Desenvolvimento de Sistemas Embarcados em Tempo Real

Prof. Hermano Cabral

Departmento de Eletrônica e Sistemas — UFPE

4 de dezembro de 2017

Plano de Aula

Tema central

- ChibiOS — sistema operacional

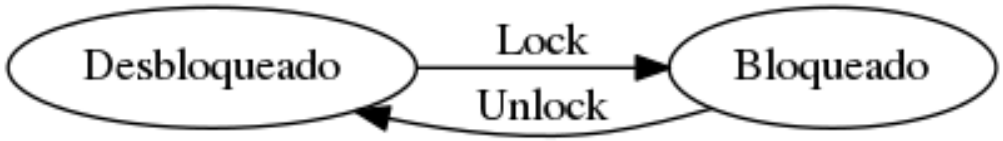
Plano de Aula

Tema central

- ChibiOS — sistema operacional

Objetivos

- Conhecer as características dos mutexes, variáveis de condição e eventos do RT, sistema operacional do ChibiOS.



Introdução

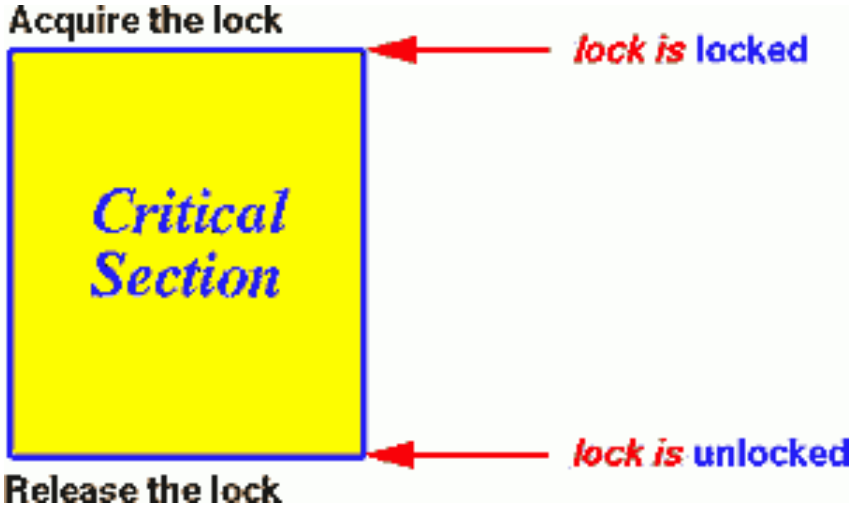
- Mutex é um objeto de sincronização de threads que pode estar em 2 estados distintos:
  - Desbloqueado
  - Bloqueado (adquirido por uma thread)

Operação

- Um mutex se torna bloqueado quando uma thread o adquire.

Operação

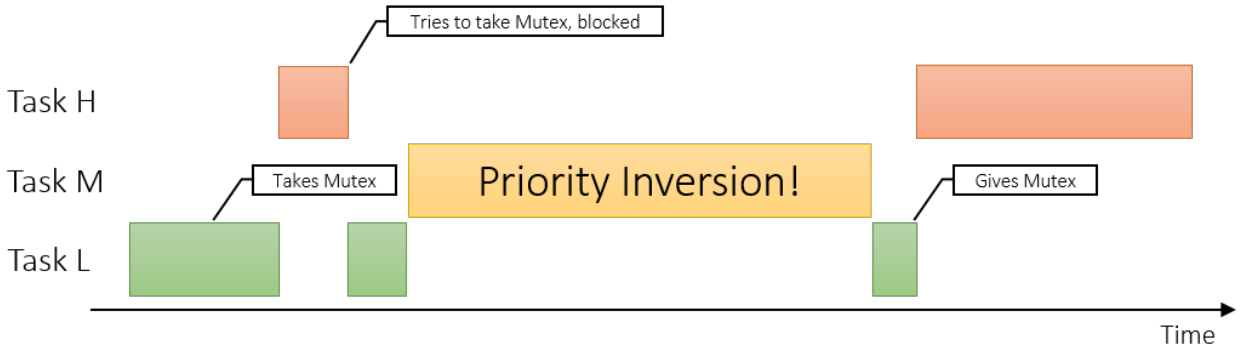
- Um mutex se torna bloqueado quando uma thread o adquire.
- Uma thread que tenta adquirir um mutex bloqueado é posta para dormir até o mutex ser liberado e ChibiOS escolhê-la para adquirir o mutex.



