

# 1. Caracterização dos Sistemas de Tempo Real

*O que é um sistema de tempo real ?*

Todas as pessoas possuem intuitivamente a noção de “tempo”, do “passar do tempo” e de “prazos”. Já a definição do termo “sistema de tempo real” é bem menos conhecida. No entanto, estamos cercados por eles. Por exemplo, usinas hidroelétricas, aviões e videogames são desenvolvimentos tecnológicos que incluem, entre outras coisas, sistemas de tempo real. Neste capítulo serão apresentados os principais aspectos que caracterizam os sistemas de tempo real. Os vários detalhes relacionados com sua construção e análise serão tratados ao longo deste livro.

## 2. Conceitos Básicos dos Sistemas de Tempo Real

*Quais são os principais conceitos dos sistemas de tempo real ?*

Como visto no [capítulo 1](#), sistemas de tempo real são aqueles submetidos a requisitos de natureza temporal não triviais. Os requisitos aparecem, na maioria das vezes, como um prazo máximo para a execução de uma dada funcionalidade do sistema ou um período no qual dada funcionalidade do sistema deve ser sempre executada. Nos sistemas de tempo real, os resultados e as ações do sistema devem ser corretos não apenas logicamente e aritmeticamente, mas também precisam ser corretos temporalmente, ou seja, precisam respeitar os requisitos temporais. E estes requisitos temporais são sempre definidos em função do ambiente físico que cerca o sistema computacional. Eles não são escolhidos na implementação, mas sim definidos na especificação.

Um sistema de tempo real deve ser construído para atender os requisitos temporais. Inclusive, podem existir vários requisitos temporais simultâneos, tais como várias malhas de controle realimentado simultâneas, respostas às ações do operador humano, requisições via rede de comunicação, etc. Isto tudo acontece junto com outras atividades que não são de tempo real mas precisam ser realizadas com desempenho razoável.

É necessário entender como o sistema comporta-se no tempo para ter segurança de que os requisitos temporais serão cumpridos quando o mesmo for utilizado. Para facilitar este entendimento, e a análise de seu comportamento, são usados modelos que descrevem como a execução do software do sistema acontece. A grande vantagem dos modelos é que eles abstraem a maioria dos detalhes, permitindo que a análise do sistema seja focada nas questões específicas de tempo real.

## 3. Implementação de Tarefas em Sistemas Pequenos

*Como tarefas de tempo real podem ser implementadas em um sistema simples?*

O conceito de tarefa de tempo real foi apresentado no [capítulo 2](#), o qual tratou dos conceitos básicos do escalonamento tempo real. O tempo de resposta (*response time*) de uma tarefa é definido pelo intervalo de tempo entre a chegada da tarefa e a sua conclusão, incluindo aqui eventuais intervalos de tempo quando a mesma é suspensa para a execução de outras atividades do sistema. No contexto de um sistema de software, várias atividades podem ser definidas e, de alguma forma, todas elas compartilham o mesmo processador (ou os mesmos processadores, no caso de máquinas com múltiplos processadores).

O tempo de resposta de uma tarefa de tempo real está diretamente associado com a forma adotada para implementá-la em um sistema de software. Em outras palavras, depende da **organização dos fluxos de execução** no sistema, ou seja, do **design do software do sistema** (*system software design*). A variedade de possibilidades neste aspecto é imensa.

Neste capítulo vamos considerar os três cenários mais usuais na implementação das tarefas de tempo real em sistemas simples, especialmente no que se refere aos sistemas de tempo real com requisitos temporais rigorosos: Executivo cíclico, laço principal com interrupções, e sistema operacional multitarefa na forma de um microkernel simples. Um outro cenário possível, usando sistema operacional multitarefa na forma de um kernel completo, será tratada no [capítulo 4](#).

## 4. Implementação de Tarefas em Kernel Completo

*Como tarefas são implementadas em sistemas operacionais tradicionais ?*

Cada vez mais aplicações complexas são incluídas em equipamentos e veículos, aplicações estas que incluem um grande número de tarefas de tempo real com muitos periféricos para gerenciar. Muitas aplicações de tempo real, mesmo aquelas que executam em computadores embutidos (*embedded*) em equipamentos, precisam de interface gráfica de usuário, sistema de arquivos, pilhas de protocolos de comunicação, controle de acesso, e estes são serviços normalmente providos por um sistema operacional. Aplicações deste tipo requerem um **kernel completo** (*full-fledge kernel*) como sistema operacional multitarefa. O exemplo clássico neste caso é o sistema operacional **Linux**.

