OPERATING SYSTEM
IT-41033

CHAPTER – 3
Q&A

3.1 What are the benefits and the disadvantages of each of the following? Consider both the system level and the programmer level.

a. Synchronous and asynchronous communication

b. Automatic and explicit buffering

c. Send by copy and send by reference

d. Fixed-sized and variable-sized messages

a) synchronous and asynchronous communication - A benefit of synchronous communication is that it allows a rendezvous between the sender and receiver. A disadvantage of a blocking send is that a rendezvous may not be required and the message could be delivered asynchronously; received at a point of no interest to the sender. As a result, message-passing systems often provide both forms of synchronization.

b) Automatic and explicit buffering - Automatic buffering provides a queue with indefinite length; thus ensuring the sender will never have to block while waiting to copy a message. There are no specifications how automatic buffering will be provided; one scheme may reserve sufficiently large memory where much of the memory is wasted. Explicit buffering specifies how large the buffer is. In this situation, the sender may be blocked while waiting for available space in the queue. However, it is less likely memory will be wasted with explicit buffering.

3.1 (Cont …)

c) Send by copy and send by reference - Send by copy does not allow the receiver to alter the state of the parameter; send by reference does allow it. A benefit of send by reference is that it allows the programmer to write a distributed version of a centralized application. Java's RMI provides both, however passing a parameter by reference requires declaring the parameter as a remote object as well.

d) Fixed-sized and variable-sized messages - The implications of this are mostly related to buffering issues; with fixed-size messages, a buffer with a specific size can hold a known number of messages. The number of variable-sized messages that can be held by such a buffer is unknown. Consider how Windows 2000 handles this situation: with fixed-sized messages (< 256 bytes), the messages are copied from the address space of the sender to the address space of the receiving process. Larger messages (variable-sized messages) use shared memory to pass the message.

3.2 Consider the RPC mechanism. Describe the undesirable consequences that could arise from not enforcing either the "at most once" or "exactly once" semantic. Describe possible uses for a mechanism that has neither of these guarantees.

If an RPC mechanism could not support either the “at most once” or “exactly once” semantics, then the RPC server cannot guarantee that a remote procedure will not be invoked multiple occurrences. Consider if a remote procedure were withdrawing money from a bank account on a system that did not support these semantics. It is possible that a single invocation of the remote procedure might lead to multiple withdrawals on the server. For a system to support either of these semantics generally requires the server maintain some form of client state such as the timestamp. If a system were unable to support either of these semantics, then such a system could only safely provide remote procedures that do not alter data or provide time-sensitive results. Using our bank account as an example, we certainly require “at most once” or “exactly once” semantics for performing a withdrawal (or deposit) However, an inquiry into an account balance or other account information such as name, address, etc. It does not require these semantics.

3.4 Palm OS provides no means of concurrent processing. Discuss three major complications that concurrent processing adds to an operating system.

a) A method of time sharing must be implemented to allow each of several processes to have access to the system. This method involves the preemption of processes that do not voluntarily give up the CPU and the kernel being reentrant.

b) Processes and system resources must have protections and must be protected from each other. Any given process must be limited in the amount of memory it can use and the operations it can perform on devices like disks.

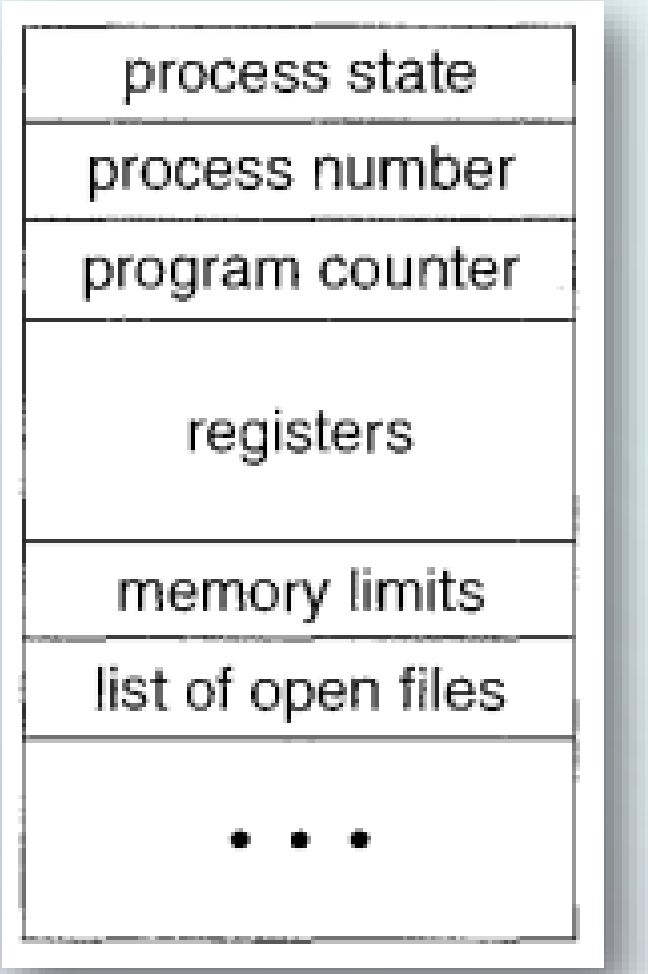
c) Care must be taken in the kernel to prevent deadlocks between processes, so processes aren't waiting for each other's allocated resources.

