B

Department of Computer Science and Engineering
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) Define** the term "**interrupt**" in the context of operating systems. **Mention** one advantages and one disadvantage of a multiprocessor system. **[3]**

The medium term scheduler swaps processes in and out memory to optimize CPU uses and manage memory allocation.

sometime it can be advantageous to remove process from the memory and thus to reduce the degree of multi-programming.

Remove process from memory, store on disk, bring back in from disk to continue execution: swapping. Swapping allows the system to pause and later resume a process, improving overall system efficiency.

b) A process from its creation till its completion will go through various states. To enter different states, the process requires the decision of different types of scheduler. **State** the name of different schedulers for different process states with justification. **[2]**

c) What is the purpose of dual mode operation? **Which** of the following instructions should be privileged? **[1+1]**
i. Access I/O device **ii.** Set value of timer **iii.** Read the date in the calendar.
iv. Clear memory **v.** Switch from user to kernel mode **vi.** Turn off interrupts.

d) Find the output of the following code snippet. **Your output should exactly match with the original output.** **[3]**

```
int main() {
    pid_t child_pid;
    int global_a = 68, b = 10;
    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);
        b *= 38;
        printf("Multiplication: %d * %d = %d\n", global_a,
b, global_a);
        printf("Division: %d / %d = %d\n", b, global_a, b);
    } else {
        global_a += 98;
        printf("%sAddition: %d + %d = %d\n", message,
global_a, b, global_a);
        printf("%sSubtraction: %d - %d = %d\n", message, b,
global_a, b);
    }
    return 0;
```

```
}
```

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 task parallelism with an example. **[1.5]**

b) You are developing a lightweight, user-level threading library for a resource-constrained embedded system. The embedded system has limited processing power and memory, and it does not provide native support for multithreading at the kernel level. The primary goal is to allow concurrent execution of multiple tasks while minimizing the overhead of managing threads. **[1.5]**

Based on the scenario, **which** multi-threading model would you recommend for implementation? **Provide** necessary justification.

c) A system has processes to execute of which **32%** is serial. If the number of cores is decreased from **8** to **2**, **Explain** the change in the performance. **[2]**