

# 1. Introduction

Q.4.3.1.1 What do mean by file structure?

Answer: File structure refers to the organization of data kept in the secondary storage device in such a way that it minimizes both the data access time and the storage space. A file structure is a combination of representations for data in files and of operations for accessing the data. It allows application programs to search, read, write and modify data in the secondary storage device efficiently and conveniently.

Q.4.3.1.2 Compare and contrast between the ideas of data structure and file structure.

Answer: The fundamental difference is that data structures deal with data kept in main memory, and file structures deal with data stored in secondary storage.

Both involve in representing data and accessing data. But, data accessing from a file that is often kept in the secondary memory is not direct. A file may contain a large volume of data. Often data seaching is fecilitated by an index structure. During program execution time, an index struicture may be kept in the main memory. Such index structure is a kind of a data structure that helps in searching data in the secondary storage.

Both data structure and file structure are created based on an application. For a particular application, data may be kept in main memory using a max heap data structure for finding maximum element repetitively among the remaining available data. A file may be organised as a collection of records / objects. A file structure is a combination of representations for data in files and of operations for accessing the data.

Q.4.3.1.2 Fill in the gap.

Disks are — as compared to memory.

(i) slow, (ii) very slow, (iii) fast, (iv) neither fast nor slow

Answer: (ii)

Q.4.3.1.3 Identify the difference between physical file and logical file.

Answer: A few points of comparison are given below.

▷ Physical files contain the actual data that are stored on the system. Also, it may contain a description of how data are to be presented to

or received using a program. Logical files do not contain data. They contain a description of records found in one or more physical files.

▷ Physical files contain only one record format, and one or more records.

A logical file is a view or representation of one or more physical files.

▷ Records in database files can be described externally in a program.

▷ A physical file is associated with a unique access path.

▷ Logical files that contain more than one format are referred to as multi-format logical files. In that case, one can use a read by record format to set the format the user wish to use.

Q.4.3.1.4 State some desirable properties of a file.

Answer: Some properties of a file are given below.

- Files are stored on a secondary storage and remain there even after a user logs off the computing system.
- A file can be sharable with the other users possessing required access permissions.
- Files can be organized into a structure to reflect the relationships among them. A relationship could be a hierarchical structure, relational structure, or some other type of relationship.

Q.4.3.1.5 What is file system?

Answer: A file system is a software that controls how data is stored and retrieved. With the absence of a file system, data placed in a storage medium would be one large body of data with no way to tell where one piece of data stops and the next begins. It is meaningful to separate data into pieces and giving each piece a name, so that the data is easily isolated and identified. Each group of data is called a file. The structure and logic rules used to manage the groups of data and their names is called a file system.

Q.4.3.1.6 What are the objectives of file system?

Answer: Objectives of file management system include the following:

- \* There are various data processing operations such as reading, writing and updation. It meets the data management needs of users.
- \* It optimizes performance of data processing operations.
- \* It provides I/O support for different storage devices.
- \* It provides standard I/O interface routines to user processes.
- \* It provides I/O support for multiple users in a multi-user system.

Q.4.3.1.7 What is inode number?

Answer: Apart from file name, each file in a Unix-based file system has an identification number, called an inode number. It is unique in its file system. The inode number refers to the physical file, in which data stored is kept in a particular location. A file also has a device number. The combination of file's inode number and device number is unique in the file system, maintained in a hierarchical manner.

Q.4.3.1.8 What happens, when you open a file?

Answer: When you have the relevant permissions, you can open a file. When you open a file, a file descriptor is created using the unique inode number associated with file name. As many processes / applications can point to the same file, inode has a link field that maintains the total count of links to the file.

Q.4.3.1.9 Explain the following terms: memory access time, disk access time.

Answer: Memory access time is the time it takes for a character to be transferred to or from the CPU. Fast RAM chips have an access time of 10 ns or less.

