INTRODUCTION GÉNÉRALE

B. Landelle

Table des matières

I Présentation **2**
1 Premières définitions 2
2 L'informatique théorique 3
3 L'ordinateur théorique 3

II Architecture matérielle **3**
1 Modèle générique d'architecture matérielle 3
2 L'unité de contrôle et l'unité arithmétique et logique 4
3 La mémoire 4
4 Les entrées-sorties 5

III Organisation matérielle **5**
1 La carte mère 5
2 Le microprocesseur 6
3 La mémoire 6
4 Le disque dur 7

IV Organisation logicielle **8**
1 Définition 8
2 Système d'exploitation 8

V La programmation **9**
1 Le traitement de l'information 9
2 Langages de programmation 10
3 Python, Idle 12

I Présentation

1 Premières définitions

Le terme *informatique* est un néologisme proposé en 1962 comme contraction de l'expression « information automatique ».

Définition 1. *L'*informatique *est la science du traitement automatique de l'information.*

Remarque : Il n'y a pas d'équivalent exact aux États-Unis où l'on parle de *computing science*, science du calcul.

L'informatique traite de deux aspects complémentaires : les programmes immatériels qui décrivent un traitement à réaliser et les machines qui exécutent ce traitement.

Définition 2. *Le* matériel informatique *est un ensemble de dispositifs utilisés pour traiter automatiquement des informations.*

L'informatique commence véritablement dans les années 1930 avec la théorisation par le mathématicien Alan Turing de ce qu'est un ordinateur. Le transistor, composant clé de l'informatique, est inventé en 1947 par des chercheurs de la compagnie *Bell Telephone*. Il remplace les tubes électroniques utilisés jusqu'alors. Son invention est incontestablement un pas décisif. Il faut ensuite attendre 1958 pour voir apparaître le premier circuit imprimé réalisé par la société *Texas Instrument*. En 1971, la société *Intel* réussit à placer tous les transistors constituant un processeur sur un seul circuit intégré, donnant naissance au premier microprocesseur. Depuis, les progrès technologiques considérables ont permis l'explosion de l'informatique et son utilisation dans une multitude de domaines.

Parmi les composantes clés de l'informatique, on distingue :
- les données numériques,
- les algorithmes,
- les machines,
- les langages.

Définition 3. *Un* algorithme *est une suite finie d'instructions permettant la résolution d'un problème.*

Les algorithmes pré-existent à l'informatique moderne. Euclide décrit ainsi en -300 av. J.C. un algorithme permettant le calcul du pgcd de deux entiers dont on ne connaît pas la factorisation. L'intérêt pratique de ces algorithmes s'est vu démultiplié avec l'émergence des ordinateurs et l'accroissement de leur puissance de calcul.

Définition 4. *Un* langage de programmation *est un langage conventionnel servant à communiquer des instructions à une machine, en particulier un ordinateur. Un* programme *est un ensemble d'instructions écrit dans un langage de programmation donné.*

Comme pour une langue parlée, un langage de programmation s'appuie sur un alphabet et est régi par des règles syntaxiques et sémantiques.

2 L'informatique théorique

L'informatique au sens commun désigne l'usage de ressources informatiques matérielles, logicielles, réseaux, ... Il s'agit de la perception habituelle de la part d'un utilisateur. Un aspect méconnu de l'informatique est celui de la science informatique en tant que science formelle au même titre que les mathématiques. Présentons quelques aspects de cette discipline :

— la *calculabilité* : une fonction f est calculable si, étant donné un argument x, on peut obtenir l'image $f(x)$ en un nombre fini d'étapes. On peut démontrer qu'il existe des fonctions non calculables ;

— *l'algorithmique* : ensemble des règles et techniques utilisées dans la définition et la conception d'algorithmes ;

— la *cryptologie* : discipline qui couvre la cryptographie (écriture secrète par utilisée par exemple lors de transactions sécurisées en mode `https://`) et la cryptanalyse (analyse de la cryptologie).

3 L'ordinateur théorique

La *machine de Turing* est un modèle abstrait d'ordinateur imaginé par le mathématicien et informaticien britannique Alan Turing (1912-1954). Cette « machine » est composée d'un ruban infini, d'une tête de lecture-écriture et d'une fonction de transition.

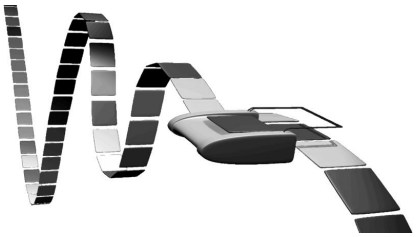


FIGURE 1 – Vue d'artiste d'une machine de Turing

Le ruban est divisé en cases consécutives et chacune contient un symbole parmi un alphabet fini. La tête de lecture-écriture peut lire et écrire les symboles sur le ruban et se déplacer le long du ruban. La fonction de transition indique à la machine quel symbole écrire et quel déplacement faire opérer à la tête de lecture.

