

Universidade de Itaúna

Grupo:

Davi Ventura Cardoso Perdigão – 82148

Edmilson Lino Cordeiro – 82920

Pedro Otávio Marques Silva – 82132

Professor:

Adriano Benigno

FILAS (QUEUES)

A implementação das Filas na linguagem C

ÍNDICE:

1 - Resumo.....	3
2 - Conceito Teórico.....	4
2.1 - Maneiras de Implementação de <i>Fila</i>	4
2.2 - Especificação da interface do tipo <i>Fila</i>	6
2.3 - Operações com <i>Fila</i>	6
3 – Exemplo Aplicado.....	7
4 – Referências Bibliográficas.....	11

RESUMO:

As filas são estruturas de dados do tipo **FIFO** (first-in first-out), onde o primeiro elemento a ser inserido, será o primeiro a ser retirado, ou seja, adicionam-se itens no fim e remove-se do início. O que diferencia a fila da pilha é exatamente a ordem de saída dos elementos, pois na pilha “o último que entra é o primeiro que sai”. A ideia fundamental da fila é que só podemos inserir um novo elemento no final dela e só podemos retirar o elemento do início.

A estrutura de fila é uma analogia natural com o conceito de fila que usamos no nosso dia a dia: quem primeiro entra em uma determinada fila é o primeiro a ser atendido (ou a sair da fila). Um exemplo de utilização em computação é a implementação de uma fila de impressão: se uma impressora é compartilhada por várias máquinas, deve-se adotar uma estratégia para determinar que documento seja impresso primeiro. A estratégia mais simples é tratar todas as requisições com a mesma prioridade e imprimir os documentos na ordem em que foram submetidos – o primeiro submetido é o primeiro a ser impresso.

Neste trabalho em questão, discutiremos duas estratégias para a implementação de uma estrutura de fila: usando vetor e lista encadeada. Lembrando que, para implementar uma fila, devemos ser capazes de inserir novos elementos em uma extremidade (fim) e retirar elementos da outra extremidade (início).

Palavras-Chave: Fila; Implementação; Vetor; Lista; Estrutura.

2 – Conceito Teórico

A ideia fundamental da fila é que só podemos inserir um novo elemento no final da fila e só podemos retirar o elemento do início. A analogia física é a fila de um banco, por exemplo. A ordem de uso é FIFO (first-in, first-out), ou seja, o primeiro que entrou é o primeiro a sair:

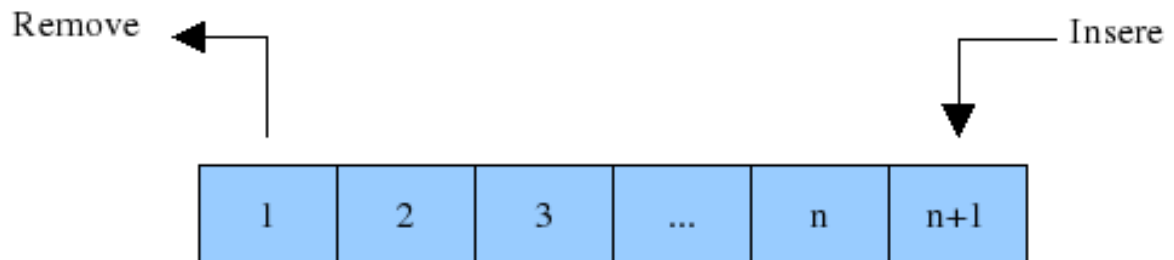


Figura 1 – Ilustração da estrutura da fila. Fonte: FARIAS, 2009.

2.1 - Maneiras de implementação de *Fila*:

■ Lista Encadeada:

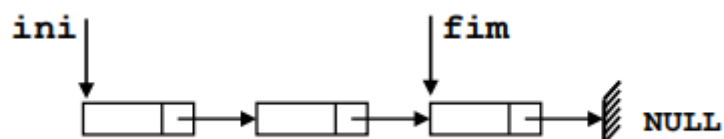


Figura 2 – Ilustração da implementação da fila como lista encadeada. Fonte: FEIJÓ, 2004.

Nesse caso, é fácil ter tamanho livre, porém requer mais memória e a memória é fragmentada. Pode ser simples ou duplamente encadeada e também pode ser circular simplesmente encadeada, ou circular duplamente encadeada.

