



LINUX

AULA 1



HISTÓRICO: DO UNIX AO LINUX

ORIGEM DO UNIX

- Nos anos 60, AT&T, GE e MIT se juntam para o desenvolvimento do sistema Multics;
- Em 1969 a AT&T se desliga do projeto;
- Aproveitando o conhecimento adquirido, Ken Thompson e Dennis Ritchie dão início a um novo sistema;
- Em 1970 , nos laboratórios Bell Labs da AT&T, é criado o Unix.

ORIGEM DO UNIX

- Em paralelo à criação do Unix, uma nova linguagem de programação foi desenvolvida e, em 1973, a linguagem C fica pronta;
- O Unix é reescrito nesta nova linguagem de alto nível;
- Em C, o Unix pôde ser compilado em diferentes máquinas;
- Código fonte é livremente distribuído pela AT&T.

ORIGEM DO UNIX

- Surgimento de diversas variações do UNIX original baseado em duas principais fontes:
 - System V (AT&T) e BSD (Berkeley).
- Alguns Unix comerciais:
 - AIX (IBM),
 - SunOS e Solaris (Sun),
 - HP-UX (HP),
 - Irix (SGI),
 - SCO (SCO);

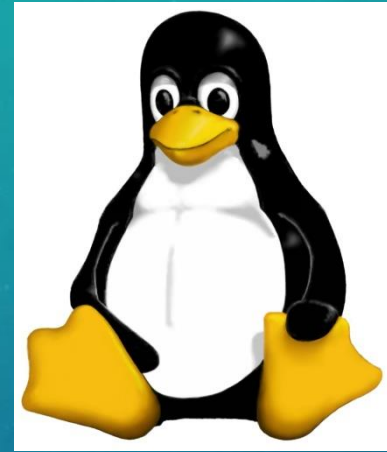
POSIX

- Falta de padronização não permitia que aplicações rodassem em sistemas Unix diferentes;
- Padronização criada pelo IEEE permitiu a construção de sistemas Unix compatíveis;
- O POSIX (Portable Operating System Interface) especifica como as aplicações interagem com o sistema.

ORIGEM DO LINUX

- No início da década 90, universitários usavam o limitado sistema **Minix** em suas máquinas pessoais;
- Linus Torvalds, um estudante finlandês, decide criar seu próprio sistema operacional;
- Surge então o Linux, com código fonte livremente distribuído na Internet;
- Muitos desenvolvedores contribuíram com o desenvolvimento do novo sistema.

O LINUX



- Atualmente o Linux continua em pleno desenvolvimento;
- Novas versões são lançadas a cada mês;
- Muitos voluntários contribuem para o desenvolvimento do kernel;
- O sistema também segue o padrão POSIX;

O PROJETO GNU



- Como forma de incentivar o desenvolvimento de softwares livre, Richard Stallman criou o projeto GNU.
- GNU é um acrônimo recursivo de “GNU’s Not Unix”.
- O projeto GNU prevê um sistema operacional e inúmeras ferramentas que são fornecidas com o código fonte;
- O projeto teve grande impulso com o surgimento do kernel Linux;

LICENÇAS GNU



- **GPL** : GNU Public License. Utilizada para licenciar softwares livres;
- **LGPL** : Lesser GNU Public License. Utilizada para licenciar bibliotecas livres;
- **FDL** : GNU Free Documentation License. Utilizada para licenciar documentações de softwares livres.

DISTRIBUIÇÕES LINUX

- As distribuições são criadas por empresas ou organizações que empacotam o Linux;
- Uma distribuição é composta pelo kernel do Linux e inúmeras ferramentas;
- As distribuições diferem entre si pela versão do kernel, processo de instalação e conjunto de ferramentas fornecidas;
- Nada impede que uma distribuição Linux seja vendida.

