

Abstract

The world of Unix is fragmented and while it is a much better operating system all around than any other especially Windows, it enjoys a very small market share apart from the web servers. There are many factors contributing to this. I will try to show some of these and most importantly the underlying reason for Unix's anti success behavior. I will be loosely using the term Unix for all variations of Linux.

I am not a Unix developer and in fact have not written a single line of code for it. The reason that I include myself in the Unix community and refer to it as "us" or "we" is purely due to the fact that I love Unix and all the people that have ever contributed to it through whatever means.

1. The philosophy

The main philosophy of Unix in its present form is freedom. This is not just a simple word or a simple claim. The Unix (Linux) community has shown it to be true. Many companies have taken advantage of Unix's superiority to make themselves fortunes, including Apple. But yet a lot more companies and many more talented individuals have worked so hard making applications or packages that at times are far more superior to their proprietary counterparts and they offer them for absolutely free. This is commendable. These people are making all this effort and they get paid nothing except the immense satisfaction that someone will be using their work to make

progress in doing what they choose. Why should the great effort made by all of us go to waste? I do not mean that we should charge fees for our software. If not enough people use Unix and its free and amazingly rich set of software, these efforts are wasted. Imagine Shakespeare writing all his works for free and giving it away for free for people to read but not enough people read them because the typeset is too small or the binding comes apart after a few pages or readers are required to read them in a vacuumed room sitting upside down hung from a rope. Meanwhile some Bill Yates makes a cheap copy of the same Shakespeare without the great content of the original and full of grammatical and spelling mistakes but publishes it in a shiny paperback form that you can read easily anywhere but charges an arm and a leg for every single copy and you can not even lend your book to a friend to read without risking jail sentences. This way Shakespeare with all his talent will remain known only to a few elite who care enough for his works to even read them hanging from a rope in a vacuumed room. The majority of the people will either never know about him or will only hear of him out of curiosity. What makes the matter even worse is when Shakespeare insists that this is the only way he can publish his works! We are in the same situation with Unix. I will show this to be the case in later sections.

The freedom from the costs associated with running an operating system and its supporting software is a giant step forward and not many people understand it fully.

According to my own experience, every time I recommend Unix to someone and tell them it is better than anything

else and it is also free, they invariably ask, “how then are they making a living giving away their work?”

See how honorable and ahead of our times we, the Unix people, are? In a world where everything is based on making huge profits and all the ideology is based on selling, making money and buying mansions, the Unix community is making all this effort just to help everyone enjoy this absolute necessity of our times, the computer and the Internet. This is by no means a small honor, this is the seed of the future of humanity where everyone shall do what they can for the common good and they will also be rewarded because others have provided for them in things they are good at providing. This is like the baker baking for free and giving away his bread while at the same time the butcher gives away his meat knowing that his bread is provided for him by the baker. Anyone does what they are good at doing and they are provided for by others who make other things they are good at. This is the reality of Unix and its people. This is the great honor they are bestowed. This is the future of humanity. This is living with and living for the 99% of the people. Let us know and feel that honor. Let us not waste it thinking that this is some simple philosophy. It seems that we are so accustomed to it that we tend to forget all about it.

Unix empowers anybody who can just afford the price of hardware to own a fully functional computer. Unix empowers people to freely do what they want with their computer with far better security and functionality. Can anyone imagine the Internet without Unix?

But if we don't act fast what happened to Novell a few years ago will eventually happen to Unix. Many old timers like me remember what happened with Netware. Microsoft had absolutely no comparable product to Netware. They only had a Lan Manager that was not even functional enough let alone well known. Microsoft pushed that broken Lan Manager so hard while Novell was comfortably sitting on its Netware complacently thinking nothing can beat it. Where is Netware now? Anybody hears about it anymore? But look at Microsoft's networking and all its offerings today. It is so dominant that even Unix has to be compatible with it. Many Unix distributions have special features to effortlessly connect to Microsoft networks but yet have problems connecting to another Unix network, case in point Ubuntu. I will get back to this later, just wanted to make a point here.

