Capítulo 19

Hardware de entrada/saída

19.1 Introdução

Um computador é constituído basicamente de processadores, memória RAM e **dispositivos de entrada e saída**, também chamados de **periféricos**. Os dispositivos de entrada/saída permitem a interação do computador com o mundo exterior de várias formas, como por exemplo:

- interação com os usuários através de *mouse*, teclado, tela gráfica, tela de toque e *joystick*;
- escrita e leitura de dados em discos rígidos, SSDs, CD-ROMs, DVD-ROMs e *pen-drives*;
- impressão de informações através de impressoras e plotadoras;
- captura e reprodução de áudio e vídeo, como câmeras, microfones e alto-falantes;
- comunicação com outros computadores, através de redes LAN, *wifi*, *Bluetooth* e de telefonia celular.

Esses exemplos são típicos de computadores pessoais e de computadores menores, como os *smartphones* (Figura 19.1). Já em ambientes industriais, é comum encontrar dispositivos de entrada/saída específicos para a monitoração e controle de máquinas e processos de produção, como tornos de comando numérico, braços robotizados e controladores de processos químicos. Por sua vez, o computador embarcado em um carro conta com dispositivos de entrada para coletar dados do combustível e do funcionamento do motor e dispositivos de saída para controlar a injeção eletrônica e a tração dos pneus, por exemplo.

É bastante óbvio que um computador não tem muita utilidade sem dispositivos periféricos, pois o objetivo básico da imensa maioria dos computadores é receber dados, processá-los e devolver resultados aos seus usuários, sejam eles seres humanos, outros computadores ou processos físicos/químicos externos.

Os primeiros sistemas de computação, construídos nos anos 1940, eram destinados a cálculos matemáticos e por isso possuíam dispositivos de entrada/saída rudimentares, que apenas permitiam carregar/descarregar programas e dados diretamente na memória principal. Em seguida surgiram os terminais compostos de teclado e monitor de texto, para facilitar a leitura e escrita de dados, e os discos rígidos, como

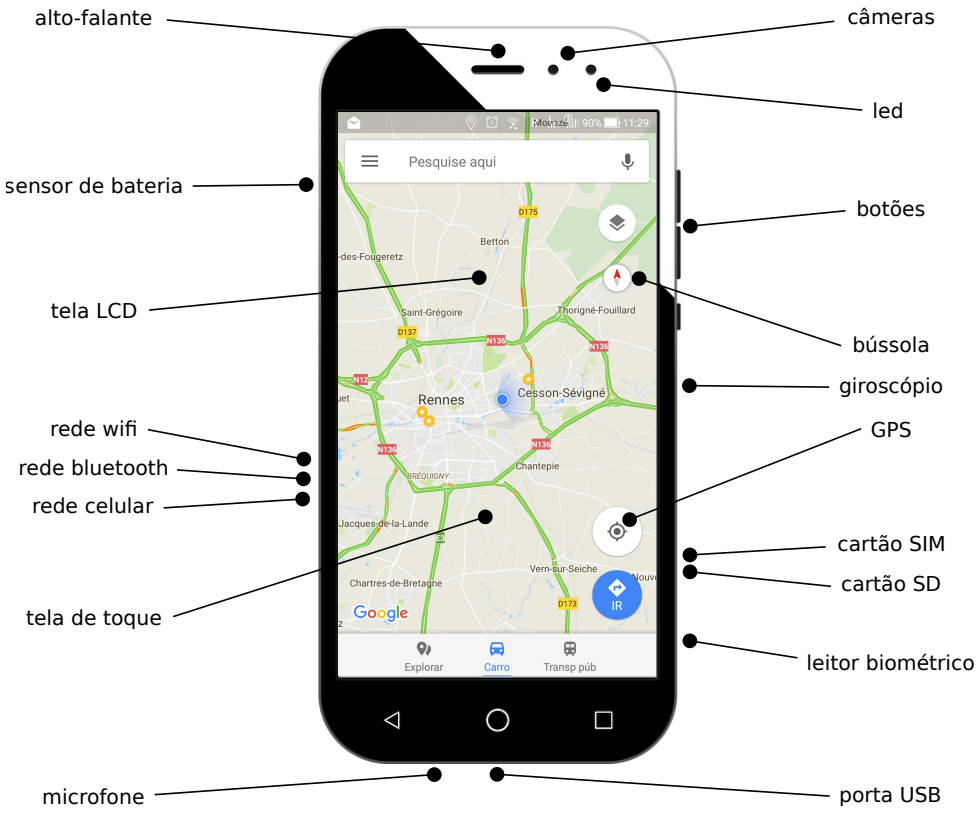

Figura 19.1: Um *smartphone* com seus dispositivos de entrada e saída.

meio de armazenamento persistente de dados e programas. Hoje, dispositivos de entrada/saída 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.

Este capítulo apresenta uma visão geral das estruturas de hardware associadas aos dispositivos de entrada/saída presentes em computadores convencionais para a interação com o usuário, armazenamento de dados e comunicação.

