1a) False. A ready process waiting to get access to CPU is in the ready state

A process in the waiting state is waiting for an event or the completion of an I/O operation

b)True. Each PCB has a pointer to other PCBs in a queue and processes in the ready queue are in the ready state

The short term scheduler uses this queue to schedule processes. The head of the queue contains PCB of the process currently in running state

c) False. The wait() system call causes the parent to wait until all child processes terminate

Also called join(), reference to fork join processing model. The parent forks several child processes for independent computations and combines their results using join()

SC2005 Operating Systems

**TUTORIAL TWO**

**Processes and Threads**

1. Indicate whether the following statements are true or false. Justify your answers.

   a) A ready process waiting to get access to the CPU is in the "waiting" state.

   b) A ready queue is a queue of Process Control Blocks (PCBs) of all processes in the "ready" state.

   c) The "wait()" system call is generally used by a child process to wait for instructions from a parent process.

   d) Message passing based Inter-Process Communication (IPC) consumes less memory than shared memory based IPC.

d)False, message passing IPC consumes more memory as it takes up more memory in the user space

2. Data contains global parameters and no local variables while Stack contains local parameters and local variables in a function

2. What are two main differences between the data and stack regions of a process memory?

Size of data region is statically fixed while stack can grow and shrink as process executes

3. Explain the difference between a single-threaded and a multi-threaded process.

4. The figure below shows the execution of processes P0 and P1 in a multiprogramming system.

   a) Identify state transitions of each process.

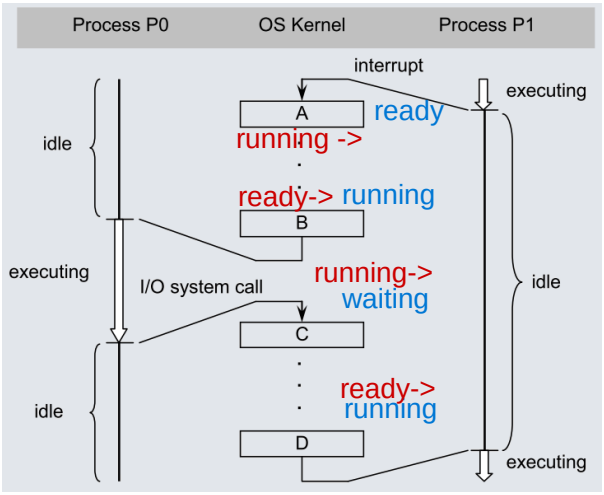
   b) Describe operations A, B, C and D performed by the operating system kernel.

Same as a process

3. Single threaded process is one unit of CPU utilization, using the register set and stack space. Multi threaded process share with its peer threads the code and data sections as well as operating system resources in the same process. While one thread is blocked and waiting, another thread can run, enabling higher throughput

Individual threads can execute concurrently

Threads in a process share code, data and heap regions of memory, whereas stack space is unique to each thread. Each thread has its own Thread Control Block similar to PCB.




4. In the running state, instructions are executed to run the process P0 and P1
An interrupt system call triggers the process to be in the ready state as it waits to be assigned to the CPU

An I/O system call causes the process to wait for the I/O event

A:
Change mode to kernel
Save P1 state into PCB1
Invoke ISR

B:
Run scheduler,select P0
Load P0 state from PCB
Change mode to user

C:
Change mode to kernel
Save P0 state into PCB0
Invoke device driver

D:
Run scheduler, select P1
Load P1 state from PCB
Change mode to user