CSL 204 OPERATING SYSTEMS LAB

OS Lab Viva

Module-1

1.Definition of Operating System

2. Computer System divided into 3 components

Diagram(refer note)

1. Hardware (CPU, memory, input/output devices
2. Application programs (word processors, spreadsheets, compilers, web browsers)
3. Operating System

3. Operating System Operations

1.modern operating systems are interrupt driven.(h/w ans s/w interrupts)

2.

1.Dual-Mode and Multi-Mode Operation

- There are 2 seperate modes of operation
  User Mode
  Kernel Mode(Supervisor Mode, System Mode, Privileged Mode)
- 'mode bit' – is the bit added to the hardware of a computer to indicate the current mode, kernel (0) or user (1)

3. 2.Timer

Use

a. to ensure that the OS maintains control over the CPU.

b. not possible to allow a user program to get stuck in an infinite loop or fail to call system services

for these reasons, a timer is used.

4. Functions of OS

Process Management

Memory Management

Storage Management

file system management

mass storage management

caching

I/O Systems

Protection and Security

5. Operating System Services
   1.services for users
   2.services for system
6. Types of System Calls *
   System Calls can be grouped roughly into six major categories.

Process Control
- end, abort
- load, execute
- create/terminate process
- get process attributes, set process attributes
- wait for time
- wait event, signal event
- allocate & free memory

*

File Management
- create/delete file
- open, close
- read, write, reposition
- get file attributes/set file attributes

Device Management
- request device, release device
- read, write, reposition
- get device attributes, set device attributes.
- logically attatch or detach attributes

Information Maintenance
- get time or date, set time or date
- get system data, set system data
- get process, file or device attributes
- set process, file or device attributes

Communications
- create, delete communication connection.
- send, receive messages
- transfer status information
- attatch/detach remote device

Protection
set_permission() and get_permission()
allow_user() and deny_user()

7. Operating System Structure

Simple Structure

Layared approach

Microkernals

Modules

8. System Boot Process

- Booting procedure of starting a computer by loading the kernel.
- Bootstrap pgm/loader locate the kernel help it to load into the memory thus start the OS. It is a small piece of code. Hardware doesn't know how to load kernel into

VIVA - Module 2

Module 2

Difference between program & Process
2. Process states
3. Process State diagram
4. PCB stands for…
5. Explain PCB
6. System calls used in process creation and process termination
7. Define thread
8. Types of schedulers
9. Scheduling queues
10. Types of scheduling algorithm

Answers

Difference between Program & Process:

- A program is a set of instructions stored in a file, representing a software application.
- A process is an executing instance of a program in memory. It includes program counter, registers, and memory space.

2. Process States:
   - New: Process is being created.
   - Ready: Process is ready to run but waiting for CPU time.
   - Running: Process is currently being executed.
   - Blocked (or Waiting): Process is waiting for an event or resource.
   - Terminated: Process has finished its execution
3. Process State Diagram: A diagram illustrating the transitions between different process states and the events triggering those transitions.
4. PCB stands for…: Process Control Block.
5. Explain PCB:
   - PCB is a data structure containing information about a process.
   - It holds details like process state, program counter, CPU registers, process ID, scheduling information, memory pointers, and more.
6. System Calls for Process Creation & Termination:
   - Creation: fork() (in Unix-like systems), CreateProcess() (in Windows).
   - Termination: exit() (in Unix-like systems), ExitProcess() (in Windows).
7. Define Thread: A thread is the smallest unit of execution within a process. Multiple threads within a single process share the same memory space and resources.

8. Types of Schedulers:
    - Long-term scheduler (admits processes to the system),
    - Short-term scheduler (allocates CPU time to ready processes),
    - Medium-term scheduler (swaps processes in and out of memory for efficiency).
9. Scheduling Queues: Queues used by the scheduler to manage processes in various states, such as the ready queue, waiting queue, etc.

Types of Scheduling Algorithms:

- Preemptive: Scheduling decisions can be interrupted and changed.
- Non-preemptive: Once a process starts running, it continues until it completes.
- Algorithms include FCFS, SJF, Priority, Round Robin, and more

Module 3

1.What is Race conditions
2. Critical section problem
3. What are the conditions satisfied by solutions of critical section problem
4.Example for two process solution for critical section problem (Peterson's solution)
5. Synchronization hardware(Test and set, Swap)
6. Semaphores
7. Types of semaphore (binary, counting)
8. What is Monitors
9.Synchronization problems Producer Consumer, Dining Philosophers and Readers-Writers
10. Conditions for Deadlocks
11.Resource allocation graphs
12 Deadlock handling methods -(prevention, Deadlock avoidance , detection,ignore)
13.deadlock prevention(mutual exclusion, no preemption,hold and wait, circular wait?
14. Deadlock avoidance (1. For single instance resources -resources allocation graph, 2. For multiple instances -Banker's algorithms, 15.Deadlock detection(1. For single instance resources -wait for graph, 2. For multiple instances -Banker's algorithms,
16. Recovery from deadlock
Short answers of above topics

