HOSTWAY
GLOBAL IT SERVICE PARTNER

Hostway CentOS 5.5 매뉴얼

Date: 2010.08.16

Hostway IDC Corporation

목 차

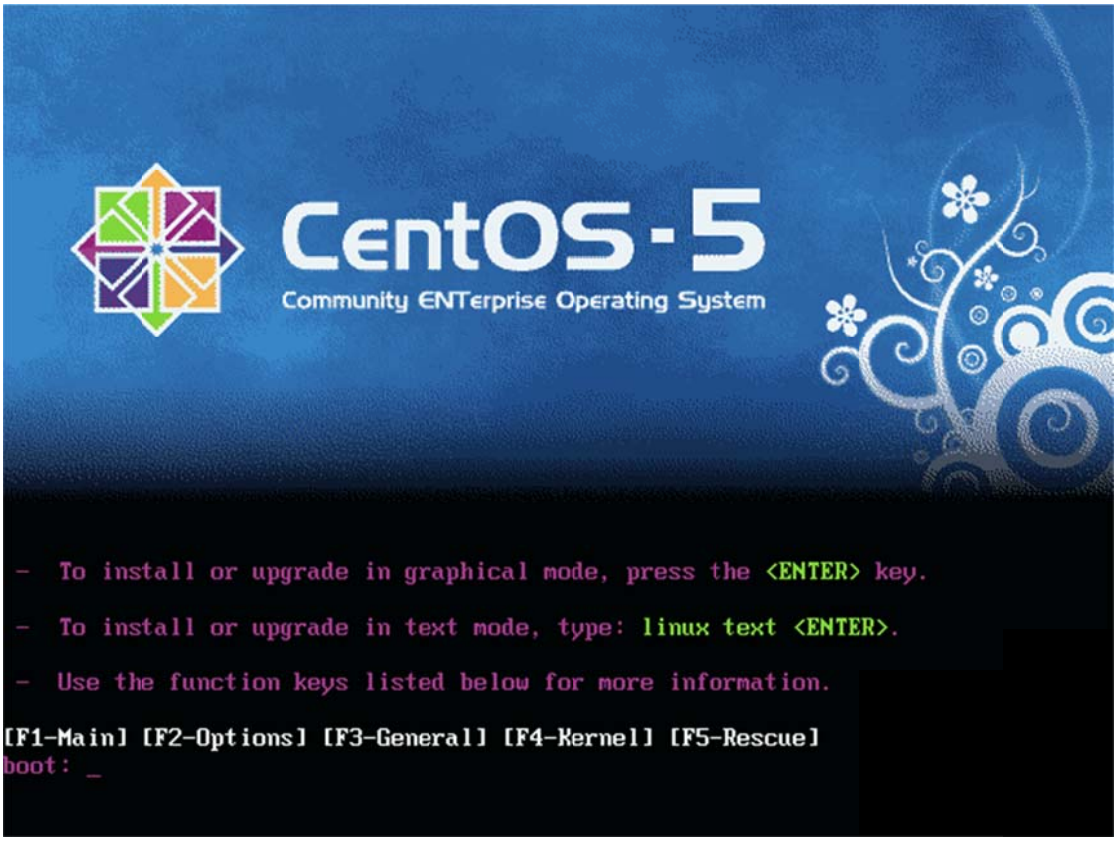
Chapter 1. 서버용 리눅스 설치

- 설치

서버 용 리눅스 설치 및 서버 구성은 Cent5.4 를 기반으로 설정되었으며, 일반적인 RedHat 계열의 운영체제에서는 동일한 방법으로 서버를 구성할 수 있습니다.

Linux CD에서 제공되는 패키지는 모두 설치 할 경우, 설치가 편리하고 여러 패키지를 볼 수 있는 장점이 있지만, 설치 시간이 오래 걸리고 디스크 공간을 많이 차지하며, 불필요한 데몬들로 인해 보안상으로 취약하게 됩니다. 서버용으로 Linux를 사용할 것이기 때문에 서버 운영상 필요한 패키지만 선택해서 설치 하도록 하겠습니다.

- 리눅스 CD 또는 DVD로 부팅되는 설치 초기 화면 입니다.
  Text / 그래픽 /network 등 다양한 설치 방법이 존재 하지만 [Enter] 키를 눌러 그래픽모드 진행 합니다.



[그림 1-1] 설치 초기 화면

- Media Test 단계 입니다.설치할 Media 에 대한 오류 검사 Test 여부를 묻는 화면입니다. [Skip] 을 선택하고, 다음으로 넘어 갑니다.

[그림 1-2] Media Test

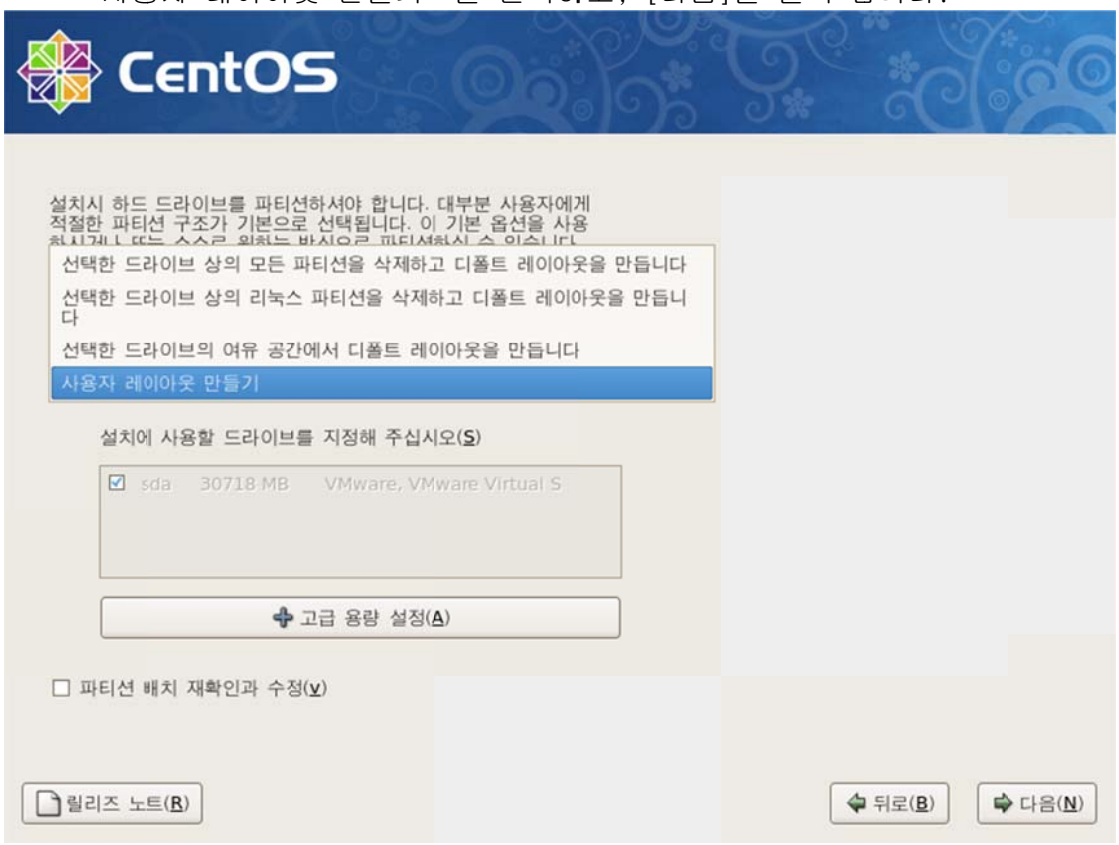
- 설치 언어 선택 화면 입니다. “Korean(한국어)” 을 선택한 후에 [Next]를 클릭합니다.

[그림 1-3] 설치 언어 선택

- 시스템에 사용할 키보드 자판 배열 선택하는 단계 입니다. Default “U.S 영어”를 선택하고, [다음]을 클릭 합니다.

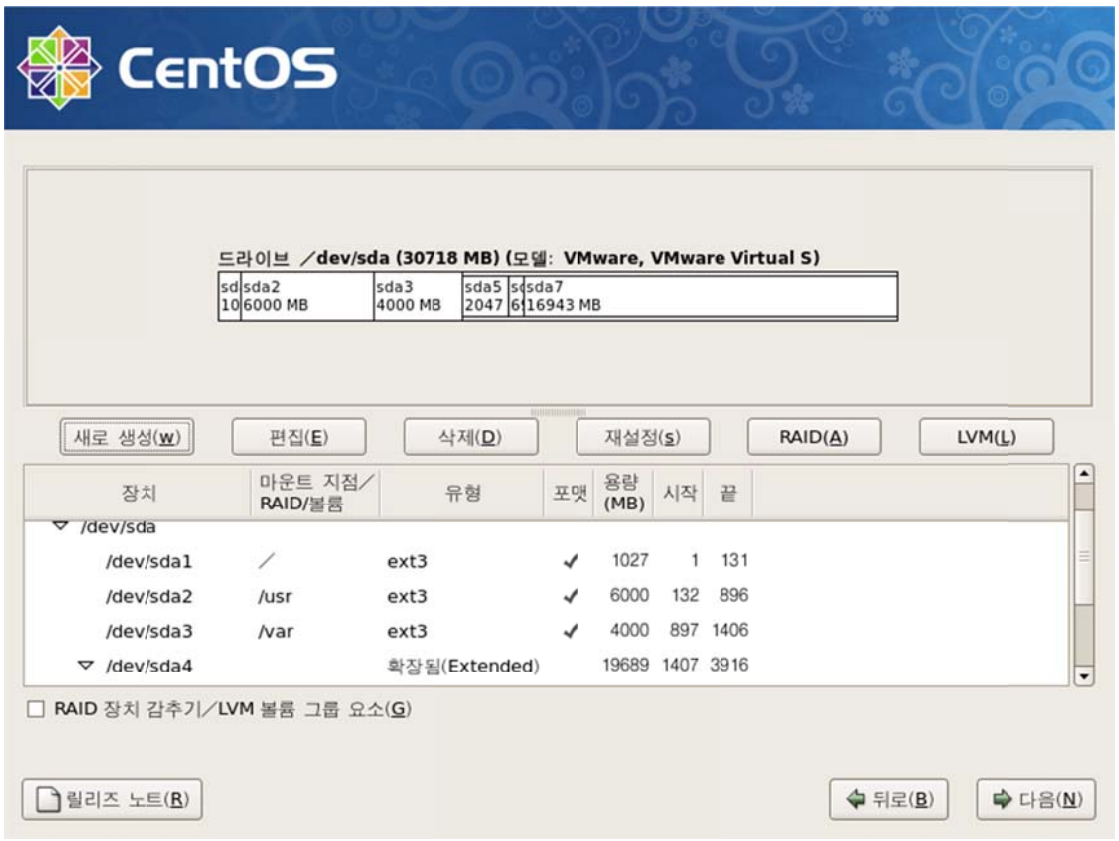
[그림 1-4] 키보드 자판 배열 선택

- 하드디스크 파티션 분할 방식 설정하는 단계 입니다.” 사용자 레이아웃 만들기”를 통하여 사용자가 원하는 파티션 및 사이즈로 분할이 가능 합니다.
  “사용자 레이아웃 만들기”를 선택하고, [다음]을 클릭 합니다.



[그림 1-5] 파티션 분할 방식 선택

- 파티션 분할을 완료한 화면입니다. 디스크를 선택하고, [새로 생성]을 클릭하여 마운트 포인트를 입력하고 파티션을 분할합니다.
-



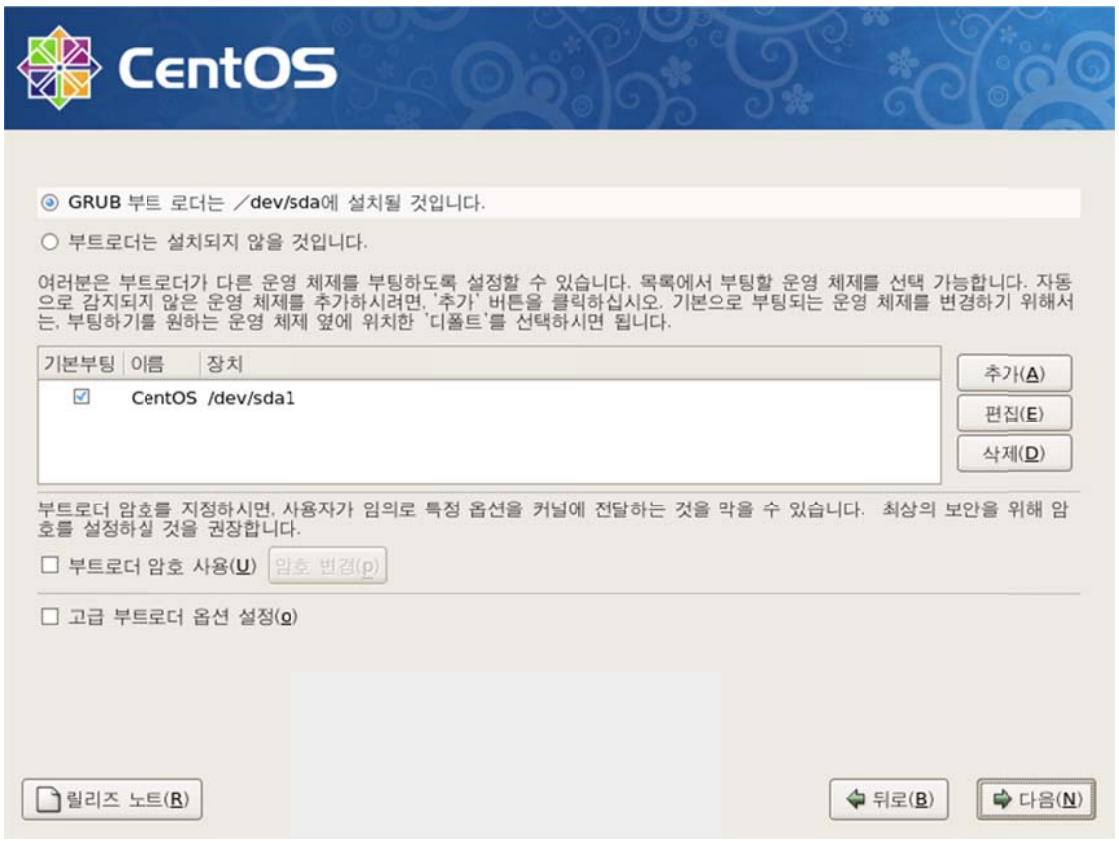
[그림 1-6] 파티션 분할

서버용으로 운영체제를 설치하기 위해서는 기본적으로 다음과 같이 파티션을 분할해서 설치할 것을 권장합니다. 운영할 서비스의 종류에 따라 파티션 사이즈는 적절하게 분할합니다.

[표 1.1] 서비스 종류에 따른 파티션 사이즈

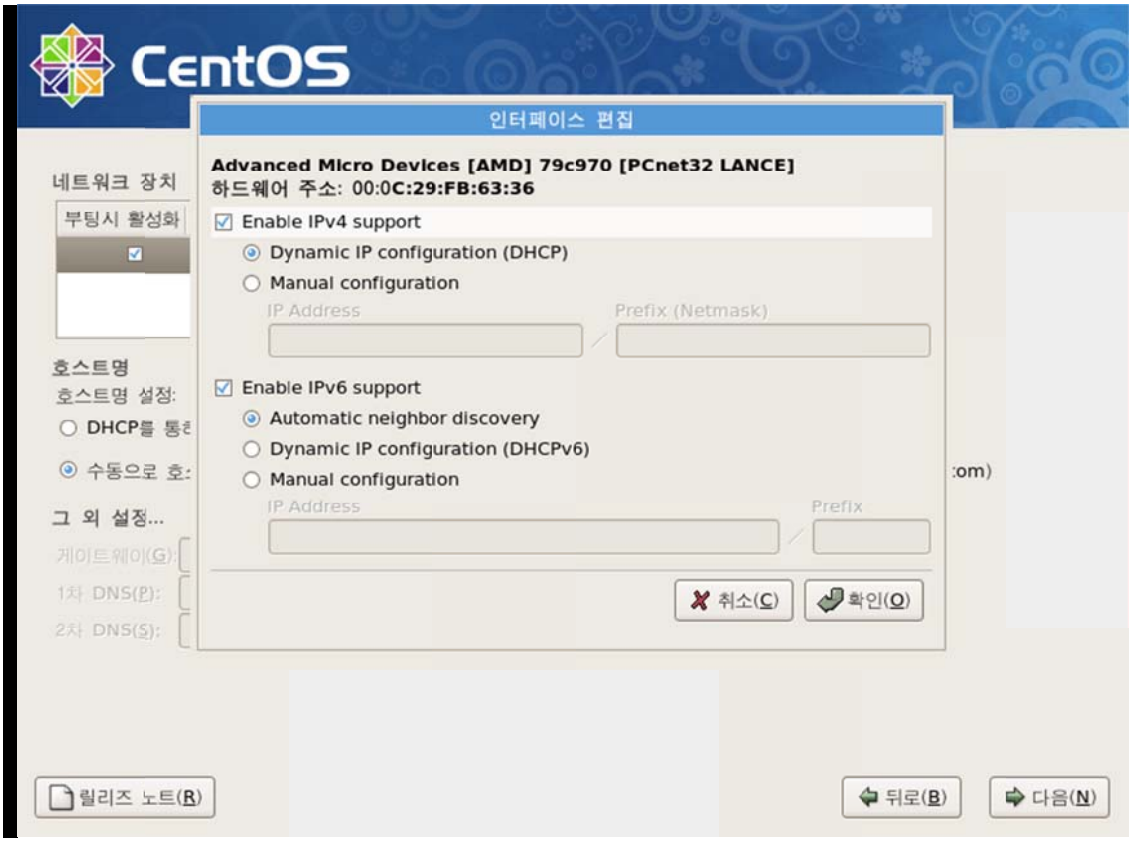
서비스 종류	파티션 사이즈
/	1G 이상
/usr	5G 이상
/var	3G 이상
swap	Memory * 2
/tmp	500M 이상
/home	나머지 공간

- 부트로더 설정 단계입니다. Default 설정으로 GRUB가 사용됩니다. Default 설정 그대로 사용하고, [다음]을 클릭합니다.



[그림 1-7] 부트로더 선택

- 서버에서 사용할 네트워크 정보를 입력하는 단계입니다.
  “수동으로 호스트 설정”을 체크한 후에 IP, NETMASK, GATEWAY 등을 입력한 후에 사용할 DNS 서버 정보를 차례로 입력하고, [다음]을 클릭합니다.



[그림 1-8] 네트워크 설정

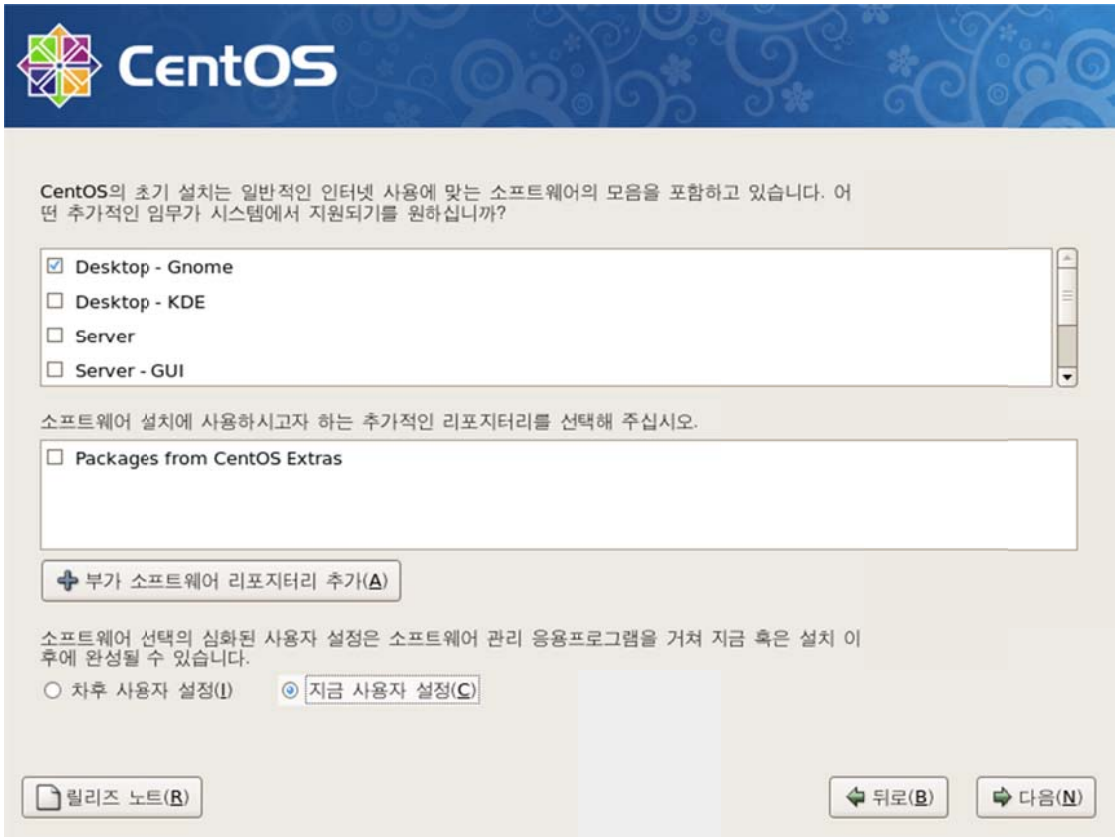
- 서버에서 사용될 시간대를 선택하는 단계입니다. “아시아/서울(Seoul)”을 선택하고 [다음]을 클릭합니다.

[그림 1-9] Time Zone 설정

- 루트 계정(서버 관리자)의 암호를 입력하는 단계입니다. 암호는 6자 이상, 숫자나 기호를 포함해서 사용할 것을 권장합니다.
  암호를 알맞게 두 번 입력 한 후, [다음]을 클릭합니다.

[그림 1-10] 루트 암호 설정

- 패키지를 선택하는 단계입니다. “지금 사용자 설정”을 선택한 후, [다음]을 클릭합니다.



[그림 1-11] 패키지 선택

- 패키지 사용자 설정 세부 선택 단계입니다. 아래의 최소 필수 패키지 이외에 필요한 패키지는 추가로 체크 합니다.
  - 응용프로그램 : 편집기
  - 개발용 도구 : 개발용 도구 / 개발용 라이브러리 /레거시 소프트웨어 개발
  - 기반 시스템 : 관리도구 / 기본 / 시스템도구
  - 언어 지원 : 한국어 지원(Koean)

  모두 선택이 완료되면 [다음]을 클릭합니다.

[그림 1-12] 패키지 사용자 설정 세부 단계

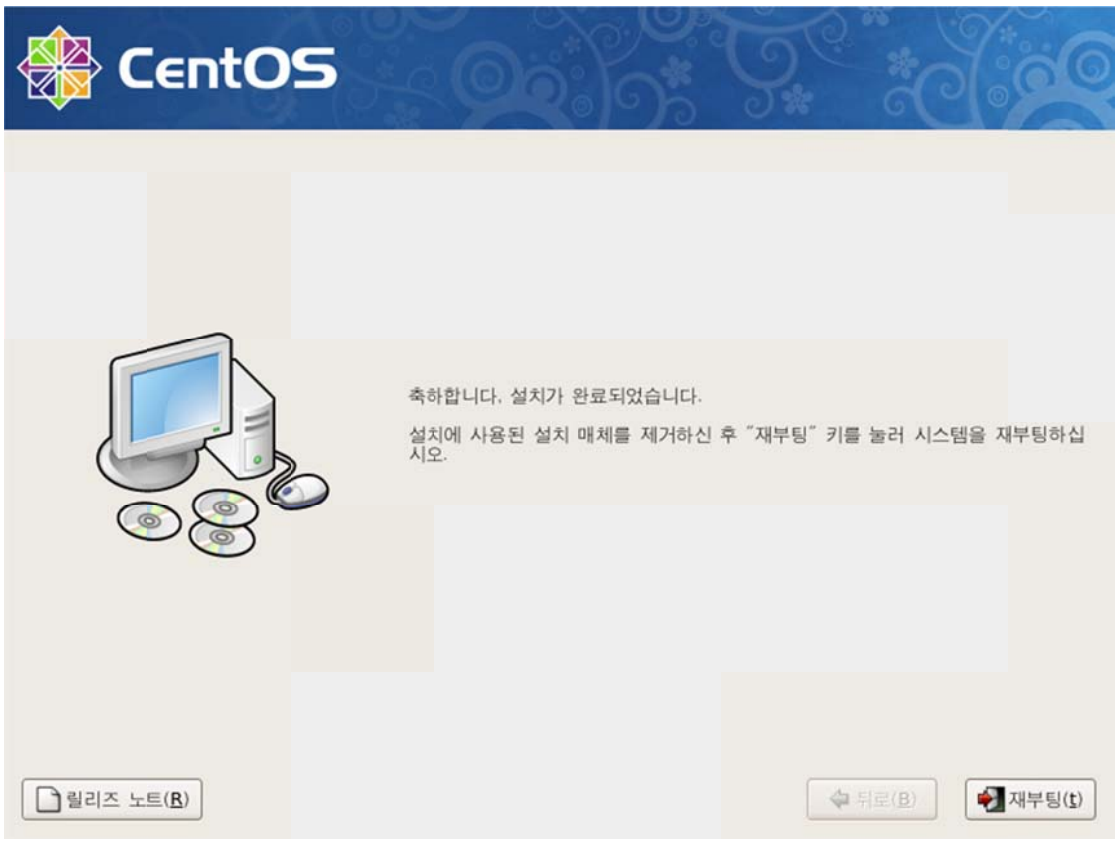
- 패키지 의존성 검사 수행 단계 입니다.

[그림 1-13] 패키지 의존성 검사

- 설치 준비가 완료된 상태입니다. 하드 디스크 파티션을 포맷하고, 패키지 설치를 진행하게 됩니다.
  이때는 설치 된 패키지 등에 따라 설치 소요 시간이 달라질 수 있습니다.

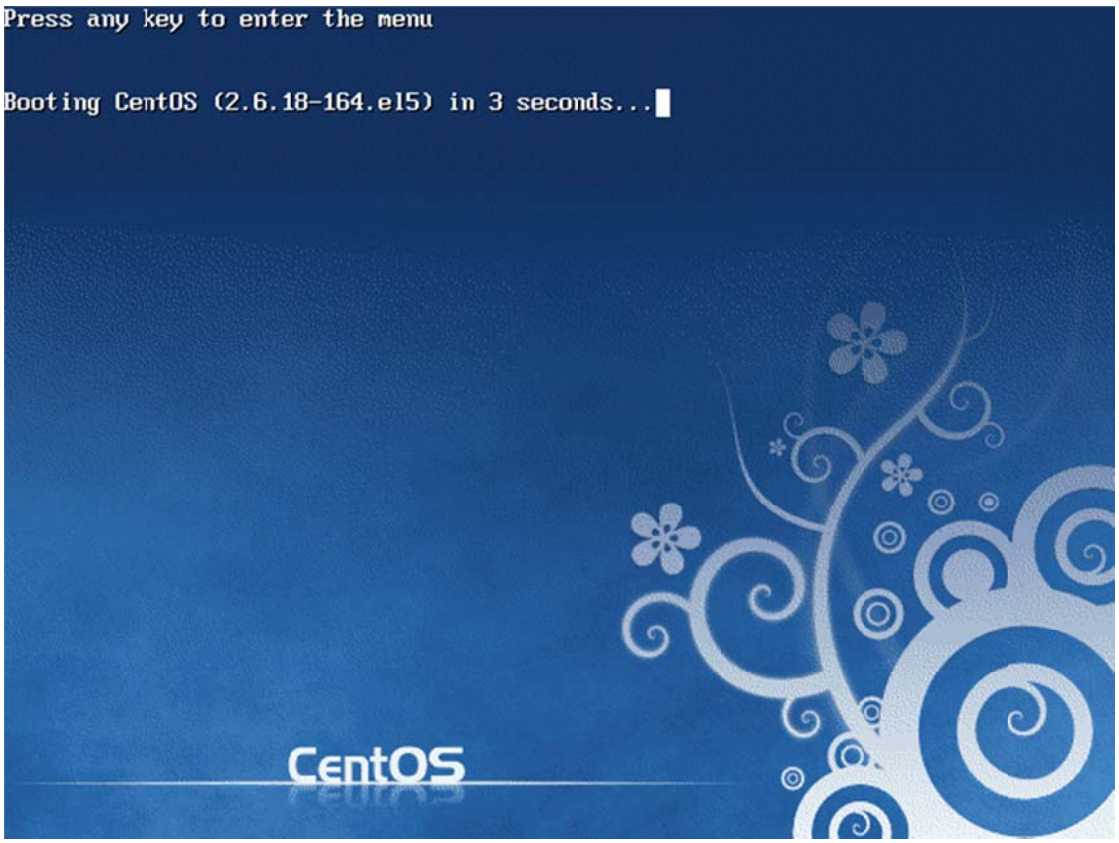
[그림 1-14] 설치 시작

- 설치가 완료된 화면입니다.[재부팅]을 클릭하면 시스템 리부팅이 진행됩니다. 하드 디스크로 부팅하기 위해서 Linux Media를 제거하고, 부팅을 진행합니다.



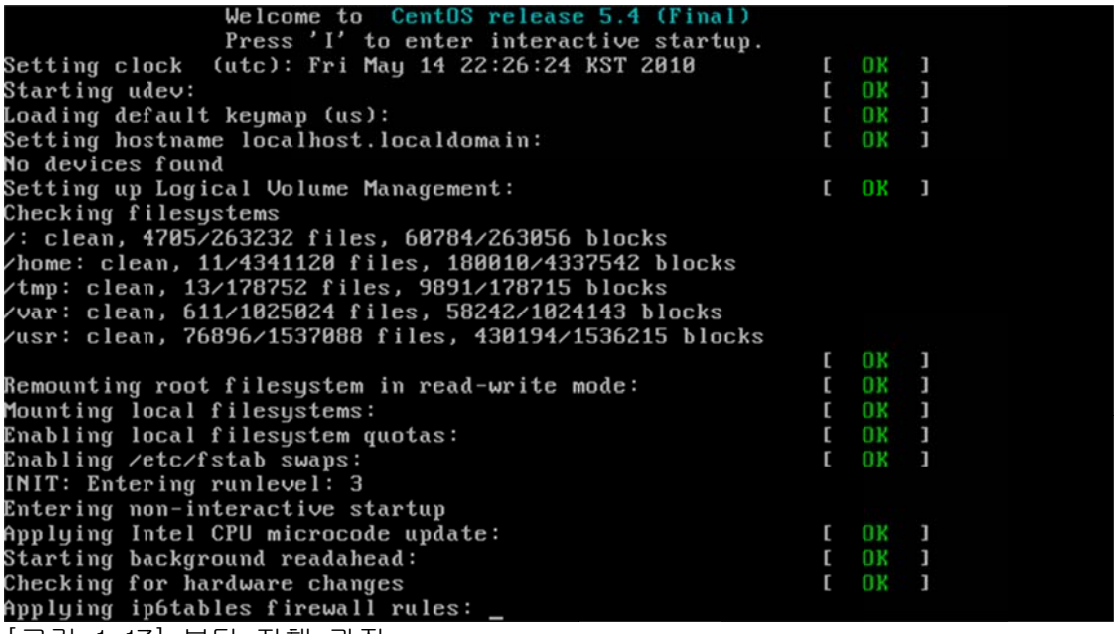
[그림 1-15] 설치 완료

- 설치 완료 된 Linux로 부팅이 진행됩니다. GRUB 부트로더 화면입니다.



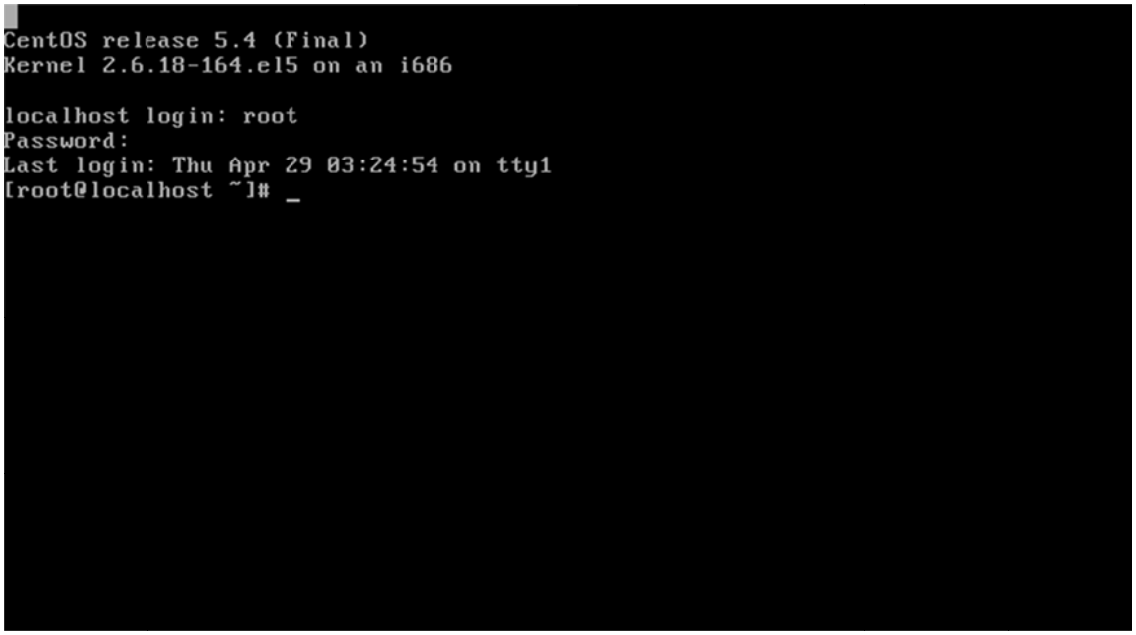
[그림 1-16] GRUB 부트로더

- Linux로 부팅이 진행되는 과정입니다.



[그림 1-17] 부팅 진행 과정

- 부팅이 완료되어 root 계정으로 로그인 한 화면 입니다.



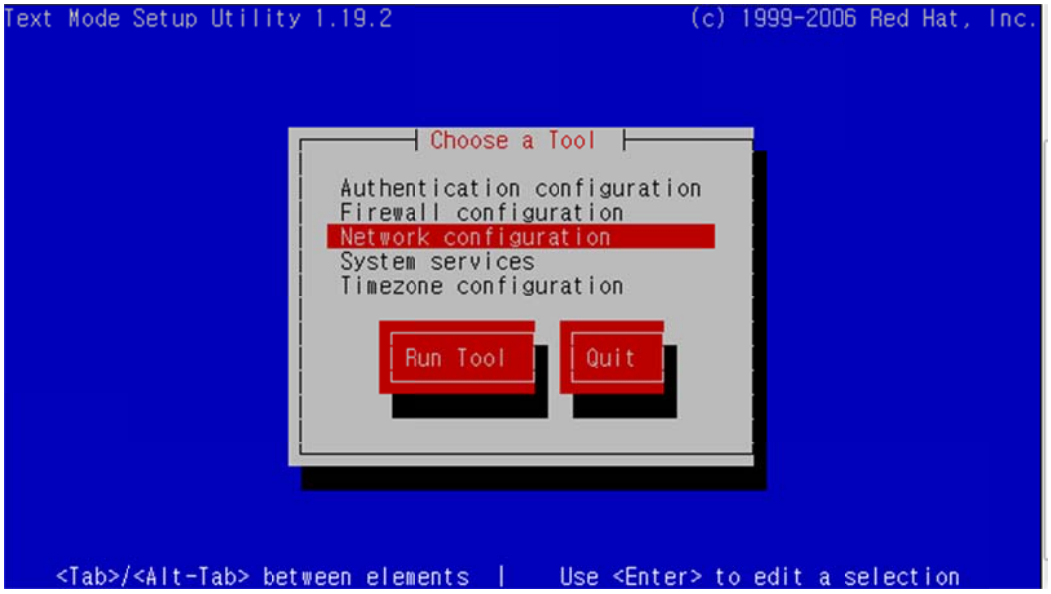
[그림 1-18] 로그인 화면

- 네트워크 환경 설정

Linux가 설치된 OS에 네트워크를 설정하는 여러 가지의 방법이 존재합니다. 각각의 방법에 대하여 알아 보겠습니다.

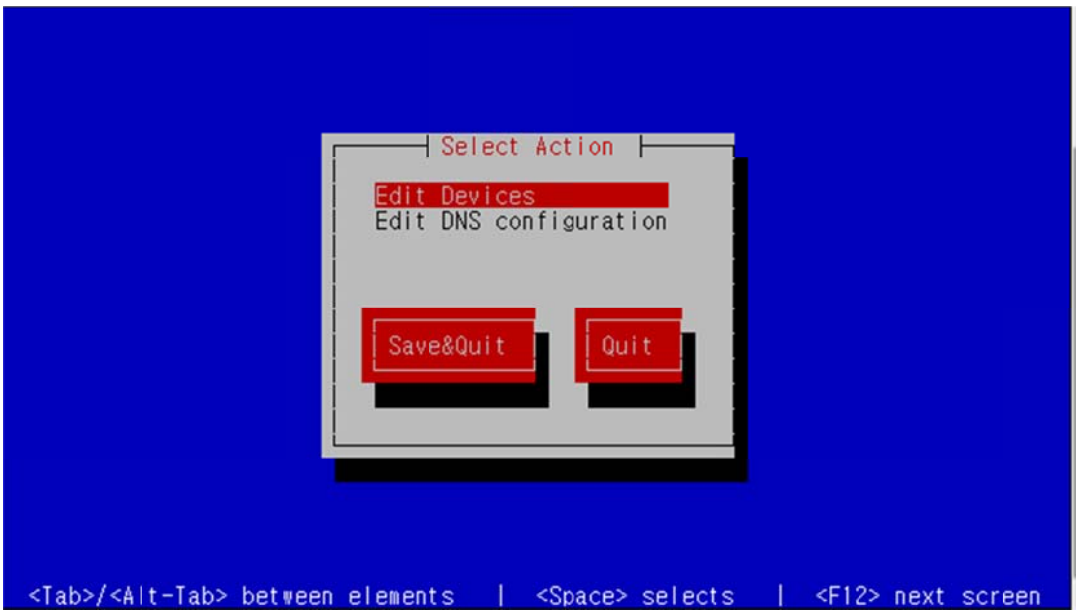
2.1 Setup 명령을 통한 네트워크 설정 방법

- 서버에 root로 로그인 한 상태에서 "setup" 입력 후 [enter]를 누르면 아래와 같은 화면이 출력됩니다.
  "Network Configuration"을 선택 후 [Run Tool] 을 선택합니다.
  메뉴 이동은 [tab] 버튼을 이용하여 이동하며 선택은 [space bar]를 누르면 선택 됩니다.



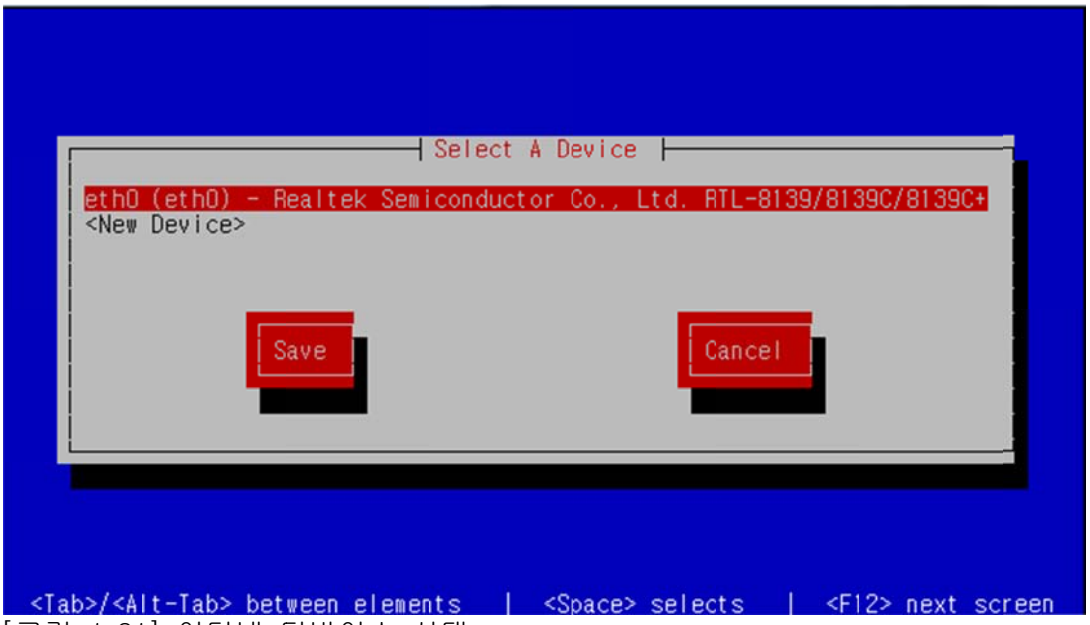
[그림 1-19] setup

- 디바이스 수정이나 DNS Configuration 을 수정하는 단계입니다. “Edit Devices”를 선택한 후, [Save&Quit]를 선택합니다.



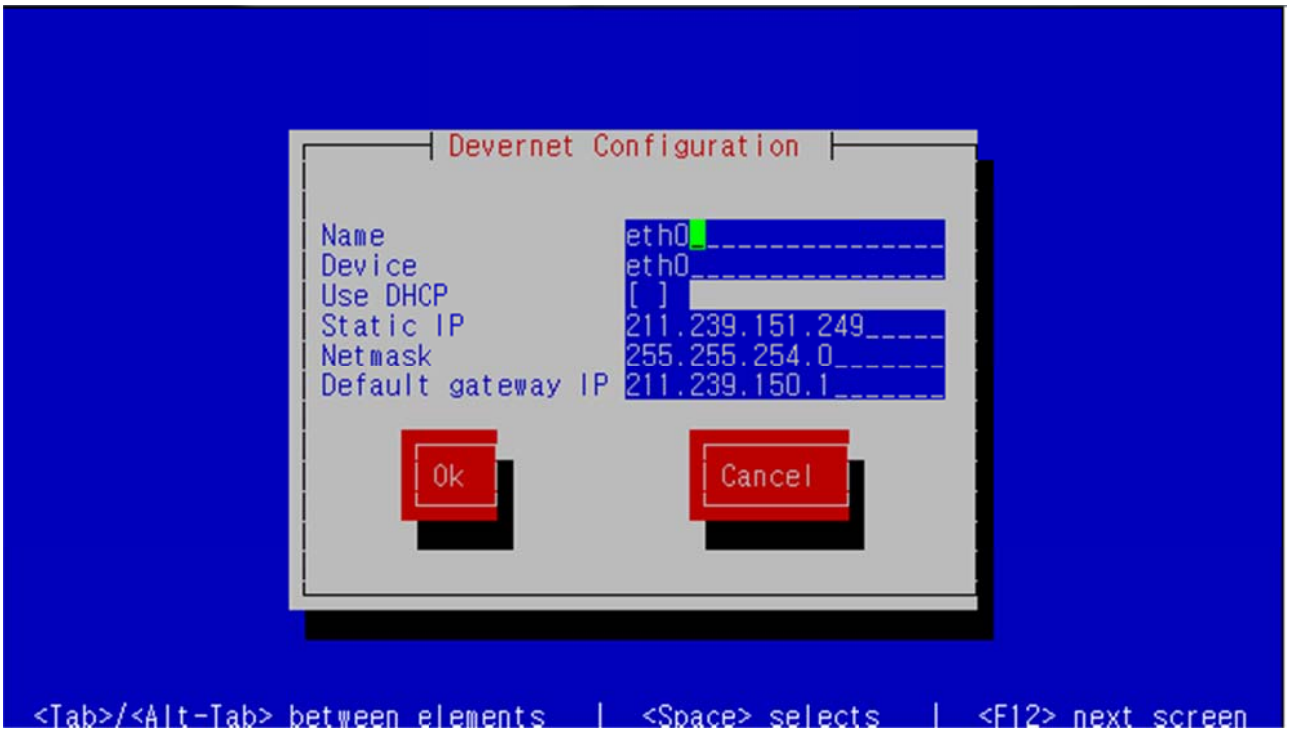
[그림 1-20] Device Setup

- OS 상에 인식된 이더넷 디바이스를 선택하는 단계입니다. 설정을 원하는 디바이스를 선택한 후, [Save]를 선택합니다.



[그림 1-21] 이더넷 디바이스 선택

- 네트워크 정보 입력 단계입니다. 필요한 정보를 입력합니다.
  입력이 완료 된 후 [OK]를 선택합니다. 이제 모두 선택이 완료 되었으므로 명령모드로 다시 복귀합니다.



[그림 1-22] 네트워크 정보 입력

➢ ifconfig 명령을 통한 네트워크 설정

ifconfig를 통해서도 임시 IP 설정이 가능 합니다. 재부팅시 설정은 저장되지 않습니다.

```
root@local:/
[root@localhost /]# ifconfig eth0 10.10.10.10 netmask 255.255.255.255.0 broadcast 10.10.10.255
up
[root@localhost ~]# route add default gw 10.10.10.1
```

➢ 라우팅 테이블 확인

netstat -nr 또는 route 명령을 통해 서버의 라우팅 테이블을 확인할 수 있습니다.

```
root@local:/
[root@localhost /]# netstat -nr
Kernel IP routing table
Destination     Gateway         Genmask         Flags   MSS Window  irtt Iface
10.10.10  .0    0.0.0.0         255.255.255.0   U         0 0          0 eth0
169.254.0.0     0.0.0.0         255.255.0.0     U         0 0          0 eth0
0.0.0.0         10.10.10.1      0.0.0.0         UG        0 0          0 eth0
```

2.4 네트워크 인터페이스 확인

ifconfig 명령을 통해 ip 설정 사항 및 netmask, gateway 설정등 네트워크 인터페이스에 관련된 부분들을 확인 할 수 있습니다.

```
root@local:/
[root@localhost /]# ifconfig
eth0      Link encap:Ethernet  HWaddr 00:13:8F:8E:33:EB
          inet addr:10.10.10.10  Bcast:10.10.10.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:130121157 errors:0 dropped:0 overruns:0 frame:0
          TX packets:98796127 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:1852565390 (1.7 GiB)  TX bytes:2267228385 (2.1 GiB)
          Interrupt:22 Base address:0x8c00

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:2592732 errors:0 dropped:0 overruns:0 frame:0
          TX packets:2592732 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:827763510 (789.4 MiB)  TX bytes:827763510 (789.4 MiB)
```

● 네트워크 설정 파일

설정한 네트워크 정보는 특정 파일에 기록되어 시스템이 리부팅 되어도 다시 설정을 불러내도록 설정됩니다.

3.1 /etc/sysconfig/network

Hostname을 등록하는 파일입니다.

```
root@local:/
[root@local /]# cat /etc/sysconfig/network
NETWORKING=yes
HOSTNAME=hostway.co.kr
```

➢ /etc/sysconfig/network-scripts/ifcfg-eth0

호스트 설정 파일로 IP, gateway 등을 설정해주는 파일이며 setup 명령으로 ip 추가 수정등의 작업을 하게 되면 이 파일에 기록되어지게 됩니다.

```
root@local:/
[root@localhost /]# cat /etc/sysconfig/network-scripts/ifcfg-eth0
DEVICE=eth0
BOOTPROTO=static
ONBOOT=yes
TYPE=Ethernet
NETMASK=255.255.255.0
IPADDR=10.10.10.10
GATEWAY=10.10.10.1
```

3.3 /etc/resolv.conf
로컬 DNS 설정 파일 입니다.
자세한 내용은 네임서버 편에서 다루도록 하겠습니다.

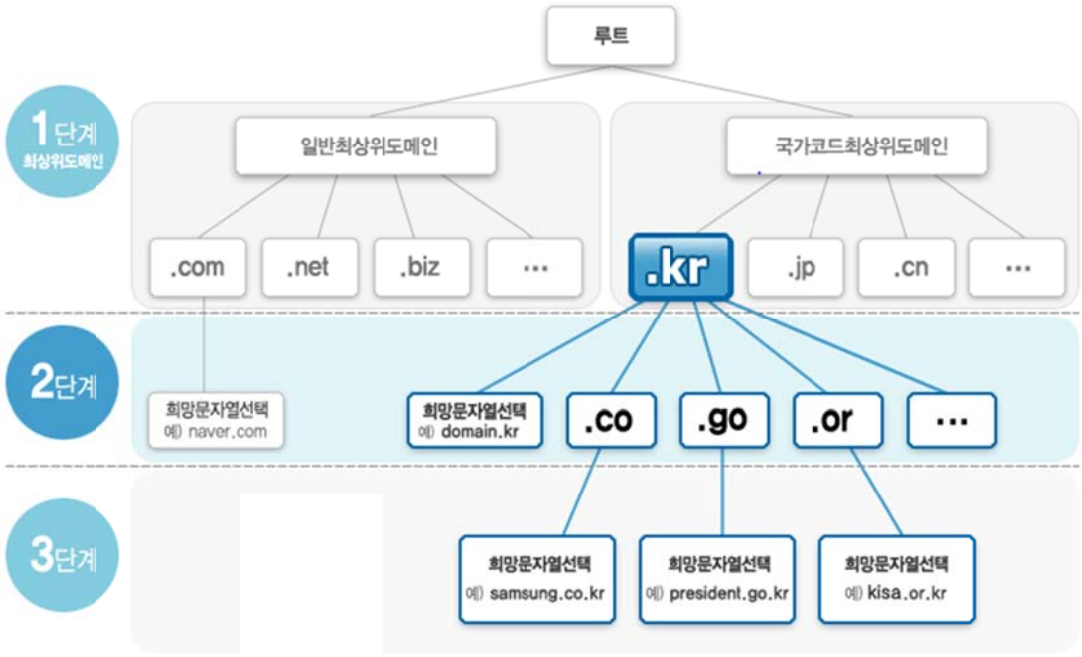
```
root@local:/
[root@localhost /]# cat /etc/resolv.conf
nameserver 61.100.13.145
nameserver 61.100.13.46
```

Chapter2. 네임서버

1. 네임서버의 정의 및 이해

- DNS(Domain Name System)은 도메인 이름을 사용하는 수직적인 체계를 말합니다. 편지나 택배를 보낼 때 주소를 알아야 하는 것처럼 홈페이지를 찾아가기 위해서는 IP주소를 알아야 합니다. IP 주소는 통신망에 연결된 컴퓨터에 부여되는 고유 식별번호 입니다.
- 홈페이지나 Mail, FTP 등의 접근을 위해서는 IP 주소가 필요로 하게 되는데 각기 다른 곳의 IP를 외우는 것은 기억하기 어려운 것에 대해 기억하기 쉽게 문자로써 만든 것이 도메인입니다. 이 도메인을 IP로 연결 해주는 것이 DNS서버 인 것입니다.

네임서버를 이해하기 위해서는 도메인의 구조를 아는 것이 중요 합니다.
아래 그림은 도메인의 역트리 구조를 계층적으로 잘 보여주고 있습니다.



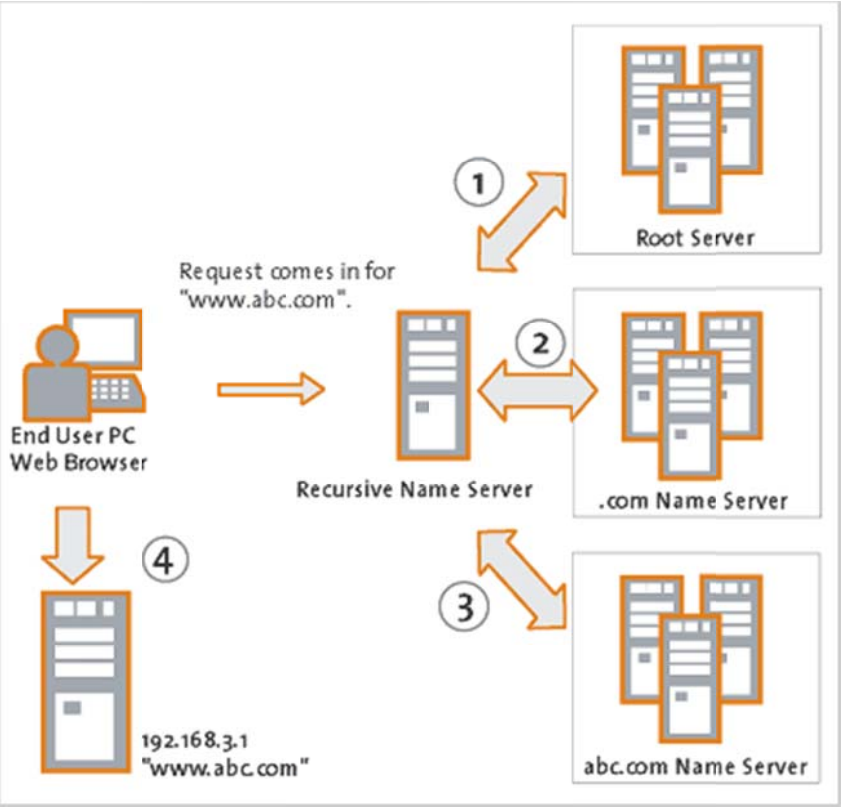
[그림 2-1] 도메인의 역트리 구조(그림 출처 : http://domain.nida.or.kr/ 한국인터넷진흥원)

웹서버나 메일서버 모두 정상 작동한다고 하더라도 사용자 입장에서는 네임서버가 제대로 작동하지 않고 있다면, 해당 도메인에 접근 할 수 없기 때문에 네임서버는 굉장히 중요한 역할을 하고 있습니다.

PC에서 도메인 입력을 통해 웹 서버나 메일서버를 찾아갈 때 다음과 같은 순서로 질의가 이뤄짐을 볼 수 있습니다.

클라이언트(PC)에서 www.abc.com을 찾아가는 경우의 예

A. PC에서 참조하는 Local DNS에서 찾고자 하는 도메인의 네임서버 정보를 가지고 있는 경우는 응답이 빠른 DNS를 이용해 해당 웹 또는 메일서버를 찾게 됩니다.
B. ①번과 같이 최초 root 서버로 www.abc.com 도메인을 질의하여 바로 결과 값을 리턴 받는게 아니라 최상위 도메인 네임서버인 .com 서버의 정보를 수신합니다.
C. ②번과 같이 root 서버로부터 수신된 .com 네임 서버로 다시 www.abc.com 도메인을 질의하면 역시 결과 값을 바로 리턴 받는게 아니라 해당 도메인의 네임 서버의 정보를 수신합니다.
D. ③번과 같이 수신된 abc.com 도메인의 네임 서버에 www.abc.com 도메인을 질의하여 www 호스트 IP의 정보를 수신 받습니다.
E. ④번과 같이 확인 된 정보를 통해 www.abc.com 웹페이지에 접속이 가능합니다.



[그림 2-2] 네임 서버 동작 원리(그림 출처 : http://forum.codecall.net/)

2. 네임서버 설치(Bind9)
리눅스 서버에서 가장 널리 사용하는 네임서버인 Bind를 이용해 네임서버를 설치 및 구축 해보도록 하겠습니다.

- 버전대별 설정이 조금씩 상이할 수 있습니다.
- 사용하시는 OS 및 Bind 버전에 맞춰 설정하시기 바랍니다.
- 본 설치 과정은 Cent 5.4 / Bind 9.3.4를 기준으로 작성 하였습니다.

각 설정 파일들에 대한 설명은 설치 과정이 끝난 이후에 하도록 하겠습니다.

