

UNIVERSIDADE FEDERAL DE MINAS GERAIS  
INSTITUTO DE CIÊNCIAS EXATAS  
DEPARTAMENTO DE CIÊNCIA DA COMPUTAÇÃO

ALGORITMOS E ESTRUTURA DE DADOS 2  
TRABALHO PRÁTICO 1

Aluno: Breno Muniz de Castro Fonseca

Belo Horizonte

Maio 2017

## 1) Introdução

Foi pedido que se criasse um programa que calculasse a média dos usuários na cantina do ICEX desde o momento que entrassem na fila do caixa até o momento em que colocassem o último tipo de comida no prato. Para cumprir todo o processo era necessário comprar a ficha no caixa, pegar a bandeja numa pilha de bandejas e depois colocar 4 tipos de comida no prato. (Arroz, feijão, guarnição e salada). A solução baseia-se em criar uma fila para armazenar as pessoas que estão esperando para comprar uma ficha no caixa, depois outra fila para as pessoas que estão esperando para pegar a bandeja, de forma que a pessoa gasta 1 minuto para andar uma posição nas filas, mais um minuto para pegar a ficha no caixa e mais um minuto pra pegar a bandeja e 1 minuto pra pegar cada um dos 4 tipos de comida. A solução proposta é de criar 2 estruturas de dados fila para armazenar as pessoas que estão na fila do caixa e da bandeja e uma estrutura de dados pilha para armazenar a pilha de bandejas. Foi criado um vetor de pessoas para armazenar uma pessoa em cada posição, de forma que a medida que essa pessoa fosse percorrendo a cantina o seu tempo seria aumentado em 1.

## 2) Desenvolvimento

```
main.c
#include <stdio.h>
#include <stdlib.h>
#include "bandejao.h"

int main()
{
    int i, t, soma, vetor_pessoa[481];
    float media_tempo;

    soma = 0;

    Fila_Caixa *fila_do_caixa;
    fila_do_caixa = (Fila_Caixa*)malloc(sizeof(Fila_Caixa));

    Fila_Bandeja *fila_da_bandeja;
    fila_da_bandeja = (Fila_Bandeja*)malloc(sizeof(Fila_Bandeja));

    Faz_Fila_Caixa_Vazia(fila_do_caixa);
    Faz_Fila_Bandeja_Vazia(fila_da_bandeja);

    for(i = 0; i < 481; i++){
        vetor_pessoa[i] = 0;
    }

    for(t = 1; t <= 480; t++){
        if(Testa_Fila_Caixa_Vazia(*fila_do_caixa) &&
        Testa_Fila_Bandeja_Vazia(*fila_da_bandeja)){
```

```

        vetor_pessoa[i] = vetor_pessoa[i] + 2;

    }
    if(!Testa_Fila_Caixa_Vazia(*fila_do_caixa) &&
Testa_Fila_Bandeja_Vazia(*fila_da_bandeja)){

        Enfileira_Caixa(fila_do_caixa, &vetor_pessoa[i]);
        vetor_pessoa[i]++;

    }

    if(!Testa_Fila_Caixa_Vazia(*fila_do_caixa) &&
!Testa_Fila_Bandeja_Vazia(*fila_da_bandeja)){

        Enfileira_Caixa(fila_do_caixa, &vetor_pessoa[i]);
        vetor_pessoa[i]++;
        Desenfileira_Caixa(fila_do_caixa, &vetor_pessoa[i]);
        Enfileira_Bandeja(fila_da_bandeja, &vetor_pessoa[i]);
        vetor_pessoa[i]++;
        Desenfileira_Bandeja(fila_da_bandeja, &vetor_pessoa[i]);
    }
    vetor_pessoa[i] = vetor_pessoa[i] + 4;

    soma = soma + vetor_pessoa[i];

}

media_tempo = (float)soma/480;

printf("O tempo medio na fila de espera eh de: %f minutos", media_tempo);

    return 0;
}

//bandejao.c
#include <stdio.h>
#include <stdlib.h>
#include "bandejao.h"

void Faz_Fila_Caixa_Vazia(Fila_Caixa *fila){

    fila->primeiro_da_fila_caixa =
(Celula_Pessoa_Caixa*)malloc(sizeof(Celula_Pessoa_Caixa));
    fila->primeiro_da_fila_caixa->tras =
(Celula_Pessoa_Caixa*)malloc(sizeof(Celula_Pessoa_Caixa));
    fila->ultimo_da_fila_caixa = fila->primeiro_da_fila_caixa;
    fila->ultimo_da_fila_caixa->tras = NULL;

}

int Testa_Fila_Caixa_Vazia(Fila_Caixa fila){

    return(fila.primeiro_da_fila_caixa == fila.ultimo_da_fila_caixa);
}

void Enfileira_Caixa(Fila_Caixa *fila, int *n_pessoa){

