

From Windows to MEPIS

A Windows user's guide to MEPIS

Alan D Moore

<http://www.alandmoore.com>

Introduction

Welcome to the land of MEPIS and Linux! Like many new Linux users, you probably have a good bit of experience using Microsoft Windows, and you're not sure what to expect from this new operating system.

In years past, the learning curve for Linux was steep, with a user culture populated mostly by hobbyists, programmers, and self-styled “hackers”; as a result, it gained a certain reputation as being only good for servers, or requiring advanced computer knowledge just to do simple things.

Times have changed, however, and these days desktop-oriented Linux distributions¹ such as MEPIS provide a friendly environment that smooths out the transition and provides a desktop that even the most average user can work with. In the words of MEPIS creator Warren Woodford, MEPIS was designed to be “easy to try, easy to install, and easy to use”.

Even so, without some grounding in the fundamentals of Linux, especially in the areas which it differs from Windows, many newcomers quickly become frustrated and give up on Linux before they really get a chance to use it. With that in mind, I have written this “orientation” to help Windows users migrate smoothly to MEPIS by highlighting the major differences you will encounter.

The Right Frame of Mind

A different tool requires a different approach

Suppose you are an experienced carpenter. Despite your experience, though, you've only ever used a hammer and nails to fasten boards together. You're extremely skilled with a hammer and nails, so that it only takes you one or two hammer blows to drive a nail perfectly. Now suppose that someone tells you about wood-screws, and how much better they are for holding boards together. So you decide to try them out: You get some wood-screws and a power screwdriver. You set the wood-screw against the board and start whacking its head with the screwdriver in an attempt to nail it into the board. You end up breaking the screwdriver while failing to drive the screw more than a few millimeters into the board. Naturally, you conclude that wood-screws aren't as great as you were told.

Screws and nails are used to accomplish the same basic task: attaching two pieces of wood together. However, they differ greatly in design, and therefore in method of operation. If you try to operate a screw like a nail, or a screwdriver like a hammer, the results will be unsatisfactory and unfruitful. The same could be said of Linux and Windows; both are designed to accomplish similar tasks, but because they are designed differently, they each require a different method of operation.

When you try out Linux, you cannot expect to work with it the same way that you worked with Windows and get good results. Some of your skills with Windows will transfer to Linux, but many will not. If you try to operate a Linux system exactly like you operated Windows (especially if you are

¹ See appendix B for an explanation of “Linux distributions”

an advanced user), chances are your efforts will end in frustration and possibly a broken system; and if you judge Linux solely on how much it is or isn't like Windows, you are likely to be disappointed. The best approach is to put aside your assumptions, follow the recommended practices, and try to get a handle on how the system is designed to be used.

Some practical advice

Enough with the analogies, here are some practical points:

- **Try it on a decent computer:** Linux has earned a reputation for working well on older hardware, and many people have used it to give a “second life” to an older computer running an unsupported legacy OS. Even so, it isn't going to work miracles; if your hardware is defective, broken, or extremely old, MEPIS will likely perform poorly and leave you unimpressed². The MEPIS website (<http://www.mepis.org>) recommends an absolute minimum of 128 MB of RAM and 2.5 GB of hard drive space to use MEPIS with at least a Pentium II processor, though it cautions that these figures will give very poor results. It also states that MEPIS will give you about the same level of performance you'd get from Windows XP on the same hardware.
- **Don't judge performance by the live CD:** The live CD³, which allows you to run a live MEPIS session from a CD without actually installing, gives you a great way to get a glimpse of the “look and feel” of MEPIS and see the kinds of applications you can run on it. But it is not truly representative of an actual install; there are some things that won't work properly from the live CD, and more importantly the performance of the live CD desktop is *awful* compared to an actual hard drive install, especially on older machines. Bear these things in mind before passing judgment.
- **It might not work 100% on your machine:** Linux hardware compatibility has come a long way. Most hardware should give at least basic functionality in Linux and be perfectly usable. But there are still problem areas; some hardware requires extra tweaking or configuration, and some hardware doesn't work at all. We'll discuss some of the most common difficulties later on⁴.
- **Start with the basics:** It's amazing how many new Linux users start out by trying to run Windows software in Wine⁵ or compiling software from source code to get the latest versions. Start out slow, stick to the preinstalled software or software from the repositories⁶, and try to get a feel for this new OS without trying to modify the system too much. Take some time to find out what MEPIS can do well before trying to make it do things it might not do well.

2 A derivative of MEPIS, called AntiX, has been designed specifically for older hardware. More information is at <http://antix.mepis.org>.

3 See “live CD” in glossary

4 See section III

5 See “Windows applications in MEPIS”, under section II

6 See “Installing software”, under section II

Part I: Fundamental Differences

Modularity

If you want to buy a stereo system, you have two basic options: an integrated system, or a component system. An integrated system comes with everything in one chassis: CD player, tuner, speakers, amplifier, equalizer, etc. It may have additional inputs for things like an mp3 player or turntable, but essentially it's a closed system: you can't swap out the equalizer for one with more bands, you can't trade out the CD player for an MP3 CD player, etc. A component system, on the other hand, allows you to do just that: you can replace components that you don't like, or leave out components that you don't need. You can even add exotic pieces like a DAT player or reel-to-reel if you require them, as long as the right inputs and outputs are present.

Windows is, in many ways, like an integrated stereo system. It comes prepackaged with all the basic parts; while you can add software to it, you can't really mess much with the basic system. You can install Firefox or Mozilla, but you can't really remove Internet Explorer and replace it with one of them; you can't rip out Windows Explorer and replace it with the Windows 3.1 desktop. Linux, on the other hand, is like a component system. You can add, remove, or replace any piece of software on the system, from the very lowest-level operating system components to the desktop environment⁷ or core applications. This is why Linux has always been considered ideal for customized solutions or for those who like to tinker. Of course, doing so requires a bit of knowledge, and many people are content to have a system that “just works”, with or without the options.

MEPIS gives you the best of both worlds; although it still allows you the option of customizing your system to your heart's content, everything you need to have a complete operating system (and more) is preinstalled and configured. As you become a more experienced Linux user, you may find yourself removing parts of the OS you don't use, installing tools that make certain tasks easier, or replacing components to create an environment more suited to your needs or your computer's hardware. In the meantime, it's only important that you understand that MEPIS is simply one combination of components that have been packaged together for convenience; other versions of Linux, or even another user's customized version of MEPIS, may contain very different components⁸.

