

Exploring CS1101C On Windows With Cygwin

(Installation And Usage Instructions)

BY ASHWIN NANJAPPA

Web: <http://www.comp.nus.edu.sg/~ashwinna/>

January 27, 2007

Why?

Have you been wanting to have a setup on your Windows similar to the one you experience when you connect to Sunfire servers?

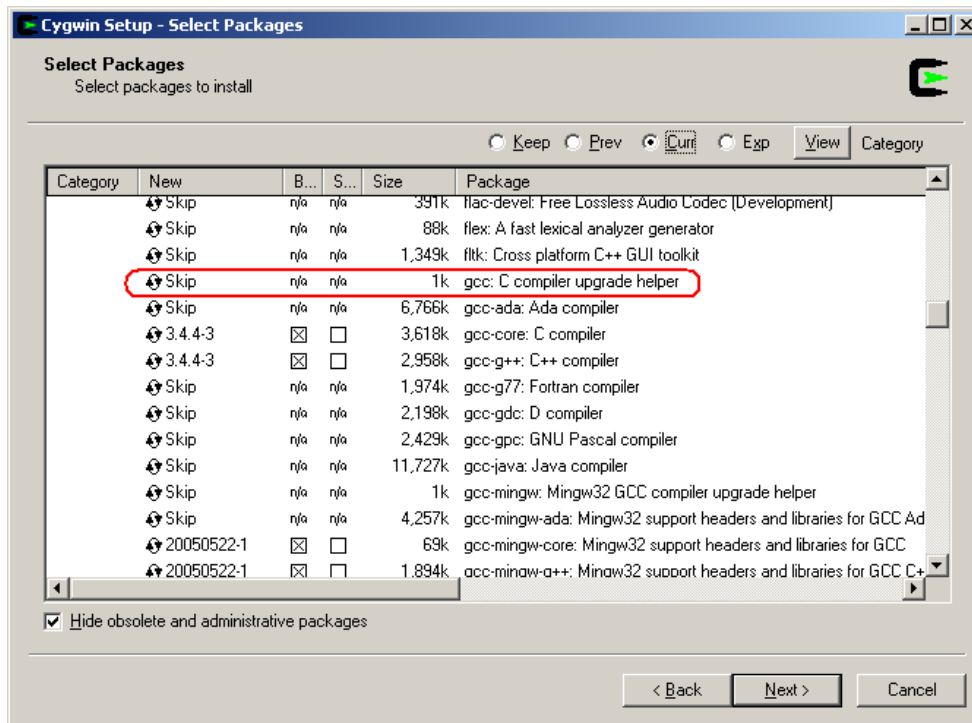
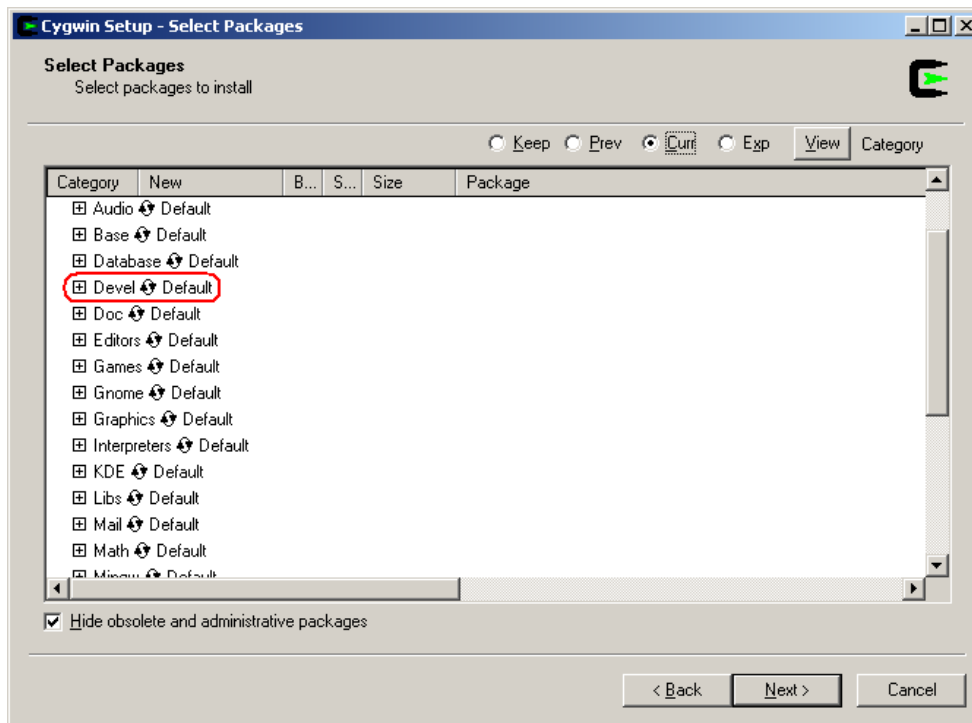
Cygwin is a Linux/UNIX-like environment for Windows. A lot of Linux applications (like the *gcc* C compiler and the *ViM* editor) have been ported to Cygwin. So, by installing Cygwin and these applications you can experience the same environment as you do in your lab sessions when you login to the Sunfire servers. Let us see how.

