

<b>UNIFEI</b>	Universidade Federal de Itajubá Instituto de Engenharia de Sistemas e Tecnologias da Informação-IESTI
6º Laboratório de ECO013 – Prof. João Paulo R.R. Leite e Prof. Rodrigo de Paula Rodrigues Baseado no material desenvolvido pela Profª Katia Cristina Lage dos Santos	

O objetivo do laboratório é implementar um simulador de escalonador de processos da CPU usando filas dinâmicas encadeadas. Serão realizadas simulações com processos em três níveis de prioridade: baixo, médio e alto.

Utilize o Arquivo “**ecoo13-labo6.c**” disponibilizado em conjunto com o guia como ponto de partida para seu código.

A variável **filaDeProcessos** é definida como global para todas as funções desse programa, sendo que representa 3 filas de processos. Por convenção: filaDeProcessos[0] armazena processos com prioridade entre 0 e 3; filaDeProcessos[1] armazena processos com prioridade entre 4 e 6; filaDeProcessos[2] armazena processos com prioridade entre 7 e 9. Cada elemento nas filas representa um único processo na CPU do tipo noProcesso.

Abaixo são dadas algumas definições que serão usadas no programa.

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

#define NUM_FILAS 3
#define TAM_NOME 50

typedef enum {false, true} bool;

//estrutura de um processo
typedef struct no
{
    float tamanho;
    char nome[TAM_NOME];
    int prioridade;
    struct no* prox;
} noProcesso;

//fila de processos a serem executados na CPU
noProcesso * filaDeProcessos[NUM_FILAS];

// vetor que indica a existência de pelo menos um processo dessa prioridade
bool prioridadesProcessos[10];
```

a) (10 pontos) Faça uma função chamada **inicializaFila** que não recebe parâmetros e não retorna nada. Essa função deve inicializar cada elemento do vetor chamado filaDeProcessos com o valor NULL.

a) (10 pontos) Faça uma função chamada **inicializaVetorPrioridades** que não recebe parâmetros e não retorna nada. Essa função deve inicializar cada elemento do vetor chamado prioridadesProcessos com o valor *false*.

b) (20 pontos) Complete a função **insereFila**insereFila(float tamanho, char nome[TAM\_NOME], int prioridade) que insere na fila correspondente ao valor da prioridade do processo (variável prioridade) um elemento do tipo ponteiro para *noProcesso* com os dados de *tamanho*, *nome* e *prioridade* do processo correspondente.

c) (10 pontos) Complete a função **void imprimeDadosListasDeProcessos()** que percorre todas as filas, imprimindo os dados de cada um dos processos. Esta função ainda contabiliza o número de processos e realiza somatório do tamanho dos processos de cada fila.

d) (5 pontos) Inclua a definição do método *main* abaixo:

```

int main(int argc, char ** argv)
{
    inicializaFila();
    inicializaVetorPrioridades();

    insereFila(1, "BrOfficeCalc", 3);
    insereFila(2.3, "CodeBlocks", 9);
    insereFila(0.5, "Calculadora", 1);
    insereFila(6.4, "Firefox", 7);
    insereFila(3.1, "PhpMyAdmin", 2);
    insereFila(2.9, "Kile", 4);
    insereFila(4.5, "Gimp", 9);
    insereFila(5.5, "BrOfficeWriter", 6);
    insereFila(5.8, "GCC", 2);

    imprimeDadosListasDeProcessos();

    return 0;
}

```

d) (15 pontos) Complete a função **bool retiraPrimeiroDaFila** abaixo que deve retirar o primeiro elemento da fila cujo índice foi recebido como parâmetro. Ela retorna *true* se a fila não estava vazia e *false* caso contrário.

Inclua no método *main* uma chamada à função **retiraPrimeiroDaFila** enviando o valor 2 como parâmetro. Em seguida, faça uma chamada à função que imprime os dados das listas para exibir a retirada com sucesso do primeiro processo da fila 2.

e) (15 pontos) Complete a função **bool retiraMaiorPrioridadeDaFila(int indiceFila)** que retorna *true* se foi possível retirar da fila o processo de maior prioridade. Retorna *false* caso contrário. A fila que deve ser analisada contém o índice recebido como parâmetro pela função.

f) (15 pontos) Adicione ao final do main chamadas consecutivas às funções **retiraMaiorPrioridadeDaFila** e **imprimeDadosListasDeProcessos** enviando à primeira como parâmetro o índice 1 até que a fila esteja vazia. Coloque o tratamento no método *main* para imprimir a sentença “A fila 1 está vazia” quando isso ocorrer.