

# Prise en main d'un système informatique

## 1 Introduction

Ce document est destiné aux personnes amenées à travailler avec un ordinateur et désirant connaître le minimum indispensable pour se livrer à leur occupation favorite : écrire des rapports ou du courrier, compiler des programmes, utiliser un logiciel de dessin, etc.

Beaucoup d'utilisateurs débutants de systèmes informatiques sont déroutés par cet univers complexe qu'ils découvrent, et que certains comparent -à juste titre- à une jungle qu'ils ne savent par où pénétrer. La complexité, dans ce cas, naît de la quantité énorme d'informations disponibles. Le risque est ici plus grand d'être submergé d'informations, que d'en manquer !

Mon but sera donc d'ouvrir un premier chemin dans cette jungle, qui permettra à l'utilisateur de parvenir rapidement à ses fins dans un premier temps, et lui donnera les moyens pour explorer lui-même par la suite les secteurs qui l'intéressent plus particulièrement.

La présente version de ce document est destinée aux utilisateurs de compatibles PC sous MS-DOS. Il existe cependant des versions de ce document adaptées à divers environnements informatiques.

## 2 Premières notions

### 2.1 Structure d'un ordinateur

Les principales composantes sont le **processeur** ou unité centrale, la **mémoire centrale** et les **unités d'échange**.

Le processeur est "l'âme" de l'ordinateur, c'est lui qui effectue les calculs et qui contrôle l'exécution des programmes.

La mémoire centrale sert à stocker les programmes et les données en cours de traitement. Il existe d'autres mémoires qui peuvent être extérieures à l'ordinateur : les disques, disquettes, bandes magnétiques, sur lesquelles nous reviendrons plus en détail dans la section suivante.

Les unités d'échange permettent la communication entre l'ordinateur et ses auxiliaires, appelés *périphériques* : clavier, écran, souris, imprimantes, mémoires externes, etc.

Processeur, mémoire centrale et unités d'échange communiquent entre eux par un ensemble de connexions appelé **bus**.

## 2.2 Mémoires

Comme nous l'avons vu, un ordinateur possède plusieurs mémoires. Cela ne doit pas nous étonner : nous possédons, nous aussi êtres humains, plusieurs types de mémoire qui nous servent dans des situations spécifiques. Par exemple, si nous suivons un raisonnement, nous utilisons notre **mémoire à court terme** pour stocker les étapes intermédiaires, tandis que si nous vivons un événement ``inoubliable'', nous nous arrangeons pour en graver les principaux traits dans notre **mémoire à long terme**.

Les caractéristiques de ces deux types de mémoire sont différentes : la mémoire à court terme permet de se rappeler très vite une donnée, mais elle est *volatile*, ce qui signifie qu'au bout de quelques heures ou quelques jours, les données mémorisées sont perdues. De plus, la mémoire à court terme est très limitée, elle ne peut contenir beaucoup d'informations à la fois. Par contre, la mémoire à long terme est beaucoup moins volatile, moins limitée mais parfois aussi plus lente : il faut souvent faire un effort, qui peut parfois prendre plusieurs secondes voire plus longtemps, pour se rappeler un souvenir ancien.

La mémoire d'un ordinateur se divise elle aussi en mémoire à court terme et mémoire à long terme.

La mémoire à court terme, dite aussi *mémoire vive* ou *mémoire centrale* dans notre jargon, est constituée de circuits mémorisant situés près de l'organe de traitement (le ``processeur''). Elle est **rapide** : une donnée élémentaire peut être lue ou écrite dans cette mémoire en moins d'un millionième de seconde. Elle est **volatile** : si vous éteignez votre ordinateur, le contenu de la mémoire vive sera perdu. Elle est **limitée** : bien que de plus en plus grandes, les mémoires centrales des ordinateurs sont relativement ``petites" devant la quantité d'informations que l'on doit stocker.

La mémoire à long terme, également appelée *mémoire de masse*, est le plus souvent basée sur des disques magnétiques (disques durs, disques souples ou disquettes). Plus **lente** que la mémoire centrale (une lecture ou écriture élémentaire sur un disque a une durée de l'ordre de plusieurs millièmes de seconde), elle est par contre **permanente** (sauf accident !) et facilement **extensible** pour couvrir tous besoins en matière de capacité de stockage.

