1. What is kernal? Difference between molithic kernal and microlithic.

🡪Kernal is central component of an operating system that manages operations of computer and hardware. It basically manages operations of memory and CPU time. It is core component of an operating system. Kernel acts as a bridge between applications and data processing performed at hardware level using inter-process communication and system calls.

|  |  |  |
| --- | --- | --- |
| **Features** | **Microkernel** | **Monolithic Kernel** |
| **Definition** | It is a kernel type that implements an operating system by providing low-level address space management, IPC, and thread management. | It is a type of kernel in which the complete operating system runs at the kernel speed. |
| **Size** | It is smaller in size. | It is bigger than the microkernel. |
| **Speed** | Its process execution is slower. | Its process execution is faster. |
| **Basic** | It implements kernel and user services in different address spaces. | It implements both user and kernel services in the same address space. |
| **Security** | It is more secure than the monolithic kernel. | It is less secure than the microkernel. |
| **Stability** | A single process failure does not affect other processes. | In a monolithic kernel, if a service fails, the entire system fails. |
| **Extendible** | It is easy to extend. | It is hard to extend. |
| **Code** | More code is necessary to write a microkernel. | Less code is necessary to write a monolithic kernel. |
| **Inter-Process** | Communication Microkernels use the messaging queues to achieve IPC. | The monolithic kernels use signals and sockets to achieve IPC. |
| **Maintainability It is easily maintainable.** | Maintenance takes extra time and resources. |  |
| **Debug** | It is easy to debug. | It is hard to debug. |
| **Example** | Symbian, L4Linux, K42, Mac OS X, PikeOS, HURD, etc. | Linux, BSDs, Solaris, OS-9, DOS, OpenVMS, etc. |

1. Explain different service/functions provided by operating system.

->An operating system is collection of software that manage hardware resources and also provides various service to the users.

1. Pogram Devlopment

->It provides editors and debuggers to assist(help) the programmer in creating programs.

Example:visual studio, sublimetext , spacemacs

1. Program execution

->following tasks need to be perform execute a program:

1>Instructions and data must be loaded into main memory

2>I/O devices and files must be initialized.

1. Memory management

->OS manages memory hierarchy.

->OS keeps the track of which part of memory area in use and free memory

->It allocates memory to program when they need it.

->It de-allocate the memory when the program finish execution

1. Access to I/O devices (Resource allocation)

->A running program may require I/O, which may involve file or an I/O device.

-> For efficiency and protection, users cannot control 1/0 devices directly.

->Therefore, the OS controls these I/O devices and provides to program as per requirement.

1. Controlled access to file

->In case of file access, OS provides a directory hierarchy for easy access and management of file.

->OS provides various file handling commands using which user can easily read, write and modify file.

1. Error detection and response

->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.

1. Accounting

->Keeping a track of which users are using how much and what kinds of computer resources can be used for accounting or simply for accumulating usage statistics.

->Usage statistics is used to reconfigure the system to improve computing services.

1. Protection & Security

->Protection involves ensuring that all accesses to system resources is controlled.

->To make a system secure, the user needs to authenticate himself or herself to the system.

1. Communication

->In multitasking environment, the processes need to communicate with each other and to exchange their information.

->Operating system performs the communication among various types of processes in the form of shared memory.

1. Discuss role of OS as a resource manager.

->There are lots of resources in computer system

-CPU (Processor)

-Memory

-I/O devices such as hard disk, mouse, keyboard, printer, scanner etc.

-> If a computer system is used by multiple applications (or users), then they will compete for these resources.

->It is the job of OS to allocate these resources to the various applications so that:

-The resources are allocated fairly (equally).

-The resources are protected from cross-access.

-Acccess to the recourses is synchronized so that operations are correct and consistent.

-Example:7+9-6\*4/2=20 is wrong in C languages.

-Deadlock are detected, resolved and avoided.

->Two different ways:

1.In space sharing

- **in-space** sharing, in this, the users share the space of resource.

For example**,** both primary and secondary memories are space shared, where OS allocates them to the user program and keeps the track of it.

The best example of in-space sharing is when the main memory consists of several processes at a time so that the processes can share the resources.

2.In time sharing

-When a resource is shared **in time**, different programs can take the chance to use it.

-The best example of in-time sharing is **CPU Scheduling on OS.**

-In that, OS fixes the time slot for the CPU, First one of the processes gets the CPU, when the time slot expires, the CPU switch to the next process in the ready queue. In this example, the CPU acts as a time resource.

1. Explain the features of Time-sharing system.

->A time sharing operating system allows many users to share the computer resources simultaneously.

-The CPU time is divided into small intervals, or time slices, and each user is given a small portion of time during which their program or task can execute.

Parallel Processing Operating System

-> Parallel Processing Operating Systems are designed to speed

up the execution of programs by dividing the program into multiple fragments and processing these fragments simultaneously.

Multiprocessing Operating System

-> Such systems are multiprocessor systems.

-> Parallel systems deal with the simultaneous use of multiple computer resources that can include a uingle computer with multiple processors.

