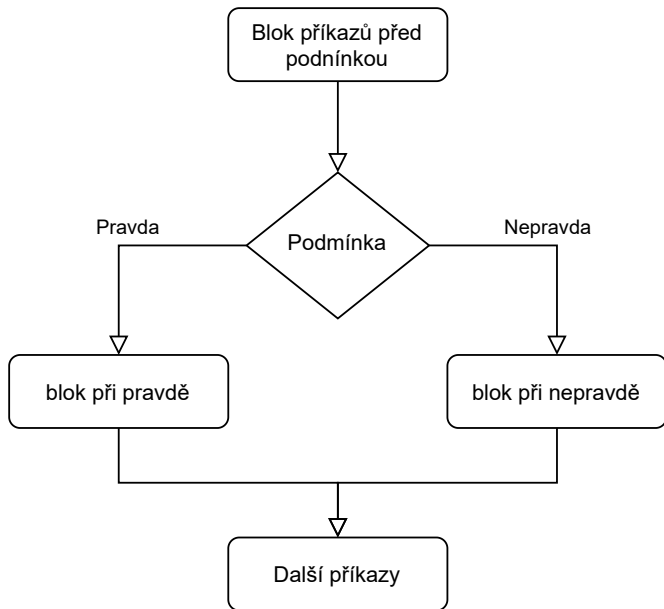
(Donald Knuth)

Sekvenční vykonávání programu

- **Program flow** = tok programu



Větvení



Větvení

Větvení pomocí if a else

```
if (podminka)
    blok_pri_pravde
else
    blok_pri_nepravde
```

Větvení

Podmínky

Logické operátory

- <, <= menší, menší rovno
- >, >= větší, větší rovno
- == rovnost (pozor nezaměňovat s operátorem přiřazení =)
- != nerovnost

Příklad (priklad1.c)

Co vypíše následující část kódu?

```
int a = 0;

if (a = 0){
    printf("a je rovno 0");
}
else{
    printf("a není rovno 0");
}
```

Větvení

Podmínky

Logické spojky

- || nebo
- && a zároveň
- ! negace

Líné vyhodnocování || a &&

Příklad (vyhodnoceni.c)

```
if ((y != 0) && ((x / y) < z))
```

Větvení

Příklad – výpis většího čísla

Příklad (vetsi.c)

```
if (a >= b){  
    printf("%d", a);  
}  
else{  
    printf("%d", b);  
}
```


Větvení

Podmínky

- Štábní kultura – podmínky v kladném znění

Příklad

Jak by vypadala podmínka v kladném znění?

```
if (!(c == '\0' || c == ' ' || c == '1'))
```

Větvení

Podmínky

Příklad (Řešení)

```
if (c != '\0' && c != ' ' && c != '1')
```

Větvení

Vnořené větvení

Příklad (if1.c)

```
int a = 5, b = 1, c = 3, foo = 10;

if (a > b){
    if (b > c)
        foo = b;
}
else foo = c;
```

Příklad (if2.c)

```
int a = 5, b = 1, c = 3, foo = 10;

if (a > b)
    if (b > c)
        foo = b;
    else foo = c;
```

Větvení

Větvení pomocí switch

```
switch (vyraz){  
    case konstanta1:  
        blok1  
    case konstanta2:  
        blok2  
    ...  
    default:  
        blok  
}
```

Větvení

Větvení pomocí switch

Příklad

```
if (a == 1)
    printf("a je jedna");
else if (a == 2)
    printf("a je dva");
else if (a == 3)
    printf("a je tri");
else
    printf("a není 1, 2 ani 3");
```

Příklad

```
switch (a){
    case 1:
        printf("a je jedna");
        break;
    case 2:
        printf("a je dva");
        break;
    case 3:
        printf("a je tri");
        break;
    default:
        printf("a není 1, 2 ani 3");
}
```

Větvení

Podmínkový operátor

podminka ? vyraz1 : vyraz2;

Příklad

```
if (a < b)
    x = a;
else
    x = b;
```

Příklad

```
x = (a < b) ? a : b;
```

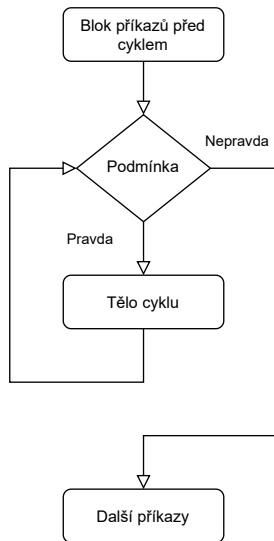
Cykly

- cyklus while,
- cyklus for,
- cyklus do while.