Cent에서는 bind를 rpm 패키지로 기본 제공하고 있습니다. yum으로 아래와 같이 쉽게 설치할 수 있습니다. (Domain : hostway.co.kr, IP : 10.10.10.10)

- Bind 패키지가 설치되어 있는지 확인하고 설치가 되어 있지 않은 경우 bind, caching-nameserver 패키지를 설치합니다.

```
root@local:/
[root@local /]# rpm -qa | grep bind
[root@local /]# yum install bind* caching
Loaded plugins: fastestmirror
```

- named.caching-nameserver.conf 파일을 수정합니다.

```
root@local:/
[root@localhost /]# vi /etc/named.caching-nameserver.conf
//
// named.caching-nameserver.conf
//
// Provided by Red Hat caching-nameserver package to configure the
//
options {
        listen-on port 53 { any; };
        listen-on-v6 port 53 { any; };
        directory       "/var/named";
        dump-file       "/var/named/data/cache_dump.db";
        statistics-file "/var/named/data/named_stats.txt";
        memstatistics-file "/var/named/data/named_mem_stats.txt";

        // Those options should be used carefully because they disable port
        // randomization
        query-source    port 53;
        query-source-v6 port 53;

        allow-query     { any; };
        allow-query-cache { any; };
};
logging {
        channel default_debug {
                file "data/named.run";
                severity dynamic;
        };
};
view localhost_resolver {
        match-clients      { any; };
        match-destinations { any; };
        recursion yes;
        include "/etc/named.rfc1912.zones";
};
```

- named.rfc1912.zones 파일에 도메인 설정을 추가 합니다.

root@local:/

```
[root@localhost /]# vi /etc/named.rfc1912.zones
// named.rfc1912.zones:
//
// Provided by Red Hat caching-nameserver package
// See /usr/share/doc/bind*/sample/ for example named configuration files.
//

zone "hostway.co.kr" IN {
        type master;
        file "hostway.zone";
        allow-update { none; };
};

zone "." IN {
        type hint;
        file "named.ca";
};

zone "localdomain" IN {
        type master;
        file "localdomain.zone";
        allow-update { none; };
};

zone "localhost" IN {
        type master;
        file "localhost.zone";
        allow-update { none; };
};
```

- Rfc1912에서 정의한 hostway.zone 파일을 생성합니다.

```
root@local:/
[root@localhost /]# vi /var/named/chroot/var/named/hostway.zone
$TTL 86400
@          IN SOA @          root (
                                42        ; serial (d. adams)
                                3H        ; refresh
                                15M       ; retry
                                1W        ; expiry
                                1D )      ; minimum

           IN NS              ns.hostway.co.kr.
           IN NS              ns2.hostway.co.kr.
           IN MX 10           mail
           IN A               10.10.10.10
mail       IN A               10.10.10.10
www          IN A               10.10.10.10
ns         IN CNAME           @
*          IN CNAME           ns
```

- named-checkzone, named-checkconf 명령을 통하여 네임 서버 설정이 올바르게 설정되어 있는지 확인 후 named 데몬을 시작해 줍니다.

```
root@local:/
[root@localhost /]# named-checkconf /etc/named.rfc1912.zones
[root@localhost /]# named-checkconf /etc/named.caching-nameserver.conf
[root@localhost /]# named-checkzone hostway.co.kr/var/named/chroot/var/named/hostw
ay.zone
[root@localhost /]# /etc/init.d/named start
```

3. 설정 파일

네임 서버 설치 및 설정은 자체 네임 서버를 구축하기 위한 용도입니다. 그 보다 우선적으로 시스템 내에 설정되어야 하는 설정 파일을 확인해 보겠습니다.

3.1 /etc/host.conf

- 이 파일을 통해 도메인 찾아가기 위해 어떤 파일을 먼저 참조할 것인지 우선순위를 결정할 수 있습니다.

[표 2-1] host.conf 퍼알애 입력할 수 있는 옵션

옵션	설명
order	다른 네임resolve 매커니즘이 시도 되는 순서를 지정하며, 설정 값으로는 hosts, bind, nis 가 있습니다. Hosts : 로컬 /etc/hosts 파일을 참조하여 도메인 연결 시도 Bind : DNS 네임서버를 통해 질의 Nis : 네트워크 정보 NIS 프로토콜 사용하여 도메인 연결 시도
Alert	설정값으로는 on /off가 있으며, 이 옵션을 on 하게 되면 IP주소에 대한 Spoof 시도가 syslog 기능을 통해 로그가 기록됩니다.

nospoof	이 옵션의 주된 목적은 IP주소를 spoof하지 못하게 하는 것이며, spoof란, DNS가 hostname과 IP를 검사할 때 두 값이 서로 일치하지 않을 경우를 말하며 이 경우 서버는 해당 name을 받아들이지 않고 거절하며, error를 return하게 된다. (이 옵션을 사용함으로써 서버에 접속할때마다 값을 비교하게 되면 서버의 추가 로드가 증가할 수 있기 때문에 주의) 필요 시에만 on으로 설정하고,기본값은 off.
Trim	로컬서버에 여러 도메인이 있는 경우, 기본 호스트를 결정하기 위해 사용할 수 있습니다. /etc/hosts 파일과 같은 기능을 하며, 기본값은 off.
multi	설정 값으로는 on / off가 있으며, 하나의 호스트가 /etc/hosts 파일에 여러 개의 IP주소를 가질 수 있도록 허용합니다.

아래는 /etc/host.conf 파일에 실제 옵션을 적용해본 화면입니다.
기본 값으로 도메인을 찾아갈 때 hosts파일 ,bind(네임서버) 순으로 지정되어 있는 것을 볼 수 있습니다.

```
root@local:/
[root@local /]# cat /etc/host.conf
order hosts,bind
alert on
trim hostway.co.kr
```

3.2 /etc/resolv.conf
/etc/hosts 파일에 찾고자 하는 도메인의 정보가 없을 경우 서버에서 사용할 DNS서버 주소를 설정하는 파일입니다. 위 내용에서 처럼 자체 네임서버를 구성한 경우라면 127.0.0.1(loop backup)을 설정해도 정상적으로 구동하게 됩니다. 네임서버를 정의한 순서대로 참조하게 됩니다.

```
root@local:/
[root@localhost /]# cat /etc/resolv.conf
search localdomain
nameserver 10.10.10.10
nameserver 61.100.13.145
```

3.3 /etc/hosts
서버에서 도메인을 찾고자 할 때 제일 처음 참조하게 되는 파일입니다. 그 이유는 위에서 언급한 것처럼 /etc/host.conf파일에 순서가 정의되어 있기 때문입니다.
예를 들어 hostway.co.kr의 IP를 아래와 같이 지정하게 되면 서버에서는 hostway.co.kr의 페이지가 아닌 10.10.10.10의 웹 페이지를 출력하게 됩니다.

```
root@local:/
[root@localhost /]# cat /etc/hosts
10.10.10.10       hostway.co.kr              hostway
```

hosts 파일 안에 등록하는 형식으로는 첫 번째 필드에 IP 주소, 두번째 필드에 호스트 네임, 세번째 필드는 호스트 네임에 대한 닉네임을 설정하는 부분입니다.

해당 서버에서는 hostway.co.kr 이라는 도메인을 입력하게 되면 10.10.10.10의 웹 페이지를 보여주게 됩니다.

위와 같이 관리하는 서버들에 대해서 쉽게 임의의 호스트네임을 지정하여 많은 IP 들을 기억하고 있지 않아도 되고 IP Address대신 지정한 호스트 네임을 통해 쉽게 접근 할 수 있게 됩니다. 자주 접속 하는 서버를 등록해 놓고 사용하면 네임서버에 질의하는 시간이 단축되게 됩니다.

윈도우 에도 동일한 역할을 하는 hosts 파일이 존재합니다.
NT계열의 경우는 C:₩WINDOWS₩system32₩drivers₩etc₩hosts 경로에 위치하고 있습니다.
도메인을 추가하고 적용하기 전에 테스트시에 유용하게 사용할 수 있습니다.

3.4 /etc/named.caching-nameserver.conf
기존에 chroot가 적용되지 않았던 rpm 버전들은 named.conf를 기본 사용하도록 되어 있었으나, chroot가 적용되기 시작하면서 named.caching-nameserver.conf 파일을 사용하고 있습니다.
chroot의 root path는 /var/named/chroot 이며, 네임서버 실행시에 해당 위치가 root가 되는 것입니다. 설정 파일들도 해당 경로 아래에 존재하고 있습니다.
도메인 설정은 기존 버전들에서는 named.con 파일안에 모두 설정 했으나, rfc1912.zone파일에 대신하고 있으며, zone파일들의 위치 또한 chroot상의 /var/named,
즉, /var/named/chroot/var/named가 됩니다.

아래 화면은 위 설명과 같이 chroot로 시작해서 /etc/named.caching-nameserver.conf 설정을 가져온 것을 볼 수 있습니다.

3.5 /etc/named.rfc1912.zones
이전 버전들과 다르게 named.rfc1912.zones 파일에 사용할 도메인과 해당 도메인의 zone 파일들을 정의하게 됩니다.
Bind의 홈디렉토리인 /var/named/chroot/var/named/ 경로에 zone들이 기본 설정 zone 파일들이 존재하고 있음을 확인 할 수 있습니다. (localdomain.zone , localhost.zone 등)
이 곳이 실제 추가되어 운영될 zone 파일들이 위치할 곳입니다.
존 파일 세팅은 2.네임서버 설치를 참고해 주시기 바랍니다.

4. 네임서버 정보 검색 유틸리티
DNS 서버가 정상적으로 설치되어 구동중인지, 또 도메인에 대한 상태를 확인하기 위해서 dig, nslokkup 등의 명령을 사용합니다.
4.1 dig (domain information groger)
도메인에 대한 검색 결과를 상세하게 출력해주고, 다른 DNS서버의 설치 정보(bind 버전 등)을

확인할 수도 있으며, 사용 형식은 다음과 같습니다.

```
dig [@server] [도메인] [쿼리형태] [쿼리클래스]
```

옵션 설명

- Server : DNS 서버의 domain 혹은 IP주소
- 도메인 : 정보를 요청하고자 하는 도메인 이름
- 쿼리형태 :
  - a : 네트워크 주소
  - any : 특정 도메인에 대한 모든 정보
  - mx : 도메인의 메일 교환
  - ns : 네임서버
  - soa : Zone 파일 상단의 authority 레코드
  - hinfo : 호스트 정보
  - axfr : Zone 파일 교환(transfer)
  - txt : arbitrary number of strings
- 쿼리클래스 :
  - in : 인터넷 클래스 도메인
  - any : 모든 클래스의 정보

예제로 hostway.co.kr 도메인의 ns 레코드를 검색한 결과입니다.

```
root@local:/
[root@localhost /]# dig @ns5.cninet.co.kr hostway.co.kr ns
; <<>> DiG 9.3.6-P1-RedHat-9.3.6-4.P1.el5_4.2 <<>> @ns5.cninet.co.kr hostway.co.kr ns
; (1 server found)
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 63297
;; flags: qr rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 2

;; QUESTION SECTION:
;hostway.co.kr.                 IN      NS

;; ANSWER SECTION:
hostway.co.kr.          31      IN      NS      ns.siteprotect.co.kr.
hostway.co.kr.          31      IN      NS      ns2.siteprotect.co.kr.

;; ADDITIONAL SECTION:
ns.siteprotect.co.kr.   9386    IN      A       66.232.139.10
ns2.siteprotect.co.kr.  3       IN      A       61.100.13.50

;; Query time: 1 msec
;; SERVER: 61.100.13.145#53(61.100.13.145)
;; WHEN: Tue May 25 02:44:26 2010
;; MSG SIZE  rcvd: 110
```

4.2 host

host 유틸리티는 인터넷 호스트 정보를 IP로 변환 해주는 기능을 가지고 있으며 nslookup과 유사합니다.

다음은 host 유틸리티를 통해 옵션 없이 hostway.co.kr을 검색 해보도록 하겠습니다.

```
root@local:/
[root@localhost ~]# host hostway.co.kr
hostway.co.kr has address 61.100.13.104
hostway.co.kr mail is handled by 10 filter.hostway.co.kr.
hostway.co.kr mail is handled by 20 mailstore1.hostway.co.kr.
hostway.co.kr mail is handled by 30 mailstore2.hostway.co.kr.
```

hostway.co.kr 이라는 도메인은 61.100.13.104 라고 결과를 보여주고 있습니다.
옵션을 추가하여 조금 더 구체적인 정보를 얻을 수도 있습니다.

```
root@local:/
[root@localhost ~]# host -v -r -t ns hostway.co.kr
Trying "hostway.co.kr"
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 60563
;; flags: qr ra; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 2

;; QUESTION SECTION:
;hostway.co.kr.                 IN      NS

;; ANSWER SECTION:
hostway.co.kr.          45      IN      NS      ns2.siteprotect.co.kr.
hostway.co.kr.          45      IN      NS      ns.siteprotect.co.kr.

;; ADDITIONAL SECTION:
ns2.siteprotect.co.kr.  46793   IN      A       61.100.13.50
ns.siteprotect.co.kr.   11385   IN      A       66.232.139.10

Received 110 bytes from 61.100.13.145#53 in 1 ms
```

Charter3. FTP, SSH

1. FTP (File Transfer Protocol)

FTP 란 “File Transfer Protocol”의 약자로써 인터넷 상의 컴퓨터들 간에 파일을 교환하기 위한 표준 프로토콜 입니다. FTP는 2개의 port를 사용하는데, 20번 port는 데이터 전송을 하며 21번은 연결을 하는데 사용하고 있습니다.

자료의 업로드나 백업 등의 작업을 서버에 접속하지 않아도 쉽게 자료를 업로드/다운로드 할 수 있기 때문에 ftp를 많이 사용하고 있습니다.

1.1 vsftpd(Very Secure FTP Daemon)

vsftpd는 이름에서도 알 수 있듯이 보안 부분을 강조한 데몬으로 RedHat, Suse, Open-BSD에서 기본 FTP로 채택하고 있으며, 빠른 퍼포먼스, 안정성, 성능 등 다방면에서 좋은 평가를 받고 있는 ftp입니다.
Config 파일을 통한 설정 또한 어렵지 않기 때문에 많은 사용자들이 이용하고 있습니다.
소스와 rpm 두가지 설치 방법이 존재 하며, 설치 하기 쉽고 많이 사용하는 방법인 yum(RPM버전)을 통해 설치 하는 방법을 설명하도록 하겠습니다.

리눅스 뿐만 아니라 Solaris, IRIX, HP-UX, 등의 유닉스계열 운영체제 대부분에서 사용이 가능합니다.

초기 설치시에 vsftpd 데몬을 설치하지 않았다면 위와 같이 확인 시 아무런 결과가 출력되지 않습니다.
yum 이라는 툴을 통해 의존성 문제 해결과 함께 쉽게 설치를 하실 수 있습니다.
설치만 한 상태에서도 /etc/rc.d/init.d/vsftpd start 명령을 통해 바로 구동이 가능 하지만, 보안이나 관리를 위해 환경 설정을 해주시는 것이 좋습니다.

1.1.1 환경설정

vsftpd의 경우 위와 같이 yum을 통해 설치 한 후 별도의 설정 없이 기본 설정으로도 사용 가능합니다.
하지만, 보안상이나 FTP서버를 운영, 관리를 위해서는 아래 설정들을 필요에 따라 추가하고 수정하는 작업들을 해주시는 것이 좋습니다.

설정 파일인 vsftpd.conf 내용에 대해서 자세히 알아보도록 하겠습니다.

root@local:/

```
# Example config file /etc/vsftpd/vsftpd.conf
#
# The default compiled in settings are fairly paranoid. This sample file
# loosens things up a bit, to make the ftp daemon more usable.
# Please see vsftpd.conf.5 for all compiled in defaults.
#
# READ THIS: This example file is NOT an exhaustive list of vsftpd options.
# Please read the vsftpd.conf.5 manual page to get a full idea of vsftpd's
# capabilities.
#
# Allow anonymous FTP? (Beware - allowed by default if you comment this out).
anonymous_enable=YES
```

파일 내용안은 옵션 위에 #(주석처리) 된 상태로 설명이 존재하고 있으나, 각 주요 옵션들에 대해 설명 하도록 하겠습니다.

[표 3-1] vsftpd.conf 파일 옵션 설명

지시자	설 명
anonymous_enable=YES	익명 사용자의 접속을 허가 여부를 결정합니다. 익명접속을 허가 하고자 한다면 YES 로 설정합니다.
local_enable=YES	서버내 /etc/passwd파일에 정의된 로컬 사용자들의 접속 허가 여부를 결정합니다. NO로 설정시 익명사용자만 접근이 가능하게 됩니다.
write_enable=YES	FTP 전송 명령어중 write 명령어의 사용여부를 결정하는 옵션으로 기본 설정은 YES 입니다.
local_umask=022	로컬 사용자들의 umask 값을 설정(umask는 퍼미션에서 해당 권한을 뺀 수로써 022일 경우는 디렉토리 생성시 기본 퍼미션은 755외 되는 것입니다.) 이는 용도에 맞게 설정값을 변경해주시면 됩니다.
anon_upload_enable=YES	익명 사용자에게 파일을 업로드가 가능하게 허용해줄 것인지에 대한 지시자입니다. 허용시에는 업로드 할 수 있는 디렉토리가 존재해야 합니다. 보안상 NO로 설정하시는 것이 좋습니다.
anon_mkdir_write_enable=YES	위 옵션에서는 파일만 업로드 가능했던 것으로 디렉토리는 생성할 수 없었으나, 디렉토리 생성 허용 여부를 결정해주는 지시자로 이 또한 보안상 NO로 설정하시는 것이 좋습니다.
dirmessage_enable=YES	사용자가 특정디렉토리로 접근시 메시지를 보여줄 것인지에 대한 설정입니다. 아래 내용중 message_file 지시자를 통해 사용할 파일명을 정의할 수 있습니다.
message_file=.message	특정 디렉토리로 접근시 디렉토리 안내메세지 파일로 사용될 파일명을 지정해주는 지시자로 dirmessage_enable가 YES로 설정되어 있어야 적용 됩니다.

xferlog_enable=YES	ftp상에서 파일 업로드/다운로드 로그를 기록할 것인지에 대한 여부를 설정하는 지시자입니다.
connect_from_port_20=YES	ftp는 20,21번 port를 사용하는데, 20번 포트의 역할인 포트의 데이터전송 연결을 허용할 것인지에 대한 여부를 설정해줍니다.
chown_uploads=YES	익명 사용자가 등록한 파일의 소유권을 변경할 것인지를 묻는 지시자입니다. chown_username을 통해 변경될 소유권자를 지정할 수 있습니다.
chown_username=whoever	익명사용자가 등록한 파일을 chown_username으로 지정된 이름으로 소유권을 변경되게 됩니다.
xferlog_file=/var/log/vsftpd.log	xferlog_enable=YES 지시자를 통해 로그를 기록할 경우 로그의 위치를 지정해주는 지시자입니다.
xferlog_std_format=YES	로그파일의 포맷을 기본 포맷으로 남길 것인지에 대한 설정입니다.
idle_session_timeout=600	ftp 연결시 타임아웃값의 설정이며, 설정한 시간동안 아무런 작업이 없을 경우 강제로 세션을 끊어 로그아웃 되게 됩니다.
data_connection_timeout=120	데이터 전송시의 타임아웃값을 말하며, 큰 파일을 업로드 하거나 다운로드 할 때 전송도중 끊기는 현상이 발생한다면 이 설정값을 늘리거나 주석처리 합니다.
ascii_upload_enable=YES ascii_download_enable=YES	ASCII 모드의 업로드/다운로드 사용여부를 설정 해줍니다.
ftpd_banner=Welcome to blah FTP service.	ftp 접속시 환영메세지를 지정하는 지시자입니다.
deny_email_enable=YES banned_email_file=₩ /etc/vsftpd.banned_emails	익명 접속시 패스워드에 일반 이메일 주소를 거부할 것인지에 대해 설정이 가능하며, vsftpd.banned_emails에 정의된 이메일 주소만 허용합니다.
chroot_list_enable=YES chroot_list_file=₩ /etc/vsftpd.chroot_list	모든사용자가 아닌 특정사용자들에 대해 자신의 홈디렉토리를 루트디렉토리로 인식하도록 하는 기능으로 사용자의 홈디렉토리의 상위디렉토리로 벗어나지 못하게 하는 기능을 합니다. /etc/vsftpd.chroot_list에 지정된 사용자는 홈디렉토리가 최상위 디렉토리가 됩니다.
chroot_local_user=YES	YES로 설정하게 되면 특정사용자가 아닌 전체사용자를 대상으로 자기자신의 홈디렉토리의 상위디렉토리로 이동하지 못하게 됩니다.
ls_recurse_enable=YES	ftp 접속시에 ls 명령의 -R 옵션인 서브디렉토리내의 파일 목록까지 모두 확인할 수 있게끔 해주는 것으로 기본 설정은 NO 입니다.

listen=YES	xinetd모드가 아닌 standalone 모드로 서비스할 때 YES로 설정합니다.
pam_service_name=vsftpd	PAM 설정 파일명을 지정해주는 것입니다.
userlist_enable=YES	/etc/vsftpd/user_list 파일에 정의된 계정사용자들의 접근을 차단하거나 허용할 수 있습니다.
tcp_wrappers=YES	Tcp_wrapper의 접근제어를 받을지에 대한 여부를 결정해주는 지시자입니다.
max_clients=30	동시 접속자수를 제한 할 수 있으며, 0으로 설정시 제한하지 않습니다.
max_per_ip=3	Max_per_ip는 한 IP에서 사용할 수 있는 동시접속 수를 제한 할 수 있습니다.

1.1.2 vsfptd 실행과 종료

```
root@local:/
[root@localhost /]# /etc/rc.d/init.d/vsftpd start
Starting vsftpd for vsftpd:                                [  OK  ]
[root@localhost /]# /etc/rc.d/init.d/vsftpd stop
Shutting down vsftpd:                                      [  OK  ]
```

위와 같이 시작 및 종료가 가능하며, 정상적으로 데몬이 구동되었는지 아래와 같이 확인 할 수 있습니다.

```
root@local:/
[root@localhost /]# netstat -atnp | grep vsftpd
tcp        0      0 0.0.0.0:21                  0.0.0.0:*               LISTEN      31533/vsftpd
[root@localhost /]# ps aux | grep vsftpd
root      31533  0.0  0.0   5188   504 ?   Ss   11:59   0:00 /usr/sbin/vsftpd
/etc/vsftpd/vsftpd.conf
root      31542  0.0  0.0   3916   668 pts/0  R+   12:03   0:00 grep vsftpd
```

1.1.3 FTP 로그 확인

vsftpd.conf 파일에서 로그를 기록하게끔 설정하게 되고 그 위치를 지정했을 경우 해당 파일에서 언제, 어디서, 어떤 파일을 어디서 가져갔는지 어디에 업로드 했는지에 대한 정보들이 기록되게 됩니다.

```
root@local:/
[root@localhost /]# cat /var/log/xferlog
Fri May 21 12:26:59 2010 40 211.115.196.50 1328020 /home/201C.mp3 b _ i r hostway ftp 0 * c
```

로그 설명

- 업/다운로드 된 시간 : Fri May 21 12:26:59 2010
- 파일전송지속시간 : 40
- 원격호스트 : 211.115.196.50
- 파일사이즈 : 1328020
- 파일의 경로 및 이름 : /home/2010.mp3
- 파일종류 : b
- Sp-flag : _
- 전송방향 : o (o : outgoing, I : incoming 업/다운로드를 구분)
- 접근모드 : r (A : 익명, g : 손님, r : 인증된 사용자)
- 사용자명 : hostway
- 서비스 : ftp
- 인증방법 : 0 (0 : 인증사용안함, 1 : 인승사용)
- 인증사용자 ID :

Xferlog를 통해 어떤 파일들이 누가 어떻게 어디서 업로드 되고 다운로드 되는지 확인하실 수 있습니다.

2. Open SSH (open secure shell)

Open SSH는 리눅스 서버에서 보안 접속을 하는데 가장 널리 사용되는 SSH 프로토콜을 구현한 소프트웨어로써 원격에서 서버에 접속하고자 할 때 사용 됩니다.
과거에 telnet의 경우는 서버와 클라이언트가 통신할 때 전송한 아이디가 평문 그대로 노출되어 전송되어 스니핑을 통해 아이디가 쉽게 노출되었으나, ssh는 모든 문자들을 암호화하여 누군가 가로채더라도 그 내용을 알아내기 어렵게 됩니다.

현재 거의 모든 리눅스 배포판에 기본으로 설치되고 있습니다.

2.1 ssh 사용법

리눅스와 같이 콘솔상에서 접속하는 방법은 아래와 같습니다.

```
형식 : ssh -l 유저네임서버주소
형식 : ssh 유저네임@서버주소
```

예) hostway.co.kr 의 admin이라는 계정으로 접속한다고 가정한 방법입니다.

Windows 환경에서 리눅스에 접근하고자 하는 경우는 별도의 remote 프로그램을 설치해줘야 합니다. 많이 널리 사용되는 프로그램은 프리웨어인 putty, SSH Secure Shell Client, Zterm, Tunnelier 등이 존재합니다. 상용 소프트웨어인 secure CRT도 많은 사용자들이 사용하고 있습니다.

2.2 ssh 환경 설정

ssh 환경 설정 파일은 /etc/ssh/ssh_config 파일이며, ssh의 포트 변경이라던지 root 로그인 허용여부 결정등의 설정을 변경할 수 있습니다.

[표 3-2] /etc/ssh/ssh_config 의 주요 지시자

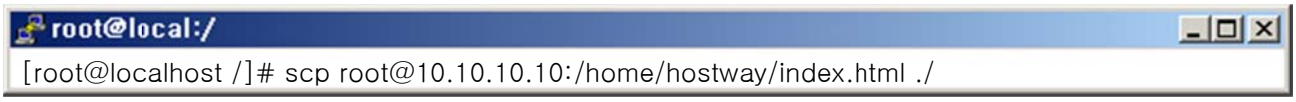
지시자	설명
Port 22	기본 주석 처리되어 22번 포트를 사용하며, 다른 port로 변경하고자 할때는 주석처리를 풀고 사용하고자 하는 포트 번호를 적어주면 됩니다
Protocol 2	허용 프로토콜을 정의합니다. 최근에는 보안상의 이유로 protocol 1은 사용하지 않습니다.
ListenAddress 0.0.0.0	접근 허용할 IP대역을 정의 합니다. 기본값으로 모두 허용 됩니다.
LoginGraceTime 2m	유저의 로그인이 성공적으로 이루어지 않았을 때 이 시간 후에 서버가 연결을 끊는 시간입니다.
PermitRootLogin yes	Root(관리자계정)으로 바로 접근 허용 여부를 설정하는 지시자입니다. 기본값은 YES이나, 보안상 no 설정할 것을 권장합니다.
MaxAuthTries 6	로그인 실패할 경우 재시도 횟수를 설정해줍니다.

환경 설정 파일을 변경한 후에는 /etc/rc.d/init.d/sshd restart와 같이 데몬을 재시작 해주어야 변경한 내용이 적용 됩니다.

2.3 scp / sftp를 이용한 파일 복사 및 전송

Open ssh에서 제공되는 scp라는 명령을 통해 자신의 서버에서 원격의 서버로 또는 원격의 서버에서 자신의 서버로 파일을 간단하게 전송할 수 있습니다.

다음은 10.10.10.10 IP를 가진 서버에 root 계정을 통해 /home/hostway/Index.html 파일을 자신의 서버로 현재 위치한 디렉토리로 복사하는 명령입니다.



자신의 서버에 /home/test 디렉토리를 10.10.10.10 IP 서버쪽 admin 계정을 통해 /home/admin/로 복사하는 명령은 다음과 같습니다.

파일을 복사하거나 가져오고자 하는 서버의 ssh port 번호가 22번이 아닌 다른 port를 사용할 경우는 -P 옵션을 사용해 지정해주어야 합니다.
아래는 2233번 ssh port 를 사용하는 경우의 예제 입니다.

sftp는 대화식의 파일전송 프로그램으로 ssh와 접속방법이 비슷하며, 모든 작동은 암호화된 ssh 전송상에서 실행이 됩니다. 접속후에는 일반 ftp 사용법과 비슷합니다.

```
root@local:/
[root@localhost /]# sftp admin@hostway.co.kr
```

2.4 SSH 보안

위에서 언급했던 /etc/ssh/sshd_config 파일 지시자 수정을 통하여 보안을 강화하는 방법을 설명해 드리도록 하겠습니다.

- PermitRootLogin no (Root 로그인 차단)
  - 악의적인 패스워드 대입공격 등을 통해 root 계정을 해커가 취득 당하게 되면 굉장히 큰 문제가 될 수 있습니다. Root 로그인을 허용하지 않고 일반계정을 통해 로그인하는 방법을 권장합니다.
- AllowTcpForwarding no (Forwarding 막기)
- X11Forwarding no
  - TCP포트와 X11에 대해 포워딩 기능을 사용하지 않는다면 No 사용하시는 것이 좋습니다.
- IgnoreRhosts yes (호스트 기반 인증 차단)
- HostbasedAuthentication no
- RhostsRSAAuthentication no
  - Rhosts 사용이나 hosts.equiv를 통한 인증을 차단합니다.
  - Rhost는 보안상 취약함으로 비활성화 합니다
- AllowUsers 계정명
- AllowGroups 그룹명
  - 특정 계정 또는 그룹만 ssh 접속을 허용하는 방법입니다. 다수 계정 및 그룹은 스페이스(공백)으로 구분

2.5 SSH 자동 로그인

ssh 접속시에 패스워드를 통한 접속이 아닌 dsa 인증을 통해 패스워드 없이 접속하는 방법에 대해 설명해 드리겠습니다. 이는 관리하는 서버간 backup 등의 작업시 용이합니다.

메인 서버에서 아래의 과정을 통해 개인, 공개키를 생성해 줍니다.

```
root@local:/
[root@localhost ~]# ssh-keygen -t rsa -N ""
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_rsa.
Your public key has been saved in /root/.ssh/id_rsa.pub.
The key fingerprint is:
8f:39:93:50:3a:4b:25:4a:6f:80:4a:f6:3e:e9:08:87 root@localhost
```

/root/.ssh/ 아래 id_rsa , id_rsa.pub 파일이 생성된 것을 볼 수 있습니다.

```
root@local:/
[root@localhost .ssh]# ls -l
total 20
-rw------- 1 root root 1675 Mar  8 21:26 id_rsa
-rw-r--r-- 1 root root    0 Mar 22 13:23 id_rsa.pub
```

id_rsa는 개인키이고 서버에 접속하는 클라이언트쪽에서 필요한 파일입니다. id_rsa.pub는 공개키로 서버가 가지고 있어야 합니다. 공개키 파일(id_rsa.pub)을 접속하고자 하는 서버로 복사하고 접속하게 되면 패스워드 없이 접속됨을 볼 수 있습니다.

```
root@local:/
[root@localhost .ssh]# scp id_rsa.pub hostway.co.kr:~/.ssh/authorized_keys
root@hostway.co.kr's password:
id_rsa.pub                                    100%    0     0.0KB/s   00:00
[root@localhost ~]# ssh hostway.co.kr
Last login: Mon May 24 05:21:17 2010 from 114.200.142.106
[root@hostway ~]#
```

Chapter 4.Apache

웹 브라우저를 통해서 서비스되는 http/80/tcp, ssl/443/tcp 등의 웹 서비스를 서버에서 서비스가 되기 위해서는 일반적으로 apache 웹 서버를 통해서 이뤄집니다.
Apache 웹 서버는 현재 세계에서 가장 인기있는 웹 서버이며, 2009년 11월 현재 전세계 웹 서버 중 47%를 차지하고 있습니다.
또한 아파치 소프트웨어 재단에서 관리되고 무료로 배포되고 있으며, 리눅스등의 유닉스 계열뿐 아니라 마이크로소프트 윈도우즈 서버등의 비 유닉스 계열의 운영 체제에서도 성공적이며 광활하게 사용되고 있습니다.

1. apache, php, mysql 설치

웹 서버 운영에 필요한 설치 파일들을 다운받아서 설치하도록 하겠습니다. 개발된 웹 프로그램의 환경에 맞는 버전을 다운 받아 설치하는 것이 일반적이며 여기서는 최신 버전을 선택하여 설치해보도록 하겠습니다.

아래 표에 설치할 파일 및 배포되는 사이트를 참고하시기 바랍니다.

[표 4-1] 설치 프로그램 및 다운로드 사이트

httpd-2.2.15	http://httpd.apache.org/download.cgi
MySQL-5.1.46	http://dev.mysql.com/downloads/mysql/
php-5.2.13	http://www.php.net/downloads.php
ZendOptimizer-3.3.9	http://www.zend.com/en/products/guard/downloads

설치 순서는 MySQL, apache, php, ZendOptimizer 순이며, 경우에 따라서 설치 방법이 달라질 수도 있습니다. 다운로드 받은 프로그램들은 모두 /usr/local/src 디렉토리에 저장을 하고 설치를 하도록 하겠습니다.

1.1 MySQL 설치

MySQL의 경우 다음 장에서 사용법을 설명하겠지만, 일반적으로 웹서버와 연동해서 설치를 하기 때문에 이번장에서는 설치 부분을 설명하도록 하겠습니다.

설치와 관련된 매뉴얼은 mysql-5.1.46 디렉토리내 INSTALL-SOURCE 에 아래와 같이 설치 과정을 확인할 수 있으며, 기타 자세한 사항은 해당 파일을 참조하면 됩니다.

```
root@local:/
[root@localhost /]#
2.3.1. Source Installation Overview

    The basic commands that you must execute to install a MySQL source
    distribution are:
shell> groupadd mysql
shell> useradd -g mysql mysql
shell> gunzip < mysql-VERSION.tar.gz | tar -xvf -
shell> cd mysql-VERSION
shell> ./configure --prefix=/usr/local/mysql
shell> make
shell> make install
shell> cp support-files/my-medium.cnf /etc/my.cnf
shell> cd /usr/local/mysql
shell> chown -R mysql .
shell> chgrp -R mysql .
shell> bin/mysql_install_db --user=mysql
shell> chown -R root .
shell> chown -R mysql var
shell> bin/mysqld_safe --user=mysql &
```

위의 순서대로 mysql 그룹 및 사용자를 생성한 후에 이미 다운로드 받은 압축 파일을 압축 해제합니다.

```
root@local:/
[root@localhost ~]# groupadd mysql
[root@localhost ~]# useradd -g mysql mysql
[root@localhost ~]# tar zxvfp mysql-5.1.46.tar.gz
```

압축이 모두 해제되면 소스 디렉토리로 이동 후 mysql 설치 디렉토리는 /usr/local/mysql 로 지정하고 기본 charset 은 euckr 로 정의합니다.

```
root@local:/
[root@localhost /]# cd mysql-5.1.46
[root@localhost ~]# ./configure -prefix=/usr/local/mysql --with-charset=euckr --with-extra-charsets=all
checking build system type... i686-pc-linux-gnu
checking host system type... i686-pc-linux-gnu
checking target system type... i686-pc-linux-gnu
checking for a BSD-compatible install... /usr/bin/install -c
checking whether build environment is sane... yes
checking for a thread-safe mkdir -p... /bin/mkdir -p
checking for gawk... gawk
checking whether make sets $(MAKE)... yes
checking how to create a ustar tar archive... gnutar
checking for style of include used by make... GNU
checking for gcc... gcc
checking for C compiler default output file name... a.out
checking whether the C compiler works... yes
checking whether we are cross compiling... no
checking for suffix of executables...
```

./configure 가 완료되면 아래와 같은 내용이 출력되며, make 와 make install 을 실행하면 빌드된 mysql 파일들이 /usr/local/mysql 디렉토리로 이동이 됩니다.

```
root@local:/
config.status: executing default commands

Thank you for choosing MySQL!

Remember to check the platform specific part of the reference manual
for hints about installing MySQL on your platform.
Also have a look at the files in the Docs directory.

[root@localhost ~]# make && make install
```

이제 mysql 환경 설정 파일인 my.cnf 파일을 생성하고 초기 데이터베이스를 생성합니다.

root@local:/

```
[root@localhost /]# cp /usr/local/mysql/share/mysql/my-medium.cnf /etc/my.cnf
[root@localhost /]# cd /usr/local/mysql
[root@localhost /]# ./bin/mysql_install_db --user=mysql
Installing MySQL system tables...
OK
Filling help tables...
OK

To start mysqld at boot time you have to copy
support-files/mysql.server to the right place for your system

PLEASE REMEMBER TO SET A PASSWORD FOR THE MySQL root USER !
To do so, start the server, then issue the following commands:

/usr/local/ mysql/bin/mysqladmin -u root password 'new-password'
/usr/local/ mysql/bin/mysqladmin -u root -h rootda.org password 'new-password'

Alternatively you can run:
/usr/local/ mysql/bin/mysql_secure_installation

which will also give you the option of removing the test
databases and anonymous user created by default.  This is
strongly recommended for production servers.

See the manual for more instructions.

You can start the MySQL daemon with:
cd /usr/local/ mysql ; /usr/local/ mysql/bin/mysqld_safe &

You can test the MySQL daemon with mysql-test-run.pl
cd /usr/local/ mysql/mysql-test ; perl mysql-test-run.pl

Please report any problems with the /usr/local/ mysql/bin/mysqlbug script!
```

이제 mysql 설치가 모두 완료되었으며, mysql 데몬을 구동 후 정상적으로 mysql 로 연결이 이뤄지는지 확인합니다.

```
root@local:/
[root@localhost ~]# /usr/local/ mysql/share/mysql/mysql.server start
Starting MySQL.                                            [  OK  ]
[root@localhost ~]# /usr/local/mysql/bin/mysql
Welcome to the MySQL monitor.  Commands end with ; or ₩g.
Your MySQL connection id is 1
Server version: 5.1.46-log Source distribution

Copyright (c) 2000, 2010, Oracle and/or its affiliates. All rights reserved.
This software comes with ABSOLUTELY NO WARRANTY. This is free software,
and you are welcome to modify and redistribute it under the GPL v2 license

Type 'help;' or '₩h' for help. Type '₩c' to clear the current input statement.

mysql>
```

또한, Mysql 데몬이 구동중인지 알 수 있는 방법은 아래와 같이 ps 명령어와 netstat 로 tcp/3306 포트가 LISTEN 상태 확인으로도 가능합니다.

```
root@local:/
[root@localhost /]# ps aux | grep mysql
root     16481  0.0  0.1   5612  1160 pts/0    S    13:47   0:00 /bin/sh /usr/local/mysql
mysql    16584  0.0  0.5  33512  5712 pts/0    Sl   13:47   0:00 /usr/local/src/mysql
[root@localhost ~]# netstat -antp | grep mysql
tcp        0      0 0.0.0.0:3306            0.0.0.0:*               LISTEN      16584/mysqld
```

서버가 부팅 시에 자동으로 mysql 데몬이 구동되도록 하기 위해서는 아래와 같이 설정합니다.

```
root@local:/
[root@localhost /]# echo "/usr/local/mysql/share/mysql/mysql.server start" >> /etc/rc.d/rc.local
[root@localhost ~]# cat /etc/rc.d/rc.local
#!/bin/sh
#
# This script will be executed *after* all the other init scripts.
# You can put your own initialization stuff in here if you don't
# want to do the full Sys V style init stuff.

touch /var/lock/subsys/local

/usr/local/mysql/share/mysql/mysql.server start
```

1.2 apache 설치
앞선 mysql 설치 방법과 같이 소스 파일을 압축 해제 후 해당 디렉토리로 이동합니다.

```
root@local:/
[root@localhost~]# tar zxvf httpd-2.2.15.tar.gz
httpd-2.2.15/
httpd-2.2.15/emacs-style
httpd-2.2.15/httpd.dsp
httpd-2.2.15/libhttpd.dsp
httpd-2.2.15/.deps
httpd-2.2.15/Makefile.in
httpd-2.2.15/include/
httpd-2.2.15/include/scoreboard.h
httpd-2.2.15/include/ap_regkey.h
httpd-2.2.15/include/ap_compat.h
httpd-2.2.15/include/http_config.h
httpd-2.2.15/include/util_time.h
httpd-2.2.15/include/ap_mmn.h
.....
.....
.....

[root@localhost~]# cd httpd-2.2.15
```

디렉토리로 이동하면 INSTALL 파일을 참조하면 설치와 관련된 사항을 확인할 수 있습니다.

```
root@local:/
[root@localhost~]# vi INSTALL

  APACHE INSTALLATION OVERVIEW

  Quick Start - Unix
  ------------------

  For complete installation documentation, see [ht]docs/manual/install.html or
  http://httpd.apache.org/docs/2.2/install.html

      $ ./configure --prefix=PREFIX
      $ make
      $ make install
      $ PREFIX/bin/apachectl start
```

컴파일은 아파치 설치 후에 아파치를 재컴파일 할 필요없이 모듈을 동적으로 추가할 수 있도록 DSO 방식으로 컴파일을 하며, 인증서 서비스도 가능하도록 ssh 모듈을 추가합니다.

```
root@local:/
[root@localhost~]#  ./configure  --prefix=/usr/local/apache2  --enable-so  --enable-mods-shared --enable-ssl
checking for chosen layout... Apache
checking for working mkdir -p... yes
checking build system type... i686-pc-linux-gnu
checking host system type... i686-pc-linux-gnu
checking target system type... i686-pc-linux-gnu

Configuring Apache Portable Runtime library ...

checking for APR... yes
   setting CC to "gcc"
   setting CPP to "gcc -E"
   setting CFLAGS to " -g -O2 -pthread"
  setting  CPPFLAGS  to  "  -DLINUX=2  -D_REENTRANT  -D_GNU_SOURCE  -D_LARGEFILE64_SOURCE"
.....
.....
.....
```

./configure 가 완료되면 아래와 같은 내용이 출력되며, make 와 make install 을 실행하면 빌드된 apache 관련 파일들이 /usr/local/apache2 디렉토리로 이동이 됩니다.

```
root@local:/
config.status: creating support/logresolve.pl
config.status: creating support/phf_abuse_log.cgi
config.status: creating support/split-logfile
config.status: creating build/rules.mk
config.status: creating build/pkg/pkginfo
config.status: creating build/config_vars.sh
config.status: creating include/ap_config_auto.h
config.status: executing default commands

[root@localhost~]# make && make install
Making all in srclib
make[1]: Entering directory `/usr/local/src/httpd-2.2.15/srclib'
Making all in pcre
make[2]: Entering directory `/usr/local/src/httpd-2.2.15/srclib/pcre'
make[3]: Entering directory `/usr/local/src/httpd-2.2.15/srclib/pcre'
.....
.....
.....
```

설치가 모두 완료되면, apache 데몬을 구동하여 정상적으로 작동중인지 아래와 같이 확인을 합니다.

```
root@local:/
[root@localhost~]# /usr/local/apache2/bin/apachectl start

[root@localhost~]# ps aux | grep apache
root      14015  0.2  0.2  7100  2080 ?        Ss   14:17   0:00
/usr/local/apache2/bin/httpd -k start
daemon    14022  0.0  0.1  7236  1944 ?        S    14:17   0:00
/usr/local/apache2/bin/httpd -k start
daemon    14023  0.0  0.1  7236  1940 ?        S    14:17   0:00
/usr/local/apache2/bin/httpd -k start
daemon    14024  0.0  0.1  7236  1936 ?        S    14:17   0:00
/usr/local/apache2/bin/httpd -k start
daemon    14025  0.0  0.1  7236  1940 ?        S    14:17   0:00
/usr/local/apache2/bin/httpd -k start
daemon    14026  0.0  0.1  7236  1936 ?        S    14:17   0:00
/usr/local/apache2/bin/httpd -k start
daemon    14028  0.0  0.1  7236  1948 ?        S    14:17   0:00
/usr/local/apache2/bin/httpd -k start
daemon    14029  0.0  0.1  7236  1948 ?        S    14:17   0:00
/usr/local/apache2/bin/httpd -k start
daemon    14030  0.0  0.1  7236  1936 ?        S    14:17   0:00
/usr/local/apache2/bin/httpd -k start
[root@localhost~]# netstat -antp | grep httpd
tcp        0      0 0.0.0.0:80              0.0.0.0:*               LISTEN      14015/httpd
```

위에서와 같이 프로세스 목록과 LISTEN 중인 포트가 확인되면, apache 데몬은 정상적으로 구동중이라고 보면 됩니다.

또한 서버가 부팅 시에 자동으로 apache 데몬이 구동되도록 하기 위해서는 아래와 같이 설정합니다.

```
root@local:/
[root@localhost~]# echo "/usr/local/apache2/bin/apachectl start" >> /etc/rc.d/rc.local
[root@localhost~]# cat /etc/rc.d/rc.local
#!/bin/sh
#
# This script will be executed *after* all the other init scripts.
# You can put your own initialization stuff in here if you don't
# want to do the full Sys V style init stuff.

touch /var/lock/subsys/local

/usr/local/mysql/share/mysql/mysql.server start
/usr/local/apache2/bin/apachectl start
```

1.3 php 설치

앞선 apache 설치 방법과 같이 소스 파일을 압축 해제 후 해당 디렉토리로 이동합니다.

```
root@local:/
[root@localhost~]# tar zxvf php-5.2.13.tar.gz
php-5.2.13/
php-5.2.13/ext/
php-5.2.13/ext/gd/
php-5.2.13/ext/gd/gd.c
php-5.2.13/ext/gd/gd_ctx.c
php-5.2.13/ext/gd/libgd/
php-5.2.13/ext/gd/libgd/gd.c
php-5.2.13/ext/gd/libgd/gd.h
php-5.2.13/ext/gd/libgd/gdtables.c
php-5.2.13/ext/gd/libgd/gd_arc.c
php-5.2.13/ext/gd/libgd/gd_gd2.c
php-5.2.13/ext/gd/libgd/gd_png.c
php-5.2.13/ext/gd/libgd/pngtogd2.c
php-5.2.13/ext/gd/libgd/gddemo.c
.....
.....
.....
```

설치 순서는 ./configure, make , make install 순이며, 컴파일 옵션은 ./configure --help 또는 INSTALL 파일을 참조하시기 바랍니다.

```
root@local:/
[root@localhost~]# vi INSTALL
.....
.....
.....
10. Now, configure your PHP.  This is where you customize your PHP
    with various options, like which extensions will be enabled.  Do a
    ./configure --help for a list of available options.  In our example
    we'll do a simple configure with Apache 1 and MySQL support.  Your
    path to apxs may differ from our example.

      ./configure --with-mysql --with-apxs=/www/bin/apxs

11. make
12. make install
.....
.....
.....
```

일반적인 php 운용 환경에서는 아래와 같이 관련 라이브러리를 설치 후 ./configure 를 수행합니다.

```
root@local:/
[root@localhost~]# yum -y install gd-* freetype* libpng* libjpeg* libtiff* libjpeg* libc-cli*
.....
.....
.....


[root@localhost~]# ./configure --prefix=/usr/local/php --with-mysql=/usr/local/mysql --with-
apxs2=/usr/local/apache2/bin/apxs --enable-sysvshm=yes --enable-sysvsem=yes --enable-
debug=no --with-png-dir=/usr --with-zlib-dir --with-jpeg-dir=/usr --enable-mbstring --
enable-sockets --with-freetype-dir=/usr --enable-mbregex --enable-exif --with-gd --
enable-gd-native-ttf --enable-calendar --with-openssl=/usr

.....
.....
.....

creating main/internal_functions_cli.c

+--------------------------------------------------------------------+

| License:                                                           |

| This software is subject to the PHP License, available in this     |

| distribution in the file LICENSE.  By continuing this installation |

| process, you are bound by the terms of this license agreement.     |

| If you do not agree with the terms of this license, you must abort |

| the installation process at this point.                            |

+--------------------------------------------------------------------
--------+
```

./configure 가 정상적으로 완료되고 아래와 같이 make, make install 을 수행하면 빌드된 php 관련 파일들이 /usr/local/php 와 /usr/local/apache2/modules 디렉토리로 이동되게 됩니다.

이후 php 의 환경 설정 파일을 생성합니다.

root@local:/

```
[root@localhost~]# make && make install
.....
.....
.....
Build complete.
Don't forget to run 'make test'.

Installing PHP SAPI module:          apache2handler
/usr/local/apache2/build/instdso.sh     SH_LIBTOOL='/usr/local/apache/build/libtool'     libphp5.la
/usr/local/apache2/modules
/usr/local/apache/build/libtool --mode=install cp libphp5.la /usr/local/apache2/modules/
cp .libs/libphp5.so /usr/local/apache2/modules/libphp5.so
cp .libs/libphp5.lai /usr/local/apache2/modules/libphp5.la
libtool: install: warning: remember to run `libtool --finish /usr/local/src/php-5.2.13/libs'
chmod 755 /usr/local/apache2/modules/libphp5.so
[activating module `php5' in /usr/local/apache2/conf/httpd.conf]
Installing PHP CLI binary:           /usr/local/php/bin/
Installing PHP CLI man page:         /usr/local/php/man/man1/
Installing build environment:        /usr/local/php/lib/php/build/
Installing header files:             /usr/local/php/include/php/
Installing helper programs:          /usr/local/php/bin/
You may want to add: /usr/local/php/lib/php to your php.ini include_path


[root@localhost~]# cp php.ini-dist /usr/local/src/php/lib/php.ini
```

이제 httpd.conf 에 php 모듈 로드와 AddType 을 지정하도록 하며, 우선 php5 module 이 정상적으로 LoadModule 이 적용중인지 아래와 같이 확인합니다.

```
root@local:/
[root@localhost~]# vi +53 /usr/local/apache2/conf/httpd.conf

     43 # Dynamic Shared Object (DSO) Support
     44 #
     45 # To be able to use the functionality of a module which was built as a DSO you
     46 # have to place corresponding `LoadModule' lines at this location so the
     47 # directives contained in it are actually available _before_ they are used.
     48 # Statically compiled modules (those listed by `httpd -l') do not need
     49 # to be loaded here.
     50 #
     51 # Example:
     52 # LoadModule foo_module modules/mod_foo.so
     53 LoadModule php5_module          modules/libphp5.so
     54 #
     55

Complete!
[root@local /]# cat /etc/host.conf
order hosts,bind
```

현재까지는 php 로 작성된 프로그램은 아파치에서는 단순한 text 로만 인식이 되므로 apache 가 php 파일을 정상적으로 인식하도록 type 을 설정하고 index 를 설정해야 합니다.

http.conf 파일의 311 번째 라인에 AddType 을 설정하고 167 번째 라인에는 index 를 설정합니다.

```
root@local:/
[root@localhost /]# vi +311 /usr/local/apache2/conf/httpd.conf
     308       AddType application/x-compress .Z
     309       AddType application/x-gzip .gz .tgz
     310
     311       AddType application/x-httpd-php .php .php3 .html .htm .conf .con .db .inc
     312       AddType application/x-httpd-php-source .phps
     313


[root@localhost /]# vi +167 /usr/local/apache2/conf/httpd.conf
     166 <IfModule dir_module>
     167       DirectoryIndex index.html index.htm index.php index.php3 index.phtml index.cgi
     168 </IfModule>
```

이제 /apache 데몬을 restart 하게 되면 apache, php, mysql 은 DSO 방식으로 정상적으로 연동이 완료되었습니다.

1.4 ZendOptimizer 설치

ZendOptimizer 는 Zend Guard 로 인코딩된 php 스크립트를 사용할 수 있게 해주는 프로그램입니다.

우선 압축 파일을 해제한 후 ZendOptimizer 디렉토리로 이동 후 ZendOptimizer 모듈을 아래와 같이 경로에 복제하면 설치가 마무리됩니다.

```
root@local:/
[root@localhost /]# tar zxvf ZendOptimizer-3.3.9-linux-glibc23-i386.tar.gz

[root@localhost /]# cd ZendOptimizer-3.3.9-linux-glibc23-i386
[root@localhost /]# cd ./data/5_2_x_comp
[root@localhost /]# cp ZendOptimizer.so /usr/local/php/lib/
[root@localhost /]# echo "zend_extension=/usr/local/php/lib/ZendOptimizer.so" >>
/usr/local/php/lib/php.ini
```

이제 apache 데몬을 restart 한 후 http://IP/phpinfo.php URI 로 웹 브라우저를 통해서 접속하시어, apache, php, ZendOptimizer 가 정상적으로 작동중인지 확인함으로 apache, php, mysql, ZendOptimizer 설치 작업을 완료합니다.

```
root@local:/
[root@localhost /]# /usr/local/apache2/bin/apachectl restart
[root@localhost /]# cat /usr/local/apache2/htdocs/phpinfo.php
<?
 phpinfo();
?>
```

2. apache 설정

위와 같이 apache 설치가 완료된 후 웹 브라우저를 통해서 접속시에 /usr/local/apache2/htdocs 디렉토리의 index.html 의 내용이 출력됩니다.
실제로 서비스에 이용될 도메인이나 아이피로 연결시 웹 브라우저를 통해서 출력되는 위치를 변경하기 위해서는 apache 에서 설정 변경을 해야 합니다.

2.1 httpd.conf 설정

/usr/local/apache2/conf/httpd.conf 의 내용은 크게 3가지 부분으로 나눌 수 있습니다. 첫 부분은 apache 설치 경로, 기본적으로 서비스할 서비스 포트, 아파치에서 로드할 모듈을 지시하는 등의 환경 설정 부분과 메인 서버 설정 부분, 가상 호스팅에 대한 설정을 할 수 있는 부분이다. 각 부분의 주요 설정 부분의 내용에 대해서 간략히 알아보도록 하겠습니다.

