

# Estrutura de Dados (CC4652)

## Aula 4 - Filas

Prof. Luciano Rossi - Turmas: 030 e 630

Prof. Leonardo Anjoletto Ferreira - Turma: 730

Ciência da Computação  
Centro Universitário FEI

2º Semestre de 2023

# Arranjos

## Introdução

- Um arranjo ou vetor é uma estrutura de dados que armazena, a princípio, dados primitivos desde que sejam do mesmo tipo (estrutura homogênea);
- Aqui é importante não confundirmos com a classe Vector do Java, ou com outras linguagens que permitem o uso de um vetor de tipos de dados diferentes;
- No contexto deste curso, estamos considerando um vetor como sendo uma estrutura homogênea de dados estaticamente alocados.

# Arranjos

## Introdução

- A alocação de um vetor é feita a partir da definição do tipo específico de dado que será armazenado, bem como da quantidade de elementos a ser armazenada;
- Considere um bloco de memória como sendo um conjunto de unidades de armazenamento;
- Assim, será reservada uma quantidade específica de bits contíguos na memória principal;
- O acesso aos elementos no vetor é facilitado por conta de sua organização sequencial na memória.

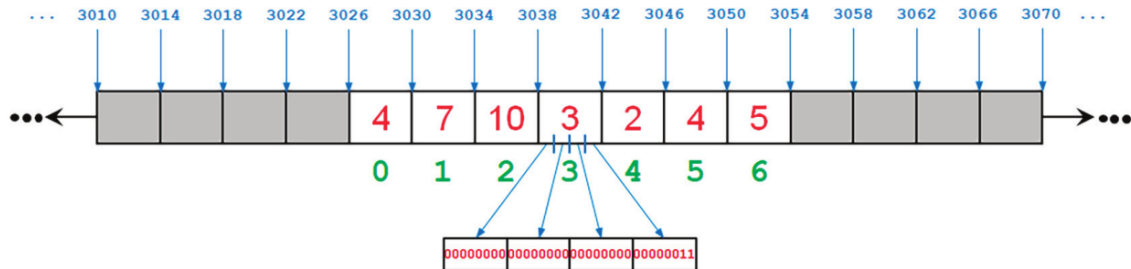
# Arranjos

## Introdução

- Considere o armazenamento de um vetor de inteiros positivos  $V = [4, 7, 10, 3, 2, 4, 5]$  em memória;
- Suponha que para cada valor inteiro, será alocado um total de quatro bytes e que a unidade de armazenamento, para esse exemplo, é igual a um byte;
- Nesse caso, deverão ser identificados um total de 28 bytes que sejam contíguos e estejam livres na memória;
- O processo de alocação retornará o endereço físico da primeira unidade de armazenamento alocada;
- Na prática, esse endereço será associado ao nome que foi atribuído ao vetor pelo programador.

# Arranjos

## Introdução



# Arranjos

## Introdução

- Vamos supor que queiramos acessar a posição cujo indexador é cinco;
- Assim, a partir de uma operação aritmética básica, é possível realizar esse acesso multiplicando o indexador pelo tamanho do tipo de dado e somando esse resultado ao endereço associado ao nome do vetor;

# Arranjos

## Introdução

- Em outras palavras, o indexador que queremos acessar (5) multiplicado pelo tamanho do tipo de dado em bytes (4 bytes para cada inteiro) resultará na distância, em bytes, entre o endereço armazenado e o endereço da posição objetivo ( $5 \times 4 = 20$ );
- Somando esse valor ao endereço armazenado, localizamos a posição objetivo ( $3026 + 20 = 3046$ );
- O processo descrito é transparente ao programador, ou seja, é realizado automaticamente, sem que haja a necessidade de qualquer intervenção do programador.

# Arranjos

## Fila Estática

- O vetor, como uma estrutura de dados estática, pode ser utilizado para a implementação de outras estruturas similares;
- Nesse sentido, a fila é uma dessas estruturas e considera um vetor estático e uma determinada regra de acesso;
- Uma regra de acesso define a forma pela qual podemos realizar manipulações sobre a estrutura.



# Arranjos

## Fila Estática

- Considere uma analogia com uma fila de banco; nesse caso, o cliente que chega primeiro à fila será, também, atendido em primeiro lugar;
- Essa ideia nos ajuda a definir a regra de acesso a uma fila. Denominamos por FIFO (First In First Out) a regra que diz: o primeiro que entra é o primeiro que sai;
- Desse modo, como em uma fila de banco, o primeiro elemento a entrar na fila será o primeiro a sair dela.

