



Haute école d'ingénierie et d'architecture Fribourg
Hochschule für Technik und Architektur Freiburg

Systèmes d'exploitation

Systèmes de fichiers



- La RAM a une taille limitée.
- La RAM est non persistante.
- La RAM n'est pas partagée entre les processus.



Grande capacité de stockage : On doit pouvoir enregistrer une grande quantité d'information.

Persistance L'information doit être conservée après la fin du processus qui les utilise.

Mémoire partagée Plusieurs processus doivent pouvoir accéder simultanément à la même information.



Un disque peut être considéré comme une suite séquentielle de **blocs** de taille fixe. Le disque supporte deux opérations :

- Lire le bloc k .
- Ecrire le bloc k .



Voici les questions que l'on se pose avec des disques :

- Comment trouve-t-on l'information?
- Comment empêche-t-on un utilisateur de lire les données d'un autre utilisateur?
- Comment sait-on qu'un bloc est libre?





La partie du système d'exploitation qui gère les fichiers est appelée :

Système de fichiers (file system).



DOS, FAT12, FAT16	MS-DOS (1977)
FAT32	Windows 95 OSR 2 (1996)
NTFS	Windows NT 3.1 (1993)
UFS	Unix File System (1983)
EXT2, EXT3, EXT4	Linux (1992, 1993, 2006)
S5FS	System V File System (1969)
ZFS ¹	Open Solaris (2005)
Btrfs	Linux (2009 ²)
UBIFS	Unsorted Block Image Filesystem ³ (2008)
F2FS	Samsung, Flash-Friendly File System (2012)

¹Peut stocker 256 Zebibyte (2^{64} Byte ou 274'877'906'944 TByte)

²instable

³optimisé pour les mémoires FLASH



Chaque système de fichier définit des règles concernant les noms des fichiers :

- Caractères autorisés.
- Encodage des caractères (ISO/IEC 8859-1/Latin1, UTF-8, UTF-16, ...).
- Nombre maximum de caractères.
- Distinction entre les lettres minuscules et majuscules.



Le système FAT définit la convention «8.3» :

- 8 caractères pour le nom du fichier, 3 caractères pour l'extension.
- Encodage sur 8 bit.
- Les caractères suivants sont interdits : \ / : # ? " < >
- Les espaces sont autorisés.
- Pas de distinction entre les majuscules et les minuscules.



- NTFS autorise 255 caractères Unicode codés en UTF-16
- EXT autorise 255 bytes pour les noms de fichiers, mais ne définit pas l'encodage des caractères (ISO/8859 ou Unicode)



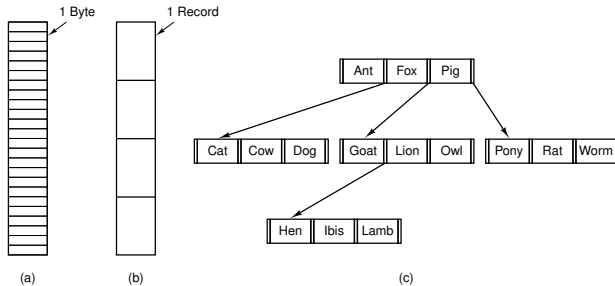
Définition (Wikipedia)

«En informatique, une **extension de nom de fichier** (ou simplement **extension de fichier**, voire **extension**) est un suffixe ajouté au nom d'un fichier pour identifier son format. Ainsi, on dira qu'un fichier nommé exemple.txt a l'extension txt ou .txt».

- Pour UNIX, l'extension est juste une convention et n'est pas imposée par le système.
- Dans UNIX encore, on peut avoir plusieurs extensions (par exemple : archive.tar.gz).
- Pour Windows, les extensions sont associées au programme qui peut traiter les fichiers correspondants.



Extension	Signification
fichier.bak	Fichier de sauvegarde
fichier.c	Fichier source d'un programme C
fichier.gif	Fichier image de format GIF (Graphical Interchange Format)
fichier.hlp	Fichier d'aide
fichier.html	Fichier document en langage HTML (HyperText Markup Language)
fichier.jpg	Fichier image de format standard JPEG
fichier.mp3	Fichier de musique codée en MPEG de niveau 3
fichier.mpg	Fichier de vidéo codée en MPEG
fichier.o	Fichier objet (source compilée, non encore liée)
fichier.pdf	Fichier document au format PDF (Portable Document File)
fichier.ps	Fichier document au format PostScript
fichier.tex	Fichier document au format TEX
fichier.txt	Fichier document au format texte
fichier.zip	Fichier archive compressé



Trois sortes de fichiers :

- (a) Suite d'octet (byte sequence).
- (b) Suite d'enregistrements (record sequence).
- (c) Arbre (tree).



