



Capítulo 3

Entrada/Saída





Roteiro

Princípios do hardware de E/S

Princípios do software de E/S

Camadas do software de E/S

Impasses

Dispositivos baseados em bloco

Dispositivos baseados em sequência de caracteres

Introdução



O controle da E/S é uma das tarefas centrais de um sistema operacional, que:

- emite comandos para os controladores de dispositivo, processa interrupções e trata de erros
- esconde detalhes específicos dos diferentes dispositivos para o resto do S.O. através de uma interface uniforme
- tenta paralelizar E/S e processamento da CPU/Memória para *minimizar latência* e *maximizar vazão* de dados no acesso aos dispositivos de E/S
- controla o acesso concorrente aos dispositivos

Sub-sistema de E/S

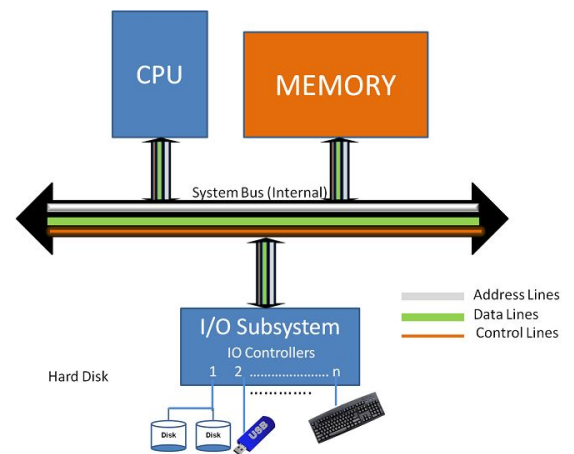


O sub-sistema de E/S consiste de duas camadas de SW:

- Software independente dos dispositivos (para cada classe de dispositivo)
- Software dependente do dispositivo (drivers)

Hardware:

- Dispositivos e Controladora
- Controladora DMA
- Barramento (p.ex. PCI, SCSI, ...)



Tipos de Dispositivo



O Dispositivo de E/S é definido por:

- Conjunto de comandos básicos disponibilizados
- As funções que executa
- As mensagens de status/erro que emite
-

Existem os seguintes tipos de dispositivo:

Dispositivos de bloco = transferem ou armazenam dados em blocos de tamanho fixo, **cada bloco possui um endereço** e cada bloco pode ser acessado independentemente (acesso aleatório). Exemplos: Discos (SCSI, IDE), CD/DVD, Fitas, SSDs

Dispositivos de caractere = lê ou escreve uma **sequência de caracteres**. Cada caractere não possui endereço e não pode ser acessado aleatoriamente. Para cada caractere (ou conjunto pequeno deles), é gerada uma interrupção. Exemplos: Teclado, Mouse, interface serial, etc.

Obs: Esta classificação é geral: nem todo dispositivo de E/S se enquadra exatamente em uma dessas categorias. (e.g. fita magnética, vídeo mapeado em memória, modem, Interface de Rede etc.

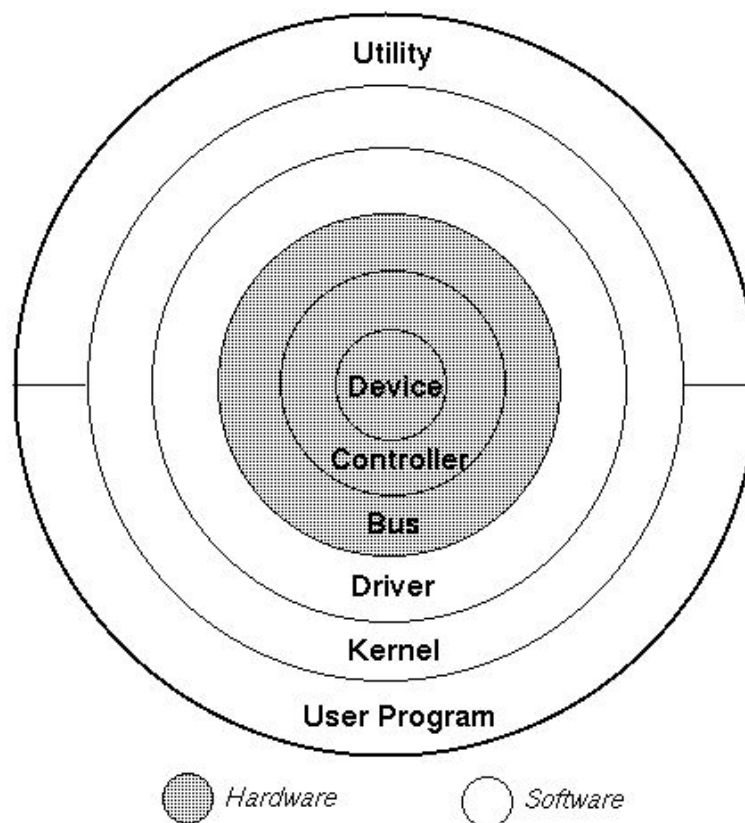
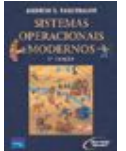
Categorias de Dispositivos