The command line and the GUI

To most Windows users, the command line interface (CLI) is a relic of the past that died with DOS; the mere suggestion of interacting with a computer via typed commands tends to inspire everything from revulsion to outrage in many new Linux users. Yet the command line interface remains alive and well with many Linux users. It is a powerful tool with decades of development and user experience behind it, augmented by a wide selection of command line applications and utilities that allow you to do everything from system updates to CD burning. Nearly anything you can do through a graphical user interface (GUI) can be done with greater speed and flexibility through the command line, and eventually even “average users” find ways to exploit its power.

If you remain unconvinced, however, don't despair; MEPIS is made for you. MEPIS has a complete graphical user interface with icons, buttons, menus, windows, and everything else you expect from a modern computer system. You can install, administrate, and use MEPIS without ever touching a command line. The power of the command line is still there if you wish to use it, but it is completely optional.

Just bear in mind, as you interact with more experienced Linux users (especially if you are asking for

⁷ See “desktop environment” in glossary

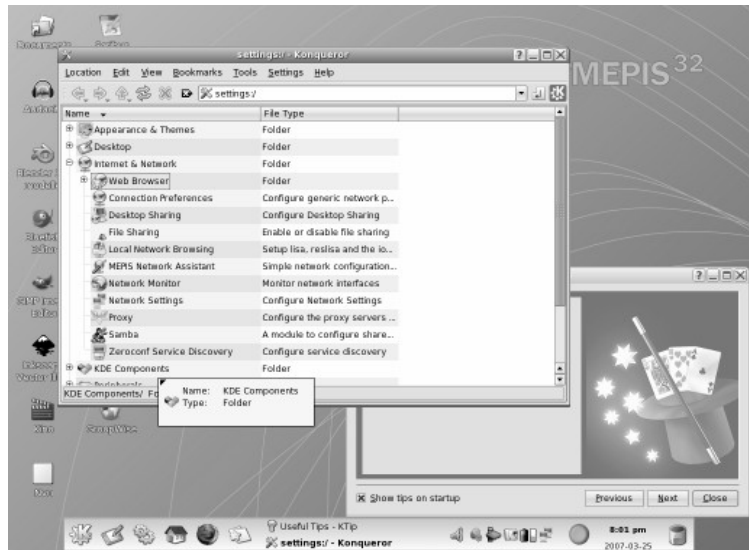
⁸ A good example of the modularity of Linux can be seen by comparing MEPIS with its derivative, AntiX.

technical help), you are likely to see a lot of command line usage going on. You may even be expected to run some commands yourself, if someone is helping you troubleshoot a problem. This is not a reflection on MEPIS's capacity to be administrated graphically, but simply a deeply entrenched part of Linux user culture (which, despite criticism, is not going to change anytime soon).

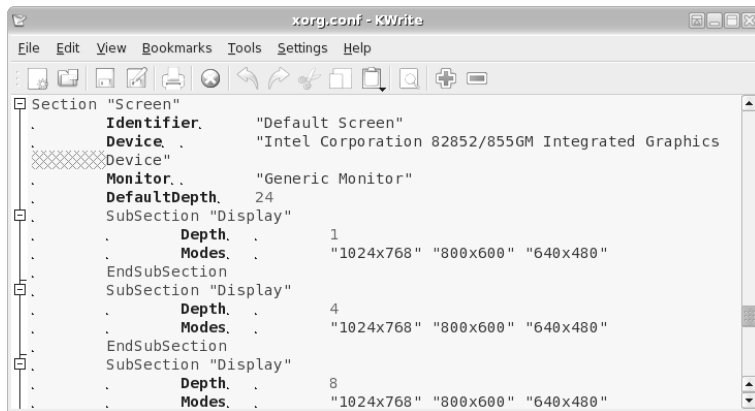
Configuration files

When you change a setting on Windows, such as the color of your desktop or the network address of your computer, Windows stores that data in the “registry”. The Windows registry is a database that holds almost every bit of configuration data on your Windows operating system. It is made up of binary⁹ files that can only be read from or written to by special software such as your control panel applets or regedit.exe.

Linux does not have a “registry”. Instead, nearly all the configuration information on your system is stored in discrete, human-readable text files¹⁰. These files are usually referred to as “config files” or “conf files”. In the “old days”, administering a Linux system was largely a matter of editing text files; even now, the slick graphical tools we have for administering MEPIS are doing little more than making changes to text files which knowledgeable users could just as well edit directly¹¹. As a new user you probably won't be hacking into any configuration files just yet, but it's important to keep this difference in mind as you learn more about Linux.



MEPIS sports a complete graphical environment -- no command line required!



Part of the config file that controls video display

Case Sensitivity

Like all Linux distributions, MEPIS is case-sensitive. This means that “MEPIS.txt” and “Mepis.TXT” are not the same file, and “FireFox” is not the same command as “firefox”. For “point-and-click” users, this generally isn't an issue, but it's important to be aware of, especially if you use the command-line at all.

⁹ See “binary(file)” in Glossary

¹⁰ If you are old enough to have used DOS or Windows 3.1, you may remember “.ini” files which were used to configure those systems. Most Linux configuration files use a similar syntax and structure.

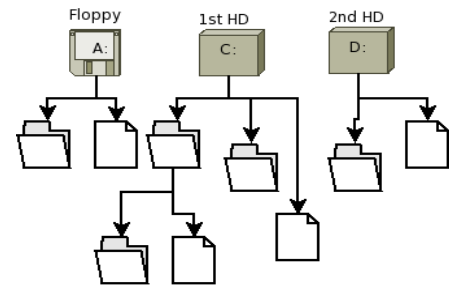
¹¹ It's worth noting that typical Linux configuration files are *meant* to be hand-edited, and as such they are typically both easily readable, and heavily documented internally (with comments) and externally (in a man page or handbook). Contrast this to the Windows registry, which is largely cryptic nonsense understood only by the system's specialized configuration tools.

File system

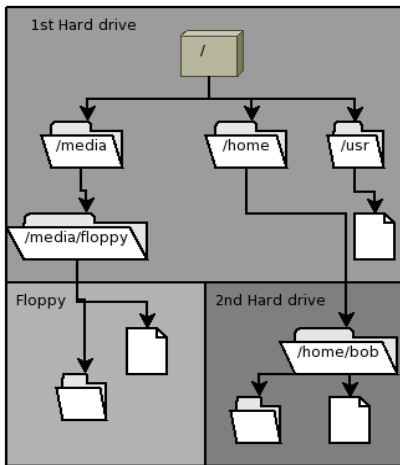
Basic file system structure

One of the first problems I struggled with as a Linux user was how the file system¹² worked. If you have been looking around your MEPIS system trying to find the “C:\” drive or “A:\” drive, you are searching in vain. Linux handles storage media very differently.

