

Aula 1 – Visão geral de Sistemas Operacionais

Objetivos

Compreender os conceitos básicos de Sistemas Operacionais.

Conhecer suas funções principais.

Analisar o Sistema Operacional como uma máquina de níveis.

1.1 Conceitos básicos

Diferentemente do que muitas pessoas imaginam, o computador não faz nada sozinho. Ele apenas processa uma série de informações inseridas pelo usuário para então fornecer os resultados. As informações inseridas e os resultados que recebemos precisam estar num formato que nós humanos conseguimos entender. Para facilitar essa comunicação entre homem e computador, foram criados os *softwares* ou programas de computador. Na realidade, tudo que fazemos com um computador é pela execução desses programas.

De acordo com um dos principais autores da área,

Um sistema operacional é um programa que atua como intermediário entre o usuário e o *hardware* de um computador. O propósito de um sistema operacional é propiciar um ambiente no qual o usuário possa executar outros programas de forma conveniente, por esconder detalhes internos de funcionamento e eficiência, por procurar gerenciar de forma justa os recursos do sistema (Silberschatz, Galvin e Gagne, 2000, p.22).

Vamos estender o conceito de **sistema operacional** ao longo do curso, mas podemos defini-lo, de forma simples, como um conjunto de rotinas executadas pelo processador com a principal função de controlar o funcionamento do computador, gerenciando os diversos recursos disponíveis no sistema. Na Figura 1.1 vemos a posição que um Sistema Operacional ou simplesmente “SO” ocupa dentre os vários elementos que compõem um sistema de computação. Você deve observar que a palavra “Usuários” está sendo usada com dois sentidos diferentes: para as pessoas que utilizam o computador e para os programas e utilitários instalados no computador.

A-Z

Sistema Operacional

Sistema Operacional: segundo o Aurélio (verbete sistema), sistema operacional é um conjunto integrado de programas básicos, projetado para supervisionar e controlar a execução de programas de aplicação em um computador.

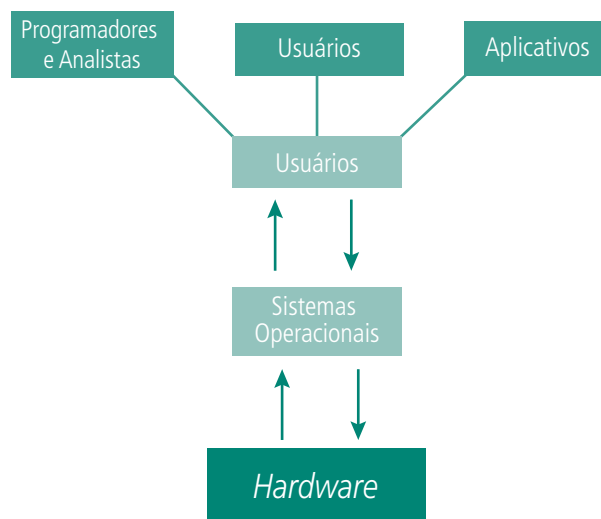


Figura 1.1: Visão do Sistema Operacional

Fonte: Adaptado de Machado, 2004



Resumidamente, o sistema operacional tem a função de proteger a máquina do usuário e proteger o usuário da máquina.

1.2 Funções principais

Na Figura 1.1 foi destacado o controle de *hardware*. Esta é uma das funções básicas do SO e pode ser desmembrada em:

a) Facilidade de acesso aos recursos do sistema

Um sistema de computação possui, normalmente, diversos componentes, como monitores, impressoras e discos rígidos. Quando utilizamos um desses dispositivos, não nos preocupamos com a maneira como é realizada esta comunicação e os inúmeros detalhes envolvidos.



