Sistemas Operacionais

Entrada e Saída

Norton Trevisan Roman
Marcelo Morandini
Jó Ueyama

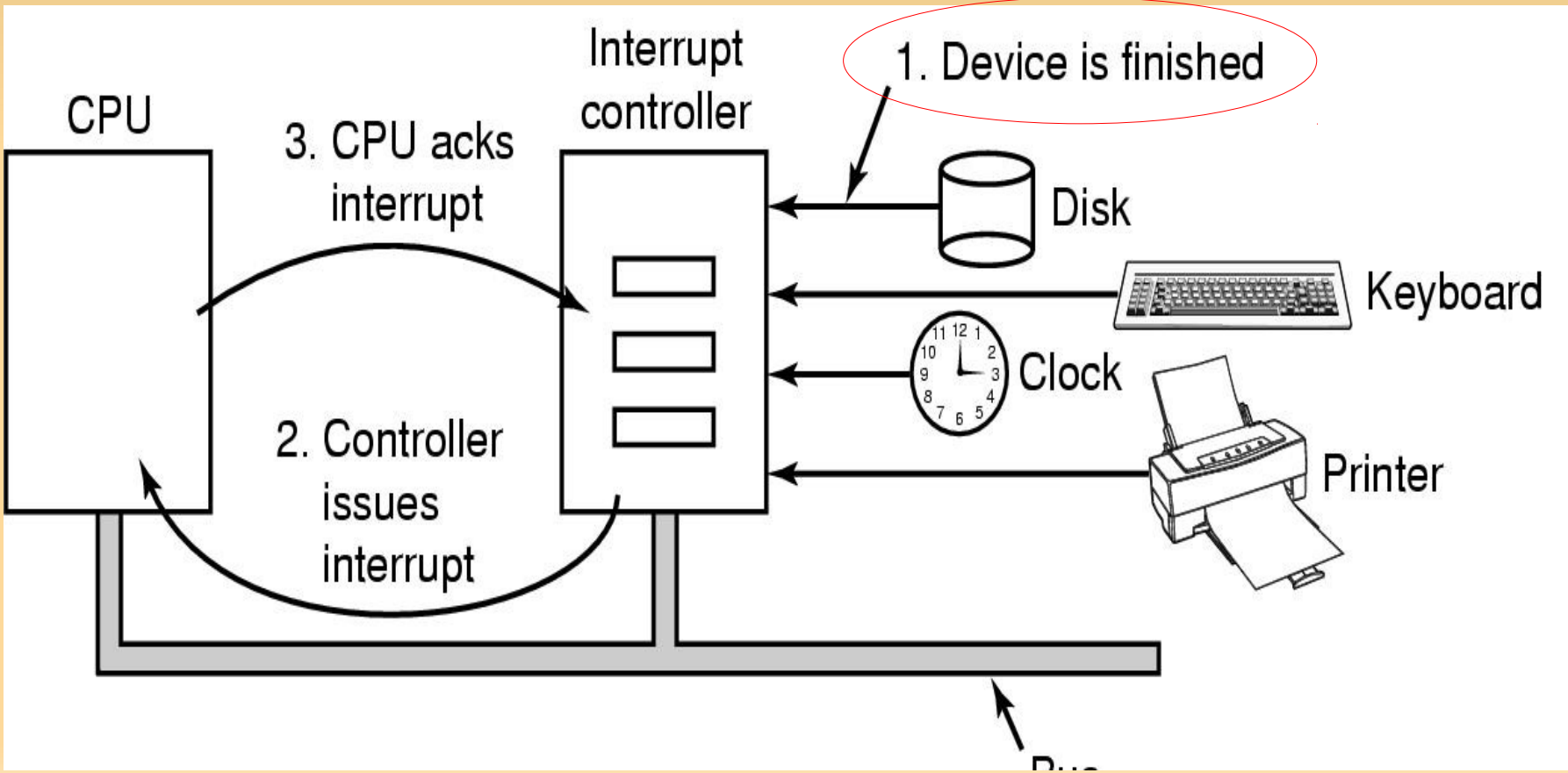
Apostila baseada nos trabalhos de Kalinka Castelo Branco, Antônio Carlos Sementille, Luciana A. F. Martimiano e nas transparências fornecidas no site de compra do livro "Sistemas Operacionais Modernos"

Interrupção Revista

- Funcionamento

Ao terminar seu trabalho, o dispositivo causa uma interrupção;

Faz isso mandando um sinal em uma linha do barramento

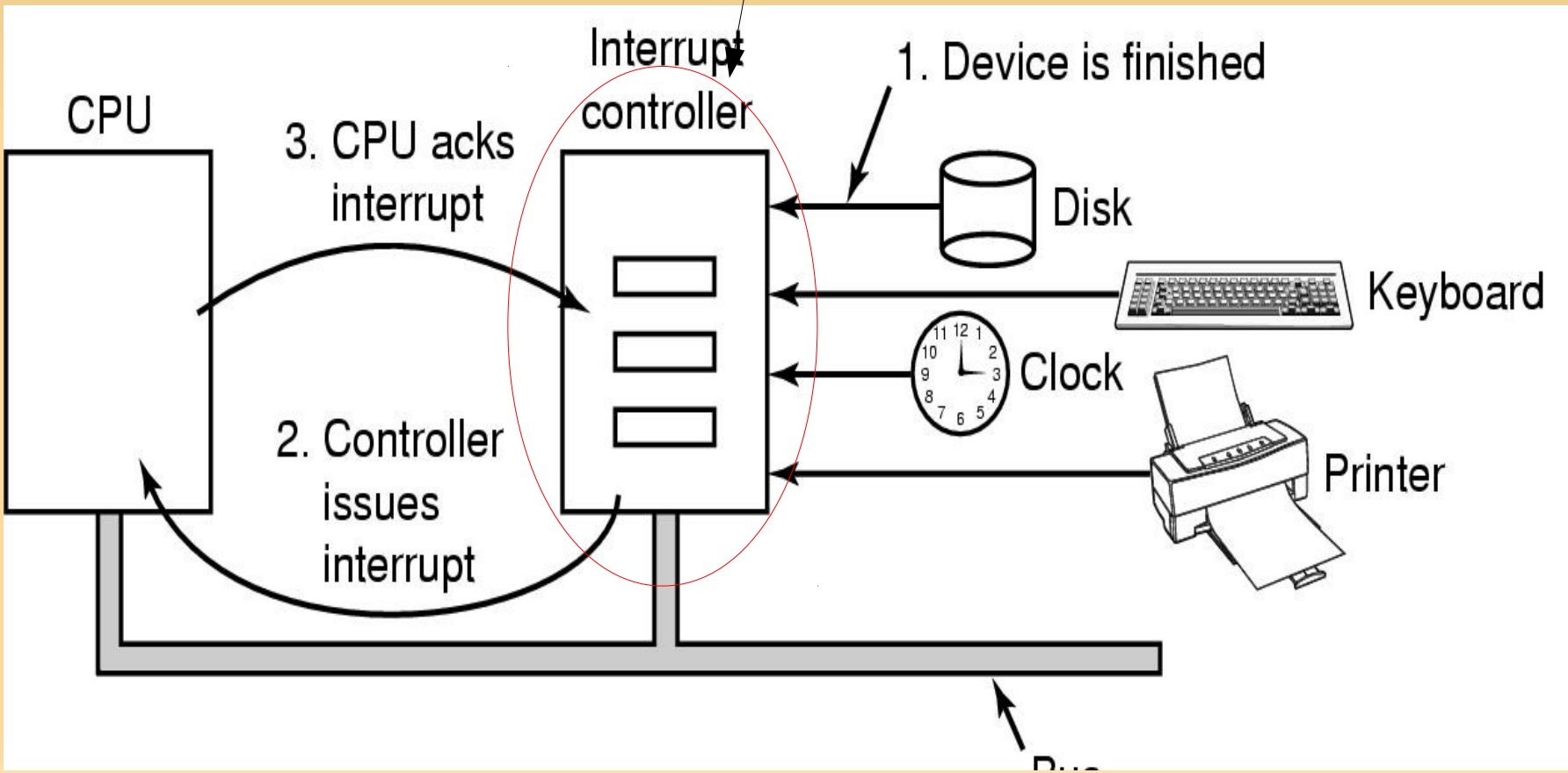


Interrupção Revista

- Funcionamento

O sinal é detectado pelo controlador de interrupções, que decide o que fazer

Se nenhuma outra interrupção estiver pendente, o controle processa a interrupção imediatamente

