ECE/CS 250 – Spring 2017 – Prof. Sorin
Recitation #1 – Unix

Objective: In this recitation, you will learn how to login to a Unix machine and do some basic file manipulation and text editing. You will need these skills so that you can develop C programs. These are also useful skills if you plan to have a career in computing.

Complete as much of this as you can during recitation. If you run out of time, please complete the rest at home.

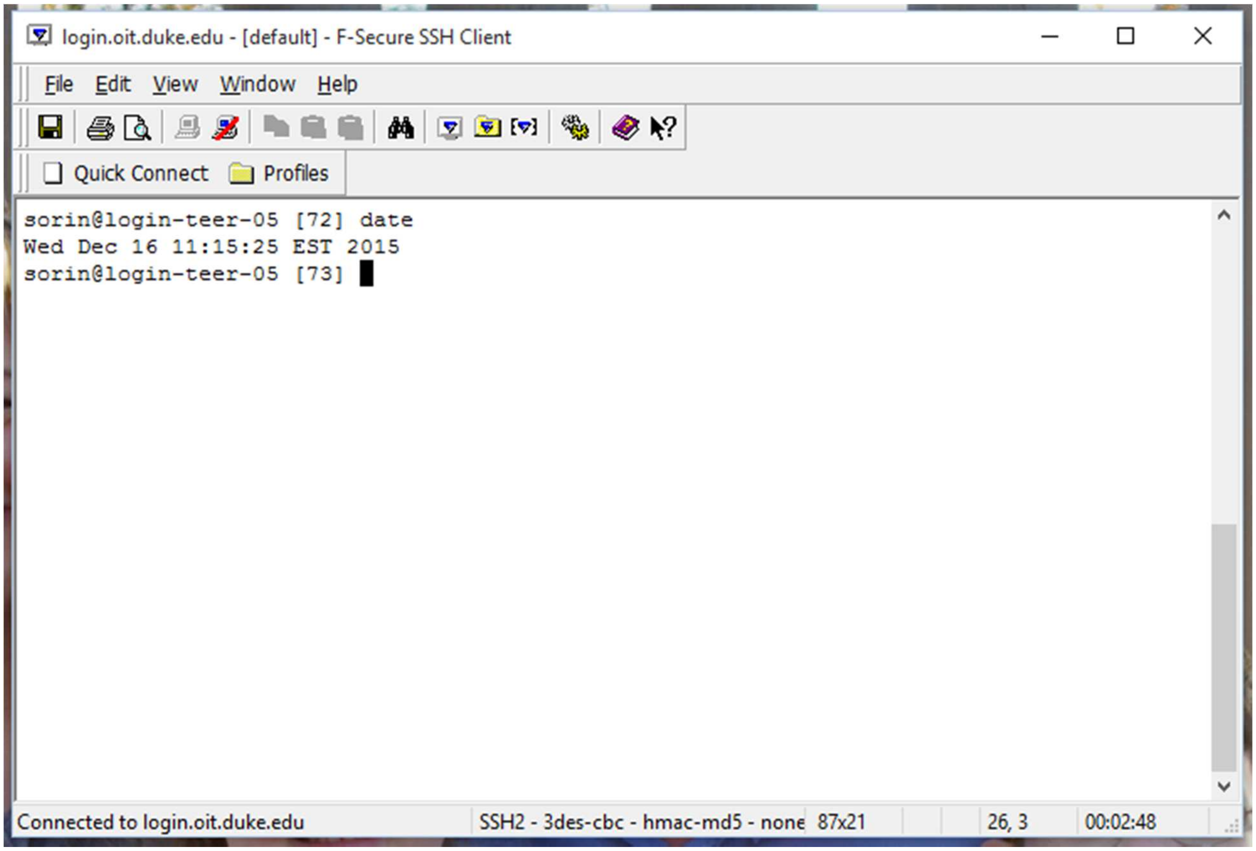
1. Logging in to a Unix machine

Duke maintains a cluster of x86/Linux machines in a *top-secret* location. Fortunately, through the magic of networking, we can use them from wherever we are. To access them, we need to use a secure shell (SSH) client, and it can also be useful to use X-windows so that we can run programs with graphical user interfaces (GUIs). Below are instructions for installing/running X-windows and SSH.

Mac	**Windows**
Installing: X-Windows (To display GUI xterm screens)	
1) Download and install Xquartz from http://xquartz.macosforge.org. 2) Logout of your MAC 3) Login to your MAC	1) Download and install x-win32 from the OIT software website at https://software.duke.edu/node/197 You should be sure to download the latest version and follow the instructions in the PDF for licensing. If you want to use X-Win from off campus you should use the Activation License not the License Server.

