|  |  |  |
| --- | --- | --- |
| **KING SAUD UNIVERSITY**  **COLLEGE OF COMPUTER AND INFORMATION SCIENCES Computer Science Department** | | |
| **CSC 227: Operating System** | **Tutorial# 3**  **Due: Sun, March 5 (12-1)** | **2nd Semester 1437-1438**  **Spring 2017** |

**Question#1:**

*Select (T) for true or (F) for False.*

|  |  |  |
| --- | --- | --- |
| True/False | | Statement |
| T | **F** | 1. When you open two Skype sessions for your friend and your sister, the OS considers them as one execution sequence |
| T | **F** | 2. Once a process becomes ready, the OS assigns an identifier to it. |
| **T** | F | 3. The heap is the part of main memory that is dedicated to dynamic memory allocation. |
| T | **F** | 4. When the number of processes in the waiting queue increases, the probability that the CPU becomes idle decreases. |
| **T** | F | 5. It is possible in multi-processor environment to dispatch many processes at the same time. |
| T | **F** | 6. The CPU scheduler executes frequently so it is considered a long-term scheduler. |
| **T** | F | 7. Spooling provides a mechanism to solve the issue of having different speed devices attached to the computer. |
| **T** | F | 8. It is the responsibility of CPU scheduler to pick a process and assign it to a processor. |
| T | **F** | 9. There is a ready queue for each processor in a multiprocessor system. |
| T | **F** | 10. A process that exceeds the quantum time is interrupted and turned to the wait status. |

**Question#2:**

*Circle ALL correct answers. Note that there might be multiple correct answers; also, there might be no correct answer.*

1. If a running process needs to read a value form the user, the process then should be stored in:
   1. Ready Queue.
   2. Job pool.
   3. Waiting Queue.
   4. CPU Pool.
2. During the program execution, the functions and the parameters are stored in:
   1. Program counter.
   2. Stack.
   3. Heap.
   4. Job pool
3. What is the role of the program counter:
   1. Stores the address of the next instruction to be executed
   2. Stores the next instruction to be executed
   3. Assigns a unique number to each instruction statement
   4. Assigns a unique number to each process as soon as it is created
4. The CPU Scheduler dispatches the process, this means:
   1. The process is created.
   2. The process is sent to the Ready Queue.
   3. The process is running.
5. Which of the following information is part of PCB information:
   1. The page tables.
   2. The process priority.
   3. Base and limit register.
6. During the running state a process may be:
   1. Terminate.
   2. Turned to Ready.
   3. Interrupted.
7. Which of the following is not true about job scheduler:
   1. It reads from a disk.
   2. It executes frequently.
   3. It selects the program to be loaded in main memory.
   4. Its execution time is not long.
   5. It is known to be a short-term scheduler.
8. A process could be executed when:
   1. It is in the ready queue.
   2. The CPU scheduler selects it.
   3. The CPU is idle.
9. When the time slice assigned to a running process is exceeded:
   1. The process is moved to the waiting queue.
   2. The process is moved to the ready queue.
   3. The process is terminated.
10. When a process is spooled, it is stored in:
    1. The processor.
    2. A register.
    3. A buffer.

**Question #3:**

A process P1 running in a multitasking environment with time slice = 3 ms. The code of P1 is shown below:

|  |  |  |
| --- | --- | --- |
| *#* | *Process P1* | *Execution Time* |
| *1*  *2*  *3* | *Load (from memory)*  *Print*  *Assignment* | *3 ms*  *6 ms*  *3 ms* |

Given that:

* the waiting time in the I/O device queue is 3 ms;
* the dispatch time is 1 ms;
* push time in any queue is 2 ms

Complete the table below. The initial state is given in the first line:

1. What is the lifetime of the process? Accumulative Time = lifetime = 25 ms
2. What is the utilization percentage of the CPU? Utilization percentage = 12 / 25 \* 100 = 48%

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| *Accumulative*  *Time* | *Process*  *State* | *Executing*  *Instruction* | *Queue*  *Name* | *Execution*  *Time* | *Dispatch Time* | *Push*  *Time* | *Wait*  *Time in Q* |
| ***0*** | ***Ready*** |  | ***Ready Q*** |  | ***1*** |  |  |
| ***1*** | *Running* | *Load* |  | *3ms* | *1* |  |  |
| *5* | *Running* | *Print* |  | *3ms* |  | *2ms* |  |
| *10* | *Ready* |  | *Ready Q* |  | *1* |  |  |
| *11* | *Running* | *Print* |  | *3ms* |  | *2ms* |  |
| *16* | *Waiting* |  | *I/O device Q* |  |  |  | *3ms* |
| *21* | *Ready* |  | *Ready Q* |  | *1* |  |  |
| *22* | *Running* | *Assignment* |  | *3ms* |  |  |  |
| *25* | *Terminated* |  |  |  |  |  |  |