

# SISTEMAS OPERACIONAIS



**Professor Fábio Angelo**  
**E-mail: [fabio.angelo@unisul.br](mailto:fabio.angelo@unisul.br)**

# O QUE ROLOU NA ÚLTIMA AULA?

- Apresentado o conceito da Concorrência nos Sistemas Operacionais;
- Conversado sobre Interrupção e Exceção, bem como a diferença entre estes tipos de eventos;
- Demonstrado algumas estratégias para otimizar as operações de entrada e saída, como o Buffering e Spooling;
- Fechamento os conceitos falando sobre Reentrância.

RESERVADO UM MOMENTO PARA OS GRUPOS CONVERSAREM SOBRE O TRABALHO 1

# CONVERSANDO SOBRE AS RESPOSTAS RECEBIDAS... (GRUPO GABI)

1. O que é concorrência e como este conceito está presente nos sistemas operacionais?

R: Capacidade do processador de executar mais de uma instrução ao mesmo tempo, este conceito é visto nos sistemas multiprogramáveis.

2. Por que o mecanismo de interrupção é fundamental para a implementação da multiprogramação?

R: Utilizada para sincronizar a execução de todas as suas rotinas e dos programas dos usuários, além de controlar dispositivos.

# CONVERSANDO SOBRE AS RESPOSTAS RECEBIDAS... (GRUPO GABI)

3. Descreva os passos que são executados quando um programa trata uma interrupção ou exceção.

R: O programa salva o conteúdo dos registradores na pilha de controle, identifica a origem do evento, obtém o endereço da rotina de tratamento, após execução da rotina de tratamento é realizada a restauração do conteúdo dos registradores.

4. O que são eventos síncronos e assíncronos? Como estes eventos estão relacionados ao mecanismo de interrupção e exceção?

R: Eventos síncronos são resultado da execução de um programa corrente, sendo estes eventos previsíveis e que podem ocorrer um de cada vez, neste caso tratam-se de exceções. Já os eventos assíncronos podem ocorrer múltiplas vezes e são imprevisíveis, neste caso tratam-se das interrupções.

# CONVERSANDO SOBRE AS RESPOSTAS RECEBIDAS... (GRUPO GABI)

5. O que é DMA e qual a vantagem desta técnica?

R: DMA (Direct Access Memory) é uma técnica usada para liberar o processador quando arquivos grandes são acessados, geralmente é usada para otimização da operação de entradas e saídas.

6. Como a técnica de buffering permite aumentar a concorrência em um sistema?

R: Já começar a usar o dado mesmo que não tivesse acabado, assim que o processador pede o dado é criado um carregamento intermediário que já vai sendo consumido. Muito usado para gravação de videos em FTP.

# CONVERSANDO SOBRE AS RESPOSTAS RECEBIDAS... (GRUPO GABI)

7. Explique o mecanismo de spooling de impressão.

R: São enviados os arquivos para impressão para uma área intermediária, liberando o programa para outras atividades, junto com todos os outros arquivos enviados por outras máquinas criando uma fila em que o primeiro arquivo enviado será impresso primeiro e assim por diante.

8. Em um sistema multiprogramável, seus usuários utilizam o mesmo editor de textos (2 Mb), software de correio eletrônico (1.5 Mb) e uma aplicação corporativa (900 Kb). Caso o sistema não implemente reentrância, qual o espaço de memória principal ocupado pelos programas quando 10 usuários estiverem utilizando todas as aplicações simultaneamente? Qual o espaço liberado quando o sistema implementa reentrância em todas as aplicações?

