ECE 118 – Introduction to Unix

What is Unix?

- Unix is the name for a **family of operating systems** that originated from AT&T Bell Labs in the 1970's
    - Linux, **FreeBSD** (the one we use for Rabbit), and MacOS are examples of operating systems within this family.

How do I use Unix?

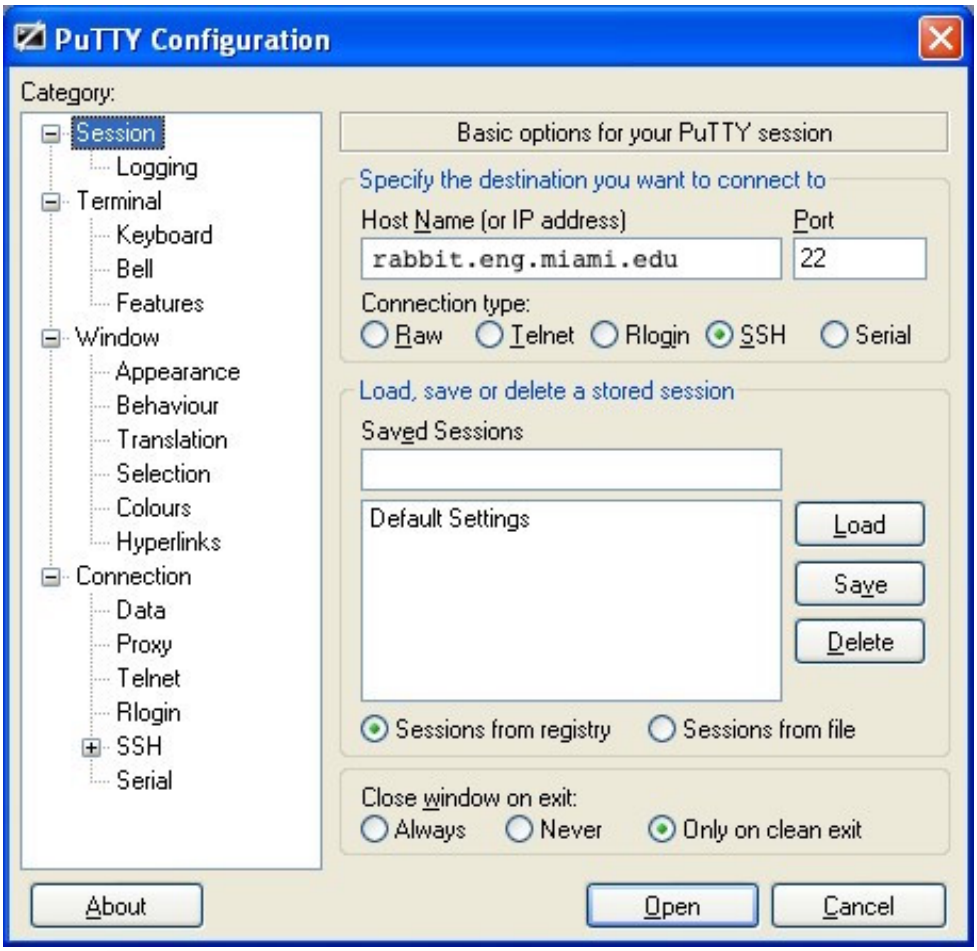
- The ECE department has a **server** computer (that you know as Rabbit) running **FreeBSD** on which you all now have an account. With this account, you can log onto the server. Your account gives you access to the Unix server via a **shell**.

