

## REQUISITOS PARA ADOÇÃO DE SISTEMAS OPERACIONAIS EMBARCADOS REQUIREMENTS FOR EMBEDDED OPERATING SYSTEMS ADOPTION

**Maiko R. Moroz\***

**Ricardo P. Jasinski\*\***

**Volnei A. Pedroni\*\*\***

Fecha de envío: Mayo 2011

Fecha de recepción: Mayo 2011

Fecha de aceptación: Septiembre 2011

### Abstract

Embedded systems are part of numerous applications, most of which include real-time or specialized operating systems. The operating system must be chosen taking into account functional and technical characteristics of the embedded product. This paper presents a survey on the main requirements to help determine whether an operating system should be used in an embedded design, further we show how these requirements may be used for assessing the operating system selection.

### Keywords

Embedded Systems, Real-time Systems, RTOS, Operating System Selection.

### Resumo

Os sistemas embarcados compõem a grande maioria dos dispositivos eletrônicos. A decisão de usar um sistema operacional é umas das mais importantes em um projeto embarcado. Para auxiliar nestas decisões, este artigo apresenta os principais requisitos que podem ser considerados em um projeto embarcado. Em seguida apresentamos os critérios para decisão sobre o uso ou não de um sistema operacional e os requisitos técnicos e comerciais utilizados na escolha de um sistema operacional embarcado.

### Palavras chave

Sistemas embarcados, Sistemas de tempo real, seleção de sistemas operacionais.

### Introdução

Sistemas embarcados estão presentes em, praticamente, todos os dispositivos eletrônicos, sendo encontrados em eletrodomésticos, carros e aviões. Estes dispositivos desempenham funções de controle ou monitoramento no ambiente ou equipamento em que estão inseridos.

Os sistemas embarcados podem ser implementados com ou sem a utilização de um sistema operacional, porém a pesquisa re-

\* Mestrando em Engenharia Elétrica e Informática Industrial na Universidade Tecnológica Federal do Paraná. Correo: maikomoroz@gmail.com

\*\* Pesquisador do Laboratório de Microeletrônica da UTFPR e doutorando em Engenharia Elétrica e Informática Industrial na Universidade Tecnológica Federal do Paraná. Correo: jasinski@solvis.com.br

\*\*\*Professor Associado da Universidade Tecnológica Federal do Paraná, doutor em Electrical Engineering pelo California Institute of Technology - Caltech. Correo: pedroni@utfpr.edu.br

realizada pelo site EETimes em 2010 [1] revelou que em 70% dos projetos embarcados pesquisados foi utilizado um sistema operacional. Sendo este um dos principais componentes em um projeto embarcado. Mas, apesar da sua importância para o projeto, não há uma abordagem consistente ou um método para auxiliar na decisão de adoção de um sistema operacional e em como selecionar um. Frequentemente a escolha por um sistema operacional é baseada apenas em critérios não técnicos como experiência prévia com determinado produto.

Este artigo aborda o assunto de sistemas embarcados com o objetivo de suprir a carência de material sobre adoção e seleção de sistemas operacionais. Apresentamos os requisitos dos sistemas embarcados que podem ser utilizados na decisão de utilização de um sistema operacional. Em seguida apresentamos como estes requisitos influenciam a escolha do sistema operacional, quando decidido pela utilização de um.

### Sistemas Embarcados

Um sistema embarcado é um sistema computacional projetado para um propósito específico e geralmente parte de uma aplicação maior [2]. Frequentemente, estes sistemas operam em tempo real e sem intervenção do usuário.

Sistemas embarcados são compostos por hardware (processador, memórias e periféricos) e software. Um esquema representando um sistema embarcado é mostrado na Figura 1. Dependendo da aplicação podem incluir interfaces de comunicação, sensores e atuadores.

