

# 1 INTRODUÇÃO

## 1.1 VISÃO GERAL DE UM SISTEMA COMPUTACIONAL

Um sistema computacional pode ser dividido em 6 camadas

- ❑ **Dispositivos Físicos** - são os equipamentos em si. Componentes como *chips* e transistores;
- ❑ **Microcódigo** - é um executor de instruções, as quais são manuseadas para interagir diretamente com os dispositivos físicos;
- ❑ **Linguagem de máquina** - é um conjunto de instruções a serem executadas pelo Microcódigo.
- ❑ **Sistema Operacional** - é um *software* utilizado para gerenciar os recursos do sistema computacional. Ele é uma interface entre o usuário e outros *softwares* e os dispositivos físicos.
- ❑ **Programas de sistema** - são *softwares* que permitem a gerência do sistema computacional (compiladores, interpretadores de comandos e editores de texto);
- ❑ **Programas de aplicação** - que permitem que o sistema computacional seja efetivamente útil ao usuário final (sistema bancário, jogos).

## 1.2 DEFINIÇÃO DE UM SISTEMA OPERACIONAL

O Sistema Operacional é um *software* que atua como intermediário entre o usuário e o computador rodando em modo *kernel* ou supervisor. Dessa maneira, ele tem acesso direto e irrestrito a todos os recursos do sistema computacional. Os demais *softwares* são executados no modo usuário, ou seja, não tem acesso direto aos recursos computacionais. Eles precisam do Sistema Operacional para utilizar a máquina.

Mas, por que esta proteção? Se o usuário tivesse acesso direto e irrestrito aos recursos da máquina, ele poderia causar danos diretos ao computador, além de usar de forma irracional os recursos disponibilizados. Além disso, manusear diretamente a máquina através da programação em linguagem de máquina é uma atividade muito complexa. Deve ser lembrado que, acima de tudo, o computador é um dispositivo eletrônico.

Dessa maneira existem duas visões e funções básicas do Sistema Operacional:

- **Máquina Estendida** - sob esta óptica, o Sistema Operacional é a representação do *hardware*. Entretanto, ele é muito mais simples e fácil de ser manuseado. Ou seja, o Sistema Operacional esconde a complexidade do *hardware* do usuário final, tornando, portanto mais conveniente o uso do computador;
- **Gerenciador de Recursos** - nesta função, o Sistema Operacional supervisiona a utilização dos recursos do Sistema Computacional. Dessa maneira, o *hardware* estará protegido da ação direta do usuário. Com um gerenciador o uso da máquina torna-se mais eficiente também, alocando os recursos conforme a necessidade demonstrada.

## 1.3 BREVE HISTÓRICO DOS SISTEMAS OPERACIONAIS

### 1.3.1 SISTEMAS BATCH

Os primeiros sistemas computacionais eram muito grandes fisicamente. A entrada e saída de dados eram feitas através de cartões perfurados e unidades de fitas magnéticas. O usuário não interagia diretamente com os sistemas. Os programadores faziam *jobs*, que eram compostos por uma pilha de cartões contendo o programa em si, os dados e algumas informações de controle. Estas pilhas eram destinadas para os operadores do computador que era responsável pelo uso do mesmo.

Naquela época, única tarefa do sistema operacional era coordenar a troca de um *job* por outro. Como forma de acelerar o processamento, os operadores reuniam os *jobs* com necessidades semelhantes em lotes (*batches*).

Neste ambiente, o nível de ociosidade do processador era muito alto, devido ao fato de que os sistemas de I/O (Entrada/Saída de dados) eram muito mais lentos que os circuitos elétricos. Muitas vezes chegava a três vezes a ordem de grandeza a diferença de velocidade.

Com o advento das tecnologias de armazenamento em disco, foi possível alocar todos os lotes em um único espaço, ao invés de um conjunto de cartões. O aspecto mais importante do escalonamento de *jobs* é a multiprogramação.

Este conceito se refere ao fato de que nenhum usuário utiliza todos os recursos do sistema computacional ao mesmo tempo. A multiprogramação aumenta a taxa de uso do processador organizando os *jobs* de maneira que o processador sempre tenha alguma tarefa a cumprir.

O processo para aumentar a taxa de uso do processador é bastante simples de entender:

- ❑ Todos os *jobs* são alocados no disco e dispostos no pool de *jobs*;

- ❑ Um deles é escolhido para entrar em execução;
- ❑ Na grande maioria das vezes, em algum momento a execução do job terá que esperar algum evento ocorrer para continuar a sua execução;
- ❑ Ao invés do processador ficar ocioso, é escolhido um outro job para ser executado;

Sem a multiprogramação, o processador não executaria outro job enquanto o que ele iniciou estiver bloqueado. Ela é uma primeira situação, onde o Sistema Operacional precisa tomar decisões no lugar do usuário. Para executar esta tarefa de forma correta, o sistema operacional deve ter uma estrutura mais sofisticada.

### 1.3.2 SISTEMAS DE TEMPO COMPARTILHADO

Os sistemas de tempo compartilhado foram desenvolvidos a partir dos sistemas batch. Ao invés do sistema executar um *job* até o seu final, o processador fica alternando a execução dos *jobs*. Ele está muito associado aos programas interativos, pois eles precisam do contato direto com o usuário.

Ele permite o processador atender a vários usuários, escalonando entre eles, dessa maneira o tempo de ociosidade diminui bastante. Isto é possível devido, como mencionado anteriormente, à lentidão dos sistemas de I/O em relação ao processador.

### 1.3.3 COMPUTADORES PESSOAIS

O avanço da tecnologia de computação fez com que aparecessem computadores menores destinados a monousuários. Entretanto, eles não tinham recursos de multiprogramação ou tempo compartilhado, o que foi revisto com o passar dos anos.

Com o advento das redes de computadores, houve a necessidade de compartilhar os recursos dos componentes das redes, e os sistemas operacionais tiveram que assimilar características que possibilitam esse compartilhamento.

Ainda teve o surgimento dos Sistemas Paralelos e Distribuídos, os quais são alvo de estudo em uma outra oportunidade.