# Fila Sequencial

# Definição de Fila

- Filas são conjuntos de itens a partir dos quais podemos eliminar itens de uma extremidade e acrescentar itens na outra extremidade.

# Definição de Fila

- Diferentemente da pilha que é uma estrutura do tipo LIFO ( Last-In First-Out ), a fila é conhecida por sua política FIFO ( First-In First-Out ) - primeiro elemento a entrar será o primeiro elemento a sair.

# Definição de Fila

- Filas são usadas em sistemas operacionais para gerenciar as tarefas a serem executadas em um determinado tempo pelo processador.

- Os elementos da fila são ordenados conforme o tempo de chegada.

- Somente o elemento que entrou na fila primeiro, ou seja, o que está na fila a mais tempo é que pode ser removido e visualizado primeiro.

# Aplicações de Filas

- Fila de processos do sistema operacional
- Fila de um banco
- Fila para o check-in de um vôo
- Tratamento de teclas acionadas no teclado do computador
- Fila de impressão

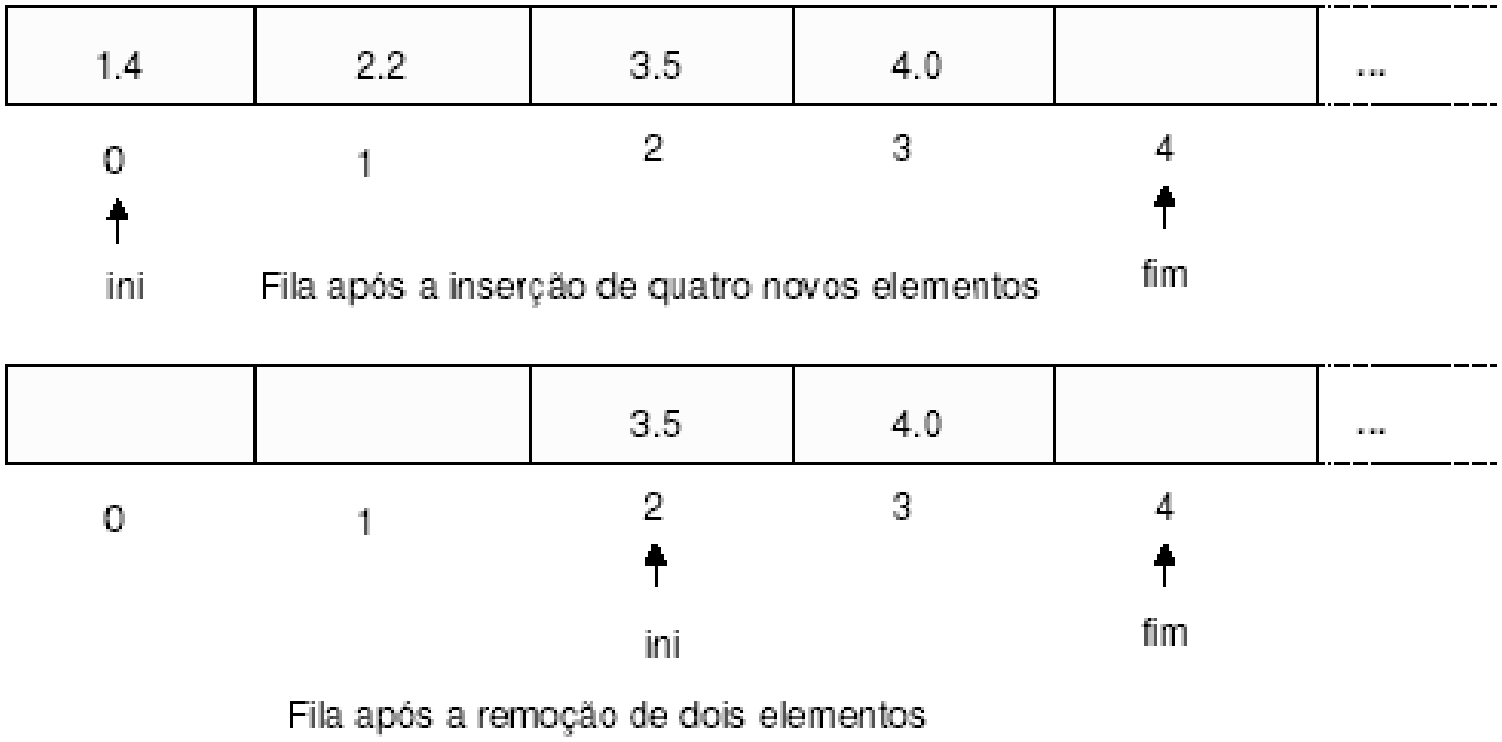
# Operações Básicas sobre Filas

- Criar uma fila vazia
- Inserir um elemento no fim
- Retirar um elemento do início
- Listar um elemento
- Listar toda a fila
- Liberar a fila

# Implementação de Filas

- Uma fila pode ser implementada considerando as seguintes características:
  - O novo elemento sempre é inserido no final (ordem de chegada);
  - O elemento removido é sempre o que chegou há mais tempo na fila;
  - A consulta retorna o primeiro elemento da fila.