R: Sem usar a reentrância:  $(2\text{Mb} + 1.5\text{Mb} + 0,9\text{Mb}) * 10 = \mathbf{44\text{Mb}}$  / Se usar a reentrância:  $(2\text{Mb} + 1.5\text{Mb} + 0,9\text{Mb}) = \mathbf{4,4\text{Mb}}$

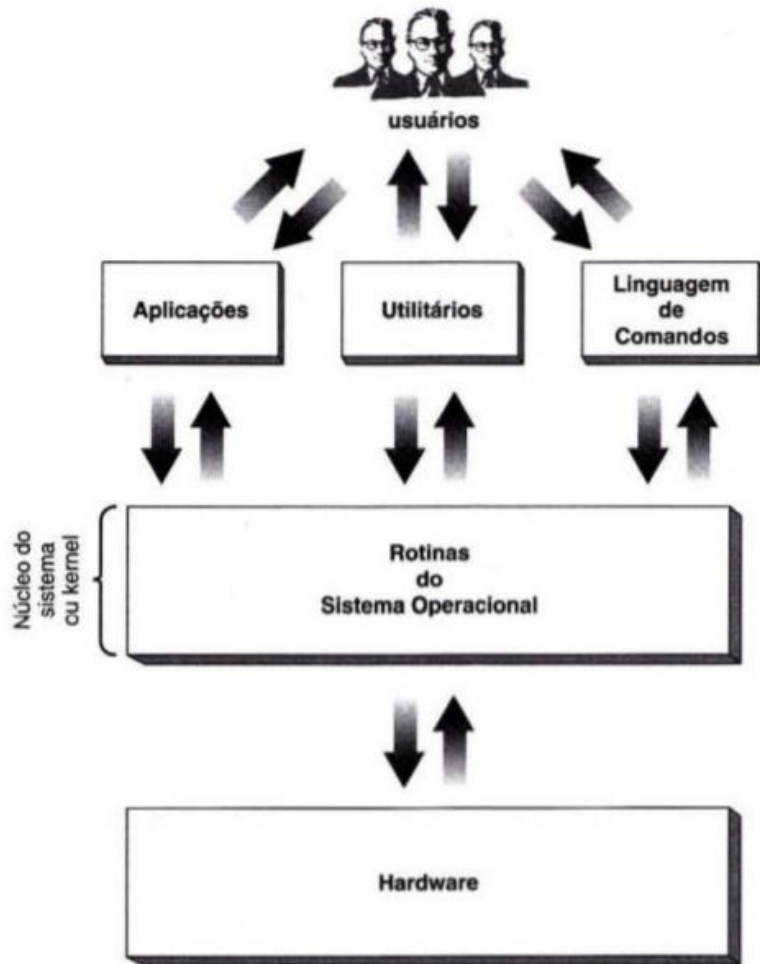
Espaço liberado ao usar reentrância: **39,6Mb**

# ESTRUTURA DO SISTEMA OPERACIONAL

- O sistema operacional é formado por um conjunto de rotinas que oferece serviços aos usuários e às suas aplicações;
- Esse conjunto de rotinas é denominado núcleo do sistema ou kernel;
- Entretanto, nem todo o conjunto de rotinas e aplicativos fazem parte do núcleo do sistemas operacional, mas servem de ponte ao kernel.

# ESTRUTURA DO SISTEMA OPERACIONAL

- **A Comunicação com o Sistema Operacional pode ocorrer de 3 maneiras:**
  - por intermédio das chamadas de rotinas do sistema realizadas por aplicações;
  - acionando os utilitários do próprio sistema operacional;
  - Usando a linguagem de comandos nativa do sistema operacional (compiladores);
- **Serão apresentadas as funções do núcleo e os conceitos relativos à segurança, proteção do sistema, modos de acesso, rotinas do sistema, system calls, linguagem de comandos e ativação/desativação do sistema**

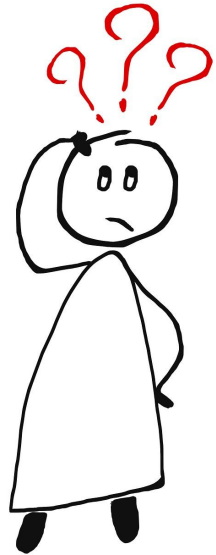




# UMA DÚVIDA...

Quando acionamos o gerenciador de tarefas do Sistema Operacional, de que maneira estamos interagindo com ele?