Le modèle de Turing est toujours utilisé en informatique théorique pour aborder les questions de complexité algorithmique et de calculabilité.

II Architecture matérielle

1 Modèle générique d'architecture matérielle

L'architecture matérielle d'un ordinateur repose sur le modèle de Von Neuman (mathématicien et physicien hongrois, 1903-1957) et distingue quatre parties :

1. l'*unité de traitement* (UAL : unité arithmétique et logique) qui effectue les opérations de base ;
2. l'*unité de contrôle* qui séquence les opérations ;
3. la *mémoire* qui contient à la fois les données et les instructions ;

4. les *entrées-sorties* qui permettent la communication vers le monde extérieur.

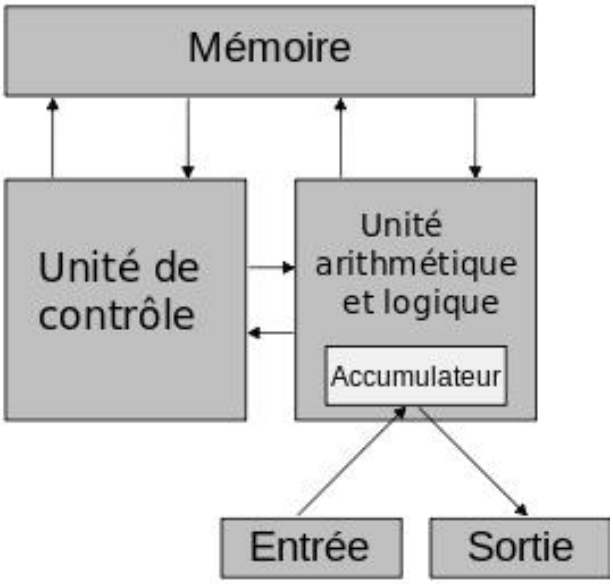


FIGURE 2 – Architecture de Von Neuman

Les différents organes du système sont reliés par des voies de communication appelées *bus*.

2 L'unité de contrôle et l'unité arithmétique et logique

Ces unités sont rassemblées au sein d'une même entité : le *microprocesseur*.

L'unité de contrôle détermine la succession des instructions dans le temps. L'unité de traitement exécute les instructions transmises par l'unité de commande. Le microprocesseur possède en son sein des emplacements de mémoire appelés registres dont il se sert pour réaliser les calculs (processeur 32-bits ou 64-bits, le nombre désigne la taille des registres). Dans l'architecte précédemment décrite, les registres sont considérées comme partie intégrante des unités de traitement et de contrôle.

3 La mémoire

Historiquement, on distingue deux types de mémoire au sein de l'ordinateur :

— la *mémoire morte* ROM (Read Only Memory) qui est une mémoire permanente ou rémanente ou encore non-volatile qu'on ne peut que lire (elle contient par exemple les instructions nécessaires au démarrage de l'ordinateur) ;

— la *mémoire vive* RAM (Random Access Memory) utilisée pour traiter les processus en cours dans laquelle on peut lire et écrire et dont le contenu est perdu hors alimentation (mémoire dite volatile ou non-rémanente).

L'appellation de mémoire morte désigne aujourd'hui une réalité plus large : l'usage de mémoires ROM « reprogrammables » (donc non mortes au sens originel du mot) s'est généralisée avec les EEPROM (mémoires « mortes » effaçables et reprogrammables) et en particulier les mémoires flash, variété de mémoire EEPROM rapide.

Une des contraintes importantes à la performance d'un ordinateur est le délai de communication entre ces différentes entités notamment entre le microprocesseur et la mémoire RAM qui est la mémoire de travail. Pour pallier à cette contrainte, on intercale des mémoires dites *caches* plus rapides et qui conservent les données les plus fréquemment utilisées.

On distingue également la *mémoire de masse* qui désigne généralement une mémoire de masse non volatile de grande capacité comme un disque dur interne ou externe, CD, DVD, carte mémoire, clé USB.

4 Les entrées-sorties

Ces interfaces assurent la communication entre le microprocesseur et les périphériques (écran, clavier, souris, imprimante, ...)

III Organisation matérielle

Au cœur d'un PC, on trouve :
- la carte mère ;
- le microprocesseur ;
- la mémoire ;
- le disque dur ;
- etc. ...

