Varun Verma
CS → B1 (Roll no - 38)
1801010170

Operating System Assignment - 1

Question / Answer

Q1 → Explain the principle of concurrency.

Ans → Concurrency is the execution of several instruction sequences at the same time (Many threads running in Parallel). In an operating system, this happens when there are several process threads or functions running in parallel. These threads may communicate with each other through shared memory.

In concurrent programming, programmers codes and executes several simultaneous executing processes / threads that can be addressed individually. In a concurrent system, the coder need to see for deadlock situation since many threads are running in parallel.

Q2 → What is the used of inter process communication.

Ans → Inter process communication (IPC) is

used for exchanging data between multiple threads in one or more processes or programs. The Processes may be running on single or multiple computers connected by a network.

It is a set of programming interface which allow a programmer to coordinate activities among various program processes which can run concurrently in an operating system. This allow a specific program to handle many user request at the same time.

Since every single user request may result in multiple processes running in the OS, the process may require to communicate with each other.

Q3) Explain the need of process synchronization How can the interprocess communication be achieved.?

Ans. Process synchronization is a way to coordinate processes that uses shared data. It occurs in an operating system among co-operating processes. While executing many concurrent processes, process synchronization helps to maintain shared data

consistency and cooperating process execution. Processes have to be scheduled to ensure that concurrent access to shared data does not create inconsistencies. Data inconsistency can result in what is called a race condition.

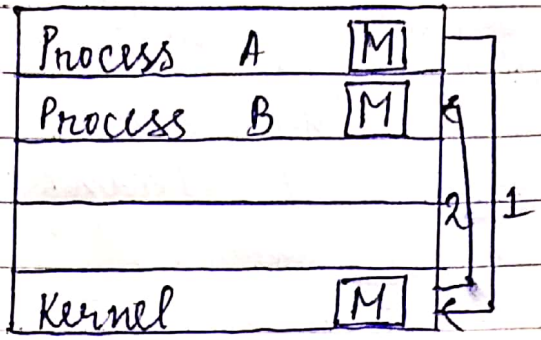
There are various methods by which IPC can be achieved, two of them are explained below :-

Shared memory → Shared memory share memory between two or more processes that are established using shared memory between all the processes. This type of memory requires to protected from each other by synchronizing access across all the process.

Message Passing → It is a mechanism for a process to communicate and synchronize using message passing, the process communicates with each other without resorting to shared variables.

Q4) Define Message passing interprocess communication.

Ans → In message passing model communication take place by means of message exchanged between the cooperating process.



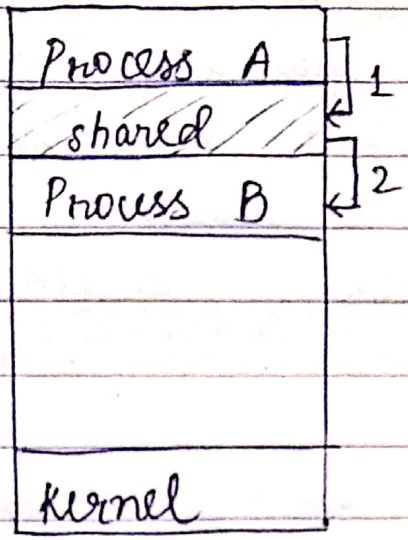
Message passing is the basis of most interprocess communication in distributed system. It is at the lowest level to abstraction and requires the application programmers to be able to identify

→ Message
→ Name of source
→ Destination process
→ Data types expected for process.

Q5) Define shared memory interprocess communication

Ans → In the shared memory model, a region of memory that is shared by cooperating processes is established. Process can then

exchange information by reading and writing data to the shared region.



shared memory is a implementation for IPC where a memory section is shared between different processes.

Q6) State the producer-consumer problem.

Ans → A producer process produces information that is consumed by consumer process.

Example → a compiler may produce assembly code which is consumed by an assembler.

→ One solution to the producer-consumer problem is that to use shared memory.

→ To allow producer and consumer process to run concurrently. We must have a

a buffer that reside in a region of shared memory.

→ a producer can produce one time while the consumer is consuming another time. items. The producer and consumer must be synchronized so that the consumer does not try to consume an item that has not yet been produced.

Q7) Explain how message passing can be used to solve to buffer Producer/Consumer problem with infinite buffer

Ans. → There are mainly three problems in the producer consumer problem listed below :—

i) consumer cannot consume the data if the buffer is empty.

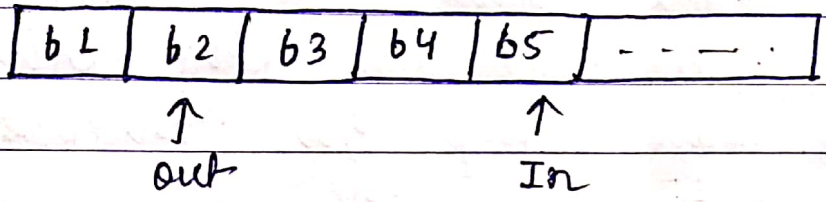
ii) Producer cannot produce the data if the buffer is full.

iii) Producer and consumer cannot produce or consume data at the same time because it will lead to data inconsistency and race condition may occur.

Solution of producer consumer problem with infinite buffer :—

1.) Let us first consider the case in which the buffer is infinite.

2.) Figure shows two pointers, in and out, which are used respectively indicating the next space for producer to put a newly created product and the next product that is available for consumers to consume.



3.) Instead of two pointers, in and out, an integer variable, n is used to indicate the number of products available for consumption and a set of functions are provided for convenience : produce(), consume(), append(), and take().

4.) The semaphore s is used to enforce the mutual exclusion, the semaphore delay is used to force the consumer to wait if the buffer is empty.

Q8) What is race condition?

Ans → A situation where several processes access & manipulate the same data concurrently & the final value of the shared data depends upon which process finishes ~~first~~ last. To prevent race condition concurrent process must be synchronized.

It occurs when two or more threads can access shared data and they try to change it at the same time. Because the thread scheduling algo can swap b/w threads at any time, you don't know the order in which the threads will attempt to access the shared data.

Example → A Kernel data structure that maintains a list of all open files in the system. This list must be modified when a new file is opened or closed. If two processes were to open file simultaneously, the separate updates to this list could result in a race condition.

Q9) Differentiate between Message passing and shared memory interprocess communication

Ans → Both message passing and shared memory are models of inter process communication

whose differences are mentioned below :-

Message Passing	Shared Memory
1) It is useful for exchanging smaller amounts of data.	1) It can be useful for exchanging large amount of data.
2) Easy implementation	2) Difficult to implementation
3) Message passing is slower than shared memory.	3) Shared memory is faster than message passing
4) No info./data exchange will be direct but through the OS/Kernel.	4) No info./data exchange will be direct but through the shared memory.
5) OS interaction	5) Limited OS interaction

Q10) Write the code of Producer & Consumer process.

* Code for Producer Process

```
while (true)
{
    /* produce an item in next Produced */
    while (counter == BUFFER_SIZE)
```

```
        ; /* do nothing */
        buffer[in] = nextProduced;
        in = (in + 1) % BUFFER_SIZE;
        counter++;
    }
```

* Code for Consumer Process

```
while (true)
{
    while (counter == 0)
    ; /* do nothing */
    nextConsumed = buffer[out]
    out = (out + 1) % BUFFER_SIZE;
    counter--;
    /* consume the item in nextConsumed */
}
```