

# COMPONENTES DE UM COMPUTADOR

O processador, que executa as instruções; a memória, que armazena dados e programas temporariamente; o armazenamento, que guarda informações permanentemente; e a placa-mãe, que conecta todos esses componentes.

## PROCESSADOR / CPU

O processador (ou CPU - Unidade Central de Processamento) é o cérebro do computador. Ele é responsável por executar as instruções de programas e coordenar as atividades de outros componentes da máquina. Ele realiza operações seguindo uma sequência de comandos chamada de código de máquina. Sua potência está diretamente relacionada à quantidade de núcleos e à frequência de processamento.

Existem diferentes arquiteturas de processadores, que determinam como a CPU entende e executa as instruções. As mais comuns são x86, x64 e ARM. A principal diferença entre elas está no formato e na capacidade de processar dados, o que afeta o desempenho, a compatibilidade de software e o consumo de energia.

**X86 e X64:** são amplamente usadas em computadores pessoais e servidores. Ambas foram desenvolvidas pela Intel, mas com algumas diferenças:

**x86:** É uma arquitetura de 32 bits, o que significa que o processador pode processar dados de 32 bits de cada vez. Isso limita a quantidade de memória que o sistema pode endereçar, geralmente até 4 GB de RAM. Apesar de ser mais antiga e menos eficiente para tarefas que exigem grande poder de processamento ou memória, ainda é encontrada em muitos sistemas legados e em dispositivos mais simples.

**x64:** É uma arquitetura de 64 bits, mais moderna e capaz de processar dados de 64 bits por vez. Isso permite que o processador acesse uma quantidade maior de memória, suportando até vários terabytes de RAM, o que a torna ideal para computadores modernos, servidores e dispositivos que exigem alto desempenho e maior capacidade de memória.

### ARM

É uma das mais populares em dispositivos móveis, mas também está ganhando espaço em servidores e computadores pessoais. Sua principal característica é a eficiência energética, o que os torna ideais para dispositivos que dependem de baterias. Além disso, a ARM é uma arquitetura RISC (Reduced Instruction Set Computing), o que significa que ela tem um conjunto mais simples de instruções, o que pode levar a um desempenho mais eficiente em certas tarefas.

## MEMÓRIA RAM

A memória RAM (Memória de Acesso Aleatório) é responsável por armazenar temporariamente dados e instruções que o processador precisa acessar rapidamente. Ela é chamada de "memória volátil", o que significa que os dados armazenados nela são apagados assim que o dispositivo é desligado.

A RAM funciona como um espaço temporário de armazenamento de dados de acesso rápido, diferente do armazenamento permanente, como discos rígidos (HDs) ou SSDs. Quando você abre um programa ou carrega um arquivo, as informações são carregadas da memória de armazenamento permanente para a memória RAM, permitindo que o processador acesse e execute essas informações de maneira muito mais rápida. Quanto maior a quantidade de memória RAM disponível, mais dados podem ser armazenados nela de forma rápida, o que melhora o desempenho geral do sistema, especialmente ao trabalhar com programas pesados ou ao executar várias tarefas ao mesmo tempo (multitarefa).

### Como a memória RAM funciona

A RAM é acessada diretamente pelo processador e pode ler ou gravar dados rapidamente, com um tempo de resposta muito menor que o de outros tipos de armazenamento, como o HD ou o SSD. Cada vez que o processador executa uma instrução ou precisa de dados, ele procura primeiramente na memória RAM, pois ela oferece um desempenho muito superior em relação a outras formas de armazenamento.

Por exemplo, ao abrir um programa como um navegador de internet, ele é carregado na memória RAM para que você consiga usá-lo de forma rápida. Se você tiver apenas uma quantidade limitada de RAM, o sistema pode ter que usar o disco rígido (ou o SSD) como uma memória "virtual" para compensar, o que é muito mais lento.

**Memória Virtual** é o conceito geral de ampliar a capacidade da memória do sistema, permitindo que o computador use o disco rígido (ou SSD) para simular mais RAM. Isso é feito para que o sistema consiga rodar programas que exigem mais memória do que a RAM disponível. A memória virtual, então, é a ideia de usar uma parte do disco rígido/SSD como "extensão" da RAM.

