



Centro  
Universitário  
FMU

# FMU

Professor Lima

Sistemas Operacionais

### Aula Anterior

Conceito e Evolução dos  
Sistemas Operacionais

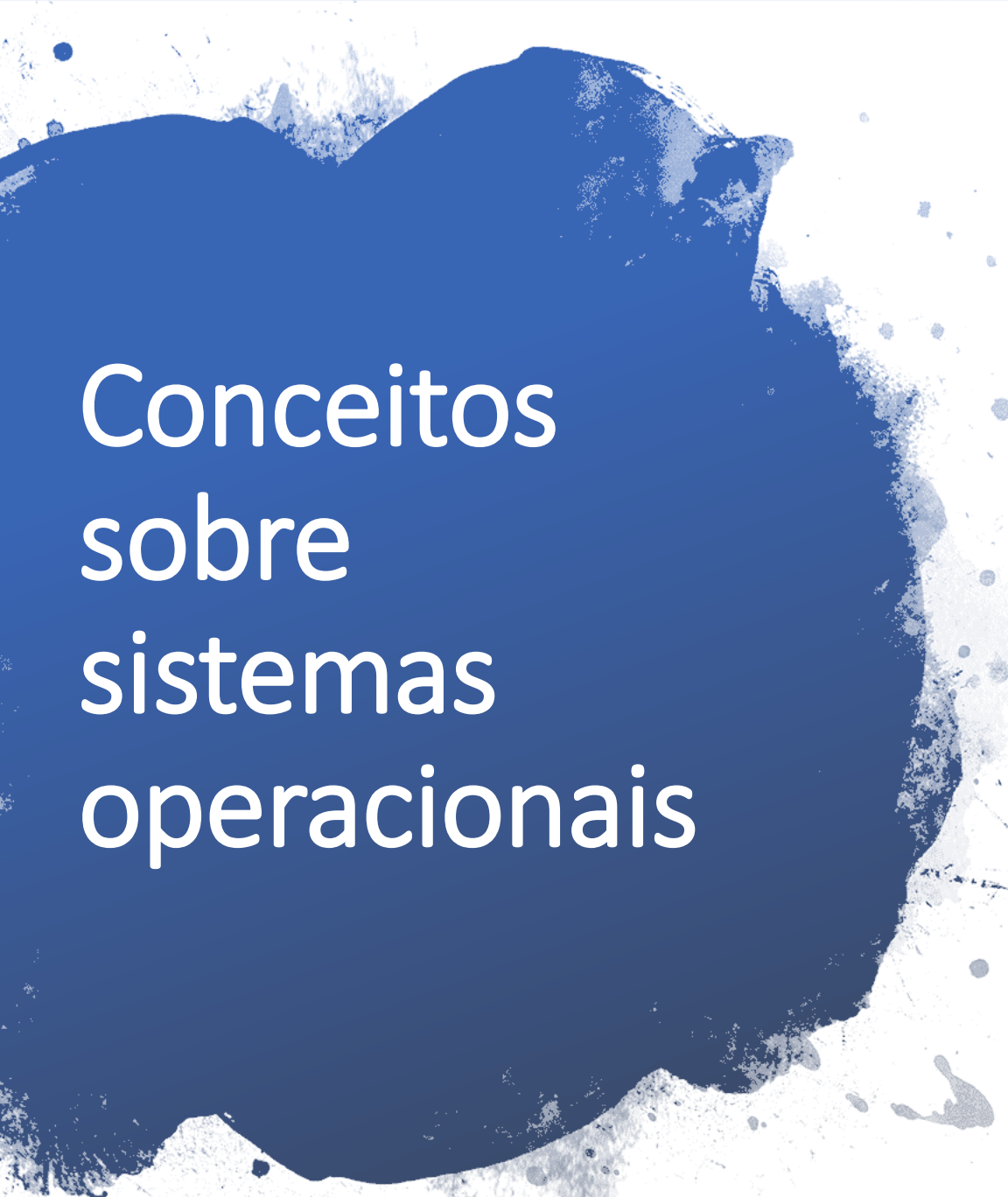
### Aula de Hoje

Gerenciamento de Recursos  
Conceitos Básicos  
Aplicabilidade

### Próxima Aula

Gerenciamento de  
Processos  
Principais Conceitos  
sobre Processo  
Diagrama de  
Execução/de Estados  
do Processo

