Ubuntu files

Autor:
Data de publicació: 26-12-2015

Learn Basics of Ubuntu directory structure

Hellow my name is Vipul kumar. And i am going to tell you about Basics of Ubuntu Linux, for new comers .1st let me tell you, you shud know about Directory structure in ubuntu linux.

In Ubuntu Linux, this is the basic format

/Folder1or Directory/subfolder or sub-Directory/file.txt

it's something like that in my ubuntu 12.04

The Main Directories are in Ubuntu Linux

11.jpg

1. / – Root
2. /bin – User Binaries
3. /sbin – System Binaries
4. /etc – Configuration Files
5. /dev – Device Files
6. /proc – Process Information
7. /var – Variable Files
8. /tmp – Temporary Files
9. /usr – User Programs
10. /home – Home Directories
11. /boot – Boot Loader Files
12. /lib – System Libraries
13. /opt – Optional add-on Applications
14. /mnt – Mount Directory
15. /media – Removable Media Devices
16. /srv – Service Data

Let me explain each of them

1. / – Root

The root directory. The starting point of your directory structure. This is where the Linux system begins. Every other file and directory on your system is under the root directory. Usually the root directory contains only subdirectories, so it’s a bad idea to store single files directly under root.Only root user has write privilege under this directory.Please note that /root is root user’s home directory, which is not same as / .

you can go inside root directory via

cd /

#ls

1.png

2. /bin – User Binaries, /usr/bin

These two directories contain a lot of programs (binaries, hence the directory’s name) for the system. Common linux commands you need to use in single-user modes are located under this directory.The /bin directory contains the most important programs that the system needs to operate, such as the shells, ls, grep, and other essential things For example: ps, ls, ping, grep, cp. /usr/bin in turn contains applications for the system’s users. However, in some cases it really doesn’t make much difference if you put the program in /bin or /usr/bin.

cd /usr/bin

ls

######################

3. /sbin – System Binaries

Most system administration programs are stored in these directories. In many cases you must run these programs as the root user.For example: iptables, reboot, fdisk, ifconfig, swapon. It also contains binary executables you can check by

cd /sbin

ls

##################

4. /etc – Configuration FilesThe configuration files for the Linux system.It contains configuration files required by all programs Most of these files are text files and can be edited by hand. Some interesting
stuff in this directory:

cd /etc/

ls

#######################

5. /dev – Device Files

The devices that are available to a Linux system. Remember that in Linux, devices are treated like files and you can read and write devices like they were files. For example, /dev/fd0 is your first floppy drive,/dev/cdrom is your CD drive, /dev/hda is the first IDE hard drive, and so on. All the devices that a Linux kernel can understand are located under /dev, and that’s why it contains hundreds of entries.These include terminal devices, usb, or any device attached to the system.Example /dev/tty1, /dev/usbmon0

cd /dev

ls

##################

6. /proc – Process Information

This is a special directory. Well, actually /proc is just a virtual directory, because it doesn't exist at all! It contains some info about the kernel itself. There's a bunch of numbered entries that correspond to all processes running on the system, and there are also named entries that permit access to the current configuration of the system. Many of these entries can be viewed.by #cd /proc and then type#lsThis is a pseudo filesystem contains information about running process. For example: /proc/{pid} directory contains information about the process with that particular pid.Last but not least it Contains information about system process.

cd /proc/

ls

##########################

7. /var – Variable Files

This directory contains variable data that changes constantly when the system is running.

cd /var/

ls

#######################

8. /tmp – Temporary Files

Programs can write their temporary files here.

cd /tmp

ls

##########

9. /usr – User Programs

This directory contains user applications and a variety of other things for them, like their source codes, and pictures, docs, or config files they use. /usr is the largest directory on a Linux system, and some people like to have it on a separate partition. Some interesting stuff in /usr