19.2 Componentes de um dispositivo

Conceitualmente, a entrada de dados em um computador inicia com um **sensor** capaz de converter uma informação externa (física ou química) em um sinal elétrico analógico. Como exemplos de sensores temos o microfone, as chaves internas das teclas de um teclado ou o foto-diodo de um leitor de DVDs. O sinal elétrico analógico fornecido pelo sensor é então aplicado a um **conversor analógico-digital** (CAD), que o transforma em informação digital (sequências de bits). Essa informação digital é armazenada em um *buffer* que pode ser acessado pelo processador através de um **controlador de entrada**.

Uma saída de dados inicia com o envio de dados do processador a um **controlador de saída**, através do barramento. Os dados enviados pelo processador são armazenados em um *buffer* interno do controlador e a seguir convertidos em um sinal elétrico analógico, através de um **conversor digital-analógico** (CDA). Esse sinal será

aplicado a um **atuador**[1] que irá convertê-lo em efeitos físicos perceptíveis ao usuário. Exemplos simples de atuadores são a cabeça de impressão e os motores de uma impressora, um alto-falante, uma tela gráfica, etc.

Vários dispositivos combinam funcionalidades tanto de entrada quanto de saída, como os discos rígidos: o processador pode ler dados gravados no disco rígido (entrada), mas para isso precisa ativar o motor que faz girar o disco e posicionar adequadamente a cabeça de leitura (saída). O mesmo ocorre com uma placa de áudio de um PC convencional, que pode tanto capturar quanto reproduzir sons. A Figura 19.2 mostra a estrutura básica de captura e reprodução de áudio em um computador pessoal, que é um bom exemplo de dispositivo de entrada/saída.

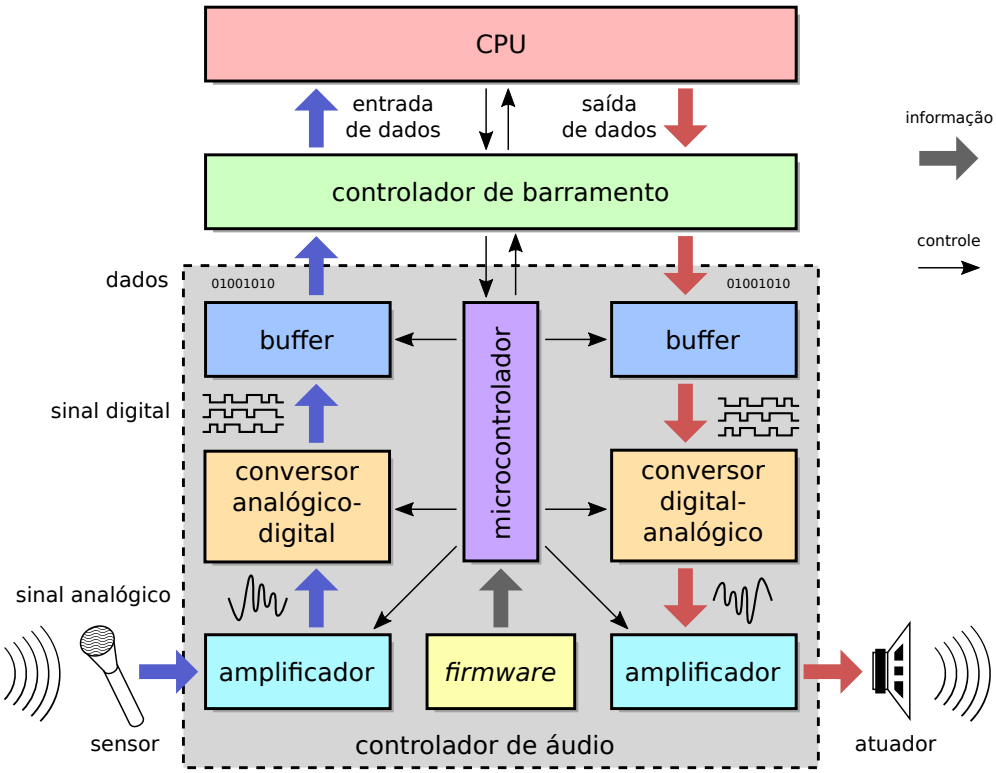

Figura 19.2: Estrutura básica da entrada e saída de áudio.

Os dispositivos de entrada/saída mais complexos, como discos rígidos, placas de rede e placas gráficas, possuem um processador ou microcontrolador interno para gerenciar sua operação. Esse processador embutido no dispositivo executa um código criado pelo fabricante do mesmo, denominado *firmware*. O código do *firmware* é independente do sistema operacional do computador e contém as instruções necessárias para operar o restante do hardware do dispositivo, permitindo realizar as operações solicitadas pelo sistema operacional.

[1]Sensores e atuadores são denominados genericamente *dispositivos transdutores*, pois transformam energia externa (como luz, calor, som ou movimento) em sinais elétricos, ou vice-versa.

