

## **Conjunto de Experiências de Baixo Custo para o Ensino de Sincronização de Relógios**

Rogério Leite Alves Pinto, Carlos Eduardo Manchini e Rômulo Silva de Oliveira  
Departamento de Automação e Sistemas  
Universidade Federal de Santa Catarina  
Caixa Postal 476, Florianópolis-SC, 88040-900, Brasil  
lapinto@matrix.com.br, toctoc@ligbr.com.br e romulo@das.ufsc.br

### **Resumo**

*Um dos aspectos fundamentais dos sistemas de tempo real é a capacidade de medir a passagem do tempo e associar timestamps com eventos específicos. No caso de sistemas distribuídos de tempo real, a sincronização entre relógios é importante para que os timestamps gerados em diferentes computadores possam ser considerados juntos. Na maioria dos cursos de graduação aulas de laboratório são utilizadas para apresentar aos alunos os conceitos fundamentais de diversas matérias. O objetivo deste artigo é descrever um conjunto de experiências de baixo custo, criadas com o propósito de apresentar aos alunos de graduação os conceitos fundamentais da sincronização de relógios.*

### **Abstract**

*A fundamental aspect of real-time systems is the ability to measure time intervals and to associate timestamps with specific events. Considering distributed real-time systems, clock synchronization is important to assure that timestamps generated at different computers are consistent. Most under-graduate courses have laboratory classes to present the fundamental concepts of many subjects. The goal of this paper is to present a set of low cost experiences created with the purpose of presenting to under-graduate students the fundamental concepts of clock synchronization.*

### **1. Introdução**

Um dos aspectos fundamentais dos Sistemas de Tempo Real é a capacidade de medir a passagem do tempo e associar *timestamps* com eventos específicos. Muitas aplicações utilizam esta capacidade para gerar históricos de eventos e/ou disparar eventos em momentos previamente determinados. No caso de sistemas distribuídos de tempo real, a sincronização entre relógios é importante para que os *timestamps* gerados em diferentes computadores possam ser considerados juntos [4]. A literatura sobre sincronização de relógios é extensa e a área de pesquisa bastante produtiva (por exemplo [2] e [6]).

Na maioria dos cursos de graduação, e na totalidade dos cursos de graduação na área das ciências exatas, aulas de laboratório são utilizadas para apresentar aos alunos os conceitos fundamentais de diversas matérias. Por exemplo, laboratórios de física e de eletrônica são recursos fundamentais empregados pelos educadores para cristalizar nos alunos conceitos como ótica e amplificadores operacionais, respectivamente.

Apesar da importância da sincronização de relógios para as áreas de tempo real e sistemas distribuídos, não é do conhecimento dos autores conjuntos de experiências especialmente criados para o ensino desta matéria. Em particular, dada a dificuldade financeira crônica das

universidades brasileiras, um conjunto de experiências de baixo custo, que dispense equipamentos interessantes porém caros, como receptores GPS de tempo [1].

O objetivo deste artigo é descrever um conjunto de experiências de baixo custo, criadas com o propósito de apresentar aos alunos de graduação os conceitos fundamentais da área de sincronização de relógios. A seção 2 deste artigo descreve o instrumento de medição usado, elemento fundamental em todas as experiências. A seção 3 descreve as experiências como apresentadas aos alunos. Finalmente, a seção 4 contém as considerações finais, incluindo um breve relato do uso deste material com uma turma de alunos na UFSC.

## 2. Instrumento de Medição

O instrumento básico empregado nas aulas de laboratório de sincronização de relógios é usado para medir a diferença na hora entre dois computadores diferentes (*skew*). Para medir o erro entre os relógios pode-se ler a hora dos dois computadores diferentes ao mesmo tempo e fazer a diferença. Uma forma de garantir que a hora será lida ao mesmo tempo nos dois computadores é gerar uma interrupção de hardware simultaneamente nos dois processadores. Basta fazer o sistema operacional informar na tela a hora local, no instante seguinte ao atendimento à interrupção e, após, calcular a diferença (em segundos) entre as horas informadas pelos computadores.

