Operating System

UNIT - I

Operating System - Introduction, Structures - Simple Batch, Multiprogrammed, Time-shared, Personal Computer, Parallel, Distributed Systems, Real-Time Systems, System components, Operating System services, System Calls

Introduction of Operating System

An operating system acts as an intermediary between the user of a computer and computer hardware. The purpose of an operating system is to provide an environment in which a user can execute programs in a convenient and efficient manner.

An operating system is a software that manages the computer hardware. The hardware must provide appropriate mechanisms to ensure the correct operation of the computer system and to prevent user programs from interfering with the proper operation of the system.

Operating System – Definition:

- An operating system is a program that controls the execution of application programs and acts as an interface between the user of a computer and the computer hardware.
- A more common definition is that the operating system is the one program running at all times on the computer (usually called the kernel), with all else being application programs.
- An operating system is concerned with the allocation of resources and services, such as memory, processors, devices, and information. The operating system correspondingly includes programs to manage these resources, such as a traffic controller, a scheduler, memory management module, I/O programs, and a file system.

Functions of Operating system – Operating system performs three functions:

- **Convenience:** An OS makes a computer more convenient to use.
- **Efficiency:** An OS allows the computer system resources to be used in an efficient manner.
- **Ability to Evolve:** An OS should be constructed in such a way as to permit the effective development, testing and introduction of new system functions at the same time without interfering with service.

Every general-purpose computer consists of the hardware, operating system, system programs, and application programs. The hardware consists of memory, CPU, ALU, and I/O devices, peripheral device, and storage device. System program consists of compilers, loaders, editors, OS, etc. The application program consists of business programs, database programs.

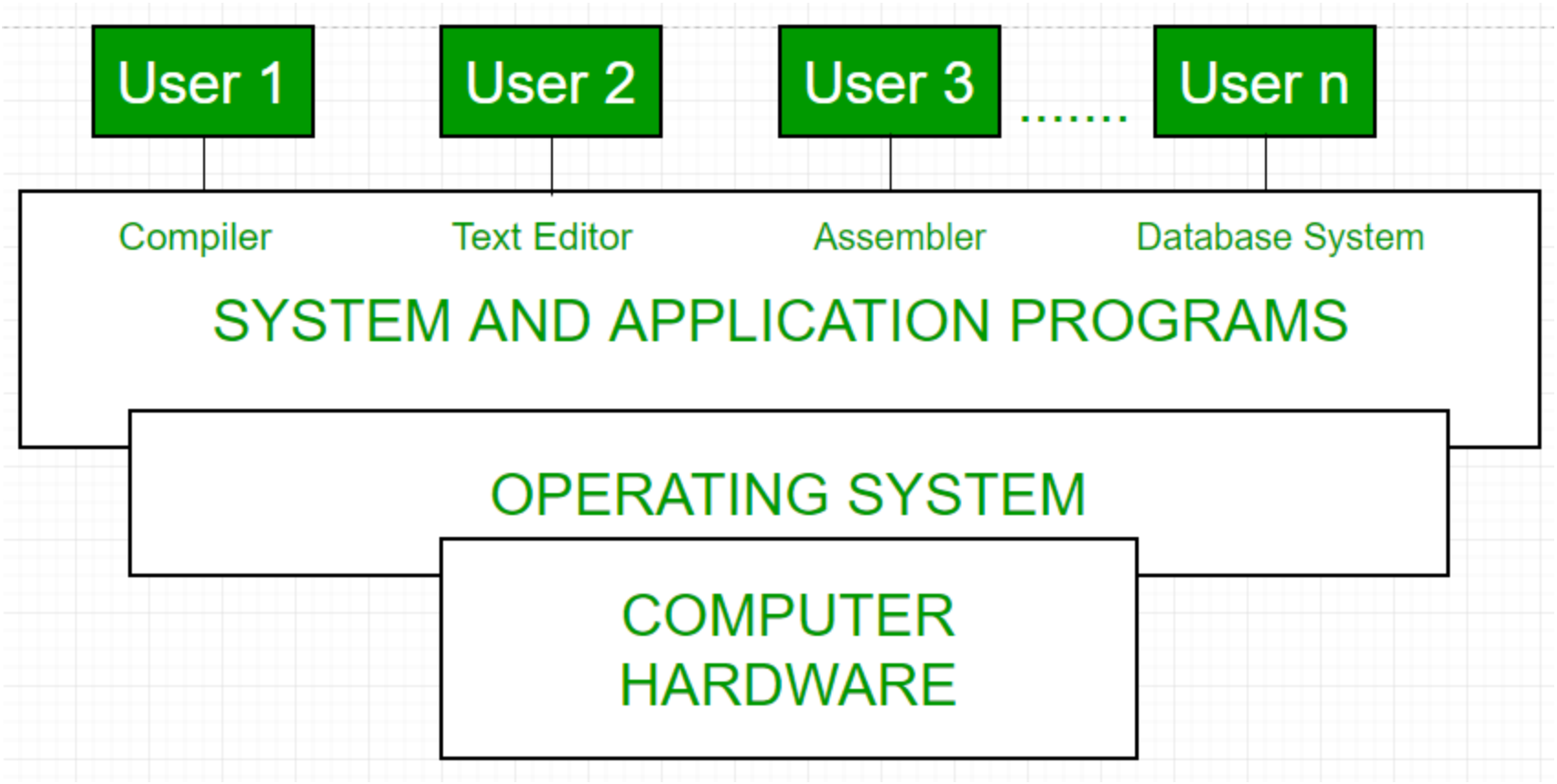


Fig: Conceptual view of a computer system

- Every computer must have an operating system to run other programs. The operating system coordinates the use of the hardware among the various system programs and application programs for various users. It simply provides an environment within which other programs can do useful work.
- The operating system is a set of special programs that run on a computer system that allows it to work properly. It performs basic tasks such as recognizing input from the keyboard, keeping track of files and directories on the disk, sending output to the display screen and controlling peripheral devices.

OS is designed to serve two basic purposes:

- It controls the allocation and use of the computing System's resources among the various user and tasks.
- It provides an interface between the computer hardware and the programmer that simplifies and makes feasible for coding, creation, debugging of application programs.

