Studying Main Differences Between Linux & Windows Operating Systems

Lecturer/ Hadeel Tariq Al-Rayes

***Abstract*—Comparisons between the Microsoft Windows and Linux computer operating systems are a long-running discussion topic within the personal computer industry. Throughout the entire period of the Windows 9x systems through the introduction of Windows 7, Windows has retained an extremely large retail sales majority among operating systems for personal desktop use, while Linux has sustained its status as the most prominent Free Software and Open Source operating system. After their initial clash, both operating systems moved beyond the user base of the personal computer market and share a rivalry on a variety of other devices, with offerings for the server and embedded systems markets, and mobile internet access.**

Linux and Microsoft Windows differ in philosophy, cost, versatility and stability, with each seeking to improve in their perceived weaker areas. Comparisons of the two operating systems tend to reflect their origins, historic user bases and distribution models.

***Index Term*— Kernel, Linux, Operating Systems, Windows**

I. INTRODUCTION

Linux is a Unix-like operating system that was designed to provide personal computer users a free or very low-cost operating system comparable to traditional and usually more expensive Unix systems. Linux has a reputation as a very efficient and fast-performing system. Linux's kernel (the central part of the operating system) was developed by Linus Torvalds at the University of Helsinki in Finland. To complete the operating system, Torvalds and other team members made use of system components developed by members of the Free Software Foundation for the GNU Project.

The researcher tried to give the main differences between the previous operating system and windows from many view points.

Lecturer/ Hadeel Tariq Al-Rayes
M.Sc. Computer Science
Basic Education College-Computer Science Department
Diyala University
hadeelalrayes@yahoo.com

II. THE ESSENTIAL DIFFERENCES BETWEEN LINUX & WINDOWS (BEGINNERS LEVEL)

1- Drives don't have letters, they have mountpoints

The first thing that usually trips up people who come from Windows to Linux is that filesystems aren't assigned letters the way they are in Windows. Instead, there is a single root filesystem whose path is "/". If you must use the Windows analogy to help you conceptualize it, you can think of this as the C:\ drive; it's the top of the filesystem. The Disk Analyzer Shows your Filesystem Usage and Layout Linux mounts new drives in folders inside the root filesystem. You'll notice that if you plug in a USB key, it will be mounted to a path like "/media/partition-name". This makes more sense because in Linux, there is a single filesystem layout, starting at root ("/" or "slash") and everything is located beneath it; for example, user files go in /home ("slash home"). If you want to keep your user files on a separate disk or partition, you don't have an E: drive; instead you simply mount your separate disk as /home.

2. There is no registry

Linux doesn't use a single database of configuration options, the way Windows does with its registry. Instead, there are many individual configuration files, typically in a simple text format (but increasingly in XML), that can be edited by hand using a text editor if need-be. You'll find the system-wide configuration files in the /etc directory; your user-specific configuration files will typically be located in hidden directories in your home folder. This is better than Windows because it means that there is no single point of failure for system configuration. If one configuration file becomes corrupt, only that function breaks and everything else works fine. It also makes it easy to backup configuration files — it's the same as copying all other files — and to troubleshoot problems: often if you ask for help on the internet, the first thing people will ask for is a copy of your configuration files.

3. Software comes from the repos, not CDs or websites

In the Windows world, if you need to find a program to perform a task you'll typically have to Google for it and install it using its own installer. Or, you can go to the store and

browse the CDs in the software isle. You wind up hoping that it uninstalls correctly later, and that you don't pick up viruses from some unknown program you found on an obscure website. Popular Linux distributions, by contrast, have done away with this "hunt and peck" style of software installation, and replaced it with the concept of "software repositories."
Add/Remove Programs in Fedora
When you want to install a program, you simply fire up the add/remove programs utility, search for the program you need, and install it. The package manager will figure out all the requirements and do the work for you — and the same goes for when you want to uninstall the program, too. This keeps your computer clean of unnecessary programs, and helps to make sure you don't wind up installing some program that is going to steal all your personal data.

4. Don't login as root
It's common practice for Windows users to login as the system administrator all the time. This is one of the major reasons why Windows computers are so easily infected with viruses and malware; you are always playing God, and any program can do anything it wants. In Linux, the administrator is called "root", and you should use root access only when you absolutely need it. Modern distributions and interfaces will prompt you for the root password when it's required, such as when you try to install programs or modify system settings. Aside from this, you should always log in as a regular user. You'll find that you don't need to be an administrator user all the time, and that your system survives much longer because of it!

