**Introdução da aula**



**Qual é o foco da aula?**

Nesta aula, vamos explicar a estrutura interna dos sistemas operacionais, conheceremos os tipos e aprenderemos sobre os sistemas operacionais monoprogramáveis, multiprogramáveis e com múltiplos processadores.

**Objetivos gerais de aprendizagem**

Ao longo desta aula, você irá:

* explicar o principal componente dos sistemas operacionais, responsável pelo gerenciamento dos recursos do computador;
* identificar os modelos das principais arquiteturas dos sistemas operacionais: monolíticos, em camadas, máquinas virtuais e modelo cliente-servidor;
* classificar os sistemas operacionais: monoprogramáveis/monotarefa, multiprogramáveis/multitarefa, sistemas com múltiplos processadores.

**Situação-problema**

Você já pensou sobre como os sistemas operacionais conseguem gerenciar diversos programas e aplicativos do usuário e do sistema sem comprometer a performance do computador?

Quando um computador está ligado, além dos seus aplicativos que estão sendo executados (como editor de textos, internet, aplicativos de música, entre outros), outros programas estão rodando em paralelo, como o antivírus (rastreando as possíveis ameaças que poderiam danificar o sistema operacional), atualização de versões de aplicativos instalados, etc. Esse gerenciamento eficaz somente é possível pelo fato de os sistemas operacionais atuais serem multitarefa.

Vamos relembrar da nossa situação-problema sobre o curso de sistemas operacionais ministrado por você durante o simpósio “Informática para todos” entre as escolas municipais da região. Durante o curso, outro artigo foi disponibilizado aos estudantes, que trata da classificação dos sistemas operacionais quanto à sua capacidade de controlar mais de uma tarefa ao mesmo tempo (monotarefa, multitarefa e multiprocessamento) e sua performance durante este controle.

Desta forma, você é questionado:

* quais as principais estruturas dos sistemas operacionais?
* como os sistemas operacionais da atualidade, como Windows e Linux, conseguem realizar diversas tarefas ao mesmo tempo sem prejudicar o desempenho do computador?
* quais as principais características e aplicações dos sistemas operacionais multiprogramáveis/multitarefa?
* qual a principal diferença entre os sistemas multiprogramáveis?
* para os sistemas operacionais multiusuários, o Linux e Windows oferecem suporte?

**Núcleo do sistema operacional**



O sistema operacional é composto por um conjunto de rotinas chamado kernel ou núcleo do sistema operacional. O *kernel* é o principal componente dos sistemas operacionais, pois ele é quem faz o gerenciamento dos recursos do computador (MACHADO; MAIA, 2007).

Quando o computador é ligado, o *kernel* executa programas que inicializam o computador e o conjunto de rotinas é executado seguindo uma ordem lógica.

As principais funções do núcleo são (SIQUEIRA, 2018):

* tratamento de interrupções e exceções.
* gerenciamento, sincronização,  comunicação    e escalonamento de processos e *threads*.
* gerenciamento da memória.
* gerenciamento dos sistemas de arquivos.
* gerenciamento dos dispositivos de entrada/saída.
* auditoria e segurança do sistema.

Os sistemas operacionais limitam as ações executadas pelos programas em função da segurança e da estabilidade, como o acesso à memória do computador. Assim, os modos de acesso aos serviços do núcleo do computador 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: usuário e *kernel* (núcleo). No modo usuário, os programas podem executar ações sem privilégios, como leitura de um arquivo. No modo *kernel*, o acesso pode ser realizado em modo usuário ou em modo com privilégio total no computador, como acesso ao disco.

Quando um programa é executado, o *kernel* é consultado para saber se o acesso será realizado em modo usuário ou *kernel*. Os acessos aos serviços do núcleo são realizados através de chamadas ao sistema ou *system call*. As chamadas ao sistema fazem a interface entre o sistema operacional e os programas dos usuários, além de poderem solicitar serviços de acesso ao disco rígido ou para a criação e execução de processos, por exemplo.

