



Relógios

FUNÇÕES PRINCIPAIS:

- mantém a hora do dia
- auxilia o escalonador de processos, impedindo que um processo monopolize a CPU

Observação:

SW do relógio pode tomar a forma de um driver, apesar de um relógio não ser nem um dispositivo de bloco e nem de caracter

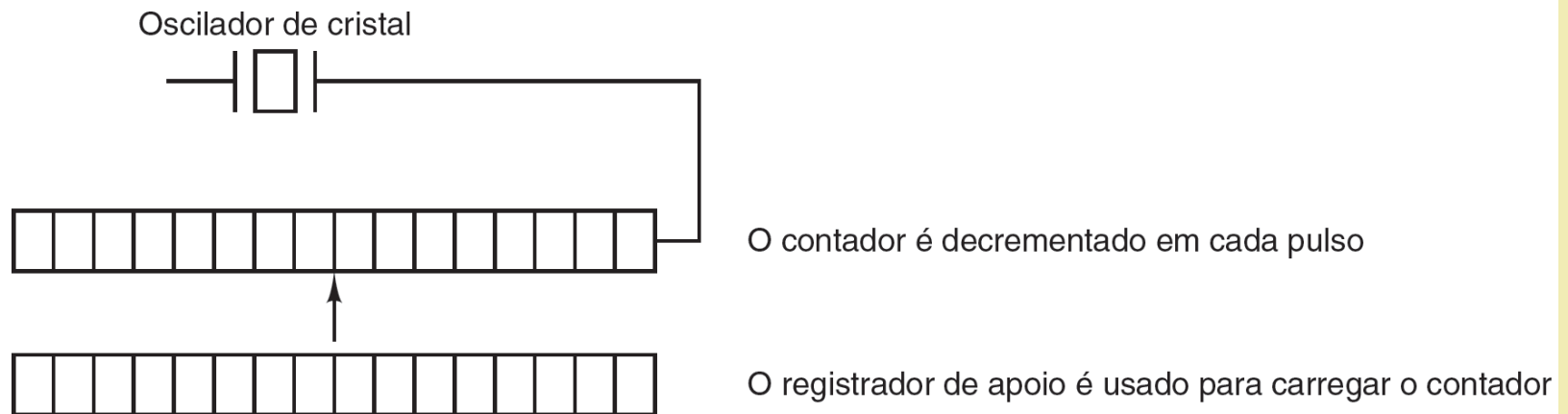
SISTEMAS OPERACIONAIS MODERNOS

3ª EDIÇÃO

Hardware de relógio

Antigamente: aproveitava-se a frequência da rede elétrica
(50 ou 60 Hz)

Atualmente:



■ **Figura 5.29** Um relógio programável.



Hardware de relógio

- ✓ Frequência do cristal pode ser multiplicada eletronicamente para gerar frequências ainda maiores;
⇒ fornece sinal de sincronização para os circuitos do computador

- ✓ Relógio programável pode funcionar em dois modos:
 - disparo único
 - onda quadrada



Hardware de relógio

Modo disparo único:

- ✓ na inicialização, copia valor do registrador de apoio para contador
- ✓ contador é decrementado a cada pulso do cristal
- ✓ quando chega a zero, gera uma interrupção na CPU e para até que seja reinicializado por SW



Hardware de relógio

Modo onda quadrada:

Após atingir o valor zero e causar interrupção, registrador de apoio é copiado automaticamente para o contador e processo se repete

As interrupções periódicas são denominadas *TIQUES DE RELÓGIO*



Hardware de relógio

Vantagem do relógio programável:

frequência pode ser controlada pelo SW

Exemplo:

Com um cristal de 500 MHz o Contador é pulsado a cada 2 ns.

Com um registrador de 32 bits pode-se programar as interrupções para serem feitas desde intervalos de 2 ns até intervalos de 8,6 seg.

