CS 303 - Operating System Concepts (3 credit).

Assignment 02. S/17/355

1. Define the term "process" in Computer applications?

process is a program in execution and can be Considered as a unit of work in a modern time-sharing system.

2. what is the difference between a process and a program?

* program is a passive entity stored on a disk (executable file), process is a active entity.

* program becomes a process when the executable file is loaded into the memory.

- one program can be several processes - when Considering multiple users executing the same program.

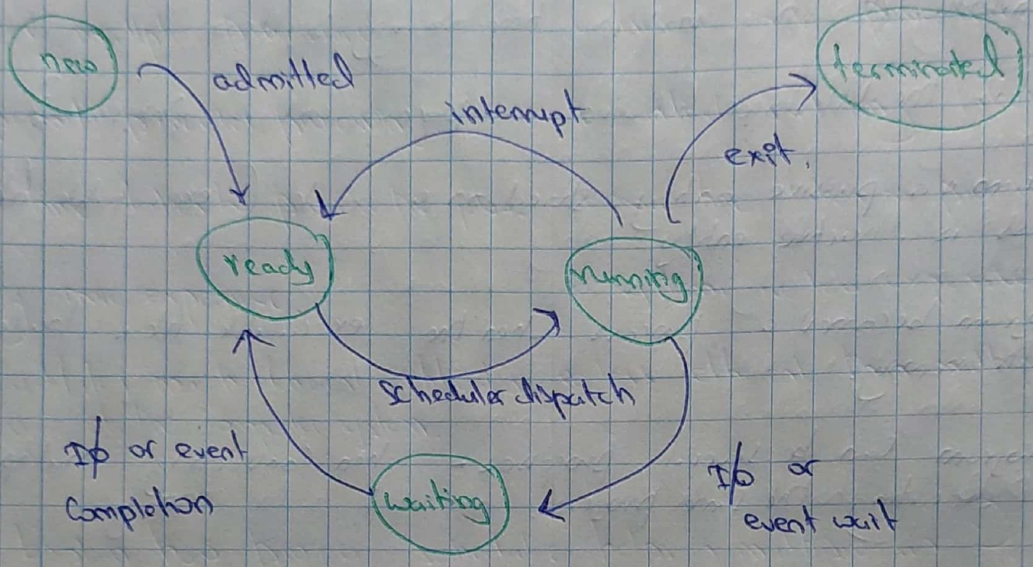
3. Briefly discuss the difference between a Job and a process?

* Batch systems execute "Jobs" while time shared systems has user programs or tasks", However term Job and process are used almost interchangeably in the text.

4 process memory is partitioned into 4 Sections. state and discuss each of them briefly.

- Text section - The program Code.
* program Counter, processor registers.
- Stack - Contain temporary data (function parameters, local variables, return addresses).
- Heap - Containing memory dynamically allocated during run time.
- Data section - Contain global variable.

5. The following diagram illustrated the various state a process can be in. draw arrows connecting each pair of states that a preemptive OS may move a process between. label each arrow with a brief brief description of a situation where the OS would move the process as indicated.



6. Explain why the OS would move a process in the running state to a waiting state?

- Some process with an higher priority is existing in the process queue
- Process need to perform I/o Operations.
- Seek time of the process is over.

- Current process need some resources that is acquired by some other process so it goes for waiting state

7. Discuss a situation where the OS would move a process from waiting state to a ready state.

- The I/o operation or event for which the process was waiting get Completed.

8. list down the typical content associated with a process control block? (PCB)

- process state.
- process number
- program counter.
- registers.
- memory limits.
- list of open files.

9. Briefly explain the purpose of having PCBs in process management

- process control block (PCB) is a data structure maintained by the operating system to store information of each process.
- Role of the PCB is to identify each process so that operating system can easily distinguish between processes.
- PCB contains fields like process ID, process priority, process state, accounting information, list of open files etc.
- PCBs are stored in the form of linked list in the memory.
- Process table contains a reference to every process currently being executed in the system. So whenever OS perform context switching it refers to the process table for the required process.

10. What are the advantages of Multi-programming?

- objective of multi-programming is to have some process running at all times to maximise CPU utilization.
- objective of time sharing is to switch the CPU among processes so frequently that users can interact with each program while it is running.
- CPU is used most of time and never become idle.
- The system looks fast as all the tasks run in parallel.
- Short time jobs are completed faster than long time jobs.
- multiprogramming systems support multi users.
- Resources are used nicely.
- Response time is shorter.

11. Tabulate the difference between multi-processing and Multiprogramming.

multiprocessing	multi programming
* multiprocessing refers to processing of multiple processes at same time by multiple cpu's.	• multi programming keeps several programs in main memory at the same time and execute them concurrently using a single cpu.
• utilizes multiple processors.	• uses a single processor.
• It permits parallel processing.	• Context switching takes place.
* less time taken to process the jobs.	• more time taken to process the jobs.
• usually more expensive.	• Systems are less expensive.
* It facilitates much efficient utilization of devices of the Computer system.	* less efficient than multiprocessing.

12. Explain how process scheduler helps in multiprocessing?.

* process scheduling is an essential part of multi-programming operating systems, allows more than one process to be loaded onto the executable memory at a time and the loaded process shares the cpu using time multi-plexing.

13 Explain Scheduling Queues?.

• maintains scheduling queues for processes.
(types of scheduling Queues).

