

Relatório de atividades

Adriano Antonio da Silva 10773756
Greghory do Amaral 11375354

Aula 03 - Sistema CPAR
PCS3568 - Sistemas de Computação de Alto Desempenho
Profa.: Liria M. Sato

- **Exercício 01**

Para o exercício 1 no início do programa foram criadas as Tasks de 1 a 6 que imprimem as letras de “a” a “f”. Cada Task imprime o caracter e em seguida limpa o buffer. O programa “main” imprime a string “Seq1”, em seguida entra no comando “cobegin” “also” que altera entre a execução de duas sequências. Na primeira sequência estão a chamada de função da task 1 e 2, após executar as funções correspondentes a estas Tasks a primeira sequência de comandos imprime “Seq2” e em seguida chama as funções que executam as funções correspondentes às Tasks 5 e 6. A segunda sequência abaixo do comando “also” chama as funções que executam as Tasks 3 e 4 e em seguida imprime “Seq3”. O resultado do programa pode ser visto logo abaixo:

```
[grp01@seya cpar-2024]$ ./exer1
SE01
aaaaaaaaabbbbbbbcccccccc
SE02
dddddddddfffffffddeeeeeeee
SE03
[grp01@seya cpar-2024]$
```

- **Exercício 02**

No segundo exercício o código começa pedindo o tamanho da matriz. Em seguida há a chamada de duas funções, “init” e “mult”. “init” realiza a inicialização das matrizes em paralelo. “mult” realiza a multiplicação em paralelo conforme o exemplo dado. Usamos o comando create 4 para especificar o uso de 4 processadores na paralelização da multiplicação em 4 processos. Esperamos a inicialização terminar para começar a multiplicação e a multiplicação terminar para contar o tempo. O programa sequencial teve um tempo de execução de 4.7 s enquanto o programa paralelizado executou num tempo de 1.1 s, o resultado dos programas pode ser visto logo abaixo:

```
[grp01@seya cpar-2024]$ ./exercicio2-seq
Tamanho da matriz: 1000
TEMPO [1000]: 4.7322320
c[ 0][0]=1288221239250000.000000 c[ 300][300]=1427585339250000.000000 c[ 999][999]=1780235732250028.500000
[grp01@seya cpar-2024]$ ./exer2
Tamanho da matriz: 1000
Tempo da operacao de tamanho 1000: 1.1143560
c[ 0][0]=1288221239250000.000000 c[ 300][300]=1427585339250000.000000 c[ 999][999]=1780235732250028.500000
[grp01@seya cpar-2024]$
```

- **Exercício 03**

No último programa, nós criamos algumas variáveis globais para auxiliar, uma task de inicialização da matriz e uma task para realizar a busca, no caso da busca, utilizamos um vetor de 4 posições para que cada ocorrência pudesse ser contabilizada em uma posição do vetor sem que houvesse interferência de diferentes threads escrevendo em momentos diferentes, ao final, contabilizamos as ocorrências em uma só variável e exibimos na tela. O programa sequencial teve um tempo de execução de 0.02s enquanto o programa paralelo executou num tempo de 0.003s o funcionamento do programa pode ser visto logo abaixo:

```
[grp01@seya cpar-2024]$ ./busca-seq
Tamanho da matriz (100 ou 1000 ou 2000): 2000
digite um numero de 0 a 9999:0
0
TEMPO [2000]: 0.0211410
NUMERO DE OCORRENCIAS=3999
[grp01@seya cpar-2024]$ ./exer3
Tamanho da matriz (100 ou 1000 ou 2000): 2000
digite um numero de 0 a 9999:0
0
2499 500 500 500TEMPO [2000]: 0.0039720
NUMERO DE OCORRENCIAS=3999
```