Geralmente, sistemas embarcados não existem isolados, mas precisam se comunicar com outros dispositivos (I2C, SPI, USB,

Ethernet, Bluetooth e ZigBee) e/ou usuários (botões, teclados, vídeo e som).

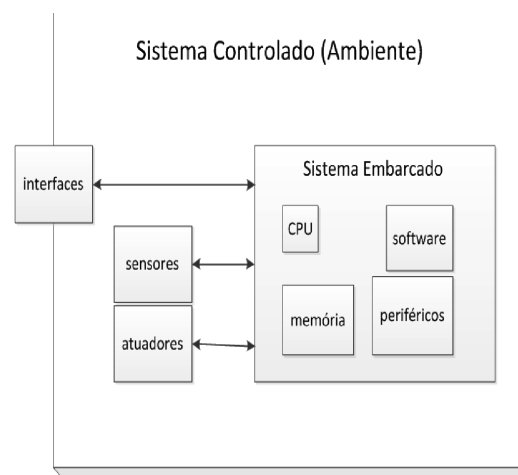
São exemplos de aplicações de sistemas embarcados os roteadores sem fio, câmeras fotográficas digitais, console de videogame, GPS automotivo e injeção eletrônica.

### Sistemas Operacionais de Tempo Real

O sistema operacional é uma camada de software que oculta os detalhes do hardware e disponibiliza aos desenvolvedores de software uma melhor interface de programação [3]. As principais funções dos sistemas operacionais são fornecer serviços aos programas, abstrair os detalhes do hardware e coordenar acesso ao mesmo.

Os sistemas operacionais de tempo real possuem relação intrínseca com o tempo e são construídos para permitirem o cumprimento de requisitos temporais específicos. Eles podem ser classificados em *hard real-time* e *soft real-time*. No primeiro, uma falha grave poderá acontecer se um *deadline* for perdido,

**Figura 1. Esquema representativo de um sistema embarcado.**



enquanto no segundo a única penalidade é a degradação na performance. *Hard real-time* e *soft real-time* são dois pontos extremos no espectro de classificação dos sistemas operacionais quanto ao seu desempenho de tempo real. Na prática, os sistemas operacionais de tempo real estão entre estes dois pontos.

Para suportar múltiplas tarefas em aplicações de tempo real, o escalonador deve ser preemptivo baseado em prioridades [2]. Neste caso, o escalonador deve ser capaz de interromper qualquer tarefa para permitir a execução de outra com maior prioridade. Além disto, também deve possuir vários níveis de interrupção para permitir tratamento de interrupções com diferentes níveis de prioridade.

Para permitir que várias tarefas se comuniquem entre si em um tempo hábil, é necessário que a sincronização e a comunicação entre os processos sejam previsíveis. O tempo para iniciar uma tarefa, latência entre troca de tarefas e latência de interrupção devem ser bem definidos e consistentes mesmo com o aumento da carga do sistema [4].

### Requisitos dos Sistemas Embarcados

Cada projeto possui uma série de requisitos que guiam a escolha de cada componente individual no sistema embarcado. Os requisitos fornecem descrições dos serviços fornecidos pelo sistema e suas restrições; podem ser classificados em funcionais e não funcionais [5].

- Os requisitos funcionais definem a utilidade do sistema, são descrições das atividades que o sistema deve ser capaz de executar. De maneira geral, requisitos funcionais especificam o que o sistema deve fazer.
- Os requisitos não funcionais especificam as restrições e limitações do sistema.

Eles se relacionam com as características e detalhes de implementação do sistema. Geralmente, requisitos não funcionais definem as qualidades do sistema, ou como este deve ser.

Os principais requisitos não funcionais [6], os quais podem influenciar na escolha de cada componente do sistema embarcado são listados a seguir.