Você pode obter mais informações sobre "setor" e "trilha" no livro *Organização Estruturada de Computadores*, de Andrew S. Tanenbaum, 5ª Edição, Editora Prentice-Hall, ou na própria internet em sites especializados como o "Clube do Hardware" em <http://www.clubedohardware.com.br>

Uma operação frequente como, por exemplo, a leitura de um arquivo em um CD ou disco pode parecer simples. Existe um conjunto de rotinas específicas, controladas pelo sistema operacional, que são responsáveis por acionar a cabeça de leitura e gravação da unidade de disco, posicionar na trilha e setor onde estão os dados, transferir os dados do disco para a memória e, finalmente, informar ao programa a chegada dos dados.

O sistema operacional, então serve de interface entre o usuário e os recursos de *hardware*, tornando esta comunicação transparente (ou imperceptível) e permitindo ao usuário um trabalho mais eficiente e com menos possibilidades de erros.

b) Compartilhamento de recursos de forma organizada e protegida

Se imaginarmos, por exemplo, que uma impressora pode ser utilizada por vários usuários do sistema, então deverá existir algum controle para impedir que a impressão de um usuário interrompa a impressão de outro. O sistema operacional é o responsável por permitir o acesso organizado a esse e a outros recursos disponíveis no computador.

O compartilhamento de recursos permite a diminuição de custos, na medida em que mais de um usuário pode utilizar as mesmas facilidades concorrentemente, tais como discos, impressoras, linhas de comunicação, etc. Com isto, uma mesma impressora (ou linha de comunicação ou outro recurso) pode atender a vários usuários.

Não é só no controle do acesso a hardware compartilhado que o sistema operacional atua, ele nos permite executar várias tarefas, como imprimir um documento, copiar um arquivo pela internet ou processar uma planilha, entre outros. O SO deve ser capaz de controlar a execução concorrente de todas essas tarefas. Ainda podemos dizer que, embora alguns programas sejam escritos baseados nas instruções de um determinado processador, será responsabilidade do sistema operacional executar tarefas básicas do micro, ou seja, ensinar ao processador como desenhar uma janela ou imprimir um documento.

De um modo geral, os programas que os usuários executam não são escritos para um processador, mas sim para um SO. Isto facilita a comunicação do programa com o *hardware* do computador. As tarefas são executadas pelo SO, tornando os programas menores e mais fáceis de serem programados (Machado e Maia, 2004. p.1-3).

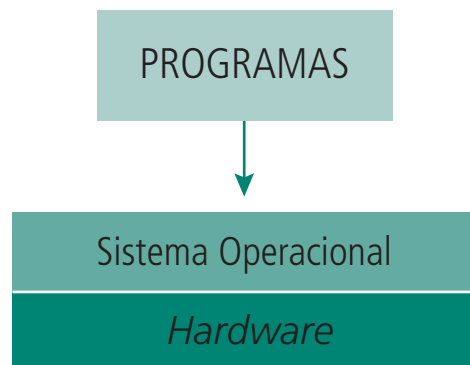


Figura 1.2: O Sistema Operacional funciona como uma interface entre o hardware e os programas de usuários

Fonte: Adaptado de Machado, 2004

Conforme mostra a Figura 1.2, o SO é o intermediário entre *hardware* e programas utilizados pelos usuários.

Em uma situação ideal, somente o sistema operacional deve ter acesso ao hardware do computador. Um programa que desejasse, por exemplo, fazer um desenho no monitor, obrigatoriamente teria de repassar esta tarefa ao sistema operacional. Este, por sua vez, iria analisar o pedido do programa e, considerando o pedido válido, o executaria. Caso um determinado programa resolvesse fazer um pedido estranho (por exemplo, apagar todos os dados do disco rígido), o SO simplesmente poderia ignorar tal pedido, terminar a execução do programa e informar a ocorrência ao usuário.



Um programa de usuário não deve acessar recursos do computador diretamente, deve antes passar pela intermediação e autorização do sistema operacional.

Essa é a condição ideal de um sistema operacional ESTÁVEL e SEGURO. Isto acontece, sobretudo, em sistemas operacionais para gerenciamento de rede local (*Windows Server, Unix e Linux*) e entre os sistemas operacionais para PCs que não foram desenvolvidos para serem servidores de rede como o *MacOS* e *Windows* nas suas versões *XP, Vista e Windows 7*.