A forma adotada para gerar uma interrupção simultânea nos dois computadores foi a transmissão serial de dados [5]. A escolha decorre da simplicidade da solução e do fato de todo computador possuir porta serial, quase sempre disponível. Além disso, o custo do instrumento limita-se a confecção de um cabo serial, algo entre R\$10,00 e R\$ 20,00.

Na comunicação serial (figura 1) os bits são transferidos um a um, através de um único par condutor. Os bytes a serem transmitidos são serializados, isto é, são "desmontados" bit a bit, e são individualmente transmitidos, um a um. Na outra extremidade do condutor, os bits são contados e quando formam 8 bits, são remontados, reconstituindo os bytes originais. Como todos os bits são transferidos pelo mesmo meio físico (mesmo par de fios), as eventuais irregularidades afetam todos os bits igualmente.



Figura 1: Comunicação serial.

Para fazer um computador se comunicar com o outro via serial, não basta apenas ligar o cabo serial, é necessário ter um programa que utilize as portas seriais e faça a interface entre eles. Um programa que apenas envia e recebe caracteres já é suficiente neste trabalho. Em nossas experiências vamos utilizar um programa chamado "Term", que necessita de apenas um parâmetro, que é justamente a porta serial que irá ser utilizada.

Exemplo: *TERM COM1*

As portas seriais comuns COM1 e COM2 podem ser de 9 ou 25 pinos, sendo a COM1 geralmente usada pelo mouse. Atualmente existem outros tipos de porta serial, como a porta PS/2 (usada pelos mouses mais novos) e a USB (Universal Serial Bus). Nas nossas experiências vamos utilizar a porta serial RS-232 (COM1/COM2) para fazer a comunicação entre os dois computadores. A figura 2 mostra os conectores.

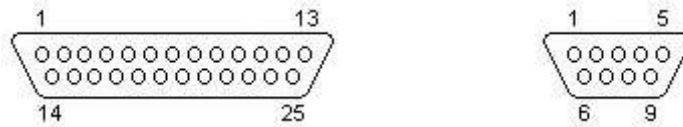


Figura 2: Conectores empregados.

### 2.1 Cabo serial cruzado com *loop-back*

Embora a conexão serial possa ter tanto 9 pinos quanto 25 pinos, somente 9 pinos são normalmente utilizados [7]:

Sigla	Pinos D-9	Pinos D-25	Descrição
TxD	3	2	linha usada para transmitir os dados
RxD	2	3	linha usada para receber os dados
RTS	7	4	usado pelo <i>Emissor</i> para pedir permissão ao <i>Receptor</i> p/enviar dados
CTS	8	5	usado pelo <i>Receptor</i> para avisar o <i>Emissor</i> que está apto a receber dados
DSR	6	6	utilizado pelo <i>Receptor</i> para comunicar ao <i>Emissor</i> que está pronto para estabelecer comunicação
GND	5	7	signal de referência tanto para o <i>Emissor</i> quanto para o <i>Receptor</i>
DCD	1	8	usado pelo modem para indicar uma conexão com outro modem
DTR	4	20	utilizado pelo <i>Emissor</i> para indicar ao <i>Receptor</i> que está ligado e pronto para estabelecer comunicação
RI	9	22	indicador de chamada, feita por outro modem

A figura 3 mostra a configuração de um cabo serial cruzado, que é utilizado para comunicação entre dois computadores. Os dois computadores podem tanto enviar quanto receber dados.

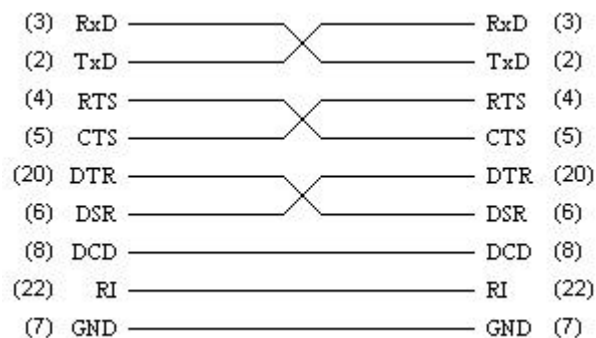


Figura 3: Esquema do cabo cruzado.

