

SOFTENG 206

Software Engineering Design 1

Lecture 1: *Linux shell, files, commands*

Nasser Giacaman

Department of Electrical and Computer Engineering
The University of Auckland, New Zealand

Welcome to SOFTENG206

- Your first project-based Software Engineering design course.
- Handouts:
 - Course Outline
 - Project
 - Assignment 1
 - Due in week 4!

The next few weeks

- Week 1 [*Library searching*]
 - **Library tutorial:** 3 slots available to choose from
- Week 2 [*Linux: shell, files, processes, permissions*]
 - **LAB:** Linux
- Week 3 [*Environment, scripts*]
 - **LAB:** Help with Assignment 1
- Week 4 [*Java GUI, file&user I/O, system calls, VLC*]
 - **Assignment 1 due**
 - **LAB:** Linux with Java GUI
- Week 5 [*ffmpeg, GUI concurrency, review*]
 - **LAB:** SwingWorker
- Week 6 [*Git, ...*]
 - **LAB:** Git
 - **Assignment 2 due**
- Week 7 [*...*]
 - **Test**

Aims of this lecture

Today we will:

- Motivate why we use Linux and command line 😊
- Understand what the Shell is
- Learn how to navigate and manipulate the Linux file system
- Understand the basics of executing commands

Operating systems and Linux

- **Operating system:** the program responsible for *managing* hardware and other programs. Examples include:
 - Windows
 - Mac OS X
 - Unix (\$\$)
 - Linux
 - Unix-like, open-source, free and easy to install

- Who uses Linux?



- Dr Sheldon Cooper (The Big Bang Theory)
- TOP500: most powerful computers in the world
 - [June 2000] 28 ran Linux, 453 Unix, 17 BSD based
 - [June 2004] 291 ran Linux, 182 Unix, 1 Windows, 10 BSD based
 - [June 2014] 485 ran Linux, 12 Unix, 2 Windows, 1 mixed
 - **[June 2016] 497 ran Linux, 3 Unix**
- Android: The world's #1 smartphone platform is Linux (sorry Apple)
- In this department, we have 16-core and 64-core Linux systems
- Most servers... plus I'm using Linux right now!

Kernel and distribution

- There is one Linux “kernel”
 - This is the actual “brains” of the operating system, the “real operating system” that manages processes, memory, devices, I/O, etc
- Linux distributions
 - Includes the Linux kernel, along with a collection of other applications (e.g. media players, graphics apps, word editor, file browser, etc)
 - 100s of different distros out there!
 - Ubuntu, OpenSUSE, Fedora, ...
- Desktop environments
 - The graphical component (windows, toolbars, icons, etc)
 - e.g. KDE or Gnome in Linux, Aqua in Mac OS X, Luna in Windows XP, Aero in Windows 7
- How to get Linux?
 - Lab machines: reboot into Ubuntu
 - Live and installation CDs: free download (also on ECE IT page)
 - VirtualBox: don't need to repartition your existing system (but won't run as fast as having a real installation)

Graphical user interface

- Easy for users, interaction becomes user-friendly
- Sometimes faster to just click here and there (rather than remembering commands)
- Some tasks are naturally visual, eg painting
- Not so good to use remotely
- Can be tedious for repetitive tasks
- More computation and memory intensive

Command line

- More control, more options
- For many things is faster, no scrolling, clicking, just typing
- Great to connect to remote machines, in fact this is sometimes the only possibility to connect to some machines!
- Learning curve, requires practice, learning commands
- Great at repeating common tasks
- Can be used in combination with GUI (e.g. create a script file that launches some GUI application by setting up various options.. easier than navigating the program list)
- People will think you're brainy or a hacker

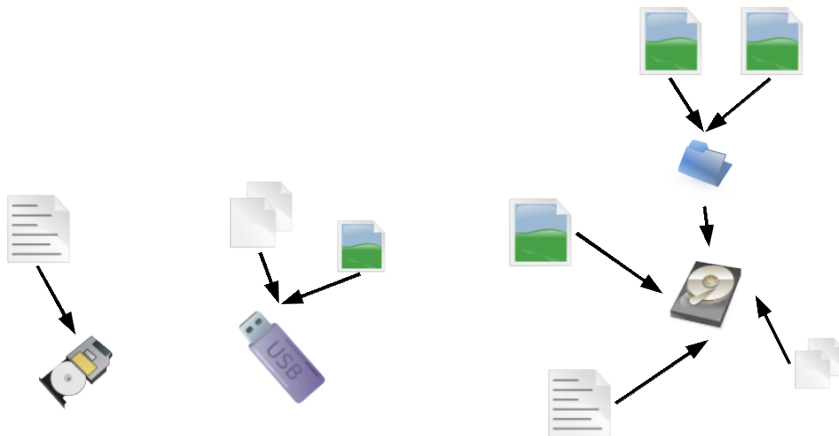


- You hardly ever see hackers in movies use the mouse!

