

# Introduction au Langage C,C++

## \* Les Pointeurs

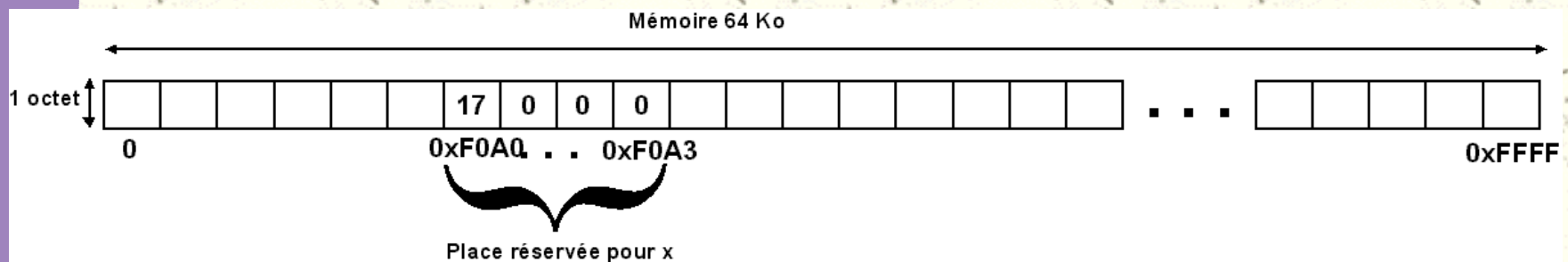
### ■ Deux manières d'utiliser une variable

#### ■ Par son nom → adressage direct

\* compilateur réserve de la mémoire pour la variable

\* exemple:

```
int x = 17;
```



# Introduction au Langage C, C++

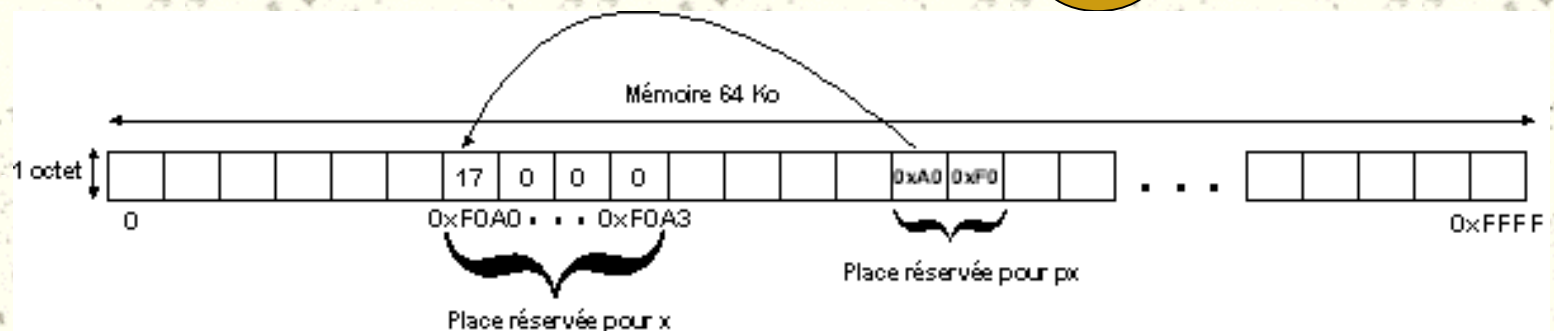
## \* Les Pointeurs

- Par son adresse → adressage indirect
  - \* une variable de type pointeur contient l'adresse d'une autre variable
  - \* Le lien entre pointeur et la variable pointée est géré par le programmeur
  - \* exemple:

```
int x = 17;
```

```
int * px = &x;
```

px pointe  
vers x



# Introduction au Langage C,C++

## \* Les Pointeurs

- ▣ déclaration d'un pointeur

- ▣ `type * nom_du_pointeur`

- ▣ exemples

- ▣ `char *buffer;`

- ▣ `int *pf;`

- ▣ `float *ppi;`



C'est au programmeur d'initialiser le pointeur



# Introduction au Langage C,C++

## \* Les Pointeurs

### ■ initialisation d'un pointeur

- Il faut lui fournir l'adresse d'une variable existant en mémoire
- Il doit pointer vers une zone mémoire réservée par le compilateur comme étant une variable

### ■ utilisation de l'opérateur d'adressage &

```
int f;  
int *pf,*pg;  
pf = &f;
```

création du  
lien entre le  
pointeur et la  
variable

```
pg =(int*)malloc(sizeof(int));
```

allocation  
dynamique  
d'une variable  
de type int

création du  
lien entre pg  
et l'espace  
mémoire  
réservé

```
free(gf);
```

libération de  
l'espace  
réservé

# Introduction au Langage C,C++

## # Les Pointeurs

- utilisation de l'opérateur d'indirection \*
- travaille la valeur de la variable pointée
- exemples

```
int i = 8;  
int *pi;  
pi = &i;  
cout << *pi << '\n';
```

8

```
*pi = 19;  
cout << i << '\n';
```

19