**Department of Computer Science and Engineering**

**A**

**Midterm Examination Summer 2023**

**CSE 321: Operating Systems**

|  |  |
| --- | --- |
| **Duration:** 1 Hour 15 Minutes | **Total Marks:** 25 |

Answer the following questions.

Figures in the right margin indicate marks.

|  |  |  |
| --- | --- | --- |
| **1.**  **CO1** | **a) Why** do we need cooperating processes, and what are the two models of interprocess communication? Additionally, **mention** the strengths and weaknesses of each approach? | **[3]** |
|  | **b)** **Discuss** one drawback faced by multi-programmed OS architecture and suggest possible way(s) to overcome it. | **[2]** |
|  | **c)** **What** is the purpose of system calls? **Which** of the following instructions should be privileged?  **i.** Set value of timer **ii.** Read the clock **iii.** Clear memory **iv.** Issue a trap instruction. **v.** Turn off interrupts **vi.** Modify entries in device-status table | **[1+1]** |
|  | **d)** **Find** the output of the following code snippet. Your output should exactly match with the original output.   |  | | --- | | **int main() {**  **pid\_t child\_pid;**  **int global\_a = 90, b = 11;**  **char message[] = "Hello, from the ";**  **printf("Parent process started\n");**  **child\_pid = fork();**  **if (child\_pid == -1) {**  **printf(“Fork Failed\n”);**  **} else if (child\_pid > 0) {**  **wait(NULL);**  **global\_a += 97;**  **printf("%sAddition: %d + %d = %d\n", message, global\_a, b, global\_a);**  **printf("%sSubtraction: %d - %d = %d\n", message, b, global\_a, b);**  **} else {**  **b \*= 7;**  **printf("Multiplication: %d \* %d = %d\n", global\_a, b, global\_a);**  **printf("Division: %d / %d = %d\n", b, global\_a, b);**  **}**  **return 0;**  **}** | | **[3]** |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **2.**  **CO2** | |  |  |  |  | | --- | --- | --- | --- | | **Processes** | **Arrival Time** | **Burst Time** | **Priority** | | **P1** | 0 | 4 | 2 | | **P2** | 6 | 4 | 1 | | **P3** | 7 | 6 | 6 | | **P4** | 7 | 1 | 3 | | **P5** | 8 | 7 | 4 | | **P6** | 19 | 7 | 5 | |  |
|  | **a)** **Draw** a Gantt chart and illustrate the execution of the process using the **Round Robin** scheduling algorithm **(time quantum = 5 units)**. **Calculate** the **average waiting** and **turnaround** **time.** | **[3+2]** |
|  | **b)** **Apply** **Preemptive Priority** scheduling algorithm. **Draw** the Gantt chart and **Calculate** the **average waiting** and **turnaround time.** | **[2+2]** |
|  | **c)** **Compare** the results and **identify** the most suitable scheduling algorithm in this scenario. | **[1]** |
| **3.**  **CO3** | **a)** **Explain** **data parallelism** with an example. | **[1.5]** |
|  | **b)** You are developing a real-time embedded system for a safety-critical application, such as an advanced driver assistance system (ADAS) for autonomous vehicles. The system's primary goal is to process sensor data, make critical decisions, and take actions in real-time to ensure the safety of passengers and pedestrians. However, you may assume there’s no limitation of computational resources on the project you are working on.  Based on the scenario, **which** multi-threading model would you recommend for implementation? **Provide** necessary justification. | **[1.5]** |
|  | **c)** A system has processes to execute of which **30%** is serial. If the number of cores is decreased from **9** to **4**, **Explain** the change in the performance. | **[2]** |