Franck CELLIER

Formateur en Informatique

Technicien Qualité d'Organisme de Formation

http://cellierfranck.alwaysdata.net

MS LANGAGE & DEVELOPPEMENT POUR LE WEB

NOTIONS DE PROGRAMMATION

A. Définition

Un **programme informatique** est un ensemble d'opérations destinées à être exécutées par un ordinateur.

- Un **programme source** est un code écrit par un informaticien dans un langage de programmation. Il peut être compilé vers une forme binaire, ou directement interprété.
- Un **programme binaire** décrit les instructions à exécuter par un microprocesseur sous forme numérique. Ces instructions définissent un langage machine.

Un programme fait généralement partie d'un logiciel : un ensemble de composants numériques destiné à fournir un service informatique ; un logiciel peut comporter plusieurs programmes. On en retrouve ainsi dans les appareils informatiques (ordinateur, console de jeu, guichet automatique bancaire…), dans des pièces de matériel informatique, ainsi que dans de nombreux dispositifs électroniques (imprimante, modem, GPS, téléphone mobile, machine à laver, appareil photo numérique, décodeur TV numérique, injection électronique, pilote automatique…).

Les programmes informatiques sont concernés par le droit d'auteur, et font l'objet d'une législation proche des œuvres artistiques.

B. Langages de programmation

Un langage de programmation est une notation utilisée pour exprimer des algorithmes et écrire des programmes. Un algorithme est un procédé pour obtenir un résultat par une succession de calculs, décrits sous forme de **pictogrammes** et **de termes simples** dans une **langue naturelle**. Jusqu'en 1950, les programmeurs exprimaient les programmes dans des langages machines ou assembleur, des langages peu lisibles pour des êtres humains et ou chaque instruction fait peu de choses, ce qui rendait le travail pénible et le résultat sujet à de nombreuses erreurs. Dès 1950, les programmes ont été décrits dans des langages différents dédiés à l'humain et non plus à la machine, des langages de programmations, ce qui rendait les opérations plus simples à exprimer. Le programme était ensuite traduit automatiquement sous une forme qui permet d'être exécuté par l'ordinateur.

2 rue Gabriel FAURÉ - 59150 WATTRELOS - fcellier34@gmail.com - ☎ 06 41 12 70 19 - www.linkedin.com/in/formateur-franck-cellier

Sur demande, l'ordinateur exécutera les instructions du programme. Bien qu'il exécute toujours exactement ce qui est instruit et ne se trompe jamais, il peut arriver que les instructions qu'il exécute soient erronées à la suite d'une erreur humaine lors de l'écriture du programme. Les langages de programmation visent à diminuer le nombre de ces **bugs** ; ceux-ci sont cependant inévitables dans des programmes de plusieurs milliers de lignes. Un programme de **traitement de texte** peut être fait de plus de 750 000 lignes de code, et un **système d'exploitation** peut être fait de plus de 50 millions de lignes. En moyenne un programmeur prépare, écrit, teste et documente environ 20 lignes de programme par jour, et la création de grands programmes est le fait d'équipes et peut nécessiter plusieurs mois voire plusieurs années.

La programmation est un sujet central en informatique. Les instructions qu'un ordinateur devra exécuter doivent pouvoir être exprimées de manière précise et non ambiguë. Pour ce faire, les langages de programmation combinent la lisibilité de l'anglais avec l'exactitude des mathématiques. Les programmes sont créés par des programmeurs, ou des ingénieurs logiciels. La création d'un programme comprend une série d'activités telles que la conception, l'écriture, le test et la documentation. En vue d'obtenir un programme de meilleure qualité, le travail de programmation se fait selon une démarche systématique et planifiée.

Un langage de programmation est un vocabulaire et un ensemble de règles d'écriture utilisées pour instruire un ordinateur d'effectuer certaines tâches. La plupart des langages de programmation sont dits **de haut niveau**, c'est-à-dire que leur notation s'inspire des **langues naturelles** (généralement l'anglais).

Le processeur est le composant électronique qui exécute les instructions. Chaque processeur est conçu pour exécuter certaines instructions, dites **instructions machine**. La palette d'instructions disponibles sur un processeur forme **le langage machine**. Par exemple, le processeur **Intel 80486** a une palette de 342 instructions.

Le langage d'assemblage est une représentation textuelle des instructions machine, **un langage de bas niveau**, qui permet d'exprimer les instructions machine sous une forme symbolique plus facile à manipuler, où il y a une correspondance 1-1 entre les instructions machines et les instructions en langage d'assemblage.

Les langages de programmation de haut niveau permettent d'exprimer des instructions de manière synthétique, en faisant abstraction du langage machine. Par rapport au langage d'assemblage, ils permettent d'exprimer des structures, permettent d'écrire des programmes plus rapidement, avec moins d'instructions, les programmes écrits dans des langages de haut niveau sont plus simples à modifier et portables — ils peuvent fonctionner avec différents processeurs. Cependant un programme exprimé en langage de haut niveau, puis compilé est moins efficace et comporte plus d'instruction que s'il avait été exprimé en langage d'assemblage.

