

# Sistemas Operacionais

Aula 12 – Gestão de Entrada/Saída – Hardware

Prof. Igor da Penha Natal

# Conteúdo

- 1 Dispositivos de E/S
- 2 Componentes de um dispositivo
- 3 Barramentos
- 4 Interface de E/S
- 5 Endereçamento
- 6 Interrupções

# Dispositivos de entrada/saída

Interação da CPU com o mundo exterior.

Função: informações físicas  $\rightleftharpoons$  informações digitais.

Grande variedade de dispositivos:

- **Interação** com o usuário: *mouse*, teclado, tela, fones
- **Armazenamento** de dados: discos, SSDs, CDROMs
- **Impressão e captura** de dados: impressora, scanner
- **Comunicação** com outros sistemas: *ethernet*, *Bluetooth*
- **Gerência** do sistema: tempo, energia, temperatura
- **Localização**: GPS, bússola, giroscópio
- ...

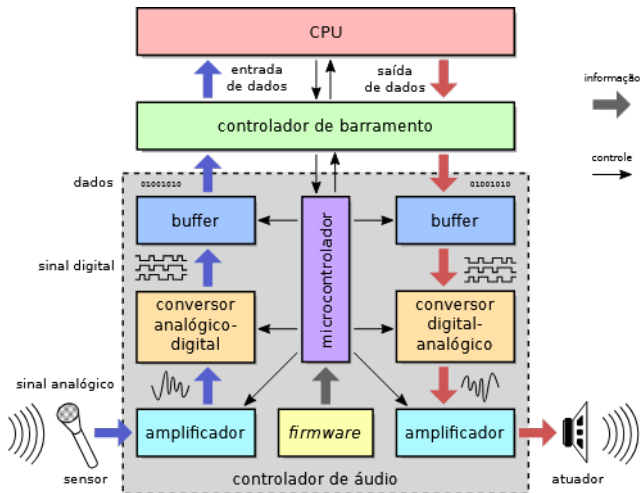
# Dispositivos de E/S em um *smartphone*



# Componentes de um dispositivo de E/S

- Entrada de dados:
  - **Sensor:** transforma grandeza física em sinal elétrico.
  - **Amplificador:** aumenta e limpa o sinal elétrico.
  - **Conversor AD:** converte o sinal em informação digital.
- Saída de dados:
  - **Conversor DA:** converte dados em sinais elétricos.
  - **Transdutor:** transforma sinais elétricos em ações físicas.
- Elementos comuns:
  - **Buffer:** armazena dados coletados e/ou enviados.
  - **Controlador de barramento:** permite acesso da CPU.
  - **Microcontrolador:** gerencia o hardware do dispositivo.
  - **Firmware:** código executado pelo controlador.

# Um dispositivo de E/S de áudio



# Barramentos

Barramento: via de comunicação:

- interliga CPU, memória e dispositivos de E/S
- parte do *chipset* da placa-mãe

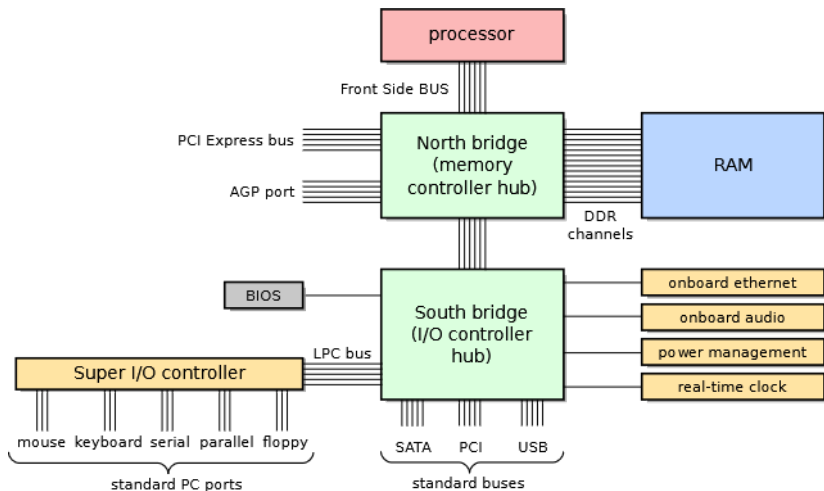
*North-bridge*: componentes rápidos

- CPU e RAM
- Portas AGP e PCI-express (vídeo)

*South-bridge*: componentes lentos

- PCI, USB, SATA
- BIOS, *legacy controllers*

# Barramentos em um PC típico





# Velocidades de transferência de dados

Dispositivo	velocidade
Teclado	10 B/s
Mouse ótico	100 B/s
Interface paralela padrão	125 KB/s
Interface de áudio digital S/PDIF	384 KB/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

