



**Arquitetura e Organização de Computadores - 5COP090**

**Atividades Práticas de Laboratório n. 10\_b – 16/10/2019 (3º Bimestre)**

**Peso 30 % na composição da nota do 3º Bimestre**

**Arquivos: Escrita**

**Entrega: 31/10/2019**

- 1) Um número primo é considerado gêmeo se o próximo número primo for igual a ele mais dois. Elaborar um programa em MIPS para salvar todos os números gêmeos de 1 até N em um arquivo.
- 2) Elaborar um programa, em código MIPS, que receba do usuário um arquivo texto. Crie outro arquivo texto contendo o texto do arquivo de entrada, mas com as vogais substituídas por '\*'.  
por '\*'.
- 3) O programa abaixo abre um arquivo com um vetor de inteiros, lê a i-ésima posição do vetor (i indicado pelo usuário), incrementa uma unidade e grava no mesmo ponto do arquivo. Implementar o programa em código MIPS.

```
1  #include <stdio.h>
2  int main (void)
3  {
4      int vet[4] = {5, 10, 15, 20};
5      int i, elem;
6      FILE *arq;
7      arq = fopen ("vet.dat", "w+b");
8      if (arq != NULL) {
9          fwrite(vet, sizeof (int), 4, arq);
10         scanf ("%d", &i);
11         if (i >= 0 && i < 4) {
12             // Posiciona o cursor no i-esimo elemento
13             fseek (arq, i * sizeof (int), SEEK_SET);
14             // le um inteiro
15             fread (&elem, sizeof (int), 1, arq);
16             elem = elem + 1;
17             // Volta 1 posicao (reposiciona no i-esimo)
18             fseek (arq, -sizeof (int), SEEK_CUR);
19             // grava o novo elemento
20             fwrite (&elem, sizeof (int), 1, arq);
21         }
22         rewind(arq);
23         fread(vet, sizeof(int), 4, arq); // carrega todo o arquivo no vetor
24         for (i = 0; i < 4; i++) // imprime o arquivo alterado
25             printf("%d ", vet[i]);
26         fclose (arq);
27     }
28     else
29         printf ("Nao foi possivel abrir o arquivo");
30     return 0;
31 }
```

✓ Exemplo de código para escrita em um arquivo

```
1 .data
2 Arquivo: .asciiz "teste_escrita.txt"
3 Msg: .asciiz "Escrevendo em um arquivo"
4 .text
5 main:
6     la $a0, Arquivo # Nome do arquivo
7     li $a1, 1 # Somente escrita
8     li $v0, 13 # Código de abertura de arquivo
9     syscall # Abre o arquivo (se não existir, será criado)
10    move $a0, $v0 # Parâmetro file descriptor
11    li $v0, 15 # Código de escrita em arquivo
12    la $a1, Msg # Buffer de saída
13    li $a2, 24 # 24 caracteres
14    syscall # Escreve a mensagem no arquivo
15    li $v0, 16 # Código para fechar o arquivo
16    syscall # Fecha o arquivo
17    li $v0, 10 # Código para finalizar o programa
18    syscall # Finaliza o programa
```

4) Elaborar um programa, em MIPS, que leia um arquivo que contenha as dimensões de uma matriz (linha e coluna), a quantidade de posições que serão anuladas, e as posições a serem anuladas (linha e coluna). O programa lê esse arquivo e, em seguida, produz um novo arquivo com a matriz com as dimensões dadas no arquivo lido, e todas as posições especificadas no arquivo ZERADAS e o restante recebendo o valor 1.

Ex: arquivo "matriz.txt"

3 3 2 /\*3 e 3 dimensões da matriz e 2 posições que serão anuladas\*/

1 0 /\*Posição da matriz que serão anuladas.

1 2 /\*Posição da matriz que serão anuladas.

arquivo "matriz saida.txt"

saída:

1 1 1

0 1 0

1 1 1

5) Escreva um programa, em MIPS, que receba dois arquivos textos e determine se existe pelo menos uma mesma sequência de palavras de tamanho maior ou igual a cinco (5) em ambos os arquivos.

✓ Exemplo de código para escrita de valores ímpares em um arquivo

```

1  .data
2  Buffer: .space 20
3  newline: .asciiz "\r\n"
4  Arquivo: .asciiz "impares.txt"
5  .text
6
7  main:
8      la $a0, Arquivo # Nome do arquivo
9      li $a1, 1 # Somente escrita
10     li $v0, 13 # Código de abertura de arquivo
11     syscall # Abre o arquivo (se não existir, será criado)
12     move $s0, $v0 # Salva o file descriptor
13     jal escreveImpar # Le números do usuário e escreve no arquivo
14     move $a0, $s0 # Parâmetro file descriptor
15     li $v0, 16 # Código para fechar o arquivo
16     syscall # Fecha o arquivo
17     li $v0, 10 # Código para finalizar o programa
18     syscall # Finaliza o programa
19
20 intstring:
21     div $a0, $a0, 10 # n = n / 10
22     mfhi $t0 # aux = resto
23     subi $sp, $sp, 4 # Espaço para 1 item na pilha
24     sw $t0, ($sp) # Empilha o resto da divisão
25     addi $v0, $v0, 1 # Número de caracteres++
26     bnez $a0, intstring # if(n != 0) goto intstring
27 i:    lw $t0, ($sp) # Desempilha um resto de divisão
28     addi $sp, $sp, 4 # Libera espaço de 1 item na pilha
29     add $t0, $t0, 48 # Converte a unidade (0-9) para caractere
30     sb $t0, ($a1) # Armazena no buffer de saída
31     addi $a1, $a1, 1 # Incrementa o endereço do buffer
32     addi $t1, $t1, 1 # i++
33     bne $t1, $v0, i # if(iterações != num. carac.) goto i
34     sb $zero, ($a1) # Armazena NULL no buffer de saída
35     jr $ra # Retorna para o caller
36
37 escreveImpar:
38     move $t2, $ra # aux = retorno para a main
39 e:    li $v0, 5 # Código de leitura de inteiro
40     syscall # Faz a leitura de um inteiro N
41     blez $v0, s # if(N == 0) goto s
42     andi $t0, $v0, 1 # Armazena 0 se N for par, 1 se ímpar
43     beqz $t0, e # if(N par) goto e
44     move $a0, $v0 # Parâmetro N para intstring
45     la $a1, Buffer # Buffer de saída
46     li $v0, 0 # Zera o contador
47     li $t1, 0 # Zera o contador
48     jal intstring # Converte N para string em $a1
49     move $a0, $s0 # Parâmetro file descriptor
50     la $a1, Buffer # Buffer de saída
51     move $a2, $v0 # Número de caracteres para escrita
52     li $v0, 15 # Código de escrita em arquivo
53     syscall # Escreve o número no arquivo
54     la $a1, newline # Endereço da string "\r\n"
55     li $a2, 2 # 2 caracteres
56     li $v0, 15 # Código de escrita em arquivo
57     syscall # Pula linha no arquivo
58     j e # goto e
59 s:    jr $t2 # Retorna para a main

```