Procedūrinio programavimo pagrindai

Bitų valdymas

lekt. Irmantas Radavičius
irmantas.radavicius@mif.vu.lt
Informatikos institutas, MIF, VU

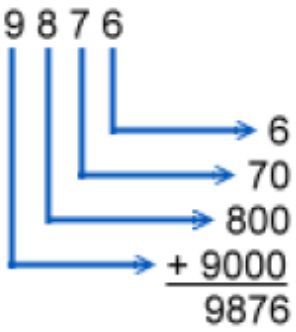
Turinys

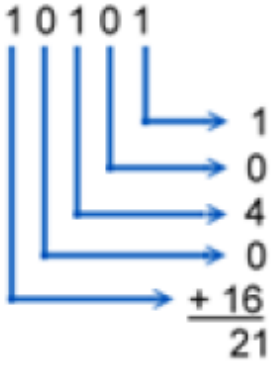
Pozicinės skaičiavimo sistemos

Kompiuteryje – “dvejetainė kalba”.

"There are 10 types of people in the world:

Those who understand binary, and those who don't."





Konvertavimas!

Binary	Hex	Decimal
0000	0	0
0001	1	1
0010	2	2
0011	3	3
0100	4	4
0101	5	5
0110	6	6
0111	7	7
1000	8	8
1001	9	9
1010	A	10
1011	B	11
1100	C	12
1101	D	13
1110	E	14
1111	F	15

Pozicinės skaičiavimo sistemos C kalboje

Dešimtainė sistema: `char x = 100;` // 100

Aštuntainė sistema: `short x = 0100;` // 64

Šešioliktainė sistema: `int x = 0x100;` // 256

Printf

`d, i`	`int`; decimal number.
`o`	`int`; unsigned octal number (without a leading zero).
`x, X`	`int`; unsigned hexadecimal number (without a leading `0x` or `0X`), using `abcdef` or `ABCDEF` for 10, ..., 15.

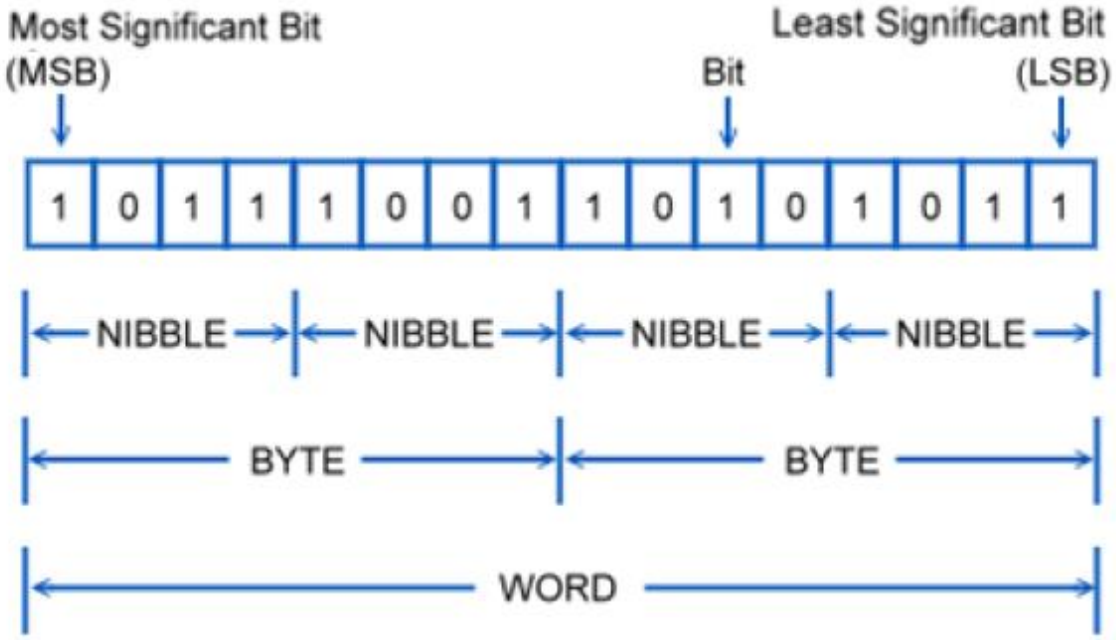
Scanf

`d`	decimal integer; `int *`.
`i`	integer; `int *`. The integer may be in octal (leading `0`) or hexadecimal (leading `0x` or `0X`).
`o`	octal integer (with or without leading zero); `int *`.
`u`	unsigned decimal integer; `unsigned int *`.
`x`	hexadecimal integer (with or without leading `0x` or `0X`); `int *`.