5. Help is available — and it's free!
Changing to Linux can be fun and educational — but it can also be frustrating when you find something that doesn't work quite right, or when you can't figure out how to do something you need to get done. One of the best parts about switching to Linux is that there are plenty of
people who have done it before, and who were once in the same place you are now, and they are
more than happy to help you out. Don't hesitate to ask for help — see my article on eight ways to get help with Linux for tips on where to ask and how to get your questions answered.[1]

III. What will Android do for Linux?

Whether or not Android is a Linux distribution, it will still bring attention to the Linux operating system. Very few people in the mainstream understand that there is a difference between the Linux kernel and a Linux distribution. Android will be a Linux "flavor" whether it wants to be or not.When Android was only able to run on mobile phones, this was not too big of a deal. People are used to having different apps for the iPhone than there are for the Blackberry. Not being able to use an app designed for a LiMo based phone on an Android based phone is not an issue.

However, now that Android is being ported to Netbooks, the fact that it does not natively support applications designed for Linux may be a bit confusing at first. Fortunately, the operating system is completely open source. There are already suggestions for the creation of an Android application that allows for the use of a sandboxed Linux environment in order to run native Linux applications.[3]

IV. Main Differences between Linux & Windows (Programmers Level)

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.

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. 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

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.With Windows, you know that most every piece of hardware will work with the operating system. Of course, there are times 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).

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. 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.

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 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.[1]

8- 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.[2]

V. Main Differences between the previous two operating systems (Administrator Level)

1- User interface

a- Graphical user interface

In Linux, a number of desktop environments are available, of which GNOME and KDE are the most widely used.In windows, window manager is the Desktop Window Manager on Windows Vista, and a Stacking window manager built on top of GDI in older versions. The desktop environment may be modified by a variety of third party products such as WindowBlinds; or completely replaced, for example by Blackbox for Windows, or LiteStep.[3]

b- Command-line interface

Linux is strongly integrated with the system console. The command line can be used to recover the system if the graphics subsystem fails. In Windows, The Command Prompt exists to provide direct communication between the user and the operating system. A .NET-based command line environment called Windows PowerShell has been developed.[4]

2- Installation

a- Ease of Installation

In Linux, Varies greatly by distribution. Most distributions intended for new or intermediate users provide simple graphical installers. On Windows Server 2003 and prior, the installation is divided into two stages; the first, text-mode; the second, graphical. On Windows Vista and newer, the installation is single stage and graphical.

b- Device drivers

Linux kernels in most distributions include the majority of drivers available as modules. They are loaded at boot without user interaction. Most drivers are included in the kernel source tree, however there are several manufacturers which distribute proprietary drivers. The Windows installation media usually contains enough drivers to make the operating system functional. To this end, "generic" drivers may be used to provide basic functionality.[5]

c- Installation via Live Environments

Almost all Linux distributions now have a live CD that may be used for testing, install or recovery. In Windows, May be installed through the Windows Preinstallation Environment or BartPE, but only the former is endorsed by Microsoft.

d- Partitioning

In Linux, Most file systems support resizing partitions without losing data. LVM provide dynamic partitioning. All Linux distributions have bundled partitioning software such as fdisk or gparted. In Windows, expanding NTFS partitions is possible without problems, and on Vista it is possible to shrink partitions as well. Dynamic Disks provide dynamic partitioning.

e- File Systems

In Linux, Supported: ext2, ext3, ext4, ReiserFS, FAT, ISO 9660, UDF, NFS, NTFS, JFS, XFS, Minux and GmailFS. Archives and FTP sites also can be mounted as filesystems. Windows supported: NTFS, FAT, ISO 9660, UDF, and others; 3rd-party drivers available for ext2, reiserfs, HFS and the Dokan (a FUSE equivalent) UserSpace filesystem, which allows user-space programs to mount drives.[6]

3- Stability

In Linux, there are several indirection levels since all applications are separated from the graphic subsystem (X Server) which itself is detached from the Linux kernel. As a result of that and because most device drivers are integral parts of the Linux kernel, it almost never crashes. Windows operating systems based on the NT kernel (including all currently supported versions of desktop Windows) are technically much more stable than some older versions (including Windows 3.1 and 95/98), as these older versions do not properly protect the kernel's data structures.[7]

