

Laboratório 18

Instruções de Ponto Flutuante

1) Traduza a função “calc1” para Assembly:

```
float calc1(double a, float b) {  
    return (a+b)*(a-b);  
}
```

Note que:

- Só podemos efetuar as operações de soma, subtração e multiplicação com operandos do mesmo tipo.
- O retorno da função é um *float*.

Faça um arquivo em C com uma função “main” para testar a função “calc1”.

Dica: compare a saída da “calc1” em Assembly com a “calc1” de C.

2) Traduza a função “calc2” para Assembly:

```
float calc2(float a, float b) {  
    return sin(a) + b;  
}
```

Lembre-se que os registradores %xmm são *caller-saved*, ou seja, eles podem ser sobrescritos quando uma outra função é chamada. Se necessário, use a pilha para salvar os valores.

A função “sin” possui a seguinte assinatura:

```
double sin(double x);
```

Se for usar em C, deve-se adicionar o cabeçalho “math.h”

Para compilar, é necessário adicionar a biblioteca matemática “-lm” na linha do GCC:

```
gcc -o main main2.c calc2.s -lm
```

Continua →

3) Traduza a função “calc3” para Assembly:

```
double calc3(float a, float b) {  
    return sin(a) + cos(b);  
}
```

A função “cos” possuem as seguintes assinaturas:

```
double cos(double x);
```

4) Traduza a função “calc4” para Assembly:

```
double calc4(double *a, int n) {  
    int i;  
    double r = 0.0;  
    for (i = 0; i < n; i++) {  
        r += sin(*a);  
        a++;  
    }  
    return r;  
}
```