National University of Computer and Emerging Sciences

Laboratory Manual # 06
Operating Systems

Course Instructor	Mubashar Hussain
Lab Instructor	Muhammad Hashir Mohsineen
Section	BCS-4E
Date	11-March-2025
Semester	Spring 25

Instructions:

- Submit a world/LibreOffice file containing screenshots of terminal commands/ Output
- Submit your .c (Code files)
- In case of any explanation you can add a multiline comment.

Objectives:

- Threads

Reading material:
https://docs.google.com/document/d/1OvZZ-MAkXwX8xKyqZh4ay4uPhf_Q2yReEAwARfOOgAA/edit?usp=sharing

1. Exercise: [10]

Create a program that creates three threads. Each thread should perform a different task:

a. Thread 1: Print numbers from 1 to 10.
b. Thread 2: Print the first 10 Fibonacci numbers.
c. Thread 3: Print the first 10 prime numbers.

- Use pthread_create to create the threads.
- Use pthread_join to wait for all threads to finish.

2. Exercise: [10]

Write a program which takes some positive integers (let's say N number of positive integers) as command line parameters, creates N synchronous threads, and send s the corresponding integer as parameter to the thread function fibonacciGenerator. The function returns the generated series to the main thread. The main thread will then print the thread number and the series generated by that thread. The output will be like:
Thread 1:01 12 3 5 8 13

Example:

If you pass as command line argument the following numbers: 3 13 34 89
Then the program will create 4 threads. The first thread will find Fibonacci terms until 3 is generated, the second Fibonacci term will find Fibonacci terms until the term generated is 13 so on and so forth. All generated terms will be output on the screen by the main thread as follows:

Thread 0: 0, 1, 1, 2, 3
Thread 1: 0, 1, 1, 2, 3, 5, 8, 13
Thread 2: 0, 1, 1, 2, 3, 5, 8, 13, 21, 34
Thread 3: 0, 1, 1, 2, 3, 5, 8, 1, 2, 3, 5, 8, 13, 21, 34 13, 21, 34, 55, 89

It is possible that the number passed to the thread is not a Fibonacci number. In this case the thread will generate numbers until the term generated is greater than the passed number. For example if 7 is passed as parameter to a thread, then the thread will return the following series:
0, 1, 1, 2, 3, 5, 8