Outra classificação



- Para Interação Homem-Computador
 - Teclado, mouse, Display
 - Impressoras
- Para armazenamento e processamento interno
 - Discos
 - Pen-drives
 - Sensores
 - Controladores
 - Atuadores
- Para comunicação remota
 - Interfaces de rede (Ethernet, 802.11, Bluetooth, ...)
 - Modems

IO em camadas



Princípios do Hardware de E/S



Dispositivo	Taxa de dados
Teclado	10 bytes/s
Mouse	100 bytes/s
Modem 56 K	7 KB/s
Canal telefônico	8 KB/s
Linhas ISDN dual	16 KB/s
Impressora a laser	100 KB/s
Scanner	400 KB/s
Ethernet clássica	1,25 MB/s
USB (<i>universal serial bus</i> — barramento serial universal)	1,5 MB/s
Câmara de vídeo digital	4 MB/s
Disco IDE	5 MB/s
CD-ROM 40x	6 MB/s
Ethernet rápida	12,5 MB/s
Barramento ISA	16,7 MB/s
Disco EIDE (ATA-2)	16,7 MB/s
FireWire (IEEE 1394)	50 MB/s
Monitor XGA	60 MB/s
Rede SONET OC-12	78 MB/s
Disco SCSI Ultra 2	80 MB/s
Ethernet Gigabit	125 MB/s
Dispositivo de Fita Ultrium	320 MB/s
Barramento PCI	528 MB/s
Barramento da Sun Gigaplane XB	20 GB/s

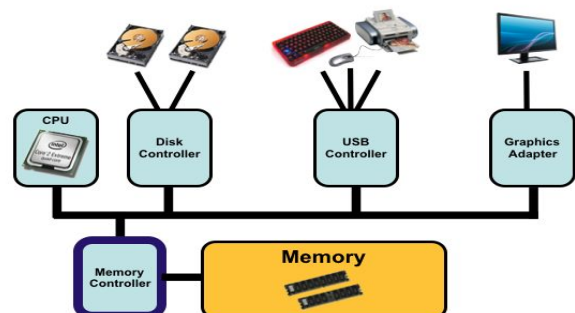
Taxas de dados típicas de dispositivos, redes e barramentos

Controladores de Dispositivos



- Partes de dispositivos de E/S
 - Componente eletro-mecânico (e-motor, Electronic Speed Controller, cabeçote magnético, sensores, etc.)
 - controlador do dispositivo (um circuito integrado ou processador)
- A interface física (pinos) entre o controlador e a componente mecânica é padronizada (ISO, SCSI, IDE)
- Um controlador pode controlar um ou mais dispositivos do mesmo tipo (componentes eletro-mecânicas). Suas tarefas típicas:
 - converter fluxo serial de bits para conjuntos de bytes
 - Verificação da consistência dos dados (checksum) e correção de erros
 - Bufferização: agrupar bloco de bytes para transferência para a memória principal