#cd /usr

ls

#########3

10. /home – Home Directories

This is where users keep their personal files. Every user has their own directory under /home, and usually it’s the only place where normal users are allowed to write files. You can configure a Linux system so that normal users can’t even list the contents of other users’ home directories. This means that if your family members have their own user accounts on your Linux system, they won’t see all the w4r3z you keep in your home directory. Or you Porn stuff hheeh

cd /home/

ls

##########

11. /boot – Boot Loader FilesAs the name suggests, this is the place where Linux keeps information that it needs when booting up. For example, this is where the Linux kernel is kept. If you list the contents of /boot, you'll see a file called vmlinuz – that's the kernel.Kernel initrd, vmlinux, grub files are located under /boot For example: initrd.img-2.6.32-24-generic, vmlinuz-2.6.32-24-generic

cd /boot/

ls

##########

12. /lib – System Libraries

The shared libraries for programs that are dynamically linked. The shared libraries are similar to DLL's on Winblows

##########

13. /opt – Optional add-on Applications

opt stands for optional.Contains add-on applications from individual vendors.add-on applications should be installed under either /opt/ or /opt/ sub-directory

##########

14. /mnt – Mount Directory

Temporary mount directory where sysadmins can mount filesystems.This directory is used for mount points. The different physical storage devices (like the hard disk drives, floppies, CD-ROM's) must be attached to some directory in the file system tree before they can be accessed. This attaching is called mounting, and the directory where the device is attached is called the mount point.The /mnt directory contains mount points for different devices, like /mnt/floppy for the floppy drive, /mnt/cdrom for the CD-ROM, and so on. However, you're not forced to use the /mnt directory for this purpose, you can use whatever directory you wish. Actually in some distros, like Debian and SuSE, the default is to use/floppy and /cdrom as mount points instead of directories under /mnt

cd/mnt/

ls

######################

15. /media – Removable Media Devices

Temporary mount directory for removable devices.For examples, /media/cdrom for CD-ROM; /media/floppy for floppy drives; /media/cdrecorder for CD writer

cd /media/

ls

#########################

16./srv – Service Data

srv stands for service.Contains server specific services related data.For example, /srv/cvs contains CVS related data.

cd /srv/

ls

What next?

If you’re completely new to Ubuntu Linux, you might want to learn some commands for moving around in the file system, viewing text files, or manipulating the files. In that case I suggest you take a look at my, new articles coming at my Linux command line section sTAY TuNe

---

clear Clears the current screen of all text.

ls Lists the current directory.

It's like the MS-DOS dir command.

ls dir Lists the specified directory.

cat file Concatenates (adds) the contents of the file to your screen.

It's like the MS-DOS type command.

more file Displays the file one page at a time.

tail file Displays the file one page at a time from the end of the file.

pwd Shows name of the current directory.

cd dir Changes to the specified directory.

It's like the MS-DOS chdir command.

rm file or dir Deletes the specified file or directory.

Linux commands are shown in this table.

CommandDescription

cat [filename]
Display file’s contents to the standard output device (usually your monitor).

cd /directorypath
Change to directory.

chmod [options] mode filename
Change a file’s permissions.

chown [options] filename
Change who owns a file.

clear
Clear a command line screen/window for a fresh start.

cp [options] source destination
Copy files and directories.

date [options]
Display or set the system date and time.

df [options]
Display used and available disk space.

du [options]
Show how much space each file takes up.

file [options] filename
Determine what type of data is within a file.

find [pathname] [expression]
Search for files matching a provided pattern.

grep [options] pattern [filesname]
Search files or output for a particular pattern.

kill [options] pid
Stop a process. If the process refuses to stop, usekill -9 pid.

less [options] [filename]
View the contents of a file one page at a time.

ln [options] source [destination]
Create a shortcut.

locate filename
Search a copy of your filesystem for the specified filename.

lpr [options]
Send a print job.

ls [options]
List directory contents.

man [command]
Display the help information for the specified command.

