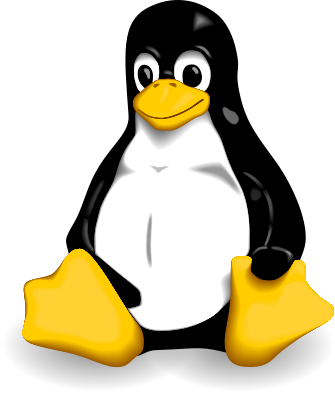
**LiNUx operating system**

**Linux** is a generic term referring to Unix-like computer operating systems based on the Linux kernel. Their development is one of the most prominent examples of free and open source software collaboration; typically all the underlying source code can be used, freely modified, and redistributed, both commercially and non-commercially, by anyone under licenses such as the GNU General Public License.

Linux can be installed on a wide variety of computer hardware, ranging from embedded devices such as mobile phones, smart phones and wristwatches to mainframes and supercomputers. Linux is predominantly known for its use in servers; in 2007 Linux's overall share of the server market was estimated at 12.7%, while a 2008 estimate suggested that 60% of all web servers ran Linux. Most desktop computes run either Mac OS X or Microsoft Windows, with Linux having only 1–2% of the desktop market. However, desktop use of Linux has become increasingly popular in recent years, partly owing to the popular Ubuntu distribution and the emergence of netbooks and smartbooks.

Typically Linux is packaged in a format known as a *Linux distribution* for desktop and server use. Linux distributions include the Linux kernel and all of the supporting software required to run a complete system, such as utilities and libraries, the X Window System, the GNOME and KDE desktop environments, and the Apache HTTP Server. Commonly-used applications with desktop Linux systems include the Mozilla Firefox web-browser, the OpenOffice.org office application suite and the GIMP image editor.

The name "Linux" comes from the Linux kernel, originally written in 1991 by Linus Torvalds. The main supporting Userland in the form of system tools and libraries from the GNU Project (announced in 1983 by Richard Stallman) is the basis for the Free Software Foundation's preferred name *GNU/Linux*.

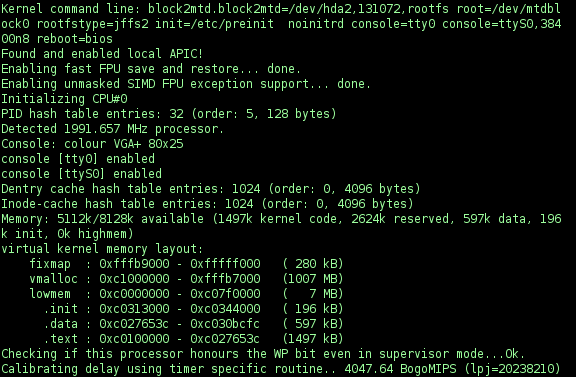
**Design**

A Linux-based system is a modular Unix-like operating system. It derives much of its basic design from principles established in Unix during the 1970s and 1980s. Such a system uses a monolithic kernel, the Linux kernel, which handles process control, networking, and peripheral and file system access. Device drivers are integrated directly with the kernel.

Separate projects that interface with the kernel provide much of the system's higher-level functionality. The GNU userland is an important part of most Linux-based systems, providing the most common implementation of the C library, a popular shell, and many of the common Unix tools which carry out many basic operating system tasks. The graphical user interface (or GUI) used by most Linux systems is built on top of an implementation of the X Window System.

**Linux Kernel**

The **Linux kernel** is an operating system kernel used by the Linux family of Unix-like operating systems. It is one of the most prominent examples of free and open source software.



The Linux kernel is released under the GNU General Public License version 2 (GPLv2), (plus some firmware images with various licenses), and is developed by contributors worldwide. Day-to-day development takes place on the Linux kernel mailing list. The Linux kernel was initially conceived and created by Finnish computer science student Linus Torvalds in 1991. Linux rapidly accumulated developers and users who adopted code from other free software projects for use with the new operating system. The Linux kernel has received contributions from thousands of programmers. Many Linux distributions have been released based upon the Linux kernel.

