

Gerência de Entrada/Saída: Parte 2

Prof. Gustavo Girão
girao@imd.ufrn.br

Roteiro

- Dispositivos de Entrada
- Dispositivos de Saída
- Relógios
- Gerenciamento de Disco
 - Escalonamento
 - Armazenamento
 - Erros

DISPOSITIVOS DE ENTRADA

Software de entrada

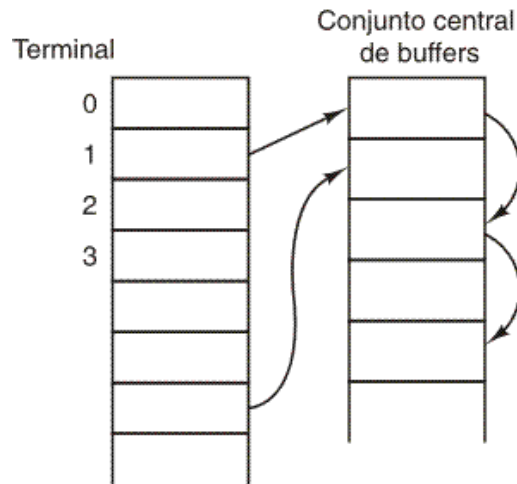
- Driver de teclado entrega um número
 - driver converte para caracteres
 - usa uma tabela ASCII
- Exceções, adaptações necessárias para outras linguagens
 - muitos SOs fornecem mapas de teclas ou páginas de códigos carregáveis

Software de Teclado

- Utiliza um código de varredura
 - Não é um código ASCII. Representa as TECLAS e não os caracteres
 - Determina também se a tecla foi pressionada ou liberada
 - **PRESS SHIFT, PRESS A, RELEASE A, RELEASE SHIFT**
 - ✧ Tradução: o usuário digitou A com o shift pressionado
- Modo Natural (raw)
 - Repassa a aplicação, tudo que o usuário digitou
 - Inclui backspaces que o usuário digitou para corrigir erros
 - Bastante detalhado
- Modo Canônico
 - Retorna a linha digitada pelo usuário
 - Um mecanismo de bufferização precisa ser utilizado

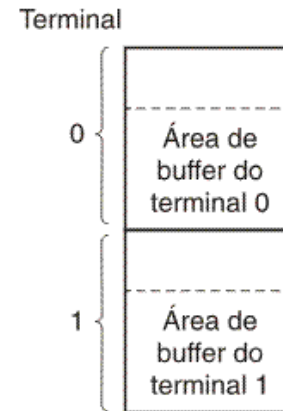
Driver de Teclado

Estrutura de dados do terminal



(a)

Estrutura de dados do terminal



(b)

Caractere	Nome POSIX	Comentário
CTRL-H	ERASE	Retrocede um caractere
CTRL-U	KILL	Apaga a linha toda que está sendo digitada
CTRL-V	LNEXT	Interpreta literalmente o próximo caractere
CTRL-S	STOP	Pára a saída
CTRL-Q	START	Inicia a saída
DEL	INTR	Interrompe o processo (SIGINT)
CTRL-\	QUIT	Força a gravação da imagem da memória (SIGQUIT)
CTRL-D	EOF	Finaliza o arquivo
CTRL-M	CR	Retorno do carro (inalterável)
CTRL-J	LF	Próxima linha (inalterável)

Software de entrada

- Teclados e monitores não são dispositivos completamente separados
- Processo **eco**
 - Envia todos os caracteres digitados pelo usuário ao monitor
- Pode ser atrapalhado por solicitações de escrita da aplicação
- Software do Mouse
 - Precisa caracterizar os botões pressionados e o posicionamento
 - ✧ Antigamente: trackball
 - ✧ Hoje: mouse ótico
 - Distância mínima de movimentação: em torno de 0,1mm
 - ✧ Chamada de **Mickey**

DISPOSITIVOS DE SAÍDA

Janelas de texto

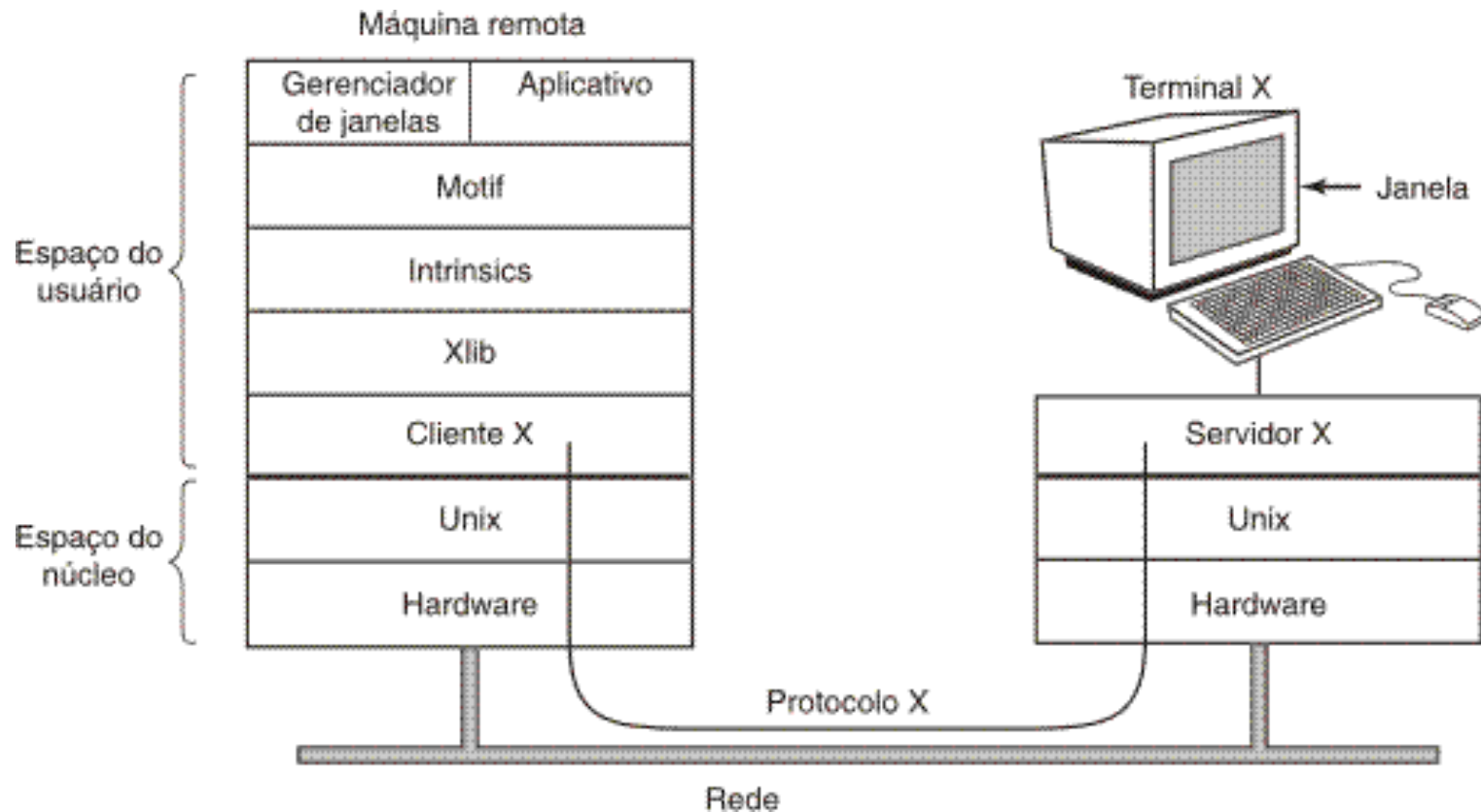
- Mais simples que teclado
- Precisa ser capaz de atualizar a telas em momentos importantes
- São necessários comandos de controle chamados **seqüências de escapes**
 - Apagar caracteres, reposicionar cursor, quebra de linha, etc
 - ESC[3 ; 1 H ESC[0 K ESC [1 M