# Implementação de Fila com Vetor



# Criando Fila Usando Vetor

- Uma forma de implementar uma fila é utilizando um vetor e duas variáveis para armazenar, respectivamente, as posições do primeiro e quantidade de elementos da fila.

- Declaração de uma fila:

```
#define MAX_FILA 100
typedef struct {
   int itens [MAX_FILA];
   int n; /*numero de elementos armazenados na fila*/
   int ini; /* indice para o inicio da fila */
} TFila;
typedef TFila *PFila;
```

# Criando Fila Usando Vetor

- Primeiro inicializamos uma pilha como vazia inicializando os campos `n` e `ini` com zero:

```
PFila cria_fila(){
   PFila fila = (PFila) malloc (sizeof (TFila));
   fila->n = 0;
  fila->ini = 0;
   return (fila);
}
```

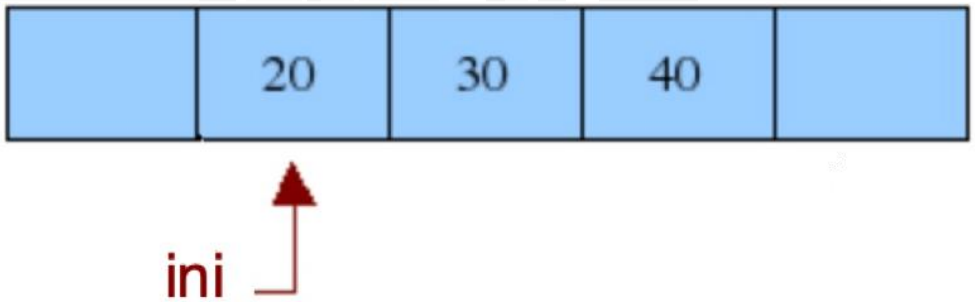
# Inserindo na Fila

- Para inserir na fila, basta inserir o elemento na última posição do vetor e atualizar a quantidade de elementos.

# Inserindo na Fila

- Para inserir na fila, basta inserir o elemento na última posição do vetor e atualizar a quantidade de elementos.

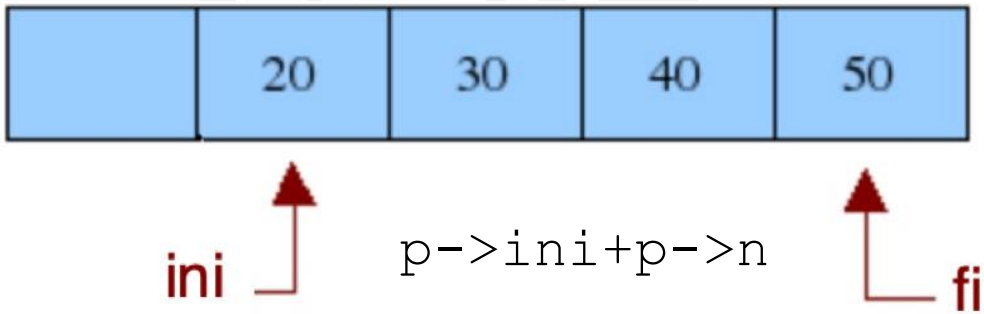
```
int insere (PFila p, int val){
      int fim;
   if (p==NULL) {
      prinft("Fila nao inicializada!\n");
            return -1; }
   if (p->n == MAX_FILA ) {
       printf("Fila cheia!\n");
      return -1; }
   fim =
```



# Inserindo na Fila

- Para inserir na fila, basta inserir o elemento na última posição do vetor e atualizar a quantidade de elementos.

```
int insere (PFila p, int val){
      int fim;
   if (p==NULL) {
      prinft("Fila nao inicializada!\n");
                return -1; }
   if (p->n == MAX_FILA ) {
       printf("Fila cheia!\n");
      return -1; }
   fim =
```



# Inserindo na Fila

- Para inserir na fila, basta inserir o elemento na última posição do vetor e atualizar a quantidade de elementos.

```
int insere (PFila p, int val){
      int fim;
   if (p==NULL) {
      prinft("Fila nao inicializada!\n");
                return -1; }
   if (p->n == MAX_FILA ) {
       printf("Fila cheia!\n");
      return -1; }
   fim = (p->ini+p->n)%MAX_FILA;
   p->itens[fim] = val;
   p->n++;
   return 0; }
```

# Retirando na Fila

- Para retirar um elemento da fila, basta retirar o primeiro elemento da fila e atualizar o índice.

```
int retira (PFila p, int* val){
   if (p==NULL) {
      prinft("fila nao inicializada!\n");
       return -1; }
   if (p->n ==0) {
      printf("Fila vazia!\n");
      return -1; }
   *val = p->itens[p->ini];
    p->ini = incr_indice(p->ini);
    p->n--;
   return 0;
}
```

# Problema na Implementação de Fila com Vetor

- A parte ocupada do vetor pode chegar à última posição e ainda haver espaço antes do início da fila.

