

Seleção de Sistemas Operacionais de Tempo Real para Sistemas Embarcados

RESUMO

O presente documento pretende apresentar as principais características dos Sistemas Operacionais de Tempo Real para aplicação em Sistemas Embarcados, a proposta é sugerir uma metodologia para escolha de um Sistema Operacional mais apropriado a ser aplicado em um hardware específico, levando-se em conta à arquitetura do microprocessador ou microcontrolador escolhido.

Palavras-Chave: Sistemas Embarcados, Sistemas Operacionais, Tempo Real

1. INTRODUÇÃO

Os Sistemas Embarcados encontram-se cada vez mais presentes em nosso dia-a-dia através de dispositivos como telefones celulares, decodificadores de TV, aparelhos de DVD e PALMs, porém, a maioria destes dispositivos é controlada por um conjunto dedicado e especialista constituído por hardware, software e periféricos dedicados, mas a crescente demanda por maior conectividade, acesso a Internet e funções multimídia, exige destes dispositivos maior sofisticação, com essa utilização e importância crescente tornam-se necessários estudos na área de Sistemas Operacionais de Tempo Real modularizados e robustos, atendendo assim a nova necessidade de mercado.

No mesmo sentido, a escolha de um RTOS(Sistema Operacional em Tempo Real) leve, tolerante a falhas e que suporte o desenvolvimento de aplicações de forma simples é muito necessária.

2. SISTEMAS EMBARCADOS

Um sistema embarcado, ou sistema embutido, é um sistema no qual a CPU (Unidade Central de Processamento) é completamente dedicada ao sistema que ela controla. Realiza um conjunto de tarefas pré-definidas, geralmente com requisitos específicos, e por ser dedicada pode-se otimizar o projeto reduzindo tamanho, recursos computacionais e custo do produto. Sistemas embarcados são desenvolvidos para uma tarefa específica.

Por questões como segurança e empregabilidade, possuem restrições para computação em tempo real.

O software desenvolvido para sistemas embarcados é muitas vezes chamado firmware¹, e armazenado em uma memória do tipo ROM ou memória do tipo flash ao invés de um disco rígido.

Por vezes o sistema também é executado com recursos computacionais limitados sem tela, sem teclado, e com pouca memória.

2.1 HISTÓRICO

O primeiro sistema embarcado reconhecido foi para o Projeto Apollo, um computador de guia, que operava em tempo real, e era considerado o item mais arriscado do sistema. Já o

¹ Software armazenado em memória do tipo ROM

primeiro sistema embarcado de produção em massa foi o computador guia do míssil nuclear LGM -30 Minuteman, lançado em 1961.

Em 1978 foi lançada pela National Engineering Manufacturers Association a norma para microcontroladores programáveis.

2.2 PROCESSAMENTO

Processadores embarcados podem ser classificados em duas categorias (dependendo da necessidade do hardware), microprocessadores e microcontroladores.

Existem diferentes arquiteturas de processadores e microcontroladores disponíveis tais como ARM, MIPS, Coldfire/68k, PowerPC, x86, PIC, 8051, Atmel AVR, Renesas H8, SH, V850, FR-V, M32R, Z80 e Z8.

Para cada uma dessas arquiteturas deve-se usar o RTOS(Sistema Operacional de Tempo Real) apropriado, uma vez que cada uma das arquiteturas acima possuem restrições de projeto, tais como capacidade de memória, tamanho e principalmente consumo.

