

Análise das Estatísticas

Trabalho 02 - OSP cpu: com filas de prioridade

Gabriela Aimée Guimarães - 20131551

Abaixo estão os resultados de três execuções realizadas com o algoritmo de escalonamento com filas de prioridade implementado. Para cada execução, passou-se os parâmetros dos arquivos par.avg, par.low e par.high. Por último, foram passados os mesmos parâmetros para a última execução, utilizando OSP.demo, para efeito comparativo.

	Execução 1		
	avg	low	high
system throughput	2.85	5.3241	0.9248
average waiting time	3074.3869	794.5375	9449.5965
average turnaround time	3174.8248	887.2958	9610.2456
cpu utilization	0.3138	0.5366	0.2116
memory utilization	0.8636	0.5939	0.9421
	Execução 2		
	avg	low	high
system throughput	2.4989	5.525	0.7498
average waiting time	3996.6585	867.5691	10808.4815
average turnaround time	4112.7236	951.5894	10990.963
cpu utilization	0.3212	0.4975	0.2037
memory utilization	0.9524	0.529	0.9112
	Execução 3		
	avg	low	high
system throughput	2.3749	5.5472	0.575
average waiting time	3688.7265	881.2984	15196.1333
average turnaround time	3798.0513	975.0565	15388.0889
cpu utilization	0.3013	0.5465	0.2042
memory utilization	0.8901	0.6631	0.9714
	OSP.demo		

	avg	low	high
system throughput	1.975	4.475	0.3749
average waiting time	4741.4519	924.081	17921.439
average turnaround time	4846.3462	1027.6238	18094.6829
cpu utilization	0.2551	0.5271	0.1679
memory utilization	0.9349	0.6103	0.9832

	Trabalho OSP 1		
	avg	low	high
system throughput	2.0249	5.4250	0.7247
average waiting time	4755.7573	905.6984	12354.7358
average turnaround time	4874.7087	994.3214	12495.7547
cpu utilization	0.2717	0.5429	0.1667
memory utilization	0.9642	0.6597	0.9606

Como analisado no trabalho 01, uma performance melhor está relacionada a um baixo rendimento de sistema(system throughput), baixa média de tempo de espera(average waiting time) e tempo médio de resposta(average turnaround time), além de alta utilização de memória e cpu. Logo, podemos observar que a implementação não deixou a desejar quando comparada com a execução padrão, OSP.demo.

Como previsto, os resultados mostram que os valores da nova implementação se aproximaram mais da padrão, com variantes pequenas, diferentemente da primeira implementação, em que não havia escalonamento por prioridade entre processos. Isso ocorre pois o tempo de ociosidade da CPU é diminuído, uma vez que processos que ficam mais tempo sem executar podem utilizar da CPU por mais tempo; além de priorizar aqueles que bloqueiam com menos frequência e, conseqüentemente, finalizá-los mais rapidamente.

Idealmente, para uma melhoria do algoritmo, seria recomendável uma otimização para diminuir o tempo em que a CPU espera um novo processo a ser escalonado, pois o tempo para calcular uma nova prioridade e adicionar em uma fila específica consome tempo(tempo este que o primeiro algoritmo não consumia), para aproximar ainda mais dos resultados verificados na execução do OSP.demo.