Durante uma chamada ao sistema, o sistema operacional recebe um comando (por exemplo, abrir um arquivo) e seus parâmetros e, como resposta à chamada, recebe um outro código sinalizando sucesso (se a abertura do arquivo aconteceu com sucesso), falha (se houve erro na abertura do arquivo) ou o resultado do próprio comando (o arquivo é aberto para o usuário).

\_\_\_\_\_\_

**💭 Reflita**

Os usuários de computador conseguem identificar claramente quais são as funções do núcleo, diferenciando-as das funções do *software* aplicativo (editores de texto, Internet Explorer, entre outros)? Como pode ser feita essa diferenciação?

\_\_\_\_\_\_

A estrutura do sistema operacional, ou seja, a maneira como o código do sistema é organizado e o inter-relacionamento entre seus diversos componentes podem variar conforme a concepção do projeto. Veremos nesta aula os modelos das principais arquiteturas dos sistemas operacionais, os quais, segundo Tanenbaum (2003), são:

* sistemas Monolíticos.
* sistemas em Camadas.
* máquinas Virtuais.
* modelo Cliente-Servidor.

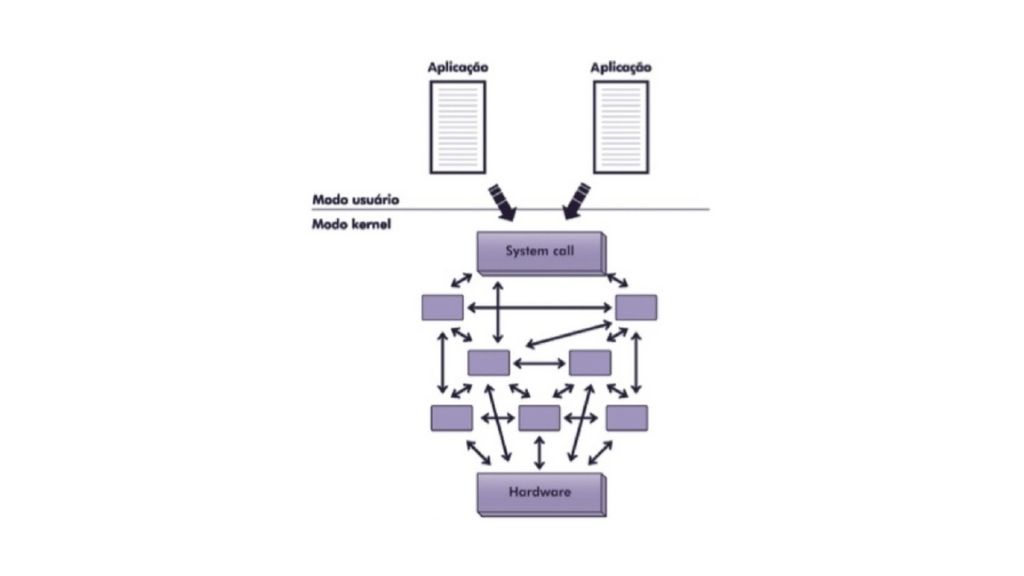
**Modelos das principais arquiteturas dos sistemas operacionais**



**Sistemas Monolíticos**

No sistema monolítico, o sistema operacional é escrito como um conjunto de módulos compilados separadamente e depois agrupados em um arquivo executável, em que cada procedimento pode ser chamado por outro quando precisar, conforme apresentado na figura.

