CSE 306 Operating Systems

Install Linux, QEMU, Kernel source, git, …

YoungMin Kwon

Install Linux on your USB drive

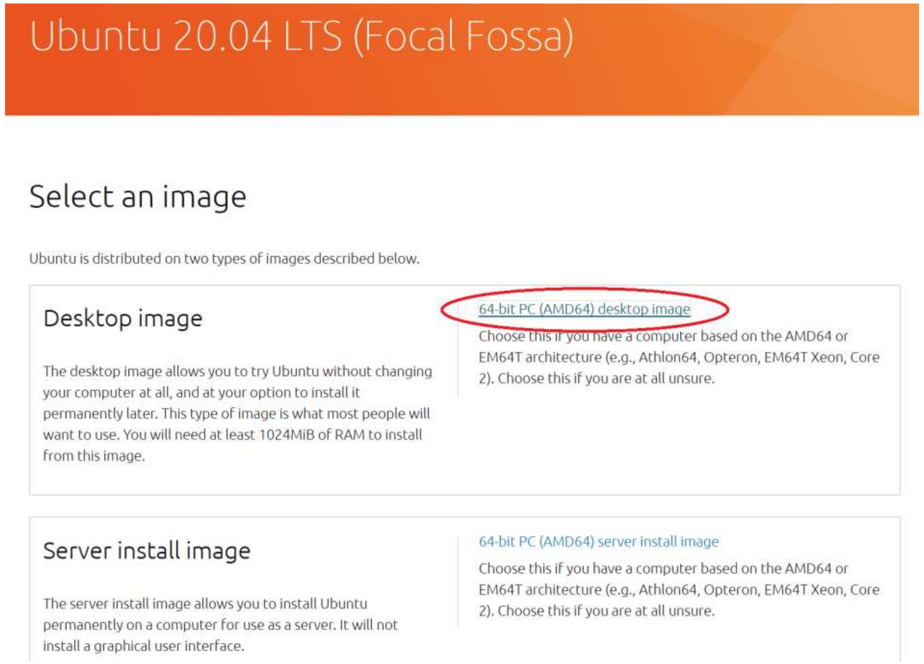
- Overall process
  - Download necessary files
  - Get your PC's network settings
  - Make your PC bootable from USB
  - Install Linux installer on USB 1 (~ 4 GB)
  - Install Linux on USB 2 (64 GB): dev machine
  - Install QEMU on USB 2
  - Install Linux on a VM (QEMU): test machine
  - Unzip Kernel source code
  - Install and configure git
  - Make a COW image (in case you break the installed Linux)
  - Build Kernel

Download: Ubuntu iso image

- Download an Ubuntu installer iso image for your development system
  - It can be different from your test system (Ubuntu 20.04), but I would recommend you to install Ubuntu 20.04
  - Download the Ubuntu 20.04 from https://releases.ubuntu.com/20.04/

Download: Ubuntu iso image

- Download an Ubuntu installer iso image for your test system
  - Downloaded the iso image at a local HDD (e.g. c:\cse306\) https://releases.ubuntu.com/20.04/ubuntu-20.04-desktop-amd64.iso



Ubuntu 20.04 LTS (Focal Fossa)

Select an image

Ubuntu is distributed on two types of images described below.

Desktop image

The desktop image allows you to try Ubuntu without changing your computer at all, and at your option to install it permanently later. This type of image is what most people will want to use. You will need at least 1024MiB of RAM to install from this image.

64-bit PC (AMD64) desktop image

Choose this if you have a computer based on the AMD64 or EM64T architecture (e.g., Athlon64, Opteron, EM64T Xeon, Core 2). Choose this if you are at all unsure.

Server install image

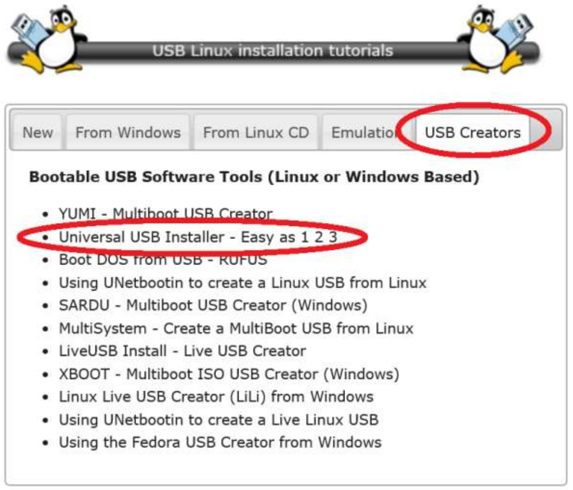
The server install image allows you to install Ubuntu permanently on a computer for use as a server. It will not install a graphical user interface.

