

PUC-Rio – Software Básico – INF1018  
Prova 2 – 26/06/2012 – Turma 3WB

1. (3,0 pontos) Traduza a função `foo` abaixo para assembly IA-32 do (o assembly visto em sala), utilizando as regras usuais de alinhamento, uso de registradores, passagem de parâmetros e resultados em C.

**Comente o seu código!**

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

```
double f (float x, float y);
int g (double x);

struct F {
    float f1;
    float f2;
};

int foo(struct F *pf, int n) {
    int i,res;
    double mul = 1.0;

    for (i = 0; i < n; i++) {
        mul *= f(pf->f1, pf->f2);
        res += g(mul);
        pf++;
    }
    return res;
}
```

2. (2,5 pontos) Considere o programa C a seguir:

```
#include <stdio.h>
void dump(void *p, int n) {
    unsigned char *p1 = (unsigned char *)p;
    while (n--) {
        printf("%p - %02x\n", p1, *p1);
        p1++;
    }
}

struct X {
    float f1;
    double d;
    short s;
    char c;
} x = {2050.75, -2.0, -5000, (0xa5 & 0x0c)};

int main(void) {
    dump((void *)&x, sizeof(x));
    return 0;
}
```

Supondo que  $x$  seja armazenado no endereço de memória `0x80500e4`, diga o que o programa irá imprimir quando executado, explicando como você chegou a esses valores. Considere que a máquina de execução é *little-endian*, e que as convenções de alinhamento são as do Linux no IA-32. Se houver posições de *padding*, indique seu conteúdo com **pp**. (ATENÇÃO: valores sem contas e explicações **NÃO** valem ponto!)

3. Dada a seguinte função na linguagem SB do segundo trabalho:

```
v0 = $0 + $1
if p0 6 6 3
v0 = v0 * p0
p0 = p0 - $1
if p0 6 6 3
ret v0
```

- (a) (1,5 pontos) Escreva o código *assembly* (**não o código de máquina**) da função que seria criada pelo seu procedimento *compila*. Use *labels* simbólicos para os desvios (*jumps*) do seu código *assembly*.
- (b) (1,0 ponto) Qual seria o resultado da chamada dessa função se ela recebesse como parâmetro o valor 5? (Você reconhece o procedimento implementado pela função SB? O que ela calcula?)

4. Considere os arquivos abaixo:

- arquivo `arq1.c`:

```
#include <stdlib.h>
extern int lmax, lmin;
void mostra_res();

char smaior[] = "é maior que maximo";
char smenor[] = "é menor que minimo";
char sinter[] = "está dentro do intervalo";

int main(int argc, char *argv[]) {
    int i = atoi(argv[1]);
    char *s;

    if (i > lmax) s = smaior;
    else if (i < lmin) s = smenor;
    else s = sinter;
    mostra_res(i, s);
    return 0;
}
```

- arquivo `arq2.c`:

```
#include <stdio.h>
double lmax = 32.0;
double lmin = 0;

void mostra_res(int i, char *s) {
    printf("\nvalor fornecido %d %s\n", i, s);
}
```

- (a) (1,0 ponto) Suponha que compilemos esses dois arquivos em separado, gerando arquivos objeto `arq1.o` e `arq2.o`. Liste todos os símbolos exportados (definidos pelo módulo) e importados (esperados de outros módulos) para cada um dos arquivos `arq1.o` e `arq2.o`, ou seja, o que apareceria como D (símbolo da área de dados exportado), T (símbolo da área de código, exportado) e U (símbolo indefinido/importado) na saída do programa `nm`.
- (b) (1,0 ponto) Suponha agora que geremos um arquivo executável chamado `programa`, composto por esses dois módulos. Se executarmos esse programa passando como argumento, na linha de comando, o valor 10, conforme abaixo:

```
./programa 10
```

qual seria a saída impressa por essa execução? **Justifique sua resposta!**