

PUC-Rio – Software Básico – INF1018
Prova Final – 15/04/2016

1. (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 f;
    char c;
    double d;
    short s;
} x = {-2050.25, -9, 72.375, 1022};

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

Supondo que x seja armazenado no endereço de memória 0x905A6E0, diga o que o programa irá imprimir quando executado, deixando claro 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-64. Se houver posições de *padding*, indique seu conteúdo com **PP**. (ATENÇÃO: valores sem contas e explicações **NÃO** valem ponto!)

2. (2,5 pontos) Considere um inteiro sem sinal (unsigned int) que empacota uma mensagem com os campos indicados abaixo:



Escreva uma função em C com o seguinte protótipo:

```
unsigned int completa_msg(unsigned int msg, int ver, int seq);
```

onde

- msg: é um inteiro sem sinal, no formato indicado, com os campos P e PAYLOAD preenchidos com valores adequados
- ver: é um inteiro contendo um valor para o campo VER
- seq: é um inteiro contendo um valor para o campo SEQ

Essa função deverá retornar um novo inteiro sem sinal, no formato indicado, mantendo os valores dos campos P e PAYLOAD e contendo nos campos VER e SEQ os valores fornecidos nos parâmetros ver e seq.

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

(a) (2,5 pontos)

```
int foo(int *u, int *v, int *f) {
    int i;
    int s = 0;
    for (i = 0; i < n; i++) {
        if (f[i])
            s = s + u[i] - v[i];
    }
    return s;
}
```

(b) (2,5 pontos)

```
struct S {
    char *nome;
    double d;
    float f;
    struct S *prox;
};
int check(char s);

double boo(struct S *cd) {
    double sum = 0.0;
    int cont = 0;
    while (cd) {
        if (check(cd->nome[0])) {
            sum += cd->d * cd->f;
            cont ++;
        }
        cd = cd->prox;
    }
    return sum/(double)cont;
}
```