
Linux System 관리

Rocky Linux 8

1.1 Linux 개요

I . Operating System

- 컴퓨터의 Hardware와 다른 응용프로그램들을 제어, 관리하는 역할을 하는 System Software
- 운영 체제의 목적
 - 사용자가 프로그램을 편리하게 사용할 수 있는 환경 제공
 - 자원을 최대한 공정하고 효율적으로 프로세스에 할당
 - 올바르게 많은 자원의 낭비를 막기 위해 프로세스 관리, 입출력 장치 제어
- 운영 체제의 역할
 - 프로세스 관리(프로세스 생성, 삭제, 중단 및 재개)
 - 메모리 관리(메모리 할당 여부 감시, 프로세스의 메모리 할당 및 해제)
 - 2차 기억장치 관리(저장 장치 할당, 디스크 스케줄링)
- 운영 체제의 종류
 - Unix, Linux, Windows , MAC OS 등

II . Linux 소개

- Linus Torvalds(리누스 토발즈)가 Linux 커널 개발
 - 어원 : Linus' unix
 - Unix System의 MINIX를 i386 CPU의 IBM PC에 호환될 수 있도록 제작
 - 1991년 9월 Linux Kernel 0.01 개발, 같은 해 10월 0.02 발표

(1) Linux 역사

- 1971년 Unix Server 운영체제 발표
- 1970년대 'BSD 계열' 과 'System V' 계열로 발전
- 1985년 오픈 소스의 사상이 되는 'GNU 선언문' 발표
- 1991년 리누스 토발즈에 의해 Linux Kernel 공개
- 강력한 네트워크 기반의 운영체제로 현재도 계속해서 발전

(2) 특징

- 공개형 오픈 소스(Open Source) 운영체제(GNU/Linux)
- 다중 사용자(Multi-User), 다중 작업(Multi-Tasking), 가상 터미널(Virtual Terminal)
 - 단일 시스템 상에서, 여러 사용자가 가상 터미널을 이용하여 동시에 다중 작업이 가능
- TUI, GUI 환경 제공
 - Console(TUI; Text User Interface)과 X-Window(GUI; Graphic User Interface)
- 강력하면서 안정적인 네트워크 지원
- Unix와 완벽한 호환 : UNIX 표준인 POSIX 표준에 따라 개발됨.
- 다양한 플랫폼을 지원

- 가상 메모리 시스템 구현

「GNU 선언문」

- 1985년 FSF(Free Software Foundation)을 결성한 Richard Stallman에 의해 발표
- 오늘날까지 Linux 발전의 철학적 배경, Linux의 발전 방향 지침 제공
- <http://www.gnu.org/gnu/manifesto.ko.html>
 - GNU란 무엇인가? : GNU is Not Unix
 - 왜 GNU를 작성해야만 했는가?
 - 유닉스와 호환성을 가지는 이유
 - GNU를 사용하는 방법
 - 많은 프로그래머들이 동참을 원하는 이유
 - 당신이 기여할 수 있는 방법
 - 모든 컴퓨터 사용자가 이득을 얻게 되는 이유
 - 몇 가지 GNU의 목표에 대한 반대 의견

「FSF」

- Free Software의 개발과 보급을 위해 Richard Stallman이 1984년에 설립한 비영리 민간단체
- GNU Project 지원
- GPL(General Public License) 배포

「GPL」

- 일반 공중 라이선스(General Public License)
- FSF 재단의 GNU Project에 의해서 배포
- 공개 소프트웨어의 대표적인 적은 '라이선스 체계'
- copyleft 라고도 불림(→ 저작권을 의미하는 copyright의 반대 의미)
- 라이선스 사용료, 사용상의 제약 조건을 자유롭게 하여 소프트웨어 유통 활성화 목적
- GNU GPL이 적용된 Linux 관련 모든 소스코드는 누구나 자유롭게 사용, 변경, 배포 가능
- 1990년대의 GPL v2.0에서 2005년 GPL v3.0 발표

III. Linux 구성 요소

응용 프로그램
시스템 콜 인터페이스
커널(Kernel)
하드웨어

(1) Kernel(커널)

- Linux System 구성의 기초를 이루는 핵심(Core) 부분
- OS라 불리는 실질적인 요소
- Hardware와 사용자 프로그램 사이에 위치하여 CPU, Memory, Disk등의 자원을 제어
- 프로세스 스케줄링, 기억장치 관리, 파일 관리, 시스템 호출 인터페이스, I/O 기능
- Linux Kernel의 최신 정보 확인 → <http://www.kernel.org>

(2) Shell(셸)

- User가 입력한 명령어를 기계어로 번역해 주는 명령 해석기(Command Interpreter)
- DOS의 command.com의 역할과 유사

-
- Kernel에 내장되어 있지 않은 하나의 독립 프로그램
 - Kernel과 사용자 프로그램 사이에 위치하는 System 제어의 수단
 - User가 로그인시에 자동으로 형성
 - Shell에는 다양한 종류가 있으며 Linux의 기본 Shell은 Bash Shell
 - Shell의 종류
 - Bourne Shell(/bin/sh)
 - AT&T Bell 연구소의 Steve Bourne이 개발한 Original Unix Shell
 - Shell Programming 용도로 권장
 - Unix System의 시스템 관리자 계정(root)의 기본 Shell
 - 프롬프트 : #(관리자 계정), \$(일반 계정)
 - C Shell(/bin/csh)
 - 버클리 대학의 Bill Joy가 C언어를 기본으로 개발
 - Bourne Shell에 기능 강화(command history, alias, job control 기능 등..)
 - 프롬프트 : #(관리자 계정), %(일반 계정)
 - Korn Shell(/bin/ksh)
 - AT&T Bell 연구소의 David Korn이 개발
 - Bourne Shell에 C Shell의 몇몇 기능을 추가
 - 프롬프트 : #(관리자 계정), \$(일반 계정)
 - Bash Shell(GNU/Bourne Again Shell)(/bin/bash)
 - FSF(Free Software Foundation)에서 개발한 POSIX 호환 Shell
 - Linux에서 가장 많이 사용되는 Shell
 - Bourne Shell에 csh과 ksh의 몇몇 기능을 추가
 - 프롬프트 : #(관리자 계정), \$(일반 계정)
 - 기타 TC Shell, Z Shell 존재

1.2 Linux 배포판

리눅스 배포판은 리누스 토발즈가 개발한 GNU 커널에 X 윈도우 시스템과 KDE 또는 GNOME 등의 데스크탑 환경 및 수많은 응용프로그램들을 안정적으로 사용할 수 있도록 개발된 리눅스 운영체제이다.

1. 배포판 종류

- MCC Interim Linux 배포판
 - 1992년 최초로 리눅스 배포판의 배포 시작
 - 영국 맨체스터 대학의 MCC(Manchester Computing Center) Interim Linux
 - 리눅스 커널 및 프로그램 포함, X Window 포함되지 않음
- SLS Linux 배포판
 - Soft Landing Software사에서 SLS(Softlanding Linux System) 발표
- Slackware Linux 배포판(슬랙웨어)
 - 가장 먼저 대중화된 최초의 리눅스 배포판(1992. Patrick Volkerding)
 - SLS Linux 문제점을 해결한 배포판
 - 초기 사용하기 쉽고 안정된 리눅스 운영체제로 인기
- Yggdrasil Linux 배포판

-
- 최초의 라이브 CD 배포판
 - RedHat Linux 배포판(레드햇)
 - X Window 기반의 설치 및 설정 도구 제공
 - Redhat Linux 2.0 버전부터 RPM(Redhat Packaging Manager) 추가
 - Anaconda(아나콘다 : Graphic Install 환경) 제공
 - 보다 쉬운 설치, 삭제, 업데이트
 - 현재는 Redhat Enterprise Linux(상용)과 Fedora Project 진행
 - Debian Linux 배포판(데비안)
 - 1993년 최초 배포판 작업 시작
 - GNU 정신에 가장 충실한, FSF GNU Project의 후원을 받는 유일한 배포판
 - 공개적인 배포판을 만들기 위한 목적으로 시작, 자원 봉사자들의 참여로 만들어 짐
 - SuSE Linux 배포판(수세)
 - 1994년 독일에서 개발(Software und System Entwicklung)
 - 2003년 노벨사에서 인수
 - Mandrake Linux 배포판(맨드레이크, 맨드리바)
 - 1998년 프랑스에서 개발
 - Desktop 환경 강화, KDE 그래픽 환경 통합
 - 회사의 합병을 통해 제품명을 Mandriva로 변경
 - Gentoo Linux 배포판(젠투)
 - 2002년 배포 시작
 - x86 기반으로 pc, SPARC 지원, 개발자와 네트워크 전문가를 위해 개발
 - Ubuntu Linux 배포판(우분투)
 - 가장 최근에 발표되어 데스크탑 분야에서 탁월한 성과
 - Debian GNU/Linux 기반
 - rocky Linux 배포판
 - Community ENTERprise Operating System
 - Redhat Enterprise 버전의 소스 코드를 이용해 만든 공개 버전
 - Rocky Linux 배포판
 - Redhat의 클론 버전인 rocky를 더 이상 사용할 수 없게 되자 2021년 6월에 커뮤니티에서 새롭게 배포한 클론버전

1.3 Rocky Linux 설치

1. Rocky Linux 소개

(1) Rocky Linux 8의 특징

- Rocky Linux는 RedHat Enterprise Linux의 복제본 중 하나
 - RHEL 8(RedHat Enterprise Linux 8)의 복제본으로 2021년 6월 최초 버전 발표
 - RHEL 8의 특징을 거의 그대로 유지하고 있지만 기술 지원은 불가
- Rocky Linux 8의 주요 기능
 - Linux Kernel 4.18 기반의 안정적이고 강력한 운영체제
 - SELinux 지원

「SELinux」

- Security Enhanced Linux : 보다 정교한 시스템 보안
- 정의된 보안 정책을 Kernel에 강제 적용하여 Application의 공격으로부터 차단이 가능

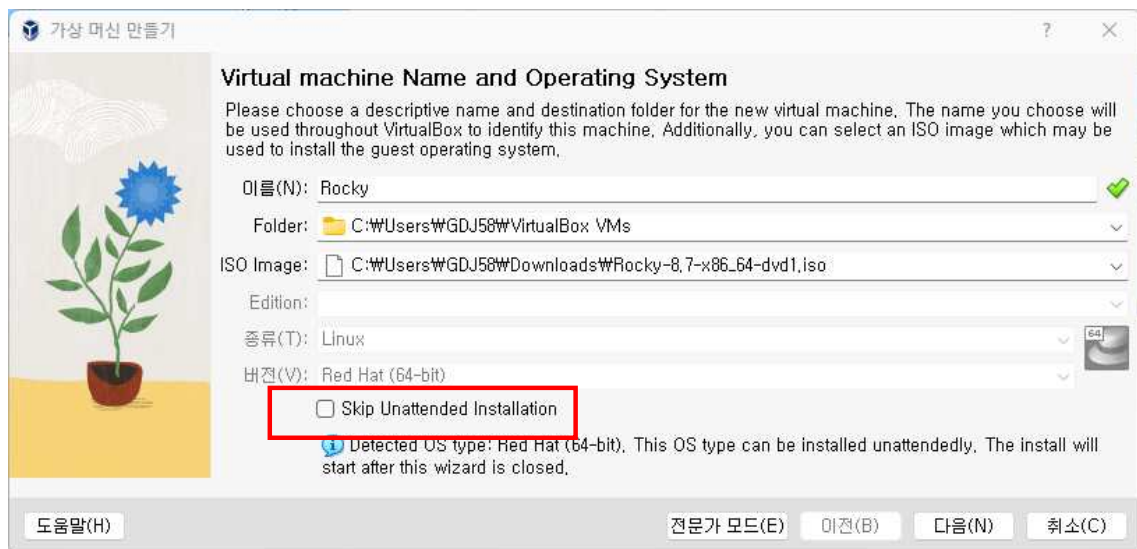
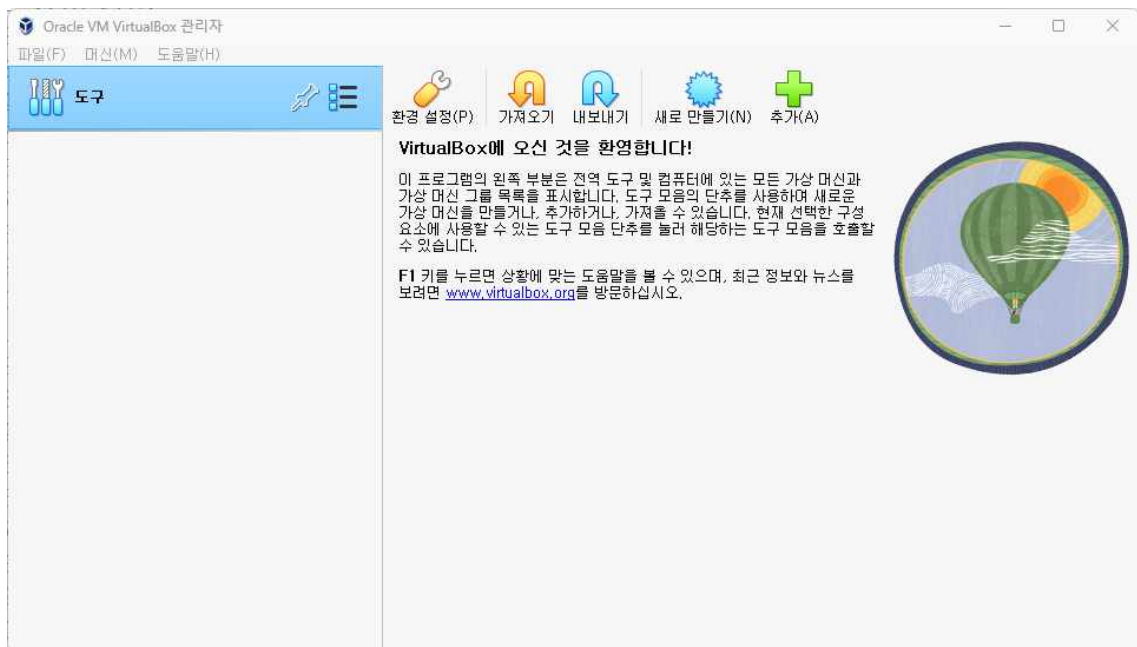
- Storage, File System 기능 향상
 - xfs 파일시스템을 기본으로 채택
 - GRUB2 지원
 - Bonding과 유사하지만 좀 더 강력해진 Teaming기술 지원
- Systemd 사용

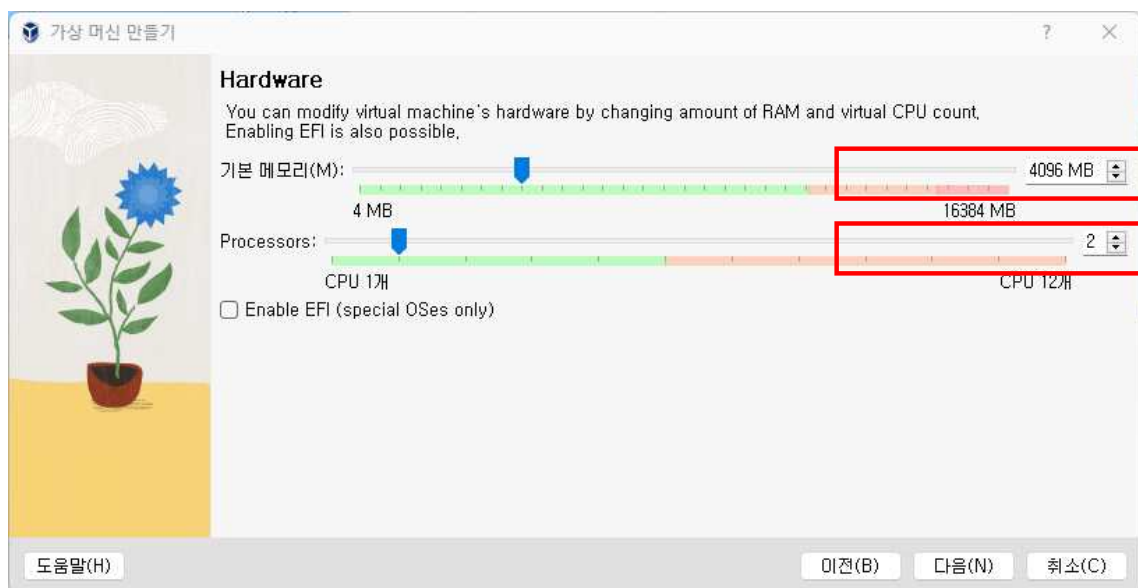
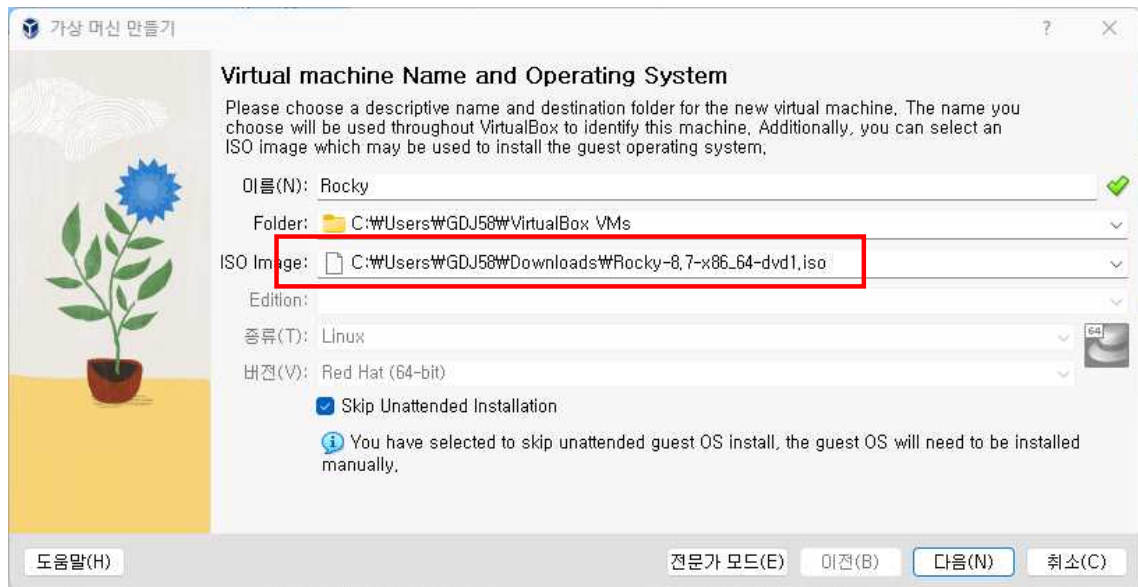
(2) Rocky Linux 8 설치 준비

- rocky를 지원하는 시스템인지 확인
 - 64비트를 지원하는 x86 계열
 - 최소 1GB 이상의 Memory
 - 표준 Hercules, CGA, EGA, VGA, SVGA Graphic Card 지원(GUI 환경 사용 시)
 - <https://hardware.redhat.com/index.cgi>에서 Hardware 호환 여부 확인
- 디스크 여유 공간
 - rocky의 전체 설치 시 최소 20GB의 공간 필요
 - 추가 패키지 설치와 각각의 사용자를 위한 공간을 감안하여 넉넉하게 준비
- rocky Disk
 - DVD 제공
 - <http://www.rocky.org>
 - DVD-ROM 이외에 ISO 이미지, 네트워크 이용 설치 가능
- System 부팅 순서 변경
 - CMOS Setup 화면에서 부팅 순서를 변경하거나 Boot Menu를 이용

II . Rocky Linux 8 설치

(1) VirtualBox 설정

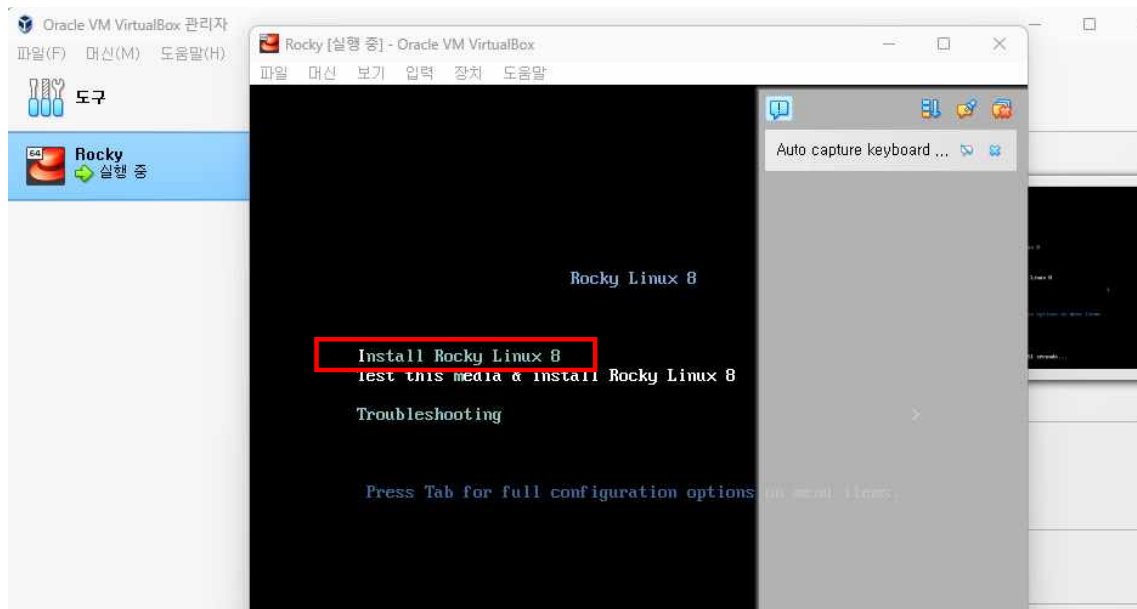


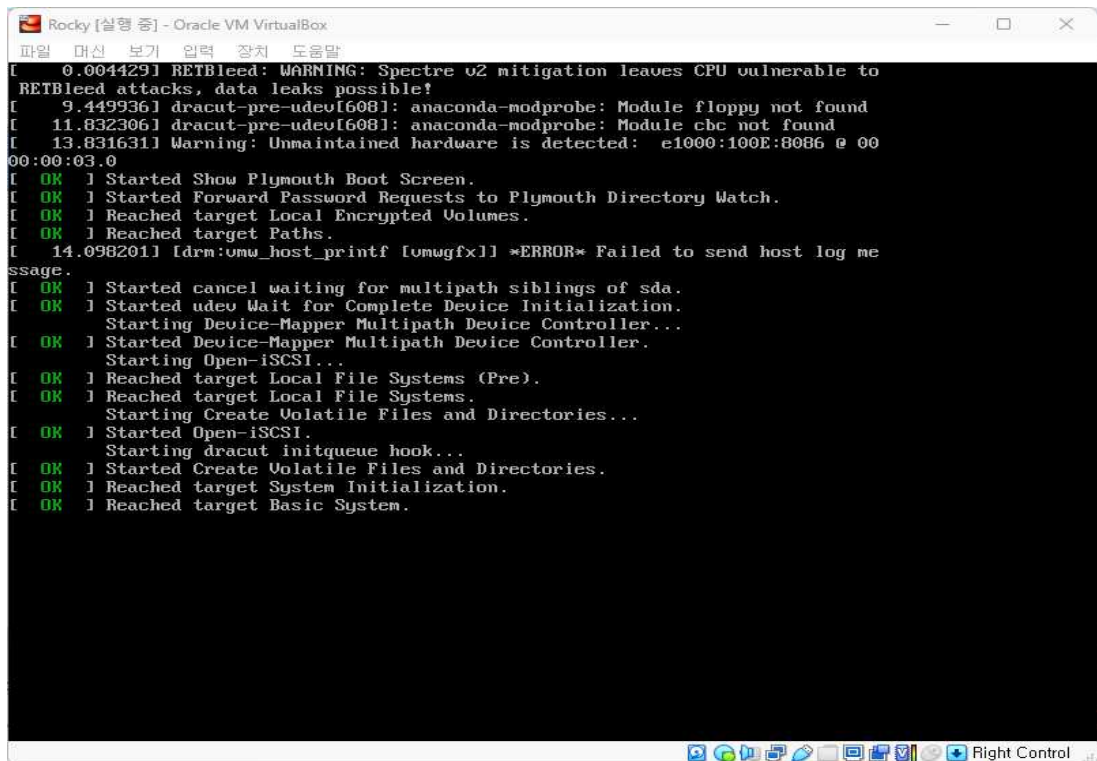






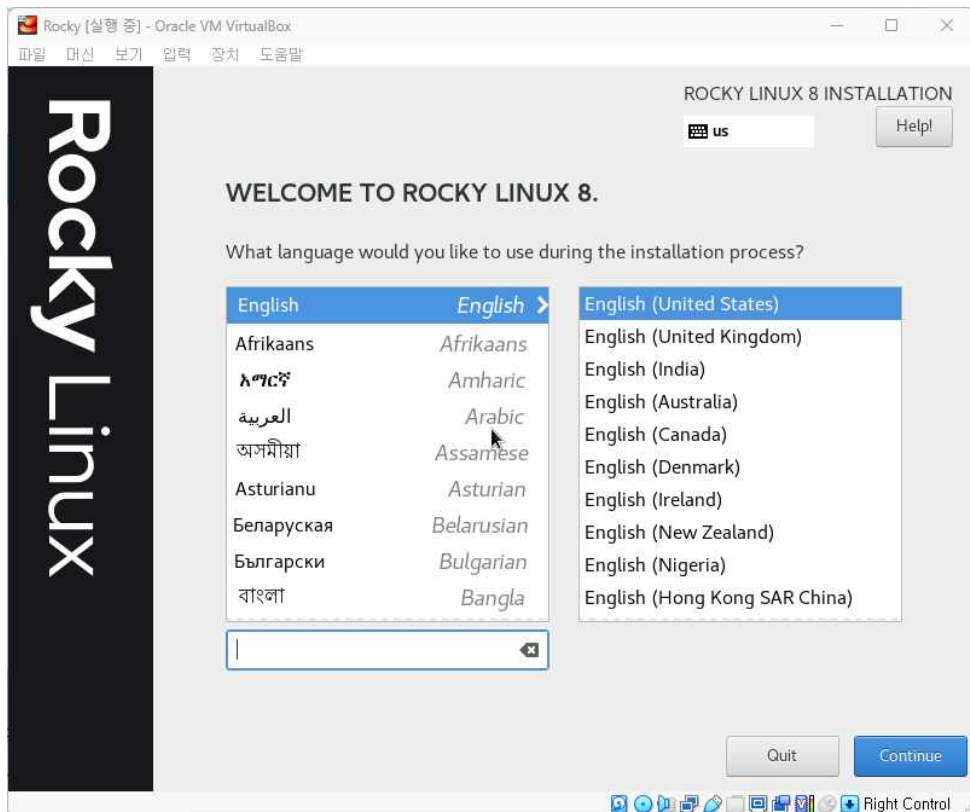
Rocky Linux DVD를 넣고 처음 부팅



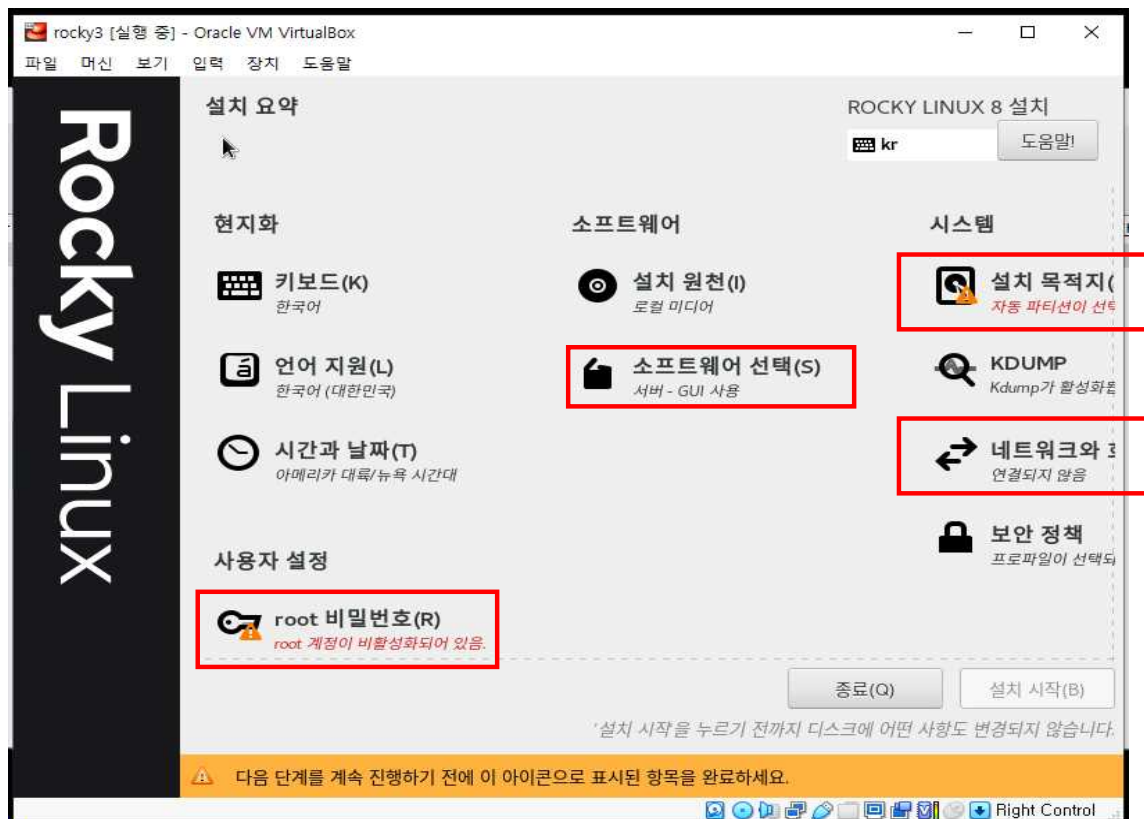
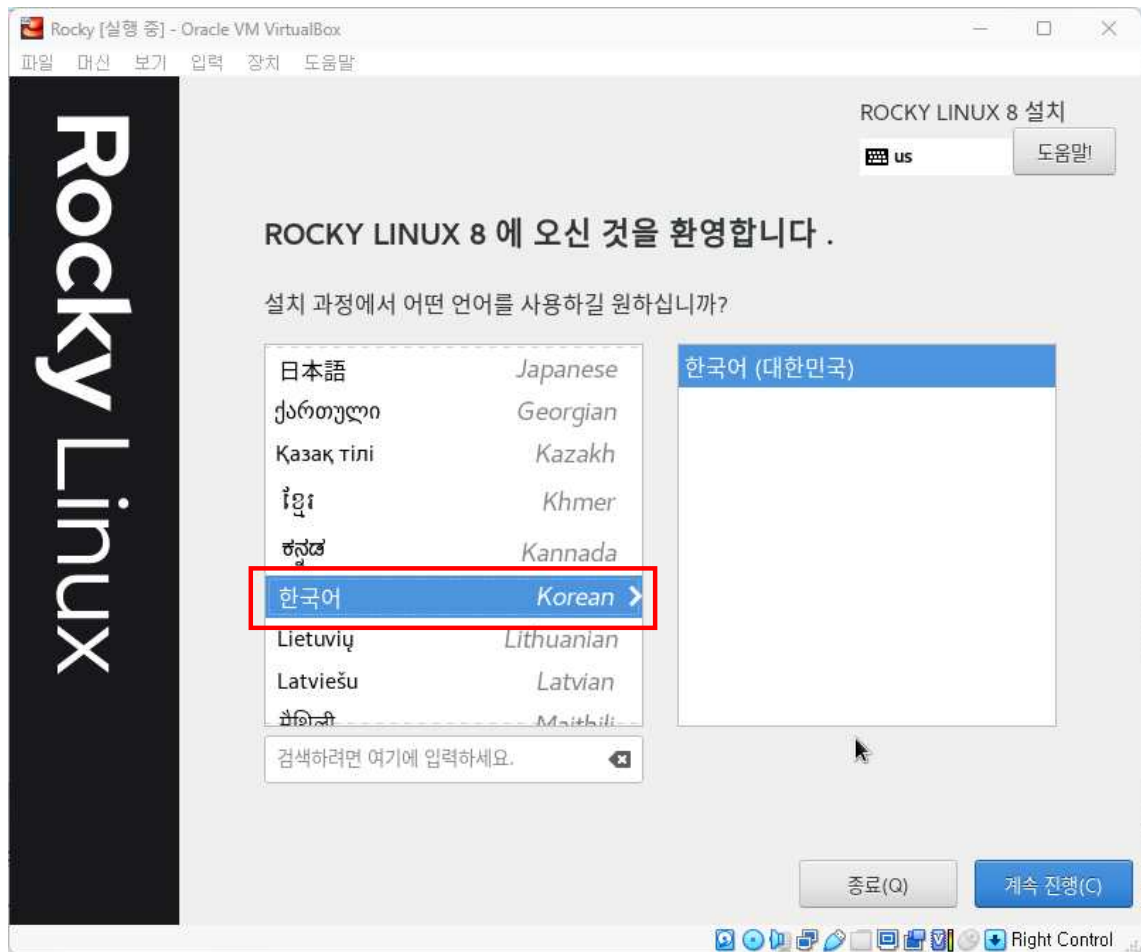


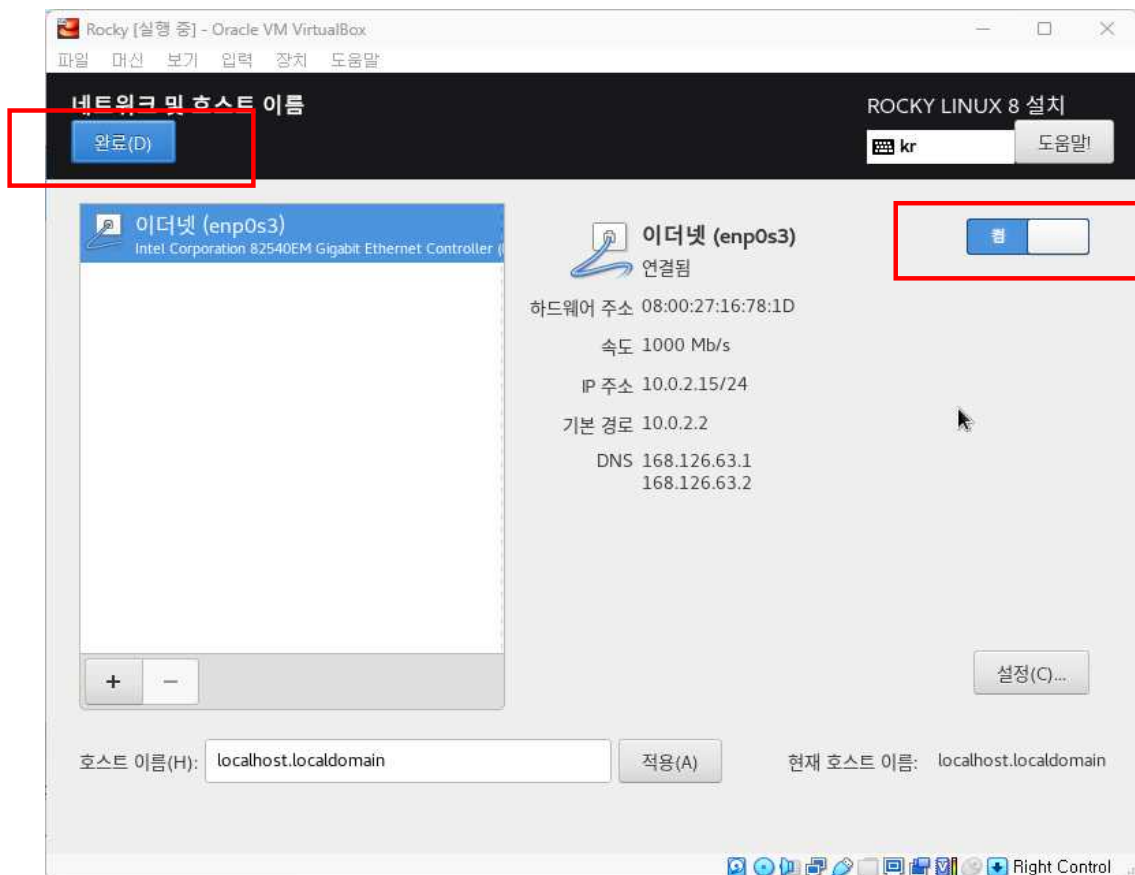
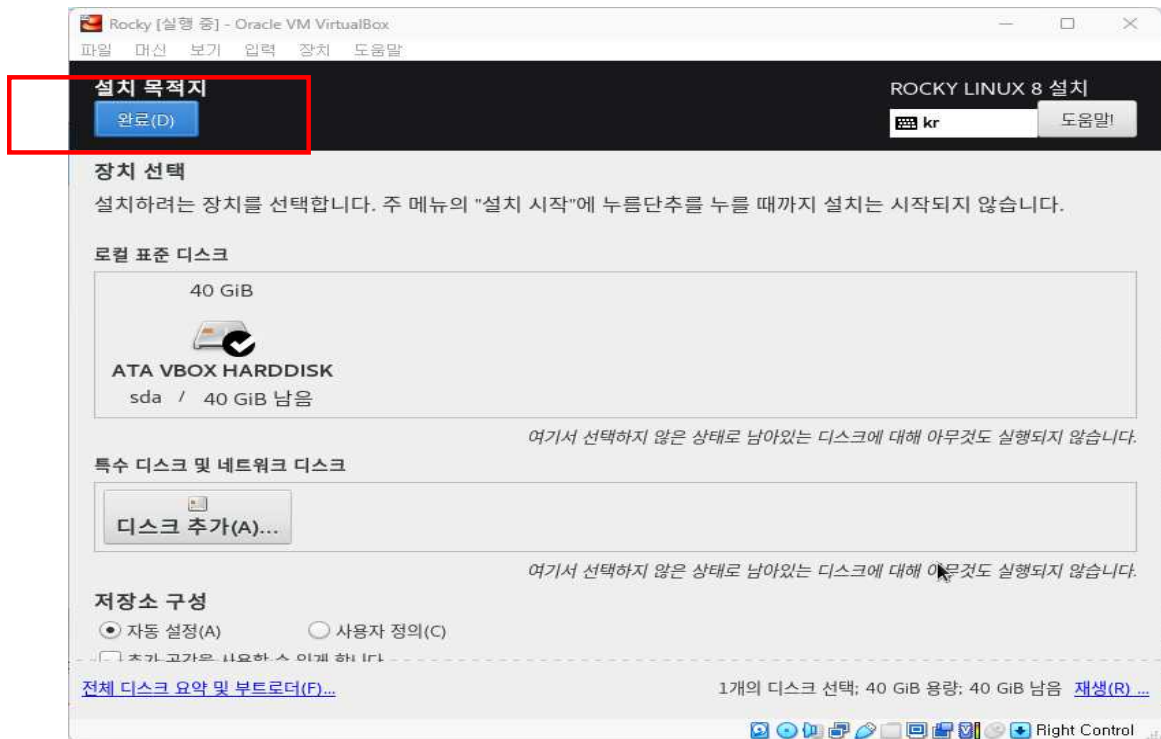
(4) 언어 선택

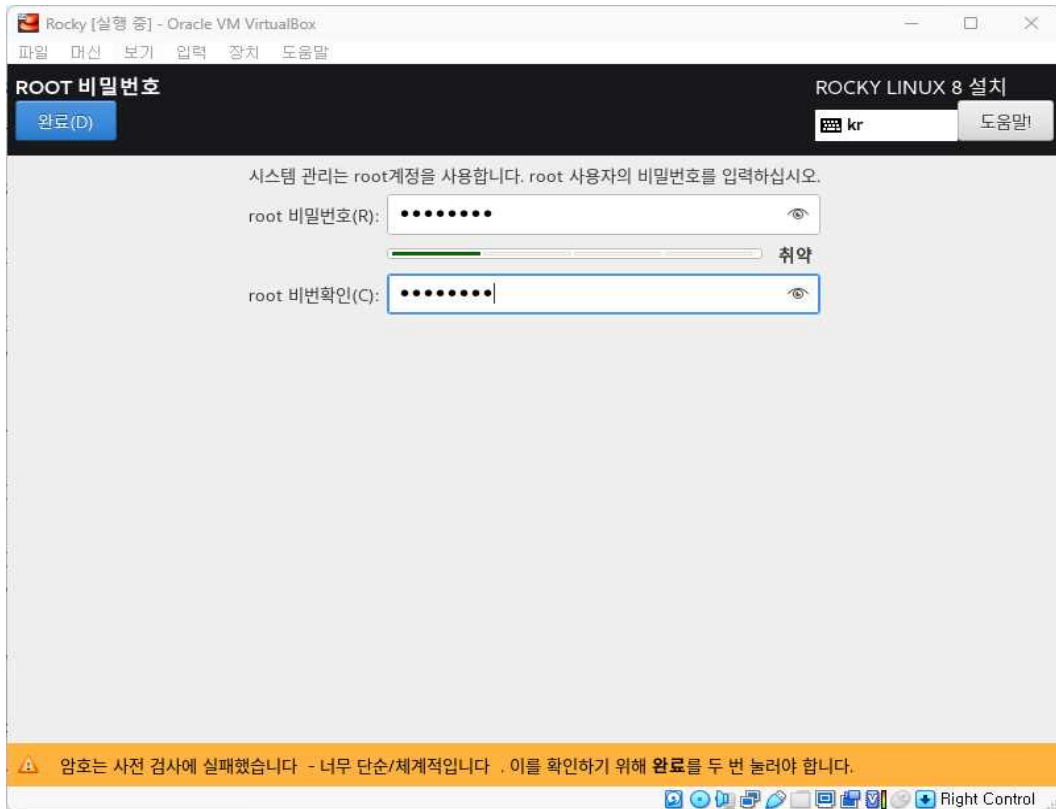
[그림 1.4] 언어 선택 화면

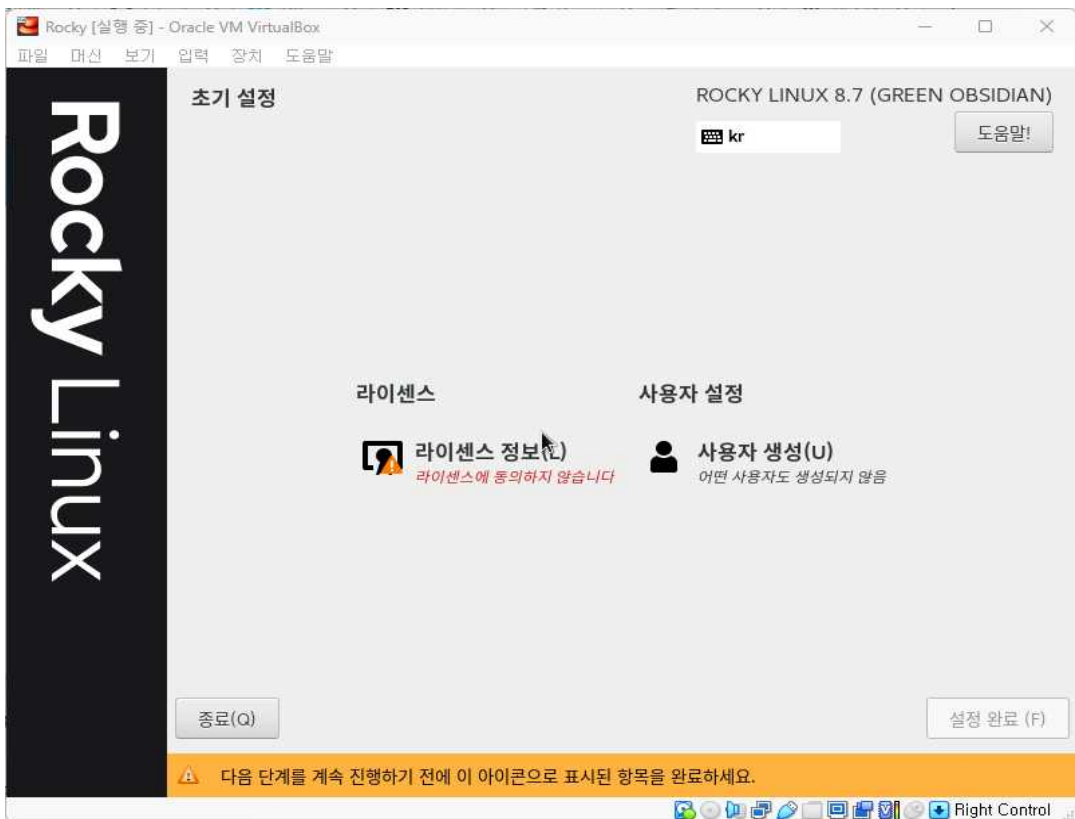
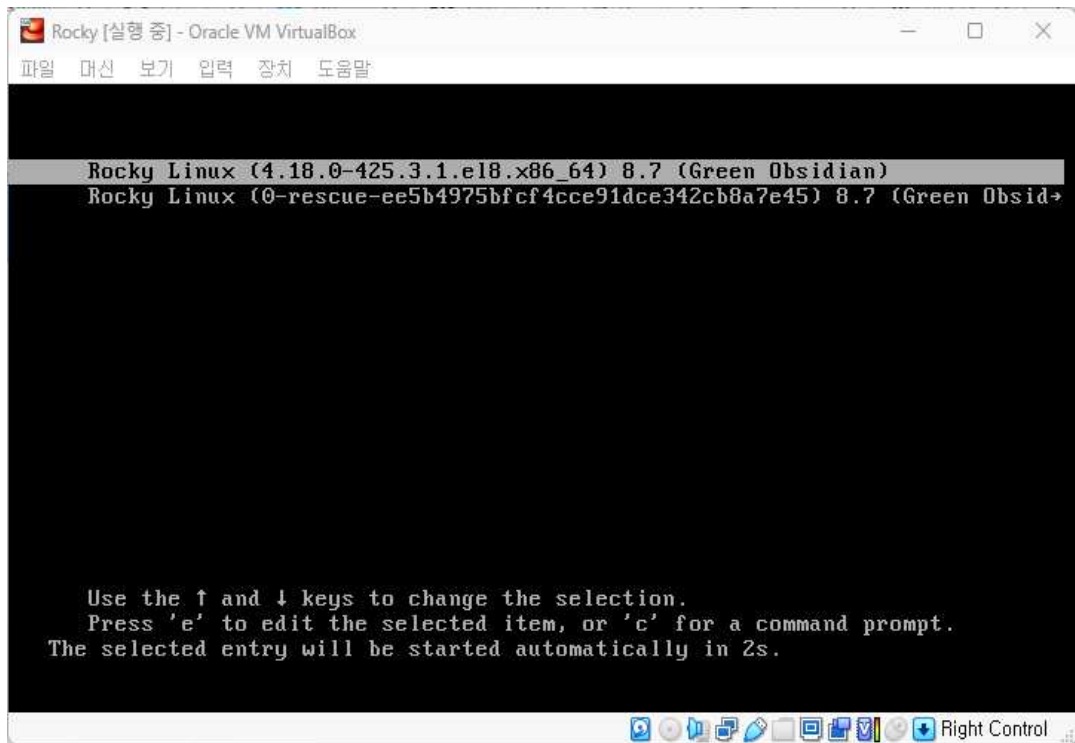


