

# Les logiciels

## 1. Définition

Un programme informatique est une liste d'ordres indiquant à un ordinateur ce qu'il doit faire. Il se présente sous la forme d'une ou plusieurs séquences d'instructions, comportant souvent des données de base, devant être exécutées dans un certain ordre par un processeur ou par processus informatique (cas des systèmes multitâches).

## 2. Fonction

Un ordinateur sans programme ne fait absolument rien, il attend des instructions. En fait, la capacité à suivre un programme enregistré sert même souvent, d'un point de vue historique, à distinguer un ordinateur d'une simple machine à calculer.

Ecrire un programme, c'est donc écrire une suite d'instructions élémentaires s'enchaînant les unes après les autres pour réaliser un traitement sur des données. Dans le disque dur et la mémoire centrale, ces programmes sont codés sous forme de bits. De tels programmes sont aussi appelés des «programmes exécutables», puisqu'ils sont directement prêts à être exécutés.

Au début de l'informatique, dans les années 50, programmer revenait ainsi à écrire de telles suites d'instructions élémentaires. Les langages de programmation les plus proches de ce codage de «bas niveau» sont appelés «langages d'assemblage» ou «Assembleurs». Programmer dans un tel langage nécessite de connaître l'architecture matérielle de l'ordinateur sur lequel il s'exécutera ; il y a ainsi presque autant de langages d'assemblage différents que de microprocesseurs.

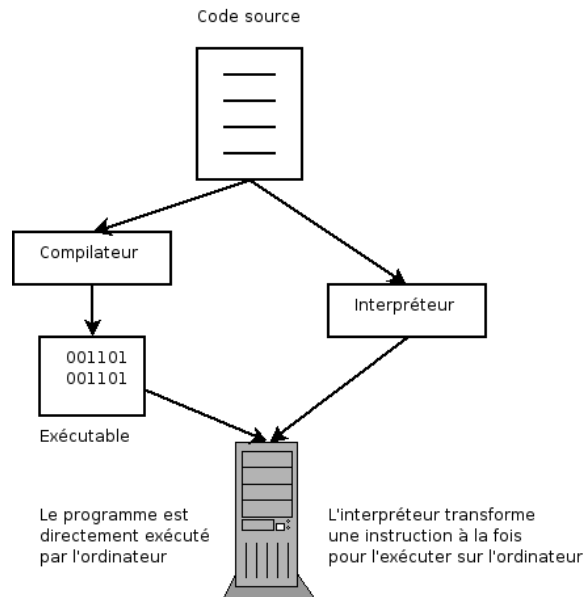
Les programmes écrits en Assembleur se traduisent immédiatement en programmes exécutables. Mais programmer en Assembleur n'est ni très simple ni très agréable. Aussi, de nouveaux langages de programmation ont peu à peu été définis pour faciliter la tâche des programmeurs.

## 3. Conception

À l'origine d'un programme, il y a un code source écrit par un programmeur dans un langage de programmation compréhensible par le dit programmeur.

Selon le langage utilisé, ce code est ensuite soit :

- Traduit avec un jeu d'instructions spécifique à un processeur par un compilateur, ensuite le programme obtenu peut alors être exécuté directement par l'ordinateur.
- Ou bien est pris en charge par un interpréteur (autre programme), qui décode à la volée les instructions du langage évolué en instructions spécifiques au processeur et qui les lui transmet directement pour exécution.



Le terme « programme informatique » est souvent improprement, utilisé comme synonyme de logiciel, les logiciels actuels étant souvent composés de plusieurs programmes. Les logiciels incluent souvent en plus, des fichiers de ressources contenant des données de toutes sortes, celles-ci ne font pas à proprement parlé partie du programme. Par exemple, Microsoft Internet Explorer, Mozilla Firefox, etc. sont des logiciels plutôt que des programmes, car ils sont le fruit de la concaténation de multiples programmes exécutant différentes actions et fonctions.

Un programme simple et souvent abstrait est souvent appelé algorithme. Les programmes d'ordinateur sont aujourd'hui souvent les sujets de la logique et des mathématiques : voir les méthodes formelles, la sémantique des langages de programmation, etc.

