

**DAS** Departamento de Automação e Sistemas  
**CTC** **Centro Tecnológico**  
**UFSC** Universidade Federal de Santa Catarina

# **Conjunto de Experiências Sobre Sincronização de Relógios**

## **Relatório Técnico**

Carlos Eduardo Manchini

Rogério Leite Alves Pinto

Rômulo Silva de Oliveira

Florianópolis, Dezembro de 2001.

# SUMÁRIO

SUMÁRIO .....	2
Capítulo 1 – Introdução à Medição do Tempo .....	3
1.1 SOL .....	3
1.2 RELÓGIO ATÔMICO E TAI .....	3
1.3 UTC.....	4
1.4 MEIOS DE OBTENÇÃO PRECISA DO TEMPO .....	4
Capítulo 2 – Medição do Tempo no Computador .....	5
2.1 CRISTAL DE QUARTZO .....	5
2.2 RELÓGIOS DIGITAIS .....	5
2.3 RELÓGIO PERFEITO .....	5
2.4 TIPOS DE SINCRONIZAÇÃO .....	6
2.5 APLICAÇÕES .....	6
Capítulo 3 – Medição do Erro Entre os Relógios de Dois Computadores.....	8
3.1 MEDIÇÃO DO ERRO .....	8
3.2 COMUNICAÇÃO ENTRE COMPUTADORES.....	8
3.3 PORTA SERIAL .....	9
3.4 CABO SERIAL CRUZADO COM LOOP-BACK.....	9
3.4.1 TEMPO DE PROPAGAÇÃO NO CABO.....	11
Experiência – Introdução à Sincronização de Relógios.....	12
Capítulo 4 – Impacto do Boot na medição.....	14
4.1 DIFERENÇA DO BOOT .....	14
Experiência – Medida da Variação do Boot .....	15
Capítulo 5 – Medição do Drift-Rate .....	17
5.1 O QUE É DRIFT-RATE.....	17
5.2 CAUSA DO DRIFT-RATE.....	17
5.3 COMO MEDIR O DRIFT-RATE .....	17
Experiência – Medição do Drift-Rate .....	18
Capítulo 6 – Sincronização Manual e Automática Entre Dois Computadores.....	23
6.1 SINCRONIZAÇÃO MANUAL .....	23
6.2 SINCRONIZAÇÃO AUTOMÁTICA .....	23
Experiência - Diferença Entre Sincronização Manual e Automática .....	24
Capítulo 7 – Sincronização Via Internet.....	26
Experiência – Sincronização Via Internet.....	27
CONCLUSÃO .....	29
REFERÊNCIAS BIBLIOGRÁFICAS.....	30

# Capítulo 1 – Introdução à Medição do Tempo

## 1.1 SOL

Desde o invento do relógio mecânico no século XVII, o tempo tem sido calculado astronomicamente. Todo dia, o sol aparece no leste, alcança seu maior ponto no céu e desaparece no oeste. Esse evento pelo qual o sol alcança seu maior ponto no céu é chamado *transito solar* e ocorre normalmente ao meio dia. O intervalo entre duas transições do sol é chamado de *dia solar*. Então, como o dia tem 24 horas e cada hora tem 3600 segundos, o *segundo solar* é definido exatamente como 1/86400 do dia solar.

Em 1940, foi estabelecido que o período de rotação da Terra não é constante. A Terra, na verdade, está diminuindo sua velocidade de rotação devido à fricção do mar e atmosférica. Baseado em estudos em corais antigos, geólogos acreditam que a cerca 300 milhões de anos atrás o ano era composto de 400 dias. O tamanho do ano, ou seja, o tempo de uma volta da Terra ao redor do Sol, não mudou, apenas o dia é que tem se tornado mais longo. Essas revelações fizeram com que os astrônomos computassem o tamanho do dia medindo um variado número de dias e fazendo uma média antes de dividir por 86400. O resultado dessa conta foi chamado de *segundo solar médio (mean solar second)*. [1]

## 1.2 RELÓGIO ATÔMICO E TAI

Com a invenção do relógio atômico em 1948, tornou-se possível medir o tempo de forma muito mais precisa e independente das variações que a Terra possui em relação ao seu dia. O relógio atômico funciona a partir da contagem das transições do átomo de césio 133. Assim o trabalho de medição do tempo ficou a cargo dos físicos e não mais dos astrônomos. O segundo no relógio atômico foi definido como o tempo em que o césio 133 leva para ter exatamente 9.192.631.770 transições. A escolha desse valor foi dada para fazer o segundo no relógio atômico igual ao do segundo solar médio no dia de sua inauguração. Atualmente, cerca de 50 laboratórios ao redor do mundo possuem o relógio com o césio 133. Periodicamente, cada laboratório informa ao Bureau International de l'Heure (BIH), em Paris, quantas vezes seu relógio tem pulsado. O BIH com essas informações produziu o *International Atomic Time* (Tempo atômico internacional), cuja abreviatura é TAI. Assim, TAI é a média do número de pulsos do césio 133 desde a meia noite do dia 1 de Janeiro de 1958, dividido por 9.192.631.770. [4]

Apesar do TAI ser altamente estável e disponível a qualquer um que queira comprar um relógio atômico, existe um sério problema, 86400 segundos do TAI é hoje em dia cerca de 3 milissegundos a menos que o dia solar médio (por causa do dia estar ficando mais longo). Usando o TAI para manter o tempo significaria que com o

passar dos anos, meio dia seria cada vez mais cedo, até que certo dia ocorreria nas primeiras horas da manhã.[1]

### 1.3 UTC

O BIH dando-se conta disso solucionou o problema introduzindo *leap seconds* (*pulos nos segundos*). Esses pulos ocorreriam sempre que a discrepância entre o TAI e o tempo solar ficasse na ordem de 800 milisegundos. Essa correção fez margem a um novo sistema de tempo baseado nos constantes segundos do TAI mas que permanece em fase com o aparente movimento do sol. Esse sistema foi chamado de UTC, Universal Coordinated Time. UTC é a base de toda marcação atual do tempo, e substituiu a medição antiga (Greenwich Mean Time), que era a medida feita astronomicamente.