1 La carte mère

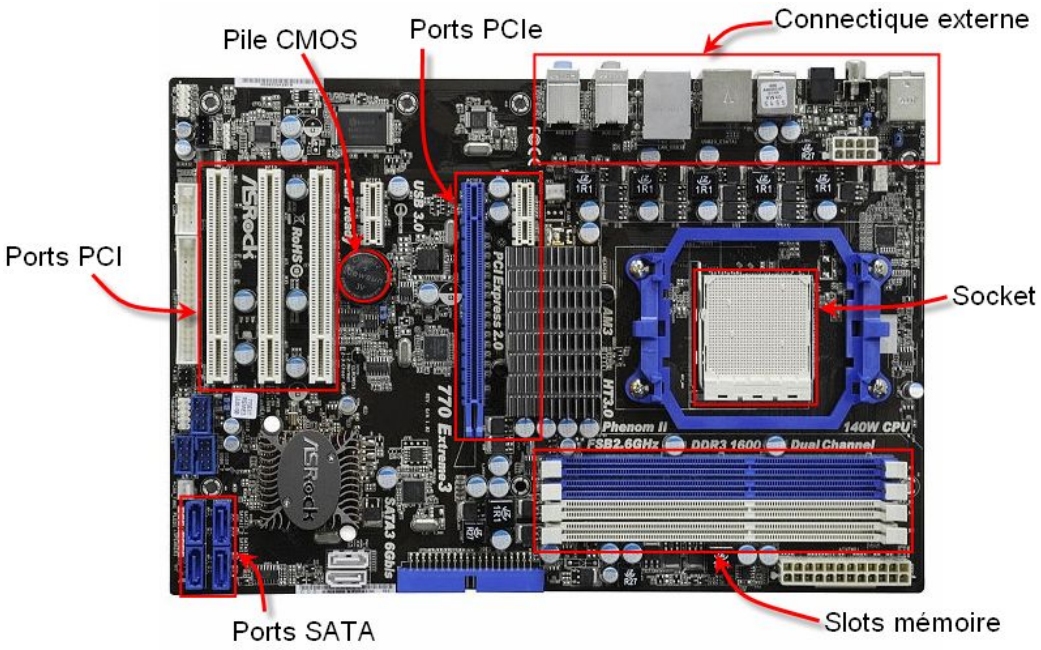


FIGURE 3 – Carte mère

L'élément constitutif principal de l'ordinateur est la carte mère (*motherboard* en anglais). Elle assure la connexion physique des différents composants (processeur, mémoire, ...) par l'intermédiaire de différents bus (adresses, données et commandes). Sur la carte mère, on trouve :

— le chipset : circuit qui gère les transferts de données entre les différents composants de l'ordinateur ; il est souvent séparé en deux puces : NorthBridge et SouthBridge ;
— l'UEFI (anciennement le BIOS) : c'est l'interface entre un micrologiciel (firmware) stocké sur une mémoire flash sur la carte mère et le système d'exploitation ; il constitue en lui-même un mini-système d'exploitation ;
— l'horloge : elle permet de cadencer le traitement des instructions par le microprocesseur ou la transmissions des informations sur les différents bus ;
— la mémoire interne (CMOS) qui conserve même lorsque l'ordinateur est éteint la configuration, la date et l'heure ;
— des bus : bus système (FSB Front Side Bus) reliant le chipset au microprocesseur, bus mémoire reliant le chipset à la mémoire vive, bus d'extension reliant le microprocesseur aux connecteurs ;
— des connecteurs internes : le socket, connecteur destiné au microprocesseur, des connecteurs de mémoire vive (memory slot), des connecteurs prévus pour des cartes d'extension (PCI, ...) ou pour des périphérique de stockage (SATA, IDE, ...) ;
— des connecteurs d'entrée/sortie : port USB, connecteur RJ45, ...

2 Le microprocesseur

Le microprocesseur est le cerveau de l'ordinateur. Il s'agit d'un circuit intégré complexe résultant de l'intégration sur une même puce de fonctions logiques combinatoires (logiques et/ou arithmétiques) et séquentielles (registres, compteur, etc. ...). Il est capable d'interpréter et d'exécuter les instructions d'un programme. Les modèles actuels contiennent plus d'un milliard de transistors.

FIGURE 4 – Microprocesseur

3 La mémoire

La mémoire RAM se présente sous forme de petites cartes.

