

Nota: Este material complementar, disponível em <https://prettore.github.io/lectures.html> representa uma cópia resumida de conteúdos bibliográficos disponíveis gratuitamente na Internet.

# Gerenciamento de Entrada e Saída

<b>Introdução</b>	<b>1</b>
Princípios de Hardware de Entrada e Saída	1
Princípios de Software de Entrada e Saída	2
Objetivos do Software de E/S	2
<b>As Diferentes Camadas de Entrada e Saída</b>	<b>3</b>
<b>Discos e SSDs (Solid State Drives)</b>	<b>4</b>
Discos Magnéticos (Hard Disk Drives - HDDs)	4
Solid State Drives (SSDs)	4
<b>Relógio (Clock)</b>	<b>4</b>
<b>Interfaces com Usuário</b>	<b>5</b>
<b>Gerenciamento de Energia</b>	<b>5</b>
<b>Conclusão</b>	<b>5</b>
<b>Referências</b>	<b>6</b>

## Introdução

O subsistema de Entrada e Saída (E/S) de um sistema operacional é responsável por gerenciar a comunicação entre a CPU/memória e a vasta gama de dispositivos periféricos conectados ao computador. Esses dispositivos variam enormemente em velocidade, funcionalidade e complexidade, incluindo teclados, mouses, monitores, discos rígidos, SSDs, interfaces de rede, impressoras e sensores especializados. O principal objetivo do gerenciamento de E/S é fornecer uma interface simples e uniforme para os processos acessarem esses dispositivos, abstraindo os detalhes complexos de hardware, ao mesmo tempo em que otimiza o desempenho e a eficiência do sistema. Este capítulo explora os princípios de hardware e software de E/S, as diferentes camadas envolvidas no processamento de uma solicitação de E/S, o gerenciamento de dispositivos específicos como discos e SSDs, o papel do relógio (clock) e das interfaces com o usuário, e considerações sobre o gerenciamento de energia relacionado à E/S.

## Princípios de Hardware de Entrada e Saída

O hardware de E/S consiste em dispositivos periféricos e seus respectivos controladores.

- **Dispositivos de E/S (I/O Devices):** Podem ser classificados em dispositivos de bloco (e.g., discos, SSDs, que armazenam informações em blocos de tamanho fixo com endereços próprios), dispositivos de caractere (e.g., teclados, mouses, portas

seriais, que entregam ou aceitam um fluxo de caracteres sem estrutura de blocos) e dispositivos de rede.

- **Controladores de Dispositivo (Device Controllers):** Cada dispositivo periférico é conectado ao barramento do computador através de um controlador. O controlador é um conjunto de eletrônicos que opera uma porta, um barramento ou um dispositivo. Ele possui registradores para dados e controle, e pode ter sua própria memória (buffer). O SO interage com o controlador, não diretamente com o dispositivo.
- **Portas de E/S (I/O Ports):** Os controladores de dispositivo possuem registradores que são acessados pela CPU através de portas de E/S. Existem duas maneiras principais de endereçar essas portas:
  - **E/S Mapeada em Portas (Port-Mapped I/O):** Usa um espaço de endereçamento de portas separado do espaço de endereçamento de memória. Instruções especiais de CPU (e.g., IN, OUT em arquiteturas x86) são usadas para ler/escrever nas portas.
  - **E/S Mapeada em Memória (Memory-Mapped I/O - MMIO):** Os registradores do controlador de dispositivo são mapeados no espaço de endereçamento de memória do sistema. A CPU usa as mesmas instruções de acesso à memória para interagir com os dispositivos. MMIO é mais comum em sistemas modernos.
- **Comunicação CPU-Controlador:**
  - **Polling (Sondagem):** A CPU repetidamente verifica o bit de status no controlador para ver se o dispositivo está pronto para a próxima operação. Desperdiça tempo de CPU.
  - **Interrupções (Interrupts):** Quando um dispositivo completa uma operação ou precisa de atenção, o controlador envia um sinal de interrupção para a CPU. A CPU suspende sua execução atual, salva seu estado e executa uma rotina de tratamento de interrupção (Interrupt Service Routine - ISR) específica para o dispositivo. Mais eficiente que polling.
  - **Acesso Direto à Memória (Direct Memory Access - DMA):** Para transferências de grandes volumes de dados (e.g., de/para discos), o DMA permite que o controlador de dispositivo transfira dados diretamente de/para a memória principal sem envolvimento constante da CPU. A CPU configura o DMA com o endereço de origem, endereço de destino e contagem de bytes, e o DMA notifica a CPU (via interrupção) quando a transferência está completa. Isso libera a CPU para outras tarefas.

## Princípios de Software de Entrada e Saída

O software de E/S é organizado em camadas para gerenciar a complexidade e fornecer abstração.

