

# **File & Input and Output Systems**



**Content:-**

1. **Attribute Of File**
2. **Operation On Files**
3. **File Accessing Mechanism**
4. **File Allocation Method**
5. **System Call**

**File :-** It is an abstract data type.

**Attributes of files are as follows :-**

- a. **Name :-** The Symbolic file name is the only information that is kept in human readable form.
- b. **Identifier :-** This is unique tag that is represented by a number which identifies the file within the file system. It is in form of non-human readable for the file
- c. **Type :-** This is the information about the file which is required for the system having different types of files.
- d. **Location :-** This information is a pointer to a device and to location of the file on that device
- e. **Size :-** The current size of file(in bytes, words or block), and possibly the maximum allowed sizes are included in this attribute.
- f. **Time, Date and User Identification :-** This information may be kept for creation , last modification and last usage. These data can be useful for the protection and security.

**Operations Performed on Files:-** As file is an abstract data type .To define a file property , we need to define various operations that can be performed on files. Operations on files includes:

1. **Creating a File :-** To create a file , two operation is necessary :



- a. Space in the file system must be found for file.
- b. An entry for the new file must be made in the directory. The directory entry records the name of the file and the location in the file system and some other information.

2. **Writing a file:-** To write a file, we make a system call specifying both the name of the file and the information to be written to the file. For the given file name, system searches the directory to find the location of file. The system must keep a write pointer to the location in file where the next write is to take place. Updating of write pointer has been done whenever the write occurs.
3. **Reading a File:-** To read from file, a system call specifies the file name and when the next block of the file should be put in the memory. Again, the directory is search for the associated directory entry and the system needs to keep a read pointer to the location in the file where the next read is to take place.
4. **Repositioning File:-** Repositioning does not involve any actual I/O operation. This operation is also known as file seek.
5. **Deleting a File:-** To delete a file, we search the directory for the named file. Having formed the associated directory entry. We release all file space and invalidate the directory entry.
6. **Truncating a File:-** The user may want to erase the content of a file but keep its attributes. This field allows all attributes to remain unchanged except file length, rather than forcing user to delete the file.

**File Accessing Mechanism:-** There are basically three mechanism for accessing the file :

1. **Sequential Access**
2. **Direct Access**



### 3. Index Access

1. **Sequential Access:-**It is the simplest access mechanism, in which information stored in a file are accessed in an order such that one record is processed after the other. For example , editors and compiler usually access files in this manner.

The main operations that we perform on a file are read and write . A read operation reads the next portion of the file and automatically advance a file pointer , which tracks the I/O location . A write operation appends the end of the file and advances to the end of newly written material. This method is based on the tape model of a file works well sequential access device .

Photo ---- 1

2. **Direction Access:-** It is an alternative method for accessing file , which is based on the disk model of a file , since disks allow random access to any block or record of a file .A file is viewed as a numbered sequence of blocks or records which are read/written in an arbitrary manner i.e. there is no restriction on the order of reading or writing.

Direct access mechanism is well suited for database management system in which we have to access a large amount of information. When a query arrives then we compute the block containing the answer and then read the block directly which provides the desired information. File operation in the method uses block number as a parameter which is provided by the user to the OS. Block numbers are given w.r.t. the beginning position of the file .



- 3. Indexed Access:-**In this method an index is created which contains a key filed and pointers to the various blocks. For finding an entry in the file for a key value, first the index is searched and then use the pointer to directly access a file and find the desired entry.

**File Allocation Methods:-**The space allocation strategy in memory is often closely related to the efficiency of file accessing. In many cases, many files can be stored on a particular disk. For a good space allocation strategy to take place various related and interactive factors are considered such as:

- i. Processing speed of sequential access to files, random access to files, allocation and de allocation of blocks.
- ii. The ability to make use of multi-sector and multi-track transfer.
- iii. Disk space utilization
- iv. Main memory requirement

The major methods of allocating disk space are in widely used :-

1. Contiguous allocation
2. Linked allocation
3. Indexed allocation

**Contiguous Allocation :-** The contiguous allocation method requires each file to occupy a set of contiguous address on disk. Linear ordering of disk is defined as disk address.

Accessing a file which has been contiguous allocated is easy. For sequential access, file system remembers disk address of the last block and when requirement , reads the next block. Thus, both direct access and sequential access can be supported by contiguous allocation.



**Dynamic Storage Allocation (An Application of Contiguous Allocation):-** Disk can be viewed as a large array of block. At any given time, Some of these blocks are allocated to files and others are free. Disk space can be seen as a collection of free and used segments in which each segment is a contiguous set of disk blocks. A free segment(or unallocated segment) is called a hole. The dynamic storage allocation problem is how to satisfy a request of n- size from a list of free holes. There are many solution this problem . The set of hole is searched to determine which hole is best to allocate. First-fit, Best-fit and worst-fit are the most common strategies used to select a free hole from the set of various holes.

**First-fit :**

Allocate the first hole that is big enough. Here searching can be started either at the beginning of the set of holes or where the previous first-fit search ended(Next-fit). We stop searching when we find large enough free space for free hole.

