

Algoritmos e Estruturas de Dados: Fila

Rômulo Silva de Oliveira
Departamento de Automação e Sistemas – DAS – UFSC

romulo@das.ufsc.br
http://www.das.ufsc.br/~romulo
Maio/2011

Rômulo Silva de Oliveira, DAS-UFSC, maio/2011 1

Fila – Interface 1/6

- **queue_init**
- void queue_init(Queue *queue, void (*destroy)(void *data));
- **Retorno**
 - Nenhum.
- **Descrição**
 - Inicializa a fila especificada por *queue*. Esta operação deve ser chamada para uma fila antes que a fila seja usada com qualquer outra operação. O argumento *destroy* fornece uma maneira para liberar dados alocados dinamicamente quando *queue_destroy* for chamada. Ela trabalha de forma similar àquela descrita para *stack_destroy*. Para uma fila contendo dados que não devem ser liberados, *destroy* deve ser NULL.

Rômulo Silva de Oliveira, DAS-UFSC, maio/2011 4

Referências

- Mastering Algorithms with C
Kyle Loudon
O'Reilly, 1999
- Livros de algoritmos e estruturas de dados em geral

Rômulo Silva de Oliveira, DAS-UFSC, maio/2011 2

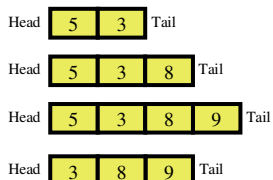
Fila – Interface 2/6

- **queue_destroy**
- void queue_destroy(Queue *queue);
- **Retorno**
 - Nenhum.
- **Descrição**
 - Destroi a fila especificada por *queue*. Nenhuma operação é permitida após a chamada de *queue_destroy* a não ser que *queue_init* seja chamada novamente. A operação *queue_destroy* remove todos os elementos de uma fila e chama a função passada como *destroy* para *queue_init* uma vez para cada elemento que é removido, desde que *destroy* não seja NULL.

Rômulo Silva de Oliveira, DAS-UFSC, maio/2011 5

Fila – Introdução

- Dados são retirados na mesma ordem na qual foram inseridos
- Insere sempre no fim da fila
- Retira sempre do início da fila



Rômulo Silva de Oliveira, DAS-UFSC, maio/2011 3

Fila – Interface 3/6

- **queue_enqueue**
- int queue_enqueue(Queue *queue, const void *data);
- **Retorno**
 - 0 se a inserção do elemento na fila tenha sucesso, ou -1 caso contrário.
- **Descrição**
 - Enfilera um elemento na cauda de uma fila especificada por *queue*. O novo elemento contém um pointer para *data*, logo a memória referenciada por *data* deve permanecer válida pelo tempo que o elemento permanecer na fila. É responsabilidade do chamador gerenciar a memória associada com *data*.

Rômulo Silva de Oliveira, DAS-UFSC, maio/2011 6

Fila – Interface 4/6

- `queue_dequeue`
- `int queue_dequeue(Queue *queue, void **data);`
- **Retorno**
 - 0 se a remoção do elemento da fila tiver sucesso, ou -1 caso contrário.
- **Descrição**
 - Remove da fila um elemento da cabeça da fila especificada por `queue`. Após o retorno, `data` aponta para os dados armazenados no elemento que foi removido da fila. É responsabilidade do chamador gerenciar a memória associada com os dados.

Fila – Header 1/1

```
#ifndef QUEUE_H
#define QUEUE_H

#include <stdlib.h>
#include "list.h"

typedef List Queue;

#define queue_init list_init

#define queue_destroy list_destroy

int queue_enqueue(Queue *queue, const void *data);

int queue_dequeue(Queue *queue, void **data);

#define queue_peek(queue) ((queue)->head == NULL ? NULL : (queue)->head->data)

#define queue_size list_size

#endif
```

Fila – Interface 5/6

- `queue_peek`
- `void* queue_peek(const Queue *queue);`
- **Retorno**
 - Dados armazenados no elemento na cabeça da fila, ou NULL se a fila estiver vazia.
- **Descrição**
 - Macro que obtém os dados armazenados no elemento na cabeça da fila especificada por `queue`.

Fila – Implementação 1/1

```
#include <stdlib.h>

#include "list.h"
#include "queue.h"

int queue_enqueue(Queue *queue, const void *data) {
    return list_ins_next(queue, list_tail(queue), data);
}

int queue_dequeue(Queue *queue, void **data) {
    return list_rem_next(queue, NULL, data);
}
```

Fila – Interface 6/6

- `queue_size`
- `int queue_size(const Queue *queue);`
- **Retorno**
 - Número de elementos na fila.
- **Descrição**
 - Macro que obtém o número de elementos na fila especificada por `queue`.

Fila – Sumário

- Trata-se de um caso especial de Lista
- Sempre insere no fim
- Sempre retira da frente
- Também conhecida como FIFO: First-In First-Out

