

# Unix

Origem: Wikipédia, a enciclopédia livre.

**Unix** é um sistema operativo (ou sistema operacional) portátil (ou portável), multitarefa e multiutilizador (ou multiusuário) originalmente criado por Ken Thompson, Dennis Ritchie, Douglas McIlroy e Peter Weiner, que trabalhavam nos Laboratórios Bell (Bell Labs) da AT&T. A marca **UNIX** é uma propriedade do The Open Group, um consórcio formado por empresas de informática.<sup>1</sup>

## Índice

- 1 História
- 2 Características
  - 2.1 Sistema operacional multitarefa
  - 2.2 Sistema operacional multiutilizador
  - 2.3 Arquivos de dispositivo
- 3 Estrutura
  - 3.1 Processos
  - 3.2 Sistema de arquivos
    - 3.2.1 Estrutura de diretórios
- 4 Particularidades
- 5 Aplicações
- 6 Ambiente gráfico do Unix
  - 6.1 X Window System
- 7 Comandos
  - 7.1 Comandos de manipulação de diretório
  - 7.2 Comandos para manipulação de arquivos
  - 7.3 Comandos para administração
  - 7.4 Comandos para administração de rede
- 8 Notas e referências
- 9 Ver também
- 10 Ligações externas

## Unix



<b>Modelo</b>	Código fechado, agora alguns projectos Unix (BSD família e Open Solaris) são de código aberto.
<b>Lançamento</b>	1969
<b>Arquitetura(s)</b>	x86, x64
<b>Página oficial</b>	<a href="http://www.unix.org">www.unix.org</a> ( <a href="http://www.unix.org">http://www.unix.org</a> )

## História

Em 1965 formou-se um grupo de programadores, incluindo Ken Thompson, Dennis Ritchie, Douglas McIlroy e Peter Weiner, num esforço conjunto da *AT&T (Laboratórios Bell)*, da *General Electric (GE)* e do *MIT (Massachusetts Institute of Technology)* para o desenvolvimento de um sistema operacional chamado Multics.<sup>2</sup>

