Chapitre 7

Les périphériques et les fichiers

7.1 Introduction

Les périphériques sont des éléments externes au triplet processeur, bus, mémoire de l'architecture de Von Neumann. Ce sont pourtant des éléments indispensables à l'utilisation d'un ordinateur. En effet, c'est grâce à eux (clavier, souris, écran) qu'un utilisateur peut dialoguer avec l'ordinateur. C'est aussi grâce à eux qu'il peut produire des documents sur support papier (imprimante, table traçante). La troisième grande fonction est de conserver l'information (disques magnétiques, lecteurs de cédérom, dérouleur de bande, lecteur de cartouche, etc.).

Enfin, les ordinateurs ne sont plus aujourd'hui des machines isolées, et la mise en réseau nécessite elle aussi une connexion qui doit être pilotée par le système d'exploitation, mais ici on ne peut plus considérer qu'il s'agit d'un périphérique au sens strict. L'étude des réseaux, du point de vue matériel ou logiciel sort du cadre d'un cours d'architecture des ordinateurs.

7.1.1 Périphérique–Connexion–Contrôleur–Pilote

Un périphérique est donc un élément extérieur à l'ordinateur qui est capable d'exécuter des ordres sous le contrôle de celui-ci. Typiquement, ce contrôle s'exerce au moyen d'un dialogue (qui peut parfois être extrêmement fruste) entre l'ordinateur et le périphérique. Des signaux électriques sur des fils sont le support de ce dialogue, on parle de la connexion entre l'ordinateur et le périphérique. Les prises, leurs brochage, les valeurs des tensions, courants ou fréquence qu'on y trouve font partie de la définition de cette connexion. Selon les périphériques, on trouve de nombreuses normes :

- RS-232, V-24, RS-422, pour la connexion des modems et des terminaux alphanumériques,
- RS-232, Centronics, pour les imprimantes,
- ESDI, ST506, IDE, SCSI, pour les disques magnétiques,
- HP-IB, IEEE-488, pour les périphériques d'instrumentation, voire pour les disques,
- etc.

Cependant, il y a deux grandes catégories de connexions, les connexions séries et les connexions parallèles comme nous le verrons dans la section 7.2.

Du côté de l'ordinateur, il faut aussi une adaptation entre le processeur et la prise de sortie. Le processeur ne sait dialoguer qu'au moyen de son bus d'adresses et de son bus de données. Le *contrôleur* est donc chargé de faire cette adaptation : d'un côté, il s'interface avec le processeur sur son bus d'adresses et son bus de données, et de l'autre côté, il assure le dialogue avec le périphérique.

Toutefois, le contrôleur ne s'occupe que d'un interfaçage physique, il doit lui-même être *piloté* par le système d'exploitation : ainsi pour chaque périphérique connecté à un ordinateur, il faut dans le système d'exploitation le *pilote* associé pour que le périphérique puisse être pris en compte.

En résumé, pour un périphérique, il faut avoir dans l'ordinateur le *contrôleur* correspondant (élément matériel) et le *pilote* (élément logiciel) pour qu'il soit utilisable.