DISTRIBUIÇÕES LINUX

- Muitas distribuições surgiram atendendo às mais diferentes necessidades.
- **Debian**
 - Desenvolvida por voluntários;
 - Possui mais de 8 mil pacotes;
 - Só possui ferramentas livres;
 - Distribuição oficial do projeto GNU.

DISTRIBUIÇÕES LINUX

- **Slackware**
 - Muito popular na comunidade open source;
 - Não possui uma empresa por trás;
 - Visa aproximar mais o sistema de um ambiente UNIX-like.
- **SuSE**
 - Distribuição orgiginária da Alemanha;
 - Comprada pela Novell, é compatível com os produtos oferecidos pela empresa;

DISTRIBUIÇÕES LINUX

- **Red Hat**
 - Distribuição comercial;
 - Focada no mercado corporativo;
 - Fornece soluções de missão crítica.
- **Fedora**
 - Linha open source e gratuita da Red Hat;
 - Não possui suporte da empresa.
- **CentOS**
 - Baseado na distribuição Red Hat;
 - Não possui suporte da empresa.

CARACTERÍSTICAS DO LINUX

- Multitarefa e Multiusuário;
- Compatível com várias arquiteturas (intel, powerpc, sparc, alpha e arm);
- Compatibilidade com muitos protocolos de rede;
- Novas versões são lançadas constantemente (em torno de 1 mês);
- Forte controle de acesso dos usuários.

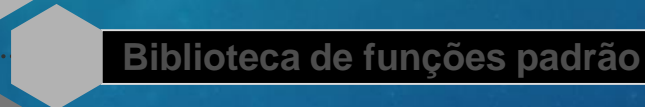
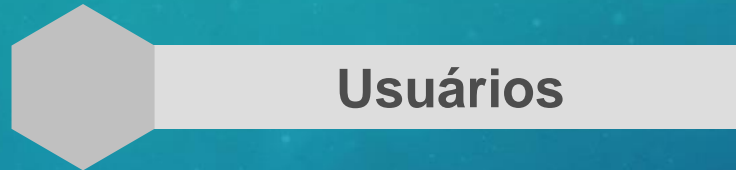


ORGANIZAÇÃO DO SISTEMA OPERACIONAL

SISTEMA OPERACIONAL

- O Sistema Operacional é uma camada de software que faz a interface entre as aplicações e o hardware.
- É responsável por gerenciar o uso dos dispositivos de entrada e saída de modo a compartilhá-los entre as aplicações de forma eficiente.
- Tarefas comuns:
 - Organizar os dados em arquivos, gerenciar o uso do processador, gerenciar a memória etc.

ORGANIZAÇÃO DO UNIX/LINUX



HARDWARE

- O hardware é composto pelos recursos computacionais;
- São os dispositivos de entrada e saída, barramentos, processador(es) e memória;
 - Monitor, Disco rígido, drive de cd-rom, controladoras de disco, barramento USB, impressora, placa de rede etc.
- Um conjunto de hardware com especificações próprias é chamado de arquitetura (ex: Intel);

KERNEL

- O Kernel é o núcleo do sistema operacional;
- Responsável por fazer o gerenciamento e a interface entre as aplicações e o hardware;
- Funções
 - Detecção de hardware;
 - Gerenciamento de entrada e saída;
 - Manutenção do sistema de arquivos;
 - Gerenciamento de memória e swapping;
 - Controle da fila de processos (escalonamento).

APLICAÇÕES

- As aplicações são os softwares que usam os componentes de hardware para executar uma tarefa útil para o usuário;
- As aplicações podem ser:
 - **Em modo console;**
 - **Daemons (Serviços);**
 - **Gráficas;**
 - **Web**

USUÁRIOS

- Os usuários são os agentes que irão fazer uso dos recursos computacionais;
- Qualquer tarefa executada no Linux pertence a um usuário;
- O sistema de usuários é fundamental para o controle de uso dos recursos;
- O usuário administrador é chamado de **root**;
- Use o **root** somente para tarefas administrativas.



