Design Assignment 2 - CS330

Ginuga Saketh 150257 Arjun Sinha 150134 Shubh Gupta 14670
Deepak Gangwar 14208
Group No˙ 11

October 26, 2017

1 Part I

Values of quantums used are:

- $\frac{1}{4}^{th}$ quantum: 32
- $\frac{1}{2}^{th}$ quantum: 65
- $\frac{3}{4}^{th}$ quantum: 96
- Minimum value of quantum for maximum CPU utilization: 20

1.1 Batch 1

CPU utilization is as follows:

S.No.	Scheduling Algorithm	CPU Utilization
1	Non-Preemptive Default Nachos Scheduling	56
2	Non-Preemptive Shortest Burst First	56
3	Round Robin with 1/4 quantum	72
4	Round Robin with 1/2 quantum	65
5	Round Robin with 3/4 quantum	62
6	Round Robin with minimum quantum	80
7	UNIX Scheduler with 1/4 quantum	75
8	UNIX Scheduler with 1/2 quantum	68
9	UNIX Scheduler with 3/4 quantum	64
10	UNIX Scheduler with minimum quantum	85

Average Wait Time is as follows:

S.No.	Scheduling Algorithm	Average Wait Time
1	Non-Preemptive Default Nachos Scheduling	22190
2	Non-Preemptive Shortest Burst First	22190
3	Round Robin with 1/4 quantum	73260
4	Round Robin with 1/2 quantum	60584
5	Round Robin with 3/4 quantum	56965
6	Round Robin with minimum quantum	101678
7	UNIX Scheduler with 1/4 quantum	72009
8	UNIX Scheduler with 1/2 quantum	59091
9	UNIX Scheduler with 3/4 quantum	55334
10	UNIX Scheduler with minimum quantum	100610

1.2 Batch 2

CPU utilization is as follows:

S.No.	Scheduling Algorithm	CPU Utilization
1	Non-Preemptive Default Nachos Scheduling	82
2	Non-Preemptive Shortest Burst First	82
3	Round Robin with 1/4 quantum	96
4	Round Robin with 1/2 quantum	97
5	Round Robin with 3/4 quantum	93
6	Round Robin with minimum quantum	97
7	UNIX Scheduler with 1/4 quantum	97
8	UNIX Scheduler with 1/2 quantum	96
9	UNIX Scheduler with 3/4 quantum	94
10	UNIX Scheduler with minimum quantum	97

Average Wait Time is as follows:

S.No.	Scheduling Algorithm	Average Wait Time
1	Non-Preemptive Default Nachos Scheduling	22141
2	Non-Preemptive Shortest Burst First	22141
3	Round Robin with 1/4 quantum	74330
4	Round Robin with 1/2 quantum	60665
5	Round Robin with 3/4 quantum	57879
6	Round Robin with minimum quantum	101566
7	UNIX Scheduler with 1/4 quantum	72937
8	UNIX Scheduler with 1/2 quantum	58613
9	UNIX Scheduler with 3/4 quantum	56621
10	UNIX Scheduler with minimum quantum	100371

1.3 Batch 3

CPU utilization is as follows:

S.No.	Scheduling Algorithm	CPU Utilization
1	Non-Preemptive Default Nachos Scheduling	94
2	Non-Preemptive Shortest Burst First	94
3	Round Robin with 1/4 quantum	99
4	Round Robin with 1/2 quantum	99
5	Round Robin with 3/4 quantum	99
6	Round Robin with minimum quantum	99
7	UNIX Scheduler with 1/4 quantum	99
8	UNIX Scheduler with 1/2 quantum	99
9	UNIX Scheduler with 3/4 quantum	99
10	UNIX Scheduler with minimum quantum	99

Average Wait Time is as follows:

S.No.	Scheduling Algorithm	Average Wait Time
1	Non-Preemptive Default Nachos Scheduling	22141
2	Non-Preemptive Shortest Burst First	22141
3	Round Robin with 1/4 quantum	73517
4	Round Robin with 1/2 quantum	59578
5	Round Robin with 3/4 quantum	57971
6	Round Robin with minimum quantum	101181
7	UNIX Scheduler with 1/4 quantum	73505
8	UNIX Scheduler with 1/2 quantum	58520
9	UNIX Scheduler with 3/4 quantum	56321
10	UNIX Scheduler with minimum quantum	100167