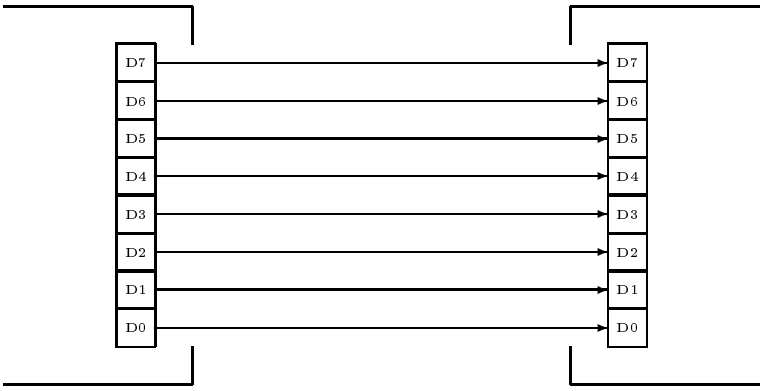

FIG. 7.1 – Connexion parallèle

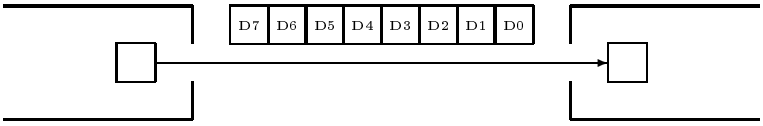

FIG. 7.2 – Connexion série

7.2 Connexions série et parallèle

On sait qu'un bit peut coder, stocker une quantité élémentaire d'information : une valeur binaire. Pour pouvoir coder et stocker des quantités d'informations plus grandes, plusieurs bits sont nécessaires, et autant de supports de stockage sont utilisés. Une taille courante d'unité d'information est l'octet qui est un groupement de huit bits. Lorsqu'une unité d'information (par exemple un octet) doit être transmise d'un point à un autre, on peut mettre autant de supports physiques de transmission qu'il y a de bits à transmettre (par exemple huit fils qui transmettent l'information sous forme d'une tension électrique) : on parle d'une *transmission parallèle*, les différents bits d'information sont transmis en même temps. L'autre possibilité consiste à transmettre les différents bits les uns à la suite des autres dans le temps sur le même support physique : on parle d'une *transmission série*.

Aux connexions des fils qui transportent les données, peuvent être associés des fils pour transporter des informations de contrôle, par exemple sur une connexion parallèle, on trouve un signal qui indique la validité de l'état des fils de données, un autre positionné par le récepteur qui indique s'il est prêt à recevoir une donnée, etc.

7.2.1 Connexion série

Dans une connexion série, il y a une notion d'horloge qui est associée. On peut avoir une transmission asynchrone, l'horloge n'est pas transmise et la ligne n'est pas utilisée en permanence pour transmettre des données. Dans ce cas, la ligne est normalement à l'état zéro. Le *bit de start* (un bit à un) indique le début de transmission d'un flot de bit, suivent les bits de données et éventuellement un bit de parité et un ou deux bits de stops.

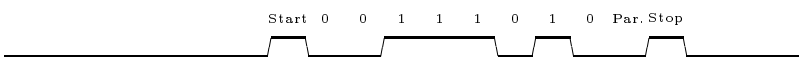

FIG. 7.3 – Transmission sur une ligne série

7.2.1.1 RS-232

Avant transmission, les deux correspondants doivent être d'accord sur la configuration de la ligne c'est-à-dire sur le nombre de bits de données transmis, la présence d'un bit de parité, le nombre de bits de stop, et surtout la vitesse de transmission. Les vitesses normalisées de transmissions vont de 75 à 38400 bits par seconde en multipliant par 2 à chaque pas (75, 150, 300, ...). Cette configuration est locale à chacun des deux correspondants : elle n'est pas faite à travers la ligne. Enfin, les niveaux électriques, le brochage des connecteurs doivent bien sur être compatible. Selon la vitesse choisie la distance maximale varie et peut atteindre 300 mètres.

Deux normes (à peu près équivalentes) RS-232 et V-24 normalisent le mode de connexion entre un ordinateur et un modem (MOdulateur-DEModulateur). Ce type de jonction est très courant sur tous les ordinateurs et sert à connecter aussi bien des terminaux que des imprimantes en plus des modems.

7.2.1.2 MIDI

Une autre norme, très utilisée maintenant sur les instruments de musique électroniques, est la norme MIDI (Musical Instrument Digital Interface) qui utilise la présence d'un courant (à la place d'une tension) pour véhiculer les deux états binaires. Autrement, c'est une transmission série à 31250 bps avec huit bits de données, un bit de start et un bit de stop.

7.2.1.3 Autres utilisations des liaisons séries

Ce type de connexion est aussi utilisé (sous des formes propres aux constructeurs) pour relier le clavier et/ou la souris à l'unité centrale. Une autre utilisation plus originale d'une liaison série est celle qui est faite dans les transputers. Un transputer est un processeur prévu pour construire des machines parallèles (c'est-à-dire avec plusieurs processeurs qui communiquent entre eux). Pour assurer cette communication, chaque transputer possède quatre *liens de communications*. Chaque lien est en fait une connexion série à 10 Mbps. Bien sûr, pour pouvoir atteindre une telle vitesse sur une liaison série, la longueur de connexion est très limitée et restreinte à l'intérieur d'un boîtier.

7.2.2 Connexion parallèle

Les connexions séries permettent d'économiser du cablage et d'avoir de grandes longueurs de connexion au détriment de la vitesse de transfert. Souvent, pour avoir de grands débits, les liaisons parallèles sont préférées. Ainsi on les utilise couramment pour les disques magnétiques, nous le verrons dans la section sur les disques (7.6).

