

A Estrutura de Diretórios do Linux

A Estrutura de Diretórios do Linux

A estrutura de diretórios do Linux pode parecer um pouco estranha a princípio para quem vem do ambiente Windows. Por exemplo, não temos letras de unidade como C: e D:, e as partições são numeradas, entre outras diferenças.

A estrutura dos sistemas de arquivos do Linux é definida por um padrão denominado *Filesystem Hierarchy Standard* (Padrão de Hierarquia do Sistema de Arquivos), a qual define também as estruturas de outros sistemas como o BSD, por exemplo.

O Filesystem Hierarchy Standard (FHS) evoluiu a partir de padrões históricos originados de versões mais antigas do UNIX, como a Berkeley Software Distribution (BSD) e outras. O FHS fornece aos desenvolvedores Linux e administradores de sistemas uma estrutura de diretórios padrão para o sistema de arquivos, trazendo consistência entre sistemas e distribuições.

No site <http://www.pathname.com/fhs/> podemos acessar o documento padrão FHS e baixá-los para consulta e estudo.

A página oficial do projeto é <http://www.linuxfoundation.org/collaborate/workgroups/lsb/fhs>

Vejamos um resumo sobre a estrutura de diretórios no Linux e uma breve descrição dos principais diretórios:

/ – O diretório Root (Raiz)

Tudo o que há no seu sistema Linux fica localizado dentro do diretório raiz, representado por /. É como se fosse um “C:\” do Windows, porém outras partições e discos também se localizam sob o diretório raiz no Linux, enquanto no Windows cada partição teria sua própria letra de unidade separada. No Linux, as demais partições se encontram “montadas” em pastas dentro da hierarquia de diretórios, sob a raiz (/).

Vejamos na figura abaixo o diretório raiz do Linux e seu conteúdo básico, listado com o comando **tree -L 1**:

```
fabio@ubuntu-fabio:/$ tree -L 1
.
├── bin
├── boot
├── cdrom
├── dev
├── etc
├── home
├── initrd.img -> boot/initrd.img-3.13.0-32-generic
├── lib
├── lib64
├── lost+found
├── media
├── mnt
├── opt
├── proc
├── root
├── run
├── sbin
├── srv
├── sys
├── tmp
├── usr
├── var
└── vmlinuz -> boot/vmlinuz-3.13.0-32-generic

21 directories, 2 files
fabio@ubuntu-fabio:/$
```

/bin – Binários essenciais dos usuários

O diretório **/bin** contém binários essenciais aos usuários – ou seja, programas – que precisam estar presentes quando o sistema é inicializado no modo monousuário. Aplicativos comuns, como navegadores e jogos geralmente se localizam no diretório **/usr/bin**, ao passo que programas e utilitários importantes são armazenados no diretório **/bin**.

O diretório **/bin** contém binários executáveis, comandos essenciais que são utilizados quando em modo monousuário e também muitos comandos essenciais que são requeridos por todos os usuários do sistema, tais como **ls**, **rmdir** e **date**.

Já os comandos que não são essenciais para o sistema quando em modo monousuário são colocados no diretório **/usr/bin**, ao passo que o diretório **/sbin** é usado para armazenar binários essenciais que tem relação com a administração do sistema.

Veja abaixo uma figura mostrando o conteúdo básico do diretório **/bin**:

