| CS2006 Operating System<br>Course Instructor<br>Ms. Mahzaib Younas | | |
|---|---|---|
| **Time allowed = 60 min** | **Quiz 5 Grand Quiz** | **Total Marks = 40** |

**BCS Section F**

| | | |
|---|---|---|
| **Roll No** | **Name** | **Signature** |

**Question No.1: Choose the correct one. [6 Makrs]**

| | |
|---|---|
| **1. The key difference between threads and processes is:** | **2. Which scheduling algorithm is least likely to cause starvation?** |
| A. Threads share memory space<br>B. Threads do not share code<br>C. Processes are faster to create<br>D. Processes are better for multithreading | A. Shortest Job First<br>**B. Round Robin**<br>C. Priority Scheduling<br>D. Multilevel Queue Scheduling |
| **3. In multilevel feedback queue scheduling, what happens to a process that waits too long in a lower-priority queue?** | **4. If process P1 forks a child process P2 and P2 performs an exec(), the outcome is:** |
| A. It is moved to a higher-priority queue<br>B. It is terminated<br>C. It remains in the same queue<br>D. It skips CPU bursts | A. P2 terminates<br>B. P1 starts a new program<br>**C. P2 starts a new program**<br>D. P2 remains identical to P1 |
| **5. In Priority Scheduling, a major drawback is:** | **6. Shortest Job First (SJF) scheduling is optimal in terms of:** |
| A. All processes are treated equally<br>B. Excessive context switching<br>**C. Starvation of lower-priority processes**<br>D. It only works for batch processing | A. Minimum throughput<br>B. Maximum CPU utilization<br>C. Maximum turnaround time<br>**D. Minimum average waiting time** |

**Question No 02: What are the advantages of thread-local storage (TLS)? [3 Marks]**

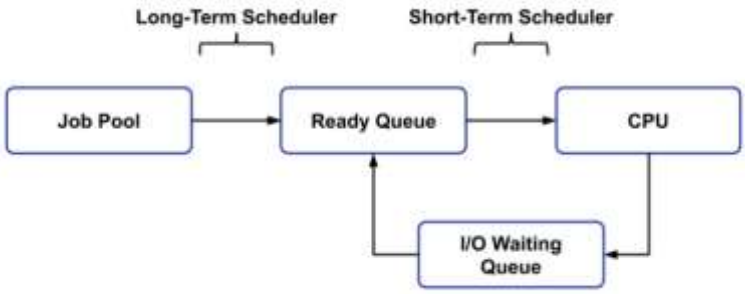
1. provide unique data for each thread that the process can access using a global index.
2. One thread allocates the index, which can be used by the other threads to retrieve the unique data associated with the index
3. Static local storage

**Question No 03: Describe all threading models. Also explain the limitations of each model.** **[8 Marks]**

| **One to one** | **Many to One** |
|---|---|
| The one-to-one model (one user thread to one kernel thread) is among the earliest implementations of true multithreading<br>**Limitation:**<br>main problem with this model is that it places a restriction on you to be careful with threads, as each additional thread adds more "weight" to the process. | allow the application to create any number of threads that can execute concurrently.<br>**Limitation:**<br>In a many-to-one (user-level threads) implementation, all threads activity is restricted to user space. |
| **Many to Many** | **Two Mode Model** |
| Many user level threads can map on the many kernel Thread.<br><br>Many-to-many model (many user-level threads to many kernel-level threads) avoids many of the limitations of the one-to-one model, while extending multithreading capabilities<br><br>**Limitation**:<br>Identifying and resolving problems may become more demanding due to the intricate nature of the code. | |

**Question No. 04: What is the role of middle-level scheduler? Explain with the help of diagram.** **[3 Marks]**

Short-Term Scheduler is also known as CPU scheduler and is responsible for selecting one process from the ready state for scheduling it on the running state




**Question No 05:** **[8 Marks]**

| **Process** | **Arrival Time** | **Burst Time** |
|---|---|---|
| **P1** | **7** | **12** |

| | | |
|---|---|---|
| **P2** | **5** | **6** |
| **P3** | **2** | **8** |
| **P4** | **1** | **10** |
| **P5** | **0** | **14** |

**Use SJF (Preemptive)**

a) Draw its Gantt chart.
b) Calculate average waiting and turnaround time.
c) Calculate throughput of each process.

| Process | Burst Time | Completion time | Waiting Time | Turnaround Time |
|---|---|---|---|---|
| **P1** | 12 | 37 | 18 | 30 |
| **P2** | 6 | 16 | 5 | 11 |
| **P3** | 8 | 10 | 0 | 8 |
| **P4** | 10 | 25 | 14 | 24 |
| **P5** | 14 | 50 | 36 | 50 |

| P5 | P4 | P3 | P3 | P3 | P2 | P4 | P1 | P5 |
|---|---|---|---|---|---|---|---|---|
| 0 1 | 2 | 5 | 7 | 10 | 16 | 25 | 37 | 50 |

$$Average\ Turanaround\ Time = \frac{18 + 5 + 0 + 14 + 36}{5} = \frac{73}{5}$$

$$Average\ Waiting\ Time = \frac{30 + 11 + 8 + 14 + 50}{5} = \frac{123}{5}$$

$$Throughput\ = \frac{1}{50}$$

**Question no 06: Discuss the PCS and SCS, Explain with the help of diagram. [4 Marks]**

**PCS**
user-level threads to run on LWP Known as since scheduling competition is within the process-contention scope (PCS)
Typically done via priority set by programmer

**SCS**
Kernel thread scheduled onto available CPU is system-contention scope (SCS) – competition among all threads in system.

**Question No 07: Discuss the term load balancing in Multi-Processor Scheduling?**

**[4 Marks]**

If SMP, need to keep all CPUs loaded for efficiency
• Load balancing attempts to keep workload evenly distributed
• Push migration
periodic task checks load on each processor, and if found pushes task from overloaded CPU to other CPUs
• Pull migration
idle processors pulls waiting task from busy processor

**Question no 08: Discuss the method to terminate the process in C/C++ language in Parent child Process Relation.** **[4 Marks]**

1. Wait ( )
2. Abort ( )
3. Exit ( )