

Introdução

Laboratório de Sistemas Operacionais

Prof. MSc. João Tavares



JESUÍTAS BRASIL

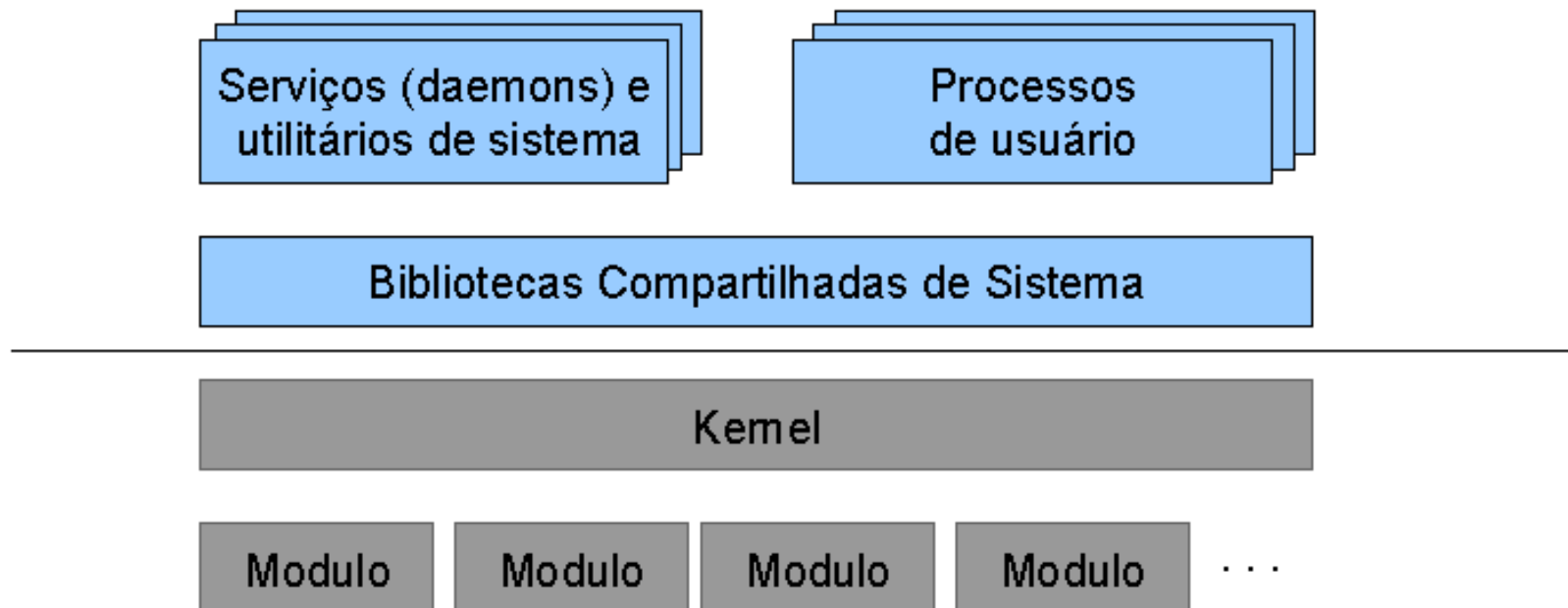


Somos infinitas possibilidades

GNU/Linux

- Sistema operacional Moderno e Livre **baseado no padrão Unix**
 - Estritamente falando, não é um Unix
- Kernel desenvolvido inicialmente por Linus Torvalds em 1991
- Evoluiu através da colaboração de diversas pessoas, distribuídas pelo mundo, através da Internet
- Executa muitos programas existentes no mundo Unix como os do projeto GNU (<http://www.gnu.org>)

Estrutura do Linux



Estrutura do Linux

- **Kernel** (núcleo): mantém as abstrações importantes e essenciais do sistema operacional
 - Código do kernel executa em **modo kernel** com acesso completo aos recursos físicos do computador
 - Todo código do kernel e suas estruturas de dados são mantidos no mesmo espaço de endereçamento

Estrutura do Linux

- **Bibliotecas do Sistema**: definem o conjunto padrão de funções através do qual os aplicativos interagem com o kernel
 - Também implementam muito da funcionalidade do SO que não demande os mesmos privilégios de execução do código do kernel
- **Serviços e Utilitários do Sistema**: realizam tarefas de gerenciamento individuais e especializadas

