

Desenvolv. de Sistemas Embarcados em Tempo Real

Prof. Hermano Cabral

Departamento de Eletrônica e Sistemas — UFPE

9 de julho de 2024

Tema central

- Sistemas operacionais de tempo real

Tema central

- Sistemas operacionais de tempo real

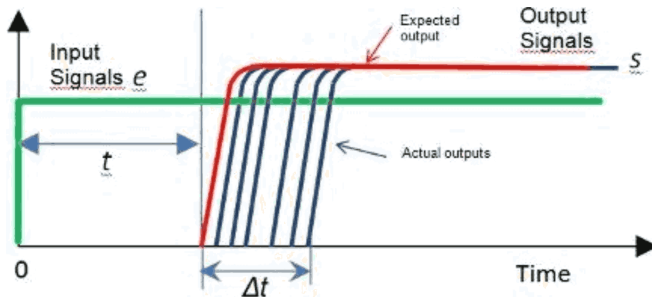
Objetivos

- Conhecer as características de sistemas operacionais de tempo real (RTOS)

Introdução

- Um sistema de tempo real é um sistema que não só é correto funcionalmente mas também temporalmente.

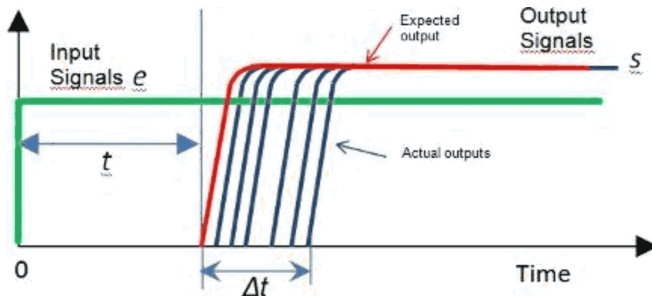
Sistemas de tempo real



Introdução

- O sistema deve ter um tempo de resposta máximo determinístico.

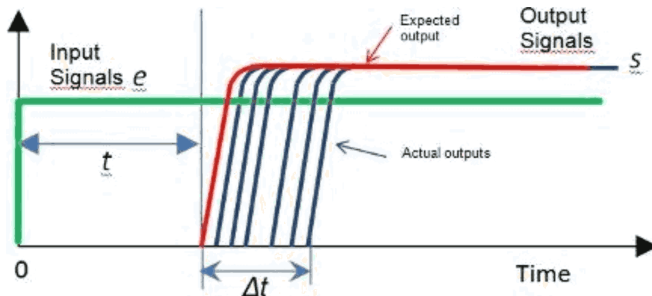
Sistemas de tempo real



Introdução

- O sistema deve ter um tempo de resposta máximo determinístico.
- Um parâmetro importante é o *jitter*, que é a variação no tempo de resposta entre o melhor e o pior caso.

Sistemas de tempo real



Introdução

- O sistema deve ter um tempo de resposta máximo determinístico.
- Um parâmetro importante é o *jitter*, que é a variação no tempo de resposta entre o melhor e o pior caso.
- Em geral, quanto menor o *jitter*, melhor.

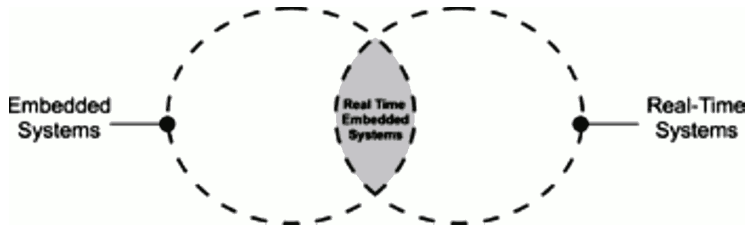
Criticidade

- Um sistema de tempo real *soft* tem sua operação **degradada** se as restrições de tempo não forem obedecidas.

