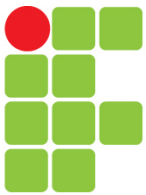
Filas e filas circulares– *Queues*

Programação de computadores II

Prof. Renan Augusto Starke

Instituto Federal de Santa Catarina – IFSC
Campus Florianópolis
renan.starke@ifsc.edu.br

30 de setembro de 2016



INSTITUTO FEDERAL
SANTA CATARINA

Ministério da Educação
Secretaria de Educação Profissional e Tecnológica
INSTITUTO FEDERAL DE SANTA CATARINA

Tópicos da aula

Tópico

Objetivos

- Entender o conceito de filas – *queue*
- Aprender sua utilização
- Aprender sua implementação
- Utilizar o conceito de "dados abstratos e estruturados" para fornecer funções simples de manipulação de filas

Tópico

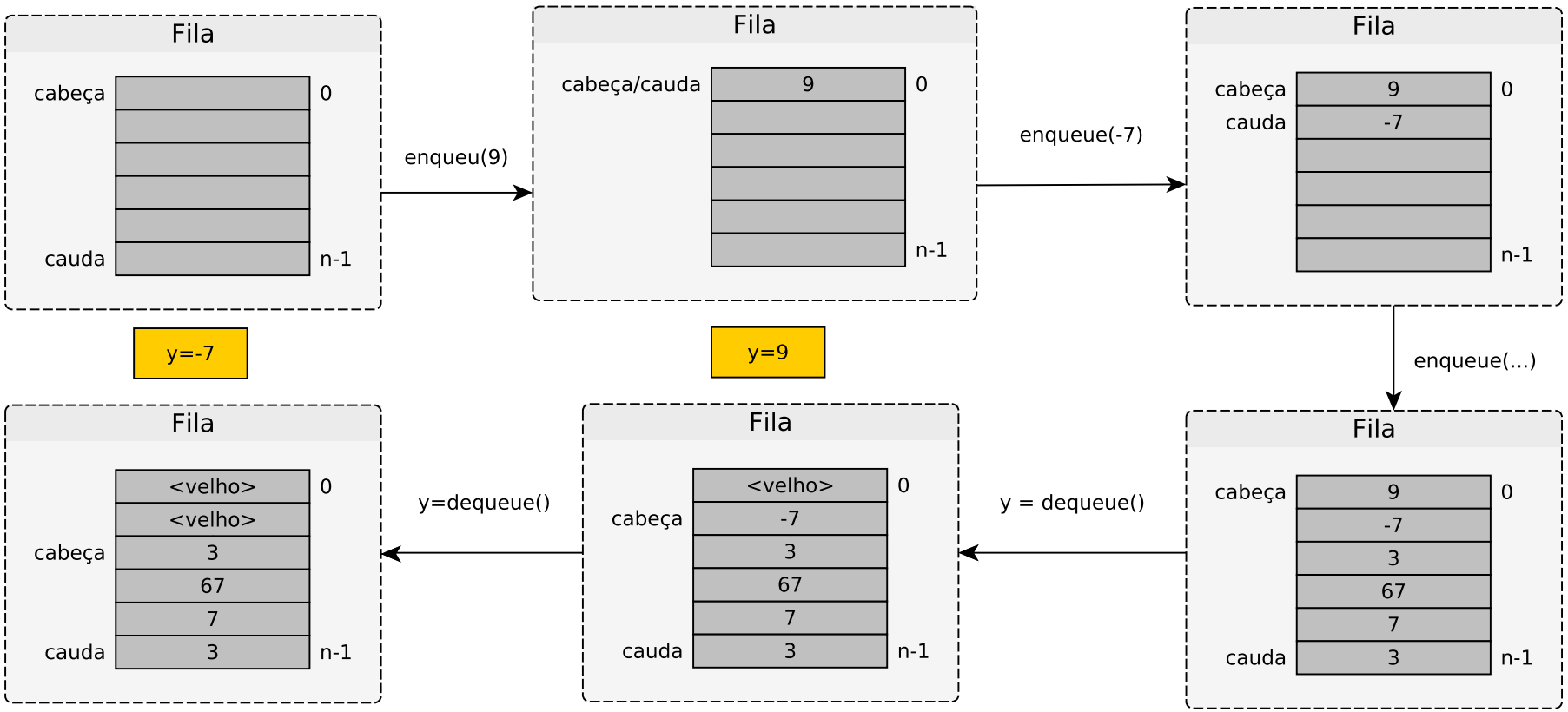
Filas

Fila – *queue* – é uma estrutura de dados simples onde os dados são "enfileirados". Idealmente implementa o conceito de FIFO – *First-in, First-out*: primeiro a entrar é o primeiro a sair.