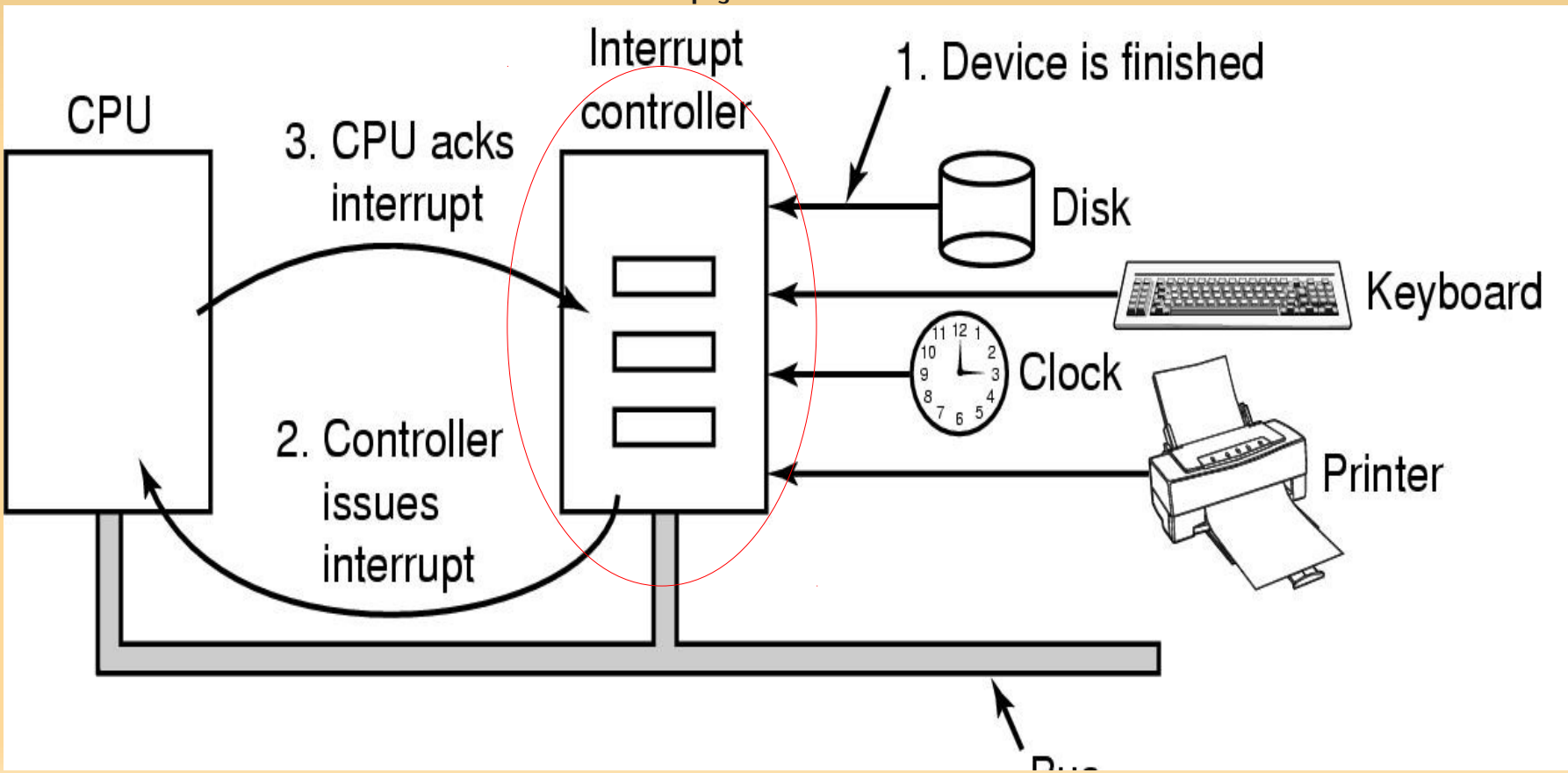


Interrupção Revista

- Funcionamento

Se alguma outra estiver em progresso, ou outro dispositivo fez um pedido simultâneo, em uma linha de interrupção de maior prioridade no barramento, o dispositivo é ignorado.

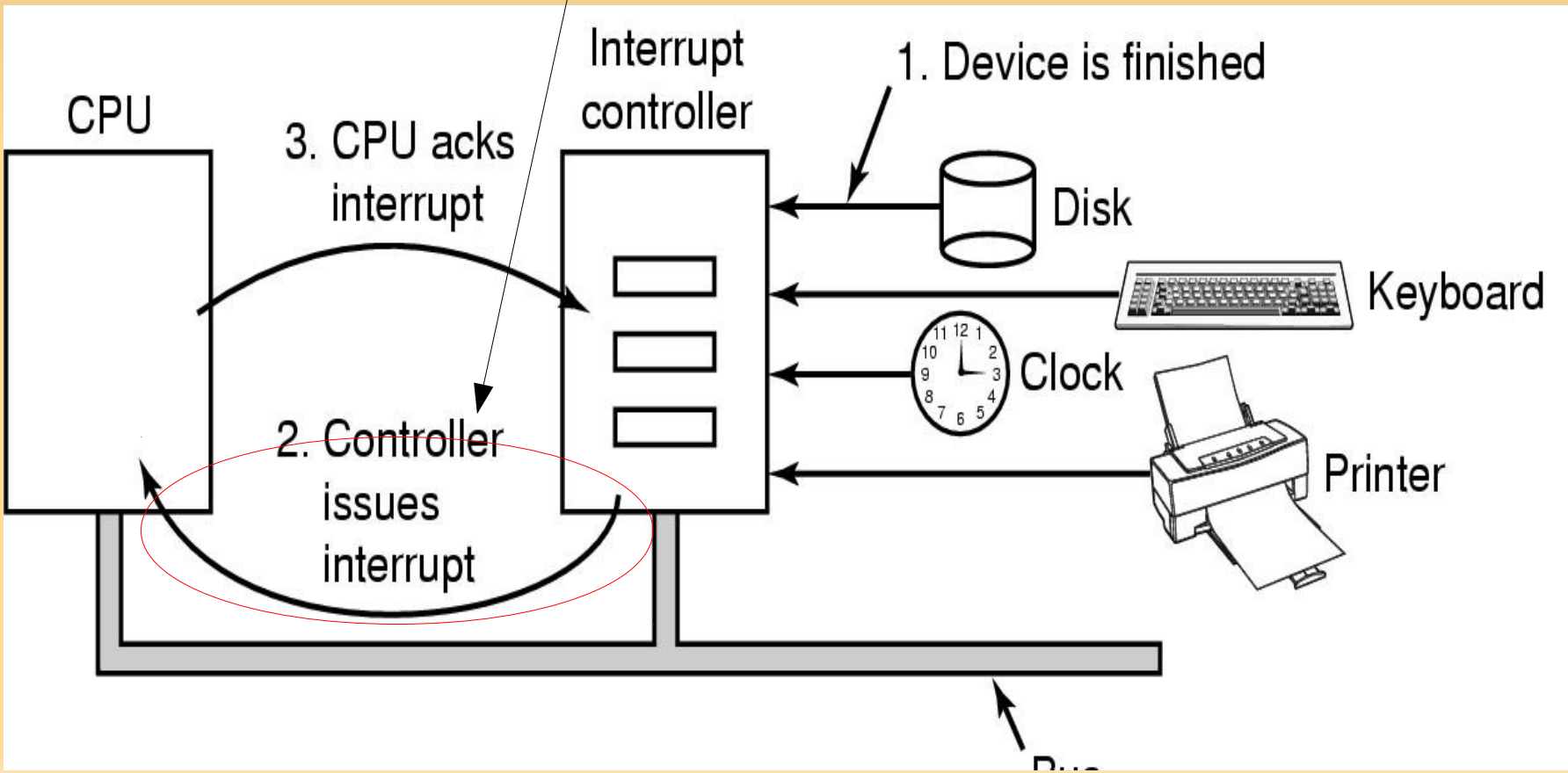
Caso em que continua a mandar o sinal de interrupção ao barramento