4- Performance

a- Process Scheduling

Linux kernel 2.6 once used a scheduling algorithm favoring interactive processes. Here "interactive" is defined as a process that has short bursts of CPU usage rather than long ones. It is said that a process without root privilege can take advantage of this to monopolize the CPU, when the CPU time accounting precision is low. NT-based versions of Windows use a CPU scheduler based on a multilevel feedback queue, with 32 priority levels defined. The kernel may change the priority level of a thread depending on its I/O and CPU usage and whether it is interactive (i. e. accepts and responds to input from the user), raising the priority of interactive and I/O bounded processes and lowering that of CPU bound processes, to increase the responsiveness of interactive applications.[8]

b- Memory Management Disk (Paging)

Most hard drive installations of Linux utilize a "swap partition", a partition dedicated exclusively for paging operations. This reduces slowdown due to disk fragmentation from general use. Windows NT family (including 2000, XP, Vista, Win7) most commonly employs a dynamically allocated pagefile for memory management. A pagefile is allocated on disk, for less frequently accessed objects in memory, leaving more RAM available to actively used objects.[9]

c- Default File Systems

Linux most commonly uses the Ext4 filesystem, which is unsupported by Windows. Ext4 avoids fragmenting the disk as much as possible, far more so than NTFS. Linux can, if desired by the user, install and run on an NTFS file system - though no mainstream distributions do this by default. The way the default Windows' file system NTFS works causes files to become fragmented, degrading the performance of the system significantly over time, and it requires regular defragmenting to combat this.[10]

5- Emulation and virtualization

a- Hardware emulation and virtualization

VMware, VirtualBox, Xen, Parallels, Win4Lin. KVM and QEMU can be used to run other operating systems within Linux. VMware, VirtualBox, Virtual PC, Virtual Server, Hyper-V (only available on 64-bit versions of Vista SP2, Windows 7, and Windows Server 2008), Parallels, QEMU can be used to run other operating systems within Windows.

b- Binary emulation, alternative API

In Linux, several projects including Bordeaux, Cedega, CrossOver, and Wine attempt to implement Windows API on top of Linux. In Windows, Several projects attempt to set up Unix-like environments in Windows. Only Microsoft Windows Services for UNIX is supported by Microsoft. Unsupported alternatives are LINA, and Cygwin.[9]

6- Security

a- Malware

More than 800 pieces of Linux malware had been discovered. Some malware has propagated through the Internet. However, in practice, reports of bonafide malware presence on Linux-based systems are extremely rare. Nonetheless, anti-malware tools such as ClamAV and Panda Security's DesktopSecure for Linux do exist. Once malicious software is present on a Windows-based system, it can sometimes be incredibly difficult to locate and remove.[11]

b- Open vs. Closed

In Linux, Claims its platform is more secure because all of its code is reviewed by so many people that bugs are detected (referred to as Linus's law). Anyone with programming experience is free to fix bugs and submit them for inclusion in future releases and updates. In Windows, Claims its platform is more secure because of a comprehensive approach to security using the Security Development Lifecycle. However,

because Windows is closed-source, only Microsoft-employed programmers (or licensed third-parties) can fix bugs.[12]

c- User Account

In Linux, Users typically run as limited accounts, having created both administrator (named "root") and at least one user account during installation. In Windows Vista, all logged-in sessions (even for those of "administrator" users) run with standard user permissions, preventing malicious programs (and inexperienced users) from gaining total control of the system.[13]

5-RELATED WORKS

1- "Shatter"

Some applications on Windows are running asboth a service and an interactive Graphics UserInterface (GUI) front end. And very often, theseservices are running under "LocalSystem" privilege.

Since they are also enabled to interact with users, amalicious user could take advantage of the GUI,specifically a text box to inject and redirect thesystem to run arbitrary code under the privileged"LocalSystem" [14], [15]. With a piece of specially crafted code called "shellcode", the malicious usercould escalate her privilege by spawning a new shellwith the all powerful "LocalSystem" privilege. Thistype of attack is dubbed "Shatter" attack, i.e.breaking the "Windows". "Shatter" attack takesadvantage of the inherent weakness within theWindows messaging infrastructure [14], [15], i.e. it does not check the security context of the sourceinput sent to a service running under the"LocalSystem" privilege.In spite of the claim that these vulnerabilitiescannot be fixed in Paget's papers [14], vendors ofthese vulnerable applications were able to fix theirsoftware so that they are no longer a threat. This isverified with the current versions of some softwarementioned in Lavery's paper [16]. Kerio PersonalFirewall (Version 4.1.3), Sygate Personal FirewallPro (Version 4), McAfee VirusScan (Version 7.0)and WinVNC (Version 4.1.1) are no longervulnerable to the "Shatter" attacks. As Pagetacknowledged in a later paper of his, these types ofvulnerabilities could be fixed by applying a technique called filtering [15]. What filtering does is basicallychecking the security context of the source of theinput and thus preventing un-trusted user fromsending specially crafted input to the privilegedservices. However, the best way of preventing thistype of attacks all together is to stop having aprivileged service to interact with users directly.

