

Instituto Federal de Educação, Ciência e Tecnologia do Piauí-IFPI

Curso de Tecnologia em Análise e Desenvolvimento de Sistemas

Campus Central- Prof.º Ricardo Ramos

Disciplina de Introdução à Computação

Aluno: Vinícius Gomes Araújo Costa

Comandos do Linux

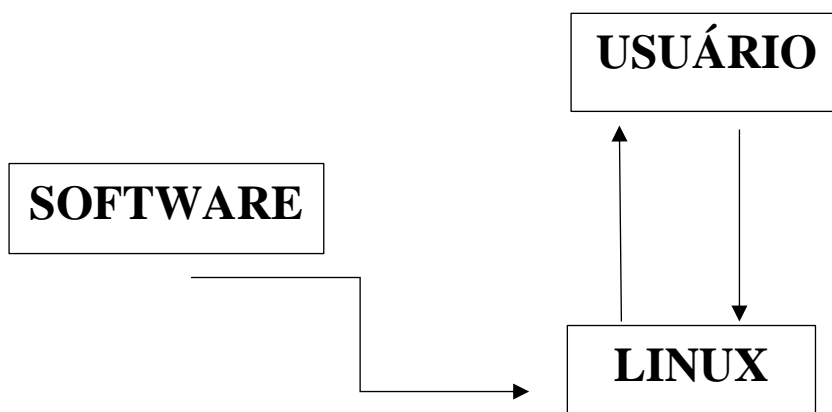
Teresina, 16 de Julho;

2019.

Introdução de Linux

O “terminal Linux” é um dos principais aspectos do Sistema Operacional, sendo este, muitas vezes, motivo de temor por parte dos iniciantes acostumados a interagir com interfaces gráficas. Para quem almeja tornar-se um excelente profissional de T.I, é necessário o conhecimento do terminal, pois, sem dúvidas, Estrutura de Redes e Linguagens de Programação irá requerer um prévio momento de participação mútua de usuário-máquina como pré-requisito para quem deseja ser um bom profissional na área de Tecnologia da Informação.

Diagrama de Caso:



Legenda do Diagrama de Caso:

A interpretação deste diagrama personifica a ideologia introdutória do terminal Linux. Ou seja, o software é o próprio terminal do Linux, que intermedeia a interação com o Sistema Operacional Linux por meio do shell. Um programa que recebe os comandos do usuário a partir do teclado. O usuário(que poderá ser eu, você ou um especialista em T.I), executa o shell e propõe a updata(LINUX-USUÁRIO) e, por conseguinte, a downdata(USUÁRIO-LINUX) por utilização do seus serviços.

A posteriori, o usuário, que executa o shell, é um instrumento de comunicação que utiliza o S.O(Sistema Operacional) na update; e, o Linux, diferenciado; permite utilizar os componentes do microcomputador por meio de comandos bem específicos. Outrossim, a downdata. Finalizando assim, o sistema de comunicação homem-máquina de forma calculada, programada e exata para a existência de um S.O bem dinâmico e ininteligível.

Desenvolvimento do trabalho

Entendendo a estrutura de diretórios do Linux:

Você seguiu nosso tutorial de como escolher uma distro Linux ou de como instalar o Ubuntu em uma máquina virtual, sem precisar remover o Windows ou o OS X do seu computador. Mas bastaram alguns minutos usando o sistema para perceber algo muito curioso: não existe "C:/", "D:/" ou "System32", como no Windows. Em vez disso, os arquivos estão armazenados em pastas com nomes bem diferentes.

A boa notícia é que, apesar desse estranhamento inicial, a estrutura de diretórios do Linux é muito bem definida e, ao ler este artigo, você perceberá que ela está organizada em termos lógicos e que fazem muito sentido.

Para manter a organização, desenvolvedores de distribuições Linux e softwares diversos seguem o *Filesystem Hierarchy Standard* (padrão para sistema de arquivos hierárquico), ou *FHS*, uma espécie de referência que padroniza quais pastas do sistema recebem determinados tipos de arquivo.

