TÓPICO 8

Gerência de E/S

Introdução

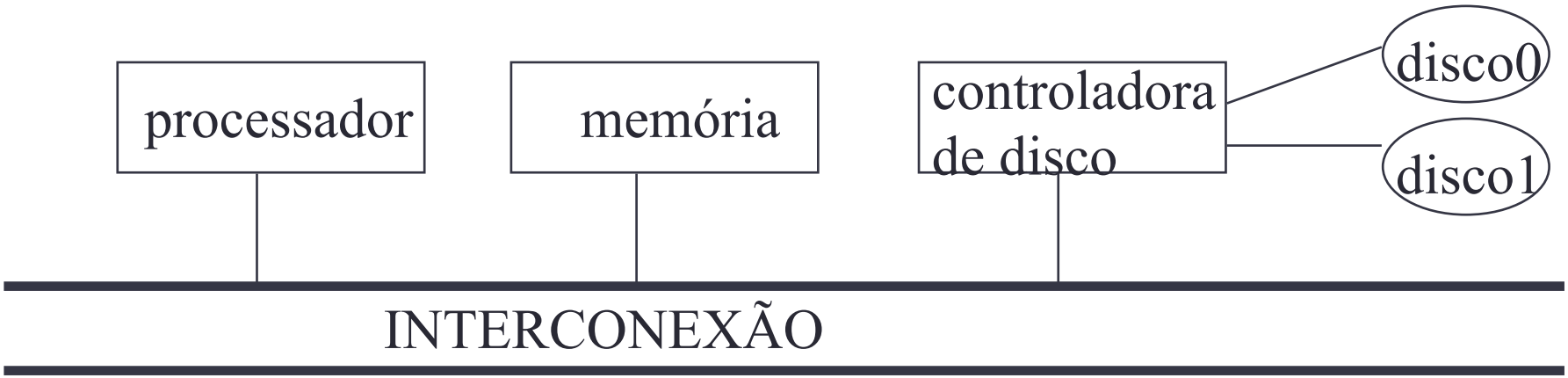
- Uma das funções do sistema operacional é criar uma máquina abstrata para o usuário que seja mais simples que a máquina física.
- Para implementar esta abstração, o SO deve conhecer muito bem o hardware sobre o qual ele opera.
- Cada dispositivo de hardware possui comandos específicos e, para utilizá-los, o SO deve conhecer muito bem o funcionamento interno de cada dispositivo.

Classificação dos dispositivos de E/S

- De uma maneira bem geral, os dispositivos de E/S podem ser divididos em duas categorias:
  - Dispositivos de bloco
    - A informação é armazenada em unidades de tamanho fixo (blocos).
    - A recuperação da informação é feita por acesso direto.
    - Exemplos: disco
  - Dispositivos de caracter
    - A informação é aceita em unidades de tamanho variável.
    - Geralmente a informação é armazenada temporariamente ou não é armazenada.
    - Exemplo: terminais, impressoras, mouse, etc.

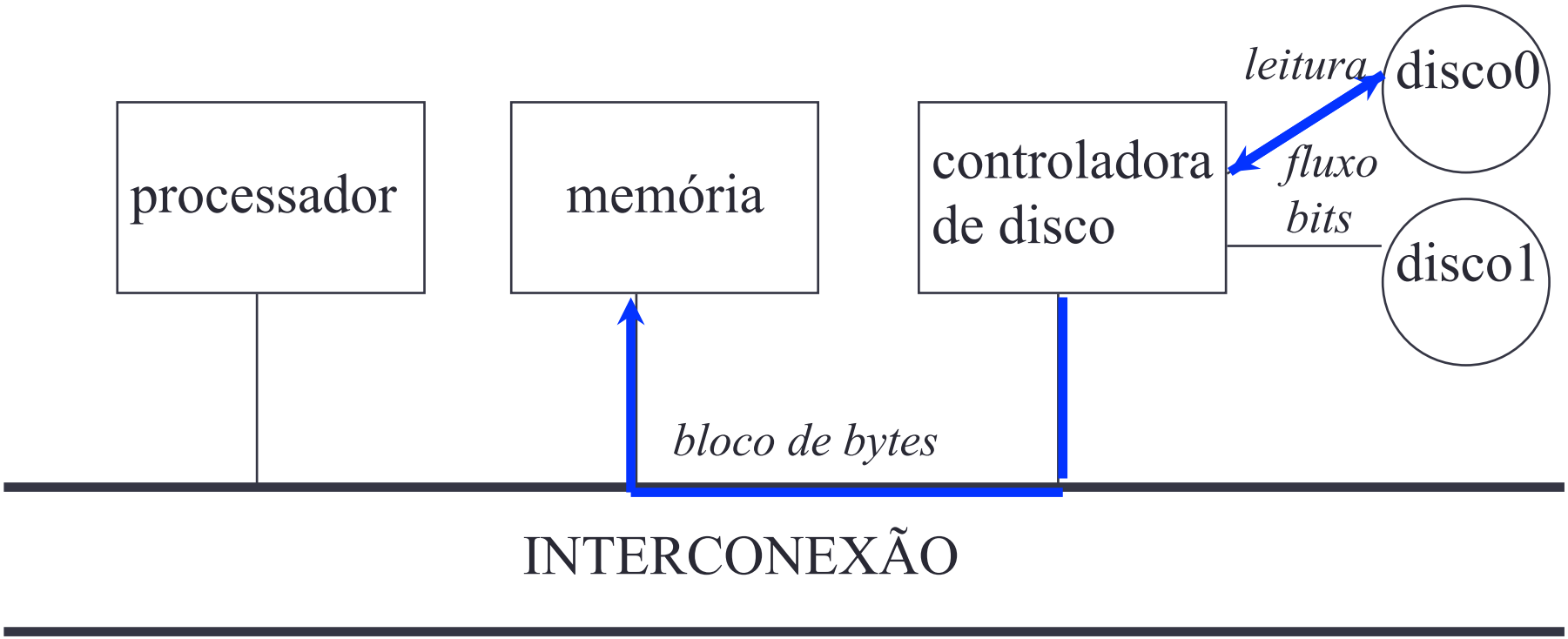
Controladoras de Dispositivos

- Geralmente, as unidades de E/S são compostas de uma parte mecânica (o dispositivo propriamente dito) e uma parte eletrônica (controladora de dispositivo) que controla a operação do dispositivo.
- Os sistemas operacionais programam a controladora e não o dispositivo.