bash	fgconsole	nc	sed
bunzip2	fgrep	nc.openbsd	setfacl
busybox	findmnt	netcat	setfont
bzcat	fuser	netstat	setupcon
bzcmp	fusermount	nisdomainname	sh
bzdiff	getfacl	ntfs-3g	sh.distrib
bzegrep	grep	ntfs-3g.probe	sleep
bzexe	gunzip	ntfs-3g.secaudit	ss
bzfgrep	gzexe	ntfs-3g.usermap	static-sh
bzgrep	gzip	ntfscat	stty
bzip2	hostname	ntfsck	su
bzip2recover	ip	ntfsccluster	sync
bzless	kbd_mode	ntfscmp	tailf
bzmore	kill	ntfsdump_logfile	tar
cat	kmod	ntfsfix	tempfile
chacl	less	ntfsinfo	touch
chgrp	lessecho	ntfsls	true
chmod	lessfile	ntfsmftalloc	udevadm
chown	lesskey	ntfsmove	ulockmgr_server
chvt	lesspipe	ntfstruncate	umount
cp	ln	ntfswipe	uname
cpio	loadkeys	open	uncompress
dash	login	openvt	unicode_start
date	loginctl	pidof	vdirc
dbus-cleanups-sockets	lowntfs-3g	ping	vmouse_detect
dbus-daemon	ls	ping6	which
dbus-uuidgen	lsblk	plymouth	whiptail
dd	lsmod	plymouth-upstart-bridge	ypdomainname
df	mkdir	ps	zcat
dir	mknod	pwd	zcmp

/boot – Arquivos estáticos de inicialização

O diretório /boot contém arquivos necessários para inicializar o sistema, como os arquivos do carregador de inicialização **GRUB** e o kernel (ou kernels) do Linux. Alguns arquivos de configuração se localizam no diretório /etc.

/cdrom – Ponto de montagem para drives ópticos, como CD-ROMs

Este diretório não faz parte do padrão de hierarquia FHS, porém ele ainda é encontrado em algumas distribuições Linux. Ele é usado como local temporário para CDs e DVDs inseridos no drive – porém, o local padrão para essas mídias é o diretório /media.

/dev – Arquivos de Dispositivos

No Linux os dispositivos (hardware e software) são representados como arquivos, e esse diretório contém uma grande quantidade de arquivos especiais que representam esses dispositivos.

Estes arquivos de dispositivos não são arquivos comuns, e seu conteúdo não é legível – tente rodar o comando **cat /dev/sda1** para ver o que aparece.

Na verdade, o /dev/sda representa o primeiro disco SATA instalado no sistema. Para trabalhar com esse disco, é necessário usar um utilitário especial como o fdisk ou o mkfs, por exemplo.

Neste diretório também encontramos os chamados “pseudo-dispositivos”, que são dispositivos virtuais que não correspondem a hardware real da máquina. Como exemplo podemos citar o dispositivo especial /dev/null que não produz saída

nenhuma e automaticamente descarta toda a entrada direcionada a ele – é como um “buraco negro” de dados.

O diretório /dev contém nós de dispositivos, que são uma espécie de pseudo-arquivos usados por muitos dispositivos de hardware (e de software) com a exceção de alguns dispositivos de rede. O diretório /dev/está sempre vazio quando não for montado, e se for montado conterá entradas que são criadas pelo sistema udev, o qual cria e também gerencia os nós no Linux, criando-os de forma dinâmica quando os dispositivos são encontrados.

Este diretório é interessante pois mostra uma característica marcante do sistema operacional Linux: no Linux, tudo é um arquivo ou diretório. Usamos esses arquivos para configurar e acessar vários dispositivos de hardware. Veremos com mais detalhes essa teoria e o funcionamento do diretório /dev em outro tópico. Abaixo, a definição de alguns dos arquivos encontrados dentro de /dev:

Arquivo em /dev	Definição
/dev/dsp	Digital Signal Processor; interface entre o software que produz um som e a placa de som
/dev/fd0	Primeiro drive de 3 ½
/dev/hda	Drive Master na controladora primária IDE
/dev/hdb	Drive Slave na controladora primária IDE
/dev/hdc e /dev/hdd	Drives master e slave na controladora secundária IDE
/dev/ht0	Primeiro drive de fita IDE
/dev/lp0	Primeiro dispositivo de impressora paralela
/dev/psaux	Porta de mouse PS/2
/dev/loop0	Primeiro dispositivo de loopback. Usado para montar sistemas de arquivos não localizados em outros dispositivos de blocos, como uma imagem ISO9660 sem gravá-la em mídia
/dev/null	“Balde de bits”; Buraco negro para onde dados podem ser enviados – e nunca mais vistos
/dev/random	Gerador de números aleatórios do kernel
/dev/sda	Primeiro dispositivo SCSI ou SATA
/dev/sdb	Segundo dispositivo SCSI ou SATA
/dev/ttySO	Primeira porta serial

