

# Modelagem em B do FreeRTOS

Stephenson Galvão, David Déharbe

Universidade Federal do Rio Grande do Norte  
Departamento de Informática e Matemática Aplicada  
Programa de Pos-Graduacao em Sistemas e Computacao  
Laboratório de Métodos Formais e Linguagem

Fevereiro/2009

Modelagem em  
B do FreeRTOS

Stephenson  
Galvão, David  
Déharbe

FreeRTOS

Características  
Principais Conceitos

Projeto de  
Modelagem

Modelagem

Introdução  
Desenvolvimento  
Task\_Core  
Task\_Basic  
Queue\_Core  
Queue\_Basic  
Queue  
Obrigações de prova

Considerações  
Finais

- 1 FreeRTOS
  - Características
  - Principais conceitos
- 2 Projeto de Modelagem
- 3 Modelagem
  - Introdução
  - Desenvolvimento
  - Obrigações de prova
- 4 Considerações Finais

Modelagem em  
B do FreeRTOS

Stephenson  
Galvão, David  
Déharbe

FreeRTOS

Características  
Principais Conceitos

Projeto de  
Modelagem

Modelagem

Introdução  
Desenvolvimento  
Task\_Core  
Task\_Basic  
Queue\_Core  
Queue\_Basic  
Queue  
Obrigações de prova

Considerações  
Finais

- 1 FreeRTOS
  - Características
  - Principais conceitos
- 2 Projeto de Modelagem
- 3 Modelagem
  - Introdução
  - Desenvolvimento
  - Obrigações de prova
- 4 Considerações Finais

## Modelagem em B do FreeRTOS

Stephenson  
Galvão, David  
Déharbe

## FreeRTOS

### Características

Principais Conceitos

## Projeto de Modelagem

## Modelagem

Introdução

Desenvolvimento

Task\_Core

Task\_Basic

Queue\_Core

Queue\_Basic

Queue

Obrigações de prova

## Considerações Finais

- 1 FreeRTOS
  - Características
  - Principais conceitos
- 2 Projeto de Modelagem
- 3 Modelagem
  - Introdução
  - Desenvolvimento
  - Obrigações de prova
- 4 Considerações Finais

## Modelagem em B do FreeRTOS

Stephenson  
Galvão, David  
Déharbe

## FreeRTOS

### Características

Principais Conceitos

## Projeto de Modelagem

## Modelagem

Introdução

Desenvolvimento

Task\_Core

Task\_Basic

Queue\_Core

Queue\_Basic

Queue

Obrigações de prova

## Considerações Finais

- Trabalha em ambientes com recursos limitados
- Acessível
- Simples
- Popular
- Fácil de usar
- Portável

Modelagem em  
B do FreeRTOS

Stephenson  
Galvão, David  
Déharbe

FreeRTOS

Características

Principais Conceitos

Projeto de  
Modelagem

Modelagem

Introdução

Desenvolvimento

Task\_Core

Task\_Basic

Queue\_Core

Queue\_Basic

Queue

Obrigações de prova

Considerações  
Finais

- 1 FreeRTOS
  - Características
  - Principais conceitos
- 2 Projeto de Modelagem
- 3 Modelagem
  - Introdução
  - Desenvolvimento
  - Obrigações de prova
- 4 Considerações Finais

## Modelagem em B do FreeRTOS

Stephenson  
Galvão, David  
Déharbe

## FreeRTOS

Características

Principais Conceitos

## Projeto de Modelagem

## Modelagem

Introdução

Desenvolvimento

Task\_Core

Task\_Basic

Queue\_Core

Queue\_Basic

Queue

Obrigações de prova

## Considerações Finais

- Tarefa
- Co-rotina
- Fila de mensagens
- Semáforo
- Mutex

## Definição

Unidade de trabalho independente que compoem um sistema

- Características:
  - Estado - (executando, pronto, bloqueado e suspenso);
  - Prioridade - (0..*configMAX\_PRIORITIES*);
  - Função de execução;
  - Pilha de contexto
- Realiza escalonamento preemptivo



