1. A file system is a method for storing and organizing computer files and the data they contain to make it easy to find and access them. It organizes data into files and directories and tracks where they are located on disk.
2. File attributes include:

* Name - The name of the file
* Identifier - A unique identifier for the file like an inode number
* Type - Such as regular file, directory, symbolic link, etc.
* Location - Where the file is stored on disk
* Size - The size of the file in bytes
* Protection - Access permissions like read/write/execute
* Time - Timestamps like creation, modified and access time
* Date - The date the file was created or last modified
* Permissions - Controls who can access the file and what operations they can perform

1. On Linux:

* touch - Creates a new empty file
* rm - Deletes a file
* mkdir - Creates a new directory
* rmdir - Removes an empty directory
* cd - Changes to a directory
* ls - Lists directory contents

On Windows:

* copy - Copies files
* del - Deletes files
* md - Makes a directory
* rd - Removes a directory
* cd - Changes directory
* dir - Lists directory contents
* start - Opens a file

1. Sequential access reads the file from start to end, like reading a book. Random access can jump around to access any part of the file directly.
2. An absolute path starts from the root directory like /home/user/file. A relative path starts from current directory like ../file or file.
3. A hard link points directly to the data blocks on disk. Deleting the original file does not delete the data as long as hard links point to it. A soft link is a pointer to another path. If the original is deleted, the soft link breaks.
4. Block special files represent devices and allow software to interact with hardware devices. Character special files represent data streams and allow reading/writing data.
5. With 10 byte records and 25 byte blocks, there would be 15 bytes wasted and unused in each block. This is internal fragmentation within each block.
6. With variable length records, fragmentation can range from 0 bytes (with a 30 byte record) up to 15 bytes (with a 10 byte record). This is internal fragmentation.
7. Linked list allocation uses pointers in each block to link to the next free block in the list. This allows finding free blocks in O(1) time but requires traversing the list to find a large enough free space.
8. Indexed allocation maintains an index table of free blocks. This allows finding free blocks and free space in O(1) time but requires updating the index when blocks are allocated/freed.
9. The indirect block can hold 256 disk addresses. Together with the 10 direct disk addresses, the maximum file has 266 blocks. Since each block is 1 KB, the largest file is 266 KB