



FUNDAÇÃO EDSON QUEIROZ  
UNIVERSIDADE DE FORTALEZA  
ENSINANDO E APRENDENDO

## T569 –SISTEMAS DE TEMPO REAL

---

# Aula 16

Prof. Marcelo Sousa



## Agenda

- Gerenciamento de filas
  - Como criar uma fila
  - Como uma fila gerencia os dados contidos
  - Como enviar dados para uma fila
  - Como receber dados de uma fila



# Filas no FreeRTOS

- Mecanismo de comunicação entre as tasks
- Um fila é uma primitiva de camada inferior utilizada pelo FreeRTOS para mecanismos de comunicação e sincronismo

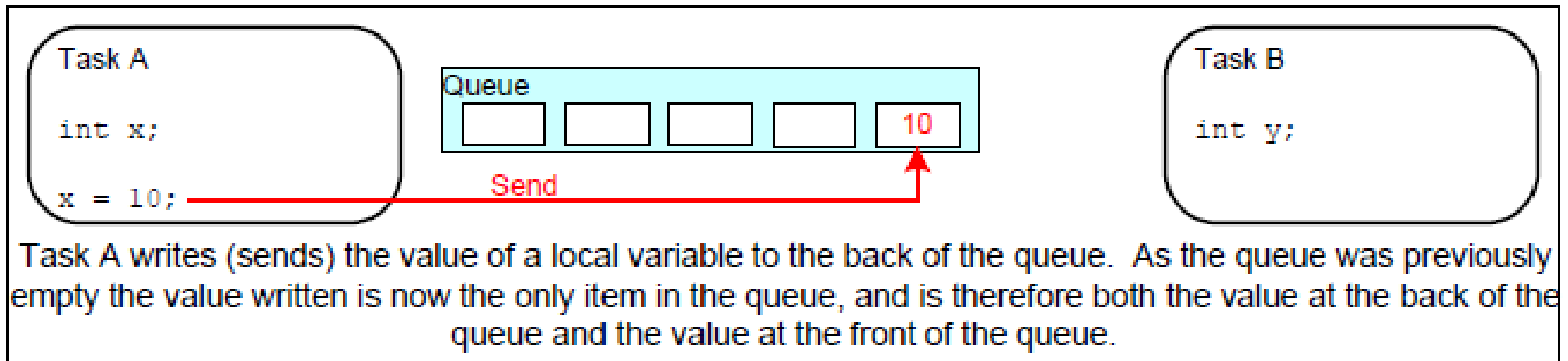
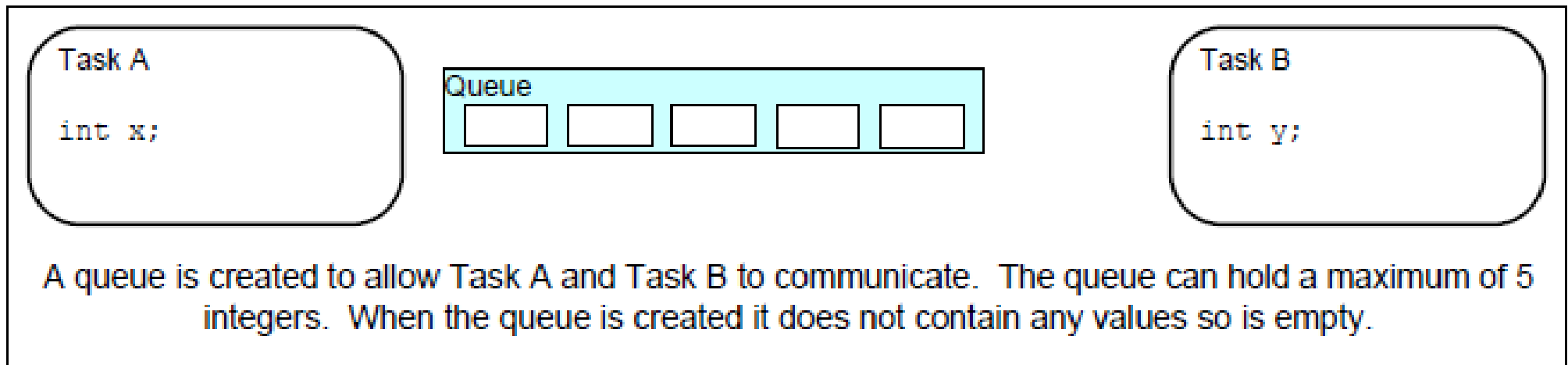


