

# Fundamentos de Sistemas Operacionais

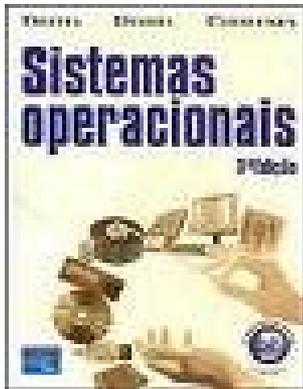
---

Prof. Érico José Ferreira

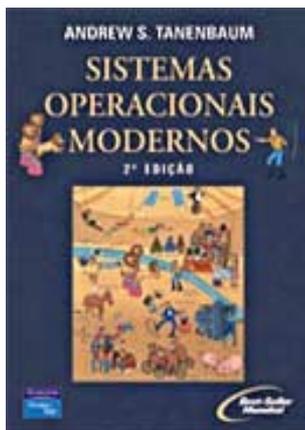
[ericonet@ericonet.com.br](mailto:ericonet@ericonet.com.br)

# Fundamentos de Sistemas Operacionais

## Referências Bibliográficas



- DEITEL,H.; DEITEL,P. & CHOFFNES. **Sistemas Operacionais**. 3.ed. São Paulo: Pearson, 2005



- TANENBAUM, Andrew S.; **Sistemas Operacionais Modernos**. 2.ed. São Paulo: Pearson, 2003.

# Fundamentos de Sistemas Operacionais

## Linux

- Originalmente desenvolvido como um passatempo de Linus Torvalds. Ele queria um sistema operacional que fosse semelhante a um Unix, com todas as suas funcionalidades e, ainda, que pudesse utilizá-lo num PC.
- A partir dessa idéia, Linus começou a trabalhar nesse que seria o futuro kernel do sistema operacional que hoje é chamado de *Linux*. Isso tudo aconteceu em meados de 1991, quando Linus cursava a faculdade de Computação na Finlândia.

# Fundamentos de Sistemas Operacionais

## Linux

- Em 5 de outubro de 1991 a seguinte mensagem circulou na usenet:
- “ *...Como eu mencionei há um mês, estou trabalhando em uma versão free de um sistema semelhante ao Minix para computadores AT-386. Ele já alcançou o estágio de ser usável (embora possa não ser, dependendo do que você quer fazer), e pretendo distribuir o código fonte. É apenas a versão 0.02... mas já consegui rodar nele o bash, gcc, gnu-make, gnu-sed, compress, etc.*”

# Fundamentos de Sistemas Operacionais

## Linux

- Esta mensagem era assinada por Linus Torvalds, e ninguém adivinharia que ela estaria marcando o início de um movimento que, menos de dez anos depois, já tem mais de trinta milhões de seguidores.
- Assim surgiu o que seria o primeiro kernel utilizável do Linux.
- O kernel é o núcleo do sistema operacional, é a parte que controla diretamente o hardware da máquina. Quando se fala de Linux, está se referindo somente ao kernel do sistema.
- Tudo que existe *ao redor* do kernel são aplicativos que compõem uma distribuição do Linux.

# Fundamentos de Sistemas Operacionais

## Linux- Distribuições

---

- São compostas pelo núcleo do sistema operacional, um conjunto de softwares aplicativos e uma interface de instalação
- Existem várias distribuições (sabores)
  - Diferenciam-se pelo conjunto de aplicativos, interface de instalação e estrutura de manutenção
  - Questão de gosto pessoal

# Fundamentos de Sistemas Operacionais

## Linux- Distribuições

- Existe uma base comum grande de softwares aplicativos (editores gráficos, planilhas, bancos de dados, ambientes de programação, formatação de documentos, firewalls, etc.)
  - Desenvolvidos pela GNU (GNU is not Unix) da Free Software Foundation
- Em uma distribuição o que se paga não é o Linux, mas uma série de serviços disponibilizados
  - Cdrom, manuais de instalação, *hot-line*, software proprietário, etc.

# Fundamentos de Sistemas Operacionais

## Linux - Distribuições

- Algumas distribuições Linux são maiores que outras, dependendo da quantidade de aplicativos e a finalidade a que se propõem. Existem desde distribuições que cabem num disquete de 1.44Mb até distribuições que ocupam vários CDs.
- Cada uma delas têm seu público-alvo e finalidades específicas.
- As minidistribuições têm como objetivo desde a recuperação de um sistema danificado até o monitoramento de uma rede de computadores.
- Algumas distribuições incluem ferramentas de configuração que facilitam a vida do administrador do sistema.

# Fundamentos de Sistemas Operacionais

## Linux - Distribuições

- Debian
- <http://www.debian.org> - Distribuição desenvolvida e atualizada através do esforço de voluntários espalhados ao redor do mundo, seguindo o estilo de desenvolvimento GNU/Linux. Por este motivo, foi adotada como a distribuição oficial do projeto *GNU*. Possui suporte a língua Portuguesa, é a única que tem suporte a 10 arquiteturas diferentes (i386, Alpha, Sparc, PowerPc, Macintosh, Arm, etc.) e aproximadamente 15 sub-arquiteturas. A instalação da distribuição pode ser feita tanto através de Disquetes, CD-ROM, Tftp, Ftp, NFS ou através da combinação de vários destes em cada etapa de instalação.

# Fundamentos de Sistemas Operacionais

## Linux - Distribuições

- Acompanha cerca de 15.000 programas distribuídos em forma de pacotes divididos em 4 CDs binários e 2 de código fonte. Cada um destes programas são mantidos e testados pela pessoa responsável por seu empacotamento. Os pacotes são divididos em diretórios de acordo com sua categoria e gerenciados através de um avançado sistema de gerenciamento de pacotes (o dpkg) facilitando a instalação e atualização de pacotes. Possui tanto ferramentas para administração de redes e servidores quanto para desktops, estações multimídia, jogos, desenvolvimento, web, etc.

# Fundamentos de Sistemas Operacionais

## Linux - Distribuições

- A atualização da distribuição ou de pacotes individuais pode ser feita facilmente através de 2 comandos, não requerendo adquirir um novo CD para usar a última versão da distribuição. É a única distribuição não comercial onde todos podem contribuir com seu conhecimento para o seu desenvolvimento. Para gerenciar os voluntários, conta com centenas de listas de discussão envolvendo determinados desenvolvedores das mais diversas partes do mundo.

# Fundamentos de Sistemas Operacionais

## Linux - Distribuições

- São feitos extensivos testes antes do lançamento de cada versão para atingir um alto grau de confiabilidade. As falhas encontradas nos pacotes podem ser relatados através de um *sistema de tratamento de falhas* que encaminha a falha encontrada diretamente ao responsável para avaliação e correção. Qualquer um pode receber a lista de falhas ou sugestões sobre a distribuição cadastrando-se em uma das lista de discussão que tratam especificamente da solução de falhas encontradas na distribuição (disponível na página principal da distribuição).

# Fundamentos de Sistemas Operacionais

## Linux - Distribuições

- Os pacotes podem ser instalados através de Tarefas contendo seleções de pacotes de acordo com a utilização do computador (servidor Web, desenvolvimento, TeX, jogos, desktop, etc.), *Perfis* contendo seleções de pacotes de acordo com o tipo de usuário (programador, operador, etc.), ou através de uma seleção individual de pacotes, garantindo que somente os pacotes selecionados serão instalados fazendo uma instalação enxuta.

# Fundamentos de Sistemas Operacionais

## Linux - Distribuições

- Existe um time de desenvolvedores com a tarefa específica de monitorar atualizações de segurança em serviços (apache, sendmail, e todos os outros pacotes) que possam comprometer o servidor, deixando-o vulnerável a ataques. Assim que uma falha é descoberta, é enviado uma alerta (DSA - Debian Security Alert) e disponibilizada uma atualização para correção das diversas versões da Debian. Isto é geralmente feito em menos de 48 horas desde a descoberta da falha até a divulgação da correção. Como quase todas as falhas são descobertas nos programas, este método também pode ser usado por administradores de outras distribuições para manterem seu sistema seguro e atualizado.

# Fundamentos de Sistemas Operacionais

## Linux - Distribuições

- O suporte ao usuário e desenvolvimento da distribuição são feitos através de listas de discussões e canais IRC. Existem uma lista de consultores habilitados a dar suporte e assistência a sistemas Debian ao redor do mundo, na área consultores do site principal da distribuição.
- <ftp://ftp.debian.org/> - Endereço ftp para download.

# Fundamentos de Sistemas Operacionais

## Linux - Distribuições

- Conectiva
  - <http://www.conectiva.com.br/> - É um sistema tipicamente para pc e notebooks, com muitos pacotes de escritório e desenho e etc. É necessário um conhecimento razoável do hardware que você possui, pois ele **tenta** detectar sua placa de vídeo e rede, monitor e teclado, mas nem sempre a escolha é acertada. A versão 10.0 saiu graficamente maravilhosa, mas com 3 cds. Existe um gerenciador gráfico de pacotes, o Synaptic, que é muito bom, principalmente para quem está começando.
  - A instalação é muito simples e intuitiva, com todo o gerenciamento centralizado. Além de ter a maioria dos manuais online e traduzidos para português.
  - <ftp://ftp.conectiva.com.br/> - ftp da distribuição Conectiva.

# Fundamentos de Sistemas Operacionais

## Linux - Distribuições

- Slackware
  - <http://www.slackware.com/> - Distribuição desenvolvida por Patrick Volkerding, desenvolvida para alcançar facilidade de uso e estabilidade como prioridades principais. Foi a primeira distribuição a ser lançada no mundo e costuma trazer o que há de mais novo enquanto mantém uma certa tradição, provendo simplicidade, facilidade de uso e com isso flexibilidade e poder. Desde a primeira versão lançada em Abril de 1993, o Projeto Slackware Linux tem buscado produzir a distribuição Linux mais UNIX-like, ou seja, mais parecida com UNIX. O Slackware segue os padrões Linux como o Linux File System Standard, que é um padrão de organização de diretórios e arquivos para as distribuições.
  - Enquanto as pessoas diziam que a Red Hat era a melhor distribuição para o usuário iniciante, o Slackware é o melhor para o usuário mais "velho", ou seja programadores, administradores, etc.
  - <ftp://ftp.slackwarebrasil.org/linux/slackware/> - ftp da distribuição Slackware.

# Fundamentos de Sistemas Operacionais

## Linux - Distribuições

- SuSE
- <http://www.suse.com/> - Distribuição comercial Alemã com a coordenação sendo feita através dos processos administrativos dos desenvolvedores e de seu braço norte-americano. O foco da Suse é o usuário com conhecimento técnico no Linux (programador, administrador de rede, etc.) e não o usuário iniciante no Linux (até a versão 6.2). A distribuição possui suporte ao idioma e teclado Português, mas não inclui (até a versão 6.2) a documentação em Português. Eis a lista de idiomas suportados pela distribuição: English, Deutsch, Français, Italiano, Espanhol, Português, Português Brasileiro, Polski, Cesky, Romanian, Slovensky, Indonésia.
- Possui suporte as arquiteturas Intel x86 e Alpha. Sua instalação pode ser feita via CD-ROM ou CD-DVD (é a primeira distribuição com instalação através de DVD).

# Fundamentos de Sistemas Operacionais

## Linux - Distribuições

- Uma média de 1500 programas acompanham a versão 6.3 distribuídos em 6 CD-ROMs. O sistema de gerenciamento de pacotes é o RPM padronizado. A seleção de pacotes durante a instalação pode ser feita através da seleção do perfil de máquina (developer, estação kde, gráficos, estação gnome, servidor de rede, etc.) ou através da seleção individual de pacotes.
- A atualização da distribuição pode ser feita através do CD-ROM de uma nova versão ou baixando pacotes de <ftp://ftp.suse.com/>. Usuários registrados ganham direito a suporte de instalação via e-mail. A base de dados de suporte também é excelente e está disponível na web para qualquer usuário independente de registro.
- <ftp://ftp.suse.com/> - ftp da distribuição SuSE.

# Fundamentos de Sistemas Operacionais

## Linux - Distribuições

- Red Hat Enterprise Linux
  - <http://www.redhat.com/> - Distribuição comercial suportada pela Red Hat e voltada a servidores de grandes e medias empresas. Também conta com uma certificação chamada RHCE específica desta distro. Ela não está disponível para download, apenas vendida a custos a partir de 179 dólares (a versão workstation) até 1499 dólares (advanced server).

# Fundamentos de Sistemas Operacionais

## Linux - Distribuições

- Fedora
- <http://fedora.redhat.com/> - O Fedora Linux é a distribuição de desenvolvimento aberto patrocinada pela RedHat e pela comunidade, originada em 2002 e baseada em versão da antiga linha de produtos RedHat Linux, a distribuição mais utilizada do mundo. Esta distribuição não é suportada pela Red Hat como distribuição oficial (ela suporta apenas a linha Red Hat Enterprise Linux), devendo obter suporte através da comunidade ou outros meios.
- A distribuição Fedora dá prioridade ao uso do computador como estação de trabalho. Além de contar com uma ampla gama de ferramentas de escritório possui funções de servidor e aplicativos para produtividade e desenvolvimento de softwares. Considerado um dos sistemas mais fáceis de instalar e utilizar, inclui tradução para português do Brasil e suporte às plataformas Intel e 64 bits.

# Fundamentos de Sistemas Operacionais

## Linux - Distribuições

- Por basear-se no RedHat, o Fedora conta com o up2date, um software para manter o sistema atualizado e utiliza pacotes de programas no formato RPM, um dos mais comuns. Por outro lado, não possui suporte a MP3, Video Players ou NTFS (Discos do Windows) em virtude de problemas legais sendo necessário o download de alguns plugins para a utilização destas funções.
- O Fedora não é distribuído oficialmente através de mídias ou CDs, se você quiser obtê-lo terá de procurar distribuidores independentes ou fazer o download dos 4 CDs através do site oficial.
- <http://download.fedora.redhat.com/pub/fedora/linux/core/2/i386/iso/> - Download da distribuição Fedora.

