



SISTEMAS OPERACIONAIS

Módulo 10 – Entrada e Saída

Prof. Daniel Sundfeld
daniel.sundfeld@unb.br



INTRODUÇÃO

- A gerência de E/S é uma das maiores e mais complexas funções de um SO
- Principal causa é a grande diversidade de dispositivos de E/S disponíveis no mercado
- Uma das funções do SO é criar uma máquina abstrata para as aplicações e o usuário
- Para criar essa máquina, é necessário saber o funcionamento interno dos dispositivos

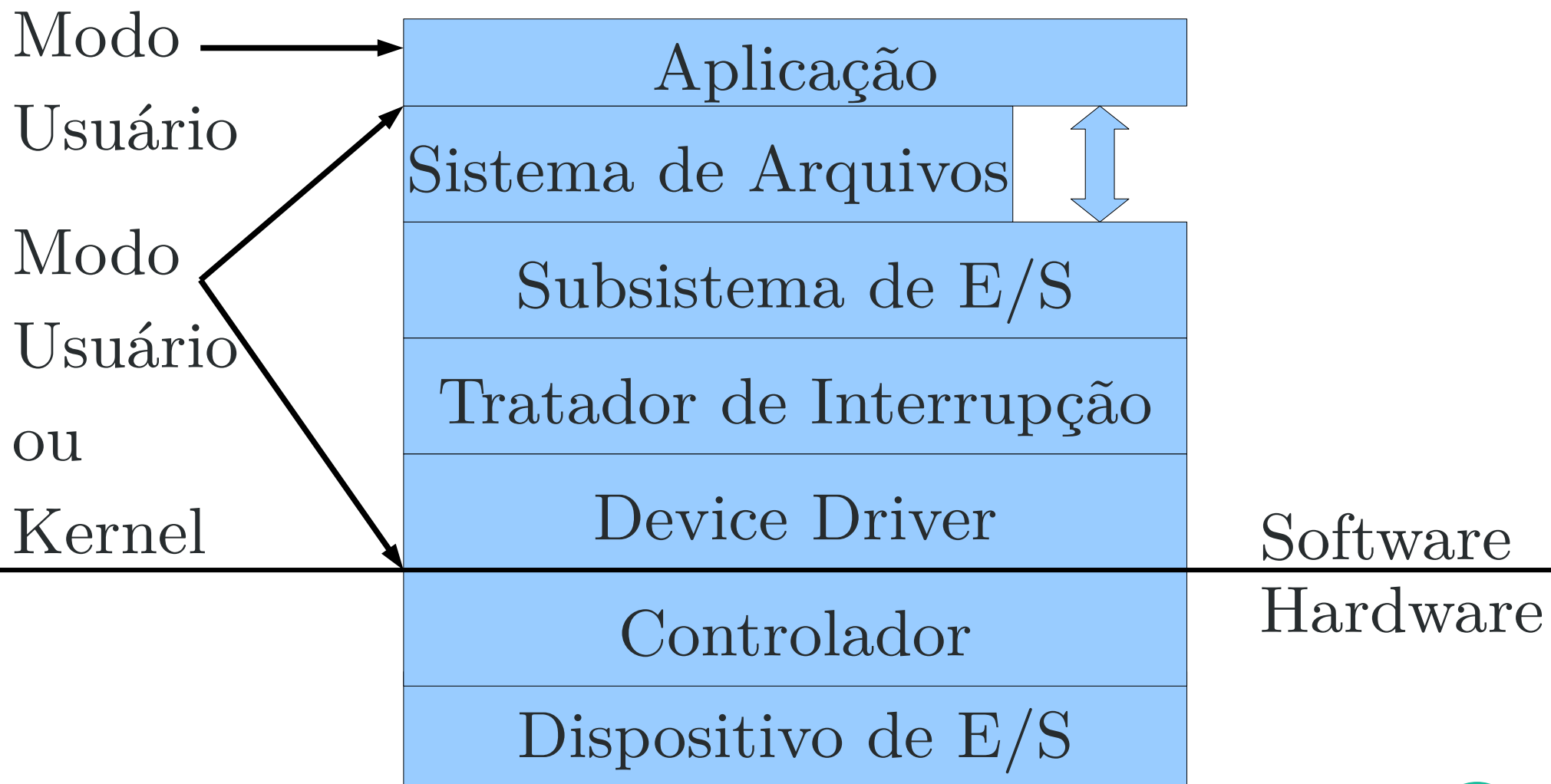


INTRODUÇÃO

- Para conseguir implementar essa abstração o SO divide em múltiplas camadas a gerência de E/S:
 - Subsistema de E/S: isola a complexidade dos dispositivos físicos de E/S;
 - Device Driver: implementa detalhes de cada periférico em modo protegido;
 - Tratador de Interrupção: camada que implementa a lógica de interrupções;
 - Controlador e Dispositivo de E/S: o hardware em questão.



DISPOSITIVOS DE E/S





DISPOSITIVOS DE E/S

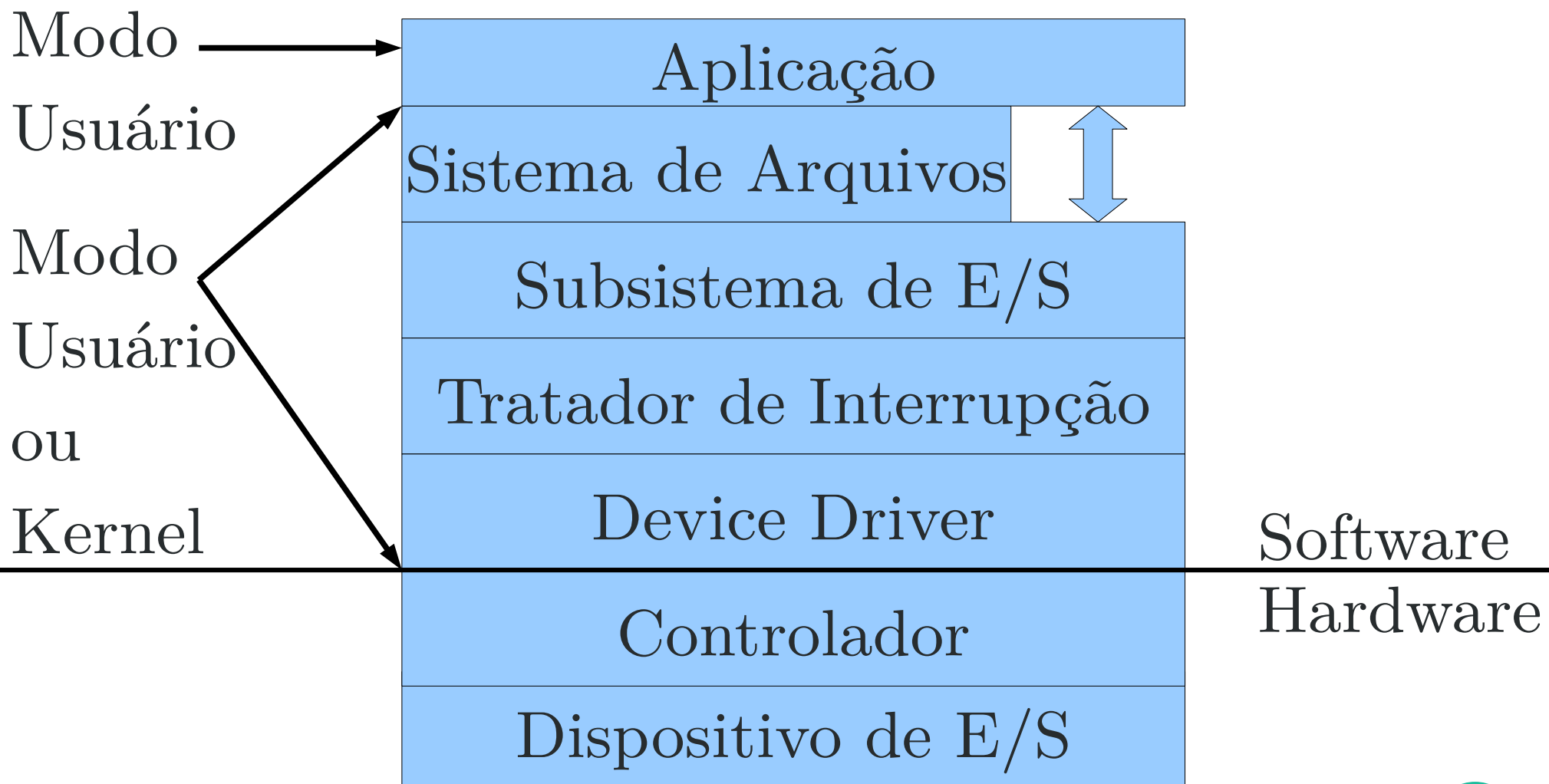
- Os dispositivos de E/S são divididos na categoria de blocos ou caracteres:
- **Blocos:** armazena informações de tamanho fixo, cada uma com seu próprio endereço (Tamanhos entre 512 a 64k)
- Cada bloco pode ser lido e escrito independentemente de todos os outros
 - HDs, Blu-Ray, Pendrives



DISPOSITIVOS DE E/S

- **Caracteres:** envia e aceita um fluxo de caracteres, desconsiderando qualquer estrutura de bloco
- Não é endereçável, não tem operações de busca
- Impressoras, interfaces de rede, mouses

SUBSISTEMA DE E/S





SUBSISTEMA DE E/S

- A comunicação com o Subsistema de E/S é realizado através das **rotinas de E/S**
- Essas rotinas permitem a comunicação com o dispositivo sem se preocupar com detalhes de implementação
- Por exemplo: ao criar um arquivo não é necessário se preocupar com detalhes além do nome e permissões. Trilhas, setores, disco, são calculados pelo SO



SUBSISTEMA DE E/S

- As rotinas de E/S são independentes do dispositivo, por exemplo, um arquivo não altera o método chamado se ele está no disco ou pendrive, as chamadas de vídeo não dependem se são um monitor LCD, LED ou CRT
- As rotinas de E/S podem ser efetuadas diretamente no subsistema de E/S (como para vídeo) ou podem ser executadas em uma outra subcamada, como o sistema de arquivos ou a gerência de USB



SUBSISTEMA DE E/S

- Quando as chamadas são realizadas diretamente no subsistema de E/S elas são **chamadas explícitas**
- Caso contrário, é uma chamada implícita, onde a partir de alguma chamada de sistema são geradas chamadas para o subsistema de E/S a partir de algum outro módulo do Sistema Operacional



SUBSISTEMA DE E/S

- As operações de E/S também podem ser classificadas em:
 - Síncronas: onde a aplicação aguarda o término, ficando em um estado bloqueado. A maioria das chamadas é assim;
 - Assíncronas: a aplicação continua a execução após realizar a chamada. Neste caso, é necessário implementar uma forma de avisar a aplicação o término da chamada de E/S.



