

# SISTEMAS OPERACIONAIS

## AULA 18 – GERENCIAMENTO DE ENTRADA/SAÍDA, PARTE I

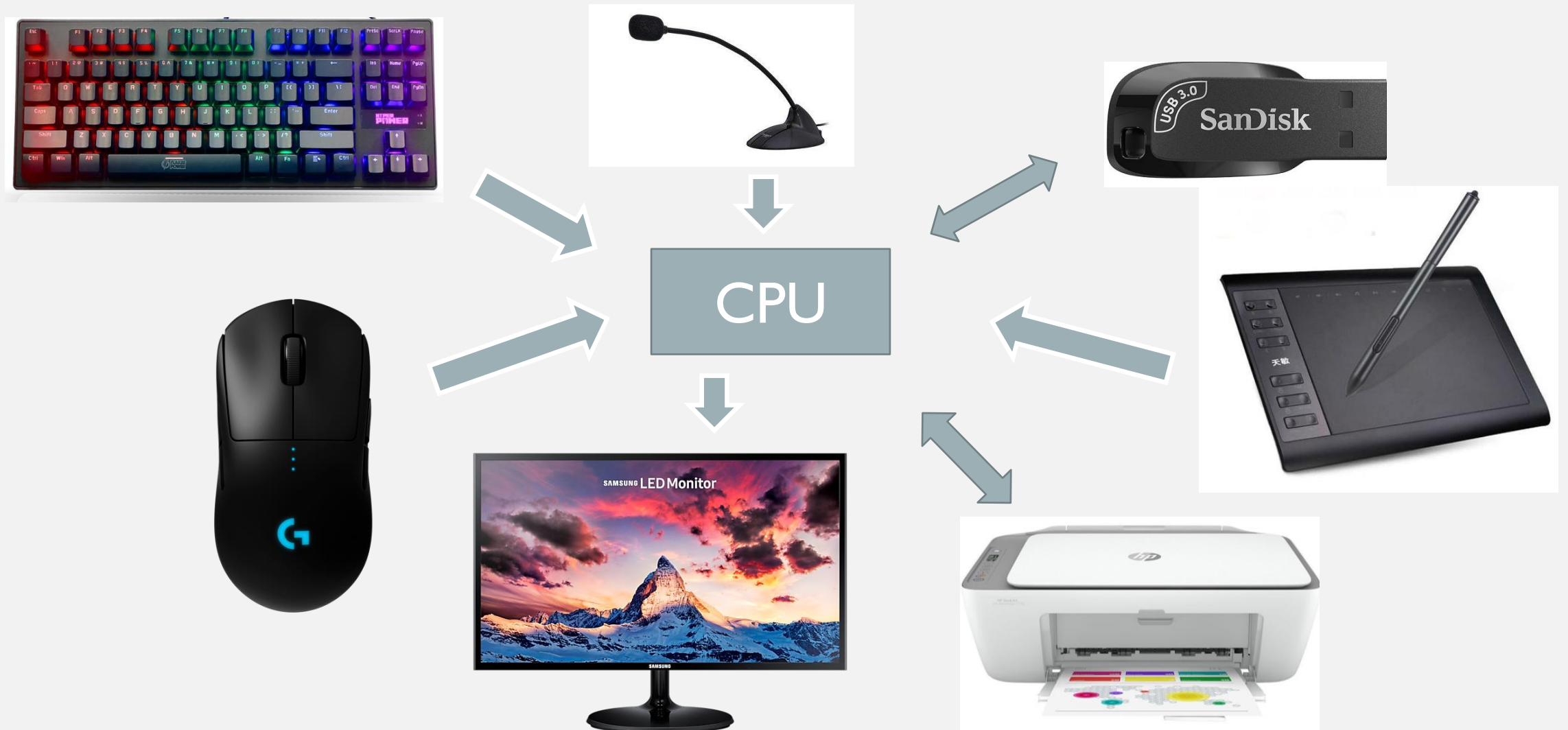
Prof.<sup>a</sup> Sandra Cossul, Ma.



## E/S - INTRODUÇÃO

- A função do sistema operacional no contexto de entrada e saída de dados é:
  - **controlar as operações de E/S**
  - **controlar os dispositivos de E/S**
- Os dispositivos de E/S permitem a interação do computador com o mundo exterior de várias formas.

# EXEMPLOS DE DISPOSITIVOS DE E/S



## E/S - INTRODUÇÃO

- Hoje, dispositivos de E/S dos mais diversos tipos podem estar conectados a um computador
- A grande **diversidade de dispositivos periféricos** é um dos maiores desafios presentes na construção e manutenção de um sistema operacional, pois **cada um deles tem especificidades e exige mecanismos de acesso específicos.**

# E/S - INTRODUÇÃO

Um *smartphone* com  
seus dispositivos de E/S:



## E/S - INTRODUÇÃO

- Funções específicas do SO:
  - **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** (que seja simples e fácil de usar)

## DISPOSITIVOS DE E/S - CLASSIFICAÇÃO

- **Interface humana:** utilizados para se comunicar com o usuário do computador (ex.: impressoras, monitores, teclado, fone de ouvido, mouse, etc.)
- **Armazenamento:** utilizados para armazenamento de dados (ex.: HD, SSD, pen-drives, etc.)
- **Comunicação:** utilizados para transmissão de dados (ex.: conexões de rede – modems, bluetooth, etc.)

## E/S - CARACTERÍSTICAS

- **Taxa de transferência** – dispositivos tem diferentes taxas de transferência de dados
- **Aplicação** – o uso para o qual o dispositivo será utilizado
- **Complexidade de controle** – complexidade da interface de controle (impressora mais simples que HD)
- **Unidade de transferência** – dados podem ser transferidos como uma sequência de bytes ou caracteres ou em blocos
- **Representação dos dados** – diferentes métodos de codificação podem ser utilizados pelos dispositivos
- **Condições de erro** – tipos de erros, forma como os erros são reportados



