System Administration HW6

B04705003 資工三 林子雋

1 Reference

1.1 Short Answer Questions

1. (a) An introduction to the Linux boot and startup processes
   (b) 6 Stages of Linux Boot Process (Startup Sequence)
   (c) 第十九章、開機流程、模組管理與 Loader
   (d) Linux startup process
   (e) What’s the Difference Between GPT and MBR When Partitioning a Drive?
   (f) Analyzing the Linux boot process
   (g) BIOS
   (h) Linux Boot Process

2. What Is UEFI, and How Is It Different from BIOS?

3. (a) Network File System
   (b) Network file systems and Linux
   (c) `https://www.csie.ntu.edu.tw/~hsinmu/courses/_media/nasa_18spring/slides/nfs.pdf`
   (d) NFS 各个版本之间的比较

4. (a) Preboot Execution Environment
   (b) 第二章、安裝伺服器與 kickstart 大量部署用戶端電腦

1.2 Network Ninja

1. `https://wiki.archlinux.org/index.php/Installation_guide`
2. How to Install Arch Linux [Step by Step Guide]
3. Partitioning
4. Beginners Arch Linux Installation Guide
5. How To Install Arch Linux Latest Version
6. How to Clone a Virtual Machine in VMWare Player

1.3 PXE NFS

1. (a) Arch Linux NFS

2. (a) PXE

   (b) https://wiki.archlinux.org/index.php/Dnsmasq

   (c) https://wiki.archlinux.org/index.php/TFTP

3. (a) https://wiki.archlinux.org/index.php/NFS

2 Problem

2.1 Short Answer Questions

1. (a) **Hardware Initialization and Testing phase**: At this phase, the BIOS(a nonvolatile firmware) or UEFI(most modern PC uses) would perform hardware initialization and test the system hardware components(also known as Power-On Self Test). After its tasks done, it would load and execute the boot loader at master boot record(MBR) if BIOS is used and GPT partition if UEFI is used.

   (b) **Boot loader phase**: There are three boot loaders used by most Linux distribution, LILO, GRUB and GRUB2. In this phase, LILO, GRUB or GRUB2 would get the Linux kernel loaded into memory and running.

   (c) **Kernel phase**: The kernel, if compressed to save space, would decompress itself. Then, the kernel would use /boot/initrd or /boot/initramfs as temporary root file system until the kernel is booted and the real root file system is mounted. /boot/initrd or /boot/initramfs can let the kernel load necessary drivers and modules like USB, RAID, LVM, SCSI.
   After the kernel fully loaded, the kernel would call the first process - systemd. First, systemd mounts the filesystem as defined by /etc/fstab. It uses its configuration file, /etc/systemd/system/default.target to decide which state or target it should boot the host into. The default.target is always aliased with a symbolic link to either multi-user.agent or graphical.target. However, before default.target runs, it requires some dependencies such like sysinit.target and basic.target.
   The sysinit.target starts up all of the low-level services and units required to enable moving on the basic.target. After the sysinit.target is fulfilled, systemd next starts the basic.target. Finally, the user-level targets, multi-user.target or graphical.target is loaded and the boot process is ended.

2. See Table 1.

3. See Table 2.

	BIOS	UEFI
Drive size limitation	Can only boot from drives of 2.1 TB or less	Can boot from drives of 9.4 zettabytes
Boot loader place	MBR	GPT
Processor mode	16 bit	32-bit or 64 bit mode
Boot process time	slow	fast
Secure boot	not support	support
Network feature	not support	support

Table 1: Differences between BIOS and UEFI

	State	Security	Transport Protocol	Write Method	File Support
NFSv2	Stateless	X	UDP	Sychronous	32-bit
NFSv3	Stateless	X	TCP, UDP	Asynchronous and synchronous	Up to 64 bit (larger than 2GB)
NFSv4	Stateful	Support RPCSEC-GSS authentication	TCP only	Asynchronous and synchronous	Up to 64 bit

Table 2: Differences between NFSv1 4

4. PXE work as the Figure 1. A PXE client side would want to retreive a boot loader and a kernel file, so it need two services: DHCP and TFTP sevices. A DHCP server will provide that client with IP and other related network parameters. And a TFTP server will give the client boot loader and a kernel file's download links. After a boot loader and a kernel file are downloaded, the client can start to boot. In summary, only by both of DHCP and TFTP service can a PXE client complete its booting process.

