
Sincronização de Threads com Phaser - Execução de Tarefas Concorrentes em Fases



STATUS

DESENVOLVIDA

Objetivo do Projeto



Este projeto demonstra o uso da classe Phaser para coordenar a execução de threads em fases. O programa busca arquivos com uma extensão específica dentro de diretórios predefinidos, filtrando apenas aqueles modificados nas últimas 24 horas. Cada thread executa sua busca de forma sincronizada, garantindo que todas avancem juntas para a próxima fase.

Estrutura do Projeto

Classe FileSearch: Implementa Runnable e executa a busca nos diretórios especificados. Classe Main: Responsável por criar e iniciar as threads, bem como aguardar sua finalização.

Phaser: Utilizado para coordenar as fases de execução das threads.

Fluxo de Execução

Fase 1: Início da execução das threads. Fase 2: Busca de arquivos com a extensão desejada. Fase 3: Filtragem de arquivos modificados nas últimas 24 horas. Fase 4: Exibição dos resultados. Fase 5: Finalização das threads e encerramento do Phaser.



Cenários teste



Cenário 1: Execução Normal

Todas as threads encontram arquivos com a extensão .log. Os arquivos recentes são filtrados corretamente. O programa finaliza normalmente.

=====

Cenário 2: Sem Resultados

Algumas threads não encontram arquivos com a extensão desejada. As threads sem resultados são desregistradas do Phaser. As demais continuam a execução normalmente.

=====

Cenário 3: Filtro de 24 Horas

Apenas arquivos modificados nas últimas 24 horas são considerados. O filtro funciona corretamente, excluindo arquivos antigos.

Conclusão:



Este projeto demonstra de forma clara a utilidade do Phaser para coordenar tarefas concorrentes em fases distintas. O uso dessa abordagem permite uma execução organizada e sincronizada, garantindo maior controle sobre a execução das threads.

Desenvolvedor



Marcio Fonseca



=====