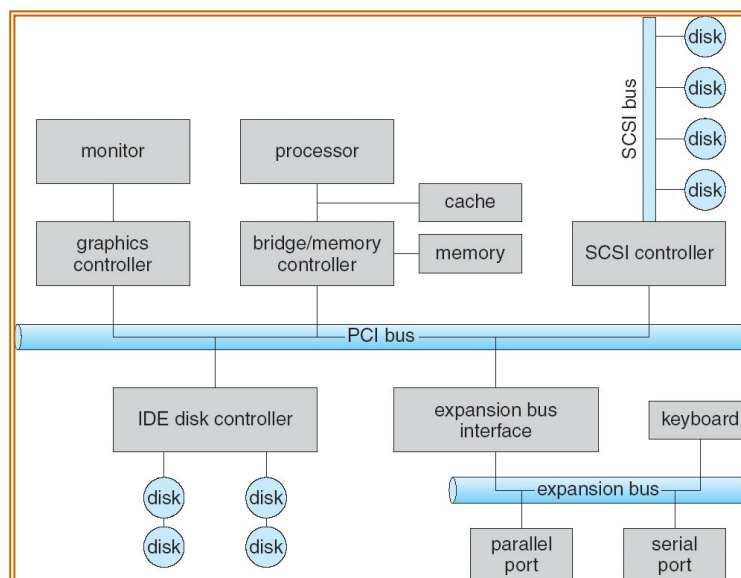
Do ponto de vista conceitual,
o controlador é o dispositivo de E/S





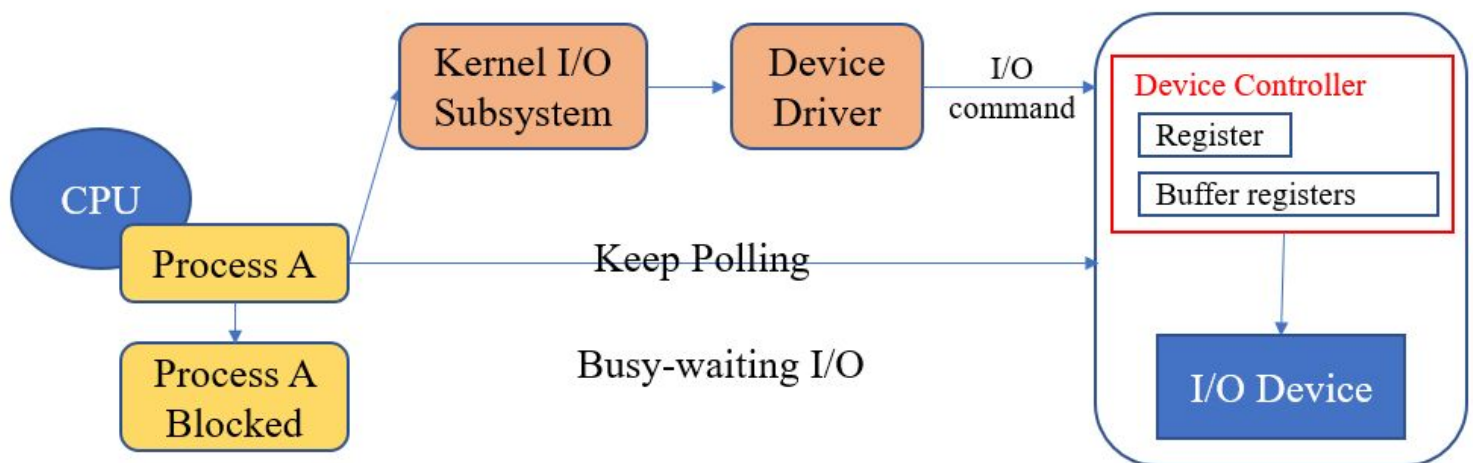
Arquitetura Conceitual

Conceitualmente, é como se houvesse conexão direta entre CPU/Memória e todos os controladores



Em arquiteturas reais, existem vários barramentos e circuitos dedicados para seu acesso, que servem de ponte entre os barramentos

Driver e Controladora de Dispositivo



Foonte: Cheng Chang: Medium: Operating System Chapter 2, Nov 2022

O Device Driver é o módulo de software do núcleo específico para interagir com o Device Controller.