# Fundamentos de Sistemas Operacionais

## Linux - Distribuições

- Mandrake
  - <http://www.mandrakelinux.com/> - Uma distribuição francesa que se instala praticamente sozinha. Boa auto-deteccção de periféricos, inclusive web-cams. <ftp://mandrake.mirrors.pair.com/Mandrakelinux/official/iso/move/2.0/i586/> - Download da distribuição.
  - Recentemente se fundiu a Conectiva, formando a Mandriva.

# Fundamentos de Sistemas Operacionais

## Linux - Distribuições

- Kurumin
  - <http://guiadohardware.net/kurumin/index.php/> - Uma distribuição baseada em Debian que roda diretamente a partir do CD, sendo ideal para quem deseja testar uma distribuição Linux. Caso goste, pode ser instalada diretamente no disco rígido. Distribuída a partir do CD, é maravilhosa e suporta boa quantidade de hardwares disponíveis. A versão instalada possui suporte a maioria dos winmodems mais encontrados no Brasil.
  - <http://fisica.ufpr.br/kurumin/> - Download da distribuição.

# Fundamentos de Sistemas Operacionais

## Linux - Distribuições

- Ubuntu
  - <http://www.ubuntu-br.org/> - Uma distribuição baseada em Debian que roda diretamente a partir do CD, ou pode ser instalada em seu computador. Ideal para desktops. Tem versões com Gnome e KDE. É maravilhosa e tem excelente detecção de hardware. Vem com a maioria dos aplicativos típicos de usuários finais, como pacote de automação de escritório, ferramentas para tratamento de imagens e vídeos, gravação de CD e DVD, acesso a internet, correio, IM e etc. Sofre constantes atualizações que podem ser instaladas automaticamente.
  - <http://www.ubuntu-br.org/download> - Download da distribuição.

# Fundamentos de Sistemas Operacionais

## Sistema Operacional Linux

- O *Sistema Operacional* é o conjunto de programas que fazem a interface do usuário e seus programas com o computador. Ele é responsável pelo gerenciamento de recursos e periféricos (como memória, discos, arquivos, impressoras, CD-ROMs, etc.), interpretação de mensagens e a execução de programas.
- No Linux o Kernel mais o conjunto de ferramentas GNU compõem o Sistema Operacional. O kernel (que é a base principal de um sistema operacional), poderá ser construído de acordo com a configuração do seu computador e dos periféricos que possui.

# Fundamentos de Sistemas Operacionais

## Sistema Operacional Linux

- Sistema de numeração de versões é baseado em três números:
  - Versão principal: três até hoje – 0, 1 e 2
  - Indicativo se o kernel é estável(par) ou em desenvolvimento(impar)
  - Release
- Para saber sua versão utilize o seguinte comando:
  - `uname -r`

# Fundamentos de Sistemas Operacionais

## Sistema Operacional Linux - Características

---

- É livre e desenvolvido voluntariamente por programadores experientes, hackers, e contribuidores espalhados ao redor do mundo que tem como objetivo a contribuição para a melhoria e crescimento deste sistema operacional.
- Muitos deles estavam cansados do excesso de propaganda (Marketing) e baixa qualidade de sistemas comerciais existentes.

# Fundamentos de Sistemas Operacionais

## Sistema Operacional Linux - Características

- Convive sem nenhum tipo de conflito com outros sistemas operacionais (com o DOS, Windows, OS/2) no mesmo computador.
- Multitarefa real
- Multiusuário
- Suporte a nomes extensos de arquivos e diretórios (255 caracteres)
- Conectividade com outros tipos de plataformas como *Apple, Sun, Macintosh, Sparc, Alpha, PowerPc, ARM, Unix, Windows, DOS, etc.*
- Proteção entre processos executados na memória RAM
- Suporte a mais de 63 terminais virtuais (consoles)

# Fundamentos de Sistemas Operacionais

## Sistema Operacional Linux - Características

- Modularização - O GNU/Linux somente carrega para a memória o que é usado durante o processamento, liberando totalmente a memória assim que o programa/dispositivo é finalizado
- Devido a modularização, os drivers dos periféricos e recursos do sistema podem ser carregados e removidos completamente da memória RAM a qualquer momento. Os drivers (módulos) ocupam pouco espaço quando carregados na memória RAM (cerca de 6Kb para a Placa de rede, por exemplo)
- Não há a necessidade de se reiniciar o sistema após modificar a configuração de qualquer periférico ou parâmetros de rede. Somente é necessário reiniciar o sistema no caso de uma instalação interna de um novo periférico, falha em algum hardware (queima do processador, placa mãe, etc.).

# Fundamentos de Sistemas Operacionais

## Sistema Operacional Linux - Características

- Não precisa de um processador potente para funcionar. O sistema roda bem em computadores 386Sx 25 com 4MB de memória RAM (sem rodar o sistema gráfico X, que é recomendado 8MB de RAM). Já pensou no seu desempenho em um 486 ou Pentium ;-)
- O crescimento e novas versões do sistema não provocam lentidão, pelo contrário, a cada nova versão os desenvolvedores procuram buscar maior compatibilidade, acrescentar recursos úteis e melhor desempenho do sistema (como o que aconteceu na passagem do kernel 2.0.x para 2.2.x).
- Não é requerida uma licença para seu uso. O GNU/Linux é licenciado de acordo com os termos da GPL.

# Fundamentos de Sistemas Operacionais

## Sistema Operacional Linux - Características

- Acessa corretamente discos formatados pelo DOS, Windows, Novell, OS/2, NTFS, SunOS, Amiga, Atari, Mac, etc.
- Utiliza permissões de acesso a arquivos, diretórios e programas em execução na memória RAM.
- O LINUX NÃO É VULNERÁVEL A VÍRUS! Devido a separação de privilégios entre processos e respeitadas as recomendações padrão de política de segurança e uso de contas privilegiadas (como a de root, como veremos adiante), programas como vírus tornam-se inúteis pois tem sua ação limitada pelas restrições de acesso do sistema de arquivos e execução.

# Fundamentos de Sistemas Operacionais

## Sistema Operacional Linux - Características

- Frequentemente são criados exploits que tentam se aproveitar de falhas existentes em sistemas desatualizados e usá-las para danificar o sistema. *Erroneamente* este tipo de ataque é classificado como vírus por pessoas mal informadas e são resolvidas com sistemas bem mantidos. Em geral, usando uma boa distribuição que tenha um bom sistema de atualização resolve em 99.9% os problemas com exploits. Qualquer programa (nocivo ou não) poderá alterar partes do sistema que possui permissões.

# Fundamentos de Sistemas Operacionais

## Sistema Operacional Linux - Características

---

- Rede TCP/IP mais rápida que no Windows e tem sua pilha constantemente melhorada. O GNU/Linux tem suporte nativo a redes TCP/IP e não depende de uma camada intermediária como o WinSock. Em acessos via modem a Internet, a velocidade de transmissão é 10% maior.
- Jogadores do Quake ou qualquer outro tipo de jogo via Internet preferem o GNU/Linux por causa da maior velocidade do Jogo em rede. É fácil rodar um servidor Quake em seu computador e assim jogar contra vários adversários via Internet.

# Fundamentos de Sistemas Operacionais

## Sistema Operacional Linux - Características

- Roda aplicações *DOS* através do DOSEMU, QEMU, BOCHS. Para se ter uma idéia, é possível dar o boot em um sistema *DOS* qualquer dentro dele e ao mesmo tempo usar a multitarefa deste sistema.
- Roda aplicações *Windows* através do WINE.
- Suporte a dispositivos infravermelho.
- Suporte a rede via rádio amador.
- Suporte a dispositivos Plug-and-Play.
- Suporte a dispositivos USB.
- Suporte a Firewire.
- Dispositivos Wireless.

# Fundamentos de Sistemas Operacionais

## Sistema Operacional Linux - Características

---

- Vários tipos de firewalls de alta qualidade e com grande poder de segurança de graça.
- Roteamento estático e dinâmico de pacotes.
- Ponte entre Redes.
- Proxy Tradicional e Transparente.
- Possui recursos para atender a mais de um endereço IP na mesma placa de rede, sendo muito útil para situações de manutenção em servidores de redes ou para a emulação de "mais computadores" virtualmente.
  - O servidor WEB e FTP podem estar localizados no mesmo computador, mas o usuário que se conecta tem a impressão que a rede possui servidores diferentes.

# Fundamentos de Sistemas Operacionais

## Sistema Operacional Linux - Características

---

- O sistema de arquivos usado pelo GNU/Linux (Ext2) organiza os arquivos de forma inteligente evitando a fragmentação e fazendo-o um poderoso sistema para aplicações multi-usuárias exigentes e gravações intensivas.
- Permite a montagem de um servidor Web, E-mail, News, etc. com um baixo custo e alta performance. O melhor servidor Web do mercado, o Apache, é distribuído gratuitamente junto com a maioria das distribuições Linux. O mesmo acontece com o Sendmail.

# Fundamentos de Sistemas Operacionais

## Sistema Operacional Linux - Características

- Por ser um sistema operacional de código aberto, você pode ver o que o código fonte (instruções digitadas pelo programador) faz e adaptá-lo as suas necessidades ou de sua empresa. Esta característica é uma segurança a mais para empresas sérias e outros que não querem ter seus dados roubados (você não sabe o que um sistema sem código fonte faz na realidade enquanto esta processando o programa).
- Suporte a diversos dispositivos e periféricos disponíveis no mercado, tanto os novos como obsoletos.
- Pode ser executado em 10 arquiteturas diferentes (Intel, Macintosh, Alpha, Arm, etc.).
- Consultores técnicos especializados no suporte ao sistema espalhados por todo o mundo.

# Fundamentos de Sistemas Operacionais

## Processos no Linux

- Nos sistemas operacionais, um processo é a estrutura responsável pela manutenção de todas as informações necessárias a execução de um programa (conteúdo de registradores, espaço de memória, etc.).
- É o processo que utiliza os recursos do computador - processador, memória, etc. - para a realização das tarefas para as quais a máquina é destinada.

# Fundamentos de Sistemas Operacionais

## Processos no Linux

---

- Diferenciação entre o programa e sua execução
- Programa
  - Entidade estática e permanente
    - Sequência de instruções
    - Passivo sob o ponto de vista do sistema operacional
- Processo
  - Entidade efêmera e dinâmica
    - Altera seu estado a medida que avança sua execução
  - Composto por programa(código), dados e contexto(valores)

# Fundamentos de Sistemas Operacionais

## Processos no Linux

- Abstração que representa um programa em execução
- Um programa pode ter várias instâncias em execução  
→ diferentes processos
- Mesmo programa(código) porém dados e momentos de execução(contexto) diferentes
- Forma pela qual o sistema operacional “enxerga” um programa e possibilita sua execução
- Processos executam:
  - Programas de usuários
  - Programas do próprio sistema operacional(daemons)

# Fundamentos de Sistemas Operacionais

## Processos no Linux

---

- Ciclos de vida de um processo:
  - Criação
  - Execução
  - Término

# Fundamentos de Sistemas Operacionais

## Processos no Linux

- Ciclos de vida de um processo – Criação:
  - Momento da execução
  - Chamadas de sistemas
    - `fork()`, `fork_exec()`, `spawn()`, etc.
- Podem ser associados a uma sessão de trabalho:
  - Ex: login de usuário: login + senha → shell (processo)
- Identificado por um número único

## Processos no Linux

- Ciclos de vida de um processo – Execução:
  - Processos apresentam dois ciclos básicos de operação:
    - Ciclo de processador: tempo que ocupa a CPU
    - Ciclo de entrada e saída: tempo em espera pela conclusão de um evento (p.e. abertura de arquivo)
  - Primeiro ciclo é sempre de processador
  - Troca de ciclos por:
    - CPU -> E/S: chamada de sistema
    - E/S -> CPU: ocorrência de evento (interrupção)

## Processos no Linux

- Ciclos de vida de um processo – Execução:
  - Processos com ciclo de processador muito maior que o ciclo de E/S são chamados *CPU bound*
  - Processos com ciclo de E/S muito maior que o ciclo de processador são chamados *I/O bound*
  - Sem quantificação exata
  - Ideal para benefícios de escalonamento, misturar processos *CPU bound* e *I/O bound*

- Ciclos de vida de um processo – Término:
  - Final normal de execução
  - Por erros: proteção, aritméticos, E/S, falta de memória, excede tempo de limite, tentativa de executar instruções inválidas, etc.
  - Intervenção de outros processos (kill)
  - *Log off* de usuários

## Processos no Linux

- O sistema operacional lida com uma infinidade de processos e, por isso, é necessário ter meios que permitam controlá-los.
- Para isso, os processos contam com um conjunto de características, dentre as quais:
  - Proprietário do processo;
  - Estado do processo (em espera, em execução, etc.);
  - Prioridade de execução;
  - Recursos de memória.

# Fundamentos de Sistemas Operacionais

## Processos no Linux

---

- O trabalho de gerenciamento de processos precisa contar com essas informações e com outras de igual importância para que as tarefas sejam executadas da maneira mais eficiente.
- Um dos meios usados para isso é atribuir a cada processo um PID.

## Processos no Linux

---

- Um PID (**P**rocess **I**dentifier) é um número de identificação que o sistema dá a cada processo.
- Para cada novo processo, um novo número deve ser atribuído, ou seja, não se pode ter um único PID para dois ou mais processos ao mesmo tempo.