## 2.3 Fichiers

Si j'insiste tant sur les divers types de mémoire dans ce manuel qui concerne principalement les notions de base des systèmes d'exploitation, c'est précisément parce que la gestion de la mémoire, et en particulier de la mémoire de masse, est l'une des principales tâches dont est chargé le système d'exploitation. De fait, votre première préoccupation en tant qu'utilisateur est certainement de conserver et d'organiser le fruit des longues heures de travail que vous vous préparez à passer devant votre clavier.

Le système d'exploitation (UNIX, MS-DOS, ou autre) est un logiciel qui vous aide à tirer parti de la machine qui, sans lui, ne vous serait sans doute pas d'une grande utilité. Il joue le rôle d'interprète entre vos désirs et le niveau où ils se réalisent effectivement, celui des registres, des ``traps'', des ``checksums" et autres ``boot blocks", bref, un univers dans lequel vous n'avez pas forcément l'envie (ou le temps) de vous plonger. La première chose que vous

exigerez de lui, c'est qu'il prenne soin de vos informations personnelles, même quand vous n'êtes pas là.

De votre point de vue, vos données peuvent être de natures très différentes : des programmes, des images, des tables de nombres, du courrier, que sais-je encore. Mais pour le système d'exploitation, vos données sont des données, un point c'est tout. De SON point de vue, tout ensemble de données que l'utilisateur désire garder en mémoire de masse (sur un disque ou une disquette, par exemple) constitue un *fichier* (*file* en anglais).

Ce mot ``fichier'', qui nous rappelle l'époque héroïque où programmes et données étaient conservés sous forme de fiches perforées, recouvre en fait une notion très simple et générale : si vous écrivez un programme ou du courrier grâce à un éditeur de texte, l'éditeur sauvegardera votre travail dans un fichier ; si vous compilez un programme pour produire du code machine exécutable, ce code machine sera placé par le compilateur dans un fichier ; si vous dessinez un avion avec un logiciel de dessin, ce dernier produira un fichier pour garder votre dessin de façon permanente.

Pour retrouver vos informations par la suite, vous devez bien sûr donner un nom à chacun de vos fichiers. Selon le système d'exploitation, des contraintes diverses s'appliquent à la construction d'un nom de fichier : MS-DOS, par exemple, les limite à huit caractères pour le nom et trois pour l'*extension*, qui est un appendice du nom séparé de celui-ci par un point (par exemple : PROGRAM.PAS). De plus, MS-DOS ne fait pas de différence entre majuscules et minuscules, et certains caractères spéciaux (notamment < et >) ne peuvent être utilisés dans un nom de fichier. Vous éviterez des ennuis, au début, en n'utilisant à cet effet que des lettres et des chiffres, ou en vous référant à la documentation du système.

## 2.4 Répertoires

Lorsque vous commencerez à travailler sur une machine, vous ne manipulerez que quelques fichiers ; mais très vite, si vous continuez à travailler et à produire régulièrement de nouveaux documents, vous serez submergé par une masse d'informations qu'il vous faudra organiser pour vous y retrouver.

La quasi-totalité des systèmes d'exploitation utilisent un modèle hiérarchique d'organisation des fichiers, basé sur la notion de *répertoire* (en anglais *directory*).

Le répertoire est l'unité d'organisation, il peut ``contenir'' des fichiers, ainsi que d'autres répertoires. Il est également nommé, les noms des répertoires obéissant en général aux mêmes contraintes que les noms de fichiers.

Prenons un exemple : vous êtes programmeur, vous devez écrire un logiciel de gestion baptisé HAPPY TAX PAYERS (HTP en abrégé), vous devez également produire la documentation de ce logiciel, et de plus vous apprenez le Japonais par la Formation Continue grâce à un programme d'E.A.O. (Enseignement Assisté par Ordinateur). Vous aurez tout intérêt à séparer clairement les produits de ces différentes activités en créant trois répertoires, nommés HTP, DOCS et JAPON par exemple.