Another major point about Unix being free is that its developers need not make stupid changes just for making a newer version to sell and make more profit. Case in point is Microsoft Windows XP. We all know that Windows XP has everything a modern operating system needs albeit a lot inferior than Unix and, let's face it, at the same time a lot better than many Unix versions in many aspects. That is why for the first time in Microsoft's history, it was forced to accept a downgrade offer demanded by its customers. Many people refused to accept Vista or Windows 7 because they are all used to XP and it provides all they need to do any meaningful work with their computers. There are no new features in either one of the newer Windows offerings that really warrant an upgrade. They were simply made to

persuade an upgrade for the sake of making more revenue. Since Unix does not thrive to make money, it can stay the same for years and no one would complain as long as it does what a computer should do securely and effectively. Therefore most of the Unix development muscle can go to more applications and make only changes that are truly necessary and vital to the operating system itself. That is in line with Unix's other main philosophy, simplicity that works rather than complexity with lots of holes and non functionality.

2. The Paradigm

Before I delve into some of the pitfalls of Unix I would like to make a major point here. I want to show that it is not inevitable for Unix to work as it does today. It can work a lot better. I want to show why we tolerate Unix the way it is. I am not a Unix basher, I consider myself a Unix person despite the fact that I have no major experience with it. I admire Unix and every single Unix developer. I do not consider Unix developers stupid, quite the contrary they are geniuses. However, one thing prevents Unix users and developers from seeing a better way. That one single obstacle is their paradigm. A paradigm is a very powerful thing. It can bring about major changes and innovations but at some point it can become an obstacle in the way of innovation because we are so used to it that we can not see any other way. Here I would like to recommend watching the excellent video "Discovering the Future: The Business of Paradigms" by Joel Barker published in 1986. But since the video is so expensive, contrary to Unix philosophy, it

may be hard for most of us to watch. But I am sure a lot of us can find book versions of it rather cheaply and study it. Simply stated Mr. Barker shows that we do so many things because we are accustomed to doing them in a certain way and that alone prevents us from being able to see other ways that the same task can be done more effectively. Just imagine if a person who has worked all his life with an IBM punch computer finds out that he can simply type the same commands on a modern day computer all the time seeing it on a monitor and do the same thing better without ever knowing about a single punched card. I remember working with such a computer in my youth but it was not very long as to shape a paradigm in my attitude. But I experienced the same thing with a PC because I was used to an old VAX machine at the university and working with a PC represented a huge paradigm shift. I can go on and on about paradigms but I prefer to leave it to Mr. Barker to explain it and will go on to the discussion at hand here. Many Unix developers and users are happy with it the way it is and can not see it working any other way because they are so entrenched in their current methods that can not see other ways of doing things through Unix. While at the same time we expect all the other Windows users to adapt our way of thinking and start going through a steep learning curve just to find out that Windows way was indeed better. I am sure that if we put ourselves in the shoes of an average computer user and start seeing through his eyes, we will agree that we are indeed doing many things the harder way and can in fact change them and make them better. Let us all remember that Apple has always been based on Unix and it was for years the dream of Microsoft to copy it and

now that it has done it to a large extent it has enjoyed a much larger user base thanks to the non proprietary open standard IBM hardware. I think we can all still learn a lot from Steve Jobs in creating not one but two unique Unix based operating systems that are far more efficient to use than many of today's Unix distributions. Although the shortcoming of Mr. Jobs was that he finally succumbed to the glare of money and started creating junk rather than technology in his late years just to make Apple's stakeholders happy. Our role model should be Steve Jobs in his days at NeXT computer. Those not familiar with it are encouraged to take a brief look at the marvel he created but finally was forced to abandon it and only port its OS to current Apple computers.

To sum up this section, I do believe that Unix is stuck the way it is not because a bunch of idiots are at its helm but because a lot of hard working geniuses are working for it with an old paradigm. Once we change our paradigms Unix will fly like an eagle soaring to unlimited heights leaving everything else including Apple in the dust below.

3. The Problems

Most of the problems of Unix are invisible to the developers because they have been used to doing things the way they have thus far and they think that by nature and by design Unix is supposed to work the way it does. I believe that even if that is the case in some instances, it can and it should change even if it means that we change so many things from the ground up.