## Processos no Linux

- Os sistemas baseados em Unix precisam que um processo já existente se duplique para que a cópia possa ser atribuída a uma tarefa nova.
- Quando isso ocorre, o processo "copiado" recebe o nome de "processo pai", enquanto que o novo é denominado "processo filho".
- É nesse ponto que o PPID (**P**arent **P**rocess **I**dentifier) passa a ser usado: o PPID de um processo nada mais é do que o PID de seu processo pai.

## Processos no Linux

---

- Como já mencionado, cada processo precisa de um proprietário, um usuário que seja considerado seu dono.
- A partir daí, o sistema saberá, através das permissões fornecidas pelo proprietário, quem pode e quem não pode executar o processo em questão.
- Para lidar com os donos, o sistema usa os números UID e GID.

## Processos no Linux

- O Linux gerencia os usuários e os grupos através de números conhecidos como UID (**U**ser **I**dentifier) e GID (**G**roup **I**dentifier).
- UID são números de usuários e GID são números de grupos.
- Os nomes dos usuários e dos grupos servem apenas para facilitar o uso humano do computador.

## Processos no Linux

---

- Cada usuário precisa pertencer a um ou mais grupos.
- Como cada processo (e cada arquivo) pertence a um usuário, logo, esse processo pertence ao grupo de seu proprietário.
- Assim sendo, cada processo está associado a um UID e a um GID.

- Os números UID e GID variam de 0 a 65536.
- Dependendo do sistema, o valor limite pode ser maior.
- No caso do usuário root, esses valores são sempre 0 (zero).
- Assim, para fazer com que um usuário tenha os mesmos privilégios que o root, é necessário que seu GID seja 0.

## Processos no Linux

- Quando um processo é criado, isso não significa que ele será imediatamente executado. Além disso, determinados processos podem ser temporariamente paralisados para que o processador possa executar um processo prioritário. Isso quer dizer que os processos, em certos momentos, podem estar em situações de execução diferentes.

# Fundamentos de Sistemas Operacionais

## Processos no Linux

- O Linux trabalha, essencialmente, com quatro tipos de situação, isto é, estados:
  - **Executável:** o processo pode ser executado imediatamente;
  - **Dormente:** o processo precisa aguardar alguma coisa para ser executado. Só depois dessa "coisa" acontecer é que ele passa para o estado executável;
  - **Zumbi:** o processo é considerado "morto", mas, por alguma razão, ainda existe;
  - **Parado:** o processo está "congelado", ou seja, não pode ser executado.

# Fundamentos de Sistemas Operacionais

## Uma palavra sobre *Threads*

- Uma *thread* é usualmente definida como um fluxo de controle no interior de um processo.
- Quando um novo processo é criado, o núcleo do Linux copia os atributos do processo corrente para o que está sendo criado.
- É o procedimento de `fork-exec()`.

# Fundamentos de Sistemas Operacionais

## Uma palavra sobre *Threads*

- O Linux, entretanto, prevê uma segunda forma de criação de processos: a clonagem.
- Um processo clone compartilha os recursos (arquivos abertos, memória virtual, etc.) com o processo original.
- Quando dois ou mais processos compartilham as mesmas estruturas, eles atuam como se fossem diferentes *threads* no interior de um único processo.
- O Linux não diferencia a estrutura de dados de *threads* e de processos, e, por consequência, ambos são tratados indistintivamente por todos os mecanismos de gerência do núcleo.
- *Threads* e processos são tratados da mesma forma

# Fundamentos de Sistemas Operacionais

## Uma palavra sobre *Threads*

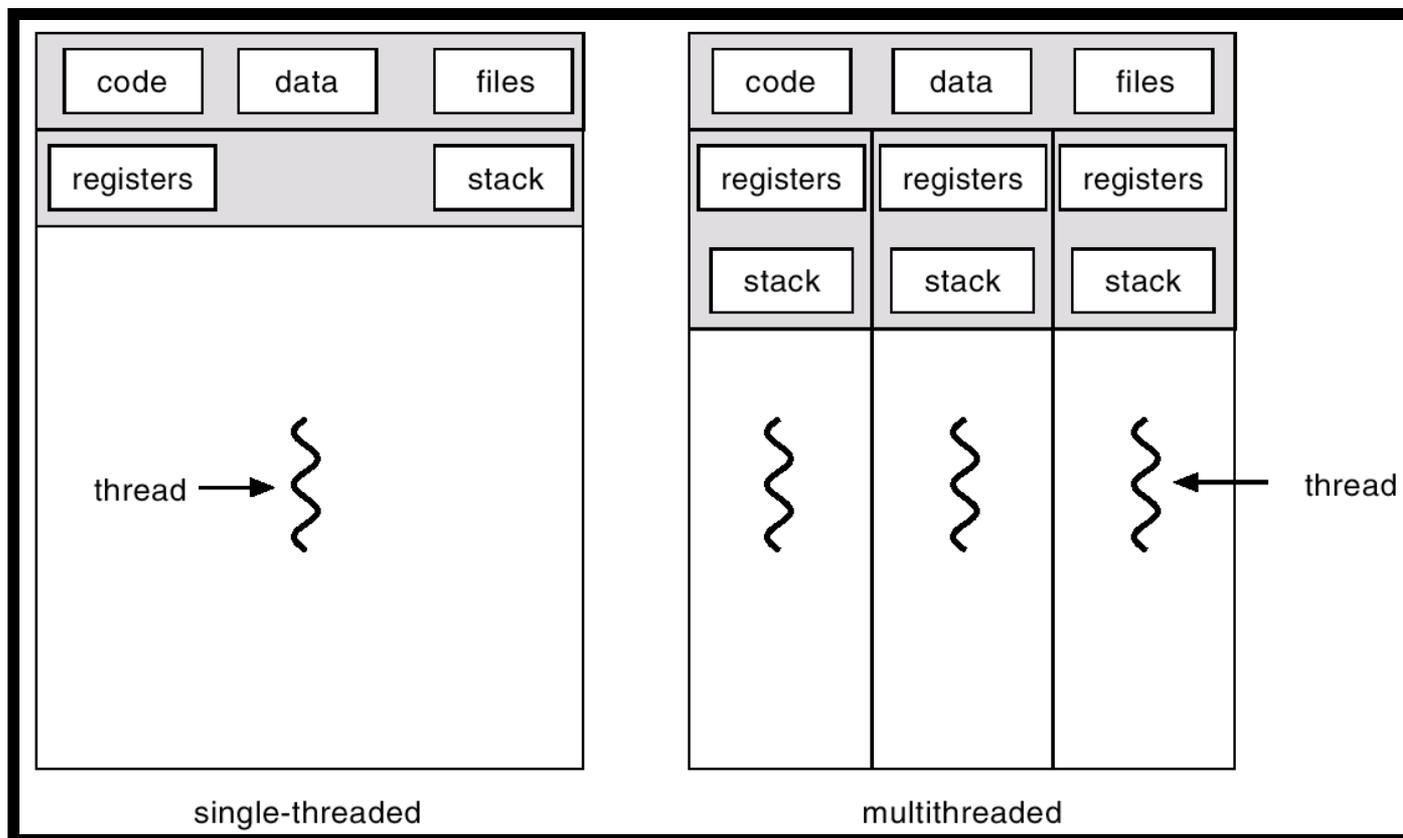
---

- Vantagens do uso de *threads*:
  - Tempo de resposta
  - Compartilhamento de recursos
  - Economia
  - Utilização da arquitetura de multiprocessadores

# Fundamentos de Sistemas Operacionais

## Uma palavra sobre *Threads*

- Processos com *thread* único e com *multithreads*



# Fundamentos de Sistemas Operacionais

## Escalonamento

---

- Conjunto de regras utilizado para determinar como, quando e qual processo deverá ser executado.
- Problema básico é como satisfazer simultaneamente objetivos conflitantes:
  - tempo de resposta rápido
  - bom *throughput* para processos *background*
  - evitar postergação indefinida
  - conciliar processos de alta prioridade com de baixa prioridade, etc.

# Fundamentos de Sistemas Operacionais

## Escalonamento

- Tradicionalmente, os processos são divididos em três grandes classes:
  - processos interativos
  - Processos *batch*
  - processos tempo real.
- Em cada classe, os processos podem ser ainda subdivididos em *I/O bound* ou *CPU bound* de acordo com a proporção de tempo que ficam esperando por operações de entrada e saída ou utilizando o processador.

## Escalonamento

- O escalonador do Linux não distingue processos interativos de processos *batch*, diferenciando-os apenas dos processos tempo real.
- Como todos os outros escalonadores UNIX, o escalonador Linux privilegia os processos *I/O bound* em relação aos *CPU bound* de forma a oferecer um melhor tempo de resposta às aplicações interativas.

## Escalonamento

- O escalonador do Linux é baseado em *time-sharing*, ou seja, o tempo do processador é dividido em fatias de tempo (*quantum*) as quais são alocadas aos processos.
- Se, durante a execução de um processo, o *quantum* é esgotado, um novo processo é selecionado para execução, provocando então uma troca de contexto.
- Esse procedimento é completamente transparente ao processo e baseia-se em interrupções de tempo.
- Esse comportamento confere ao Linux um escalonamento do tipo preemptivo.

## Escalonamento

- O algoritmo de escalonamento do Linux divide o tempo de processamento em épocas (*epochs*).
- Cada processo, no momento de sua criação, recebe um *quantum* calculado no início de uma época.
- Diferentes processos podem possuir diferentes valores de *quantum*.
- O valor do *quantum* corresponde à duração da época, e essa, por sua vez, é um múltiplo de 10 ms inferior a 100 ms.

# Fundamentos de Sistemas Operacionais

## Escalonamento

- Outra característica do escalonador Linux é a existência de prioridades dinâmicas.
- O escalonador do Linux monitora o comportamento de um processo e ajusta dinamicamente sua prioridade, visando a equalizar o uso do processador entre os processos.
- Processos que recentemente ocuparam o processador durante um período de tempo considerado “longo” têm sua prioridade reduzida.
- De forma análoga, aqueles que estão há muito tempo sem executar recebem um aumento na sua prioridade, sendo então beneficiados em novas operações de escalonamento.

# Fundamentos de Sistemas Operacionais

## Escalonamento

- Na realidade, o sistema Linux trabalha com dois tipos de prioridades: estática e dinâmica.
- As prioridades estáticas são utilizadas exclusivamente por processos de tempo real correspondendo a valores na faixa de 1-99.
- Nesse caso, a prioridade do processo tempo real é definida pelo usuário e não é modificada pelo escalonador.
- Somente usuários com privilégios especiais têm a capacidade de criar e definir processos tempo real.
- O esquema de prioridades dinâmicas é aplicado aos processos interativos e *batch*.
- Aqui, a prioridade é calculada, considerando-se a prioridade base do processo e a quantidade de tempo restante em seu *quantum*.

## Escalonamento

- O escalonador do Linux executa os processos de prioridade dinâmica apenas quando não há processos de tempo real.
- Em outros termos, os processos de prioridade estática recebem uma prioridade maior que os processos de prioridade dinâmica.
- Para selecionar um processo para execução, o escalonador do Linux prevê três políticas diferentes :

# Fundamentos de Sistemas Operacionais

## Escalonamento

- **SCHED\_FIFO**: Essa política é válida apenas para os processos de tempo real.
- Na criação, o descritor do processo é inserido no final da fila correspondente à sua prioridade.
- Nessa política, quando um processo é alocado ao processador, ele executa até que uma de três situações ocorra:
  - um processo de tempo real de prioridade superior torna-se apto a executar
  - o processo libera espontaneamente o processador para processos de prioridade igual à sua
  - o processo termina, ou bloqueia-se, em uma operação de entrada e saída ou de sincronização.

## Escalonamento

- **SCHED\_RR**: Na criação, o descritor do processo é inserido no final da fila correspondente à sua prioridade.
- Quando um processo é alocado ao processador, ele executa até que uma de quatro situações ocorra:
  - seu período de execução (*quantum*) tenha se esgotado nesse caso o processo é inserido no final de sua fila de prioridade
  - um processo de prioridade superior torna-se apto a executar
  - o processo libera espontaneamente o processador para processos de prioridade igual a sua
  - o processo termina, ou bloqueia-se, em uma operação de entrada e saída ou de sincronização.
- Essa política também só é válida para processos de tempo real.
- Idêntica a SCHED-FIFO, só que com fatias de tempo – algoritmo do escalonador em rodízio com tempo real.

## Escalonamento

---

- **SCHED\_OTHER**: Corresponde a um esquema de filas multinível de prioridades dinâmicas com *timesharing*.
- Os processo interativos e *batch* recaem nessa categoria.

# Fundamentos de Sistemas Operacionais

## Escalonamento

- A criação de processos em Linux é baseada em uma operação do tipo fork-exec, ou seja, um processo cria uma cópia sua (fork) e em seguida substitui o seu código por um outro (exec).
- No momento da criação, o processo pai (o que fez o fork) cede metade de seu *quantum* restante ao processo filho.
- Esse procedimento é, na verdade, uma espécie de proteção que o sistema faz para evitar que um usuário, a partir de um processo pai, crie um processo filho que execute o mesmo código do pai.
- Sem essa proteção, a cada criação o filho receberia um novo *quantum* integral.
- Da mesma forma, o núcleo Linux previne-se contra o fato de um mesmo usuário lançar vários processos em *background*, ou executar diferentes sessões shell.

## Escalonamento

- O escalonador do Linux é executado a partir de duas formas diferentes.
- A primeira é a forma direta através de uma chamada explícita à rotina que implementa o escalonador.
- Essa é a maneira utilizada pelo núcleo do Linux quando, por exemplo, detecta que um processo deverá ser bloqueado em decorrência de uma operação de entrada e saída ou de sincronização.
- A segunda forma, denominada de *lazy*, também é consequência do procedimento de escalonamento, ocorrendo tipicamente em uma de três situações.

