

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).

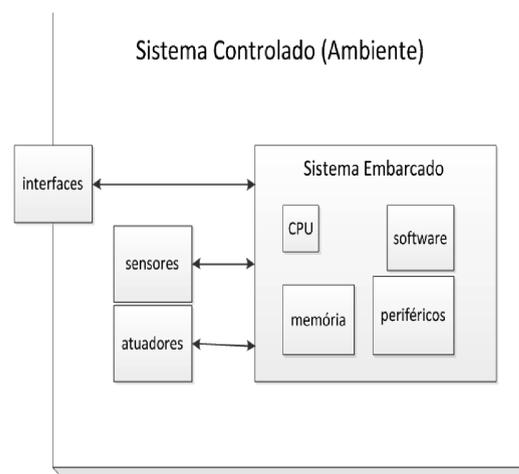
São exemplos de aplicações de sistemas embarcados os roteadores sem fio, câmeras fotográficas digitais, console de videogame, GPS automotivo e injeção eletrônica.

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

Figura 2. Exemplo de tratamento de eventos pelo método *Super Loop* [8].

```

Function Main_Function ()
{
Initialization ();
Do_Forever
{
Check_Status ();
Do_Calculations ();
Output_Response ();
}
}
    
```

de CPU e memórias de programa e dados. Desta forma, sistemas operacionais não são adequados em projetos cujo hardware previamente definido é muito restritivo, embora existam sistemas operacionais projetados especificamente para microcontroladores com pouca disponibilidade de recursos. Por exemplo, o microcontrolador AT89S51 da Atmel cujas características relevantes estão reproduzidas na Tabela 1; devido aos poucos recursos oferecidos por este microcontrolador, não é recomendável a utilização de um sistema operacional.

A utilização de um sistema operacional pode encarecer um produto à medida que exige mais recursos de hardware. Em produtos onde o custo é um fator crucial na escolha dos componentes pode ser preferível optar pela

Tabela 1. Características do microcontrolador AT89S51.

Memória de programa	4 KB
Memória de dados	128 B
Máxima frequência de operação	24 MHz

Tabela 2. Serviços oferecidos por sistemas operacionais.

Suporte a processador de vários núcleos
 Gerenciamento de memória por hardware
 Comunicação e sincronização de processos
 Tarefas com diferentes prioridades
 Tarefas de tempo real

- Protocolos de comunicação de alta complexidade:
- Local: PCI, PCIe;
- Ponto a ponto: USB, FireWire;
- Rede: Ethernet, Wi-Fi, Bluetooth.

Sistemas de arquivos
 Interface gráfica com o usuário
 Interface homem-máquina

não utilização de um sistema operacional, visto que além do custo do hardware há o custo de aquisição do sistema operacional e, em alguns casos, royalty por unidade produzida.

Apesar das exigências impostas pela utilização de um sistema operacional, estes oferecem muitas vantagens aos desenvolvedores e ao processo de desenvolvimento. Os benefícios mais evidentes são a modularização, reutilização de código fonte e facilidade de integração de partes de software.

Adicionalmente, a maioria dos sistemas operacionais oferece alguns dos serviços listados na Tabela 2. Quando algum destes serviços estiver presente nos requisitos do projeto é recomendável sua implementação utilizando um sistema operacional.

Seleção de sistemas operacionais

A escolha de um sistema operacional é uma das decisões mais importantes no projeto de um sistema embarcado. Se após as considerações da seção 5 foi decidido pela utilização de um sistema operacional, ainda é necessário escolher

qual o mais adequado para o projeto. Os requisitos apresentados nas seções 4.1 e 4.2 a seguir são usados para este processo de seleção.

Requisitos técnicos

Os requisitos técnicos dizem respeito às propriedades e limitações dos sistemas operacionais. Os principais requisitos técnicos estão listados na Tabela 3.

Em projetos onde a migração entre diferentes processadores é uma possibilidade, o suporte a diferentes famílias de processadores é considerado um requisito de portabilidade. Nestes casos, é importante que o sistema operacional suporte diferentes processadores.

Os sistemas operacionais necessitam de memórias de código e de dados. A quantidade requerida destas memórias aumenta com o número de funcionalidades providas pelo sistema operacional. Caso as quantidades disponíveis de memória sejam uma limitação do projeto, estas podem impedir a adoção de um sistema operacional com mais funcionalidades. Porém a maioria dos sistemas operacionais é altamente configurável, o que permite a inclusão de apenas os componentes necessários para o sistema embarcado, originando uma imagem otimizada do sistema operacional.

Tabela 3. Requisitos técnicos dos sistemas operacionais.

Processadores suportados
Requisitos de memória
Desempenho
Tarefas e processos
Recursos e Serviços
Protocolos de comunicação
Ferramentas de desenvolvimento

O desempenho do sistema operacional está relacionado à sua capacidade de lidar com tarefas de tempo real, o tempo gasto para troca entre tarefas e a latência no atendimento de interrupções. Tudo isto influencia na capacidade do sistema atender aos requisitos temporais especificados no projeto.

