

PUC-Rio – Software Básico – INF1018
Prova Final – 05/07/2018

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

```
#include <stdio.h>

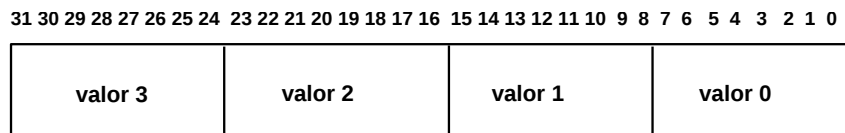
void dump (void *p, int n) {
    unsigned char *p1 = (unsigned char *) p;
    while (n--) {
        printf("%p - %02x\n", p1, *p1);
        p1++;
    }
}

struct X {
    double d;
    char c[2];
    int i;
    float f;
} x = {35.25, {'d'-1, 1 << 3}, -5, -2.0};

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

Supondo que x seja armazenado no endereço de memória 0x601040, 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*, que as convenções de alinhamento são as do Linux no IA-64 e que o valor do caractere 'a' na tabela ASCII é 97, em decimal. Se houver posições de *padding*, indique seu conteúdo com **PP**. (ATENÇÃO: valores sem contas e explicações **NÃO** valem ponto!)

2. (2,0 pontos) Suponha que armazenemos em um inteiro *unsigned* de 32 bits quatro valores inteiros de um byte cada, conforme mostrado na figura a seguir:



Escreva uma função C (ou *assembly*, se preferir) que receba um inteiro *unsigned* nesse formato e retorne um novo inteiro *unsigned*, nesse mesmo formato, onde **cada um** dos valores de um byte tenha sido multiplicado por dois. (Você pode ignorar casos de *overflow*).

O protótipo da função pedida é

```
unsigned int pack_mul2(unsigned int packed);
```

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) (2,75 pontos)

```
struct S {
    int val;
    struct S *esq;
    struct S *dir;
}

int foo (struct S *s) {
    int n = 0;
    if (s != NULL) {
        if (s->val >= 0) {
            n = foo(s->dir) + 1;
        }
    }
    return n;
}
```

(b) (2,75 pontos)

```
double f(double x, int n);

float boo(double *v, float ant, int n) {
    int i;
    double r = (double) ant;

    for (i = 0; i < n; i++) {
        r += f(v[i], i);
    }
    return (float) r;
}
```