

CS206 Lab0 Unix and Eclipse

The Unix operating system consists of three parts: the kernel, the shell and the application programs. The kernel is the heart of the operating system, it allocates processor time and memory, handles file operations and user tasks. The shell acts as an interface between the user and the kernel. The shell is what you are typing to at the prompt after you log in.

Part 1: Be sure you are familiar with all of the following UNIX commands

- Linux/Unix basics
 - o Login
 - o Commands:
 - ls
 - Displays all the files in the directory.
 - Example: ls 206
 - Example: ls
 - mkdir
 - Create a directory
 - Example: mkdir cs206
 - cd
 - Change directory
 - example: cd cs206 (change to the directory cs206)
 - Example: cd .. (change to the directory one above where you are, for instance if you are in /home/xena/cs206, then cd .. would put you in /home/xena)
 - Example: cd with no argument
 - pwd
 - Displays the current working directory
 - o Paths
 - The file system groups all files together in a hierarchical tree structure.
 - The top of the hierarchy is traditionally called the root.
 - Root is denoted by /
 - Any path name that starts with a / is a so-called absolute path name
 - Traversing the path, e.g. whatever pwd returned:
 - cd
 - ls
 - cd ~
 - o File related commands
 - cp
 - Copies a file from one location to another.
 - Example: cp /home/gtowell/206/data/l01/test.txt test.txt
 - Example: cp test.txt test2.txt
 - mv

- Move or rename files and directories
 - mv test.txt test3.txt
 - Moves a file from one location to another. Works across directories.
 - cat
 - spew an entire file (or files) to the screen
 - cat test2.txt
 - cat test2.txt test3.txt
 - less
 - paginate file to screen
 - less text3..txt
 - rm
 - rm test2.txt
 - **Unix does NOT have undelete!!!**
- Unix FAQ
 - <http://www.faqs.org/faqs/unix-faq/faq/part1/preamble.html>
 -

Part 2:

Make a simple hello world program in java.

In your home directory, use the Unix commands above to do the following:

```
make directory cs206
switch to directory cs206
make directory lab01
switch to directory lab01
```

Using your favorite editor write a hello world program in java. Here is the entire program:

```
public class HelloWorld
{
    public static void main(String[] args)
    {
        System.out.println("Hello World");
    }
}
```

Store the program in a file named HelloWorld.java and be sure that this file is in the lab01 directory you just created. If it is not there, find it and move it. Finding is usually a process of changing directories (cd and using ls).

Compile and execute your program

```
javac HelloWorld.java
```

java HelloWorld

Part 3 Using Eclipse for Java:

1. Start Eclipse
 - Menu: Applications / Programming / Eclipse
 - Command line: eclipse &
2. You should be asked to "Select a directory as workspace". Say you want it in "/home/YOU/cs206" (replace YOU with your login name). Also, check the box "Use this as default ...". Finally click on "Launch"
3. If you see "Welcome" (you should) in upper left tap in the 'X' next to it
4. Create a new project
 1. File menu > New > Java Project
 2. In popup window name you project HelloWorld2
3. Tap on "Finish"
4. Notice in left "Package Explorer" panel there is now "HelloWorld2" highlighted in green
5. Right click on "HelloWorld2" then select "New > Class"
6. Fill in the name for your class, use "Main". Check that you would like a method stub for public static void main and Comments. Click on "Finish".
7. Notice in the center window that the class Main has been created and it has an almost empty instance of the main function.
8. Replace "//TODO ..." with System.out.println("Hello world again"); As you type note that eclipse offers helpful suggestions and documentation.
9. Run your program. Tap on the circular green ">" (just below Refactor)
10. Find the output in the panel at the bottom of the screen

Another class (in the same project) in Eclipse:

1. Create a second class within the "HelloWorld2" project. (Right click on the project in the package explorer, then "New > Class". Name the class something new (e.g. Main2).
2. Add a main method to this new class and have it print something other than "Hello world again".
3. Experiment with Eclipse to understand when eclipse executes the main function of each class.
4. Change the name of the main method in Main2 to main2
5. Now run. What happens?
6. Change the main method of the first class to call the main method of the second class. If you did everything above, then this line should work

```
Main2.main2 (null)
```

Wrapping up in the Terminal

Eclipse creates a directory for every project. Find those directories. This should be as simple as

```
cd /home/YOU/cs206
ls
```

Notice that the HelloWorld2 directory has 2 subdirectories: bin and src. Look in each. Seek understanding for what each directory contains.