

Operating Systems

Andy Pepperdine

You may have heard the term Operating System (OS) and wondered what it is and why you need one. This paper attempts to answer that question.

The opinions expressed here are mine, and you may quarrel with all, some, or none.

History

The first computers could do only one thing at once. They might be computing accounts, payrolls or some scientific calculation, but they could not do these things simultaneously. But when the electronics got fast enough, it was soon realised that there was a valuable resource going to waste. The machine was idle when data was being read from or written to some external medium.

The answer was to add a layer of program on top of the applications which would start and stop other programs, and switch between them to keep the central processing unit busy. This ability to do more than one thing at a time was crucial to the early commercial success of computers, and it the thing that enabled that to happen was called the Operating System.

As technical advances occurred, the OS had more and more to do. There were more resources that could be conserved or protected or re-allocated, like memory, disk drives, networks, CD drives, displays, wireless connections, sound input and output, etc. And the OS got more and more complex in order to handle these disparate items.

Not all OSes do the same thing, and different features were demanded by different parts of the computing world. Some machines do little more than one thing at a time, but do need to bring up alternatives rapidly, like your use of your netbook, or desktop, for e-mail, browsing and document preparation. Others are used to serve up web pages, and they may indeed have to serve several streams at once, and be able to cope with adding new pages at the same time as looking up and serving existing ones. There are several things going on at once in these machines. Another common area these days is in a mobile phone, where the machine must keep connected as you move around and be able to wait for calls as well as being able to allow you to type a text message or look up something on the web. Then there is the largest area of all, so-called embedded systems, in cars, machinery, power plants, wristwatches, aircraft, digital radios and TVs, etc. which must be able to react in a determined amount of time to an external prompt – they cannot wait for the brake fluid pressure to finish being measured before applying the brakes when you depress the pedal. So different OSes were devised for different purposes.

Only the use you make of a machine to do “normal” computing work will be looked at here, with some pros and cons mentioned for each available system.

Unix

During the 1960s, theoretical computer science, and engineering experience, together progressed rapidly, producing a large amount of information and knowledge with regard to the important aspects of what an OS should incorporate, and how to keep different accounts from interfering with one another. This culminated in a very sophisticated, and ultimately overly complex, system known as MULTICS built by Honeywell. IBM also had lots of experience in this and built several different

systems all using the same well-understood theory as a base.

Three engineers at the Bell Labs of AT&T in 1969 were looking for a use for a small surplus machine, and decided to write what they called, first Unics then Unix, a small OS which distilled the known wisdom of the time into a few clearly understood principles, and simple security model to prevent accounts and programs from getting in one another's way. They also enunciated a method of allowing programs to co-operate, and so started the idea that a program should do one job only, but do it well. This was the basis of the shell scripting technique and the supporting functions. It spread widely due to it being distributed free of charge, with source code, to many universities, and became soon recognised as an excellent basis for any number of applications and experiments.

The simple idea of using one process for one task became a mantra of good software design.

But in those days, the notion that networks would become essential to all general purpose computers was not anticipated. That was some way in the future.

But there was an important side effect of the way AT&T distributed their work, in that it became well known among software and hardware engineers, so much so that eventually a standard was created to define some of the applications that were considered part of "Unix", and also the way an application interacted with the OS to get its requirements met. This became known as the POSIX standard, and is instrumental in allowing many competing entities to contribute to a central common set of work. The first version was issued in 1988, and was known as IEEE Std 1003.1. It then went through several later iterations, and is still being worked on.

However, there was also an important negative side effect in that there was no common code-base for the implementation of Unix, and competing companies tried to outdo one another with the result that the POSIX standard arrived too late to save Unix from a fragmented marketplace and incompatible applications across systems. It made it too difficult in the end for independent developers of applications to support all the different versions of Unix on offer.

The Personal Computer

During the 1970s, several hardware manufacturers started to experiment with small, but expensive, machines for single person use, or small kits were available at a more reasonable price for hobby engineers to build their own machines for playing with.