Q Draw and explain process control block in detail.

Each process is represented as a process control block (PCB) in the operating system. It contains information associated with specific process.

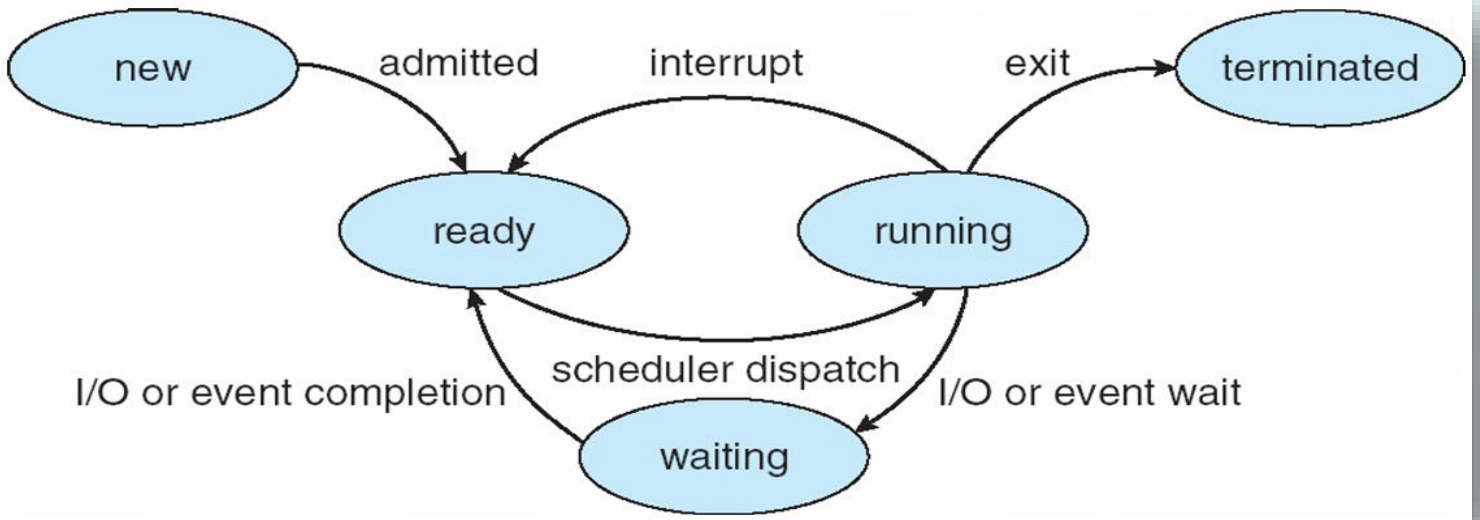
1. Process Number: Each process is identified by its process number; called process identification number (PID).
2. Priority: Each process is assigned a certain level of priority that corresponds to the relative importance of the event that it services.
3. Process State: This information is about the current state of the process. i.e, whether process is in new, ready, running, waiting or terminated state.
4. Program Counter: This contains the address of the next instruction to be executed for this process.
5. CPU Registers: CPU registers vary in number and type, depending upon the computer architectures. These include index registers, stack pointers and general purpose registers etc. When an interrupt occurred, information about the current status of the old process is saved in registers along with the program counters. This information is necessary to allow the process to be continued correctly after the completion of an interrupted process.
6. CPU Scheduling Information: This information includes a process priority, pointers to scheduling queues and any other scheduling parameters.
7. Memory Management Information: This information may include such information as the value of base and limit registers, the page table or the segment table depending upon the memory system used by operating system.
8. Accounting: This includes actual CPU time used in executing a process in order to charge individual user for processor time.
9. I/O Status: It includes outstanding I/O request, allocated devices information, pending operation and so on.
10. File Management: It includes information about all open files, access right etc.