# Cronograma



# Conceitos sobre sistemas operacionais

- A maioria dos sistemas operacionais fornece certos conceitos e abstrações básicos, como processos, espaços de endereçamento e arquivos, que são fundamentais para entendê-los

# Processos

- Um conceito fundamental para todos os sistemas operacionais é o de **processo**. Um processo é basicamente um programa em execução. Associado a cada processo está o seu **espaço de endereçamento**, uma lista de posições de memória, que vai de 0 até um máximo, que esse processo pode ler e escrever.
- O espaço de endereçamento contém o programa executável, os dados do programa e sua pilha.



# Processos

- Também associado a cada processo está um conjunto de recursos, normalmente incluindo registradores (que incluem o contador de programa e o ponteiro para a pilha), uma lista dos arquivos abertos, alarmes pendentes, listas de processos relacionados e todas as demais informações necessárias para executar um programa.
- Um processo é fundamentalmente um contêiner que armazena todas as informações necessárias para executar um programa



# Processos

- Em muitos sistemas operacionais, todas as informações relativas a um processo — que não sejam o conteúdo de seu próprio espaço de endereçamento — são armazenadas em uma tabela do sistema operacional denominada **tabela de processos**, que é um arranjo (ou uma lista encadeada) de estruturas, uma para cada processo existente.



# Processos

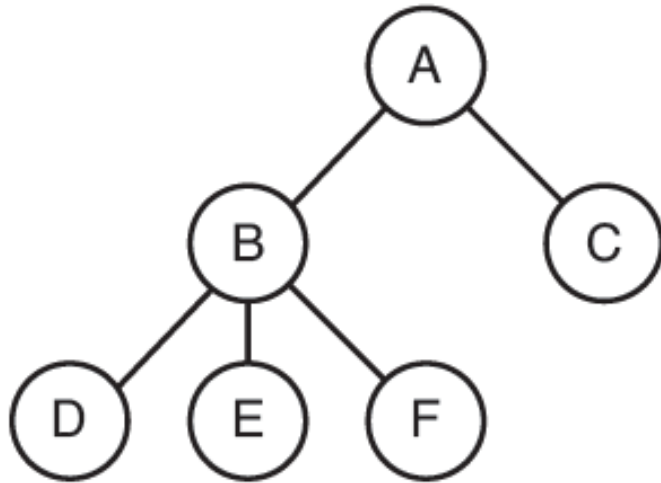
- Considere um exemplo típico: um processo denominado **interpretador de comandos** ou **shell** lê os comandos de um terminal.
- O usuário acaba de digitar um comando pedindo que um programa seja compilado.
- O shell deve então criar um novo processo, que executará o compilador.
- Assim que esse processo tiver terminado a compilação, ele executa uma chamada de sistema para se auto finalizar.





# Processos

- Se um processo pode criar um ou mais processos (chamados **processos filhos**), e se esses processos, por sua vez, puderem criar outros processos filhos, rapidamente chegaremos a uma estrutura de árvores.



Uma árvore de processo.  
O processo *A* criou dois processos filhos, *B* e *C*. O processo *B* criou três processos filhos, *D*, *E* e *F*.

- Processos relacionados que estiverem cooperando para executar alguma tarefa precisam frequentemente se comunicar um com o outro e sincronizar suas atividades.
- Essa comunicação é chamada de **comunicação entre processos**



# Espaços de endereçamento

- Todo computador tem alguma memória principal que utiliza para armazenar programas em execução.
- Em um sistema operacional muito simples, apenas um programa por vez está na memória.
- Para executar um segundo programa, o primeiro tem de ser removido e o segundo, colocado na memória.
- Sistemas operacionais mais sofisticados permitem que
- múltiplos programas estejam na memória ao mesmo tempo.