### 1.4 MEIOS DE OBTENÇÃO PRECISA DO TEMPO

Empresas de tecnologia, usam em seus aparelhos relógios quem funcionam a frequência de 50Hz ou 60Hz. Então, quando o BIH anuncia um pulo (*leap second*), as companhias aumentam as frequências de seus relógios para 51Hz ou 61Hz durante 50 ou 60 segundos, avançando todos seus relógios da sua área de distribuição. Como 1 segundo é um intervalo grande para um computador, o sistema operacional que precisa manter seu tempo de forma precisa necessita de softwares especiais para fazer os *leap seconds* assim que eles forem anunciados. O número total de *leap seconds* introduzidos no UTC até hoje foram cerca de 32.

Existem inúmeras maneiras de se obter essa precisão do UTC. Nos EUA o National Institute of Standard Time (NIST) opera uma estação de ondas curtas de radio chamada de WWV em Fort Collins, Colorado. WWV envia um pulso de cada segundo do UTC. A precisão do WWV é de  $\pm 1$  milissegundo, mas devido às mudanças atmosféricas o comprimento do sinal pode chegar alterado piorando sua precisão (ficando, na prática, com  $\pm 10$  milisegundos). Na Inglaterra, a estação MSF, fornece o mesmo serviço, como também outras estações em diversos outros países. Outra maneira de se obter o UTC é oferecida por satélites, que possuem uma precisão melhor do que a de ondas de rádio, cerca de  $\pm 0,5$  milisegundos, porém esse recurso de precisão requer altos custos. Uma maneira mais barata, porém bem menos precisa, seria enviar as medições por telefone (modem) ou também pela internet.

O acesso ao UTC por satélite se dá por:

- GPS “civil” com um erro de 1 ms;
  - GPS “militar” com um erro de 1  $\mu$ s;
- (O GPS “militar” é acessível ao público desde o ano de 2000).

Assim, como o valor do UTC leva tempo a propagar-se até seu usuário, a qualidade com que conseguimos saber o tempo depende da qualidade da fonte original e da estabilidade do tempo de propagação do valor lido.[1]

## Capítulo 2 – Medição do Tempo no Computador

### 2.1 CRISTAL DE QUARTZO

O Cristal de Quartzo é uma substância cristalina que tem a propriedade de transformar a energia mecânica em energia elétrica e vice-versa. Este fenômeno é conhecido como Efeito Piezo-elétrico. O cristal oscilador é constituído por uma delgada lâmina de cristal translúcido, colocado entre duas placas de metal condutor ligadas aos seus terminais. Se o cristal é comprimido, aparece entre as duas placas metálicas uma tensão elétrica. Reciprocamente, se uma tensão elétrica alternada for aplicada nos terminais do cristal, ele oscilará. A frequência natural de oscilação do cristal depende da espessura da lamina e do ângulo de como foi cortado. Quanto mais fino é o laminado, maior é a frequência natural de sua oscilação. [1]

A construção de cristais osciladores para as frequências muito altas, se torna muito difícil devido a sua espessura e na prática a sua frequência natural de oscilação (frequência fundamental) é limitada. Normalmente os cristais osciladores vêm encapsulados em invólucros metálicos de diversos formatos e vedados contra a umidade.

### 2.2 RELÓGIOS DIGITAIS

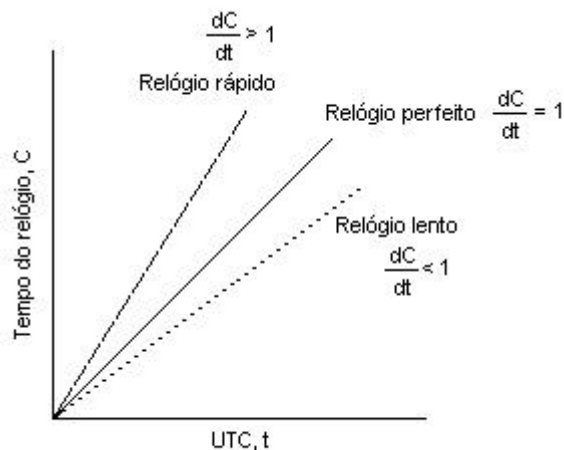
Os computadores de hoje possuem um circuito para manter a contagem do tempo. Tal circuito é baseado na frequência de oscilação dos cristais de quartzo. Associado a cada cristal, há dois registradores, um *contador* e um *holding register*. Cada oscilação do cristal decrementa um do contador. Quando o contador chega a zero, uma interrupção é gerada e o contador recebe o valor contido no *holding register*. Desta maneira, é possível programar o timer para gerar uma interrupção 60 vezes por segundo, ou a qualquer outra frequência desejada. Cada interrupção é chamada de *clock tick*.

