

# UNIX Basics

*Made for the Geeks..by the Geeks*

Apurv Bhartia

PhD Student, Computer Science

# Why?

- The "UNIX Philosophy," as summarized by Doug McIlroy –

*“Write programs that do one thing and do it well.  
Write programs to work together. Write programs  
to handle text streams, because that is a universal  
interface”*

# UNIX File System

- File: unit of storage.. Could be program source code or text or an executable program.
- Directories: organized into a hierarchical structure that resembles an inverted tree.
  - Top-most parent is called the root directory (/)
  - Root, as well as any other directory can contain other directories

# UNIX File System

- Home directory - `/u/username`
- Z server directory - `/u/z/users/username`
- Absolute Pathnames
  - Begins at the root
  - `cd /u/username/blah`
- Relative Pathnames:
  - Begins at the current directory
  - `cd blah` (provided you are in the `/u/username`)

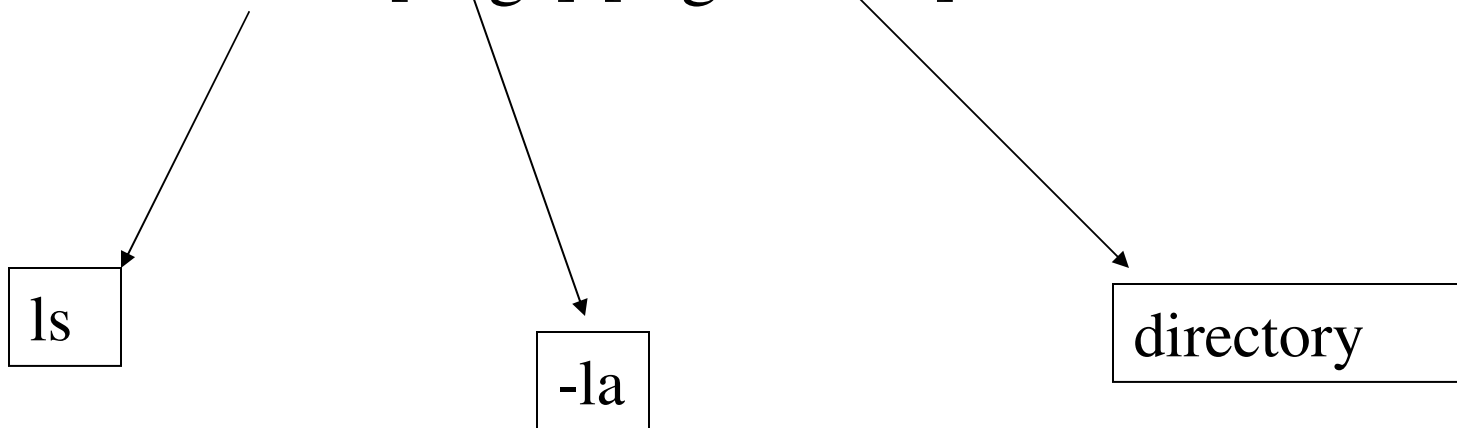
# Not without subtleties

- “.” – signifies current directory
- “..” – signifies parent directory

# UNIX Commands

- General format

– command [flags] [arguments]



‘lists’ ‘all the files’ in the ‘directory’ in the ‘long’ format

# Alright.. some commands

- pwd: **P**resent **W**orking **D**irectory
- cd directory: **C**hange 'to' the **D**irectory
- ls: **L**ist
- cat: view a file right away by automatic scrolling
- less: view a file one screen at a time
- cp oldfile newfile: **C**opy files/directories
- mv oldfile newfile: **M**ove files/directories

# Some more commands..

- `rm filename`: **Remove**
- `mkdir dirname`: **Make Directory**
- `rmdir dirname`: **Remove Directory**
- `passwd`: **Password**
- `find`: find a file (later)
- `chkquota`: check disk space (home directory)
- `du -sk`: alternative to `chkquota`

# Users, groups and file permissions

- Three *levels* of file/directory permissions
  - User
  - Group
  - Everyone else
- Three *types* of permissions
  - Read [r]
  - Write [w]
  - Execute [x]
- ---,---,--- (owner, group, else)

# Redirection

- Well.. most commands read input from *standard input* (typed by you) or produce output to *standard output* (terminal screen).
- You can change this behavior
- i.e. input from and output to *files*, rather than *standard* devices
- Use > (>>) to redirect
  - ls > foo : lists the files in the current directory and puts them in foo
  - cat filename1 > filename2: puts the contents of filename1 into filename2
  - Cat filename1 >> filename2: does the same as above.. But “appends”

# Pipes

- A pipe `|` connects the *standard output* of a command directly to the *standard input* of another command.
- `cmd1 arguments | cmd2 arguments`
  - *Pipes* the output of `cmd1` to the input of `cmd2`
- `cat foo | wc`
  - Returns the count of number of words in *foo*

# Some other commands

- grep
  - grep pattern filename: searches for the *pattern* in the *filename* and prints out each line that matches
- find
  - find . -name filename: searches *recursively* from the current directory for the *filename*
- Chmod
  - chmod ug+rw mydir: assigns read/write permissions to the owner and group classes of *mydir*

# VIsual Editor

- Call it vi
- Text editor to create new files
- vi filename
  - Opens the editor and creates/edit the *filename*
- Editor modes
  - Insert Mode
    - Allows you to edit your text file
  - Escape Mode
    - Other commands for the editor (save the file, quit, cut/copy/paste)

# Visual Editor

- Insert mode: Type 'I'
- Escape mode: Hit Esc Key
  - Save : w
  - Quit: q
  - Save and quit: wq
  - Undo: u
  - etc etc..
- Reference -  
<http://arioch.unomaha.edu/~jclark/vim.html>

# Some final points

- UNIX has a huge repository of commands, makes it extremely powerful
- If you don't get something.. google it..
- man (manual) pages are extremely helpful
  - man *command*
- Else.. TA hours.. :D