

2. Langage de plus haut niveau et compilation

Bonjour, bienvenue dans la séquence sur les langages de haut niveau et la compilation de la partie "Programmation".

La compilation, qu'est-ce que c'est? Quand on veut écrire un programme, on a d'abord une idée, qu'est-ce qu'on veut calculer, qu'est-ce qu'on veut faire? Après, on pense aux grandes lignes, c'est-à-dire une espèce de séquençage, comme on a vu dans la partie algorithmique. Après, on choisit l'algorithme, on trouve l'algorithme et on décide des structures de données. Après, on va écrire le programme dans un langage de son choix, on en reparlera dans la séquence suivante. Et après, on a besoin du langage machine. La compilation, c'est exactement cette étape-là : la partie qui va du programme au langage machine donc du langage de programmation, qui est compréhensible par un humain vers le langage machine, qui est compréhensible par la machine et pas forcément par l'humain.

Qu'est-ce que c'est un compilateur? Le compilateur, c'est un programme comme les autres. Qu'est-ce qu'il fait? En première approximation, il va prendre un texte dans un langage précis, par exemple C ou OCaml et il faut que ce texte corresponde exactement à ce qu'on attend d'un programme, un certain nombre de règles d'écriture de programme. Et qu'est-ce qu'il fait à partir de ce texte? Il le transforme en un autre texte, en assembleur ou en exécutable, qui, lui, est compréhensible par le processeur.

Donc ça, c'est pour les langages compilés par contre on a aussi des langages interprétés où là c'est l'interpréteur qui parle directement à la machine et qui est capable de considérer un texte et de le faire exécuter par la machine. Qu'est-ce que fait de plus le compilateur? En fait, il va vérifier un certain nombre de choses. Donc, c'est une première vérification automatique que votre programme n'est pas n'importe quoi. Donc, il vérifie qu'il n'y a pas d'erreurs de syntaxe, que vous avez bien mis les points-virgules au bon endroit, que vous avez fermé les parenthèses, ce genre de choses. Il va vérifier aussi, mais, ça, ça dépend du langage, qu'il n'y a pas d'erreur de type, que par exemple vous n'avez pas ajouté une liste et un entier ou ce genre d'erreurs qui sont simples à trouver pour un compilateur, qui sont automatiques à trouver. Et ça peut aussi vérifier un certain nombre de critères simples qui sont syntaxiques, par exemple vérifier qu'une variable est utilisée ou ce genre de choses. Donc, ça peut émettre un certain nombre d'avertissements qui vous disent : "je ne sais pas si tu t'es trompé, mais là c'est quand même bizarre que cette variable-là tu la declares, mais tu ne l'utilises pas après". Bref, il vérifie au moins que votre programme peut être transformé en un exécutable, c'est-à-dire que ce n'est pas vraiment n'importe quoi. Par contre, ce qu'il ne fait pas, c'est être sûr qu'il fait ce que vous, vous voulez. Parce qu'effectivement vous écrivez un programme, vous le décrivez très précisément, mais vous n'êtes pas sûr qu'il va faire exactement ce que vous voulez et c'est l'objet de la prochaine séquence sur les bugs.