National University of Computer and Emerging Sciences Chiniot-Faisalabad Campus

Lab 08

CL1006 – Operating System - Lab

Course Instructor	Juhinah Batool Asif
Lab Instructor	Juhinah Batool Asif
Semester	Fall 2024

FAST School of Computing

Department of Computer Science

Instructions

1. Make a word document with the convention "ROLLNO_ LAB#_ SECTION" and put all your source code and snapshots of its output in it. You have to submit pdf file. 2. Plagiarism is strictly prohibited.

3. Do not discuss solutions with one another. Copying the solution from any source can lead to ZERO marks.

Lab Tasks:

Task 1

Write a program that creates threads based on the input given by user. Each thread should execute function print () and display its thread ID. The output should be like:
Hello I am thread 1 my ID is 123
Hello I am thread 2 my ID is 234....
The main thread should wait for the child threads to terminate and then call exit.

Use pthread_self()
pthread_t ID= pthread_self (void);
Returns the unique thread ID of the calling thread

Task 2

Write a program which calculates a list of prime numbers in a given range. The main program (i.e. Main thread) is passed a range of numbers as a command line argument. **Main thread performs the following tasks.**

- It performs standard error checking on command line arguments.
- Divides the range into equal parts.
- Creates a set of worker threads.
- Passes each range as a parameter to worker threads.

Worker thread performs the following tasks

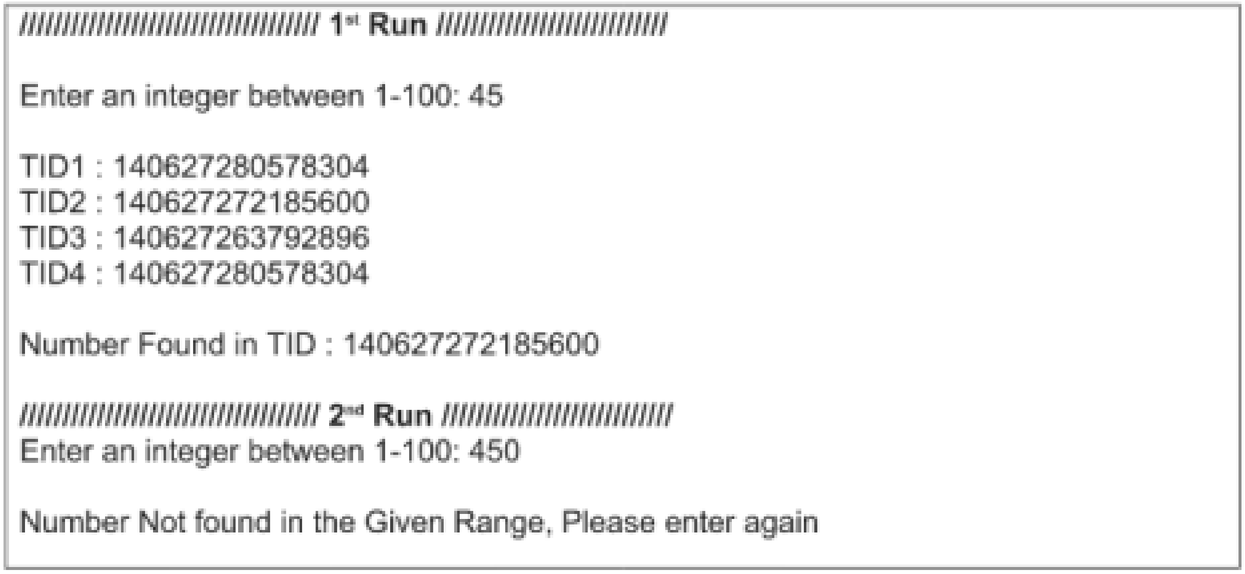
- It performs standard error checking on command line arguments.
- Divides the range into equal parts.

- Creates a set of worker threads.
- Passes each range as a parameter to worker threads.
- **Worker thread performs the following tasks**
    - Calculates prime numbers in the range passed as thread argument.
    - It returns back the prime number found to the main thread.

Task 3

1) Create a global array of size 100. Fill it from 1 – 100.
2) Take a number from user and search it in an array using 4 threads.
3) You have to divide searching areas between 4 threads. Array index from 0 - 24 will be assigned to Thread 1, 25 – 49 to Thread 2, and so on.
4) If any thread found the given number, print found and cancel all other threads which were searching for the same number. For example, if number to be found is 45, each thread will search in its own region, once this number found in Thread 2, there is no need to search this number in other threads anymore, so thread 2 must cancel all other threads immediately.
5) Show Thread Id of each thread first after that Print Thread Id of that thread along with its number such as TID1, TID2 etc. which has successfully found the required number. 6) If number not found in the given range, display a message that number does not lie in the given range.

Sample Output:



```
//////////////////////////////////// 1st Run //////////////////////////////

Enter an integer between 1-100: 45

TID1 : 140627280578304
TID2 : 140627272185600
TID3 : 140627263792896
TID4 : 140627280578304

Number Found in TID : 140627272185600

//////////////////////////////////// 2nd Run //////////////////////////////
Enter an integer between 1-100: 450

Number Not found in the Given Range, Please enter again
```

Task 4

Write C/C++ code that creates three 3 threads, First Thread creates a file, named ‘text.txt’, and writes some text data on it. While the second thread read the file and Capitalize first and last letter of each word in line, while the third thread reverses the each word in the line of file.

Thread One: Create the file text.txt and write text: “The quick brown fox jumps over lazy dog”.

Thread Two: “The QuicK BrowN FoX JumpS OveR LazY DoG” Thread Three: etaerC eht…………

Each thread creates and overrides its own file and stores intermediate data init.