

- I. What is Linux?
- II. Trying it out
- III. Installing
- IV. What to do now
- V. The Console

Intro:

This tutorial is written with the total Linux n00b in mind.
I've seen too many n00bs get totally left in the dark by asking what the best distro is. They seem to only get flooded with too many answers in so short a time. I'm a little bit of a n00b too, so I know how it feels. I will cover a grand total of two basic distros. You may learn to strongly prefer other ones (I do!) but this is just to get you started. I touch on a number of topics that would be impossible to go into in depth in one tutorial, so I encourage you to actively seek out more about the concepts I make reference to.

I. What is Linux?

Linux is basically an operating system (OS for short). The Windows machine you're (probably) using now uses the Mcft Windows operating system.

Ok, so what's so different about Linux?

Linux is part of a revolutionary movement called the open-source movement. The history and intricacies of that movement are well beyond the scope of this tutorial, but I'll try and explain it simply. Open source means that the developers release the source code for all their customers to view and alter to fit what they need the software to do, what they want the software to do, and what they feel software should do. Linux is a programmer's dream come true, it has the best compilers, libraries, and tools in addition to its being open-source. A programmer's only limit then, is his knowledge, skill, time, and resolve.

What is a distro?

A distro is short for a distribution. It's someone's personal modification or recreation of Linux.

What do you mean by distros? I just want Linux!

Since Linux is open source, every developer can write his own version. Most of those developers release their modifications, or entire creations as free and open source. A few don't and try to profit from

their product, which is a topic of moral debate in the Linux world. The actual Linux is just a kernel that serves as a node of communication between various points of the system (such as the CPU, the mouse, the hard drive etc.). In order to use this kernel, we must find a way to communicate with it. The way we communicate is with a shell. Shells will let us enter commands in ways that make sense to us, and send those commands to the kernel in ways that makes sense to it. The shell most Linux's use it the BASH shell (Bourne Again SHell). The kernel by itself will not do, and just a shell on top of the kernel won't either for most users; we are then forced to use a distribution.

What distro is best?

This is not the question you want to ask a large number of people at one time. This is very much like asking what kind of shoe is best, you'll get answers anywhere from running shoes, hiking boots, cleats, to wingtips. You need to be specific about what you plan on using Linux for, what system you want to use it on, and many other things. I will cover two that are quick and easy to get running. They may not be the best, or the quickest, or the easiest, or the most powerful, but this is a guide for getting started, and everyone has to start somewhere.

How much does it cost?

computer + electricity + internet + CD burner and CDs = Linux
I'll let you do your own math.

Note however that a few do charge for their distros, but they aren't all that common, and can be worked around. Also, if you lack internet access or a CD burner or CDs or you just want to, you can normally order CDs of the distro for a few dollars apiece.

II. Trying it out.

Wouldn't it stink if you decide to wipe out your hard drive and install Linux as the sole operating system only to learn that you don't know how to do anything and hate it? Wouldn't it be better to take a test drive? 95 out of a 100 of you know where I'm heading with this section and can therefore skip it. For those of you who don't know, read on.

There are many distros, and most distros try to have something that makes them stand out. Knoppix was the first live-CD distro. Although most of the other main distros have formed their own live-CDs, Knoppix is still the most famous and I will be covering how to acquire it.

A live-CD distro is a distribution of Linux in which the entire OS can be run off of the CD-ROM and your RAM. This means that no installation is required and the distro will not touch your hard disk or current OS (unless you tell it to). On bootup, the CD will automatically detect your hardware and launch you into Linux. To get back to Windows, just reboot and take the CD out.

Go to the Knoppix website (www.knoppix.com). Look around some to get more of an idea on what Knoppix is. When you're ready, click Download. You'll be presented with a large amount of mirrors, some of which have ftp and some of which have http also.

note: the speed of the mirrors vary greatly, and you may want to change mirrors should your download be significantly slow.

Choose a mirror. Read the agreement and choose accept. You'll probably want to download the newest version and in your native language (I'll assume English in this tutorial). So choose the newest file ending in -EN.iso

note: you might want to also verify the md5 checksums after the download, if you don't understand this, don't worry too much. You just might have to download it again should the file get corrupted (you'll have to anyway with the md5). Also, a lot of times a burn can be botched for who-knows what reason. If the disk doesn't work at all, try a reburn.

Once the .iso file is done downloading, fire up your favorite CD-burning software. Find the option to burn a CD image (for Nero, this is under copy and backup) and burn it to a disk. Make sure you don't just copy the .iso, you have to burn the image, which will unpack all the files onto the CD.

