



# Aula 8 , Exemplos de Funções e Recursão - 22/12/22

## Exemplos

Exemplo 1: em C

```
int exemplo1(int g, int h, int i, int j)
{
    int f;
    f = (g+h) - (i+j);
    return f;
}
```

Em Assembly:

```
exemplo1:
    add $t0, $a0, $a1
    add $t1, $a2, $a3
    sub $v0, $t0, $t1
    jr $ra ;retornar o procedimento
;Para chamar esse procedimento:
    jal exemplo1 |li $a0, 1| li $a2, 10
                    |li $a1, 7| li $a3, 8
```

Exemplo 2: Em C

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

Em Assembly:

```
fat:
    slti $t0, $a0, 1 ; n < 1 ?
    beq $t0, $zero, rec
    jr $ra

rec:
    addi $sp, $sp, -8 ; Salva na pilha o conteúdo de $a0 e $ra
    sw $a0, 0($sp)    ; que devem ser preservados
    sw $ra, 4($sp)
    addi $a0, $a0, -1 ; $a0 = n - 1
    jal fat
    lw $a0, 0($sp)
    lw $ra, 4($sp)
    addi $sp, $sp, 8
    mul $v0, $a0, $v0 ; n*fat(n-1)
    jr $ra
```

## Caracteres

A codificação padrão de caracteres é feita de acordo com a tabela ASCII.

→ caracteres correspondem a um código numérico de 0 a 127.

Para processar uma string em assembly MIPS:

1. Salva a String na memória
  - a. cada caractere ocupa 1 byte (8 bits - observe a tabela ASCII)
2. Acessa o caractere usando a instrução lbu (load byte unsigned)
  - a. lbu \$t0, 0(\$s0)
  - b. Salva em \$t0 o byte que estiver no endereço \$s0 + 0. Completa os demais bits de \$t0 com 0s.

