

Sistemas Operacionais

Gestão de entrada/saída - hardware

Prof. Carlos Maziero

DInf UFPR, Curitiba PR

Abril de 2019

Conteúdo

- 1 Dispositivos
- 2 Componentes
- 3 Barramentos
- 4 Interface
- 5 Endereçamento
- 6 Interrupções

Dispositivos de entrada/saída

Interação com o usuário

- *mouse*, teclado, tela, *joystick*, alto-falantes

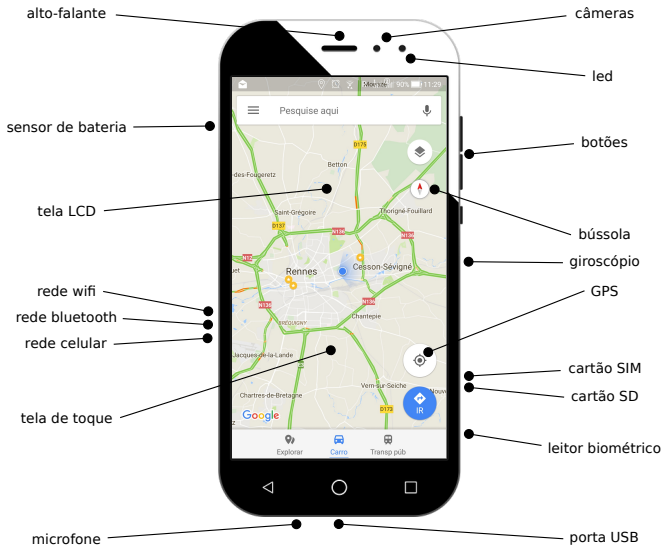
Escrita/leitura de dados

- discos rígidos, DVD-ROMs, *pen-drives*

Comunicação com outros computadores

- redes LAN, WLAN, *Bluetooth*, celular

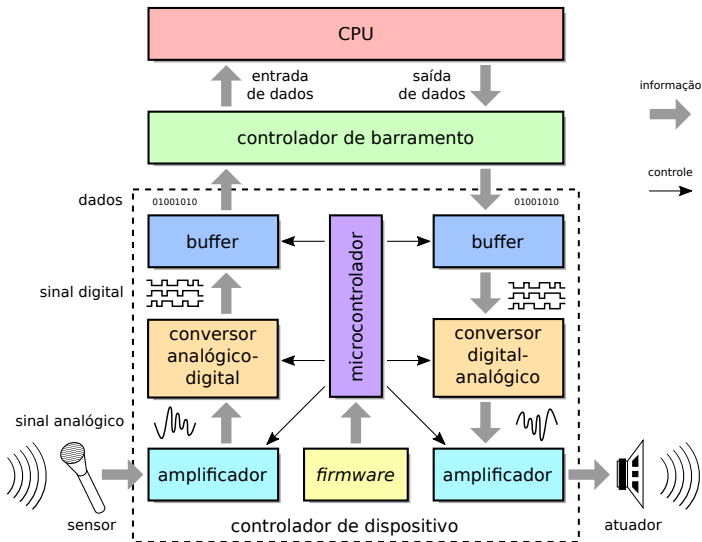
Dispositivos de entrada/saída



Componentes de um dispositivo de E/S

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

E/S



Barramentos

Barramento: via de comunicação:

