

University of Rajshahi
Department of Computer Science and Engineering
CSE3242: Operating Systems, Lab Test
Submission Deadline: 17-06-2025

Submission Type: Individual submission in both hand-written format and PDF format

Conditions:

- Students need to submit hand-written code on 17-6-2025 at the CSE office and upload a PDF version to the classroom.
- Students can take help from any source including AI based systems except any other subject having biological intelligence. Students need to disclose the source of information.
- Students need to show a screenshot of their output screens in the PDF format.
- Students need to include explanations about the output, reason behind output as much as possible.

1.	Write a C program to create a main process named 'parent_process' having 3 child processes without any grandchildren processes. Trace parent and child processes in the process tree. Show that child processes are doing addition, subtraction and multiplication on two variables initialized in the parent_process	10
2.	Write a C program to create an orphan process.	05
3.	Write a C program to create a zombie process.	05
4.	Write a C program to create a main process named 'parent_process' having 3 child processes without any grandchildren processes. Child Processes' names are child_1, child_2, child_3. Trace the position in the process tree.	10
5.	Write a C program to create a main process named 'parent_process' having 'n' child processes without any grandchildren processes. Child Processes' names are child_1, child_2, child_3,....., child_n. Trace the position in the process tree. Number of child processes (n) and name of child processes will be given in the CLI of Linux based systems. Example: \$./parent_process 3 child_1 child_2 child_3	10

6.	Write a C program to create a main process named 'parent_process' having 3 child processes without any grandchildren processes. Child Processes' names are child_1, child_2, child_3. Trace the position in the process tree.	10
7.	Write a C program to analyze the effect of local and global variables on a parent process and a child process.	05
8.	Write a C program to show how two related processes can communicate with each other by an unnamed pipe.	05
9.	Write a C program to show how two unrelated processes can communicate with each other by a named pipe.	05
10.	Write a C program to show how two related processes can communicate with each other by a named pipe.	05
11.	Write a C program to show how two unrelated processes can communicate with each other by a message queue.	05
12.	Write a C program to show how two related processes can communicate with each other by a message queue.	05
13.	Write a C program to show how data inconsistency arises in a multi-threaded process.	05
14.	Write a C program to show how data inconsistency arises in two related processes (e.g., parent & child processes) when they share a memory space.	05
15.	Write a C program to show how data inconsistency arises in two unrelated processes when they share a memory space.	05
16.	Write a C program to handle racing situations in a multi-threaded process.	05
17.	Write a C program to handle racing situations in multiple unrelated processes.	05
18.	Write a C program to show the usage of a Counter semaphore.	05
19.	Write a C program for creating a multi-threaded process and check: A. If one thread in the process calls fork(), does the new process duplicate all threads, or is the new process single-threaded? B. If a thread invokes the exec() system call, does it replace the entire code of the process? C. If exec() is called immediately after forking, will all threads be duplicated?	15

20.	<p>Write a multithreaded program that calculates various statistical values for a list of numbers. This program will be passed a series of numbers on the command line and will then create three separate worker threads. One thread will determine the average of the numbers, the second will determine the maximum value, and the third will determine the minimum value. For example, suppose your program is passed the integers</p> <p style="padding-left: 40px;">90 81 78 95 79 72 85</p> <p>The program will report</p> <ul style="list-style-type: none"> A. The average value is 82 B. The minimum value is 72 C. The maximum value is 95 <p>The variables representing the average, minimum, and maximum values will be stored globally. The worker threads will set these values, and the parent thread will output the values once the workers have exited.</p>	15
21.	<p>The Fibonacci sequence is the series of numbers 0, 1, 1, 2, 3, 5, 8, Formally, it can be expressed as:</p> <p style="padding-left: 40px;">$fib_0 = 0$ $fib_1 = 1$ $fib_n = fib_{n-1} + fib_{n-2}$</p> <p>Write a multithreaded program that generates the Fibonacci sequence. This program should work as follows: On the command line, the user will enter the number of Fibonacci numbers that the program is to generate. The program will then create a separate thread that will generate the Fibonacci numbers, placing the sequence in data that can be shared by the threads (an array is probably the most convenient data structure). When the thread finishes execution, the parent thread will output the sequence generated by the child thread. Because the parent thread cannot begin outputting the Fibonacci sequence until the child thread finishes, the parent thread will have to wait for the child thread to finish.</p>	15
22.	<p>Implement a server-client model to provide services to client processes running in different terminals. Explain what you experience when you:</p> <ul style="list-style-type: none"> • Server process being a single threaded process tries to provide services to multiple client processes. • Server process being a multi-threaded process tries to provide services to multiple client processes. • Server process being a single threaded process tries to provide services to multiple client processes with multiple child processes. 	20