Mais precisamente, para a elaboração deste trabalho de Linux, é preciso saber que a EXT(Extended Filesystem), em parceria com a EXT2(Extended Filesystem 2) foram, com o tempo, a evolução e criação do Sistema de arquivos Minix, do journaling; que deu origem ao sistema atualizado de hoje, mais conhecido como Linux. Além disso, a EXT3(Extended Filesystem) implementa este conceito de journaling. Ademais, o sistema EXT3 journaling grava qualquer operação que seja feita no disco em uma área especial chamada journal, assim, se acontecer algum problema durante a operação de disco, ele pode voltar ao estado anterior do arquivo, ou finalizar a operação.

O diretório raiz (/)

Todos **os arquivos e diretórios do sistema Linux instalado no computador** partem de uma única origem: **o diretório raiz**. Mesmo que estejam armazenados em outros dispositivos físicos, é a partir do diretório raiz – representado pela barra (/) – que você poderá acessá-los.

Também vale lembrar que o único **usuário** do sistema capaz de criar ou mover arquivos do diretório raiz é o **root**, ou seja, o **usuário-administrador**. **Isso evita que usuários comuns cometam erros e acabem comprometendo a integridade de todo o sistema de arquivos.**

Binários executáveis: /bin

No diretório /bin estão localizados os **binários executáveis** que podem ser utilizados por qualquer usuário do sistema. São comandos essenciais, usados para trabalhar com arquivos, textos e alguns recursos básicos de rede, como o **cp**, **mv**, **ping** e **grep**.

Binários do sistema: /sbin

Assim como o /bin, este **diretório** armazena executáveis, mas com um diferencial: são aplicativos utilizados por administradores de sistema com o propósito de realizar funções de manutenção e outras tarefas semelhantes. Entre os comandos disponíveis estão o **ifconfig**, para **configurar e controlar interfaces de rede TCP/IP**, e o **fdisk**, que **permite particionar discos rígidos, por exemplo**.

Programas diversos: /usr

Se você não encontrar um comando no diretório /bin ou /sbin, ele certamente está aqui. O /usr reúne executáveis, bibliotecas e até documentação de softwares usados pelos usuários ou administradores do sistema. Além disso, sempre que você compilar e instalar um programa a partir do código-fonte, ele será instalado nesse diretório.

Configurações do sistema: /etc

No diretório /etc ficam arquivos de configuração que podem ser usados por todos os softwares, além de scripts especiais para iniciar ou interromper módulos e programas diversos. É no /etc que se encontra, por exemplo, o arquivo *resolv.conf*, com uma relação de servidores DNS que podem ser acessados pelo sistema, com os parâmetros necessários para isso.

Bibliotecas: /lib

Neste ponto do sistema de arquivos ficam localizadas as bibliotecas usadas pelos comandos presentes em /bin e /sbin. Normalmente, os arquivos de bibliotecas começam com os prefixos *ld* ou *lib* e possuem "extensão" *so*.

Opcionais: /opt

Aplicativos adicionais, que não são essenciais para o sistema, terminam neste diretório.

Arquivos pessoais: /home

No diretório /home ficam os arquivos pessoais, como documentos e fotografias, sempre dentro de pastas que levam o nome de cada usuário. Vale notar que o diretório pessoal do administrador não fica no mesmo local, e sim em /root.

Inicialização: /boot

Arquivos relacionados à inicialização do sistema, ou seja, o processo de boot do Linux, quando o computador é ligado, ficam em /boot.

Volumes e mídias: /mnt e /media

Para acessar os arquivos de um CD, pendrive ou disco rígido presente em outra máquina da rede, é necessário "montar" esse conteúdo no sistema de arquivos local, isso é, torná-lo acessível como se fosse apenas mais um diretório no sistema.