Si maintenant votre logiciel de gestion est déjà opérationnel, que vous désirez garder la version actuelle mais apporter des améliorations pour une seconde version, la meilleure solution consistera à créer un répertoire VERSION1 dans le répertoire HTP, où vous

recopiez l'ensemble des fichiers constituant votre logiciel, ainsi vous pourrez ensuite modifier les fichiers originaux (qui deviendront ainsi la seconde version) en toute tranquillité, sachant qu'une version propre a été sauvegardée dans le sous-répertoire VERSION1.

Supposons de plus que vous travailliez sur un système informatique dont les disques magnétiques sont partagés par plusieurs utilisateurs (situation courante dans le cadre de gros ou moyens systèmes ou encore de réseaux), dans ce cas chaque utilisateur se voit attribuer un répertoire protégé vis à vis des autres utilisateurs, mais dans lequel il est libre d'organiser son travail à sa façon.

#### 2.4.1 Chemin d'accès

La structure hiérarchique constituée par les répertoires s'appelle un arbre. Pour vous représenter ce type de structure, pensez à un arbre généalogique. Un répertoire particulier, qui n'appartient lui-même à aucun répertoire de rang plus élevé, est appelé *racine* (*root* en anglais). La racine est désignée par un nom ou un symbole standardisé :

\

Comme votre ordinateur peut être équipé de plusieurs unités de disquettes et disques durs, ce symbole désignant la racine peut être précédé par le nom d'une unité, par exemple A:, B:, C: ... Exemple :

C: \

Pour retrouver un fichier dans cette structure, vous devez en connaître le *chemin d'accès* (*path* en anglais), c'est-à-dire la suite des sous-répertoires qu'il faut ``traverser'', depuis la racine, pour arriver dans celui où se trouve le fichier. Par exemple :

C:\Users\E3\Desktop\Cours-C-esiee\site-public.html

#### 2.4.2 Répertoire courant - Unité courante

Heureusement, il n'est pas nécessaire de taper au clavier tout le chemin d'accès à chaque fois que l'on désire utiliser un fichier. Une commande (**cd**) permet de définir un répertoire courant, que l'on peut changer à volonté. Par exemple, si le répertoire courant est \E3\Desktop\Cours-C-esiee\, il me suffira de taper site-public.html pour faire référence au fichier C:\Users\E3\Desktop\Cours-C-esiee\site-public.html

De même, la notion d'unité courante permet de rendre implicite le nom de l'unité sur laquelle on travaille. Pour changer d'unité courante, il suffit de taper son nom, par exemple :

a :

ou U :

### 2.4.3 Caractères "joker"

Imaginons la situation suivante : vous avez créé des fichiers temporaires que vous avez appelés TEMP1, TEMP2, ... , TEMP9, TEMPOR10 et TEMPOR20. Les ayant exploités, vous désirez maintenant vous en débarrasser à l'aide de la commande **del** qui détruit des fichiers. Devrez-vous taper 11 fois cette commande, une fois pour chacun de ces fichiers ? Il existe une façon plus efficace de réaliser ce travail, en utilisant un caractère "joker" (*wildcard* en anglais). Sous MS-DOS, le caractère "?" est un "joker" qui peut prendre la place de n'importe quel caractère (**hormis le caractère "."**). Cela signifie que si vous tapez :

**del** TEMP? vous détruirez d'un seul coup les fichiers TEMP1, TEMP2, ... , TEMP9. Il existe un second "joker", le caractère "\*", qui remplace une séquence quelconque de caractères (ne comportant pas de "."). Donc, si au lieu de la commande ci-dessus vous aviez tapé :

**del** TEMP\* tous les fichiers TEMP1, TEMP2, ... , TEMP9, TEMPOR10 et TEMPOR20 auraient été détruits.

Mais **attention** : si vous lancez la commande :

**del \*.\*** vous détruirez **tous les fichiers** qui se trouvent sur le répertoire courant. Moralité : les caractères joker, associés aux commandes d'effacement, doivent être utilisés avec la plus grande prudence.