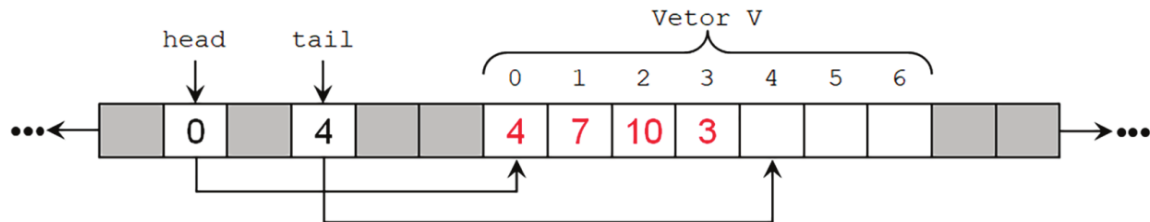
# Arranjos

## Fila Estática

- Uma fila, considerada sobre um vetor, apresenta dois pontos de acesso;
- Um ponto de acesso é a cabeça da fila, o qual denominaremos de head, e é a partir desse ponto que os elementos são retirados da fila;
- Outro ponto de acesso é o final da fila, o qual denominaremos de tail, onde os elementos entram na fila;
- Cada um dos pontos de acesso referencia uma posição na fila (ou, nesse caso, um vetor) utilizando os indexadores.

# Arranjos

## Fila Estática



# Arranjos

## Fila Estática

- As estruturas de dados estão associadas a operações possíveis de serem realizadas sobre elas;
- No caso da fila, podemos ter as operações de inserção e remoção de elementos, as quais serão denominadas de enqueue e dequeue, respectivamente;
- Para realizar a operação enqueue (inserção), é necessário que haja espaço na fila para um novo elemento;
- Ou seja, se a fila estiver cheia não é possível inserir um novo elemento;
- A fila estará cheia quando a variável tail estiver armazenando um valor igual ao tamanho da fila;
- Veja que, nesse caso, a variável tail estará apontando para uma posição fora dos limites da fila.

# Arranjos

Fila Estática - Função para verificar se uma fila está cheia

IsFull(V)

1.     **se**  $tail = |V|$
2.         **retorna** VERDADEIRO
3.     **senão**
4.         **retorna** FALSO

$\triangleleft |V|$  indica o tamanho de  $V$

# Arranjos

Fila Estática - Procedimento para inserir o valor  $x$  na fila  $V$

Enqueue ( $V, x$ )

1.     **se** ! *IsFull* ( $V$ )
2.          $V[\textit{tail}] \leftarrow x$
3.          $\textit{tail} \leftarrow \textit{tail} + 1$
4.     **senão**
5.         **erro** *overflow*

◁ ! indica a operação lógica NÃO

# Arranjos

Fila Estática - Função para verificar se uma fila V está vazia

*IsEmpty()*

1.       **se** *head = tail*
2.               **retorna** VERDADEIRO
3.       **senão**
4.               **retorna** FALSO

# Arranjos

Fila Estática - Função para remover um elemento da fila

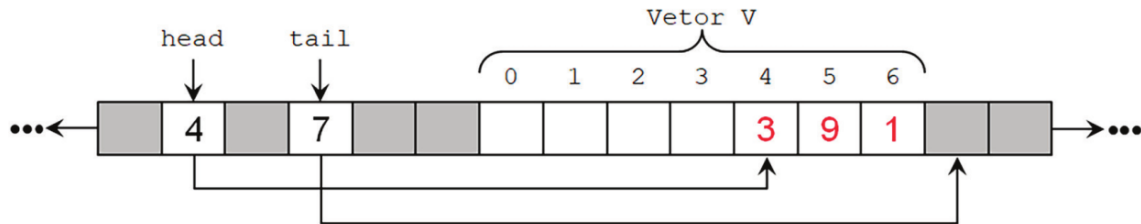
Dequeue (*V*)

1.       **se** ! *IsEmpty* ()
2.            $x \leftarrow V[head]$
3.            $head \leftarrow head + 1$
4.       **retorna** *x*
5.       **senão**
6.           **erro** *underflow*



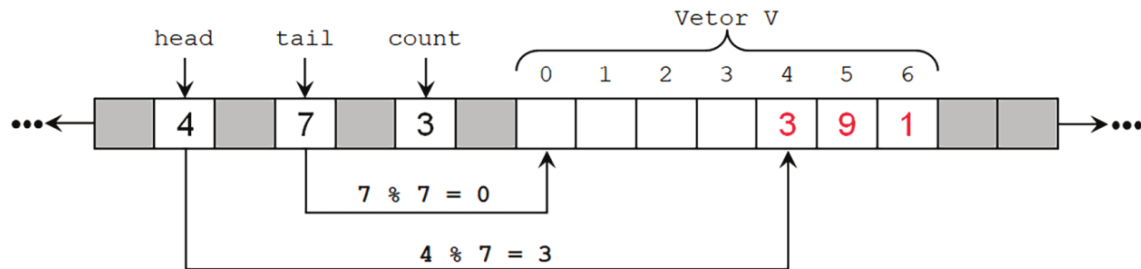
# Arranjos

Fila Estática - Problema com essa estrutura



# Arranjos

## Fila Estática Circular



# Estrutura de Dados (CC4652)

## Aula 4 - Filas

Prof. Luciano Rossi - Turmas: 030 e 630

Prof. Leonardo Anjoletto Ferreira - Turma: 730

Ciência da Computação  
Centro Universitário FEI

2º Semestre de 2023