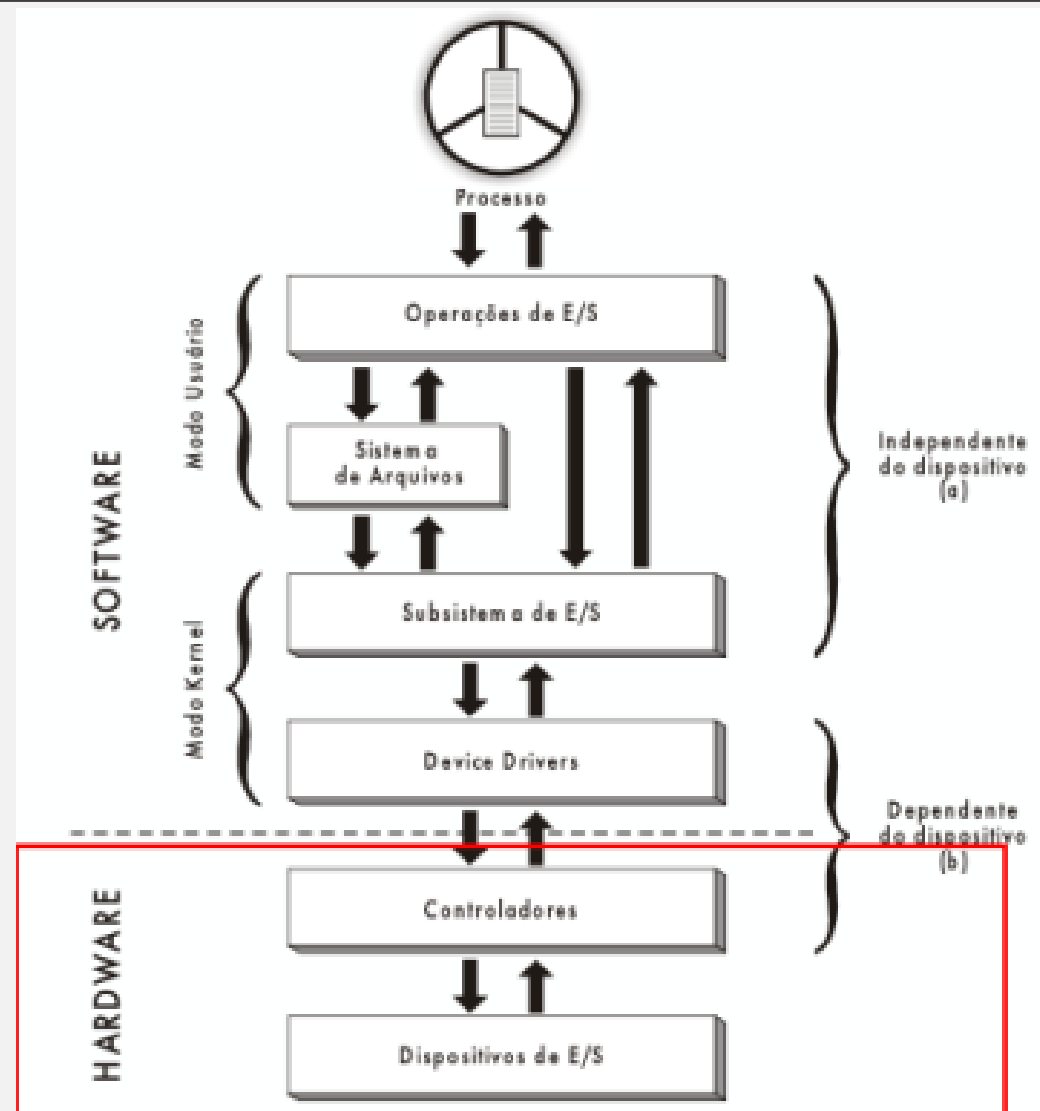
## E/S – VISÃO GERAL

- Como os dispositivos de E/S variam muito em sua **função** e **velocidade**, são necessários **diferentes métodos** fornecidos pelo SO para realizar o **controle destes dispositivos**
- **Padronização das interfaces de hardware e software** (o que ajuda a incorporar versões mais modernas dos dispositivos)
- **Aumento da variedade de dispositivos de E/S** (desafio de incorporar novos dispositivos ao computador e ao SO)

## E/S - HARDWARE

- **Portas** – são pontos de conexão do dispositivo ao computador
  - porta USB, porta HDMI, porta Ethernet, etc
- **Barramentos** – é um conjunto de linhas de comunicação que permitem a interligação entre dispositivos, como a CPU, a memória e outros periféricos
  - PCI express, SATA, USB, SAS, etc.
- **Controladores** – hardware que opera uma porta, um barramento ou um dispositivo
- O SO sempre trata com o controlador, não com os dispositivos.

# DISPOSITIVOS DE E/S

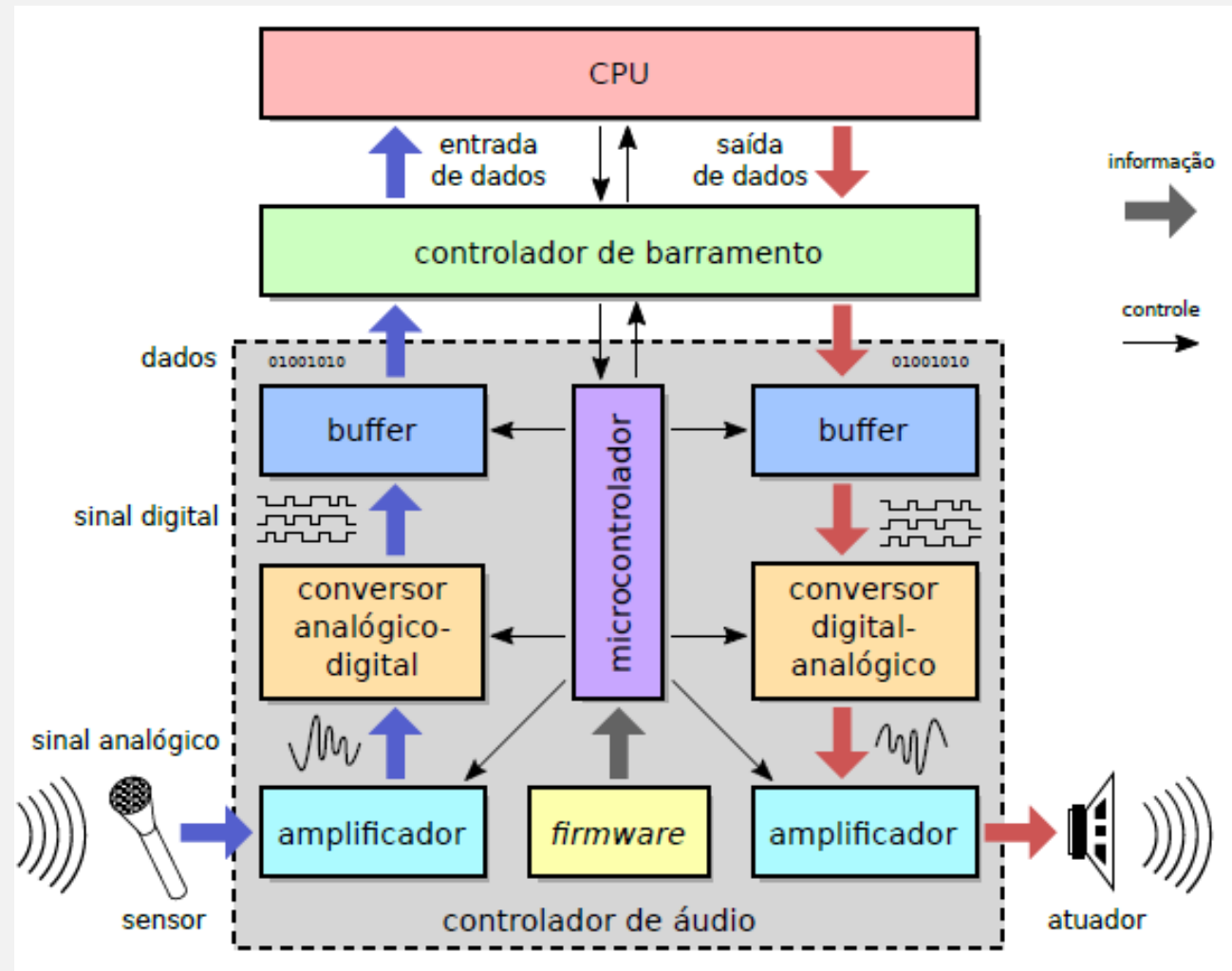


## COMPONENTES DE UM DISPOSITIVO DE E/S

- **Entrada de dados** – sensor capaz de converter uma informação externa em um sinal elétrico analógico
- **Conversor analógico-digital** – transforma a informação analógica recebida em informação digital (sequencia de bits)
- **Buffer** – armazena a informação digital que pode ser acessada pelo CPU através de um **controlador de entrada**
- **Saída de dados** – envio de dados do CPU a um **controlador de saída**, através do **barramento**
- **Conversor digital-analógico** – transforma a informação em um sinal elétrico analógico que será aplicado a um **atuador**