Criticidade

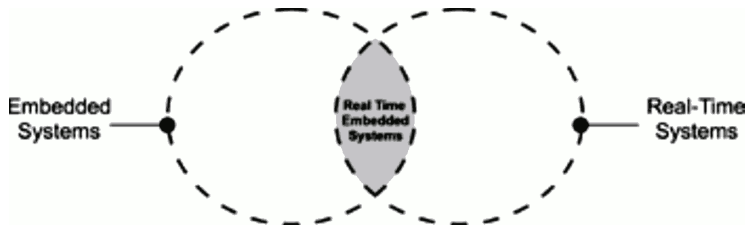
- Um sistema de tempo real *soft* tem sua operação **degradada** se as restrições de tempo não forem obedecidas.
- Um sistema de tempo real *hard* tem sua operação **incorreta** se as restrições de tempo não forem obedecidas.

Sistemas de tempo real



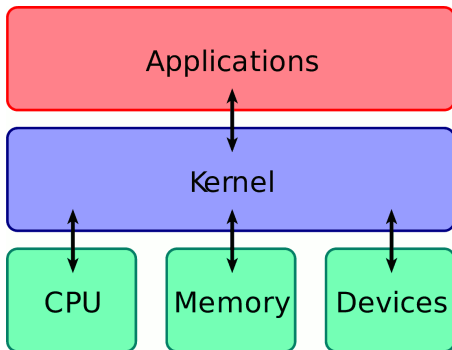
Observações

- Um sistema embarcado não precisa ser de tempo real.



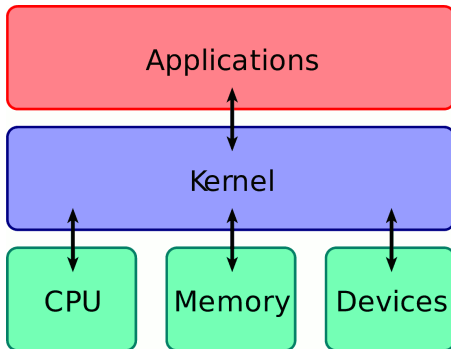
Observações

- Um sistema embarcado não precisa ser de tempo real.
- Além disso, um sistema de tempo real não precisa ser embarcado.



Introdução

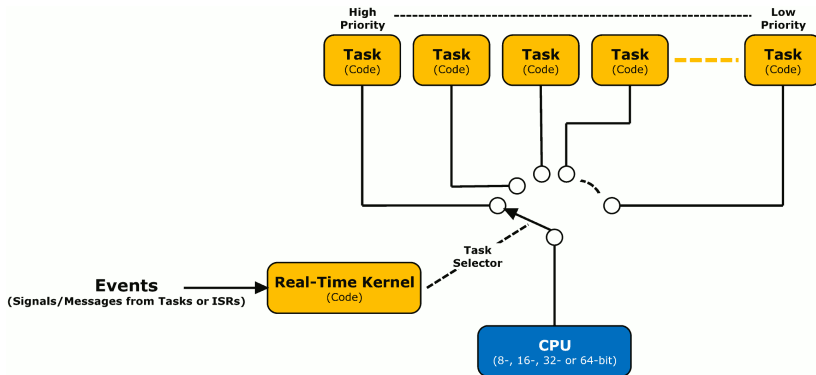
- Um sistema operacional é um software que gerencia os recursos de hardware e software de um microcontrolador.



Introdução

- Um sistema operacional é um software que gerencia os recursos de hardware e software de um microcontrolador.
- Ele também provê uma interface do hardware para as aplicações.

Sistemas operacionais de tempo real



Introdução

- Um RTOS provê a funcionalidade de várias tarefas executarem concomitantemente em ordem de prioridade.

Observações

- Nem toda aplicação embarcada necessita de um RTOS.

