PUC-Rio – Software Básio – INF1018

Prova 2 – Turma 3WA – 30/11/2023

Questão 1 (3 pontos)

Considere o seguinte código em linguagem C:

```
#include <stdio.h>

struct S {
      char c;
      double d;
      float f;
} s = {'K', 63.500, -1.8125};

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

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

a) Supondo que x seja armazenado no endereço de memória 0x50A22010, crie uma tabela mostrando como a variável x será armazenada na memória, incluindo os endereços de cada BYTE de x e seus respectivos valores em hexadecimal.
b) Mostre EXATAMENTE o que o programa irá imprimir quando executado, deixando claro como você chegou a esses valores. Considere que a máquina de execução é *little-endian*, e que as convenções de alinhamento são as do Linux no IA-64 e que o formato de ponto flutuante é IEEE-754. Se houver posiçõoes de padding, indique seu conteúdo com PP.

ATENÇÃO: valores sem contas e explicações NÃO valem ponto!

Dica: o caractere A na tabela ASCII ocupa a posição 0x41.

Questão 2 (3 pontos)

Considere o seguinte código em linguagem C composto por 2 módulos:

header.h

```
extern int minhaVariavel;

void funcaoModulo1();
void funcaoModulo2();
```

arquivo.c

```
#include <stdio.h>
#include "header.h"

int minhaVariavel = 42;

void funcaoModulo1() {
    printf("Função do Módulo 1\n");
    printf("Valor da variável: %d\n", minhaVariavel);

    funcaoModulo2();
}
```

main.c

```
#include <stdio.h>
#include "header.h"

void funcaoModulo2() {
    printf("Função do Módulo 2\n");
    printf("Valor da variável: %d\n", minhaVariavel);

    funcaoModulo1();
}

int main() {
    funcaoModulo1();
    return 0;
}
```

Considere a compilação em separado dos dois códigos acima com os comandos:

- gcc -Wall -c -o arquivo arquivo.c
- gcc -Wall -c -o main main.c

Ao examinarmos os módulos objetos gerados, 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).

Questão 3 (4 pontos)

Traduza o código abaixo o mais fielmente possível para assembly IA-64 utilizando as regras usuais de alinhamento, passagem de parâmetros, salvamento de registradores e resultados em C/Linux.

Dica: antes de começar a codificar, desenhe como a variável reg será armazenada na memória.

```
#define TAM 5

struct S {
  char c;
  double d;
};

double corta (float f);
int indexa(int qtd);

double boba (float val, double *x, int n) {
  int i;
  struct S reg[TAM];

  while(n) {
    i = indexa(TAM);
    reg[i].c = i;
    reg[i].d = corta(val);
    n--
  }
  return reg[0].d;
}
```

Boa Prova!