Seqüência de escapes	Significado
ESC [<i>n</i> A	Move <i>n</i> linhas acima
ESC [<i>n</i> B	Move <i>n</i> linhas abaixo
ESC [<i>n</i> C	Move <i>n</i> espaços à direita
ESC [<i>n</i> D	Move <i>n</i> espaços à esquerda
ESC [<i>m</i> ; <i>n</i> H	Move o cursor para (<i>m</i> , <i>n</i>)
ESC [<i>s</i> J	Limpa a tela a partir do cursor (0 até o final, 1 desde o início, 2 tudo)
ESC [<i>s</i> K	Limpa a linha a partir do cursor (0 até o final, 1 desde o início, 2 tudo)
ESC [<i>n</i> L	Insere <i>n</i> linhas a partir da posição do cursor
ESC [<i>n</i> M	Remove <i>n</i> linhas a partir da posição do cursor
ESC [<i>n</i> P	Remove <i>n</i> caracteres a partir da posição
ESC [<i>n</i> @	Insere <i>n</i> caracteres a partir da posição do cursor
ESC [<i>n</i> <i>m</i>	Habilita substituição do tipo <i>n</i> (0=normal, 4=negrito, 5=piscante, 7=invertido)
ESC M	Rola a tela para trás se o cursor está na linha do topo

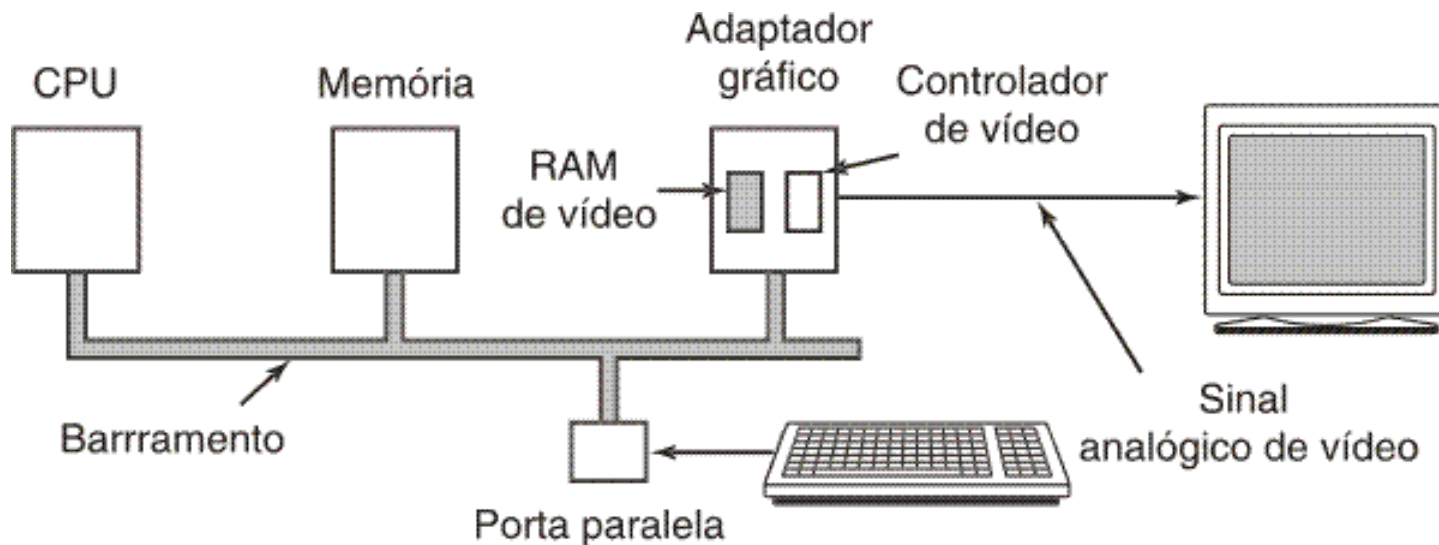
Sistema X-Window

- Servidor X
 - Controla a janela atualmente ativa
 - Precisa saber para qual aplicação enviar a entrada do teclado, por exemplo
- Cliente X
 - Envia entradas do teclado e mouse
 - Aceita comandos de exibição
- O servidor X fica nas máquinas remotas e atende aos envios de atualização de tela do cliente X
 - Pode parecer estranho, mas lembre-se de que este é um processo de saída. Por tanto quem determina o que deve ser exibido (em resposta às solicitações) é o cliente X (a aplicação que realiza as atividades)

Sistema X-Window



Hardware de vídeo

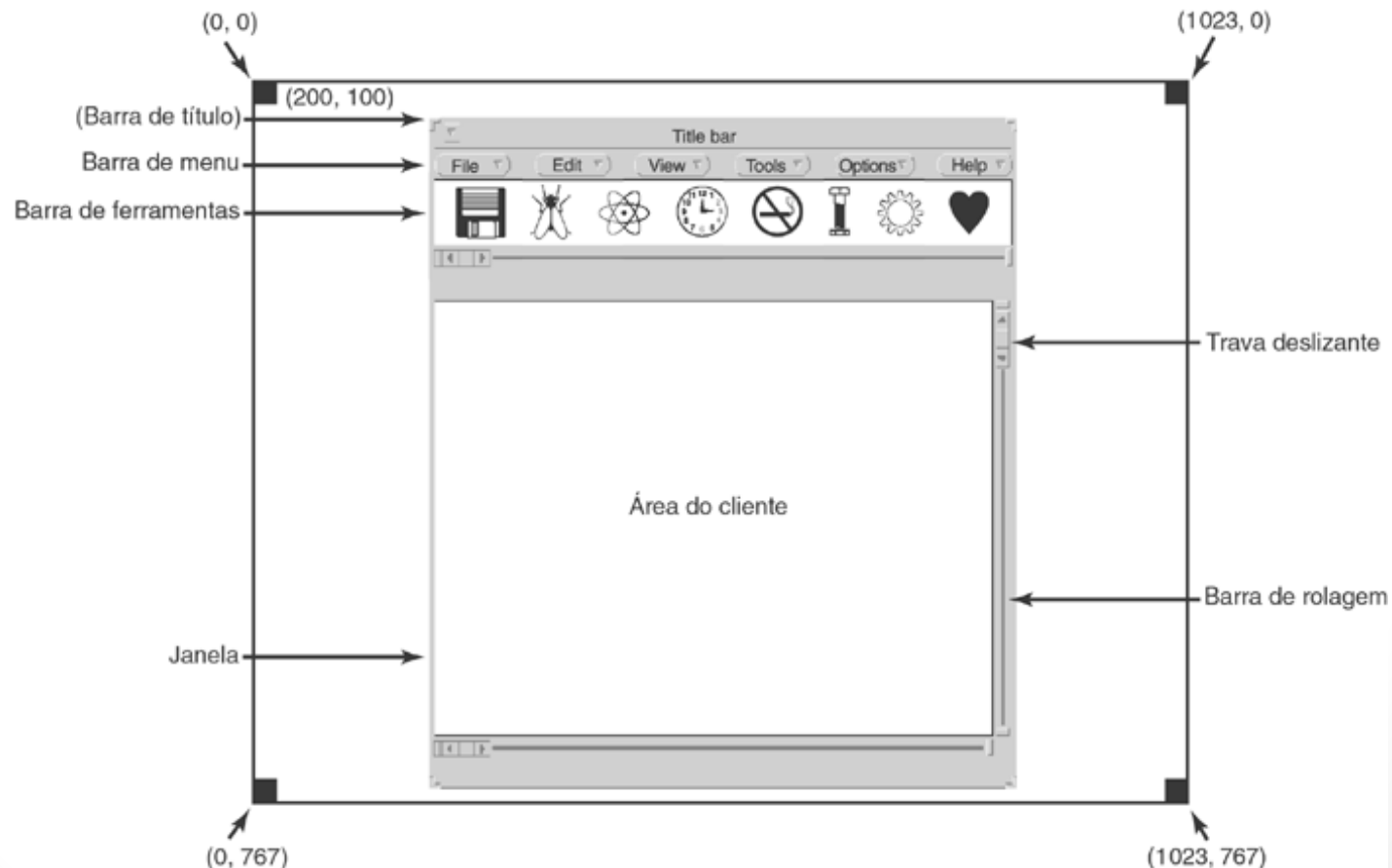


Interfaces Gráficas de Usuário

- GUI (Graphical User Interface)
 - Criada no Instituto de Pesquisa Stanford
 - Copiada por pesquisadores da XEROX...
 - ...E então copiada por Steve Jobs
 - Que se recusou a licenciar para Bill gates...
 - ...Que usou mesmo assim!
- Sistema WIMP (windows, icons, menus and pointing)
- A saída é direcionada para um adaptador gráfico que faz uso de uma memória RAM de vídeo que bufferiza elementos da GUI para serem exibidos na tela
 - A latência de acesso da memória de vídeo deve ser correspondente a taxa de refresh da tela