- Fichiers ordinaires (regular files).
- Répertoires (directories)
- Fichiers spéciaux caractère (character special files) : terminal, imprimante, réseau.
- Fichiers spéciaux bloc (block special files) : disque.



```
$ find /dev/ -type c
```

```
/dev/midi           /dev/tty5
/dev/snd/controlC0  /dev/tty4
/dev/snd/seq        /dev/tty3
/dev/snd/timer      /dev/tty2
/dev/cpu/microcode  /dev/tty1
/dev/sg1            /dev/tty0
/dev/sg0            /dev/console
/dev/bus/usb/002/005 /dev/tty
/dev/net/tun        /dev/urandom
/dev/pts/0          /dev/random
/dev/rtc0           /dev/zero
/dev/input/mouse0   /dev/null
/dev/ttyS1          /dev/mem
/dev/ttyS0          ...
```




```
$ find /dev/ -type b | column -c 67 | expand
```

/dev/loop7	/dev/loop3	/dev/sr0	/dev/sda
/dev/loop6	/dev/loop2	/dev/sda5	/dev/fd0
/dev/loop5	/dev/loop1	/dev/sda2	
/dev/loop4	/dev/loop0	/dev/sda1	



- Fichiers «Textes»
- Fichiers «Binaires»

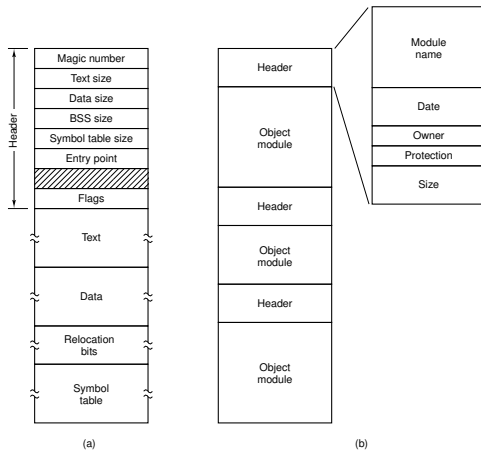


- Encodage des caractères (ASCII, ISO/IEC 8859-1/Latin1, UTF-8, UTF-16, ...)
- Conventions pour coder la fin d'une ligne (End Of Line – EOL).

Les deux caractères couramment utilisés pour les fins de lignes sont :

Carriage Return	CR	chr(13)	^M	\r	\x0D
Line Feed	LF	chr(10)	^J	\n	\x0A

Mac OS (≤ 9), Apple II	→	CR
UNIX, Mac OS X	→	LF
DOS, Windows	→	CR + LF



(a) Un fichier exécutable.

(b) Un fichier d'archive.



Fichier Jpeg	0xFF 0xD8
PDF	%PDF
DOS exécutable	0x4D 0x5A («MZ» en ASCII / Mark Zbikowski ⁴)
MS Office	0xD0 0xCF 0x11 0xE0 (D0CF11E0 / DocFile)
ZIP	0x50 0x4B («PK» en ASCII / Phil Katz ⁵)

Ces nombres magiques sont utilisés par les programmes pour identifier un fichier. C'est une des méthodes utilisées par le programme «file».

⁴Employé par Microsoft, concepteur du format de fichier exécutable DOS

⁵inventeur de du format ZIP en 1989



- L'accès séquentiel (sequential access) : Utilisé principalement pour les bandes magnétiques.
- L'accès direct ou aléatoire (random access) : Utilisé par les disques. Permet un déplacement à une position donnée (fonction seek).



