Informatique

Cours

Comparaison des langages C et OCaml

Ipesup

Les langages C et OCaml partagent un certain nombre de concepts, en particulier des constructions de programmation impérative (boucles, affectations, séquences, blocs) et un mode de passage limité au passage par valeur. Ils se distinguent néanmoins par beaucoup d'aspects qu'il est intéressant de résumer ici.

— L'exécution d'un programme C démarre à la fonction `main`.
En OCaml, on évalue les déclarations dans l'ordre où elles apparaissent dans le fichier.

— Dans la syntaxe de C, le point-virgule fait partie de la syntaxe d'une *instruction*.
Dans la syntaxe d'OCaml, le point-virgule est un opérateur binaire qui combine deux *expressions*.

— En C, l'exécution d'une fonction se termine sur l'instruction `return` ou à la fin du flot d'exécution pour un type de retour `void`.
En OCaml, le corps de la fonction est une unique expression dont la valeur est renvoyée.

— Les pointeurs d'OCaml ne sont pas explicites. En OCaml, il n'y a pas de valeur `NULL` ni d'équivalent de l'opérateur `&` de C.
Un pointeur OCaml pointe toujours sur une valeur bien formée et son déréférencement ne peut pas échouer.

— L'allocation dynamique en C se fait au travers de la fonction `malloc` et la libération est explicite avec `free`.
En OCaml, la gestion de la mémoire est assurée automatiquement par le GC.

— En C, une structure ou un tableau peut être alloué sur la pile. Le programmeur le déclare explicitement et il est conscient des risques liés au fait de continuer à utiliser cette donnée au-delà du retour de la fonction.
En OCaml, les données sont allouées par le GC (en général sur le tas) et ce risque n'existe pas.

— Une valeur OCaml est toujours atomique, occupant exactement un mot mémoire (64 bits).
Une valeur C, en revanche, peut être une structure de taille arbitraire, qui est intégralement copiée lorsqu'elle est passée en paramètre, renvoyée par une fonction ou affectée à une variable.

— L'espace de noms est plat en C : il faut écrire `list_length`, `queue_length`, etc.
En OCaml, l'espace de noms est structuré : à l'intérieur du module `List`, on peut définir une fonction `length` ; à l'extérieur, on y fait référence via `List.length`. On peut ainsi aussi avoir `Queue.length`, `Array.length`, etc.

— Les variables OCaml sont immuables alors que les variables C sont mutables.
Les références d'OCaml s'apparentent à des variables mutables mais leur comportement diffère de celui des variables mutables de C.