Langages de programmation [pr]
Support de Cours

Socle – Université de Haute Alsace

Unisciel algoprog Version 31 mai 2018

Table des matières

Langages de programmation (Cours)

Mots-Clés Langages de programmation ■

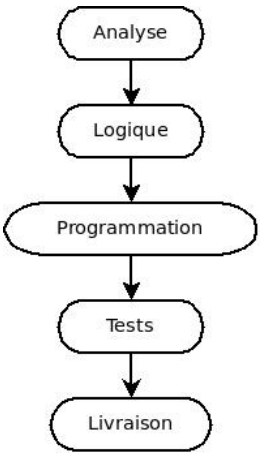
Introduction

Ce module décrit la petite histoire des langages informatiques, présente la typologie des langages de programmation puis précise les phases d'élaboration d'un programme. En *Compléments*, il expose brièvement les constituants principaux de l'ordinateur et l'exécution d'un programme.

1 Phases d'élaboration d'un programme

Schéma simplifié

Voici un schéma **simplifié** des phases par lesquelles il faut passer quand on développe un programme :



Analyse

Cette étape consiste à comprendre et à préciser clairement par écrit « ce qu'il faut faire ». Elle demande une bonne culture générale des méthodes d'analyse et nécessite de réelles capacités de dialogue.

Logique = Conception d'une solution

Une fois le problème analysé, il faut réfléchir à l'**algorithme** qui va permettre de résoudre le problème, en essayant de minimiser le temps de calcul et/ou l'espace mémoire occupé, sans se soucier des contraintes techniques de l'écriture du programme. C'est le but d'un cours d'algorithmique.

Programmation = Codage

Cette étape est quasi-automatique puisqu'elle consiste à traduire le ou les algorithmes de l'étape précédente dans le langage de programmation choisi. Vos cours de langage (C/C++, Java, Python...) sont dédiés à cette phase.

Tests = Vérification et Validation

La **validation** est un processus expérimental qui consiste à définir des jeux de données pertinents, élaborés à partir des spécifications et à les tester avec le programme. La **vérification** est effectuée sur la solution. Elle peut théoriquement prendre deux formes : la preuve ou un processus expérimental.

Cette phase de **tests** ne manquera pas de montrer qu'il subsiste des problèmes qu'il faut encore corriger. (Vous aurez maintes fois l'occasion de vous en rendre compte lors des séances de travaux pratiques.)

Livraison

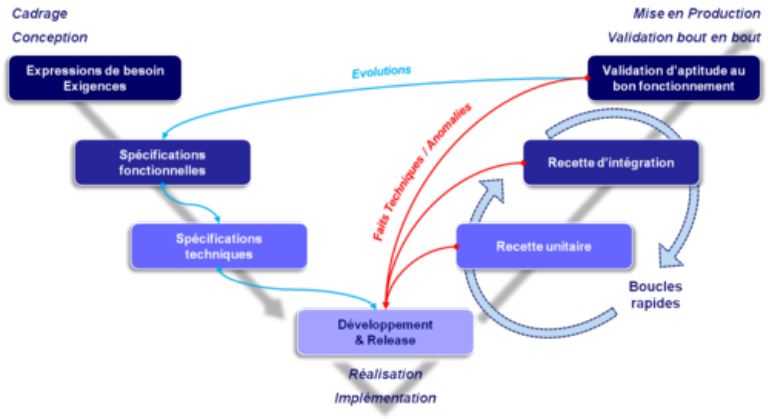
Le produit sans bug (connu) peut être **mis en application** ou **livré** à la personne qui vous en a passé la commande.

Processus non linéaire

À chaque phase, on pourra détecter des erreurs, imprécisions ou oublis des phases précédentes et revenir en arrière.

Autre représentation

Il existe plusieurs représentations, une des plus connues étant ce que l'on appelle le *cycle en V* (google-images) :



Le pourquoi de la phase Logique

Pourquoi ne pas directement passer à la programmation ?

Voilà une question que vous ne manquerez pas de vous poser. Apportons quelques éléments de réflexion.

