

**UNIVERSIDADE FEDERAL DE ALAGOAS
INSTITUTO DE COMPUTAÇÃO**

PROGRAMAÇÃO PARA SISTEMAS EMBARCADOS

CÉSAR AUGUSTO DOS SANTOS CAETANO

DEFINIÇÃO

Na matéria de programação para sistemas embarcados, aprendemos a utilizar algumas técnicas de programação para otimizar o desempenho de algumas operações matemáticas. Esse trabalho visa a implementação e a comparação entre funções otimizadas.

IMPLEMENTAÇÃO

Seguiremos a seguinte ordem de operações:

- POLINÔMIO
- SENO
- DIVISÃO POR CONSTANTE
- MEDIA

Na função slowfunction implementamos as funções da maneira mais fácil porém computacionalmente custosa.

- Polinômio implementado com 6 multiplicações
- Seno nativo da biblioteca math.h
- Divisão padrão (operação muito custosa)
- Media calculada em cada iteração

```
long int slowfunction()
{
    clock_t start, end;
    start = clock();
    int x;
    slowmedia = 0;

    for(int i = 0 ; i < 1000 ; i++) {

        x = i>>4;
        // 3X³+5X²+8 x +5 POLINÔMIO
        double y= (3*x*x*x) + (5*x*x) + (8*x) +5;
        //SENO
        int z= sin (y) ;
        //DIVISÃO POR CONSTANTE
        double w = z/10;
        //MEDIA
        slowmedia = (slowmedia +w) /(i+1);
    }

    end = clock();

    return end-start;
}

long int fastfunction(){
```

Na função fastfunction foram implementadas as melhorias nas operações.

- Fatoração do polinômio. Diminui a o número de multiplicações de 6 para 3.
- Seno calculado através da série de Taylor
- Divisão feita através de multiplicação e deslocamento
- Média calculada uma vez após o final do laço de repetição.

```
36
37 long int fastfunction(){
38     clock_t start, end;
39     start = clock();
40     int x;
41     fastmedia =0;
42     for(int i = 0 ; i < 1000 ; i++) {
43         x = i>>4;
44         // 3X3+5x2+8 x +5 POLINÔMIO
45         double y = (((3*x) + 5)*x+8) *x)+5;
46         //SENO EM SÉRIES DE TAYLOR DE 4ª ORDEM
47         int z = y - (y*y*y / 6.0) + (y*y*y*y*y / 120.0) - (y*y*y*y*y*y*y / 5040.0);
48         //DIVISÃO POR CONSTANTE
49         double w = (z *171)>>10;
50         //MÉDIA RÁPIDA
51         fastmedia = (fastmedia+ w);
52         samplesFastMean ++;
53
54
55
56
57
58     }
59     fastmedia = fastmedia /samplesFastMean;
60
61
62     end = clock();
63     //printf("Tempo total = %lf", v[1]);
64     return end-start;
65 }
66
```

AValiação

```
68
69 int main()
70 {
71     double x = slowfunction();
72     double y = fastfunction();
73
74     printf("SLOW-> CLOCKS: %f \n",x );
75     printf("FAST-> CLOCKS: %f \n",y );
76     long double perc = ((x-y)/x)*100;
77     printf("PORCENTAGEM-> %Lf\n ", perc);
78
79
80     return 0;
81 }
82
```

Obtemos as seguintes porcentagens

```
cesar@cesar-Lenovo-G40-80: ~/Documentos/UFAL/OITAVO/PSE/PSE/Desafio02
SLOW-> CLOCKS: 240.000000
FAST-> CLOCKS: 27.000000
PORCENTAGEM-> 88.750000
cesar@cesar-Lenovo-G40-80:~/Documentos/UFAL/OITAVO/PSE/PSE/Desafio02$ ./main
SLOW-> CLOCKS: 272.000000
FAST-> CLOCKS: 36.000000
PORCENTAGEM-> 86.764706
cesar@cesar-Lenovo-G40-80:~/Documentos/UFAL/OITAVO/PSE/PSE/Desafio02$ ./main
SLOW-> CLOCKS: 420.000000
FAST-> CLOCKS: 47.000000
PORCENTAGEM-> 88.809524
cesar@cesar-Lenovo-G40-80:~/Documentos/UFAL/OITAVO/PSE/PSE/Desafio02$ ./main
SLOW-> CLOCKS: 420.000000
FAST-> CLOCKS: 48.000000
PORCENTAGEM-> 88.571429
cesar@cesar-Lenovo-G40-80:~/Documentos/UFAL/OITAVO/PSE/PSE/Desafio02$ ./main
SLOW-> CLOCKS: 417.000000
FAST-> CLOCKS: 48.000000
PORCENTAGEM-> 88.489209
cesar@cesar-Lenovo-G40-80:~/Documentos/UFAL/OITAVO/PSE/PSE/Desafio02$ ./main
SLOW-> CLOCKS: 416.000000
FAST-> CLOCKS: 49.000000
PORCENTAGEM-> 88.221154
cesar@cesar-Lenovo-G40-80:~/Documentos/UFAL/OITAVO/PSE/PSE/Desafio02$
```