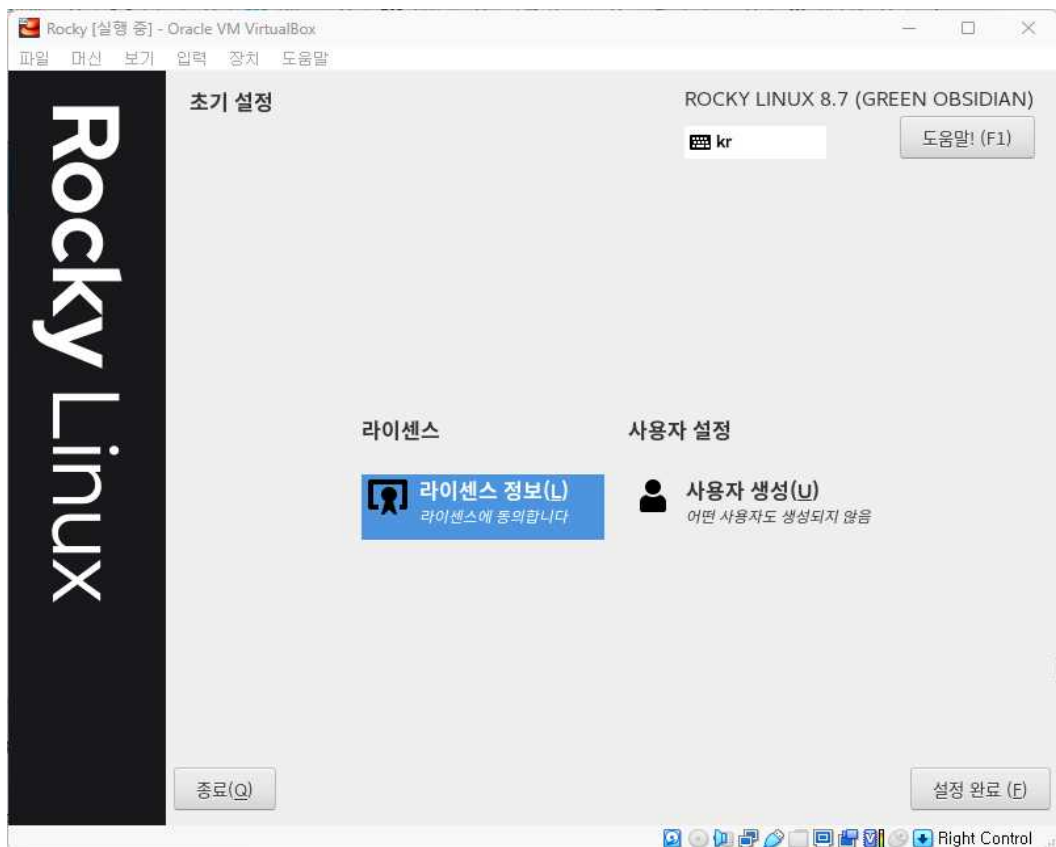
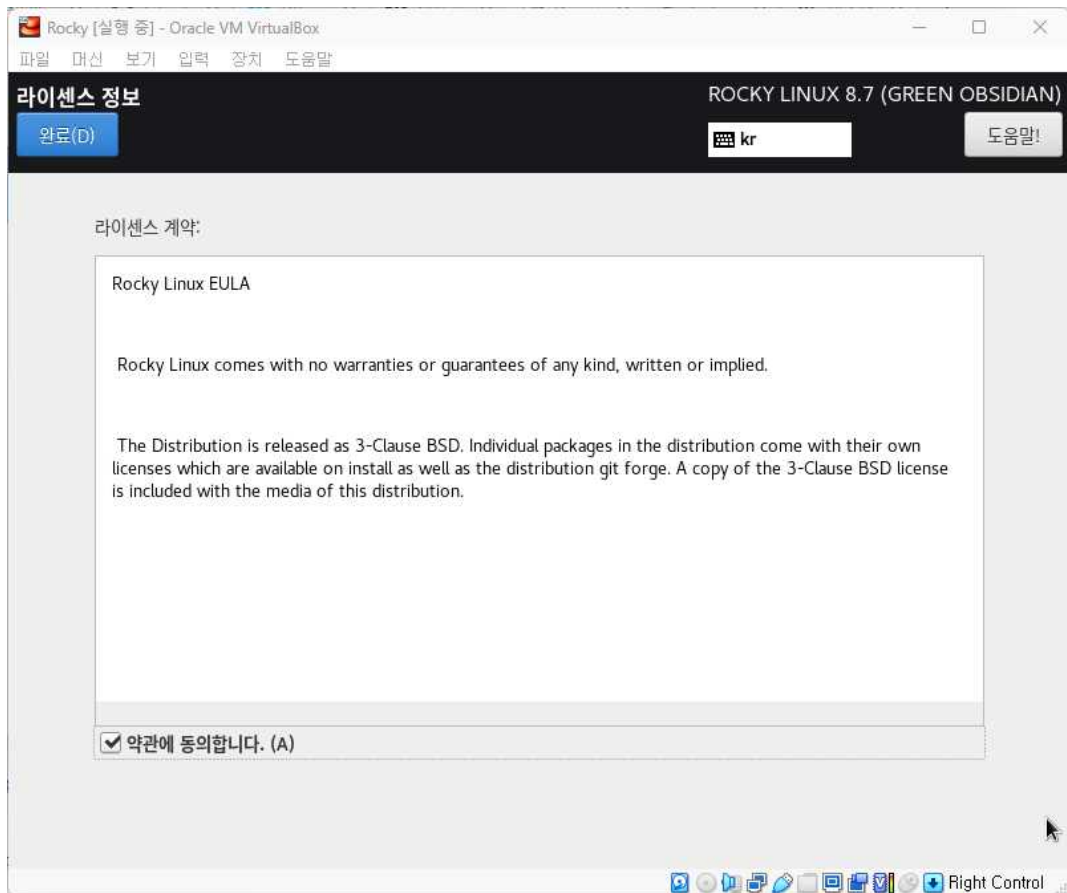
- rocky Linux 8 설치 과정에서 사용할 언어 선택
- 설치가 완료된 System에서 사용할 언어는 뒤에서 선택

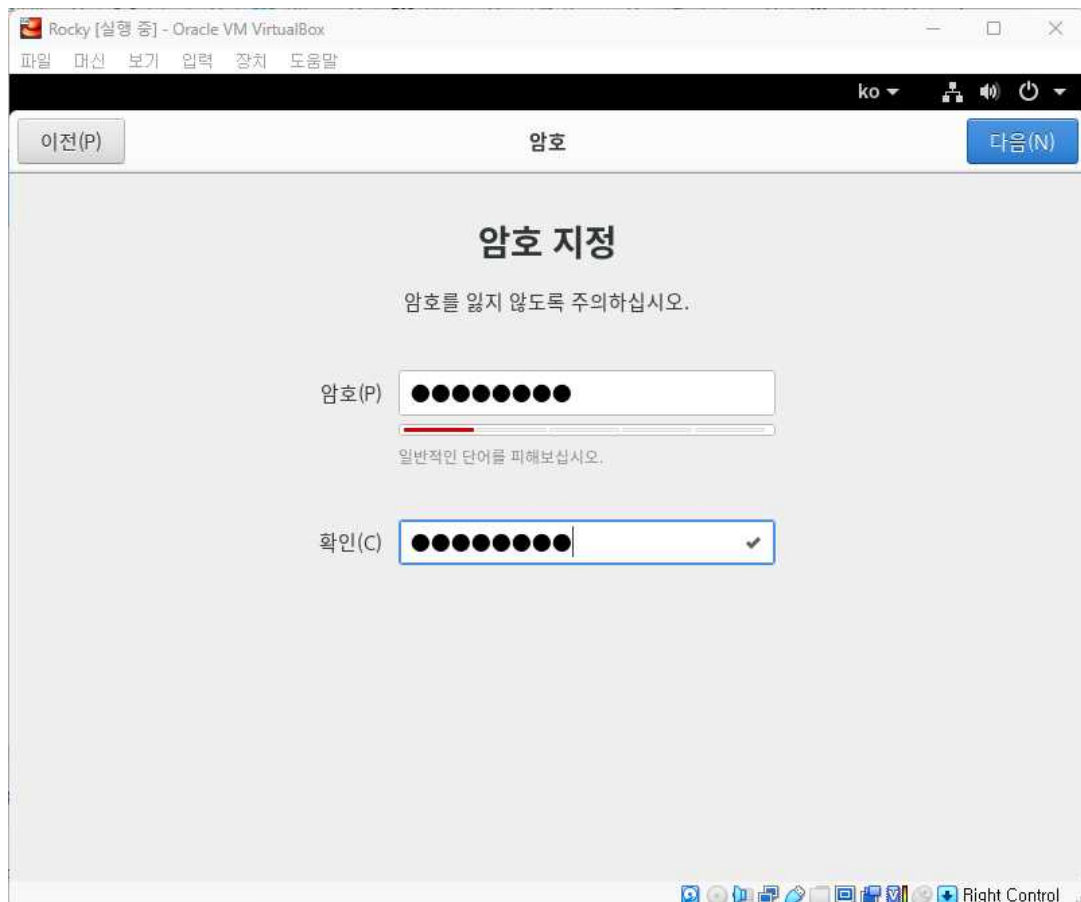
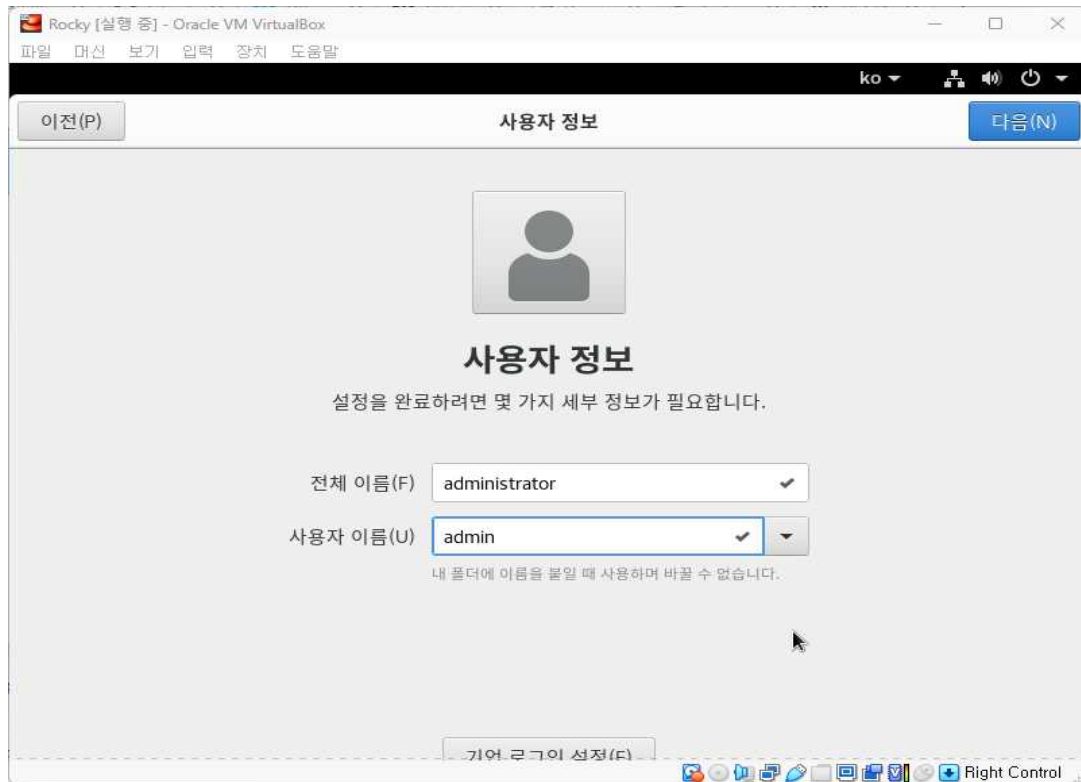


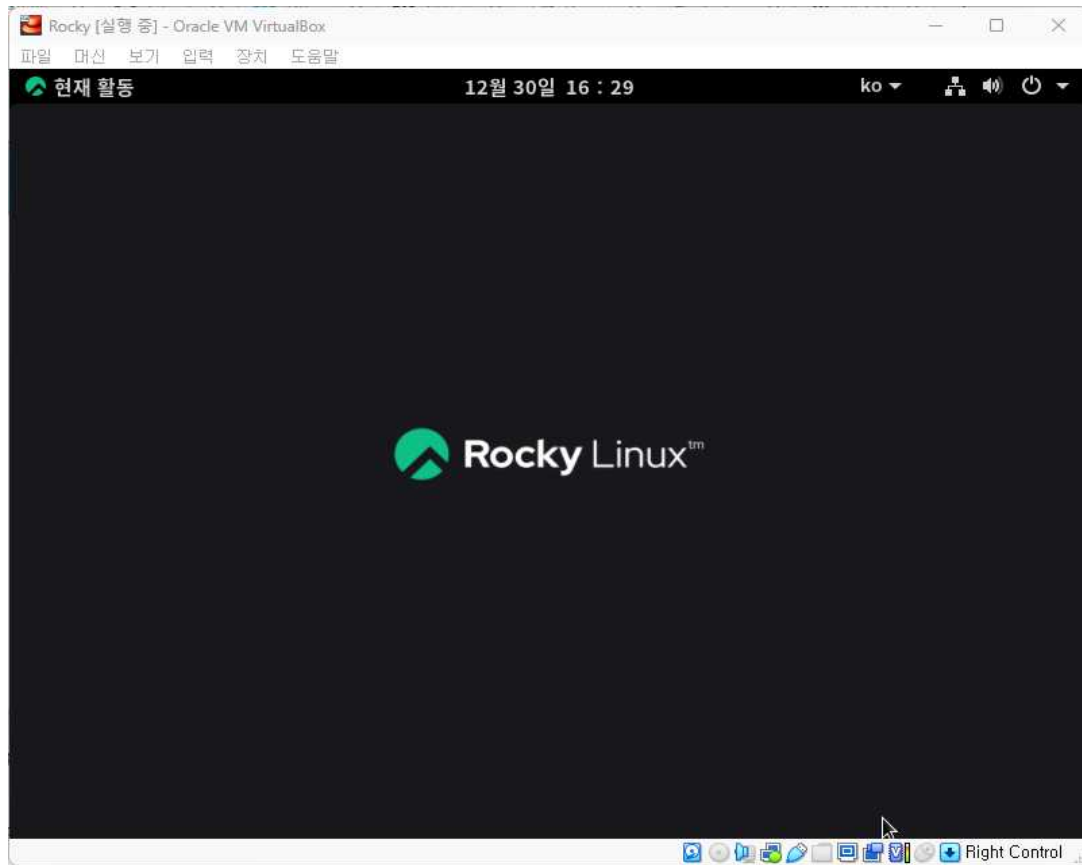












2.1 네트워크 설정

I. 네트워크 설정 파일

```
/etc/sysconfig/network-scripts/ifcfg-eth#
```

II. 부팅시에 NIC 활성화하기

- 부팅시 네트워크 설정에 사용될 설정 파일을 /etc/sysconfig/network-scripts 디렉터리에 생성
- 파일형식 : ifcfg-[인터페이스이름]

```
DEVICE=enp0s3
IPADDR=10.0.2.15
PREFIX=24
BROADCAST=110.0.2.255
ONBOOT=yes
BOOTPROTO=static
```

설치시 DHCP를 선택한 경우

```
TYPE=Ethernet
BOOTPROTO=dhcp
DEFROUTE=yes
NAME=enp0s3
UUID=ca2c5db5-f571-46a8-9e7c-5d867e3df315
DEVICE=enp0s3
ONBOOT=yes
```

- 네트워크 변경 적용 방법
systemctl restart NetworkManager

III. 네트워크 관련 명령들

- nmcli 명령 : NetworkManager에서 제공하는 명령
- ```
nmcli con show
nmcli con modify ens32 ipv4.address 192.168.30.150/24 ipv4.gateway 192.168.30.253 ipv4.dns 8.8.8.8
nmcli con up ens32
ip addr => IP주소 확인
```

---

```
nmcli con del ens32 => ens32 삭제
nmcli con add con-name ens32 ifname ens32 type ethernet ipv4.address 192.168.30.150/24
ipv4.gateway 192.168.30.253 ipv4.dns 8.8.8.8
- 인터넷이 안될 경우 다음 사항 확인
cat /etc/resolv.conf
echo "nameserver 8.8.8.8" > /etc/resolv.conf
```

- ip 명령

```
ip addr add 192.168.80.100/24 dev ens32 : IP주소 설정
ip route add default via 192.168.80.2 dev ens32 : 디폴트 루트 추가
ip link set ens32 up/down : 인터페이스 활성화/비활성화
```

- ifconfig 명령

- 네트워크 인터페이스 카드 셋업 관련 명령
- 커맨드라인에서 네트워크 설정을 확인하고, 변경할 수 있는 명령
- 사용형식:

```
ifconfig NIC인터페이스이름 IP주소 [netmask] [broadcast]
```

- 특정 인터페이스 활성화/비활성화 하기

```
ifconfig 장치명 up/down
```

- route 명령어

- 라우팅 테이블에 디폴트 게이트웨이, 네트워크를 추가/삭제할 수 있으며, 설정내용을 확인할 수 있다. 설정내용이 메모리에 저장되기 때문에 시스템이 재부팅되면 설정내용이 사라진다.
- 사용형식 :

```
route add/delete default gw <Gateway IP주소> dev <장치명>
route add/delete -net <네트워크 주소> netmask <netmask> dev <장치명>
```

- 라우팅 테이블 확인

```
route -n
netstat -rn
```

- 네트워크 진단 프로그램

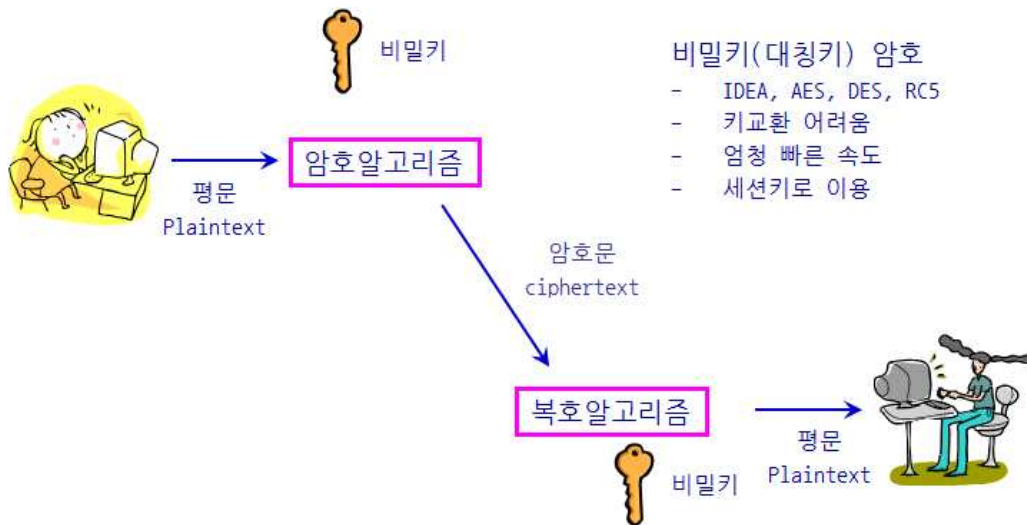
- ping, traceroute, netstat

### 3.1 SSH 개요

#### I. SSH란?

SSH(Secure Shell)이란, Telnet과 마찬가지로 원격지에 있는 시스템을 제어하기 위해 사용되는 프로토콜이다. 하지만 Telnet이 평문을 사용하여 보안에 취약한 반면, SSH는 주고 받는 데이터를 암호화하여 보안을 강화시켰다. SSH는 22번 포트를 사용한다.

#### (1) 암호화 방식



#### < 대칭키 암호화 >

위 그림에서와 같이 대칭키(비밀키) 암호화 방식은 데이터를 암호화하고 복호화하는데 동일한 키를 사용한다. 따라서 송신자와 수신자가 동일한 키를 가지고 있어야지만 데이터를 암호화하고 복호화할 수 있다.



#### < 비대칭키 암호화 >

- 반면 비대칭키 암호화 방식은 서로 다른 두 키를 이용하여 암호화와 복호화에 사용을 하게 된다. 이 두 키를 공개키, 개인키라고 불러서 공개키 암호화 방식이라고도 한다.



#### < SSH를 이용한 원격지 시스템 접속 과정 >

---

## II . OpenSSH

Secure Shell 표준을 비상업용으로 구현한 자유 소프트웨어

SSH 프로토콜: IETF에서 제정한 국제 표준

-개발자들은 자유롭게 SSH 클라이언트 소프트웨어 개발 가능

원격 로그인 과정에서 발생할 수 있는 패스워드 보안 문제를 해결

서버와 클라이언트 간의 통신을 암호화 함

-private/public key cryptography

### (1) OpenSSH 설치 확인 및 설치

```
dnf install openssh-server (기본적으로 설치되어 있음)
```

### (2) SSH 데몬 동작 확인 및 서비스 시작하기

```
systemctl status sshd
```

```
systemctl start sshd
```

### (3) SSH 설정 파일

- /etc/ssh/sshd\_config

- ‘#’ : 주석이 아니고 default 설정

- root 로그인 허용여부

- PermitRootLogin [yes|no]

- 패스워드를 이용한 로그인 허용 여부

-

### (4) SSH에서 사용할 key 생성 및 key를 이용한 접속

SSH1은 RSA를 사용하고 SSH2에서는 DSA, ECDSA, RSA를 이용. 지정하지 않으면 SSH2에서 사용하기 위한 RSA 생성

key 생성은 어디에서 해도 상관없음. 접속하고자하는 서버에 공개키, 클라이언트에 개인키가 있으면 됨.

```
ssh-keygen -t dsa
```

Generating public/private dsa key pair.

Enter file in which to save the key (/root/.ssh/id\_dsa):

Enter passphrase (empty for no passphrase):

Enter same passphrase again:

Your identification has been saved in /root/.ssh/id\_dsa.

Your public key has been saved in /root/.ssh/id\_dsa.pub.

The key fingerprint is:

6e:85:fc:c3:84:25:57:46:97:d8:aa:da:57:0b:fc:95 root@localhost

-private key 는 \$HOME/.ssh/id\_dsa에 저장

-public key 는 \$HOME/.ssh/id\_dsa.pub에 저장

Public key(id\_rsa.pub)를 authorized\_keys로 파일명을 변경하고, 서버에 복사한다.

추후에 서버에 ssh로 접속하려는 시스템은 id\_rsa.pub의 내용을 authorized\_keys에 추가하면 된다.

이제 ssh를 이용해서 서버에 접속할 때 해당 사용자의 패스워드가 아니라 ssh-keygen을 이용해 생성한 passphrase를 입력한다.

- passphrase 입력없이 접속하기

ssh를 사용할 때 ssh-agent를 이용하면 passphrase 입력없이 로그인 가능하다. 다음 작업을 ssh 서버에 접속하려는 클라이언트에서 진행한다.

>> 자식 셸을 만들지 않고 현재 셸에서 실행

```
eval `ssh-agent`
```

```
Agent pid 14057
```

>> passphrase를 미리 입력해 놓는다.

```
ssh-add id_dsa
```

```
Enter passphrase for id_dsa:
```

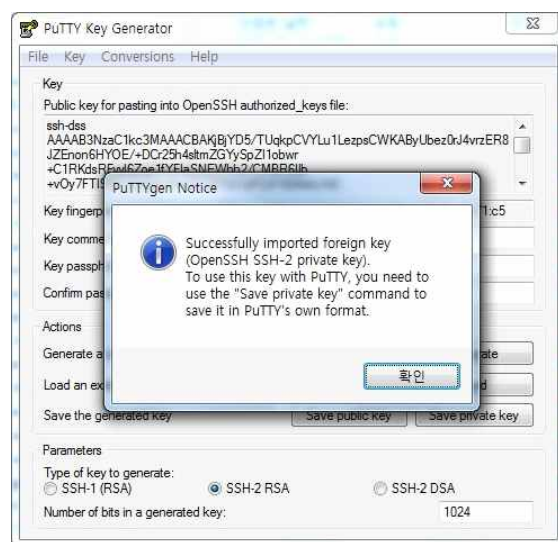
```
Identity added: id_dsa (id_dsa)
```

이제 ssh 서버에 passphrase 입력없이 로그인이 된다.

(5) 윈도우에서 putty를 이용하여 원격 리눅스 서버에 접속하기(passphrase 인증)

앞에서와 같이 ssh-keygen을 이용하여 개인키(id\_rsa)와 공개키(id\_rsa.pub)를 생성한 후, 공개키(id\_rsa.pub)를 authorized\_keys를 변경하고, 개인키(id\_rsa)를 윈도우로 복사한다.

이제 윈도우에서 그림과 같이 puttygen을 실행해서 복사해 온 개인키(id\_rsa)를 불러오기한다. passphrase가 있다면 입력한다. 성공하면 다음과 같은 화면이 나타난다.

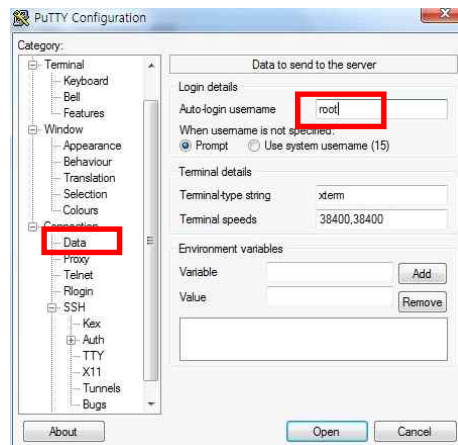


Save private key를 눌러서 Putty에서 사용할 수 있도록 저장한다. 저장한 후 Putty를 실행하고 아래 그림과 같이 SSH-Auth 부분을 클릭해서 저장된 개인키를 불러온다. 이제는 passphrase를

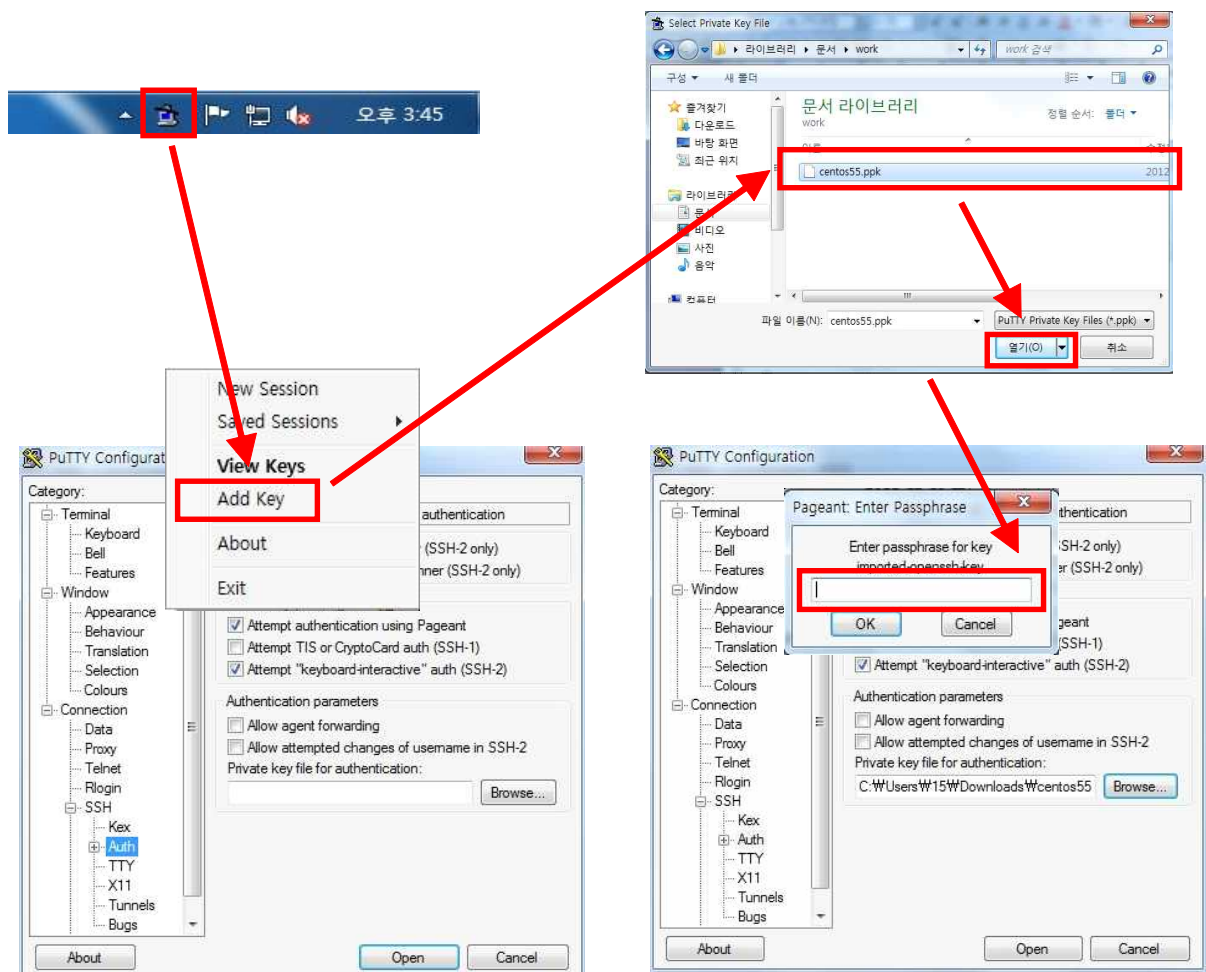


이용해 접속이 가능하다.

ssh 접속시 고정된 사용자를 이용한다면 다음과 같이 data부분에서 username을 입력하여 저장한다.



마지막으로 passphrase 입력없이 로그인을 하기 위해서 pageant를 이용한다. pageant를 실행하면 해당 아이콘이 시스템 트레이에 추가된다. 추가된 아이콘을 오른쪽 클릭하면 나타나는 메뉴에서 Add Key를 선택하고 저장되어 있는 개인키를 선택하면 passphrase를 요구하고, 입력하게 되면 이때부터 passphrase 입력없이 로그인이 가능하게 된다.



## 4.1 부팅 순서

### I. 부트 프로세스

#### (1) BIOS 프로그램 실행

- 시스템 전원 공급을 통해 CPU 동작
- CPU는 메인보드의 ROM-BIOS에 있는 BIOS 프로그램을 실행
- ROM-BIOS가 하는 기능에는 자체지단기능과 시스템부팅기능 두 가지가 있다.

#### (2) POST(Power On Self Test)

- CMOS 검사, 메모리 테스트, 그래픽 카드, 키보드, 마우스 등의 장치 유무 검사

#### (3) 부트로더 검색

- 하드디스크의 가장 첫 번째 부분의 MBR(Master Boot Record) 확인
- MBR에 존재하는 부트로더(Boot Loader) 인식 및 로딩

#### (4) 부트로더 실행

- /boot/grub2/grub.cfg 파일을 읽어 부팅할 커널 선택
- 부트로더(GRUB, LILO 등)는 커널을 메모리에 로딩

#### (5) 커널 이미지(vmlinuz) 로딩

- /boot/grub2/grub.cfg 파일을 읽어 부팅할 커널 선택
- /boot/vmlinuz-4.18.0-425.3.1.el8.x86\_64
- 커널 이미지 로딩 후 시스템 제어권을 커널에 넘김.
- 로드된 커널은 swapper(pid 0인 프로세스) 호출
- swapper는 커널 이미지 압축해제 및 커널이 사용할 각 장치드라이브들을 초기화

#### (6) root("/") 파일시스템 마운트

- 하드웨어 검색 후 "/" 파일시스템 마운트
- 파일시스템 안전을 위해 읽기 모드(Only-read)로 마운트 후 파일시스템 검사를 한다.
- "/" 파일시스템 검사 후 읽기/쓰기 모드로 다시 마운트한다.

#### (7) systemd 프로세스 실행

- PID가 1인 systemd 프로세스 실행, systemd 프로세스부터 모든 프로세스가 실행
- systemd 프로세스는 사용자가 시스템을 사용할 수 있도록 환경을 만들어 줌

#### (8) systemd에 의해 콘솔(첫번째 가상콘솔) 로그인 프롬프트 실행.

## II. GRUB2(Grand Unified Bootloader2)

### (1) 부트로더

- 컴퓨터가 시작될 때 처음 실행되는 프로그램
- 운영체제를 불러오는 역할

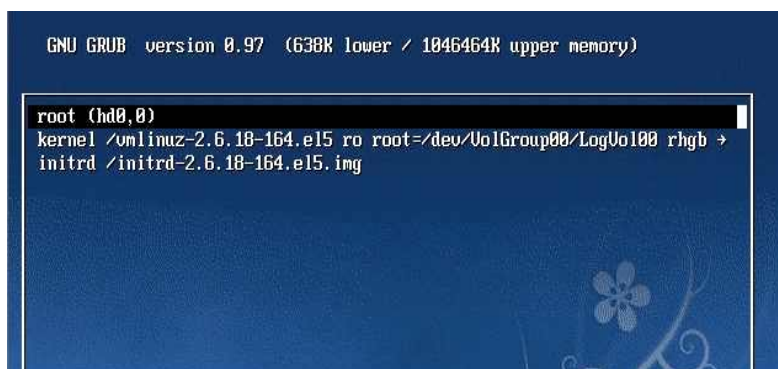
### (2) GRUB2

- GNU에서 개발된 부트로더로 레드햇 7.2 버전부터 지원
- 하드 디스크의 첫 번째 섹터인 MBR에 저장
- 리눅스 계열에서 가장 많이 사용되는 부트로더, Unix에서도 사용함.
- 장애 발생 시에 응급복구를 할 수 있도록 지원

[그림 4.1] rocky 부트로더 확인



- 부트로더(GRUB) 확인: 부팅 시 화면과 같이 나올 때 방향키 ⇒ 첫 번째 화면 출력
  - b: 선택된 부트 과정의 명령어로 부트
  - e: 선택된 부트 과정의 명령어 편집
  - c: GRUB 명령어 편집
  - o: 새로운 행을 커서 다음 행에 삽입
  - d: 선택된 행 삭제
  - esc: GRUB 메인 메뉴



- 부팅 메뉴 편집: 첫 번째 화면 ⇒ "e" 입력

---

### (3) 부팅 후 GRUB 확인

- 부팅이 완료된 후 GRUB 확인 및 버전 확인

```
[root@rocky ~]# grub2-install --version
grub2-install (GRUB) 2.03
```

### (4) GRUB 기본설정 파일

/etc/default/grub 파일을 변경

```
GRUB_TIMEOUT=5
GRUB_DISTRIBUTOR="$(sed 's, release .*,g' /etc/system-release)"
GRUB_DEFAULT=saved
GRUB_DISABLE_SUBMENU=true
GRUB_TERMINAL_OUTPUT="console"
GRUB_CMDLINE_LINUX="resume=/dev/mapper/rl-swap rd.lvm.lv=rl/root
rd.lvm.lv=rl/swap rhgb quiet"
GRUB_DISABLE_RECOVERY="true"
GRUB_ENABLE_BLSCFG=true
```

GRUB에 password를 설정하는 방법은 아래와 같다.

- 현재 root에 대한 패스워드 설정을 처음하는 경우

```
[root@rocky ~]# grub2-set-password
```

설정 후 다음 명령을 이용하여 반영시킨다.

```
grub2-mkconfig -o /boot/grub2/grub.cfg
```

### (5) root 비밀번호 분실시 변경

커널 옵션값 수정

rhgb quiet LANG=ko\_KR.UTF-8 삭제 후 init=/bin/bash 입력 후 Ctrl + X로 재부팅

부팅된 후, 다음과 같이 명령 실행

```
mount -o remount,rw /
```

password 명령으로 패스워드 변경

touch /.autorelable => 생성하지 않으면 패스워드가 변경되지 않는다.

exit 또는 exec /sbin/reboot

### (6) GRUB 패스워드 변경

```
#grub2-mkpasswd-pbkdf2
```

패스워드 2번 입력

```

출력된 해시값을 복사한 후
vi /etc/grub2.cfg
BEGIN /etc/grub.d/10_linux
set superusers="root"
export superusers
password_pbkdf2 root
grub.pbkdf2.sha512.10000.24E8207980BBB7F923BF48DA0E8C68F59A33682A58761C0EB93ADB47CDFB04146CD889476
B1168A360FD18C334FFB311821CC010B9483F0FFCD3885443B98843.356DFBED3C6AAD67046A6FD846E29E92218B953D03
D42E2B1887226572419DB1A7A7786A01288AD252FB05FA32B5B8419927F9C27E23B366F3D038EAF4506A61

```

저장 후 reboot

GRUB 화면에서 e를 누르면 username을 물어보고 이때 해시값에 해당하는 패스워드 입력하면 GRUB 옵션창을 들어간다.

## 4.2 systemd 프로세스 실행

1. /etc/systemd/system.conf

(1)

- systemd 프로세스 설정 파일

- Default Run Level

→ graphical.target

→ Default Run Level 값 변경: multiuser.target으로 변경 시 Text 모드로 부팅

```

[root@rocky system]# ls -l default.target
lrwxrwxrwx. 1 root root 40 1월 3 00:14 default.target ->
/usr/lib/systemd/system/graphical.target

```

- Default Run Level 실행 후 실행되는 스크립트(/etc/systemd/system/graphical.target.wants)

→ 시스템 초기화 스크립트

## 4.3 기본 실행레벨(Default RunLevel)

; 실행레벨은 init 프로세스에 의해서 시스템이 어떻게 작동할 것인가를 결정해주는 실행 값.

- 실행레벨의 값과 의미

| 실행레벨(RunLevel) |                              |
|----------------|------------------------------|
| 0              | 시스템 중지(halt)                 |
| 1              | 단일 사용자 모드, 시스템 초기화없이 바로 쉘 부팅 |
| 2              | NFS 기능을 제외한 다중 사용자 모드        |
| 3              | 모든 기능을 포함한 다중 사용자 모드, 콘솔 부팅  |
| 4              | 시스템 예약(사용하지 않음.)             |
| 5              | X-Window로 부팅.                |
| 6              | 시스템 리부팅(reboot)              |

---

## Chapter 05. 방화벽(firewalld)

---

Linux 커널 2.4부터 netfilter로 프레임워크가 교체 되었다.

iptables은 netfilter 프레임워크의 최상단에 위치하는 사용자 레벨의 프로그램으로 시스템 관리자는 iptables 명령어로 리눅스 서버로 들어오고 나가는 패킷을 필터링하거나 포트 포워딩을 설정할 수 있으며 방화벽으로도 사용할 수 있다.

iptables 는 숙련된 관리자가 아니면 사용이 어려운 단점이 있었는데 이런 문제를 해결하고자 RHEL/CentOS 7 부터는 방화벽을 firewalld 라는 데몬으로 교체하였고 이에 따라 사용자 레벨의 프로그램은 iptables 명령어 대신 명령행에서는 firewall-cmd, GUI 환경에서는 firewall-config 를 사용하게 되었습니다.

# 서비스 시작

```
[root@localhost ~]# systemctl start firewalld
```

# 서비스 재시작

```
[root@localhost ~]# systemctl restart firewalld
```

# 서비스 중지

```
[root@localhost ~]# systemctl stop firewalld
```

# 서비스 등록 : 부팅시 firewalld 활성화

```
[root@localhost ~]# systemctl enable firewalld
```

# 서비스 상태 확인

```
[root@localhost ~]# systemctl status firewalld
```

# firewalld 정책 저장 또는 적용

```
[root@localhost ~]# firewall-cmd --reload
```

# firewalld 설정파일 경로

```
[root@localhost ~]# vi /etc/firewalld/firewalld.conf
```

# 사전 정의된 zone 목록 확인

```
[root@localhost ~]# firewall-cmd --get-zones
```

```
block dmz drop external home internal public trusted work
```

# 기본 설정 zone 확인

```
[root@localhost ~]# firewall-cmd --get-default-zone
```

```
public
```

# 기본 zone 변경

```
[root@localhost ~]# firewall-cmd --set-default-zone=external
```

---

# 현재 설정 확인

```
[root@localhost ~]# firewall-cmd --list-all
```

# 전체 zone 목록 상세 확인

```
[root@localhost ~]# firewall-cmd --list-all-zones
```

# 네트워크 인터페이스에 적용된 zone 리스트 확인 (변경 전)

```
[root@localhost ~]# firewall-cmd --get-active-zone
```

external

interfaces: enp0s8 enp0s3

# 네트워크 인터페이스에 zone 변경

```
[root@localhost ~]# firewall-cmd --change-interface=enp0s3 --zone=public
```

# 네트워크 인터페이스에 적용된 zone 리스트 확인 (변경 후)

```
[root@localhost ~]# firewall-cmd --get-active-zone
```

external

interfaces: enp0s8

public

interfaces: enp0s3

# 신규 zone 추가

```
[root@localhost ~]# firewall-cmd --permanent --new-zone=test
```

success

# 기존 zone 삭제

```
[root@localhost ~]# firewall-cmd --permanent --delete-zone=test
```

success

# 사전 등록된 서비스 목록 확인

```
[root@localhost ~]# firewall-cmd --get-services
```

# 사전 정의된 서비스 파일 위치

```
[root@localhost ~]# firewall-cmd --get-services
```

# 특정 zone에 허용된 서비스 리스트 확인 (서비스 추가 전)

```
[root@localhost ~]# firewall-cmd --list-service --zone=public
```

cockpit dhcpv6-client ssh

# 특정 zone에 서비스 추가 (http, https, dns, telnet)

```
[root@localhost ~]# firewall-cmd --permanent --zone=public --add-service=http
```

```
[root@localhost ~]# firewall-cmd --permanent --zone=public --add-service=https
```

```
[root@localhost ~]# firewall-cmd --permanent --zone=public --add-service=dns
```

---

```
[root@localhost ~]# firewall-cmd --permanent --zone=public --add-service=telnet
```

```
특정 zone에 서비스 삭제 (telnet)
```

```
[root@localhost ~]# firewall-cmd --permanent --zone=public --remove-service=telnet
```

```
서비스 추가 후 서비스 적용을 위해 방화벽 재시작
```

```
[root@localhost ~]# firewall-cmd --reload
```

```
특정 zone에 허용된 서비스 리스트 확인 (서비스 추가 후)
```

```
[root@localhost ~]# firewall-cmd --list-service --zone=public
```

```
cockpit dhcpv6-client dns http https ssh
```

```
허용 포트 리스트 확인 (추가 전)
```

```
[root@localhost ~]# firewall-cmd --list-port --zone=public
```

```
특정 포트 TCP 8080, 8081 추가
```

```
[root@localhost ~]# firewall-cmd --permanent --zone=public --add-port=8080/tcp
```

```
[root@localhost ~]# firewall-cmd --permanent --zone=public --add-port=8081/tcp
```

```
특정 포트 TCP 8081 삭제
```

```
[root@localhost ~]# firewall-cmd --permanent --zone=public --remove-port=8081/tcp
```

```
포트 적용을 위한 방화벽 재시작
```

```
[root@localhost ~]# firewall-cmd --reload
```

```
허용 포트 리스트 확인 (추가 후)
```

```
[root@localhost ~]# firewall-cmd --list-port --zone=public
```

```
8080/tcp
```



## Chapter 06. vi Editor

### 6.1 vi Editor 개요

#### 1. vi Editor 사용

- Linux에서 지원하는 편집기로 vi(visual editor), emacs, pico 등 다양한 문서 편집기가 존재
- vi는 유닉스의 표준 편집기이기도 하며, 리눅스의 기본 편집기이다.
- vi를 이용하여 기존 파일 내용을 수정하거나 생성. 설정파일들이 텍스트 형태이기 때문에 리눅스 시스템을 관리하기 위해서는 편집기의 자유로운 사용이 필수.
- RPM을 이용한 vim(기존 vi Editor의 향상 버전-Vi IMproved) 설치

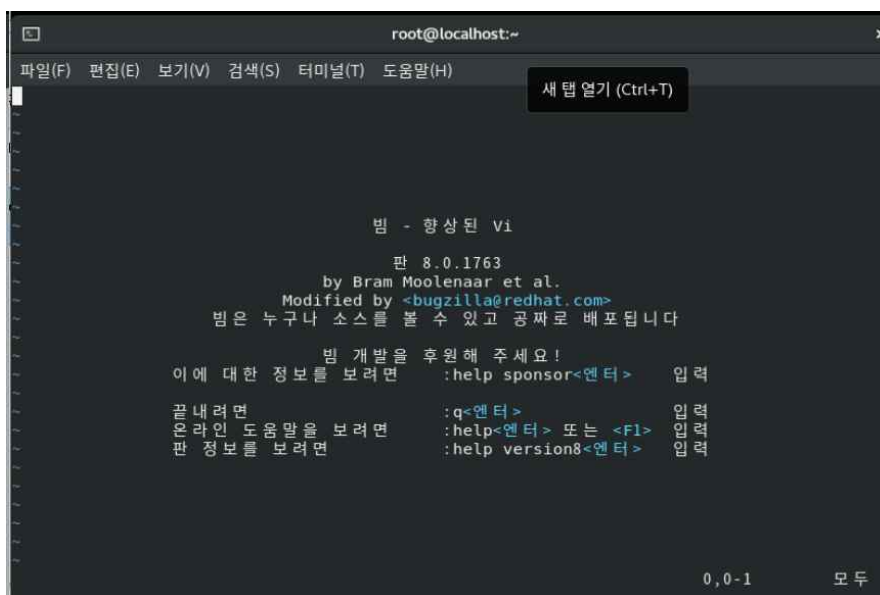
- [root@rocky ~]# rpm -qa | grep vim  
→ 설치 확인

[그림 6.1] vim 설치 확인



- [root@rocky ~]# rpm -ivh vim-common-(버전).rpm
- [root@rocky ~]# rpm -ivh vim-minimal-(버전).rpm
- [root@rocky ~]# rpm -ivh vim-enhanced-(버전).rpm