Arquivo em /dev	Definição
/dev/zero	Muitos zeros – retorna o valor 0

Uma listagem bem completa de arquivos de dispositivos pode ser encontrada em <http://www.lanana.org/docs/device-list/devices-2.6+.txt>

/etc – Arquivos de configuração diversos

O diretório /etc contém muitos arquivos de configuração do sistema, os quais podem geralmente ser editados manualmente usando-se um editor de textos, como o vi ou o emacs.

O diretório /etc não contém programas binários, contendo apenas scripts executáveis. Veja abaixo uma parte do conteúdo do diretório /etc:

```
fabio@ubuntu-fabio:/etc$ ls
acpi                  hosts                protocols
adduser.conf         hosts.allow         pulse
alternatives        hosts.deny          python
anacrontab          hp                  python2.7
apg.conf            ifplugd            python3
apm                 iftab              python3.4
apparmor            init               rc0.d
apparmor.d          init.d             rc1.d
appport            initramfs-tools    rc2.d
apt                 inputrc            rc3.d
aptdaemon           inserv             rc4.d
at-spi2             inserv.conf        rc5.d
avahi               inserv.conf.d      rc6.d
bash.bashrc         iproute2           rc.local
bash_completion     issue              rcS.d
bash_completion.d   issue.net          resolvconf
bindresvport.blacklist kbd                resolv.conf
blkid.conf          kernel             rmt
blkid.tab           kernel-img.conf    rpc
bluetooth           kerneloops.conf    rsyslog.conf
brlapi.key          ldap              rsyslog.d
brlitty             ld.so.cache        samba
brlitty.conf        ld.so.conf         sane.d
ca-certificates     ld.so.conf.d       securetty
ca-certificates.conf legal              security
calendar            libaudit.conf      selinux
chatscripts         libnl-3            sensors3.conf
colord.conf         libpaper.d         sensors.d
```

Como exemplo na figura vemos o arquivo hosts que possui mapeamentos entre endereços IP e nomes de computadores, e o arquivo resolv.conf, que possui informações sobre resolução de nomes (servidores DNS).

/home – Diretórios Home dos usuários

O diretório /home contém um diretório padrão (de perfil) para cada usuário. Se o nome de seu usuário é fabio, então você encontrará um diretório de nome fabio dentro de /home, portando /home/fabio. Este diretório contém arquivos do usuário fabio e arquivos de configuração específicos dessa conta de usuário. Os usuários possuem permissão de gravação apenas em seu próprio diretório padrão, e apenas

permissão de leitura em outros diretórios do sistema (em alguns casos, permissão nenhuma).

Veja abaixo a listagem de um diretório /home, que contém três diretórios pertencentes às contas dos usuários fabio, mariana e daniela:

```
fabio@ubuntu-fabio:/home$ ls -l
total 12
drwxr-xr-x  2 daniela daniela 4096 Ago 21 21:02 daniela
drwxr-xr-x 21 fabio   fabio  4096 Ago 21 20:52 fabio
drwxr-xr-x  2 mariana mariana 4096 Ago 21 21:02 mariana
fabio@ubuntu-fabio:/home$
```

/lib – Bibliotecas compartilhadas essenciais do sistema

O diretório **/lib** contém bibliotecas que são necessárias aos programas localizados nos diretórios /bin /sbin.

Já as bibliotecas usadas pelos programas do diretório /usr/bin se localizam no diretório /usr/lib.

