

# Aula 16

# Sistemas Operacionais I

## **Gerenciamento de E/S - Parte 1**

Prof. Julio Cezar Estrella

jcezar@icmc.usp.br

*Material adaptado de*

*Sarita Mazzini Bruschi*

*baseados no livro Sistemas Operacionais Modernos de A.  
Tanenbaum*

# Gerenciamento de dispositivos de Entrada e Saída

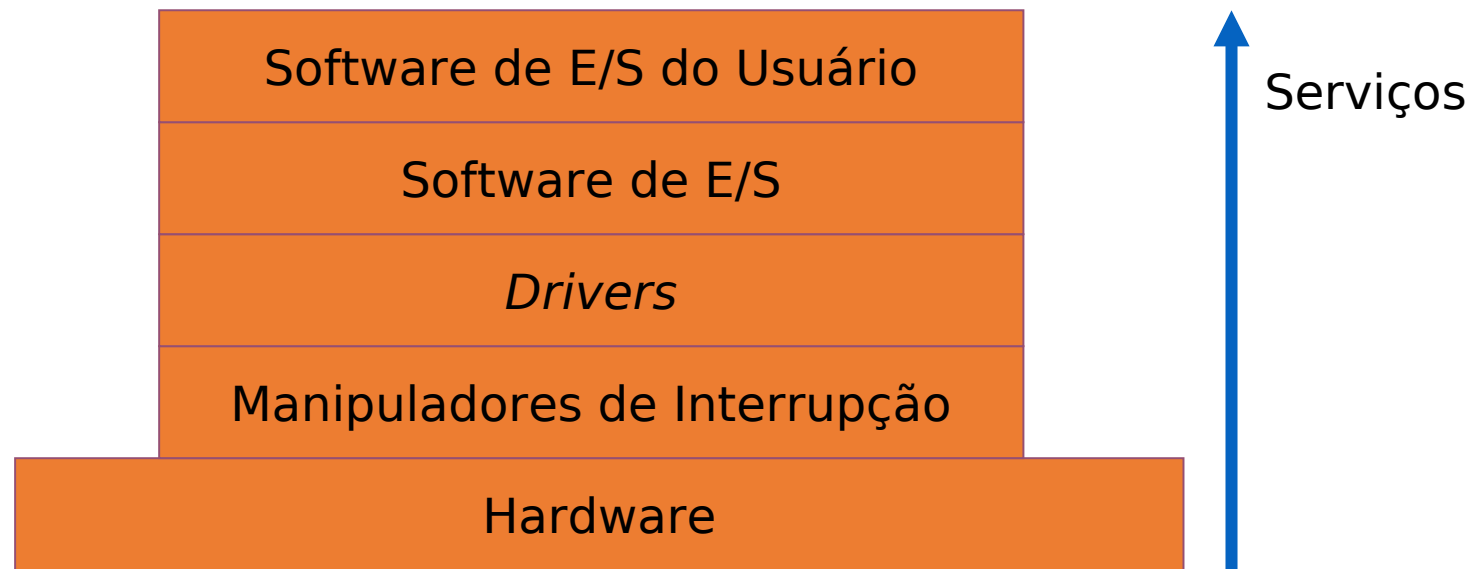
- SO pode atuar de duas maneiras diferentes:
  - Como máquina estendida (*top-down*) – tornar uma tarefa de baixo nível mais fácil de ser realizada pelo usuário;
  - Como gerenciador de recursos (*bottom-up*) – gerenciar os dispositivos que compõem o computador;

# Gerenciamento de dispositivos de Entrada e Saída

- Funções específicas:
  - Enviar sinais para os dispositivos;
  - Atender interrupções;
  - Gerenciar comandos aceitos e funcionalidades (serviços prestados);
  - Tratar possíveis erros;
  - Prover interface entre os dispositivos e o sistema;
- Princípios:
  - Hardware;
  - Software;