- pode-se ter dois ou mais relógios independentes
- relógios podem ser usados para desabilitar interrupções



Software de relógio

Obrigações típicas de um *driver* de relógio:

1. Manter a hora do dia
2. Evitar que processos executem por um tempo maior que o permitido
3. Contabilizar o uso da CPU
4. Tratar a chamada de sistema *alarm* feita pelos processos dos usuários
5. Fornecer temporizadores para outras partes do sistema operacional
6. Gerar perfis, monitorar, coletar estatísticas.



Software de relógio

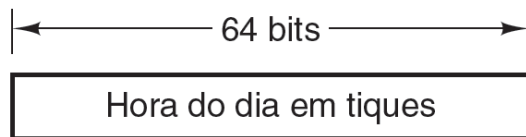
1. Manter a hora do dia

- requer o incremento do contador em cada tique do relógio;
Ex: frequência de 60 Hz, contador de 32 bits \Rightarrow aproximadamente 2 anos, no máximo!
- Sol. 1: aumentar o contador para 64 bits – dispendiosa
- Sol. 2: usar um contador auxiliar, que conta o n° de tiques até alcançar 1 seg. \Rightarrow só, então, altera o contador de seg. \Rightarrow 136 anos
- Sol. 3: carregar o tempo do relógio de segurança na memória quando da inicialização do sistema e iniciar contador.

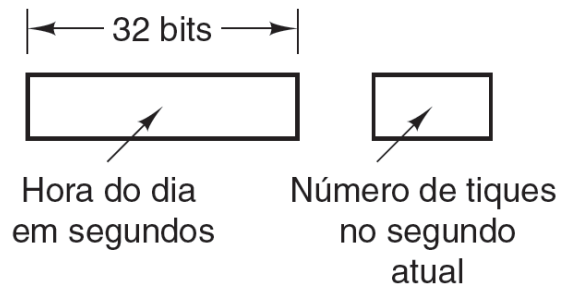
Hora atual = hora armazenada na memória + contador

SISTEMAS OPERACIONAIS MODERNOS

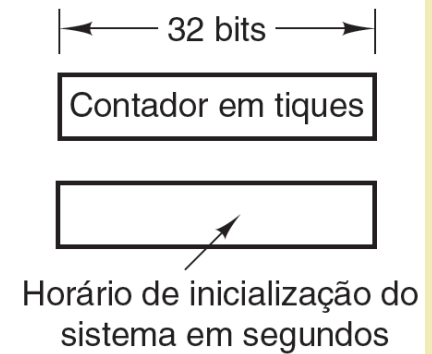
3ª EDIÇÃO



(a)



(b)



(c)

■ **Figura 5.30** Três maneiras para manter a hora do dia.



Software de relógio

2. Evitar que processos executem por um tempo maior que o permitido

- Quando processo é iniciado, escalonador inicializa o contador com o valor do quantum do processo (em tiques de relógio)
- A cada tique de relógio, contador é decrementado de 1 unidade
- Qdo contador chega em zero, *driver* chama o escalonador para selecionar outro processo



Software de relógio

3. Contabilizar o uso da CPU

1ª forma

- usar um segundo temporizador que é iniciado quando o processo é chamado e congelado (e lido) qdo o processo é interrompido
- quando ocorre uma interrupção, temporizador deve ser salvo e restaurado na continuação da execução do processo

2ª forma

- usar um ponteiro para a entrada da tabela de processos que indica qual processo está executando e a cada tique do relógio incrementa um campo naquela entrada