64-bit PC (AMD64) server install image

Choose this if you have a computer based on the AMD64 or EM64T architecture (e.g., Athlon64, Opteron, EM64T Xeon, Core 2). Choose this if you are at all unsure.

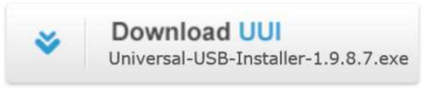
Download: USB Installer

- Download USB installer
  - https://www.pendrivelinux.com
  - Download Universal-USB-Installer-1.9.8.7.exe to c:\cse306



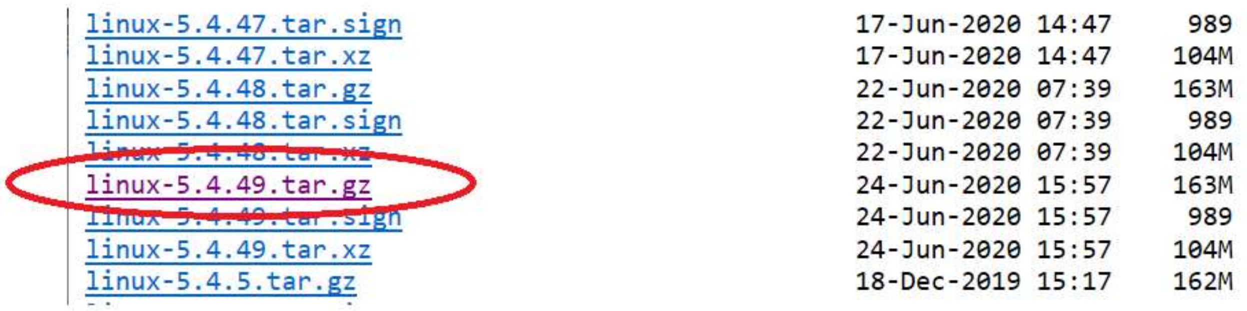
Bootable USB Software Tools (Linux or Windows Based)

- YUMI - Multiboot USB Creator
- Universal USB Installer - Easy as 1 2 3
- Boot DOS from USB - RUFUS
- Using UNetbootin to create a Linux USB from Linux
- SARDU - Multiboot USB Creator (Windows)
- MultiSystem - Create a MultiBoot USB from Linux
- LiveUSB Install - Live USB Creator
- XBOOT - Multiboot ISO USB Creator (Windows)
- Linux Live USB Creator (LiLi) from Windows
- Using UNetbootin to create a Live Linux USB
- Using the Fedora USB Creator from Windows



Download: Linux Kernel Source

- Download the Linux kernel source
  - https://mirrors.edge.kernel.org/pub/linux/kernel/v5.x
  - Download linux-5.4.49.tar.gz to c:\cse306



```
linux-5.4.47.tar.sign                             17-Jun-2020 14:47     989
linux-5.4.47.tar.xz                               17-Jun-2020 14:47    104M
linux-5.4.48.tar.gz                               22-Jun-2020 07:39    163M
linux-5.4.48.tar.sign                             22-Jun-2020 07:39     989
linux-5.4.48.tar.xz                               22-Jun-2020 07:39    104M
linux-5.4.49.tar.gz                               24-Jun-2020 15:57    163M
linux-5.4.49.tar.sign                             24-Jun-2020 15:57     989
linux-5.4.49.tar.xz                               24-Jun-2020 15:57    104M
linux-5.4.5.tar.gz                                18-Dec-2019 15:17    162M
```