- Sometimes command line is the only way to connect to a remote machine, especially servers, or special multi-user machines
- You can do this by using Secure SHell:
 - `> ssh ngia003@shell.ece.auckland.ac.nz`
- Try this out yourself! You should have access to this machine
- To log out when you finish:
 - `> exit`
- You can do all the things we talk about in the following lectures on that machine (or any other Linux machine you have access to)

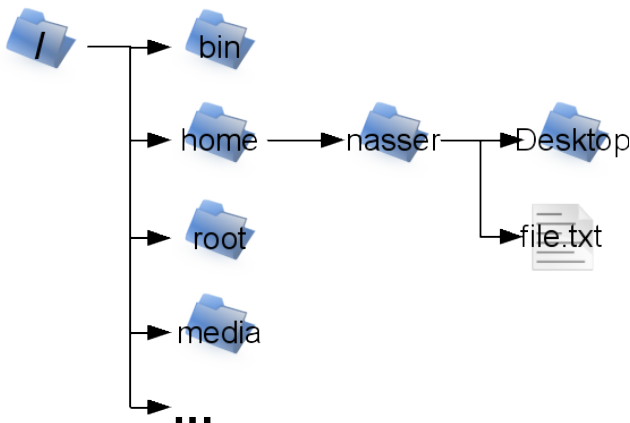
The Linux file system

- We have lots of files and lots of directories...
- ... which might be stored on multiple devices.



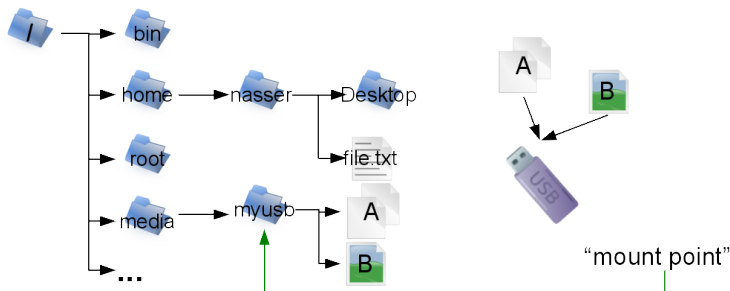
The Linux file system

- Linux has the idea of a single hierarchy of files
- There is no concept of “drive letters” as in Windows
- Appears as if all files are under one folder – the root “/”
 - Do not confuse the root folder “/” with the “root” folder



The Linux file system

- Mounting: if you want to access files from a new device, you need to specify where in the tree those files should appear
 - Nowadays, USBs, CDs, DVDs are mounted automatically
- In most cases, you will only bother with what is inside your “home”
 - `/home/nasser` (on your own system)
 - `/afs/ec.auckland.ac.nz/users/n/g/ngia003/unixhome` (@ uni)



Shells and terminals

- Shell

- The program that takes the commands you type and asks the operating system to perform them. There are different shells you can use, but we will use **bash** (Bourne Again SHell).. others: `sh`, `ksh`, `csh`, etc:

- `> cat /etc/shells`

- Terminal emulator

- The GUI application that allows you to interact with the shell
- `gnome-terminal` → Gnome
- `konsole` → KDE

- `> jkdsjfksdi`

- `bash: jkdsjfksdi: command not found`

- Command history

- up, down, left, right arrow keys

- Try:

- `> df` `> date` `> cal`

Navigation

- Print working directory
 - `> pwd`
- Change directory
 - `> cd ~` `> cd` `> cd ~username`
 - `> cd .`
 - `> cd ..`
 - `> cd -` [change to previous working directory]
 - `> cd assignments/se206/a1`
 - `> cd ./assignments/se206/a1`
 - `> cd /lib`
 - `> cd .kde` [the "." makes a file/folder hidden]
- Stacking working directory (when "cd -" is not enough)
 - `> pushd mydir`
 - `> popd`
- **Tip:** avoid spaces in file names, use *hyphens* or *underscores* instead!