Driver e Controladora de Dispositivo

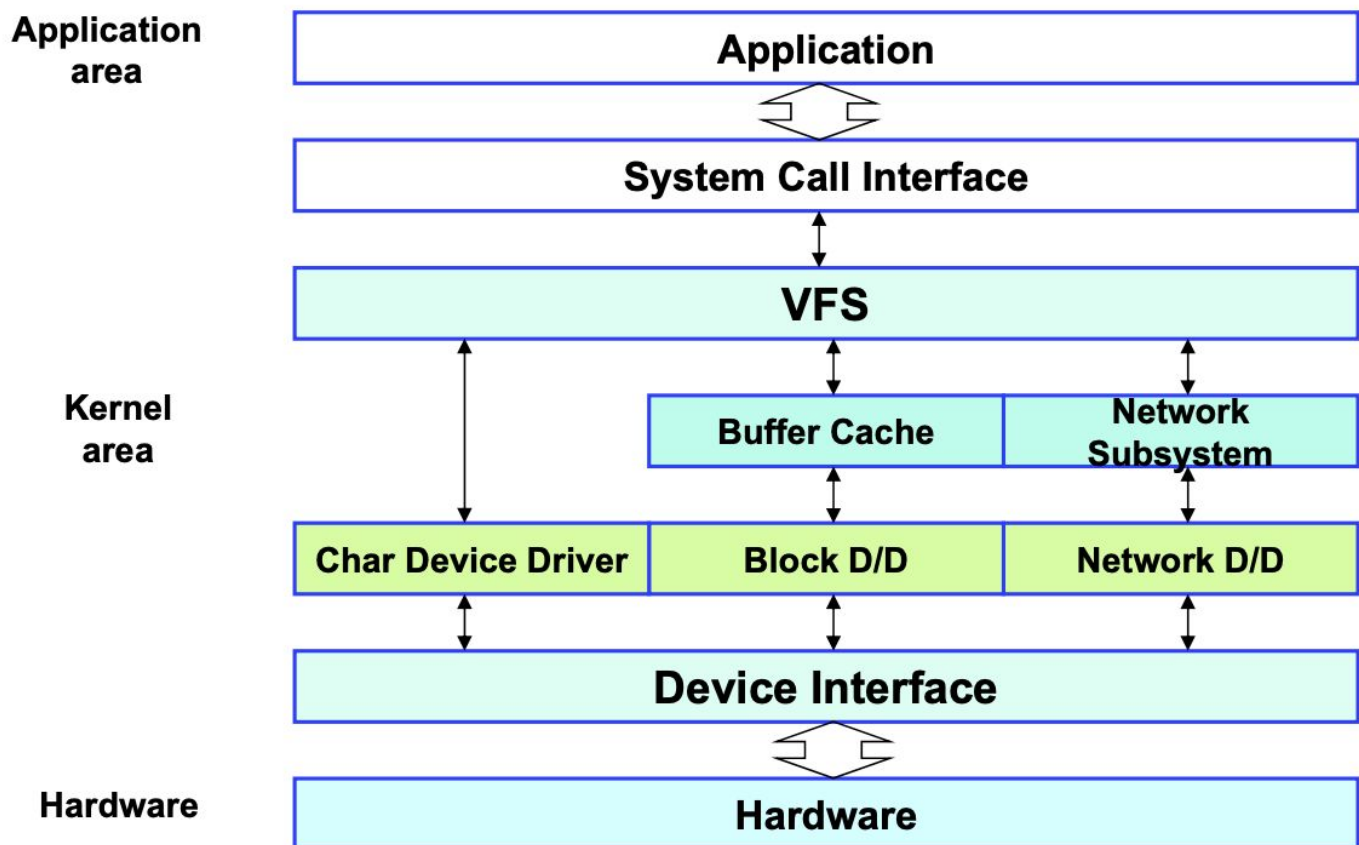
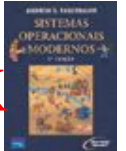


Passo a Passo para E/S programada (Polling)

1. O processo em execução faz uma sys_call de um serviço de E/S e causa um TRAP no núcleo (por exemplo, ler um arquivo no disco)
2. O núcleo recebe a solicitação, **bloqueia o processo**, e a CPU fica aguardando a conclusão da operação de E/S.
3. O subsistema de E/S no núcleo trata a solicitação. Por exemplo, **verifica se os blocos do arquivo estão no block cache**: se sim → busca os dados no cache; caso contrário, executa a E/S de fato no disco.
4. O subsistema de E/S passa o comando **para o driver do dispositivo**.
5. O driver do dispositivo gera os comandos básicos específicos do tipo do seu dispositivo (HD, SSD, USB pen drive) para a controladora.
6. A controladora escolhe o dispositivo para efetuar as operações de E/S (caso sejam várias unidades)
7. Como a CPU está ociosa o escalonador do núcleo do sistema operacional pode alocar a CPU para outros processos.
8. Repetidamente a CPU consulta o controlador do dispositivo de E/S (através do driver) para verificar se a operação de E/S foi concluída ou não (com erro ou não).

