

RTOS

1.1) Defina tarefas de um sistema operacional

1.2) Dê exemplos reais da utilização de RTOS HARD e SOFT

- Não-Críticos (*soft* RTS - também chamados de *moderados*)

O STR HARD é aquele cuja uma determinada tarefa terá que ser feita dentro de um certo prazo e sua falha não é aceitável, um exemplo real é o freio ABS, ele tem um determinado tempo para responder assim que ele é acionado. Outro exemplo é o sistema embarcado de proteção de linhas de alta.

O STR SOFT tem o mesmo princípio (relacionado ao tempo) do HARD, a diferença é que se houverem falhas, as mesmas serão aceitáveis, por exemplo o sistema de um leitor de DVD.

1.3) Liste exemplos de RTOS opensource e proprietários

Opensource: FreeRTOS, OSEK, scmRTOS, SOOS Project, TRON project

Proprietários: Abassi, AltOS, AMX RTOS, DNIX, Integrity, MicroC/OS-II, MP/M

1.4) Ilustre em camadas (estilo camada OSI) os dois modos: BareMetal e RTOS, identifique a onde está a aplicação, e o microcontrolador (core)

A diferença entre BareMetal e RTOS está na prioridade como as tarefas são tratadas. Um sistema BareMetal faz chamadas de funções por meio de eventos (já possuindo endereço absolutos da execução), quando o mesmo precisa ser tratado em nível de prioridade é ideal a utilização do RTOS

Os cores podem ser: (M0, M3, A1, A8, M4)

1.5) Por que uma interrupção em um RTOS tem que ser resolvida o quanto antes e em um OS tipo Windows não?

Pois como o RTOS trabalha com prioridades, as interrupções são executadas em primeiro plano, atendendo o laço principal em segundo plano enquanto não houver nenhum evento que dispare uma interrupção.

2.1) Qual a função do estado blocked?

É bloquear a tarefa numa determinada parte do tempo, para que apenas uma tarefa possa ser executada, ou seja, aguardar um recurso ou passagem de tempo para que possa executar.

2.2) Faça uma pesquisa como o escalonador do FreeRtos funciona.

O Free RTOS é um sistema operacional escrito em linguagem C, tem tempo real, disponibilizando suporte para várias arquiteturas de sistemas. Ele possui um suporte a tarefas que pode assumir 4 estados: "Em execução", "pronta", "bloqueada" e "suspensa". O escalonador é responsável por gerenciar o processador, implementando uma política de escalonamento baseada em prioridades, ordenando suas tarefas, os recursos entre tarefas serão tratados por meio de um protocolo de prioridades, impedindo assim a tarefa de maior prioridade de ser bloqueada enquanto uma menor seja executada.