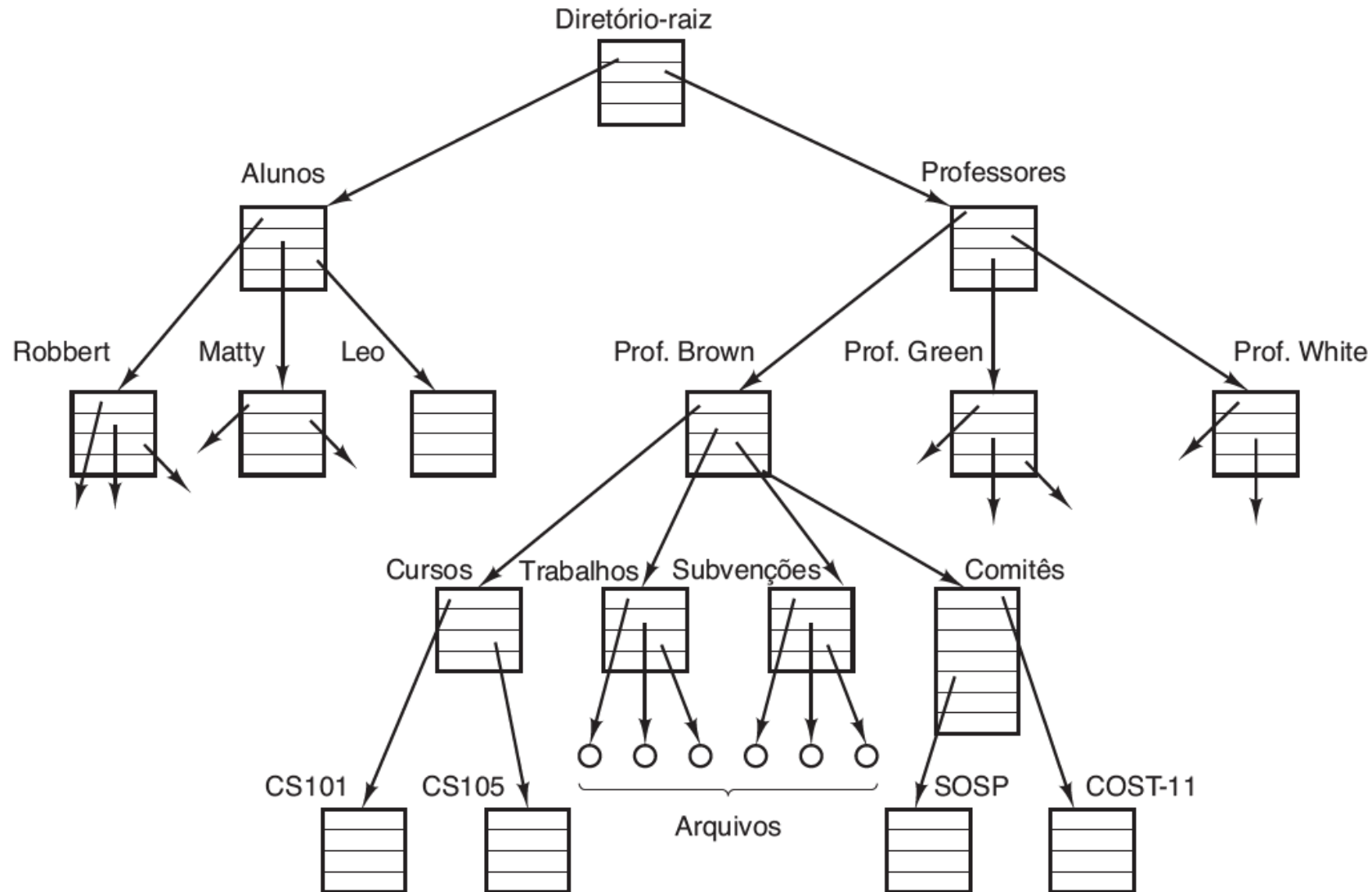


# Arquivos

- Outro conceito fundamental que compõe praticamente todos os sistemas operacionais é o sistema de arquivos.
- Como se observou anteriormente, uma das principais funções do sistema operacional é ocultar as peculiaridades dos discos e de outros dispositivos de E/S, fornecendo ao programador um modelo de arquivos agradável e claro, independentemente de dispositivos.
- Chamadas de sistema são obviamente necessárias para criar, remover, ler e escrever arquivos. Antes que possa ser lido, um arquivo deve ser localizado no disco, aberto e, depois de lido, ser fechado. Desse modo, chamadas de sistema são fornecidas para fazer essas tarefas.



# Sistema de arquivos para um departamento universitário.



# Entrada e saída

- Todos os computadores têm dispositivos físicos para entrada e saída.
- Afinal, para que serviria um computador se os usuários não pudessem dizer o que deve ser feito e não conseguissem verificar os resultados depois do trabalho solicitado?
- Existem vários tipos de dispositivos de entrada e saída, como teclados, monitores e impressoras
- Consequentemente, todo sistema operacional possui um subsistema de E/S para gerenciar seus dispositivos de E/S.



# O interpretador de comandos (shell)

- O sistema operacional é o código que executa as chamadas de sistema. Editores, compiladores, montadores, ligadores (*linkers*) e interpretadores de comandos não fazem, com certeza, parte do sistema operacional, mesmo sendo importantes e úteis.
- Existem muitos shells, dentre eles o *sh*, o *csh*, o *ksh* e o *bash* do *Unix/Linux* e do *Windows* o *prompt de comando*.



# O interpretador de comandos (shell)

- Quando um usuário se conecta, um shell é iniciado.
- Este tem o terminal como entrada-padrão e saída-padrão.
- Ele inicia emitindo um caractere de **prompt** (*prontidão*) — por exemplo, o cifrão —, que diz ao usuário que o shell está esperando receber um comando..
- Atualmente, muitos computadores pessoais usam uma interface gráfica GUI.