2.2 Network Ninja

1. First, select dos label type and partition the disk.

   ```
   cfdisk
   ```

   As Figure 2

2. Set up file systems and swap memory.

   ```
   mkfs.ext4 /dev/sda1
   mkfs.ext4 /dev/sda5
   mkswap /dev/sda2
   ```

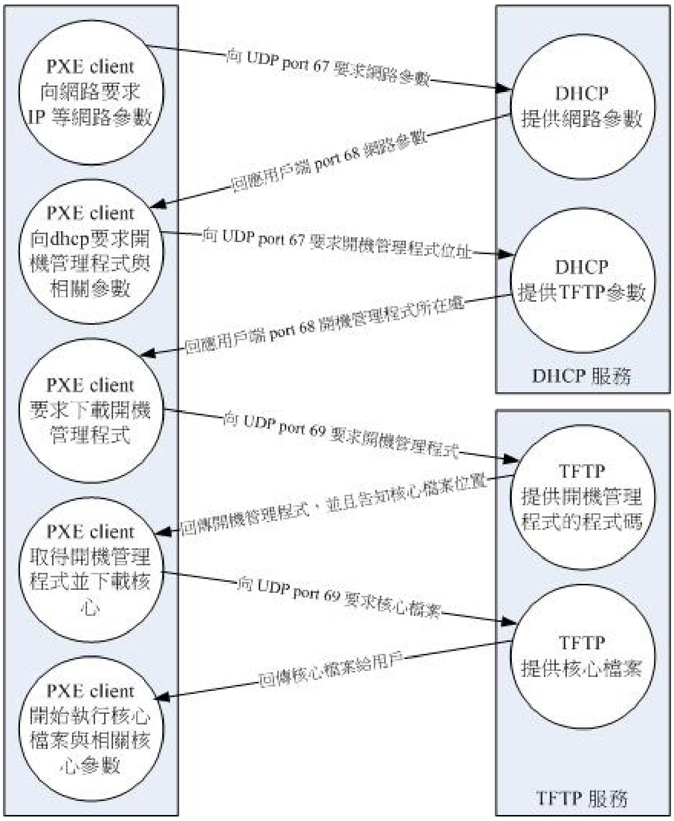
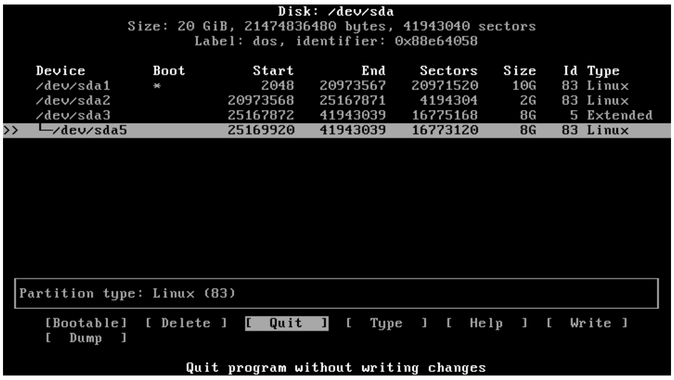

PXE client 向網路要求 IP 等網路參數
向 UDP port 67 要求網路參數
DHCP 提供網路參數
回應用戶端 port 68 網路參數
PXE client 向dhcp要求開機管理程式與相關參數
向 UDP port 67 要求開機管理程式位址
DHCP 提供TFTP參數
回應用戶端 port 68 開機管理程式所在處
DHCP 服務
PXE client 要求下載開機管理程式
向 UDP port 69 要求開機管理程式
TFTP 提供開機管理程式的程式碼
回傳開機管理程式，並且告知核心檔案位置
PXE client 取得開機管理程式並下載核心
向 UDP port 69 要求核心檔案
TFTP 提供核心檔案
回傳核心檔案給用戶
PXE client 開始執行核心檔案與相關核心參數
TFTP 服務