**User Interface**

Users can control a Linux-based system through a command line interface (or CLI), a graphical user interface (or GUI), or through controls attached to the associated hardware (this is common for embedded systems). For desktop systems, the default mode is usually graphical user interface, where the CLI is available through terminal emulator windows or on a separate virtual console.

On desktop machines, KDE, GNOME, and Xfce are the most popular user interfaces, though a variety of additional user interfaces exist. Most popular user interfaces run on top of the X Window System (often simply called "X"), which provides network transparency, enabling a graphical application running on one machine to be displayed and controlled from another.

Other GUIs include X window managers such as FVWM, Enlightenment, and Window Maker. The window manager provides a means to control the placement and appearance of individual application windows, and interacts with the X Window System. This is a more minimalist goal than KDE, GNOME et al., which are termed desktop environments.

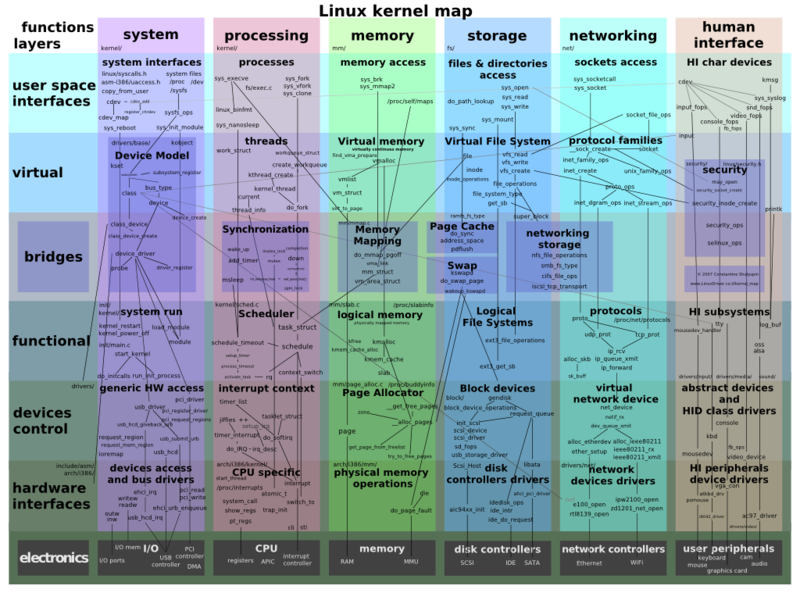
A Linux system typically provides a CLI through a shell, which is the traditional way of interacting with a Unix system. A Linux distribution specialized for servers may use the CLI as its only interface. A headless system that runs without even a monitor can be controlled by the command line via a remote-control protocol such as SSH or telnet.

Most low-level Linux components, including the GNU userland, use the CLI exclusively. The CLI is particularly suited for automation of repetitive or delayed tasks, and provides very simple inter-process communication. A graphical terminal emulator program is often used to access the CLI from a Linux desktop.

**Features**

Linux supports true preemptive multitasking (both in user mode and kernel mode), virtual memory, shared libraries, demand loading, shared copy-on-write executables, memory management, the Internet protocol suite, and threading.

**Architecture**

Linux is a monolithic kernel. Device drivers and kernel extensions run in kernel space (ring 0 in many CPU architectures), with full access to the hardware, although some exceptions run in user space. The graphics system most people use with Linux doesn't run in the kernel, in contrast to that found in Microsoft Windows.

Kernel mode preemption allows device drivers to be preempted under certain conditions. This feature was added to handle hardware interrupts correctly and improve support for symmetric multiprocessing (SMP). Preemption also improves latency, increasing responsiveness and making Linux more suitable for real-time applications.

**Kernel Panic**

In Linux, a "panic" is an unrecoverable system error detected by the kernel as opposed to similar errors detected by user space code. It is possible for kernel code to indicate such a condition by calling the panic function located in the header file sys/system.h. However, most panics are the result of unhandled processor exceptions in kernel code, such as references to invalid memory addresses. These are typically indicative of a bug somewhere in the call chain leading to the panic. They can also indicate a failure of hardware, such as a failed RAM cell or errors in arithmetic functions in the processor caused by a processor bug, overheating/damaged processor, or a soft error.

