

A GNU/Linux Users' Guide to Making Windows Work Properly *

by Nex Ami[†]

November 27, 2004

	Linux Guide To Windows
Document Name:	linuxguidetowindows.pdf
Version:	0.1

Contents

1 Emulation (a simple solution?):	1
1.1 KDE Cygwin	2
1.2 MinGW	2
2 Windows Tools	3
2.1 Emacs	3
2.2 MinGW	4
2.3 LaTeX tools	4
2.4 WinSCP	7
2.5 PuTTY	7

GNU/Linux is the best operating system in the world. Well, a lot of Linux users would like to think that anyway. In fact, I'm one of them. But from time to time (and if I was to be honest, an annoyingly large proportion of the time) I still need to venture into windows to do some of the things I want to do.

This little guide is for people like me who want so desperately to use linux, but often find themselves in Windows. I'm mainly going to cover the basic stuff I use in Windows for things that would usually be considered 'GNU/Linux' domain. In fact, the reason they are considered GNU/Linux domain is mainly because they *are* GNU/Linux domain (in fact most of the 'windows' tools I talk about here are just ports from the original GNU/Linux programs). These are essentially the tools I require for my GNU/Linux lab at University - but the windows 'equivalents'. So read on and find out how you too can make windows just that bit more bareable.

*well... almost

[†]<http://www.n-e-x.co.uk>

1 Emulation (a simple solution?):

Emulation is the process of making one type of system behave like another. For our purposes, this means making windows behave like GNU/Linux. On the offset, this seems like the perfect solution: You can go about your windows tasks and have a GNU/Linux window open to do your linux stuff. But there are a few drawbacks to this sytem.

One of the hardest hitting is speed. Not only will the software you run in your GNU/Linux window be slow, the rest of your system will slow down as well. When you think about it, this makes a lot of sense because you are really just running two operating systems at the same time.

Second is the issue of file swapping across operating systems (i.e. being able to work on the same file in both Operating Systems, even if the Operating Systems are on seperate partitions). This is not usually defaultly supported by emulators (and certainly not by windows). You can get around this problem, but it is beyond the scope of this essay (it's such a cop-out saying that, but hey, I'm already writing the rest of this article out of the goodness of my heart).

Last thing to say about emulators before I tell you how to install one is that there are no 'free' ones that are totally stable and finished. They all still have their own little quirks and problems. But here is my favourite: KDE Cygwin.

1.1 KDE Cygwin

Well, to be honest, the website tells you most of what you need to know. so go there now: <http://kde-cygwin.sourceforge.net/kde3/installation.php>

Here are some notes I use when installing:

Install cygwin (N.B. to avoid problems, you may have to first choose 'install all', most likely producing an error, then run setup again choosing all default options. This is not always the case on all systems, so try without doing this first).

Run cygwin and type:

```
ps -ea
```

to make sure that only bash and ps are running. Then type:

```
rebaseall -v
```

Then type:

```
startx -rootless &
```

to set up the KDE.

A slight drawback to using cygwin is to do with the way it emulates. You see, it can actually run any DOS based program in bash (which you usually can't do in GNU/Linux) but can only compile programs to work in cygwin (not even to work in GNU/Linux). This annoyed me immensely, but is the price you pay for getting the free version instead of the full, unrestricted one. However, I have found a handy way round this. After installing KDE Cygwin, just follow the steps below:

1.2 MinGW

MinGW is a system similar to cygwin in many ways, but much more specific. It is an attempt to provide just enough unix tools to use the GNU Compiler Collection. Although we won't be using all the facilities much, the fact that it has a version of GCC that produces windows native executables is going to make our little Cygwin system much more useful.

So go to <http://www.mingw.org/download.shtml> and find the table that says "Current". What you want to do now is download anything in that table that ends with a .exe,.sh,.tar.gz and tar.bz2. You don't need to download anything with an "src" in it. Put all those files in a folder called "mingw" then run all the exe files except MinSYS. Once you've done that. Extract all the other files into that folder (maintaining folder names within the archives). You might need an archive manager to do this. Try this one: <http://www.7-zip.org>

Install MinSYS correctly pointing to MinGW directory.

Edit the `c:\cygwin\etc\profile` file (assuming you installed Cygwin to `c:\cygwin`, if not replace this with the correct directory) and then insert `/cygdrive/c/mingw/bin:` at the beginning of the path string (assuming MinGW was installed at `c:\mingw`).