Em /media ficam montadas todas as mídias removíveis, como dispositivos USB e DVDs de dados. Já o diretório /mnt fica reservado aos administradores que precisam montar temporariamente um sistema de arquivos externo.

Serviços: /srv

Dados de servidores e serviços em execução no computador ficam armazenados dentro desse diretório.

Arquivos de dispositivos: /dev

No Linux, tudo é apresentado na forma de arquivos. Ao plugar um pendrive no computador, por exemplo, um arquivo será criado dentro do diretório /dev e ele servirá como interface para acessar ou gerenciar o drive USB. Nesse diretório, você encontra caminhos semelhantes para acessar terminais e qualquer dispositivo conectado ao computador, como o mouse e até modems.

Arquivos variáveis: /var

Todo arquivo que aumenta de tamanho ao longo do tempo está no diretório de arquivos variáveis. Um bom exemplo são os logs do sistema, ou seja, registros em forma de texto de atividades realizadas no Linux, como os logins feitos ao longo dos meses.

Processos do sistema: /proc

Lembra da história de que tudo funciona como um arquivo no Linux? Pois o /proc é a prova disso. Nesse diretório são encontrados arquivos que revelam informações sobre os recursos e processos em execução no sistema. Quer um exemplo? Para saber há quanto tempo o Linux está sendo usado desde a última vez em que foi iniciado, basta ler o arquivo /proc/uptime.

Arquivos temporários: /tmp

Arquivos e diretórios criados temporariamente tanto pelo sistema quanto pelos usuários devem ficar nesse diretório. Boa parte deles é apagada sempre que o computador é reiniciado.

Como fica fácil perceber, os nomes dos diretórios dão dicas do que pode ser encontrado em seu interior e, com alguns meses de uso, você estará navegando por eles com facilidade.

Este artigo faz parte de nossa biblioteca de conteúdo "Tudo o que você precisa saber sobre o Linux". Não deixe de acessar e conferir todo o conteúdo publicado sobre o Pinguim.

Depois de conhecer bem **a estrutura de diretórios do Linux**, chegou o momento de se aventurar com alguns dos comandos mais utilizados do sistema. A princípio, pode parecer bobagem, visto que hoje quase todas as operações podem ser feitas **na interface gráfica do sistema** e muitas **distros** fornecem assistentes até mesmo para a configuração de servidores. Mas saiba que há muitas vantagens no uso do modo console ou de um emulador de terminal.

O modo texto do Linux oferece mais liberdade ao usuário, que pode abusar dos parâmetros de cada comando, executando ações muito personalizadas. Além disso, depois de se acostumar com o uso do sistema por esse meio, você perceberá a agilidade que o teclado oferece, principalmente ao conhecer teclas de atalho, variáveis e outros truques permitidos pelo terminal.

Como se não bastasse, aprender a trabalhar bem com os comandos em modo texto é entender um pouco dos bastidores do Linux e o que se passa por trás dos assistentes gráficos que facilitam algumas tarefas do sistema. Assim, quando esses softwares usados com o mouse não derem conta do recado, você poderá contornar as limitações rapidamente.

Nesta lista estão dez comandos básicos e úteis utilizados à exaustão não apenas por aqueles que chegaram agora no mundo do Linux, mas também por usuários experientes. Note que a intenção deste artigo não é explicar extensivamente o uso de cada um dos comandos, mas fornecer meios para que iniciantes possam conhecê-los e, posteriormente, caminhar sozinhos em busca dos seus próprios interesses e uso.

1. man: aprenda mais sobre os comandos

O comando man deveria estar no topo de toda lista de comandos importantes do Linux. A razão é muito simples: basta executá-lo para carregar uma página de manual (man page) sobre os comandos do sistema, com definições não apenas do uso de cada ferramenta, mas também descrições detalhadas dos inúmeros parâmetros do software e exemplos de uso.