On your Windows computer, each drive or partition receives a letter, like “C”. The top-level directory on C is C:\ -- this is called the “root” of C. Under that root directory, there are various subdirectories and subdirectories of subdirectories branching out into a tree structure we call the “file system” of C. Each device contains its own file system.



Windows File System



Linux File System

On Linux, there is only one file system, and its root is labeled simply “/”. When a storage device is added to the system, it is attached to a directory or subdirectory of the file system; this is called “mounting” a drive or device, just as you might mount a trophy or picture to your wall. When you mount a disk to a directory on the system, the contents of the disk appear under that directory, and files or directories you create under that directory will be on the new disk. By default, new devices are usually mounted under the /mnt or /media directories (for instance a floppy drive might be mounted to /media/floppy), but in reality they can be mounted anywhere in the file system¹³.

The advantages of doing things this way may not be immediately apparent, but think about this situation: you have a relatively small hard drive in your system, and you find that it is filling up with pictures and other files. You go out to the store and buy a second hard drive and install it in your system. In Windows, the disk might show up as D:\ or E:\ (whatever the next free letter is, usually), and you’re going to have to copy your files over to it and remember to put new files on the new drive from now on. In Linux, you would install the disk, copy your home directory to it, then mount the disk as your home folder. After that, you would carry on just as before, without having to think about the fact that a second hard drive is involved. It’s just that suddenly your old home directory will have a lot more room!

Important folders

MEPIS has several directories right under the root of the file system. These directories are based on the Unix¹⁴ filesystem hierarchy standard, which you can read more about at <http://www.pathname.com/fhs/>. As a new user, there are only a few of these directories you need to be concerned with.

By far the most important to you is the /home directory. Under /home, each user on the system has a

12 See “file system” in glossary

13 It’s worth noting that network file shares (such as shared folders on a file server or NAS) are mounted in the same way as physical devices.

14 See “Unix” in glossary

folder, called the user's "home folder". By default this folder is the only folder on the system in which you can save or edit files. Everything about your user account, from your desktop background to your email messages, is saved somewhere in your home folder. This may seem restricting at first, as Windows users are used to being able to put files anywhere on the system they want to. However, keeping all your files, folders and settings in one directory makes it very easy to back up all your data, and protects you from accidentally overwriting or deleting system files.

The next folders you want to know, which I've already mentioned, are /mnt or /media. Removable devices, such as floppy drives, CD's and USB drives, show up under /media when you attach them to the system. Additional hard drive partitions (such as your Windows partitions, if you are dual-booting) show up under /mnt by default.

Finally, the /etc folder is where all the global configuration files are stored. You can think of it as the Linux equivalent of the Windows registry¹⁵. If a program has global or default settings that can be adjusted, you will find the appropriate text configuration file for it under /etc. Also, system settings such as your network configuration, startup scripts, and file system configuration are stored in this directory.

Users and security

Accounts and permissions

Linux, much like Windows NT, 2000, XP, and Vista, is an account-based operating system. This means that every program that runs on the system does so under the credentials of a specific user account. If a user "Bob" logs in and launches a web browser, the web browser is running on Bob's account. If user "Mary" logs in and launches the same program, it is instead running with Mary's credentials. No program can run without an account to run under, and no program running under a given account can do more to the system than what that account has been allowed. If Mary's user account does not have permission to save data to one of Bob's folders, no program Mary runs will be able to do so either.

To simplify the management of multiple user accounts, account-based systems also implement "groups". A group is simply a collection of permissions and restrictions that can be applied to a user; for instance, on MEPIS being a member of the "cdrom" group allows you to use local CD ROM drives, while membership in the "scanner" group allows you to use scanning devices¹⁶.

As mentioned before, newer versions of Windows operate on this principle as well (3.1, 95, 98, and ME did not); in theory Linux and Windows operate mostly the same in this respect. However, in practice, there is a big difference. Most people set up their Windows user account as a member of the "administrators" group, meaning that their regular user account has unlimited privileges on the system (in fact, the user account you create in the setup wizard for Windows XP defaults to an administrator account – with no password!).

On Linux, there is only one administrative account. It is called "root", and it should only be used for doing system administration and special low-level tasks. **Do not operate your system as root for normal tasks!** Doing so undermines the whole security model of Linux.

¹⁵ See "Configuration files" under section I

¹⁶ Naturally, a user account can be a member of multiple groups, allowing you to have multiple system permissions. Users in MEPIS are, by default, added to a variety of groups.

Because we do need to run programs with root privileges, though (for tasks like installing software or editing global configurations, for example), Linux provides us with ways of running individual programs with root credentials without having to be fully logged in as root. The most common include:

- **kdesu:** This command is for running graphical programs as root in KDE¹⁷. To run a program with root credentials, you would type “kdesu someProgram”, after which you'll be prompted for root password¹⁸.
- **Administrator Mode:** In the system configuration tools, you may notice a button at the bottom of the page on some configuration screens that says “Administrator Mode”. Typically, if a setting involves global changes to the system, such as adding a printer or setting the network settings, you need to have root privileges to do this. Just press the “Administrator Mode” button and enter your root password.
- **su:** This command is used on the command line to switch to another user account within a terminal session. You can specify the user account, or specify no user account to become root.
- **sudo:** This is a highly-customizable command that allows only specific programs to be run by specific users with root privileges. MEPIS does not use sudo for anything by default, but it can be configured if you need it.



KDE asking for your root password

When running a program as root, keep in mind that any user-specific configurations that you do will be configured for the root user. If you later run the program as a normal user, you will no longer have your specific configurations. For example, if you were to run the KDE settings utility as root and changed the desktop settings, the changes would affect root's desktop, not yours.

It's tempting for many new Linux users who are used to Windows to want to try to disable as many of these security measures as they can, and make their user as close to “administrator” as Linux will allow, because this is common practice on Windows. There is really no clean way to do this, and in any case this sort of thing should be avoided as much as possible. There is rarely any good reason to have administrative privileges for regular use in a properly-designed system (which we like to think Linux is).

Anti-virus, firewalls, and other security software

For many people, the switch to Linux is motivated by problems they've had with viruses or malware in Windows. If this is part of your reason, you are probably interested in getting some anti-virus, anti-malware, and anti-intrusion software going on your MEPIS system. Fortunately for you, MEPIS ships with an excellent anti-virus (clamav) and firewall (guarddog) already installed.

There are veritable gigabytes of debate on the web about what the future holds for Linux viruses, and whether the increasing popularity of the platform will prompt more viruses to be written for it. What is beyond debate, however, is the current state of affairs on viruses and malware for Linux: there are virtually none to concern yourself about. There are a handful of known Linux viruses, but the majority of them are no more than proof-of-concept programs that never left the lab. The remainder largely

¹⁷ The GNOME desktop environment has a similar program, gksu, which can also be used under KDE.