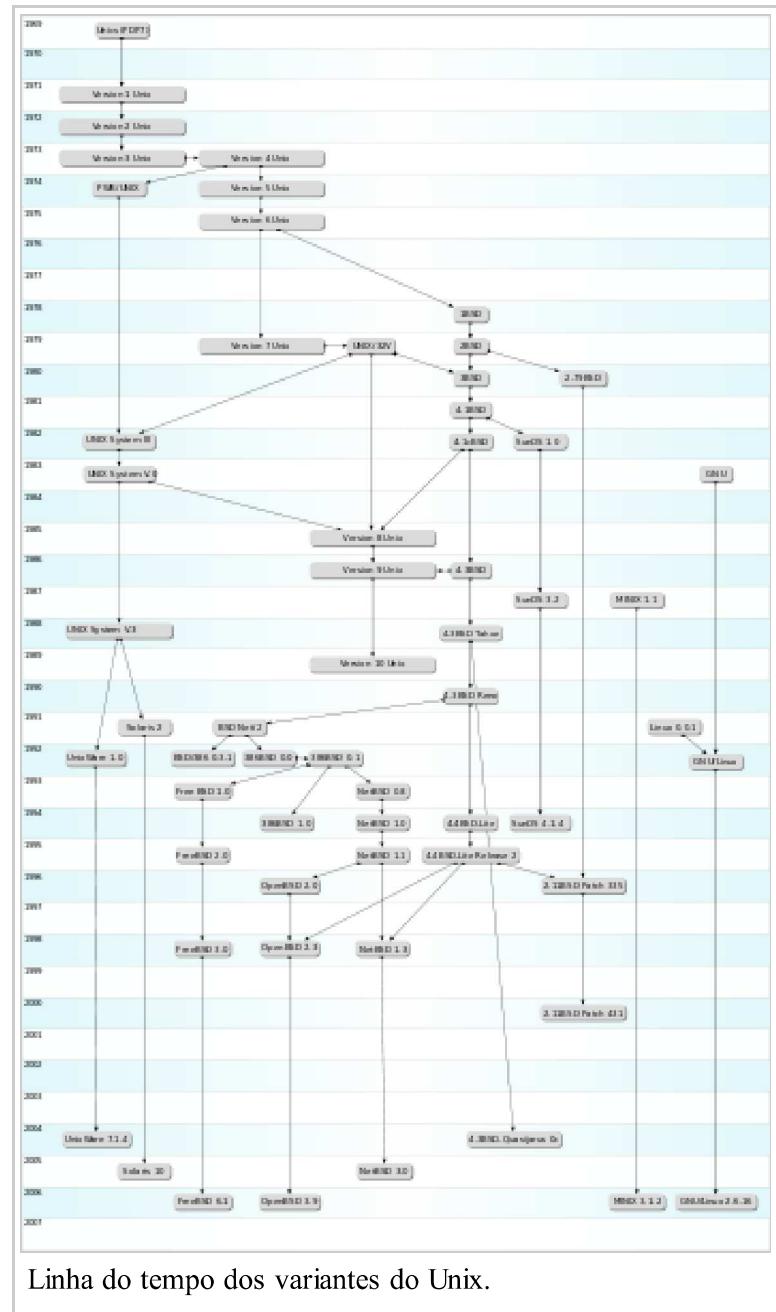
O Multics deveria ser um sistema de tempo compartilhado para uma grande comunidade de usuários. Entretanto, os recursos computacionais disponíveis à época, particularmente os do computador utilizado, um GE 645, revelaram-se insuficientes para as pretensões do projeto. Em 1969, a Bell retirou-se do projeto. Duas razões principais foram citadas para explicar a sua saída. Primeira: três instituições com objetivos díspares dificilmente alcançariam uma solução satisfatória para cada uma delas (o MIT fazia pesquisa, AT&T monopolizava os serviços de telefonia americanos e a GE queria vender computadores). A segunda razão é que os participantes sofriam da síndrome do segundo projeto e, por isso, queriam incluir no Multics tudo que tinha sido excluído dos sistemas experimentais até então desenvolvidos.

Ainda em 1969, Ken Thompson, usando um ocioso computador PDP-7, começou a reescrever o Multics num conceito menos ambicioso, batizado de Unics, usando linguagem de montagem (*assembly*). Mais tarde, Brian Kernighan rebatizou o novo sistema de Unix.

Um marco importante foi estabelecido em 1973, quando Dennis Ritchie e Ken Thompson reescreveram o Unix, usando a linguagem C, para um computador PDP-11. A linguagem C havia sido desenvolvida por Ritchie para substituir e superar as limitações da linguagem B, desenvolvida por Thompson. O seu uso é considerado uma das principais razões para a rápida difusão do Unix.

Finalmente, ao longo dos anos 70 e 80 foram sendo desenvolvidas as primeiras distribuições de grande dimensão como os sistemas BSD (na Universidade de Berkeley na Califórnia) e os System III e System V (nos Bell Labs).

Em 1977, a AT&T começou a fornecer o Unix para instituições comerciais. A abertura do mercado comercial para o Unix deve muito a Peter Weinert - cientista de Yale e fundador da Interactive System Corporation. Weinert conseguiu da AT&T, então já desnudada de seu monopólio nas comunicações e liberada para atuação no mercado de **software**, licença para transportar e comercializar o Unix para o computador Interdata 8/32 para ambiente de automação de escritório. O Unix saía da linha das máquinas PDP, da Digital Equipment Corporation (DEC), demonstrando a relativa facilidade de migração (transporte) para outros computadores, e que, em parte, deveu-se ao uso da linguagem C. O sucesso da Interactive de Weinert com seu produto provou que o Unix era vendável e encorajou outros fabricantes a seguirem o mesmo curso. Iniciava-se a abertura do chamado **mercado Unix**.



Linha do tempo dos variantes do Unix.

