

< VOLTAR

filas
queues, queue, row
Translated from Portuguese [More »](#)



Filas

Apresentar o conceito de filas e mostrar alguns exemplos da utilização dessa estrutura de armazenamento de dados.

NESTE TÓPICO

- > Introdução
- > Definição
- > Operações Básicas de uma Fila
- > Implementação de filas



Introdução

Fila é uma estrutura muito utilizada em computação. O primeiro elemento da lista representa o início da fila e o último representa o final da mesma. Nessa estrutura os acessos aos elementos que a compõem seguem uma regra definida.

Analisando este conceito, dizemos que só podemos inserir um novo elemento no final da fila e só podemos retirar o elemento do início da fila.

Definição

Uma fila é um tipo especial de lista linear em que as inserções são realizadas em um extremo, ficando as remoções restritas ao outro. O extremo onde os elementos são inseridos é denominado final da fila, e aquele de onde são removidos é denominado início da fila. Dessa forma, na fila o primeiro elemento que entra é o primeiro que sai (FIFO – first in, first out). Esta sigla é utilizada para descrever essa estratégia.

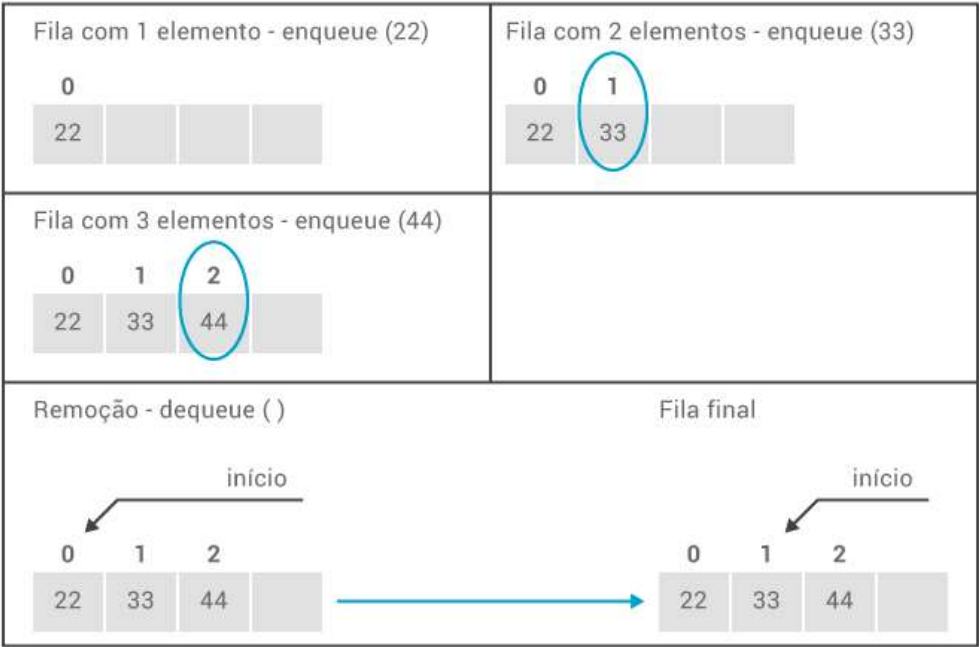
Operações Básicas de uma Fila

Existem, basicamente, duas operações básicas que devem ser implementadas numa estrutura de fila: a operação para inserir um novo elemento no final da fila e a operação para remover um elemento do início. É comum se referir a essas duas operações pelos termos em inglês enqueue (inserção no final) e dequeue (remoção no início).

Seja *f* uma variável do tipo fila e *x* um elemento qualquer. Então, temos:

- enqueue (*f*, *x*): procedimento que insere *x* no final de *f*.
- dequeue (*f*): função que remove o elemento do início de *f*, devolvendo o valor removido.

A figura a seguir ilustra o funcionamento conceitual das operações enqueue e dequeue em uma fila.