The Operating system must support the following tasks. The task are:

- Provides the facilities to create, modification of programs and data files using an editor.
- Access to the compiler for translating the user program from high level language to machine language.
- Provide a loader program to move the compiled program code to the computer’s memory for execution.
- Provide routines that handle the details of I/O programming.

Drivers for specific hardware devices.

- **Assembler –**
  The input to an assembler is an assembly language program. The output is an object program plus information that enables the loader to prepare the object program for execution. At one time, the computer programmer had at his disposal a basic machine that interpreted, through hardware, certain fundamental instructions. He would program this computer by writing a series of ones and Zeros (Machine language), place them into the memory of the machine.
- **Compiler –**
  The High-level languages- examples are FORTRAN, COBOL, ALGOL and PL/I are processed by compilers and interpreters. A compiler is a program that accepts a source program in a "high-level language "and produces a corresponding object program. An interpreter is a program that appears to execute a source program as if it was machine language. The same name (FORTRAN, COBOL, etc.) is often used to designate both a compiler and its associated language.

Loader –

A Loader is a routine that loads an object program and prepares it for execution. There are various loading schemes: absolute, relocating and direct-linking. In general, the loader must load, relocate and link the object program. The loader is a program that places programs into memory and prepares them for execution. In a simple loading scheme, the assembler outputs the machine language translation of a program on a secondary device and a loader places it in the core. The loader places into memory the machine language version of the user's program and transfers control to it. Since the loader program is much smaller than the assembler, those make more core available to the user's program.

History of Operating system –

Operating system has been evolving through the years. Following Table shows the history of OS.

Generation	Year	Electronic device used	Types of OS Device
First	1945-55	Vaccum Tubes	Plug Boards
Second	1955-65	Transistors	Batch Systems
Third	1965-80	Integrated Circuits(IC)	Multiprogramming
Fourth	Since 1980	Large Scale Integration	PC

Types of Operating System –

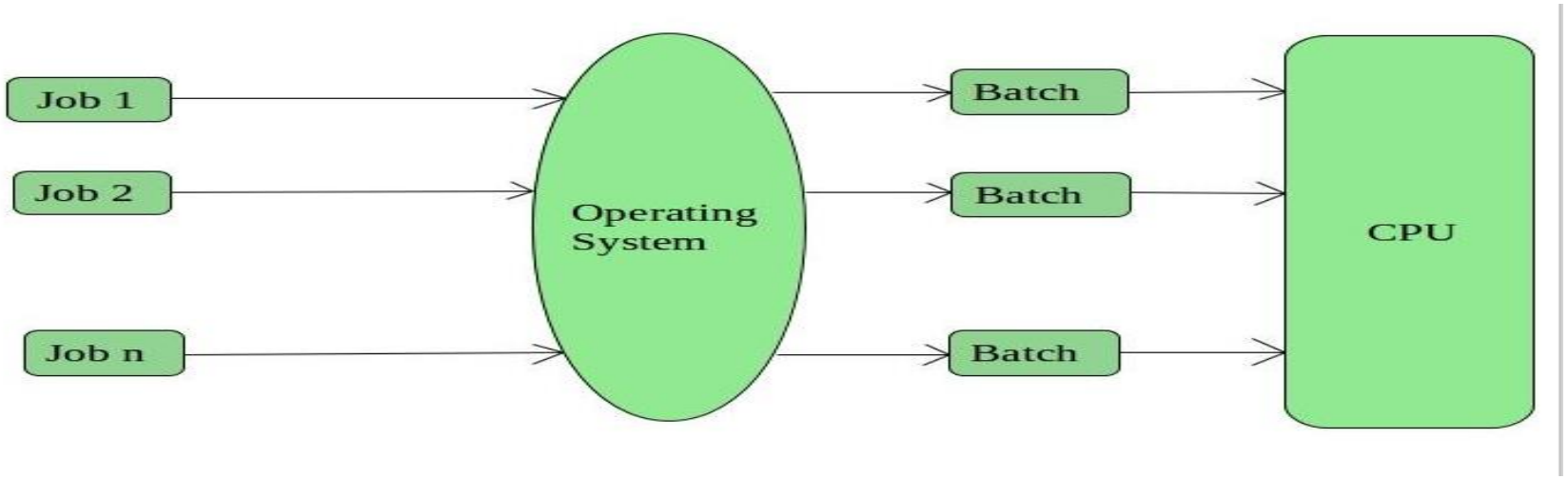
- Batch Operating System- Sequence of jobs in a program on a computer without manual interventions.
- Time sharing operating System- allows many users to share the computer resources.(Max utilization of the resources).
- Distributed operating System- Manages a group of different computers and make appear to be a single computer.
- Network operating system- computers running in different operating system can participate in common network (It is used for security purpose).
- Real time operating system – meant applications to fix the deadlines.

Examples of Operating System are –

- Windows (GUI based, PC)
- GNU/Linux (Personal, Workstations, ISP, File and print server, Three-tier client/Server)
- macOS (Macintosh), used for Apple's personal computers and work stations (MacBook, iMac).
- Android (Google's Operating System for smartphones/tablets/smartwatches)
- iOS (Apple's OS for iPhone, iPad and iPod Touch)
- Types of Operating Systems
- An Operating System performs all the basic tasks like managing files, processes, and memory. Thus operating system acts as the manager of all the resources, i.e. **resource manager**. Thus, the operating system becomes an interface between user and machine.

Types of Operating Systems: Some widely used operating systems are as follows-

1. **Batch Operating System –**
   This type of operating system does not interact with the computer directly. There is an operator which takes similar jobs having the same requirement and group them into batches. It is the responsibility of the operator to sort jobs with similar needs.



