NachOS - Assignment 2

Scheduling Algorithms

Group 1
Vaibhav Nagar (14785)
Tushant Mittal(14762)
Borkar Pranay Shrihari (14189)

We denote different algorithms with numbers as given in the table. We wrote a bash script

Algorithm Number	Algorithm Description
1	Non-preemptive First Come First Serve
2	Non-preemptive Shortest Next CPU Burst First
3	Preemptive Round Robin with quantum 31
4	Preemptive Round Robin with quantum 62
5	Preemptive Round Robin with quantum 93
6	Preemptive Round Robin with quantum 20
7	Preemptive UNIX Scheduler with quantum 31
8	Preemptive UNIX Scheduler with quantum 62
9	Preemptive UNIX Scheduler with quantum 93
10	Preemptive UNIX Scheduler with quantum 20

maxutil.sh in userprog directory which stores the CPU utilization of Round Robin corresponding to different values of quantas from 11 to 100 and store them in 1.txt and find that the CPU utilization is max for quantum value of 11. But since we are supposed to pick Quanta_4 above 20 we choose Quanta_4 to be 20

- Average CPU burst of testloop.c = 124
- Optimised quantum value for Round-robin(3) = 11
- Quanta_4 = 20

1. Part 1

1.1 Batch 1

Scheduling Policy	Quanta	CPU Utilization	Average Wait Time
1	100	56.21	159516
2	100	56.21	159516
3	31	66.09	192628
4	62	61.93	168038
5	93	60.23	163812
6	20	72.06	235486
7	31	82.49	129656
8	62	75.31	121856
9	93	67.62	137749
10	20	92.68	138349

1.2 Batch 2

Scheduling Policy	Quanta	CPU Utilization	Average Wait Time
1	100	82.86	101035
2	100	82.86	101035
3	31	90.56	132922
4	62	88.64	109814
5	93	86.82	106447
6	20	92.61	176093
7	31	96.46	97880
8	62	95.68	80408
9	93	95.39	81657
10	20	98.37	116280

1.3 Batch 3

Scheduling Policy	Quanta	CPU Utilization	Average Wait Time
1	100	94.72	86735
2	100	94.72	86735
3	31	98.96	120282
4	62	98.98	95612
5	93	98.43	91525
6	20	99.17	162708
7	31	99.73	91250
8	62	99.43	73854
9	93	99.63	70632
10	20	99.77	112194

1.4 Batch 4

Scheduling Policy	Quanta	CPU Utilization	Average Wait Time
1	100	99.86	44693
2	100	99.86	44693
3	31	99.88	119453
4	62	99.87	96171
5	93	99.86	90464
6	20	99.87	161538
7	31	99.88	89851
8	62	99.87	72444
9	93	99.86	68981
10	20	99.87	110358

1.5 Inference

We can see that we get the maximum utilization by keeping quanta to be very less that is around 20 ticks. This might be because lesser quanta implies that .But the drawback is that small values of quanta leads to large number of context switches which wastes a lot of CPU time.

2. Part 2

Algorithm used	CPU Utilization	Average waiting time
1	99.82	61622
2	99.82	46367

2.1 Inference

Although the CPU utilization is very high for both the schedulers the average waiting time is lesser for the Non-preemptive algorithm with burst estimation. This is because the algorithm effectively shuffles around the processes and each process waits less compared to the default FIFO wherein the process that is at the queue waits for a very long time.

3. Part 3

Batch	OUTER_BOUND=4	OUTER_BOUND=8	OUTER_BOUND=12
1	1.41 (724)	1.17 (782)	1.08 (824)
2	1.50 (190)	1.21 (230)	1.11 (270)
3	1.41 (190)	1.16 (100)	1.07 (140)
4	0.99 (10)	0.99 (10)	0.99 (10)
5	0.68 (50)	0.36 (90)	0.24 (130)

Table 1: Estimation error in CPU bursts (value in bracket is Number of CPU bursts)

3.1 Inference

We see that the error in estimating the CPU burst time decreases. This is because as we increase the OUTER BOUND value the number of CPU bursts increases. This is a ramification of the fact that the exponential estimator fits better as it sees more bursts (more data) and and thus produces better approximations.

4. Part 4

4.1 Round-robin

- Maximum of job completion time = 257614
- Minimum of job completion time = 250384
- Average of job completion time = 25765
- Variance of job completion time = 194942892

4.2 UNIX scheduling algorithm

- Maximum of job completion time = 198041
- Minimum of job completion time = 49376
- Average of job completion time = 19805
- Variance of job completion time = 265187416

4.3 Inference

It can be seen that the minimum job completion time decreases significantly i.e. it is one-fifth in the UNIX scheduler compared to Round Robin. This is because the UNIX scheduler also incorporates priorities whereas the Round Robin doesn't. So, the process with priority 100 gets the CPU for most of the time and thus finishes quickly whereas in Round Robin the process enjoys no such privilege. This is also the reason why the variance is higher as processes with different priorities have different times but are closer to each other in Round Robin.