Interrupção Revista

- Funcionamento

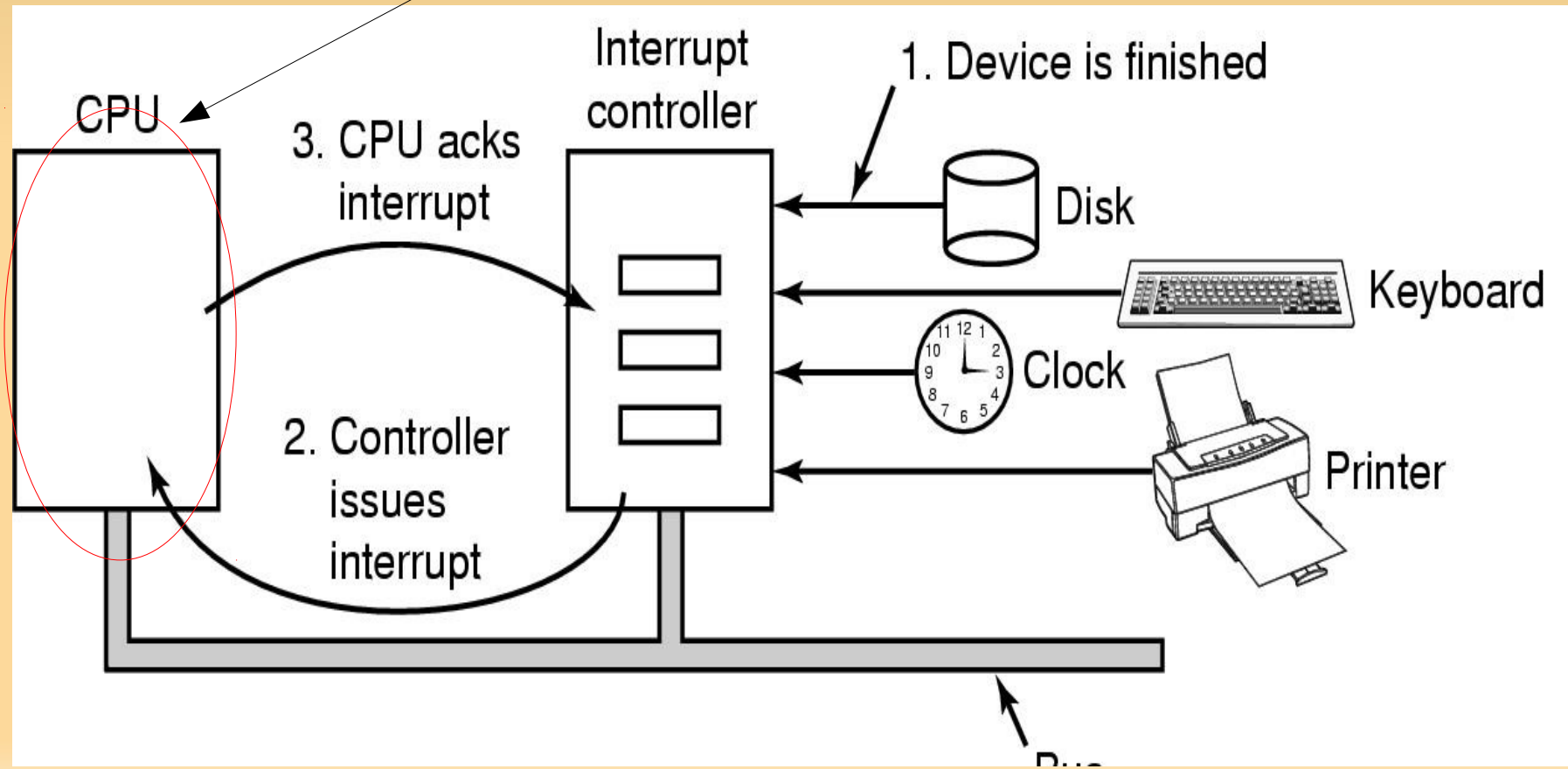
Para tratar a interrupção, o controlador coloca um número, especificando o dispositivo, nas linhas de endereço do barramento, mandando um sinal para interromper a CPU



Interrupção Revista

- Funcionamento

O sinal faz com que a CPU pare o que está fazendo. Ela então usa o número nas linhas de endereço como índice no vetor de interrupções, obtendo um novo contador de programa



