

# PUC-Rio – Software Básico – INF1018

## Prova Final – Turmas 3WB e 3WC

Q1	
Q2	
Q3	

Nome: \_\_\_\_\_

1) (3,0 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 S {
    int a;
    float *b;
    char c;
    double d;
    float e;
} s = {-62, &s.e, 62, 62.625, 62.625};

int main(void) {
    dump(&s, sizeof(struct S));
    return 0;
}
```

Mostre o que este programa irá imprimir quando executado, justificando os valores exibidos mostrando as contas e outras informações usadas para chegar ao resultado. Suponha que a máquina de execução seja *little-endian* com as convenções de alinhamento do Linux no IA-64 vistas em sala. Além disto, considere que *s* seja alocado na posição de memória **0x61b79fc2b020**. Coloque **PP** nas posições correspondentes a *padding*.

2) (2,0 pontos) Considere o módulo escrito em linguagem C a seguir:

```
#include <stdio.h>

extern int a;
extern int c;

void foo(void);

int a = 1024;
static int b = 512;

void bar(void) {
    return;
}

int main (void) {
    int d = 2;
    foo();
    bar();
    printf("%d %d %d %d\n", a, b, c, d);
    return 0;
}
```

Liste quais símbolos do módulo objeto apareceriam na saída do programa **nm** como visto em sala de aula e no laboratório de *linker* como (informe o tipo e o símbolo):

- a) D símbolo na área de dados, exportado
- b) d símbolo na área de dados, local
- c) T símbolo na área de código, exportado
- d) t símbolo na área de código, local
- e) U símbolo indefinido, referência externa

3) (5,0 pontos) Traduza o mais fielmente possível a função a seguir para *assembly* IA-64 (o *assembly* visto em sala), 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*.

**Comente seu código!**

```
struct exam {
    float val;
    int tipo;
};

double inv(double val);

float boo(double d, float bias) {
    int i;
    struct exam temp[5];

    for (i=0; i<5; i++) {
        temp[i].val = inv(d);
        temp[i].tipo = i;
    }

    return temp[0].val + bias;
}
```

Boa Prova!