

PUC-Rio – Software Básico – INF1018

Prova 1 – Turma 3WB

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 {
struct Estrutura {
    short s;
    char c;
    char *p;
    int i;
};
int main(void) {
    struct Estrutura variavel;
    variavel.c = 'H'
    variavel.p = &variavel.c;
    variavel.i = -29;
    variavel.s = 34;
    dump(&s, sizeof(struct Estrutura));
    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 **0x7ffc0131d9a0** e que o valor do caractere 'A' na tabela ASCII seja 65 (decimal). Coloque **PP** nas posições correspondentes a *padding*.

- 2) (2,0 pontos) Implemente uma função em C chamada **contaBitsZero** que recebe um inteiro de quatro bytes sem sinal chamado **numero** e retorna a quantidade de bits em "0" em sequência a partir do bit menos significativo. Considere que o bit menos significativo é o bit de posição "0". Use o seguinte protótipo:

```
int contaBitsZero(unsigned int numero);
```

- 3) (5,0 pontos) Traduza 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!**

```
int eh_par(int x);
int boo(int *v, int n) {
    int i, pares = 0;
    for (i = 0; i < n; i++) {
        if (eh_par(v[i]))
            pares++;
    }
    return pares;
}
```

Boa Prova!