



- 1) Elaborar um programa, em código em MIPS, que faça a leitura de um arquivo de inteiros (crie o arquivo dados1.txt com n elementos inteiros separados por um espaço em branco entre os números) e determine (implementando com funções): a) o maior valor; b) o menor valor; c) o número de elementos ímpares; d) o número de elementos pares; e) a soma dos valores; f) os valores em ordem crescente; g) os valores em ordem decrescente; h) o produto dos elementos e i) o número de caracteres do arquivo.
- 2) Elaborar um programa, em código MIPS, que realize todas as operações do exercício 1), com o arquivo dados2.txt composto por n elementos inteiros, sendo um valor inteiro por linha no arquivo.

✓ Exemplo de código para abertura de um arquivo

```
1  .data
2  Arquivo: .asciiz "dados.txt"
3  Erro: .asciiz "Arquivo não encontrado!\n"
4  .text
5  main:
6
7      li $v0, 13 # Código de abertura de arquivo
8      la $a0, Arquivo # Nome do arquivo
9      li $a1, 0 # Somente leitura
10     syscall # Tenta abrir o arquivo
11     bgez $v0, fim # if(file_descriptor >=0) goto fim
12     la $a0, Erro # else erro: carrega o endereço da string
13     li $v0, 4 # Código de impressão de string
14     syscall # Imprime o erro
15 fim: li $v0, 10 # Código para finalizar o programa
16     syscall # Finaliza o programa
17
```

✓ Exemplo de código para contagem de caracteres de um arquivo

```
1 .data
2 buffer: .asciiz " "
3 Arquivo: .asciiz "dados.txt"
4 Erro: .asciiz "Arquivo não encontrado!\n"
5
6 .text
7 main:
8     la $a0, Arquivo # Nome do arquivo
9     li $a1, 0 # Somente leitura
10    jal abertura # Retorna file descriptor no sucesso
11    move $s0, $v0, # Salva file descriptor em $s0
12    move $a0, $s0 # Parâmetro file descriptor
13    la $a1, buffer # Buffer de entrada
14    li $a2, 1 # caractere por leitura
15    jal contagem # Retorna em $v0 o número de caracteres
16    move $a0, $v0 # Move o resultado para impressão
17    li $v0, 1 # Código de impressão de inteiro
18    syscall # imprime o resultado
19    li $v0, 16 # Código para fechar o arquivo
20    li $v0, 10 # Código para finalizar o program
21    syscall # Finaliza o programa
22
23 contagem:
24     li $v0, 14 # Código de leitura de arquivo
25     syscall # Faz a leitura de 1 caractere
26     addi $t0, $t0, 1 # n++
27     bnez $v0, contagem # if (ch != EOF) goto contagem
28     subi $t0, $t0, 1 # Desconsidera EOF
29     move $v0, $t0 # Move resultado para retorno
30     jr $ra # Retorna para main
31
32 abertura:
33     li $v0, 13 # Código de abertura de arquivo
34     syscall # Tenta abrir o arquivo
35     bgez $v0, a # if(file_descriptor >= 0) goto a
36     la $a0, Erro # else erro: carrega o endereço da string
37     li $v0, 4 # Código para impressão de string
38     syscall # Imprime o erro
39     li $v0, 10 # Código para finalizar o programa
40     syscall # Finaliza o programa
41 a:    jr $ra # Retorna para a main
```

✓ Exemplo de código para leitura e soma de valores em um arquivo

```
1 .data
2 buffer: .asciiz " "
3 Arquivo: .asciiz "dados2.txt"
4 Erro: .asciiz "Arquivo não encontrado!\n"
5 .text
6 main:
7     la $a0, Arquivo # Nome do arquivo
8     li $a1, 0 # Somente leitura
9     jal abertura # Retorna file descriptor no sucesso
10    move $s0, $v0 # Salva o file descriptor em $s0
11    move $a0, $s0 # Parâmetro file descriptor
12    la $a1, buffer # Buffer de entrada
13    li $a2, 1 # 1 caractere por leitura
14    jal leitura # Retorna a soma dos números do arquivo
15    move $a0, $v0 # Move o resultado para impressão
16    li $v0, 1 # Código de impressão de inteiro
17    syscall # Imprime o resultado
18    li $v0, 16 # Código para fechar o arquivo
19    move $a0, $s0 # Parâmetro file descriptor
20    syscall # Fecha o arquivo
21    li $v0, 10 # Código para finalizar o programa
22    syscall # Finaliza o programa
23 leitura:
24    li $v0, 14 # Código de leitura de arquivo
25    syscall # Faz a leitura de 1 caractere
26    beqz $v0, f # if (EOF) termina a leitura
27    lb $t0, ($a1) # Carrega o caractere lido no buffer
28    beq $t0, 13, leitura # if (carriage return) ignora
29    beq $t0, 10, l # if (newline) goto l
30    subi $t0,$t0,48 # char para decimal
31    mul $t1, $t1, 10 # casa decimal para a esquerda
32    add $t1, $t1, $t0 # soma a unidade lida
33    j leitura # continua leitura do número
34 l:   add $t2, $t2, $t1, # Soma o número lido
35    li $t1, 0 # Zera o número
36    j leitura # Leitura do próximo número
37 f:   add $v0, $t2, $t1 # Resultado da soma em $v0
38    jr $ra
39 abertura:
40    li $v0, 13 # Código de abertura de arquivo
41    syscall # Tenta abrir o arquivo
42    bgez $v0, a # if (file_descriptor >= 0) goto a
43    la $a0, Erro # else erro: carrega o endereço da string
44    li $v0, 4 # Código de impressão de string
45    syscall # Imprime o erro
46    li $v0, 10 # Código para finalizar o programa
47    syscall # Finaliza o programa
48 a:   jr $ra # Retorna para a main
```