# COMPONENTES DE UM DISPOSITIVO DE E/S

Estrutura básica da E/S de áudio:



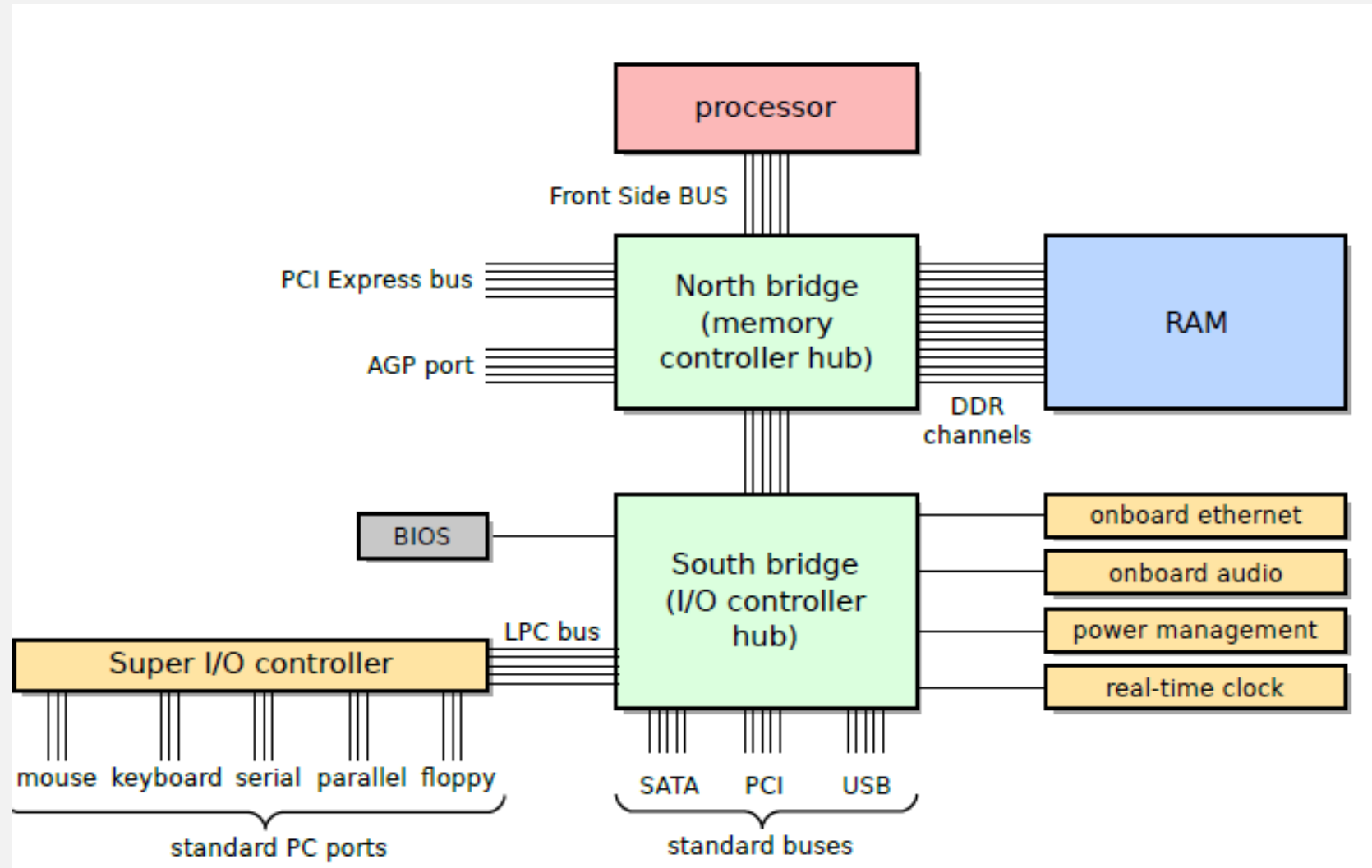
## COMPONENTES DE UM DISPOSITIVO DE E/S

- Alguns dispositivos possuem um processador ou microcontrolador interno para gerenciamento da operação.
- Esse processador embutido no dispositivo executa um código criado pelo fabricante do mesmo, denominado **firmware**.
- O **código do firmware** é independente do SO do computador e contém as instruções necessárias para operar o restante do hardware do dispositivo, permitindo realizar as operações solicitadas pelo SO.

## BARRAMENTOS DE ACESSO

- O acoplamento dos dispositivos de entrada/saída ao computador é feito através de barramentos
- É um conjunto de linhas de comunicação que permitem a interligação entre dispositivos, como a CPU, a memória e outros periféricos
- Controle dos barramentos em um sistema desktop é feito pelos controladores de hardware, parte do chipset da placa-mãe: *north-bridge* e *south-bridge*.

