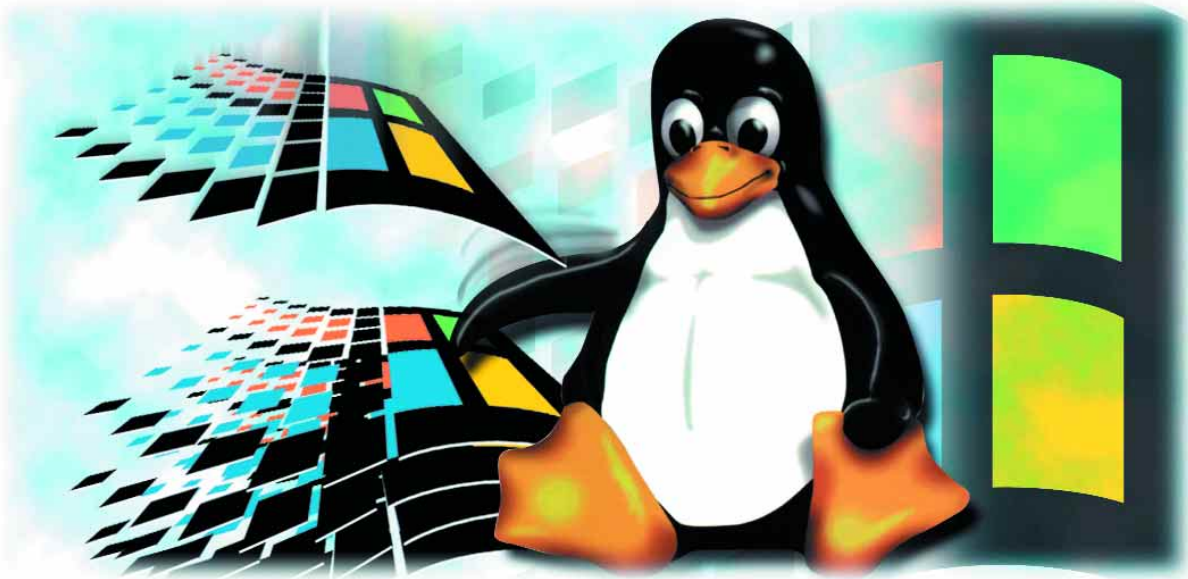


Running Windows applications under Linux

PUTTING WINDOWS IN ITS PLACE

JULIAN MOSS

The need to continue to run key Windows applications for which there are no suitable alternatives can often be an obstacle that prevents the adoption of Linux from being considered. In this article we look at three possible solutions



It's unfortunate, but true, that the success of an operating system depends not on its own excellence but on the quality and quantity of the applications that run on it. This is one reason why Microsoft Windows dominates the PC market despite its flaws.

New users choosing a computing platform for the first time will tend to choose Windows because of the wide range of applications that are available for it. Once they have made that decision it becomes hard for them to change because the applications they are using lock them in to the original platform.

Any benefits that could be gained by switching to a superior operating system such as Linux must

be offset against the money and time that will have to be spent converting to alternative applications. In some cases there may be no obvious alternatives available. Therefore many companies or individuals who wish to adopt Linux will only be able to do so if a way is found to run these legacy Windows applications for as long as they are needed.

Fortunately, as is almost always the case in the Linux world, where there is a problem there is a solution. In fact, as is also often the case, there is more than one solution. In this article we will look at three ways in which users may run - or appear to run - Windows applications on their Linux worksta-

tion. These three methods may not be the only way to integrate Windows systems with Linux but they are commonly used, inexpensive and capable of working well.

