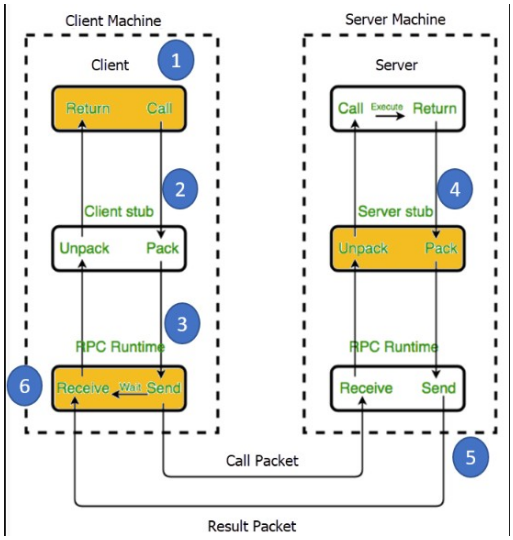
MICHAEL ORINA
SCII/00825/2019

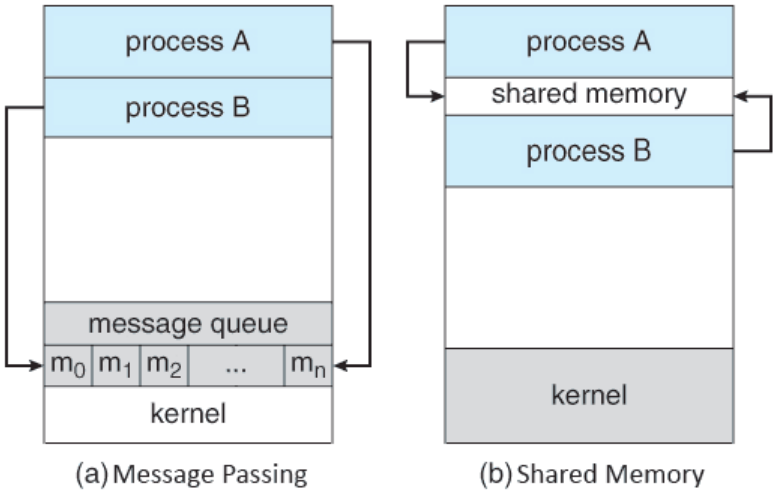
With the aid of diagrams, explain the following:

1. **Remote Procedural Call (RPC)**



Remote Procedure Call (RPC) is a protocol that allows a computer program to make a request to another program running on a different computer, without having to understand the network's details. The requesting program appears to the called program as if it runs in the same machine. It is a method for making a computer program run a function on a remote computer as if it was running on the local computer. It allows for different parts of a software system to be located on different computers and communicate with each other seamlessly, as if they were all running on the same machine. The remote computer receives the request, executes the function, and returns the result to the requesting computer, which can use it as if the function was executed locally. This enables the creation of distributed systems, where different parts of a system can run on different machines, but appear to the end user as a single, integrated system.

2. **Inter-Process Communication**



In computer science inter-process communication or interprocess communication (IPC) refers specifically to the mechanisms an operating system provides to allow the processes to manage shared data. Typically, applications can use IPC, categorized as client and servers where the client

requests data and the server responds to client requests. Many applications are both clients and servers, as commonly seen in distributed computing

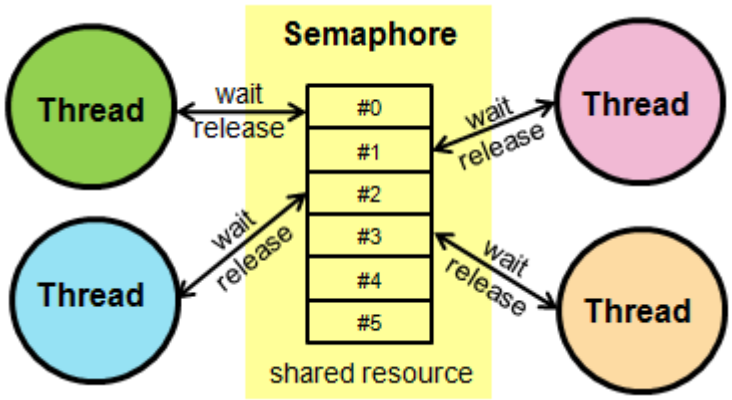
IPC is very important to the design process for microkernels and nanokernels, which reduce the number of functionalities provided by the kernel. Those functionalities are then obtained by communicating with servers via IPC, leading to a large increase in communication when compared to a regular monolithic kernel. IPC interfaces generally encompass variable analytic framework structures. These processes ensure compatibility between the multi-vector protocols upon which IPC models rely.

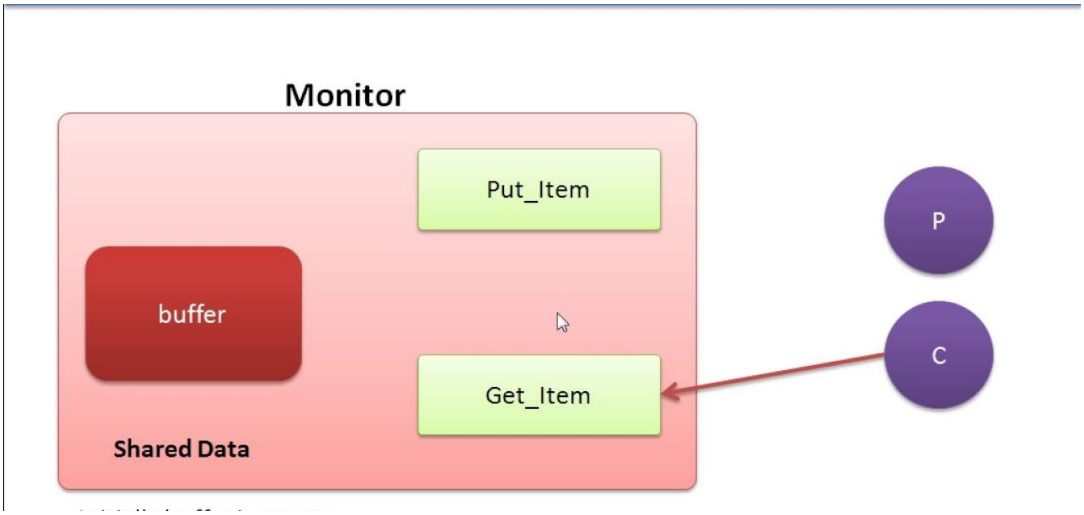
An IPC mechanism is either synchronous or asynchronous. Syncronization primitive may be used to have synchronous behavior with an asynchronous IPC mechanism.

3. Semaphores and Monitors

Semaphore and Monitor both allow processes to access the shared resources in mutual exclusion. Both are the process synchronization tool. Instead, they are very different from each other. Where Semaphore is an integer variable which can be operated only by wait() and signal() operation apart from the initialization.

On the other hand, the Monitor type is an abstract data type whose construct allow one process to get activate at one time. In this article, we will discuss the differences between semaphore and monitor with the help of comparison chart shown below.




Monitor
Put_Item
buffer
Get_Item
Shared Data
P
C