# BARRAMENTOS DE ACESSO





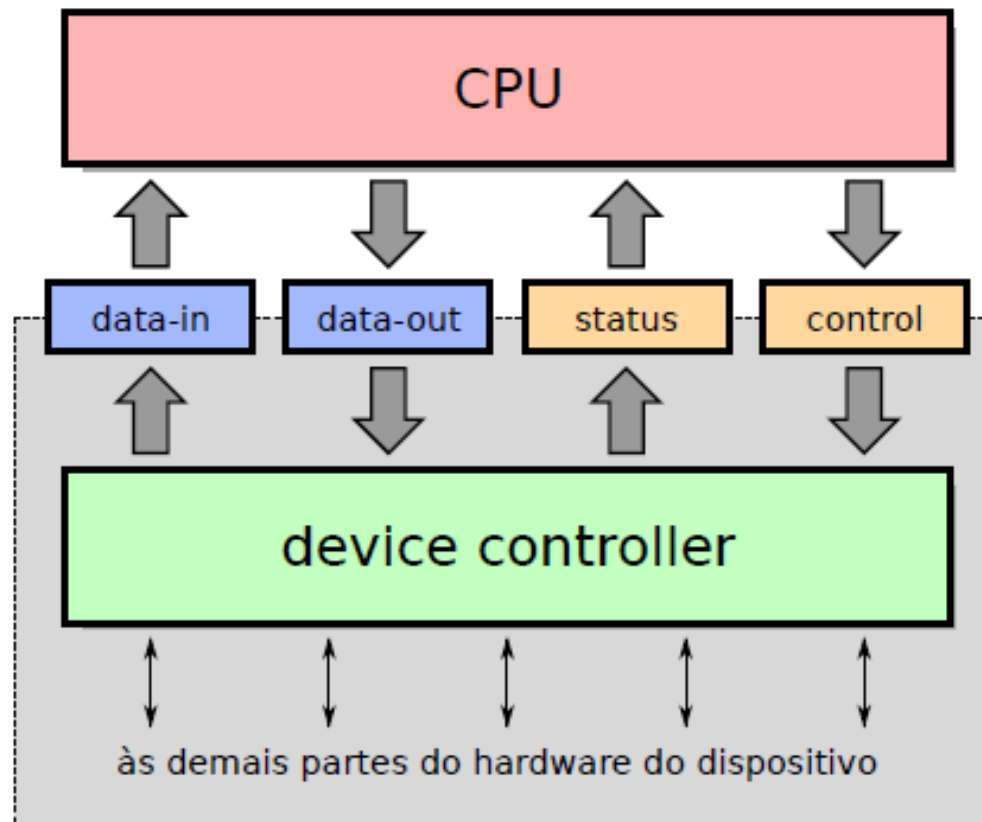
## DISPOSITIVOS E SUAS VELOCIDADES TÍPICAS DE TRANSMISSÃO

Dispositivo	velocidade
Teclado	10 B/s
Mouse ótico	100 B/s
Interface infravermelho (IrDA-SIR)	14 KB/s
Interface paralela padrão	125 KB/s
Interface de áudio digital S/PDIF	384 KB/s
Interface de rede <i>Fast Ethernet</i>	11.6 MB/s
<i>Pendrive</i> ou disco USB 2.0	60 MB/s
Interface de rede <i>Gigabit Ethernet</i>	116 MB/s
Disco rígido SATA 2	300 MB/s
Interface gráfica <i>high-end</i>	4.2 GB/s

# INTERFACE DE ACESSO

- Aspecto mais relevante de um dispositivo de E/S, do ponto de vista do SO
- Abordagem usada para acessar, configurar e enviar dados
- **Portas de entrada/saída** – conjunto de registradores acessíveis através do barramento usadas para a comunicação entre o dispositivo e o CPU
  - **saída** – enviar dados
  - **entrada** – receber dados
  - **status** – estado interno do dispositivo, verificação de erros
  - **controle** – envio de comandos ou configurações do CPU para o dispositivo

# INTERFACE DE ACESSO



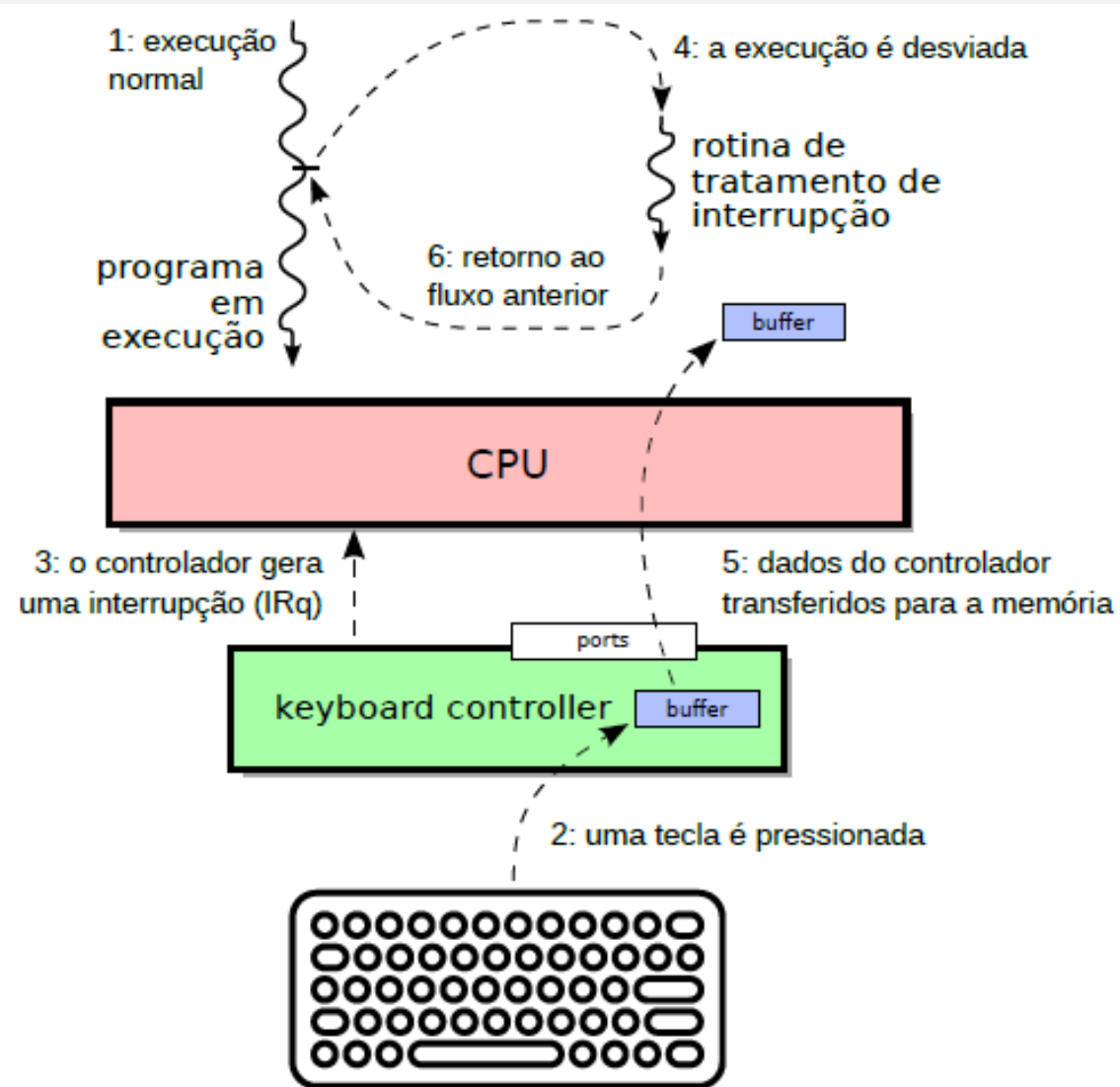
# INTERFACE DE ACESSO

- **Endereçamento de portas**
  - **Mapeada em portas** - as portas que compõem a interface são acessadas pelo processador através de instruções específicas para operações de E/S.
  - **Mapeada em memória** – uma parte não ocupada do espaço de endereços de memória é reservado para mapear as portas de acesso aos dispositivos. As portas são vistas como se fossem parte da memória principal e acessadas com as mesmas instruções de acesso à memória.
  - **Canais de E/S** – uso de um hardware independente com processador dedicado que comunica com o processador principal, através de um barramento específico. Ex.: interfaces de vídeo de alto desempenho (GPU).