- Job Queue - Set of all processes in the System
  • as process enter the system, They are put into a Job Queue.
- Ready Queue. - set of all processes residing in main memory and are ready and waiting to execute.
  • generally stored as linked list.
  • Each PCB includes a pointer field that points to the next PCB in the ready Queue
- Device Queue - Set of all processes waiting for an I/o devices.
  * each device has it's own device Queue.

14. Define the Terms "Short term Scheduler" and "long-term scheduler" and clarify the differences between the two.

Short term scheduler (or Cpu Scheduler)

- selects which process should be executed next and allocates cpu.
- Sometimes the only Scheduler in the System.
- Short term scheduler is invoked in (millisecond).

long term scheduler (or Job scheduler).

- selects which processes should be brough into the ready Queue.
- long term scheduler is invoked infrequently (seconds, minutes).
- long term Scheduler Control the degree of multiprogramming.

15. "The system with the best performance will have a Combination of Cpu bound and IO bound Processes" Justify the Statement.

There are mainly 2 types of processes.

1. I/o bound processes - spends more time doing I/o than Computation, many short Cpu burst.

2. Cpu bound processes - Spends more time doing Computations, few very long Cpu bursts.

- a good Scheduler Strives for a good mix of the above two typed ~~because while the processor~~. It is important to distinguish between I/o bound and Cpu bound processes to give an optimal usage of System resources.

let's say all the jobs are Cpu bound, then I/o waiting queue will be almost empty and if all jobs are I/o bound ready Queue will be empty. There by a mix of Cpu bounded and I/o bound is neccessary for optimal system utilization.

16. what is Context switching?

- when Cpu switches to another process, the system must save the state of the old process and load the saved state for the new process via a Context switch. The Context of a process is represented in the pCB.

17. Explain the term "Cascading termination" in process operations.

* Some systems do not allow a child to exist if its parent has terminated. In such systems, if a process terminates (either normally or abnormally), then all of it's children must also be terminated. This phenomenon, refered to a cascading termination is normally initiated by the OS.

18. State three techniques in which processes on the same processor can communicate with each other. If any of the techniques requires hardware support to achieve communication, explain?

Shared memory.

- multiple processes can communicate with each other simultaneously by accessing or sharing the memory. The multiple processes can exchange the data using read/write to shared memory because the OS creates a common memory in RAM for sharing. It requires protection by the sychronization of access of all multiple processes in the communication. It is a more efficient method and shares the data very quickly. To understand this consider the 2 processes as process 1 and process 2. If one process creates memory, then the other process will access it by means of sharing.
The information generated by process 1 about a particular resource and stored the record in shared memory. If process 2 want to use this information it checks the stored record in shared memory, takes the information generated by process 1 and processes it accordingly. Thus processes 1 and 2 can utilize shared memory to retrieve information such as records from such as record from other processes and transfer certain information to other processes.

Advantages.

- The communication process is very fast and bidirectional.
- Multiple processes can use the shared memory.
- Resources are saved.

disadvantages.

- Required concurrency control.
- No protection of data.
- inconsistency of data occured as update was lost.

Message passing.

- It is the other method of inter processing Communicating for synchronization and Communication between the processes. It is very slow and easy to implement using system calls Compared to shared memory method. The processes can exchange the data each other without using any shared resources or variables.

In message passing, the Communication between the processes is performed through Communication link. The operations performed by the processes are sending the message and receiving the message. Since the message might be variable or fixed type.

process 1 and process 2 Communicated together by establishing the Communication link as shown in the figure above. After that, they exchange the message through send and receive operations.

(19) "Messages passing is typically faster than shared memory".

NO. Shared memory allows maximum speed and Convenience of Communication, as it Can be done at memory speeds when within a Computer. Shared memory is faster than message passing as message-passing systems are typically implemented using system calls and thus require more time Consuming task of kernel intervention.

(20). discuss the difference between direct Communication and indirect Communication in message passing (message based) systems.

- Direct Communication.
- processes must name each other explicitly.
  send (p, message) - send a message to process p.
  receive (Q, message) - Receive a message from process Q.

properties of Communication link

- links are established automatically.
- link is associated with exactly one pair of Communication process.
- Between each pair there exist exactly one link.
- the link may be uni-directional, but usually is bidirectional.

Asymmetric in addressing.

- processes must name each other explicitly.
  Send (P, message) - send a message to process P
  receive (id, message) - Receive a message from any process)

Indirect Communication.

- messages are directed and received from mailboxes (also refered to as ports)

  - each mailbox has a unique id.
  - processes can communicate only if they share a mailbox.

Properties of Communication link.

- link established only if processes share a Common mailbox.
- a link may be associated with many processes.
- each pair of processes may share several Communication links.
- link may be uni-directional or bi-directional.

operations.

- create a maibox (port)
- send or receive messages through mailbox.
- destroy a mailbox.

primitives are defined as:

Send (A, message) – Send a message to mailbox A.
Receive (A, message) – Receive Message from mail box A.

mailbox sharing.

- suppose that processes P_1 and P_2 and P_3 all share mailbox A.
- P_1 sends, P_2 and P_3 receive.

who get the message.

Solutions.

- Allow a link to be associated with at most two processes.
- Allow only one process a time to execute a receive operation.
- Allow the system to select a arbitary the receiver, sender is notified who the receiver was.