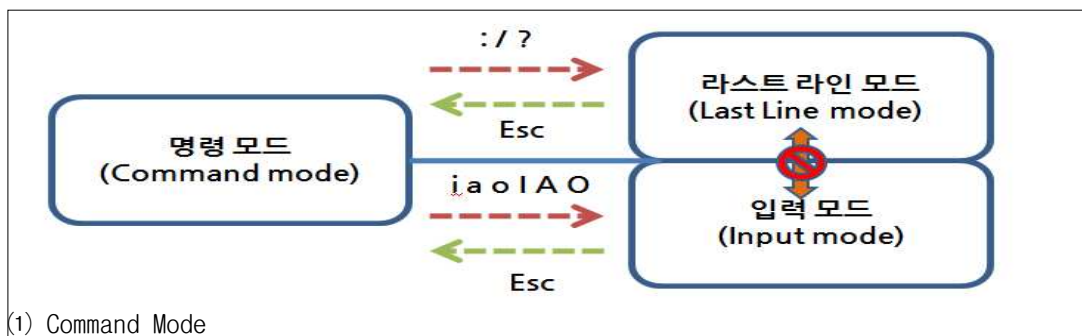
- 명령어 위치 : /usr/bin/vim
- [root@rocky ~]# mv /bin/vi /bin/vi\_old
- [root@rocky ~]# ln -s /usr/bin/vim /bin/vi
- [root@rocky ~]# vi



[그림 6.2] vi Editor 실행 화면

- 명령어 사용 방법 : # vi "File Name"
- 사용 Ex.>
  - [root@rocky ~]# vi /etc/hosts
    - /etc/hosts 파일 내용을 확인 or 편집하고자 할 때
  - [root@rocky ~]# vi test.txt
    - 현재 위치한 디렉토리에 test.txt 파일이 존재하지 않을 경우 test.txt 파일을 생성하며 동시에 원하는 내용으로 편집 가능

## II. vi Editor Mode



- vi Editor의 기본 모드
- 커서의 이동과 입력·편집 모드, 명령 모드로 전환하기 위한 대기 상태
- "Input Mode, Last Line Mode"에서 Command Mode로 전환하기 위해 "ESC Key" 사용

### (2) Input Mode

- Command Mode에서 "a(A), i(I), o(O)"중 하나를 입력
- 파일에 새로운 내용을 입력하거나 기존의 내용을 삭제하는 등의 편집이 가능한 상태

### (3) Last Line Mode

- Command Mode에서 ":(콜론)" or "/(슬래시)" 등을 입력
- 파일 내용 편집 후 저장, 편집 취소, vi Editor 종료 등의 명령을 실행

## 6.2 vi Editor 명령어

### I. Command Mode 명령어

#### (1) 기본 명령어

- 방향키를 이용할 수도 있지만 "h", "j", "k", "l" 키를 사용하여 커서를 이동
- "h" : 커서를 왼쪽으로 이동
- "j" : 커서를 아래쪽으로 이동
- "k" : 커서를 위쪽으로 이동

- "l" : 커서를 오른쪽으로 이동
- 앞에 원하는 숫자를 붙여서 입력 가능(예: "3h" : 왼쪽으로 세 칸 이동)

## (2) 고급 명령어

| 명령어      | 의미          | 명령어      | 의미               |
|----------|-------------|----------|------------------|
| Ctrl + f | 한 화면 아래로 이동 | Ctrl + d | 한 화면의 반만큼 아래로 이동 |
| Ctrl + b | 한 화면 위로 이동  | Ctrl + u | 한 화면의 반만큼 위로 이동  |

|    |                 |           |             |
|----|-----------------|-----------|-------------|
| G  | 파일의 제일 마지막으로 이동 | [number]G | 지정한 라인으로 이동 |
| gg | 파일의 가장 처음으로 이동  | 0         | 라인의 시작으로 이동 |
| ^  | 라인의 시작으로 이동     | \$        | 라인의 끝으로 이동  |
| H  | 화면의 처음으로 이동     | M         | 화면의 중간으로 이동 |
| L  | 화면의 마지막으로 이동    | [number]b | 한 단어 뒤로 이동  |

| 명령어       | 의미                               | 명령어       | 의미             |
|-----------|----------------------------------|-----------|----------------|
| [number]B | 특수문자/기호 등은 단어로 치지 않고 한 단어 뒤로 이동  | [number]w | 한 단어 앞으로 이동    |
| [number]W | 특수문자/기호 등은 단어로 치지 않고 한 단어 앞으로 이동 | [number]+ | 다음 라인의 처음으로 이동 |
| [number]- | 이전 라인의 처음으로 이동                   |           |                |

## II . Input Mode 명령어

### (1) 기본 명령어

- "i" : 현재 커서 위치의 왼쪽에 입력
- "I" : 현재 커서가 있는 문장의 가장 왼쪽에 입력
- "a" : 현재 커서 위치의 오른쪽에 입력
- "A" : 현재 커서가 있는 문장의 가장 오른쪽에 입력
- "o" : 현재 커서가 있는 문장의 아래쪽에 입력
- "O" : 현재 커서가 있는 문장의 위쪽에 입력

### (2) 고급 명령어

| 명령어        | 의미                                                        |
|------------|-----------------------------------------------------------|
| r          | 한 글자 수정 후 command 모드로 전환                                  |
| R          | 현재 위치부터 덮어쓰기                                              |
| [number]s  | 원하는 글자 수를 삭제한 후 Input 모드로 전환                              |
| cw         | 한 단어를 삭제 후 Input 모드로 전환                                   |
| u          | 바로 전에 실행한 내용 되돌리기                                         |
| U          | 특정 행의 실행된 내용을 모두 복구, 바로 전에 실행한 명령 반복 사용                   |
| [number]J  | 두 행을 한 행으로 붙임                                             |
| [number]y  | 버퍼에 문자열 복사                                                |
| [number]Y  | 버퍼에 한 라인 복사                                               |
| [number]yy | 버퍼에 한 라인 복사                                               |
| p          | 'y' 실행 시 현재 커서 위치의 앞으로 복사, 'Y' or 'yy' 실행 시 현재의 아래 라인에 복사 |
| P          | 'y' 실행 시 현재 커서 위치의 뒤로 복사, 'Y' or 'yy' 실행 시 현재의 위 라인에 복사   |
| [number]db | 현재 커서 위치의 앞으로 한 단어 삭제                                     |
| [number]dw | 현재 커서 위치의 뒤로 한 단어 삭제                                      |
| [number]dd | 현재 커서가 위치한 라인 삭제                                          |
| [number]x  | 현재 커서가 위치한 문자 삭제                                          |
| Shift + D  | 현재 커서 위치로부터 라인 끝까지 삭제                                     |

### III. Last Line Mode 명령어

- vi Editor의 각종 환경 설정, 파일 불러오기, 파일 저장, vi Editor 종료 등이 가능

#### (1) vi Editor 환경 설정 명령어

| 명령어       | 의미       | 명령어           | 의미       |
|-----------|----------|---------------|----------|
| :set nu   | 행 번호 보이기 | :set number   | 행 번호 보이기 |
| :set nonu | 행 번호 감추기 | :set nonumber | 행 번호 감추기 |

| 명령어                     | 의미                        | 명령어             | 의미               |
|-------------------------|---------------------------|-----------------|------------------|
| :set autoindent         | 들여 쓰기                     | :set ai         | 들여 쓰기            |
| :set noautoindent       | 들여 쓰기 제거                  | :set noai       | 들여 쓰기 제거         |
| :set nolist             | 문단 및 조판 부호 미출력            | :set list       | 문단 및 조판 부호 출력    |
| :set<br>window=[number] | 한 화면당 행의 개수               | :set w=[number] | 한 화면당 행의 개수      |
| :set all                | 전체 환경 설정 확인               | :set            | 전체 환경 설정 확인      |
| :set noignorecase       | 검색할 때 대소문자 구분             | :set ignorecase | 검색할 때 대소문자 구분 없음 |
| :set showmode           | 현재 작업 모드 표시               | :set noshowmode | 현재 작업 모드 표시 해제   |
| :set ts=[number]<br>Tab | [Tab] 키의 공백 넓이 지정(기본 값:8) |                 |                  |

#### (2) 검색 및 수정 명령어

| 명령어                                                | 의미                               |
|----------------------------------------------------|----------------------------------|
| /문자열                                               | 커서가 위치한 라인 아래쪽으로 입력한 문자열을 검색     |
| /                                                  | 아래쪽으로 문자열 계속 검색                  |
| ?문자열                                               | 커서가 위치한 라인 위쪽으로 입력한 문자열을 검색      |
| ?                                                  | 위쪽으로 문자열 계속 검색                   |
| : <b>[number]</b>                                  | 지정한 라인으로 이동                      |
| :s/찾는 문자열/변경 문자열/g                                 | 커서가 위치한 라인에서 문자열을 찾아 지정한 문자열로 변경 |
| : <b>[number]</b> s/찾는 문자열/변경 문자열/g                | 지정한 라인에서 문자열을 찾아 지정한 문자열로 변경     |
| : <b>[시작 number],[종료 number]</b> s/찾는 문자열/변경 문자열/g | 지정한 라인 사이에서 문자열을 찾아 지정한 문자열로 변경  |
| :%s/찾는 문자열/변경 문자열/gc                               | %는 전체를 의미하며, c는 변경 전 확인          |
| :n                                                 | 여러 개의 파일을 열었을 경우 다음 파일을 읽어옴      |
| :N                                                 | 여러 개의 파일을 열었을 경우 이전 파일을 읽어옴      |

### (3) 저장 및 종료 명령어

| 명령어                                                   | 의미                          |
|-------------------------------------------------------|-----------------------------|
| :w                                                    | 저장하기                        |
| :w <b>[file name]</b>                                 | 지정한 파일명으로 저장                |
| : <b>[시작 number],[종료 number]</b> w <b>[file name]</b> | 지정한 라인 사이의 내용을 지정한 파일명으로 저장 |
| :q                                                    | vi Editor 종료                |
| :wq                                                   | 저장 후 vi Editor 종료           |
| :x                                                    | 저장 후 vi Editor 종료           |

### (4) 기타 명령어

| 명령어                                | 의미                                          |
|------------------------------------|---------------------------------------------|
| :0                                 | 파일 내용의 가장 앞으로 이동                            |
| :\$                                | 파일 내용의 가장 뒤로 이동                             |
| :e <b>[file name]</b>              | 다른 파일을 불러옴                                  |
| :e!                                | 편집한 내용을 저장하지 않고 최종 저장된 상태로 파일 열기            |
| : <b>[시작 number],[종료 number]</b> d | 지정한 라인 사이를 삭제                               |
| :sh                                | Shell 상태로 전환, 'exit'를 입력하면 다시 vi Editor로 전환 |
| :! <b>[command]</b>                | Shell 상태로 전환하여 명령 실행                        |
| :ab <b>[약어]</b> <b>[문자열]</b>       | 약어를 입력하면 지정한 문자열로 자동 대체                     |

| 명령어              | 의미                                          |
|------------------|---------------------------------------------|
| :una <b>[약어]</b> | 약어 지정 취소                                    |
| :ab              | 현재 설정되어 있는 약어 확인                            |
| :map q :wq!      | Command Mode에서 'q'를 입력하면 ':wq!'가 실행, 매크로 설정 |
| :unmap q         | 매크로 해제                                      |
| :map             | 현재 설정되어 있는 매크로 확인                           |

## IV. 그 밖의 기능

- vi Editor를 사용할 때 마다 환경 설정, 약어 및 매크로를 지정하는 것은 번거로움
- 다양한 설정 내용을 .exrc 파일로 만들어 저장하게 되면 자동으로 적용이 가능
- .exrc 파일의 위치
  - "/root/.exrc" : root 계정의 .exrc 파일 위치 (참고: .vimrc도 같은 역할)
  - "/home/[User Name]/.exrc" : 일반 계정의 .exrc 파일 위치
- .exrc 파일 생성 예
  - 조건> Line Number 출력, [Tab] 키의 간격 = 4, 'q' 입력 시 ':wq!'으로 대체(매크로)
  - [root@rocky ~]# pwd  
/root
  - [root@rocky ~]# vi .exrc

[그림 4.3] .exrc 파일 생성



## - vi 사용

### 1. 명령모드

- h,j,k,l : 커서 이동
- i,l,a,A,o,O : 입력 전환
- x : 한글자 삭제
- dd : 한 라인 삭제 => 숫자로 시작하면 해당 숫자의 행만큼(5dd)
- yy : 한 라인 복사 => " (5yy)
- p : 현재 커서 아래로 붙여넣기
- P : 현재 커서 위로 붙여넣기
- J : 현재 커서 라인과 아래 라인 합치기 => 숫자로 시작하면 해당 숫자의 행만큼 합치기
- /패턴 : 아래로 찾기
- ?패턴 : 위로 찾기
- cw : 단어 삭제 후 입력 모드
- dw : 단어 삭제
- 행번호G : 해당 행으로 이동 => :행번호와 동일
- G : 문서의 마지막행으로 이동
- gg : 문서의 첫행으로 이동
- ^ 또는 0 : 라인의 처음으로 이동
- \$ : 라인의 마지막으로 이동
- r : 한글자 치환
- L : 화면의 마지막행으로 이동
- u : 직전 입력내용 취소
- U : 해당 라인의 입력내용 취소

---

ZZ : 저장 후 종료

## 2. 입력모드

명령모드에서 i,l,a,A,o,O 입력시 입력모드로 전환

## 3. 라스트 라인 모드

명령모드에서 :, / ,? 입력시 전환

/ , ? : 패턴 검색

: => 저장, 종료, 치환, 환경설정

:%s/end/END/gc => 문서 전체에서 end 단어를 END로 치환. g 옵션은 행의 모든  
end 치환. %는 1,\$ 와 같은 의미. c는 바꾸기전에 확인

:w => 저장

:wq => 저장 후 종료

:e 파일명 => 해당 파일 열기

:q! => !는 force 의미 강제로 종료.

:set number => 행번호 보이기 :set nonumber

:set ts=4 => 탭키 입력시 4칸씩 이동

:ab s set => s입력후 스페이스바 누르면 set 이 입력된다. => :unab s

환경설정은 vi를 종료하면 해제됨. 다시 vi 실행해도 적용이 안됨.

vi를 실행할때마다 적용하려면 사용자 홈디렉토리에 .vimrc 또는 .exrc 파일에  
환경설정내용을 저장한다.

ex) /root/.vimrc , /home/admin/.exrc

:!ls => 편집기를 종료하지 않고 쉘명령 수행

:sh => 편집기를 종료하지 않고 쉘로 이동. 작업 후 exit 입력하면 편집기로 복귀.

### 7.1 Shell

; 셸은 시스템과 사용자 사이에서 중계역할을 담당하는 인터페이스로써, 사용자의 명령을 해석하여 커널에 전달하고, 결과값을 사용자에게 알려주는 역할을 한다.

#### I. bash(bourne again shell)

- 리눅스의 기본 셸
- 사용자의 셸 프롬프트 확인
  - root 사용자의 셸 프롬프트 : #
  - 일반 사용자의 셸 프롬프트 : \$

#### II. 기본 환경 설정파일

- .bash\_profile : 이 파일에 셸의 환경 변수를 설정하면 로그인시에 자동으로 실행되어 사용자의 셸 환경을 설정.
  - bash\_profile은 로그인시에 참조되고, 새로운 셸을 생성하면 참조되지 않는다.
  - 따라서, 환경변수값에 사용자가 원하는 값을 넣고 전역화시키면 새로운 셸에도 적용됨.
- .bashrc : 명령어에 의해 사용자가 원하는 셸 환경을 만들어주고자 할 때, 이 파일에 입력한다.
  - 예) alias 명령 입력
- 자주 사용되는 환경변수(반드시 대문자)
  - PATH : 실행파일의 검색 경로
  - PWD : 현재 위치(절대경로)
  - SHELL : 셸의 경로
  - HOME : 사용자의 홈 경로
  - HOSTNAME : 호스트 명
  - LOGNAME : 로그인 명
  - ENV : 환경 지정파일의 위치

현재 환경변수 값 보기 : # echo \$환경변수

전체 환경변수 값 보기 : # env

#### III. 프롬프트 꾸미기

- 리눅스의 기본 bash 프롬프트 상태 :
  - [사용자@호스트명 ~]#
- 프롬프트 상태는 PS1 셸변수로 제어된다. 따라서, 프롬프트 상태를 바꾸려면 PS1의 값을 변경하면 된다.



---

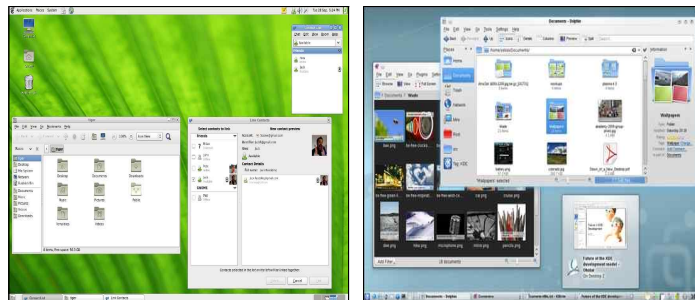
## 7.2 사용자 로그인 & 로그아웃

### 1. 사용자 로그인(login)

- System의 자원을 사용하기 위해 ID/PASSWORD를 통해 시스템으로 접근
  - Linux는 하나의 시스템에 사용자별 동시 접근(Multi Users-다중 사용자) 및 동시 작업(Multi Tasking-다중 작업) 환경 제공
- GUI/TUI를 통한 로그인으로 구분

#### (1) GUI(Graphic User Interface) 로그인

[그림 5.1] GNOME, KDE 화면



- KDE(K Desktop Environment) 세션
  - 자유 소프트웨어 데스크톱 환경으로 노키아의 Qt 툴킷 기반
- GNOME(그놈) 세션
  - 사용자를 위한 자유롭고 사용하기 쉽도록 구현된 데스크탑 환경

## 7.3 시스템 기본 관리 명령

### 1. 시스템 종료/재부팅

#### (1) 정상적인 시스템 종료/재부팅

- 전원, 리셋 버튼을 이용한 종료 금지
  - 시스템상의 작업 수행은 Disk(Data) → Memory(Data 로딩/수정) → Disk(Data 저장)
  - 전원, 리셋 버튼을 이용할 경우 Memory의 Data 손실 초래
  - Linux는 다중 사용자 환경을 지원하기 때문에 작업 중인 사용자에게 종료 시간 부여
- 단계별 종료 프로세스 수행을 통한 정상 종료 진행
- GNOME 환경에서의 종료
  - 상단 패널의 '시스템' 클릭 후 '끄기' 선택
- 커맨드 라인에서의 시스템 종료 방법
  - [root@rocky ~]# shutdown -h now
  - [root@rocky ~]# halt
  - [root@rocky ~]# poweroff
  - [root@rocky ~]# init 0

---

## (2) shutdown

- 명령어 위치 : /sbin/shutdown
- 명령어 사용 방법 : # shutdown <option> <time> <message>
- option
  - -k : 실제로는 종료하지 않고 모든 사용자에게 경고 메시지를 보냄
  - -r : 재시작(Rebooting)
  - -h : 종료(halt)
  - -f : fsck(File System Check)를 실행하지 않고 Fast Reboot
  - -c : shutdown을 취소(Cancel), 혹은 "Ctrl + C" 키를 이용한 shutdown 취소 가능
- time
  - shutdown 명령어 실행 시간 지정 가능
  - "hh:mm" : 절대적 시간 지정, 지정된 시간에 shutdown 명령어 실행
  - "n" : 상대적 시간 지정, n분 후에 shutdown 명령어 실행(단위 : 분(Minute))
- 사용 예
  - [root@rocky ~]# shutdown -h now → 시스템 즉시 종료
  - [root@rocky ~]# shutdown -r now → 시스템 즉시 재부팅
  - [root@rocky ~]# shutdown +3 "System Shutdwon" → 3분 후 시스템 종료 및 메시지 전달
  - [root@rocky ~]# shutdown -h 11:00 → 11:00에 시스템 종료

## (3) init

- 명령어 위치 : /sbin/init
- 명령어 사용 방법 : # init <option> <Run Level>
- option
  - -t(sec) : 사용자가 지정한 시간 뒤에 명령어 실행(단위 : 초(Second))
- Run Level(0~6)
  - Run Level 0 : 시스템 종료(halt)
  - Run Level 6 : 시스템 재부팅(reboot)
- 사용 예
  - [root@rocky ~]# init 0 → 시스템 종료
  - [root@rocky ~]# init 6 → 시스템 재부팅
  - [root@rocky ~]# init -t 60 6 → 60초 뒤에 시스템 재부팅

## (4) halt

- shutdown 명령 호출
- 명령어 위치 : /sbin/halt
- 옵션
  - -n : reboot나 종료 전에 sync하지 않음.
  - -w : 실질적으로 reboot나 종료는 하지 않지만, wtmp 기록은 남김.
  - -d : wtmp파일에 기록을 남기지 않음. -n 옵션은 -d 옵션 동반
- 명령어 사용 방법 : # halt
- 사용 예
  - [root@rocky ~]# halt → 시스템 종료

---

(5) poweroff

- 명령어 위치 : /sbin/poweroff
- halt의 심볼릭 링크 파일
- 명령어 사용 방법 : # poweroff
- 사용 예
  - [root@rocky ~]# poweroff → 시스템 종료

(6) reboot

- 명령어 위치 : /sbin/reboot
- halt의 심볼릭 링크 파일
- 명령어 사용 방법 : # reboot
- 사용 예
  - [root@rocky ~]# reboot → 시스템 재부팅

(7) halt, poweroff, reboot

- 공통 option
  - -n : /var/log/wtmp 파일에 기록을 남기지 않고, sync 하지 않은 상태로 종료(재부팅)
  - -w : 실제로 종료는 하지 않고, /var/log/wtmp 파일에 기록만 남김
  - -d : /var/log/wtmp 파일에 기록을 남기지 않음
  - -f : 강제로 종료(재부팅)

## II. Linux System 명령어

(1) which

- 환경 변수 PATH에 설정되어 있는 경로 내에서 명령어의 경로를 확인하는 명령어
- 명령어 위치 : /usr/bin/which
- 명령어 사용 방법 : # which
- 사용 예
  - [root@rocky ~]# which useradd
  - /usr/sbin/useradd

(2) whereis

- 명령의 실행 파일, 소스, 매뉴얼 페이지가 어디 있는지 보여주는 명령어
- 명령어 위치 : /usr/bin/whereis
- 명령어 사용 방법 : # whereis <option> 명령어
- option
  - -b 실행 파일만 출력
  - -m 매뉴얼 페이지만 출력
  - -s 소스만 출력
- 사용 예
  - [root@rocky ~]# whereis useradd
  - useradd: /usr/sbin/useradd /usr/share/man/man8/useradd.8.gz

(3) date

- Linux System의 현재 시간과 날짜를 확인하거나 설정하는 명령어
- 명령어 위치 : /bin/date
- 명령어 사용 방법 : # date, # date [MMDDhhmmYY]
- 사용 예

- [root@rocky ~]# date  
2023. 01. 05. (화) 15:26:52 KST
- [root@rocky ~]# date 0115150023  
2023. 01. 15. (금) 15:00:01 KST  
→ 시스템의 날짜와 시간 설정
- [root@rocky ~]# date --date '3 days ago'  
2010. 01. 02. (토) 15:28:43 KST  
→ 3일전의 날짜 정보 출력
- [root@rocky ~]# date -d '2 months 5 day'  
2010. 03. 10. (일) 15:31:01 KST  
→ "2개월 2일"이 지난 날짜 출력
- [root@rocky ~]# date +%d  
→ 오늘 날짜가 이번 달의 며칠인지 출력

| 시간필드 | 출력 내용                               | 시간필드 | 출력 내용                                            |
|------|-------------------------------------|------|--------------------------------------------------|
| %H   | (시간) 00, 01..., 23                  | %I   | (시간) 01, 02..., 12                               |
| %k   | (시간) 0, 1..., 23                    | %l   | (시간) 1, 2..., 12                                 |
| %M   | (분) 00, 01..., 59                   | %p   | AM 또는 PM                                         |
| %r   | 시간, 12-시간                           | %s   | 1970년 1월 1일 00:00:00 UTC(비표준 확장 기능)로부터 경과된 시간(초) |
| %T   | 시간, 24-시간                           | %X   | 로케일에서 정의한 시간 표현                                  |
| %Z   | 시간대, 시간대를 결정할 수 없을 때는 아무 값도 출력하지 않음 |      |                                                  |

| 날짜필드 | 출력 내용                                                   | 날짜필드 | 출력 내용                     |
|------|---------------------------------------------------------|------|---------------------------|
| %a   | 로케일의 약식 요일 이름(Sun, Sta...)                              | %A   | 로케일의 완전한 요일 이름            |
| %b   | 로케일의 약식 월 이름(Jan, Dec...)                               | %B   | 로케일의 완전한 월 이름             |
| %c   | 로케일의 날짜와 시간                                             | %d   | 월 중 일(01, 02,... 31)      |
| %D   | 날짜(mm/dd/yy)                                            | %h   | %b와 동일                    |
| %j   | 연 중 일 (001, 366)                                        | %m   | 월(01,02..., 12)           |
| %U   | 연 중 주 번호, 일요일을 주의 첫 번째 날로 계산(00, 01..., 53)             | %w   | 요일 번호(0, 1..., 6), 0은 일요일 |
| %W   | 연 중 주 번호, 월요일을 주의 첫 번째 날로 계산(00, 01..., 53)             | %x   | 로케일의 날짜 표현식(mm/dd/yy)     |
| %y   | 연 중 일의 마지막 두 숫자(00,01, ..., 99), %Y 연 (1970, 1971, ...) |      |                           |

#### (4) cal

- 간단한 달력을 출력해주는 명령어
- 명령어 위치 : /usr/bin/cal
- 명령어 사용 방법 : # cat <option> 년도
- option
  - -y : 해당 년도의 모든 달력을 보여줌
- 사용 예
  - [root@rocky ~]# cal

- 
- 현재 월의 달력을 출력
  - [root@rocky ~]# cal -y 2023
  - 2023년도의 달력을 모두 출력

(5) hwclock

- 하드웨어(ROM-BIOS) 시간을 확인하거나 리눅스와의 시간을 동기화 시킬 때 사용하는 명령어
- 명령어 위치 : /sbin/hwclock
- 명령어 사용 방법 : # hwclock <option>
- option
  - -r : 하드웨어 시간 확인
  - -w : 하드웨어 시간을 시스템 시간과 동기화
  - -s : 시스템 시간을 하드웨어 시간과 동기화
- 사용 예
  - [root@rocky ~]# hwclock -r
  - 하드웨어 시간 확인
  - [root@rocky ~]# hwclock -s
  - 시스템 시간을 하드웨어 시간과 동기화

(6) 서버의 시간을 시간서버(NTP)와 동기화 하기

- [root@rocky ~]# systemctl restart chronyd

(7) id

- 사용자 정보인 UID와 GID, 사용자가 속한 그룹을 출력하는 명령어
- 명령어 위치 : /usr/bin/id
- 명령어 사용 방법 : # id <user name>
- 사용 예
  - [root@rocky ~]# id
  - uid=0(root) gid=0(root)
  - groups=0(root),1(bin),2(daemon),3(sys),4(adm),6(disk),10(wheel)
  - 현재 로그인된 사용자의 정보 출력
  - [root@rocky ~]# id student
  - uid=500(student) gid=500(student) groups=500(student)
  - 특정 사용자(student 계정)의 정보 출력

(8) logname

- 시스템에 로그인한 사용자 이름을 출력하는 명령어
- 명령어 위치 : /usr/bin/logname
- 명령어 사용 방법 : # logname
- 사용 예
  - [root@rocky ~]# logname
  - root

(9) w

- 시스템에 접속한 사용자의 접속 정보와 작업에 대한 정보를 확인하는 명령어
- 명령어 위치 : /usr/bin/w

- 명령어 사용 방법 : # w <option> <user name>
- option
  - -h : head 라인을 출력하지 않음
  - -s : login time, JCPU, PCPU 정보를 생략하고 간단히 출력
  - -f : 접속 IP 주소 정보를 생략하고 출력(FROM 필드 생략)
- 사용 예

[그림 7.1] w 명령어 실행 화면

```

root@localhost:~# w
10:31:51 up 2:16, 3 users, load average: 0.00, 0.01, 0.00
USER TTY FROM LOGIN@ IDLE JCPU PCPU WHAT
root :0 Thu17 ?xdm? 30.19s 0.31s /usr/bin/gnome-
root pts/1 :0.0 10:27 0.00s 0.10s 0.01s w
centos pts/4 192.168.0.38 Thu19 15:13m 0.05s 0.05s -bash
[root@localhost ~]#

```

#### (10) who

- 현재 시스템에 로그인한 모든 사용자를 보여주는 명령어. /var/run/utmp의 내용을 보여줌.
- 명령어 위치 : /usr/bin/who
- 명령어 사용 방법 : # who <option>
- option
  - -m : 'who am i' 명령어와 같은 결과 출력, 현재 사용중인 사용자의 정보만 출력
  - -b : 시스템이 마지막으로 부팅된 시간 출력
  - -q : 로그인명과 로그인한 횟수 출력
  - -i, -u, --idle : 로그인한 사용자가 얼마의 시간동안 한가하게 있었는지 출력
    - '.'는 조금 전까지 활동을 의미하며 'old'는 24시간을 의미
  - -w : 사용자 이름 옆에 사용자의 메시지 상황을 문자로 출력
    - '+'는 write 메시지 허가, '-'는 write 메시지 불허, '?'는 터미널 장치를 찾을 수 없음
  - -H : 헤더 라인 출력
- 사용 예

[그림 7.2] who 명령어 실행 화면

```

root@localhost:~# who
root :0 2011-01-27 17:10
root pts/1 2011-01-28 10:27 (:0.0)
centos pts/4 2011-01-27 19:10 (192.168.0.38)
[root@localhost ~]# who -u
root :0 2011-01-27 17:10
root pts/1 2011-01-28 10:27
centos pts/4 2011-01-27 19:10 15:12
[root@localhost ~]# who -w
root ? :0 2011-01-27 17:10
root + pts/1 2011-01-28 10:27 (:0.0)
centos + pts/4 2011-01-27 19:10 (192.168.0.38)
[root@localhost ~]#

```

#### (11) last

- 사용자들의 마지막 로그인 기록을 보여주는 명령어.
- /var/log/wtmp파일이거나 -f로 지정한 파일을 통해서 사용자의 로그인 로그아웃시간을 보여줌.
- 명령어 위치 : /usr/bin/last
- 명령어 사용 방법 : # who <option> <사용자명>
- option
  - -num : num만큼의 줄만 보여준다.

- -R : hostname 필드를 보여주지 않는다.
- -a : hostname 필드를 마지막에 보여줌.

```

root@localhost ~]# last
root pts/3 :0.0 Fri Jan 28 11:15 still logged in
root pts/2 192.168.0.38 Fri Jan 28 10:34 still logged in
root pts/1 :0.0 Fri Jan 28 10:27 still logged in
centos pts/4 192.168.0.38 Thu Jan 27 19:10 still logged in
root pts/3 192.168.0.38 Thu Jan 27 18:13 - 10:28 (16:15)
root pts/2 192.168.0.38 Thu Jan 27 17:21 - 10:29 (17:08)
root pts/2 192.168.0.38 Thu Jan 27 17:19 - 17:20 (00:00)
root pts/2 192.168.0.38 Thu Jan 27 17:18 - 17:19 (00:00)
root pts/2 192.168.0.38 Thu Jan 27 17:18 - 17:18 (00:00)
root pts/2 192.168.0.38 Thu Jan 27 17:17 - 17:17 (00:00)
root pts/1 :0.0 Thu Jan 27 17:11 - 19:17 (02:06)
root :0 Thu Jan 27 17:10 still logged in
root :0 Thu Jan 27 17:10 - 17:10 (00:00)
reboot system boot 2,6,18-128.el5 Thu Jan 27 17:09 (18:06)
reboot system boot 2,6,18-128.el5 Thu Jan 27 17:04 (00:03)

wtmp begins Thu Jan 27 17:04:48 2011
root@localhost ~]#

```

- -d : 다른 호스트에서 접속한 것만 보여줌.
- -x : shutdown이 일어난 상태나, run level이 바뀐 상태도 보여줌.

- 사용 예

**[그림 7.3] last 명령어 실행 화면**

(12) whoami => 'who am i'와 비교

- 가상 콘솔을 사용하거나 여러 계정으로 동시에 로그인하고 있는 경우, 현재 시스템을 사용하고 사용자 계정이 누구인지 확인할 때 사용하는 명령어
- 명령어 위치 : /usr/bin/whoami
- 명령어 사용 방법 : # whoami
- 사용 예
  - [root@rocky ~]# whoami
  - root

(13) users

- 현재 접속한 사용자들의 사용자 ID 정보를 확인할 때 사용
- 명령어 위치 : /usr/bin/users
- 명령어 사용 방법 : # users
- 사용 예
  - [root@rocky ~]# users
  - root root rocky

(15) msg

- 메시지를 받을 것인지, 받지 않을 것인지를 설정하고 또한 현재의 상태를 확인하는데 사용하는 명령어
- 명령어 위치 : /usr/bin/msg
- 명령어 사용 방법 : # msg <option>

- option
  - n : 메시지 수신 불가
  - y : 메시지 수신 가능
- 사용 예
  - [root@rocky ~]# mesg  
is y  
→ 현재 메시지 수신 상태 확인
  - [root@rocky ~]# mesg n  
→ 메시지 수신이 불가능 하도록 설정

#### (16) write

- 상대방에게 메시지를 보낼 수 있는 명령어
- 명령어 위치 : /usr/bin/write
- 명령어 사용 방법 : # write <user name> <terminal>
- 사용 예
  - [root@rocky ~]# write student pts/2  
Hello! Mr. Student!  
<Ctrl + D>

#### (17) wall

- 시스템에 접속해 있는 모든 사용자에게 동시에 메시지를 보낼 때 사용하는 명령어
- 보통 시스템 관리자가 점검을 위해 접속한 사용자들에게 메시지를 보낼 때 주로 사용
- 명령어 위치 : /usr/bin/wall
- 명령어 사용 방법 : # wall <메시지>
- 사용 예
  - [root@rocky ~]# wall Message Test

#### (18) history

- 사용자의 홈 디렉토리에 존재하는 .bash\_history 파일에는 사용자가 사용했던 명령어가 기록되어 있으며, 이를 확인하는 명령어.
- 명령어 위치 : Bash BuiltIn
- 명령어 사용 방법 : # history <number> <option>
- option
  - -w <file name> : 지정한 파일에 history 명령어 실행 결과 저장
  - -a <file name> : 지정한 파일에 새로운 history 행 추가
  - -r <file name> : 지정한 history 파일 내용을 읽어 현재의 history로 사용
  - -c : 저장된 history 내용을 모두 삭제
- 이벤트 지시자

| 지시자  | 의미                           |
|------|------------------------------|
| !!   | 이전 명령어 실행                    |
| !n   | history 명령어 수행 결과의 n번 명령어 실행 |
| !-n  | 현재 명령 행에서 n을 뺀 행을 실행         |
| !문자열 | 지정 문자열로 시작하는 가장 최근 명령어 실행    |



- 
- 사용 예
    - [root@rocky ~]# history
    - [root@rocky ~]# !!
    - [root@rocky ~]# !232
    - [root@rocky ~]# !-3
    - [root@rocky ~]# !p

#### (19) hostname

- 현재 시스템의 HOSTNAME을 확인할 때 사용하는 명령어
- 관련 파일 : /etc/hosts
- 명령어 위치 : /bin/hostname
- 명령어 사용 방법 : # hostname <hostname>
- 사용 예
  - [root@rocky ~]# hostname  
rocky
  - [root@rocky ~]# hostname linux  
[root@rocky ~]# hostname  
linux

#### (20) man

- 명령어에 대한 정보를 확인하는 명령어
- 명령어의 이름, 사용 방법, 설명, 각종 옵션, 사용 환경, 버그, 팁, 제작자 등의 확인 가능
- 자세한 명령어 사용 방법은 man 명령어를 이용
- 명령어 위치 : /usr/bin/man
- 명령어 사용 방법 : # man <option> <섹션> <command>
- option
  - -a : man은 기본적으로 첫 번째 발견된 매뉴얼 페이지만 출력하지만, -a 옵션을 이용하여 모든 매뉴얼 페이지의 출력이 가능
  - -f : whatis와 같이 간단한 설명만 출력
- man 명령어 출력 형식
  - NAME : 설명할 명령어
  - SYNOPSIS : 명령어의 사용 방법
  - DESCRIPTION : 명령어가 어디에, 어떻게 쓰이는지 설명
  - OPTIONS : 명령어에서 사용되는 옵션
  - EXAMPLES : 명령어의 사용 예
  - ENVIRONMENT : 명령어가 사용되는 환경 설명
  - SEE ALSO : 대상 명령어와 비슷한 용도로 사용하는 명령어들을 표시
  - FILES : 명령어의 사용과 관련된 파일들 표시
  - BUGS : 명령어의 버그 정보
  - AUTHOR : 명령어 제작자 정보
  - COPYRIGHT : 명령어 소유권 정보
- 사용 예
  - [root@rocky ~]# man useradd

---

#### IV. 파일 및 디렉토리 관리 명령어

##### (1) pwd(Print Working Directory)

- 사용자의 현재 디렉토리의 위치를 절대 경로로 출력해주는 명령어
- 명령어 위치 : /bin/pwd
- 명령어 사용 방법 : # pwd
- 사용 예
  - [root@rocky ~]# pwd
  - /root

##### (2) cd(Change Directory)

- 현재 디렉토리에서 다른 디렉토리로 이동할 경우 사용하는 명령어
- 명령어 위치 : Bash BuiltIn
- 명령어 사용 방법 : # cd <PATH>
- 사용 예
  - [root@rocky ~]# cd /etc
  - 절대 경로명을 이용한 디렉토리 변경
  - [root@rocky etc]# pwd
  - /etc
  - [root@rocky etc]# cd sysconfig
  - 상대 경로명을 이용한 디렉토리 변경
  - [root@rocky sysconfig]# pwd
  - /etc/sysconfig

##### 「cd 명령어 사용의 응용」

- # cd . : 현재 디렉토리로 이동
- # cd .. : 현재 디렉토리의 상위 디렉토리로 이동
- # cd - : 현재 위치한 디렉토리로 오기 이전에 위치했었던 디렉토리로 이동
- # cd ~ : 현재 사용중인 계정의 홈 디렉토리로 이동
  - (# cd ~test01 : test01 사용자의 홈 디렉토리)
  - (# cd ~/Desktop : )

##### (3) mkdir(Make Directory)

- 디렉토리를 생성할 때 사용하는 명령어
- 현재 디렉토리에 쓰기 권한이 없거나 동일한 이름의 디렉토리가 존재할 경우에는 디렉토리 생성 불가
- 명령어 위치 : /bin/mkdir
- 명령어 사용 방법 : # mkdir <option> <directory name>
- option
  - -m : 생성할 디렉토리의 퍼미션을 별도로 지정
  - -p : 상위 디렉토리까지 함께 생성
- 사용 예
  - [root@rocky /]# mkdir test001
  - [root@rocky /]# ls -l
  - drwx----- 2 root root 4096 1월 10 13:20 test001
  - [root@rocky /]# cd test001

- 
- [root@rocky test001]# pwd  
/test001

(4) rmdir(Remove Directory)

- 아무 내용이 없는 비어있는 디렉토리를 삭제할 경우 사용하는 명령어
- 명령어 위치 : /bin/rmdir
- 명령어 사용 방법 : # rmdir <option> <directory name>
- option
  - -p : 상위 디렉토리까지 함께 삭제할 경우 사용
- 사용 예
  - [root@rocky /]# rmdir test001
  - [root@rocky /]# ls -l

(5) ls(List)

- 특정 디렉토리에 포함된 파일 목록을 확인할 경우 사용하는 명령어
- 명령어 위치 : /bin/ls
- 명령어 사용 방법 : # ls <option> <directory name | file name>
- option
  - -a : 숨김 파일까지 출력
  - -d : 입력한 경로에 해당하는 디렉토리 자체의 정보를 출력
  - -l : 파일, 디렉토리의 자세한 정보를 출력
  - -t : 파일을 최근 파일부터 시간순으로 출력
  - -F : 파일, 디렉토리의 이름과 함께 파일 타입까지 함께 출력
  - -k : l 옵션과 함께 사용하여 파일의 크기를 Kilo-Byte 단위로 출력
  - -n : -l과 유사하지만 파일의 소유자, 소유 그룹을 UID, GID 번호로 출력
  - -R : 지정한 디렉토리의 하위 디렉토리가 포함하는 파일까지 출력
- 사용 예
  - [root@rocky ~]# ls
  - [root@rocky ~]# ls -l

(6) cat(concatenate)

- 텍스트 파일의 내용을 표준 출력으로 표시하거나 간단한 텍스트 파일을 작성할 때 사용하는 명령어로서 편집 기능은 제공하지 않음
- 명령어 위치 : /bin/cat
- 명령어 사용 방법 : # cat <option> <file name>  
# cat > <file name>
- option
  - -n : 파일 내용의 행마다 라인 번호를 출력
- 사용 예
  - [root@rocky /]# cat /etc/sysconfig/network
  - [root@rocky /]# cat > file1  
I Love U  
<Ctrl + D>
  - [root@rocky /]# cat file1  
I Love U

---

(7) more

- 텍스트 파일의 내용을 표준 출력으로 표시하지만 파일 내용이 많을 경우에 결과를 나눠서 출력해주는 명령어
- 명령어 위치 : /bin/more
- 명령어 사용 방법 : # more <option> <file name>
- option
  - +n : 위에서 n번째 라인부터 마지막까지 출력
  - -n : 처음부터 n개의 라인씩 나눠서 출력
- 사용 예
  - [root@rocky /]# more /etc/bashrc
  - [root@rocky /]# more +8 /etc/bashrc
  - [root@rocky /]# ls -l | more
- 내부명령
  - :f : 현재 파일 이름과 라인을 출력
  - = : 현재 라인 번호 출력

(8) less

- more 명령어와 마찬가지로 파일 내용이 많을 경우 나눠서 출력해주는 명령어
- 명령어 위치 : /usr/bin/less
- 명령어 사용 방법 : # less <option> <file name>
- option
  - +n : 위에서 n번째 라인부터 마지막까지 출력
  - -n : 처음 한 페이지 출력 후 다음부터 n개의 라인씩 나눠서 출력
- 사용 예
  - [root@rocky /]# less /etc/profile
  - [root@rocky /]# more -2 /etc/bashrc

「more & less 그리고 vi」

- more와 less 명령어 모두, 실행 중 “v” 키를 입력하면 vi 편집기로 전환됨

(9) head

- 텍스트 파일의 내용을 지정한 행의 수 만큼 출력
- 명령어 위치 : /usr/bin/head
- 명령어 사용 방법 : # head <option> <file name>
- option
  - -n : 처음부터 n개의 라인 출력

(10) tail

- 텍스트 파일의 내용의 뒷부분부터 지정한 행의 수만큼 출력
- 명령어 위치 : /usr/bin/tail
- 명령어 사용 방법 : # tail <option> <file name>
- option
  - -n : 파일의 마지막부터 n개의 라인만 출력

- 
- -f : 파일을 열어 놓은 채 추가되는 내용을 지속적으로 출력 => 모니터링에 사용됨.

(11) touch

- Size가 0인 파일을 생성하거나 기존 파일의 시간을 현재 시간, 혹은 특정 시간으로 변경해주는 명령어
- 명령어 위치 : /bin/touch
- 명령어 사용 방법 : # touch <option> <file name>
- option
  - -a : 파일의 Access Time 변경
  - -c : 파일이 존재하지 않을 시에는 파일을 생성하지 않음
  - -d <time> : 현재 시간이 아닌 지정한 시간으로 변경
  - -m : 파일의 Modify Time만 변경
  - -r <file name> : 현재 시간이 아닌 지정한 파일의 시간으로 변경
  - -t <time> : 현재 시간이 아닌 지정한 시간[MMDDhhmm[[CC]YY][.SS]으로 변경
- 사용 예
  - [root@rocky /]# touch touchtest → size가 0인 파일 생성
  - [root@rocky /]# ls -l touchtest  
-rw-r--r-- 1 root root 0 1월 10 14:10 touchtest
  - [root@rocky /]# touch touchtest → 시간을 현재 시간으로 변경
  - [root@rocky /]# ls -l touchtest  
-rw-r--r-- 1 root root 0 1월 10 14:12 touchtest
  - [root@rocky /]# touch -t 200712311200 touchtest → 시간을 특정 시간으로 변경
  - [root@rocky /]# ls -l touchtest  
-rw-r--r-- 1 root root 0 12월 31 2007 touchtest
  - [root@rocky /]# touch -d 2005/12/30 touchtest → 시간을 특정 시간으로 변경
  - [root@rocky /]# ls -l touchtest  
-rw-r--r-- 1 root root 0 12월 30 2005 touchtest

(12) cp(CoPy)

- 파일을 복사할 경우 사용하는 명령어
- 명령어 위치 : /bin/cp
- 명령어 사용 방법 : # cp <option> <source file> <destination file>
- option
  - -b : 같은 이름의 대상 파일이 이미 존재하는 경우 백업 파일을 생성한 뒤 복사
  - -f : 같은 이름의 대상 파일이 이미 존재하는 경우 강제로 복사
  - -p : 소유자, 소유 그룹, 권한, 시간 등의 속성을 그대로 유지하며 복사
  - -v : 복사하는 상황을 표시
  - -i : 같은 이름의 대상 파일이 이미 존재하는 경우 사용자에게 물어봄
  - -R(r) : 원본 파일이 디렉토리일 경우 하위 디렉토리 내용까지 복사
- 사용 예
  - [root@rocky /]# cp .bash\_history cptest → 동일한 디렉토리 내에서 파일 복사
  - [root@rocky /]# cp cptest /cpdata → 다른 디렉토리로 파일 복사
  - [root@rocky /]# cp -r /etc /cpdata → 특정 디렉토리의 하위 내용 모두를 복사
  - [root@rocky /]# cp a\* /cpdata → 특정 파일만 복사
  - [root@rocky /]# cp -p /etc/aliases.db /cpdata → 파일 속성을 유지하며 복사

---

(13) mv(MoVe)

- 파일을 다른 디렉토리로 이동시키거나 이름을 변경하고자 할 때 사용하는 명령어
- 명령어 위치 : /bin/mv
- 명령어 사용 방법 : # mv <option> <source file> <destination file>
- option
  - -f : 같은 이름의 대상 파일이 이미 존재하는 경우 강제로 덮어쓰기
  - -i : 같은 이름의 대상 파일이 이미 존재하는 경우 덮어쓸지의 여부를 물어봄
- 사용 예
  - [root@rocky cpdata]# mv cpdata mvdata → 파일 이름 변경
  - [root@rocky cpdata]# mv mvdata /home → 다른 디렉토리로 파일 이동

(14) rm(ReMove)

- 파일, 디렉토리를 삭제할 경우 사용하는 명령어
- 명령어 위치 : /bin/rm
- 명령어 사용 방법 : # rm <option> <directory name | file name>
- option
  - -f : 파일을 삭제할 경우 삭제 여부를 묻지 않고 바로 삭제
  - -i : 파일을 삭제할 경우 삭제 여부를 물어봄
  - -r : 디렉토리를 삭제할 경우
- 사용 예
  - [root@rocky /]# rm /home/mvdata → 파일 삭제
  - [root@rocky /]# rm -r /cpdata → 디렉토리 삭제

(15) lsattr

- 파일, 디렉토리의 속성을 확인하는 명령어
- 명령어 위치 : /usr/bin/lsattr
- 명령어 사용 방법 : # lsattr <option> <directory name | file name>
- option
  - -a : '.'을 포함한 모든 파일을 출력
  - -d : 디렉토리일 경우 디렉토리 내의 파일 대신 일반파일과 같이 취급

(16) chattr

- 파일, 디렉토리의 속성을 변경하는 명령어
- 명령어 위치 : /usr/bin/chattr
- 명령어 사용 방법 : # chattr <option> [+|=mode] <directory name | file name>
- option
  - -R : 대상이 디렉토리이면 그 하위 경로에 있는 파일과 디렉토리까지 모드 적용
- mode
  - i : Read-Only 속성.
  - a : 해당 파일을 추가만 할 수 있음.