This section is by no means complete and it is not just a Unix wish list. I will enhance it over time as I see new areas that need improvement. However I am sure that if we change our paradigms I will not be alone and a lot of others will see a lot more problems and still lots more solutions. Almost all my comments are directed at Ubuntu and Debian since these are the only ones that I have worked on.

a) Installation of Unix

The installation of Unix is good. In fact it has one great advantage over Windows that can not be overlooked and that is the Live CD/ Live USB usage. This is a great advantage that permits the use of a computer from a single USB flash drive without touching the host computer's files and hard drive. The Live CD/ Live USB feature is a major breakthrough of Unix that Windows lacks and will probably be lacking for a long time to come.

But that is the end of the advantages. The rest of the installation requires a revamp. The partitioning sub-program is very bad compared to even XP's. Unless you let the installation program do the partitioning as it pleases there is no basic and obvious way of doing a manual partitioning. If you do, the swap partition may become unrecognizable or it may partition other areas that you already have files on. Please note that I am stating these facts from the point of view of a person switching from Windows. I am sure the Unix experts will want to disagree here since they know a lot about manually partitioning the

drives the arduous ways they expect everyone else to know or learn them too.

The rest of the installation is pretty straight forward up to the point of connecting to the Internet to download the updates. Here there are two major pitfalls.

1. The firewall in both Debian and Ubuntu is disabled by default. That is a very serious security issue. While in Ubuntu the user can unplug the Internet during the Installation phase and before reconnecting can enable the firewall by a rather simple command, in Debian it is a serious pain to do so. Add to it the fact that in Debian the firewall, even when enabled, defaults to allow everything in and everything out. What kind of protection is that? Windows has a primitive and unsatisfactory firewall included that is on by default. It is no match for Unix's IPTables but how many average users can we name that can comfortably and reliably work with IPTables? While on this subject, there is no support in either distribution's firewall to prevent Internet access at an application level unless you are a professional with IPTables. A very good example of a small and effective Windows based firewall that was eventually swallowed by a bigger company and afterwards destroyed by it is Tiny Personal Firewall version 2. I still have that free version installed on my Windows partition and it belittles even the most modern free and costly firewall. It is highly configurable, almost any user can make any rule they want not by necessarily manually

making the rules but by responding to the firewall's pop-ups and making them permanent if they so wish. It intercepts every application's attempt at connecting to the Internet and asks for your permission, even if it is your word processor. If you upgrade an application, upon launch it compares the MD5 signature of that application to the previous one and asks for permission to let the new upgraded application connect. Everyone will agree that this is a very good security precaution because what if an intruder installs a different application than the one you have originally installed and approved for connection? You will hardly notice that, the only way would be by some means like checking the MD5 that Tiny Firewall does on the fly. The user need not worry or do some heavy duty coding at the terminal to achieve that. Is it really hard for us to implement a firewall like that in Unix? There are tons of comments and literature in the Internet about IPTables bragging about its security and comprehensiveness. What is the use of all that if most can not even use it? It is like having a Ferrari in your garage that is completely disassembled. You can brag all you want about it but if you want to drive it, you either need to take a long auto assembly course at the Ferrari plant or have a team of professionals assemble it for you. Even if you choose the second option, what will you do in case of a minor accident? Since you have no idea about the assembly, you have to ask the same team of professionals to do it for you again. A minor accident in operating system terms means a simple reinstall. Wouldn't you be better off not having

- that kind of a Ferrari but instead having a Chevy Vega that simply runs as expected?
2. The other problem with the installation is the security updates. This is not really an installation only issue. It is an issue even after the installation. Most of us know that for an average unpatched computer to get infected by connecting to the Internet takes less than 12 minutes. In Windows the fix is pretty easy. You can download and save all the security updates while using a fully patched PC. Then on any new installation you can apply all the updates before connecting to the Internet. Where is that capability in Unix? Is that very hard to do? Apart from being a serious security issue, it takes a lot of bandwidth and Internet resources to connect and download all the updates every time you install Unix on a PC or on more than one machine if you have multiple Unix PCs. Whereas if you have downloaded the update files separately you can update as many PCs that you want without the need to connect every PC to the Internet and let it download the same updates again and again. Ubuntu has one of the fastest paced upgrade mechanisms in the Unix world but even that takes six months. Who can afford not to update their machine for six months? Some may say that Unix worms and security breaches are far less common. That is a very good thing but it is hardly an answer to this argument. If they are so uncommon why bother with any security update?