## Escalonamento

- A primeira dessas situações é a rotina de tratamento de interrupção de tempo que atualiza os temporizadores e realiza a contabilização de tempo por processo.
- Essa rotina, ao detectar que um processo esgotou seu *quantum* de execução aciona o escalonador para que seja efetuada uma troca de processo.

## Escalonamento

- A segunda situação ocorre quando um processo de mais alta prioridade é desbloqueado pela ocorrência do evento que esperava.
- A parte do código que efetua o desbloqueio, isto é, trata os eventos de sincronização e de entrada e saída, consulta a prioridade do processo atualmente em execução e compara-a com a do processo que será desbloqueado.
- Se o processo a ser desbloqueado possuir uma prioridade mais alta, o escalonador é acionado, e ocorre uma troca de processo.

## Escalonamento

- A terceira forma de execução *lazy*, é quando um processo explicitamente invoca o escalonador através de uma chamada de sistema do tipo *yield*.
- Essa chamada de sistema permite a um processo “passar sua vez de execução” a outro processo, e, para isso é necessário executar o escalonador.

# Fundamentos de Sistemas Operacionais

## Gerência de Memória

---

- Multiprogramação implica em manter-se vários processos em memória
- Memória precisa ser alocada de forma eficiente para permitir o máximo possível de processos
- Existem diferentes técnicas para gerência de memória

- Um sistema de memória possui pelo menos dois níveis:
  - Memória física – acessada pela CPU
  - Memória secundária – discos
- Programas ficam armazenados em disco. Executá-los significa transferi-los da memória secundária para a memória física.

- Memória Lógica
  - Aquela que o processo enxerga
  - Endereços lógicos são manipulados por processos
- Memória Física
  - Implementada pelos circuitos integrados de memória
  - Endereços físicos correspondem a um endereço real de memória

# Fundamentos de Sistemas Operacionais

## Gerência de Memória

- No Linux a gerência de memória é feita pelo LMM – Linux Memory Manager.
- Para facilitar o mapeamento o sistema operacional divide todo e qualquer processo em páginas de tamanho fixo sendo que o tamanho destas, varia de arquitetura para arquitetura.
- Na arquitetura i386 a página é de 4Kbytes.

# Fundamentos de Sistemas Operacionais

## Gerência de Memória

---

- Para cada página é referenciado um PFN – Page Frame Number - que irá ser utilizado para saber qual a página na memória física o processador deverá acessar, e um número de offset, que irá indicar após a tradução do PFN quantos bytes o processador deve passar para que este acesse as instruções ou informações corretas.

# Fundamentos de Sistemas Operacionais

## Gerência de Memória

- Principais Fatores de Gerenciamento:
  - Proteção: o gerenciador de memória virtual evita que processos no modo Kernel e no modo User se misturem.
  - Memória virtual dividida: no Linux cada processo possui a sua própria tabela de páginas sendo assim a conversão de endereços virtuais em endereços físicos é de tratamento facilitado, pois toda a tabela de páginas pode possuir um endereço virtual de mesmo número apontando cada um para um endereço físico diferente.

# Fundamentos de Sistemas Operacionais

## Gerência de Memória

- Principais Fatores de Gerenciamento:
  - Mapeamento e Paginação: mapeamento é a utilização de endereços virtuais com endereços físicos de memória. Todos os processos são divididos em páginas e representados por uma estrutura de dados chamada pelo Linux de **mm\_struct**. Esta estrutura é composta de informações sobre o que está executando atualmente, e possui ponteiros para uma outra estrutura de dados chamada **vm\_area\_structure** que contém todos os PFN.

# Fundamentos de Sistemas Operacionais

## Gerência de Memória

- Principais Fatores de Gerenciamento:
  - Mapeamento e Paginação: o Linux utiliza uma técnica de envelhecimento de páginas para fazer a escolha de qual página será removida pelo sistema. Este esquema envolve todas as páginas no sistema sendo que a idade de uma página muda a medida que esta vai sendo acessada. Quanto mais uma página é acessada mais jovem ela fica, enquanto que quanto menos acessada uma página for, mais velha ela fica, se tornando uma ótima candidata a troca.

- Principais Fatores de Gerenciamento:
  - Alocação: O Linux utiliza o algoritmo de Buddy para alocar e desalocar páginas sendo que ao alocar páginas este o faz utilizando um conjunto de páginas(blocos). O gerenciador aloca e desaloca memória com extremo dinamismo causando uma fragmentação da memória. Este problema é resolvido pelo Kernel que junta os espaços da memória semelhantes - 'desfragmentador'.

- Principais Fatores de Gerenciamento:
  - Memória Cache: É utilizado para otimizar o acesso a informações.
  - O Linux utiliza os seguintes tipos de Cache:
    - Buffer: Contém dados(512Kb) de tamanho fixo que são usados por drivers de dispositivos. Se a informação pode ser encontrada no buffer então não é preciso ler algum dispositivo físico(ex.: HD), tornando o acesso mais rápido.

# Fundamentos de Sistemas Operacionais

## Gerência de Memória

---

- Page: É usado para tornar mais rápido o acesso a vários tipos de informação (imagens, dados) no disco.
- Swap: Apenas as páginas modificadas são salvas no arquivo. Quando se lê uma página, ela só será salva novamente se for escrito algo na página. Caso não, ela pode ser descartada.

# Fundamentos de Sistemas Operacionais

## Sistemas de Arquivos no Linux

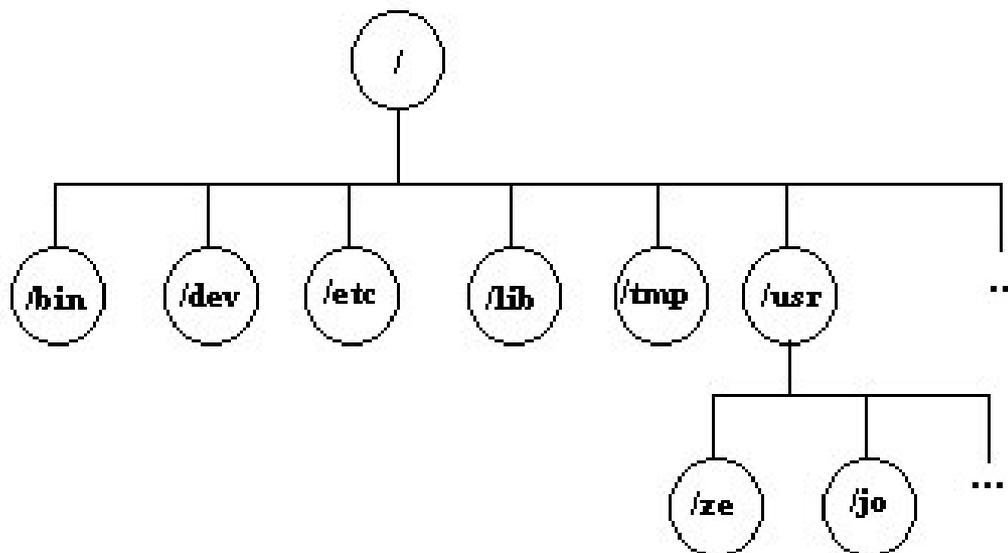
---

- Arquivos - Mecanismos de abstração que fornecem uma forma de armazenar e recuperar informações.
- O sistema de arquivos é a parte mais visível de um sistema operacional, pois a manipulação de arquivos é uma atividade frequentemente realizada pelos usuários, devendo sempre ocorrer de maneira uniforme, independentemente dos diferentes dispositivos de armazenamento.

# Fundamentos de Sistemas Operacionais

## Sistemas de Arquivos no Linux

- O sistema de arquivos aparece para o usuário como uma hierarquia organizada, transparente, na forma de uma árvore invertida.



# Fundamentos de Sistemas Operacionais

## Sistemas de Arquivos no Linux

- O Linux olha o arquivo como uma seqüência de bytes, sem nenhuma estrutura, e isto dá uma flexibilidade espantosa ao sistema de arquivos.
- Todos os sistemas de arquivos tem três componentes em comum:
  - Superblock - Informações em dados físicos e a própria estrutura de arquivos, incluindo o tamanho do sistema de arquivo, e o número, lista e índice de blocos livres, mais o número e índice de inodes livres;
  - Lista de Inode (ou tabela) - Um índice de informação sobre localidade e características de arquivos;
  - Bloco de Dados - Armazenamento dos dados atuais.

# Fundamentos de Sistemas Operacionais

## Sistemas de Arquivos no Linux

- Quando um processo cria um arquivo, é preciso que tal arquivo receba um nome, normalmente dado pelo processo.
- Quando tal processo termina sua execução, o arquivo continua a existir, podendo ser acessado por outros processos, usando para tanto o nome atribuído ao arquivo.
- O Linux faz distinção entre nomes maiúsculos e minúsculos.
- Normalmente um nome de arquivo é composto de nome e uma extensão, separada por ponto.
- No Linux, o tamanho da extensão, se houver, fica a critério do usuário, e um arquivo pode até ter duas ou mais extensões.
- Exemplo: backup.tar.gz

# Fundamentos de Sistemas Operacionais

## Sistemas de Arquivos no Linux

- Não há limite de números de caracteres utilizados para dar nome a arquivos.
- O sistema de arquivo do Linux identifica os drivers de dispositivos como uma estrutura hierárquica comum.
- O Linux soma cada novo sistema de arquivo dentro desta árvore de sistema de arquivos enquanto é montado.
- Todos sistemas de arquivos, de qualquer tipo, são montados sobre um diretório e os arquivos do sistema de arquivo, montados sobre os conteúdos existentes daquele diretório.

# Fundamentos de Sistemas Operacionais

## Sistemas de Arquivos no Linux

- Quando discos são iniciados (usando fdisk), eles têm uma estrutura de partição importada que divide o disco físico dentro de um número de partições lógicas.
- Cada parte da partição gera um sistema de arquivo único.
- Os sistemas de arquivos organizam os arquivos dentro de uma lógica hierárquica com diretórios, links flexíveis e assim montados em blocos, em dispositivos físicos.
- Dispositivos que podem conter sistemas de arquivo são conhecidos como blocos de dispositivos.

# Fundamentos de Sistemas Operacionais

## Sistemas de Arquivos no Linux

- O primeiro sistema de arquivo projetado especificamente para o Linux, o sistema de arquivo estendido, ou EXT, foi introduzido em abril de 92 e veio para corrigir bugs do Minix (primeiro sistema de arquivo que o Linux tinha acesso - escasso e de baixa performance), assim como melhorar sua performance.
- Em 93, o segundo sistema de arquivo estendido, ou EXT2, foi adicionado ao Linux.

# Fundamentos de Sistemas Operacionais

## Sistema de Arquivo Extendido EXT2

- O sistema de arquivo EXT2 enxerga muitos sistemas de arquivos.
- Foi construído com a premissa de que os dados são seguramente guardados em blocos de dados.
- Esses blocos de dados tem o mesmo comprimento e, embora este comprimento possa variar entre diferentes sistemas de arquivos EXT2, o tamanho de bloco de um sistema de arquivo EXT2 particular é atribuído quando ele é criado.

# Fundamentos de Sistemas Operacionais

## Sistema de Arquivo Extendido EXT2

- Os blocos no sistema de arquivo contém também informações que descrevem a estrutura do sistema de arquivo.
- O EXT2 define a topologia de sistema de arquivo por descrever cada arquivo no sistema com uma estrutura de dados do inode.
- No EXT2, o inode é o bloco básico.
- Todo arquivo e diretório no sistema de arquivo é descrito por um, e unicamente um inode.
- O EXT2 inode para cada grupo de bloco, é guardado na tabela do inode juntamente com um mapa de bits que permite ao sistema guardar a trilha de alocação e desalocação do inode.

# Fundamentos de Sistemas Operacionais

## Sistema de Arquivo Extendido EXT2

---

- Os inodes do sistema de arquivo são todos guardados juntamente em tabelas de inode.
- Diretórios EXT2 são simplesmente arquivos especiais que contêm ponteiros aos inodes de suas entradas de diretório.

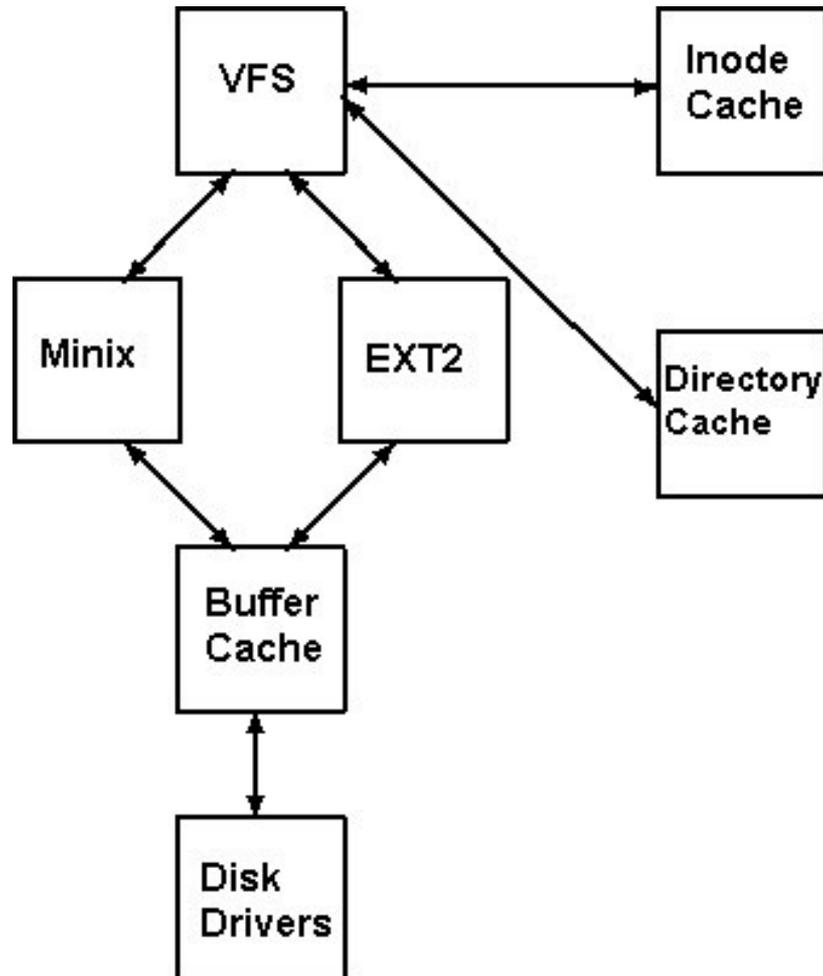
# Fundamentos de Sistemas Operacionais

## Sistema de Arquivo Extendido EXT2

- Um grande avanço ocorreu no lançamento do EXT Linux: os sistemas de arquivos reais foram separados do sistema operacional e serviços de sistema por uma camada de interface conhecida como o sistema de arquivo virtual, ou VFS.
- O VFS permite ao Linux suportar muitos sistemas de arquivos mostrados ao usuário final como um único sistema de arquivo. A camada de sistema de arquivo virtual do Linux permite ao usuário transparentemente montar os muitos sistemas de arquivos diferentes ao mesmo tempo.