Neste capítulo vamos considerar a implementação das tarefas de tempo real no contexto de sistemas operacionais tradicionais, ou seja, através de um sistema operacional multitarefa implementado por um kernel que oferece funcionalidade completa.

## 5. Escalonamento de Tarefas

*Como escalonar as tarefas em um sistema de tempo real ?*

Como visto no [capítulo 3](#) e no [capítulo 4](#) sobre a implementação de tarefas, normalmente existe um número de tarefas aptas a executar que é maior do que o número de processadores. Desta forma é preciso definir que algoritmo será usado para escolher qual tarefa será executada a seguir (ou quais no caso de *multicore*).

Em sistemas de propósito geral existem algumas formas básicas, as quais serão apresentadas na [seção 5.1](#) deste capítulo. Porém, no caso de sistemas de tempo real, a forma mais utilizada de escalonamento é aquela baseada em prioridades preemptivas. Neste caso ainda resta um problema importante: definir quais critérios utilizar para atribuir a prioridade de cada tarefa.

Este capítulo discute os principais algoritmos de escalonamento encontrados na prática, com principal atenção às prioridades preemptivas, dada sua importância para sistemas de tempo real. A questão da inversão de prioridades é apresentada, quando o sistema não consegue executar as tarefas na ordem exata de suas prioridades. As características das principais políticas de atribuição de prioridades são também apresentadas, incluindo aqui o conceito de utilização, o qual é fundamental na teoria de escalonamento tempo real.

## 6. Sincronização e Comunicação entre Tarefas

*Como coordenar a ação das tarefas da aplicação para que colaborem entre si ?*

Aplicações de tempo real precisam reagir a estímulos externos ao mesmo tempo que respeitam requisitos temporais como deadlines e períodos. Isto acontece porque a aplicação interage de alguma forma com o mundo físico, e é este mundo físico que define os requisitos temporais. Por exemplo, a dinâmica de um motor elétrico define o período da sua tarefa de controle, o funcionamento do ouvido humano define o período da tarefa que reproduz um áudio.

O mundo físico é intrinsecamente paralelo. Várias coisas acontecem simultaneamente e, muitas vezes, a aplicação de tempo real precisa reagir a várias delas. Por exemplo, a aplicação precisa, ao mesmo tempo, controlar o motor elétrico, registrar informações em arquivos e interagir com um operador humano através de tela e teclado. Neste caso, fica muito mais fácil dividir a aplicação em várias tarefas de software, cada uma cuidando de um aspecto da aplicação. Tais tarefas não são independentes. Elas precisam colaborar entre si, ou seja, é preciso existir sincronização e comunicação (troca de dados) entre as tarefas.

Este capítulo trata dos problemas e soluções relacionados com a sincronização e comunicação entre tarefas. Inicialmente o termo Programação Concorrente é definido e são apresentados cenários onde seu emprego é vantajoso. A sincronização e comunicação entre tarefas pode ser feita através da troca de mensagens ou do acesso a variáveis compartilhadas. As duas formas são apresentadas e discutidas. No [capítulo 7](#) serão apresentados mecanismos criados especialmente para sistemas de tempo real, enquanto no [capítulo 8](#) serão tratadas situações mais complexas de sincronização.

## 7. Seções Críticas em Sistemas de Tempo Real

*É possível melhorar o mutex para uso em sistemas de tempo real ?*

O [capítulo 6](#) apresentou alguns problemas e soluções relacionados com a sincronização e comunicação entre tarefas. Programação Concorrente foi definida e mostrado como a sincronização e comunicação entre tarefas pode ser feita através da troca de mensagens ou do acesso a variáveis compartilhadas. No caso de variáveis compartilhadas, foi apresentado o problema da seção crítica e alguns mecanismos de sincronização possíveis, notadamente o mutex.

O mutex tradicional, apresentado no [capítulo 6](#), resolve o problema da seção crítica em sistemas de propósito geral. Entretanto, em sistemas de tempo real, ele pode gerar situações onde ocorre uma demorada inversão de prioridades entre as tarefas. Em sistemas de tempo real é possível empregar mutex especiais, os quais são projetados exatamente para minimizar as situações de inversão de prioridades, as quais são prejudiciais para a qualidade temporal do sistema.

Neste capítulo será discutido o problema da inversão de prioridades e apresentados os principais protocolos de alocação de recursos usados em versões do mutex especialmente criadas para sistemas de tempo real.

