



GRIFFITH COLLEGE DUBLIN

Course: Parallel and Distributed Programming

Module: PDP

Semester: Semester II

Assignment Number: 2 OpenMP and MPI C programming

Date of Title Issue: 28th November 2022

Assignment Deadline: 18th. December 2022

Assignment Weighting: 20/50

Please state the assignment title / brief. Please specify details such as: Answer the questions on the accompanying sheet.

Learning Outcomes

Please state the programme and related module learning outcomes that this assignment is assessing. 1,2,4, 5,6

Assessment Criteria

Please state the assessment criteria applied to this assignment, such as:

- Correctness of the work.
- Presentation, including compliance with the specified file format.
- Evidence of critical thinking and analysis.
- Originality, quality and thoroughness of the work.
- Research correct academic approach.
- Proper treatment of sources.

*Academic Dishonesty: All of your assignments need to represent your own effort. Assignments should be done without consultation with other students and you should not share your source code with others. Any assignment submitted that is essentially the same, as someone else's will not be accepted. **ALL matching assignments will receive 0 credits.***

Your task is to search for a **palindrome string of length (n)** in a two dimensional matrix of random single characters between a-z, the idea of the solution is to parallelize the search between all possible threads and count each occurrence. The matrix size is 1000x1000, you can input the number you want to search or hard code it, search right to left, up to down and diagonally up to down.

Make sure to include timing to your code, you need to submit two solutions using MPI and OpenMP:

- 1. OpenMP C file and results of execution. 50%**
- 2. Repeat part 1 using MPI C file and results of execution. 50%**

Code below can be used to generate the matrix:

```
for(i = 0; i < ROWS; i++)
  for(j = 0; j < COLUMNS; j++)
    a[i][j] = (rand() % 26) + 'A';
```

Hint: start your testing with a small matrix, 10x10 to make sure it is working, then scale up to 1000x1000.

You can use function for the search or string libraries.

Example of execution output:

Search for 6 letters words:

47345 words found

Execution time: 24 msec using 4 threads