2.3 PANORAMA DOS TIPOS DE ARQUITETURAS EMBARCADAS MAIS EMPREGADAS

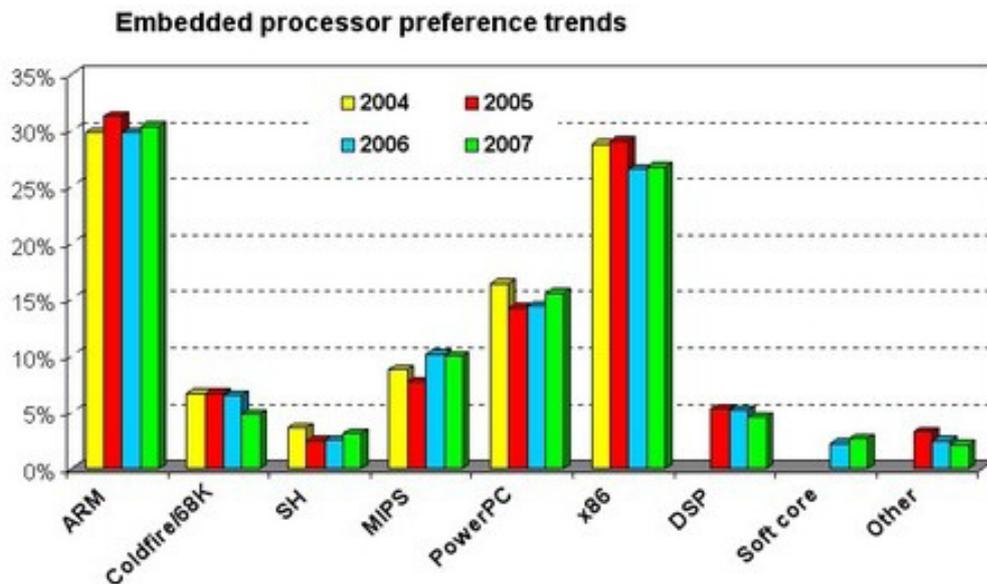


Fig- 1 Panorama das Arquiteturas mais empregadas

2.4 APLICAÇÕES DE SISTEMAS EMBARCADOS

- Aviônicos, como sistemas de controle inercial, controle de vôo e outros sistemas integrados nas aeronaves, como sistemas de orientação de mísseis.
- Telefones celulares e centrais telefônicas.
- Equipamentos de redes de computadores, como roteadores, hubs, switches e firewalls.
- Impressoras.

- Controladores da tração, do motor e do antibloqueio em automóveis: freios ABS (Anti-lock Braking System) e controle de tração.

2.5 RESTRIÇÕES IMPOSTAS

Sistemas Embarcados operam forçosamente com recursos de memória e processamentos limitados e necessitam prover serviços em curto espaço de tempo, são esses confinamentos de memória, velocidade e tempo que ditam a escolha do Sistemas Operacionais de Tempo Real mais indicado para tal aplicação.

3 SISTEMAS OPERACIONAIS DE TEMPO REAL

3.1 DEFINIÇÃO

Um Sistema Operacional de Tempo Real (RTOS) , é um sistema operacional destinado à execução de múltiplas tarefas(multitasking) onde o tempo de resposta a um evento (externo ou interno) é pré-definido. São sistemas computacionais nos quais podem garantir que todas as funções programadas serão executadas em um intervalo máximo de tempo definido para cada função em toda e qualquer momento.

3.2 ATUAÇÃO DE UM SISTEMA OPERACIONAL DE TEMPO REAL

Um Sistema Operacional de Tempo Real deve atuar basicamente:

- Sistema de controle, responsável por responder aos estímulos do ambiente em tempo hábil.
- Interfaces de entrada e saída, portas de comunicação entre o sistema de controle e o sistema controlado, geralmente neste, são sensores, atuadores, receptores de sinais de rádio, entre outros.
- Sistema controlado, é o ambiente com que o computador interage.

3.3 CARACTERÍSTICAS

3.3.1 CARACTERÍSTICAS PRINCIPAIS DE UM RTOS

- **Previsibilidade** - importante propriedade de um sistema em tempo-real, dita seu comportamento funcional e temporal que deve ser tão **determinístico** quanto impõe as especificações do sistema.
- **Confiabilidade** – está relacionada à exatidão no funcionamento do sistema, ou seja, a falha do sistema é que pode gerar uma resposta fora do tempo esperado.

3.3.2 CARACTERÍSTICAS INERENTES AOS SISTEMAS DE TEMPO REAL:

- Utilizam-se diversos processos concorrentes.
- É preciso escalonar os processos e determinar prioridades.
- Existe necessidade de comunicação (sincronizada) entre os diversos processos.
- Existe necessidade de sincronismo entre processos.
- Existe necessidade de sincronismo com eventos externos.

- Necessário lidar com diferentes escalas de arquiteturas, desde pequenos microprocessadores de 4-bits ou 8-bits, até computadores de grande porte interligados em rede.
- Possuem forte interação com o ambiente externo.

3.4 ASPECTOS FUNCIONAIS DE UM RTOS

O Kernel² de um Sistema Operacional de Tempo Real (RTOS) provém os cinco principais serviços básicos para o software de aplicação numa arquitetura embarcada. Estes conjunto de serviços permitem ao desenvolvedor do software de aplicação projetar o software como um número separado de “módulos”, sendo cada módulo responsável por uma tarefa distinta[2].

3.4.1 GERENCIADOR DE TAREFAS

- Agendador de Processos (Task Scheduler) das operações dos sistemas embarcados.
- Trabalha com o conceito de preempção³ baseada em prioridades.
- Processos marcados como alta prioridade tem primazia sobre os demais.