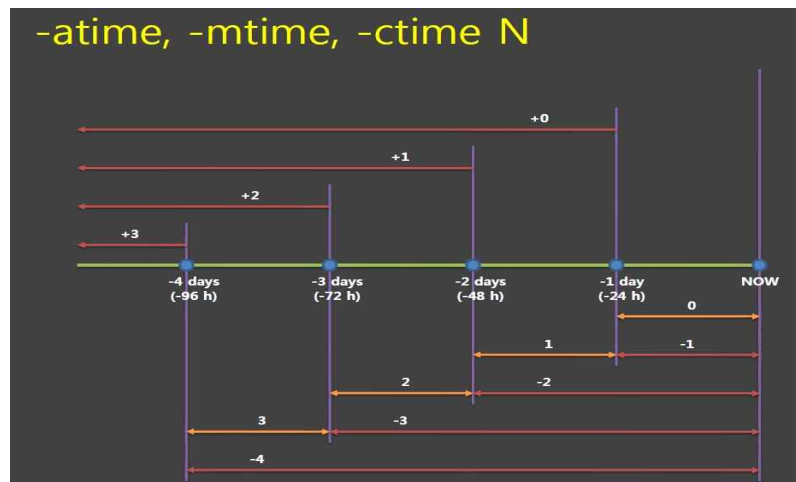
V. 검색 관련 명령어

(1) find

- 파일을 검색할 때 사용하는 명령어
- 명령어 위치 : /usr/bin/find
- 명령어 사용 방법 : # find [경로] [옵션] 파일명
- 옵션
  - -name <pattern> : pattern으로 주어진 파일
  - -perm : 지정한 퍼미션을 갖는 파일
  - -type : 검색하고자 하는 파일 종류 지정
  - -maxdepth : 검색하고자 하는 디렉토리의 깊이 지정
  - -mtime +/-n : n\*24시간 기준으로(이후/이전) 내용이 변경된 파일 검색
  - -atime +/-n : n\*24시간 기준으로(이후/이전) 액세스된 파일 검색
  - -ctime +/-n : n\*24시간 기준으로(이후/이전) 속성이 변경된 파일 검색

참고) 파일의 mac time

m time : 파일의 생성, 수정시간을 나타냄.      ls -lt  
a time : 파일에 접근한 시간을 나타냄.      ls -ult  
c time : 파일의 속성이 변경된 시간을 나타냄.      ls -clt



< mac time 적용 >

(2) find 명령과 함께 정규식을 사용하면 보다 다양한 검색 가능

- 사용 예 : /usr/bin 디렉토리에서 첫 글자가 v 또는 b로 시작하면서 두 번째 글자가 o 또는 x인 모든 파일을 찾는 명령  
=> # find /usr/bin -name "[v,b][o,x]\*" -print      : -print는 생략가능

(3) find에 의해 검색된 파일을 다른 명령어로 처리하기

- find 명령은 파일을 검색하여 잇따른 명령이 적용될 수 있도록 다른 명령어와 함께 사용할 수 있는 행동 옵션 지원
- 지원되는 행동 옵션 : -exec 명령어 {} W;  
                                  -ok    명령어 {} W; => 실행여부 확인
- 사용 예 : # find /test -perm -4000 -exec rm {} W;  
                  => test라는 디렉토리 및 하위 디렉토리에서 SUID가 적용된 파일을 찾아 삭제.

```
find /download -maxdepth 1 -type f -name '[d]*' -ok rm {} W;
=> download 디렉토리에서 'd'로 시작하는 파일만 찾아서 삭제. -ok 옵션이
있으므로 지우기 전에 확인함.
```

#### (4) grep(global regular expression print)

- 파일에서 정규표현식에 맞는 텍스트 라인을 찾아서 출력해 주는 명령
- 정규표현식이란 사전에 정의된 특수 문자로 문자열 패턴을 표현하는 것을 말함.

| 정규표현식 | 내 용                                                                                  | 예 제          |
|-------|--------------------------------------------------------------------------------------|--------------|
| ^     | 라인의 시작                                                                               | '^pattern'   |
| \$    | 라인의 끝                                                                                | 'pattern\$'  |
| \<    | 단어의 시작                                                                               | '\<love'     |
| \>    | 단어의 끝                                                                                | 'love>\'     |
| .     | 하나의 문자 일치                                                                            | 'p.....n'    |
| *     | 바로 앞의 문자를 나타내며 그 문자가 없거나 하나 이상을 의미                                                   | '[a-z]*'     |
| .*    | 모든 문자, '.'은 임의의 한 문자                                                                 | 'a.*'        |
| []    | 패턴(pattern)에 해당하는 한 문자. 찾는 문자열은 'Pattern','pattern'이다.                               | '[Pp]attern' |
| [x-y] | 일정한 범위 안에 오는 하나의 문자. 대문자로 시작하고 'ove' 문자열이 있는 패턴 찾기                                   | '[A-Z]love'  |
| [^]   | 패턴(pattern)에 해당하지 않는 한 문자. 대문자가 아닌 문자열이 있고 'ove' 문자열이 있는 패턴 찾기                       | '[^A-Z]love' |
| \     | 이스케이프 시퀀스(Escape Sequence)라고 부른다. 역슬래시(\) 뒤에 나오는 특수기호의 의미를 없애준다. 'love.'이 들어간 문자열 찾기 | 'love\.'     |

- 사용법 : grep [옵션] '패턴' 파일명  
명령어 | grep [옵션] '패턴'
- 옵션
  - -w : grep은 패턴을 포함하기만 하면 되지만, 이 옵션을 쓰면 정확히 일치해야만 검색됨.
  - -v : 패턴을 포함하지 않은 라인을 검색하고자 할 때 사용.
  - -c : 찾고자 하는 패턴이 포함된 라인의 수를 출력.



## Chapter 08. 사용자 계정 관리

### 8.1 계정과 그룹 정보

#### 1. 계정과 그룹 정보 관련 파일

##### (1) /etc/passwd

- 시스템에 존재하는 모든 계정의 정보가 기록된 파일
- 사용자의 로그인 요청이 있을 경우 가장 먼저 읽어 들이는 보안상 중요한 파일
- 계정 정보는 총 7개의 필드(계정 이름, UID, GID, Home Directory, Shell 등)로 구성
- 각 필드는 콜론(:)으로 구분
- 각 필드 값은 명령어나 vi Editor를 이용하여 수정 가능

[그림 8.1] /etc/passwd



- 첫 번째 필드 : 사용자 계정 이름
  - /etc/shadow, /etc/group 등의 사용자 정보를 갖는 여러 파일에서 사용자에게 대한 정보를 찾는 기준이 되는 필드이며 사용자의 Login-ID를 정의

##### 「사용자 이름의 조건」

- 사용자의 이름은 중복되지 않는 유일한 이름을 사용
- 사용자 이름에 사용하는 문자열은 영문자와 숫자로 제한
- 사용자의 이름에는 공백이 없어야 하며, '-', '\_'와 같은 기호는 사용 불가
- 사용자의 이름의 첫 문자열은 숫자도 가능

- 두 번째 필드 : 계정의 패스워드(실제 패스워드는 /etc/shadow에 기록)
  - 이전 버전의 리눅스는 실제로 이 위치에 패스워드를 저장하였으나 현재의 리눅스는 'x'로 대체하며 /etc/shadow 파일에 실제의 패스워드를 암호화하여 저장
- 세 번째 필드 : UID(시스템에서 사용자를 판단하기 위한 숫자)
  - UID(User IDentification)는 각각의 사용자 계정을 구별하는 숫자를 의미
  - 서로 다른 사용자 계정의 UID 중복은 가능하지만 권장하지 않음.
  - 0~99번은 시스템을 위해 예약되어 있고 일반 사용자 계정은 500번부터 지정이 가능
- 네 번째 필드 : GID(시스템에서 그룹을 판단하기 위한 숫자)
  - GID(Group IDentification)는 각각의 사용자 계정이 속한 그룹을 의미
  - 0~99번은 시스템을 위해 예약되어 있고 일반 그룹은 500번부터 지정이 가능
- 다섯 번째 필드 : 사용자 계정에 대한 Comment
  - 사용자 계정에 대한 주석을 정의하며 반드시 입력해야 하는 값은 아님
- 여섯 번째 필드 : 사용자 계정의 Home Directory
  - 사용자 계정이 로그인 할 경우 가장 처음 위치하는 Home Directory를 지정

- 일반 사용자 계정은 보통 '/home' Directory 하위에 Home Directory를 생성
- 일곱 번째 필드 : 사용자 계정이 로그인 할 경우 사용하는 Shell
  - 사용자 계정이 로그인 할 경우 사용하는 Shell을 지정
  - /etc/shells 파일에 사용가능한 모든 Shell이 정의되어 있으며 리눅스 시스템은 기본적으로 Bash Shell(/bin/bash)을 사용

## (2) /etc/shadow

- 각 사용자의 실제 패스워드(암호화된)가 암호화되어 저장되는 파일
- root(루트) 사용자에게 의해서만 파일 내용 확인이 가능
- 암호 정보는 총 9개의 필드로 구성되며 사용자 이름, 패스워드와 함께 패스워드를 보다 강력하게 보안하기 위한 패스워드 에이징 관련 정보를 포함 ⇨ 『page XXX』

### 「패스워드 에이징」

- 사용자의 현재 사용 중인 패스워드가 변경될 수 있는 최소의 시간과 최대의 시간을 지정하여 동일한 패스워드를 오랜 시간 사용하는 것을 방지하는 역할을 수행

- 각 필드는 콜론(:)으로 구분
- 사용자에게 의해 파일 내용이 직접 수정되어서는 안 됨

[그림 8.2] /etc/shadow



- 첫 번째 필드 : 사용자 계정 이름
  - /etc/passwd와 마찬가지로 사용자의 Login ID를 정의
- 두 번째 필드 : 사용자 계정의 패스워드
  - 각 사용자의 패스워드를 암호화 하여 저장
- 세 번째 필드 : 최종 암호 변경일
  - 사용자 계정의 마지막 패스워드 변경 일을 기록하며 1970년 1월 1일을 기준으로 며칠이 지났는지를 일수로 표현
- 네 번째 필드 : 암호 변경 최소 일 수
  - 최종 암호 변경 일을 기준으로 하여 최소 일 수가 지난 후에 암호 변경이 가능
- 다섯 번째 필드 : 암호 변경 최대 일 수
  - 현재 사용 중인 암호를 최종 암호 변경일 기준으로 하여 얼마나 더 사용 할 수 있는지를 표현
- 여섯 번째 필드 : 암호 변경 경고 일 수
  - 암호 변경 최대 일 수를 앞두고 며칠 전 부터 경고 메시지를 출력할 것인지를 표현
- 일곱 번째 필드 : 계정 사용 불가 날짜
  - 일반 사용자 계정의 로그인을 제한하기 위한 설정으로 설정된 날짜까지 계정이 로그인을 하지 않을 경우 사용자는 잠김
- 여덟 번째 필드 : 계정 만기일
  - 사용자 계정이 시스템을 마지막으로 사용할 수 있는 날짜를 표현하며, 만기일이 지난 사용자는 더 이상 로그인이 불가능

- 아홉 번째 필드 : 예약된 필드로서 현재 사용되지 않음

### (3) /etc/group

- 현재 시스템에 정의되어 있는 모든 그룹에 대한 정보가 기록된 파일
- 사용자 계정은 반드시 하나의 그룹에 소속이 되어야 하며, 사용자의 주 그룹은 /etc/passwd 파일에 정의되어 있는 GID로 부여됨
- 그룹은 사용자를 하나로 묶는 역할을 하여 권한 부여 등에서 효율적인 관리가 가능
- 총 4개의 필드로 구분되며 각 필드는 콜론(:)으로 구분

[그림 8.3] /etc/group



- 첫 번째 필드 : 그룹 이름  
→ 그룹의 이름을 의미
- 두 번째 필드 : 그룹의 패스워드(실제 패스워드는 /etc/gshadow에 기록)  
→ 이전 버전의 리눅스는 실제로 이 위치에 패스워드를 저장하였으나 현재의 리눅스는 'x'로 대체하며 /etc/gshadow 파일에 실제의 패스워드를 암호화하여 저장
- 세 번째 필드 : GID  
→ 해당 그룹을 표현하는 숫자
- 네 번째 필드 : member  
→ 해당 그룹에 속하는 사용자 계정의 목록을 표현하며 사용자들은 콤마(,)로 구분

### (4) /etc/default/useradd

- 사용자 계정 생성 시 할당되는 기본 정보를 갖는 파일
- 기본 설정 값의 변경 방법으로는 vi Editor를 이용하여 직접 내용을 변경하는 방법과 useradd 명령어를 이용하여 변경하는 방법을 이용

[그림 8.4] /etc/default/useradd



- GROUP : 기본 그룹의 ID를 지정
- HOME : 사용자의 홈 디렉토리의 경로를 지정
- INACTIVE : 사용자의 패스워드 사용 기간이 만료된 후 계정이 비활성화 되기까지의 날짜를 의미  
→ 0 = 패스워드 만료 후 즉시 계정 비활성화, -1 = 해당 기능을 사용하지 않음
- EXPIRE : 사용자의 패스워드 만료일 지정
- SHELL : 사용자가 로그인할 때 사용하는 기본 Shell 지정

- SKEL : 사용자가 만들어질 때 사용하게 될 기본적인 환경설정 파일이 존재하는 위치
- CREATE\_MAIL\_SPOOL : 사용자의 메일계정 파일 생성 여부

- useradd 명령어를 이용한 기본 설정 값 확인 및 변경

- # useradd -D → 기본 설정 값 확인
- # useradd -D -s /bin/sh → 기본 설정 값의 Shell 항목의 설정 값 변경
  - -g : 기본 그룹 변경
  - -b : 기본 Home Directory 변경
  - -f : INACTIVE 값 변경
  - -e : 기본 계정 종료일 변경
  - -s : 기본 Shell 변경

[그림 8.5] /etc/default/useradd 설정 값 변경

```

[root@rocky ~]# useradd -D
GROUP=100
HOME=/home
INACTIVE=-1
EXPIRE=
SHELL=/bin/bash
SKEL=/etc/skel
CREATE_MAIL_SPOOL=yes
[root@rocky ~]# useradd -D -s /bin/sh
[root@rocky ~]# useradd -D
GROUP=100
HOME=/home
INACTIVE=-1
EXPIRE=
SHELL=/bin/sh
SKEL=/etc/skel
CREATE_MAIL_SPOOL=yes
[root@rocky ~]#

```

(5) /etc/login.defs

- 시스템에 존재하는 전체 사용자 계정을 제한하기 위해 사용하는 설정 파일

- MAIL\_DIR /var/spool/mail
  - 사용자의 메일 박스 위치를 지정, 기본 값 : '/var/spool/mail/사용자명'
- PASS\_MAX\_DAYS 99999
  - 암호 변경 최대 일수를 지정
- PASS\_MIN\_DAYS 0
  - 암호 변경 최소 일수를 지정
- PASS\_MIN\_LEN 5
  - 최소한의 패스워드 길이를 지정
- PASS\_WARN\_AGE 7
  - 암호 변경 경고 일수를 지정
- UID\_MIN 500
  - 사용자 계정을 생성할 때 부여할 수 있는 최소 UID 설정
- UID\_MAX 60000
  - 사용자 계정을 생성할 때 부여할 수 있는 최대 UID 설정
- GID\_MIN 500
  - 그룹 사용자를 생성할 때 부여할 수 있는 최소 GID 설정

- GID\_MAX      60000  
→ 그룹 사용자를 생성할 때 부여할 수 있는 최대 GID 설정
- CREATE\_HOME      yes  
→ 사용자 계정을 생성할 때 사용자의 Home Directory를 생성할 것인지의 여부

(6) /etc/skel

- 사용자를 생성할 경우 사용자의 Home Directory에 기본적으로 생성 되는 환경 파일이 위치하는 디렉토리

[그림 8.6] /etc/skel 디렉토리에 저장되어 있는 파일



## 8.2 사용자 계정과 그룹 관리

### 1. 사용자 관리

(1) useradd(사용자 생성)

- 시스템에 새로운 사용자를 추가할 경우에 사용하는 명령어
- 명령어 위치 : /usr/sbin/useradd
- 명령어 사용 방법 : # useradd <option> <username>
- option
  - -u : 사용자 계정의 UID 지정
  - -g : 사용자 계정의 주 그룹의 GID 지정
  - -G : 사용자 계정의 보조 그룹의 GID 지정
  - -d : 사용자 계정의 홈 디렉토리를 지정
  - -e : 사용자 계정의 만료일 지정
  - -f : 사용자 계정의 유효일 지정
  - -s : 사용자 계정이 로그인 쉘 지정
  - -M : 사용자 계정의 Home Directory를 생성하지 않음
  - -c : 사용자 계정의 Comment 지정
- 사용 예
  - [root@rocky /]# useradd user01  
→ 'user01' 이라는 이름의 계정 생성
  - [root@rocky ~]# useradd -d /user02 -s /bin/sh user02  
→ Home Directory는 '/user02', 로그인 쉘은 '본 쉘(/bin/sh)'을 사용하는 'user02' 계정 생성
  - [root@rocky ~]# useradd -M user03  
→ Home Directory를 만들지 않고 'user03' 계정 생성
  - [root@rocky ~]# useradd -e 2022-12-15 user04

---

→ 계정 만료일이 2022년 12월 15일인 'user04' 계정 생성

(2) passwd(사용자 패스워드 생성, 변경, 삭제)

- 암호 지정 명령어
  - 명령어 위치 : /usr/bin/passwd
  - 명령어 사용 방법 : # passwd <option> <username>
  - option
    - -l : 사용자 계정을 잠근다.
    - -u : 사용자 계정을 푼다.
    - -d : 사용자 계정의 패스워드를 삭제한다.
    - -S : 사용자 계정의 패스워드 상태를 확인한다.
- ```
[root@rocky ~]# passwd student
Changing password for user student.
New UNIX password :
Retype new UNIX password :
passwd : all authentication tokens updated successfully.
```
- 'student' 계정의 password로 사용할 문자열 입력 후 재입력

「암호 설정의 권고안」

- 최소 6자 이상의 길이
- 최소한 1주일에 1회 암호 변경
- 적어도 2개의 알파벳과 1개 이상의 숫자 또는 특수 문자 포함
- 쉽게 연상할 수 있는 단어, 특정 문자의 반복, 개인 신상 관련 내용은 배제
- 'space bar(공백 문자열)' 포함 가능

(3) usermod(사용자 속성 변경)

- 이미 생성된 사용자의 속성을 변경할 경우에 사용하는 명령어
 - 명령어 위치 : /usr/sbin/usermod
 - 명령어 사용 방법 : # usermod <option> <username>
 - option
 - -u : 사용자 계정의 UID 변경
 - -g : 사용자 계정의 주 그룹의 GID 변경
 - -G : 사용자 계정의 보조 그룹의 GID 변경
 - -d : 사용자 계정의 홈 디렉토리를 변경
 - -e : 사용자 계정의 만료일 변경
 - -f : 사용자 계정의 유효일 변경
 - -s : 사용자 계정이 로그인 쉘 변경
 - -c : 사용자 계정의 Comment 변경
 - 사용 예
 - [root@rocky ~]# usermod -d /home/user02 -s /bin/sh user02
- 'user02'의 Home Directory를 '/home/user02'로 로그인 쉘을 /bin/sh로 변경

(4) userdel(사용자 삭제)

- 시스템에 존재하는 사용자를 삭제할 경우에 사용하는 명령어
- 명령어 위치 : /usr/sbin/userdel

-
- 명령어 사용 방법 : # userdel <option> <username>
 - option
 - -r : 사용자 계정의 Home Directory와 mail 저장파일(/var/spool/mail/계정명)까지 함께 삭제
 - 사용 예
 - [root@rocky ~]# userdel user04
→ 시스템에서 'user04' 계정을 삭제
 - [root@rocky ~]# userdel -r user03
→ 'user03' 계정을 계정의 Home Directory와 함께 삭제

(5) chage

- 사용자 계정의 패스워드 에이징 설정과 관련하여 사용하는 명령어
- 명령어 위치 : /usr/bin/chage
- 명령어 사용 방법 : # chage <option> <username>
- option
 - -l : 패스워드 에이징 설정 확인
 - -m : 패스워드를 설정한 후 다음 변경이 가능한 날짜 지정
 - -M : 설정한 패스워드의 유효 날짜 지정
 - -i : 패스워드 유효 기간이 만료된 후 사용자 계정이 비활성화될 일 수 지정
 - -E : 사용자 계정 만료일
 - -d : 패스워드의 마지막 변경일 지정
 - -W : 패스워드 유효 기간이 끝나기 전에 경고 메시지를 보낼 날짜 지정
- 사용 예
 - [root@rocky ~]# chage -l user02
→ 'user02' 계정의 패스워드 에이징 설정 확인
 - [root@rocky ~]# chage -m 5 -M 100 -W 10 -E 2011/12/31 user02
→ 'user02' 계정의 패스워드를 5일 후 변경 가능
→ 'user02' 계정의 패스워드는 100일 동안 유효
→ 'user02' 계정의 패스워드 만료일 이전 10일 동안 경고 메시지 출력
→ 'user02' 계정은 2011년 12월 31일까지만 사용 가능하도록 제한

II. 그룹 관리

(1) groups(그룹 확인)

- 특정 사용자 계정이 속한 그룹을 확인하는 경우에 사용하는 명령어
- 명령어 위치 : /usr/bin/groups
- 명령어 사용 방법 : # groups <username>
- 사용 예
 - [root@rocky ~]# groups
→ 시스템에 현재 로그인한 사용자('root')가 속한 그룹 목록을 출력
 - [root@rocky ~]# groups user01
→ 'user01' 계정이 속한 그룹 목록을 출력

(2) groupadd(그룹 생성)

-
- 시스템에 새로운 그룹을 추가하는 경우에 사용하는 명령어
 - 명령어 위치 : /usr/sbin/groupadd
 - 명령어 사용 방법 : # groupadd <option> <groupname>
 - option
 - -g : 생성할 그룹의 GID 지정
 - -o : 생성할 그룹의 GID를 다른 그룹의 GID와 동일하게 지정
 - -r : 생성할 그룹에게 0~499번 사이의 GID 자동 지정
 - 사용 예
 - [root@rocky ~]# groupadd group1
→ 시스템에 'group1' 이라는 이름의 그룹을 생성
 - [root@rocky ~]# groupadd -g 1000 group2
→ 시스템에 'group2' 라는 이름의 그룹을 GID 1000번으로 생성

(3) gpasswd(그룹 패스워드 설정)

- 그룹에 패스워드를 설정하거나 그룹의 관리자를 지정하는 등의 경우에 사용하는 명령어
- 명령어 위치 : /usr/bin/gpasswd
- 명령어 사용 방법 : # gpasswd <option> <username> <groupname>
- option
 - -A : 그룹 관리자 추가
 - -a : 그룹 멤버 추가
 - -d : 그룹 멤버 삭제
 - -M : 그룹 멤버 변경
- 사용 예
 - [root@rocky ~]# gpasswd -A user01 group1
→ 'user01' 사용자를 'group1' 그룹 관리자로 지정
 - [root@rocky ~]# gpasswd -a user02 group1
→ 'user02' 사용자를 'group1' 그룹 멤버로 지정

(4) groupmod(그룹 속성 변경)

- 이미 생성된 그룹의 속성을 변경할 경우에 사용하는 명령어
- 명령어 위치 : /usr/sbin/groupmod
- 명령어 사용 방법 : # groupmod <option> <groupname>
- option
 - -g : 그룹의 GID 변경
 - -o : 그룹의 GID를 다른 그룹의 GID와 동일하게 변경
 - -n : 그룹명 변경
- 사용 예
 - [root@rocky ~]# groupmod -g 700 group1
→ 그룹 'group1'의 GID를 700번으로 변경
 - [root@rocky /]# groupmod -n group3 group1
→ 그룹 'group1'의 이름을 'group3'으로 변경

(5) groupdel(그룹 삭제)

- 이미 생성된 그룹의 속성을 변경할 경우에 사용하는 명령어
- 명령어 위치 : /usr/sbin/groupdel

-
- 명령어 사용 방법 : # groupdel <groupname>
 - 사용 예
 - [root@rocky ~]# groupdel group3
 - 시스템에서 'group3' 그룹을 삭제

III. 사용자 전환(su;substitute user)

- ; 로그인한 사용자가 아닌 다른 사용자의 권한이 필요한 경우 현재 사용자를 로그아웃하지 않고, 원하는 사용자 계정의 권한을 가질 수 있도록 하는 명령. 대개 관리자가 일반 사용자로 로그인한 후, 루트 권한을 요구하는 작업을 해야할 경우 사용한다.
- 사용법 : # su 옵션 사용자명
- 옵션 : 옵션없이 사용하면 다른 환경 변화없이 해당 사용자 계정으로만 바뀐다. 해당 사용자의 .bash_profile을 읽어들이고, 사용자 홈을 이용할 수 있도록 하려면 '-' 나 '-l'을 함께 사용한다.
- 사용자명을 지정하지 않으면 루트로의 전환을 의미한다. 또한 루트외에 모든 사용자는 su 사용시 전환하려는 사용자의 패스워드를 입력해야만 한다.

IV. 루트권한 행사하기(sudo)

- ; su명령에 의해 루트로 전환시 루트의 모든 권한을 갖게된다. 일반사용자에게는 이런 방식을 사용할 수 없다. 따라서 일반사용자에게 루트의 권한을 필요로 하는 명령을 실행할 수 있도록 하는 명령인 sudo를 사용하여 필요한 명령만을 실행할 수 있도록 한다.
- /etc/sudoers : 일반사용자가 sudo명령으로 루트 권한을 행사하기 위해서는 이 설정 파일에 일반사용자에 대한 권한 설정이 포함되어 있어야 한다. 이 설정파일을 편집하기 visudo 명령을 사용한다.
- /etc/sudoers 파일의 설정 형식

사용자	호스트	명령어
-----	-----	-----
- sudo -i : root 사용자로 전환

9.1 Process 및 서비스

I. Process란?

- System에서 동작하는 모든 프로그램
 - Find 같은 명령어를 실행하거나 Application 실행 시
 - 시스템에서 동작하는 모든 프로그램이 Process
- 시스템은 데몬(Daemon)이라는 Process도 시작
 - 데몬은 커널상에서 백그라운드 모드로 작동하며, 요청이 있을 때만 동작하는 프로세스
 - 데몬이 커널상에서 실행되고, 요청이 없어 작동하지 않을 때 CPU에는 부하를 주지 않지만 시스템 자원(메모리)을 차지하므로, 커널에 데몬이 많이 존재하면 자원이 부족한 시스템에서는 시스템이 느려지거나 다운될 경우도 있다. 따라서 시스템 관리자는 데몬관리에 주의를 요한다.(참고: Xinetd)
- 모든 프로세스는 유일한 값인 PID를 가짐
- 커널, 관리자는 PID를 이용하여 프로세스를 관리하고 추적

II. 서버 메모리 상태 확인 - free 명령어

- 메모리 중 유휴메모리 확인
- 사용 중인 메모리 양
- Swap 메모리 양
- 메모리의 전체 용량 확인
- 명령어 사용 방법: # free [Option]
 - -k: Kilo Byte 단위로 표시(Default)
 - -m: Mega Byte 단위로 표시
 - -t: 마지막에 Total 용량 표시
 - -s: 지정한 초 단위로 Refresh 후 내용 표시

```
[root@rocky ~]# free
              total        used         free       shared    buffers     cached
Mem:      254732      236688         18044            0         15076      126364
-/+ buffers/cache:      95248      159484
Swap:      1052248           0      1052248
```

- total: 전체 메모리에 대한 사용량 표시
- buffers: Buffering 된 메모리 표시
- cached: Caching 메모리 표시
- Swap: Swap 메모리 현황 설명

III. 프로세스 확인

(1) Parent Process

- 현재 프로세스를 만든 부모 프로세스
 - PPID(Parent Process Identifier)

- 부모에 의해 생성된 프로세스를 자식 프로세스라고 함

(2) ps Command

- 시스템에서 현재 동작하고 있는 프로세스 목록 표시

- PID
- Terminal Identifier(TTY)
- Cumulative Execution Time(TIME)
- Command(CMD)

- 명령어 사용 방법: # ps [Options]

- -a: 실행중인 프로세스 중 제어 터미널을 가지고 있는 프로세스의 목록 출력
- -u: 사용자 이름, 실행 시간 등 상세한 정보 출력
- -e: 현 시스템에서 동작하고 있는 모든 프로세스 목록
- -f: PID, TTY, TIME, CMD 이외에 UID, PPID, Start Time 출력
- -x: 터미널에 종속되지 않은 프로세스 출력

```
[root@rocky ~]# ps
  PID TTY          TIME CMD
 4232 pts/1    00:00:00 bash
 5490 pts/1    00:00:00 ps
```

```
[root@rocky ~]# ps -ef | more
UID          PID  PPID  C STIME TTY          TIME CMD
root           1     0  0 21:29 ?            00:00:00 init [5]

root           2     1  0 21:29 ?            00:00:00 [migration/0]
root           3     1  0 21:29 ?            00:00:00 [ksoftirqd/0]
....
```

- UID: 사용자의 UID
- PID: Process ID
- PPID: 부모 프로세스
- STIME: 프로세스가 시작된 시간
- TTY: 프로세스가 실행된 터미널
- TIME: 총 CPU 사용 시간
- CMD: 실행된 명령 라인

```
[root@rocky ~]# ps aux | more
USER          PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root           1  0.0  0.2 2916  548 ?        S    19:01   0:00 init [5]

root           2  0.0  0.0     0     0 ?        S    19:01   0:00 [migration/0]
root           3  0.0  0.0     0     0 ?        SN   19:01   0:00 [ksoftirqd/0]
..
```

- User: 프로세스의 권한을 갖고 있는 사용자
- %CPU: 프로세스가 사용하고 있는 CPU 점유율
- %MEM: 프로세스가 사용하고 있는 메모리 점유율
- VSZ: 프로세스가 사용하고 있는 가상 메모리의 크기
- RSS: 프로세스가 사용하고 있는 메모리의 크기
- STAT: 실행되고 있는 프로세스의 상태
 - S: Sleeping 상태, 중지 가능
 - R: 현재 동작중이거나 동작할 수 있는 상태

- Z: 좀비 상태
- <: 우선순위가 높은 프로세스
- N: 우선순위가 낮은 프로세스
- I: 멀티쓰레드
- s: Session Leader
- +: 포그라운드 상태로 동작하는 프로세스

(2) pstree Command

- 실행중인 프로세스 상태를 트리구조로 보여주는 명령어. 사용자별 프로세스 점검
- 사용형식 : pstree [-option] [pid|usr]

(3) top Command

- 실시간으로 프로세스들을 점검
- 사용형식 : top [-] [d delay] [p pid] [u user]

(4) pgrep Command

- 모니터링하고자 하는 프로세스만 출력해주는 명령어.
- 사용형식 : pgrep [-option] 패턴

IV. 프로세스 제어

(1) 시그널

- 프로세스 우선순위 조정 및 프로세스를 종료하도록 하는 메시지

시그널 이름	시그널 번호	시그널	명 령
SIGHUP	1	HUP	프로세스를 멈추지 않고 재실행
SIGINT	2	INT	키보드 인터럽트인 Ctrl+C와 동일, 프로세스를 종료
SIGKILL	9	KILL	시그널이 프로세스에 전달되면 커널이 직접 종료시킴, 강제 종료
SIGTERM	15	TERM	자신이 직접 프로세스를 종료, 데이터 저장 후 프로세스 종료

(2) kill/killall 명령어

- kill 명령어를 통해 시그널 전송
- 명령어 사용 방법: # kill [-Signal] [PID]

[root@rocky ~]# kill -l			
1) SIGHUP	2) SIGINT	3) SIGQUIT	4) SIGILL
5) SIGTRAP	6) SIGABRT	7) SIGBUS	8) SIGFPE
9) SIGKILL	10) SIGUSR1	11) SIGSEGV	12) SIGUSR2
13) SIGPIPE	14) SIGALRM	15) SIGTERM	16) SIGSTKFLT
17) SIGCHLD	18) SIGCONT	19) SIGSTOP	20) SIGTSTP
21) SIGTTIN	22) SIGTTOU	23) SIGURG	24) SIGXCPU
25) SIGXFSZ	26) SIGVTALRM	27) SIGPROF	28) SIGWINCH
29) SIGIO	30) SIGPWR	31) SIGSYS	34) SIGRTMIN
35) SIGRTMIN+1	36) SIGRTMIN+2	37) SIGRTMIN+3	38) SIGRTMIN+4
39) SIGRTMIN+5	40) SIGRTMIN+6	41) SIGRTMIN+7	42) SIGRTMIN+8
43) SIGRTMIN+9	44) SIGRTMIN+10	45) SIGRTMIN+11	46) SIGRTMIN+12
47) SIGRTMIN+13	48) SIGRTMIN+14	49) SIGRTMIN+15	50) SIGRTMAX-14
51) SIGRTMAX-13	52) SIGRTMAX-12	53) SIGRTMAX-11	54) SIGRTMAX-10
55) SIGRTMAX-9	56) SIGRTMAX-8	57) SIGRTMAX-7	58) SIGRTMAX-6
59) SIGRTMAX-5	60) SIGRTMAX-4	61) SIGRTMAX-3	62) SIGRTMAX-2
63) SIGRTMAX-1	64) SIGRTMAX		

- bash명령을 통해 자식 쉘을 더 만들고 kill 명령으로 제거해 본다.
- kill 명령의 기본 시그널은 15(SIGTERM)이므로 강제로 프로세스를 종료시키기 위해서는 9(SIGKILL) 시그널을 보낸다.
- killall 명령은 해당 이름의 프로세스를 모두 종료시킨다.
killall httpd

(3) pkill 명령어

- pkill 명령어를 통해 시그널 전송
- 명령어 사용 방법: # pkill [-Signal] [Pattern]

```
[root@rocky ~]# sleep 300 &
[root@rocky ~]# pkill -9 sleep
```

(4) nice

- 프로세스에게 우선권 순위 설정
- 명령어 사용 방법: # nice -n command
 - n(우선권 순위) : -20 ~ 19까지 설정 가능. 기본값 10. 작을수록 높은 우선권.

```
[root@localhost ~]# ps -l
F S  UID  PID  PPID  C PRI  NI ADDR SZ WCHAN  TTY          TIME CMD
4 S   0 31283 31279  0 75   0 - 1511 wait  pts/3    00:00:00 bash
4 S   0 31673 31283  0 87  10 - 1228 -    pts/3    00:00:00 sleep
4 R   0 31796 31283  0 77   0 - 1353 -    pts/3    00:00:00 ps
[root@localhost ~]# nice --15 top &
[4] 31797
[root@localhost ~]# sleep 3000 &
[5] 31798

[4]+  Stopped                  nice --15 top
[root@localhost ~]# ps -l
F S  UID  PID  PPID  C PRI  NI ADDR SZ WCHAN  TTY          TIME CMD
4 S   0 31283 31279  0 75   0 - 1511 wait  pts/3    00:00:00 bash
4 S   0 31673 31283  0 87  10 - 1228 -    pts/3    00:00:00 sleep
4 T   0 31797 31283  0 62 -15 -  537 finish pts/3    00:00:00 top
0 S   0 31798 31283  0 77   0 - 1228 -    pts/3    00:00:00 sleep
4 R   0 31801 31283  0 77   0 - 1353 -    pts/3    00:00:00 ps
[root@localhost ~]#
```

- PRI: 시스템이 NI 데이터를 이용하여 계산한 실제 우선순위, 변경 불가능
- NI: 사용자가 변경 가능한 값

(5) renice

- 프로세스의 우선권 순위 변경
- 명령어 사용 방법: # snice [n/-n] [-p] pid [-u user]
 - n(우선순위) : -20 ~ 19까지 설정 가능. 기본값 10. 작을수록 높은 우선권.

```
[root@localhost ~]# ps -l
F S  UID  PID  PPID  C PRI  NI ADDR SZ WCHAN  TTY          TIME CMD
4 S   0 31283 31279  0 75   0 - 1511 wait  pts/3    00:00:00 bash
4 T   0 31797 31283  0 60 -20 -  537 finish pts/3    00:00:00 top
0 S   0 31798 31283  0 87  10 - 1228 -    pts/3    00:00:00 sleep
4 R   0 31978 31283  0 77   0 - 1353 -    pts/3    00:00:00 ps
[root@localhost ~]# renice -5 31798
31798: old priority 10, new priority -5
[root@localhost ~]# ps -l
F S  UID  PID  PPID  C PRI  NI ADDR SZ WCHAN  TTY          TIME CMD
4 S   0 31283 31279  0 75   0 - 1511 wait  pts/3    00:00:00 bash
4 T   0 31797 31283  0 60 -20 -  537 finish pts/3    00:00:00 top
0 S   0 31798 31283  0 72 -5 - 1228 -    pts/3    00:00:00 sleep
4 R   0 31982 31283  0 77   0 - 1353 -    pts/3    00:00:00 ps
[root@localhost ~]#
```

9.2 job control

멀티태스킹 시스템인 리눅스는 명령을 실행하여 프로세스를 만들 때 백그라운드(background)와 포그라운드(foreground)로 만드는 방법이 있음.

I. 백그라운드(background) 실행

- 키보드의 입력없이 프로세스를 장시간 실행해야하는 경우에 사용하는 방법
- 일반적인 명령 뒤에 앰퍼센트(&)를 붙이게 되면, 작업 번호와 PID를 출력하면서 명령이 실행되고, 다음 명령을 입력받을 수 있는 대기 상태가 됨.

```
[root@rocky ~]# sleep 500&
[1] 14310
[root@rocky ~]#
```

II. 포그라운드(foreground) 실행

- 작업 종료시까지 다른 작업을 수행하지 못함.

III. Job 관리 : jobs

- 터미널에 의해 제어되는 프로세스로 모든 Job은 셸에 의해 작업번호가 부여됨.
- 사용자가 실행시킨 프로세스의 리스트를 확인하는 셸 내부명령
- 백그라운드에서 실행되는 작업을 확인하기 위해 사용.
- 사용법 : # jobs

```
[root@rocky ~]# jobs
[1]+  Running                  sleep 500 &
[root@rocky ~]#
```

IV. 포그라운드 작업 전환 : fg

- 백그라운드로 실행되고 있는 작업을 포그라운드로 전환하기 위한 명령
- 사용법 : # fg %n (n은 작업번호)

V. 백그라운드 작업 전환 : bg

- 포그라운드로 실행되고 있는 작업을 백그라운드로 전환하기 위한 명령
- 사용법 : # bg %n (n은 작업번호)

VI. nohup

- 사용자가 시스템에 작업을 실행한 후 로그아웃을 해야하는 경우에 사용하는 명령
- 백업이나 tar를 통해 파일을 묶는 경우에 유용하게 사용됨.

```
[root@rocky ~]# nohup tar zcvf /test/backup.tar /etc &
[1] 14829
[root@rocky ~]# nohup: appending output to `nohup.out'
```

9.3 데몬 관리

부팅시 자동적으로 데몬이 적재될 수 있도록 선택하려면 `systemctl enable` 데몬명을 실행하면 된다. X-window 상에서는 `system-config-services`를 이용하여 데몬 관리

9.4 at / cron : 작업스케줄 자동화

I. at

- 특정 시간에 특정 작업을 자동으로 실행할 때 사용하는 명령어
- 명령어 사용 방법: # at [-m] [-r 작업 번호] [시간] [날짜]
 - at 명령어를 사용하기 위한 데몬: atd
- 날짜 시간 확인 후 작업 지정

```
[root@rocky ~]# date
2010. 07. 08. (목) 19:14:04 KST

[root@rocky ~]# at -m 2:00 tomorrow
at> shutdown -r now
job 1 at 2010-07-09 02:00
```

(1) at 작업 확인

- 설정 내용 확인

```
[root@rocky ~]# atq
1          2010-07-09 02:00 a root

[root@rocky ~]# at -l
1          2010-07-09 02:00 a root
```

(2) at 작업 삭제

- 지정된 at 작업 삭제 : atrm <작업번호>

(3) at 사용자 제한

- 특정 사용자에게 at 명령어 제한
- /etc/at.allow, /etc/at.deny 파일을 이용해 사용자 제한

```
[root@rocky ~]# cat > /etc/at.allow
user1

[root@rocky ~]# cat /etc/at.allow
```

II. cron

- 주기적으로 반복하는 작업을 실행
- 명령어 사용 방법: # crontab [-u user] [Option] [File]
 - -e: 작업할 내용 작성
 - -l: 현재 설정 내용 확인
 - -r: 설정 내용 삭제
- cron 데몬: crond
- crontab 구성: [분] [시간] [날짜] [달] [요일] [Command]
 - 요일: 0 ~ 6 (0: 일요일)

```
[root@rocky ~]# pgrep -fl crond
2957 crond
[root@rocky ~]# crontab -l
no crontab for root
[root@rocky ~]# crontab -u test -e
```

(1) /etc/crontab

- 시스템에서 주기적으로 해야 할 작업 설정

```
[root@rocky ~]# cat /etc/crontab
SHELL=/bin/bash
PATH=/sbin:/bin:/usr/sbin:/usr/bin
MAILTO=root
HOME=/

# run-parts
01 * * * * root run-parts /etc/cron.hourly
02 4 * * * root run-parts /etc/cron.daily
22 4 * * 0 root run-parts /etc/cron.weekly
42 4 1 * * root run-parts /etc/cron.monthly
```

(2) cron 사용자 제한

- 모든 사용자에게 허용되지만 /etc/cron.allow, /etc/cron.deny 파일 통해 접근 제어

```
[root@rocky ~]# cat /etc/cron.allow
[root@rocky ~]# cat > /etc/cron.allow
test
```


10.1 File System

I. Linux File System

(1) 파일시스템

- 컴퓨터에 파일들과 디렉토리들을 저장하고 빠르게 검색하기 위해 논리적으로 구성한 것.
- 파일시스템은 물리적인 저장 공간(하드디스크, CD-ROM), 네트워크(NFS) 등 구성이 가능.
- 모든 파일시스템은 독특한 특징을 가지고 있지만, 리눅스들의 파일시스템은 거의 비슷함.

II. File System 종류

(1) ext

- 리눅스 초기에 사용되던 파일시스템
- 호환성이 없던 ext2의 구 버전
- 현재는 거의 사용하지 않음

(2) ext2

- ext3 파일시스템의 기본 파일시스템
- ext3 파일시스템이 사용되기 전 가장 많이 사용하던 파일시스템
- 오랜 기간 사용으로 안정적이고 호환성이 높음

(3) ext3

- ext2 파일시스템의 단점을 보완해 Redhat 7.2 버전부터 기본 파일시스템으로 사용
- 저널링 기능을 포함
- 무결성 및 뛰어난 복구 기능 포함
- 데이터 신뢰성 강화
 - ext2의 데이터 포맷을 공유
- 가용성
 - 시스템의 비정상 종료 발생하더라도 시스템의 일관성 유지를 위해 빠르게 수행함
- 손쉽게 ext2에서 ext3로 변환이 가능
- 하드디스크의 헤드 모션을 최소화(동일한 데이터를 저장하더라도 빠른 속도로 작업)

「저널링(Journaling) 기술」

기존의 fsck 명령어를 실행하는데 걸리는 시간을 단축하기 위해 데이터를 디스크에 쓰기 전 로그에 남겨 시스템이 비정상 종료를 하더라도 로그를 위해 fsck 명령어를 수행, 데이터를 빠르고 안정적으로 복구 기능을 제공

(4) ext4

- ext3 파일시스템의 단점인 쓰기속도를 향상 시킴.

(5) xfs

- 1993년 실리콘 그래픽스(SGI)가 만든 고성능 64비트 저널링 파일 시스템이다.
버전 5.3을 기점으로 SGI의 IRIX 운영 체제의 기본 파일 시스템이었고, 2001년에는 리눅스 커널로 포팅되었다. 2014년 6월 기준으로 XFS는 대부분의 리눅스 배포판에서

지원되며, 그중 일부는 기본 파일 시스템으로 채택하고 있다.