O antigo DOS não trabalhava nessas condições. Na época em que foi criado, o PC tinha pouquíssima memória RAM (1 MB) e o sistema operacional, como ficava residente em memória, tinha de ser o menor possível. Uma solução para diminuir o tamanho do SO foi permitir aos programas que acessassem diretamente o *hardware* do micro para tarefas especiais, como desenhar gráficos ou enviar dados à impressora. A Figura 1.3 ilustra isto.

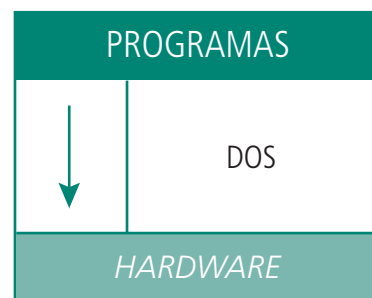


Figura 1.3: O Sistema DOS permitia acesso direto ao hardware pelos programas de usuário
Fonte: Adaptado de Machado, 2004

No entanto, isso acabava gerando um problema maior: se um programa fizesse um acesso indevido diretamente ao *hardware* do computador ou se o programa não estivesse bem escrito, isso inevitavelmente era refletido no hardware, fazendo com que o programa parasse por travamento. Esse problema continuou em versões do *Windows 3.x, 95, 98 e ME*, por utilizarem o mesmo núcleo do DOS, permitindo acessos direto ao *hardware*.

1.3 Máquina de níveis

A linguagem entendida pelo computador é uma linguagem binária de difícil entendimento pelos seres humanos, sendo chamada de linguagem de “baixo nível” ou “de máquina”. As linguagens mais próximas aos seres humanos são classificadas como linguagens de “alto nível”. Os computadores entendem apenas programas feitos em sua linguagem binária. Os seres humanos, no entanto, elaboram programas em linguagens de alto nível.

Um computador, visto somente como um gabinete composto de circuitos eletrônicos, cabos e fontes de alimentação (*hardware*), não tem nenhuma utilidade. É por meio de programas (*software*) que o computador consegue armazenar dados em discos, imprimir relatórios, gerar gráficos, realizar cálculos, entre outras funções. O *hardware* é o responsável pela execução das instruções de um programa, com a finalidade de se realizar alguma tarefa.

Nos primeiros computadores, a programação era realizada em painéis, através de fios, exigindo um grande conhecimento do *hardware* e de linguagem de máquina. Isso trazia uma grande dificuldade para os programadores da época, que normalmente eram os próprios engenheiros projetistas e construtores desses computadores.

A solução para esse problema foi o surgimento do Sistema Operacional, que tornou a interação entre usuário e computador mais simples, confiável e eficiente. A partir desse acontecimento, não existia mais a necessidade de o programador se envolver com a complexidade do *hardware* para poder trabalhar; ou seja, a parte física do computador tornou-se transparente para o usuário.

Podemos considerar o computador como uma máquina de níveis ou camadas, em que inicialmente existem dois níveis: o nível 0 (*hardware*) e o nível 1 (sistema operacional). Desta forma, o usuário pode enxergar a máquina como sendo apenas o sistema operacional, ou seja, como se o *hardware* não existisse. Esta visão modular e abstrata é chamada **máquina virtual**.

Para o sistema operacional, o programador e os programas também são usuários, pois usam recursos disponibilizados pelo SO. Em vários pontos deste texto, você poderá ver que a palavra usuário se aplica ao programador ou ao programa.

Entretanto, um computador não possui apenas dois níveis, e sim tantos níveis quantos forem necessários para adequar o usuário às suas diversas aplicações.



Existem muitas linguagens de alto nível utilizadas para os mais diversos fins, como C, C++ e Java. Os códigos-fontes escritos nessas linguagens precisam ser convertidos em linguagem binária. Por exemplo, a linguagem C utiliza uma forma de conversão diferente da utilizada pela linguagem Java.