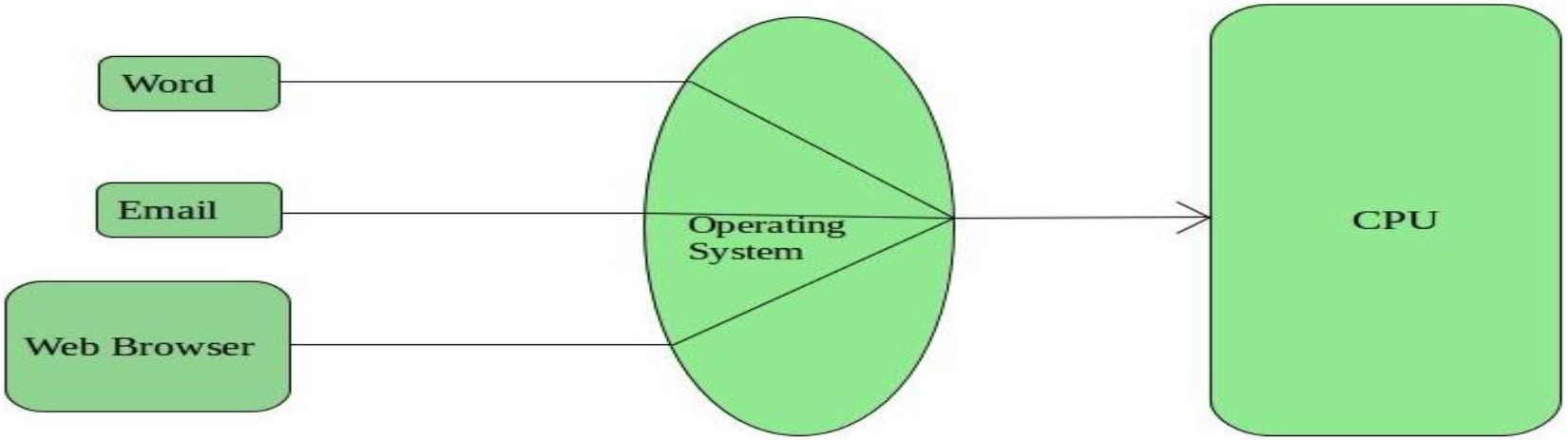
Advantages of Batch Operating System:

- It is very difficult to guess or know the time required for any job to complete. Processors of the batch systems know how long the job would be when it is in queue
- Multiple users can share the batch systems
- The idle time for the batch system is very less
- It is easy to manage large work repeatedly in batch systems
- **Disadvantages of Batch Operating System:**
- The computer operators should be well known with batch systems
- Batch systems are hard to debug
- It is sometimes costly
- The other jobs will have to wait for an unknown time if any job fails

Examples of Batch based Operating System: Payroll System, Bank Statements, etc.

2. Time-Sharing Operating Systems –

Each task is given some time to execute so that all the tasks work smoothly. Each user gets the time of CPU as they use a single system. These systems are also known as Multitasking Systems. The task can be from a single user or different users also. The time that each task gets to execute is called quantum. After this time interval is over OS switches over to the next task.



Advantages of Time-Sharing OS:

- Each task gets an equal opportunity
- Fewer chances of duplication of software
- CPU idle time can be reduced

Disadvantages of Time-Sharing OS:

- Reliability problem
- One must have to take care of the security and integrity of user programs and data
- Data communication problem

Examples of Time-Sharing OSs are: Multics, Unix, etc.

3. Distributed Operating System –

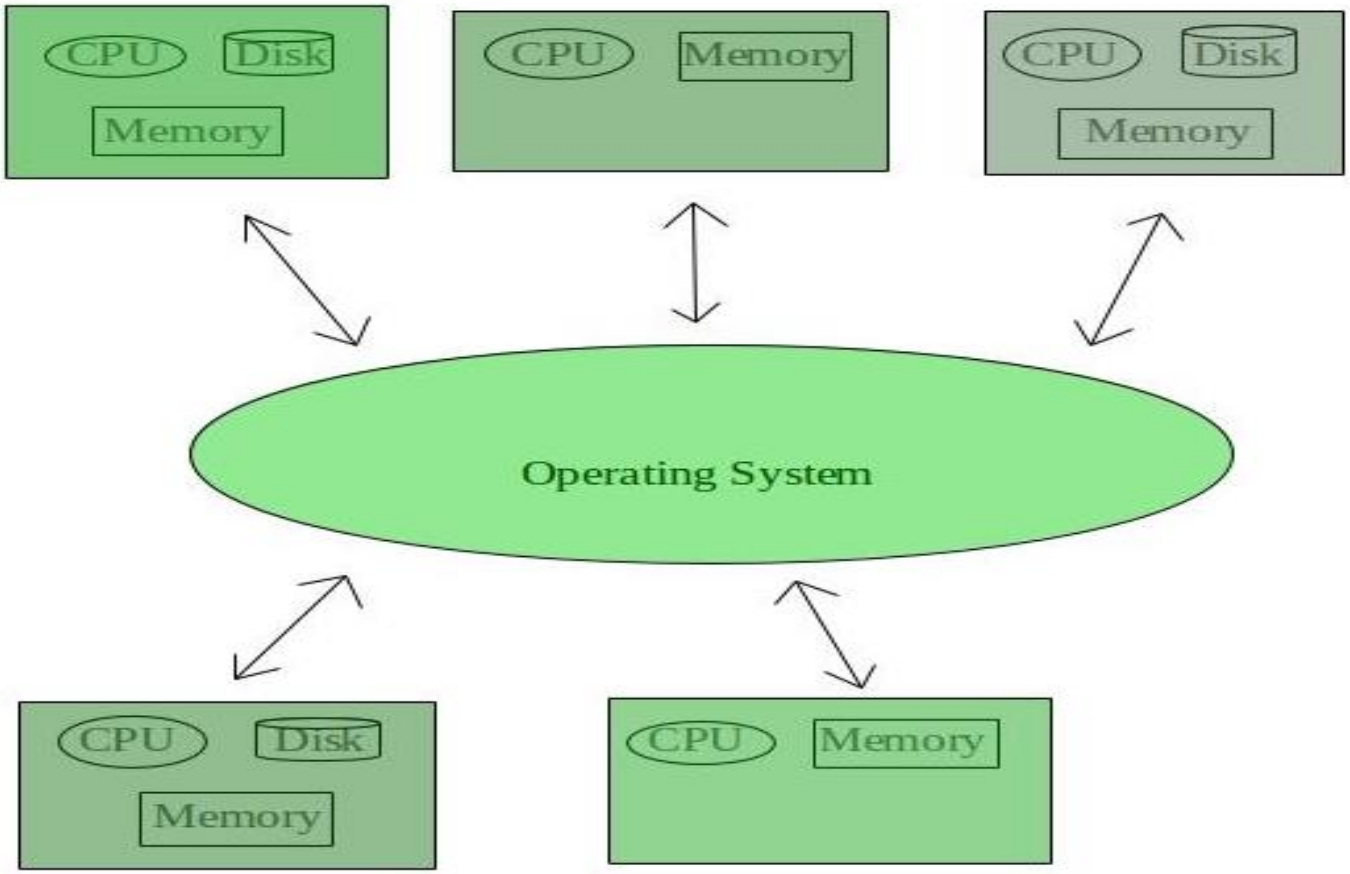
These types of the operating system is a recent advancement in the world of computer technology and are being widely accepted all over the world and, that too, with a great pace.