Disk access time is how long it takes to obtain the first data character after initiating a request. It includes the time to move the read / write head to the track (seek time) and time to rotate the platter to the sector (latency). Disk access time is always given as an average, because seek time and latency vary depending on the current position of the head and platter. While access times of fast hard disks are typically from 5 to 10 ms, solid state drive (SSD) access times are in the range 25 - 100  $\mu$ s.

Q.4.3.1.10 Compare the following storage types: primary storage, secondary storage, tertiary storage.

Answer: Primary storage is a key component of a computer system that enables it to function. Primary storage includes random access memory (RAM), read only memory (ROM) and cache memory. They are the fastest storage media.

Secondary storage is non-volatile type of memory having long-term storage capacity. There are three main types of secondary storage: solid state storage devices, such as USB memory sticks; optical storage devices, such as CD, DVD and Blu-ray discs; magnetic storage devices, such as hard disk drives. They are known as online storage.

Tertiary storage devices have high capacity. Some example are magnetic tape and optical disc. The removable media are normally not stored in

## 2. File Processing Operations

Q.4.3.2.1 Differentiate between physical address and logical address.

Answer: The concepts of logical and physical addresses are elaborated below.

*Logical address:* It is generated by CPU during program execution. A logical address is also called virtual address. This address is used as a reference to access the physical memory location by CPU. Logical address space is the set of all logical addresses generated by a program. The hardware device called memory management unit (MMU) is used for mapping a logical address to its corresponding physical address. Address translation technique is shown in Fig. 1.3.

*Physical address:* It identifies a physical location of required data in a memory. A user never directly deals with a physical address, but can access by its corresponding logical address. The user program generates logical addresses. A logical address must be mapped to a physical address by MMU. Physical address space is the set of all physical addresses corresponding to the logical addresses in the logical address space of a program.

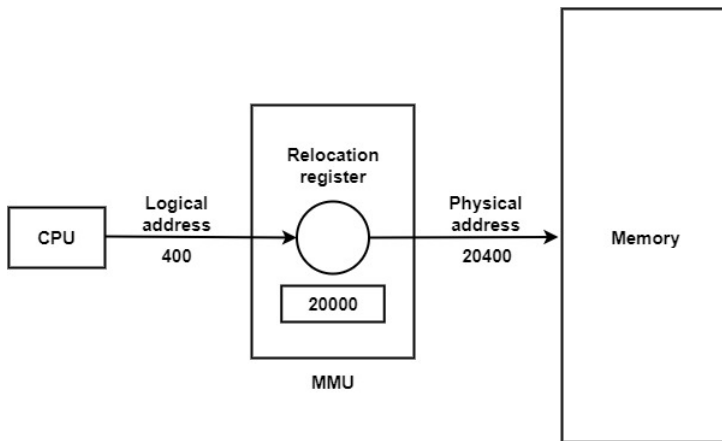


Figure 1.3: Address translation from logical to physical

A few contrasting points are given below:

\* A logical is generated by the CPU, but the corresponding physical address is computed by the MMU.

\* A user can view the logical address of a program, but the corresponding physical address cannot be viewed.

Q.4.3.2.2 Discuss the ideas of reading and writing files at a low level.

Answer: A general format of a low-level read instruction is given below:

*read (sourceFile, destinationAddr, size)*

*sourceFile* specifies the name of file that has been opened earlier in the read mode. *destinationAddr* is the address of memory location, where the data gets stored after reading from file. *size* specifies the amount of data (in bytes) is to be read.

A general format of low-level write function is given below:

*write (destinationFile, sourceAddr, size)*

*destinationFile* specifies the name of file that has been opened earlier in the write mode. *sourceAddr* is the address of memory location, where the data is available for writing into the file. *size* specifies the amount of data (in bytes) is to be written.

The syntaxes of *read()* and *write()* operations vary from a programming language to another programming language.

Q.4.3.2.3 Discuss some methods of organizing records in a file.

Answer: A few methods of organizing records in a file are given below.

*Fixed-length records:* Each record contains the same number of bytes.