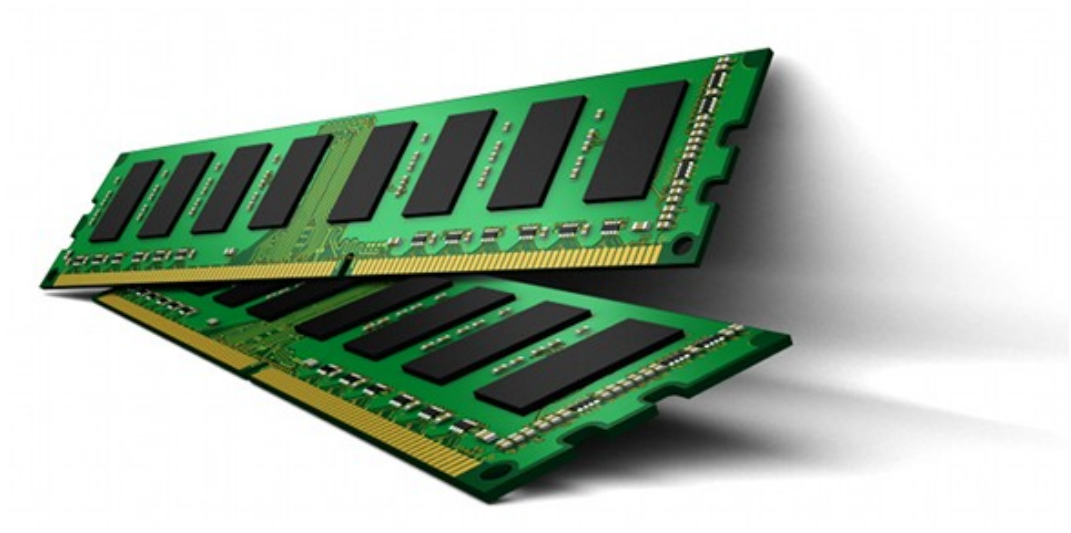


FIGURE 5 – Barrettes mémoire

Il s'agit de mémoire vive volatile. Chaque information stockée peut à chaque instant être consultée ou modifiée. Les informations sont perdues lors de la mise hors tension de l'appareil. La quantité et la qualité de la mémoire influent sur les performances de l'ordinateur. Le microprocesseur doit pouvoir accéder et sauvegarder rapidement dans cette mémoire. De même la capacité de stockage doit être dimensionnée pour limiter l'accès aux périphériques de stockage de masse qui sont plus lents (typiquement lors d'utilisation de mémoire virtuelle).

4 Le disque dur

Le disque dur est une mémoire de masse ; la technologie des disques magnétiques HDD (Hard Disk Drive) a longtemps dominé mais elle est désormais supplanté par le le SSD (Solid State Drive) constitué de mémoire flash, beaucoup plus rapide et moins fragile.

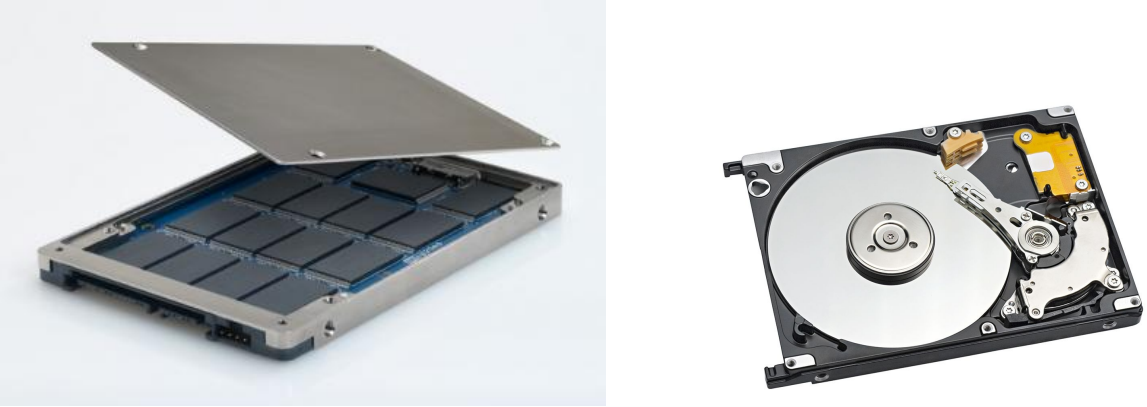


FIGURE 6 – Disques SSD et HDD

IV Organisation logicielle

1 Définition

Définition 5. *Un* logiciel *est un ensemble d'informations relatives à des traitements automatisés par un appareil informatique. Il est composé d'instructions et de données. Les instructions mettent en application les algorithmes en rapport avec le traitement de l'information voulu.*

On distingue trois catégories de logiciels :

- le logiciel applicatif : il est dédié à une certaine activité automatisée (traitement de texte, tableur, etc. ...)
- le logiciel système ou *système d'exploitation* est le premier programme exécuté après l'amorçage ; il contient l'ensemble des instructions et informations relatives à l'utilisation commune du matériel informatique par des logiciels applicatifs ;
- le micrologiciel (*firmware* en anglais) : il contient les instructions et informations d'une opération sur un équipement précis comme par exemple le firmware UEFI.

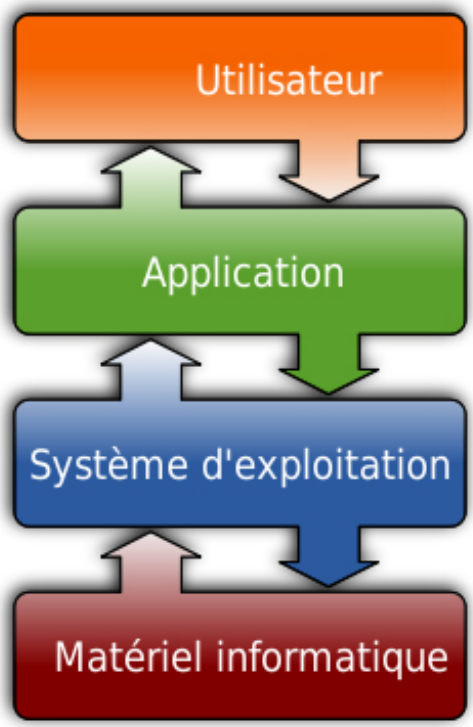


FIGURE 7 – Hiérarchie logicielle

2 Système d'exploitation

Le système d'exploitation (OS pour *operating system* en anglais) est la composante logicielle de base d'un ordinateur qui sert d'interface entre le matériel et les logiciels applicatifs.