# Fundamentos de Sistemas Operacionais

## VFS – Sistema de Arquivos Virtual



# Fundamentos de Sistemas Operacionais

## VFS – Sistema de Arquivos Virtual

- O sistema de arquivos virtual do Linux deve administrar não menos que os sistemas de arquivos diferentes que são montados em qualquer tempo.
- Para fazer isso, o kernel mantém uma estrutura de dados (virtual) que descreve todo o arquivo de sistema real montado.
- O VFS descreve os sistemas de arquivos em termos de superblocks e inodes, do mesmo modo que o sistema de arquivos EXT2 usa superblocks e inodes.
- Como o EXT2 inode, o VFS inode descreve arquivos e diretórios dentro do sistema, seus conteúdos e topologia.

# Fundamentos de Sistemas Operacionais

## VFS – Sistema de Arquivos Virtual

- Quando cada sistema de arquivos é inicializado, ele se registra como um VFS.
- Isto acontece quando o sistema operacional se inicializa no sistema.
- Os sistemas de arquivos reais são todos construídos pelo próprio Kernel ou por módulos de inicialização.
- Os módulos do sistema de arquivos são inicializados quando o sistema os requer
- Exemplo: se o sistema de arquivos VFAT é implementado como modulo de kernel, então ele só é carregado quando um sistema de arquivos VFAT é montado.

# Fundamentos de Sistemas Operacionais

## VFS – Sistema de Arquivos Virtual

---

- Quando um bloco de dispositivo baseado em sistema de arquivos é montado, e isto inclui o root file system, o VFS tem que ler seus superblocks.
- Cada superblock de leitura de rotinas dos tipos de sistemas de arquivos deve movimentar a topologia do sistema de arquivos e mapear essas informações em uma estrutura de dados de VFS superblock.
- O VFS guarda uma lista dos sistemas de arquivos montados no sistema junto com seu VFS superblock.

# Fundamentos de Sistemas Operacionais

## VFS – Sistema de Arquivos Virtual

---

- Cada superblock contém informações e ponteiros para rotinas que desempenham funções particulares.
- Essa rotina de leitura do EXT2 inode, como as rotinas de leitura de inode de todos os sistemas de arquivos, completam os campos com um VFS inode.
- Cada VFS superblock contém um ponteiro para o primeiro VFS inode no sistema de arquivos.

# Fundamentos de Sistemas Operacionais

## Gerência de Entrada e Saída

- A gerência de entrada e saída no Linux é implementada através de device drivers, um device para cada dispositivo.
- Os device drivers são acoplados ao sistema operacional quando o kernel é gerado.
- Sempre que um novo dispositivo é acrescentado ao sistema, o driver correspondente deve ser acoplado ao núcleo.
- O acesso a dispositivos é na forma de arquivos especiais.

# Fundamentos de Sistemas Operacionais

## Gerência de Entrada e Saída

---

- No Linux, todas as operações de E/S são realizadas como uma seqüência de bytes, não existindo o conceito de registro ou métodos de acesso.
- Dessa forma, as system calls de E/S podem manipular qualquer tipo de dispositivo de forma uniforme.

# Fundamentos de Sistemas Operacionais

## Gerência de Entrada e Saída

---

- Os arquivos especiais podem ser acessados da mesma forma que qualquer outro arquivo, utilizando simplesmente as system calls de leitura e gravação.
- O Linux trabalha com dois tipos de operações de entrada e saída:
  - orientada a blocos
  - orientada a caracter

# Fundamentos de Sistemas Operacionais

## Gerência de Entrada e Saída

---

- As operações orientadas a bloco estão geralmente associadas a dispositivos com altas taxas de transferência, como discos, e têm o objetivo de minimizar o número de transferências entre o dispositivo e a memória, utilizando buffer caches.
- Por exemplo, quando uma operação de leitura a disco é realizada, um bloco é transferido para a memória e, posteriormente, processado.

# Fundamentos de Sistemas Operacionais

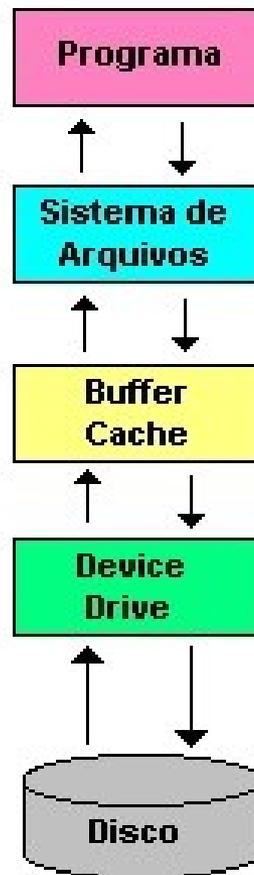
## Gerência de Entrada e Saída

---

- Dispositivos orientados a caracter estão associados normalmente a dispositivos lentos, como terminais, onde a taxa de transferência entre o dispositivo e a memória é realizada caracter a caracter.

# Fundamentos de Sistemas Operacionais

## Gerência de Entrada e Saída



**Fig. Buffer Cache**

# Fundamentos de Sistemas Operacionais

## Redirecionamento de Entrada e Saída

---

- O usuário pode, através do redirecionamento de E/S, redefinir de onde um comando ou programa receberá sua entrada e para onde enviará sua saída.
- A entrada de um comando são os dados sobre os quais o comando irá operar.
- Estes dados podem vir de um arquivo especificado pelo usuário, de um arquivo de sistema, do terminal ou da saída de outro comando.

# Fundamentos de Sistemas Operacionais

## Redirecionamento de Entrada e Saída

- A saída de um comando é o resultado da operação que ele realiza sobre a entrada.
- A saída dos comandos pode ser impressa na tela do terminal, enviada a um arquivo, ou servir de entrada para outro comando.
- Um comando Linux, normalmente requer uma entrada e uma saída, a fim de exibir os resultados.

# Fundamentos de Sistemas Operacionais

## Redirecionamento de Entrada e Saída

---

- Quando nenhum nome de arquivo é especificado, o shell admite que o teclado do usuário será sua entrada.
- O teclado é considerado a **entrada padrão** do sistema.
- E quando alguns resultados precisam ser exibidos, o shell assume que a tela será a **saída padrão** do sistema.

# Fundamentos de Sistemas Operacionais

## Estrutura do Sistema de Arquivos Linux

---

- Arquivos
  - É onde gravamos nossos dados. Um arquivo pode conter um texto feito por nós, uma música, programa, planilha, etc.
  - Cada arquivo deve ser identificado por um nome, assim ele pode ser encontrado facilmente quando desejar usá-lo. Se estiver fazendo um trabalho de história, nada melhor que salva-lo com o nome historia. Um arquivo pode ser binário ou texto.

# Fundamentos de Sistemas Operacionais

## Estrutura do Sistema de Arquivos Linux

- Arquivos

- O GNU/Linux é *Case Sensitive* ou seja, ele diferencia letras *maiúsculas* e *minúsculas* nos arquivos. O arquivo *historia* é completamente diferente de *Historia*. Esta regra também é válido para os *comandos* e *diretórios*. Prefira, sempre que possível, usar letras minúsculas para identificar seus arquivos, pois quase todos os comandos do sistema estão em *minúsculas*.
- Um arquivo oculto no GNU/Linux é identificado por um "." no início do nome (por exemplo, `.bashrc`). Arquivos ocultos não aparecem em listagens normais de diretórios, deve ser usado o comando `ls -a` para também listar arquivos ocultos.

# Fundamentos de Sistemas Operacionais

## Estrutura do Sistema de Arquivos Linux

---

- Extensão de arquivos
  - A extensão serve para identificar o tipo do arquivo. A extensão são as letras após um "." no nome de um arquivo, explicando melhor:
  - relatório.**txt** - O .txt indica que o conteúdo é um arquivo texto.
  - script.**sh** - Arquivo de Script (interpretado por /bin/sh).
  - system.**log** - Registro de algum programa no sistema.
  - arquivo.**gz** - Arquivo compactado pelo utilitário gzip.
  - index.**html** - Página de Internet (formato Hypertexto).

# Fundamentos de Sistemas Operacionais

## Estrutura do Sistema de Arquivos Linux

- Extensão de Arquivos
  - A extensão de um arquivo também ajuda a saber o que precisamos fazer para abri-lo. Por exemplo, o arquivo relatório.txt é um texto simples e podemos ver seu conteúdo através do comando cat, já o arquivo index.html contém uma página de Internet e precisaremos de um navegador para poder visualizá-lo (como o lynx, Mosaic ou o Netscape).
  - A extensão (na maioria dos casos) não é requerida pelo sistema operacional GNU/Linux, mas é conveniente o seu uso para determinarmos facilmente o tipo de arquivo e que programa precisaremos usar para abri-lo.

# Fundamentos de Sistemas Operacionais

## Estrutura do Sistema de Arquivos Linux

- Arquivo texto e binário
  - Quanto ao tipo, um arquivo pode ser de texto ou binário:
    - ✓ Texto :Seu conteúdo é compreendido pelas pessoas. Um arquivo texto pode ser uma carta, um script, um programa de computador escrito pelo programador, arquivo de configuração, etc.
    - ✓ Binário :Seu conteúdo somente pode ser entendido por computadores. Contém caracteres incompreensíveis para pessoas normais. Um arquivo binário é gerado através de um arquivo de programa (formato texto) através de um processo chamado de compilação. Compilação é basicamente a conversão de um programa em linguagem humana para a linguagem de máquina.

# Fundamentos de Sistemas Operacionais

## Estrutura do Sistema de Arquivos Linux

- Diretórios
  - Diretório é o local utilizado para armazenar conjuntos de arquivos para melhor organização e localização. O diretório, como o arquivo, também é "*Case Sensitive*" (diretório /teste é completamente diferente do diretório /Teste).
  - Não podem existir dois arquivos com o mesmo nome em um diretório, ou um subdiretório com um mesmo nome de um arquivo em um mesmo diretório.
  - Um diretório nos sistemas Linux/UNIX são especificados por uma "/" e não uma "\" como é feito no DOS e Windows. Para detalhes sobre como criar um diretório, veja o comando mkdir.

# Fundamentos de Sistemas Operacionais

## Estrutura do Sistema de Arquivos Linux

---

- Diretório Raiz
  - Este é o diretório principal do sistema. Dentro dele estão todos os diretórios do sistema. O diretório Raiz é representado por uma "/", assim se você digitar o comando `cd /` você estará acessando este diretório.

# Fundamentos de Sistemas Operacionais

## Estrutura do Sistema de Arquivos Linux

- **Diretório Raiz**

➤ Nele estão localizados outros diretórios como o /bin, /sbin, /usr, /usr/local, /mnt, /tmp, /var, /home, etc. Estes são chamados de *sub-diretórios* pois estão dentro do diretório "/". A estrutura de *diretórios* e *sub-diretórios* pode ser identificada da seguinte maneira:

- ✓ /
- ✓ /bin
- ✓ /sbin
- ✓ /usr
- ✓ /usr/local
- ✓ /mnt
- ✓ /tmp
- ✓ /var
- ✓ /home

# Fundamentos de Sistemas Operacionais

## Estrutura do Sistema de Arquivos Linux

---

- Diretórios

- A estrutura de diretórios também é chamada de *Árvore de Diretórios* porque é parecida com uma *árvore* de cabeça para baixo. Cada diretório do sistema tem seus respectivos arquivos que são armazenados conforme regras definidas pela *FHS (FileSystem Hierarchy Standard - Hierarquia Padrão do Sistema de Arquivos)* versão 2.0, definindo que tipo de arquivo deve ser armazenado em cada diretório.

# Fundamentos de Sistemas Operacionais

## Estrutura do Sistema de Arquivos Linux

---

- **Diretório atual**

- É o diretório em que nos encontramos no momento. Você pode digitar `pwd` para verificar qual é seu diretório atual.
- O diretório atual também é identificado por um "." (ponto). O comando `ls .` pode ser usado para listar seus arquivos (é claro que isto é desnecessário porque se não digitar nenhum diretório, o comando `ls` listará o conteúdo do diretório atual).