## 8. Mecanismos de Sincronização com Variáveis Compartilhadas

*Que outros problemas de sincronização existem com variáveis compartilhadas ?*

No capítulo 6, sobre a sincronização e comunicação entre tarefas, foi apresentado o problema da seção crítica quando variáveis compartilhadas são usadas. Naquele capítulo também foram descritos mecanismos de baixo nível para lidar com este problema, e o tipo abstrato de dado mutex, o qual é amplamente empregado. No [capítulo 7](#) foram apresentadas as versões do mutex para sistemas de tempo real.

Embora o problema da seção crítica seja realmente o problema de sincronização mais frequente com variáveis compartilhadas, ele não é o único. Neste capítulo são apresentados exemplos de outras situações onde é necessário sincronizar tarefas que colaboram através de variáveis compartilhadas, em cenários mais complexos que a simples seção crítica. Neste caso, mecanismos adicionais além do mutex são necessários. Especificamente, são apresentados os semáforos e os monitores. Para o caso dos monitores são apresentados dois exemplos baseados em situações que surgem no contexto das aplicações de controle e automação.

## 9. O Tempo Real

*O que é o tempo real ?*

O tempo é um conceito interessante. Todo mundo sabe o que é o tempo, até o momento que precisamos defini-lo com precisão. Foge dos propósitos deste livro tentar definir o tempo em termos da física teórica, históricos ou mesmo filosóficos. Estamos interessados no tempo real, o qual corresponde ao tempo do mundo físico, o tempo do relógio da parede. É diferente, por exemplo, do tempo simulado como temos em simulações meteorológicas ou em videogames como Civilização de Sid Meier. Também é diferente do tempo lógico usado em alguns sistemas distribuídos para estabelecer uma ordem entre eventos, porém numericamente dissociada do tempo físico.

Vamos nos ater ao objetivo de medir a passagem do tempo físico. Um relógio é qualquer dispositivo usado para medir a passagem do tempo. Todo relógio consiste de um oscilador mais ou menos estável e uma forma de contar os ciclos deste oscilador. Mesmo este propósito simples teve uma evolução constante nos últimos séculos, com alguns saltos de vez enquanto. As várias seções deste capítulo descrevem como o conceito de segundo foi definido ao longo do, sim, do tempo.

Para entender a padronização do tempo é importante ter em mente duas coisas. Primeiramente, este não foi um desenvolvimento linear, uma simples evolução. Na verdade muitas diferentes tecnologias habilitadoras e técnicas específicas foram desenvolvidas concorrentemente, muitas vezes competindo entre si, até chegarmos ao estado de coisas que temos hoje. Outro fato importante a ser considerado é que a padronização do tempo ocorre em diferentes instituições e organizações, para diferentes propósitos. Desta forma, a adoção desta ou daquela definição de “segundo” pode ocorrer em diferentes momentos, conforme a organização de padronização que consideramos.

Neste capítulo será discutido o conceito de tempo real, como ele é definido em termos astronômicos e em termos atômicos, como é padronizado através da UTC, como pode ser obtido via GPS ou NTP e, finalmente, como é mantido nos computadores através de osciladores baseados em cristais de quartzo.

## 10. Variância dos Tempos de Execução

*Por que o tempo de execução de uma tarefa varia ?*

Quando um programador implementa, por exemplo, uma função na linguagem C, ele tende a pensar que esta função, ao executar, demorará um tempo de execução mais ou menos constante. Na verdade, a realidade é completamente outra. Diversos aspectos inerentes aos programas de computador (software), somados às características dos processadores modernos (hardware), faz com que o tempo de execução de uma função C pareça mais uma variável aleatória do que uma constante. E isto vale não apenas para a linguagem de programação C, mas para qualquer linguagem de programação.

Neste capítulo vamos examinar quais são os aspectos construtivos do hardware e do software que fazem os tempos de execução das tarefas variarem, mesmo que elas estejam completamente sozinhas no computador. Um aspecto semelhante porém diferente, a variância do tempo de resposta, será discutida no [capítulo 13](#). A diferença entre tempo de execução e tempo de resposta foi apresentada no [capítulo 2](#) e será revisada na próxima seção.

## 11. Estimação do WCET usando Análise Estática

*Quais são as técnicas para estimar analiticamente o tempo de execução no pior caso ?*

Sistemas de tempo real são classificados de acordo com a criticalidade dos requisitos temporais em sistemas críticos (*hard real-time*) e sistemas não críticos (*soft real-time*). Para sistemas críticos é necessário oferecer garantias quanto ao atendimento dos deadlines. Para isto é necessário conhecer o tempo de execução no pior caso (WCET – *Worst Case Execution Time*) de cada tarefa. Mesmo para sistemas não críticos esta é uma informação relevante.