Obs: A interface de comandos, status e interrupções de um dispositivo é chamada de porta de E/S.

Tipos de dispositivos Linux/Unix

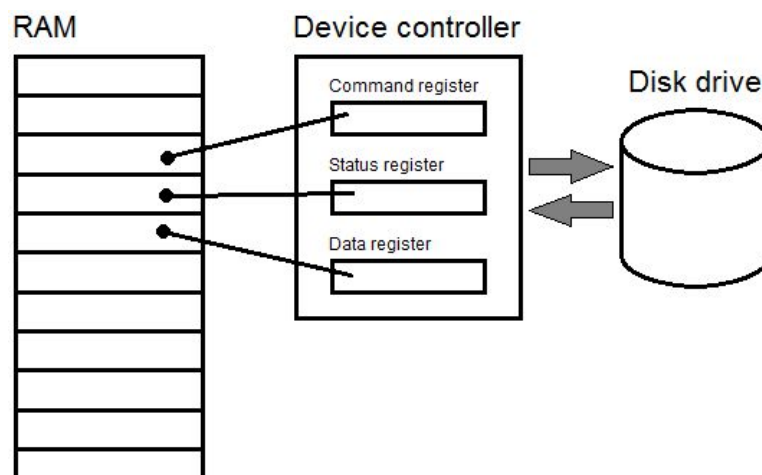




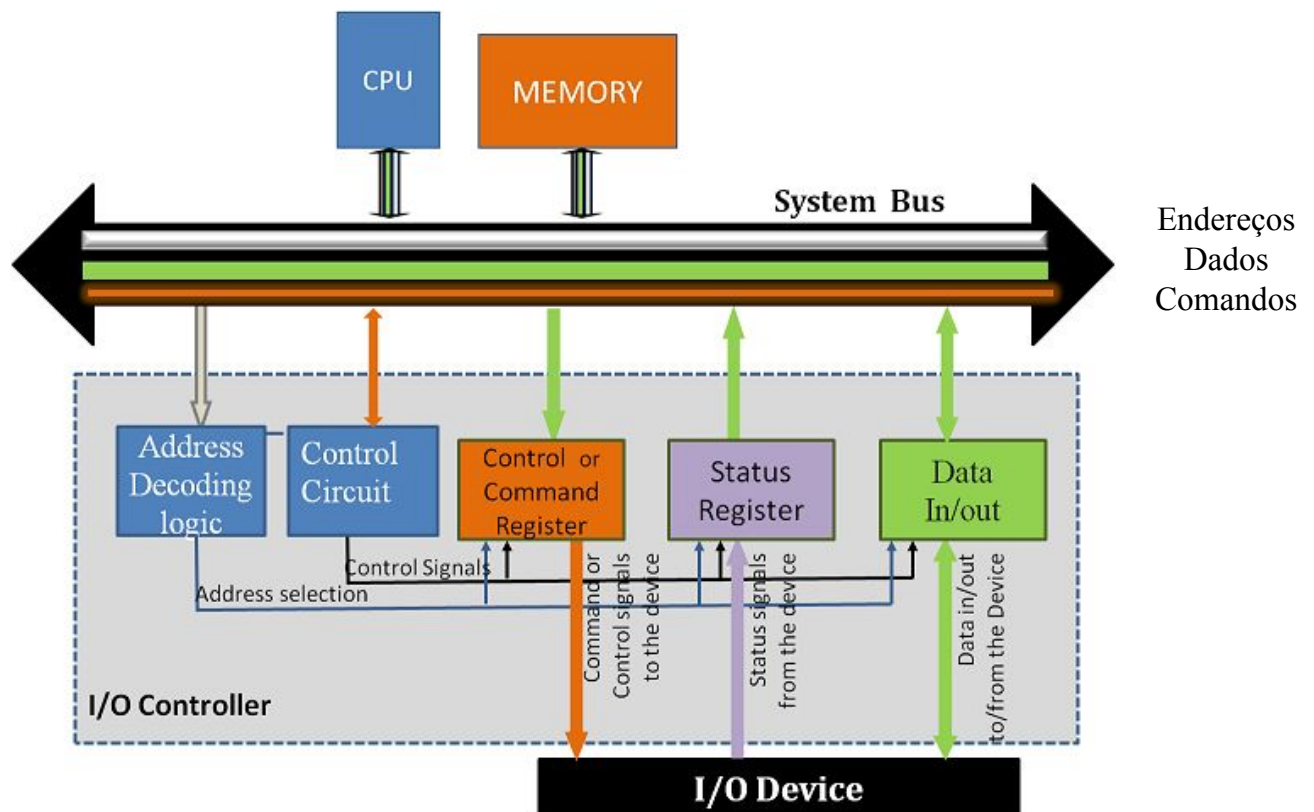
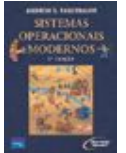
Controladores de Dispositivos