#### 4. Logiciel

Un logiciel ou application est un ensemble de programmes, qui permet à un ordinateur ou à un système informatique d'assurer une tâche ou une fonction en particulier (exemple : logiciel de gestion de la relation client, logiciel de production, logiciel de comptabilité, logiciel de gestion des prêts).

On distingue en général, dans un système informatique, la partie matérielle (l'ordinateur et ses périphériques) et la partie logicielle, immatérielle (les programmes « écrits » sur le disque dur).

Le terme logiciel est souvent employé pour désigner un programme informatique, et inversement, bien qu'un logiciel puisse être composé d'un seul ou d'une suite de programmes.

Ce dernier cas est d'autant plus fréquent que la capacité réduite de calcul de l'ordinateur oblige à une segmentation des tâches en plusieurs modules séparés ; cependant, les énormes capacités des micro-ordinateurs actuels en regard des applications typiques de la bureautique ont permis la réalisation d'applications monolithiques.

Généralement, les programmes sont accompagnés d'un ensemble de données permettant de les faire fonctionner (par exemple, un jeu viendra avec de nombreuses images, animations, sons, etc.).

Pour fonctionner, un logiciel nécessite l'utilisation d'un ordinateur (micro-ordinateur, station de calcul, mainframe, supercalculateur, etc.) sur lequel existe à l'origine un « logiciel-moteur » (système d'exploitation) qui accepte le « logiciel-application ».

#### **4.1. Les licences**

Le droit d'utilisation du logiciel est généralement règlementé par une licence d'utilisation et le droit d'auteur.

Les grandes familles de licences les plus connues sont :

- Les licences autour du logiciel libre (free software en anglais ex: Projet GNU) ;
- Le gratuiciel (freeware ex: icq.com) ;
- Le partagiiciel (shareware ex: 6def.com) ;
- D'autres types de licences, telles que les licences monoposte ou multiposte ;
- Certains logiciels enfin sont internes à des entreprises et leur diffusion est interdite.

#### **4.2. Diverses présentations de logiciels**

Les programmes peuvent être de différentes formes :

- Exécutables : ils peuvent être exécutés directement par l'ordinateur (généralement, ils ne peuvent être exécutés que sur un type de machine et de système d'exploitation particulier (exemple : Microsoft Windows sur un compatible PC) ;

Cependant, il existe des exécutables (en bytecode) exécutables sur une variété de plates-formes (comme ceux du langage Java) ; ils visent en fait l'exécution pour une machine virtuelle, qui est elle-même un logiciel disponible sur les diverses plates-formes.

- Fichiers sources : il s'agit généralement d'un texte respectant les règles d'écriture d'un langage de programmation particulier ; à titre indicatif, l'ordre de grandeur de la taille d'un logiciel comme Microsoft Word est d'un million de lignes de code ;
- Bibliothèques : il s'agit de programmes exécutables ou source qui, en eux-mêmes, ne sont pas exécutables directement et n'offrent pas de fonctionnalité à l'utilisateur, mais fournissent des services à d'autres programmes (par exemple, on trouvera des bibliothèques permettant à un programme de charger des animations ou de jouer des sons) ; on trouve en particulier des bibliothèques dynamiques (dll Windows ou so GNU/Linux).