Neste capítulo é mostrado como Análise Estática pode ser usada na obtenção de uma estimativa que represente um limite superior seguro (*upper-bound*) para o WCET, ou seja, uma estimativa que será sempre igual ou maior do que o verdadeiro valor do WCET.

## 12. Estimação do WCET usando Medições

*Quais são as técnicas para estimar com medições o tempo de execução no pior caso ?*

Para sistemas críticos de tempo real é necessário conhecer o tempo de execução no pior caso de cada tarefa (WCET – *Worst Case Execution Time*). Entretanto, mesmo para sistemas não críticos, uma estimativa do WCET é relevante. Ela permite ao projetista do sistema antecipar os piores cenários com respeito ao cumprimento dos requisitos temporais.

O tempo de execução no pior caso acontece através da combinação dos piores comportamentos para o software e para o hardware. Ao mesmo tempo, a demanda crescente por capacidade de processamento nos sistemas de tempo real requer o uso de arquiteturas de computador modernas e complexas, onde a análise estática do WCET é muito difícil ou impossível. Neste capítulo é mostrado como medições do tempo de execução podem ser usadas na obtenção de uma estimativa que represente um limite superior seguro (*upper-bound*) para o WCET, ou pelo menos uma estimativa aproximada, mesmo que insegura.

## 13. Variância dos Tempos de Resposta

*Por que o tempo de resposta de uma tarefa varia ?*

Como descrito no [capítulo 2](#), sobre os conceitos básicos dos sistemas de tempo real, o tempo de execução (*execution time*) de uma tarefa considera que ela está sozinha no sistema, ao passo que o tempo de resposta (*response time*) da tarefa é definido pelo intervalo de tempo entre a chegada da tarefa e a sua conclusão. O tempo de resposta inclui eventuais intervalos de tempo quando a tarefa é suspensa para a execução de outras atividades do sistema, tais como tratadores de interrupções e outras tarefas. O tempo de resposta é muito importante pois é ele que deverá ser menor ou igual ao deadline da tarefa, caso esta possua um deadline.

O [capítulo 3](#) e o [capítulo 4](#) descreveram formas usadas na implementação de tarefas de tempo real em sistemas de software. A **variância dos tempos de resposta** está diretamente associada com o design do software do sistema (*system software design*). Ainda que, hipoteticamente, o tempo de execução de uma tarefa fosse constante, na grande maioria dos sistemas o seu tempo de resposta variaria, em função das outras atividades do sistema.

Neste capítulo será descrito como estas outras atividades do sistema somam no tempo de resposta de uma tarefa, fazendo-o variar. Uma questão igualmente relevante, a determinação do tempo de resposta no pior caso (WCRT – *Worst-Case Response Time*), será tratada no [capítulo 14](#).

## 14. Estimação do Tempo de Resposta no Pior Caso

*Como estimar o tempo de resposta máximo que uma tarefa pode apresentar ?*

Como visto no [capítulo 2](#), sobre conceitos básicos do escalonamento tempo real, o requisito temporal mais comum é o deadline da tarefa. O **tempo de resposta** (*response time*) de uma tarefa, o qual é definido pelo intervalo de tempo entre a chegada da tarefa e a sua conclusão, deve ser menor ou igual ao seu deadline relativo. O deadline é dito relativo pois o mesmo é medido com relação ao instante de chegada da tarefa. O tempo de resposta inclui o tempo de execução da tarefa, mais os eventuais intervalos de tempo quando a mesma foi suspensa para a execução de outras atividades do sistema, tais como tratadores de interrupções ou outras tarefas de prioridade mais alta.

O [capítulo 13](#) mostrou como o tempo de resposta de uma mesma tarefa varia de ativação para ativação. Se pudessemos determinar, de alguma maneira, o **tempo de resposta no pior caso** (WCRT – *Worst-Case Response Time*) da tarefa, bastaria comparar este valor com o deadline relativo. Se o tempo de resposta no pior caso for menor ou igual ao deadline relativo, esta tarefa jamais perderá um deadline.

