CSI 5/7343 Operating Systems and System Software

Spring 2022

Programming Homework 1

Due date: 2/22 (Tue) 11:59pm. No late submissions.

You are given the following function, which is used for generate random numbers. (where a, c, and m are user-defined constants, x is the previous generated random number, and what is returned is the next random number)

```
int  f1(int x, int a, int c, int m) {
   long int x1 = x * a + c;
   return (int)(x1 % m);
}
```

You should write the following program:

The program will create a set of threads, where each thread will have its own value of (a, c, and m, and a an initial value of x) assign to. The program will then go on through a set of rounds. Each round each thread will generate a random number based on f1(). The thread that generated the largest and smallest number will get 1 point each. However, if there is a tie, then no one get a point.

The main process should keep tab on the points, and at the end, the thread that has the highest number of points is declared the winner.

Program structure: main process

The main process should first get the name of a file from the command line (using the argc/argv functionality of C). The file has the following format:

- The first line contains two numbers. The first is the number of threads to be created, and the second is the number of rounds.
- Then for each thread there is a line of four numbers, which is the value of a, c, m and the initial value of x for that thread.

(Hint: since the input is a fixed format, you should use fscanf() to read the input).

(Important: Points will be deducted if you hardcode the filename in your code; or if you ask the user for the filename when you run the program)

Once the file is read, the main process has the following logic:

Create all threads (each thread should be assigned a number from 0 to number_of_threads- - 1)
For each round
Message the threads to generate the next number
Wait for all the threads to send in the numbers
Score the round

Message the threads to quit
Print the final score for each thread, and print who is the winner

Program structure: threads

The structure for each thread is as follows

Wait to receive a message from the main process
While message is not quit
    Generate a number x_new by calling f1(x, a, c, m)
    Send x_new back to the main process
    Set x = x1new
Exit

Notice that message passing between main program and threads can be done by global variables.

Program Output

Your program should provide the following (and only the following) output:

- When the program starts, you should print the name of the file
- Right before the main process send a message for the start of a round, it should print "Main process start round <n>" (where <n> should be a number denoting the current round, The first round is denoted as round 1).
- Each time after a thread finish calling f1(), it should print the following: "Thread <id> call f1() returns <value>", where <id> is the thread number assigned by the main process (between 0 and number of threads – 1), and <value> is the value generated from f1(). (<id> have the same meaning for subsequent requirements).
- At the end of each round, once the round is scored, the main process should print "Round <n> finished. Winners are <id>", where <n> is the round number, and <id> is the list of thread id that wins a point, each thread id separated by a space. The threads are listed in increasing order of <id>. Notice that if no thread wins, then it should be empty.
- After all the rounds has finished, the main process should print, for each thread the following line: "Score for thread <id> : <score>", where <id> is the id of the thread, and the score is the score for each thread.
- And then the main process should print the following line: "Overall winner : <id>" where <Id> is the thread that have the highest score. If there is a tie, all the threads id are listed (in increasing order of the id), each of them separated by a space characters.

Each item above should be printed on a separate line. Trailing space characters are allowed.

What to hand in

You should name your program "hw1.c", and upload your source code ONLY to Canvas. Points will be deducted if you submit more than one file.