1. Passer par une phase de « logique » permet de séparer deux difficultés : quelle est la marche à suivre ? Et comment l'exprimer dans le langage de programmation choisi ? Le langage algorithmique est plus souple et plus général qu'un langage de programmation (où il faut être précis au « ; » près).
2. Un algorithme est plus adapté à la communication car plus lisible : il facilite donc le dialogue dans une équipe de développement.
3. Un algorithme pourra facilement être traduit dans n'importe quel langage de programmation. La traduction d'un langage de programmation à un autre est un peu moins facile à cause des particularités propres à chaque langage.
4. L'ordinateur n'est que le troisième outil pour le développeur. Le premier est la gomme, suivi du crayon parce qu'il s'efface (on fait 7% d'erreurs dans tout ce que l'on fait).

Conclusion

L'algorithme est donc primordial car ce sont les solutions qu'il propose qui seront programmées. La qualité et la rapidité du programme final découlent directement de la phase *Logique*.

Bien sûr, cela n'a de sens que si le problème présente une réelle difficulté logique. Certains problèmes (en pratique, certaines parties de problèmes) sont suffisamment simples pour être directement programmés. Mais qu'est-ce qu'un problème simple ? Cela va évidemment changer tout au long de votre apprentissage. Un problème qui vous paraîtra difficile en début d'année vous paraîtra (enfin, il faut l'espérer !) une évidence en fin d'année.

2 Compléments

2.1 Constituants principaux de l'ordinateur

Le matériel (hardware)

Il est constitué de l'ordinateur proprement dit et regroupe :

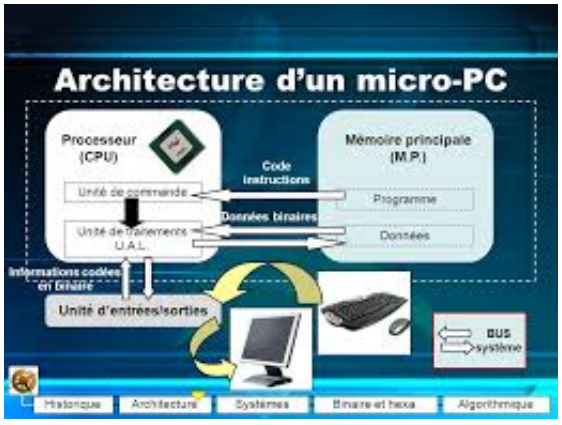
L'organe de contrôle : Cerveau de l'ordinateur. Il est l'organisateur, le contrôleur suprême de l'ensemble. Il assume l'enchainement des opérations élémentaires. Il s'occupe également d'organiser l'exécution effective de ces opérations élémentaires reprises dans les programmes.

L'organe de calcul : Calculateur où ont lieu les opérations arithmétiques ou logiques. Avec l'organe de contrôle, il constitue le **processeur** ou **unité centrale**.

La mémoire centrale : Dispositif permettant de mémoriser, pendant le temps nécessaire à l'exécution, les programmes et certaines données pour ces programmes.

Les unités d'échange avec l'extérieur : Dispositifs permettant à l'ordinateur de recevoir des informations de l'extérieur (unités de lecture telles que clavier, souris, écran tactile...) ou de communiquer des informations vers l'extérieur (unités d'écriture telles que écran, imprimantes, signaux sonores...).

Les unités de conservation à long terme : Mémoires auxiliaires (disques durs, CD ou DVD de données, clés USB...) sur lesquelles sont conservées les procédures (programmes) ou les informations résidentes dont le volume ou la fréquence d'utilisation ne justifient pas la conservation permanente en mémoire centrale.