7.2.2.1 Centronics

La norme *Centronics* définit une connexion pour les imprimantes (voir le brochage dans la table 7.1). Le protocole est très simple. Par le bit *Busy*, l'imprimante informe le contrôleur si elle peut accepter un caractère ou non. Lorsque le bit *Busy* est bas, le contrôleur positionne les bits de données, puis valide le bit *Strobe*. L'imprimante échantillone alors le bus de données pour recevoir le caractère émis par le contrôleur.

7.3 Les claviers

Un clavier est une matrice de contacts. Le traitement est assuré le plus souvent par un processeur local au clavier qui balaye en permanence l'ensemble des contacts et transmet une information prête à utiliser par le processeur central. Cette information est le plus souvent transmise grâce à une ligne série et c'est soit un code ASCII, soit un code de numérotation propre au clavier qui est transmis.

1	STROBE*
2	D0
3	D1
4	D2
5	D3
6	D4
7	D5
8	D6
9	D7
10	Non connectée
11	BUSY
12–17	Non connectées
18–25	Masse

TAB. 7.1 – Brochage Centronics

7.4 Les périphériques de visualisation

7.4.1 Les terminaux alphanumériques

Naguère très employés dans les terminaux, les périphériques de visualisation alphanumériques sont moins utilisés de nos jours et ont laissé la place à des périphériques de visualisation graphiques. Pratiquement, ce sont des micro-ordinateurs spécialisés avec une carte mémoire qui contient une zone où chaque cellule conserve un caractère à afficher sur l'écran. Le plus souvent l'écran lui-même affiche 24 lignes de 80 caractères chacune, et la zone mémoire doit au moins avoir cette taille ; elle peut éventuellement être plus grande et conserver alors plusieurs écrans ou plus de lignes que celles qui sont effectivement affichées.

Chacune des cellules de l'écran code le numéro du caractère (par exemple en ASCII) et éventuellement d'autres informations comme le numéro de la police à utiliser, ou divers attributs (souligné, double largeur, gras, inverse vidéo,etc.).

Cette zone mémoire est à *double port* : elle est accessible par le processeur (le processeur local s'il s'agit d'un terminal ou d'une carte de visualisation autonome) et par une circuiterie qui balaye en permanence cette zone mémoire pour fabriquer les signaux vidéos nécessaires au pilotage d'un tube cathodique. Un *générateur de caractères* contient le dessin en pixels des caractères. Ce générateur est généralement une mémoire à lecture seule (une ROM). Voici par exemple son contenu pour le caractère A majuscule :

```
 ***          00011100
*   *         00100010
*   *         00100010
*****         00111110
*   *         00100010
*   *         00100010
              00000000
              00000000
```

7.4.2 Les terminaux graphiques

Les mémoires de trame sont les plus courantes, il existe aussi des terminaux à balayage cavalier. Dans une mémoire de trame, l'écran est découpé en zones rectangulaires (le plus souvent carrées d'ailleurs) d'assez petite taille : les *pixels*, agencés en lignes et colonnes. Des résolutions courantes vont de 640×400 à 1280×1024.

Chaque pixel possède un ou plusieurs bits de mémoire. Si un seul bit est disponible, l'écran sera dit monochrome et la visualisation montrera le pixel soit dans l'état allumé, soit dans l'état éteint. Si

plusieurs bits sont disponibles pour chaque pixel, ceux-ci pourront être utilisés pour coder différents niveaux d'intensité ou plusieurs couleurs.

Encore une fois, la mémoire associée au pixel est de la mémoire double port adressée par un processeur et par le circuit de *rafraichissement* de l'écran, qui balaye la totalité de cette mémoire en permanence entre 25 et 70 fois par seconde pour générer les signaux destinés au tube cathodique. Il existe d'ailleurs des mémoires RAM dynamiques spécialisées pour cet usage : les VRAMs (Vidéo RAM).

Pour afficher des choses sur l'écran, il faut donc écrire certaines valeurs dans certains emplacements mémoire. Ce peut être soit le processeur principal qui le fait ou un processeur dédié à cette tâche : le processeur graphique. Dans ce deuxième cas, le processeur principal donnera des ordres au processeur graphique à travers un contrôleur.

7.5 Les imprimantes

7.5.1 Les imprimantes *caractères*

Les imprimantes "caractères" ne font que peu d'interprétation sur les octets reçus. La plupart sont imprimés tels quels les uns à la suite des autres. Il existe certains *caractères de contrôle* comme l'octet 10 qui code un LF (*Line Feed*) ou passage à la ligne : le papier avance d'une ligne, ou encore comme l'octet 13 qui code un CR (*Carriage Return*) qui ramène le chariot d'impression en colonne 0. Il en existe quelques autres qui sont plus ou moins universellement reconnus.

