|  |
| --- |
| *Operating System – Introduction:* ~~Operating system objectives, User view, System view,~~ ~~Operating system definition~~, ~~Computer System Architecture~~, ~~OS Operations~~, Process Management, Memory Management, Storage Management, Protection and Security, Computing Environments.  *System Structures:* Operating System services, User and OS Interface, ~~System Calls, Types of System Calls~~, System Programs, Operating System Design and Implementation, OS Structure. |

WHAT IS AN OS?

A program which manages system hardware, provides basis for application programs and acts as a intermediatory b/w user and hardware.

* *System Hardware*
* *Intermediatory b/w user and hardware*

OS OBJECTIVES

An OS needs to fulfill the needs of both the user and the system for it to be a successful application. The objectives can be categorized into 2 sub categories:

* *USER VIEW*
* *SYSTEM VIEW*

USER VIEW

The user view focuses on providing a user-friendly interface and services that enable users to interact with the computer system efficiently. Here are some key aspects of the user view:

User Interface (UI):

* D-TRACE: Text / Shell commands
* Batch Interface: Shell commands in files (.sh files)
* GUI: Graphical User Interface

Application Support (Application Programming Interfaces):

* Libraries: Pre written functions and routines, used for common functionality tasks.
* Compilers & Dev tools: Tools that allow the programmer to write, compile and test code.
* Runtime Environment: Supports program execution

Convenience and Efficiency:

* Multitasking: Concurrent application running
* File Management: Organizes and manages files
  + - * File modification support: Create, update, del, move, copy
      * File permissions allow controlled access to protected files
* Copy-Paste: Copy paste functionality over applications
* Networking: Connect to networks, browse the web

Security and Privacy:

* USER AUTHENTICATION: mechanisms to ensure only authorized users can access the system
* ACCESS CONTROL: Restrict access based on current user
* ENCRYPTION: Encode data
* MALWARE PROTECTION: Protect against malicious threats

SYSTEM VIEW

Resource Management:

Managing system hardware resources

* CPU: Scheduling processes
* MEMORY: Manages mem resources
* STORAGE: Managing disks
* DEVICE: Interacts with I/O devices

System Security:

Managing system resources and allowing only authorized access (same as above)

* USER AUTHENTICATION: mechanisms to ensure only authorized users can access the system
* ACCESS CONTROL: Restrict access based on current user
* ENCRYPTION: Encode data
* MALWARE PROTECTION: Protect against malicious threats

Accounting:

Tacking and monitoring resource allocation

* REASOURCE USAGE & ALLOCATION:
* PERFORMANCE ANALYSIS:

COMPUTER SYSTEM ARCHITECTURE

A computer may be organized as

* Single Processor
* Multi-Processor

Multi v/s Single

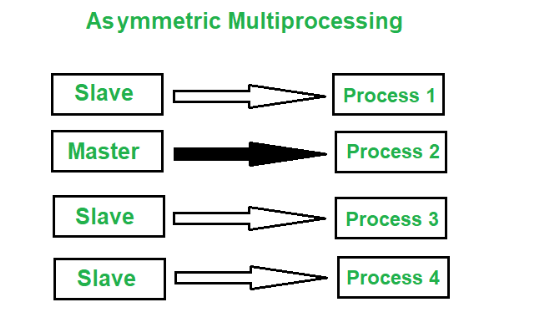
* Increased Throughput: Faster executions
* Economy of scale: Shared resources hence economical
* Increased Reliability: Failure at one processor will not halt system

Graceful Degradation

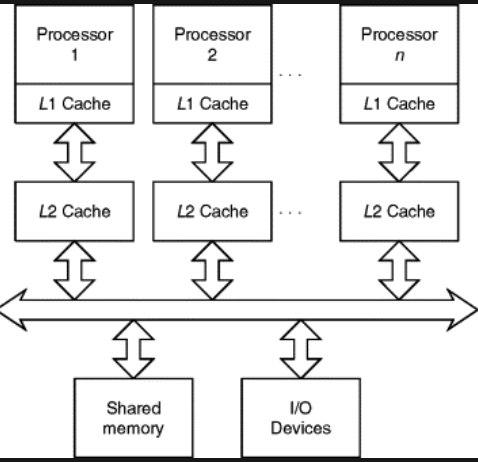
Ability to continue functioning in the presence of errors

Multi-Processor

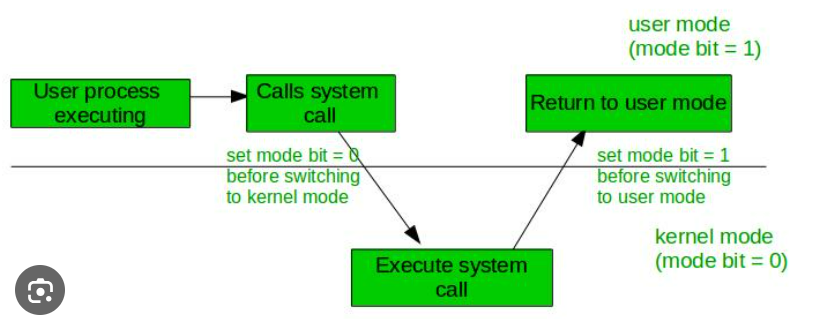
* Asymmetric
  + Mater Slave relationship
  + Master processor controls the system
  + Schedules and allocates work
  + No shared memory