- a) Aplicação
- b) Utilitário
- c) Linguagem de Comando



# FUNÇÕES DO KERNEL

- A compreensão da estrutura e do funcionamento de um sistema operacional não é simples, pois as rotinas do sistema são executadas concorrentemente sem uma ordem pré definida, com base em eventos dissociados do tempo (eventos assíncronos);
- Muitos desses eventos estão relacionados ao hardware e a tarefas internas do próprio sistema operacional;
- Trabalharemos um conjunto dessas funcionalidades para compreender o modus operandi dos Sistemas Operacionais.

# PRINCIPAIS FUNÇÕES DO KERNEL

- Tratamento de interrupções e exceções;
- Criação e eliminação de processos e threads;
- Sincronização e comunicação entre processos e threads;
- Escalonamento e controle dos processos e threads;
- Gerência de memória;
- Gerência do sistema de arquivos;
- Gerência de dispositivos de E/S;
- Suporte a redes locais e distribuídas;
- Contabilização do uso do sistema;
- Auditoria e segurança do sistema.



**Questão boa  
para uma  
prova!**

# DESAFIOS DO AMBIENTE MULTIPROGRAMÁVEL

- Considerar a situação em que diversos usuários compartilham os mesmos recursos, como memória, processador e dispositivos E/S;
- Garantir a confiabilidade na execução concorrente de todos os programas e nos dados dos usuários, além da garantia da integridade do próprio sistema operacional;
- O sistema operacional deve ser responsável pelo controle da utilização da UCP, de forma a impedir que algum programa monopolize o seu uso inadequadamente;
- Devido a ocupação simultânea da memória, cada usuário deve possuir uma área reservada onde seus dados e código são armazenados.

# DESAFIOS DO AMBIENTE MULTIPROGRAMÁVEL

- Para que diferentes programas tenham o direito de compartilhar uma mesma área de memória, o sistema operacional deve oferecer mecanismos para que a comunicação seja feita de forma sincronizada e controlada, evitando, desta forma, problemas de inconsistência;
- O sistema operacional deve garantir a integridade e a confidencialidade dos dados no disco (memória secundária), permitindo ainda que dois ou mais usuários possam ter acesso simultâneo ao mesmo arquivo;
- Em resumo, no ambiente multiprogramável, o sistema operacional deve implementar mecanismos de proteção que controlem o acesso concorrente aos diversos recursos computacionais.

# MODO DE ACESSO

- Recurso utilizado nos Sistemas Operacionais e também nos processadores para garantir a integridade de todo o ambiente;
- Os processadores em geral possuem dois modos de acesso:
  - Modo Usuário
  - Modo Kernel;
- Quando o processador trabalha no modo usuário, uma aplicação só pode executar instruções conhecidas como não-privilegiadas, tendo acesso a um número reduzido de instruções.

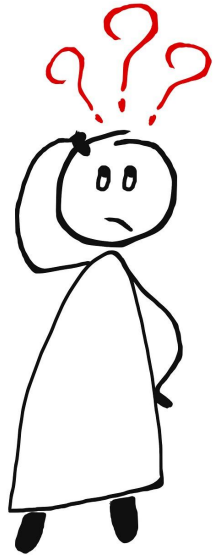
# MODOS DE ACESSO

- No modo kernel a aplicação pode ter acesso ao conjunto total de instruções do processador;
- O modo de acesso é determinado por um conjunto de bits, localizado no registrador de status do processador, que indica o modo de acesso corrente;
- As instruções privilegiadas não devem ser utilizadas de maneira indiscriminada pelas aplicações, pois isso poderia ocasionar sérios problemas à integridade do sistema.

# UMA DÚVIDA...

O aplicativo DEFRAG do  
Windows opera em qual modo?

