CPU Scheduling Custom Algorithm

Optimize the Response time and Turnaround Time
Based on the Processes' Priority

Qin Yang

October 28, 2018

1 Introduction

I design this algorithm through optimizing the turnaround time and response time based on the processes' priority. Since considering about in more real case, in order to increase the CPU's ultilization, we can not only care about the one of those factors, but also consider about the efficiency of the whole system.

So we let us divide the processes into several categories based on the priority of the processes, the response time and turnaround time of threads. And the general categories are shown as follow:

1) High priority, short response time and turnaround time;
2) High priority, long response time and short turnaround time;
3) High priority, short response time and long turnaround time;
4) High priority, long respone time and turnaround time;
5) low priority, short response time and turnaround time;
6) low priority, long response time and short turnaround time;
7) low priority, short response time and long turnaround time;
8) low priority, long respone time and turnaround time.

Considering the scheduling fair of the algorithm, we should let every kinds of threads have opportunity running in the CPU. So we can build a new method to calculate the weight of each thread, when the algorithm schedule the next thread. And the formula can repesent as follow:

$$Weight = Priority * Response\ Time + Priority * Turnaround\ Time \tag{1}$$

So we can build a new queue based on the weight of each thread, and sort the new queue from the minimum to maximum. Then the scheduling algorithm just select the top thread(the minimum weight thread) of the queue everytime running in CPU. In this scenario, the thread with high priority, short response time and turnaround time will run firstly. Since

our algorithm is preemption, we can choose different time slice scheduling the next thread based on the dynamical weight, which can guarantee each kinds of threads have relative opportunity running in the CPU.

2 Analysing

At this step, we list the value of main parameters comparing the FCFS, RR, Priority with our algorithm.

Algorithm				FCFS	Priority	RR
						3
1	Priority	System Thread	Average Response Time	23.33	18.33	21.67
			Average Turnaround Time	94.67	68.33	109.33
		Interaction Thread	Average Response Time	10.00	34.50	9.50
			Average Turnaround Time	73.50	124.50	108.50
2	Total Elapsed Time			130.00	134.00	162.00
3	Total Service Time			53.00	53.00	53.00
4	Total I/O Time			34.00	34.00	34.00
5	Total Dispatch Time			69.00	81.00	101.00
6	Total Idle Time			8.00	0.00	8.00
7	CPU Utilization(%)			93.85	100.00	95.06
8	CPU Efficiency(%)			40.77	39.55	32.72

Table 1: Algorithm Parameter Comparing

Through the data of the table 1 and 2, we can clearly see that our algorithm has improve the ultilization of CPU and the whole system general performance. Especially to the low priority thread, they have more chances running in the CPU and their response time and turnaround time also reduce. But actually, we can also notice that selecting the different time slice, the system's performance also has deeply changed. So accoding to this input, we choose the time slice with six, which can get the relative best performance comparing with others.

Since our algorithm consider about different categories can run in the CPU fairly, and the starvation will also be limited.b

3 Conclusion

In general, through this experiment, we can realize that it is hard to define a best algorithm in CPU scheduling. Since in different scenario, facing various tasks and considering

CUSTOM								
Time Slice	1	2	3	4	5	6	7	8
Average Response Time	17.00	20.00	21.67	23.33	23.33	23.33	23.33	23.33
Average Turnaround Time	227.00	139.33	124.67	110.67	106.00	106.00	103.00	101.00
Average Response Time	8.50	9.00	9.50	10.00	10.00	10.00	10.00	10.00
Average Turnaround Time	119.50	113.00	91.00	76.00	86.00	86.00	83.00	81.50
Total Elapsed Time	255.00	194.00	163.00	144.00	138.00	138.00	135.00	132.00
Total Service Time	53.00	53.00	53.00	53.00	53.00	53.00	53.00	53.00
Total I/O Time	34.00	34.00	34.00	34.00	34.00	34.00	34.00	34.00
Total Dispatch Time	199.00	138.00	109.00	89.00	83.00	83.00	80.00	77.00
Total Idle Time	3.00	3.00	1.00	2.00	2.00	2.00	2.00	2.00
CPU Utilization(%)	98.82	98.45	99.39	98.61	98.55	98.55	98.52	98.48
CPU Efficiency(%)	20.78	27.32	32.52	36.81	38.41	38.41	39.26	40.15

Table 2: Custom Algorithm Parameter Comparing

kinds of factors we might need to choose suitable algorithm adapting them dynamically. And the most improtant factor is that we can not predict the future processes and get global information exactly.