Controladoras de Dispositivos

- Leitura de um bloco



Entrada/Saída mapeada em memória

- A controladora e o SO trocam constantemente informações. Esta comunicação é feita através de registradores.
- Os registradores são geralmente mapeados em posições fixas da memória.
- O SO solicita E/S escrevendo comandos nos registradores das controladoras.
- Cada controladora tem um conjunto de comandos específicos e de baixo nível que são descritos no manual da controladora.
- Uma vez aceito o comando, a controladora trabalha simultaneamente com a CPU, somente interrompendo-a no final do serviço.

Direct Memory Access (DMA)

- Quando a controladora lê dados de um dispositivo, estes são armazenados inicialmente em um buffer interno.
- Estando o bloco completo e correto no buffer da controladora, o mesmo deve ser escrito em memória.
- Quem vai escrever estes dados em memória?
  - A CPU
    - Muito tempo da CPU é gasto com operações de cópia
  - A controladora
    - Através de operações de acesso direto à memória (DMA)

Direct Memory Access (DMA)

- Leitura de dados com DMA
  1. O SO informa o endereço de memória para o qual o dado deve ser lido e o número de bytes a serem transferidos.
  2. Estas informações são escritas pela controladora em dois registradores DMA, um de endereço e um contador.
  3. A controladora lê o bloco do dispositivo de E/S.

Direct Memory Access (DMA)

4. Após ter o bloco completo em seu buffer, a controladora transfere o conteúdo de seu buffer (byte a byte) para o endereço de memória especificado no registrador DMA.
5. Quando o contador chega a zero, a controladora gera uma interrupção.
6. O SO inicia a execução sabendo que o bloco já se encontra no endereço de memória destino.

Software de E/S - Introdução

- A maioria dos sistemas operacionais modernos utiliza o conceito de independência de dispositivos.
  - *Independência de dispositivos*: execução de um mesmo programa com dispositivos físicos diferentes. Dispositivos lógicos de E/S.
- *Uniformidade da identificação:* o nome de um dispositivo lógico ou de um arquivo não deve depender do dispositivo físico ao qual está associado no momento.

Software de E/S - Estruturação

- O Software de E/S pode ser estruturado da seguinte maneira (do nível mais baixo para o nível mais alto):
  - Manipuladores de interrupção
  - Drivers de dispositivos
  - SO independente do dispositivo
  - Aplicações de usuário

Software de E/S
Manipuladores de Interrupção

- As operações de E/S podem ser tratadas de maneira síncrona ou assíncrona.
- A maioria dos SO tradicionais as trata de maneira síncrona, ou seja, o processo que solicitou uma operação de E/S fica bloqueado até que esta operação se complete.
- Assim, o SO tradicional pode tratar várias operações ao mesmo tempo, porém somente uma interrupção por processo pode estar sendo tratada.
- Quando ocorre uma interrupção, isto indica que a operação se completou e o SO coloca o processo (que estava bloqueado) na fila ready.

Software de E/S
Drivers de dispositivo

- O driver de dispositivo é a parte do SO que executa código que depende do hardware do dispositivo.
- Cada dispositivo ou classe de dispositivos possui seu driver específico.
- No caso do disco, o driver de disco é o responsável por escrever os comandos da controladora do disco.
- O driver recebe solicitações abstratas e as traduz para comandos específicos (e.g. comandos de controladora).

Software de E/S
Software independente de dispositivo

- Existem operações, de muito baixo nível, que só podem ser executadas com comandos dependentes de dispositivos.
- A maioria das operações, no entanto, pode ser executada de maneira independente.
- Uma operação que pode ser executada de maneira independente de dispositivo pode ser executada com comandos de baixo nível, por questões de performance.
- Assim, fica a cargo do projetista do SO que comandos serão independentes de dispositivo ou não.

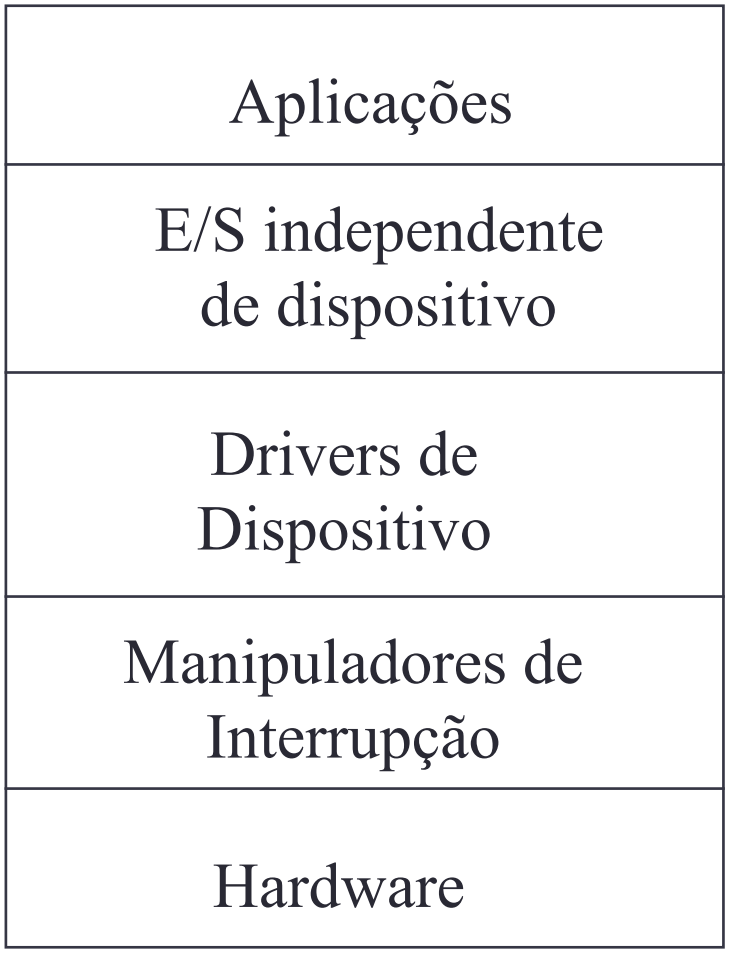
Software de E/S
Software independente de dispositivo

- Funções típicas independentes de dispositivo:
  - Fornecimento de uma interface uniforme ao usuário.
  - Mapeamento do nome simbólico do dispositivo para seu driver específico.
  - Proteção de acesso
  - Fornecimento de um tamanho de bloco independente do dispositivo.