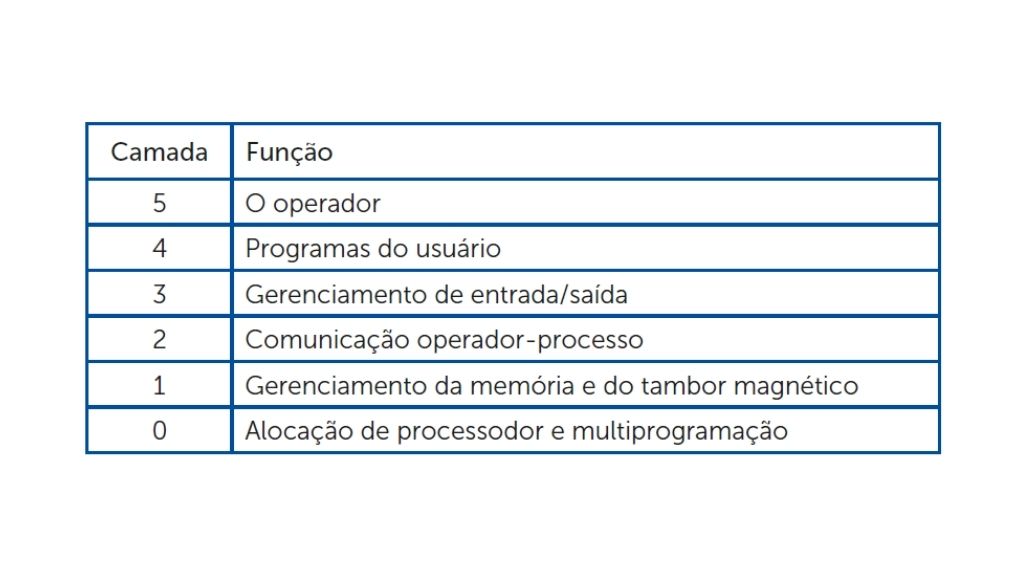
A vantagem do sistema monolítico refere-se à simplicidade e à eficiência em função da integração interna dos componentes, permitindo que os detalhes de *hardware* sejam explorados. Um exemplo de sistema monolítico é o sistema operacional MS-DOS.

Modelo de um sistema monolítico. Fonte: Machado; Maia (2018, p. 54).

**Sistemas em camadas**

O sistema em camadas organiza o sistema operacional como uma hierarquia de camadas construídas umas sobre as outras. Cada camada oferece um conjunto de funções que podem ser utilizadas pelas camadas superiores (TANENBAUM, 2003).

A figura apresenta um modelo de sistema de camadas.

Modelo do sistema em camadas. Fonte: Tanenbaum (2003, p. 43).

A camada 0 faz a alocação do processador alternando entre os processos e realizando a multiprogramação.

A camada 1 gerencia a memória, alocando espaço na memória principal e em um tambor (meio magnético de armazenamento de dados).

A camada 2 gerencia a comunicação entre o processo e o console (componentes de entrada (teclado) e saída (monitor) de dados) do operador.

A camada 3 gerencia os dispositivos de entrada/saída.

Na camada 4 estão os programas de usuários e na camada 5 está o processo do operador do sistema.

**Máquinas Virtuais**

Uma máquina virtual (VM) é similar a uma máquina real. Assim, um nível intermediário é criado entre o sistema operacional e o *hardware*. Esse nível intermediário cria várias máquinas virtuais independentes, cada qual possuindo uma cópia virtual do *hardware*. Em um mesmo *hardware* e em plataformas diferentes podem rodar mais de uma máquina virtual (MACHADO; MAIA, 2007). As principais vantagens das máquinas virtuais são o aproveitamento dos recursos das máquinas, a garantia da segurança e confiabilidade, pois se acontecer um problema com uma máquina virtual, as demais não serão impactadas, além de facilitar a recuperação de dados e *backups*.

