

PUC-Rio – Software Básico – INF1018
Prova 2 – Turma 3WA – 18/06/2024

1. (3,0 pontos)

```
#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 X {
    int i;
    float f;
    double d;
    char c;
} x = {-0xF, 0.625, -2.375, 17};

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

Supondo que x seja armazenado no endereço de memória 0x5607ea64b010, diga o que o programa irá imprimir quando executado, deixando claro como você chegou a esses valores, mostrando as contas e outras informações usadas. Considere que a máquina de execução é *little-endian* e que as convenções de alinhamento são as do Linux no IA-64. Se houver posições de *padding*, indique seu conteúdo com **PP**. (ATENÇÃO: valores sem contas e explicações **NÃO** valem ponto!)

2. (2,5 ponto) Considere o seguinte código prog1.c:

```
#include <stdio.h>
extern int i;

static void sub () {
    i--;
    printf ("i = %d\n", i);
}

void calcula () {
    sub ();
}
```

e o código de prog2.c seguinte:

```
#include <stdio.h>
int i = 12;

int sub ();
void calcula ();

int main (void) {
```

```

printf("i = %08x\n", i);
sub ();
calcula ();
return 0;
}

```

- (a) Considere a compilação em separado dos dois códigos acima com os comandos `gcc -Wall -c prog1.c` e `gcc -Wall -c prog2.c`. Ao examinarmos o módulo objeto gerado com o comando `nm`, liste para cada módulo quais símbolos aparecem como **T** (símbolo na área de código exportado), **t** (símbolo na área de código local), **D** (símbolo na área de dados exportado), **d** (símbolo na área de dados local), e **U** (símbolo indefinido).
- (b) Diga o que acontece ao compilarmos os dois módulos conjuntamente com o comando seguinte
`gcc -Wall -o prog prog1.c prog2.c` Justifique sua resposta.
3. Traduza as funções `foo` e `boo` a seguir para assembly IA-64, utilizando as regras usuais de alinhamento, passagem de parâmetros, salvamento de registradores e resultados em C/Linux.

- (a) (2,0 pontos)

```

void foo (double *vd, int n) {

    while (n--) {
        *vd = 0.0;
        vd++;
    }
}

```

- (b) (2,5 pontos)

```

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!