**Swap** (ou arquivo de paginação, em inglês *paging file*) é a **área do disco rígido/SSD** onde os dados que não estão sendo usados ativamente são armazenados para liberar espaço na RAM. O sistema operacional move parte dos dados da RAM para o swap para liberar espaço, e quando necessário, esses dados podem ser carregados de volta para a RAM.

## ARMAZENAMENTO

HD (Disco Rígido) e SSD (Unidade de Estado Sólido) são dois tipos de dispositivos de armazenamento usados em computadores, laptops e outros dispositivos eletrônicos para guardar dados de forma permanente. Ambos têm a mesma função básica — armazenar arquivos, programas e o sistema operacional — mas funcionam de maneiras diferentes.

### HD (Disco Rígido)

O HD, ou HDD (Hard Disk Drive, em inglês), é um dispositivo de armazenamento magnético que utiliza discos giratórios (chamados de pratos) e cabeças de leitura/gravação para acessar os dados. Ele é um dos tipos mais antigos de armazenamento, mas ainda é amplamente usado devido ao seu custo mais baixo em comparação com os SSDs.

Funcionamento: O HD possui discos internos que giram a altas velocidades enquanto uma cabeça magnética lê e grava os dados. Como o HD tem partes móveis, ele é mais suscetível a danos por quedas ou choques. Além disso, o tempo de acesso e a velocidade de leitura/gravação são muito mais lentos do que os de um SSD. O HD tem um custo por gigabyte mais baixo, o que o torna uma boa escolha para quem precisa de muito espaço de armazenamento por um preço mais acessível.

### SSD (Unidade de Estado Sólido)

O SSD é um tipo de armazenamento mais moderno, que utiliza memória flash para armazenar os dados, o que significa que ele não possui partes móveis, como o HD. Em vez disso, os dados são armazenados em chips de memória flash, de forma semelhante ao que ocorre em pen drives e cartões de memória. Os SSDs costumam ser mais caros do que os HDs, especialmente em capacidades maiores. Além disso, a vida útil dos SSDs pode ser limitada devido ao número finito de ciclos de gravação das células de memória. O SSD é muito mais rápido que o HD e como não tem partes móveis, também é mais resistente a choques e quedas.

# BIOS

A BIOS (Basic Input/Output System) é um software fundamental que fica armazenado em um chip de memória dentro da placa-mãe de um computador. Ela inicia o hardware do sistema e prepara o computador para a execução do sistema operacional. A BIOS é o primeiro programa a ser executado quando o computador é ligado, e sua função principal é garantir que todos os componentes do sistema estejam funcionando corretamente antes que o sistema operacional assuma o controle.

## Principais funções da BIOS

### - Inicialização do Sistema (POST):

A BIOS executa o POST (Power-On Self Test), que é um processo de verificação dos principais componentes do computador, como a memória RAM, o processador, o teclado, o disco rígido e outros dispositivos essenciais. Se algum componente estiver com defeito, a BIOS normalmente emite um código de erro ou som de beep para indicar o problema.

### - Carregamento do Sistema Operacional (Boot):

Após o POST, a BIOS localiza o setor de inicialização do sistema operacional (geralmente no disco rígido ou SSD) e o carrega na memória RAM para iniciar o processo de boot (inicialização do sistema).

### - Configuração de Hardware:

A BIOS fornece uma interface para que o usuário configure parâmetros do sistema, como a ordem de inicialização dos dispositivos (por ex., se o computador deve iniciar primeiro pelo disco rígido, unidade USB, CD/DVD, etc.), a configuração de data e hora, as configurações de energia e as características do processador e memória.

### - Interface com o Hardware:

Ela também fornece uma camada básica de comunicação entre o sistema operacional e o hardware, utilizando rotinas e drivers básicos para o funcionamento de dispositivos como o teclado, o mouse, e outros periféricos.

### - Acesso a Configurações Avançadas:

A BIOS oferece um menu chamado Setup, acessado normalmente pressionando uma tecla (como F2, DEL, ou ESC) durante a inicialização do sistema. Nesse menu, o usuário pode modificar configurações de hardware, como a frequência do processador, o gerenciamento de energia, a ordem de inicialização e outras opções.

