



Codage Types en C Paramètres en C

Patrick FULCONIS



Codage - bases

Conversion décimal - binaire

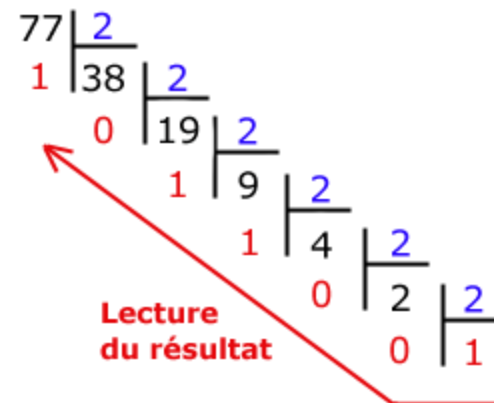
Convertissons 01001101 en décimal à l'aide du schéma ci-dessous:

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
0	1	0	0	1	1	0	1

Le nombre en base 10 est $2^6 + 2^3 + 2^2 + 2^0 = 64 + 8 + 4 + 1 = 77$.

Dans l'autre sens : écrivons 77 en base 2. Il s'agit de faire une suite de divisions euclidiennes par 2. Le résultat sera la juxtaposition des restes.

77 s'écrit donc en base 2: 1001101.





Codage - bases

Conversion hexadécimal - binaire

01001101 en hexadécimal : Il suffit de regrouper les bits par 4 (en commençant depuis la droite):

Binaire	0100	1101
Pseudo-décimal	4	13
Hexadécimal	4	D

1001101 s'écrit donc en base 16 : 4D

Pour aller dans l'autre sens, on lit le tableau de bas en haut.

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

Conversion hexadécimal - décimal

15AACF7 en décimal :

$$1 \times 16^6 + 5 \times 16^5 + 10 \times 16^4 + 10 \times 16^3 + 12 \times 16^2 + 15 \times 16^1 + 7 \times 16^0 = 22719735$$



Codage

Codage des entiers

Les entiers sont écrits en base 2

C'est le codage utilisé à l'intérieur des machines pour faire des opérations sur les entiers MAIS les processeurs ne savent effectuer des opérations que sur des entiers codés sur un nombre fixe de bits (8, 16, 32 ...).

Les langages de programmation choisissent donc également de représenter les entiers sur des tailles fixes (correspondant à celles offertes par le processeur). Seul un (petit) sous-ensemble est donc considéré ...

Quel est le plus grand entier naturel que l'on peut écrire en base 2 sur p bits ?

(réponse : $2^p - 1$)



Codage

Codage des caractères

Codage standard pour représenter l'ensemble des caractères : le code Ascii
Chaque caractère est codé par un entier unique (entre 0 et 127).

Question : combien de bits sont-ils nécessaires pour coder les caractères ?

Par exemple :

'A' a pour code 65 (en decimal), soit 0100001 en binaire

'2' a pour code 50 (en decimal), soit 0110010 en binaire

' ' a pour code 32 (en décimal), soit 0010000 en binaire

Les lettres majuscules (et minuscules, chiffres) se suivent : 'D'='A'+3, '5'='2'+3
(cf *man ascii*)

Remarque: il existe diverses extensions à 8 bits de ce code pour représenter les caractères accentués, etc.

Tout texte qui ne contient que des caractères du code Ascii peut donc être codé en concaténant les codes Ascii des caractères qui le composent.



Types en C

Types en C

La notion de type permet de définir dans un programme à la fois des domaines de valeurs et des opérations permises sur ces domaines.

En C il existe des types de base (fournis par le langage), et des constructeurs de type (permettant au programmeur de définir de nouveaux types).

Parmi les types de base :

Entiers (int) :

opérateurs : +, -, *, / (division entière !), % (modulo), ==, <=, >=, <, >, !=

différentes "variantes" : **int** (entiers relatifs),
unsigned int (entiers naturels), (unsigned) short int (entiers "courts")



Types en C

Caractères (char) :

domaine de valeurs : un ensemble (fini !) de caractères ;
les caractères sont codés sur 8 bits, et la valeur de l'entier correspondant est le code Ascii ou Ascii étendu du caractère :

opérateurs : ==, <=, >=, <, >, !=,

mais aussi des opérations arithmétiques :

'a' + 1 est le caractère 'b'

'x' - 2 est le caractère 'v'



Types en C

Tableaux

La notion de "tableau" permet de regrouper sous un même nom une séquence de N éléments de même type, chaque élément étant repéré (indiqué) par un numéro unique entre **0** et **N-1**.

ex : `int T[7]` est une séquence de 7 entiers notés `T[0]`, `T[1]`, `T[2]`, .. `T[6]`

Rq : on ne peut pas appliquer des opérateurs "globalement" sur l'ensemble des éléments d'un tableau :

```
int T1[5], T2[5] ;
```

`T1 + T2`, `T1 == T2`, etc. ne sont pas définis.



Types en C

Chaînes de caractères

Le cas des **tableaux de caractères** est un peu spécial (on parle de "chaînes de caractères") :

- le langage fournit des opérateurs permettant de manipuler plus facilement des chaînes de caractères (copie de chaîne, comparaison, etc)
 - il existe une notation pour les chaînes "constantes"
 - > il faut pour cela respecter une convention :
 - le tableau de caractères doit être terminé par un caractère spécial, le caractère "nul" noté `'\0'`
 - (cette convention permet de repérer la fin de la chaîne).
- ex : "bonjour" désigne un tableau contenant les caractères
`'b', 'o', 'n', ..., 'r', '\0'`

Rq : pour mémoriser une chaîne de n caractères il faut donc prévoir un tableau de n+1 caractères



Codage

Paramètres en C

Passage de paramètres en C :

Passage par valeurs :

```
void echange (int a, int b) {  
    int c ;  
        c = a ;  
        a = b ;  
        b = c ;  
}
```

avec un appel comme :

```
x= 42 ;  
y= 13 ;  
echange(x, y) ;  
printf("maintenant x vaut %d et y vaut %d \n", x, y) ;
```



Codage

Paramètres en C

Passage de paramètres en C :

Passage par **variables** (adresses, pointeurs) :

```
void echange(int *pa, int *pb) {  
    int c ;  
    c = *pa ;  
    *pa = *pb ;  
    *pb = c ;  
}
```

et l'appel :

```
x = 42 ;  
y = 13 ;  
echange(&x, &y) ;
```