

Busca em arquivos com Threads

Uma das tecnologias implementadas em servidores e/ou computadores que visam um alto desempenho é o RAID 0. A técnica visa dividir um arquivo em partes e colocá-las em discos diferentes visando reduzir o tempo de busca. Dessa forma, dois cabeçotes podem buscar um arquivo na metade do tempo, por exemplo.

Com o uso de *threads* conseguimos simular algo semelhante ao que é feito em um RAID 0, em que tem-se um número *n* de *threads* desenvolvendo tarefas pseudo-paralelas, visando fragmentar o problema para tentar extrair ganho de desempenho em um computador.

O trabalho visa implementar um simulador de disco magnético, no qual existem várias *threads* (cada *thread* corresponde a um HD) buscando uma determinada informação (uma *string* passada via linha de comando).

Implementação:

- O trabalho deve ser implementado em linguagem C, utilizando a biblioteca *pthread*.
- O objetivo do trabalho é buscar, em arquivos de texto, uma determinada palavra informada pelo usuário (via linha de comando).
- O trabalho deverá implementar 3 casos, cada um em um arquivo .c independente:
 - 1º caso: o programa deve fazer a busca no arquivo "File1_test1.txt" **sem o uso de threads** (esse caso consiste na existência de um processo com um único fluxo – 1 thread) → nome do programa "buscathread1.c".
 - 2º caso: o programa deve fazer a busca nos arquivos "File1_test2.txt" e "File2_test2.txt" **com 2 threads** (cada *thread* busca em um dos arquivos). → nome do programa "buscathread2.c".
 - 3º caso: o programa deve fazer a busca nos arquivos "File1_test3.txt", "File2_test3.txt", "File3_test3.txt" e File4_test3.txt" **com 4 threads** (cada *thread* busca em um dos arquivos) → nome do programa "buscathread4.c".
- Ao final, cada programa deve exibir:
 - se a palavra procurada foi encontrada ou não;
 - o tempo de busca para encontrar a palavra informada. Observe que a contagem de tempo deve parar no momento em que a palavra for encontrada. Para a contagem de tempo, deve ser usada a função `clock()` da biblioteca `time.h`.
 - qual foi a thread que encontrou a palavra.

Análise de desempenho:

- Deverão ser realizadas, no mínimo, 10 execuções de cada um dos 3 casos para que se obtenha uma amostra aceitável e fiel. Com base nas execuções:
 - a) monte uma tabela com os dados obtidos (médias); e
 - b) elabore um gráfico comparando o desempenho em cada um dos 3 casos. O gráfico deve exibir a quantidade de threads utilizado (1, 2 ou 4) e o tempo decorrido (média dos valores obtidos nas execuções) até encontrar a palavra.
- Deve ser organizado um documento contendo a tabela com os valores obtidos nas execuções e o gráfico elaborado a partir da tabela. O documento (em formato pdf) deve ser enviado juntamente com os 3 arquivos .c em um arquivo .zip.

Observações:

- Trabalho individual.
- A string a ser buscada é **sistemas_operacionais**. Encontram-se no moodle, em um arquivo .zip, os seguintes arquivos para serem utilizados como entrada:
 - 1º caso: arquivo "File1_test1.txt" (a string encontra-se no arquivo);
 - 2º caso: arquivos "File1_test2.txt" e "File2_test2.txt" (a string está no File2_test.txt);
 - 3º caso: arquivos "File1_test3.txt", "File2_test3.txt", "File3_test3.txt" e File4_test3.txt" (a string está no File4_test3.txt)
- A apresentação do trabalho será agendada posteriormente.