O escalonamento de processos é uma das principais vantagens da utilização de um sistema operacional. Está relacionado a esta função o algoritmo de escalonamento, a quantidade de tarefas simultâneas suportadas, a quantidade de níveis de prioridade, a possibilidade de criação e remoção de tarefas dinamicamente e o mecanismo para prevenção da inversão de prioridades.

Além do escalonamento de tarefas, os sistemas operacionais também oferecem outros recursos e serviços. Como por exemplo, suporte a processadores com vários núcleos, gerenciamento de memória, mecanismos de comunicação e sincronização entre processos, sistemas de arquivos, protocolos de comunicação e interface gráfica com o usuário.

Os protocolos de comunicação são regras para permitir a troca de informações entre sistemas. Muitos protocolos de comunicação são suficientemente complexos para desencorajar o seu uso sem a presença de um sistema operacional. Nestes casos, a utilização de um sistema operacional que possua os requeridos protocolos embutidos é altamente recomendada.

As ferramentas de desenvolvimento são softwares e hardwares utilizados no processo de desenvolvimento. Apesar do grande impacto que estas ferramentas causam na produtividade, em muitos casos sua importância é negligenciada. As ferramentas de desenvolvimento mais comuns são ambiente integrado de

desenvolvimento (IDE), compiladores, *profilers*, *debuggers*, simuladores e emuladores.

Requisitos comerciais

Os requisitos comerciais dizem respeito às regras de negócio, ao orçamento disponível e aos demais fatores não técnicos. Os principais fatores comerciais estão listados na Tabela 4.

A reputação do sistema operacional é importante em alguns projetos, especialmente aqueles projetos críticos ou de longa duração. Um fator geralmente considerado é o fabricante, ou seja, quem é responsável pelo desenvolvimento e manutenção do sistema operacional. Além disso, políticas corporativas podem exigir que os softwares utilizados sejam de empresas confiáveis. Finalmente, são considerados os prêmios e certificações recebidos pelo sistema operacional.

O sistema operacional pode ser distribuído sob uma licença comercial ou livre (*free software* ou *open source*). No caso de uma licença livre, restrições sobre o uso, modificação ou redistribuição podem ser impostas ao produto que a utilize. Por exemplo, a licença GNU General Public License (GPL) exige que modificações feitas ao código fonte sejam disponibilizadas para outros desenvolvedores. Em contrapartida, algumas licenças comerciais

permitem acesso ao código fonte sem necessidade de redistribuição das modificações.

Os principais custos envolvidos na adoção de um sistema operacional comercial são o licenciamento inicial e em alguns casos royalties por unidade produzida. De forma geral, o fabricante fornece, juntamente com o sistema operacional adquirido, suporte técnico por um tempo limitado, o qual pode ser ampliado posteriormente. Por outro lado, no caso de sistemas operacionais livres, o suporte pode ser contratado de consultores, o que pode ser considerado um risco adicional para o projeto.

A fim de possibilitar melhor portabilidade do código fonte entre diferentes sistemas operacionais algumas interfaces padrão de programação (API) foram padronizadas, por exemplo, POSIX e ITRON. A compatibilidade com padrões pode obrigatória para algumas aplicações comerciais e contratos com o governo, devendo ser considerados nestes casos.

Conclusão

Neste artigo abordamos os requisitos dos sistemas embarcados e sistemas operacionais, os quais podem auxiliar no processo de decisão pela utilização ou não de um sistema operacional embarcado. No caso de um projeto baseado em um sistema operacional, mostramos como selecionar o sistema operacional mais adequado, baseando-se nos requisitos deste projeto.

Referências

- [1] EETimes Group. 2010. Disponível em: <http://www.eetimes.com/electrical-engineers/education-training/webinars/4006580/2010-Embedded-Market-Study>.

Tabela 4. Requisitos comerciais dos sistemas operacionais..

Reputação
Licença
Disponibilidade do código fonte
Custo
Suporte
Compatibilidade com padrões

- [2] S. Baskiyar y N. Meghanathan. "A Survey of Contemporary Real-time Operating Systems". *Informatica* (2005): 233-240,.
- [3] Andrew S. Tanenbaum. *Sistemas Operacionais Modernos*. 2 ed. Sao Pablo: Prentice Hall, 2003.
- [4] T. N. B. Anh y Su-Lim Tan. "Real-time operating systems for small microcontrollers". *Micro*, 29(5) (sept.-oct. 2009): 30-45. IEEE.
- [5] Ian Sommerville. *Engenharia de Software*. 8 ed. Sao Pablo: PearsonAddison-Wesley, 2007.
- [6] Luís F. Friedrich. "A Survey on Operating System Support for Embedded Systems Properties". *Anais do Workshop de Sistemas Operacionais*, VI (julio 2009): 2393-2404.
- [7] Michael Pont. *Embedded C*. Londres: Addison-Wesley, 2002.
- [8] Wikibooks. Marzo de 2011. Disponible en: [http://en.wikibooks.org/wiki/ Embedded_Systems/Super_Loop_Architecture](http://en.wikibooks.org/wiki/Embedded_Systems/Super_Loop_Architecture)