Ainsi, le système d'exploitation offre une large palette de services aux applications : il permet, comme son nom l'indique, l'exploitation des périphériques dont il coordonne l'utilisation par une allocation précise des ressources. Il met à disposition des applications une interface de programmation standardisée d'utilisation des matériels. Il assure aussi la fiabilité (tolérance aux pannes, isolation partielle des fautes de mémoire) et la sécurité informatique de base.

Le système d'exploitation coordonne encore l'utilisation du ou des processeur(s) et accorde un temps défini pour l'exécution de chaque tâche.

Au niveau du stockage, il réserve de l'espace dans les mémoires pour les besoins des programmes.

Enfin, par le formatage, il organise le contenu des disques durs en système de fichiers, permettant ainsi leur utilisation.

Le système d'exploitation crée aussi un espace qui contiendra les interfaces des différents programmes. Il agit en chef d'orchestre, et coordonne l'interaction entre les applications et l'utilisateur, en traitant les manipulations effectuées via le clavier, la souris ou d'autres périphériques d'entrée.

Une fois le travail achevé, il transmettra les ordres d'enregistrement, d'effacement, de copie et d'impression vers les périphériques, ainsi que d'extinction de l'ordinateur.

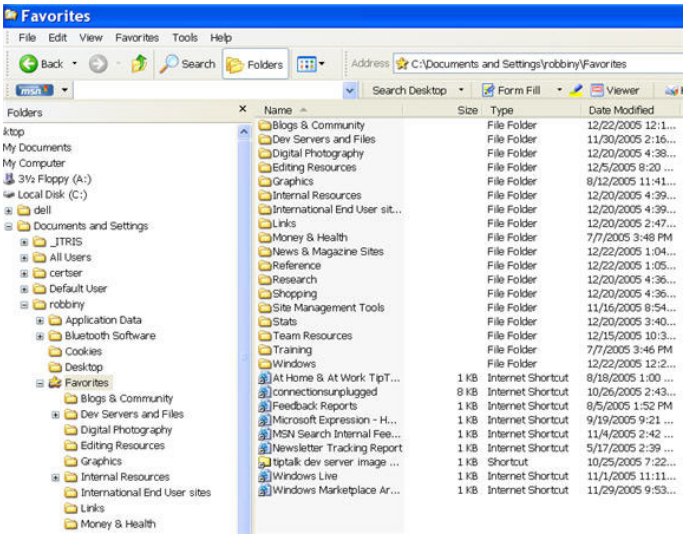


FIGURE 8 – Arborescence d'un système de fichiers

V La programmation

1 Le traitement de l'information

La numération usuelle à laquelle nous sommes familiers est la numération décimale positionnelle qui s'appuie sur la base 10. En informatique, c'est la base deux qui est adaptée. Les transistors, constituants élémentaires des microprocesseurs, gèrent des systèmes à deux états d'où le choix du système binaire.

Définition 6. *L'unité d'information élémentaire valant 0 ou 1 s'appelle le* bit *(contraction de binary digit). Une séquence de* $2^3 = 8$ *bits s'appelle un* octet *(byte en anglais, attention à ne pas confondre avec le bit). Un kilo-octet (Ko) est constitué de* 10^3 *octets, un méga-octet (Mo) est constitué de* 10^6 *octets, un gigaoctet (Go) de* 10^9 *octets etc. . . .*