Entre 1950 et 2000, plus de 50 langages de programmation sont apparus. Chacun apportait un lot de nouveaux concepts, de raffinements et d'innovations. Jusque dans les années 1950, l'utilisation des langages de programmation était semblable à l'écriture d'instructions machines. L'innovation des années 1960 a été de permettre une notation proche des mathématiques pour écrire des instructions de calcul. Les innovations des années 1970 ont permis l'organisation et l'agrégation des informations manipulées par les programmes — voir **structure de données** et **structure de contrôle**. Puis l'arrivée de la notion d'**objet** a influencé l'évolution des langages de programmation postérieurs à 1980.

Ci-dessous, le programme **Hello world** exprimé en langage de programmation **Java** :

```java
public class HelloWorld {
  public static void main(String[] args) {
    System.out.println("Hello world!");
  }
```

```
}
```

Le même programme, exprimé dans le langage d'assemblage des processeurs **x86** :

```
main proc
jmp debut
mess db 'Hello world!$'
debut:
mov dx, offset mess
mov ah, 9
int 21h
ret
main endp
cseg ends
end main
```

C. Structure

Un programme est typiquement composé d'un ensemble de **procédures** et de **fonctions**. Une procédure est une suite d'instructions destinées à réaliser une opération ; par exemple, trier une liste. Une fonction est une suite d'instructions destinées à produire un résultat ; par exemple, un calcul.

D. Dysfonctionnements

Un **bug** est un défaut de construction dans un programme. Les instructions, que l'appareil informatique **exécute,** ne correspondent pas à ce qui est attendu, ce qui provoque des dysfonctionnements et des pannes. La pratique de la programmation informatique nécessite des outils pour traquer ou éviter les bugs, ou vérifier la correction du programme.

E. Exécution

L'exécution des programmes est basée sur le principe de la machine à programme enregistré, de **John Von Neumann** : les instructions de programme sont exécutées par un processeur. Ce composant électronique exécute chaque instruction de programme par une succession d'opérations charger/décoder/exécuter : l'instruction est tout d'abord copiée depuis la mémoire vers le processeur, puis elle est décomposée bit par bit pour déterminer l'opération à effectuer, qui est finalement exécutée. La plupart des opérations sont d'ordre arithmétique (addition, soustraction), ou logiques. L'exécution de programmes par le **processeur central** (anglais CPU) contrôle la totalité des opérations effectuées par l'ordinateur.

L'exécution du cycle charger-décoder-exécuter est rythmé par une horloge branchée au processeur.

En 2011, la fréquence d'horloge supportée par les processeurs contemporains se compte en mégahertz ou en gigahertz, ce qui correspond à des millions voire des milliards de cycles par seconde.

Les processeurs contemporains peuvent traiter plusieurs instructions simultanément : lorsqu'une instruction est chargée, le processeur charge immédiatement l'instruction suivante, sans attendre que cette instruction soit décodée puis exécutée, et les processeurs peuvent également charger/décoder/exécuter plusieurs instructions en un seul cycle d'horloge.

F. Processus

Pour être exécuté, un programme doit être chargé dans la mémoire de la machine. Le chargement d'un programme peut être soit automatique ou programmé lors de l'amorce de l'ordinateur par exemple, soit interactif et être déclenché par un ordre d'exécution explicite de l'utilisateur (une commande explicite, un appui sur une touche, un bouton, une icône…). Suivant la nature de l'action à effectuer, un programme peut être exécuté de manière ponctuelle (impression d'un texte), de manière répétitive (mise à jour de coordonnées GPS), ou de manière (presque) permanente (surveillance de capteurs).

Un programme est une suite d'instructions qui spécifie étape par étape, de manière non ambiguë, des représentations de données et des calculs. Les instructions sont destinées à manipuler les données lors de l'exécution du programme. Le programme lui-même est défini par un (ou des) algorithme(s) ou par une spécification. Un programme décrit de manière exacte les différentes étapes d'un algorithme : ce qu'il y a à faire, quand et avec quelles informations. Selon l'architecture de Von Neumann créée en 1945, un programme est chargé dans la mémoire de l'ordinateur, ce qui permet de l'exécuter de manière répétée sans intervention humaine, et surtout d'utiliser la même machine pour exécuter autant de programmes que l'on veut. La mémoire dédiée aux programmes est aussi la mémoire dédiée aux données, ce qui permet de traiter les programmes comme des données comme les autres, par exemple écrire de nouveaux programmes de la même manière qu'on écrirait un document textuel, puis par la suite les exécuter.