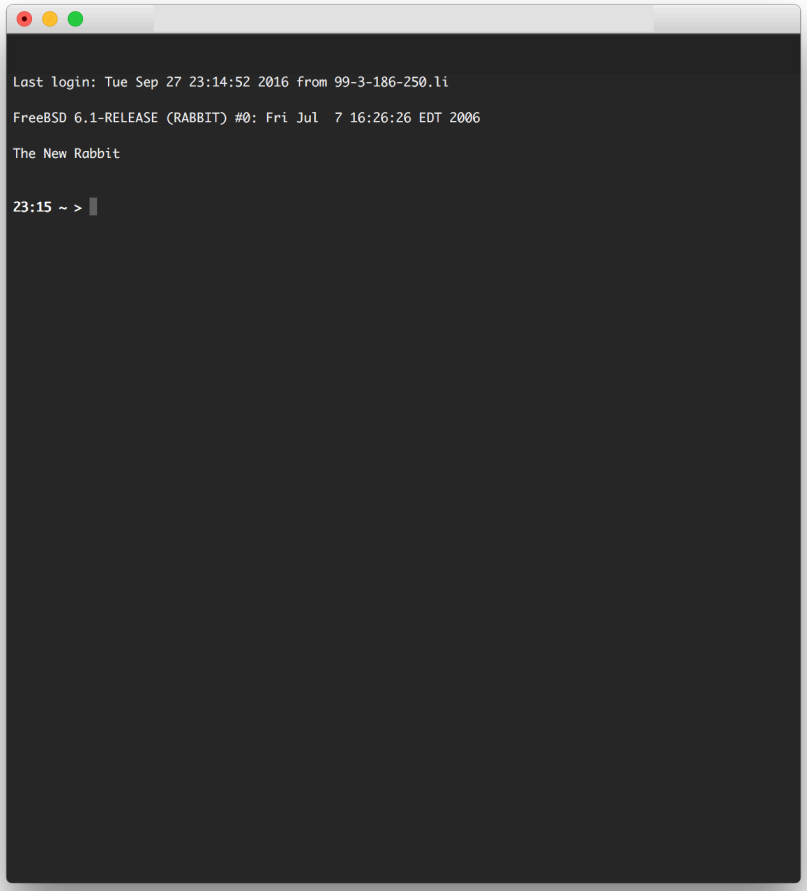
What is a shell?

- A shell is a graphical or command line interface that allows a user to interact with the operating system. For example, the windows shell has a desktop (with folder icons and shortcuts), a taskbar (with a start button, shortcuts, etc.), and other things. Putting it simply, **a shell is your interface with the computer**.
- There are lots of different kinds of shells (bash, tcsh, etc.), but you don't need to worry about that in this class.
- When you connect to the server computer with your account, you will instantiate a new shell (just as you would when you log onto your laptop, this one just looks different)

How do I log on?

- You have been given an account that most likely consists of the **first letter of your first name followed by your entire last name** (e.g. Sarah Smith would be "ssmith"). **Your initial password will be your c-number** (e.g. Sarah's password is C12345678).
- If you have a **Mac**, you can open the application **Terminal** (notice at the top of terminal when you open it, you will see "bash" – that stands for Born Again SHell).
    - Enter the following: `ssh <username>@rabbit.eng.miami.edu`
    - When prompted for your password, type it and press enter. **You will not be able to see your password as it is being typed, but rest assured it is in fact there**
- If you have **Windows**, you will need to download a telnet client like PuTTY (← click that or Google search PuTTY)
    - Make your configuration look as it does in the picture on the following page (host name = rabbit.eng.miami.edu; port = 22; connection type = ssh) and press open
    - If you did this correctly, you be prompted with `login as:`
    - Enter your username only (no @rabbit.eng.miami.edu) and then enter your password in the following prompt (**again, it will not be visible as you enter it)**
- Note that "ssh" stands for Secure SHell since you are instantiating a new, secure shell with the server in order to connect to it


PuTTY Configuration
Category:
Session
Logging
Terminal
Keyboard
Bell
Features
Window
Appearance
Behaviour
Translation
Selection
Colours
Hyperlinks
Connection
Data
Proxy
Telnet
Rlogin
SSH
Serial
Basic options for your PuTTY session
Specify the destination you want to connect to
Host Name (or IP address)
Port
rabbit.eng.miami.edu
22
Connection type:
Raw
Telnet
Rlogin
SSH
Serial
Load, save or delete a stored session
Saved Sessions
Default Settings
Load
Save
Delete
Sessions from registry
Sessions from file
Close window on exit:
Always
Never
Only on clean exit
About
Open
Cancel


Last login: Tue Sep 27 23:14:52 2016 from 99-3-186-250.li
FreeBSD 6.1-RELEASE (RABBIT) #0: Fri Jul 7 16:26:26 EDT 2006
The New Rabbit
23:15 ~ >

Now what?