Observações

- Mutexes são muito usados para controlar o acesso a algum recurso ou seção crítica.

Kernel RT — mutexes



Observações

- A implementação de mutex de ChibiOS evita o problema acima de inversão de prioridade.

Observações

- Mutexes em ChibiOS são não-recursivos a não ser que ChibiOS seja compilado com a opção CH_CFG_USE_MUTEXES_RECURSIVE.

Operação

- Em ChibiOS, criamos um mutex do tipo mutex_t com o nome "name" através da função MUTEX_DECL(name).

Operação

- Em ChibiOS, criamos um mutex do tipo mutex_t com o nome "name" através da função MUTEX_DECL(name).
- Depois, usamos as funções chMtxLock() e chMtxUnlock() but adquirir ou liberar um mutex, passndo sempre um ponteiro para um mutex.

Kernel RT — eventos

Introdução

- ChibiOS implementa o conceito de eventos, onde uma thread pode esperar por um evento ser sinalizado por outra thread.

Introdução

- ChibiOS implementa o conceito de eventos, onde uma thread pode esperar por um evento ser sinalizado por outra thread.
- Um evento pode ter múltiplas threads esperando por ele.

Introdução

- ChibiOS implementa o conceito de eventos, onde uma thread pode esperar por um evento ser sinalizado por outra thread.
- Um evento pode ter múltiplas threads esperando por ele.
- Uma thread pode esperar por múltiplos eventos, embora possa sinalizar um evento apenas de cada vez.

Operação

- Uma thread possui uma lista de eventos pendentes e uma lista de eventos nos quais está esperando dormindo.

Operação

- Uma thread possui uma lista de eventos pendentes e uma lista de eventos nos quais está esperando dormindo.
- Um evento torna-se pendente quando a thread é sinalizada com ele através da função chEvtSignal(thread_t *tp, eventmask_t events).

Operação

- Uma thread possui uma lista de eventos pendentes e uma lista de eventos nos quais está esperando dormindo.
- Um evento torna-se pendente quando a thread é sinalizada com ele através da função chEvtSignal(thread_t *tp, eventmask_t events).
- Uma thread pode esperar um ou mais eventos se tornarem pendentes através das funções chEvtWaitOne (eventmask_t events), chEvtWaitAny (eventmask_t events) ou chEvtWaitAll (eventmask_t events).

Operação

- Uma vez que os eventos que está esperando aconteçam, a thread é acordada.

Operação

- Uma vez que os eventos que está esperando aconteçam, a thread é acordada.
- A thread pode recuperar os eventos que se tornaram pendentes com a função chEvtGetAndClearEvents(eventmask_t events).

Operação

- Existem 2 tipos mais complexos de operação:
  - Registro de um monitor em uma fonte de evento

Operação

- Existem 2 tipos mais complexos de operação:
  - Registro de um monitor em uma fonte de evento
  - Broadcast de um evento

Conceitos

- Existem 3 tipos de objetos envolvidos no tratamento de eventos:
  - Fontes de eventos: objetos que transmitem eventos para o resto do sistema

Conceitos

- Existem 3 tipos de objetos envolvidos no tratamento de eventos:
  - Fontes de eventos: objetos que transmitem eventos para o resto do sistema
  - Flags de eventos: informação extra associada ao evento

Conceitos

- Existem 3 tipos de objetos envolvidos no tratamento de eventos:
  - Fontes de eventos: objetos que transmitem eventos para o resto do sistema
  - Flags de eventos: informação extra associada ao evento
  - Monitores de eventos: objetos interessados em saber quando eventos ocorrem

Operação

- Uma thread pode se associar a um ou mais monitores, e registrá-los a uma ou mais fontes de eventos através da função chEvtRegisterMaskWithFlags (event_source_t *esp, event_listener_t *elp, eventmask_t events, eventflags_t wflags).

Operação

- Uma thread pode se associar a um ou mais monitores, e registrá-los a uma ou mais fontes de eventos através da função chEvtRegisterMaskWithFlags (event_source_t *esp, event_listener_t *elp, eventmask_t events, eventflags_t wflags).
- Uma thread pode propagar um evento sinalizando uma fonte de eventos através da função chEvtBroadcastFlags (event_source_t *esp, eventflags_t flags).