Il peut exister des codes non standard qui permettent d'obtenir des caractères spéciaux, par exemple les caractères accentués du français. Enfin, certaines imprimantes "caractères" permettent d'accéder à un mode graphique au prix d'une grande quantité d'informations à transférer entre l'ordinateur et l'imprimante.

Un pilote d'imprimante doit pouvoir faire quelques transformations sur le flot de caractères à imprimer avant de l'envoyer effectivement. Par exemple remplacer chaque LF par un LF et un CR, interpréter des codes de soulignement en engendrant des séquences contenant un BS (*BackSpace*) et le caractère de soulignement (_) après chaque caractère à imprimer. En plus il doit respecter le protocole prévu par le type de connexion de l'imprimante et tamponner le flot de caractères de façon à ne pas bloquer trop tôt le programme qui imprime.

7.5.2 Les imprimantes Postscript

Il existe depuis plusieurs années des imprimantes qui sont en fait des ordinateurs avec un processeur puissant et qui exécutent les programmes qu'ils reçoivent sur une connexion. L'exécution de ces programmes réalisent l'impression des pages. Voici un exemple de programme en Postscript qui imprime le mot *exemple* sur une page avec une police *Times-Roman* et de grande taille (ce sont exactement les octets ASCII de ce texte qu'il faut envoyer à une imprimante Postscript pour obtenir l'impression de la page) :

```
gsave
200 30 translate
65 rotate
/Times-Roman findfont
216 scalefont
setfont
0 0 moveto
(exemple) show
grestore
showpage
```

7.6 Les disques

Un disque est composé d'un ou plusieurs plateaux, chaque plateau possédant deux faces. Les faces de ces plateaux sont recouvertes d'un oxyde ferreux magnétisable. Pour les disques durs, ces surfaces doivent avoir un poli et une qualité de planéité très grande. En effet, les têtes magnétiques de lecture/écriture volent à une altitude de l'ordre de 2 microns au dessus de ces surfaces[1]. Cependant malgré tous les efforts, il reste toujours des parties du disque qui sont inutilisables. Le rôle du *formattage* est entre autres de détecter ces zones.

Sur chaque surface sont physiquement réparties des pistes concentriques. Une tête de lecture est associée à chaque face, toutes les têtes sont solidaires entre elles et se déplacent simultanément. Pour une position de tête, correspond une piste sur chaque face. L'ensemble des pistes correspondant à une position de tête sur l'ensemble des faces est un cylindre. Sur une disquette, il y a 40 ou 80 cylindres et donc 80 ou 160 pistes pour une disquette double face. Enfin chaque piste est découpée en secteurs. Un secteur comprend un entête, des données utiles (typiquement 256, 512, 1024 ou 4096 octets) et une queue. L'entête est composé de signaux de synchronisation et des informations comme le numéro de piste, le numéro de secteur, etc. La queue comprend un ou des octets de contrôle, un contrôle de parité, etc. Les informations sont écrites "en série" dans les secteurs. Différentes méthodes de codage existent : le problème est (comme pour une transmission série sur une ligne) d'avoir une horloge, de savoir où commencent les données, ...

7.6.1 Gestion des secteurs

L'unité de gestion d'un disque est le secteur logique. Ces secteurs logiques sont identifiés par un numéro entre 0 le nombre de ces secteurs logiques (N_s) moins 1. Le nombre de secteurs est le produit du nombre de surfaces (N_S) par le nombre de cylindres (N_c) et par le nombre de secteurs par piste (N_{spp}) : $N_s = N_S \times N_c \times N_{spp}$.

Le numéro de secteur logique est converti en numéro de piste et en numéro de secteur, et la surface (donc la tête) qui le supporte est déterminée. Ces opérations sont réalisées pour les disques les plus rustiques par le système d'exploitation ou le contrôleur. Pour les disques plus récents, cette conversion a lieu sur le disque lui-même. On peut classer ainsi les interfaces disque en deux classes, celles qui manipulent des numéros de secteur logique, et celle pour lesquelles le contrôleur doit sélectionner la tête, positionner les têtes sur le bon cylindre et enfin attendre que le bon secteur passe sous la tête.

