
Gerenciamento de Tarefas com ThreadPoolExecutor - Criando um Executor de Threads de Tamanho Fixo



STATUS

DESENVOLVIDA

Objetivo do Projeto



Este projeto demonstra o uso de um ThreadPoolExecutor com um pool de tamanho fixo para gerenciar a execução concorrente de tarefas. Ele simula um servidor que recebe e processa múltiplas tarefas simultaneamente usando um número limitado de threads.

Estrutura do Código

Classe Server

Gerencia o pool de threads e a execução das tarefas. Inicializa um ThreadPoolExecutor com 5 threads. Submete tarefas ao executor. Exibe informações sobre o status do pool. Encerra o executor após a conclusão das tarefas.

Classe Task

Representa uma tarefa executável. Registra a data de criação e execução. Simula um tempo de processamento aleatório. Exibe informações sobre a execução da tarefa.

Classe Main

Cria um servidor com um pool de threads. Submete 100 tarefas ao servidor. Finaliza o servidor após a submissão.



Cenários teste



Cenário 1: Execução Normal

Descrição: O Server executa 100 tarefas com um pool de 5 threads.

Saída Esperada:

As tarefas são distribuídas entre as 5 threads do pool. O status do pool é atualizado dinamicamente. Todas as tarefas são concluídas corretamente.

=====

Cenário 2: Pool de Threads Pequeno

Descrição: O Server é configurado com um pool de apenas 2 threads.

Saída Esperada:

As tarefas são executadas em lotes de 2 por vez. O tempo total de execução aumenta devido ao menor número de threads.

=====

Cenário 3: Tarefas de Longa Duração

Descrição: Algumas tarefas têm uma duração maior (por exemplo, 15 segundos).

Saída Esperada:

As tarefas de longa duração não bloqueiam a execução das demais, pois o pool permite concorrência.

Conclusão:



O ThreadPoolExecutor melhora a eficiência na execução concorrente. O código garante que as tarefas sejam distribuídas corretamente entre as threads. A saída do programa demonstra claramente o funcionamento do pool.

Desenvolvedor



Marcio Fonseca



=====