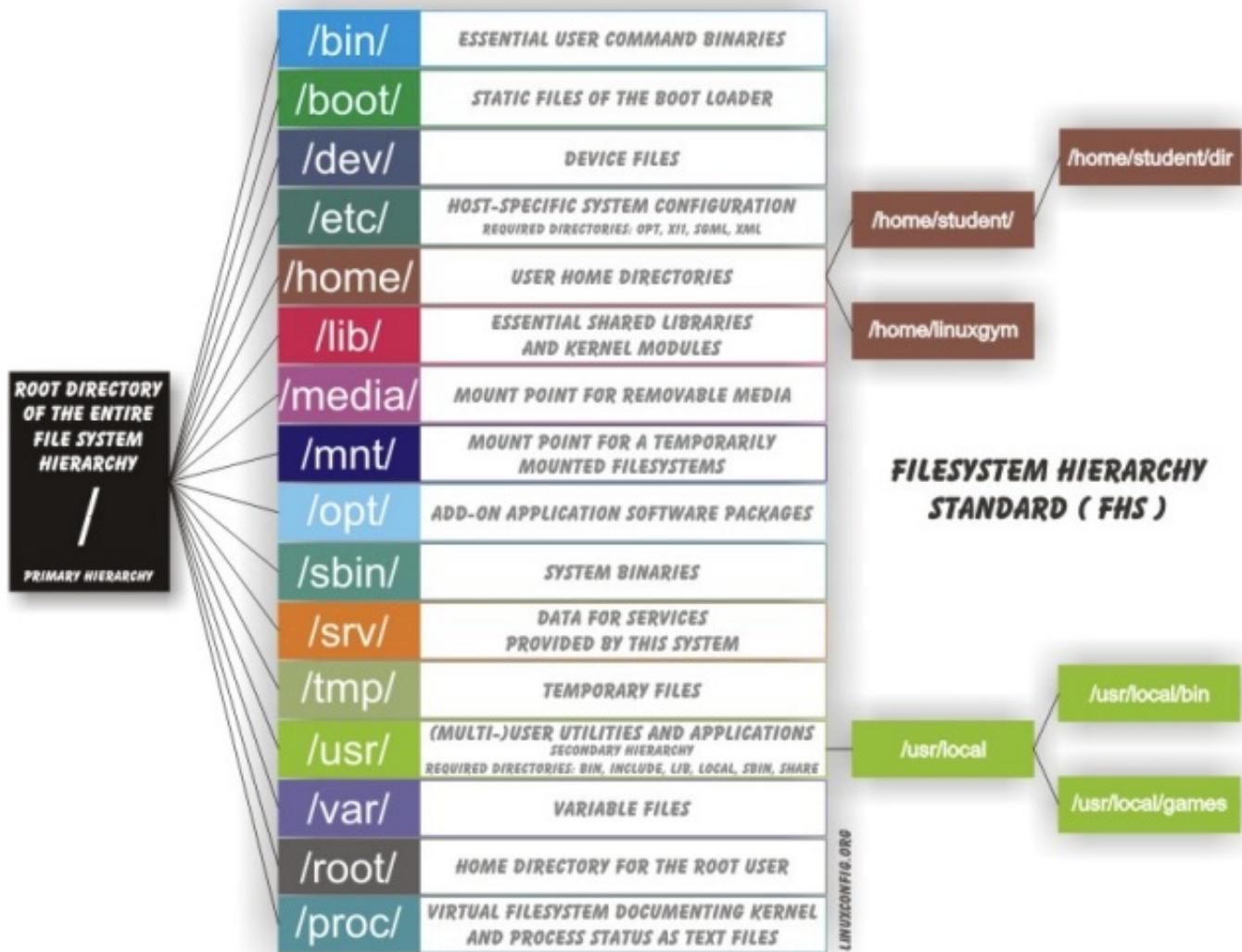


Show directory structure bash script on Linux howto - See hierarchical directory tree structure one liner shell script

Author : admin



If you have Sys Adminned Linux or *Nix OS like, whether for some shell scripting purpose or just for sake of keeping a backup you should have definitely come into some need to list a tree of a directories content in a hierarchical order.