## Definição

Semelhante a tarefa, são unidades de trabalho indepedentes que compartilha o mesmo contexto

- Características:
  - Estado - (executando, pronto e bloqueado);
  - Prioridade - (0..*configMAX\_PRIORITIES*);
  - Função de execução
- Não possuem pilha de contexto
- Realiza escalonamento cooperativo

## Definição

Estrutura primitiva de comunicação entre as tarefas. Através dela tarefas enviam e recebem mensagem

- Características:
  - Lista de itens;
  - Tamanho do item ;
  - Quantidade de itens;
  - Listas de eventos para tarefas que:
    - Aguardam para enviar uma mensagem para fila
    - Aguardam para receber uma mensagem da fila
- Permite definir o tempo máximo que uma tarefa deve aguardar por uma fila

## Definição

Objeto do kernel que uma ou mais tarefas podem solicitar ou liberar com o proposito de sincronizar-se.

## Utilização

Quando uma tarefa deseja utilizar um recurso compartilhando, ela primeiramente solicita o semáforo desse recurso e, após o uso do recurso,liberar o semáforo.

## Analogia

São semelhantes a uma fila de um único item, sendo que fila cheia indica semáforo liberado e fila vazia indica semáforo em uso.

## Definição

Um semáforos com o mecanismo de herança de prioridade. São utilizados na implementação da exclusão mútua.

- Herança de prioridade:
  - Eleva temporariamente a prioridade da tarefa com o semáforo
  - Evita a inversão de prioridade - Bloqueio da tarefa mais prioritária por uma de menor prioridade
  - Exemplo - Quando a tarefa mais prioritária é bloqueada ao solicitar o semáforo ocupado por uma menos prioritária

Modelagem em  
B do FreeRTOS

Stephenson  
Galvão, David  
Déharbe

FreeRTOS

Características  
Principais Conceitos

Projeto de  
Modelagem

Modelagem

Introdução  
Desenvolvimento  
Task\_Core  
Task\_Basic  
Queue\_Core  
Queue\_Basic  
Queue  
Obrigações de prova

Considerações  
Finais

## 1 FreeRTOS

- Características
- Principais conceitos

## 2 Projeto de Modelagem

## 3 Modelagem

- Introdução
- Desenvolvimento
- Obrigações de prova

## 4 Considerações Finais

- Conceitos
  - Tarefa
  - Fila de mensagens
- Conjunto de funcionalidades
  - Criação de tarefa
  - Controle de tarefa
  - Utilidade de tarefa
  - Controle do kernel
  - Gerenciamento de fila de mensagens

## ■ Criação de tarefa:

- *vTaskCreate*
- *vTaskDelete*
- *xTaskHandle*

## ■ Controle de tarefa:

- *vTaskSuspend*
- *vTaskResume*
- *uxTaskPriorityGet*
- *vTaskPrioritySet*
- *vTaskDelay*
- *vTaskDelayUntil*

## ■ Utilidades de tarefa:

- *xTaskGetCurrentTaskHandle*
- *xTaskGetSchedulerState*
- *uxTaskGetNumberOfTasks*
- *xTaskGetTickCount*

## ■ Controle do kernel:

- *vTaskStartScheduler*
- *vTaskEndScheduler*
- *vTaskSuspendAll*
- *xTaskResumeAll*

## ■ Gerenciamento de fila:

- *xQueueCreate*
- *xQueueSend*
- *xQueueSendToBack*
- *xQueueSendToFront*
- *xQueueReceive*
- *xQueuePeek*
- *vQueueDelete*

## Modelagem em B do FreeRTOS

Stephenson  
Galvão, David  
Déharbe

## FreeRTOS

Características  
Principais Conceitos

## Projeto de Modelagem

## Modelagem

Introdução  
Desenvolvimento  
Task\_Core  
Task\_Basic  
Queue\_Core  
Queue\_Basic  
Queue  
Obrigações de prova