### Objetivos do Software de E/S

- **Independência de Dispositivo (Device Independence):** Permitir que programas acessem qualquer dispositivo de E/S de maneira uniforme, sem precisar conhecer os detalhes específicos do dispositivo. Por exemplo, um programa deve poder

escrever em um arquivo em um disco rígido, SSD ou pen drive usando as mesmas chamadas de sistema.

- **Nomenclatura Uniforme (Uniform Naming):** Nomes de arquivos e dispositivos devem ser strings ou inteiros, não dependendo do dispositivo em si.
- **Tratamento de Erros:** Erros devem ser tratados o mais próximo possível do hardware. Se o controlador detectar um erro, ele deve tentar corrigi-lo (se possível) ou reportá-lo ao software de nível superior.
- **Transferências Síncronas vs. Assíncronas:**
  - **Síncrona (Bloqueante):** O processo que solicita a E/S é bloqueado até que a E/S seja concluída.
  - **Assíncrona (Não Bloqueante):** O processo continua a execução enquanto a E/S está em andamento. O SO notifica o processo quando a E/S é concluída (e.g., via sinal, callback).
- **Buffering, Caching e Spooling:** Técnicas para melhorar o desempenho e desacoplar a velocidade dos dispositivos da velocidade da CPU.
  - **Buffer:** Uma área de memória que armazena dados temporariamente enquanto eles estão sendo transferidos entre dispositivos ou entre um dispositivo e uma aplicação.
  - **Cache:** Uma região de memória rápida que armazena cópias de dados de uma memória mais lenta (e.g., cache de disco na RAM).
  - **Spooling (Simultaneous Peripheral Operations On-Line):** Coloca trabalhos em um buffer (uma fila em disco) para que um dispositivo (como uma impressora) possa acessá-los quando estiver pronto. Permite que múltiplos processos gerem saída para o dispositivo sem esperar que ele fique disponível.
- **Compartilhamento de Dispositivos Dedicados vs. Compartilhados:** Alguns dispositivos podem ser usados exclusivamente por um processo (dedicados), enquanto outros podem ser compartilhados entre múltiplos processos (compartilhados).

## As Diferentes Camadas de Entrada e Saída

O software de E/S é tipicamente estruturado em camadas:

1. **Rotinas de Tratamento de Interrupção (Interrupt Handlers):** Camada mais baixa, acionada por interrupções de hardware. Salva o estado, identifica a causa da interrupção e chama o driver de dispositivo apropriado.
2. **Drivers de Dispositivo (Device Drivers):** Código específico do dispositivo que entende o controlador de dispositivo e fornece uma interface padronizada para o resto do kernel. Implementa as operações do dispositivo (e.g., ler bloco, escrever bloco).
3. **Software de E/S Independente de Dispositivo (Device-Independent I/O Software):** Camada do kernel que fornece uma interface uniforme para os drivers

de dispositivo. Responsável por funções como nomenclatura de dispositivos, proteção, bloqueio, buffering e alocação de blocos.

4. **Software de E/S no Espaço do Usuário (User-Space I/O Software):** Consiste em bibliotecas (e.g., stdio em C) que fornecem chamadas de sistema de E/S convenientes para as aplicações (e.g., printf, scanf) e, às vezes, formatação de E/S e spooling.

## Discos e SSDs (Solid State Drives)

Discos magnéticos e SSDs são os principais dispositivos de armazenamento secundário.

### Discos Magnéticos (Hard Disk Drives - HDDs)

- **Estrutura:** Consistem em pratos (platters) que giram, com superfícies magnéticas onde os dados são armazenados. Cabeças de leitura/escrita montadas em braços se movem sobre os pratos. Cada prato é dividido em trilhas (tracks) concêntricas, e cada trilha é dividida em setores (sectors).
- **Tempo de Acesso:** O tempo para acessar um bloco no disco consiste em: tempo de busca (seek time - mover o braço para a trilha correta), latência rotacional (rotational latency - esperar o setor desejado girar até a cabeça) e tempo de transferência (transfer time - transferir os dados).
- **Escalonamento de Disco:** Algoritmos para otimizar a ordem em que as solicitações de E/S de disco são atendidas, visando minimizar o tempo de busca. Exemplos: FCFS, SSTF (Shortest Seek Time First), SCAN (Elevador), C-SCAN (Circular SCAN), LOOK, C-LOOK.

