Beginners: Learn Linux

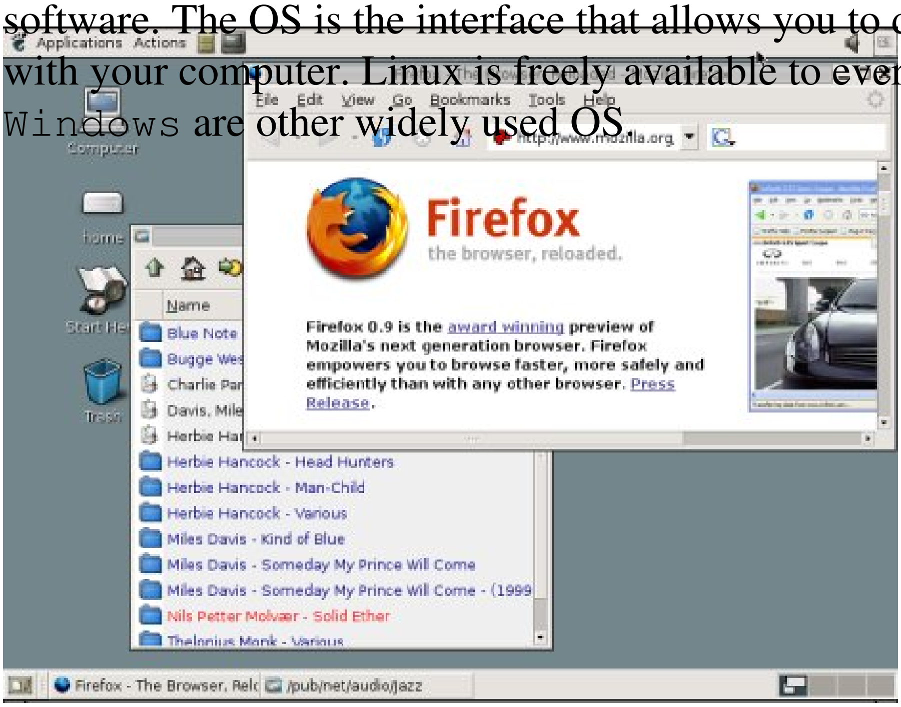
Beginners: Learn Linux v0.99.3 (en), xiando

A beginners guide to Linux for those with little or no computer experience.

1. What is Linux?
2. Understanding files and folders
3. Understanding users and permissions
4. Who and what is root
5. Opening a command shell / terminal
6. Your first Linux commands
    - ♦ 6.1. ls - short for list
    - ♦ 6.2. pwd - print name of current/working directory
    - ♦ 6.3. cd - Change directory
7. The basic commands
    - ♦ 7.1. chmod - Make a file executable
    - ♦ 7.2. df - view filesystem disk space usage
    - ♦ 7.3. du - View the space used by files and folders
    - ♦ 7.4. mkdir - makes folders
    - ♦ 7.5. passwd - changes your login password
        - ◊ 7.5.1. KDE
    - ♦ 7.6. rm - delete files and folders, short for remove
    - ♦ 7.7. ln - make symbolic links
    - ♦ 7.8. tar archiving utility - tar.bz2 and tar.gz
        - ◊ 7.8.1. tar files (.tar.gz)
        - ◊ 7.8.2. bzip2 files (.tar.bz2)

1. What is Linux?

Linux is a free Unix-type operating system for computer devices. The operating system is what makes the hardware work together with the software. The OS is the interface that allows you to do the things you want with your computer. Linux is freely available to everyone. `OS X` and `Windows` are other widely used OS.



Firefox 0.9 is the award winning preview of Mozilla's next generation browser. Firefox empowers you to browse faster, more safely and efficiently than with any other browser. Press Release.

Linux gives you a graphical interface that makes it easy to use your computer, yet it still allows those with know-how to change settings by adjusting `0` to `1`.

It is only the kernel that is named Linux, the rest of the OS are GNU tools. A package with the kernel and the needed tools make up a `Linux distribution`. Mandrake , SUSE Linux, Gentoo and Redhat are some of the many variants. GNU/Linux OS can be used on a large number of boxes, including i386+ , Alpha, PowerPC and Sparc.

2. Understanding files and folders

Linux is made with one thought in mind: **Everything is a file**.

A blank piece of paper is called a file in the world of computers. You can use this piece of paper to write a text or make a drawing. Your text or drawing is called information. A computer file is another way of storing your information.

If you make many drawings then you will eventually want to sort them in different piles or make some other system that allows you to easily locate a given drawing. Computers use folders to sort your files in a hieratic system.

A file is an element of data storage in a file system (file systems manual page). Files are usually stored on harddrives, cdroms and other media, but may also be information stored in RAM or links to devices.