Quando o usuário está trabalhando em um desses níveis, não necessita saber da existência das outras camadas, acima ou abaixo de sua máquina virtual.

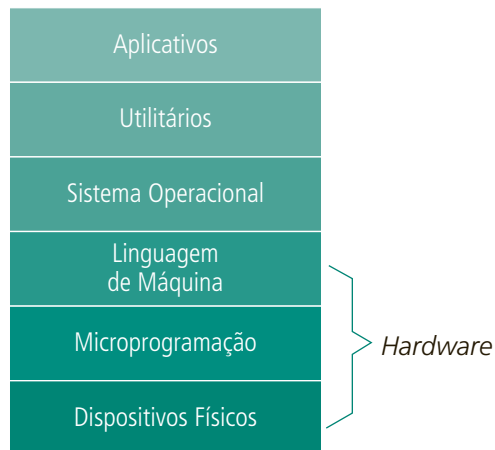


Figura 1.4: O computador como máquina de níveis

Fonte: Adaptado de Machado, 2004

Atualmente, a maioria dos computadores possui a estrutura mostrada na Figura 1.4, podendo conter mais ou menos camadas. A linguagem utilizada em cada um desses níveis é diferente, variando da mais elementar (baixo nível) à mais sofisticada (alto nível). Os aplicativos são programas executados pelo usuário. Os utilitários são programas de uso genérico e frequente, geralmente fornecidos junto com o SO. Cada um desses níveis será abordado com mais detalhes nas próximas aulas.



Um sistema operacional pode então ser definido sob dois aspectos: como uma máquina estendida ou máquina virtual e como um gerenciador de recursos.

O sistema operacional, como uma máquina virtual, esconde do programador detalhes do *hardware*, apresentando uma visão simples, mais conveniente e mais fácil de utilizar.

O sistema operacional, como um gerenciador de recursos, fornece uma alocação controlada e ordenada dos recursos do computador entre os vários programas que competem por esses recursos. Os recursos incluem processadores, memórias, dispositivos de E/S (unidades de disco, impressoras, *mouse*, etc.), interfaces de rede, dentre outros.

Quando um computador tem vários usuários, existe a necessidade de se proteger a memória, os dispositivos de E/S e os outros recursos. O sistema operacional então mantém informação sobre quem está usando qual recur-

so (para garantir os recursos a quem precisa deles), contabilizar o uso (para evitar que um usuário use por um período injustamente longo) e mediar quando há pedidos conflitantes sobre um mesmo recurso.

Aula 2 – Histórico e classificação

Objetivos

Conhecer o histórico de Sistemas Operacionais.

Compreender como as inovações de *hardware* colaboraram com o desenvolvimento dos Sistemas Operacionais.

Classificar os sistemas conforme suas características principais.

2.1 Histórico

Vimos que o sistema operacional interage diretamente com o *hardware* e, com isso, é influenciado diretamente pela evolução do mesmo. Portanto, a evolução dos sistemas operacionais está, em grande parte, relacionada ao desenvolvimento de equipamentos cada vez mais velozes, compactos e de custos baixos e à necessidade de aproveitamento e controle destes recursos. Assim, ao falar sobre o histórico dos sistemas operacionais, estaremos recordando um pouco a evolução do *hardware*. Devemos lembrar que as datas das fases da evolução são aproximadas.

Desde os tempos do computador programado por chaves e cabos até o surgimento do teclado e impressora de caracteres, procurou-se ao longo do processo evolutivo do computador tornar a sua utilização mais amigável, precisa, rápida e eficaz.