Manpage

Para ler as man pages é muito fácil e basta executar o man seguindo do nome do comando que você deseja buscar ajuda. Não se esqueça de pressionar a tecla Enter após digitar o comando, caso contrário ele não será executado.

Ao executar man cp, por exemplo, você poderá ler todas as instruções para o uso do comando cp. E antes de passarmos para o próximo item da lista, valem duas dicas: a primeira é que existe o comando man man, para o caso de dúvidas sobre o uso do próprio man. A segunda é que é possível traduzir o conteúdo das páginas de manual para o português, caso estejam em inglês, instalando o manpages-pt pela Central de Programas do Ubuntu.

2. ls: listagem de arquivos

Para listar os arquivos existentes em algum diretório, basta usar o comando `ls`. Se executado sem parâmetros, ele listará o conteúdo do diretório em que você se encontra. Mas você pode indicar um caminho para ele, como `ls /usr/bin`, por exemplo.

Também é possível usar o `ls` para conferir o tamanho e a data de criação de cada arquivo ou pasta. Para isso, use o parâmetro `-lh`, como no exemplo a seguir: `ls -lh`. E se você também quiser listar os arquivos ocultos, que começam com um ponto, use a opção `-a` (`ls -lha`).

Listagem de arquivos no Linux:

3. cd: navegue pelo sistema de arquivos

Para pular de pasta em pasta, não precisa abrir o gerenciador de arquivos. No próprio terminal é possível navegar pelo sistema de arquivos usando o comando `cd` seguido do caminho que você deseja seguir. Exemplo: `cd /usr`.

Vale a pena notar que existem alguns atalhos que podem facilitar a vida do usuário. Se você executar o comando `cd` sem parâmetros, ele retorna para a pasta do usuário, localizada em `/home`. Para voltar um nível acima na árvore de diretório, use `"cd .."`, sem as aspas. Dessa forma, se você estiver em `/usr/bin` e executar `"cd .."`, voltará para o diretório `/usr`.

4. cp: copiar arquivos e pastas

Copiar um arquivo pelo terminal também é moleza. Use o comando `cp` seguindo do arquivo de origem e o destino para ele, que pode ser tanto uma nova pasta quando um novo arquivo, com nome diferente. Exemplo: `cp arquivo1.txt arquivo2.txt` ou, então, `cp arquivo1.txt pastanova/`.

Para copiar um diretório todo, não se esqueça de inserir o parâmetro `-r`. Se quiser clonar uma pasta, use `cp -r pasta1 pasta2`, por exemplo.

5. mv: mover arquivos e pastas

Para mover arquivos existe o comando `mv` e ele pode ser usado tanto para remanejar arquivos como para renomeá-los. Se quiser enviar o arquivo de uma pasta para outra, basta seguir o exemplo `mv pasta1/arquivo1 pasta2/`. Se preferir apenas renomeá-lo, use `mv arquivo1 arquivo2`.

6. more: ler arquivos de texto

Caso você precise ler o conteúdo de um arquivo de texto, use o comando `more` seguido do caminho e nome do arquivo, como em `more /home/user/arquivo.txt`.

Todo conteúdo do arquivo será exibido no terminal, preenchendo a tela com texto. Para prosseguir com a leitura, pressione a barra de espaço e, caso precise voltar uma ou mais páginas, use a tecla `"b"`. Se quiser sair antes do fim do arquivo, pressione `"q"`.

7. df: conferir o espaço em disco

Quer saber qual é o espaço total e quantos GB disponíveis existem em cada partição do sistema? Use o comando `df -h`. A opção `-h`, aliás, quer dizer human-readable, ou seja, legível para humanos. Se você executar o comando sem ela, as informações serão exibidas em kilobytes e será necessário convertê-las mentalmente para outras unidades.