7.6.2 Le bus SCSI

Les adresses de secteurs qui circulent dans les commandes SCSI sont des adresses logiques plutôt que des adresses physiques. Avec les interfaces de conception plus anciennes comme ESDI, c'est le système d'exploitation qui devait savoir où se trouvaient les secteurs (ou les pistes) défectueuses de façon à ne pas les utiliser. Le passage de l'adresse logique à une adresse physique étant faite au niveau du périphérique, cette conversion pour la non-utilisation des secteurs en mauvais état peut être beaucoup plus adaptée et beaucoup plus efficace. Ainsi avec un disque SCSI, le système d'exploitation voit une séquence de blocs contigus utilisables.

De plus, une interface SCSI permet de connecter plusieurs ordinateurs et plusieurs périphériques. Plusieurs commandes peuvent être en cours d'exécution en même temps : l'hôte qui initie une commande peut se déconnecter du bus après le lancement de la commande, ainsi il libère le bus jusqu'à ce que le périphérique soit prêt à lui répondre. Dans l'intervalle de temps ainsi libéré, le même hôte peut dialoguer avec d'autres périphériques.

Un périphérique SCSI peut aussi mettre plusieurs commandes en file d'attente et optimiser les déplacements de tête pour répondre plus rapidement à l'ensemble des commandes qu'il a reçu en implémentant une stratégie analogue à celle d'un ascenseur.

Un bus SCSI peut supporter jusqu'à huit appareils (périphériques ou hôtes), et chaque appareil peut supporter à son tour huit unités logiques, lesquelles peuvent avoir 256 sous-unités logiques.

[1] Une trace de doigt ou une poussière ont une épaisseur de 6 à 50 microns !

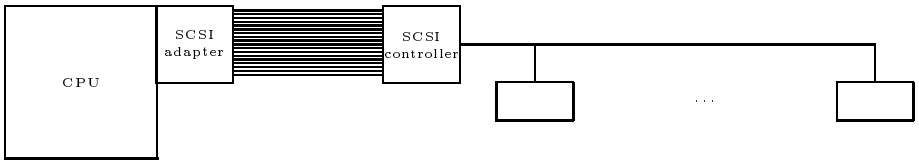

FIG. 7.4 – Un initiateur, une cible

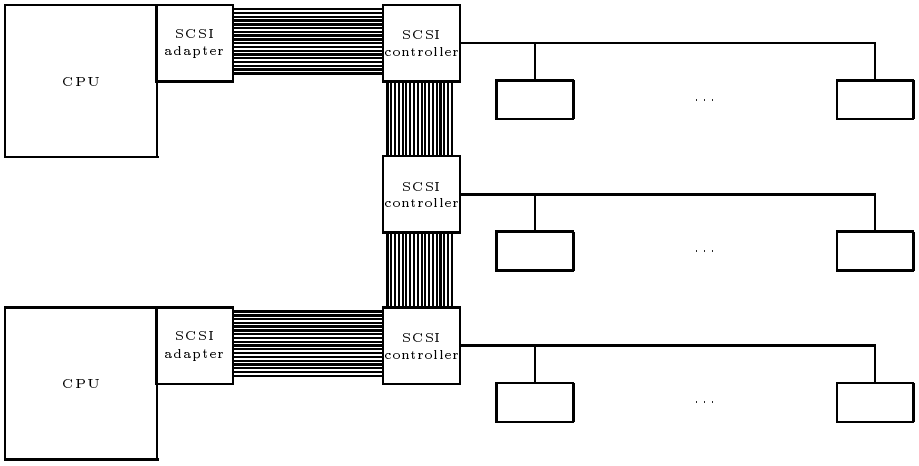

FIG. 7.5 – Multiples initiateurs, Multiples cibles

Chaque appareil peut être un *initiateur* (un appareil qui est à l'origine d'une commande), une *cible* (un appareil qui exécute une commande), ou les deux.

7.7 Les supports d'archivage et de sauvegarde

Il existe des périphériques d'un usage analogue aux disques magnétiques : les disques magnéto-optiques et les disques optiques numériques. C'est le mode d'écriture physique sur le support qui est différent. Les supports physiques pour ces disques sont amovibles et emballés dans une cartouche plastique. Ils sont insérés dans un périphérique lui-même. De ce fait et aussi grâce à leur capacité élevé, ils peuvent être utilisés comme moyen de sauvegarde ou d'archivage. Certains de ces disques ne peuvent être écrits qu'une seule fois ((en anglais : Write Once Read Many, WORM)). Ils sont le plus souvent connectés grâce à un port SCSI.