# INTERRUPÇÕES

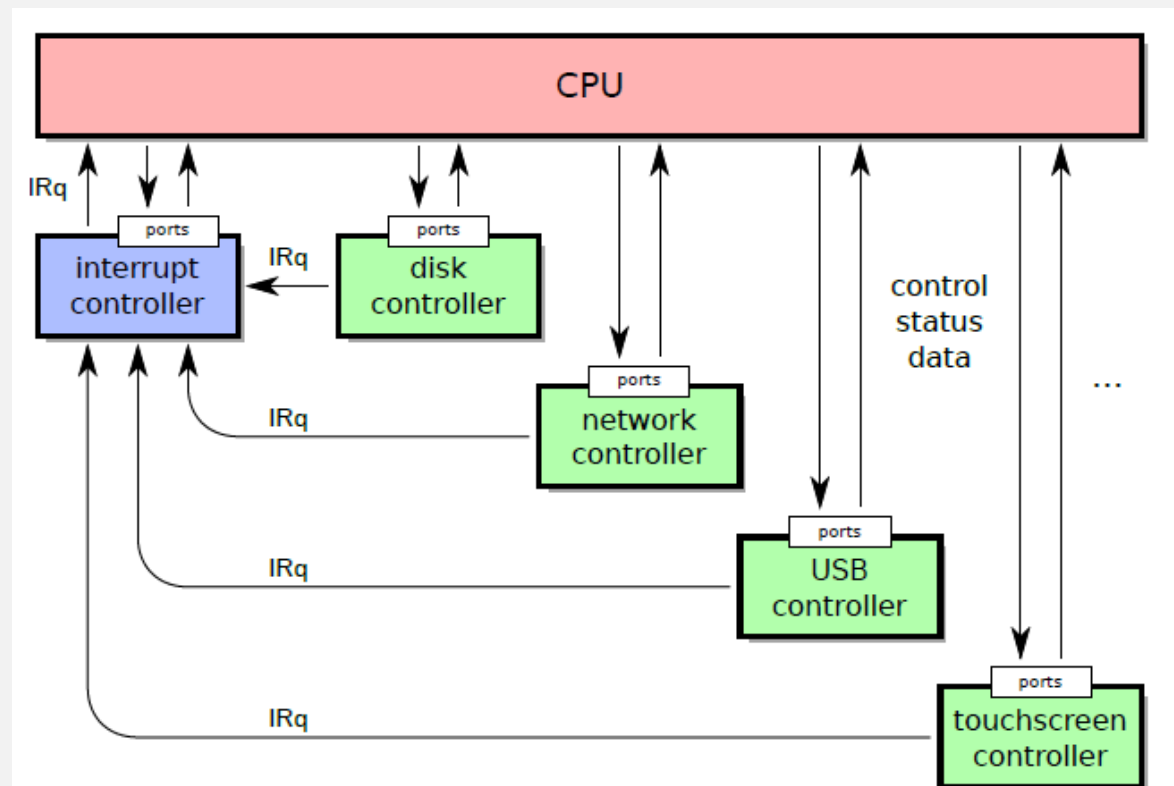
- **Forma do controlador notificar o processador sobre um evento interno** (clique de mouse, chegada de um pacote de rede, conclusão de uma operação de disco)
  - *Requisição de interrupção (IRq)*
  - Sinais enviados através do barramento de controle do computador
- Ao receber uma determinada requisição de interrupção, o processador suspende seu fluxo de instruções corrente e desvia a execução para um endereço predefinido, onde se encontra uma **rotina de tratamento de interrupção**.

# INTERRUPÇÕES



# INTERRUPÇÕES

- Nas arquiteturas de hardware atuais, as interrupções geradas pelos dispositivos de E/S não são transmitidas diretamente ao processador, mas a um **controlador de interrupções programável**, que faz parte do *chipset* do computador.

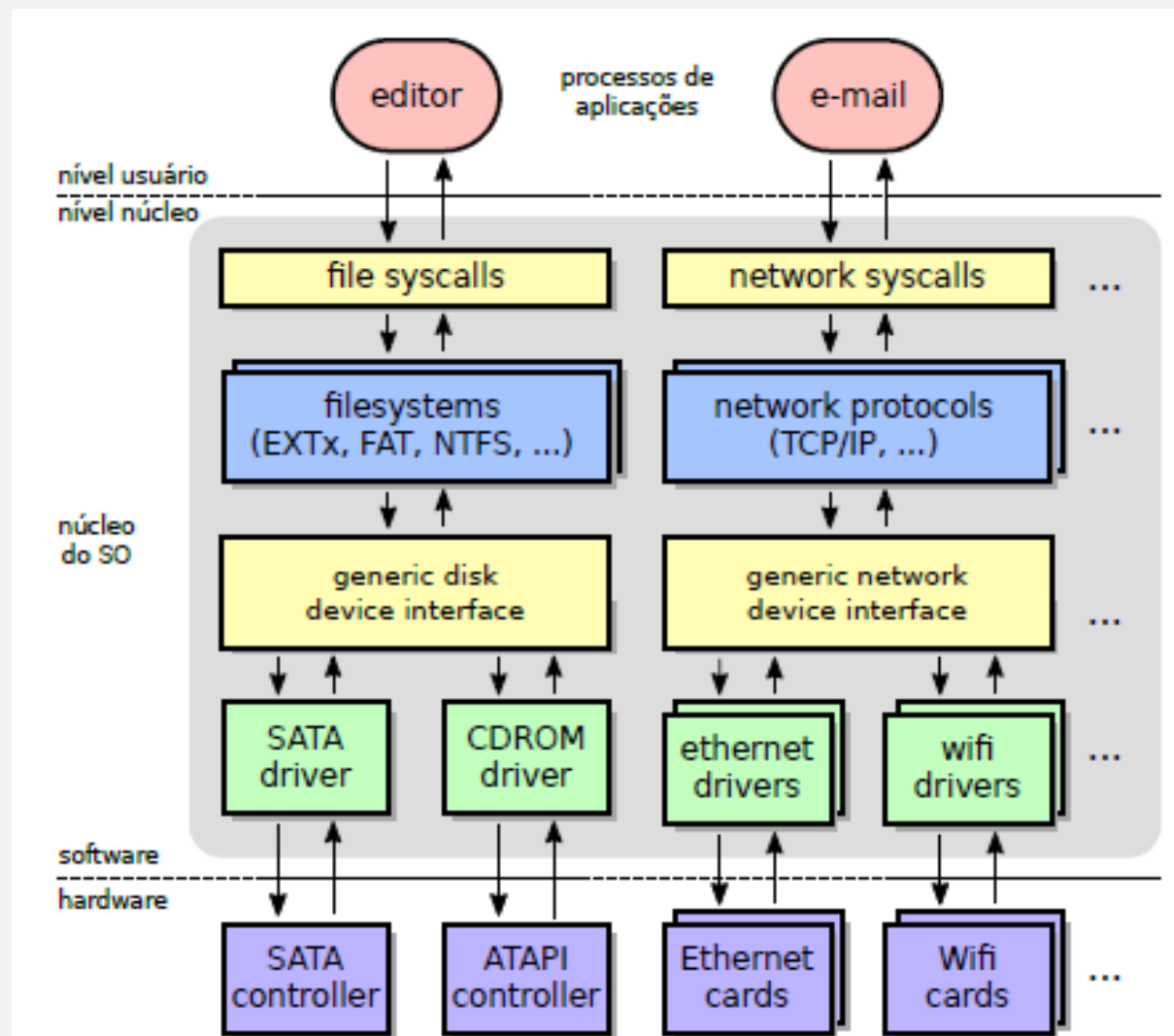


## E/S - SOFTWARE

- O SO deve prover **acesso eficiente, rápido e confiável** a um conjunto de periféricos com características diversas de comportamento, velocidade de transferência, volume de dados produzidos/consumidos e diferentes interfaces de hardware.
- O código do SO é estruturado em camadas
  - Camada inferior – interação direta com o hardware
  - Camada superior – interfaces de acesso às aplicações