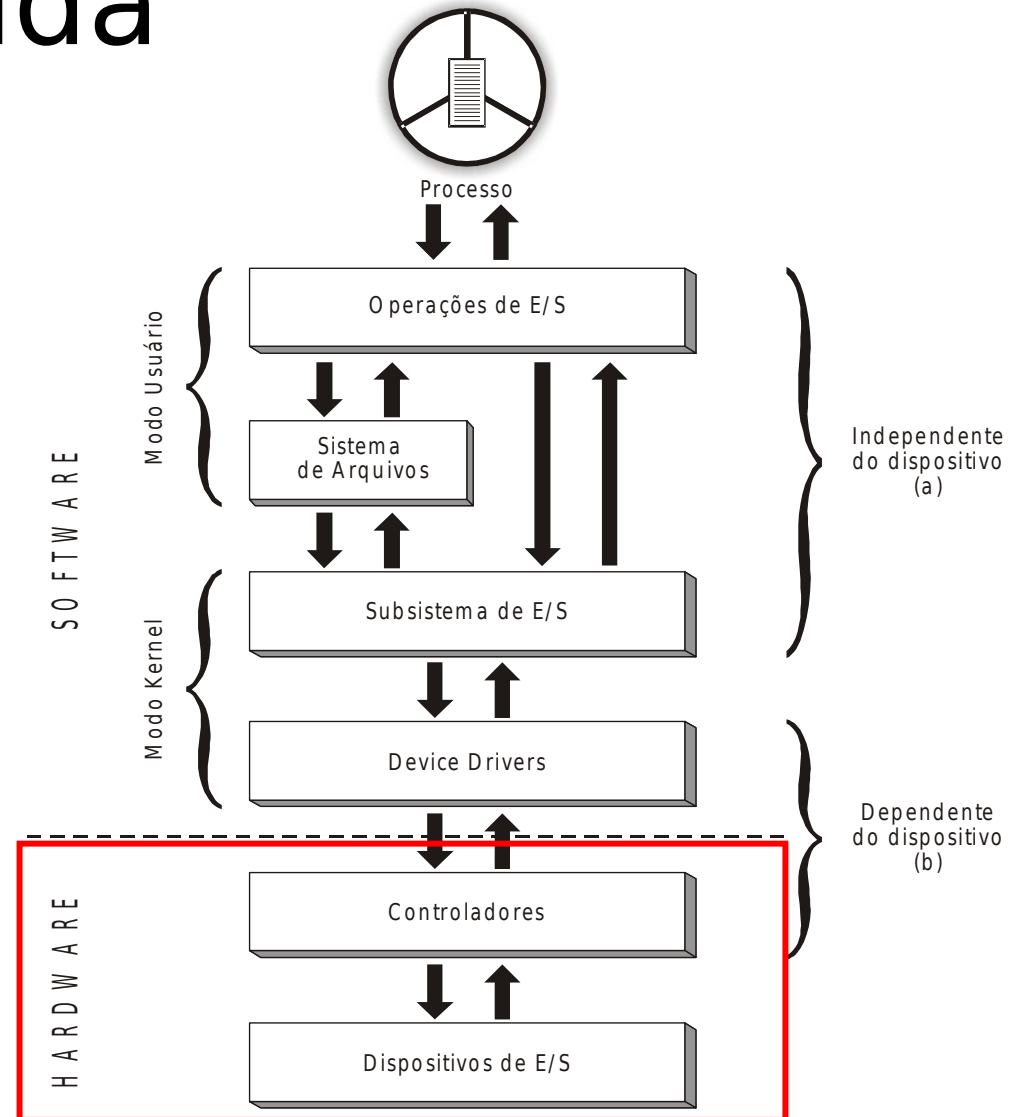
# Gerenciamento de dispositivos de Entrada e Saída

- Visão de camadas do Tanenbaum



# Gerenciamento de dispositivos de Entrada e Saída

- Visão de camadas



# Dispositivos de E/S

## Princípios de Hardware

- Podem ser divididos em duas categorias:
  - Dispositivos baseados em bloco: informação é armazenada em blocos de tamanho fixo, cada um com um endereço próprio;
    - Tamanho varia entre 512 bytes e 32.768 bytes;
    - Permitem leitura e escrita independentemente de outros dispositivos;
    - Permitem operações de busca;
    - Ex.: discos rígidos;