Bitai C kalboje

Visose operacijose mažiausias operandas – baitas (char).

Bitai – operandų (baitų, žodžių, struktūrų, etc) sudėtinės dalys.



Name	Bits	Max value
Bit	1	1
Nibble	4	15
Byte	8	255
Word	16	65535
Doubleword	32	4294967295

Bitų laukai struktūrose

struct-declarator:

declarator: constant-expression
:constant-expression

```
struct {
 unsigned int a : 1;
 unsigned int   : 0;
 unsigned int   : 1;
}  class;
```

```
struct ARGB1555 {
     union {
         uint16_t value;
         struct {
             unsigned int blue : 5;
             unsigned int green : 5;
             unsigned int red : 5;
             unsigned int alpha : 1;
         } comp;
     } u;

};
```

Galima valdyti:

- laukų dydį bitais
- tarpus (bevardžiai laukai)
- išlygiavimą (nulinis dydis)

Savybės:

- ✓ galima taupyti vietą
- – priklauso nuo implementacijos (eiliškumas, struktūros dydis, ...)

Bitų operacijos sveikiesiems skaičiams

Loginės operacijos dirba su visu operandu. Bitų operacijos dirba su atskirais operando bitais. Bitų operacijos – "žemo lygio" operacijos.

Loginės operacijos	Bitų operacijos
&& || !	& | ~ ^ >> <<

X	Y	X and Y	X or Y	not X	not Y	X xor Y
1	1	1	1	0	0	0
1	0	0	1	0	1	1
0	1	0	1	1	0	1
0	0	0	0	1	1	0

Bitų panaudojimas

```
enum Days {
    MON = 1,     // 0..0 0000 0001
    TUE = 2,     // 0..0 0000 0010
    WED = 4,     // 0..0 0000 0100
    THU = 8,     // 0..0 0000 1000
    FRI = 16,    // 0..0 0001 0000
    SAT = 32,    // 0..0 0010 0000
    SUN = 64     // 0..0 0100 0000
};

int main(){
    int weekend = SAT | SUN;                    // 0..0 0110 0000
    int allDays = 0x7F;                         // 0..0 0111 1111
    int workingDay = allDays & (~weekend);      // 0..0 0001 1111
    return 0;
}
```

Bitų valdymas

Svarbios savybės

x AND 0 = 0	x OR 0 = x	x XOR 0 = x
x AND 1 = x	x OR 1 = 1	x XOR 1 = ~x

Bitų valdymas:

nustatymas 0	–	operacija AND	‘a’ & 0xdf
nustatymas 1	–	operacija OR	‘A’ | 0x20
keitimas priešingu	–	operacija XOR	x ^ 0x20

Postūmio operacijos

```
[integer] [operator] [number of places];
```

Postūmis kairėn (<<) – daugyba iš 2.

Dešinėje atsiranda nuliniai bitai.

Postūmis dešinėn (>>) – dalyba iš 2.

Kairėje atsirandančių bitų turinys priklauso nuo implementacijos.

Galimi variantai: nuliniai arba ženklo (!) bitai.

Reikšmių nuskaitymas ir įrašymas

```
dwColor:   AAAA AAAA RRRR RRRR GGGG GGGG BBBB BBBB
mask:    & 0000 0000 1111 1111 0000 0000 0000 0000
         -----------------------------------------
result:    0000 0000 RRRR RRRR 0000 0000 0000 0000
Shift:                                    >> 16
         -----------------------------------------
Result:    0000 0000 0000 0000 0000 0000 RRRR RRRR  ==  RRRR RRRR
```

Reikšmės nuskaitymas

```
Previous:  0000 0000 0000 0000 0000 0000 GGGG GGGG  ==  GGGG GGGG
Shift:                                    << 8
         -----------------------------------------
Result:    0000 0000 0000 0000 GGGG GGGG 0000 0000

dwColor:   AAAA AAAA RRRR RRRR 0000 0000 BBBB BBBB
mask:    | 0000 0000 0000 0000 GGGG GGGG 0000 0000
         -----------------------------------------
result:    AAAA AAAA RRRR RRRR GGGG GGGG BBBB BBBB
```

Reikšmės įrašymas

XOR panaudojimas

Svarbi XOR operacijos savybė:

jei c = a ^ b,

tai a ^ c = b ir b ^ c = a

XOR panaudojimas:

- lyginumo bitai
- diskų masyvai
- šifravimas
- ...

b & y | e