\_\_\_\_\_\_

**➕ Pesquise mais**

Você sabia que um sistema operacional pode “rodar” dentro de outro? Para saber mais sobre máquina virtual acesse o artigo [**O que são máquinas virtuais?**](https://www.tecmundo.com.br/maquina-virtual/232-o-que-sao-maquinas-virtuais-.htm)

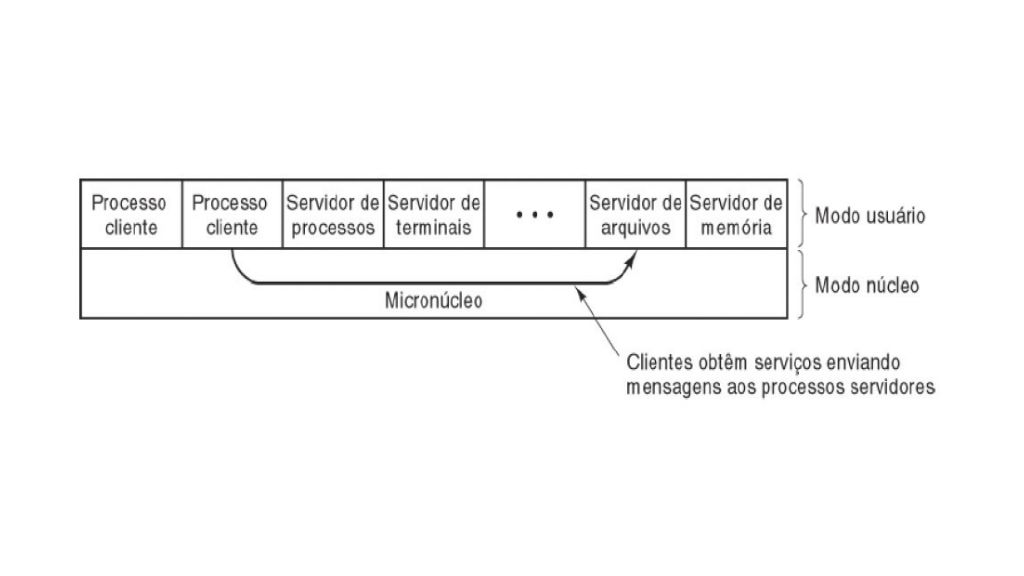
\_\_\_\_\_\_

**Modelo cliente-servidor**

No modelo cliente-servidor a ideia é implementar a maior parte das funções em modo usuário.

Tudo o que o *kernel* faz é cuidar da comunicação entre cliente e servidor, conforme apresentado na figura “Modelo do sistema em camadas”, que também apresenta a divisão do sistema operacional em várias partes, como servidor de processos, de terminais, de arquivos e de memória, tornando cada parte gerenciável.

Todos os servidores executam em modo usuário e, com isso, caso ocorra um erro no servidor de arquivos, não impactará nos demais servidores (TANENBAUM, 2003).

Modelo do sistema em camadas. Fonte: Tanenbaum (2003, p. 46).

**Classificação dos sistemas operacionais**



Segundo Machado e Maia (2007), os sistemas operacionais podem ser classificados em:

1. monoprogramáveis/monotarefa.
2. multiprogramáveis/multitarefa.
3. sistemas com múltiplos processadores.

**1. Sistemas Monoprogramáveis/monotarefa**

Segundo Machado e Maia (2007), os sistemas operacionais monoprogramáveis/monotarefa referem-se aos primeiros computadores pessoais criados em 1960. Os sistemas monoprogramáveis/monotarefa executam um único programa por vez e todos os recursos da máquina são alocados exclusivamente para o programa em execução, mesmo que não esteja utilizando o recurso.

Por exemplo, você deseja atualizar uma planilha eletrônica e editar um texto, ou você atualiza a planilha eletrônica e fecha o editor de textos ou edita o texto e fecha a planilha eletrônica. A implementação de sistemas monoprogramáveis/monotarefa é simples porque não é necessário se preocupar com a concorrência de recurso. Um exemplo de monoprogramáveis/monotarefa foi o MS-DOS.

