Operating systems

Sheet3

(EED)

Name: Samuel AymanShawky

ID: 20010750

3.1 WHAT IS A PROCESS?

1. Define the following terms:

a. Process: it's a sequence of instructions that are executed a program in execution.

3.2 PROCESS STATES

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

The disadvantage of the two-state model is, there are processes in not running state are ready and the others are blocked in the same queue which is not efficient.

To solve this problem not running state should be separated to ready state and blocked state.

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?

The system will crash as a loop will keep creating processes until it exceeds the limits of resources (CPU, memory,..)

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?

Controls like:

Process creation limit: set a maximum number of processes that process can spawn in a certain time if it exceeds this number terminate process.

Process monitoring: to monitor the process behavior and if there is abnormal behavior terminate the process.

c. Would the controls you propose hinder a process's ability to spawn new processes?

Yes, it would if the process created a number of processes larger than max number.

d. How would the implementation of the controls you propose affect the design of the system's process handling mechanism?

The operating system should be able to count the number of processes created by one process and if it exceeds the maximum number, the system should terminate it.

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.

Automatic Destruction of Child Processes with Parent:

Advantages: Simplifies process management by ensuring that all child processes are cleaned up automatically when their parent terminates, reducing the risk of orphaned processes.

Disadvantages: Limits flexibility and may not be suitable for scenarios where child processes need to continue executing independently of their parent.

Independent Child Processes:

Advantages: Offers greater flexibility and independence, allowing child processes to continue executing even after their parent terminates.

Disadvantages: Requires explicit management of child processes by the parent or the operating system, increasing complexity and the risk of orphaned processes if not properly handled.

b. Give an example of a situation in which destroying a parent should specifically not result in the destruction of its children.

A word was opened to print a document. The word process spawned a child process for printing and then the word process is terminated. In this case, the document will continue printing as this process is executed independently of its parent.

5. Define the following terms:

a. Suspended process?

Suspended process: is a process swapped to disk to free main memory for another ready process.

b. Zombie process

A zombie process is a process that has completed execution but still retains an entry in the process table.

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.

Answer)

An intermediate policy for process dispatching, called Hybrid Priority-Based Dispatching, balances priority and performance concerns. It prioritizes dispatching high-priority processes from the Ready/Suspend state.

7- Define the following terms:

a. Process control block

it's a block containing information of process like identification and program counter and priority and more.

8-Compare process switching and mode switching considering the following points:

a. Definition. b. How to do? c. Time consumption.

	process switching	mode switching
Definition	Process switching involves transitioning the CPU from executing one process to another	involves changing the CPU's operating mode between different processor modes, such as user mode and kernel mode.
How to do	Save the state of current process,	Mode switching is triggered by mechanisms, such as system calls, interrupts

	Select the next process, load its state then resume the execution	
Time consumption	high time overhead due to the need to save and restore the entire process state	Lower time overhead compared to process switching as no need to save and restore

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

By using two modes (kernel mode and user mode) and changing in PCB will be only by the system and in kernel mode which will prevent user from changing in PCB.

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

multiple blocked queues improve the source utilization and reduces the waiting time and achieve fairness.

There are many ways to handle priority like:

Assigning each block a priority level and starting execution from higher priority to lower.

Or we can use aging mechanism to prevent starving of lower priority processes.

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

On a fork () command the following happens

a. It allocates a slot in the process table for the new process.

b. It assigns a unique process ID to the child process.

c. It makes a copy of the process image of the parent, except for shared memory.

d. It increments counters for any files owned by the parent, to reflect that an additional process now also owns those files.

e. It assigns the child process to the Ready to Run state.

f. It returns the ID number of the child to the parent process, and a 0 value to the child process.

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:

P1	1	1	2	2	3	3	4	4	5	5	6	6							
P2			3	3	4	4													
P3					2	2	3	3	4	4	5	5	6	6					
P4							2	2	3	3	4	4	5	5	6	6	7	7	8
dispatcher																			
Time (ms)	0	2.5	5	7.5	10	12.5	15	17.5	20	22.5	25	27.5	30	32.5	35	37.5	40	42.5	

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.

Give an example of an event that can cause each of the above transitions.

Ready->running: occurs when dispatcher choose this process from ready queue.

Run -> ready: occurs when process ends its time slice.

Run-> blocked: occurs when this process waiting for action.

Blocked ->ready: occurs when action which process wait happens.

Ready->suspended: occurs when memory is full, and a process is temporarily swapped out of memory.

Blocked->suspended: occurs when memory is full, and a process is temporarily swapped out of memory.

List 3 impossible transitions and explain their impossibility.

1-Ready->blocked: impossible as process can't know if it's waiting for action before being executed in running state.

2-Run->suspended: it can't move running process to suspended ready process should be suspended rather than running process

3-blocked->Run: impossible as blocked process is moved to ready queue to be selected by dispatcher.