Once the disk is done, put it in the CD-ROM drive and reboot the computer. While your computer is booting, enter CMOS (how to get to CMOS varies for each computer, some get to it by F1 or F2 or F3, etc.) Go to the bootup configuration and place CD-ROM above hard disk. Save changes and exit. Now, Knoppix will automatically start. You will be presented with a boot prompt. Here you can input specific boot parameters (called cheatcodes), or just wait and let it boot up using the default.

note: Sometimes USB keyboards do not work until the OS has somewhat booted up. Once you're actually in Knoppix, your USB keyboard should work, but you may not be able to use cheatcodes. If you need to, attach a PS/2 keyboard temporarily. Also, if a particular aspect of hardware detection does not work, look for a cheatcode to disable it. Cheatcodes can be found on the Knoppix website in text format (or in HTML at www.knoppix.net/docs/index.php/CheatCodes).

Upon entering the KDE desktop environment, spend some time exploring around. Surf the web, get on IM, play some games, explore the filesystem, and whatever else seems interesting. When your done, open up the console (also called terminal, xterm, konsole, or even shell) and get ready for the real Linux. See section V for what to do from here.

note: to function as root (or the superuser) type su.

It's not entirely necessary that you are a console wizard at this point

(although you will need to be sooner or later), but a little messing around won't hurt.

Just as there are many Linux distros, so there are also many types of Knoppix. I won't go into using any of them, but they should all be somewhat similar. Some of them include: Gnoppix, Knoppix STD, Morphix, and PHLAK. Other distros also have live-CDs.

III. Installing

I will guide you through the installation of Fedora Core 2. The reason I chose Fedora is because it contains the Anaconda installer, which is a very easy installer.

Download the discs from here:

<http://download.fedora.redhat.com/pub/fedora-core/2/i386/iso/>

If the link doesn't work, then go to www.redhat.com and navigate your way to downloading Fedora (odds are your architecture is i386).

You will want to download the FC2-i386-disc1.iso and burn it using the method for Knoppix. Do the same for all the discs.

Note: do NOT download the FC2-i386-SRPMS-disc1.iso files.

Now, once you're ready, insert disc 1 into the drive and reboot.

The installer should come up automatically (if not, then see the Knoppix section on CMOS).

Note: installer may vary depending on version. Follow directions best you can using your best judgement.

1. Language: choose English and hit enter
2. Keyboard: choose us (probably) and hit enter
3. Installation media: choose local CDROM (probably) and hit enter
4. CD test: you can choose to test or skip
5. Intro: click next
6. Monitor: choose your monitor to the best of your ability, if you're unsure, choose one of the generic ones
7. Installation type: choose whichever you want (default should be fine)
8. Partition: choose to automatically partition (unless you know what you're doing)
9. Partition: the default partitions should suffice
10. Boot loader: choose your boot loader (grub for default)
11. Network settings: choose the correct settings for your network (generally, don't mess with anything unless you know what you're doing)
12. Firewall: you can choose a firewall if you want to
13. Language support: choose any additional language support you want
14. Time zone: pick your time zone
15. Root password: set your root password (root is the admin, or superuser; you want it to be very secure)
16. Packages: choose which packages you want to install. For hard drives over 10 gigs, you can go ahead and choose all packages (depending on how much disk space you plan on taking up later, note that most everything you'll need is a package: the exception being large media files). You will generally want to install all the packages you think you'll ever need. Two desktop environments aren't necessary. Make sure you have at least one and the X window system! (if you want a GUI that is). I suggest you get all

the servers too.

Note: Knoppix uses the KDE Desktop environment

17. Make sure everything is all right, and install

18. You can create a boot disk if you want

Note: Desktop environments might have a set-up once you enter them

IV What to do now

Now that you have a Linux set-up and running, there are many paths you can head down. First, you should explore your GUI and menus. Browse the web with Mozilla, get on IM with GAIM, play games, add/delete users, check out OpenOffice, and anything else that might be part of your daily use. Also, set up a few servers on your computer to play around with, specifically SMTP (*wink*wink*), FTP (vsftp is a good one), and either telnet or SSH (OpenSSH is a good one). The setup and use of these are beyond the scope of this tutorial, but researching them could prove to be very educational.

The filesystem

The Linux (and Unix) filesystem is different from the normal Windows that you're used to. In Windows, your hard drive is denoted ?C:\? (or whatever). In Linux, it is called the root directory and is denoted ?/??. In the / directory, there are several default folders, including dev (device drivers) mnt (mount) bin (binaries) usr (Unix System Resources) home, etc, and others. I encourage you to explore around the whole file system (see section V) and research more.

Once you are well situated, it's time to get into the heart and power of Linux: the console. The next session will guide you through it and set you on the path to finding out how to do stuff for yourself. You will (probably) want to start learning to rely less and less on the GUI and figure out how to do everything through the console (try launching all your programs from the console, for example).