Funcionamento conceitual das operações enqueue e dequeue em uma fila.

Fonte: Do próprio autor

Veremos, a seguir, a implementação das seguintes operações para manipular e acessar as informações da fila:

- Inserir um elemento no final.
- Remover o elemento do início.
- Mostrar os elementos da fila.

Implementação de filas

Uma maneira muito comum de implementar a fila é por meio da utilização de vetores. Para tanto, é necessário sabermos a quantidade máxima de elementos que podem ser armazenados na fila, ou seja, neste caso, a estrutura fila tem seu limite conhecido.

Dado um vetor conhecido de números inteiros para armazenar os elementos da fila, os primeiros elementos ocupam as primeiras posições do vetor (n posições ocupadas). Temos, então, n elementos armazenados na fila, logo o elemento n-1 representa o elemento que está no final da fila. O tamanho do vetor é determinado por meio da constante **MAX**, a qual deve ser definida como constante no início do programa.

Abaixo é mostrado o procedimento “enqueue” para inserir um elemento no final da fila. O procedimento recebe como parâmetros o vetor “fila” e o novo elemento x que será inserido. Para a implementação desse procedimento, é preciso considerar que as variáveis *inicio* e *final* são globais e, inicialmente, zeradas.

```
1. void enqueue (int fila [MAX], int x)
2. {
3.     if (final == MAX)
4.         printf ("\n Fila Cheia!");
5.     else{
6.         fila [final] = x;
7.         final++;
8.     }
9. }
```

Para remover elementos da fila, deve ser utilizada a função “dequeue”. Além de remover o elemento, a função também retorna o valor que foi retirado da início da fila. Vale lembrar que a remoção na fila é feita por meio da variável *inicio*, que também é global e inicializada com zero. A cada elemento retirado, é feito um incremento nesta variável. Abaixo é mostrada a implementação da função “dequeue”.

```
1. int dequeue (int fila [MAX])
2. {
3.     int x;
4.     if (final >= 1 || inicio < final){
5.         x = fila [inicio];
6.         inicio++;
7.     }
8.     else printf ("\n Fila Vazia");
9.
10.    return (x);
11. }
```

A exibição dos elementos da fila deve ser feita por meio da utilização do laço *para*, onde o início do laço é definido pela variável *inicio* e o final pela variável *final*. Abaixo é mostrada a implementação do procedimento para exibir os elementos da fila.

```
1. void exibe (int fila [MAX])
2. {
3.     int i;
4.     if (final >= 1 && inicio < final){
5.         for (i=inicio; i<final; i++)
6.             printf ("\n fila [%d] = %d", i, fila [i]);
7.     }
8.     else printf ("\n Fila Vazia");
9. }
```

Exemplo 1

A seguir, apresentamos um exemplo completo de um programa que trabalha com fila.

```
1. #include <stdio.h>
2. #include <stdlib.h>
3. #define MAX 10
4.
5. //prototipos das funcoes
6. void enqueue (int fila [MAX], int x);
7. int dequeue (int fila [MAX]);
8. void exhibe (int fila [MAX]);
9.
10. int final, inicio;
11.
12. main(){
13.     int fila [MAX];
14.     inicio = 0;
15.     final = 0;
16.     enqueue (fila, 22);
17.     enqueue (fila, 33);
18.     enqueue (fila, 44);
19.     dequeue (fila);
20.     exhibe (fila);
21.     system ("PAUSE");
22. }
23.
24. void exhibe (int fila [MAX])
25. {
26.     int i;
27.     if (final >= 1 && inicio < final){
28.         for (i=inicio; i<final; i++)
29.             printf ("\n fila [%d] = %d", i, fila [i]);
30.     }
31.     else printf ("\n Fila Vazia");
32. }
33.
34. int dequeue (int fila [MAX])
35. {
36.     int x;
37.     if (final >= 1 || inicio < final){
38.         x = fila [inicio];
39.         inicio++;
40.     }
41.     else printf ("\n Fila Vazia");
42.
43.     return (x);
44. }
45.
46. void enqueue (int fila [MAX], int x)
47. {
48.     if (final == MAX)
49.         printf ("\n Fila Cheia!");
50.     else{
51.         fila [final] = x;
52.         final++;
53.     }
54. }
```

Aplicações de filas

A estrutura de fila é uma analogia natural com o conceito de fila que usamos no nosso dia a dia, tais como um atendimento bancário e as vendas de ingressos no geral.