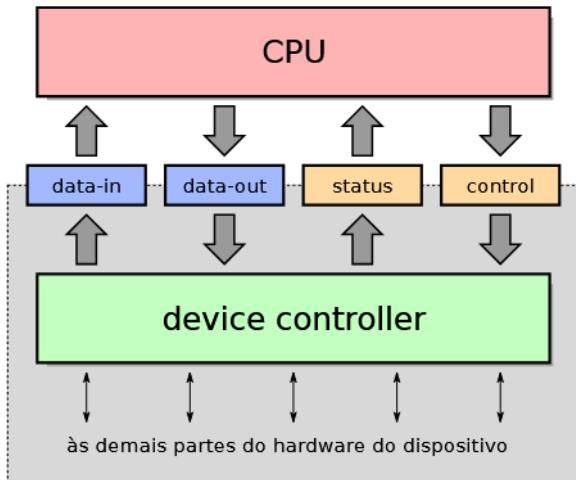
# Interface de acesso

Interação CPU  $\rightleftharpoons$  *device* feita por **portas de E/S**.

Existem 4 tipos básicos de portas:

tipo de porta	sentido	função
entrada	disp $\rightarrow$ CPU	Receber dados do dispositivo.
saída	CPU $\rightarrow$ disp	Enviar dados ao dispositivo.
status	disp $\rightarrow$ CPU	Consultar o <b>estado</b> do dispositivo; Verificar status de uma operação.
controle	CPU $\rightarrow$ disp	Enviar <b>comandos</b> ao dispositivo; Alterar configuração do dispositivo.

# Interface de acesso



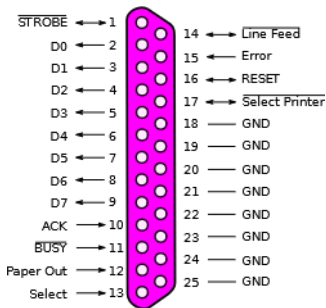
# Exemplo: interface paralela (SPP)

Interface muito antiga, já em desuso.

Usada para E/S em impressoras, scanners, plotadoras, etc.

Estrutura:

- Porta de dados (8 bits)
- Porta de status (8 bits)
- Porta de controle (8 bits)
- Portas dos modos estendidos



## Exemplo: interface paralela

$P_0$  (*data port*): porta de saída/entrada, 8 bits

$P_1$  (*status port*), 8 bits

- 0 reservado
- 1 reservado
- 2  $\overline{\text{nIRQ}}$ : se 0, gerou uma interrupção
- 3 error: se 1, há um erro interno na impressora
- 4 select: se 1, a impressora está pronta (*online*)
- 5 paper\_out: se 1, falta papel na impressora
- 6  $\overline{\text{ack}}$ : pulso em 0, dado foi recebido
- 7  $\overline{\text{busy}}$ : se 0, controlador está ocupado

## Exemplo: interface paralela

$P_2$  (control port):

- 0 strobe: pulso em 0, há um dado em  $P_0$
- 1 auto\_lf: se 0, *line feed* a cada *carriage return*
- 2 reset: se 1, reiniciar a impressora
- 3 select: se 0, a impressora foi selecionada
- 4 enable\_IRQ: se 1, permite gerar interrupções
- 5 bidirectional: se 1, ativa modo bidirecional
- 6 reservado
- 7 reservado

$P_3$  a  $P_7$ : usadas em modos estendidos (EPP e ECP)

# Funcionamento da interface paralela

## ■ Porta $P_0$ : dados da CPU

- Um byte por vez

## ■ Porta $P_1$ : status

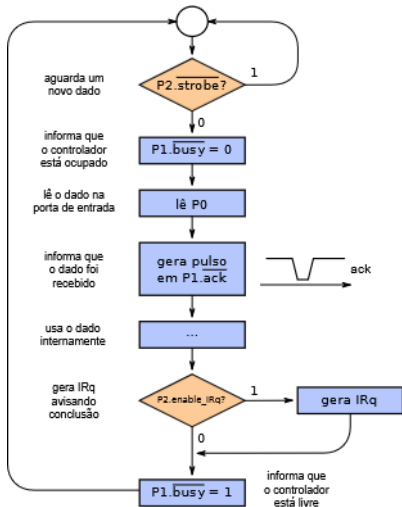
■  $\overline{\text{busy}}$

■  $\overline{\text{ack}}$

## ■ Porta $P_2$ : controle

■  $\overline{\text{strobe}}$

■  $\text{enable\_IRQ}$



# Endereçamento

Como acessar os registradores da interface do dispositivo?

