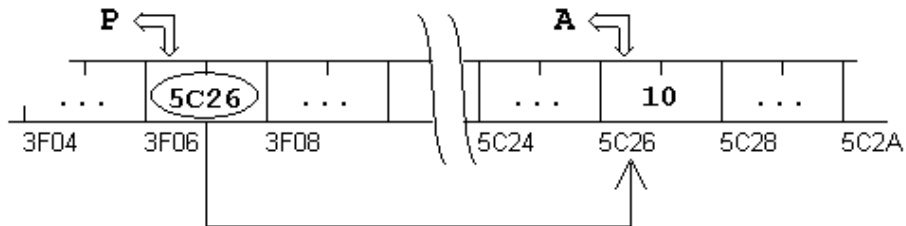


Programmation C Les Pointeurs

1 Adressage indirect et mémoire centrale

La mémoire centrale d'un ordinateur est composée d'un grand nombre d'octets. Chaque octet est repéré par un numéro appelé **adresse** de l'octet.

Chaque variable occupe dans la mémoire des octets contigus, par exemple un **float** occupe 4 octets.



L'adresse d'une variable :

```
int A=10 ;  
  
&A // représente l' adresse de la variable A (5c26)
```

2 variable de type pointeur

L'adresse d'une variable peut elle-même être mémorisée dans une variable. Les variables dont les valeurs sont des adresses s'appellent des **pointeurs**.

3 déclaration d'un pointeur

```
int *p = NULL; /* déclaration d' un pointeur p  
*/
```

l'étoile devant p sert juste à indiquer qu'on veut créer un pointeur .

Pour initialiser un pointeur, c'est-à-dire lui donner une valeur par défaut, on n'utilise généralement pas le nombre 0 mais le mot-clé **NULL** (veillez à l'écrire en majuscules) :

4 Affectation d'un pointeur

```
p = &A ;
```

L'adresse de la variable est transférée dans le pointeur p. Si un pointeur **P** contient l'adresse d'une variable **A**, on dit que **'P pointe sur A'**.

5 Valeur d'une variable pointée

Nous pouvons retrouver l'information de la variable **A** en passant par le pointeur **P**.

*p représente la valeur de la variable sur laquelle pointe p

```
int b ;  
  
b = *p ; // b contient la valeur de A soit 10
```

6 Exemple fonction pour inverser le contenu de 2 variables

```
int echange (int *a,int *b){  
    int temp;  
    temp = *a;  
    *a = *b;  
    *b = temp;  
    return 0;  
}
```