

SISTEMAS OPERACIONAIS
PROF. CALEBE CONCEIÇÃO

Aula 6 - Exercícios sobre Processos e Threads

Atividade avaliativa, conforme plano de ensino apresentado. Atente-se aos prazos descritos no AVA.

Aluna(o): _____

Turma: _____

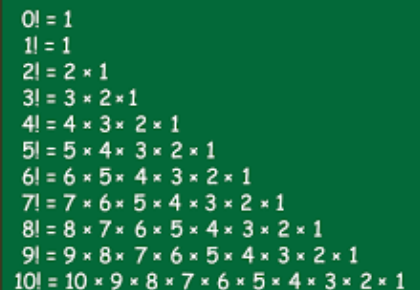
Nesta atividade você deve implementar em ambiente Linux diferentes soluções para o cálculo fatorial de um número. Você deve implementar e executar 4 soluções. Uma solução direta com laços de repetição (fornecida), outra usando processos que se comunicam por meio de memória compartilhada, uma terceira que dispara o uso de 2 usando threads, e uma quarta que dispara 10 threads. A seguir explico mais detalhes sobre como as unidades de processamento devem ser estruturadas.

Ao final, você deve comparar o tempo de execução de cada solução e reportar. Na entrega final devem constar a listagem do código fonte e suas impressões a respeito do tempo de execução de cada solução, respondendo às seguintes questões: os tempos de processamento convergem com o previsto em teoria? Se não convergem, o que pode ter acontecido? Quais fatores influenciam no tempo de processamento de cada solução?

INSTRUÇÕES GERAIS

O cálculo do fatorial de um número N consiste na multiplicação e acumulação dos números N por todos os seus antecessores até 1, conforme ilustrado ao lado. Nesta atividade, sua tarefa é colocar duas unidades de execução (processos ou threads) a cooperar para solução do problema para alguns valores de N pré-definidos nesta atividade. Os programas devem ser implementados em C/C++, usando a biblioteca PThreads quando necessário. Para todas as variações de implementação, use o comando **hyperfine** para medir o tempo de execução, conforme segue:

`hyperfine -warmup 3 -m 10 ./seu_codigo_binario_aqui`



$0! = 1$
 $1! = 1$
 $2! = 2 \times 1$
 $3! = 3 \times 2 \times 1$
 $4! = 4 \times 3 \times 2 \times 1$
 $5! = 5 \times 4 \times 3 \times 2 \times 1$
 $6! = 6 \times 5 \times 4 \times 3 \times 2 \times 1$
 $7! = 7 \times 6 \times 5 \times 4 \times 3 \times 2 \times 1$
 $8! = 8 \times 7 \times 6 \times 5 \times 4 \times 3 \times 2 \times 1$
 $9! = 9 \times 8 \times 7 \times 6 \times 5 \times 4 \times 3 \times 2 \times 1$
 $10! = 10 \times 9 \times 8 \times 7 \times 6 \times 5 \times 4 \times 3 \times 2 \times 1$

Nota 1) o `./` na frente do seu binário é importante para o comando funcionar.

Nota 2) mande mensagem no grupo da turma se precisar de ajuda com os comandos necessários para compilação

Este comando realiza a execução de uma tarefa 10 vezes, coleta e apresenta o resultado do processamento, contendo mediana, desvio padrão, e os valores máximo e mínimo. Anote esses valores na tabela do verso da folha.

A entrega dessa atividade consiste no preenchimento da Tabela 1 - Resultados Obtidos no verso desta folha com as saídas do comando **hyperfine** para cada cenário, juntamente com suas considerações sobre os experimentos realizados e o envio desta folha de atividades preenchida juntamente com os códigos-fonte das aplicações indicadas.

*Obs.: Ignore o resultado da saída de execuções individuais para N muito grandes. Estamos mais interessados em fazer o computador trabalhar, e por isso considere só a execução do **hyperfine**. ;-)*

#Bons estudos.

Tabela 1 - Resultados obtidos

| Valor de N | Nome do Programa | Mediana | desvio padrão (σ) | (user) | (system) | max | min |
|---------------|--------------------------------------|---------|----------------------------|--------|----------|-----|-----|
| 1.000 | fatorial_simples | | | | | | |
| | fatorial_dois_processos_memoria_comp | | | | | | |
| | fatorial_2_threads | | | | | | |
| | fatorial_10_threads | | | | | | |
| 1.000.000 | fatorial_simples | | | | | | |
| | fatorial_dois_processos_memoria_comp | | | | | | |
| | fatorial_2_threads | | | | | | |
| | fatorial_10_threads | | | | | | |
| 1.000.000.000 | fatorial_simples | | | | | | |
| | fatorial_dois_processos_memoria_comp | | | | | | |
| | fatorial_2_threads | | | | | | |
| | fatorial_10_threads | | | | | | |

Suas considerações

Apresente no quadro abaixo suas considerações sobre os experimentos realizados. Atente-se às questões apresentadas no enunciado da atividade.

DESCRIÇÃO DAS APLICAÇÕES

Aplicação 1 - Fatorial simples

Para ajudar, o código fonte desta aplicação em C está listado a seguir. Você pode alterar o valor de N no corpo do programa para realizar os experimentos.

| Código-fonte: <i>fatorial.c</i> |
|--|
| <pre>#include <stdio.h> // funcao recursiva que calcula o fatorial de n unsigned long long int fatorial(int inicio, int n) { unsigned long long int resultado = 1; for (int i = inicio; i <= n; i++) { resultado *= i; } return resultado; } int main() { int numero = 1000; // defina AQUI o valor de N printf("%llu\n", fatorial(1, numero)); return 0; }</pre> |

Aplicação 2 - Fatorial com 2 processos, comunicação com memória compartilhada.

Nessa implementação, são executados 3 processos distintos. O processo main, que controla a criação de outros dois processos filhos, juntando seus resultados ao final; e dois processos filhos, responsáveis por executar metade do cálculo fatorial de N. O código dessa atividade está disponível nos arquivos *fatorial_memoria_compartilhada_main.c* e *fatorial_auxiliar.c*. Os códigos encontram-se sem qualquer comentário, sua tarefa é ler o código e entender. A partir desse entendimento, inferir como rodar, pois vai demandar uma ordem de compilação e nomenclatura dos binários. Ao final, você deverá adicionar comentários explicativos para os comandos utilizados, e reportar os resultados na tabela.

Aplicações 3 e 4 - Fatorial com 2 threads e com 10 threads

Da mesma forma que o problema foi dividido pela metade na aplicação anterior, você deve dividir a tarefa em duas threads e em 10 threads nessa aplicação, usando Pthreads. Na aplicação 3 cada thread fica responsável por 50% do trabalho, na aplicação 4 cada thread fica responsável por 10% do trabalho. Em ambas, mantém-se a thread principal, controlando as demais e juntando os resultados. Tome por base o código de exemplo de uso de Pthreads apresentado no Capítulo 4 do livro Fundamentos de Sistemas Operacionais: princípios básicos para implementar essa solução. Faça então as devidas medições e anotações.

Qualquer dúvida, perguntem! Estou à disposição no telegram: @calebemicael

Bons estudos!