8. sudo: permissões especiais

Por razões de segurança, o Linux trabalha com permissões de usuários. Por isso, determinados comandos ou arquivos são acessíveis apenas pelo próprio dono ou pelo usuário administrador (root). Para que você não tenha que trocar de usuário a todo instante, existe o comando `sudo`, que garante credenciais de usuário root temporariamente, mediante a informação de uma senha.

Para fazer o teste, tente executar o comando `ls /root`. Você receberá um aviso de permissão negada. Em seguida, execute `sudo ls /root`. Depois de informar a senha do seu próprio usuário (no caso do Ubuntu), o comando será executado normalmente e os arquivos a pasta root serão listados no terminal.

9. grep: buscas em textos

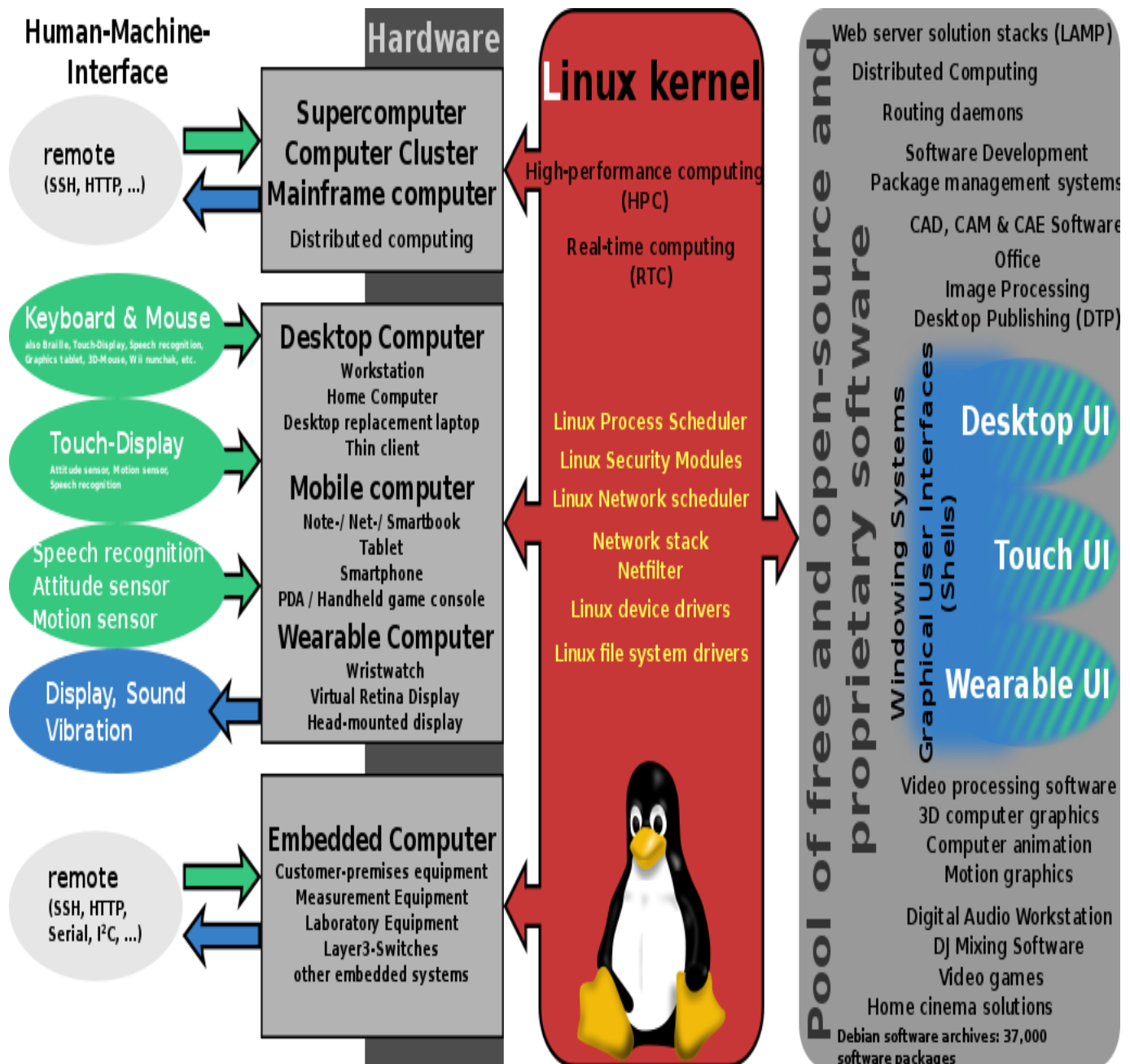
Imagine a seguinte situação: você tem um arquivo de texto com cerca de 200 nomes de alunos de certa escola, mas não tem certeza se um nome em específico está listado. O `grep` ajuda você a procurar por esse aluno e a fazer muito mais com a ajuda de expressões regulares.