As bibliotecas são códigos de programas que são compartilhados entre aplicações e são necessários para que elas funcionem. Essas bibliotecas geralmente tem nomes que se iniciam com ld or lib.

```
fabio@ubuntu-fabio:/lib$ ls
apparmor          libip6tc.so.0      plymouth
brltty            libip6tc.so.0.1.0  recovery-mode
cpp              libiptc.so.0        resolvconf
crda             libiptc.so.0.0.0    security
firmware         libxtables.so.10     systemd
hdparm          libxtables.so.10.0.0 terminfo
ifupdown        linux-sound-base     udev
init            lsb                  ufw
klibc-P2s_k-gf23VtrGg02_4pGkQgwMY.so modprobe.d          x86_64-linux-gnu
libip4tc.so.0     modules             xtables
libip4tc.so.0.1.0 modules-load.d
fabio@ubuntu-fabio:/lib$
```

Algumas distribuições Linux possuem o diretório /lib64, o qual contém bibliotecas de 64 bits, ao passo que o diretório /lib armazena as bibliotecas de 32 bits. Em sua maioria essas bibliotecas são chamadas de dlls, “dynamically loaded libraries”, ou ainda shared libraries (bibliotecas compartilhadas) ou também Shared Objects (SO).

Neste diretório também costumamos encontrar os módulos do Kernel, que são códigos do kernel, geralmente drivers de dispositivos carregáveis e descarregáveis sem que seja necessária a reinicialização do sistema. Esses módulos são encontrados no subdiretório /lib/modules/<kernel-version-number>.

/lost+found – Arquivos recuperados (“perdidos+encontrados”)

Caso ocorra um travamento no sistema, uma verificação do sistema de arquivos será realizada na próxima reinicialização. Se forem encontrados arquivos corrompidos eles serão colocados no diretório lost+found, permitindo assim que recuperemos esses dados, ou ao menos boa parte deles.

/media – Mídias Removíveis

O diretório **/media** contém subdiretórios onde são montados dispositivos de mídias removíveis inseridas no computador, como por exemplo um CD inserido no drive de CD/DVD, o qual será montado em um diretório criado automaticamente dentro de /media, nos permitindo acessar o conteúdo da mídia.

/mnt – Pontos de montagem temporários

O diretório **/mnt** é onde sistemas de arquivos temporários podem ser montados pelos administradores de sistemas durante seu uso. Por exemplo, podemos montar uma partição do Windows nesse diretório para efetuarmos tarefas de recuperação de arquivos, criando o diretório **/mnt/windows**.

Na verdade, podemos montar esses sistemas de arquivos em vários locais do sistema, não necessariamente em /mnt.

/opt – Pacotes opcionais

O diretório **/opt** é um contêiner para pacotes de software opcionais. Muito usado por determinados softwares proprietários que não obedecem ao FHS.

Muitos sistemas unix armazenam software compilado localmente no diretório **/opt**, e se esse for o caso, é interessante mantê-lo em uma partição separada.

Neste caso também é interessante criar um link simbólico do diretório **/usr/local** para o diretório **/opt**.

Algumas distribuições instalam softwares pre-compilados em outros diretórios, como o **/usr/bin**.

/proc – Arquivos de Processos e de Kernel

O diretório **/proc** contém arquivos especiais que representam informações sobre processos e sobre o sistema.

O sistema de arquivos virtual proc

O sistema de arquivos montado no diretório **/proc** é também conhecido como pseudo sistema de arquivos, pois seus arquivos não existem fisicamente no disco rígido do computador.

O sistema de arquivos **/proc** contém arquivos virtuais, que são arquivos que existem apenas na memória RAM da máquina, os quais permitem visualizar dados do kernel que mudam constantemente. Não há arquivos reais aqui, mas informações em tempo real do sistema, como configurações de hardware e memória física. Podemos citar alguns arquivos presentes no diretório **/proc**:

/proc/cpuinfo – Informações sobre a CPU

/proc/interrupts – Informações sobre IRQs

/proc/meminfo – Informações sobre a memória do sistema

/proc/mounts – Informações sobre dispositivos e pontos de montagem

/proc/partitions – Informações sobre as partições dos discos

/proc/version – Versão do Kernel do Linux e do compilador gcc.

E o diretório /proc também possui alguns subdiretórios, como o diretório /proc/<ID_processo>, que se trata de um diretório que contém informações sobre um processo rodando no sistema. Há um diretório desses para cada processo em execução.

