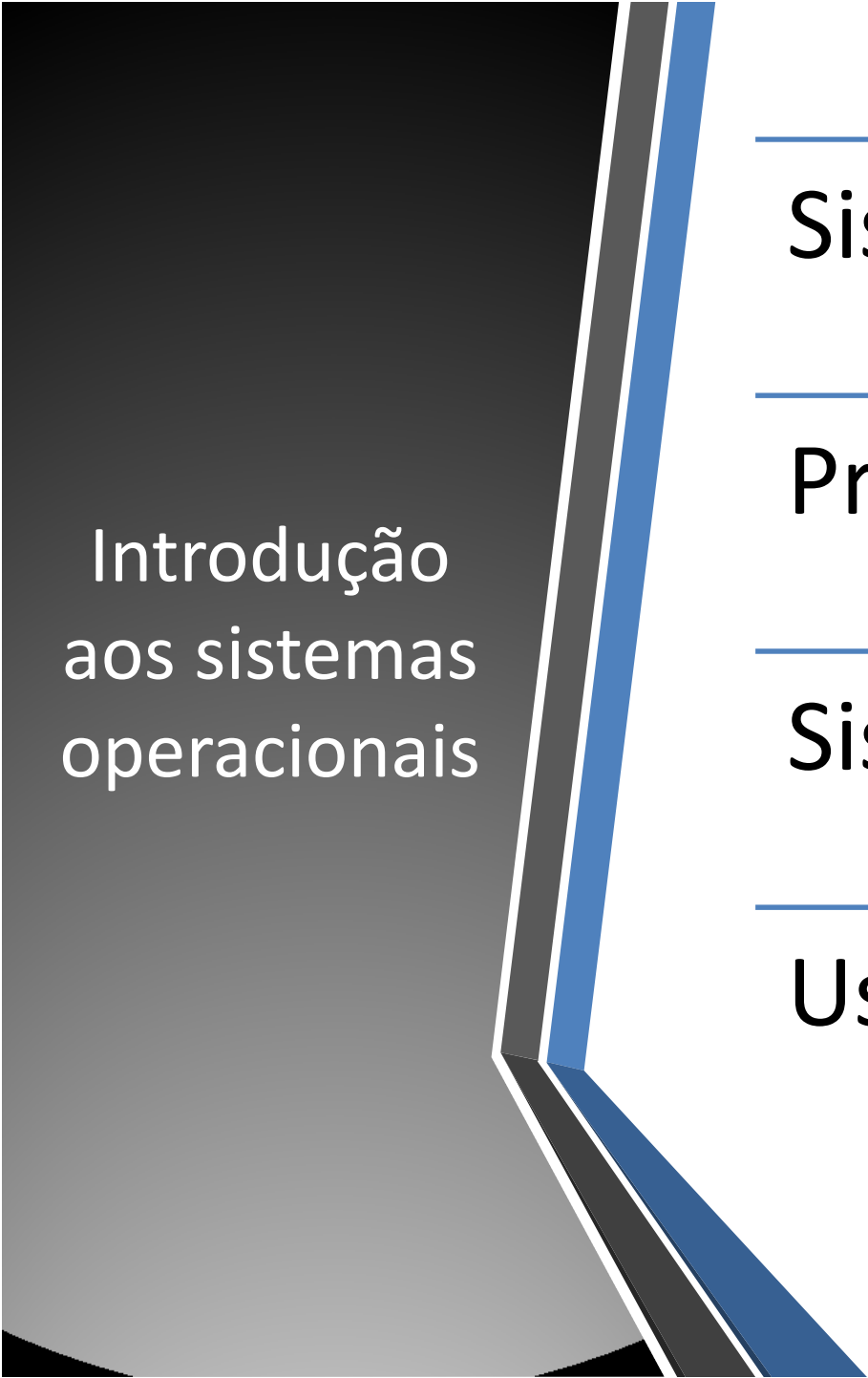




# Introdução à Ciência da Computação





# Introdução aos sistemas operacionais

---

Sistema Operacionais

---

Processos

---

Sistema de Arquivos

---

Usuários

# Sistema Operacional

**É o principal software instalado em um computador, sendo responsável pela interação entre o hardware e demais softwares.**


Cuida do gerenciamento dos recursos disponíveis em um computador.


- Permissão de acesso a determinados dados da memória
- Acesso aos periféricos
- Controle de usuários
- Utilização de energia elétrica


**Exemplos: MS-DOS, Windows, UNIX, Linux, MacOS, Android, iOS, etc.**

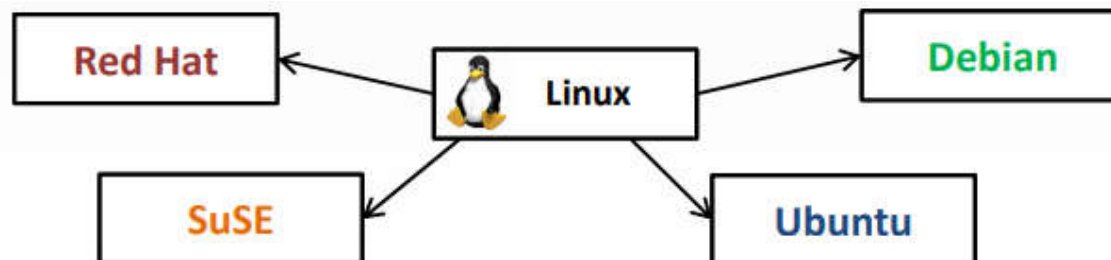


# Versões e Distribuições dos Principais Sistemas Operacionais

 <b>Windows (Microsoft)</b>	
Versão	Ano
Windows 3.1	1992
Windows 95	1995
Windows 98	1998
Windows ME	2000
Windows XP	2001
Windows Vista	2006
Windows 7	2008
Windows 8	2012
Windows 8.1	2013
Windows 10	2015

 <b>MAC OS X (Apple)</b>	
Versão	Ano
Cheetah (10.0)	2001
Puma (10.1)	2002
Jaguar (10.2)	2003
Panther (10.3)	2005
Tiger (10.4)	2007
Leopard (10.5)	2009
Snow Leopard (10.6)	2011
Lion (10.7)	2012
Mountain Lion (10.8)	2013
Mavericks (10.9)	2014
Yosemite (10.10)	2015
El Capitan (10.11)	2016
Sierra	2016
High Sierra	2017
Mojavi	2018

 <b>Android (Google)</b>	
Distribuição	Ano
Cupcake (1.5)	2009
Donut (1.6)	2009
Eclair (2.0–2.1)	2009
Froyo (2.2)	2010
Gingerbread (2.3)	2010
Honeycomb (3.0)	2011
Ice Cream Sandwich (4.0)	2011
Jelly Bean	2012
KitKat	2013
Lollipop	2014
Mashmallow	2015
Nougat	2016
Oreo	2017
P	2018



Camadas de Software e Hardware: o Sistema Operacional é um software que gerencia os recursos (hardware e software) disponíveis em um computador.