- a) Usuário
- b) Kernel



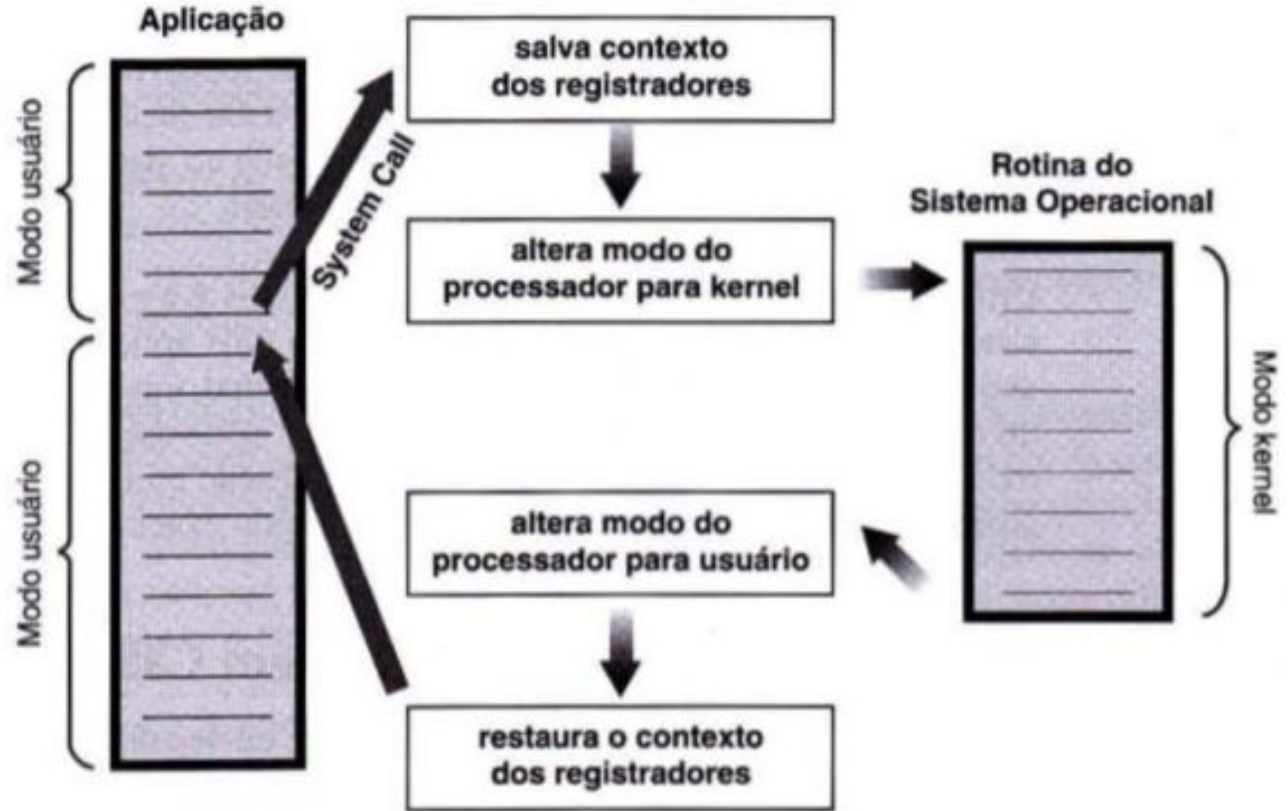


# SYSTEM CALLS

- As rotinas de Kernel executam um conjunto de instruções privilegiadas e partindo deste princípio, o processador deve estar obrigatoriamente em modo Kernel;
- Essa premissa exige a implementação de mecanismos de proteção para garantir a confiabilidade do sistema, chamada de System Calls;
- Toda vez que uma aplicação deseja chamar uma rotina do sistema operacional, o mecanismo de system call é ativado, com objetivo de aferir se o acionador tem permissão para a chamada.