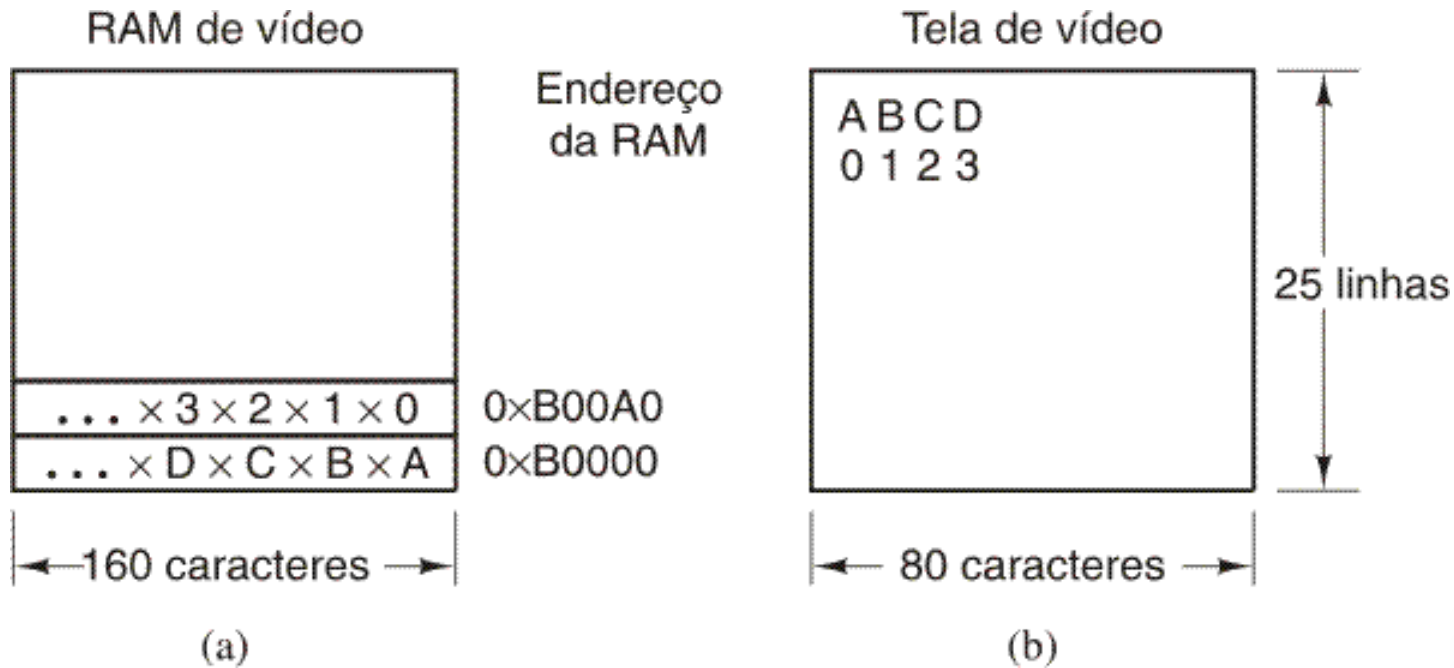
Software de saída

- Sistema WIMP (windows, icons, menus and pointing)



Software de saída

- Buffer de vídeo



RELÓGIOS

Drivers de Relógio

- Hardware
 - Dois tipos:
 - ✧ Ligado à rede elétrica (gera interrupção em ciclos de 1/60 de segundo)
 - ✧ Cristal (oscilador de cristal + contador + registrador com constante): contador é inicializado e decrementado a cada sinal periódico do cristal (de 1 a mais de 20Mhz), ao zerar controlador gera interrupção.
 - Relógios programáveis:
 - ✧ Dois modos: deve ser reiniciado a cada interrupção (one-shot mode) e reinicialização automática (square-wave mode)
 - ✧ Pode-se escolher periodicidade da interrupção variando-se constante armazenada no registrador
 - Hardware somente gera interrupções a intervalos predefinidos

Drivers de Relógio: Funções

- Manter hora correta
 - 64 bits
 - Tempo em segundos +tics
 - Hora do boot em segundos + tics (32 bits)
- Não deixar processo rodarem por mais que sua fatia de tempo
 - ✧ Quando processo começa a rodar, relógio é iniciado pelo valor do quantum, quando interrupção é gerada, escalonador é chamado
- Contabilizar uso de CPU
 - Campo na tabela de processos contabilizado por tics
 - Atualização feita a cada interrupção (não a cada tic- muito caro)

Drivers de Relógio: Funções

- Implementar ALARMES (chamada ao sistema por processo de usuário)
 - Alarme geralmente é um sinal
 - Pode-se usar lista ligada com temas dos alarmes, com diferença para próximo alarme e contador de tics até próximo alarme; quando hora do dia atualizada, driver verifica se algum timer venceu
- Fornecer alarmes especiais para o sistema
 - Watchdog timers; semelhante aos sinais para usuário, mas driver chama procedimento estabelecido quando se programou alarme (e.g. Stop_motor no floppy Minx)
- Coleta de estatísticas em geral
 - Profiling: a cada tic, driver verifica se processo está sendo “profiled” e atualiza contador de tics do “bin” correspondente ao PC

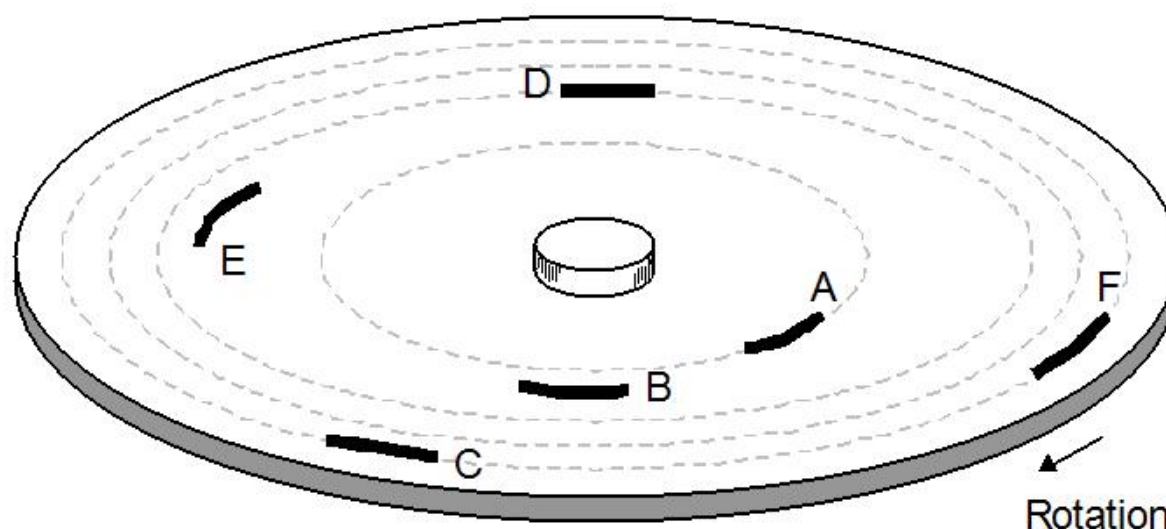
GERENCIAMIENTO DE DISCO

Taxa de transferência efetiva

- Medidas de desempenho
 - Tempo de seek – tempo de mover os cabeçotes
 - ✧ Aproximação: $(\# \text{ de trilhas} \times C) + \text{tempo de inicialização}$
 - Atraso de rotação – tempo necessário para rotar o setor correto sob o cabeçote de leitura/gravação
 - ✧ Média: $\frac{1}{2}$ rotação
 - HD: 5400rpm \rightarrow 5.6ms, 10000rpm \rightarrow 3ms
 - Floppy: 300 rpm \rightarrow 100ms
- Tempos efetivos
 - Tempo de acesso – soma do tempo de seek e atraso de rotação
 - Tempo de transferência – tempo necessário para realizar leitura ou escrita (considerando que o cabeçote já está em cima do setor desejado)
- Tempo depende da localidade