# Inicialização do Sistema Operacional

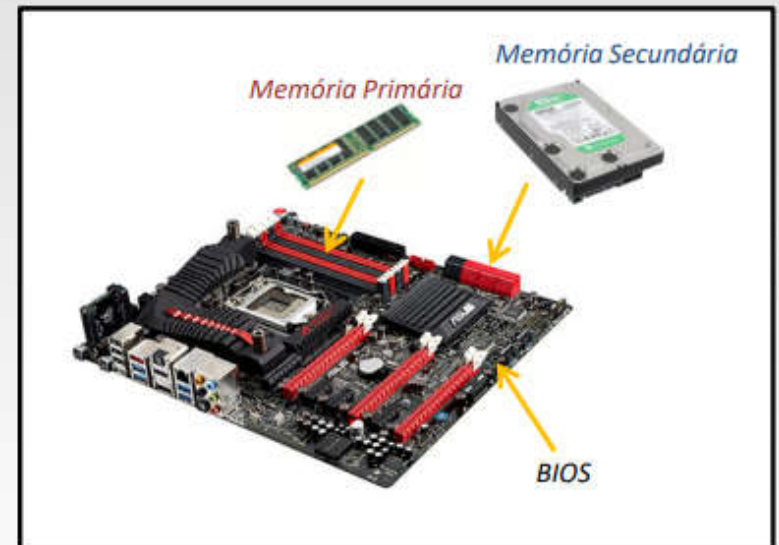
A **BIOS** (*Basic Input Output System*) faz a checagem do hardware e busca pelo Sistema Operacional em uma **Memória Secundária**



Na **Memória Secundária** (trilha 0, setor 0) são encontradas as informações iniciais do **Sistema Operacional**



O **Sistema Operacional** é carregado na **Memória Principal** e entra em **execução**, preparando o sistema para executar suas funções e interagir com o **usuário**



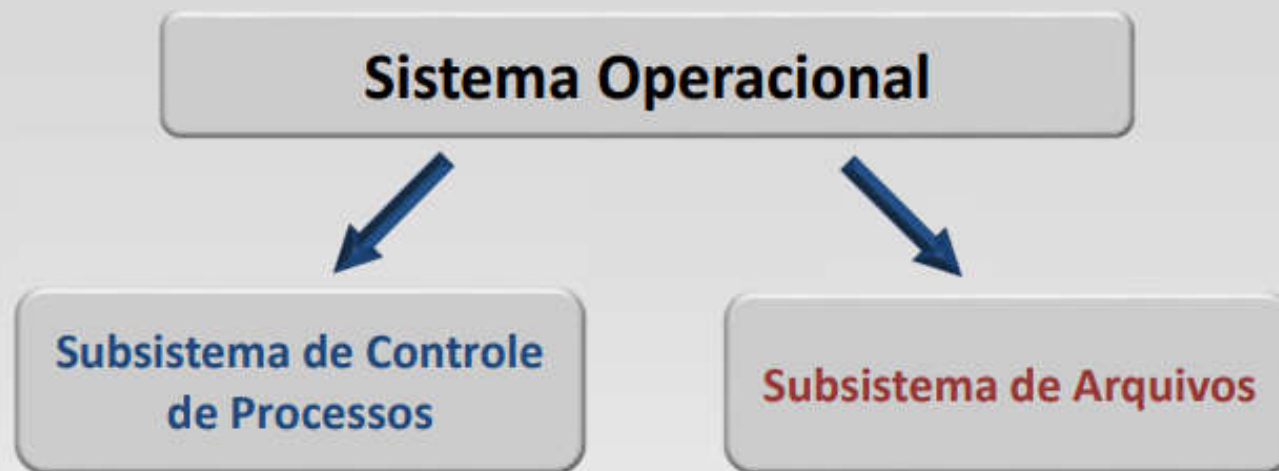


## Chamadas de Sistemas

Essas chamadas de sistemas podem ser de duas categorias:

- Processos;
- Sistema de Arquivos;





**Processo** → É um **programa** em **execução**

**Arquivo** → É uma **sequência de dados binários** que segue uma regra estrutural de armazenamento em uma **memória secundária** (memória de massa)



## Serviços do Sistema Operacional

### Subsistema de Controle de Processos

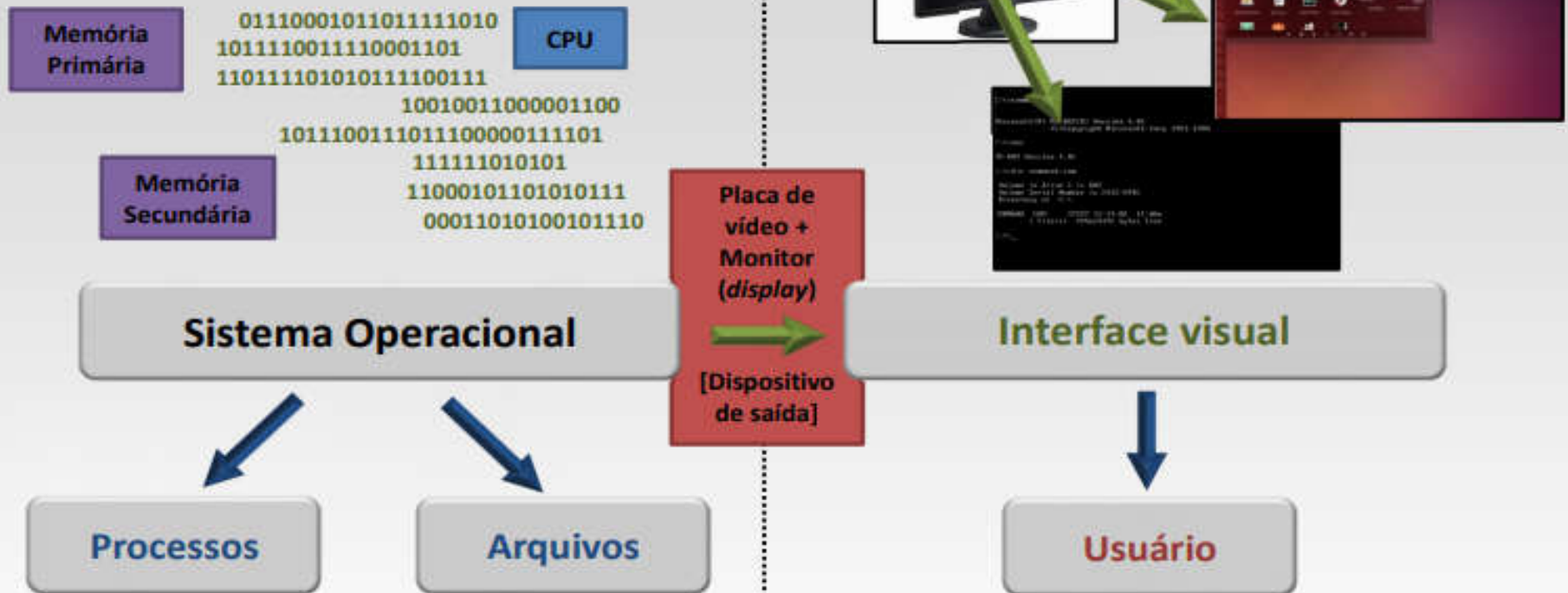
- **Controle de execução de processos**
  - Criação
  - Finalização
  - Comunicação entre processos
- **Alocação de memória principal para execução de processos**
  - Carregamento de programas na memória principal
- **Escalonamento de Processos**
  - Gerenciamento de execução de processos na memória principal

### Subsistema de Arquivos

- **Alocação/recuperação de dados em memória secundária**
  - Armazenamento organizado de dados em memória secundária
- **Controle dos dispositivos periféricos**
  - Controle de acesso aos monitores, memórias secundárias, rede, etc.