10.2 계층적 디렉토리 구조

1. Linux 주요 디렉토리

- Linux System은 데이터를 효율적으로 관리하고자 Directory Tree 구조를 사용
- 최상위 디렉토리로서 /(root-루트) 디렉토리가 존재

(1) 루트(/) Directory

- "/" Directory는 Linux 계층적 파일 구조에서 최상위에 존재하는 단 하나의 디렉토리
- Linux에 존재하는 모든 시스템 운영 File과 Directory 포함
- 모든 File과 Directory가 시작하는 곳
- 절대 경로 방식의 기준이 되는 Directory

(2) 기본 실행 명령어(/bin) Directory

- Linux System을 사용하면서 가장 많이 사용하는 디렉토리 중 하나
- 시스템 관리 명령어를 제외한 기본 명령어 파일들이 위치

(3) 시스템 운영 명령어(/sbin) Directory

- Standalone Binary Directory
- System 운영에 필요한 명령어 파일들이 위치
- root(Super User)만이 사용할 수 있는 명령어들로 구성

(4) 시스템 설정 파일(/etc) Directory

- System의 전반적인 관리 목적의 설정 파일들이 포함된 디렉토리
- 백업이 매우 중요
- 주요 파일
 - /etc/fstab : 마운트 정보
 - /etc/passwd : 계정 정보와 패스워드, 사용하는 Shell, Home Directory 정보
 - /etc/shadow : /etc/passwd 파일의 패스워드 정보가 암호화되어 저장
 - /etc/group : 사용자 그룹에 대해 정의
 - /etc/cron.* : 시스템에 주기적인 스케줄링 시 이용하는 cron 데몬 관련 설정
 - /etc/xinetd.d : xinetd 데몬에 의해 동작하는 서비스 데몬들의 설정 파일 포함
 - /etc/init.d : Linux System의 부팅 시 실행되는 프로그램들을 실행하거나, System 종료 시 작동 중인 프로그램을 중단하는 Shell Script 존재
 - /etc/skel : 사용자 계정을 생성할 때 계정의 Home Directory에 생성되는 환경 파일들이 존재
 - /etc/mail : Sendmail의 설정 파일들이 존재

(5) 마운트(/media) Directory

- CD-ROM, Flopy Disk, USB Memory등이 임시로 마운트 되는 디렉토리
- 기존에는 주로 /mnt 디렉토리를 이용하여 마운트

-
- (6) 부트 파일(/boot) Directory
 - Linux System 부팅에 관련된 모든 파일이 포함된 디렉토리
 - (7) 장치 파일(/dev) Directory
 - Linux System에서 인식한 모든 장치들이 파일의 형태로 존재
 - (8) 사용자 홈(/home) Directory
 - 일반 계정들의 Home Directory 존재
 - (9) 라이브러리(/lib) Directory
 - 명령어나 응용 프로그램이 실행되거나 새로운 프로그램을 개발하는데 필요한 동적 라이브러리와 정적 라이브러리 존재
 - (10) 응용 프로그램(/usr) Directory
 - 실행 가능한 명령어, 시스템 프로그램, 라이브러리 루틴 등이 포함
 - (11) 가변 자료 저장(/var) Directory
 - Linux가 운영되거나 여러 프로그램들이 실행되면서 만들어지는 Log 파일이나 Spool 파일들이 존재
 - (12) 프로그램 추가 설치(/opt) Directory
 - Linux에서 Bundle로 제공되지 않는 프로그램을 추가로 설치할 경우에 사용
 - (13) lost+found Directory => xfs 파일시스템에는 없음.
 - Disk의 오류나 부적절한 시스템 종료에 의해 잃어버리는 파일을 위해 사용
 - 파티션 별로 하나의 lost+found Directory 가짐.

II. 파일명과 경로명

- (1) File Name(파일명)
 - Linux의 File Name은 대·소문자를 엄격하게 구분
 - Windows System에서 사용하는 확장자 형식이 없음
 - File Name이 “.”으로 시작하면 숨김 파일(→ 주로 "ls -a" 명령어로 확인)
- (2) Path Name(경로명)
 - Linux System은 계층적 트리 구조로 형성되어 있기 때문에 동일한 이름의 파일은 가질 수 있지만 동일한 Directory 내에서는 파일명이 중복될 수 없음
 - Directory Tree 구조 안에 있는 File 혹은 Directory의 위치를 나타내는 방법
 - Directory와 Directory, Directory와 File을 구별하는 구분자로서 “/”를 사용
 - 절대 경로명(Absolute Path Name)
 - 최상위 디렉토리(/)를 기준으로 나타내는 File과 Directory의 위치
 - 사용자의 현재 위치와 상관없이 변함이 없음
 - 절대 경로의 표현은 최상위 디렉토리(/)부터 현재 디렉토리까지의 위치를 전부 표현
 - 상대 경로명(Relative Path Name)

- 현재의 위치를 기준으로 File과 Directory의 위치를 표현
- 사용자의 현재 위치에 따라 변함
- 상대 경로의 표현 시 사용되는 디렉토리에 대한 약어
 - "." : 현재 디렉토리를 의미
 - ".." : 현재 위치한 디렉토리의 상위 디렉토리를 의미

10.3 파일 관리

- Linux에서는 모든 것이 파일 단위로 구성되며 계층적 구조를 형성
- 파일은 생성, 수정, 삭제 및 보관이 가능하며 이름과 크기 등의 특성을 가짐
- 일반 파일이나 디렉토리 파일 외에도 링크 파일, 디바이스 파일, 소켓 파일, 파이프 파일 등의 다양한 종류의 파일이 존재

I. File의 구성요소

(1) File name

- 사용자가 파일에 접근하고 조작하기 위해서 사용.

(2) Inode

- 파일을 기술하는 디스크 상의 데이터 구조로써 파일의 Data block 위치 등을 저장.
- 각각의 파티션마다 0부터 시작하는 정수의 형태.
- Inode는 고유한 식별번호를 가지는데 이 숫자는 파일시스템이 가질 수 있는 최대 파일의 수
- Inode에 저장되는 정보는 다음과 같다.
 - 파일의 소유권
 - 파일의 액세스 모드
 - 파일의 타임스탬프(파일의 마지막 수정, 액세스)
 - 파일의 종류
- Inode 확인
 - "ls -li" : 파일의 Inode 확인, "df -li" : 파일시스템의 Inode 확인

(3) Data Block

- 데이터 블록은 실제 파일 데이터 영역으로 각 파일의 데이터는 이 영역에 블록 단위로 분산 저장되며, 이 저장된 내용에 따라 파일의 종류가 결정됨.
- 일반파일, 디렉토리, 심볼릭 링크는 data block을 사용함.

II. 파일의 종류

(1) 일반 파일(Regular File)

- 일반적인 거의 대부분의 파일
- ls -li 실행 결과의 첫 번째 필드에서 파일 형식이 "-"로 표시
- 파일 내부 구조에 제한이 없어 어떠한 형식의 데이터도 저장이 가능
 - 예) Database, 동영상, 음악, 텍스트, 실행 가능한 Binary 등..

(2) 디렉토리 파일(Directory File)

- 서로 연관되어 있는 파일들을 하나의 그룹으로 만들어 저장하도록 구분된 공간

-
- Inode에 대한 Pointer(Inode 번호)만 저장

(3) 링크 파일(Link File)

- 여러 파일이 하나의 I-node에 연결되기 때문에 실제 파일이나 디렉토리, 또는 또 다른 링크를 가리키도록 연결
- Windows System의 “바로 가기”, “단축 아이콘”은 파일 관리자가 담당하는 기능이지만 Linux의 링크 파일은 파일시스템 차원에서 제공
- 심볼릭 링크(소프트 링크)
 - Windows의 “바로 가기”와 유사
 - 실제 파일의 경로를 가리키는 파일
 - 심볼릭 링크 파일이 삭제가 되어도 실제 원본 파일에는 영향을 미치지 못함
 - 서로 다른 파일시스템 내에서도 링크가 가능
 - ls -l 실행 결과에서 File Type이 “l”로 표시되며 “->” 표시를 이용해 원본 파일의 위치를 표시
 - 명령어 위치 : /bin/ln
 - 명령어 사용 방법 : # ln -s <원본 파일> <심볼릭 링크 파일>
- 하드 링크
 - 원본 파일을 복사한 사본 파일
 - 원본 파일의 내용을 수정하면 원본의 하드 링크 파일의 내용까지 수정되어 동일한 내용 유지가 가능
 - 원본과 동일한 I-node 번호를 갖으며 Link Count 증가
 - 명령어 위치 : /bin/ln
 - 명령어 사용 방법 : # ln <원본 파일> <하드 링크 파일>

(4) 장치 파일(Device File)

- Linux에서는 모든 장치를 파일로 간주하여 하드웨어나 주변 기기를 나타내는 파일
- ls -l 실행 결과에서 File Type이 “c” 또는 “b”로 표시
- 블록 디바이스 파일(Block Device File)
 - 수십 혹은 수백 바이트 크기인 블록 단위로 데이터를 입·출력
 - 효과적이고 신속한 입·출력을 위하여 버퍼링 기법을 도입
 - Hard Disk, Floppy Disk, Tape Drive 등의 장치가 블록 디바이스 파일을 사용
- 캐릭터 디바이스 파일(Character Device File)
 - 시스템의 I/O 버퍼를 사용하지 않고 바이트 단위로 데이터를 입·출력
 - Terminal, Printer, 보조 기억 장치 등의 장치가 캐릭터 디바이스 파일을 사용

10.4 허가권(permission)과 소유권(ownership)

1. 허가권의 이해

(1) 허가권의 필요성

- Linux System은 모든 파일과 디렉토리에 '소유자', '소유 그룹', '기타 사용자' 별로 허가권(Permission)을 지정
- 허가권은 크게 Read(읽기), Write(쓰기), eXecute(실행)으로 구분

- 각 파일과 디렉토리에 접근한 계정들이 해당 자원에 적절한 허가권을 행사할 수 있도록 올바른 허가권을 명시해야만 함

(2) 허가권의 확인

- ls -l 명령어 사용
- 결과의 첫 번째 필드는 총 10개의 문자로 구성되어 있으며 가장 첫 번째 문자는 파일의 유형을 의미하고 나머지 9개의 문자가 허가권을 의미
- 확인 예
 - [root@rocky ~]# ls -l /etc/hosts
 - rw-r--r-- 1 root root 156 12월 8 12:47 /etc/hosts
 - /etc/hosts 파일에 설정된 허가권 확인

(3) 파일 유형

- ls -l 명령어 결과 화면의 첫 번째 필드에서 첫 문자
- 대표적인 파일 유형의 종류

기호	의미
-	일반 파일
l	심볼릭 링크 파일
d	디렉토리 파일
b	Block Device 파일
c	Character Device 파일

(3) 파일 허가권(Permission)

- ls -l 명령어 결과 화면의 첫 번째 필드에서 파일 유형(첫 문자)을 제외한 나머지 9개의 문자
- 9개의 문자는 3개씩 나뉘서 해당 파일에 소유자(User), 소유 그룹(Group), 기타 사용자(Other)에 대한 허가권으로 구분
 - 소유자(User) : 9개의 문자 중에서 처음 3개의 문자
 - 소유자는 해당 파일이나 디렉토리를 생성하거나 해당 파일을 소유한 사용자를 의미
 - 소유 그룹(Group) : 소유자 권한의 다음 3개의 문자
 - 소유 그룹은 해당 파일이나 디렉토리에 권한을 갖는 시스템의 특정 그룹을 의미
 - 기타 사용자(Other) : 마지막 3개의 문자
 - 소유자, 소유 그룹에 포함되지 않는 기타 나머지 모든 사용자가 해당 파일과 디렉토리에 갖는 허가권을 의미
- 기본적인 퍼미션의 종류와 역할

기호	퍼미션	일반 파일	디렉토리
r	읽기 권한	파일 내용을 읽을 수 있는 권한	ls 등의 명령어를 이용하여 디렉토리 내의 리스트를 확인할 수 있는 권한
w	쓰기 권한	파일 내용을 쓰거나 지울수 있는 권한	디렉토리 내에 파일을 생성하거나, 삭제할 수 있는 권한
x	실행 권한	파일의 실행 권한	디렉토리 위치를 변경할 수 있는 권한

II. 파일의 소유권 변경

(1) 소유권 변경

- 새로운 파일이나 디렉토리를 생성하는 경우, 특정 파일을 복사 또는 이동을 할 경우, 네트워크를 통하여 다운로드 할 경우에는 항상 소유자와 소유 그룹이 지정되는데 시스템을 운영하는 도중 이를 변경해야 할 경우가 발생할 경우 사용하는 방법
 - `chown` : 파일의 소유자, 소유 그룹을 변경하고자 할 때 사용하는 명령어
 - `chgrp` : 파일의 소유 그룹을 변경하고자 할 때 사용하는 명령어

(2) `chown`

- 명령어 위치 : `/bin/chown`
- 명령어 사용 방법 : `# chown <option> <계정명>:<그룹명> <Filename>`
- option
 - `-R` : 하위 디렉토리의 파일들 까지 소유자와 소유 그룹을 변경
- 사용 예
 - `[root@rocky ~]# chown user01:student01 test`
→ test 파일의 소유자와 소유 그룹을 각각 'user01', 'student01'로 변경
 - `[root@rocky /]# chown user02 test`
→ test 파일의 소유자를 'user02'로 변경

(3) `chgrp`

- 명령어 위치 : `/bin/chgrp`
- 명령어 사용 방법 : `# chgrp <option> <그룹명> <Filename>`
- option
 - `-R` : 하위 디렉토리의 파일들 까지 소유 그룹을 변경
- 사용 예
 - `[root@rocky ~]# chgrp student01 test`
→ test 파일의 소유 그룹을 'student01'로 변경

III. 파일의 퍼미션 변경

(1) 퍼미션 변경

- 파일의 접근시 소유자, 소유 그룹, 기타 사용자에게 대한 권한을 변경하는 작업으로 루트(root) 사용자 또는 파일의 소유주만이 변경이 가능
- 퍼미션을 변경하는 명령어로 `chmod`를 사용
- 퍼미션 변경의 방법으로는 상대 모드(Symbolic Method), 절대 모드(Absolute Method) 두 가지가 존재

(2) 상대 모드(Symbolic Method)

- 파일과 디렉토리의 퍼미션 변경에 있어 Symbolic(기호)을 사용

기호	의미	기호	의미	기호	의미
u	user	+	추가	r	읽기
g	group	-	삭제	w	쓰기
o	other	=	지정	x	실행
a	all				

- 사용 예

```

• [root@rocky ~]# ls -l /dir1/test1
-rw-r--r-- 1 root root 156 12월 8 12:47 /dir1/test1
[root@rocky ~]# chmod u+x /dir1/test1
→ test1 파일의 소유자 퍼미션에 'x' 권한 추가
[root@rocky ~]# ls -l /dir1/test1
-rwxr--r-- 1 root root 156 12월 8 12:47 /dir1/test1

```

(3) 절대 모드(Absolute Method)

- 파일과 디렉토리의 허가권 변경에 있어서 Octal(8진수)을 사용
- 읽기(r) : 4, 쓰기(w) : 2, 실행(x) : 1
- 100단위 : 소유자, 10단위 : 소유 그룹, 1단위 : 기타 사용자

Octal(8진) 값	파일 퍼미션	의미	Octal(8진) 값	파일 퍼미션	의미
7(4+2+1)	rwx	읽기,쓰기,실행	6(4+2)	rw-	읽기,쓰기
5(4+1)	r-x	읽기,실행	4	r--	읽기
3(2+1)	-wx	쓰기,실행	2	-w-	쓰기
1	--x	실행	0	---	권한 없음

- 사용 예

```

• [root@rocky ~]# ls -l /dir1/test2
-rw-r--r-- 1 root root 156 12월 8 12:47 /dir1/test2
[root@rocky ~]# chmod 740 /dir1/test2
→ test2 파일의 소유자, 소유 그룹, 기타 사용자 퍼미션을 변경
[root@rocky ~]# ls -l /dir1/test2
-rwxr----- 1 root root 156 12월 8 12:47 /dir1/test2
→ 소유자 : rwx(7)
→ 소유 그룹 : r--(4)
→ 기타 사용자 : --- (0)

```

IV. 특별한 퍼미션

(1) SUID(4000) 퍼미션

- 파일 및 명령어가 실행되는 동안에는 실행한 사용자의 권한이 아닌 파일 소유자의 권한으로 변경되는 역할을 하는 퍼미션
- 기존의 'r,w,x' 문자 외에 's' 문자를 사용하며 절대 모드에서는 4000으로 표현
- 사용 예
 - [root@rocky ~]# ls -l /usr/bin/passwd

```
-r-s--x--x 1 root root 21200 8월 22 2005 /usr/bin/passwd  
→ 'r-s--x--x' = '(r-x--x--x)+SetUID' = '511+4000'
```

(2) SGID(2000) 퍼미션

- 파일 및 명령어가 실행되는 동안에는 실행한 사용자의 권한이 아닌 파일 소유 그룹의 권한으로 변경되는 역할을 하는 퍼미션
- 디렉토리에도 명시 가능
- 기존의 'r,w,x' 문자 외에 's' 문자를 사용하며 절대 모드에서는 2000으로 표현
- 사용 예
 - [root@rocky ~]# ls -l /dir1/test2
 - rwxr----- 1 root root 156 12월 8 12:47 /dir1/test2
 - [root@rocky ~]# chmod 2750 /dir1/test2
 - 'rwxr-s---' = '(rwxr-x---)+SetGID' = '750+2000'

(3) Sticky Bit(1000) 퍼미션

- 디렉토리에 부여하는 퍼미션
- Sticky Bit가 설정된 디렉토리에서는 모든 사용자가 읽기, 쓰기, 삭제가 가능하지만 파일 삭제는 오로지 디렉토리의 소유자만 가능하도록 하는 퍼미션
- 기존의 'r,w,x' 문자 외에 't' 문자를 사용하며 절대 모드에서는 1000으로 표현
- 사용 예
 - [root@rocky ~]# ls -ld /dir1
 - drwxr-xr-x 1 root root 4096 12월 8 13:11 /dir1
 - [root@rocky ~]# chmod 1755 /dir1
 - 'rwxr-s---' = '(rwxr-x---)+SetGID' = '755+1000'

10.5 Disk Partition

1. Disk Partitioning

- 리눅스 디스크의 파티션을 생성, 수정, 삭제하는 작업
- 리눅스 파티션 작업은 fdisk 유틸리티를 이용
- 디스크 명칭
 - IDE 디스크 인식: /dev/hda, /dev/hdb, /dev/hdc, /dev/hdd
 - SCSI 디스크 인식: /dev/sda, /dev/sdb, /dev/sdc ...
 - IDE 디스크 파티션 명칭(/dev/hda)
 - /dev/hda1, /dev/hda2, /dev/hda3 ...
 - SCSI 디스크 파티션 명칭
 - /dev/sda1, /dev/sda2, /dev/sda3 ...

(1) 파티션

- 하드디스크의 파티션은 MBR(Master Boot Record)에 저장
- fdisk 명령어를 통해 확인 가능

- Primary Partition은 4개까지만 존재(1 ~ 4)
- 파티션을 더 많이 설정해야 하는 경우 확장 파티션(Extended)을 통해 구성 가능

(2) 파티션 확인

- fdisk -l: 인식된 디스크 목록 확인
- SCSI 디스크 1개가 설정(설치할 때 설정한 내용)

```
[root@rocky ~]# fdisk -l

Disk /dev/sda: 42.9 GB, 42949672960 bytes
255 heads, 63 sectors/track, 5221 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

   Device Boot      Start         End      Blocks   Id  System
/dev/sda1  *           1           13       104391    83  Linux
/dev/sda2                14          144      1052257+    83  Linux
/dev/sda3             145          275      1052257+    82  Linux swap
/dev/sda4             276          5221      39728745     5  Extended
/dev/sda5             276          5221      39728713+    83  Linux
```

→ /dev/sda1: Boot 파티션

→ /dev/sda3: Swap 파티션

→ /dev/sda4: 확장 파티션(/dev/sda5를 구성하기 위해 존재)

- SCSI 디스크 추가

```
[root@rocky ~]# fdisk -l

Disk /dev/sda: 42.9 GB, 42949672960 bytes
Units = cylinders of 16065 * 512 = 8225280 bytes

   Device Boot      Start         End      Blocks   Id  System
/dev/sda1  *           1           13       104391    83  Linux
....

Disk /dev/sdb: 8589 MB, 8589934592 bytes
255 heads, 63 sectors/track, 1044 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

Disk /dev/sdb doesn't contain a valid partition table
```

- 추가 된 디스크 파티션 설정(/dev/sdb)

```
[root@rocky ~]# fdisk /dev/sdb

Device contains neither a valid DOS partition table, nor Sun, SGI or OSF disklabel
Building a new DOS disklabel. Changes will remain in memory only,
until you decide to write them. After that, of course, the previous
content won't be recoverable.

The number of cylinders for this disk is set to 1044.
There is nothing wrong with that, but this is larger than 1024,
and could in certain setups cause problems with:
1) software that runs at boot time (e.g., old versions of LILO)
```

```

2) booting and partitioning software from other OSs
   (e.g., DOS FDISK, OS/2 FDISK)
Warning: invalid flag 0x0000 of partition table 4 will be corrected by w(rite)

Command (m for help): m
Command action
  a   toggle a bootable flag
  b   edit bsd disklabel
  c   toggle the dos compatibility flag
  d   delete a partition
  l   list known partition types
  m   print this menu
  n   add a new partition
  o   create a new empty DOS partition table
  p   print the partition table
  q   quit without saving changes
  s   create a new empty Sun disklabel
  t   change a partition's system id
  u   change display/entry units
  v   verify the partition table
  w   write table to disk and exit
  x   extra functionality (experts only)

Command (m for help):

```

- a: 부팅 파티션 설정
- d: 파티션 삭제
- l: 설정 가능한 파티션 타입 확인
- n: 새로운 파티션을 생성
- p: 현재 설정된 파티션 정보 출력
- t: 파티션 타입을 변경
- w: 설정한 파티션 정보를 저장 후 fdisk를 빠져나감

(3) 파티션 설정

- 추가된 SCSI 디스크 파티션 구성

- Data를 저장할 파티션으로 Primary 파티션으로 설정
- 1024MB 크기로 설정

```

[root@rocky ~]# fdisk /dev/sdb

Command (m for help): n
Command action
  e   extended
  p   primary partition (1-4)
p
Partition number (1-4): 1
First cylinder (1-1044, default 1): 1
Last cylinder or +size or +sizeM or +sizeK (1-1044, default 1044): +1024M

```

→ 실행 순서: fdisk 명령어 ⇒ n("new") ⇒ p(primary) ⇒ 파티션 번호: 1
 ⇒ First Cylinder: 1 ⇒ Size 설정: +1024M

- 추가된 SCSI 디스크 파티션 구성
 - Swap 영역으로 Primary 파티션 설정
 - 512MB 크기로 설정

```
Command (m for help): l

 0 Empty                1e Hidden W95 FAT1 75 PC/IX                be Solaris boot
 1 FAT12                 24 NEC DOS                80 Old Minix                bf Solaris
 2 XENIX root            39 Plan 9                  81 Minix / old Lin c1 DRDOS/sec (FAT-
 3 XENIX usr             3c PartitionMagic 82 Linux swap        c4 DRDOS/sec (FAT-
 4 FAT16 <32M           40 Venix 80286            83 Linux                    c6 DRDOS/sec (FAT-
 5 Extended              41 PPC PReP Boot         84 OS/2 hidden C: c7 Syrinx
 6 FAT16                 42 SFS                    85 Linux extended da Non-FS data
 7 HPFS/NTFS             4d QNX4.x                 86 NTFS volume set db CP/M / CTOS / .
 8 AIX                   4e QNX4.x 2nd part 87 NTFS volume set de Dell Utility
 9 AIX bootable          4f QNX4.x 3rd part 8e Linux LVM                df BootIt
 a OS/2 Boot Manag 50 OnTrack DM                93 Amoeba                    e1 DOS access
 b W95 FAT32             51 OnTrack DM6 Aux 94 Amoeba BBT                e3 DOS R/O
 c W95 FAT32 (LBA) 52 CP/M                      9f BSD/OS                    e4 SpeedStor
 e W95 FAT16 (LBA) 53 OnTrack DM6 Aux a0 IBM Thinkpad hi eb BeOS fs
 f W95 Ext'd (LBA) 54 OnTrackDM6                a5 FreeBSD                   ee EFI GPT
10 OPUS                  55 EZ-Drive               a6 OpenBSD                   ef EFI (FAT-12/16/
11 Hidden FAT12          56 Golden Bow             a7 NeXTSTEP                   f0 Linux/PA-RISC b
12 Compaq diagnost 5c Priam Edisk          a8 Darwin UFS                f1 SpeedStor
14 Hidden FAT16 <3 61 SpeedStor                a9 NetBSD                    f4 SpeedStor
16 Hidden FAT16          63 GNU HURD or Sys ab Darwin boot                f2 DOS secondary
.....
```

```
Command (m for help): n
Command action
  e   extended
  p   primary partition (1-4)

p
Partition number (1-4): 2
First cylinder (126-1044, default 126):
Using default value 126
Last cylinder or +size or +sizeM or +sizeK (126-1044, default 1044): +512M
```

```
Command (m for help): t
Partition number (1-4): 2
Hex code (type L to list codes): 82
Changed system type of partition 2 to 82 (Linux swap)
```

```
Command (m for help): p
```

```
Disk /dev/sdb: 8589 MB, 8589934592 bytes
255 heads, 63 sectors/track, 1044 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
```

Device	Boot	Start	End	Blocks	Id	System
/dev/sdb1		1	125	1004031	83	Linux
/dev/sdb2		126	188	506047+	82	Linux swap

→ 실행 순서

1) Partition 생성: fdisk ⇒ n ⇒ p ⇒ 설정

2) Type 변경: Command (m for help): t ⇒ 변경 파티션 선택: 2 ⇒ Hex Code: 82

(4) 파티션 삭제

- 설정된 파티션 삭제

```
Command (m for help): p

Disk /dev/sdb: 8589 MB, 8589934592 bytes
255 heads, 63 sectors/track, 1044 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

   Device Boot      Start         End      Blocks   Id  System
/dev/sdb1            1          125     1004031    83  Linux
/dev/sdb2           126          188     506047+    82  Linux swap

Command (m for help): d
Partition number (1-4): 1

Command (m for help): p

Disk /dev/sdb: 8589 MB, 8589934592 bytes
255 heads, 63 sectors/track, 1044 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

   Device Boot      Start         End      Blocks   Id  System
/dev/sdb2           126          188     506047+    82  Linux swap
```

→ 실행 순서: Command: d ⇒ 삭제할 파티션 선택:1

10.6 File System 구성

1. File System 구성

(1) mkfs

- 디스크 파티션 후 파일시스템을 구성하는 명령어
- ext2, ext3 파일시스템을 구성하는 명령어
- 명령어 사용 방법: mkfs [옵션] 장치 명 ⇒ 기본 파일시스템은 ext2
mkfs.ext3 [옵션] 장치 명
 - -t: File System 지정
 - -c: 파일시스템 생성하기 전 배드블록 확인
 - -n: 슈퍼블록 출력
 - -v: 결과를 자세히 보여줌
- ext2 파일시스템으로 /dev/sdb1 구성

```
[root@localhost /]# mkfs -t ext2 /dev/sdb5
mke2fs 1.39 (29-May-2006)
Filesystem label=
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
122112 inodes, 244140 blocks
12207 blocks (5.00%) reserved for the super user
First data block=0
Maximum filesystem blocks=251658240
8 block groups
32768 blocks per group, 32768 fragments per group
15264 inodes per group
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376

Writing inode tables: done
Writing superblocks and filesystem accounting information: done
```

- ext3 파일시스템으로 /dev/sdb2 구성

```
[root@rocky /]# mkfs.ext3 /dev/sdb2
```

(2) mke2fs => mkfs의 확장버전

- 디스크 파티션 후 파일시스템을 구성하는 명령어
- ext2, ext3, ext4 파일시스템을 구성하는 명령어
- 명령어 사용 방법: mke2fs [Option] 장치명
 - -T: File System 지정
 - -b: Block Size 크기 지정
 - -j: ext2 파일시스템을 ext3 파일시스템으로 변경
- ext2 파일시스템으로 /dev/sdb1 구성

```
[root@rocky /]# mke2fs /dev/sdb1
mke2fs 1.39 (29-May-2006)
Filesystem label=
OS type: Linux
Block size=1024 (log=0)
Fragment size=1024 (log=0)
50200 inodes, 200780 blocks
10039 blocks (5.00%) reserved for the super user
First data block=1
Maximum filesystem blocks=67371008
```

- ext3 파일시스템으로 /dev/sdb2 구성

```
[root@rocky /]# mke2fs -j /dev/sdb2
mke2fs 1.39 (29-May-2009)
Filesystem label=
OS type: Linux
Block size=1024 (log=0)
Fragment size=1024 (log=0)
26208 inodes, 104420 blocks
5221 blocks (5.00%) reserved for the super user
First data block=1
Maximum filesystem blocks=67371008
13 block groups
8192 blocks per group, 8192 fragments per group
2016 inodes per group
Superblock backups stored on blocks:
    8193, 24577, 40961, 57345, 73729

Writing inode tables: done
Creating journal (4096 blocks): done
Writing superblocks and filesystem accounting information: done
```

(3) tune2fs

- 파일시스템의 파라미터를 변경한다.
- 명령어 사용 방법: tune2fs [옵션] 장치명
- [옵션]
 - j : journal 사용. ext3으로 변경
 - l : 해당 장치의 슈퍼블록 정보 출력
 - c : 파일시스템 최대 마운트횟수 지정.
 - r : 블록개수 직접 지정하여 예비블록 변경하기
 - m : 파일시스템 전체크기 대비 %로 지정하여 예비블록 변경하기(-m)
 - L : 파일시스템의 레이블명 설정

- ext2 파일시스템으로 구성된 장치(/dev/sdb1)를 ext3 파일시스템으로 변경

```
[root@localhost /]# tune2fs -j /dev/sdb5
tune2fs 1.39 (29-May-2006)
Creating journal inode: done
This filesystem will be automatically checked every 22 mounts or
180 days, whichever comes first. Use tune2fs -c or -i to override.
```

(4) dumpe2fs

- ext2, ext3 파일시스템의 슈퍼블록 정보, 블록그룹 정보를 확인하는 명령어
- 명령어 사용 방법: tune2fs [옵션] 장치명
- [옵션]
 - h : 블록그룹정보를 제외한 슈퍼블록정보만을 출력

(5) fsck

- 리눅스 파일시스템의 오류를 점검하고 복구한다.

(6) e2fsck

- 리눅스 파일시스템을 검사하는데 사용되는 fsck의 확장 명령
- 명령어 사용 방법: e2fsck [옵션] 장치명

[옵션]

- b : 백업수퍼블록을 이용한 파일시스템 복구
- c : 파일시스템 점검과정에서 배드블록 발견시 마킹

(7) df [옵션] [파일]

- 지정한 파일이 있는 경우 해당 파일이 속한 파일시스템의 사용 가능한 공간을 출력

[옵션]

- h : 크기 단위를 KB(1024Bytes=1KB)로 출력.
- T : 파일시스템 타입을 포함하여 출력.
- t type : 특정 파일시스템만 출력.

(8) du [옵션] 파일

- 파일에 사용되는 블록 수를 출력. 지정된 파일 매개변수가 디렉토리이면, 그 안의 모든 파일에 대한 블록 수를 출력

[옵션]

- h : 크기 단위를 KB(1024Bytes=1KB)로 출력.
- s : 지정된 파일의 총 디스크 사용량을 나타내거나 디렉토리에 있는 모든 파일의 총 디스크 사용량을 출력.

10.7 File System Mount

1. File System Mount

(1) mount

- 디스크에 데이터를 저장하기 위해 물리적인 디스크와 논리적인 디렉토리를 연결
- 파일시스템 구성 후 mount 명령어를 통해 연결
- 최상위 디렉토리인 루트("/") 하위에 마운트 포인트(Mount Point)가 위치
- 마운트 포인트로 사용되는 디렉토리는 빈 디렉토리를 이용하는 것이 좋음
- 마운트 정보를 담고 있는 마운트 테이블 파일: /etc/mtab
- 명령어 사용법: mount [Option] [물리적 디스크] [논리적인 디렉토리]
 - 옵션 없이 사용할 경우 시스템의 마운트 정보 확인 가능

```
[root@rocky /]# mount
/dev/sda5 on / type ext3 (rw)
none on /proc type proc (rw)
none on /sys type sysfs (rw)
none on /dev/pts type devpts (rw,gid=5,mode=620)
/dev/sda1 on /boot type ext3 (rw)
...
```

(2) /etc/fstab

- 리눅스 부팅 시 마운트할 파일시스템 정보
- 관리자가 수정을 통해 부팅 시 마운트하고자하는 파일시스템 정보 포함 가능.


```
[root@localhost /]# cat /etc/fstab
LABEL=/ / ext3 defaults 1 1
LABEL=/boot /boot ext3 defaults 1 2
tmpfs /dev/shm tmpfs defaults 0 0
devpts /dev/pts devpts gid=5,mode=620 0 0
sysfs /sys sysfs defaults 0 0
proc /proc proc defaults 0 0
LABEL=SWAP-sda2 swap swap defaults 0 0
```

- 첫 번째 필드: 장치 명 지정
- 두 번째 필드: 마운트 포인트 지정
- 세 번째 필드: 파일시스템 설정
- 네 번째 필드: 마운트 옵션 지정
- 다섯 번째 필드: dump 명령어를 통해 파일시스템 덤프 여부 판별, "0"일 경우 덤프하지 않고, "1"일 경우 파일시스템 덤프
- 여섯 번째 필드: fsck 명령어를 통해 무결성 확인, "0"일 경우 체크 안하고, "1" 이상의 정수를 설정 할 경우 순서가 작은 순서대로 체크

(3) 파일시스템 마운트

- 명령어 사용 방법: `mount -t [파일시스템] -o [옵션] [물리 장치] [논리 장치] [옵션]`
 - `async`: 마운트 된 파일시스템에 비동기 입출력(asynchronous I/O) 사용
 - `nosuid`: 실행파일에 존재하는 Setuid, Setgid 비트의 실행 기능 제한
 - `nouser`: Root 사용자 이외의 사용자가 파일시스템을 마운트하거나 언마운트 기능 제한
 - `ro`: 읽기 전용 파일시스템
 - `rw`: 읽기와 쓰기 가능한 파일시스템
 - `suid`: 실행파일에 존재하는 Setuid, Setgid 비트의 실행 가능
- `/dev/sdb1` 장치를 `/data` 디렉토리에 마운트(옵션 설정 하지 않음)
- `-a` 옵션은 `/etc/fstab`의 설정대로 모두 마운트 함.

```
[root@rocky ~]# mkdir /data
[root@rocky ~]# mount -t ext3 /dev/sdb1 /data
[root@rocky ~]# df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/sda5       38G   3.0G   33G   9% /
/dev/sda1       99M   12M   82M  13% /boot
none            125M    0 125M   0% /dev/shm
/dev/sda2      1012M   34M  927M   4% /home
/dev/hdc        632M   632M    0 100% /media/cdrom
/dev/sdb1       190M   5.6M  175M   4% /data
[root@rocky ~]# mount | grep /data
/dev/sdb1 on /data type ext3 (rw)
```

- 이미 마운트 된 `/dev/sdb1`를 옵션 변경 후 다시 마운트(`ro`, `nosuid`)

```
[root@rocky ~]# mount -t ext3 -o ro,nosuid,remount /dev/sdb1 /data
[root@rocky ~]# mount | grep /data
/dev/sdb1 on /data type ext3 (ro,nosuid)
```

- 플로피 디스크 마운트
 - 플로피 디스크 장치 명: `/dev/fd0`

```
[root@rocky ~]# mount /dev/fd0 /media/floppy/
mount: block device /dev/cdrom is write-protected, mounting read-only
```

- CD-ROM 마운트

- CD-ROM 장치 명: /dev/cdrom, /dev/hdc 등 다양함 (/dev 디렉토리에서 cdrom 검색)

```
[root@rocky ~]# mount -t iso9660 /dev/cdrom /media/cdrom/
mount: block device /dev/cdrom is write-protected, mounting read-only
```

(4) 파일시스템 언마운트

- 명령어 사용 방법: umount [마운트 포인트] or [장치명]

```
[root@rocky /]# mount /dev/sdb1 /data
[root@rocky /]# df -h | grep /data
/dev/sdb1          190M  5.6M  175M   4% /data
[root@rocky /]# umount /data
[root@rocky /]# df -h | grep /data
```

- -a 옵션은 /etc/mtab을 기준으로 마운트 된 모든 파티션을 언마운트 함.
(단, '/' 와 시스템에서 사용하는 파티션 제외)

(5) Swap 파티션 생성

- 명령어 사용 방법: mkswap

```
[root@rocky /]# mkswap /dev/sdb2
Setting up swap space version 1, size = 106921 kB
[root@rocky /]# swapon /dev/sdb2
[root@rocky /]# more /proc/swaps
Filename                                Type              Size      Used      Priority
/dev/sda3                              partition         1052248    0         -1
/dev/sdb2                              partition         104412     0         -2
```

- swap 파일 생성

```
# dd if=/dev/zero of=swapfile bs=1024 count=8192
# mkswap swapfile 8192
# sync
# swapon swapfile
```

- 스왑 파일로 사용할 파일은 어떠한 공백이 있어도 안됨.
따라서, cp 명령으로 스왑 파일을 만들면 안됨.

```
[root@rocky /]# dd if=/dev/zero of=/swap-file bs=1M count=1024
1024+0개의 레코드를 입력하였습니다
1024+0개의 레코드를 출력하였습니다
[root@rocky /]# mkswap /swap-file
Setting up swap space version 1, size = 1073737 kB
[root@rocky /]# swapon /swap-file
[root@rocky /]# more /proc/swaps
Filename                                Type              Size      Used      Priority
/dev/sda3                              partition         1052248    208       -1
/dev/sdb2                              partition         104412     0         -2
/swap-file                             file              1048568    0         -3
```

- swapoff: swapon으로 붙인 swap-file System 삭제

10.8 LVM(Logical Volume Manager)

I. LVM

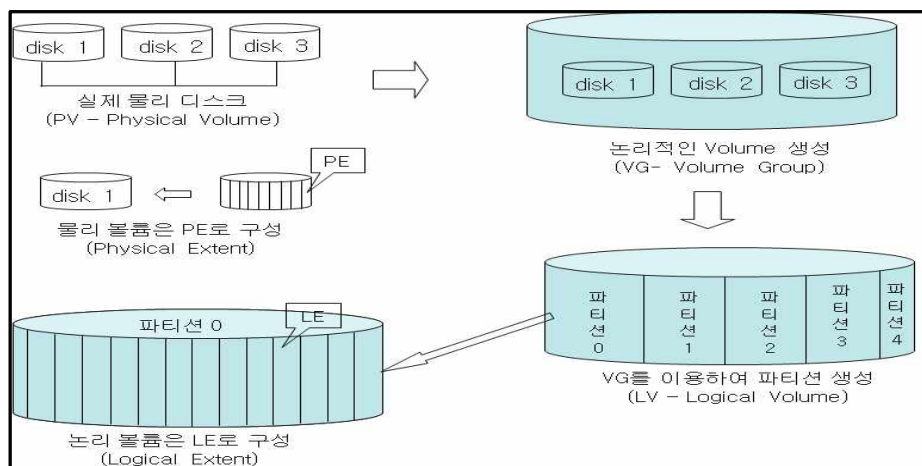
(1) 하드디스크를 좀 더 효율적이고, 유연하게 관리할 수 있는 프로그램

- 손쉬운 관리 및 뛰어난 확장성, 안정성, 효율성
- 스냅샷 기능 지원
- 서로 다른 종류의 많은 디바이스의 조합 지원

(2) LVM 용어

- PV(Physical Volume): 하드디스크의 블록 디바이스인 /dev/sda1, /dev/sdb1과 같은 파티션
- VG(Volume Group): LVM의 가장 기본적인 요소로 PV의 그룹을 의미
- LV(Logical Volume): 논리적 볼륨으로 VG를 나누어 생성된 파티션을 의미
- PE(Physical Extent): 물리적 확장으로 물리적 할당 단위를 의미, PV는 일정한 PE를 가짐
- LE(Logical Extent): 논리적 확장으로 논리적 할당 단위를 의미, PE가 LV에 할당되면 LE가 됨

[그림 7.1] LVM



(3) LVM 구성 전 준비

- 1G 하드디스크 3개 추가
- LVM 패키지 확인

```
[root@rocky system]# lvm version
LVM version:      2.03.14(2)-RHEL8 (2021-10-20)
Library version:  1.02.181-RHEL8 (2021-10-20)
Driver version:   4.46.0
```

- 1G 하드디스크 파티션 후 LVM 형식으로 저장(/dev/sdc1, /dev/sdd1, /dev/sde1)

```

Command (m for help): n
Command action
  e   extended
  p   primary partition (1-4)
p
Partition number (1-4): 1
First cylinder (1-130, default 1):
Using default value 1
Last cylinder or +size or +sizeM or +sizeK (1-130, default 130):
Using default value 130

Command (m for help): t
Selected partition 1
Hex code (type L to list codes): 8e
Changed system type of partition 1 to 8e (Linux LVM)

Command (m for help): w
The partition table has been altered!

Calling ioctl() to re-read partition table.

```

(4) LVM 구성 I - PV 생성

- PV 생성, PV 생성하기 위해 pvcreate 명령어 사용
- 명령어 사용 방법: pvcreate 장치명

```

[root@rocky ~]# pvcreate /dev/sdc1
Physical volume "/dev/sdc1" successfully created
[root@rocky ~]# pvcreate /dev/sdd1
Physical volume "/dev/sdd1" successfully created
[root@rocky ~]# pvcreate /dev/sde1
Physical volume "/dev/sde1" successfully created

```

- PV 정보 확인(pvdisplay)

```

[root@localhost ~]# pvdisplay
"/dev/sdc1" is a new physical volume of "1019.72 MB"
--- NEW Physical volume ---
PV Name                /dev/sdc1
VG Name
PV Size                 1019.72 MB
Allocatable             NO
PE Size (KByte)         0
Total PE                0
Free PE                 0
Allocated PE            0
PV UUID                 JQpFPs-0niZ-e422-jqNG-G4Af-6UR4-M2r9m2
....

```

(5) LVM 구성 II - VG 생성

- VG 그룹 생성, VG 생성하기 위해 vgcreate 명령어 사용
- 명령어 사용 방법: vgcreate [VG 그룹명] [장치 1] [장치 2]...

```

[root@localhost ~]# vgcreate VG1 /dev/sdc1 /dev/sdd1 /dev/sde1
Volume group "VG1" successfully created

```

- VG 정보 확인

- 명령어 사용 방법: `vgdisplay -v [VG 그룹 명]`

```
[root@rocky ~]# vgdisplay -v VG1
Using volume group(s) on command line
Finding volume group "VG1"
--- Volume group ---
VG Name          VG1
System ID
Format           lvm2
Metadata Areas   3
Metadata Sequence No 1
...
```

(6) LVM 구성 III - LV 생성

- 실제로 사용될 LV 생성, LV를 사용하기 위해 `lvcreate` 명령어 사용
- 명령어 사용 방법: `lvcreate -L [Size] -n [LV 명] VG`
`lvcreate -l [PE 개수] VG -n [LV 명]`

```
[root@localhost ~]# lvcreate -L 2G -n hongbo VG1
Logical volume "hongbo" created
```

- VG1(3G)에서 2G의 용량으로 hongbo 이름의 LV 구성

- LV 정보 확인(`lvdisplay`)

```
[root@localhost ~]# lvdisplay
--- Logical volume ---
LV Name          /dev/VG1/hongbo
VG Name          VG1
LV UUID          16MUBx-siqX-flxe-pcYk-E47O-T4CS-92odry
LV Write Access   read/write
LV Status         available
# open           0
LV Size          2.00 GB
Current LE        512
Segments          3
Allocation        inherit
Read ahead sectors    auto
- currently set to  256
Block device      253:0
```

(7) LVM 구성 IV - 파일시스템 구성 및 Mount

- 구성된 LV에 파일 시스템 구성, /hongbo 디렉토리와 Mount

```
[root@rocky /]# mkfs.ext3 /dev/VG1/hongbo
mke2fs 1.39 (29-May-2006)
Filesystem label=
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
262144 inodes, 524288 blocks
26214 blocks (5.00%) reserved for the super user
First data block=0
Maximum filesystem blocks=536870912
...
[root@rocky /]# mkdir /hongbo
[root@rocky /]# mount /dev/VG1/hongbo /hongbo/
[root@rocky /]# df -h
[root@rocky /]# df -h
```

Filesystem	Size	Used	Avail	Use%	Mounted on
/dev/sda3	19G	3.8G	14G	22%	/
/dev/sda1	190M	12M	169M	7%	/boot
tmpfs	252M	0	252M	0%	/dev/shm
/dev/mapper/VG1-hongbo	2.0G	68M	1.9G	4%	/hongbo

(8) LVM 구성 V - LVM 상태 체크

- LVM의 구성 요소 인 PV, VG, LV 확인
- 명령어 사용 방법: pvscan

```
[root@rocky /]# pvscan
PV /dev/sdc1   VG VG1   lvm2 [1016.00 MB / 0   free]
PV /dev/sdd1   VG VG1   lvm2 [1016.00 MB / 0   free]
PV /dev/sde1   VG VG1   lvm2 [1016.00 MB / 1000.00 MB free]
Total: 3 [2.98 GB] / in use: 3 [2.98 GB] / in no VG: 0 [0   ]
```

(9) LVM 구성 VI - VG 체크, LV 체크

- VG 체크: vgscan

```
[root@rocky /]# vgscan
Reading all physical volumes. This may take a while...
Found volume group "VG1" using metadata type lvm2
```

- LV 체크: lvscan

```
[root@rocky /]# lvscan
ACTIVE                  '/dev/VG1/hongbo' [2.00 GB] inherit
```

(10) LVM 구성 VII - LV 용량 증가

- LV로 구성한 디스크에 용량이 부족할 경우 용량 증가, lvextend 명령어를 사용
- 명령어 사용 방법: lvextend -L+[용량] [LV 명]

```
[root@localhost /]# lvextend -L+0.9G /dev/VG1/hongbo
Rounding up size to full physical extent 924.00 MB
Extending logical volume hongbo to 2.90 GB
Logical volume hongbo successfully resized
```

- 용량 확장 후 사용

```
[root@rocky /]# cat > /hongbo/ls.txt
test
[root@rocky /]# umount hongbo
[root@rocky /]# e2fsck -f /dev/VG1/hongbo
e2fsck 1.39 (29-May-2006)
Pass 1: Checking inodes, blocks, and sizes
Pass 2: Checking directory structure
Pass 3: Checking directory connectivity
Pass 4: Checking reference counts
Pass 5: Checking group summary information
/dev/VG1/hongbo: 12/262144 files (8.3% non-contiguous), 25407/524288 blocks

[root@rocky /]# resize2fs /dev/VG1/hongbo5G
resize2fs 1.39 (29-May-2006)
Resizing the filesystem on /dev/VG1/hongbo to 760832 (4k) blocks.
The filesystem on /dev/VG1/hongbo is now 760832 blocks long.

[root@rocky /]# mount /dev/VG1/hongbo5G /hongbo
[root@rocky /]# cd hongbo
[root@rocky hongbo]# ls
lost+found  ls.txt
[root@rocky /]# df -h | grep hongbo
/dev/mapper/VG1-hongbo      2.90G   68M   2.7G   3% /hongbo
```

- resize2fs 명령어 사용 시 umount 해야 함

(11) LVM 구성 VIII - LV 용량 감소

- LV로 구성한 디스크에 용량이 남을 경우, lvreduce 명령어를 사용
- 명령어 사용 방법: lvreduce -L[용량] [LV 명]

```
[root@rocky /]# lvreduce -L-1G /dev/VG1/hongbo
WARNING: Reducing active and open logical volume to 1.90 GB
THIS MAY DESTROY YOUR DATA (filesystem etc.)
Do you really want to reduce hongbo? [y/n]: y
Reducing logical volume hongbo to 1.90 GB
Logical volume hongbo successfully resized
```

(12) LVM 구성 IX - LVM 삭제

- LV 삭제
- 명령어 사용 방법: lvremove [LV 명]

```
[root@rocky /]# lvremove /dev/VG1/hongbo
Do you really want to remove active logical volume "hongbo"? [y/n]: y
Logical volume "hongbo5G" successfully removed
```

- VG 삭제
- 명령어 사용 방법: vgremove [VG 명]
 - 삭제하기 전 VG를 비활성화

```
[root@rocky /]# vgchange -a n VG1
0 logical volume(s) in volume group "VG1" now active
```

```
[root@rocky /]# vgremove /dev/VG1
Volume group "VG1" successfully removed
```

(13) LVM 구성 X - VG에 PV 추가

- 1G 하드디스크 추가(Linux LVM)

```
[root@rocky /]# fdisk /dev/sdf

[root@rocky /]# pvcreate /dev/sdf1
```

- VG에 PV 추가

- 명령어 사용 방법: `vgextend [VG 명] [추가할 PV 명]`

```
[root@rocky /]# vgextend VG1 /dev/sdf1
Volume group "VG1" successfully extended
[root@rocky /]# vgdisplay
[root@rocky /]# pvdisplay
```

- 실습

1. LVM에 사용할 파티션 생성

파티션을 생성하고 system-id 타입을 Linux LVM(8e)으로 변경. 기본은 Linux

2. PV(Physical Volume) 생성

`pvcreate 파티션1 파티션2 파티션3`

`pvcreate /dev/sdd1 /dev/sde1 /dev/sdf1`

pvs로 확인

3. VG(Volume Group) 생성

`vgcreate VG명 PV1 PV2 PV3`

`vgcreate gdn14 /dev/sdd1 /dev/sde1 /dev/sdf1`

vgs로 확인

4. LV(Logical Volume) 생성

`lvcreate -L용량 -n LV명 VG명`
-l 100%FREE => VG의 남은 용량 모두 할당하는 옵션

`lvcreate -L4G -n network gdn14`

`lvcreate -l 100%FREE -n server gdn14`

5. VG 확장 : VG의 공간이 부족할 경우 pv를 추가하여 확장한다.

```
vgextend gdn14 /dev/sdf1
```

6. LV 확장

```
lvextend -L+용량 VG명/LV명
```

-l +100%FREE => 남은 공간을 모두 확장에 사용하는 옵션

```
lvextend -L+1G gdn8/network
```

```
lvextend -l +100%FREE gdn15/network
```

확장 후에 `resize2fs /dev/gdn15/network` 명령 실행

파일 시스템이 xfs 일 경우 `xfs_growfs /dev/gdn15/network` 명령 실행

7. lvreduce, vgreduce 등 축소할 수 있는 명령은 일반적으로 사용하지 않음.