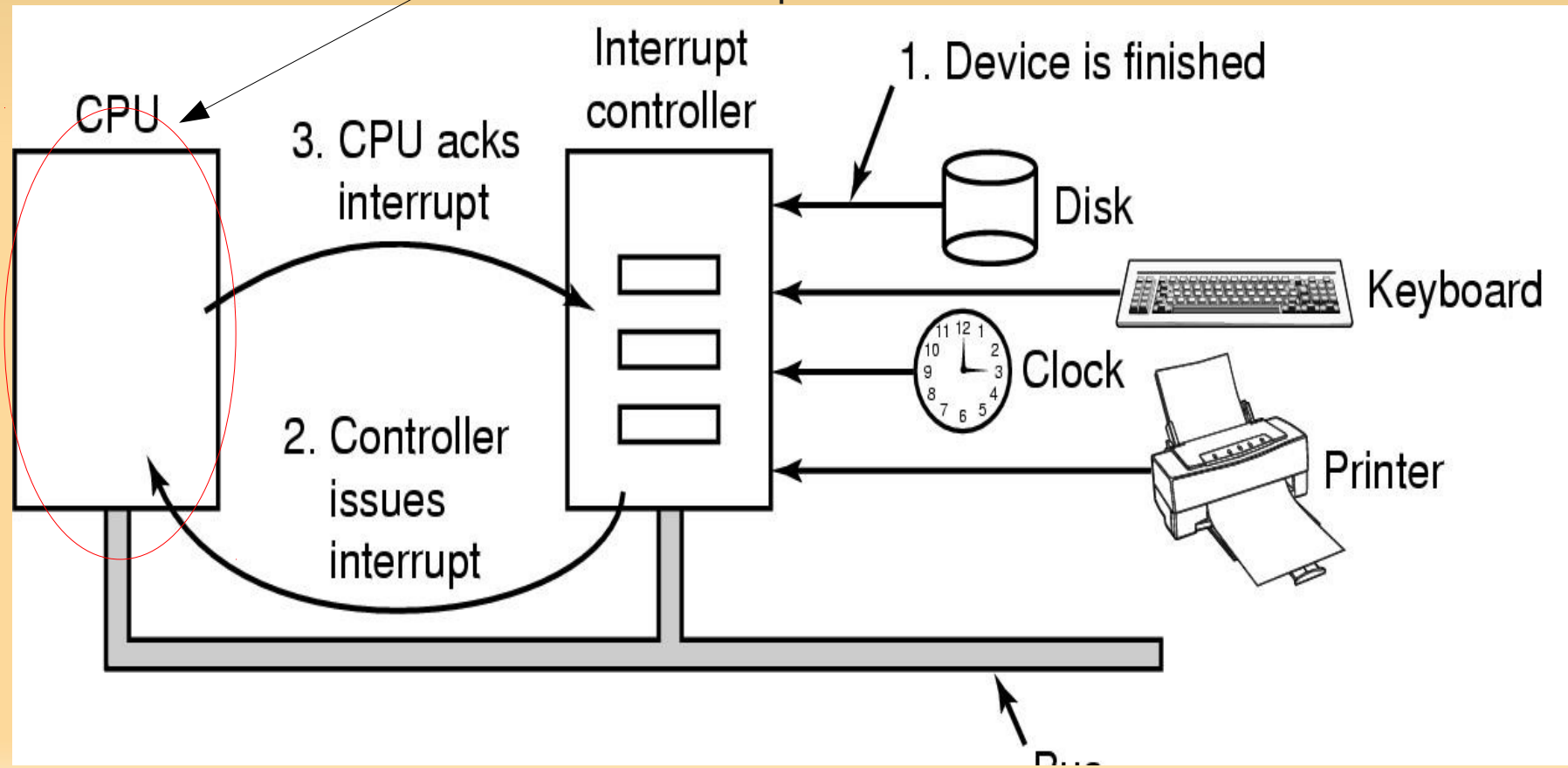
Interrupção Revista

- Funcionamento

O novo contador de programa aponta para o início do procedimento de tratamento da interrupção.

Traps e interrupções usam o mesmo mecanismo a partir desse ponto

O vetor de interrupções pode estar tanto no hardware quanto na memória

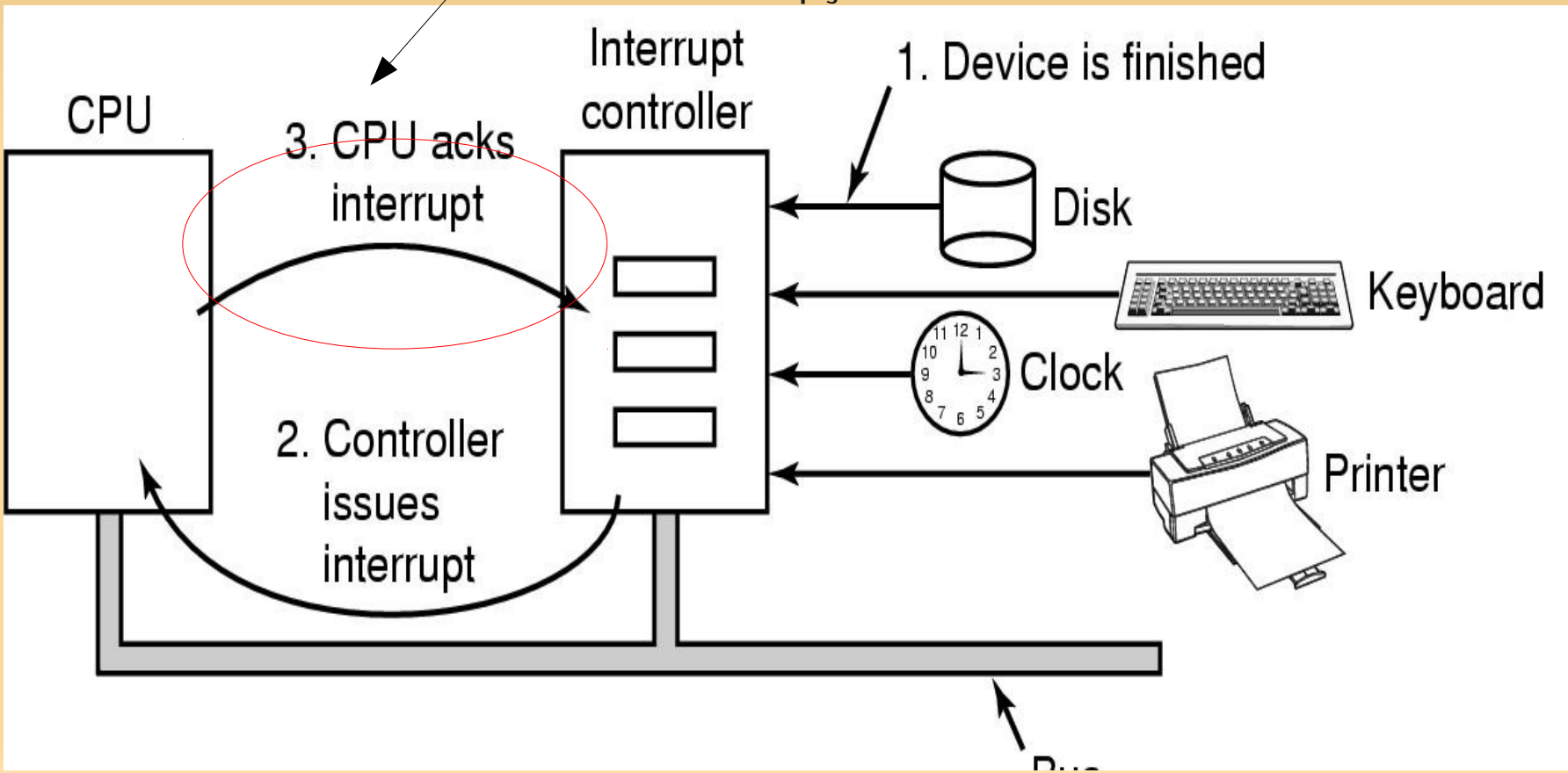


Interrupção Revista

- Funcionamento

Logo após começar a rodar, o procedimento acusa recebimento da interrupção, escrevendo um certo valor em uma das portas de E/S do controlador de interrupções.