5)Write different types of system call.

->A system call is a way for programs to interact with the operating system.

-> A system call is a mechanism that provides the interface between a process and the operating system.

-> A computer program makes a system call when it makes a request to the operating system's software kernel.

->It is a programmatic method in which a computer program requests a service from the kernel of the OS. Hardware

-> System call provides the services of the operating system to the user programs via Application Program Interface(API)

->System calls are the only entry points for the kernel system.

1.Process Control: This system calls perform the task of process creation, process termination, etc.

->Functions:

-End and abort

-Load and execute

-Create process and terminate process

-Wait and signed event

-Allocate and free memory

2.File Management: File management system calls handle file manipulation jobs like creating a file, reading, and writing, etc

->Functions:

-Create a file

-Delete file

-Open and close file

-Read, write and reposition

-Get and set file attributes

3. Device Management: Device management does the job of device manipulation like reading from device buffers, writing into device buffers, etc.

->Functions

Request and release device

Logically attach detach devices

Get and Set device attributes

4.Information Maintenance: It handles information and its transfer between the OS and user program

->Functions:

Get or set time and date

Get process and device attributes

5. Communication: These types of system calls are specially used for interprocess communications (IPC).

->Functions:

-Create, delete communications connections

-Send, receive message

-Help OS to transfer status information

-attach or detach remote devices

=>Example: In Unix Read system call is count = read(fd, buffer, nbytes)

-fd is a file descriptor.

-When a file is opened, permissions are checked.

-If access is allowed, a number (fd) is returned. Then file can be read/written, nbytes is number of bytes to read.

-buffer is where read deposits (stores) the data.

6)What is system call? Explain steps for system call execution.

* + - * Steps involved in making a system call
      * In steps 1-3, the calling program pushes the parameters onto the stack. The first and third parameters are called by value, but the second one is called by its address as denoted by the & symbol.
      * In step 4, the actual call to the library procedure is made. This instruction is the normal procedure call Instruction used to call all procedures.
      * In step 5, the library procedure places the system call number in a place where the operating system expects it, such as a register.
      * In step 6, the library procedure executes a TRAP instruction to switch from user mode to kernel mode and start execution at a fixed address within the kernel.
      * In step 7, the kernel examines the system call number and then dispatches it to the correct system call handler. This correct number is given in the table of system call handlers by pointers referenced at the system call number.
      * In step 8, the system call handler runs.
      * In step 9, the operation is completed, and the user is given back control once the TRAP instruction is set.
      * In step 10, this procedure returns to the user program, like how all normal library procedures do.
      * In step 11, the operating system has to clear the stack, so it increments it enough so that it is empty.

7) List out types of operating system and explain batch OS and time-sharing OS in brief.

#### **Types of operating system:**

1. Mainframe Operating Systems
2. Server Operating Systems
3. Multiprocessor Operating Systems
4. Personal Computer Operating Systems
5. Handhelds Computer Operating Systems
6. Embedded Operating Systems
7. Sensor Node Operating Systems
8. Real Time Operating Systems
9. Smart Card Operating Systems

#### **Batch Operating System**

* Batch operating system is one that processes routine jobs without any interactive user presents. Such as claim processing in insurance and sales reporting etc.
* To improve utilization, the concept of batch operating system was developed. Jobs with similar needs were batched together and were run through the computer as a group.
* Thus, the programmer would leave their program with operator, who in turn would sort program into batches with similar requirements.
* The operator then loaded a special program (the ancestor of today’s operating system), which read the first job from magnetic tape and run it.
* The output was written onto a second magnetic tape, instead of being printed.
* After each job finished, the operating system automatically read the next job from the tape and began running it.
* When the whole batch was done, the operator removed the input and output tapes, replaced the input tape with the next batch, and brought the output tape for offline printing.
* With the use of this type of operating system, the user no longer has direct access to machine.

#### **Time sharing Operating System**

* Time Sharing is a logical extension of multiprogramming.
* Multiple jobs are executed by the simultaneously by switching the CPU back and forth among them.
* The switching occurs so frequently (speedy) that the users can identify the presence of other users or programs and also he may interact with his own program while it is running.
* Processor’s time is shared among multiple users. An interactive or hands on computer system provides online communication between the user and the system.
* A time shared operating system uses CPU scheduling and multiprogramming to provide each user with a small portion of a time shared computer. Each user has at least one separate program in memory.
* A time shared operating system allows many users to share computer simultaneously.
* Since each action or command in a time shared system tends to be short, only a little CPU time is needed for each user

8)Distributed operating system.

A distributed operating system **(DOS)** is an essential type of operating system. Distributed systems use many central processors to serve multiple real-time applications and users. As a result, data processing jobs are distributed between the processors.

It connects multiple computers via a single communication channel. Furthermore, each of these systems has its own processor and memory. Additionally, these **CPUs** communicate via high-speed buses or telephone lines. Individual systems that communicate via a single channel are regarded as a single entity. They're also known as **loosely coupled systems**.