O conjunto de equipamentos e recursos utilizados para que o homem possa controlar o computador é genericamente denominado interface. O aprimoramento da interface atingiu o ponto em que o usuário passou a interagir com pequenos desenhos ou símbolos de objetos comuns ao seu trabalho. Pensou-se em representar, por exemplo, a tarefa de impressão de documentos pelo desenho de uma pequena impressora e a eliminação de um documento por uma pequena lixeira. Surgiram os ícones. O histórico da evolução foi dividido em fases, cada uma marcada pela evolução significativa do *hardware*, do *software*, da interação com o sistema ou por aspectos de conectividade. Primeiramente, devemos ressaltar que o mapeamento das datas de evoluções

e gerações dos Sistemas Operacionais e das Arquiteturas de Computadores são, de certa forma, vagas e imprecisas, mas com certa estrutura.

a) Primeira fase (1945-1955) - Válvulas e Painéis de Programação

No início da Segunda Guerra Mundial, surgiram os primeiros computadores digitais, formados por milhares de válvulas, que ocupavam áreas enormes, sendo de funcionamento lento e duvidoso.

O ENIAC (*Electronic Numerical Integrator and Computer*) foi o primeiro computador digital de propósito geral. Criado para a realização de cálculos balísticos, sua estrutura possuía 17.468 válvulas, 10 mil capacitores, 70 mil resistores e pesava 32 toneladas. Quando em operação era capaz de realizar cinco mil adições por segundo.

A programação era feita em painéis, através de fios, utilizando linguagem de máquina. Não existia o conceito de sistema operacional. Outros computadores foram construídos nessa época, mas eram utilizados apenas em universidades e órgãos militares.

Muitas empresas foram fundadas ou investiram no setor, como, por exemplo, a IBM, o que levou à criação dos primeiros computadores para aplicações comerciais.

b) Segunda fase (1956-1965) - Transistores e Sistemas em Lote (*batch*)

A criação do transistor e das memórias magnéticas contribuiu para o enorme avanço dos computadores da época. O transistor permitiu o aumento da velocidade e da confiabilidade do processamento; as memórias magnéticas permitiram o acesso mais rápido aos dados, maior capacidade de armazenamento e computadores menores.

Surgiram os primeiros sistemas operacionais, para tentar automatizar as tarefas manuais até então realizadas e as primeiras linguagens de programação, como *Assembly* e *Fortran*. Os programas deixaram de ser feitos diretamente no *hardware*, o que facilitou enormemente o processo de desenvolvimento de programas. Surgiu o processamento em *batch*, em que um lote (*batch*) de programas e de dados era submetido ao computador por vez.

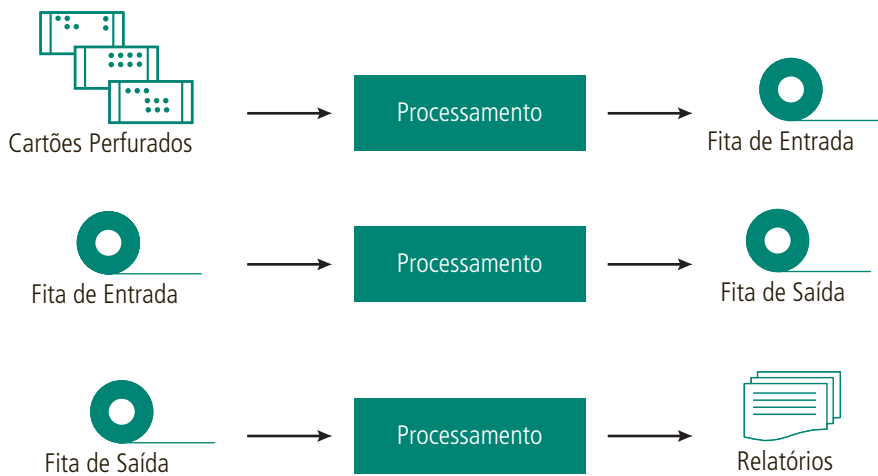


Figura 2.1: Ciclos de processamento na segunda fase

Fonte: Adaptado de Machado, 2004

Os programas passaram a ser perfurados em cartões que, submetidos a uma leitora, eram processados e gravados em uma fita de entrada, conforme Figura 2.1. A fita de entrada, então, era lida pelo computador, que executava um programa de cada vez, processando e gravando o resultado em uma fita de saída. Ao término de todos os programas, as fitas de saída eram lidas e processadas novamente para serem impressas, gerando assim os relatórios.

