Topics to cover in Linux Demo

<Notes here are rather sparse, will flesh out more details during class>

Linux is a multi-user operating system. Many users can be logged in at once and share the machine resources. Unless there are a lot of processes running it usually appears to each user as if they have the machine to themselves.

The linux shell is a command-line environment. There are many shells, but here we will assume you are using bash (bourne again shell). While some things are more complex or difficult to do in a command-line environment compared to a GUI environment, others are faster and more efficient.

Unlike Windows, which has a rather finite built-in set of commands (like copy, paste, etc.), in Linux each "command" is basically running a program. For example, "ls" runs a program that lists the files in the current directory. "g++" runs the C++ compiler. "nano" runs a text editor. Most of these programs take input arguments to control how they run, just like you send arguments into a method in Java. This makes Linux nicely extensible because we can add new programs to our repertoire.

Look at http://linuxcommand.org tutorial:

1. Changing your password

   passwd <new password>

2. Understanding the file hierarchy (compare to Windows and C:\)

   The root is /
   Executable files are in /bin
   Users directories are in /home/

3. Navigate using "cd" and "pwd"

   Navigate using ~ and user home directories

   Idea of "." and ".."

   Using tab completion

4. Using the "ls" command

   ls -l to get file details

   Use the man command to get more help on a program

5. Viewing files with cat or more or less

6. Using wildcards, *

7. Editing with nano or vi

8. Copying a file with cp

9. Deleting a file with rm

10. Renaming or moving a file with mv

11. Copying a file to local machine, using WinSCP

12. Redirecting input to a file using > or from a file using <

13. Piping the output of one program into another program

    ```
    ls -l  | more

    cat words.txt | sort -r

    cat words.txt | head -10

    cat words.txt | tail -5

    ps aux | more

    ps aux | grep kjmock

    du | sort -nr | more
    ```

14. We can run GUI apps, under a client/server model. Need an X server on the client end. Do demo using mobaXterm and run programs like xterm, xeyes, firefox, subl

15. Process management

    Stop a program, control-c

    Pause a program, control-z

    Run in the background, bg

    Return to a program, fg

    Start a program in the background with &

    List of running or stopped processes started by this shell, jobs

    List of running processes, ps

Example task easier to do in Linux?   (grading…)