INTERFACE DE CARACTERES

O QUE É O SHELL?

- O shell é um programa interpretador de comandos;
- É responsável por:
 - Ler os comandos do teclado (entrada padrão);
 - Interpretar os comandos;
 - Criar novos processos em resposta.

Exemplo

```
[einstein@physic.edu /home]$ ls  
copernico  galileu  einstein  newton  gauss  
[einstein@physic.edu /home]$
```

Comando

Saída
processo "ls"

INTERAÇÃO COM O SHELL

- Existem dois modos de interagir com o shell:
- **Modo interativo:**
 - Usuário digita o comando;
 - O shell os interpreta após ser pressionada a tecla Enter.
- **Modo não-interativo**
 - O shell recebe um arquivo de script;
 - O arquivo contém vários comandos em sequência para serem executados.

HISTÓRICO DOS SHELLS

- **Bourne Shell (sh)**
 - Criado por Stephen Bourne no Bell Labs no final da década de 70;
 - Possui uma linguagem de programação mas era ineficiente para usuários interativos;
- **C shell (csh)**
 - Suporte a jobs, alias, e uma nova linguagem com sintaxe similar ao C.
 - Primeiras versões com muitos bugs.
- **Trusted C shell (tcsh)**
 - Originou-se para resolver os bugs do csh. Suporte a edição de comandos de linha.
- **Korn shell (ksh)**
 - Criado na AT&T, possuía todas as vantagens do tcsh com a linguagem de scripts originária do sh. Não era gratuito.
- **Bourne Again Shell (bash)**
 - Shell originário do sistema operacional GNU sob a licença GPL;
- **Outros shells**
 - rc, es, zsh

BASH

- Bash é o shell criado pelo projeto GNU;
- Acrônimo de Bourne Again SHell;
- É compatível com a especificação POSIX.2;
- Possui funcionalidades incorporadas do ksh e do csh;
- O Bash é um dos shells mais completos existentes hoje e é o shell padrão do Linux.

LOGIN

- O Login é o processo de autenticação do usuário;
- O usuário deverá informar um nome de usuário e uma senha válida;
- Se os dados fornecidos estão corretos, o sistema irá abrir o shell do usuário;

```
Trustix Secure Linux release 2.2
Kernel 2.6.5-1
physic.edu login: galileu
Password:
```


LOGOUT

- O processo de encerramento da sessão do bash é chamado de logout;
- Você pode sair do bash das seguintes formas:
 - Comando `logout`;
`logout`
 - Comando `exit`;
`exit`
 - Teclas `Ctrl+d`;

USANDO O BASH

- O shell é identificado por um aviso de comando ou prompt.

```
[einstein@physic.edu /home]$
```

- **einstein**: Nome do usuário corrente;
 - **physic.edu**: Nome da máquina;
 - **/home**: Diretório corrente;
 - **\$**: Símbolo que indica que é um usuário comum.
- O prompt é definido por dois símbolos:
 - **#**: Shell de root
 - **\$**: Shell de usuário comum

USANDO O BASH

- Histórico
 - O Bash é capaz de armazenar o histórico dos comandos utilizados, mesmo em outras sessões
 - seta para cima e seta para baixo;
- Rolagem da tela
 - Você pode rolar a tela para visualizar um conteúdo que já foi exibido
 - shift+page up: tela para cima
 - shift+page down: tela para baixo;

TERMINAL

- Um terminal é um conjunto teclado+tela;
- O Linux faz uso de terminais virtuais;
 - Terminais virtuais são a simulação de vários terminais a partir de apenas um;
 - Cada terminal virtual é independente dos outros;
 - Para acessá-los quando logados no terminal pressione as teclas Alt+F1, Alt+F2, ..., Alt+F6.
 - Para acessá-los quando logados no Ambiente Gráfico pressione as teclas Ctrl+Alt+F1, Ctrl+Alt+F2, ..., Ctrl+Alt+F6.
 - O terminal gráfico é acessado no terminal 7 Alt+F7



MANIPULAÇÃO DE ARQUIVOS E DIRETÓRIOS



MANIPULANDO DIRETÓRIOS

LISTANDO O DIRETÓRIO (LS)

- Para listar o conteúdo de um diretório usamos o comando `ls` (*list system*).