**2. Sistemas Multiprogramáveis/multitarefa**

Os sistemas operacionais multiprogramáveis/multitarefa dividem os recursos do computador com os demais programas e, com isso, a CPU não fica ociosa como no caso de sistemas operacionais monoprogramáveis/monotarefa (MACHADO; MAIA, 2007). Por exemplo, você pode atualizar uma planilha eletrônica e utilizar

um editor texto ao mesmo tempo. Os sistemas multiprogramáveis podem ser classificados pela maneira com que as aplicações são gerenciadas e pelo número de usuários que utilizam o sistema. Os sistemas multiprogramáveis em relação ao número de usuários que os utilizam são:

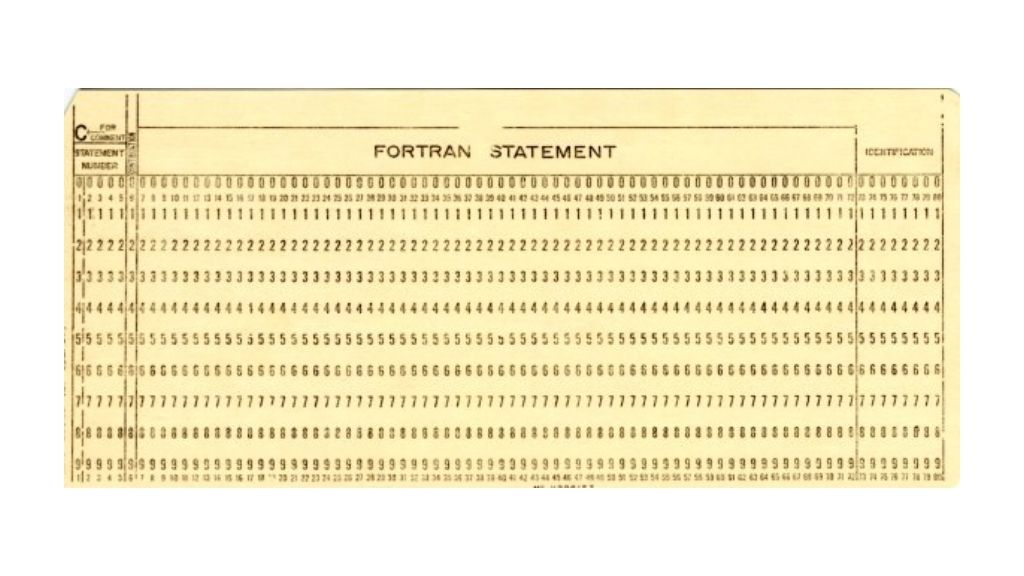
* **monousuários**: somente um usuário utilizando os recursos da máquina.
* **multiusuários**: vários usuários logados utilizando os recursos da máquina.

Os sistemas multiprogramáveis, em relação à maneira com que as aplicações são gerenciadas, podem ser classificados como *batch*, de tempo compartilhado ou de tempo real.

**Sistemas *batch***

Os sistemas *batch* foram os primeiros sistemas operacionais multiprogramáveis e o processamento era realizado por meio de grupos de registros (lote). Os programas, conhecidos como *jobs*, eram executados através de cartões perfurados, conforme mostra a figura “Exemplo de cartão perfurado”, e armazenados em disco ou fitas, onde aguardavam para serem processados.

Nos sistemas *batch* não havia interação do usuário com a aplicação. *Backups*, cálculos numéricos e compilações são exemplos de processamento *batch*.

Exemplo de cartão perfurado. Fonte: Wikimedia Commons.

**Sistemas de tempo compartilhado (*time-sharing*)**

Os sistemas de tempo compartilhado (*time-sharing*) permitem que vários programas sejam executados a partir da divisão do tempo do processador em pequenos intervalos, conhecidos como fatia de tempo (*time-slice*) (MACHADO; MAIA, 2007).

