

PUC-Rio – Software Básico – INF1018  
Prova Final – 30/06/2011

**Atenção:** Não esqueça de colocar o seu nome e número de matrícula na(s) folha(s) de respostas. Seja claro e objetivo. **Boa sorte!**

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 {
    short s;
    double d;
    char c[2];
} x = {-1025, -14.25, {'c', 300}};

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

Considerando que  $x$  seja armazenado no endereço de memória 0x2A000004, mostre o que o programa irá imprimir quando executado, **explicando como você chegou a esses valores**. Valores sem contas e explicações não serão considerados!

Suponha que a máquina de execução é *little-endian*, e que as convenções de alinhamento são as do Linux no IA-32. Escreva 'LL' nos bytes usados para *padding*. O código ASCII de 'a' é 97.

2. (1,5 pontos) Escreva uma função C que recebe um *float*  $e$ , usando instruções de manipulação de bits, retorne um inteiro que representa o *valor real* do expoente desse float. O protótipo da função é mostrado a seguir:

```
int expoente (float f);
```

Lembre que o valor real do expoente é o valor de  $E$  tal que o valor codificado segundo o padrão IEEE-754 represente  $(-1)^s * M * 2^E$

3. (3,0 pontos) Escreva em assembly IA-32 uma função `write_strings` que recebe um inteiro e um *array* de *strings* e escreve cada uma dessas *strings* na saída padrão, excluindo da escrita o caracter terminador de cada *string* (`'\0'`).

O primeiro parâmetro recebido pela função especifica o número de *strings* a escrever. A função deve retornar o número total de caracteres escritos no arquivo.

Lembre-se de respeitar as regras usuais de alinhamento, passagem de parâmetros, valor de retorno e salvamento de registradores para C/Linux. O protótipo da função é mostrado a seguir:

```
int write_strings (int n, char *strings[]);
```

A função deve chamar diretamente o serviço `write` do Sistema Operacional para fazer a escrita. Esse serviço recebe três parâmetros:

- o descritor do arquivo (um inteiro)
- o endereço do buffer a ser escrito
- o número de bytes a escrever (um inteiro)

Para construir a chamada ao sistema, lembre-se que o número da chamada é colocado em `%eax` e os parâmetros do serviço nos demais registradores. O número da chamada `write` é 4. O descritor da saída padrão é o inteiro `1`.

4. (3,0 pontos) Traduza a função `foo` a seguir para assembly IA-32, utilizando as regras usuais de alinhamento, passagem de parâmetros, retorno e salvamento de registradores para C/Linux.

**Comente o seu código!**

```
double f1 (double d);
double f2 (float f);

double foo (int i, int j) {
    return f1((double)i) + f2((float)j);
}
```