

Relatorio de Execucao: LAB5 - INF1316 - Sistemas Operacionais

Exercicio com Threads

Exercicio 1: Paralelismo

Objetivo:

Criar um vetor de 10.000 posicoes inicializado com o valor 5 e dividir a carga de trabalho entre 10 threads (trabalhadores). Cada thread e responsavel por multiplicar sua parte do vetor por 2 e retornar a soma das posicoes ao processo coordenador, que exibira a soma final das parcelas. No segundo caso, o exercicio e repetido com 100 threads e um vetor de 100.000 posicoes.

Cenario 1:

Configuracao:

- Tamanho do vetor: 10.000 posicoes
- Numero de trabalhadores (threads): 10
- Funcao executada: Cada thread multiplica os valores de sua parte do vetor por 2 e retorna a soma dessa secao ao processo coordenador.

Divisao de trabalho:

Cada thread manipula 1.000 posicoes consecutivas no vetor.

Resultado da soma total:

- Soma total esperada: Cada posicao do vetor inicializado com 5 e multiplicada por 2, resultando em 10. Portanto, a soma total esperada e $10.000 * 10 = 100.000$.

Resultado obtido:

Soma total calculada pelas threads: 100.000.

Cenario 2:

Configuracao:

- Tamanho do vetor: 100.000 posicoes
- Numero de trabalhadores (threads): 100
- Funcao executada: Cada thread multiplica os valores de sua parte do vetor por 2 e retorna a soma dessa secao ao processo coordenador.

Divisao de trabalho:

Cada thread manipula 1.000 posicoes consecutivas no vetor.

Resultado da soma total:

- Soma total esperada: Cada posicao do vetor inicializado com 5 e multiplicada por 2, resultando em 10. Portanto, a soma total esperada e $100.000 * 10 = 1.000.000$.

Resultado obtido:

Soma total calculada pelas threads: 1.000.000.

Analise de Desempenho:

1. Cenario 1:

O uso de 10 threads para manipular o vetor de 10.000 posicoes distribuiu a carga de forma eficiente. Como cada thread manipula exatamente 1.000 posicoes, o processamento foi dividido de forma equitativa, resultando em uma execucao rapida e com baixo overhead de gerenciamento de threads.

2. Cenario 2:

O uso de 100 threads com o vetor de 100.000 posicoes tambem resultou em uma divisao equitativa de 1.000 posicoes por thread. No entanto, o overhead de criacao e sincronizacao de um numero maior de threads e maior do que no primeiro caso. Dependendo das especificacoes do hardware,

pode haver um impacto no tempo de execucao devido a sobrecarga de threads e a necessidade de sincronizacao entre elas.

Conclusao:

Ambos os cenarios resultaram nos valores esperados para a soma final do vetor. A implementacao com 10 threads foi eficiente, com menor overhead e rapida execucao. No cenario com 100 threads, o aumento do numero de trabalhadores introduziu um maior overhead de gerenciamento, mas a tarefa foi igualmente dividida, e o resultado final foi alcancado conforme esperado.