8. PV, VG, LV를 삭제하는 순서는 만들때와 반대.

```
lvremove gdn14/network gdn14/server
```

```
vgremove gdn14
```

```
pvremove /dev/sdd1 /dev/sde1 /dev/sdf1
```

- sdd1, sde1, sdf1 을 pv로 생성

- pv를 묶어서 vg(cert) 생성

- lv(security) - 8G : ext3 => mp3

lv(monitor) - 나머지 전체 : ext4 => mp4

fstab에 등록 옵션 defaults dump(0) fsck(0)

```
mkfs.ext3 /dev/cert/security
```

```
mkfs -t ext4 /dev/cert/monitor
```

```
umount -a
```

```
mount -a
```

10.9 RAID(Redundant Array Independant Disk)

I . RAID란

- 여러개의 디스크를 하나의 디스크처럼 사용하도록 함으로써, 물리적으로는 작은 용량의 디스크이지만 논리적으로는 보다 큰 디스크처럼 사용할 수도 있고, 같은 내용을 여러군데 저장함으로써 장애에 대비할 수 있는 기술이다.

II . RAID 레벨

RAID 0(디스크 스트라이핑)

- 최소 드라이브 개수 : 2
- 최대 용량 : 디스크의 수 x 디스크의 용량
- 설명 : 데이터를 블록으로 쪼개서 저장하는데 각 블록은 다른 디스크로 나뉘어 저장된다.
- 장점 : 매우 빠르다.
- 단점 : 드라이브 하나가 고장 나면 복구가 불가능하다.

RAID 1 (디스크 미러링)

- 최소 드라이브 개수 : 2
- 최대 용량 : (디스크의 수/2) x 디스크의 용량
- 설명 : 스토리지에 저장되는 모든 데이터는 두 개의 물리적인 디스크에 각각 저장되고 모든 데이터는 중복된다.
- 장점 : 드라이브 하나가 고장 나도 복구가 가능하다.
- 단점 : 각 드라이브는 미러링되기 때문에 전체 용량의 절반밖에 사용하지 못한다.
드라이브 두 개에 동일한 데이터를 써야 하기 때문에 쓰기 성능이 떨어진다.

RAID 5(복구를 위해 패리티 저장)

- 최소 드라이브 개수 : 3
- 최대 용량 : (디스크의 수 - 1) x 디스크의 용량
- 설명 : 데이터의 블록은 모든 디스크에 나뉘어 저장되지만 항상 균등하진 않고 패리티 정보도 모든 디스크에 나뉘어 저장된다.
- 장점 : 한 개의 드라이브가 고장 나는 것을 허용한다.
- 단점 : 디스크 재구성(rebuild)이 매우 느리고 쓰기 성능은 패리티 정보를 끊임없이 갱신해야 하기 때문에 우수하다고 할 수는 없다.

RAID 10 (디스크 미러링 + 디스크 스트라이핑)

- 최소 드라이브 개수 : 4
- 최대 용량 : (디스크의 수/2) x 디스크의 용량
- 설명 : RAID1과 RAID0의 장점을 가진 레벨.
- 장점 : 어떤 디스크가 고장나도 해당 디스크만 교체하면 된다.
- 단점 : 비용이 많이 든다.

III . RAID구성을 위한 파티션 타입 변경하기

- 디스크를 추가하고, 해당 파티션 타입을 raid로 설정한다.

```
# fdisk /dev/sdb
```

```
The number of cylinders for this disk is set to 121601.
```

```
There is nothing wrong with that, but this is larger than 1024,  
and could in certain setups cause problems with:
```

- 1) software that runs at boot time (e.g., old versions of LILO)
- 2) booting and partitioning software from other OSs
(e.g., DOS FDISK, OS/2 FDISK)

```
Command (m for help): p
```

```
Disk /dev/sdb: 1000.2 GB, 1000204886016 bytes
```

```
255 heads, 63 sectors/track, 121601 cylinders
```

```
Units = cylinders of 16065 * 512 = 8225280 bytes
```

Device	Boot	Start	End	Blocks	Id	System
--------	------	-------	-----	--------	----	--------

```
Command (m for help): n
```

```
Command action
```

```
  e   extended
```

```
  p   primary partition (1-4)
```

```
p
```

```
Partition number (1-4): 1
```

```
First cylinder (1-121601, default 1):
```

```
Using default value 1
```

```
Last cylinder or +size or +sizeM or +sizeK (1-121601, default 121601):
```

```
Using default value 121601
```

```
Command (m for help): t
```

```
Selected partition 1
```

```
Hex code (type L to list codes): l
```

0	Empty	1e	Hidden W95 FAT1	80	Old Minix	bf	Solaris
1	FAT12	24	NEC DOS	81	Minix / old Lin	c1	DRDOS/sec (FAT-
2	XENIX root	39	Plan 9	82	Linux swap / So	c4	DRDOS/sec (FAT-
17	Hidden HPFS/NTF	64	Novell Netware	b7	BSDI fs	fc	VMware VMKCORE
18	AST SmartSleep	65	Novell Netware	b8	BSDI swap	fd	Linux raid auto
1b	Hidden W95 FAT3	70	DiskSecure Mult	bb	Boot Wizard hid	fe	LANstep
1c	Hidden W95 FAT3	75	PC/IX	be	Solaris boot	ff	BBT

```
Hex code (type L to list codes): fd
```

```
Changed system type of partition 1 to fd (Linux raid autodetect)
```

```
Command (m for help): p
```

```
Disk /dev/sdb: 1000.2 GB, 1000204886016 bytes
255 heads, 63 sectors/track, 121601 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
```

Device	Boot	Start	End	Blocks	Id	System
/dev/sdb1		1	121601	976760001	fd	Linux raid autodetect

/dev/sdc1 도 동일하게 진행합니다.

Device	Boot	Start	End	Blocks	Id	System
/dev/sdc1		1	121601	976760001	fd	Linux raid autodetect

IV. mknod 명령을 이용하여 장치 추가하기

- 장치의 주변호와 부번호를 먼저 알아야함.
more /proc/devices => 장치의 주변호 확인
ls -l /dev => 장치의 주변호와 부번호 확인
- 기본적으로 md0는 생성되어 있음.
md1을 추가할 경우
mknod /dev/md1 b 9 1
- 장치 삭제
rm /dev/md1

V. mdadm 명령어를 이용해 raid1 구성하기.

- ```
[root@rocky ~]# mdadm --version
mdadm - v4.2 - 2021-12-30 - 5
```
- RAID 장치 생성  
# mdadm --create --verbose /dev/md1 --level=1 --raid-devices=2 /dev/sdb1 /dev/sdc1  
mdadm: size set to 976759936K  
mdadm: array /dev/md1 started.
  - 구성과정 확인  
# watch -n1 'cat /proc/mdstat' => 명령을 주기적으로 실행하기. 보기는 1초마다 cat 명령 실행  
  
# cat /proc/mdstat  
Personalities : [raid1]  
md1 : active raid1 sdc1[1] sdb1[0]  
976759936 blocks [2/2] [UU]  
[>.....] resync = 0.2% (2698112/976759936) finish=270.3min  
speed=60042K/sec  
unused devices: <none>

---

```

- /etc/mdadm.conf 설정하기
 # echo "DEVICES /dev/sdb1 /dev/sdc1" > /etc/mdadm.conf
 # mdadm --detail --scan >> /etc/mdadm.conf
 # mdadm --detail --scan
 ARRAY /dev/md1 level=raid1 num-devices=2 metadata=0.90 UUID=xxx

- 완료 후, 마운트하기 위해 mkfs -j /dev/md1
 # mkdir /data
 # mount /dev/md1 /data

- fstab 설정하기
 /dev/md1 /mnt ext3 defaults 1 2 ==> 추가

- md1 장치 확인
 # mdadm --detail /dev/md1
 or
 # mdadm --query --detail /dev/md1

- 중지
 # mdadm --stop /dev/md1

- 다시 시작하기
 # mdadm --assemble /dev/md1 /dev/sdc1 /dev/sdd1
 # mdadm -A -s => /etc/mdadm.conf에 설정된 내용으로 raid 시작

```

## VI. 장애가 발생한 디스크 교체하기

```

- RAID 장치 상태 확인
 # cat /proc/mdstat
 Personalities : [raid1]
 md1 : active raid1 sdb1[2](F) sdc1[1]
 1044096 blocks [2/1] [_U] ==> _U로 표시됨.
 unused devices: <none>

 # mdadm --query --detail /dev/md1

- RAID 장치 제거
 # mdadm /dev/md1 --remove /dev/sdb1
 => mdadm /dev/md1 -f /dev/sdb1과 같이 해당 파티션을 -f(--fail) 옵션과 함께 문제가 있는
 것으로 표시해야 제거할 수 있음.

- 새로운 하드(sdd)로 교체
 # mdadm /dev/md1 --add /dev/sdd1

```

---

## Chapter 11. 소프트웨어 관리

---

rocky에서 특정 소프트웨어를 사용하기 위해서는 추가로 설치를 해야한다. 리눅스 소프트웨어는 대부분 tar 라는 도구에 의해 묶여져 배포되는데, 소스코드는 리눅스에서 사용할 수 있도록 컴파일해야 하는 불편한 작업이 따르지만, 사용자가 원하는 옵션을 부여하여 바이너리 파일을 만들어 설치할 수 있다. 특히, 바이너리 패키지가 배포되지 않은 최신 버전의 소프트웨어를 설치하거나 업데이트하는데 유용할 수 있다. RPM 패키지는 소스코드를 RPM 제작자들이 미리 컴파일하여 소프트웨어를 쉽게 설치할 수 있게 해 주는 형태이다. 또 다른 방법은 YUM이라는 도구를 이용하는 것이다. RPM을 이용하여 설치할 때 발생할 수 있는 의존성 문제를 해결할 수 있다.

### 11.1 파일 압축 관리

#### 1. tar(tape archive)

- 하나 또는 여러 개의 파일들을 하나로 묶는 명령어
- 명령어 사용 방법: tar [ Option ] [ 생성할 파일 명 ] [ 파일 ]
  - -c: 정의한 파일 이름으로 파일을 묶음
  - -f: 파일명 혹은 디바이스 명시
  - -t: 리스트 확인
  - -x: 전체 파일을 추출
  - -C: tar 파일이 해제 될 위치 지정
  - -g: --listed-incremental 과 같은 옵션으로 증분 백업 옵션
  - -z: tar.gz 형태로 압축 또는 해제
  - -v: tar.bz2 형태로 압축 또는 해제

```
[rocky@rocky ~]$ touch test1 test2 test3 test4 test5
[rocky@rocky ~]$ ls
test1 test2 test3 test4 test5
[rocky@rocky ~]$ tar cvf test.tar *
test1
test2
test3
test4
test5
[rocky@rocky ~]$ ls -l test.tar
-rw-rw-r-- 1 rocky rocky 10240 2월 1 17:55 test.tar
[rocky@rocky ~]$
```

```
[rocky@rocky ~]$ mkdir tar
[rocky@rocky ~]$ cp test.tar tar
[rocky@rocky ~]$ cd tar
[rocky@rocky tar]$ tar xvf test.tar
test1
test2
test3
test4
test5
[rocky@rocky tar]$ ls
test.tar test1 test2 test3 test4 test5
[rocky@rocky tar]$
```

## II. gzip / gunzip

- 압축하는 명령어로 압축 후엔 .gz 확장자가 생성
- 명령어 사용 방법: gzip [ Option ] [ 파일 명 ]  
gunzip [ 파일 명 ]
- -r: 디렉토리내의 모든 파일과 하위 디렉토리에 존재하는 모든 파일까지 압축
- -d: gzip 명령어에서 사용하는 명령어로 gunzip 명령어와 동일

```
[root@rocky /]# mkdir /gzip
[root@rocky /]# cp -r /bin/* /gzip/
[root@rocky /]# ls /gzip
[root@rocky /]# gzip -r gzip
```

```
[root@rocky /]# cd /gzip
[root@rocky gzip]# ls
alsaunmute.gz dnsdomainname kbd_mode.gz nisdomainname sync.gz
arch.gz doexec.gz keyctl.gz pgawk.gz tar.gz
....
[root@rocky gzip]# gzip -d *.gz
```

## III. bzip2 / bunzip2

- 압축하는 명령어로 압축 후엔 .bz 확장자가 생성
- 명령어 사용 방법: bzip2 [ Option ] [ 파일 명 ]  
bunzip2 [ 파일 명 ]
- -r: 디렉토리내의 모든 파일과 하위 디렉토리에 존재하는 모든 파일까지 압축
- -d: gzip 명령어에서 사용하는 명령어로 gunzip 명령어와 동일

```
[root@rocky /]# ls /gzip
[root@rocky /]# bzip2 /gzip/*
```

```
[root@rocky /]# cd /gzip
[root@rocky gzip]# ls
....
df.bz2 igawk.bz2 netstat.bz2 stty.bz2 zcat.bz2
dmesg.bz2 ipcalc.bz2 nice.bz2 su.bz2 zsh.bz2
[root@rocky gzip]# bunzip2 *
```

## 11.2 소스 코드 컴파일

---

## I . 소스 코드 컴파일

- rpm으로 제공하지 않는 경우 소스 코드를 이용하여 설치

- 컴파일 환경 설정: configure
- 컴파일: make
- 컴파일 된 패키지 설치: make install

### (1) 컴파일 환경 설정

- <http://mirror.apache-kr.org/httpd/httpd-2.4.54.tar.bz2> 다운로드
- # yum install -y zlib-devel openssl-devel libxml2-devel libjpeg-devel libpng-devel
- 컴파일 환경 설정 help: ./configure
- 설치 경로 지정: --prefix

```
./configure --prefix=/usr/local/apache2
```

### (2) 컴파일

- configure 명령어를 통해 컴파일 환경 설정 완료 후 소스 컴파일
- 컴파일을 다시 해야 할 경우 "make clean" 명령어 사용

```
make
make clean
```

### (3) 파일 설치

- 컴파일 후 설치

```
make install
ls /usr/local/apache2
```

## 11.3 RPM

### I . RPM(Redhat Package Manager)

- 프로그램, 어플리케이션들의 설치, 관리, 삭제, 업데이트를 쉽게 하도록 만들

- rpm 명령어를 통해 관리 가능

- 명령어 사용 방법: rpm [ Option ] Package

- -i: 패키지 설치
- -U: 패키지 업그레이드
- -e: 패키지 제거
- -q: 패키지 질의
- -V: 패키지 검증
- -b: 패키지 제작

### (1) 패키지

- 설치될 모든 파일들의 디렉토리 위치를 지정하여 하나로 묶어놓은 것
- RPM 패키지 하나만 설치하면 서비스에 필요한 모든 실행 파일이 정해진 곳에 복사



- 패키지 구조: MySQL-python-1.2.1\_p2-1.el4.1.i386.rpm  
[ MySQL-python ][ 1.2.1\_p2 ][ 1.el4.1 ][ i386 ].rpm  
[ 패키지 이름 ][ 패키지 버전 ][ 릴리즈 버전 ][ 아키텍처 ]

## (2) 패키지 설치

- i 옵션을 사용한 패키지 설치
- v(Verbose), -h: 설치 할 때 "#" 표시
- 이미 설치된 패키지 무시하고 설치할 경우 "--replacepkgs" 옵션 사용
- 다른 패키지의 파일과 중복될 경우에 무시하고 설치할 경우 "--replacefiles" 옵션 사용
- "--force" 옵션을 사용하여 설치할 경우 "--replacepkgs", "--replacefiles"을 동시에 실행하는 효과`
- 의존성 문제 무시하여 설치할 경우 "--nodeps" 옵션 사용

## (3) 패키지 업그레이드 설치

- U 옵션을 이용한 패키지 업그레이드 설치
- 기존 파일이 있을 경우 "파일명.rpm.save"로 저장 후 새롭게 설치

## (4) 패키지 삭제

- e 옵션을 사용한 패키지 삭제
- 명령어 사용 방법: rpm -e [ Options ] 패키지 이름
- 의존성 때문에 삭제되지 않을 경우 "--nodeps" 옵션 사용

```
[root@rocky /]# rpm -qa | grep telnet
telnet-0.17-31.EL4.5
telnet-server-0.17-31.EL4.5
[root@rocky /]# rpm -e telnet-server
[root@rocky /]# rpm -qa | grep telnet
telnet-0.17-31.EL4.5
```

## (5) 패키지 설치 정보 확인

- 명령어 사용 방법: rpm -q [ Options ] 패키지 이름

```
[root@rocky /]# rpm -qa | more
filesystem-2.3.0-1
bzip2-libs-1.0.2-13.EL4.3
hdparm-5.7-2
checkpolicy-1.17.5-1
audit-1.0.15-3.EL4
....
```

## (6) 파일이 패키지에 속한 내용 확인

- 명령어 사용 방법: rpm -qf 파일 이름

```
[root@rocky /]# rpm -qf /bin/ps
procps-ng-3.3.15-9.el8.x86_64
```

## (7) 특정 패키지에 의해 설치된 파일 확인

- 명령어 사용 방법: rpm -ql 패키지 이름

```
[root@rocky /]# rpm -ql procps-3.2.3-8.9
/bin/ps
/lib/libproc-3.2.3.so
/sbin/sysctl
/usr/bin/free
/usr/bin/pgrep
.....
```

#### (8) 패키지 상세 정보 확인

- 패키지 이름, 버전, 개발, Release, 컴파일 날짜, Size, 라이선스 정보 등의 내용 포함
- 명령어 사용 방법: rpm -qi 패키지 명

```
[root@rocky /]# rpm -qi procps-3.2.3-8.9
Name : procps Relocations: (not relocatable)
Version : 3.2.3 Vendor: rocky
Release : 8.9 Build Date:
Install Date: Build Host: builder6
Group : Applications/System Source RPM: procps-3.2.3-8.9.src.rpmSize : 323839
License: GPL
Signature : DSA/SHA1, 2007년 11월 19일 (월) 오전 05시 55분 10초, Key ID a53d0bab443e1821
Summary : System and process monitoring utilities.
Description:
The procps package contains a set of system utilities that provide
system information. Procps includes ps, free, skill, pkill, pgrep,
snice, tload, top, uptime, vmstat, w, and watch. The ps command
.....
```

#### (9) 패키지에 의해 설치된 파일들 확인

- 설치된 파일이 정상 여부 확인
- 명령어 사용 방법: rpm -qs 패키지 명

```
[root@rocky /]# rpm -qs httpd
정상(normal) /etc/httpd
정상(normal) /etc/httpd/conf
정상(normal) /etc/httpd/conf.d
정상(normal) /etc/httpd/conf.d/README
정상(normal) /etc/httpd/conf.d/welcome.conf
정상(normal) /etc/httpd/conf/httpd.conf
정상(normal) /etc/httpd/conf/magic
정상(normal) /etc/httpd/logs
정상(normal) /etc/httpd/modules
.....
```

#### (10) 패키지의 설정 파일 확인

- 설치된 패키지의 파일 중 설정 파일 확인
- 명령어 사용 방법: rpm -qc 패키지 명

```
[root@rocky RPMS]# rpm -qc httpd
/etc/httpd/conf.d/welcome.conf
/etc/httpd/conf/httpd.conf
/etc/httpd/conf/magic
/etc/logrotate.d/httpd
/etc/rc.d/init.d/httpd
/etc/sysconfig/httpd
/var/www/error/HTTP_BAD_GATEWAY.html.var
/var/www/error/HTTP_BAD_REQUEST.html.var
/var/www/error/HTTP_FORBIDDEN.html.var
/var/www/error/HTTP_GONE.html.var
/var/www/error/HTTP_INTERNAL_SERVER_ERROR.html.var
/var/www/error/HTTP_LENGTH_REQUIRED.html.var
/var/www/error/HTTP_METHOD_NOT_ALLOWED.html.var
/var/www/error/HTTP_NOT_ACCEPTABLE.html.var
/var/www/error/HTTP_NOT_FOUND.html.var
....
```

#### (1) 패키지의 의존성 확인

- 명령어 사용 방법: rpm -qRp 패키지 명

```
[root@rocky ~]# cd /media/rocky_5.3_Final/rocky/
[root@rocky rocky]# rpm -qRp telnet-server-0.17-31.EL4.5.i386.rpm
경고: telnet-server-0.17-31.EL4.5.i386.rpm: V3 DSA signature: NOKEY, key ID 443e1821
config(telnet-server) = 1:0.17-31.EL4.5
libc.so.6
libc.so.6(GLIBC_2.0)
libc.so.6(GLIBC_2.1)
libc.so.6(GLIBC_2.1.3)
libc.so.6(GLIBC_2.3)
.....
```

## 8.4 YUM(Yellowdog Updater, Modified)

### 1. YUM 설정

- 지정된 서버로부터 업데이트된 패키지들을 검사하여 다운 및 설치
- 설정 파일: /etc/yum.conf

```
[main]
cachedir=/var/cache/yum
keepcache=0
debuglevel=2
logfile=/var/log/yum.log
distroverpkg=redhat-release
tolerant=1
exactarch=1
obsoletes=1
gpgcheck=1
plugins=1
Note: yum-RHN-plugin doesn't honor this.
metadata_expire=1h
installonly_limit = 5
PUT YOUR REPOS HERE OR IN separate files named file.repo
in /etc/yum.repos.d
```

- 업데이트 관련 서버 주소 목록 확인: /etc/yum.repos.d/rocky-Base.repo

#### (1) 업데이트 리스트

- 업데이트 가능한 패키지 확인

```
[root@rocky yum.repos.d]# cd /
[root@rocky /]# yum check-update
Loaded plugins: fastestmirror, security
Loading mirror speeds from cached hostfile
* base: rocky.tt.co.kr
* extras: rocky.tt.co.kr
* updates: rocky.tt.co.kr
Skipping security plugin, no data

crash.x86_64 5.1.8-2.el5.rocky updates
```

#### (2) 업데이트 |

- 업데이트 가능한 모든 패키지 업데이트

```
[root@rocky /]# yum update
Setting up Update Process
Setting up repositories
Reading repository metadata in from local files
Resolving Dependencies
--> Populating transaction set with selected packages. Please wait.
---> Downloading header for libmthca to pack into transaction set.
libmthca-1.0.5-1.el4.i386 100% |=====| 4.2 kB 00:00
---> Package libmthca.i386 0:1.0.5-1.el4 set to be updated
---> Downloading header for ltrace to pack into transaction set.
ltrace-0.4-6.el4.i386.rpm 100% |=====| 7.0 kB 00:00
---> Package ltrace.i386 0:0.4-6.el4 set to be updated
---> Downloading header for gnome-vfs2 to pack into transaction set.
gnome-vfs2-2.8.2-8.9.EL4. 100% |=====| 34 kB 00:00
---> Package gnome-vfs2.i386 0:2.8.2-8.9.EL4 set to be updated
---> Downloading header for gcc-c++ to pack into transaction set.
gcc-c%2B%2B-3.4.6-11.el4_ 64% |=====| 24 kB 00:00 ETA
.....
```

---

### (3) 업데이트 II

- 지정한 패키지 업데이트
- 명령어 사용 방법: `yum -y install` 패키지 명

```
[root@rocky /]# yum --help
usage: yum [options] < update | install | info | remove | list |
 clean | provides | search | check-update | groupinstall |
 groupupdate | grouplist | groupinfo | groupremove |
 makecache | localinstall | erase | upgrade | whatprovides |
 localupdate | resolvedep | shell | deplist >

options:
 -h, --help show this help message and exit
 -t, --tolerant be tolerant of errors
 -C run entirely from cache, don't update cache
```

```
 -R [minutes] maximum command wait time
 -d [debug level] debugging output level
 -e [error level] error output level
 -y answer yes for all questions
 --version show Yum version and exit
 --installroot=[path] set install root
 --enablerepo=[repo] enable one or more repositories (wildcards allowed)
 --disablerepo=[repo] disable one or more repositories (wildcards allowed)
 --exclude=[package] exclude package(s) by name or glob
 --obsoletes enable obsoletes processing during updates
 --noplugins disable Yum plugins

[root@rocky /]# yum list | grep wireshare-gnome
[root@rocky /]# yum -y install wireshare-gnome
```

### (4) 패키지 삭제

- 명령어 사용 방법: `yum -y remove` 패키지 명

## Chapter 12. System Log

### 12.1 Log

#### 1. 로그?

- 서버 운영체제에는 시스템 내에 발생하는 많은 일들에 대해 기록을 남김
- 기록이 남는 곳을 로그 파일이라고 함
- 시스템이 해킹, 시스템 이상이 발생해 정상적인 운영이 어려운 경우 로그 파일이 문제 해결의 단서가 됨
- ID, IP Address, Port 등의 정보를 확인

#### (1) 시스템 로그

- 대부분 시스템의 중요한 사항은 리눅스 시스템에 위치한 로그 파일에 기록
- 로그 파일에서 확인해야 할 사항
  - 짧거나 불완전한 기록
  - 시간 표시가 이상하거나 없어진 기록
  - 잘못된 퍼미션 및 소유권 리스트
  - su 사용 기록 및 의심 접속 확인

#### (2) 로그 파일 위치

| 로 그    | 로그 파일 명                             | 로그 발생 프로그램 |
|--------|-------------------------------------|------------|
| 시스템 로그 | /var/log/messages                   | syslogd    |
| 콘솔 로그  | /dev/console                        | kernel     |
| 보안 로그  | /var/log/secure                     | xinetd     |
| 메일 로그  | /var/log/maillog                    | sendmail   |
| 부트 로그  | /var/log/boot.log                   | kernel     |
| 부트 로그  | /var/log/dmesg                      | kernel     |
| 크론 로그  | /var/log/cron                       | cron       |
| FTP 로그 | /var/log/xferlog                    | ftpd       |
| 웹로그    | /var/log/httpd/access_log,error_log | httpd      |

#### (3) rsyslog

- /etc/rsyslog.conf, syslogd, openlog, logger로 구성
- 사용자 및 관리자의 실수로 발생할 수 있는 데이터 손실 방지
- Selector 구성(Facility.Level)
  - Facility

| Facility | 의 미                              | Facility        | 의 미                         |
|----------|----------------------------------|-----------------|-----------------------------|
| authpriv | 인증 절차에 관련한 메시지                   | auth            | 사용자 인증 관련 메시지               |
| cron     | cron, at 관련 메시지                  | daemon          | 데몬 관련 메시지                   |
| kern     | 커널에 관련한 메시지                      | lpr             | 프린터 관련 메시지                  |
| mark     | timestamp를 생성하기 위한 syslog 내부 메시지 | local0 ~ local7 | 특정 프로그램이 사용하도록 남겨둔 Facility |

---

| Facility | 의 미               | Facility | 의 미                |
|----------|-------------------|----------|--------------------|
| syslog   | syslogd가 생성하는 메시지 | user     | 사용자에 의해 발생하는 메시지   |
| uucp     | uucp가 생성하는 메시지    | news     | 뉴스 데몬에 의해 발생하는 메시지 |
| *        | mark를 제외한 모두      |          |                    |

• Level

| Level   | 의 미                                           |
|---------|-----------------------------------------------|
| emerg   | 패닉(panic) 상황. 로그인한 모든 사용자들에게 로그 메시지를 뿌려준다.    |
| alert   | 즉각적인 조치를 취해야 할 상황                             |
| crit    | 하드웨어 에러(error) 같이 급한 상황은 아니지만 치명적인 시스템 문제 발생. |
| err     | 위 세 가지 레벨을 제외한 에러(error) 의미.                  |
| warning | 단순 경고 메시지.                                    |
| notice  | 에러가 아닌 알림에 관한 메시지.                            |
| info    | 단순한 프로그램에 대한 정보 메시지.                          |
| debug   | 프로그램의 디버깅 모드(Debugging Mode) 메시지              |
| none    | 지정한 facility로 메시지를 보내지 않음.                    |

- 모든 레벨을 적용할 때는 ‘\*’ 를 사용.

• /etc/syslog.conf 설정 파일 확인

```

rsyslog configuration file

For more information see /usr/share/doc/rsyslog-*/rsyslog_conf.html
or latest version online at http://www.rsyslog.com/doc/rsyslog_conf.html
If you experience problems, see http://www.rsyslog.com/doc/troubleshoot.html

MODULES

module(load="imuxsock" # provides support for local system logging (e.g. via
 logger command)
 SysSock.Use="off") # Turn off message reception via local log socket;
 # local messages are retrieved through imjournal now.
module(load="imjournal" # provides access to the systemd journal
 StateFile="imjournal.state") # File to store the position in the journal
#module(load="imklog") # reads kernel messages (the same are read from journald)
#module(load="immark") # provides —MARK— message capability

Provides UDP syslog reception
for parameters see http://www.rsyslog.com/doc/imudp.html
#module(load="imudp") # needs to be done just once
#input(type="imudp" port="514")

Provides TCP syslog reception
for parameters see http://www.rsyslog.com/doc/imtcp.html
#module(load="imtcp") # needs to be done just once
#input(type="imtcp" port="514")

GLOBAL DIRECTIVES

Where to place auxiliary files
global(workDirectory="/var/lib/rsyslog")

Use default timestamp format
module(load="builtin:omfile" Template="RSYSLOG_TraditionalFileFormat")

Include all config files in /etc/rsyslog.d/
include(file="/etc/rsyslog.d/*.conf" mode="optional")

RULES

Log all kernel messages to the console.
Logging much else clutters up the screen.
#kern.* /dev/console

Log anything (except mail) of level info or higher.
Don't log private authentication messages!
*.info:mail.none;authpriv.none;cron.none /var/log/messages

The authpriv file has restricted access.
authpriv.* /var/log/secure

Log all the mail messages in one place.
mail.* -/var/log/maillog

Log cron stuff
cron.* /var/log/cron

Everybody gets emergency messages
.emerg :omusrmsg:

Save news errors of level crit and higher in a special file.
uucp,news.crit /var/log/spooler

Save boot messages also to boot.log
local7.* /var/log/boot.log

```



#### (4) 시스템 로그 확인

##### - /var/log/messages

- 실시간 로그 확인: `tail -f /var/log/messages`

```
[root@rocky httpd]# more /var/log/messages
```

```
Jul 11 18:52:36 ra syslogd 1.4.1: restart.
Jul 11 18:53:09 ra hald[3297]: Timed out waiting for hotplug event 638. Rebasing to 638
Jul 11 18:56:29 ra kernel: SCSI device sdb: 16777216 512-byte hdwr sectors (8590 MB)
Jul 11 18:56:29 ra kernel: sdb: cache data unavailable
```

```
Jul 11 18:56:29 ra kernel: sdb: assuming drive cache: write through
Jul 11 18:56:29 ra kernel: sdb: sdb1 sdb2
Jul 11 18:56:31 ra kernel: SCSI device sdb: 16777216 512-byte hdwr sectors (8590 MB)
Jul 11 18:56:31 ra kernel: sdb: cache data unavailable
Jul 11 18:56:31 ra kernel: sdb: assuming drive cache: write through
Jul 11 18:56:31 ra kernel: sdb: sdb1 sdb2
Jul 12 21:31:12 ra kernel: eeepro100: Unknown parameter `speed_duplex'
Jul 12 21:31:12 ra kernel: eeepro100.c:v1.09j-t 9/29/99 Donald Becker
http://www.scyld.com/network/eeepro100.html
Jul 12 21:31:12 ra kernel: eeepro100.c: $Revision: 1.36 $ 2000/11/17 Modified by
Andrey V. Savochkin <saw@saw.sw.com.sg> and others
Jul 12 21:38:42 ra passwd(pam_unix)[13984]: password changed for test
Jul 12 21:42:42 ra http_server[3206]: Client shut down the connection owned by im_id(1).
Jul 13 11:14:32 ra kernel: SCSI device sdb: 16777216 512-byte hdwr sectors (8590
--More--(45%)
```

##### - /var/log/wtmp

```
[root@rocky httpd]# last
```

```
root pts/3 :0.0 Tue Jul 13 20:55 still logged in
root pts/2 :0.0 Tue Jul 13 20:55 - 20:55 (00:00)
root pts/1 :0.0 Tue Jul 13 20:55 - 20:55 (00:00)
root pts/2 :0.0 Mon Jul 12 21:29 - 20:55 (23:25)
root pts/1 :0.0 Sun Jul 11 17:47 - 20:55 (2+03:07)
root :0 Sun Jul 11 17:47 still logged in
.....
```

```
wtmp begins Thu Jul 8 19:04:46 2010
```

##### - /var/log/lastlog

```
[root@rocky httpd]# lastlog
```

| 사용자명   | 포트    | ~로부터        | 최근정보                       |
|--------|-------|-------------|----------------------------|
| 사용자명   | 포트    | ~로부터        | 최근정보                       |
| root   | pts/2 | 192.168.0.2 | 월 2월 7 23:01:56 +0900 2011 |
| bin    |       |             | **한번도 로그인한 적이 없습니다**       |
| daemon |       |             | **한번도 로그인한 적이 없습니다**       |
| adm    |       |             | **한번도 로그인한 적이 없습니다**       |
| lp     |       |             | **한번도 로그인한 적이 없습니다**       |
| sync   |       |             | **한번도 로그인한 적이 없습니다**       |
| ....   |       |             |                            |

#### (5) logrotate

- 로그 파일은 지속적으로 쌓이기 때문에 일정 주기로 정리를 해야 함

- logrotate 프로그램을 사용해 주기적으로 로그 파일을 백업하고 일정 주기로 로그 파일 삭제
- 설정 파일: /etc/logrotate.conf

```
[root@rocky httpd]# cat /etc/logrotate.conf
see "man logrotate" for details
rotate log files weekly
weekly

keep 4 weeks worth of backlogs
rotate 4
```

```
create new (empty) log files after rotating old ones
create

uncomment this if you want your log files compressed
#compress

RPM packages drop log rotation information into this directory
include /etc/logrotate.d

no packages own wtmp -- we'll rotate them here
/var/log/wtmp {
 monthly
 create 0664 root utmp
 rotate 1
}

system-specific logs may be also be configured here.
```

- /etc/logrotate.d: 데몬의 실행 정보 포함한 디렉토리
  - week: 주 단위로 백업 파일 생성
  - rotate: 일주일마다 하나의 로그 파일을 만들고 4주 간격으로 돌아감
  - 백업 파일이 생성된 후에는 새로운 로그 파일을 생성

```
[root@rocky logrotate.d]# ls
acpid httpd named ppp rpm samba syslog up2date yum
cups mgetty opensm psacct sa-update squid tux vsftpd.log
```

#### - 실습

```
logger -p local1.warning "rsyslog test1"
logger -p local1.info "rsyslog test2"
logger -p local1.err "rsyslog test2"
```

=> 로그서버를 구성하기 위해 TCP/UDP 514 포트 리스닝 설정

/etc/rsyslog.conf

# Provides UDP syslog reception

# for parameters see <http://www.rsyslog.com/doc/imudp.html>

module(load="imudp") # needs to be done just once => 주석제거

input(type="imudp" port="514") => 주석제거

# Provides TCP syslog reception

# for parameters see <http://www.rsyslog.com/doc/imtcp.html>

---

```
module(load="imtcp") # needs to be done just once => 주석제거
input(type="imtcp" port="514") => 주석제거
```

- 방화벽 개방

```
firewall-cmd --permanent --add-service=syslog => UDP/514
firewall-cmd --permanent --add-port=514/TCP => TCP/514
firewall-cmd --reload
```

---

## Chapter 13. Backup

---

### 13.1 Backup

#### I . Backup

; 시스템의 고장, 장애로 인한 사고에 대비

##### (1) Backup 필요성

- 장애 발생 시 신속한 서비스의 재개로 업무 손실 최소화
- 사용자 및 관리자의 실수로 발생할 수 있는 데이터 손실 방지
- 바이러스, 천재지변에 의한 장애 대비

#### II . Backup 종류

##### (1) 전체 백업(Full Backup)

- 파일시스템의 전체 내용을 그대로 백업
- Backup의 방법 중 가장 최선의 방법
- 데이터의 양이 많을 경우 부담

##### (2) 증분 백업(Incremental Backup)

- 증분 백업은 이전 백업한 시점부터 변경된 데이터만 백업
- 전체 백업이 먼저 이루어져야 함
- 백업 공간이 줄어들고 빠르게 백업을 진행

#### III . Backup 전 고려할 사항

- 백업 데이터 선택 여부
- 백업 주기
- 매체 선택 여부
- 백업 방법

#### IV . Backup 툴 선택

(1) tar : 유닉스와 리눅스 시스템에서 사용되는 유틸리티로 파일이나 디렉토리를 하나의 파일로 묶어주는 역할을 한다. 증분백업기능 지원

##### (2) cpio

- tar와 비슷한 유틸리티로 많은 양의 데이터에 대해서는 tar보다 빠르다. 장치파일이나 네트워크 파일등의 특수 파일도 백업이 가능하고, 백업본의 크기도 작으며, 백업본에 손상된 부분이 있더라도 손상된 부분을 제외하고 나머지 부분을 복구한다. 증분백업기능 지원 안함.

##### (3) dump

- ext2나 ext3 파일시스템만 지원하고, 파일들을 백업하지 않고 파일시스템 자체를 직접

---

읽어내는 방법을 사용한다. 전체백업과 증분백업이 가능하다.

## 13.2 Backup 툴 사용법

### 1. tar

#### (1) 전체 백업

- /data 디렉토리 생성 후 데이터 생성
- /backup 디렉토리 생성 후 데이터 백업
- /restore 디렉토리 생성 후 백업 된 데이터를 이용하여 복구

```
[root@rocky /]# mkdir /data
[root@rocky /]# mkdir /backup
[root@rocky /]# mkdir /restore

[root@rocky /]# man ls > /data/full.data
[root@rocky ~]# tar zcfp /backup/full.bak.tgz -g /backup/info /data
tar: Removing leading `/' from member names
[root@rocky ~]# ls -l /backup
합계 12
-rw-r--r-- 1 root root 11 7월 11 21:00 info
-rw-r--r-- 1 root root 5241 7월 11 21:00 full.bak.tgz
```

#### (2) 전체 백업 복구

```
[root@rocky restore]# cd /backup
[root@rocky backup]# tar xzf full.bak.tgz -C /restore
[root@rocky backup]# ls -l /restore
합계 4
drwxr-xr-x 2 root root 4096 7월 11 20:53 data
[root@rocky backup]# ls -l /restore/data/
합계 16
-rw-r--r-- 1 root root 14648 7월 11 20:53 full.data
[root@rocky backup]#
```

#### (3) 증분 백업

- 기본적으로 tar 명령에 -g, --listed-incremental 옵션을 제공
- /data 디렉토리에 데이터 추가 후 백업

```

[root@rocky ~]# man pwd > /data/1day.data
[root@rocky ~]# tar zcfp /backup/1day.bak.tgz -g /backup/info /data
tar: Removing leading `/' from member names
[root@rocky ~]# ls -l /backup
합계 16
-rw-r--r-- 1 root root 713 7월 11 21:03 1day.bak.tgz
-rw-r--r-- 1 root root 11 7월 11 21:03 info
-rw-r--r-- 1 root root 5241 7월 11 21:00 full.bak.tgz

[root@rocky ~]# man cd > /data/2day.data
[root@rocky ~]# tar zcfp /backup/2day.bak.tgz -g /backup/info /data
tar: Removing leading `/' from member names
[root@rocky ~]# ls -l /backup
합계 20
-rw-r--r-- 1 root root 713 7월 11 21:03 1day.bak.tgz
-rw-r--r-- 1 root root 585 7월 11 21:04 2day.bak.tgz
-rw-r--r-- 1 root root 11 7월 11 21:04 info
-rw-r--r-- 1 root root 5241 7월 11 21:00 full.bak.tgz

```

#### (4) 증분 백업 복구

```

[root@rocky backup]# tar xzf 1day.bak.tgz -C /restore
[root@rocky backup]# ls -l /restore/data/
합계 20
-rw-r--r-- 1 root root 1061 7월 11 21:03 1day.data
-rw-r--r-- 1 root root 14648 7월 11 20:53 full.data
[root@rocky backup]# tar xzf 2day.bak.tgz -C /restore
[root@rocky backup]# ls -l /restore/data/
합계 24
-rw-r--r-- 1 root root 1061 7월 11 21:03 1day.data
-rw-r--r-- 1 root root 671 7월 11 21:04 2day.data
-rw-r--r-- 1 root root 14648 7월 11 20:53 full.data

```

## II. cpio

- cpio(copy in / copy out)으로 여러 개의 리눅스 파일을 테이프 드라이브에 저장 / 복구
- 명령어 사용 방법: cpio [ Option ] [ 장치 이름 ] [ 파일 이름 ]
  - -o: 하드디스크의 데이터를 cpio 형식의 파일로 데이터 백업
  - -i: 표준 입력으로 정의된 cpio 형식의 파일을 읽어와 원래의 파일과 디렉토리로 복구
  - -u: 같은 이름의 파일이 있는 경우 겹쳐쓰기 한다.
  - -d: restore중 필요한 디렉토리를 자동으로 생성한다.
  - -p: 디렉토리의 데이터를 다른 디렉토리로 복사
  - -t: -i 옵션과 같이 사용, cpio 파일에 포함된 디렉토리와 파일의 목록 출력

#### (1) 파일로 데이터 백업

```
[root@rocky ~]# ls | cpio -ocv > /backup.cpio
Desktop
anaconda-ks.cfg
cpio
install.log
install.log.syslog
254 blocks
[root@rocky ~]# ls -l /backup.cpio
-rw-r--r-- 1 root root 130048 7월 11 19:42 /backup.cpio
```

## (2) 파일 복원

```
[root@rocky ~]# cd /test
[root@rocky test]# cpio -ivc < /backup.cpio
Desktop
anaconda-ks.cfg
cpio
install.log
install.log.syslog
254 blocks
[root@rocky test]# ls
Desktop anaconda-ks.cfg cpio install.log install.log.syslog
```

## III. dump

- (1) dump는 파일들이 아닌 파일시스템 전체를 백업한다. 보통 파티션단위로 백업할 때 사용하면 편리. dump는 0-9 단계의 레벨을 가지고 증분백업을 지원. 레벨 0이 전체 백업이고 나머지 레벨들은 부분백업을 하게 됨.

- (2) 사용법(백업) : dump -[option] 백업미디어 백업할 대상

- option

0-9 : 덤프의 레벨 지정. 기본레벨 9

f : 파일명 또는 디바이스명

u : dump를 통해 백업한 후에 /etc/dumpdates라는 파일에 작업에 대한 정보 기록

예) # dump -0f /dev/st0 /home

=> /home 디렉토리 전체를 /dev/st0 장치에 전체 백업

# dump -0uf home /dev/hda7

=> /dev/hda7 파일시스템 전체를 home이라는 파일명으로 전체 백업하며

그 작업정보를 /etc/dumpdates 라는 파일에 기록.

참고) 파일시스템 단위의 백업일 경우에만 /etc/dumpdates 파일에 작업정보를 남긴다.

# dump -4uf /dev/st0 /dev/hda7

=> level 4 수준 이하의 바뀐 내용을 증분백업한다.

또한 그 작업정보를 /etc/dumpdates 라는 파일에 기록.

---

(3) 사용법(복원) : `restore -[option]` 디바이스명 또는 백업파일명

- option

    i : 대화식(interactive)으로 복구할 파일을 선택한 후에 복원.

    f : 파일명 또는 디바이스명 지정

    r : 전체 복원

예) # `restore -rf /dev/st0`

=> /dev/st0에 백업된 데이터를 전체 복원.

예) # `restore -if /dev/st0`

=> /dev/st0에 백업된 데이터를 하나씩 확인해가면서 복원.

# `restore -if home.bakup`

=> home.bakup이라는 이름으로 백업된 데이터를 복원.



### 14.1 FTP Service 개요

#### I. FTP Service 소개

##### (1) FTP Service

- File Transfer Protocol(TCP 20, 21 Port 사용)
- TCP/IP 환경에서 파일 전송을 가능하게 해주는 서비스
- Log-In을 통한 원격 시스템간의 파일 교환 목적으로 사용
- 기본 FTP 서비스는 보안상 취약한 단점이 존재하여 보안이 고려된 별도의 프로그램을 설치하여 사용(ProFTP, vsFTP 등..)

