# Pre-Installation

### 1.1 Acquire an installation image

- Installed ISO from HTTP from the mirror site listed

### 1.2 Verify Signature

- Verify the image signature by matching checksums using 7zip to compare the SHA256 in the ISO

### 1.3 Prepare an Installation Medium

- Create a new Virtual Machine in VMware Workstation and select the ISO file
- Choose Linux 6.x Kernel 64-bit and allocate for 2 GB of memory and 20GB of Ram space.

### 1.4 Booting

- boot the environment and select *Arch Linux install medium* and press enter to enter the environment

### 1.5 Set console keyboard layout and font

- `#localectl list-keymaps` : to show available layouts
- `#loadkeys us` : to set keyboard layout to US

### 1.6 Verify Boot Mode

- use # `cat /sys/firmware/efi/fw_platform_size` to verify boot mode
  - Since it returned No such file or directory, my system is in BIOS

### 1.7 Connecting to Internet

- Check network interface with `ip link` to make sure state up
- `ping -c 3 archlinux.org` to see if internet is working

### 1.8 Update systemclock

- use `timedatectl` to make sure the system clock is synchronized
- Since my time was of by 5 hours, i had to set my time zone from universal to Central by using:
  `timedatectl set-timezone America/Chicago`

### 1.9 Partitioning

- Confirm if your boot mode by using:
  ls `/sys/firmware/efi/efivars`
  since it said no such file or directory, my boot mode is in BIOS/Legacy mode
- use `fdisk /dev/sda` to open disk
- create new MBR partition table:
  1. `o` (creates clean DOS/MBR table)
- swap partition:
  1. `n` (new partition)
  2. `p` (primary)
  3. `1` (partition number)
  4. `Enter` (first sector(default))
  5. `+4G` (last sector = 4 GB)
  6. `t` (change type)
  7. `1` (select partition 1)
  8. `82` (change type of partition to Linux swap / Solaris)
- create root /partition (rest of disk):
  1. `n` (new partition)
  2. `p` (primary)
  3. `2` (partition number)
  4. `Enter` (first sector(next available default))
  5. `Enter` (last sector(default, rest of disk))
- verify the partitions is correct with `p`
- Save partition table and exits fdisk with `w`

### 1.10 Format Partitioning

- `mkfs.ext /dev/sd2` to create ext4 filesystem on my root / partition
- `mkswap /dev/sda1` to set up partition for swap
- `swapon /dev/sda1` to activates it immediately

### 1.11 Mount the File Systems

- `mount /dev/sda2 /mnt` to set /mnt as the root of the new system
- verify mounted partitions with `lsblk -f`

# Installation

- install the essentials packages using:
  - `pacstrap -K /mnt base linux linux-firmware`

# Configuring the System

### Generate Fstab

```
- `genfstab -U /mnt >> /mnt/etc/fstab`
- `cat /mnt/etc/fstab` to verify that partitions correctly listed
```

### Chroot into new system

- arch-chroot /mnt

### Time

- ln -sf /usr/share/zoneinfo/America/Chicago etc/localtime
- hwclock --systohc

### Localization

- install nano with pacman -Sy nano
  - -s install package
  - -y refresh package database
- open the locale file using nano /etc/locale/gen and uncomment UTF-8 locales of choice
- generate locales with locale-gen and then create a file called /etc/locale.conf by using nano
  - type LANG=en_US.UTF-8 inside and then save and exit

### Network Configuration

- echo toan > /etc/hostname to set my hostname
- configure host file using nano /etc/hosts and add 127.0.1.1 Toan
- install networkmanagaer package with pacman -S networkmanager and allow it to automatically start at boot with systemctl enable NetworkManager

### Initramfs

- `mkinitcpio -P`

### Root Password

- use `passwd` to set password of choosing

### Install Bootloader

- I chose to do grub so i did `pacman -S grub`
- i install grub directly to my drive `grub-install /dev/sda` since im on bios mode
- `grub-mkconfig -o /boot/grub/grub.cfg` to generate the config file that grub uses to display boot menu
- then reboot by typing reboot after exiting the chroot

# Extra Modifications

### Installing a Desktop Environment

- `pacman -S xorg` to download Xorg (X server)
  - install entire xorg group by pressing enter
  - chooose 1 for `man-db`
- then install LXQt and display manager
  - `pacman -S lxqt sddm`
    - install all 24 members by pressing Enter
    - pick out of the 11 providers available(I chose 5)
  - `systemctl enable sddm.service` to allow sddm to start at boot

### Create user accounts

- creater evil toan and set password
  - useradd -m -G wheel -s /bin/bash eviltoan
  - passwd eviltoan
- create codi and set password
  - useradd -m -G wheel -s /bin/bash codi
  - passwd codi

### Install different shell other than bash(fish)

- `pacman -S fish` to install fish
- change default shell for eviltoan to fish with
  - chsh -s /usr/bin/fish eviltoan

### Install ssh

- pacman -S openssh to install openssh
- enable ssh and start it immediately without rebooting
  - `systemctl enable sshd.service`
  - `systemctl start sshd.service`

- check status with `systemctl status sshd.service`