Chapter VII: File management system

I/ Definition:

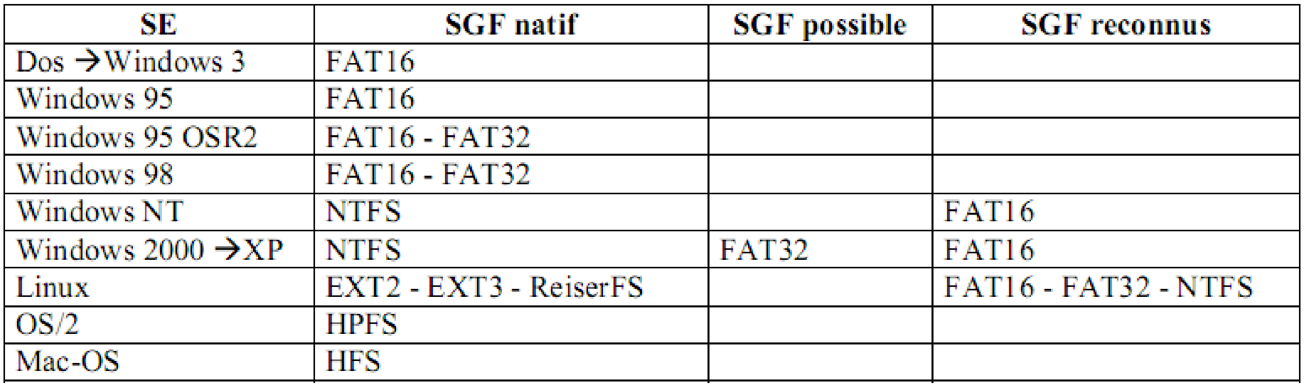
A file system is the collection of information stored in secondary memory, and a file management system is the set of programs that organize and manage files, providing an interface between the user and the lower layers of the system.

It also ensures correspondence between file names and their physical location, and includes MSDOS (FAT16, FAT32) and Windows NTFS file system managers. They are based on cluster management (allocation units).

II/ SGF functions:

- Provide users with the ability to perform operations on file system components (files/directories).
- Keep track of information on all stored files.
- Define a file access method.
- Define a method for allocating and releasing disk space.

III/ SGF types:



SE	SGF natif	SGF possible	SGF reconnus
Dos →Windows 3	FAT16		
Windows 95	FAT16		
Windows 95 OSR2	FAT16 - FAT32		
Windows 98	FAT16 - FAT32		
Windows NT	NTFS		FAT16
Windows 2000 →XP	NTFS	FAT32	FAT16
Linux	EXT2 - EXT3 - ReiserFS		FAT16 - FAT32 - NTFS
OS/2	HPFS		
Mac-OS	HFS		

IV/ Files management:

1/ *Definition* :

A file is a set of structured data of the same kind stored on a disk. Each file has, in addition to its name, an extension designating the type of data represented.

2/ *File properties* :

- An external name assigned by the user.
- A type (program, text, document, image, audio, archive, source code, etc.).

- Current size in bytes, words or blocks.
- Date of creation, modification and last access.
- Password (if any).
- Attributes (Archive, Read-only, Hidden or System).

These properties are stored in the structure of the directory where the file resides.

3/ *File operations* :

A file is manipulated by the user through operations managed by the FMS:

- Creation: where the system allocates disk space and creates a new entry in the destination directory.
- Opening and closing: opening consists in informing the system that the file is active and will be inserted into the table of open files; closing consists in deleting its entry in this table.
- Read and write: a system call is executed to read/write to a file in a consistent manner.
- Deletion.
- And more: rename, copy, update, change properties, etc.

4/ File access methods :

- Sequential access: consists of accessing and processing information in a file in the order in which it is stored, in a sequential manner. A pointer is used for this purpose. (Example: a compiler accesses a source code file sequentially).
- Direct access: to access information directly at any point in the blocks.
- Other, ...

V/ Folders management:

1/ *Definition* :

Is a file system object, containing the names of other objects. It can be seen as an array that stores information about files (properties).