Escalonamento de disco

- Quando é feita uma solicitação de leitura/escrita, o disco pode estar ocupado
- Todas as solicitações são colocadas em uma fila de espera
 - Estas solicitações podem ser escalonadas para otimizar a utilização
- Escalonamento de disco aumenta a largura de banda do disco
 - A quantidade de informação que pode ser transferida dentro uma janela de tempo



Escalonamento de Disco

- O sistema operacional é responsável por utilizar o hardware de maneira eficiente – para discos, isso significa ter um rápido tempo de acesso alta largura de banda
- Tempo de acesso tem dois grandes componentes
 - Tempo de seek é o tempo para que o disco mova seus cabeçotes para o cilindro contendo o setor desejado
 - Atraso de rotação é o tempo adicional de espera para que o disco rotacione até o setor desejado
- Minimizar tempo de busca
- Tempo de seek \approx distancia até o setor
- Largura de banda é o número total de bytes transferidos divididos pelo total de tempo entre a **primeira** requisição de serviço e a completude da **última** transferência

Escalonamento de disco

- Existem vários algoritmos para escalonar o serviço de requisição de I/O ao disco
- Utilizando um exemplo de uma fila de espera de 0 a 199
- Considere que os seguintes setores foram solicitados

98, 183, 37, 122, 14, 124, 65, 67

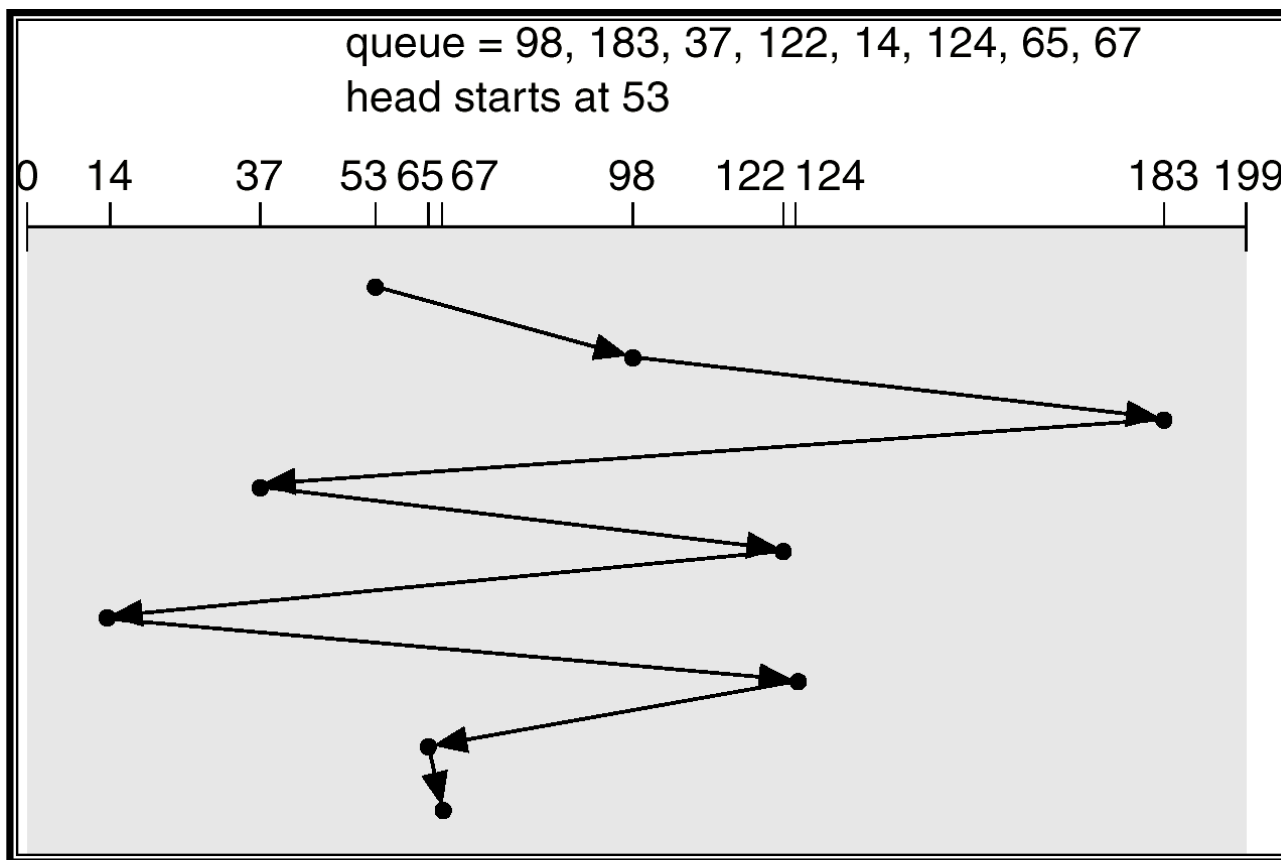
Considere ainda que, atualmente, o cabeçote se encontra no setor **53**

FCFS

- O algoritmo mais intuitivo a ser utilizado é o First Come First Served
- Algoritmo mais simples
- Atende às demandas na ordem cronológica em que foram solicitadas
 - Promove um senso de justiça
- Entretanto....
 - ...provavelmente não será o mais eficiente
 - Como determinamos eficiência??
 - ✧ Tempo de seek: por quantos cilindros o cabeçote terá que percorrer para atender a todas as solicitações da fila de espera

FCFS

Por quantos cilindros o cabeçote irá passar para atender a todas as requisições?