```

```

        fila->ultimo_da_fila_caixa->tras =
(Celula_Pessoa_Caixa*)malloc(sizeof(Celula_Pessoa_Caixa));
        fila->ultimo_da_fila_caixa = fila->ultimo_da_fila_caixa->tras;
        fila->ultimo_da_fila_caixa->num_pessoa = *n_pessoa;
        fila->ultimo_da_fila_caixa->tempo_fila_caixa = 0;
        fila->ultimo_da_fila_caixa->tras = NULL;
        *n_pessoa++;
    }

void Desenfileira_Caixa(Fila_Caixa *fila, int *n_pessoa){

    if(Testa_Fila_Caixa_Vazia(*fila)){
        printf("\nErro! A Fila está cheia!\n");
        return;
    }
    Celula_Pessoa_Caixa *aux;
    aux = fila->primeiro_da_fila_caixa;
    fila->primeiro_da_fila_caixa = fila->primeiro_da_fila_caixa->tras;
    *n_pessoa = fila->primeiro_da_fila_caixa->num_pessoa;
    free(aux);
}

void Faz_Fila_Bandeja_Vazia(Fila_Bandeja *fila){

    fila->primeiro_da_fila_bandeja =
(Celula_Pessoa_Bandeja*)malloc(sizeof(Celula_Pessoa_Bandeja));
    fila->primeiro_da_fila_bandeja->tras =
(Celula_Pessoa_Bandeja*)malloc(sizeof(Celula_Pessoa_Bandeja));
    fila->ultimo_da_fila_bandeja = fila->primeiro_da_fila_bandeja;
    fila->ultimo_da_fila_bandeja->tras = NULL;
}

int Testa_Fila_Bandeja_Vazia(Fila_Bandeja fila){

    return(fila.primeiro_da_fila_bandeja == fila.ultimo_da_fila_bandeja);
}

void Enfileira_Bandeja(Fila_Bandeja *fila, int *n_pessoa){

    fila->ultimo_da_fila_bandeja->tras =
(Celula_Pessoa_Bandeja*)malloc(sizeof(Celula_Pessoa_Bandeja));
    fila->ultimo_da_fila_bandeja = fila->ultimo_da_fila_bandeja->tras;
    fila->ultimo_da_fila_bandeja->num_pessoa = *n_pessoa;
    fila->ultimo_da_fila_bandeja->tempo_fila_bandeja = 0;
    fila->ultimo_da_fila_bandeja->tras = NULL;
    *n_pessoa++;
}

void Desenfileira_Bandeja(Fila_Bandeja *fila, int *n_pessoa){

    Celula_Pessoa_Bandeja *aux;
    aux = fila->primeiro_da_fila_bandeja;
    fila->primeiro_da_fila_bandeja = fila->primeiro_da_fila_bandeja->tras;
    *n_pessoa = fila->primeiro_da_fila_bandeja->num_pessoa;
    free(aux);
}

```

```

}

void Faz_Pilha_Cheia(Pilha_Bandeja *pilha){
    pilha->topo = 30;
}

int Testa_Pilha_Vazia(Pilha_Bandeja pilha){
    return(pilha.topo == 0);
}

void Empilha(Pilha_Bandeja *pilha){
    pilha->topo = pilha->topo + 10;
}

void Desempilha(Pilha_Bandeja *pilha){
    if(Testa_Pilha_Vazia(*pilha)){
        printf("Erro: pilha vazia\n");
    }
    pilha->topo--;
}

//bandejao.h

typedef struct pessoa_fila_caixa{
    int num_pessoa;
    int *num_pessoa2;
    int tempo_fila_caixa;
    struct pessoa_fila_caixa *tras;
}Celula_Pessoa_Caixa;

typedef struct fila_caixa{
    Celula_Pessoa_Caixa *primeiro_da_fila_caixa;
    Celula_Pessoa_Caixa *ultimo_da_fila_caixa;
}Fila_Caixa;

typedef struct pessoa_fila_bandeja{
    int num_pessoa;
    int *num_pessoa2;
    int tempo_fila_bandeja;
    struct pessoa_fila_bandeja *tras;
}Celula_Pessoa_Bandeja;

typedef struct fila_bandeja{

```

```

        Celula_Pessoa_Bandeja *primeiro_da_filha_bandeja;
        Celula_Pessoa_Bandeja *ultimo_da_filha_bandeja;

}Fila_Bandeja;

typedef struct unidade_bandeja{

    int bandeja;

}Unidade_Bandeja;

typedef struct pilha_bandeja{

    int topo;
    int fundo;
    Unidade_Bandeja vetor_de_bandejas[30];

}Pilha_Bandeja;

void Faz_Fila_Caixa_Vazia(Fila_Caixa*);
int Testa_Fila_Caixa_Vazia(Fila_Caixa);
void Enfileira_Caixa(Fila_Caixa*, int*);
void Desenfileira_Caixa(Fila_Caixa*, int*);
void Faz_Fila_Bandeja_Vazia(Fila_Bandeja*);
int Testa_Fila_Bandeja_Vazia(Fila_Bandeja);
void Enfileira_Bandeja(Fila_Bandeja*, int*);
void Desenfileira_Bandeja(Fila_Bandeja*, int*);
void Faz_Pilha_Cheia(Pilha_Bandeja*);
int Testa_Pilha_Vazia(Pilha_Bandeja);
//void Empilha(Pilha_Bandeja*, int);
void Empilha(Pilha_Bandeja*);
//void Desempilha(Pilha_Bandeja*, Unidade_Bandeja*);
void Desempilha(Pilha_Bandeja*);

```

### 3) Resultados

Os resultados não puderam ser obtidos pois não foi possível solucionar o problema. A grande dificuldade estava em incrementar o tempo de cada pessoa em relação ao tempo que se passou. Na entrada da primeira pessoa na fila do caixa, isso não foi um problema, pois bastava somar 1 ao tempo dessa pessoa. Mas de acordo com o tempo, a pilha de bandejas ficava vazia e uma nova fila para pegar as bandejas iria se formando, ou seja, como saber o tempo que uma pessoa ficou nas duas filas? O problema maior nem era saber esse tempo, mas sim saber como incrementar todos os usuários a cada unidade de tempo, sendo que alguns seriam incrementados e outros não.

### 4) Conclusão

O problema proposto é complicado de resolver para pessoas com capacidade limitada, com isso seria necessário mais conhecimento e treino, como ter feito um curso técnico de computação antes de entrar na universidade.