Q Draw and explain process control block in detail. (Cont…)



Process Control Block (PCB)

Q Draw process state diagram and describe each state.



Process State Diagram

Q Draw process state diagram and describe each state. (Cont…)

New – The process being created is available in the new state. It is the new state because the system is not permitted it to enter the ready state due to limited memory available in the ready queue. If some memory becomes available, then the process from the new state will go to ready state.

Ready State – The process which is not waiting for any external event such as I/O operation and which is not running is said to be in ready state. It is not in the running state because some other process is already running. It is waiting for its turn to go to the running state.

Running State – The process which is currently running and has control of the CPU is known as the process in running state. In single user system, there is only one process which is in the running state. In multiuser system, there are multiple processes which are in the running state.

Blocked State – The process that is currently waiting for external event such as an I/O operation is said to be in blocked state. After the completion of I/O operation, the process from blocked state enters in the ready state and from the ready state when the process turn will come it will again go to running state.

Terminated State – The process whose operation is completed, it will go the terminated state from the running state. In halted state, the memory occupied by the process is released.

Q Explain Process.

A process is a program in execution. Process is called as job, task or unit of work. The execution of a process must progress in a sequential fashion. Process is an active entity.

When a program is loaded into the memory and it becomes a process, it can be divided into four sections – stack, heap, text and data.

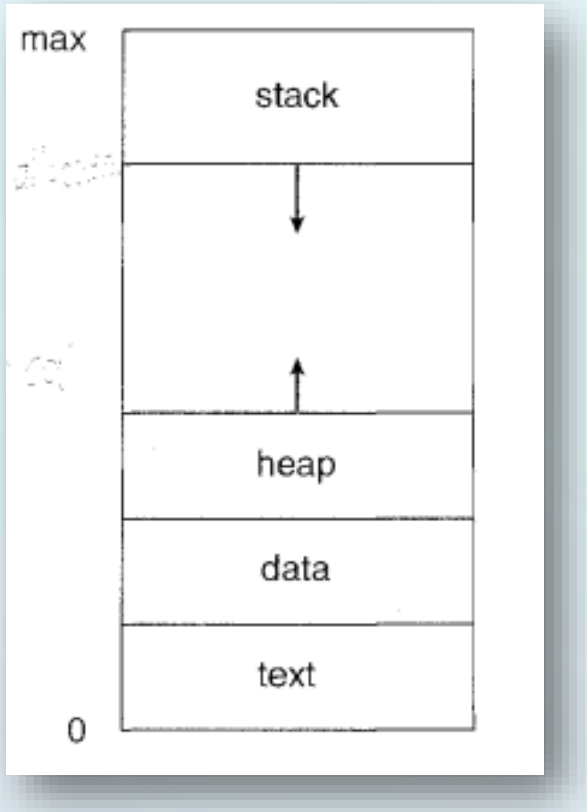
Text section – it is made up of the compiled program code, read in from non-volatile storage when the program is launched.

Data section – it is made up the global and static variables, allocated and initialized prior to executing the main.

Heap – it is used for the dynamic memory allocation, and is managed and calls to new, delete, malloc, free, etc.

Stack – it is used for local variables. Space on the stack is reserved for local variables when they are declared.

Q Explain Process. (Cont…)