Les autres supports magnétiques sont les bandes, soit simplement enroulées sur une bobine, soit plus protégées dans des cartouches en plastique. La connection se fait aussi couramment en SCSI. Les capacités vont de 60 à 120 méga-octets. Sur support 8mm ou 4mm, on trouve aussi des périphériques qui peuvent stocker de l'ordre de 4 giga-octets.

7.8 Le système de gestion de fichiers

7.8.1 Un système de fichiers

Dans la hiérarchie mémoire, après la mémoire centrale, on trouve les disques (éventuellement les disquettes) magnétiques ou magnéto-optiques. Au niveau du support, ces média contiennent des informations sous forme de secteurs sur des pistes concentriques, elles-mêmes sur différents plateaux. Le niveau le plus bas du système d'exploitation (le *pilote* (en anglais : driver)) gère ces périphériques

de façon à ce que le niveau supérieur les voit en tant qu'ensembles de secteurs logiques, c'est-à-dire que le niveau supérieur voit une suite de N secteurs numérotés de 0 jusqu'à $N-1$. Cependant, cela reste une vision de bas niveau par rapport aux besoins de l'utilisateur. Aussi le système d'exploitation fournit d'autres services par rapport à ces média sous la forme de *systèmes de fichiers*.

On peut comparer un fichier à un livre— il y en a des petits et des gros— et un système de fichiers à une bibliothèque. La bibliothèque fournit des rayonnages pour poser les livres, mais possède aussi des index qui permettent de retrouver rapidement où est rangé un livre à partir de son titre. De la même façon, un *système de fichiers* permet d'organiser l'espace disque pour pouvoir retrouver les informations qui y sont conservés.

Un fichier est entièrement déterminé par son *chemin d'accès* qui est une chaîne de caractères. Avec le système MS-DOS, les chemins d'accès de fichiers sont de la forme :

<*support*> :\ <*hiérarchie de répertoire*> \ <*nom de fichier*> . <*extension*>

où :

support désigne une unité physique de stockage (`A` ou `B` pour les lecteurs de disquettes, `C`, `D`, ...pour les disques durs),

hiérarchie de répertoire donne une liste (éventuellement vide) de répertoires à "traverser" avant d'arriver au fichier lui-même, cela permet d'avoir une vision hiérarchique dans l'organisation de ses fichiers, chaque nom de répertoire est lui-même de la forme <*nom de fichier*> . <*extension*>, et ils sont séparés par des \ dans la liste,

nom de fichier est une suite d'au plus huit caractères alphanumériques,

extension est une suite d'au plus trois caractères alphanumériques.

En voici, quelques exemples : `A:\lettre.txt`, `C:\AUTOEXEC.BAT`, `C:\BIN\TEX.EXE`, `D:\MBEIG\COURS\CHAP07.TXT`.

Dans le système UNIX, les chemins d'accès de fichiers sont de la forme :

/ <*hiérarchie de répertoire*> / <*nom de fichier*>

où :

hiérarchie de répertoire donne une liste (éventuellement vide) de répertoires à "traverser" avant d'arriver au fichier lui-même, cela permet d'avoir une vision hiérarchique dans l'organisation de ses fichiers, chaque nom de répertoire est lui-même de la forme <*nom de fichier*>, et ils sont séparés par des / dans la liste,

nom de fichier est une suite de caractères, la seule restriction est que ce ne sont pas des /.

En voici, quelques exemples : `/vmunix`, `/etc/rc`, `/users/mbeig/COURS/chap07.tex`, `/usr/local/bin/tex`. On remarquera qu'avec ce système, les différents supports sont cachés à l'utilisateur : il voit l'ensemble des différents disques effectivement disponibles comme une seule hiérarchie.

7.8.2 Un fichier

Un fichier est identifié par son chemin d'accès et contient des données : une suite d'octets. Cependant d'autres informations sont associées à cette suite d'octets. Certaines intéressent l'utilisateur :

- la taille du fichier,
- les heures et dates de création, de dernier accès ou de dernière modification du contenu,
- le type du contenu (fichier texte, module exécutable, etc.)
- le propriétaire des données,
- des droits d'accès aux données, etc.

D'autres informations n'intéressent que le système :

- position sur le support,
- l'état instantanné : le fichier est-il ouvert, où en est la lecture, est-il en accès exclusif, etc.

