

INF01151 – SISTEMAS OPERACIONAIS II N
SEMESTRE 2015/1
TRABALHO 1: PROCESSOS E THREADS

1. ESPECIFICAÇÃO DO TRABALHO:

O trabalho consiste em implementar em C (não em C++) um programa concorrente para multiplicação de matrizes. O programa deve ser implementado em duas versões diferentes, uma delas utilizando processos UNIX e outra utilizando pthreads. Os programas deverão executar obrigatoriamente em ambientes Unix (Linux) mesmo que tenham sido desenvolvidos sobre outras plataformas.

2. MÚLTIPLICAÇÃO DE MATRIZES

O programa deve receber como entrada duas matrizes M_{mx} e M_{xn} , de números inteiros, e calcular a matriz resultante M_{mn} . As matrizes de entrada, M_{mx} e M_{xn} , devem ser lidas dos arquivos **in1.txt** e **in2.txt**, respectivamente. O formato de cada um dos arquivos deve seguir a seguinte especificação:

```
LINHAS = #LINHAS  
COLUNAS = #COLUNAS  
M11 M12 M13 ... M1x  
M21 M22 M23 ... M2x  
M31 M32 M33 ... M3x  
...  
MM1 MM2 MM3 ... MMX
```

Por exemplo:

```
LINHAS = 3  
COLUNAS = 3  
17 8 14  
4 23 9  
24 7 10
```

DICA: durante a avaliação, o programa será executado com uma série de casos de teste, portanto siga rigorosamente o formato especificado acima. O caracter separador entre cada dois números de uma linha deve ser **um único espaço em branco**.

A matriz resultante deve ser gerada em um arquivo chamado **out.txt**, com formato idêntico ao dos arquivos de entrada.

3. VERSÕES: PROCESSOS E THREADS

Você deverá implementar duas versões para a solução do problema de multiplicação de matrizes: uma versão ilustrando a utilização processos UNIX (criados através da primitiva **fork**), e outra versão utilizando pthreads. Em ambos os casos, o número de processos ou pthreads a serem criados deve ser especificado como um parâmetro n de entrada passado pela linha de comando.

Conforme o parâmetro n , você deverá utilizar n processos ou threads para realizar a **multiplicação de linhas de forma concorrente**. Por exemplo, tendo sido passado n com valor igual a 4 pela linha de

comando, você deverá criar 4 processos ou threads (conforme a versão do programa) para calcular concorrentemente a matriz resultante. Cada processo ou thread ficará responsável por um subconjunto de linhas, que devem ser distribuídos **proporcionalmente** entre as unidades de execução.

Nos testes utilize $n = 2, 4$ e 8 , com diferentes tamanhos de matrizes de entrada. Com o auxílio do comando *time*, executar cada versão do programa 10 (dez) vezes e anotar o tempo total de execução. Você deverá especificar e criar casos de testes com matrizes de diversos tamanhos. No final calcular a média aritmética simples dos tempos de execução para cada caso.

4. DESCRIÇÃO DO RELATÓRIO A SER ENTREGUE

Deverá ser produzido e entregue um relatório fornecendo os seguintes dados:

- Descrição do ambiente de teste: versão do sistema operacional e distribuição, configuração da máquina (processador(es) e memória) e compiladores utilizados (versões).
- Descrição dos casos de testes elaborados:
 - Parte da avaliação será baseada na qualidade dos casos de testes elaborados. Esses testes devem incluir o uso de matrizes pequenas, médias e grandes, onde se possa (ou não) identificar ganhos de desempenho com o uso do paralelismo.
- Respostas aos seguintes questionamentos:
 - Para cada caso de teste, execute a multiplicação de matrizes de forma sequencial ($n = 1$), 10 (dez) vezes, e compare o tempo de execução com as versões que fazem o cálculo de forma concorrente. Há ganho de desempenho? JUSTIFIQUE, e apresente os dados coletados e uma **análise dos resultados**.
 - Há diferenças de tempo entre a versão com processo e a versão com threads para $n = 2, 4$ e 8 ? Se sim, em que casos? JUSTIFIQUE, e apresente os dados coletados e uma **análise dos resultados**.
 - Explique como a comunicação entre as unidades de execução (na implementação utilizando processos) foi implementada.

Entregar dois programas C diferentes (implementação em processos UNIX e implementação em pthreads).

5. DATAS E MÉTODO DE AVALIAÇÃO

O trabalho pode ser feito em grupos de **2 OU 3 INTEGRANTES**. Não esquecer de identificar claramente os componentes do grupo no relatório.

Faz parte do "pacote" de entrega os fontes e o relatório em um arquivo ZIP. O trabalho deverá ser entregue até 23:55 horas do dia **05/05/2015 (turma A)**, e até 23:55 do dia **04/05/2015 (turma B)**. A entrega deverá ser via moodle (link para submissão na Aula 07).

Após a data de entrega o trabalho deverá ser entregue via e-mail para alberto@inf.ufrgs.br (subject do e-mail deve ser "INF01151: Trabalho 1"). Neste caso, será descontado 02 (dois) pontos por semana de atraso. O atraso máximo permitido é de duas semanas, isto é, nenhum trabalho será aceito após 19/05/2015 (turma A) e 18/05/2015 (turma B).