19.3 Barramentos de acesso

Historicamente, o acoplamento dos dispositivos de entrada/saída ao computador é feito através de barramentos, seguindo o padrão estabelecido pela arquitetura de *Von Neumann*. Enquanto o barramento dos primeiros sistemas era um simples agrupamento de fios, os barramentos dos sistemas atuais são estruturas de hardware bastante complexas, com circuitos específicos para seu controle. Além disso, a diversidade de velocidades e volumes de dados suportados pelos dispositivos fez com que o barramento único dos primeiros sistemas fosse gradativamente estruturado em um conjunto de barramentos com características distintas de velocidade e largura de dados.

O controle dos barramentos em um sistema *desktop* moderno está a cargo de dois controladores de hardware que fazem parte do *chipset*[2] da placa-mãe: a *north bridge* e a *south bridge*. A *north bridge*, diretamente conectada ao processador, é responsável pelo acesso à memória RAM e aos dispositivos de alta velocidade, através de barramentos dedicados como AGP (*Accelerated Graphics Port*) e PCI-Express (*Peripheral Component Interconnect*).

Por outro lado, a *south bridge* é o controlador responsável pelos barramentos e portas de baixa ou média velocidade do computador, como as interfaces seriais e paralelas, e pelos barramentos dedicados como o PCI padrão, o USB e o SATA. Além disso, a *south bridge* costuma integrar outros componentes importantes do computador, como controladores de áudio e rede *on-board*, controlador de interrupções, controlador DMA (*Direct Memory Access*), temporizadores (responsáveis pelas interrupções de tempo usadas pelo escalonador de processos), relógio de tempo real (que mantém a informação de dia e hora atuais), controle de energia e acesso à memória BIOS. O processador se comunica com a *south bridge* indiretamente, através da *north bridge*.

A Figura 19.3 traz uma visão da arquitetura típica de um computador pessoal moderno. A estrutura detalhada e o funcionamento dos barramentos e seus respectivos controladores estão fora do escopo deste texto; informações mais detalhadas podem ser encontradas em [Patterson and Henessy, 2005].

Como existem muitas possibilidades de interação do computador com o mundo exterior, também existem muitos tipos de dispositivos de entrada/saída, com características diversas de velocidade de transferência, forma de transferência dos dados e método de acesso. A **velocidade de transferência de dados** de um dispositivo pode ir de alguns bytes por segundo, no caso de dispositivos simples como teclados e *mouses*, a gigabytes por segundo, para algumas placas de interface gráfica ou de acesso a discos de alto desempenho. A Tabela 19.1 traz alguns exemplos de dispositivos de entrada/saída com suas velocidades típicas de transferência de dados.

[2] O *chipset* de um computador é um conjunto de controladores e circuitos auxiliares de hardware integrados à placa-mãe, que proveem serviços fundamentais ao funcionamento do computador, como o controle dos barramentos, acesso à BIOS, controle de interrupções, temporizadores programáveis e controladores *on-board* para alguns periféricos, como discos rígidos, interfaces paralelas e seriais e entrada/saída de áudio.

