**Introduction to Linux**

Linux is a Unix-like computer operating system assembled under the model of free and open source software development and distribution. The defining component of Linux is the Linux kernel, an operating system kernel first released 5 October 1991 by Linus Torvalds.

Linux was originally developed as a free operating system for Intel x86-based personal computers. It has since been ported to more computer hardware platforms than any other operating system. It is a leading operating system on servers and other big iron systems such as mainframe computers and supercomputers more than 90% of today's 500 fastest supercomputers run some variant of Linux, including the 10 fastest. Linux also runs on embedded systems (devices where the operating system is typically built into the firmware and highly tailored to the system) such as mobile phones, tablet computers, network routers, televisions and video game consoles; the Android system in wide use on mobile devices is built on the Linux kernel.

A distribution oriented toward desktop use will typically include the X Window System and an accompanying desktop environment such as GNOME or KDE Plasma. Some such distributions may include a less resource intensive desktop such as LXDE or Xfce for use on older or less powerful computers. A distribution intended to run as a server may omit all graphical environments from the standard install and instead include other software such as the Apache HTTP Server and an SSH server such as OpenSSH. Because Linux is freely redistributable, anyone may create a distribution for any intended use. Applications commonly used with desktop Linux systems include the Mozilla Firefox web browser, the LibreOffice office application suite, and the GIMP image editor.Since the main supporting user space system tools and libraries originated in the GNU Project, initiated in 1983 by Richard Stallman, the Free Software Foundation prefers the name *GNU/Linux*.

**History of Unix**

The Unix operating system was conceived and implemented in 1969 at AT&T's Bell Laboratories in the United States by Ken Thompson, Dennis Ritchie, Douglas McIlroy, and Joe Ossanna. It was first released in 1971 and was initially entirely written in assembly language, a common practice at the time. Later, in a key pioneering approach in 1973, Unix was re-written in the programming language C by Dennis Ritchie (with exceptions to the kernel and I/O). The availability of an operating system written in a high-level language allowed easier portability to different computer platforms.

Today, Linux systems are used in every domain, from embedded systems to supercomputers, and have secured a place in server installations often using the popular LAMP application stack. Use of Linux distributions in home and enterprise desktops has been growing. They have also gained popularity with various local and national governments. The federal government of Brazil is well known for its support for Linux. News of the Russian Linux Programming

military creating its own Linux distribution has also surfaced, and has come to fruition as the G.H.ost Project. The Indian state of Kerala has gone to the extent of mandating that all state high schools run Linux on their computers.

**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 either integrated directly with the kernel or added as modules loaded while the system is running.

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.

**Programming on Linux**

Most Linux distributions support dozens of programming languages. The original development tools used for building both Linux applications and operating system programs are found within the GNU toolchain, which includes the GNU Compiler Collection (GCC) and the GNU build system. Amongst others, GCC provides compilers for Ada, C, C++, Java, and Fortran. First released in 2003, the Low Level Virtual Machine project provides an alternative open-source compiler for many languages. Proprietary compilers for Linux include the Intel C++ Compiler, Sun Studio, and IBM XL C/C++ Compiler. BASIC in the form of Visual Basic is supported in such forms as Gambas, FreeBASIC, and XBasic.

Most distributions also include support for PHP, Perl, Ruby, Python and other dynamic languages. While not as common, Linux also supports C# (via Mono), Vala, and Scheme. A number of Java Virtual Machines and development kits run on Linux, including the original Sun Microsystems JVM (HotSpot), and IBM's J2SE RE, as well as many open-source projects like Kaffe and JikesRVM.

**Linux Advantages**

1. **Low cost:** You don’t need to spend time and money to obtain licenses since Linux and much of its software come with the GNU General Public License. You can start to work immediately without worrying that your software may stop working anytime because the free trial version expires. Additionally, there are large repositories from which you can freely download high quality software for almost any task you can think of.

2. **Stability:** Linux doesn’t need to be rebooted periodically to maintain performance levels. It doesn’t freeze up or slow down over time due to memory leaks and such. Continuous up-times of hundreds of days (up to a year or more) are not uncommon.

3. **Performance:** Linux provides persistent high performance on workstations and on networks. It can handle unusually large numbers of users simultaneously, and can make old computers sufficiently responsive to be useful again.

Linux Programming

4. **Network friendliness:** Linux was developed by a group of programmers over the Internet and has therefore strong support for network functionality; client and server systems can be easily set up on any computer running Linux. It can perform tasks such as network backups faster and more reliably than alternative systems.

5. **Flexibility:** Linux can be used for high performance server applications, desktop applications, and embedded systems. You can save disk space by only installing the components needed for a particular use. You can restrict the use of specific computers by installing for example only selected office applications instead of the whole suite.

6. **Compatibility:** It runs all common Unix software packages and can process all common file formats.

7. **Choice:** The large number of Linux distributions gives you a choice. Each distribution is developed and supported by a different organization. You can pick the one you like best; the core functionalities are the same; most software runs on most distributions.

8. **Fast and easy installation:** Most Linux distributions come with user-friendly installation and setup programs. Popular Linux distributions come with tools that make installation of additional software very user friendly as well.

9. **Full use of hard disk:** Linux continues work well even when the hard disk is almost full.

10. **Multitasking:** Linux is designed to do many things at the same time; e.g., a large printing job in the background won’t slow down your other work.

11. **Security:** Linux is one of the most secure operating systems. “Walls” and flexible file access permission systems prevent access by unwanted visitors or viruses. Linux users have to option to select and safely download software, free of charge, from online repositories containing thousands of high quality packages. No purchase transactions requiring credit card numbers or other sensitive personal information are necessary.