SUBSISTEMA DE E/S

- Cada dispositivo trabalha com unidades de tamanho diferentes. O subsistema de E/S cria uma unidade de transferência básica
- Ele repassa as chamadas às camadas inferiores e obtém as informações das camadas superiores sem conhecimento da informação



SUBSISTEMA DE E/S

- O tratamento de erro de operações é realizado na camada mais baixa
- Porém alguns erros não são tratados e são enviados à camada superior, como erros de permissão
- Erros de gravação e leitura dependem do tratamento dado pelo SO



SUBSISTEMA DE E/S

- Além disso, os diferentes usuários e aplicações querem compartilhar os dispositivos de E/S
- Cabe ao SO gerenciar esse compartilhamento
- O uso simultâneo de arquivos, por exemplo, é uma funcionalidade implementada em alguns sistemas de arquivos.
- Locks de arquivos, permitindo o uso exclusivo também existem em alguns sistemas operacionais



SUBSISTEMA DE E/S

- No caso de dispositivos de E/S, deve-se ter um tratamento dependendo do dispositivo
- Alguns dispositivos como placa de rede, podem ser utilizados simultaneamente por diversas aplicações
- Outros dispositivos, como a impressora, podem utilizar de técnica de spooling para gerenciar a ordem de operações garantido a integridade e que uma operação não interfira na outra



SUBSISTEMA DE E/S

- A técnica de buffering, quando implementada no SO, pode ser realizada na camada de subsistema de E/S
- Uma região de memória é reservada para a comunicação dos dispositivos de E/S, assim quando uma aplicação acessa essa região, ela lê da memória principal e não precisa aguardar o término de uma operação de E/S

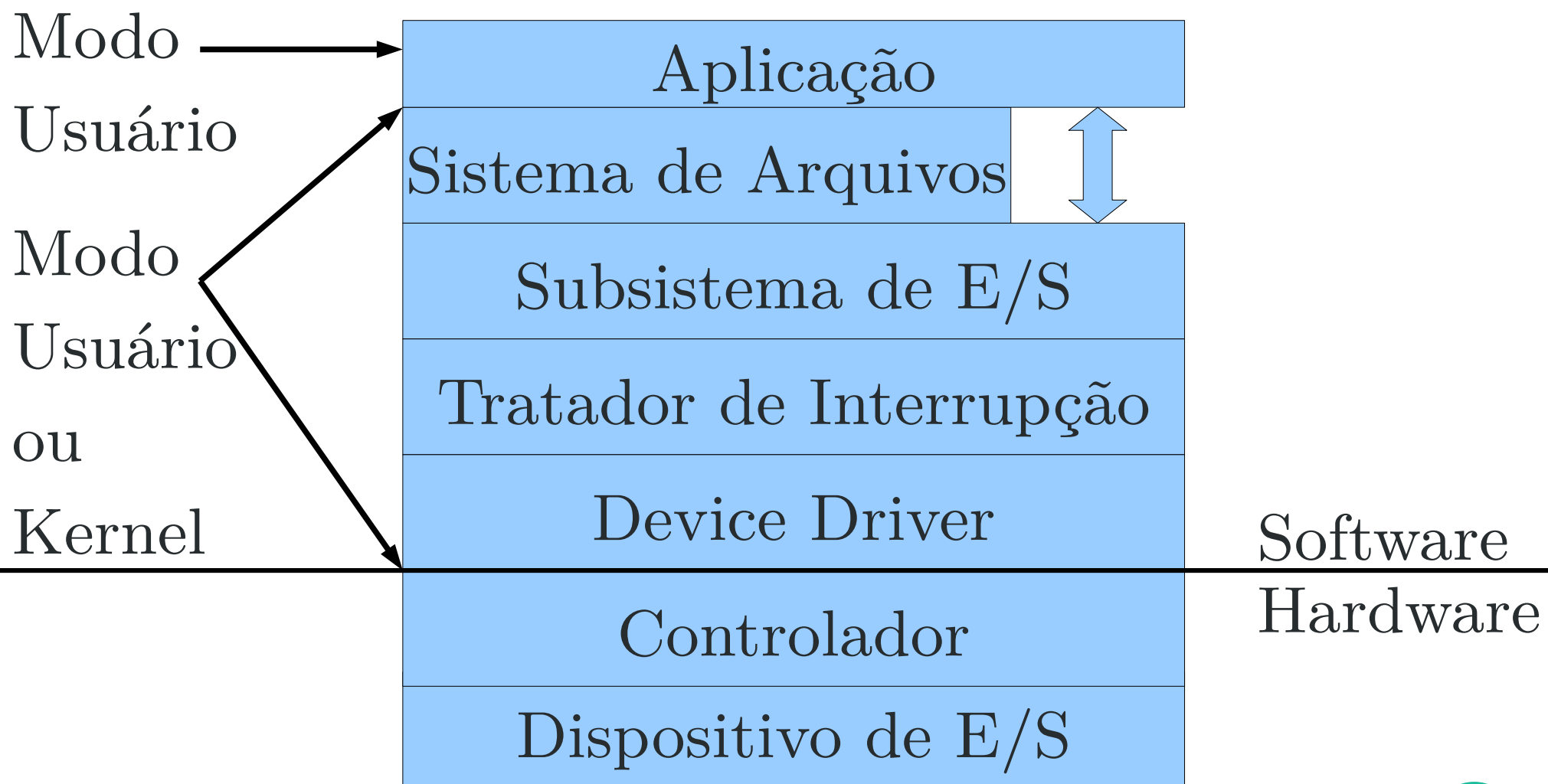


SUBSISTEMA DE E/S

- A principal função do subsistema de E/S é criar uma interface padronizada, independente dos dispositivos, com o device driver
- Ele deve permitir a inclusão de novos dispositivos e novos drivers, sem a interferência ou alteração na camada do subsistema de E/S



