Processes

3.1 WHAT IS A PROCESS?

1. Define the following terms:
   a. Process

3.2 PROCESS STATES

2. What are the disadvantages of the two-state model of processes? How to solve them?

3. The ability of one process to spawn a new process is an important capability, but it is not without its dangers. Consider the consequences of allowing a user to run the process in the following figure. Assume that fork() is a system call that spawns a child process.

```
int main() {
        while(true) {
                fork();
        }
}
```

   a. If a system allowed such a process to run, what would the consequences be?
   b. Suppose you decide that it is inappropriate to reject certain processes and that the best approach is to place certain runtime controls on them. What controls might the operating system use to detect processes like the above at runtime?
   c. Would the controls you propose hinder a process's ability to spawn new processes?
   d. How would the implementation of the controls you propose affect the design of the system's process handling mechanism?

4. In some systems, a spawned process is destroyed automatically when its parent is destroyed; in other systems spawned processes proceed independently of their parents, and the destruction of a parent has no effect on its children.
   a. Discuss the advantages and disadvantages of each approach.
   b. Give an example of a situation in which destroying a parent should specifically not result in the destruction of its children.

5. Define the following terms:

a. Suspended process

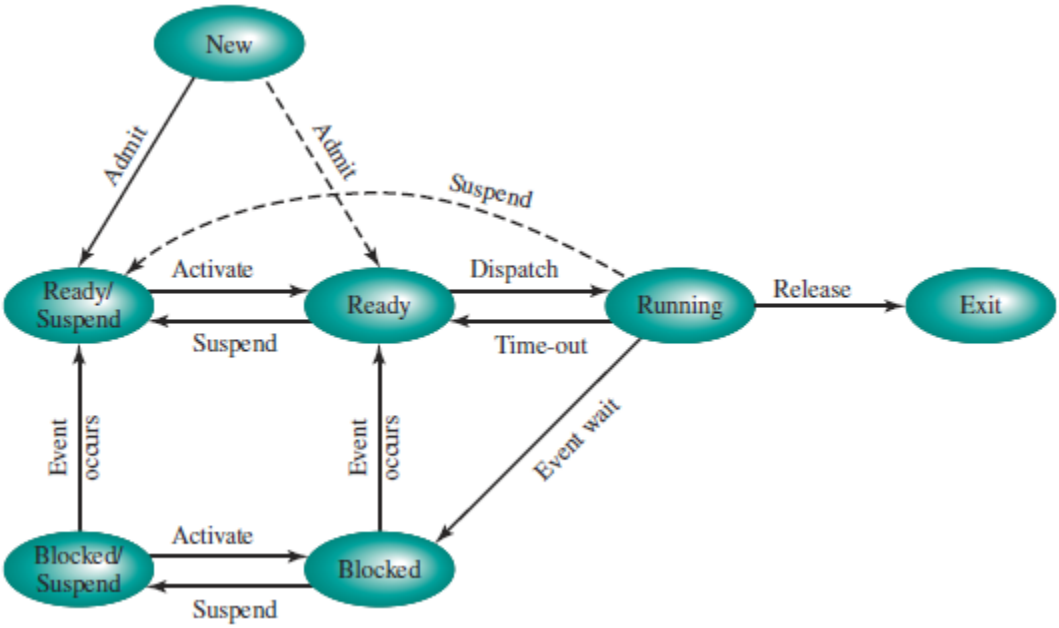
b. Zombie process

6. Consider the state transition diagram of the seven-state process model below, where there is two states for suspended processes (Ready/suspended ad Blocked/ Suspended). Suppose it is time for the OS to dispatch a process and there are processes in both the Ready state and the Ready/Suspend state, and at least one process in the Ready/Suspend state has higher scheduling priority than any of the processes in the Ready state. Two extreme policies are as follows:

a. Always dispatch from a process in the Ready state, to minimize swapping, and

b. always give preference to the highest-priority process, even though that may mean swapping when swapping is not necessary.

c. Suggest an intermediate policy that tries to balance the concerns of priority and performance.



3.3 PROCESS DESCRIPTION

7. Define the following terms:
   a. Process control block

3.4 PROCESS CONTROL

8. Compare process switching and mode switching considering the following points:
   a. Definition.
   b. How to do?
   c. Time consumption.

9. How to protect PCB from damage by other processes?

3.5 EXECUTION OF THE OPERATING SYSTEM

10. What are the advantages of multiple blocked queues over the ordinary model? How to handle priorities?

3.6 UNIX SVR4 PROCESS MANAGEMENT

11. What happens when the UNIX system call fork() is called?

General Questions

12. A system adopts a priority-based preemptive scheduling where the initial priority of a process increases by 1 after every 5 ms. In a recorded time span, the system has four processes, P1, P2, P3 and P4, as shown in the following table:

Process id	Initial priority	Arrival time in ms	Total cpu time in ms
P1	1	0	15
P2	3	5	7.5
P3	2	10	5
P4	2	15	10

Draw a timing diagram and find the turnaround time for each process. Assume that the dispatcher takes 2.5 milliseconds for a process switch.

13. The following state transition table is a simplified model of process management, with the labels representing transitions between states of READY, RUN, BLOCKED and SUSPENDED.

	READY	RUN	BLOCKED	SUSPENDED
READY	-	1	-	5
RUN	2	-	3	-
BLOCKED	4	-	-	6

Give an example of an event that can cause each of the above transitions. List 3 impossible transitions and explain their impossibility.