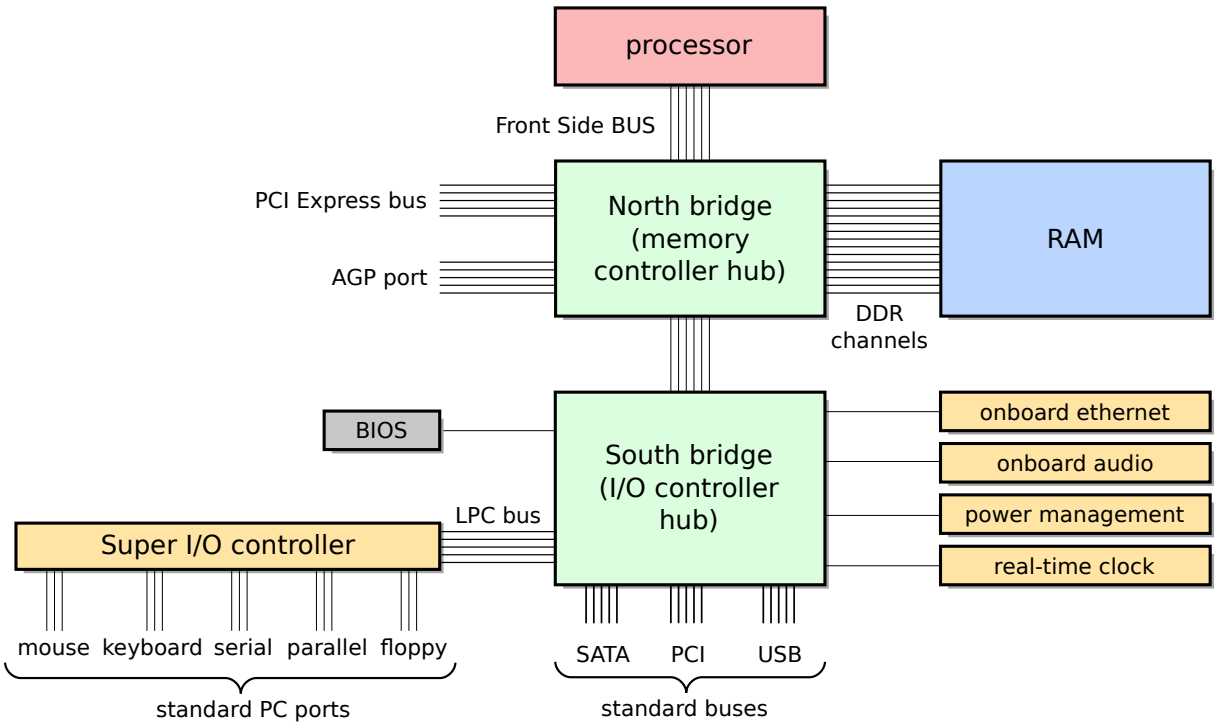

Figura 19.3: Arquitetura típica de um PC atual.

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 *Fast Ethernet*	11.6 MB/s
Pendrive ou disco USB 2.0	60 MB/s
Interface de rede *Gigabit Ethernet*	116 MB/s
Disco rígido SATA 2	300 MB/s
Interface gráfica *high-end*	4.2 GB/s

Tabela 19.1: Velocidades típicas de alguns dispositivos de entrada/saída.

19.4 Interface de acesso

Para o sistema operacional, o aspecto mais relevante de um dispositivo de entrada/saída é sua interface de acesso, ou seja, a abordagem a ser usada para acessar o dispositivo, configurá-lo e enviar dados para ele (ou receber dados dele). Normalmente, cada dispositivo oferece um conjunto de registradores acessíveis através do barramento, também denominados **portas de entrada/saída**, que são usados para a comunicação entre o dispositivo e o processador. As portas oferecidas para acesso a cada dispositivo de entrada/saída podem ser divididas nos seguintes grupos (conforme ilustrado na Figura 19.4):

Portas de entrada (*data-in ports*): usadas pelo processador para receber dados provindos do dispositivo; são escritas pelo dispositivo e lidas pelo processador;

Portas de saída (*data-out ports*): usadas pelo processador para enviar dados ao dispositivo; essas portas são escritas pelo processador e lidas pelo dispositivo;

Portas de status (*status ports*): usadas pelo processador para consultar o estado interno do dispositivo ou verificar se uma operação solicitada ocorreu sem erro; essas portas são escritas pelo dispositivo e lidas pelo processador;

Portas de controle (*control ports*): usadas pelo processador para enviar comandos ao dispositivo ou modificar parâmetros de sua configuração; essas portas são escritas pelo processador e lidas pelo dispositivo.

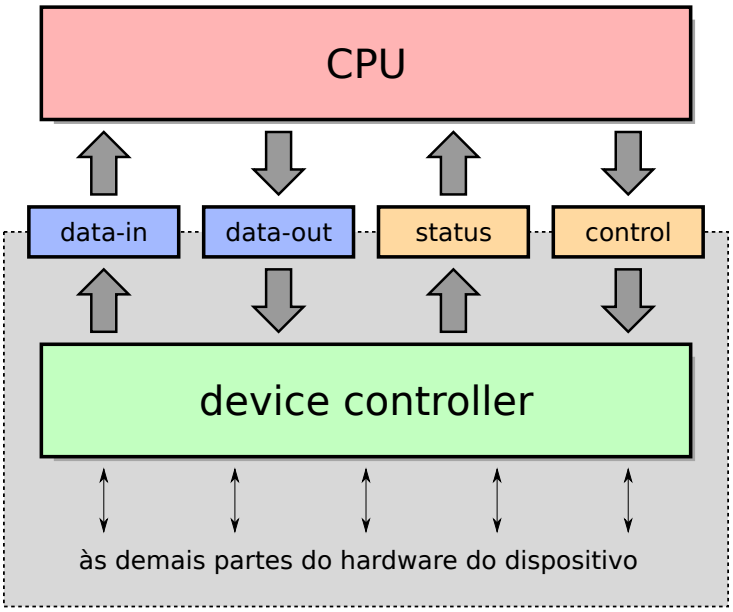

Figura 19.4: Portas de interface de um dispositivo de entrada/saída.

