



UNIVERSIDADE
FEDERAL DO CEARÁ

UNIVERSIDADE FEDERAL DO CEARÁ

Trabalho de Sistemas Operacionais

Layssa Oliveira - 564923

Emilly Pinheiro - 567253

Gabriel Ribeiro - 567908

Resolução das Questões

Questão 1 - O código da presente questão consta no github. Respectivo link:

<https://github.com/Layssaoliveira26/Entrega-Sistemas-Operacionais?tab=readme-ov-file>

g) Não houve muita diferença em relação a escrita do código, a maior diferença nesse quesito foi a forma de acordar as *threads* consumidoras, anteriormente havíamos utilizado apenas *signal*, mas com a adição da nova *thread* consumidora precisamos usar *signal* e *broadcast* para que ambas fossem avisadas no momento de checar a condição de parada. Em relação a saída, houve uma maior lentidão devido ao aumento das iterações de ambos os tipos de *thread*, e um outro ponto bem interessante foi a iteração das *threads* consumidoras, anteriormente, quando havia apenas uma *thread* ela retornava sempre 15 iterações no final (isso ocorria pois a quantidade de iterações que cada *thread* produtora deveria efetuar estava fixa em 25), quando a segunda foi adicionada esse número final passou a ser algo aleatório, cada *thread* devolvia um número diferente a cada execução(a soma era sempre 30), ajudando a visualizar melhor a questão da concorrência.

Questão 2 - O código da presente questão consta no github. Respectivo link:

<https://github.com/Layssaoliveira26/Entrega-Sistemas-Operacionais?tab=readme-ov-file>

f) A tabela abaixo apresenta as diferenças na execução do código com um diferente número de Threads:

Threads	2	4	8	16	32	64	128	256	512	1024	2048	4096
Total Processo (paralelo)	2.58	1.83	1.73	1.68	1.66	1.61	1.51	1.90	1.72	1.86	2.84	1.87
Total Threads	5.15	7.21	13.36	24.62	49.21	95.75	166.02	353.40	220.41	69.70	71.93	14.19

Com base nos testes realizados para a execução do código da questão dois, foi possível observar que, à medida que o número de threads aumentava, o tempo de execução tendia a diminuir, evidenciando os benefícios da paralelização. No entanto, constatamos que, a partir de um certo ponto, o aumento do número de threads deixou de gerar melhorias significativas e até passou a estabilizar ou aumentar o tempo de execução. Isso indica que há um limite prático para o paralelismo eficiente, e que um número excessivo de threads pode resultar em mais competição por recursos do que em ganhos de desempenho.