Com a crescente oferta de microcomputadores, outras empresas transportaram o Unix para novas máquinas. Devido à disponibilidade dos fontes do Unix e à sua simplicidade, muitos fabricantes alteraram o sistema, gerando variantes personalizadas a partir do Unix básico licenciado pela AT&T. De 1977 a 1981, a AT&T integrarou muitas variantes no primeiro sistema Unix comercial chamado de System III. Em 1983, após acrescentar vários melhoramentos ao System III, a AT&T apresentava o novo Unix comercial, agora chamado de System V. Hoje, o Unix System V é o padrão internacional de fato no mercado Unix, constando das licitações de compra de equipamentos de grandes clientes na América, Europa e Ásia.

Atualmente, Unix (ou \*nix) é o nome dado a uma grande família de Sistemas Operativos que partilham muitos dos conceitos dos Sistemas Unix originais, sendo todos eles desenvolvidos em torno de padrões como o POSIX (Portable Operating System Interface) e outros. Alguns dos Sistemas Operativos derivados do Unix são: BSD (FreeBSD, OpenBSD e NetBSD), Solaris (anteriormente conhecido por SunOS), IRIX, AIX, HP-UX, Tru64, SCO, Linux (nas suas centenas de distribuições), e até o Mac OS X (baseado em um núcleo Mach BSD chamado Darwin). Existem mais de quarenta sistemas operacionais \*nix, rodando desde celulares a supercomputadores, de relógios de pulso a sistemas de grande porte.

## Características

### Sistema operacional multitarefa

Multitarefa significa executar uma ou mais tarefas ou processos simultaneamente. Na verdade, em um sistema monoprocessado, os processos são executados seqüencialmente de forma tão rápida que **parecem** estar sendo executados simultaneamente. O Unix escalona sua execução e reserva-lhes recursos computacionais (intervalo de tempo de processamento, espaço em memória RAM, espaço no disco rígido, etc.).

O Unix é um sistema operacional de multitarefa preemptiva. Isso significa que, quando esgota-se um determinado intervalo de tempo (chamado **quantum**), o Unix suspende a execução do processo, salva o seu contexto (informações necessárias para a execução do processo), para que ele possa ser retomado posteriormente, e coloca em execução o próximo processo da fila de espera. O Unix também determina quando cada processo será executado, a duração de sua execução e a sua prioridade sobre os outros.

A multitarefa, além de fazer com que o conjunto de tarefas seja executado mais rapidamente, ainda permite que o usuário e o computador fiquem livres para realizarem outras tarefas com o tempo economizado.

### Sistema operacional multiutilizador

Uma característica importante do Unix é ser multiusuário (multiutilizador). Bovet e Cesati [4] definem um sistema multiusuário como "aquele capaz de executar, concorrente e independentemente, várias aplicações pertencentes a dois ou mais usuários". O Unix possibilita que vários usuários usem um mesmo computador simultaneamente, geralmente por meio de terminais. Cada terminal é composto de um monitor, um teclado e, eventualmente, um mouse. Vários terminais podem ser conectados ao mesmo computador num sistema Unix. Há alguns anos eram usadas conexões seriais, mas atualmente é mais comum o uso de redes locais, principalmente para o uso de terminais gráficos (ou terminais X), usando o protocolo XDMCP.

O Unix gerencia os pedidos que os usuários fazem, evitando que um interfira com outros. Cada usuário possui direitos de propriedade e permissões sobre arquivos. Quaisquer arquivos modificados pelo usuário conservarão esses direitos. Programas executados por um usuário comum estarão limitados em termos de quais arquivos poderão acessar.

O sistema Unix possui dois tipos de usuários: o usuário **root** (também conhecido como superusuário), que possui a missão de administrar o sistema, podendo manipular todos os recursos do sistema operacional; e os usuários comuns, que possuem direitos limitados.