DEVICE DRIVER





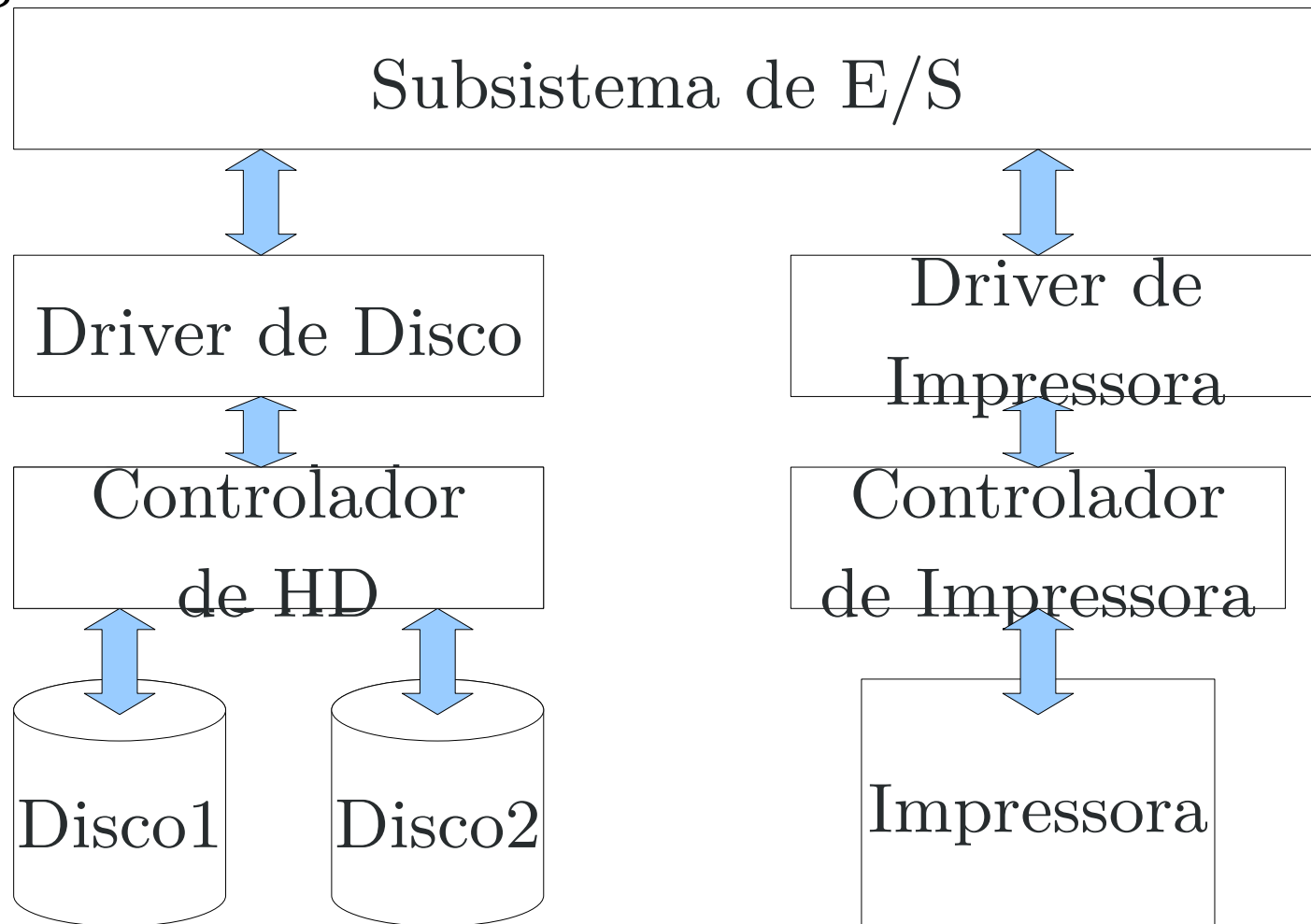
DEVICE DRIVER

- O device driver (driver de dispositivo) tem como função implementar a comunicação do subsistema de entrada e saída com os controladores de E/S
- O subsistema de E/S é uma camada genérica e que possui uma interface para todos os dispositivos
- O driver de dispositivo implementa as chamadas específicas do dispositivo, recebendo comandos genéricos e traduzindo para específicos



DEVICE DRIVER

- Visão geral do SO:





DEVICE DRIVER

- Geralmente o driver de dispositivo é escrito pelo fabricante e fornecido junto com ele
- Alguns fabricantes provêm driver para diversos sistemas operacionais
- Drivers de dispositivo lidam com uma classe de dispositivos: um driver SCSI pode lidar com uma classe enorme de HDs
- No entanto, mouse e joystick são muito diferentes e múltiplos drivers são necessários

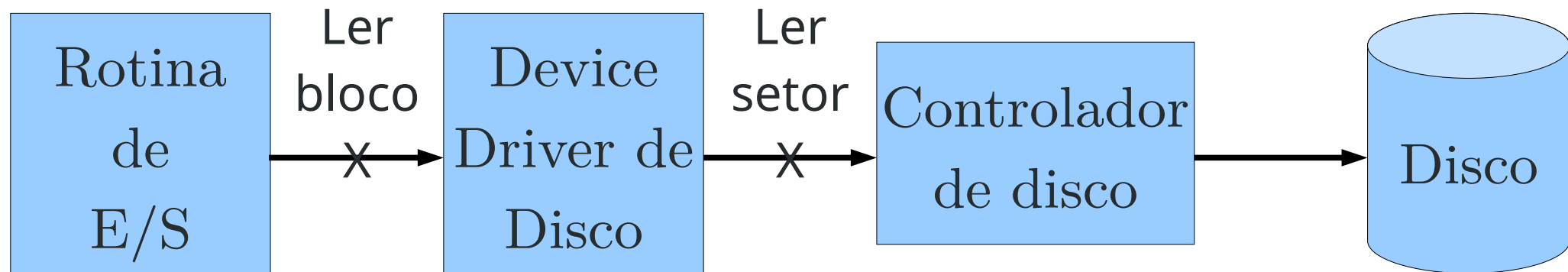


DEVICE DRIVER

- Para alguns drivers de dispositivos, faz sentido tratar múltiplos dispositivos, como os drivers USB
- Drivers USB estão disponíveis para discos, câmeras, pendrives, mouses, teclados, etc.
- Para lidar com essa situação, os drivers são empilhados no sistema operacional, assim como as camadas TCP/IP são empilhadas na rede



DEVICE DRIVER





DEVICE DRIVER

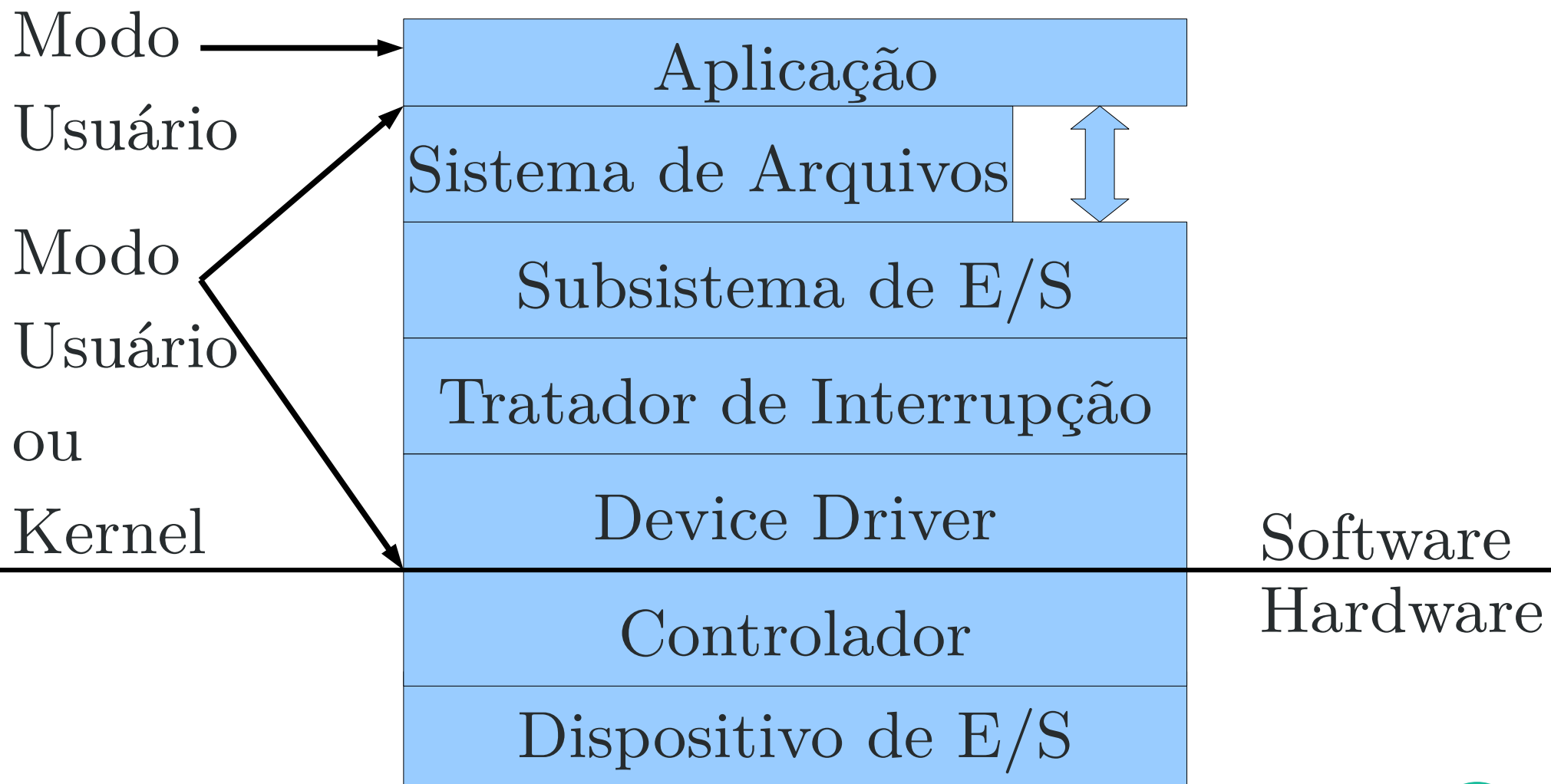
- Na maioria dos SOs, os device drivers fazem parte do núcleo: erros nesse trecho de código podem causar um kernel panic/tela azul
- Eles são escritos em linguagem de alta performance: **NÃO** são normalmente escritos em Assembly e dependem da interface do SO*
- Depende do sistema operacional e driver do dispositivo a necessidade de reinicialização e recompilação do kernel para adicionar o dispositivo ao kernel

Fonte:

<https://docs.microsoft.com/en-us/windows-hardware/drivers/kernel/writing-wdm-drivers>