# Dispositivos de E/S

## Princípios de Hardware

- Dispositivos baseados em caracter: aceita uma seqüência de caracteres, sem se importar com a estrutura de blocos; informação não é endereçável e não possuem operações de busca;
  - Ex.: impressoras, interfaces de rede (placas de rede); placas de som;

# Dispositivos de E/S

## Princípios de Hardware

- Classificação não é perfeita, pois alguns dispositivos não se encaixam em nenhuma das duas categorias:
  - *Clocks*: provocam interrupções em intervalos definidos;
- Classificação auxilia na obtenção de independência ao dispositivo;
  - Parte dependente está a cargo dos *drivers* → software que controla o acionamento dos dispositivos;



# Dispositivos de E/S

## Princípios de Hardware

- Os dispositivos de E/S podem apresentar uma grande variedade de velocidade

Device	Data rate
Keyboard	10 bytes/sec
Mouse	100 bytes/sec
56K modem	7 KB/sec
Scanner	400 KB/sec
Digital camcorder	3.5 MB/sec
802.11g Wireless	6.75 MB/sec
52x CD-ROM	7.8 MB/sec
Fast Ethernet	12.5 MB/sec
Compact flash card	40 MB/sec
FireWire (IEEE 1394)	50 MB/sec
USB 2.0	60 MB/sec
SONET OC-12 network	78 MB/sec
SCSI Ultra 2 disk	80 MB/sec
Gigabit Ethernet	125 MB/sec
SATA disk drive	300 MB/sec
Ultrium tape	320 MB/sec
PCI bus	528 MB/sec

# Dispositivos de E/S

## Princípios de Hardware

- Dispositivos de E/S possuem basicamente dois componentes:
  - Mecânico → o dispositivo propriamente dito;
  - Eletrônico → controladores ou adaptadores (placas);
- O dispositivo (periférico) e a controladora se comunicam por meio de uma **interface**:
  - Serial ou paralela;
  - Barramentos: IDE, ISA, SCSI, AGP, USB, PCI, etc.

# Dispositivos de E/S

## Princípios de Hardware

- Cada controladora possui um conjunto de registradores de controle, que são utilizados na comunicação com a CPU;
- Além dos registradores, alguns dispositivos possuem um *buffer* de dados:
  - Ex.: placa de vídeo; algumas impressoras;
- O SO, utilizando os drivers, gerencia os dispositivos de E/S escrevendo/lendo nos/dos registradores/*buffers*,;
  - Comunicação em baixo nível – instruções em Assembler;
  - Enviar comandos para os dispositivos;
  - Saber o estado dos dispositivos;

# Dispositivos de E/S

## Princípios de Hardware

- Como a CPU se comunica com esses registradores de controle?
  - **Porta**: cada registrador de controle possui um número de porta (ou porto) de E/S de 8 ou 16 bits;
    - Instrução em *Assembler* para acessar os registradores;
    - Espaço de endereçamento diferente para a memória e para os dispositivos de E/S;
    - *Mainframes* IBM utilizavam esse método;
    - SOs atuais fazem uso dessa estratégia para a maioria dos dispositivos;

# Dispositivos de E/S

## Princípios de Hardware

- Comunicação com os registradores de controle:
  - **Memory-mapped (mapeada na memória)**: mapear os registradores de controle em espaços de memória:
    - Cada registrador possui um único endereço de memória;
    - Em geral, os endereços estão no topo da memória protegidos em endereços não utilizados por processos;
    - Uso de linguagem de alto nível, já que registradores são apenas variáveis na memória;
    - SOs utilizam essa estratégia para os dispositivos de vídeo;

# Dispositivos de E/S

## Princípios de Hardware

- Comunicação com os registradores de controle:
  - **Estratégia híbrida:**
    - Registradores → Porta;
    - *Buffers* → Mapeado na memória;
      - Exemplo: Pentium - endereços de 640k a 1M para os *buffers* e as portas de E/S de 0 a 64k para registradores;

