

# **Projeto SPIM - Agenda**

**Carlos O. Cunha Filho**

**César H. Kallas**

O Centro de Ciências Exatas, Ambientais e de Tecnologias  
Pontifícia Universidade Católica de Campinas – Campinas – Brasil  
Faculdade de Engenharia de Computação

carcunha @ yahoo.com.br RA: 00066936

cesarkallas @ cesarkallas.net RA: 02099224

## **Resumo:**

*O projeto consiste no desenvolvimento de uma aplicação em MIPS que execute a função de uma agenda simples de contatos telefônicos.*

*A agenda de contatos terá as seguintes opções:*

- *Adicionar contato*
- *Remover contato*
- *Pesquisar contato*
- *Status dos contatos*
- *Imprimir contatos de forma ordenada (crescente)*

## **1. Propósito do projeto**

O projeto consiste no desenvolvimento e implementação de uma agenda no simulador SPIM de processadores MIPS, que executasse as instruções de manipulação de contatos.

A implementação foi feita na linguagem assembly do MIPS, usando como auxílio o software SPIM.

### **1.2 Descrição do projeto**

O projeto possui 1 arquivo de programação:

- agenda-mips.asm

Possui o código da agenda, que inclui as opções de manipulação de contatos, como: adicionar, remover, pesquisar e ordenar.

## 1.3 Especificação

### 1.3.1 Armazenamentos dos dados

Os dados dos contatos da agenda são armazenados em um registrador (\$s6), esse registrador mantém os valores enquanto o programa é executado.

No registrador (\$s7) é armazenado o número de contatos que a agenda possui, para que seja possível manipulá-los posteriormente.

O registrador de dados (\$s6) é iniciado a partir da soma do endereço do registrador (\$gp) e o número inteiro 5000.

O armazenamento de cada contato é feita em porção de 8 posições no registrador (\$s6), sendo as 4 primeiras posições para o nome do contato e as outras 4 posições para o número de telefone (inteiro).

### 1.3.2 Opções da agenda

Opção	Função
1	Adiciona contato na agenda.
2	Remove contato da agenda, a partir do telefone.
3	Pesquisa contato, a partir do telefone.
4	Imprime os contatos de forma ordenada (crescente).
5	Exibe o total de contatos na agenda.
0	Sai da agenda.

### 1.3.3 Registradores de dados

Registrador	Descrição
\$s6	Índice do vetor de dados
\$s7	Total de contatos na agenda
\$t5	Último nome do contato manipulado (adicionado ou pesquisado).
\$t6	Último telefone do contato manipulado (adicionado ou pesquisado).

### 1.3.4 Funções

#### Adicionar contato

Lê o nome e o telefone do contato

Grava na posição atual do registrador (\$s6) e o incrementa

Incrementa o registrador (\$s7).

#### Remover contato

Lê o número de telefone do contato a ser removido

Pesquisa nos contatos a partir do número de telefone, se achar, sobrescreve com o último contato da agenda, se não achar, retorna não encontrado.

Decrementa o total de contatos em 1.

#### Ordenar contatos

Usa o algoritmo Bubble Sort ( $O_n^2$ ) para ordenar de forma crescente a partir do telefone.