- a) Limitação de recursos. Sistemas embarcados frequentemente são projetados com recursos limitados de hardware, assim devem utilizar eficientemente os recursos disponíveis.
- b) Tempo real. Muitas aplicações possuem requisitos temporais que devem ser cumpridos para garantia do seu correto funcionamento.
- c) Confiabilidade. As aplicações de sistemas embarcados geralmente exigem alta disponibilidade, que é o funcionamento contínuo por longos períodos de tempo. O sistema deve ser confiável para garantir a disponibilidade.
- d) Robustez. Algumas aplicações são sujeitas a operação em condições adversas. A robustez é a habilidade de o sistema funcionar aceitavelmente quando estiver fora das suas condições nominais. Por exemplo, vibração, altas temperaturas ou flutuações na fonte de alimentação.
- e) Estabilidade. Em muitas aplicações críticas, o sistema não pode ser contaminado por uma falha isolada. A estabilidade está relacionada a habilidade do sistema manter falhas contidas.
- f) Gerenciamento de falhas. Algumas aplicações precisam continuar produzindo resultados coerentes mesmo na presença de

falhas. Por exemplo, um sistema de navegação em uma aeronave. O gerenciamento de falhas compreende detecção, confinamento, recuperação e tratamento das falhas.

- g) Proteção. Algumas aplicações lidam com dados que devem ser mantidos em segredo, por exemplo, um sistema eletrônico de pagamentos. A proteção é a capacidade de o sistema evitar o acesso não autorizado às informações sigilosas.
- h) Segurança. Algumas aplicações podem oferecer riscos em caso de falhas, para serem consideradas seguras devem prevenir danos à vida, às pessoas, aos equipamentos e ao ambiente. Por exemplo, uma usina nuclear.
- i) Privacidade. Algumas aplicações necessitam permanecer anônimas mesmo em um ambiente público. A privacidade é a habilidade de o sistema isolar informações sobre si mesmo e as revelar seletivamente.
- j) Escalabilidade. Muitas aplicações são projetadas para uma determinada carga de trabalho, mas em algumas situações precisam lidar com uma sobrecarga de trabalho. A escalabilidade está relacionada com o sistema continuar funcionando normalmente nestas situações de sobrecarga.
- k) Atualização. O tempo de vida de algumas aplicações é longo e o sistema embarcado deve ser capaz de ser atualizado ao longo do tempo. A atualização ocorre pela adição ou substituição de componentes do sistema.
- l) Consumo de energia. Em muitas aplicações o consumo de energia deve ser minimizado ao máximo, sendo um fator crítico na escolha de cada componente. Por exemplo, redes de sensores sem fio.
- m) Custo. Muitos custos estão envolvidos em um projeto, sendo este um fator importante

para a maioria das aplicações. Produtos produzidos em larga escala costumam ser sensível ao custo, devendo este ser minimizado para garantir a viabilidade do produto.

- n) Manutenção. A manutenção é a substituição de componentes defeituosos de hardware ou partes do software.
- o) Usabilidade. Aplicações que interagem com o usuário precisam ser de fácil utilização. A usabilidade do sistema é a facilidade de uso de um determinado sistema.
- p) Portabilidade. A portabilidade é a migração do sistema para outra plataforma de hardware ou software.

### Necessidade de um Sistema Operacional

Antes de se escolher um sistema operacional deve-se avaliar a sua necessidade no projeto. Não é clara a divisão entre projetos que necessitam ou não sistemas operacionais, sendo que muitos projetos aceitam ambas as soluções. A experiência dos desenvolvedores com determinada solução possui grande peso na escolha, mas algumas considerações podem ajudar neste processo.

De forma geral, sistemas modelados em uma ou poucas tarefas não necessitam de um sistema operacional. Estes podem ser implementados utilizando a arquitetura conhecida como *Super Loop* [7], que consiste em um laço infinito contendo todas as tarefas a serem executadas. Um exemplo do *Super Loop* é mostrado na Figura 2; a rotina *Initialization* é executada apenas uma vez, enquanto as demais rotinas são executadas infinitamente.

A execução de um sistema operacional consome recursos do sistema, como ciclos