b) Installation of Application Programs

This is a very thorny issue. It is perhaps the worst thing about Unix. We are all used to the concept of repositories. But that is a horribly outdated concept. This is one area that needs serious redesigning. This is where we should really learn from Windows, unfortunately. Please do not be biased. Even the worst thing that you may hate can teach you something because nobody and nothing is perfect. In Windows all a user has to do is download any application that he desires and run its installation program. A user need not know anything about what the installation needs to do in order to make the application functional. If there are dependencies, which I am sure there are, they are all taken care of by the single installation file that the user has downloaded and run. That is it. After the installation program is finished you can safely start running your application. Try doing that in Unix! Unless one uses Synaptic or some variation thereof there is no way that you can install and run your downloaded application peacefully and easily. THIS IS A MAJOR PITFALL IN Unix. We may say, well that is the way Unix is! But that is not the right way and if we keep insisting then no one will be migrating to Unix from Windows. Remember Shakespeare? We are insisting that the user uses our applications in a vacuumed room hung from a rope. Not too many people are willing to do that and we will remain just a curiosity to them despite our great technology. The same thing applies to upgrades. For instance Firefox is included in Ubuntu. What happens when a new version of Firefox comes out? Canonical says that they will test the new version in their good time and release it through their update manager. I am sorry but that is not good enough. In

the age of certificate hacking and so many other on-line security breaches and threats, that is simply not a good answer. Now let us be honest and see it through the eyes of a simple Unix user with only elementary skills. All the newly released versions of Firefox are available at its site, Unix versions included. Can we simply download the latest version, check its signature and run it to install over our old version? No. Why should we not be able to? This same argument is valid for any other software installed even using Synaptic and the repositories. Most of them are outdated in Ubuntu's or Debian's repositories and if you want to upgrade to a newer version from the developer's site you have to jump through hoops to do it. Another example is Pidgin, an excellent open source IM software. Try to upgrade Pidgin on Linux to a new version. Now do the same in Windows. It is a snap. We don't even let Firefox or Chromium upgrade themselves once installed. Is making an executable installation program under Unix that difficult? Why do we have to deal with tar and gz packages that take us nowhere all the time? Yet the same program is available to Windows users with a double click. I can write a lot more about this but to the very intelligent Unix community this much suffices. Let me just say this: It is an excellent idea to be a good and able car mechanic. But if a car manufacturer requires anyone who wants to drive its cars to be an able mechanic, how many people will buy its cars let alone abandon their current cars and switch to those even if they are free? Would that company survive among others who just demand a basic driving ability? It is true that the company gives away its cars but that does not mean that there is no hard work behind the cars it makes. Doesn't

that approach make the hard work it has done seem valueless and under-appreciated?

No one can deny that in order to use a computer, one needs to learn to use it. But requiring people to become an operating system professional in order to use a computer is just like that mechanic requirement from the car company.

c) The GUI

Here we are up against the wall again. There is Gnome, Unity, KDE and Xfce to name a few. Each has a few good features and a few bad ones. Why can't we unify these by combining their advantages and eliminating their pitfalls? We can make a GUI like Gnome the default with the ability to painlessly switch to others if a user so wishes. A good example is installing Gnome in Ubuntu 11.10. It took me a while to figure it out but once done I can switch to Gnome or Unity with a simple log out. But usually there are different distributions for each GUI. There are Kubuntu for KDE, Xubuntu for Xfce and Ubuntu for Unity. Debian uses the same kind of approach although not with so many names and they are all part of the same distribution which is far better than the Ubuntu variants, which are each managed by a separate group. The advantage of Ubuntu here is that you can have an Xfce GUI installed on top of the current installation whereas in Debian you have to make up your mind and download what you want in advance, at least that is my understanding. I have KDE, Xfce, Unity and Gnome all installed on my Ubuntu 11.10 for testing purposes and they all work fine. But that approach is only good for testing. It adds a lot of overhead

to the system. Furthermore, I have no idea how to safely remove KDE, for instance, if I see that I don't like to use it any more in the future without breaking the system or at the very least without compromising the security and integrity of my system.