Para que o sistema opere adequadamente em modo multiusuário, existem alguns mecanismos: (i) um sistema de autenticação para identificação de cada usuário (o programa `login`, p.ex., autentica o usuário verificando uma base de dados, normalmente armazenada no arquivo `/etc/passwd`); (ii) sistema de arquivos com permissões e propriedades sobre arquivos (os direitos anteriormente citados); (iii) proteção de memória, impedindo que um processo de usuário acesse dados ou interfira com outro processo. Esse último mecanismo é implementado com a ajuda do hardware, que consiste na divisão do ambiente de processamento e memória em modo supervisor (ou modo núcleo) e modo usuário.

## Arquivos de dispositivo

Uma característica singular no Unix (e seus derivados) é a utilização intensiva do conceito de arquivo. Quase todos os dispositivos são tratados como arquivos e, como tais, seu acesso é obtido mediante a utilização das chamadas de sistema `open`, `read`, `write` e `close`.

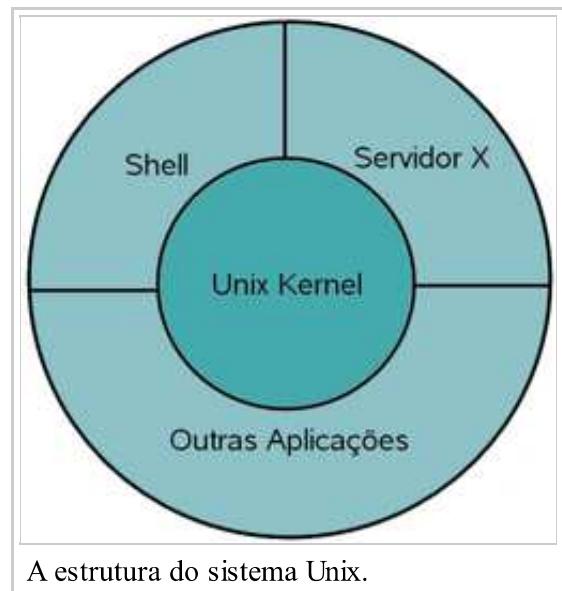
Os dispositivos de entrada e saída são classificados como sendo de bloco (disco, p.ex.) ou de caractere (impressora, modem, etc.) e são associados a arquivos mantidos no diretório `/dev` (v. detalhamento mais adiante).

## Estrutura

Um sistema Unix consiste, basicamente, de duas partes:

- **Núcleo** - o núcleo do sistema operacional, a parte que relaciona-se diretamente com o hardware, e que executa num espaço de memória privilegiado. Agenda processos, gerencia a memória, controla o acesso a arquivos e a dispositivos de hardware (estes, por meio dos controladores de dispositivo - *drivers* - e interrupções). O acesso ao núcleo é feito por chamadas de sistema, que são funções fornecidas pelo núcleo; essas funções são disponibilizadas para as aplicações por bibliotecas de sistema C (`libc`).
- **Programas de sistema** - são aplicações, que executam em espaços de memória não privilegiados, e que fazem a interface entre o usuário e o núcleo. Consistem, principalmente, de:

- **Conjunto de bibliotecas C** (`libc`)
- **Shell** - um ambiente que permite que o usuário digite comandos.
- **Programas utilitários diversos** - são programas usados para manipular arquivos, controlar processos, etc.
- **Ambiente gráfico (GUI)** *graphics user interface* - eventualmente utiliza-se também um ambiente gráfico para facilitar a interação do usuário com o sistema.



A estrutura do sistema Unix.

Em um sistema Unix, o espaço de memória utilizado pelo núcleo é denominado espaço do núcleo ou supervisor (em inglês: *kernel space*); a área de memória para os outros programas é denominada espaço do usuário (*user space*). Essa separação é um mecanismo de proteção que impede que programas comuns interfiram com o sistema operacional.

## Processos

Um processo, na visão mais simples, é uma instância de um programa em execução. Um programa, para ser executado, deve ser carregado em memória; a área de memória utilizada é dividida em três partes: código (*text*), dados inicializados (*data*) e pilha (*stack*).