CONTOLADOR DE E/S





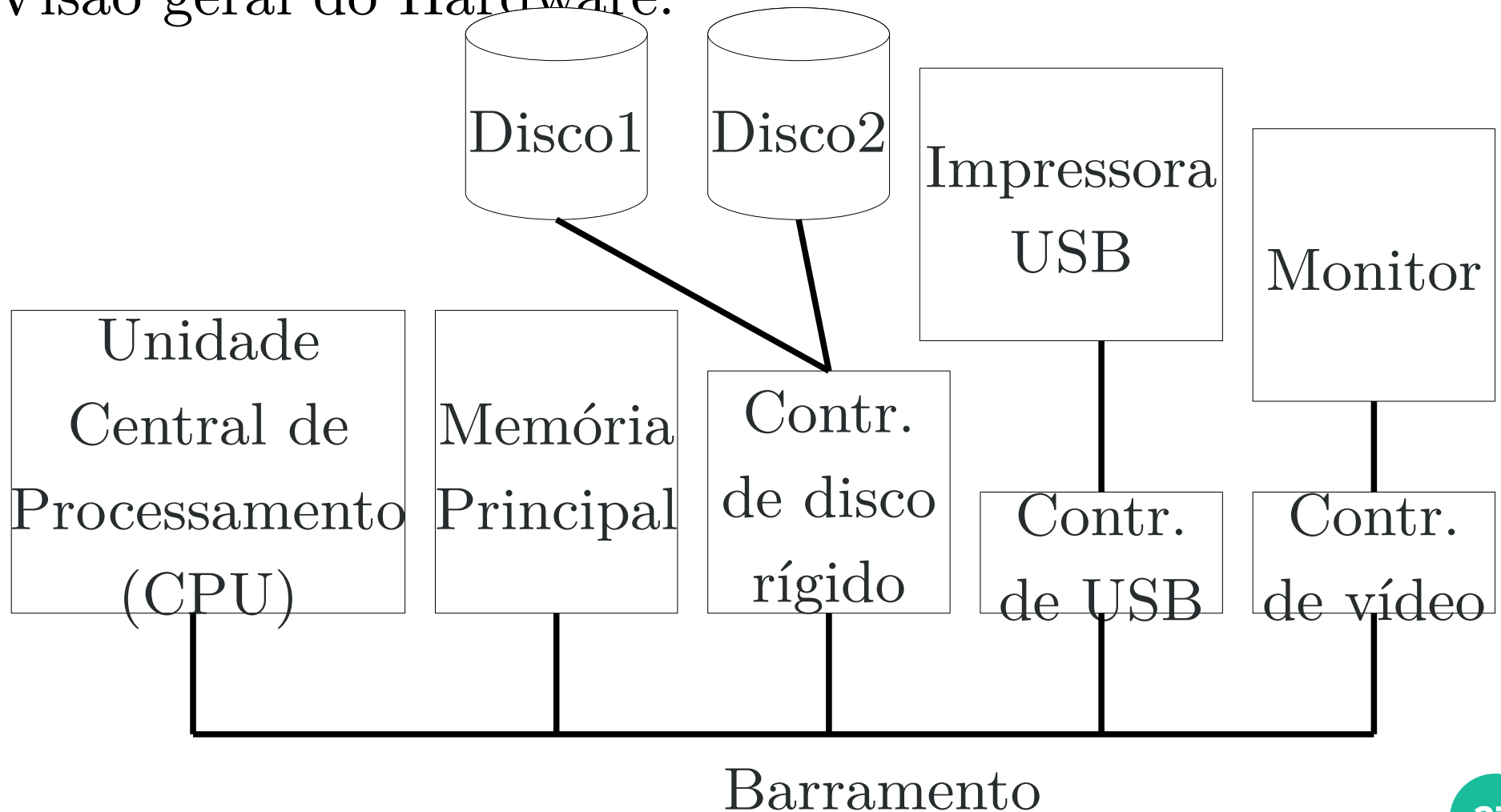
CONTROLADOR DE E/S

- Geralmente, os dispositivos de E/S são compostos por uma parte mecânica (o hardware) e uma parte eletrônica que controla o hardware
- O sistema operacional programa a controladora e não o dispositivo diretamente



CONTROLADOR DE E/S

- Visão geral do Hardware:





CONTROLADOR DE E/S

- Antigamente a CPU era conectada num único barramento compartilhado entre a memória principal e diversos dispositivos de E/S
- Devido a dificuldade em manter o controle direto dos dispositivos, foi criado um novo componente: o controlador de E/S
- Esse controlador de E/S normalmente está na forma de um chip na placa-mãe, em computadores pessoais



CONTROLADOR DE E/S

- A interface entre o controlador e o dispositivo pode ser padronizada, garantindo compatibilidade entre diferentes fabricantes
- Controladores de disco podem ser produzidos com interfaces SATA, SCSI, USB, Thunderbolt ou FireWire
- Controlador de E/S também é chamado de módulo de E/S



CONTROLADOR DE E/S

- O controlador possui memória e registradores próprios usados na execução de instruções enviadas pelo device driver
- A principal função do controlador de E/S é receber instruções da/para CPU e repassar para o/do o dispositivo
- Durante esse processo, os bytes que são enviados serialmente podem ser unidos em blocos e também pode-se verificar erros de transmissão



CONTROLADOR DE E/S

- Considere que o dispositivo em questão é um monitor LCD
- O monitor recebe caracteres a serem exibidos e gera sinais para modificar a polarização da retroiluminação dos pixels no monitor
- O controlador de tela nesse caso é fundamental para a simplificação do processo
- Se não fosse ele, o Sistema Operacional deveria calcular explicitamente o campo elétrico de todos os pixels em questão



PRINCIPAIS TÉCNICAS DE E/S

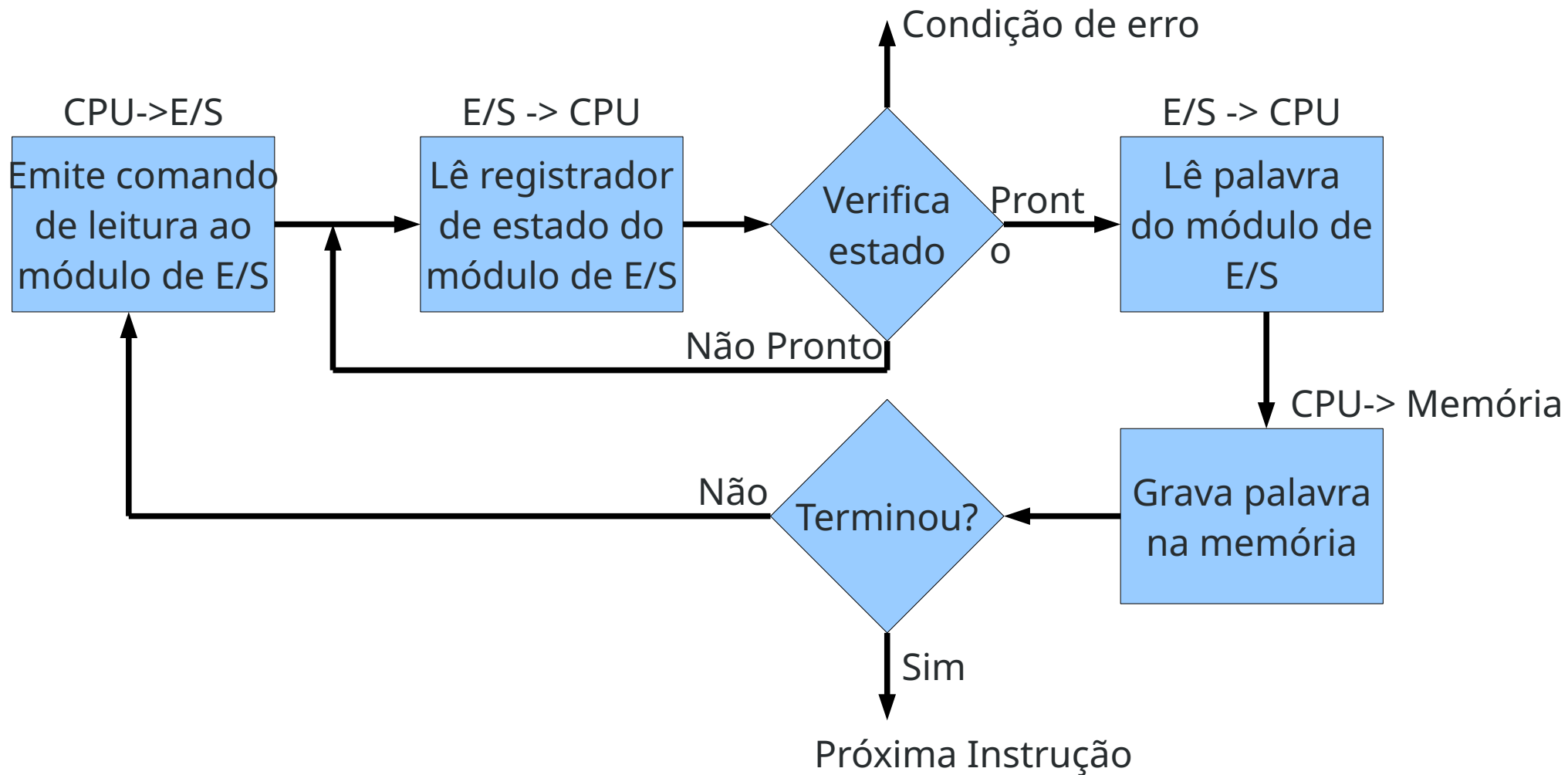
- A comunicação entre o processador e o módulo de E/S pode ser realizadas através de 3 possíveis operações:
 - E/S programada;
 - E/S controlada por interrupção;
 - Acesso direto à memória (DMA).