No sistema de tempo compartilhado, cada usuário possui um ambiente de trabalho próprio, dando a impressão de que o sistema está totalmente dedicado a ele. Se um tempo destinado à execução de um programa não for suficiente, o programa é interrompido pelo sistema operacional, que passa a vez de execução para outro programa, e o que foi interrompido fica aguardando uma nova fatia de tempo para executar.

Nos sistemas de tempo compartilhado, os comandos dados pelos usuários aos sistemas operacionais são respondidos em segundos e, por isso, ficaram conhecidos como sistemas on-line. Muitas aplicações comerciais utilizam este sistema por ter um tempo de resposta razoável.

**Sistemas de tempo real (*real-time*)**

Nos sistemas de tempo real, o tempo é o principal parâmetro. Um programa utiliza os recursos do computador até aparecer outro com maior prioridade. Os prazos são rigorosos quanto à execução de uma tarefa (MACHADO; MAIA, 2007) e aplicados em processos de controles industriais, como monitoramento de refinarias de petróleo, controle de tráfego aéreo, de usinas termoelétricas e nucleares ou em qualquer aplicação onde o tempo de respostas é fator fundamental.

**3. Sistemas com múltiplos processadores**

Segundo Machado e Maia (2007), os sistemas com múltiplos processadores possuem duas ou mais CPUs interligadas e permitem que vários programas sejam executados ao mesmo tempo. Os sistemas com múltiplos processadores permitem que um programa seja subdividido e executado ao mesmo tempo em mais de um processador. Com isso, além de esses sistemas serem rápidos, possuem uma maior capacidade de processamento e controle da CPU.

Os sistemas com múltiplos processadores são complexos, gastam uma parte do tempo gerenciando a CPU e também estão presentes nas aplicações de simulações e prospecção de petróleo.

Os sistemas com múltiplos processadores podem ser classificados em:

* **Sistemas fortemente acoplados**

Possuem dois ou mais processadores, compartilham a mesma memória e os dispositivos de entrada/saída são controlados por um único sistema operacional. São utilizados em sistemas que usam intensivamente a CPU, voltando-se à solução de apenas um problema.

Os sistemas fortemente acoplados são classificados em:

* **Simétricos**

Nos sistemas simétricos os processadores compartilham uma única memória e utilizam o mesmo sistema operacional. Usam a técnica de paralelismo em que os programas podem ser divididos em partes, podendo rodar concorrentemente em processadores que estejam disponíveis. Com o processamento paralelo, há um aumento do processamento e caso haja uma falha em um dos processadores, não impacta na performance do sistema.

O Windows e o Linux suportam os sistemas simétricos.

* **Assimétricos**

Nos sistemas assimétricos existe um processador principal (mestre) que controla o sistema e delega atividades para os demais processadores. Se acontecer algum problema com o processador principal, os demais não continuam o processamento.

* **Sistemas fracamente acoplados**

Os sistemas fracamente acoplados funcionam de forma independente, possuindo seu sistema operacional e gerenciando seus próprios recursos como memória, CPU e dispositivos de entrada e saída.

O sistema é fracamente acoplado por depender de uma rede para distribuir as atividades do processamento (MACHADO; MAIA, 2007).

\_\_\_\_\_\_

**🔁 Assimile**

Uma das diferenças entre os sistemas operacionais fortemente acoplados e os fracamente acoplados é que os sistemas fortemente acoplados possui apenas uma memória compartilhada pelos processadores, já os fracamente acoplados possuem a sua memória individualmente.

\_\_\_\_\_\_

Os sistemas fortemente acoplados são classificados em:

* **operacionais de rede**

Os sistemas operacionais de rede são independentes e estão conectados por meio de uma rede. Na rede, cada estação de trabalho ou nó possui um sistema operacional próprio e tem a capacidade de processamento das suas aplicações (MACHADO; MAIA, 2007).

Eles também permitem o compartilhamento de recursos como impressora, diretórios, cópia de arquivos, entre outros.