Various autonomous interconnected computers communicate with each other using a shared communication network. Independent systems possess their own memory unit and CPU.

These are referred to as **loosely coupled systems** or distributed systems. These system's processors differ in size and function.

The major benefit of working with these types of the operating system is that it is always possible that one user can access the files or software which are not actually present on his system but some other system connected within this network.

i.e., remote access is enabled within the devices connected in that network.


CPU
Disk
Memory
CPU
Memory
CPU
Disk
Memory
Operating System
CPU
Disk
Memory
CPU
Memory

Advantages of Distributed Operating System:

- Failure of one will not affect the other network communication, as all systems are independent from each other
- Electronic mail increases the data exchange speed
- Since resources are being shared, computation is highly fast and durable
- Load on host computer reduces
- These systems are easily scalable as many systems can be easily added to the network
- Delay in data processing reduces

Disadvantages of Distributed Operating System:

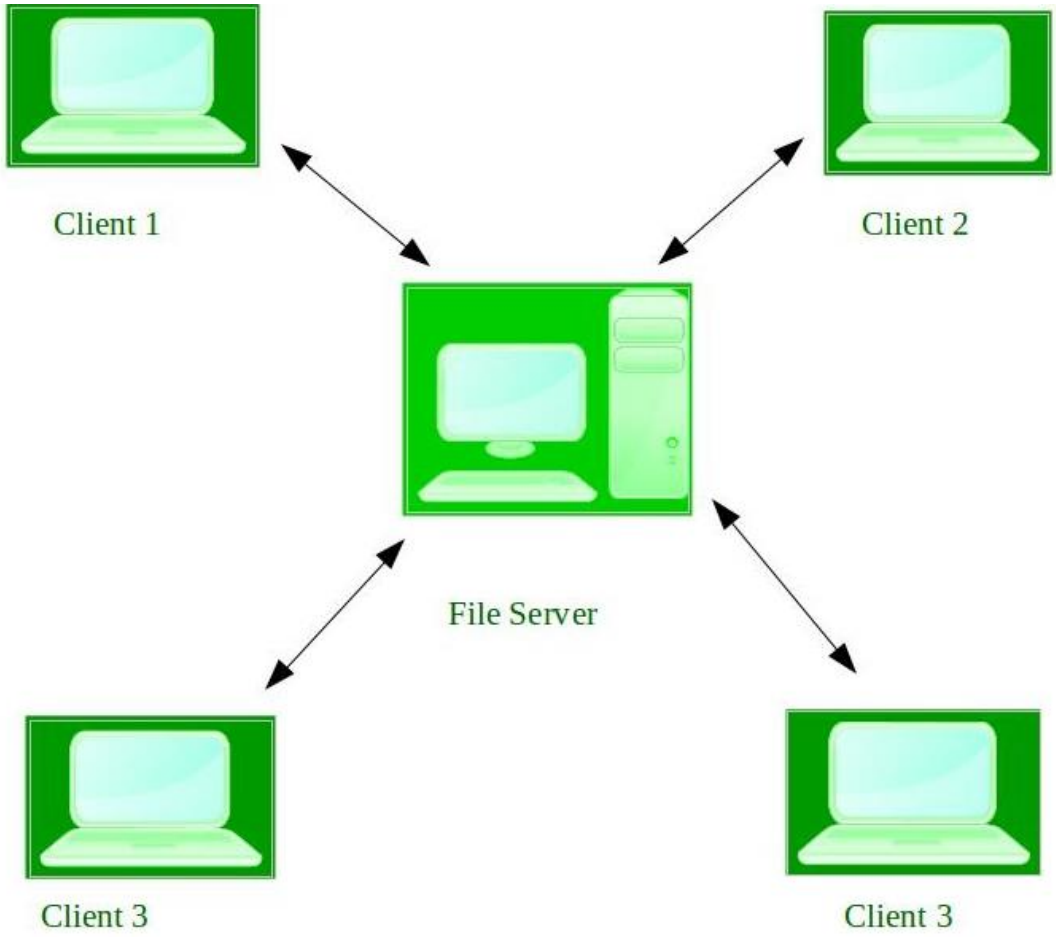
- Failure of the main network will stop the entire communication
- To establish distributed systems the language which is used are not well defined yet
- These types of systems are not readily available as they are very expensive. Not only that the underlying software is highly complex and not understood well yet
- **Examples of Distributed Operating System are-** LOCUS, etc.

4. Network Operating System –

These systems run on a server and provide the capability to manage data, users, groups, security, applications, and other networking functions.

These types of operating systems allow shared access of files, printers, security, applications, and other networking functions over a small private network.

One more important aspect of Network Operating Systems is that all the users are well aware of the underlying configuration, of all other users within the network, their individual connections, etc. and that's why these computers are popularly known as **tightly coupled systems**.