Download all files locally

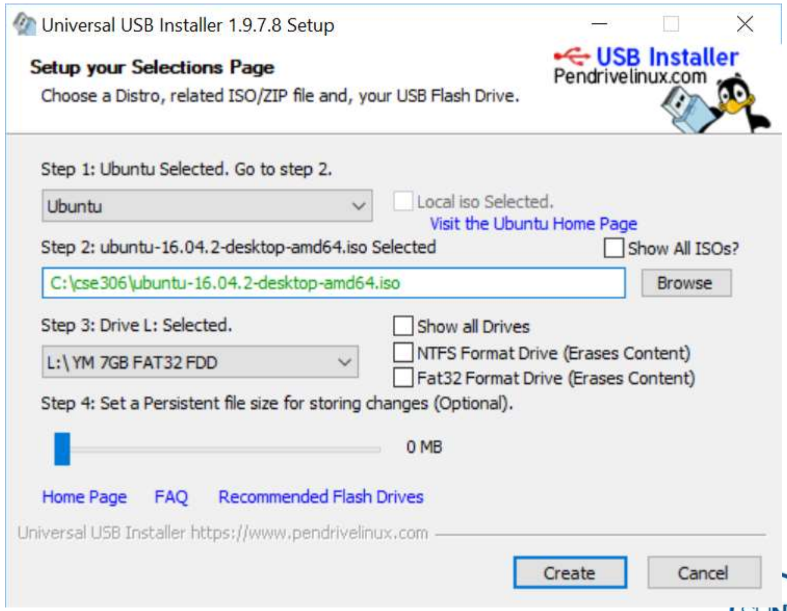
- Download Ubuntu-iso, USB-installer, and Linux kernel source locally
  - Download psftp.exe from https://www.putty.org/
    - Goto Download PuTTY
  - psftp cse306@10.12.21.61
    - Passwd cse306
  - cd linux
  - mget *
  - exit

Get the network settings of your PC

- `ipconfig /all` will show you
  - IPv4 address
  - Subnet mask
  - Default Gateway
  - DNS servers
- Write them down on a note or
- `ipconfig /all > c:\cse306\netinfo.txt`

Install Linux installer on USB1

- Insert USB 1 into your PC
- Run C:\cse306\Universal-USB-Installer-1.9.7.8.exe
  - Step 1: Select Ubuntu
  - Step 2: Click the Browse button and select the iso image for development (e.g. c:\cse306\ubuntu-20.04-desktop-amd64.iso)
  - Step 3: Select the USB drive you inserted
  - Click the Create button



Universal USB Installer 1.9.7.8 Setup

Setup your Selections Page

Choose a Distro, related ISO/ZIP file and, your USB Flash Drive.

Step 1: Ubuntu Selected. Go to step 2.

Ubuntu

Local iso Selected.

Visit the Ubuntu Home Page

Step 2: ubuntu-16.04.2-desktop-amd64.iso Selected

Show All ISOs?

C:\cse306\ubuntu-16.04.2-desktop-amd64.iso

Step 3: Drive L: Selected.

L:\ YM 7GB FAT32 FDD

Show all Drives

NTFS Format Drive (Erases Content)

Fat32 Format Drive (Erases Content)

Step 4: Set a Persistent file size for storing changes (Optional).

0 MB

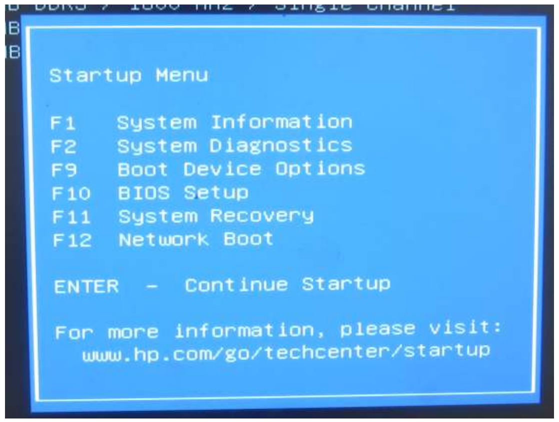
Universal USB Installer https://www.pendrivelinux.com

Enable Booting from USB

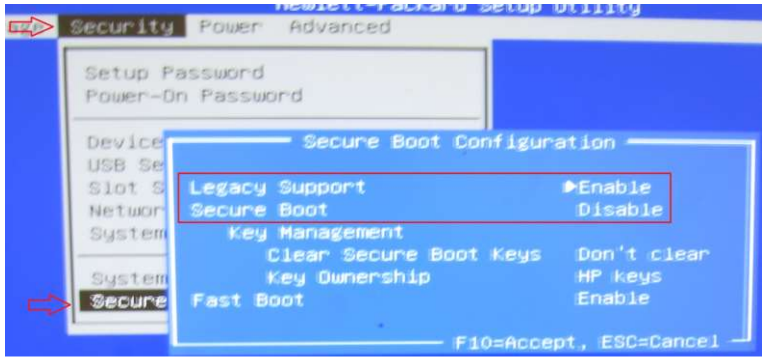
- Search Online for How to make the brand of your laptop boot from USB

- Before making any changes, email your BitLocker key to yourself

- Example (HP PCs in the game lab)
  - Restart your PC
  - Enable booting from USB from BIOS setup
    - Keep typing ESC when your PC is about to reboot
    - F10 to go to the BIOS setup
    - Disable Secure Boot
    - Enable Legacy Support
    - Save and Exit