# Sistemas Operacionais

## ➤ Interface com o usuário



## ➤ Gerenciamento/Organização

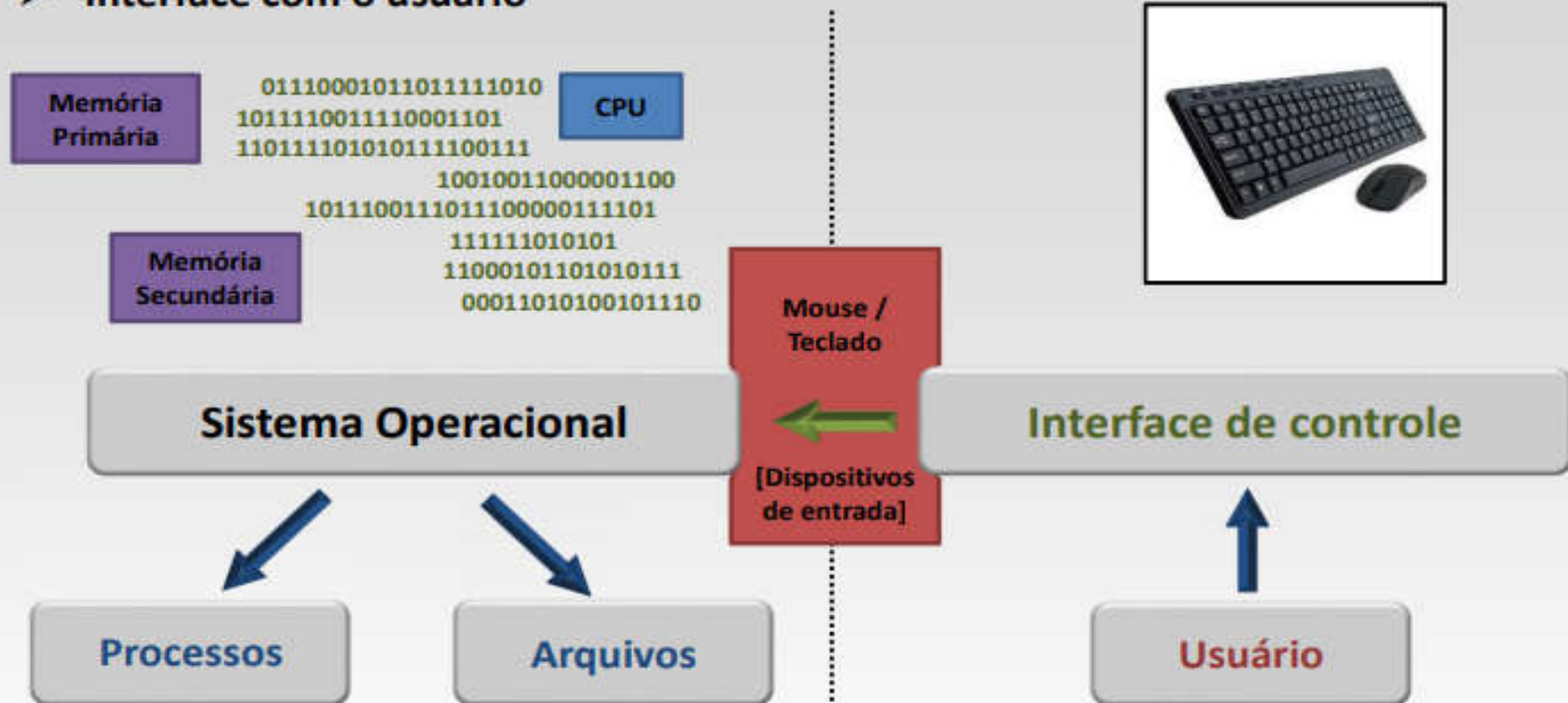
- Processos
- Arquivos
- Memória RAM
- Processamento
- Dispositivos I/O

## ➤ Visualização/saída de dados

- Interface de visualização textual
  - Shell
  - Terminal
- Interface de visualização gráfica
  - Janelas

# Sistemas Operacionais

## ➤ Interface com o usuário



## ➤ Gerenciamento/Organização

- Processos
- Arquivos
- Memória RAM
- Processamento
- Dispositivos I/O

## ➤ Inserção/entrada de dados

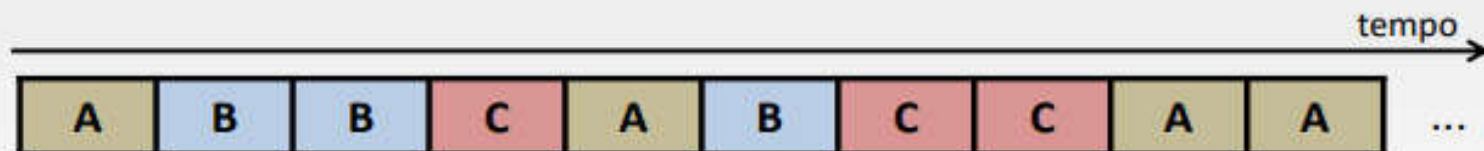
- Dispositivo de entrada textual
  - Teclado
- Dispositivo de posição gráfica
  - Mouse



# Sistemas Operacionais

## Pseudo-paralelismo

- Sistemas operacionais **multitarefa** (Windows, UNIX, etc.) permitem a execução de mais de um programa “ao mesmo tempo”. Na realidade, o Sistema Operacional **intercala a execução entre os programas (processos)**, dando a sensação de paralelismo devido a alta frequência de execução das instruções presentes na memória do computador
- Cada pedaço de um processo (conjunto de instruções de um programa) executa durante determinado tempo, sendo interrompido para que o pedaço de outro processo execute

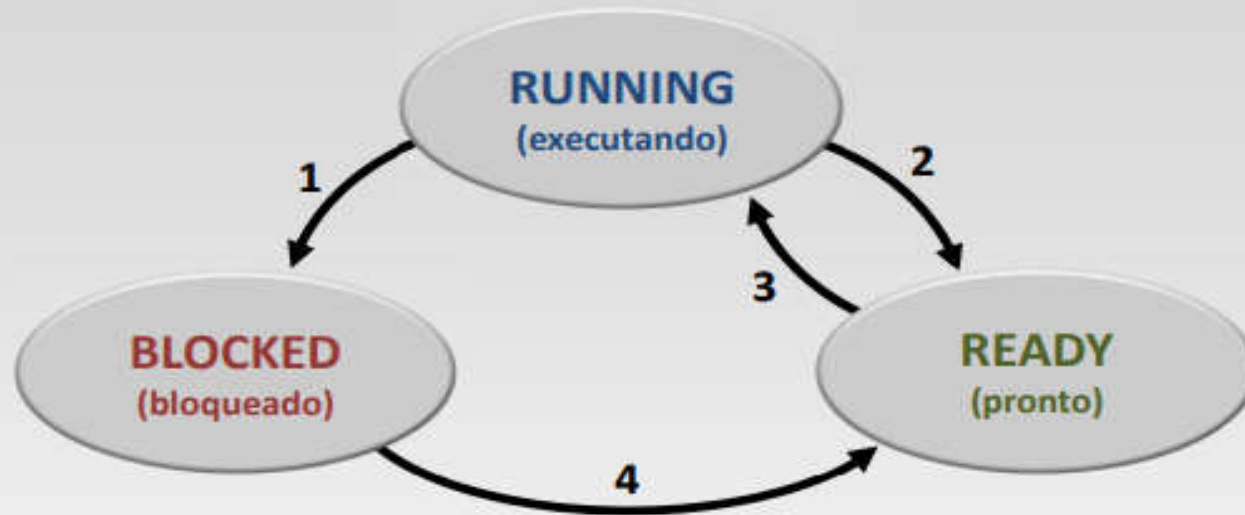


Exemplo: um computador com frequência de 4GHz executa 4.000.000.000 instruções por segundo