Por ser um sistema multitarefa, o Unix utiliza uma estrutura chamada tabela de processos, que contém informações sobre cada processo, tais como: identificação do processo (PID), dono, área de memória utilizada, estado (*status*). Apenas um processo pode ocupar o processador em cada instante - o processo encontra-se no estado "executando" (*running*). Os outros processos podem estar "prontos" (*ready*), aguardando na fila de processos, ou então estão "dormindo" (*asleep*), esperando alguma condição que permita sua execução.

Um processo em execução pode ser retirado do processador por duas razões: (i) necessita acessar algum recurso, fazendo uma chamada de sistema - neste caso, após sua retirada do processador, seu estado será alterado para "dormindo", até que o recurso seja liberado pelo núcleo; (ii) o núcleo pode interromper o processo (preempção) - neste caso, o processo irá para a fila de processos (estado "pronto"), aguardando nova oportunidade para executar - ou porque a fatia de tempo esgotou-se, ou porque o núcleo necessita realizar alguma tarefa.

Existem quatro chamadas de sistema principais associadas a processos: `fork`, `exec`, `exit` e `wait`. `fork` é usada para criar um novo processo, que irá executar o mesmo código (programa) do programa chamador (processo-pai); `exec` irá determinar o código a ser executado pelo processo chamado (processo-filho); `exit` termina o processo; `wait` faz a sincronização entre a finalização do processo-filho e o processo-pai.

## Sistema de arquivos

Sistema de arquivos é uma estrutura lógica que possibilita o armazenamento e recuperação de arquivos. No Unix, arquivos são contidos em diretórios (ou pastas), os quais são conectados em uma árvore que começa no diretório raiz (designado por `/`). Mesmo os arquivos que se encontram em dispositivos de armazenamento diferentes (discos rígidos, disquetes, CDs, DVDs, sistemas de arquivos em rede) precisam ser conectados à árvore para que seu conteúdo possa ser acessado. Cada dispositivo de armazenamento possui a sua própria árvore de diretórios.

O processo de conectar a árvore de diretórios de um dispositivo de armazenamento à árvore de diretórios raiz é chamado de "montar dispositivo de armazenamento" (montagem) e é realizada por meio do comando `mount`. A montagem associa o dispositivo a um subdiretório.

### Estrutura de diretórios

A árvore de diretórios do Unix é dividida em várias ramificações menores e pode variar de uma versão para outra. Os diretórios mais comuns são os seguintes:

- / — Diretório raiz - este é o diretório principal do sistema. Dentro dele estão todos os diretórios do sistema.

**/bin** — Contém arquivos, programas do sistema, que são usados com freqüência pelos usuários.

**/boot** — Contém arquivos necessários para a inicialização do sistema.

**/dev** — Contém arquivos usados para acessar dispositivos (periféricos) existentes no computador.

**/etc** — Arquivos de configuração de seu computador local.

**/home** — Diretórios contendo os arquivos dos usuários.

**/lib** — Bibliotecas compartilhadas pelos programas do sistema e módulos do núcleo.

**/mnt** — Diretório de montagem de dispositivos.

**/mnt/cdrom** — Subdiretório onde são montados os CDs. Após a montagem, o conteúdo do CD se encontrará dentro deste diretório.

**/mnt/floppy** — Subdiretório onde são montados os disquetes. Após a montagem, o conteúdo do disquete se encontrará dentro deste diretório.

**/proc** — Sistema de arquivos do núcleo. Este diretório não existe, ele é colocado lá pelo núcleo e usado por diversos programas.

**/root** — Diretório do usuário root.

**/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 freqüência pelos programas do sistema.

## Particularidades

Um sistema Unix é orientado a arquivos, quase tudo nele é arquivo. Seus comandos são na verdade arquivos executáveis, que são encontrados em lugares previsíveis em sua árvore de diretórios, e até mesmo a comunicação entre entidades e processos é feita por estruturas parecidas com arquivos. O acesso a arquivos é organizado através de propriedades e proteções. Toda a segurança do sistema depende, em grande parte, da combinação entre as propriedades e proteções definidas em seus arquivos e suas contas de usuários.