¹⁸ The menu or desktop shortcuts for programs that typically require root privileges to use are already configured to run with sudo, kdesu, or gksu as appropriate in MEPIS. Thus, as a new user you probably won't have to worry about these commands much.

depend on exploits limited to certain programs or distributions, most of which have now been patched. And as for malware (spyware, adware, etc), it is unheard of to date.

Commercial software companies have made a fortune selling security as a product; but there is an old adage that every security professional knows, “security is not a product, it's a process”¹⁹. In other words, keeping your computer secure is not a question of what anti-virus or anti-malware products you buy, or even whether you use Linux or Windows; it's a question of using your computer securely. Where Linux excels over Windows is in providing us with tools and environments more conducive to good security practices – tools that can be scaled to the knowledge and experience of the user.

What are these practices? As mentioned above, user security is vital; making limited use of the root account and password goes a long way toward keeping your computer secure. But here are a few more:

- Use your firewall to block incoming *and* outgoing programs that you don't use.
- Remove or disable network-aware programs that you don't use.
- Download software only from legitimate, trusted sources (the repositories are a good start).
- Keep your computer behind a router on a private subnet, even if it's the only computer in the house.
- Don't follow links in spam emails.
- If someone using your computer doesn't follow good practices, limit their administrative access.
- Update regularly. Nearly all updates you will get using the stable version of MEPIS will be security patches.

If some of these terms don't make sense to you and you need more help with staying secure, see the “getting help” section below. Good security on any operating system requires a certain amount of technical knowledge, but fortunately MEPIS is fairly secure by default.

19 This quote is believed to be attributable to security guru Bruce Schneier, <http://www.schneier.com>.

Part II: Software

Using software

Applications in MEPIS

Depending on what software you used in Windows, you may or may not see any familiar applications the first time you log in to MEPIS. Although few well-known commercial applications are published for Linux, a rich selection of free alternatives are available for easy installation (see the next section). MEPIS even comes with many commonly used applications preinstalled and configured.

The following chart shows some common Windows software and the closest preinstalled MEPIS equivalents. For a more extensive list of equivalents, check the MEPIS documentation wiki (<http://www.mepis.org/docs>) or <http://www.osalt.com>.

<i>Windows Application</i>	<i>function</i>	<i>MEPIS equivalent(s)</i>
Internet Explorer	Web Browsing	Firefox, Konqueror
Windows Explorer	File Browsing	Konqueror
Outlook Express	Email client	Thunderbird, Kmail
Windows Media Player	Audio / Video player	Kaffeine, Amarok, KMplayer
Microsoft Office	Office Suite	OpenOffice.org
MSN messenger, YIM, AIM	Instant messaging	Pidgin, Kopete
Dreamweaver, Front Page	Web page design	Quanta Plus, Kompozer
Outlook	Personal Information Management	Kontakt
Bittorrent	Torrent client	Ktorrent, Azureus
Notepad	Raw text editor	Kwrite, Kate

Keep in mind that these programs are not “drop-in” replacements for their Windows counterparts; they are “equivalent alternatives”, so expect to do a little relearning and readjusting of your work flow (remember that wood-screw analogy?).

Windows Applications in MEPIS

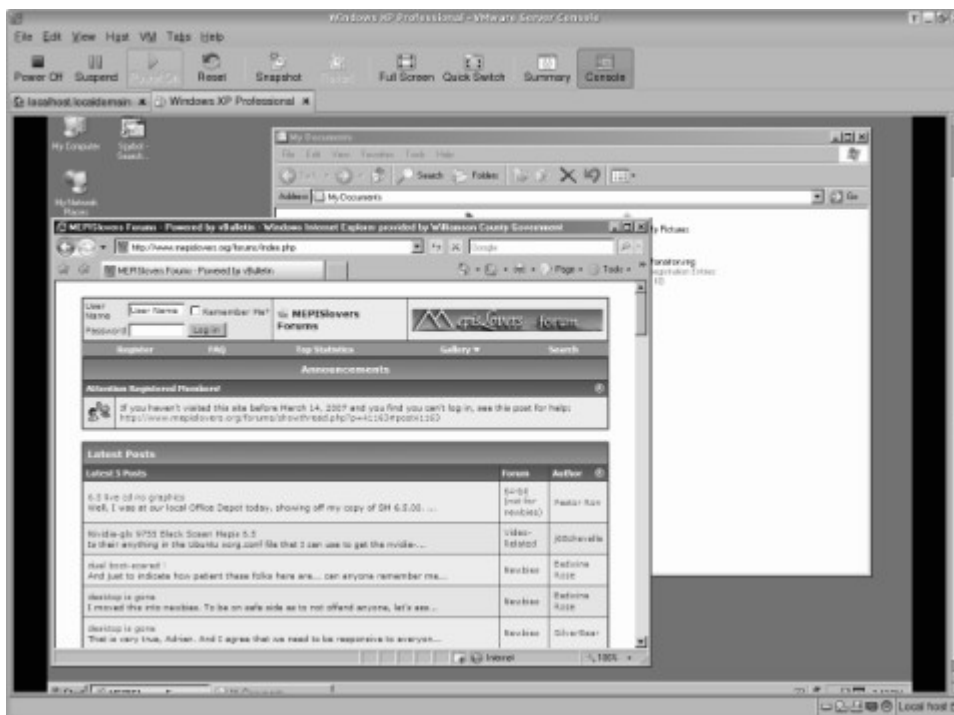
Despite the wide array of available free software for Linux, many new and experienced users find that there are Windows programs they simply can't replace with Linux-compatible alternatives. There are a few different approaches to dealing with this situation:

- **Dual Boot:** You can set up Linux and Windows on the same computer, selecting the OS you want to use when you start the computer. The downside of this is that you have to reboot every time you want to go back and forth between your Windows software and your Linux software.
- **Wine and its derivatives:** “Wine” is a program that provides Windows-compatible software with a fake Windows environment to run in. Wine is not very easy to set up, and the results are often disappointing (if you even get results). Still, it typically works great for smaller, older applications that don't need to deal directly with your hardware, or for well-known applications that have been out for a while (such as Microsoft Office 2000). Wine is free, but there are several commercial derivatives that add user-friendly features and extended compatibility with

certain types of Windows Software. Cedega (which is designed for games) and Crossover Office (designed for productivity applications) are the two leading products.

