

# The Command-Line Interface (CLI)

John D. Muccigrosso

Tuesday, 14 February 2017

## Using the Command Line Interface at Drew

### Getting there

1. Log into <http://www.cloudpc.drew.edu/>.
2. Load *Putty* as one of your applications, if it isn't there already.
3. Run *Putty*. (You may have to install the Citrix cloud viewer your first time.)
4. In the “Host Name” window, type `pandoc2.cs.drew.lan`.
5. For convenience, type a name into the “Saved Sessions” field and click the “Save” button to have it remember the server name. (You can also go in and change various settings for your session. Most won't make sense, but under “Window”, “Window/Appearance”, and “Window/Colours” are some options you can play with.)
6. Click the “Open” button or double-click the session name you saved to log in using your usual Drew username and password.

### Intro to the CLI

1. The **command-line interface** is another way (pre-1984 Apple Macintosh) to access your computer (or a remote one, in this case). It's also called the **terminal** after the machines (called “terminals”) that were used to access big mainframe computers back in the 20th century.
2. You must type in commands to do things, including running programs or looking at files and folders.
3. There's no **undo**, so be careful!

### Using commands

1. Type a command and hit the return key.
2. Try `date` and `whoami` to see what output you get.

### Playing with the file system

### Directories (a.k.a. folders)

1. Whenever you're logged in, you are located in a folder on the computer, but we're going to call those folders *directories* now.
2. Where are you now? Try `pwd` for "present working directory".
3. Are there files in this directory? List them with `ls`.
4. Let's get some more info on them by adding an option (or switch or flag) to the command: `ls -l`.
5. Let's make another directory inside this one (a "subdirectory") and call it "practice": `mkdir practice`. (Guess what `mkdir` is short for.)
6. Let's change the directory we're in to that new one: `cd practice`.
7. Let's list the files here with the more complete method `ls -al`.
  - There are no files yes, but still you can see two entries with dot names. Those are shorthands for the current directory (`.`) and the directory that contains it (`..`).
  - So you can use the shorthand to move back to the containing directory: `cd ..`
  - Now go back to the "practice" directory.

## Files

1. Let's make a new file in "practice". There are lots of ways to do this, but the command `touch` will update the modification date on any file, creating it if it doesn't yet exist, so try `touch temp.txt`.
2. Copy it via `cp temp.txt newtemp.txt` and rename it via `mv newtemp.txt test.txt` (`mv` is for "move"). Notice how the first name is the existing file and the second is the new one. That happens a lot.
3. Now remove it completely with `rm test.txt`. NB No undo!
4. There are several ways to edit files within the CLI. Easiest perhaps is with the command `nano <filename>`. This will open the file `<filename>` in a simple text editor. You can move around with the arrow keys and when you're done, use the commands at the bottom. Control-X will exit and if you've made changes, it will ask if you want to save them and then with what name.
5. To look at your files, you also have several options:
  - `cat` (for "catalogue") will scroll the entire file across the window without stopping.
  - `head` will show the first 10 lines of the file.
  - `tail` will show the last 10 lines.
  - `less` will show you one screenful at a time until it's gone through the whole file. Use the spacebar or arrow keys to move, and Q to quit at any time.

## Handy commands

1. `man` will show the how-to file for any command (like a "manual").
2. `history` will show the last several hundred of your commands. Use `!` followed by the number of the command you'd like to run again to re-execute it, e.g., `!234`.

### Other goodies

1. The `tab` key will try to autocomplete a filename or command when it can.
2. Hitting the up arrow will bring up the previous command. It's like stepping through `history`.