The most obvious way to do that on Linux is by simply using:

1. "tree" command (not installed by default on most Linux distributions so in order to have it on Deb / Debian based Linux do:

```
# apt-get install --yes tree
```

On Fedora / CentOS Redhat Linux (RHEL) etc. install with:

```
# yum --yes install tree
```

By the way for those that needs *tree on FreeBSD / BSD UNIX*, tree is also available on that platform you can install it with:

```
pkg_add -vr tree
```

Then simply check man tree to get idea on how to use it, the easiest way to use the command tree once package is installed is to run **tree** inside directory of choice, i.e.

```
$ cd /somedir
$ tree -a

.
?
?? acpi
? ?? events
? ? ?? powerbtn-acpi-support
? ?? powerbtn-acpi-support.sh
?? adduser.conf
?? adjtime
```

??? aliases
??? alternatives
? ??? ABORT.7.gz -> /usr/share/postgresql/9.5/man/man7/ABORT.7.gz
? ??? aclocal -> /usr/bin/aclocal-1.11
? ??? aclocal.1.gz -> /usr/share/man/man1/aclocal-1.11.1.gz
? ??? ALTER_AGGREGATE.7.gz ->
/usr/share/postgresql/9.5/man/man7/ALTER_AGGREGATE.7.gz
? ??? ALTER_COLLATION.7.gz ->
/usr/share/postgresql/9.5/man/man7/ALTER_COLLATION.7.gz
? ??? ALTER_CONVERSION.7.gz ->
/usr/share/postgresql/9.5/man/man7/ALTER_CONVERSION.7.gz

To get a list of only directories with tree use:

```
$ tree -d /
```

```
? ??? bin
? ? ??? boot
? ? ? ??? grub
? ? ?     ??? locale
? ? ??? disk
? ? ? ??? Books
? ? ? ? ??? 200 E-BOOKS
? ? ? ? ? ??? McGraw-Hill - Windows Server 2003
? ? ? ? ? ??? Oreilly.Access.Cookbook.2nd.Edition-LiB
? ? ? ? ? ??? Oreilly.ActionScript.Cookbook.eBook-LiB
? ? ? ? ? ??? OReilly.ActionScript.The.Definative.Guide.WinAll.Retail-EAT
? ? ? ? ? ??? Oreilly.Active.Directory.2nd.Edition.eBook-LiB
? ? ? ? ? ??? Oreilly.Active.Directory.Cookbook.eBook-LiB
? ? ? ? ? ??? Oreilly.ADO.Dot.NET.Cookbook.eBook-LiB
```

? ? ? ? ? ??? *Oreilly.Amazon.Hacks.eBook-LiB*
? ? ? ? ? ??? *OREILLY.ANT.THE.DEFINITIVE.GUIDE-JGT*
? ? ? ? ? ??? *Oreilly.Apache.Cookbook.eBook-LiB*
? ? ? ? ? ??? *Oreilly.AppleScript.The.Definitive.Guide.eBook-LiB*
? ? ? ? ? ??? *Oreilly.ASP.Dot.NET.In.A.Nutshell.2nd.Edition.eBook-LiB*
? ? ? ? ? ??? *OReilly.Better.Faster.Lighter.Java.Jun.2004.eBook-DDU*
? ? ? ? ? ??? *Oreilly.BLAST.eBook-LiB*
? ? ? ? ? ??? *OReilly.BSD.Hacks.May.2004.eBook-DDU*
? ? ? ? ? ??? *Oreilly.Building.Embedded.Linux.Systems.eBook-LiB*
...

If you have a colorful terminal and you like colors for readability the **-C** option is quite handy

```
$ tree -C /
```

```
desktop:~/projects$ tree -a
.
└── awesomesite
    ├── awesomesite
    │   ├── __init__.py
    │   ├── settings.py
    │   ├── urls.py
    │   └── wsgi.py
    └── .git
        ├── branches
        ├── config
        ├── description
        ├── HEAD
        └── hooks
            ├── applypatch-msg.sample
            ├── commit-msg.sample
            ├── post-update.sample
            ├── pre-applypatch.sample
            ├── pre-commit.sample
            ├── prepare-commit-msg.sample
            ├── pre-rebase.sample
            └── update.sample
        ├── info
        └── objects
            ├── info
            └── pack
        └── refs
            ├── heads
            └── tags
    └── manage.py

12 directories, 17 files
desktop:~/projects$
```