Instead, there should be a separated module to handlethe user interface. As a proxy, this module should notbe running under any privileged security context andthen this module will do the validation of the inputand pass the input onto the more privileged services.

2- Windows and Linux security models

The intent of this study, "An Approach to Analyzing the Windows and Linux Security Models", was to provide acomparison research study of the security relatedelements such as capabilities versus privilegesbetween the Linux and Windows operating systems.While the research was being conducted, we realizedthat a set of common metrics is lacking to quantifythe security risks in general among differentoperating systems. We decided to focus on the aspectof measurement of security risks. Specifically, thisstudy will examine features in the operating systems,which could potentially lead to privilege escalation. Itfurther proposes a set of metrics to assess thispotential. The goal is to show that there is a need toquantify and measure this type potential in a muchbroader domain of security issues, and to use this as aprototype to demonstrate what can be done.A formula is developed in the study to measurethe severity of security risks, which exist in certainfeatures of different operating systems. The result iscalled the risk factor, which can be used to predict thelevel of risks associated with the underlyingoperating system features. This provides a clear andeasy approach to assess risks involved, so that theresponsible information technology staff can bebetter informed.As an example to demonstrate how this set ofmetrics can help quantify the severity of the potentialsecurity risks, a specific feature in Windows isexamined.

The essence of this study is to demonstrate a setof proposed metrics for quantifying the security risksinvolved in different operating systems, andcomparing the risks between operating systems. Theproof of concept code implemented generatesanddemonstrates a scenario of how such risks can bemeasured using the metrics developed. The paperalso shows that the same set of metrics and formulacan be used for other operating systems such asLinux. Therefore, the comparison is not limited toonly within Windows, but similar features across themultiple operating systems can be compared.[17]

VI. DESCRIPTIVE STATISTICS

We'll examine basic descriptive statistics and several issues withrespect to our data. We begin by considering Table 1. The tableclearly shows that Windows is dominant in the server segment.In this table, the shares of each operating system are the meanvalues of the dummy variable for whether a firm uses the givenoperating system for the given segment. Because a firm can usemore than one kind of operating system, the sum of shares forserver.windows, server.linux, and server.other can be larger thanone. Firms may use multiple operating systems, either becauseof the complementarity between different operating systems, orbecause of potential testing—for example, a firm may use Windowsfor all servers, except one server for which it installs Linux to testwhether Linux would meet its need. Since this kind of testing raisesPanel B of Table 1 shows thatmost firms in our data use only one kind of operating system forthe server segment.

Table 2 presents the changes in the use of operating systemsand the number of computers in each segment over time.Three observations emerge from Table 2. First, the dominance ofWindows is persistent in both the server segment and the PCsegment, except for the non-PC segment, in which

other operatingsystems are the most popular, presumably because most non-PCsare IBM computers running IBM operating systems. The persistentdominance of Windows can be explained by either lock-in orunobserved preferences for Windows operating systems, whichwe investigate further in the next section.Second, the total number of server computers has increasedover time. If a firm purchased a new server computer, it is likely tohave made a decision on its server operating system. The increasein total.server throughout the sample period thus suggests that firms in our data are likely to have repeatedly made decisions ontheir server operating systems, which is one reason why we focuson the server segment. Another reason for focusing on the serversegment is that a substantial fraction of firms have adopted eitheran Internet server computer or a network server computer for thefirst time during our sample period. This is shown in Panel A ofTable 3, which reports that about 32.3% of firms have adoptedserver computers for the first time. For example, if a firm didnot use an Internet server until 2002, then there is no previousdecision on whether to use a particular operating system for anInternet server before 2002. Hence, the adoption decision of thisfirm in 2002 is less likely to depend on the previous decisions.20 Incontrast, the proportion of firms that adopted PCs for the first timeis insignificant in Table 3, though total.pc is increasing over timein Table 2. Notice also that total.non-pc is decreasing in Table 2,although the fractions of firms that adopted non-PC for the first time are not negligible in Table 3. Therefore, it is unclear whetherfirms have made decisions on their operating systems for PCs ornon-PCs frequently during our sample period, which is the otherreason why we focus only on the server segment.