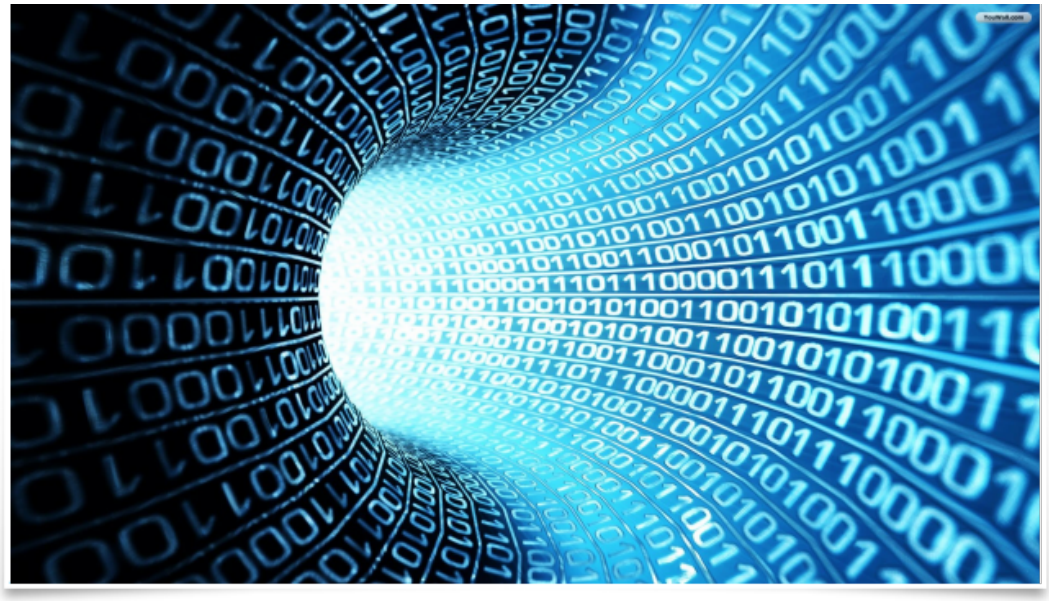


FIGURE 9 – Un monde binaire

Remarque : Aux débuts de l'informatique, le sens des préfixes était entendu différemment, à savoir : un kilo-octet pour $1024 = 2^{10}$ octets, un méga-octets pour 1024 kilo-octets soit 2^{20} octets, etc. ... Depuis décembre 1998, l'IEC (International Electrotechnical Commission) a décidé d'uniformiser l'usage des préfixes avec leur usage courant pour les autres unités : kilo pour 10^3, méga pour 10^6, ... Toutefois, cette normalisation est artificielle car l'information numérique est binaire par nature et les puissances de 2 restent, quoi qu'en disent les organismes décidant des appellations, les bases de dimensionnement des entités mémoires.

Définition 7. *La base* $2^4 = 16$ *dite* hexadécimale *est également utilisée en informatique. On compte ainsi :*

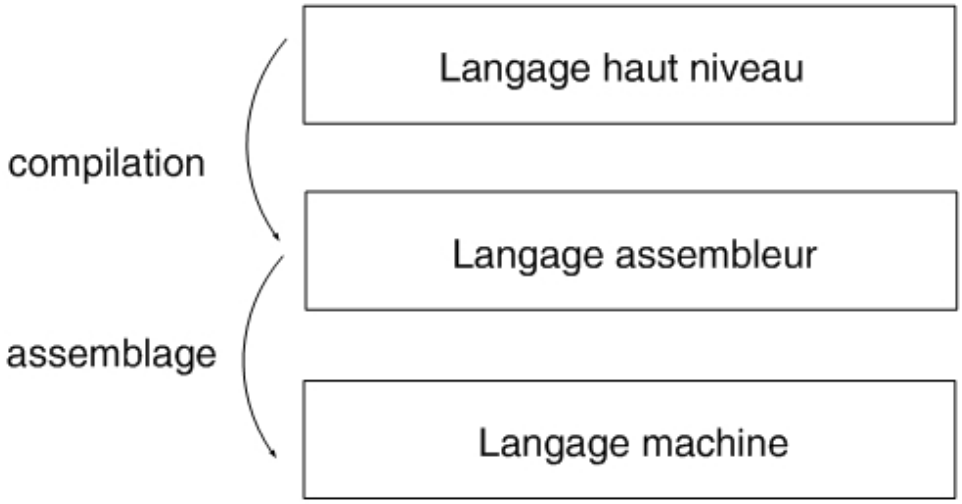
$$0, 1, \ldots, 9, \mathrm{A}, \mathrm{B}, \ldots \mathrm{F}, 10, 11, \ldots, 1\mathrm{A}, \ldots, 1\mathrm{F}, 20, \ldots$$

Cette base est pratique et compacte puisque deux caractères hexadécimaux permettent le codage d'un octet pouvant prendre 2^8 valeurs distinctes, de 0 à 255 en base décimale et de 00 à FF en base hexadécimale.

2 Langages de programmation

Il existe une hiérarchie des langages de programmation :

- Le langage *machine* ;
- le langage *assembleur* ;
- le langages de haut-niveau.



Le langage machine est le langage compris par le microprocesseur. Ce langage n'est pas du tout adapté à la compréhension humaine puisque chaque instruction est codée par une séquence de bits, autrement dit une succession de 0 et de 1.

Le langage assembleur est un langage proche du langage machine (bas niveau) tout en étant compréhensible par le programmeur. Il est composé par des instructions en général assez rudimentaires appelées *mnémoniques*. Ce sont essentiellement des opérations de transfert de données entre les registres et l'extérieur du microprocesseur (mémoire ou périphérique) ou des opérations arithmétiques ou logiques. Le langage assembleur est propre à chaque architecture de microprocesseur. Cette forte dépendance à la machine et le caractère véritablement fastidieux de la programmation en mnémoniques ont nécessité la conception de langages de haut niveau.

Le langage de haut niveau est un langage permettant d'écrire des programmes en utilisant des mots usuels (l'anglais essentiellement) et des symboles mathématiques familiers. Faisant abstraction de toute architecture de machine, ces langages permettent l'expression d'algorithmes sous une forme plus facile à apprendre et à employer. Chaque instruction en langage de haut niveau correspondra à une succession d'instructions en langage assembleur. Les langages de haut niveau sont par exemple le C, java, python, etc. ...