- A não ser que o computador seja *multicore* (*dual-core*, *quad-core*, etc.), o paralelismo na execução de tarefas não existe de fato nos microcomputadores

# Processos

## ➤ Estados de um Processo



**RUNNING** (executando) ➡ o processo está utilizando a CPU (executando suas instruções)

**BLOCKED** (bloqueado) ➡ o processo está parado aguardando um evento externo

**READY** (pronto) ➡ o processo está temporariamente parado, pois outro processo está em execução

# Processos

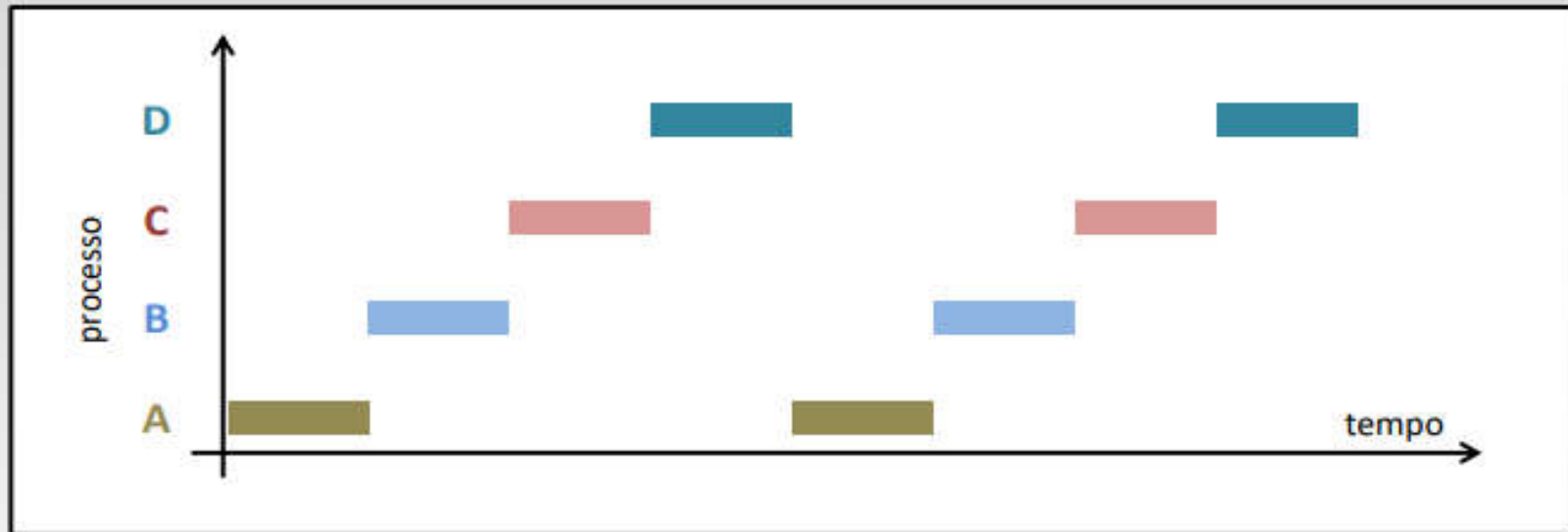
## ➤ Estados de um Processo



- 1 – O **processo** em execução entra em estado bloqueado na **espera de um recurso**
- 2 – O **escalonador** de processos interrompe a execução de um **processo** para que **outro processo** seja executado
- 3 – O **escalonador** de processos permite que o **processo** continue executando
- 4 – O **recurso esperado** está disponível e o **processo** se desbloqueia

# Processos

## ➤ Escalonamento de Processos



## ➤ Tipos de Escalonamento de Processos

- Round Robing
- Por prioridade (*Priority Scheduling*)
- Menor processo primeiro (*Shortest Job First*)
- Loteria (*Lottery Scheduling*)

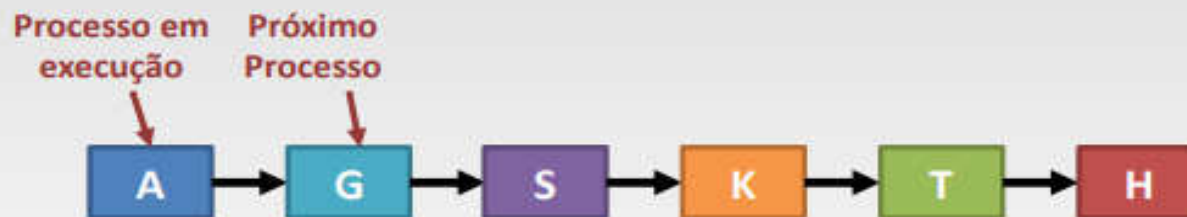


# Processos

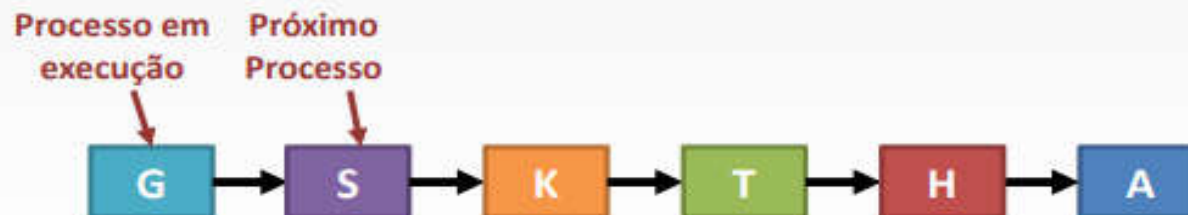
## ➤ Escalonamento de Processos

### ➤ Round Robing

- Os processos são executados durante um intervalo de tempo (*quantum*)
- Se o processo em execução não terminar antes do seu *quantum* chegar ao fim, o próximo processo na lista entra em execução e o que estava executando anteriormente vai para o final da lista



... após o *quantum*, se o processo em execução (A) não terminou, ele vai para o final da lista e o próximo processo (G) entra em execução...

