Linux (Ubuntu)

After install you will find it's much like any windowing OS, although without the polish (for instance pinta, my MSpaint replacement, does not have a print function, openOffice and Libreoffice are not as good as MS Office). Desktop icons are a bit of a pain to make.

But its fast, free, customizable (different types of linux, different window managers (or none for servers) , setup for client use or server use, can be run from just the command line (low demand on your video card then) so you can ssh in to a remote machine and run the OS through a low bandwidth connection (just a few chars back and forth over the wire verses streaming all the graphics of a window manager over the wire).

To install software you need extra privileges, so most of the following needs to be preceded by sudo. You can get/install programs directly(like eclipse) but usually you'll go through package managers, the software may be a little older than the latest available, but its easier to install)

The following is from http://www.control-escape.com/linux/lx-swinstall.html

Debian, Ubuntu: APT

There is a broad array of tools for working with DEB packages, but the one you will commonly use is `apt-get`, arguably the easiest of Linux package management tools. `apt-get`is so easy because it not only keeps track of what packages are installed, but also what other packages are available. It will even download them from the Internet for you (if properly configured).

```
sudo apt-get install ${packagename}
```

To remove software is just as easy.

```
sudo apt-get remove ${packagename}
```

Although the repositories that contain installable packages might live on the Internet or on a disc somewhere, APT keeps a local database with a list of all available packages and where to find them. This database needs to be explicitly updated. The following command will update the APT database:

```
sudo apt-get update
```

A common idiom is to update your package database, and then upgrade all the packages that have patches or security updates to install. The following command will do this all at once.

```
sudo apt-get update; apt-get upgrade
```

For a more indepth `apt-get` tutorial and other resources, see ***Managing Software with APT and dpkg***.

Open terminal (can do everything from here)
Ctrl-alt-T

Environmental vars
echo $PATH #where the os looks for executables and other stuff if you do not specify a path

some commands
cd (dirname) #change directory
cd ~ #go to home dir /home/keith for me
ls –la #directory listing with permisions, owners, hidden files
`find . -name "*boo*"  #recursively search for any file with boo in name`
df #disk usage
ps –e #all running processes
kill processnumber #from ps –e, choose process number and kill with this command
alias #way to simplify complex commands (demo alias and cd327)
grep -R cos . #recursively search all files for the letters cos from this directory on

especially useful stuff
locate
which
whatis
whereis
./executableprogram #run program from current dir

See linux tutorials on course home page for help. My advice, just start using it, over time you will learn all this stuff.

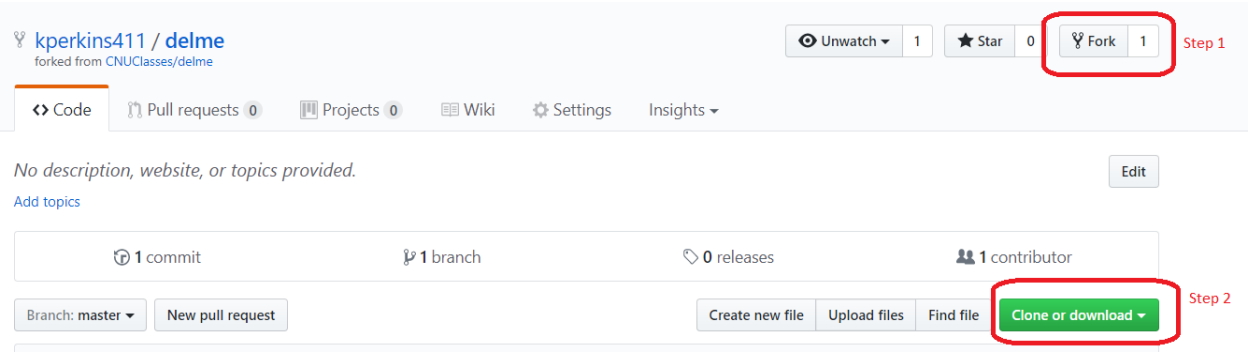
Git

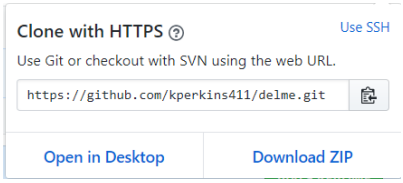
See course website for simple guide to git

Git keeps a local copy and remote copy, manages most of merges for you, very fast. You will use version control at your job, probably git, learn it now. This section follows an in class demo

Workflow (pull from my repo at github.com to your github.com account)

First fork the repo, then clone





Use above link to get a local copy

On your local machine,

git clone https://github.com/kperkins411/delme.git #make a local copy of remote repo

Workflow (create your own local repo, push to your github.com)

git init #now have a local repo with .git file

touch .gitignore #put all the files/dirs. You want to ignore in here

git remote –v #connected to a remote? not yet

#go and create a remote repo, I created one at github called delme with url

#https://github.com/CNUClasses/delme.git

#now tie local repo to remote

git remote add origin https://github.com/CNUClasses/delme.git

Workflow (change local repo, push to your github.com repo)

#create a local file
touch file
git status #see gits view of things
git add file #or use interactive (great if you have many files that you cannot filter with .gitignore)
git add –i
    #probably use update (2) and add untracked the most(4)
    #just choose the files you want to add from list, then hit return on empty line to quit
    #presto you have a staged change to commit, but its not in yet
git status
git commit –m “informative message” #makes local repo changes
git log --oneline #what have I commuted
git push origin master #pushes those changes to remote

In depth tutorial at bitbucket (along with free private repos)

https://www.atlassian.com/git/tutorials/learn-git-with-bitbucket-cloud