**Programming Languages**

Linux is written in the version of the C programming language supported by GCC (which has introduced a number of extensions and changes to standard C), together with a number of short sections of code written in the assembly language (in GCC's "AT&T-style" syntax) of the target architecture. Because of the extensions to C it supports, GCC was for a long time the only compiler capable of correctly building Linux. In 2004, Intel claimed to have modified the kernel so that its C compiler also was capable of compiling it. There was another such reported success in 2009 with a modified 2.6.22.

Many other languages are used in some way, primarily in connection with the kernel build process (the methods whereby the bootable image is created from the sources). These include Perl, Python, and various shell scripting languages. Some drivers may also be written in C++, Fortran, or other languages, but this is strongly discouraged. Linux's build system only officially supports GCC as a kernel and driver compiler.

**Portability**

While not originally designed to be portable, Linux is now one of the most widely ported operating system kernels, running on a diverse range of systems from the iPAQ (a handheld computer) to the IBM System z9 (a massive mainframe server that can run hundreds or even thousands of concurrent Linux instances). Linux runs as the main operating system on IBM's Blue Gene supercomputers. As of June 2009, Linux is the OS on more than 88% of systems on the Top 500 supercomputers list. Also, Linux has been ported to various handheld devices such as TuxPhone and Apple's iPod. The Google Android and Nokia Maemo operating systems, developed for mobile phone devices, both use modified versions of the Linux kernel.

**Virtual Machine Architectures**

The Linux kernel has extensive support for and runs on many virtual machine architectures both as the *host* operating system and as a *guest* operating system. The virtual machines usually emulate Intel x86 family of processors, though in a few cases PowerPC or AMD processors are also emulated.

**Estimated Cost to Redevelop**

The cost to redevelop the Linux kernel version 2.6.0 in a traditional proprietary development setting has been estimated to be $612 million USD (€467 million euro) in 2004 prices using the COCOMO man-month estimation model. In 2006, a study funded by the European Union put the redevelopment cost of kernel version 2.6.8 higher, at €882 million euro ($1.14 billion USD).

This topic was revisited in October 2008 by Amanda McPherson, Brian Proffitt and Ron Hale-Evans. Using David A. Wheelers methodology, they estimated redevelopment of the 2.6.25 kernel now costs $1.3 billion (part of a total $10.8 billion to redevelop Fedora 9).

**Uses**

As well as those designed for general purpose use on desktops and servers, distributions may be specialized for different purposes including: computer architecture support, embedded systems, stability, security, localization to a specific region or language, targeting of specific user groups, support for real-time applications, or commitment to a given desktop environment. Furthermore, some distributions deliberately include only free software. Currently, over three hundred distributions are actively developed, with about a dozen distributions being most popular for general-purpose use.

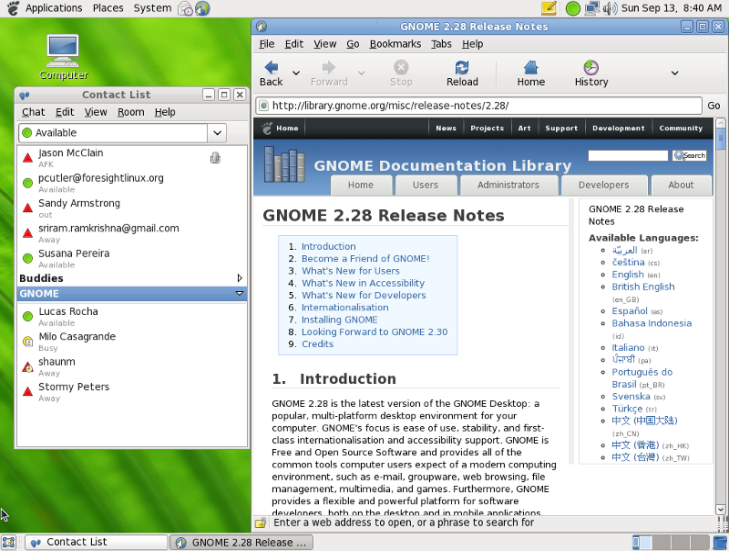
Linux is a widely ported operating system kernel. The Linux kernel runs on a highly diverse range of computer architectures: in the hand-held ARM-based iPAQ and the mainframe IBM System z9, System z10 in devices ranging from mobile phones to supercomputers. Specialized distributions exist for less mainstream architectures. The ELKS kernel fork can run on Intel 8086 or Intel 80286 16-bit microprocessors, while the µClinux kernel fork may run on systems without a memory management unit. The kernel also runs on architectures that were only ever intended to use a manufacturer-created operating system, such as Macintosh computers (with both PowerPC and Intel processors), PDAs, video game consoles, portable music players, and mobile phones.