- httpd.conf 파일

```
#
# This is the main Apache HTTP server configuration file.  It contains the
# configuration directives that give the server its instructions.
# See <URL:http://httpd.apache.org/docs/2.2> for detailed information.
# In particular, see
# <URL:http://httpd.apache.org/docs/2.2/mod/directives.html>
# for a discussion of each configuration directive.
#
# Do NOT simply read the instructions in here without understanding
```

```
# what they do.  They're here only as hints or reminders.  If you are unsure
# consult the online docs. You have been warned.
#
# Configuration and logfile names: If the filenames you specify for many
# of the server's control files begin with "/" (or "drive:/" for Win32), the
# server will use that explicit path.  If the filenames do *not* begin
# with "/", the value of ServerRoot is prepended -- so "logs/foo_log"
# with ServerRoot set to "/usr/local/apache2" will be interpreted by the
# server as "/usr/local/apache2/logs/foo_log".

#
# ServerRoot: The top of the directory tree under which the server's
# configuration, error, and log files are kept.
#
# Do not add a slash at the end of the directory path.  If you point
# ServerRoot at a non-local disk, be sure to point the LockFile directive
# at a local disk.  If you wish to share the same ServerRoot for multiple
# httpd daemons, you will need to change at least LockFile and PidFile.
#
# apache 설치시에 지정한 설치 경로입니다.
# 설치시 prefix 를 타 디렉토리로 정의하면 해당 경로가 ServerRoot 로 나타납니다.
ServerRoot "/usr/local/apache2"

#
# Listen: Allows you to bind Apache to specific IP addresses and/or
# ports, instead of the default. See also the <VirtualHost>
# directive.
#
# Change this to Listen on specific IP addresses as shown below to
# prevent Apache from glomming onto all bound IP addresses.
#
#Listen 12.34.56.78:80
# apache 데몬이 동작할 포트 번호를 지정합니다.
# Listen 8080 등으로 타 포트 번호로 변경하여 구동도 가능합니다.
Listen 80

#
# Dynamic Shared Object (DSO) Support
#
# To be able to use the functionality of a module which was built as a DSO you
# have to place corresponding `LoadModule' lines at this location so the
# directives contained in it are actually available _before_ they are used.
# Statically compiled modules (those listed by `httpd -l') do not need
# to be loaded here.
#
# Example:
# LoadModule foo_module modules/mod_foo.so
# apache ./configure 시에 기입하지 않은 동적으로 로드할 모듈을 나열합니다.
# /usr/local/apache2/bin/httpd -l 에 나타나는 내용들은 별도로 나열할 필요가 없습니다.
# 우리는 php 를 DSO 방식으로 apache 설치 후에 적재하였으므로 LoadModule 지시자를 이용하여
# 모듈을 로드하여야 하지만, 앞서 설명한 설치 과정대로 진행하였다면 기본적으로 등재되어 있습니다.
LoadModule php5_module        modules/libphp5.so
#
```

```
<IfModule !mpm_netware_module>
<IfModule !mpm_winnt_module>
#
# If you wish httpd to run as a different user or group, you must run
# httpd as root initially and it will switch.
#
# User/Group: The name (or #number) of the user/group to run httpd as.
# It is usually good practice to create a dedicated user and group for
# running httpd, as with most system services.
#
# apache 데몬이 구동될 때의 소유자, 그룹 설정합니다.
# 일반적으로 nobody, apache, daemon 등이 대표적이며,
# 변경을 원한다면 서버에 존재하는 사용자와 그룹 이름으로 변경도 가능합니다.
User daemon
Group daemon

</IfModule>
</IfModule>

# 'Main' server configuration
#
# The directives in this section set up the values used by the 'main'
# server, which responds to any requests that aren't handled by a
# <VirtualHost> definition.  These values also provide defaults for
# any <VirtualHost> containers you may define later in the file.
#
# All of these directives may appear inside <VirtualHost> containers,
# in which case these default settings will be overridden for the
# virtual host being defined.
#

#
# ServerAdmin: Your address, where problems with the server should be
# e-mailed.  This address appears on some server-generated pages, such
# as error documents.  e.g. admin@your-domain.com
#
# 웹 페이지가 구동시에 에러등이 발생시에 웹페이지에 노출되는 apache 웹 서버 관리자의
# 이메일 주소를 입력합니다.
ServerAdmin you@example.com

#
# ServerName gives the name and port that the server uses to identify itself.
# This can often be determined automatically, but we recommend you specify
# it explicitly to prevent problems during startup.
#
# If your host doesn't have a registered DNS name, enter its IP address here.
#
# http://www.hostway.co.kr/~user1 과 같이 각 계정별로 홈페이지의 로딩이 필요치 않다면
# 기본적으로 주석 처리를 하여도 무방합니다.
#ServerName www.example.com:80

#
```

```
# DocumentRoot: The directory out of which you will serve your
# documents. By default, all requests are taken from this directory, but
# symbolic links and aliases may be used to point to other locations.
#
# 가상 호스트를 설정치 않고 아이피나 도메인 접속시에 출력되는 페이지를 지정합니다.
DocumentRoot "/usr/local/apache2/htdocs"

#
# Each directory to which Apache has access can be configured with respect
# to which services and features are allowed and/or disabled in that
# directory (and its subdirectories).
#
# First, we configure the "default" to be a very restrictive set of
# features.
#
<Directory />
    Options FollowSymLinks
    AllowOverride None
    Order deny,allow
    Deny from all
</Directory>

# 대표적으로 사용 가능한 옵션 지시자는 아래와 같습니다.
# None : 모든 옵션 설정을 허가하지 않음
# All : 모든 옵션 설정을 허용
# Indexes : DirectoryIndex에 지정된 파일이 해당 디렉토리에 없을 경우에 디렉토리의 파일 목록을 나열
# Includes : shtml 파일 사용시 적용
# FollowSymLinks : 심볼릭 링크를 사용할 수 있음
# ExecCGI : CGI 스크립트를 실행시킬 수 있음
# MultiViews : 브라우저의 인코딩에 따라서 각기 다른 언어의 페이지 제공할 수 있음
# AllowOverride : 사용자 인증에 관련된 지시자로 클라이언트가 웹 서버의 특정 디렉토리에 접근할 때
해당 디렉토리에 있는 유저 인증 파일인 .htaccess 의 허용 유무 (None, All)
# Order : 서버가 access 컨드롤을 수행하는 순서를 지정
# Order allow,deny ( allow 기능을 먼저 수행한 후에 deny 기능을 수행)
# Allow from : 나열되는 주소들에 대한 access 허용 (all, 호스트명, IP)
#
# Note that from this point forward you must specifically allow
# particular features to be enabled - so if something's not working as
# you might expect, make sure that you have specifically enabled it
# below.
#

.....
.....
.....

#
# DirectoryIndex: sets the file that Apache will serve if a directory
# is requested.
#
# 디렉토리에 접근시 출력되는 페이지의 순서와 파일명을 설정합니다.
# 아래의 예에서는 index.html index.htm index.php index.php3 index.phtml index.cgi 와 같은 모든
# 파일이 존재시에 맨 좌측에 있는 index.html 파일을 인식합니다. Index.html 이 없다면, index.htm 을
```

```
# 인식하게 됩니다.

<IfModule dir_module>
    DirectoryIndex index.html index.htm index.php index.php3 index.phtml index.cgi
</IfModule>

#
# ErrorLog: The location of the error log file.
# If you do not specify an ErrorLog directive within a <VirtualHost>
# container, error messages relating to that virtual host will be
# logged here.  If you *do* define an error logfile for a <VirtualHost>
# container, that host's errors will be logged there and not here.
#
# 앞서 설정된 DocumentRoot 에 접근 시에 발생되는 error 에 대해서 로그가 생성됩니다.
ErrorLog "logs/error_log"

#
# LogLevel: Control the number of messages logged to the error_log.
# Possible values include: debug, info, notice, warn, error, crit,
# alert, emerg.
#
LogLevel warn
# 로그를 쌓을 수 있는 LogLevel 은 아래와 같으며, emerge 가 가장 적은,
# debug 가 가장 많은 로그가 생성됩니다.
# 사용할 수 있는 것은 emerge, alert, crit, error, warn, notice, info, debug 이며,
# 일반적으로 warn 로그만으로도 충분히 서버의 문제를 파악할 수 있으며, 문제가 발생시에는
# tail -f error_log 와 같이 실시간으로 살펴보는 것이 좋습니다.

<IfModule log_config_module>
    #
    # The following directives define some format nicknames for use with
    # a CustomLog directive (see below).
    #
    LogFormat "%h %l %u %t ₩"%r₩" %>s %b ₩"%{Referer}i₩" ₩"%{User-Agent}i₩"" combined
    LogFormat "%h %l %u %t ₩"%r₩" %>s %b" common

    <IfModule logio_module>
      # You need to enable mod_logio.c to use %I and %O
      LogFormat "%h %l %u %t ₩"%r₩" %>s %b ₩"%{Referer}i₩" ₩"%{User-Agent}i₩" %I %O"
combinedio
    </IfModule>

    #
    # The location and format of the access logfile (Common Logfile Format).
    # If you do not define any access logfiles within a <VirtualHost>
    # container, they will be logged here.  Contrariwise, if you *do*
    # define per-<VirtualHost> access logfiles, transactions will be
    # logged therein and *not* in this file.
    #
    CustomLog "logs/access_log" common

# common 로그 형식(CLF) 이라는 형식으로 로그 항목을 기록한다. 웹서버의 표준 형식으로 쌓이는
# 항목은 아래와 같이 예를 들 수 있습니다.
```

```
# 211.115.223.215 - euntae [10/Mar/2010:12:10:35 +0900] "GET /main_t.png HTTP/1.0" 200 4115
# 위의 로그를 분석해보면 211.115.223.215 (%h) 은 서버에 요청을 한 원격 클라이언트의 IP(호스트명)
# 입니다.. HostnameLookups 가 On 이라면 호스트명(FQDN) 으로 IP 주소가 대체되며, 이 설정은
# hostnamelookup 과정을 추가로 거쳐야하므로 웹서버가 매우 느리게 작동될 수 있으므로, 사용하지
# 않을 것을 추천합니다.
# euntae (%u) 는 HTTP 인증으로 해당 URI 에 요청한 사용자의 userid 입니다.
# [10/Mar/2010:12:10:35 +0900] 는 해당 URI 에 211.114.223.215 에서 접속한 euntae 사용자의 접근
# 요청이 서버에서 처리된 시간을 의미합니다.
# "GET /main_t.png HTTP/1.0" 매우 중요하며 유용한 정보를 담고 있으며, 첫번째로 211.115.223.215
# 에서 접근한 사용자의 메쏘드는 GET 입니다. 즉, 서버에서는 요청에 단순히 해당 파일을 송출했다는
# 의미로, 만약 메쏘드가 POST 라면 211.115.223.215 에서 서버로의 엑션이 이뤄지도록 요청이 있었
# 음을 의미하며, 웹 페이지 속도 저하시에 POST 메쏘드가 없는 파일이나, 페이지에 요청이 있다면
# 서버스 거부 공격 또는 잘못된 웹 프로그램으로 의심할 수 있습니다. 서버 관리자는 매우 주의 깊게
# 살펴 봐야할 부분입니다.
# 200 (%>s) 는 서버가 211.115.223.215 클라이언트에 보낸 상태 코드입니다. 2로 시작하는 코드는
# 요청이 성공했음을 의미하며, 4로 시작되는 코드는 클라이언트에 오류가 있는지, 5로 시작되는 코드는
# 서버에 오류가 있는지 알려주는 매우 유용한 정보 이다. 상태 코드의 목록은 RFC2616 에서도 확인이
# 가능합니다.
# 4115 (%b) 는 응답 헤더의 값을 제외하고 클라이언트에게 보내는 내용의 크기를 의미합니다.
#
# combined 로그 형식은 다른 형식문자열이 결합된 로그 형식(Combined Log Format)입니다.
# 사용하는 방법은 common 을 combined 로 변경하여 설정이 가능하며, 일반적으로 Referer (직전에
# 방문하여 접근한 URI) 와 User-Agent (클라이언트의 브라우저명과 버전) 값을 확인시에 설정하여
# 많이 이용됩니다.
    #
    # If you prefer a logfile with access, agent, and referer information
    # (Combined Logfile Format) you can use the following directive.
    #
    #CustomLog "logs/access_log" combined
</IfModule>

<IfModule alias_module>
    #
    # Redirect: Allows you to tell clients about documents that used to
    # exist in your server's namespace, but do not anymore. The client
    # will make a new request for the document at its new location.
    # Example:
    # Redirect permanent /foo http://www.example.com/bar
# Redirect 지시자는 이전 요청 URI 를 새로운 URI 로 전달해줍니다.
# 위의 내용은 http://도메인/foo/a.htm 를 요청하면 대신 http://www.example.com/bar/a.htm
# 에 접근하라는 응답을 주게 됩니다.

    #
    # Alias: Maps web paths into filesystem paths and is used to
    # access content that does not live under the DocumentRoot.
    # Example:
    # Alias /webpath /full/filesystem/path
    #
    # If you include a trailing / on /webpath then the server will
    # require it to be present in the URL.  You will also likely
    # need to provide a <Directory> section to allow access to
    # the filesystem path.
# Alias 지시자는 해당 alias 로 접속시에 지정한 서버의 절대경로의 내용이 출력되도록 전달해줍니다.
```

```
# 예를 들어서 Alias /images /bighard/images 라고 설정하였다면, http://도메인/images/ver2/file.htm
# 을 요청하면 /bighard/images/vers2/file.htm 이 응답합니다.
# 즉, 해당 도메인의 DocumentRoot 아래에 컨텐츠를 둘 수 없는 상황에서는 매우 유용하게
# 활용할 수 있습니다.

    #
    # ScriptAlias: This controls which directories contain server scripts.
    # ScriptAliases are essentially the same as Aliases, except that
    # documents in the target directory are treated as applications and
    # run by the server when requested rather than as documents sent to the
    # client.  The same rules about trailing "/" apply to ScriptAlias
    # directives as to Alias.
    #
    ScriptAlias /cgi-bin/ "/usr/local/apache2/cgi-bin/"

</IfModule>

.....
.....
.....


#
# DefaultType: the default MIME type the server will use for a document
# if it cannot otherwise determine one, such as from filename extensions.
# If your server contains mostly text or HTML documents, "text/plain" is
# a good value.  If most of your content is binary, such as applications
# or images, you may want to use "application/octet-stream" instead to
# keep browsers from trying to display binary files as though they are
# text.
#
DefaultType text/plain

<IfModule mime_module>
    #
    # TypesConfig points to the file containing the list of mappings from
    # filename extension to MIME-type.
    #
    TypesConfig conf/mime.types
# 웹 서버가 데이터를 전송하는 문서 형식(mime type) 을 정의해둔 파일입니다.
# 추가가 필요시에는 /usr/local/apache2/conf/mime.types 하단에 기입을 해주시면 됩니다.

    #
    # AddType allows you to add to or override the MIME configuration
    # file specified in TypesConfig for specific file types.
    #
    #AddType application/x-gzip .tgz
    #
    # AddEncoding allows you to have certain browsers uncompress
    # information on the fly. Note: Not all browsers support this.
    #
    #AddEncoding x-compress .Z
    #AddEncoding x-gzip .gz .tgz
```

```
    #
    # If the AddEncoding directives above are commented-out, then you
    # probably should define those extensions to indicate media types:
    #
    AddType application/x-compress .Z
    AddType application/x-gzip .gz .tgz

    AddType application/x-httpd-php .php .php3 .html .htm .conf .con .db .inc
    AddType application/x-httpd-php-source .phps
# 기본적으로 apache 는 php 의 문서 형식을 인식하지 못하고 text 파일로 인식하므로, 위와 같이
# apache 에서 php 를 인식하도록 mime type 을 설정해야 합니다.
# apache 와 php 설치 후 웹 페이지로 php 파일을 요청했을 경우, 웹 브라우저에 php 소스 코드가
# 그대로 출력된다면 위와 같은 mime type 설정을 하지 않았음을 의미합니다.
# AddType application/x-httpd-php .php .php3 .html .htm .conf .con .db .inc 는 해당 확장자명에 php
# 코드가 입력되어 있다면 php 문서라고 인식시키며, AddType application/x-httpd-php-source .phps
# 는 .phps 확장자를 갖는 파일은 text 로 출력하라는 의미입니다.

    #
    # AddHandler allows you to map certain file extensions to "handlers":
    # actions unrelated to filetype. These can be either built into the server
    # or added with the Action directive (see below)
    #
    # To use CGI scripts outside of ScriptAliased directories:
    # (You will also need to add "ExecCGI" to the "Options" directive.)
    #
    #AddHandler cgi-script .cgi

    # For type maps (negotiated resources):
    #AddHandler type-map var

    #
    # Filters allow you to process content before it is sent to the client.
    #
    # To parse .shtml files for server-side includes (SSI):
    # (You will also need to add "Includes" to the "Options" directive.)
    #
    #AddType text/html .shtml
    #AddOutputFilter INCLUDES .shtml
</IfModule>

#
# The mod_mime_magic module allows the server to use various hints from the
# contents of the file itself to determine its type.  The MIMEMagicFile
# directive tells the module where the hint definitions are located.
#
#MIMEMagicFile conf/magic

#
# Customizable error responses come in three flavors:
# 1) plain text 2) local redirects 3) external redirects
#
# Some examples:
#ErrorDocument 500 "The server made a boo boo."
```

```
#ErrorDocument 404 /missing.html
#ErrorDocument 404 "/cgi-bin/missing_handler.pl"
#ErrorDocument 402 http://www.example.com/subscription_info.html
#
# 위는 각각의 에러 상태에 따라서 웹 브라우저에 출력해줄 내용을 설정할 수 있습니다.
# 단순히 text 문서를 출력할 수 있으며, 서버내의 특정 웹 페이지 호출, cgi 스크립트, 타 URI 로
# 이동시킬 수 있습니다.
# 예를 들어서, 실제 존재하지 않는 파일로의 접근시에 도메인의 DocumentRoot 로 이동하는 방법은
# 이와 같이 작성할 수 있습니다.
# ErrorDocument 404 http://도메인/

#
# EnableMMAP and EnableSendfile: On systems that support it,
# memory-mapping or the sendfile syscall is used to deliver
# files.  This usually improves server performance, but must
# be turned off when serving from networked-mounted
# filesystems or if support for these functions is otherwise
# broken on your system.
#
#EnableMMAP off
#EnableSendfile off
# EnableMMAP 는 커널의 매커니즘을 정확히 알고 사용해야 합니다. Apache core 문서상으로는
# Apache 의 성능 향상을 할 수 있으나, 일반적인 경우에서는 apache 의 성능 저하가 발생되며,
# segfault 등이 발생되면서 crash 되는 경우가 많다고 합니다. 기본 값을 유지하시기 바랍니다.


# 아래에 나오는 Include 지시자로 불러오는 값들은, Include 되는 설정의 특성에 맞춰서 별도로 작성된
# 환경 설정 파일입니다. 별도로 불리하지 않아도 상관이 없으나, apache 버전 변경 등의 서버 작업시
# 별도로 불리하여 Include 하여 사용하였다면, 작업 시간 단축에 이로움이 있습니다.

# 아래에서는 간략히 알아보고 별도로 설명하도록 하겠습니다.
# Supplemental configuration
#
# The configuration files in the conf/extra/ directory can be
# included to add extra features or to modify the default configuration of
# the server, or you may simply copy their contents here and change as
# necessary.

# Server-pool management (MPM specific)
#Include conf/extra/httpd-mpm.conf
# 다중 처리 모듈을 설정하는데 이용됩니다.
# 별도로 살펴보도록 하겠습니다.

# Multi-language error messages
#Include conf/extra/httpd-multilang-errordoc.conf
# 에러 메시지를 요청한 웹 브라우저의 언어셋에 맞춰서 출력되도록 설정하는데 이용됩니다.

# Fancy directory listings
#Include conf/extra/httpd-autoindex.conf
# 요청한 웹 브라우저에게 보여줄 디렉토리 목록을 설정하는 파일입니다.

# Language settings
#Include conf/extra/httpd-languages.conf
```

```
# apache 의 다중 언어 지원을 위한 파일입니다.

# User home directories
#Include conf/extra/httpd-userdir.conf
# 사용자의 홈 디렉토리 설정 시에 이용됩니다.

# Real-time info on requests and configuration
#Include conf/extra/httpd-info.conf
# apache 의 구동 상태와 환경 설정 등에 대해서 보여주는 설정 파일입니다.

# Virtual hosts
#Include conf/extra/httpd-vhosts.conf
# 가장 많이 이용되는 VirtualHost 설정 시에 이용되는 환경 설정 파일입니다.
# 별도로 살펴보겠습니다.

# Local access to the Apache HTTP Server Manual
#Include conf/extra/httpd-manual.conf
# apache 의 매뉴얼을 직접 본 서버에서 제공하기 위해서 설정하는 파일입니다.

# Distributed authoring and versioning (WebDAV)
#Include conf/extra/httpd-dav.conf
# 웹 페이지 개발 도구 WebDAV 를 사용할 수 있도록 설정하는 파일입니다.

# Various default settings
#Include conf/extra/httpd-default.conf
# apache 기본 구동 환경 설정 파일입니다.

# Secure (SSL/TLS) connections
#Include conf/extra/httpd-ssl.conf
# apache 에서 SSL 인증서 서비스에 설정하는 파일입니다.
#
# Note: The following must must be present to support
#       starting without SSL on platforms with no /dev/random equivalent
#       but a statically compiled-in mod_ssl.
#
<IfModule ssl_module>
SSLRandomSeed startup builtin
SSLRandomSeed connect builtin
</IfModule>
```

- httpd-mpm.conf

```
# apache 웹 서버가 구동될 때에 프로세스는 몇 개로 구동할지, 자식 프로세스는 몇 개로 할 것인가
# 등에 대해서 설정하는 파일입니다.

# 사용되는 MPM 의 종류는 prefork, worker, BeOS, NetWare, OS/2, WinNT MPM 이 있으며, apache
# 데몬이 구동되는 OS 환경에 따라 자동으로 MPM 은 선택됩니다.
# CentOS-5.4 리눅스에 apache 를 설치하였으므로, 기본적으로 apache 데몬은 prefork MPM 이 선택
# 됩니다.

<IfModule mpm_prefork_module>
    StartServers          5
    # apache 가 구동될 때에 시작되는 프로세스의 개수를 의미합니다. 특별히 변경은 불필요합니다.
```

```
    MinSpareServers         5
    # 빠른 응답 속도를 위해서 대기중인 프로세스의 최저 개수를 의미합니다. 이보다 작다면 추가로
    # httpd 가 생성되며, 특별히 변경은 불필요합니다.
    MaxSpareServers         10
    # MinSpareServers 의 반대 개념으로 대기중인 프로세스의 최대 개수를 의미합니다. 이보다 많다면
    # 자동으로 apache 가 kill 하게 됩니다.
    MaxClients              150
    # apache 에 접속하는 웹 클라이언트의 최대 개수를 제한하는 의미입니다.
    # 제한하지 않으며, 무한정으로 웹 서버의 하드웨어 자원을 소모하여, 결국에는 서버가 다운되므로
    # 이를 방지하기 위해서 최대 접속할 수 있는 웹 클라이언트의 개수를 제한 설정합니다.
    MaxRequestsPerChild     0
    # apche 의 자식 프로세스들이 웹 클라이언트의 요청을 받을 수 있는 개수를 지정합니다.
    # 위에 설정된 0 은 제한을 두지 않음을 의미합니다.
</IfModule>

# 서버의 사양에 따라 유동적으로 값들을 변경할 수 있으나, apache 컴파일시에 기본적으로 컴파일
# 하였다면, MaxClients 는 최대 256 까지만 지정이 가능합니다.
# 변경을 위해서는 "apache소스/server/mpm/prefork/prefork.c" 파일에서 DEFAULT_SERVER_LIMIT
# 256 값을 변경 후에 apache 를 컴파일해야 합니다.
# 고 사양의 웹 서버라도 512 이상은 무의미합니다. 가급적 256 을 최대 값으로 지정하시고,
# 하드웨어 L4 스위치나 오픈 소스 LVS 등을 이용하여 로드벨런싱을 적용하고 웹 서버를 증설하여
# 웹 서비스를 하실 것을 추천드립니다.
```

- httpd-vhosts.conf

```
# 여러 개의 도메인이나 호스트를 apache 를 통해서 웹 서비스를 하려면, VirtualHost 설정을
# 해야하는데, 이때 설정하는 파일이며, 활성화를 위해서는 httpd.conf 파일에서 아래와 같이
# 주석을 해제합니다.
Include conf/extra/httpd-vhosts.conf
# 위와 같이 주석 해제를 해야만 httpd-vhosts.conf 파일의 설정 내용을 apache 가 인식하게 됩니다.

# Please see the documentation at
# <URL:http://httpd.apache.org/docs/2.2/vhosts/>
# for further details before you try to setup virtual hosts.
#
# You may use the command line option '-S' to verify your virtual host
# configuration.

#
# Use name-based virtual hosting.
#
NameVirtualHost *:80

#
# VirtualHost example:
# Almost any Apache directive may go into a VirtualHost container.
# The first VirtualHost section is used for all requests that do not
# match a ServerName or ServerAlias in any <VirtualHost> block.
#
<VirtualHost *:80>
    ServerAdmin webmaster@dummy-host.example.com
    # ServerAdmin 에는 해당 도메인의 관리자 이메일 주소를 입력합니다.
    DocumentRoot "/usr/local/src/apache2/docs/dummy-host.example.com"
```

```
    # 해당 도메인에서 서비스할 웹 소스의 index 파일의 위치를 절대 경로로 입력합니다.
    ServerName dummy-host.example.com
    # ServerName www 호스트를 제외하고 입력합니다.
    ServerAlias www.dummy-host.example.com
    # ServerAlias 에는 www 를 포함하여 입력합니다. 또는 제공되는 웹 소스가 동일하다면 FQDN
    # 으로 나열하시면 되겠습니다.(예, www.dummy-host2.example.com, dumy-host2.example.com)
    ErrorLog "logs/dummy-host.example.com-error_log"
    # 해당 도메인으로 접근 시에 발생되는 error 들을 저장할 파일명을 입력합니다.
    CustomLog "logs/dummy-host.example.com-access_log" common
    # 해당 도메인으로 접근 시에 발생되는 error 를 제외한 모든 로그가 저장될 파일명을 입력합니다.
</VirtualHost>

# 실제 적용 예는 다음 장에서 설명하도록 하겠습니다.
```

2.2 가상 호스트 설정
다음은 실제 작동할 수 있는 가상 호스트를 설정해보도록 하겠습니다.
조건은 다음과 같습니다.

```
도메인 관리자 이메일 주소 : webadmin@hostway.co.kr
홈페이지 파일들이 존재하는 곳 : /home/hostway/public_html
도메인 네임 : hostway.co.kr, www.hostway.co.kr
```

```
root@local:/
[root@localhost~]# vi +392 /usr/local/apache2/conf/httpd.conf

391 # Virtual hosts
392 Include conf/extra/httpd-vhosts.conf
```

와 같이 /usr/local/apache2/conf/httpd.conf 파일의 392 번째 라인을 주석 해제 처리합니다.(주석 해제 방법은 “#”을 라인에서 삭제합니다.)

두번째로 /usr/local/apache2/conf/extra/httpd-vhosts.conf 파일의 최상단에 다음와 같은 내용을 추가로 입력합니다.

```
<Directory "/home/*">
        AllowOverride None
        Options +ExecCGI MultiViews
        Order allow,deny
        Allow from all
</Directory>
```

이제 VirtualHost 를 설정 추가한 후에 apache 를 restart 해보도록 하겠습니다.

root@local:/

```
[root@localhost~]# vi /usr/local/apache2/conf/extra/httpd-vhosts.conf
.....
.....
<Directory "/home/*">
        AllowOverride None
        Options +ExecCGI MultiViews
        Order allow,deny
        Allow from all
</Directory>

#
# Use name-based virtual hosting.
#
NameVirtualHost *:80

.....
.....
.....
# 아래의 내용을 추가합니다.
<VirtualHost *:80>
    ServerAdmin webadmin@hostway.co.kr
    DocumentRoot "/home/hostway/public_html"
    ServerName hostway.co.kr
    ServerAlias www.hostway.co.kr
    ErrorLog "logs/www.hostway.co.kr-error_log"
    CustomLog "logs/www.hostway.co.kr-access_log" common
</VirtualHost>


[root@localhost~]# /usr/local/apache2/bin/apachectl restart
```

이제 웹 브라우저를 통해서 http://www.hostway.co.kr 나 http://hostway.co.kr 로 접속을 해 봅니다.

2.3 인증서 적용

SSL (Secure Sockets Layer) 는 웹 서버와 클라이언트의 웹 브라우저 사이에서 발생되는 패킷을 암호화를 하여 민감한 정보의 노출을 막는 데에 사용됩니다.

현재 국내는 2007년부터 영리를 목적으로 개인정보를 수집하는 모든 온라인 사이트의 SSL 보안 서버 구축이 의무화되어, SSL 보안 서버를 구축하지 않는 경우 “정보통신망 이용촉진 및 정보보호 등에 관한 법률” 에 의해 1천만원 이하의 과태료 등 행정조치가 시행되고 있습니다.

SSL 인증서의 종류 및 가격은 http://hostway.co.kr/server/secure/securi_ssl_price.html 페이지를 참고해주시기 바랍니다.
SSL 보안 서버를 적용하여 이용하는 방법은 아래와 같습니다.

[그림 4-1] SSL 보안 서버 적용 방법

즉, 서버 관리자는 서버에서 CSR 코드를 생성한 후 호스트웨이에서 검증 후 인증서를 발급받고, 발급받으신 인증서를 apche 웹 서버에 적용하는 기술적인 조치를 해야 합니다.

이제 자세히 알아보도록 하겠습니다.

httpd.conf 파일에서 Include conf/extra/httpd-ssl.conf 를 활성화 한 후에 설정하는 방식입니다.

```
root@local:/
[root@localhost /]# vi +404 /usr/local/apache2/conf/httpd.conf
…….
……
403 # Secure (SSL/TLS) connections
404 #Include conf/extra/httpd-ssl.conf
```

이제, 서버키와 CSR 코드를 생성합니다.

```
root@local:/
[root@localhost /]# cd /usr/local/apache2/conf
[root@localhost /]# openssl genrsa 1024 > www.hostway.co.kr.key
Generating RSA private key, 1024 bit long modulus
.............++++++
.....................................++++++
e is 65537 (0x10001)
[root@localhost~]# openssl req -new -key www.hostway.co.kr.key > www.hostway.co.kr.csr
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [GB]:KR
State or Province Name (full name) [Berkshire]:Seoul
Locality Name (eg, city) [Newbury]:Seoul
Organization Name (eg, company) [My Company Ltd]:HOSTWAY Korea
Organizational Unit Name (eg, section) []:System Division
Common Name (eg, your name or your server's hostname) []:www.hostway.co.kr
Email Address []:webadmin@hostway.co.kr

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:

[root@localhost /]# cat www.hostway.co.kr.csr
-----BEGIN CERTIFICATE REQUEST-----
MIIB4zCCAUwCAQAwgaIxCzAJBgNVBAYTAktSMQ4wDAYDVQQIEwVTZW91bDEOMAwG
.....
.....
rRdjPz5rmA==
-----END CERTIFICATE REQUEST-----
```

서버키는 구분하기 쉽게 www.hostway.co.kr.key 로 생성하였으며, 생성한 서버키를 이용하여 www.hostway.co.kr.csr 과 같이 CSR 코드를 생성하였습니다.

CSR 코드 입력시에는 Country Name 에 KR 을, Organization Name 에는 영문 회사 명칭을, Organizational Unit Name 에는 영문 부서명을, Email Address 에는 관리자 이메일 주소를 입력하는데, Common Name 입력이 가장 중요하며, 인증서를 받을 호스트명을 정확히 FQDN 으로 입력합니다.

이제 생성된 www.hostway.co.kr.csr 파일을 호스트웨이 영업팀 SSL 인증서 담당자에게 전달해 주시고, 결제가 완료되면 www_hostway_co_kr.zip 파일을 받아보실 수 있습니다.

압축을 해제하시면 총 2개의 파일이 존재합니다. FTP 등을 이용하여 /usr/local/apache2/conf/

디렉토리 내에 전송합니다.
www_hostway_co_kr.ca-bundle 는 CA(인증서 발행사) 의 인증 파일이며, www_hostway_co_kr.crt 는 정식 SSL 인증서 입니다.

이제, /usr/local/apache2/conf/extra/httpd-ssl.conf 파일을 설정해보도록 하겠습니다.

● httpd-ssl.conf

```
#
# This is the Apache server configuration file providing SSL support.
# It contains the configuration directives to instruct the server how to
# serve pages over an https connection. For detailing information about these
# directives see <URL:http://httpd.apache.org/docs/2.2/mod/mod_ssl.html>
#
# Do NOT simply read the instructions in here without understanding
# what they do.  They're here only as hints or reminders.  If you are unsure
# consult the online docs. You have been warned.
#

#
# Pseudo Random Number Generator (PRNG):
# Configure one or more sources to seed the PRNG of the SSL library.
# The seed data should be of good random quality.
# WARNING! On some platforms /dev/random blocks if not enough entropy
# is available. This means you then cannot use the /dev/random device
# because it would lead to very long connection times (as long as
# it requires to make more entropy available). But usually those
# platforms additionally provide a /dev/urandom device which doesn't
# block. So, if available, use this one instead. Read the mod_ssl User
# Manual for more details.
#
#SSLRandomSeed startup file:/dev/random  512
#SSLRandomSeed startup file:/dev/urandom 512
#SSLRandomSeed connect file:/dev/random  512
#SSLRandomSeed connect file:/dev/urandom 512


#
# When we also provide SSL we have to listen to the
# standard HTTP port (see above) and to the HTTPS port
#
# Note: Configurations that use IPv6 but not IPv4-mapped addresses need two
#       Listen directives: "Listen [::]:443" and "Listen 0.0.0.0:443"
#

NameVirtualHost *:443
# NameVirtualHost *:443 을 추가합니다.

Listen 443

##
##  SSL Global Context
```

```
##
##  All SSL configuration in this context applies both to
##  the main server and all SSL-enabled virtual hosts.
##

#
#  Some MIME-types for downloading Certificates and CRLs
#
AddType application/x-x509-ca-cert .crt
AddType application/x-pkcs7-crl    .crl

#  Pass Phrase Dialog:
#  Configure the pass phrase gathering process.
#  The filtering dialog program (`builtin' is a internal
#  terminal dialog) has to provide the pass phrase on stdout.
SSLPassPhraseDialog  builtin

#  Inter-Process Session Cache:
#  Configure the SSL Session Cache: First the mechanism
#  to use and second the expiring timeout (in seconds).
#SSLSessionCache         "dbm:/usr/local/apache2/logs/ssl_scache"
SSLSessionCache        "shmcb:/usr/local/apache2/logs/ssl_scache(512000)"
SSLSessionCacheTimeout  300

#  Semaphore:
#  Configure the path to the mutual exclusion semaphore the
#  SSL engine uses internally for inter-process synchronization.
SSLMutex  "file:/usr/local/apache2/logs/ssl_mutex"

##
## SSL Virtual Host Context
##

<VirtualHost *:443>
# VirtualHost *:443 으로 수정을 합니다.

#  General setup for the virtual host
DocumentRoot "/home/hostway/public_html"
ServerName www.hostway.co.kr
ServerAdmin webadmin@hostway.co.kr
ErrorLog "/usr/local/apache2/logs/www.hostway.co.kr-error_log"
TransferLog "/usr/local/apache2/logs/www.hostway.co.kr-access_log"
# 앞서 가상 호스트 설정 시에 등록한 것과 같이 www.hostway.co.kr 호스트의 가상 호스트를 설정
# 합니다.

#  SSL Engine Switch:
#  Enable/Disable SSL for this virtual host.
SSLEngine on

#  SSL Cipher Suite:
#  List the ciphers that the client is permitted to negotiate.
#  See the mod_ssl documentation for a complete list.
SSLCipherSuite ALL:!ADH:!EXPORT56:RC4+RSA:+HIGH:+MEDIUM:+LOW:+SSLv2:+EXP:+eNULL
```

```
#    Server Certificate:
#    Point SSLCertificateFile at a PEM encoded certificate.  If
#    the certificate is encrypted, then you will be prompted for a
#    pass phrase.  Note that a kill -HUP will prompt again.  Keep
#    in mind that if you have both an RSA and a DSA certificate you
#    can configure both in parallel (to also allow the use of DSA
#    ciphers, etc.)
SSLCertificateFile "/usr/local/apache2/conf/www_hostway_co_kr.crt"
# 정식 SSL 인증서의 위치를 절대 경로로 입력합니다.

#SSLCertificateFile "/usr/local/apache2/conf/server-dsa.crt"

#    Server Private Key:
#    If the key is not combined with the certificate, use this
#    directive to point at the key file.  Keep in mind that if
#    you've both a RSA and a DSA private key you can configure
#    both in parallel (to also allow the use of DSA ciphers, etc.)
SSLCertificateKeyFile "/usr/local/apache2/conf/www.hostway.co.kr.key"
# 서버키의 위치를 절대 경로로 입력합니다.

#SSLCertificateKeyFile "/usr/local/apache2/conf/server-dsa.key"

#    Server Certificate Chain:
#    Point SSLCertificateChainFile at a file containing the
#    concatenation of PEM encoded CA certificates which form the
#    certificate chain for the server certificate. Alternatively
#    the referenced file can be the same as SSLCertificateFile
#    when the CA certificates are directly appended to the server
#    certificate for convinience.
#SSLCertificateChainFile "/usr/local/apache2/conf/server-ca.crt"

#    Certificate Authority (CA):
#    Set the CA certificate verification path where to find CA
#    certificates for client authentication or alternatively one
#    huge file containing all of them (file must be PEM encoded)
#    Note: Inside SSLCACertificatePath you need hash symlinks
#          to point to the certificate files. Use the provided
#          Makefile to update the hash symlinks after changes.
#SSLCACertificatePath "/usr/local/apache2/conf/ssl.crt"
SSLCACertificateFile "/usr/local/apache2/conf/www_hostway_co_kr.ca-bundle"
# CA(인증서 발행사) 에서 발급한 인증 파일을 절대 경로명으로 입력합니다.

#    Certificate Revocation Lists (CRL):
#    Set the CA revocation path where to find CA CRLs for client
#    authentication or alternatively one huge file containing all
#    of them (file must be PEM encoded)
#    Note: Inside SSLCARevocationPath you need hash symlinks
#          to point to the certificate files. Use the provided
#          Makefile to update the hash symlinks after changes.
#SSLCARevocationPath "/usr/local/apache2/conf/ssl.crl"
#SSLCARevocationFile "/usr/local/apache2/conf/ssl.crl/ca-bundle.crl"
```

```
#    Client Authentication (Type):
#    Client certificate verification type and depth.  Types are
#    none, optional, require and optional_no_ca.  Depth is a
#    number which specifies how deeply to verify the certificate
#    issuer chain before deciding the certificate is not valid.
#SSLVerifyClient require
#SSLVerifyDepth  10

#    Access Control:
#    With SSLRequire you can do per-directory access control based
#    on arbitrary complex boolean expressions containing server
#    variable checks and other lookup directives.  The syntax is a
#    mixture between C and Perl.  See the mod_ssl documentation
#    for more details.
#<Location />
#SSLRequire (    %{SSL_CIPHER} !~ m/^(EXP|NULL)/ ₩
#            and %{SSL_CLIENT_S_DN_O} eq "Snake Oil, Ltd." ₩
#            and %{SSL_CLIENT_S_DN_OU} in {"Staff", "CA", "Dev"} ₩
#            and %{TIME_WDAY} >= 1 and %{TIME_WDAY} <= 5 ₩
#            and %{TIME_HOUR} >= 8 and %{TIME_HOUR} <= 20       ) ₩
#           or %{REMOTE_ADDR} =~ m/^192₩.76₩.162₩.[0-9]+$/
#</Location>

#    SSL Engine Options:
#    Set various options for the SSL engine.
#    o FakeBasicAuth:
#      Translate the client X.509 into a Basic Authorisation.  This means that
#      the standard Auth/DBMAuth methods can be used for access control.  The
#      user name is the `one line' version of the client's X.509 certificate.
#      Note that no password is obtained from the user. Every entry in the user
#      file needs this password: `xxj31ZMTZzkVA'.
#    o ExportCertData:
#      This exports two additional environment variables: SSL_CLIENT_CERT and
#      SSL_SERVER_CERT. These contain the PEM-encoded certificates of the
#      server (always existing) and the client (only existing when client
#      authentication is used). This can be used to import the certificates
#      into CGI scripts.
#    o StdEnvVars:
#      This exports the standard SSL/TLS related `SSL_*' environment variables.
#      Per default this exportation is switched off for performance reasons,
#      because the extraction step is an expensive operation and is usually
#      useless for serving static content. So one usually enables the
#      exportation for CGI and SSI requests only.
#    o StrictRequire:
#      This denies access when "SSLRequireSSL" or "SSLRequire" applied even
#      under a "Satisfy any" situation, i.e. when it applies access is denied
#      and no other module can change it.
#    o OptRenegotiate:
#      This enables optimized SSL connection renegotiation handling when SSL
#      directives are used in per-directory context.
#SSLOptions +FakeBasicAuth +ExportCertData +StrictRequire
<FilesMatch "₩.(cgi|shtml|phtml|php)$">
    SSLOptions +StdEnvVars
```

```
</FilesMatch>
<Directory "/usr/local/apache2/cgi-bin">
    SSLOptions +StdEnvVars
</Directory>

#   SSL Protocol Adjustments:
#   The safe and default but still SSL/TLS standard compliant shutdown
#   approach is that mod_ssl sends the close notify alert but doesn't wait for
#   the close notify alert from client. When you need a different shutdown
#   approach you can use one of the following variables:
#   o ssl-unclean-shutdown:
#     This forces an unclean shutdown when the connection is closed, i.e. no
#     SSL close notify alert is send or allowed to received.  This violates
#     the SSL/TLS standard but is needed for some brain-dead browsers. Use
#     this when you receive I/O errors because of the standard approach where
#     mod_ssl sends the close notify alert.
#   o ssl-accurate-shutdown:
#     This forces an accurate shutdown when the connection is closed, i.e. a
#     SSL close notify alert is send and mod_ssl waits for the close notify
#     alert of the client. This is 100% SSL/TLS standard compliant, but in
#     practice often causes hanging connections with brain-dead browsers. Use
#     this only for browsers where you know that their SSL implementation
#     works correctly.
#   Notice: Most problems of broken clients are also related to the HTTP
#   keep-alive facility, so you usually additionally want to disable
#   keep-alive for those clients, too. Use variable "nokeepalive" for this.
#   Similarly, one has to force some clients to use HTTP/1.0 to workaround
#   their broken HTTP/1.1 implementation. Use variables "downgrade-1.0" and
#   "force-response-1.0" for this.
BrowserMatch ".*MSIE.*" ₩
         nokeepalive ssl-unclean-shutdown ₩
         downgrade-1.0 force-response-1.0

#   Per-Server Logging:
#   The home of a custom SSL log file. Use this when you want a
#   compact non-error SSL logfile on a virtual host basis.
CustomLog "/usr/local/apache2/logs/ssl_request_log" ₩
          "%t %h %{SSL_PROTOCOL}x %{SSL_CIPHER}x ₩"%r₩" %b"

</VirtualHost>
```

위와 같이 설정 후 apache 데몬을 restart 하시고 아래와 같이 netstat -antp | grep LISTEN 시에 443 포트가 아래와 같이 확인되면 되겠습니다.

```
root@local:/
[root@localhost /]# /usr/local/apache2/bin/apachectl restart
[root@localhost /]# netstat -antp | grep LISTEN
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN      1820/sshd
tcp        0      0 0.0.0.0:80              0.0.0.0:*               LISTEN      12582/httpd
tcp        0      0 0.0.0.0:443             0.0.0.0:*               LISTEN      12582/httpd
```

위와 같은 상태에서 https://www.hostway.co.kr 로 웹 브라우저를 통해서 접속이 가능함을 확인합니다.

만약 연결이 이뤄지지 않는다면 방화벽에서 TCP/443 을 연결 허용 설정을 하시면 되겠습니다.

이제 고단했던 서버 관리자의 기술적인 조치는 모두 완료 되었습니다. 홈페이지를 유지 보수하는 웹 프로그램머에게 해당 사항을 안내한 후에, 법률로 규정하고 있는 SSL 통신이 필요한 웹 소스를 수정하실 것을 안내하시면 되겠습니다.

Chapter 5. Mysql

MySQL 은 리눅스 기반의 OS 에서 PostgreSQL 등과 함께 가장 많이 이용되는 데이터베이스입니다. 공개된 관계형 데이터베이스로서 최근 SUN 을 인수한 ORACLE 에 의해서 배포되고 있습니다. 일반적으로 Community Server 버전의 MySQL 을 서버에서 구동하여 이용되며, MySQL Enterprise 버전에서는 일반 유료 데이터베이스와 마찬가지로 엔터프라이즈 급의 기능을 제공하고 유상 기술 지원 서비스도 제공되고 있으므로 대기업 등에서도 점진적으로 많은 관심을 보이고 있습니다.

1. MySQL 의 구동

두가지 아래와 같이 두가지 방법으로 MySQL 데몬을 구동하실 수 있습니다.

```
root@local:/
[root@localhost /]# /usr/local/mysql/share/mysql/mysql.server start
Starting MySQL                                             [  OK  ]

[root@localhost /]# /usr/local/mysql/bin/mysqld_safe &
[3] 27260
100515 09:32:05 mysqld_safe Logging to '/usr/local/mysql/var/localhost.err'.
100515 09:32:05 mysqld_safe Starting mysqld daemon with databases from
/usr/local/src/mysql/var
```

2. MySQL 의 종료

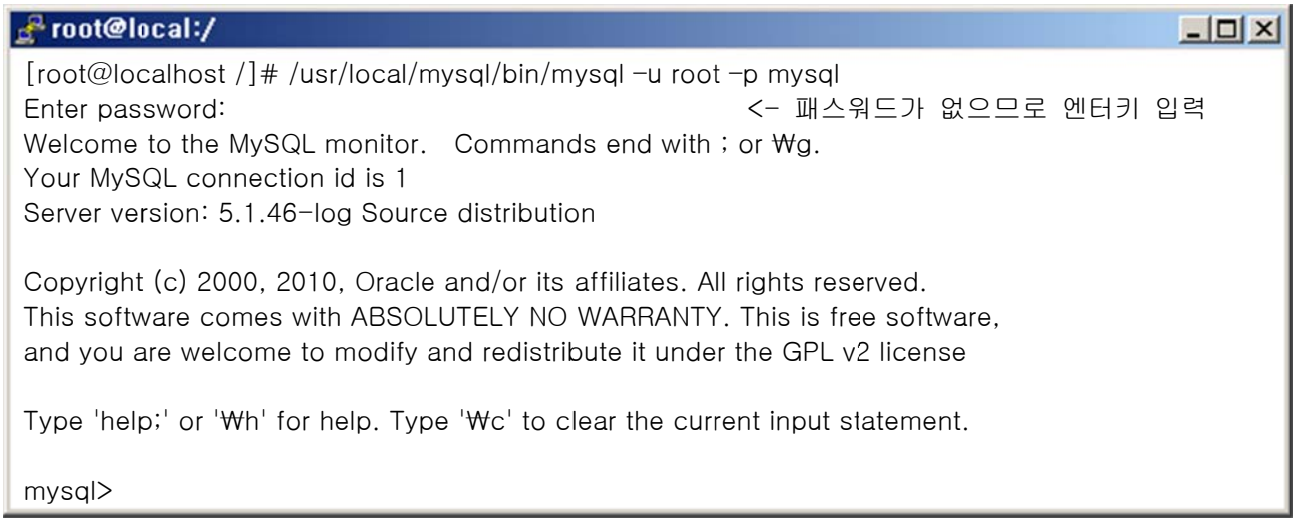
3가지 방법으로 MySQL 을 종료할 수 있습니다.

```
root@local:/
[root@localhost /]# /usr/local/mysql/share/mysql/mysql.server stop
Shutting down MySQL.                                       [  OK  ]

[root@localhost /]# /usr/local/mysql/bin/mysqladmin -u root -p shutdown
Enter password:

[root@localhost /]# pkill mysqld
[root@localhost /]# ps aux | grep mysql
```

3. MySQL 에 접속

형식 : /usr/local/mysql/bin/mysql -u 사용자명 -p 데이터베이스명

위의 형식과 같이 접속이 가능합니다.



4. MySQL 관리자 패스워드 설정
기본적으로 MySQL 을 설치하게 되면, MySQL 데몬의 관리자(root) 패스워드는 비어 있는 상태입니다. 초기 설치 후에 반드시 관리자 패스워드를 지정하시기 바랍니다.

형식 : /usr/local/mysql/bin/mysqladmin -u root -p password 패스워드

위와 같이도 설정이 가능하나 가능하면 mysql> 콘솔에서 적용하시기를 추천해 드립니다.

아래 예에서는 MySQL 관리자인 root 사용자의 패스워드를 ghtmxmdnpdl!@)(로 설정하는 예입니다.

root@local:/

```
[root@localhost~]# /usr/local/mysql/bin/mysql -u root -p mysql
Enter password:                                          <- 패스워드가 없으므로 엔터키 입력
Welcome to the MySQL monitor.  Commands end with ; or ₩g.
Your MySQL connection id is 1
Server version: 5.1.46-log Source distribution

Copyright (c) 2000, 2010, Oracle and/or its affiliates. All rights reserved.
This software comes with ABSOLUTELY NO WARRANTY. This is free software,
and you are welcome to modify and redistribute it under the GPL v2 license

Type 'help;' or '₩h' for help. Type '₩c' to clear the current input statement.

mysql> update user set password = password('ghtmxmdnpdl!@)(') where user = 'root';
Query OK, 3 rows affected (0.03 sec)
Rows matched: 3  Changed: 3  Warnings: 0

mysql> flush privileges;
Query OK, 0 rows affected (0.00 sec)

mysql> exit;                                             <- 접속 확인을 위해서 종료

[root@localhost~]# /usr/local/mysql/bin/mysql -u root -p mysql
Enter password:                                          <- ghtmxmdnpdl!@)( 입력
Welcome to the MySQL monitor.  Commands end with ; or ₩g.
Your MySQL connection id is 3
Server version: 5.1.46-log Source distribution

Copyright (c) 2000, 2010, Oracle and/or its affiliates. All rights reserved.
This software comes with ABSOLUTELY NO WARRANTY. This is free software,
and you are welcome to modify and redistribute it under the GPL v2 license

Type 'help;' or '₩h' for help. Type '₩c' to clear the current input statement.

mysql>                                                   <- 변경한 패스워드로 로그인됨
```

5. MySQL 사용자 추가 및 데이터베이스 생성

MySQL 에서는 사용자 및 데이터베이스 이름, 권한 설정 등의 전반적인 데몬의 구동 정보는 'mysql' 데이터베이스에 기록됩니다. 따라서 MySQL 사용자 추가, 삭제, 데이터베이스의 생성, 삭제 등의 작업 시에는 'mysql' 데이터베이스에 접속하여 처리합니다.

'mysql' 데이터베이스에 연결하는 방법은 앞서 3차례 연습한 것과 같습니다.

root@local:/

```
[root@localhost~] /usr/local/mysql/bin/mysql -u root -p mysql
```

위와 같이 'mysql' 데이터 베이스에 로그인한 후 아래와 같이 수행합니다.

5.1 MySQL 데이터베이스 생성

```
mysql> create database 데이터베이스명;
```

5.2 MySQL 사용자 추가

```
mysql> insert into user (host,user,password) values('localhost','사용자명',password('패스워드'));
mysql> flush privileges;

일반적으로 사용자와 사용할 데이터베이스는 지정되어 있으므로, 아래와 같이도 가능
mysql> grant all privileges on 데이터베이스.* to 사용자명@'localhost' identified by '패스워드';
mysql> flush privileges;
```

```
mysql> insert into user (host,user,password) values('localhost','사용자명',password('패스워드'));
mysql> flush privileges;

일반적으로 사용자와 사용할 데이터베이스는 지정되어 있으므로, 아래와 같이도 가능
mysql> grant all privileges on 데이터베이스.* to 사용자명@'localhost' identified by '패스워드';
mysql> flush privileges;
```

반대로 hostway_db 데이터베이스, hostway MySQL 사용자를 삭제하는 방법은 아래와 같습니다.

```
mysql> delete database hostway_db;

mysql> delete from user where user = 'hostway';
```

5.3 데이터베이스 접근 권한 설정

insert 문으로 'db' 테이블에 접근 가능한 호스트명(또는 IP, %), 데이터베이스, 데이터베이스에 접근 가능한 user, user 에게 부여할 접근 권한 설정을 합니다. 아래에서 'y' 부분은 권한을 부여해 준 것이며, 권한을 부여하지 않을 경우는 'n' 으로 설정 합니다. 어떠한 권한을 부여할지는, "desc db;" 쿼리문으로 db 테이블의 필드를 확인하면 됩니다. 여기서 'y'의 개수는 db 테이블의 필드 수만큼 입력해야 하며, 데이터베이스 버전에 따라 필드 수가 다르기 때문에 주의해야 합니다.

```
mysql> insert into db values('localhost','디비명','추가계정','y','y','y','y','y','y','y','y','y','y','y','y','y');
```

각 테이블에 추가 완료한 후 "flush privileges;" 쿼리문으로 추가한 사항을 적용시켜 줍니다.

하지만 일반적으로 아래와 같이 간단하게 데이터베이스에 접근 권한을 설정하셔도 무방합니다.

```
mysql> grant all privileges on 데이터베이스.* to 사용자명@'localhost' identified by '패스워드';
mysql> flush privileges;
```

이제 앞서 설명드린 내용을 참고로 새로운 데이터베이스와 사용자를 생성하고 권한을 부여해보도록 하겠습니다.

예)

사용자명	hostway
패스워드	ghtmxmdnpdl
데이터베이스명	hostway_db
접속 허용 호스트	localhost

'mysql' 데이터베이스에 mysql 관리자인 root 사용자로 로그인 후 아래와 같이 데이터베이스 및 계정 생성, 패스워드 지정 등을 절차를 수행합니다.

```
root@local:/
[root@localhost~]# /usr/local/mysql/bin/mysql -u root -p mysql
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or ₩g.
Your MySQL connection id is 1
Server version: 5.1.46-log Source distribution

Copyright (c) 2000, 2010, Oracle and/or its affiliates. All rights reserved.
This software comes with ABSOLUTELY NO WARRANTY. This is free software,
and you are welcome to modify and redistribute it under the GPL v2 license

Type 'help;' or '₩h' for help. Type '₩c' to clear the current input statement.

mysql> create database hostway_db;
Query OK, 1 row affected (0.00 sec)

mysql> grant all privileges on hostway_db.* to hostway@'localhost' identified by 'ghtmxmdnpdl';
Query OK, 0 rows affected (0.00 sec)

mysql> flush privileges;
Query OK, 0 rows affected (0.00 sec)

mysql>
```

5.4 원격 연결 허용 설정

앞서 추가해준 hostway 사용자가 외부 서버에서도 hostway_db 에 연결이 가능하도록 요청이 있을 경우가 있을 것 입니다.

원격지를 표현하는데에는 FQDN(호스트명), IP 로 특정 지을 수 있으며, 전체 네트워크는 % 를 이용하면 됩니다.

아래는 211.115.223.215 아이피에서 연결을 허용하는 방법입니다.

```
mysql> grant all privileges on hostway_db.* to hostway@'211.115.223.215' identified by 'ghtmxmdnpdl';
mysql> flush privileges;
```

아래 예는 전체 네트워크에서 MySQL 서버로 hostway 사용자가 hostway_db db 에 연결을 허용하는 방법입니다.

```
mysql> grant all privileges on hostway_db.* to hostway@'%' identified by 'ghtmxmdnpdl';
mysql> flush privileges;
```

6. 자주 사용하는 쿼리문
데이터베이스 목록 및 테이블 목록을 보기 위해서는 아래와 같이 show databases 와 show tables 로 확인이 가능합니다.

```
mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| hostway            |
| mysql              |
| test               |
+--------------------+
4 rows in set (0.02 sec)

mysql> show tables;
+---------------------------+
| Tables_in_mysql           |
+---------------------------+
| columns_priv              |
| db                        |
| event                     |
| func                      |
| general_log               |
| help_category             |
| help_keyword              |
| help_relation             |
| help_topic                |
| host                      |
| ndb_binlog_index          |
| proc                      |
| procs_priv                |
| servers                   |
| slow_log                  |
| tables_priv               |
| time_zone                 |
| time_zone_leap_second     |
| time_zone_name            |
| time_zone_transition      |
| time_zone_transition_type |
| user                      |
+---------------------------+
```

```
23 rows in set (0.00 sec)
```

사용할 데이터베이스를 선택시에는 아래와 같이 할 수 있습니다.

```
mysql> use 데이터베이스명;
```

desc 를 사용하면 테이블의 구조에 대해서도 확인이 가능합니다.

```
mysql> desc 테이블명;
```

MySQL 데몬의 구동중인 환경을 보기 위해서는 아래와 같이 합니다.

```
mysql> show variables;
```

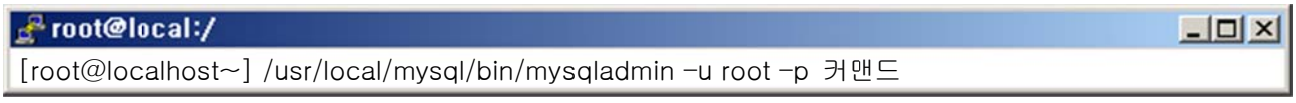
실행중인 SQL 쿼리문의 확인이 필요시에는 아래와 같이 합니다.

```
mysql> show processlist;
```

실행중인 SQL 쿼리를 종료를 위해서는 아래와 같이 합니다.

```
mysql> show processlist;                                    <- 먼저 구동중인 쿼리들을 확인합니다.
+----+------+-----------+-------+---------+------+-------+------------------+
| Id | User | Host      | db    | Command | Time | State | Info             |
+----+------+-----------+-------+---------+------+-------+------------------+
| 3 | root | localhost  | mysql | Query   |    0 | NULL | show processlist |
| 9 | hostway    | cloud.hostway.co.kr:34876 | hostway_db    | Query       | 23   | Sorting result    |
select * from cloud_users o |
+----+------+-----------+-------+---------+------+-------+------------------+
1 row in set (0.00 sec)

Id 9 번 쿼리문을 강제로 종료하려면 아래와 같이 합니다.
mysql> kill 9;
Query OK, 0 rows affected (0.00 sec)

mysql>
```

7. mysqladmin 사용법
mysqladmin 은 아래와 같은 다양한 기능을 제공하고 있습니다.
기본적인 사용법은 아래와 같습니다.



사용 가능한 커맨드는 아래 표를 참고해주시기 바랍니다.

[표 5-1] mysqladmin 명령어 설명

명령어	설명
create 데이터베이스명	데이터베이스를 생성
debug	MySQL 의 debug 메세지를 생성
drop 데이터베이스명	데이터베이스를 삭제
extended-status	MySQL 의 환경 값과 구동중인 상태값을 출력
flush-hosts	연결된 호스트를 모두 끊고 초기화
flush-logs	hostname.err 로그를 비움
flush-status	MySQL 의 상태값을 초기화
flush-tables	테이블의 내용을 비움
flush-threads	MySQL thread(process) 를 모두 종료
flush-privileges	grant table 을 재인식시킴
kill id,id,...	MySQL thread(process) 를 id 별로 종료
ping	MySQL 데몬의 구동 유무 상태를 출력
processlist	MySQL thread(process) 를 모두 출력
reload	MySQL 데몬을 재구동
shutdown	MySQL 데몬을 종료
status	MySQL 구동 상태 정보를 간략히 출력
start-slave	MySQL replication slave 서버를 시작
stop-slave	MySQL replication slave 서버를 종료
variables	MySQL 데몬의 구동 환경 값을 출력
version	MySQL 버전 등을 출력

8. MySQL 데이터베이스 백업
백업은 아래와 같이 간단히 2가지 방법을 이용할 수 있습니다.

첫번째는 /usr/local/mysql/var datadir 을 통째로 압축하여 보관하는 방법입니다.

또는 testdb 데이터베이스 한개만 백업을 원하면 아래와 같은 방법으로도 가능합니다.

두번째는 mysqldump 명령어를 이용하시는 방법이 있습니다.

전체 데이터베이스를 모두 한 개의 파일로 백업을 위해서는 아래와 같이 합니다.