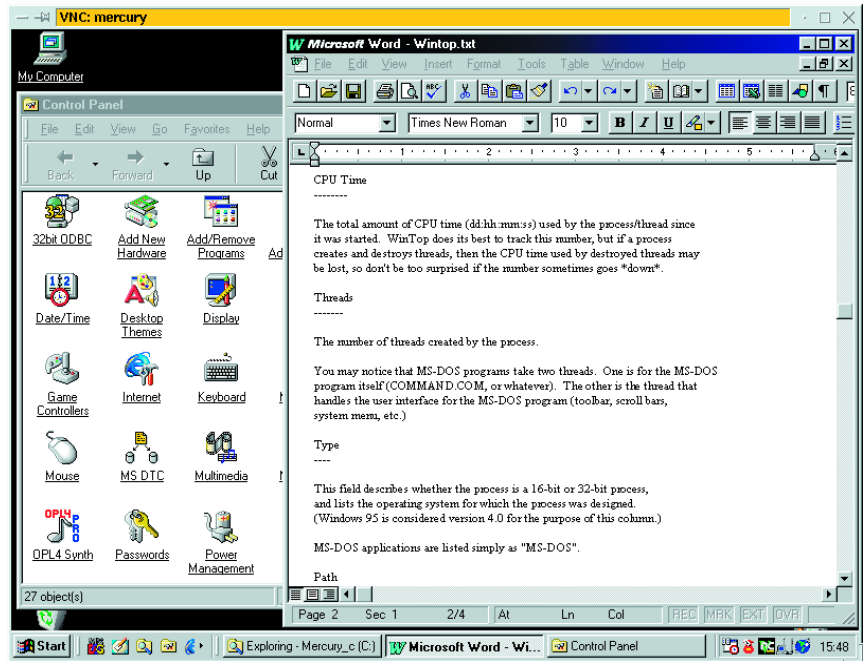
Virtual Network Computing

Virtual Network Computing, usually known by its acronym VNC, is a technology that allows you to view the desktop of one computer system on another. Readers with Windows experience may have used similar technology in remote access products like PC Anywhere. It is the easiest of the solutions described here to set up, but the most expensive in terms of hardware, because the Windows software must run under its own copy of Windows on its own computer, just as it did before [Fig. 1] right.

VNC is a simple protocol whose implementation consists of two components: a server and a viewer. The server runs on the computer whose desktop you want to be able to access remotely. You use the viewer on a different computer to view it. Currently there are implementations of both servers and viewers for Unix (Solaris, DEC Alpha), Linux, Windows (32-bit) and Macintosh (PPC and M64K.) There is also a beta version of a viewer that runs under Windows CE and a Java viewer that will run on any platform that supports Java. VNC is free and is released under the GNU General Public License so you can download the source code as well as binaries from the developer's web site.

VNC's developers originally foresaw VNC being used to allow users to access Unix applications running on a big Unix system in the computer room from the PCs on their desks. However, with the growing popularity of Linux many people are now using VNC to view Windows screens from their Linux desktops. Unlike some proprietary remote access products the VNC software is small and needs no complicated installation. The Windows version comes with a setup program which installs the server in the Registry so that it starts automatically whenever you start the PC. For Linux there is a pre-compiled binary version that will run under any distribution that uses glibc.

First, you need to make sure that the VNC server is running on the remote PC. Then, to view its desktop on your screen you simply start the viewer, enter the remote system's name or IP address and its access password. Once the desktop is visible you can interact with it as if you were actually sitting at that computer. You can copy and paste between Linux and Windows. If you need to share files between Windows and Linux you can enable file sharing on the Windows box and use `smbmount` to mount the Windows drives from Linux. You have full control over the Windows computer, including the ability to shut down or reboot it, so it is perfectly possible for the Windows box to run without its own keyboard, mouse and monitor.



VNC allows you to view the remote desktop on a system on which the VNC viewer has not been installed. Launch Netscape, enter the address `http://remote:5800/` (where *remote* is the name or IP address of the remote computer) and, after the password has been entered, the Windows desktop will appear in the web browser. This is possible because each VNC server is also a web server that accepts connections on port 5800 which serves the Java viewer to any web browser that connects to it.

The VNC server uses a fair bit of processor time on the host machine and response times can be a little bit sluggish even over a 10MB/s Ethernet connection with no other traffic. You may experience problems with parts of Windows not being updated in the viewer display: in that case there are settings which can be changed on the VNC server. VNC may not be the most efficient way to provide access to Windows programs when you're running Linux but it provides a method that is very easy to set up and use.

Using VMWare

If you want a solution to the problem of running Windows applications from Linux that doesn't require a separate computer to run Windows you could try using VMWare. As the name implies, VMWare creates a virtual machine under which you can run another operating system. You can get VMWare for Linux which allows you to run Windows under Linux, or you can use VMWare for Windows NT and Windows 2000 which enables you to run Linux under Windows. The latter solution may be attractive if you are dipping a tentative toe into Linux waters and would like to be able to switch between operating systems without the inconvenience of dual-booting. VMWare is a commercial product and costs £199 US for a licence, or £66 for home use, but it is available for download and 30-day evaluation [Fig. 2].