O número exato de portas e o significado específico de cada uma dependem do tipo de dispositivo considerado. Um exemplo simples de interface de acesso a dispositivo é a interface paralela, geralmente usada para acessar impressoras mais antigas. As portas de uma interface paralela operando no modo padrão (SPP - *Standard Parallel Port*) têm um byte cada e estão descritas a seguir [Patterson and Henessy, 2005]:

- P_0 (*data port*): porta de saída, usada para enviar dados à impressora; pode ser usada também como porta de entrada, se a interface estiver operando em modo bidirecional;
- P_1 (*status port*): porta de status, permite ao processador consultar vários indicadores de status da interface paralela ou do dispositivo ligado a ela. O significado de cada um de seus 8 bits é:
  0. reservado;
  1. reservado;
  2. $\overline{\texttt{nIRq}}$: se 0, indica que o controlador gerou uma interrupção (Seção 19.6);
  3. `error`: há um erro interno na impressora;
  4. `select`: a impressora está pronta (*online*);
  5. `paper_out`: falta papel na impressora;
  6. $\overline{\texttt{ack}}$: um pulso em 0 indica que o dado na porta P_0 foi recebido pelo controlador (pulso com duração $t \geq 1\mu s$);
  7. `busy`: indica que o controlador está ocupado processando um comando.
- P_2 (*control port*): porta de controle, usada para configurar a interface paralela e para solicitar operações de saída (ou entrada) de dados através da mesma. Seus 8 bits têm o seguinte significado:
  0. $\overline{\texttt{strobe}}$: um pulso em 0 informa o controlador que há um dado disponível em P_0 (pulso com duração $t \geq 0,5\mu s$);
  1. `auto_lf`: a impressora deve inserir um *line feed* a cada *carriage return* recebido;
  2. `reset`: a impressora deve ser reiniciada;
  3. `select`: a impressora está selecionada para uso;
  4. `enable_IRq`: permite ao controlador gerar interrupções (Seção 19.6);
  5. `bidirectional`: informa que a interface será usada para entrada e para saída de dados;
  6. reservado;
  7. reservado.
- P_3 a P_7: estas portas são usadas nos modos estendidos de operação da interface paralela, como EPP (*Enhanced Paralel Port*) e ECP (*Extended Capabilities Port*).

O algoritmo básico implementado pelo hardware interno do controlador da interface paralela para coordenar suas interações com o processador está ilustrado no fluxograma da Figura 19.5. Considera-se que os valores iniciais dos flags de status da porta P_1 são `nIRq=1`, `error=0`, `select=1`, `paper_out=0`, `ack=1` e `busy=0`.

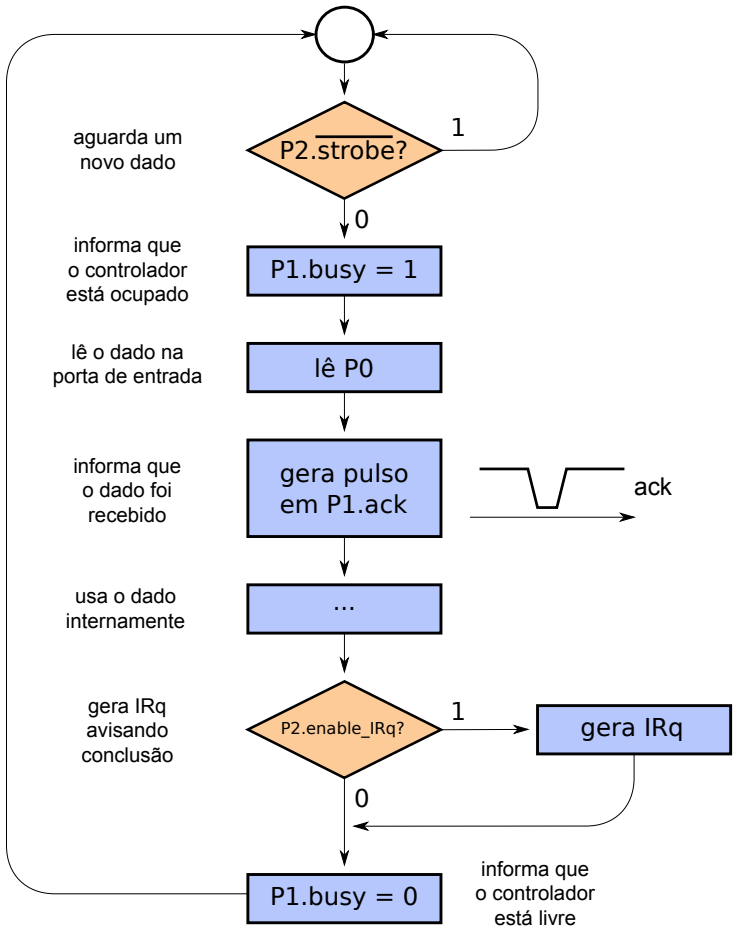

Figura 19.5: Comportamento básico do controlador da interface paralela.

19.5 Endereçamento de portas

A forma de acesso às portas que compõem a interface de um dispositivo varia de acordo com a arquitetura do computador. Alguns sistemas utilizam **entrada/saída mapeada em portas** (*port-mapped I/O*), onde as portas que compõem a interface são acessadas pelo processador através de instruções específicas para operações de entrada/saída. Por exemplo, os processadores da família Intel usam a instrução "`IN` *reg port*" para ler o valor presente na porta "*port*" do dispositivo e depositá-lo no registrador "*reg*" do processador, enquanto a instrução "`OUT` *port reg*" é usada para escrever na porta "*port*" o valor contido no registrador "*reg*".