Software de relógio

4. Tratar a chamada de sistema *alarm* feita pelos processos dos usuários.

Algumas aplicações necessitam de avisos do SO

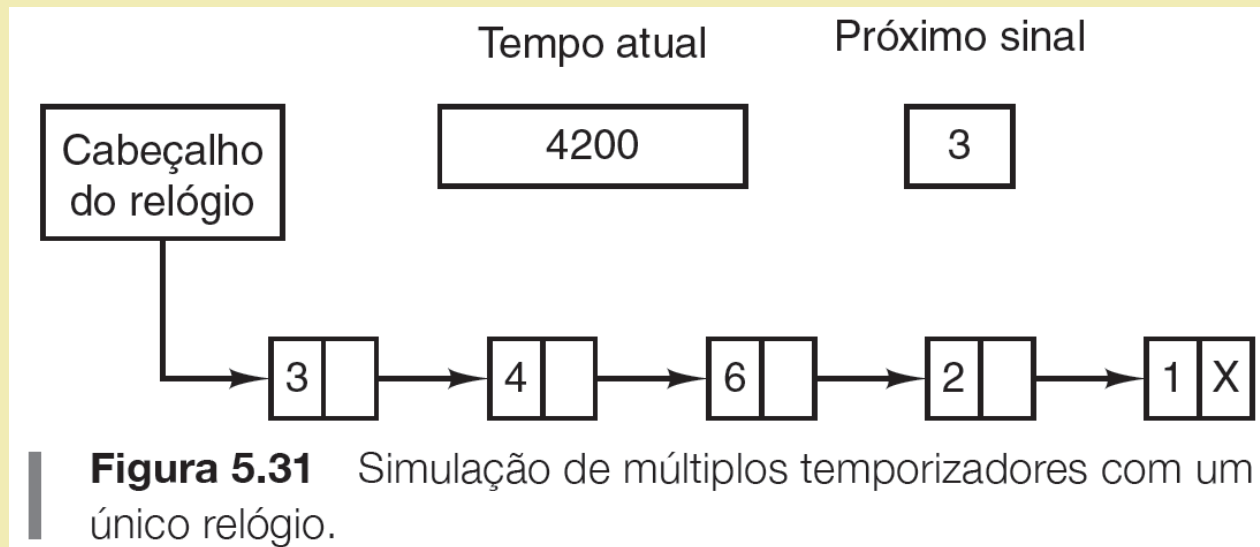
Ex: 1. um protocolo que trabalha com temporização do ACK
2. ensino por computador: resposta ao final de um tempo

- Se existe apenas 1 relógio físico, o *driver* deve simular vários relógios, ordenando todas as requisições no tempo e organizando-as em uma lista encadeada
- Cada nó da lista diz quantos tiques de relógio seguintes ao anterior devem ser esperados
- Cada item atendido é removido da lista

SISTEMAS OPERACIONAIS MODERNOS

3ª EDIÇÃO

- Tempo de atendimento dos sinais: 4203, 4207, 4213, 4215 e 4216
- Durante uma interrupção do relógio, driver incrementa o tempo atual, decrementa o temporizador e o compara com zero





Software de relógio

5. Fornecer temporizadores para outras partes do sistema operacional

Ex.: travamentos do sistema -- temporizador pode reiniciar um sistema que para de executar

- Enquanto o Sistema estiver executando, ele reinicia um temporizador, de forma que ele nunca expira.



Software de relógio

6. **Gerar perfis de execução, realizar monitoramentos e coletar estatísticas.**
 - Alguns SOs fornecem suporte ao usuário para que este saiba onde seu programa está gastando mais tempo
 - Mostra quantos tiques de relógio o programa ficou em cada instrução



Temporizadores por software

Motivação:

Existem duas formas de gerenciar E/S por HW: *interrupções e polling*

- Vantagem da interrupção: baixa latência (empilhar os valores atuais do PC e PSW e carregar os novos valores)
- Desvantagem: sobrecarga de chaveamento de contexto + TLB + cache (principalmente em máquinas modernas)



Temporizadores por software

Motivação:

- Vantagem do *pooling*: evita a sobrecarga das interrupções
- Desvantagem: latência pode ser alta se evento esperado ocorrer imediatamente após a última verificação
 - ⇒ atraso médio: metade do tempo de *polling* ⇒ algumas aplicações não toleram o atraso do *polling* e nem o atraso causado pelo chaveamento de contexto de uma interrupção

Ex: Ethernet Gigabit -- deve enviar ou receber um pct a cada 12 μ s (só de sobrecarga são 4,5 μ s)



Temporizadores por software

Ideia: Evitar os atrasos causados pelas interrupções

- Qdo a CPU estiver no modo núcleo por alguma razão, ela verifica o temporizador antes de voltar ao modo usuário
- Caso o temporizador tenha expirado, o evento escalonado é executado sem necessidade de chavear para o modo núcleo (já está neste modo)
- Temporizador é reiniciado após execução do trabalho