Third, the use of Linux has increased in both the server segmentand the PC segment in Table 2, while the use of other operatingsystems has declined over time. One possibility for these trendsis that firms may have switched to Linux, not from Windows,but from a proprietary Unix operating system. However, it is alsopossible that firms have switched from Windows to Linux whileothers have simultaneously switched from Unix to Windows.

To examine these possibilities, we compute the fraction of firmsthat switched from an operating system to a different operatingsystem, where switching means that a firm used an operatingsystem before, and then stopped using it, while starting to use adifferent operating system at the same period. Table 3 presents theresults. Panel B shows that more firms switched from Windowsto Linux than from other operating systems to Linux, and that anontrivial number of firms switched from other operating systemsto Windows, thus suggesting that the presence of Windows hasalso affected the usage of Linux. Panel B also shows that asignificant fraction of firms did switch from one operating systemto another operating system in the server segment.Firms' decisions on server operating systems are not limitedto switching their operating systems. They also include updatingone version to another version of the same operating system.Panel C of Table 3 reports the fractions of firms that updated theiroperating systems, where updating means that a firm stoppedusing a version of an operating system (say, Windows 2000), andstarted to use a different version of the same family of the operatingsystem (say, Windows 2003).[18]

The table shows that about 85.9% of firms updated either Windows or Linux during our sample period.Therefore, both Panels B and C suggest that most firms in our dataindeed made decisions on either switching or updating at leastonce during our sample period. In our robustness checks, we alsorestrict our sample to the firms that made the usage decision morefrequently.

Panel D of Table 3 presents the proportion of firms that mighthave tested an operating system in the server segment, wheretesting an operating system means that a firm has used it fora single year while also continuing to use a different operatingsystem for the entire sample period. The table shows that onlya small fraction of firms tested an operating system during oursample period, and thus, the possibility of testing is unlikely to becritical in our data.[19]

TABLE I

Summary statistics[a].

A. Shares of operating systems	
server.windows	0.93
server.linux	0.13
server.other	0.25
B. Kinds of OS in servers	
Windows only	0.76
Linux only	0.04
Other only	0.08
Windows and Linux	0.07
Windows and other	0.14
Linux and other	0.01
C. Firm characteristics	
total.pc	251.7
total.non-pc	2.1
total.server	9.3
revenue (in $million)	59.6
employees	325.4
desk.workers	149.3
internet.users	113.4
internet.developers	0.8
programmers	3.4
#observations	36,690

[a] The table reports the mean of each variable in the 2000–2004 balanced panel data. The samples include only observations with any server operating system and with up-to-date information. The share is the mean of a dummy variable for whether an observation uses each operating system in the server segment.

TABLE II

Changes in the use of operating systems[a].

Variable	2000	2001	2002	2003	2004
server.windows	0.92	0.93	0.94	0.94	0.93
server.linux	0.08	0.12	0.15	0.17	0.15
server.other	0.29	0.27	0.25	0.24	0.22
pc.windows	0.96	0.98	0.99	0.98	0.98
pc.linux	0.07	0.13	0.15	0.18	0.17
pc.other	0.12	0.09	0.08	0.06	0.03
non-pc.windows	0.03	0.04	0.07	0.06	0.03
non-pc.linux	0.00	0.00	0.01	0.01	0.01
non-pc.other	0.42	0.31	0.31	0.22	0.12
total.pc	213.9	242.6	257.6	269.1	275.2
total.non-pc	2.34	2.25	2.71	1.98	1.04
total.server	7.10	8.14	9.65	10.32	11.07
#observations	7338	7338	7338	7338	7338

[a] The table reports the mean of each variable.

TABLE III

First-time computer adopters and switching patterns[a].

A. First-time adoption	
Adopted a server for the first time	0.323
Adopted a PC for the first time	0.020
Adopted a non-PC for the first time	0.239
B. Switching in servers	
Switching from Windows to Linux	0.135
Switching from other to Linux	0.066
Switching from Linux to Windows	0.115
Switching from other to Windows	0.182
Switching from Linux to other	0.043
Switching from Windows to other	0.120
C. Updating	
Updating in server.windows	0.646
Updating in server.linux	0.213
Updating in server.other	0.000
D. Testing	
Testing Linux in server	0.014
Testing Windows in server	0.009
Testing other in server	0.012

[a] The table first reports the fractions of the firms that did not have a computer in each segment and then adopted a computer in that segment for the first time during the sample period. The table next reports the fractions of the firms that updated, switched, or tested an operating system for the server segment during the sample period. Updating means that a firm stopped using a version of an operating system, and started to use a different version of the same family of the operating system. Switching means that a firm stopped using an operating system, and started to use a different operating system. Testing an operating system means that a firm did not use it before, and started to use it, and then stopped using it in the following year, while the firm also continued to use a different operating system for the entire sample period.