Também existe o subdiretório **/proc/sys**, o qual possui muitas informações sobre o sistema, como configurações e dados sobre o hardware.

/root – Diretório home do usuário root

Este diretório é o diretório padrão do usuário root. Veja que o root não tem seu diretório home como subdiretório de /home, como todos os demais usuários.

Não confunda o diretório /root com o diretório / (root), que é o diretório raiz do sistema.

/run – Arquivos de Estado de Aplicações

O diretório **/run** fornece às aplicações um local para armazenamento de arquivos transientes que elas necessitem usar, como por exemplo sockets.

Apesar desses arquivos serem transientes, eles não são armazenados em /tmp pois lá os arquivos poderiam ser excluídos causando problemas às aplicações que os utilizam.

/sbin – Binários para Administração do Sistema

O diretório **/sbin** é muito parecido com o diretório /bin. Ele possui muitos programas binários essenciais que são geralmente utilizados pelo administrador do sistema em suas tarefas de gerenciamento.


```

fabio@ubuntu-fabio:/sbin$ ls
acpi_available  fstrim-all      mkfs             restart
agetty         gdisk           mkfs.bfs        rmmod
alsa           getcap          mkfs.cramfs     route
apm_available  getpcaps        mkfs.ext2       rtacct
apparmor_parser  getty          mkfs.ext3       rtmon
badblocks      halt            mkfs.ext4       runlevel
blkid          hdparm          mkfs.ext4dev    setcap
blockdev       hwclock         mkfs.fat        setvtrgb
bridge        ifconfig        mkfs.minix      sfdisk
brlty         ifdown          mkfs.msdos      sgdisk
brlty-setup    ifquery         mkfs.ntfs       shadowconfig
capsh         ifup            mkfs.vfat       shutdown
cfdisk        init            mkhomedir_helper  slattach
cgdisk        initctl         mkntfs          start
crda          insmod          mkswap          startpar
ctrlaltdel     installkernel  mntctl          startpar-upstart-inject
debugfs        ip              modinfo         start-stop-daemon
depmod         ip6tables      modprobe        status
dhclient       ip6tables-apply  mountall        stop
dhclient-script  ip6tables-restore  mount.fuse      sulogin
dmsetup        ip6tables-save   mount.lowntfs-3g  swapon
dosfsck        ipmaddr         mount.ntfs       swapoff
dosfslabel     iptables        mount.ntfs-3g    swapon
dumpe2fs       iptables-apply   mount.vboxsf     switch_root
e2fsck        iptables-restore  nameif          sysctl
e2image       iptables-save    ntfsclone        tc
e2label       iptunnel         ntfsck           telinit

```

Como exemplo podemos citar os programas formatadores de partições **mkfs.ext3** e **mkfs.vfat**, e os comandos **shutdown** e **runlevel**.

/srv – Dados de Serviços

O diretório **/srv** possui dados que são utilizados por serviços. Como exemplo podemos citar o web server Apache, que usa o diretório **/srv** (na verdade, um subdiretório dentro dele) para armazenamento das páginas (arquivos) de um website.

/sys – Informações sobre o sistema e hardware

Sistema de arquivos pseudo-virtual que fornece informações sobre o sistema e o hardware do computador. Pode ser usado para alterar parâmetros do sistema e também para tarefas de debugung.

Função: facilitar a troca de informações entre os programas que rodam no espaço do kernel, como os drivers, com os programas que rodam no espaço do usuário (aplicações).

Quando um dispositivo é adicionado ao sistema, o kernel cria um nome de dispositivo em **/sys**, e notifica o utilitário **udev**, o qual gerencia os nomes de dispositivos dinamicamente, criando então um arquivo de dispositivo, geralmente em **/dev** (ou o remove)

Alguns subdiretórios de **/sys** e suas respectivas funções:

/sys/bus – Barramentos de dados do sistema

/sys/module – Módulos carregados no kernel

/sys/devices – Todos os dispositivos conectados

/sys/block – Dispositivos de bloco, como HDs

/tmp – Arquivos temporários

As aplicações podem armazenar arquivos temporários no diretório **/tmp**. Estes arquivos são geralmente excluídos quando o sistema é reiniciado e podem ser excluídos a qualquer momento por utilitários como o `tmpwatch`.