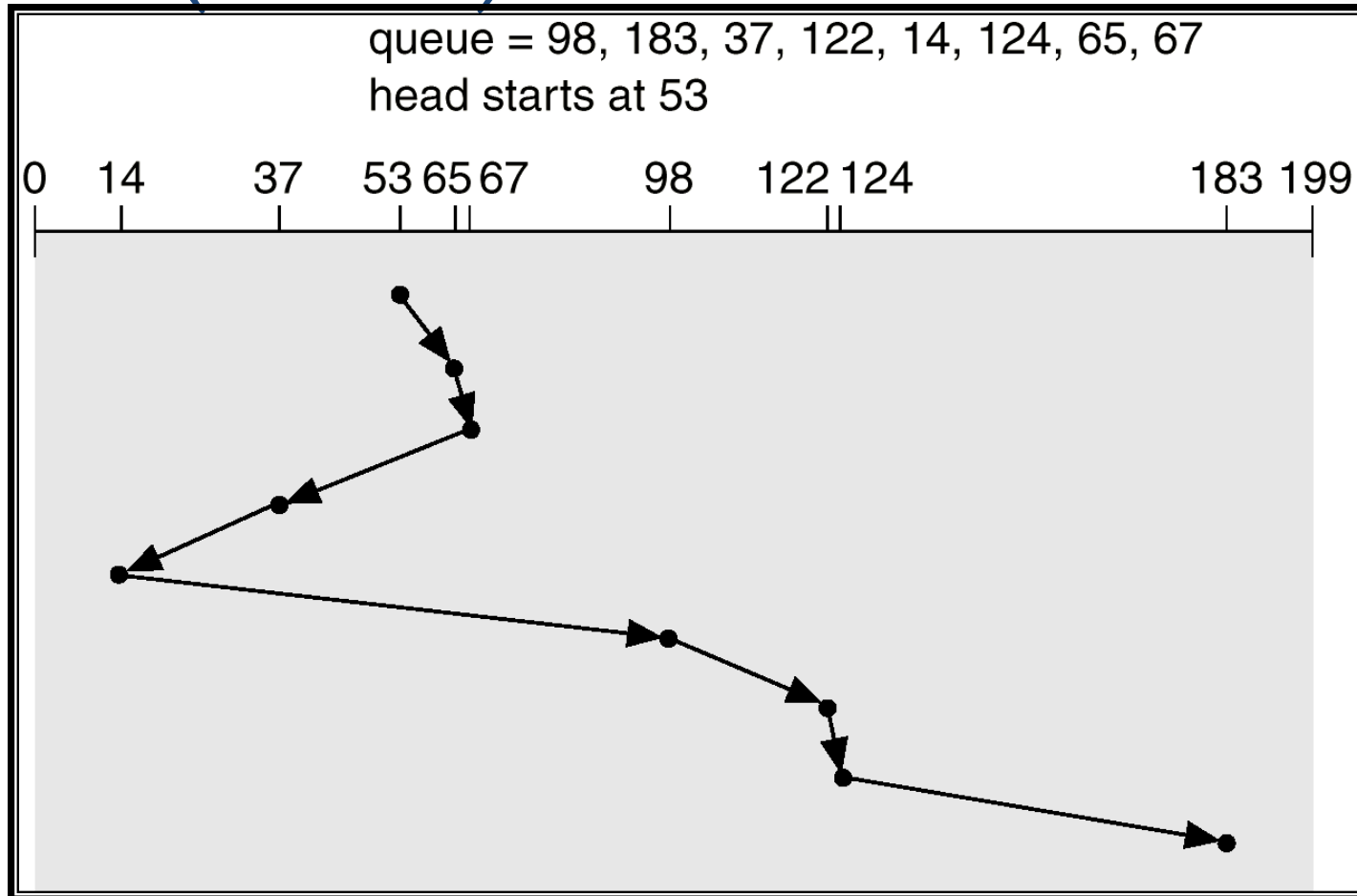


Resposta: 640 cilindros!

SSTF

- Agora que temos uma métrica de eficiência, podemos desenvolver algoritmos melhores (ao custo de uma implementação mais elaborada)
- Shortest-Seek-Time-First
- Seleciona a requisição com o menor tempo de seek da atual posição do cabeçote
- O escalonamento SSTF é uma forma de SJF (Shortest Job First)
 - Qual o principal problema?
 - Pode causar starvation?
 - ✧ Lembre-se de que a fila de espera é mutável
- Qual o movimento total do cabeçote considerando o SSTF no exemplo anterior?

SSTF (Cont.)



Resposta: 236 cilindros!

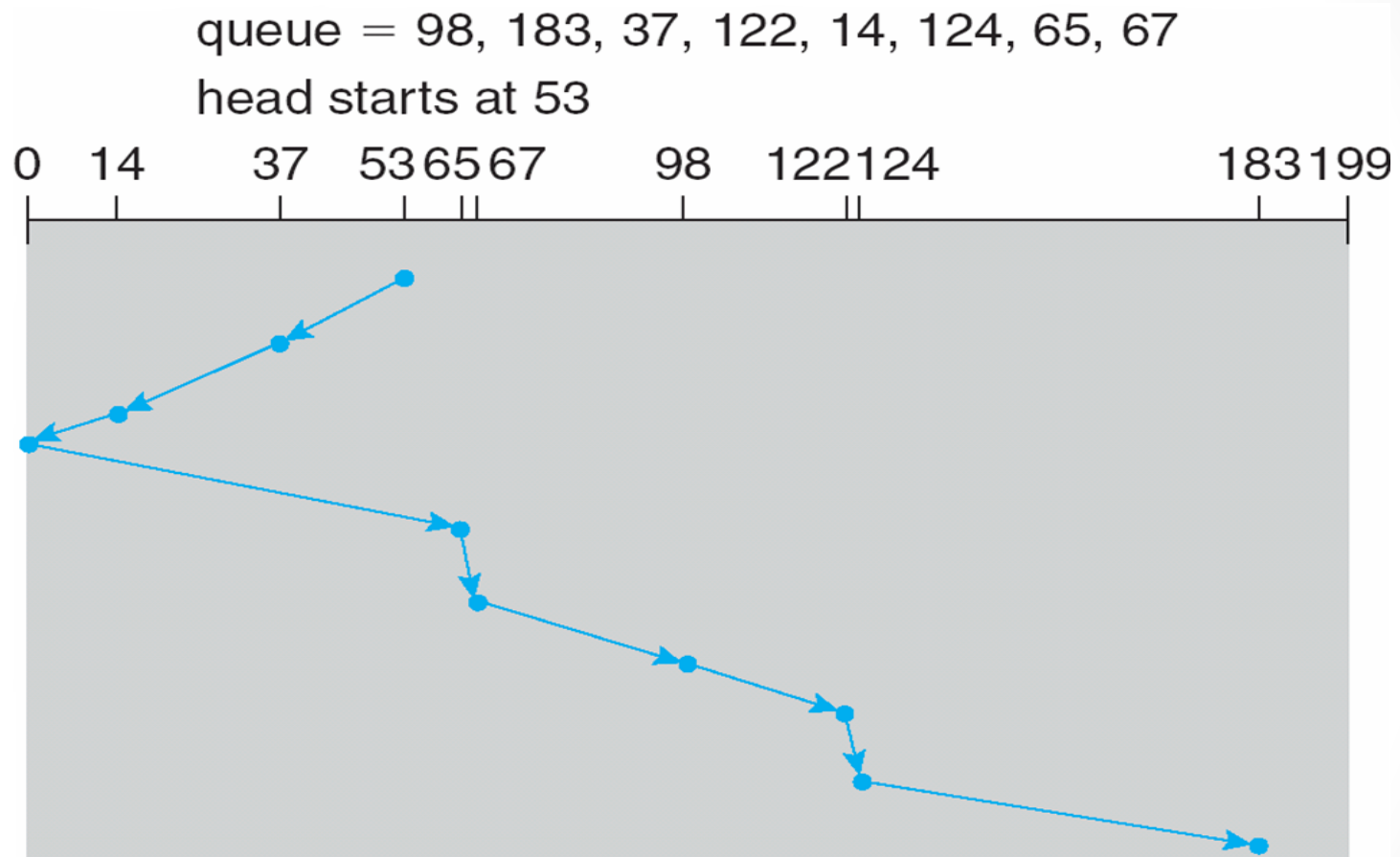
Esta é a sequência de acessos com o menor número de movimentos?

Resposta: Não!

SCAN

- O braço do disco inicia em uma extremidade e se movimenta na direção da outra extremidade atendendo requisições até que chegue ao outro lado.
 - Depois o cabeçote se movimenta no sentido oposto voltando até a outra extremidade. Como se estivesse escaneando todo o disco em busca de requisições
- Também conhecido como algoritmo do elevador
 - Mantém um bit de sentido indicando se sobe (indo para o fim do disco) ou desce (indo para o início do disco)
- O tempo de atendimento a uma requisição, mais do que nunca depende de sorte:
 - Se uma requisição chegar e o cabeçote estiver no setor logo antes
 - ✧ E se o cabeçote estiver logo depois? Azar.....

SCAN (de um lado a outro)



Se começar indo para o início do disco (setor 0)?