# Características de uma Fila

- Dados armazenados
  - Size: Tamanho dos itens armazenados
  - Length: Número máximo de itens armazenados
  - FIFO: *First In First Out*.
    - Itens inseridos no final da fila (tail) e removidos do início (head), podendo haver escrita no início.

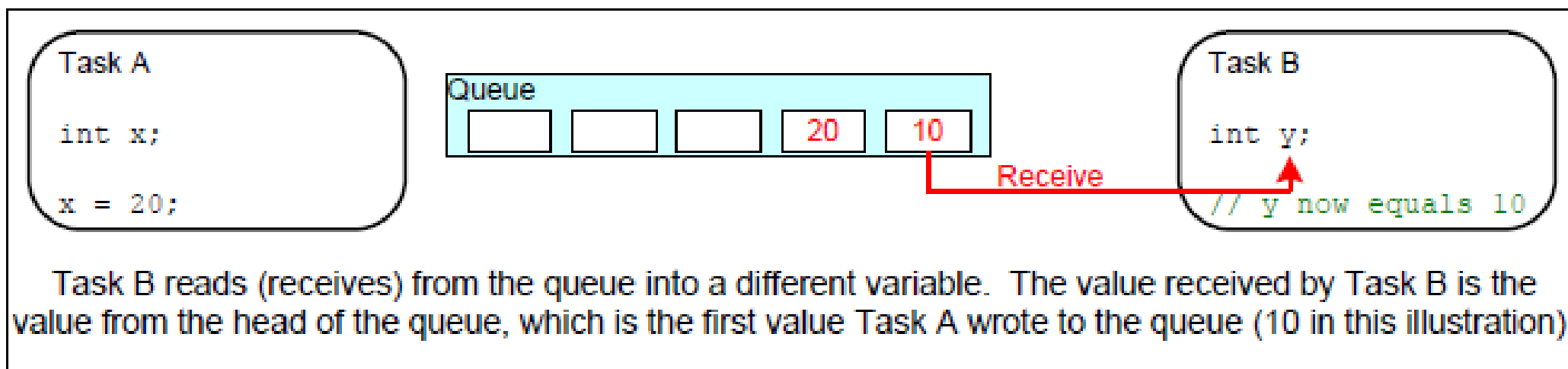
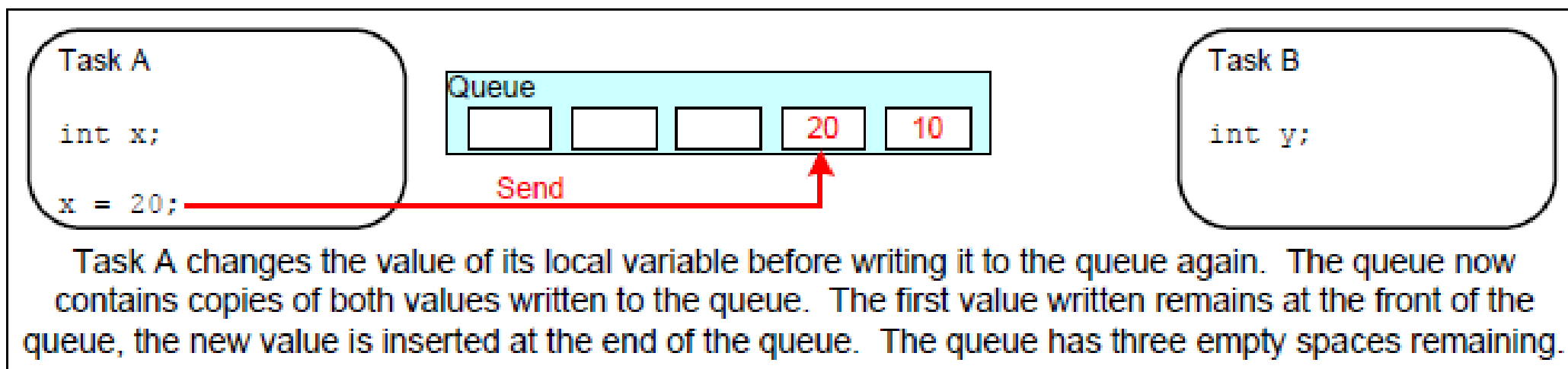