Client 1
Client 2
File Server
Client 3
Client 3

Advantages of Network Operating System:

- Highly stable centralized servers
- Security concerns are handled through servers
- New technologies and hardware up-gradation are easily integrated into the system
- Server access is possible remotely from different locations and types of systems

Disadvantages of Network Operating System:

- Servers are costly
- User has to depend on a central location for most operations
- Maintenance and updates are required regularly

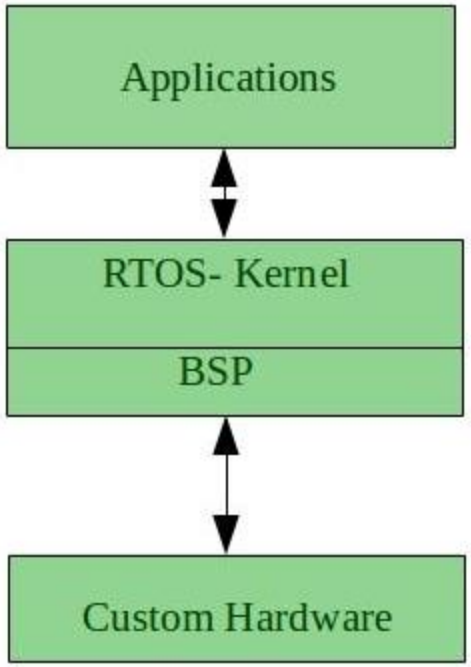
Examples of Network Operating System are: Microsoft Windows Server 2003, Microsoft Windows Server 2008, UNIX, Linux, Mac OS X, Novell NetWare, and BSD, etc.

5. Real-Time Operating System –
These types of OSs serve real-time systems. The time interval required to process and respond to inputs is very small. This time interval is called **response time**.

Real-time systems are used when there are time requirements that are very strict like missile systems, air traffic control systems, robots, etc.

Two types of Real-Time Operating System which are as follows:

- **Hard Real-Time Systems:**
  These OSs are meant for applications where time constraints are very strict and even the shortest possible delay is not acceptable. These systems are built for saving life like automatic parachutes or airbags which are required to be readily available in case of any accident. Virtual memory is rarely found in these systems.
- **Soft Real-Time Systems:**
  These OSs are for applications where for time-constraint is less strict.


Applications
RTOS- Kernel
BSP
Custom Hardware

Advantages of RTOS:

- **Maximum Consumption:** Maximum utilization of devices and system, thus more output from all the resources
- **Task Shifting:** The time assigned for shifting tasks in these systems are very less. For example, in older systems, it takes about 10 microseconds in shifting one task to another, and in the latest systems, it takes 3 microseconds.
- **Focus on Application:** Focus on running applications and less importance to applications which are in the queue.
- Real-time **operating system in** the **embedded system:** Since the size of programs are small, RTOS can also be used in embedded systems like in transport and others.
- **Error Free:** These types of systems are error-free.
- **Memory Allocation:** Memory allocation is best managed in these types of systems.

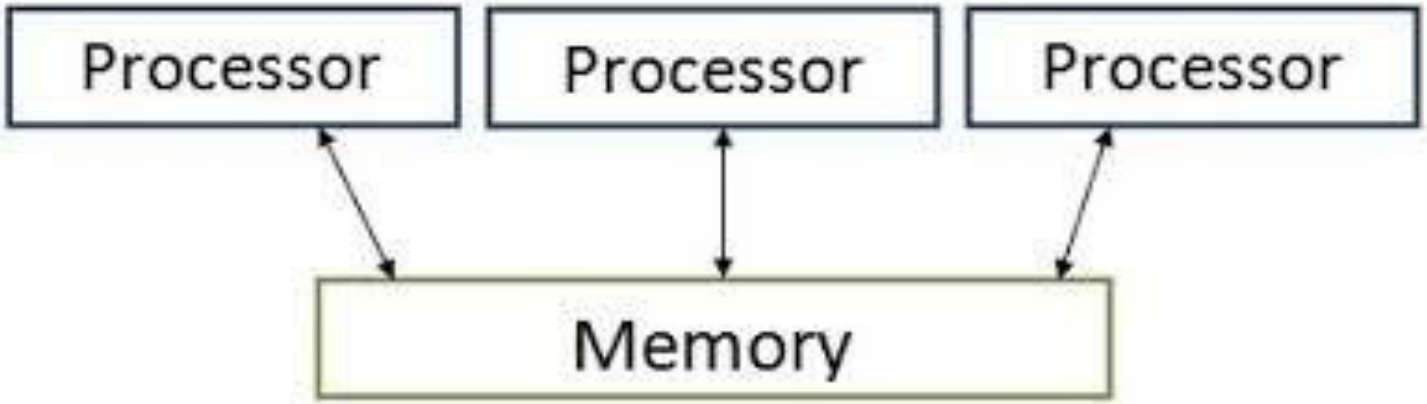
Disadvantages of RTOS:

- **Limited Tasks:** Very few tasks run at the same time and their concentration is very less on few applications to avoid errors.
- **Use heavy system resources:** Sometimes the system resources are not so good and they are expensive as well.
- **Complex Algorithms:** The algorithms are very complex and difficult for the designer to write on.
- **Device driver and interrupt signals:** It needs specific device drivers and interrupts signals to respond earliest to interrupts.
- **Thread Priority:** It is not good to set thread priority as these systems are very less prone to switching tasks.

Examples of Real-Time Operating Systems are: Scientific experiments, medical imaging systems, industrial control systems, weapon systems, robots, air traffic control systems, etc.

Parallel Operating System

- A parallel operating system works by dividing sets of calculations into smaller parts and distributing them between the machines on a network. To facilitate communication between the processor cores and memory arrays, routing software has to either share its memory by assigning the same address space to all of the networked computers, or distribute its memory by assigning a different address space to each processing core.
- Scientists, researches, and industries often choose to use parallel operating systems because of its cost effectiveness as well. It costs far less money to assemble a parallel computer network than it costs to develop and build a super computer for research.


Processor
Processor
Processor
Memory

7. Multi-programmed Batched Systems:

In multi-programmed batched operating systems, the operating system reads jobs from disk drives where a list of jobs are already being stored through card readers.

The operating system then pull and store as much job as it can in the memory. Then from the memory, operating system start working on a job.

Now, whenever a job reaches a situation where is has to be waiting for one or more tasks to be completed like use of any IO devices, the operating system pulls another job from the memory and starts working on it.

Whenever this job also starts waiting, for example it need to use the same IO which is already in use by its previous job, the operating systems pulls another job. This is how, a multi-programmed batched systems harness the power of disk drives and memory.