- **Virtual Computing:** One of the newest solutions to the Windows software dilemma is to simply run Windows on a “virtual computer” inside your Linux install. It requires a relatively powerful computer with lots of memory, but when set up properly you can run Windows in a window on your desktop and have instant access to your old software. The most popular products for accomplishing this are VMware (free), qemu (free and open-source), VirtualBox²⁰ (free and mostly open-source) and Parallels (commercial).

The biggest drawback to this approach is performance. Not only are you splitting your hardware resources between two operating systems, but the “guest” system does not have access to your physical hardware (it runs on emulated hardware). Thus, if you need Windows for resource-intensive applications such as 3D gaming or multimedia authoring, dual-booting may be a better option.



Windows XP running in vmware. Think of it as "Windows in a window".

Installing software

When you install software from the Internet on Windows, you typically go to some download website, download the software, unzip it, run some kind of installer program, go through a wizard, click a bunch of “next” or “yes” buttons, then your software is installed. Not so bad, especially if you only have one or two programs downloaded from the web.

But on Linux the vast majority of your software is downloaded from the web, even things like your desktop environment²¹ and OS kernel²². So in order to keep track of it all, MEPIS uses a system called

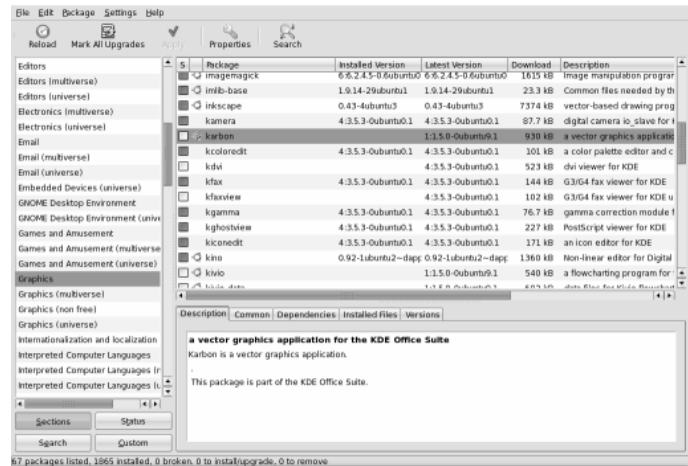
²⁰ MEPIS 7 includes the VirtualBox repository by default, but you need to activate it. After that, VirtualBox is installable via Synaptic.

²¹ See “desktop environment” in glossary

²² See “kernel” in glossary

Advanced Package Tool (APT). With the APT system, you don't have to go to websites or download packages by hand; you use a single program, **Synaptic**, to find, select, install, and upgrade all the software on your system. The interface is very easy and consistent no matter what software package you install. In addition, Synaptic does a kind of “background check” on every package you install with it, making sure you have all the required support software to run a given program, checking the packages to make sure they haven't been tampered with or corrupted, and ensuring that you are getting the latest versions available for your OS.

Installing software with Synaptic is a simple point-and-click procedure. When you launch Synaptic, you are presented with a massive list of software packages that you can install. All of the software packages listed are located on special Web servers called “repositories”; you can download the latest list by clicking the “reload button”. Once you've done this, you can locate software you want to install using the search feature, or by browsing through category sections like “games” or “graphics”. When you find a package you want, simply select it and hit “apply”. Synaptic automatically downloads the package and any necessary support packages and installs them. For more information about APT and repositories, see this excellent article on Wikipedia:



Installing software with Synaptic

http://en.wikipedia.org/wiki/Advanced_Packaging_Tool.

Of course, not every software title you might like to run is in an online APT repository; or you might want a newer version than the one available through Synaptic. There are several other methods of installing software on MEPIS which vary in difficulty; I'll highlight them here briefly, but in the interest of keeping things short, we won't go in to them in any detail.

- **Deb packages:** The software packages installed through APT are in a format called “deb” (short for “Debian”, the Linux distribution that came up with APT). You can manually install deb packages using the command line tool “dpkg”, or the graphical tool “Kpackage”. Keep in mind that installing deb packages this way does not automatically take care of dependencies like APT does. You may need to manually install the dependencies of the package, if it has any.
- **RPM packages:** Some distributions of Linux use the RPM packaging system (short for RedHat Package Manager, developed by RedHat Linux, of course). RPM packages are similar to deb packages in many ways, and there is a command line program available on MEPIS to convert RPM packages to debs called “alien”.
- **Source code:** Any open-source program can be compiled from source if there is no other option. This isn't as hard as it sounds most of the time, but sometimes you can run into errors that require a little more skill to sort through. Source is usually distributed as a “tarball”²³, which is a compressed file (like a zip file) usually ending in .tar.gz, .tar.bz2, or .tgz.
- **Miscellaneous:** Many software developers package software in their own custom ways, usually

23 See “tarball” in glossary

distributed as tarballs or zip files. They may contain setup scripts²⁴, ready-to-run binaries²⁵, or binary installer programs similar to Windows “setup.exe” programs (these are often suffixed with “.bin” or “.run” in Linux).

Upgrades

As I mentioned above, Synaptic can also be used to upgrade your system. This works similarly to the “Windows Update” system you used in Windows, except that it not only updates your OS but all the software on your system (at least, all the software you installed through Synaptic).

To upgrade, open Synaptic, click “Reload” to get the latest package information from the server, then click “Mark all upgrades”. You can also click the “sections” button at the bottom of the window and check out the software in the “Installed (upgradeable)” section. Sometimes, Synaptic will not automatically mark an upgrade if the upgrade requires additional operations (which you might not want it to do), such as installing or removing other packages. You can manually mark these for upgrade to see what additional changes are required.

Because the repositories which MEPIS uses by default receive only security updates and minor bug-fixes, upgrades are pretty safe and shouldn't cause problems on your system. However, as a general rule, you should be cautious of upgrades that involve the following:

- Kernel or driver upgrades (packages like “linux-image-2.6.x.x.”, or “modules-2.6.x.x”)²⁶
- Upgrades to the GUI engine (anything that says “xorg” in the name)
- Upgrades to KDE (The MEPIS desktop environment -- usually says “kdebase” or “kdelibs”)
- Upgrades that involve significant changes in version number. Usually, version numbers might look a bit like this: 6.2.01-2. Although programmers don't have a 100% consistent approach to these numbers, a change in one of the first couple of digits (e.g. moving 6.2.01-2 to 6.5.0 or 7.0) generally indicates some kind of major change in the program. On the other hand, a change to something like 6.2.01-3 is probably just a minor patch.

Before doing such upgrades, you may want to consult the MEPIS forums to see if anyone has had problems with them. Generally speaking, though, unless you are running a beta version of MEPIS, or have changed from the default repositories, you shouldn't see any “dangerous” upgrades.