Cada computador tem seu próprio relógio e seu próprio cristal. Embora a frequência na qual o cristal oscila seja fielmente estável, é impossível garantir que cristais em diferentes computadores oscilem exatamente na mesma frequência. Na prática, quando um sistema tem  $n$  computadores, todos os  $n$  cristais vão ter oscilações ligeiramente diferentes, fazendo com que alguns relógios andem mais rápidos ou mais devagar do que outros, sendo praticamente impossível encontrar relógios que andem exatamente no mesmo ritmo. [1]

### 2.3 RELÓGIO PERFEITO

Vamos considerar apenas um computador, que por sua vez possui um cristal. Podemos chamar o valor do relógio deste computador de  $C$ . Mais especificamente, quando a hora UTC é  $t$ , o valor do relógio do computador é  $C(t)$ . Num mundo

perfeito, teríamos  $C(t) = t$  para todo  $t$ . Ou em outras palavras,  $dC/dt$  idealmente deveria ser 1. Porém, já sabemos que é muito difícil atingir tal relação com exatidão. A figura abaixo nos mostra o relógio perfeito, o rápido e o lento.[1]



## 2.4 TIPOS DE SINCRONIZAÇÃO

Podemos dizer que existem basicamente dois tipos de sincronização de relógios em sistemas distribuídos, uma chamada de sincronização interna e a outra de sincronização externa.

Sincronização interna consiste em sincronizar todos os relógios de um determinado sistema entre si, sem considerar nenhuma outra fonte de relógio externa, como a hora do UTC por exemplo. Neste tipo de sincronização não importa se tempo dos relógios está correndo mais rápido ou mais devagar do que a hora real do UTC, o que importa neste caso é apenas que os relógios caminhem juntos.

Sincronização externa é justamente o tipo de sincronização que necessita de uma fonte externa, como a do UTC, para que os relógios sigam tal fonte. Por consequência da sincronização externa, haverá também uma sincronização interna entre os relógios que compõem o sistema.[2]

## 2.5 APLICAÇÕES

Em sistemas centralizados a fonte de tempo é a mesma para todos os processos. Se um evento A é executado antes do evento B, temos a certeza de que a hora em que A foi executado é menor do que a hora em que B foi executado. Já em sistemas distribuídos isso pode não ser uma certeza.

Vamos tomar como exemplo o comando *make* no UNIX. Normalmente, no UNIX, programas muito grandes são divididos em vários arquivos fontes. Uma mudança em um arquivo fonte necessita apenas que este arquivo seja recompilado,

evitando assim recompilar arquivos que não sofreram alteração e assim garantir uma maior agilidade no trabalho dos programadores. A maneira como o *make* funciona é bastante simples. Após o programador ter feito todas as alterações nos arquivos fonte, ele executa o *make*, que examina os tempos em que todos os arquivos fonte e arquivos objeto sofreram a última alteração. Se o arquivo fonte *input.c* possui o tempo de alteração maior do que o correspondente arquivo objeto *input.o*, então o *make* sabe que o *input.c* precisa ser recompilado.

Agora vamos imaginar o que poderia acontecer em sistemas distribuídos em que não há uma sincronização devida entre os relógios dos computadores. Suponha que *output.o* possua o tempo 2144, e que instantes depois *output.c* é modificado mas é associado ao tempo 2143 porque o relógio do computador onde ele está sendo editado é ligeiramente mais devagar. O *make* não chamará o compilador. O resultado disto é um programa com alguns arquivos novos e outros velhos e provavelmente não funcionará adequadamente levando o programador a loucura tentando entender o que está errado no código. Este exemplo do *make* necessitaria apenas de uma sincronização interna, visto que o importante neste caso é o tempo que está marcando nos relógios exclusivamente dentro do sistema, não importando o tempo real fora do sistema.

Sistemas em Tempo Real também necessitam de sincronização, visto que os relógios destes computadores necessariamente devem estar em sincronia com a hora real do mundo, UTC. Neste caso, temos uma sincronização externa e que pode ser feita com a ajuda de um receptor GPS (Global Positioning System), que através de uma antena, comunica-se com vários satélites que estão ao redor do planeta e consegue então calcular a hora exata (UTC). Ligando o GPS no computador, podemos então sincronizar a hora do PC com a hora real com bastante precisão.[1]

## Capítulo 3 – Medição do Erro Entre os Relógios de Dois Computadores

### 3.1 MEDIÇÃO DO ERRO

Para medirmos o erro entre os relógios de dois computadores diferentes, precisamos ler a hora dos dois PCs ao mesmo tempo. Para isto, devemos gerar uma interrupção de hardware simultaneamente nos dois PCs. Como sabemos, uma das formas de se gerar interrupções é através do teclado, toda vez que pressionamos uma tecla, é gerada uma interrupção para avisar ao sistema operacional que um caractere está sendo enviado. Portanto, podemos fazer um sistema que ao digitar-se uma tecla no teclado, seja enviado o mesmo caractere aos dois terminais simultaneamente. Agora basta fazer o sistema operacional informar na tela a hora de seu PC, no instante seguinte ao atendimento à interrupção. Após, basta calcular a diferença (em segundos) entre as horas que estão sendo marcadas nos computadores.