En plus du nom du fichier, le système associe d'autres attributs au fichiers

Attributs	Signification
Protection	Qui peut accéder au fichier et de quelle manière
Mot de passe	Mot de passe nécessaire pour accéder au fichier
Créateur	Créateur du fichier
Propriétaire	Propriétaire actuel du fichier
Indicateur lecture seule	0 pour la lecture/écriture, 1 pour la lecture seule
Indicateur fichier caché	0 pour un fichier normal, 1 un pour fichier caché
Indicateur fichier système	0 pour un fichier normal, 1 pour un fichier système
Indicateur d'archivage	0 si le fichier a été archivé, 1 s'il doit être archivé
Indicateur fichier ASCII/binaire	0 pour un fichier ASCII, 1 pour un fichier binaire
Indicateur fichier accès aléatoire	0 pour un accès séquentiel, 1 pour un accès aléatoire
Indicateur fichier temporaire	0 pour un fichier normal, 1 pour supprimer le fichier lorsque le processus se termine
Indicateur de verrouillage	0 pour un fichier non verrouillé, 1 pour un fichier verrouillé
Longueur d'enregistrement	Nombre d'octets dans l'enregistrement
Position de la clé	Position de la clé dans chaque enregistrement
Longueur de la clé	Nombre d'octets du champ clé
Date de création	Date et heure de création du fichier
Date du dernier accès	Date et heure du dernier accès au fichier
Date de modification	Date et heure de la dernière modification
Taille courante	Nombre d'octets du fichier
Taille maximale	Taille maximale autorisée pour le fichier



Principales opérations (appels système) relatives aux fichiers :

- Create
- Delete
- Open
- Close
- Read
- Write
- Append
- Seek
- Get attributes
- Set attributes
- Rename



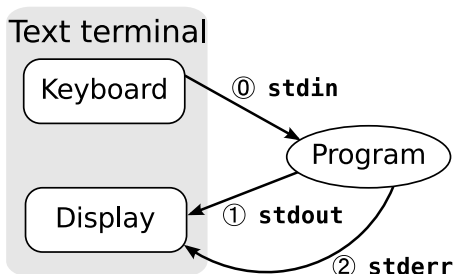
```
copy (source, destination)
```

```
sf = open(source)           # sf is a file descriptor
df = create(destination)    # df is a file descriptor
buffer_size = 4096
buffer = array[buffer_size] of byte
while true:
    count = read(sf, buffer, buffer_size)
    if count <= 0:
        break
    write(df, buffer, count)
close(sf)
close(df)
```



Définition (Wikipedia)

En informatique, un **descripteur de fichier** est une clé abstraite pour accéder à un fichier (c'est un entier). On utilise généralement ce terme pour les systèmes d'exploitation POSIX.





```
open("/proc/filesystems", O_RDONLY)      = 3
read(3, "nodev\tsysfs\nnodev\trootfs\nnodev\tb"... , 1024) = 337
read(3, "", 1024)                        = 0
close(3)                                = 0
geteuid()                               = 1000
stat("/tmp/", {st_mode=S_IFDIR|S_ISVTX|0777, ...}) = 0
stat("/etc/issue", {st_mode=S_IFREG|0644, st_size=26, ...}) = 0
stat("/tmp/issue", 0x7fffffff1ded30)     = -1 ENOENT
open("/etc/issue", O_RDONLY)             = 3
open("/tmp/issue", O_WRONLY|O_CREAT|O_EXCL, 0644) = 4
read(3, "Debian GNU/Linux 7 \n \l\n\n", 32768) = 26
write(4, "Debian GNU/Linux 7 \n \l\n\n", 26) = 26
read(3, "", 32768)                       = 0
close(4)                                 = 0
close(3)                                 = 0
lseek(0, 0, SEEK_CUR)                    = -1 ESPIPE (Illegal seek)
close(0)                                 = 0
close(1)                                 = 0
close(2)                                 = 0
```



```
open("/var/log/syslog", O_RDONLY)          = 3
lseek(3, 0, SEEK_CUR)                      = 0
lseek(3, 0, SEEK_END)                     = 185264
lseek(3, 180224, SEEK_SET)                 = 180224 /* 176*1024 */
read(3, "bian dhclient: All rights reserv"... , 5040) = 5040
write(1, "Nov 17 22:35:01 debian /USR/SBIN"... , 110) = 110
write(1, "Nov 17 22:39:42 debian dhclient:"... , 78) = 78
write(1, "Nov 17 22:39:42 debian dhclient:"... , 60) = 60
write(1, "Nov 17 22:39:42 debian dhclient:"... , 83) = 83
write(1, "Nov 17 22:43:04 debian mpt-statu"... , 69) = 69
write(1, "Nov 17 22:45:01 debian /USR/SBIN"... , 110) = 110
write(1, "Nov 17 22:52:34 debian dhclient:"... , 78) = 78
write(1, "Nov 17 22:52:34 debian dhclient:"... , 60) = 60
write(1, "Nov 17 22:52:34 debian dhclient:"... , 83) = 83
write(1, "Nov 17 22:53:04 debian mpt-statu"... , 69) = 69
read(3, "", 0)                            = 0
close(3)                                  = 0
close(1)                                  = 0
close(2)                                  = 0
```