- Entrada/saída mapeada em portas
- Entrada/saída mapeada em memória
- Canais de entrada/saída



# Entrada/saída mapeada em portas

Registradores acessados por instruções específicas.

Na família Intel: instruções IN e OUT.

```

1  IN  %AL, $0x60    // lê caractere do teclado em AL
2
3  OUT $0x3f8, %AL   // escreve byte em AL na saída paralela
  
```

Espaço de endereços de E/S:

- Separado da memória principal.
- Geralmente de 8 ou 16 bits.
- CPU usa um sinal  $IO/\overline{M}$  no barramento de controle.

## Tabela de endereços de portas (típico)

Dispositivo	Endereços
teclado e mouse PS/2	0060h e 0064h
barramento IDE primário	0170h a 0177h
barramento IDE secundário	01F0h a 01F7h
relógio de tempo real	0070h e 0071h
interface serial COM1	02F8h a 02FFh
interface paralela LPT1	0378h a 037Fh

No Linux, consultar arquivo `/proc/ioports`

# Entrada/saída mapeada em memória

Registradores dos dispositivos vistos como RAM:

- São mapeados em faixas de endereços de RAM.
- Podem ser usadas as instruções de acesso à memória.
- Usado para dispositivos de rede, áudio e vídeo.

No Linux, consultar arquivo `/proc/iomem`

# Canais de entrada/saída

Uso de um hardware independente com processador dedicado.

Deixa o processador principal livre para outras tarefas.

Adotada em sistemas de grande porte (*mainframes*).

Usada em periféricos de alto desempenho (GPU vídeo).

# Interrupções

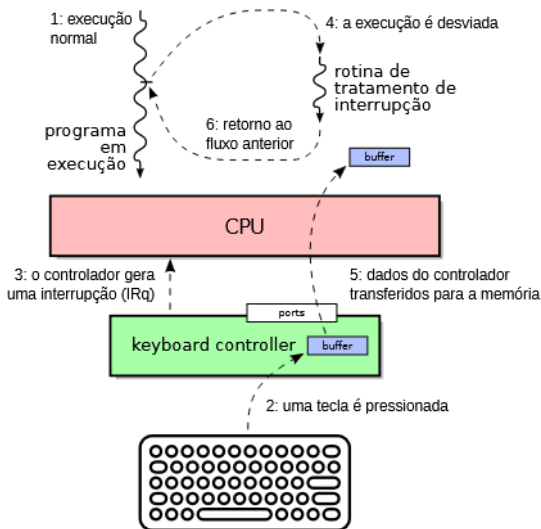
As portas servem para as interações iniciadas pela CPU.

Como fazer para interações iniciadas **pelo controlador**?

**Requisição de interrupção** (IRQ - *Interrupt Request*):

- Sinal elétrico enviado através do barramento de controle.
- Notifica o processador sobre algum evento importante.
- Desvia a execução para uma rotina de tratamento.

# Roteiro de uma interrupção



# Roteiro de uma interrupção

- 1 A CPU está executando um programa.
- 2 O usuário aciona uma tecla do teclado.
- 3 O controlador gera uma interrupção.
- 4 A CPU recebe a interrupção e desvia sua execução para uma rotina de tratamento da interrupção.
- 5 A rotina interage com o controlador do teclado para buscar os dados do buffer.
- 6 A rotina conclui e o programa anterior retoma a execução.

# Interrupções típicas

Dispositivo	Interrupção
teclado	1
mouse PS/2	12
barramento IDE primário	14
barramento IDE secundário	15
relógio de tempo real	8
interface serial COM1	4
interface paralela LPT1	7

No Linux, consultar arquivo `/proc/interrupts`



# Exceções típicas

Exceção	Descrição
0	divide error
3	breakpoint
5	bound range exception
6	invalid opcode
11	segment not present
12	stack fault
13	general protection
14	page fault
16	floating point error

# PIC - Programmable Interrupt Controller

Hardware dedicado no *chipset* da placa-mãe ou na CPU.

Organiza o sistema de interrupções de hardware:

- Recebe as interrupções dos dispositivos.
- Associa cada dispositivo a um número.
- Informa a CPU sobre cada interrupção ocorrida.
- Enfileira as interrupções não-tratadas (pendentes).
- Pode ser programado pela CPU.
- Pode ignorar/mascarar ou priorizar interrupções.

# PIC - Programmable Interrupt Controller

