

ITCS 4145/5145 Parallel Computing
Test 2
5:00 pm - 6:15 pm, Thursday October 28, 2010

Name:

This is a closed book test. Do not refer to any materials except those supplied for the test.

Supplied: “*Summary of OpenMP 3.0 C/C++ Syntax.*”

Answer questions in space provided below questions. Use additional paper if necessary but make sure your name is on every sheet.

Total /50

Qu. 1 Answer each of the following briefly:

(a) Give one hardware feature of a shared memory system that does not exist in a distributed message passing system. 2

(b) What does the Unix/Linux system call:

`pid = fork();`

do?

2

(c) What is meant by a thread-safe routine? 2

(d) In the following code:

```
a = b - 5;  
x = x * 4;  
p = x * 9;
```

Why might a compiler or a processor re-order instructions so that they are not in program order when executed?

2

(e) What is a mutex in Pthreads?

2

(f) Why can critical sections cause parallel code sequences on different processors to be executed serially (one after the other in time)?

2

(g) Use Bernstein's conditions to determine whether the two code sequences:

```
forall (i = 0, j = 4; i < 4; i++, j--)  
    a[i] = a[j+3];
```

```
for (i = 0, j = 4; i < 4; i++, j--)  
    a[i] = a[j+3];
```

always produce the same results.

4

(h) Fully explain the following Java code:

```
public class HelloThread extends Thread {  
  
    public void run() {  
        System.out.println("Hello from a thread!");  
    }  
  
    public static void main(String args[ ]) {  
        (new HelloThread()).start();  
    }  
}
```

2

(i) Give one reason why one might create threads in Java by implementing the interface Runnable rather than extending the class Thread.

2

(j) In Assignment 3, one needed to synchronise threads so that they all complete one iteration before they start the next iteration. How can one achieve that?

2

(k) Fully explain the following OpenMP code sequence:

4

```
#pragma omp parallel shared(a,b,c,d) private(i)
{
    #pragma omp sections
    {
        #pragma omp section
        {
            for (i=0; i<N; i++) {
                c[i] = a[i] + b[i];
            }
        }
        #pragma omp section
        {
            for (i=0; i<N; i++) {
                d[i] = a[i] * b[i];
            }
        }
    }
}
```

(l) Fully explain the following OpenMP code sequence:

2

```
#pragma omp parallel shared(a,b,c) private(i)
{
    #pragma omp for
    for (i=0; i<N; i++) {
        c[i] = a[i] + b[i];
    }
}
```

(m) In Assignment 4, you are asked to log into coit-grid01 and then issue the command

2

```
ssh coit-grid05.uncc.edu
```

Why?

(n) Explain how to multiply two matrices using the systolic array approach.

4

(o) Parallelizing Gaussian elimination can lead to poor processor utilization. Why and how can the utilization be improved?

2

(p) What is a Jacobi iteration (that is, its basic characteristic compared to other types of iterations such as Gauss-Seidel relaxation)?

2

Supplied: “*Summary of OpenMP 3.0 C/C++ Syntax.*”

Qu. 2 Write an OpenMP program that adds two 256 x 256 matrices together using 16 threads. Provide comments in your code to help the grader! Briefly describe your method. ***If I do not understand the code, I will assume it is incorrect.***

12