Answers

1. Race conditions occur when multiple processes or threads access shared resources concurrently, leading to unpredictable and undesired behavior.
2. The critical section problem involves coordinating multiple processes or threads to ensure that only one can access a shared resource at a time to avoid conflicts.
3. Solutions to the critical section problem must satisfy mutual exclusion, progress, and bounded waiting conditions.
4. Peterson's solution is a two-process solution to the critical section problem, which uses shared variables like "turn" and "flag" to ensure mutual exclusion.
5. Synchronization hardware includes instructions like "Test and Set" and "Swap," used to build low-level synchronization mechanisms.
6. Semaphores are synchronization primitives used to control access to resources, using atomic operations to manage concurrent execution.
7. Binary and counting are two types of semaphores. Binary semaphores have two values (0 and 1), while counting semaphores can have multiple values.

8. Monitors are high-level synchronization constructs that encapsulate shared resources and their associated synchronization mechanisms.
9. Producer-Consumer, Dining Philosophers, and Readers-Writers are synchronization problems involving managing access to shared resources among multiple processes.
10. Deadlock conditions involve mutual exclusion, hold and wait, no preemption, and circular wait.
11. Resource allocation graphs visually represent the relationships between processes and resources in a system, helping to analyze potential deadlocks.
12. Deadlock handling methods include prevention, avoidance, detection, and ignoring the issue.
13. Deadlock prevention involves addressing one of the four deadlock conditions (mutual exclusion, hold and wait, no preemption, circular wait).
14. Deadlock avoidance uses algorithms to ensure that the system remains in a safe state, like the Banker's algorithm for both single and multiple instances of resources.
15. Deadlock detection methods include constructing a wait-for graph for single instance resources and using the Banker's algorithm for multiple instances.
16. Recovery from deadlock involves terminating processes, preempting resources, or rolling back to a safe state to resolve the deadlock.

MODULE 4

1.difference between logical address and physical address
2. Swapping
3.Contiguous memory allocation,
4.Segmentation
5 Paging.
6.Virtual memory
7.Demand paging
8.Page replacement algorithms

Answers

1.Logical address refers to the address generated by the CPU, seen by processes as their memory locations. Physical address is the actual location in main memory where data is stored.

1. Swapping involves moving entire processes in and out of main memory to balance the number of processes with available memory, typically facilitated by the operating system.
2. Contiguous memory allocation assigns processes consecutive blocks of physical memory, requiring memory to be available in one contiguous chunk.
3. Segmentation divides memory into segments of varying sizes for different parts of a program, but fragmentation can occur due to uneven memory allocation.
4. Paging divides physical memory into fixed-sized blocks and logical memory into pages of the same size. It eliminates external fragmentation but can lead to internal fragmentation.
5. Virtual memory is a memory management technique where parts of a process's memory are stored in secondary storage, allowing processes to use more memory than physically available.
6. Demand paging brings in only necessary pages of a process into memory when they are required, reducing initial loading time and memory waste.
7. Page replacement algorithms decide which page to replace in memory when bringing in a new page. Examples include FIFO (First-In-First-Out), LRU (Least Recently Used), and Optimal algorithms

MODULE 5

1. File allocation methods
2. File access methods
3. Disk scheduling algms
4. Access control matrix (protection)
5. File attributes
6. Directory structures

Answers

1. File Allocation Methods: File allocation methods determine how files are stored on disk. Some common methods are contiguous allocation, linked allocation, and indexed allocation.
2. File Access Methods: File access methods define how data can be read from or written to files. Common methods include sequential access, direct access (random access), and indexed sequential access.
3. Disk Scheduling Algorithms: Disk scheduling algorithms manage the order in which read and write requests are serviced by the disk. Examples are FCFS (First-Come-First-Served), SSTF (Shortest Seek Time First), SCAN, and C-SCAN.
4. Access Control Matrix (Protection): An access control matrix is a security model that specifies the rights and permissions each user or process has on various objects (files, resources). It enhances system security by regulating access.
5. File Attributes: File attributes are metadata associated with files, such as file name, type, size, creation date, modification date, and permissions. They provide information about the file and its characteristics.

6. Directory Structures: Directory structures organize files into a hierarchical order for easy management. Examples include single-level directories, two-level directories, tree-structured directories, and acyclic-graph directories.