

PUC-Rio – Software Básico – INF1018
Prova 2 – 19/11/2011

Atenção: Não esqueça de colocar o seu nome e número de matrícula na(s) folha(s) de respostas. Seja claro e objetivo. **Boa sorte!**

1. (2,5 pontos) Considere uma implementação correta da função `cria_func`, do segundo trabalho,

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

- (a) Mostre o código assembly equivalente ao código de máquina gerado por `cria_func` para o seguinte caso:

```
int f(int i, double d, int j);
func_ptr f1;

params[0].tipo = INT_PAR;
params[0].e_constante = 1;
params[0].valor.v_int = 10;
params[1].tipo = DOUBLE_PAR;
params[1].e_constante = 0;
params[2].tipo = INT_PAR;
params[2].e_constante = 0;

f1 = (func_ptr) cria_func(f, 3, params);
```

- (b) Complete a definição do tipo `func_ptr`:

```
typedef int (*func_ptr)(...);
```

Ou seja, que parâmetros devem ser passados à função gerada por `cria_func` (de que tipo, e em que ordem)?

2. (3,0 pontos) Traduza a função `foo` a seguir para assembly IA-32 do gcc/Linux (visto em sala), utilizando as regras usuais de alinhamento, passagem de parâmetros, salvamento de registradores e retorno de resultado em C/Linux. **Não esqueça de comentar o seu código!**

(Não se preocupe se você não entender o que a função faz, apenas traduza-a literalmente).

```
double boo(double x, int y);

struct X {
    float fx;
    int ix;
};

double foo(struct X *px, int n) {
    int i;
    double acc = 0.0;
    double mul = 1.0;
    for (i = 0; i < n; i++) {
        acc += boo(px->fx, px->ix);
        mul *= acc;
        px++;
    }
    return mul;
}
```

3. (2,0 pontos) Considere os arquivos abaixo:

- arquivo `arq1.h`:

```
extern double last;
void func(double x);
```
- arquivo `arq2.h`:

```
extern int s;
double transf(double d);
```
- arquivo `arq1.c`:

```
#include <math.h>
#include "arq1.h"
#include "arq2.h"

static double min = INFINITY;
double last = NAN;

void func(double x) {
    double y;
    if (s) {
        last = x;
        y = pow(x, transf(x));
        if (y < min)
            min = y;
    }
}
```

Liste todos os símbolos exportados (definidos pelo módulo) e importados (definidos por outros módulos) pelo módulo objeto `arq1.o`, ou seja, o que apareceria como D (dados) ou T (texto/código), na primeira categoria, e como U (undefined), na segunda categoria, na saída do programa `nm`.

4. (2,5 pontos) Escreva em C uma função `compara` que, **sem usar operações de ponto flutuante** (isto é, operando diretamente sobre a representação binária) compare dois valores do tipo `float`:

```
int compara(float a, float b);
```

Essa função deve retornar -1, 0 ou 1 dependendo do primeiro valor ser menor, igual ou maior que o segundo.

Para acessar a representação binária dos valores `float` você pode usar uma `union` como a que usamos no laboratório:

```
typedef union {
    float f;
    unsigned int i;
} U;
```