

P2 de Software Básico – 2003.1

03/07/2003

Departamento de Informática — PUC-Rio

1. (5.0 pontos) Traduza as funções a seguir para Assembly da família de processadores Intel Pentium, utilizando as regras “usuais” de alinhamento e de passagem de parâmetros e resultados em C. Comente seu código.

```
(a) struct Arv {
    int info;
    struct Arv* esq;
    struct Arv* dir;
};

int altura (Arv* a) {
    if (a == 0)
        return -1;
    else {
        int h1 = altura(a->esq);
        int h2 = altura(a->dir);
        if (h1 > h2)
            return h1 + 1;
        else
            return h2 + 1;
    }
}

(b) struct Aluno {
    unsigned int matricula;
    double p1;
    double p2;
    char turma;
};

double calc (int n, double* vals) {
    struct Aluno aluno = { 100, 6.75, 7.3, 'c' };
    double media = (aluno.p1 + aluno.p2)/2;
    double total = 1.0;
    int i;

    for (i = 0; i < n; i++)
        total *= (media - vals[i]);

    return total;
}
```

Obs.: Primeiro dê a representação em memória, byte a byte, da variável `aluno` (o valor `ascii` de `'a'` é 97 (0x61)).

2. (2.5 pontos) Implemente, em C ou Assembly, a função

```
void converte (float val, double* novo_val);
```

que converte um número em ponto flutuante de precisão simples para um número de precisão dupla. Essa função deve ser implementada manipulando apenas a representação em memória dos números em ponto flutuante.

3. (2.5 pontos) Dada a função `cria_func` corretamente implementada,

```
void* cria_func (void* f, int n, Parametro params[]);
```

e a função `f`,

```
double f(int a, char b);
```

de que forma as funções `f2` resultantes das seguintes chamadas diferem? Forneça o código Assembly da função `f2` gerada pela `cria_func` para ambos os casos.

- chamada 1:

```
params[0].tipo = INT_PAR;
params[0].e_constante = 0;
params[1].tipo = CHAR_PAR;
params[1].e_constante = 1;
params[1].valor.v_char = 'a';
f1 = cria_func(f, 2, params);

params[0].tipo = INT_PAR;
params[0].e_constante = 1;
params[0].valor.v_int = 53;
f2 = cria_func(f1, 1, params);
```

- chamada 2:

```
params[0].tipo = INT_PAR;
params[0].e_constante = 1;
params[0].valor.v_int = 53;
params[1].tipo = CHAR_PAR;
params[1].e_constante = 1;
params[1].valor.v_char = 'a';
f2 = cria_func(f, 2, params);
```

Boa Sorte!