1. I transfer the ISO file from the flash drive passed around class. Using the ISO file created started the Arch system in VM Workstation. I gave the VM 20 Gb of storage space, 4 Gb of RAM, and 2 processors. I also ran the Arch VM in UEFI mode.
2. To setup UEFI mode, I went to the library, clicked on settings, went to options, and then under advanced, I changed the firmware type to UEFI.
3. After setting up the specs, I ran the VM from workstation and did a ping test to see if I was connected to the internet. I wasn’t so I used that following commands to connect. I checked a list of all the network interfaces available to me and then told system to start the DHCP service. After that I ran another ping test and I was all good.
   1. ping -c 4 [www.google.com](http://www.google.com)
   2. ip address
   3. ip link set up enp0s3
   4. systemctl start dhcpcd.service
   5. ping -c 4 www.google.com
4. Next, I set up the system clock.
   1. timedatectl set-ntp true
5. Then I ran a command to see the current disk layout. The layout showed that sda had 20 Gb of storage and loop0 had a little over 500 Mib.
   1. fdisk -1
   2. lsblk
6. Next, I created the partitions for the system. I created three partitions. One of the partitions was given 500 MB and was my EFI partition. The other partition was used 18.5 Gb was used for the Arch OS. The third partition was a swap partition that was given 1Gb. For creating the partitions, I used cfdisk instead of fdisk because I find cfdisk to be easier to use.
   1. cfdisk /dev/sda
7. I created the swap file using the following commands.
   1. mkswap /dev/sda3
   2. swapon /dev/sda3
8. I created the root file system using the following commands.
   1. mkfs.ext4 /dev/sda2
9. I created the EFI file system using the following commands. The EFI system partition must contain FAT32 file system.
   1. mkfs.fat -F32 /dev/sda1
10. Next, I mounted each partition. The first partition I mounted was the root partition.
    1. Mount /dev/sda2 /mnt
11. I created a boot directory where I will mount the EFI partition.
    1. mkdir /mnt/boot
12. I mounted the EFI partition to that directory.
    1. Mount /dev/sda1 /mnt/boot
13. I installed the essential packages for my Arch Linux system. This required using pacstrap.
    1. pacstrap /mnt base linux linux-firmware
14. I generated an fstab file. This will allow the system to know where to mount the partitions when it boots.
    1. Genfstab -U /mnt >> /mnt/etc/fstab
15. Since everything should be in order, we can now chroot into it.
    1. Arch-chroot /mnt
16. Now I set the region that I am currently in.
    1. ln -sf /usr/share/zoneinfo/Region/City /etc/localtime
17. Changed root password
    1. Passwd
18. Installed vim as my text editor.
    1. Pacman -S vim
19. Set up my language using vim on locale.conf file.
    1. Vim /etc/locale.conf
20. Next I started install other needed program in order to complete. I installed OpenSSH for Arch.
    1. Pacman -Syu openssh
    2. Systemctl enable sshd
21. I added in the different user accounts. First I had to install sudo into my Arch VM.
    1. Pacman –sync sudo
22. I then started creating the users for my system and setting their passwords.
    1. useradd –create-home sal
    2. passwd sal
    3. GraceHopper1906
23. I then added to the users to the wheel group
    1. usermod –append –groups wheel sal
24. I used vim to edit the /etc/sudoers file. I uncommented the line %wheel ALL=(ALL) ALL
    1. vim /etc/sudoers
25. I changed the default shell to zsh. To do this, I first installed zsh onto my ArchVM and then changed the path for the users.
    1. Pacman -S zsh
    2. Chsh -s /usr/bin/zsh
    3. Type in password
26. I allowed the alias command to change L to ‘ls -lah’ and I changed c to ‘clear’. I also needed to save these for future use so I used vim to edit the .zshrc file and then I added in the following commands. I also added chvim to go to editing the zshrc file.
    1. Alias l=’ls -lah’
    2. Alias c=’clear’
    3. Alias chvim=’vim ~/.zshrc’
    4. Vim ~/.zshrc
27. I installed a GUI for my Arch VM. I started by installing a graphics driver, then installed the display server Xorg for the desktop environment. After both of those installations, I installed the desktop environment. For my VM I chose KDE Plasma for my desktop environment. After installing the desktop environment, I also installed the display manager. I used Gnome display manager for this build.
    1. lspci | grep -e VGA
    2. pacman -Syyu
    3. pacman -S nividia nividia-utils nvidia-settings
    4. pacman -S xorg xterm xorg-xinit
    5. startx
    6. pacman -S plasma kdeplasma-addons
    7. pacman -S gdm
    8. systemctl enable gdm
    9. systemctl start gdm
28. Next I installed a sound system, terminal, firefox, and even LibreOfice because why not.
    1. Pacman -S pulseaudio pulseaudio-alsa pavucontrol
    2. Pacman -S gnome-terminal gnome-system-monitor
    3. Pacman -S firefox vlc audacious
    4. Pacman -S libreoffice