8.Personal Computer Systems:

Personal Computer Operating Systems are used in PCs.

These operating systems are slightly different from the other sophisticated operating systems in the way that other operating systems tends to give priority to utilizing the hardware and maximizing the security.

On the other hand, personal computer operating systems tend to maximize the user's convenience and ease of use of the operating system. PCs are made to be used by a single user, resource utilization gets less priority over the user's comfort of using the operating system.

9. Parallel Systems:

Parallel operating systems (also tightly coupled systems) are supposed to be used in machines with more than one processor connected closely where these multiple processors will be sharing memory, clock, buses and other peripherals. Parallel systems are designed to distribute a job to multiple processors to have faster speed.

But the speed-ratio is for n processor is not n the times for a single processor. As, when multiple processors cooperate on a task, a certain amount of overhead is incurred in keeping all the parts working correctly.

Different model of parallel systems exists, such as symmetric-multiprocessing model where each processors runs a copy of the operating system, these operating systems communicate each other when required.

Another model is asymmetric-multiprocessing where each processors get a specific task to do.

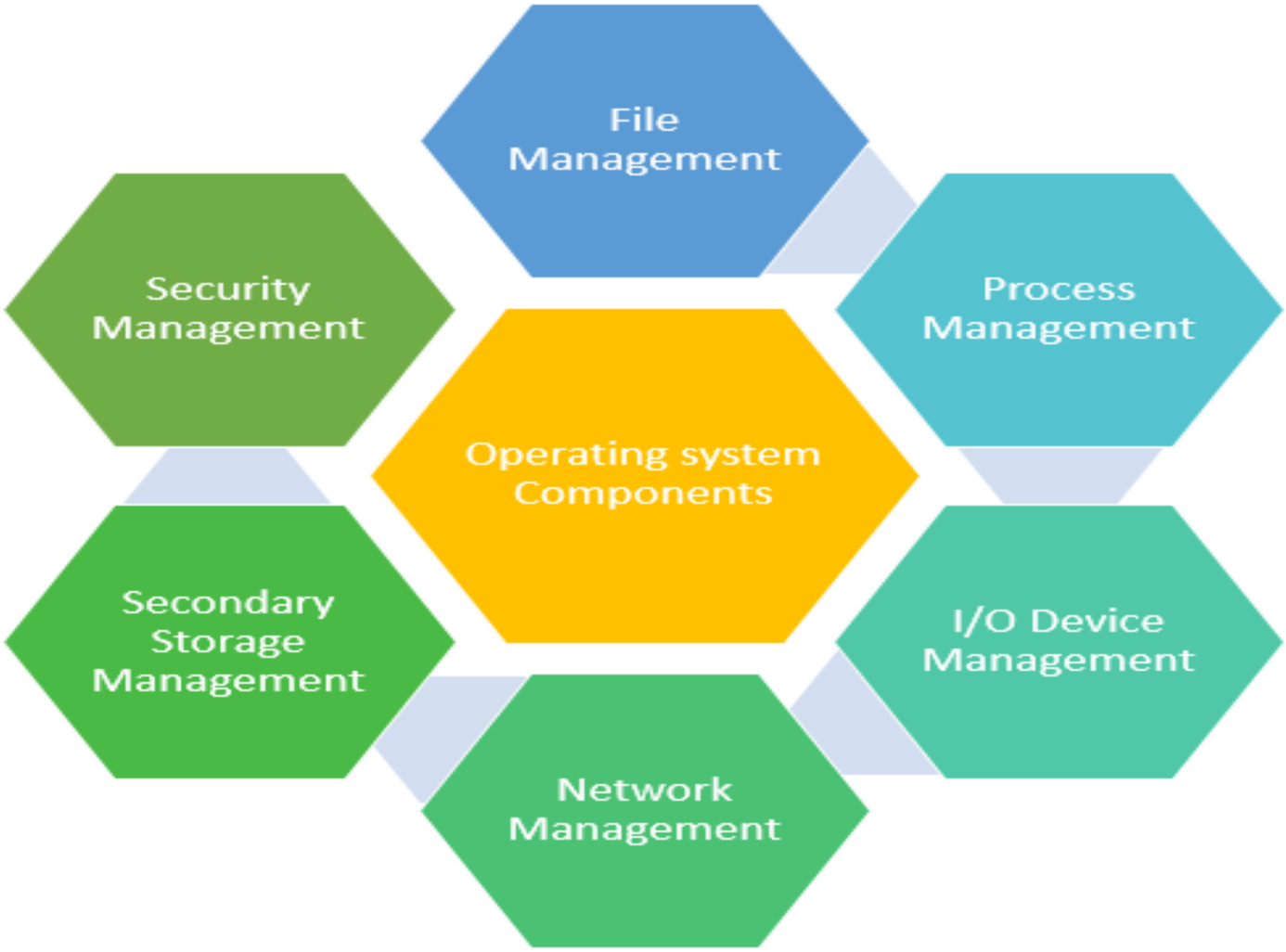
A master processor controls other processors and maintain the workflow.

This is a master-slave situation where slaves' checkout for tasks from masters or sit idle.

Components of Operating Systems

What are OS Components?

- An operating system is a large and complex system that can only be created by partitioning into small pieces. These pieces should be a well-defined portion of the system, which carefully defined inputs, outputs, and functions.
- Although Mac, Unix, Linux, Windows, and other OS do not have the same structure, most of the operating systems share similar OS system components like File, Process, Memory, I/O device management.
- File Management
- Process Management
- I/O Device Management
- Network Management
- Main Memory management
- Secondary-Storage Management
- Security Management
- Other Important Activities


File Management
Process Management
Security Management
Operating system Components
Secondary Storage Management
I/O Device Management
Network Management

File Management

- A file is a set of related information which is should define by its creator. It commonly represents programs, both source and object forms, and data. Data files can be numeric, alphabetic, or alphanumeric.

Function of file management in OS:

- The operating system has the following important given activities in connections with file management:
- File and directory creation and deletion.
- For manipulating files and directories.
- Mapping files onto secondary storage.
- Backup files on stable storage media.