Process in memory

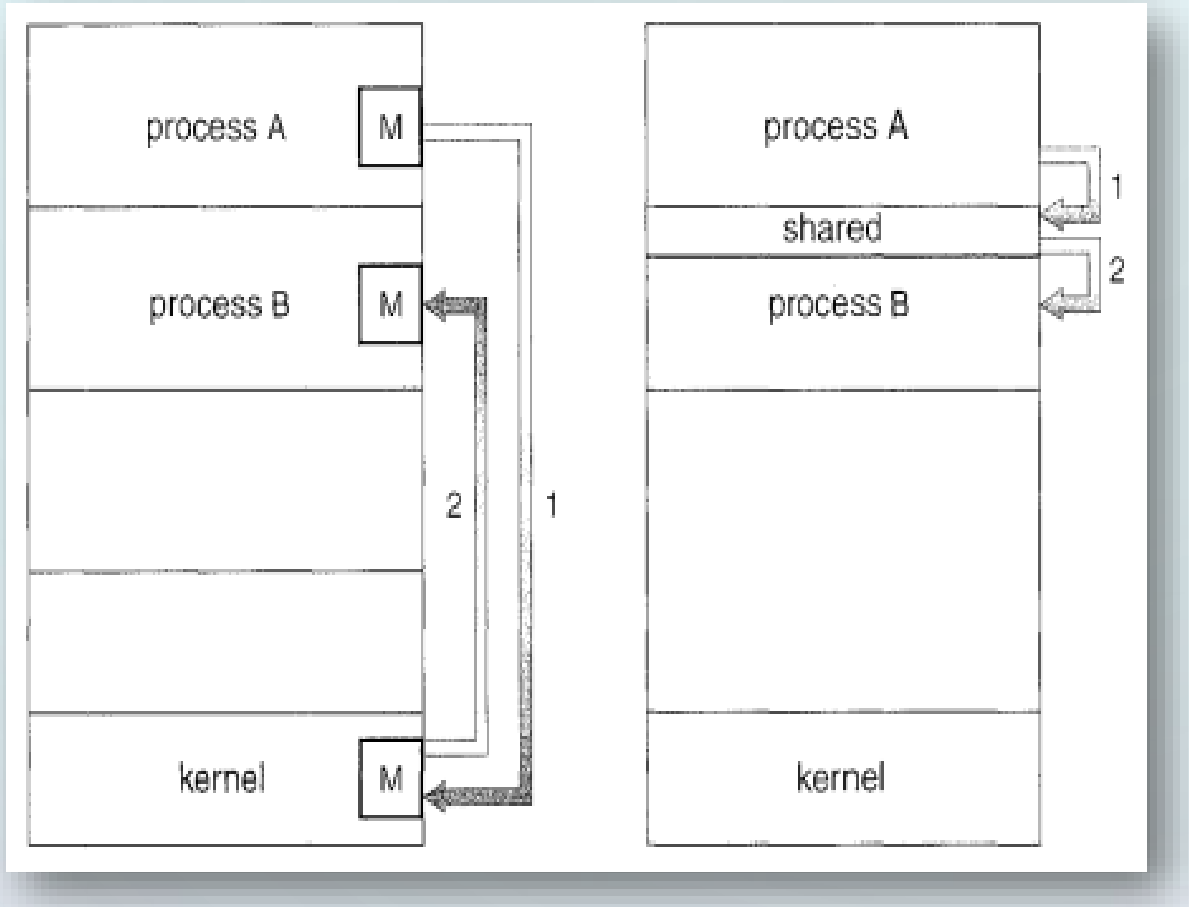
Q Draw and explain Inter-Process Communication model

Inter-Process Communication – cooperating processes required an Inter-Process Communication (IPC) mechanism that allow them to exchange data and information. There are two models of IPC.

Shared Memory – a region of the memory residing in an address space of a process creating a shared memory segment can be accessed by all processes who want to communicate with other processes. All the processes using the shared memory segment should attach to the address space of the shared memory. All the processes can exchange information by reading or writing data in shared memory segment. The form of data and location are determined by these processes who want to communicate with each other. These processes are not under the control of the operating system. The processes are responsible for ensuring that they are not writing to the same location simultaneously. After establishing shared memory segment, all accesses to the shared memory segment are treated as routine memory access and without assistance of kernel.

Message Passing – communication takes place by exchanging message between cooperating processes. It allows processes to communicate and synchronize their action without sharing the same address space. It is useful in a distributed environment when communication process resides on a different computer connected by a network. Communication requires sending and receiving message through the kernel. Between each pair of processes exactly one communication link.

Q Draw and explain Inter-Process Communication model. (Cont…)



Message passing

Shared memory

3.9 Describe the differences among long-term and medium-term scheduling.

No.	Long Term Scheduling	Medium Term Scheduling
1	Long term scheduler selects a process from job pool and loads into the memory	Medium term scheduler selects a process from swapped queue and loads into the memory
2	It works with job pool and memory	It works with swapped process queue and memory
3	It is a job scheduler	It is a swapped process scheduler
4	Processes switch its state from new to ready. A process selected from job pool enters to the memory for the first time.	Processes switch its state from blocked to ready. A process selected from swapped queue reenters into the memory.
5	It controls the degree of multiprogramming	It does not the degree of multiprogramming.