Unit 4 os

Q. What are files ?

Ans:- A file is a collection of information, it is stored in the form of records which contains fields for name, address and invoice amount. Storage devices store the files in the form of blocks. The file management system provides abstraction from storage blocks to data structures which is used by the application.

FILE SYSTEM FUNCTION

1. Users should be able to create, modify and delete files.
2. Users should be able to share each other file in a carefully controlled manner.
3. The mechanism for sharing files should provide various types of controlled access. Such as read access, write access or various combination of these task.
4. Users should be able to structure their files in a appropriate manner for each application.
5. Users should be able to order the transfer of information between files.
6. Backup and recovery capabilities must be provided to prevent either accidental loss of information.

Q. What is computer virus? What is the difference between VIRUS and WORM?

Ans:- A computer virus is a type of malicious software, or malware, that spreads between computers and causes damage to data and software.

Computer viruses aim to disrupt systems, cause major operational issues, and result in data loss and leakage. A computer viruses is designed to spread across programs and systems.

Virus	Worm
viruses use executable files to spread from one system to others.	Worms usually spread using a computer network,
while viruses require human action to replicate.	Worms can automatically replicate to different systems
The spreading speed of viruses is comparatively slower than worms.	The spreading speed of worms is faster than viruses, Because worms can replicate automatically, they spread at a much faster speed.
The viruses are designed to corrupt, delete, or modify the target device's data	worms don't harm the stored data but aim to harm the resources.
The viruses require hosts to spread from one device to another.	Worms, on the other side, don't need any host.
Viruses usually destroy and damage the stored data,	worms can harm the entire network by using maximum resources

Hierarchical Directory System

Hierarchical directory systems are used for users with a large number of files, as the single-level and two-level directory systems are not satisfactory.
Even on a computer with only one user, the two-level directory system is difficult and time-consuming to use. Users frequently express a desire to organize their files in a manner that makes sense to them, as a result. As a consequence of this, a general hierarchy of some kind is required.

The term "hierarchy" refers to a tree structure made up of directories.

The following hierarchical directory structure is depicted in the figure that follows:

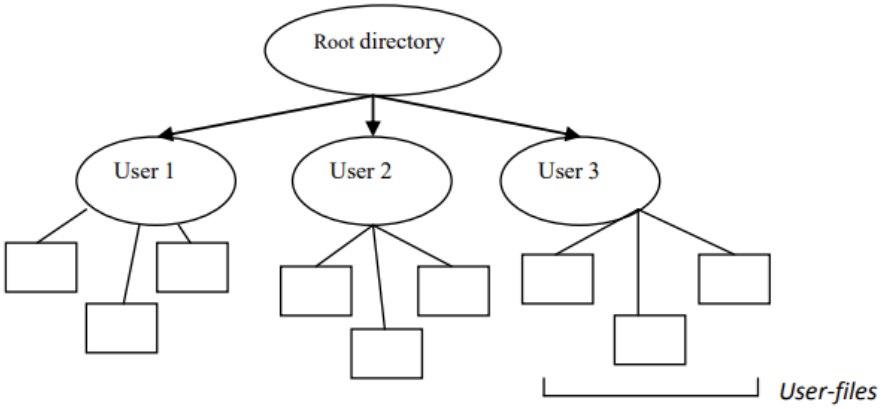


Fig: Two level hierarchical file system

As can be seen in the figure that is located above, the directories user1, user2, and user3 that are located within the root directory each belong to a different user

Q. Explain in details the UNIX FILE SYSTEM.

Ans. UNIX treats everything as files. Even a directory is treated as a file that contains entries for several other files. All devices such as devices, storage devices etc are all treated as files. There are 3 types of UNIX files viz-

(a) ORDINARY FILES

These files include the plain documents, program source code, program data, executable binary files and computer programs. Each ordinary files has a file name, its size in bytes, access permissions and a unique number called inode no. .

(b) SPECIAL FILES(SPECIAL DEVICE FILES)-

These files include physical on the system, such as hard disc, floppy disk, terminals, printers, system memory etc. these are all trcated as files by the UNIX.

(c) DIRECTORY FILES

These files are the files that contain the name and inode numbers of ordinary and/or directory files within it.

