

UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE INSTITUTO METRÓPOLE DIGITAL

Professor: Julio Melo

Trabalho Prático 1 – Processos, Memória compartilhada e Sincronização

A utilização de memória compartilhada não é uma prática mandatória para comunicação entre processos. Como vimos em aula, podemos realizar comunicação entre processos usando arquivos ou sinais, que, embora menos flexíveis podem ser usados para resolver problemas de comunicação mais simples.

No entanto, nem a utilização de sinais nem a de arquivos resolve o problema de sincronização entre os processos. Como vimos, a sincronização entre processos é feita através de semáforos ou *mutexes* (que são um caso exclusivo dos semáforos).

Os semáforos indicam, para os outros processos, quando um processo entra em uma região crítica, onde altera o valor de uma memória compartilhada, por exemplo, ou acessa um dispositivo específico que não suporte acesso simultâneo.

A utilização de semáforos é, praticamente, necessária quando dois processos escrevem na mesma memória. No entanto, nem sempre precisamos usar semáforos quando um dos processos apenas lê a memória em questão.

Exemplo prático de programação paralela

Uma aplicação direta da utilização de memórias compartilhadas, processos e sincronização é a paralelização de operações matemáticas. Uma instância clássica desse exemplo é a multiplicação de matrizes:

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix} * \begin{pmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ b_{31} & b_{32} & b_{33} \end{pmatrix} = \begin{pmatrix} c_{11} & c_{12} & c_{13} \\ c_{21} & c_{22} & c_{23} \\ c_{31} & c_{32} & c_{33} \end{pmatrix}$$

$$c_{11} = a_{11} * b_{11} + a_{12} * b_{21} + a_{13} * b_{31}$$

O caso acima mostra o resultado da multiplicação de matrizes para uma matriz 3x3 e é um exemplo prático que mostra a clara independência dos dados na multiplicação de matrizes. No caso mostrado cada elemento da matriz resultado, C_{nm} , pode ser calculado independentemente dos demais, apenas lendo os valores das matrizes a serem multiplicadas.

Implemente, dessa forma, a multiplicação paralela entre duas matrizes:

- O programa principal deve receber como entrada duas matrizes A e B a serem multiplicadas, de tamanho variado (possivelmente maiores do que 3x3), porém compatíveis para a multiplicação, via linha de comando ou através de um arquivo.
- O programa deve criar um processo para o cálculo de cada elemento da matriz resultado.
- O programa principal deve imprimir a matriz resultado no console ou gerar um arquivo com o resultado, após todos os cálculos serem realizados.

Comparação entre implementação paralela e sequencial

A multiplicação de matrizes pode parecer um programa simples, mas o ganho através da computação paralela e utilização de processos é extremamente alto para esse tipo de problema:

- a) Implemente um programa para calcular a multiplicação de matrizes que receba entradas no mesmo formato do programa paralelo.
- b) Calcule o tempo que o programa leva para multiplicar as matrizes.
- c) Adicione, ao programa paralelo, uma lógica no programa principal que permita calcular o tempo demorado para multiplicar as matrizes.
- d) Mostre, através de um gráfico, o ganho/perda entre a multiplicação sequencial e multiplicação paralela de matrizes para matrizes com tamanhos 1x1, 2x2, 3x3, 4x4, 5x5, 6x6, 8x8, 10x10, 20x20, 30x30, 40x40, 50x50, 75x75 e 100x100. (Dica: Utilize um gerador aleatório para os elementos das matrizes).

Sincronizando atividades paralelas

Utilize um processo para monitorar todos os demais processos de forma que permita parar a execução de todos e monitorar quais operações já foram executadas:

- a) O processo monitor deve ser capaz de parar a execução de todos os processos ainda ativos e listar quais elementos da matriz resultado já foram calculados e quais ainda faltam.
- b) O processo monitor deve ainda ser capaz de controlar qual elemento da matriz resultado será calculado naquele instante; ex: caso o usuário deseje calcular o elemento R_{nm} , o monitor deve desbloquear os processos responsáveis por calcular esses elementos e exibir o resultado do elemento calculado assim que esse terminar.