mkdir [options] directory
Create a new directory.

mv [options] source destination
Rename or move file(s) or directories.

passwd [name [password]]
Change the password or allow (for the system administrator) to change any password.

ps [options]
Display a snapshot of the currently running processes.

pwd
Display the pathname for the current directory.

rm [options] directory
Remove (delete) file(s) and/or directories.

rmdir [options] directory
Delete empty directories.

ssh [options] user@machine
Remotely log in to another Linux machine, over the network. Leave an ssh session by typing exit.

su [options] [user [arguments]]
Switch to another user account.

tail [options] [filename]
Display the last n lines of a file (the default is 10).

tar [options] filename
Store and extract files from a tarfile (.tar) or tarball (.tar.gz or .tgz).

top
Displays the resources being used on your system. Press q to exit.

touch filename
Create an empty file with the specified name.

who [options]
Display who is logged on.

Continguts

Instructions
Ubuntuzilla repository
Manual Installation
Uninstallation
Restore your old profile
Rationale

This page describes how to run Mozilla builds of Firefox in Ubuntu. This can be useful if you want to check whether a bug exists in upstream versions of Firefox, or if you prefer to run the official Mozilla releases.

Instructions

Unless you have a specific reason to run the Mozilla build, it's recommended to use the packages from the official Ubuntu repositories.
See the recommended procedure for getting Firefox.

Ubuntuzilla repository

An easy way of installing a Mozilla build is to use the Ubuntuzilla project repository. Ubuntuzilla is for ordinary users that want to use the latest Mozilla build as their default browser.

The Ubuntuzilla project's repository contains .deb repacks of the unmodified official Mozilla builds of Firefox, Thunderbird, and SeaMonkey. Just add the repository to your sources, and you'll be kept up to date on the latest Mozilla releases through the package manager.

See the Ubuntuzilla website for details and usage instructions.

Manual Installation
The following instructions will install Firefox in your home directory, and will use a different profile for Mozilla and Ubuntu builds of Firefox.

Go to the Firefox download page

Download Firefox for Linux

Extract the file you downloaded into ~/firefox (e.g. by doing tar jxf firefox*.tar.bz2 -C $HOME)

Make a backup of your ~/.mozilla directory (e.g. by doing tar zcPf ~/mozilla-backup-$(date +%s).tar.gz ~/.mozilla)

Next, if this is the first time you have installed a Mozilla build, do the following:

Quit any instances of Firefox you have open.

Run ~/firefox/firefox -ProfileManager to start the Firefox profile manager

Create a new profile called mozilla-build

Make sure that the default profile is still selected

Click exit to close the profile manager (do not start Firefox)

Run the following shell script:

```
mkdir ~/bin
cat > ~/bin/firefox <<END
#!/bin/bash

exec "$HOME/firefox/firefox" -P mozilla-build "$@"
```

END
chmod 755 ~/bin/firefox

The firefox command in your ~/bin directory will now run Firefox with the mozilla-build profile.

Uninstallation

Delete the ~/firefox directory and the ~/bin/firefox file

Run firefox -ProfileManager and delete mozilla-build

This completely uninstalls the Mozilla build of Firefox. If you install a Mozilla build in future, you will need to perform all of the instructions above, including the first time instructions.

Restore your old profile
To restore the backup of your profile, do the following:

Close any open Firefox windows

Rename ~/.mozilla (e.g. mv ~/.mozilla ~/.mozilla-broken)

Extract your archive (e.g. by doing tar zxPf ~/mozilla-backup-DATE.tar.gz

This will completely reset your Firefox profile to the point when you made the backup.

Rationale
There are many ways to run Mozilla builds alongside Ubuntu builds, depending on your needs. This section will discuss some of the choices made in the above instructions.