- Solução: usar uma estratégia circular (se o último elemento da fila ocupa a última posição do vetor, insere-se novos elementos a partir do início do vetor, casa haja espaço).

# Atualização do Índice com Fila Circular

- A função de incremento do índice deve levar em consideração que a fila está sendo implementada utilizando fila circular.

```
int incr_indice(int i) {
    if (i==MAX_FILA-1)
       return 0;
    else
       return i+1;
}
```

ou

```
int incr_indice(int i) {
    return (i+1)%MAX_FILA;
}
```

# Exercícios

- Faça um programa que leia duas filas numéricas ordenadas crescentemente a partir do início da fila e que contenham uma função que transfira os elementos dessas filas para uma terceira fila, inicialmente vazia, de modo que ela também fique ordenada crescentemente com o menor valor no início da fila.

```c
int main(int argc, const char * argv[]) {
    PFila fila1, fila2, fila3;
    int val, i;
    fila1 = cria_fila();
    fila2 = cria_fila();

    //inicializa as listas
    printf("Digite os elementos da primeira fila (digite -99999 para sair):\n");
    i = 1;
    do{ printf("Elemento %d: ",i++);
        scanf("%d", &val);
        if (val!=-99999)
          if (insere(fila1,val)==-1)
            printf("Não foi possivel inserir o elemento\n");
    }while(val!=-99999);

    printf("Digite os elementos da segunda fila (digite -99999 para sair):\n");
    i = 1;
    do{ printf("Elemento %d: ",i++);
        scanf("%d", &val);
        if (val!=-99999)
          if (insere(fila2,val)==-1)
            printf("Não foi possivel inserir o elemento\n");
    }while(val!=-99999);
```

```
// une as duas filas criando uma nova fila
   fila3 = cria_fila();
   while (fila1->n>0 || fila2->n>0){
      if(fila2->n==0 || (fila1->n>0 &&  fila1->itens[fila1->ini]<=fila2->itens[fila2->ini])){
         //retira um elemento da fila 1 e o insere na nova fila
        if (retira(fila1,&val)==-1)
            printf("Não foi possivel retirar o elementoda fila 1.\n");
        else{
            if (insere(fila3,val)==-1){
              printf("Não foi possivel inserir o elemento\n");
              break; // sai do laço
      }   }   } //fecha os 3 ifs
      else{ //retira um elemento da fila 2 e o insere na nova fila
           if (retira(fila2,&val)==-1)
                printf("Não foi possivel retirar o elementoda fila 2.\n");
           else{
                     if(insere(fila3,val)==-1){
                           printf("Não foi possivel inserir o elemento\n");
                           break;  // sai do laço
                     }
           }
       }
   }    //while
```

```
// imprime os elementos da fila criada
   printf("\nFila resultante:\n");
   while (fila3->n>0){
      if (retira(fila3,&val)==-1){
         printf("Não foi possivel retirar o elementoda fila 3.\n");
         break;
      }
      else{
         printf("%d   ", val);
      }
   }

// libera a memória alocada
   free(fila1);
   free(fila2);
   free(fila3);
   printf("\n\n");
   return 0;
}
```

# Exercícios

- Em um supermercado há vários guichês, mas os clientes esperam sempre numa fila única e são atendidos conforme a disponibilidade dos guichês. Sendo assim, cada pessoa que chega, entra no final desta fila. No entanto, cada vez que chega uma pessoa com atendimento prioritários (por exemplo, um idoso) ele deverá ser atendido antes de qualquer outra pessoa não prioritária, porém respeitando a ordem de chegada das pessoas com atendimento prioritário na fila. Sendo assim, ao chegar a primeira pessoa com atendimento prioritário, ela deverá entrar no começo da fila. Ao chegar a próxima pessoa com atendimento prioritário, esta deverá entrar logo atrás do primeiro que chegou necessitando de atendimento prioritário, e assim por diante. Armazene apenas o tipo da pessoa como informação, pois isso será necessário para o controle da fila. Considere que cada tipo é representado por uma única letra: P para prioritário e O para outros. Faça um programa que controle a entrada das pessoas na fila e o atendimento de cada uma pessoas. A cada passo do laço o programa deverá perguntar ao usuário se ele quer inserir uma pessoa na fila ou retirar e imprimir a fila a cada iteração. O programa deverá terminar quando não houver mais pessoas na fila.