위와 같이 /usr/local/mysql/var 디렉토리를 전체 압축하여 백업하는 방법과 mysqldump 를 활용해서 sql row 단위로 백업하는 방법이 있으며, 두가지 방법을 동시에 적용하여 데이터베이스의 다양한 백업본을 보유하시기를 추천해드립니다. 또한 cron 을 활용하여 주기적인 백업을 자동화 시켜두시면 데이터베이스 유실 등의 장애 시에 빠른 복구가 가능합니다.

9. MySQL 데이터베이스 복구

앞서 설명해드린 백업본을 이용하여 복구하는 방법에 대해서 알아 보겠습니다.

/usr/local/mysql/var 디렉토리를 전체 압축하여 보관 중이라면, /usr/local/mysql/var 디렉토리를 백업 받아둔 것을 압축 해제하여 덮어씌우기를 합니다.

mysqldump 를 활용하여 sql row 단위로 testdb.sql 로 백업한 백업본을 이용하여 복구하는 방법은 아래와 같습니다.

MySQL 버전을 낮추거나 높여서 데이터베이스를 복구하는 방법은 MySQL-4.0.X 이하 버전에서는 한국어 언어셋이 euckr 입니다. 이상의 버전에서는 euckr 로 변경되어, 두 방법으로 설명하겠습니다.

우선 MySQL-4.0.X 이하 버전에서 백업 후 MySQL-4.1.X 이상의 버전으로 복구하는 방법은 아래와 같습니다. 이때에는 /usr/local/mysql/var 디렉토리를 압축해서 덮어씌우는 방법으로는 할 수 없습니다. /usr/local/mysql/bin/mysql 명령어를 통해서 아래와 같이 하시면 됩니다.

반대로 MySQL-4.1.X 이상에서 MySQL-4.0.X 이하로 다운그레이드시에도 한국어 언어셋의 차이로 인해서 에러가 발생됩니다. 다운그레이드시에는 /usr/local/mysql/bin/mysqldump 를 이용하여 백업시에 -default-character-set=euckr 옵션을 추가하여 백업을 받은 후 백업받은 sql 파일을 vi 등의 편집기를 활용하여 추가로 수정 후 복구가 가능합니다.

“DEFAULT CHARSET=euckr" 삭제 " collate euckr_bin" 삭제 “euckr” 을 “euc_kr” 로 변경

10. myisamchk 사용하기

MySQL 테이블의 손상으로 인해서 복구가 필요할 경우도 있습니다.
이때에도 마찬가지로 백업본을 활용하여 복구도 가능하나, 더 간단하게는 아래와 같이 /usr/local/mysql/bin/myisamchk 명령어를 활용하는 방법이 있습니다. 사용 방법은 아래와 같습니다.

```
root@local:/
[root@localhost~]# /usr/local/mysql/bin/myisamchk -r -q /usr/local/mysql/var/testdb/testtb.MYI
- check record delete-chain
- recovering (with sort) MyISAM-table '/usr/local/mysql/var/testdb/testtb.MYI'
Data records: 6
- Fixing index 1
```

11. MySQL 관리자(root) 계정의 패스워드 분실
MySQL 관리자(root) 계정의 패스워드를 분실 시에는 MySQL 데몬을 skip-grant 옵션을 활용하여 구동 후 패스워드 수정이 가능합니다.

```
root@local:/
[root@localhost~]# /usr/local/mysql/share/mysql/mysql.server stop   <- MySQL 을 종료
Shutting down MySQL.                                        [  OK  ]

[root@localhost~]# /usr/local/mysql/bin/mysqld_safe --skip-grant & <- MySQL 을 시작
100515 14:19:54 mysqld_safe Logging to '/usr/local/mysql/var/localhost.err'.
100515 14:19:54 mysqld_safe Starting mysqld daemon with databases from /usr/local/ mysql/var

[root@localhost~]# /usr/local/mysql/bin/mysql -u root mysql            <- 로그인
Welcome to the MySQL monitor.  Commands end with ; or ₩g.
Your MySQL connection id is 1
Server version: 5.1.46-log Source distribution

Copyright (c) 2000, 2010, Oracle and/or its affiliates. All rights reserved.
This software comes with ABSOLUTELY NO WARRANTY. This is free software,
and you are welcome to modify and redistribute it under the GPL v2 license

Type 'help;' or '₩h' for help. Type '₩c' to clear the current input statement.

mysql> update user set password = password('새로운패스워드') where user = 'root';
mysql> flush privileges;
mysql> exit;

[root@localhost~]# /usr/local/mysql/share/mysql/mysql.server stop   <- MySQL 을 종료
Shutting down MySQL.100515 14:25:46 mysqld_safe mysqld from pid file /usr/local/
mysql/var/localhost.pid                                                    ended
[  OK  ]
[3]+  Done                    /usr/local/mysql/bin/mysqld_safe --skip-grant


[root@localhost~]# /usr/local/mysql/share/mysql/mysql.server start   <- MySQL 을 시작
Starting MySQL.                                             [  OK  ]
```

Chapter 6. sendmail

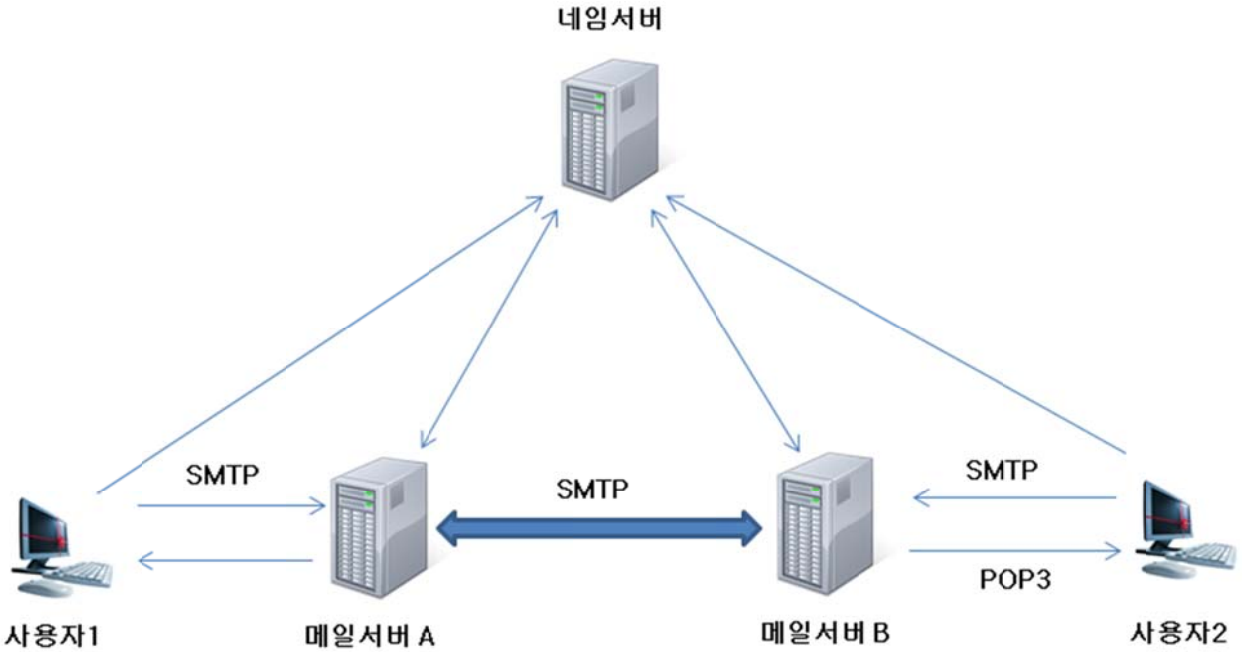
1. 메일서버

인터넷을 이용하는 거의 대부분 사용자들은 서너 개 이상의 이메일을 가지고 있을 것입니다. google이나 naver등의 주요 포털 사이트에서 무료로 제공을 해주는 이메일을 사용하고 계신 분들도 많이 있겠지만, 리눅스 서버에서 보유중인 도메인으로 이메일을 송, 수신할 있도록 메일 서버를 설정해 보도록 하겠습니다.
우선, 메일 서버에서 사용되는 프로토콜에 대해서 간략히 알아보겠습니다. 리눅스 기반의 메일 서버에서 사용되는 프로토콜은 크게 세 가지로 나눌 수 있습니다.

[표 6-1] 메일 서버 비교

메일 서버	설명
SMTP(MTA)	Simple Mail Transfer, 메일 사용자의 MUA와 메일서버, 또는 메일서버간 실제 메일을 송수신시에 이용되는 프로토콜, TCP/25 포트번호를 이용
POP3	클라이언트의 MUA에서 메일서버에 수신된 메일을 가져올 때 이용되는 프로토콜, TCP/110 번 포트번호를 이용
IMAP	클라이언트의 MUA 에서 메일서버에 수신된 메일을 가져올 때 이용되는 프로토콜로 POP3 와 달리 원본 메일을 삭제하지 않음 TCP/143 번 포트번호를 이용

작동 방식은 아래 그림을 보면서 설명을 드리겠습니다.



[그림 6-1] 메일 서버를 위한 네임 서버 작동 방식

사용자1 이 사용자2 에게 메일을 작성하여 전송하였다면, 사용자1 PC 에 설정된 SMTP 인 메일서버A 로 SMTP 프로토콜을 이용하여 메일을 전송됩니다.
메일서버 A 는 사용자1로부터 수신받은 메일을 사용자2의 받는 이메일 주소의 MX 인 메일서버B 에게 메일을 전달합니다.
마지막으로 사용자2는 POP3 프로토콜을 이용하여 메일서버B 에 수신된 사용자1 로부터 전송된 메일을 내려받기를 하게 됩니다. 또한 사용자2가 사용자1에게 메일을 전송 시에는 반대로 방향으로 이해를 하시면 되겠습니다.

2. 메일 서버의 종류

리눅스 기반의 서버에서 가장 많이 이용되는 메일 서버들은 sendmail, postfix, qmail 이 있습니다. 이번에는 sendmail 과 qmail 을 활용하여 메일 서버를 구성하는 방법에 대해서 알아보도록 하겠습니다.

2.1 sendmail 과 qmail 의 비교

[표 6-2] Sendmail 과 Qmail 비교

비교 항목	Sendmail	Qmail
송신	한 개의 프로세서가 담당	수신 담당 프로세서가 담당(qmail-send)
수신		송신 담당 프로세서가 담당 (qmail-smtp)
로그	/var/log/maillog (로그 정보가 많음)	/var/log/maillog (로그 정보가 적음)
Mailbox	/var/spool/mail/계정	일반적으로 사용자의 홈 디렉토리 아래에 생성됨(new / cur / tmp)
설치 경오	/etc/mail	/var/qmail
프로세스 수	1	설치에 따라서 12~14개
속도	Qmail에 비해서 속도가 느림	개별 프로세스가 각각의 job을 수행하므로 상대적으로 속도가 빠름
사용 규모	소 / 중 규모에서 사용	대형 규모에서 사용
특징	- 로그 분석이 쉬움 - 많은 사용자들이 사용하고 있어 자료나 문제 해결 시 편리함 - rpm으로도 제공되어 설치가 쉬움 - 프로세서 관리가 쉬움 - 서버안의 실제 계정에 대해서만 메일이 관리	- 여러 프로세서가 분업화되어 동시적으로 작동함으로 속도가 빠름 - 각각의 메일이 파일 별로 저장되어 메일 관리가 쉬움 - 가상유저 방식으로 가상호스팅에 적합

3. sendmail로 메일 서버 설정하기

보유중인 도메인을 이용하여 메일 서버를 설정하기 위해서는 해당 도메인에 MX 값을 설정해야 합니다. DNS 에서 MX 는 메일서버를 의미하며, IP 나 FQDN 으로 설정이 가능합니다.

3.1 MX 레코드 설정
설명에 이용될 hostway.co.kr 도메인의 존파일 내용은 아래와 같습니다

```
$TTL    3600
…(중략)
IN      MX 10 mail
IN      TXT     "v=spf1 mx ip4:211.115.223.215 -all"
mail    IN      A       211.115.223.215
…(중략)
```

위와 같이 MX 레코드는 메일 서버의 아이피로 리졸빙이 되도록 설정을 해야 합니다.

3.2 메일 서버 설정
이제 본격적으로 sendmail 을 이용해서 메일 서버를 구성하겠습니다.

우선, 메일 서버 구성에 필요한 서버에 설치된 sendmail 과 dovecot, cyrus-sasl 를 설치합니다.

```
root@local:/
[root@localhost~]# yum -y install sendmail* dovecot cyrus-sasl
Loaded plugins: fastestmirror
Loading mirror speeds from cached hostfile
 * addons: mirror.hostway.co.kr
 * base: mirror.hostway.co.kr
 * extras: mirror.hostway.co.kr
 * updates: mirror.hostway.co.kr
Setting up Install Process
Resolving Dependencies
 Resolution

Dependencies Resolved
……
……
……
Installed:
  cyrus-sasl.i386 0:2.1.22-5.el5_4.3  dovecot.i386 0:1.0.7-7.el5 sendmail.i386 0:8.13.8-8.el5
sendmail-cf.i386  0:8.13.8-8.el5  sendmail-devel.i386  0:8.13.8-8.el5  sendmail-doc.i386
0:8.13.8-8.el5

Complete!
```

이제 sendmail 부터 설정을 시작해보겠습니다.

우선, /etc/mail/sendmail.mc 파일에서는 52, 53 번째 라인을 아래와 같이 dnl #(주석) 을 삭제합니다.

```
52 TRUST_AUTH_MECH(`EXTERNAL DIGEST-MD5 CRAM-MD5 LOGIN PLAIN')dnl
53 define(`confAUTH_MECHANISMS', `EXTERNAL GSSAPI DIGEST-MD5 CRAM-MD5 LOGIN PLAIN')dnl
```

두번째로 /etc/mail/sendmail.mc 파일의 116 라인을 아래와 같이 수정합니다.

```
116 DAEMON_OPTIONS(`Port=smtp,Addr=0.0.0.0, Name=MTA')dnl
```

위와 같이 3 라인을 수정하면 아래와 같이 수정이 됩니다.

```
divert(-1)dnl
dnl #
dnl # This is the sendmail macro config file for m4. If you make changes to
dnl # /etc/mail/sendmail.mc, you will need to regenerate the
dnl # /etc/mail/sendmail.cf file by confirming that the sendmail-cf package is
dnl # installed and then performing a
dnl #
dnl #       make -C /etc/mail
dnl #
include(`/usr/share/sendmail-cf/m4/cf.m4')dnl
VERSIONID(`setup for linux')dnl
OSTYPE(`linux')dnl
dnl #
dnl # Do not advertize sendmail version.
dnl #
dnl define(`confSMTP_LOGIN_MSG', `$j Sendmail; $b')dnl
dnl #
dnl # default logging level is 9, you might want to set it higher to
dnl # debug the configuration
dnl #
dnl define(`confLOG_LEVEL', `9')dnl
dnl #
dnl # Uncomment and edit the following line if your outgoing mail needs to
dnl # be sent out through an external mail server:
dnl #
dnl define(`SMART_HOST', `smtp.your.provider')dnl
dnl #
define(`confDEF_USER_ID', ``8:12'')dnl
dnl define(`confAUTO_REBUILD')dnl
define(`confTO_CONNECT', `1m')dnl
define(`confTRY_NULL_MX_LIST', `True')dnl
define(`confDONT_PROBE_INTERFACES', `True')dnl
define(`PROCMAIL_MAILER_PATH', `/usr/bin/procmail')dnl
define(`ALIAS_FILE', `/etc/aliases')dnl
define(`STATUS_FILE', `/var/log/mail/statistics')dnl
define(`UUCP_MAILER_MAX', `2000000')dnl
define(`confUSERDB_SPEC', `/etc/mail/userdb.db')dnl
define(`confPRIVACY_FLAGS', `authwarnings,novrfy,noexpn,restrictqrun')dnl
define(`confAUTH_OPTIONS', `A')dnl
dnl #
dnl # The following allows relaying if the user authenticates, and disallows
dnl # plaintext authentication (PLAIN/LOGIN) on non-TLS links
dnl #
dnl define(`confAUTH_OPTIONS', `A p')dnl
dnl #
```

```
dnl # PLAIN is the preferred plaintext authentication method and used by
dnl # Mozilla Mail and Evolution, though Outlook Express and other MUAs do
dnl # use LOGIN. Other mechanisms should be used if the connection is not
dnl # guaranteed secure.
dnl # Please remember that saslauthd needs to be running for AUTH.
dnl #
TRUST_AUTH_MECH(`EXTERNAL DIGEST-MD5 CRAM-MD5 LOGIN PLAIN')dnl
define(`confAUTH_MECHANISMS', `EXTERNAL GSSAPI DIGEST-MD5 CRAM-MD5 LOGIN PLAIN')dnl
dnl # 위의 2 라인에서 dnl #(주석) 을 제거합니다.
dnl #
dnl # Rudimentary information on creating certificates for sendmail TLS:
dnl #       cd /etc/pki/tls/certs; make sendmail.pem
dnl # Complete usage:
dnl #       make -C /etc/pki/tls/certs usage
dnl #
dnl define(`confCACERT_PATH', `/etc/pki/tls/certs')dnl
dnl define(`confCACERT', `/etc/pki/tls/certs/ca-bundle.crt')dnl
dnl define(`confSERVER_CERT', `/etc/pki/tls/certs/sendmail.pem')dnl
dnl define(`confSERVER_KEY', `/etc/pki/tls/certs/sendmail.pem')dnl
dnl #
dnl # This allows sendmail to use a keyfile that is shared with OpenLDAP's
dnl # slapd, which requires the file to be readble by group ldap
dnl #
dnl define(`confDONT_BLAME_SENDMAIL', `groupreadablekeyfile')dnl
dnl #
dnl define(`confTO_QUEUEWARN', `4h')dnl
dnl define(`confTO_QUEUERETURN', `5d')dnl
dnl define(`confQUEUE_LA', `12')dnl
dnl define(`confREFUSE_LA', `18')dnl
define(`confTO_IDENT', `0')dnl
dnl FEATURE(delay_checks)dnl
FEATURE(`no_default_msa', `dnl')dnl
FEATURE(`smrsh', `/usr/sbin/smrsh')dnl
FEATURE(`mailertable', `hash -o /etc/mail/mailertable.db')dnl
FEATURE(`virtusertable', `hash -o /etc/mail/virtusertable.db')dnl
FEATURE(redirect)dnl
FEATURE(always_add_domain)dnl
FEATURE(use_cw_file)dnl
FEATURE(use_ct_file)dnl
dnl #
dnl # The following limits the number of processes sendmail can fork to accept
dnl # incoming messages or process its message queues to 20.) sendmail refuses
dnl # to accept connections once it has reached its quota of child processes.
dnl #
dnl define(`confMAX_DAEMON_CHILDREN', `20')dnl
dnl #
dnl # Limits the number of new connections per second. This caps the overhead
dnl # incurred due to forking new sendmail processes. May be useful against
dnl # DoS attacks or barrages of spam. (As mentioned below, a per-IP address
dnl # limit would be useful but is not available as an option at this writing.)
dnl #
dnl define(`confCONNECTION_RATE_THROTTLE', `3')dnl
dnl #
```

```
dnl # The -t option will retry delivery if e.g. the user runs over his quota.
dnl #
FEATURE(local_procmail, `', `procmail -t -Y -a $h -d $u')dnl
FEATURE(`access_db', `hash -T<TMPF> -o /etc/mail/access.db')dnl
FEATURE(`blacklist_recipients')dnl
EXPOSED_USER(`root')dnl
dnl #
dnl # For using Cyrus-IMAPd as POP3/IMAP server through LMTP delivery uncomment
dnl # the following 2 definitions and activate below in the MAILER section the
dnl # cyrusv2 mailer.
dnl #
dnl define(`confLOCAL_MAILER', `cyrusv2')dnl
dnl define(`CYRUSV2_MAILER_ARGS', `FILE /var/lib/imap/socket/lmtp')dnl
dnl #
dnl # The following causes sendmail to only listen on the IPv4 loopback address
dnl # 127.0.0.1 and not on any other network devices. Remove the loopback
dnl # address restriction to accept email from the internet or intranet.
dnl #
DAEMON_OPTIONS(`Port=smtp,Addr=0.0.0.0, Name=MTA')dnl
dnl # 위의 라인에서 Port=smtp,Addr=127.0.0.1, Name=MTA 를
dnl # Port=smtp,Addr=0.0.0.0, Name=MTA 으로 수정을 합니다.
dnl #
dnl # The following causes sendmail to additionally listen to port 587 for
dnl # mail from MUAs that authenticate. Roaming users who can't reach their
dnl # preferred sendmail daemon due to port 25 being blocked or redirected find
dnl # this useful.
dnl #
dnl DAEMON_OPTIONS(`Port=submission, Name=MSA, M=Ea')dnl
dnl #
dnl # The following causes sendmail to additionally listen to port 465, but
dnl # starting immediately in TLS mode upon connecting. Port 25 or 587 followed
dnl # by STARTTLS is preferred, but roaming clients using Outlook Express can't
dnl # do STARTTLS on ports other than 25. Mozilla Mail can ONLY use STARTTLS
dnl # and doesn't support the deprecated smtps; Evolution <1.1.1 uses smtps
dnl # when SSL is enabled-- STARTTLS support is available in version 1.1.1.
dnl #
dnl # For this to work your OpenSSL certificates must be configured.
dnl #
dnl DAEMON_OPTIONS(`Port=smtps, Name=TLSMTA, M=s')dnl
dnl #
dnl # The following causes sendmail to additionally listen on the IPv6 loopback
dnl # device. Remove the loopback address restriction listen to the network.
dnl #
dnl DAEMON_OPTIONS(`port=smtp,Addr=::1, Name=MTA-v6, Family=inet6')dnl
dnl #
dnl # enable both ipv6 and ipv4 in sendmail:
dnl #
dnl DAEMON_OPTIONS(`Name=MTA-v4, Family=inet, Name=MTA-v6, Family=inet6')
dnl #
dnl # We strongly recommend not accepting unresolvable domains if you want to
dnl # protect yourself from spam. However, the laptop and users on computers
dnl # that do not have 24x7 DNS do need this.
dnl #
```

```
FEATURE(`accept_unresolvable_domains')dnl
dnl #
dnl FEATURE(`relay_based_on_MX')dnl
dnl #
dnl # Also accept email sent to "localhost.localdomain" as local email.
dnl #
LOCAL_DOMAIN(`localhost.localdomain')dnl
dnl #
dnl # The following example makes mail from this host and any additional
dnl # specified domains appear to be sent from mydomain.com
dnl #
dnl MASQUERADE_AS(`mydomain.com')dnl
dnl #
dnl # masquerade not just the headers, but the envelope as well
dnl #
dnl FEATURE(masquerade_envelope)dnl
dnl #
dnl # masquerade not just @mydomainalias.com, but @*.mydomainalias.com as well
dnl #
dnl FEATURE(masquerade_entire_domain)dnl
dnl #
dnl MASQUERADE_DOMAIN(localhost)dnl
dnl MASQUERADE_DOMAIN(localhost.localdomain)dnl
dnl MASQUERADE_DOMAIN(mydomainalias.com)dnl
dnl MASQUERADE_DOMAIN(mydomain.lan)dnl
MAILER(smtp)dnl
MAILER(procmail)dnl
dnl MAILER(cyrusv2)dnl
```

생성한 sendmail.mc 파일을 sendmail.cf 파일로 변환하고 /etc/mail/local-host-names 에 메일에 이용될 도메인을 입력합니다.

이제, sendmail 데몬을 구동하게 되면 후 telnet localhost 25 와 같이 smtp 연결 테스트가 잘 된다면, sendmail 설정은 모두 끝난 것입니다.

```
root@local:/
[root@localhost /]# m4 /etc/mail/sendmail.mc > /etc/mail/sendmail.cf

[root@localhost /]# echo “hostway.co.kr” >> /etc/mail/local-host-names

[root@localhost /]# /etc/init.d/sendmail start
Starting sendmail:                                         [  OK  ]
Starting sm-client:                                        [  OK  ]

[root@localhost /]# telnet localhost 25
Trying 127.0.0.1...
Connected to localhost.localdomain (127.0.0.1).
Escape character is '^]'.
220 localhost ESMTP Sendmail 8.13.8/8.13.8; Sat, 15 May 2010 16:38:01 +0900
ehlo hostway.co.kr
250-hostway.co.kr Hello localhost.localdomain [127.0.0.1], pleased to meet you
250-ENHANCEDSTATUSCODES
250-PIPELINING
250-8BITMIME
250-SIZE
250-DSN
250-ETRN
250-STARTTLS
250-DELIVERBY
250 HELP
quit
```

POP3 와 IMAP 서비스를 설정하여 아웃룩 등의 메일 클라이언트(MUA) 에서 메일을 송수신 할 수 있도록 설정합니다.

/etc/dovecot.conf 파일에서 20번째 라인을 아래와 같이 주석(#)을 삭제하고 수정합니다.

```
protocols = imap pop3
```

이제 모두 완료 되었습니다.

수정하시면 아래와 같이 됩니다.

```
## Dovecot configuration file

# If you're in a hurry, see http://wiki.dovecot.org/QuickConfiguration

# "dovecot -n" command gives a clean output of the changed settings. Use it
# instead of copy&pasting this file when posting to the Dovecot mailing list.

# '#' character and everything after it is treated as comments. Extra spaces
```

```
# and tabs are ignored. If you want to use either of these explicitly, put the
# value inside quotes, eg.: key = "# char and trailing whitespace  "

# Default values are shown for each setting, it's not required to uncomment
# any of the lines.

# Base directory where to store runtime data.
#base_dir = /var/run/dovecot/

# Protocols we want to be serving: imap imaps pop3 pop3s
# If you only want to use dovecot-auth, you can set this to "none".
protocols = imap pop3

.....
.....
.....
```

3.3 sendmail 작동 테스트

위와 같이 수정을 하였다면, dovecot 데몬과 saslauthd 데몬을 구동하여 정상적으로 POP3 서버가 구동중인지 확인을 해보겠습니다.
테스트에 사용되는 hostway@hostway.co.kr 이메일 사용자의 계정명은 hostway 이고, 패스워드는 hostway!@#$ 라고 예를 들겠습니다.

```
root@local:/
[root@localhost /]# /etc/init.d/dovecot start
Starting Dovecot Imap:                                     [  OK  ]

[root@localhost /]# /etc/init.d/saslauthd start
Starting saslauthd:                                        [  OK  ]

[root@localhost /]# telnet localhost 110                   <- POP3 로그인 테스트
Trying 127.0.0.1...
Connected to localhost (127.0.0.1).
Escape character is '^]'.
+OK Dovecot ready.
user hostway                                  <- hostway 계정 입력
+OK
pass hostway!@#$                              <- hostway 계정의 패스워드 입력
+OK Logged in.
LIST                                          <- LIST 입력하여 메일 리스트가 출력되는지 확인
+OK 0 messages:
.

[root@localhost /]# telnet localhost 143                   <- IMAP 로그인 테스트
Trying 127.0.0.1...
Connected to localhost.localdomain (127.0.0.1).
Escape character is '^]'.
* OK Dovecot ready.
a01 login hostway hostway!@#$                 <- "a01 login 계정명 패스워드" 입력
a01 OK Logged in.                             <- IMAP 로그인이 정상적으로 이뤄짐
a02 logout
* BYE Logging out
a02 OK Logout completed.
Connection closed by foreign host.
```

위와 같이 몇 개의 파일에서 몇 라인을 수정하는 단순한 작업을 통해서 소유중인 도메인을 활용하여 자체 SMTP, POP3, IMAP 서버를 구성할 수 있습니다.

Chapter 7. Qmail

sendmail 로 구동중인 메일 서버를 qmail 로 변경을 해보도록 하겠습니다. qmail 은 소스 패키지를 컴파일하여 이용해야 설치합니다. 아래 내용를 천천히 따라하시면 qmail로 메일 서버를 쉽게 구성이 가능합니다..

패키지 컴파일시에 errno.h 에러가 발생하면, 각 패키지의 소스중에서 error.h 파일의 상단에 **#include "errno.h"** 를 추가하시기 바랍니다.

1. 필요한 패키지 다운로드

qmail 은 yum repository 에서 구할 수 없습니다. 각각의 패키지를 다음와 같이 다운로드 합니다.

```
root@local:/
[root@localhost /]# cd /usr/local/src/
[root@localhost /]# wget ftp://ftp.eu.uu.net/pub/unix/mail/qmail/qmail-1.03.tar.gz
[root@localhost /]# wget http://cr.yp.to/ucspi-tcp/ucspi-tcp-0.88.tar.gz
[root@localhost /]# wget http://cr.yp.to/daemontools/daemontoo s-0.76.tar.gz
[root@localhost /] wget http://downloads.sourceforge.net/project/vpopmail/vpopmail-devel/5.5.0/vpopmail-5.5.0.tar.bz2?use_mirror=cdnetworks-kr-1
[root@localhost /]# wget http://www.inter7.com/devel/autorespond-2.0.2.tar.gz
```

2. 패키지 컴파일

qmail 의 설치 경로 생성과 사용자, 그룹을 생성 후 qmail 패키지를 컴파일 합니다.

```
root@local:/
[root@localhost /]# mkdir /var/qmail
[root@localhost /]# groupadd nofiles
[root@localhost /]# groupadd qmail
[root@localhost /]# useradd -g nofiles -d /var/qmail/alias alias
[root@localhost /]# useradd -g nofiles -d /var/qmail qmaild
[root@localhost /]# useradd -g nofiles -d /var/qmail qmaill
[root@localhost /]# useradd -g nofiles -d /var/qmail qmailp
[root@localhost /]# useradd -g qmail -d /var/qmail qmailq
[root@localhost /]# useradd -g qmail -d /var/qmail qmailr
[root@localhost /]# useradd -g qmail -d /var/qmail qmails
[root@localhost /]# cp -af /usr/local/mysql/include/mysql /usr/include
[root@localhost /]# cp -af /usr/local/mysql/include/ /usr/include/mysql
[root@localhost /]# cp -f /usr/local/mysql/lib/mysql/* /usr/lib
[root@localhost /]# cp -af /usr/local/mysql/lib/* /usr/lib
[root@localhost /]# mkdir /usr/lib/mysql/
[root@localhost /]# ln -s /usr/lib/libmysqlclient.a /usr/lib/mysql/libmysqlclient.a
[root@localhost /]# tar zxvf qmail-1.03.tar.gz
[root@localhost /]# cd qmail-1.03
[root@localhost /]# make
[root@localhost /]# make setup check
[root@localhost /]# ./config-fast hostway.co.kr
```

qmail 컴파일이 모두 완료되었습니다. 이제 ucspi-tcp 를 설치하도록 하겠습니다.

```
root@local:/
[root@localhost /]# cd /usr/local/src/
[root@localhost /]# tar xzvf ucspi-tcp-0.88.tar.gz
[root@localhost /]# cd ucspi-tcp-0.88
[root@localhost /]# make
[root@localhost /]# make setup check
```

daemontools 를 설치합니다.

```
root@local:/
[root@localhost /]# cd /usr/local/src/
[root@localhost /]# mkdir -p /package
[root@localhost /]# chmod 1755 /package
[root@localhost /]# tar zxvf daemontools-0.76.tar.gz
[root@localhost /]# mv admin/ /package/
[root@localhost /]# cd /package/admin/daemontools-0.76/
[root@localhost /]# ./package/install
```

daemontools 까지 정상적으로 설치가 완료되었다면, 마지막으로 autorespond 를 설치합니다.

```
root@local:/
[root@localhost /]# cd /usr/local/src/
[root@localhost /]# tar zxvf autorespond-2.0.2.tar.gz
[root@localhost /]# cd autorespond-2.0.2
[root@localhost /]# make
[root@localhost /]# cp autorespond /usr/local/bin/
```

3. qmail 설정

qmail 설치시 필요한 패키지는 모두 컴파일이 완료되었으며, 이제 설정을 해보도록 하겠습니다.

qmail 구동시에 필요한 파일을 생성합니다.

root@local:/

```
[root@localhost /]# cat /var/qmail/rc
#!/bin/sh
exec env - PATH="/var/qmail/bin:$PATH" ₩
qmail-start ./Maildir/

[root@localhost /]# chmod a+x /var/qmail/rc
[root@localhost /]# mkdir -p /var/qmail/supervise/qmail-send/log
[root@localhost /]# mkdir -p /var/qmail/supervise/qmail-smtpd/log
[root@localhost /]# chmod +t /var/qmail/supervise/qmail-send
[root@localhost /]# chmod +t /var/qmail/supervise/qmail-smtpd

[root@localhost /] cat /var/qmail/supervise/qmail-send/run
#!/bin/sh
exec /var/qmail/rc

[root@localhost /] cat /var/qmail/supervise/qmail-send/log/run
#!/bin/sh
exec /usr/local/bin/setuidgid qmaill ₩
     /usr/local/bin/multilog t /var/log/qmail

[root@localhost /] cat /var/qmail/supervise/qmail-smtpd/run
#!/bin/sh
Q_UID=`id -u vpopmail`
Q_GID=`id -g vpopmail`
exec /usr/local/bin/softlimit -m 9000000 ₩
          /usr/local/bin/tcpserver -vRHl 0 ₩
          -x /home/vpopmail/etc/tcp.smtp.cdb ₩
          -u $Q_UID -g $Q_GID 0 25 /var/qmail/bin/qmail-smtpd hostway.co.kr ₩
          /home/vpopmail/bin/vchkpw /bin/true 2>&1

[root@localhost /] cat /var/qmail/supervise/qmail-smtpd/log/run
#!/bin/sh
exec /usr/local/bin/setuidgid qmaill ₩
     /usr/local/bin/multilog t /var/log/qmail/smtpd

[root@localhost /]# chmod 755 /var/qmail/supervise/qmail-send/run
[root@localhost /]# chmod 755 /var/qmail/supervise/qmail-send/log/run
[root@localhost /]# chmod 755 /var/qmail/supervise/qmail-smtpd/run
[root@localhost /]# chmod 755 /var/qmail/supervise/qmail-smtpd/log/run
[root@localhost /]# mkdir -p /var/log/qmail/smtpd
[root@localhost /]# chown qmaill /var/log/qmail /var/log/qmail/smtpd
[root@localhost /]# echo mailadmin@hostway.co.kr > /var/qmail/alias/.qmail-root
[root@localhost /]# echo mailadmin@hostway.co.kr > /var/qmail/alias/.qmail-postmaster
[root@localhost /]# echo mailadmin@hostway.co.kr > /var/qmail/alias/.qmail-mailer-daemon
[root@localhost /]# cd /var/qmail/alias/
[root@localhost /]# chmod 644 .qmail-root .qmail-postmaster .qmail-mailer-daemon
```

이제 qmail 구동 스크립트를 생성합니다. 스크립트 파일은 /etc/init.d/qmail 로 생성합니다.

```
#!/bin/sh

# For Red Hat chkconfig
# chkconfig: - 80 30
# description: the qmail MTA

PATH=/var/qmail/bin:/bin:/usr/bin:/usr/local/bin:/usr/local/sbin
export PATH

case "$1" in
  start)
    echo "Starting qmail"
        if [ -e /service/qmail-send ] ; then
                if svok /service/qmail-send ; then
                        svc -u /service/qmail-send
                else
                        echo qmail-send supervise not running
                fi
        else
                ln -s /var/qmail/supervise/qmail-send /service/
        fi

        if [ -e /service/qmail-smtpd ] ; then
                if svok /service/qmail-smtpd ; then
                        svc -u /service/qmail-smtpd
                else
                        echo qmail-smtpd supervise not running
                fi
        else
                ln -s /var/qmail/supervise/qmail-smtpd /service/
        fi

    if [ -d /var/lock/subsys ]; then
      touch /var/lock/subsys/qmail
    fi
   killall readproctitle > /dev/null 2>&1
    ;;
  stop)
    echo "Stopping qmail..."
    echo "   qmail-smtpd"
    svc -dx /service/qmail-smtpd /service/qmail-smtpd/log
        rm -f /service/qmail-smtpd
    echo "   qmail-send"
    svc -dx /service/qmail-send /service/qmail-send/log
        rm -f /service/qmail-send
    if [ -f /var/lock/subsys/qmail ]; then
      rm /var/lock/subsys/qmail
    fi
    ;;
  stat)
```

```
    svstat /service/qmail-send
    svstat /service/qmail-send/log
    svstat /service/qmail-smtpd
    svstat /service/qmail-smtpd/log
    qmail-qstat
    ;;
  doqueue|alrm|flush)
    echo "Flushing timeout table and sending ALRM signal to qmail-send."
    /var/qmail/bin/qmail-tcpok
    svc -a /service/qmail-send
    ;;
  queue)
    qmail-qstat
    qmail-qread
    ;;
  reload|hup)
    echo "Sending HUP signal to qmail-send."
    svc -h /service/qmail-send
    ;;
  pause)
    echo "Pausing qmail-send"
    svc -p /service/qmail-send
    echo "Pausing qmail-smtpd"
    svc -p /service/qmail-smtpd
    ;;
  cont)
    echo "Continuing qmail-send"
    svc -c /service/qmail-send
    echo "Continuing qmail-smtpd"
    svc -c /service/qmail-smtpd
    ;;
  restart)
    echo "Restarting qmail:"
    echo "* Stopping qmail-smtpd."
    svc -d /service/qmail-smtpd
    echo "* Sending qmail-send SIGTERM and restarting."
    svc -t /service/qmail-send
    echo "* Restarting qmail-smtpd."
    svc -u /service/qmail-smtpd
    ;;
  cdb)
    tcprules /home/vpopmail/etc/tcp.smtp.cdb /home/vpopmail/etc/tcp.smtp.tmp <
/home/vpopmail/etc/tcp.smtp
    chmod 644 /home/vpopmail/etc/tcp.smtp.cdb
    echo "Reloaded /home/vpopmail/etc/tcp.smtp."
    ;;
  help)
    cat <<HELP
   stop -- stops mail service (smtp connections refused, nothing goes out)
  start -- starts mail service (smtp connection accepted, mail can go out)
  pause -- temporarily stops mail service (connections accepted, nothing leaves)
   cont -- continues paused mail service
   stat -- displays status of mail service
```

```
      cdb -- rebuild the tcpserver cdb file for smtp
restart -- stops and restarts smtp, sends qmail-send a TERM & restarts it
doqueue -- schedules queued messages for immediate delivery
 reload -- sends qmail-send HUP, rereading locals and virtualdomains
   queue -- shows status of queue
    alrm -- same as doqueue
   flush -- same as doqueue
      hup -- same as reload
HELP
      ;;
   *)
      echo "Usage: $0 {start|stop|restart|doqueue|flush|reload|stat|pause|cont|cdb|queue|help}"
      exit 1
      ;;
esac

exit 0
```

스크립트 파일을 생성 한 후 퍼키션을 700으로 수정하며 부팅 시 자동 실행 될 수 있도록 권한을 부여 합니다.

```
root@local:/
[root@localhost /]# chmod 700 /etc/init.d/qmail
chkconfig --add qmail
chkconfig --level 3 qmail on
```

4. vpopmail 설치

vpopmail 에서 사용할 그룹과 계정을 생성한 후 소스 설치 합니다.

root@local:/

```
[root@localhost /]# cd /usr/local/src/
[root@localhost /]# groupadd vchkpw
[root@localhost /]# useradd -g vchkpw vpopmail
[root@localhost /]# tar zxvf vpopmail-5.5.0.tar.gz
[root@localhost /]# cd vpopmail-5.5.0
[root@localhost /]# ./configure --enable-auth-module=mysql ₩
--enable-tcprules-prog=/usr/local/bin/tcprules --enable-relay-clear-minutes=15
[root@localhost /]# make
[root@localhost /]# make install-strip

[root@localhost /]# cat ~vpopmail/etc/defaultdomain
hostway.co.kr

[root@localhost /]# cat ~vpopmail/etc/vpopmail.mysql
localhost|0|root|MySQL root 사용자의 패스워드 입력|vpopmail

[root@localhost /]# /usr/local/mysql/bin/mysqladmin -u root -p create vpopmail
[root@localhost /]# echo "127.0.0.1:allow,RELAYCLIENT=₩"₩"" > ~vpopmail/etc/tcp.smtp
[root@localhost /]# echo "211.115.223.215:allow,RELAYCLIENT=₩"₩"" >> ~vpopmail/etc/tcp.smtp

[root@localhost /]# mkdir /var/qmail/supervise/vpop
[root@localhost /]# cat /var/qmail/supervise/vpop/run
#!/bin/sh
VPOP_UID=`id -u vpopmail`
VPOP_GID=`id -g vpopmail`

exec /usr/local/bin/softlimit -m 9000000 ₩
        tcpserver -vRHl 0 -u $VPOP_UID -g $VPOP_GID 0 110 ₩
        /var/qmail/bin/qmail-popup hostway.co.kr ₩
        /home/vpopmail/bin/vchkpw /var/qmail/bin/qmail-pop3d Maildir 2>&1

[root@localhost /]# chmod 755 /var/qmail/supervise/vpop/run
```

위와 같이 모두 마무리 하였다면, 이제 sendmail 을 qmail 로 소프트 링크를 설정하여야 합니다.

root@local:/

```
[root@localhost /]# mv /usr/lib/sendmail /usr/lib/sendmail.ORI
[root@localhost /]# mv /usr/sbin/sendmail /usr/sbin/sendmail.ORI
[root@localhost /]# ln -s /var/qmail/bin/sendmail /usr/lib
[root@localhost /]# ln -s /var/qmail/bin/sendmail /usr/sbin
[root@localhost /]# ln -s /var/qmail/supervise/qmail-send /service/
[root@localhost /]# ln -s /var/qmail/supervise/qmail-smtpd /service/
[root@localhost /]# ln -s /var/qmail/supervise/vpop /service
```

5. Qmail 시작
모든 설치 작업이 완료되었습니다. 이제 tcp.smtp.db 파일을 생성 후 qmail 을 시작합니다.

```
root@local:/
[root@localhost /]# /etc/init.d/qmail cdb
[root@localhost /]# /etc/init.d/qmail start
```

6. 메일 계정 생성
메일 계정은 /home/vpopmail/vadduser 명령어를 이용합니다.
우선 hostway.co.kr 도메인을 등록 후 hostway@hostway.co.kr 메일 계정을 생성하도록 하겠습니다.

```
root@local:/
[root@localhost /]# cd /home/vpopmail/bin
[root@localhost /]# ./vadddomain hostway.co.kr
[root@localhost /]# ./vadduser hostway@hostway.co.kr
```

참고로 생성한 사용자의 삭제 및 패스워드 변경은 아래와 같습니다.

```
root@local:/
[root@localhost /]# cd /home/vpopmail/bin
[root@localhost /]# ./vdeldomain hostway.co.kr
[root@localhost /]# ./vdeluser hostway@hostway.co.kr
```

앞서 sendmail 작동 테스트와 마찬가지로 telnet localhost 25 로 qmail smtp 연결이 이뤄지는지, telnet localhost 110 과 POP3 로그인을 정상적으로 이뤄지는지 확인을 해보는 것으로 모든 설치 및 설정 작업이 완료되었습니다.

Chapter 8. 커널

커널은 리누스 토발즈에 의해서 최초로 개발되어 배포되었으며 리눅스 시스템에서 가장 중요한 부분이라고 할수 있습니다.
이 커널은 프로세스와 시스템 메모리를 관리하며 수많은 하드웨어 드라이버들을 제공하는 등 리눅스 상의 모든 동작을 제어하는 실제적인 핵심 소프트웨어 라고도 할수가 있습니다.

1. 소스 커널 컴파일 하는 이유

리눅스 배포판 CD에서 제공되는 커널은 여러 가지의 다양한 환경에서 작동 될수 있도록 다양하고 많은 기능들이 포함되어 있어 용량도 많고 무겁게 작동되는 경향이 있습니다.
소스 커널은 커널의 많은 기능중에서 현 시스템에서 사용하지 않는 불필요한 기능은 제거하고 자기 시스템에서 필요한 기능만 선택하여 컴파일 할수 있어 현재 시스템에 맞게 최적화된 커널을 만들수가 있습니다.

2. 소스 커널 다운 받기

커널 소스는 아래 소스 사이트(ftp.kernel.org)에서 다운 로드 할수 있습니다.
그리고 커널 소스 외에 커널 패치 파일이라는 것이 있는데 이는 현재의 커널 버전을 상위 버전으로 업그레이드할 수 있도록 해 주거나 현재 커널 버전의 문제점들과 기능이 보완된 수정 파일을 말합니다.

커널 FTP 공식 site : ftp://ftp.kernel.org/pub/linux/kernel/
우리나라사이트 : ftp://ftp.kr.kernel.org/pub/linux/kernel/
미러사이트 리스트 확인 : http://kernel.org/mirrors/

2.1 커널 버전 의미 및 종류

커널 소스 파일을 보면 숫자가 있는데 의미는 다음과 같이 보시면 되겠습니다.

커널 : linux-2.6.33.tar.gz
2 :(major number) 커널 주 버전
6 :(minor number) 홀수 일때: 개발 버전 / 짝수 일때: 안정 버전
33 : 패치된 횟수

커널 소스는 일반적으로 /usr/src/ 디렉토리에 올려 놓고 컴파일 작업을 하는데, 이 디렉토리와 심볼릭 링크를 할 수 있습니다.

종류는 아래와 같이 나누어 볼수가 있으므로 참고 하시기 바랍니다.
vanilla source : 리누스 토발즈가 릴리즈하는 것으로 오리지널 리눅스 소스.
bk snapshot : bk는 리눅스 소스 관리에 쓰이는 도구인 BitKeeper의 약자로 리눅스 2.6부터는 매일 리눅스 코드의 스냅샷을 배포함
bk tree : 리눅스의 여러가지 서브 시스템별로 관리되는 소스

2.2 최신 커널 확인 하기

서버에서 @finger.kernel.org 명령으로 확인 하시면 커널 버전별로 최신 버전을 확인 할수 있습니다.

```
root@local:/
[root@localhost /]# finger @finger.kernel.org
The latest linux-next version of the Linux kernel is:          next-20100611
The latest snapshot 2.6 version of the Linux kernel is:         2.6.35-rc2-git5
The latest mainline 2.6 version of the Linux kernel is:        2.6.35-rc2
The latest stable 2.6 version of the Linux kernel is:          2.6.34
The latest stable 2.6.33 version of the Linux kernel is:       2.6.33.5
The latest stable 2.6.32 version of the Linux kernel is:       2.6.32.15
```

2.3 소스 커널 다운 받기

첫번째 방법으로 ftp 명령으로 ftp.kernel.org 주소 접속후 경로 이동후 get 명령으로 최신 커널 버전파일을 다운로드 합니다.

```
root@local:/
[root@localhost~]# # ftp ftp.kernel.org
Trying 199.6.1.164...
Connected to ftp.kernel.org (199.6.1.164).
220 Welcome to ftp.kernel.org.
Name (ftp.kernel.org:root): root
331 Please specify the password.
Password:
230-                        Welcome to the
230-
230-                    LINUX KERNEL ARCHIVES
230-                        ftp.kernel.org
230-
230-                "Much more than just kernels"
:
230-
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp>
:
ftp> cd pub/linux/kernel/v2.6
250 Directory successfully changed.
ftp>
:
ftp> pwd
257 "/pub/linux/kernel/v2.6"
ftp>
:

ftp> get linux-2.6.33.3.tar.bz2
local: linux-2.6.33.3.tar.bz2 remote: linux-2.6.33.3.tar.bz2
227 Entering Passive Mode (199,6,1,164,98,206).
150 Opening BINARY mode data connection for linux-2.6.33.3.tar.bz2 (66199228 bytes).
```

두번째 방법으로는 ftp://ftp.kernel.org 와 같이 URL로 ftp 접속후 최신 버전이 있는 디렉토리로 이동후 해당 최신 커널 파일의 경로를 확인후 wget 명령으로 서버에서 직접 다운로드 할수 있습니다.

2.4 커널 작업전 확인 내용(필요한 패키지)

커널 컴파일이란 커널 소스에서 제공하는 설정 요소들 가운데 시스템에 맞는 설정 요소들을 선택하여 컴파일러를 통하여 새로운 커널 이미지와 모듈을 생성하는 과정을 말합니다.

커널 컴파일을 하기 위해서는 현재 시스템 사양에 대하여 정확히 알고 있어야 합니다.
시스템에서 사용하지 않는 불필요한 모듈들이 포함되어 컴파일될 경우 시스템의 리소스등이 낭비 되어 질수 있습니다.

서버의 구성(사양) 확인 예)
하드디스크 종류 :IDE,SCSI,SATA,SAS / 스카시 콘트롤러(사용시),메인보드 칩샛 종류
랜카드 : 제조사 및 모델명
CPU : 개수 및 사양
파일시스템 종류 : ext2,ext3,ext4등 (ext3를 사용으로 built-in해줌)

커널 컴파일 방식 선택 : built-in(커널에 포함시킴) / module (모듈 형태로 만듦)

커널 소스 폴더에서 보시면 README 파일과 Changes(Documentation/Changes) 파일이 있는데 꼭 읽어 보시기 바랍니다. Changes 파일은 새 커널을 문제 없이 설치하기 위해 업그레이드 되어야 하는 프로그램 리스트및 설치 방법을 보여줍니다
Changes 화일에서는 아래와 같이 커널 컴파일시 필요한 프로그램들을 안내 하고 있으므로 해당 패키지 프로그램의 버전을 확인 하여 해당 프로그램이 설치 안되어 있거나 버전이 낮은 경우 설치할 필요가 있습니다.

Documentation/Changes 예)

o Gnu C	3.2	# gcc --version
o Gnu make	3.80	# make --version
o binutils	2.12	# ld -v
o util-linux	2.10o	# fdformat --version
o module-init-tools	0.9.10	# depmod -V
o e2fsprogs	1.41.4	# e2fsck -V
o jfsutils	1.1.3	# fsck.jfs -V
o reiserfsprogs	3.6.3	# reiserfsck -V 2>&1|grep reiserfsprogs
o xfsprogs	2.6.0	# xfs_db -V
o squashfs-tools	4.0	# mksquashfs -version
o btrfs-progs	0.18	# btrfsck
o pcmciautils	004	# pccardctl -V
o quota-tools	3.09	# quota -V
o PPP	2.4.0	# pppd --version
o isdn4k-utils	3.1pre1	# isdnctrl 2>&1|grep version
o nfs-utils	1.0.5	# showmount --version
o procps	3.2.0	# ps --version

o	oprofile	0.9	# oprofiled --version
o	udev	081	# udevinfo -V
o	grub	0.93	# grub --version
o	mcelog	0.6	
o	iptables	1.4.1	# iptables -V

3. 커널 컴파일 방법

컴파일 방법으로는 커널 이미지로만 사용하는 단일 컴파일 방법과 모듈과 같이 올려서 사용하는 방법이 있습니다.

방법 1) 단일 커널 이미지로 사용할 경우(built-in)
Loadable module support에서 선택을 해제 해야 하고 모든 옵션에 [*]로 표시가 됩니다.
nomodules 명령을 사용하여 grub.conf 파일에 kernel 줄을 추가하거나 lilo.conf 파일에 append=nomodules 줄을 추가해야 합니다.

방법 2) 커널에서 모듈로 올려서 사용할 경우 (module)
커널 2.6 에서부터는 모듈을 사용하려면 module-init-tools 패키지를 설치 해야 에러 없이 컴파일을 할수 있습니다.
Loadable module support에서 선택을 하고 메뉴에서 < > 부분은 <M>로 선택 하여야 합니다.

4. 커널 컴파일 순서

방법 1) 단일 커널 이미지로 사용할 경우(built-in)

```
make mrproper
make menuconfig
make bzImage
make install
```

방법 2) 커널에서 모듈로 올려서 사용할 경우 (module)

```
make mrproper
make menuconfig
make bzImage

make modules
make modules_install
make install
```

- **make mrproper** (커널 설정 초기화)
  이 명령은 커널 소스를 원래 상태로 초기화 할때 사용하는 명령으로, 이미 설정된 커널 성정값을 모두 초기화 시키고 컴파일된 오브젝트 파일을 모두 제거하여 커널 상태를 원래의 소스 상태로 되돌려 놓는 초기화 명령입니다.

- **make menuconfig** (커널 메뉴 설정)

make menuconfig, make xconfig, make oldconfig 등의 여러가지 방법으로 커널 설정을 할 수 있습니다.
make menuconfig -> 이 방법을 가장 많이 사용하고 여기서도 이 방법으로 합니다.
make xconfig -> X-window 에서 작업할수 있는 화면입니다.
make oldconfig -> 이전 커널의 config 파일을 복사하여 사용시 사용하는 명령
allmodconfig -> 모든 기능을 모듈로 선택시 방법입니다.
allyesconfig -> 모든 옵션을 커널에 추가 할때 방법입니다.

- **make 또는 make bzImage**

make menuconfig를 실행하고 난 이후 저장한 설정값을 이용하여 make를 실행하면 해당 옵션에 맞게 컴파일이 진행합니다. 하드웨어 사양에 맞게 커널을 컴파일합니다.
make 를 수행하고 나면 소스 디렉토리에 vmlinux파일과 vmlinux.o 파일이 생성됩니다.
arch/i386/boot/디렉토리 아래 bzImage 파일을 찾을 수 있습니다

- **make install**

System.map-2.6.33, vmlinuz-2.6.33, initrd-2.6.33.img 화일을 생성하고 컴파일된 새 커널 이미지화일과 관련 파일들을 /boot 폴더로 복사합니다.

4.1 모듈 컴파일

모듈이 생성되는 디렉토리는 /lib/modules/2.6.33 이고 만약 현재 커널의 모듈을 다시 생성하려고 한다면 현재 커널의 모듈 디렉토리를 지우고 다시 컴파일 하시기 바랍니다.

- **make modules**

make menuconfig 에서 <M>로 선택된 옵션들을 compile 합니다.

- **make modules_install**

module을 /lib/modules/<커널버전> 디렉토리에 생성하고 설치 합니다.

- **make install**

System.map-2.6.33, vmlinuz-2.6.33, initrd-2.6.33.img 화일을 생성하고 컴파일된 새 커널 이미지화일과 관련 파일들을 /boot 폴더로 복사합니다.
grup.conf, menu.list 파일을 자동으로 변경해 주므로 따로 부트로더를 설정할 필요가 없습니다.