# E/S - SOFTWARE



## E/S - SOFTWARE

- **Camada inferior** – controladores de dispositivos e aos controladores de DMA e de interrupções, implementados no *chipset* do computador
- **Primeira camada de software** – drivers de dispositivos
- **Segunda camada de software** – *generic device interface*, cuja finalidade é construir uma visão genérica de dispositivos similares de forma que o SO possa tratar os dispositivos por famílias ou classes
- **Terceira camada de software** – implementação de abstrações mais complexas, como sistemas de arquivos e protocolos de rede.
- **Topo da arquitetura de software** – implementação das chamadas de sistemas

## CLASSES DE DISPOSITIVOS

- Para **simplificar a construção de aplicações e das camadas mais elevadas do próprio SO**, os dispositivos de E/S são geralmente agrupados em **classes ou famílias com características similares**, para os quais uma interface genérica pode ser definida
- **Famílias:**
  - **Dispositivos orientados a caracteres** – fluxo contínuo e sequencial de E/S byte a byte. Não é possível alterar o valor de um byte que já foi enviado.
    - Ex.: Interfaces paralelas do computador: mouse, teclado

# CLASSES DE DISPOSITIVOS

- **Famílias:**
  - **Dispositivos orientados a blocos** – operações de E/S são feitas usando blocos de bytes de tamanho fixo. Os blocos são endereçáveis.
    - Ex.: discos rígidos e outros dispositivos de armazenamento
  - **Dispositivos de rede** – permitem enviar e receber mensagens entre processos e computadores distintos. Mensagens são blocos de dados de tamanho variável, com envio e recepção sequencial.
    - Ex.: Ethernet, Wifi, Bluetooth, GPRS
  - **Dispositivos gráficos** – permitem a renderização de texto e gráficos em terminais de vídeo. Exigem um alto desempenho, sobretudo para jogos e filmes, na transferência de dados. Uso de bibliotecas específicas como DirectX em ambientes Windows.

## DRIVERS

- Componentes de código que **interagem diretamente com cada controlador**, para realizar as operações de entrada/saída, receber as requisições de interrupção e fazer o gerenciamento do dispositivo correspondente.
- **Cada dispositivo de E/S** tem um **controlador de dispositivo** e um **driver de dispositivo específico** para se comunicar com o sistema operacional.

# DRIVERS

- Grupos de funções implementadas por um driver:
  - **Funções de E/S** – responsável pela transferência de dados entre o dispositivo e o SO, de acordo com a classe do dispositivo
  - **Funções de gerência** – gestão do dispositivo e do próprio driver (configuração, desligar, colocar em espera, tratar erros)
  - **Funções de tratamento de eventos** – funções ativadas quando uma requisição de interrupção é gerada pelo dispositivo
- Um driver mantém estruturas de dados locais, para **armazenar informações sobre o dispositivo e as operações em andamento.**

## DRIVERS

- Drivers normalmente executam dentro do núcleo do SO, em **modo privilegiado**
- Por ser **código de terceiros** executando com acesso total ao hardware, eles constituem um dos maiores **riscos à estabilidade e segurança do SO**
- Drivers mal construídos ou mal configurados são fontes frequentes de problemas como travamentos ou reinicializações inesperadas.

## METODOLOGIAS DE COMUNICAÇÃO

- **Cada driver deve interagir com seu respectivos dispositivo de entrada/saída** para realizar as operações desejadas, através das portas de seu controlador



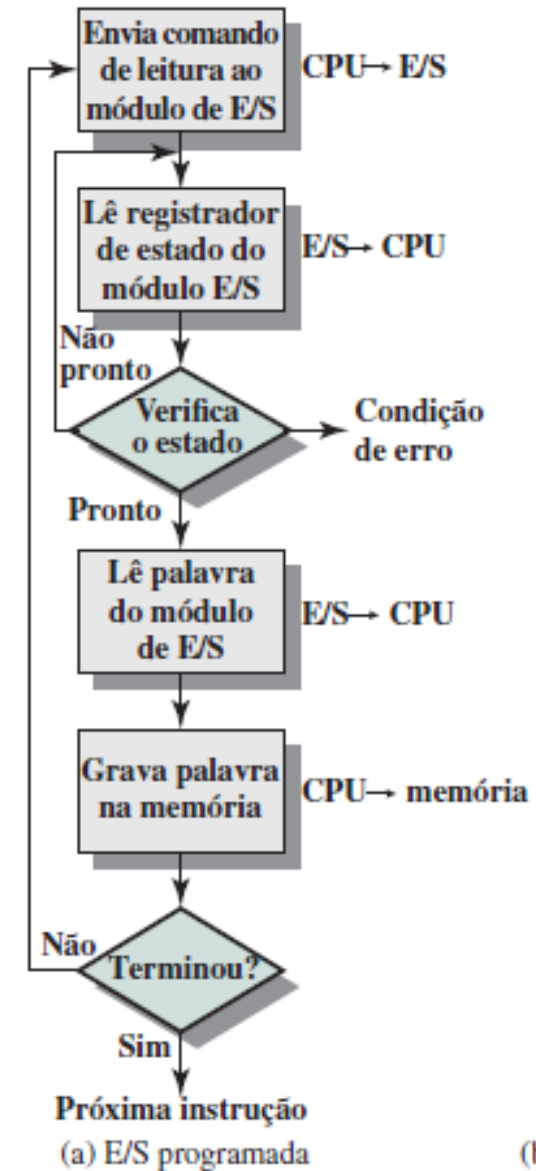
# METODOLOGIAS DE COMUNICAÇÃO

- 1. Entrada e saída programada (ou polling)**
  - 2. Entrada e saída por interrupção**
  - 3. Acesso direto à memória (DMA)**
- O que distingue as três formas?
    - participação da CPU
    - utilização das interrupções

# I. ENTRADA E SAÍDA PROGRAMADA (OU POLLING)

- Forma mais simples de E/S
- O driver solicita uma operação ao controlador do dispositivo e aguarda a conclusão da operação solicitada, monitorando continuamente os bits da respectiva porta de status.
- Os dados são trocados entre o **processador e o controlador**
- O **CPU tem controle direto** da operação de E/S
  - Verificação do estado de dispositivo
  - Envio de um comando de leitura ou escrita
  - Transferência de dados

