

PUC-Rio – Software Básico – INF1018
Prova 1 – Turmas 3WA, 3WB e 3WC – 08/10/2024

Nome:
Turma:

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 {
    char c;
    short s;
    short *p;
    int i;
};
int main(void) {
    struct S s;
    s.c = 'e';
    s.s = 12;
    s.p = &s.s;
    s.i = -20;
    dump(&s, sizeof(struct S));
    return 0;
}
```

Descreva 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 0x7ffc54b9fd90 e que o valor do carácter 'a' na tabela ASCII seja 97 (decimal). Coloque **PP** nas posições correspondentes a *padding*.

2. (2,0 pontos) Implemente uma função em C chamada `configuraBits` que recebe um inteiro de um byte sem sinal chamado `byteConf` e mais dois inteiros de 4 bytes `bitInicial` e `bitFinal`. Estes dois últimos parâmetros contêm valores entre 0 e 7 representando o bit inicial e o bit final que deve ser ligado (colocado em 1) da variável `byteConf`. Os valores dos outros bits devem ser mantidos. A função deverá retornar o 1º parâmetro modificado conforme descrito. Use o seguinte protótipo:

```
unsigned char configuraBits(unsigned char byteConf, int bitInicial, int bitFinal);
```

Por exemplo, se a entrada `byteConf` for igual a 0x80 e os parâmetros `bitInicial` e `bitFinal` forem iguais a 1 e 3 respectivamente, a saída deveria ser 0x8E.

3. (5,0 pontos) Traduza a função `foo` abaixo 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 passa_alta(short v, int limite);

int foo(short *v, int n, long filtro) {
    int i, soma = 0;
    for(i=0; i<n; i++) {
        if(v[i] > filtro) {
            soma += passa_alta(v[i], 3);
        }
    }
    return soma;
}
```

Boa Prova!