# Gerenciamento de Recursos

- Imagine o que aconteceria se três programas sendo executados em um computador tentassem todos imprimir suas saídas simultaneamente na mesma impressora. As primeiras linhas da saída impressa poderiam ser do programa 1, as linhas seguintes do programa 2 e, então, algumas do programa 3 etc.
- O resultado seria o caos. O sistema operacional pode trazer ordem ao caos em potencial, armazenando no disco, por programa, toda a saída destinada à impressora.
- Quando um programa tiver terminado, o sistema operacional envia para a impressora a sua saída armazenada no arquivo em disco, enquanto, ao mesmo tempo, um outro programa poderá continuar gerando mais saída, ignorando o fato de que ela não está realmente indo para a impressora (ainda).



# Gerenciamento de Recursos

- Quando um computador (ou uma rede) tem vários usuários, a necessidade de gerenciar e proteger a memória, dispositivos de E/S e outros recursos é ainda maior, pois os usuários poderiam interferir uns com os outros.
- Além disso, os usuários freqüentemente precisam compartilhar não apenas o hardware, mas também informações (arquivos, bancos de dados etc.).
- Em resumo, essa visão do sistema operacional sustenta que sua principal tarefa é controlar quem está usando qual recurso, garantir os pedidos de recursos, medir a utilização e mediar pedidos conflitantes de diferentes programas e usuários.

O gerenciamento de recursos inclui a multiplexação (compartilhamento) de recursos de duas maneiras: no **tempo** e no **espaço**.

- Quando um recurso é multiplexado no tempo, diferentes programas ou usuários o utilizam por turnos: primeiro um deles utiliza o recurso, depois outro e assim por diante.
- Por exemplo, com apenas uma CPU e vários programas que queiram ser executados nela, o sistema operacional primeiramente aloca a CPU para um programa e, depois, após o programa ter executado o suficiente, outro programa utiliza a CPU, depois outro e, finalmente, o primeiro programa novamente.



