3.1WHAT IS A PROCESS?

1- A process can be defined as a program that is being executed.

3.2 PROCESS STATES

2- Because the two state model doesn't take into considerations the processes that are waiting for I\O operation. These processes are blocked waiting for the I\O operation to complete. In this case, using a single queue, the dispatcher will have to scan the queue for the process that is not blocked and has been waiting the longest.

3-

a- An infinite number of child processes will be created which will overwhelm the system.

b-

- The OS can limit the maximum number of processes that can be spawned by one process.
- The OS can also set a limit on the maximum memory space that a process can use.

c- Yes, because each process will have a limit on the maximum number of processes that it can create. Any attempt to create a new process beyond this limit will be rejected.

d- The process handling mechanism will need some modification. For example, the OS will have to monitor the processes making sure that each process doesn't spawn a number of child processes that exceeds the maximum allowable number of spawned processes. Also, error handling must be implemented in the case of exceeding the maximum limit.

4-

- Destroying the child processes when the parent is destroyed makes sure that no zombie processes reside in main memory consuming system resources. On the other hand, destroying a child process along with its parent while the child process is running may be inefficient or even, in some cases, dangerous.

- Allowing a child process to run independently of its parent gives the OS more control over this process. But, sometimes this approach will cause misuse of resources by this child process.

5-

a- A suspended process is a process that was residing in the main memory in the blocked state and got swapped out to the secondary memory to save some space in the main memory for a new process. This happens when there are no processes in the ready state or when a process in the ready state requires more memory space.
b- A zombie process is a process that has already finished execution but still resides in main memory consuming system resources. This usually happens when a main process creates a child process.

6- A balanced policy can be as follows: if the processor knows that the process in the ready/suspend state will not need much work, it can be swapped to the main memory to be executed. Otherwise, choose the one in the ready state. Another solution can be derived. If there are already too many processes in the main memory, the process should first finish executing some of them before adding in the process that is in the ready/suspend state. This will help not to add too much load on the system.

3.3 PROCESS DESCRIPTION

7- A process control block contains all the data needed by the OS to manage the process like the process id and the current state of the process.

3.4 PROCESS CONTROL

8-

	Process switching	Mode switching
Definition	A running process, at some time, may get interrupted and the OS assigns another process to the running state and	A mode switch is a switch that occurs without changing the state of the process that is

	turns control over to that process.	currently in the running state.
How to do	When a process is interrupted, its context is saved onto a stack. When the execution of that process resumes, the context is popped of the stack and the execution continues from where it left off.	The context of the process that got interrupted is saved into the process control block of the interrupted program.
Time consumption	Typically, a process switch needs more time to finish than a mode switch.	Mode switching has a little overhead compared to process switching.

9- The protection of the PCB is achieved by having 2 modes of operation, the user mode and the kernel mode. In the kernel mode, the user program has no access to certain instructions and certain portions of memory in which the PCB is stored.

3.5 EXECUTION OF THE OPERATING SYSTEM

10- Having multiple blocked queues in which each device has its own queue helps increase the performance of the system. For example, if a process needs to use the printer, it will get blocked until the I/O operation is over. When the printer issues an interrupt indicating that the operation is over, the process is now ready to resume execution. Now, the dispatcher will choose that process that is associated with the printer from the queue of the printer. In case of one queue model, the dispatcher will have to scan the entire queue looking for this process which will degrade the performance in cases where hundreds or even thousands of processes are waiting in this queue.

3.6 UNIX SVR4 PROCESS MANAGEMENT

11- when fork() is called, a child process will be created. The OS will allocate memory to the new process, a unique identifier will be assigned to the child process, and a copy of the process image of the parent will be created.

General Questions

13-

- Ready/Run: a process will go from the ready state to the running state when it’s chosen by the dispatcher.
- Run/Ready: a process will go from the running state to the ready state when it consumes its time slice or when it’s pre-empted by a higher-priority process.
- Run/Blocked: this occurs when the running process requests an I/O operation or, in general, any request that the OS can’t perform immediately.
- Blocked/Ready: this transition occurs when the event that the process has been waiting for is over and it can resume execution.
- Blocked/Suspend: this transition may happen when there are no processes in the ready state. Rather, they are all blocked. In this case, it will be better to swap one of the blocked processes to the secondary memory leaving some space for a new process to be executed. This transition may also occur when some process in the ready state needs more memory to be executed. So, one of the blocked processes will be swapped out.