

Operating Systems (CS2006)

Date: March 26, 2024

Course Instructor(s)

Dr. Nadeem Kafi, Ms. Safia Baloch, Ms.
Mubashra Fayyaz, Mr. Minhaz Raza, Ms. Hania
Shah

Sessional-I Re-Exam

Total Time: 1 Hours

Total Marks: 15

Total Questions: 02

Semester: SP-2024

Campus: Karachi

Dept: Computer Science

Student Name

Roll No

Section

Student Signature

CLO # 1: Describe, discuss, and analyze, services provided by the modern Operating Systems.

Q1. [1.5 marks x 5 = 7.5 marks]

Write short textual answers. Drawings are not allowed.

- Operating systems should use computing hardware efficiently but when it is appropriate to abandon this principle and to waste hardware resources? Give an example.
- Describe advantages and disadvantages of layered architecture.
- How is context-switch implemented in a multitasking operating system? Draw and explain.

Give only labelled diagrams. *Note: textual answers will not be graded.*

- For two executing processes, **illustrate all file descriptors in each process when they are connected using an ordinary pipe.**
- Illustrate memory layout** of the process when the below C program is executed.

```
#include <stdio.h>
#include <stdlib.h>

int main() {
    int n=10, *arr;
    scanf("%d", &n);
    arr = (int *)malloc(n * sizeof(int));
    free(arr);
    return 0;
}
```

CLO # 2: Understand, design, and implement solutions employing concepts of Processes/Threads.

Q2. [1.5 marks + 2 marks + 4 marks = 7.5 marks]

Understanding and Design

- Consider a producer-consumer scenario with two processes. How shared memory is used to share a data structure between the two processes? Explain. [1.5]

National University of Computer and Emerging Sciences

b) Given the following C program. When and how will "LINE J" printed? Give technical explanation. [2]

```
#include <sys/types.h>
#include <stdio.h>
#include <unistd.h>

int main()
{
    pid_t pid;

    /* fork a child process */
    pid = fork();

    if (pid < 0) { /* error occurred */
        fprintf(stderr, "Fork Failed");
        return 1;
    }
    else if (pid == 0) { /* child process */
        execlp("/bin/ls", "ls", NULL);
        printf("LINE J");
    }
    else { /* parent process */
        /* parent will wait for the child to complete */
        wait(NULL);
        printf("Child Complete");
    }

    return 0;
}
```

Implementation

- c) Assume the C language program shown below is executed from the bash command line.
- Draw diagram with hints to show the sequence in which the program will create parent and child process. *Do not write a description and calculate the total number of processes.* [2]
 - Suppose 4K memory is allocated by the operating system when a process is created. How much memory is consumed by this program just before the execution of the return 0 statement? [2]

```
#include <stdio.h>
#include <unistd.h>

int main()
{
    /* fork a child process */
    fork();

    /* fork another child process */
    fork();

    /* and fork another */
    fork();

    return 0;
}
```