E/S PROGRAMADA

- Na técnica de E/S programada o processador é utilizado intensamente para realizar a operação de E/S.
- Durante o período da operação, a CPU fica a todo momento questionando se o dispositivo está pronto ou não para uma nova operação

E/S PROGRAMADA





E/S DIRIGIDA POR INTERRUPÇÃO

- A grande desvantagem da E/S programa é utilizar o tempo do processador para verificar o dispositivo
- Um método de interrupção gera uma linha de controle adicional, de forma que o processador pode executar outras funções. Quando o dispositivo terminar, ele gera uma interrupção para o processador



E/S DIRIGIDA POR INTERRUPÇÃO

- Quando um processador recebe uma interrupção, ele realiza a cópia do módulo de E/S para a memória e posteriormente retoma o processamento que estava executando
- Isto é possível, pois o hardware possui funções de troca de contexto, que permite salvar os registradores da CPU para restaurar posteriormente



E/S DIRIGIDA POR INTERRUPÇÃO

- Comunicação realizada da seguinte forma:
- O processador emite uma instrução para o módulo de E/S
- Como a resposta não é rápida, ele espera o periférico ficar pronto e atende outro programa em execução

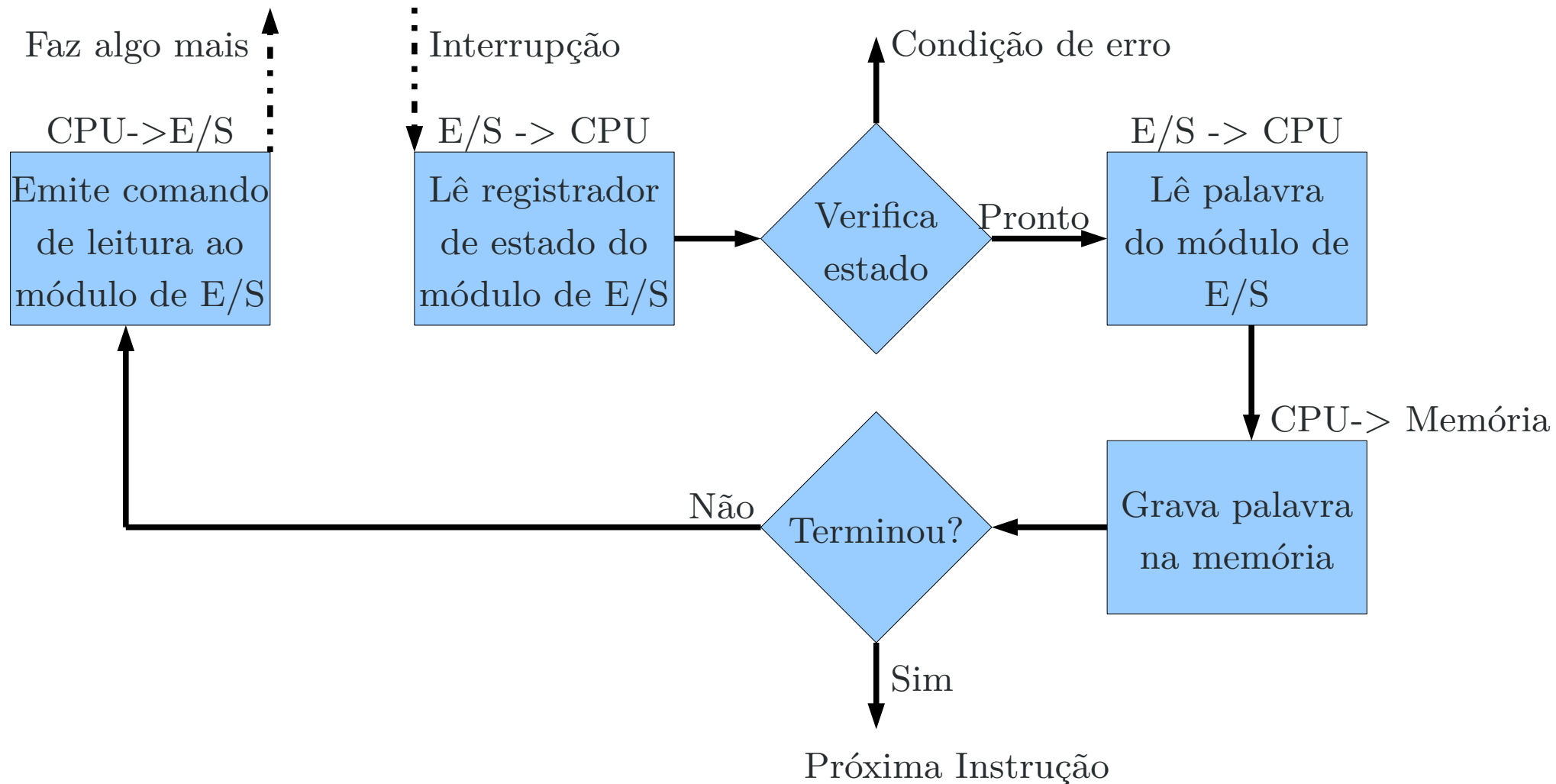


E/S DIRIGIDA POR INTERRUPÇÃO

- Quando o periférico está pronto, seu módulo de E/S avisa o processador por um sinal de interrupção, o processador para o que está fazendo e atende ao dispositivo que o chamou
- Finalmente, o processador inicia a transferência de dados do módulo de E/S com a memória, etc.



E/S DIRIGIDA POR INTERRUPÇÃO





ACESSO DIRETO À MEMÓRIA (DMA)

- A E/S controlada por interrupção ainda exige a intervenção ativa do processador para realizar a transferência de dados entre a memória e o módulo de E/S
- Inviável quando se trata de um grande volume de dados a serem movidos
- A taxa de transferência é limitada pela velocidade que o processador é capaz de testar e atender
- Diversas instruções precisam ser executadas para uma única operação de E/S



ACESSO DIRETO À MEMÓRIA (DMA)

- A melhor alternativa para realizar operações de E/S com melhor rendimento para o processador é a DMA
- Transferência de dados entre uma determinada interface e a memória principal praticamente sem a intervenção do processador



ACESSO DIRETO À MEMÓRIA (DMA)

- O módulo de E/S usa o barramento de memória quando não está sendo usado pelo processador
- Ou força o processador a não usar o barramento (roubo de ciclo)
- O módulo de DMA é capaz de imitar um processador no que diz respeito a transferência de dados para a memória



ACESSO DIRETO À MEMÓRIA (DMA)

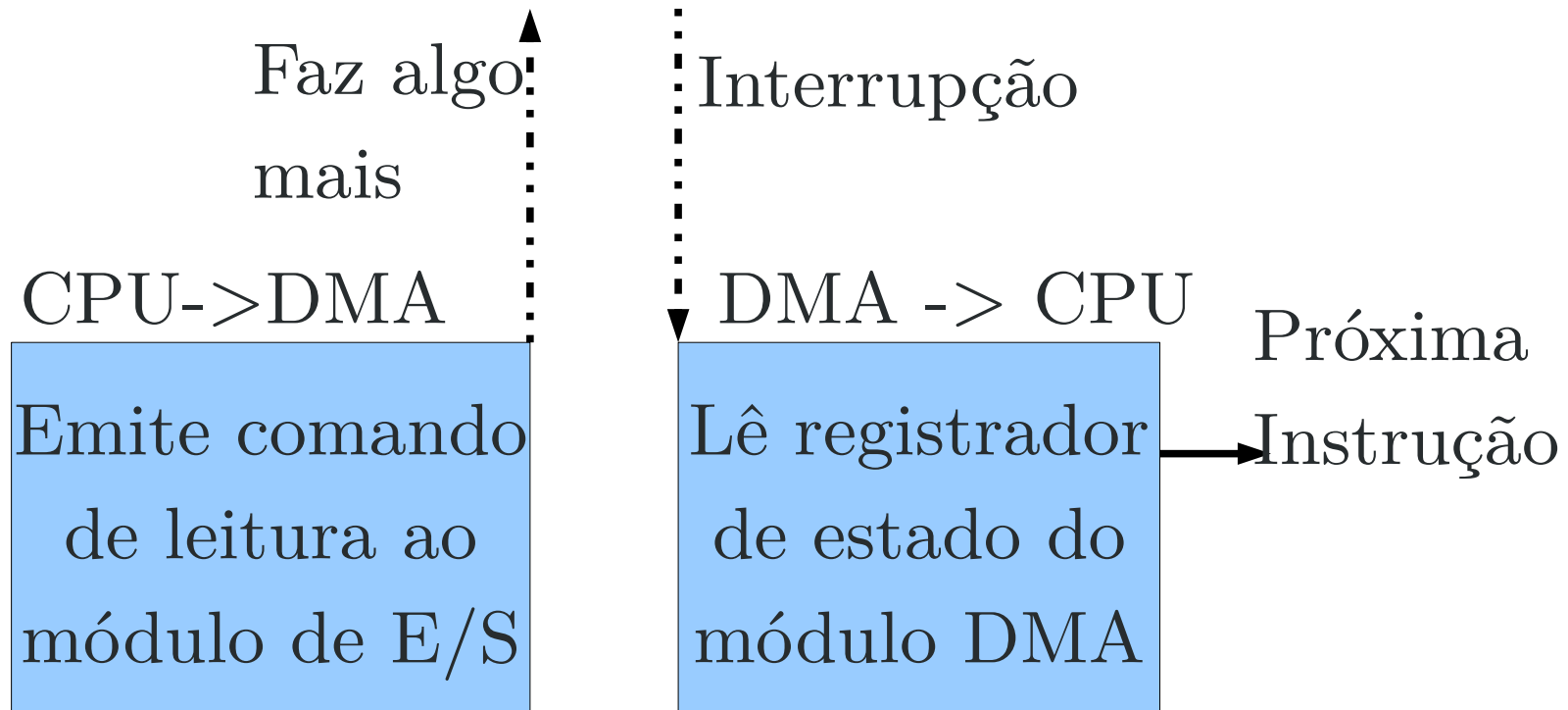
- Para implementar o DMA, o processador envia um comando com as seguintes informações:
 - Indicação de uma operação de leitura ou escrita, usando a linha de controle
 - O endereço do dispositivo de E/S envolvido
 - O endereço de memória para início da leitura ou escrita de dados
 - Número de palavras a serem lidas ou escritas.



