2019 Semester 1
COMP20007 Design of Algorithms
School of Computing and Information Systems
The University of Melbourne

Accessing Dimefox

A guide to accessing the School of Engineering's student machines (`dimefox`) to test your assignment before submitting via the LMS. For additional general information on the university's engineering servers, check out `https://ithelp.eng.unimelb.edu.au/itservices/students/general_unix.html`.

1 Accessing the university network

From campus

If you are on a lab machine, or your own machine connected to UniWireless, you are already on the university network.

From home

If you want to access the engineering servers from home, you'll need to be connected to the university's virtual private network (VPN). `https://studentit.unimelb.edu.au/findconnect/vpn` shows you how to install *Cisco AnyConnect VPN* software on your machine, and log in to the university Virtual Private Network through `remote.unimelb.edu.au/student`.

An open source alternative to Cisco AnyConnect called *openconnect* is also available: see `https://people.eng.unimelb.edu.au/lucasjb/oc.html`.

2 Getting your files to the server

Before you can compile your assignment on `dimefox`, you'll need to get your files to your university hard drive. There are a few ways to do this:

From the lab machines

Your university hard drive accessible from within `dimefox` is the same drive as your H drive on the lab machines. So, if you have your files somewhere on your H drive on the lab machine, they are already also accessible from `dimefox` and you can go to the next step.

From your personal Windows machine

If not installed already, install `pscp` from `http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html`. Also, make sure you are connected to the university network (see above).

1. In `cmd`, navigate (using `cd`) to the directory that has your source code files in it.
2. Copy these files to your university hard drive using a command like

   ```
   pscp <list of my files> <username>@dimefox.eng.unimelb.edu.au:
   ```

   For example, say I wanted to copy all the `.c` files and `.h` files in the current directory, and my Makefile to my engineering hard drive:

   ```
   pscp *.c *.h Makefile <username>@dimefox.eng.unimelb.edu.au:
   ```

3. Enter your university password.

These steps will copy your files to the root folder of your engineering hard drive, and you are ready to access them from `dimefox`.

Note: if you would like to copy the files into a particular folder on your engineering harddrive, rather than the root folder, you can specity its path after the `:` in your `pscp` command. For example, say I want to copy my files into the H drive folder `comp20007/assignment1`, the command would be

```
pscp *.c *.h Makefile <username>@dimefox.eng.unimelb.edu.au:comp20007/assignment1/
```

From your personal Mac or Linux machine

Before you start, make sure you are connected to the university network (see above).

1. Open up a terminal and navigate (using `cd`) to the directory that has your source code files in it.

   Note: For this step to work, you must not be logged into `dimefox` in the current shell window! If you are currently logged into `dimefox`, I suggest opening a second shell logged into your machine. That way, you can copy from one shell and test on the other, without having to log in and out repeatedly.

2. Copy these files to your university hard drive using a command like

   ```
   scp <list of my files> <username>@dimefox.eng.unimelb.edu.au:
   ```

   For example, say I wanted to copy all the `.c` files and `.h` files in the current directory, and my Makefile to my engineering hard drive:

   ```
   scp *.c *.h Makefile <username>@dimefox.eng.unimelb.edu.au:
   ```

3. Enter your university password.

These steps will copy your files to the root folder of your engineering hard drive, and you are ready to access them from `dimefox`.

Note: if you would like to copy the files into a particular folder on your engineering harddrive, rather than the root folder, you can specity its path after the `:` in your `scp` command. For example, say I want to copy my files into the H drive folder `comp20007/assigment1`, the command would be

```
scp *.c *.h Makefile <username>@dimefox.eng.unimelb.edu.au:comp20007/assignment1/
```

3 Testing your files on the server

You're now ready to log into `dimefox` and compile your assignment. There are a few ways to do this:

From the lab machines

1. Open PuTTY and type in as the Host Name `dimefox.eng.unimelb.edu.au`. Do not change any of the other settings. Press Open. This will launch a terminal inside `dimefox`, where you can run commands like `gcc` and make to test your assignment.
2. Navigate (using `cd`) to the folder where your assignment files are on your H drive. Once you can see your files by typing `ls`, you're in the right place.
3. Once you are in the right folder, type `make` to compile your assignment (as usual). If you see no errors, all is well.
    - If your directory has leftover `.o` files, `make` may not compile your assignment. If this is the case, it might be because your `.o` files were compiled by windows gcc while you were compiling on the lab machines. Run `make clean` to remove all the Windows `.o` files, and try running make again.
    - If you run into any other errors, fix your code until it compiles.
    - From the lab machines, you can make changes to your source files and then recompile without needing to copy them over to your engineering H drive, because you are editing the right files directly from the lab machines.
4. At this point, you probably also want to test your program to see if it produces the right output on `dimefox`. Run your executable like normal: `./programname ...` and compare its output with the expected output.
5. When you are finished testing, you can close PuTTY.

From your personal Windows machine

If not installed already, install PuTTY from `http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html`. Also, make sure you are connected to the university network (see above).

1. Open PuTTY and type in as the Host Name `dimefox.eng.unimelb.edu.au`. Do not change any of the other settings. Press Open. This will launch a terminal inside `dimefox`, where you can run commands like `gcc` and make to test your assignment.
2. Navigate (using cd) to the folder where your assignment files landed during Section 2. If you did not specify a custom folder, you will already be in the right folder when you launch PuTTY. You can verify this by typing ls, which will show the files in your current directory.
3. Once you are in the right folder, type `make` to compile your assignment (as usual). If you see no errors, all is well.
    - If you have copied over your `.o` files, `make` may not compile your assignment. If this is the case, it might be because your `.o` files were compiled on Windows. Run `make clean` to remove all the Windows `.o` files, and try running make again.
    - If you run into any other errors, fix your code until it compiles.
    - If you need to make any changes to your files to recompile them, make sure you copy the new files over to `dimefox` again! (So, back Section 2 "Getting your files to the server").

    For the copy step, make sure you are using `cmd`. Don't try and copy files from within PuTTY.

    You could also experiment with a command-line editor to edit your files from within `dimefox`. Try `vim` or `emacs` (or just nano if you don't want to enter into both an incredibly steep learning curve).

4. At this point, you probably also want to test your program to see if it produces the right output on `dimefox`. Run your executable like normal: `./programname ...` and compare its output with the expected output.

5. When you are finished testing, you can close PuTTY.

From your personal Mac or Linux machine

Make sure you are connected to the university network (see above).

1. From within a terminal, run the command `ssh`. This will launch a terminal inside `dimefox`, where you can run commands like `gcc` and `make` to test your assignment. Run `ssh` using a command like:

   ```
   ssh <username>@dimefox.eng.unimelb.edu.au
   ```

2. Navigate (using `cd`) to the folder where your assignment files landed during Section 2. If you did not specify a custom folder, you will already be in the right folder when you log in with `ssh`. You can verify this by running `ls`.

3. Once you are in the right folder, type `make` to compile your assignment (as usual). If you see no errors, all is well.

   - If you have copied over your `.o` files, `make` may not compile your assignment. If this is the case, it might be because your `.o` files were compiled with a different compiler. Run `make clean` to remove all the old `.o` files, and try running `make` again.

   - If you run into any other errors, fix your code until it compiles.

   - If you need to make any changes to your files to recompile them, make sure you copy the new files over to `dimefox` again! (So, back Section 2 "Getting your files to the server").

     For the copy step, make sure you are not logged into `dimefox` in your current shell! I recommend opening a new terminal window and running `scp` from that one, to avoid having to log in and out of `dimefox` repeatedly.

     You could also experiment with a command-line editor to edit your files from within `dimefox`. Try `vim` or `emacs` (or just nano if you don't want to enter into both an incredibly steep learning curve).

   - At this point, you probably also want to test your program to see if it produces the right output on `dimefox`. Run your executable like normal: `./programname ...` and compare its output with the expected output.

   - When you are finished testing, you can log out of `ssh` using the command `logout`.