O gerenciamento de recursos inclui a multiplexação (compartilhamento) de recursos de duas maneiras: no **tempo** e no **espaço**.

- Determinar como o recurso é multiplexado no **tempo** – quem vem em seguida e por quanto tempo – é tarefa do sistema operacional.
- Outro exemplo de multiplexação no **tempo** é o compartilhamento da impressora.
- Quando várias tarefas de impressão são enfileiradas em uma única impressora, uma decisão precisa ser tomada com relação a qual das tarefas deve ser impressa a seguir.



O gerenciamento de recursos inclui a multiplexação (compartilhamento) de recursos de duas maneiras: no **tempo** e no **espaço**.

- O outro tipo de multiplexação é a no **espaço**. Em vez dos clientes atuarem por turnos, cada um deles recebe parte do recurso.
- Por exemplo, normalmente, a memória principal é dividida entre vários programas que estejam em execução para que cada um possa estar residente ao mesmo tempo (por exemplo, para utilizar a CPU por turnos).
- Supondo que haja memória suficiente para conter múltiplos programas, é mais eficiente manter vários programas na memória de uma vez do que alocar toda ela para um único programa, especialmente se ele precisar apenas de uma pequena fração do total.



O gerenciamento de recursos inclui a multiplexação (compartilhamento) de recursos de duas maneiras: no **tempo** e no **espaço**.

- É claro que isso levanta problemas de imparcialidade, proteção etc., e fica por conta do sistema operacional resolvê-los.
- Outro recurso multiplexado no espaço é o disco (rígido). Em muitos sistemas, um único disco pode conter arquivos de muitos usuários ao mesmo tempo. Alocar espaço em disco e controlar quem está usando quais blocos de disco é uma típica tarefa de gerenciamento de recursos do sistema operacional.




# Chamadas de sistema (system calls)

Vimos que os sistemas operacionais têm duas funções principais: fornecer abstrações aos programas de usuários e administrar os recursos do computador



Em sua maior parte, a interação entre programas de usuário e o sistema operacional lida com a primeira; por exemplo, criar, escrever, ler e excluir arquivos.



A parte de gerenciamento de recursos é, em grande medida, transparente para os usuários e feita automaticamente.