When a record contains a fixed number of fields and each field is of fixed-size, the record becomes of fixed-length. This is a commonly used method of organizing a file.

*Records with predictable number of fields:* In some applications, it is useful to fix the number of fields in a record. Fields of a record are placed physically in a consecutive order.

*Record with length indicator:* One can specify the length of a record at the beginning. It tells the number of bytes in rest of the record.

*Index to keep track of addresses:* One can maintain byte offset for each record in the original file. Byte offset helps finding the beginning of each record and length of each record.

*Delimiter at the end of each record:* We often use a character as record delimiter.

Q.4.3.2.4 Write a C program to find number of lines, tabs, blanks and characters in a file.

Answer: First, we read a file from the user, and we open in reading mode. We read one character in the file at a time, and increase different

counters based on the character type. We repeat reading a character until we reach the end of the file. Finally, we print different statistics.

```
#include <stdio.h>
#define TRUE 1
int main() {
    FILE *fp;
    char ch, fname[20];
    int nl, nt, nb, nc;
    nl = nt = nb = nc = 0;
    printf("Read a file: "); scanf("%s", fname);
    fp = fopen (fname, "r");
    while (TRUE) {
        ch = fgetc (fp);
        if (ch == EOF) break;
        nc++ ;
        if (ch == ' ') nb++;
        if (ch == '\n') nl++;
        if (ch == '\t') nt++ ;
    }
    fclose (fp);
    printf ("\nNumber of characters: %d", nc);
    printf ("\nNumber of blanks: %d", nb);
    printf ("\nNumber of tabs: %d", nt);
    printf ("\nNumber of lines: %d", nl);
    return 0;
}
```

A sample output of the program is given below:

```
Read a file: main.c
```

```
Number of characters: 941
Number of blanks: 227
Number of tabs: 0
Number of lines: 32
```

Q.4.3.2.5 What is data buffer? State a few applications of buffering.

Answer: A data buffer, or simply buffer, is a region of a physical memory storage used to temporarily store data, while it is being moved from one place to another. A buffer may be used when moving data between processes within a computer. Buffers can increase application performance

### 3. Secondary Storage

Q.4.3.3.1 Make a comparison between primary memory and secondary memory.

Answer: We have given here some contrasting points between these two types of memory device.

- Data and instructions are stored in the main memory and are accessed directly by the CPU during program execution time. There are two types of primary (main) memory: random-access memory (RAM) and read-only memory (ROM). Secondary memory is not directly accessed by the CPU.

- Primary memory devices are more expensive than secondary storage devices.

- ROM is non-volatile, but RAM is volatile. Secondary memory is non-volatile.

- Examples of primary memory are RAM, ROM, cache memory and CPU registers. Some examples of secondary memory are hard disk, floppy disk, magnetic tape, optical disk and pen drive.

Q.4.3.3.2 Give relationships between different storage units.

Answer: 1 byte =  $2^3$  bits; 1 kilobyte (KB) =  $2^{10}$  bytes

1 megabyte (MB) =  $2^{10}$  KB =  $2^{20}$  bytes

1 gigabyte (GB) =  $2^{10}$  MB =  $2^{20}$  KB =  $2^{30}$  bytes

1 terabyte (TB) =  $2^{10}$  GB =  $2^{20}$  MB =  $2^{30}$  KB =  $2^{40}$  bytes

$2^{10}$  terabytes = 1 petabyte

$2^{10}$  petabytes = 1 exabyte

$2^{10}$  exabytes = 1 zettabyte

$2^{10}$  zettabytes = 1 yottabyte

$2^{10}$  yottabytes = 1 brontobyte

$2^{10}$  brontobytes = 1 geopbyte

Sometime we approximate  $2^{10}$  by  $10^3$ .

Thus, 1 TB  $\approx 10^3$  GB; 1 GB  $\approx 10^3$  MB; 1 MB  $\approx 10^3$  KB.