* **operacionais distribuídos**

Os sistemas operacionais distribuídos permitem que um programa seja dividido em partes e que cada parte seja executada em nós diferentes da rede. Os usuários do sistema veem como um sistema centralizado e não como um sistema em rede (MACHADO; MAIA, 2007).

\_\_\_\_\_\_

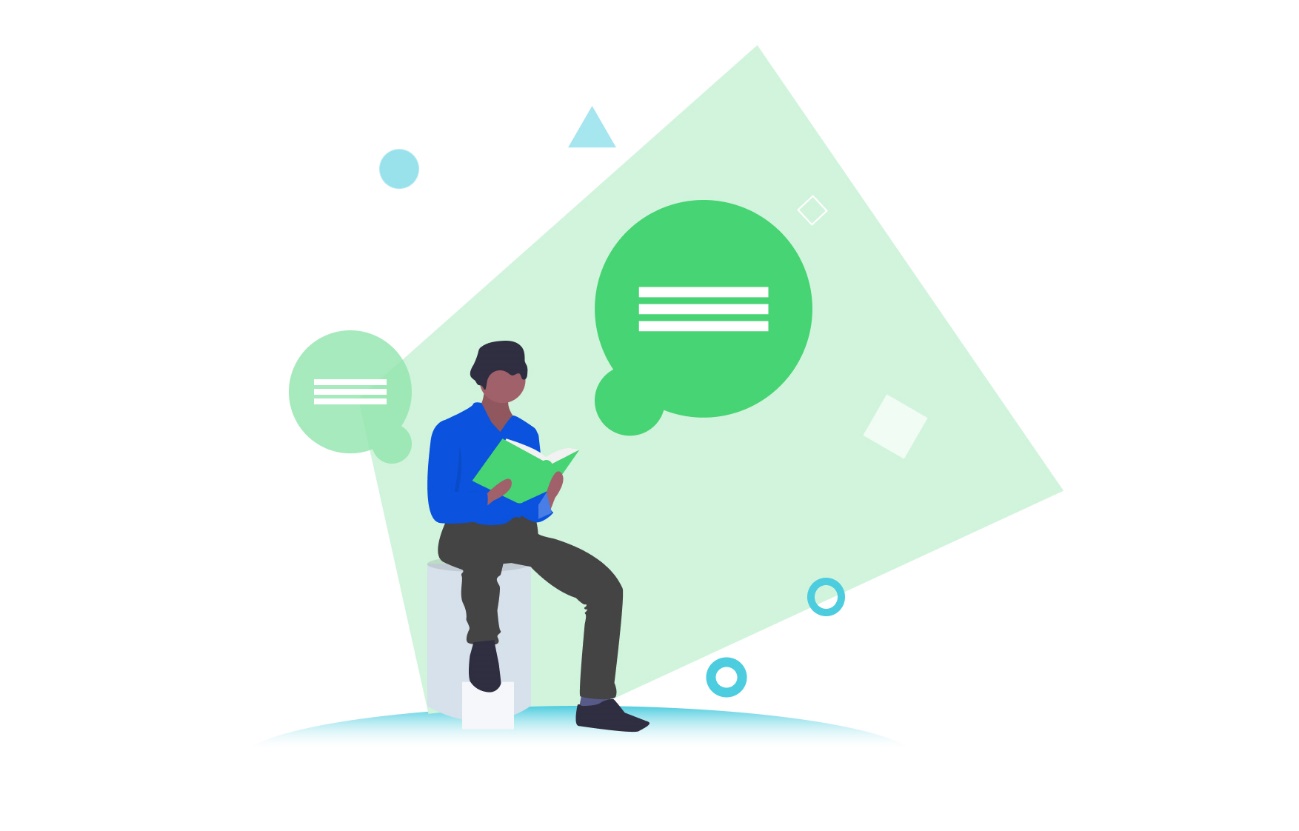
**📝 Exemplificando**

Um exemplo de sistemas distribuídos são os *clusters*. Um *cluster* é um conjunto de computadores conectados como se fossem um grande sistema. Esse conjunto consegue processar aplicações como se fossem supercomputadores. Cada computador que faz parte do cluster recebe um nome de um nó, e o usuário não conhece os nomes dos membros do cluster e nem quantos são. *Clusters* são muito utilizados em servidores de bancos de dados e Web.