24 See “script” in glossary

25 Also “binary”: see “binary” in glossary

26 Generally, the MEPIS kernel is NOT upgraded except between major version changes (e.g., 6.x to 7.x). If you are seeing a kernel upgrade available, chances are it is not a kernel from MEPIS. Check with the community forum, or look for an announcement at mepis.org.

Part III: Practical Considerations

Problems to expect

By far the hardest and most problematic part of switching to Linux is getting it installed and set up; granted, MEPIS has probably the easiest and friendliest installation process in the history of operating systems, but there are still some pitfalls to watch out for. If you have a computer made up of 100% Linux-compatible parts and you plan to devote 100% of the hard drive to Linux, you should have no problems. But most Linux installations are complicated by two factors: 1). installing on hardware that is not completely Linux-compatible, and 2). wanting to dual-boot with Windows.

Hardware issues

The four big hardware trouble spots are wireless networking, dialup modems, laptop power management, and 3d acceleration for video.

- **Wi-fi:** Support for wireless network cards has improved dramatically in recent years, but because the technology itself is in a rapid state of development, Linux developers have not always been able to keep up or get the information they need to write drivers. Fortunately, if your card doesn't have a Linux driver, MEPIS has a utility called “ndiswrapper” which can translate the Windows drivers for use in Linux. In addition to driver issues, support for some forms of wireless authentication (such as certain types of WPA or EAP) is experimental and unstable.
- **Modems:** Many cheap, on-board or PCI modems are of a variety known as “winmodems”; these are not true hardware modems, as they rely on software to do a lot of the work for them. Unfortunately, the software involved is in some cases proprietary and only works on Windows. MEPIS actually has pretty good support for a number of these “winmodems”, but they can still be finicky and not all features are supported. If you have trouble with your on-board modem, you might get counseled to get a “real” modem, which usually means a stand-alone serial modem. All standalone serial modems communicate using a standardized command set that works with any OS platform; if you must use dial-up, they are highly recommended.
- **Laptop power management:** Due to lack of standardization in laptop hardware, MEPIS may require some additional tweaking to get the “hibernate” (a.k.a. suspend-to-disk) and “suspend” (a.k.a. suspend-to-RAM) features working correctly on your laptop. You can usually find help for your particular laptop model at <http://www.linux-on-laptops.com>.
- **Video:** If you have an Intel or newer S3 graphics card, MEPIS usually provides 3d acceleration “out-of-the-box”²⁷. If you have an Nvidia or ATI graphics card, MEPIS includes a special tool to install the proprietary drivers for these cards. If you have some other type of card, you should be able to get good video for basic operations, but 3d acceleration is probably not going to happen.

Keep in mind, Linux hardware support is in a rapid state of improvement. If you are not running the latest version of MEPIS and having hardware troubles, it is highly recommended that you try a newer version.

²⁷ Some Intel chipset versions, notably the 845 and 965, have known issues with the version of the GUI engine (xorg) included in MEPIS. Check with the community for workarounds if you have one of these.

Dual booting issues

Generally speaking, installing an operating system on a computer is simply not an easy task; setting up two operating systems on your computer is even less so. It involves knowing a bit about hard drives, file system types, master boot records, and other technical concepts. Setting up your Windows computer to dual-boot with Linux typically involves doing some things to Windows that Windows was not designed to sustain. For instance, you will probably have to resize your Windows disk partition, since in most cases it fills the entire hard drive; or, you might have to overwrite the master boot record (MBR), which is the part of your hard drive that tells the computer where to find the operating system.

Fortunately, Linux developers have put a lot of work into simplifying and stabilizing the process; MEPIS boasts one of the most user-friendly installation wizards around. What's more, you can find some excellent guides to help you install Linux in a variety of scenarios at the "MEPIS guides" website (<http://www.mepisguides.com>). Whatever you do, you're strongly recommended to backup all your important data and programs before installing MEPIS, as the installation process could potentially destroy your Windows installation²⁸.

Getting help

Built-in documentation

Nearly every command, program, or configuration file in MEPIS has some form of built-in documentation. Just as in Windows, most graphical programs have a "help" menu with a link to the program's user manual; this can also be called up by hitting the "F1" key (just as in Windows).

There are a few other methods of getting help, however:

- **KDE handbooks:** The help files for KDE programs are referred to as "handbooks"; they are usually available from the help menu in the programs, but can also be accessed directly through the KDE help center (the "help" option on the K menu)²⁹.
- **man pages:** man (short for manual) pages are the most venerable form of help documentation, dating back to the early days of Unix³⁰. They are generally used for documenting command line programs, configuration files, and the command line options of graphical programs. Man pages can be read at the command line using the command "man programName", or in Konqueror by entering the address "man:/programName".
- **info pages:** info is very similar to man, but is slightly newer and has more features than man. Even so, it is not as commonly used, and generally if a command has an info page the same text will be copied into a man page. You can view info pages at the command line with "info programName", or in Konqueror by entering the address "info:/programName".



KDE help center

28 As this statement might alarm some people, it's worth noting that the only truly "dangerous" parts of the install are in resizing the Windows partition and installing the bootloader (GRUB). This danger *can* be eliminated by adding a second hard drive and doing some special configuration of the Windows bootloader, if you are truly worried about destroying Windows.

29 Occasionally, due to space reasons, programs in MEPIS get installed without their documentation. If you find that a program does not have a "handbook" or help file, try searching in Synaptic for the program, and look for a package of the same name but with "-doc" appended. This package usually contains the help documentation for the package of the same name.

30 See "Unix" in glossary

On the Internet

As with most free Linux distributions, the main source of technical support for MEPIS is community-based and found on the web. The official MEPIS website can be found at <http://www.mepis.org>; MepisLovers (<http://www.mepislovers.org>) is the community support forum, and LinuxQuestions (<http://www.linuxquestions.org>) has a forum specifically for MEPIS. The MEPIS documentation wiki, <http://www.mepis.org/docs>, is a large “online manual” created by volunteers from the MEPIS community; it has an excellent “Getting Started” section to help newcomers get things up-and-running.

When searching for help on the Internet, keep in mind that MEPIS 6.x is based on Ubuntu Dapper Drake (6.06), while MEPIS 7.x is based on Debian 4.0 (“Etch”) and many times what is true for Dapper or Etch, respectively, is true for MEPIS as well. This can be helpful if you're looking for information about hardware or software compatibility, for example. In a broader sense, Ubuntu itself is derived from Debian, so often more general advice about Debian can apply to any version of MEPIS. Be warned, though: given the rapid pace of development in Linux and MEPIS, and some major changes that took place in MEPIS between the 3.x, 6.x, and 7.x releases, much of the information on the Internet is outdated or misleading, especially to newcomers. When in doubt, ask on the forums, where you can get the latest information.