Com o processamento em *batch*, um grupo de programas era submetido de uma só vez, o que diminuía o tempo existente entre a execução dos programas, permitindo, assim, melhor uso do computador.

c) Terceira fase (1966-1980) - Circuitos Integrados e Multiprogramação

Por meio dos circuitos integrados e, posteriormente, dos microprocessadores, foi possível viabilizar e difundir o uso de sistemas computacionais por empresas, devido à diminuição de seus custos de aquisição. Houve um aumento no poder de processamento e diminuição no tamanho dos equipamentos.

A evolução dos processadores de entrada/saída permitiu que, enquanto um programa esperasse por uma operação de leitura/gravação, o processador executasse um outro programa. Para tal, a memória foi dividida em partições, em que um programa esperava sua vez para ser processado. A essa técnica de compartilhamento da memória principal e processador deu-se o nome de **multiprogramação**.

A-Z

Multiprogramação

Multiprogramação é a execução simultânea de dois ou mais programas.



Duas inovações de *hardware* foram fundamentais para o surgimento da multiprogramação: os discos magnéticos e as interrupções de *hardware*. Os discos magnéticos compõem praticamente todas as máquinas atuais, com vários *Gigas* e até *Terabytes* de capacidade de armazenamento. As interrupções de *hardware* são sinais que as controladoras de dispositivos enviam à CPU para avisar que as operações de entrada ou saída foram finalizadas ou tiveram algum problema.



Nos sistemas *time-sharing*, os usuários possuíam um terminal que podia interagir com o programa em execução. Esses usuários tinham a ilusão de possuir a máquina dedicada à execução de seu programa. O que não era verdade! Essa ilusão vinha da divisão de tempo de processamento de CPU entre os usuários



Existem na internet muitas informações sobre o sistema UNIX, sistema pai de muitos sistemas operacionais atuais, como o *Linux*, *Solaris* e outros. Procure o vídeo na internet chamado de "Revolução dos Sistemas Operacionais", do Inglês "*Revolution OS*", que conta um pouco dessa história.

Com a substituição das fitas por discos no processo de submissão dos programas, o processamento *batch* tornou-se mais eficiente, pois permitia a alteração na ordem de execução das tarefas, até então somente sequencial. A essa técnica de submissão de programas chamou-se *spooling*, que, mais tarde, também viria a ser utilizada no processo de impressão.

Os sistemas operacionais, mesmo com a evolução do processamento *batch* e a multiprogramação, ainda estavam limitados a processamentos que não exigiam comunicação com o usuário. Para permitir a interação rápida entre o usuário e o computador, foram adicionados terminais de vídeo e teclado (interação *on-line*).

A multiprogramação evoluiu, preocupada em oferecer aos usuários tempos de resposta razoáveis e uma interface cada vez mais amigável. Para tal, cada programa na memória utilizaria o processador em pequenos intervalos de tempo. A esse sistema de divisão de tempo do processador chamou-se *time-sharing* (tempo compartilhado).

Outro fato importante nessa fase foi o surgimento do sistema operacional UNIX.

Ao final dessa fase, com a evolução dos microprocessadores, surgiram os primeiros microcomputadores, muito mais baratos que qualquer um dos computadores até então comercializados.

d) Quarta fase (1981-1990) - Computadores Pessoais

Os mini e superminicomputadores se firmaram no mercado e os microcomputadores ganharam um grande impulso. Surgem as estações de trabalho (*workstations*) que, apesar de monousuárias, permitem que se executem diversas tarefas concorrentemente, criando o conceito de multitarefa.

No final dos anos 80 os computadores tiveram um grande avanço, decorrente de aplicações que exigiam um enorme volume de cálculos. Para acelerar o processamento, foram adicionados outros processadores, exigindo dos sistemas operacionais novos mecanismos de controle e sincronismo. Com o multiprocessamento, foi possível a execução de mais de um programa simultaneamente, ou até de um mesmo programa por mais de um processador. Foram introduzidos processadores vetoriais e técnicas de paralelismo de processamento, fazendo com que os computadores se tornassem ainda mais poderosos.