Des programmes peuvent être exécutés non seulement par les ordinateurs, mais par les nombreux appareils qui sont basés sur des composants informatiques, par exemple certains robots ménagers, téléphones, fax, instruments de mesure, récepteur de télévision, ainsi que les pièces de matériel informatique telles que disque durs, routeurs, imprimantes, les consoles de jeu vidéo, les assistants personnels et les automates bancaires. Contrairement aux ordinateurs, ces appareils ne contiennent souvent pas de système d'exploitation, les programmes sont enregistrés dans l'appareil lors de la fabrication, et la vitesse d'exécution des programmes est souvent d'importance mineure.

Sans contre-indication, les instructions d'un programme sont exécutées une après l'autre, de manière linéaire. Les langages de programmations permettent d'exprimer des alternatives : une suite d'instructions est exécutée uniquement si une condition donnée est remplie, dans le cas contraire une autre suite est exécutée. Les langages de programmation permettent également de faire répéter l'exécution d'une suite d'instructions jusqu'à ce qu'une condition donnée soit remplie.

Schéma général d'exécution de programmes

G. Compilation et interprétation

L'exécution se déroule de manière différente suivant si le langage de programmation s'utilise avec un **compilateur** ou un **interpréteur**.

- Un compilateur lit le programme source en entier, et le transforme en instructions machines. La transformation peut se faire en plusieurs étapes et nécessiter plusieurs lectures du programme. Une fois traduit, le programme est ensuite enregistré en vue d'être plus tard copié en mémoire et exécuté par le processeur tel quel;
- Un interpréteur opère ligne par ligne : lit une ligne de programme source, puis exécute immédiatement les instructions machines correspondantes. L'avantage d'un interpréteur est que les erreurs peuvent être immédiatement corrigées. Le désavantage est que l'exécution du programme est 10 à 100 fois moins rapide que si le programme avait été préalablement traduit et exécuté tel quel.

Critère	Compilation	Interprétation
Efficacité	Code natif de la machine Il peut être optimisé	10 à 100 fois plus lent Appel de sous-programmes Pas de gain sur les boucles
Mise au point	Lien erreur ↔ source complexe	Lien instruction ↔ exécution trivial Trace et observations simples
Cycle de développement	Cycle complet à chaque modification : compilation, édition de liens, exécution	Cycle très court : modifier et ré-exécuter

H. Environnement d'exécution

Les ordinateurs modernes démarrent à leur lancement un programme « maître » dit système d'exploitation. Il permet d'exécuter des sous-programmes, qui peuvent alors profiter des fonctionnalités offertes par le système, voire doivent s'adapter à cet environnement. Les systèmes d'exploitation contemporains permettent d'exécuter simultanément plusieurs programmes dans des processus, même avec un seul processeur : un programme planificateur (en anglais : scheduler) du système d'exploitation interrompt régulièrement le programme en cours d'exécution pour donner la main à un autre. La vitesse de rotation donne l'illusion que les programmes sont exécutés en même temps.

Un sous-programme du système d'exploitation peut lui-même être un environnement permettant d'exécuter des programmes (avec une interface différente) ; par exemple, une machine virtuelle.

I. Aspects juridiques

En droit, un programme est une œuvre écrite, protégée par le droit d'auteur. Celui-ci s'applique au programme du moment qu'il est enregistré de manière permanente, même s'il n'existe pas d'édition sur papier. Le droit d'auteur protège autant le programme source, que le programme binaire.

J. LEXIQUE

Algorithme

Un algorithme est une suite finie et non ambiguë d'opérations ou d'instructions permettant de résoudre un problème ou d'obtenir un résultat.

Le mot algorithme vient du nom d'un mathématicien perse du IXème siècle, Al-Khwârizmî (en arabe : الخوارزمي). Le domaine qui étudie les algorithmes est appelé l'algorithmique. On retrouve aujourd'hui des algorithmes dans de nombreuses applications telles que le fonctionnement des ordinateurs, la cryptographie, le routage d'informations, la planification et l'utilisation optimale des ressources, le traitement d'images, le traitement de texte, la bio-informatique, etc.

Langage machine

Le langage machine, ou code machine, est la suite de bits qui est interprétée par le processeur d'un ordinateur exécutant un programme informatique. C'est le langage natif d'un processeur, c'est-à-dire le seul qu'il puisse traiter. Il est composé d'instructions et de données à traiter codées en binaire.

Chaque processeur possède son propre langage machine, dont un code machine qui ne peut s'exécuter que sur la machine pour laquelle il a été préparé. Si un processeur A est capable d'exécuter toutes les instructions du processeur B, on dit que A est compatible avec B. L'inverse n'est pas forcément vrai : A peut avoir des instructions supplémentaires que B ne connaît pas.

Le code machine est aujourd'hui généré automatiquement, généralement par le compilateur d'un langage de programmation ou par l'intermédiaire d'un bytecode.