# I. ENTRADA E SAÍDA PROGRAMADA (OU POLLING)



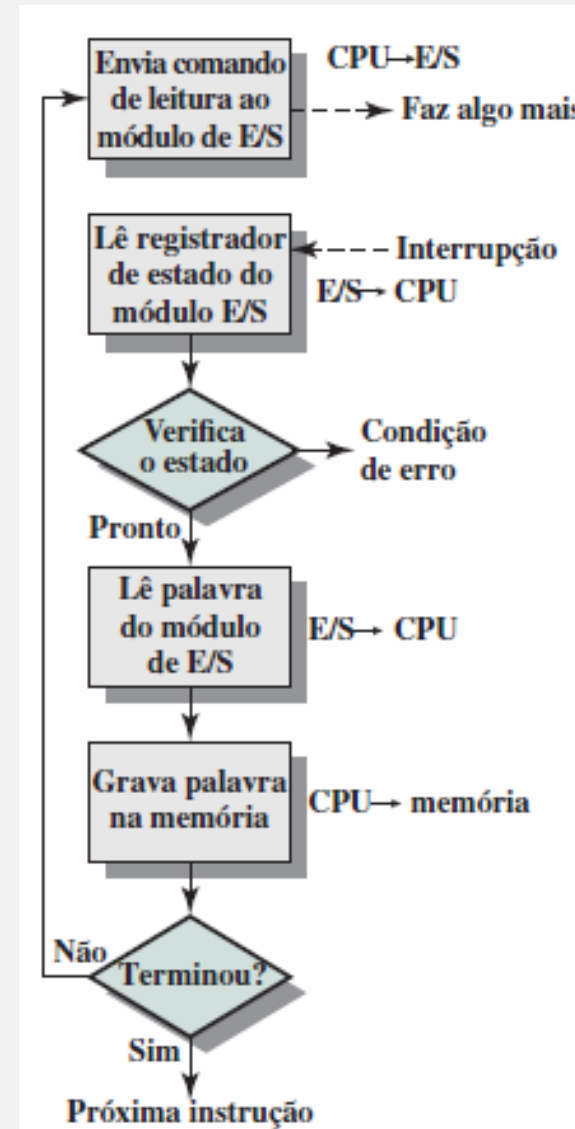
# I. ENTRADA E SAÍDA PROGRAMADA (OU POLLING)

- **Desvantagem:** processo demorado que mantém o CPU ocupado desnecessariamente; CPU é continuamente verificada para testar se o dispositivo está pronto para aceitar outro caractere – espera ocupada.
- Estratégia pouco usada em SOs de propósito geral
- Seu uso se concentra em sistemas embarcados dedicados, nos quais o CPU só tem uma tarefa (ou poucas tarefas) a realizar

## 2. ENTRADA E SAÍDA POR INTERRUPÇÃO

- O CPU emite um comando de E/S, continua a executar outras instruções e **é interrompido pelo módulo de E/S (controlador) quando este estiver pronto** para trocar dados com o CPU.
- O CPU, então, executará a transferência de dados, como antes, e depois retomará seu processamento anterior.
- Ainda **consome tempo do CPU**, pois cada palavra de dados que vem da memória para o módulo de E/S ou do módulo de E/S para a memória deve passar pelo CPU.

## 2. ENTRADA E SAÍDA POR INTERRUPTÃO



## 2. ENTRADA E SAÍDA POR INTERRUPÇÃO

- O maior problema no uso de interrupções: geralmente se dispõe de poucas linhas de interrupção ligadas diretamente ao processador
- SOs modernos utilizam um sistema de **interrupções por prioridade**
  - usualmente, são assinalados números para as interrupções, onde o menor número tem prioridade sobre o maior.

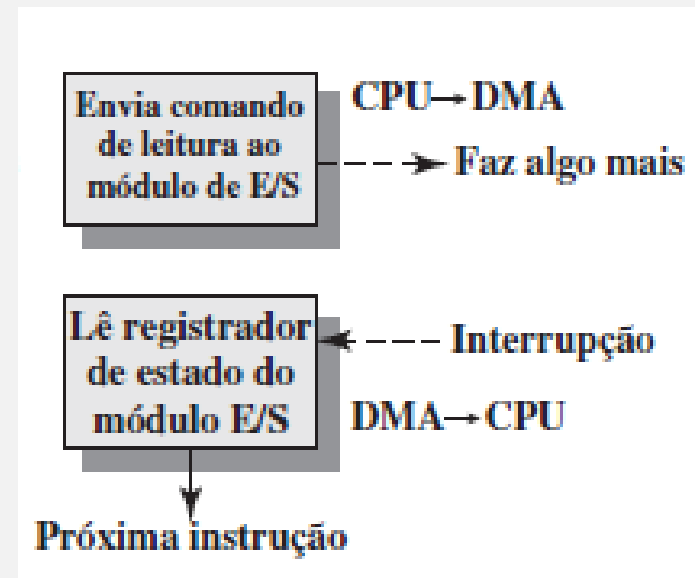
## TAXA DE TRANSFERÊNCIA DE E/S

- **Na E/S programada e na E/S por interrupção**, temos duas **desvantagens**:
  - A taxa de transferência de E/S é limitada pela velocidade com a qual o processador pode testar e atender a um dispositivo.
  - O processador fica ocupado no gerenciamento de uma transferência de E/S; diversas instruções precisam ser executadas para cada transferência de E/S.
- **Técnica mais eficiente**: acesso direto à memória (**DMA**)



### 3. ACESSO DIRETO À MEMÓRIA (DMA)

- Envolve um módulo adicional no barramento do sistema (**módulo de DMA**)
- **Módulo de DMA realiza por si só a transferência de dados** entre a memória principal e os controladores de E/S e o CPU fica liberado para realizar outras atividades.
- Quando o controlador DMA termina a transferência, ele avisa o CPU através de uma interrupção.
- DMA é independente do processador!



### 3. ACESSO DIRETO À MEMÓRIA (DMA)

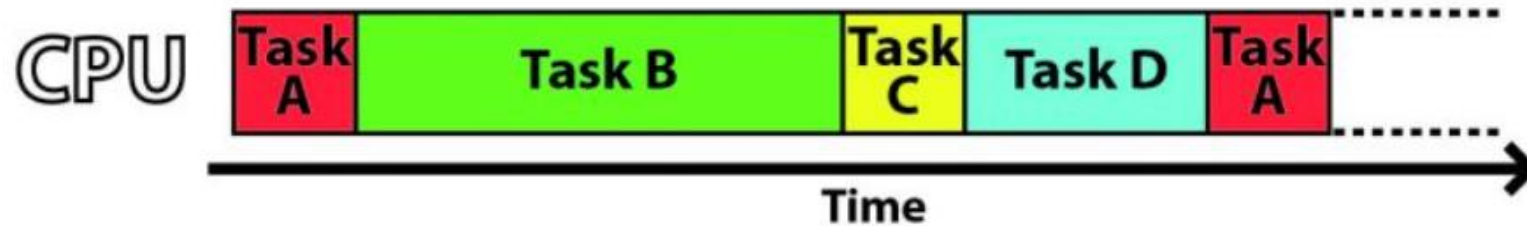


Figura 1 – CPU responsável por executar todas as Tasks.