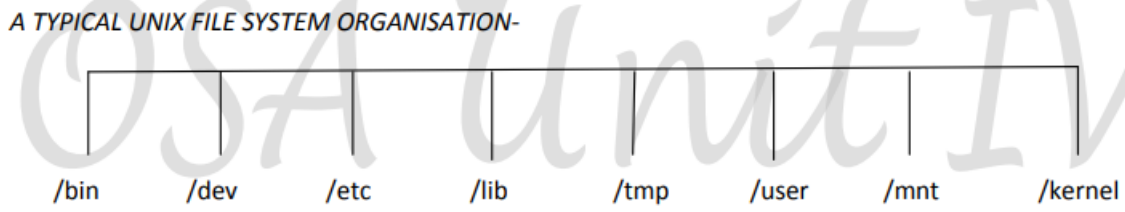


Fig. UNIX system directories

UNIX follows the three structured or hierarchical directory structure. The Unix file system is organized as hierarchy that starts with the root directory. The root is represented by the slash (/) under the root directory. There are several system and the home directories.

/ bin - This directory contains executable program files [binary files]. In this directory one can find the files for the UNIX commands.

/ dev - This directory contains the special device files.

/ etc - This contains all the system wide configration information as text files.

/ lib – This directory contains the library files. Library file contain the reusable functions and routines for the program to use.

/ temp – This directory contains all the temporary files which will eventually be deleted from the system. This is similar to the C ; \windows\temp\directory is windows based.

/mnt:- Used to mount other temporary file systems, such as cdrom and floppy for the CD-ROM drive and floppy diskette drive, respectively

/ usr - This directory contains all the home directories of the users. Source text for the online pages , games and other directories. There is one home directory for each user.

/ kernel - This directory contains all the kernel specific code .Kernel is the heart of the UNIX system. It is responsible for resource allocation security and lower level H/W interfaces.

File sharing

File sharing is the practice of sharing or offering access to digital information or resources, including documents, multimedia (audio/video), graphics, computer programs, images and e-books. It is the private or public distribution of data or resources in a network with different levels of sharing privileges.

File sharing can be done using several methods. The most common techniques for file storage, distribution and transmission include the following:

- Removable storage devices
- Centralized file hosting server installations on networks
- World Wide Web-oriented hyperlinked documents
- Distributed peer-to-peer networks

Distributed File System

A **distributed file system (DFS)** is a file system that is distributed on various file servers and locations. It permits programs to access and store isolated data in the same method as in the local files. It also permits the user to access files from any system. It allows network users to share information and files in a regulated and permitted manner. Although, the servers have complete control over the data and provide users access control.

DFS's primary goal is to enable users of physically distributed systems to share resources and information through the **Common File System (CFS)**.

DFS has two components in its services, and these are as follows:

- Local Transparency
  It is achieved via the namespace component.
- Redundancy
  It is achieved via a file replication component.

FILE OPERATION-

A file is an abstract data type. To define a file properly we need to consider the operations that can be performed on files. The O/S can provide system calls to create, write, read, delete and terminate files.

(1) Creating a file - The main step to create a file-space in the file system must be formed for the file.

(2) Writing a file - To write a file we make a system-call specifying between the name of the file and the information to be written to the file.

(3) Reading a file - To read from a files we use system calls that specifies the name of the file and the next block of the file should be put.

(4) Deleting a file - To delete a file we search the directory for the name file.

(5) Truncating a file – The user may want to erase the contents of a file but kept its attributes.

Unit-1

Real Time Operating System

A Real Time Operating System, commonly known as an RTOS, is a software component that rapidly switches between tasks, giving the impression that multiple programs are being executed at the same time on a single processing core.

In actual fact the processing core can only execute one program at any one time, and what the RTOS is actually doing is rapidly switching between individual programming threads (or Tasks) to give the impression that multiple programs are executing simultaneously.