Para melhor entendimento de como é feito todo o processo, desde a digitação no teclado até a leitura da hora na tela, precisamos estudar um pouco sobre comunicação serial.

### 3.2 COMUNICAÇÃO ENTRE COMPUTADORES

O primeiro passo para medir a hora dos relógios é fazer existir comunicação entre os dois computadores que iremos utilizar. Há diversas maneiras de se fazer tal comunicação. Nestas experiências utilizaremos a comunicação via serial por ser a mais conveniente.



*figura 3.1*

Na comunicação serial, 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. Nesse modo, o controle é comparativamente muito mais simples que no modo paralelo e é de implementação mais barata. Como todos os bits são transferidos pelo mesmo meio físico (mesmo par de fios), as eventuais irregularidades afetam todos os bits igualmente.



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 para nós. 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*

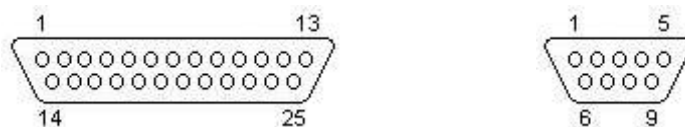
### 3.3 PORTA SERIAL

É uma das diversas portas existentes no computador, a qual possibilita a comunicação entre o próprio computador e outros dispositivos compatíveis. 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.

### 3.4 CABO SERIAL CRUZADO COM LOOP-BACK

Como citado acima, este tipo de serial pode ser tanto de 9 pinos quanto de 25, sendo que 9 pinos são indispensáveis.[3] A figura abaixo mostra os conectores:



visão a partir do cabo  
*figura 3.2*

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	utilizado pelo <i>Emissor</i> para pedir permissão ao <i>Receptor</i> para enviar os dados
CTS	8	5	utilizado pelo <i>Receptor</i> para comunicar ao <i>Emissor</i> que está apto a receber os dados
DSR	6	6	utilizado pelo <i>Receptor</i> para comunicar ao <i>Emissor</i> que está pronto para estabelecer comunicação
GND	5	7	este pino serve para indicar o sinal de referência tanto para o <i>Emissor</i> quanto para o <i>Receptor</i>
DCD	1	8	utilizado pelo modem para sinalizar que estabeleceu 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

Abaixo podemos ver a configuração de um cabo serial cruzado, que é utilizado para comunicação entre dois computadores, onde os dois PCs podem tanto enviar quanto receber dados.

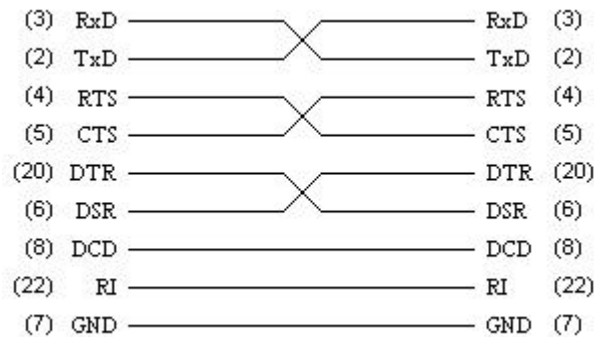


figura 3.3

No nosso caso, a comunicação é mais complexa. Queremos que um dos PCs envie o caractere digitado 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 PCs ao mesmo tempo, como desejamos. Temos abaixo um esquema ilustrativo da ligação que deve ser feita entre Tx e Rx dos dois PCs para termos um *loop-back*.

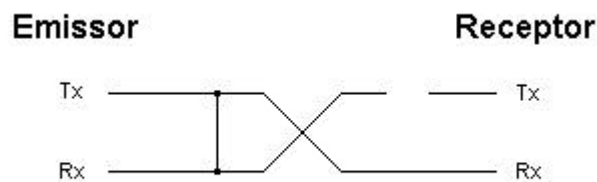


figura 3.4

Um ponto importante que devemos ressaltar, é o fato de termos que cortar o pino do Tx do *Receptor*. Isso se faz necessário porque se mantivermos esta ligação, o Rx do *Emissor* poderá receber dados ao mesmo tempo do *Receptor* e do próprio *Emissor*. Assim, certamente teria um conflito de informações em Rx, o que não é desejável que ocorra.

### 3.4.1 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 3.4). Sabemos 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 pela corrente elétrica entre Tx e Rx1 é menor do que a distância entre Tx e Rx2, e depende das medidas do cabo.

Vamos supor que a distância entre Tx e Rx1 seja igual a 15 cm, e a distância entre Tx e Rx2 seja igual a 180 cm. Agora, queremos ler a hora dos dois PCs. Então enviamos uma instrução ao mesmo tempo solicitando ao Sistema Operacional de cada PC que leia a sua hora. Porém, devido ao tempo de propagação no cabo, o *Emissor* vai ler a hora antes do *Receptor*, o que vai contra o que nós queremos.

Vejamos isso em números. A velocidade da luz é de 300.000 Km/s. Então, ela percorre uma distância de 15 cm em 0,5 ns e 180 cm em 6 ns.