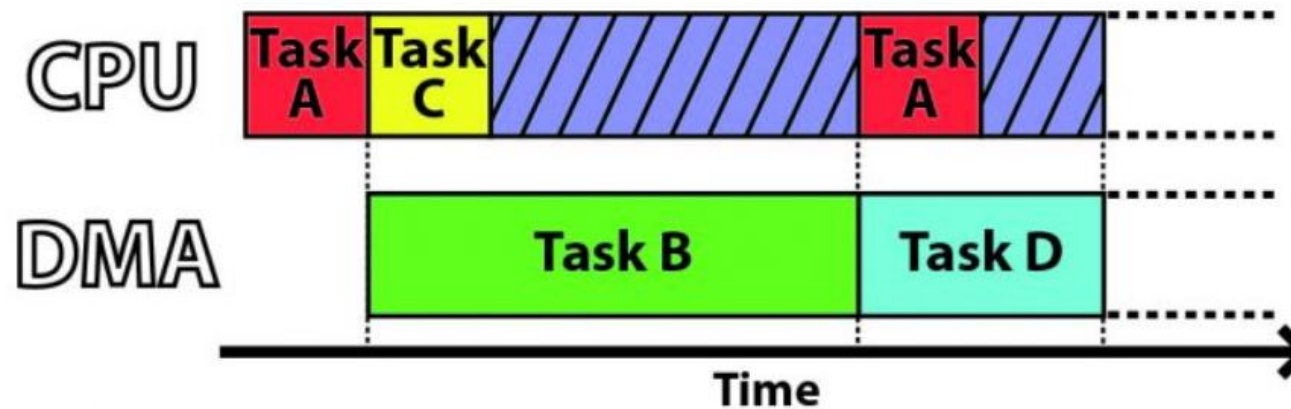
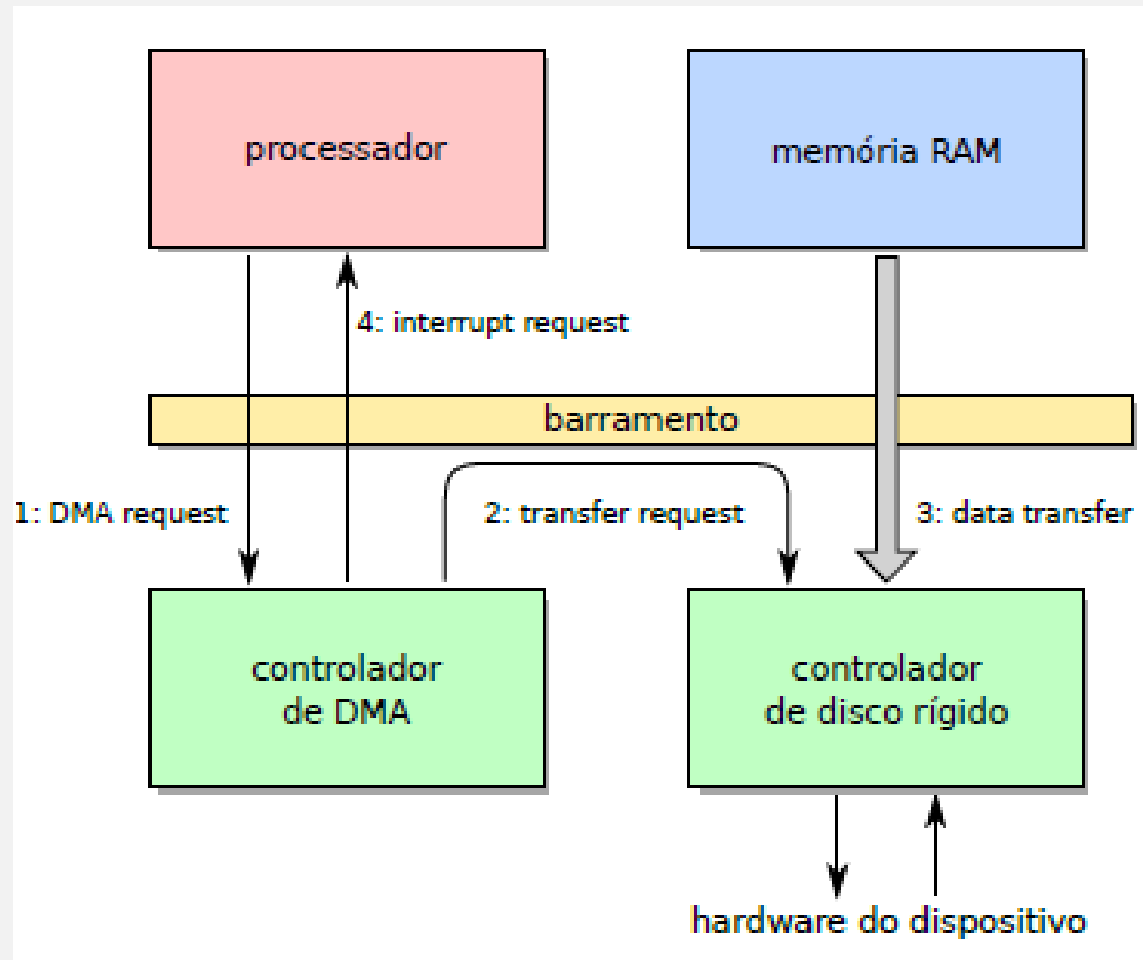


Figura 2 – Divisão de Tasks entre a CPU e o DMA.

### 3. ACESSO DIRETO À MEMÓRIA (DMA)



### 3. ACESSO DIRETO À MEMÓRIA (DMA)

- **Vantagens:**

- Transferir dados sem o envolvimento do processador vai acelerar a tarefa de E/S
- A implementação de DMA também reduz a sobrecarga do processador
- O controlador de DMA pode suportar, tipicamente, o trabalho com vários periféricos diferentes, cada um utilizando um canal de DMA (DMA channel)
- Pode ser implementada em hardware de diversas formas diferentes, conforme a quantidade de dispositivos e o desempenho pretendido

- **Desvantagens:**

- Por se tratar de uma unidade de hardware, existe um custo de implementação de um controlador DMA no sistema
- DMA é mais lenta que o CPU

## BIBLIOGRAFIA

- Tanenbaum, A. S. **Sistemas Operacionais Modernos**. Pearson Prentice Hall. 3<sup>rd</sup> Ed., 2009.
- Silberschatz, A; Galvin, P. B.; Gagne G.; **Fundamentos de Sistemas Operacionais**. LTC. 9<sup>th</sup> Ed., 2015.
- Stallings, W.; **Operating Systems: Internals and Design Principles**. Prentice Hall. 5th Ed., 2005.
- Oliveira, Rômulo, S. et al. **Sistemas Operacionais - VII - UFRGS**. Disponível em: Minha Biblioteca, Grupo A, 2010.