##### (2) 종류

- Private FTP
  - FTP Server에 등록된 사용자 계정과 비밀번호를 통한 일반적인 FTP 접속
  - 계정별로 FTP Server에 접근할 수 있는 권한(Download/Upload)의 명시 가능
- Anonymous FTP
  - FTP Server에 등록된 사용자 계정이 없을 경우 anonymous 계정을 통한 익명 FTP 접속
  - 공용 FTP Server에서 자주 사용되며 불특정 다수를 대상으로 FTP Service를 운영할 경우 사용

##### (3) vsftpd

- Red Hat 7.X 버전 이전에는 ProFTP 프로그램을 주로 사용
- Red Hat Enterprise Linux가 출시되면서 vsFTPD를 기본 FTP 프로그램으로 포함
- vsFTPD(very secure FTPD)는 보안, 성능, 안정성이 우수한 FTP 프로그램
- 주요 기능
  - 가상 IP 별 별도의 환경 설정 가능
  - 가상 사용자 설정
  - 전송 대역폭 지정
  - PAM 지원(PAM을 통한 wtmp에 로그 기록 생성)
  - xferlog 표준 로그 파일보다 상세한 자체 로그 파일 형식 지원
  - Standalone 방식과 xinetd를 통한 운영 방식 모두 지원

##### (4) Active 모드 / Passive 모드

### 14.2 vsftpd Service 설치 및 설정

#### I. vsftpd Server 설치 및 활성화

##### (1) 설치 확인 및 설치

- [root@rocky ~]# rpm -qa | grep vsftp
- vsftpd-(버전)

- 
- [root@rocky ~]# rpm -ivh vsftpd-(버전).rpm

## II. vsftpd Server 설정

### (1) vsftpd Server 주요 설정 파일

- /etc/vsftpd/vsftpd.conf  
→ vsFTPd 환경 설정 파일
- /usr/sbin/vsftpd  
→ vsftpd 바이너리 파일
- /etc/rc.d/init.d/vsftpd  
→ vsftpd 데몬 실행 스크립트 파일
- /etc/vsftpd.ftpusers  
→ vsftpd 접속을 막을 사용자 설정
- /etc/xinetd.d/vsftpd  
→ xinetd 서비스 파일(파일명 변경 가능)

### (2) vsftpd 설정 파일 편집

- 설정 파일 : /etc/vsftpd/vsftpd.conf
  - anonymous\_enable=YES  
→ FTP Server에서 익명 사용자(anonymous) 연결을 허용할 것인지 결정
  - local\_enable=YES  
→ 로컬 시스템 계정의 로그인 허용 여부 결정
  - write\_enable=YES  
→ FTP Server에 접속 후 쓰기 권한(저장, 삭제, 디렉토리 생성 등)의 허용 여부  
→ 익명 사용자(anonymous)는 해당되지 않음
  - local\_umask=022  
→ 로컬 시스템 계정이 파일을 업로드 할 경우 파일의 퍼미션을 정의하는 항목
  - #anon\_upload\_enable=YES  
→ 익명 사용자(anonymous)의 파일 업로드 허용 여부 결정
  - #anon\_mkdir\_write\_enable=YES  
→ 익명 사용자(anonymous)의 디렉토리 생성 허용 여부 결정
  - dirmessage\_enable=YES  
→ FTP Server에 접속한 사용자가 디렉토리를 이동할 경우 메시지 파일 내용을 보여줄 것인지의 여부
  - message\_file=.message  
→ 디렉토리를 이동할 경우 보여줄 메시지 파일 이름을 정의
  - xferlog\_enable=YES  
→ FTP Server 접속자의 업로드/다운로드 상황을 로그 파일(/var/log/xferlog)로 저장할지 여부
  - connect\_from\_port\_20=YES  
→ FTP Server를 Standalone 타입으로 운영할 때 데이터 전송에 20번 포트를 사용할지 여부
  - #chown\_uploads=YES  
→ 익명 사용자(anonymous)가 업로드 한 파일에 대해 소유권을 변경할지 여부
  - #chown\_username=whoever

- 
- 익명 사용자(anonymous)가 업로드 한 파일의 소유권 변경 시 사용할 유저
  - #xferlog\_file=/var/log/vsftpd.log
    - 로그 파일 저장 시 사용할 파일명 지정
  - xferlog\_std\_format=YES
    - 로그 파일을 표준 포맷으로 기록할지 여부 결정
    - 표준 포맷은 사용자의 로그인, 디렉토리 생성 여부 등의 로그를 남기지 않음
    - vsFTPd 포맷의 로그는 로그인 정보, 디렉토리 생성 여부 등을 포함하여 상세한 로그를 남김
  - #idle\_session\_timeout=600
    - 접속자가 접속 후 아무 작업도 하지 않고 600초가 경과되면 자동으로 FTP 연결 종료
  - #data\_connection\_timeout=120
    - 파일 전송이 종료된 후 또 다른 데이터 전송을 위해 연결을 유지하는 시간
  - #ascii\_upload\_enable=YES
    - 아스키 파일 업로드의 허용 여부
  - #ascii\_download\_enable=YES
    - 아스키 파일 다운로드의 허용 여부 결정
  - #ftpd\_banner=Welcome to blah FTP Service
    - FTP Server에 접속한 사용자에게 보여줄 텍스트 환영 메시지 설정
  - #deny\_email\_enable=YES
    - 익명 사용자(anonymous)를 통한 로그인 과정에서 패스워드에 입력하는 E-Mail 주소를 확인해 특정 E-Mail 주소의 접근을 거부할지의 여부
  - #banned\_email\_file=/etc/vsftpd.banned\_emails
    - 접근을 거부할 E-Mail 주소의 목록을 갖는 파일 이름을 정의
  - #chroot\_local\_user=YES
    - FTP Server에 접근한 사용자에게 홈 디렉토리 이상의 상위 디렉토리 접근 허용 여부(chroot 설정)
  - #chroot\_list\_enable=yes
    - chroot 설정을 한 뒤 일부 사용자를 chroot에서 제외할 경우 설정
  - #chroot\_list\_file=/etc/vsftpd.chroot\_list
    - chroot에서 제외할 사용자들의 목록을 갖는 파일 이름을 정의
  - #ls\_recurse\_enable=YES
    - 사용자들에게 "ls -R" 명령어를 사용 가능 하도록 할 것인지 설정
  - #pam\_service\_name=vsftpd
    - PAM 인증으로 사용자들의 접근을 허용할 경우 사용할 인증 파일 이름 정의
  - #max\_clients=100
    - 최대 동시 접속자수 설정(단, Standalone 타입으로 운영될 경우에만 해당)
  - #max\_per\_ip=5
    - 동일 IP당 허용할 동시 접속자 수 설정(단, Standalone 타입으로 운영될 경우에만 해당)
  - listen=YES
    - listen\_port에 들어온 요청을 데몬이 받아들일지 여부를 설정(Standalone 타입으로 운영될 경우에 YES)
-

---

## Chapter 15. DNS Service

---

### 15.1 DNS Service 개요

#### I. DNS Service 소개

##### (1) Domain Name

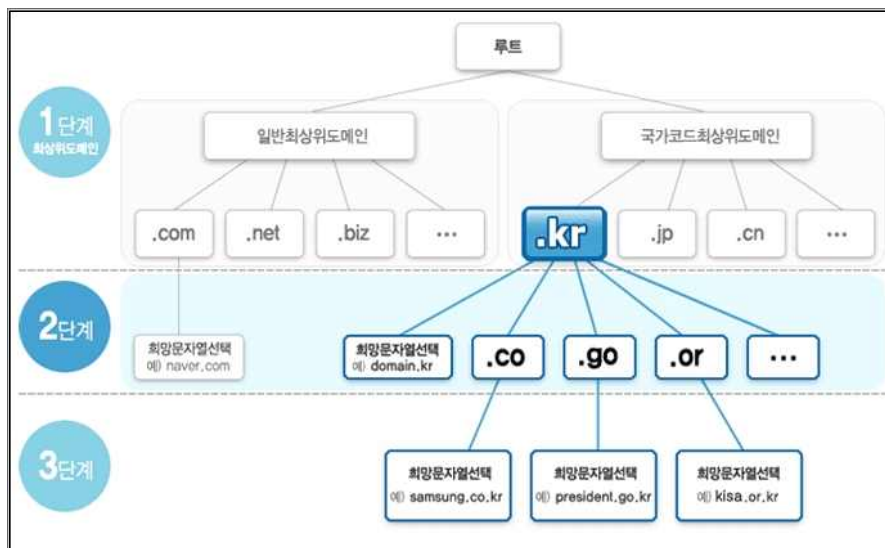
- Internet 상에서 Host 구별 방법
- 기억하기 까다로운 IP Address의 대체 수단
  - 숫자로 이루어진 IP Address를 수 많은 Host 마다 기억하기는 어려움
  - IP Address를 사람이 기억하기 쉬운 문자와 숫자의 조합인 Domain Name과 매핑
- 현재 Internet Service의 기본 구성 요소
- Domain Name 등록 원칙
  - 영문자(a-z) 26개, 숫자(0-9) 10개, 특수 기호(-) 1개를 합쳐 총 37개 글자 조합
  - 영문자의 대·소문자 구분은 없으며, 특수 기호(-)는 처음과 끝에는 올 수 없음
  - 분산 관리 원칙에 따라 각 단계별로 독자적인 관리를 하며 각 단계별 구분은 마침표(.) 사용
  - 전 세계적으로 중복되지 않아야 하며, 선착순 원칙(First Come, First Served)에 따라 부여
- Domain Name 관리 기구
  - ICANN(The Internet Corporation for Assigned Names and Numbers)
    - <http://www.icann.org>
    - 과거 Domain Name을 총괄하던 IANA의 후신으로 최상위 도메인 이름을 생성하고 그 등록 기관을 결정
  - IANA(Internet Assigned Numbers Authority)
    - <http://www.iana.org>
  - NIC(Network Information Center)
    - <http://www.krnic.or.kr>
    - 국가 최상위 도메인은 각 국가 NIC에서 관리
  - KISA(Korea Internet & Security Agency, 한국 인터넷 진흥원)
    - KRNIC이 현재는 KISA의 인터넷주소자원 관리 담당부서

##### (2) DNS 구성 요소

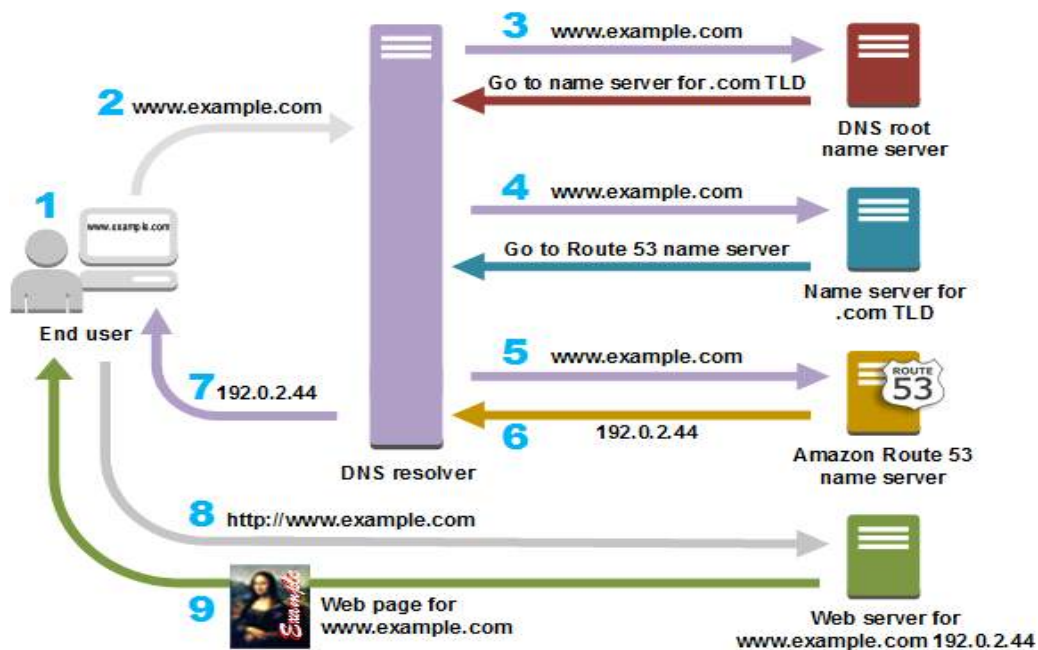
- Domain Name Space
  - 인터넷에서 사용되고 있는 도메인 네임의 계층적 구조 공간을 의미
  - Root Domain(.) 이하로 Tree(트리) 형태의 계층적인 구조로 구성
    - “. → kr. → co.kr. → google.co.kr.”
  - Root Domain은 최상위 도메인 정보를 관리하고 최상위 도메인은 그 하위 도메인에 대한 정보를 관리
    - 이러한 계층 구조로 인하여 정보는 각 도메인의 네임 서버로 분산, 관리
- Resource Record
  - Domain Name Space에서 지정된 Domain Name에 대해 필요한 인터넷 자원 정보를 매핑하는 수단을 제공

- Domain Name과 IP Address를 연결하여 DNS Database 구성
- 하나의 도메인 네임이 갖는 속성 정보(예: www, ftp, mail 등)를 지정하는 수단
  - 리소스 레코드는 확장이 가능
    - 예) IPv4:A type의 리소스 레코드 사용, IPv6:AAAA Type의 레코드 추가 정의
- Name Server
- Domain Zone(예: netcollege.co.kr)의 정보를 소유하고 이에 대한 질의에 대해 응답하는 역할을 수행
  - 보통 DNS Server라 칭하며 특정 질의에 대해서 자신이 소유한 Domain Zone의 정보만 응답
- Resolver
- Name Server에 의해 구성된 도메인 데이터 베이스를 검색하는 역할
  - 요청된 리소스 레코드가 존재하는 위치를 DNS에서 찾고, 찾은 정보를 최종 응답으로 되돌려 주는 기능을 담당
  - 전체 도메인 데이터 베이스를 검색할 수 있도록 DNS 검색의 시작점이 되는 Root Name Server의 IP Address를 가짐.
  - 짧은 시간동안 동일한 Domain Name에 대한 질의 반복을 방지하기 위해 Cache 사용

[그림 15.1] DNS 체계



### (3) DNS 동작 과정



## 15.2 DNS Service 설치 및 설정

### I . DNS Server 설치 및 활성화

#### (1) 설치 확인 및 설치

- DNS Server 구축을 위해 BIND(Berkeley Internet Name Domain) program 필요
- rocky 서버모드로 설치된 경우에는 기본적으로 bind RPM 패키지가 설치된다.
- `[root@rocky ~]# dnf install bind-chroot` : name server 동작시  
root의 위치를 변경하기 위한 패키지  
`/var/named/chroot/var/named`  
`/etc`

#### (2) 서비스 시작하기 :

- `[root@rocky ~]# systemctl start named` 또는
- `[root@rocky ~]# systemctl start named-chroot`  
→ DNS Service 시작

#### (3) DNS Server 방화벽 설정

##### 「DNS Server와 방화벽 설정」

- DNS Server는 TCP 53번 Port, UDP 53번 Port를 사용
- TCP 53 Port : Primary DNS Server와 Secondary DNS Server 간의 동기화 관련 통신에 사용
- UDP 53 Port : DNS Server를 이용한 Client의 Query / Answer 관련 통신에 사용

#### • 방화벽 설정

```
firewall-cmd --permanent --add-service=dns : 포트 추가
firewall-cmd --reload : 적용
```

---

## II . DNS Server 설정

### (1) DNS Server 주요 설정 파일

- /etc/named.conf  
→ named 데몬의 설정파일
- /etc/named.rfc1912.zones  
→ 각 Domain의 속성을 설정하는 영역 파일

### (2) DNS 설정 파일 편집

- 설정 파일 : /etc/named.conf => 심볼릭 링크 파일
- Name Server 데몬이 시작하면서 가장 먼저 읽는 Name Server 설정 파일  
→ chroot 환경에서는 /var/named/chroot/etc/named.caching-nameserver.conf
- 오타가 하나라도 존재하면 DNS Server가 정상적으로 작동하지 않음.
- named.conf
- options 영역 : Name Server 전역에 관한 옵션 설정
  - options {  
    directory "/var/named";  
    dump-file "/var/named/data/cache\_dump.db";  
    statistics-file "/var/named/data/named\_stats.txt";  
};  
→ directory "/var/named";  
    : Name Server 관련 Database 파일들이 존재할 디렉토리 지정  
    (named.ca, \*.zone...)  
    : chroot 환경에서는 /var/named/chroot/var/named에 파일이 존재  
→ dump-file "/var/named/data/cache\_dump.db";  
    : Name Server가 갖고 있는 DNS 정보가 갱신될 때 저장되는 파일의 위치와  
    파일명 지정  
→ statistics-file "/var/named/data/named\_stats.txt";  
    : Name Server의 통계를 처리하는 용도로 사용될 파일의 위치와 파일명 지정  
→ allow-transfer { IP address; };  
    : Name Server를 2개 이상으로 구성시 Master(1차) Name Server에서 Slave(2차)  
    Name Server로 Zone transfer가 가능하도록 설정.
- zone 영역 : root 영역, localhost 영역, master/slave 설정, 복수 영역 설정
  - zone “도메인명” IN {  
    type ( hint | master | slave );  
    file "zone file name";  
    allow-update { IP Address | none; };  
};  
→ 도메인명  
    : 도메인명 값에 따라 두 개의 존(Forward Zone, Reverse Zone)으로 구분  
    : Forward Zone Database는 “.” (hint) 또는 “Domain Name”을 사용  
    : Reverse Zone Database는 네트워크 주소를 정의하되 네트워크 주소를

---

반대로 표시(Ex. 10.168.192)

→ Type

: 캐싱 전용 Name Server일 경우 "hint", 1차 Name Server일 경우 "master",  
2차 Name Server일 경우 "slave"

→ file

: options에서 지정한 /var/named 디렉토리에 존재하는 각 zone의 구체적인  
설정 값을 갖는 파일명 입력

→ allow-update

: 1차(Master) Name Server의 존재에 대해 업데이트가 가능한 IP주소 혹은 서브네트워크  
대역을 지정한다. 즉, Dynamic DNS가 활성화된다.

: 업데이트를 허용하지 않을 경우 "{ none; };" 입력

#### ◆ /etc/host.conf

- order : URL 주소, hostname 등으로 질의를 받았을 때 참조할 대상과 순서를  
결정하는 키워드

- order 키워드에 대한 option : hosts, bind

- 기본 설정 확인

- [root@rocky /]# cat /etc/host.conf  
order hosts,bind

- /etc/host.conf 파일 작동 원리

① [root@rocky /]# ping www.yahoo.co.kr → 사용자가 요청했다고 가정

② www.yahoo.co.kr에 해당하는 IP Address를 알아오는 과정 필요

③ /etc/host.conf 파일의 설정 내용 확인

→ 설정 내용이 기본값(order hosts, bind) 이라고 가정

④ 첫 번째로 hosts(⑤번 과정)를 통해 해석 시도, 실패 시 두 번째로 bind(⑥번  
과정)를 통해 해석 시도

⑤ /etc/hosts 파일을 참조하여 www.yahoo.co.kr에 대한 IP Address 해석 시도

⑥ bind 참조 → /etc/resolv.conf 에 설정된 name server에 질의

#### ◆ /etc/hosts

- 네트워크에 연결되어 있는 시스템들에 대한 IP Address와 hostname 맵핑

- 자주 방문하는 사이트나 시스템을 등록해 주어 편리하게 사용 가능

- 형식 : IP 주소 | hostname | Alias(별칭)

- 사용 예

- [root@rocky ~]# cat /etc/hosts  
127.0.0.1 localhost.localdomain localhost  
192.168.0.161 www.white.co.kr www

- [root@rocky /]# ping www.white.co.kr  
→ www.white.co.kr(hostname)과 매핑된 192.168.0.161로 ping 시도

- [root@rocky /]# ping www  
→ www(alias)와 매핑된 192.168.0.161로 ping 시도

#### ◆ /etc/resolv.conf



- 로컬 시스템에서 특정 도메인(혹은 IP)에 대한 질의가 있을 경우 Resolver(이름 풀이 프로그램)가 실행됨
- Resolver가 이름 풀이를 수행하기 위해 이용할 DNS Server를 지정하는 파일
- 설정 예
  - [root@rocky ~]# cat /etc/resolv.conf
  - nameserver 168.126.63.1
  - nameserver 168.126.63.2
- nameserver 지시자 : 로컬 시스템이 사용할 DNS Server를 지정하며 3차까지 지정이 가능
  - nameserver 168.126.63.1
  - nameserver 168.126.63.2
  - nameserver 164.124.101.2
- domain 지시자 : 기본적으로 사용할 도메인 이름을 지정
  - domain white.co.kr
  - ping www라고 치면 ping www.white.co.kr 과 같은 결과
- search 지시자 : domain 지시자와는 달리 기본 도메인 이름을 여러 개 지정
  - search white.co.kr, black.co.kr, red.co.kr
  - ping www : www.white.co.kr에게 ping을 시도하고 응답이 없으면  
www.black.co.kr에게 ping 시도
  - www.black.co.kr에게도 ping 응답이 없으면 www.red.co.kr에게 ping 시도

#### ◆ Zone 파일 생성

- /etc/named.conf에서 명시한 각 도메인 별로 zone파일 생성
  - zone 파일 : chroot 환경에서의 DNS Server 구축이므로  
/var/named/chroot/var/named에 생성
- white.co.kr 도메인에 대한 zone 파일 생성
  - [root@rocky ~]# cd /var/named/chroot/var/named
  - [root@rocky ~]# vi white.co.kr.zone → /etc/named.conf에서 명시한  
이름으로 파일 생성
- Resource Record(RR)
  - A(AAAA) : 호스트 이름에 할당되어 있는 IP주소를 지정함. AAAA는 IPv6주소에 대응.
  - NS : Name Server를 지정함.
  - MX : Mail eXchanger의 약자로 메일 서버를 의미함.
  - CNAME : Canonical NAME의 약자로 정규이름을 의미함. 별칭을 위해 사용됨.
  - PTR : Pointer의 약자로 IP주소로 호스트명을 변환하기 위해 사용됨.
  - SPF : 특정 도메인에서 이메일을 보낼 수 있도록 승인된 모든 서버를 열거.
  - TXT : 호스트나 기타 이름과 임의의 텍스트를 연계하는 기능을 제공.
  - SRV : 특정 서비스를 호스팅하는 컴퓨터를 식별하는 데 사용.

---

- zone 파일 분석

- \$TTL 86400

→ Time To Live의 약자로 white.co.kr 도메인에 대한 호스트 이름을 질의해 간 다른 네임 서버가

해당 호스트의 IP 주소를 캐시에 저장하는 시간을 의미

- @ IN SOA ns.white.co.kr. admin.white.co.kr. (

→ @ : /etc/named.conf에 정의된 white.co.kr을 의미(@ 대신 white.co.kr.로 입력해도 무방

→ IN : IN(Internet)은 네트워킹 어드레스 클래스를 의미.

레코드 리소스(RR)를 지정할 때 사용.

→ SOA : Start Of Authority의 약자.

네임서버가 해당 도메인에 대해 모든 리소스 레코드의 권한을 갖는다는 것을 의미

→ SOA 필드 값의 의미

- Serial : 네임서버의 데이터버전을 나타냄. 네임서버 설정이 변경될 경우 이 시리얼을 높여야 한다. 2차 네임서버 존재시 이 Serial을 자신의 Serial과 비교 후 동기화.

- Refresh : 2차 네임서버가 1차 네임서버에게 새롭게 업데이트된 정보가 있는지 요청할 때까지의 시간.

- Retry : 2차 네임서버와 1차 네임서버가 연결되지 않을 경우 재 연결시까지의 대기시간

- Expire : 2차 네임서버가 일정시간동안 1차 네임서버에 접속하지 못했을 경우 기존

정보를 의미 없는 것으로 간주하여 파기할 때까지의 시간.

- Minimum(TTL) : zone설정파일의 맨 처음에 있는 값으로 다른 서버에서 자신의 정보를 가져갔을 때, 그 쪽 서버의 캐시에 그 정보가 얼마나 오랫동안 머물 것인지를 결정

- IN NS ns.white.co.kr.

→ NS : Name Server의 약자, 해당 도메인(white.co.kr.)의 네임 서버 역할을 하는 시스템을 지정

- IN MX 10 mail.white.co.kr.

→ MX : Mail Exchanger의 약자, Mail Server 시스템을 지정

→ 10 : 순위를 나타낸다. 여러 개의 메일 서버 존재시 낮은 순위의 메일서버를 이용. 1차 메일서버에 문제 발생시 2차 메일서버에 메일을 저장한 후, 1차 메일서버 복구시 저장하고 있던 메일을 1차 메일서버로 넘겨준다. => 백업서버의 개념.

- www IN A 192.168.0.160

→ A : 호스트 이름에 해당하는 IP 지정(Ex. www.white.co.kr 시스템의 IP Address는 192.168.0.160)

- 역방향 존 설정

/etc/named.rfc1912.zones

zone "80.168.192.in-addr.arpa" IN {

type master;

file "abc.reverse";

};

---

- 역방향 존파일 설정

/var/named/abc.reverse

\$TTL 86400

```
@ IN SOA ns.abc.com. admin.abc.com. (
 20210430 ; Serial
 1H ; Refresh
 3H ; Retry
 1W ; Expire
 1D) ; Minimum
```

```
 IN NS ns.gdn15.com.
```

```
206 IN PTR ns.abc.com.
```

```
141 IN PTR ftp.abc.com.
```

```
201 IN PTR mail.abc.com.
```

```
101 IN PTR www.abc.com.
```

Master/Slave : Master의 존정보를 Slave에 동기화해서 DNS서버의 장애를 대비한다.

nameserver 192.168.80.206(M)

nameserver 192.168.80.141(S)

Master 에서 존정보를 설정하고 Slave에서는 Master의 존정보를 가져온다.

Master 서버> Master 서버의 경우 TCP 53이 열려있어야 함.

/etc/named.rfc1912.zones

```
zone "abc.com" IN {
 type master; => Master로 동작
 allow-transfer { 192.168.80.141; }; => Slave IP주소 지정
 file "abc.zone";
};
```

Slave 서버> : 존 설정만 하고 존 파일은 생성하지 않음.

```
zone "abc.com" IN {
 type slave; => Slave로 동작
 masters { 192.168.80.206; };
 file "slaves/abc.zone";
};
```

zone transfer로 전송된 zone파일은 data 형식이므로 cat으로 볼 수 없음. 내용을 확인하기 위해 다음과 같은 명령을 이용한다.

---

```
named-compilezone -f raw -F text -o gdn15.txt abc.com /var/named/slaves/abc.zone
```

#### (4) 설정파일의 구문 오류 체크 유틸리티

##### - named-checkconf

- /etc/named.conf 파일의 구문 오류 또는 잘못된 철자를 체크해 준다.
  - 사용법 : named-checkconf 파일명
  - # named-checkconf /etc/named.conf
- 오류가 없으면 아무 메시지도 보이지 않는다.

##### - named-checkzone

- /var/named/chroot/var/named 디렉토리에 있는 zone파일의 구문오류 또는 잘못된 철자를 체크해 준다.
  - 사용법 : named-checkzone 도메인명 존파일명
  - # named-checkzone white.co.kr /var/named/chroot/var/named/white.zone  
zone localhost/IN: loaded serial 42 ==> 오류가 없을 경우  
OK

### 16.1 Web Service 개요

#### I. Web Service 소개

##### (1) Web Service

- Internet의 가장 대표적인 서비스
- Web Service를 제공하는 Web Server와 Client 사이에 HTTP 사용
- Apache, IIS(Internet Information Service) 등의 프로그램 이용

##### (2) Apache

- 1995년 NCSA HTTPD 1.3 버전을 기반으로 탄생. 최신버전 2.3
- HTML, PHP, JSP 등의 웹 페이지가 실행될 수 있는 환경을 제공
- 공개형 Web Server Program으로 다양한 환경의 플랫폼에서 작동
- 지속적인 패치를 통해 안정성 확보
- 전 세계 약 60%, 국내의 70% 이상의 점유율을 차지
- Apache 장점
  - 동적 공유 객체 지원
  - 모듈 사용으로 기능 추가가 용이
  - 보조적 프로그램이 매우 다양
  - 서버 설정상 최적화된 성능 발휘
  - 유연한 설정
  - 다양한 상황에 따른 서비스 지원
  - 공개된 형태로 개발됨
  - 안정성 점검이 전 세계적으로 이루어짐
  - 서버 운영체제에 최적화된 성능 제공

##### (3) Web Server 작동 원리

- 정적 Web Server 작동 방식
  - Client(Web Browser)를 통해 80번 Port로 Web Server에 연결
  - Client가 웹 문서(예: index.html)를 요청
  - Web Server는 Client가 요청한 파일을 시스템에서 검색
  - Web Server가 요청한 파일을 읽어 들임
  - Web Server는 읽어 들인 결과를 Web Browser에게 제공
- 동적 Web Server 작동 방식
  - Client(Web Browser)를 통해 80번 Port로 Web Server에 연결
  - Client가 동적인 페이지를 표현하는 PHP Program(예: board.php)을 요청
  - Web Server는 요청받은 PHP Program을 실행시킨 후 Client가 요청한 매개 변수를 전달
  - Web Server는 요청받은 PHP Program으로부터 생성된 결과를 읽어 들임
  - PHP Program으로부터 받은 결과를 Client(Web Browser)에게 제공

---

## 16.2 Web Service 설치 및 설정

### I. Web Server 설치 및 활성화

#### (1) 설치 확인 및 설치

- [root@rocky ~]# rpm -qa | grep httpd  
httpd-(버전)  
system-config-httpd-(버전)  
httpd-manual-(버전)

```
「Web Server 관련 패키지」
- httpd : httpd Web Server
- system-config-httpd : Apache Configuration Tool
- httpd-manual : httpd Web Server 문서 자료
- mod_perl : Apache Web Server를 위한 내장된 펄 해석기
- mod_ssl : Apache Web Server에 사용되는 암호 해독 지원
- mod_auth_mysql : MySQL을 이용한 Apache Web Server에 사용되는 기본 사용자 인증 프로그램
- crypto-utils : SSL Certificate and Key Management Utilities
- mod_python : Apache Web Server에 사용되는 내장 Python 해석기
```

- [root@rocky ~]# rpm -ivh httpd-(버전).rpm  
[root@rocky ~]# rpm -ivh httpd-manual-(버전).rpm  
[root@rocky ~]# rpm -ivh system-config-httpd-(버전).rpm
- yum 으로도 설치 가능

### II. Web Server 설정 파일

- 설정 파일 : /etc/httpd/conf/httpd.conf
- Section 1 : Global Environment(Web Server 전체에 영향을 미치는 환경 설정)
  - ServerTokens OS
    - Server HTTP 응답 헤더 설정, 오류문 등 Server 메시지에 출력되는 Server에 대한 정보 설정
    - 이 지시자의 값으로 Full, OS, Minor, Min, Major, Prod 값 설정 가능)
  - ServerRoot "/etc/httpd"
    - Apache가 정상적으로 작동하는데 필요한 여러 파일(설정 파일, 로그 파일, 바이너리 파일, 모듈 등)들의 위치 설정
  - PidFile run/httpd.pid
    - Apache가 실행될 때 자신의 프로세스 아이디(PID)를 기록하고자 하는 파일 지정
  - Timeout : 120
    - Client의 요청을 받은 뒤 응답해 주기까지의 시간을 의미, 시간 초과시 Client 접속은 해제
    - Client가 접속되어 있는 상태에서 여러 요청을 허용할 수 있는데 On 설정시 요청에 대한 처리 속도 향상
  - KeepAlive Off
    - 한 번 연결된 연결을 유지할 것인지 결정하는 지시자
  - MaxKeepAliveRequests : 100

- 
- 하나의 연결에서 허용할 최대 요청 수를 지정하는 지시자
  - 제한을 두지 않으면 0
  - KeepAliveTimeout 15
    - 동일한 Client가 지속적인 접속으로 다음 요청을 위해서 기다리는 시간을 설정
    - 지정된 시간동안 요청이 없을 시 접속을 해제
  - Listen : 80
    - Apache가 작동할 Port를 지정하는 지시자
    - 다중 IP Address 중 특정 IP Address에게만 별도의 Port를 부여할 경우에도 사용
  - Include conf.d/\*.conf
    - Apache의 설정 파일 httpd.conf 외에 별도로 특정 모듈들의 설정 파일을 따로 만들어서 사용
  - User apache
    - Group apache
    - Apache 프로세스를 실행할 때 소유권을 갖게 되는 사용자와 그룹 지정
    - Apache 메인 프로세스는 root의 권한으로 실행되고 나머지 자식 프로세스는 apache 권한으로 실행
- Section 2 : 'Main' Server Configuration(메인 서버 환경 설정)
- ServerAdmin root@localhost
    - 시스템 관리자의 E-Mail 주소 설정
  - ServerName new.host.name:80
    - Client에게 보내줄 호스트 이름을 설정(DNS에 등록된 이름을 사용)
    - 등록된 도메인이 없는 경우 IP Address 입력(Ex. 192.168.0.160:80)
  - DocumentRoot "/var/www/html"
    - Apache가 Service를 하는데 필요한 기본 웹 페이지 파일이 위치하는 디렉토리
  - <Directory />
    - Options FollowSymLinks
    - AllowOverride None</Directory>
    - Apache가 접근하는 각각의 디렉토리에 대해 어떤 기능/서비스를 허용하거나 거부할 것인지 설정
    - 보안을 고려하여 최상위 /(root) 디렉토리는 매우 제한적인 기본값 설정
  - <Directory "/var/www/html">
    - </Directory>
    - Apache의 기본 웹 문서 Directory의 접근 제한 설정
    - <Directory “경로”>와 </Directory> 지시자 사이에 Options, AllowOverride 설정을 통해 해당 디렉토리 접근 제한 설정
    - 접근 제한 설정은 지정된 경로의 하위 Directory까지 적용
-

| 「Options」            |                                                                                     |
|----------------------|-------------------------------------------------------------------------------------|
| 옵션                   | 의미                                                                                  |
| None                 | 옵션을 사용하지 않음                                                                         |
| All                  | Multiviews를 제외한 모든 옵션 사용                                                            |
| Indexes              | DirectoryIndex 파일(Ex. index.html)이 없을 경우 Directory의 내용을 보여주는 것을 허용, 보안상 사용하지 않음     |
| Includes             | Server측의 SSI(Server-Side Includes) 허용                                               |
| FollowSymLinks       | Directory의 Symbolic Link를 따를 것인지 결정                                                 |
| SymLinksifOwnerMatch | 대상 파일이 Symbolic Link 또는 Directory일 경우 원본 파일이 소유권자와 같은 권한일 경우 Symbolic Link를 따르도록 결정 |
| ExecCGI              | CGI Script 실행 허용 여부 결정                                                              |
| Multiviews           | client의 Browser에 따른 차별화된 내용을 보임                                                     |

|                                                                                     |  |
|-------------------------------------------------------------------------------------|--|
| 「AllowOverride」                                                                     |  |
| - 웹 문서 Directory에 접근 제어 설정 파일(.htaccess)을 생성하여 각 Directory의 권한 설정을 재정의 할 것인지 여부를 설정 |  |
| - All 또는 None으로 설정하거나 Options, FileInfo, AuthConfig, Limit를 조합하여 설정                 |  |

• UserDir disable

→ Web Server에 일반 계정으로 Home Page를 운영하고자 할 경우 일반 사용자의 웹 문서 Directory 지정

→ 설정 방법

① UserDir public\_html

② UserDir /var/html

③ UserDir /var/www/\*/docs

예 1> UserDir Public\_html 설정 시 지정되는 웹 문서 위치

: /home/username/Public\_html/index.html

예 2> UserDir enable

UserDir disable root user01 user02

: 일반 사용자에게 Web Directory 기능을 허용하되, root,user01, user02 사용자는 불허

→ 사용자 홈페이지 설정 후 해당 홈페이지 접근 : http://NS주소/~userid

'~'를 제거하려면 /etc/httpd/conf/httpd.conf 파일에 다음과 같이 Alias 추가  
Alias /userid/ "/home/userid/public\_html/"

• DirectoryIndex index.html index.html.var

→ Client가 Web Server에게 웹 문서를 요청할 때 Web Server가 웹 문서를 찾는 순서 지정

• AccessFileName .htaccess

→ 각각의 Directory에 대해 AllowOverride 지시자에 의해 주어진 옵션에 한하여 Directory의 접근 권한을 설정할 수 있는 파일을 지정

• HostnameLookups off

→ Apache에 접근한 Client의 hostname 또는 IP Address를 Log에 기록

→ on : Hostname을 Log에 기록

→ off : IP address를 Log에 기록

• ErrorLog logs/error\_log

→ Apache의 오류 로그 이름과 위치 지정

→ Server 작동 관련 문제 발생시 원인과 해결 방법 제공)



- LogLevel warn
  - error\_log 파일에 기록되는 에러 메시지의 레벨 지정
  - debug, info, notice, warn, error, crit, alert, emerg 값 중에서 지정
- LogFormat
  - combined 형식, 또는 common 형식 Log File Format 정의
  - combined 형식은 common 형식과 큰 차이는 없으며 %{Referer}i, %User-Agent}i 사용
  - %{Referer}i : HTTP 요청 헤더(Client가 참조한 페이지)
  - %User-Agent}i : Client가 사용한 Web Browser
- CustomLog logs/access\_log combined
  - Client가 Apache에 접근한 로그를 기록해주는 access\_log 파일의 위치 지정
- CustomLog logs/referer\_log referer
  - Client가 Apache에 요청한 페이지 정보 출력
- CustomLog logs/agent\_log agent
  - Client가 Apache에 접근할 때 사용한 Web Browser 정보 출력

#### - Section 3 : Virtual Hosts(가상 호스트에 대한 환경 설정)

##### 「Virtual Hosts 설정이란?」

- Web Server 한 대를 이용하여 다수의 도메인(Web Site)을 운영할 경우에 사용
- 이름 기반(Name-Based) 방식, IP 기반(IP-Based) 방식, 이름-IP 기반(Name & IP-Based) 방식 존재
  - 이름 기반 가상 호스트 방식 : 하나의 IP Address를 갖는 Web Server를 이용하여 다수의 웹 사이트를 운영
  - IP 기반 가상 호스트 방식 : 다수의 IP Address를 갖는 Web Server를 이용하여 다수의 웹 사이트를 운영
- 가상 호스트가 정상적으로 설정 되었는지 확인하는 방법 = httpd -S

- NameVirtualHost \*:80
  - 이름 기반(Name-Based) 방식에서만 사용하는 지시자이며 \* 대신 Web Server의 IP Address 입력
- <VirtualHost \*:80>
  - </VirtualHost>
  - Virtual Hosting 설정과 관련된 설정

##### 「Virtual Host 설정 지시자」

| 지시자          | 의미                              |
|--------------|---------------------------------|
| ServerAdmin  | 가상 호스트 관리자의 E-Mail 설정           |
| DocumentRoot | 웹 문서 Directory                  |
| ServerName   | 가상 호스트의 Domain Name             |
| ErrorLog     | 가상 호스트 도메인의 에러 로그 파일의 위치와 이름 지정 |
| CustomLog    | 가상 호스트 도메인의 접속 로그 파일의 위치와 이름 지정 |

## 16.3 Web Service Test

### I . Apache 환경 설정

1. Apache 환경 설정 파일 : /etc/httpd/conf/httpd.conf

- 
- Apache 환경 설정 파일은 크게 3가지 섹션으로 구성
    - ① 전역 환경 설정(아파치 서버 프로세서의 전반적인 동작을 제어하는 역할)
    - ② Main Server에 의해 사용되는 값(가상호스트 정의에 의해 처리되지 않는 모든 요청)
    - ③ 다중 웹(IP기반, hostname기반) 요청을 했을 경우의 가상 호스트를 위한 설정

## 2. ### Section 1: Global Environment (전역 환경 설정)

- ServerRoot "/etc/httpd"
  - ; 아파치가 정상적으로 동작하기 위해 필요한 여러 파일들(웹서버 설정 파일, 로그 파일, 바이너리 파일, 모듈들이 존재하는 디렉토리)의 위치를 설정
  - Apache를 소스 컴파일 설치할 경우는 설치 옵션에 따라서 경로가 달라지게 됨
- Timeout 120
  - ; 클라이언트의 요청을 받은 뒤 응답해 주기까지의 시간을 의미(Default : 120초)
  - 설정된 시간을 초과하게 되면 클라이언트는 접속이 해제되며, 서버로 클라이언트의 요청이 많을 시에는 시간을 적게 설정하는 것을 권장
- KeepAlive Off
  - ; 한 번 연결된 연결을 유지할 것인지 결정하는 지시자
  - 클라이언트는 연결 성공 후 서버에 여러 가지 요청을 수시로 할 수 있는데 빠른 응답을 할 수 있도록 KeepAlive 지시자를 활성화하는 것을 권장(Default : Off)
- MaxKeepAliveRequests
  - ; 하나의 연결에서 허용할 최대 요청 수를 지정하는 지시자(Default : 100)
  - KeepAlive 지시자가 Off 일 경우는 의미가 없으며 0으로 설정하면 제한이 없음
- Listen
  - ; Apache 서버를 특정 IP 주소와 포트에 대해 연결할 수 있도록 설정할 경우 사용하는 지시자
  - 예) 192.168.1.15의 IP를 갖는 클라이언트에게 8080포트를 이용하여 서비스 하고자 할 경우의 설정 --> Listen 192.168.1.15:8080
- User apache
- Group apache
  - ; 웹서버를 실행할 때 소유권을 갖게 되는 사용자와 그룹명을 지정한다. 사용자 소유권을 root로 하게 되면 보안상 위험하기 때문에 대개 nobody로 설정하지만, 일반적으로 로그인 불가능한 사용자 계정을 생성하여 이 계정(apache)과 그룹(apache)으로 지정하여 사용. 소스 설치 시 기본값은 daemon으로 설정되어 있으며, 이 값으로 사용하려면 계정과 그룹을 생성해 놓아야 함.

## 3. ### Section 2: 'Main' server configuration (메인 서버 환경 설정)

- ServerAdmin
  - ; 서버에 문제가 생겼을 경우, 시스템 관리자에게 E-Mail을 보낼 수 있도록 시스템 관리자의 E-Mail 주소 설정 --> Ex. ServerAdmin sysadmin@netcollege.co.kr
- ServerName
  - ; 클라이언트에게 다시 돌려 보내줄 호스트 이름을 설정
  - DNS에 등록된 이름을 사용해야 하며, 등록된 도메인이 없을 경우에는 IP주소 입력
- DocumentRoot
  - ; 아파치 웹서버가 서비스를 하기 위한 웹 문서가 저장된 디렉토리 지정
  - 클라이언트의 요청시 해당 디렉토리에서 요청받은 파일을 찾아 제공
  - (Default : /var/www/html)
- DirectoryIndex
  - ; 클라이언트가 웹 서버에 요청을 하게 되면, 웹서버는 웹 문서가 저장된 디렉토리에서

---

어떤 파일을 우선적으로 보여줄 것인지 설정

예) DirectoryIndex index.html index.htm index.php index.cgi main.html

- ErrorLog

; Apache 웹 서버의 오류 로그의 이름과 위치를 지정

예) ErrorLog logs/error\_log

#### 4. ### Section 3: Virtual Hosts (가상 호스트)

- 가상 호스트를 이용하기 위한 방법 : 이름 기반, IP 기반

- 가상 호스트 실습을 하기 위해서는 DNS 설정(/etc/named.conf, \*.zone 파일...)도 문제가 없어야 한다.