Process Management

- The process management component is a procedure for managing the many processes that are running simultaneously on the operating system. Every software application program has one or more processes associated with them when they are running.
- For example, when you use a browser like Google Chrome, there is a process running for that browser program. The OS also has many processes running, which performing various functions.
- All these processes should be managed by process management, which keeps processes for running efficiently. It also uses memory allocated to them and shutting them down when needed.
- The execution of a process must be sequential so, at least one instruction should be executed on behalf of the process.

Functions of process management in OS:

- The following are functions of process management.
- Process creation and deletion.
- Suspension and resumption.
- Synchronization process
- Communication process

I/O Device Management

- One of the important use of an operating system that helps you to hide the variations of specific hardware devices from the user.

Functions of I/O management in OS:

- It offers buffer caching system
- It provides general device driver code
- It provides drivers for particular hardware devices.
- I/O helps you to knows the individualities of a specific device.

Network Management

- Network management is the process of administering and managing computer networks. It includes performance management, fault analysis, provisioning of networks, and maintaining the quality of service.
- A distributed system is a collection of computers/processors that never share their own memory or a clock. In this type of system, all the processors have their local Memory, and the processors communicate with each other using different communication lines, like fiber optics or telephone lines.
- The computers in the network are connected through a communication network, which can be configured in a number of different ways. With the help of network management, the network can be fully or partially connected, which helps users to design routing and connection strategies that overcome connection and security issues.

Functions of Network management:

- Distributed systems help you to various computing resources in size and function. They may involve microprocessors, minicomputers, and many general-purpose computer systems.
- A distributed system also offers the user access to the various resources the network shares.
- It helps to access shared resources that help computation to speed-up or offers data availability and reliability.

Main Memory management

- Main Memory is a large array of storage or bytes, which has an address. The memory management process is conducted by using a sequence of reads or writes of specific memory addresses.
- In order to execute a program , it should be mapped to absolute addresses and loaded inside the Memory. The selection of a memory management method depends on several factors.
- However, it is mainly based on the hardware design of the system. Each algorithm requires corresponding hardware support. Main Memory offers fast storage that can be accessed directly by the CPU. It is costly and hence has a lower storage capacity. However, for a program to be executed, it must be in the main Memory.

Functions of Memory management in OS:

- An Operating System performs the following functions for Memory Management:
- It helps you to keep track of primary memory.
- Determine what part of it are in use by whom, what part is not in use.
- In a multiprogramming system, the OS takes a decision about which process will get Memory and how much.
- Allocates the memory when a process requests
- It also de-allocates the Memory when a process no longer requires or has been terminated.

Secondary-Storage Management

- The most important task of a computer system is to execute programs. These programs, along with the data, helps you to access, which is in the main memory during execution.
- This Memory of the computer is very small to store all data and programs permanently. The computer system offers secondary storage to back up the main Memory. Today modern computers use hard drives/SSD as the primary storage of both programs and data. However, the secondary storage management also works with storage devices, like a USB flash drive, and CD/DVD drives.
- Programs like assemblers, compilers, stored on the disk until it is loaded into memory, and then use the disk as a source and destination for processing.

Functions of Secondary storage management in OS:

- Here, are major functions of secondary storage management in OS:
- Storage allocation
- Free space management
- Disk scheduling

Security Management

- The various processes in an operating system need to be secured from each other's activities. For that purpose, various mechanisms can be used to ensure that those processes which want to operate files, memory CPU, and other hardware resources should have proper authorization from the operating system.
- For example, Memory addressing hardware helps you to confirm that a process can be executed within its own address space. The time ensures that no process has control of the CPU without renouncing it.
- Lastly, no process is allowed to do its own I/O, to protect, which helps you to keep the integrity of the various peripheral devices.

Other Important Activities

- Here, are some other important activities of OS:
- The user's program can't execute I/O operations directly. The operating system should provide some medium to perform this.
- OS checks the capability of the program to read, write, create, and delete files.
- OS facilitates an exchange of information between processes executing on the same or different systems.
- OS components help you to makes sure that you get the correct computing by detecting errors in the CPU and memory hardware.

Operating System - Services

- An Operating System provides services to both the users and to the programs.
- It provides programs an environment to execute.
- It provides users the services to execute the programs in a convenient manner.
- Following are a few common services provided by an operating system −
- Program execution
- I/O operations
- File System manipulation
- Communication
- Error Detection
- Resource Allocation
- Protection

Program execution

- Operating systems handle many kinds of activities from user programs to system programs like printer spooler, name servers, file server, etc. Each of these activities is encapsulated as a process.
- A process includes the complete execution context (code to execute, data to manipulate, registers, OS resources in use). Following are the major activities of an operating system with respect to program management −
- Loads a program into memory.
- Executes the program.
- Handles program's execution.
- Provides a mechanism for process synchronization.
- Provides a mechanism for process communication.
- Provides a mechanism for deadlock handling.

I/O Operation

- An I/O subsystem comprises of I/O devices and their corresponding driver software. Drivers hide the peculiarities of specific hardware devices from the users.
- An Operating System manages the communication between user and device drivers.
- I/O operation means read or write operation with any file or any specific I/O device.
- Operating system provides the access to the required I/O device when required.

File system manipulation

- A file represents a collection of related information. Computers can store files on the disk (secondary storage), for long-term storage purpose. Examples of storage media include magnetic tape, magnetic disk and optical disk drives like CD, DVD. Each of these media has its own properties like speed, capacity, data transfer rate and data access methods.
- A file system is normally organized into directories for easy navigation and usage. These directories may contain files and other directions. Following are the major activities of an operating system with respect to file management −
- Program needs to read a file or write a file.
- The operating system gives the permission to the program for operation on file.
- Permission varies from read-only, read-write, denied and so on.
- Operating System provides an interface to the user to create/delete files.
- Operating System provides an interface to the user to create/delete directories.
- Operating System provides an interface to create the backup of file system.

Communication

- In case of distributed systems which are a collection of processors that do not share memory, peripheral devices, or a clock, the operating system manages communications between all the processes. Multiple processes communicate with one another through communication lines in the network.
- The OS handles routing and connection strategies, and the problems of contention and security. Following are the major activities of an operating system with respect to communication −
- Two processes often require data to be transferred between them
- Both the processes can be on one computer or on different computers, but are connected through a computer network.
- Communication may be implemented by two methods, either by Shared Memory or by Message Passing.