Unit-2

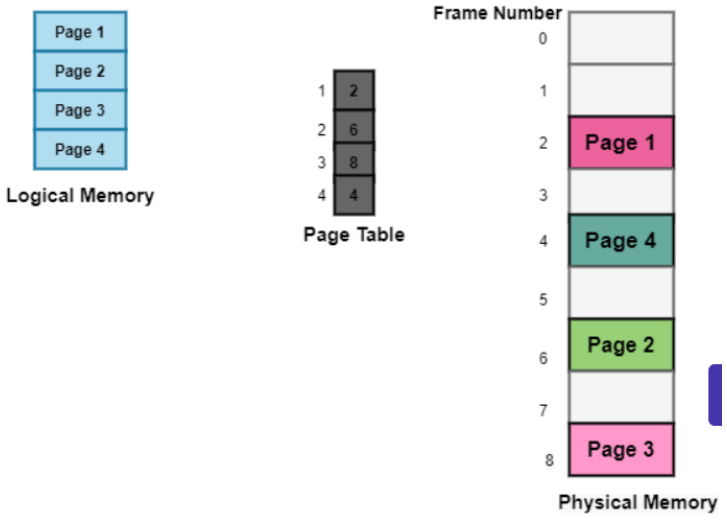
Memory compaction

- memory compaction is a form of memory management in which the operating system reclaims memory from inactive processes and makes it available to active processes.
- Memory compaction is the process of moving objects around and leaving empty space between them.
- Using this method, we can store larger processes in our memories.

Page Table.

The data structure that is used by the virtual memory system in the operating system of a computer in order to store the mapping between physical and logical addresses is commonly known as **Page Table**.

page table mainly provides the corresponding frame number where that page is stored in the main memory.



The above diagram shows the paging model of Physical and logical memory.

Unit-3

Race condition

A race condition is a condition when there are many processes and every process shares the data with each other and accessing the data concurrently, and the output of execution depends on a particular sequence in which they share the data and access.

RAMDisk

RAMDisk is a program that takes a portion of your system memory and uses it as a disk drive. The more RAM your computer has, the larger the RAMDisk you can create.

Interrupt handler

Interruptions are signals sent from a device or software to the operating system that are received from another device. Interrupt signals are handled by the interrupt handler, which is part of the operating system.

Q) What is redundant array of independent disk(RAID)?

➔ RAID is a way of storing the same data in different places on multiple hard disks or solid state drives(SSDs) to protect data in the case of a drive failure. RAID is a technique which makes use of a combination of multiple disks instead of using a single disk for increased performances, data redundancy or both.

Q) Characteristics of input output device.

-> i) Data Transfer Rate

ii) External storage capacity

iii) Types of access (Sequential access & direct access)

iv) Access Time

v) Device addressing

Unit-5

Indefinite Postponement:-

In any system that keeps processes waiting while it makes resource allocation and process scheduling decisions, it possible to delay indefinitely the scheduling of a process while other processes receives the system attention. This situation is called indefinite postponement or indefinite blocking or starvation.

Mutual Exclusion Problem:-

According to mutual exclusion when one process access the shared variables all other process are executed from doing so. The processes are allowed to access the variable only when the first one completes its execution. In this while one process uses the shared variables, all other processes are kept waiting.

Inter Process Communication

"Inter-process communication is used for exchanging useful information between numerous threads in one or more processes (or programs)." In general, Inter Process Communication is a type of mechanism usually provided by the operating system (or OS). The main aim or goal of this mechanism is to provide communications in between several processes. In short, the intercommunication allows a process letting another process know that some event has occurred.

Critical section

Critical Section is the part of a program which tries to access shared resources. That resource may be any resource in a computer like a memory location, Data structure, CPU or any IO device.The critical section cannot be executed by more than one process at the same time; operating system faces the difficulties in allowing and disallowing the processes from entering the critical section.The critical section problem is used to design a set of protocols which can ensure that the Race condition among the processes will never arise.

Peterson's algorithm –

```
flag[j] = true;
turn = i;
while (flag[i] == true && turn == i)
    flag[j] = false;
flag[i] = true;
    turn = j;
```

```
while (flag[j] == true && turn == j)
flag[i] = false;
```

Explanation of Peterson's algorithm –

Peterson's Algorithm is used to synchronize two processes. It uses two variables, a bool array **flag** of size 2 and an int variable **turn** to accomplish it.
In the solution i represents the Consumer and j represents the Producer. Initially the flags are false. When a process wants to execute it's critical section, it sets it's flag to true and turn as the index of the other process. This means that the process wants to execute but it will allow the other process to run first. The process performs busy waiting until the other process has finished it's own critical section.
After this the current process enters it's critical section and adds or removes a random number from the shared buffer. After completing the critical section, it sets it's own flag to false, indication it does not wish to execute anymore.