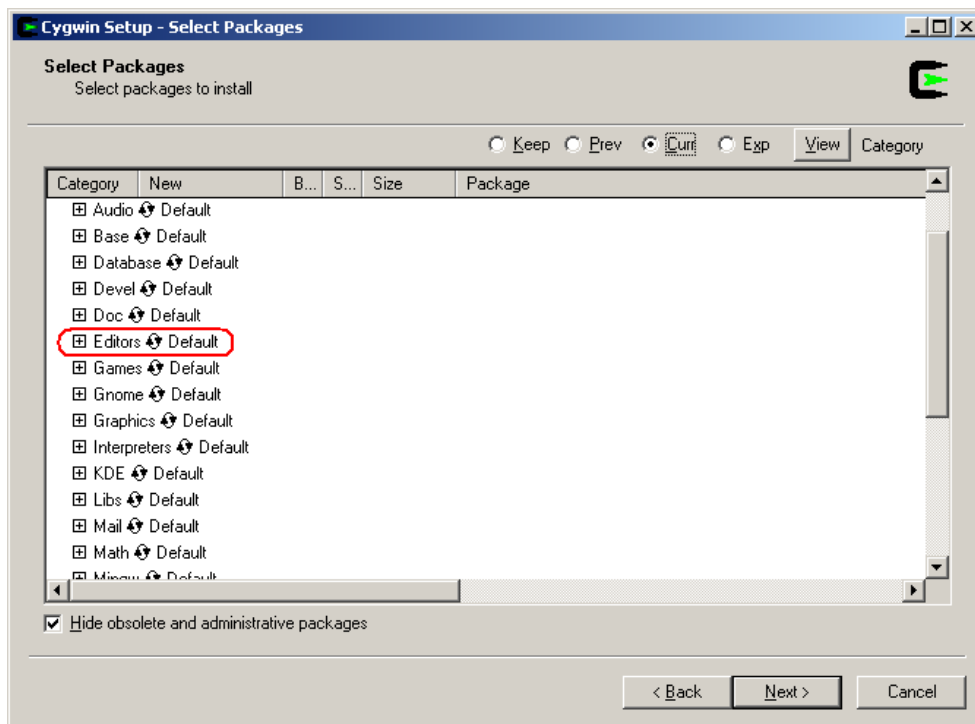
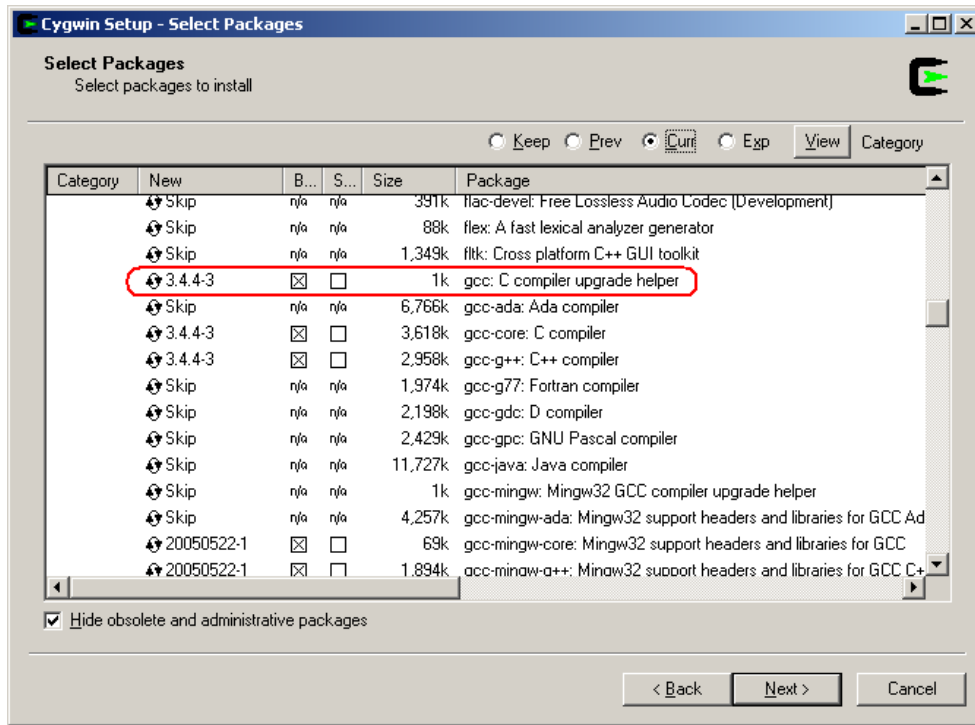
Installation