# Fundamentos de Sistemas Operacionais

## Estrutura do Sistema de Arquivos Linux

- **Diretório home**
  - Também chamado de diretório de usuário. Em sistemas GNU/Linux cada usuário (inclusive o root) possui seu próprio diretório onde poderá armazenar seus programas e arquivos pessoais.
  - Este diretório está localizado em `/home/[login]`, neste caso se o seu login for "joao" o seu diretório home será `/home/joao`. O diretório home também é identificado por um `~(til)`, você pode digitar tanto o comando `ls /home/joao` como `ls ~` para listar os arquivos de seu diretório home.
  - O diretório home do usuário root (na maioria das distribuições GNU/Linux) está localizado em `/root`.
  - Dependendo de sua configuração e do número de usuários em seu sistema, o diretório de usuário pode ter a seguinte forma: `/home/[1letra do nome]/[login]`, neste caso se o seu login for "joao" o seu diretório home será `/home/j/joao`.

# Fundamentos de Sistemas Operacionais

## Estrutura do Sistema de Arquivos Linux

---

- Diretório Superior
  - O diretório superior (Upper Directory) é identificado por “..” (2 pontos).
  - Caso estiver no diretório `/usr/local` e quiser listar os arquivos do diretório `/usr` você pode digitar, `ls ..`. Este recurso também pode ser usado para copiar, mover arquivos/diretórios, etc.

# Fundamentos de Sistemas Operacionais

## Estrutura do Sistema de Arquivos Linux

---

- Diretório Anterior

- O diretório anterior é identificado por "-". É útil para retornar ao último diretório usado.
- Se estive no diretório `/usr/local` e digitar `cd /lib`, você pode retornar facilmente para o diretório `/usr/local` usando `cd -`.

# Fundamentos de Sistemas Operacionais

## Estrutura do Sistema de Arquivos Linux

- Caminho na estrutura de diretórios
  - São os diretórios que teremos que percorrer até chegar no arquivo ou diretório que procuramos. Se desejar ver o arquivo `/usr/doc/copyright/GPL` você tem duas opções:
  - Mudar o diretório padrão para `/usr/doc/copyright` com o comando `cd /usr/doc/copyright` e usar o comando `cat GPL`
  - Usar o comando "cat" especificando o caminho completo na estrutura de diretórios e o nome de arquivo:  
`cat /usr/doc/copyright/GPL.`

# Fundamentos de Sistemas Operacionais

## Estrutura do Sistema de Arquivos Linux

- Caminho na estrutura de diretórios
  - As duas soluções acima permitem que você veja o arquivo GPL. A diferença entre as duas é a seguinte:
    - ✓ Na primeira, você muda o diretório padrão para `/usr/doc/copyright` (confira digitando `pwd`) e depois o comando `cat GPL`. Você pode ver os arquivos de `/usr/doc/copyright` com o comando `"ls"`.  
`/usr/doc/copyright` é o caminho de diretório que devemos percorrer para chegar até o arquivo GPL.
    - ✓ Na segunda, é digitado o caminho completo para o "cat" localizar o arquivo GPL: `cat /usr/doc/copyright/GPL`. Neste caso, você continuará no diretório padrão (confira digitando `pwd`). Digitando `ls`, os arquivos do diretório atual serão listados.
  - O *caminho de diretórios* é necessário para dizer ao sistema operacional onde encontrar um arquivo na "árvore" de diretórios.

# Fundamentos de Sistemas Operacionais

## Estrutura do Sistema de Arquivos Linux

- Estrutura básica de diretórios
  - /bin : Contém arquivos programas do sistema que são usados com frequência pelos usuários.
  - /boot : Contém arquivos necessários para a inicialização do sistema.
  - /cdrom : Ponto de montagem da unidade de CD-ROM.
  - /dev : Contém arquivos usados para acessar dispositivos (periféricos) existentes no computador.
  - /etc : Arquivos de configuração de seu computador local.
  - /floppy : Ponto de montagem de unidade de disquetes
  - /home : Diretórios contendo os arquivos dos usuários.
  - /lib : Bibliotecas compartilhadas pelos programas do sistema e módulos do kernel.

# Fundamentos de Sistemas Operacionais

## Estrutura do Sistema de Arquivos Linux

- Estrutura básica de diretórios
  - /lost+found : Local para a gravação de arquivos/diretórios recuperados pelo utilitário fsck.ext2. Cada partição possui seu próprio diretório lost+found.
  - /mnt : Ponto de montagem temporário.
  - /proc : Sistema de arquivos do kernel. Este diretório não existe em seu disco rígido, ele é colocado lá pelo kernel e usado por diversos programas que fazem sua leitura, verificam configurações do sistema ou modificar o funcionamento de dispositivos do sistema através da alteração em seus arquivos.
  - /root : Diretório do usuário root.

# Fundamentos de Sistemas Operacionais

## Estrutura do Sistema de Arquivos Linux

- Estrutura básica de diretórios
  - /sbin :Diretório de programas usados pelo superusuário (root) para administração e controle do funcionamento do sistema.
  - /tmp :Diretório para armazenamento de arquivos temporários criados por programas.
  - /usr : Contém maior parte de seus programas. Normalmente acessível somente como leitura.
  - /var : Contém maior parte dos arquivos que são gravados com frequência pelos programas do sistema, e-mails, spool de impressora, cache, etc.

# Fundamentos de Sistemas Operacionais

## Sistemas Operacionais de Rede

---

- Coleção de computadores conectados através de uma rede
  - Cada computador possui seu SO local
  - Cada máquina possui alto grau de autonomia
- Implementação relativamente simples
  - SOs incorporam módulos para acessar recursos remotos
  - Comunicação entre sistemas através de protocolos de transporte
    - ✓ Sockets
    - ✓ RPC/RMI

# Fundamentos de Sistemas Operacionais

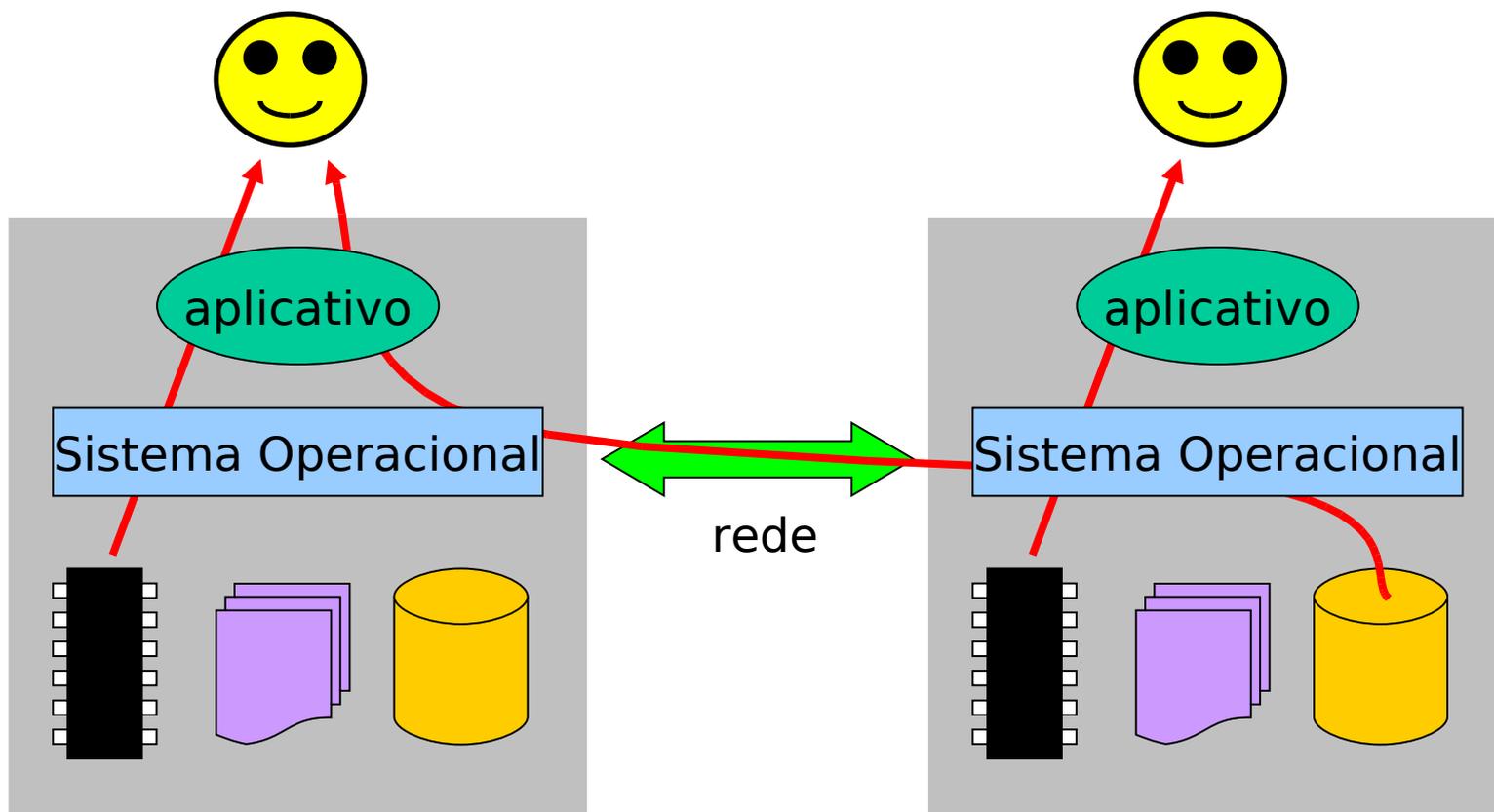
## Sistemas Operacionais de Rede

---

- Transferências explícitas
  - O usuário deve conhecer a localização dos recursos
  - Os recursos pertencem a computadores específicos
- Exemplos:
  - Compartilhamento de impressoras e arquivos
  - Web, E-mail
  - Serviços de autenticação

# Fundamentos de Sistemas Operacionais

## Sistemas Operacionais de Rede



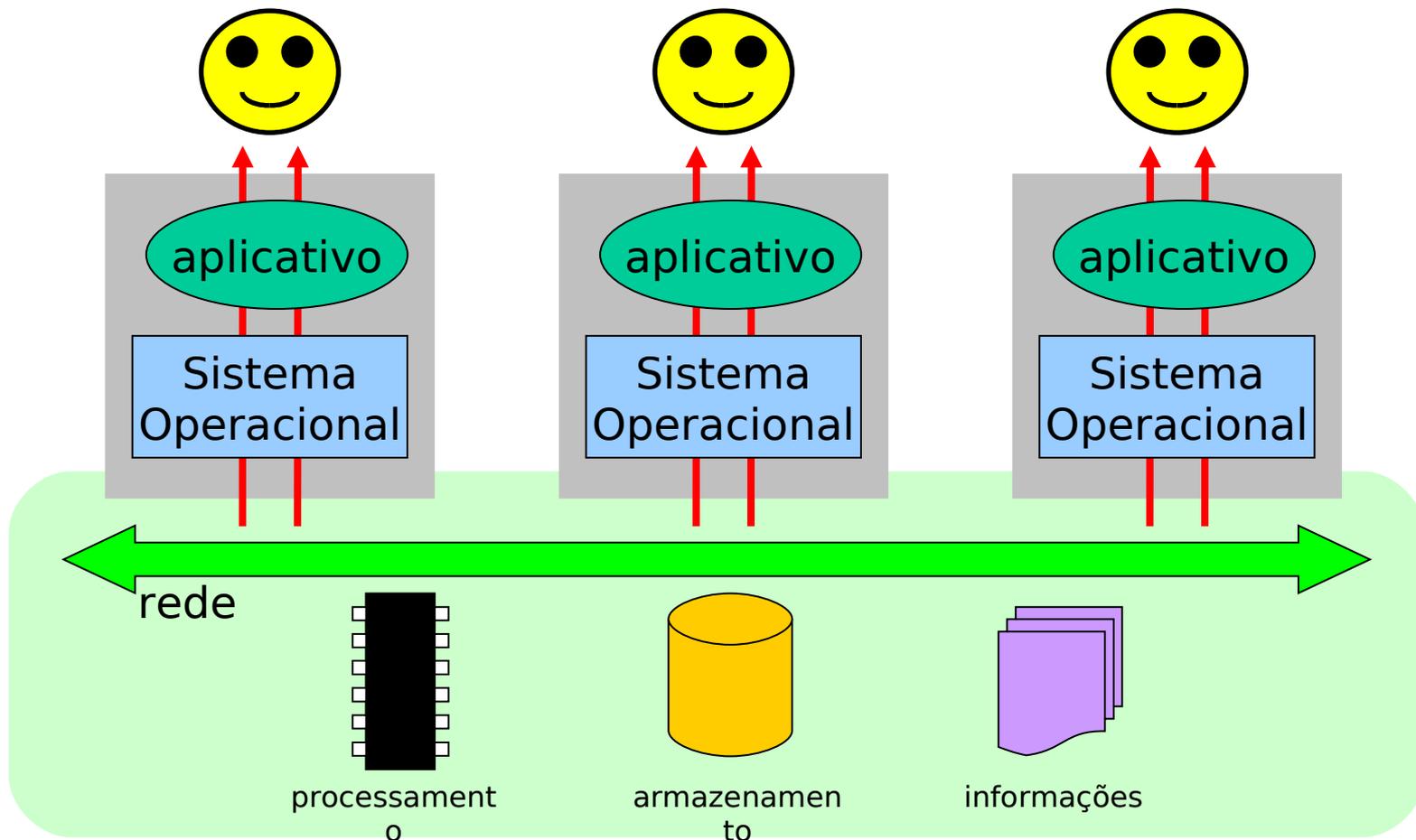
# Fundamentos de Sistemas Operacionais

## Sistema Operacional Distribuído

- Objetivos:
  - Construção de um ambiente computacional virtual
  - Localização dos recursos é abstraída
  - Localização do processamento é abstraída
  - Mecanismos transparentes de distribuição, replicação e tolerância a falhas
- O usuário vê o sistema como um ambiente virtual, e não como um conjunto de computadores conectados por uma rede
- O Sistema Operacional Distribuído deve:
  - Controlar a alocação de recursos para tornar seu uso eficiente
  - Prover um ambiente de computação virtual de alto nível
  - Esconder a distribuição dos recursos e do processamento