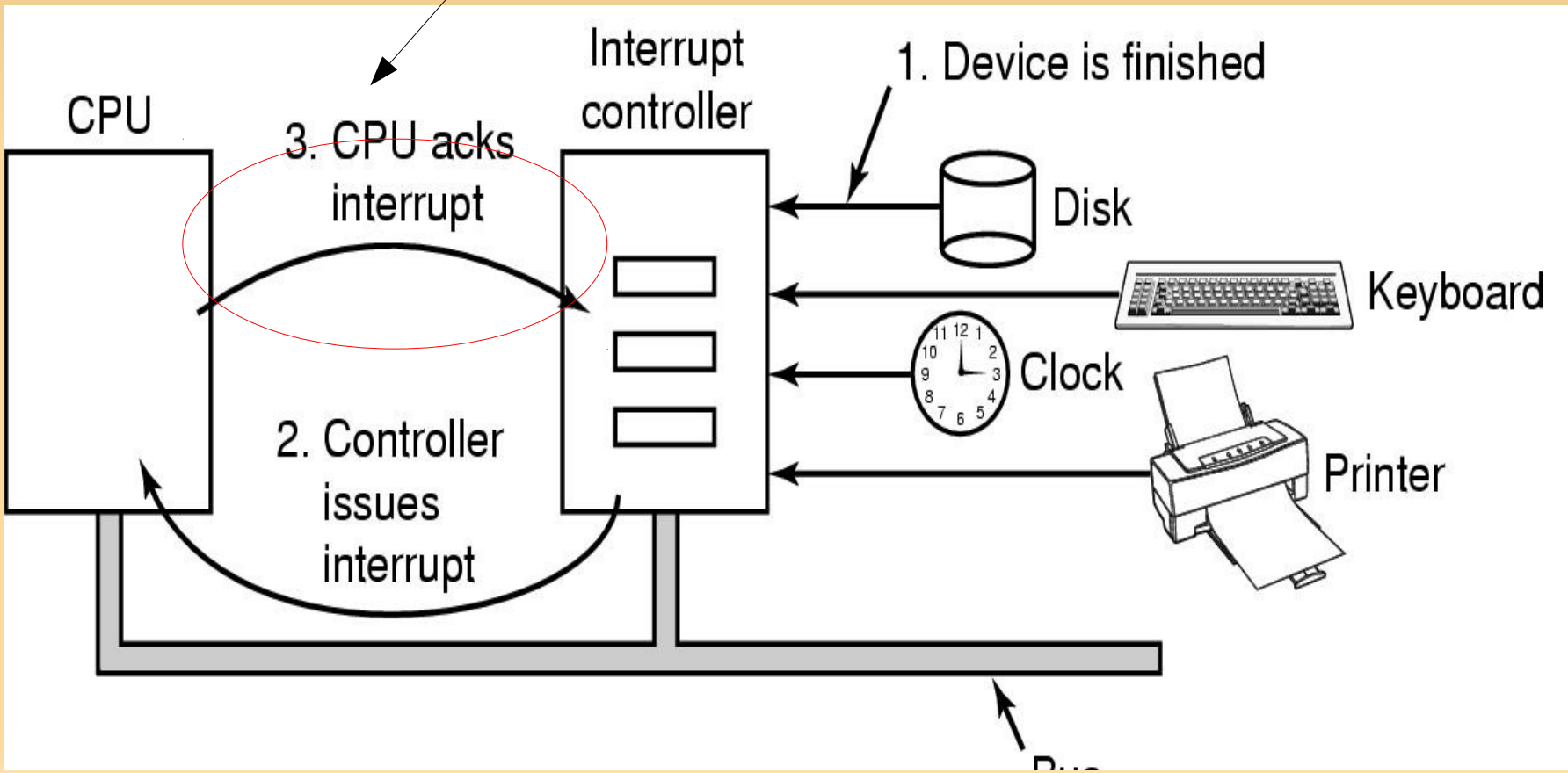
Isso diz ao controlador que está livre para mandar outra interrupção



Interrupção Revista

- Funcionamento

Contudo, podem haver condições de disputa entre múltiplos (quase simultâneos) sinais de interrupção. A solução é só enviar esse último sinal quando a CPU estiver pronta para tratar a próxima interrupção.



Interrupção Revista

- Informação salva pelo hardware antes de iniciar o procedimento de tratamento:
  - Pelo menos, o contador de programa
  - Registradores
  - O que salva varia de CPU para CPU
- Onde salvar (não há solução perfeita)?
  - Registradores internos
    - Não há como enviar o ACK ao controlador até que toda a informação neles tenha sido lida (evitando sobrescrita) → toma tempo

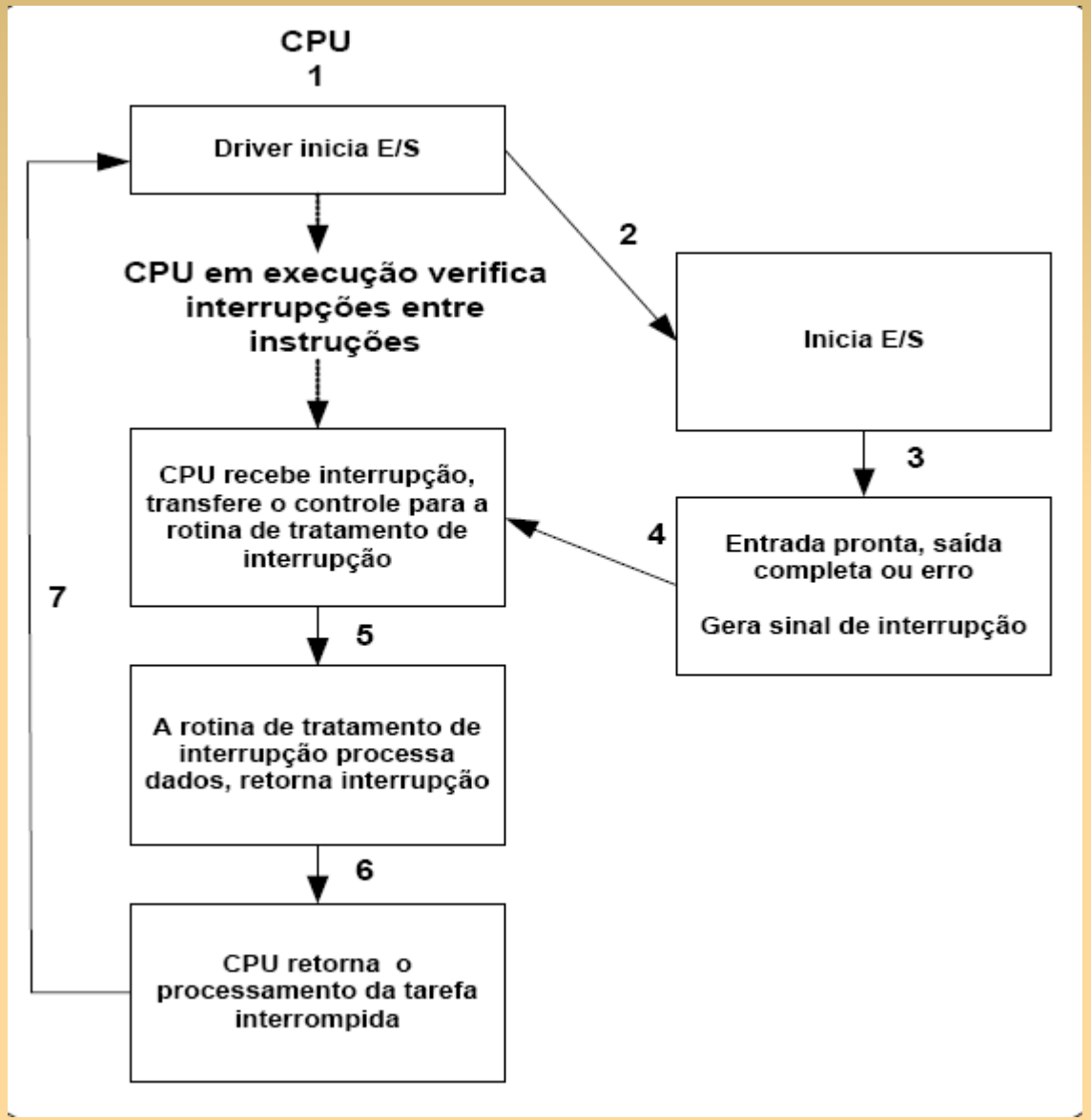
Interrupção Revista

- Onde salvar (não há solução perfeita)?
  - Pilha atual (do processo do usuário)
    - O ponteiro da pilha pode não ser legal (por escalonamento, página não na memória, por exemplo)
    - Poderia ser o final de uma página, levando a um page fault durante a interrupção
  - Pilha do kernel
    - Maior chance do ponteiro ser legal
    - Mudança de contexto (*user to kernel mode*) podendo mudar cache e TLB → toma tempo