V. The Console

The Console might look familiar to DOS if you've ever used it. The prompt should look something like the following:

```
AvatharTri@localhost avathartri$
```

With the blinking _ following it. This can vary greatly as it is fully customizable. Let's get started with the commands.

First, let's explore the file system. The command ls will "list" the files in the current directory. Here's an example:

```
AvatharTri@localhost avathartri$ ls
```

It should then display the contents of the current directory if there are any. Almost all commands have options attached to them. For

example, using the -l option, which is short for "long" will display more information about the files listed.

```
AvatharTri@localhost avathartri$ ls -l
```

We will get into how to find out the options for commands and what they do later.

The second command to learn will be the cd command, or "change directory". To use it, you type cd followed by a space and the directory name you wish to go into. In Linux, the top directory is / (as opposed to C:\ in Windows). Let's get there by using this command:

```
AvatharTri@localhost avathartri$ cd /  
AvatharTri@localhost /$
```

Now, we are in the top directory. Use the ls command you learned earlier to see everything that's here. You should see several items, which are directories. Now, let's go into the home directory:

```
AvatharTri@localhost /$ cd home  
AvatharTri@localhost home$
```

And you can now ls and see what's around. In Linux there are some special symbol shortcuts for specific folders. You can use these symbols with cd, ls, or several other commands. The symbol ~ stands for your home folder. One period . represents the directory your currently in. Two periods .. represent the directory immediately above your own. Here's an example of the commands:

```
AvatharTri@localhost home$ cd ~  
AvatharTri@localhost avathartri$
```

This moved us to our user's personal directory.

```
AvatharTri@localhost avathartri$ cd .  
AvatharTri@localhost avathartri$ cd ..  
AvatharTri@localhost home$
```

The cd .. moved us up to the home directory. As you've probably noticed by now, the section behind the prompt changes as you change folders, although it might not always be the case as it's up to the personal configuration.

You can use these symbols with the ls command also to view what is in different folders:

```
AvatharTri@localhost home$ ls ~  
AvatharTri@localhost home$ ls ..
```

And you can view what is in a folder by specifying its path:

```
AvatharTri@localhost home$ ls /  
AvatharTri@localhost home$ ls /home
```

The last command we will cover as far as finding your way around the filesystem is the cat command. The cat command will show the contents of a file. Find a file by using the cd and ls commands and then view its contents with the cat command.

```
AvatharTri@localhost home$ cd [directory]
AvatharTri@localhost [directory]$ ls
AvatharTri@localhost [directory]$ cat [filename]
```

Where [directory] is the directory you want to view and [filename] is the name of the file you want to view. Omit the brackets. Now, if the file you viewed was a text file, you should see text, but if it wasn't, you might just see jumbled garbage, but this is ok. If the file goes by too fast and goes off the screen, don't worry, we will get to how to scroll through it later.

One of the most useful commands is the man command, which displays the "manual" for the command you want to know more about. To learn more about the ls command:

```
AvatharTri@localhost home$ man ls
```

And you will see the manual page for ls. It displays the syntax, a description, options, and other useful tidbits of information. Use the up and down arrows to scroll and press q to exit. You can view the manual pages for any command that has one (most commands do). Try this out with all the commands that you know so far:

```
AvatharTri@localhost home$ man cd
AvatharTri@localhost home$ man cat
AvatharTri@localhost home$ man man
```

One very crucial option to the man command is the -k option. This will search the descriptions of manual pages for the word you specify. You can use this to find out what command to do what you need to do. For example, let's say we want to use a text editor:

```
AvatharTri@localhost home$ man -k editor
```

And you should see a list of apps with a short description and the word "editor" in the description.

With a blank prompt, you can hit tab twice for Linux to display all the possible commands. For Linux to display all the commands beginning with a certain letter or series of letters, type those letters and hit tab twice.

Note: This is actually a function of BASH and not Linux, but BASH is the default Linux shell.

Now that you know a little about moving around the filesystem and viewing manual pages, there is one more trick that we will cover to help you out. Remember how the man pages were scrollable as in you

could use the arrow keys to scroll up and down? That is because the man pages use something called the less pager. We're not going to go into what this does exactly and how it works, but that's definitely something that you will want to look up. Here's how to use the less pager with a file:

```
AvatharTri@localhost home$ cat [filename] | less
```

That uses something called a pipe. The line is the vertical line above enter on your keyboard. Briefly, what this does is take the output from the cat command, and stick it in the less pager. By doing this, you can view files that would normally run off the screen and scroll up and down.

Some final commands to check out:

- mkdir - make directories
- cp - copy file
- mv - move file
- rm - remove file
- rmdir - remove directory
- grep - search a file for a keyword
- pwd - display current working directory
- top - display system resources usage (kill the program with control + c)