There are several industry associations and hardware conferences devoted to maintaining and improving support for diverse hardware under Linux, such as FreedomHEC.

**Desktop**

The popularity of Linux on standard desktops (and laptops) has been increasing over the years. Currently most distributions include a graphical user environment. The two most popular such environments are GNOME and KDE, both of which are mature and support a wide variety of languages.

The performance of Linux on the desktop has been a controversial topic; for example in 2007 Con Kolivas accused the Linux community of favoring performance on servers. He quit Linux kernel development because he was frustrated with this lack of focus on the desktop, and then gave a "tell all" interview on the topic. Since then a significant effort has been expended improving the desktop experience. Projects such as upstart aim for a faster boot time. In the field of gaming, the Linux desktop still lags behind Windows. However, there are several companies that do port their own or other companies' games to Linux.

Many types of applications available for Microsoft Windows and Mac OS X are also available for Linux. Commonly, either a free software application will exist which does the functions of an application found on another operating systems, or that application will have a version that works on Linux (such as Skype). Furthermore, the Wine project provides a Windows compatibility layer to run unmodified Windows applications on Linux. CrossOver is a proprietary solution based on the open source Wine project that supports running Windows versions of Microsoft Office,Intuit applications such as Quicken and QuickBooks, Adobe Photoshop versions through CS2, and many popular games such as *World of Warcraft* and *Team Fortress 2*. In other cases, where there is no Linux port of some software in areas such as desktop publishing and professional audio, there is equivalent software available on Linux.

Many popular applications are available for a wide variety of operating systems. For example Mozilla Firefox and OpenOffice.org have downloadable versions for all major operating systems. Furthermore, some applications were initially developed for Linux (such as Pidgin, and GIMP) and, due to their popularity, were ported to other operating systems (including Windows and Mac OS X).

A growing number of proprietary desktop applications are also supported on Linux, see List of proprietary software for Linux. In the field of animation and visual effects, most high end software, such as AutoDesk Maya, Softimage XSI and Apple Shake, is available for Linux, Windows and/or Mac OS X.

The collaborative nature of free software development allows distributed teams to localize Linux distributions for use in locales where localizing proprietary systems would not be cost-effective. For example the Sinhalese language version of the Knoppix distribution was available significantly before Microsoft Windows XP was translated to Sinhalese. In this case the Lanka Linux User Group played a major part in developing the localized system by combining the knowledge of university professors, linguists, and local developers.

To install new software in Windows, users either download a digital distribution and double-click to install it or use a traditional installation medium (such as CD-ROM). Both of these methods usually provide a "Software Installation Wizard" to guide the user through the setup. On most Linux distributions, there are utilities for browsing a list of thousands of applications installed with a single click. Some of these programs are the Synaptic Package Manager, PackageKit, and Yum Extender. However, installing software not in the official repositories is not always as easy - especially for users new to Linux - and sometimes the only option is to compile from source.

**Servers, Mainframes and Supercomputers**

Linux distributions have long been used as server operating systems, and have risen to prominence in that area; Netcraft reported in September 2006 that eight of the ten most reliable internet hosting companies ran Linux distributions on their web servers. (As of June 2008, Linux distributions represented five of the top ten, FreeBSD three of ten, and Microsoft two of ten; as of February 2010, Linux distributions represented six of the top ten, FreeBSD two of ten, and Microsoft one of ten.)

Linux distributions are the cornerstone of the LAMP server-software combination (Linux, Apache, MySQL, Perl/PHP/Python) which has achieved popularity among developers, and which is one of the more common platforms for website hosting.

Linux distributions have become increasingly popular on mainframes in the last decade due to pricing, compared to other mainframe operating systems. In December 2009, computer giant IBM reported that it would predominantly market and sell mainframe-based Enterprise Linux Server.