Example: entry in an MSDOS directory :

File name	Extension	Attributes	Reserved	Hour	Date	N° of 1st bloc	Size

2/ *Folders structure* :

- Single-level directory: a single directory containing files only.
- Two-level directory: a Master File Directory (MFD) contains the User File Directory (UFD) for improved management between multiple users.
- Tree directory: a directory contains sub-directories which may contain files or other directories.
- Acyclic graph directory: a graph in a cycle where one directory can be contained in two directories.

3/ ***The paths*** **:**

When directories are organized in a tree structure, it is necessary to name files according to the path of the directories through which they are accessed.

4/ ***Folder operations*** **:**

- Create a new directory.
- Delete a directory.
- Open and read directory contents.
- Close a directory.
- Rename a directory.
- .....

5/ ***Folder implementation*** **:**

- Linear list: a list containing file names and pointers to data blocks.
- Hash table: a linear list containing directory entries and a hash table data structure is used which takes a value calculated from the file name and returns a pointer to the file entry in the list.

VI/ Disk space management:

1/ ***Space allocation*** **:**

Three allocation methods are used:

- Contiguous allocation: the file occupies a set of contiguous blocks on the disk. Two pieces of information are required for access: the address of the 1st block and the file size.
- Chained allocation: chaining blocks allocated to a file without them necessarily being contiguous (using a chained list).
- Indexed allocation: use its own index for each file, in the form of an array of block addresses grouping together all the pointers to the blocks making up the file.

2/ **Free *space management*** **:**

FMS needs to know how much free space is available on disk in order to allocate space to files. It also needs to reuse space from deleted files for new files.

VII/ File protection and security:

You can use :

- ✓ Access type limitation: for reading, writing, executing, modifying, ...
- ✓ Use of file passwords.

VIII/ Windows file management systems:

Windows Vista supports several file systems, the most important of which are FAT16, FAT32 (File Allocation Table) and NTFS (NT File System). FAT16 is the old MS-DOS file system, using 16-bit addresses, which limits partition size to 2 GB, and is mainly used to access floppy disks. FAT32 uses 32-bit hard disk addresses (actually 28 bits) and accepts partition sizes up to 2 TO (tira), it is weak on security and today is used for removable media.

NTFS is the file system developed specifically for the NT version of Windows. Since Windows XP, it has become the default file system installed by most computer manufacturers. NTFS uses 64-bit disk addresses and can theoretically handle partitions of up to 64 bits.

1/ ***NTFS system features:***

- File names are limited to 255 characters.
- Complete paths must not exceed 32,767 characters.
- File names are in Unicode, to allow the use of non-Latin alphabets (Greek, Japanese, Indian, etc.).
- NTFS is case-sensitive (fich is a different file from FICH or Fich).
- An NTFS file is not a simple sequence of bytes, as in the case of FAT32 and UNIX, but is made up of numerous attributes, each of which is represented by a byte stream. Most files have only a few short bytes, such as the file name and ID (64-bit), and a longer byte stream for data. Each stream has a name made up of the file name, a ":" and the short name of the stream, e.g. myfile:stream1. Each stream has its own size and can be placed independently of the other streams in the file.
- NTFS is a hierarchical file system, similar to the UNIX file system.
- The separator between component names is "\".
- The concept of current directory, physical links to the current directory (.) and parent directory (..) are implemented as conventions, not as a fundamental part of NTFS.
- Compression, encryption, fault tolerance, logging and hollow files are also supported.

2/ ***NTFS system structure:***

- Each NTFS volume (Partition) contains files, directories, bitmaps and other data structures. Each volume is organized as a sequence of blocks (or clusters), with a fixed cluster size for each volume (ranging from 512 bytes to 64 KB); most NTFS disks have a block size of 4 KB.
- The main data structure of each volume is the MFT (Master File Table). This is a sequence of 1-KB records.
- Each record describes a file or directory.
- The record contains file attributes, such as name and access dates, and a list of the disk addresses of its blocks.

The MFT is also a file, and can be placed anywhere in the volume, growing up to 2^{48} records.