

# Sistemas Operacionais

Gerência de Dispositivos

---

# Gerência de Entrada e Saída

---

Um periférico é qualquer dispositivo conectado a um computador de forma a possibilitar a interação do computador com o mundo externo.

A comunicação com o mundo externo inclui:

- **Pessoas**, através de:
  - Teclado, mouse, monitor, caixa de som, etc.
- **Outros equipamentos**, através de:
  - Modems, placas de rede, etc.

# Gerência de Entrada e Saída

---

De acordo com o **sentido do fluxo** de informações definimos dispositivos:

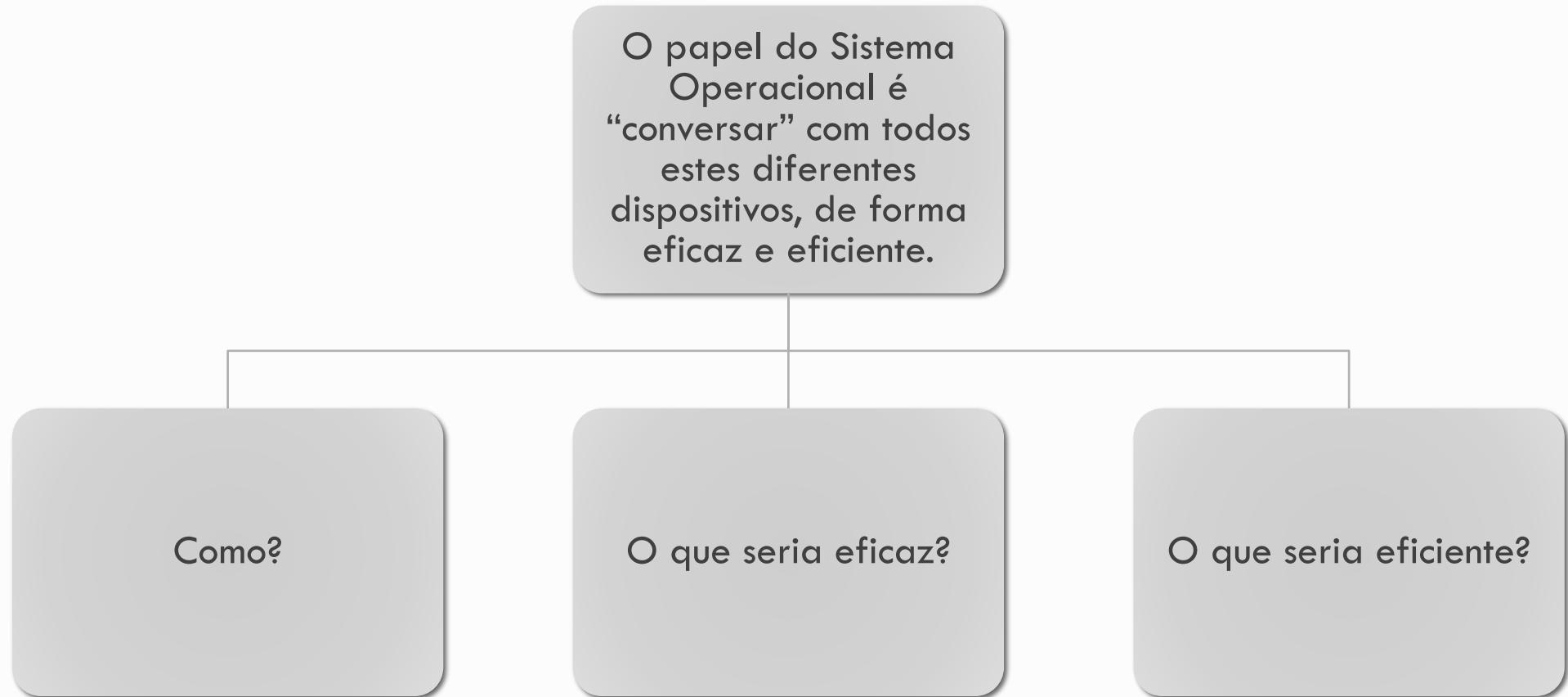
De entrada.

