

1. Exercício 1

O programa fonte do exercício 1 encontra-se no arquivo `adiciona_matrizes_parl.c`. Ele foi feito aproveitando-se o programa fonte disponibilizado pela professora que iniciava duas matrizes com 4 processos. No caso, apenas adicionou-se uma linha que adiciona os valores nas matrizes A e B numa nova matriz C. Para que isto seja possível, adicionou-se uma nova estrutura para o tipo definido no início do programa.

Seguem abaixo as modificações no programa e o resultado obtido no terminal.

Figura 1: Matriz C adicionada ao tipo matrizes.

```
typedef struct {  
    double a[1000][1000], b[1000][1000], c[1000][1000];  
} matrizes;  
  
matrizes *matriz;  
pid_t pid_p;  
  
void main()  
{
```

Figura 2: adicionada linha que soma os valores de A e B e guarda em C

```
for (i=inicio; i<fim; i++){  
    for (j=0; j<size; j++){  
        ((matrizes *)matriz)->a[i][j]=1.5;  
        ((matrizes *)matriz)->b[i][j]=2.6;  
        ((matrizes *)matriz)->c[i][j] = matriz->a[i][j] + matriz->b[i][j];  
    }  
  
if ((int)pid_p == 0){ //filhos  
    shmdt (matriz);  
    exit(1);  
}  
  
// fim da inicializacao
```

Figura 3: Resultado do programa

```
luciana@luciana-K46CA:~/Documents/PolI/PCS3868-Desempenho/aula02/programas$ gcc inicia-matrizes-proc2.c  
inicia-matrizes-proc2.c: In function 'main':  
inicia-matrizes-proc2.c:50:14: warning: implicit declaration of function 'fork' [-Wimplicit-function-declaration]  
    pid_p=fork();  
          ^~~~~  
inicia-matrizes-proc2.c:79:9: warning: implicit declaration of function 'waitpid' [-Wimplicit-function-declaration]  
    waitpid(pid[k],&status,0);  
    ^~~~~~  
luciana@luciana-K46CA:~/Documents/PolI/PCS3868-Desempenho/aula02/programas$ ./a.out  
inicio=250 fim=500 k=0 n_iter=250  
inicio=0 fim=250 k=2 n_iter=250  
inicio=750 fim=1000 k=2 n_iter=250  
wait pid  
inicio=500 fim=750 k=1 n_iter=250  
wait pid  
wait pid  
saiu  
c[0][0]=4.100000 c[15][20]=4.100000 c[size-1][size-1]=4.100000
```

2. Exercício 2

O segundo exercício da aula também visa paralelizar um algoritmo de soma de matrizes, porém utilizando threads invés de processos. Utilizou-se como ponto de partida o programa fornecido `inicia-matrizes-pthread.c`, sendo este modificado para salvar o valor das matrizes A e B numa nova matriz C.

Figura 4: Matriz C adicionada ao programa

```
double a[1000][1000], b[1000][1000], c[1000][1000];
```

Figura 5: Adicionado comando que salva a soma de A e B em C

```
for (i=inicio;i<fim;i++)  
    for (j=0;j<size;j++){  
        a[i][j]=1.5;  
        b[i][j]=2.6;  
        c[i][j] = a[i][j] + b[i][j];  
    }
```

Figura 6: Execução do programa

```
luciana@luciana-K46CA:~/Documents/Pol/PCS3868-Desempenho/aula02$ gcc -pthread soma-matrizes-pthread.c  
luciana@luciana-K46CA:~/Documents/Pol/PCS3868-Desempenho/aula02$ ./a.out  
inicio=500 fim=750 k=1 n_iter=250  
inicio=0 fim=250 k=-1 n_iter=250  
inicio=250 fim=500 k=0 n_iter=250  
inicio=750 fim=1000 k=2 n_iter=250  
c[0][0]=4.100000 c[15][20]=4.100000 c[size-1][size-1]=4.100000
```