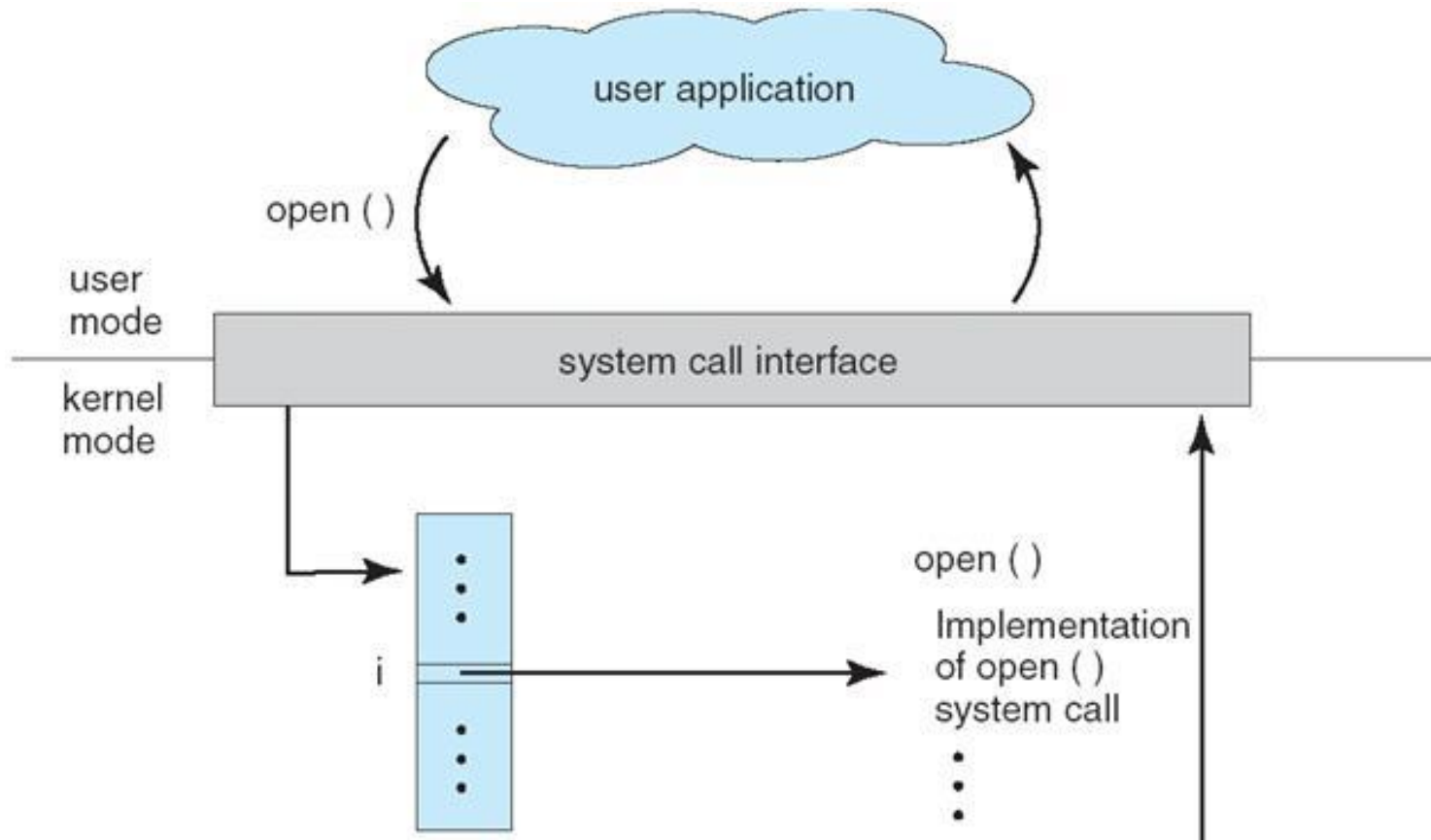
## System Calls

- Desse modo, a interface entre o sistema operacional e os programas de usuários trata primeiramente sobre como lidar com as abstrações.

# System Calls (chamadas de sistemas)

- As chamadas de sistema disponíveis na interface variam de um sistema operacional para outro (embora os conceitos básicos tendam a ser parecidos).
- Qualquer computador com uma única CPU pode executar somente uma instrução por vez.
- Se um processo estiver executando um programa de usuário em modo usuário e precisar de um serviço do sistema como ler dados de um arquivo, terá de executar uma instrução para transferir o controle ao sistema operacional.
- Este verifica os parâmetros para, então, descobrir o que quer o processo que está chamando





## Procedimento de uma chamada de sistemas

- Em seguida, ele executa uma chamada de sistema e retorna o controle para a instrução seguinte à chamada.
- Em certo sentido, fazer uma chamada de sistema é como realizar um tipo especial de chamada de rotina, só que as chamadas de sistema fazem entrar em modo núcleo e as chamadas de rotina, não.

## Questões

1. Qual é a diferença entre modo núcleo e modo usuário? Por que a diferença é importante para um sistema operacional?
2. Qual é a finalidade de uma chamada de sistema em um sistema operacional?
3. Quais das seguintes instruções devem ser permitidas apenas no modo núcleo?
  - (a) Desabilitar todas interrupções.
  - (b) Ler o relógio de hora do dia.
  - (c) Configurar o relógio de hora do dia.
  - (d) Mudar o mapeamento da memória.
4. O que é multiplexação?

# *Santos Dumount*

- “Não se espante com a altura do vôo. Quanto mais alto, mais longe do perigo. Quanto mais você se eleva, mais tempo há de reconhecer uma pane. É quando se está próximo do solo que se deve desconfiar.”

ProfessorLima.Com