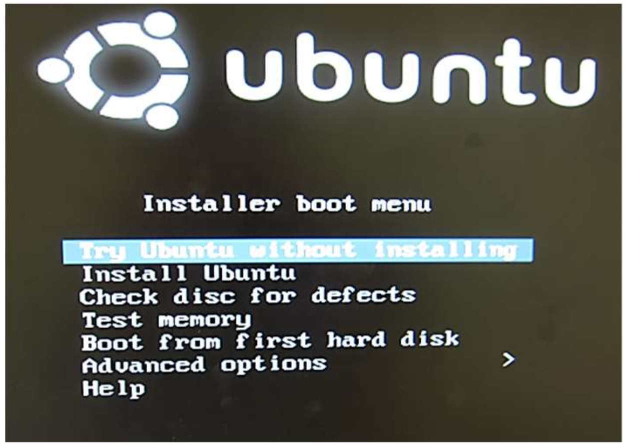
```
Startup Menu

F1     System Information
F2     System Diagnostics
F9     Boot Device Options
F10    BIOS Setup
F11    System Recovery
F12    Network Boot

ENTER  -  Continue Startup

For more information, please visit:
  www.hp.com/go/techcenter/startup
```



```
Security  Power  Advanced

Setup Password
Power-On Password

        Secure Boot Configuration
Legacy Support                          ▶Enable
Secure Boot                              Disable
   Key Management
      Clear Secure Boot Keys             Don't clear
      Key Ownership                      HP keys
Fast Boot                                Enable

                          F10=Accept, ESC=Cancel
```

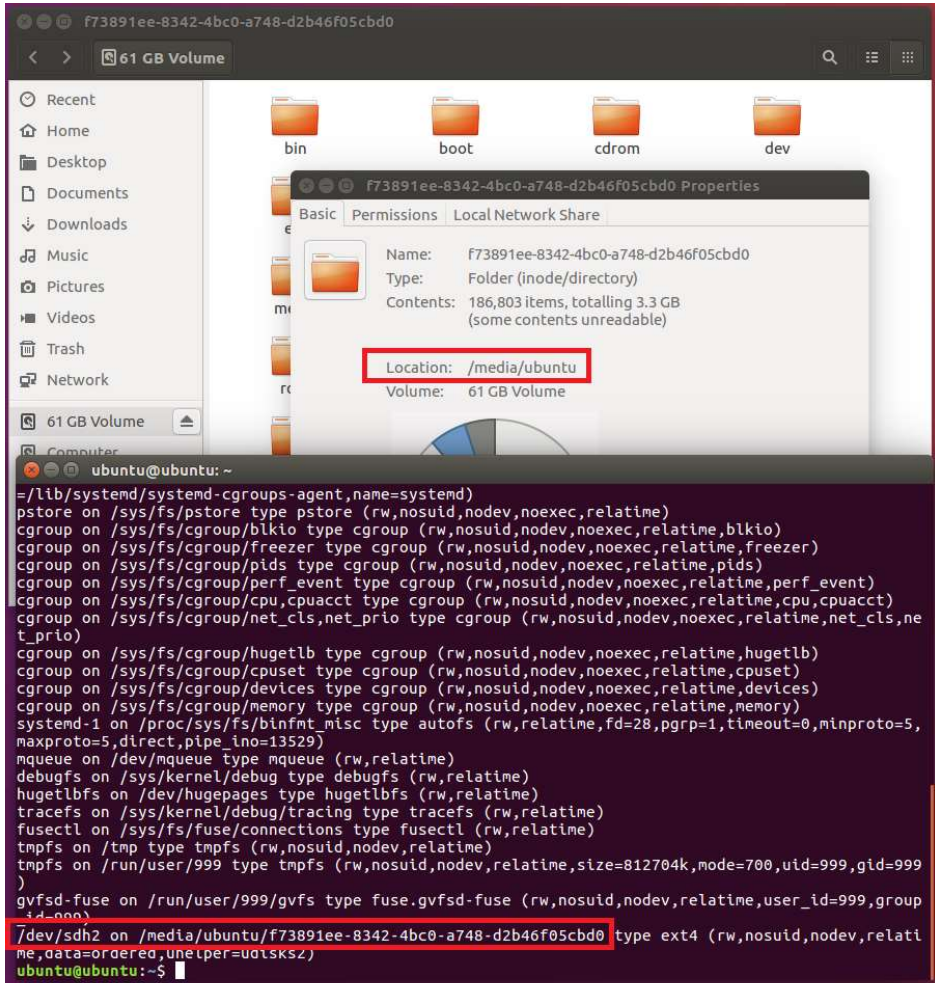
Install Linux on USB 2

- Boot using USB
  - Insert USB 1 (Linux installer) and restart your PC
  - Keep typing ESC when your PC is about to reboot
  - F9 to go to the boot option
  - Select your USB drive
- From the Installer
  - Try Ubuntu without Installing



