Universidade de Brasília
Instituto de Ciências Exatas
Departamento de Ciência da Computação
Programa de Pós-graduação em Informática

Exercício de Programação 2 (EP2)

1. Identificação da Disciplina

Código da Disciplina	Nome da Disciplina	Turma	Professor	Período	Exercício
316342	Programação Paralela	A	George Luiz Medeiros Teodoro	2014/2	EP2

2. *Shoelace Theorem*

O teorema *Shoelace* é uma fórmula para encontrar a área de um polígono irregular, a partir das coordenadas de seus vértices. A fórmula é dada por:

$$A = \frac{1}{2}\left|\sum_{i=1}^{n-1}(x_i y_{i+1} + x_n y_1) - \sum_{i=1}^{n-1}(x_{i+1} y_i - x_1 y_n)\right|$$

Onde A é a área do polígono, n é o número de lados do polígono e (x_i, y_i) representa o vértice i do polígono (com i variando de $1 \ldots n$).

A aplicação desse teorema requer as seguintes propriedades:

- Os vértices do polígono devem estar ordenados no sentido "horário";
- O polígono deverá ser fechado e o último vértice deverá ser igual ao primeiro.
- O polígono não deve atravessar, sobrepor ou interceder (gerar área negativa por sobreposição) consigo mesmo.

3. Enunciado

Deve-se implementar um algoritmo paralelo utilizando *PThreads* que calcule a área de polígonos irregulares, utilizando o teorema Shoelace.

3.1. Entrada do Programa

O programa deverá receber, em sua chamada, dois parâmetros como entrada: uma string s que representará os tipos de saída ("$area$" para imprimir somente a área, "$time$" para imprimir somente o tempo e "all" para imprimir a área na primeira linha e o tempo na segunda linha) que devem ser impressas e um inteiro t para o número de threads. Após a chamada, serão informados, no início da execução, um inteiro n com a quantidade de vértices do polígono e n vértices x e y (tipo *double*) representando a posição dos vértices.

3.2. Saída do programa

O programa deverá imprimir a área com 2 casas decimais e o tempo em microssegundos. O resultado deverá ser impresso na saída padrão (stdout) no seguinte o formato, de acordo com a string s de entrada:

- "$area$": Uma linha contendo a área com duas casas decimais.
- "$time$": Uma linha contendo o tempo de execução em microssegundos.
- "all": A primeira linha contendo a área com duas casas decimais e e a segunda linha com o tempo de execução em microssegundos.

Após a exibição, acrescente uma quebra de linha - `printf("\n")` - na saída.

4. Exemplos de Execução

A Tabela 1 possui alguns exemplos de execução, de acordo com os parâmetros exigidos na entrada e saída do programa.

5. Tempo de execução

Como *PThreads* não oferece uma função para cálculo de tempo, recomenda-se o uso da função `gettimeofday()` da biblioteca time.h (ou sys/time.h dependendo do sistema). Para isso, deve-se declarar duas variáveis do tipo *struct timeval* e chamar a função da seguinte maneira:

```
gettimeofday(&start,NULL);

//Código a ser medido

gettimeofday(&end,NULL);
```

Para calcular o tempo de execução, utiliza-se o seguinte comando:

```
time =
((endT.tv_sec*1000000+endT.tv_usec)-(beginT.tv_sec*1000000+beginT.tv_usec));
```

Table 1. Tabela com exemplos de execução

Nr	Exemplo de entrada	Exemplo de Saída
01	./prog time 3 5 0 0 0 8 8 8 8 0 0 0	304
02	./prog area 3 5 0 0 0 8 8 8 8 0 0 0	64.00
03	./prog all 3 5 0 0 0 8 8 8 8 0 0 0	64.00 263
04	./prog all 3 < exemplo.txt	64.00 263

6. Arquivos de entrada

Para a execução do programa desenvolvido, alguns arquivos contidos em `http://164.41.209.62` constituem polígonos irregulares e podem ser utilizados como entrada para o programa (mostrado no exemplo 4 da Tabela 1).