This hurts the popularity of Unix a lot. We are still experimenting here. Regular users don't like our experimentations. They should be given a good looking and capable GUI and if they so desire, the chance to experiment with us. If we can, we should give them all and let them switch between them with a small overhead. Ubuntu has been in such a hurry to bring about Unity that all its help pages still refer to Gnome and the user has no clue where the same feature is in Unity. For instance try to delete your recently opened documents from within Unity. Unity is a totally immature GUI that is being forced on the users of this popular Unix system just because someone at Canonical liked it. We don't need yet another GUI to get used to, we need a single good GUI and later when all the other major issues with Unix's usability are dealt with we can start testing new ones. We all know that old habits, or new ones for that matter, die hard. Once someone is used to Gnome for instance, it is pretty hard for them to switch to Xfce just because we are introducing a new version.

That discussion aside, there are a lot of features missing from the GUI. Here I am mostly referring to Ubuntu. Take the search engine for example. It is very slow and in its basic form lacks the ability to find a file based on what it contains. For that you need to download another package. Is there anything more basic than a search command?

There are no options in the GUI to control the default behavior of the search. An important feature is missing here. It is the ability to right click on a file found with search and to ask the OS to open its containing directory. This is a feature that has been available in Windows for years now. At present, once you find a file you have to manually navigate to its directory in the GUI.

In Xfce you can hover your mouse on an icon and it tells you a summary of its properties, in Gnome there is no such thing. In Xfce setting an automatic desktop background is fairly easy; in Unity there is no visible way to do it. There are so many of these little advantages and disadvantages in all these major GUIs that I can not mention them all here. And once again we are so unkind about the great features we have compared to Windows. Take multiple workspaces for instance. Unity excels here. One can just click on the Workspace Switcher and it immediately brings up the four default workspaces and you can drag and drop between all four. Try to do that with Gnome or Xfce. I am sure you will be frustrated.

A secure operating system like Unix should not keep track of all the commands the user enters at the terminal or all the documents the user has ever accessed. At the very least, it should make deleting all the traces as easy as clearing a browser's cache. I have seen Ubuntu's terminal remember commands I used many days before even after rebooting the machine many times, very secure indeed! Deleting recently accessed documents is also a pain. Yet both Ubuntu and Debian have the excellent feature to encrypt your entire drive at installation. While Ubuntu lets you decrypt your volume with the login password, Debian

requires you to enter a separate password for that. I have no idea why they should be different. While this excellent feature is there, there is no ability to securely delete a file or directory or free disk space through the GUI. You have to be a Unix guru to be able to do that. This is a must in any secure operating system. The secure delete should be present in the file sub-menu. We are not running DOS and this is not the 1980's.

One may argue that there are lots of Windows commands that can be run only through a command prompt. But let us remember that the average user does not need to bother with most of them in order to be productive. In Unix the opposite is true. If you want to be productive you have to be dealing with the terminal and its cryptic commands. What adds insult to injury is that a lot of Unix users tell you that this is the way to go and it is the best approach. I am not against the terminal or the command prompt, I started working with computers in the 70's, but today's user should be isolated as much as possible from entering commands. The Unix commands are sometimes so non-intuitive that I find using a native Unix program like gnupg easier in Windows than in Unix. Isn't that sad? Let us again remember that Apple is run by Unix and it is the favorite of many GUI loving people. If Steve Jobs could do that why can't all of us together do the same and make it even better?

The GUI needs a lot of more improvements and I welcome all users and developers to actively take part in it. What I

said above is not a wish list, it is a small basis that is absolutely necessary for Unix to be really popular.

d) The Ability to Keep and Reinstall Applications

Here as in the case of security updates the user should be able to download the application that they like and keep it in their backup for future re-installation. This is a common situation if a user wants to reinstall the OS for any reason. The way it is now, he or she will have to remember or write down exactly what he or she has installed and upon installation of the new OS download the whole thing all over again. Some of these applications are quite large and take hours to download on some slow Internet connection. I have seen that there are some efforts made to this end but they are by no means adequate. The problem again relates to the same dilemma faced when installing any application using an installation file as discussed before.

e) Simplicity of Performing Tasks

This subject might have been dealt with in the GUI section but it is so important that it deserves its own heading.

