Operações com Máscaras de Bits

Thierson Couto Rosa

24 de março de 2012

Operações lógicas - recordação

a	b	$!a$	$a \;\&\&\; b$	$a \;\|\|\; b$	$a \wedge b$
0	0	1	0	0	0
0	1	1	0	1	1
1	0	0	0	1	1
1	1	0	1	1	0

As operações com sequências de bits são semelhantes, porém são aplicadas entre bits correspondentes de duas sequências. Exemplo: Seja $A == 1010$ e $B == 1100$:

- $A\&B == 1000$ - & é operador *and bit-a-bit*.
- $A|B == 1110$ - — é o *operador or* bit-a-bit.
- $A \wedge B == 0110$ - $\wedge$ é o *operador xor bit-a-bit*.
- $\sim A == 11110101$ - $\sim$ é o operador complemento. O número de bits do resultado depende tipo de A (int, char, etc).

Operadores de deslocamento *Shift-left* ($<<$) e *shift-right* ($>>$)

- $a << b$ - desloca os bits da variável a b bits à esquerda. Equivale a: $a = 2^b$.
- $a >> b$ - desloca os bits da variável a b bits à direita. O novo valor corresponde a $\frac{a}{2^b}$.

bits de valor zero são incluídos a partir da primeira (última) posição á medida que o deslocamento ocorre, se a for positivo.

Uso de máscaras de bits para representar conjuntos

- Um inteiro de 32/64 bits pode ser utilizado para representar um conjunto de até 32/64 elementos.
- Um bit com valor 1 indica que o elemento, correspondente à posição do bit na variável inteira, está presente no conjunto. Se o bit contiver o valor zero, implica que o elemento não está presente no conjunto.

Marcando o i-ésimo elemento como presente no conjunto

$$x| = (1 << i)$$

- Exemplo: $x == 34$ (base 10) $==$ 100010 (base 2), $i == 3$.
- $1 << j == 001000$
- $x = x|(1 << i)$:

$$\begin{array}{rr} & 100010 \\ | & 001000 \\ \hline = & 101010 \end{array}$$

Marcando o i-ésimo elemento como ausente no conjunto

$$x \;\&= \sim (1 << i)$$

- Exemplo: $x == 101010$ (base 2), $j == 3$.
- $1 << j == 001000$
- $x = x \;\& \sim (1 << j)$:

$$\begin{array}{rr} & 101010 \\ \& & 110111 \\ \hline = & 100010 \end{array}$$

Verificando o estado do i-ésimo elemento de um conjunto - ausente/presente

$T = x$ & $(1 << i)$, se T==0, então i-ésimo bit igual zero. Senão, i-ésimo bit igual a1

- Exemplo: $x == 101010$ (base 2), $i == 3$.
- $1 << j == 001000$
- $T = x$ & $(1 << i)$:

$$\begin{array}{rr} & 101010 \\ \& & 001000 \\ \hline = & 001000 \end{array}$$

Invertendo o estado do i-ésimo elemento de um conjunto - ausente/presente

$$x \wedge = (1 << i)$$

- Exemplo: $x == 101010$ (base 2), $i == 3$.
- $1 << j == 001000$
- $x = x \wedge (1 << i)$:

$$\begin{array}{rr} & 101010 \\ \wedge & 001000 \\ \hline = & 100010 \end{array}$$

Operações entre conjuntos

Seja $A == 01010101$, $B == 00011110$, $AllSet = 11111111$

- $A \cup B \rightarrow A|B == 01011111$
- $A \cap B \rightarrow A\&B == 00010100$
- $A - B \rightarrow$
  $A\&(\sim B) == (01010101)\&(11100001) == (01000001)$

Outras operações

Seja $A == 01010100$,

- Encontrar o primeiro bit presente de A a partir da direita: $T = A\&(-A)$

Exemplo:

$$\begin{array}{rll} A & 01010100 & \\ -A & 10101100 & \text{(complemento de dois)} \\ \& & \text{---------} & \\ = & 00000100 & \end{array}$$