

Segunda Prova de Software Básico (inf1612)

Duração: 1 hora e 45 minutos

19 de junho de 2007

Atenção: Seja preciso e objetivo em suas respostas. Clareza conta ponto. Não esqueça de colocar o seu nome e número de matrícula na folha de respostas.

Boa sorte!

Questão 1 (*3 pontos*)

Escreva o código assembly IA32 para o seguinte procedimento `soma_map`, que recebe um ponteiro de função (`map`), aplica essa função sobre os $n-1$ últimos elementos do segundo argumento (um vetor `p[n]`), e devolve em `p[0]` a soma dos resultados.

```
typedef double (*map) (double);

void soma_map (map f, float p[], int n) {
    int i;
    float acc = 0.0;
    for (i=1; i<=n; i++) {
        acc = acc + f(p[i]);
    }
    p[0] = acc;
}
```

Questão 2 (*3 pontos*)

Descreva todos os passos para converter 4 bytes em representação hexadecimal para a representação em ponto flutuante (`float`), e vice-versa. Como exemplo, utilize os valores `0x00A000A0` e `656.8125`.

Questão 3 (*2 pontos*)

Assumindo que `x` seja uma variável inteira, `d` uma variável `double` e `f` um `float`, e que nem `d` nem `f` sejam NAN (not a number), diga se as seguintes expressões em C são sempre verdadeiras ou falsas: **Justifique!**

- `f == - (-f)`
- `4/5 == 4/5.0`
- `(d+f)-d == f`
- `d = (double) (float) d`

Questão 4 (2 pontos)

Liste todos os símbolos exportados (definidos neste arquivo) e importados (esperados de outros arquivos) pelo módulo abaixo, ou seja, quais nomes aparecerão como D (dados), ou T (texto), ou como U (undefined) quando o programa `nm` é executado para esse módulo?

Além disso, diga o que vai ocorrer se um outro módulo amarrado a este tiver uma chamada para esta função `h`.

```
extern float b;
static int c;
int a = 5;

int g (int *x, float z);

static int h(int x) {
    float f = 0.0;
    int y;
    return g(&x, f);
}
```