



# A free consult from Doctor Unix, GP



*The definitive authority on pressing Unix matters...*

[Dr.Unix@nluug.nl](mailto:Dr.Unix@nluug.nl)

**Q:** *Dear Dr. Unix, I find myself in an emotional conflict. I recently installed Linux on my computer, and I must say that I am very, very satisfied. However, I still have a couple of DOS and Windows programs for which there is no real Linux alternative. My friends tell me that I have to configure my system for "dual boot". However, I wonder if they're really my friends when they come with advice like that. So I want to ask you: "Is it possible to run these legacy applications under Linux?". Please, you are my only hope!*

-- "Eager Beaver" from Bumpitt, Alabama.

**A:** Dear "Eager",

You find yourself in a predicament indeed. Not so much a technical predicament, but surely an emotional one! First of all, let's consider the advice of your friends. Guess what? Wrong friends! If you want to take a personal advice from an experienced Unix person: find a new set of friends who are emotionally more in tune with the Linux vibes that so strongly permeate our very existence at this point in time.

Now, moving over to your question. In short, it is possible to run DOS/Windows applications under Linux. That said, emulation is usually the worst of both worlds, multiplied. So, to start with I would strongly urge anyone contemplating creating such a hybrid to look very, very hard if there is really no Linux alternative available. I mean, there really is a lot of software out there, and it is almost always possible to find a suitable Open Source or Commercial Linux application that fullfills your wishes. The biggest problems usually are with custom programs or utilities. Take for instance the Dutch Tax Service (since when is taxation a service, I hear you

think. Well, that question has been much on my mind as well, but let's save that discussion for another day). The Dutch Tax Service makes it possible to file a tax return electronically on a floppy or through a modem. To facilitate this they have written a Tax Floppy which contains the software to do your tax return. As you can imagine for a dusty government agency whose mind set is determined by the events in the Jurassic age, this software is only available for a certain operating system (not being Linux). Such a program is a typical example of a program that you do want to run, but for which no Linux or other Unix alternative is available. Mind you, when someone recently asked whether he could have the source so that the program could be ported, the Tax Service declined. The software is presumably too shabby to release to the tax payers in source code format!

One of the solutions you mention is creating a dual boot system. Although technically possible, I generally advise against it. First of all there is the emotional ballast of still carrying inferior operating systems around. Although barely noticeable at first, the uneasy feeling that comes from this starts gnawing at your conscience, increasing in strength over time until it finally completely absorbs you, and before you know it you'll take a sawed off shotgun into a computer shop and start blowing away hapless sales people and clueless customers. We would not want this, would we?

Linux allows a dual boot situation where you have one or more Linux partitions on a hard disk, and a DOS partition. The Linux loader LILO also supports booting from non-Linux partitions. By specifying an entry like the following in `/etc/lilo.conf`, you can specify that partition `/dev/hda3` is a bootable partition from some other operating system. By entering the entry's label to the LILO prompt, the alternative operating system is booted:

```
other = /dev/hda3
label = duhos
```

For the really twisted minds, the "duhos" operating system can even be made the default OS (by specifying `default=duhos` in the `/etc/lilo.conf`). Please note that a lot of inferior operating systems have a tendency to completely zap the master boot record of your hard disk, removing LILO from the boot sequence!

The biggest disadvantage of a multi-boot system is the utter inconvenience of booting the other OS, and then rebooting the machine back into normality. This inconvenience is especially great for laptop users which, due to Linux's superior power management features, usually never reboot (but instead only suspend and resume).

Fortunately, there are a number of alternatives available to run DOS and Windows programs under Linux. None of these solutions is ideal, but they work to a larger or lesser extent. It's a lot like being on dialysis after a kidney failure: It's not ideal, but much better than not being on it at all :-)

