

Apresentação

Filas são estruturas de dados que armazenam os elementos de forma sequencial (linear). As inserções e retiradas dos elementos são feitas em extremidades diferentes. Existe acesso às duas extremidades: começo, onde é feita a retirada, e término, onde é feita a inserção.

Nesta Unidade de Aprendizagem, você vai aprender como é o funcionamento de uma fila, criar funções e procedimentos para ela, além de ver exemplos de sua utilização.

Bons estudos.

Ao final desta Unidade de Aprendizagem, você deve apresentar os seguintes aprendizados:

- Identificar as regras de funcionamento de uma fila: FIFO: *First In First Out*.
- Criar funções e procedimentos de uma fila.
- Praticar exemplos com a utilização de filas.

Desafio

As estruturas de dados definem a organização, os métodos de acesso e as opções de processamento para coleções de itens de informação manipulados por um programa. Elas ajudam-nos a estruturar, organizar, armazenar e acessar os dados de um programa.

Considere o seguinte problema:

Você deve organizar alguns alunos a partir do código para inserir na fila conforme demanda.

[Clique aqui](#)

Agora, você deve consultar o nome do primeiro aluno da fila, digitando o código utilizado.

Infográfico

Uma fila é um conjunto ordenado de itens a partir do qual se pode eliminar itens em uma extremidade (chamada início da fila) e no qual se pode inserir itens na outra extremidade (chamada final da fila).

As filas são usadas para simulação de situações do mundo real, tal como pessoas esperando na fila de um banco, supermercado, cinema, etc. A primeira pessoa que entrar na fila será a primeira a ser atendida e a sair dela.

FILAS

Segue o critério **FIFO – First in, first out**. Ou seja, o primeiro elemento inserido será o primeiro retirado: inserção por uma extremidade e exclusão pela outra.

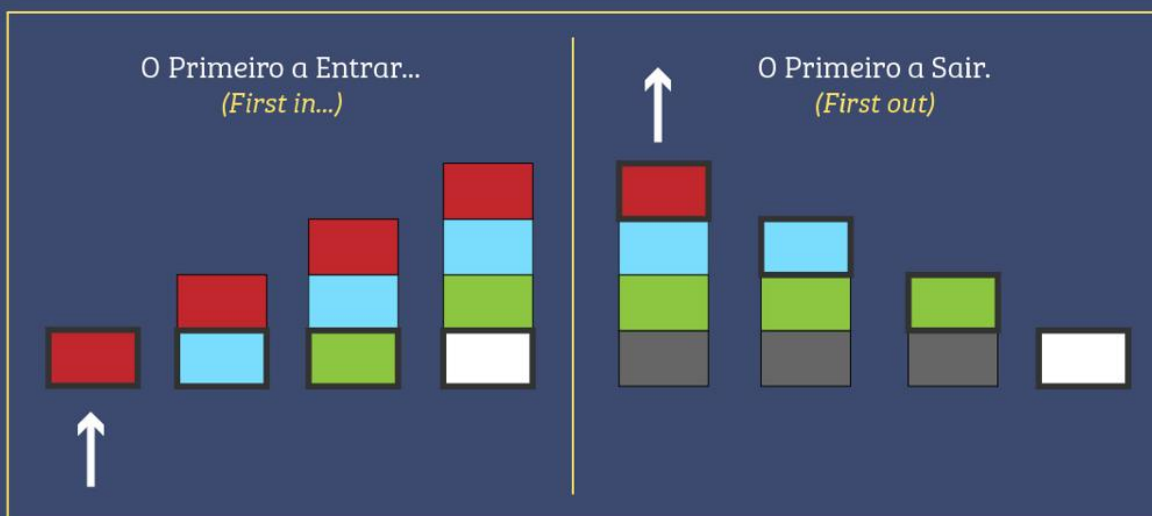
INSERE ELEMENTOS NA FILA



REMOVE ELEMENTOS NA FILA



Ou, ainda:



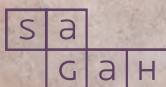
Conteúdo do Livro

Estrutura de dados é o modo particular de armazenamento e organização de dados no computador, para que possam ser usados eficientemente, facilitando sua busca e modificação.

No capítulo Filas, da obra *Estrutura de dados*, você compreenderá o funcionamento e a aplicação dessa estrutura no dia a dia da computação.

ESTRUTURA DE DADOS

Adriana de Souza Vettorazzo



SOLUÇÕES
EDUCACIONAIS
INTEGRADAS



Filas

Objetivos de aprendizagem

Ao final deste texto, você deve apresentar os seguintes aprendizados:

- Identificar as regras de funcionamento de uma fila *first in first out* (FIFO).
- Criar funções e procedimentos de uma fila.
- Praticar exemplos de utilização de fila.