# Fundamentos de Sistemas Operacionais

## Sistema Operacional Distribuído



## Arquiteturas de Redes

---

- Projetistas organizam protocolos em **camadas**
- Há uma porção de camada  $n$  em cada entidade de rede
  - Protocolo de uma camada  $n$  é *distribuído*
- Estas porções se comunicam trocando mensagens de camada  $n$ 
  - $n$ -PDU (*protocol data units*)



# Fundamentos de Sistemas Operacionais

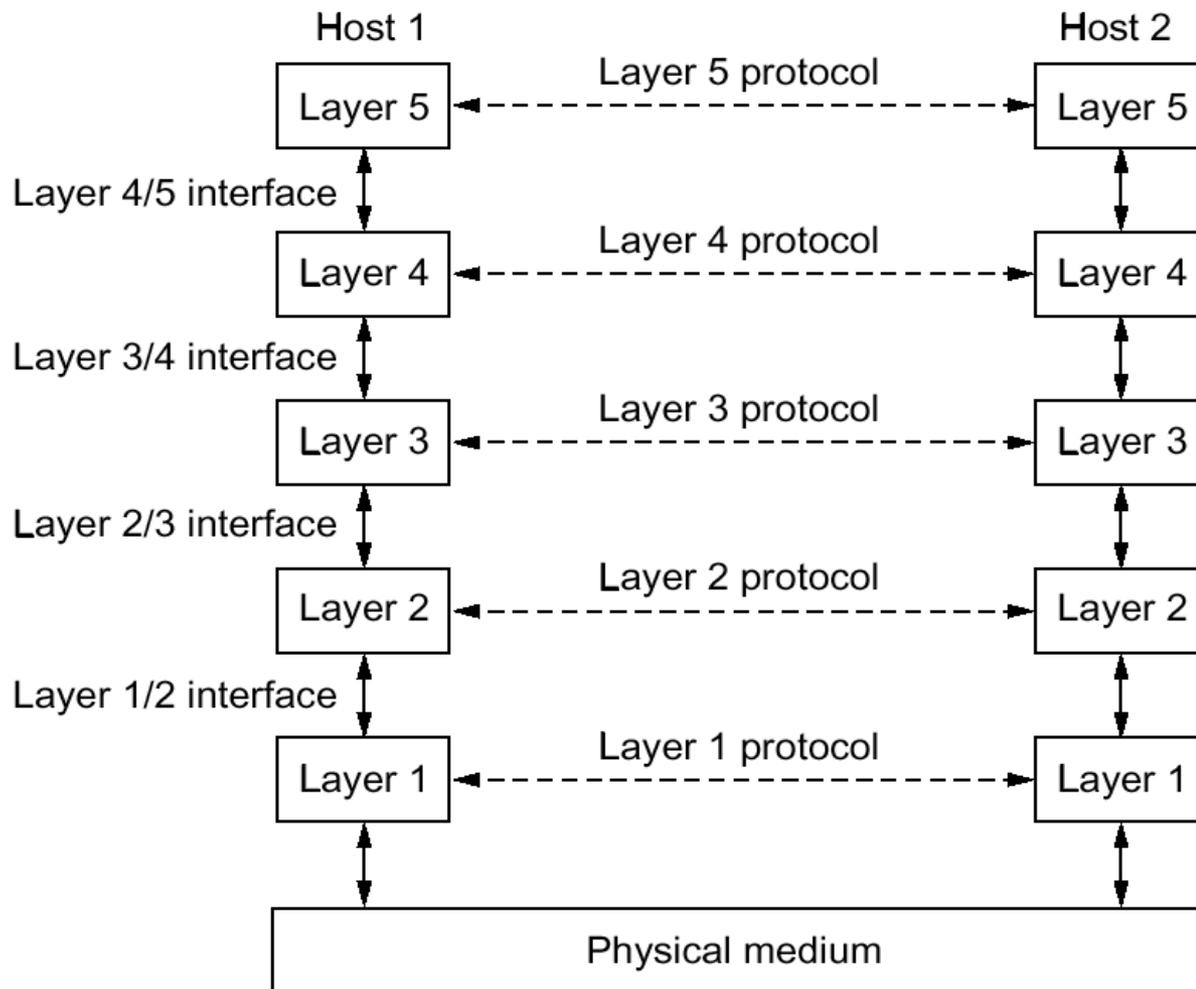
## Arquitetura de Redes

---

- Hierarquias de Protocolos
  - Níveis (Camadas)
  - Protocolos
  - Interfaces
- Virtual/Horizontal **X** Real/Vertical
- Pilhas de Protocolos
- Arquitetura da Rede
  - Níveis e Protocolos

# Fundamentos de Sistemas Operacionais

## Camadas, Protocolos e Interfaces



# Fundamentos de Sistemas Operacionais

## Modelos de Referência

---

- Modelo OSI da ISO
  - *International Standards Organization*
  - *Open Systems Interconnection*
    - 7 camadas
- Modelo TCP/IP
  - *Transport Control Protocol*
  - *Internet Protocol*
    - 4 camadas

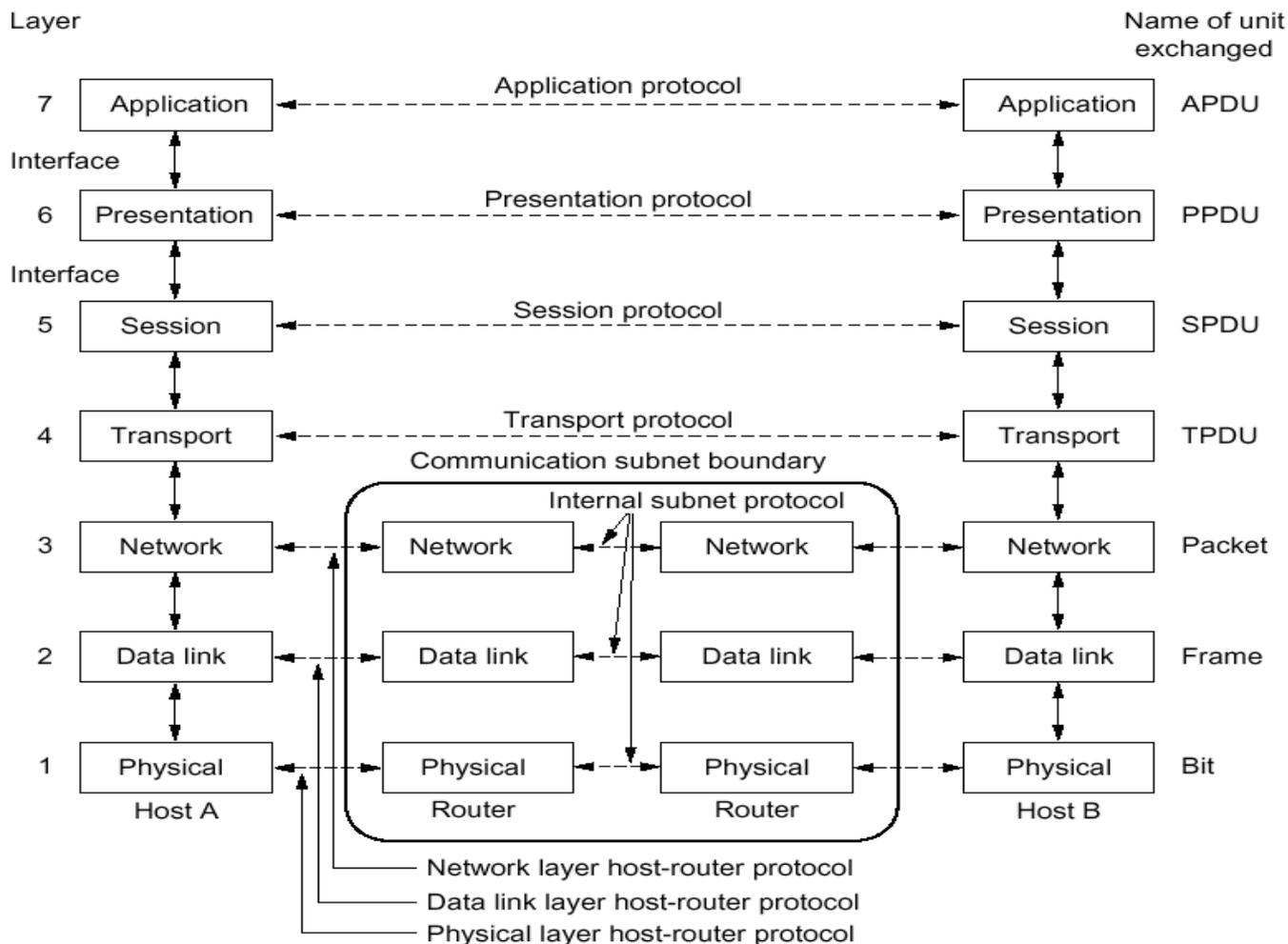
## Modelo OSI

---

- Princípios básicos:
  - Um nível de abstração para cada camada
  - Cada camada com sua função bem definida
  - A cada camada, protocolos padronizados internacionalmente
  - Camadas escolhidas de modo a limitar o fluxo de informações entre as camadas
  - Número de camadas nem grande demais, nem pequeno demais

# Fundamentos de Sistemas Operacionais

## O Modelo de Referência OSI



## Camada Física

---

- Trata das características mecânicas e elétricas do meio físico de transmissão
  - Cabos, antenas, conectores, equipamentos, etc.
- Deve garantir que o **bit** enviado pelo transmissor seja entendido corretamente pelo receptor

# Fundamentos de Sistemas Operacionais

## Camada de Enlace de Dados

---

- Trata do fluxo de dados no enlace entre o transmissor e receptor.
  - Controle de fluxo
  - Detecção e recuperação de erro
  - Reconhecimento de carona
- As unidades de dados são **quadros**
- Difusão: controle de acesso ao canal

# Fundamentos de Sistemas Operacionais

## Camada de Rede

---

- Controle de operações na sub-rede
  - Roteamento
    - Estático ou dinâmico
  - Controle de congestionamento
  - Contabilização (tarifação)
  - Interconexão de redes
- As unidades de dados são os **pacotes**

# Fundamentos de Sistemas Operacionais

## Camada de Transporte

---

- Divide as mensagens em pacotes
- É a primeira camada fim-a-fim
- Deve garantir que os pacotes foram entregues sem erro, na seqüência e sem perdas ou duplicação
  - controle de fluxo entre *hosts*

# Fundamentos de Sistemas Operacionais

## Camada de Sessão

---

- Permite aplicações em *hosts* separados partilharem uma conexão (sessão)
- Provê:
  - controle de diálogo
  - Sincronização
  - *checkpointing*

# Fundamentos de Sistemas Operacionais

## Camada de Apresentação

---

- Traduz dados entre os formatos requeridos pela rede e os formatos requeridos pelos *hosts*
- Provê
  - conversão de protocolos
  - tradução de dados
  - conversão de caracteres

# Fundamentos de Sistemas Operacionais

## Camada de Aplicação

---

- Provê serviços que suportam diretamente as aplicações dos usuários, tais como
  - e-mail,
  - acesso a banco de dados e
  - transferência de arquivos

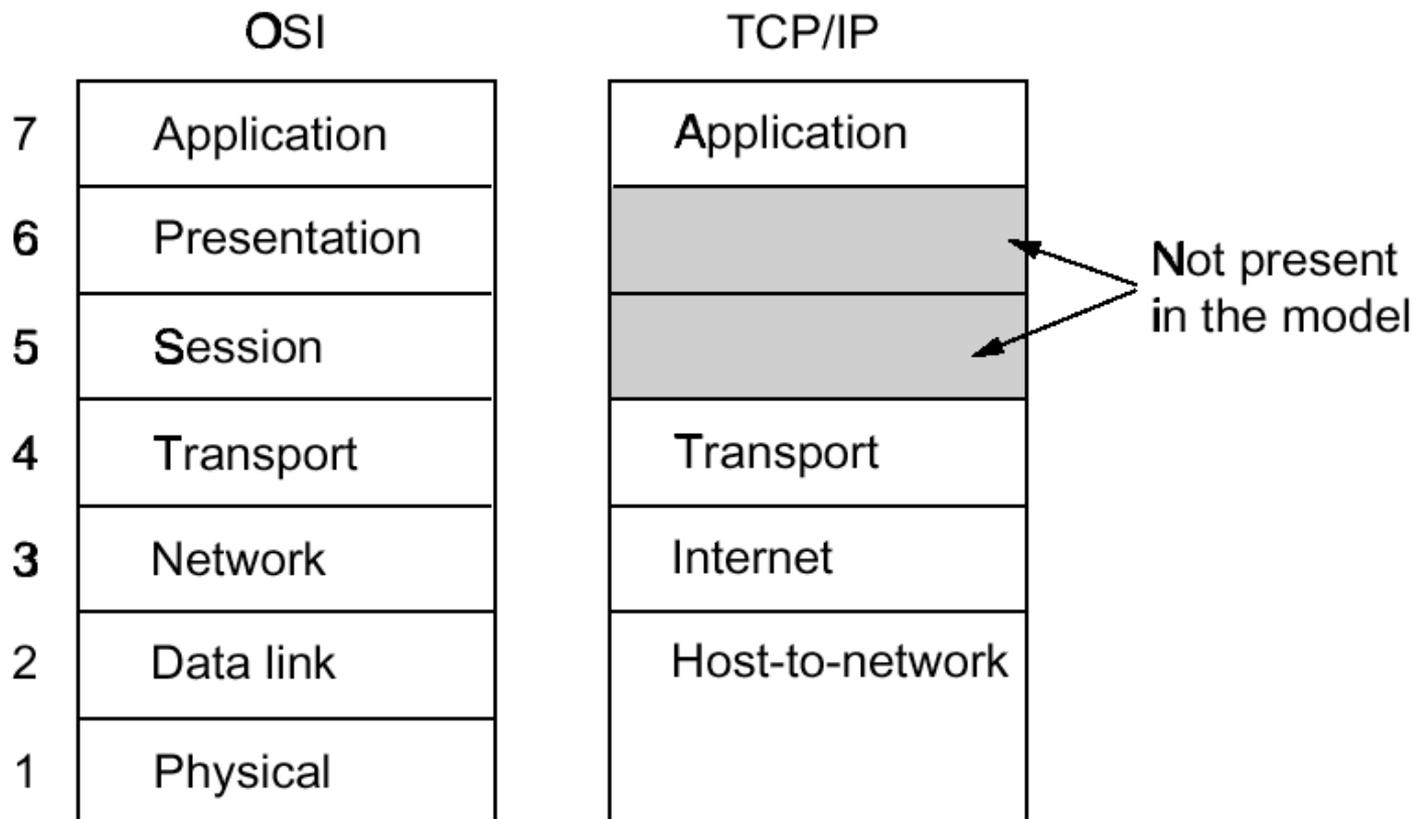
# Fundamentos de Sistemas Operacionais

## O Modelo de Referência TCP/IP

- Surgiu com a ARPANET
- Camada Inter-rede (**Internet**)
  - Serviço de Comutação de Pacotes Não Orientado a conexões: habilidade de sobreviver a falhas no *hardware* da sub-rede
  - Protocolo IP
- Nível **TCP**
  - TCP: Orientado a conexões  confiável
  - UDP: Não Orientado a Conexões  não confiável