Sintaxe

```
ls [opcoes...] [arquivo ou diretório...]
```

Exemplo

```
[gauss@physic.edu gauss]$ ls  
docs                festa1.jpg          manual.pdf          relatorios  
declaracao.rtf      festa2.jpg          palm
```

LISTAGEM LONGA (LS -L)

- Para exibir uma listagem longa do diretório utilize o parâmetro “-l”

Exemplo

```
[gauss@physic.edu /]$ ls -l /var/log
total 647
-rw----- 1 root root 321220 Feb 28 11:43 secure
drwxr-xr-x 2 root root 4096 Oct 11 02:59 mail/
-rw----- 1 root root 54476 Feb 28 11:43 messages
-rw-rw-r-- 1 root utmp 271872 Feb 28 11:43 wtmp
```

Permissões

Links

Usuário
Dono

Grupo
Dono

Tamanho

Data

Nome

LISTAR ARQUIVOS OCULTOS (LS -A)

- Arquivos com nome iniciado pelo caractere “.” são considerados arquivos ocultos;
- Eles podem ser vistos com o comando “ls -a”;

Exemplo

```
[gauss@physic.edu /]$ ls -aF /var/log
total 647
./          cron          messages      spooler
../         cron.1        mysqld.log    spooler.1
acpid       cups/         news/         squid/
```

Arquivos Ocultos

DIRETÓRIOS OCULTOS

- “./” : Referencia o diretório corrente do usuário;
- “../” : Referencia o diretório imediatamente anterior ao diretório corrente;

Exemplo

```
[gauss@physic.edu /proc/sys]$ cd ..  
[gauss@physic.edu /proc]$ cd ../etc/init.d  
[gauss@physic.edu /etc/init.d]$ ./firewall  
[gauss@physic.edu /etc/init.d]$ ls ..
```


MUDANDO O DIRETÓRIO (CD)

- Para mudar o diretório corrente utilize o comando `cd` (*change directory*).

Sintaxe

```
cd <dir>
```

Exemplo

```
[gauss@physic.edu gauss]$ cd /  
[gauss@physic.edu /]$
```

DIRETÓRIO CORRENTE (PWD)

- Para exibir o diretório corrente utilize o comando `pwd` (*print working directory*)

Sintaxe

```
pwd
```

Exemplo

```
[gauss@physic.edu pub]$ pwd  
/var/ftp/pub  
[gauss@physic.edu pub]$
```

CRIANDO UM DIRETÓRIO (MKDIR)

- Para criar um diretório vazio use o comando mkdir (make directory).

Sintaxe

```
mkdir [opcoes] <diretorio[...]>
```

Exemplo

```
[gauss@physic.edu pub]$ mkdir papers  
[gauss@physic.edu pub]$
```

REMOVENDO DIRETÓRIO (RMDIR)

- Para remover um diretório vazio utilize o comando `rmdir` (remove directory).

Sintaxe

```
rmdir [opcoes] <diretorio[...]>
```

Exemplo

```
[gauss@physic.edu pub]$ rmdir docs pesquisas  
[gauss@physic.edu pub]$
```

Removerá os diretórios
docs/ e pesquisas/

LAB

- Faça os exercícios práticos abaixo:

1. Crie os diretórios:

```
~/arquivos/  
~/arquivos/docs/  
~/arquivos/scripts/  
~/arquivos/relatorios/
```

2. Qual é o arquivo com maior tamanho no diretório /var/log?