

# Organização de Computadores I - Trabalho 1

Guilherme Bezerra dos Santos - 2018055040

## 1 Evens and Odds

---

```
.data
argumento: .word 7

.text
main:
    lw x10, argumento
    call Impar_Par
    jal Fim

    Impar_Par:
        addi sp, sp, -4
        sw x5, 0(sp)
        addi x5, x0, 2
        rem x10, x10, x5
        lw x5, 0(sp)
        jalr x0, 0(x1)

Fim:
```

---

## 2 Factorial

---

```
.data
argumento: .word 5

.text
main:
    lw x10, argumento
    call Fatorial
    jal Fim

    Fatorial:
        # Guardar x5 e x6 na memoria
        addi sp, sp, -4
        sw x5, 0(sp)
        # x5 = x10
        add x5, x10, x0
```

```

bne x5, x0, Loop
# Fatorial de 0 = 1
addi x10, x0, 1
beq x0, x0, Exit
# enquanto x5 for diferente de 0
Loop:
# x10 = x5 * (x5-1)
    addi x5, x5, -1
    beq x5, x0, Exit
    mul x10, x10, x5
    beq x0, x0, Loop
Exit: # x5 == 0
    lw x5, 0(sp)
    jalr x0, 0(x1)

```

Fim:

---

### 3 Permutation

---

```

.data
tamanho: .word 4
vetor2: .word 5 3 3 5
vetor1: .word 3 5 5 3

.text
main:
    lw x10, tamanho # x10 = n
    la x11, vetor1 # x11 = Vetor 1
    call Sort # Ordena vetor 1
    add x5, x0, x11 # x5 = x11 = Vetor 1
    la x11, vetor2 # x11 = Vetor 2
    call Sort # Ordena vetor 2
    add x12, x11, x0 # x12 = Vetor 2
    add x11, x0, x5 # x11 = Vetor 1
    call ChecaPermutacao
    jal Fim

```

```

Sort:
    #x10 = tamanho, x11 = "vetor"
    #x5 = auxiliar
    addi sp, sp, -20
    sw x5, 0(sp)
    sw x6, 4(sp)
    sw x28, 8(sp)
    sw x29, 12(sp)
    sw x30, 16(sp)
    addi x5, x0, 0

```

```

addi x28, x0, 0
addi x29, x0, 0
addi x30, x0, 0
addi x10, x10, -1
Sort_Loop1:
    addi x6, x0, 0
    Sort_Loop2:
        slli x30, x6, 2 # x30 = j
        add x30, x11, x30 # x30 = vetor[j] (ponteiro)
        lw x28, 0(x30) #x28 = vetor[j]
        lw x29, 4(x30) #x29 = vetor[j+1]
        blt x28, x29, Pos_Troca # se x28 > x29: Troca
    Troca:
        sw x28, 4(x30) # vetor[j+1] = vetor[j]
        sw x29, 0(x30) # vetor[j] = vetor[j+1] (valor antigo)
    Pos_Troca:
        addi x6, x6, 1 # j++
        bne x6, x10, Sort_Loop2 # se j != n

    addi x5, x5, 1
    bne x5, x10, Sort_Loop1

lw x5, 0(sp)
lw x6, 4(sp)
lw x28, 8(sp)
lw x29, 12(sp)
lw x30, 16(sp)
addi x10, x10, 1
jalr x0, 0(x1)

ChecaPermutacao:
    #x10 = tamanho; x11 = vetor 1; x12 = vetor 2
    #x10 = 1 se sao permutacao, 0 se nao
    addi sp, sp, -16
    sw x5, 0(sp) #x5 = iterador
    sw x6, 4(sp) #x6 = vetor1[i]
    sw x7, 8(sp) #x7 = vetor2[i]
    sw x28, 12(sp) #x28 = i
    add x5, x0, x0
    Loop_P:
        slli x28, x5, 2 # x28 = i
        add x6, x11, x28 # x6 = vetor1[i] (ponteiro)
        lw x6, 0(x6)
        add x7, x12, x28 # x7 = vetor2[i] (ponteiro)
        lw x7, 0(x7)
        bne x6, x7, Diferentes # se x6 != x7: Retorna 0
        beq x0, x0, Iguais
    Diferentes:
        addi x10, x0, 0
        lw x5, 0(sp)

```

```

        lw x6, 4(sp)
        lw x7, 8(sp)
        jalr x0, 0(x1)
    Iguais:
        addi x5, x5, 1 # i++
        bne x5, x10, Loop_P # se i != n
addi x10, x0, 1
lw x5, 0(sp)
lw x6, 4(sp)
lw x7, 8(sp)
lw x28, 12(sp)
jalr x0, 0(x1)

```

Fim:

---

## 4 Referências

Patterson, D., Hennessy, J.: Computer Organization and Design - RISC-V Edition. Acesso em 27 ago. 2019.