No caso deste trabalho, a comunicação é mais complexa. Queremos que um dos computadores envie um caractere qualquer para ele mesmo, além de enviar para o outro. Para isto, necessitamos de uma configuração especial do cabo serial, que é chamada de *loop-back*. Tal configuração permite que o Emissor, além de enviar dados para o Receptor, envie também os mesmos dados para si próprio, de tal maneira a gerar a interrupção nos dois processadores ao mesmo tempo. A figura 4 apresenta um esquema ilustrativo da ligação *loop-back*. O pino “Tx” do Receptor é cortado, pois se mantivermos esta ligação, o “Rx” do Emissor poderá receber dados ao mesmo tempo do Receptor e do próprio Emissor. Haveria um conflito de informações em “Rx”, o que não é desejável que ocorra.

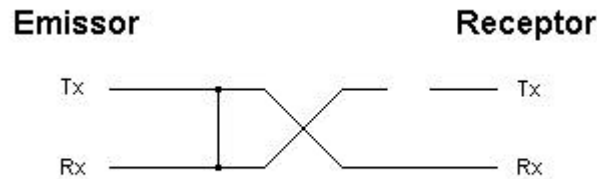


Figura 4: Ligação em loop-back.

## 2.2 Tempo de propagação no cabo

O tempo de propagação da informação a ser transmitida pelo cabo serial é bastante relevante no caso que estamos estudando, visto que apesar da velocidade da luz ser bastante grande, ela não é infinita. Vamos tomar como exemplo o nosso caso onde usamos o cabo com *loop-back* (figura 4). Sabe-se que a informação parte de “Tx” do Emissor e vai até o “Rx” do próprio Emissor (que vamos chamar de “Rx1”) e até o “Rx” do Receptor (que vamos chamar de “Rx2”). A distância a ser percorrida pelo sinal, entre “Tx” e “Rx1”, é menor do que a distância entre “Tx” e “Rx2”, e depende das medidas do cabo.

Suponha que a distância entre “Tx” e “Rx1” seja igual a 15cm, e a distância entre “Tx” e “Rx2” seja igual a 180cm. Agora, queremos ler a hora dos dois computadores. Então enviamos um caractere ao mesmo tempo para os dois computadores, sinalizando que a hora local deve ser lida. Devido ao tempo de propagação no cabo, o Emissor vai ler a hora antes do Receptor. A velocidade da luz sendo aproximadamente 300.000Km/s, ela percorre 15cm em 0,5ns e 180cm em 6ns (figura 5). Nas experiências propostas este valor é muito menor que a precisão aumejada pela medição, logo o limite da velocidade da luz pode ser ignorado.

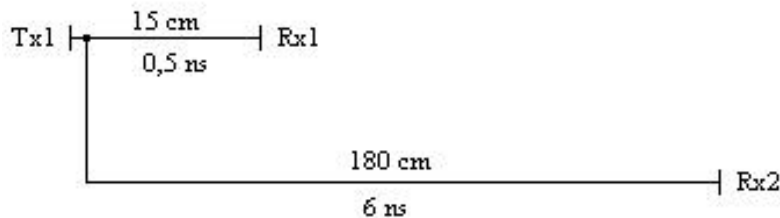


Figura 5: Atraso no cabo.

### 3. Experiências

Esta seção do artigo descreve as diversas experiências sugeridas aos alunos. Para cada uma delas o artigo traz o objetivo, uma breve explicação, e as instruções apresentadas aos alunos. Uma descrição mais detalhada pode ser encontrada no relatório técnico associado [3], inclusive com resultados numéricos para os experimentos.

#### 3.1 Experiência 1: Introdução à Sincronização de Relógios

O objetivo desta experiência é ganhar familiaridade com o instrumento de medição. Isso inclui estudar o funcionamento da porta serial, conhecer o uso do cabo serial cruzado com *loop-back* e a pinagem dos conectores. Para tanto o aluno deve medir a diferença entre os relógios de dois computadores.