Path and PATH

- A path is a set of instructions to navigate to a directory on the system
 - **Relative:** are respective to the current directory
 - `> cd assignments/se206`
 - `> cd ../courses`
 - **Absolute:** start with /
 - `> cd /media/myusb`
 - `> cd ~/documents`
- PATH is a series of paths separated by colons
 - `/home/nasser/bin:/usr/bin:/bin`
 - `> echo $PATH`
 - `> PATH=$PATH:/home/nasser/test`
 - `> PATH=/home/nasser/test:$PATH`

Viewing the file system

- `> ls`
- `> ls -l` *[long format]*
- `> ls -a` *[include hidden files]*
- `> ls -al ~` *[display contents of home directory]*
- `> ls -dl myfolder` *[details of directory, not contents]*
- `> ls -hl` *[human readable file sizes]*
- `> ls -rl` *[display in reverse order]*
- `> ls -Sl` *[sort by file size]*
- `> ls -tl` *[sort by modification time]*
- `> man ls`

Finding and viewing files

- Finding files

- `> locate some_file` *[fast search # updatedb]*
- `> find` *[include hidden files]*
- `> find -maxdepth 2` *[descend at most 2 levels]*
- `> find -type d` *[report only directories]*
- `> find -name "fruit*.png"` *[match name]*
- `> find -size +1M` *[greater than 1 megabyte]*
- `> find -size -10k` *[less than 10 kilobytes]*
- **Tip:** a 50kb file is not "less than 1M" due to rounding up
- `> man find`

- `> file filename` *[determine type of file]*

- Viewing contents

- `> cat filename`
- `> less filename` *[scroll, /ch search, n for next, N for prev, q to quit]*

Wildcards

- `*` *[any characters]*
- `?` *[any one character]*
- `[chars]` `[!chars]` `[:class:]` *[one character from set]*
- Character classes:
 - `[:alnum:]` `[:alpha:]` `[:digit:]`
 - `[:lower:]` `[:upper:]` `[:space:]`
- Examples:
 - `> echo *.java` *[files ending with .java]*
 - `> echo ??` *[files with 2 characters]*
 - `> echo Do*` *[Documents, Downloads]*
 - `> echo [![:upper:]]*` *[any file doesn't start with an upper letter]*
 - `> echo LOG[[:digit:]][[:digit:]]` *[LOG followed by 2 digits]*
 - `> echo *[[[:digit:]]abc]` *[any file ending with a digit, a, b or c]*
 - `> ls -dl [[:digit:]]*` *[display only names, start with digit]*
 - `> find [[:lower:]]* -maxdepth 0 -type d`

Manipulating the file system

- Creating directories

- `> mkdir songs movies pictures` *[create all these directories]*
- `> mkdir -p media/movies/comedy` *[create (sub-dirs) as needed]*

- Copying files and directories

- `> cp .settings .settings.bak` *[copy/paste file]*
- `> cp -i docs/*.txt dir` *[prompt before overwriting]*
- `> cp -r songs media` *[recursively copy directory]*
- `> cp -u *.java *.html source` *[update older or non-existent files]*

- Moving and renaming files and directories

- `> mv -i file1 file2 directory` *[don't overwrite if exists]*
- `> mv dir1 dir2` *[moves dir1 content into dir2]*
- `> mv src/*.class bin` *[move compiled files into bin folder]*

- Removing files and directories

- `> rm -i [[:digit:]]*` *[prompt before deleting]*
- `> rm -r movies` *[delete directory recursively]*
- `> rm -f *.class` `rm -f *.class` *[alias rm='rm -i']*

Commands

- What commands will we encounter?
 - **Executable:** compiled binaries or scripts (shell, perl, etc)
 - `> which executable`
 - **Shell builtin:** commands internal to the shell
 - `> help` `> help command`
 - **Alias:** a command built from other commands
 - `> alias`
- `> type command`
 - `> type cp` `> type date` `> type myscript.sh` *[executable]*
 - `> type cd` `> type history` `> type type` *[shell builtin]*
 - `> type ..` `> type ll` `> type dir` *[alias]*
- Creating your own commands
 - `> type test` *[is "test" already an existing command? Yes!]*
 - `> type silly`
 - `> alias silly='cd ~; ls -lSh; cd -'`