Programmation impérative

Elle consiste à décrire les opérations en séquences d'instructions exécutées par l'ordinateur pour modifier l'état du programme. Ce type de programmation est le plus répandu dans l'ensemble des langages de programmation existants.

La quasi-totalité des processeurs sont de nature impérative : ils exécutent une suite d'instructions élémentaires, codées sous forme d'opcodes (operation codes). Ces opcodes forment le langage machine spécifique à l'architecture du processeur. L'état du programme à un instant donné est défini par le contenu de la mémoire centrale à cet instant.

Les langages de plus haut niveau utilisent des variables et des opérations plus complexes, mais suivent le même paradigme. Les recettes de cuisine et les vérifications de processus industriel sont deux exemples de concepts familiers qui s'apparentent à de la programmation impérative ; de ce point de vue, chaque étape est une instruction, et le monde physique constitue l'état modifiable. Puisque les idées de base de la programmation impérative sont à la fois conceptuellement familières et directement intégrées dans l'architecture des microprocesseurs, la grande majorité des langages de programmation est impérative.

source Wikipedia

Instructions de la base impérative

La plupart des langages de haut niveau comporte cinq types d'instructions principales :

1. la séquence d'instructions
2. l'assignation ou affectation
3. l'instruction conditionnelle
4. la boucle
5. les branchements

Séquence d'instructions

Une séquence d'instructions, (ou bloc d'instruction) désigne le fait de faire exécuter par la machine une instruction, puis une autre, etc., en séquence, par opposition au fait d'exécuter en parallèle des instructions.

```
fd = open(file, O_RDONLY);
json = mmap(NULL, st.st_size, PROT_READ, MAP_PRIVATE, fd, 0);
close(fd);
obj = json_tokener_parse(json);
json_object_object_foreach(obj, key, val) {
    printf("key = %s value = %s\n",key, json_object_get_string(val));
}
```

Instructions d'assignation

Les instructions d'assignation effectuent une opération sur l'information en mémoire et y enregistrent le résultat pour un usage ultérieur.

```
int a = 512;
double prod = 55. * a;
```

Instructions conditionnelles

Les instructions conditionnelles permettent à un bloc d'instructions de n'être exécuté que si une condition prédéterminée est réalisée. Dans le cas contraire, les instructions sont ignorées et la séquence d'exécution continue à partir de l'instruction qui suit immédiatementla fin du bloc.

```
if (condition) {
    //A faire si condition est vraie
} else {
    //Si condition est fausse
}
```

Instructions de bouclage

Les instructions de bouclage servent à répéter une suite d'instructions un nombre prédéfinide fois (for/foreach), ou jusqu'à ce qu'une certaine condition soit réalisée (while).

```
while (condition) {
    //Instructions à exécuter tant que condition est vraie
}

for (int i=0; i<size; i++) {
    //Suite d"instructions à exécuter size fois
}
```

Ces quatre constructions permettent de faire tous les programmes informatiques possibles (système Turing-complet).

Branchements sans condition

Les branchements sans condition permettent à la séquence d'exécution d'être transférée à un autre endroit du programme. Cela inclut le saut, appelé « goto » dans de nombreux langages, et les sous-programmes, ou appels de procédures. Les instructions de bouclage peuvent être vues comme la combinaison d'un branchement conditionnel et d'un saut. Les appels à une fonction ou une procédure (donc un Sous-programme) correspondent à un saut, complété du passage de paramètres, avec un saut en retour.

From:
http://slamwiki2.kobject.net/ - **SlamWiki 2.1**

Permanent link:
http://slamwiki2.kobject.net/cnam/utc503/imperative

Last update: **2024/08/20 07:13**