Observações

- Nem toda aplicação embarcada necessita de um RTOS.
- Contudo, um RTOS ajuda em 2 aspectos:
 - Provê funcionalidades de programação paralela

Observações

- Nem toda aplicação embarcada necessita de um RTOS.
- Contudo, um RTOS ajuda em 2 aspectos:
 - Provê funcionalidades de programação paralela
 - Ajuda a tornar a aplicação independente do hardware

Observações

- Nem toda aplicação embarcada necessita de um RTOS.
- Contudo, um RTOS ajuda em 2 aspectos:
 - Provê funcionalidades de programação paralela
 - Ajuda a tornar a aplicação independente do hardware
- Uma desvantagem é que uma parcela dos recursos de hardware são usados pelo próprio sistema

Funcionalidades

- Entre as funcionalidades providas pelo sistema operacional encontram-se:
 - Tarefas ou threads com prioridade

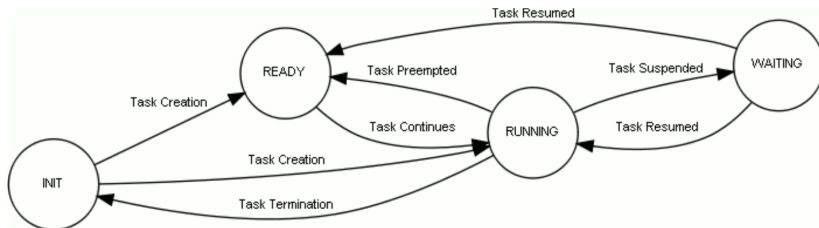
Funcionalidades

- Entre as funcionalidades providas pelo sistema operacional encontram-se:
 - Tarefas ou threads com prioridade
 - Semáforos, eventos e message queues

Funcionalidades

- Entre as funcionalidades providas pelo sistema operacional encontram-se:
 - Tarefas ou threads com prioridade
 - Semáforos, eventos e message queues
 - Drivers para o hardware

Sistemas operacionais de tempo real



Estados de uma thread

- Existem 3 estados principais que uma thread pode estar:
 - Esperando por um evento
 - Sendo executada
 - Pronta para ser executada

Threads

- O chaveamento entre threads é feito através do salvamento e recuperação do estado da CPU imediatamente antes e imediatamente depois da troca entre threads.

Threads

- O chaveamento entre threads é feito através do salvamento e recuperação do estado da CPU imediatamente antes e imediatamente depois da troca entre threads.
- Esse chaveamento leva em conta as prioridades das threads atribuídas pelo programador.

Threads

- Cada thread precisa de um stack próprio além de um pouco de memória para armazenar dados sobre a thread

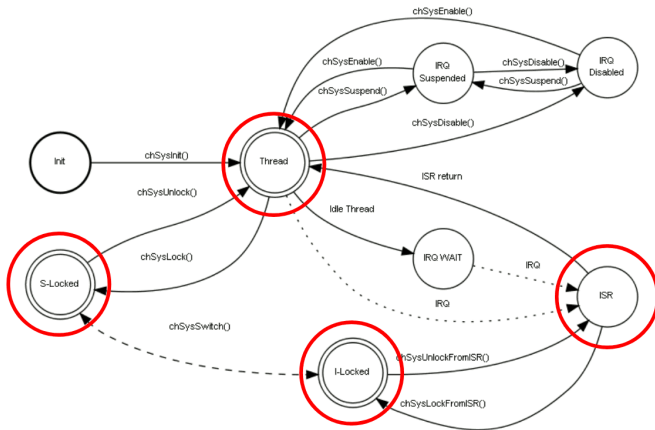
Threads

- Cada thread precisa de um stack próprio além de um pouco de memória para armazenar dados sobre a thread
 - Este espaço de memória é a área de trabalho

Threads

- Precisa-se então definir esta área de trabalho além da própria função da thread:
 - `THD_WORKING_AREA(wa_name, 128);`
 - `THD_FUNCTION(Counter, arg) {}`
- Essas duas definições são usadas na criação da thread:
 - `chThdCreateStatic(wa_name, sizeof(wa_name), NORMALPRIO, Counter, NULL);`

Sistemas operacionais de tempo real



Estados do sistema operacional

- O sistema operacional pode estar em um dos estados, como no exemplo acima.

Threads

- As funções de um RTOS são divididas em classes relacionadas aos estados do sistema.

Threads

- As funções de um RTOS são divididas em classes relacionadas aos estados do sistema.
- No caso do Chibios, temos:
 - Funções da classe I
 - Funções de classe X
 - Funções normais