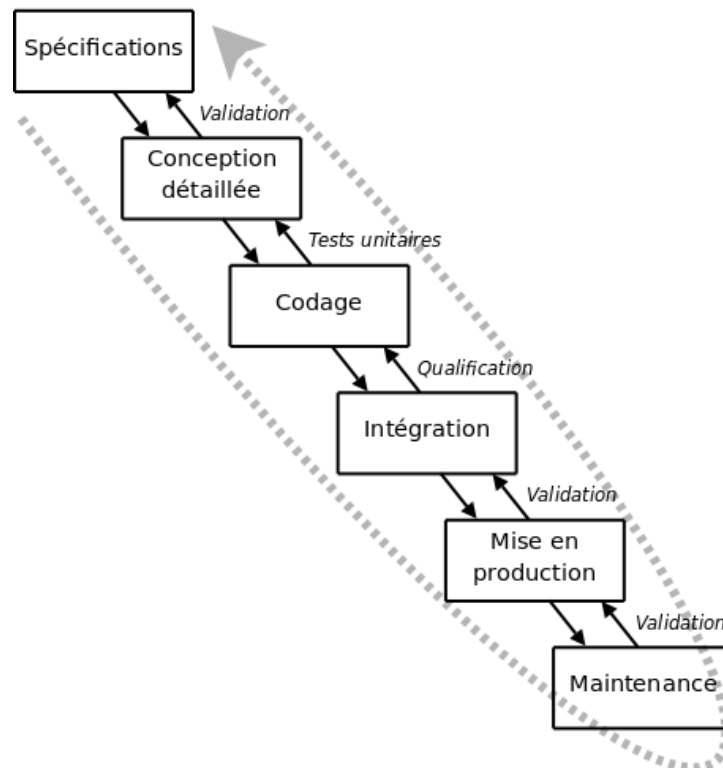
Les données associées au logiciel peuvent également être de différents formats : fichiers classiques, bases de données (relationnelles, hiérarchiques, etc.). Les données du logiciel peuvent être éclatées en un grand nombre de fichiers, ou tout le logiciel peut être rassemblé en un seul fichier ; par exemple, sous Windows, la

définition de l'interface utilisateur, le dessin des icônes etc., sont souvent intégrés dans le même fichier que l'application principale.

### 4.3. Développement de logiciels

Les logiciels, suivant leur taille, peuvent être développés par une personne seule, une petite équipe, ou un ensemble d'équipes coordonnées. Le développement de grands logiciels par de grandes équipes pose de grands problèmes de coordination, en raison de la quantité importante d'informations à communiquer entre les intervenants : documentation, réunions. Pour ces raisons, le développement de logiciels dans un contexte professionnel suit souvent des règles strictes permettant le travail en groupe et la maintenance du code ; en effet, souvent, les personnes qui doivent opérer des modifications ultérieures dans le code ne sont plus les personnes qui l'ont développé.

Le développement d'un logiciel correspond alors au cycle de vie décrit par la figure suivante :



Un logiciel en version bêta (ou bêta-test) est un logiciel non finalisé, pour lequel on effectue une série de tests jusqu'à ce qu'une stabilité relative soit atteinte. Les personnes qui cherchent les dernières failles de ces versions de logiciels sont appelées des bêta-testeurs.

### 4.4. Bogues

Des erreurs de conception dans les logiciels peuvent causer des comportements incorrects, souvent appelés bogues. La gravité de ceux-ci peut aller de très mineure (p.ex., apparence légèrement incorrecte d'un élément d'interface graphique), à des événements catastrophiques (explosion de la fusée Ariane vol 501, irradiation incorrecte de patients par une machine de traitement...) en passant par des pertes plus ou moins grandes de données, et, rarement, par une détérioration du matériel. Il est difficile, pour des raisons fondamentales, de produire des logiciels sans bogue, cependant, il existe des méthodologies, d'autre part les technologies de recherche de bogues dans les logiciels. La recherche en informatique a développé un domaine d'étude, la vérification formelle, dont l'objectif est de certifier la qualité des logiciels et de garantir leur fiabilité. Plus les anomalies sont détectées tôt au long du développement du logiciel, moins leur correction est coûteuse.

#### **4.5. Ouverture du code source**

On classe les logiciels d'après la disponibilité du code source et de la licence qui régit la distribution du programme :

- code ouvert : tout le monde peut lire le code source. Ce terme n'est pas synonyme de logiciel libre ;
- code fermé : le code source n'est disponible que pour une minorité de personnes ;
- Logiciel libre : tout le monde peut étudier, copier, modifier et distribuer des versions modifiées du logiciel (définition de la free software foundation). Les logiciels libres sont protégés pour la plupart par une licence d'utilisation ; Pour autant cela ne signifie pas que le logiciel est gratuit.
- Logiciel propriétaire : au moins un de ces droits n'est pas rempli pour les utilisateurs. La plupart du temps, acquérir une licence d'utilisation nécessite le paiement d'une certaine somme aux créateurs du logiciel ;
- Logiciel commercial : logiciel destiné à la vente, il peut être libre ou propriétaire.