Cykly

Cyklus while

```
while (podminka)  
    telo_cyku
```



Cykly

Cyklus while – příklad

Příklad

```
int j;  
  
j = 0;  
while(j < 10){  
    printf("%d ", j);  
    j = j + 1;  
}
```

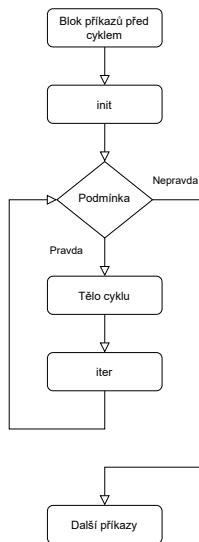
Cykly

Cyklus for

```
for (init; podminka; iter)  
    telo_cyklad
```

- **Krokovací proměnná**

- `init` a `iter` jsou výrazy



Cykly

Cyklus for

Příklad

```
int j;  
  
for(j = 0; j < 10; j = j + 1){  
    printf("%d ",j);  
}
```

Cykly

Cyklus for

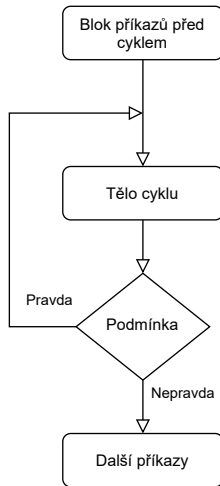
Příklad

```
int i = 0;
/* doporučené použití cyklu for */
for(i = 0; i < 10; i++){
    printf("%d ", i);
}
/* využití inicializace i při definici. */
for(; i < 10; i++){
    printf("%d ", i);
}
/* řídicí proměna se mění v těle cyklu */
for(i = 0; i < 10; ){
    printf("%d ", i++);
}
/* využití operátoru čárky */
for(i = 0; i < 10; printf("%d ", i), i++)
    ;
```

Cykly

Cyklus do while

```
do  
    telo-cyklad  
while (podminka);
```



Příkazy break a continue

Příklad (break.c)

```
int j;  
for(j = 1; j < 10; j = j + 1){  
    if((j % 3) == 0)  
        break;  
    printf("%d ", j);  
}
```

Příklad (continue.c)

```
int j;  
for(j = 1; j < 10; j = j + 1){  
    if((j % 3) == 0)  
        continue;  
    printf("%d ", j);  
}
```

Příkaz goto

```
goto navesti;
```

```
...
```

```
navesti:
```

```
...
```

Úkol 1

Přepište následující kód bez použití příkazu goto.

Příklad (goto.c)

```
for (i = 0; i < 10; i++){
    for (j = 0; j < 10; j++){
        for (k = 0; k < 10; k++){
            /* x je pole a x[k] vraci jeho k-ty prvek */
            if (x[k] == 0)
                goto chyba;
            printf("%d", (x[i] + x[j]) / x[k]);
        }
    }
}
...
chyba:
printf("Nelze delit nulou\n");
```


Příkaz return

```
return 0;
```

- pozor na rozdíl mezi return a exit

Cvičení

- 1 Za použití podmínkového operátoru napište program, který pro zadané číslo vypíše jeho absolutní hodnotu.
- 2 Napište program, který rozhodne, zda zadaný rok je přestupný. (Definici přestupného roku naleznete například na wikipedii https://cs.wikipedia.org/wiki/P%C5%99estupn%C3%BD_rok).
- 3 Napište program, který rozhodne, zda je zadané písmeno malé nebo velké.
- 4 Pro zadané číslo od 1 do 10, vypíše sestupnou posloupnost čísel od zadaného čísla po 1. (pro 4 bude výstup 4 3 2 1). K řešení zkuste vhodně použít konstrukci `switch`. (Vhodnější je použít cyklus. Tento úkol je spíše k zamyšlení.)
- 5 Napište program, který načte celá čísla a a b a pak
 - 1 vypíše prvních a násobků čísla b
 - 2 spočítá a -tou mocninu čísla b
 - 3 určí kolik číslic má číslo a
 - 4 vypočítá a -té Fibonacciho číslo
 - 5 sečte všechna čísla větší než a a menší než b

Cvičení

6 Pro zadané n vykreslete do konzole následující obrázky

1 pro $n=3$

```
  *
 * * *
* * * * *
```

pro $n = 4$

```
  *
 * * *
* * * * * *
```

2 pro $n=2$

```
 *
 * * *
 *
```

pro $n = 3$

```
  *
 * * *
* * * * *
 * * *
  *
```

Cvičení

7 3 šachovnici o straně n pro $n = 4$

```
. * . *
* . * .
. * . *
* . * .
```

8 Napište program, který pro zadané číslo vrátí číslo zapsané pozpátku. (Pro 1234 vrátí číslo 4321)

Cvičení

Přiřaďte výstupy k blokům kódu po dosažení bloku do následujícího kódu.

Příklad

```
#include <stdio.h>

int main(){
    int x = 0;
    int y = 0;
    while (x < 5){
        /* sem
           vložte
           blok
           kódu */
        printf("%d%d ", x, y);
        x = x + 1;
    }
    return 0;
}
```

Cvičení

Bloky kódu:

```
y = x - y;
```

```
y = y + x;
```

```
y = y + 2;
```

```
if (y > 4)
```

```
    y = y - 1;
```

```
x = x + 1;
```

```
y = y + x;
```

```
if (y < 5){
```

```
    x = x + 1;
```

```
    if (y < 3)
```

```
        x = x - 1;
```

```
}
```

```
y = y + 2;
```

Možné výstupy:

22 46

11 34 59

02 14 26 38

02 14 36 48

00 11 21 32 42

11 21 32 42 53

00 11 23 36 410

02 14 25 36 47