Q.4.3.3.3 Give examples of some cheap secondary memory devices.

Answer: The least expensive secondary memory is magnetic tape. The next expensive secondary memory is CD-ROM. They are known as mass storage devices.

Q.4.3.3.4 Write a note on flash memory.

Answer: Flash memory is an electronic non-volatile computer memory storage medium that can be electrically erased and reprogrammed. There are two main types of flash memory: NOR flash and NAND flash, are named after the NAND and NOR logic gates.

The NAND type is found primarily in memory cards, USB flash drives, solid-state drives (those produced in 2009 or later), feature phones, smartphones and similar products, for general storage and transfer of data. NAND or NOR flash memory is also often used to store configuration data in numerous digital products, a task previously made possible by EEPROM or battery-powered static RAM.

Most NAND flash drives use a process called program / erase to store data. Unless the number of new writes is reduced, this process eventually wears and tears down the flash drives. As a result, flash memories do not have the capability to hold up heavy write loads. Another drawback of flash memory is that it rewrites data in block units. If it is done one by one for each byte, it can take considerably long amount of time. Due to this, the editing time has been cut down significantly. However, this process does not ensure that each block can be edited specifically.

Flash memory has been used in computers, PDAs, digital audio players, digital cameras, mobile phones, synthesizers, video games, scientific instrumentation, industrial robotics, medical electronics, and other gadgets. In addition to being non-volatile, flash memory offers fast read access time, although not as fast as static RAM or ROM.

Q.4.3.3.5 What does WORM refer to?

Answer: Write-once, read-many (WORM) refers to a recordable or data storage device, where information once written cannot be modified or overwritten. It is of the form of an optical storage. It allows data to be written once and read any number of times without the possibility of erasing. The CD-R and DVD-R optical disks are common WORM devices.

Q.4.3.3.6 Explain the terms:

i. seek time, ii. rotational latency, iii. access time, iv. disk bandwidth, v. cluster, vi. extent, vii. blocking factor.

Answer: i. The seek time measures the amount of time required for the read / write head of a hard disk to move between tracks over the surfaces of the platter. Seek time is normally expressed in milliseconds (ms). The average seek time of most modern drives today lies between 8 - 10 ms



for common hard disks.

- ii. Rotational latency is the delay due to waiting for the rotation of the disk to bring the required disk sector under the read / write head. It depends on the rotational speed of a disk, measured in revolutions per minute (RPM). It is expressed in milliseconds, and the average is equal to the one-half of the rotational period.
- iii. Average disk access time = Average seek time + Average rotational latency + Transfer time + Controller overhead + Queue overhead
- iv. Disk bandwidth = Number of bytes transferred / (Time of completion of the last transfer - Time of the first request for service)
- v. A cluster is formed using a fixed number of contiguous sectors.
- vi. An extent is a group of contiguous clusters.
- vii. Blocking factor = Number of records stored in block of a file

Q.4.3.3.7 Name a purely sequential access device.

Answer: Magnetic tape provides only sequential access of data.

Q.4.3.3.8 Explain the concept of optical juke box memory.

Answer: It is an array of CD-ROM platters, which are loaded onto drives on demand. Although optical juke boxes have capacities in the hundreds of gigabytes, their retrieval times are in the hundreds of milliseconds. They are quite a bit slower than magnetic disks.

Q.4.3.3.9 Consider the following specification of a conventional hard disk.

Number of surfaces = 20

Number of tracks/surfaces = 400

Number of sectors/track = 60

Number of bytes/sector = 180

What is storage capacity of the hard disk?

Answer: Total capacity =  $(20 \times 400 \times 60 \times 180)$  bytes = 86.4 MB, assuming that 1 MB =  $10^6$  bytes.

Q.4.3.3.10 Discuss some physical characteristics of magnetic disks.

Answer: Some physical characteristics are given below:

- i. Head motion
  - Fixed head (one per track)
  - Movable head (one per surface)
- ii. Platters
  - Single platter
  - Multiple platter