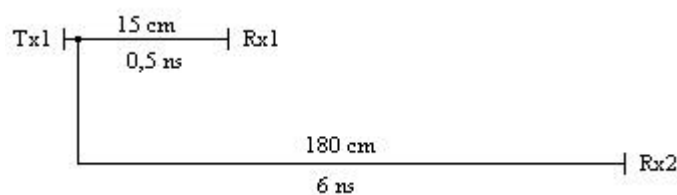


figura 3.5

A princípio isso pode parecer insignificante, até porque no nosso caso não pretendemos chegar a tal precisão. Porém, estes são dados que nos fazem ver que estamos chegando a um limite da física, e como sabemos, não podemos ultrapassar a velocidade da luz.

## Experiência – Introdução à Sincronização de Relógios

### OBJETIVOS

- Estudar funcionamento da porta serial do computador;
- Conhecer o uso do cabo cruzado serial com loop-back e a pinagem dos conectores;
- Medir a diferença entre os relógios de dois PCs.

### 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 com intervalo de 30 segundos entre cada medida;
5. Calcular a média da diferença entre os relógios e o desvio padrão.

**RESULTADOS OBTIDOS NA EXPERIÊNCIA**

<b>Computador A</b>	<b>Computador B</b>	<b>Diferença entre A e B</b>
16:05:05,31	16:05:10,36	5,05
16:05:36,54	16:05:41,60	5,06
16:06:06,80	16:06:11,84	5,04
16:06:35,57	16:06:40,62	5,05
16:07:06,10	16:07:11,20	5,10
16:07:36,23	16:07:41,29	5,06
16:08:05,75	16:08:10,80	5,05
16:08:35,15	16:08:40,20	5,05
16:09:05,32	16:09:10,37	5,05
16:09:36,30	16:09:41,36	5,06

A média e o desvio padrão da diferença entre os computadores A e B são:

Média = 5,06

Desvio Padrão = 0,02

## Capítulo 4 – Impacto do Boot na medição

### 4.1 DIFERENÇA DO BOOT

Primeiramente, é de se esperar que as medidas feitas no capítulo anterior sejam consistentes, isto é, mantenham sua diferença com uma pequena variação. Mas o que você diria se reiniciássemos a máquina e fizéssemos a mesma medição? A diferença continuaria pequena ou iria aumentar bruscamente ou até diminuir? Neste capítulo tentaremos verificar qual a variação ocorrida na diferença do tempo ao se dar o *boot* no computador.

Você sabe que no computador existe um cristal usado na medida do tempo do relógio, porém ao ligá-lo, o computador troca essa medição do tempo do cristal para o relógio de hardware, que também possui cristal. Após o *boot* o tempo passa a ser contado novamente pelo temporizador. Então essa troca de relógios feita no boot faz com que ocorram variações, muitas vezes até bruscas, no tempo. Com isso, não se pode afirmar com certeza para qual valor o relógio “pulará”, mas é certo que o *boot* invalida qualquer medida feita anteriormente.

Em nossas experiências, evidenciou-se que o boot acarretava um atraso de 1,4 segundos no relógio do computador A e 1 segundo no relógio do computador B, quando somente um deles era reinicializado. Porém esse desvio ocorria toda vez que sincronizava os relógios pelo recurso set time do DOS, não usando esse recurso, a variação era de 0,5 seg.

## Experiência – Medida da Variação do Boot

### OBJETIVOS

→ Verificar a variação ocorrida pelo boot e provar que a medida se torna inconsistente após tal boot.

### 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. Relizar o boot na maquina 1 e voltar para o modo DOS;
8. Repetir os passos 4,5 e 6;

## RESULTADOS OBTIDOS NA EXPERIÊNCIA

Ajustamos os relógios dos dois computadores para que eles tivessem aproximadamente 5 segundos de diferença entre eles e executamos o "Term". Medimos a seguinte defasagem:

Hora do Computador A – 15:30:10,21

Hora do Computador B – 15:30:15,31

Defasagem = 5,10 s

Reiniciamos o Computador A e medimos a defasagem novamente:

Hora do Computador A – 15:32:37,54

Hora do Computador B – 15:32:44,06

Defasagem = 6,52 s

Podemos ver que o simples fato de reiniciar o computador afetou demais a diferença entre os relógios, levando a defasagem de 5,10 segundos para 6,52 segundos.



## Capítulo 5 – Medição do Drift-Rate

### 5.1 O QUE É DRIFT-RATE

Drift-Rate nada mais é do que a taxa de afastamento dos relógios. Esta taxa pode ser tanto em relação ao UTC (hora correta) quanto a outro relógio qualquer.

Neste capítulo vamos fazer a medição do drift-rate entre os relógios de dois computadores independentemente da hora UTC. Neste tipo de medição devemos considerar a hora de um dos computadores como se fosse a hora de referência, e que a hora do outro computador fosse a hora que estivesse se afastando da hora correta.

### 5.2 CAUSA DO DRIFT-RATE

O que ocasiona a existência do drift-rate é o erro existente na frequência de oscilação dos cristais. Como já foi dito anteriormente no Capítulo 2, é praticamente impossível obter dois cristais exatamente iguais, o que causa uma certa diferença na frequência de oscilação dos mesmos. Tal erro, por menor que seja, se propaga no tempo linearmente, e à medida que o tempo passa o valor do erro aumenta.

### 5.3 COMO MEDIR O DRIFT-RATE

O processo de medição da taxa de afastamento dos relógios é dado da seguinte maneira: devemos medir as horas dos 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 podemos de forma alguma reiniciar um dos computadores após o início das medições, visto que o Boot afeta nossas medições. Caso um dos computadores seja reiniciado, deve-se começar todo o processo novamente.

