

PUC-Rio – Software Básico – INF1018
Prova Final – 07/07/2015

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++;
    }
}
char cadeia[] = "uma cadeia";
struct X {
    int i;
    char c;
    char *p;
    float f;
} x = {-520, 'f' & 0x3, cadeia, -65.5};
int main(void) {
    dump((void *)&x, sizeof(x));
    return 0;
}
```

Supondo que a estrutura x seja armazenada no endereço de memória $0x0804a020$, que o vetor $cadeia$ seja armazenado no endereço de memória $0x0804a014$, e sabendo que o código ASCII de ‘a’ é 97 (na base 10), 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-32. Se houver posições de *padding*, indique seu conteúdo com **pp**. (ATENÇÃO: valores sem contas **NÃO** valem ponto!)

2. (1,5 pontos) Considere um valor inteiro de 32 bits sem sinal (*unsigned*) que armazene dois valores inteiros de 16 bits **com sinal** ($v0$ e $v1$), conforme ilustrado a seguir:



Escreva uma função C que receba um inteiro sem sinal nesse formato e extraia um dos valores armazenados, retornando-o **como um valor inteiro de 32 bits com sinal**.

O protótipo da função é

```
int extrai(unsigned int u, int v);
```

A função deverá retornar o valor $v0$ (em 32 bits) caso o parâmetro v seja igual a 0 e o valor $v1$ (em 32 bits) caso o parâmetro v seja igual a 1.

3. Traduza as funções `foo` e `boo` a seguir para assembly IA-32 do gcc/Linux (visto em sala), utilizando as regras usuais de alinhamento, uso de registradores, passagem de parâmetros e retorno de resultado em C. Comente o seu código!

(Não se preocupe se você não entender o que as funções fazem, apenas traduza-as literalmente).

- (a) (3,0 pontos)

```
int bar(char *v, int n);

int boo(char a[], char b[], int n) {
    int i;
    char *po = a;
    char *pd = &b[n-1];

    for (i = 0; i < n; i++) {
        *pd = *po;
        pd++;
        po--;
    }
    return bar(b,n);
}
```

- (b) (3,0 pontos)

```
struct X {
    float val;
    struct X * next;
};

float boo (struct X * lista, float v) {
    if (lista == NULL)
        return v;
    else
        return boo (lista->next, v+(lista->val));
}
```