A determinação do tempo de resposta no pior caso é o assunto deste capítulo. A técnica conhecida como **Análise do Tempo de Resposta** (RTA – *Response-Time Analysis*) procura fazer exatamente isto para sistemas escalonados com prioridades preemptivas. A partir de uma descrição das características do sistema e de suas tarefas, métodos algébricos são usados para estimar um valor que é igual ou superior ao verdadeiro WCRT de cada tarefa. Este valor pode ser então comparado com o deadline relativo da respectiva tarefa para mostrar que deadlines jamais serão perdidos.

Quando o sistema é muito complexo, como por exemplo aqueles baseados em Linux, a aplicação de RTA não é viável. Nestes casos, dados estatísticos podem ser coletados para caracterizar o



comportamento de cada tarefa. Em muitos sistemas a perda de deadlines é tolerada, desde que não ocorra com grande frequência. Na parte final deste capítulo são apresentados alguns critérios que podem ser usados no momento de avaliar a qualidade temporal deste tipo de sistema.

## 15. Diferentes Abordagens

*Como são verificados os requisitos temporais ?*

No [capítulo 1](#) deste livro, onde Sistemas de Tempo Real foram caracterizados, foi possível perceber o quão amplo é o mercado dos sistemas de tempo real. As aplicações com requisitos temporais variam enormemente e diferentes abordagens são necessárias para lidar com os sistemas de tempo real. Este capítulo descreve em linhas gerais as principais abordagens encontradas para a verificação de requisitos temporais. Classificações semelhantes podem ser encontradas em muitos livros, tais como [LIU2000], [BUR2009] e [BUT2011]. Infelizmente, não existe uma classificação única aceita por todos os profissionais da área. Além disto, existem diferenças entre a classificação aceita usualmente no meio acadêmico e a prática da engenharia de sistemas de tempo real.

## 16. Sistemas Operacionais de Tempo Real

*O que é um sistema operacional de tempo real ?*

Como visto no [capítulo 3](#) e no [capítulo 4](#), os quais trataram da implementação de tarefas de tempo real, a maioria das aplicações são construídas a partir dos serviços oferecidos por um microkernel ou kernel de sistema operacional. Em função disto, o atendimento dos requisitos temporais da aplicação depende não somente do código da aplicação, mas também do comportamento do sistema operacional usado.

Muitas vezes os requisitos temporais da aplicação são tão rigorosos ou a plataforma de hardware é tão limitada que o sistema operacional é substituído por um executivo cíclico, ou um laço principal auxiliado por tratadores de interrupção. No entanto, a maioria das aplicações de tempo real utilizam algum tipo de sistema operacional, podendo ser um microkernel simples, um kernel completo, ou algo entre estes extremos.

Este capítulo apresenta o conceito de sistema operacional de tempo real, ou seja, sistemas operacionais cujo propósito é suportar aplicações de tempo real. Inicialmente são descritas algumas demandas específicas das aplicações de tempo real, as quais não são usualmente consideradas em sistemas operacionais de propósito geral. Mais adiante no capítulo são apresentadas duas soluções de código livre existentes, as quais empregam abordagens bem distintas para lidar com a questão.

É preciso ter em mente, ao longo deste capítulo, que idealmente o que o desenvolvedor de uma aplicação de tempo real quer é determinar o tempo de resposta no pior caso das tarefas de tempo real, como discutido no [capítulo 14](#). Porém, como ficará claro, esta não é uma missão fácil quando um sistema operacional está envolvido. O capítulo termina com uma seção onde são feitas considerações gerais sobre esta questão.

## 17. Tópicos Adicionais

*Que outros temas relativos a sistemas de tempo real podem ser estudados ?*

A caracterização dos sistemas de tempo real apresentada no [capítulo 1](#) mostrou como são variadas as classes de aplicações que incluem requisitos temporais. Uma consequência disto é a possibilidade de visitar praticamente todas as áreas da computação com um olhar de “tempo real”. Por exemplo, existem trabalhos sobre banco de dados para tempo real [BES1997] ou processamento de imagens em tempo real [VIO2004].

Ao longo dos capítulos deste livro foram apresentados vários aspectos dos sistemas de tempo real. Principalmente as questões centrais ligadas à implementação de tarefas, seu escalonamento e a verificação do cumprimento dos requisitos temporais. É impossível cobrir em um só livro todos os aspectos dos sistemas de tempo real tratados na literatura científica.

Neste capítulo são apresentados vários tópicos adicionais, ou quais são brevemente descritos. O objetivo é mostrar que existem trabalhos importantes sobre estes tópicos. O leitor especialmente interessado em um dos tópicos descritos neste capítulo poderá buscar mais informações na literatura especializada da área em questão.