## Considerações Finais

- 1 FreeRTOS
  - Características
  - Principais conceitos
- 2 Projeto de Modelagem
- 3 Modelagem
  - Introdução
  - Desenvolvimento
  - Obrigações de prova
- 4 Considerações Finais



## Modelagem em B do FreeRTOS

Stephenson  
Galvão, David  
Déharbe

## FreeRTOS

Características  
Principais Conceitos

## Projeto de Modelagem

## Modelagem

Introdução  
Desenvolvimento  
Task\_Core  
Task\_Basic  
Queue\_Core  
Queue\_Basic  
Queue  
Obrigações de prova

## Considerações Finais

- 1 FreeRTOS
  - Características
  - Principais conceitos
- 2 Projeto de Modelagem
- 3 Modelagem
  - Introdução
  - Desenvolvimento
  - Obrigações de prova
- 4 Considerações Finais

- Dividida basicamente em 5 máquinas:
  - Task\_Core
  - Task\_Basic
  - Queue\_Core
  - Queue\_Basic
  - Queue
- Passagem de parâmetros por referência
- Alto nível de abstração

## Modelagem em B do FreeRTOS

Stephenson  
Galvão, David  
Déharbe

## FreeRTOS

Características  
Principais Conceitos

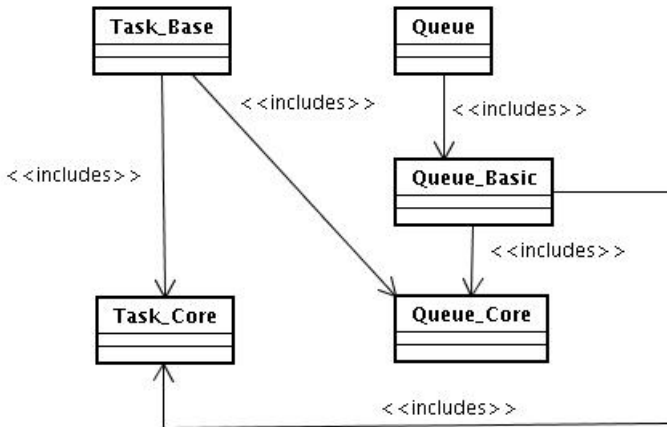
## Projeto de Modelagem

## Modelagem

Introdução  
**Desenvolvimento**  
Task\_Core  
Task\_Basic  
Queue\_Core  
Queue\_Basic  
Queue  
Obrigações de prova

## Considerações Finais

- 1 FreeRTOS
  - Características
  - Principais conceitos
- 2 Projeto de Modelagem
- 3 Modelagem**
  - Introdução
  - Desenvolvimento**
  - Obrigações de prova
- 4 Considerações Finais



- Especifica o conceito de tarefa
  - Característica especificada:
    - Estado
  - Características abstraídas:
    - Prioridade
    - Função de execução
    - Pilha de contexto
- Especifica operações que:
  - Gerenciam as características de uma tarefa (Estado)
  - Preocupam-se principalmente como a troca de estado
  - Servem de base para especificação das funcionalidades da API

## SETS

*TASK;*

...

## VARIABLES

*tasks,*  
*blocked,*  
*running,*  
*ready,*  
*suspended*

...

## INVARIANT

*tasks*  $\in$  **FIN**(*TASK*)  $\wedge$   
*tasks*  $\neq \emptyset \wedge$

*blocked*  $\in$  **FIN**(*TASK*)  $\wedge$   
*blocked*  $\subset$  *tasks*  $\wedge$

*ready*  $\in$  **FIN**(*TASK*)  $\wedge$   
*ready*  $\subset$  *tasks*  $\wedge$

*suspended*  $\in$  **FIN**(*TASK*)  $\wedge$   
*suspended*  $\subset$  *tasks*  $\wedge$   
*running*  $\in$  *TASK*  $\wedge$

...