Cada controlador possui endereços na memória RAM que são usados para monitorar e controlar o dispositivo de E/S e emitir informações sobre status e condições de erro.

- Cada um desses endereços de memória é uma **porta de E/S**:
 - Reg. de status (bits)
 - Reg. de controle (bits)
 - Reg. de entrada de dados (1-4 bytes)
 - Reg. de saída de dados (1-4 bytes)



Controlador de Dispositivo de E/S

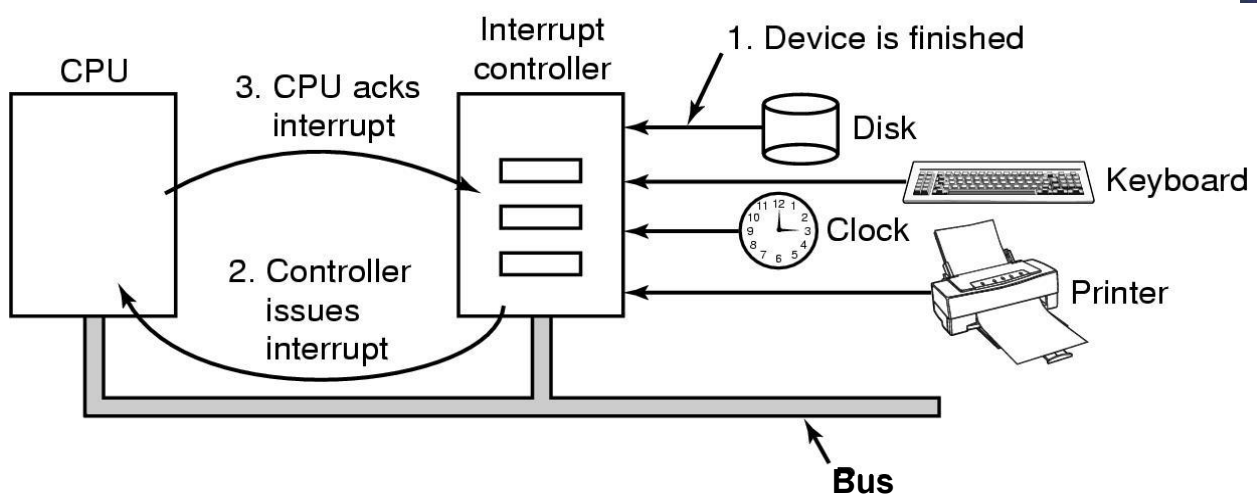


Controladores de Dispositivos



- Em algumas arquiteturas, o espaço de endereçamento da memória RAM e das portas de E/S é único (**E/S mapeada em memória**) Exemplo: Motorola 680x0
- Em outras arquiteturas, existe um espaço de endereçamento específico para E/S (**E/S fora da memória**), usado apenas para interação com os controladores.
- A associação de um endereço com a porta de E/S de um controlador é feita no circuito de decodificação de endereços do barramento.

Interrupções de Hardware



Obs: Conexão entre dispositivos e controlador de interrupções usam “linhas de interrupção” nos barramentos (em vez de cabos dedicados)

Tecnologia Plug’nPlay: Ao se conectar um dispositivo de E/S na placa mãe, a BIOS automaticamente atribui um IRQ livre para o dispositivo (Exemplo: Pentium PCs possuem 15 IRQs.)