* Symmetric
  + Equal level, peer processors
  + Share secondary mem
  + Each processor has its own registers and cache



OS OPERATIONS:



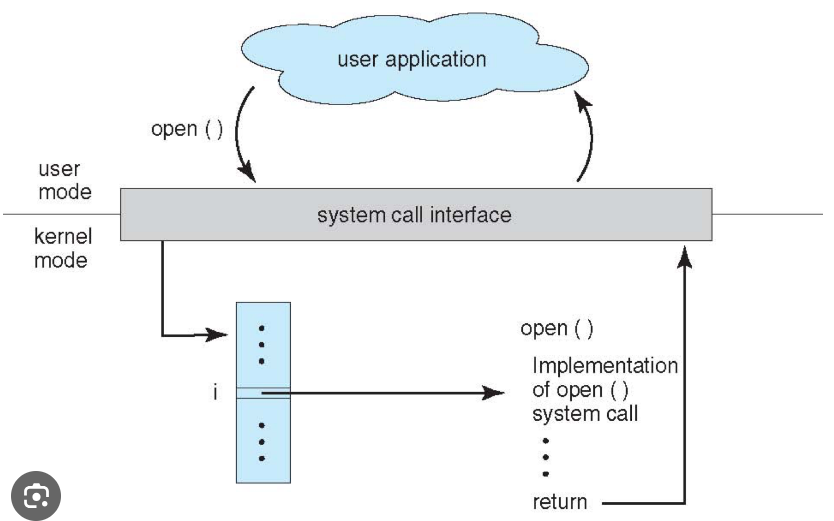
Two layers

* User
  + Mode bit = 1
  + User applications are run
* System
  + Mode bit = 0
  + System calls

SYSTEM CALLS

* The Interface/Bridge b/w user and system layers
* Generate interrupt to gain control of processor
* Available as routines in C C++ (open(), read())
* Accessed via API’s rather than direct system call use

Implementation



* Number associated with every sys call
* Invokes intended call and returns generated value

Parameter Passing:

Data being passed to the system call

* Registers(Simplest): Value stored as itself
* Pointers(Block method): Pointer to a block of mem, passing address of mem block
* Stack : Values pushed onto a stack

Types:

* Process Control
  + End, abort
  + Load, Execute
  + Create and terminate process
  + Wait for time
  + Wait event
* File management
  + Create and del file
  + Open, close
  + Read write reposition
  + Get and set file attributes
* Device Management
  + Request and release device
  + Read write reposition
  + Get and set device attributes
  + Attach detach devices
* Information Maintenance
  + Get and Set time or data,
  + Get and Set system data
  + Get and Set processes file or device attributes
* Communication
  + Create delete communication connection
  + Send receive messages
  + Transfer status info
  + Attach detach remote devices
* Protection
  + Set file security
  + Init & set security descriptor

QUESTIONS

True/False:

a) True or False: An operating system is not required for a single-user computer.

b) True or False: The main objective of virtual memory is to provide additional storage space for the CPU cache.

Fill in the Blanks:

a) The mode bit is used to switch between \_\_\_\_\_\_\_ mode and \_\_\_\_\_\_\_ mode.

b) The primary responsibility of the \_\_\_\_\_\_\_ is to manage the allocation and deallocation of memory resources.

Multiple Choice:

a) Which of the following is NOT an example of a real-time operating system?

i) Windows

ii) Linux

iii) VxWorks

iv) QNX

b) Which memory management technique allows processes to be divided into fixed-size blocks?

i) Paging

ii) Segmentation

iii) Swapping

iv) Fragmentation

True/False:

a) True or False: A system call is a mechanism that allows user applications to directly access hardware devices.

b) True or False: Round-robin scheduling guarantees fairness by assigning equal CPU time to each process.

Fill in the Blanks:

a) The process of converting a high-level programming language into machine code is performed by a \_\_\_\_\_\_\_.

b) The \_\_\_\_\_\_\_ is responsible for managing the communication between a user and the operating system.

The \_\_\_\_\_\_\_ mode is a privileged mode of operation in which the operating system kernel has complete control over the system resources.

The mode bit, also known as the \_\_\_\_\_\_\_ bit, is a flag in the CPU's status register that indicates whether the CPU is currently running in user mode or kernel mode.

A \_\_\_\_\_\_\_ is a well-defined interface provided by the operating system that allows user applications to request services from the kernel.

\_\_\_\_\_\_\_ are routines or programs provided by the operating system that perform specific system-related tasks, such as file management or system configuration.

The design and implementation of an operating system is typically guided by a specific \_\_\_\_\_\_\_ that defines the overall structure and organization of the system.

The \_\_\_\_\_\_\_ structure of an operating system refers to the services and functionalities provided by the operating system to user applications.

\_\_\_\_\_\_\_ are low-level software functions that allow user applications to interact with the operating system and request services or perform privileged operations.

In a multi-processor system, \_\_\_\_\_\_\_ allow multiple processors to work together, with each processor having its own specific tasks and responsibilities.

\_\_\_\_\_\_\_ management is the component of an operating system that is responsible for allocating and managing the computer's primary memory.

\_\_\_\_\_\_\_ management is the component of an operating system that deals with the allocation and management of secondary storage devices, such as hard disks or solid-state drives.