## Experiência – Medição do Drift-Rate

### OBJETIVOS

→ Medir o afastamento dos relógios entre dois computadores

### 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.

## RESULTADOS OBTIDOS NA EXPERIÊNCIA

Realizamos a experiência acima e observamos os seguintes resultados:

	Computador A	Computador B	Diferença entre A e B
1	14:44:20,71	14:44:15,39	5,32
2	15:48:41,54	15:48:35,99	5,55
3	16:38:00,43	16:37:54,91	5,72
4	17:40:59,30	17:40:53,31	5,99
5	18:33:30,27	18:33:24,06	6,21

*Tabela 5.1*

Aqui, já podemos perceber claramente que a diferença entre os relógios foi aumentando a cada medida, chegando à aproximadamente 0,9 segundo em pouco menos de 4 horas, o que significa que os relógios estão se afastando à medida que o tempo passa.

Montamos agora uma tabela calculando a diferença da defasagem dos relógios entre os intervalos de tempo que foram feitas as medidas.

1-2	0,23	2-3	0,17	3-4	0,27	4-5	0,22
-----	------	-----	------	-----	------	-----	------

*Tabela 5.2*

Montamos agora outra tabela calculando (em segundos) todos os intervalos de tempo em que foram feitas as medidas.

1-2	3.861	2-3	2.959	3-4	3.779	4-5	3.151
-----	-------	-----	-------	-----	-------	-----	-------

*Tabela 5.3*

Ao dividirmos os dados da *Tabela 5.2* pelos dados da *Tabela 5.3*, considerando seus respectivos intervalos, obtemos a seguinte tabela:

$59,6 \times 10^{-6}$	$57,5 \times 10^{-6}$	$71,4 \times 10^{-6}$	$69,8 \times 10^{-6}$
-----------------------	-----------------------	-----------------------	-----------------------

*Tabela 5.4*

Calculando a média e o desvio padrão dos dados desta tabela, obtemos:

$$\text{Média} = 64,6 \times 10^{-6}$$

$$\text{Desvio Padrão} = 7,0$$

Podemos então dizer que os relógios dos Computadores A e B estão se afastando em média a uma taxa de  $64,6 \mu\text{s/s}$  ( $64,6$  micro-segundos por segundo).

Nesta taxa, ao longo de um dia, temos um afastamento de:

$$D1\text{dia} = (24 \text{ horas}) \times (3600 \text{ s/hora}) \times (64,6 \times 10^{-6} \text{ s/s})$$

$$D1\text{dia} = 5,6 \text{ segundos}$$

e ao longo de um mês:

$$D1\text{mes} = (30 \text{ dias}) \times (5,6 \text{ s/dia})$$

$$D1\text{mes} = 168 \text{ segundos ou quase 3 minutos}$$

Realizamos a mesma experiência novamente, porém agora vamos medir o drift-rate entre o mesmo Computador A e um terceiro computador que chamamos de Computador C. Os Resultados estão a seguir:

	Computador A	Computador C	Diferença entre A e C
1	14:42:20,59	14:42:10,54	10,05
2	15:47:09,15	15:46:58,44	10,71
3	16:36:19,80	16:36:08,60	11,20
4	17:38:27,82	17:38:15,95	11,87
5	18:40:58,24	18:40:45,77	12,47

Tabela 5.5

Podemos perceber que a diferença entre os relógios também está aumentando.

Montamos agora uma tabela calculando a diferença da defasagem dos relógios entre todos os intervalos de tempo que foram feitas as medidas.

1-2	0,66	2-3	0,49	3-4	0,67	4-5	0,60
-----	------	-----	------	-----	------	-----	------

Tabela 5.6

Montamos agora outra tabela calculando (em segundos) todos os intervalos de tempo em que foram feitas as medidas.

1-2	3.889	2-3	2.951	3-4	3.728	4-5	3.750
-----	-------	-----	-------	-----	-------	-----	-------

Tabela 5.7

Ao dividirmos os dados da Tabela 5.6 pelos dados da Tabela 5.7, considerando seus respectivos intervalos, obtemos a seguinte tabela:

$170 \times 10^{-6}$	$166 \times 10^{-6}$	$179 \times 10^{-6}$	$160 \times 10^{-6}$
----------------------	----------------------	----------------------	----------------------

Tabela 5.8

Calculando a média e o desvio padrão dos dados desta tabela, obtemos:

$$\text{Média} = 168,8 \times 10^{-6}$$

$$\text{Desvio Padrão} = 8,0$$

Podemos então dizer que os relógios dos Computadores A e C estão se afastando em média a uma taxa de  $168,8 \mu\text{s/s}$  ( $168,8$  micro-segundos por segundo).

Nesta taxa, ao longo de um dia, temos um afastamento de:

$$D_{\text{dia}} = (24 \text{ horas}) \times (3600 \text{ s/hora}) \times (168,8 \times 10^{-6} \text{ s/s})$$

$$D_{\text{dia}} = 14,6 \text{ segundos}$$

e ao longo de um mês:

$$D_{\text{mes}} = (30 \text{ dias}) \times (14,6 \text{ s/dia})$$

$$D_{\text{mes}} = 437 \text{ segundos ou pouco mais de 7 minutos}$$

Agora, fizemos a mesma experiência para os Computadores B e C. Obtivemos os seguintes resultados:

	Computador B	Computador C	Diferença entre B e C
1	14:46:31,82	14:46:27,10	4,72
2	15:49:44,92	15:49:39,76	5,16
3	16:39:18,64	16:39:13,15	5,49
4	17:41:56,04	17:41:50,16	5,88
5	18:37:36,39	18:37:30,13	6,26

Tabela 5.9

Também para este caso, podemos perceber que a diferença entre os relógios está aumentando.

