

PUC-Rio – Software Básico – INF1018
Prova 1 – Turma 3wa – 30/4/2015

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++;
    }
}

struct X {
    char c1;
    struct X *next;
    int i;
    short s;
};

int main (void) {
    struct X x1 = {1<<6, &x2, -4100, 'c'};
    struct X x2 = {1, NULL, -100, 100};
    dump (&x1, sizeof(struct X));
    return 0;
}
```

Supondo que *x* seja alocado na posição de memória 0xffef8408 e *x2* no endereço de memória 0xffef8430, diga o que esse programa irá imprimir quando executado. Coloque **PP** nas posições correspondentes a *padding*.

Considere que o valor do caracter ‘a’ na tabela ASCII é 97, e suponha que a máquina de execução é *little-endian* e que as convenções de alinhamento são as do Linux no IA-32 (vistas em sala).

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

2. (2 pontos) Implemente em C uma função `descreveTipos`, com o seguinte protótipo:

```
void descrevetipos (unsigned char buffer[], int numcampos, char *tipos);
```

A função `descreveTipos` recebe um vetor de chars, `desc`, que contém em cada char a descrição do tipo de 4 campos de structs, seguindo a convenção do trabalho:

01 char

10 short

11 int

Ao todo, `desc` contém a descrição de `numcampos` campos, Para simplificar, vamos supor que `numcampos` é *sempre* múltiplo de 4.

A função deve “escrever” na string `tipos` os caracteres correspondentes aos tipos encontrados em `desc`, isto é, ‘c’ para cada char, ‘s’ para short e ‘i’ para int. (Por exemplo, se o buffer recebido tiver valor 0x5b, 0x5b, a string `tipos` deve ser retornada contendo “ccsiccsi”).

3. Traduza as funções `foo` e `boo` abaixo para assembly IA-32 (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 as funções fazem, apenas traduza-as literalmente.)

Comente seu código!

- (a) (3 pontos)

```
int foo (int *v1, int n) {
    int s = 0;
    int i;
    for (i=0;i<n;i++) {
        if ((v1[i] < 0) || (v1[i] > i))
            s += v1[i];
    }
    return s;
}
```

- (b) (2,5 pontos)

```
struct x {
    int v1;
    char v2;
    struct x *next;
};

int boo (struct x *px, int val) {
    if (px==NULL)
        return val;
    else
        return boo (px->next, val + px->v1);
}
```

Boa Prova!