Introdução

As estruturas de dados são muito utilizadas na computação em geral. Sistemas operacionais e aplicativos utilizam estrutura de dados para várias atividades importantes, como gerenciamento de memória, execução de processos, armazenamento e gerenciamento de dados no disco, entre outras.

Neste capítulo, você verá que as filas existem para estruturar, organizar, facilitar o acesso aos elementos e armazenar elementos.

Filas

As filas são estruturas de dados que armazenam os elementos de forma sequencial (linear). As inserções e retiradas dos elementos são feitas em extremidades diferentes. É possível acessar duas extremidades: começo, em que é feita a retirada; e término, em que é feita a inserção.

Podemos implementar a fila simplesmente colocando as restrições adequadas nas operações de adicionar e remover elementos de uma lista. Seu funcionamento é semelhante ao da pilha, só que na fila os elementos são adicionados e removidos de extremidades diferentes.

A pilha segue o critério *last in first out* (LIFO), ou seja, o último elemento a ser inserido na estrutura será o primeiro a ser retirado da estrutura; já a fila segue o critério FIFO, ou seja, o primeiro a ser inserido é o primeiro a ser retirado da estrutura, como você pode observar na Figura 1.

Fila



Figura 1. Critério FIFO.

Podemos fazer uma comparação com uma fila de atendimento, em que a primeira pessoa a entrar na fila, será a primeira pessoa a ser atendida e a sair da fila. As pessoas que chegarem para serem atendidas, serão posicionadas no final da fila. Portanto, essa manipulação é feita apenas pelas duas extremidades da fila. Veja a ilustração desse conceito na Figura 2.

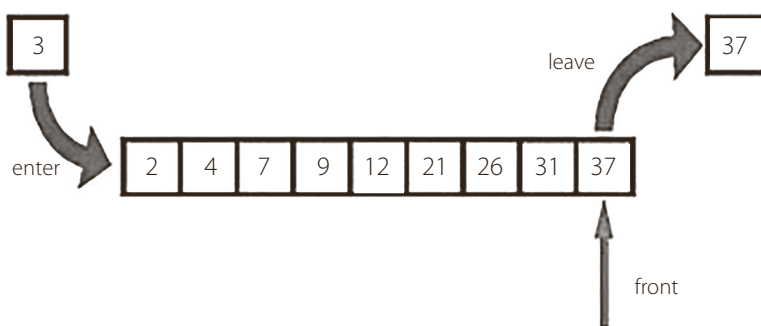
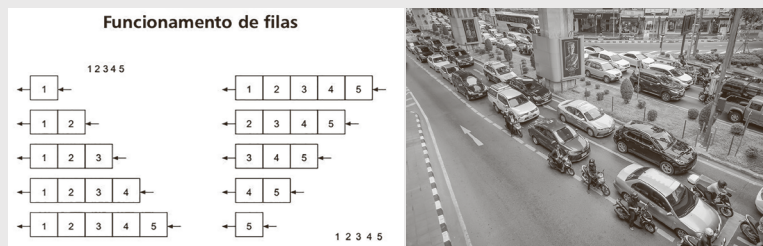


Figura 2. Manipulação da fila.



Fique atento

Na implementação de fila, a inserção é feita no final, e a remoção, no início da fila.



Fonte: Fotos593/Shutterstock.com.

Representação de filas

A implementação de filas se distingue pela natureza dos seus elementos (tipo do dado armazenado), pela maneira como os elementos são armazenados (estática ou dinamicamente) e pelas operações disponíveis para o tratamento da fila. A fila pode ser representada por duas formas, como descrito a seguir.

- **Estática:** caracteriza-se por utilizar um vetor (estrutura estática) para representar a fila.
- **Dinâmica:** caracteriza-se por utilizar uma lista encadeada (estrutura dinâmica) para representar a fila.

Implementando uma fila estática

As operações básicas suportadas em uma fila estática são:

- inserção;
- remoção;
- consulta;
- listagem.