Figure 1: PXE Workflow



```
                           Disk: /dev/sda
          Size: 20 GiB, 21474836480 bytes, 41943040 sectors
                 Label: dos, identifier: 0x88e64058

    Device          Boot       Start        End    Sectors   Size   Id Type
    /dev/sda1       *           2048   20973567   20971520    10G   83 Linux
    /dev/sda2               20973568   25167871    4194304     2G   83 Linux
    /dev/sda3               25167872   41943039   16775168     8G    5 Extended
>>  └─/dev/sda5             25169920   41943039   16773120     8G   83 Linux

 Partition type: Linux (83)

     [Bootable]  [ Delete ]  [  Quit  ]  [  Type  ]  [  Help  ]  [  Write ]
     [  Dump  ]

                     Quit program without writing changes
```

Figure 2: cfdisk

```
swapon /dev/sda2
```

3. Mount primary partition.

   ```
   mount /dev/sda1 /mnt
   mkdir /mnt/home
   mount /dev/sda5 /mnt/home
   ```

4. Install arch linux base system

   ```
   pacstrap /mnt base base-devel
   ```

5. Create fstab

   ```
   genfstab /mnt >> /mnt/etc/fstab
   ```

6. Switch to arch linux base system

   ```
   arch-chroot /mnt
   ```

7. Set up locale.

   ```
   vi /etc/locale.gen # Uncomment en_US.UTF-8 UTF-8
   locale-gen
   echo "LANG=en_US.UTF-8" > /etc/locale.conf
   ```

8. Set up time zone.

   ```
   ln -sf /usr/share/zoneinfo/Asia/Taipei /etc/localtime
   ```

9. Set the time standard to UTC.

   ```
   hwclock --systohc --utc
   ```

10. Set up host name.

    ```
    echo "JimLin" > /etc/hostname
    ```

11. Install grub.

    ```
    pacman -S grub os-prober
    grub-install /dev/sda
    grub-mkconfig -o /boot/grub/grub.cfg
    ```

12. Reboot.

    ```
    umount /mnt/home
    umount /mnt
    reboot
    ```

13. After rebooting, use dhcpcd to get ip address

```
 vi /etc/systemd/network/ens33.network
 ########################
 [Match]
 Name=ens33

 [Network]
 DHCP=ipv4
 ########################
 dhcpcd ens33
```

14. Update system package, synchronize repository and update mirrorlist.

```
 pacman -Syu
```

15. Install packages

```
pacman -S openssh
vim /etc/ssh/sshd_config
#########################
PermitRootLogin yes
#########################
```

16. Add iptables rule to log all OUTPUT packets:

```
 iptables -A OUTPUT -o ens33 -j LOG
 iptables-save
```

17. Then, you will find all log messages by so-called ”journal” in arch linux.

```
 journalctl -k | grep 'IN=.*OUT=.*'
```

18. See Figure 3 for the result.

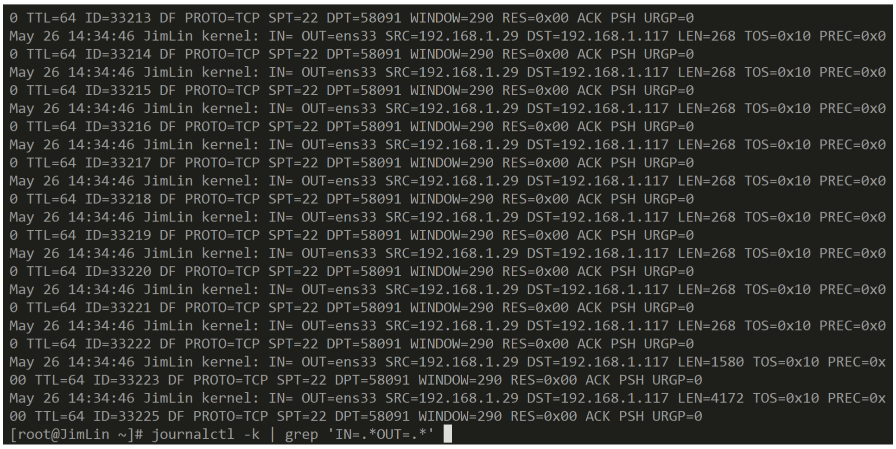
2.3 PXE & NFS

Enviroment: VMware Workstation 14 Player on Windows 10

1. 3.1:

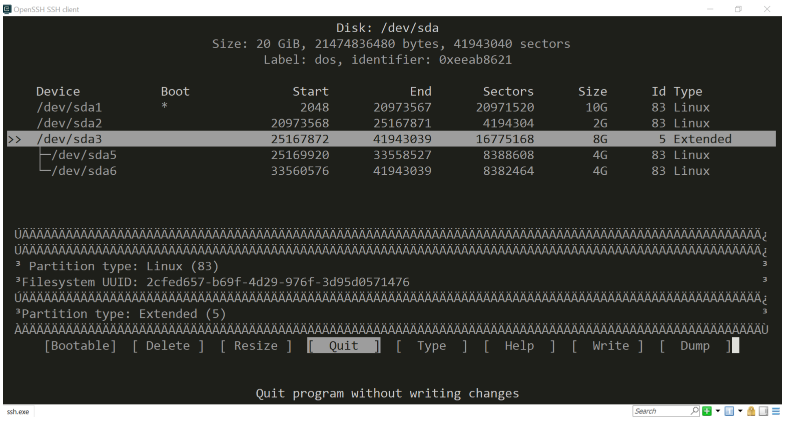
   (a) First, use the arch linux install as ”Network Ninja”, but add a new partition for /nfs as Figure 4

   (b) Make file system for /dev/sda6, mount on /nfs directory and add record in /etc/fstab for permanent mount.



```
0 TTL=64 ID=33213 DF PROTO=TCP SPT=22 DPT=58091 WINDOW=290 RES=0x00 ACK PSH URGP=0
May 26 14:34:46 JimLin kernel: IN= OUT=ens33 SRC=192.168.1.29 DST=192.168.1.117 LEN=268 TOS=0x10 PREC=0x0
0 TTL=64 ID=33214 DF PROTO=TCP SPT=22 DPT=58091 WINDOW=290 RES=0x00 ACK PSH URGP=0
May 26 14:34:46 JimLin kernel: IN= OUT=ens33 SRC=192.168.1.29 DST=192.168.1.117 LEN=268 TOS=0x10 PREC=0x0
0 TTL=64 ID=33215 DF PROTO=TCP SPT=22 DPT=58091 WINDOW=290 RES=0x00 ACK PSH URGP=0
May 26 14:34:46 JimLin kernel: IN= OUT=ens33 SRC=192.168.1.29 DST=192.168.1.117 LEN=268 TOS=0x10 PREC=0x0
0 TTL=64 ID=33216 DF PROTO=TCP SPT=22 DPT=58091 WINDOW=290 RES=0x00 ACK PSH URGP=0
May 26 14:34:46 JimLin kernel: IN= OUT=ens33 SRC=192.168.1.29 DST=192.168.1.117 LEN=268 TOS=0x10 PREC=0x0
0 TTL=64 ID=33217 DF PROTO=TCP SPT=22 DPT=58091 WINDOW=290 RES=0x00 ACK PSH URGP=0
May 26 14:34:46 JimLin kernel: IN= OUT=ens33 SRC=192.168.1.29 DST=192.168.1.117 LEN=268 TOS=0x10 PREC=0x0
0 TTL=64 ID=33218 DF PROTO=TCP SPT=22 DPT=58091 WINDOW=290 RES=0x00 ACK PSH URGP=0
May 26 14:34:46 JimLin kernel: IN= OUT=ens33 SRC=192.168.1.29 DST=192.168.1.117 LEN=268 TOS=0x10 PREC=0x0
0 TTL=64 ID=33219 DF PROTO=TCP SPT=22 DPT=58091 WINDOW=290 RES=0x00 ACK PSH URGP=0
May 26 14:34:46 JimLin kernel: IN= OUT=ens33 SRC=192.168.1.29 DST=192.168.1.117 LEN=268 TOS=0x10 PREC=0x0
0 TTL=64 ID=33220 DF PROTO=TCP SPT=22 DPT=58091 WINDOW=290 RES=0x00 ACK PSH URGP=0
May 26 14:34:46 JimLin kernel: IN= OUT=ens33 SRC=192.168.1.29 DST=192.168.1.117 LEN=268 TOS=0x10 PREC=0x0
0 TTL=64 ID=33221 DF PROTO=TCP SPT=22 DPT=58091 WINDOW=290 RES=0x00 ACK PSH URGP=0
May 26 14:34:46 JimLin kernel: IN= OUT=ens33 SRC=192.168.1.29 DST=192.168.1.117 LEN=268 TOS=0x10 PREC=0x0
0 TTL=64 ID=33222 DF PROTO=TCP SPT=22 DPT=58091 WINDOW=290 RES=0x00 ACK PSH URGP=0
May 26 14:34:46 JimLin kernel: IN= OUT=ens33 SRC=192.168.1.29 DST=192.168.1.117 LEN=1580 TOS=0x10 PREC=0x
00 TTL=64 ID=33223 DF PROTO=TCP SPT=22 DPT=58091 WINDOW=290 RES=0x00 ACK PSH URGP=0
May 26 14:34:46 JimLin kernel: IN= OUT=ens33 SRC=192.168.1.29 DST=192.168.1.117 LEN=4172 TOS=0x10 PREC=0x
00 TTL=64 ID=33225 DF PROTO=TCP SPT=22 DPT=58091 WINDOW=290 RES=0x00 ACK PSH URGP=0
[root@JimLin ~]# journalctl -k | grep 'IN=.*OUT=.*'
```