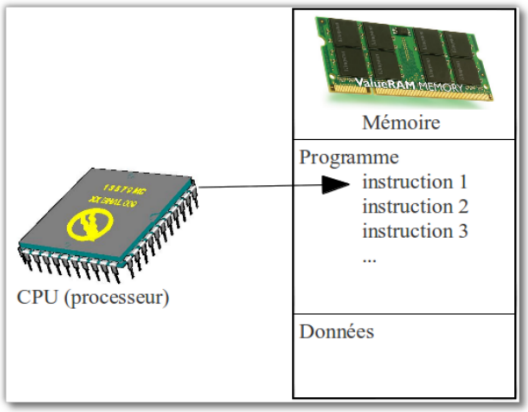


Le logiciel (software) d'exploitation

C'est l'ensemble des procédures (programmes) s'occupant de la gestion du fonctionnement d'un système informatique et de la gestion de l'ensemble des ressources de ce système (le matériel — les programmes — les données). Il contient notamment des logiciels de traduction permettant d'obtenir un programme écrit en langage machine (langage technique qui est le seul que l'ordinateur peut comprendre directement, c.-à-d. exécuter) à partir d'un programme écrit en langage de programmation plus ou moins « évolué » (c.-à-d. plus ou moins proche du langage naturel).

2.2 Exécution d'un programme

Isolons (en les simplifiant) deux constituants essentiels de l'ordinateur afin de comprendre ce qui se passe quand un ordinateur exécute un programme. D'une part, la mémoire contient le programme et les données manipulées par ce programme. D'autre part, le processeur va « exécuter » ce programme.



De façon très simplifiée, on passe par les étapes suivantes :

1. Le processeur lit l'instruction courante.
2. Il exécute cette instruction. Cela peut amener à manipuler les données.
3. L'instruction suivante devient l'instruction courante.
4. On revient au point 1.

On voit qu'il s'agit d'un travail automatique ne laissant aucune place à l'initiative.

3 Références générales

Programmation

Chamoret-PG1 @BookChamoret-PG1, author = Chamoret Thierry, title = Le langage PASCAL ISO, publisher = Ellipses, year = , isbn = 2-7298-1140-0, type = , edition = , school = Licence L1, note =

Chaty-PG1 @BookChaty-PG1, author = Chaty, Guy AND Vicard, Jean, title = L'algorithmique, publisher = Nathan, ISBN = 2-09191-722-2, year = 1989, type = , edition = , school = , note =

Felea-PG1 @BookFelea-PG1, author = Felea, Violeta AND Felea, Victor AND Steen, Jean-Pierre, title = Apprendre à programmer avec C et Python, publisher = Vuibert, ISBN = 978-2-311-40199-8, year = 2015, type = , edition = , school = DUT et Licence 1 informatique, BTS services informatiques aux organisations, note = Cours, complet. Algorithmiques, codages, tests. Problèmes intégralement résolus. Exercices d'approfondissement,

Rohaut-PG1 @BookRohaut-PG1, author = Rohaut, Sébastien AND Ebel, Franck, title = Algorithmique. Techniques fondamentales de programmation, Exemples en Python, publisher = eni-editions, year = 2014, isbn = 978-2-7460-8919-8, type = , edition = , school = BTS, DUT Informatique, note =

Tarlowski-PG1 @BookTarlowski-PG1, author = Tarlowski Alain, title = Algorithmie et Programmation, publisher = Bertrand-Lacoste, year = 2001, 1e Edition, isbn = 2-7352-1737-X, type = , edition = , school = , note =

Warin-PG1 @BookWarin-PG1, author = Warin, Bruno, title = L'algorithmique - passeport informatique pour la programmation, publisher = Ellipses, year = 2002, 1è Edition, isbn = 2-7298-1140-0, type = , edition = , school = , note =

Exercices

Maysonnave-CC1 @BookMaysonnave-AL1, author = Maysonnave, title = Introduction à l'algorithmique générale et numérique, publisher = Masson, ISBN = 2-225-85375-4, year = 1996, type = , edition = , school = , note =

Tartier-AL1 @BookTartier-AL1, author = Tartier, Annie AND Vailly, Alain, title = Premiers pas en algorithmique, publisher = ellipses, ISBN = 9782340-001251, year = 2014, type = Technosup, edition = , school = , note = De l'enoncé à la solution. Exercices analysés, corrigés et comentés