12. **Open Source:** If you develop software that requires knowledge or modification of the operating system code, Linux’s source code is at your fingertips. Most Linux applications are Open Source as well.

**The difference between Linux and UNIX operating systems?** UNIX is copyrighted name only big companies are allowed to use the UNIX copyright and name, so IBM AIX and Sun Solaris and HP-UX all are UNIX operating systems. The Open Group holds the UNIX trademark in trust for the industry, and manages the UNIX trademark licensing program.

Most UNIX systems are commercial in nature.

**Linux is a UNIX Clone**

But if you consider Portable Operating System Interface (POSIX) standards then Linux can be considered as UNIX. To quote from Official Linux kernel README file:

Linux is a Unix clone written from scratch by Linus Torvalds with assistance from a loosely-knit team of hackers across the Net. It aims towards POSIX compliance.

However, "Open Group" do not approve of the construction "Unix-like", and consider it misuse of their UNIX trademark.

**Linux Is Just a Kernel** Linux Programming

Linux is just a kernel. All Linux distributions includes GUI system + GNU utilities (such as cp, mv, ls,date, bash etc) + installation & management tools + GNU c/c++ Compilers + Editors (vi) + and various applications (such as OpenOffice, Firefox). However, most UNIX operating systems are considered as a complete operating system as everything come from a single source or vendor.

As I said earlier Linux is just a kernel and Linux distribution makes it complete usable operating systems by adding various applications. Most UNIX operating systems comes with A-Z programs such as editor, compilers etc. For example HP-UX or Solaris comes with A-Z programs.

**License and cost**

Linux is Free (as in beer [freedom]). You can download it from the Internet or redistribute it under GNU licenses. You will see the best community support for Linux. Most UNIX like operating systems are not free (but this is changing fast, for example OpenSolaris UNIX). However, some Linux distributions such as Redhat / Novell provides additional Linux support, consultancy, bug fixing, and training for additional fees.

**User-Friendly**

Linux is considered as most user friendly UNIX like operating systems. It makes it easy to install sound card, flash players, and other desktop goodies. However, Apple OS X is most popular UNIX operating system for desktop usage.

**Security Firewall Software**

Linux comes with open source netfilter/iptables based firewall tool to protect your server and desktop from the crackers and hackers. UNIX operating systems comes with its own firewall product (for example Solaris UNIX comes with ipfilter based firewall) or you need to purchase a 3rd party software such as Checkpoint UNIX firewall.

**Backup and Recovery Software**

UNIX and Linux comes with different set of tools for backing up data to tape and other backup media. However, both of them share some common tools such as tar, dump/restore, and cpio etc.

**File Systems**

▪ Linux by default supports and use ext3 or ext4 file systems.

▪ UNIX comes with various file systems such as jfs, gpfs (AIX), jfs, gpfs (HP-UX), jfs, gpfs (Solaris).

**System Administration Tools**

1. UNIX comes with its own tools such as SAM on HP-UX.

2. Suse Linux comes with Yast

3. Redhat Linux comes with its own gui tools called redhat-config-\*.

However, editing text config file and typing commands are most popular options for sys admin work under UNIX and Linux.

**System Startup Scripts**

Almost every version of UNIX and Linux comes with system initialization script but they are located in different directories:

1. HP-UX - /sbin/init.d 2. AIX - /etc/rc.d/init.d 3. Linux - /etc/init.d Linux Programming

**End User Perspective**

The differences are not that big for the average end user. They will use the same shell (e.g. bash or ksh) and other development tools such as Perl or Eclipse development tool.

**System Administrator Perspective**

Again, the differences are not that big for the system administrator. However, you may notice various differences while performing the following operations:

1. Software installation procedure

2. Hardware device names

3. Various admin commands or utilities

4. Software RAID devices and mirroring

5. Logical volume management

6. Package management

7. Patch management

**UNIX Operating System Names**

A few popular names:

1. HP-UX

2. IBM AIX

3. Sun Solairs

4. Mac OS X

**5. IRIX**

**Linux Distribution (Operating System) Names**

A few popular names:

1. Redhat Enterprise Linux

2. Fedora Linux

3. Debian Linux

4. Suse Enterprise Linux

5. Ubuntu Linux

**Common Things Between Linux & UNIX**

Both share many common applications such as:

1. GUI, file, and windows managers (KDE, Gnome)

2. Shells (ksh, csh, bash)

3. Various office applications such as OpenOffice.org

4. Development tools (perl, php, python, GNU c/c++ compilers)

5. Posix interface

FUNDAMENTAL DIFFERENCES 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. Linux Programming

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.

**#2: 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.

**#3: 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?”

**#4: 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 Linux Programming

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.

**#5: 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?

**#6: Centralized vs. noncentralized 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.

**#7: 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 Linux Programming

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.

**#8: 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.

**#9: Automated vs. nonautomated 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?

**#10: 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 Linux Programming

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.

**FILE HANDLING UTILITIES:**

|  |  |
| --- | --- |
| **cat Command:** cat linux command concatenates files and print it on the standard output. **SYNTAX:** The Syntax is cat [OPTIONS] [FILE]... **OPTIONS:** -A | Show all. |
| -b | Omits line numbers for blank space in the output. |
| -e | A $ character will be printed at the end of each line prior to a new line. |
| -E | Displays a $ (dollar sign) at the end of each line. |
| -n | Line numbers for all the output lines. |
| -s | If the output has multiple empty lines it replaces it with one empty line. |
| -T | Displays the tab characters in the output. |
| -v | Non-printing characters (with the exception of tabs, new-lines and form-feeds) are printed visibly. |