UNIFESP

UNIVERSIDADE FEDERAL DE SÃO PAULO

1933

Relatório da Atividade 6

Programação concorrente e distribuída

Rosangela Miyeko Shigenari 92334

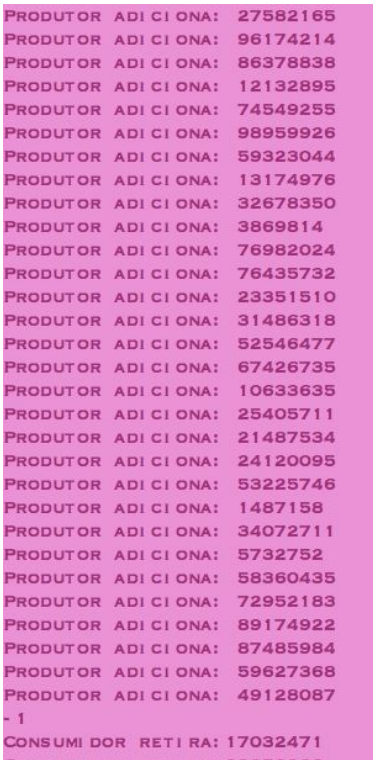
1) Conside um programa concorrente, utilizando C+PThreas, ou C+OpenMP ou ainda JavaThreads, onde existem duas categorias de processos/threads. Os processos da primeira categoria, denominados TPs, deverão produzir N números inteiros aleatórios (todos na faixa entre 1 e 10^8) e inseri-los em uma determinada fila. Os demais processos de outro tipo, denominados TCs, deverão retirar (se houver) dados da fila em comum e verificar se os mesmos são números primos. Caso os TCs recebam números negativos os mesmos deverão ser finalizados. Simule o funcionamento do sistema nas seguintes configurações:
   a) TP=1, TC=2, N=100
   b) TP=2, TC=4, N=100
   c) TP=4, TC=2, N=100
   Onde N representa a quantidade total de números gerados pelos TPs.
   Obs: Considere que após a quantidade total de números aleatórios gerados for igual a N os TPs deverão emitir valores negativos para a quantidade exata de TCs, de forma a encerrá-los e também encerrar-se posteriormente.
   Demonstre em relatório o funcionamento correto do código desenvolvido para todos os casos requeridos.

Resposta:
O código está em anexo (produtorConsumidor.java).
Para o mesmo código foram testados os seguintes casos
a) TP=1, TC=2, N=100
Neste caso, foram criadas 1 *thread* como produtor e 2 para o consumidor, podendo notar que houve um tempo de execução satisfatório uma vez que ter mais consumidor que o produtor é possível que todos os produtores sejam atendidos paralelamente.

b) TP=2, TC=4, N=100
Neste caso, foram criadas 2 *thread* como produtor e 4 para o consumidor, podendo notar que houve um tempo de execução foi menor que o anterior que já que se aumentou o número de *threads* produtoras e o número de consumidores é o suficiente para verificar as inserções realizadas pelas TPs.

c) TP=4, TC=2, N=100
Neste caso, foram criadas 4 *thread* como produtor e 2 para o consumidor, podendo notar que houve um tempo de execução desfavorável já que houve um acréscimo no número de

TPs mas o número de consumidores não seria o suficiente para verificar todas as inserções na fila em um tempo aceitável.
A imagem abaixo representa a compilação do programa, onde o produtor adicionou N elementos aleatórios (definido N = 100), quando as inserções na fila acabem é impresso -1. Após adicionar na fila, o consumidor irá retirar elementos da fila, como observado na imagem, onde um número primo é retirado.



2) Crie um programa em C com PThreads que implementem rotinas que fazem operações de Wait e Signal em um Semáforo Geral. Lembrando que a linguagem C apenas implementa, nativamente, semáforos binários (mutex). Assim, use mutex para implementar rotinas que fazem operações sobre semáforos gerais. Crie um exemplo de uso para Semaforos Gerais que ateste seu funcionamento através do uso das rotinas criadas.

Resposta: O exemplo trata de 2 threads, uma que realiza a leitura de uma string e a outra que realiza a escrita, e a leitura só pode ser executada após a execução da thread escritora (em anexo como sem.c)