

# NASA hw2

B04902045

April 19, 2017

## System Administration

Referenced from <https://wiki.archlinux.org/> (Official)

I installed Arch Linux on Oracle Virtual Box.

I started the machine by loading archlinux-dual.iso which I downloaded from the Arch Linux official website.

I followed similar steps that are taught this week at the Lab.

### Task 1

```
#enter GNU parted
parted /dev/sda
#create partition table
mklabel gpt
#make partition
mkpart \n primary \n ext4 \n 0% \n 2000MB \n
mkpart \n primary \n ext4 \n 2000MB \n 3000MB \n
#format the partition
mkfs.ext4 /mnt/dev/sda1
mkfs.ext4 /mnt/dev/sda2
#create directory
mkdir /mnt/boot
#mount
mount /mnt/dev/sda1 /mnt
mount /mnt/dev/sda1 /mnt/boot
```

### Task 2

```
parted /dev/sda mkpart primary ext4 3000MB 5000MB
parted /dev/sda set 3 lvm on
#create a physical volume on device
pvcreate /dev/sda3
#create a volume group on this physical volume
vgcreate nasavg /dev/sda3
```

```
#create logical volumes var and home
lvcreate -n var -L 1024MB nasavg
lvcreate -n home -l 100%FREE nasavg
#format logical disk
mkfs.ext4 /dev/nasavg/var
mkfs.ext4 /dev/nasavg/home
#mount logical device
mount /dev/nasavg/var /mnt/var
mount /dev/nasavg/home /mnt/home
```

### Task 3

```
#Using tune2fs to accomplish the task
tune2fs -m 10 /dev/nasavg/var
message: Setting reserved blocks percentage to 10%
```

### Task 4

```
#set up a partition as Linux swap area
parted /dev/sda mkpart primary ext4 5000MB 6000MB
mkswap /dev/sda4
WARNING: All data on the specified partition will be lost
#enable the device for pagen
swapon /dev/sda4
#check result
swapon --show
```

### Task 5

```
Network Interfaces: enp0s3
#install base packages for arch linux
#use /mnt(temporary mount points) as root directory
pacstrap /mnt base base-devel
#If I want to access the internet after booting from GRUB
#type the following command
echo any_name > /etc/hostname
systemctl enable dhcpcd
reboot
```

### Task 6

```
#Install GRUB to disk
#Create a BIOS Boot partition on a GPT system
parted /dev/sda set 5(partition number) bios_grub on
#Change root
arch-chroot /mnt
```

```
#Get GRUB-related commands
pacman -S grub
#The following commands will:
1.Set up GRUB in the 440-byte Master Boot Record boot code region
2.Populate the /boot/grub directory
3.Generate the /boot/grub/i386-pc/core.img file
4.Embed it in the 31 KiB (minimum size - varies depending on partition alignment)
  post-MBR gap in case of MBR partitioned disk
5.In the case of a GPT partitioned disk it will embed it in the BIOS Boot Partition,
  denoted by bios_grub flag in parted and EF02 type code in gdisk

grub-install --target=i386-pc /dev/sdx
grub-mkconfig -o /boot/grub/grub.cfg
exit
```

## Task 7

```
#Use genfstab to write into static file system information
genfstab -U -p /mnt >> mnt/etc/fstab
```

## Task 8

```
lvresize --resizefs -L -1G /dev/nasavg/home
#message: Size of logical volume nasavg/home changed from 6.04 GiB to 5.04 GiB
          Logical volumne nasavg/home successfully resized.
lvresize --resizefs -L +1G /dev/nasavg/home
#message: Size of logical volumn /nasavg/home changed from 5.04GiB to 6.04 GiB
          Logical volumen nasavg/home succesfully resized.
```

## Task 9

BIOS was the first popular firmware for desktop PC.  
 Unified Extensible Firmware Interface is the successor to BIOS.  
 (BIOS and UEFI are both formats specifying  
 physical partitioning information on the hard disk.)  
 UEFI uses the GPT whereas BIOS uses the MBR partitioning scheme.  
 UEFI does not launch any boot code in the MBR whether it exists or not.  
 Instead it uses a special partition in the partion table, EFI System Partition  
 in which files required to be launched by the firmware are stored.  
 Each vendor can use the firmware or its shell (UEFI shell)  
 to launch the boot program.  
 P.S. An EFI System Partition is usually formatted as FAT32 or FAT16.  
 #Reboot and Enter iso OS  
 1.Use cfdisk to create a partition with type EFI system(sda2)  
 2.Mount properly  
 mount /dev/sda1 /mnt

```
mount /dev/sda2 /mnt/boot
#Install grub-uefi package
pacman -S efibootmgr grub
grub-install --target=x86_64-efi --efi-directory=esp_mount --bootloader-id=grub
arch-chroot /mnt /bin/bash
grub-mkconfig -o /boot/grub/grub.cfg
exit
#Shut down the virtual machine and click Enable EFI in the settings
Power on the virtual machine, we can see the grub menu with Arch Linux x86 UEFI
Done!!
```