Cygwin should now work, with KDE added on. Also, gcc should now compile windows executables that do not require additional dll files to run. Please note, however, that there is a minor incompatibility between cygwin emacs and mingw. Hopefully this won't bother you too much. And that's you with a near enough full version of GNU/Linux on Windows!

2 Windows Tools

So, emulators are all good and well, but they slow everything down and can take a while to load, especially since you have to run that silly rebaseall command everytime you start it up. So what's the alternative?

Well, as it happens, there is a Windows equivalent tool for everything on Gnu/Linux, and we're going to talk about some of them here. The number of useful conversions of GNU/Linux tools is too numerous to mention here, so I will once again concentrate on the ones I feel are most useful to me (in many ways, this is why the emulator solution is better - you don't have to find an individual program equivalent for everything you're used to doing). So first up, is the wonderful emacs editor.

2.1 Emacs

The Emacs editor was one of the first wonders to come out of the Free Software Foundation (<http://www.fsf.org>) who created much (if not most) of the GNU/Linux Operating System. It is one of those sort of tools that you think is terribly complicated and difficult to use when you first get it, but then you soon can't imagine life without it. It is for this reason that I include it first as the must have GNU/Linux tool for Windows.

As it happens, it's also surprisingly easy to get hold of, since the fsf actually produce a windows binary version. So pop along to:

<http://ftp.gnu.org/pub/gnu/emacs/windows>

and look for the file with the largest filesize on the page. This is probably the one you want and will usually have the word 'fullbin' in the name. All you

need to do once you've downloaded it, is to extract it to the root directory on your hard drive (i.e. c:\). Then rename the new folder that begins with 'emacs' to c:\emacs. In there will be a folder called bin and in that bin folder will be a program called runemacs.exe. Need I say more? (if I do need to say more, there is a good chance you won't understand how to use emacs anyway and have probably not understood much of this article so far. If this is the case, don't lose heart, just go and read up a bit on GNU/Linux and all will be well).

2.2 MinGW

MinGW stands for 'Miniature GNU for Windows'. It is also an emulator of sorts but is very small as it only supplies enough functionality to use the GCC compilers. So, in essence, it *is* the GCC compilers, but for Windows. You can install this as stated in section 1.2, but after you have installed MinSYS, the rest of the instructions are unimportant to you unless you are also running Cygwin. Instead, you must set up some environment variables in Windows so that it knows where the GCC program files are. The details of how to show this are shown below:

Windows 95/98

Include the following in your c:\autoexec.bat (eg. using Emacs):

```
set PATH=%PATH%;c:\mingw\bin;
```

Restart the computer.

Windows NT/2000/XP

Right-click on the "My computer" icon and choose Properties. (For Win2000, click the 'Advanced' tab). Select the "Environment" tab.

If a "Path" variable does not exist, create a "user variable" called "Path" with the following value in one line.

```
c:\mingw\bin;
```

If the Path variable exists, append the above to it (a semi-colon should separate all path definitions).

Restart the computer.

2.3 LaTeX tools

LaTeX is probably the hardest thing to set up in this tutorial. If you don't know what it is, then you probably don't need it. But I need it for University and so a windows equivalent is useful for me (as it happens I am writing up this article on Emacs in Windows running LaTeX). So here is the way I did it. I'm pretty sure there isn't an easier/better way. But what do I really know at the end of the day?

Windows 95/98 Environment Settings

Make a folder called c:\temp and another called c:\tmp. Include the following in your c:\autoexec.bat (eg. using Emacs, which the rest of this section will assume you installed using my instructions).

```
set TEMP=C:\TEMP
set TMP =C:\TEMP
set HOME=C:\
set DICTDIR=C:\usr\local\lib
set PATH=%PATH%;c:\usr\local\bin;
set PATH=%PATH%;c:\program files\texmf\miktex\bin;
set PATH=%PATH%;c:\emacs\bin;
set PATH=%PATH%;c:\program files\ghostgum\ghostview\gsview;
set PATH=%PATH%;c:\program files\ghostgum\gs\gs8.14\lib;
set PATH=%PATH%;c:\program files\ghostgum\gs\gs8.14\bin
```

Restart the computer.

Note: Some systems do not like long filenames, so you might need to change the appropriate lines to:

```
set PATH=%PATH%;c:\emacs\bin;c:\progra~1\texmf\miktex\bin;
set PATH=%PATH%;c:\progra~1\ghostgum\ghostview\gsview;
.etc
```

Windows NT/2000/XP Environment Settings

Right-click on the "My computer" icon and choose Properties. (For Win2000, click the 'Advanced' tab). Select the "Environment" tab.