Fig 1: VNC puts a Windows desktop in a window on your Linux desktop

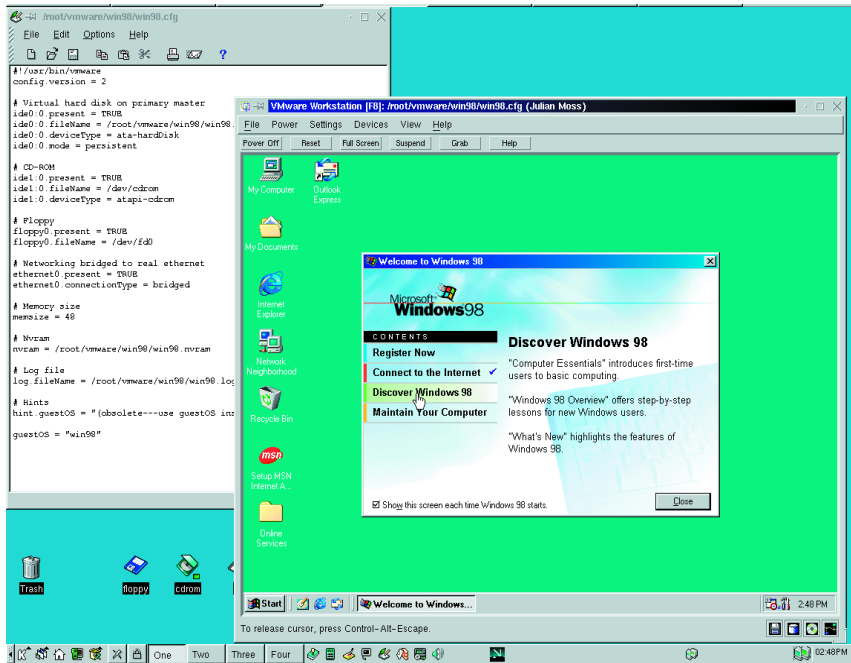


Fig 2: VMWare lets you run Windows under Linux in a virtual machine

You'll need a reasonably powerful PC to use VMWare, since in effect it will be running two operating systems at once. VMWare recommends a system with a minimum 400MHz processor and 128MB of RAM for one virtual machine with applications active on both the guest and the host. We'd go along with that. The product is supported under recent versions of Red Hat, Caldera OpenLinux, TurboLinux and SuSE Linux and requires XFree86 version 3.3.4 or higher. It will run on other distributions: for the conditions check VMWare's web site.

Setting up a Windows guest operating system on a Linux host involves the following steps: installing VMWare, configuring a virtual machine using the VMWare Configuration Wizard; installing Windows and installing VMWare Tools for Windows. The process is very straightforward and takes about a couple of hours, most of which is spent installing Windows.

The virtual machine is created in its own directory under the Linux file system, along with a virtual hard disk which is created in a file under the Linux file system. You specify its size when you run the Configuration Wizard, and it must be partitioned and formatted before you can install Windows.

The operating system running in the virtual machine cannot see the real hard disk, so even if a Windows virus activates it can't possibly harm your Linux system. In fact, anti-virus companies use VMWare in their labs to see what happens when viruses activate. To restore a clean system they simply restore a copy of the virtual machine from a backup.

VMWare allows you to install Windows 95, Windows 98, Windows NT 4.0 or Windows 2000 as well as various distributions of Linux. Using VMWare you can create multiple virtual machines and install multiple operating systems within them. You can even run them all at the same time if the PC is powerful enough. One of the most popular uses of the

product is by developers and technical support staff who need access to different setups for test purposes. Windows developers also benefit from the fact that when their program crashes and brings the Microsoft operating system down with it only the virtual machine is affected, not the host.

With Windows running under VMWare you can see the Windows desktop in a window on your Linux desktop and use it just as if it was a real Windows machine. You can copy and paste between Linux and Windows. If you need to share files between Linux and the virtual Windows machine this must be done using Samba, since the virtual machine's hard disk is not a real hard disk and cannot be mounted in the usual way.

VMWare has two networking options: host-only networking and bridged networking. Using host-only networking only the host can communicate with the virtual machine's drives. Using bridged networking the virtual machine appears as if on a virtual network connected to your real network and it can see and be seen by other systems that have the appropriate access privileges.

VMWare 2.0 has a feature called Suspend and Instant Restore. This allows you to save a snapshot of the virtual machine with all the windows and files that are currently open. You can then instantly restore from this image. This is a big time-saver as it avoids having to boot up the guest operating system from scratch each time you need it.

Compared to VNC, VMWare is less expensive as it avoids the need for a separate Windows PC, though a commercial license for VNC itself is quite costly. Because of its hardware requirements your system might need upgrading and you'll still need a Windows software license. However, if you want to run Linux but still need to run Windows VMWare may be the best technical solution to your needs.