### 1.3.6 Anexos

#### agenda-mips.asm

```
# Projeto SPIM - Agenda
# Cesar H. Kallas                                cesarkallas @ cesarkallas.net RA: 02099224
# Carlos O. Cunha Filho  carcunha @ yahoo.com.br RA: 00066936

# Especificação do Projeto 2:
# Implemente um programa em linguagem de montagem que funciona como uma agenda de telefones,
# onde é apresentado ao usuário um menu com as seguintes opções:
# 1) Incluir contato: onde o usuário insere na agenda as informações nome e número de telefone de um
# contato;
# 2) Excluir contato: onde o usuário informa um número de telefone e o contato que possui tal telefone é
# removido da agenda;
# 3) Buscar contato: onde o usuário informa um número de telefone e o contato que possui tal telefone é
# exibido na tela; e
# 4) Imprimir agenda: onde são exibidos na tela todos os contatos existentes na agenda, organizados em
# ordem crescente por número de telefone.

# versao final   - 15/04/2007

.data
msgBemVindo:      .asciiz "\n\nAgenda em MIPS"
msgOpcoes:        .asciiz "\nOpcoes: \n1. Adicionar\n2. Remover\n3. Pesquisar\n4. Imprimir\n5. Sta-
tus \n0. Sair\n\n: "
msgLinha:         .asciiz "\n===== \n"
msgNome:          .asciiz "\nNome  : "
msgTelefone:      .asciiz "\nTelefone: "
msgTotalContatos: .asciiz "\nTotal de contatos: "
msgContatos:      .asciiz "\nContatos: "
msgPesquisarContato: .asciiz "\nPesquisar contato:"
msgRemoverContato: .asciiz "\nRemover contato:"
msgContatoNaoEncontrado: .asciiz "\nContato nao encontrado."
msgContatoRemovidoSucesso: .asciiz "\nContato removido com sucesso!"
msgI:             .asciiz "\ni: "
msgJ:             .asciiz "\nj: "
msgFim:           .asciiz "\nFim."
```

```

nome: .space 30
.text
.globl main

main:

agenda:
    move $s6, $gp                                # Dados da agenda
    addi $s6, $s6, 5000
    addi $s7, $zero, 0                          # Numero de contatos na agenda
    add $t6, $zero, $zero                       # $t5 = Nome
    add $t7, $zero, $zero                       # $t6 = Telefone

inicio:
    addi $t1, $zero, 1                          # $t1 = Adicionar contato
    addi $t2, $zero, 2                          # $t2 = Remover contato
    addi $t3, $zero, 3                          # $t3 = Pesquisar contato
    addi $t4, $zero, 4                          # $t4 = Imprimir contatos
    addi $t5, $zero, 5                          # $t5 = Status contato
    addi $t0, $zero, 0                          # $t0 = Sair

menuOpcoes:
    li $v0, 4                                    # syscall escrever string
    la $a0, msgBemVindo                         # mensagem de inicio da agenda
    syscall

    la $a0, msgLinha
    syscall

    la $a0, msgOpcoes
    syscall

    # le opcao
    li $v0, 5                                    # syscall para entrada de inteiro ($v0)
    syscall

    beq $v0, $t1, adicionarContato
    beq $v0, $t2, removerContato
    beq $v0, $t3, pesquisarContato
    beq $v0, $t4, ordenarContatos
    beq $v0, $t5, statusAgenda
    beq $v0, $t0, fim # Se nao for a opcao sair, volta ao inicio
    j inicio

lerNome:
    li $v0, 4
    la $a0, msgNome
    syscall
    addi $v0, $zero, 8
    la $a0, nome
    add $a0, $a0, $s2
    addi $s2, $s2, 10
    addi $a1, $zero, 30
    syscall
    add $t6, $a0, $zero
    jr $ra

lerNumeroTelefone:
    li $v0, 4
    la $a0, msgTelefone
    syscall
    # le numero do telefone
    li $v0, 5                                    # syscall para entrada de inteiro ($v0)
    syscall
    add $t7, $v0, $zero
    jr $ra

imprimirNome:
    li $v0, 4                                    # syscall escrever string
    add $a0, $t6, $zero                         # mensagem de inicio da agenda
    syscall
    jr $ra

imprimirTelefone:
    li $v0, 1

