

PUC-Rio — Software Básico — INF1612
Prova Final — 01/12/05

1. Traduza as funções `zee` e `add` 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.

(a) (2,5 pontos)

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

(b) (2,5 pontos)

```
int add (int *a, int *b, int n) {
    int i;
    int s = 0;
    for (i = b[n]; i < b[n+1]; i++)
        s = s + a[i];
    return s;
}
```

2. (2,0 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;
    int i;
};
struct um a = {31.5, -100, -1000.0, -1000};
int main (void) {
    dump (&a, sizeof(a));
    return 0;
}
```

Considerando que `a` seja alocado na posição de memória `0xbffff000`, 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 (inteiros de 32 bits, little-endian), ou seja, a representação de dados é a vista no curso.

3. (1,5) Escreva uma função que receba um float `f` (padrão IEEE 754) e retorne o float `f*2`. Essa função deve ser escrita em assembly IA-32 utilizando as regras usuais de alinhamento e passagem de parâmetros em C/Linux, mas *não deverá utilizar nenhuma instrução do co-processador de ponto flutuante*. O resultado deverá ser retornado em `%eax`. (Dica: se você não souber como

calcular $f*2$, escreva a representação em float dos números 3 e 6 e reflita sobre o que muda de um para o outro.)

Sua função pode ignorar os casos especiais da representação IEEE 754 (infinito, NaN, números não normalizados), mas deverá tratar corretamente o caso especial de f ser zero.

4. (1,5 pontos) Considere um inteiro contendo os campos indicados abaixo:

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																															
comp. total																TOS								IHL				versão			

Escreva uma função que receba um inteiro nesse formato e retorne a soma do valor do campo `comp. total` com o valor do campo `IHL`. Essa função deve ser escrita em assembly IA-32 utilizando as regras usuais de alinhamento, passagem de parâmetros e retorno em C/Linux.