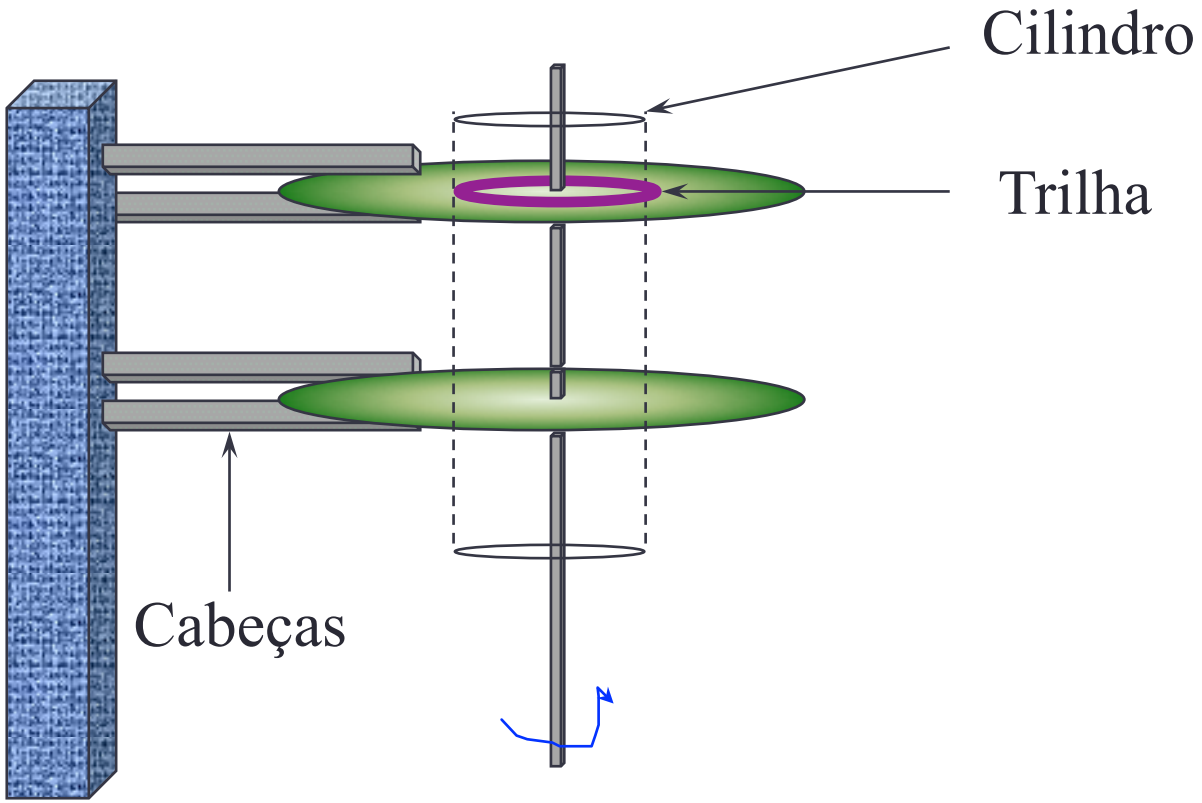
Software de E/S
Software independente de dispositivo

- Funções típicas independentes de dispositivo (cont):
  - Tratamento da bufferização
    - Conversão de solicitações de tamanho arbitrário em solicitações de tamanho fixo
  - Alocação/Liberação de blocos
  - Alocação/Liberação de dispositivos de acesso exclusivo
  - Tratamento de erros não transientes

Software de E/S
Camadas envolvidas



O Disco
Hardware



O Disco
Hardware

- Cada disco é uma estrutura plana e circular.
- A informação pode ser gravada nos dois lados.
- Para ler dados, a cabeça de leitura/gravação deve ser colocada na posição do disco onde o dado se encontra.
- A superfície do disco é organizada em trilhas.
- Cada trilha é organizada em setores.
- Um disco com cabeças fixas possui várias cabeças que permitem a leitura rápida de dados.
- O conjunto de trilhas acessadas ao mesmo tempo pelas diversas cabeças é chamado cilindro.

O Disco
Hardware

- A velocidade de acesso a um disco é afetada por 3 componentes:
  - tempo de seek
  - latência
  - tempo de transferência
- O tempo total para tratar uma solicitação de disco é a soma destes 3 tempos.

O Disco - Hardware
Tempo de seek

- Para acessar um bloco no disco, é necessário primeiro mover a cabeça para a trilha ou cilindro apropriado.
- Este movimento da cabeça é chamado *seek.*
- O tempo necessário para executar o seek é chamado *seek time.*

O Disco - Hardware
Latência

- Estando a cabeça posicionada na trilha correta, é necessário esperar que o bloco desejado se posicione exatamente abaixo da cabeça de leitura/gravação (através da rotação).
- Este atraso é conhecido como *latência.*

O Disco - Hardware
Transferência

- Estando posicionada no bloco correto, a cabeça de gravação lê o dado e o dado é transferido do disco para a memória.
- Este tempo é conhecido como *tempo de transferência.*

Algoritmos de escalonamento do braço

- Dentre os 3 tempos citados anteriormente, o tempo de seek é geralmente o dominante.
- Assim, a maioria dos algoritmos de escalonamento de braço visa reduzir o tempo de seek.
- Algoritmos mais conhecidos:
  - FCFS
  - SSTF
  - Elevador e suas variantes