## BIOS != UEFI

A UEFI (Unified Extensible Firmware Interface) é uma versão mais moderna e avançada da BIOS. Ela foi projetada para superar várias limitações da BIOS tradicional, oferecendo suporte para discos rígidos maiores, tempos de inicialização mais rápidos e uma interface gráfica mais amigável.

### Diferenças entre BIOS e UEFI:

- **Tamanho do disco:** A BIOS tradicional tem um limite para a inicialização de discos grandes, enquanto a UEFI pode lidar com discos muito maiores.
- **Interface gráfica:** A BIOS tradicional geralmente tem uma interface de texto, enquanto a UEFI pode ter uma interface gráfica mais moderna e amigável.
- **Velocidade de inicialização:** A UEFI oferece tempos de inicialização mais rápidos do que a BIOS tradicional.
- **Segurança:** A UEFI possui recursos de segurança mais avançados, como o Secure Boot, que ajuda a proteger o computador contra malware que possa tentar se infiltrar no processo de inicialização.

# SISTEMAS OPERACIONAIS

É um conjunto de programas que gerencia os recursos de hardware e software de um computador, permitindo que o usuário interaja com o dispositivo de maneira eficiente. Ele serve de interface entre o hardware e os programas de aplicação (como navegadores, editores de texto, jogos, etc).

## Funções Principais do Sistema Operacional

### - Gerenciamento de Processos:

O SO gerencia a execução de processos, que são programas em execução no computador. Ele garante que o processador seja alocado de maneira eficiente, controlando o tempo que cada processo deve usar e possibilitando que múltiplos processos sejam executados simultaneamente (multitarefa).

### - Gerenciamento de Memória:

O sistema operacional controla o uso da memória RAM, alocando espaço para os processos em execução e liberando essa memória quando os processos são finalizados. Ele também pode gerenciar o uso da memória virtual (usando o disco rígido como memória adicional) quando a RAM física está cheia.

### - Gerenciamento de Armazenamento:

O SO controla a leitura e a gravação de dados nos dispositivos de armazenamento, como discos rígidos (HD), unidades de estado sólido (SSD), CDs, DVDs, entre outros. Ele organiza os arquivos e diretórios, além de garantir que os dados sejam armazenados de forma segura e eficiente.

### - Gerenciamento de Dispositivos de Entrada e Saída (E/S):

O sistema operacional gerencia os dispositivos periféricos conectados ao computador, como teclado, mouse, impressora, scanner, monitores, etc. Ele permite que os dispositivos de entrada enviem informações para o computador e que os dispositivos de saída recebam informações.

### - Interface com o Usuário:

O SO fornece uma interface (shell) que permite a interação do usuário com o computador. Isso pode ser feito através de interfaces gráficas (GUI), como janelas, ícones e menus, ou por meio de interfaces de linha de comando (CLI), onde o usuário digita comandos diretamente.

### **- Segurança e Controle de Acesso:**

O sistema operacional gerencia a segurança do sistema, garantindo que os usuários só possam acessar recursos aos quais têm permissão. Isso envolve controle de acesso a arquivos e pastas, autenticação de usuários (como senhas), e proteção contra malware e outras ameaças.

### **- Comunicação entre Sistemas (Rede):**

O SO gerencia a comunicação entre o computador e outros dispositivos na rede, permitindo que o computador acesse a Internet, compartilhe arquivos, e interaja com outros dispositivos na mesma rede.

## **KERNEL**

Kernel é o núcleo do sistema operacional, responsável por gerenciar os recursos de hardware do computador e servir como intermediário entre o software e o hardware. Ele controla o acesso dos programas ao processador, à memória, aos dispositivos de entrada/saída (E/S) e a outros recursos do sistema.

### **Funções Principais do Kernel**

#### *Gerenciamento de Processos:*

O kernel é responsável por criar, agendar e finalizar processos (programas em execução). Ele garante que os processos sejam executados de maneira eficiente e aloca recursos, como o tempo do processador, de forma justa entre os diferentes processos.

#### *Gerenciamento de Memória:*

O kernel gerencia o uso da memória RAM, alocando e liberando memória para os processos conforme necessário. Ele também pode usar técnicas de memória virtual para expandir o espaço disponível, utilizando o disco rígido como uma "memória extra" quando a RAM estiver cheia.