To organize our files into a system we use **folders**. The lowest possible folder is root / where you will find the user homes called /home/.

```
/
/home/
/home/mom/
/home/dad/
```

Behind every configurable option there is a simple human-readable text file you can hand-edit to suit your needs. These days most programs come with nice GUI (graphical user interface) like Mandrakes Control Center and Suses YAST that can smoothly guide you through most configuration. Those who choose can gain full control of their system by manually adjusting the configuration files from foo=yes to foo=no in an editor.

Almost everything you do on a computer involves one or more files stored locally or on a network.

Your filesystems lowest folder root / contains the following folders:

/bin	Essential user command binaries (for use by all users)
/boot	Static files of the boot loader, only used at system startup
/dev	Device files, links to your hardware devices like /dev/sound, /dev/input/js0 (joystick)
/etc	Host-specific system configuration
/home	User home directories. This is where you save your personal files
/lib	Essential shared libraries and kernel modules
/mnt	Mount point for a temporarily mounted filesystem like /mnt/cdrom
/opt	Add-on application software packages
/usr	/usr is the second major section of the filesystem. /usr is shareable, read-only data. That means that /usr should be shareable between various FHS-compliant hosts and must not be written to. Any information that is host-specific or varies with time is stored elsewhere.
/var	/var contains variable data files. This includes spool directories and files, administrative and logging data, and transient and temporary files.
/proc	System information stored in memory mirrored as files.

The only folder a normal user needs to use is `/home/you/` - this is where you will be keeping all your documents.

```
/home/elvis/Documents
/home/elvis/Music
/home/elvis/Music/60s
```

Files are case sensitive, "myfile" and "MyFile" are two different files.

For more details, check out:

3. Understanding users and permissions

Linux is based on the idea that everyone using a system has their own username and password.

Every file belongs to a **user** and a **group**, and has a set of given attributes (read, write and executable) for users, groups and all (everybody).

A file or folder can have permissions that only allows the user it belongs to to read and write to it, allowing the group it belongs to to read it and at the same time all other users can't even read the file.

4. Who and what is root

Linux has one special user called `root` (this is the user name). Root is the "system administrator" and has access to all files and folders. This special user has the right to do anything.

You should *never* log on as this user unless you actually need to do something that requires it!

Use `su -` to temporary become root and do the things you need, again: never log into your sytem as root!

Root is only for system maintenance, **this is not a regular user** (LindowsOS don't have any user management at all and uses root for everything, this is a *very bad* idea!).

You can execute a command as root with:

```
su -c 'command done as root'
```

Gentoo Linux: Note that on Gentoo Linux only users that are member of the **wheel** group are allowed to su to root.

5. Opening a command shell / terminal

To learn Linux, you need to learn the shell command line in a terminal emulator.

In KDE: `K` -> `System` -> `Konsoll` to get a command shell)

Pressing **CTRL-ALT-F1** to **CTRL-ALT-F6** gives you the console command shell windows, while **CTRL-ALT-F7** gives you XFree86 (the graphical interface).

xterm (manual page) is the standard XFree console installed on all boxes, run it with **xterm** (press ALT F2 in KDE and Gnome to run commands).

Terminals you probably have installed:

- **xterm** http://dickey.his.com/xterm/
- **konsole** (KDEs terminal)
- **gnome-terminal** (Gnomes terminal)

Non-standard terminals should install:

- **rxvt** http://www.rxvt.org/
- **aterm** http://aterm.sourceforge.net

6. Your first Linux commands

Now you should have managed to open a terminal shell and are ready to try your first Linux commands. Simply ask the computer to do the tasks you want it to using it's language and press the enter key (the big one with an arrow). You can add a `&` after the command to make it run in the background (your terminal will be available while the job is done). It can be practical to do things like moving big divx movies as a background process: `cp movie.avi /pub &`. Jobs - the basics of job control

6.1. ls - short for list

ls lists the files in the current working folder. This is probably the first command to try out. It as a number of options described on the ls manpage.

Examples:

```
ls
```

```
ls -al --color=yes
```

6.2. pwd - print name of current/working directory

`pwd` prints the fully resolved name of the current (working) directory. pwd manpage.

6.3. cd - Change directory

cd stands for change (working) directory and that's what it does. The folder below you (unless you are in /, where there is no lower directory) is called "..".

To go one folder down:

```
cd ..
```

Change into the folder Documents in your current working directory:

```
cd Documents
```

Change into a folder somewhere else:

```
cd /pub/video
```

The / in front of pub means that the folder pub is located in the / (lowest folder).

7. The basic commands

7.1. chmod - Make a file executable

To make a file executable and runnable by any user:

```
chmod a+x myfile
```

Refer to the chmod manual page for more information.