A entrada/saída mapeada em portas usa um espaço de endereços de entrada/saída (*I/O address space*) independente da memória principal, normalmente compreendido entre 0 e 64KB. Ou seja, o endereço de porta de E/S 001*Fh* e o endereço de memória 001*Fh* apontam para informações distintas. Para distinguir entre endereços de memória e endereços de portas de entrada/saída, o barramento de controle do processador possui uma linha $IO/\overline{M}$, que indica se o endereço presente no barramento de endereços se refere a uma posição de memória (se $IO/\overline{M} = 0$) ou a uma porta de entrada/saída (se $IO/\overline{M} = 1$). A Tabela 19.2 apresenta os endereços típicos de algumas portas de entrada/saída de dispositivos em computadores pessoais que seguem o padrão IBM-PC.

Uma outra forma de acesso aos dispositivos de entrada/saída é a **entrada/saída mapeada em memória** (*memory-mapped I/O*). Nesta abordagem, uma parte não ocupada

Dispositivo	Endereços das portas
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 serial COM2	03F8h a 03FFh
interface paralela LPT1	0378h a 037Fh

Tabela 19.2: Endereços de portas de E/S de alguns dispositivos.

do espaço de endereços de memória é reservado para mapear as portas de acesso aos dispositivos. Dessa forma, as portas são vistas como se fossem parte da memória principal e podem ser lidas e escritas através das mesmas instruções usadas para acessar o restante da memória, sem a necessidade de instruções especiais como `IN` e `OUT`. Algumas arquiteturas de computadores, como é caso do IBM-PC padrão, usam uma abordagem híbrida para certos dispositivos como interfaces de rede e de áudio: as portas de controle e status são mapeadas no espaço de endereços de entrada/saída, sendo acessadas através de instruções específicas, enquanto as portas de entrada e saída de dados são mapeadas em memória (normalmente na faixa de endereços entre 640 KB e 1MB) [Patterson and Henessy, 2005].

Finalmente, uma abordagem mais sofisticada para o controle de dispositivos de entrada/saída é o uso de um hardware independente, com processador dedicado, que comunica com o processador principal através de um barramento específico. Em sistemas de grande porte (*mainframes*) essa abordagem é denominada **canais de entrada/saída** (*IO channels*); em computadores pessoais, essa abordagem costuma ser usada em interfaces para vídeo ou áudio de alto desempenho, como é o caso das placas gráficas com aceleração, nas quais um processador gráfico (GPU – *Graphics Processing Unit*) realiza a parte mais pesada do processamento da saída de vídeo, como a renderização de imagens em 3 dimensões e texturas, deixando o processador principal livre para outras tarefas.

19.6 Interrupções

O acesso aos controladores de dispositivos através de suas portas é conveniente para a comunicação no sentido *processador* → *controlador*, ou seja, para as interações iniciadas pelo processador. Entretanto, essa forma de acesso é inviável para interações iniciadas pelo controlador, pois o processador pode demorar a acessar suas portas, caso esteja ocupado em outras atividades.

Frequentemente um controlador de dispositivo precisa informar ao processador com rapidez sobre um evento interno, como a chegada de um pacote de rede, um clique de mouse ou a conclusão de uma operação de disco. Nesse caso, o controlador pode notificar o processador sobre o evento ocorrido através de uma **requisição de interrupção** (IRq - *Interrupt Request*).

As requisições de interrupção são sinais elétricos veiculados através do barramento de controle do computador. Cada interrupção está geralmente associada a um número inteiro que permite identificar sua origem. A Tabela 19.3 informa os números de interrupção associados a alguns dispositivos periféricos típicos em um PC no padrão *Intel x86*.

Dispositivo	**Interrupção**
teclado	1
interface serial COM2	3
interface serial COM1	4
interface paralela LPT1	7
relógio de tempo real	8
mouse PS/2	12
barramento ATA primário	14
barramento ATA secundário	15

Tabela 19.3: Interrupções geradas por alguns dispositivos em um PC.

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** (*interrupt handler*). Essa rotina é responsável por tratar aquela requisição de interrupção, ou seja, executar as ações necessárias para acessar o controlador de dispositivo e tratar o evento que a gerou. Ao final da rotina de tratamento da interrupção, o processador retoma o código que estava executando quando foi interrompido. A Figura 19.6 representa os principais passos associados ao tratamento de uma interrupção envolvendo o controlador de teclado, detalhados a seguir:

1. O processador está executando um programa qualquer;
2. O usuário pressiona uma tecla no teclado;
3. O controlador do teclado identifica a tecla pressionada, armazena seu código em um *buffer* interno e envia uma requisição de interrupção (IRq) ao processador;
4. O processador recebe a interrupção, salva na pilha seu estado atual (o conteúdo de seus registradores) e desvia sua execução para a rotina de tratamento da interrupção associada ao teclado;
5. Ao executar, essa rotina acessa as portas do controlador de teclado para transferir o conteúdo de seu *buffer* para uma área de memória do núcleo. Depois disso, ela pode executar outras ações, como acordar algum processo ou *thread* que esteja esperando por entradas do teclado;
6. Ao concluir a execução da rotina de tratamento da interrupção, o processador retorna à execução do fluxo de instruções que havia sido interrompido, usando a informação de estado salva no passo 4.