## Aplicações

O Unix permite a execução de pacotes de softwares aplicativos para apoio às diversas atividades empresariais. Dentre estes pacotes destacam-se:

- geradores gráficos
- planilhas eletrônicas
- processadores de textos

- geradores de aplicações
- linguagens de 4º geração
- banco de dados

O Unix possui recursos de apoio à comunicação de dados, que proporcionam sua integração com outros sistemas Unix, e até com outros sistemas operacionais distintos. A integração com sistemas heterogêneos permite as seguintes facilidades:

- compartilhamento de recursos e informações
- transferência de informações
- comunicação entre usuários remotos
- submissão de programas para serem executados em computadores remotos
- utilização dos terminais de uma máquina Unix como terminais de outras máquinas remotas, mesmo com sistemas operacionais distintos.

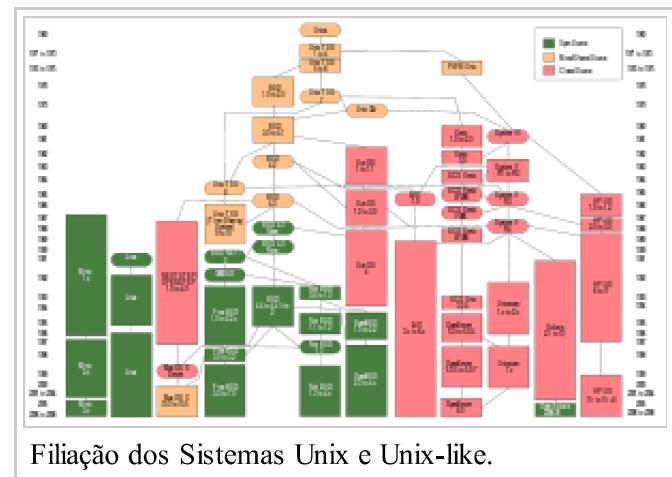
Para última, o Unix oferece um ambiente integrado e amigável, voltado para a gestão automatizada de escritório, com serviços que atenderão às seguintes áreas:

- arquivamento eletrônico de informações
- processador de documentos
- agenda e calendário
- calculadora
- correio eletrônico

## Ambiente gráfico do Unix

### X Window System

Além do *shell*, o Unix suporta interface gráfica para o usuário. Nas primeiras versões do Unix as interfaces do usuário eram baseadas apenas em caracteres (modo texto) e o sistema compunha-se apenas do núcleo, de bibliotecas de sistema, do *shell* e de alguns outros aplicativos. As versões mais recentes do Unix, além de manterem o *shell* e seus comandos, incluem o X Window System que, graças ao gerenciador de exibição e ao gerenciador de janelas, possui uma interface atraente e intuitiva que aumenta em muito a produtividade do usuário.



Desenvolvido no MIT (Massachusetts Institute of Technology), o X Window System (também pode ser chamado de Xwindow) tornou-se o sistema gráfico do Unix. O Xwindow funciona como gerenciador de exibição e por si só, não faz muita coisa. Para termos um ambiente gráfico produtivo e completo, precisamos também de um gerenciador de janelas.

O gerenciador de janelas proporciona ao ambiente gráfico a aparência e as funcionalidades esperadas incluindo as bordas das janelas, botões, truques de *mouse*, menus etc. Como no sistema Unix o gerenciador de exibição (X Window System) é separado do gerenciador de janelas, dizemos que seu ambiente gráfico é do tipo cliente-servidor. O Xwindow funciona como servidor e interage diretamente com o *mouse*, o teclado e o vídeo. O gerenciador de janelas funciona como cliente e se aproveita dos recursos disponibilizados pelo Xwindow.