- Now that you have logged onto the server computer using a secure shell, you may begin to **program**
    - Note that up to this point, we have used Microsoft Visual Studio for programming. MS Visual Studio is an **IDE** (integrated development environment). What this means is that you can develop, compile, and debug code all in the same environment. **Your Unix shell is not an IDE**
- Your Unix shell is a command line interface with the FreeBSD OS of the Rabbit server. This means **you may run commands by typing them out and pressing enter**. Every command is simply a function that the OS understands and has stored in memory. Often, when you type a command, you will include certain arguments after the name of the command; you may think of these as the parameters to the command/function you are running
- **If this is your first time logging in, change your password**.
    - You may do this by entering the command: `passwd`
    - You will be prompted to enter your old password and then the new one of your choosing. Your password won't be displayed back to you (echoed).
- Unix file systems are organized in a hierarchical manner, such that **everything belongs under the root directory `/`**.
- Note that, upon login, you are currently in your home directory `~`. This directory really exists at `/usr/home/<user name>`

Useful Unix Commands

- `passwd` – allows you to change your password
- `whoami` – echoes information about you as a user
- `man` – entering `man <command>` will open up the manual page for using that command. You may exit the manual when you are finished by pressing `q`. For example, `man man` will open up the manual page for the `man` command
- `ls` – lists the contents of the current directory. Note that some contents may be hidden, you can use the manual page to find out how to view what is normally hidden
- `pwd` – displays the full path of your current directory
- `mkdir` – creates an empty sub-directly within your current directory with the name following `mkdir` (e.g. `mkdir Lab5` will make a new, empty directory called `Lab6` within your current directory)
- `cd` – enters the directory whose name follows `cd` (e.g. `cd Lab6` will move you into the **existing** directory `Lab5`)
- `rm` – entering `rm <file>` will remove the file from the current directory
- `rmdir` – entering `rmdir <directory>` will remove the **empty** sub-directory
- `exit` – logs you out of your current shell
- You may look up other standard Unix commands online (or on Rabbit), but this should be good enough to get you familiar with your account. Now, onto programming…

Pico

- `pico` is a command that opens up a **text editor** wherein you may create and edit the contents of a file (such as a .cpp file). If I want to create a C++ source code file called `source.cpp` (note that I include the file extension as part of the name, in order to avoid confusion with anyone else or myself who will use this file in the future, this is good practice), I simply enter the command: `pico source.cpp`
- If `source.cpp` does not yet exist, pico will first create the file for me in my current directory. If the file does exist, it will allow me to edit the file. Below are a list of useful shortcuts to use within pico (**note that I will denote pressing the Ctrl key as ^**)
    - `^X` – exits out of pico and returns you back to your current directoy
    - `^Y` – pages up in your file
    - `^V` – pages down in your file
    - `^K` – cuts an entire line of text. You may cut multiple lines by pressing this consecutively, but the lines must be next to each other. If you cut some lines and then move in the file and cut more lines, the first set of cut lines will be lost
    - `^U` – un-cuts (pastes) the most recently cut lines
    - `^W` – allows you to search the file
    - `^O` – serves as a "save as" function. This isn't used too often, though.
- Once you have written out your code in `pico`, you will need compile and run your code. Remember, your Unix shell isn't an IDE, so you can't perform those two things with `pico`
- `^X` will exit you from `pico`. If you have made any changes to the file on which you were working, you will be prompted to save the file or not. `^Y`, `^N`, and `^C` give you the option to save, not save, or cancel exiting `pico`, respectively. When you enter `^Y`, you must enter the name you wish to save the file as. By default, the name you originally gave the file is showm. Press `enter` if you want to overwrite your file (that is, save it in place)

CC

- `CC <filename>` compiles the file into a runnable program (by default, this program will be called `a.out`)
- More information about `CC` can be found under the "Guide to using Unix" on Rabbit (found in the middle column on the home page).
- When you have successfully compiled your program, simply enter `a.out` to run your program

Pine

- Entering `pine` will run an email utility program for which most of the navigation is fairly self-explanitory within the pine interface.

*Note that more comprehensive information on Unix can be found on Rabbit under "Guide to using Unix" (currently under the middle column of the home page)