Bits Operators

Prof. Salvatore Venticinque
Prof. Emanuele Bellini

Operandi

Il C mette a disposizione degli operatori che lavorano su numeri di tipo intero

(char, int, long int)

manipolando il dato a livello di singolo bit.

Non si applicano ad operandi floating point o a dati di tipo strutturato.

Operatori

&	AND (bit a bit)
|	OR
^	OR Esclusivo (XOR)
~	NOT (complemento a uno)
>>	SHIFT a destra
<<	SHIFT a sinistra

AND

- L'operatore AND & effettua un and, bit a bit, tra due operandi, ovvero
- setta ad 1 un bit in una certa posizione quando entrambi i bit in quella posizione nei due operandi valgono 1, altrimenti lo setta a 0.

```
unsigned char op1, op2, op3;
op1=63;                                  0 0 1 1 1 1 1 1
op2=240;                                  1 1 1 1 0 0 0 0
op3 = op1  &  op2;                        0 0 1 1 0 0 0 0
```

op3 assume valore 48

OR

L'operatore OR | effettua un or bit a bit tra due operandi, ovvero:

setta ad 1 un bit in una certa posizione quando almeno uno dei bit in quella posizione dei due operandi vale 1 altrimenti lo setta a 0.

```
unsigned char
op1, op2, op3;
Op1=60;                    0 0 1 1 1 1 0 0
op2=240;                   1 1 1 1 0 0 0 0
op3 = op1 | op2;           1 1 1 1 1 1 0 0
```

op3 assume valore 252

XOR

L'operatore XOR ^ effettua un or esclusivo bit a bit tra due operandi, ovvero

setta ad 1 un bit in una certa posizione quando i bit in quella posizione nei

due operandi sono diversi, altrimenti lo setta a 0.

```
unsigned char
op1, op2, op3;
Op1=60;                    0 0 1 1 1 1 0 0
op2=240;                   1 1 1 1 0 0 0 0
op3 = op1  ^  op2;         1 1 0 0 1 1 0 0
```

op3 assume valore 204

Operatore NOT

L'operatore NOT ~

- è un operatore unario che inverte lo stato di ogni bit,
- cioè setta ad 1 i bit che valgono 0, e viceversa setta a 0 i bit che valgono 1.

```
op1=60;                    0 0 1 1 1 1 0 0
op3 =  ~op1;               1 1 0 0 0 0 1 1
```

op3 assume valore 195

Operatori di SHIFT

- Gli operatori di scorrimento (SHIFT) sono operatori unari che realizzano lo spostamento a sinistra o a destra dei bit di una variabile di tipo intero, mettendo a 0 i bit che entrano.

- La forma genera le dell'istruzione di SHIFT A DESTRA è del tipo:

variabile_intera >> numero_posizioni

Significa che i bit di variabile_intera vengono tutti spostati a destra di numero_posizioni.

```
op1=61;                  0 0 1 1 1 1 0 1
op3 =  op1>>1;           0 0 0 1 1 1 1 0
```

op3 assume valore 30 (in neretto il bit entrante)

Si noti come l'operatore corrisponda ad una divisione intera per multipli di 2.

```
op3 =  op1>>3;           0 0 0 0 0 1 1 1
```

op3 assume valore 7 (in neretto i bits entranti)

SHIFT a Sinistra

- La forma generale dell'istruzione di SHIFT A SINISTRA è del tipo: variabile_intera << numero_posizioni

- Significa che i bit di variabile_intera vengono tutti spostati a sinistra di numero_posizioni. I bits entranti da destra sono posti a 0.

```
op1=61;              0 0 1 1 1 1 0 1
op3 =  op1<<1;       0 1 1 1 1 0 1 0
```

op3 assume valore 122 (in neretto il bit entrante)

Notare come questo operatore corrisponda al risultato di una moltiplicazione Intera per un multiplo di due finché a sinistra non esce qualche bit ad 1.

```
op3 =  op1<<3;         1 1 1 0 1 0 0 0
```

op3 assume valore 232 (in neretto i bits entranti)

Operatori compositi

- Il C mette a disposizione anche operatori che coniugano un'operazione bit a bit all'operazione di assegnamento.

- AND bit a bit e Assegnamento

  `x &=  y equivale a x = x&y`

- OR bit a bit e Assegnamento

  `x |=  y equivale a x = x|y`

- XOR bit a bit e Assegnamento

  `x ^=  y  equivale a x = x^y`

- Shift a Destra e Assegnamento

  `x  >>=  y equivale a x = x>>y`

- Shift a Sinistra e Assegnamento

  `x  <<=  y equivale a x = x<<y`

Esercizi

- Azzerare il terzo bit d un intero
- Controllare il terzo bit di un intero.
- Visualizzare la rappresentazione di un intero
- Contare il numero di bit alti
- Trovare l’intero immediatamente più grande di n che abbia stesso numero di bit alti