O fato de o Unix possuir o gerenciador de exibição (Xwindow) separado do gerenciador de janelas tornou possível o surgimento de dezenas de gerenciadores de janelas diferentes. Os gerenciadores de janelas mais comuns no mundo Unix são o Motif, Open Look, e o CDE. Também existem outros gerenciadores de janelas que são bastante utilizados no Unix, principalmente nos sistemas Unix-Like (versões gratuitas e clones do Unix). São eles: KDE, Gnome, FVWM, BlackBox, Enlightenment, WindowMaker etc.

## Comandos

Esta é uma lista de programas de computador para o sistema operacional Unix e os sistemas compatíveis, como o Linux. Os comandos do Unix tornam-se acessíveis ao usuário a partir do momento em que ele realiza o login no sistema. Se o usuário utiliza tais comandos, então ele se encontra no modo shell, também chamado de modo texto (ou Unix tradicional). Quando estiver utilizando o modo gráfico, o usuário também poderá se utilizar de tais comandos desde que abra uma janela de terminal (Xterm).

A linha de comando do sistema operacional Unix permite a realização de inúmeras tarefas através de seus comandos, de manipulação de arquivos a verificação do tráfego em rede. Para exibir uma descrição detalhada de cada comando abra uma console ou xterm e digite *man comando*, onde comando é o comando em questão.

### Comandos de manipulação de diretório

- `mkdir` - Cria um diretório vazio exemplo: `mkdir docs`
- `rmdir` - Exclui um diretório (se estiver vazio)
- `rm -rf` - Exclui um diretório e todo o seu conteúdo (cuidado com este comando)
- `cd` - Entra num diretório (exemplo: `cd docs`) ou retorna para *HOME*
- `cd /` - Muda para o diretório raiz
- `cd ~` - vai direto para o diretório home do usuário logado.
- `cd ..` - volta ao último diretório acessado
- `pwd` - Exibe o local do diretório atual
- `ls` - Lista o conteúdo do diretório
- `ls -alh` - Mostra o conteúdo detalhado do diretório
- `ls -a` - Exibe os arquivos "ocultos" do determinado diretório.
- `ls -ltr` - Mostra os arquivos no formato longo(l) em ordem inversa(r) de data (t)
- `df` - Mostra a utilização dos sistemas de arquivos montados
- `du -ms` - Mostra o tamanho do diretório em Megabytes
- `whereis` - Mostra onde se encontra determinado arquivo (binários) exemplo: `whereis samba`

### Comandos para manipulação de arquivos

- `cat` - Mostra o conteúdo de um arquivo binário ou texto
- `tac` - Semelhante ao cat mas inverte a ordem
- `tail` - Mostra as últimas linhas de um arquivo. Ex: `tail -f <arquivo>` Útil para visualizar arquivos de log continuamente.
- `head` - Mostra as primeiras linhas de um arquivo. Ex: `head -100` visualiza as 100 primeiras linhas do arquivo.
- `less` - Mostra o conteúdo de um arquivo de texto com controle
- `vi` - Editor de ficheiros de texto
- `vim` - Versão melhorada do editor supracitado



Gerenciador de janelas Window Maker.

`rm` - Remoção de arquivos (também remove diretórios, mas com o parâmetro `-r`, que significa recursividade)

`cp` - Copia diretórios '`cp -r`' copia recursivamente

`mv` - Move ou renomeia arquivos e diretórios

`chmod` - Altera as permissões de arquivos ou directórios

`chown` - Altera o dono de arquivos ou directórios

`cmd>txt` - Cria um novo arquivo(txt) com o resultado do comando(cmd)

`cmd>>txt` - Adiciona o resultado do comando(cmd) ao fim do arquivo(txt)

`touch foo.txt` - Cria um arquivo foo.txt vazio; também altera data e hora de modificação para **agora**

`> arquivo.txt` - Mais rápido que o touch para criação de arquivos

`split` - Divide um arquivo

`recode` - Recodifica um arquivo ex: `recode iso-8859-15..utf8 file_to_change.txt`

## Comandos para administração

`man` - Mostra o manual do comando.

`adduser` - Adiciona usuários. O `useradd` pode também ser usado.

`addgroup` - Adiciona grupos. O `groupadd` pode também ser usado.