O uso das redes distribuídas se difundiu por todo o mundo, permitindo o acesso a outros sistemas de computação, independentemente de cidade, país e, até mesmo, fabricante. Os *softwares* de rede passaram a estar intimamente relacionados com o sistema operacional de cada máquina e surgem os sistemas operacionais de rede.

e) Quinta fase (1991-2000)

Houve grandes avanços em termos de *hardware*, *software* e telecomunicações como consequência da evolução das aplicações, que necessitavam cada vez mais de capacidade de processamento e armazenamento de dados. Sistemas especialistas, sistemas multimídia, bancos de dados distribuídos, inteligência artificial e redes neurais são apenas alguns exemplos da necessidade cada vez maior de informação e de capacidade de processamento.

O conceito de processamento distribuído é explorado nos sistemas operacionais, de forma que suas funções estejam espalhadas por vários processadores através de redes de computadores.

A década de 90, foi definitiva para a consolidação dos sistemas operacionais baseados em interfaces gráficas (TANENBAUM, 2000, p. 4 a 12).

Atualmente temos as plataformas *multicore* (vários núcleos) disponíveis em computadores pessoais e até *notebooks*, possibilitando cada vez mais a paralelização de aplicações e elevando, em muito, o tempo de execução dos programas. Os fanáticos pelos jogos de computador que o digam!

Vamos relembrar as principais características e evoluções de cada fase? Faça um quadro, resumindo cada uma das cinco fases, para ajudá-lo no estudo e entendimento da evolução de *hardware* necessária para o desenvolvimento dos sistemas operacionais. Para cada fase, anote os acontecimentos mais importantes.



2.2 Tipos de Sistemas Operacionais

Os tipos de sistemas operacionais e sua evolução estão relacionados diretamente com a evolução do *hardware* e das aplicações por ele suportadas e podem ser classificados conforme Figura 2.2.

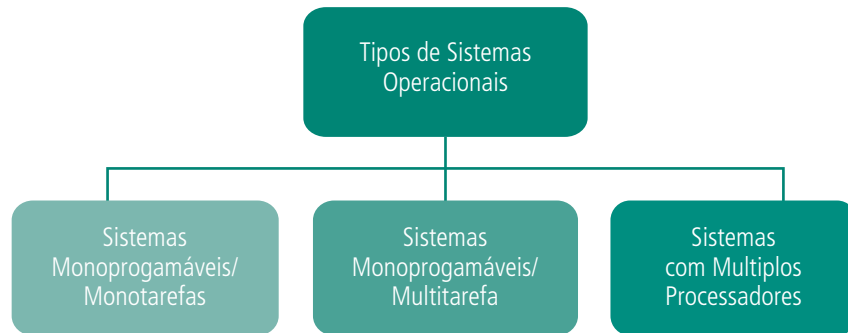


Figura 2.2: Tipos de sistemas operacionais

Fonte: Adaptado de Macha, 2004

Considerando o processamento, podemos classificar os sistemas operacionais de acordo com a quantidade de tarefas que podem ser executadas simultaneamente.

A-Z

Monoprogramáveis ou Monotarefa

Sistemas monoprogramáveis ou monotarefa são aqueles em que é executado, por vez, um único programa ou uma única tarefa.

Monoprogramáveis ou **Monotarefa** podem executar apenas um programa por vez. Para que um usuário possa executar outro programa, deverá aguardar a finalização do programa corrente. Esta era uma característica dos primeiros sistemas operacionais que estavam relacionados ao surgimento dos primeiros computadores na década de 60.

Caracterizavam-se por permitir que todos os recursos de *hardware* ficassem exclusivamente dedicados a um único programa. Em consequência, sua principal desvantagem residia no fato de que enquanto um programa aguardava por um evento externo, como a digitação de um caractere do teclado, o processador permanecia ocioso. Além disso, tanto a memória principal quanto os recursos de E/S (Entrada e Saída) como impressoras e discos eram subutilizados, uma vez que todos estariam dedicados a um único programa como mostra a Figura 2.3.