4.2 Initrd 이미지 만들기

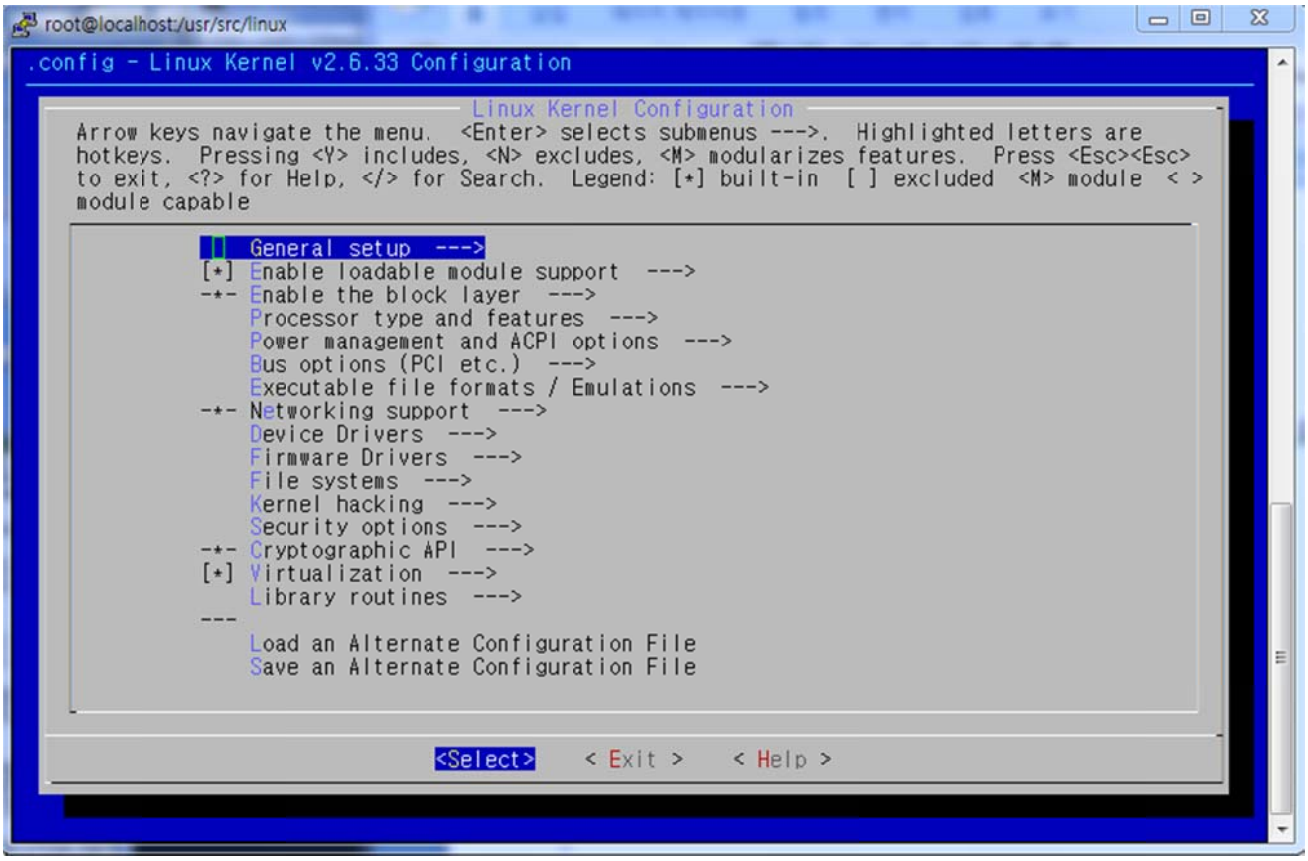
모듈을 만들고, 때에 따라서 initrd 이미지도 만들어줄 필요가 있는데 이럴 경우 mkinitrd 를 사용합니다. (일반적으로 커널 버전으로 함)

```
root@local:/
[root@localhost /]# cd /lib/modules
[root@localhost /]# mkinitrd 2.6.33 -o /boot/initrd-2.6.33
```

5. menuconfig 세부 옵션

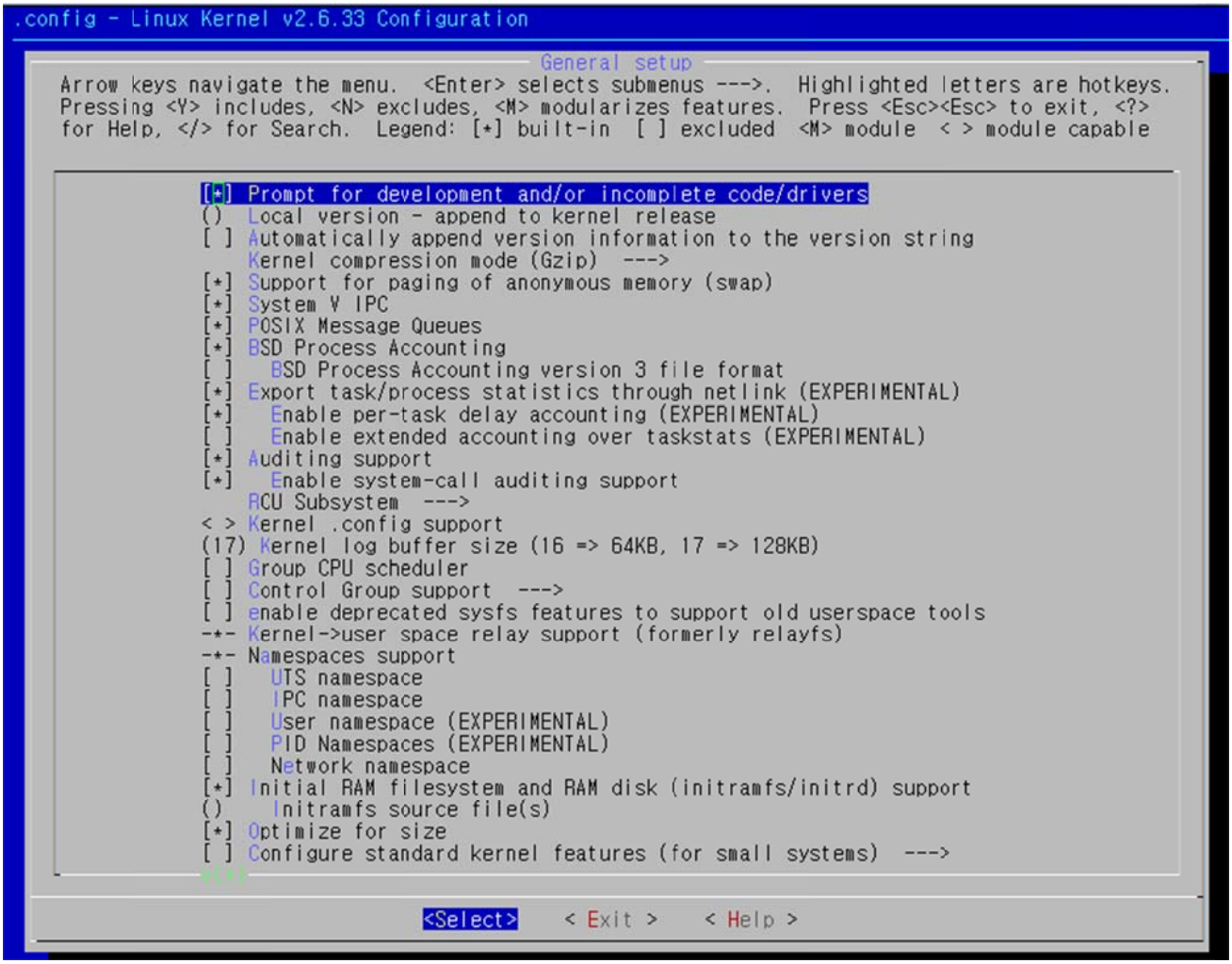
자신 리눅스 환경에 맞게 질문에 Y,N,M을 설정해 주고, Y는 커널에 포함, N은 제외, M은 모듈을 의미합니다. 해당 라인으로 키보드 방향키로 이동후 스페이스 바를 이용하여 Y,N,M을 선택하거나 해당 라인에서 키보드 y,n,m을 직접 눌러 설정해 주시면 됩니다.
"-->"모양이 있는 라인은 하위 설정값이 있기 때문에 <Enter> 또는 <Space bar>를 이용하여 하위 설정 화면으로 전환할 수 있습니다. 반대로 상위 설정화면으로 전환하기 위해선 <Esc>키를 두번 누르거나 설정화면의 아래 <Exit>를 선택하면 상위 화면으로 전환할 수 있습니다.

아래 menuconfig 부분은 커널 2.6.33 버전에서 볼 수 있는 목록입니다.



[그림 8-1] menuconfig

5.1 General setup



[그림 8-2] General setup

커널의 일반 기능에 대한 설정부분 입니다.
프로세스의 통신을 위한 SystemV 유닉스에서 지원하는 IPC 형식을 여기서 선택할 수 있으며 공유 메모리(Shared Memory)도 지원하며 또한 acct 함수로 프로세스에 대한 여러 가지 정보를 파일에 저장할 수 있는 설정도 제공됩니다.

[*] Prompt for development and/or incomplete code/drivers
-> 커널에 개발중인 새로운 기능과 새로운 드라이버가 포함되어 있는데 이것을 사용할 것인지를 설정해 주는 옵션입니다.

() Local version - append to kernel release
-> uname 명령으로 커널 버전명을 확일할 때 나타나도록 하는 옵션

[*] Automatically append version information to the version string
-> 현재 tree의 top에 속하는 git tags를 찾음으로써
현재 tree가 release tree 이면 자동적으로 localversion에 추가함

[*] Support for paging of anonymous memory (swap)
-> swap devices나 swap files을 사용 할 것인지 설정하는 것으로 선택하는 것이 좋습니다.

[*] System V IPC
  -> 이 설정은 IPC를 지원하게 하며 Shared Memory도 여기서 지원합니다.
    프로세스 사이에서 동기화와 정보교환을 위한 라이브러리 함수와 시스템 콜 모음입니다.
    특히, Dos emulator와 같은 프로그램을 사용하려 한다면 동기화를 위해 선택해야 합니다.

[*] POSIX Message Queues
  -> 커널에서 전역적으로 관리되며 모든 프로세스에서 접근이 가능하도록 되어 있으므로
    하나의 메시지큐 서버가 커널에 요청해서 메시지큐를 작성하게 되면, 메시지큐의 접근자를
    아는 모든 프로세스는 동일한 메시지큐에 접근함으로써, 데이터를 공유할 수 있게 된다.
    포직스 메시지 큐에 관한 옵션으로 커널에서 지원하도록 하는 것이 좋습니다.

[*] BSD Process Accounting
  -> BSD 계열 프로세스 어카운팅을 가능하도록 설정하는 것으로 이것을 선택하면 프로세스가
    존재하는 시간에 사용자가 커널 프로세스 정보를 알 수 있습니다.

[*]   BSD Process Accounting version 3 file format
  -> 프로세스가 끝날 때 커널에 의해 프로세스 정보가 파일에 기록.

[*] Auditing support
  -> SElinux와 같은 다른 커널 서브 시스템과 함께 사용되는 구조 검사 기능을 활성화 합니다.
    Selinux는 기록이나 검사(감사), 정책에 의해 허용되었거나 거부된 접근시도에 대한
    광범위한 기능을 가지고 있고 이 감사 메시지를 종종 AVC message라고 부른며 접그시도가
    허용되었거나 거부되었거나 또는 소스나 타겟의 보안 문맥, 그리고 접근 시도에 포함된
    자원에 대하여 세부정보를 제공합니다.

[*]   Enable system-call auditing support
  -> 독립적으로 사용되거나 SElinux와 같은 다른 커널 서브 시스템과 함께 사용되는 구조
    검사의 system-call 기능을 활성화 합니다.

RCU Subsystem --->
  -> RCU는 updater들과 cuncurent하게 reader들을 동작하도록 하여 시스템의 scalability를
    향상시키는 lock free 메커니즘입니다.

[ ] Kernel .config support
  -> 현재 커널이 빌드될 때 사용된 설정을 /proc/config.gz에서 읽을 수 있도록 해줌.

(15) Kernel log buffer size (16 => 64KB, 17 => 128KB)
  -> Kernel의 로그 메시지들을 저장해야 하므로 Log Buffer라는 circular buffer가 정의됨

[*] Group CPU scheduler
  -> 같은 task group에서는 cpu scheduler가 task groups을 인식하고 cpu bandwidth
    allocation을 제어하도록 합니다.
    프로세스들을 하나의 group으로 만들기 위해서 사용합니다.

[*] Control Group support --->
  -> 그룹으로 묶인 프로세스들의 집합을 지원하기 위한것으로 process control subsystem을
    사용할 것인지를 설정 합니다.

[*] Create deprecated sysfs layout for older userspace tools
  -> sysfs의 layout을 구 버전으로 바꿔주는 옵션

  -*- Kernel->user space relay support (formerly relayfs)
  -> 커널 영역의 많은 양의 data를 user 영역으로 전달을 지원하기 위해 사용함

[*] Initial RAM filesystem and RAM disk (initramfs/initrd) support
  -> 램을 하드 디스크와 같은 블록 디바이스처럼 사용하는 기능으로 Linux를 인스톨하는 동안
     램에 작은 루트 파일시스템을 생성하기 위해 사용되기도 합니다.

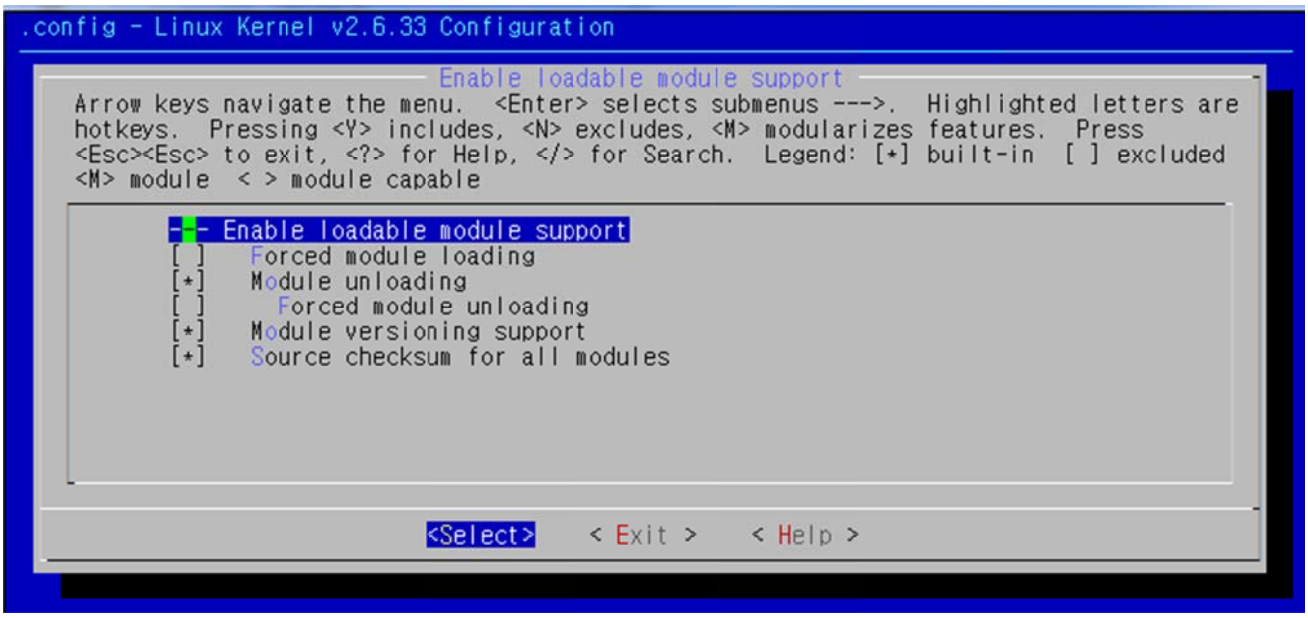
  () Initramfs source file(s)
    -> inital ram filesystem의 이미지를 생성합니다.

  [*] Optimize for size
    -> gcc 옵션을 -O2 대신에 -Os를 사용 합니다.

  [ ] Configure standard kernel features (for small systems) --->
    -> non-standard kernel 을 특정한 환경에서 사용하기 위한 것입니다.

  [*] Disable heap randomization
    -> 무작위의 heap의 위치는 harder를 이용해 heap을 만들지만
       libc5를 기반으로 만들어진 binary들은 사용 할 수 없습니다.
       이 옵션은 bootup에 기본을 heap randomization disable로 변경 합니다.

5.2 Enable loadable module support



[그림 8-3] Enable loadable module support

이 설정은 커널 모듈에 대한 설정으로 Module 이란 Kernel에 포함되지 않고 Kernel이 운영되고 있는 상태에 모듈 파일을 읽어 들여서 Kernel의 기능(feature)을 더하는 것입니다.
커널에서 모듈을 사용 하지 않는 경우는 커널 옵션에서 선택된 기능들이 커널이미지 파일로

만들어져 시스템에 상주하며 작동을 합니다만 모듈로 만들어 진 경우는 이미지화일과는 달리 모듈로 존재 하며 시스템에서 그 기능을 사용시에만 로드되어 작동을 합니다.

--- Enable loadable module support
->커널 옵션 기능들을 모듈로 만들어 필요할 때마다 모듈로 적재하고자 할 경우에 설정합니다.

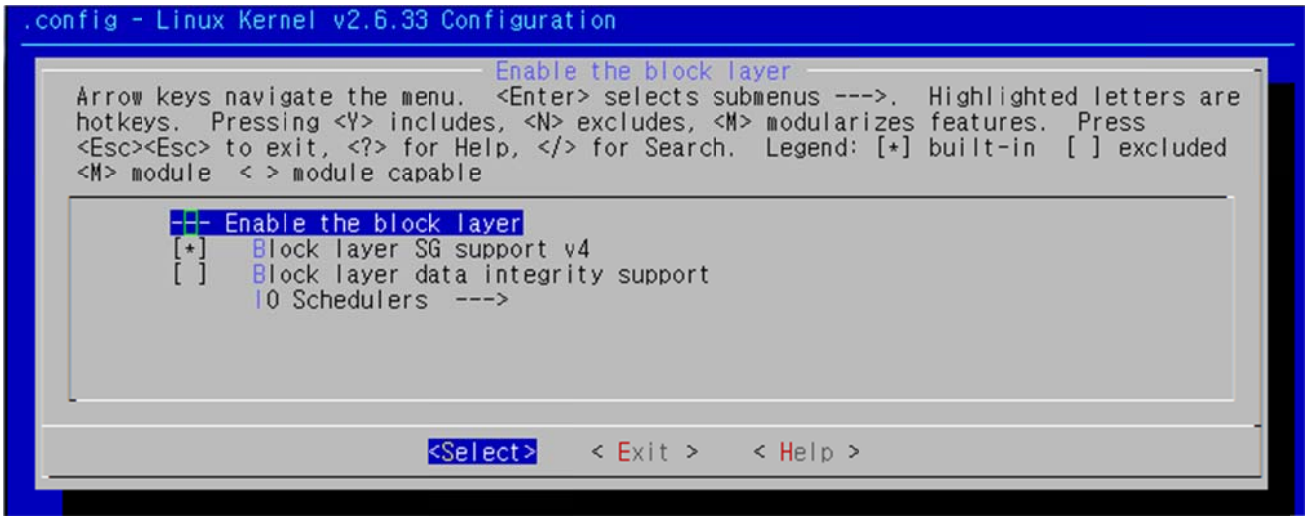
[ ] Forced module loading
-> 이 설정은 Module unloading을 선택시 보이는데 현재 커널이 모듈에 포함된 기능을 쓰고 있더라도 강제로 모듈을 지울수 있도록 합니다. 정상적인 시스템에서는 되도록이면 사용 안하는것이 좋습니다.

[*] Module unloading
-> 모듈을 커널에서 내릴 수 있는 (제거) 옵션. Forced module unloading 옵션은 커널이 모듈 기능을 사용중에 강제적으로 모듈을 내리는 것으로 일반적으로 이 옵션은 시스템에는 바람직한 옵션은 아닙니다

[*] Module versioning support
-> 버전이 다른 커널에서 만들어진 모듈 또는 커널에서 없는 다른 특별한 모듈을 사용할 수 있도록 해주는 옵션이며 이 옵션은 바이너리 형태로 지원되는 드라이버의 커널 버전이 사용 중인 커널 버전과 다를 경우에도 커널에 적재될 수 있도록 할 때 유용한 기능입니다.

[*] Source checksum for all modules
-> 모듈을 띄울 때 modprobe 명령으로 사용해야 하지만, 이옵션을 선택하면 자동으로 모듈이 커널에서 작동 됩니다.

5.3 Enable the block layer



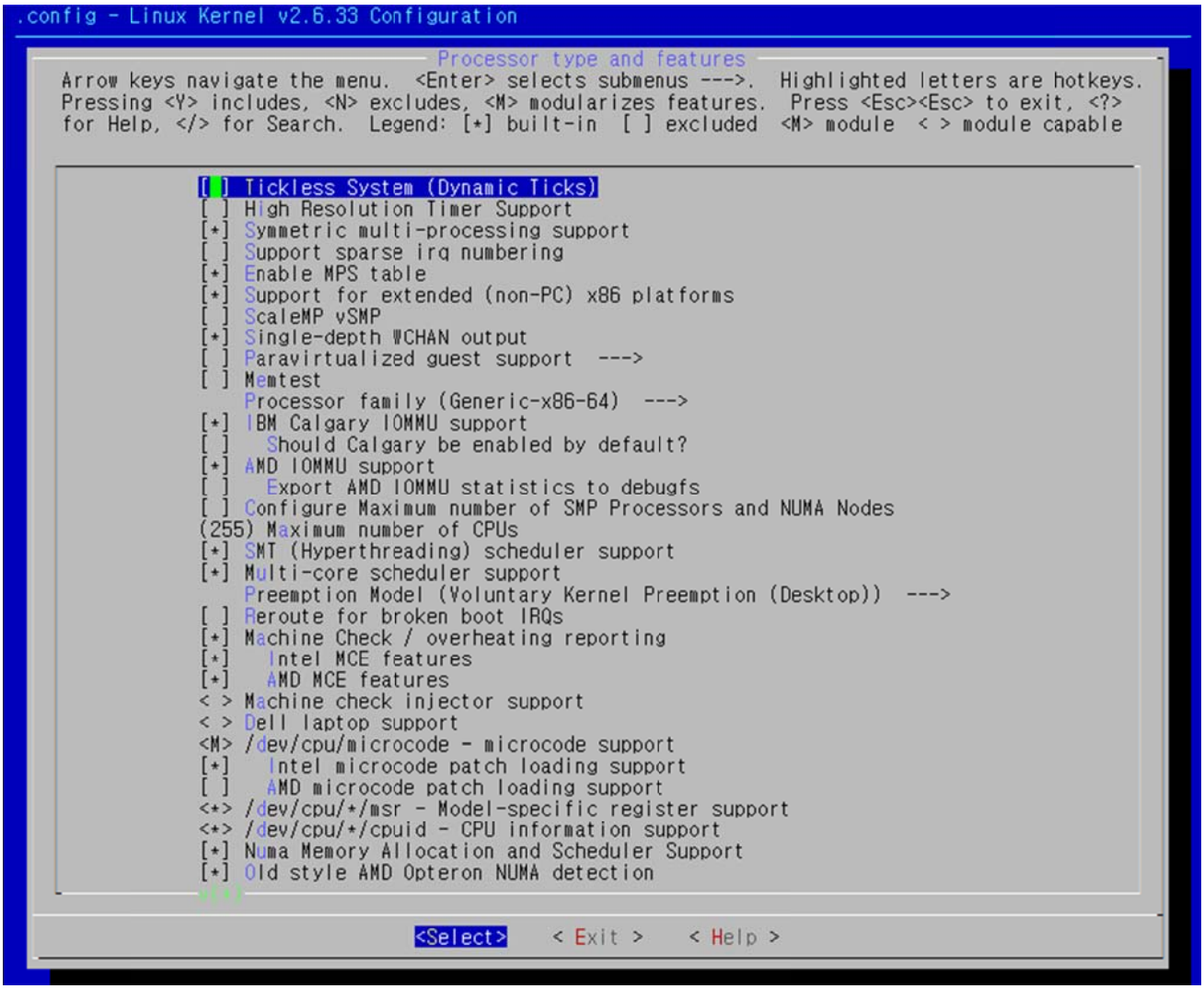
[그림 8-4] Enable the block layer

커널 2.6에서 커널 확장성과 퍼포먼스를 위해 재작성되었는데 2.6 커널의 block I/O 버퍼(kiobuf)는 PAGE_SIZE 보다 큰 I/O 요청을 허용하고 SCSI 지원에 있어서 많은 향상을 보인다고 할수 있습니다.

--- Enable the block layer

```
[*]   Block layer SG support v4
[ ]   Block layer data integrity support
IO Schedulers  --->
```

5.4 Processor type and features



[그림 8-5] Processor type and features

```
Processor type and features  --->
```

리눅스는 각각 여러 가지 하드웨어 기기에서 동작하는데 하드웨어 기기의 특성마다 다른 옵션을 선택할 수 있습니다. 여기서는 x86 CPU를 쓰는 하드웨어에 대한 설정을 할것이며 커널 2.6 에서는 같은 CPU를 사용하더라도 세부적인 하드웨어까지 지원하도록 하였습니다.

리눅스에서는 여러 개의 CPU를 쓸 수 있도록 하는 SMP 기능이 포함되어 있습니다. 만약 자신의 시스템에 CPU가 하나 이상 설치되어 있다면 이 옵션을 켜두는 것이 좋습니다. 하이퍼쓰레딩을 지원하는 펜티엄 4가 있으면 마찬가지로 SMP를 사용하는 것이 좋습니다.

또한 대형 메모리 지원 옵션으로 현재 시스템의 램 크기가 1GB 이상이면 크기에 알맞은 시스템의 메모리를 모두 쓸 수 있다. 그 외에 수치연산 프로세서가 없는 CPU를 대신해

수치연산 프로세서가 동작하는 것처럼 흉내 내는 기능을 포함하고 있으며, P6 이상의 프로세서에 있는 MTRR 레지스터 지원을 사용함으로써 그래픽카드에서 많은 양의 데이터를 전송할 때 속도 향상이 있다고 한다.

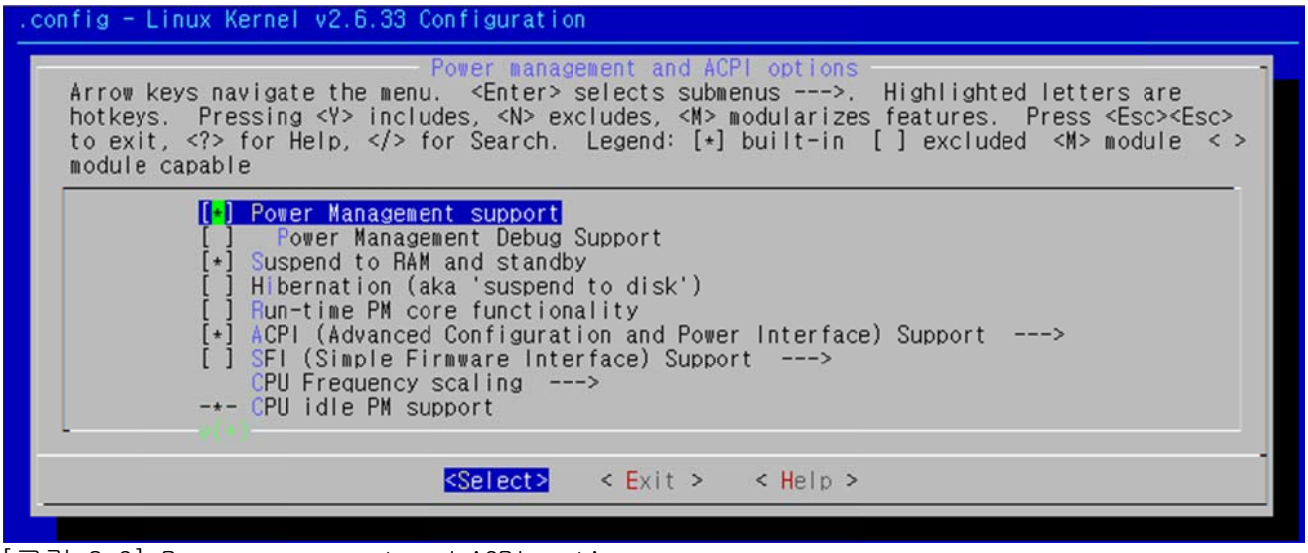
```
Processor type and features --->

 [*] Symmetric multi-processing support
  -> 하이퍼 쓰레딩을 지원하는 단일 프로세서나 듀얼 프로세서 시스템에서는 이 옵션을
     선택합니다

 [*] Enable MPS table
  Subarchitecture Type (PC-compatible)  --->
  -> PC-compatible으로 선택합니다

 [ ] Memtest
    Processor family (Pentium-III/Celeron(Coppermine)/Pentium-III Xeon)  --->
  -> 자신의 시스템에 장착된 프로세서 종류를 선택해 줍니다.

[*] Generic x86 support
  -> x86 기반의 프로세서의 일반적인 최적화를 위해서 이 옵션을 선택해 줍니다.
[ ] HPET Timer Support
  -> 커널 내부 타이머로 8254 대신에 차세대 타이머인 HPET를 사용합니다.
     리눅스와 바이오스에서 이 기능을 지원하지 않으면 8254 타이머가 사용됩니다.

[*] Machine Check Exception
  -> 이 기능은 CPU에서 과열이나 컴포넌트 오류 등의 문제점이 발생하였을 때 CPU가 커널이
     알 수 있도록 해 줍니다.
     문제의 심각성에 따라서 커널은 콘솔 상에서 단지 경고 메시지를 보여 주거나. 시스템을
     정지시켜 시스템을 보호할 수 있는 기능을 제공하며, 이 기능을 지원하지 않는
     시스템에서도 이 옵션을 사용할 수 있으므로 기본적으로 선택하는 것이 좋습니다.

[*] /dev/cpu/microcode - microcode support
  -> microcode_ctl로 P6이상의 CPU에서 지원하는 마이크로코드를 업데이트 할 수 있습니다.

  File systems 항목 가운데에서 "/dev 파일시스템 지원" 옵션과 함께 설정해야 하며,
  펜티엄프로, 펜티엄2~4 등의 IA 등의 IA32 인텔계열 프로세스에서 마이크로코드를
  업데이트할 수 있게 해 주는 옵션입니다.그러나 리눅스 커널에서는 포함되어 있지 않은 실제
  마이크로코드 바이너리 데이터를 가지고 있어야 합니다. 모듈로 이 기능을 선택하는 경우
  /etc/modules.conf 파일에 alias char-major-10-184 microcode를 설정해 주어야 합니다.

[*]   Intel microcode patch loading support
[*]   AMD microcode patch loading support
[ ] /dev/cpu/*/msr - Model-specific register support
  -> 특권을 가진 프로세스들이 x86 Model-Specific Registers(MSRs)에 접근할 수 있도록 해
     주는 장치로 /dev/cpu/0/msr부터 /dev/cpu21/msr 디바이스에 주 장치번호 202와 부
     장치번호 0~31을 가집니다. 주로 멀티프로세서 시스템에서 적용 합니다.
```

[*] /dev/cpu/*/cpuid - CPU information support
  -> /dev/cpu/0/cpuid부터 /dev/cpu/31/cpuid에 주 장치번호 203과 부 장치번호 0~31을 가지는 디바이스로 프로세스들이 특정 프로세서에서 실행될 수 있도록 지시하는 x86 CPUID에 접근할 수 있도록 해 줍니다. 멀티프로세서 시스템에서 이 옵션을 적용 함

High Memory Support (64GB) --->
  -> x86 기반의 리눅스 시스템에서 물리적인 메모리를 최대 64기가바이트까지 사용할 수 있는데 32 비트 계열의 인텔 프로세서에서는 메로리 주소 공간이 4기가바이트까지만 지원되어 그 이상의 물리적인 메모리를 가지고 있더라도 커널에서 모든 메모리가 영구적으로 맵핑되지 못하는데 이 때 영구적으로 맵핑되지 않는 물리적인 메모리르 상위 메모리라고 부르는데 이 상위 메모리 설정을 위한 옵션입니다. 시스템의 램 메모리가 1기가 바이트 이하일 경우는 off 1기가 내지 4기가일 경우에는 4GB로 설정하고 4기가 이상일 경우에는 64GB로 상위 메모리를 설정합니다.

[*] Math emulation
  -> 386이나 486SX와 같이 수학 코프로세서를 가지고 있지 않은 시스템에서 에뮬레이터 기능이 요구될 때 선택하는 옵션
[*] MTRR (Memory Type Range Register) support
  -> 이 옵션은 인텔 Pentium Pro와 Pentium 2 시스템에 있는 MTRR을 지원하는 것으로 프로세스가 메모리 영역에 접근할 수 있도록 제어해 주는 기능을 합니다. 이를 선택하여 PCI 또는 AGP VGA 카드의 성능이 향상되며 X 서버도 이 기능을 사용하므로 Pentium Pro 이상의 시스템을 사용하고 있다면 선택해 주는 것이 좋습니다.
[*]   MTRR cleanup support
(0)   MTRR cleanup enable value (0-1)
(1)   MTRR cleanup spare reg num (0-7)

5.5 Power management and ACPI options



[그림 8-6] Power management and ACPI options

---Power management and ACPI options
 전원 관리 옵션을 선택할 수 있습니다. 현재 리눅스에서는 APM 과 ACPI 라는 두 가지 방식 중 하나를 전원 관리 기능으로 쓸 수 있습니다. 그리고 윈도우의 하이버네이션 기능과 동일하게 시스템의 메모리를 하드디스크에 저장시켰다가 다음 부팅 때 읽어들여 이전 상태로 되돌리는

기능을 제공합니다.

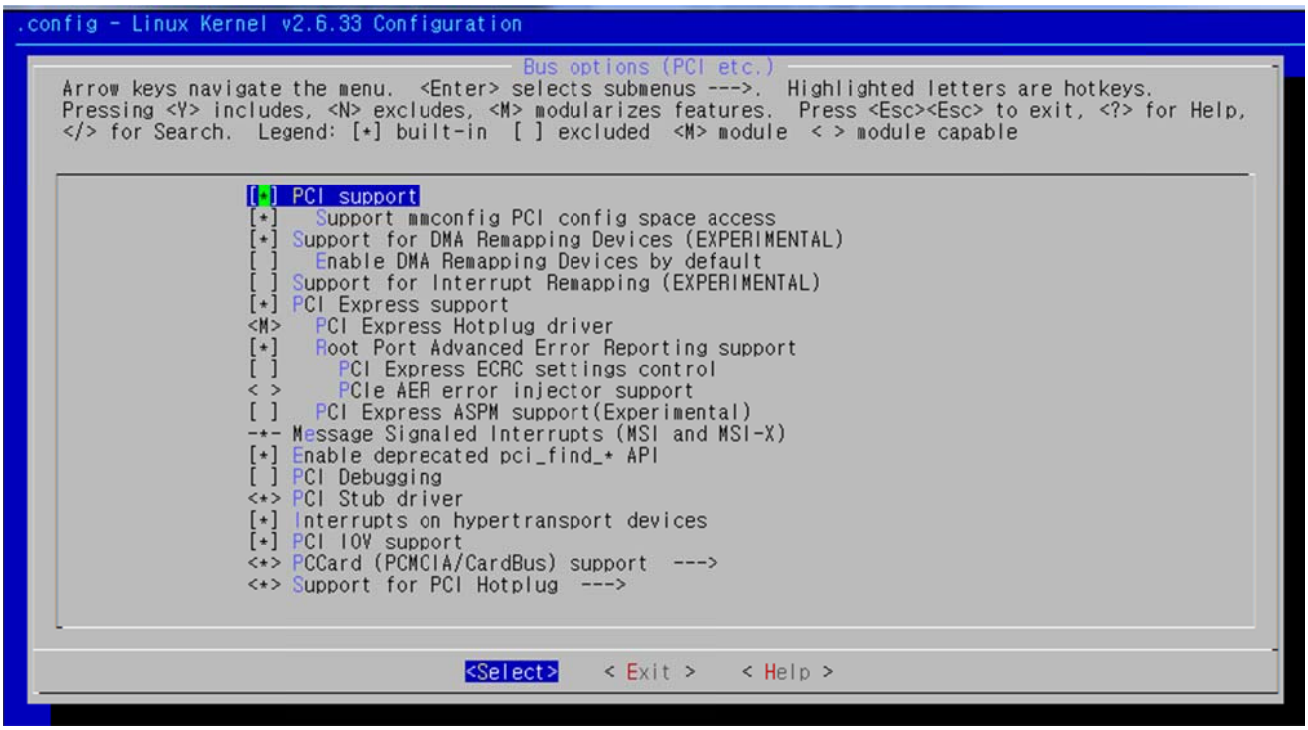
[*] Power Management support
  - > 전원 관리 기능은 시스템이 꺼지도록 하거나 사용하지 않을 경우 절전 모드로 전환하여 전원 소비를 줄여 주는 기능으로 이 옵션이 가능하도록 선택 합니다.

[*] Suspend to RAM and standby
[*] ACPI (Advanced Configuration and Power Interface) Support --->
  -> ACPI는 전원 관리와 하드웨어 설정을 OS와 유기적으로 하기 위한 표준이며 acpi 기능을 쓰고 싶다면 이 옵션을 켜고 acpi daemon을 쓰도록 합니다.

[*] APM (Advanced Power Management) BIOS support --->
  -> APM은 BIOS에서 전원 관리를 하기 위한 표준입니다. apm 기능을 쓰고 싶다면 이 옵션을 켜고 apmd를 설치하는 것이 좋습니다. ACPI와 APM은 동시에 사용할 수 없으며 만약 두 기능을 동시에 켜놨다면 ACPI가 동작할 것입니다.

CPU Frequency scaling --->
  -> CPU Frequency scaling은 각종 모바일 및 임베디드 CPU의 소비전력을 절약하기 위한 기능을 쓸 수 있게 해줍니다.

5.6 Bus options



[그림 8-7] Bus options

[*] PCI support
PCI access mode (Any) --->
  -> PCI 주변 기기를 탐색하는 방법으로 "Bios"를 선택하면 바이오스가 이용되고, "Direct"를 선택하면 바이오스가 이용되지 않습니다.
  "Any"를 선택하면 커널이 직접 탐색하고 실패하는 경우 바이오스에 의해서 실행하도록

합니다

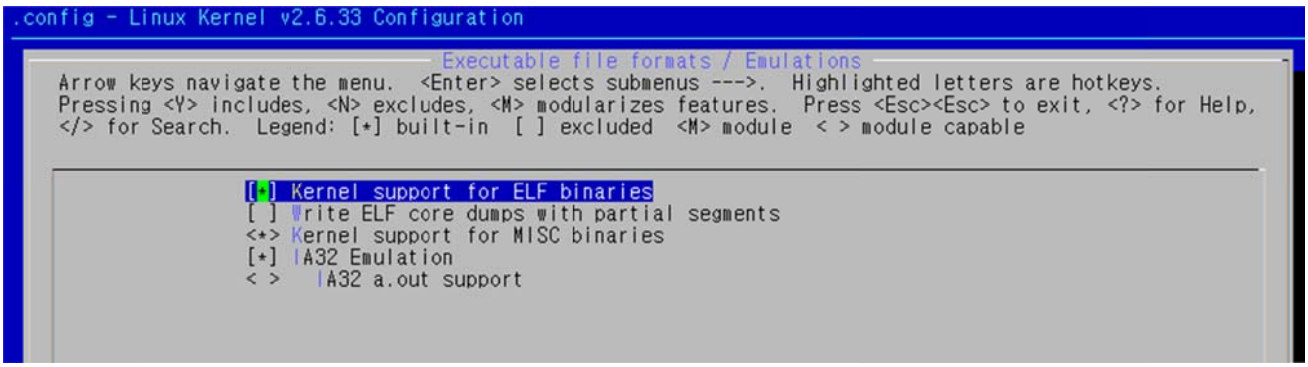
[*] PCI Express support
[*] ISA support
-> ISA 방식의 이더넷 카드나 SCSI 카드를 이용한다면 이 옵션을 선택합니다.

[*] EISA support
-> EISA(Extended Industry Standard Architecture)버스는 IBM 마이크로 채널 버스를 대체하기 위해 개발된 것으로 요즘에는 사용하지 않는 ISA버스 체계이므로 설정하지 않습니다

[*] MCA support
-> MicroChanneL Architecture는 IBM PS/2 머신에서 찾아볼 수 있는데 국내에서는 잘 쓰이지 않습니다. [N]을 선택합니다.

[*] Legacy MCA API Support
[*] Support for PCI Hotplug --->
-> PCMCIA-나 PC-카드 등은 컴퓨터가 동작하는 동안 장치를 새로 꽂거나 뽑는 일을 할 수 있습니다.
/proc/sys/kernel/hotplug에 프로그램을 등록해서 시스템이 구동되는 도중에 주변장치를 붙이고 뗄수 있는 기능을 사용하도록 합니다. 커널에서 지원하는 핫플러그 드라이버를 선택합니다.

5.7 Executable file formats



[그림 8-8] Executable file formats

Executable file formats --->
실행 파일 로더에 대한 옵션입니다. 일반적인 유닉스 시스템에서 많이 사용되는 ELF는 꼭 커널에 포함하도록 합니다.. 그리고 예전의 유닉스에서 사용되던 a.out 형식 지원은 호환성을 위해 모듈로 남겨두었으나 거의 쓰이지 않으므로 커널에서 제거해도 상관없음.

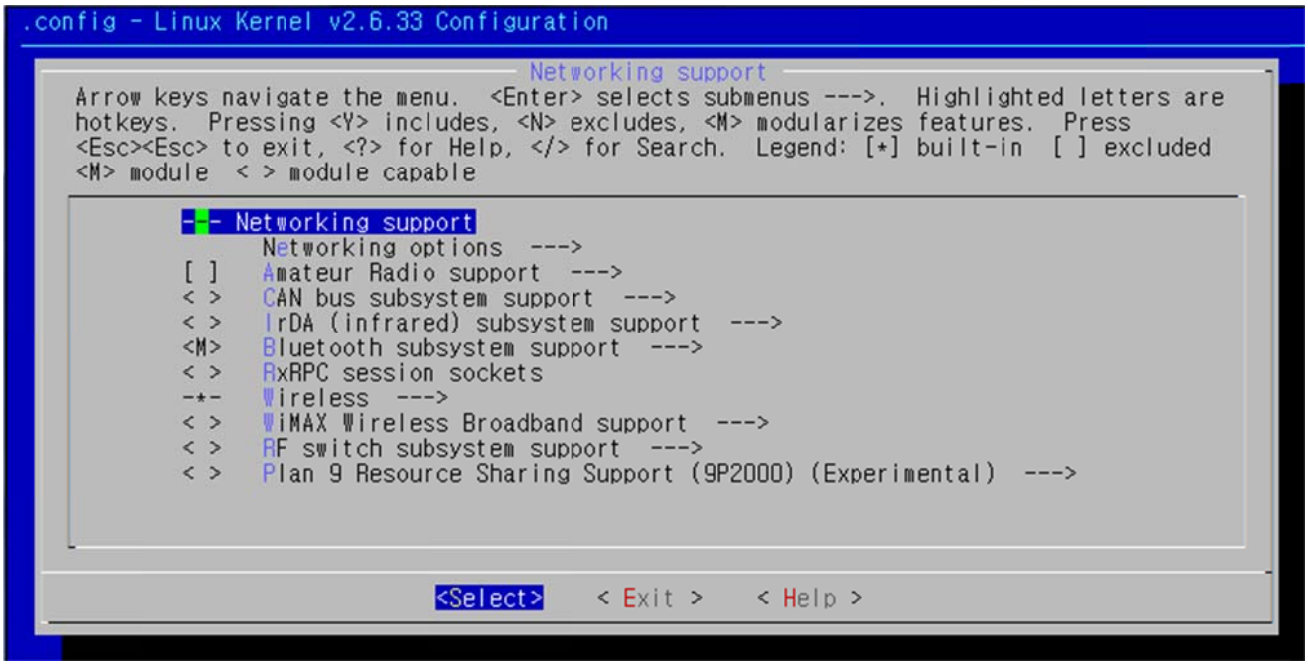
[*] Kernel support for ELF binaries
-> ELF(Executable and Linkable Format)은 서로 다른 OS나 Architecture에 호환이 될 수 있도록 표준화된 Binary File Format입니다. 또, ELF는리눅스 바이너리 포맷의 표준이기도 하므로 반드시 [Y]를 선택합니다. 리눅스 커널과 많이 쓰이는 모든 프로그램들이 ELF 포맷으로 컴파일 됩니다. ELF는 a.out에 비해 진보된 기능들을 포함합니다.

[*] Write ELF core dumps with partial segments

[*] Kernel support for a.out and ECOFF binaries
  -> a.out(aSSembLer.output)은 ELF 이전에 쓰이던 바이너리 포맷으로 점차 사라져 가는 포맷입니다. ELF는 리눅스 바이너리 표준 포맷이므로 선택하여 줍니다.

[*] Kernel support for MISC binaries
  -> Java, EmacS-LiSp, DOS 실행파일 등을 커널 바이너리 클래스에 등록했다면 인터프리터를 거치지 않고 셸 프롬프트에서 파일 이름을 쓰는 것만으로 간단히 프로그램을 시작할 수 있습니다. [Y] 라고 답했다면 "KerneL Support for JAVA binarieS", "kerneLSupport for Linux/InteLELF bianrieS" 기능은 필요 없습니다.

5.8 Networking support



[그림 8-9] Networking support

--- Networking support
리눅스에서 제공하는 물리적인 네트워크 장치에 대한 여러 가지 드라이버를 제공합니다. 일반적으로는 이더넷으로 연결이 가능하지만 환경에 따라 PPP, SLIP, 와이어리스, 토큰 링, WAN, ATM 등의 여러 장치를 사용할 수 있도록 합니다. 특히 커널 2.6에서는 블루투스 장비도 지원됩니다.

Networking options --->
[ ] Amateur Radio support --->
  -> Amateur radio를 이용해 무선 네트워킹을 하는 기술입니다.

[ ] IrDA (infrared) subsystem support --->
 -> 무선 네트워킹, 장비 연결에 쓰이는 적외선 프로토콜을 지원합니다.

Networking options --->
  -> 네트워크의 여러 가지 논리적 계층을 지원하는 옵션이다. 특히 기존의 IP 계층에서 보안 기능을 제공하는 IPSec, 그리고 네트워크 부하를 배분시키는 IPVS 기능이 리눅스 2.6에 새로 추가되었습니다.

<*> Packet socket
  -> 커널에 구현된 중간 네트워크 프로토콜을 거치지 않고 직접 네트워크 장치와 통신하는데
    사용되며 리눅스가 DHCP 클라이언트로 사용되는 경우도 필요합니다.

[*]   Packet socket: mmapped IO
  -> 이 옵션을 활성화하면 패킷 프로토콜 드라이버는 더 빠른 통신을 지원하는 IO 메커니즘을
    사용할 것입니다.

<*> Unix domain sockets
  -> 유닉스 도메인 소켓 지원에 필요하며 네트워크에 연결되어 있지 않은 경우도 선택함.

[*] TCP/IP networking
  -> 인터넷 연결에 필요한 TCP/IP 프로토콜을 지원합니다.
    인터넷과 이더넷 등 거의 대부분 네트워크에서 사용하는 표준 프로토콜입니다.
    TCP/IP는 하드웨어나 운영체제에 독립적으로 일관성 있는 사용자 서비스를 제공하기 위해
    인터넷 접속뿐만 아니라 다른 많은 프로그램에 꼭 필요하므로 더 말할 것도 없이 반드시
    선택합니다.

[*]   IP: multicasting
  -> 네트워크에 연결된 여러 대의 컴퓨터에 동시에 패킷을 미리 정한 여러 목적지에 보내는
    기능입니다. 적은 대역폭으로 많은 멀티미디어 정보를 전송할 수 있으므로 위성을 통한
    멀티미디어 방송, 화상 교육시스템 등에 유망합니다.

[ ]   IP: advanced router
  -> 리눅스를 주로 라우터로 사용하는 경우 선택합니다.
    리눅스를 라우터로 사용하려면 ip forwarding이 enable 되야 하는데 이는 "/proc
    filesystem support" "Sysctl support" 두 가지를 선택하고 /proc 파일 시스템을 마운트
    한 후 아래 명령을 주어야 합니다.
    echo "1" > /proc/sys/net/ipv4/ip_forward

[*]   IP: kernel level autoconfiguration
  -> 다른 디스크 없는 시스템이 네트워크를 통해 부팅하는 것을 가능하게 합니다.
    클라이언트 시스템이 부팅할때 BOOTP 서버로부터 네트워크 설정 정보를 가져오는
    기능입니다. 디스크가 없이 부트하는 시스템에 쓰이며, "NFS를 통한 루트 파일시스템"
    항목도 [Y]를 선택해야 합니다.

< >   IP: tunneling
  -> 한 프로토콜 안에 다른 프로토콜의 자료를 캡슐화하여 서로 다른 프로토콜 사이에서
    전송하는 기능입니다. 터널은 매우 특이하고 훌륭한 기능을 제공하지만 설정하기에 따라
    아주 끔찍한 일이 벌어질 수 있습니다 .게다가, 터널은 IP 헤더를 복잡하게 만들어서
    패킷마다 20 bytes 정도가 커지므로 기본 MTU인 1500 대신 1480을 써야합니다.
    대부분 이 기능이 필요 없습니다.

[ ]   IP: GRE tunnels over IP
  -> 터널링 목적지에 시스코 라우터가 있을 때 사용합니다.

[*]   IP: ARP daemon support

-> 이 옵션을 선택하면, 커널 내부 ARP 캐쉬가 256 엔트리(entry)이하로 유지됩니다..
일반적으로 커널은 로컬 네트워크에서 IP 주소와 하드웨어 주소 MAP을 내부 캐시로 가지고 있습니다. 수백개 이하의 호스트가 연결된 소규모 네트워크에서는 ARP(Address Resolution Protocol) 캐시를 커널 차원에서 관리하지만, 매우 큰 네트워크(switched network)에서는 커널이 직접 ARP 테이블을 관리하는 방법이 좋지 않습니다. 만일 네트워크 연결(TCP/IP)들이 많다면 커널 메모리 가운데 많은 부분을 ARP 캐시로 사용하기 때문입니다.

[*] IP: TCP syncookie support (disabled per default)
-> DOS(denial-of-service) 공격시 정식 사용자를 보호합니다.

[*] The IPv6 protocol --->
-> 새로운 Internet Protocol 버전에 대한 실험적인 지원입니다.
새 프로토콜은 다음과 같은 특징을 갖습니다. 주소 공간이 128 bit로 늘어나 주소가 고갈되는 일이 없을 것입니다. 라우팅 기능이 확장되어 훨씬 간편하게 자동 주소를 할당할 수 있으며, IP주소에서 호스트 부분으로 MAC 어드레스를 넣으면 IP 충돌을 효과적으로 막을 수 있습니다. 프로토콜 내부에 인증과 비밀 유지 등 보안성이 향상됩니다. 헤더가 간편해지며 더 합리적으로 구성되어 처리속도가 빨라집니다. 그 밖에 중간 연결방법 없이 현재 버전 IP (IP version 4) 프로토콜과 상호 작용이 가능합니다.

[ ] Asynchronous Transfer Mode (ATM)
-> ATM(비동기전송모드)은 WAN(Wide Area Networks)또는 LAN에서 쓰이는 고속 네트워킹 기술입니다. ATM은 기존의 패킷 교환방식과 시분할 다중 장치(TDM)의 장점을 따서 개발한 기술로 가상채널을 통해 데이터와 화상, 음성 트래픽을 일정한 크기의 패킷(53바이츠)에 실어 보냅니다.
ATM을 사용하려면 여러분의 리눅스 박스에 ATM 네트워킹 카드가 필요합니다. 만약 ATM 카드가 있다면 이 곳에서 선택하고 아래에서 맞는 드라이버를 고릅니다. 그리고, 커널 지원 외에도 유저 스페이스의 프로그램들이 필요합니다.

[ ] 802.1d Ethernet Bridging
-> 리눅스박스를 이더넷 브리지로 사용합니다. 일반적으로 브리지보다 효율적인 기능을 가진 라우터가 더 많이 쓰입니다. 요즘은 대부분 라우터 속에 브리지 기능이 들어 있습니다.

[*] 802.1Q VLAN Support
-> 이 기능을 선택하면 여러분의 이더넷 인터페이스에 802.1Q VLAN 인터페이스를 생성할 수 있습니다. 802.1Q VLAN은 방화벽, 브릿징, IP 트래픽 등 일반적인 이더넷 인터페이스가 하는 거의 모든 기능을 지원합니다. VLAN을 이용하려면 VLAN 프로젝트로부터 'vconfig' 툴을 가져와야 합니다.

[ ] DECnet Support
->디지털사에서 만든 많은 제품들이 DECnet 네트워킹 프로토콜을 사용합니다.

[ ] ANSI/IEEE 802.2 LLC type 2 Support
-> 일반적인 이더넷 카드를 이용하는 네트워크에서 X.25 네트워크에 연결하는 802.2 Logical LinkLayer 프로토콜입니다.

[ ] The IPX protocol
-> NoveLL 네트워크에 연결할 때 설정합니다.

[ ] Appletalk protocol support
 -> Apple컴퓨터를 위한 네트워크 프로토콜로 Appletalk를 다룰 수 있는 기회는 거의 없을 것입니다.

[ ] CCITT X.25 Packet Layer (EXPERIMENTAL)
 -> 공중 데이터통신망에서 사용하는 X.25 프로토콜을 사용하려면 필요합니다.
  몇 가지의 싱크보드 에 X.25 지원이 들어 있다.

[ ] LAPB Data Link Driver (EXPERIMENTAL)
  -> Link Access Prodedure for BaLanced는 X.25 프로토콜의 하위 레벨 구성요소입니다.

[ ] Acorn Econet/AUN protocols (EXPERIMENTAL)
  -> Econet은 Arcon 컴퓨터에서 파일, 프린터 서버에 액세스하기 위해 사용되던 아주 오래되고 느린 네트워킹 프로토콜입니다.

[ ] WAN router
  -> 리눅스에 싱크보드(라우터카드)를 꼽아 라우터로 사용하고자 할 때 선택합니다.

[ ] QoS and/or fair queueing --->
  QoS and/or fair queueing
   -> 리얼타임장치 처럼 패킷 도달 순서가 중요한 경우에 기존의 패킷 스케줄링 알고리즘을 사용하지 않고 새로운 알고리즘을 적용하고자 할 때 선택합니다.

Network testing --->
[ ] Packet Generator (USE WITH CAUTION)
  -> 이 모듈은 지정한 비율로, 불특정 패킷들을 생성해서, 지정한 인터페이스로 내보냅니다.
    네트워크 인터페이스 스트레스를 시험하고, 성능을 분석해낼 때 쓸만합니다.

[*] Network packet filtering framework (Netfilter) --->
  -> 넷필터는 이전 커널에서 방화벽, 혹은 매스커레이딩이란 이름으로 불리던 옵션들의 새 이름입니다. 넷필터는 리눅스 박스를 지나가는 네트워크 패킷을 걸러내고(filtering) 조작하기(mangling) 위한 구조(체제)입니다.
    패킷 필터링의 일반적인 용도는 여러분의 리눅스 박스를 방화벽으로 만들어 로컬 네트워크를 인터넷으로부터 보호하는 것입니다. 방화벽으로 쓰일 때 이 기능을 "패킷 필터"라 부르며, 네트워크 패킷을 형태(type), 근원 혹은 출발지(source), 목적지(destination) 등을 기초로 거절/거부할 수 있습니다.

--- Network packet filtering framework (Netfilter)

[*]   Network packet filtering debugging
  -> 넷필터 코드를 디버깅하는데 유용한 정보들을 추가합니다.
[*]   Advanced netfilter configuration
      Core Netfilter Configuration --->
[*]   IP virtual server support --->
      IP: Netfilter Configuration --->

Core Netfilter Configuration

[ ] "MARK" target support
  -> 이 옵션은 라우팅에 앞서 'mangle'테이블 안에 패킷 패킷과 관련된 netfilter
    mark(nfmark) 필드를 바꾸는 규칙들을 만드는 'MARK' 타깃을 추가합니다. 이 기능은
    라우팅 메쏘드를 바꿀 수 있고 다른 서브시스템에 의해 그들을 행태(behavior)를 바꾸도록
    이용될 수도 있습니다.

[*] "conntrack" connection tracking match support
  -> 일반적인 연결추적 매치 모듈로 상태 매치 슈퍼셋(superset)입니다. 더 많은 컨트랙
    정보를 매칭할 수 있으므로 다중 인터넷 링크나 터널에 쓰이는 NAT 게이트웨이처럼 복잡한
    환경에서 유용하게 쓸 수 있습니다.

[*] "helper" match support
  -> conntrack-helper를 이용해 동적인 연결들을 추적할 때 쓰입니다.

[*] "length" match support
  -> 패킷의 길이를 매치할 수 있습니다.

[*] "limit" match support
  -> limit matching은 매치되는 룰에서 속도를 제어합니다.
     LOG 타깃과 (아래에 서 "LOG target support") 서비스 거부 공격(DOS: Denial of
     Service) 회피 기능을 조합할 때 유용합니다.

[*] "mac" address match support
   -> MAC 매칭은 출발지 이더넷 주소에 기반한 패킷 매치를 제공합니다.

[*] Multiple port match support
   -> 멀티포트 매칭은 출발지나 목적지 포트의 시리즈에 기반한 TCP나 UDP 패킷 매치를
     다룹니다.

[*] "pkttype" packet type match support
[*] "quota" match support
  -> 이 매치는 네트워크 쿼터를 지원합니다.

[*] "realm" match support
  -> iptables 안에 라우팅 서브시스템으로부터 realm 키를 이용하여 매칭합니다.

[*] "sctp" protocol match support (EXPERIMENTAL)
[*] "state" match support
  -> 연결 상태 매칭은 추적한 커넥션의 관계에 기반한(예를 들어 이전 패킷들) 패킷 매치를
   다룹니다. 이 옵션은 강력한 패킷 분류 도구입니다.

[*] "statistic" match support
[*] "string" match support
  -> 특정한 문자열이나 캐릭터가 들어있는 패킷을 찾아냅니다.

[*] "tcpmss" match support

-> 네트워크 연결에서 최대 패킷 크기를 제어하는 TCP SYN 패킷의 MSS 값을 검사해서
매치할 수 있습니다

*TCP flags:
-> TCP 연결을 설정하거나 닫기 위해 쓰이는 제어 플래그입니다. 연결을 초기화하기 이한
동기 순차 번호(SYN), SYN에 대한 응답 프레임(ACK), 연결 재설정(RST), 전송을 완료하고
TCP 연결을 닫음(FIN), 긴급 데이터(URG), 그리고 가능한 신속하게 데이터를 전달하라는
PSH 까지 모두 여섯입니다.