De saída.

De entrada e saída.

# Gerência de Entrada e Saída

---



# Gerência de Entrada e Saída

---

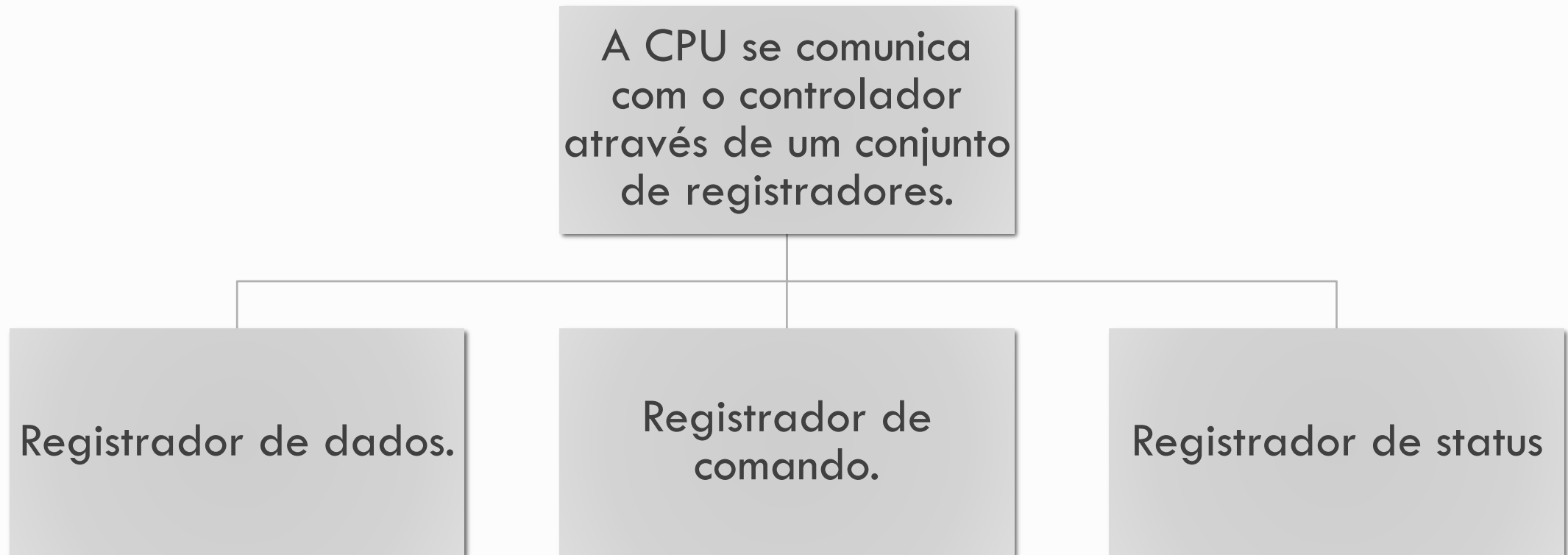
Um periférico é conectado ao computador através de um componente de hardware denominado interface.

Para tratar a diversidade, as interfaces utilizam controladores, que são projetados para periféricos específicos.

- Por exemplo, para um microprocessador enviar o endereço físico de dados para um disco rígido, o controlador (placa de circuitos anexa) traduz esse endereço e aciona os dispositivos mecânicos específicos do disco para que os dados possam ser enviados para o processador.

# Gerência de Entrada e Saída

---



# Gerência de Entrada e Saída

---

Exemplo de uma impressão:

- A CPU coloca a palavra *w* no registrador de dados.
- A CPU coloca o comando *write* no registrador de comando.
- O controlador aciona os dispositivos elétricos e/ou mecânicos necessários.
- O controlador escreve no registrador de status se a operação foi ou não bem sucedida.
- Uma interrupção é gerada para avisar a CPU do fim da operação.