Install Linux on USB 2

- Insert USB2
  - A file explorer will pop up
  - Right click on the highlighted item
    - 61 GB Volume in this case
    - Remember the Location info (/media/ubuntu)
  - Open a terminal and type `mount`
    - Look for a line with /media/ubuntu
    - /dev/sdh is the USB2 device



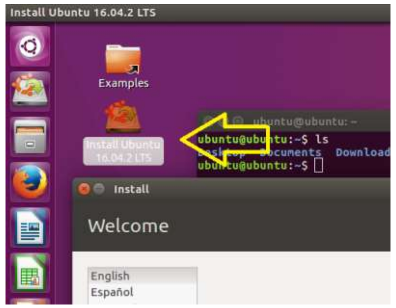
Name: f73891ee-8342-4bc0-a748-d2b46f05cbd0
Type: Folder (inode/directory)
Contents: 186,803 items, totalling 3.3 GB
(some contents unreadable)

Location: /media/ubuntu
Volume: 61 GB Volume

```
=/lib/systemd/systemd-cgroups-agent,name=systemd)
pstore on /sys/fs/pstore type pstore (rw,nosuid,nodev,noexec,relatime)
cgroup on /sys/fs/cgroup/blkio type cgroup (rw,nosuid,nodev,noexec,relatime,blkio)
cgroup on /sys/fs/cgroup/freezer type cgroup (rw,nosuid,nodev,noexec,relatime,freezer)
cgroup on /sys/fs/cgroup/pids type cgroup (rw,nosuid,nodev,noexec,relatime,pids)
cgroup on /sys/fs/cgroup/perf_event type cgroup (rw,nosuid,nodev,noexec,relatime,perf_event)
cgroup on /sys/fs/cgroup/cpu,cpuacct type cgroup (rw,nosuid,nodev,noexec,relatime,cpu,cpuacct)
cgroup on /sys/fs/cgroup/net_cls,net_prio type cgroup (rw,nosuid,nodev,noexec,relatime,net_cls,ne
t_prio)
cgroup on /sys/fs/cgroup/hugetlb type cgroup (rw,nosuid,nodev,noexec,relatime,hugetlb)
cgroup on /sys/fs/cgroup/cpuset type cgroup (rw,nosuid,nodev,noexec,relatime,cpuset)
cgroup on /sys/fs/cgroup/devices type cgroup (rw,nosuid,nodev,noexec,relatime,devices)
cgroup on /sys/fs/cgroup/memory type cgroup (rw,nosuid,nodev,noexec,relatime,memory)
systemd-1 on /proc/sys/fs/binfmt_misc type autofs (rw,relatime,fd=28,pgrp=1,timeout=0,minproto=5,
maxproto=5,direct,pipe_ino=13529)
mqueue on /dev/mqueue type mqueue (rw,relatime)
debugfs on /sys/kernel/debug type debugfs (rw,relatime)
hugetlbfs on /dev/hugepages type hugetlbfs (rw,relatime)
tracefs on /sys/kernel/debug/tracing type tracefs (rw,relatime)
fusectl on /sys/fs/fuse/connections type fusectl (rw,relatime)
tmpfs on /tmp type tmpfs (rw,nosuid,nodev,relatime)
tmpfs on /run/user/999 type tmpfs (rw,nosuid,nodev,relatime,size=812704k,mode=700,uid=999,gid=999
)
gvfsd-fuse on /run/user/999/gvfs type fuse.gvfsd-fuse (rw,nosuid,nodev,relatime,user_id=999,group
_id=999)
/dev/sdh2 on /media/ubuntu/f73891ee-8342-4bc0-a748-d2b46f05cbd0 type ext4 (rw,nosuid,nodev,relati
me,data=ordered,uhelper=udisks2)
ubuntu@ubuntu:~$
```

Install Linux on USB 2

- Click on Install Ubuntu 20.04 LTS icon to start install



Install Ubuntu 16.04.2 LTS

Examples

Install Ubuntu 16.04.2 LTS

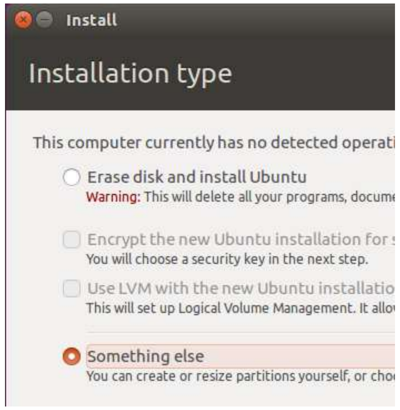
```
ubuntu@ubuntu: ~
ubuntu@ubuntu:~$ ls
Desktop  Documents  Download
ubuntu@ubuntu:~$ 
```