Temporizadores por software

Temporizadores por software são dependentes da frequência na qual as entradas no núcleo são feitas por causa de motivos, como:

1. Chamadas de sistema.
2. Faltas na TLB.
3. Faltas de páginas.
4. Interrupções de E/S.

Obs: Frequência média de entrada no núcleo varia entre 2 e 18 μ s



Temporizadores por software

Voltando ao exemplo da GigaBit Ethernet:

- é possível ter temporizadores operando a $12 \mu\text{s}$.
- o mesmo pode expirar, porém um atraso de $10 \mu\text{s}$ é melhor que interrupções consumindo 35% do tempo de CPU

Problema: talvez existam alguns períodos em que não existam chamadas ao sistema e a CPU não entre no modo núcleo

Solução colocar um limite superior para que a reinicialização do temporizador seja feita por HW, por exemplo, a cada 1 ms

Consequência: **em intervalos ocasionais a taxa de tx/rx de pacotes cairá a 1000 bits/seg**



Gerenciamento de energia

Economia de energia. Preocupação em 3 níveis:

Desktops – seria possível economizar milhares de megawatts

Notebooks – preocupação é com o *consumo da bateria* dos computadores, já que o progresso na duração das mesmas é mínimo

No nível de HW – tornar circuitos eletrônicos mais eficientes

- Duas estratégias:**
1. SO desligar partes do computador que não estejam em uso (principalmente, parte dos dispositivos de E/S)
 2. Aplicativo usar menos energia, degradando a qualidade da interface com o usuário para esticar o tempo de bateria (ex., celular)

SISTEMAS OPERACIONAIS MODERNOS

3ª EDIÇÃO

Gerenciamento de energia

Questões de Hardware:

Projetos permitem múltiplos estados do sistema: desligado, ligado, dormindo e hibernando

Dormir: desliga o monitor e faz uma cópia da memória para o HD, reduzindo o consumo de energia durante um curto período de tempo. Mais fácil voltar ao estado “ligado”

Hibernar: desliga o monitor, copia todos os aplicativos que estão sendo usados para o HD e desliga o computador.

- No retorno, o ambiente atual é recarregado na memória.
- Mais apropriado para um longo período de inatividade.
- Consumo de energia é menor que dormindo.
- Porém, gasta-se mais tempo e energia para voltar ao estado “ligado”



Gerenciamento de energia

Questões do Sistema Operacional:

SO controla todos os dispositivos \Rightarrow deve decidir quais dispositivos desligar e quando

- Compromisso entre atraso de reiniciação e o consumo
- Encontrar algoritmos para decidir o que e quando desligar

Problema: “boa decisão” é dependente da paciência de cada usuário

SISTEMAS OPERACIONAIS MODERNOS

3ª EDIÇÃO

Gerenciamento de energia

Consumo de cada dispositivo

| Dispositivo | Li et al. (1994) | Lorch e Smith (1998) |
|--------------|------------------|----------------------|
| Tela | 68% | 39% |
| CPU | 12% | 18% |
| Disco rígido | 20% | 12% |
| Modem | | 6% |
| Som | | 2% |
| Memória | 0,5% | 1% |
| Outros | | 22% |

Tabela 5.7 Consumo de energia de diferentes partes de um laptop.



Gerenciamento de energia

Monitor:

Item que mais consome energia no computador: normalmente SO desliga o monitor depois de alguns minutos de inatividade (programável) (estado *dormir*)

Proposta de economia de energia:

Dividir o monitor em zonas que possam ser desligadas independentemente umas das outras, dependendo da utilização do monitor. Fig. 5.39a e b