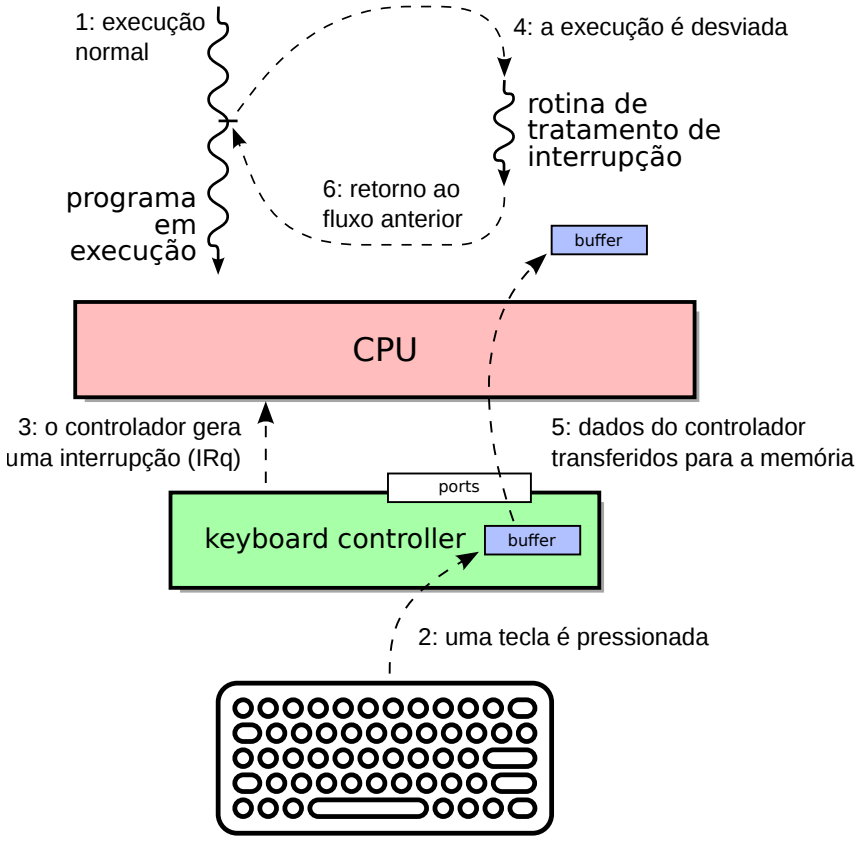

Figura 19.6: Roteiro típico de um tratamento de interrupção

Essa sequência de ações ocorre a cada requisição de interrupção recebida pelo processador. Como cada interrupção corresponde a um evento ocorrido em um dispositivo periférico (chegada de um pacote de rede, movimento do mouse, conclusão de operação do disco, etc.), podem ocorrer centenas ou mesmo milhares de interrupções por segundo, dependendo da carga de trabalho e da configuração do sistema (número e tipo de periféricos). Por isso, as rotinas de tratamento de interrupção devem realizar suas tarefas rapidamente, para não prejudicar o funcionamento do restante do sistema.

Como cada tipo de interrupção pode exigir um tipo de tratamento diferente (pois os dispositivos são diferentes), cada requisição de interrupção deve disparar uma rotina de tratamento específica. A maioria das arquiteturas de processador atuais define uma tabela de endereços de funções denominada *Tabela de Interrupções* (IVT - *Interrupt Vector Table*); cada entrada dessa tabela contém o endereço da rotina de tratamento da interrupção correspondente. Por exemplo, se a entrada 5 da tabela contém o valor `3C20h`, então a rotina de tratamento da IRq 5 iniciará na posição `3C20h` da memória RAM. Dependendo do hardware, a tabela de interrupções pode residir em uma posição fixa da memória RAM, definida pelo fabricante do processador, ou ter sua posição indicada pelo conteúdo de um registrador da CPU específico para esse fim.

Além das requisições de interrupção geradas pelos controladores de dispositivos, eventos internos ao processador também podem ocasionar o desvio da execução usando o mesmo mecanismo de interrupção: são as **exceções**. Eventos como instruções inválidas, divisão por zero ou outros erros de software disparam exceções internas no processador, que resultam na ativação de rotinas de tratamento de exceção registradas na tabela de

interrupções. A Tabela 19.4 apresenta algumas exceções previstas pelo processador Intel Pentium (extraída de [Patterson and Henessy, 2005]).