Estrutura do Linux

- **Módulos do Kernel** – seções do código do kernel que podem ser compiladas, carregadas e descarregadas independente do restante do kernel
 - Semelhantes a plugins
 - Um módulo tipicamente implementa drivers, sistemas de arquivos ou protocolos de rede
 - Módulos permitem que o sistema Linux seja configurado com um kernel padrão mínimo sem nenhum driver de dispositivo extra incluído

Interface de linha de comandos

- **Terminal** → disp. virtual de interface c/ o usuário
 - Ex.: janela
- **Shell** (interpretador de comandos) → atribui significado a sequencia de caracteres recebida do terminal e reage de acordo com o **comando selecionado**
 - Diversos shells estão disponíveis (bash, sh, csh, etc.)
- **Comando** é uma construção qualquer da **linguagem script** entendida pelo **shell**
 - **Extensibilidade**: comando pode ser implementado pelo próprio shell ou por outro programa que é disparado pelo shell

Comandos de ajuda

- **man** → exibe páginas de manual sobre um comando ou função

- Uso básico:

```
man [seção] tópico
```

- Obs: q sai, espaço avança página

- **help** → exibe informação sobre os comandos internos do interpretador de comandos
- **info** → semelhante ao **man**, porém com suporte a navegação estilo hipertexto

Páginas de manual

- **Seções** são identificadas por números:
 - 1 - **General Commands** - comandos gerais do shell
 - 2 - **System Calls** - chamadas de sistemas
 - 3 - **Subroutines** - funções de bibliotecas
 - 4 - **Special Files** - arquivos especiais
 - 5 - **File Formats** - formatos de arquivos
 - 6 - **Games** - jogos
 - 7 - **Macros and Conventions** - macros e convenções
 - 8 - **Maintenance Commands** - comandos de manutenção
 - 9 - **Kernel Interface** - interface do kernel

Alguns comandos...

echo → ecoa caracteres na saída padrão

who → mostra quem está logado

date → mostra data e hora

passwd → permite alterar a senha

cal → mostra o calendário

logout → encerra a sessão atual

Manipulação de arquivos

- ls** → mostra arquivos presentes no diretório atual (parâmetro -l)
- cat** → mostra o conteúdo de um ou mais arquivos
- wc** → conta o número de caracteres/ palavras/ linhas em um arquivo
- cp** → copia arquivos
- mv** → move/renomeia arquivos
- rm** → apaga arquivos

Manipulação de diretórios

pwd → mostra o diretório de trabalho corrente

mkdir → cria um sub-diretório

cd → troca o diretório corrente

rmdir → remove um diretório vazio

ln → cria links (vínculos)

du → mostra estatísticas sobre o uso do disco

. → denota diretório de trabalho (atual)

.. → denota diretório superior (pai)

Metacaracteres (caracteres coringa)

- Usado para abreviar nomes de arquivos

? - substitui um caractere

***** - substitui qualquer seqüência de caracteres

[ab] - define um conjunto de caracteres que pode ser utilizado na substituição

[a-b] - define uma faixa de caracteres que pode ser utilizada na substituição

Atributos de arquivos

-rw-rw-r--	1	pbg	staff	31200	Sep 3 08:30	intro.ps
drwx-----	5	pbg	staff	512	Jul 8 09:33	private/
drwxrwxr-x	2	pbg	staff	512	Jul 8 09:35	doc/
drwxrwx---	2	pbg	student	512	Aug 3 14:13	student-proj/
-rw-r--r--	1	pbg	staff	9423	Feb 24 2003	program.c
-rwxr-xr-x	1	pbg	staff	20471	Feb 24 2003	program
drwx--x--x	4	pbg	faculty	512	Jul 31 10:31	lib/
drwx-----	3	pbg	staff	1024	Aug 29 06:52	mail/
drwxrwxrwx	3	pbg	staff	512	Jul 8 09:35	test/

**modo do
arquivo**

UID

GID

tamanho

data

nome

número de links (hard links)