Install

Welcome

English
Español

- Select Something else on Installation Type screen



Install

Installation type

This computer currently has no detected operati

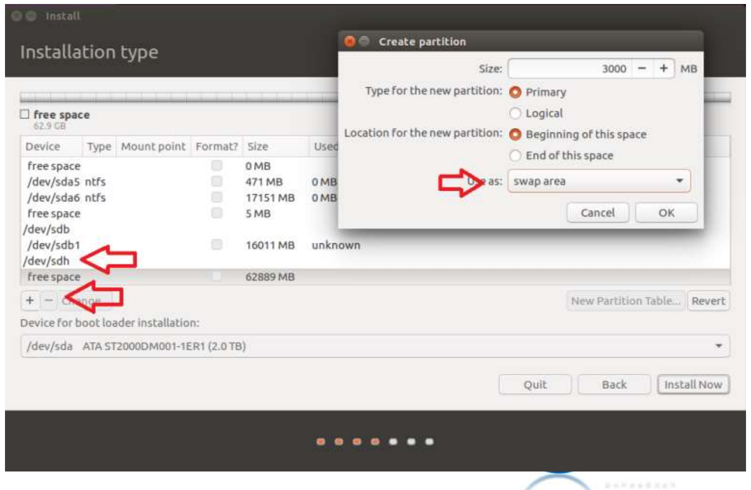
( ) Erase disk and install Ubuntu
Warning: This will delete all your programs, docume

☐ Encrypt the new Ubuntu installation for
You will choose a security key in the next step.

☐ Use LVM with the new Ubuntu installatio
This will set up Logical Volume Management. It allo

[✓ Something else]
You can create or resize partitions yourself, or cho

- Add a swap partition (3GB in the figure) to /dev/sdh
    - /dev/sdh is your USB 2
    - Please be extra careful to not to install on other devices



Install

Installation type

☐ free space
62.9 GB

Device	Type	Mount point	Format?	Size	Used
free space			☐	0 MB	
/dev/sda5	ntfs		☐	471 MB	0 MB
/dev/sda6	ntfs		☐	17151 MB	0 MB
free space			☐	5 MB	
/dev/sdb					
/dev/sdb1			☐	16011 MB	unknown
/dev/sdh					
free space			☐	62889 MB	

Device for boot loader installation:
/dev/sda ATA ST2000DM001-1ER1 (2.0 TB)

Create partition

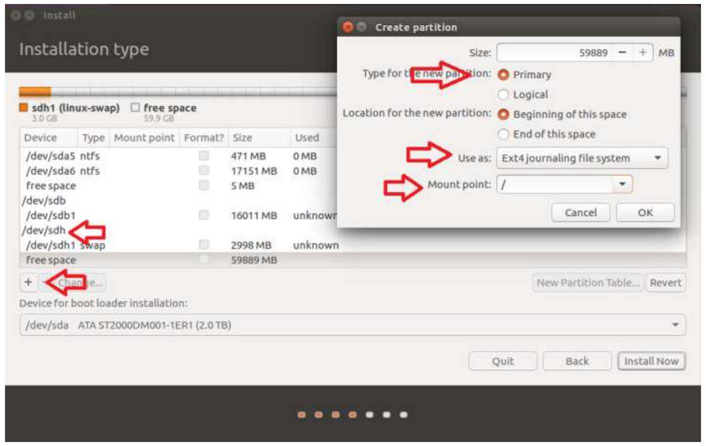
Size: 3000 MB
Type for the new partition: [✓ Primary] Logical
Location for the new partition: [✓ Beginning of this space] End of this space
Use as: swap area

Install Linux on USB 2

- Add an Ext4 partition for your main storage

- Select the Ext4 partition for the boot loader installation
  - Be careful not to select your HDD; it will ruin your Laptop

- Click on the Install Now button



Installation type

🟧 sdh1 (linux-swap) 3.0 GB ☐ free space 59.9 GB

Device	Type	Mount point	Format?	Size	Used
/dev/sda5	ntfs		☐	471 MB	0 MB
/dev/sda6	ntfs		☐	17151 MB	0 MB
free space			☐	5 MB	
/dev/sdb					
/dev/sdb1			☐	16011 MB	unknown
/dev/sdh					
/dev/sdh1	swap		☐	2998 MB	unknown
free space				59889 MB	