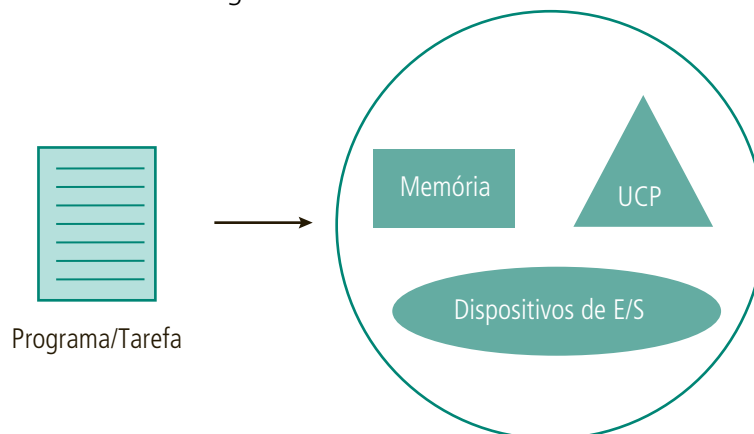


Figura 2.3: Sistema monoprogramável ou monotarefa

Fonte: Adaptado de Machado, 2004

Sistemas Multiprogramáveis ou Multitarefa: os recursos computacionais são compartilhados entre os diversos usuários e suas aplicações. Podemos observar o compartilhamento de memória e do processador. Nesse caso, o compartilhamento de tempo no processador é distribuído. Assim, o usuário tem a impressão que vários processos estão sendo executados simultaneamente. Um dos processos ocupa o processador enquanto os outros ficam enfileirados, aguardando a sua vez de entrar em execução. Cabe ao sistema operacional o papel de gerenciar de forma ordenada e protegida o acesso concorrente aos recursos disponíveis.

Sistemas multiprogramáveis ou multitarefa permitem o compartilhamento dos recursos computacionais entre diversos usuários e aplicações, permitindo sua execução concorrente.

A vantagem desse tipo de sistema é uma melhor utilização dos recursos disponíveis, o que resulta em menor tempo de resposta das aplicações. Além de um custo reduzido, uma vez que haverá o compartilhamento dos recursos entre as diferentes aplicações e aumento da produção do usuário.

Graças aos sistemas multiprogramáveis é possível editar um documento no *MS Word*, navegar na internet, ouvir música, utilizar os mensageiros instantâneos (MSN), tudo ao mesmo tempo!

Sistemas com múltiplos processadores: o sistema operacional distribui as tarefas entre dois ou mais processadores. A vantagem desse tipo de sistema é permitir que mais de um programa possa ser executado simultaneamente ou que um mesmo programa seja dividido em várias partes e executado simultaneamente nos vários processadores, aumentando o desempenho.

Esse tipo de sistema surgiu da necessidade de aplicações que requeriam um grande poder computacional, como sistemas de previsão do tempo, modelagens, simulações, desenvolvimento aeroespacial, entre outros. Com múltiplos processadores, é possível reduzir drasticamente o tempo de processamento destas aplicações. Inicialmente, as configurações limitavam-se a poucos processadores, mas, atualmente existem sistemas com milhares de processadores.



Sistemas com múltiplos processadores

Sistemas com múltiplos processadores caracterizam-se por possuir duas ou mais CPU's interligadas e trabalhando de forma conjunta na solução de um problema.



Os sistemas com múltiplos processadores podem ser classificados em fortemente acoplados e fracamente acoplados, em função da comunicação entre CPU's e o grau de compartilhamento da memória. Em sistemas fortemente acoplados, há uma única memória principal compartilhada por todos os processadores, enquanto em sistemas fracamente acoplados cada sistema tem sua própria memória. Com isso, a taxa de transferência entre processadores em sistemas fortemente acoplados é bem maior que em sistemas fracamente acoplados.