#### *Gerenciamento de Dispositivos de Entrada e Saída (E/S):*

O kernel controla os dispositivos periféricos do sistema, como o teclado, mouse, disco rígido, impressoras e outros. Ele facilita a comunicação entre o hardware e os programas, permitindo que os dispositivos recebam e enviem dados.

#### *Controle de Acesso e Segurança:*

O kernel garante que apenas os processos e usuários autorizados tenham acesso a recursos críticos, como arquivos e dispositivos. Ele define permissões de acesso e implementa políticas de segurança para proteger o sistema contra acessos não autorizados e malwares.

### *Comunicação entre Processos:*

O kernel também fornece mecanismos para que os processos possam se comunicar entre si de forma segura e eficiente, como as chamadas de sistema para troca de mensagens ou dados.

## **Modos de Acesso**

Referem-se aos privilégios de execução de um programa garantindo que a memória não seja invadida por outro programa, por exemplo. Os modos de acesso são realizados de duas formas:

### *Modo usuário*

Os programas podem executar ações sem privilégios, como leitura de um arquivo.

### *Modo kernel*

O acesso pode ser realizado em modo usuário ou em modo com privilégio total no computador, como acesso ao disco.

## **Principais Modelos de Arquiteturas dos Sistemas Operacionais**

### *Monolítica*

Nesse modelo, o sistema operacional é composto por um único núcleo (kernel) que gerencia todos os aspectos do sistema, como gerenciamento de processos, memória, sistemas de arquivos e controle de dispositivos. Ex.: Linux, UNIX.

### *Microkernel*

O núcleo do sistema operacional é reduzido ao mínimo. A maior parte dos serviços (como gerenciamento de arquivos e redes) é executada fora do núcleo, em espaços de usuário.

### *Arquitetura de Camadas*

É organizado em camadas ou níveis, onde cada camada é responsável por uma parte do sistema, como a camada de gerenciamento de memória, a camada de gerenciamento de processos, etc.



# Time-Sharing System

Um **Sistema Operacional de Tempo Compartilhado** (ou **Time-Sharing System**) é um tipo de sistema operacional que permite que múltiplos usuários ou processos compartilhem os recursos de um computador de forma eficiente e simultânea. A principal característica desses sistemas é que o processador (CPU) é compartilhado entre diferentes tarefas ou usuários, de forma que cada um tem a sensação de estar sendo atendido ao mesmo tempo, mesmo que o computador tenha apenas um único processador.

## Como funciona

Imagine que você tem um computador com apenas um processador (CPU) e várias pessoas ou programas precisando usar esse processador. Se não houvesse um sistema de tempo compartilhado, o processador ficaria ocupado com uma única tarefa por vez, o que faria com que os outros processos ou usuários ficassem esperando.

Em um **sistema de tempo compartilhado**, a solução é dividir o tempo de CPU em pequenos **intervalos de tempo** chamados de **quanta de tempo** (ou **time quantum**). Cada processo ou usuário recebe um tempo fixo para usar o processador, e depois o sistema passa para o próximo processo. Isso acontece tão rapidamente que os usuários e os processos têm a impressão de que estão sendo atendidos simultaneamente.

## Como é feita a divisão de tempo

1. **Processos ou usuários:** Cada processo ou usuário que está utilizando o sistema vai ter um **tempo de CPU** alocado, chamado **quantum**.
2. **Quantum (ou fatia de tempo):** é um pequeno intervalo de tempo durante o qual o processador é dedicado a um único processo. Pode ser algo como 10 milissegundos ou menos, dependendo do sistema operacional.
3. **Escalonamento (Scheduling):** O sistema operacional usa um **algoritmo de escalonamento** para decidir qual processo deve ser executado em um dado momento e por quanto tempo. Após o quantum de tempo ser usado, o processador é "cedido" ao próximo processo ou usuário na fila, e o processo anterior é colocado de volta na fila para ser executado quando chegar a sua vez novamente.

## Vantagens dos Sistemas de Tempo Compartilhado

1. **Multitarefa interativa:** Com o tempo compartilhado, vários processos ou usuários podem usar o sistema ao mesmo tempo, mas cada um tem a impressão de que está usando o computador exclusivamente.
2. **Eficiência no uso da CPU:** Em sistemas sem tempo compartilhado, o processador pode ficar ocioso ou muito ocupado com uma única tarefa. Com o tempo compartilhado, a CPU é utilizada de maneira mais eficiente, pois ela está sempre alternando entre os processos, sem deixar ocioso.

