
# Προγραμματισμός Υπολογιστών

## Δυαδικοί τελεστές

**Νικόλαος Ζ. Ζάχαρης**
**Καθηγητής**

**Πανεπιστήμιο Δυτικής Αττικής**
**Τμήμα Μηχανικών Πληροφορικής και Υπολογιστών**

# Τελεστές επιπέδου bit

Οι τελεστές & | ^ << ~ >>εφαρμόζονται bit προς bit στη δυαδική αναπαράσταση των αριθμών. Αν το x = 5 (0000 0101) και το y = 3 (0000 0011)

| Τελεστής | Περιγραφή | Αποτέλεσμα |
|---|---|---|
| & | Σύζευξη των bits. Το αποτέλεσμα είναι 1 μόνο αν και τα δυο bit είναι 1. | x & y = 0000 0001 |
| \| | Διαζευξη των bits. Το αποτέλεσμα είναι 1 μόνο αν ένα από τα δυο bit είναι 1. | x \| y = 0000 0111 |
| ^ | Αποκλειστική διαζευξη των bits. Το αποτέλεσμα είναι 1 μόνο αν ένα από τα δυο bit είναι 1 αλλά όχι και τα δύο. | x ^ y = 0000 0110 |
| << | Αριστερή ολίσθηση των bits. Οι κενές θέσεις που προκύπτουν στα δεξιά γεμίζουν με μηδενικά ενώ "χάνονται" τα bits που βγαίνουν εκτός του εύρους του αριθμού. | x << 2 = 0001 0100 |
| >> | Δεξία ολίσθηση των bits. Οι κενές θέσεις που προκύπτουν στα αριστερά γεμίζουν με μηδενικά ενώ "χάνονται" τα bits που βγαίνουν εκτός του εύρους του αριθμού. | x >>2 = 0000 0001 |
| ~ | Άρνηση σε επίπεδο bit. Αντιστρέφει τα 0 σε 1 και το αντίστροφο. | ~x = -6 = 11111010 |

# Εφαρμογή για τη χρήση των τελεστών

[*] main.c

```c
#include <stdio.h>
#include <stdlib.h>

int main(int argc, char *argv[]) {

    unsigned int a = 80; /* 80 = 0101 0000 */
    unsigned int b = 55; /* 55 = 0011 0111 */
    int apot = 0;

    apot = a & b;         /* 16 = 0001 0000 */
    printf("a & b is %d\n", apot );

    apot = a | b;        /* 119 = 0111 0111 */
    printf("a | b is %d\n", apot );

    apot = a ^ b;        /* 103 = 0110 0111 */
    printf("a ^ b is %d\n", apot );

    printf("\na = %d\n",a);
    apot = ~a;           /*-81 = (1)1010 1111 */
    printf("~a is %d\n", apot );

    apot = a << 2;     /* 320 = 0001 0100 0000 */
    printf("a << 2 is %d\n", apot );

    apot = a >> 2;     /*  20 =  0001 0100 */
    printf("a >> 2 is %d\n", apot );

    return 0;
}
```

# Χρήση των bitwise τελεστών (1-4)

Αν θεωρήσουμε ότι έχουμε τη δυνατότητα να εφαρμόσουμε σε ένα κείμενο τα παρακάτω εφέ, τότε αντί ν α έχουμε ξεχωριστές μεταβλητές για το καθένα μπορούμε εναλλακτικά να τα έχουμε όλα σε μια μεταβλητή.

| ΕΦΕ | ΔΕΚΑΔΙΚΗ | ΔΥΑΔΙΚΗ |
|---|---|---|
| shadow | 128 | 10000000 |
| blinking | 64 | 01000000 |
| lower | 32 | 00100000 |
| upper | 16 | 00010000 |
| strikethrough | 8 | 00001000 |
| underline | 4 | 00000100 |
| italic | 2 | 00000010 |
| bold | 1 | 00000001 |

Οπότε θα μπορούσε να συσχετίσουμε ένα κείμενο με μια μεταβλητή στην οποία θα αποθηκεύουμε τα εφέ του. Οπότε αρχικά το κείμενο δεν έχει κανένα εφέ

unsigned int efe = 0;

Αν εφαρμόσουμε το εφε shadow θα μπορούσαμε μόνο σε αυτή τη περίπτωση, επειδή το efe έχει τιμή 0 να χρησιμοποιήσουμε το τελεστή της ανάθεσης = ή εναλακτικά το|

efe = 128;  ή  efe = efe | 128;

Αν επιπλέον εφαρμόσουμε το εφε italic

efe = efe | 2 ;  Οπότε η μεταβλητή efe έχει τη τιμή 130 (10000010)

# Χρήση των bitwise τελεστών (2-4)

Αν στη συνέχεια θέλουμε να ελέγξουμε ότι το κείμενο έχει efe italic τότε μπορούμε να χρησιμοποιήσουμε το τελεστή &

int apot = efe & 2 ;

Αν η efe έχει το εφέ italic τότε η apot έχει τη τιμή 2 (τη τιμή του εφέ) διαφορετικά έχει τη τιμή 0.

Αν στη συνέχεια θέλουμε να απενεργοποιήσουμε το efe italic χρησιμοποιούμε το τελεστή ^

efe = efe ^ 2 ;

Πλέον η efe έχει τη τιμή 128;

# Χρήση των bitwise τελεστών (3-4)

[*] main.c

```c
#include <stdio.h>
#include <stdlib.h>

#define SHADOW 128
#define LOWER  32
#define ITALIC 2

int main(int argc, char *argv[]) {
    unsigned int efe = 0;
    // add effect SHADOW
    efe = SHADOW;      //equivalent efe = efe | SHADOW;

    // add effect italic
    efe = efe | ITALIC;
    printf("Combination of effects %d\n", efe);

    // contains effect italic?
    int apot = efe & ITALIC;
    printf("The text has effect italic  %d\n", apot);

    // contains effect lower?
    apot = efe & LOWER;
    printf("The text has effect lower  %d\n", apot);

    // remove effect italic
    efe = efe ^ ITALIC;
    printf("Combination of effects %d\n", efe);
    return 0;
}
```

```
Combination of effects 130
The text has effect italic  2
The text has effect lower  0
Combination of effects 128
```

# Χρήση των bitwise τελεστών (4-4)

[*] main.c

```c
#include <stdio.h>
#include <stdlib.h>

int main(int argc, char *argv[]) {

  int x = 3200;
  printf("x / 2 = %d\n", x >> 1);
  printf("x / 4 = %d\n", x >> 2);
  printf("x / 8 = %d\n", x >> 3);

  x = 5;
  printf("x * 2 = %d\n", x << 1);
  printf("x * 4 = %d\n", x << 2);
  printf("x * 8 = %d\n", x << 3);

  return 0;
}
```

```
x / 2 = 1600
x / 4 = 800
x / 8 = 400

x * 2 = 10
x * 4 = 20
x * 8 = 40
```