##### Instruções:

1. Montar a ligação serial entre os dois computadores;
2. Iniciar o computador no modo Windows e abrir uma janela do MS-DOS;
3. Executar o programa "Term <porta>" (com1 ou com2) nos dois PCs;
4. Medir a hora dos dois computadores 10 vezes, intervalo de 30 segundos entre medidas;
5. Calcular a média da diferença entre os relógios e o desvio padrão.

#### 3.2 Experiência 2: Interferência da Inicialização (*Boot*) no Relógio do Computador

O objetivo desta experiência é observar a perturbação introduzida pelo processo de inicialização do sistema operacional. Nos computadores existe um cristal usado no relógio que mantém a hora permanentemente. Entretanto, na inicialização o sistema operacional lê a hora do relógio permanente e passa a usar o temporizador para manter esta hora atualizada. A hora atualizada é descartada quando o computador é desligado. Esta troca de cristais durante a inicialização faz com que ocorram variações, muitas vezes até bruscas, na hora informada. Com isso, não se pode afirmar com certeza para qual valor o relógio "pulará", mas é certo que o *boot* invalida as medidas feitas anteriormente.

##### Instruções:

1. Montar a ligação serial entre os dois computadores;
2. Iniciar os computadores no modo MS-DOS;
3. Ajustar os dois computadores para ter uma diferença de aproximadamente 5 segundos entre eles usando o comando time do DOS;
4. Executar o programa "Term <porta>" (com1 ou com2) nos dois PCs;
5. Medir a hora dos dois computadores;
6. Calcular a diferença entre os relógios;
7. Realizar o boot na máquina 1 e voltar para o modo DOS;
8. Repetir os passos 4, 5 e 6.

#### 3.3 Experiência 3: Medição da Taxa de Afastamento (*Drift-Rate*)

O objetivo desta experiência é medir o afastamento dos relógios entre dois computadores. O *drift-rate* nada mais é do que a taxa de afastamento dos relógios. Neste tipo de medição deve-se considerar a hora de um dos computadores como a hora de referência, e a hora do outro computador como a hora se afastando da hora de referência.

O que ocasiona a existência do *drift-rate* é o erro existente na frequência de oscilação dos cristais. É praticamente impossível obter dois cristais exatamente iguais, o que causa uma certa diferença na frequência de oscilação dos mesmos. Tal diferença, por menor que seja, é acumulada ao longo do tempo, e a medida que o tempo passa o valor do erro aumenta.

O processo de medição da taxa de afastamento dos relógios é dado da seguinte maneira: é medida a hora nos dois computadores várias vezes com um intervalo de tempo entre as medidas de aproximadamente uma hora. Com estes dados, podemos calcular a diferença entre os relógios no início da medição (na primeira medida) e no final da medição (na última medida). A partir daí podemos ver o quanto os relógios se afastaram neste intervalo de tempo entre as medidas. É importante lembrar que para medir a taxa de afastamento dos relógios não pode-se reiniciar um dos computadores, visto que o *boot* afeta as medições.

Instruções:

1. Montar a ligação serial entre os dois computadores;
  2. Iniciar os computadores no modo MS-DOS;
  3. Ajustar os dois computadores para ter uma diferença de aproximadamente 5 seg. entre eles usando o comando *time* do DOS;
  4. Executar o programa "Term <porta>" (com1 ou com2) nos dois PCs;
  5. Medir a hora dos dois computadores;
  6. Calcular a diferença entre os relógios;
  7. Aguardar uma hora e repetir os passos 5 e 6;
- Obs: Deve-se obter pelo menos 5 medidas, cada uma delas com intervalo de uma hora.
8. Calcular a diferença entre os relógios em todos os intervalos de tempo;
  9. Calcular, em segundos, os intervalos de tempo entre as medidas;
  10. Calcular para cada intervalo de tempo, o *drift-rate* (em  $\mu\text{s/s}$ );
  11. Calcular a média e o desvio padrão com os dados do item 10.

