UNIVERSITY OF RUHUNA

Faculty of Engineering

End-Semester 8 Examination in Engineering: February 2020

Module Number: EE8203 Module Name: High Performance Computing

[Three Hours]

[Answer **all questions**, each question carries 10 marks]

Q1 a) Flynn's taxonomy is a classification of computer architectures proposed in 1966. Briefly explain the four classifications defined by Flynn.

[2 Marks]

b) Explain how Peterson's algorithm satisfies mutual exclusion and bounded waiting for two process.

[2 Marks]

c) Dining philosopher problem is an example problem used in concurrent algorithm design to illustrate synchronization issues and techniques for resolving them.

i) Describe how the "Resource hierarchy solution" to the Dining philosopher problem avoids deadlock.

[1 Mark]

ii) Describe how the "Arbitrator solution" to the Dining philosopher problem guarantees that a philosopher can only pick up both forks or none by introducing an arbitrator.

[2 Marks]

iii) Describe how the solution proposed by K. Mani Chandy and J. Misra to the Dining philosopher problem solves starvation.

[1 Mark]

iv) Describe how the solution proposed by K. Mani Chandy and J. Misra to the Dining philosopher problem can leads to a deadlock.

[2 Marks]

Q2 a) Describe the Fork-Join parallelism used in OpenMp programming model.

[2 Marks]

b) Write a program using OpenMp to get the sum of an integer array containing 1000 elements, set the number of threads to 4.

[2 Marks]

c) Describe how to remove the implicit barrier available at the end of a work sharing construct of OpenMp.

[1 Mark]

d) What is the purpose of the atomic keyword used in OpenMp?

[2 Marks]

e) Write a program using OpenMp which does the following. Assume all A, B, C & D are arrays of 1000 elements. Set the number of threads as 4. Calculate the output array F using the following equation.

$$F[i] = 2*(5*(A[i]+B[i])+7*(C[i]+D[i]))$$

Where i changes from 0 - 999

[3 Marks]

Q3 a) Write the output of the following MPI program if the number of tasks are 2.

```
#include "mpi.h"
#include <stdio.h>
#include <stdlib.h>
int main (int argc, char *argv[])
{
  int numtasks, rank, dest, source, rc, count, tag=1;
  char inmsg, outmsg='x';
  MPI_Status Stat;
  MPI_Init(&argc,&argv);
  MPI_Comm_size(MPI_COMM_WORLD, &numtasks);
  MPI_Comm_rank(MPI_COMM_WORLD, &rank);
  if (rank == 0) {
    dest = 1;
    source = 1;
    outmsg='y';
    rc = MPI_Recv(&inmsg, 1, MPI_CHAR, source, tag, MPI_COMM_WORLD, &Stat);
    rc = MPI_Send(&outmsg, 1, MPI_CHAR, dest, tag, MPI_COMM_WORLD);
    printf("The character %c sent to rank 1",outmsg);
  }
  else if (rank == 1) {
    dest = 0;
    source = 0;
    rc = MPI_Recv(&inmsg, 1, MPI_CHAR, source, tag, MPI_COMM_WORLD, &Stat);
    rc = MPI_Send(&outmsg, 1, MPI_CHAR, dest, tag, MPI_COMM_WORLD);
    printf("The character %c sent to rank 0",outmsg);
  }
  MPI_Finalize();
}
```

[2 Marks]

b) Write a programme using MPI which does the following.
Assume there are 5 tasks running and the task having "rank 0" will send the number **one** to all the other tasks. All the other tasks have to wait until it receives the number **one** from "rank 0" task. Once the other tasks receive the number **one,** those tasks should send their rank as a number to the "rank 0" task. The "rank 0" task should print the number received from the other tasks.

[3 Marks]

c) Describe how it is possible to do Hybrid Parallel Programming using MPI.

[2 Marks]

d) What are the differences between blocking vs non-blocking versions of MPI Receive functions?

[2 Marks]

e) What are the advantages of MPI_Reduce operation over MPI_Send and MPI_Recv operations?

[1 Mark]

Q4 a) What are the differences in hardware architectures of CPU and GPU?

[1 Mark]

b) Write a program using CUDA to do the following.
Threshold a 1024 x 1024 image by the intensity value 100. Use one thread per-pixel and if the intensity value is > 100, mark the corresponding output image pixel as **one** and if the intensity value <=100 mark it as **zero**.

[3 Marks]

c) What are the advantages of using Shared Memory in the CUDA Programming model.

[2 Marks]

d) Describe what is Heterogeneous Programming in the CUDA Programming Model.

[2 Marks]

e) Describe what are the advantages of using a thread pool in C# programming?

[2 Marks]

Q5 a) Write a possible output of the following program written using POSIX threads.

```
#include <pthread.h>
#include <stdio.h>
#include <stdlib.h>
#define NUM_THREADS     5
int count = 0;
void *Counter(void *threadid) {
  int tid = (int) threadid;
  for (int i=0; i<tid; i++){
    count++;
  }
}
```

```
int main (int argc, char *argv[])
{
  pthread_t thread[NUM_THREADS];
  pthread_attr_t attr;
  int rc;
  long t;
  void *status;
  pthread_attr_init(&attr);
  pthread_attr_setdetachstate(&attr, PTHREAD_CREATE_JOINABLE);
  for(t=0; t<NUM_THREADS; t++) {
    rc = pthread_create(&thread[t], &attr, Counter, (void *)t);
  }
  pthread_attr_destroy(&attr);
  for(t=0; t<NUM_THREADS; t++) {
    rc = pthread_join(thread[t], &status);
  }
 printf("The final count is:%d\n", count);
 pthread_exit(NULL);
}
```

[2 Marks]

b) Write a program using POSIX threads which does the following.

There are two matrices A & B of having 4 rows and 1000 columns. Create 4 threads to add those two matrices and store it in a matrix named C of the same size.

[2 Marks]

c) Describe what are Condition Variables and how those are used with POSIX threads.

[2 Marks]

d) What are the special features available with Hadoop Distributed File System (HDFS) compared to a local file system.

[2 Marks]

e) What are functionalities of the Map and Reduce methods used in the Mapper and the Reducer classes in the MapReduce programming model.

[2 Marks]