ACESSO DIRETO À MEMÓRIA (DMA)

- Depois disso o processador continua o trabalho que quiser executar
- O módulo de DMA transfere o bloco de dados diretamente de/para a memória sem passar pelo processador
- Quando a transferência termina, o módulo de DMA envia um sinal de interrupção ao processador
- Assim o processador é envolvido apenas no início e final de cada transferência

ACESSO DIRETO À MEMÓRIA (DMA)





CONTROLADORES DE E/S

- Alguns controladores de E/S implementam um sistema de cache, semelhante a técnica de buffering implementada no sistema de arquivos
- Busca otimizar as operações de E/S
- Dados são salvos em uma memória temporária do dispositivo e criam a ilusão que as operações terminam rapidamente



TRATADOR DE INTERRUPÇÃO

- Processo de tratamento de interrupção:
- Salvar quais registros que ainda não foram salvos
- Estabelecer um contexto para a rotina de tratamento de interrupção
- Sinalizador o controlador de interrupção
- Copiar os registradores de onde eles foram salvos (pilha no hardware) para a tabela de processos
- Executar a rotina de tratamento de interrupções, utilizando as informações dada pelo controlador

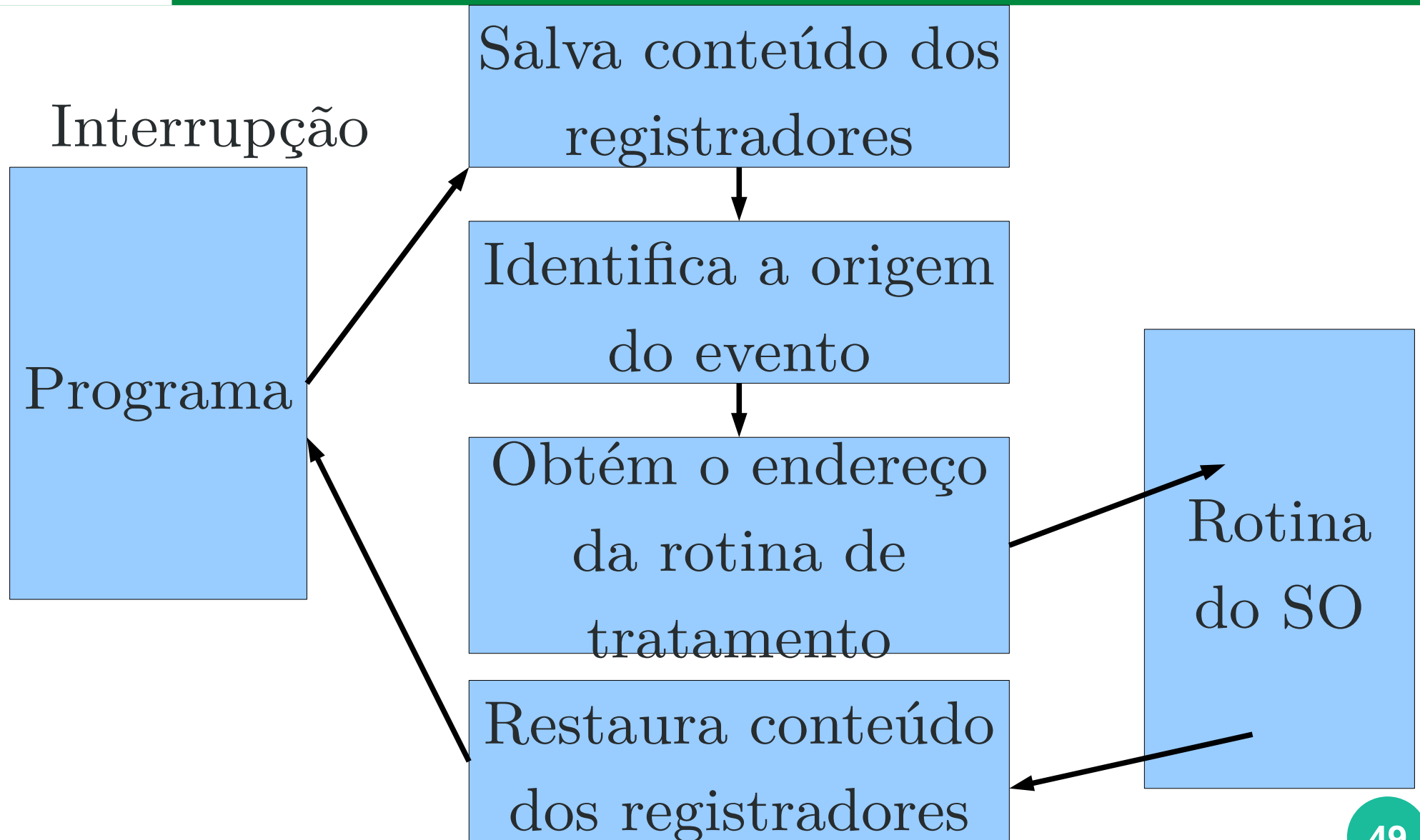


TRATADOR DE INTERRUPÇÃO

- Ao finalizar a execução da interrupção, faça o desbloqueio do driver do dispositivo
- Execute o algoritmo de escalonamento, para escolher algum processo para executar
 - Não será executado necessariamente o processo que foi interrompido
- Estabelecer o contexto do processo que está executando: contexto da MMU, alterações na TBL, carregar registradores do novo processo
- Iniciar a execução do novo processo



TRATADOR DE INTERRUPÇÃO





TRATADOR DE INTERRUPÇÃO

- Há um problema sério quando múltiplas interrupções são geradas simultaneamente
- Primeiro, a rotina de tratamento de interrupções precisa ser reentrante: deve ser seguro executá-la uma segunda vez, mesmo que ainda não terminou de executar a primeira
- Um driver de rede pode estar no processamento de um pacote de rede e outro pacote chega



TRATADOR DE INTERRUPÇÃO

- O processo de tratamento de interrupções não é trivial em um sistema operacional
- As interrupções precisam afetar a menor parte possível do SO e devem ser muito pequenas, deixando funções complexas para outras camadas
- Em sistemas multiprocessados, elas devem bloquear utilizando um semáforo ou mutex
- Após o tratamento da interrupção o driver é desbloqueado



SCSI

- O padrão mais popular para a conexão de dispositivos a um computador é o SCSI
- Define padrões de hardware e software que podem conectar ao sistema computacional dispositivos de diversos fabricantes
- O SO possui um driver SCSI que é responsável por realizar as comunicações com um controlador SCSI



DISCOS MAGNÉTICOS

- Os discos magnéticos (HDs) são amplamente utilizados e estão sendo parcialmente substituídos pelos discos de estado sólidos (SSDs)
- Um disco é constituído por vários discos sobrepostos, unidos por um eixo vertical e girando a uma velocidade constante

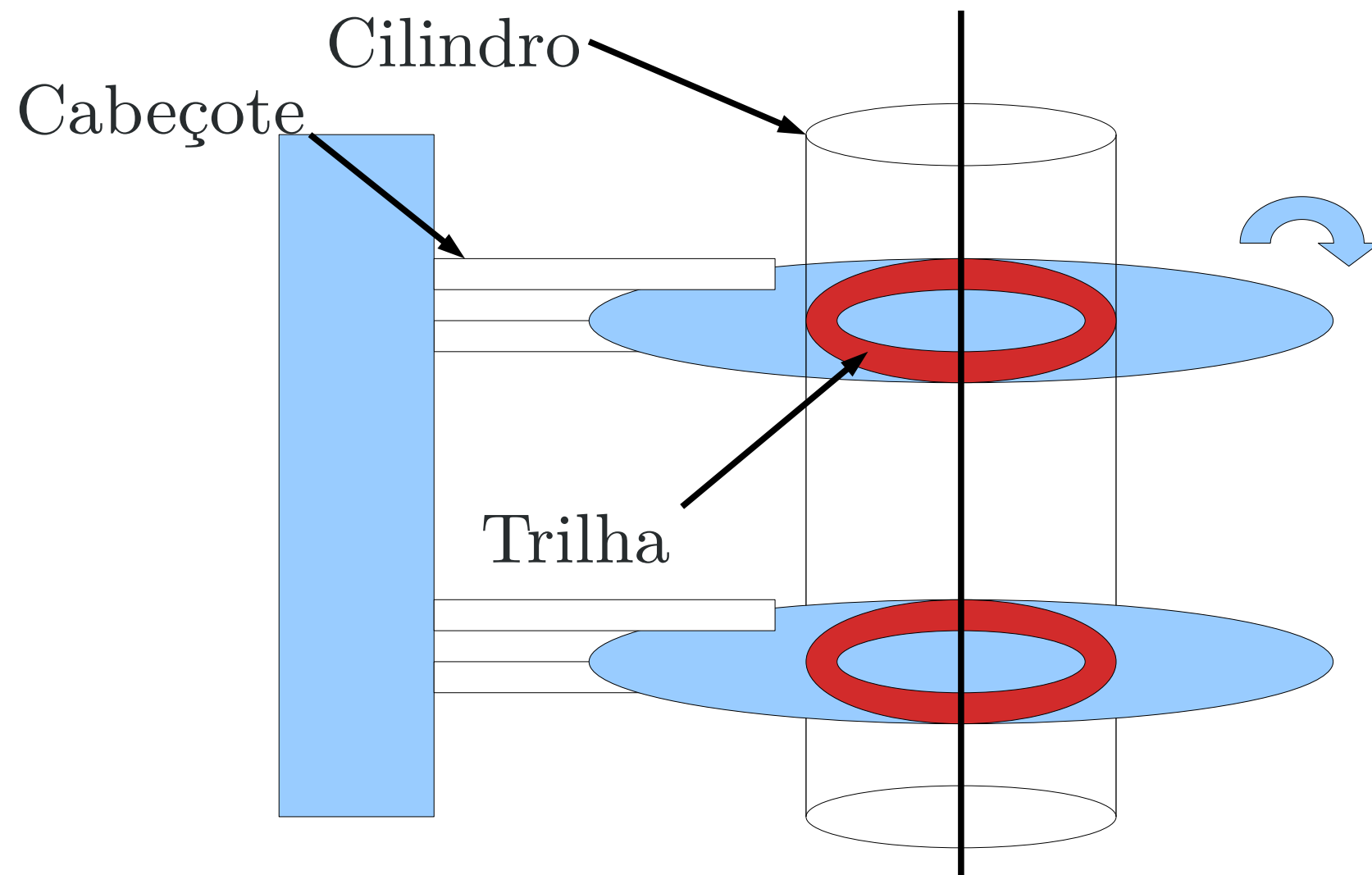


DISCOS MAGNÉTICOS

- Todos os discos possuem uma determinada capacidade de rotação
- Todos os pratos giram numa velocidade constante
- Velocidades típicas de rotação: 5400 RPM para notebooks e 7200 RPM em desktop
- Já foram construídos HDs com até 15000 RPM