L'étape traduisant le langage de haut-niveau en langage assembleur est la *compilation* suivi de l'étape d'assemblage qui convertit le langage assembleur en code machine compréhensible par le microprocesseur.

Définition 8. *Un* compilateur *est un programme informatique qui traduit un langage source en langage cible, en général assembleur.*

Exemple : Voici une même ligne de code écrite en C, en assembleur et en code machine.

Code machine (68HC11)	Assembleur (68HC11)	Langage C
C664 B600 1B 5A 2603	LDAB #100 LDAA #0 ret ABA DECB BNE ret	A=0 for(i=1 ;i<101 ;i++) A=A+i

On peut aussi procéder à l'exécution des instructions d'un programme au fur et à mesure de leur lecture. Ce procédé, *l'interprétation*, consiste en l'exécution dynamique du programme par un interpréteur sans conversion en un langage cible.

Définition 9. *Un* interpréteur *est un programme informatique ayant pour tâche d'analyser et d'exécuter un programme écrit dans un langage source.*

L'interprétation est idéale en phase d'apprentissage d'un langage ou en cours d'expérimentation sur un projet. On peut facilement tester des portions de programmes sans passer par une compilation parfois lourde de l'ensemble d'un programme source.

Un compilateur traduit un code source en un fichier indépendant exécutable tandis que l'interpréteur est nécessaire à chaque lancement du programme interprété. La traduction à la volée est donc source de lenteur. Mais l'amélioration constante des processeurs attenuant ce problème de lenteur, la facilité d'évolution d'un programme interprété et la rapidité de mise en place de petits programmes rendent les langages interprétés aujourd'hui indispensables.

3 Python, Idle

Le langage python est un langage puissant, lisible et simple à apprendre. Il est à mi-chemin entre le langage compilé et le langage interprété : le langage python est compilé en langage intermédiaire appelé *bytecode*, lui-même interprété au sein d'une machine virtuelle.

L'environnement retenu pour l'apprentissage sous python s'appelle *Idle*. C'est un environnement simple et robuste.

Pour installer python sur son ordinateur :
`https://sourceforge.net/projects/winpython/files/latest/download`

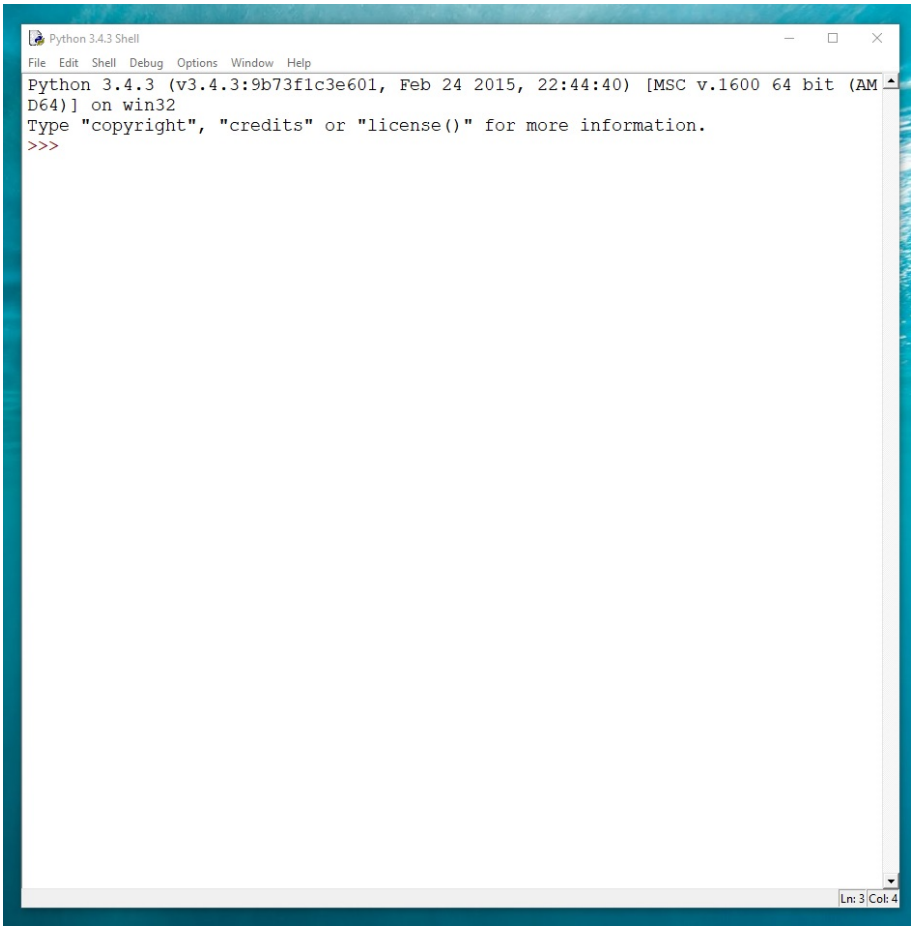


FIGURE 10 – Lancement de la console Idle

```
>>> 1+2
3
>>> 5*3
15
>>> |
```

Dans la console, on saisit des instructions comme sur une calculatrice et celles-ci sont exécutées après appui de la touche `Entrée`.

