

PUC-Rio – Software Básico – INF1018
Prova 1 – Turma 3WA – 13/10/2016

1. (3,0 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 N {
    unsigned short s; char *v;
};
struct R {
    char a; int b; struct N *p;
};
char ss[] = "abc";
int main (void) {
    struct N n = {-1, &ss[1]};
    struct R r = {'c', 58 << 12, &n};

    dump (&n, sizeof(struct N));
    dump (&r, sizeof(struct R));

    return 0;
}
```

Descreva o que este programa irá imprimir quando executado, *justificando* os valores exibidos mostrando as contas e outras informações usadas para chegar ao resultado. Suponha que a máquina de execução seja *little-endian* com as convenções de alinhamento do Linux no IA-64 vistas em sala. Além disto considere que *v*, *n* e *r* sejam alocados nas posições de memória 0xBF0000000, 0x2A52000020 e 0x2A52000010 respectivamente e que o valor do caracter 'a' na tabela ASCII seja 97 (decimal). Coloque **PP** nas posições correspondentes a *padding*.

2. (2,0 pontos) No primeiro trabalho deste período, implementamos algumas operações básicas sobre inteiros *signed* de 128 bits representados pela estrutura a seguir:

```
typedef struct int128 {
    long high; /* parte alta do valor: bits 127 a 64 */
    long low ; /* parte baixa do valor: bits 63 a 0 */
} Int128;
```

Escreva uma função que retorne o **módulo** do valor do inteiro de 128 bits fornecido **a**. O protótipo da função é

```
void int128_mod (Int128 *res, Int128 *a);
```

3. Traduza as funções `boo` e `foo` 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 as funções fazem, apenas traduza-as literalmente.)

Comente seu código!

- (a) (2,0 pontos)

```
int boo (int a[], int n)
{
    if (n == 0)
        return 0;
    else
        return boo(a, n-1) + a[n-1];
}
```

- (b) (3,0 pontos)

```
struct no {
    int cha;
    int val[2];
    struct no* prox;
};

int foo (struct no *n) {
    int res = 0;
    while (n != NULL) {
        if (n->cha >= 0)
            res += n->val[0];
        else
            res -= n->val[1];

        n = n->prox;
    }
    return res;
}
```

Boa Prova!