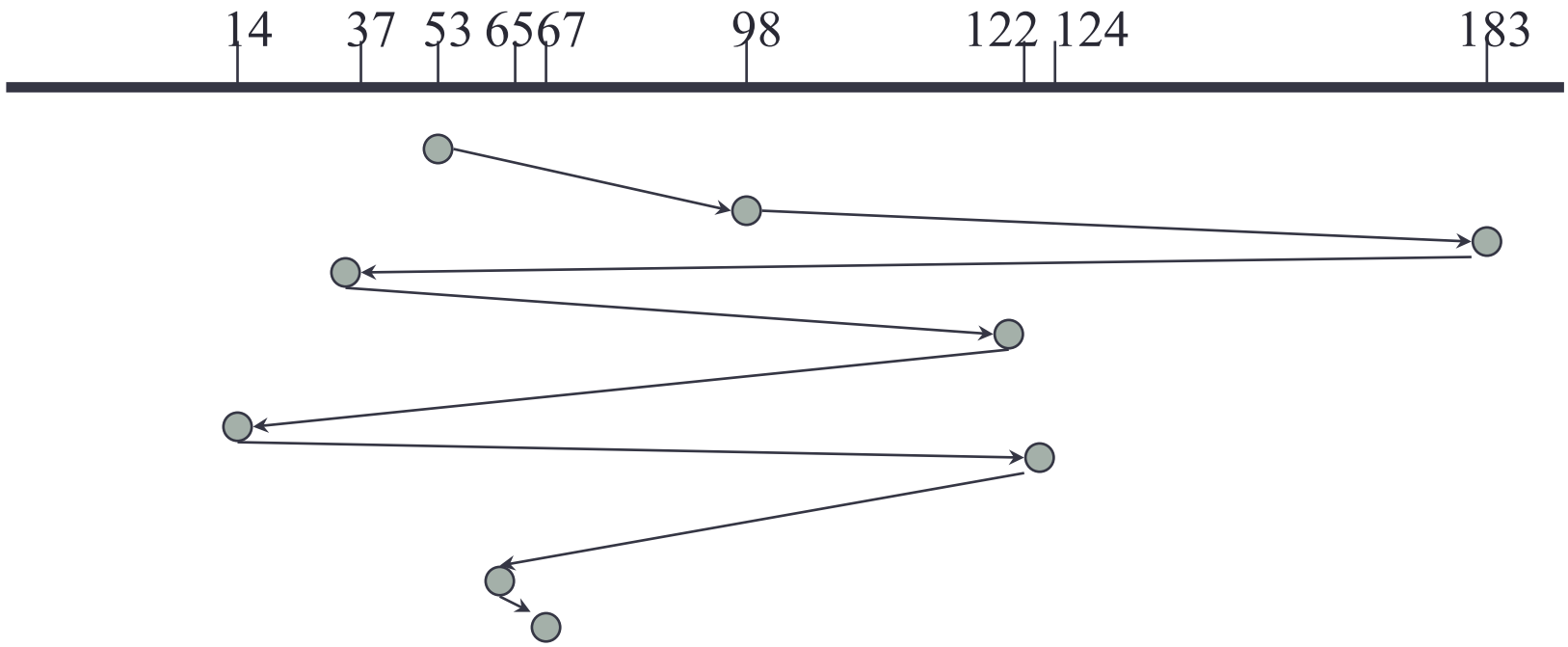
Algoritmos de escalonamento do braço - FCFS

- First Come First Served
  - Este algoritmo serve as requisições segundo a ordem FIFO.
  - Simples implementação
  - Justo
  - Os tempos de serviço podem ser muito longos.

Algoritmos de escalonamento do braço - FCFS

- First Come First Served

Fila de requisições: 98,183,37,122,14,124,65,67



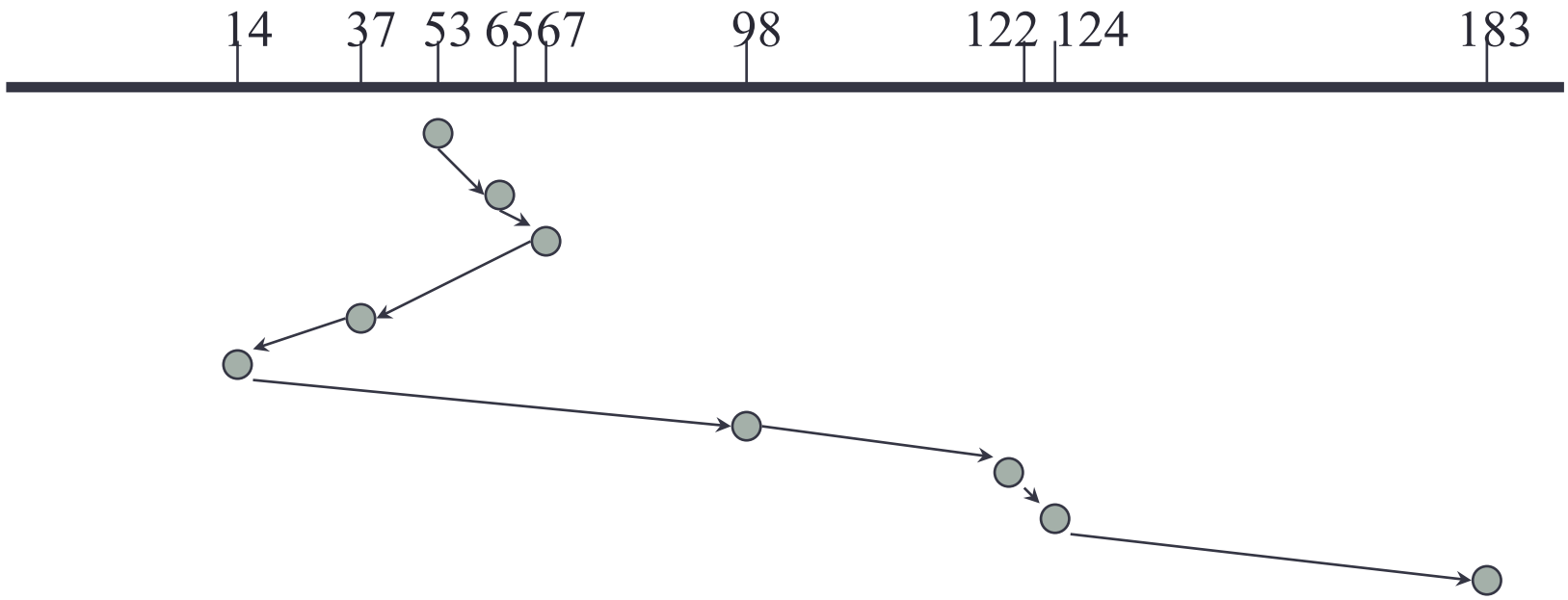
Algoritmos de escalonamento do braço - SSTF