# Gerência de Entrada e Saída

---

A função básica do controlador é implementar um conjunto de comandos/operações para o seu dispositivo específico.

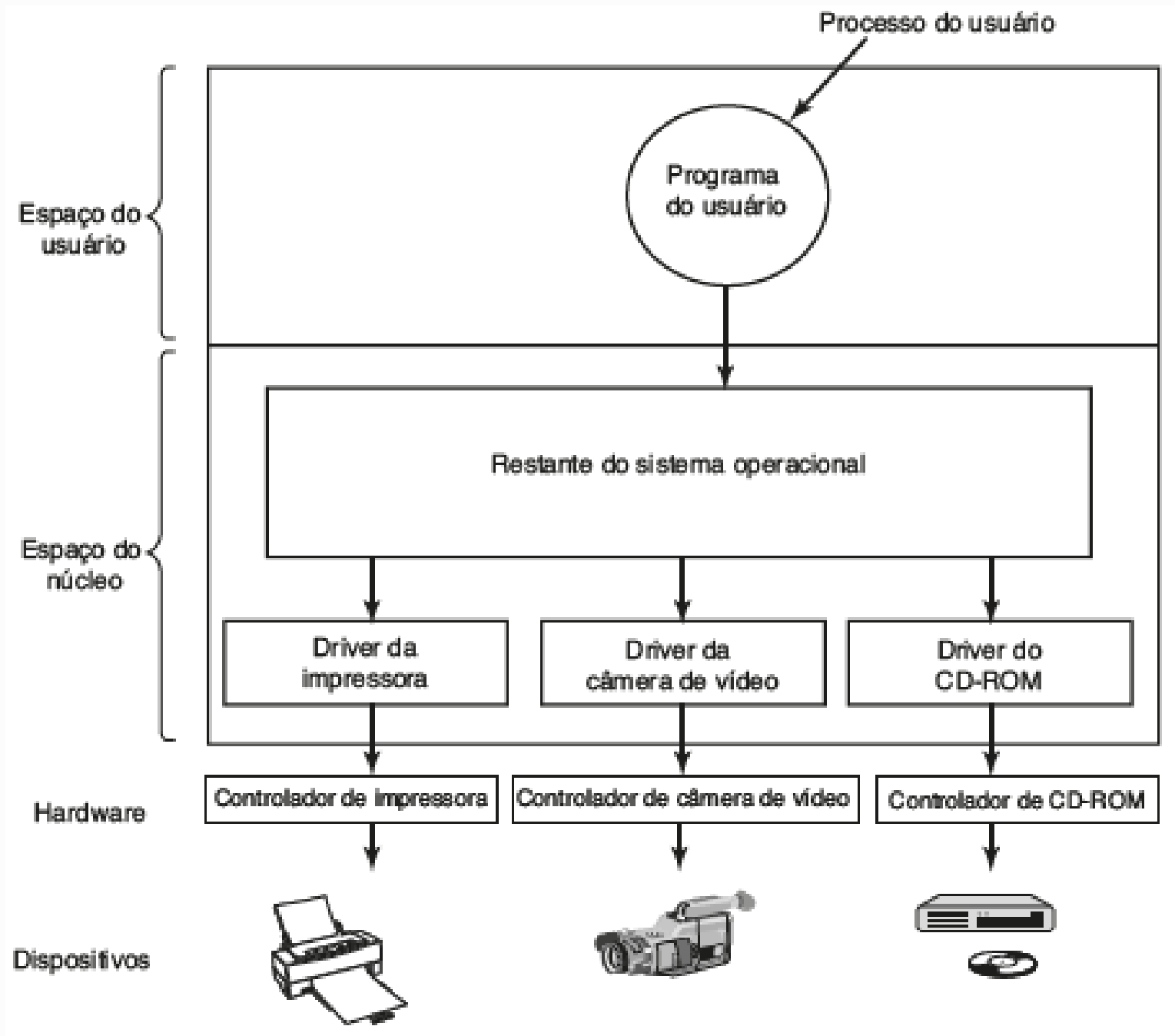
O controlador traduz cada ordem colocada no registrador de comandos numa sequência específica de acionamentos eletrônicos, elétricos e mecânicos.