Les systèmes proposent des services très différents à ce niveau. Généralement, plus l'ordinateur devra supporter des opérations à grandes échelles, plus le système de gestion de fichiers devra être performant et puissant dans la qualité des services fournis. Sur un ordinateur individuel et monotâche, il n'est pas besoin de conserver le nom du propriétaire, ni de gérer des droits d'accès. Sur un système multi-utilisateurs, cela est indispensable. Enfin, pour un système de réservation de

Adresse	Contenu	Type
+0x00	Instruction de saut à la routine de boot	3 octets
+0x03	Nom du fabricant et numéro de version	8 octets
+0x0B	Octets par secteur	1 mot
+0x0D	Secteurs par cluster	1 octet
+0x0E	Nombre de secteurs réservés	1 mot
+0x10	Nombre de FATs	1 octet
+0x11	Nombre d'entrées dans le répertoire racine	1mot
+0x13	Nombre de secteurs dans le volume	1mot
+0x14	Descripteur de support	1 octet
+0x15	Nombre de secteurs par FAT	1mot
+0x18	Secteurs par piste	1mot
+0x1A	Nombre de tête de lecture/écriture	1mot
+0x1C	Distance du premier secteur du volume au premier secteur du support de mémoire de masse	1mot
+0x1E-0x1FF	Routine de boot	482 octets

TAB. 7.2 – Structure du secteur de démarrage d'un volume

Taille de volume jusqu'à	128 Mo	256 Mo	512 Mo	1024 Mo	2048 Mo
Taille de cluster	2 ko	4 ko	8 ko	16 ko	32 ko
Secteurs par cluster	4	8	16	32	64

TAB. 7.3 –

billets d'avion connectés à des centaines de terminaux, il faut que chaque terminal puisse *locker* le fichier le temps de faire une réservation, ...

7.8.3 La structure des fichiers dans MS-DOS

Un *volume* est une succession de secteurs de 512 octets numérotés à partir de zéro. Ainsi un disque de 10Mo se compose de 20480 secteurs numérotés de 0 à 20479. C'est au pilote de gérer ces numéros de secteurs à sa guise.

7.8.3.1 Le secteur de démarrage (Boot sector)

La table 7.2 montre l'organisation du secteur de démarrage d'un volume. Ce secteur est toujours le premier secteur du volume physique. De cette façon, le pilote sait où le trouver. Outre les informations destinées au pilote lui-même, il contient le programme de démarrage qui permet d'amorcer le chargement du système d'exploitation ((en anglais : *Bootstrap loader*)).

7.8.3.2 La table d'allocation des fichiers (FAT)

L'allocation de place aux fichiers se fait par unité *cluster*. Un cluster est un groupe de secteurs dont la taille dépend de la taille du disque (voir la table 7.3).

La table d'allocation des fichiers ((en anglais : *FAT : File Allocation Table*)) indique quel est le statut des clusters du volume. La table 7.4 donne les valeurs ayant une signification spéciale. Toute autre valeur indique en fait que le cluster est occupé par un fichier et quel est le numéro du cluster suivant. Les premières versions de DOS date d'une époque où les disques étaient relativement de petite capacité. Trois octets par entrée de la FAT semblait alors suffisant, puisqu'ils permettaient d'utiliser 4096 clusters sur un volume. On parle alors d'une FAT 12 bits. L'évolution des capacités

Code cluster FAT 12 bits	Signification
0x000	Cluster libre
0xFF0–0xFF6	Cluster réservé
0xFF7	Cluster défectueux, inutilisé
0xFF8–0xFFF	Dernier cluster d'un fichier
Code cluster FAT 16 bits	**Signification**
0x0000	Cluster libre
0xFFF0–0xFFF6	Cluster réservé
0xFFF7	Cluster défectueux, inutilisé
0xFFF8–0xFFFF	Dernier cluster d'un fichier

TAB. 7.4 – Codification des entrées de la FAT

Adresse	Contenu	Type
+0x00	Nom du fichier (comblé avec des espaces)	8 octets
+0x08	Extension du fichier (comblé avec des espaces)	3 octets
+0x0B	Attribut du fichier	1 octet
+0x0C	Réservé	10 octets
+0x16	Heure de la dernière modification	1 mot
+0x18	Date de la dernière modification	1 mot
+0x1A	Numéro du premier cluster du fichier	1 mot
+0x1C	Taille du fichier (en octets)	1 double mot

TAB. 7.5 – Structure d'une entrée de répertoire

disques a obligé ensuite à utiliser quatre octets par entrée de la FAT, permettant ainsi 65536 clusters sur un volume, et on parle alors d'une FAT 16 bits.

