Virtualization
What is it?

- A virtual machine (VM) provides a layer, emulating one or more operating systems (guests) on top of a single operating system (host)
- Guest thinks it is running on top of hardware
- Host thinks guests are just regular applications
- Examples
  - VMware
  - VirtualBox
  - Virtual PC
  - Xen
Advantages

- Distributing a preconfigured OS
- Can take snapshots of current state
  - A rollback if there's a problem
- Easily portable
  - VM abstracts hardware and host OS details
  - Guest image is a handful of files and can be moved
- Provides a sandbox
  - Problems in guest do not affect host
- Can access remotely over network
Why Should I Care?

- Project 2 is required to be implemented on lab machines
  - You will give your demo on the assigned machine
- Optionally, you can implement in a VM first
  - Make sure to give yourself time to port to lab machine!!!
- Allows you to quickly recover from kernel errors by recovering a snapshot
  - Seconds for a snapshot recovery
  - Minutes for system reboot
- Also allows for more portability
  - Can work when another class is using the lab
VM Settings

• Use VirtualBox and LinuxMint
  – You can use other VMs / distributions, but I will only help with these

• CPU
  – Set the execution cap to 80% or lower

• Memory
  – Leave at least 1GB for host
Using VMs

• First, create settings within the VM
• Then, launch the newly created guest
  – Install image as you would a typical OS
• If you need to switch back to host, you'll need to type a special command to leave the guest
  – In VirtualBox the host key defaults to right ctrl
• Setup a means of outside communication using one or more of
  – External devices (usb)
  – Network (ssh)
  – Shared files (host)
SSH Setup

- **Guest**
  - Set VM network port forwarding rule
    - Name=SSH
    - Host port = 2222
    - Guest port = 22
  - `sudo apt-get install openssh-server`
  - `sudo restart ssh`

- **Host**
  - `ssh -p 2222 username@127.0.0.1`