OPERATING SYSTEMS

UNIT 3 SEMINAR
CONCURRENCY

- **1- You want to do the following operations**
    1. R = a + b
    2. S = R + c
    3. M = S + d

    So that the operation 1 and 3 are performed by a process and the operation 2 by a different one. How can we synchronize both processes?
- **2- We want to simulate the input to the computer room** using semaphores. If we know that the room has 20 seats, how can we do so that only 20 people to enter the room?
- **3-** We simulate with two processes the **waiting room of a dentist** and **the dentist** who is serving patients one by one. How can we synchronize?

Semaphores

*Also used **wait(s)** and **signal(s)** instead of P (s) and V (s) respectively*

- Devised by Dijkstra in 1965
- Structure with three atomic operations defined:
  - *Initialization, P, and V.*
- All operations are atomic.

No active standby . *The OS manages a queue of blocked processes*

```
P(s){
   s = s - 1;
   if (s < 0)
      block;
}
```

```
V(s){
   s = s + 1;
   if (s <= 0)
        wake up a process blocked on s;
}
```

The operating system implements these operations and offers them as services

Synchronization Mechanisms

Semaphores

Utility

1. **Solution to the critical section on mutual exclusion** *(Initializing S to 1)*

2. **Limiting the number of processes or threads concurrently accessing a critical section** (control units of a resource being used simultaneously) *(Initializing S to the number of resources)*

3. **Synchronization**

```
S = 1;
```

```
P(s);
   CRITCAL SECTION
V(s);
```

```
S = 4;
```

```
P(s);
   resource usage
V(s);
```

```
S = 0;
```

Thread 1	Thread 2
`A = x + y`	`P(s);`
`V(s);`	`print(A);`

Exercise

We want to create two processes named **Even** and **Odd**

- They alternately write numbers from 1 to 100 in an array
- The odd process writes 1, then even process writes 2 and so on…

- Write a pseudocode using semaphores for synchronize these operations

Exercise

Add a third process that prints on screen the contents of the array but synchronized with the previous two

The numbers should be printed one by one after they has entered the array 1, 2, 3…

- Write a pseudocode using semaphores for synchronize the operations

Exercise

- Modify the previous solution so that now we use pipes to communicate the processes
  - Replace the array with a pipe

Exercises

- Solve the previous problem using pipes instead of semaphores to synchronize

Exercise

Now the processes are executed on separate machines, each in a different one. The Producer processes must send the generated value to the consumer using messages. The consumer will print the values as they arrive.

- Solve the previous problem using message passing to synchronize and communicate