Device for boot loader installation: /dev/sda ATA ST2000DM001-1ER1 (2.0 TB)

Create partition

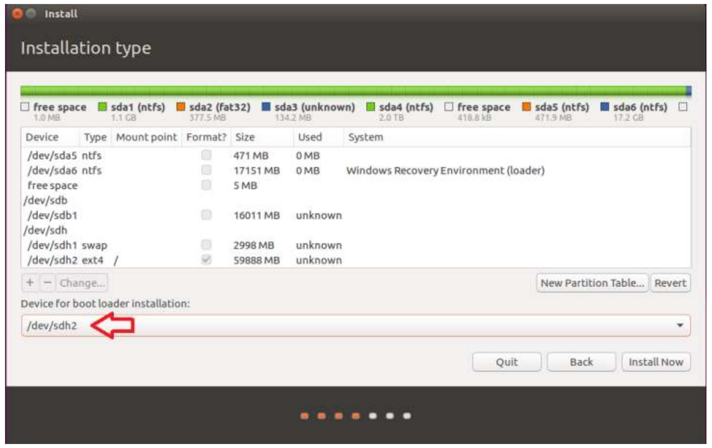
Size: 59889 MB

Type for the new partition: [✓ Primary] Logical

Location for the new partition: [✓ Beginning of this space] End of this space

Use as: Ext4 journaling file system

Mount point: /



Installation type

☐ free space 1.0 MB 🟩 sda1 (ntfs) 1.1 GB 🟧 sda2 (fat32) 377.5 MB 🟦 sda3 (unknown) 134.2 MB 🟩 sda4 (ntfs) 2.0 TB ☐ free space 418.8 kB 🟧 sda5 (ntfs) 471.9 MB 🟦 sda6 (ntfs) 17.2 GB ☐

Device	Type	Mount point	Format?	Size	Used	System
/dev/sda5	ntfs		☐	471 MB	0 MB	
/dev/sda6	ntfs		☐	17151 MB	0 MB	Windows Recovery Environment (loader)
free space			☐	5 MB		
/dev/sdb						
/dev/sdb1			☐	16011 MB	unknown	
/dev/sdh						
/dev/sdh1	swap		☐	2998 MB	unknown	
/dev/sdh2	ext4	/	✓	59888 MB	unknown	

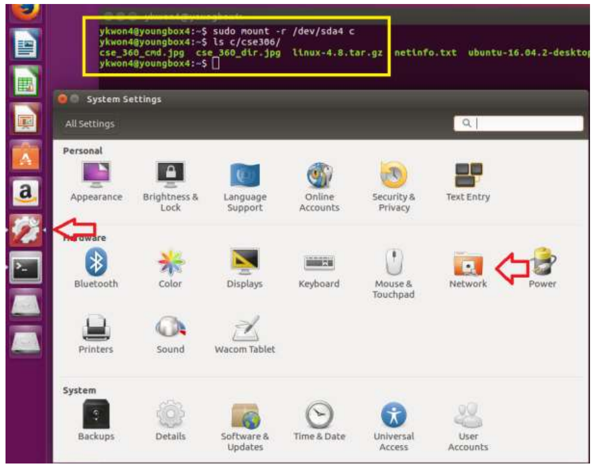
Device for boot loader installation: /dev/sdh2

Copy Files from Windows to Linux

- Copy necessary files
  - Open a terminal (Ctrl + Alt + T)
  - Mount your HDD (Windows) to c directory
    - `mkdir c`
    - `sudo mount -r /dev/sda4 ./c`
  - Copy the necessary files from the HDD
    - `cp ./c/cse306/netinfo.txt .`
    - `cp ./c/cse306/ubuntu-20.04-desktop-amd64.iso .`
    - `cp ./c/cse306/linux-5.4.49.tar.gz .`
  - Unmount the HDD
    - `sudo umount ./c`
    - `rmdir c`

