Steven Canon-Almagro

10155792 - CPSC457 - ASG3

Q1.

CPU utilization = 1 – (0.75 * 0.75 * 0.75 * 0.75 * 0.75 * 0.75 * 0.75 * 0.75)

CPU utilization = 0.8999 = 89.99%

Q2.

Processes	Arrival	Burst	Start	Finish	Turnaround	Waiting
P1	0	12	0	22	22	10
P2	2	1	2	3	1	0
P3	3	3	3	6	3	0
P4	5	1	6	7	2	1
P5	9	5	9	14	5	0

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22

1	1	2	3	3	3	4	1	1	5	5	5	5	5	1	1	1	1	1	1	1	1

Average wait time: (10+0+0+1+0)/5 = 2.2s

Steven Canon-Almagro

10155792 - CPSC457 - ASG3

Q3.

Processes	Arrival	Burst	Start	Finish	Turnaround	Waiting
P1	0	12	0	22	22	10
P2	2	1	3	4	2	1
P3	3	3	5	11	8	5
P4	5	1	7	8	3	2
P5	9	5	11	20	11	6

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22

1	1	1	2	1	3	1	4	3	1	3	5	1	5	1	5	1	5	1	5	1	1

There are 19 context switched in this RR scheduled example

Steven Canon-Almagro

10155792 - CPSC457 - ASG3

Q4.

Processes	Arrival	Burst	Start	Finish	Turnaround	Waiting
P1	0	12	0	22	22	10
P2	2	1	11	16	7	2
P3	3	3	6	7	5	4
P4	5	1	7	8	3	2
P5	9	5	8	11	8	5

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22

1	1	1	1	1	1	2	4	3	3	3	5	5	5	5	5	1	1	1	1	1	1

Q6.

Test file: medium.txt			
#threads	**Observed timing**	**Observed speedup compared to original**	**Expected speedup**
Original program	20.696s	1.0	1.0
1	20.678s	1.0	1.0
2	11.993s	1.7	2.0
3	8.202s	2.5	3.0
4	6.286s	3.2	4.0
8	5.109s	4.0	8.0
16	5.105s	4.0	16.0

Steven Canon-Almagro

10155792 - CPSC457 - ASG3

Test file: hard.txt			
#threads	**Observed timing**	**Observed speedup compared to original**	**Expected speedup**
Original program	7.046s	1.0	1.0
1	7.187s	1.0	1.0
2	3.610s	2.0	2.0
3	2.450s	2.9	3.0
4	1.886s	3.7	4.0
8	1.529s	4.6	8.0
16	1.525s	4.6	16.0

Test file: hard2.txt			
#threads	**Observed timing**	**Observed speedup compared to original**	**Expected speedup**
Original program	7.050s	1.0	1.0
1	7.095s	1.0	1.0
2	3.580s	2.0	2.0
3	2.446s	2.9	3.0
4	1.901s	3.7	4.0
8	1.547s	4.6	8.0
16	1.546s	4.6	16.0

Steven Canon-Almagro

10155792 - CPSC457 - ASG3

The results were close to what I expected the reason why there is such a large difference between the expected speed up to the actual speed up, as the number of threads increases is because each thread checks if a partition of the square of that number is not prime and if its not then it changes a variable that stops the prime counter from going up. Therefore the reason why the speed up isn't as good with larger threads is because the threads aren't stopped when a number is found to be not prime so the program needs to wait for all the threads to finish their execution.