Inserindo um elemento na fila

O código descrito a seguir demonstra a inserção de um elemento na fila.

```
void insere (node *FILA)
{
    node *novo=aloca();
    novo->prox = NULL;

    if (vazia (FILA))
        FILA->prox=novo;
    else{
        node *tmp = FILA->prox;

        while (tmp->prox != NULL)
            tmp = tmp->prox;

        tmp->prox = novo;
    }
    tam++;
}
```

Removendo um elemento na fila

Na estrutura de dados Fila, o elemento a ser removido é sempre o primeiro que foi inserido (FIFO).

```
node *retira (node *FILA)
{
    if (FILA->prox == NULL) {
        printf("Fila ja esta vazia\n");
        return NULL;
    } else {
        node *tmp = FILA->prox;
        FILA->prox = tmp->prox;
        tam--;
        return tmp;
    }
}
```

Consultando um elemento na fila

Veja a seguir como consultar um elemento na fila.

```
void consultarprimeiro ( ) {
    cabec();
    printf("\nConsulta primeiro aluno da
    fila\n"); if (tamfila !=0) {
        printf("\nMatricula Nome Polo\n");
        printf("-----\n");
        printf("%4d %-15s %2d\n",
        Aluno[0].Matricula, Aluno[0].Nome,
        Aluno[0].Polo); printf("-----\n");
    }
    else
        printf("\n\nA fila está vazia!!\n\n");
    printf("\n\nTecle enter para voltar para o
    menu\n"); getch();
}
```

Listando um elemento na fila

O código descrito a seguir mostra como listar um elemento na fila.

```
void listar () {  
    int i; cabec (  
    );  
    printf("\nListagem de alunos da  
fila\n"); if (tamfila != 0) {  
        printf("\nMatricula Nome Polo\n");  
        printf("-----  
-----\n");  
        for (i=0;i<tamfila;i++)  
            printf("%4d %-15s %2d\n",  
                Aluno[i].Matricula, Aluno[i].Nome,  
                Aluno[i].Polo); printf("-----  
-----\n");  
        printf("\n\nQuantidade de alunos na fila =  
%d\n",tamfila);  
    }  
    else  
        printf("\n\n Nao tem nenhum aluno na  
fila"); printf("\n\nTecle enter para voltar  
para o menu\n"); getch ( );  
}
```



Saiba mais

Em uma fila, a consulta é feita no primeiro elemento da fila, dessa forma, teremos sempre a informação de qual será o próximo elemento a ser retirado da estrutura.

Implementando uma fila dinâmica

A fila dinâmica é sempre implementada com dois ponteiros, um ponteiro no início da fila e o outro ponteiro no final da fila.

As operações básicas suportadas em uma fila dinâmica são as mesmas da fila estática.

Inserindo um elemento na fila dinâmica

Veja como inserir um elemento na fila dinâmica.

```
void inserir () {  
  
    TAluno *novono;  
  
    int i, Matriculal, Polol, continuar;  
    char Nomel[15];  
    do{  
        printf("\n Chegada de novo aluno na fila \n");  
        printf("\n Numero da Matricula: ");  
        scanf("%d", &Matriculal);  
        printf("\n Nome: ");  
        fflush(stdin);  
        gets(Nomel);  
        printf("\n Polo do Aluno(1- Batalha, 2- Valenca): ");  
        scanf("%d", &Polol);  
        tamfila++;  
        novono = (TAluno *) malloc(sizeof(TAluno));  
        novono->Matricula = Matriculal;  
        for (i=0; i<=14; i++)  
            novono->Nome[i] = Nomel[i];  
        novono->Polo = Polol;  
        novono->prox = NULL;  
        // Inserir novo Aluno na fila de Alunos  
        if (inicio == NULL) {  
            inicio = novono;  
            fim = novono;  
        }  
        else {  
            fim->prox = novono;  
            fim = novono;  
        }  
        printf("\n\nAluno Inserido com Sucesso!!!!\n\n");  
        printf("\n Continuar inserindo (1-sim/2-nao)? ");  
        scanf("%d", &continuar);  
    }while (continuar == 1); // verifica se quer continuar  
    inserindo}
```

Retirando um elemento da fila

No código a seguir você pode ver como retirar um elemento da fila.

```
void retirafila() {
    int confrem, continuar;
    do{
        printf("\nRetira primeiro Aluno da fila \n\n");
        noatual = inicio;
        if (noatual != NULL) {
            printf("\n\nMatricula Nome Polo\n");
            printf("-----\n");
            printf("%4d %-15s %2d\n", noatual->Matricula;
            noatual->Nome, noatual->Polo);
            printf("-----\n");
            printf("\n\nconfirma retirada do aluno (1-sim, 2-nao)? ");
            scanf("%d",&confrem);
            if (confrem ==1) {
                inicio = inicio->prox;
                free(noatual);
                tamfila--;
                printf("\n\n Retirado da fila com sucesso!!!!\n\n");
            }
            else
                printf("\n\n Retirada cancelada\n\n");
        }
        else // fila vazia
            printf("\n\nFila vazia!!\n\n");
        printf("\n\nDeseja retirar outro Aluno(1-sim, 2-nao)? ");
        scanf("%d",&continuar);
    }while (continuar ==1);
    // continuar retirando cliente da fila
}
```