SISTEMAS OPERACIONAIS MODERNOS

3ª EDIÇÃO

Gerenciamento de energia

Monitor

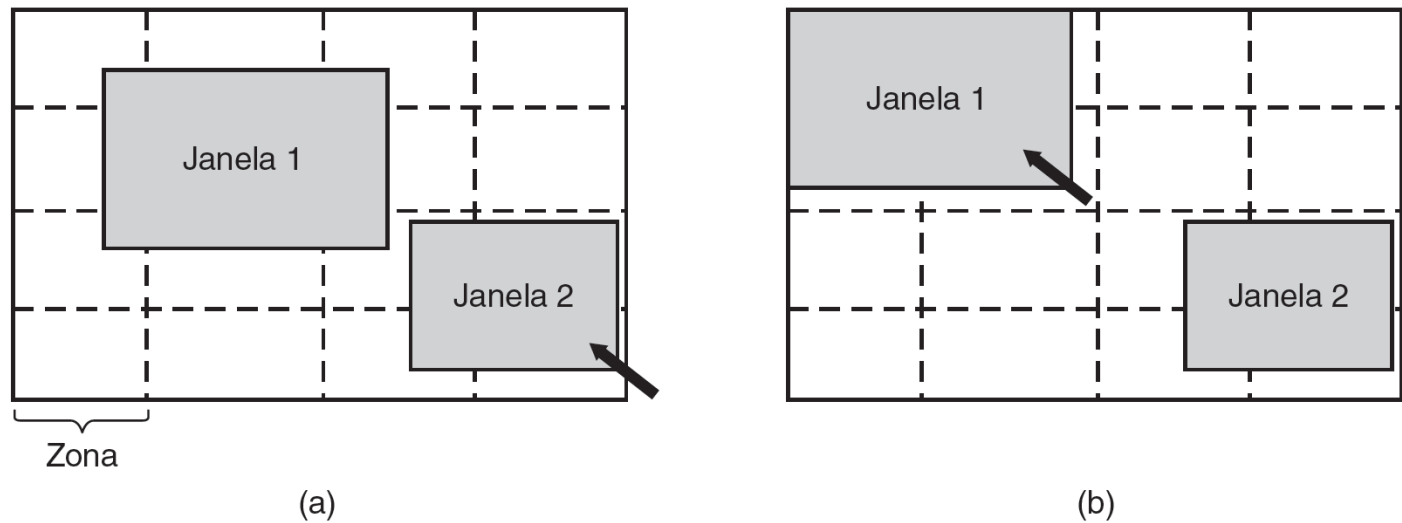


Figura 5.39 O uso de zonas para reanimar a iluminação do monitor. (a) Quando a janela 2 é selecionada, ela não é movida. (b) Quando a janela 1 é selecionada, ela é movida para reduzir o número de zonas iluminadas.



Gerenciamento de energia

Disco rígido:

Normalmente se mantém girando, mesmo sem acessos \Rightarrow consome energia

Laptops e Notebooks: ficam parados, hibernando

\Rightarrow atrasos para o usuário reutilizá-lo e consumo maior para reiniciá-lo

Três abordagens são possíveis:

1) Cada disco tem um tempo característico T_d

Se próximo acesso ao disco $t < T_d$

\Rightarrow preferível manter disco ligado

Se próximo acesso $> T_d \Rightarrow$ preferível desligar o disco e religá-lo depois

Mas, SOs não são inteligentes o suficiente para tratar automaticamente

SOs são conservadores: desligam o disco após alguns minutos



Gerenciamento de energia

Outras formas de economia:

2) Manter uma Cache grande de disco em RAM.

- Se leitura de bloco solicitado está na cache
⇒ disco não precisa ser religado até que ocorra uma lacuna de leitura
- Escritas podem ser colocadas na cache até a mesma ficar cheia
⇒ só então o disco é ligado

3) SO mantém os programas informados sobre o estado do disco através de sinais ou mensagens.

⇒ Programa atrasa a escrita até que o disco esteja ligado

Ex. processador de texto que escreve periodicamente no disco



Gerenciamento de energia

CPU

Propostas:

1. Consumo de energia é proporcional ao quadrado da voltagem.
Cortando metade da voltagem a CPU perde metade da rapidez
 \Rightarrow consumo cai a $\frac{1}{4}$
2. CPU dorme qdo não tiver trabalho. Acordá-la por uma interrupção.

