



FACULDADE DE TECNOLOGIA

Documentação do Código
Sistemas Operacionais
FT - UNICAMP
Bacharelado em Sistemas de Informação

Leonardo Furone - 201192
Henrique Campiotti Marques - 198760

Introdução

No projeto proposto pela disciplina, devemos desenvolver um código em que seja possível encontrar, dentro uma matriz $M \times N$, um determinado valor através de várias threads. Além disso, no relatório, colocamos o desempenho do programa com 2, 4, 8 e 16 threads. Porém, aqui, o objetivo é mostrar o código e fazer uma documentação básica, buscando maior entendimento de todos do código.

Código*

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <pthread.h>

int      M, N, T, l, c, ret;
float    **matriz, V;

void* SearchFunction (void *arg);
float** CreateMatriz (int M, int N);

int main()
{
    char    nome_arquivo[15];
    int     i, j, aux;

    printf("Dimensao: ");      scanf("%d %d", &M, &N);
    printf("NUM_Threads: ");  scanf("%d", &T);
    printf("Valor: ");        scanf("%f", &V);
    getchar();
    printf("Nome_Arquivo: ");  scanf("%s", nome_arquivo);

    FILE* file = fopen(nome_arquivo, "r");
    if(file == NULL){
        printf("Arquivo nao encontrado\n");
        return 0;
    }

    matriz = CreateMatriz(M, N);
    pthread_t thread_id[T];

    for(l = 0; l < M; l++){
        for(c = 0; c < N; c++){
            fscanf(file, "%f", &matriz[l][c]);
        }
    }
}
```

```
        aux = M / T;    l = 0;          c = 0;    ret = 0;

        for(j = 0 ;j < aux; j++){
            for(i = 0; i < T; i++){
                pthread_create(&thread_id[i], NULL,
SearchFunction, NULL);

                pthread_join(thread_id[i], NULL);
            }
            //    for(i=0; i<T; i++){ pthread_join(thread_id[i], NULL); }
        }

        if(T < M){
            aux = M - aux * T;
            for(i = 0; i < aux; i++){
                pthread_create(&thread_id[i], NULL,
SearchFunction, NULL);

                pthread_join(thread_id[i], NULL);
            }
            //    for(i=0; i<aux; i++){ pthread_join(thread_id[i], NULL); }
        }

        if(ret == 0){ printf("Valor nao encontrado\n"); }

return 0;
}

void* SearchFunction (void *arg)
{
    while(c < M){
        if(matriz[l][c] == V){ printf("%d - %d\n", l, c); ret = 1; }
        c++;
    }

    c = 0;    l++;
}

float** CreateMatriz(int M, int N)
{
    int i, j;

    float **m = (float**)malloc(M * sizeof(float*));

    for (i = 0; i < M; i++){
        m[i] = (float*) malloc(N * sizeof(float));
        for (j = 0; j < N; j++) { m[i][j] = 0.0; }
    }

    return m;
}
```

*Esse é o mesmo código do GitHub

Documentação

Dessa forma, temos:

Na linha 6:

M número de linhas da matriz;
N número de colunas da matriz;
l, c contadores.

Na linha 7:

matriz ponteiro para uma
matriz.

Nas linhas 9 e 10:

protótipos de função.

Na linha 12:

função principal.

Nas linhas 18 a 20:

dados de entrada.

Na linha 21:

remover preenchimento do
buffer.

Na linha 25:

cria ponteiro para o arquivo em
modo de leitura.

Nas linhas 26 a 29:

verifica a existência do arquivo.

Na linha 31:

cria matriz com as dimensões
de entrada.

Na linha 32:

declara as threads.

Nas linhas 34 a 38:

lê os dados do arquivo para a
matriz

Na linha 40:

inicializa as variáveis.

Na linha 43:

faz a chamada do inteiro de
linhas.

Na linha 45:

cria a thread.

Na linha 46:

dispara a thread.

Na linha 51 a 56:

faz a chamada do restante se
existir.

Na linha 64:

função das threads.

Na linha 66:

percorre as colunas da matriz.

Na linha 67:

verifica a existência de valor.



Na linha 68:

incrementa o valor das colunas.

Na linha 71:

zera as colunas e passa pra próxima linha.

Na linha 74:

função de criação da matriz por alocação.

Na linha 79:

aloca as linhas.

Na linha 82:

aloca as colunas.

Na linha 83:

inicializa com 0.0 os elementos da mesma.

Na linha 85:

retorna ponteiro para a matriz.

Considerações Finais

Dessa forma, acreditamos que o código esteja funcionando, com o código desenvolvido e explicado para sanar qualquer dúvida que possa ter ficado pendente de sua criação.