# Dispositivos de E/S

## Princípios de Hardware

- Como funciona a comunicação da CPU com os dispositivos?
  - Quando a CPU deseja ler uma palavra, ela coloca o endereço que ela está desejando no barramento de endereço e manda um comando **READ** no barramento de controle;
  - Essa comunicação pode ser controlada pela própria CPU ou pela DMA;

# Dispositivos de E/S

## Maneiras de realizar E/S

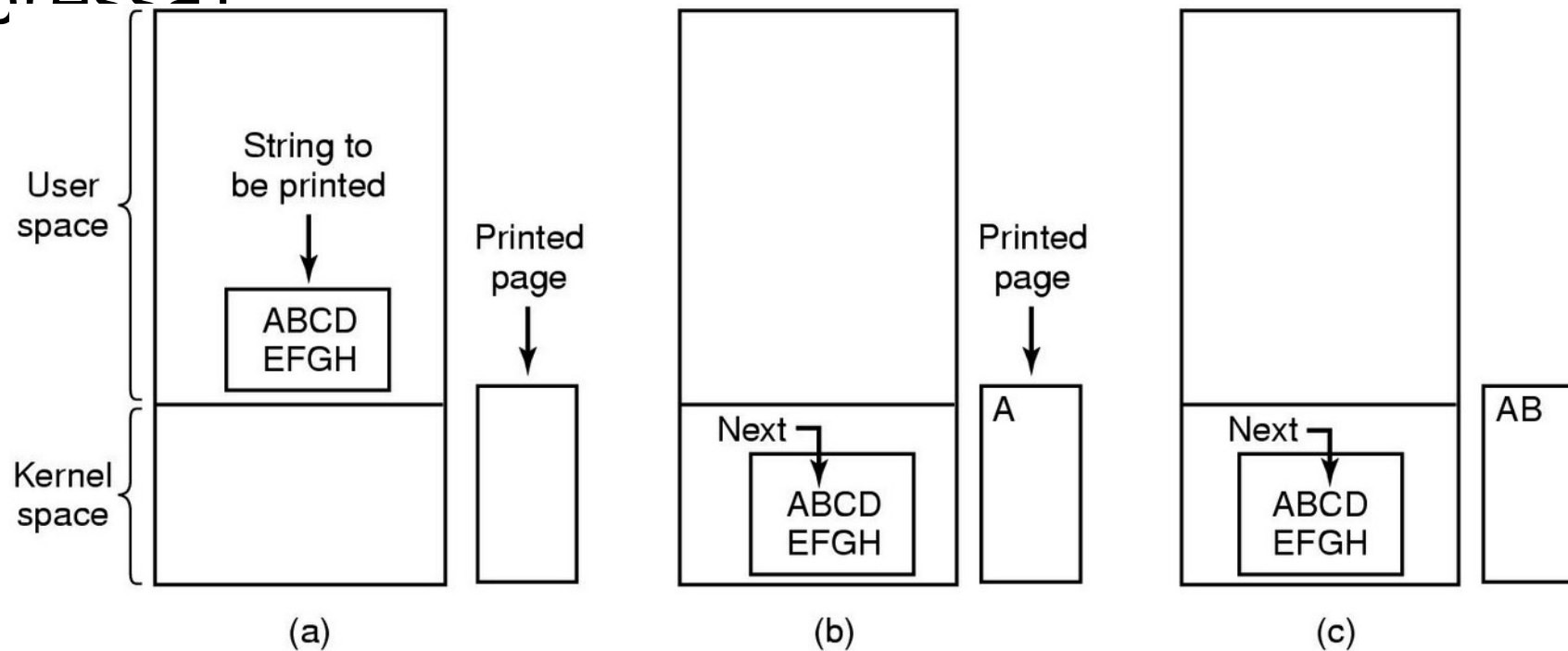
- E/S programada;
  - Mais usada em sistemas embarcados/embutidos;
- E/S orientada à interrupção;
- E/S com uso da DMA;



# Dispositivos de E/S

## Maneiras de realizar E/S

- E/S programada: passos para impressão de uma cadeia de caracteres (laço até que toda a cadeia tenha sido impressa):