Figure 3: Log Result



```
                                 Disk: /dev/sda
               Size: 20 GiB, 21474836480 bytes, 41943040 sectors
                     Label: dos, identifier: 0xeeab8621

    Device          Boot          Start       End     Sectors     Size     Id Type
    /dev/sda1       *              2048  20973567    20971520      10G     83 Linux
    /dev/sda2                  20973568  25167871     4194304       2G     83 Linux
>>  /dev/sda3                  25167872  41943039    16775168       8G      5 Extended
    ├─/dev/sda5                25169920  33558527     8388608       4G     83 Linux
    └─/dev/sda6                33560576  41943039     8382464       4G     83 Linux



 ÚÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄ¿
 ÚÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄ¿
 ³ Partition type: Linux (83)                                                                          ³
 ³Filesystem UUID: 2cfed657-b69f-4d29-976f-3d95d0571476                                                ³
 ÚÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄ¿
 ³Partition type: Extended (5)                                                                         ³
 ÀÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÙ
     [Bootable]  [ Delete ]  [ Resize ]  [  Quit  ]  [  Type  ]  [  Help  ]  [  Write ]  [  Dump  ]


                          Quit program without writing changes
```

Figure 4: cfdisk partition

```
mkfs.ext4 /dev/sda5
mkfs.ext4 /dev/sda6
mkdir /nfs && chmod 777 /nfs
mount /dev/sda5 /home
mount /dev/sda6 /nfs
vim /etc/fstab
#########################################
# UUID=41720df7-88a1-4adb-8051-5fbf8646fef9
/dev/sda6 /nfs ext4 rw,relatime,data=ordered 0 2
#########################################
```

(c) Install `nfs-utils` as the official guide says.

```
pacman -S nfs-utils
```

(d) Modify /etc/exports file.

```
vim /etc/exports
##############################
/nfs 192.168.255.0/24(rw,async,subtree_check)
##############################
```

(e) Refresh /etc/exports.

```
exportfs -rav
```

(f) Open nfs server.

```
systemctl start nfs-server.service
```

2. 3.2:

(a) Install `dnsmasq` package,

```
pacman -S dnsmasq
```

(b) Preparation:

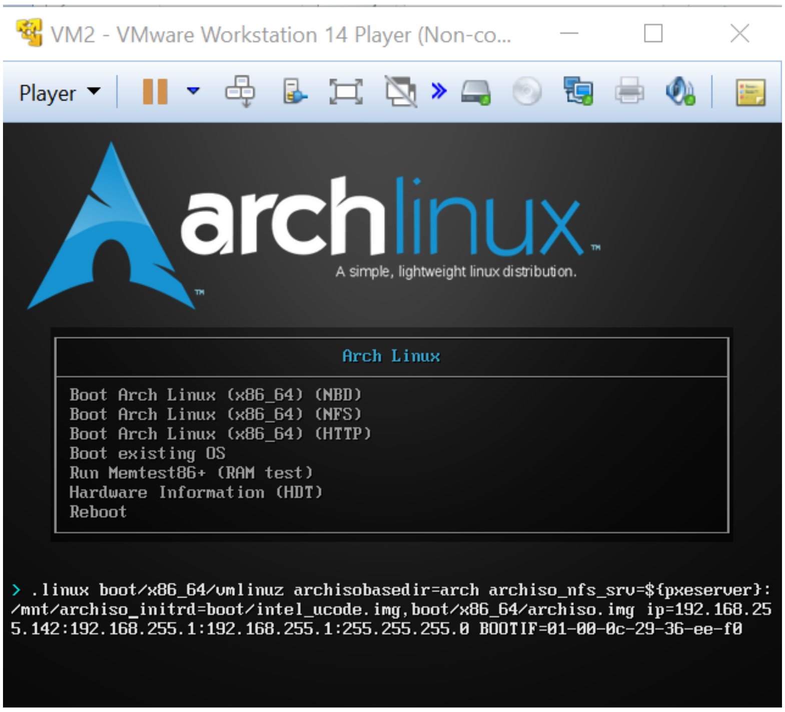
```
mkdir /mnt/archiso
mount -o loop,ro archlinux-2018.05.01-x86_64.iso
/mnt/archiso
```

(c) Network setup:

```
ip address add 192.168.255.1/24 broadcast + dev ens37
ip link set ens37 up
```

(d) Set up DCHP + TFTP and start service

```
vim /etc/dnsmasq.conf
######################################
port=0
interface=ens37
bind-interfaces
dhcp-range=192.168.255.2,192.168.255.254,12h
dhcp-boot=/arch/boot/syslinux/lpxelinux.0
dhcp-option-force=209,boot/syslinux/archiso.cfg
dhcp-option-force=210,/arch/
dhcp-option-force=66,192.168.255.1
enable-tftp
tftp-root=/mnt/archiso
######################################
systemctl start dnsmasq.service
```


VM2 - VMware Workstation 14 Player (Non-co...
Player
archlinux
A simple, lightweight linux distribution.

```
                         Arch Linux
 Boot Arch Linux (x86_64) (NBD)
 Boot Arch Linux (x86_64) (NFS)
 Boot Arch Linux (x86_64) (HTTP)
 Boot existing OS
 Run Memtest86+ (RAM test)
 Hardware Information (HDT)
 Reboot

> .linux boot/x86_64/vmlinuz archisobasedir=arch archiso_nfs_srv=${pxeserver}:
/mnt/archiso_initrd=boot/intel_ucode.img,boot/x86_64/archiso.img ip=192.168.25
5.142:192.168.255.1:192.168.255.1:255.255.255.0 BOOTIF=01-00-0c-29-36-ee-f0
```

Figure 5: Choose NFS mode

(e) Set up NFS to export installation media and open nfs-server.

```
vim /etc/exports
####################################
/mnt/archiso 192.168.255.0/24(ro,no_subtree_check)
####################################
exportfs -rav
systemctl start nfs-server.service
```

(f) Remeber to press F12(to get into PXE mode) and change one variable at NFS mode in client VM when booting. Figure 5 shows it.

```
archiso_nfs_srv= ${pxeserver}:/mnt/archiso
```

3. 3.3:

(a) Installation is same as ”Network Ninja” setting. And make sure you have to turn on **bridge network** to let VM2 and VM3 to connect to the Internet.

(b) Then, install `nfs-utils` packages, start nfs-client.target and mount /mnt/nfs on VM1’s /nfs.

```
pacman -S nfs-utils
systemctl start nfs-client.target
```

```
mkdir /mnt/nfs
mount 192.168.255.1:/nfs /mnt/nfs
```