La structure représentée dans la FAT est en fait une liste de cluster. En effet, l'entrée numéro i (FAT$[i]$) contient le numéro du secteur suivant (ou un des codes spéciaux indiqué dans le tableau 7.4). Ainsi, à partir d'un numéro de cluster, on peut connaître la liste des numéros de tous les clusters suivants jusqu'à rencontrer un code supérieur à `0xFF8` (ou à `0xFFF8`).

La FAT a une importance capitale dans l'utilisation d'un volume, aussi plusieurs copies en sont disponibles sur le support physique, de cette façon, si l'une d'entre elle est endommagée, il reste quelque espoir de pouvoir récupérer les informations stockées sur le volume.

7.8.3.3 Le répertoire principal et les sous-répertoires

Structure d'une entrée de répertoire La table 7.5 montre la structure d'une entrée de répertoire pouvant décrire un fichier ou un sous-répertoire. Le premier octet du nom du fichier peut prendre quelques valeurs particulières données dans la table 7.6.

Code	Signification
0x00	Dernière entrée du répertoire
0x05	Le premier caractère du nom de fichier est en fait 0xE5
0x2E	Fichier concernant le répertoire actuel. si un autre 0x2E suit, il se rapporte au répertoire père
0xE5	Le fichier a été supprimé

TAB. 7.6 – Signification spéciale du premier caractère du nom

Les attributs L'octet d'adresse 0x0B dans une entrée de répertoire est un champ de bits qui donne des informations sur le type du fichier :

bit 0 Read-Only, le fichier ne peut pas être modifié

bit 1 Fichier caché

bit 2 Fichier système

bit 3 Nom de volume

bit 4 Sous-répertoire

bit 5 Bit d'archivage

L'heure Le mot d'adresse 0x16 dans une entrée de répertoire donne l'heure de la dernière modification du fichier. Cette heure est codée sous la forme suivante :

bits 0–4 Secondes par unités de 2

bits 5–10 Minutes

bits 11–15 Heures

La date Le mot d'adresse 0x18 dans une entrée de répertoire donne la date de la dernière modification du fichier. Cette date est codée sous la forme suivante :

bits 0–4 Jour du mois

bits 5–8 Mois (1 à 12)

bits 9–15 Année (par rapport à 1980)

7.8.3.4 La zone de données

La zone de données se trouve après le répertoire racine du volume. Le numéro du premier secteur de cette zone est le résultat du calcul suivant :

$$n_r + n_{\text{FAT}} \times n_s + n_e \times \frac{32}{512}$$

où, n_r est le nombre de secteurs réservés (donné par le mot à l'adresse 0x0E du secteur de démarrage) ; n_{FAT} est le nombre de FAT (donné par l'octet à l'adresse 0x10 du secteur de démarrage) ; n_s est le nombre de secteurs par FAT (donné par le mot à l'adresse 0x16 du secteur de démarrage) ; n_e est le nombre d'entrées dans la racine (donné par le mot à l'adresse 0x11 du secteur de démarrage).

7.9 Exercices

Exercice 7.1 (Système de fichier) ***Question 7.1.1*** *Quelques systèmes d'exploitation fournissent l'appel système* RENAME *pour renommer un fichier. Y a-t-il la moindre différence entre cette attribution d'un nouveau nom et la copie du fichier sous un nouveau nom suivi de la suppression du fichier original ?*

Exercice 7.2 (FAT) ***Question 7.2.1*** *Voici un extrait d'une configuration de disque comportant une zone répertoire REP, elle-même contenant deux fichiers, ainsi que le morceau correspondant de la FAT. Un groupe d'allocation est égal à un secteur de 512 octets.*

1. *zone REP*

nom	*taille*	*numéro de début*
HEFER3	*1991*	*26*
CANALPLU.S	*2901*	*27*

2. *zone FAT*

Position	*Contenu*
26	*28*
27	*30*
28	*35*
29	FFF_{16}
30	*33*
31	FFF_{16}
32	*0*
33	*28*
34	*0*
35	*29*

Est-ce que cette configuration est cohérente ? Sinon, que peut-on faire ?

```
99 blocks of crud on the disk,
99 blocks of crud!
You patch a bug, and dump it again :
100 blocks of crud on the disk!

100 blocks of crud on the disk,
100 blocks of crud!
You patch a bug, and dump it again :
101 blocks of crud on the disk! ...
```

```
Due to lack of disk space, this fortune database has been
discontinued.
```