

PUC-Rio – Software Básico – INF1018
Prova 1 – 30/9/2014

1. (3 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 no {
    struct no *next;
    short x;
    int y;
    char c;
};

int main (void) {
    struct no N0 = {NULL, 34, -1025, 'a'};
    struct no N1 = {&N0, -1031, 508, 82>>3};
    dump ((void *)&N1, sizeof(struct no)); return 0;
}
```

Considerando que N0 seja alocado na posição de memória 0xbfd98e64 e N1 na posição de memória 0xbfd98e74, diga o que esse programa irá imprimir quando executado, explicando como você chegou aos valores exibidos (ATENÇÃO: valores sem contas e explicações NÃO valem ponto!!!). 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.

2. (1,5 pontos) O caractere BOM (código U+FEFF) é um caractere especial inserido no início de um arquivo UTF-32. O BOM funciona como uma assinatura do arquivo: além de identificar o conteúdo do arquivo como UNICODE ele também define a ordem de armazenamento do arquivo (big-endian ou little-endian).

Escreva uma função que recebe como entrada um arquivo contendo um texto codificado em UTF-32 e retorna 0 se o arquivo é little-endian e 1 se o arquivo é big-endian.

```
int ordenacao (FILE* arq_entrada);
```

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. Traduza o mais diretamente possível o código de C para assembly. **Comente seu código!**

- (a) (2,5 pontos)

```
struct x {
    int key;
    int val;
    struct x *next;
};
```

```
int procura (struct x *xp, int key) {
    if (xp == NULL) return -1;
    if (xp->key == key) return xp->val;
    else return procura(xp->next, key);
}
```

(b) (3 pontos)

```
int temmenor (int *v, int n, int val, int *qual) {
    int i;
    for (i=0;i<n;i++)
        if (v[i]<val) {
            *qual = v[i];
            return 1;
        }
    return 0;
}
```

Boa Prova!