```

```

    add $a0, $t7, $zero
    syscall
    jr $ra

adicionarContato:
    jal lerNome
    sw $t6, 0($s6)
    addi $s6, $s6, 4

    jal lerNumeroTelefone
    sw $t7, 0($s6)
    addi $s6, $s6, 4

    addi $s7, $s7, 1
    j inicio

removerContato:
    li $v0, 4
    la $a0, msgRemoverContato
    syscall
    jal lerNumeroTelefone

    add $t0, $zero, $s6          # enderecamento do vetor
    add $t4, $zero, $s7        # numero de elementos na agenda (vai ser decrementado)
forPesquisarRemoverContato:
    beq $t4, $zero, naoAchouNoPesquisarRemoverContato
    addi $t0, $t0, -4
    lw $t5, ($t0)
    addi $t0, $t0, -4
    lw $t6, 0($t0)
    beq $t5, $t7, achouNoPesquisarRemoverContato
    addi $t4, $t4, -1
    j forPesquisarRemoverContato

achouNoPesquisarRemoverContato:
    mul $t1, $s7, -8
    add $t1, $t1, $s6

    lw $t6, 0($t1)
    lw $t7, 4($t1)

    sw $t6, 0($t0)
    addi $t0, $t0, 4
    sw $t7, 0($t0)
    addi $t0, $t0, 4

    addi $s7, $s7, -1
    li $v0, 4
    la $a0, msgContatoRemovidoSucesso
    syscall
    j inicio

naoAchouNoPesquisarRemoverContato:
    li $v0, 4
    la $a0, msgContatoNaoEncontrado
    syscall
    j inicio

ordenarContatos:
    mul $s0, $s7, -8
    add $s0, $s0, $s6
    add $t2, $zero, $zero
    forOrdenarContatos_i:
        beq $t2, $s7, fimOrdenarVetor
        addi $t2, $t2, 1
        addi $t3, $zero, 1
        add $t0, $s0, $zero
        forOrdenarContatos_j:
            beq $t3, $s7, forOrdenarContatos_i
            addi $t3, $t3, 1
            add $t1, $t0, $zero
            lw $t4, 0($t0)
            lw $t5, 4($t0)
            addi $t0, $t0, 8
            lw $t6, 0($t0)

```

# Bubble Sort

# i

# j

# indice j

```

        lw $t7, 4($t0)
        beq $t7, $t5, forOrdenarContatos_j
        slt $t8, $t7, $t5
        beq $t8, $zero, swapOrdenarContato
        j forOrdenarContatos_j

swapOrdenarContato:
    sw $t6, 0($t1)
    sw $t7, 4($t1)
    addi $t1, $t1, 8
    sw $t4, 0($t1)
    sw $t5, 4($t1)
    j forOrdenarContatos_j

fimOrdenarVetor:
    j imprimirContatos

pesquisarContato:
    li $v0, 4
    la $a0, msgPesquisarContato
    syscall
    jal lerNumeroTelefone

    add $t0, $zero, $s6           # enderecamento do vetor
    add $t4, $zero, $s7         # numero de elementos na agenda (vai ser decrementado)
    forPesquisarContatos:
        beq $t4, $zero, naoAchouNoPesquisarContato
        addi $t0, $t0, -4
        lw $t5, ($t0)
        addi $t0, $t0, -4
        lw $t6, 0($t0)
        beq $t5, $t7, achouNoPesquisarContato
        addi $t4, $t4, -1
        j forPesquisarContatos

    achouNoPesquisarContato:
        jal imprimirNome
        jal imprimirTelefone
        li $v0, 4
        la $a0, msgLinha
        syscall
        j inicio

    naoAchouNoPesquisarContato:
        li $v0, 4
        la $a0, msgContatoNaoEncontrado
        syscall
        j inicio

imprimirContatos:
    li $v0, 4
    la $a0, msgContatos
    syscall
    la $a0, msgLinha
    syscall
    add $t0, $zero, $s6           # enderecamento do vetor
    add $t4, $zero, $s7         # numero de elementos na agenda (vai ser decrementado)
    forImprimirContatos:
        beq $t4, $zero, inicio
        addi $t0, $t0, -4
        lw $t7, ($t0)
        addi $t0, $t0, -4
        lw $t6, 0($t0)
        jal imprimirNome
        jal imprimirTelefone
        li $v0, 4
        la $a0, msgLinha
        syscall
        addi $t4, $t4, -1
        j forImprimirContatos

    j inicio

statusAgenda:
    li $v0, 4
    la $a0, msgTotalContatos

```

```
syscall
```

```
# escreve o numero de contatos na agenda  
li $v0, 1  
add $a0, $zero, $s7  
syscall  
j inicio
```

```
fim:
```

```
li $v0, 4  
la $a0, msgLinha  
syscall
```

```
li $v0, 4  
la $a0, msgFim  
syscall
```

```
li $v0, 0  
syscall
```