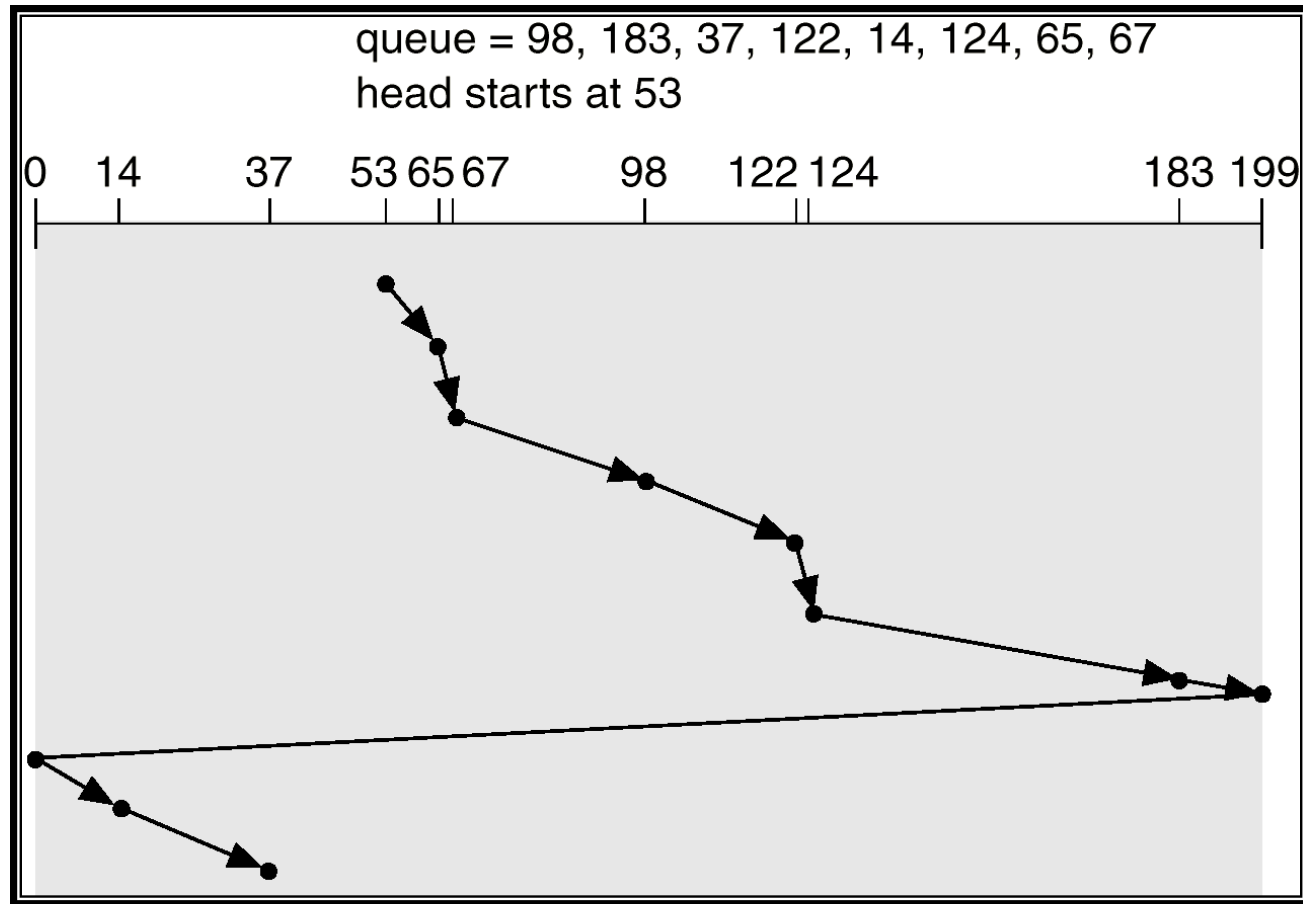
Resposta: 208 cilindros!

Se começar indo para o final do disco (setor 199)?

C-SCAN

- Provê um tempo de espera mais **uniforme** do que o SCAN
- O cabeçote se movimenta de uma extremidade do disco para a outra atendendo requisições conforme passa.
- Quando chega ao final, entretanto, ele imediatamente retorna ao **início** do disco **sem atender** a nenhuma requisição na viagem de volta
- Trata os cilindros como uma **lista circular** (i.e. o próximo cilindro após o último é o primeiro)
- Considerando uma distribuição uniforme nas solicitações, **poucas requisições vão estar logo adiante no andamento do cabeçote**. Muitas delas estarão do outro lado do disco e levarão **muito tempo** para serem atendidas...

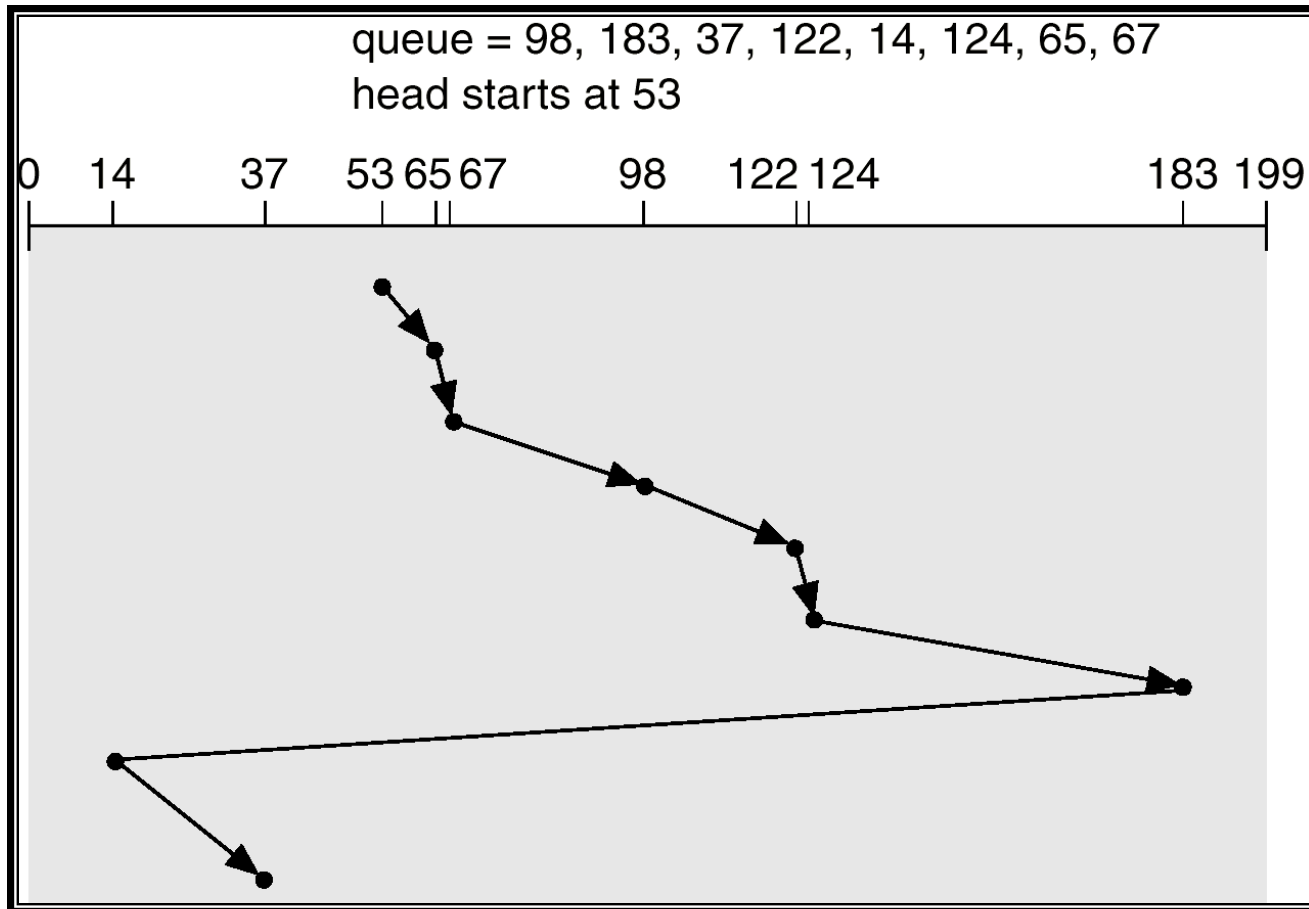
C-SCAN (Cont.)



