Installing Linux Mint 18.1 “Serena” - Xfce as a host on your system

note that this will wipe your entire hard drive
- this is OK for the racks that you will be using throughout the course
- this is probably not OK for you personal systems
- for your personal systems, consider installing Linux Mint in a VM (virtual machine)
  - see separate instructions for how to do this

download the Linux Mint ISO (drive image)
- 32-bit: https://www.linuxmint.com/edition.php?id=229
- 64-bit: https://www.linuxmint.com/edition.php?id=230

get the ISO to a USB stick
- first, make sure that you have a USB stick that is formatted (with the FAT32 file system)
  - on Linux, there is something called **USB Stick Formatter**
- also make sure that it is large enough to contain the ISO
- next, use a USB image writer to write the ISO to the USB stick
  - on Linux, there is something called **USB Image Writer**

insert the USB stick into your system and boot it up
- this may require modifying the boot-up sequence to first boot from the USB stick
  - probably by pressing F12 or something similar
  - or by editing the BIOS (usually by pressing DEL or F2 or something similar)

once the Linux Mint ISO has booted, click on the Install Linux Mint icon
when prompted for the installation type, select Something else
- your hard drive should be referred to as something like /dev/sda
  - it will most likely not be empty
  - so select the partitions and remove them (this will destroy all data on the drive!)
  - a new drive will need a new partition table

- then, add the following new partitions to the free space
  - 1: size=2048 MB
    type=primary
    location=beginning
    use as=ext2
    mount point=/boot
  - 2: size=RAM size if RAM size >= 4 GB; 2x RAM size otherwise; e.g.,
      2 GB RAM: 4 GB swap
      4 GB RAM: 4 GB swap
      but no more than 8 GB
    type=primary
    location=end
    use as=swap area
  - 3: size=~35% of space left
    type=primary
    location=beginning
    use as=ext4
    mount point=/

4: size=~65% of space left
type=primary
location=beginning
use as=ext4
mount point=/home

/boot: stores the Linux kernel images
swap: memory swap space (also used for hibernation)
/: stores the OS, OS configuration files, and applications
/home: stores your files (application configuration, settings, pictures, etc)

the strategy is that updating the OS means wiping /boot, /, and swap
/home won't be touched
updating then takes little time (~30 minutes)

the Chicago timezone is fine (for Central)

set the user; mine is
name: jgourd
computer name: jgourd-cyberstorm
username: jgourd
password: something fairly strong
require my password to log in
don't encrypt my home folder

reboot into Linux Mint
play around!

Things to Install and/or Perform After Installing Linux Mint

these all happen in the terminal
click on the monitor icon at the bottom-left of the desktop
or click on the “start button” (the LM logo at the bottom-left of the desktop)
then, click on System and scroll to Terminal Emulator

make sure that you are connected to the Internet!

update the Linux Mint system

```
sudo apt-get update
sudo apt-get upgrade
sudo apt-get dist-upgrade
```

install Google Chrome (only for 64-bit OS)
the process is to first get the signing key that Google puts out for their repository for authentication
then, we add the Google repository and update the list of packages available
finally, we install Google Chrome

```
wget -q -O - https://dl-ssl.google.com/linux/linux_signing_key.pub | sudo apt-key add -
sudo sh -c 'echo "deb [arch=amd64] http://dl.google.com/linux/chrome/deb/ stable main" >> /etc/apt/sources.list.d/google-chrome.list'
sudo apt-get update
sudo apt-get install google-chrome-stable
```

install Microsoft core fonts (for compatibility across systems)

```
sudo apt-get install ttf-mscorefonts-installer
```

install essential build tools (for programming)

```
sudo apt-get install build-essential
```

install Vim (the ubiquitous text editor)

```
sudo apt-get install vim
```

next, do yourself a favor and add my Vim run commands
these customize Vim in a way that I like
if you wish, grab my `.vimrc` from the web site and place it in `~`
it usually downloads in `~/Downloads`; therefore,

```
mv ~/Downloads/.vimrc ~
```

cleanup the system

```
sudo apt-get autoremove
sudo apt-get clean
sudo apt-get autoclean
```

set the autorun script when opening up a terminal

we'll first copy the default provided by the OS
then, we ensure that we own it (more on this later)
finally, we'll load it into the current terminal

```
sudo cp /etc/bash.bashrc ~/.bashrc
sudo chown $USER:$USER ~/.bashrc
source ~/.bashrc
```

add my useful aliases to the autorun script (i.e., add these to ~/.bashrc)

first edit via on of the following text editors:

`vim ~/.bashrc` (text-based)
`nano ~/.bashrc` (text-based)
`gedit ~/.bashrc` (GUI)

then, add the following at the end of the file:

```
alias c='clear'
alias df='df -Th'
alias h='history'
alias h2='history | awk '"'"'{ CMD[$2]++; count++; } END
{ for (a in CMD) print CMD[a] " " CMD[a] / count * 100 "% " a; }'"'"'
| grep -v "./" | column -c3 -s " " -t | sort -nr | nl | head -n25'
alias j='jobs'
alias ls='ls -CF --group-directories-first --color=auto $*'
alias lss='ls -Alh $*'
alias m='more'
alias netstat='netstat -lnp | grep " LISTEN "'
alias p='ps -ef'
alias rm='mv -f -t $HOME/.local/share/Trash/files $*'
alias rmdir='mv -f -t $HOME/.local/share/Trash/files $*'
alias ~='cd ~'
```

finally, load them into the current terminal via

```
source ~/.bashrc
```