### **3.4 Experiência 4: Diferença Entre Sincronização Manual e Automática**

O objetivo desta experiência é mostrar ao aluno a diferença entre a sincronização manual e automática, calculando o erro médio de cada tipo de sincronização. Nos filmes de ação é comum os personagens ajustarem os seus relógios com uma mesma hora, e em seguida "dar a partida" neles ao mesmo tempo. Assim, eles saberão a hora exata que devem ser feitas as coisas. Na sincronização manual a proposta é semelhante, sincronizar os relógios dos dois computadores com o uso do comando *Time* do MS-DOS. Vamos simplesmente ajustar a mesma hora nos dois computadores, e apertar a tecla *Enter* simultaneamente.

Na sincronização automática procura-se diminuir o efeito do erro causado pelo tempo de reação no momento de apertar a tecla *Enter*. Para isto, uma pequena modificação é feita no programa *Term*, que agora se chama *Sync*. Ao invés de "dar a partida" nos relógios usando dois teclados, o mesmo é feito usando apenas um teclado. Ao digitar um caractere no teclado de um dos computadores, este caractere será enviado aos dois computadores ao mesmo tempo via cabo serial. Assim que o computador recebe o caractere, o mesmo ajusta a hora do seu computador. Para isto, foi utilizada a chamada de sistema *settime()* no programa.

Assim como usando o comando *Time* do DOS é necessário informar a hora desejada, no *Sync* o mesmo acontece. Isto é passado para o programa como um parâmetro na chamada do mesmo. Por exemplo, para que o computador, quando receber o caractere da porta serial, ajuste a hora para 14:35:00,00 deve-se digitar:

*Exemplo:* SYNC COM2 14 35

Instruções:

1. Montar a ligação entre os dois computadores;
2. Iniciar os computadores no modo MS-DOS;
3. Digite *Time* nos dois PCs, e ajuste-os com a mesma hora (sem apertar o *Enter*);
4. Aperte a tecla *Enter* dos dois PCs ao mesmo tempo;

5. Execute o programa *Term* e meça a diferença entre os relógios;
6. Repita os passos 3, 4 e 5 por 5 vezes;
7. Calcule a média e o desvio padrão da diferença entre os relógios;
8. Executar o programa “Sync <porta> <hora> <minuto> <segundo>” nos dois computadores;
- OBS1: Ajuste os relógios com a mesma hora também;
9. Apertar uma tecla do computador *Emissor* (o PC que emite os caracteres) para “dar a partida nos relógios”;
- OBS2: Depois de dois caracteres recebidos, o programa volta a funcionar como o *Term*, para que você possa medir a hora dos computadores e calcular a diferença que há entre eles;
10. Meça a diferença entre os relógios;
11. Repita os passos 8, 9 e 10 por 5 vezes;
12. Calcule a média e o desvio padrão da diferença entre os relógios;
13. Compare a diferença entre os dois métodos;
- OBS3: As medidas devem ser feitas logo em seguida ao *Set Time*, e sem reiniciar o computador, para eliminar o efeito do *Drift-Rate* e do *boot*.

### 3.5 Experiência 5: Sincronização Via Internet

O objetivo desta experiência é realizar uma sincronização externa, mostrando ao aluno a diferença entre sincronização interna e externa. Na experiência anterior havia duas maneiras de sincronizar relógios. Tanto a sincronização manual quanto a sincronização automática mostradas são consideradas sincronização interna. A sincronização externa consiste em sincronizar relógios com base em um relógio externo padrão.

Existem diversos meios de se obter a hora padrão mundial (UTC). A mais precisa delas e a mais confiável, é com o auxílio do GPS (*Global Positioning System* [1]). Com o propósito de manter baixo o custo das experiências pode-se sincronizar via Internet, onde é possível obter a hora UTC de forma mais simples e suficientemente confiável para muitas aplicações.

Para sincronizar via Internet vamos utilizar um programa chamado *Atom Time* [8], que sincroniza automaticamente o relógio do computador com o relógio de referência UTC. Para isto, basta estar conectado à Internet e ter configurado o relógio do Windows corretamente. Para configurar o relógio do Windows deve-se entrar nas propriedades do relógio e configurar o fuso horário no qual você se encontra (exemplo: GMT-03:00 - Brasília), e caso esteja no período de horário de verão, esta opção deve ser marcada também. Na experiência são sincronizados dois relógios de diferentes computadores via Internet, e depois é medida a diferença que há entre os relógios dos dois computadores.