Interrupção Revista

- O sinal (linha) de interrupção é exibido dentro de cada ciclo de instrução do processador;
  - A cada ciclo de instrução, a CPU:
    - Verifica se existe interrupção
    - Se não → busca próxima instrução,…
    - Se existe interrupção pendente:
      - Suspende a execução do programa;
      - Salva contexto;
      - Atualiza PC (Program Counter) → PC aponta para a ISR (rotina de atendimento de interrupção);
      - Executa interrupção;
      - Recarrega contexto e continua processo interrompido;

Interrupção Revista



Princípios de Software

- E/S via Interrupção
  - Utilizado para superar o problema da espera da UCP por operações nos periféricos
    - Ex: impressora que não armazena caracteres. Quando está pronta para receber mais dados, gera interrupção
  - A UCP:
    - envia um comando para o módulo de E/S e passa a executar outra tarefa;
    - quando a operação for concluída, o módulo de E/S interrompe a UCP;
    - a UCP executa a troca de dados, liberando o módulo de E/S e retomando o processamento anterior.

Princípios de Software

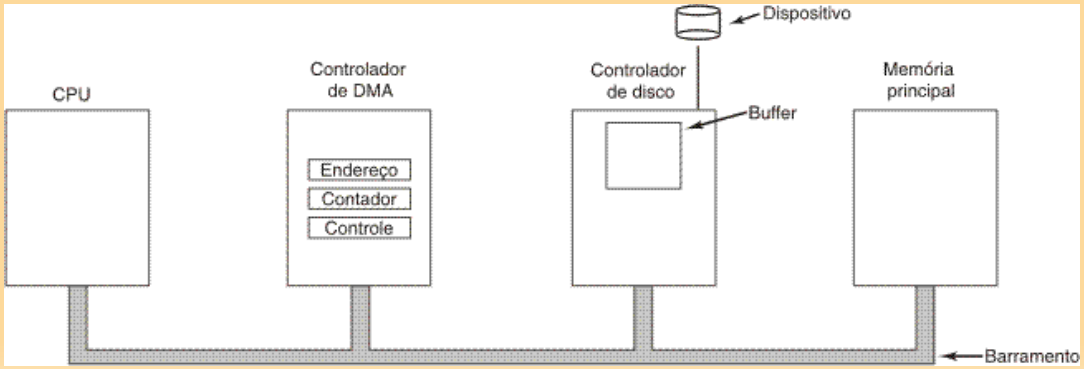
- E/S via Interrupção
  - Permite que uma unidade ganhe a atenção imediata de outra, de forma que a primeira possa finalizar sua tarefa
  - Geralmente são associados números às interrupções, onde o menor número tem prioridade sobre o maior (vetor de interrupção - índices)
  - Ex:Mapeamento das interrupções em um sistema compatível com IBM
    - 1 – teclado; 4 porta COM1; 5 – placa de som;
    - 6 – controlador de floppy; 7 – LPT1

Princípios de Software

- E/S via Acesso Direto à Memória
  - Inconvenientes das técnicas anteriores:
    - Limitam a capacidade de transferência da UCP, entre o módulo de E/S e a Memória Principal
    - UCP fica ocupada no gerenciamento
    - Se a quantidade de dados for grande, o desempenho do sistema será comprometido
      - No caso de interrupção, gera uma a cada caractere

Princípios de Software

- E/S via Acesso Direto à Memória
  - Solução: permitir o acesso direto à memória, tirando o gerenciamento da CPU
    - Direct Memory Access (DMA)
  - Necessita de um controlador de DMA
    - Hardware da CPU, placa mãe, ou dispositivo
  - Deve ter acesso independente ao barramento da CPU:
    - Além de registradores que podem ser lidos e escritos pela CPU



Princípios de Software

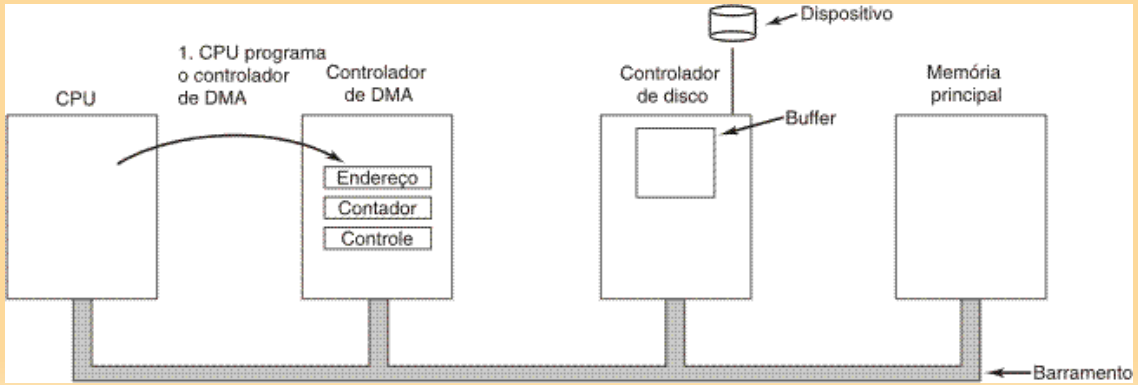
- E/S via Acesso Direto à Memória
  - Registradores:
    - Registrador de endereço de memória
    - Contador de bytes
    - Um ou mais registradores de controle, especificando:
      - O porto de E/S a ser usado
      - Direção da transferência (do dispositivo ou para o dispositivo)
      - Unidade de transferência (byte ou palavra por vez)
      - Número de bytes a serem transferidos em uma única operação

Princípios de Software

- E/S via Acesso Direto à Memória
  - E/S sem DMA – disco:
    - O controlador do disco lê o bloco (um ou mais setores) do disco (bit a bit, em série)
      - Até que o bloco inteiro esteja no buffer interno do controlador
    - Calcula o checksum para ver se houve erros de leitura
    - Gera uma interrupção
      - Quando o SO começa a rodar, pode ler o bloco do buffer do controlador
        - Um byte ou palavra por vez, em um laço
        - Cada byte é lido de um registrador no controlador de dispositivo, e armazenado na memória principal

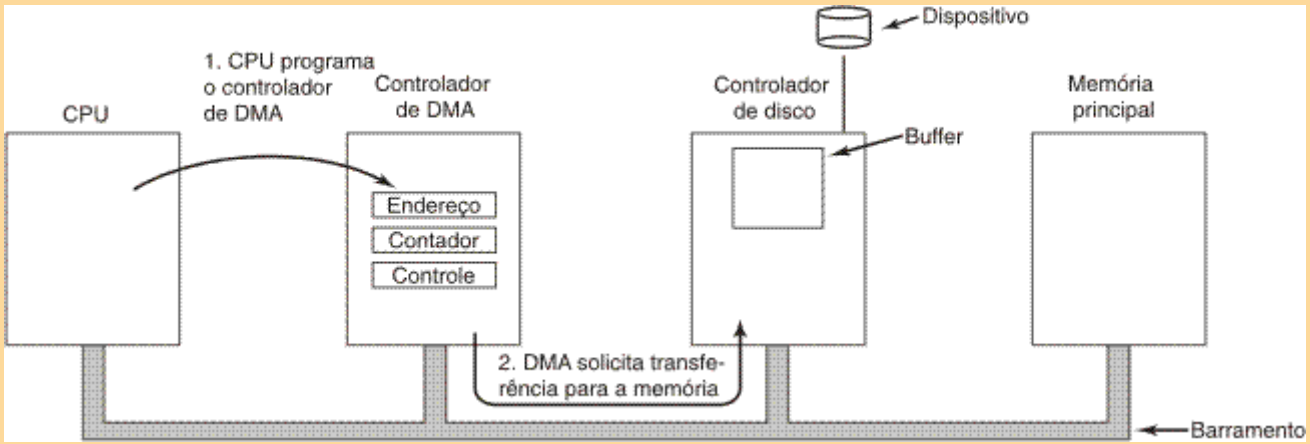
Princípios de Software

- E/S via Acesso Direto à Memória
  - E/S com DMA – disco:
    - A CPU programa o controlador de DMA
      - Colocando valores em seus registradores
        - Ele sabe então o que transferir, quanto e para onde
      - CPU continua seu trabalho
    - O controlador envia um comando ao controlador de disco, dizendo-o para ler dados do disco, armazenar em seu buffer interno e verificar o checksum
    - Com dados válidos, no buffer do controlador. DMA pode começar