7-CONCLUSIONS

This study strongly suggests that IT professionals who are considering deployment of the workloads evaluated should consider far more than the acquisition costs of the technologies that they are investigating. Other factors, such as strategic IT choices, company standards, IT staff skills and competencies, application availability, application deployment, and performance considerations, should be considered as part of a total platform evaluation. IT professionals who are considering the broader strategic deployment of Linux within their IT environments, in particular, should carefully consider these findings and examine all aspects of the cost associated with Linux server systems. Many drivers of cost need to be uncovered in such an examination and evaluation, and the "risk/return"trade-offs of Linux versus Windows may not be as obvious as they appear at first glance.

REFERENCES

[1] Jonathan DePrizio, "5 Things New Linux Converts Should Know".
[2] http://www.techrepublic.com/blog/10things/10-fundamental-differences-between-linux-and-windows/406
[3] "Operating system Family share for 11/2010". November 2010. http://top500.org/stats/list/36/osfam.
[4] "Linux vs Windows (a comparison)". 20 June 2005. http://www.michaelhorowitz.com/Linux.vs.Windows.html#agui. Retrieved 16 March 2008.
[5] "Nvidia driver installation procedure". help.ubuntu.com. https://help.ubuntu.com/community/BinaryDriverHowto/Nvidia. Retrieved 20 February 2011.
[6] "Ext2 File System Driver for Windows". Sourceforge.net. http://sourceforge.net/projects/ext2fsd
[7] "Microsoft Halloween documents leak". http://www.catb.org/~esr/halloween/halloween2.html.
[8] Dan Tsafrir, Yoav Etsion, Dror G. Feitelson. "Secretly Monopolizing the CPU Without Superuser Privileges" (PDF). http://www.cs.huji.ac.il/~dants/papers/Cheat07Security.pdf. Retrieved 20 February 2011.
[9] "Swap Faq". Community Ubuntu Documentation. help.ubuntu.com. 26 November 2010. https://help.ubuntu.com/community/SwapFaq. Retrieved 20 February 2011.
[10] "Why doesn't Linux need defragmenting?". 17 August 2006. http://geekblog.oneandoneis2.org/index.php/2006/08/17/why_doesn_t_linux_need_defragmenting. Retrieved 20 February 2011.
[11] Technet.microsoft.com
[12] "Microsoft issues Vista security scorecard, gives itself an A-plus".http://www.computerworld.com/action/article.do?command=viewArticleBasic&articleId=9014080&pageNumber=2.
[13] Ubuntu.com.
[14] Chris Paget, "Next-Generation Win32 exploitsfundamental API flaws"
[15] Chris Paget, "More on Shatter".
[16] Oliver Lavery, "Win32 Message VulnerabilitiesRedux".
[17] Xinyue Song, Michael Stinson, Roger Lee, and Paul Albee,
[18] "An Approach to Analyzing the Windows and Linux Security Models"
[19] Computer Science Department, Central Michigan University, Mt Pleasant, MI 48859, U.S.A.
[20] Seetharaman, P., 2004." Modeling multiple sources of state dependence in random utility models: a distributed lag approach", Marketing Science 23,263–271
[21] Seung-Hyun Honga, Leonardo Rezendeb, "Lock-in and unobserved preferences in server operating systems: A case of Linux vs. Windows"

a Department of Economics, University of Illinois, United States
b Department of Economics, PUC-Rio, Brazil

Author's biography

Hadeel Tariq Ibrahim Al-Rayes. obtained her Bachelor inComputer Science from Baghdad University, College of Science and Master's degrees in Computer Science/Information Technology from Iraqi Committee for computers and Informatic, Informatic Institute for Postgraduate Studies.Iraq,Baghdad. She is currently working as a head of Computer Science Dept. in Basic Education Col., Diyala University, Iraq now.
E-Mail: hadeelalrayes@yahoo.com
Mobile: +964 7902 162334