CSE 422- Studio 8

1. Selamawit Tegegn(sntegegn@wustl.edu) , Leiquan Pan(lpan22@wustl.edu), yiying Lu([lu.yiying@wustl.edu](mailto:lu.yiying@wustl.edu))

2.

Output1: 1

Output2: 0

3.

Top:

PID USER PR NI VIRT RES SHR S %CPU %MEM TIME+ COMMAND P

1751 pi 20 0 1676 324 264 R 100.0 0.0 1:11.60 studio8 3

Trace-cmd:

execve("./studio8", ["./studio8", "3"], [/\* 53 vars \*/]) = 0

4.

When we are running at 4 cores at the same time, opening a new browser takes a longer time. When we terminate the tasks one by one, we noticed that the browser would run faster.

5.

pi@raspberrypi:~/test\_programs $ time ./dense\_mm 300

Generating matrices...

Multiplying matrices...

Multiplication done!

real 0m2.172s

user 0m2.160s

sys 0m0.000s

running studio8 in one processor

pi@raspberrypi:~/test\_programs $ time ./dense\_mm 300

Generating matrices...

Multiplying matrices...

Multiplication done!

real 0m2.152s

user 0m2.140s

sys 0m0.000s

running studio8 in two processors

pi@raspberrypi:~/test\_programs $ time ./dense\_mm 300

Generating matrices...

Multiplying matrices...

Multiplication done!

real 0m2.087s

user 0m2.070s

sys 0m0.000s

running studio8 in three processors

pi@raspberrypi:~/test\_programs $ time ./dense\_mm 300

Generating matrices...

Multiplying matrices...

Multiplication done!

real 0m2.272s

user 0m2.060s

sys 0m0.010s

running studio8 in four processors

pi@raspberrypi:~/test\_programs $ time ./dense\_mm 300

Generating matrices...

Multiplying matrices...

Multiplication done!

real 0m2.854s

user 0m2.080s

sys 0m0.010s

6.

running studio8 in 0 core

pi@raspberrypi:~/test\_programs $ time ./dense\_mm 300

Generating matrices...

Multiplying matrices...

Multiplication done!

real 0m2.070s

user 0m2.060s

sys 0m0.000s

two tasks running in 0 core

pi@raspberrypi:~/test\_programs $ time ./dense\_mm 300

Generating matrices...

Multiplying matrices...

Multiplication done!

real 0m2.080s

user 0m2.070s

sys 0m0.010s

Conclusion: When there are one or two tasks running in one processor, the real, user times doesn’t change a lot.

Prediction: No matter how many tasks are running in one processor, the real, user times doesn’t change a lot.

7.

Four tasks running in 0 core

pi@raspberrypi:~/test\_programs $ time ./dense\_mm 300

Generating matrices...

Multiplying matrices...

Multiplication done!

real 0m2.076s

user 0m2.070s

sys 0m0.000s

The result support our prediction. If we run multiple tasks in one processor, the program can still find a processor to execute.

8.

pi@raspberrypi:~/test\_programs $ time sudo nice -n -20 ./dense\_mm 300

Generating matrices...

Multiplying matrices...

Multiplication done!

real 0m2.198s

user 0m2.140s

sys 0m0.020s

Proportion: user/real = 0.9736

9.

Priority: -10

Proportion: 0.7992

pi@raspberrypi:~/test\_programs $ time sudo nice -n -10 ./dense\_mm 300

Generating matrices...

Multiplying matrices…

Multiplication done!

real 0m2.640s

user 0m2.110s

sys 0m0.000s

Priority: -5

Proportion: 0.7000

pi@raspberrypi:~/test\_programs $ time sudo nice -n -5 ./dense\_mm 300

Generating matrices...

Multiplying matrices...

Multiplication done!

real 0m3.257s

user 0m2.280s

sys 0m0.010s

Priority: 0

Proportion: 0.4677

pi@raspberrypi:~/test\_programs $ time sudo nice -n 0 ./dense\_mm 300

Generating matrices...

Multiplying matrices...

Multiplication done!

real 0m5.089s

user 0m2.380s

sys 0m0.000s

Priority: 5

Proportion: 0.2378

pi@raspberrypi:~/test\_programs $ time sudo nice -n 5 ./dense\_mm 300

Generating matrices...

Multiplying matrices...

Multiplication done!

real 0m10.260s

user 0m2.440s

sys 0m0.010s

Priority: 10

Proportion: 0.0097

pi@raspberrypi:~/test\_programs $ time sudo nice -n 10 ./dense\_mm 300

Generating matrices...

Multiplying matrices...

Multiplication done!

real 0m26.050s

user 0m2.520s

sys 0m0.000s

Priority: 19

Proportion: 0.004

pi@raspberrypi:~/test\_programs $ time sudo nice -n 19 ./dense\_mm 300

Generating matrices...

Multiplying matrices...

real 0m15.428s

user 0m0.230s

sys 0m0.000s