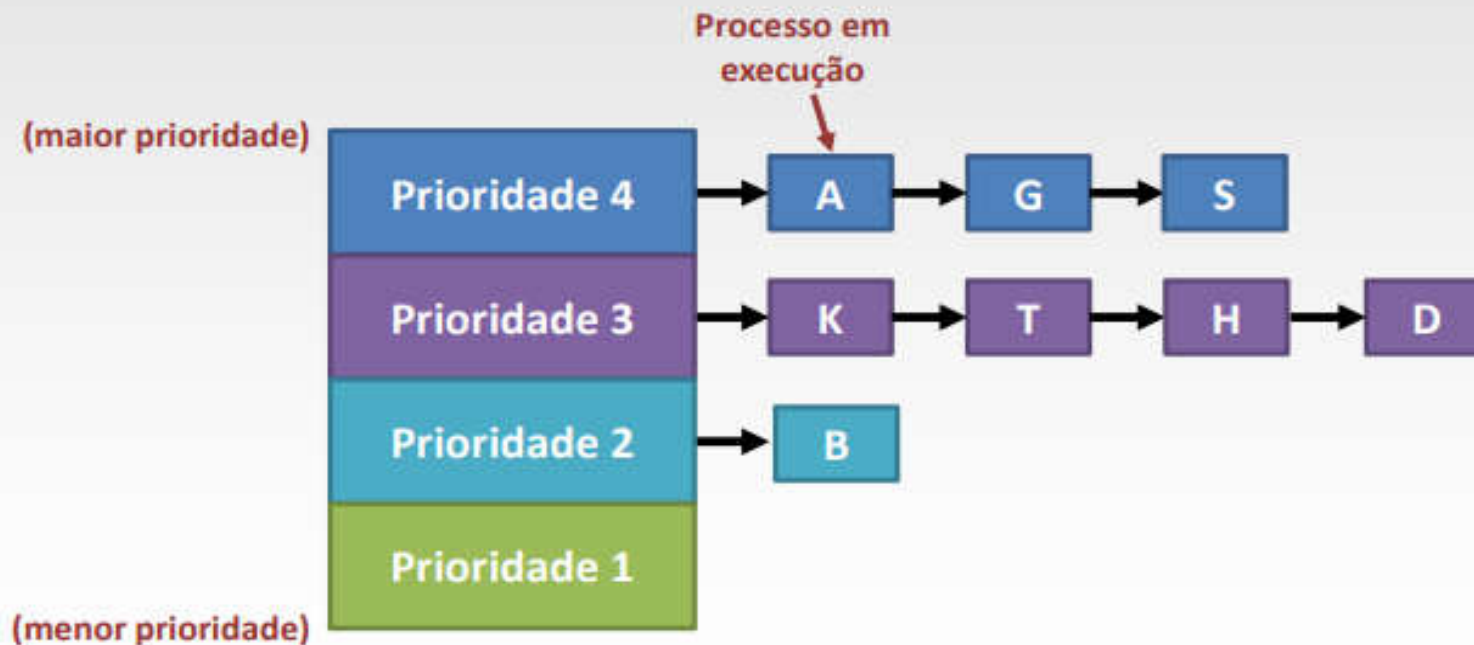


# Processos

## ➤ Escalonamento de Processos

### ➤ Por prioridade (*Priority Scheduling*)

- Cada processo possui uma prioridade associada
- **Processos com prioridades mais elevadas executam primeiro**
- Para evitar que processos com prioridades mais altas executem indefinidamente, o escalonador pode diminuir a prioridade dos processos, dando oportunidade para processos com menor prioridade executar



# Processos

## ➤ Escalonamento de Processos

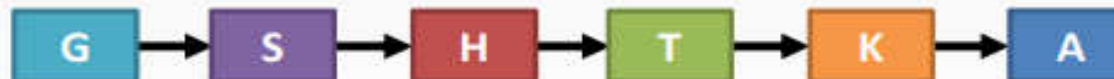
### ➤ Menor processo primeiro (*Shortest Job First*)

- O processo de “mais curto” (mais rápido) será executado primeiro

Processos a serem executados:



Processos escalonados (sequência de execução):



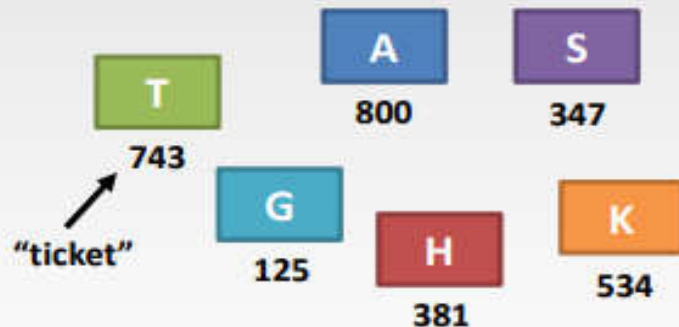
# Processos

## ➤ Escalonamento de Processos

### ➤ Loteria (*Lottery Scheduling*)

- Cada processo recebe um "ticket"
- O escalonador faz um sorteio aleatório para selecionar qual o próximo processo a ser executado

Processos a serem executados:



Escalonador sorteia o próximo processo:

[ 743, 800, 347, **125**, 381, 534 ]

Processos com o "ticket" 125 é executado:

Escalonador sorteia o próximo processo:

[ 743, 800, 347, 381, **534** ]

Processos com o "ticket" 534 é executado:

Escalonador sorteia o próximo processo:

[ 743, **800**, 347, 381 ]

Processos com o "ticket" 800 é executado:

⋮



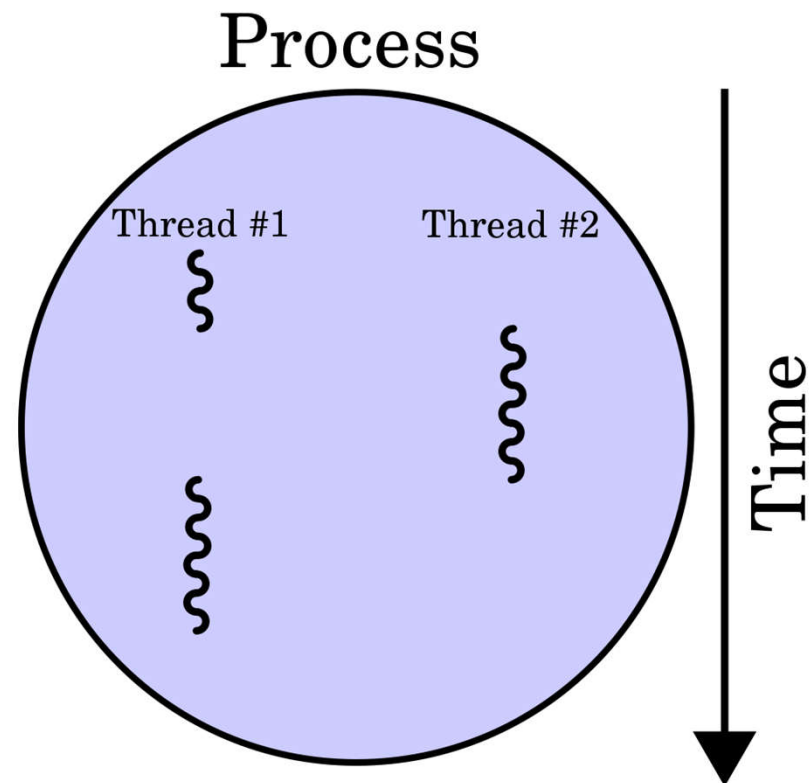
# Threads

## Threads

- § *Threads são “processos leves”.*
- § *É uma maneira de um processo dividir a si mesmo em duas ou mais tarefas que podem ser executadas concorrentemente.*
- § *Threads permitem que o usuário de um programa interaja com os componentes visuais de controle do programa enquanto cálculos e operações estão sendo realizadas “simultaneamente”*