Por exemplo, o Google contém vários *datacenters* (compostos por várias máquinas em cluster) espalhados pelo mundo e quando uma solicitação de pesquisa é feita na web, a informação é procurada no *datacenter* mais próximo, para que tenha um tempo de resposta mais rápido.

Para aprofundar seus conhecimentos sobre o *cluster*, indicamos o artigo [**Cluster: conceito e características**.](https://www.infowester.com/cluster.php)

**Conclusão**



Agora que você já conheceu a estrutura dos sistemas operacionais e estudou sobre seus tipos, monoprogramáveis e multiprogramáveis e com múltiplos processadores, vamos voltar ao contexto sobre o curso de sistemas operacionais ministrado por você durante o simpósio “Informática para todos” entre as escolas municipais da região.

Um dos artigos disponibilizados por você aos estudantes refere-se à capacidade dos sistemas operacionais de controlar mais de uma tarefa ao mesmo tempo e sua performance durante este controle.

Os questionamentos dos estudantes direcionados a você foram:

* quais as principais estruturas dos sistemas operacionais?
* como os sistemas operacionais da atualidade, como o Windows e Linux, conseguem realizar diversas tarefas ao mesmo tempo sem prejudicar o desempenho do computador?
* quais as principais características e aplicações dos sistemas operacionais multiprogramáveis/multitarefa?
* qual a principal diferença entre os sistemas multiprogramáveis?
* para os sistemas operacionais multiusuários, o Linux e Windows oferecem suporte?

Os sistemas operacionais possuem a estrutura de acordo com o projeto e as principais arquiteturas são: os sistemas monolíticos, caracterizados pela simplicidade de implementação, os sistemas em camadas, em que o sistema operacional é organizado em hierarquia de camadas, as máquinas virtuais, que simulam as máquinas reais e são instaladas dentro de outro sistema operacional, e o modelo cliente-servidor, que executa as funções em modo usuário.

Os sistemas operacionais da atualidade, como Windows e Linux, permitem a execução de vários programas ao mesmo tempo e gerenciam, de forma eficaz, todos os recursos da máquina por serem multitarefa e multiusuário. Os sistemas operacionais multiprogramáveis/multitarefa gerenciam o tempo de processamento de um programa de acordo com as prioridades e o tempo que este programa já executou.

Assim, todos os programas possuem uma fatia de tempo para uso dos recursos da máquina (processador, memória e dispositivos de entrada/saída). Uma das principais características dos sistemas multiprogramáveis/multitarefa é a redução do tempo de resposta dos programas, além da redução de custos no compartilhamento de recursos do computador entre os demais programas.

Os sistemas multiprogramáveis podem ser do tipo *batch*, tempo compartilhado e sistemas em tempo real. Os sistemas *batch* executam por meio de lotes de tarefas e o sistema operacional transfere a execução para a próxima tarefa após a finalização completa da que está em execução. Um exemplo de uso do sistema *batch* é o processamento de uma conta telefônica.

Os sistemas de tempo compartilhado dividem o processamento com vários usuários do sistema e muitas aplicações comerciais utilizam este sistema. Já nos sistemas em tempo real, o tempo é o fator principal, pois o tempo de resposta precisa ser rápido, como, por exemplo, a esteira de produção de um carro.

O suporte aos usuários Linux é realizado por meio de grupos e fóruns na internet ou assistência especializada (serviço pago). O suporte do Windows é realizado através de uma central de atendimento e o usuário precisa pagar por este suporte.