

Trabalho 4 – Carlos Alberto Rosa dos Santos

Tempo de leitura E/S = 10

Tempo de escrita E/S = 10

RR = algoritmo de escalonamento – round-robin

Q = quantum

PR = algoritmo de escalonamento – prioridade preemptivo

PE = período de escalonamento(usado no algoritmo de prioridade)

Conjunto de 50 processos divididos entre limitados por E/S e limitados por CPU

	RR com Q = 1	RR com Q = 100	PR com PE = 1	PR com PE = 100
Tempo de execução	78	78	78	78
Tempo pronto	1854	1839	2844	2861
Tempo bloqueio	46	46	46	46
Tempo de retorno	1978	1963	2968	2986
Tempo de resposta	0.91	21.75	0.89	23.14

Conclusão

Com quantum pequeno, round robin e prioridade tem tempo médio de respostas similares, mas o tempo de retorno é significativamente menor no round robin. Desta forma, conclui-se que os processos terminam antes no round robin mas mantém o mesmo tempo de resposta, o que torna este algoritmo mais vantajoso neste caso.

Com quantum grande, o mesmo cenário se repete, mas o tempo de resposta de ambos aumentou bastante, devido aos processos escalonarem muito menos.

Caso tivéssemos adotado um algoritmo de escalonamento prioridade não-preemptivo, o tempo de resposta seria ainda maior para ambos, pois escalonaria ainda menos.

Conjunto de 50 processos limitados por E/S

	RR com Q = 1	RR com Q = 100	PR com PE = 1	PR com PE = 100
Tempo de execução	65	65	65	65
Tempo pronto	1588	1564	3060	2971
Tempo bloqueio	100	100	100	100
Tempo de retorno	1753	1729	3225	3136
Tempo de resposta	0.80	3.27	0.81	4.44

Conclusão

Com quantum pequeno, round robin e prioridade tem tempo médio de respostas similares, mas o tempo de retorno é significativamente menor no round robin. Desta forma, conclui-se que os processos terminam antes no round robin mas mantém o mesmo tempo de resposta, o que torna este algoritmo mais vantajoso neste caso.

Com quantum grande, o mesmo cenário se repete, mas o tempo de resposta de ambos aumentou muito menos que no conjunto de processos divididos entre limitados por CPU e limitados por E/S, pois neste caso como os processos fazem muita E/S, o escalonador entra em ação muito mais vezes, embora o quantum seja pequeno.

Trabalho 5 – Carlos Alberto Rosa dos Santos

Tempo de leitura E/S = 10
Tempo de escrita E/S = 10

Tamanho das páginas/quadros = 2
Tamanho da memória de cada processo = 32
Tamanho do programa limitado por E/S = 16
Tamanho do programa limitado por CPU = 18

Conjunto de 50 processos divididos em 20 processos limitados por E/S e 30 processos limitados por CPU

Memória total dos processos = 860

Para memória principal com tamanho 16(aproximadamente 2% da memória total dos processos)

	FIFO	LRU
Número de falhas de páginas	3634	3112

Para memória principal com tamanho 512(aproximadamente 60% da memória total dos processos)

	FIFO	LRU
Número de falha de páginas	420	420

Para memória principal com tamanho de 1024(maior que a memória total dos processos)

	FIFO	LRU
Número de falha de páginas	420	420

Conclusão

Percebe-se que quando a memória principal é muito pequena em relação ao total de memória dos processos o algoritmo LRU teve menos falhas de páginas. O número de falhas de páginas foi o mesmo no caso da memória ter 60% ou mais da memória total dos processos.