## Algoritmos de Escalonamento (Scheduling)

O escalonamento é o processo usado pelo sistema operacional para decidir qual processo ou thread deve ser executado em seguida. Alguns algoritmos de escalonamento comuns são:

1. **First-Come, First-Served (FCFS):** O primeiro processo a chegar é o primeiro a ser executado.
2. **Round Robin (RR):** O processador vai alternando entre os processos em ordem circular, dando a cada um uma fatia de tempo fixa.
3. **Shortest Job Next (SJN):** O processo que tem o menor tempo de execução estimado é executado primeiro.

O **Round Robin (RR)** é o mais associado aos sistemas de tempo compartilhado, pois ele garante que todos os processos ou usuários recebam uma fatia de tempo do processador.

# THREADS

Uma **thread** é uma unidade de execução dentro de um processo. Um **processo** é, basicamente, um programa em execução. Dentro de um processo, você pode ter várias threads, que são "linhas" independentes de execução que compartilham recursos do mesmo processo.

## Diferença entre Processo e Thread

**Processo:** É uma instância de um programa em execução. Cada processo tem seu próprio espaço de memória e recursos. Quando você executa um programa, ele se torna um processo. Exemplo: um navegador de internet aberto é um processo.

**Thread:** É uma unidade de execução dentro de um processo. Quando um processo é criado, ele geralmente começa com uma thread principal. Se esse processo for projetado para dividir o trabalho, ele pode criar outras threads para executar tarefas simultaneamente. Exemplo: no navegador, a thread principal pode ser responsável por carregar a página, enquanto outras threads podem ser responsáveis por fazer download de imagens, tocar vídeos ou processar o histórico.

Metáfora: O Processo seria uma empresa inteira, com seus recursos e a capacidade de dividir seu trabalho entre diferentes departamentos (as threads). E uma Thread seria o departamento (ou uma função dentro da empresa), responsável por uma tarefa específica dentro do processo.

## Deadlock (Interbloqueio)

Um **deadlock** ocorre quando dois ou mais processos (ou threads) ficam **bloqueados** esperando um pelo outro, de forma que nenhum deles pode continuar a execução. Ou seja, há um ciclo de dependências em que cada processo espera um recurso que é mantido por outro processo do ciclo, e nenhum deles pode seguir.

## Race Condition (Condição de Corrida)

Ocorre quando dois ou mais processos ou threads tentam acessar e modificar dados compartilhados ao mesmo tempo, e o resultado da operação depende da ordem em que as threads executam as ações. Isso pode levar a comportamentos inesperados ou incorretos. Ex.: suponha que dois processos A e B estão tentando atualizar um valor compartilhado. O processo A lê o valor, faz uma alteração e o escreve de volta, mas antes que o processo A termine, o processo B também lê o valor, faz sua própria alteração e escreve o valor de volta. O problema é que ambos os processos podem sobrescrever as alterações do outro, causando um erro.

# NÚCLEO

Um **núcleo** (ou **core**, em inglês) é a parte de um processador (CPU) responsável por executar instruções de programas.

## Processador e Núcleos

Um processador (ou CPU) é o cérebro do computador, responsável por executar as instruções dos programas. O processador pode ter um ou mais núcleos.

- **Processador de 1 núcleo:** um processador com apenas 1 núcleo pode executar uma tarefa por vez. Ou seja, ele realiza uma tarefa e, quando termina, começa a próxima.
- **Processador de múltiplos núcleos:** Quando o processador tem mais núcleos, ele pode executar várias tarefas simultaneamente. Cada núcleo pode ser responsável por processar uma tarefa separada. Isso melhora o desempenho, especialmente em sistemas que precisam de multitarefa.

## Hyper-Threading (HT)

Com **Hyper-Threading**, um processador pode **simular** ter mais núcleos lógicos. Por exemplo, um processador com 4 núcleos físicos pode, com Hyper-Threading, se comportar como se tivesse 8 núcleos lógicos, já que cada núcleo físico pode processar dois threads (ou tarefas) ao mesmo tempo.