Bastaria executar o comando `grep "Nome do Aluno" arquivo.txt` para que o terminal busque pelo nome indicado dentro da relação. Caso você não tenha certeza se o nome do aluno foi escrito respeitando as letras maiúsculas, adicione o parâmetro `-i` para que `grep` passe a ignorar essa distinção durante a busca.

10. clear: limpar o buffer

Por último, um comando que ajuda a organizar um pouco a confusão de letras que ficam no terminal depois de horas de uso. Para limpar toda a tela, execute o comando `clear`. Depois, é só voltar a usar o terminal normalmente, como se nada tivesse acontecido.

Este artigo faz parte de nossa biblioteca de conteúdo "Tudo o que você precisa saber sobre o Linux". Não deixe de acessar e conferir todo o conteúdo publicado sobre o Pinguim.



abhishek@itsfoss: ~

```
*      ,d8888b.      888      *
```

```
*      d88P  Y88b      888      *
```

```
*      Y88b.      888      *
```

```
* 888888b.  Y888b.  888888b.  88888b. 888 888 .d88b.      *
```

```
* 888 88b      8Y88b. 888 88b      88b 888 .88P d8P Y8b      *
```

```
* 888 888      888b 888 888 .d888888 888888K 88888888      *
```

```
* 888 888 Y88b d88P 888 888 888 888 888 88b Y8b.      *
```

```
* 888 888 Y8888P 888 888 Y888888 888 888 Y88888 v1.5      *
```

Game Mode: Borders On
 Borders Off

```
* Use --help for guidelines
```

```

-rw-r--r-- 1 ppinto ppinto 68 2007-01-31 22:36 teste.c
-rw-r--r-- 1 ppinto ppinto 55 2007-01-31 22:35 teste.c~
-rw-r--r-- 1 ppinto ppinto 12288 2007-01-31 22:35 .teste.c.swp
-rw-r--r-- 1 ppinto ppinto 27 2007-01-31 22:07 teste.txt
drwx----- 2 ppinto ppinto 4096 2006-12-23 22:24 .trash

```

Nome do ficheiro / directório

Data/hora de modificação

Tamanho do ficheiro (em bytes)

Grupo a que pertence o ficheiro/Directório

"Dono" do ficheiro / directório

Nº de links existentes no sistema para o ficheiro ou directório

Permissões do ficheiro/directório (permissões para user, grupo e others)

Indica se é ficheiro ou directório ('-' ou 'd')

Referências Bibliográficas

Link de acesso às referências do contexto in LINUX, disponível em
<https://www.youtube.com/watch?v=uZeMQz89pfw&t=36s>

Link de acesso aos comandos essenciais de como usar o software, disponível em
<https://canaltech.com.br/linux/10-comandos-essenciais-do-linux/>

Link de acesso para estudo do S.O <https://www.devmedia.com.br/comandos-importantes-linux/23893>

Site para fontes acadêmicas e afins de pesquisa de trabalho, disponível em
<https://blog.profissionaislinux.com.br/linux/terminal-linux/>