



Haute école d'ingénierie et d'architecture Fribourg
Hochschule für Technik und Architektur Freiburg

Systèmes Embarqués 1 & 2

a.08 – C – Les pointeurs de fonctions

Classes T-2/I-2 // 2018-2019



Contenu

- **Déclaration**
- **Assignment et appel**
- **Tableaux**



Les pointeurs de fonctions sont des éléments très puissants du langage de programmation C. Le pointeur de fonction est une variable qui contient l'adresse d'une fonction (point d'entrée de la fonction).

Les pointeurs de fonctions sont très souvent utilisés en C car ils permettent de:

- ▶ Concevoir des interfaces avec des fonctions de rappel (callback)
- ▶ Concevoir des algorithmes génériques pour traitement de données (OO)

La déclaration d'un pointeur de fonction prend la forme suivante:

```
int(*function) (int, int);
```

Il est souvent pratique d'utiliser un `typedef` pour déclarer un pointeur de fonction, p.ex.:

```
typedef int(*function_t) (int, int);  
function_t function;
```



L'assignation d'un pointeur de fonction peut prendre les formes suivante:

```
function = my_function
```

ou

```
function = &my_function
```

Pour appeler un pointeur de fonction, il suffit d'utiliser la syntaxe d'un appel normal de fonction, p. ex:

```
int value = function (10, 20);
```

il est également possible d'utiliser la syntaxe utilisée avec les pointeurs:

```
int value = (*function) (10, 20);
```



Retourner un pointeur de fonction

Bien naturellement, il est possible que l'appel d'une fonction retourne un pointeur de fonction.

Il existe deux manières de déclarer de telles fonctions:

```
int (*get_function(int which_one)) (int, int);
```

ou

```
typedef int (*function_t) (int, int);  
function_t get_function (int which_one);
```



Tableau de pointeurs de fonction

La déclaration d'un tableau de pointeurs de fonction peut prendre les formes suivantes :

```
int(*functions[20]) (int, int);
```

ou

```
typedef int(*function_t) (int, int);  
function_t functions[20];
```

L'appel de la fonction:

```
int value = functions[1] (10, 20);
```