■ Vetor (Forma Circular):

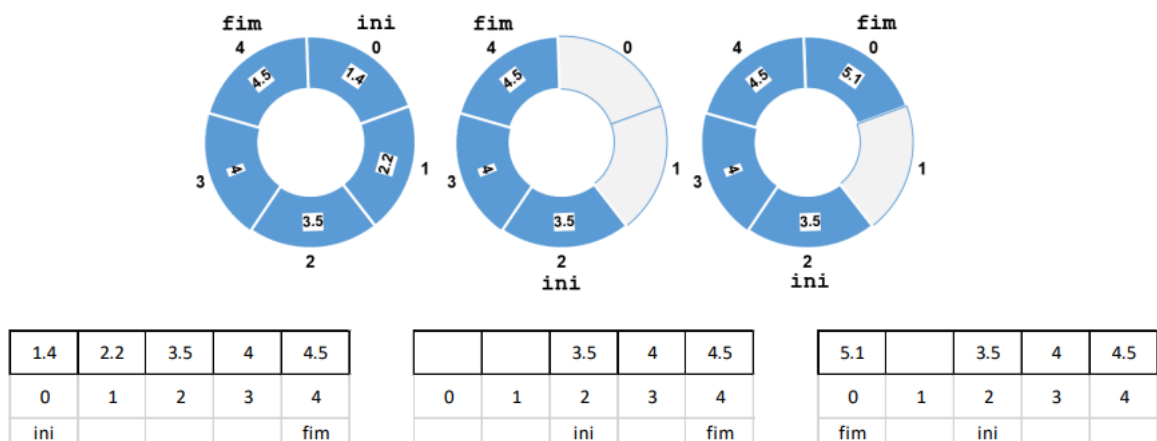


Figura 3 – Ilustração da implementação da fila como vetor em forma circular. Fonte: FEIJÓ, 2004.

Nesse caso, teremos um acesso mais ágil, porém pouca memória. No exemplo (N=5), há primeiro a retirada de dois elementos (1.4 e 2.2) e depois a inserção de um novo elemento (5.1).

■ Fila de Prioridade (Priority Queue):

Onde cada elemento tem um atributo de prioridade e a função de remoção encontra o elemento com a prioridade mais alta.

■ Fila Dupla (Deque):



Figura 4 – Ilustração da implementação de uma fila dupla. Fonte: FEIJÓ, 2004.

Nesse caso, é possível inserir novos elementos no início e no fim, e também retirar elementos de ambos os extremos. Se usar listas encadeadas, ela deve ser a duplamente encadeada (a simplesmente encadeada seria ineficiente). Ela simula dentro de uma mesma estrutura, duas filas, com os elementos em ordem inversa uma da outra. É uma generalização simultânea de pilha e fila.

2.2 - Especificação da interface do tipo *Fila*:

- Interface do tipo abstrato *Fila*: [fila.h](#)

- função [fila_cria](#)

- aloca dinamicamente a estrutura da fila
- inicializa seus campos e retorna seu ponteiro

- função [fila_libera](#)

- destrói a fila, liberando toda a memória usada pela estrutura

- função [fila_vazia](#)

- informa se a fila está ou não vazia

- função [fila_insere](#) e [fila_retira](#)

- insere e retira, respectivamente, um valor na fila
- retirar valor de uma fila vazia gera uma exceção

2.3 - Operações com *Fila*:

Todas as operações em uma fila podem ser imaginadas como as que ocorre numa fila de pessoas num determinado estabelecimento, por exemplo, exceto que os elementos não se movem na fila, conforme o primeiro elemento é retirado. O que se faz na realidade é indicar quem é o primeiro.

- criação da fila (informar a capacidade no caso de implementação sequencial - vetor);
- enfileirar (enqueue) - o elemento é o parâmetro nesta operação;
- desenfileirar (dequeue);
- mostrar a fila (todos os elementos);
- verificar se a fila está vazia (isEmpty);
- verificar se a fila está cheia (isFull - implementação sequencial - vetor).

3 – Exemplo Aplicado

Exemplo de Fila em C (Elaborado no Dev-C++):

```
#include <stdio.h>

#include <stdlib.h>

#include <locale.h>

// By: VOLTECX (Davi, Edmilson, Pedro Otávio)

#define tamanho 5

struct tfila
{
    int dados[tamanho];
    int ini;
    int fim;
} fila;

void fila_entrar(); //Declarar as funções
void fila_sair();
void fila_mostrar();
void menu_mostrar();

int main(void)
{
    setlocale(LC_ALL, "Portuguese");
    int op=1;
    fila.ini=0;
    fila.fim=0;
```

```

while (op!=0) //Selecionar opção no menu
{
    system("cls");
    fila_mostrar();
    menu_mostrar();
    scanf("%d",&op);
    switch (op)
    {
        case 1:
            fila_entrar();
            break;
        case 2:
            fila_sair();
            break;
    }
}
return(0);
}

