Roll Number:_____ Name_____

THAPAR INSTITUTE OF ENGINEERING & TECHNOLOGY, PATIALA
MECHATRONICS ENGINEERING
Operating System (UCS303), Mid Semester Test

Date & Time: 11/03/2024 & 3:00 PM MM: 30 & MT: 120 Min

Note: Attempt all questions with sub-parts a and b at the same place and write the page number on the front page of the answer sheet . Assume missing data (if any). Each answer should be to the point.

<table>
<tr><th>Q. No</th><th>Question</th><th>Marks</th><th>CO</th><th>BL</th></tr>
<tr><td>Q 1</td><td>Explain each and every state of the process lifecycle. Examine the table containing arrival times and burst times for four processes. If the CPU scheduling utilizes the Shortest Remaining Time First algorithm, the scheduler will prioritize processes with the shortest remaining time. In instances of a tie, the scheduler will favor the process with higher priority (lower number representing higher priority).

<table>
<tr><th rowspan="2">Process No.</th><th rowspan="2">Arrival Time</th><th rowspan="2">Priority</th><th colspan="3">Execution Time</th></tr>
<tr><th>CPU Burst</th><th>I/O Burst</th><th>CPU Burst</th></tr>
<tr><td>P1</td><td>0</td><td>2</td><td>5</td><td>5</td><td>2</td></tr>
<tr><td>P2</td><td>3</td><td>5</td><td>2</td><td>18</td><td>2</td></tr>
<tr><td>P3</td><td>7</td><td>6</td><td>8</td><td>0</td><td>0</td></tr>
<tr><td>P4</td><td>23</td><td>8</td><td>10</td><td>2</td><td>1</td></tr>
</table>

Considering scheduling and context switching overhead as negligible, answer the following:
1. Draw the Gantt chart showing the sequence of execution for the processes.
2. Calculate the average waiting time, CPU utilization and average turnaround time.</td><td>6</td><td>CO2</td><td>L4</td></tr>
<tr><td>Q 2</td><td>Explain role of system calls in dual mode of operation of operating system. Describe the purpose of the fork system call and Illustrate the behavior of the parent and child processes after a fork operation. Given the following code snippet, how many times “Mid Semester Test” would be printed, design the parent child hierarchy.
<pre>
int main()
{
   fork();
   printf("Mid Semester Test \n");
   fork();
   printf("Mid Semester Test \n");
   if(fork());
   printf("Mid Semester Test \n");
   return 0;
}
</pre></td><td>6</td><td>CO1</td><td>L4</td></tr>
</table>

<table>
<tr>
<td>Q3</td>
<td>Assume that there are 5 processes, P0 through P4, and 4 types of resources. At T0 we have the following system state: Max Instances of Resource Type A = 3, Max Instances of Resource Type B = 17, Max Instances of Resource Type C = 16, Max Instances of Resource Type D = 12.

<table>
<tr><th></th><th colspan="4">Allocation Matrix
(N0 of the allocated resources By a process)</th><th colspan="4">Max Matrix
Max resources that may be used by a process</th></tr>
<tr><th></th><th>A</th><th>B</th><th>C</th><th>D</th><th>A</th><th>B</th><th>C</th><th>D</th></tr>
<tr><td>P0</td><td>0</td><td>1</td><td>1</td><td>0</td><td>0</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>P1</td><td>1</td><td>2</td><td>3</td><td>1</td><td>1</td><td>6</td><td>5</td><td>2</td></tr>
<tr><td>P2</td><td>1</td><td>3</td><td>6</td><td>5</td><td>2</td><td>3</td><td>6</td><td>6</td></tr>
<tr><td>P3</td><td>0</td><td>6</td><td>3</td><td>2</td><td>0</td><td>6</td><td>5</td><td>2</td></tr>
<tr><td>P4</td><td>0</td><td>0</td><td>1</td><td>4</td><td>0</td><td>6</td><td>5</td><td>6</td></tr>
</table>
1. Use the safety algorithm to test if the system is in a safe state or not? If yes, find out the safe sequence.
2. Can the following requests be granted, why or why not? Please also run the safety algorithm on each request as necessary.
a. P1 requests (2,1,1,0)
b. P1 requests (0,2,1,0)
c. P4 request (0, 5, 2 ,0)</td>
<td>6</td><td>CO2</td><td>L3</td>
</tr>
<tr>
<td rowspan="2">Q4</td>
<td>a) Define how to eliminate circular wait conditions to prevent the deadlock with a clear example? In a system, the following state of process and resources are given: P0→R2, P0→R3, R1→P0, R3→P0, R2→P1, R2→P1, P1→R1, R1→P1, R2→P2, P2→R3, R2→P3,P3→R2, P3→R2.
1. Draw a suitable resource allocation graph.
2. Draw the corresponding wait-for-graph.
3. Identify whether the system is in a deadlock state or not.</td>
<td>4</td><td>CO2</td><td>L3</td>
</tr>
<tr>
<td>b) Define a Thread? Compare and contrast the advantages and disadvantages of multithreading versus multiprocessing. What is the difference between user-level threads and kernel-level threads?</td>
<td>3</td><td>CO2</td><td>L2</td>
</tr>
<tr>
<td>Q5</td>
<td>a) Compare and contrast the structure of monolithic kernels and microkernels.
b) Explain following
1. Critical section problem
2. Aging</td>
<td>3
2</td><td>Co1
Co5+ Co2</td><td>L2
L2</td>
</tr>
</table>

Marks Distribution