

Primeira Prova

1) Esta questão tem três itens. (i) Escreva uma função em C que concatena duas cadeias de caracteres apontadas pelo primeiro e segundo argumentos para a posição apontada pelo terceiro argumento, e que retorna o número de caracteres da cadeia resultado, incluindo o terminador, conforme o protótipo abaixo. (ii) Traduza sua função para o assembly do MIPS; e (iii) escreva em assembly o trecho de código em que `strconcat()` é invocada, e mostre o leiaute do registro de ativação. [10 pontos]

```
int strconcat(char *pri, char *seg, char *res)
```

2) Esta questão tem dois itens. (i) Escreva uma função que compute o endereço de cada um dos componentes de qualquer elemento do vetor `vet[]` — a função deve calcular a distância, em bytes, de qualquer componente a partir do endereço do primeiro componente do primeiro elemento do vetor; (ii) traduza o código C na direita da figura para assembly. [10 pontos]

```
typedef struct elem {                                // &(vet[0]) = 0x8000.8000;
    int a;                                           ...
    int b;                                           acum = 0;
    int c;                                           for (i=0; i<N; i++)
    unsigned char d[4];                               acum += vet[i].c + (int)vet[i].d[2];
} elemType;                                         vet[N+1].a = acum;
int i,acum;                                         ...
elemType vet[N+1];
```

3) Para o programa de multiplicação de matrizes mostrado abaixo, e traduzido para assembly na próxima página, preencha a tabela de símbolos para a versão em assembly. [10 pontos]

```
#define TAM 8
int A[TAM][TAM] = {-1,-1,-1,-1,-1,-1,-1,-1};
int B[TAM][TAM] = {2,2,2,2,2,2,2,2};
int C[TAM][TAM];

int main (int argc, char** argv) {
    int num=0; int i,j,k;
    for (i = 0; i < TAM; i++)
        for (j = 0; j < TAM; j++) ]
            for (k = 0; k < TAM; k++)
                C[i][j] = C[i][j] + A[i][k] * B[k][j];
    return(num);
}
```

```

.data
.globl A
.align 2
.size A, 256
A: .word -1
.word -1
.word -1
.word -1
.word -1
.word -1
.word -1
.space 224
.globl B
.align 2
.size B, 256
B:
.word 2
.word 2
.word 2
.word 2
.word 2
.word 2
.word 2
.word 2
.space 224
.text
.align 2
.globl main
main:
.frame $fp,24,$31
.mask 0x40000000,-8
.fmask 0x00000000,0
subu $sp,$sp,24
sw $fp,16($sp)
move $fp,$sp
sw $4,24($fp)
sw $5,28($fp)
sw $0,0($fp)
sw $0,4($fp)
$L2:
lw $2,4($fp)
slt $2,$2,8
bne $2,$0,$L5
j $L3
$L5:
sw $0,8($fp)
$L6:
lw $2,8($fp)
slt $2,$2,8
bne $2,$0,$L9
j $L4
$L9:
sw $0,12($fp)
$L10:
lw $2,12($fp)
slt $2,$2,8
bne $2,$0,$L13
j $L8

$L13:
lw $2,4($fp)
sll $3,$2,3
lw $2,8($fp)
addu $2,$3,$2
sll $3,$2,2
la $2,C
addu $5,$3,$2
lw $2,4($fp)
sll $3,$2,3
lw $2,8($fp)
addu $2,$3,$2
sll $3,$2,2
la $2,C
addu $6,$3,$2
lw $2,4($fp)
sll $3,$2,3
lw $2,8($fp)
addu $2,$3,$2
sll $3,$2,2
la $2,A
addu $4,$3,$2
lw $2,12($fp)
sll $3,$2,3
lw $2,8($fp)
addu $2,$3,$2
sll $3,$2,2
la $2,B
addu $2,$3,$2
lw $3,0($4)
lw $2,0($2)
mult $3,$2
mflo $3
lw $2,0($6)
addu $2,$2,$3
sw $2,0($5)
lw $2,12($fp)
addu $2,$2,1
sw $2,12($fp)
j $L10
$L8:
lw $2,8($fp)
addu $2,$2,1
sw $2,8($fp)
j $L6
$L4:
lw $2,4($fp)
addu $2,$2,1
sw $2,4($fp)
j $L2
$L3:
lw $2,0($fp)
move $sp,$fp
lw $fp,16($sp)
addu $sp,$sp,24
j $31
.end main
.comm C,256

```

Segunda Prova

1) Esta questão tem três itens: (i) dê dois exemplos de diretivas do montador que *não* causam a inclusão de bits adicionais no arquivo objeto, e explique suas funções; (ii) dê dois exemplos de diretivas do montador que produzem saída no arquivo objeto, e explique suas funções; (iii) qual a função da diretiva `.align`? Esta diretiva não pode estar incluída nas suas respostas anteriores. [5 pontos]

2) Descreva as operações envolvidas na ligação de um arquivo objeto que necessita de funções de duas bibliotecas. Sua resposta deve conter uma descrição das estruturas de dados contidas nos três arquivos objeto e das estruturas de dados criadas e mantidas pelo ligador. [15 pontos]

3) Para o trecho de código mostrado abaixo, indique em quais instruções pode ser necessário aplicar relocação, e nos casos em que há relocação, de quais tipos pode ser a relocação. [10 pontos]

```

1  R: la t4, X
2    la t5, Y
3    lw s3, A(t4)
4    sw s3, B(t5)
5    jal C
6    nop
7    ori t1,r0,D
8    beq s3,t1,E
9    j S

```

4) Explique as diferenças entre ligação estática e ligação dinâmica. [10 pontos]

Exame Final

1) Esta questão tem dois itens: (i) escreva um programa em C que encontra o valor mínimo de um vetor de inteiros; (ii) traduza seu programa em C para assembly do MIPS. [20 pontos]

2) Esta questão tem dois itens: (i) descreva as operações envolvidas na ligação de um arquivo objeto que necessita de funções de duas bibliotecas; (ii) descreva as estruturas de dados contidas nos três arquivos objeto, e as estruturas de dados criadas e mantidas pelo ligador. [25 pontos]

3) Esta questão tem três itens: (i) descreva o processo de ligação estática; (ii) descreva o processo de ligação dinâmica; (iii) quais as vantagens de uma em relação à outra? [25 pontos]

4) Considere um sistema de memória virtual com as seguintes características: (a) endereço virtual de 32 bits (endereço de byte); (b) páginas com 4 Kbytes; e (c) endereço físico com 34 bits. A tabela de páginas deve conter os bits de *status* VÁLIDO, SUJO, USADO, e os bits de proteção READ-ONLY, EXECUTÁVEL.

(i) Qual seria o tamanho de uma tabela de páginas linear (vetor) nesta máquina?

(ii) mostre como implementar a tabela de páginas em *dois* níveis.

(iii) Suponha que na sua implementação do item (ii), 1/4 dos elementos da tabela de primeiro nível sejam inválidos. Calcule os tamanhos máximo e mínimo do espaço de endereçamento utilizado pelo programa? [30 pontos]