3.4.1.1 ESTRATÉGIAS DE ESCALONAMENTO DE PROCESSOS PARA SEREM ALOCADOS EM UMA ÚNICA CPU:

- Cíclica - a CPU é alocada para um processo de cada vez.
- Preemptiva - o mecanismo de escalonamento é denominado preemptivo se um processo em execução pode ser interrompido

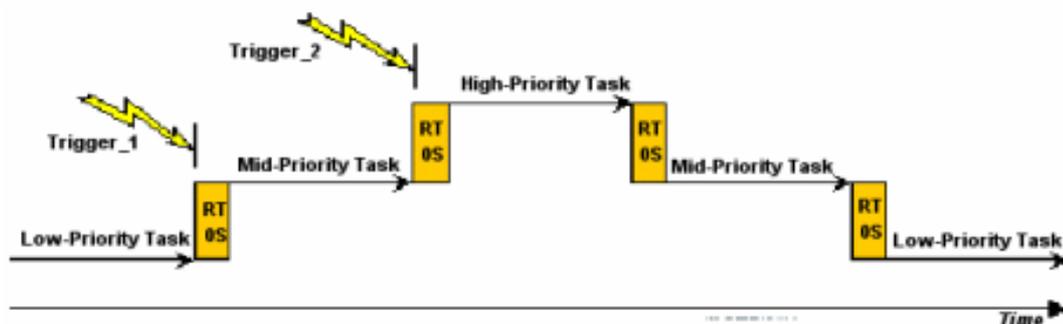


Fig- 2 Processo de escalonamento de tarefas preemptivo

As mudanças de processos acompanham o agendamento preemptivo, dependendo dos gatilhos externos (chaves) ou internos por software(chegada de mensagens).

A mudança dos processos seguem 5 passos principais:

- Determinar se o processo que está rodando atualmente pode continuar a ser executado, se não;
- Determinar qual processo será o próximo
- Salvar o ambiente de execução do processo que será parado (para que o mesmo possa voltar)

² Camada de software mais próxima do hardware, sendo responsável por gerenciar os recursos do sistema computacional como um todo

³ Capacidade de alterar a ordem ou tirar de execução um processo

- Ajustar ambiente para o novo processo que será executado
- Chamar o próximo processo

3.4.1.1.1 EXCLUSÃO MÚTUA

Em programação concorrente é usual que mais de um processo precise compartilhar dados. Para garantir a integridade desses dados, quando um determinado processo está acessando uma área de dados compartilhados, outros processos tem de esperar para poder usá-la.

3.4.1.1.2 SEMÁFOROS

- Criados inicialmente para resolver o problema da exclusão mútua, especialmente no que diz respeito ao problema da espera ocupada.
- Mecanismo eficiente de sincronização, independente de características particulares da arquitetura dos computadores.

3.4.2 TEMPORIZADORES

Aplicações que envolvem a passagem de tempo

- Controla os atrasos dos processos (time-out)
- Ler data e hora (atualizam de histórico)
- Realizar determinada ação depois de X unidades de tempo
- Realizar determinada ação depois de Y unidades de tempo a partir do instante atual (protocolos)
- Realizar determinada ação a partir do instante absoluto de tempo Z

3.4.3 RECURSOS DE MEMÓRIA

- Capacidade dependente do sistema embarcado
- Exemplo de uma configuração de memória para um RTOS Linux (32Mb de SDRAM, 16Mb de flash (8Mb paralela e 8Mb serial))
- Alocação de memória dinâmica.
- Permite que o processo tome emprestado determinadas áreas de memória RAM.
- Temporariamente para uso no software de aplicação.

3.4.4 SUPERVISOR DE DISPOSITIVOS DE E/S

- Controla e organiza acesso de muitos drivers de dispositivos de hardware que são típicos dos sistemas embarcados

3.4.5 OUTROS SERVIÇOS DE UM RTSO:

- Organização de sistemas de arquivos
- Comunicação de redes
- Banco de dados
- Interfaces gráficas

4 ALGUNS TIPOS DE SISTEMAS OPERACIONAIS DISPONÍVEIS

4.1 Free RTOS

Sistema operacional de código aberto e direitos autorais livre, possui um mini Kernel, podendo ser utilizado também em aplicações comerciais. Suporta as seguintes arquiteturas de microprocessadores e microcontroladores (ARM&, ARM9, AVR, PIC24, DSPPIC e microcontrolador 8051).

4.2 NetBSD(baseado no UNIX)

É um sistema operacional em tempo real, que trabalha com código aberto, como desvantagem apresenta à necessidade de memória, podendo chegar à 16Mb de RAM. Boa opção para trabalhar com dispositivos de redes (roteadores , switches).

4.3 Windows CE Embedded

Sistema operacional lançado pela Microsoft, não disponível em código aberto, com tamanho relativamente compacto e possibilidade de incorporar novos módulos e componentes.

4.4 Windows NT Embedded

SO também desenvolvida pela Microsoft, de código fechado, necessita de grande área de memória , não sendo indicado para embarcados.

4.5 Tinys OS

- Sistema operacional de código aberto e projetado para redes de sensores wireless⁴.
- Biblioteca de componentes inclui protocolos de redes.
- Portabilidade de uma dúzia de arquiteturas de controladores

4.6 uClinux

SO de código aberto e completamente voltado para sistemas embarcados. Com Kernel menor que 1 Gb, e suporte ao protocolo de redes TCP/IP.