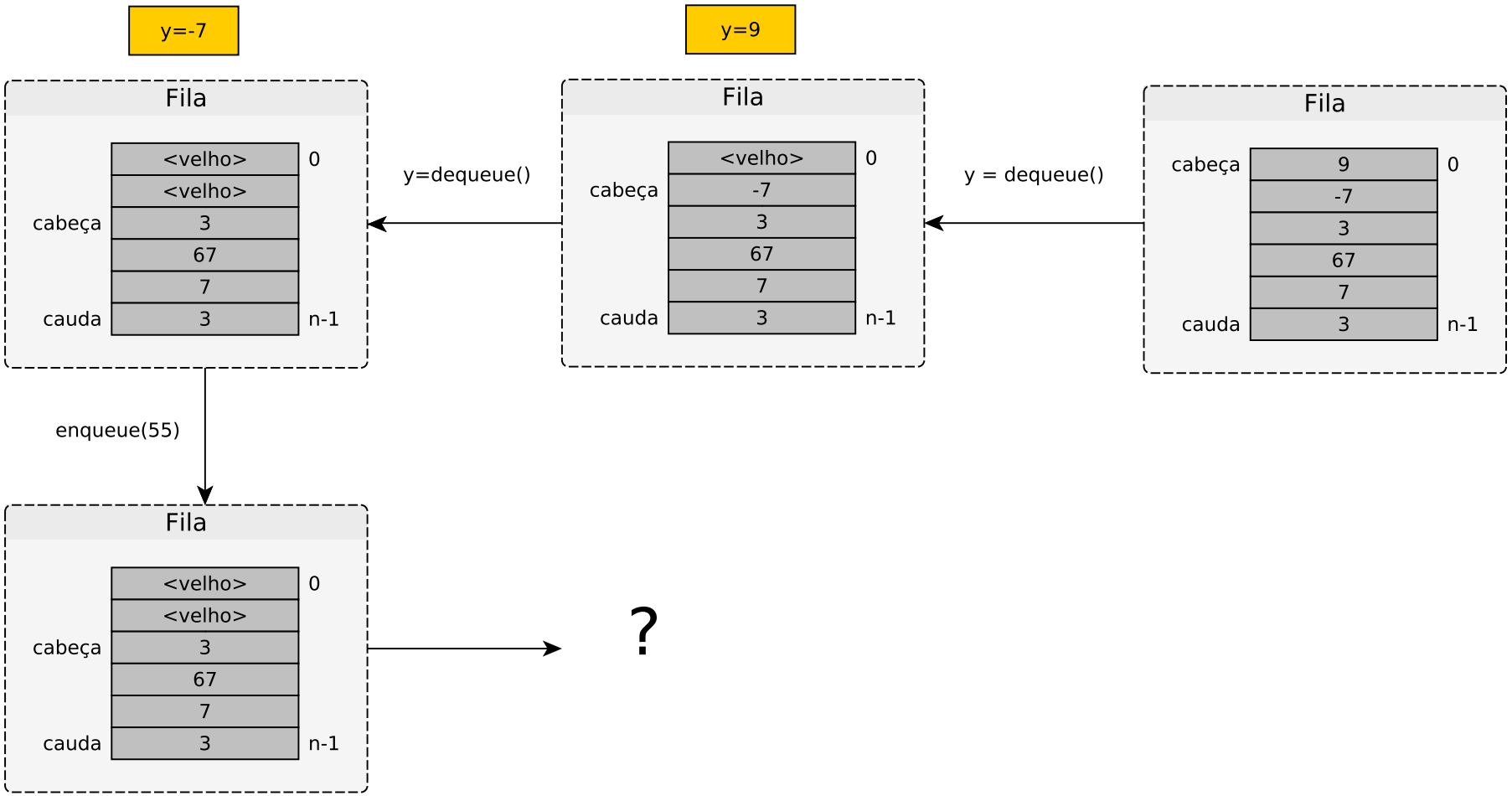
Funções básicas:

- *enqueue*: adiciona um elemento
- *dequeue*: remove um elemento

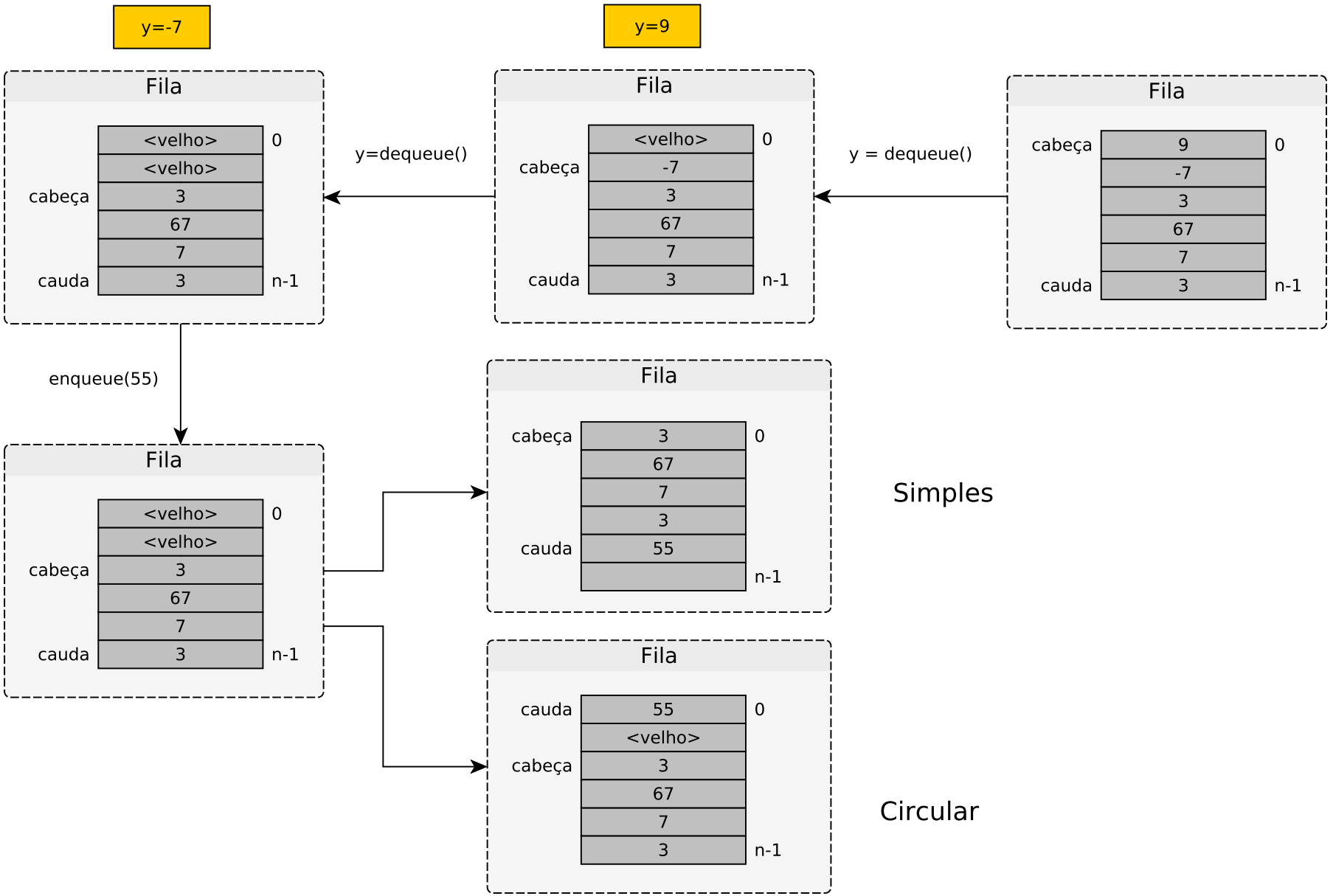
Filas


Fila
cabeça
cauda
0
n-1
enqueu(9)
Fila
cabeça/cauda
9
0
n-1
enqueue(-7)
Fila
cabeça
cauda
9
-7
0
n-1
enqueue(...)
y=-7
y=9
Fila
<velho>
<velho>
cabeça
3
67
7
cauda
3
0
n-1
y=dequeue()
Fila
<velho>
cabeça
-7
3
67
7
cauda
3
0
n-1
y = dequeue()
Fila
cabeça
9
-7
3
67
7
cauda
3
0
n-1