Montamos agora uma tabela calculando a diferença da defasagem dos relógios entre todos os intervalos de tempo que foram feitas as medidas.

1-2	0,44	2-3	0,33	3-4	0,39	4-5	0,38
-----	------	-----	------	-----	------	-----	------

Tabela 5.10

Montamos agora outra tabela calculando (em segundos) todos os intervalos de tempo em que foram feitas as medidas.

1-2	3.793	2-3	2.974	3-4	3.757	4-5	3.340
-----	-------	-----	-------	-----	-------	-----	-------

Tabela 5.11

Ao dividirmos os dados da Tabela 5.10 pelos dados da Tabela 5.11, considerando seus respectivos intervalos, obtemos a seguinte tabela:

$116 \times 10^{-6}$	$111 \times 10^{-6}$	$104 \times 10^{-6}$	$114 \times 10^{-6}$
----------------------	----------------------	----------------------	----------------------

Tabela 5.12

Calculando a média e o desvio padrão dos dados desta tabela, obtemos:

$$\text{Média} = 111,2 \times 10^{-6}$$

$$\text{Desvio Padrão} = 5,2$$

Podemos então dizer que os relógios dos Computadores B e C estão se afastando em média a uma taxa de  $111,2 \mu\text{s/s}$  ( $111,2$  micro-segundos por segundo).

Nesta taxa, ao longo de um dia, temos um afastamento de:

$$D_{\text{dia}} = (24 \text{ horas}) \times (3600 \text{ s/hora}) \times (111,2 \times 10^{-6} \text{ s/s})$$

$$D_{\text{dia}} = 9,6 \text{ segundos}$$

e ao longo de um mês:

$$D_{\text{mes}} = (30 \text{ dias}) \times (9,6 \text{ s/dia})$$

$$D_{\text{mes}} = 288 \text{ segundos ou quase 5 minutos}$$

Você deve estar se perguntando: Para que fazer a mesma experiência 3 vezes?

Mas analisando os resultados das 3 experiências podemos demonstrar o seguinte:

Os PCs A e B estão se afastando a uma taxa de  $64,6 \mu\text{s/s}$ .

Os PCs B e C estão se afastando a uma taxa de  $111,2 \mu\text{s/s}$ .

Então, fazendo uma medida indireta, é de se esperar que o afastamento entre os PCs A e C seja de aproximadamente  $175,8 \mu\text{s/s}$ . E o valor calculado diretamente foi de  $168,8 \mu\text{s/s}$ , o que é um resultado bastante satisfatório.

## Capítulo 6 – Sincronização Manual e Automática Entre Dois Computadores

### 6.1 SINCRONIZAÇÃO MANUAL

Em um primeiro momento podemos pensar em sincronizar dois relógios diferentes ajustando a mesma hora para eles ao mesmo tempo. Você certamente já deve ter visto em filmes de ação, onde os personagens ajustam os seus relógios com uma mesma hora, e em seguida “dão a partida” neles ao mesmo tempo. Assim, eles saberão a hora exata que devem ser feitas as coisas.

A nossa proposta é mais ou menos a mesma, 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 nos dois PCs.

Obviamente esta não é a melhor forma de sincronizar relógios, até porque não se conseguiria uma precisão muito grande e muito menos uma regularidade nas sincronizações, porém é uma maneira bastante rápida e simples.

Para tentar melhorar um pouco este método, reduzindo a irregularidade proveniente do fato - “apertar a tecla *Enter* simultaneamente” - e aumentando um pouco a precisão da sincronização, podemos estabelecer um método automático que faça o mesmo que a sincronização manual, porém com mais eficiência.

### 6.2 SINCRONIZAÇÃO AUTOMÁTICA

Na Sincronização Automática queremos diminuir o efeito do erro causado pelo tempo de reação no momento de apertar a tecla *Enter*. Para isto, fizemos uma pequena modificação no programa *Term*, usado nas experiências anteriores.

O princípio é bastante simples e bem parecido com o *Term*, que já foi apresentado anteriormente. Ao invés de “darmos a partida” nos relógios usando dois teclados, vamos agora fazer o mesmo usando apenas um. Ao digitar um caractere no teclado de um dos PCs, este caractere será enviado aos dois PCs ao mesmo tempo via cabo serial. Assim que o computador recebe o caractere, o mesmo ajusta a hora do seu computador. Para isto, foi utilizado o comando *settime* no programa, que agora se chama *Sync*.

Mas qual a hora que o computador vai ajustar? Assim como usando o comando *Time* do DOS nós temos que dizer a hora que queremos ajustar, no *Sync* é exatamente igual. Temos que dizer qual a hora que queremos ajustar para os computadores. Isto é passado para o programa como um parâmetro na chamada do mesmo.

*Exemplo:* SYNC COM2 14 35

O relógio do computador quando receber o caractere da porta serial vai ser ajustado para 14:35:00,00.

