MAC0219 - Relatório Mini-EP 4

Nathalia Yukimi Uchiyama Tsuno

September 2024

1 Uma Breve<-íssima> Introdução Sobre Threads

Threads são sequências de instruções que permitem múltiplas execuções de processos/tarefas simultanea e concorrentemente.

Isso permite uma computação mais simplicada, rápida e eficiente e, por isso, é usada por quem procura bom desempenho.

Contudo, é necessário ter cautela ao se usar dessa ferramenta. Iniciando-se pelo fato de garantir que os processos sejam independentes para evitar sobrescrição e cômputos errôneos.

2 Parte I - Uma Soma Triangular Muito Estranha

Joãozinho é um estudante de segundo ano do Bacharelado em CIência da Computação da USP (Universidade Specialista em Paralela) e cursa a disciplina de MAC0219. Numa dessas aulas, ele precisava criar um trabalho que demonstrasse eficiência.

Assim, após pensar muito, ele criou o programa "MiniEP4.c", em linguagem C, que aplicava o uso de Threads para esse fim.

De modo simples, o programa inicializa 8 Threads e cada uma executa a função "Executa". Nesta, sabendo-se que cada Thread está identificada por um id entre 0 e 7, ela computa a soma triangular em t. Isto é, por exemplo, se a id da Thread for 4, ela deve imprimir 10.

Além disso, como ele adora variáveis globais, Joãozinho usa uma chamada "soma", que é compartilhada e acessada por todas as Threads. Esta variável faz o cômputo soma <- $\sum_{i=0}^{t} i$.

Contudo, após uma análise, Joãozinho percebeu que muitas das Threads não retornavam o valor esperado e resolveu investigar a causa.

3 Parte II - No Meio Do Caminho Tinha Uma Variável Global, Tinha Uma Variável Global No Meio Do Caminho

Uma problemática das Threads é o acesso e escrita a recursos compartilhados. A ausência de um cuidado pode ocasionar o que foi experenciado por Joãozinho.

Note que, enquanto uma Thread faz a leitura da variável compartilhada e global "soma", uma outra pode estar a sobreescrevendo e uma terceira pode estar a recebendo modificada do resultado desejado e a imprimindo errôneamente.

Assim, enquanto uma Thread de id = 4 pode estar fazendo um cômputo para soma triangular em 4, ela pode estar influenciando o cálculo da Thread de id = 6, que já não começa com soma = 0, e a de id = 1 pode estar a imprimindo.

4 Parte III - Até Que O MUTEX Nos Separe

Uma solução para essa problemática é o uso do MUTEX e do Join. Isso está especificado em no arquivo "MiniEP4Mutex.c"

Ao aplicar o pthread_mutex_lock() e pthread_mutex_unlock() na função "Executa", bloqueamos o acesso à variável global e compartilhada "soma"por Threads terceiras, garantindo que não haja sobreescrita no cálculo da soma.

E, ao aplicar pthread_join(), garantimos o bloqueio de execução da Thread de id t, também garantindo ordem e que a variável soma comece com 0.

Uma outra solução mais simples seria o uso de variáveis locais "MiniEP4Local.c".