Filas


y=-7
Fila
<velho>
<velho>
cabeça
3
67
7
cauda
3
0
n-1
y=dequeue()
y=9
Fila
<velho>
cabeça
-7
3
67
7
cauda
3
0
n-1
y = dequeue()
Fila
cabeça
9
-7
3
67
7
cauda
3
0
n-1
enqueue(55)
Fila
<velho>
<velho>
cabeça
3
67
7
cauda
3
0
n-1
?

Filas


y=-7
y=9
Fila
<velho>
<velho>
3
67
7
3
cabeça
cauda
0
n-1
y=dequeue()
y = dequeue()
9
-7
enqueue(55)
55
Simples
Circular

Filas

Considerações com implementação com *array*:

- A região de elementos contínuos não necessariamente ocupará as posições inicias do *array*
  - Em algumas situações as posições "farão a volta".
  - Ou devemos deslocar todos os elementos.
- Dado um *array* de tamanho n, então:
  - $0 \leq cabeca < n$
  - $0 \leq cauda < n$
  - $0 \leq cabeca - cauda \leq n - 1$
  - Como há apenas n diferenças possíveis, poderá haver $0, 1, ...n - 1$ comprimentos de fila.
- Dado estas considerações, é impossível descobrir o tamanho da fila observando somente os valores da cabeça e cauda.
- Quando há somente um elemento: cauda = cabeça

Filas

Funções adicionais:

- *cabeca*: retorna o valor do elemento da cabeça da fila
- *imprimir*: imprimi, sem retirar, todos os elementos da fila
- *contem*: procura um item
- *tamanho*: retorna tamanho da fila
- *vazia*: retorna se fila está vazia

Tópico

Exemplo – fila.h

```c
#ifndef FILA_H_INCLUDED
#define FILA_H_INCLUDED

typedef struct filas fila_t;

int dequeue(fila_t* fila);
void enqueue(int x, fila_t* fila);


fila_t* cria_fila(int tamanho);
void libera_fila(fila_t* fila);


#endif // FILA_H_INCLUDED
```

Exemplo – fila.c: struct fila

```c
#include <stdio.h>
#include <stdlib.h>

#include "fila.h"

struct filas {
    int cabeca;
    int cauda;
    int tamanho;
    int tamanho_max;
    int *dados;
};
```

Exemplo – fila.c: cria_fila

```
//cria uma nova fila
fila_t* cria_fila(int tamanho)
{
    fila_t *p = NULL;

    //aloca memoria
    p = (fila_t*)malloc(sizeof(fila_t));


    //variaveis de controle
    p->cabeca = 0;
    p->cauda = tamanho-1;
    p->tamanho = 0;
    p->tamanho_max = tamanho;

    p->data = (int) malloc(sizeof(int) * tamanho);

    return p;
}
```

Exemplo – fila.c: libera

```c
//libera memoria
void libera_fila(fila_t* fila)
{
    fila_t *p = fila;

    if (p == NULL)
    {
        fprintf(stderr, "Erro desalocando memoria da fila, ponteiro invalido!");
        exit(EXIT_FAILURE);
    }


    free(p->dados);
    free(p);
}
```

Exemplo – fila.c: main

```c
int main()
{
    fila_t* fila = NULL;

    fila = cria_fila(10);


    enqueue(fila, 10)
    enqueue(fila, 45)
    enqueue(fila, 33)


    x = dequeue(fila)
    printf("%d\n", fila)
    x = dequeue(fila)
    x = dequeue(fila)


    //(...)

    libera_fila(fila);

    return 0;
}
```

Tópico

Exercício

Baseando-se na implementação exemplo desta aula, implemente uma fila através de uma lista simplesmente encadeada com as seguintes interfaces:

- *enqueue*: enfileirar
- *dequeue*: desenfileirar
- *topo*:retorna o valor do elemento do topo da fila
- *imprimir*: imprimi, sem retirar, todos os elementos da fila
- *tamanho*: retorna tamanho da fila
- *vazia*: retorna se a fila está vazia