When it comes to asking for help on the forums, you should remember a few important guidelines about community help:

- **Search first:** Mepislovers has a good search engine; chances are if you have a common problem, someone has already posted an answer. In addition, the search feature at Mepis.org searches the documentation wiki, so a great deal of MEPIS information is readily available; this should be your first stop for finding answers. Of course, Google can turn up all kinds of information; Google even has a special search engine for Linux-related topics at <http://www.google.com/linux>.
- **Be polite, friendly, and positive:** Nobody on the forums is paid or obligated to support you; it's a community. Demanding service from a community forum is like complaining about the food at a family potluck. Many people like to help newcomers for the pure joy of being helpful and making use of their knowledge. Be respectful and appreciative of this fact. Also, be aware that the MEPIS community, unlike some other Linux communities, has historically been more of a *user* community than a *developer* community.
- **Be teachable and self-sufficient:** Everyone on the community site started out just like you; over time, they asked questions, read responses, broke things, fixed things, and just generally got their hands dirty digging-in to what Linux is all about. If you post a problem, you will probably be asked to run some commands, read some documentation, or check out a post at another forum. If you aren't willing to learn and do some legwork -- if your expectation is that someone will wave a magic wand and make your problem go away -- community support probably isn't for you; you might want to check out a commercially supported Linux, or find someone in your area (through Craigslist or similar sites) who is willing to do paid support for Linux.
- **Report back on your results:** If you end up finding the solution to an issue, or if the community manages to get you fixed, make sure you post back saying how the issue was resolved, and mark your post as “solved”. This helps other people who may be having a similar problem and are searching the forums for ideas.

What Next?

Hopefully you have taken in more than enough information to get you started using MEPIS Linux. If you haven't done so already, here are a few recommended steps for getting started with MEPIS:

- **Install it!** Download the latest MEPIS live CD image from <http://www.mepis.org>, and install it on your computer.
- **Read the manual:** For the release of MEPIS 7.0, the MEPIS user community got together and produced an excellent, practical guide for installing, configuring, and using MEPIS. You can find a link to it on your MEPIS 7.0 desktop, or point Konqueror to `/usr/share/mepis-manual/en/index.html`.
- **Check out the wiki:** Check out the “Getting Started” section on the MEPIS documentation wiki, <http://www.mepis.org/docs>. Instructions on setting up many common (and some uncommon) devices and services can be found there, as well as general usage information.
- **Join the community:** Pop on over to <http://www.mepislovers.org> and join the MEPIS community. You'll find lots of helpful people ready to answer your questions and help you get productive with your new OS.
- **Contribute:** MEPIS is free! But if you decide MEPIS is the OS for you, why not contribute by purchasing items from the MEPIS store at <http://www.mepis.org/store>. Doing so keeps MEPIS development going and helps this great distribution to improve.
- **Have fun!** Tinker, experiment, and explore! MEPIS is loaded with great software and features, just waiting to be used. See what you can discover!

Thanks for reading, and welcome to MEPIS!

Appendix A: Why open source matters to non-geeks

No introduction to Linux would be complete without a discussion of open-source software and why open-source, free-as-in-speech³¹ software is so important. Many new users discount the open nature of Linux and the programs that run on it, maintaining that there is no practical difference between proprietary freeware and open-source software. In other words, the fact that it costs nothing is all that matters. There are several differences, however, that affect ordinary users in major ways.

Granted, most of us (even those with some programming skills) aren't likely to crack open the source code of our office suite or desktop environment and start hacking around. But consider these advantages that are unique to open-source:

- **Longevity:** An open source application will live on as long as anyone takes interest in it. Even if the company or programmer that started it stops developing it for some reason, anyone else can pick up the project and keep it going.
- **Software meets the needs of all users:** As applications develop over time, they sometimes go through major evolutionary changes. Sometimes these changes don't rest well with all users, who prefer the old versions. When this happens to an open-source program, the source code can be forked³² – thus, the old version continues to be developed along a more conservative vector while the new version can press on more progressively.
- **Software works on more platforms:** The source code of an application can be compiled to run on a variety of operating system platforms. An open-source application can be ported to any platform that users desire, and an open-source operating system can be ported to run on different hardware platforms (the Linux kernel, for instance, can be run not only on “PC” and Apple hardware, but on everything from handheld devices to giant mainframes). This means that you never have to worry about your operating system not working with the software you like to use. You don't have to buy a new office suite because you upgraded your OS, and you can run your new OS on your old computer.
- **The software will always be free:** A piece of proprietary freeware can go commercial any time the copyright holder wants it to. If the rights to the software get bought by a company, or if the author just decides to start charging, there is nothing to stop it. Now, there's nothing wrong with someone charging you for their work, but it's nice to know that your free software will always be free, and nobody can change that.
- **Flexibility and modularity:** Open source tends to go hand-in-hand with open standards. Interoperability and cooperation between developers is a goal of open source software, and open standards provide a way for developers around the world to create compatible software without having to work in tight collaboration. For you as a user, this means having options; when systems are based on open standards, your files will open up in your choice of programs, your network services will work with your choice of client software, and your hardware will “just work” on any platform without having to install special driver software. This scenario is contrary to the goals of many proprietary developers, however, who would rather snare you into using only their software or hardware.
- **Trust:** While you may not be scouring the source code of your open-source applications looking for malicious content, rest assured that many others are. It would be very difficult for a malicious software author to sneak some spyware or adware into a piece of open-source software, and it would not go undiscovered for long. Of course, the proprietary freeware world is chock full of malicious software of all types. Games, media players, utilities and screensavers have all served as vehicles for malware. Which kitchen would you rather eat food from: one kept behind locked doors and tinted glass, or one that was out in the open with nothing to hide?

31 See “free as in speech” in glossary

32 See “fork” in glossary

- **Security:** While it may seem counter-intuitive to some, open-source development methods generally produce more secure applications. Source code obscurity is often a crutch to make up for inherent weaknesses in the software's design. After all, if you are assuming that no one will ever look at your source, you may find it tempting to leave a few things less than secure. On the other hand, if you know that your source code will be visible to the world, a watertight design is absolutely necessary.

A second reason why OSS tends to be more secure is that many programmers can look at the code for flaws and offer fixes. With a closed-source program, only programmers who are working on the code have the ability to make fixes or find flaws in the programming. Suppose there are two teams of architects working on plans for a new building. One team only allows the team members access to the plans. The other team posts its plans on publicly and request comment from their peers. Who is likely to have a sturdier, safer building in the end?

People who say it better than me...