## 3 Les principales commandes

Les principales commandes utilisées sont :

- md : **Make Directory**, pour créer un répertoire)
- cd : **Change Directory**, pour changer de répertoire de travail)
- rd : **Remove Directory**, pour détruire un répertoire)
- copy : pour copier des fichiers)
- dir : pour lister les fichiers)
- cls : abréviation de **CLear Screen** pour effacer l'écran)
- del : pour détruire un fichier)

avant d'aborder les commandes proprement dites, signalons que la plupart d'entre elles peuvent être utilisées telles quelles ou accompagnées d'*options* : une option est un appendice que l'on rajoute à une commande pour en modifier le sens ; par exemple la commande

DIR donne une liste détaillée indiquant le nom des fichiers, leur taille et leur date de création, tandis que la même commande utilisée avec l'option /w :

DIR /w donne simplement la liste des fichiers sur un minimum de lignes. Signalons encore l'option /p qui, lorsque la liste est longue, permet d'obtenir une pause chaque fois que l'écran est plein.

**Pour connaître toutes les options d'une commande, il faut ajouter l'option /? après la commande.**

Certaines commandes nécessitent d'être accompagnées d'un ou de plusieurs *paramètres*. Le cas le plus courant est celui d'une commande agissant sur un fichier, par exemple la commande del (delete) qui efface un fichier. Cette commande doit bien sûr être suivie d'un paramètre, le nom du fichier à détruire :

DEL *file*

**Dans la suite comme dans cet exemple, les paramètres (à remplacer par vos chemins d'accès) apparaîtront en italique.**

Voici maintenant une brève description des quelques commandes indispensables que vous devez connaître pour gérer vos fichiers.

## 3.2 Commandes agissant sur les fichiers

### ◆ Copier un fichier dans un autre

**COPY** *source\_file destination\_file*

(COPY= recopier ou dupliquer) Le contenu du fichier *destination\_file* est remplacé par celui du fichier *source\_file*. Si le fichier destination n'existait pas avant l'opération, il est créé.

### ◆ Déplacer un fichier vers un autre répertoire

**MOVE** *source\_file destination\_directory*

(MOVE= déplacer) Le fichier *source\_file* n'est plus dans son répertoire d'origine mais dans le *destination\_directory*. Si un fichier de même nom existait déjà dans le répertoire destination, la commande est refusée.

Il est également possible de profiter de cette commande pour changer le nom du fichier ; il suffit de le préciser à la fin de la commande :

MOVE *source\_file destination\_directory\new\_filename*

### ◆ Effacer un fichier

DEL *file*

(DELeTe= effacer) Le contenu du fichier *file* est détruit, et son nom est retiré de la liste du répertoire.

#### ♦ Changer le nom d'un fichier

**REN** *old\_file\_name new\_file\_name*

(REName= renommer) Le fichier de nom *old\_file\_name* est renommé et prend le nom *new\_file\_name*.

### 3.3 Commandes agissant sur les répertoires

#### ♦ Demander la liste des fichiers et des sous-répertoires

**DIR**

(DIRectory= répertoire) Donne la liste des fichiers se trouvant dans le répertoire (directory) courant. De nombreuses options sont généralement disponibles avec cette commande, pour obtenir des renseignements supplémentaires sur les fichiers, ou pour formater agréablement l'affichage.

Il est également possible de demander le contenu d'un autre répertoire ; il suffit de le préciser à la fin de la commande :

**DIR** *répertoire*

#### ♦ Changer de répertoire courant

**CD** *path*

( Change Directory= changer de répertoire) Le paramètre *path* spécifie le chemin d'accès du nouveau répertoire courant de l'unité courante.

#### ♦ Afficher le chemin du répertoire courant

**CD**

#### ♦ Créer un nouveau répertoire (directory)

**MKDIR** *path*

(MaKe DIRectory (ou MD)= créer un répertoire) Un nouveau (sous-)répertoire est créé. Son chemin d'accès est spécifié par le paramètre *path*.

#### ♦ Détruire un répertoire (directory)

**RMDIR** *path*

(ReMove DIRectory (ou RD)= retirer un répertoire) Le répertoire spécifié par le chemin d'accès *path* est détruit.

**Attention :** certains systèmes d'exploitation (comme MS-DOS) exigent pour détruire un répertoire que celui-ci soit vide de fichiers et de sous-répertoires ; tandis que d'autres détruisent sans complexe des arborescences complètes.