- interliga CPU, memória e dispositivos
- parte do *chipset* da placa-mãe

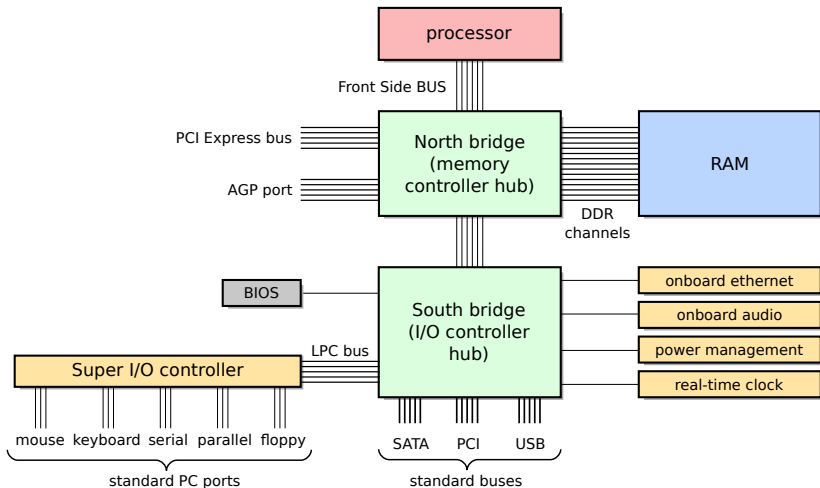
North-bridge: componentes rápidos

- CPU e RAM
- Portas AGP e PCI-express

South-bridge: componentes lentos

- PCI, USB, SATA
- BIOS, legacy controllers

Barramentos em um PC típico



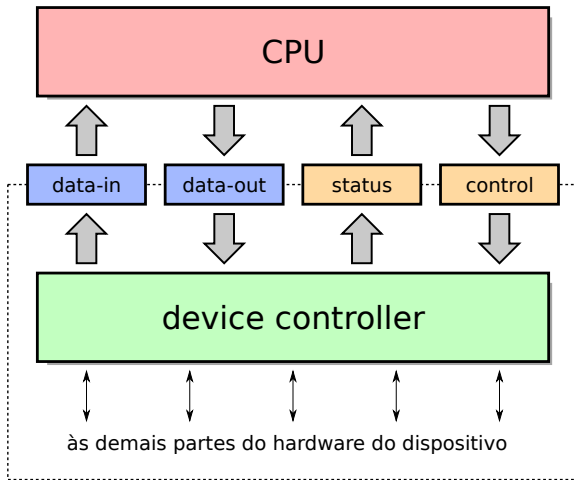
Velocidades de transferência de dados

| 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 |
| Interface de rede <i>Fast Ethernet</i> | 11.6 MB/s |
| Chave 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

| tipo de porta | direção | função |
|----------------------|----------------|---|
| entrada | dev → CPU | receber dados do dispositivo |
| saída | CPU → dev | enviar dados ao dispositivo |
| status | dev → CPU | consultar o estado do dispositivo; verificar status de uma operação |
| controle | CPU → dev | enviar comandos ao dispositivo; modificar configuração do dispositivo |

Interface de acesso



Exemplo: porta paralela (SPP)

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

P_1 (*status port*), 8 bits

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

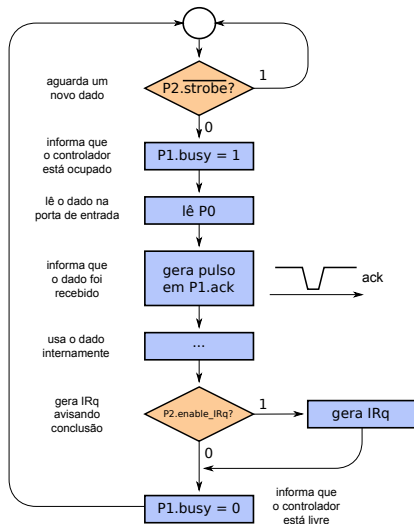
Exemplo: porta paralela

P_2 (*control port*):

- 0 strobe: há um dado em P_0
- 1 `auto_lf`: *line feed* a cada *carriage return*
- 2 `reset`: a impressora deve ser reiniciada
- 3 `select`: a impressora está selecionada para uso
- 4 `enable_IRQ`: permite gerar interrupções
- 5 `bidirectional`: ativa modo bidirecional
- 6 reservado
- 7 reservado

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

Funcionamento da porta paralela



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: “IN *reg port*” e “OUT *port reg*”

`in $0x60, %al // lê caractere do teclado em AL`

I/O address space separado da memória principal

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 |
| porta serial COM1 | 02F8h a 02FFh |
| porta paralela LPT1 | 0378h a 037Fh |

No Linux, consultar arquivo `/proc/ioports`

Entrada/saída mapeada em memória

Registradores dos dispositivos mapeados em endereços de memória

Podem ser usadas as mesmas instruções de acesso à memória

Usado em PCs 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

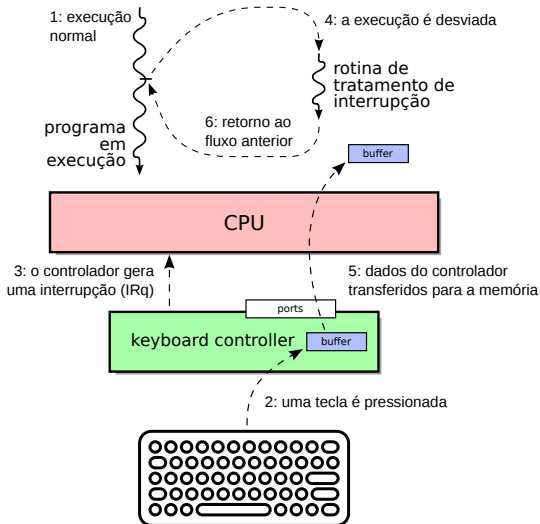
Os registradores servem para as interações iniciadas pela CPU

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

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

- notifica o processador sobre algum evento importante
- sinal elétrico veiculado através do barramento de controle
- Cada interrupção está associada a um número inteiro
- Execução é desviada 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 |
| porta serial COM1 | 4 |
| porta paralela LPT1 | 7 |

No Linux, consultar arquivo `/proc/interrupts`

Exceções

| 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