## Experiência - Diferença Entre Sincronização Manual e Automática

### OJETIVOS

- ➔ Ver a diferença entre a sincronização manual e automática;
- ➔ Calcular o erro médio de cada tipo de sincronização.

### 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.



## RESULTADOS OBTIDOS NA EXPERIÊNCIA

Fazendo a sincronização manual (Time), obtivemos a seguinte tabela:

Computador A	Computador B	Defasagem
11:30:21,62	11:30:21,52	0,10
11:35:10,30	11:35:10,25	0,05
11:40:17,12	11:40:17,23	0,11
11:45:12,23	11:45:12,23	0,00
11:50:25,47	11:50:25,47	0,00

Média = 0,05

Desvio Padrão = 0,05

Fazendo a sincronização automática (Sync), obtivemos a seguinte tabela:

Computador A	Computador B	Defasagem
11:50:01,25	11:50:01,31	0,05
11:55:01,80	11:55:01,80	0,00
12:00:02,18	12:00:02,23	0,05
12:05:01,32	12:05:01,32	0,00
12:10:01,48	12:10:01,48	0,00

Média = 0,02

Desvio Padrão = 0,03

Podemos ver que a sincronização automática é muito mais regular e constante do que a sincronização manual. Apesar de ser possível conseguir sincronizar manualmente de forma perfeita (com defasagem zero) como foi feita na quarta e na quinta medida, é muito mais fácil conseguir tal resultado utilizando a sincronização automática.

## Capítulo 7 – Sincronização Via Internet

No capítulo anterior vimos duas maneiras de sincronizar relógios dentre tantas existentes. Tanto a sincronização manual quanto a sincronização automática mostradas naquele capítulo são consideradas sincronização interna, devido a falta de algum elemento externo como base para tal.

Neste capítulo vamos ver um outro tipo de sincronização, que é a sincronização externa. A sincronização externa, como já foi citada em capítulos anteriores, 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 de um GPS (Global Positioning System). Porém, não dispomos de tal tecnologia no momento. Uma maneira muito mais simples e um tanto quanto satisfatória para nossos estudos é sincronizar via Internet, onde também podemos obter a hora UTC de forma bem mais simples e suficientemente confiável para muitas aplicações.

Para sincronizar via Internet vamos utilizar um programa chamado *Atom Time* [5], que sincroniza automaticamente o relógio do seu 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 deste capítulo vamos sincronizar dois relógios de diferentes computadores via Internet, e depois com o auxílio do programa "Term", vamos medir a diferença que há entre os relógios dos dois computadores. Esperamos que o resultado seja uma sincronização entre os dois relógios também.

## Experiência – Sincronização Via Internet

### OBJETIVOS

- ➔ Realizar uma sincronização externa.
- ➔ Ver as diferenças entre sincronização interna e externa.

### 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.

Obs: 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*).

## RESULTADOS OBTIDOS NA EXPERIÊNCIA

Após sincronizarmos os relógios pela primeira vez com a UTC via internet, medimos a hora dos relógios com o auxílio do *Term* e obtivemos os seguintes resultados:

Hora do Computador A - 21:48:17,20  
Hora do Computador B - 21:48:17,37  
Diferença = 0,17 segundos

Para termos uma confiabilidade maior nas medidas, realizamos os mesmos passos anteriores mais três vezes, e obtivemos o seguinte:

Hora do Computador A - 21:58:03,42  
Hora do Computador B - 21:58:03,59  
Diferença = 0,17 segundos

Hora do Computador A - 22:08:06,78  
Hora do Computador B - 22:08:06,29  
Diferença = 0,59 segundos

Hora do Computador A - 16:41:19,20  
Hora do Computador B - 16:41:19,53  
Diferença = 0,33 segundos

**Média da Diferença = 0,32**  
**Desvio Padrão = 0,2**

Podemos perceber que a sincronização via internet, não é totalmente eficaz, pois apresenta um certo erro, porém é bastante satisfatória.

## CONCLUSÃO

Na realização deste projeto tivemos a oportunidade de compreender melhor sobre a sincronização de relógios de um modo geral e todos os assuntos a ele relacionados.

Nos capítulos iniciais vimos 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.

Vimos também os erros que podem surgir nas medidas e a maneira de medi-los, como tempo de propagação no cabo, o impacto do *Boot* na medição e o *drift-rate*.

Nos capítulos finais pode-se encontrar informações sobre a sincronização manual, automática e via Internet e seus respectivos erros na medição.

O assunto exige alguns conhecimentos mais específicos, porém é bastante simples e qualquer pessoa tem condições de ler o relatório e entender como funciona e qual a importância da sincronização de relógios.

Este relatório também foi feito com o intuito de ser aproveitado na disciplina de Informática Industrial II do Curso de Engenharia de Controle e Automação da UFSC e será utilizado de tal forma que os alunos possam entender melhor na prática as questões ligadas à sincronização de relógios em sistemas computacionais.

## REFERÊNCIAS BIBLIOGRÁFICAS

- [1] TANENBAUM, Andrew S. **Computer Networks**. Prentice Hall, Third Edition, EUA, 1996.
- [2] <http://www.inf.ufrgs.br/pos/SemanaAcademica/Semana98/agnolett.html>
- [3] <http://www.ctv.es/pckits/tpserie.html>
- [4] <http://www.dwelle.de/portuguese-brazilian/calendario/20001206/texto.html>
- [5] <http://www.atomtime.com>