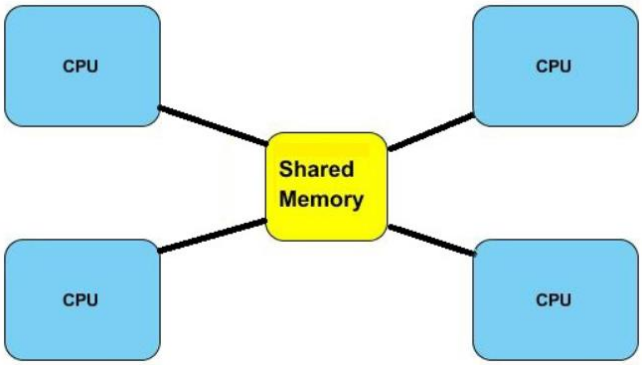
Concurrent Process:-

Process are concurrent if they exist at the same time .In such a case all of them can simultaneously for the system resources .concurrent process can function completely independent of one another or they need occurring synchronization .

There are two common models for concurrent programming: shared memory and message passing.

The Shared Memory Model

In this model stored information in a shared region of memory is processed, possibly under the control of a **supervisor** process.



The Message Passing Model

In this model, data is shared by sending and receiving messages between co-operating processes, using system calls .

Bernstein's Condition:-

Bernstein's Conditions are the conditions applied on two statements S1 and S2 that are to be executed in the processor. It states that three conditions that are explained below must be satisfied for two successive statements S1 and S2 to be executed concurrently and still produce the same result.

The intersection between read-write set, write-read set and write-write set of S1 and S2 must be null.

The above statement can be expressed in the form of expressions as following:

1. $R(S1) \cap W(S2) = \{ \}$

2. $W(S1) \cap R(S2) = \{ \}$

3. $W(S1) \cap W(S2) = \{ \}$

The read and write set for the statement c = a – b;

R(c=a-b) = {a, b}

W(c=a-b) = {c}

The read and write set for the statement w = c + 1;

R(w=c+1) = {c}

W(w=c+1) = {w}

The read and write set for the statement x = x + 2;

R(x=x+2) = {x}

W(x=x+2) = {x}

Let us take an example -

S1 : a = x + y

S2 : b = z + 1

Check whether two statements S1 and S2 satisfies Bernstein's conditions.

Solution:

R(S1) = {x, y}

W(S1) = {a}

R(S2) = {z}

W(S2) = {b}

1. $R(S1) \cap W(S2) = \{x, y\} \cap \{b\} = \{ \}$

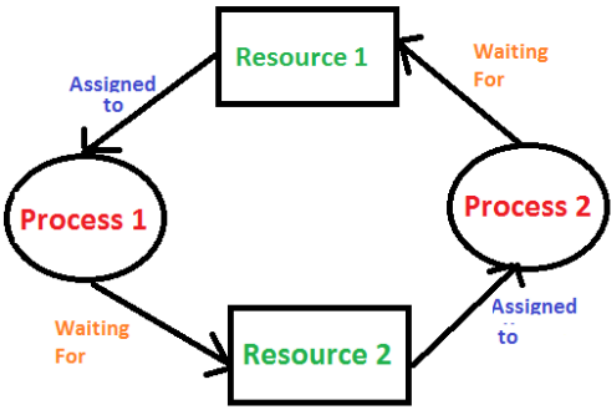
2. $W(S1) \cap R(S2) = \{a\} \cap \{z\} = \{ \}$

3. $W(S1) \cap W(S2) = \{a\} \cap \{b\} = \{ \}$

Therefore S1 ans S2 can be executed concurrently i.e., Bernstein's conditions are satisfied.

Deadlock:-Deadlock is a situation where a set of processes are blocked because each process is holding a resource and waiting for another resource acquired by some other process.

For example, in the below diagram, Process 1 is holding Resource 1 and waiting for resource 2 which is acquired by process 2, and process 2 is waiting for resource 1.



Deadlock Avoidance

When a process requests a resource, the deadlock avoidance algorithm examines the resource-allocation state. If allocating that resource sends the system into an unsafe state, the request is not granted.

Therefore, it requires additional information such as how many resources of each type is required by a process. If the system enters into an unsafe state, it has to take a step back to avoid deadlock.

Deadlock Detection and Recovery

We let the system fall into a deadlock and if it happens, we detect it using a detection algorithm and try to recover.

Some ways of recovery are as follows.

- Aborting all the deadlocked processes.
- Abort one process at a time until the system recovers from the deadlock.
- Resources are taken one by one from a process and assigned to higher priority processes until the deadlock is resolved.