- Shortest Seek Time First
  - Este algoritmo visa reduzir o tempo de seek, servindo todas as solicitações que estão próximas da cabeça de gravação.
  - SSF pode causar starvation facilmente.
  - Apesar de fornecer uma boa performance, o algoritmo não é ótimo.

Algoritmos de escalonamento do braço - SSTF

- Shortest Seek Time First

Fila de requisições: 98,183,37,122,14,124,65,67



Algoritmos de escalonamento do braço - Elevador

- Este algoritmo possui duas variantes: SCAN e LOOK.
- O algoritmo do elevador trata as requisições de maneira similar ao comportamento dos elevadores.
- Começa-se em uma direção, servindo todos as requisições que se encontram no caminho. Ao chegar ao fim, volta-se na direção contrária.

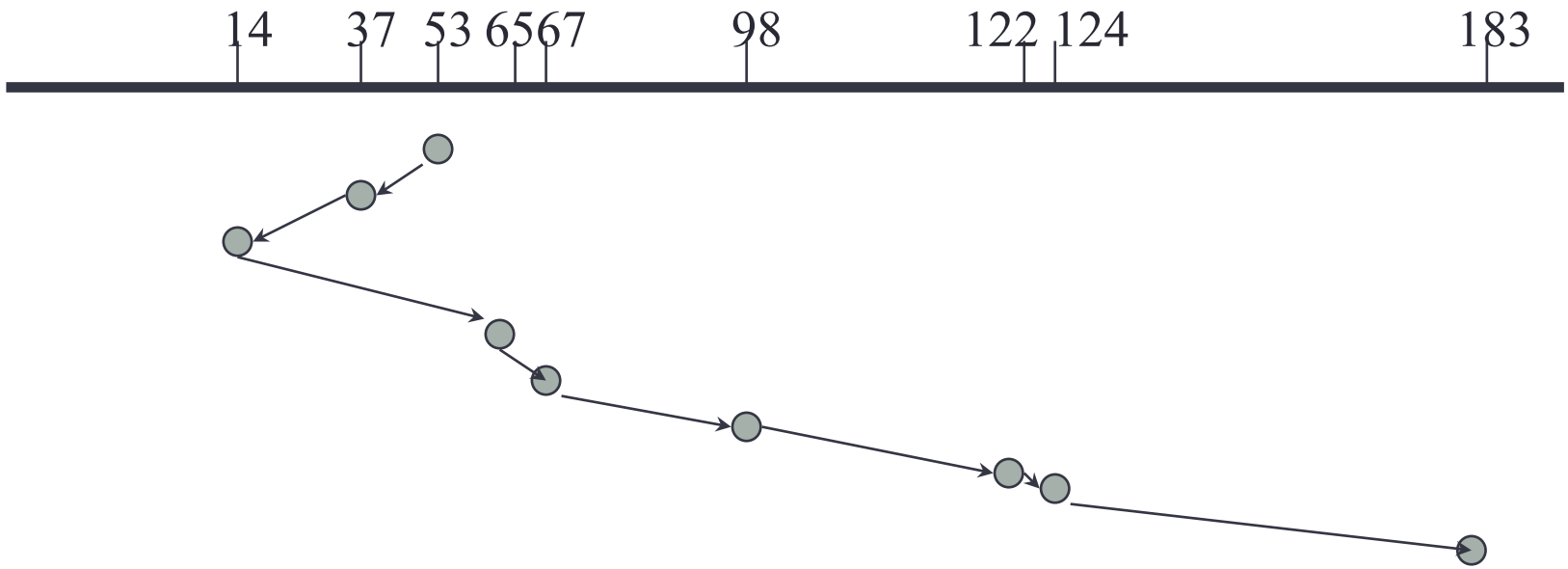
Algoritmos de escalonamento do braço - Elevador

- Variações:
  - SCAN: O braço do disco vai até o final do disco e depois volta, fazendo um scan, até o setor 0.
  - LOOK: O braço não vai até o final do disco. Ele só precisa ir até o último bloco na fila de requisições.
  - C-SCAN e C-LOOK: ao invés de servir as requisições na direção oposta, o braço volta ao início e serve as requisições sempre na mesma direção.

Algoritmos de escalonamento do braço – Elevador (variante Look)

- Funcionamento:

