

Universidade de São Paulo
Instituto de Ciências Matemáticas e de Computação
Departamento de Sistemas de Computação
Laboratório de Sistemas Distribuídos e de Programação Concorrente

Caderno de Desafio para Programação Paralela

Caderno Desafio C/MPI/OpenMP
Encontrar Palavras em Strings

por

Paulo Sérgio Lopes de Souza

Baseado em desafio anterior feito por Guilherme Martins e Paulo S. L. de Souza

Este caderno de desafio representa um Recurso Educacional Aberto para ser usado por alunos e professores, como uma introdução aos estudos de programação paralela com C/MPI/OpenMP. Este material pode ser utilizado e modificado desde que os direitos autorais sejam explicitamente mencionados e referenciados. Utilizar considerando a licença *GPLv2 (GNU General Public License version 2)* ou posterior.

São Carlos/BR, dezembro de 2020

1. Desafio

O objetivo deste desafio é desenvolver um algoritmo paralelo em C/MPI/OpenMP que encontre uma relação de palavras em *strings* dispostas em uma matriz, conforme especificação a seguir.

Considere uma matriz de *strings* **MS** com **L** linhas e **C** colunas, contendo dígitos decimais entre 0 e 9 (inclusive). Considere também outra matriz de palavras **MF** com **W** palavras. Verifique se cada uma das **W** palavras de **MF** existem em **MS**.

As **L** palavras de **MS** e as **W** palavras de **MF** têm tamanhos não superiores a **C** (sem contar o '\0') e assume-se que as palavras de **MF** não se repetem e só aparecem uma vez em **MS**. Uma *string* de **MS** pode ter mais de uma palavra de **MF**.

A aplicação concorrente executará com **P** processos *MPI* e cada processo com **T** *threads* *OpenMP*, onde considera-se que **L**, **C** e **W** são bem maiores que **P** e **T**. O valor de **P** pode ser determinado pelo *mpirun* com o parâmetro **-np <P>**. O valor de **T** pode ser dinâmico, i.e., em função do número de núcleos do computador onde o processo criador das *threads* vai executar (em outras palavras, não precisa especificar o número **T** de *threads*, deixe o *OpenMP* escolhê-lo em função do processador usado).

Os dados de entrada (**MS**, **L**, **C**, **MF** e **W**) são carregados de um arquivo texto no início da execução. O arquivo de entrada contém em sua primeira linha os valores de **L** e **C** (números de linhas e colunas de **MS**); nas próximas **L** linhas as *strings* de **MS** (cada linha de **MS** está em uma linha do arquivo de entrada); a quantidade **W** de palavras de **MF** e nas próximas **W** linhas as palavras em si de **MF**, novamente uma palavra por linha.

No final, a aplicação mostra como resultado, para cada palavra de **MF** a ser buscada em **MS**, em qual linha e coluna esta palavra se encontra em **MS**, caso ela esteja presente em **MS**. Se a palavra não for encontrada em **MS**, o número da linha e da coluna no arquivo de saída devem ser ambos **-1**. O arquivo de saída contém uma linha para cada palavra a ser procurada na matriz, com os seguintes resultados: <palavra>,<linha>,<coluna> (sem espaços entre eles e com uma quebra de linha no final da linha).

Para executar no *bash*, por exemplo, utilize este padrão:

```
mpirun -np 4 findwords entrada.txt -fopenmp
```

Obs: na linha de comando acima, considera-se que o programa foi inserido em *findwords.c* e o executável chama-se *findwords* e está no diretório atual. O número de processos (**-np 4**) é um exemplo; e não um requisito, pois o algoritmo deve funcionar para um número variado de processos e de *threads*. Os dados para a computação necessária estão em *entrada.txt*, seguindo o padrão já explicado.

Exemplo de arquivos de entrada e saída (os valores a partir de “<<” são comentários apenas e não fazem parte do arquivo de entrada ou da saída):

10 10	<< L e C
0000012345	<< dígitos da matriz MS
9876543111	<< dígitos da matriz MS
5122222222	<< dígitos da matriz MS
3333753193	<< dígitos da matriz MS
4445494444	<< dígitos da matriz MS
0695555555	<< dígitos da matriz MS
6666826666	<< dígitos da matriz MS
7777777777	<< dígitos da matriz MS
2929292988	<< dígitos da matriz MS
1818181818	<< dígitos da matriz MS
12	<< W
12345	<< palavra a buscar. MF
9876543	<< palavra a buscar. MF
512	<< palavra a buscar. MF
75319	<< palavra a buscar. MF
549	<< palavra a buscar. MF
69	<< palavra a buscar. MF
826	<< palavra a buscar. MF
0101	<< palavra a buscar. MF . Esta não existe.
29292929	<< palavra a buscar. MF
1818181818	<< palavra a buscar. MF
3416	<< palavra a buscar. MF . Esta não existe.
4444	<< palavra a buscar. MF .

Exemplo de arquivo de saída contendo **palavra a buscar,linha,coluna**. Os caracteres depois de **palavra a buscar,linha,coluna** não são impressos. Siga o padrão dos fontes fornecidos. Há uma quebra de linha (“\n”) após cada linha impressa.

12345,0,5	<< palavra procurada, posição linha e coluna
9876543,1,0	<< palavra procurada, posição linha e coluna
512,2,0	<< palavra procurada, posição linha e coluna
75319,3,4	<< palavra procurada, posição linha e coluna
549,4,3	<< palavra procurada, posição linha e coluna
69,5,1	<< palavra procurada, posição linha e coluna
826,6,4	<< palavra procurada, posição linha e coluna
0101,-1,-1	<< palavra procurada, esta não existe
29292929,8,0	<< palavra procurada, posição linha e coluna
1818181818,9,0	<< palavra procurada, posição linha e coluna
3416,-1,-1	<< palavra procurada, esta não existe
4444,4,6	<< palavra procurada, posição linha e coluna