Then, at the end of the decade, IBM had the idea of trying to get a small computer onto every desk in every home, as well as the standard office uses that were becoming commonplace. In order to make it as widely available as possible, they contracted out the central processor to Intel, and the OS to Microsoft. The specifications did not consider that these machines would become networked, let alone have access to innumerable servers around the world. The result was that it was not considered important at the time to have the ability to separate programs and accounts, and to have good security aspects, even though the existence and success of Unix showed it was a very useful attribute to bear in mind.

But the theoretical computer science was ignored, since MS produced first DOS and Windows, which assumed that the machine would be isolated from others, and used by a single person. Consequently, they thought it had no real need of any security theory. However, IBM recognised the problem, and attempted to counter MS's strategy with their own OS/2 system but failed to prevail in the marketplace which became dominated by Microsoft.

Windows

The most common OS used on a home PC is a version of Windows from Microsoft. It is, in the UK, very difficult to buy an Intel based laptop that does not come with Windows pre-loaded. This is what you get when you buy a “PC” from a high street shop.

Because it is the most common, it is the first target for any commercial organisation when they have an idea for a new piece of software – or when they have a new piece of hardware, like a printer.

But, because of its historical origins, it does not have a sensible security model underlying the OS, and has been vulnerable to many attempts to hijack the OS for malicious purposes. The problem has been compounded by the defaults set in the normal applications, and the desire to make things “easy” for users to do what they think they want.

If you want a system that can most easily interact with other systems, then choose Windows. There is also a wealth of professional support if things go wrong, and tuition when starting computing from scratch.

The downside is that it is vulnerable to attack by viruses, etc., which means that you have to spend extra effort on protecting yourself, either by learning how anti-virus programs work, or paying for frequent updates to them. The OS is supplied by only one corporation, and so if anything should go awry with MS, then you may be stuck with programs you can no longer use.

Like all companies, MS are interested in ensuring that you, as a customer, will continue to want to upgrade when new features are released. This places you on a treadmill that MS controls the speed of, and whatever version you use will at some point become unsupported. Admittedly, this is not as serious as might initially appear as computer hardware is also being continually updated and more functions can be implemented which older computers cannot manage, making upgrading necessary due to hardware improvements, not just software.

Apple OS/X

Apple Corporation sell machines with their own OS on them. They give no option. They have concentrated on graphical design work and good user interfaces, and are the preferred machines for the creative industries.

They are a large corporation, and makers of new hardware will usually supply drivers for their devices for Apple machines, but not always.

From the beginning, Apple have based their OS offering on a version of Unix, and do not have the same issues of security as Microsoft do. They have taken this aspect seriously enough to reduce the problems of viruses to a negligible amount, although you should always be careful of what you yourself install on any machine you own.

If you want a no hassle, solid computer for home work, then Apple would be a good choice.

The downside is that they are a proprietary organisation, and can change things at will as they produce new products, in the same way that Microsoft can and does.

Apple iOS

When Apple branched out from desktops and laptops to devices with touch sensitive screens and no

normal keyboard. They adapted their Unix-based offering for phones for this purpose, but they also carried over from their phone offering the strict control over the type of application they will allow to be placed on the computer. Thus, although the product does what most people want from a mobile device, there are limits on its use that Apple have imposed. Again, you get no choice of OS from Apple when you buy one of their products.

Apple do not like the way Adobe's Flash operates, and so there is no Apple support for Flash from Adobe, which makes some websites inaccessible from an iPad. When elephants fight, the mice get trampled on.

Linux

Linux is part of the free software movement, and is successfully supporting a vast array of different types of applications and machines. It has a very strong presence among web servers, and almost a complete monopoly among the most powerful machines in the world. It is now the most flexible OS ever made, and there are versions for more types of processor than for any other OS at any time in history.

They can do this because they implement as strictly as possible standards where they are defined. It started as supporting the POSIX interface, but has long since expanded beyond that. Computer manufacturers like it because it is so adaptable, and there is so much code that they can use without paying anything for it. They must contribute their work back to everyone else, but typically that happens only after they have established their product in the market and got an edge over their competitors in so doing.

Since it is based on the Unix experience, they have little difficulty with security issues, and in fact places where security really matters (like web servers) have taken to using Linux in large numbers and it is the dominant OS in many areas of computing now.