Outras tarefas do controlador:

- converter fluxo serial de bits em bloco de bytes.
- executar toda correção de erro necessária (se possível).



# Gerência de Entrada e Saída



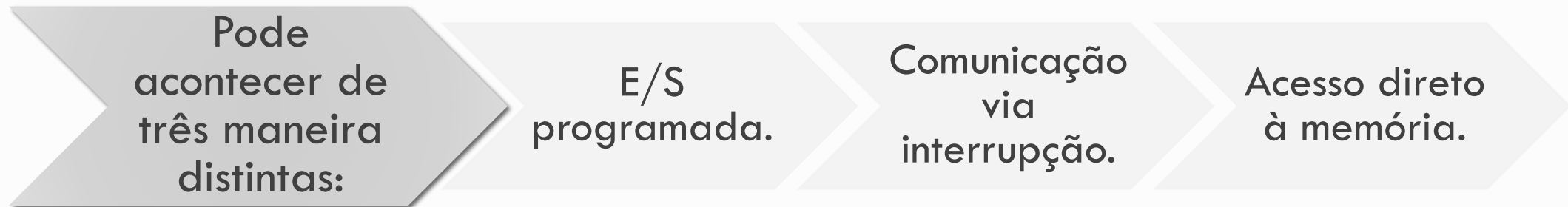
# Interação

Entre CPU e Controladores de Periféricos

# Gerência de Entrada e Saída

---

**Interação entre CPU e Controladores de Periféricos:**



# Interação

Entre CPU e Controladores de Periféricos

E/S Programada

# Gerência de Entrada e Saída

---

## **Interação entre CPU e Controladores de Periféricos:**

### E/S programada:

- Técnica usada quando não existe sistema de interrupção.
- Ou seja, não é possível avisar a CPU que uma operação finalizou, por exemplo.
- Toda a interação é de responsabilidade exclusiva do programador.

# Gerência de Entrada e Saída

---

## Interação entre CPU e Controladores de Periféricos:

### E/S programada:

- **Ex.:** O processador envia uma requisição na rede e fica testando continuamente se os pacotes chegaram.
- Famosa ESPERA OCUPADA.
- Problema similar a solução de Peterson, ao garantir apenas um processo na região crítica.

# Gerência de Entrada e Saída

---

## **Interação entre CPU e Controladores de Periféricos:**

### E/S programada:

- A perda de tempo de CPU pode ser minimizada se outras tarefas são executadas entre um teste e outro (enquanto o periférico não responde).
- Mas nem toda aplicação possui esta natureza.
- Isso acaba retirando o tempo de processamento de outros processos, pois o processo atual não dorme.

# Interação

Entre CPU e Controladores de Periféricos

Comunicação via Interrupção



# Gerência de Entrada e Saída

---

## **Interação entre CPU e Controladores de Periféricos:**

### Comunicação via Interrupção:

- Neste caso, a CPU é responsável apenas por iniciar uma operação de E/S.
- Quando a operação de E/S termina, o controlador interrompe o processador para que o mesmo acesse dados, ou verifique o status da operação.
- O processo dorme até o momento exato.
- Não existe espera ocupada neste caso.

# Gerência de Entrada e Saída

---

## **Interação entre CPU e Controladores de Periféricos:**

A utilização de interrupções resolve o problema de determinar o momento exato em que o dispositivo necessita de atenção do processador.

