Project 2
CS 342-Operating Systems

EMRE DERMAN - 21703508

Introduction

The term of thread refers to the thread of execution of a program.Threads are used to split the execution of a multiple concurrently running tasks. Although threads have significance importance for parallel computing, crucial issue about threading is synchronization. Since threads are usually allowed to access and manipulate shared data. Concurrent access could create a problem about manipulating data. Pthread library used to provide routines to create threads and tools to ensure synchronization.To handele the synchronization POSIX mutex variables used and the synchronization maintained.

Experiments

In experiments the number of bursts, average waiting time A and the average waiting time B are independent variables and the average waiting time for each algorithm was the dependent variables.The thread count that could work parallel kept 3 to increase the experiment speed.The aim of the experiments was to emphasize the effect of the number of bursts on the average waiting time. In the experiments the number of threads kept constants and the. Number of bursts created by each thread incremented and the result emphasized according to the appropriate algorithm. I interested on the effect of the burst count on the average waiting tiem.

Change Burst - Algorithm Average Waiting time correlation table figure-1

Burst	FCFS	SJF	PRIO	VRUNTIME
5	15(ms)	30(ms)	50(ms)	10(ms)
25	975(ms)	375(ms)	975(ms)	1050(ms)
50	5000(ms)	900(ms)	1550(ms)	2450(ms)
75	16500(ms)	3900(ms)	20325(ms)	6450(ms)
100	45300(ms)	6200(ms)	21800(ms)	17300(ms)
150	55650(ms)	15750(ms)	23550(ms)	58200(ms)

For such amount of bursts until 75 to use VRUNTIME algorithm seems the most effective algorithm to implement to the scheduler.However; for such an amount of bursts that over 75 to use SJF is better for the effectivity of the scheduler.

Change AvgA - Algorithm Average Waiting time correlation table figure -2

AVGA	FCFS	SJF	PRIO	VRUNTIME
400	31230(ms)	5310(ms)	11370(ms)	7320(ms)
650	1380(ms)	1230(ms)	1620(ms)	1140(ms)
850	1380(ms)	480(ms)	1830(ms)	1620(ms)
1150	870(ms)	645(ms)	810(ms)	1170(ms)
1450	1125(ms)	540(ms)	1470(ms)	1365(ms)

Change AvgB - Algorithm Average Waiting time correlation figure - 3

AVGB	FCFS	SJF	PRIO	VRUNTIME
100	5040(ms)	1290(ms)	1290(ms)	2430(ms)
250	915(ms)	840(ms)	2325(ms)	1005(ms)
500	2820(ms)	1245(ms)	1170(ms)	1140(ms)
750	4830(ms)	2445(ms)	2025(ms)	3510(ms)
1000	3330(ms)	1800(ms)	3930(ms)	5085(ms)

As it can been seen from figure 1, number of bursts have a strong correlation with the bursts waiting time, naturally. On the other hand the time incrementation is not same as other algorithms such as for FCFS the incrementation was dramatically after the 75 burst. There are several reasons for this.One is that the implemented algorithms does not cares the time length of the bursts therefore the longer bursts occludes the other process. As an example of heavy trucks could congests roads and blocks the faster cars(less length of bursts). Therefore the effect of the implementation has of utmost significance. For the VRUNTIME algorithm implementation since that was not the most effective algorithm to implement or use for scheduler such cases.However; it could be effective for such cases that the length of the bursts does not have a pattern. Also since the data(the waiting time) could be changeable since the execution of the multithreading processes could be differs according to the workload of the system.

Conclusion

As a result of our experiment, we conclude that an algorithm choice has of utmost significance to increase the performance without an increase in the number of threads.Therefore according to the average parameter the most suitable algorithm could change.Also in a synchronization problem, the mutex variables usage solves the issue. For the change in the averageB parameter the most suitable algorithm seems to use SJF and for the averageA parameter change using the same algorithm SJF could be the most effective solution for the general case.