Consultando um elemento na fila

Para consultar um elemento na fila você deve usar o seguinte código.

```
void consultarprimeiro ( ) {
    printf("\nConsulta primeiro aluno da fila\n\n");
    noatual = inicio; // coloca ponteiro no inicio da lista
    if (noatual != NULL) {
        printf("\n\n Matricula Nome Polo\n");
        printf("-----\n");
        printf("%4d %-15s %2d\n", noatual->Matricula,
            noatual->Nome, noatual->Polo);
        printf("-----\n");
    }

    else
        printf("\nA fila está vazia!!\n\n");
    printf("\nTecle enter para voltar para o menu\n");
    getche();
}
```

Listando um elemento na fila

O código a seguir demonstra como listar um elemento na fila.

```
void listar ( ) {
    noatual = inicio;
    cabec();
    printf("\nListagem de Aluno da fila\n\n");
    if (tamfila != 0) {
        printf("\nMatricula Nome Polo\n");
        printf("-----\n");

        while( noatual != NULL) {
            printf("%4d %-15s %2d\n",noatual->Matricula,
                noatual->Nome, noatual->Polo);
            noatual = noatual->prox;
        }

        printf("-----\n");

        printf("\n\nQuantidade de Aluno na fila = %d\n",tamfila);
    }

    else
        printf("\n\n Nao tem nenhum aluno na fila");
    printf("\n\nTecle enter para voltar para o menu\n");
    getche();
}
```




Fique atento

Na alocação estática, o tamanho da fila é definido na sua criação, quando a alocação é dinâmica, a fila pode ter um tamanho inicial, porém é possível alocar mais elementos quando necessário. Dessa forma o tamanho da fila expande dinamicamente.



Exemplo

São exemplos de aplicação de filas:

- *Spool* de impressão.
- Gerenciamento de processos em sistemas operacionais em que os processos aguardam na fila de pronto para execução.
- *Buffer* para gravação de dados em mídia.
- Processos de comunicação em redes de computadores.



Leituras recomendadas

SHACKELFORD, R. L. *Introduction to computing and algorithms*. Boston: Addison-Wesley, 1997. 399 p.

TENENBAUM, A. M.; LANGSAM, Y.; AUGENSTEIN, M. J. *Estruturas de dados usando C*. São Paulo: Makron Books, 1995. 904 p.

Encerra aqui o trecho do livro disponibilizado para esta Unidade de Aprendizagem. Na Biblioteca Virtual da Instituição, você encontra a obra na íntegra.

Conteúdo:



Dica do Professor

As estruturas de dados do tipo fila podem ser implementadas de forma dinâmica ou estática. No vídeo, você encontrará orientações de como diferenciá-las e aprenderá como reconhecer a melhor situação para a utilização de cada uma.



Aponte a câmera para o código e acesse o link do conteúdo ou clique no código para acessar.

Exercícios

- 1) Uma das estruturas de dados utilizadas na programação de computadores funciona conforme o princípio conhecido como *FIFO - First In First Out* e *LIFO - Last In First Out*. Essas estruturas são denominadas, respectivamente:
 - A) Lista circular e árvore.
 - B) Árvore e lista linear.
 - C) Pilha e lista circular.
 - D) Lista linear e fila.
 - E) Fila e pilha.

- 2) Suponha o seguinte cenário: Uma fila FIFO foi criada, e um nodo foi inserido a cada minuto, chegando a um total de dez elementos (dez minutos depois da criação da fila). A partir desse momento, decide-se remover um nodo. Qual deles será removido?
 - A) O primeiro (inserido no minuto 1).
 - B) O último (inserido no minuto 5).
 - C) O primeiro (inserido no minuto 10).
 - D) Qualquer nodo par, desde que o índice seja informado.
 - E) Qualquer nodo ímpar, desde que o índice seja informado.

- 3) Considerando os conceitos de estrutura de dados, analise as afirmativas a seguir e marque verdadeiro (V) ou falso (F):
 - () As filas são utilizadas para controlar o acesso de arquivos que concorrem a uma única impressora.
 - () A pilha é uma estrutura de dados baseada no princípio LIFO, no qual os dados que foram inseridos primeiro na pilha serão os últimos a serem removidos.

