

**PUC-Rio – Software Básico – INF1018**  
**Prova 2 – 24/06/2010**

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 um {
    float f;
    double d;
    char c;
};
struct um a = {52.625, 0.25/4, -4};
int main (void) {
    dump (&a, sizeof(a));
    return 0;
}
```

Considerando que `a` seja alocado na posição de memória `0xbffffeb4`, diga o que esse programa irá imprimir quando executado, explicando como você chegou aos valores exibidos (*mostre suas contas!!*). Suponha que a máquina de execução é IA32-Linux, ou seja, a representação de dados é a vista no curso.

2. (3,0 pontos) Traduza a função `foo` abaixo para assembly IA-32 (o assembly visto em sala), utilizando as regras usuais de alinhamento, passagem de parâmetros e resultados em C. Comente seu código.

(Não se preocupe se você não souber o que a função faz, apenas traduza-a literalmente! Também não se preocupe com o que as funções `f` e `g` fazem!)

```
double f (double x);
double g (double *x, double *y, int n);

double foo (double *l, double *k, int n) {
    int i;
    double local[20];
    for (i=0;i<n;i++)
        local[i] = f(l[i]) + k[i];
    return g(local, k, n);
}
```

3. (2,5 pontos) Considere o programa formado pelos dois arquivos abaixo:

- arquivo `t1.c`

```
#include <stdio.h>

static float x = 1.0;
short s[2] = {2,3};
```

```

int main (void) {
    int i;

    i = boba1();
    printf("%d\n", i);
    for (i = 0; i < sizeof(s); i++)
        printf("%d\n", a[s]);
    return 0;
}

```

- arquivo t2.c

```

#include <math.h>      extern int a[8];
int foo () {
    return 1;
}
float boba1 (void) {
    int i;
    a[0] = 0;
    i = foo();
    return 0.0;
}

```

- Liste todos os símbolos exportados (definidos pelo módulo) e importados (esperados de outros módulos) para cada um dos arquivos `t1.o` e `t2.o`, ou seja, o que apareceria como D (dados exportados) ou T (texto, ou código exportado), na primeira categoria, e, na segunda categoria, o que apareceria como U (undefined), na listagem do `nm`.
  - Ao executar o programa gerado a partir desses dois módulos, qual seria a saída do programa, isto é, o que seria exibido para o usuário? **Justifique sua resposta.**
  - O que acontece se no 2o arquivo definirmos a função `boba1` como `static`? Haverá erro na geração do executável? Que tipo de erro (compilação, ligação, outro)?
4. (2 pontos) Escreva, em assembly, uma função que elimine um diretório (pasta) com o nome dado. A função deve ter a seguinte assinatura:

```
int removeDir (char *nomeArq);
```

Sua função deve fazer uma chamada ao sistema operacional (O número da chamada `rmdir` é 40).