`apropos` - Realiza pesquisa por palavra ou string

`dmesg` - Exibe as mensagens da inicialização(log)

`du` - Exibe estado de ocupação dos discos/partições

`find` - Comando de busca ex: `find ~/ -cmin -3`

`userdel` - Remove usuários

`usermod` - Modifica informações de um determinado usuário.

`groupmod` - Modifica informações de um determinado grupo.

`chfn` - Altera informação relativa a um utilizador (usuário).

`who` - Informa quem está logado no sistema. Em algumas versões do Linux, o comando `w` pode ser usado, e retorna informações mais detalhadas, como o shell do usuário.

`whoami` - Informa com qual usuário você está logado

`passwd` - Modifica senha (password) de usuários

`umask` - Define padrões de criação de arquivos e diretórios

`ps` - Mostra os processos correntes

`ps aux` (ou `ps -ef`) - Mostra todos os processos correntes no sistema

`kill` - Mata um processo

`killall` - Mata todos os processos com o nome informado

`su` - Troca para o super-usuário root (é exigida a senha)

`su user` - Troca para o usuário especificado em 'user' (é exigida a senha)

`chown` - Altera o proprietário de arquivos e pastas (dono)

## Comandos para administração de rede

`ifconfig` - mostra as interfaces de redes ativas e as informações relacionadas a cada uma delas

`route` - Mostra as informações referentes as rotas

`mtr` - Mostra rota até determinado IP

`netstat` - Exibe as portas e protocolos abertos no sistema.

`iptraf` - Analisador de tráfego da rede com interface gráfica baseada em diálogos

`tcpdump` - Sniffer muito popular. Sniffer é uma ferramenta que "ouve" os pacotes que estão passando pela rede.

**traceroute** Traça uma rota do host local até o destino mostrando os roteadores intermediários

**nslookup** - Consultas a serviços DNS

**dig** - Consultas a serviços DNS

## Notas e referências

1. ↑ Uma breve história do Unix (<http://www.geek.com.br/posts/10647-uma-breve-historia-do-unix>)
  2. ↑ Principles of operating systems: design & applications, Brian L, publicação Thompson Learning, local Boston, Massachusetts ISBN 1-4188-3769-5, Pag. 23
- [1] BACH, Maurice J. **The design of the Unix operating system**. New Jersey: Prentice Hall, 1990.
  - [2] TANENBAUM, Andrew. **Sistemas operacionais modernos**. Rio de Janeiro: LTC, 1999.
  - [3] MCKUSICK, Marshall K.; NEVILLE-NEIL, George V. **The design and implementation of the FreeBSD operating system**. Boston: Addison-Wesley, 2004.
  - [4] BOVET, Daniel P.; CESATI, Marco. **Understanding the Linux kernel**. Sebastopol: O'Reilly, 2005.

## Ver também

- unix-like - Sistemas operacionais do tipo unix .
- ARX - SO da Acorn Computers baseado no Unix
- Plan 9 - baseado em alguns conceitos do UNIX. Também desenvolvido pelo Bell Labs. Leva a filosofia de tudo é um arquivo mais a fundo. Onde até os servidores são arquivos.
- Filosofia Unix
- Era Unix

## Ligações externas

- Unix.org (<http://www.unix.org>)
- Foca GNU/Linux (<http://focalinux.cipsga.org.br/guia/iniciante/index.htm>)
- UNIX ON-LINE Man Pages (<http://unixhelp.ed.ac.uk/CGI/man-cgi?>)

Obtida de "<http://pt.wikipedia.org/w/index.php?title=Unix&oldid=37645682>"

Categorias: Unix | Software de 1969

- Esta página foi modificada pela última vez à(s) 17h24min de 15 de dezembro de 2013.
- Este texto é disponibilizado nos termos da licença Atribuição-Partilha nos Mesmos Termos 3.0 não Adaptada (CC BY-SA 3.0); pode estar sujeito a condições adicionais. Consulte as condições de uso para mais detalhes.