

Filas de Prioridade em C

INF0286 | INF0447 – Algoritmos e Estruturas de Dados I

Prof. Me. Raphael Guedes

raphaelguedes@ufg.br

2024

INF

INSTITUTO DE
INFORMÁTICA



Operações Básicas

- criação da fila;
- inserção de um elemento na fila com prioridade;
- remoção do elemento da fila com maior prioridade;
- acesso ao elemento do início da fila (maior prioridade);
- destruição da fila;
- tamanho, fila cheia ou fila vazia.

Modularização

- Fila.h
 - tamanho do vetor;
 - informação a ser armazenada na fila: paciente, aluno, ordem de serviço, etc;
 - tipo opaco da fila: `typedef struct fila_prioridade Filaprio;`
 - protótipos das funções;
- Fila.c
 - definição da fila de prioridade;
 - implementação das funções.
- main.c
 - declaração das informações a serem armazenadas;
 - uso das funções implementadas.

Importante

- A depender da implementação (vetor com prioridade descendente, no caso):
 - o início do vetor é o final da fila (prioridade menor);
 - o final do vetor é o início da fila (prioridade maior):
 - onde a remoção acontece.

Filas de Prioridade: inserção

Inserindo um elemento na fila de prioridade (array ordenado)

```
01  int insere_FilaPrio(FilaPrio* fp, char* nome, int prio){
02      if(fp == NULL)
03          return 0;
04      if(fp->qtd == MAX)//fila cheia
05          return 0;
06
07      int i = fp->qtd-1;
08      while(i >= 0 && fp->dados[i].prio >= prio){
09          fp->dados[i+1] = fp->dados[i];
10          i--;
11      }
12
13      strcpy(fp->dados[i+1].nome,nome);
14      fp->dados[i+1].prio = prio;
15      fp->qtd++;
16      return 1;
17  }
```

Fila inicial

| | | | | | | |
|-------|----|----|----|--|--|-------|
| dados | 16 | 23 | 33 | | | |
| | 0 | | | | | MAX-1 |

Busca posição na fila deslocando os elementos (se necessário)

```
i = fp->qtd-1;
while(i >= 0 && fp->dados[i].prio >= prioridade){
    fp->dados[i+1] = fp->dados[i];
    i--;
}
```

| | | | | | | |
|-------|----|----|----|----|--|-------|
| dados | 12 | 16 | 23 | 33 | | |
| | 0 | | | | | MAX-1 |

| | | | | | | |
|-------|----|----|----|----|--|-------|
| dados | 16 | 19 | 23 | 33 | | |
| | 0 | | | | | MAX-1 |

Inserir elemento

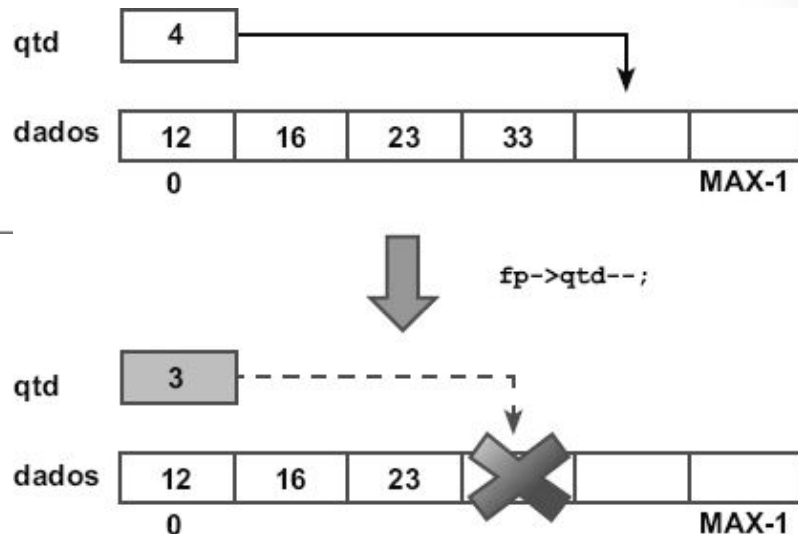
```
strcpy (fp->dados[i+1].nome,nome);
fp->dados[i+1].prio = prioridade;
fp->qtd++;
```

| | | | | | | |
|-------|----|----|----|----|--|-------|
| dados | 16 | 23 | 33 | 40 | | |
| | 0 | | | | | MAX-1 |

Filas de Prioridade: remoção

Removendo um elemento da fila de prioridade (*array ordenado*)

```
01  int remove_FilaPrio(FilaPrio* fp){  
02      if(fp == NULL)  
03          return 0;  
04      if(fp->qtd == 0)  
05          return 0;  
06      fp->qtd--;  
07      return 1;  
08  }
```



Filas de Prioridade: consulta

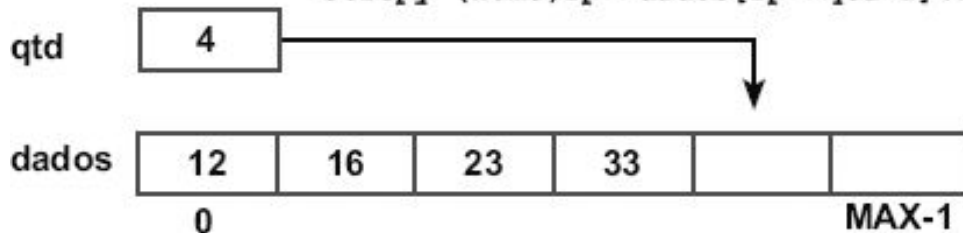
- Segue a política da fila:
 - o acesso é realizado no início da fila (fim do vetor ordenado)

Retorna o elemento com maior prioridade na fila (*array ordenado*)

```
01  int consulta_FilaPrio(FilaPrio* fp, char* nome){
02      if(fp == NULL || fp->qtd == 0)
03          return 0;
04      strcpy(nome, fp->dados[fp->qtd-1].nome);
05      return 1;
06  }
```

O primeiro da fila é o último do *array*

`strcpy (nome, fp->dados[fp->qtd-1].nome);`



Referências

- BACKES, André Ricardo. **Algoritmos e Estruturas de Dados em C**. Rio de Janeiro: LTC, 2023.

Obrigado!

raphaelguedes@ufg.br
raphaelguedes@inf.ufg.br

