PUC-Rio – Software Básico – INF1612
Prova 2 – 22/11/05

1. Traduza as funções `add` e `zee` abaixo para assembly IA-32 (o assembly visto em sala), utilizando as regras usuais de alinhamento, passagem de parâmetros e retorno em C/Linux. Comente seu código.

   (Não se preocupe se você não souber o que as funções fazem, apenas traduza-as literalmente!)

   (a) (3,0 pontos)

   ```
   struct X {
     int val;
     struct X *l;
     struct X *r;
   };
   int add (struct X *x) {
     if (x==0)
       return 0;
     else
       return x->val + add(x->l) + add(x->r);
   }
   ```

   (b) (3,0 pontos)

   ```
   double f (double d);
   double g (float h);
   double zee (int i) {
     return f(i) + g(i);
   }
   ```

2. (1,5 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;
     char c;
     double d;
   };
   struct um a =  {-3.75, -4, 313.0};
   int main (void) {
     dump (&a, sizeof(a));
     return 0;
   }
   ```

   Considerando que `a` seja alocado na posição de memória `0xbffffee0`, 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 Pentium-Linux, ou seja, a representação de dados é a vista no curso.

3. (2,5 pontos) Escreva uma função `mywrite` em assembly IA-32 (convenções Linux) que receba uma string (cadeia de caracteres terminada em `'\0'`), e que escreva a string na *saída padrão* (que normalmente é a tela):

```
int mywrite (char *s);
```

Essa função não deve chamar nenhuma função da biblioteca C para fazer a escrita. Ela deve chamar diretamente o serviço `write` do Sistema Operacional. Segue uma descrição resumida e simplificada da página `man` da função `write` da biblioteca C:

```
     int
     write(int d, const void *buf, int nbytes);
DESCRIPTION
     Write() attempts to write nbytes of data to the object referenced by the
     descriptor d from the buffer pointed to by buf.
RETURN VALUES
     Upon successful completion the number of bytes which were written is
     returned.
```

Para construir a chamada ao sistema, lembre-se que o número da chamada é colocado em `%eax` e os argumentos, na ordem em que aparecem no protótipo, nos registradores `%ebx`, `%ecx`, `%edx`, `%esi` e `%edi` (quantos forem necessários). O número da chamada `write` é 4. O descritor correspondente à saída padrão é o valor inteiro 1. A interrupção que aciona o sistema operacional é a `0x80`.