

# MiniEp 04 – Overhead e Starvation em algoritmos de acesso à SC

André Luiz Abdalla Silveira – 8030353

3 de maio de 2018

## Introdução

Nesse EP, somos convocados à contemplar e refletir acerca dos algoritmos apresentados nos arquivos `bakery.c` e `gate.c`, bem como fazer alterações para verificar e apontar determinados efeitos ou fenômenos. A primeira parte desse trabalho consiste em apresentar os dados obtidos, e a segunda.

É possível notar que a presença do `_sync_synchronize (ss)` permite que os testes em `test.c` sejam bem sucedidos ao invés do que acontece quando essa instrução não está presente. Entretanto, esse é o único efeito observável.

Além de testar os resultados com e sem `_sync_synchronize`, outras intervenções efetuadas envolvem a escolha dos escalonadores, e quaisquer alterações que se fizeram necessárias para a devida execução desses algoritmos, como comentar um `return 1` na linha 109 do arquivo de testes. Os efeitos dessas alterações encontram-se comentados abaixo

## Legenda

- $T_{alg}$  é a média obtida nos "Elapsed Time" para as 30 execuções do algoritmo  $alg$ <sup>1</sup>
- $\mu_{alg}$  média obtida considerando-se as quantidades médias de acesso à SC para  $alg$
- $\sigma_{alg}$  média obtida considerando-se os desvios padrão relativos às quantidades médias de acesso à SC para  $alg$
- `SCHED_OTHER` é o escalonador padrão do sistema operacional
- `SCHED_FIFO` é o escalonador `First In, First Out`
- `SCHED_RR` é o escalonador `Round Robbin`

---

<sup>1</sup>  $alg \in \{bakery, gate\}$

# 1 Tabelas

N° Threads	Tempo total	$T_{bakery}$	$\mu_{bakery}$	$\sigma_{bakery}$	$T_{gate}$	$\mu_{gate}$	$\sigma_{gate}$
20	3000000	230.69	1500.00	143.47	3.39	1872.87	3252.87
30	3000000	388.21	1000.00	34.00	3.67	1251.83	2641.13
60	3000000	757.86	500.00	8.97	3.79	593.20	1954.27
60	6000000	1201.03	1000.00	10.20	27.97	1153.87	3318.77
60	9000000	1788.04	1500.00	11.10	47.04	1692.90	4309.57

Tabela 1: Resultados obtidos sem `_sync_synchronize`

N° Threads	Tempo total	$T_{bakery}$	$\mu_{bakery}$	$\sigma_{bakery}$	$T_{gate}$	$\mu_{gate}$	$\sigma_{gate}$
20	3000000	238.69	1500.03	142.67	3.20	1802.07	3155.90
30	3000000	366.49	1000.00	46.70	3.53	1246.47	2790.07
60	3000000	706.51	500.00	12.87	4.89	579.93	1878.17
60	6000000	1386.40	1000.00	9.70	35.04	1253.13	3575.74
60	9000000	1939.28	1500.00	9.63	80.37	1687.27	4481.27

Tabela 2: Resultados obtidos com `_sync_synchronize`

Escalonador	ss?	$T_{bakery}$	$\mu_{bakery}$	$\sigma_{bakery}$	$T_{gate}$	$\mu_{gate}$	$\sigma_{gate}$
SCHED_OTHER	N	388.21	1000.00	34.00	3.67	1251.83	2641.13
SCHED_OTHER	S	366.49	1000.00	46.70	3.53	1246.47	2790.07
SCHED_FIFO	N	50.50	1000.00	3.53	1.11	1278.67	2351.63
SCHED_FIFO	S	47.17	1000.00	2.57	0.14	1000.00	1413.53
SCHED_RR	N	38.53	1000.00	4.08	0.96	1325.97	2365.50
SCHED_RR	S	39.72	1000.00	2.80	1.60	1000.00	1427.13

Tabela 3: Resultados obtidos com `_sync_synchronize`, 30 threads, 3 M de tempo total

# 2 Conclusões

Com os resultados das execuções, ficou clara que a instrução que funciona como barreira de memória não causa um efeito significativo na execução dos algoritmos, sendo notável somente o fato de que a sua presença garante que ambos os algoritmos possam viabilizar exclusão mútua com 100 threads, o que não se observa na ausência do (ss).

O aumento na quantidade de threads, bem como o do tempo total provocaram aumentos no tempo total. Isso é facilmente explicável, pelo fato de que aumenta-se a quantidade de threads para acessar a SC no primeiro caso, e depois as tentativas de acesso.

Outro fenômeno curioso é a uniformidade de resultados obtidos em  $\mu_{bakery}$  e a quase uniformidade no caso de  $\mu_{gate}$ . Ainda que pareça especulativo, é notável que a relação entre tempo total e quantidades de threads é idêntica entre experimentos que apresentam média das médias com

resultados rigorosa ou aproximadamente iguais. O motivo disso provavelmente deve-se ao fato de haver proporções idênticas de tempo e threads possibilitando resultados parecidos ou mesmo idênticos.

Há de observar-se também que a o desvio padrão diminuía no caso do bakery de acordo com o aumento das quantidades de tempo e threads, enquanto no algoritmo gate, observa-se o desvio padrão diminui juntamente à diminuição da razão tempo/threads.

Os resultados mais interessantes observam-se na tabela 3. Veja:

- $\mu_{bakery}$  mantém-se inutável, enquanto  $\mu_{gate}$  diminui com a presença de ss associada ao RR e FIFO
- Levando em conta os desvios padrão:
  - Para as execuções de bakery, o valor médio vai decrescendo de escalonador para escalonador.
  - Ainda levando o bakery em consideração, não parece ser possível traçar uma correlação entre os resultados com e sem ss
  - Para as execuções de gate, o único padrão observável é que a média dos resultados com ss é sempre maior que o sem.