### Solid State Drives (SSDs)

- **Tecnologia:** Usam memória flash (geralmente NAND) para armazenar dados, sem partes móveis.
- **Vantagens sobre HDDs:** Tempo de acesso muito mais rápido (sem busca ou latência rotacional), maior taxa de transferência, menor consumo de energia, mais resistentes a choques.
- **Desvantagens:** Custo por gigabyte geralmente maior, e as células de memória flash têm um número limitado de ciclos de escrita (embora técnicas de wear leveling ajudem a mitigar isso).
- **Gerenciamento em SSDs:** O escalonamento de E/S é menos crítico do que em HDDs devido à ausência de busca e latência rotacional. No entanto, o SO e o controlador do SSD precisam gerenciar a escrita de forma a otimizar o desempenho e a vida útil do dispositivo (e.g., comando TRIM para informar ao SSD quais blocos não estão mais em uso).

## Relógio (Clock)

O relógio (ou temporizador) é um dispositivo de hardware crucial que gera interrupções em intervalos regulares. O SO usa essas interrupções para:

- **Manter a hora do dia.**
- **Prevenir que um processo monopolize a CPU:** Implementar o quantum de tempo em algoritmos de escalonamento preemptivo como Round Robin.
- **Agendar eventos futuros:** Por exemplo, acordar processos que estão dormindo por um certo período.
- **Coletar estatísticas de uso do sistema.**

## Interfaces com Usuário

O subsistema de E/S também gerencia a interação com o usuário através de dispositivos como:

- **Teclado e Mouse:** Dispositivos de caractere que geram interrupções a cada evento (tecla pressionada, movimento do mouse). O SO converte esses eventos em entrada para as aplicações.
- **Monitor (Display):** O SO gerencia o frame buffer (memória de vídeo) para exibir texto e gráficos. Em sistemas com interface gráfica (GUI), o gerenciador de janelas é uma parte importante do software de E/S no espaço do usuário.

## Gerenciamento de Energia

Com a proliferação de dispositivos móveis e a crescente preocupação com o consumo de energia, o gerenciamento de energia tornou-se uma função importante do SO, especialmente em relação aos dispositivos de E/S.

- **Estados de Energia:** Muitos dispositivos e componentes do sistema (CPU, discos, interfaces de rede) podem operar em diferentes estados de energia (e.g., ativo, ocioso, dormindo) para economizar energia.
- **Políticas de Gerenciamento de Energia:** O SO implementa políticas para decidir quando mover componentes para estados de baixo consumo (e.g., desligar o disco após um período de inatividade) e quando acordá-los. Isso envolve um trade-off entre economia de energia e latência de resposta (tempo para acordar o dispositivo).
- **ACPI (Advanced Configuration and Power Interface):** Um padrão aberto que permite ao SO controlar o gerenciamento de energia do hardware.

## Conclusão

O gerenciamento de Entrada e Saída é uma das funções mais complexas e críticas de um sistema operacional. Ele abrange desde a interação de baixo nível com o hardware de E/S, através de controladores e interrupções, até a apresentação de uma interface abstrata e uniforme para as aplicações. O uso de camadas de software, drivers de dispositivo, buffering, caching e spooling são técnicas essenciais para gerenciar a diversidade de dispositivos e otimizar o desempenho. O gerenciamento eficiente de discos e SSDs, o uso do relógio para temporização e escalonamento, a interação com o usuário e o gerenciamento de energia são todos aspectos vitais do subsistema de E/S. Uma compreensão sólida desses princípios é fundamental para o desenvolvimento de sistemas operacionais eficientes e aplicações responsivas.

## Referências

- Silberschatz, A., Galvin, P. B., & Gagne, G. (2018). *Operating System Concepts* (10th ed.). Wiley.
- Tanenbaum, A. S., & Bos, H. (2015). *Modern Operating Systems* (4th ed.). Pearson Education.
- Stallings, W. (2018). *Operating Systems: Internals and Design Principles* (9th ed.). Pearson.
- Arpaci-Dusseau, R. H., & Arpaci-Dusseau, A. C. (2018). *Operating Systems: Three Easy Pieces*. Arpaci-Dusseau Books. (Capítulos sobre E/S e Dispositivos)

### Isenção de Responsabilidade:

Os autores deste documento não reivindicam a autoria do conteúdo original compilado das fontes mencionadas. Este documento foi elaborado para fins educativos e de referência, e todos os créditos foram devidamente atribuídos aos respectivos autores e fontes originais.

Qualquer utilização comercial ou distribuição do conteúdo aqui compilado deve ser feita com a devida autorização dos detentores dos direitos autorais originais. Os compiladores deste documento não assumem qualquer responsabilidade por eventuais violações de direitos autorais ou por quaisquer danos decorrentes do uso indevido das informações contidas neste documento.

Ao utilizar este documento, o usuário concorda em respeitar os direitos autorais dos autores originais e isenta os compiladores de qualquer responsabilidade relacionada ao conteúdo aqui apresentado.