The best results can be obtained with straight DOS programs. A program called "[dosemu](#)" strives to be the perfect DOS emulator under Linux. It runs a lot of DOS programs (including some DPMS applications) in a virtual DOS machine that either runs full screen or in a window under the X Window System. One of my apprentices uses dosemu to run the DOS program that manages his ISDN telephone switch at home. His report can be found [here](#). Fortunately, a lot of Linux distributions package dosemu as a standard feature. In order to make the most of dosemu, you still need a DOS license. That said, some of the latest versions of dosemu come with a free DOS replacement called [FreeDOS](#).

```

xterm
Welcome to DOSEMU 0,98!
C:\>dir

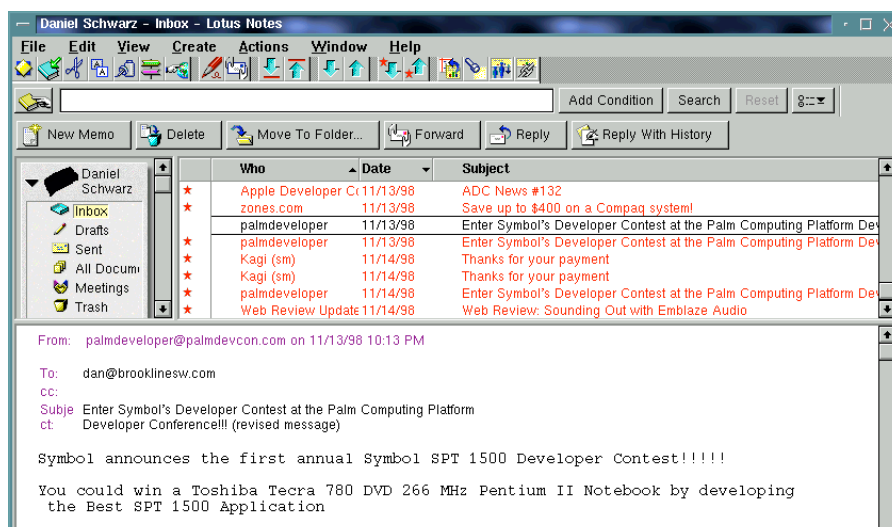
Volume in drive C is drives/c
Directory of C:\

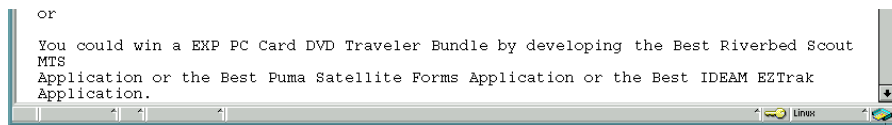
IPL      SYS      54928  12-29-97  10:18a
SUSE     TXT      1136   10-28-99  11:28p
AUTOEXEC BAT      124   10-28-99  11:26p
BOOT     BIN      512   2-04-98   12:47p
COMMAND  COM     41568  12-09-98  12:19a
CONFIG   SYS      53     10-28-99  11:26p
DOSEMU   <DIR>    3-05-00 10:13a
KERNEL  EXE     68112  7-04-99   7:15p
README   EMU      972    7-03-99   7:51p
README   TXT     2215   12-10-98  12:10a
          10 file(s)      169620 bytes
          410484736 bytes free

C:\>ver
DOS-C version 1.0,2
Compatibility level 3,31
C:\>

```

Life gets much more complicated if you want to run Windows programs. The Windows (ahem) operating system is a much more difficult animal to tame. Lots of Windows applications depend on all kinds of undocumented behaviour of the real Windows, and it is not easy to make a Linux (or other Unix) system behave exactly like the original. However, in an extremely brave attempt a number of gifted individuals (with some help of Corell) are trying to create a Windows compatible environment under Unix called [WINE](#) (which reportedly stands for Wine Is Not an Emulator). WINE allows you to run 16 and 32-bit Windows programs (as well as DOS programs by the way) under Linux. Several hundred popular Windows applications reportedly run under WINE, so if the DOS/Windows application you're trying to run is on the WINE compatibility list, you're a man in bonus! I met Corell's project leader for WINE (Gavriel State) a couple of weeks ago, and he is a very competent chap dedicated to getting Corell's WordPerfect suite to run with WINE. And, best of all: WINE is completely Microsoft code free, so you don't need a Windows license to run an application under WINE. Oh, by the way, one of WINE's features (according to the feature list on the [WINEhq web site](#)) is a "Reverse engineering design (that) assures bug-for-bug compatibility". Hmmm, I don't know about you, but remarks like these makes me shiver! Brrrrr.....

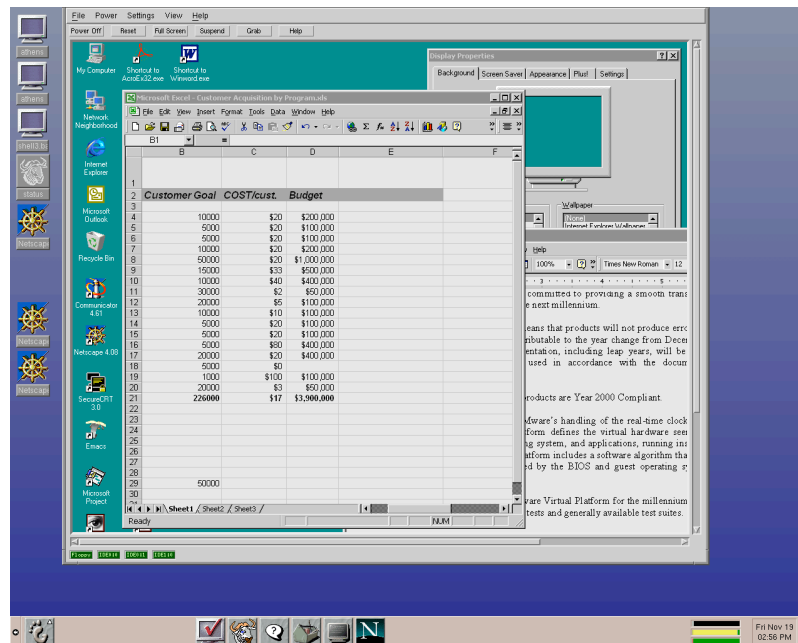




Lotus Notes running on Linux under WINE. (click for full image)