Exceção	Descrição
0	erro de divisão por zero
3	breakpoint (parada de depurador)
5	erro de faixa de valores
6	operação inválida
7	dispositivo não disponível
11	segmento de memória ausente
12	erro de pilha
14	falta de página
16	erro de ponto flutuante
19-31	valores reservados pela Intel

Tabela 19.4: Algumas exceções dos processadores *Intel x86*.

Nas arquiteturas de hardware atuais, as interrupções geradas pelos dispositivos de entrada/saída não são transmitidas diretamente ao processador, mas a um **controlador de interrupções programável** (PIC - *Programmable Interrupt Controller*, ou APIC - *Advanced Programmable Interrupt Controller*), que faz parte do *chipset* do computador. As linhas de interrupção dos controladores de periféricos são conectadas aos pinos desse controlador de interrupções, enquanto suas saídas são conectadas às entradas de interrupção do processador.

O controlador de interrupções recebe as interrupções dos dispositivos e as encaminha ao processador em sequência, uma a uma. Ao receber uma interrupção, o processador deve acessar a interface do PIC para identificar a origem da interrupção e depois "reconhecê-la", ou seja, informar ao PIC que aquela interrupção foi tratada e pode ser descartada pelo controlador. Interrupções já ocorridas mas ainda não reconhecidas pelo processador são chamadas de **interrupções pendentes**.

O PIC pode ser programado para bloquear ou ignorar algumas das interrupções recebidas, impedindo que cheguem ao processador. Além disso, ele permite definir prioridades entre as interrupções. A Figura 19.7 mostra a operação básica de um controlador de interrupções; essa representação é simplificada, pois as arquiteturas de computadores mais recentes podem contar com vários controladores de interrupções interligados.

O mecanismo de interrupção torna eficiente a interação do processador com os dispositivos periféricos. Se não existissem interrupções, o processador perderia muito tempo consultando todos os dispositivos do sistema para verificar se há eventos a serem tratados. Além disso, as interrupções permitem construir funções de entrada/saída assíncronas: o processador não precisa esperar a conclusão de cada operação solicitada, pois o dispositivo emitirá uma interrupção para "avisar" o processador quando a operação estiver concluída.

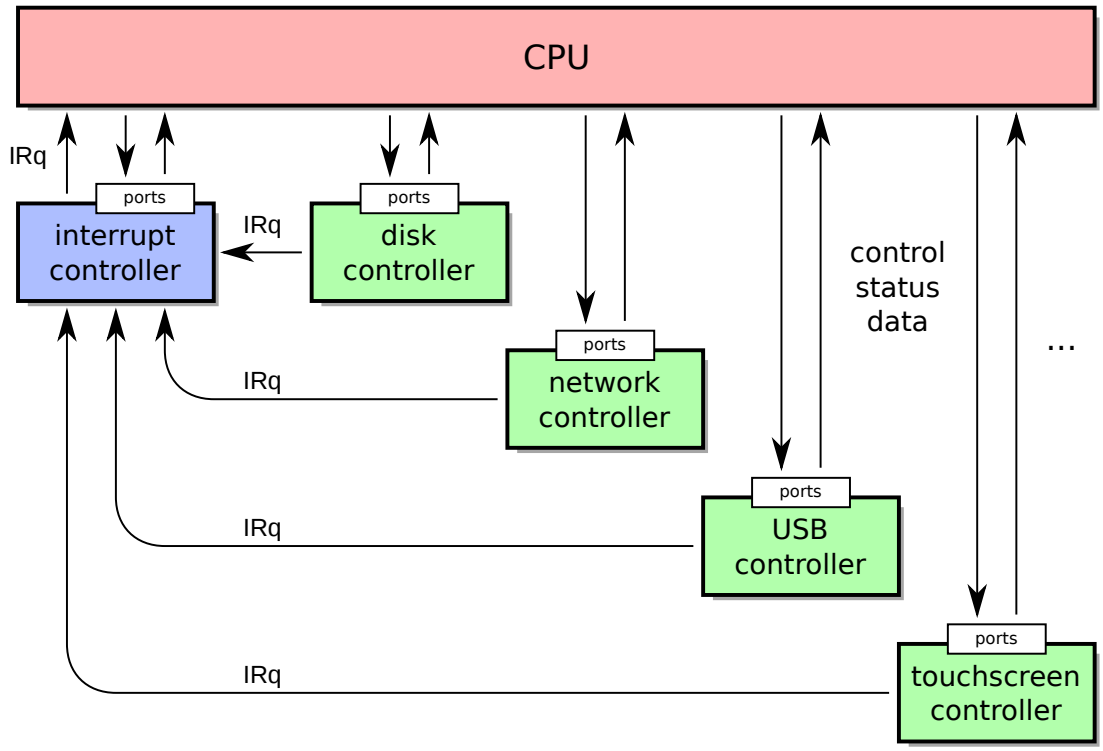

Figura 19.7: Uso de um controlador de interrupções.

Referências

D. Patterson and J. Henessy. *Organização e Projeto de Computadores*. Campus, 2005.