ourdiu.com | Shared by Ferdosh Riyad

# Daffodil International University

**Faculty of Science & Information Technology**
**Department of Computer Science & Engineering**
**Midterm Examination, Fall 2026**
**Course Code: CSE 323, Course Title: Operating Systems**
**Level: 3 Term: 3 Batch: 66**

**Time: 01:30 Hrs** **Marks: 25**

## Answer ALL Questions

*[The figures in the right margin indicate the full marks and corresponding course outcomes. All portions of each question must be answered sequentially.]*

| | | | | |
|---|---|---|---|---|
| **1.** | | A tech startup is developing a smart-home automation system. The system must immediately trigger fire alarms and water sprinklers if a fire is detected, while simultaneously running background tasks like syncing daily temperature logs to a cloud server. The development team decides to use an operating system with dual-mode operation. | | **CO1** |
| | ***a)*** | Based on the computing environment discussed, which type of operating system must be implemented for the fire detection module? Justify your answer. | **[2]** | |
| | ***b)*** | Explain how the OS prevents the background cloud-syncing application from accidentally modifying the critical fire alarm handlers. | **[3]** | |
| **2.** | | A software engineering team is developing a high-performance database application for a custom Unix-based operating system. When a user runs a query, the application must read raw data from the hard drive and output the formatted results to the screen. To improve system security and prevent total system crashes if a driver fails, the OS architects decide to restructure the custom OS from Monolithic to another.<br><br>i) Identify which system structure will be appropriate here. Justify your answer.<br>ii) What performance drawbacks the system may face in your proposed structure. | **[3]** | **CO1** |
| **3.** | | `#include <stdio.h>`<br>`#include <unistd.h>`<br>`#include <sys/wait.h>`<br>`int main()`<br>`{`<br>`  int pid = fork();`<br>`  if (pid == 0) {`<br>`  printf("Child: Processing student records...\n");`<br>`  sleep(2);`<br>`  printf("Child: Processing completed.\n");`<br>`  return 0;`<br>`  }`<br>`  // Turn over the page` | | **CO1** |

ourdiu.com | Shared by Ferdosh Riyad

```
    printf("Parent: Updating the database...\n");
    sleep(5);
    printf("Parent: Database update completed.\n");

    return 0;
}
```

| | | | | |
|---|---|---|---|---|
| | *a)* | After the child finishes its execution, the parent continues working for several more seconds without collecting the child's termination status.<br><br>Explain what happens to the child process during this period. | [2] | |
| | *b)* | Suppose the system requirements change where the parent process must not start updating the database until the child terminates and returns the status to parent.<br><br>Explain your approach and rewrite the given code to demonstrate your solution. | [2] | |
| | *c)* | Suppose the child is given access to some specific database files that the parent is updating, while the parent restricts the access to all its other resources.<br><br>Explain which resource-sharing arrangement is being utilized here. | [2] | |
| **4.** | | Intel 4004 was released on November 15, 1971, featuring a single processing unit that runs at 740 kHz. To handle multiple programs, it switched back and forth between tasks so quickly that it felt simultaneous to the user. For any process the system takes 0.05 seconds to save a PCB and 0.05 seconds to load a PCB during context switching.<br><br>The system receives the following processes after a successful bootup where the ready queue follows FIFO order. | | **CO2** |

| Process | Arrival Time | Burst Time |
|---|---|---|
| P1 | 0 | 1 |
| P2 | 3 | 13 |
| P3 | 5 | 9 |
| P4 | 7 | 8 |
| P5 | 9 | 6 |

| | | | | |
|---|---|---|---|---|
| | *a)* | If the system uses Round Robin scheduling with a time quantum of 3.5 seconds,<br>i) Draw the gantt chart for process execution including the context-switch intervals.<br>ii) Compute the completion time, turnaround time, and waiting time. | [6] | |
| | *b)* | Discuss what happens if the time quantum is a bit longer and the context-switching duration is shorter. How does it affect the overall system efficiency? | [2] | |
| | *c)* | Draw the gantt chart using Shortest Remaining Time First algorithm considering the context-switch overhead for process execution . | [3] | |