National University
of computer and emerging sciences

Project Report

Operating Systems

Prepared by:

1) Umer Farooq (22i-0518)

2) Arsalan Javed (22i-1632)

3) Saif Ur Rehman (22i-1697)

Concept Demonstration:

Our project of Ludo demonstrates several Operating System (OS) concepts, including:

Multithreading: The code uses multiple threads (PlayerT and Masterthread) to manage game logic and player moves concurrently. It is one of the fundamental concepts of OS.

```
loop while game is not finished
    for i from 0 to nbPlayer-1
        create thread Playthreads[i] with function playerT and parameter address of turn[i]

    for i from 0 to nbPlayer-1
        join thread Playthreads[i]
```

Synchronization and resource management: Binary Semaphores (m1) is used to ensure exclusive access to critical sections of code, preventing race conditions and data corruption. They are also used to efficiently share resources among multiple threads in order to avoid any sort of deadlock condition.

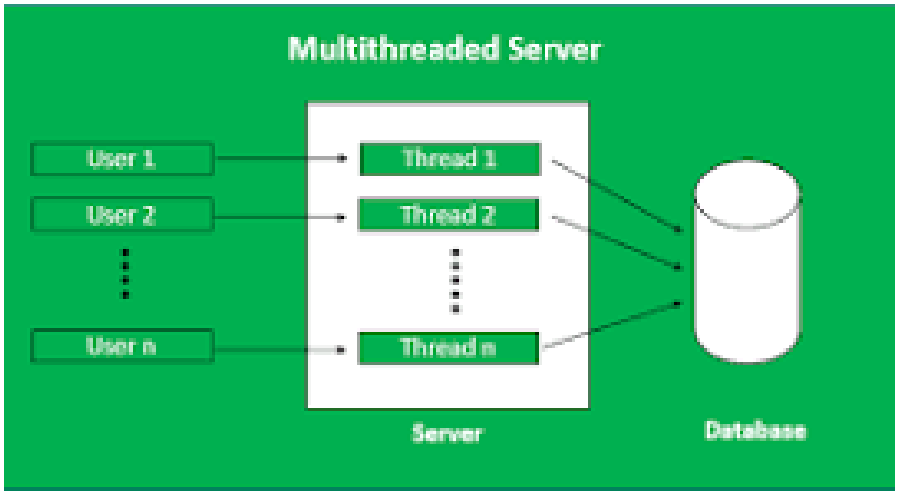
```
procedure playerT with parameter arg
    wait on semaphore m1 to enter critical section

    display "Turn : "
    for each index i in turn
```

Thread Management: Creation, execution, and termination of threads are managed explicitly.

Uses of Concepts:

These OS concepts can be applied to various scenarios, such as:

Server-Client Architecture: Multithreading and synchronization can be used to handle multiple client requests concurrently, ensuring efficient and safe data access. Avoiding Overwriting in send and receive buffers is a critical synchronization problem which is solved by using these concepts of OS.



Real-time Systems: Multithreading is critically involved in Realtime Systems to ensure timely execution of critical tasks and maintain system stability.

Distributed Systems: Synchronization and race condition avoidance is critically implemented in distributed systems such as in banking systems or in core networks to manage communication and resource sharing between nodes, ensuring efficient and reliable data exchange.

In summary, the OS concepts demonstrated in this code are essential for building efficient, scalable, and reliable systems in various domains, including server-client architecture, database management, real-time systems, distributed systems, and embedded systems. These concepts are critical to deal with non-linear systems and helps programmers design a sequence for task execution and maximizes the resource utilization.