Using Wine

The trouble with both of the solutions described so far is that they still require you to have a Windows software license, so depriving you of the cost benefit of moving to free software. A better method still would be to use Wine, an open source free software project that aims to allow programs written for Microsoft Windows to run under Linux without any Microsoft software being present at all. However, Wine is still under development. In its current state there's no guarantee that the Windows programs you need to use will run under Wine. But it is getting better all the time. It's certainly worth a try.

The name Wine is said to be a recursive acronym derived from "Wine Is Not an Emulator." If so, it's a little disingenuous, because what Wine does is provide an environment that looks to a Windows program like Windows even though it isn't, thereby allowing the program to run under Linux. It does this by providing functional equivalents to all the Windows Application Program Interfaces (APIs)

the program may use. That seems to me a lot like emulation.

Importantly, Wine doesn't have the performance penalty usually associated with emulators that need to simulate a different hardware environment in software. All Wine has to do is implement each Windows API function under the environment of X Windows running on Linux: converting Windows graphics operations to X calls, making the Linux file system look like a DOS drive and so on. Wine is an interface between the Windows program and Linux and X. Because of the greater efficiency of Linux some applications might actually run faster than they would under Windows on the same computer!

Windows is not a monolithic whole so much as a collection of dynamically linked libraries (DLLs) working together in approximate harmony. Since Microsoft often adds to these DLLs (as well as periodically modifying them to make them incompatible with one another) the Wine developers have a moving target to hit. The likelihood that a particular program will run under Wine depends on whether it uses any APIs that haven't yet been fully implemented.

Most older Windows programs written in C, and many programs developed using Borland C++ or Delphi, run well under Wine since they use few APIs other than those provided by the Windows core components. These basic APIs have long been supported by Wine and are pretty thoroughly tested. Programs developed using Visual C++ or Visual Basic, which are dependent on the Microsoft Foundation Classes and COM / ActiveX components, generally won't run, or behave unacceptably when they do. This means, for example, that Wine won't run Microsoft Office. The only way to find out whether any given program will work is to try it.

As an interim measure, some programs that don't work well under Wine can be made to work better by letting them use the original Windows DLLs instead of the Wine replacements. This can be done by setting up a Windows directory on the Linux file system, copying over the relevant Windows DLLs and making some changes to the Wine configuration file. Of course, if you use even one Microsoft DLL in order to get your program to work of course you are legally obliged to have a Microsoft Windows software license for the system.

A good example of what is possible using Wine is Corel's WordPerfect Office 2000 for Linux [Fig. 3]. Corel did not port the applications in this office suite to Linux. Instead, it invested in the Wine Project to accelerate development to the point at which the Windows executables in the suite would run. This work has undoubtedly helped Wine to run many more applications than WordPerfect Office but there is still much work to be done. Nevertheless, the fact that a major office suite now runs under Linux using Wine is a feather in the cap of the Wine developers.

Using Wine is a matter of downloading and installing the latest version of the software and then

editing the configuration file /etc/wine.conf to map Windows drive letters on to directories in your Linux file system. You must also set the location of the directories that will substitute for the Windows, system and temporary files directories and set a path. You can point Wine at real Windows directories mounted on your PC, but if you do, you won't know whether your application is only running courtesy of files on the Windows PC.

To run a Windows program type " wine myprog.EXE params" into a console window and see what happens. You may see messages appear in the console window as the program runs. They can provide useful diagnostic information if the program doesn't run properly or crashes. Don't think you are home and dry if the program's main window comes up. It's important to test every feature of the program thoroughly to be sure that it all works as it would under Windows itself.

Conclusion

The best way to run Windows programs under Linux, if you can, is to run them using Wine. It uses the least resources, performance is better and you should not need a Windows software license. But Wine may not yet support your program, particularly if it was developed using Microsoft tools. Wine is an open source project, however, so if you have the skills consider trying to solve the problems that prevent your application from running and so contribute to the project.

If you can't get your Windows programs to run using Wine, running them under Windows using VMWare or on a separate PC running VNC will almost certainly solve the problem. Whichever solution you adopt, the fact is that the need to run certain Windows programs need not be an obstacle to migrating from Windows to Linux. ■

INFO

VMWare:
<http://www.vmware.com/> VNC Home Page:
<http://www.uk.research.att.com/vnc/> The Wine HQ:
<http://www.winehq.com/>

Fig 3: Corel WordPerfect Office runs under Linux using Wine