DISCOS MAGNÉTICOS





DISCOS MAGNÉTICOS

- Tempo de acesso: tempo gasto do início ao término da leitura ou escrita. Divido em 4 fases:
 - Tempo de interpretação do comando: Gasto pelo SO para interpretar o comando e passar ao controlador de disco
 - Tempo de busca: Tempo gasto para realizar o movimento do braço
 - Tempo de latência: Tempo gasto pela passagem do setor desejado sobre a cabeça (vel. rotação)
 - Tempo de transferência: tempo gasto para os sinais elétricos chegarem ao destino



EVOLUÇÃO EM 2 DÉCADAS

Parâmetro	Disco Flexível IBM 360 KB	Disco Rígido WD 18300
Número de Cilindros	40	10 601
Trilhas por Cilindro	2	12
Setores por Trilha	9	281 (média)
Setores por Disco	720	35 742 000
Bytes por Setor	512	512
Capacidade do Disco	360 KB	18,3 GB
Tempo de posicionamento	6 ms	0,8 ms
Tempo médio de posicionamento	77 ms	6,9 ms
Tempo de rotação	200 ms	8,33 ms
Tempo de pára/inicia do motor	250 ms	20 s



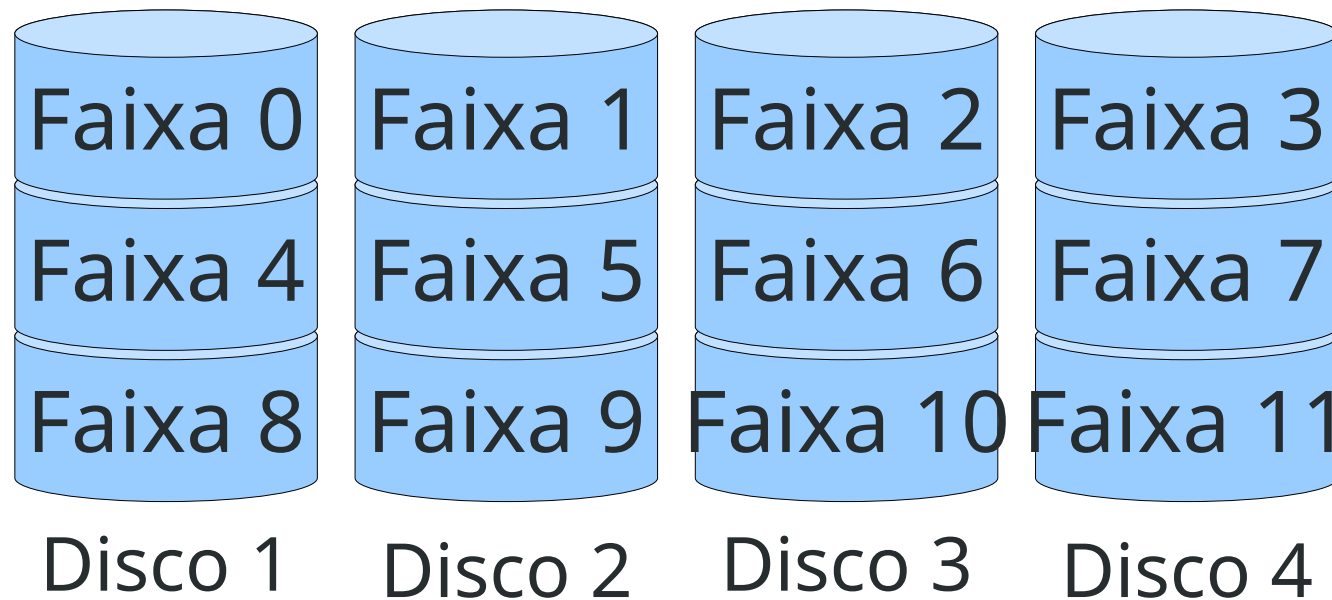
RAID

- O desempenho da CPU sendo superior aos dos discos, levou a uma motivação por novas técnicas de aceleração dos discos
- Patterson et al, propuseram em 1988 uma nova classe de dispositivos chamados de RAID
- Redundant Array of Inexpensive Discs
- Arranjo redundante de discos Baratos
 - A indústria renomeou o Inexpensive para Independent



RAID 0

- O RAID 0 (stripping) consiste em ver um disco virtual simulado pelo RAID dividido em faixas com uma quantidade igual de setores





RAID 0

- No RAID 0 os discos são divididos em pedaços
- As operações de escrita de múltiplas faixas podem ocorrer paralelamente entre os discos
- As operações de leitura também ocorrem paralelamente
- Assim as operações em faixas consecutivas ocorrem simultaneamente, sem que o software tenha conhecimento disto



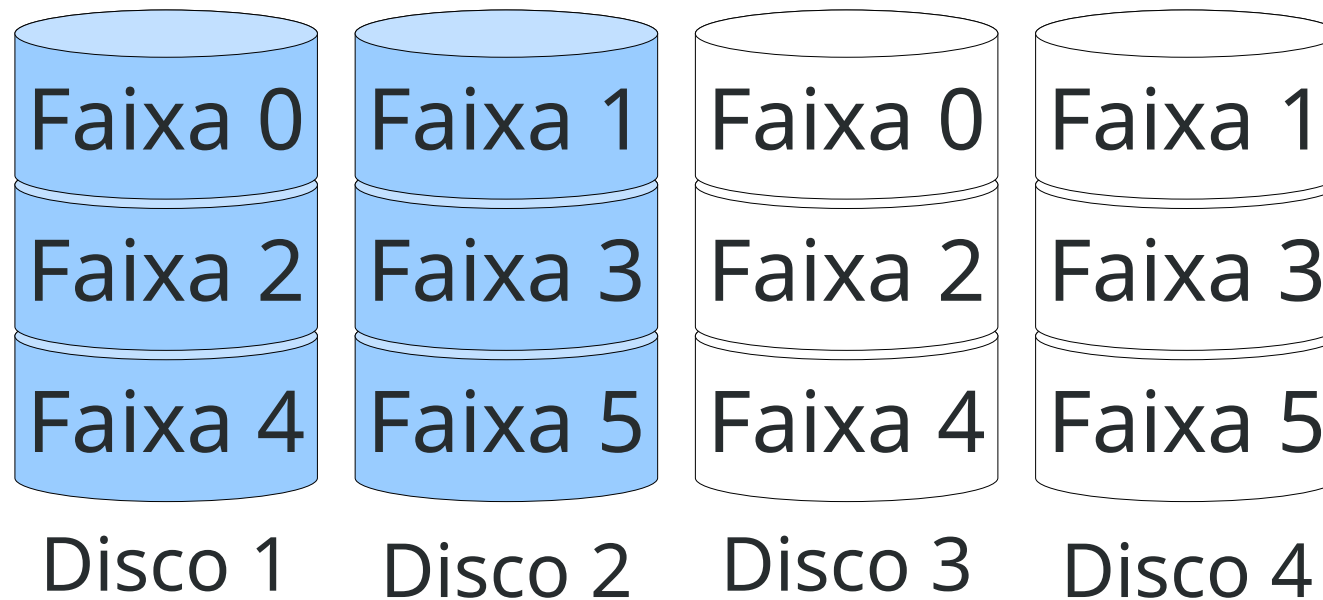
RAID 0

- Para obter esse desempenho, é melhor que as faixas consecutivas sejam solicitadas
- No RAID 0 não há redundância entre as informações: a perda de um disco leva a perda de todas as informações em todos os discos



RAID 1

- O RAID 1 duplica as informações existentes em todos os setores, para garantir a confiança no armazenamento





RAID 1

- Em uma operação de escrita, as faixas são escritas simultaneamente nos discos
 - Não há ganho de velocidade neste caso.
- Nas operações de leitura, as faixas podem ser lidas paralelamente
 - Velocidade de leitura similar ao RAID 0 neste caso
- Excelente tolerância a falhas: se um disco falhar, as informações estão armazenadas no outro disco e podem ser facilmente recuperadas



PARIDADE

- No entanto, o RAID 1 leva a duplicação completa da informação
- Existem diversos algoritmos baseados na ideia de paridade
- Um dos mais simples, é a operação lógica “ou exclusivo” (XOR)



PARIDADE

Obviamente, é possível preencher os valores corretos da última coluna a partir das outras duas

A	B	A XOR B
0	0	
0	1	
1	0	
1	1	



PARIDADE

Obviamente, é possível preencher os valores corretos da última coluna a partir das outras duas