5 REQUISITOS PARA ESCOLHA DE UM SISTEMA OPERACIONAL DE TEMPO REAL

5.1 REQUISITOS RELEVANTES PARA ESCOLHA DE UM SISTEMA OPERACIONAL DE TEMPO REAL PARA UMA DETERMINADA ARQUITETURA

Na tabela abaixo, são apresentados critérios relevantes para auxiliar na escolha de um RTOS (Sistema Operacional de Tempo Real) a ser empregado em um sistema embarcado, utilizando uma arquitetura de processador específica. Sempre levando em conta as limitações básicas impostas a um sistema embarcado, como capacidade mínima de memória, processamento limitado, além do consumo de energia por parte do sistema móvel (embarcado).

O propósito deste é estabelecer parâmetros de comparação entre os vários RTOS disponíveis no mercado.

⁴ Redes sem fio

Tabela 1 – Requisitos relevantes para escolha de um RTOS

Categoria	Critérios de Avaliação
Kernel	Arquitetura Multi-processadores Multiprocessamento Falha de tolerância
Agendamento	Algoritmo Mecanismo de prioridade Tempo de realização das tarefas (processos)
Modelos das Tarefas	Número de níveis e prioridades Número máximo de tarefas Dinâmica de troca de prioridades
Memória	Espaço máximo e mínimo por tarefas Espaço máximo e mínimo de ROM Alocação Dinâmica Memória Virtual
Aplicações Programação Interface	Timer interno e externo Sincronização Comunicação entre processos Protocolos de redes
Desenvolvimento	Tipo de código fonte do RTOS se aberto ou não Compilador Processadores suportados Linguagens para desenvolvimento

5.2 COMPARAÇÃO DE ALGUNS SISTEMAS OPERACIONAIS DE TEMPO REAL

Tabela 2 – Comparação de alguns RTOS mais utilizados

	FreeRTOS	eCOS	RTX-3.2
Plataformas	Windows Linux	Windows Linux	Windows Solaris
Arquiteturas	ARM/AVR/ PIC18 8051 X86	X86 PowerPC ARM Freescale 68k	PowerPC ARM Hitachi

Compiladores	WINAVR GCC KEIL IAR	GCC	Keil TI IAR
Distribuição	Gratuito www.freeRTOS.org	Gratuito www.ecoscentric.com	Não gratuito
Suporte para redes		TCP/IP Telnet HTTP	TCP/IP Telnet FTP
Kernel ROM Kernel RAM	2Kbytes – 5Kbytes 200bytes	<1250bytes <600bytes	
Aplicações	Aplicações comerciais	Play Station 3 Radio do Satélite Sirius Missil Patriot TVS de LCD da Samsung	

6 CONCLUSÃO:

Até esse ponto sobre o estudo apresentado sobre um RTOS, foram levantados que os mesmos são dependentes de características básicas de um sistema embarcado, limitando-os no hardware (memória e economia de energia), software (aplicações em tempo real) e custo.

Pelo acima exposto é necessário considerar na escolha de um bom RTOS (Sistema Operacional de Tempo Real) alguns quesitos como tamanho do Kernel (compatível com tamanho de memória disponível no hardware) a ser empregado, o algoritmo para tratamento e número de interrupções atendidas pela arquitetura utilizada, aplicações internas e externas com o sistema, temporizações, documentação disponível, e outros implementos (compiladores, linguagens, outros) fornecidos pelo fabricante, e conforme demonstrado através da tabelas- 1.

Conclui-se com isso, ser possível chegar a uma escolha do mais próximo possível de um RTOS apropriado a um sistema embarcado com arquitetura específica.

7 REFERÊNCIAS BIBLIOGRÁFICAS:

- TIMMERMAN, Martin .RTOS STATE OF THE ART Disponível em:
<http://www.freertos.org/implementation/index.html> - Acesso em: 20/04/08
- PRADO, Sergio. DESENVOLVIMENTO DE SOFTWARE PARA SISTEMAS EMBARCADOS – Disponível em:
<http://www.embedded.com/design/multicore/> - Acesso em: 20/04/08

MORIMOTO, Carlos E. INTRODUÇÃO SOBRE SISTEMAS EMBARCADOS

Disponível em:

<http://www.embarcados.com.br/content/view/> - Acesso em: 22/04/08

Fabricantes de Sistemas Operacionais de Tempo Real. Disponível em:

www.embarcados.com.br

www.tinyos.net

www.armkits.com

<http://linuxabordo.com.br>

<http://linuxdevices.com>

<http://freeelectrons.com>

<http://ucdot.org>

Qemu: <http://fabrice.bellard.free.fr/qemu/>

<http://www.busybox.net/>

<http://www.codesourcery.com/>