To list the directory tree with permissions included use **tree** cmd like so:

```
$ tree -L 2 -p /usr
```

```
/usr/
???
[drwxr-xr-x] bin
? ???
[-rwxr-xr-x] [
? ???
[lrwxrwxrwx] 2to3 -> 2to3-2.6
? ???
[-rwxr-xr-x] 2to3-2.6
? ???
[-rwxr-xr-x] 411toppm
? ???
[-rwxr-xr-x] 7z
? ???
[-rwxr-xr-x] 7za
? ???
[-rwxr-xr-x] a2p
? ???
[-rwxr-xr-x] ab
? ???
[-rwxr-xr-x] ac
? ???
[lrwxrwxrwx] aclocal -> /etc/alternatives/aclocal
? ???
[-rwxr-xr-x] aclocal-1.11
? ???
[-rwxr-xr-x] acpi
```

Another truly handy option of tree is to list the directory structure index with included file sizes information

```
$ tree -L 2 -sh /bin
```

```
/bin
???
[903K] bash
???
[147K] bsd-csh
???
[ 30K] bunzip2
???
[681K] busybox
???
[ 30K] bzcat
???
[ 6] bzcmp -> bzdiff
???
[2.1K] bzdiff
???
[ 6] bzegrep -> bzgrep
???
[4.8K] bzexe
???
[ 6] bzfgrep -> bzgrep
???
[3.6K] bzgrep
```

```
??? [ 30K] bzip2
??? [ 14K] bzip2recover
??? [ 6] bzless -> bzmore
??? [1.3K] bzmore
??? [ 51K] cat
??? [ 59K] chgrp
??? [ 55K] chmod
??? [ 63K] chown
??? [ 10K] chvt
??? [127K] cp
??? [134K] cpio
??? [ 21] csh -> /etc/alternatives/csh
??? [104K] dash
```

To list a directory tree of a search pattern, lets say all files with .conf extensions use:

```
$ tree -P *.conf
```

```
/etc/ca-certificates.conf [error opening dir]
/etc/dante.conf [error opening dir]
/etc/debconf.conf [error opening dir]
/etc/deluser.conf [error opening dir]
/etc/discover-modprobe.conf [error opening dir]
/etc/fuse.conf [error opening dir]
/etc/gai.conf [error opening dir]
/etc/gpm.conf [error opening dir]
/etc/gssapi_mech.conf [error opening dir]
/etc/hdparm.conf [error opening dir]
/etc/host.conf [error opening dir]
/etc/idmapd.conf [error opening dir]
/etc/inetd.conf [error opening dir]
/etc/insserv.conf [error opening dir]
/etc/irssi.conf [error opening dir]
/etc/kernel-img.conf [error opening dir]
/etc/ld.so.conf [error opening dir]
```

```
/etc/libao.conf [error opening dir]
/etc/libaudit.conf [error opening dir]
/etc/logrotate.conf [error opening dir]
/etc/memcached.conf [error opening dir]
/etc/mke2fs.conf [error opening dir]
/etc/mongodb.conf [error opening dir]
/etc/mtools.conf [error opening dir]
/etc/multitail.conf [error opening dir]
/etc/nsswitch.conf [error opening dir]
/etc/ntp.conf [error opening dir]
/etc/ocamlfind.conf [error opening dir]
```

tree -I option does exclude all patterns you don't want tree to list

Here are few other **tree** useful options:

- **tree -u /path/to/file** - displays the users owning the files
- **tree -g /path/to/file** - display the groups owning the files
- **tree -a /path/to/file** - display the hidden files/folders
- **tree -d /path/to/file** - display only the directories in the hierarchy

However there might be some cases where you have to support a Linux server or you just have to write a script for a non-root user and you might not have the permissions to install the tree command to make your life comfortable. If that's the case then you can still **use a couple of command line tools and tricks (assuming you have permissions) to list a log a directory / files and subdirectories tree structure in a hierarchical tree like command order**

2. Print a list of all sub-directories and files within a directory tree

To print all directories within any path of choice on a server use

```
$ find /path/ -type d -print
```

```
...
```

To print all files within a root filesystem hierarchically with find command

Another way to do it in a more beautiful output is by using find in conjunction with awk

```
$ find . -type d -print 2>/dev/null|awk '!/.$/ {for (i=1;i
```