Mas os processos não irão acessar os registradores dos dispositivos.

- Eles acessam a memória virtual do computador.

# Gerência de Entrada e Saída

---

## **Interação entre CPU e Controladores de Periféricos:**

Então é necessário efetuar a transferência de dados.

- O processador precisa levar o dado do registrador da interface para a memória do computador.

Quando o volume de dados é considerável, o procedimento de interrupções se torna ineficiente porque precisa utilizar o processador para carregar os dados dos registradores para a memória.

# Gerência de Entrada e Saída

---

**Interação entre CPU e Controladores de Periféricos:**

Uma alternativa surgiu: **O acesso direto a memória**  
(feito pelos dispositivos).

# Interação

Entre CPU e Controladores de Periféricos

Acesso direto à memória

# Gerência de Entrada e Saída

---

## **Acesso direto à memória:**

Agora é possível transferir diretamente os dados da interface do dispositivo para a memória virtual.

Este procedimento é conhecido como DMA (*direct memory access*).

Ao terminar a cópia dos dados, pode invocar o processador através de uma interrupção.

# Gerência de Entrada e Saída

---

## **Acesso direto à memória:**

Apesar de concorrer com o processador no barramento para acessar a memória RAM e o disco, esta interação se mostra mais eficiente de modo geral.

- Pois nem sempre o processador está acessando a memória.

Significa uma computação paralela de operações.

- Enquanto o processador resolve outros problemas de outros processos, a interface insere dados na memória antes de causar uma interrupção.

# Princípios

## Básicos de Entrada e Saída



# Princípios básicos de Entrada e Saída

---

O módulo do S.O. que implementa o acesso a diferentes dispositivos deveria ser muito complexo:

- Devido a diversidade de elementos que ele deve tratar.

Desde o início, os projetistas se preocuparam em padronizar ao máximo a forma de acesso aos periféricos.

# Princípios básicos de Entrada e Saída

---

Como primeira melhoria, os programas de usuário deixaram de ser preocupar com o acesso direto aos dispositivos.

Mas isso não resolveu o problema de acessar diferentes dispositivos.

Para isso, foi proposto um módulo de 4 camadas.