A	B	A XOR B
0	0	0
0	1	1
1	0	1
1	1	0



PARIDADE

No entanto, é possível obter os valores de qualquer coluna, a partir das outras duas, afinal

$$A \text{ XOR } (A \text{ XOR } B) = B$$

A	B	A XOR B
0	0	0
0	1	1
1	0	1
1	1	0



PARIDADE

No entanto, é possível obter os valores de qualquer coluna, a partir das outras duas, afinal

$$A \text{ XOR } (A \text{ XOR } B) = B$$

$$0 \text{ XOR } 0 = 0$$

A	B	A XOR B
0	0	0
0	1	1
1	0	1
1	1	0



PARIDADE

No entanto, é possível obter os valores de qualquer coluna, a partir das outras duas, afinal

$$A \text{ XOR } (A \text{ XOR } B) = B$$

$$0 \text{ XOR } 1 = 1$$

A	B	A XOR B
0	0	0
0	1	1
1		1
1		0



PARIDADE

No entanto, é possível obter os valores de qualquer coluna, a partir das outras duas, afinal

$$A \text{ XOR } (A \text{ XOR } B) = B$$

$$1 \text{ XOR } 1 = 0$$

A	B	A XOR B
0	0	0
0	1	1
1	0	1
1	1	0



PARIDADE

No entanto, é possível obter os valores de qualquer coluna, a partir das outras duas, afinal

$$A \text{ XOR } (A \text{ XOR } B) = B$$

$$1 \text{ XOR } 0 = 1$$

A	B	A XOR B
0	0	0
0	1	1
1	0	1
1	1	0



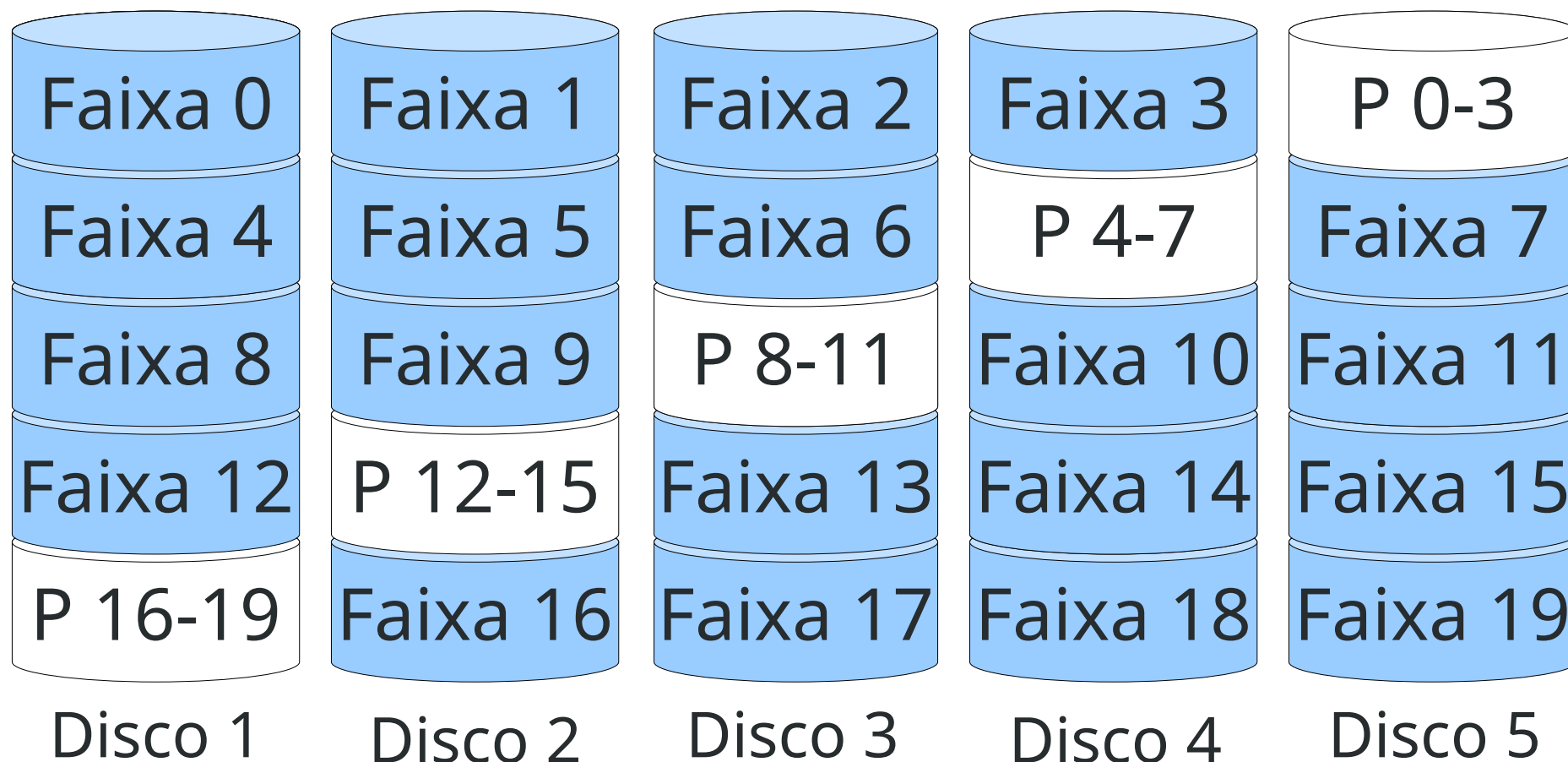
RAID 5

- O RAID 5 busca aumentar a confiabilidade dos arranjos de discos, mas evitam o desperdício de espaço do RAID 1: usam um código de paridade ao invés de duplicar a palavra
- O código de paridade garante a recuperação da faixa de um disco, caso algum deles falhe



RAID 5

- O RAID 5 com faixas e paridades:





RAID 5

- Ao se modificar uma faixa, todas as outras precisam ser lidas para calcular a paridade do disco
- Concentrar todas as operações de paridade em um único disco podem fazê-lo tornar-se um gargalo
- Por isso o RAID 5 distribui as paridades entre os diversos discos disponíveis no arranjo



RAID

- Existem outros níveis de RAID:
- RAID 2 e 3 trabalham com palavras ao invés de faixas. Exigem discos com rotações sincronizadas e muito esforço do controlador.
 - O RAID 3 é uma versão simplificada do 2: no 3 apenas um bit de paridade é utilizado, enquanto no 2 múltiplos são



RAID

- O RAID 4 é muito similar ao 5, porém concentra todos os bits de paridade em um único HD, o que pode ser um problema
- O RAID 6 também é similar ao 5, porém utiliza múltiplas faixas de paridade. Por exemplo: armazena-se as faixas F0 no Disco 1, F1 no Disco 2 e F2 no Disco 3. A paridade P 0-2 é armazenada nos Discos 2 e 3
 - Permite uma maior confiabilidade e desempenho de leitura de paridade, maior desperdício de espaço



CLOCKS

- Os relógios (clocks, temporizadores ou timers) são essenciais para a operação de qualquer sistema multiprogramado
- Eles mantêm o dia e a hora do computador e evitam que um processo monopolize a CPU
- Eles combinam funções de hardware e software



CLOCKS

- Os relógios são construídos por um oscilador de cristal, um contador e um registrador de apoio
- O cristal cortado corretamente e colocado sobre uma determinada tensão, gera um sinal periódico de alta precisão
- Pelo menos um circuito desse tipo é encontrado em todos os computadores modernos



CLOCKS

- Esse clock pode ser programável: um registrador armazena um valor a cada oscilação do cristal
um registrador é decrementado
- Ao se atingir 0 o hardware gera uma interrupção



CLOCKS

- Para não perder a hora quando desligado, a maioria dos computadores possui um relógio de backup, mantido por uma bateria como os relógios digitais
- Caso o tempo esteja errado, o sistema pede ao usuário para atualizar
- Existem protocolos de rede que permite a atualização da hora: NTP



CLOCKS

- Já a parte de software dos Clocks, implementa diversas funções que são utilizadas pelos SOs:
 - Obter o tempo real através de chamadas de sistema;
 - Implementar o quantum do escalonador de processos;
 - Contabilizar o tempo de uso do processador;
 - Gerenciar chamadas ALARM: onde um sinal é gerado a um processo depois de um período de tempo;
 - Entre outros.



CLOCKS

- Já o relógio interno utiliza outro registrador de apoio, que armazena o número de ticks do relógio desde as 12h de 1º de janeiro de 1970 (epoch time). Conhecido como Tempo Universal Coordenado (UTC)
- Desta forma, são usados 31 bits para representar o tempo, como o número de segundos



O PROBLEMA DE 2038

- No dia 19 de janeiro de 2038, às 03:14:07, é o último segundo capaz de ser representado utilizando um inteiro de 32 bits.
- Após isso, o tempo será representado como de volta ao século 20

Binary : 01111111 11111111 11111111 11110000

Decimal : 2147483632

Date : 2038-01-19 03:13:52 (UTC)

Date : 2038-01-19 03:13:52 (UTC)

- Qual seria a solução para esse problema?



O PROBLEMA DE 2038

- Uma solução seria utilizar 64 bits para representar o tempo
- Desta forma, apenas processadores 32 bits não iriam funcionar
- Vocês acham que é uma solução permanente?



O PROBLEMA DE 2038

- Lembre-se que ao adicionar um bit, você duplica a capacidade. Ao adicionar 2 bits, a capacidade é quadruplicada
- Como é exponencial, adicionar 32 bits permite que o tempo seja marcado até 290 bilhões de anos
- A idade do universo não é nem 14 bilhões de anos...



REFERÊNCIAS

- Capítulo 5 – TANENBAUM, A. S. *Sistemas Operacionais Modernos*. 4ª ed. São Paulo: Pearson Education do Brasil, 2016.
- Capítulo 12 – MACHADO, F. B.; MAIA, L. P. *Arquitetura de Sistemas Operacionais*. 5ª ed. Rio de Janeiro: LTC, 2013.
- **Complementar:** Capítulo 7 – STALLINGS, W. *Arquitetura e organização de computadores*. 10ª ed. São Paulo: Pearson Education do Brasil, 2017.