

Assignment 3

Satyaki Ghosh

10077685

T #01

1. We can calculate the CPU utilization using this formula. CPU Utilization = $1 - p^n$, where p is the fraction of time the process is waiting on I/O and n is the number of processes. Thus, if we are running 3 programs that spend 80% of the time doing I/O then the CPU utilization is $1 - 0.8^3 = 0.488 = 48.8\%$

2. SRTN Gantt Chart

Process	Arrival Time	Burst Time	Start	Finish	Turnaround	Waiting
P1	0	10	0	14	14	4
P2	1	1	1	2	1	0
P3	3	2	3	5	2	0
P4	5	1	5	6	5	0
P5	9	5	14	19	10	5

The average wait time is 1.8

3. RR Gantt Chart

Process	Arrival Time	Burst Time	Start	Finish	Turnaround	Waiting
P1	0	10	0	19	19	9
P2	1	1	1	2	1	0
P3	3	2	3	6	3	1
P4	5	1	7	8	3	2
P5	9	5	9	18	9	4

There are 18 context switches.

4. A

5. Solution in file count.cpp

6. Timings for original and the multi-threaded solution for counting the number of primes in a file.

Test File: medium.txt			
# threads	Observed Timing (s)	Observed Speedup compared to original	Expected Speedup
Original	45.210	1.0	1.0
1	45.208	1.0	1.0
2	24.797	1.82	2.0
3	19.486	2.32	3.0
4	16.444	2.75	4.0
8	14.517	3.11	8.0
16	14.285	3.16	16.0

Test File: hard.txt			
# threads	Observed Timing (s)	Observed Speedup compared to original	Expected Speedup
Original	15.390	1.0	1.0
1	15.388	1.0	1.0
2	15.387	1.0	2.0
3	15.388	1.0	3.0
4	15.389	1.0	4.0
8	15.380	1.0	8.0
16	15.385	1.0	16.0

Test File: hard2.txt			
# threads	Observed Timing (s)	Observed Speedup compared to original	Expected Speedup
Original	15.389	1.0	1.0
1	15.390	1.0	1.0
2	15.393	1.0	2.0
3	15.390	1.0	3.0
4	15.389	1.0	4.0
8	15.380	1.0	8.0
16	15.385	1.0	16.0

The timings are not what I expected them to be hard.txt and hard2.txt, but they are what I expected for medium.txt. The timings are different because it's easier to speed up the counting primes process if there are a lot of numbers to test. But if there are only a few numbers to test then the multiple threads don't add that much value to speeding up the process. Even if we have multiple threads, each number can only be tested by a single thread, so if we have 2 numbers to test and 10 threads only 2 of the threads will be utilized to test the 2 numbers, making the other 8 threads useless. Thus, if we have a few numbers to test for primes, then we don't require a lot of threads to speed up the process. If there are a lot of numbers to test, then more threads will speed up the process substantially.

7. Solution in file scheduler.cpp