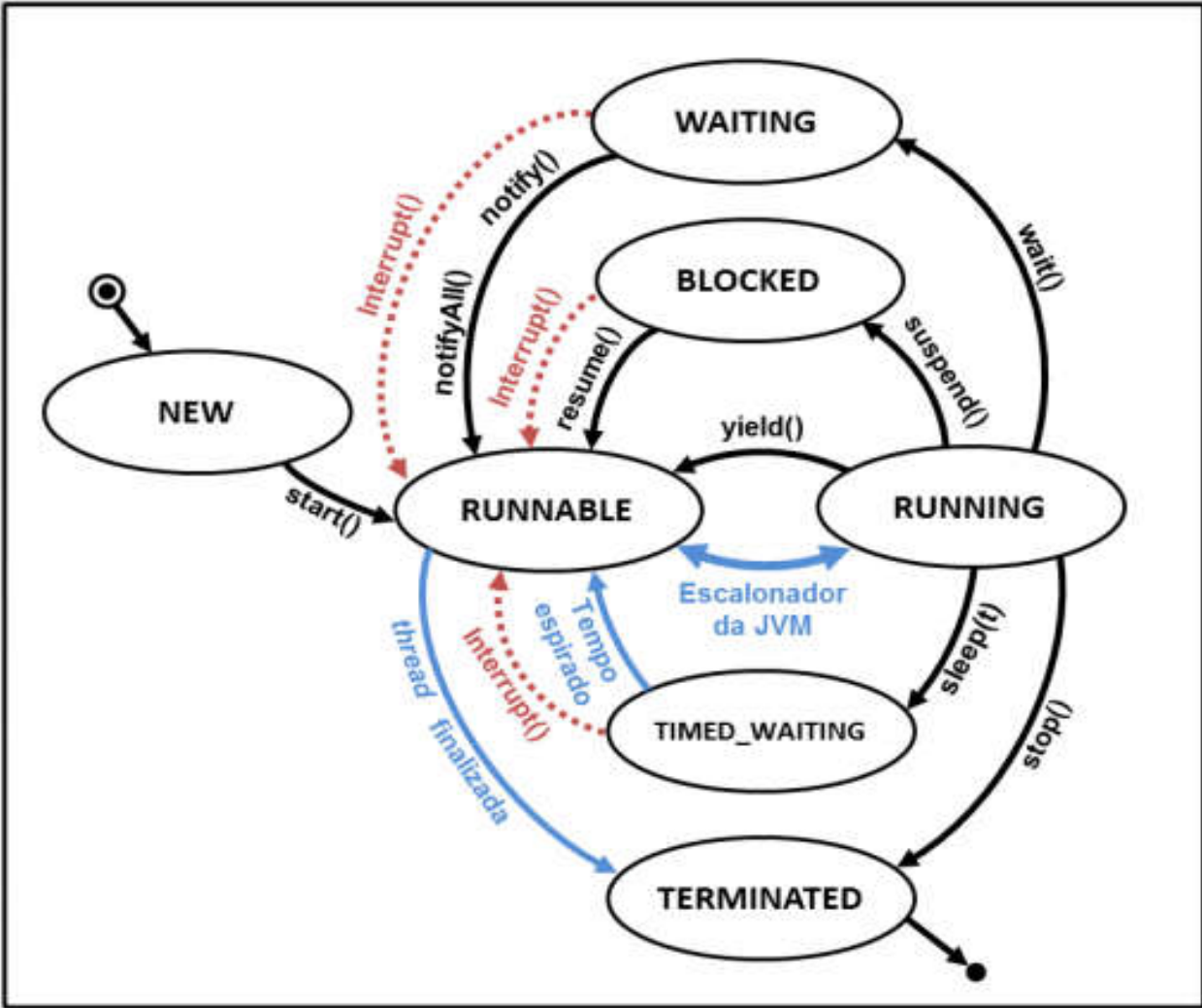
### Diferenças entre Threads e Processos

- § *Processos são independentes, enquanto Threads existem como um subconjunto de um processo.*
- § *Processos possuem espaços de memória independentes, enquanto Threads compartilham seus espaços de memória.*
- § *O mecanismo de comunicação entre os processos é gerenciado pelo Sistema Operacional, enquanto a comunicação entre Threads é realizada diretamente entre elas. Logo, a comunicação entre processos é mais lenta se comparada à comunicação entre Threads.*



## Threads

### *Estados e transição entre estados de uma Thread na Linguagem de Programação Java*





# Sistema de Arquivos

**Um sistema de arquivo é um conjunto de estruturas lógicas e procedimentos que permitem ao Sistema Operacional controlar o acesso aos dados armazenados em uma unidade de memória secundária.**

*Cada Sistema Operacional “entende” determinados sistemas de arquivos.*

Com o crescimento da capacidade de armazenamento dos discos e aumento do tamanho dos arquivos, sistemas de arquivos mais complexos e robustos são necessários.

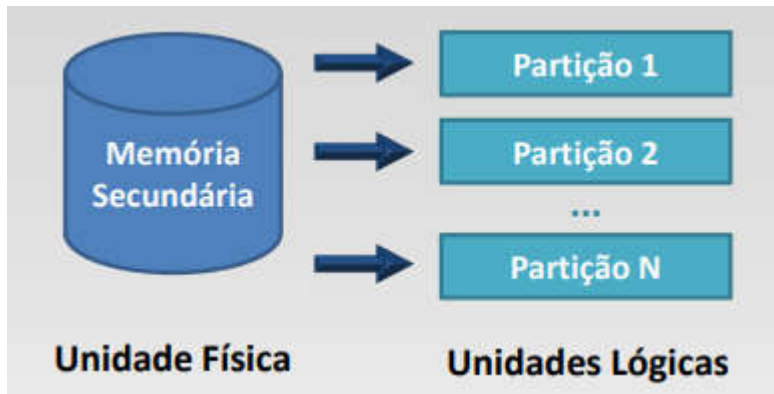
## **São exemplos de sistemas de arquivos:**

- FAT (File Allocation Table) – 12 bits [disquetes]
- FAT16 – 16 bits [MS-DOS; Windows 95]
- FAT32 – 32 bits [Windows 98; Padrão em Pendrives e Cartões de Memória]
- NTFS (New Technology File System) – 64 bits [Windows NT em diante]
- EXT (Extended File System) [Linux]
- EXT2, EXT3, EXT4 [Linux]
- APFS [Apple File System]