1. Goto <http://www.cygwin.com/> and download `setup.exe`. Run it.
2. You can click `Next` on most of the setup choices. The default selections in these cases are fine for most of us.
3. When the setup arrives at the selection of mirrors, choose any you want. In my personal

experience, <http://mirrors.kernel.org/> has worked reliably.

4. In the software selection, choose *gcc* (Devel → gcc) and *ViM* (Editors → vim). See the screenshots below if you need help with this step.



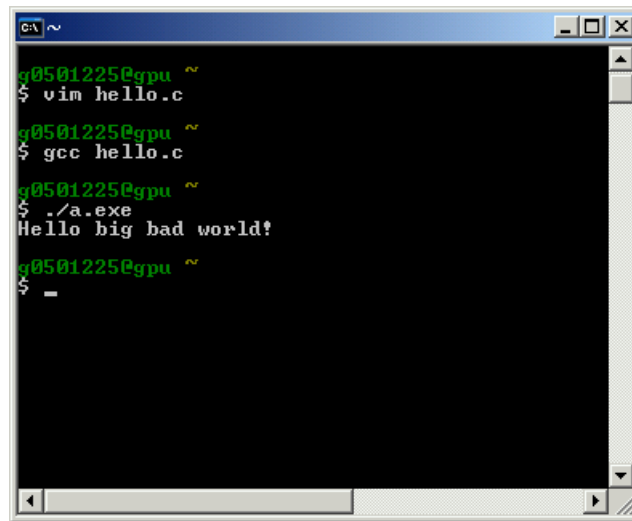


5. Click through **Next** on the further dialogs and finish the installation. Since the installation program is downloading from the Internet and installing the applications, it will take some time. Be patient.
6. You've now installed a Linux-like environment on Windows. Congrats! You can try your programming in it.

Usage

(In the steps below, the \$ symbol represents your command prompt. The command you type follows it.)

1. Click on the *Cygwin bash shell* icon to start Cygwin. You can find this icon in your Windows Start menu or on your desktop.
2. Cygwin starts up and presents a command prompt just like on Linux/Solaris in the lab Sunfire servers.
3. Use ViM and write/edit your C code. For example: `$ vim hello.c`
4. Save the C source code file and exit ViM.
5. Compile the C source code file using gcc. For example, `$ gcc hello.c`
6. If there were no errors, gcc would've produced an executable file named `a.exe`. Execute it to see your code in action. For example, `$./a.exe`



```
q0501225@qpu ~  
$ vim hello.c  
q0501225@qpu ~  
$ gcc hello.c  
q0501225@qpu ~  
$ ./a.exe  
Hello big bad world!  
q0501225@qpu ~  
$ _
```

Problems?

If you face any problems, describe it on the CS1101C IVLE forum or email me or talk to me.

Good luck and happy hacking!

□