7.2. df - view filesystem disk space usage

```
df -h
```

```
  Filesystem Size  Used Avail Use% Mounted on
  /dev/hda3   73G   67G  2.2G  97% /
  tmpfs      2.0M   24K  2.0M   2% /mnt/.init.d
  tmpfs      252M     0  252M   0% /dev/shm
```

The flags: -h, --human-readable Appends a size letter such as M for megabytes to each size.

df manpage

7.3. du - View the space used by files and folders

Use **du** (Disk Usage) to view how much space files and folders occupy. Read the du manual page for flags and usage.

du is a part of fileutils.

Example du usage:

```
du -sh Documents/
409M    Documents
```

7.4. mkdir - makes folders

Folders are created with the command **mkdir**:

```
mkdir folder
```

To make a long path, use mkdir -p :

```
mkdir -p /use/one/command/to/make/a/long/path/
```

Like most programs mkdir supports -v (verbose). Practical when used in scripts.

You can make multiple folders in *bash* and other shells with {folder1,folder2} :

```
mkdir /usr/local/src/bash/{old,new,dist,bugs}
```

mkdir manual page

The command `rmdir` removes folders.

7.5. passwd - changes your login password

To change your password in Linux, type:

```
passwd
```

The **root** user can change the password of any user by running passwd with the user name as argument:

```
passwd jonny
```

will change jonnys password. Running passwd without arguments as root changes the root password.

If you need to add several new users and give them password you can use a handy program like Another Password Generator to generate a large set of "random" passwords.

7.5.1. KDE

From KDE you can change your password by going:

- `K` -> `Settings` -> `Change Password`
- `K` -> `Settings` -> `Control Center` -> `System Administration` -> `User Account`

7.6. rm - delete files and folders, short for remove

Files are deleted with the command **rm**:

```
  rm /home/you/youfile.txt
```

To delete folders, use **rm** together with **-f** (Do not prompt for confirmation) and **-r** (Recursively remove directory trees):

```
  rm -rf /home/you/foo/
```

Like most programs rm supports -v (verbose).

rm manual page

7.7. ln - make symbolic links

A symbolic link is a "file" pointing to another file.

To make a symbolic link :

```
  ln /original/file /new/link
```

This makes /original/file and /new/link the same file - edit one and the other will change. The file will not be gone until both /original/file and /new/link are deleted.

You can only do this with files. For folders, you must make a "soft" link.

To make a soft symbolic link :

```
  ln -s /original/file /new/link
```

Example:

```
  ln -s /usr/src/linux-2.4.20 /usr/src/linux
```

Note that -s makes an "empty" file pointing to the original file/folder. So if you delete the folder a symlink points to, you will be stuck with a dead symlink (just rm it).

ln manual page

7.8. tar archiving utility - tar.bz2 and tar.gz

tar (manual page) is a very handle little program to store files and folders in archives, originally made for tapestreamer backups. Tar is usually used together with gzip (manual page) or bzip2 (manual page), comprepssion programs that make your .tar archive a much smaller .tar.gz or .tar.bz2 archive.

kde

You can use the program `ark` (`K` -> `Utilities` -> `Ark`) to handle archives in KDE. Konqueror treats file archives like normal folders, simply click on the archive to open it. The archive becomes a virtual folder that can be used to open, add or remove files just as if you were working with a normal folder.

7.8.1. tar files (.tar.gz)

To untar files:

```
  tar xvzf file.tar.gz
```

To tar files:

```
  tar cvzf file.tar.gz filedir1 filedir2 filedir2...
```

Note: A .tgz file is the same as a .tar.gz file. Both are also often refered to as tarballs.

The flags: z is for gzip, v is for verbose, c is for create, x is for extract, f is for file (default is to use a tape device).

7.8.2. bzip2 files (.tar.bz2)

To unpack files:

```
  tar xjvf file.tar.bz2
```

To pack files:

```
  tar cvjf file.tar.bz2 filedir1 filedir2 filedir2...
```

The flags: Same as above, but with j for for bzip2

You can also use bunzip2 file.tar.bz2&nbsp;, will turn it into a tar.

For older versions of tar, try tar -xjvf or -xYvf or -xkvf to unpack.There's a few other options it could be, they couldn't decide which switch to use for bzip2 for a while.

How to untar an entire directory full or archives?

.tar:

```
for i in `ls *.tar`; do tar xvf $i; done
```

.tar.gz: `for i in `ls *.tar.gz`; do tar xvfz $i; done`

.tar.bz2: `for i in `ls *.tar.bz2`; do tar xvfj $i; done`

> Linux Reviews >
Beginners: Learn Linux