```

void fila_entrar() //Adicionar um elemento no final da Fila

```

{
    if (fila.fim==tamanho)
    {
        printf("\nOps... A fila está cheia no momento. Por favor, tente outro
dia!\n\n");
        system("pause");
    }
    else

```



```

        {
            printf("\nDigite o valor a ser inserido na fila: ");
            scanf("%d",&fila.dados[fila.fim]);
            fila.fim++;
        }
    }

void fila_sair() //Retirar o primeiro elemento da Fila
{
    if (fila.ini==fila.fim)
    {
        printf("\nFila vazia, aguenta ai que logo aparecerá alguém!\n\n");
        system("pause");
    }
    else
    {
        int i;
        for (i = 0; i < tamanho; i++)
        {
            fila.dados[i] = fila.dados[i+1];
        }
        fila.dados[fila.fim] = 0;
        fila.fim--;
    }
}

```

```

void fila_mostrar() //Mostrar o conteúdo da Fila

```

```

{
    int i;
    printf("\n");
    printf("FILA VOLTECX: [ ");
    for (i=0;i<tamanho;i++)
    {
        printf("%d ",fila.dados[i]);
    }
    printf("]\n\n");
}

```

void menu_mostrar() //Menu de opções

```

{
    printf("\nEscolha uma das opções:\n");
    printf("1 - Incluir valor na Fila\n");
    printf("2 - Excluir valor da Fila\n");
    printf("0 - Finalizar\n\n");
}

```

```

EX_Filas.cpp
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <locale.h>
4  // By: VOLTECX (Davi, Edmilson, Pedro Otávio)
5  #define tamanho 5
6  struct tfila
7  {
8      int dados[tamanho];
9      int ini;
10     int fim;
11 } tfila;
12
13 void fila_entrar(); //Declarar as funções
14 void fila_sair();
15 void fila_mostrar();
16 void menu_mostrar();
17
18 int main(void)
19 {
20     setlocale(LC_ALL, "Portuguese");
21     int op=1;
22     fila.ini=0;
23     fila.fim=0;
24
25     while (op!=0) //Selecionar opção no menu
26     {
27         system("cls");
28         fila_mostrar();
29         menu_mostrar();
30         scanf("%d",&op);
31         switch (op)
32         {
33             case 1:
34                 fila_entrar();
35                 break;
36             case 2:
37                 fila_sair();
38                 break;
39         }
40     }
41     return(0);
42 }
43
44 void fila_entrar() //Adicionar um elemento no final da Fila
45 {
46     if (fila.fim==tamanho)
47     {
48         printf("\nOps... A fila está cheia no momento. Por favor, tente outro dia!\n\n");
49         system("pause");
50     }
51     else
52     {
53         printf("\nDigite o valor a ser inserido na fila: ");
54         scanf("%d",&fila.dados[fila.fim]);
55         fila.fim++;
56     }
57 }
58
59 void fila_sair() //Retirar o primeiro elemento da Fila
60 {
61     if (fila.ini==fila.fim)
62     {
63         printf("\nFila vazia, aguenta ai que logo aparecerá alguém!\n\n");
64         system("pause");
65     }
66     else
67     {
68         int i;
69         for (i = 0; i < tamanho; i++)
70         {
71             fila.dados[i] = fila.dados[i+1];
72         }
73         fila.dados[fila.fim] = 0;
74         fila.fim--;
75     }
76 }
77
78 void fila_mostrar() //Mostrar o conteúdo da Fila
79 {
80     int i;
81     printf("\n");
82     printf("FILA VOLTECX: [ ");
83     for (i=0;i<tamanho;i++)
84     {
85         printf("%d ",fila.dados[i]);
86     }
87     printf("]\n\n");
88 }
89
90 void menu_mostrar() //Menu de opções
91 {
92     printf("\nEscolha uma das opções:\n");
93     printf("1 - Incluir valor na Fila\n");
94     printf("2 - Excluir valor da Fila\n");
95     printf("0 - Finalizar\n\n");
96 }

```

Figura 5 – Programa demonstrando um exemplo de Fila em C.

Fonte: Autoria própria.

3 – Referências Bibliográficas

- FARIAS, Ricardo. **Filas (Queue)**, 2009.
- LIMA, Maria. **Pilhas e Filas: Conceitos e implementação em linguagem C**. Porto Alegre. 2009.
- SPADARI, Ana. **As filas em linguagem C: Primeiro a Entrar e Primeiro a Sair**. 2016.