

PUC-Rio – Software Básico - INF1018

Prova Final – Turmas 3WA e 3WB – 12/12/2023

1) (3,0 pontos) Considere o seguinte programa em C:

```
#include <stdio.h>

union uniao {
    float a;
    int b;
    unsigned int c;
};

void dump(void *p, int n) {
    unsigned char *p1;
    p1 = p;
    while(n--) {
        printf("p: 0x%02X\n", p1, *p1);
        p1++;
    }
}

int main(void) {
    union uniao todos;
    // essa linha foi removida do programa para este enunciado
    dump(&todos, sizeof todos);
    printf("Complete isso => %f, %d, %u\n", todos.a, todos.b, todos.c);
    return 0;
}
```

Considerando que a saída do comando `dump(&todos, sizeof todos)` é a seguinte:

```
0x7ffee0242704: 0x00
0x7ffee0242705: 0x00
0x7ffee0242706: 0xFD
0x7ffee0242707: 0xC1
```

Forneça qual será a saída do comando `printf` acima (logo após o `dump`). Justifique sua resposta.

x	2^x	x	2^x	x	2^x	x	2^x
0	1	8	256	16	65536	24	16777216
1	2	9	512	17	131072	25	33554432
2	4	10	1024	18	262144	26	67108864
3	8	11	2048	19	524288	27	134217728
4	16	12	4096	20	1048576	28	268435456
5	32	13	8192	21	2097152	29	536870912
6	64	14	16384	22	4194304	30	1073741824
7	128	15	32768	23	8388608	31	2147483648

2) (2,0 pontos) Implemente em C ou assembly a função

```
void converte (float val, double* novo_val);
```

que converte um número ponto flutuante de precisão simples para um número ponto flutuante de precisão dupla. Essa função deve ser implementada manipulando apenas a representação em memória dos números correspondentes, ou seja, usando apenas operações com números inteiros, sem usar operações de ponto flutuante.

3) Traduza as funções `boo` e `diff` abaixo para assembly IA-64, utilizando as regras usuais de alinhamento, passagem de parâmetros, salvamento de registradores e resultados em C/linux. Traduza o mais diretamente possível o código de C para assembly.

a) (2,5 pontos)

```
int corta(int v[], int i);

long int boo (int *a, int n) {
    long int acc = 0;

    while (n--)
    {
        acc += corta(a, n);
        a++;
    }

    return acc;
}
```

b) (2,5 pontos)

```
double diff(double a, double b, int n)
{
    double s[2] = {0.0, 0.0};

    while (n--)
    {
        s[0] += a;
        s[1] += b;
    }

    return (s[0] - s[1]);
}
```

Dica: NÃO existe instrução que mova constante para registrador de ponto flutuante.

Boa Prova!