**Best-fit :**

Allocation of the smallest hole that is big enough .Searching of entire list is done unless the list is kept ordered by size. This strategy produces the smallest left-over hole.

**Worst-fit:**

Allocate the largest hole . Again we have search the entire list unless it is kept in a sorted order by the size. This strategy produces the largest left-over hole, which may be more useful than the smallest-left over hole obtained in best-fit approach.

These algorithms suffer from external fragmentation simulations have shown that first fit and best fit strategies are better than worst fit in terms of storage utilization.



**Linked Allocation** : There is no external fragmentation with linked allocation. Any free block on the free space list can be used to satisfy a request since all blocks are linked together. The major problem with linked allocation is that it can be used only sequential-access files.

Another Problem of linked association is reliability. Since the files are linked together by pointer scattered all over the disk. What will happen if a pointer is lost or damaged.

**Indexed Allocation** :-In indexed allocation all pointer are brought together into one location called Index Allocation. Direct access is supported by indexed allocation without suffering from external fragmentation because any free block on the disk may satisfy a request for more space .

### **System Calls:-**

To write any user program, we have to go to user mode & to access any functionality of system we have to go to kernel mode.

We are using user mode and to go to kernel mode system call is used. System call is a way to shift from user mode to kernel mode i.e. accessing functionality of systems.

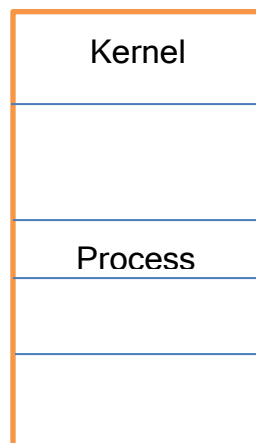
**Example:-** To access a file or resource we have to do it with the help of kernel i.e. with the help of OS which is done by system calls .

**Example:-** To print something , printer is to be accessed with the help of system call in kernel (OS) mode.

- Main logical unit in OS is kernel .
- OS has hundreds of system call. Some OS give facility to write system calls directly but some OS don't let us do this there we have local libraries like printf (to print everything ,scanf - they are not the system calls , they are functions that access system calls)

## Categories of System Calls :-

- 1. File Related System Calls :-** We have a C program & when we are executing it then it becomes a process & comes to RAM for execution & kernel (part of kernel) is also residing in RAM. Now, if process wants to access a file, then process has to ask kernel to give access of the file. Process does not have privilege of that file, so kernel will generate system call to access file.



RAM

### File Related System Calls :-

Open(), Read(), Write(), Close(), Create file etc .

- 2. Device Related :-** If we want to access a device/hardware ex- monitor, printer. User can't access directly, it has to take privilege from kernel through system call.

**Device Related System Calls:-** Read, Write, Repositioning

- 3. Information Related System calls:-** Process/device attribute information is accessed by this.



**Ex-**

**getpid** - to know id of a process

**getppid** - to know id of parents

To access information about process also known as meta data (data related to data) ex- content inside a file is its data, but information like size, name ,permission , extension etc.

are attributes known as meta data , we can access such information through information related system calls

**Ex** - get system time & date.

**4. Process Control :-**To load a process in main memory we use process control system calls . Example - load , execute ,abort , Fork , Wait, signal , Allocate etc.

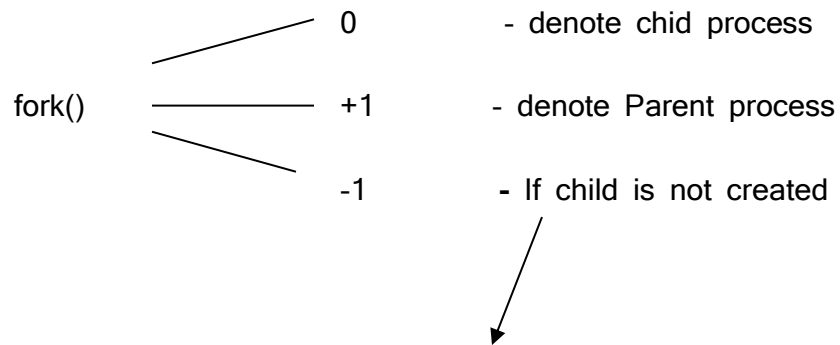
**5. Communication :-**Inter process communication ex - pipe() , shmget() - shared memory , client/delete connection.

### **Most important System Call :-**

**Fork() :-** In OS,we talk about multiprocessing environment i.e. we want multiple processes/ task to do at one time , for that we use multithreading or else we can use fork() system call - which a process creates child process ,parents doing its work & child doing its work so multi processes environment is created .

fork() system call is created through program control system calls. Ex- In a parent C program whenever we write fork() then a child of parent is created .

If parent is busy , then a child is created , child has its own id .



We do not consider this : we assume child is created

- Child process id is 0
- Parent process id is +1

gradeup





# Gradeup UGC NET Super Superscription

## Features:

1. 7+ Structured Courses for UGC NET Exam
2. 200+ Mock Tests for UGC NET & MHSET Exams
3. Separate Batches in Hindi & English
4. Mock Tests are available in Hindi & English
5. Available on Mobile & Desktop

---

Gradeup Super Subscription, Enroll Now