() Para gerenciar processos, sistemas operacionais utilizam filas para organizar processos que aguardam processamento.

A) V, V, V

B) V, F, F

C) F, V, F

D) F, F, F

E) V, V, F

4) Um conjunto ordenado de itens a partir do qual podem ser eliminados itens em uma extremidade e no qual podem ser inseridos itens na outra extremidade, é denominado:

A) Filas

B) Pilhas

C) Lista Simples

D) Lista encadeada

E) Árvore

5) Estrutura de Dados básicas como Fila são usadas em uma gama variada de aplicações computacionais. Marque a alternativa correta quanto a estas aplicações.

A) Sequência de trabalhos submetidas à impressora compartilhada em rede de computadores, onde o último trabalho recebido será o primeiro trabalho a ter sua solicitação atendida.

B) Estrutura hierárquica de diretórios e subdiretórios de alguns Sistemas Operacionais tais como Linux e Windows.

C) Atendimento de processos de E/S em um Sistema Operacional

D) A organização dos aplicativos em seu celular, quanto a abertura e fechamento dos mesmos.

E) Buffer para gravação de dados em mídia.

Na prática

Um item importante a ser estudado sobre filas é a fila de prioridade, que é utilizada em diversas situações.

Em sistemas operacionais:

No descalonamento de processos para ganhar tempo de processador.

Na manipulação de interrupções.

Em sistemas:

Um sistema de filas de transplantes.

Fila de pouso de aviões (prioridade por combustível disponível).

Portanto, usamos fila de prioridades toda vez que precisar estabelecer uma prioridade de acesso aos elementos.

Na implementação de filas de prioridades, é possível utilizar as mesmas operações das filas tradicionais, ou seja:

inserção;

remoção;

consulta;

e listagem.

```

struct cel{
    struct cel*prox;
    char dado;
    int prioridade;
};
typedef struct cel célula;
célula* inicializa ();
célula* cria (char v, int prioridade);
célula* inserir_dado(célula* f, char dado, int prioridade);
célula* retira (célula*f, char v);
int vazia (célula* f);
void imprime (célula* f);
char prox_da_fila (célula* f);
célula* inicializa ()
{
    Return NULL;
}
int vazia (célula* f)
{
    return (f==NULL);
}
Void imprime (célula* f)
{
    célula* t=f;
    printf ("%c(%d) - ", t-> prioridade);

    while (t-> prox!=NULL)
    {
        T=t-> prox;
        Printf("%c(%d) - ", t-> prioridade);
    };

    return;
}

célula* cria (char v, int prioridade)
{
    Célula* p = (célula*) malloc (sizeof (célula));
    p->dado = v;
    p->prioridade=prioridade;
    return p;
}
célula* inserir_dado (célula* f, char dado, int prioridade)
{
    célula* novo = cria(dado, prioridade);
    célula*ant = NULL;
    célula*p=f;

    while (p!=NULL && p->prioridade <-prioridade)
    {
        ant=p;
        p=p->prox;
    }
    If (ant ==NULL)
    {
        novo->prox=f;
        f=novo;
    }
    else
    {
        novo->prox=ant ->prox;
        ant-> prox = novo;
    }

    return f;
}
célula* retira(célula*f, char v)
{
    célula* ant =NULL;
    célula* p=f;
    while (p!=NULL && p-> dado!=v)
    {
        ant = p;
        p = p -> prox;
    }
    If ( p==NULL)
    {
        Return f;
    }
    If (ant == NULL)
    {
        f=p -> prox;
    }
    else
    {
        ant -> prox = p -> prox;
    }
    free (p);
    return f;
}
char prox_da_fila (célula*f)
{
    célula* temp = f;
    return temp -> dado;
}

```



Aponte a câmera para o código e acesse o link do conteúdo ou clique no código para acessar.

Baixe o código de uma fila de Prioridade

Saiba mais

Para ampliar o seu conhecimento a respeito desse assunto, veja abaixo as sugestões do professor:

Fila: definição

Saiba mais sobre o funcionamento de uma fila em linguagem C descomplicada.



Aponte a câmera para o código e acesse o link do conteúdo ou clique no código para acessar.

Listas, pilhas e filas

Saiba mais sobre outros tipos de estruturas de dados.



Aponte a câmera para o código e acesse o link do conteúdo ou clique no código para acessar.

As filas em linguagem C

Conheça outras funções da estrutura fila.



Aponte a câmera para o código e acesse o link do conteúdo ou clique no código para acessar.