Modelagem em  
B do FreeRTOS

Stephenson  
Galvão, David  
Déharbe

FreeRTOS

Características

Principais Conceitos

Projeto de  
Modelagem

Modelagem

Introdução

Desenvolvimento

**Task\_Core**

Task\_Basic

Queue\_Core

Queue\_Basic

Queue

Obrigações de prova

Considerações  
Finais

- Create
- Delete
- Suspend
- Resume
- GetPriority
- GetCurrent
- GetNumberOfTasks
- DelayTask
- Unblock
- StartScheduler
- EndScheduler
- ResumeAll

## Modelagem em B do FreeRTOS

Stephenson  
Galvão, David  
Déharbe

## FreeRTOS

Características  
Principais Conceitos

## Projeto de Modelagem

## Modelagem

Introdução  
Desenvolvimento  
**Task\_Core**  
Task\_Basic  
Queue\_Core  
Queue\_Basic  
Queue  
Obrigações de prova

## Considerações Finais

- **Create**
- Delete
- Suspend
- Resume
- GetPriority
- GetCurrent
- GetNumberOfTasks
- DelayTask
- Unblock
- StartScheduler
- EndScheduler
- ResumeAll



```

result  $\leftarrow$  Create(priority) =
PRE
priority  $\in$  PRIORITY
THEN
  ANY
    task
  WHERE
    task  $\in$  TASK  $\wedge$ 
    task  $\notin$  tasks
  ...

```

```

THEN
  tasks := {task}  $\cup$  tasks ||
  CHOICE
    ready := {running}  $\cup$  ready ||
    running := task
  OR
    ready := task  $\cup$  ready
  END ||
  result := task
END
END;

```

- Especifica característica do escalonador:
  - Estado
  - Tempo de execução
- Especificam operações que:
  - Compoem os seguintes conjunto de funcionalidade:
    - Criação de tarefa
    - Controle de tarefa
    - Utilidades de tarefa
    - Controle do Kernel
  - Utilizam-se da máquina *Task\_Core*

## SETS

$$SCHEDULER\_STATE = \{taskSCHEDULER\_NOT\_STARTED, \\ taskSCHEDULER\_RUNNING, taskSCHEDULER\_SUSPENDED\}$$

## VARIABLES

*tickCount*,  
*tickMissed*,  
*scheduler*

## INVARIANT

$tickCount \in TICK \wedge$   
 $tickMissed \in TICK \wedge$   
 $scheduler \in SCHEDULER\_STATE$

## ■ Criação de tarefa

- xTaskCreate
- vTaskDelete

## ■ Controle de tarefa

- vTaskSuspend
- vTaskResume
- uxTaskPriorityGet
- uxTaskPrioritySet
- vTaskDelay
- vTaskDelayUntil

## ■ Utilidades de tarefas

- xTaskGetCurrentTaskHandle
- uxTaskGetNumberOfTasks
- xTaskGetSchedulerState
- xTaskGetTickCount

## ■ Gerenciamento do Kernel

- vTaskStartScheduler
- vTaskEndScheduler
- vTaskSuspendAll
- xTaskResumeAll
- incrementTick

## ■ Criação de tarefa

- **xTaskCreate**
- vTaskDelete

## ■ Controle de tarefa

- vTaskSuspend
- vTaskResume
- uxTaskPriorityGet
- uxTaskPrioritySet
- vTaskDelay
- vTaskDelayUntil

## ■ Utilidades de tarefas

- xTaskGetCurrentTaskHandle
- uxTaskGetNumberOfTasks
- xTaskGetSchedulerState
- xTaskGetTickCount

## ■ Gerenciamento do Kernel

- vTaskStartScheduler
- vTaskEndScheduler
- vTaskSuspendAll
- xTaskResumeAll
- incrementTick

```

result, handle  $\leftarrow$ 
    xTaskCreate(pvTaskCode,
    pcName, usStackDepth,
    pvParameters, uxPriority) =
PRE
    pvTaskCode  $\in$  TASK_CODE  $\wedge$ 
    pcName  $\in$  NAME  $\wedge$ 
    usStackDepth  $\in$   $\mathbb{N}$ 
    pvParameters  $\subset$  PARAMETER
    uxPriority  $\in$  PRIORITY

```

...