# Princípios básicos de Entrada e Saída

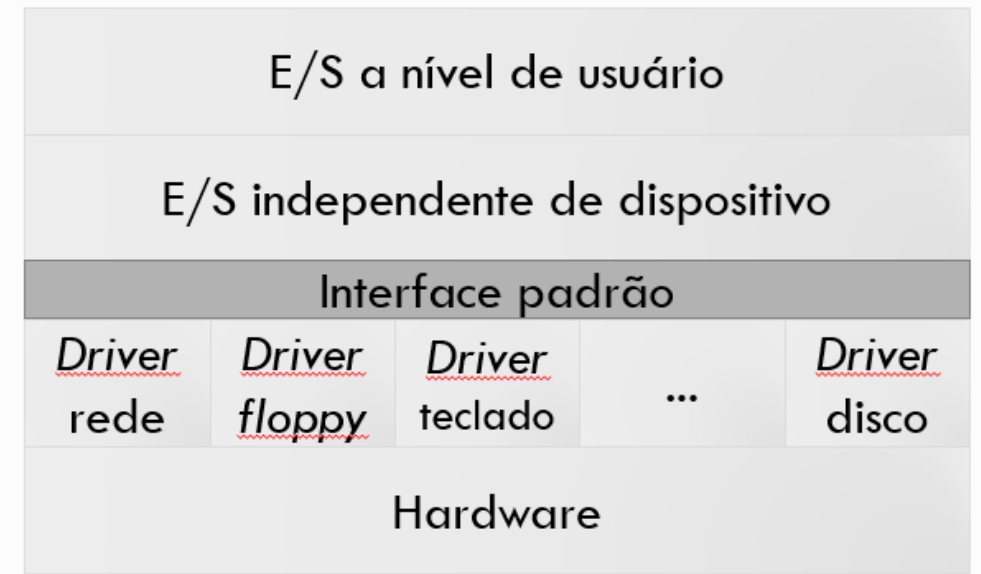
---



# Princípios básicos de Entrada e Saída

## Drivers de dispositivo:

- Responsáveis por implementar um acesso específico.
- Seu objetivo é ocultar a diferença entre vários dispositivos.
- Geralmente disponibilizados pelos fabricantes.
- O S.O. disponibiliza alguns drivers por padrão, mas que não garantem a utilização de todos os recursos do hardware.

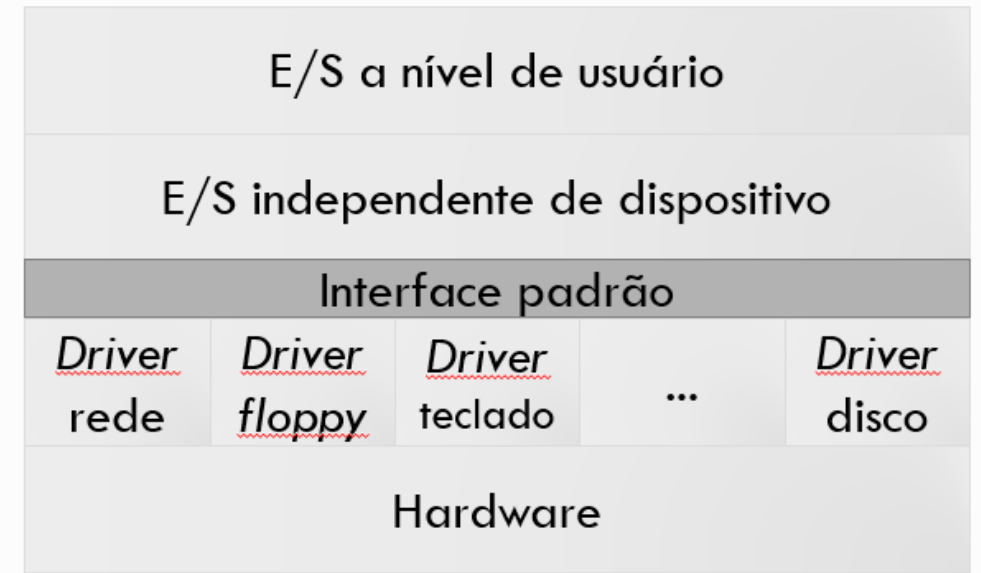


# Princípios básicos de Entrada e Saída

---

## E/S independente de dispositivo:

- Contem funções genéricas, que funcionarão até para dispositivos ainda não inventados!
- Exemplo:
  - Ler 512KB de um volume de dados. Não interessa qual é o tipo de volume. Disco, usb, cd-rom.
  - Quem deve se preocupar com isso é seu driver na camada inferior.