Error handling

- Errors can occur anytime and anywhere. An error may occur in CPU, in I/O devices or in the memory hardware. Following are the major activities of an operating system with respect to error handling −
- The OS constantly checks for possible errors.
- The OS takes an appropriate action to ensure correct and consistent computing.

Resource Management

- In case of multi-user or multi-tasking environment, resources such as main memory, CPU cycles and files storage are to be allocated to each user or job. Following are the major activities of an operating system with respect to resource management −
- The OS manages all kinds of resources using schedulers.
- CPU scheduling algorithms are used for better utilization of CPU.

Protection

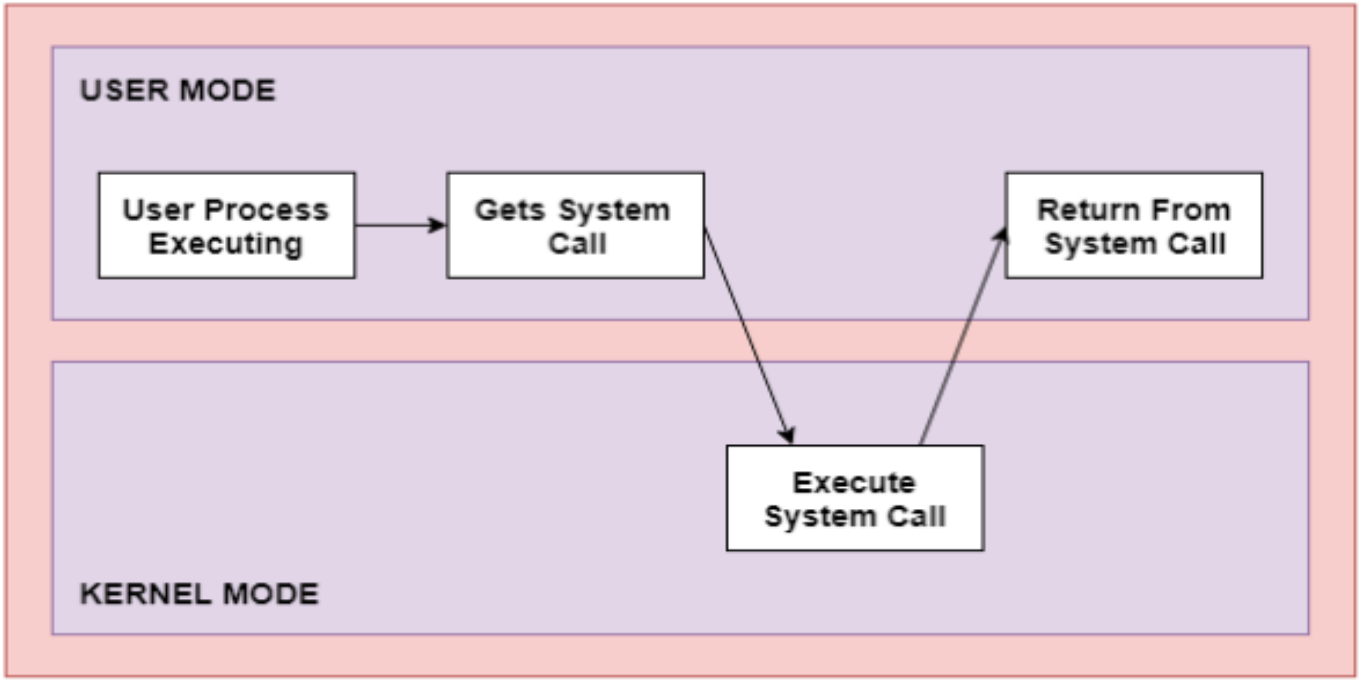
- Considering a computer system having multiple users and concurrent execution of multiple processes, the various processes must be protected from each other's activities.
- Protection refers to a mechanism or a way to control the access of programs, processes, or users to the resources defined by a computer system. Following are the major activities of an operating system with respect to protection –
- The OS ensures that all access to system resources is controlled.
- The OS ensures that external I/O devices are protected from invalid access attempts.
- The OS provides authentication features for each user by means of passwords.

What are system calls in Operating System?

The interface between a process and an operating system is provided by system calls. In general, system calls are available as assembly language instructions.

They are also included in the manuals used by the assembly level programmers. System calls are usually made when a process in user mode requires access to a resource. Then it requests the kernel to provide the resource via a system call.

A figure representing the execution of the system call is given as follows −



- As can be seen from this diagram, the processes execute normally in the user mode until a system call interrupts this. Then the system call is executed on a priority basis in the kernel mode. After the execution of the system call, the control returns to the user mode and execution of user processes can be resumed.
- In general, system calls are required in the following situations −
- If a file system requires the creation or deletion of files. Reading and writing from files also require a system call.
- Creation and management of new processes.
- Network connections also require system calls. This includes sending and receiving packets.
- Access to a hardware devices such as a printer, scanner etc. requires a system call.

Types of System Calls

- There are mainly five types of system calls. These are explained in detail as follows −

Process Control

- These system calls deal with processes such as process creation, process termination etc.

File Management

- These system calls are responsible for file manipulation such as creating a file, reading a file, writing into a file etc.

Device Management

- These system calls are responsible for device manipulation such as reading from device buffers, writing into device buffers etc.

Information Maintenance

- These system calls handle information and its transfer between the operating system and the user program.

Communication

- These system calls are useful for interprocess communication. They also deal with creating and deleting a communication connection.

Types of System Calls	Windows	Linux
Process Control	CreateProcess() ExitProcess() WaitForSingleObject()	fork() exit() wait()
File Management	CreateFile() ReadFile() WriteFile() CloseHandle()	open() read() write() close()
Device Management	SetConsoleMode() ReadConsole() WriteConsole()	ioctl() read() write()
Information Maintenance	GetCurrentProcessID() SetTimer() Sleep()	getpid() alarm() sleep()
Communication	CreatePipe() CreateFileMapping() MapViewOfFile()	pipe() shmget() mmap()