Fila de requisições: 98,183,37,122,14,124,65,67



Sistemas RAID

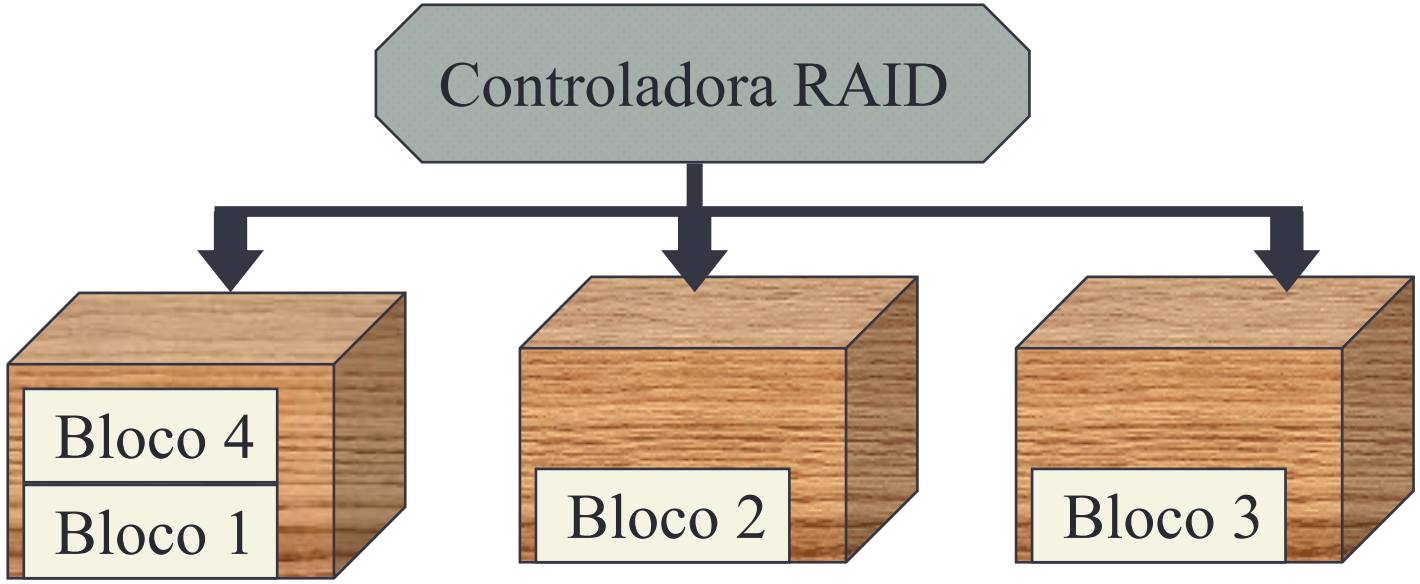
- Atualmente, para se reduzir o tempo de serviço de uma requisição, utilizamos vários discos trabalhando juntos.
- Estes discos que trabalham juntos e armazenam partes da requisição compõem o sistema RAID (redundant array of inexpensive disks).

Sistemas RAID

- O sistemas que incorporam a tecnologia RAID visam oferecer performance e confiabilidade a um preço compatível.
- Existem basicamente os seguintes níveis:
  - RAID 0 - striping
  - RAID 1 - mirroring
  - RAID 5 - striping com paridade

Sistemas RAID
RAID 0

- Neste nível, os dados são “derramados” em diversos discos, para melhorar o desempenho de operações de leitura/escrita.

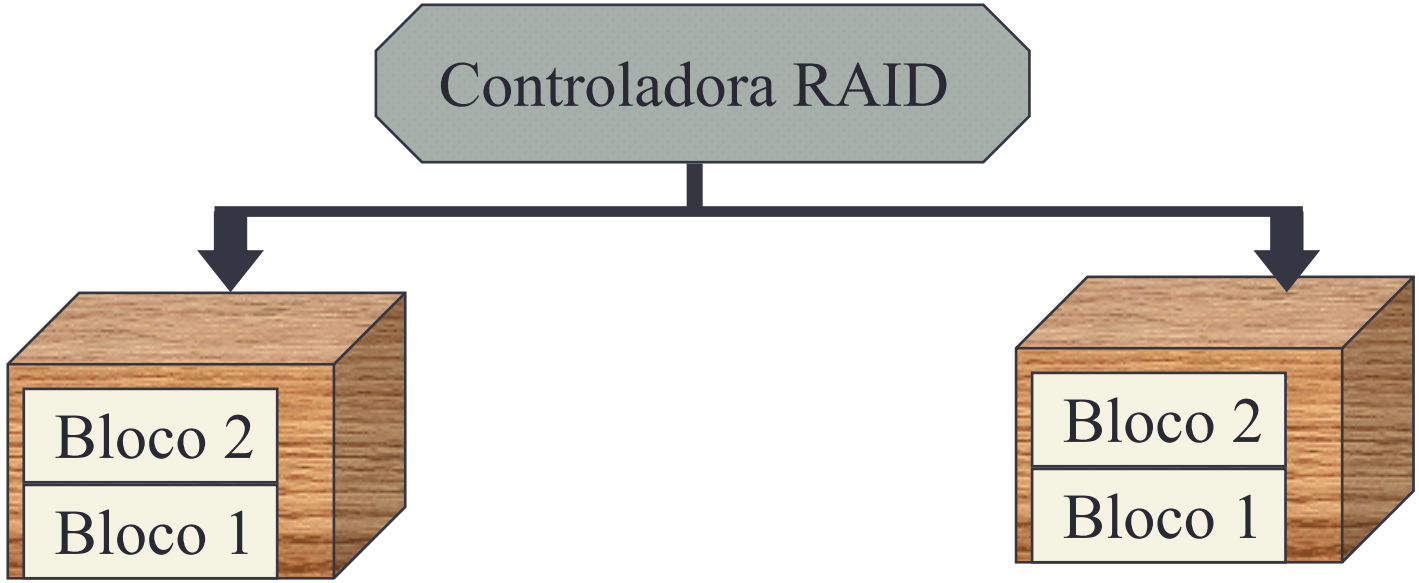


Sistemas RAID
RAID 0

- Vantagens
  - Um stripe de *n* discos, faz com que o tempo de serviço seja *n* vezes menor, já que o acesso é feito em paralelo
  - Facilidades de implementação de multivolumes
- Desvantagens
  - Confiabilidade

Sistemas RAID
RAID 1

- No nível 1, os drives de disco são agrupados em pares. Todo dado escrito no primeiro drive é duplicado no segundo.