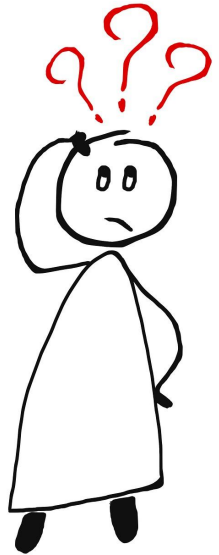
# SYSTEM CALLS

O fluxo ao lado mostra com uma aplicação com privilégio de execução é tratada pelo Sistema Operacional



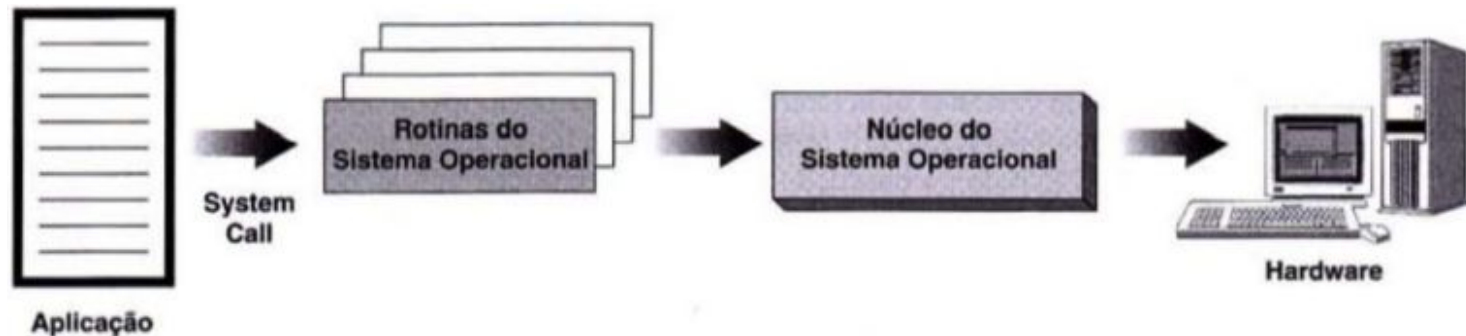
# UMA DÚVIDA...

Quando você salva um arquivo do Word no disco, podemos dizer que existe a troca do modo de acesso de usuário para kernel?



# ROTINAS DO S.O. E AS SYSTEM CALLS

- Porta de entrada entre os aplicativos e o Kernel;
- Por intermédio dos parâmetros fornecidos na system call, a solicitação é processada e uma resposta é retornada à aplicação juntamente com um estado de conclusão indicando se houve algum erro.



# SYSTEM CALLS

- O nome System Calls é tipicamente usado em ambiente Unix;
- Para o Windows, o termo é tratado como API (Application Program Interface);
- No exemplo abaixo, escrito em Delphi, uma chamada ao Sistema é executada para obtenção da hora.

```
GetSystemTime(SystemTime);  
DataHoraT := SystemTimeToDateTime(SystemTime);  
DataHoraS := DateTimeToStr(DataHoraT);  
RichEdit1.Lines.Add(DataHoraS);
```

# SYSTEM CALLS

- As linguagens de alto nível abstraem os detalhes envolvidos nas chamadas ao Sistema Operacional;
- De forma simplificada, o comando da linguagem de alto nível é convertido pelo compilador para uma chamada de rotina específica que, quando executada, verifica a ocorrência de erros e retorna os dados ao programa de forma transparente ao usuário.

Funções	System calls
Gerência de processos e threads	Criação e eliminação de processos e threads Alteração das características de processos e threads Sincronização e comunicação entre processos e threads Obtenção de informações sobre processos e threads
Gerência de memória	Alocação e desalocação de memória
Gerência do sistema de arquivos	Criação e eliminação de arquivos e diretórios Alteração das características de arquivos e diretórios Abrir e fechar arquivos Leitura e gravação em arquivos Obtenção de informações sobre arquivos e diretórios
Gerência de dispositivos	Alocação e desalocação de dispositivos Operações de entrada/saída em dispositivos Obtenção de informações sobre dispositivos

# LINGUAGENS DE COMANDO

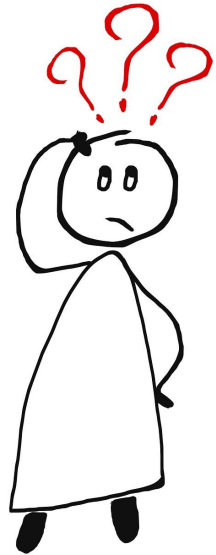
- Permite que o usuário se comunique de uma forma simples com o sistema operacional, capacitando-o a executar diversas tarefas específicas como criar, ler ou eliminar arquivos, listar diretórios ou verificar a data e hora;
- Cada sistema operacional possui a sua linguagem de comandos.