Linux distributions are also commonly used as operating systems for supercomputers: as of November 2009, out of the top 500 systems, 446 (89.2%) run a Linux distribution.

Linux was also selected as the operating system for the world's most powerful supercomputer, IBM's Sequoia which will become operational in 2011.

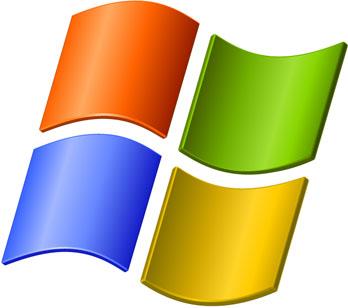
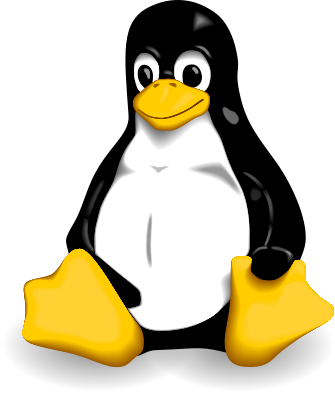
**Embedded devices**

Due to its low cost and ability to be easily modified, an embedded Linux is often used in embedded systems. Linux has become a major competitor to Symbian OS which is used in the majority of smartphones—16.7% of smartphones sold worldwide during 2006 were using Linux—and it is an alternative to the proprietary Windows CE and Palm OS operating systems on mobile devices. Cell phones or PDAs running on Linux and built on open source platform became a trend from 2007, like Nokia N810, Openmoko's Neo1973, Motorola RAZR2 v8,Motorola ROKR E8, Motorola MING series, Motorola ZINE and Google Android. The popular TiVo digital video recorder uses a customized version of Linux. Several network firewall and router standalone products, including several from Cisco/Linksys, use Linux internally, using its advanced firewall and routing capabilities. The Korg OASYS, Roland RD-700GX and the Yamaha Motif XS music workstations also run Linux. Furthermore, Linux is used in the leading stage lighting control system, FlyingPig/HighEnd WholeHogIII Console.

**Comparison between UNIX and MS-DOS**

UNIX and MS-DOS have much in common, primarily because MS-DOS was partly developed using ideas from UNIX. Here are some of the major differences:

1. UNIX filenames are arbitrary strings of up to 255 characters, MS-DOS filenames are limited to an 8 character file name and a 3 character file type, and have restrictions on allowable characters.
2. UNIX doesn't require the notion of a disk drive modifier for a file specification, i.e., you don't need A:, B:, C:, and so on like MS-DOS. Note that MS-DOS has the JOIN command which can simulate the UNIX file system.
3. UNIX uses forward slashes (/) to separate directories in a file specification, whereas MS-DOS uses backslashes (  ).
4. UNIX is a full multiuser multitasking operating system, whereas MS-DOS is a single-user OS without concepts like file-ownership and process priorities.
5. UNIX has a richer command set than MS-DOS.
6. UNIX and MS-DOS both have the concept of a PATH.
7. UNIX and MS-DOS both have pipes and I/O redirection.
8. UNIX displays characters on your terminal as you type them, regardless of whether the operating system is ready to accept a command. MS-DOS will buffer, but not display, terminal input when it is busy executing the previous command.
9. MS-DOS programs generally have a nicer user interface than traditional UNIX ones. This situation is changing with the emergence of X-windows applications on UNIX.

**Comparison between Linux and Windows**

1. **Full access vs. no access**

Having access to the source code is probably the single most significant difference between Linux and Windows. The fact that Linux belongs to the GNU Public License ensures that users (of all sorts) can access (and alter) the code to the very kernel that serves as the foundation of the Linux operating system. You want to peer at the Windows code? Good luck. Unless you are a member of a very select (and elite, to many) group, you will never lay eyes on code making up the Windows operating system.

You can look at this from both sides of the fence. Some say giving the public access to the code opens the operating system (and the software that runs on top of it) to malicious developers who will take advantage of any weakness they find. Others say that having full access to the code helps bring about faster improvements and bug fixes to keep those malicious developers from being able to bring the system down. I have, on occasion, dipped into the code of one Linux application or another, and when all was said and done, was happy with the results. Could I have done that with a closed-source Windows application? No.