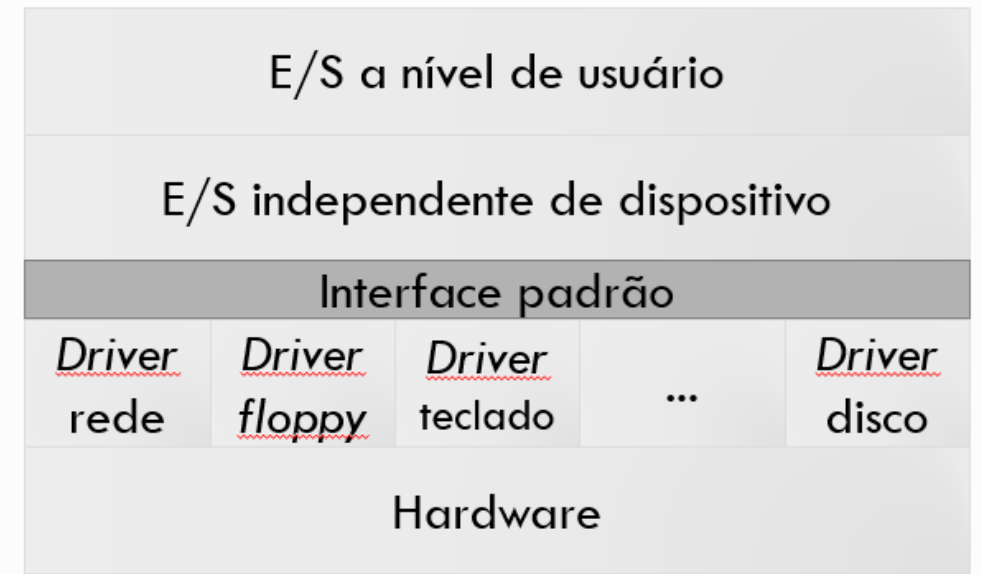


# Princípios básicos de Entrada e Saída

---

## E/S a nível de usuário:

- Tem acesso apenas aos recursos disponibilizados pelos compiladores.
  - Exemplo: *printf*.
- O programador não se preocupa se o arquivo vai ser escrito na USB, no HD ou no CD.
- As camadas mais internas fazem o redirecionamento e os devidos tratamentos.

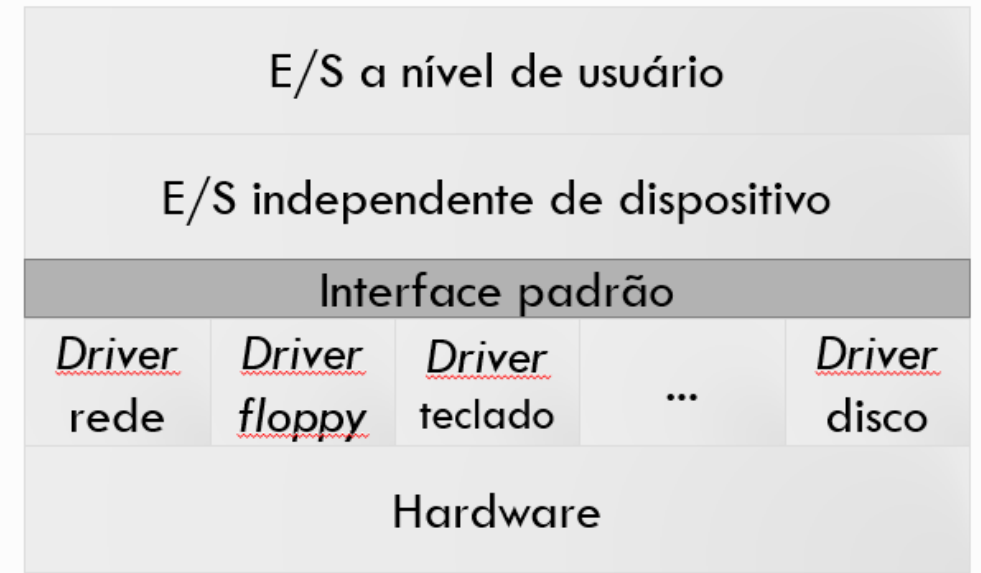


# Princípios básicos de Entrada e Saída

---

## E/S a nível de usuário:

- O compilador é responsável em oferecer uma biblioteca de E/S para cada sistema em que seu compilador vá ser utilizado.
- Esta biblioteca é que faz as chamadas ao S.O.



# Referências

---

Sistemas Operacionais Modernos. Tanenbaum, A. S. 2ª edição. 2003.

Sistemas Operacionais. Conceitos e Aplicações. A. Silberschatz; P. Galvin; G. Gagne. 2000.

Sistemas Operacionais – Projeto e Implementação. Tanenbaum, A. S. 2ª edição. 2000.

Slides Prof. Humberto Brandão