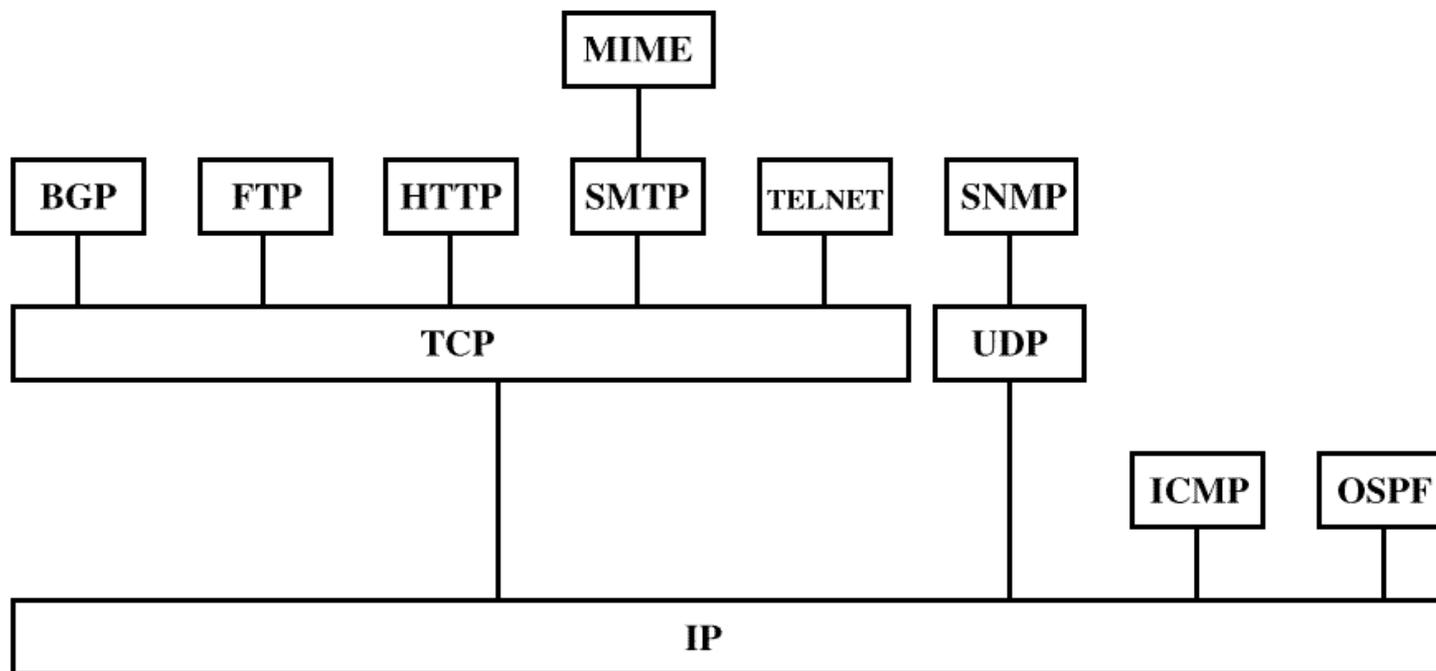
# Fundamentos de Sistemas Operacionais

## O Modelo de Referência TCP/IP



# Fundamentos de Sistemas Operacionais

## Família de Protocolos TCP/IP



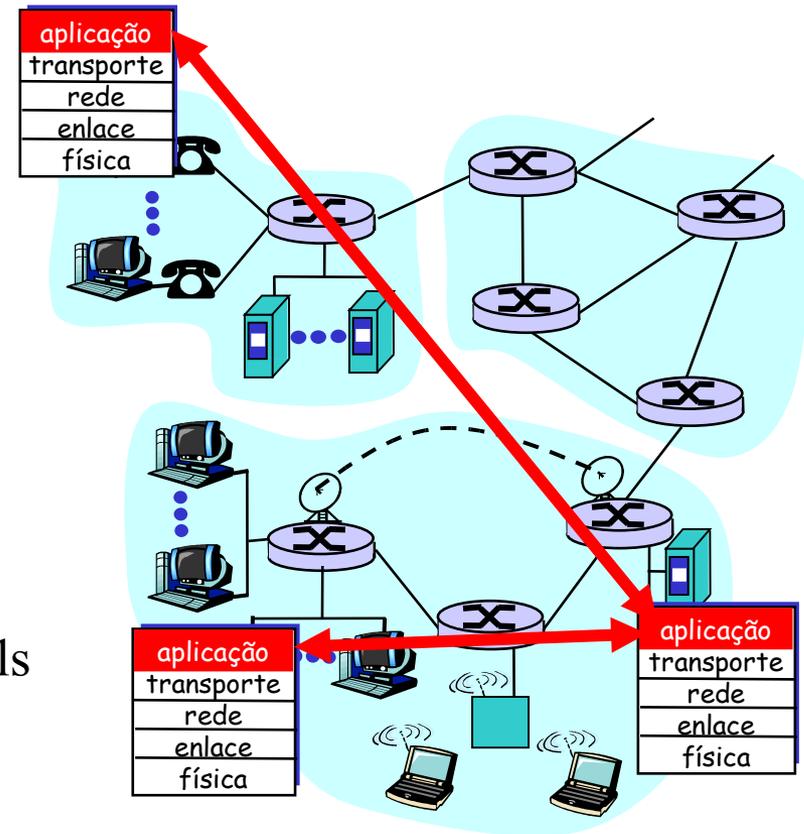
BGP = Border Gateway Protocol  
FTP = File Transfer Protocol  
HTTP = Hypertext Transfer Protocol  
ICMP = Internet Control Message Protocol  
IP = Internet Protocol  
OSPF = Open Shortest Path First

MIME = Multi-Purpose Internet Mail Extension  
SMTP = Simple Mail Transfer Protocol  
SNMP = Simple Network Management Protocol  
TCP = Transmission Control Protocol  
UDP = User Datagram Protocol

# Fundamentos de Sistemas Operacionais

## Aplicações e Protocolos da Camada de Aplicação

- Aplicação: processos distribuídos em comunicação
  - executam em hospedeiros no “espaço de usuário”
  - trocam mensagens para implementar a aplicação
  - p.ex., correio, transf. de arquivo, WWW
- Protocolos da camada de aplicação
  - uma “parte” da aplicação
  - define mensagens trocadas por apls e ações tomadas
  - usam serviços providos por protocolos da camada inferior (TCP, UDP)



# Fundamentos de Sistemas Operacionais

## Aplicações de Rede

- Um processo é um programa que executa num hospedeiro.
- 2 processos no mesmo hospedeiro se comunicam usando comunicação entre processos definida pelo sistema operacional.
- 2 processos em hospedeiros distintos se comunicam usando um protocolo da camada de aplicação.
- Um agente de usuário (UA) é uma interface entre o usuário “acima” e a rede “abaixo”.
  - Implementa o protocolo da camada de aplicação
  - WWW: browser
  - Correio: leitor/compositor de mensagens
  - streaming audio/video: tocador de mídia

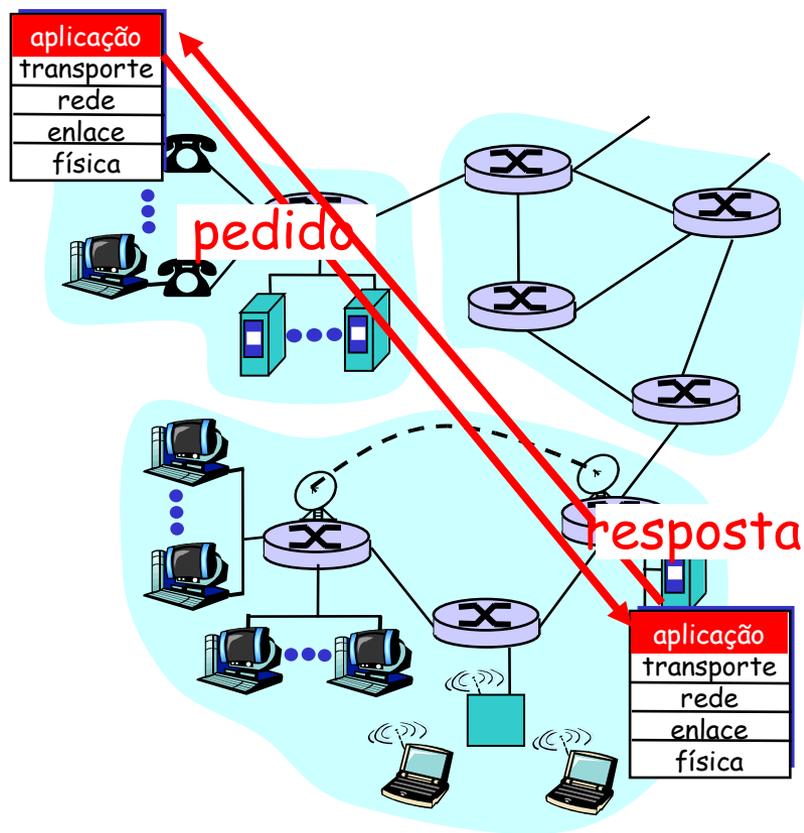
# Fundamentos de Sistemas Operacionais

## Paradigma Cliente-Servidor (C-S)

- Aplicação de rede típica tem duas partes: *Cliente* e *Servidor*
- Cliente:
  - inicia contato com o servidor (“fala primeiro”)
  - tipicamente solicita serviço do servidor
  - para WWW, cliente implementado no browser; para correio no leitor de mensagens
- Servidor:
  - provê ao cliente o serviço requisitado
  - p.ex., servidor WWW envia página solicitada; servidor de correio entrega mensagens

# Fundamentos de Sistemas Operacionais

## Paradigma Cliente-Servidor (C-S)



# Fundamentos de Sistemas Operacionais

## Protocolos da Camada de Aplicação (cont.)

- API: interface de programação de aplicações
  - define interface entre aplicação e camada de transporte
  - socket (tomada): API da Internet
    - ✓ 2 processos se comunicam enviando dados para um socket ou lendo dados de um socket
- Como um processo “identifica” o outro processo com o qual quer se comunicar?
  - endereço IP do hospedeiro do outro processo
  - “número de porta” - permite que o hospedeiro receptor determine a qual processo deve ser entregue a mensagem

## API Socket

---

- Desenvolvimento – Unix Berkeley BSD 4.3 por volta de 1980
- Independente do Protocolo de Transporte
  - TCP/IP por imposição de mercado
  - Socket's permitem acesso à famílias de protocolos de transporte

# Fundamentos de Sistemas Operacionais

## API Socket – Funções Básicas

- *Socket* – cria um descritor de *socket* para utilização na rede
- *Connect* – conecta o *socket* ao par(*peer*) remoto
- *Write* – envia mensagens
- *Read* – lê uma mensagem através de uma conexão
- *Close* – encerra a conexão e libera recursos
- *Bind* – “binda” uma porta e uma endereço IP a um socket
- *Listen* – coloca o *socket* no modo passivo e determina o tamanho da fila de espera de pedidos de conexão(servidor)
- *Accept* – aceita o próximo pedido de conexão na fila de entrada

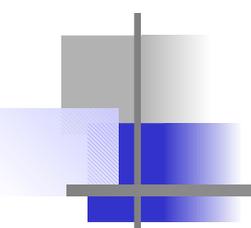
# Fundamentos de Sistemas Operacionais

## API Socket – Funções Básicas

---

- *Recv* – recebe a próxima mensagem
- *RecvFrom* – recebe a próxima mensagem e memoriza o endereço da fonte da mensagem
- *Send* – envia uma mensagem
- *SendTo* – envia uma mensagem para destino específico

# Fundamentos de Sistemas Operacionais



FIM