

## Laboratório 10

### Assembly: Chamada de Função, Parâmetros e Registro de Ativação

1. Traduza para Assembly a função *add()* a seguir, criando um arquivo “add.s”.

```
int add (int a, int b, int c) {  
    return a+b+c;  
}
```

Crie um arquivo “main\_add.c” e escreva uma função *main()* que chame essa função *add()*. Por exemplo:

```
#include <stdio.h>  
  
int add (int a, int b, int c);  
  
int main (void) {  
    printf("%d\n", add(1,2,3));  
    return 0;  
}
```

Compile os dois arquivos juntos e depois execute “main\_add”:

```
gcc -Wall -o main_add add.s main_add.c
```

2. Traduza a função *fat()* abaixo para Assembly (arquivo “fat.s”). Crie o arquivo “main\_fat.c” com uma função *main()* em C que chame a função *fat()*.

```
int fat (int n) {  
    if (n==0) return 1;  
    else      return n*fat(n-1);  
}
```

Compile os dois arquivos juntos e depois execute “main\_fat”:

```
gcc -Wall -o main_fat fat.s main_fat.c
```

Continua →

3. Traduza para Assembly a função *foo()* abaixo:

```
void foo (int a[], int n) {  
    int i;  
    int s = 0;  
    for (i = 0; i < n; i++) {  
        s += a[i];  
        if (a[i] == 0) {  
            a[i] = s;  
            s = 0;  
        }  
    }  
}
```

**Atenção:** a tradução deverá representar a indexação do array “a” (fórmula de acesso ao array).

Crie um arquivo “main\_foo.c” com uma função *main()* em C que chame *foo()*. Compile os dois módulos juntos e execute. Verificando se a saída do programa está correta.