Ex: programas de visualização multimídia que devem descomprimir e mostrar um quadro a cada 40 ms, por exemplo, deixando a CPU ociosa se descomprimir muito rapidamente (melhor diminuir velocidade da CPU)

SISTEMAS OPERACIONAIS MODERNOS

3ª EDIÇÃO

Gerenciamento de energia

CPU

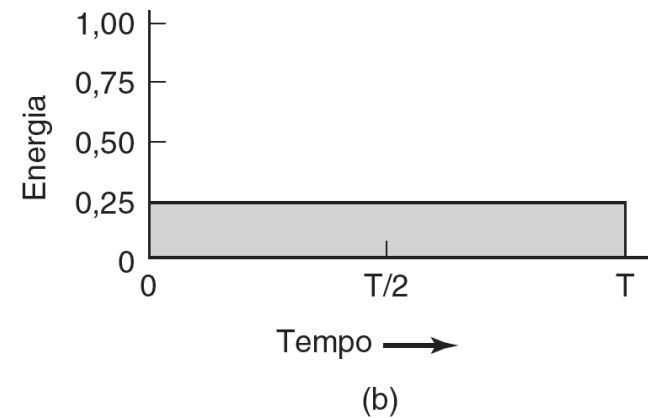
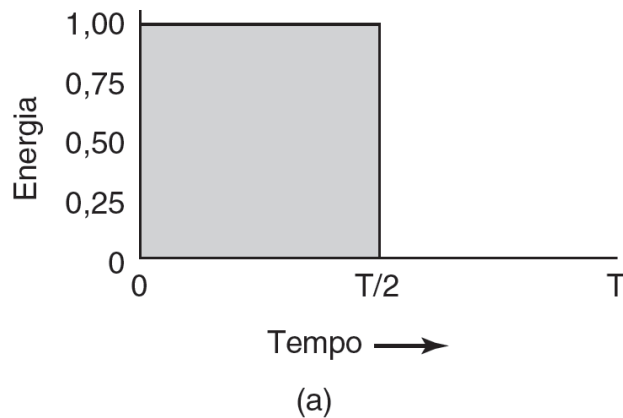


Figura 5.40 (a) Funcionamento com velocidade total. (b) Redução da voltagem à metade: metade da velocidade e um quarto da energia.



Gerenciamento de energia

Memória

Propostas:

1. **Esvaziar a cache e desligá-la:** recarga pode ser feita dinâmica e rapidamente a partir da memória
2. **Escrever conteúdo da memória principal no disco e desligá-la**

Obs: Proposta 2 - CPU provavelmente estará desligada tb. Sobrecarga de religamento, mas pode ser vantajoso dependendo do tempo que ficar desligada (por horas, por exemplo) e quisermos reiniciá-la em alguns segundos ao invés de recarregar o SO do disco



Gerenciamento de energia

Comunicação sem fio

Transmissores e receptores de rádio consomem muita energia

⇒ **bateria descarrega rapidamente**

Propostas:

- ❖ **Msgs recebidas:** computador móvel avisa à estação-base antes de desligar o rádio; estação armazena msgs que chegarem; qdo rádio é ligado novamente, estação é avisada e repassa as msgs que chegaram enquanto o rádio estava desligado
- ❖ **Msgs de saída geradas com rádio desligado:** computador móvel as armazena em um buffer. Quando o buffer enche, rádio é ligado para enviá-las

Ex: Wi-fi - rádio do computador móvel dorme entre os quadros de beacon enviados pela estação-base



Gerenciamento de bateria

Dispositivos móveis atuais usam baterias inteligentes

Baterias se comunicam com o Sistema Operacional e, sob solicitação do mesmo, fornecem informações como:

- **Voltagem máxima**
- **Voltagem atual**
- **Carga máxima**
- **Carga atual**
- **Taxa de descarga máxima**

SO avisa ao usuário quando a bateria está prestes a acabar para que o desligamento seja ordeiro, sem causar prejuízos ao Sistema de arquivos