There are so many tasks that need to be done, some on a daily basis, but the system refuses to let you do them easily. One example of that is checking the file system. If one needs to check the integrity of the file system, it can not be done with the system on because the file system is mounted. You have to resort to a boot CD to check the file system. Can't we make an option like XP to ask to check the file system on the next boot? I am sure we can. But this

is one of the other oversights that we have with Unix. The reason clearly has to do with the paradigms mentioned before. Unix was developed by highly technical people who were very comfortable at the command prompt and since they were comfortable themselves, they would not see the need for the others. Of course lots of other things have been greatly improved but many still need improvement. Who is not aware that until a few years ago mounting a drive had to be done exclusively through commands? But today, any user can just click on a device and the system automatically mounts it for them. That is a giant step forward and we should be proud of that, but there are still lots of stuff that need the same simple approach.

f) Routing and Internet Sharing

This is a very thorny issue too. The days of one computer per household are long gone. But still performing a simple Internet sharing connection that is properly firewalled and secure is very out of reach. Almost any user can setup an adequate Internet sharing scheme using Windows XP but try doing that with Unix. Please look at this from a user point of view. Please don't say in your head that, "here, it is easy. Just type these millions of commands and you are simply up and running!!" I have come across many articles on IP masquerading, routing and so on and none is simple to perform. The simple ones are all insecure. The user invariably needs to deal with IPTables and enter many commands and at the end not be sure if he is connected, let alone securely.

g) Hardware and Device Drivers

Unix has come a very long way in this area. Here too what has been done so far is commendable. Ubuntu and Debian

can detect and work with a lot of hardware without the need for the user to resort to downloading a proprietary device driver. In fact I think Unix works much better than Windows as far as the device drivers are concerned. But if a user wants to upgrade or download a different driver for his hardware from the manufacturer's web site, he is really out of luck. The same hurdles that he faces when upgrading his favorite browser or other application show their ugly face here too. Some basic devices such as floppy disks don't work properly and many basic modems refuse to work altogether. Why? Simple modems and floppy drives are two of the oldest pieces of hardware that were absolutely essential in the early days of Unix. Why don't they work properly now? There have been no major improvements in either category but still they don't work as expected.

h) The Help System

Help in Unix is very hard to come by without the Internet and user forums. There should be a lot more off-line built in help in Unix. I have seen so many confused users at Unix forums that are wondering how to do some very basic things with their systems. These can be easily be distributed within the installation media and be installed so that users can find most of their answers off-line and only go to forums for more complicated tasks. Ubuntu's help system is particularly weak.

i) Source Compilation

Here too Unix has a definite advantage. One can compile a running program from source. That is hardly a question for Windows to handle efficiently. But again the task is so hard that many people don't even know about it and many that

do can not figure out how to do it and will let it go altogether. Here again we are obscuring a very good advantage of Unix and only keep it as a secret for the elite.

j) Standardization

Let's face it. The boom of the computer industry could not have happened if it were not for standardization. Standardization has taken place in both hardware and software and to a pretty good extent. Many will remember the earlier days of personal computers. Back then the only standard was the keyboard! The only reason for that was the fact that typewriters had already standardized it for us as we still use the QWERTY keyboards. But everything else was non-standard including Disk controllers, memory, other I/O, etc. Do we even want to remember the SCSI and pre-IDE battles? Do we even care any more about CGA, VGA, XGA and other graphics technologies these days? But the hardware industry finally realized that they need to standardize their offerings and they did. Combined with the open architecture of the IBM PC, that standardization alone caused the computer usage to skyrocket. The same has happened in software. For instance, we all use Ctrl-F for find and Ctrl-C to copy. It does not matter if we are in a browser, a word processor, a spreadsheet or just inside the OS itself. Every application knows that Ctrl-F means find. It has not always been that way. I remember the time when each application had its own copy shortcut. But the new standard way is the way to go. Unix needs to be standardized. Why should some distribution use a weird command like Yum(!) while another uses something else? Here we can move a lot faster by standardizing our

commands because since we are all open source, there will not be any senseless legal battle over the use of the same command syntax. Why can't we do it? I believe this is the subject of the section The Distributions.

k) Networking

We all now that Unix is the king in networking. After all networking was built-in from the beginning whereas other operating systems most notably DOS and Windows were all stand alone and networking capability was introduced in them much later. But just because we have taken it for granted, we have fallen behind in this key advantage of Unix and have almost relinquished it to Windows. As mentioned before we can connect to a Windows share much more easily from Unix than to connect to a Unix share. We need to install Samba in Ubuntu just to be able to share a folder using Microsoft's SMB. I can access a Windows share from Ubuntu on a Windows machine without Samba but sharing a Unix folder for another Unix machine requires Microsoft's SMB (read Samba)?! Whereas before, Microsoft had to make provisions in its operating systems to accommodate Unix, we are now placed in a position to be compliant with Windows networking. That is very sad. I can cite more examples in this area such as mail servers and TCP/IP implementations that were inherent in Unix and are now being taken more seriously by Microsoft despite Unix's clear superiority.