[*] "owner" match support
-> 패킷의 소유자 매칭은 패킷을 생성한 사용자, 그룹, 프로세스나 세션에 기반하여
지역적으로-발생된 패킷을 매치하는 것을 다룹니다.

[*] Netfilter connection tracking support
-> 연결 추적(Connection tracking)은 어떤 패킷이 여러분의 머신을 거쳐 갔는지, 그들이
얼마나 연결 되었는지 그 기록을 유지하는 것을 말합니다. 이 옵션은 매스커레이딩 혹은
다른 종류의 네트워크 주소 변환(Fast NAT는 빼고)에 필요합니다. 그리고, 향상된 패킷
필터링에도 쓰입니다. 네트워크 장비를 만든다면 반드시 활성화합니다.

-*- Connection tracking flow accounting
-*- Connection mark tracking support

[ ] PPTP protocol support
-> 이 모듈은 VPN 프로토콜 가운데 하나인 PPTP (Point to Point Tunnelling Protocol,
RFC2637) 패킷을 NAT합니다. 아직 모든 PPTP 모드를 완벽하게 지원하지 못합니다. 더 많은
정보가 필요하면 net/ipv4/netfilter/ip_conntrack_ppt를 읽어보세요.

[ ] H.323 protocol support (EXPERIMENTAL)
-> H.323은 넷미팅 등 원격 화상회의 소프트웨어에서 사용하는 표준 신호전달
프로토콜입니다. ip_conntrack_h323, ip_nat_h323 모듈을 사용하면 conntrack/NAT
방화벽에서도 화상, 음성을 전달할 수 있습니다.

-------- IP: Netfilter Configuration --------
네트워크 케이블을 돌아다니는 패킷들을 검사하고 걸러내는 다양한 기능들이 추가되었습니다.

[*] IP Userspace queueing via NETLINK (OBSOLETE)
-> 넷필터는 유저 스페이스에 패킷을 큐하는 기능을 가집니다
넷링크 장치는 이 드라이버를 이용해서 그들에 엑세스 하도록 사용될 수 있습니다.

[*] IP tables support (required for filtering/masq/NAT)
-> iptable은 일반적이며, 확장할 수 있는 패킷 식별 체제입니다. 패킷 필터링과 full
NAT(masquerading, port forwarding, etc) 서브시스템은 이제 iptalbes를
사용합니다(이전에 사용하던 ipchains가 포팅되었음)

[*] "ah" match support
-> IPSec 패킷에서 AH나 ESP 헤더에 들어있는 SPI를 매치할 수 있습니다.

[*] "ecn" match support
  -> TCP 헤더에서 ECN 필드를 검사합니다. 더 많은 정보가 필요하다면 iptables -m
    ipv4options --help 명령으로 알아보세요.

[*] "ttl" match support
  -> TTL 값으로 매치할 수 있습니다.

[*] Packet filtering
  -> 패킷 필터링은 로컬 input, 포워딩과 로컬 output에서, 심플 패킷 필터링을 위한 연속된
    룰을 담는 '필터' 테이블을 규정합니다.

[*] REJECT target support
  -> REJECT 타깃은 들어오는 패킷에 대해 규칙을 검사해서 아무 소리 없이 버리는 대신
    ICMP에러 메시지로 응답하는 필터링 규칙을 다룹니다.

[*] LOG target support
  -> 이 옵션은 어느 iptables 테이블이든 syslog에 패킷 헤더를 기록하는 규칙을 만드는
    'LOG'타깃을 추가합니다.

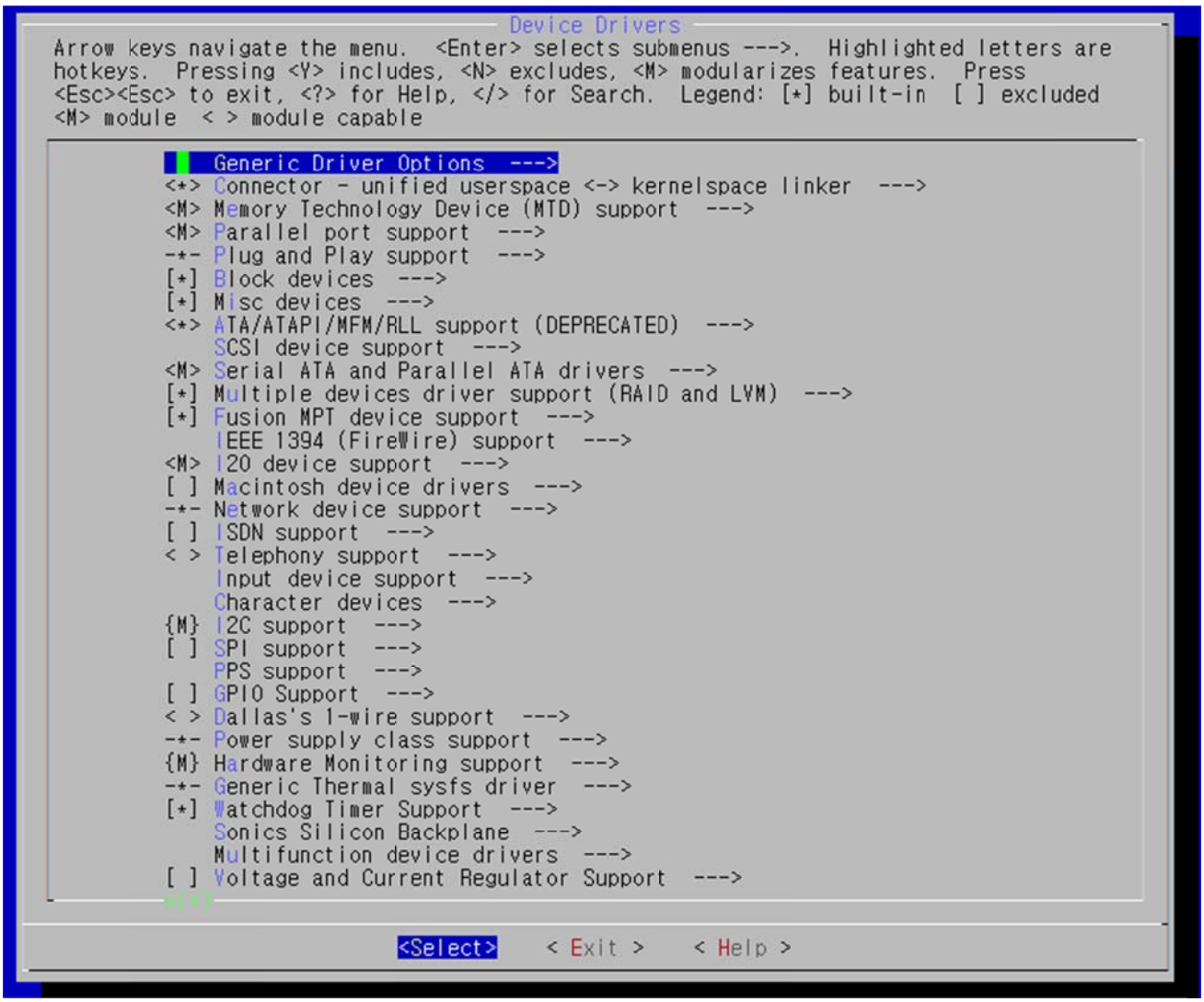
[*] ULOG target support
  -> 넷링크 멀티캐스트 소켓을 사용하는 유저스페이스 로깅 디먼에게 패킷을 넘깁니다.
    LOG 타깃은 syslog만 이용합니다.

[*] Full NAT
  -> Full NAT 옵션은 매스커레이딩, 포트 포워딩, 그리고 다른 형식의 풀 네트워크 주소와
    포트 전환을 다룹니다. 이 기능은 iptables 안에 'nat' 테이블에 의해 제어됩니다.

[*] Packet mangling
  -> 이 옵션은 iptables에 'mangle' 테이블을 추가합니다.
    이 테이블은 패킷을 라우팅할 때 이용할 다양한 패킷 변환에 사용됩니다.

[*] TTL target support
  -> TTL 타깃은 TTL 값을 정하거나 원하는 만큼 늘이고/줄일 수 있습니다.

5.9 Device Drivers



[그림 8-10] Device Drivers

Generic Driver Options --->
-> 펌웨어에 관련한 옵션을 설정합니다.

[*] Memory Technology Device (MTD) support --->
-> 소형 기기에서 쓰이는 여러 가지 저장장치를 지원합니다.
임베디드 시스템에 사용되는 메모리 장치 특히 플래시 장치에 관련된 기능과 모듈들을 선택하는 옵션입니다.

<M> Parallel port support --->
-> 병렬 포트를 지원합니다.
로컬 시스템에서 로컬 프린터를 사용할 경우 이 옵션을 [Y]로 설정하거나 모듈을 [M]으로 설정합니다. 이때 PC-Style hardware 옵션도 모듈로 설정합니다.

-*- Plug and Play support --->
-> 시스템에 부착되면 자동으로 시스템의 리소스를 할당하는 PnP(Plug and Play) 장치를 지원하는 옵션입니다.

[*] Block devices --->
  -> 리눅스에서 쓰이는 각종 블럭 장치에 대한 옵션입니다. loop 장치는 보통 파일을 블럭 장치처럼 쓰려고 할 때 필요하며 losetup이나 mount의 loop 옵션을 사용해서 쓸 수 있습니다. 그리고 RAM을 블럭 장치처럼 쓰려고 할 때 필요한 RAM 디스크 지원도 제공됩니다.
    특히 initrd는 부트로더에서 로딩할 수 있는 RAM 디스크 이미지를 쓰기 위한 옵션입니다.

[*] Block devices --->
[M]   Normal floppy disk support
 -> IBM PC나 그 호환기종에서 사용하는 일반적인 플로피 디스크 드라이브가 있다면 [Y]나 [M]을 선택합니다.

[ ] Compaq SMART2 support
 -> 컴팩의 SMART Array 컨트롤러를 지원하는 드라이버입니다.
   이 드라이버를 지원하는 보드 목록은 다음 명령으로 확인합니다.
   cat /usr/src/linux/Documentation/cpqarray.txt

[*]   Compaq Smart Array 5xxx support
  -> 컴팩의 Smart Array 5xxx 컨트롤러를 지원하는 드라이버입니다.

[*]   Mylex DAC960/DAC1100 PCI RAID Controller support
  -> 블록 디바이스를 병렬로 연결해서 속도나 안정성을 높일 수 있는 PCI RAID 컨트롤러를 지원합니다. 대개는 고가의 서버에서나 볼 수 있는 장비인데, 이 컨트롤러가 없는 장치들은 소프트웨어 RAID를 이용해서 같은 일을 할 수 있습니다. 리눅스 커널에서 지원하는 소프트웨어 RAID를 사용하려면 [N]를 선택합니다.

[*]   Loopback device support
  -> 루프백 디바이스는 파일을 블록 디바이스처럼 사용하는 기능으로 파일속에 파일시스템을 만들어서, 일반적인 블록장치를 다루듯 mount 명령으로 마운트하여 사용할 수 있습니다. 특별한 파일시스템의 실험, CDROM을 굽기 전에 또는 플로피디스크로 옮길 이미지 테스트, 그리고 암호화 등에 유용합니다.

[*]   Network block device support
  -> 네트워크 블럭장치에 대한 옵션입니다. 원격 호스트에 있는 블럭 장치에 연결할 때 쓰입니다.네트워크로 연결된 서버 또는 루프백 서버의 파일시스템을 마운트하여 블록 디바이스(/dev/nd0,...)처럼 사용하는 기능입니다. 클라이언트와 서버는 TCP/IP로 통신합니다. 이 기능이 없더라도 NFS나 Coda를 사용하면 네트워크 파일시스템을 사용할 수 있습니다.

[*]   RAM block device support
  -> RAM을 블럭장치처럼 쓰려고 할 때 필요합니다.

[M]   Packet writing on CD/DVD media
  -> 패킷 쓰기를 지원하는 CD-ROM 등을 사용하기 위해서는 이 옵션을 선택합니다.

<*> ATA/ATAPI/MFM/RLL support (DEPRECATED) --->
  -> 저가의 대용량 저장 장치인 ATA/(E)IDE, ATAPI 장치들을 지원합니다. 일반적이 PC

대부분이 IDE 하드드라이브와 ATAPI CDROM 드라이브를 사용합니다. 여러분의 시스템이 SCSI 인터페이스로만 구성되어 있다면 이 옵션에 [N]을 해도 됩니다.

--- ATA/ATAPI/MFM/RLL support
*** Please see Documentation/ide/ide.txt for help/info on IDE drives ***
[*]   Include IDE/ATAPI CDROM support
  -> IDE/ATAPI CDROM 지원 옵션입니다.
    ATAPI는 SCSI 프로토콜을 흉내낸 IDE CDROM과 TAPE 드라이버의 새로운 프로토콜입니다.

[ ]   Include IDE/ATAPI TAPE support
  -> 서버에서 흔히 사용하는 테이프 백업 장치 입니다.

[*]   IDE Taskfile Access
  -> ioctl로 Taskfile명령을 내릴수 있도록 하는 옵션이다.

[*] CMD640 chipset bugfix/support
  -> CMD640 IDE 칩셋은 많은 486과 Pentium 마더보드에 사용되는 컨트롤러지만 설계상의 문제가 있습니다. [Y]를 선택하면 커널이 몇가지 문제점을 바로잡습니다.
[ ] CMD640 enhanced support
  -> CMD640 IDE 인터페이스를 가지고 있고, 장착된 시스템의 BIOS가 제대로 동작하지 않는다면 [Y]를 선택하십시요.

<M>   PNP EIDE support
*** PCI IDE chipsets support ***
     PCI IDE칩셋 지원입니다. 아래에 있는 여러 칩셋들 중 자신에게 맞는 칩셋을 선택하도록 합니다.

[ ]   Boot off-board chipsets first support (DEPRECATED)
  -> 내장되지 않은 IDE인터페이스에 연결된 장치로 부팅할 수 있도록 하는 옵션입니다.

<*>   Generic PCI IDE Chipset Support
  -> PCI IDE 칩셋 지원 옵션입니다. 이것만 설정을 해두어도 커널에서 IDE 인터페이스는 인식이 가능합니다. 하지만 hdparm옵션이 제대로 적용안되는 경우가 많습니다.

< >   OPTi 82C621 chipset enhanced support (EXPERIMENTAL)
    -> EIDE 컨트롤러 가운데 하나로 현 서버의 마더보드에 이 칩셋이 있다면 선택 합니다.

< >   RZ1000 chipset bugfix/support
  -> PC-Technologies RZ1000 IDE 칩은 많은 486, 펜티엄 보드에서 대개 "Neptune"칩셋과 함께 많이 사용됩니다. 불행히도,이 칩은 많은 상황에서 심각하게 데이터를 손실시키는 경우가 생길 수 있는 설계상의 결점을 가지고 있습니다. 이 옵션을 활성화하면 이런 문제를 자동으로 고치고 점검을 해주는 코드가 커널에 포함됩니다. 이 옵션때문에 디스크 입출력 속도가 조금 떨어질지도 모르지만 100퍼센트 신뢰할 수 있도록 작동할 것입니다.
    SCSI장치만을 가진 시스템이라면 선택 안해도 됨

SCSI device support --->
  -> 여러 SCSI 장치, 고속, 고가의 주번기기 인터페이스인 SCSI를 지원합니다. 자신이 실제 SCSI 장치를 가지고 있지 않더라도 SATA 장치, USB 스토리지, IEEE1394 디스크(SBP2),

그리고 IDE 형식의 CD/DVD 레코더를 사용한다면 scsi의 disk, cdrom 등의 모듈을 선택하는 것이 좋습니다.

<M> RAID Transport Class
-*- SCSI device support
-> SCSI(스커지 혹은 스카시) 하드 디스크나, 테이프 드라이브, 시디롬, 혹은 그 밖의 다른 SCSI 장비를 갖고 있다면 선택합니다.

<M> SCSI disk support
-> 스커시 디스크가 부팅 디스크 라면 모듈로 설정하지 않습니다.
SCSI tape, SCSI CD-ROM은 모듈로 선택해도 좋습니다.

< > SCSI generic support
-> CD-Writer, 스캐너, 신디사이저 등 장치에 쓰입니다. 이 기능은 커널에서 바로 지원하지 않으므로, SCSI 프로토콜을 지원하는 특별한 소프트웨어들이 필요합니다.

[ ] Probe all LUNs on each SCSI device
➔ CD Jukebox와 같이 하나 이상의 LUN(논리장치번호)을 지원하는 SCSI장치가 장치되어 있는데도 단일 LUN만이 인식된다면 여기에서 선택해서 강제로 SCSI 드라이버가 여러 LUN을 검색하도록 할 수 있습니다. 여러 LUN을 지원하는 SCSI장치는 논리적으로 여러 SCSI장치처럼 동작합니다.

[ ] Verbose SCSI error reporting (kernel size +=12K)
⇨ 선택하면 SCSI 하드웨어에 관한 에러 메시지가 자세히 나와서 이해하기가 쉽습니다. 대신 커널 크기는 약 12K 증가합니다.

[ ] SCSI logging facility
➔ SCSI에 관련된 debug에 사용할 수 있는 logging 도구를 활성화합니다.

[ ] Asynchronous SCSI scanning
SCSI Transports ---->

[*] SCSI low-level drivers ---->
=> 현재 시스템이 스카시 디스크나 스카시 컨트롤러를 사용한다면 이곳에서 해당 드라이버를 찾아 선택 해줍니다.

[ ] Adaptec AIC7xxx support (old driver)
⇨ Adaptec의 제품들 가운데 274x, 284x, 294x, 394x, 3985 등 모델을 지원합니다.

[ ] SCSI Device Handlers ---->

--- Serial ATA (prod) and Parallel ATA (experimental) drivers

[*] Multiple devices driver support (RAID and LVM) ---->
-> RAID 장치를 가지고 있는 경우 사용할 수 있는 옵션으로 커널에서 지원하는 소프트웨어 RAID를 사용하고자 하거나 LVM 기능을 사용하기 위해서는 이 옵션을 설정할 수 있습니다. Device mapper는 LVM의 새로운 이름입니다.

[*] RAID-0 (striping) mode
  -> 여러 개의 하드디스크, 여러 개의 파티션, 다중 장치 혹은 루프 디바이스까지 하나의
   볼륨 그룹으로 묶어 주는 역할을 하는 것으로 일종의 가상 디스크의 개념으로 볼룸 그룹
   내에서 가상 파티션이라 할수 있는 논리 볼륨을 생성할 수 있으며, 용량에 따라서 볼륨그룹
   논리 볼륨의 크기를 조절할 수 있습니다.

[*] Fusion MPT device support --->
    IEEE 1394 (FireWire) support --->
  -> 고속 전송을 지원하는 IEEE1394를 지원하는 옵션입니다. IEEE1394 호스트 드라이버 및
    IEEE1394 포트에 부착되는 기기의 종류에 따라서 여러 가지 모듈이 지원됩니다.
    먼저 호스트 드라이버는 대부분 OHCI 호환이므로 만약에 시스템에 IEEE1394 장비를
    연결하려 하면 꼭 커널에 포함시키도록 합니다.

[*] I2O device support --->
  -> I2O을 지원하는 시스템은 호스트의 I/O 작업 부담을 덜고, 네트워크로 연결된 비디오와
    그룹웨어, 클라이언트/서버 처리 등 높은 대역폭 응용 프로그램에서 I/O 성능이 눈에 띄게
    빨라진다고 합니다.

[ ] I2O support
  -> Intelligent Input/Output (I2O) 아키텍쳐는 입출력을 CPU에 맡기지 않고 장치에 있는
    프로세서가 독립적으로 처리해서 네트워크 환경에서 I/O 병목현상을 줄여줍니다.
    결과적으로 전체 네트워크 속도가 향상되며 I2O를 지원하는 서버는 더 많은 사용자에게 더
    다양한 서비스를 원활하게 제공할 수 있습니다. 이 옵션이 제대로 동작하려면 이 카드는
    특별한 I/O 프로세서(IOP)를 탑재한 I2O 인터페이스 어뎁터 카드가 있어야합니다.

[*] Network device support --->
 -> 다양한 네트워크 디바이스에 대한 설정으로 네트워크에 연결되려면 반드시 선택합니다.
    이더넷 디바이스와 PPP(전화선을 이용한 다이얼업 네트워크), SLIP(전화선을 이용한
    다이얼업 네트워크), PLIP(패러랠포트를 이용한 네트워크) 등 네트워크 장치들을
    지원합니다.

--- Network device support
[ ] Dummy net driver support
  -> 네트워크 인터페이스가 있는것 처럼 하는 드라이버로 slip이나 ppp를 쓰려면 옵션을
    켜주어야 합니다.

[ ] Bonding driver support
  -> 여러개의 이더넷 장치를 하나의 이더넷 장치처럼 쓰도록 합니다.
    SISCO에서는 이 것을 "Etherchannel"이라 부르며, Sun에서는 "Trunking", 리눅스에서는
    "Bonding"이라 부릅니다.

[ ] EQL (serial line load balancing) support
  -> 두개의 시리얼 포트 연결을 하나의 연결처럼 쓰도록 하는 드라이버입니다.

[ ] Universal TUN/TAP device driver support
  -> TUN/TAP는 유저 스페이스의 프로그램들을 위한 패킷 수취와 전달을 제공합니다.
    TUN/TAP는 물리적인 매체(ethn 등)로부터 패킷을 받는 대신 유저 스페이스
    프로그램으로부터 패킷을 받으며, 물리적인 매체를 통해 패킷을 보내는 대신 유저

스페이스 프로그램으로 패킷을 보내는 데 쓰이는 간단한 (소프트웨어적인) Point-to-Point나 Ethernet 장치라고 보면 됩니다.

[ ] ARCnet support --->
 -> ARCnet 칩셋이 있는 네트워크 카드를 가졌다면 선택합니다. 아크넷은 전송률이 낮지만(2.5Mbps) 케이블이 훨씬 길어질 수 있어 공장 등에서 사용되기도 합니다.

[*] Ethernet (10 or 100Mbit) --->
 -> 리눅스 박스에이더넷 네트워크 인터페이스 카드 (NIC)가 설치되었다면 반드시 선택합니다.

[*] EISA, VLB, PCI and on board controllers
 -> 요즘 구입할 수 있는 거의 모든 NIC가 PCI 방식입니다. 잘 모르겠으면 선택하는 게 안전합니다. 자신의 이더넷 장비가 PCI를 사용한다면 다음 명령으로 칩셋 등 상세 정보를 확인할 수 있습니다.

[ ] AMD PCnet32 PCI support
 -> 홈랜(혹은 BnA) 형식의 ADSL 서비스에서 사용하기도 합니다. 32비트 버스 매스터링 어댑터로 가장 추천받는 NIC 가운데 하나입니다.

[*] Intel(R) PRO/100+ support
  -> 거의 모든 커널 버전에서 이 옵션이 활성화되어 있는 것을 볼 수 있습니다.
    가장 안정적인 NIC 가운데 하나로 볼수 있습니다.

[ ] RealTek RTL-8129/8130/8139 PCI Fast Ethernet Adapter support
  -> 리얼텍 시리즈는 많은 추천을 받는 대만산 저가형 NIC 가운데 하나입니다.
     모듈로 컴파일하면 이름이 8139too 입니다.

[*] Ethernet (1000 Mbit) --->
  -> 현 시스템에서 사용하는 NIC를 선택 합니다.
[*] Ethernet (10000 Mbit) --->

Wireless LAN --->
 -> radio와 무선 랜을 지원합니다.

[ ] Wan interfaces support --->
 -> 리눅스 박스와 WAN 인터페이스 카드를 이용해 저렴하게 WAN 라우터를 구현할 수 있습니다

[ ] FDDI driver support
 -> HIgh Performance Parallel Interface (HIPPI)는 구리선(25m)이나 광섬유(멀티모드에서 300미터, 싱글모드에서 10킬로미터)를 매체로 800Mbit/sec과 1600Mbit/sec의 높은 속도를 내는 네트워크입니다. 일반적으로 클러스터와 슈퍼컴퓨터 연결에 쓰이는데, 이 기능을 사용하려면 HIPPI 네트워크 카드가 있어야 합니다.

[ ] PPP (point-to-point protocol) support
 -> PPP는 SLIP보다 향상된 기능을 가진 시리얼 라인(전화선 등) 네트워킹 프로토콜입니다.

[ ] SLIP (serial line) support
 -> PPP 이전에 많이 사용하던 시리얼 라인 네트워크 프로토콜입니다.

[ ] Fibre Channel driver support
 -> Fibre Channel은 주로 대용량 저장 장치에 쓰이는 고속 시리얼 프로토콜입니다. 이
   프로토콜을 SCSI와 호환되며 대체할 수 있습니다.
[ ] Traffic Shaper (OBSOLETE)
 -> Traffic Shaper는 어떤 네트워크 장치를 지나서 나가는 데이터 흐름 속도를 제한할 수
   있는 가상의 네트워크 장치입니다

[ ] ISDN support --->
  -> ISDN은 전화선을 이용한 디지틀 종합 서비스입니다. 일반적으로 B채널 두 개를
   사용하는데(2B 방식) 한 채널이 64KB의 전송속도를 가집니다. ISDN을 사용하기 위해서는
   특별한 단말 장치를 구비해야 합니다. ISDN을 사용하지 않는다면 선택 하지 않습니다.

[ ] Telephony support --->
    Input device support --->
    -> 커널 2.6의 Input Subsystem은 완전히 새로 작성되었다고 보면 되겠습니다.
      키보드 드라이버는 더 이상 콘솔 드라이버에 포함되지 않으며 완전히 분리되었으며 현재
      시스템에 등록된 입력 장치에 대한 정보를 보려면 /proc/bus/input에 있는 파일들을
      참조하면 됩니다.
      리눅스의 마우스 인터페이스에서 모든 마우스는 기본적으로 /dev/input/mice라는 장치로
      접근이 가능하게 되고 각각의 마우스는 /dev/input/mouseX라는 장치로 접근이 가능하게
      되었습니다. 하지만 이전의 리눅스 커널과의 호환성을 위해 /dev/psaux 장치를 만들도록
      하는 옵션도 포함되었습니다.

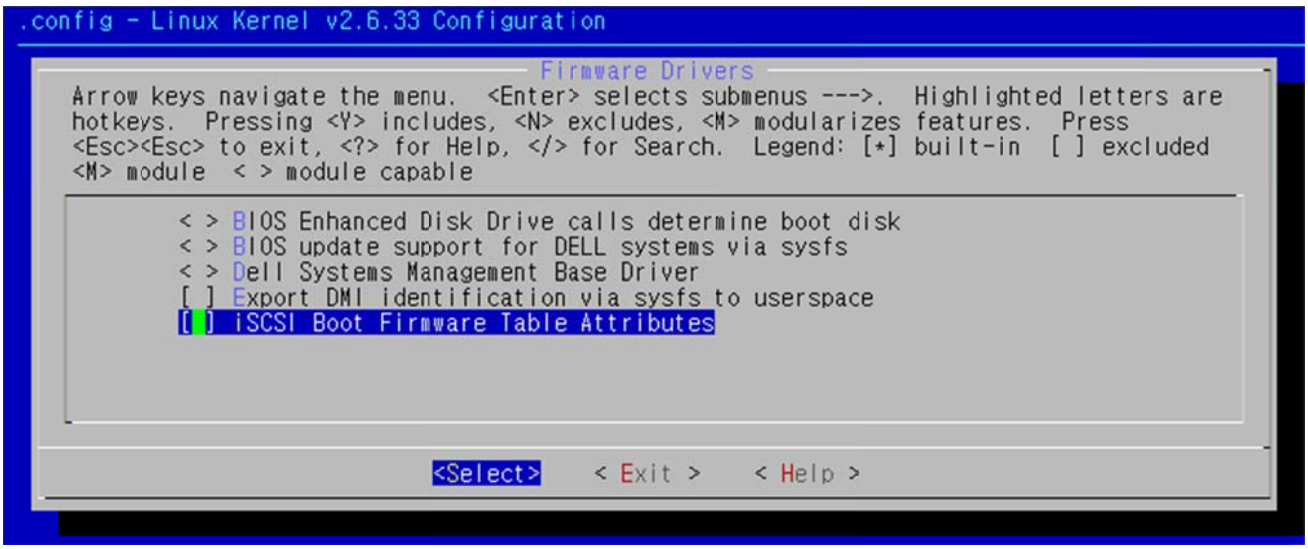
    Character devices --->
    -> 리눅스에서 여러 문자 장치를 쓸 수 있도록 하는 옵션으로 가상 터미널은 리눅스
      콘솔에서 쓰이는 유용한 기능으로 하나의 콘솔을 여러 개의 터미널처럼 쓸 수 있도록
      합니다. 그리고 여러 가지 직렬 포트 장비와 병렬 포트 장비를 지원하며, 그래픽카드를
      사용할 수 없는 환경에서도 쓸 수 있도록 직렬 및 병렬 포트로 콘솔 입출력을
      지원하도록 하는 옵션이 포함되어 있습니다

[ ] I2C support --->
 -> I2C(발음:I-square-C)는 필립스에서 개발한 느린 시리얼 버스 프로토콜로 많은 마이크로
   컨트롤러 어플리케이션에서 사용됩니다.

[ ] Sound card support --->
  -> 커널 2.6에서는 두 가지 사운드 서브 시스템이 제공되는데 한 가지는 이전부터 쓰이던
      OSS(Open Sound System)이고 다른 한 가지는 새로 작성된 ALSA(Advanced Linux Sound
      Architecture입니다.

[*] USB support --->
 -> 리눅스에서 USB 장치를 쓸 수 있도록 하는 옵션으로 리눅스 2.6에서는 480Mbps의 대역폭을
   제공하는 USB 2.0 호스트 컨트롤러의 사용이 가능하며, 2.4와 마찬가지로 USB 1.0용 호스트
   컨트롤러인 UHCI, OHCI 드라이버가 제공됩니다
   USB를 사용하려면 이 옵션을 선택하고 아래에서 UHCI support, OHCI support 가운데 하나를
   선택해야 합니다.

5.10 Firmware Drivers

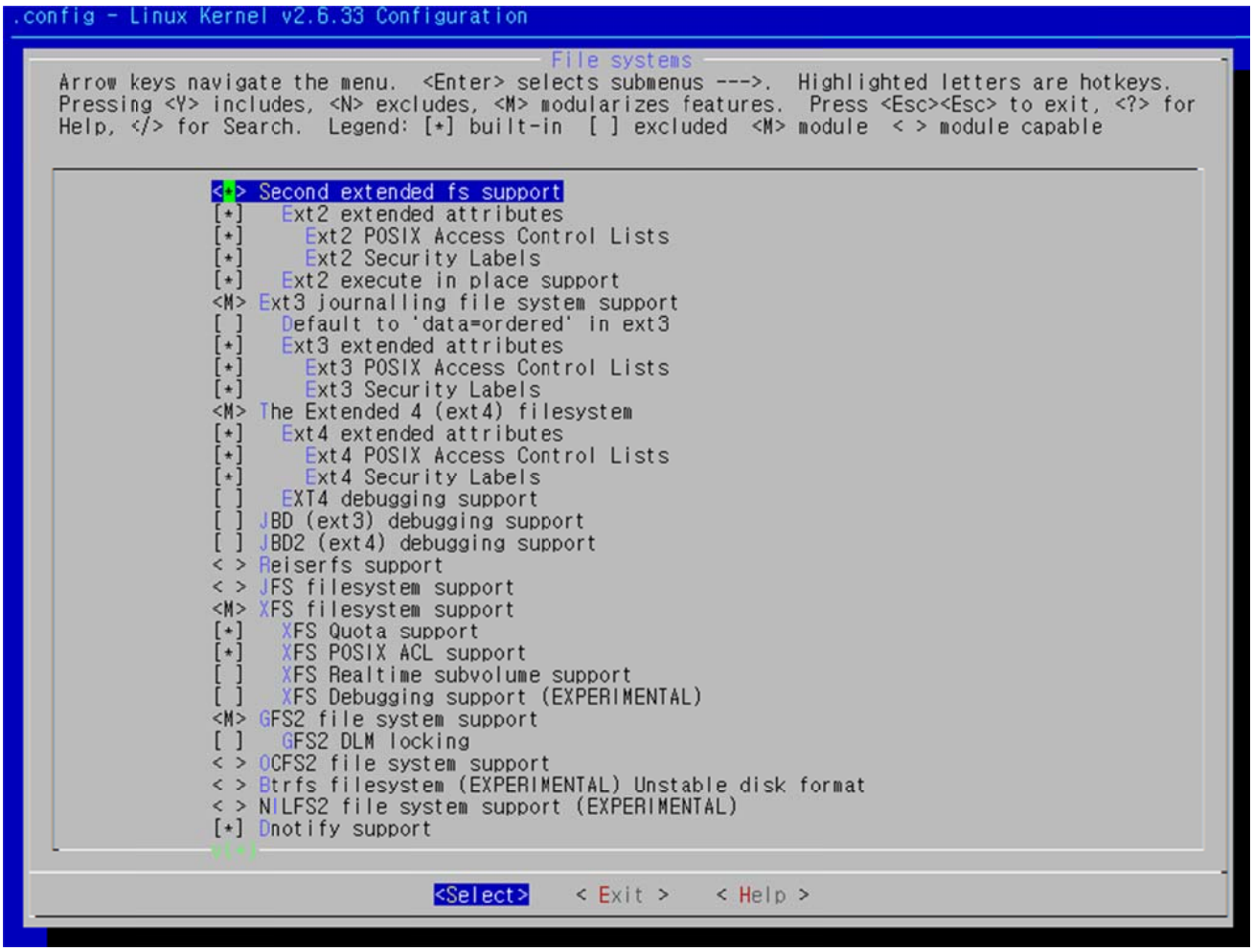


[그림 8-11] Firmware Drivers

Firmware Drivers ---->

➔ 방화벽에 관한 설정을 합니다.

5.11 File systems



[그림 8-12] File systems

File systems --->

리눅스에서 일반적으로 많이 쓰이는 파일 시스템에 대한 설정을 할 수 있습니다.

<*> Second extended fs support

리눅스에서 현재 사용하는 표준 파일시스템입니다. 반드시 활성화합니다.

[*] Ext3 journalling file system support

-> 리눅스에서 하드 디스크에 구성하는 표준 파일시스템 (저장장치에 파일들을 구조화하는 방법)인 ext2 파일시스템의 저널링 버전입니다.(흔히 ext3라 부릅니다) 저널링 코드가 들어있다면 파일시스템이 망가지더라도 e2fsck(파일시스템 점검도구)를 실행할 필요가 없습니다. 저널은 여러분의 시스템이 망가지는 그 순간까지 모든 변화를 추적하고 있다가 점검시간을 거치지 않고도 파일시스템을 바로잡을 수 있습니다. 파일 시스템이 제대로 마운트되었거나 e2fsck 유틸리티가 파일 시스템에서 실행되는 한 ext3 드라이버와 ext2 드라이버를 이용해서 자유롭게 둘을 바꿀 수 있습니다.

[*] Ext3 extended attributes
[*] Ext3 POSIX Access Control Lists
[ ] Ext3 Security Labels
[ ] The Extended 4 (ext4) filesystem
[*] JBD (ext3) debugging support

[*] Reiserfs support
  -> 저널링(journaling) 파일시스템을 지원합니다. 저널링 파일 시스템에서는 인덱스가 갱신되기 전에 관련 내용이 기록되므로 정전이나 다른 이유로 인덱스에 문제가 생기더라도 다시 시스템을 재가동하면 운영체제가 로그를 보고 복구를 할 수 있습니다. 일반적인 경우 ReiserFS는 ext2 정도의 속도를 내지만, 큰 디렉토리에 작은 파일들이 많은 경우 매우 효율적입니다.

[ ] Enable reiserfs debug mode
[ ] Stats in /proc/fs/reiserfs
[ ] ReiserFS extended attributes

[*] JFS filesystem support
  -> IBM사의 JFS 파일시스템을 쓰는 환경에서 이 옵션을 선택합니다.

[ ] XFS filesystem support
  -> SGI의 XFS 저널링 파일시스템을 사용하는 환경에서 이 옵션을 선택합니다.

[*] Quota support
  -> ext2 파일시스템에서 유저/유저그룹이 사용할 수 있는 디스크 공간의 크기를 제한하는 기능입니다.

[ ] Kernel automounter support
< > Kernel automounter support
  -> NFS와 같이 원격 파일시스템을 자동으로 마운트 될도록 하는 옵션으로, 이 옵션을 선택하려면 NFSfile System support 옵션도 선택해야 합니다

[ ] Kernel automounter version 4 support (also supports v3)
  -> 원격 파일시스템을 자동으로 마운트 될도록 하는 옵션으로, NFS 4 버전으로 3 버전 이하이므로 이 옵션을 선택하면 Kernel automounter support를 선택하지 않아도 됩니다. 만약 여러분의 시스템이 큰 분산 네트워크에 연결되어 있지 않거나, 동적으로 재설정이 필요한 랩탑의 가운데 하나가 아니라면 아마 automounter가 필요 없을 것입니다.

CD-ROM/DVD Filesystems --->
  -> CD-ROM/DVD에서 쓰이는 파일 시스템들을 포함하는 옵션입니다. iso9660 파일 시스템 지원은 CD/DVD 미디어에서 많이 쓰이는 파일 시스템으로 여기에 MS가 만든 긴 파일 이름과 유니코드를 지원하는 iso9660의 확장인 Joliet, 그리고 리눅스에서만 쓸 수 있는 압축 isofs 지원 기능을 포함시킬 수 있습니다.

DOS/FAT/NT Filesystems --->
  -> 윈도우 계열 운영체제에서 사용되는 여러 가지 파일 시스템을 리눅스에서 쓸 수 있도록 하는 옵션 입니다. 이전 8.3 형식의 파일 이름을 지원하는 FAT 지원, 긴 이름 파일이 지원되는 VFAT 또는 FAT32 파일 시스템을 지원하는 옵션, 그리고 윈도우 2000 이상에서 지원되는 NTFS 지원이 포함되어 있습니다. 특히 커널 2.6에서는 NTFS-NG가 포함되었고 여기서는 이전 NTFS에서 지원하지 못하던 몇몇 속성을 사용할 수 있습니다.

Pseudo filesystems --->
  -> 리눅스에서 쓰이는 여러 가상 파일 시스템을 쓸 수 있는 옵션입니다. proc 가상 파일 시스템은 특히 각종 시스템 도구에서 쓰이기 때문에 커널에 포함시키는 것이 좋습니다.

그외 동적으로 영역을 할당해주는 램 파일 시스템인 가상 메모리 파일 시스템은 /dev/shm 또는 /tmp에 주로 마운트하여 많이 사용됩니다.

[*] Virtual memory file system support (former shm fs)
-> Tmpfs는 모든 파일들을 가상의 메모리에 보관하는 파일시스템입니다. 어떤 파일들도 하드드라이브에 생성되지 않는다는 면에서 모든 것이 임시적이라고 할 수 있습니다.

[*] Miscellaneous filesystems --->
-> 여러 가지 다른 OS 에서 사용되는 파일 시스템 지원 옵션입니다. 리스트중 해당 파일 시스템이 존재한다면 해당 파일 시스템에 대한 지원을 켜둡니다.

[ ] ADFS file system support (EXPERIMENTAL)
Advanced Disk File System은 Acorn 시스템의 플로피와 하드 디스크에서 사용되는 파일 시스템입니다.

[ ] Amiga FFS file system support (EXPERIMENTAL)
- The Fast File System (FFS)은 AmigaOS Version 1.3 (34.20)이후로 Amiga(tm) systems에서 하드디스크에서 사용하는 일반적인 파일시스템(filesystem)입니다.

[ ] Apple Macintosh file system support (EXPERIMENTAL)
- 매킨토시 형식으로 포맷된 플로피 디스크와 하드 드라이브 파티션을 읽고 쓸 수 있습니다.

[ ] EFS file system support (read only) (EXPERIMENTAL)
- EFS는 SGI사의 IRIX OS에서 CDROM용 파일시스템과 [초기버전의] 파일시스템에서 사용하는 파일 시스템입니다.

[ ] Compressed ROM file system support (cramfs)
- 롬 기반 임베디드 시스템에서 압축된 파일시스템을 사용한다면 Cramfs 옵션을 활성화합니다. 읽기 전용이며 256MB로 크기가 제한되어 있습니다.

[ ] FreeVxFS file system support (VERITAS VxFS(TM) compatible)
- FreeVxFS는 VERITAS VxFS(TM) 파일시스템 형식을 지원하는 파일시스템 드라이버입니다.

[ ] QNX4 file system support (read only)
- QNX4운영체제에서 사용되는 파일시스템입니다.

[ ] System V/Xenix/V7/Coherent file system support
- Xenix와 Cherent는 인텔 기종을 위한 상용 유닉스 시스템입니다.

[ ] UFS file system support (read only)
- BSD와 Unix에서 파생된 버전(SunOS,FreeBSD, NetBSD, OpenBSD, NextStep)들은 UFS라는 파일시스템을 사용합니다.

[*] Network File Systems --->
-> NFS서버와 클라이언트 지원과 삼바 파일시스템 지원을 위한 옵션이다.NFS 와 삼바를 통해서 자료를 공유하는 환경에서는 이 옵션이 필요하지만, 그렇지 않는 경우에는 불필요한 옵션이므로 모듈로 선택하거나 선택하지 않음

삼바 마운트시 한글 지원이 되도록 하려면 Use a default NLS(Native Language support)를 체크한 후 DefaultRemote NLS Option 에서 값을 cp949로 설정합니다.

[*] NFS client support
-> 네트워크 파일시스템 클라이언트입니다. 만약에 SLIP이나 PLVIP, PPP, 이더넷등으로 다른 유닉스 컴퓨터에 물려있고 그 컴퓨터를 마운트해서 그 쪽 파일을 엑세스하고 싶다면 선택합니다. 상대편 컴퓨터는 NFS 서버, 여러분의 리눅스 박스는 클라이언트가 되는데 서버에는 nfsd, mountd, portmap 등이 떠 있어야 하며, /etc/export 파일에서 여러분을 허용해야 합니다. "파일을 마운트한다"는 말은 클라이언트가 보통 유닉스 명령어로 서버쪽 파일들을 자기 하드에 있는것처럼 접근할 수 있다는 의미입니다

[*] NFS server support
-> 커널 기반의 NFS 서비스입니다. 유저 스페이스의 nfsd보다 더 빠르지만 아직 불안합니다.

[ ] SMB file system support (OBSOLETE, please use CIFS)
- 리눅스 박스에서 m$ windogS 9x/NT 네트워크 자원을 공유하는 기능입니다. 대부분의 네트워크 클라이언트가 MS 윈도우즈 박스이므로 SMB 옵션을 활성화할 것을 권장합니다.

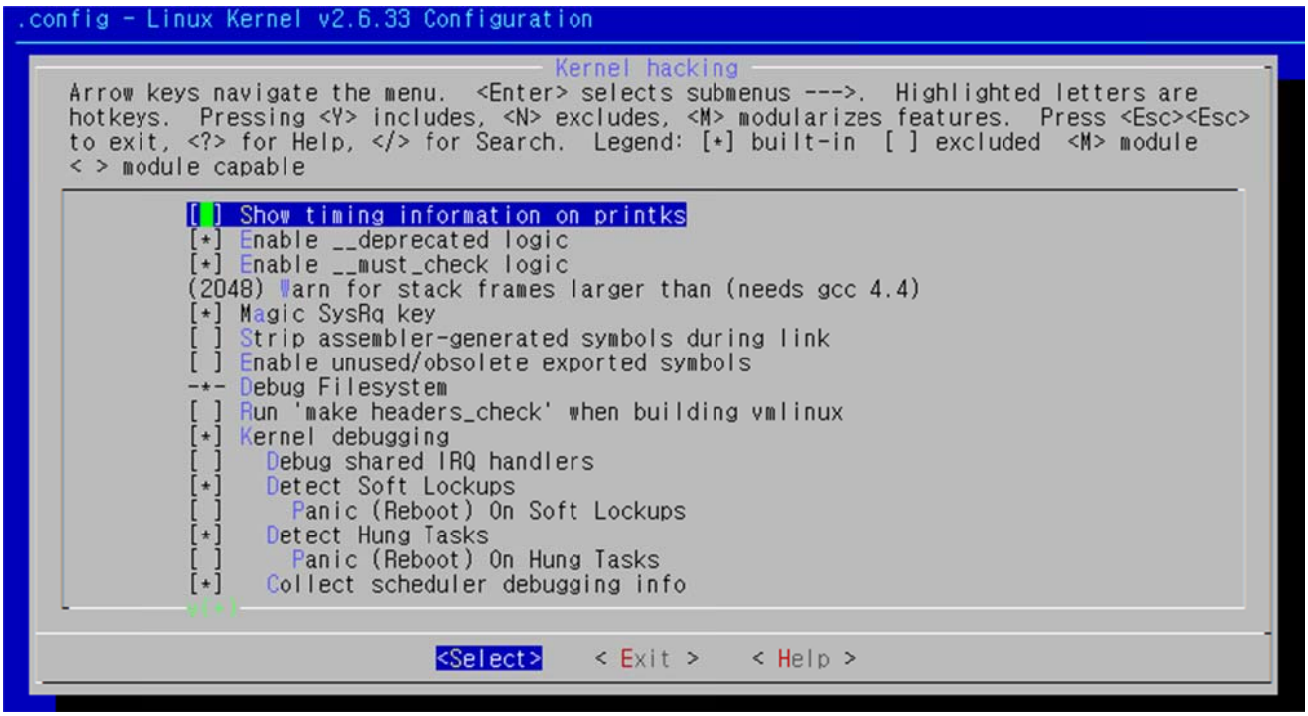
[ ] NCP file system support (to mount NetWare volumes)
- NCP(netware core protocoL)은 IPX 를 이용한 랜 프로토콜입니다. 노벨 네트웨어 클라이언트가 NCP를 통해 파일서버 볼륨을 마운트하여 사용합니다.

[ ] Coda file system support (advanced network fs)
- Coda는 NFS와 비슷하지만 더 진보된 네트워크 파일시스템입니다. Coda는 비접속 운영, 캐시, 보안과 인증 등 NFS보다 좋은 점에 몇 가지 있습니다. 서버와 클라이언트 모두가 지원해야 합니다. Coda 서버들은 사용자 공간의 프로그램들이며 커널이 지원해야 하는 것은 아닙니다.

Partition Types --->
- 매킨토시를 제외한 유닉스 파티션은 "UFS fileSyStem Support"와 함께 설정합니다.

-*- Native language support --->
-> MS의 fat 파일시스템족은 고유언어 문자셋으로 파일이름을 다룰 수 있습니다. 이런 문자셋은 DOS 코드페이지에 저장되어 있어 mS DOS/WindowS 파티션의 파일이름을 정확하게 읽으려면 필요합니다.

5.12 Kernel hacking



[그림 8-13] Kernel hacking

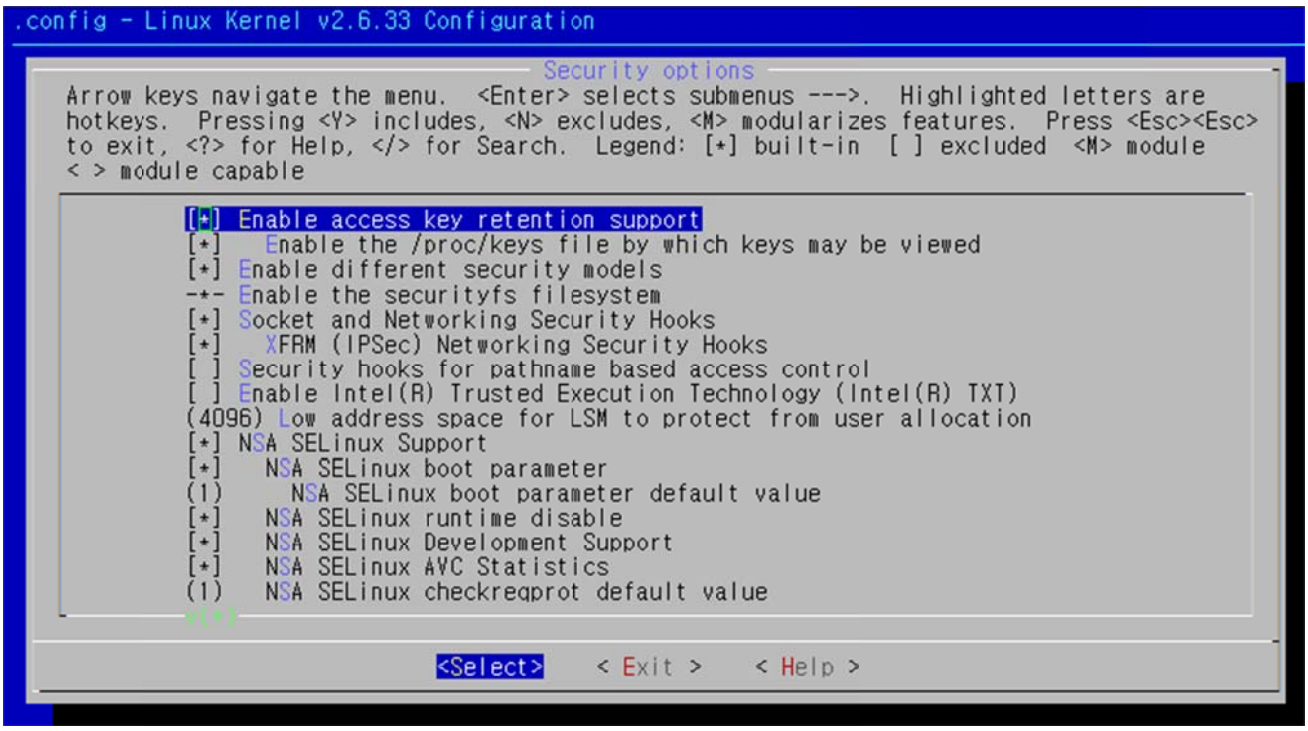
Kernel hacking --->
-> 이 옵션은 커널 디버깅 중에 시스템이 다운되었을 경우 Magic 키를 사용하여 시스템을 제어할 수 있도록 해 준다.이 옵션은 가능한 선택하는 것이 좋습니다.

[*] Magic SysRq key

-> Magic SySRq key 를 선택할 경우 시스템이 심각한 문제가(crashes) 있 어도 통제할 수 있습니다. 예를 들면 버퍼 캐시를 디스크로 옮기고, 시스템을 리부트하거나 상태 정보를 표시합니다.

이 기능은"<alt>+<SysRQ>"를 누른 채 k, r, s 등 <command key>를 눌러 사용합니다. SysRQ 키가 없는 키보드는 PrtSc 키를 누르면 됩니다. Documentation/sysrq.txt 를 참고 바랍니다.

5.13 Security options



[그림 8-14] Security options

Security options ---->

- 시스템의 보안 관련 옵션입니다.

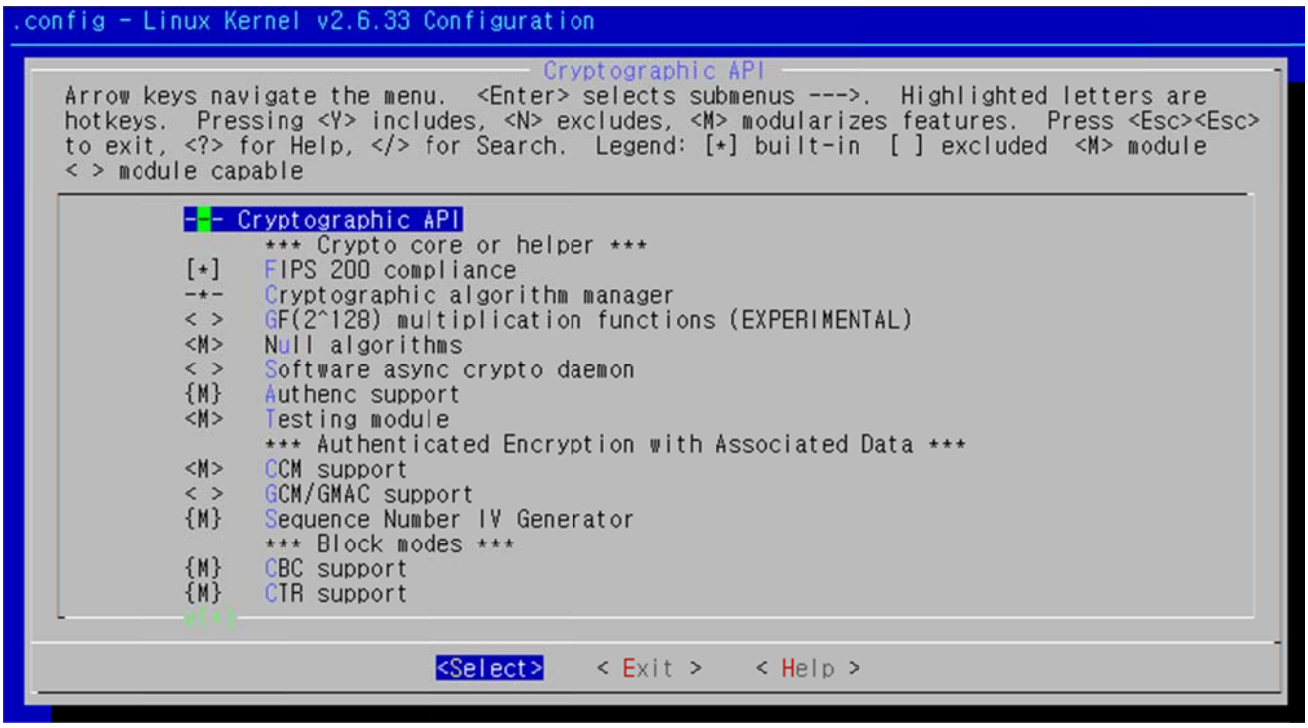
[ ] Enable access key retention support

- 커널에서 인증 토큰과 액세스키를 가질 수 있도록 설정하는 옵션

[ ] Enable different security models

- 커널에서 다른 보안 모델을 적용되도록 할 때 선택,선택되지 않으면 기본 보안 모델이 적용됩니다.

5.14 Cryptographic API



[그림 8-15] Cryptographic API

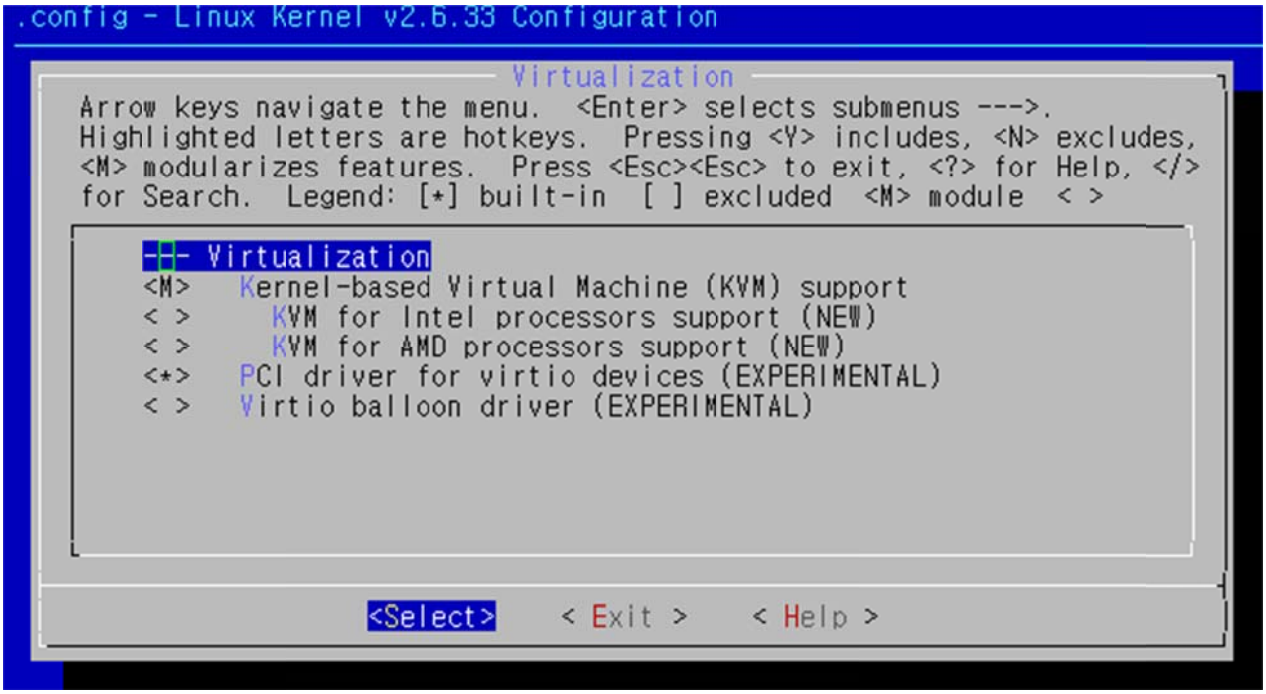
Cryptographic options --->

- 암호화 및 인증에 관련된 알고리즘 모듈을 선택하니다. 모든 암호화 알고리즘 모듈을 이해하기 어려우므로 모두 모듈로 선택합니다.