C-LOOK

- Apenas uma implementação factível do C-SCAN
- O braço que contém o cabeçote de leitura não vai até o final (ou início) do disco.
 - Assim que chega na solicitação mais extrema da fila de espera, volta até o outro lado
 - Também não volta necessariamente até o zero e sim até a requisição mais próxima do zero (que pode ser o próprio zero)

C-LOOK (Cont.)



Selecionando um algoritmo de escalonamento de disco

- SSTF é comum e tem um apelo natural
- SCAN e C-SCAN tem um desempenho melhor para sistemas que colcoam um carga muito pesada no disco (**muitos processos I/O bound que fazem acesso ao disco**)
- Performance depende do **número** e **tipos** de requisições
- Requisições de serviços do disco podem ser influenciadas pelo método de alocação de arquivos (**vistos em aulas anteriores**)
- O algoritmo de escalonamento de disco deve ser escrito como um **módulo separado do sistema operacional** permitindo que seja substituído com um algoritmo diferente caso seja necessário
- Tanto **SSTF** ou **LOOK** são escolher razoáveis para um algoritmo de escalonamento padrão

Algoritmos de escalonamento de disco

Seleção de acordo com o item selecionado		
SSTF	Menor tempo de atendimento primeiro	Alta utilização, pequenas filas
SCAN	Indo e voltando no disco	Melhor distribuição do serviço
C-SCAN	Apenas um sentido com rápida volta ao início	Menor variabilidade dos tempos de acesso

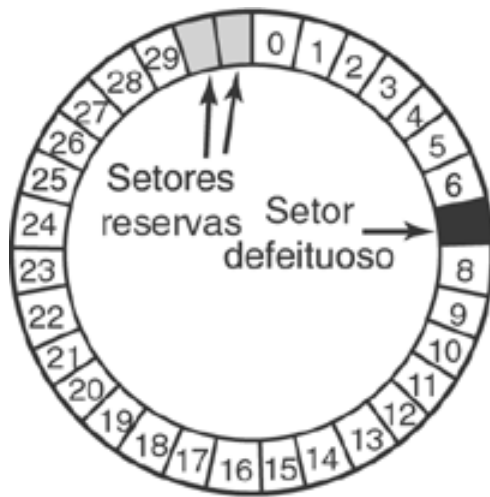
- É importante notar que outros algoritmos podem ser implementados ou podem modificar a fila de esperar considerando características de mais alto nível:
 - Quem é o dono do processo
 - Qual a prioridade do processo
 - Qual o tipo do processo (I/O bound ou CPU bound)

Tratamento de Erros

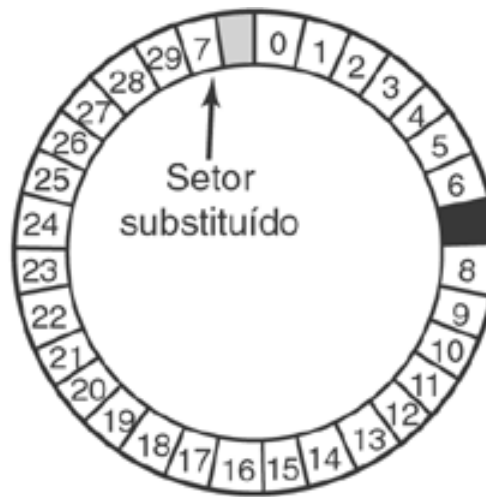
- Erro de programação (# de setor inválido) – abortar ou avisar
- Checksum error” transiente (poeira?): repetir
- Checksum error” permanente(bloco com dano físico) – marcar bloco
 - Problema: backup pode ser por bloco
 - Arquivos com blocos inválidos (sistemas de arquivo)
 - Trilhas reserva, blocos inválidos substituídos por outros reserva automaticamente
- Erro de busca: (braço foi para setor errado): alguns controladores resolvem, senão, o driver atua
- Erro de controlador: “botão de reset”, invocado pelo driver trazendo o braço para o início
- Uso de setores reserva para substituir rapidamente blocos defeituosos

Tratamento de Erros

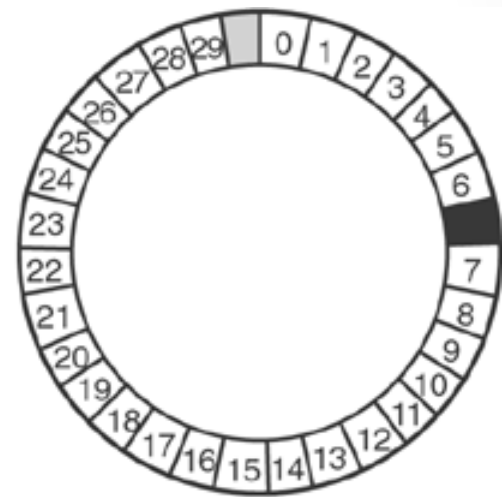
- Setores reserva



(a)



(b)



(c)

Armazenamento estável

- O objetivo é garantir que o estado dos dados no disco esteja consistente após o atendimento da requisição de leitura/escrita
- Implementado utilizando redundância
 - Premissa do RAID
- **Escritas estáveis**
 - Consiste em escrever o dado na unidade 1 e em seguida lê-lo verificando a corretude
 - ✧ Isso é feito n vezes em caso de erro. Se em todas as vezes ocorrer erro, o bloco é marcado como inválido e é remapeado
 - Caso seja bem sucedido, é escrito também na unidade 2

Armazenamento estável

- **Leituras estáveis**

- Lê primeiro da unidade 1
- Se o ECC é incorreto, a leitura é tentada n vezes
 - ✧ Se todas geraram ECC incorreto o bloco é lido da unidade 2
 - ✧ Em seguida o bloco lido da unidade 2 é copiado para a unidade 1
 - ✧ Erros espontâneos nas duas unidades no mesmo bloco tem uma probabilidade desprezível
 - Nada pode ser dito sobre desastres naturais: terremotos, o HD cair em um vulcão, ser esmagado por um disco voador, etc....

