

CS355 Setup

1. Account

You will need an account on our Linux server: `goldengate.cs.brynmawr.edu`. Students who pre-registered for 355 should have received an email (your BMC or HC email) with a username/passwd from our system administrator David Diaz. If you have not, or were not on the class list by 1/20, you need to email David (ddiaz1@brynmawr.edu) to request an account. If you have an account but can't remember your password or have other problems, you need to email David also.

2. Sever/remote access

Although goldengate is our main gateway server, in CS355 you are strongly discouraged from programming on goldengate directly. Soon you will know enough to do a lot of damages to servers (including disabling them completely). Obviously you won't do this intentionally, but an innocent programming error can have the same effect easily. It goes without saying that it would be very bad for other students and faculty if you take down goldengate. If you are sitting in our labs (230, 231 or 232), you are not programming on goldengate, but one of the lab machines. If you are connecting from elsewhere, then you should connect to goldengate first and then to another lab machine (there is a list of lab machine hostnames on the course website) and program there instead.

Lab machines in 232 are all Linux. Our lab machines in 230 and 231 are dual boot Windows/Linux. At any moment one of those lab machine may be booted into Windows and thus become unavailable to ssh connections. Prof. Towell has set up a page that checks on the accessibility of lab machines every minute:

<https://cs.brynmawr.edu/~gtowell/mstat.html>

<https://www.cs.brynmawr.edu/~gtowell/crp.html>

3. Handouts

I hand out a lot of materials by leaving them in a world-readable folder in my home directory on goldengate at "`~dxu/handouts/cs355`". That includes lecture notes, lab notes, homework/project handouts and code examples. Handing materials out this way instead of putting them on a website or Moodle forces you to navigate the Unix file system via the command line, which has obvious pedagogical reasons in an OS class.

Sample shell commands:

- a. copy to your own account on goldengate:
`cp ~dxu/handouts/cs355/lecs/*.pdf .`
- b. copy to your local computer:
`scp`
`"username@goldengate.cs.brynmawr.edu:~dxu/handouts/cs355/lecs/*.pdf" .`

Note that the `*.pdf` will match all files that have .pdf at the end of its filename. You can be more specific and specify `2*.pdf`, which will only match the week2 lecture notes, and so on.

Also note "**username**" in the sample command should be substituted with **YOUR** username, not mine, and don't type the literal word username!

In principal, **scp** works exactly like **cp**, except for the prefix **username@goldengate.cs.brynmawr.edu:**, and the fact that you have to quote the path with **"** if you have wildcard matching (*). The quoting is needed on macs, whose default shell has changed to zsh. It's not necessary if you are running bash or tcsh, but it doesn't hurt either.

4. More details on ssh (skip if you have seen this before from 151 or other classes)

- Terminal (mac)
- OpenSSH (windows 10)
- from either above, ssh to connect to goldengate.cs.brynmawr.edu

Note that this is only necessary if you choose to not come to the labs to do your assignments. In that case you **MUST** make sure that you have a functional remote connection to our server that will allow you to complete your work. All examples below are given with goldengate as the target. Recall that in 355 you are asked to go a further step and connect from goldengate to another lab machine. The process is however exactly the same and therefore left to you to extrapolate. Here are a few options that you can choose from. Make sure one of them is working for you.

1. **ssh username@goldengate.cs.brynmawr.edu** – simply execute this command from a shell. This limits you to working in a terminal window, which is restrictive. The advantage is you are directly on the server, with access to all your files and there is no need to copy anything back and forth (as opposed to 3 below), and there is no set up on the local machine (as opposed to 2)
2. **ssh -X username@goldengate.cs.brynmawr.edu** – the additional flag **"-X"** sets up X-forwarding, which means you should be able to get all the windows as you normally would sitting in front of a lab machine. However, in order for this to work properly, an X server must also be installed on your local machine, which might require more set up (I tested for macs, installing XQuartz is all that is required. You are on your own with a Windows machine). If everything is working correctly, after login, you should be able to type commands into the server terminal as if you are sitting in front of a lab machine, but anything that pops a new window will show it on your local machine. For example, **"xterm&"** will start new server terminal windows on your local machine and **"emacs&"** will start a new emacs window similarly. **"code&"** to start VS code remotely.
3. Use **"scp"** (or some other ftp client) to copy all the files from the server to your own machine, program there and then copy them back – if you go this way, you must compile and run all your work on goldengate before you submit to make sure that they will execute properly on our server. Of course, you must also install a C/C++ compiler on your local machine as well, along with your editor of choice.

5. Reviews

A list of things you should refresh if it has been a while, or your pre-req sequences differ from standard expectations. If you were not completely comfortable with some of the explanations in 3&4, then you should allocate significant time reviewing. I have provided links at the bottom of the class website as starting points but you are encouraged to google for more help:

1. Unix/Linux file systems navigation (for example, if you are not comfortable copying things from the handout directory I listed above, then you should prepare for at least a couple of hours of reading Unix tutorials and practicing)
2. Linux command line (review ssh and scp if you are going to connect remotely, and practice to make sure you can)
3. Emacs/VSCode (or some other code editor of your choice)
4. C/C++ programming (remember **malloc** and **free**, try coding a generic linked list in C (**void ***))
5. gdb (or some other C/C++ debugger)
6. valgrind