

Interface Hardware-Software (IF-817)
2º Exercício Escolar

Adriano Sarmiento
Centro de Informática-UFPE

1ª Questão (4,0p)

- (a) Complete o código assembly abaixo utilizando instruções de ponto flutuante para implementar o cálculo do volume de um cone ($\text{PI} * \text{altura} * \text{raio}^2/3$)

```
SECTION .data
    raio    dd    4.0
    altura  dd    3.0
    volume  dd    0
SECTION .text
global main
main:
    ...
```

- (b) Descreva como fica a pilha de registradores da FPU a cada instrução executada que altere os valores da pilha

2ª Questão (4,0p)

- (a) Escreva um programa em C que leia do usuário 2 valores reais que são o raio e a altura de um cone e em seguida chame a função **void cone (float* vol, float r, float h)** que tem como parâmetro o raio e a altura e um ponteiro para um número real que é o endereço de uma variável que guardará o valor do volume do cone.
- (b) A função **cone** deve ser implementada em assembly. Modifique o código assembly da questão 1 para que o código C possa de fato utilizar esta função.

3ª Questão (2,0p)

O código abaixo foi feito para rodar em processadores multicore. Nele se utiliza OpenMP. No espaço destacado onde se deve acrescentar o código, o objetivo é inicializar cada posição do vetor com um número da sequência de Fibonacci, ou seja $\text{vetor}[0] = 1$, $\text{vetor}[1] = 1$, $\text{vetor}[2] = 2$, etc.

- (a) Se o código para inicialização do vetor for colocado no lugar indicado sem usar nenhuma diretiva ou função de OpenMP, o que acontecerá? Justifique a resposta
- (b) Acrescente o código no local indicado de modo que o código execute corretamente (pode também acrescentar cláusulas nas diretivas **parallel** e **for** se achar necessário). Explique porque o código funcionará corretamente.

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <omp.h>
#define N 10
int main() {
    double vetor[N];
    int i;

    #pragma omp parallel
    {
        /** Acrescente código para Inicializar vetor com sequencia
            de Fibonacci */

        #pragma omp for
        for (i = 0; i < N; i++){
            vetor[i] = pow(vetor[i],i);
        }

        printf("\nVetor eh igual \n");
        for (i = 0; i < N; i++) {
            printf("%f ",vetor[i]);
        }
        printf("\n");
        return 0;
    }
}
```