



Sistemas Operacionais

Busca Paralela em Arquivos com Multithreading

Abrahão Picanço e Lucas Gabriel

Introdução

- O crescimento de dados textuais exige ferramentas rápidas para busca de informações.
 - Buscar palavras-chave manualmente em muitos arquivos é demorado.
 - Solução: Paralelizar a tarefa utilizando threads para acelerar a leitura e análise de arquivos de texto.
-

Objetivo

Desenvolver uma aplicação em linguagem C com suporte a multithreading (pthreads).

- Realizar busca por múltiplas palavras-chave em arquivos .txt localizados no diretório.
- Utilizar **multithreading** para processar múltiplos arquivos simultaneamente.
- Exibir o **nome do arquivo, linha contendo a palavra e o tempo total de execução.**

Conceitos Fundamentais

Multithreading com pthread

- Técnica que permite a execução concorrente de múltiplas tarefas (threads).
- Ideal para tarefas de leitura/escrita em arquivos (I/O bound).
- Reduz o tempo total de execução.

Exclusão Mútua (pthread_mutex_t)

- Garante que as mensagens impressas no terminal não se misturem.
 - Protege variáveis compartilhadas entre threads (Ex. Índice).
 - Uso da API FindFirstFile/FindNextFile para listar arquivos .txt.
-

Estrutura do Código

Etapas principais:

- Listagem dos arquivos .txt no diretório informado.
- Criação de threads com base no número de núcleos do sistema.

Cada thread:

- Abre um arquivo.
- Percorre linha por linha
- Procura palavras-chave com strstr().
- Imprime os resultados formatados.

Medição de tempo de execução com clock_gettime().

Exemplo de Execução

Usando 12 threads

arquivo1.txt [Line 10] (856): Código de teste 856 encontrado.

arquivo3.txt [Line 7] (987): Número 987 válido.

...

Tempo total de execução: 8.12 ms

Resultados Esperados

- Exibição clara e organizada dos arquivos e linhas onde as palavras foram encontradas.
- Aumento de desempenho com uso de threads em comparação à execução sequencial.
- Robustez na leitura de arquivos e no tratamento de erros.
- Memória alocada é corretamente liberada

Conclusão

O código demonstrou com sucesso o uso de multithreading para resolver um problema prático.

A aplicação é eficiente e modular.