1.4 Batch 4

CPU utilization is as follows:

S.No.	Scheduling Algorithm	CPU Utilization
1	Non-Preemptive Default Nachos Scheduling	100
2	Non-Preemptive Shortest Burst First	100
3	Round Robin with 1/4 quantum	100
4	Round Robin with 1/2 quantum	100
5	Round Robin with 3/4 quantum	100
6	Round Robin with minimum quantum	100
7	UNIX Scheduler with 1/4 quantum	100
8	UNIX Scheduler with 1/2 quantum	100
9	UNIX Scheduler with 3/4 quantum	100
10	UNIX Scheduler with minimum quantum	100

Average Wait Time is as follows:

S.No.	Scheduling Algorithm	Average Wait Time
1	Non-Preemptive Default Nachos Scheduling	33251
2	Non-Preemptive Shortest Burst First	33251
3	Round Robin with 1/4 quantum	73951
4	Round Robin with 1/2 quantum	60159
5	Round Robin with 3/4 quantum	56777
6	Round Robin with minimum quantum	101682
7	UNIX Scheduler with 1/4 quantum	73990
8	UNIX Scheduler with 1/2 quantum	60097
9	UNIX Scheduler with 3/4 quantum	56612
10	UNIX Scheduler with minimum quantum	101699

We observe that on minimizing the value of the quanta, we get maximum CPU utilization. Smaller the quanta, leads to more number of context switches. But however, this definition of CPU utilization doesn't involve work done in kernel mode, and thus we get maxed CPU utilization. For smaller value of quanta, the increase in context switches leads to a large overhead which is undesirable.

As to why smaller values of quanta leads to maximum utilization - due to the increase in the number of context switches, the CPU burst time as well as total time increases, but so does the ratio of the CPU burst time to total time (very crudely put, it is somewhat like $\frac{p+1}{q+1} > \frac{p}{q}$).

2 Part II

For Batch 5, the Average waiting time results are as follows:

- Non-Preemptive Default Nachos Scheduling Algorithm: **36482**
- Non-Preemptive Shortest Burst First Scheduling Algorithm: **33268**

We can see that the Shortest Burst First scheduling algorithm has the smaller average waiting time (i.e smaller sum of waiting times), than the naive non-preemptive Nachos scheduling algorithm. Similar to the question in the mid-semester examination, scheduling jobs with smaller CPU bursts earlier, leads to less waiting for the other threads, as can be seen by an exchange argument (if we have 2 jobs with CPU bursts **a** and **b**, with $a < b$, then scheduling a before b leads to lesser waiting times, than scheduling b before a).

3 Part III

The various errors in the estimation of CPU bursts are mentioned below:

Batch No.	Estimate Error in CPU Burst
Batch 1	0.861970
Batch 2	0.913513
Batch 3	0.810605
Batch 4	1.000000
Batch 5	0.694081

On increasing OUTER_BOUND, we see that the error decreases. This is because on increasing OUTER_BOUND, the number of IO bursts, and likewise CPU bursts increase, and the exponential estimation keeps getting closer to the real estimate.

4 Part IV

The quantum used in the calculation of the following stats is **100**.

The statistics for differentiating between the 2 preemptive algorithms are as follows:

Scheduling Algorithm	Maximum	Minimum	Average	Variance
Round Robin	68210	5692	33078	411933324
UNIX scheduler	68200	9948	36675	434216271

We can see that the maximum time is around same for both the algorithms. This is because the maximum time is approximately the sum of running times of all jobs (with scheduling and context switch overhead).

The variance for the UNIX scheduler is more than for the Round Robin scheduler. This is because for the UNIX scheduler we complete jobs with values of lower priority earlier, than scheduling other jobs. So, we have jobs with lower priority completed much earlier, and then the jobs with higer priority are scheduled. Hence, the running times are far removed from the mean, and hence the variance is more. This is unlike in Round Robin, where jobs are always scheduled after some quanta have passed.