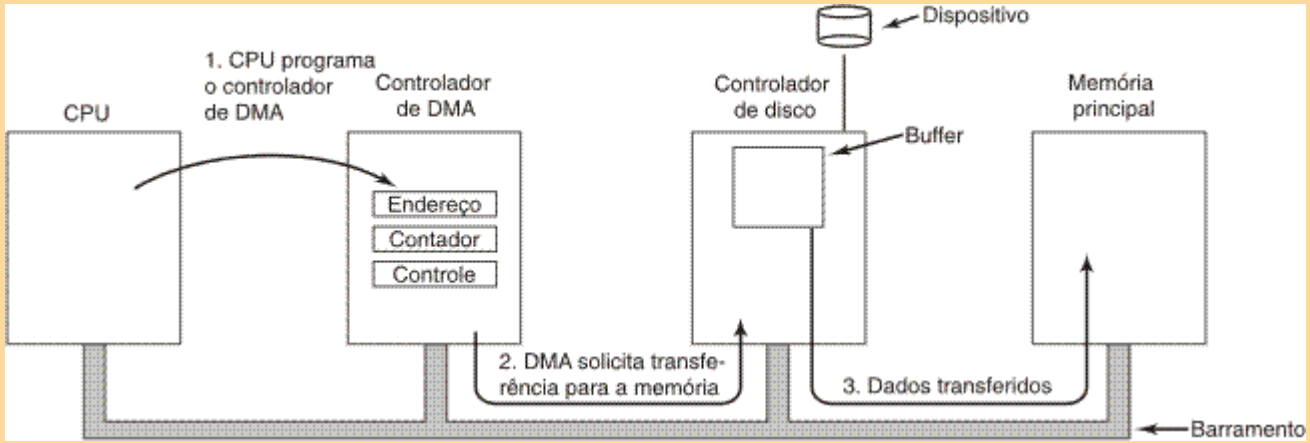
Princípios de Software

- E/S via Acesso Direto à Memória
  - E/S com DMA – disco:
    - O controlador de DMA inicia a transferência enviando um pedido de leitura (via barramento) ao controlador de disco
      - O endereço de memória onde os dados devem ser armazenados está nas linhas de endereço do barramento



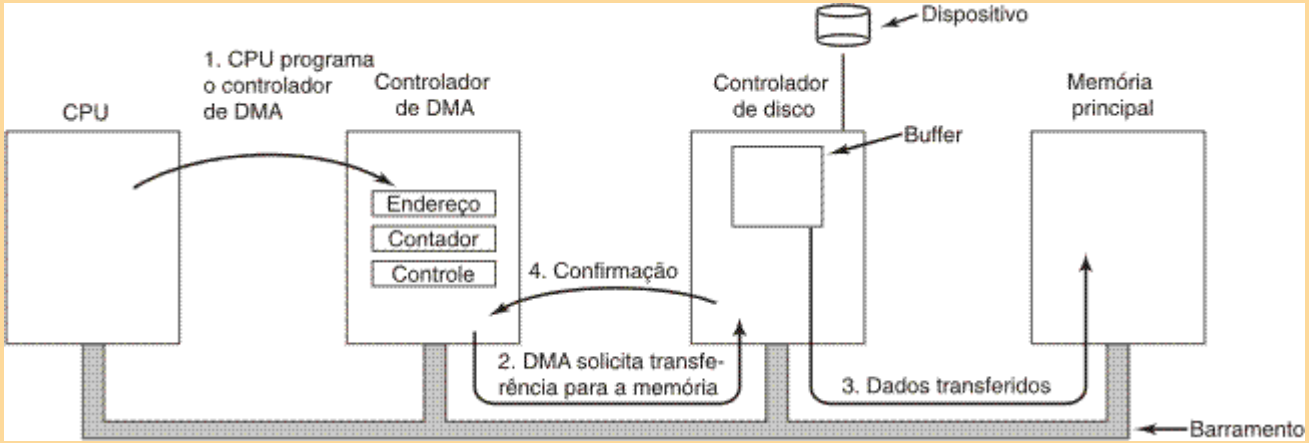
Princípios de Software

- E/S via Acesso Direto à Memória
  - E/S com DMA – disco:
    - No próximo ciclo do barramento, automaticamente ocorre a escrita na posição indicada da memória



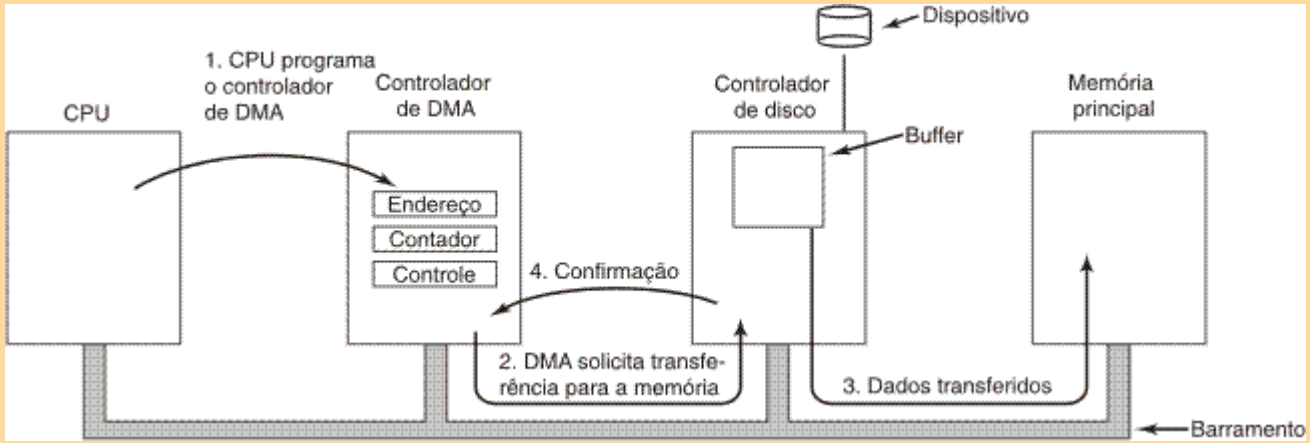
Princípios de Software

- E/S via Acesso Direto à Memória
  - E/S com DMA – disco:
    - Quando a escrita é completada, o controlador de disco envia uma confirmação ao controlador de DMA, também via barramento