1. **Licensing freedom vs. licensing restrictions**

Along with access comes the difference between the licenses. I’m sure that every IT professional could go on and on about licensing of PC software. But let’s just look at the key aspect of the licenses (without getting into legalese). With a Linux GPL-licensed operating system, you are free to modify that software and use and even republish or sell it (so long as you make the code available). Also, with the GPL, you can download a single copy of a Linux distribution (or application) and install it on as many machines as you like. With the Microsoft license, you can do none of the above. You are bound to the number of licenses you purchase, so if you purchase 10 licenses, you can legally install that operating system (or application) on only 10 machines.

1. **Online peer support vs. paid help-desk support**

This is one issue where most companies turn their backs on Linux. But it’s really not necessary. With Linux, you have the support of a huge community via forums, online search, and plenty of dedicated Web sites. And of course, if you feel the need, you can purchase support contracts from some of the bigger Linux companies (Red Hat and Novell for instance).

However, when you use the peer support inherent in Linux, you do fall prey to time. You could have an issue with something, send out e-mail to a mailing list or post on a forum, and within 10 minutes be flooded with suggestions. Or these suggestions could take hours of days to come in. It seems all up to chance sometimes. Still, generally speaking, most problems with Linux have been encountered and documented. So chances are good you’ll find your solution fairly quickly.

On the other side of the coin is support for Windows. Yes, you can go the same route with Microsoft and depend upon your peers for solutions. There are just as many help sites/lists/forums for Windows as there are for Linux. And you can purchase support from Microsoft itself. Most corporate higher-ups easily fall victim to the safety net that having a support contract brings. But most higher-ups haven’t had to depend up on said support contract. Of the various people I know who have used either a Linux paid support contract or a Microsoft paid support contract, I can’t say one was more pleased than the other. This of course begs the question “Why do so many say that Microsoft support is superior to Linux paid support?”

1. **Full vs. partial hardware support**

One issue that is slowly becoming nonexistent is hardware support. Years ago, if you wanted to install Linux on a machine you had to make sure you hand-picked each piece of hardware or your installation would not work 100 percent. I can remember, back in 1997-ish, trying to figure out why I couldn’t get Caldera Linux or Red Hat Linux to see my modem. After much looking around, I found I was the proud owner of a Winmodem. So I had to go out and purchase a US Robotics external modem because that was the one modem I *knew* would work. This is not so much the case now. You can grab a PC (or laptop) and most likely get one or more Linux distributions to install and work nearly 100 percent. But there are still some exceptions. For instance, hibernate/suspend remains a problem with many laptops, although it has come a long way.

With Windows, you know that most every piece of hardware will work with the operating system. Of course, there are times (and I have experienced this over and over) when you will wind up spending much of the day searching for the correct drivers for that piece of hardware you no longer have the install disk for. But you can go out and buy that 10-cent Ethernet card and know it’ll work on your machine (so long as you have, or can find, the drivers). You also can rest assured that when you purchase that insanely powerful graphics card, you will probably be able to take full advantage of its power.

1. **Command line vs. no command line**

No matter how far the Linux operating system has come and how amazing the desktop environment becomes, the command line will always be an invaluable tool for administration purposes. Nothing will ever replace my favorite text-based editor, ssh, and any given command-line tool. I can’t imagine administering a Linux machine without the command line. But for the end user — not so much. You could use a Linux machine for years and never touch the command line. Same with Windows. You can still use the command line with Windows, but not nearly to the extent as with Linux. And Microsoft tends to obfuscate the command prompt from users. Without going to Run and entering cmd (or command, or whichever it is these days), the user won’t even know the command-line tool exists. And if a user does get the Windows command line up and running, how useful is it really?

1. **Centralized vs. non-centralized application installation**

The heading for this point might have thrown you for a loop. But let’s think about this for a second. With Linux you have (with nearly every distribution) a centralized location where you can search for, add, or remove software. I’m talking about package management systems, such as Synaptic. With Synaptic, you can open up one tool, search for an application (or group of applications), and install that application without having to do any Web searching (or purchasing).

