

# P2 de Software Básico – 2005.1

## 18/06/2005

Departamento de Informática — PUC-Rio

### 1. Considere o seguinte módulo C:

```
#include <stdio.h>

extern float b;
static float c;
unsigned int a = 0xc20f0000;

float fator (float v);

static float uint2float (unsigned int v) {
    float* p = (float*) &v;
    return *p;
}

float converte (float x) {
    c = fator(b) * x;
    return c;
}

int main (void) {
    float f = uint2float(a);
    printf("%f\n", f);
    return 0;
}
```

- (a) (1.0 ponto) Liste todos os símbolos exportados (definidos nesse arquivo) e importados (esperados de outros arquivos) pelo módulo acima, ou seja, o que aparecerá como D (dados) ou T (código), na primeira categoria, e, na segunda categoria, o que aparecerá como U (indefinido), na listagem do programa nm.
- (b) (1.5 pontos) Indique o valor impresso por esse programa, quando executado. Justifique sua resposta com os cálculos necessários.

### 2. Traduza as funções a seguir para Assembly da família de processadores Intel Pentium (IA-32), utilizando as regras de passagem de parâmetros e resultados de C. Comente seu código.

- (a) (2.5 pontos)

```
struct Lista {
    int info;
    struct Lista* prox;
};
```

```

int max (struct Lista* l) {
    int v;
    if (l == 0)
        return 0;
    else {
        v = max(l->prox);
        if (v > l->info)
            return v;
        else
            return l->info;
    }
}

```

(b) (2.5 pontos)

```

double floor (double v);

double calc (int n, double* vals, double x) {
    double total = 0.0;
    int i;

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

    return floor(total/n);
}

```

3. (2.5 pontos) Ao executar o objdump com a opção -d em dado módulo objeto, obtemos a seguinte saída para uma função chamada *soma*:

```

00000054 <soma>:
54: 55          push   %ebp
55: 89 e5      mov    %esp,%ebp
57: 8b 45 0c   mov    0xc(%ebp),%eax
5a: 03 45 08   add   0x8(%ebp),%eax
5d: 89 e5      mov    %esp,%ebp
5f: 5d        pop   %ebp
60: c3        ret

```

Implemente em C uma função chamada `compila` que não receba parâmetros e retorne um ponteiro para uma função que execute exatamente o código de máquina da função `soma` visto acima. Em seguida implemente em C a função `main` que

- chame a função `compila`;
- execute a função retornada passando os inteiros 16 e 300 como parâmetros;
- imprima o resultado.

Boa Sorte!