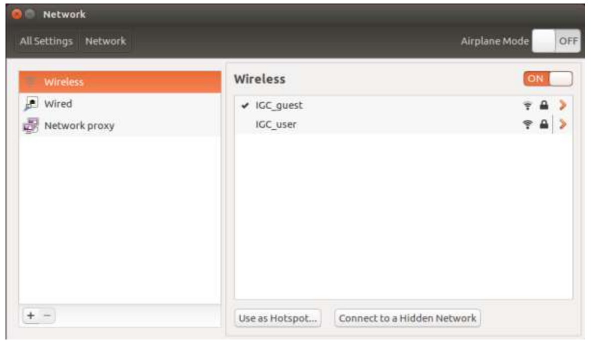
Network Setup

- Run `cat ./netinfo.txt` to get your Windows' network setting



```
ykwon4@youngbox4:~$ sudo mount -r /dev/sda4 c
ykwon4@youngbox4:~$ ls c/cse306/
cse_360_cmd.jpg  cse_360_dir.jpg  linux-4.8.tar.gz  netinfo.txt  ubuntu-16.04.2-deskto
ykwon4@youngbox4:~$
```

- You can use igc_user or igc_guest, but they might be slow



Install Linux on VM

- Use a computer connected to a fast network (igc_guest is slow)

- Install QEMU
  - `sudo apt-get update`
  - `sudo apt-get install qemu-kvm qemu virt-manager virt-viewer libvirt-bin`

- Create an HDD of 25 GB for your VM
  - `qemu-img create ubuntu_org.img 25G`

Install Linux on VM

- Install Linux on VM
  - `qemu-system-x86_64 -hda ubuntu_org.img -boot d -cdrom ./ubuntu-16.04.2-desktop-amd64.iso -m 2048 -enable-kvm`
    - `-hda ubuntu_org.img (use ubuntu_org.img as HDA)`
    - `-boot d (boot from cdrom)`
    - `-cdrom ... (use the iso file as a cdrom)`
    - `-m 2048 (use 2GB of memory)`
    - `-enable-kvm (enable kvm:` **`much much faster`**`)`
  - If you get *"Could not acces KVM kernel module: Permission denied"* error, try
    - sudo chmod 666 /dev/kvm

- Test launch Linux on VM
  - `qemu-system-x86_64 ubuntu_org.img -enable-kvm -m 2048`

Prepare Kernel Compilation

- Unzip the Kernel source code
  - `tar -xzvf ./linux-5.4.49.tar.gz`

- Install tools for building a Kernel
  - `sudo apt-get install git build-essential kernel-package fakeroot libncurses5-dev libssl-dev libelf-dev ccache bison flex`

Add Kernel source to git (optional)

- Add Kernel source files to git server
  - `git config -- global user.name "your full name"`
  - `git config -- global user.email "your email"`
  - `cd linux-5.4.49`
  - `git init`
  - `git add -A .`
  - `git commit -m "initial add"`

Make a copy of your VM image

- Make a COW image
  - `qemu-img create -f qcow2 -o backing_file=ubuntu_org.img ubuntu.img`
  - We will work on the cow image. If the image is broken, creating another cow image is easy

Build Kernel

- Build Kernel
  - Go to the Kernel source directory
    - `cd linux-5.4.49`
  - Configure the Kernel run one of these
    - `make defconfig  (recommended easiest way)`
    - `make menuconfig`
    - `make config`
  - Compile the Kernel
    - `make`

Build Kernel

- On errors
  - PIC related error: add the yellow text to linux-5.4.49/Makefile
  - _____ilog2_NaN error: add the green text to linux-5.4.49/Makefile

```
all: vmlinux

# The arch Makefile can set ARCH_{CPP,A,C}FLAGS to override the default
# values of the respective KBUILD_* variables
ARCH_CPPFLAGS :=
ARCH_AFLAGS :=
ARCH_CFLAGS :=
include arch/$(SRCARCH)/Makefile

KBUILD_CFLAGS   += $(call cc-option,-fno-pie,)
KBUILD_CFLAGS   += $(call cc-option,-no-pie,)
KBUILD_AFLAGS   += $(call cc-option,-fno-pie,)
KBUILD_CPPFLAGS += $(call cc-option,-fno-pie,)
KBUILD_CFLAGS   += $(call cc-option,--param=max-fsm-thread-path-insns=1)
```

Launch Using the Compiled Kernel

- Launch with your private Kernel image (graphic mode)

  -
    ```
    qemu-system-x86_64 -kernel linux-5.4.49/arch/x86/boot/bzImage -hda ubuntu.img -append "root=/dev/sda5 init=/sbin/init" -enable-kvm -m 4096
    ```

- Launch with your private Kernel image (text mode)

  -
    ```
    qemu-system-x86_64 -nographic -serial mon:stdio -kernel linux-5.4.49/arch/x86/boot/bzImage -hda ubuntu.img -append "root=/dev/sda5 console=ttyS0 init=/sbin/init" -enable-kvm -m 4096
    ```

Save changes to git

- When you made any changes
  - `git add <files you changed>`
  - `git commit -m "<change description>"`
  - `git format-patch -1`