# Dispositivos de E/S

## Maneiras de realizar E/S

- E/S programada:
  - Desvantagem:
    - CPU é ocupada o tempo todo até que a E/S seja feita;
    - CPU continuamente verifica se o dispositivo está pronto para aceitar outro caracter → espera ocupada;

# Dispositivos de E/S

## Maneiras de realizar E/S

- E/S orientada à interrupção:
  - No caso da impressão, a impressora não armazena os caracteres;
  - Quando a impressora está pronta para receber outros caracteres, gera uma interrupção;
  - Processo é bloqueado;

# Dispositivos de E/S

## Maneiras de realizar E/S

- E/S com uso da DMA:
  - DMA executa E/S programada → controladora de DMA faz todo o trabalho ao invés da CPU;
    - Redução do número de interrupções;
  - Desvantagem:
    - DMA é mais lenta que a CPU;

# Dispositivos de E/S

## Maneiras de realizar E/S

- DMA (*Direct Access Memory*) → acesso direto à memória:
  - Presente principalmente em dispositivos baseados em bloco → discos;
    - Controladora integrada à controladora dos discos;
  - Pode estar na placa-mãe e servir vários dispositivos → controladora de DMA independente do dispositivo;
  - DMA tem acesso ao barramento do sistema independentemente da CPU;

# Dispositivos de E/S

## Princípios de Hardware

- DMA contém vários registradores que podem ser lidos e escritos pela CPU:
  - Registrador de endereço de memória;
  - Registrador contador de bytes;
  - Registrador (es) de controle;
    - Porta de E/S em uso;
    - Tipo da transferência (leitura ou escrita);
    - Unidade de transferência (byte ou palavra);
    - Número de bytes a ser transferido;

# Dispositivos de E/S

## Princípios de Hardware

- Sem DMA: Leitura de um bloco de dados em um disco:
  - Controladora do dispositivo lê bloco (bit a bit) a partir do endereço fornecido pela CPU;
  - Dados são armazenados no *buffer* da controladora do dispositivo;
  - Controladora do dispositivo checa consistência dos dados;
  - Controladora do dispositivo gera interrupção;
  - SO lê (em um *loop*) os dados do *buffer* da controladora do dispositivo e armazena no endereço de memória fornecido pela CPU;

# Dispositivos de E/S

## Princípios de Hardware

- Com DMA: Leitura de um bloco de dados em um disco: CPU controla
  - 1. Além do endereço a ser lido, a CPU fornece à controladora de DMA duas outras informações: endereço na RAM para onde transferir os dados e o número de bytes a ser transferido;
  - 2. Controladora de DMA envia dados para a controladora do dispositivo;
  - Controladora do dispositivo lê o bloco de dados e o armazena em seu *buffer*, verificando consistência;
  - 3. Controladora do dispositivo copia os dados para RAM no endereço especificado na DMA (modo direto);