Installing: Secure Shell (SSH)	
Secure shell is easy on a Mac since it is built into the Terminal Application 1) Open the Terminal App. You can find it in the Applications/Utilities folder or by searching in Spotlight for Terminal 2) At the command prompt, type `ssh –X netID@login.oit.duke.edu`, where netID is your Duke NetID. This command initiates a secure shell connection to a Linux machine in the cluster. 3) Enter your password.	1) Download and install PuTTY from the OIT software website at https://software.duke.edu/node/329 2) Start x-win32 (i.e., run the program) 3) Open a PuTTY terminal window a. The first time, you should get a configuration screen. For Host Name put in login.oit.duke.edu. b. Connection type should be SSH and Port should be 22. c. Go to Connection category, open the SSH option, and click X11. Ensure that the check box next to X11 forwarding is checked. 4) You can save a session for subsequent use by giving it a name and saving the session. Then you can later reload the session by selecting it and clicking “load”. 5) Click Open to start the PuTTY session. This will open a Terminal Window and prompt you for your NetID (i.e., login as:). 6) Type your Duke NetID. 7) Enter your password.
Congratulations – you have now successfully connected to a remote Linux machine!	

You now have a terminal session that is connected to login.oit.duke.edu. You should see a command prompt that is something like [netID@login-<something>]. Your command prompt may be different (might have a “$” or a “%” at the end, etc.), but that doesn’t matter. At this prompt, type `date` (and then hit enter/return, as you have to do after all commands on the command line). This Unix command displays today’s date, as shown below. (Note that I use a different SSH client, but that doesn’t change what happens on the command line.)



login.oit.duke.edu - [default] - F-Secure SSH Client
File Edit View Window Help
Quick Connect
Profiles

```
sorin@login-teer-05 [72] date
Wed Dec 16 11:15:25 EST 2015
sorin@login-teer-05 [73]
```

Connected to login.oit.duke.edu | SSH2 - 3des-cbc - hmac-md5 - none | 87x21 | 26, 3 | 00:02:48

Now type `xterm` at the prompt. This should open a window on your machine's screen that gives you another terminal on the remote machine. If this fails, you've done something wrong in setting up X-windows; please review the steps above according to your operating system.

Click in the xterm window to activate it. Close the xterm window by typing `exit` at the command prompt.

2. Using a Unix machine

You're going to be using Unix machines for your C programming, and thus you're going to need to be somewhat adept at working with these machines. Please work through Tutorials #1-#5 at the following website. The subsequent tutorials are also useful but not nearly as crucial.

http://www.cs.duke.edu/~alvy/courses/unixtut/

Note that the link to info on CIFS (right near the beginning is broken). Please use these instead:

https://oit.duke.edu/help/articles/cifs-home-directories-and-web-spaces-how-connect-windows http://oit.duke.edu/help/articles/cifs-home-directories-and-web-spaces-how-connect-mac-os-x

3. Using a Text Editor

You're going to be writing programs using a text editor. There are many options, including: emacs, pico, nedit, gedit, vim, etc. Students in ECE/CS 250 tend to prefer nedit, but the choice of text editor is a personal preference (with strongly held religious attitudes). I personally think emacs is the best thing ever, but I'll try not to think less of anyone who uses pico instead. ☺

Big picture: you need to get comfortable using one text editor. To do so, create a file with one of them. If you already know some C or C++, write a program in that language. If you want to write a (simple) Java program, do that. Just make sure that you can create, edit, and save a file. To start editing a file with pico, you can type the following at the command line:

```
pico hello.c
```

This line will start pico for use in editing a file called `hello.c`. If that file already exists, it will be opened for editing. If it doesn't already exist, a new file with that name will be created and opened for editing.

Once the file is open, please type some stuff—C code, if you know some, but even gibberish is fine for now—and then save the file and exit the editor. Now from the command line, type the following command to list the files in the current working directory: `ls`

You should now see hello.c (and other files).

4. Beyond This Recitation

Duke has developed a much more thorough tutorial on using Unix machines, and this tutorial covers far more than is necessary for this class.

```
https://extend.duke.edu/courses/course-v1%3AInnovationCoLab%2B996010%2B2017/
```

I highly recommend it, particularly if you have an interest in using serious computers and/or interviewing for positions that require a basic understanding of how to use serious computers. Software jobs often require that you look like you know what you're doing with a Unix machine.