

Chapitre 3

Compléments sur les types, opérateurs, expressions

1. Conversion de type

Il y a conversion de type lorsqu'une expression fait intervenir des éléments de types différents. En C elles sont largement permises.

Conversions implicites

Expression.

Lorsqu'il y a mélange de types dans une expression, les conversions implicites se font en 2 étapes:

- 1) Les types entiers plus petits que *int* sont convertis en *int*, le type *float* est converti en *double*.
- 2) Si une des opérandes est *double*, l'autre est converti en *double* et le résultat est un *double*. Sinon, si une des opérandes est *long*, l'autre est converti en *long* et le résultat est *long*.

Exemple :

```
int i, j;
long l;
float f;

          i      *      (  l      +      f      )
1)                long      double
2)                double     double
resultat                                double
1)          int      double
2)          double   double
resultat                double
```

Affectation : *destination = source ;*

Cas 1 : types entiers

- Si source est plus importante que destination \Rightarrow perte d'informations

Ex : *int = long ;*
 char = short|int|long ;
 short = int|long ;

- Si destination + importante que source \Rightarrow ok

Cas 2 : types entiers et réels

- destination entière, source réelle \Rightarrow décimales sont supprimées (troncation)

Ex : `char|short|int|long=float|double ;`

```
int x ;
x = 3.89 ;
// x vaut 3
```

- destination réelle, source entière \Rightarrow ok

Ex : `float|double =char|short|int|long;`

Cas 3 : simple précision / double précision

- destination simple précision, source double \Rightarrow perte d'information (précision)

Ex : `float = double ;`

- destination double précision, source simple \Rightarrow ok

Conversions explicites :

On peut forcer explicitement le type d'une variable : casting
(type) expression ou type (expression)

(int) 3.5 \Rightarrow le résultat est 3

(float) 5 \Rightarrow 5.0

int (3.5) \Rightarrow le résultat est 3

float (5) \Rightarrow 5.0

\rightarrow plus lisible

\rightarrow par obligation

Exemple :

```
int x, y;
float f;

x = 23;
y = 5;
f = x/y;    /* x/y vaut 4 et f vaut 4.0 */

f = (float)x /y;    /* (float) x vaut 23.0
                    x vaut toujours 23
                    y vaut 5
                    le resultat sera un
                    double 4.6
                    converti en float
                    f = 4.6 */

f = x /float(y) ;    /* f = 4.6 * /

f = (float)x /(float)y ; /* f = 4.6 * /

/* interdit      f= x./y.   car x et y sont des
identificateurs et pas des constantes */
```

Utilisé lorsque des fonctions attendent des arguments de type donné.

```
z = sin ((double) x) ;
```

On peut aussi forcer le type pointeur.