Windows has nothing like this. With Windows, you must know where to find the software you want to install, download the software (or put the CD into your machine), and run setup.exe or install.exe with a simple double-click. For many years, it was thought that installing applications on Windows was far easier than on Linux. And for many years, that thought was right on target. Not so much now. Installation under Linux is simple, painless, and centralized.

1. **Flexibility vs. rigidity**

I always compare Linux (especially the desktop) and Windows to a room where the floor and ceiling are either movable or not. With Linux, you have a room where the floor and ceiling can be raised or lowered, at will, as high or low as you want to make them. With Windows, that floor and ceiling are immovable. You can’t go further than Microsoft has deemed it necessary to go.

Take, for instance, the desktop. Unless you are willing to pay for and install a third-party application that can alter the desktop appearance, with Windows you are stuck with what Microsoft has declared is the ideal desktop for you. With Linux, you can pretty much make your desktop look and feel exactly how you want/need. You can have as much or as little on your desktop as you want. From simple flat Fluxbox to a full-blown 3D Compiz experience, the Linux desktop is as flexible an environment as there is on a computer.

1. **Fanboys vs. corporate types**

I wanted to add this because even though Linux has reached well beyond its school-project roots, Linux users tend to be soapbox-dwelling fanatics who are quick to spout off about why you should be choosing Linux over Windows. I am guilty of this on a daily basis (I try hard to recruit new fanboys/girls), and it’s a badge I wear proudly. Of course, this is seen as less than professional by some. After all, why would something worthy of a corporate environment have or need cheerleaders? Shouldn’t the software sell itself? Because of the open source nature of Linux, it has to make do without the help of the marketing budgets and deep pockets of Microsoft. With that comes the need for fans to help spread the word. And word of mouth is the best friend of Linux.

Some see the fanaticism as the same college-level hoorah that keeps Linux in the basements for LUG meetings and science projects. But I beg to differ. Another company, thanks to the phenomenon of a simple music player and phone, has fallen into the same fanboy fanaticism, and yet that company’s image has not been besmirched because of that fanaticism. Windows does not have these same fans. Instead, Windows has a league of paper-certified administrators who believe the hype when they hear the misrepresented market share numbers reassuring them they will be employable until the end of time.

1. **Automated vs. non-automated removable media**

I remember the days of old when you had to mount your floppy to use it and unmount it to remove it. Well, those times are drawing to a close — but not completely. One issue that plagues new Linux users is how removable media is used. The idea of having to manually “mount” a CD drive to access the contents of a CD is completely foreign to new users. There is a reason this is the way it is. Because Linux has always been a multiuser platform, it was thought that forcing a user to mount a media to use it would keep the user’s files from being overwritten by another user. Think about it: On a multiuser system, if everyone had instant access to a disk that had been inserted, what would stop them from deleting or overwriting a file you had just added to the media? Things have now evolved to the point where Linux subsystems are set up so that you can use a removable device in the same way you use them in Windows. But it’s not the norm. And besides, who doesn’t want to manually edit the */etc/fstab* fle?

1. **Multilayered run levels vs. a single-layered run level**

I couldn’t figure out how best to title this point, so I went with a description. What I’m talking about is Linux’ inherent ability to stop at different run levels. With this, you can work from either the command line (run level 3) or the GUI (run level 5). This can really save your socks when X Windows is fubared and you need to figure out the problem. You can do this by booting into run level 3, logging in as root, and finding/fixing the problem.

With Windows, you’re lucky to get to a command line via safe mode — and then you may or may not have the tools you need to fix the problem. In Linux, even in run level 3, you can still get and install a tool to help you out (hello apt-get install APPLICATION via the command line). Having different run levels is helpful in another way. Say the machine in question is a Web or mail server. You want to give it all the memory you have, so you don’t want the machine to boot into run level 5. However, there are times when you do want the GUI for administrative purposes (even though you can fully administer a Linux server from the command line). Because you can run the *startx*command from the command line at run level 3, you can still start up X Windows and have your GUI as well. With Windows, you are stuck at the Graphical run level unless you hit a serious problem.

**Name : Rohit Aggarwal**

**Roll No. : 7CS-097**