The third and most spectacular option to run Windows applications under Linux is most definitely **VMWare**! Old-timers like Doctor Unix remember an IBM operating system way back in the 80's called VM (Virtual Machine). One of VM's smashing features (and more precisely, its main function) was the possibility to run multiple virtual machines, each of which could run its own operating system. So owners of an IBM mainframe could run multiple instances of OS's like VSE and MVS, each of which thought that it had the entire machine to itself. Whenever the guest OS did something like accessing the hardware, the VM hypervisor intervened and executed the instruction on behalf of the guest OS. This came with such obscure technologies like "double paging", simulated supervisor mode and other cool operating system internals stuff. Later, IBM more or less dumped VM in favour of a hardware feature called PR/SM (pronounce: Prism) which did more or less the same, but instead buried in hardware and microcode. One of VM's most spectacular features was its crash handling: if VM for one reason or another panicked head to toe, it did a "control block swap", and one of the guest operating systems (the so-called Preferred Guest) was then unleashed unto the real hardware (without it even noticing it), and could therefore survive the hypervisor crash.

Fortunately, somebody paid attention to IBM's work here, and a couple of guys created a VM for PC's (named VMWare), allowing multiple operating systems to run in a virtual machine that is entirely under VMWare's control. Now, this has to be seen to be believed! So, a picture first:



VMWare on Linux, running Microsoft Office. (click for full image)

VMWare allows multiple operating systems to run under control of VMWare's hypervisor. So, you can for instance run Microsoft Windows 95, 98 or NT under VMWare on Linux. It works truly beautifully, emulating things on the lowest level, Phoenix BIOS and all! In my modest opinion, VMWare is the best option for true and complete Windows capabilities under Linux. But, there are disadvantages as well. VMWare is not free, it is not very fast (there are performance losses across the board) and you need a separate Microsoft Windows license to legally run it under VMWare. But, all in all, surely an option worth considering! (Sidestep:

along the same lines a group of people is trying to write a free version of this technologie called [FreeMWare](#)).

My only argument with VMWare is that perhaps it works too good! I mean, if VMWare runs really, really well, and most people that run Linux run VMWare for Windows compatibility, what is the incentive for software developers to bring out Unix/Linux versions of their applications? VMWare's developers might be obstructing the revolution, and as it goes with anti-revolutionary types, when we finally take over they unfortunately will have to be put against the wall and shot. My personal optimum would be if VMWare ran just good enough to convert people to Linux, but bas enough to make them want to install a Linux native version of their applications.

Oh, and by the way, VMWare is not a completely new idea even in the Unix space. A company named [Insignia](#) specialised in creating virtual PC's some time ago with products like SoftWindows95, that even ran under Solaris and HP-UX. But, since these applications had to emulate the entire Pentium instruction set on a RISC chip, they ran so horribly slow that it almost defied belief. VMWare's advantage here is that it runs on the same chip as the guest operating system.

So, Eager Beaver, I hope this advice helps you in your difficult path ahead. But, please note, don't let the evils of emulation tempt you from the True Unix Path. Enlightenment can only be reached by hard work, dedication, and Faith!

**Q:** *Dear Dr. Unix,*

*I wrote a letter to my girl friend in Taiwan using "celvis" (a Chinese version of vi). However, I then wondered, if I executed "wc" (Word Count) on this letter, what kind of data does it give me? Should I run "cc" (Character Count) instead?*

-- "Confusedius" from Jerkwater, USA.

**A:** Dear "Confusedius",

The question you pose is an interesting one. Obviously, one would most definitely want to set "LANG=Chinese", and have "wc" give the correct data. However, not all Unices are that far already in their internationalisation efforts. A separate Chinese character counting program would be a good idea, but unfortunately "cc" is allready taken, and so is "dd". Other names such as "ccc" and "kkk" would pose problems in certain regions of the world due to the extreme left- or right- wingedness of the organisations usually denoted by these acronyms. So I would propose "ee" as the name of that program. GNU ee. Any volunteers?

---

*Is a pressing Unix matter bugging you? Why not ask [Doctor UNIX!](#)*