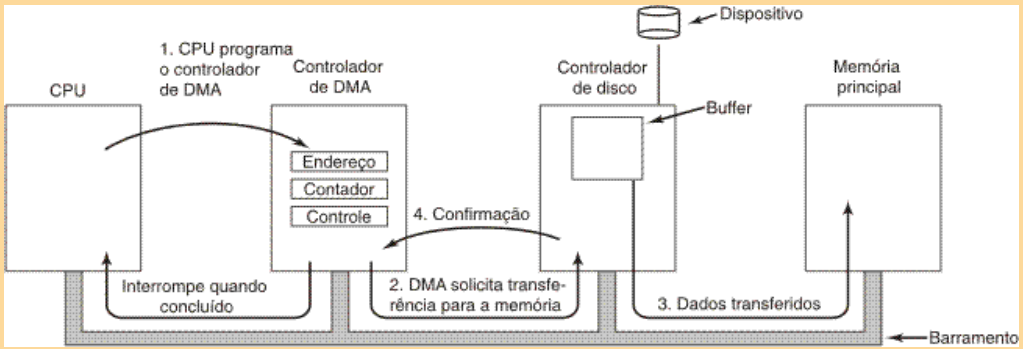
Princípios de Software

- E/S via Acesso Direto à Memória
  - E/S com DMA – disco:
    - O controlador de DMA incrementa o endereço de memória a ser usado, decrementando o contador de bytes
      - Se o contador ainda for maior que 0, repete os passos 2 a 4



Princípios de Software

- E/S via Acesso Direto à Memória
  - E/S com DMA – disco:
    - O controlador de DMA incrementa o endereço de memória a ser usado, decrementando o contador de bytes
      - Se o contador for 0, interrompe a CPU
      - A UCP pode executar a rotina de tratamento da interrupção, processando os dados lidos ou produzindo novos dados para serem escritos



Princípios de Software

- E/S via Acesso Direto à Memória
  - Transferência múltiplas
    - Controladores de DMA mais sofisticados, podem lidar com vários periféricos diferentes
      - Cada um utilizando um canal de DMA (DMA channel)
    - Devem conter múltiplos conjuntos de registradores
      - Um para cada canal
    - A CPU carrega cada conjunto com os parâmetros relevantes
    - Após cada transferência de palavra, o controlador decide quem atender em seguida
      - Escalonamento (round robin, prioridade etc)

Princípios de Software

- E/S via Acesso Direto à Memória
  - Roubo de ciclos
    - Durante o acesso à memória, o DMA acessa o barramento
    - Se a CPU quiser usar, terá que esperar
      - Há o roubo ocasional de um ciclo, atrasando um pouco a CPU
    - Há alternativas, mas também com seus contras:
      - Burst mode: Em vez de requisitar o barramento a cada palavra, o controlador de DMA:
        - Diz ao dispositivo para segurar o barramento
        - Envia uma série de transferências
        - Libera o barramento
      - Embora mais eficiente, pode bloquear o acesso da CPU ao barramento por um tempo substancial

Princípios de Software

- E/S via Acesso Direto à Memória
  - Modos do controlador
    - Fly-by mode
      - Visto até agora – o controlador de DMA diz ao controlador de dispositivo para transferir os dados à memória
    - Alternativamente
      - O dispositivo envia a palavra ao controlador de DMA (usando o barramento)
      - Este então (novamente via barramento), envia à memória
      - Vantagem:
        - Pode executar cópias de dispositivo a dispositivo, sem passar pela memória (e.g. CD-ROM para HD)
      - Problema:
        - Ciclo extra de barramento a cada transferência

Princípios de Software

- E/S via Acesso Direto à Memória
  - Desvantagem:
    - A CPU é mais rápida que o controlador de DMA, e pode fazer o trabalho mais rapidamente
      - Velocidades incompatíveis
  - Vantagem:
    - DMA executa E/S programada → controladora de DMA faz todo o trabalho ao invés da CPU;
      - Redução do número de interrupções;
        - De uma por byte ou palavra, para uma por buffer
      - Libera a CPU para executar outras operações

Princípios de Software
Camadas de Software de E/S

- Organizar o software como uma série de camadas facilita a independência dos dispositivos:
  - Camadas mais baixas apresentam detalhes de hardware:
    - Drivers e manipuladores de interrupção;
  - Camadas mais altas apresentam interface para o usuário:
    - Aplicações de Usuário;
    - Chamadas de Sistemas;
    - Software Independente de E/S ou Subsistema de Kernel de E/S;

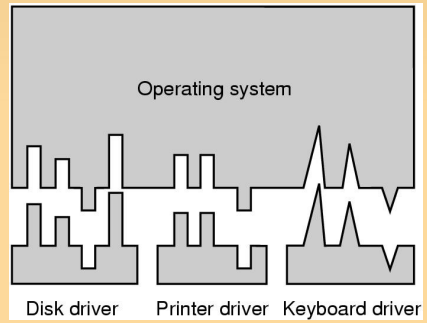
Princípios de Software
Camadas de Software de E/S

- Software de E/S Independente de Dispositivo
  - Algumas partes do software de E/S são específicas e outras independentes de dispositivo
  - Sua principal função é executar funções de E/S comuns a todos os dispositivos
  - Outras funções:
    - Fornecer interface uniforme ao software do usuário
      - Evita que a cada novo dispositivo criado, o SO tenha que ser modificado.
      - Uniformiza a interface entre o SO e seus drivers de dispositivos
        - Veremos drivers mais adiante

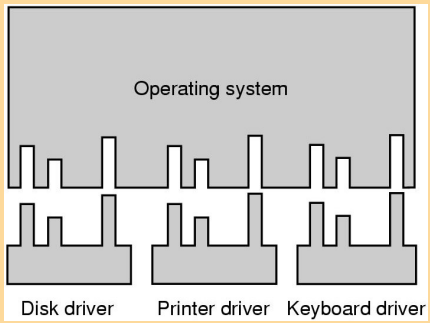
Princípios de Software
Camadas de Software de E/S

- Software de E/S Independente de Dispositivo
  - Outras funções:
    - Fornecer interface uniforme ao software do usuário:
      - Cada dispositivo fornece driver com funções diferentes × Há padrão nas funções fornecidas
        - Caso em que, para cada classe de dispositivos, o SO define um conjunto de funções que o driver deve fornecer.
    - Atribuir um nome lógico a partir do qual o dispositivo é identificado;
      - Ex.: UNIX → (/dev), put(), get()

        name and value pairs



Sem interface padrão



Com interface padrão (uniforme)

Princípios de Software
Camadas de Software de E/S

- Software de E/S Independente de Dispositivo
  - Outras funções:
    - Fazer o escalonamento de E/S;
    - Prover buffering:
      - Ajuste entre a velocidade e a quantidade de dados transferidos;
    - Cache de dados
      - Armazenar na memória um conjunto de dados freqüentemente acessados;
    - Gerenciar alocação, uso e liberação dos dispositivos
      - Acessos concorrentes tanto a recursos compartilháveis (que podem ser utilizados por vários usuários ao mesmo tempo – disco etc) quanto a dedicados (que podem ser utilizados por apenas um usuário de cada vez – impressora etc)

Princípios de Software
Camadas de Software de E/S

- Software de E/S Independente de Dispositivo
  - Outras funções:
    - Reportar erros e proteger os dispositivos contra acessos indevidos :
      - Erros de programação:
        - Ex.: tentar efetuar leitura de um dispositivo de saída (impressora, vídeo);
      - Erros de E/S:
        - Ex.: tentar imprimir em uma impressora desligada ou sem papel;
      - Erros de memória:
        - Escrita em endereços inválidos;
  - Definir tamanhos de blocos independentes do dispositivo