Cada comando, depois de digitado pelo usuário, é interpretado pelo shell ou interpretador de comandos, que verifica a sintaxe do comando, faz chamadas a rotinas do sistema e apresenta um resultado ou uma mensagem informativa

Comando	Descrição
dir	Lista o conteúdo de um diretório
cd	Altera o diretório default
type	Exibe o conteúdo de um arquivo
del	Elimina arquivos
mkdir	Cria um diretório
ver	Mostra a versão do Windows

# UMA DÚVIDA...

Esta mesma filosofia de linguagens de comando permanece ativa com as interfaces gráficas mais atualizadas?

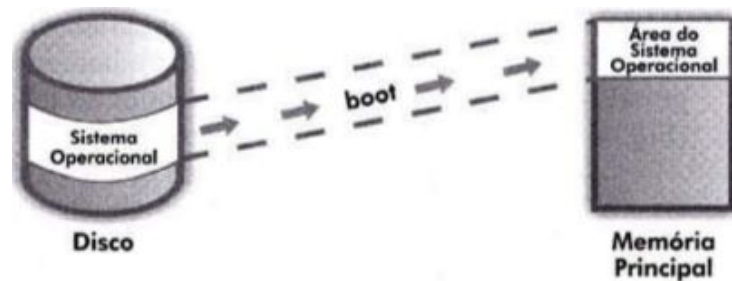




# ATIVAÇÃO/DESATIVAÇÃO DO SISTEMA

- O processo de ativação, também conhecido como “boot” carrega o Sistema Operacional para a memória do equipamento;
- Existe ainda o processo de desativação ou shutdown, que permite às aplicações e componentes do sistema operacional sejam desativados ordenadamente, para preservar a sua integridade.

O procedimento de ativação se inicia com a execução de um programa chamado boot loader, que se localiza em um endereço fixo da memória ROM da máquina. Este programa chama a execução de outro programa conhecido como POST (Power-On Self Test), que identifica possíveis problemas de hardware no equipamento.



# ARTIGO PARA ENTENDIMENTO DAS SYSTEM CALLS

- 1) Quais serviços o artigo cita que podemos invocar através de chamadas de sistema?
- 2) Qual foi o objetivo da System Call criada pelo autor (o que ela registrava)?
- 3) Esse tipo de experiência pode ser executada em qualquer Sistema Operacional? Explique sua resposta.
- 4) Qual a utilidade prática é possível destacar para este tipo de operação (criação de System Calls)?

# O QUE FICOU NA MENTE?

1. o que é o núcleo do sistema e quais são suas principais funções?
2. O que são instruções privilegiadas e não-privilegiadas? Qual a relação dessas instruções com os modos de acesso?
3. Explique como funciona a mudança de modos de acesso e dê um exemplo de como um programa faz uso desse mecanismo.
4. Como o kernel do sistema operacional pode ser protegido pelo mecanismo de modos de acesso?

# O QUE FICOU NA MENTE?

5. Por que as rotinas do sistema operacional possuem instruções privilegiadas?

6. O que é uma system call e qual sua importância para a segurança do sistema? Como as system calls são utilizadas por um programa?

7. Quais das instruções a seguir devem ser executadas apenas em modo kernel? Alterar a data e hora do sistema, somar duas variáveis declaradas dentro do programa, realizar um desvio para uma instrução dentro do próprio programa e acessar diretamente posições no disco.