Si l'on veut écrire un script (un fichier que python sait exécuter), il faut lancer l'éditeur. Pour cela, on ouvre un nouveau fichier depuis la console :

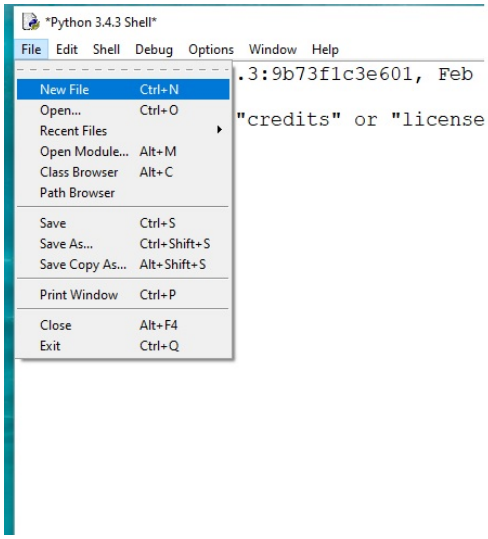


FIGURE 11 – Lancement de l'éditeur Idle

On recommande de placer les deux fenêtres côte à côte pour travailler :

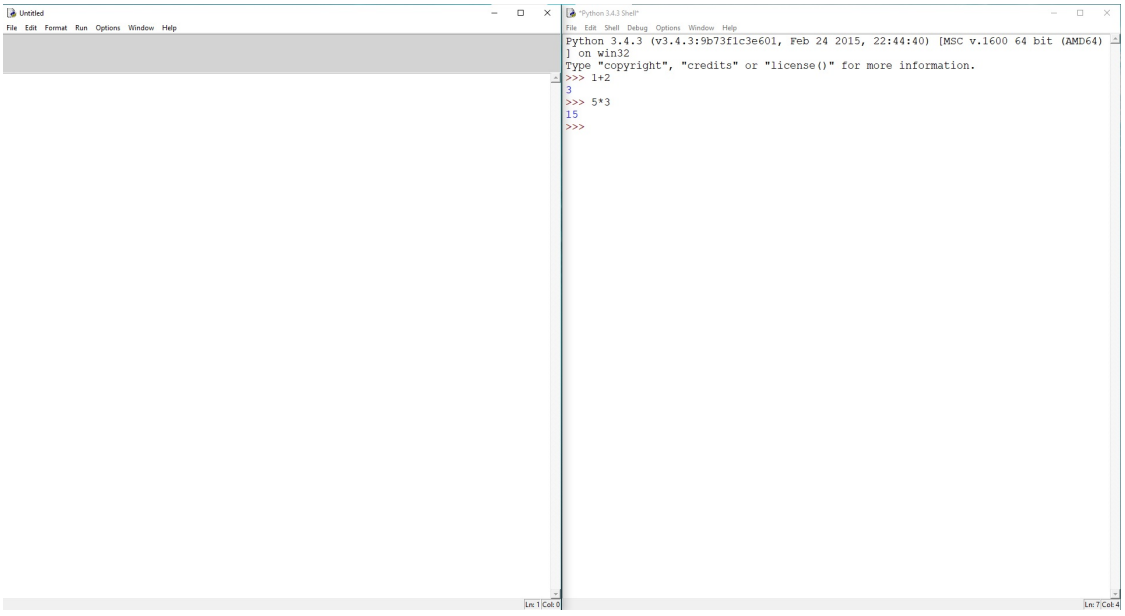


FIGURE 12 – Éditeur et console idle côte à côte

Une fois les instructions saisies dans l'éditeur, on lance son exécution avec `Run Module` ou F5. L'éditeur va demander d'enregistrer le script (choisir un répertoire et un nom de fichier adapté) puis l'exécuter dans la console.

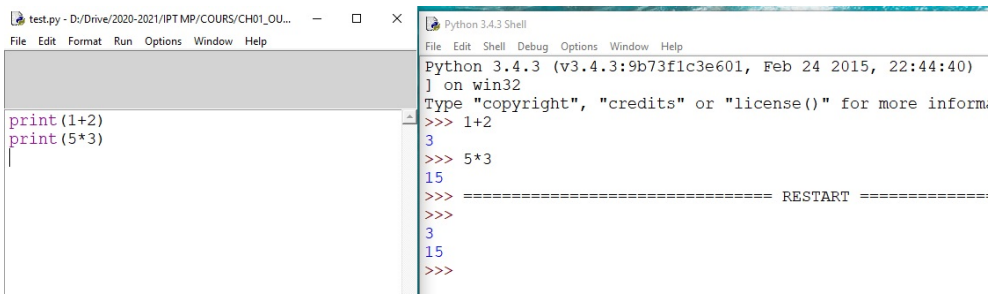


FIGURE 13 – Exécution d'un premier programme