# Sistema de Arquivos

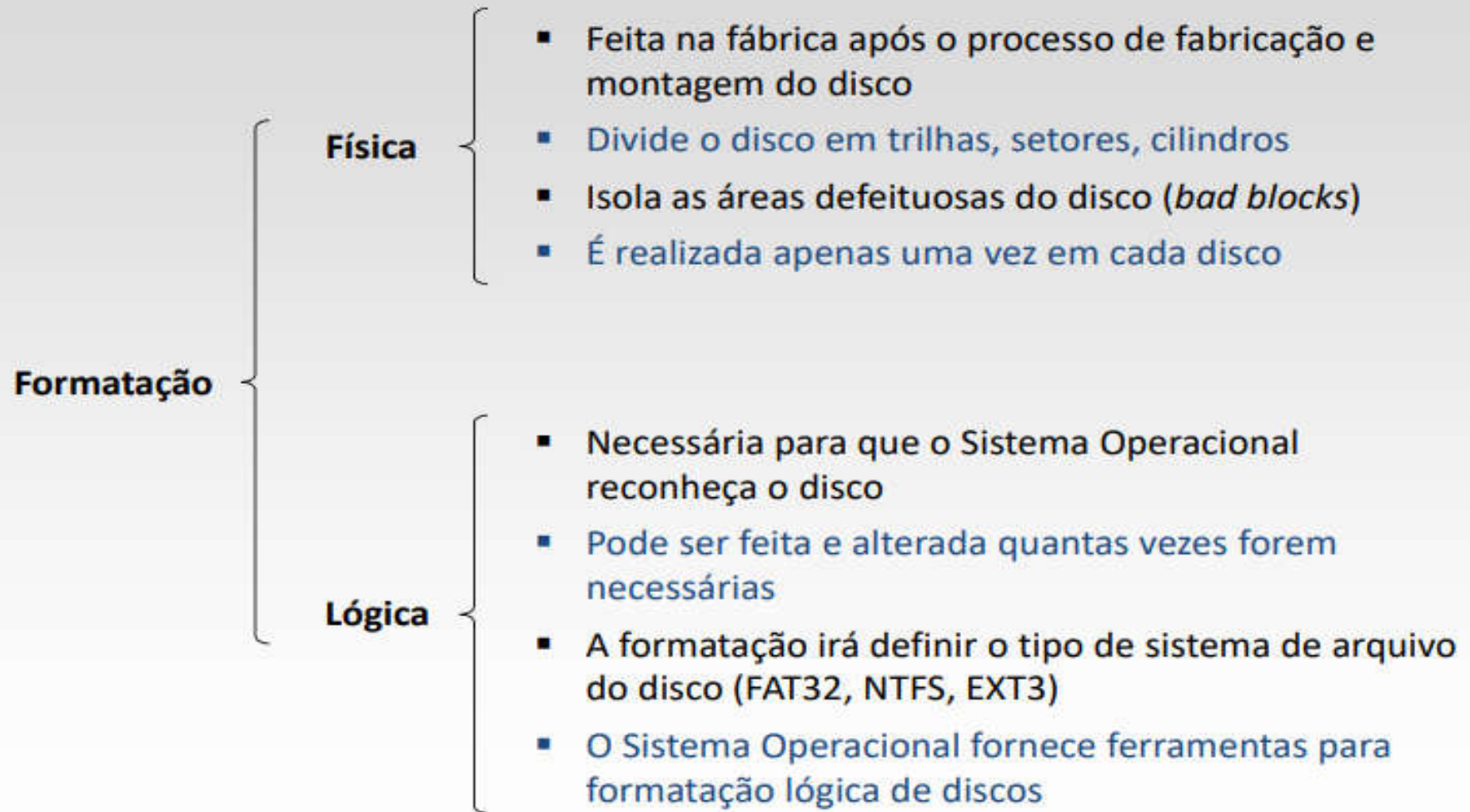
## Memória Secundária, Partições e Formatação



- A memória secundária possui muitos bits de armazenamento, mas que são inúteis até que uma tabela de alocação seja criada.
- Na tabela de alocação são especificados como os bits estão organizados (qual o sistema de arquivo utilizado).
- Pode-se dividir uma memória secundária (unidade física) em vários pedaços. chamados Partições (unidade lógica), também conhecidas como Volume ou Drive.
- Cada Partição pode ter um sistema de arquivos diferente.
- Formatar uma partição significa criar estruturas na memória secundária que permitam gravar os dados de maneira organizada e permitir acessá-los em outro momento.

# Sistema de Arquivos

## ➤ Formatação Física e Formatação Lógica



# Sistema de Arquivos

## Diretórios e Arquivos

Um diretório é uma estrutura utilizada para organizar arquivos e/ou outros diretórios

*Um diretório é um link que faz referência a um conjunto de arquivos e/ou um conjunto de diretórios*

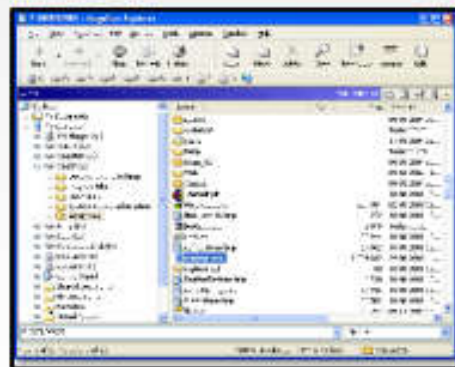
Os diretórios utilizam a estrutura de dados árvore para se organizarem

*O processo de montagem cria a árvore de diretórios.*

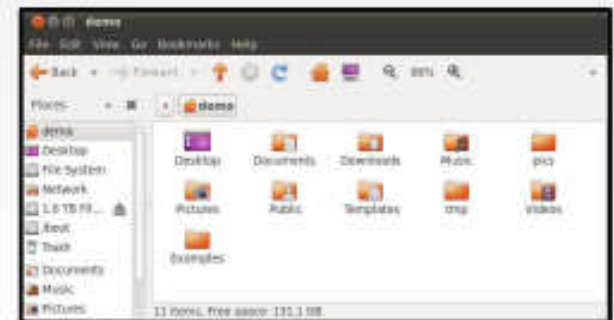
Em Sistemas Operacionais com interface gráfica o objeto visual que representa um diretório é uma pasta.



