Operating System
Assignment 2

Q.1. Define inter process communication (IPC) and explain why it is essential in modern computing environments. Provide examples of scenarios where IPC plays a crucial role.

⇒ Why IPC is essential:-
* Enables processes to share resources like memory and files.
* Allows programs to work together on complex tasks.
* Supports parallel processing for better performance
* Helps processes coordinate their activities.

Key Scenarios:-
(i) Web Browsing: Browsers processes communicate with servers processes to load web pages.
(ii) Online Gaming: Multiple players interact through IPC for chat and game events.
(iii) Database Systems: Multiple processes communicate with servers processes access and update shared data concurrently.
(iv) Cloud Computing: Virtual Machines communicate across distributed servers.

Q.2. Write a C++ code snippet that demonstrate the use of Mutex locks to protect a shared resource in a multithreaded program. Explain how the mutex ensures mutual exclusion and prevents race conditions.

```
#include<iostream>
#include<thread>
#include<mutex>
int sharedCounter = 0;
std::mutex mtx;
void increment(int id){
    for(int i=0; i<1000; i++){
        mtx.lock();
        sharedCounter++;
        mtx.unlock();
    }
}
int main(){
    std::thread t1(increment,1);
    std::thread t2(increment,2);
    t1.join();
    t2.join();
    std::cout<<"Final Counter"<<sharedCounter
                <<std::endl;
    return 0;
}
```

How mutex works?

* Mutex has two states: locked and unlocked
* When a thread calls lock(), it gets exclusive access to the critical section
* Other threads wait until unlock is called.
* This prevents race conditions where multiple threads modify shared data simultaneously.
* Ensures only one thread modifies sharedCounter at a time, preventing data corruption.

Q.3. Describe the role of deadlock detection in modern operating systems and how it contributes to system reliability and performance.

⇒ Role of Deadlock Detection:
Deadlock detection identifies situations where processes are stuck waiting for each others resources, creating a circular wait condition. The system monitors resources allocation to find these deadlock situations.

Contribution to Reliability:
* Prevents system freezes and crashes
* Identifies blocked processes before complete system failure.
* Allows recovery actions like terminating processes or releasing resources.
* Ensures continuous system operation in critical applications.

Contribution to performance:
* Frees up locked resources quickly.
* Maintains optimal CPU and memory utilization.
* Prevents resource waste from permanently blocked processes.
* Reduces system downtime and improves throughput.

Modern operating systems use algorithms like resource allocation graphs to detect deadlocks and take corrective actions automatically.

Q.4. Explain the concept of schedulability analysis in real time scheduling. How does it determine if a set of tasks can meet their deadlines? provide an example of a schedulability analysis technique.

⇒ Schedulability analysis :-

Schedulability analysis determines whether a set of tasks can be scheduled to meet all their deadlines in real time system. It analyzes task parameters like execution time, period, and deadline to check feasibility.

How it works :-

* Examines each tasks execution time and deadline.
* Calculate total CPU utilization.
* Checks if scheduling algorithm can guarantee deadline complance.
* Provides mathematical guarantees before deployment.

Example : Rate Monotonic Analysis (RMA)

for tasks to be schedulable under RMS

$$U = \sum \frac{C_i}{T_i} \leq n\left(2^{1/n} - 1\right)$$

Where,

* C_i = Execution time of task i
* T_i = Period of task i
* n = Number of tasks

Simple Example :-

* Task 1 :- Execution = 2ms, period = 10ms
* Task 2 :- Execution = 3ms, period = 15 ms

Utilization : $U = 2/10 + 3/15 = 0.2 + 0.2 = 0.4$

Bound for 2 tasks : $(2^{0.5} - 1) = 0.828$

since $0.4 < 0.828$, tasks are schedulable.

Q.5. Describe the process of simulation as a technique evaluating scheduling algorithms. Explain how simulation models are constructed, and provide an example of a scenario where simulation would be useful in assessing scheduling algorithm performance.

⇒ Simulation Process:

Simulation creates a model of the system and runs tasks using different scheduling algorithms to evaluate performance.

Model Construction Steps:

(i) Create System Model: Define CPUs, memory, and I/O devices.

(ii) Define Tasks: Specify execution times, periods, and deadlines.

(iii) Implement Algorithm: Code the scheduling algorithm.

(iv) Design workloads: Create test scenarios (light, heavy, mixed loads)

(v) Collect Metrics: Measure completion time, deadline misses, resource usage

Useful Scenario: Video streaming servers

Problem: A streaming servers must handle multiple video qualities (SD, HD, 4K) simultaneously.

Why simulation helps:

* Tests different algorithms (Round-Robin, Priority Scheduling, EDF) without real deployment.
* Models peak and off-peak usage patterns.

* Compares buffering frequency and latency across algorithms
* Identifies optimal algorithm before actual implementation.
* Cost-effective compared to building physical test services.

Simulation reveals which algorithm provides the best user experience and resource utilization.

Q.6. Suppose you are deploying a multiplayer online game. Discuss how you would use a combination of shared memory and message passing to handle player interactions, such as chat message and in-game events, efficiently.

⇒ Hybrid Approach:-

(i) Shared Memory for Game State:
* Store player positions, health, and inventory in shared memory.
* Multiple servers thread read this data simultaneously.
* Provides ultra-fast access for real time updates.
* Use mutex locks to prevent conflicts during writes.
* Ideal for time critical operations requiring low latency.

(ii) Message Passing for chat:-
* Send chat message through message queues or sockets.

* Each message contains sender, recipient, timestamp and content
* Allows asynchronous delivery without blocking gameplay.
* Enables distribution across multiple chat servers
* Better for non critical communications.

Benefits:
* Performance: Shared memory handles fast game updates.
* Scalability: Message passing supports distributed architecture.
* Reliability: Guaranteed message delivery for chat.
* Efficiency: Minimal CPU overhead for frequent data access.

This combination provides both speed for gameplay and reliability for player communication

9.7. Discuss the advantages and disadvantages of using Rate-Monotonic Scheduling (RMS) in a real-time system. Provide real ~~world~~ examples where RMS would be beneficial and situations where it might not be ~~to~~ suitable.

=> Advantages:
(i) Predictable Execution: provides deterministic task scheduling.
(ii) Optimal Static Algorithm: Best among all fixed-priority algorithms.
(iii) Simple Implementation: Easy priority rule - shorter period = higher priority.

(iv) Mathematical Guarantees: Schedulability can be proven beforehead.
(v) Efficient Resource Use: Optimize CPU utilization for suitable tasks.

Disadvantages:
(i) Utilization Limit: Maximum 69% CPU usage for large task sets.
(ii) Requires Prior Knowledge: Needs task parameters in advance.
(iii) Only for Periodic Tasks: Doesnot handle aperiodic tasks well.
(iv) Priority Inversion: Can occur with resource sharing.
(v) Poor Overload Handling: Low-priority tasks may starve.

13/10/23

Benefical Examples:
~~(i) Beneficial Examples:~~
(i) Aircraft Control Systems:
* Navigation updates (50ms), flight control (20ms), sensors (10ms)
* Perfect for RMS due to predictable periodic tasks.
* Missing deadlines could be catastrophic
(ii) Pacemakers:
* Heart monitoring, pulse generation, battery check
* Fixed periods and critical timing requirements.
* Ideal for RMS reliability.