



KATEDRA
INFORMATIKY
UNIVERZITA PALACKÉHO V OLOMOUCI

Jazyk C

Bitové operace a bitová pole

Mgr. Markéta Trnečková, Ph.D.

Jazyk C

„Computers are fast; developers keep them slow.“

(Anonymous)

Bitové operace

- bitový součin (AND)
- bitový součet (OR)
- bitový exkluzivní součet (XOR)
- bitový posun doleva
- bitový posun doprava
- jedničkový doplněk (NOT)

Bitové operace

Bitový součin

x & y

Příklad

100 & 50 = 32

100	0 1 1 0 0 1 0 0
50	0 0 1 1 0 0 1 0
100 & 50	0 0 1 0 0 0 0 0

Bitové operace

Bitový součin

Příklad (liche.c)

```
#define liche(x) (1 & (x))
```

Příklad

Jaké budou hodnoty k a l?

```
int i = 1, j = 2, k, l;
```

```
k = i && j;
```

```
l = i & j;
```

Bitové operace

Bitový součet

x | y

Příklad (namale.c)

```
#define na_mala(c) (c | 0x20)
```

Bitové operace

Bitový exkluzivní součet

$x \hat{=} y$

Příklad

```
if(x ^ y)
    /* cisla jsou rozdilna */
```

Bitové operace

Bitový posun doleva

$x \ll n$

Příklad

$x = y \ll 1$ vynásobí číslo 2

1 = 0 0 0 0 0 0 0 1

$1 \ll 1 = 0 0 0 0 0 0 1 0$

$x = y \ll 3$ vynásobí číslo 8 (2^3)

Příklad (posun.c)

```
i = j * 80; /* pomalejsi */
```

```
i = (j << 6) + (j << 4); /* rychlejsi */
```


Bitové operace

Bitový posun doleva

$x \gg n$

Příklad

$x = y \gg 1$ vydělí číslo číslem 2

$x = y \gg 3$ vydělí číslo 8

Bitové operace

Jedničkový doplněk (negace bit po bitu)

$\sim x$

Příklad (doplnek.c)

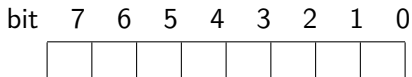
```
int main()
{
    int i = 10, i2;
    unsigned char j = 10, j2;

    i2 = ~i;
    j2 = ~j;

    printf("%d %d \n", sizeof(char), sizeof(int));
    printf("%u %d", i2, j2);
}
```

Práce se skupinou bitů

```
unsigned int status;
```



Příklad

```
#define READ 0x8  
#define WRITE 0x10  
#define DELETE 0x20
```

nastavení všech příznaků na 1: `status |= READ | WRITE | DELETE;`

nastavení všech příznaků na 0: `status &= ~(READ | WRITE | DELETE);`

je příznak nastaven na 0: `if (! (status & READ))`

Bitové pole

Příklad (bitovepole.c)

```
typedef struct{
    unsigned zacatek : 3; // bity od 0 do 2
    unsigned read : 1; // bit 3
    unsigned write : 1; // bit 4
    unsigned delete : 1; // bit 5
} STAV;
```

```
STAV status;
```

nastavení příznaku čtení na 1: `status.read = 1;`

nastavení příznaku zápisu na 0: `status.write = 0;`

je příznak nastaven na 1: `if(status.read).`

Cvičení

- 1** Z příkladu v sekci o bitovém součtu víme, že se malé písmeno a odpovídající velké liší v 5. bitu. Napište makro, které převede malé písmeno na velké.
- 2** Napište funkci, která pro zadané číslo `cislo` a číslo `n` zjistí hodnotu `n`-tého bitu čísla `cislo`. Například hodnota 1. bitu čísla 3 je 1, 2. bit má hodnotu 0. Bude se vám hodit operace bitového posunu.
- 3** Vyzkoušejte si práci se stavovým slovem. Vytvořte funkci, která bude brát jako vstup celé číslo `n` a stav a vypíše všechna čísla od 0 do `n` a bude vynechávat násobky podle stav. stav bude uchovávat informaci, zda se mají vypisovat násobky 2, 3 a 5. Například ve stav je nastaven příznak, že se nemají vypisovat násobky 2, pak funkce vypíše jen lichá čísla do `n`.
- 4** Upravte předchozí kód tak, aby stav nastavoval uživatel po dotazu v konzoli.
- 5** Upravte předchozí příklad tak, aby stav byl definován jako bitové pole.
- 6** Napište funkci, která způsobí rotaci (ne posun) čísla `x` o `n` bitů doprava. Pro číslo 3 (00000011) a `n = 3` dostaneme 96 (01100000) (pracujeme-li nad `char`).
- 7** Napište program, který bude využívat bitového pole pro úschovu času (hodiny, minuty, vteřiny). Vytvořte i funkce na převod času do bitového pole a funkci, která vypíše bitové pole jako čas.