

ECE 3058
Thread Scheduling Lab

Name: Wilson Williams
GT Username: wwilliams79

Problem 1B

Execution Times:

1 CPU: 67.6 s

2 CPUs: 36.0 s

4 CPUs: 33.2 s

From the data, there is not a linear relationship between the number of CPUs and the total execution time. There is a much larger drop in execution time from 1 CPU to 2 CPUs than there is for 2 CPUs to 4 CPUs. With 1 CPU, the controlling factor is that there is only 1 CPU to work with. With multiple CPUs though, the controlling factor is then shifted to how efficient the program can be executed. More CPUs doesn't necessarily mean sharper decreases in execution time because you don't know if the program can be split up among those CPUs to run quicker.

Problem 2B

800 ms:

Context Switches: 135

Execution Time: 67.6 s

Time Spent in Ready State: 328.2 s

600 ms:

Context Switches: 161

Execution Time: 67.6 s

Time Spent in Ready State: 314.0 s

400 ms:

Context Switches: 203

Execution Time: 67.6 s

Time Spent in Ready State: 300.3 s

200 ms:

Context Switches: 361

Execution Time: 67.6 s

Time Spent in Ready State: 289.8 s

As you can see from the data, the total time spent in the ready state decreases as the time slices decrease while the amount of context switches increases and the execution time remains constant. However, in a real OS, the time slice shouldn't be too small. This is because a really

small time slice will cause way too many context switches to a point in which the CPUs efficiency will be minimized. Finding an ideal time slice will maximize CPU efficiency, however this time shouldn't be too small or too large.

Problem 3B

No Problem 3B