**THEN**

**CHOICE**

```

    handle  $\leftarrow$  Create(uxPriority) ||
    result := pdPASS

```

**OR**

```

    result := errCOULD_NOT_ALLOCATE
    handle := TASK_NULL

```

**END**

**END;**

- Especifica a conceito de fila de mensagens
  - Características especificadas:
    - Conjunto de itens
    - Fila de evento para envio de mensagem
    - Fila de evento para o recebimento de mensagem
  - Características abstraídas
    - Tamanho do item
    - Quantidade de item
- Possui um conjunto de operações que:
  - Gerenciam as característica especificadas
  - Preocupa-se principalmente com:
    - Inserção e remoção de um item na fila
    - Inserção e remoção de uma tarefa na lista de evento
  - Server como base para a especificação da API

## SETS

*ITEM;*  
*COPY\_POSITION;*  
*QUEUE*

## VARIABLES

*queues,*  
*queue\_items,*  
*queue\_receiving,*  
*queue\_sending*

## INVARIANT

*queues*  $\in \mathbb{P}(\text{QUEUE}) \wedge$   
*queue\_items*  $\in \text{QUEUE} \rightarrow \mathbb{P}(\text{ITEM}) \wedge$   
*queue\_receiving*  $\in \text{QUEUE} \rightarrow \mathbb{P}(\text{TASK}) \wedge$   
*queue\_sending*  $\in \text{QUEUE} \rightarrow \mathbb{P}(\text{TASK}) \wedge$   
*queues* =  $\text{dom}(\text{queue\_items}) \wedge$   
*queues* =  $\text{dom}(\text{queue\_receiving}) \wedge$   
*queues* =  $\text{dom}(\text{queue\_sending})$



## Modelagem em B do FreeRTOS

Stephenson  
Galvão, David  
Déharbe

## FreeRTOS

Características

Principais Conceitos

## Projeto de Modelagem

## Modelagem

Introdução

Desenvolvimento

Task\_Core

Task\_Basic

Queue\_Core

Queue\_Basic

Queue

Obrigações de prova

## Considerações Finais

- xQueueCreate
- sendItem
- insertTaskWaitingToSend
- insertTaskWaitingToRecived
- receivedItem
- removeFromEventListQueue

## Modelagem em B do FreeRTOS

Stephenson  
Galvão, David  
Déharbe

## FreeRTOS

Características  
Principais Conceitos

## Projeto de Modelagem

## Modelagem

Introdução  
Desenvolvimento  
Task\_Core  
Task\_Basic  
Queue\_Core  
Queue\_Basic  
Queue  
Obrigações de prova

## Considerações Finais

- **xQueueCreate**
- sendItem
- insertTaskWaitingToSend
- insertTaskWaitingToRecived
- receivedItem
- removeFromEventListQueue

...

**CHOICE**

**ANY**

*queue*

**WHERE**

*queue*  $\in$  *QUEUE*  $\wedge$

*queue*  $\notin$  *queues*  $\wedge$

**THEN**

*queues* := *queues*  $\cup$  {*queue*} ||

*queue\_items* := *queue\_items*  $\cup$  {*queue*  $\mapsto$   $\emptyset$ } ||

*queue\_receiving* := *queue\_receiving*  $\cup$  {*queue*  $\mapsto$   $\emptyset$ } ||

*queue\_sending* := *queue\_sending*  $\cup$  {*queue*  $\mapsto$   $\emptyset$ } ||

*xQueueHandle* := *queue*

**END**

**OR**

*xQueueHandle* := *QUEUE\_NULL*

...

