



**ENGENHARIA ELETRÔNICA**

**EEN241 – MICROCONTROLADORES E SISTEMAS EMBARCADOS**

**4º NOTURNO**

**PROF. RAFAEL CORSI**

**HENRIQUE PEREIRA ROSA**

**11.02741-0**

**EDUARDO GALINSKAS KARWOSKI**

**13.01129-4**

**08/NOVEMBRO/2016**

## 1.1: Tarefas

Tarefa é uma estrutura básica de execução em um RTOS onde são realizadas as ações do firmware. Também denominada processo, uma task realiza ações bem definidas como acionar um led, realizar uma comunicação. O trabalho do RTOS é justamente gerenciar como as tasks são executadas, escalonando-as (Schedule) e modificando seus estados.

## 1.2: Hard vs. Soft

**HardRTOS:** Sistemas Médicos, Sistemas automotivos de segurança e controle como injeção eletrônica, Sistemas de controle de incêndio.

**SoftRTOS:** Todos que não necessitam ser HardRTOS ou FirmRTOS como sistemas de media ou sensoriamento que não ofereçam risco.

## 1.3: RTOSs

**OpenSource:** FreeRTOS, TargetOS, BRTOS (Brazuka), uC/OS-embedded, us/OS.

**Proprietary:** OpenRTOS, SafeRTOS, ThreadX.

## 1.4: Camada

### Structure of RTOS

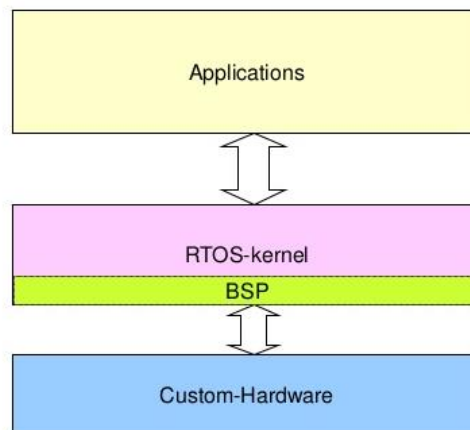


Figura 1: Camadas RTOS

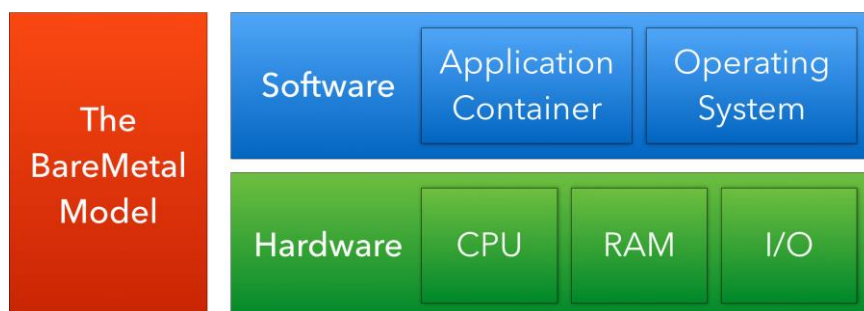


Figura 2: Camadas Baremetal

### 1.5: Prioridade

**Tarefa de mais alta prioridade:** Entra na fila.

**Tarefa entra em Sleep:** Próxima tarefa na fila é escalonada.

### 1.6: Latência

Por que a concepção do RTOS é justamente de realizar tarefas em tempos pré-determinados. Já um OS convencional todas as aplicações que não são do sistema possuem a mesma prioridade.

### 2.1: Blocked

Uma tarefa é colocada no estado Blocked quando já realizou seu ciclo e entrou na função delay para ser executada somente no próximo ciclo. Porém uma task também pode ser posta no estado blocked por um semáforo ou mutex por necessitar de algum recurso de outra task como dados de um beffer serial.

### 2.2: Funcionamento do Escalonador

O FreeRTOS utiliza uma máquina de estados para o gerenciamento das tarefas. Quando é necessário o bloqueio de uma tarefa para permitir uma de igual prioridade ou superior o sistema salva o contexto para voltar exatamente no mesmo ponto de execução. Esse evento pode acontecer por solicitação de uma tarefa de prioridade superior ou através de Context Switch executado por um timming slice.