## Características de uma Fila





## Características de uma Fila





## Características de uma Fila

Task A

```
int x;  
  
x = 20;
```

Queue



Task B

```
int y;  
  
// y now equals 10
```

Task B has removed one item, leaving only the second value written by Task A remaining in the queue. This is the value Task B would receive next if it read from the queue again. The queue now has four empty spaces remaining.



## Características de uma Fila

- Acesso por múltiplas *tasks*
  - Uma fila não possui nenhuma associação a *tasks*, podendo ser utilizada por todas as *tasks* do sistemas
- Bloqueio em leituras da fila
  - Uma *task* que tenta realizar uma leitura pode especificar um tempo de espera caso não haja dados no buffer.
  - Se a fila estiver vazia a task é *bloqueada*
  - Fica em estado *ready* caso algum dado seja inserido na fila.





## Características de uma Fila

- Bloqueio em escritas na fila
  - Assim como na leitura, uma task pode configurar um tempo de espera para escrita caso a fila esteja cheia
  - A task irá para estado bloqueado caso a fila esteja cheia.



## Utilizando uma Fila

- *Criação da Task*
  - Uma task deve ser explicitamente criada antes de ser utilizada.
  - Função: `xQueueCreate()`
    - Criação da tarefa e retorna o valor do handle
  - Tipo: `xQueueHandle`
    - Tipo de variável utilizada para referenciar uma fila



# Utilizando uma Fila

- Escrita na Fila
  - xQueueSendToBack()
    - Escreve um dado na frente da fila (head)
  - xQueueSendToFront()
    - Escreve um dado atrás da fila (tail)
  - xQueueSend() = xSendToBack()
- PS: Nunca chame as funções de escrita em uma ISR.  
Versões *interrupt-safe* devem ser utilizadas:
  - xQueueSendToFrontFromISR() e xQueueSendToBackFromISR()<sup>11</sup>



# Utilizando uma Fila

- Leitura na Fila
  - xQueueReceive()
    - Utilizado para realizar a leitura na fila.
    - O item é removido da fila.
  - xQueuePeek()
    - Utilizado para realizar a leitura na fila.
    - O item não é removido da fila.
  - PS: Nunca chame as funções de leitura em uma ISR.  
Versões *interrupt-safe* devem ser utilizadas:
    - xQueueReceiveFromISR() e xQueuePeekFromISR()