## II . Apache 가상 호스트 실습

### 1. 이름 기반의 가상 호스트 설정 실습

“3개의 웹사이트([www.white.co.kr](http://www.white.co.kr), [www.black.co.kr](http://www.black.co.kr), [www.blue.co.kr](http://www.blue.co.kr))를 이름 기반의 가상 호스트 설정을 통하여 운영할 경우 아래와 같은 과정 수행 “

(1) #NameVirtualHost \*:80

-> 주석을 제거하고 IP를 설정

예) NameVirtualHost 192.168.1.15:80

(2) #<VirtualHost \*:80>

```
ServerAdmin webmaster@dummy-host.example.com
DocumentRoot /www/docs/dummy-host.example.com
ServerName dummy-host.example.com
ErrorLog logs/dummy-host.example.com-error_log
CustomLog logs/dummy-host.example.com-access_log common
#</VirtualHost>
```

위 기본 값 내용을 아래와 같이 변경

```
<VirtualHost 192.168.1.15:80>
 ServerAdmin webmaster@white.co.kr
 DocumentRoot /var/www/html/white.co.kr
 ServerName www.white.co.kr
 ErrorLog logs/white.co.kr-error_log
 CustomLog logs/white.co.kr-access_log common
</VirtualHost>

<VirtualHost 192.168.1.15:80>
 ServerAdmin webmaster@black.co.kr
 DocumentRoot /var/www/html/black.co.kr
 ServerName www.black.co.kr
 ErrorLog logs/black.co.kr-error_log
 CustomLog logs/black.co.kr-access_log common
```

---

```
</VirtualHost>
<VirtualHost 192.168.1.15:80>
 ServerAdmin webmaster@blue.co.kr
 DocumentRoot /var/www/html/blue.co.kr
 ServerName www.blue.co.kr
 ErrorLog logs/blue.co.kr-error_log
 CustomLog logs/blue.co.kr-access_log common
</VirtualHost>
```

(3) 각 사이트별 디렉토리 생성 & 기본 웹문서(index.html) 저장

```
cd /var/www/html
mkdir white.co.kr --> white.co.kr 사이트의 index.html 저장
mkdir black.co.kr --> black.co.kr 사이트의 index.html 저장
mkdir blue.co.kr --> blue.co.kr 사이트의 index.html 저장
```

2. IP 기반의 가상 호스트 설정

“아래와 같은 조건을 갖는 3개의 웹사이트를 IP 기반의 가상 호스트 설정을 통하여 운영할 경우  
아래와 같은 과정 수행 “

◆ 조건 : [www.white.co.kr](http://www.white.co.kr) : 192.168.1.15  
          [www.black.co.kr](http://www.black.co.kr) : 192.168.1.16  
          [www.blue.co.kr](http://www.blue.co.kr) : 192.168.1.17

① 물리적인 하나의 랜카드에 여러 IP 설정하기(IP Alias)

```
ifconfig eth0:1 192.168.1.16 up
ifconfig eth0:2 192.168.1.17 up
ifconfig | grep in --> 설정 확인
```

② /etc/httpd/conf/httpd.conf 수정

- NameVirtualHost 지시자는 주석처리(이름 기반에서만 사용)  
-> #NameVirtualHost 192.168.1.15
- ServerName 192.168.1.15:80
- <VirtualHost 192.168.1.15:80>  
    ServerAdmin [webmaster@white.co.kr](mailto:webmaster@white.co.kr)  
    DocumentRoot /var/www/html/white.co.kr  
    ServerName www.white.co.kr  
    ErrorLog logs/white.co.kr-error\_log  
    CustomLog logs/white.co.kr-access\_log common  
    </VirtualHost>  
    <VirtualHost 192.168.1.16:80>

---

```
ServerAdmin webmaster@black.co.kr
DocumentRoot /var/www/html/black.co.kr
ServerName www.black.co.kr
ErrorLog logs/black.co.kr-error_log
CustomLog logs/black.co.kr-access_log common
</VirtualHost>
<VirtualHost 192.168.1.17:80>
ServerAdmin webmaster@blue.co.kr
DocumentRoot /var/www/html/blue.co.kr
ServerName www.blue.co.kr
ErrorLog logs/blue.co.kr-error_log
CustomLog logs/blue.co.kr-access_log common
</VirtualHost>
```

### III. Web Server 접속

#### (1) 일반 Web Server 접속

- Client System(Web Browser)에서 Web Server로 접근
  - `http://"Web Server IP" or http://"DNS 주소"`
    - DNS Server와 연동하면 Domain Name으로 접근 가능

#### (2) 다중 Web Server 접속

- Name-Based Virtual Hosting(이름 기반 가상 호스트)
  - `http://"DNS 주소"`

---

## Chapter 17. Samba Service

---

### 17.1 SAMBA Service 개요

#### I. SAMBA Service 소개

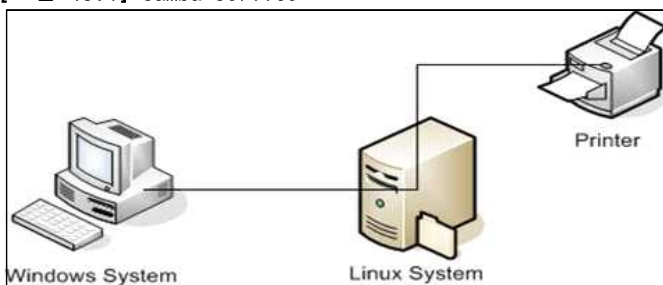
##### (1) Samba Service

- SMB, CIFS 프로토콜을 이용하여 Windows와 Linux의 파일 및 하드웨어(CD/DVD-ROM, 프린터, USB 등)를 공유할 수 있게 해 주는 서비스
- SMB(Server Message Block)
  - Microsoft와 Intel이 개발한 프로토콜
  - 서로 다른 시스템 상에 존재하는 디스크나 프린터와 같은 자원을 공유할 수 있도록 개발
  - TCP/IP 기반의 NetBIOS 프로토콜을 이용하기 때문에 NFS, NIS, Ipd 같은 유닉스의 분산 인증 구조와 유사
- CIFS(Common Internet File System)
  - SMB 프로토콜의 확장 버전으로 인터넷의 표준 파일 규약 프로토콜
  - SMB 프로토콜과 달리 여러 유닉스 업체가 참여하여 결정된 표준이기 때문에 안정성 향상

##### (2) Samba Service 특징

- Windows System과 Linux System간의 자원 공유
- 파일 공유뿐만 아니라 프린터 공유도 가능(Ex. Windows 사용자가 Linux 시스템에 연결된 프린터 사용 가능)
- SAMBA Service를 통한 NFS, FTP Service의 한계 극복 가능
  - NFS Service는 Linux와 Linux, Unix와 Unix 시스템 간에 파일 공유만 가능
  - FTP Service는 Windows와 Linux 시스템간의 파일 전송은 가능하지만 파일 전송 이외의 다른 서비스(Ex. Printer 공유)는 지원되지 않음. 또한 별도의 클라이언트 프로그램이 필요치 않음.

[그림 15.1] Samba Service



### 17.2 SAMBA Service 설치 및 설정

#### I. SAMBA Server 설치 및 smb 데몬 실행

##### (1) 설치 확인 및 설치

- [root@rocky ~]# rpm -qa | grep smb
- libsmbclient-(버전)

---

pam\_smb-(버전)  
gnome-vfs2-smb-(버전)

- [root@rocky ~]# yum install samba

(2) smb 데몬 실행

- /etc/rc.d/init.d로 이동 후, smb 데몬 실행 스크립트 파일 실행  
[root@rocky init.d]# ./smb start
- service 명령으로 데몬 실행하기 => yum 혹은 rpm으로 설치한 경우  
[root@rocky ~]# service smb start

(3) 서비스 활성화

- smb 데몬이 시스템 부팅시에 자동 실행될 수 있도록 chkconfig를 이용해 데몬툴에 등록

## II . SAMBA Server 설정

(1) SAMBA Server 주요 설정 파일

- /usr/sbin/smbd  
→ 사용자 인증과 Windows와의 자원 공유를 담당하는 SAMBA Server의 주요 데몬
- /usr/sbin/nmbd  
→ Windows의 WINS(NetBIOS Protocol) Service를 제공하는 데몬
- /etc/samba/smbusers  
→ Linux ID와 SAMBA ID가 다를 경우 이를 매칭하기 위한 매칭 테이블 파일
- /usr/bin/smbpasswd  
→ SAMBA 계정 생성, 삭제, 암호 설정 및 변경 명령어
- /usr/sbin/smbclient  
→ Windows의 공유 자원에 Linux가 접근할 때 사용하는 명령어
- /usr/bin/smbstatus  
→ 현재 SAMBA Server의 연결 상태를 확인하는 명령어

### ◆ SAMBA Server 환경 설정 방법

- /etc/samba/smb.conf 파일 수정
  - [root@rocky ~]# vi /etc/samba/smb.conf
    - SAMBA 설정 파일은 크게 [Global Setting], [Share Definitions] 두 영역으로 구분
    - Global Settings : SAMBA Service의 전체 설정을 정의하는 섹션
    - Share Definitions : SAMBA Service의 공유 설정을 정의하는 섹션
    - /etc/samba/smb.conf 설정 방법
- 시스템 도구를 이용한 수정
  - [root@rocky ~]# system-config-samba
    - 설정 파일 수정 방법보다 간단하고 쉽게 SAMBA Service 설정 가능

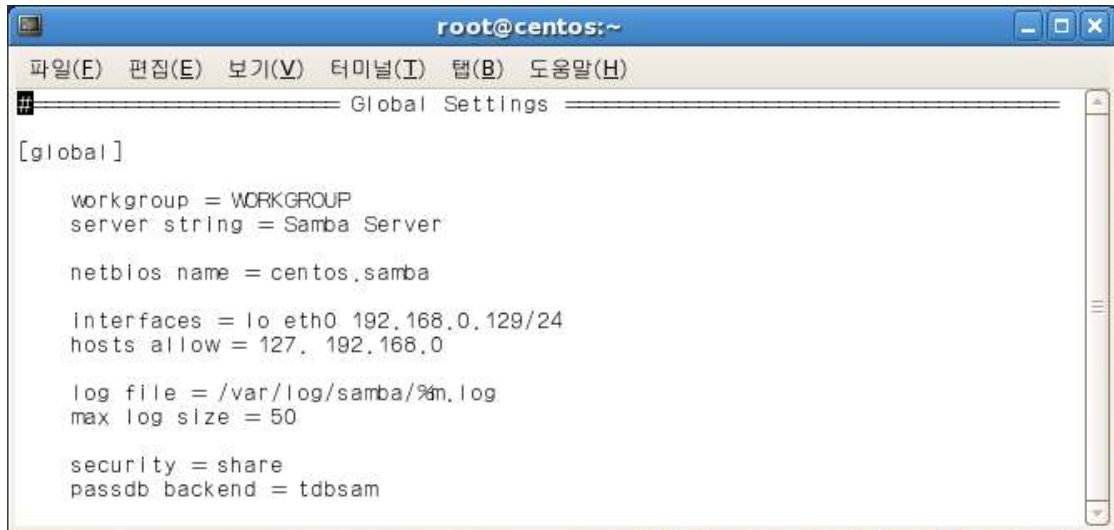
---

◆ /etc/samba/smb.conf

- ‘;’ 과 ‘#’ 으로 시작하는 라인은 주석처리를 의미  
→ 일반적으로 #은 설명을 위한 주석으로, ;은 실행(적용)을 막기 위한 주석으로 사용

- Global Settings 설정(SAMBA Service의 전체 설정을 정의)

[그림 15.2] /etc/samba/smb.conf 파일 Global Settings 부분 수정



- workgroup =
  - Windows 시스템의 작업 그룹 이름과 동일하게 설정
  - Windows 작업 그룹 이름 확인 : [내 컴퓨터] 우클릭 → [속성] → [컴퓨터 이름]
- server sting =
  - Windows “작업그룹 창” 에서 보여지는 SAMBA Server에 대한 설명
- netbios name =
  - Linux가 Windows의 Network에 참가하면서 사용하는 컴퓨터 이름
- interfaces =
  - Linux Server에 Network Interface가 여러 개일 경우 SAMBA Server가 사용할 Interface 지정
- hosts allow =
  - SAMBA Server로 접근을 허용할 hosts 지정
    - ① 192.168.0. : 192.168.0. 네트워크에 속해있는 모든 호스트들의 접속을 허용
    - ② 192.168.0.0/255.255.255.128 : 192.168.0.0 네트워크 속한 호스트들중 일부만 접속을 허용(192.168.0.1 ~ 192.168.0.126)
    - ③ client01, client02, client03 : /etc/hosts에 등록된 client01, client02, client03 호스트의 접속을 허용
    - ④ 192.168.0. EXCEPT 192.168.0.162 192.168.0.163 : 192.168.0.162, 192.168.0.163를 제외하고 192.168.0.0 네트워크에 속한 모든 호스트들의 접속을 허용
- log file =
  - SAMBA Server에 접속하는 시스템들의 접속 로그 파일 생성
- max log size =



- 로그 파일의 최대 크기를 설정(Kbyte),
- 0으로 설정하면 로그 파일의 크기를 제한하지 않음
- 설정된 크기로 로그 파일이 만들어지면 .old 확장자로 저장되고 새로운 로그파일을 기록

- security =

- SAMBA Server에 접속하고자 하는 모든 호스트들의 보안 레벨을 설정하는 항목

- ① user : 기본 값, Client의 접근 시 SAMBA Server에서 설정한 SAMBA 계정을 이용하여 접속
- ② share : username/password를 입력하는 인증 과정 없이 Client가 SAMBA Server에 접속
- ③ server : SAMBA Server로 접근 시 인증 과정을 별도의 인증 서버가 수행 실패할 경우 user 모드를 이용하여 인증을 시도
- ④ domain : SAMBA Server로 접근 시 인증 과정을 Windows의 Domain Controller가 수행
  - server 또는 domain 모드로 설정 시 아래와 같은 방법으로 password server 지시자 활성화
  - password server = <NT-Server-Name>

- ⑤ ads : Windows의 Active Directory Service를 이용하여 사용자 인증

- passwd backend =

- SAMBA 계정이 저장하는 방법을 지정

- ① tdbsam : /etc/passdb.tdb 파일에 삼바 계정의 열쇠글 저장
- ② smbpasswd : smbpasswd 파일에 삼바 계정의 열쇠글 저장 => 기존 방법
- tdbsam 환경에서 추가된 사용자는 smbpasswd 환경에서 관리가 어렵기 때문에 tdbsam 환경에서 추가된 사용자를 다 삭제한 뒤 smbpasswd환경에서 다시 추가해야한다

- Share Definitions 설정(SAMBA Service의 공유 설정을 정의하는 섹션)

[그림 14.3] /etc/samba/smb.conf 파일 Share Definitions 부분 수정 1



- [ ]

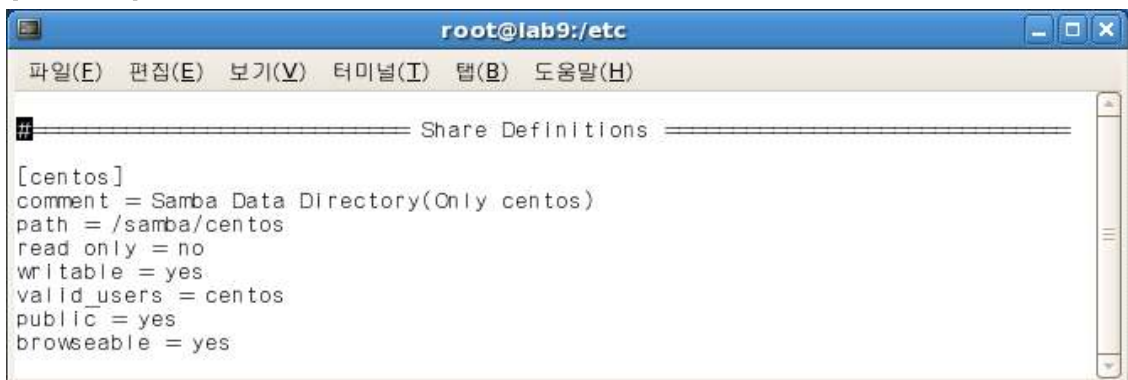
- SAMBA Server에 접근할 때 Client System이 요청하는 공유 이름 설정
- Ex. SAMBA Server 설정이 [public\_data] 일 경우, SAMBA Client의 요청 : WW192.168.0.160\public\_data

- comment =

- 공유에 대한 설명 입력

- path =  
→ SAMBA Server에서 실제로 공유하는 디렉토리의 경로 설정
- force =  
→ SAMBA가 share로 설정된 경우 공유 디렉토리에 접근한 사용자가 만드는 혹은 복사하는 파일의 소유권을 강제로 samba로 만들어줌. 그렇지 않으면 접근이 거부됨.
- browseable =  
→ 브라우저에 공유 이름 표시 여부 설정
- public =  
→ 모든 사용자에게 접근을 허용할 것인지에 대한 여부를 설정
- read only =  
→ SAMBA Client에게 읽기전용 권한을 줄 것인지에 대한 여부를 설정
- valid users =  
→ 특정 사용자에게만 접근을 허용할 경우에 사용자의 username 설정
- writeable =  
→ SAMBA Client에게 쓰기 권한을 줄 것인지 여부 설정

[그림 15.4] /etc/samba/smb.conf 파일 Share Definitions 부분 수정 2



→ Global Settings 부분의 security 모드를 user로 바꾼다.

#### ◆ SAMBA 서버의 상태 확인

- smbstatus 명령 : SAMBA 서버의 상태를 확인할 수 있다.

#### ◆ Windows(SAMBA Client) → Linux(SAMBA Server)

- Windows System에서 Linux의 공유 자원 접근
  - ① 공유 자원 생성
    - [root@rocky ~]# mkdir /data1
    - [root@rocky ~]# cd /data1
    - [root@rocky data1]# cat > testfile  
Linux SAMBA Server TEST  
^D
  - ② SAMBA 환경 설정 파일(smb.conf) 설정  
→ smb.conf 파일 수정 후 반드시 데몬 재실행
  - ③ SAMBA 계정 생성
- SAMBA 계정 관리

---

→ smb.conf 파일에서 security = user 설정일 경우 SAMBA 계정 필요

- [root@rocky ~]# useradd -s /sbin/nologin smbuser  
→ SAMBA 계정을 생성하려면 반드시 리눅스 쉘 사용자 계정이 존재해야 함.  
→ SAMBA 계정은 쉘 로그인 필요치 않기 때문에 -s /sbin/nologin 옵션 추가
- [root@rocky ~]# smbpasswd -a smbuser → SAMBA 계정 암호 생성
- [root@rocky ~]# smbpasswd -d smbuser → SAMBA 계정 비활성화
- [root@rocky ~]# smbpasswd -e smbuser → SAMBA 계정으로 활성화
- [root@rocky ~]# smbpasswd -x smbuser → SAMBA 계정 삭제

#### ◆ Windows(SAMBA Server) 폴더 공유

- Windows System의 C Drive에 share 폴더 생성
- share 폴더 공유
  - “ share 폴더 “ 우클릭 → “ 공유 및 보안 “ 선택 → 공유 탭에서  
“ 이 폴더를 공유(S) “ 선택
  - 특정 사용자에게만 접근을 허용하기 위해 공유 탭에서 “ 사용 권한(P) “ 선택  
후 계정 별 권한 설정  
→ 예) guest 사용자에게만 읽기 권한

#### ◆ Linux(SAMBA Client) → Windows(SAMBA Server)

- Linux System에서 Windows의 공유 자원 접근
  - ① smbclient 명령어 이용(명령어 위치 : /usr/bin/smbclient)
    - [root@rocky ~]# su - win01
    - [root@rocky ~]\$ smbclient //192.168.0.151/share  
→ 현재 Linux에서 Login한 계정으로 접근 시도  
Password :  
Domain=[WINDOWS01] OS=[Windows 5.1] Server=[Windows 2000 LAN Manager]  
smb: W>
    - [root@rocky ~]# smbclient -U win01 //192.168.0.151/share  
→ -U : 특정 사용자로 접근 시도
    - [root@rocky ~]# smbclient -L 192.168.0.151  
→ -L : SAMBA Server(Windows) 공유 정보 확인, IP대신 hostname 사용 가능
    - 윈도우 공유폴더 마운트하기  
→ 2.x 대 버전 : smbmount 명령 사용  
→ 3.011버전 이상 : 파일타입 cifs 사용하여 마운트  
mount -t cifs //192.168.0.151/share /마운트포인트
- SAMBA Server(Windows)에 접속한 후 get, put 등의 명령어를 이용하여 파일 전송 가능

---

## Chapter 18. NFS(Network File System)

---

### 18.1 NFS 개요

NFS란 네트워크를 통해 다른 호스트에 있는 파일시스템의 일부를 자신의 디렉토리처럼 사용할 수 있도록 해 주는 것으로 하나의 서버에 디스크를 집중관리하고 그것을 공유하여 나머지 시스템들이 사용할 수 있게 해준다.

- sun microsystem사에서 개발
- 분산처리환경의 RPC(Remote Procedure Call)를 사용하는 파일 및 자원 공유 시스템
- TCP/IP 네트워크를 통한 서버/ 클라이언트 구조
- UNIX 운영체제간의 파일 공유
- 보안에 취약하므로 주의해서 사용

#### 1. rocky 5에서 설정하기

- NFS 설치

rpm 명령어를 이용하여 NFS패키지들이 설치되어 있는지 확인.

```
[root@rocky ~]# rpm -qa portmap
```

```
portmap-4.0-65.2.2.1
```

```
[root@rocky ~]# rpm -qa nfs-utils
```

```
nfs-utils-1.0.9-40.el5
```

만약 설치되어 있지 않다면 `yum install portmap nfs-utils`를 실행하여 설치.

### 18.2 NFS 관련 데몬 및 서비스 실행

#### 1. 관련 데몬

NFS 서비스는 `nfsd`, `rpc.mountd`, `rpc.statd`, `rpc.lockd`, `rpc.rquotad` 데몬들이 구동된다.

NFS 서버 : `nfsd`, `rpc.mountd`, `rpc.statd`, `rpc.lockd`, `rpc.rquotad` 데몬이 작업

NFS 클라이언트 : `rpc.statd`, `rpc.lockd` 데몬이 작업

`rpc.nfsd` : rpc 요청처리와 서버측에서 `rpc.mountd`에 의해 마운트되면 클라이언트는 `rpc.nfsd`로 서버에 요청 처리와 명령 실행 등을 포함한 작업을 수행한다.

`rpc.mountd` : 클라이언트가 마운트를 요청하면 `/etc/exports` 파일설정에 따라 마운트 요청을 처리한다.

`rpc.statd` : NFS에 연결되어 있는 다른 시스템의 상태를 관리한다. `rpc.lockd`이 이 상태를 이용하기 때문에 함께 동작한다.파일잠금과 해제, 그리고 비정상적으로 종료시 복구를 담당한다.

`rpc.lockd` : 파일 잠금을 통해 클라이언트에서 여러 사용자가 한 파일을 수정하는 것을 방지한다.

`rpc.rquotad` : NFS파일 시스템을 마운트한 사용자에게 quota를 처리한다.

`portmap` : RPC(원격 프로세스 호출)기반하에서 작동되는 NFS 서비스의 매핑을 위해 사용된다.

---

## 2. 서비스 실행

- 서비스 실행하기 전 /etc/exports 파일에 공유할 디렉토리에 대해 설정

```
[root@rocky ~]# vi /etc/exports
```

```
[root@rocky ~]# /etc/init.d/portmap start 또는 service portmap start
portmap (을)를 시작 중: [OK]
```

```
[root@rocky ~]# /etc/init.d/nfs start 또는 service nfs start
```

```
NFS 서비스를 시작 중: [OK]
```

```
NFS 쿼터를 시작 중: [OK]
```

```
NFS 데몬을 시작 중: [OK]
```

```
NFS mountd를 시작 중: [OK]
```

- 부팅시 활성화하기 위해 chkconfig를 이용하여 portmap과 nfs를 활성화상태로 만들어 줌.

```
[root@rocky ~]# chkconfig portmap on
```

```
[root@rocky ~]# chkconfig nfs on
```

```
[root@rocky ~]# chkconfig --list | grep nfs
```

```
nfs 0:해제 1:해제 2:활성 3:활성 4:활성 5:활성 6:해제
```

```
nfslock 0:해제 1:해제 2:해제 3:활성 4:활성 5:활성 6:해제
```

- 공유 설정 내용 확인

```
[root@rocky ~]# exportfs -v
```

## 18.3 NFS 마운트 실습

### 1. NFS 서버 설정

- NFS를 통해 공유할 자원 설정 : /etc/exports 작성
- NFS서버의 주요 설정파일
- rpc.mountd, rpc.nfsd는 이 파일에 설정에 따라 구동

/etc/exports

형식

[디렉토리] [클라이언트](옵션)

```
[root@rocky ~]# vi /etc/exports
```

```
/nfsdata1 192.168.1.*/255.255.255.0(ro,sync)
```

->192.168.1.0/24 네트워크에 있는 호스트들에 대해서 읽기/쓰기, sync 옵션으로 설정

```
/nfsdata2 192.168.10.7(rw)
```

->192.168.10.7은 읽기/쓰기 옵션으로 설정

```
/nfsdata3 *.naver.com(rw,no_root_squash)
```

-> naver.com도메인을 사용하는 모든 호스트에 대해 읽기/쓰기, root권한으로 접근을 허용하는 설정

- 
- /etc/exports에서 사용 가능한 옵션
    - rw : 읽기/쓰기
    - ro : 읽기
    - secure : Mount시 1024 이하의 포트 사용
    - noaccess : 접속 거부
    - root\_squash : root 권한 획득 차단. nobody 권한
    - no\_root\_squash : Client root가 NFS Server에 접근시 root권한으로의 접근 허용
    - sync : NFS 서버에 파일 쓰기 완료 후 동기화
    - all\_squash : 서버와 클라이언트에서의 사용자를 별개의 권한으로 설정 (root를 제외)
    - no\_all\_squash : 서버와 클라이언트가 하나의 권한을 가짐 (root를 제외)
  - NFS 서버에서 공유할 디렉토리를 생성한다.
    - [root@rocky ~]# mkdir /nfs\_share1
    - [root@rocky ~]# mkdir /nfs\_share2
  - NFS 서비스 시작
    - [root@rocky ~]# service portmap start
    - [root@rocky ~]# service nfs start
  - exportfs
    - NFS서버의 exporting 정보를 출력하거나 설정 변경
    - exportfs [옵션]
    - [옵션]
      - v : export 정보 출력
      - r : 공유를 허가한 디렉토리를 다시 export
  - /var/lib/nfs/rmtab : NFS서버에 현재 마운트된 사용자 정보가 저장된 파일
    - # cat /var/lib/nfs/rmtab
  - rpcinfo : rpc관련 프로토콜이 사용하는 포트 현황 출력
- 사용법
- rpcinfo -p [host]

## 2. NFS 클라이언트 설정

- NFS 서버의 공유된 파일시스템 마운트하기
  - NFS 클라이언트에서 NFS 서버의 자원을 마운트 할때 사용
  - mount -t nfs [NFS\_server:/dir] [NFS\_client\_dir]
- 192.168.0.38 호스트에서 공유된 NFS를 마운트하기(공유된 서버가 192.168.0.200 이라면)
  - [root@rocky ~]# mkdir /nfs\_mount

---

```
[root@rocky ~]# mount -t nfs 192.168.0.200:/nfs_share1 /nfs_mount
```

- showmount

NFS서버의 마운트된 정보 출력

showmount [옵션] [host]

[옵션]

-a : 모든 마운트 정보보기

-e : NFS 서버의 export 리스트 보기

- 재부팅시 자동으로 마운트 해야 한다면 /etc/fstab에 추가

/etc/fstab

```
192.168.0.200:/nfs_share1 /nfs_mount nfs defaults 1 2
```

이렇게 설정하면 되는데 계속 nfs 연결을 시도하는데 일정시간 동안만 시도하고 없을 경우 시도를 멈추게 하려면 defaults 대신에 timeo=25,intr 해주면 25초간만 확인하고 넘어가면 그냥 부팅된다.

/etc/fstab에서 NFS 마운트 옵션

rsiz : NFS서버로부터 읽어 들이는 바이트 수 지정 기본값 1024byte

wsiz : NFS서버에 기록할 때 사용하는 바이트 수 기본 지정값 1024byte

timeo : 클라이언트에서 타임아웃이 발생되고 나서, 다시 재전송 요구를 보낼때 시간

retrans : 재전송 횟수 지정

port : NFS서버에 연결시 사용하는 포트번호 지정

fg : NFS서버에 타임아웃이 발생되면 즉각 접속 중지

intr : 인터럽트 허용

hard : 타임아웃 발생시 "server not recording" 에러메시지 표시후 계속 재시도

soft : 타임아웃 발생시 I/O 에러 표시

## 18.4 방화벽이 개방하기

- NFS 방화벽 개방

```
firewall-cmd --permanent --add-service=nfs (2049)
```

```
firewall-cmd --permanent --add-service=rpc-bind (111)
```

```
firewall-cmd --permanent --add-service=mountd (20048)
```

```
firewall-cmd --reload
```

- 리부팅 후에도 nfs를 사용하기 위해서는 /etc/fstab에 등록한다.

```
192.168.80.141:/nfs_share1 /mnt fg,soft 0 0
```

### 19.1 Mail Service 개요

#### I. Mail Service 소개

- Internet의 발달로 인해 Web Service와 함께 가장 많이 사용되는 서비스
- E-Mail(전자 우편) 송수신을 가능하게 해주는 서비스
- 상대적으로 시간적, 공간적 제약이 적음 → 사용자의 꾸준한 증가
- 다양한 분야에서의 현재의 대표적인 정보 전달 수단

#### II. Mail Service Protocol

- SMTP(Simple Mail Transfer Protocol)
  - 메일서버간 메시지를 전달하기 위해 사용되는 프로토콜
- POP3(Post Office Protocol 3)
  - 메일서버에 저장된 메시지를 아웃룩과 같은 프로그램을 이용해 사용자 컴퓨터로 가져오기 위해 사용되는 프로토콜.
- IMAP(Internet Message Access Protocol)
  - POP3와 달리 메일을 받아올때 서버에서 메일을 삭제하지 않고 보관하는 프로토콜

#### III. Mail System 구성요소

- MTA (Mail Transfer Agent)  
인터넷 상에 있는 어떤 컴퓨터로부터 다른 컴퓨터로 전자 메일을 전송하는 서버 프로그램  
대표적인 MTA : Sendmail, Qmail, PostFix
- MUA (Mail User Agent)  
사용자가 전자 메일을 송수신할 때 사용하는 클라이언트 프로그램  
대표적인 MUA : Outlook, Thunderbird
- MDA (Mail Delivery Agent)  
MTA가 수신한 메시지를 사용자의 우편함에 쓰기 위한 프로그램  
대표적인 MDA : /bin/mail이나 procmail
- MRA (Mail Retrieval Agent)  
리모트 서버에 있는 우편함으로부터 사용자의 MUA로 메시지를 가져오는 프로그램  
대표적인 MRA : Dovecot => POP3나 IMAP을 이용하여 구현



---

일반적인 메일의 전달 과정은 다음과 같다.

전자 메일은 사용자가 MUA를 이용하여 MTA로 보낸다. 이 송신 MTA는 메시지가 목적지에 도착할 때까지 메시지를 하나 혹은 그 이상의 MTA로 전송하게 되고, 목적지 MTA(수신 MTA)에 도달된 메시지는 사용자 정보에 따라서 MDA로 전송되어지며, MDA는 메시지를 수신자의 우편함에 쓴다. 수신자는 메시지를 읽기 위해서 자신의 메일 서버에 직접 접근하거나 MUA를 사용하여 자신의 메일 서버의 우편함으로부터 메시지를 로컬로 가져온다. 이 때 MUA가 메시지를 분류하거나 정렬하기도 한다. 수신자가 로컬이 아닌 리모트 메일 서버로부터 메일을 읽어 올 때 MRA를 사용한다.

MUA로부터 MTA까지는 SMTP(Simple Mail Transfer Protocol)을 사용하여 메시지를 보내며, MTA와 MTA 사이에도 SMTP가 사용된다. MUA는 MRA를 통하여 POP(Post Office Protocol)나 IMAP (Internet Message Access Protocol)을 주로 사용하여 메일 서버로부터 로컬로 메시지를 읽어 옵니다.

### III. Mail Server 설정

#### ◆ postfix 서버 설정

```
rpm -qa | grep postfix
```

```
rpm -qa | grep dovecot
```

설치확인 후 없으면 설치

```
dnf install postfix dovecot
```

#### <postfix 기본설정>

/etc/postfix/main.cf => 주 설정파일

##### 1) 메일 서버 호스트 설정

```
postconf -e 'myhostname = mail.kyt.com'
```

##### 2) 메일 도메인 설정

```
postconf -e 'mydomain = kyt.com'
```

##### 3) 발송자 도메인 설정

```
postconf -e 'myorigin = $mydomain'
```

##### 4) 리스닝 인터페이스 설정

```
postconf -e 'inet_interfaces = all'
```

##### 5) ipv4만 사용하도록 설정

```
postconf -e 'inet_protocols = ipv4'
```

##### 6) 메일을 받을 도메인 설정

```
postconf -e 'mydestination = $mydomain, localhost.$mydomain, localhost, $mydomain, /etc/postfix/local-host-names'
```

---

```
touch /etc/postfix/local-host-names
```

#### 7) SMTP 인증을 위한 설정

```
postconf -e 'smtpd_sasl_type = dovecot'
postconf -e 'smtpd_sasl_path = private/auth'
postconf -e 'smtpd_sasl_auth_enable = yes'
postconf -e 'smtpd_sasl_security_options = noanonymous'
postconf -e 'smtpd_sasl_local_domain = $myhostname'
postconf -e 'smtpd_client_restrictions = permit_mynetworks,permit'
postconf -e 'smtpd_recipient_restrictions = permit_mynetworks,permit_auth_destination,permit_sasl_authenticated,reject'
```

#### 8) Dovecot 설정 => 설치 yum install dovecot

```
vi /etc/dovecot/dovecot.conf
```

##### -프로토콜 설정

```
#protocols = imap pop3 lmtp => protocols = imap pop3
```

##### -리스닝 인터페이스 설정

```
#listen = *, :: => listen = *
```

##### -비밀번호 인증 설정

```
vi /etc/dovecot/conf.d/10-auth.conf
```

```
#disable_plaintext_auth = yes => disable_plaintext_auth = no
```

```
auth_mechanisms = plain => auth_mechanisms = plain login
```

##### -메일박스 위치 변경

```
vi /etc/dovecot/conf.d/10-mail.conf
```

```
#mail_location => mail_location = mbox:/var/empty:INBOX=/var/spool/mail/%u:INDEX=MEMORY
```

##### -SMTP 인증 설정

```
vi /etc/dovecot/conf.d/10-master.conf
```

##### # Postfix smtp-auth

```
unix_listener /var/spool/postfix/private/auth {
 mode = 0666
 user = postfix
 group = postfix
}
```

```
vi /etc/dovecot/conf.d/10-ssl.conf
```

```
ssl = no
```

---

9) 메일 서버 시작

```
systemctl restart postfix dovecot
```

10) 방화벽 개방

TCP/25(SMTP), TCP/110(POP3), TCP/143(IMAP)

11) 사용자 인증 테스트

- dovecot(pop3, imap)

```
telnet localhost 110
```

```
user admin
```

```
pass qwer1234
```

- postfix(smtp)

```
telnet localhost 25
```

```
ehlo naver.com
```

```
auth login
```

```
334 VXNlcm5hbWU6
```

```
YWRtaW4= => admin
```

```
334 UGFzc3dvcmQ6
```

```
cXdlcjEyMzQ= => qwer1234
```

```
mail -s "메일제목" 받는사람 이메일 => localhost 로 테스트
```

< 방화벽에서 25번 포트와 110번 143번 포트를 열어준다. >

```
[root@localhost]#
```

- 도메인명을 사용하기 위해서 dns 서버의 zone 파일에 mail 서버 설정

---

## Chapter 20. Bonding

---

### 20.1 Bonding 개요

본딩이란 여러장의 이더넷 카드를 묶어 하나의 ip를 부여하고 redundant link, fault tolerance 또는 load balancing을 구현하기 위한 방법. 예를들어 100Mbps 네트워크 카드를 두 개 묶어서 200Mbps의 대역폭으로 구성할 수 있다. 또한 네트워크 카드가 두 개이기 때문에 물리적인 경로 또한 두 개가 생긴다. 즉, 네트워크 이중화라고 할 수 있다.

### 22.2 Bonding 구현하기

#### 1. 네트워크 환경 구성 파일 설정

##### - ifcfg-bond0 추가 생성하기

```
cd /etc/sysconfig/network-scripts/
vi ifcfg-bond0
DEVICE=bond0
BOOTPROTO=none
BROADCAST=192.168.0.255
IPADDR=192.168.0.10
NETMASK=255.255.255.0
NETWORK=192.168.0.0
ONBOOT=yes
USERCTL=no
```

<= root가 아닌 일반계정이 해당 인터페이스를 비활성화 가능 여부

##### - ifcfg-eth# 내용 변경하기

```
vi /etc/sysconfig/network-scripts/ifcfg-eth0
DEVICE=eth0
ONBOOT=yes
BOOTPROTO=none
USERCTL=no
MASTER=bond0
SLAVE=yes
```

```
vi /etc/sysconfig/network-scripts/ifcfg-eth1
DEVICE=eth1
ONBOOT=yes
BOOTPROTO=none
USERCTL=no
MASTER=bond0
SLAVE=yes
```

```
vi /etc/sysconfig/network
NETWORKING=yes
HOSTNAME=bpan
GATEWAYDEV=bond0
```

==> 추가

---

GATEWAY=192.168.0.1 ==> 추가

#/etc/rc.d/init.d/network restart ==> 네트워크 서비스 재 시작

## II. 모듈 구성

### - bond 모듈 추가하기

# vi /etc/modprobe.conf ==> 내용 추가

alias bond0 bonding

options bond0 miimon=100 mode=1

bonding options

mode

0 : Balance-round robine

첫 번째 가능한 슬레이브부터 마지막까지 순차적으로 전송한다. 로드밸런싱과 Fault Tolerance를 제공한다.

1 : Active-backup

bond에서 하나의 슬레이브만 활성화된다. 다른 슬레이브는 활성화된 슬레이브가 장애일 경우에만 활성화 된다.

2 : Balance-XOR

(출발지 MAC 주소와 도착지 MAC 주소의 XOR)에 기초하여 전송한다. 이것은 도착지 MAC 주소에 대해서 동일한 슬레이브를 선택하게 된다.

3 : Broadcast

모든 슬레이브 인터페이스에 전송한다.

4 : 802.3ad (link aggregation) - switch에 aggregation 그룹생성이 필요하다.

5 : transmit load 밸런스로써 보내는 패킷만 밸런스 한다.

6 : receive와 transmit패킷을 밸런스 한다.

miimon

얼마나 자주 MII link를 감시할 것인가를 0.001초 단위로 설정한다. MII는 NIC가 살아있는지를 확인하는 데 사용된다.

# modprobe bonding ==> 모듈 올리기

# lsmod | grep bonding ==> 모듈 확인

bonding 68840 0

### - 네트워크 설정 확인하기

#ifconfig

bond0

Link encap:Ethernet HWaddr 00:08:02:28:FC:EC

inet addr:200.200.201.10 Bcast:200.200.201.255 Mask:255.255.255.0

UP BROADCAST RUNNING MASTER MULTICAST MTU:1500 Metric:1

RX packets:16722 errors:0 dropped:0 overruns:0 frame:0

TX packets:8125 errors:0 dropped:0 overruns:0 carrier:0

collisions:0 txqueuelen:0

RX bytes:8539251 (8.1 Mb) TX bytes:774392 (756.2 Kb)

---

```
eth0
Link encap:Ethernet HWaddr 00:08:02:28:FC:EC
inet addr:200.200.201.10 Bcast:200.200.201.255 Mask:255.255.255.0
UP BROADCAST RUNNING NOARP SLAVE MULTICAST MTU:1500 Metric:1
RX packets:2688 errors:0 dropped:0 overruns:0 frame:0
TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:1000
RX bytes:381527 (372.5 Kb) TX bytes:0 (0.0 b)
Interrupt:5 Memory:f7ef0000-f7f00000
```

```
eth1
Link encap:Ethernet HWaddr 00:08:02:28:FC:EC
inet addr:200.200.201.10 Bcast:200.200.201.255 Mask:255.255.255.0
UP BROADCAST RUNNING SLAVE MULTICAST MTU:1500 Metric:1
RX packets:14034 errors:0 dropped:0 overruns:0 frame:0
TX packets:8125 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:1000
RX bytes:8157724 (7.7 Mb) TX bytes:774392 (756.2 Kb)
Interrupt:3 Memory:f7ee0000-f7ef0000
```

```
- bond0 동작 확인하기
cat /proc/net/bonding/bond0
```

#### ✓ 본딩 설정하기

1. NIC 추가하고 nmcli 명령으로 시스템에서 인식할 수 있도록 추가

```
nmcli con add type ethernet con-name ens32 ifname ens32
```

2. bond0 인터페이스를 nmcli 명령으로 추가

```
nmcli con add type bond con-name bond0 ifname bond0
```

3. NIC 설정파일 내용 수정

```
#cd /etc/sysconfig/network-scripts
```

```
vi ifcfg-bond0
```

```
DEVICE="bond0"
TYPE="Bond"
NAME="bond0"
ONBOOT="yes"
BOOTPROTO="none"
IPADDR="192.168.80.206 "
NETMASK="255.255.255.0 "
GATEWAY="192.168.80.2 "
DNS1="8.8.8.8 "
DNS2="168.126.63.1 "
BONDING_MASTER="yes"
BONDING_OPTS="mode=1 miimon=100 fail_over_mac=1 primary=ens32 "
```

```
ifcfg-ens32
```

---

```
DEVICE=ens32
NAME=ens32
BOOTPROTO=none
MASTER=bond0
SLAVE=yes
DEFROUTE=no
ONBOOT=yes
```

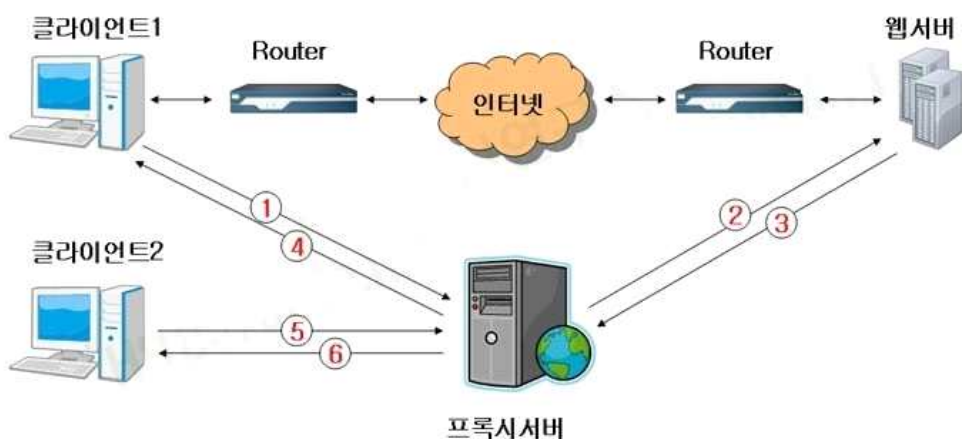
```
ifcfg-ens34
```

```
DEVICE=ens34
NAME=ens34
BOOTPROTO=none
MASTER=bond0
SLAVE=yes
DEFROUTE=no
ONBOOT=yes
```

4. `systemctl restart NetworkManager`
5. `reboot`
6. `cat /proc/net/bonding/bond0`

### 21.1 Proxy Server 개요

프록시 서버란 프록시란 단어가 의미하듯 중간에서 어떤 역할을 대행하는 서버를 의미한다. 웹 환경에서 프록시 서버는 웹 클라이언트와 웹 서버사이에서 요청한 데이터를 전달하는 역할을 한다. 즉, 클라이언트에서 인터넷상의 웹서버에 웹페이지를 요청하게 되면 프록시 서버가 그 역할을 대행하고, 받아온 웹페이지를 클라이언트에게 보내준다. 이 후 같은 페이지요청시 프록시 서버가 웹서버를 대행하게 된다. 따라서 그만큼 웹 서핑속도가 향상된다.



[그림 17.1] 프록시 서버 개념도

프록시 서버가 구축되어 있지 않은 상황에서 클라이언트들은 라우터를 통해서 인터넷을 이용하게 된다. 그런데 프록시 서버를 구축하고 클라이언트 PC에서 “프록시 서버” 사용을 체크하게 되면, 클라이언트들은 프록시 서버를 통해서 인터넷을 사용하게 된다.

과정을 위 그림을 통해서 살펴보면,

- ① 클라이언트1이 프록시서버에게 웹페이지를 요청한다.
- ② 클라이언트의 요청을 받은 프록시 서버는 요청받은 내용을 캐시에서 확인하고, 없으면 해당 웹 서버에 요청하게 된다.
- ③ 웹 서버로부터 해당 웹페이지를 전송받는다.
- ④ 프록시 서버는 웹페이지를 캐시에 저장한 후, 요청한 클라이언트에게 전송한다.
- ⑤ 클라이언트2가 같은 웹페이지를 요청한다.
- ⑥ 프록시 서버는 캐시에 저장된 웹페이지를 클라이언트2에게 전송한다.

이와 같은 방식으로 클라이언트들의 웹 서핑 속도가 향상되는 것이다.

### 22.2 Squid Proxy 서버 설치 및 설정

squid는 리눅스에서 프록시 서버를 구축할 수 있도록 해주는 오픈소스 프로그램이다.

<http://www.squid-cache.org> 에서 다운로드 할 수 있으며, rocky에는 기본적으로 설치되어 있다.



---

## I . Squid Server 설치 및 squid 데몬 실행

### (1) 설치 확인 및 설치

- [root@rocky system]# rpm -qa | grep squid  
squid-4.15-3.module+el8.6.0+1044+67ab5d0a.2.x86\_64
- 설치가 되어 있지 않다면 rpm이나 yum을 이용해서 설치한다.
- [root@rocky ~]# dnf install squid

### (2) Squid Server 설정

- [root@rocky ~]# vi /etc/squid/squid.conf  
576행 acl rocky55 src 192.168.1.0/255.255.255.0  
615행 http\_access allow manager rocky55  
636행 http\_access allow rocky55  
1579행 cache\_mem 1000 MB  
3005행 visible\_hostname rocky55 => error message 등에서 나타나게 하고싶은 호스트명

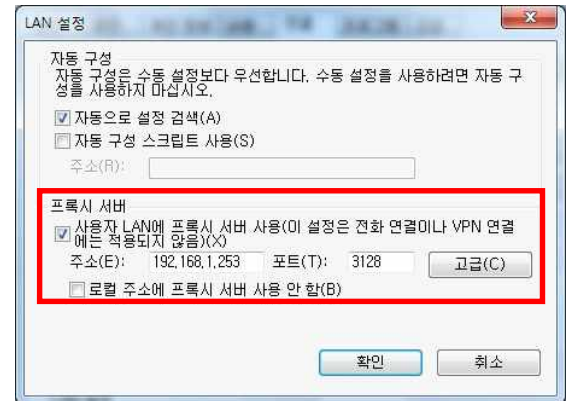
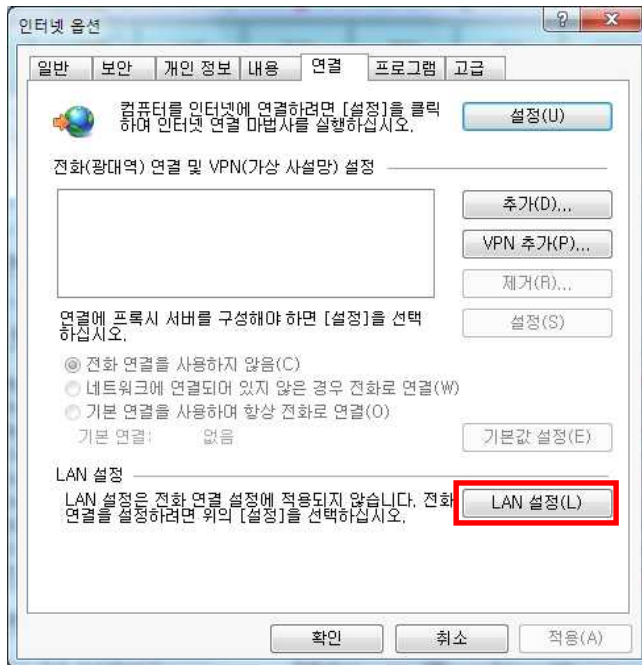
### (3) 서비스 활성화

- [root@rocky ~]# service squid start  
→ Proxy Service 활성화

### (4) 방화벽 포트 열어주기

- squid가 사용하는 포트가 TCP 3128이므로 방화벽에서 해당 포트를 열어준다.
- [root@rocky ~]# firewall-cmd --permanent --add-port=3128/tcp
- [root@rocky ~]# firewall-cmd --reload

(5) Internet Explorer에서 프록시 연결 설정하기



이상과 같이 모든 설정이 끝난 후, 파일 전송 테스트를 해본다. 첫 번째 클라이언트에서 인터넷을 통해 파일을 다운로드 한 후, 다른 클라이언트에서 같은 파일을 다운받게 되면 다운로드 시간이 대폭 줄어든다. 왜냐하면 로컬에 있는 프록시 서버로부터 파일을 가져오기 때문이다.