Modo de arquivo

- Primeiro caractere: **tipo de arquivo**
 - arquivo regular
 - d** diretório
 - b** dispositivo de bloco
 - c** dispositivo de caractere
 - l** link simbólico
 - s** link de socket (rede)
 - p** first-in first-out (FIFO) buffer, pipe nomeado

Modo de arquivo

- Demais caracteres: **permissão**

r	acesso de leitura (4)
w	acesso de escrita (2)
x	acesso de execução (1)
-	sem acesso
- ordem: dono, grupo e público

Modo de arquivo

- **chmod**: altera a permissão de um arquivo
 - escopo**: **u** (dono), **g** (grupo), **o** (outros)
 - ação**: **+** (adiciona), **-** (remove), **=** (determinada)
 - permissão**: **r** (leitura), **w** (escrita), **x** (execução)
 - ,** (separador)
- Ex:
 - `chmod u=rx,g-w,o+r file2`
 - `chmod 640 file`

Links de arquivos

- Links permitem apresentar um mesmo conteúdo em diferentes pontos do sistema de arquivos sem que haja duplicação
- O Unix oferece dois tipos de links:
 - **Hard links**: `ln`
 - Arquivos compartilham mesmo conteúdo
 - Somente dentro do mesmo sistema de arquivos
 - Não pode ser usado para diretórios
 - **Links simbólicos**: `ln -s`
 - Ponteiro para o caminho (nome) do arquivo, logo, pode apontar para algo que não existe
 - Pode ser usado entre diferentes sistemas de arquivos ou apontar para diretórios

Paginadores de texto

- **more**: mostra em páginas o conteúdo de um arquivo
- **less**: semelhante ao more, com possibilidade de movimento para baixo e para cima
- **head**: mostra as dez primeiras linhas de um arquivo
- **tail**: mostra as dez últimas linhas de um arquivo

Gerenciamento de processos

- **ps**: lista os processos em execução
- **kill**: termina um processo
- **top**: exibe os processos que mais utilizam a CPU em um determinado momento

Entrada/Saída (E/S)

- Cada processo em execução tem tipicamente 3 canais de interação com o meio externo
 - **Entrada padrão (0)**: teclado
 - **Saída padrão (1)**: tela
 - **Saída de erro padrão (2)**: tela
- Programas leem da entrada padrão e escrevem na saída padrão seus resultados ou então mensagens de erro na saída de erro padrão
- Os canais de entrada e saída de um processo podem ser redirecionados...

Redirecionamento de E/S

- É possível redirecionar a saída ou entrada de um processo **para arquivos**:
 - > redireciona a saída sobrescrevendo o arquivo
 - >> redireciona a saída anexando ao arquivo
 - < redireciona a entrada de um arquivo existente
 - 2> redireciona a saída de erro padrão
- Exemplo:
`ls -la > a.txt`

Pipes (|)

- Também é possível redirecionar a saída padrão de um processo para outro processo
- O operador | (pipe) provê essa funcionalidade
- Exemplo:
cat nomes | sort

Foco de Entrada

- Cada **terminal** pode ter um único comando em primeiro plano e múltiplos outros em segundo plano
- **Primeiro Plano** (Foreground)
 - Por padrão shell fica trancado até comando terminar execução.
 - Teclado fica reservado para a aplicação rodando em primeiro plano
- **Plano de Fundo** (Background)
 - Acrescentando "&" no final de um comando faz a execução ser concorrente com a do shell, liberando a execução do interpretador de comandos
 - Perde acesso ao teclado, mas pode escrever na tela

Dicas adicionais sobre o Shell

- Vários comandos na mesma linha
 - Separados por ";"
 - Compostos com operador "|"
- Autocompletar
 - A tecla "tab" permite que o shell termine de preencher automaticamente um caminho (nome de arquivo ou diretório) caso não existam conflitos

Na internet

- Leitura: A historia do Linux (e do Unix)

<https://www.vivaolinux.com.br/artigo/Historia-do-GNU-Linux-1965-assim-tudo-comecou/>

- Email de nascimento do Linux

<http://groups.google.com/group/comp.os.minix/msg/b813d52cbc5a044b?pli=1>

Referências Bibliográficas

- Material originalmente elaborado por Prof. Cristiano Costa. Material autorizado e cedido pelo autor. Revisado e atualizado por Prof. Luciano Cavalheiro e posteriormente pelo Prof. João Tavares.