[M] Cryptographic API

=> 암호화 API 관련 설정입니다. 각종 암호 알고리즘과 모듈들을 설정합니다.

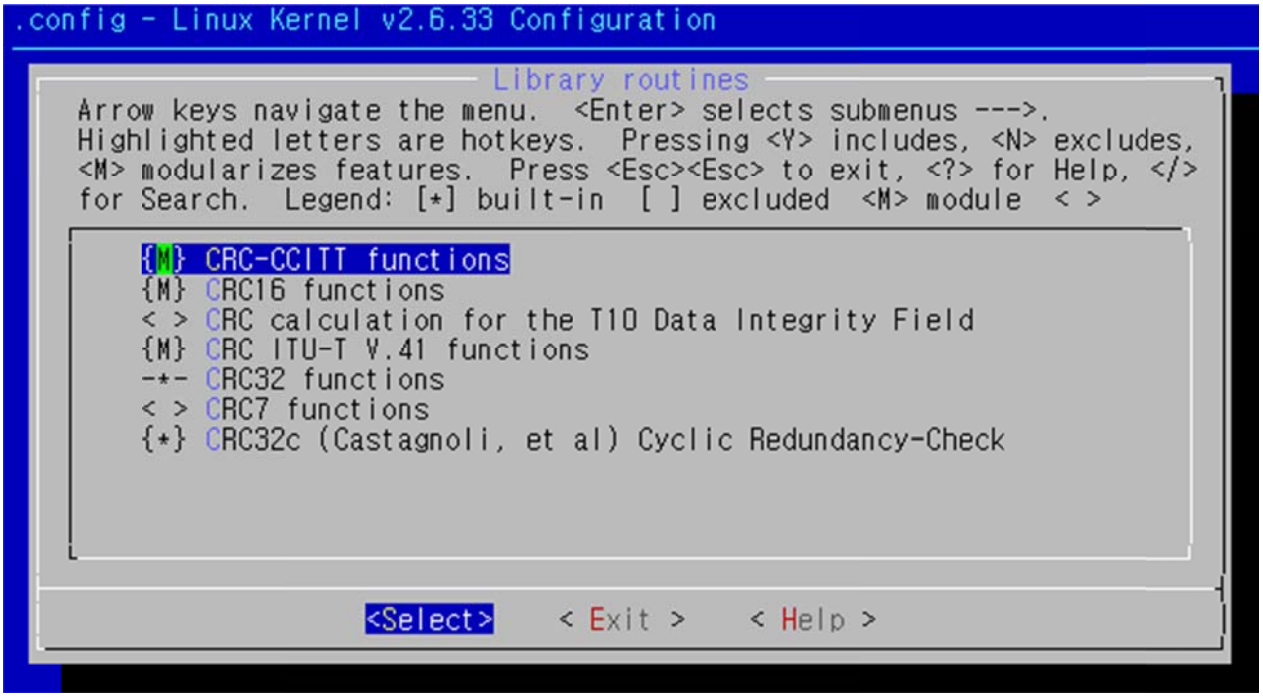
5.15 Virtualization



[그림 8-16] Virtualization

--- Virtualization
< > Kernel-based Virtual Machine (KVM) support
KVM은 리눅스 게스트 운영체제 가상화를 지원하며, 가상화를 지원하는 하드웨어에서 윈도우까지도 돌릴 수 있습니다

5.16 Library routines



[그림 8-16] Library routines

Library routines --->
⇨ CRC관련 설정을 합니다. CRC7, CRC16, CRC32의 함수와 그 모듈을 설정합니다.

[ ] CRC-CCITT functions
➢ 외부 커널 모듈이 CRC-CCITT 라이브러리 함수를 사용할수 있도록 하는 옵션으로 모듈로 설치 합니다.

[ ] CRC32c (Castagnoli, et al) Cyclic Redundancy-Check
➢ 외부 커널 모듈이 CRC32c 라이브러리 함수를 사용할수 있도록 하는 옵션으로 모듈로 선택합니다.

Load an Alternate Configuration File
⇨ 이전 저장한 설정값을 불러들여 컴파일 할수 있습니다.

Save an Alternate Configuration File
⇨ 지금까지 설정한 값을 저장합니다. (.config 파일로 저장됨)

6. 부트 로더 설정 하기

커널 컴파일 작업이 끝난후에는 새 커널을 서버에 적용하기 위해서 부트 로더에 새로운 커널을 적용해야 합니다. 리눅스 시스템에는 GRUB 나 LILO 중 하나가 설치되어 있을 것입니다. 이 중 시스템에서 사용되는 부트로더로 새로 설치된 커널을 부팅하도록 설정합니다. 참고로 RedHat Linux 9 버전에서 부터는 Grub 가 디폴트 부트로더로 되어 있지만 기존 lilo에 익숙한 사용자들은 lilo를 적용하여 사용 하기도 합니다.

컴파일이 끝난후에도 항상 부트로더가 새 커널로 부팅하도록 제대로 설정되었는지 확인해 보는 것이 좋습니다. 이것은 매우 중요한 것으로 만일 부트로더가 정확히 설정되지 않는다면, 부팅도중에 에러 메시지를 보이며 Linux 시스템을 부팅할 수 없게 됩니다.
컴파일된 새 커널에 문제가 있어서 부팅이 안될 경우 기존 커널 버전으로 부팅을 해야 합니다. 그러므로 기존 커널의 파일은 삭제 하지 마시기 바랍니다.

6.1 GRUB 설정 하기

/boot/grub/grub.conf 파일에 새 커널 정보가 추가 되어 있는지 확인 합니다.
기존에 있는 커널 정보는 그대로 두고 새 커널 정보만 추가 합니다.

root@local:/

```
[root@localhost /]# # cat /boot/grub/grub.conf
# grub.conf generated by anaconda
#
# Note that you do not have to rerun grub after making changes to this file
# NOTICE:  You do not have a /boot partition.  This means that
#          all kernel and initrd paths are relative to /, eg.
#          root (hd0,0)
#          kernel /boot/vmlinuz-version ro root=/dev/sda1
#          initrd /boot/initrd-version.img
#boot=/dev/sda
default=0
timeout=5
splashimage=(hd0,0)/boot/grub/splash.xpm.gz
hiddenmenu
title CentOS (2.6.33)
        root (hd0,0)
        kernel /boot/vmlinuz-2.6.33 ro root=LABEL=/
        initrd /boot/initrd-2.6.33.img
title CentOS (2.6.18-164.15.1.el5)
        root (hd0,0)
        kernel /boot/vmlinuz-2.6.18-164.15.1.el5 ro root=LABEL=/
        initrd /boot/initrd-2.6.18-164.15.1.el5.img
title CentOS (2.6.18-164.el5)
        root (hd0,0)
```

6.2 LILO 설정 하기

새 커널 정보가 /etc/lilo.conf 파일에 추가 되어 있는지 합니다.
추가 안되어 있을 경우 lilo.conf 파일에 추가 해주어야 합니다.
기존의 커널 이미지 정보는 삭제 하시지 말고 기존 정보 위나 밑에 추가를 합니다.
부팅시 default= 에 있는 label이름의 커널로 부팅이 이루어 지게 되므로 새커널 정보를 추가시에는 기존 커널 정보의 label = 이름을 다른 이름으로 변경주어야 합니다.
새 컬널로 부팅시 문제가 있어 부팅이 안될 경우 기존 커널로 부팅을 할 수가 있습니다.

```
root@local:/
[root@localhost /]# cat /etc/lilo.conf
prompt
timeout=50
default=linux
boot=/dev/sda
map=/boot/map
install=/boot/boot.b
message=/boot/message
lba32

image=/boot/vmlinuz-2.6.33
        label=linux
        initrd=/boot/initrd-2.6.33.img
        read-only
        append="root=LABEL=/"

image=/boot/vmlinuz- 2.6.18-164.15.1.el5
        label=linux-old
        initrd=/boot/initrd- 2.6.18-164.15.1.el5.img
```

/etc/lilo.conf 파일을 수정후에는 꼭 /sbin/lilo 명령을 실행하여 부트로더 정보가 갱신되도록 합니다.
참고로 lilo.conf 파일에 문제가가 있을 경우 에는 /sbin/lilo 명령 실행시 에러 메시지가 나타 납니다.

※ 커널 컴파일에 걸린 시간 알아보기
time 명령을 사용하면 작업에 소요된 시간을 측정할 수 있습니다. 예를 들어 의존성 설정부터 모듈 설치작업까지 모두 얼마나 시간이 걸리는지 알아보려면 다음 명령을 사용합니다:

```
root@local:/
[root@localhost /]# # time -v sh -c 'make dep && make clean && bzImage && modules ₩
&& modules install'
```

time -v : 자세한 정보를 출력합니다.
sh -c 'string..' : 명령을 뒤따르는 문자열들에서 읽어들입니다.

Chapter 9. 보안 설정

1. 시스템 기본적인 보안설정

리눅스 배포판 CD로 OS 설치를 할 경우 사용하지 않는 불필요한 패키지까지 설치 할 필요는 없습니다.
여러 패키지들이 많이 설치 되었을 경우 환경 설정이 제대로 설정(configuration)되지 않았거나, 관리자가 잘 알지 못하는 프로그램의 경우 보안상 취약점이 증가할 가능성이 높기 때문입니다
이는 특히 NFS, Portmap, Samba 등 네트워킹에 기반한 프로그램일 경우 보안쪽에 신경을 써 인스톨 하는 것이 좋습니다.
리눅스용 패키지는 설치가 쉬운 편이므로 초기에는 필요한 최소한의 패키지만 설치 하신 후 필요 패키지가 발생시 그때 별도로 설치 하시면 더욱 좋을 것입니다.

1.1 리눅스 파티션 생성 하기

초기 시스템을 설치시에 시스템의 보안을 고려 하여 파티션을 나누는 것이 좋습니다.
루트 파티션을 같이 사용 할 경우 문제 될수 있는 파티션들을 알아 봅니다.

/
리눅스의 루트 디렉토리로 최상의 디렉토리 입니다.
이 루트 디렉토리를 기준으로 모든 파일 및 디렉토리가 위치 합니다.

/usr
시스템 명령어와 사용자들의 각종 환경 설정과 여러 프로그램들이 위치하는 디렉토리입니다.

/var
로그 폴더와 메일 관련 폴더등이 사용되는 폴더로 시스템 사용량이 많거나 스팸메일이 많은 경우 크기가 매우 빨리 늘어나 시스템 파티션에 장애를 줄 수도 있으므로 별도의 파티션으로 작업합니다.

/tmp
폴더는 임시폴더로 웹응용프로그램의 취약성을 이용하여 /tmp 폴더에 스크립트를 올려 실행 시키 경우가 있는데 이를 방지 하기 위해서는 /tmp 파티션을 별도로 만들어 /tmp 폴더에는 실행 권한을 주지 않고 마운트 되도록 합니다.

/etc/fstab 에서는 주로 아래와 같은 옵션을 주어 사용합니다.

```
LABEL=/tmp          /tmp              ext3     defaults,noexec,nosuid          1 2
```

fstab 옵션 설명
* noexec : 해당 파티션에서 실행파일의 실행이 허용되지 않음
* nosuid : 해당파티션에서 setuid설정을 허용하지 않는다
* nodev : 해당파티션에서 문자나 특정 디바이스 장치를 허용하지 않음
* /dev/shm : 공유메모리 디바이스로 posix기반의 공유메모리를 사용하는 소스에서 사용함

/home
일반 사용자들의 계정 폴더가 위치하는 곳으로 도메인의 홈 데이터가 이곳에 위치합니다.

1.2 리눅스 설치후 불필요한 서비스 제거 하기

리눅스를 설치후 netstat 명령으로 포트를 확인 해 보면 여러 개의 포트들이 LISTEN 되어 있는 것을 확인 할수 있습니다.
현재 시스템에서 사용하지 않은 불필요한 패키지들은 닫아 주고 사용하는 포트만 남겨 두는 것이 보안상 취약부분을 줄여 줍니다.

포트 확인 은 netstat -an 명령등로 확인 할수 있습니다.

시스템내 현재 활성화된 데몬(패키지) 리스트는 아래 같이 ntsysv 명령이나 chkconfig 명령으로 할수 있습니다.

ntsysv 예)

[그림 9-1] ntsysv

ntsysv 라고 실행을 하면 서버 시작시 실행하는 서비스 데몬들의 리스트 박스가 나타나는데 여기서 서비스가 필요치 않는 항목은 체크를 해제한 후 시스템을 리부팅 하면 적용됩니다.

주로 많이 사용하는 데몬은 아래와 비슷하므로 이외의 데몬은 체크를 해제 한후 사용하는것이 바람직 할수 있습니다.

[*] crond
[*] dovecot (pop3 사용 안할경우 선택 해제함)
[*] iptables
[*] network
[*] named (자체 네임서버 사용을 안하는경우 선택해제함)
[*] network
[*] sendmail
[*] sshd

[*] syslog
[*] vsftpd
[*] xinetd

Chkconfig 예)

[그림 9-2] chkconfig

chkconfig --list 라고 하면 현재 시스템에 설정된 데몬들의 리스트가 보이는데 여기서 보이는 숫자 0-6은 런레벨이고 런레벨 별로 on/off 를 적용할수 있습니다.

chkconfig 사용법

```
usage:    chkconfig --list [name]
          chkconfig --add <name>
          chkconfig --del <name>
          chkconfig [--level <levels>] <name> <on|off|reset|resetpriorities>
```

예제)
chkconfig --level 3 vsftpd on
- Run level 3 에서 vsftpd 데몬을 사용 한다는 의미

Run level(런레벨) 종류
0 : 시스템 종료
1 : 싱글유저 모드로 부팅시 사용
2 : NFS를 지원하지 않는 다중 사용자 모드
3 : 네트워킹을 지원 하는 다중 사용자 모드(디폴트 모드)
4 : 사용자 나름대로 정의해서 사용하실수 있는 모드로 /etc/rc.d/rc.4/를 실행해줌
5 : X(x-windows)를 실행시키기 위한 런레벨
6 : 시스템을 재부팅시 사용되는 레벨

1.2.1 기본 시스템 정보 기록 파일 확인

리눅스 서버에서는 설치 후 접속하면 커널 정보 등을 알려주는 파일이 존재합니다.
아래의 파일들을 빈 파일로 변경 하거나 다른 내용으로 대체 하는 것이 좋습니다.

/etc/issue : 콘솔로 로그인 시도시 보여주는 메시지 설정파일
/etc/issue.net : 원격에서 로그인 시도시 보여주는 메시지 설정파일
/etc/redhat-release : 원격에서 로그인 시도시에 리눅스배포판 정보를 보여주는 메시지
/etc/motd : 로그인을 성공한 다음 보여주는 메시지

1.3 리눅스 설치 후 패치 하기

리눅스 배포판의 경우 배포 된후 시간이 지나면서 제공되는 패키지들의 경우 버그 및 보안 취약점이 발생 하게 되는데 이런 보안 이슈가 발생 하게 되면 배포판 사이트등에서 패치 파일들이 제공 됩니다.

관리자라면 매일 이런 취약점이 발생 했는지등을 점검 해야하며 패치파일이 나오면 패치파일을 받아 서버에 적용 해야 하는데 그리 쉬운일은 아닙니다.
이러한 패키지들의 업데이트를 쉽게 관리 해주는 패키중 하나가 yum 패키지 입니다.
여기서 yum 패키지의 사용방법을 잠시 알아 보겠습니다.

YUM(Yellowdog Updater,Modified)
yum은 Yellowdog Linux 의 자동 패키지 업데이트 프로그램인 YUP에서 유래 되었다고 합니다.
yum은 레드햇 패키지 관리(RPM)를 사용하는 운영체제의 패키지 관리하기 위한 자동화 도구로서 설치,삭제 및 업데이트를 자동으로 처리 할수 있습니다.

yum패키지는 redhat 배포판 CD에 포함되어 있으며 웬만하면 OS 설치시 yum패키지도 설치되는데 설치가 안된 경우 아래와 같이 rpm 패키지로 설치 하면 되겠습니다.

rpm설치 예)
rpm -ivh yum-3.2.22-20.el5.centos.noarch.rpm

주기적으로 yum 명령으로 패키지 업데이트를 확인 하시는 것이 좋으며 Cent OS 의 경우 release 도 같이 업그레이드 할수 있습니다.

참고로 yum 으로 업데이트시 mysql이나 named 패키지가 업데이트 되는 경우 my.cnf 나 named.conf 같은 환경 파일도 같이 변경되는 경우가 있으므로 업데이트후 꼭 확인이 필요 합니다.

yum 기본 사용법)

```
usage: yum [options] COMMAND
```

- pakage 설치
  - yum install pakage

- pakage 삭제
  - yum remove pakage

- pakage 업데이트
  - yum -y update

- 패키지 업데이트 목록 확인
  - yum check-update

- yum 패키지 사용 옵션

```
Loaded plugins: fastestmirror
usage: yum [options] COMMAND

List of Commands:

check-update     Check for available package updates
clean            Remove cached data
deplist          List a package's dependencies
downgrade        downgrade a package
erase            Remove a package or packages from your system
groupinfo        Display details about a package group
groupinstall     Install the packages in a group on your system
grouplist        List available package groups
groupremove      Remove the packages in a group from your system
help             Display a helpful usage message
info             Display details about a package or group of packages
install          Install a package or packages on your system
list             List a package or groups of packages
localinstall     Install a local RPM
makecache        Generate the metadata cache
provides         Find what package provides the given value
reinstall        reinstall a package
repolist         Display the configured software repositories
resolvedep       Determine which package provides the given dependency
search           Search package details for the given string
shell            Run an interactive yum shell
update           Update a package or packages on your system
upgrade          Update packages taking obsoletes into account


options:
  -h, --help              show this help message and exit
  -t, --tolerant          be tolerant of errors
  -C                      run entirely from cache, don't update cache
  -c  [config file]       config file location
  -R  [minutes]           maximum command wait time
  -d  [debug level]       debugging output level
  --showduplicates        show duplicates, in repos, in list/search commands
  -e  [error level]       error output level
  -q, --quiet             quiet operation
  -v, --verbose           verbose operation
  -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)
  -x [package], --exclude=[package]
                          exclude package(s) by name or glob
  --disableexcludes=[repo]
                          disable exclude from main, for a repo or for
                          everything
  --obsoletes             enable obsoletes processing during updates
  --noplugins             disable Yum plugins
  --nogpgcheck            disable gpg signature checking
  --disableplugin=[plugin]
                          disable plugins by name
  --enableplugin=[plugin]
                          enable plugins by name
  --skip-broken           skip packages with depsolving problems
  --color=COLOR           control whether color is used

  Plugin Options:
```

1.4 시스템 파일들 퍼미션 변경

시스템에서 주로 사용되는 시스템 정보를 보여주거나 속성,권한등을 변경 할수 있는 명령들에 대해서 일반 계정에서는 실행이 되지 않도록 퍼미션을 변경 하여야 한층 보안성을 증가 시킬수 있습니다

아래와 같은 파일들은 일반 계정에서는 꼭 필요하지 않은 명령들 이므로 일반 계정 권한에서는 사용하지 않도록 퍼미션을 변경합니다.

퍼미션 변경 예)

```
chmod 100 /usr/bin/top
chmod 100 /usr/bin/pstree
chmod 100 /usr/bin/w
chmod 100 /bin/ps
chmod 100 /usr/bin/who
chmod 100 /usr/bin/find
chmod 100 /bin/df
chmod 100 /bin/netstat
chmod 100 /sbin/ifconfig
chmod 100 /usr/sbin/lsof
chmod 100 /usr/bin/make
chmod 100 /usr/bin/gcc
chmod 100 /usr/bin/g++
chmod 100 /usr/bin/c++
```

1.4.1 SteUID / SetGID 체크하기

시스템에서 suid 와 sgid 를 검색 하여 리스트를 만들어 두면 추후 파일 무결성 검사시 변조 여부를 쉽게 파악 할수 있습니다.

```
find / -type f ₩( -perm -004000 -o -perm -002000 ₩) -exec ls -lg {} ₩;
```

/dev 체크 하기

/dev 폴더는 device 파일 외에 일반 파일들이 생성되지 않는 폴더로 일반 파일들이 있는지 확인 하며 일반 파일이 존재 할 경우 꼭 확인 합니다.

```
find /dev -type f
```

1.5 계정 관리

서버에 OS 설치 후에 시스템 계정들중 사용하지 않는 계정들은 삭제를 하는 것이 보안상 취약점을 줄일수 있습니다.

일반적으로 아래 계정 및 그룹들의 경우 삭제 하여도 서비스에 문제가 없는 것으로 삭제 하도록 합니다.

```
root@local:/
[root@localhost /]# userdel adm
[root@localhost /]# userdel lp
[root@localhost /]# userdel sync
[root@localhost /]# userdel shutdown
[root@localhost /]# userdel halt
[root@localhost /]# userdel news
[root@localhost /]# userdel uucp
[root@localhost /]# userdel operator
[root@localhost /]# userdel games
[root@localhost /]# userdel gopher
[root@localhost /]# userdel ftp (anymous FTP server를 운영하지 않으면 삭제)

[root@localhost /]# groupdel adm
[root@localhost /]# groupdel lp
[root@localhost /]# groupdel news
[root@localhost /]# groupdel uucp
[root@localhost /]# groupdel games
[root@localhost /]# groupdel dip
[root@localhost /]# groupdel pppusers
[root@localhost /]# gropudel slipusers
```

1.5.1 root 계정관리 (사용자 계정 생성 및 관리시 유의 사항)

- 비밀번호 설정하기

계정을 생성하고 비밀번호를 지정시 보안성을 강화 하기 위해 /etc/login.defs 파일에서 비밀번호 길이 등의 자리 숫자를 변경 합니다.

```
PASS_MAX_DAYS    99999   -> 90 (패스워드의 변경없이 사용할 수 있는 최대일자)
PASS_MIN_DAYS    0 -> 1 (패스워드의 변경없이 사용할 수 있는 최소일자)
PASS_MIN_LEN     5 -> 8 (패스워드 최소바이트 수)
PASS_WARN_AGE    7
```

참고로 테스트 계정등 임시 계정을 생성시 비밀번호를 알기쉬운 비밀번호로 설정 하는 경유가 많은데 SSH Brute-Force 공격등에 쉽게 노출되기 쉬우므로 비밀번호를 어렵게 하여 지정하시기 바랍니다.
그리고 테스트 계정은 사용후 삭제를 하는 것이 보안상 좋습니다.

참고)
Brute-Force(무차별 공격)은 사전파일이나 지정된 ID를 이용하여 SSH 접속을 시도하는 공격으로 패스워드 사전 파일을 이용하여 미리 지정한 아이디와 대입하여 접속계정을 알아내는 해킹 방법 입니다

- 타임아웃 설정

ssh 접속후 서버에 아무 작업 없이 경우 일정 시간이 지나면 자동 로그아웃이 되도록 설정합니다.
/etc/profile 파일이나 계정 폴더의.bash_profile 파일에 TMOUT= 을 추가 합니다.

```
(예) TMOUT=3600
```

TMOUT= 변수에 설정한 값은 1 시간을 설정 한것으로(60 * 60 = 3600 초) 사용자가 접속해서 한 시간이 지나도록 아무 것도 하지 않을 때 자동으로 로그아웃 합니다.

- 특정 그룹만의 su 사용 권한 허용하기

root 로긴은 일반 계정들중에서 특정 계정으로만 루트 접속을 할수 있도록 권한을 변경해 줍니다.

/etc/group 파일 내 wheel 그룹에 su 사용권한을 가질 계정을 추가 합니다.

```
(예) wheel:x:10:root,hostway
```

/etc/pam.d/su 파일에 wheel 그룹 설정을 추가 합니다.

```
auth            sufficient      pam_rootok.so
# Uncomment the following line to implicitly trust users in the "wheel" group.
auth            sufficient      pam_wheel.so trust group=wheel
```

2. 시스템 환경 점검 하기

2.1 FTP 보안설정

vsftp 의 경우 anonymous 같은 계정등에 대하여 허용을 해두면 서버의 보안상 취약점이 증가하므로 anonymous 계정 사용이 없다면 차단 해 두는 것이 좋으며 각 계정의 상위 폴더로 접근이 허용 되지 않도록 설정을 수정 해 주는 것이 좋습니다.

- vsftpd.conf 보안설정 확인

```
anonymous_enable=NO       # --> FTP 익명 접근 차단
chroot_local_user=YES     # --> 본인 계정의 디렉토리내에서만 접근허용
use_localtime=YES         # --> 서버의 로컬시간 사용
max_per_ip=5              # --> 한ip에서 최대접속연결
tcp_wrappers=YES          # --> /etc/hosts.allow /etc/hosts.deny 파일의 정책지원
```

2.2 ssh 보안 설정

ssh 에서도 보안상 아래 설정 정도는 확인 해 보고 설정을 해 두는 것이 좋습니다.

SSH 설정 파일 위치: /etc/ssh/sshd_config (배포 버전에 따라 다를 수 있습니다.)

- Port 22
  - sshd는 기본 22번 포트를 사용하는데 대다수의 포트 스캐너들은 서버에서 운영되는 ssh 포트를 스캔합니다.sshd포트를 1024 이상의 포트로 변경하면 포트 스캐닝에 대하여 보안을 한층 강화 할수 있습니다.(예) Port 2200

- PermitRootLogin no
  - 공격자 또는 내부 사용자가 root로 접근 가능할 경우 암호 무작위 입력으로 권한을 획득할 수 있습니다. root 로그인 허용이 yes로 되어 있으면 no로 수정합니다.

- Protocol 2
  - 대부분의 ssh 취약성은 ssh2 보다는 ssh1과 관련된 것들이 많이 있습니다. 공격자로부터 최소한의 안전을 확보하기 위하여 ssh2 사용을 권장합니다.

- AllowUsers 아이디 (기본설정에는 미포함 되어 있음)
  - 기재한 “아이디”만 ssh 접근가능하도록 설정 합니다.

- KeyRegenerationInterval 20m
  - 0분간 키입력이 없을시 로그아웃

- LogLevel INFO
  - 접속을 모두 로그에 기록합니다.

- PermitEmptyPasswords no
  - 빈 패스워드 인정 여부를 설정함

2.3 apache 보안설정

아파치(apache)는 웹 서버로 가장 많이 사용되고 있는데 설정 파일인 httpd.conf 에서 몇가지 보안 설정을 해 주는 것만으로도 웹 보안을 상당히 강화할 수 있습니다.

- ServerTokens Prod

웹 서버 버전정보등을 노출할 경우 보안상 좋지 않으므로 이 지시자의 옵션을 수정하여 정보를 차단 하도록 합니다.(옵션 : Prod,Min,OS,Full)

- 메소드 제한 설정

일반적으로 웹서비스 제공시 GET/HEAD/POST외에는 사용할 경우가 없으므로 아래 같이설정할 것을 권장합니다.

```
<Directory /home>
  <LimitExcept GET POST>
    Order allow,deny
    deny from all
  </LimitExcept>
</Directory>
```

기본적으로 웹 서버에서는 많은 메소드를 제공 하는데, 보안 관점에서 불필요한 메소드를 허용할 필요가 없으므로 반드시 몇 개의 필수 메소드만 제공 하는 것이 좋을 것입니다.

- 특정 디렉토리 웹 접근 통제

특정 디렉토리에 대하여 접금을 통제하며 특정 아이피에 대해서만 허용 하고자 할때는 아래와 같이 설정 할수 있습니다.

```
<Directory /home/hostway/>
  Order deny,all
  Deny from all
  Allow from 10.10.10.10
</Directory>
```

/home/hostway/ 디렉토리 밑으로는 10.10.10.10 대역에서만 접근이 가능하고 그외 아이피에서는 접속이 안되게 됩니다.

- 서버 사이드 파일 설정

```
<Files ~".bak$">
  Order allow,deny
  Deny from all
</Files>
<Files ~".old$">
  Order allow,deny
  Deny from all
</Files>
```

Php 등과 같은 서버 사이드 소스 파일을 임시로 file.old 나 file.bak 등과 같이 수정하여 웹에서 접근 가능하게 되는 경우가 있는데 이런 경우 심각한 보안 문제를 유발 할 수가 있습니다. 위와 같이 설정 해두면 파일 확장자가 old, bak 인 경우 웹 접근을 차단 할수 있습니다.
아니면 아래와 같이 특정 확장자를 php와 같은 사이드 언어로 설정해 웹서버에 소스를 그대로 보이지 않고 실행하도록 하는 방법도 있습니다.

```
AddType application/x-httpd-php .php .inc .bak .old .c
```

➢ 대용량 메모리 제한 설정

웹을 통하여 대용량 메모리를 사용하는 프로세스를 제한 하는 설정으로, 모든 디렉토리에 대해 사용 가능한 메모리를 20MB로 제한, /home/sangyong/디렉토리 이하에 대해서는 예외적으로 50MB 정도로 제한 합니다.

```
RLimitMEM 20000000
<Directory / home/sangyong />
    RLimitMEM 50000000
</Directory>
```

➢ 모드보안 설정

최근 자주 등장하는 공격형태로 게시판과 같은 웹 어플리케이션의 취약성을 이용하여 인증을 우회해 웹을 통해 시스템 명령어를 실행하는 경우가 있습니다.
웹로그등을 보면 wget명령을 이용하여 백도어등을 /tmp,/var/tmp 디렉토리에 업로드후 실행하는 것을 알수 있습니다.
이런 공격성을 차단하는 방법으로는 아파치에 보안모듈을 별도로 설치하여 차단하는 방법이 있습니다.
아래 예제와 같은 설정을 httpd.conf파일에 추가 하면 URI문자열에 지정된 문자(wget,tmp)등이 보이면 접속을 거부하고 로그를 남기게 합니다.

```
<IfModule mod_security.c>
SecFilterDefaultAction “deny,log,status:500”
SecFilterSelective THE_REQUEST “wget”
SecFilterSelective THE_REQUEST “/tmp”
SecFilterSelective THE_REQUEST “lynx”
</IfModule>
```

2.4 php 보안 설정

- safe_mode = Off ==> safe_mode = On
  - ➢ 이 값을 on으로 설정하면 PHP 에 의한 파일 액세스시 권한을 점검합니다. 웹 프로그램이 /etc/passwd 등 주요 시스템 파일을 액세스 하지 못하도록 제한할 수 있으나, 이로 인해 웹 프로그램이 정상 작동하지 않을 수 있으니 주의해야 합니다.

- display_errors = On ==> display_errors = Off
  - ➢ php자체 오류 내용을 통하여 취약점을 찾을수 있는 원인을 제공 할수 있습니다. 보안상 오류가 브라우져에 표시되지 않도록 Off 설정 합니다.

- register_globals = On ==> register_globals = Off
  - 이 값을 on으로 설정하면 PHP가 입력으로 받아들이는 값(환경 변수, GET, POST, 쿠키, Server 변수)을 무조건 전역(Global)변수로 다루게 됩니다.

  전역 변수는 프로그램의 동작 중 어디서나 변수값이 바뀔 수 있기 때문에, 웹 프로그램의 인자 조작, 예기치 못한 오동작 등 다양한 보안 문제가 발생할 수 있습니다. PHP 4.2.0 이후로는 보안상의 문제를 고려해 디폴트로 off로 설정되어 나오지만, 아직 많은 프로그램이 on 상태에서만 작동하도록 개발되어 있어 서버 관리자들이 on으로 변경하는 경우가 많습니다. on 값에 의존하는 프로그램이 있으면 개발자에게 해당 문제를 알리고 수정을 요구하시는 것이 바람직합니다.

- magic_quotes_gpc = Off ==> magic_quotes_gpc = On
  - 이 옵션을 on으로 설정하면 PHP가 입력으로 받아들이는 값(환경 변수, GET, POST, 쿠키, Server 변수)에 단일 인용 부호('), 이중 인용 부호("), 백슬래쉬(), 널문자(NUL)가 포함된 경우 자동으로 해당 문자 앞에 백슬래쉬를 추가하여 특수 문자 처리를 합니다. 이로 인해 웹 프로그램의 인자를 변경하는 SQL 구문 삽입(injection) 공격의 성공률을 낮춰줍니다. 이 값을 off로 설정하면 /etc/passwd%00 과 같이 널 문자를 사용해 시스템 상의 임의의 파일을 열람할 수 있으니 반드시 On으로 설정하시기 바랍니다.

- allow_url_fopen = On ==> allow_url_fopen = Off
  - 이 옵션을 on으로 설정하면 파일 액세스시 외부 사이트의 파일을 불러올 수 있습니다. 이 기능은 분산 컴퓨팅과 개발, 관리 측면에서 매우 편리하지만, 외부 공격자에 의해 서버를 침탈당하게 되는 주요 원인이 되어 왔습니다. 특히 include(), require() 계열의 함수 사용시 심각한 보안 상의 문제를 유발하게 됩니다. 특수한 경우를 제외하고는 이 기능이 필요치 않으므로 이 옵션을 반드시 off로 설정하시기 바랍니다.

  특히 제로보드를 사용한다면 꼭 off 모드로 사용 하시기 바랍니다.

- magic_quotes_sybase = off
  - Sybase 사용자의 정상적인 DB 접속을 위해 만들어진 기능이지만, 이 기능은 magic_quotes_gpc 설정을 무력화합니다. 여러분이 Sybase 사용자가 아니라면 반드시 이 값을 off로 설정하시기 바랍니다.

- safe_mode_exec_dir = 디렉터리
  - 이 옵션을 지정하면 system(), exec(), passthru() 등 외부 명령어 실행시 지정된 디렉터리에 존재하지 않는 프로그램은 실행할 수 없게 됩니다. 공격자가 임의로 업로드한 공격 도구나 wget, xterm 등 공격에 사용될 만한 명령어를 실행할 수 없도록 막을 수 있습니다.

- log_errors = On
  - display_errors = Off로 설정하면 오류 내용을 브라우저에서 확인할 수가 없으므로, PHP 오류를 로그 파일로 남기도록 합니다.

3. 서버 보안 관련 프로그램

3.1 아파치 보안 모듈- Modsecurity

ModSecurity는 Apache 웹 서버를 위한 오픈 소스 웹 방화벽이라 할수 있습니다.
가장 널리 알려져 있는 HTTP,HTTPS 를 이용한 공격을 차단 할 수 있는 웹 방화벽이며 공격의 종류는 XSS, SQL Injection, Command Execute 등을 차단하여 웹 보안에 있어서 최소한(?)의

보안을 해주는 Apache 모듈입니다.

ModSecurity는 O'Reilly사에서 출간한 "Apache Security"라는 책을 쓴 Ivan Ristic가 개발한 툴 로써, 설치 및 차단 Rule 설정 인터페이스가 조금 불편하다는 단점은 있지만 공격차단 기능은 상당히 우수합니다.

Apache는 중소기업이나 웹호스팅업체에서 많이 사용하고 있는 공개 웹서버로써, 이들 중소기업에 고가의 상용 웹방화벽 설치가 어려운 경우가 많은데 이 경우 ModSecurity는 다양한 웹 공격을 효과적으로 막는데 많은 도움을 줄 수 있을 것입니다.

3.1.1 Mod Security의 주요 특징

- 요청(request) 필터링
  - 클라이언트로부터 웹요청이 들어올 때 웹서버 또는 다른 모듈들이 처리하기 전에 ModSecurity가 요청 내용을 분석하여 사전에 필터링 합니다.

- 우회 방지 기술
  - 경로와 파라미터를 분석하기 전에 정규화시켜 우회 공격을 차단합니다.
  - 즉, "//", "W/", ".", "%00" 등 우회 공격용 스트링을 제거하고, 인코딩된 URL을 디코딩 합니다.

- HTTP 프로토콜 이해
  - 엔진이 HTTP 프로토콜을 이해하기 때문에 아주 전문적이고 미세한 필터링을 수행할 수 있습니다.

- POST 페이로드(payload) 분석
  - GET 방식 뿐만 아니라 POST 메소드를 사용해서 전송되는 컨텐츠도 가로채어 분석할 수 있습니다.

- 감사 로깅
  - MosSecurity에서 차단기능을 비활성화시킨 후, 강력한 로깅 기능만으로 침입탐지 시스템 역할을 수행할 수 있도록 합니다.

- HTTPS 필터링
  - 엔진은 웹서버에 임베디드되어 있기 때문에 복호화 한 후에 요청 데이터에 접근하여 HTTPS 를 통한 공격도 필터링할 수 있습니다.

3.1.2 ModSecurity 설치

여기서는 /usr/local/apache2/ 에 설치된 소스 아파치 2.2.14 버전을 기준으로 합니다.

먼저 현재 시스템에 Pcre 가 설치가 되어 있는지 확인을 하여 설치가 안되어 있다면 pcre 를 다운로드 받아서 컴파일 설치 해줍니다

- **PCRE 설치**

사이트 : http://www.pcre.org/
다운로드 : ftp://ftp.csx.cam.ac.uk/pub/software/programming/pcre/

```
root@local:/
[root@localhost /]# wget ftp://ftp.csx.cam.ac.uk/pub/software/programming/pcre/pcre-7.9.tar.gz
[root@localhost /]# tar xvfz pcre-7.9.tar.gz
[root@localhost /]# cd pcre-7.9
[root@localhost /]# ./configure --prefix=/usr/local/pcre
[root@localhost /]# make
[root@localhost /]# make install
```

- **Modsecurity 설치**

사이트 : http://www.modsecurity.org/
다운로드 : http://www.modsecurity.org/download/modsecurity-apache_2.5.9.tar.gz

```
root@local:/
[root@localhost /]# tar xvfz modsecurity-apache_2.5.9.tar.gz
[root@localhost /]# cd modsecurity-apache_2.5.9/apache2/
[root@localhost /]# ./configure --with-apxs=/usr/local/apache2/bin/apxs
 --with-pcre=/usr/local/pcre
 --with-apr=/usr/local/apache2/bin/apr-1-config
[root@localhost /]# make
[root@localhost /]# make install
```

httpd.conf 파일의 모듈 리스트 부분에 아래 내용이 없으면 추가 해 줍니다.

```
LoadModule security2_module      modules/mod_security2.so
```

modsecurity.conf룰셋은 아래 설명과 같이 한국정보보호진흥원에서 최근버전의 샘플을 다운받은 후 수정하여 사용 하면 되겠습니다.
아래 룰셋 부분을 참고 해주시기 바랍니다.
그리고 httpd.conf 파일의 하단쯤에 수정된 룰셋 conf파일을 Include conf/modsecurity.conf 같은 형식으로 추가 합니다.

아파치 syntax check 를 해보고 이상 유무를 확인 합니다.

```
root@local:/
[root@localhost /]# /usr/local/apache2/bin/httpd -t
Syntax OK
```

modsecurity 로그는Modsecurity 룰셋의 SecAuditLog 에서 지정한 로그 파일로 생성되며 아파치 로그 폴더와 같은 위치에 생성될 것입니다.

- **ModSecurity 의 룰셋**

ModSecurity 의 룰셋은 한국정보보호진흥원에서 배포하는 샘플 룰셋을 받아 서버의 환경에 맞게 필요한 부분을 편집을 해서 사용 합니다. 사이트 : http://www.securenet.or.kr/
여기서 주의 할점은 배포가 되는 룰셋이 모든 서버에 완벽히 동작하는 것은 아니고 각 서버마다 필요한 것을 수정/추가 하면 됩니다.
예)ModSecurity_1x_hosting_090311.conf - 1.9.x 버전 호스팅업체용
ModSecurity_1x_SMB_090311.conf - 1.9.x 버전 중소기업용
ModSecurity_2x_hosting_09311.conf - 2.x 버전 호스팅업체용
ModSecurity_2x_SMB_09311.conf - 2.x 버전 중소기업용

```
##############################
# < 중소기업용 >
#
# 이 Rule은 1대의 서버에 1개의 웹사이트가 운영되는 중소기업의 웹사이트를 위한 최소공격 차단 Rule입니다.
# 이 Rule을 참고하여 각 웹사이트에 적합한 Rule로 커스트마이징하시기 바랍니다.
# Rule 커스트마이징 후에는 공격탐지시 차단하도록 SecFilterDefaultAction 에서
# pass를 deny로 수정하시기 바랍니다.
#
##############################


##############################
# 1. ModSecurity 동작 유/무
# SecFilterEngine On | Off
# On : ModSecurity 기능 활성화
# Off : ModSecurity 기능 비활성화

SecFilterEngine On

##############################
# 2. 기본 설정
# 기본적으로 룰이 매치 될 경우 행위(Action) 지정
# SecFilterDefaultAction "행위"
# 행위 : deny, pass, allow, status:apache error code
#
# SecFilterSignatureAction 실질적인 공격패턴에 대한 SignatureAction 지정
# 행위 : deny, pass, allow, status:apache error code
#
# 룰 커스트마이징 완료 후 공격탐지시 차단되도록 SecFilterSignatureAction 에서
# pass를 deny로 수정 필요

# SecFilterSignatureAction "deny,log,status:406"
SecFilterSignatureAction "pass,log"

# 아파치의 기본 로그보다 자세한 공격관련 로그를 기록
SecAuditEngine RelevantOnly
SecAuditLog logs/modsec_audit.log

# 웹서버의 헤더 정보 변경
SecServerSignature "Microsoft-IIS/5.0"
:
:
```

3.2 웹 쉘 탐지 프로그램 Whistl(휘슬)

한국정보보호진흥원에서는 공격자에 의해 생성된 웹쉘을 손쉽게 탐지하고 대응하기 위하여 W"웹쉘 탐지 프로그램(Whistl)W"를 개발하여 보급한다고 합니다. 이 프로그램은 리눅스는 물론 윈도우에서도 동작하고 정확성도 높기 때문에 웹쉘 탐지 및 제거에 매우 추천 할 만한 프로그램이다.

프로그램은 다음 사이트의 공지사항에서 신청서를 작성한 후 신청하면 사용 할 수 있습니다.

http://www.krcert.or.kr/index.jsp

사용방법은 압축파일 해제후 README.txt를 참고 해주시고 보편적으로 쉬운편 입니다. 신청후 받은 프로그램의 압축을 풀면 프로그램과 함께 설명서에 사용법이 적혀져 있습니다.

- 옵션 소개

```
Usage: Whistl [OPTION]
-c                프로그램 실행 환경 설정
-u [id]           프로그램 아이디 지정
-p [password]     프로그램 비밀번호 지정
-r [file path]    오탐, 미탐 파일을 신고하기 위한 파일의 전체 경로 지정
-e [jsp | php]    검사할 확장자 jsp, php 선택
-d [file path]    검사 디렉터리를 추가로 지정
```

- 환경 설정

```
-c 옵션으로 Whistl를 실행하면 현재의 환경 설정이 표시
  ex) # ./Whistl -c

“[ ]” 안의 번호나 명령을 입력하고 Enter를 누르면 해당 값을 입력

[1] Checking Directory : 디폴트 검사 디렉터리, “,”로 구분하여 복수 지정 가능
[2] Inspection Center directory : 검역소 디렉터리 (탐지된 웹쉘을 이동할 디렉터리)
[4] Extension of php : "-e php" 옵션으로 Whistl를 실행할 시 검사할 대상이 되는 파일의
확장자로써 ‘,’로 구분하여 설정
[5] Extension of jsp : "-e jsp" 옵션으로 Whistl를 실행할 시 검사할 대상이 되는 파일의 확장자로써
‘,’로 구분하여 설정
[s] save : 환경 설정 저장
[q] quit : 프로그램 종료
```

다음과 같이 -c 옵션을 주어 환경 설정을 간단히 할 수 있습니다.

```
root@local:/
[root@localhost /]# ./whistl_kernel_2.6 -c

whistl Configuration
        [1] Checking Directory : /home/hostway
        [2] Inspection Center directory : /tmp
        [3] Extension of php       : inc,php,php3,php4,php5,ph
        [4] Extension of jsp       : jsp,js
        [s] save
        [q] quit
```

1번 메뉴를 선택후 검사할 디렉터리를 지정 합니다.

```
root@local:/
Choose Menu : 1
Checking Directory :/home/hostway/html
        [1] Checking Directory : /home/hostway/html
        [2] Inspection Center directory : /tmp
        [3] Extension of php       : inc,php,php3,php4,php5,ph
        [4] Extension of jsp       : jsp,js
        [s] save
        [q] quit
```

3번 메뉴을 누르고 검사할 파일의 확장자를 지정한 후 S를 눌러 저장합니다.

```
root@local:/
Choose Menu : 3
Extension of php : inc,php,php3,php4,php5,ph,htm,html,cgi,bin,txt
        [1] Checking Directory : /home/hostway/html
        [2] Inspection Center directory : /tmp
        [3] Extension of php       : inc,php,php3,php4,php5,ph,txt
        [4] Extension of jsp       : jsp,js
        [s] save
        [q] quit

Choose Menu : s

        [1] Checking Directory : /home/hostway/html
        [2] Inspection Center directory : /tmp
        [3] Extension of php       : inc,php,php3,php4,php5,ph,txt
        [4] Extension of jsp       : jsp,js
        [s] save
        [q] quit

Choose Menu : q
```

다음과 같이 검사를 실행후 아이디 비밀번호를 입력 합니다.

```
root@local:/
[root@localhost /]# ./whistl_kernel_2.6

id : testid
pwd : passwdfile

Checking the configration

        [Config] Checking directory : /home/sangyong/html
        [Config] Inspection Center directory : /tmp

Checking the update status

        [INFO] Pattern Update Finished


Checking / home/sangyong /html directory
        [5 Found] /home/sangyong/html/index2.php
        [18 Found] /home/sangyong/html/index.txt

Check Result
        [INFO] 2 Files checked
        [INFO] 2 Suspected WebShell
        [INFO] Time cost : 00:00:10
        [INFO] Finish sending the checking result
```

웹쉘이 탐지되는 것을 알 수 있습니다.
[5 Found] 나 [18 Found] 같은 숫자는 해당 파일에서 모두 5개의 웹쉘 패턴이 일치되었다는 것을 의미하며 5 이상은 웹쉘이 거의 확실하지만 일치되는 패턴 숫자가 1 이나 2 이라면 정상적인 파일이 아닌지 확인해 봐야 합니다.

3.3 Rootkit hunter 루트킷 탐지 프로그램

rkhunter는 rootkit을 찾아 주는 유틸리티 중 한가지로 설치 및 사용 방법이 쉬우므로 사용하기 편리 합니다.

다운로드 : http://www.rootkit.nl/

- 압축 해제 후 설치

```
root@local:/
[root@localhost /]# tar xvfz rkhunter-1.3.6.tar.gz
cd /usr/local/src/ rkhunter-1.3.6/
./installer.sh --layout /usr/local -install
```

- 업데이트

```
root@local:/
[root@localhost /]# rkhunter -update
```

- 실행 하기 (변조된 파일이 있을 경우 warning 라는 경로 메시지가 보임)

```
root@local:/
[root@localhost /]# rkhunter -c

Checking application versions...

     Checking version of GnuPG                          [ OK ]
     Checking version of Apache                         [ Warning ]
     Checking version of Bind DNS                       [ Warning ]
     Checking version of OpenSSL                        [ Warning ]
     Checking version of PHP                            [ Warning ]
     Checking version of Procmail MTA                   [ OK ]
     Checking version of OpenSSH                        [ Warning ]
```

체크 완료되면 /var/log/rkhunter.log 로그 파일이 생성됩니다.

3.4 Chkrootkit 설치

시스템에 루트킷(rootkit)이 설치되었는지 여부를 손쉽게 체크할 수 있는 프로그램으로 chkrootkit은 일반적인 루트킷뿐 아니라, 커널기반의 루트킷, worm까지도 탐지가 가능합니다. 공격자가 해킹에 성공한 후 다음번 침입을 쉽게 하기 위해 백도어 및 트로이잔 프로그램을 설치하는데 이런 프로그램들을 루트킷이라고 합니다.
루트킷에 포함되는 프로그램으로는 ps, ls, netstat, login등의 시스템 프로그램들이 있는데, 이런 루트킷은 시스템에 원래 있었던 프로그램들과 바꿔치기 되서 관리자가 시스템을 점검해도 이상 없게 보이도록 하고 공격자의 행동을 숨기기도 합니다.

http://www.chkrootkit.org 에서 최신 버젼을 무료로 다운로드 받을수 있습니다

- 다운로드

```
root@local:/
[root@localhost /]# wget ftp://ftp.pangeia.com.br/pub/seg/pac/chkrootkit.tar.gz
```

- 압축 풀기

```
root@local:/
[root@localhost /]# tar -xzvf chkrootkit.tar.gz
```

- 컴파일

```
root@local:/
[root@localhost /]# cd chkrootkit-0.49
[root@localhost /]# make sense
```

위와 같이 설치는 간단 합니다.

- Chkrootkit 사용하기

```
root@local:/
[root@localhost /]# # ./chkrootkit
ROOTDIR is `/'
Checking `amd'... not found
Checking `basename'... not infected
Searching for LPD Worm files and dirs... nothing found
Searching for Ramen Worm files and dirs... nothing found
Searching for Maniac files and dirs... nothing found
Searching for RK17 files and dirs... nothing found
Searching for Ducoci rootkit... nothing found
Searching for Adore Worm... nothing found
Searching for ShitC Worm... nothing found
Searching for Omega Worm... nothing found
```

검사결과 메시지 설명

---

infected : 루트킷으로 변형되었음을 의미함
not infected : 루트킷의 변형이 없음을 의미함
not tested : 점검이 수행되지 못했을때.
not found : 점검한 command가 없을 때

- 실행 옵션

-h : 사용할 수 있는 옵션을 보여줌.
-V : chkrootkit의 버전정보를 보여줌.
-l : 사용가능한 test들을 보여줌.
-d : debug 모드로 자세한 화면을 보여줌.
-q : quiet 모드로서 변조된 정보만 보여줌.
-x : 전문가 모드로 strings 결과를 보여줌.
-r dir : 디렉토리 이하에 대해 체크합니다.
-p dir1:dir2:dirN : 복수개의 디렉토리에서 체크합니다.

- 루트킷이 변경 되었을때 조치

시스템이 해킹을 당해 공격자가 루트권한을 획득하였을 가능성이 높으므로 트로이잔으로 변경된 명령어들을 찾아 원래 것으로 바꿔 줄 수 있으나 가장 안전한 방법으로는 시스템을 재설치하고 관련 취약점등을 패치하고, 불필요한 서비스등을 중지하는 등의 조치를 취해 시스템을 안전하게 한 후 사용 하는 것이 바람직합니다.

3.5 TripWire

시스템 파일의 변조 여부를 모니터링 하는 패키지로 파일 속성 및 디렉토리 정보를 데이터베이스화 하여 변조 여부를 쉽게 파악 할수 있습니다.

트립와이어는 파일의 변조 여부를 모니터링하는 툴로서 보통 공격자가 시스템에 침입을 성공하면 다시 들어오기 위한 구멍(Backdoor)을 만들게 됩니다. 이 때 공격자는 여러 프로그램을 변조하는데 대표적으로 ps, ls, netstat등 시스템의 정보를 확인할 수 있는 명령어가 주러 변조 되게 됩니다. 이렇게 침입 당한 시스템의 파일의 무결성을 검사해주는 프로그램이 tripwire 입니다.

3.5.1 TripWire 원리

tripwire는 MD5, SHA, CRC-32 등의 다양한 해쉬 함수를 제공하고, 파일들에 대한 데이터베이스를 만들어 이를 통해 해커들에 의한 파일들의 변조여부를 검사하도록 되어 있습니다.

즉 tripwire는 먼저 시스템에 존재하는 파일에 대해 데이터 베이스를 생성하고 저장한 후에 생성된 데이터베이스와 비교하여 추가,삭제되거나 변조된 파일이 있는지 점검하고 관리자에게 리포팅해주는 도구입니다.

3.5.2 TripWire 설치

- 다운로드 및 설치

tripwire 소스를 다운받아 컴파일 합니다.

```
root@local:/
[root@localhost /]# wget http://downloads.sourceforge.net/tripwire/tripwire-2.4.1.2-src.tar.bz2
[root@localhost /]# tar xvfj tripwire-2.4.1.2-src.tar.bz2
[root@localhost /]# cd tripwire-2.4.1.2-src
[root@localhost /]# ./configure --prefix=/usr/local/tripwire
[root@localhost /]# make
[root@localhost /]# make install
```

- 사용할 비밀 번호 입력

make install 를 인스톨 하는데 인스톨중에 License Agreement 부분에서 accept하면 site keyfile passphrase 와 local keyfile passphrase 부분등에서 비밀번호 입력 요청이 있는데 사용할 비밀번호를 입력 하면 됩니다

```
root@local:/
Enter the site keyfile passphrase : 설정파일 등을 업데이트하거나 DB를 생성할 때
Enter the local keyfile passphrase : DB를 초기화할 때 사용하는 키 입력
Creating signed configuration file...
Please enter your site passphrase : configuration file을 생성하기 위해 site 키 입력
Please enter your site passphrase : policy file을 생성하기 위해 site 키 입력
```

● 데이터베이스 초기화

tw.pol 라는 정책 화일을 만들고 시스템의 무결성 점검 결과 화일의 DB를 생성 합니다.

```
root@local:/
[root@localhost /]# /usr/local/ tripwire/sbin/tripwire --init
```

tripwire는 데이터베이스를 생성하고 그 결과를 출력합니다.
이후 --init 를 실행을 하는것은 마지막으로 점검했던 파일들의 무결성점검결과를 저장하고 있던 DB를 초기화 한다는 의미 입니다.

● 무결성 검사

다음 명령으로 시스템에 있는 파일들에 대한 무결성(Integrity)을 검사 합니다.

```
root@local:/
[root@localhost /]# cd /usr/local/tripwire
[root@localhost /]# cd sbin
[root@localhost /]# ./tripwire -check
```

/usr/local/etc/tripwire/tw.pol 정책 화일과 비교하여 무결성 검사를 하며 무결성 검사가 끝나면 /usr/local/tripwire/lib/tripwire/report 아래에 호스트명과 실행된 날짜 이름으로 결과 파일이 생성됩니다.

```
root@local:/
[root@localhost /]# cd /usr/local/tripwire/lib/tripwire/report/
[root@localhost /]# ls
```

ssp-desktop-201005xx-xxxxxx.twr 화일이 생성됨을 확인, twr 파일은 암호화 되어있기 때문에 twprint를 이용해 txt 파일로 변환 합니다.

```
root@local:/
[root@localhost /]# cd /usr/local/tripwire/sbin/
[root@localhost /]# ./twprint -m r --twrfile /usr/local/tripwire/lib/tripwire/report/ssp-desktop-201005xx-xxxxxx.twr > report.txt
```

저장된 report.txt 파일을 통하여 파일 속성 및 디렉토리 정보를 데이터베이스화 한 정보를 볼 수 있습니다.

● 데이터베이스 업데이트

무결성 검사가 끝난 후 자신의 시스템에 대한 데이터베이스를 만들고 저장합니다.

```
root@local:/
[root@localhost /]# ./tripwire --update
```

- TripWire 환경설정파일(/usr/local/tripwire/etc/twcfg.txt)

환경설정파일은 tripwire유틸리티와 설정파일들이 어디에 설치되어 있는지 등에 대한 정보를 저장하고 있습니다.

- TripWire 정책파일 (/usr/local/tripwire/etc/twpol.txt)
tripwire 정책파일은 tripwire가 감시할 파일 과 디렉토리 를 설정하는 파일입니다.
초기 설치시 생성된 정책 화일을 현 시스템에 맞도록 불필요한 파일들은 제거하고 필요한 파일은 추가,수정하여 사용 하는 것이 좋습니다.

