

PUC-Rio – Software Básico – INF1018
Prova 2 – Turma 3wb – 05/12/2017

1. (2,5 pontos) Considere o programa C a seguir:

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

struct X {
    double d;
    union U {
        int i;
        float f;
    } u;
} x[2];

int main(void) {
    x[0].d = 1.0;
    x[0].u.i = -260;
    x[1].d = -3.75;
    x[1].u.f = (float) x[1].d;
    dump(x, sizeof(x));
    return 0;
}
```

Sabendo que o endereço do vetor `x` na memória é `0x601060`, e supondo que a máquina de execução é *little-endian* com as convenções de alinhamento do Linux no IA-64 (vistas em sala), **mostre o que esse programa irá imprimir quando executado**. Coloque **PP** nas posições correspondentes a *padding*.

(Mostre como você chegou aos valores exibidos. Valores sem contas **NÃO** valem ponto!).

2. (2,0 pontos) Considere o arquivo `arq.c` abaixo:

```
#include <stdio.h>
#include <math.h>

extern int j;
int i = 1024;
static double f = 1.5;

int main() {
    double g = pow(f,3.0);
    printf("%d %f\n",i+j, g);
}
```

- (a) Suponha que compilamos esse arquivo com o comando `gcc -c arq.c`. Se inspecionarmos a tabela de símbolos do arquivo objeto `arq.o` com o programa `nm`, que símbolos aparecerão na saída como **D** (símbolo na área de dados, exportado), **T** (símbolo na área de código, exportado) e **U** (referência externa)?

- (b) Qual seria a modificação na saída do `nm` se substituíssemos a linha `static double f = 1.5;` por `double f = 1,5;` ?
- (c) Suponha que tentemos gerar um executável executando o comando `gcc -o arq arq1.c -lm` (a opção `-lm` é para incluir a biblioteca matemática no passo de ligação). O executável seria gerado? Se não, qual o motivo da falha?

3. Traduza as funções `foo` e `boo` abaixo para assembly IA-64, utilizando as regras usuais de alinhamento, passagem de parâmetros, salvamento de registradores e resultados em C/Linux. (Não se preocupe em entender o que as funções fazem, apenas traduza-as literalmente.)

Atenção! **Traduza o mais diretamente possível o código de C para assembly.**

- (a) (2,5 pontos)

```
#define TAM 5

void g(long x[], long y[], int n);

int foo (long v[]) {
    long local[TAM];

    g(v, local, TAM);

    if (local[0] < local[TAM-1])
        return local[0];
    else
        return local[TAM-1];
}
```

- (b) (3,0 pontos)

```
double h(double d);

struct X {
    float v1;
    double v2;
    struct X *prox;
};

double boo(struct X *px) {
    double res = 1.0;
    while (px) {
        res *= h(px->v1) + px->v2;
        px = px->prox;
    }
    return res;
}
```

Boa Prova!