4. The Distributions

The last section is by no means the least important. It is the tremendous number of Unix distributions that exist today.

Do not get me wrong. I am not in favor of forcibly abolishing all and make them all one. But I am afraid that the Unix community should get a sense of what it is doing and unite in the face of the proprietary software. The proprietary software is not only limited to Windows, there are many proprietary Unix flavors there too. How many of us remember the many efforts by IBM, HP and Sun to compromise and make a unified Unix OS in front of Windows? All those efforts failed. They failed because the chief force behind all those companies are profit making. None of them wanted to make a concession because they did not want to lose any of their base or profits, but they eventually lost big to Microsoft. How many HP PCs are running HP-UX today? How many are running Windows? Where is IBM PC and OS2, its operating system business? What has happened to Sun? Where they failed we can succeed. The only reason is that we are not driven by profits. The only thing that may get hurt, in for instance Gentoo adapting a Debian standard, is their pride. But I am sure that there is much to be gained. A huge user base and the internal satisfaction that we, Debian, Centos, Ubuntu, FreeBSD, and all others contribute to a massively successful operating system that is as powerful as Unix and as easy to operate as a TV. Let us put our differences aside and instead work all together to achieve that goal. Let us find an alternative to this disease all over the Internet called Adobe's Flash Player instead of making our own version of the same familiar install command just to state that we are different! I know that HTML5 may one day make flash obsolete, but while we wait for that can't we have our own bug free, open source version? Hardware has for many

years been way ahead of the software. We still have a very large ground to cover that gap. We still can't fully and properly take advantage of multiple CPU cores and gigabytes of RAM and 64 bit computing power. Let us focus on these issues instead of competing with each other uselessly while the profit driven competition enjoys our quarrel. I thought that Unix would be far more superior than Windows in 64 bit computing but the truth is that they are both the same. There is no performance advantage between the 32 bit and 64 bit versions of either OS apart from the handling of more RAM. My 64 bit Ubuntu 11.10 with its own dedicated hard drive is no faster than 32 bit Windows XP on the same machine! Let us ask ourselves this: Do we have to be proprietary to work? Consider this scenario. A person is employed in a company as a software developer. He works and gets his salary. The company makes a fortune off his work. Now that same person leaves the company and decides to work on his own. That way he can reap the profit of his own work. But why can't we see many of these become successful as entrepreneurs? Because being an entrepreneur is hard! You have to have strong discipline and make lots of sacrifices. The Unix community has a very strong discipline and make lots of sacrifices, the most important of which is working for free, but still can't succeed. Let us do succeed. We have all the right people and the right technology to do so. We just need to shift our paradigm.

Let's face it, how are we supposed to grow the Unix user base? Are we supposed to grow by making new users choose Unix? No. There is no significant number of new

computer users every year. There are only the younger ones and children who start using the computers they have at home and at school. So here too, the operating systems that are running on those computers will determine what the next generation of users will be using. That is why the only way we can grow is by current users switching operating systems. For that we need a very solid and convincing offering and what we have so far is not it. We need to put the user in control of the computer not the other way around and Unix is the only operating system that can do that but as I have tried to show, it does not yet empower users to be in control and it keeps its inner workings a mystery for the average user.

Let us unleash the power of Unix and let the users see how powerful Unix really is. Then just sit and enjoy the charts shoot up and the number of Unix users pass the number of Windows users in a very short time. Why are we stuck at the mere 2% of total users? I believe we can move all the way to 40% in a very short time if we take care of our own internal problems. Let us firmly believe that for the great mission that we have chosen which is making a free, secure and capable operating system and free reliable applications of all kinds, we need to be united. United we stand, divided we fall.