Relatório 22

1.1: Tarefas

• Defina tarefas de um sistema operacional.

Um processo ou tarefa de um sistema operacional pode ser definido como uma porção de um programa em alguma fase de execução. Um programa pode consistir de várias tarefas, cada uma com funcionamento próprio ou como uma unidade (talvez se comunicando entre si periodicamente).

Exemplos de tarefas são:

- Gerenciamento de recursos;
- Controle de tarefas;
- Gerenciamento de usuários;
- Interpretação de comandos;
- Gerenciamento de arquivos;
- Interfaceamento com o usuário.

1.2: Hard vs Soft RTOS

• De exemplos reais da utilização de RTOS HARD e SOFT.

- Um sensor de temperatura que gera um input para um microcontrolador, indicando que níveis críticos de temperatura foram atingidos, para que o microcontrolador possa atuar sob o sistema de refrigeração. Se o microcontrolador não atender às restrições de tempo do sensor de temperatura, a temperatura poderá atingir níveis críticos, e todo sistema poderá se danificar. Este caso é de um RTOS Hard;
- Um sistema de radar aeroespacial, que recebe informações de posicionamento das aeronaves para que possíveis colisões sejam detectadas e evitadas. Se o sistema não tratar estas entradas dentro de suas restrições de tempo, poderá causar danos sérios. Este caso, assim como o caso anterior, também é um RTOS Hard;
- Um teclado que gera inputs de teclas pressionadas para um sistema microprocessado. Se o sistema não tratar estas teclas dentro de suas restrições de tempo, o operador poderá ter a sensação de que o sistema “travou”, o que poderá causar até a desistência da utilização do produto, porém nesse caso as exigências temporais do sistema são menores, visto que a percepção humana é muito mais lenta do que a velocidade de processamento do microprocessador. Este caso é um RTOS Soft.

1.3: RTOS

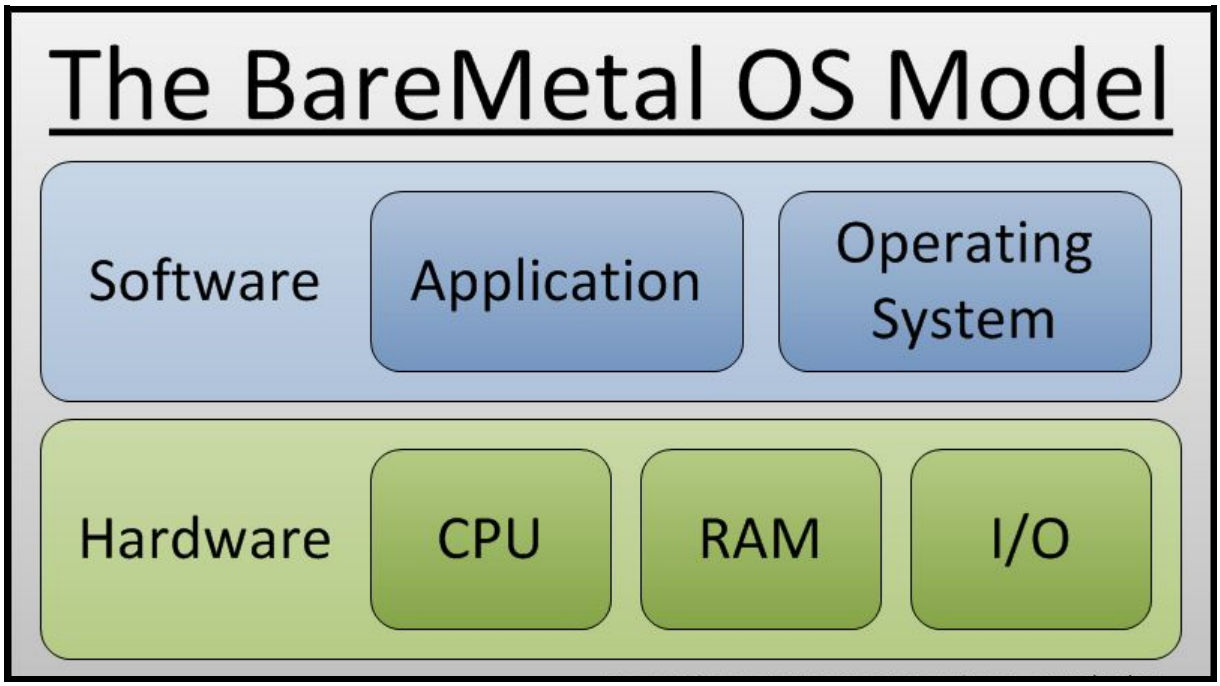
• Liste exemplos de RTOS open-source e proprietários.