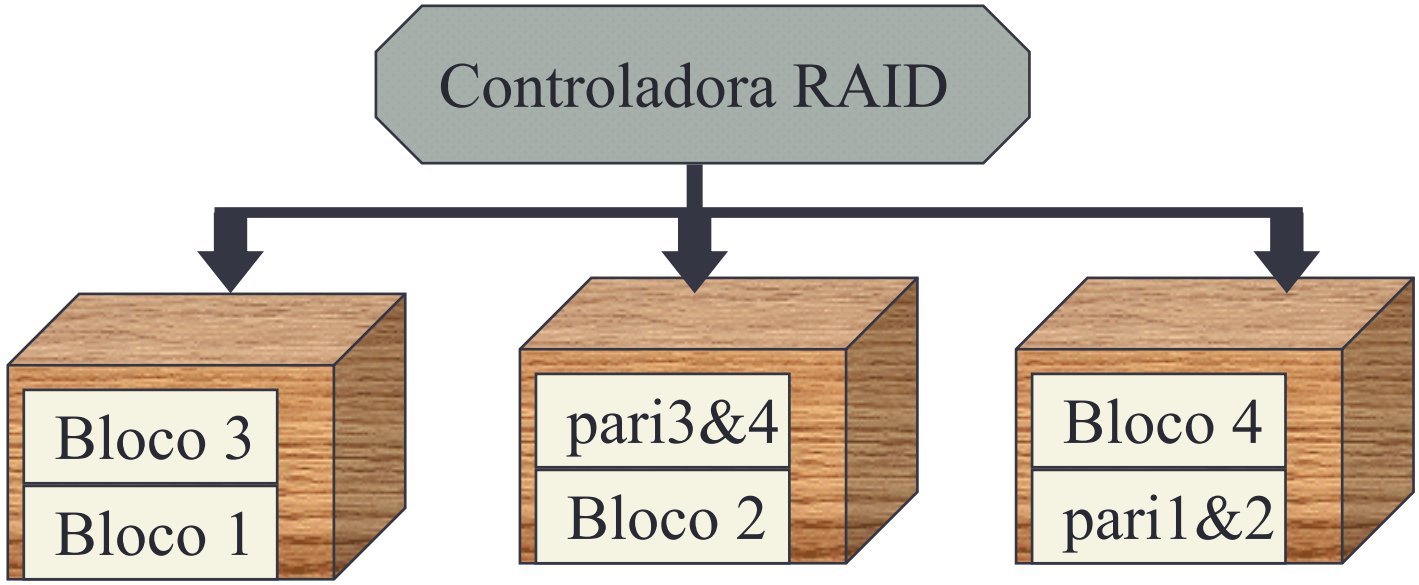


Sistemas RAID
RAID 1

- Vantagens
  - A performance das operações de leitura é maior que o disco tradicional.
  - Tolerância a falhas.
- Desvantagens
  - Custo em espaço em disco.

Sistemas RAID
RAID 5

- No RAID 5, além de fazer o striping, um bloco de paridade é calculado. Um bloco de dados e seu respectivo bloco de paridade nunca são escritos no mesmo disco.



Sistemas RAID
RAID 5

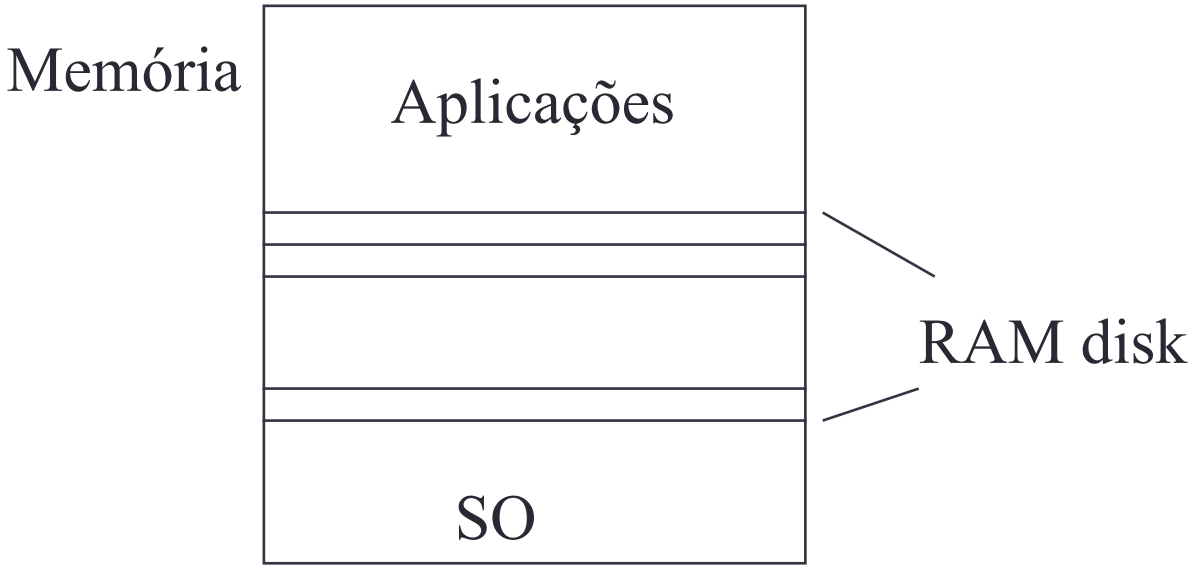
- O RAID 5 tenta obter as vantagens do RAID 0 e do RAID 1.
- Considerações:
  - Performance de leitura similar ao RAID 0.
  - Performance de escrita menor que o RAID 0 (o bloco de paridade deve ser calculado e escrito, no melhor dos casos).
  - Requer mais espaço em disco que o RAID 0, porém menos que o RAID 1.
  - Suporta falha de 1 disco.

Sistemas RAID
Implementação

- A tecnologia RAID pode ser implementada
  - por software: geralmente a nível de SO
  - por hardware: com uma controladora específica

RAM Disk

- A RAM disk é uma área em memória onde é simulado um acesso ao disco.
- As operações sobre a RAM disk são *Lê bloco* e *Escreve bloco*