- Especifica:
  - Comportamentos comuns da API Gerenciamento de tarefa
  - Operações mais complexa que as presente em *Queue\_Core*
- Formada por duas operações:
  - xQueueGenericSend
    - Envia um item para uma fila e desbloqueia uma tarefa que aguarda pela chegada de um item
    - Coloca a tarefa atual no lista de evento de tarefa que esperam para enviar um item para fila
    - Retorna uma mensagem de erro
  - xQueueGenericReceive
    - Envia um item para uma fila e desbloqueia uma tarefa que aguarda pela chegada de um item
    - Coloca a tarefa atual no lista de evento de tarefa que esperam para enviar um item para fila
    - Retorna uma mensagem de erro

- Especifica as operações da API Gerenciamento de Fila
- Basicamente utiliza-se das funções *QueueGenericReceive* e *xQueueGenericSend*

## xQueueSendToBack

...

**ANY** *copy\_position*

**WHERE**

*copy\_position*  $\in$  *COPY\_POSITION*

**THEN**

*return*  $\leftarrow$  *xQueueGenericSend*(*pxQueue*,  
*pvlItemToQueue*, *xTicksToWait*, *copy\_position*)

**END**

...

## Modelagem em B do FreeRTOS

Stephenson  
Galvão, David  
Déharbe

## FreeRTOS

Características  
Principais Conceitos

## Projeto de Modelagem

## Modelagem

Introdução  
Desenvolvimento  
Task\_Core  
Task\_Basic  
Queue\_Core  
Queue\_Basic  
Queue  
Obrigações de prova

## Considerações Finais

- 1 FreeRTOS
  - Características
  - Principais conceitos
- 2 Projeto de Modelagem
- 3 Modelagem
  - Introdução
  - Desenvolvimento
  - Obrigações de prova
- 4 Considerações Finais

<b>Máquinas</b>	<b>N<sup>a</sup> Ob.</b>	<b>N<sup>a</sup> P.A.</b>	<b>N<sup>a</sup> P.I.</b>
<i>Task_Core</i>	179	161	18
<i>Task_Base</i>	16	16	0
<i>Queue_Core</i>	27	27	0
<i>Queue_Basic</i>	2	2	0
<i>Queue</i>	0	0	0

## Modelagem em B do FreeRTOS

Stephenson  
Galvão, David  
Déharbe

## FreeRTOS

Características  
Principais Conceitos

## Projeto de Modelagem

## Modelagem

Introdução  
Desenvolvimento  
Task\_Core  
Task\_Basic  
Queue\_Core  
Queue\_Basic  
Queue  
Obrigações de prova

## Considerações Finais

- 1 FreeRTOS
  - Características
  - Principais conceitos
- 2 Projeto de Modelagem
- 3 Modelagem
  - Introdução
  - Desenvolvimento
  - Obrigações de prova
- 4 Considerações Finais



## Modelagem em B do FreeRTOS

Stephenson  
Galvão, David  
Déharbe

## FreeRTOS

Características  
Principais Conceitos

## Projeto de Modelagem

## Modelagem

Introdução  
Desenvolvimento  
Task\_Core  
Task\_Basic  
Queue\_Core  
Queue\_Basic  
Queue  
Obrigações de prova

## Considerações Finais

- Modelagem precária
- Dificuldades em especificar as características dos conceitos
- Necessidade de formalismo que possam:
  - Especificar a concorrência entre as tarefas
  - Verificar a memória heap

## Modelagem em B do FreeRTOS

Stephenson  
Galvão, David  
Déharbe

## FreeRTOS

Características  
Principais Conceitos

## Projeto de Modelagem

## Modelagem

Introdução  
Desenvolvimento  
Task\_Core  
Task\_Basic  
Queue\_Core  
Queue\_Basic  
Queue  
Obrigações de prova

## Considerações Finais

- Página do projeto
  - <http://code.google.com/p/freertosb/>
- E-mails:
  - Stephenson - [stepgalvao@gmail.com](mailto:stepgalvao@gmail.com)
  - David - [deharbe@gmail.com](mailto:deharbe@gmail.com)