The guide recommends that you back your Mozilla profile up. Regular backups are always recommended, especially before a major software upgrade. Special precaution is necessary here because new versions of Firefox can make backwards-incompatible changes to profiles. For example, you should avoid using Firefox 3 on a profile that's been used with Firefox 3.51.

The guide recommends extracting Firefox into your home directory. This is because extracting to another location (such as /usr/local) would require super-user permissions, without any benefit in the single-user case. If you want to provide new versions of Firefox on a multiple-user system, you are urged to use the recommended procedure. Installing non-Ubuntu software on a multi-user system massively increases your administrative burden - for example, you put your users at risk unless you manually update programs every time there's an update.

The guide recommends that you create a mozilla-build profile for the downloaded version of Firefox. This is partially because of the above-mentioned compatibility issues, and partially because debugging Firefox is much easier with a fresh profile.

CategoryInternet

http://irclogs.ubuntu.com/2009/07/08/%23ubuntu-mozillateam.html (search for profile damage) (1)

Known Issues

Disable Apport at Boot to turn off error reporting http://howtoubuntu.org/how-to-disabl...ting-in-ubuntu
sudo gedit /etc/default/apport
Change enabled=1 to enabled=0 save the file and close.

If Firefox windows open off screen or are too large to use, you may need to reset Firefox's controls and toolbars.

1. Close down Firefox completely: On the Firefox window, click the File menu then select Exit.
2. From a terminal window type: firefox -safe-mode
3. Firefox should start up with a Firefox Safe Mode dialog with options.
4. Check mark Reset toolbars and controls.
5. Click Make Changes and Restart to restart Firefox

Add Missing Privacy Settings in Ubuntu Gnome Desktop
sudo apt-get install activity-log-manager-control-center

Note: Activity Log Manager will show up under Accessories or by doing a search for the application.

All Startup Applications are not visible
cd /etc/xdg/autostart
sudo sed --in-place 's/NoDisplay=true/NoDisplay=false/g' *.desktop

Reverse changes...
sudo sed --in-place 's/NoDisplay=false/NoDisplay=true/g' *.desktop

Useful Commands

Searching for Packages: sudo apt-cache search some_string
Show Package Info: sudo apt-cache showpkg xxx
Show Package Dependencies: sudo apt-cache depends xxx
Install: sudo apt-get install xxx
Re-Install: sudo apt-get --reinstall install xxx
Remove: sudo apt-get remove xxx
Remove All (configs too): sudo apt-get remove --purge xxx
Upgrade: sudo apt-get -u upgrade
Show Upgrades: sudo apt-show-versions -u
Show All Installed Packages: dpkg --list
Find Package by File Name: sudo apt-file search /bin/ping
Find filenames in a Package: sudo apt-file list xxx

Updating the apt-file Cache: sudo apt-file update
Info on Installed Package: aptitude show xxx
System Hardware Info: sudo lshw > hardware.txt
Clear bash history cat /dev/null > ~/.bash_history && history -c && exit

Clean up your system and free up space with sudo apt-get clean and sudo apt-get autoremove

If you're curious (like me) or have the need to know uname -a && cat /etc/*release in a terminal window will tell you the kernel version and release date, the distro id/release/codename/description.

More to come...

Friends,

I'm more than willing to include your favorite applications (this is your post as much as mine). I do ask that you keep the description brief and on a general level with the degree of difficulty at a minimum (for new users) when making recommendations. I started posting "Ubuntu Desktop Computing Made Easy" a while back (mainly out of frustration) while trying to get my computer to perform as expected without hours of setup, search and configuration. The primary goal of this post is to make it "easy" for anyone to setup an Ubuntu Desktop Computer with everything they need or want within a short period of time ensuring the overall experience is an enjoyable one.

Note: This post can be found on Google by searching for Ubuntu Desktop Computing Made Easy. If you would like to link to this post or publish it elsewhere please give credit where credit is due.

Kind regards,

TrakerJon

Last edited by TrakerJon; May 2nd, 2014 at 0