Clocks

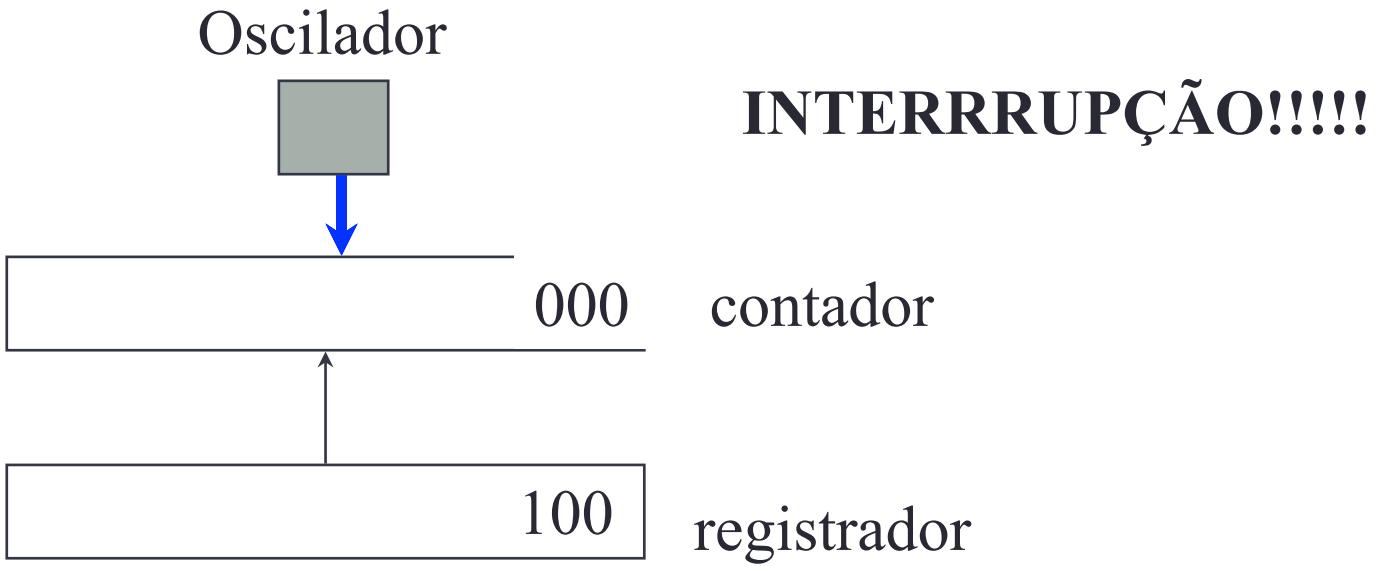
- Os clocks também são conhecidos por temporizadores e mantém o tempo da máquina.
- São acessados através de um driver de dispositivo, que não é de bloco nem de caracter.

Clocks

- O Clock do hardware do computador é realmente um temporizador.
- Os clocks são normalmente programáveis, construídos através de um oscilador de cristal, um contador e um registrador.
- A frequência de oscilação do cristal depende do cristal escolhido e é medida em MHz.

Clocks Programáveis

- A cada oscilação, o contador é decrementado. Quando o contador chega a 0, o clock interrompe a CPU.



Clocks Programáveis

- A implementação do relógio interno da máquina é a seguinte:
  - Com um dado no formato DDMMYY, o SO o transforma no número de interrupções de tempo desde as 12:00 do dia 01/01/70 até a data informada (Unix).
  - A cada nova interrupção este valor é incrementado.
  - O time-of-day é geralmente armazenado em um registrador especial alimentado por uma bateria.

Driver do clock

- Geralmente o driver do clock tem as seguintes funções:
  - Manter o tempo real
  - Implementar o quantum do escalonador
  - Contabilizar o uso do processador
  - Gerenciar a chamada ALARM
  - Fornecer watch-dogs
  - Monitorar o sistema

Driver do clock
Manter o tempo real

- Incrementar o contador a cada interrupção de tempo.
- Problema: com 16 bits e 60 MHz de frequência, só conseguimos armazenar 2 anos.
  - Solução 1: Usar 64 bits
  - Solução 2: Guardar o tempo real em segundos (136 anos).
  - Solução 3: Contar as interrupções a partir do tempo fornecido pelo usuário no boot do sistema.

Driver do clock
Implementar o quantum

- Para implementar o quantum de tempo em um escalonador preemptivo:
  - A cada transição ready-running, o escalonador inicializa um contador com o valor do quantum.
  - O driver do clock decrementa periodicamente este contador.
  - Quando o contador chega a 0, o driver ativa o escalonador, que colocará um novo processo para rodar.

Driver do clock
Contabilização de uso

- O objetivo aqui é contabilizar o uso do processador por processo.
  - Solução 1: manter um contador com o tempo de CPU utilizado pelo processo. Na troca de contexto, o contador é carregado de/para a tabela de processos.
  - Solução 2: sempre que um processo obter a CPU, é carregado em uma variável global um ponteiro para a sua entrada na tabela de processos. A cada interrupção, o campo referente ao contador na tabela de processos é decrementado.

Driver do clock
Implementar o ALARM

- Para simular o alarm, o sistema geralmente simula vários clocks virtuais em um único clock físico.
- Ao final do tempo solicitado, uma interrupção (sinal) é enviada ao processo que executou o alarm, desviando a execução para a rotina específica de tratamento.

Driver do clock
Watch-dogs

- Neste caso, ao contrário de enviar um sinal, desejamos que um determinado procedimento seja executado após decorrido um certo intervalo de tempo.

Driver do clock
Monitoração

- Controle de tempo de processamento de um programa.
- A cada interrupção de tempo, o driver verifica se o processo em execução está sendo monitorado. Se sim, o driver determina em que faixa de endereços está o PC e incrementa a variável referente a esta faixa.

Terminais
Hardware

- Tipos de terminais
  - Interface RS232
    - Hardcopy (impressora)
    - TTY de vidro (terminais burros)
    - Terminal inteligente (entendem sequências de escape)
    - Blit (terminais de alta capacidade de processamento)
  - Mapeado em memória
    - Orientado a caracter
    - Orientado a bit

Terminais Mapeados em Memória

- A interface entre o computador e o terminal é feita através de uma memória especial (RAM de vídeo), pertencente ao espaço de endereçamento do computador.
- A RAM de vídeo é controlada pelo controlador de vídeo.
- A tela é dividida em pequenas unidades chamadas pixels.

Entrada de dados

- O driver de teclado é o responsável por:
  - Coletar os dados digitados
    - Modo raw
    - Modo cooked
  - Tendo recebido o dado, o driver de teclado necessita processá-lo.
  - Antes de processá-los, os caracteres são geralmente armazenados.
  - O driver de teclado deve enviar os dados para a tela (eco) ou não.