CS F372 Operating Systems
First Semester 2022-23

Assignment 1

Important guidelines:

- The assignment is to be done in groups of 6
- This assignment is for both the L1 and L2 sections. Groups may be formed across the sections
- All members of the group must contribute to the assignment. A demo-cum-viva session will be scheduled during which it will be ascertained that all members have contributed. If any member does not contribute, the member will be awarded 0 in the assignment
- **All submissions will be passed through a code-similarity checker. If the codes of two or more groups match then all the group members will be summarily awarded 0 in the assignment. This is irrespective of if only one member of a group is the offender. There is no partial penalty for dishonesty. Lifting code from the Internet also constitutes cheating**
- Honest but incorrect submissions will be awarded partial credit through the demo-cum-viva session
- Discussion is encouraged between groups, copying is strictly prohibited. Use Piazza for discussion. What is the difference? Check the last part of this page: https://www.cse.iitd.ac.in/~mausam/courses/col772/spring2019/
- Please keep checking this document to see if any new updates have been made. Updates will be highlighted.

Problem Statement:

- Write a C program (using C libraries for fork(), pthread and synchronisation) to solve the knight's tour problem given an NxN board and the starting position of the knight. Given the starting position of the knight on an NxN board the starting position will be a coordinate (X, Y) with X and Y between (N=0…N-1). Print out *a* possible path the knight can take such that it covers all the squares (cells) in that board exactly once. If no such path is possible print "No Possible Tour". The knight may move two cells vertically and one cell horizontally or two cells horizontally and one cell vertically. The program should be compatible on an Ubuntu machine running Ubuntu 18.04 or 20.04. Please note the following details:
    - 0<N<=50 The value of N will be supplied as a command line argument.
    - 0 <= X,Y < N the starting position of the knight on the grid, supplied as command line arguments, the positions are 0 indexed.
    - So, your program should run as : ./knight.out N X Y
    - The program should print a solved path of the tour. If no tour is possible from that start point print "No Possible Tour". Otherwise print the (x,y) 0 indexed position of the Knight at after each move in the tour till the entire tour is printed out.

- There can be multiple solutions for a given N X Y triplet. Printing one possible solution is sufficient.

  Examples:

  - N = 5, X = 1, Y = 1
    - Output:

      ```
      1,1|3,0|2,2|0,3|2,4|4,3|3,1|1,0|0,2|1,4|3,3|4,1|2,0|0,1|1,3|3,4|4,2|2,1|4,0|3,2|4,4|2,3|0,4|1,2|0,0|
      ```

  - N = 8, X = 3, Y = 2
    - Output:

      ```
      3,2|5,3|7,4|6,6|4,7|2,6|0,7|1,5|3,6|5,7|4,5|3,7|1,6|0,4|2,5|4,6|6,7|5,5
      |7,6|6,4|5,6|7,7|6,5|4,4|2,3|3,5|2,7|0,6|1,4|3,3|5,2|4,0|6,1|7,3|5,4|7,
      5|6,3|7,1|5,0|3,1|1,0|0,2|2,1|0,0|1,2|2,0|0,1|1,3|3,4|4,2|3,0|1,1|0,3|2
      ,2|4,1|6,0|7,2|5,1|7,0|6,2|4,3|2,4|0,5|1,7|
      ```

  - N = 3, X = 1, Y = 1
    - Output:

      ```
      No Possible Tour
      ```

      Explanation: Knight is immobile at this position and can’t move to any cell. (It is not required to print the explanation)

- A skeleton code has been provided here. Please keep the skeleton intact to ensure compatibility. Incompatible codes will be treated as incorrect.
- Your program should take advantage of parallelism offered by the Linux fork(), IPC, pthread and synchronisation libraries. It is compulsory to use either fork() or pthread or both. Your program should spawn multiple (as many as required) child processes and/or threads to divide and conquer the problem in parallel. Marking will be based on time taken to reach a solution. Single threaded/process solutions will not be awarded any credit.

How and What to submit:

- Evaluation and Submission Details:
  - Weightage: 15%
  - You will need to scp your code to a server. The credentials will be emailed to the group members. Other instructions will follow
  - Your code will be checked for correctness and that all the guidelines are followed. Groups which submit a correct solution will be ranked on the basis of execution time of the code and results will be displayed on a leaderboard.
  - The leaderboard will be updated at a fixed interval everyday. The link to view the leaderboard will be shared in due course.
  - A demo session will be conducted. Incorrect solutions will be awarded partial points during the demo. For groups with correct solutions, the approach and logic will be ascertained and more elegant solutions will be awarded more credit. It will also be ascertained that all members of the group have contributed. **Freeloaders will be summarily awarded 0 in the assignment.**

  - Some part of the scoring will be based on the rank of a group on the leaderboard at the end of the deadline to submit the assignment. Additional points will be awarded for programs utilizing lower memory during the demo.
  - **Deadline: 27th October, 2022, 23:59 hours.**

- File Naming Convention:
  - `groupX_assignment1.c` [Where X is the group no.]
- Late Day rules:
  - Each student gets 3 free late days in the semester. The 3 late days can be used by a student across the two assignments in any manner that she/he thinks fit (e.g. 1 day for assignment 1 and 2 for assignment 2). Beyond the 3 free days a 10% penalty per day will apply (partial days will be rounded up). The late days of an individual student will carry forward even if she/he changes the group for the next assignment.