O diretório `/tmp` é onde os usuários, assim como os programas armazenam arquivos de forma temporária.

É interessante mantê-lo em uma partição separada pois seu conteúdo pode crescer muito e interferir com o restante do sistema.

/usr – Binários dos usuários e Dados Somente-Leitura

O diretório `/usr` contém aplicações e arquivos utilizados pelos usuários comuns do sistema, ao contrário das aplicações e arquivos que são usados pelo sistema em si. Por exemplo, aplicações não-essenciais estão localizadas dentro do diretório `/usr/bin` em vez do diretório `/bin` e binários de administração do sistema ficam localizados no diretório `/usr/sbin` em vez do diretório `/sbin`. Bibliotecas estão localizadas dentro do diretório `/usr/lib`.

O diretório `/usr` também contém outros diretórios – por exemplo, arquivos que não dependem de arquitetura como gráficos, que se localizam no diretório `/usr/share`.

Já o diretório **`/usr/local`** é onde aplicações compiladas localmente são instaladas por padrão – isso as impedem de bagunçar o resto do sistema.

Também é interessante manter esse diretório em uma partição separada, pois pacotes são adicionados costumeiramente ao sistema.

Subdiretórios localizados em `/usr`

O diretório `/usr` possui ao menos os seguintes subdiretórios:

Diretório	Uso
<code>/usr/include</code>	Arquivos de cabeçalho usados para compilar aplicações
<code>/usr/lib</code>	Bibliotecas para programas nos diretórios <code>/usr/bin</code> e <code>/usr/sbin</code> .
<code>/usr/lib64</code>	Bibliotecas de 64 bits para programas de 64 bits nos diretórios <code>/usr/bin</code> e <code>/usr/sbin</code> .
<code>/usr/sbin</code>	Binários de sistema não essenciais, tais como <i>daemons</i> do sistema.
<code>/usr/share</code>	Dados compartilhados usados por aplicações, geralmente não-dependentes de arquitetura.
<code>/usr/src</code>	Código-fonte, normalmente para o kernel do Linux

Diretório	Uso
/usr/X11R6	Arquivos de configuração do sistema X Window. Geralmente, obsoletos.
/usr/local	Dados e programas específicos da máquina local. Alguns subdiretórios incluem bin, sbin, lib, share, include, entre outros.
/usr/bin	Este é o diretório primário dos comandos executáveis no sistema.

/var – Arquivos de Dados Variáveis

O diretório **/var** é a contraparte com permissão de escrita do diretório **/usr**, o qual deve ser somente-leitura quando em operação normal. Arquivos de log e tudo o mais que normalmente seria escrito em **/usr** durante a operação normal são escritos no diretório **/var**. Por exemplo, encontramos arquivos de log no diretório **/var/log**.

O diretório **/var** é onde o sistema armazena seus arquivos de spool, como o spool de impressão, filas de entrada e saída de email e arquivos de log do sistema, entre outros.

Por conta disso, esses arquivos podem aumentar e diminuir de tamanho drasticamente e sem aviso.

Ele também contém arquivos cujo tamanho e conteúdo podem mudar enquanto o sistema está em execução, como as entradas nos diretórios a seguir:

- Arquivos de log do sistema: **/var/log**
- Pacotes e arquivos de banco de dados: **/var/lib**
- Filas de impressão: **/var/spool**
- Arquivos temporários: **/var/tmp**

O diretório **/var** pode ser colocado em seu próprio sistema de arquivos de modo que o crescimento dos arquivos possa ser acomodado e os tamanhos dos arquivos não afetem o sistema de forma grave. Diretórios de serviços de rede como **/var/ftp** (serviço de FTP) e **/var/www** (serviço web HTTP) também são encontrados dentro de **/var**.