#### Instruções:

1. Conectar os dois computadores à internet;
2. Executar o programa *Atom Time* nos dois computadores;
3. Abrir uma janela do MS\_DOS e executar o programa "Term" nos dois PCs;
4. Medir a diferença entre os relógios;
5. Repetir os passos 2,3 e 4 por 5 vezes;
6. Calcular a média e o desvio padrão da diferença entre os relógios;
- Obs1: Não se esqueça de verificar as configurações do relógio do windows;
- Obs2: No programa *Atom Time* deve-se marcar a opção *Local Time* (dentro de *settings*).

### 4. Conclusões

Este artigo descreveu um conjunto de experiências de baixo custo cujo propósito é auxiliar no ensino dos conceitos fundamentais da sincronização entre relógios. Além de microcomputadores convencionais, o único recurso de hardware utilizado foi um cabo serial

especialmente preparado para este fim. Embora o instrumento de medição utilizado seja aproximado, pois está sujeito às interferências do sistema operacional, sua precisão é suficiente para o seu propósito: uma ferramenta para ensino de sincronização de relógios.

As experiências propostas fazem o aluno tomar contato com as diferenças de hora entre computadores, o efeito da inicialização do sistema operacional, o *drift-rate*, a sincronização manual e automática, além da sincronização externa via Internet. Além das experiências propriamente ditas, os alunos também recebem um texto [3] contendo os fundamentos teóricos. Esse texto descreve a forma como é feita a medição do tempo, o relógio atômico, a *Universal Coordinated Time* (UTC), como é o funcionamento dos relógios e dos cristais no computador, e a comunicação entre computadores via porta serial, entre outros assuntos. O relatório técnico contendo a fundamentação teórica e as experiências para os alunos pode ser obtido diretamente (<http://www.lcmi.ufsc.br/~romulo/reltec/sincronizacao.pdf>) .

As experiências foram propostas para MSDOS por questões locais, mas podem ser facilmente portadas para Linux, por exemplo. Um aspecto importante a ser abordado no futuro é o efeito do sistema operacional sobre as medidas realizadas. No caso do MSDOS o maior problema é a resolução do seu relógio (cerca de 55ms). No caso do Linux, as interferências, os bloqueios no kernel e os momentos com interrupções desabilitadas vão diminuir a precisão da medida.

O conjunto de experiências apresentado neste artigo foi utilizado pelos alunos da disciplina “Informática Industrial II” do curso de Engenharia de Controle e Automação da UFSC, no semestre 2001/2. Embora esta seja uma afirmação subjetiva, os autores sentem-se confortáveis em afirmar que os conceitos fundamentais da sincronização de relógios são muito mais facilmente aprendidos através do uso destas experiências do que através da simples exposição teórica em sala de aula.

## Referências

- [1] Peter H. Dana. Global Positioning System (GPS) Time Dissemination for Real-Time Applications. Real-Time Systems, N. 12, The Netherlands, Kluwer Academic Publishers, 1997.
- [2] C. Fetzer, F. Cristian. Integrating External and Internal Clock Synchronization. The Real-Time Systems Journal, pp.123-171, dez 1997.
- [3] Carlos Eduardo Manchini, Rogério Leite Alves Pinto e Rômulo Silva de Oliveira. Conjunto de Experiências Sobre Sincronização de Relógios. Relatório Técnico DAS-CTC-UFSC (<http://www.das.ufsc.br/~romulo/reltec/sincronizacao.pdf>), dezembro de 2001.
- [4] Andrew S. Tanenbaum. Distributed Operating Systems. Prentice-Hall, 1995.
- [5] Andrew S. Tanenbaum. Computer Networks. Prentice Hall, Third Edition, EUA, 1996.
- [6] P. Veríssimo, L. Rodrigues, A. Casimiro. CesiumSpray: A Precise and Accurate Global Time Service for Large-scale Systems. The Journal of Real-Time Systems, 12, pp.243-294, December 1997.
- [7] <http://www.ctv.es/pckits/tpserie.html>
- [8] <http://www.atomtime.com>