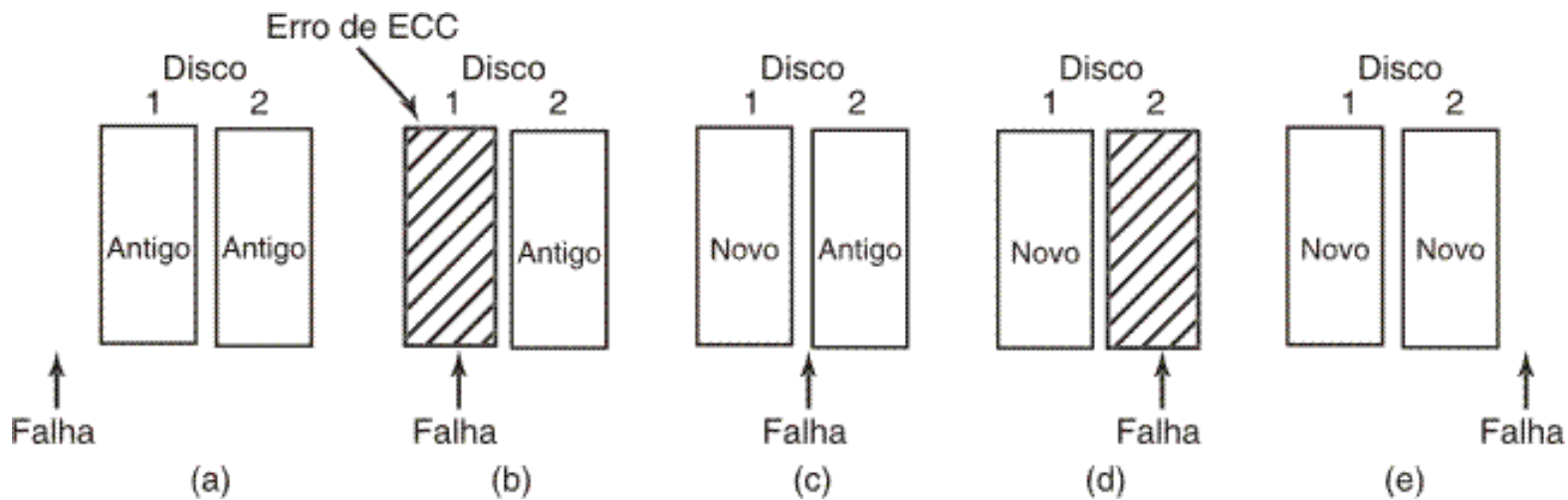
- **Recuperação de falhas**

- Após uma falha no sistema, um programa de recuperação varre todos os discos comparando os blocos
 - ✧ Caso um esteja bom e o outro defeituoso, ocorre uma cópia

Armazenamento estável

- **Recuperação de falhas**

- A falha pode ocorrer em diferentes tempos:

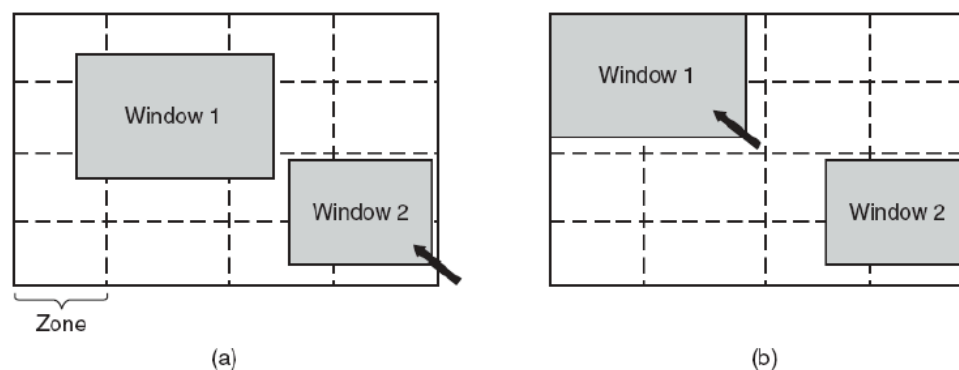


Gerenciamento de Energia

- ENIAC consumia 140 mil watts
- Com a criação dos transistores, o consumo de energia caiu drasticamente e passou a ser uma preocupação mínima.
- Hoje voltamos a ter esta preocupação
 - Até mesmo por questões ambientais....
- Maiores responsáveis por consumo de energia são dispositivos de E/S
 - Tela: 39%
 - CPU: 18%
 - Disco Rígido: 12%

Gerenciamento de Energia

- Monitor
 - Para se obter uma imagem nítida, a iluminação do monitor deve sempre ser "reanimada"
 - ✧ O que demanda um energia substancial
 - Para economizar, o SO monitora o uso e em caso de ociosidade desliga o monitor
 - Uma solução proposta consistem em dividir a tela em zonas que podem ter sua taxa de atualização minimizada (ou mesmo zerada!) caso uma zona não tivesse nenhuma janela ativa sobre ela.



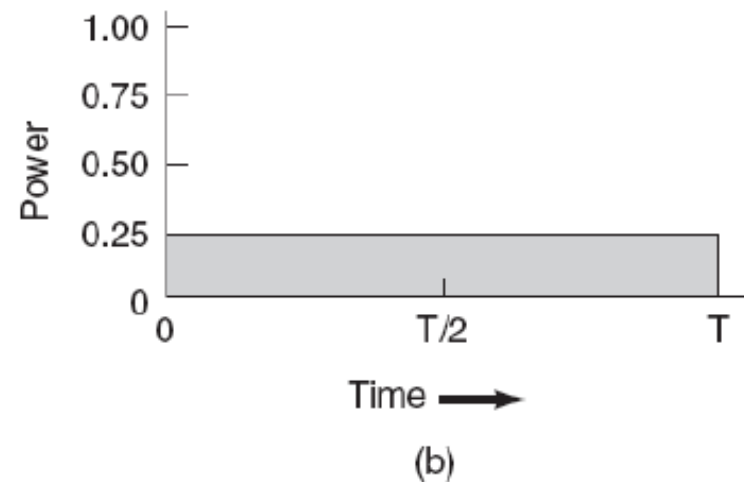
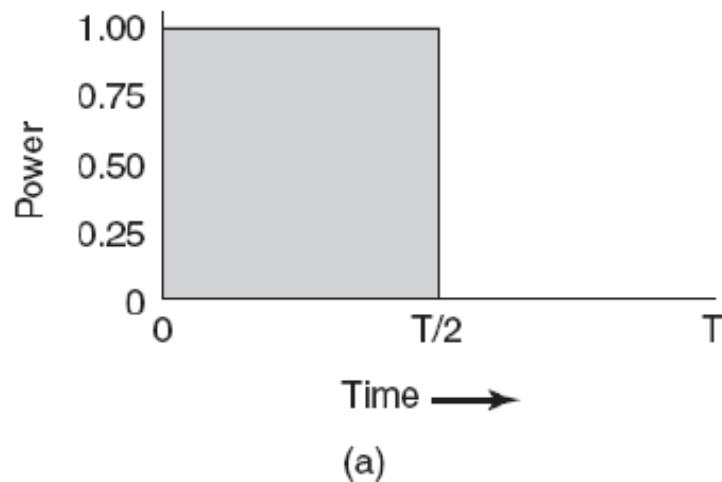
Gerenciamento de Energia

- Disco Rígido
 - Para garantir uma maior velocidade de resposta, o disco mantém a rotação nominal mesmo quando não está sendo utilizado por algum tempo.
 - ✧ Reinicializar a rotação gasta muita energia e leva bastante tempo
 - Entretanto, ao detectar inatividade, o SO pode decidir desligar o disco. Mesmo assim, é necessário ter uma segurança maior sobre a inatividade. Seria ruim desligar o disco e logo em seguida ter uma solicitação de leitura/escrita
 - Caches maiores nos controladores de disco podem ser utilizadas para maximizar o tempo de ociosidade do disco

Gerenciamento de Energia

- CPU
 - Existe uma alta correlação entre a frequência de operação da CPU e o consumo de energia.
 - Uma relação ainda mais alta existe com a tensão (voltagem)
 - Para diminuir o consumo da CPU, pode-se dinamicamente diminuir a tensão e a frequência.
 - ✧ DVFS – Dynamic Voltage and Frequency Scaling
 - Isto pode ser feito em um momento de ociosidade ou quando estiver executando uma aplicação de baixa prioridade ou com prazos (deadlines) longos.

Gerenciamento de Energia



- (a) executando na frequência de operação máxima
- (b) cortar a voltagem pela metade, corta a frequência também pela metade e o consumo de energia diminui em **quatro** vezes

Proxima aula

- Exercícios!

Referências

- TANENBAUM, Andrew S.. **Sistemas operacionais modernos**. 3. ed. São Paulo: Prentice Hall, 2009. 653 p. ISBN: 9788576052371.
 - **Capítulo 5**
- OLIVEIRA, Rômulo Silva de; CARISSIMI, Alexandre da Silva; TOSCANI, Simão Sirineo. **Sistemas operacionais**. 4. ed. Porto Alegre: Bookman, 2010. ISBN: 9788577805211.
 - **Capítulo 8**
- SILBERCHATZ, A.; Galvin, P.; Gagne, G.; **Fundamentos de Sistemas Operacionais**, LTC, 2015. ISBN: 9788521629399
 - **Capítulo 11**