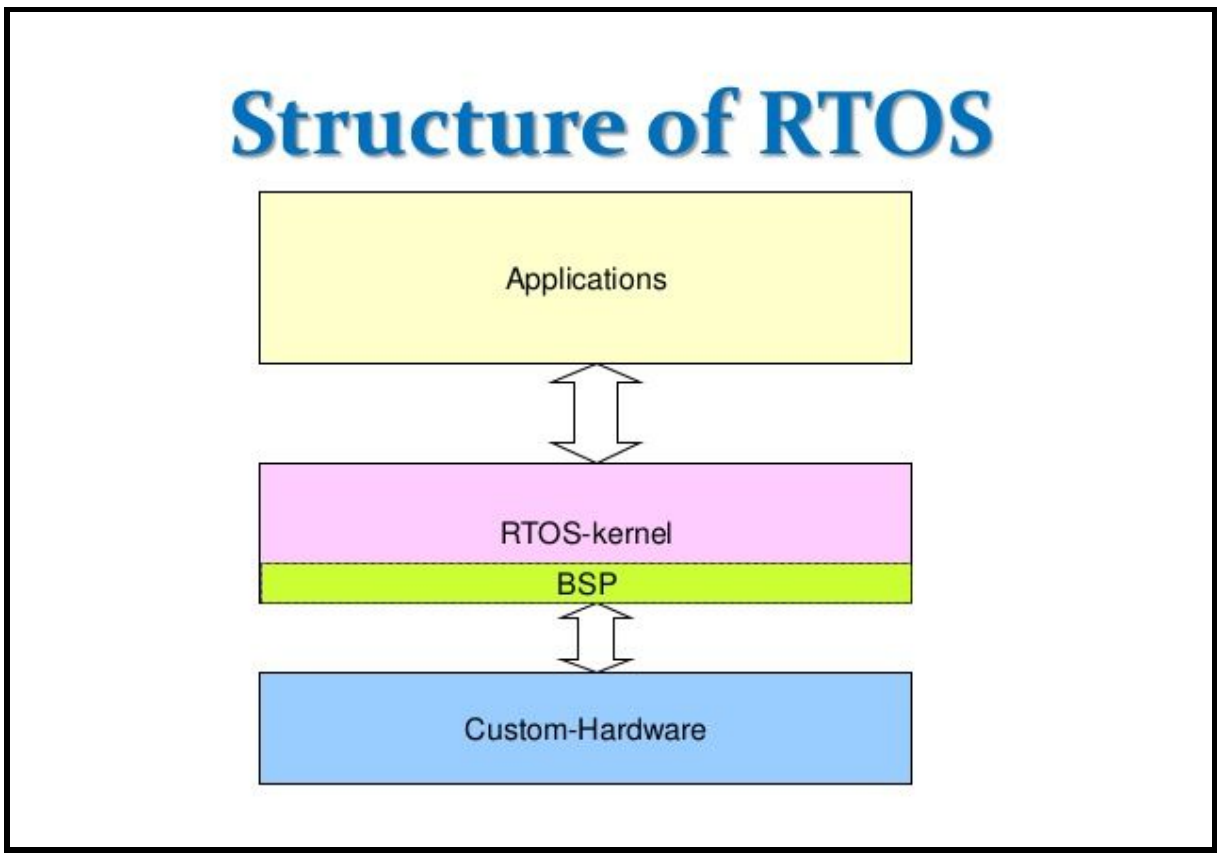
Exemplos de RTOS:

- **TI-RTOS** - TI-RTOS é um SO criado pela Texas Instruments (TI) para uso em uma grande variedade de processadores embarcados.
- **X-Real Time Kernel** - um STR desenvolvido pela eSysTech voltado a processadores ARM.
- **FreeRTOS** - um RTOS de código aberto e que já foi portado para diversas plataformas (ARM, MSP430, PIC, etc).
- **QNX** - RTOS comercial bastante utilizado em aplicações embarcadas.
- **RTA-OSEK** - Um RTOS voltado a aplicações automotivas.
- **AIX** - Advanced Interactive eXecutive - Uma versão do Unix executados em computadores de médio porte da IBM.
- **AMX** - um STR fornecido pela KADAK.
- **CMX** - fornecido pela CMX Company.
- **BRTOS Basic RTOS** - Pequeno e robusto RTOS (ARM,AVR,PIC,MSP430).
- **VxWorks** - RTOS comercial líder de mercado da Wind River.
- **Android**
- **Ubuntu**
- **Debian Linux**
- **Green Hills Software**
- **Micrium uC / OS-II, III (10%)**
- **Windows CE (8%)**
- **RTEMS open source RTOS -** projetado para sistemas embarcados, usado principalmente para mísseis e sondas espaciais controle.

1.4: Camada

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





1.5: Prioridade

• O que aconteceria em um OS tipo Windows se uma tarefa utilizasse o processador enquanto nenhuma das condições anteriores forem satisfeitas?

O OS tipo Windows efetuaria essa tarefa em conjunto com as outras tarefas de prioridade inferior, porém dedicando a maior parte de seu processamento para essa tarefa.

1.6: Latência

• Porque uma interrupção em um RTOS tem que ser resolvida o quanto antes e em um OS tipo Windows não?

Porque os sistemas operacionais em tempo real foram criados para resolver múltiplas tarefas com um tempo reduzido para solução.

2.1: Blocked

• Qual a função do estado blocked ?

Aguardar um recurso ou passagem do tempo para poder executá-la.

2.2: Escalonador

• Faça uma pesquisa como o escalonador do FreeRtos funciona.

- Uma tarefa manda uma mensagem para outra tarefa. Neste momento, o Kernel assume o controle da CPU, prepara o envio da mensagem, e verifica qual a tarefa mais prioritária para execução no momento;
- Uma tarefa requisita um tempo de espera (delay) para o Kernel. Neste momento, o Kernel assume o controle da CPU, e verifica qual a próxima tarefa mais prioritária para execução no momento;
- Uma interrupção do timer do sistema é gerada e tratada pelo Kernel. Neste caso, a tarefa em execução no momento é colocada em espera para que o Kernel possa escalonar para a tarefa mais prioritária;
- Um evento que uma tarefa de prioridade maior estava esperando ocorre e causa a interrupção da tarefa em execução. Este evento pode ser, por exemplo, uma interrupção gerada por um dispositivo qualquer.