If there is no variable called "Path", create a "user variable" called "Path" with the following value in one line (here I have broken the lines).

```
c:\emacs\bin;c:\program files\texmf\miktex\bin;
c:\program files\ghostgum\ghostview\gsview;c:\usr\local\bin;
c:\program files\ghostgum\gs\gs8.14\lib;
c:\program files\ghostgum\gs\gs8.14\bin
```

If the Path variable exists, append the above to it.

Create a "user variable" called "HOME" with value c:\

Create a "user variable" called "DICTDIR" with value c:\usr\local\lib

Create a "user variable" called "TMP" with value c:\temp (if it is not there already. The value is not important).

Create a "user variable" called "TEMP" with value c:\temp (if it is not there already. The value is not important).

Restart the computer.

Downloads

You need to download the following files:

Miktex (Run this program and choose your preferred (eg. Large) Miktex version for downloading. Download to a temporary directory.):
<http://prdownloads.sourceforge.net/miktex/setup.exe>
Ghostscript: <ftp://mirror.cs.wisc.edu/pub/mirrors/ghost/AFPL/gs814/gs814w32.exe>
GSview 4.6: <ftp://mirror.cs.wisc.edu/pub/mirrors/ghost/ghostgum/gsv46w32.exe>
AUCTeX 11.14: <http://ftp.gnu.org/pub/gnu/auctex/11.14/auctex-11.14.zip>
Ispell 3.1.20: <http://www.math.aau.dk/~dethlef/Tips/local/ispell-3.2.06-cygwin-1.3-bin.tar.gz>
Cygwin DLL v. 1.3.10: <http://www.math.aau.dk/~dethlef/Tips/local/cygwin1.dll>
.emacs: <http://www.math.aau.dk/~dethlef/Tips/local/startup1.zip>

Installation

Just follow the instructions below for each package:

AUCTeX:

- Unzip auctex-11.14.zip to c:\emacs\site-lisp (89 files)
- Rename c:\emacs\site-lisp\auctex-11.14 to c:\emacs\site-lisp\auctex
- Unzip startup1.zip to c:\ (the value of your HOME variable). This creates the startup file c:\.emacs. You must edit this file, if you install auctex to a different place.

Ispell:

- Unzip ispell-3.2.06-cygwin-1.3-bin.tar.gz to c:\ (the directory cannot be changed) (22 files)
- Put cygwin1.dll into c:\usr\local\bin. If you have cygwin installed, skip this. In addition, you should never have more than one version of cygwin1.dll in your path.

GhostScript:

- Run gs814w32.exe
- Choose c:\program files\ghostgum\gs as installation directory
- Accept everything else

Ghostview:

- Run gsv46w32.exe
- Choose c:\program files\ghostgum\ghostview as installation directory
- Accept everything else

MikTeX:

- Run setup.exe from the directory, where you downloaded Miktex.

- When prompted for "Path to installation folder", choose `c:\program files\texmf`
- When prompted for "Create local TEXMF tree", choose `c:\localtexmf` as "Path to root folder"
- Accept everything else
- Restart (might not be needed)

You should now have a version of Emacs running which can handle LaTeX documents. Syntax highlighting will be available and the Command Menu will now contain a LaTeX option and a LaTeX PDF option. You can also view dvi files with the View command in the Command Menu.

2.4 WinSCP

WinSCP is a secure shell client for Windows (i.e. a Windows version of the unix `ssh` and `scp` commands). Pop along to <http://winscp.sourceforge.net/eng/download.php#download2> and download the WinSCP installation package. Then just install it and run the program. It's all self explanatory. If you can't use it, read the documentation and wonder about why you can use complicated and hard to remember console commands but can't navigate a simple gui system.

2.5 PuTTY

PuTTY is a command line version of WinSCP for if you want direct console/terminal access to a server. Infact, it is actually the backend for WinSCP (WinSCP is essentially just a graphical user interface for it). You can get it here:

<http://the.earth.li/~sgtatham/putty/latest/x86/putty-0.56-installer.exe>

Just install and run. To access a server, something like `'sibu.dcs.gla.ac.uk'` would be appropriate and then you would be asked for your username and password. Once running, it is very similar to running a terminal on the remote machine itself (no graphical interface of course) though you may find it runs a little slower. To copy files from a remote machine to yours is more easily done with WinSCP and I recommend just doing it that way when possible. Save PuTTY for when you actually need to work on files directly on a remote machine.