Comando dir em um Command Prompt



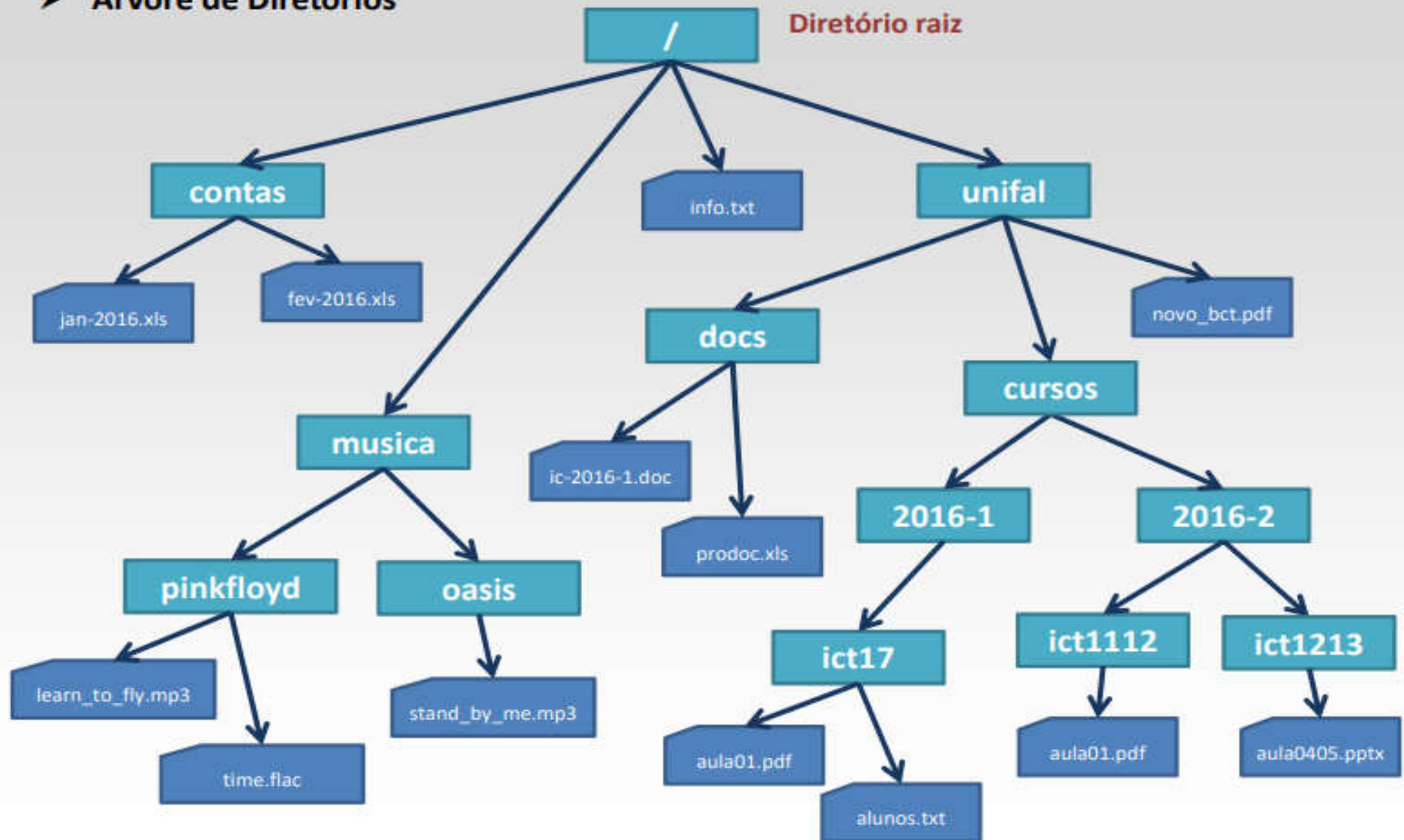
Windows Explorer



Ubuntu File Explorer

# Sistema de Arquivos

## ➤ Árvore de Diretórios



# Sistema de Arquivos

## Tipos de Arquivos

- Um arquivo é um conjunto de bits que armazena uma determinada quantidade de dados que ao ser lido/interpretado por um programa.
- A identificação de um arquivo é realizada por um nome e uma extensão.
- A extensão de um arquivo identifica o tipo deste arquivo.

Extensão	Tipo de arquivo
.bak	Arquivo de backup (cópia de segurança)
.doc	Arquivo documento no formato MS-Word
.gif	Arquivo de imagem no formato <i>Graphics Interchange Format</i>
.txt	Arquivo de texto geral
.pdf	Arquivo documento no formato <i>Portable Document Format</i>
.mpg	Arquivo de áudio ou vídeo (compactação com perdas MPEG-1 /MPEG-2)
.zip	Arquivo compactado (compactação sem perdas)
.html	Documento no formato <i>HyperText Markup Language</i>
.c	Arquivo com programa escrito na linguagem C
.java	Arquivo com programa escrito na linguagem Java

# Usuários

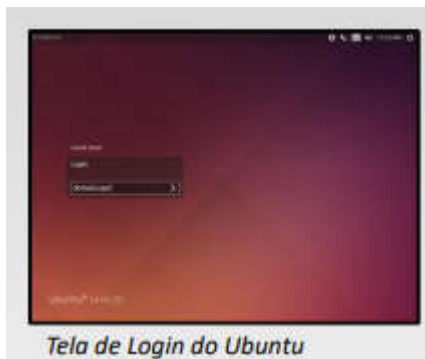
Um usuário é um agente externo ao sistema que faz uso do computador e seus periféricos para realizar um determinado trabalho.

O Sistema Operacional é responsável por identificar os usuários, permitindo ou proibindo o acesso ao sistema e estabelecendo restrições e concessões de acesso aos recursos do computador.

## Autenticação de Usuários

O acesso ao Sistema Operacional é realizado através do procedimento de login, que ao mesmo tempo identifica o usuário (através de um nome de identificação de usuário) e confirma a sua autenticidade (através de uma senha).

Quando o login de um usuário é realizado, cria-se uma sessão na qual ele pode realizar suas tarefas de acordo com as permissões concedidas para o usuário.





# Usuários

## ➤ Classes de Usuários

- Todo usuário que acessa o **Sistema Operacional** pertence a uma determinada **classe** que **permite** ou **restringe** suas **ações dentro do sistema**
- Cada **Sistema Operacional** possui suas próprias classes

## ➤ Classes de Usuários do Linux

Classe	Descrição
<b>root</b>	Permissão total (instalar, desinstalar e executar programas, controlar a execução de processos, criar novos usuários). É a classe de usuário destinada à administração do S.O.
<b>usuário do sistema</b>	São “usuários virtuais” criados junto com os programas para que esses possam executar ações dentro do S.O.
<b>usuário padrão</b>	Permissão de executar programas e salvar arquivos no seu diretório <b>home</b> . As permissões são concedidas pelo administrador que criou o usuário, mas geralmente são limitadas. Possuem controle somente sobre seus próprios processos.



# Usuários

## ➤ Acesso aos Arquivos

- Os **arquivos/diretórios** pertencentes ao sistema ou criados por usuários possuem **atributos** que permitem ou restringem determinadas ações:

**LER (r)**

**ESCREVER (w)**

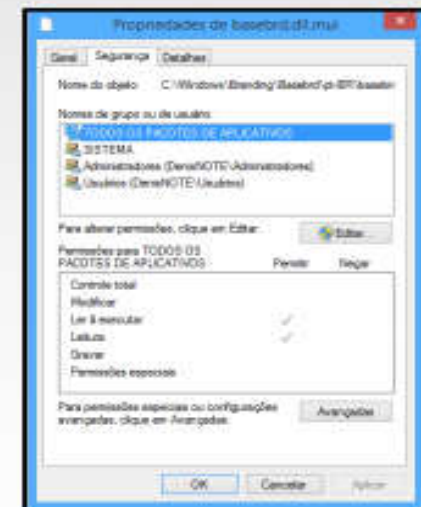
**EXECUTAR (x)**

- O usuário “proprietário” do arquivo pode modificar esses atributos a fim de permitir que outros usuários tenham acesso à esses **arquivos/diretórios**

## ➤ Grupos de Usuários

- São formados por usuários que possuem acesso a determinados componentes do sistema de acordo com a determinação do administrador que criou o grupo
- Todo usuário deve pertencer a pelo menos um grupo, mas pode fazer parte de vários grupos

**Exemplo:** Quando um arquivo é criado, este pertence ao usuário que o criou, mas o usuário pode dar permissão para os usuários de um determinado grupo (a que pertence) para ler, escrever e/ou executar esse arquivo



*Janela de propriedades de um arquivo do S. O. Windows.  
Segurança: Grupos e Permissões*