If you would like to read more about the benefits or ethical considerations of free, open-source software, take a look at these links:

Richard Stallman / The Free Software Foundation

“Why Software should be free”, <http://www.gnu.org/philosophy/shouldbefree.html>

Eric S Raymond

“The Cathedral and the Bazaar”, <http://www.catb.org/~esr/writings/cathedral-bazaar/cathedral-bazaar/>

“The Magic Cauldron”, <http://www.catb.org/~esr/writings/cathedral-bazaar/magic-cauldron/>

Bruce Perens

“The emerging economics of open source”, <http://perens.com/Articles/Economic.html>

“The open source definition”, <http://www.oreilly.com/catalog/opensources/book/perens.html>

Appendix B: Distributions

Throughout this article you may have noticed the phrase “distribution of Linux”. This can be a confusing term and concept for those who are not familiar with open-source software, particularly as Linux is often compared directly with products like OSX and Microsoft Windows.

Technically speaking, the only part of the Linux operating system that is truly “Linux” is the kernel. The kernel is the “engine” of the operating system; it's a small layer of software that works directly with your hardware to perform all the basic operations of your system, such as reading or writing to the hard drive, moving data to and from RAM, or shutting down all your software when you turn off the computer.

Of course, a kernel isn't much use by itself, any more than a car engine is useful without wheels, transmission, or chassis. So to create a usable system, the kernel must be bundled together with other software packages (such as a command-line shell, graphical interface, hardware detection system, networking subsystem, etc) to create a complete operating system. Any such bundle is referred to as a “distribution”, or just “distro” for short. MEPIS is one such distribution.

Because the Linux kernel and the software typically bundled with it are free and open-source, any company, organization, or individual is free to create a distribution and distribute it. They are even free to take other existing distributions, modify them, and redistribute the result as a new distribution. The large number of distributions that have resulted from this is often a source of criticism from both computer novices and industry veterans; indeed, the choices can be overwhelming when trying to settle on a distribution to use. But keep in mind that this variety is good for Linux and Linux users in many ways:

- It allows Linux to develop in an evolutionary manner, where the best ideas and approaches garner the most users and interest.
- It allows developers to re-use other good ideas while inventing their own. For instance, by using the Debian distribution as a base, MEPIS developer Warren Woodford only had to add the parts that he felt needed improvement in order to create a whole new distribution.
- It allows distributions to specialize. Unlike Windows, which tries to be everything from a gaming system to a secure server platform in one OS, Linux distributions can be tailored to a specific niche. For instance, MEPIS and PCLinuxOS are tailored towards home desktop users; CentOS and RedHat are geared toward secure servers; Knoppix and Morphix are designed to be useful live CD's; and Musix and Dyrn:bolic were designed for multimedia production.

If you want to learn more about the types of Linux distributions available and what sort of features they offer, a good place to start is <http://www.distrowatch.com>.

Appendix C: Glossary

binary (file): A file that is computer-readable; usually referring to software that has been compiled, or to files that cannot be viewed in a text editor. This is often contrasted to “text files” such as ASCII or Unicode files that are readable in any standard text editor.

command line interface: also known as “console”, “terminal”, “command prompt”, “shell”, or “BASH”. This is the DOS-style text interface. A **root console** is one where administrative privileges have been acquired, either by “su” or by logging in as root.

desktop environment: The software which provides the graphical environment (windows, icons, desktop, taskbar, etc) for an operating system. MEPIS uses KDE by default.

file system: also “filesystem”. In this case, it refers to the way that files and folders are logically arranged on a computer's storage devices. It can also refer to the type of formatting on a storage device, such as NTFS, FAT32, or EXT3. We are using it in the first sense in this article.

fork: This is when two different developers or development teams decide to take a piece of software in different directions, resulting in two different programs from the same original source code.

free-as-in-speech: The English word “free” has two possible meanings: “without cost” and “without restrictions”. In the open-source software community, the usual analogy used to explain the difference is “free as in beer” vs. “free as in speech”. The word “freeware” is used to refer to software that is simply without cost, whereas the phrase “free software” typically refers to software which is open source and licensed with some type of open license (such as the GPL).

GPL: The GNU General Public License. This is the license under which Linux and many open source applications are released. In a nutshell, it specifies that you may view, modify, and redistribute the source code of applications released under it, within certain limits. The complete GPL can be found at <http://www.gnu.org/copyleft/gpl.html>.

GUI: Graphical User Interface. This refers to a program or operating system interface that uses pictures (icons, windows, etc), as opposed to text-mode (command-line) interfaces.

kernel: The layer of software in an operating system that deals directly with the hardware.

live CD: A bootable compact disc from which one can run a full operating system, usually with a complete desktop environment, applications, and essential hardware functionality.

open-source: Software whose source code has been made available to the public. It also implies some kind of unrestricted licensing.

script: An executable text file, containing commands in an interpreted language. Usually refers to BASH scripts, but other languages can be used as well.

source code: The human-readable code in which software is written.

tarball: An archiving format, like zip, popular on the Linux platform. Unlike zip files, though, tarballs may use one of a number of different compression formats, such as gzip or bzip2. They usually end in file extensions like .tgz, .tar.gz, or .tar.bz2.

window manager: A component of a desktop environment that provides the basic maximize/minimize/close/move functions for windows in the GUI environment. Sometimes they can be used as an alternative to a full desktop environment.

Unix: (also UNIX) The operating system which Linux is modeled after, developed in the late 1960's at Bell Labs and used primarily for servers and mainframes. Like Linux, Unix has many variations; for more information, see this article: <http://en.wikipedia.org/wiki/Unix>.

X: also X11, X windows, xorg. The basic GUI engine for most Linux and Unix systems.

License Information

This work is copyright 2007 Alan D Moore. It is licensed under a Creative Commons Attribution - ShareAlike 3.0 unported license. This means you may legally redistribute or create derivative works based on this document provided you (1) provide attribution to the author, and (2) distribute this work or any derivatives under a compatible license.

The above is only a loose summary of the license terms. The full text of the license can be found at <http://creativecommons.org/licenses/by-sa/3.0/legalcode> An official “summary” of the license can be found at <http://creativecommons.org/licenses/by-sa/3.0>

Exception: The author of this work, in accordance with the provisions of the license, grants permission for inclusion of this work, in whole, in part, or in modified form, in the MEPIS community wiki and MEPIS documentation without the required attribution.

Attribution should be given as Alan D. Moore and include the URL <http://www.alandmoore.com>

Acknowledgements

Thanks to Warren Woodford for a great Linux distribution.

Thanks to the Mepislovers community for constructive criticisms and editing help.

Thanks to Jerry Bond for coming up with a better title than my original one.

Thanks to Linus and open source developers everywhere for all they do.