The advantages for a home user are the use of standards for file formats, etc., which means that work done now will still be available for the indefinite future.

Because all the code is accessible to everyone, and there are incentives (via the General Public License, GPL) to keep the code-base consistent, the chances of creating alternative implementations of Linux to compete with one another are greatly reduced. But that applies to the kernel, and the coding, it does not apply to a distribution of a combination of applications and the kernel which are made by different people for different purposes and different ideas of what the customer wants. It has produced a plethora of examples of what are loosely called "Linux", and dozens for the home user to choose from.

As well as producing everything in some form of recognised standard to ensure long term stability of data, it also attracts innumerable freely available applications to be downloaded and used. It is not beholden to any commercial interest and so it will always be available for someone to take up should the current engineers stop working on it.

The disadvantage, however, is that Linux on the home desktop is a tiny proportion of machines, which means it is not regularly supported by new device makers, especially not in their first implementations. Linux tends to catch up later.

A more important problem in some quarters is the lack of patent access, since most of the work is done by individuals or small teams and the resulting code is freely given away. Currently, the UK does not allow software to be patented, but there are moves afoot in Europe to change that. The

legal minefield is a possible concern, and needs monitoring.

BSD

Berkeley System Distribution (BSD) is also popular among web servers and back room activities. It is derived from the original Unix developments from AT&T and the Berkeley campus of the University of California. It can be used by anyone freely, and the license does not prevent anyone from modifying it and then selling the result, in contrast to the GPL of Linux. Apple took BSD and changed it into their own OS/X, for instance.

There are versions of BSD for the home desktop, but it requires rather more knowledge than the typical user has to get the best use out of them. There is also a danger they can drift apart over time and fall into the market fragmentation problem that Unix suffered from.

However, since there is so much in common at the OS interface to applications, there are many applications available on Linux that are also usable on BSD.

The problem of patents associated with Linux also applies to BSD when used in the home. Commercial producers of BSD can make their own licensing arrangements.

Android

This is a recent development from Google and is rightly applied to mobile phones, although it is also suitable for some of the new tablets. The underlying OS is based on Linux, but the upper levels of applications have to be re-implemented as the interface to system features has changed radically. So long as it is used for mobile devices like phones, this is not a real problem, but may become one if computer makers try to use Android on desktop or laptop machines. But, like Apple's use of iOS on touch devices, Android is also suitable for such applications.

We can expect more of Android as it develops. I see the main danger is one of fragmentation similar to the way that Unix failed to capitalise on its success. If each phone maker insists on their own application store, then it will make it too difficult for a developer of an application to make them available to everyone. Will they learn from history and create an official standard now?

Chrome OS

This is a new idea from Google, and is a restricted OS that starts up a browser and everything is done within the browser. It is thus suitable only for computers to interact with data that is held on a cloud somewhere. The references point to a review of a week doing normal work from a ChromeOS netbook, and one thing to beware of is access to other bits of hardware, like printers.

References

Lévénez, É., *Unix History*, <http://www.levenez.com/unix/> (Diagrams and references on Unix history)

The Open Group, *History and Timeline*, http://www.unix.org/what_is_unix/history_timeline.html (More information about early Unix)

Blinkenlights Archaeological Institute, *What was the first personal computer?*, <http://www.blinkenlights.com/pc.shtml> (A list of small machines prior to the IBM PC)

Wikipedia, *History of Microsoft Windows*,

http://en.wikipedia.org/wiki/History_of_Microsoft_Windows (A more complete version of early Windows than Microsoft's own site)

Beavis, G., *A complete history of Android*, <http://www.techradar.com/news/phone-and-communications/mobile-phones/a-complete-history-of-android-470327> (Sept 2008)

German, K., *A brief history of Android phones*, http://reviews.cnet.com/8301-19736_7-20016542-251.html (List of phones containing Android as of May 2011)

Vaughan-Nichols, S.J., *7 days in the cloud: My week with the Samsung Chromebook*, <http://www.itworld.com/print/183507> (Review of Chrome OS)