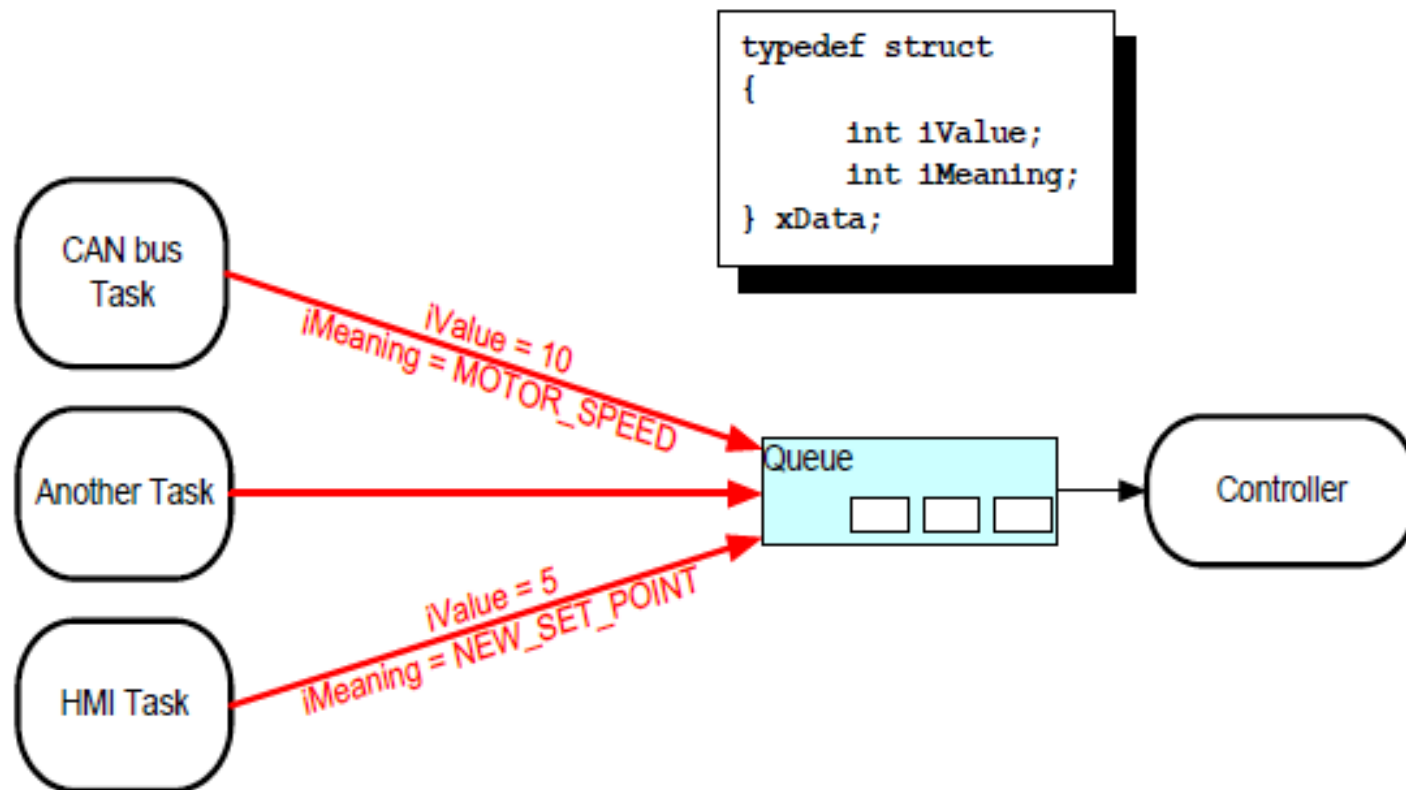


## Utilizando uma Fila

- Verificação do número de itens na fila
  - uxQueueMessagesWaiting()
    - Verifica o número de itens atuais na fila.
  - PS: Nunca chame essa função em uma ISR. Versões *interrupt-safe* devem ser utilizadas:
    - uxQueueMessagesWaitingFromISR()



## Filas com dados compostos





## Prática

- Exercício 10:
  - Implementar a criação de uma task
  - Envio de dados a partir de múltiplas tasks
  - Consumo dos dados apenas por uma única task.
  - Observações:
    - A task que envia para a fila não especifica um block time,
    - A task de leitura especifique um tick de 100ms.
    - As tasks de escrita possuem menor prioridade que a de leitura. Justifique.
    - Uma task de escrita deve escrever continuamente 100 e a outra 200. Utilize várias instancias da mesma task.
    - Utilize taskYIELD(), para passar o controle a outra task



## Prática

- Exercício 11:
  - Exemplo similar ao anterior, mas a task de recepção possui prioridade menor.
  - A fila é utilizada para passagem de estruturas, ao invés de um simples *long*.
  - A task devem possuir um tempo de espera de 100ms para inserir na fila
  - Nota: Neste exemplo a fila estará sempre cheia.