Passos de uma leitura de bloco controlada por CPU



1. CPU escreve comando (p.ex., ler bloco de determinado endereço) em registrador do controlador de disco
2. Controlador aciona a componente eletro-mecânica, e transfere os dados bit-a-bit do disco para um buffer do controlador
3. Controlador calcula a checksum para verificar se há dados corrompidos. Se houver, faz nova leitura.
4. Quando bloco de dados foi completamente lido, controlador levanta uma interrupção
5. Driver executa loop para copiar byte-by-byte o bloco do controlador para a memória principal

Obs: o driver executa na CPU

Direct Memory Access (DMA)



Em várias arquiteturas existe uma componente de HW dedicada à transferência direta de blocos de dados para a memória - Direct Memory Access (DMA):

- Evita que a CPU tenha que executar o loop para transferência do bloco de/para a memória principal.
- CPU apenas inicia E/S informando: endereço inicial na memória, endereço do bloco no disco, número de bytes a serem transferidos.

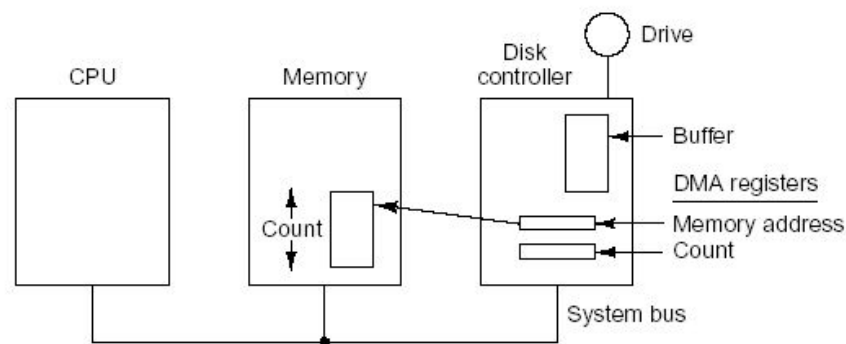
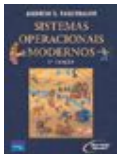
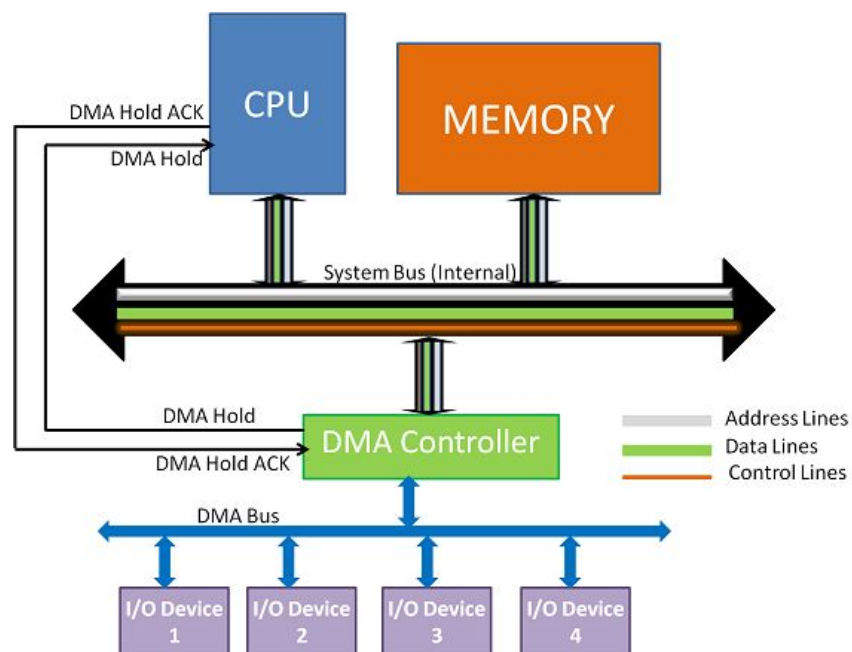


Figure 3-3. A DMA transfer is done entirely by the controller.

Direct Memory Access (DMA)



CPU só configura o DMA e a transferência de dados dispositivos (de bloco) /para a memória principal, e vice-versa, ocorre sem a participação da CPU



Fonte: <https://witscad.com/course/computer-architecture/chapter/dma-controller-and-io-processor>