Operating Systems Lab
Part(A)
(CL2006)
Date: May 5th 2025

Course Instructor

Ms. Fatima Gado, Ms.Bushra Sattar,
Ms. Mahnoor javed

Final Lab Exam(B)

Total Time: 45 Minutes
Total Marks: 20
Total Questions: 05

Semester: Fall-2025
Campus: Karachi
Dept: Computer Science

Student Name Roll No B Section Student Signature

Q1: Linux Commands (4 WTG)

Explain the purpose of the following commands with an example of their usage:

a) find . -type f -name "*.cpp" | wc –l

b) alias ll='ls -l'

Q2: CPU Scheduling (4 WTG)

Given the following processes:

Process	Arrival Time (ms)	Burst Time (ms)	Completion Time	Waiting Time	Turnaround Time
P1	0	5			
P2	2	7			
P3	3	3			

a) Explain how round robin algorithm would execute these processes:

b) compute the Gantt chart, Completion Time, Waiting Time, Turnaround Time, Average Waiting Time (AWT), and Average Turnaround Time (ATAT).

Q3: Shell/Kernel Configuration (4 WTG)

a) Explain what the following line does in a shell script cut -d ":" -f1 /etc/passwd

b) Write a shell script that display the: **1.** Current date and time. **2.** System uptime & **3.** Username of the person running the script

Q4: Makefiles (4 WTG)

Write a Makefile rule to: a) Compile addbooks.c and displaybooks.c seperatly into object files.
b) Link all object files into an executable named library.
c) Include a clean target to remove generated files.

Q5: Fork System Calls (4 WTG)

Analyze the following code:

```
int main() {
    if (fork() == 0) {
        printf("Child process. PID = %d, PPID = %d\n", getpid(), getppid());
    } else {
        fork();
        printf("Parent process. PID = %d, PPID = %d\n", getpid(), getppid());
    }
    return 0;
}
```

a) How many total processes are created? _____

b) What happens when the first fork() is executed?

National University of Computer and Emerging Sciences

Operating Systems Lab
Part(b)
(CL2006)
Date: May 5th 2025

Course Instructor

Ms. Fatima Gado, Ms. Bushra Sattar, Mahnoor javed

Final Lab Exam(B)

Total Time: 1 hour 30 min
Total Marks: 30
Total Questions: 04

Semester: Spring-2025
Campus: Karachi
Dept: Computer Science

Student Name _____ Roll No _____ Section B Student Signature _____

CLO # 1: Understand and Analyze Command Line tools for Linux OS and Shell scripts for system level programming to automate tasks such as file management, system backups, interrupts and software installations wtg [7.5+7.5]

Q1: Write a C program using shared memory to demonstrate inter-process communication (IPC) between two processes: Producer and Consumer. The Producer generates 5 random integers and stores them in shared memory. The Consumer reads the numbers from shared memory, squares them and computes their sum, and displays the result. ensure proper error handling, and perform cleanup of shared memory after execution.

Q2: Write a C program that creates a child process using fork() and demonstrates signal handling with SIGUSR1. The parent process should handle SIGUSR1 using a custom signal handler implemented with sigaction(), continuously waiting for the signal in an infinite loop by using pause(). Before waiting, the parent should print "Parent process is waiting for the signal." Once the signal is received, the parent prints "Parent process received the signal." and terminates. The child process should send the SIGUSR1 signal to the parent multiple times, with a 1-second delay between each signal, sending at least 3 signals. The parent should only terminate after receiving and handling the signal.

CLO # 2: Gain hands on experience in writing code that interacts with operating system services related process and files system, multi-thread programing and different synchronization primitives. (7.5 WTG)

Q3: You are required to implement a multithreaded simulation of a four-floor parking lot. Each floor has exactly 20 parking spots and operates independently to manage its own available spaces. There is one shared payment counter for the entire parking lot, and only one car can use it at any given time. In this simulation, each car should be implemented as a separate thread. When a car arrives, it randomly selects a floor to search for a parking space. If the selected floor is full, the car must try another floor until it finds an available spot. Once parked, the car remains in the spot for a short duration (simulating parking time), after which it proceeds to exit the parking lot. Before leaving, the car must go through the single payment counter, which must be accessed by only one car at a time to avoid concurrency issues.

Your task is to write c++ code that uses threading primitives such as Lock and Semaphore to safely manage concurrent access. Ensure that floor availability is updated in a thread-safe manner, and that the payment process enforces mutual exclusion. The program should simulate at least 100 cars arriving, parking, paying, and exiting concurrently.

CLO # 3: Understand and implement the use of semaphores for process synchronization as well as managing different types of semaphores. (7.5 WTG)

Q4: Write a C program using semaphores to synchronize two threads: a writer and a reader.

- The writer thread generates some data and writes it to a file. It then posts a semaphore to signal the reader that the data is ready.
- The reader thread waits for the semaphore to be posted, then reads the data from the file and displays it. After displaying the content, the reader thread exits the program.
- The program ensures proper synchronization between the threads using semaphores and handles file operations, ensuring data is written, read, and displayed in the correct order.
- Ensure proper error handling.