Quem primeiro entra numa fila é o primeiro a ser atendido (ao sair da fila). Normalmente as filas são utilizadas na administração de recursos compartilhados, impondo uma prioridade por ordem de chegada.

Um exemplo clássico em computação é a implementação de fila de impressão onde, cada documento espera sua vez para ser impresso. No caso de uma impressora compartilhada por várias máquinas, deve-se adotar uma estratégia para saber que documento será impresso primeiro. O comum é tratar todas as impressões com o mesmo nível de prioridade, dessa forma, o primeiro documento submetido é o primeiro a ser impresso.

Outros exemplos na área computacional:

- Fila de processos que estão aguardando os recursos computacionais do computador;
- Fila de mensagens que serão enviadas através de um servidor de e-mail.

Exemplo 2

O algoritmo abaixo mostra a utilização de 2 filas (A e B) de tamanho 10. A fila A deverá armazenar os elementos pares e a fila B os elementos ímpares digitados pelo usuário. Por fim, os elementos de ambas as filas são exibidos.

```
1.  #include <stdio.h>
2.  #include <stdlib.h>
3.  #define MAX 10
4.
5.  //prototipos das funcoes
6.  void enqueue (int fila [MAX], int x);
7.  int dequeue (int fila [MAX]);
8.  void exhibe (int fila [MAX]);
9.
10. int final, inicio;
11.
12. main(){
13.     int filaA [MAX], filaB [MAX], n, i, resto;
14.     inicio = 0;
15.     final = 0;
16.     for (i=1;i<=MAX;i++){
17.         printf ("\n Digite o numero=");
18.         scanf ("%d", &n);
19.         resto = n%2;
20.         if (resto == 0)
21.             enqueue (filaA, n);
22.         else enqueue (filaB, n);
23.     }
24.     exhibe (filaA);
25.     exhibe (filaB);
26.     system ("PAUSE");
27. }
28.
29. void exhibe (int fila [MAX])
30. {
31.     int i;
32.     if (final >= 1 && inicio < final){
33.         for (i=inicio; i<final; i++)
34.             printf ("\n fila [%d] = %d", i, fila [i]);
35.     }
36.     else printf ("\n Fila Vazia");
37. }
38.
39. int dequeue (int fila [MAX])
40. {
41.     int x;
42.     if (final >= 1 || inicio < final){
43.         x = fila [inicio];
44.         inicio++;
45.     }
46.     else printf ("\n Fila Vazia");
47.
48.     return (x);
49. }
50.
51. void enqueue (int fila [MAX], int x)
52. {
53.     if (final == MAX)
54.         printf ("\n Fila Cheia!");
55.     else{
56.         fila [final] = x;
57.         final++;
58.     }
59. }
```

Quiz

Exercício

Filas

INICIAR ➤

Quiz

Exercício Final

Filas

INICIAR ➤

Referências

MIZRAHI, V. V. *Treinamento em linguagem C*, São Paulo: Pearson, 2008.

SCHILDT, H. C – *Completo e Total*, São Paulo: Pearson, 2006.



Avalie este tópico



Ajuda?
(<https://ava.un>

Biblioteca
(<https://www.uninove.br/conhec-a-uninove/biblioteca/sobre-a-biblioteca/apresentacao/>)
Portal Uninove
(<http://www.uninove.br>)
Mapa do Site