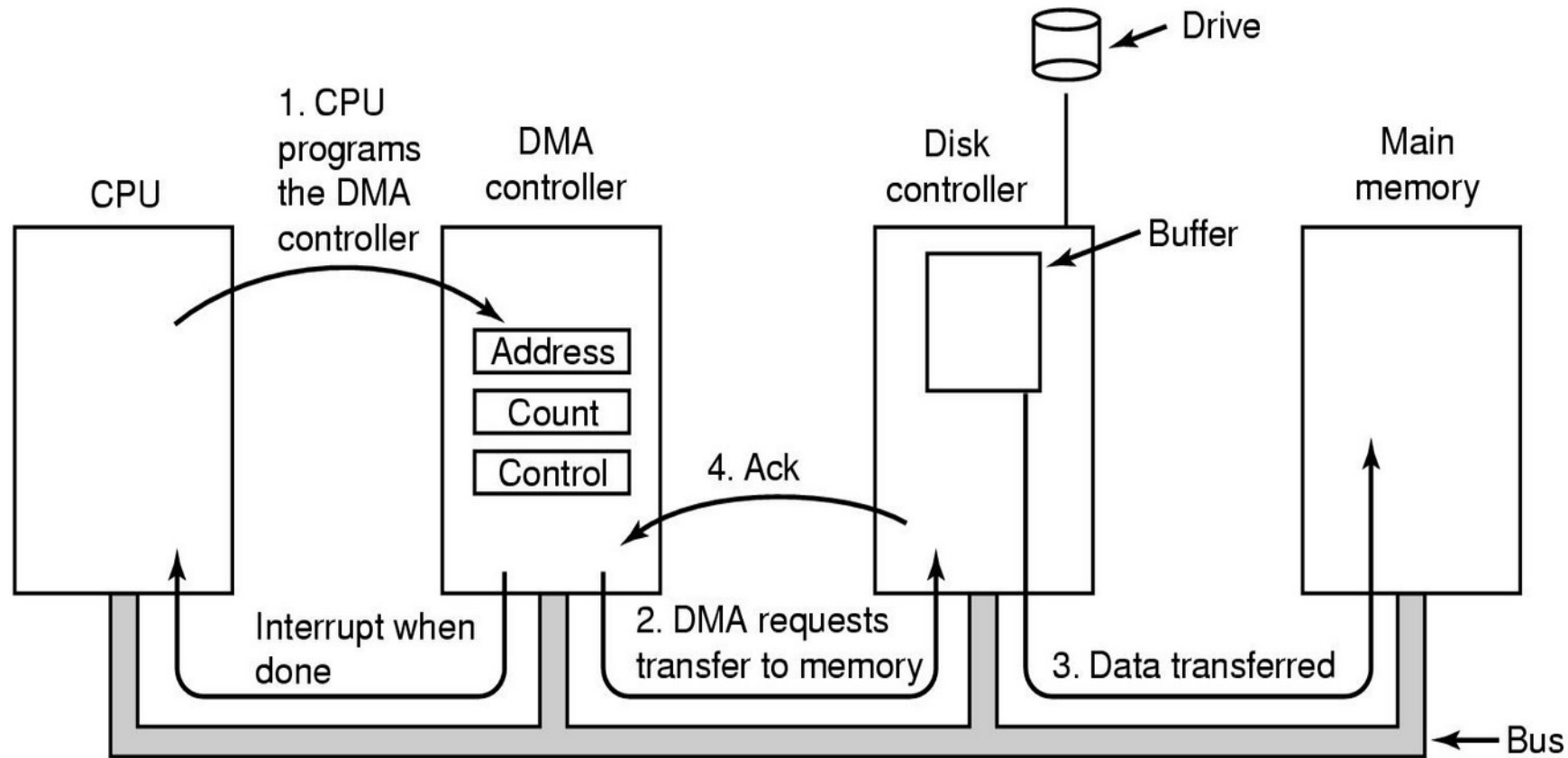
# Dispositivos de E/S

## Princípios de Hardware

- 4. Após confirmação de leitura, a controladora de DMA **incrementa o endereço de memória** na DMA e **decrementa o contador** da DMA com o número de bytes transferidos;
- Repete os passos de 2 a 4 até o contador da DMA chegar em 0. Assim que o contador chegar em zero (0), a controladora de DMA gera uma interrupção avisando a CPU;
- Quando o SO inicia o atendimento à interrupção, o bloco de dados já está na RAM;

# Dispositivos de E/S

## Princípios de Hardware



# Dispositivos de E/S

## Princípios de Hardware

- A DMA pode tratar múltiplas transferências simultaneamente:
  - Possuir vários conjuntos de registradores;
  - Decidir quais requisições devem ser atendidas → escalonamento (*Round-Robin* ou prioridades, por exemplo);

# Dispositivos de E/S

## Princípios de Hardware

- Por que a controladora de DMA precisa de um *buffer* interno? Por que não escreve diretamente na RAM?
  - Permite realizar consistência dos dados antes de iniciar alguma transferência;
  - Dados (bits) são transferidos do disco a uma taxa constante, independentemente da controladora estar pronta ou não;
  - Acesso à memória depende de acesso ao barramento, que pode estar ocupado com outra tarefa;
  - Com o *buffer*, o barramento é usado apenas quando a DMA opera;