- Tripwire 점검 결과를 메일로 받기
시스템의 무결성 검사는 매일 이루어져야 합니다.
매일 매일 점검 결과를 관리자 메일로 받아 볼수있게 crontab 에 등록 해 두면 관리자는 메일만 보아도 시스템의 변경 여부를 확인 해 볼수가 있어 편리 합니다.

```
/usr/usr/local/tripwire/sbin/tripwire | /usr/bin/Mail -s "Tripwire Report From Server Name" userID@hostway.co.kr
```

4. Nmap

Nmap(Network Mapper)은 raw IP 패킷을 사용하여 네트워크에 어느 호스트가 살아있고, 어떤 서비스(포트)를 제공하며, 운영체제(OS 버전)가 무엇이며, filter/firewall의 패킷 타입이 무엇인지 등 네트워크의 수많은 특징들을 점검할 수 있는 아주 유용한 도구 입니다.

4.1 nmap 설치

리눅스 배포판에 rpm으로 제공되므로 rpm 버전의 namp을 사용 하는것을 설명 합니다.

4.2 nmap의 사용방법

nmap [Scan Type] [Options] <host or IP>

```
[Scan Type]
 -sT  : TCP scanning의 가장 기초적인 형태로  모든 포트에 대해 스캔하는 뱡식입니다.
 -sS  : full TCP 접속을 하지 않으므로 "half-open" 스캐닝이라 하며 로그를 남기지 않습니다.
 -sF -sX -sN  : SYN 패킷을 막아놓은 방화벽이나 패킷 필터 또는 Synlogger와 Courtney 같은
                스캔을 탐지하는 프로그램들을 무사히 통과할 수 있습니다.
  -sP  : 네트워크의 어느 호스트가 살아있는지를 알고 싶을 때 사용합니다. (ping)
  -sU  : UDP 포트를 스케닝 합니다.
  -sA  : 이 방법은 방화벽의 룰셋을 정밀하게 계획하기 위해 사용됩니다..
  -sR  : 이 방법은 nmap의 다양한 포트 스캔 방법을 조합해서 이루어지는데 열린 포트에 대하여
         프로그램이나 버전등을 확인 할수 있습니다.

[Options]
 -P0  : ICMP echo requests (or responses)를 막아놓은 네트워크의 스캔을 가능하게 합니다.
 -PT  : 일반적이 ICMP ping이 아닌 ACK 패킷으로 ping 을 보내고 RST 패킷으로 응답을 받음.
 -PB  : ping 기본 형태로 ACK (-PT)와 ICMP (-PI) 모두를 사용합니다.
 -O : 시스템의 운영체제를 판별 해줍니다.
 -p <port ranges>  : 점검하고자 하는 포트를 지정하는 옵션으로. 예로 22번 포트를 점검하려면 '-
                     p 22' 하면 되고 또한 '-p 20-30,139,60000-'은 20에서 30사이의 포트와 139번
                     포트, 60000번 이상의 포트에 대해 스캔한다는 것입니다.
 -F  : nmap-services에 나열된 포트만 스캔함.
```

Nmap은 타겟을 지정함에 있어 매우 유연한 동작을 보이는데 하나의 호스트는 물론이고 연속되거나 연속되지 않은 여러 개의 호스트 스캔을 설정할 수 있습니다. 특히 "/mask"를 사용하면 클래스 단위로 스캔을 할 수가 있습니다.

- 연속되지 않은 여러 개의 호스트를 스캔할 경우는 호스트 사이에 "," 입력 하여 사용 합니다.
- 연속되는 여러 개의 호스트를 스캔할 경우는 첫 번째 호스트와 마지막 호스트 사이에 "-" 입력 합니다.
- 클래스 단위로 스캔시 에는 <B class : /16, C class : /24>와 같이 "/mask"를 이용할수 있습니다.

예로 B class를 스캔하고자 할 때는 192.168.0.0/16 또는 192.168.*.* 또는 192.168.0-255.0-255와 같이 정의해 줄수 있습니다.

4.3 사용 예제

-sP 옵션으로 대상호스트가 살아 있음을 확인 합니다.

```
root@local:/
[root@localhost /]# nmap -sP xxx.xxx.xxx.xxx
```

이젠 특정 포트(80)를 검색해 보겠습니다.

```
root@local:/
[root@localhost /]# nmap -sP -PT80 xxx.xxx.xxx.xxx
```

지정된 포트가 아니라 대상호스트의 열린 포트를 모두 검색해 봅니다.

```
root@local:/
[root@localhost /]# nmap -sT xxx.xxx.xxx.xxx
```

대상 호스트의 열린 포트를 알수는 있지만 로그가 남으므로 위험합니다.
스텔스 스캔으로 감시를 피해야 겠지요.

```
root@local:/
[root@localhost /]# nmap -sS xxx.xxx.xxx.xxx
```

UDP port 스캔입니다. 시간이 많이 걸릴 수도 있습니다.

```
root@local:/
[root@localhost /]# [root@localhost /]# nmap -sU xxx.xxx.xxx.xxx
```

이번에는 -O 옵션으로 운영체제를 알아보겠습니다.

```
root@local:/
[root@localhost /]# nmap -sS -O xxx.xxx.xxx.xxx
```

좀더 자세히보여줍니다.

여러개 호스트를 검색시 사용합니다.

10.10.10.10 서버에서 20-80번 포트,110번 포트,60000번 이상 포트를 검색 합니다.

5. tcp-wrapper

TCP_WRAPPERS 를 이용하면 허용하지 않는 외부 주소로부터 들어오는 침입을 막아 보안을 강화시켜 줍니다.

가장 좋은 정책은 /etc/hosts.deny 파일을 열어 ALL: ALL@ALL, PARANOID 줄을 추가해서 모든 호스트를 기본적으로 거부(deny)한 다음, /etc/hosts.allow 파일을 열어 신뢰할 수 있는 호스트만 열어주는 것입니다.
TCP_WRAPPERS는 아래 두 파일로 제어되는데 일단 일치하는 규칙을 찾으면 다른 규칙들은 비교하지 않습니다.(규칙을 찾는 순서는 아래 순서대로이다)

/etc/hosts.allow
/etc/hosts.deny

5.1 tcp-wrapper 의 장점

기존의 설정파일이나 소스코드에 별다른 수정이 필요치 않고, 정상적인 사용자들에게는 불편을 주지 않으면서도 허가되지 않은 접근의 제한 및 탐지가 가능하기 때문에 많은 관리자들이 이용합니다.
TCP Wrapper 는 리눅스를 설치하면 기본 rpm버전으로 설치가 되어 있을 것 입니다.
설치 여부는 rpm -qa | grep tcp_wrappers 로 확인 가능하고 설치가 안되어 있다면 리눅스 배포판 CD등에서 tcp_wrappers-7.6-34 와 같은 rpm파일을 인스톨 하면 사용 할수 있습니다.
TCP Wrapper가 이미 설치되었다면 /etc/hosts.allow 파일과 /etc/hosts.deny파일을 수정하기만 하면 되는데 이 파일들을 통해 서버에 접속을 허용할 대상과 허용하지 않을 대상을 결정합니다.

TCP Wrapper를 통해 TCP 차원에서의 1차적인 접속제한을 설정할 수 있으나, 좀 더 철저한 보안 설정을 위해 Iptables와 같은 방화벽 설정도 같이 사용하기를 권장합니다

5.2 tcp-wrapper 의 기능

tcp-wrapper는 inetd 나 xinetd 모드로 구동되는 데몬들을 보호하는 역할을 한다고 했는데 inetd 이란 시스템에서 네트워크로 접속 제어를 돕는데 inetd 가 관리하는 포트로 요청이 들어오면 inetd 는 곧 이를 tcpd 프로그램으로 전송합니다.
tcpd 는 inetd에서 들어온 접속 요청의 승낙 여부를 hosts.allow 와 hosts.deny 파일에 설정된

규칙에 따라서 요청여부를 결정하여 허용되면 해당 서버 프로세스가 시작될 수 있도록 합니다.

원격 접속 요청 --> xinetd --> Tcp-Wrapper --> Demon 실행

이름에서 알 수 있듯이 hosts.allow는 xinetd이 통제하는 네트워크 서비스로의 클라이언트 접근을 허용하는 규칙의 목록을 담고 있습니다. 또한 hosts.deny 파일에는 접근을 거부하는 규칙이 포함되어 있습니다. hosts.allow 파일은 hosts.deny 파일 보다 우선권을 가지며 개별 IP 주소 (또는 호스트명) 또는 클라이언트의 형태에 기초하여 접근을 허가 또는 거부합니다

5.3 접속 제한 및 허용 하기

먼저 /etc/hosts.deny 파일에서 ALL 이라고 주어 모두 차단을 합니다.

tcp-wrapper 기본은 host.deny에서 모든것을 거부 한 후, host.allow에서 필요한 것만 열어주는 방법을 사용하므로 /etc/hosts.deny 파일에서 모두 차단을 한 상태에서 /etc/hosts.allow 파일에 접근을 허용하려는 데몬들과 원격 접속지의 정보를 기재 합니다.

hosts.allow 에 기재하는 방법은 아래 방법과 같이 세부적으로 사용될 수 있습니다.

```
root@local:/
[root@localhost /]# cat /etc/hosts.allow

#
# hosts.allow   This file describes the names of the hosts which are
#               allowed to use the local INET services, as decided
#               by the '/usr/sbin/tcpd' server.
ALL: 10.10.10.    .                                    A
ALL: 10.10.100.10                                      B
ALL: 10.10.10. EXCEPT 10.10.10.110                     C
ALL: .hostway.co.kr EXCEPT test.hostway.co.kr          D
in.proftpd: ALL                                        E
sshd: 10.10.10.                                        F
```

A. 10.10.10.xxx IP 대역으로 부터의 모든 서비스 접근을 허용합니다.
B. 10.10.100.10 IP로부터의 모든 서비스 접근을 허용합니다.
C. 10.10.10.xxx IP대역으로 부터의 모든 서비스 접근을 허용하지만,10.10.10.110 IP 는 제외 됩니다.
D. xxx.hostway.co.kr 대역 호스트로 부터의 모든 서비스 접근을 허용하지만, test.hostway.co.kr 호스트는 제외 됩니다.
E. proftp 서비스로의 모든 접근을 허용합니다.
F. SSH(보안텔넷) 서비스로의 접속을 요청하는, 10.10.10.xxx IP 대역 호스트의 접근을 허용합니다

6. iptables 사용하기

iptables 는 커널의 패킷 필터링 테이블에 필터링 규칙을 삽입하거나 삭제하는 도구 입니다 패킷필터링은 커널에 탑제된 netfilter기능으로 하여 iptables은 단지 netfilter 의 룰을 세워 줄 뿐입니다. 즉 iptables는 룰셋 구축 도구라고 할수 있습니다.

Iptables의 사용법과 기본적인 firewall 구성법에 대해서 알아보도록 하겠습니다.

6.1 iptables 의 구조 및 정책

6.1.1 iptables 구조

iptables는 크게 2개의 테이블로 나눌 수 있습니다.
하나는 필터링을 하는 filter 라는 테이블이고, 또 다른 하나는 nat 라는 테이블입니다.
filter라는 테이블은 기본적으로 생략가능하고, nat는 네트워크의 주소를 변환할 때 사용하는 테이블을 -t 옵션을 이용하면 명기해야 합니다.

기본적으로 Iptables 에는 세가지 chain 이 있는데 모든 패킷은 이 세가지 chain중 하나를 통과하게 됩니다.
이 세가지 chain은 INPUT, OUTPUT, FORWARD chain 인데 우선 서버로 들어가는 모든 패킷은 INPUT chain을 통과하고 그리고 나가는 모든 패킷은 OUTPUT chain을 통과하게 됩니다.
그리고 하나의 네트워크에서 다른 곳으로 보내는 모든 패킷은 FORWARD chain을 통과합니다.

Iptables가 작동하는 방식은 이들 각각의 INPUT, OUTPUT, FORWARD chain 에 어떠한 rule을 세우는 지에 따라 달라집니다.

6.1.2 iptables의 정책

iptables는 테이블 형식으로 관리를 합니다.
먼저 등록 된 룰셋이 효력을 발생하기 때문에 룰셋 등록을 하는 순서가 중요합니다.
모든 것을 거부하는 설정이 먼저 오게 되면 이후에 포트를 열어주는 설정이 와도 효과가 없습니다. 그러므로 **먼저 허용하는 정책이 오고 나서 거부하는 정책이 와야 합니다.**

패킷의 처리는 크게 거부할 것인가 허가할 것인가 두 가지이지만, 세부적으로는 ACCEPT, DENY, DROP으로 관리합니다.

A) ACCEPT : 패킷을 허용하는 옵션
B) DENY : 패킷을 허용하지 않는다는 메시지를 보내면서 거부함
사실 전체정책설정(-P)에서 는 사용할 수 없음.
C) DROP : 패킷을 완전히 무시함

```
예) iptables -A INPUT -p tcp --dport 21:30 -j DROP
    iptables -A INPUT -p tcp --dport 25 -j ACCEPT
```

이 경우에는 먼저 22번부터 30번까지의 포트를 목적지로 하는 패킷이 들어오면 무시하라는 줄이 먼저 들어 있습니다. 그리고 다음에 25번 포트를 열라는 명령이 있습니다.
그러나 이 명령은 첫번째 거부 메시지 때문에 패킷이 이미 거부된 상태이어서 효력이 없습니다.
따라서 이 경우는 잘못 설정한 것으로 볼 수가 있으므로 제대로 설정을 하려면 아래와 같이 바꾸어야 합니다.

```
예) iptables -A INPUT -p tcp --dport 25 -j ACCEPT
    iptables -A INPUT -p tcp --dport 22:30 -j DROP
```

이렇게 하면 먼저 25번 포트로 들어오는 것을 허용하고 난 후에 다른 것을 막기 때문에 제대로 된 설정 입니다.

6.2 iptables 기본 형식 및 옵션

iptables는 조작하는 방법은 크게 두가지로 나눌 수 있습니다.
첫번째는 전체 사슬(chain)에 대한 설정이고 두번째는 각 사슬(chain)에 대한 규칙을 조작하는 방법입니다.
참고로 사슬에 대한 동작 설정은 대문자 옵션(-P, -L 등)을 사용하고 사슬에 대한 세부규칙은 소문자 옵션(-s,-p 등)을 사용합니다.

- **기본문법**

```
iptables [-t table] action chain pattern [-j target]
```

항목설명

```
- table : 크게 nat와 filter로 나누며, 기본값이 filter 이므로 filter인 경우는 생략
- action : 전체 사슬에 대한 정책을 지정 하며 -A, -L, -D, -F 등 대문자 옵션이 이에 해당
           합니다.
- chain: 일반적인 필터링에 속하는 INPUT, OUTPUT, FORWARD 가 있으며, nat 테이블에는 POST
         ROUTING, PREROUTING, OUTPUT 이 있습니다.
- pattern: 세부규칙을 지정하는 것으로 소문자 옵션(-s, -p, -d 등)이 이에 해당합니다.
- target: 정책을 지정하는 것으로 DROP, ACCEPT, LOG 등이 해당 합니다.
```

- **사용방법**

```
iptables -A INPUT -s[근원지] --sport[근원지 포트] -d[목적지] --dport[목적지 포트] -j[정책]
```

참고로 Iptable 을 사용할 때에는 기억해야 할 많은 옵션들이 있으므로 man 페이지(man iptables)를 잘 활용하는 것이 좋습니다.

- **명령 옵션**

전체사슬에 대한 작동 >>

```
-N : 새로운 chain을 만들기(--new)
-X : 비어있는 chain을 제거하기(--delete-chain)
-P : 미리 만들어진 체인의 기본정책을 변경하기(--policy)
-L : chain에 설정된 규칙을 나열하기(--list)
-F : chain으로부터 규칙들을 지우기(--flush)
-Z : 모든 chain의chain의 패킷과 바이트 카운터 값을 0으로 만들기(--zero)
```

사슬 내부의 규칙에 대한 작동 >>

```
-A : 체인에 새로운 규칙을 추가하기(--append)
     INPUT, FORWARD: -t가 filter인 경우 사용가능
     POSTROUTING, PREROUTING: -t가 nat인 경우 사용가능
     OUPTUT: 양쪽 다 사용가능
-D : 체인의 어떤 지점의 규칙을 제거하기(--delete)
-C : 패킷을 테스트 하기(--check)
-R : 체인의 어떤 지점에 규칙으로 교체하기(--replace)
-I : 체인의 어떤 지점에 규칙을 삽입하기(--insert)
```

● **chain 종류**

- INPUT : 로컬로 들어오는 패킷
- OUTPUT : 외부로 나가는 패킷(출력 패킷)
- FORWARD : INPUT와 OUTPUT 역할, 라우터에 방화벽을 적용할 때 쓰임

● **Options (iptables matching option)**

모든 iptablesiptables 규틱은 타겟(target)과 함께 규칙을 따르는 패킷을 어떻게 처리할지 iptables에게 알려주는 매치(Match)들을 가집니다

- s : 패킷의 발신지를 명시합니다.(--source)
- p : 패킷의 프로토콜을 명시합니다.(--protocol)
- d : 패킷의 도착지를 명시합니다.(--destination)
- i : 규칙을 적용할 인터페이스 이름을 명시합니다.(--interface)
- j : 규칙에 맞는 패킷을 어떻게 처리할 것인가를 명시합니다.(-jump)
- y : 접속 요청 패킷인 SYN패킷을 허용하지 않음.(--syn)
- f : 두 번째 이후의 조각에 대해서 규칙을 명시합니다.(--fragment)

--state : 연경 살태와의 매칭
--string : 애플리케이션 계층 데이터 바이트 순서와의 매칭
--comment : 커널 메모리메모리 내의 규칙가 연계되는 최대 256 바이트의 주석

● **테이블**

테이블(table)은 패킷 필터링이나 네트워크 주소 변환(NAT,변환(NAT, Network Address Translation)과 같은 기능의 광범위한 범주를 기술하는 iptables 구성요소 이며
filter, nat, magle, raw 와 같은 4개의 테이블이 있습니다.

filter 테이블은 필터링 규칙 적용 되며,
NAT 규칙은 nat 테이블에 적용되며,
mangle 테이블은 패킷 데이터를 변경하는 특수특수 규칙에 적용 됩니다.
raw 테이블은 텟필터의 연결추적 하위시스템과 독립적으로 기능해야 하는하는 규칙에 적용 됩니다.

● 타겟

iptables iptables 는 패킷이 규칙과 일치할 때 동작을 취하는 타겟(Target)을 지원합니다.

- ACCEPT : 패킷이 본래 라우팅대로 진행합니다.
- DROP : 패킷을 버립니다.
- LOG : 패킷을 syslog에 기록합니다.
- REJECT : 패킷을 버리고 이와 동시에 적절한 응답 패킷을 전송합니다.
- RETRUN : 호출 체인 내에서 패킷 처리를 계속합니다.

6.3 기본 사슬에 대한 사용법

6.3.1 출발지(source) 와 목적지(destination) 지정

출발지('-s', '--source', '--src')와 목적지('-d', '--destination', '--dst') IP 주소를 정하는데는 4가지 방법이 있습니다.

첫번째는 hostway.co.kr, localhost 처럼 도메인 네임을 이용하는 것입니다.

예) -s hostway.co.kr, -d localhost

두번째 방법은 '127.0.0.1'과 같은 IP 주소를 이용하는 것입니다.

예) -s 192.168.0.2

세번째와 네번째 방법은 IP 주소의 그룹을 지정하는 것으로 '192.168.1.0/24' 또는 '192.168.1.0/255.255.255.0' 같은 Netmast값을 이용한 형태 입니다.
이 둘은 모두 192.168.1.0 부터 192.168.1.255 사이의 모든 IP 주소를 지정합니다.

예) -s 192.168.1.0/24 : 192.168.1.0 ~ 192.168.1.255
    -s 192.168.0.0/16 : 192.168.0.0 ~ 192.168.255.255
    -s 192.168.1.0/255.255.255.0
    -s 192.168.0.0/255.255.0.0

사용예)

=> 모든 IP주소(0/0) 로부터 들어오는 패킷들을 모든 DROP 한다는 예제입니다.

그리고 많은 지시자들('-s'나 '-d' 같은)은 일치하지 않는 주소를 나타내기 위해 '!'('not'을 의미)로 시작하는 설정을 할 수 있습니다.
예로, '-s ! localhost' 는 localhost로 부터 오는 패킷이 아닌 경우를 나타냅니다.

6.3.2 프로토콜(-p) 지정

프로토콜은 '-p' 지시자로 지정할 수 있으며 프로토콜을 지시할 때 사용합니다.
프로토콜을 숫자가 될수 있고 'TCP', 'UDP', 'ICMP' 같은 이름이 되며 대소문자를 구별 하지 않습니다. 그리고 'tcp'는 'TCP'와 같은 역할을 하는데 프로토콜 이름 지정에도 '!'을 이용할 수 있습니다.

사용예)
-p ! TCP
=> TCP 프로토콜이 아닌경우를 나타냄

-i 사용
'-i'('--in-interface')는 패킷이 들어오는 인터페이스를 지정하는데 사용되는데
즉 -i는 INPUT과 FORWARD 사슬에 사용됩니다.

참고: -t가 nat이면 PREROUTING에서만 지정가능하고 인터페이스명 앞에 "!"를 추가하면 그 장치는 제외한다는 의미가 됩니다. 뒤에 "+"를 추가하면 그 이름으로 시작하는 모든 장치를 의미하며 디폴트가 +입니다

-o 사용
'-o'('--out-interface')는 패킷이 나가는 네트워크장치를 지정하는데 보통 OUTPUT, FORWARD 사슬에 사용됩니다.

-t(--table) 사용
table을 선택합니다. filter, nat, mangle의 세가지 선택이 있으며 커널에 해당 테이블을 지원하는 코드가 들어 있어야 하며 모듈 자동적재를 선택하면 그와 관련된 커널 모듈이 적재됩니다. 디폴트는 filter이므로 nat를 사용하려면 필히 nat라고 지정해야 합니다.

6.4 iptables 의 확장 (TCP, UDP, ICMP)

Ipchains 에서 간단하게 적용되던 -p같은 프로토콜 관련 옵션들의 기능들이 세부적인 사항들을 설정할 수 있도록 추가적인 옵션이 제공됩니다.

- TCP 확장

TCP 확장은 -p tcp(또는 --protocol tcp)가 지정되고 추가로 사용할 수 있는 다음과 같은 옵션을 제공합니다.

사용법 : -p tcp [옵션]

옵션 설명

- --tcp-flags

tcp에서 발생하는 flag를 지정하는 옵션입니다. 보통 두개의 단어를 사용하는 첫번째 것은 검사하고자 하는 지시자 리스트를 적고, 두번째 단어는 지시자에게 어떤 것이 설정될 것인지를 지정합니다.
예) iptables -A INPUT --protocol tcp --tcp-flag ALL SYN, ACK -j DENY
=> 모든 flag들이 검사되지만 (여기서 ALL은 SYN, ACK, FIN, RST, URG, PSH와 같다.)
SYN 과 ACK만 거부로 설정된다.

- --syn

'!' 옵션이 선행될 수 있으며 이것은 '--tcp-flags SYN,RST,ACK,SYN'의 약어입니다.
--tcp-flags SYN, RST, ACK를 줄여서 사용하는 옵션이며 ! 가 앞에 선행될 수 있습니다.

- --source-port,
  '!' 옵션이 선행될 수 있습니다. 이후에 하나의 TCP 포트나 포트의 범위를 지정합니다.
  발신지에서의 하나의 포트나 포트범위를 지정 하는데 보통 /etc/services에 기록된 것과 같은 포트 이름이 사용될 수 있고 숫자로 나타낼 수도 있습니다.
  범위를 표시 하기 위해 '-'를 사용할 수 있습니다.

- --sport
  /etc/services 에 기록된 것과 같은 포트 이름이 사용될 수 도 있고 숫자로 나타낼 수도 있습니다.
  범위는 두 개의 포트 이름을 '-'으로 연결해서 사용하거나 하나의 포트 뒤에뒤에 '-'를 사용하거나 하나의 포트 앞에 '-' 를 덧붙일 수 있습니다.
  --source-port 옵션과 동일함

- --destination-port, --dport
  위의 내용과 같으나 목적지를 지정합니다. 도착지 포트를 지정합니다.

- --tcp-option
  숫자와 tcp옵션이 같은 경우의 패킷을 검사 하는데 tcp옵션을 검사하려 할때 완전한 TCP 헤더를 갖지 않는 것은 자동으로 DROP 됩니다.

예제)

위 룰셋 라인은 ssh 를 사용하는 것을 허용하는 것으로 출처(source)와 목적지(destination)는 명시하지 않았기 때문에 전체 포트와 IP가 대상이 됩니다.
-dport 는 패킷이 대상으로 삼는 포트를 명시한 것이고 여기에서 22라고 표기한 것은 ssh 서비스 포트입니다. 그리고 마지막에 -j ACCEPT는 허용하도록 정책을 정하는 것입니다.
따라서 여기로의 ssh서비스를 요청하는 패킷은 모두 허용되도록 설정을 한 것입니다.

-j의 사용
    => 특정한 정책(ACCEPT, DROP, DENY, REDIRECT 등)을 설정합니다.
    사용예
       iptables -A INPUT -s 192.168.1.20 -j DROP
        => 192.168.1.20으로 부터 들어오는 모든 패킷에 대해 거부

- UDP 확장
 -p udp(또는 --protocol udp)로 지정하고 '--source-port','--sport','--destination-port','-dport'를 지원합니다.

- ICMP 확장
-p icmp(또는 --protocol icmp) 뒤에 --icmp-type만 추가옵션으로 지원합니다.
사용법
     -p icmp --icmp-type [추가명령]

- 기타 확장 적용

● Mac

이 모듈은 '-m mac' 또는 '--match mac' 이라고 지정할 수 있습니다.
이것은 들어오는 패킷의 이더넷 주소를 검사 하는 것으로 입력 체인에서만 유용합니다.

--mac-source
'!' 옵션이 선행 될 수 있고 이후에 콜론으로 분리된 16진수 숫자의 이더넷 주소가 옵니다.

● limit

이 모듈은 '-m limit' 또는 '--match limit'이라고 함으로 지정할 수 있습니다.
이것은 로그 메세지를 억제할때 처럼 적용검사의 속도를 제한하는데 사용합니다.

--limit
숫자가 따라옵니다 : 초당 평균 최대 적용 검사 수를 지정합니다.

--limit-burst
숫자가 따라오며 위의 제한이 적용되기전의 최대 Burst(?) 를 제한 합니다.

이 적용은 종종 로그의 속도를 제한하기위하여 LOG 타겟과 함께 사용됩니다.
이것을 이해하기 위하여 아래에 기본 제한설정을 하는 로그 패킷제한 을 보면

이 규칙에 도달될때까지 패킷은 로그될 것입니다. 사실 Burst의 기본값은 5 이므로 처음 5개의 패킷은 로그될것입니다.

● owner

이 모듈은 지역에서 생성된 패킷의 생성자의 여러 특징을 적용하려고 하는것으로 출력 체인에만 사용되며 어떤 패킷들(ICMP ping 응답같은)은 소유자 가 없으므로 적용되지 않습니다.

--uid-owner userid
유효한 사용자 id (숫자)의 프로세서가 생성한 패킷에 적용

--uid-owner groupid
유효한 그룹 id (숫자)의 프로세서가 생성한 패킷에 적용

--pid-owner processid
주어진 프로세서 id 의 프로세서가 생성한 패킷에 적용

--sid-owner processid
세션 그룹내의 프로세서가 생성한 패킷에 적용

● 상태 적용

가장 유용한 적용 기준은 'ip_conntrack' 모듈의 접속 추적 분석을 해석하는 'state' 확장입니다.

'-m state'를 지정함으로 '--state' 옵션을 사용할 수 있는데 이후에 콤마로 분리되는

적용할 상태들의 리스트가 오게 됩니다.

- NEW
  ⇨ 새로운 접속을 만드는 패킷

- ESTABLISHED
  ⇨ 존재하는 접속에 속하는 패킷 (즉, 응답 패킷을 가졌던 것)

- RELATED
  ⇨ 기존의 접속의 부분은 아니지만 연관성을 가진 패킷으로 . ICMP 에러 나 ftp 데이터 접속을 형성하는 패킷.

- INVALID
  ⇨ 어떤 이유로 확인할 수 없는 패킷: 알려진 접속과 부합하지 않는 ICMP 에러와 'out of memory' 등을 포함하며 보통 이런 패킷은 DROP 됩니다.

6.4.1 응용예

1) 들어오는 패킷 모두 거부하고 192.168.10.20로 부터 들어오는 모든 패킷들에 대해서만 허가하기

```
iptables -P INPUT DROP
  => 전체 사슬중에 INPUT에 대한 전체정책(-P)를 DROP하면 됩니다.
iptables -A INPUT -s 192.168.10.20 -j ACCEPT
  => 도착지에 대한 명기를 하지 않으면 현재서버를 말하며, 프로토콜을 명기하지 않으면 모든 프로토콜 입니다.
```

2) 192.168.10.20 으로 들어오는 패킷중에서 tcp프로토콜관련 패킷만 거부하기

```
iptables -A INPUT -s 192.168.10.20 -p tcp -j DROP
  => tcp기반 서비스등을 이용할 수 없습니다.그러나  ping같은 icmp프로토콜을 사용하는 패킷은 허가됨
```

3) 포트번호 22번부터 30번까지를 목적지로 들어오는 패킷들을 막고 ssh 포트인 22번포트만 허용 하기

```
iptables -A INPUT -p tcp --dport 22:30 -j DROP
iptables -A INPUT -p tcp --dport 22 -j ACCEPT
  => iptables에서 세부규칙은 먼저 등록된 것이 효력을 발생합니다.
  즉 현재의 정책설정은 포트에 대한 거부를 먼저 설정하였기 때문에 다음행에
 22번포트를 허가해도 효력이 없습니다. 정상적으로 설정하려면 다음과 같이 순서를
 바꿔야 합니다.

iptables -A INPUT -p tcp -dport 22 -j ACCEPT
iptables -A INPUT -p tcp --dport 22:30 -j DROP
```

6.4.2 iptables의 규칙 저장하고 불러오기

원하는 대로 방화벽 사슬을 설정해 놓은 후, 그 설정을 저장하여 설정된 내용을 불러올 수가 있습니다. 이 때 저장하는 명령이 iptabless-save 라는 스크립트이고, 불러오는 명령은 iptables-restore 입니다.

1) iptables-save : 설정한 내용을 저장하는 스크립트 입니다.

사용법 : iptables-save > 파일명

예제 :

```
iptables-save > rc.firewall
 => 현재 설정을 rc.firewall 라는 파일로 저장 함.
 iptables-save -v
 => 저장한 내용을 화면에 출력.
```

2) iptables-restore : iptables-save로 저장한 사슬을 복구하는 스크립트입니다.

사용법 : iptables-restore < 파일명

7. iptables 스크립트로 만들어 사용하기

아래 예제 파일 내용과 같은 방법으로 iptables 스크립트를 만들어 사용 하면 시스템의 보안성을 한층 더 강화 할 수 있습니다.

예제)

```
#!/bin/bash

IPTABLES=/sbin/iptables

# SYN Flooding 공격에 대비하고 브로드캐스트 주소에 ping을 쏘는 것을 막음.
echo 1 >/proc/sys/net/ipv4/tcp_syncookies
echo 1 >/proc/sys/net/ipv4/icmp_echo_ignore_broadcasts

# 패킷이 들어오는 인터페이스와 나가는 인터페이스가 같은 지를 검사함.
if [ -e /proc/sys/net/ipv4/conf/all/rp_filter ]; then
   for f in /proc/sys/net/ipv4/conf/*/rp_filter; do
      echo 1 > $f
   done
fi

# 방화벽 정책 초기화
하나의 체인 안의 모든 규칙을 비우는 것은 -F 명령을 사용해 간단하게 할 수 있습니다.
$IPTABLES -F INPUT
$IPTABLES -F OUTPUT

#루프백 접속 허용
다른 곳과 네트워크가 연결되어 있지 않더라도 시스템의 기본 네트워크이며 로컬 호스트의
인터페이스인 루프백에 대해서는 접속이 이뤄질 수 있도록 해야 합니다.
$IPTABLES -A INPUT -i lo -j ACCEPT
$IPTABLES -A OUTPUT -o lo -j ACCEPT

# 알수없는 패킷은 DROP
$IPTABLES -A INPUT -m state --state INVALID -j DROP
$IPTABLES -A OUTPUT -m state --state INVALID -j DROP

$IPTABLES -A INPUT -p tcp -m state --state ESTABLISHED,RELATED -j ACCEPT
# 서비스 포트들을 허용함
$IPTABLES -A INPUT -p tcp --dport 20:22 -m state --state NEW,ESTABLISHED -j ACCEPT
$IPTABLES -A INPUT -p tcp --dport 25 -m state --state NEW,ESTABLISHED -j ACCEPT
$IPTABLES -A INPUT -p tcp --dport 53 -m state --state NEW,ESTABLISHED -j ACCEPT
$IPTABLES -A INPUT -p tcp --dport 80 -m state --state NEW,ESTABLISHED -j ACCEPT
$IPTABLES -A INPUT -p tcp --dport 110 -m state --state NEW,ESTABLISHED -j ACCEPT
$IPTABLES -A INPUT -p tcp --dport 143 -m state --state NEW,ESTABLISHED -j ACCEPT
$IPTABLES -A INPUT -p tcp --dport 443 -m state --state NEW,ESTABLISHED -j ACCEPT
$IPTABLES -A INPUT -p tcp --dport 953 -m state --state NEW,ESTABLISHED -j ACCEPT
$IPTABLES -A INPUT -p tcp --dport 3306 -m state --state NEW,ESTABLISHED -j ACCEPT
# 허용 포트 제외한 포트를 DROP 합니다.
$IPTABLES -A INPUT -p tcp -j DROP
$IPTABLES -A OUTPUT -p tcp --dport 6666:6667 -j DROP
```

Chapter 10. 백업

IT 산업의 눈부신 발전으로 보존해야 하는 데이터는 점점더 많아지고 있으며 해당 데이터의 사업 의존도는 더 높아져가고 있는 추세입니다.
그러던 중 예상치 못한 장애(천재지변, 정전, 작업자의 실수, 해킹, 하드웨어 장애 등)로 인하여 몇 년간 운영해온 데이터가 순식간에 사라진다면 어떻게 하시겠습니까?

위와 같은 사례로 인하여 데이터 백업은 날로 더 부각되어 가고 있으며 또한 상상할 수 없을만큼 많은 장비와 솔루션이 출시되고 있습니다.
이번 chapter에서는 운영 서버의 백업에 대해서 알아보도록 하겠습니다.

1. 백업 종류

1.1 전체 백업(Full Backup)
스케줄러에 의하여 정기적으로 백업 대상의 파일 및 디렉토리 전체를 백업 받는 방식입니다.
특징
- 매번 전체를 백업하므로 많은 시간이 소요됨.
- 많은 저장 공간 필요.
- 복구를 원하는 날짜의 백업 본을 통한 손쉬운 복구 가능.
- 파일이 자주 변경되지 않을 때는 비효율적 백업 방식.

1.2 증분 백업(Incremental Backup)
스케줄러에 의하여 전체 백업을 수행 한 후 정기적으로 변경된 부분에 대해서만 백업 받는 방식입니다.
특징
- 전체 백업 당시에는 많은 시간이 소요되나 이 후 변경된 파일 백업 시 백업 시간 단축.
- 내용이 너무 방대하고 잦은 변경이 일어나지 않는 데이터의 백업 방식으로 적합.
- 복구 시에는 전체 백업 시점에서 현재 시점까지 변경분을 모두 복구해야 하는 복잡함.
- 복구 시 전체 백업 시점부터 현재 시점까지의 데이터 중 일부 데이터 소실 시 복구 불가능.

2. 백업 주기
백업 대상 파일의 변경 횟수, 중요도, 저장 공간, 비용 등을 고려하여 적절하게 설정합니다.
파일의 중요성에 치중하여 백업 비용이 운영 비용을 초과해서는 안되며 공간이 부족하여 백업 데이터 보관 일수가 줄어들면 운영 위험은 높아집니다.
또한 잦은 변경의 파일은 자주 백업해 주며 일반 파일과 DB 데이터 등은 별도의 스케줄에 의하여 백업 될 수 있도록 설정하시는 것이 좋습니다.

3. 백업 대상
서버 초기 구축 당시에 설정 후 잦은 변경이 없는 시스템 및 프로그램 설정 파일과 사용자의 데이터 파일로 구분해 보았습니다.
특히, OS 설정 및 RPM 패키지 설정 파일은 대부분 /etc 디렉토리에 위치하며 전체 용량이 작으므로 /etc 전체를 백업해 두시는 것도 좋은 방법입니다.

3.1 시스템 파일(프로그램 설정 파일)
OS 가 설치되어 서버가 처음 구축 될 당시 정상 기능 수행을 위하여 설정하는 파일을 의미 합니다.
해당 파일은 초기 구축 시 설정되고 나면 거의 수정되지 않는 특징이 있습니다.
따라서 해당 파일들은 시스템 재구축이나 이전 등의 작업을 진행 할 때 작업 시간을 줄여주어

서버의 다운 타임을 줄여주는 큰 역할을 하게 됩니다.

아래 나열되는 파일들은 호스트웨이에서 설치되고 있는 시스템의 프로그램 경로와 시스템 파일을 따르고 있습니다.

[표 10-1] 백업 대상 파일 - 시스템 파일

시스템 파일	용도
/etc/hosts	호스트 정보 파일
/etc/lilo.conf	커널 부팅과 관련된 설정 파일
/etc/modprobe.conf	모듈 디바이스 설정 파일
/etc/resolv.conf	네임 서버 지정 파일
/etc/sysconfig/network	호스트 네임 및 게이트웨이 정보 파일
/etc/sysconfig/network-scripts/ifcfg-eth0	네트워크 정보 파일
/etc/sysctl.conf	커널 소프트 패치에 대한 설정 파일

[표 10-2] 백업 대상 파일 - 계정 파일

계정 파일	용도
/etc/passwd	서버 내 계정 정보 파일
/etc/shadow	서버 내 그룹 정보 파일
/etc/group	계정 패스워드 암호화 파일
/etc/gshadow	그룹 패스워드 암호화 파일

[표 10-3] 백업 대상 파일 - 기타 파일

기타 파일	용도
/etc/mail/*	Sendmail 관련 설정 파일 디렉토리
/etc/exports	NFS 관련 설정 파일
/etc/samba/*	SAMBA 관련 설정 파일 디렉토리
/var/named/chroot/etc/named.conf	Bind 네임 서버 관련 설정 파일
/var/named/chroot/var/named/*	Bind 네임 서버 관련 존 파일 디렉토리
/usr/local/apache2/conf/httpd.conf	Apache 설정 파일
/usr/local/apache2/conf/extra/*	Apache 관련 include 설정 파일
/usr/local/Zend/etc/php.ini	Php 설정 파일

3.2 사용자 데이터 파일

앞 절에서 시스템 및 프로그램 파일 중에서 백업 대상을 알아 보았습니다. 사실 앞 절에서 설

명한 파일들은 솔직히 없어도 많은 시간을 들여 다시 설정하면 되기 때문에 반드시 있어야 하는 파일은 아닙니다. 하지만 설정 파일을 백업 받아 둠으로 인해서 새로운 서버로의 세팅이 그만큼 쉬워지고 빨라진다는 것은 시스템 및 프로그램 설정 파일을 백업 받아두는데 충분한 이유가 될 것 입니다.
하지만, 이번 절에서 나열하게 되는 사용자 데이터 파일은 절대 백업 하지 않고서는 다시 복구할 수 없는 것 들 입니다. 그만큼 중요한 것이므로 반드시 백업을 해 두어야 합니다.

[표 10-4] 백업 대상 파일 - 데이터 파일

데이터 파일	용도
/home/*	각 계정 사용자 데이터 디렉토리
/usr/local/mysql/var	Mysql data 디렉토리
/var/spool/mail	각 계정 사용자 메일 박스 디렉토리
/usr/local	사용자 프로그램 설치 경로

앞 서 나열되는 파일들의 경로와 파일들은 평균적이고 일반적인 경로와 파일들이며, 만약 설정을 다르게 했을 경우에는 그에 맞는 경로와 파일들을 백업해야 합니다.

4. 백업 방법
백업 방식은 하드웨어, 솔루션 등도 많지만 많은 비용을 투자하지 않으며 서버에서 손쉽에 구현할 수 있는 추가 디스크를 통한 로컬 백업과, 원격지 서버로 백업하는 외부 백업에 대하여 알아 보도록 하겠습니다.
또한 백업 디스크의 효율적 사용을 위하여 압축을 선택하게 됩니다.
본 절에서는 백업과 관련된 압축, rsync, nfs, samba 에 대해서 설명하도록 하겠습니다.

4.1 압축
백업의 의미는 데이터에 문제가 발생하였을 때 빠르게 복구하여 서비스를 재개하는 것을 목적으로 합니다. 특히 OS 가 설치된 디스크에 로컬 백업은 데이터의 삭제 등으로부터의 복구는 가능하지만 하드웨어 특히 하드 디스크의 장애 발생 시 백업 데이터까지 같이 소실되어 복구가 불가능 할 수도 있습니다.
만일의 사태를 대비하여 로컬 백업은 디스크를 추가 하는 방식으로,

4.1.1 tar
tar 는 일반적으로 파일 및 디렉토리를 하나의 파일로 묶어주는 역할을 합니다. 별도 옵션을 사용하여 압축 파일로 묶는 방법도 가능 합니다.

사용방법

```
tar [vxcf] 파일명
```

옵션 설명
- v : 명령 수행 과정을 출력
- x : 묶음을 해제
- c : 파일을 생성
- f : 파일 이름 지정
- z : 압축

- C : 파일을 풀어 놓을 경로를 지정(명령을 수행하는 디렉토리가 아닌 다른 디렉토리에 파일을 풀고자 할때 사용)

예)
tar -vcf home.tar /home/* : /home 디렉토리 하위의 모든 파일 및 디렉토리를 home.tar 파일로 묶음.
tar -vxf home.tar . : 현재 디렉토리에 home.tar 파일을 풀어 놓음.
Tar -vxf home.tar -C /home : /home 디렉토리 하위에 home.tar 파일을 풀어 놓음.

4.1.2 gzip
파일을 압축하고자 할 때 흔히 사용되는 명령어입니다.
아마 확장자가 gz 으로 끝나는 파일을 많이 보았을 것 입니다. 그것은 gzip으로 압축이 되어 있다는 것을 의미 하는 것 입니다.

사용방법

```
gzip -[d] 파일명
```

옵션 설명
- d : 압축 해제

예)
gzip home.tar : home.tar 파일을 압축하여 home.tar.gz 파일을 생성
gzip -d home.tar.gz . : home.tar.gz 파일을 압축 해제하여 home.tar 파일을 생성

리눅스에서는 tar와 gzip 의 명령을 조합하여 한번의 명령으로 사용하는 것을 허용 합니다.

예)
tar -zcvf home.tar.gz /home/* : home 디렉토리 아래의 모든 파일을 home.tar.gz 으로 묶고 압축 합니다.(-z 옵션 : gzip 을 의미)

tar -zcvf home.tar.gz /home/* : home.tar.gz 를 현재 디렉토리에 합축을 해제하고 묶인 것을 풀어 놓는다.

gzip과 tar 명령은 자주 사용하므로 반드시 숙지하여야 합니다.

다음은 특정 상황을 예로 들어서 로컬 백업을 시행하는 절차 및 방법을 알아보고자 합니다.
목표 : /home 디렉토리 내 모든 파일 및 하위 디렉토리를 /backup 디렉토리로 백업

<백업 절차>
1. /backup 디렉토리 생성
2. /home/* 디렉토리 내 모든 파일 및 하위 디렉토리 백업
3. 백업이 정상적으로 되었는지 확인

<가장 단순한 방법>
1) cd /home
2) tar -zcvf /backup/home.tar.gz *
3) cd /backup

4) ls -al home.tar.gz

위의 방법은 tar 명령을 사용해서 home 디렉토리를 통째로 묶은 후에 /backup 디렉토리에 보관하는 방법입니다. 물론 이것도 좋은 방법이기는 하지만 만약 웹호스팅을 운영한다고 가정 했을 때 /home 디렉토리 안에는 무수히 많은 디렉토리들이 존재할 것 입니다. 많은 도메인을 가상호스팅 해줘야 하기 때문입니다. 전체로 묶었다가 만약 한 도메인 사용자가 자기 파일의 백업본을 복구 시켜 달라고 하면 압축한 파일 전체를 다시 풀어야 하는 불상사가 생깁니다.물론 공간과 시간이 많다면, 지장이 없겠지만, 그리 효율적인 방법은 아닙니다. 또한 가장 중요한 문제로, 압축해 놓은 파일에 문제가 생겼다면, 그것은 복구하기가 어려울 뿐만 아니라 전체 압축파일을 못쓰게 됩니다.
그래서 개선된 방법을 알아보도록 하겠습니다.
이것은 쉘 프로그램을 이용한 방법입니다. 쉘 프로그래밍을 숙지하였다고 생각하고 기타 부가설명은 생략 합니다.

```
#!/bin/bash
for backup in $(ls /home)
do
     tar -zcvf /backup/$backup.tar.gz /home/$backup
done
```

위의 방법은 /home 안의 디렉토리 혹은 파일들을 각각의 이름으로 따로 압축하여 backup에 저장한다는 의미의 쉘 프로그램입니다.
각기 따로 압축이 되어 저장이 되므로, 훨씬 관리가 편하며, 압축 파일이 깨진다고 해도 그것은 일부분의 문제이며 전체 백업에 대해서는 문제를 일으키지 않습니다. 즉, 각 압축 파일마다 독립성이 보장됩니다.

위의 백업 방법도 아주 유용한 방법이기는 하나 여기에 더 첨가를 해 보도록 합니다.
Day by day (1일마다) 백업을 하며, 그 백업을 5일 동안 유지하고 백업이 되면 그 백업 내용을 메일로 보내주며, 백업 후 5일지 지난 파일에 대해서는 디렉토리를 삭제하여 용량을 유지하는 방법을 알아보도록 하겠습니다.

파일명 : backup.sh

```
#!/bin/bash
today=$(date + %m-%d)
rmday=$(date + %m-%d -date '5 days ago')

mkdir -p /backup/$today
cd   /backup/$today

for backup in $(ls /home)
do
     tar -zcvf /backup/$backup.tar.gz /home/$backup
done

echo "<<<< backup info >>>>" > mail.txt
echo "" >> mail.txt
date >> mail.txt
echo "" >> mail.txt
```

```
echo “<<<< backup size >>>>” >> mail.txt
du -sh >> mail.txt
echo “” >> mail.txt
echo “<<<< backup list >>>>” >> mail.txt
ls -alh >> mail.txt
echo “” >> mail.txt
echo “<<<< used hard space >>>>” >> mail.txt
df -h >> mail.txt
echo “” >> mail.txt
mail -s “hostway.co.kr backup mail” hostway@hostway.co.kr < ./mail.txt
```

위의 프로그램은 쉘 프로그램이 실행이 되면 쉘 프로그램이 실행된 시스템 날짜로 /backup/디렉토리 아래 디렉토리가 생성되며 백업이 다 되면, 백업된 내용 및 시스템의 하드 여유 공간을 mail.txt 에 저장했다가 hostway@hostway.co.kr 의 메일로 전송을 하는 쉘 프로그램입니다.

위에서 day by day 백업이라고 했는데, 날마다 직접 관리자가 실행 시켜주는 것이 아니고, cron 에 등록을 해서 시스템에서 일일 정해진 시간마다 스크립트를 실행시켜 자동 백업을 하도록 설정합니다.
cron 에 등록시킨다는 것은 /etc/crontab 에 백업 프로그램을 실행할 날짜와 시간을 지정해 주는 것이지만, cron 은 이런 작업을 더욱 쉽게 할수 있도록 cron.hourly(시간마다), cron.daily(날마다), cron.weekly(주마다), cron.monthly(월마다) 라는 디렉토리를 제공하며, 프로그램이 동작하기 원하는 주기의 디렉토리에 실행할 프로그램을 넣어두면 자동으로 실행이 됩니다.
여기에서는 day by day 백업이므로 /etc/cron.daily 라는 디렉토리 안에 작성한 backup.sh 라는 파일을 넣어주면 됩니다. 이때 chmod 명령을 사용해서 루트만 읽고 쓰고, 실행할 수 있도록 700 정도의 퍼미션을 주도록 합니다.

```
root@local:/
[root@localhost /]# chmod 700 /etc/cron.daily/backup.sh
```

여기서 cron 이라는 것은 사용자가 규칙적으로 (주기적으로) 사용하는 명령이나 프로그램을 예약하여 지정된 시간에 프로그램을 구동시켜주는 프로그램 입니다.
cron 의 설정파일은 /etc/crontab 이며, 일반적으로 이 파일은 수정할 필요가 없으며, 아래에 나열된 디렉토리에 실행할 프로그램을 넣어주기만 하면 동작을 합니다.

[표 10-5] cron 명령의 설명

/etc/cron.hourly	시간미다 실행, 동작하게 할 프로그램을 등록
/etc/cron.daily	날마다 실행, 동작하게 할 프로그램을 등록
/etc/cron.weekly	주간마다 실행, 동작하게 할 프로그램을 등록
/etc/cron.weekly	월마다 실행, 동작하게 할 프로그램을 등록

이렇게 cron.daily 에 등록을 해놓으면 관리자가 날마다 해당 프로그램을 실행시키지 않아도 cron 에서 자동으로 실행을 시켜줍니다. (cron.daily 는 crontab 의 세팅을 변경하지 않았다면 새벽 4 시 02 분에 수행됩니다.)
이렇게 함으로써 cron 에 등록한 자동 백업까지 알아 보았습니다. 물론 여기다가 몇 몇 기능을 더 추가하여 더욱 강력하고, 편리한 백업 쉘 프로그래밍을 제작할 수 있을 것입니다.
그것은 책을 읽는 관리지의 몫이고, 여기서는 활용할 수 있는 간단한 예만을 제시해 준 것입니다.
외부 백업은 글 선두에서도 언급했듯이 백업을 해놓은 것 더욱 안전성을 기하기 위해서나 서버에 공간이 부족하여 외부의 다른 저장 공간으로 자료를 이동시키거나 다운로드 하는 것을 말합니다.

4.2 rsync

rsync 의 경우는 백업을 외부 백업이나 로컬 백업시 모두 이용 할 수 있습니다.
rsync 는 RedHat 배포판에 포함되어 있어 기본으로 리눅스 설치 시 설치되어 있을 것입니다.
설치되어 있는지 확인은 다음과 같이 확인 하실 수 있습니다.

4.2.1 rsync 환경 설정

rsync 로 외부 백업을 이용 하기 위해서는 서버에서 간단한 환경 설정을 맞추어 주어야 합니다. (백업 되어지는 서버의 환경 설정입니다.)
여기서는 rsync 를 xinetd 모드로 운영하는 방법을 설명 하겠습니다.
xinetd 모드로 사용시 /etc/xinetd.d/rsync 라는 파일을 아래와 같이 disable = yes 를 no 로 수정합니다.

```
# default: off
# description: The rsync server is a good addition to an ftp server, as it ₩
#          allows crc checksumming etc.
service rsync
{
        disable = no
        socket_type      = stream
        wait             = no
        user             = root
        server           = /usr/bin/rsync
        server_args      = --daemon
        log_on_failure   += USERID
}
```

위와 같이 수정후 # /etc/rc.d/init.d/xinetd restart 를 해주면 873/tcp 포트가 LISTEN 되어 있을 것입니다.

```
root@local:/
[root@localhost /]# netstat -an | grep LISTEN
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address               Foreign Address             State
tcp        0      0 0.0.0.0:873                 0.0.0.0:*                   LISTEN
tcp        0      0 0.0.0.0:22                  0.0.0.0:*                   LISTEN
```

/etc/rsyncd.conf 라는 환경 설정 파일을 만들어 줍니다.
이 파일은 직접 만들어 주셔야 합니다. 내용은 아래 예제와 같은 방식으로 설정을 해주시면 됩니다.
여기서는 /home/hostway 라는 폴더 하나만 예제로 보여 드립니다만 다른 폴더들도 백업을 할 경우 이와 같은 형식으로 rsyncd.conf 에 추가를 해주면 됩니다.

```
root@local:/
[root@localhost /]# vi /etc/rsyncd.conf
[home-hostway]
path = /home/hostway
comment = hostway
uid = root
gid = root
use chroot = yes
read only = yes
hosts allow = 10.10.10.10
max connections = 3
```

/etc/rsyncd.conf 옵션 설명

[home-user1]	서비스명
path	백업되어지는 디렉토리
comment	설명
uid	파일전송하는 사용자의 id. 기본값은 nobody
gid	파일전송하는 사용자의 그룹 id. 기본값은 nobody
use chroot	위의 path 를 root 디렉토리로 사용. 보안상 필요함.
read only	읽기전용 (클라이언트에서 서버로 올리는 경우는 read only= no 로 설정을 해야됩니다.)
hosts allow	접속허용 아이피. 기본값은 All host 이므로 보안을 유지하려면 반드시 호스트명이나 아이피를 설정합니다.
max connections	동시접속자수

여기서는 원격서버에 미러링 하는 방법과 로컬 하드에서 백업 하는 두 가지 방법을 설명 해 드리도록 하겠습니다.

4.2.2 원격서버로 백업 하기

server A 의 서버에 있는 /home/user1 이라는 데이터를 server B 라는 서버의 /home/user1 로 데이터를 미러링 하는 과정 입니다.

server A 서버에서는 위에서 설명된 환경 설정이 되어 있어야 합니다.
(/etc/xinetd.d/rsync 와 /etc/rsyncd.conf)
각 서버의 아이피는 다음과 같다고 가정 합니다
(server A -> 10.10.10.10 server B -> 10.20.30.40)
server 2 서버에서 다음과 같은 명령으로 server 1 의 데이터를 가져올 수 있습니다.

옵션 설명

-a	아카이브 모드. 심볼릭 링크, 속성, 퍼미션, 소유권 등 보존
-v	작업 과정을 상세하게 보여줍니다.
-u	update only(새로운 파일을 덮어쓰지 않음)
-z	전송시 압축을 합니다.
--delete	서버쪽에 없고 클라이언트쪽에만 있는 파일을 지움, 원본 데이터와 백업되는 데이터를 비교하여 원본 데이터에서 삭제된 데이터는 백업 데이터에서도 삭제됨
:	rsh 나 ssh 를 사용하는 것이며 -e ssh 옵션을 사용하여 ssh 를 사용할 수 있습니다.
::	rsync 자체에서 지원하는 기능을 이용 자료를 가져오는 것으로 873 TCP 포트를 사용합니다.

- 증분 백업

증분백업은 원본에서 변경(Update)되거나 추가(Append)된 파일만 백업 받습니다. 증분백업 후 원본에서 삭제된 파일도 백업 본에는 존재합니다. 처음 백업 시에 백업시간이 많이 걸리지만 그 다음부터는 원본 데이터에서 변경되거나 추가, 삭제 등이 된 데이터만 백업 하므로 시간이 단축됩니다.

```
root@local:/
[root@localhost /]# /usr/bin/rsync -avz 10.10.10..10::home-hostway /home/hostway
```

- 풀 백업

풀 백업은 원본에서 변경(Update)되거나 추가(Append)된 파일을 백업 받고, 원본에서 삭제된 파일은 백업 본에도 삭제합니다.
백업 스케줄을 만드실 경우 백업 서버의 용량을 고려하여 위 두 가지를 혼용하여 만드시는 것이 좋습니다.
이와 같은 백업을 스크립트 등으로 만들어 cron 에 올려 두시면 편리하게 사용하실 수 있습니다.

```
root@local:/
[root@localhost /]# /usr/bin/rsync -avz --delete 10.10.10..10::home-hostway /home/hostway
```

<server2 에서 명령을 실행>

- ssh 를 사용하여 rsync 로 복사하기

```
형식 : rsync -avz -e ssh [hostname(서버IP):/백업 원본 경로]  [/백업 받을 경로]
       rsync -avz -e ssh [A서버]  [B서버]
```

위와 같은 방법으로 복사시 ssh 를 이용하므로 인증 암호를 묻는데 passwd 부분에서 암호를 입력 해주면 파일들이나 폴더가 복사되게 됩니다.
참고로 백업 원본 경로에서 “/” 를 붙인 경우와 안 붙인 경우는 차이가 있으므로 아래 예제를 참고 해주시기 바랍니다.
아래 예제와 같이 테스트 해보시기 바랍니다.

예제 1) /home/hostway 라는 폴더를 /backup/hostway 폴더로 복사합니다.

```
root@local:/
[root@localhost /]# rsync -avz -e ssh 서버아이피:/home/hostway  /backup/hostway
```

예제 2) /home/user1 라는 폴더의 파일들을 /backup/home/ 폴더로 복사합니다.

```
root@local:/
[root@localhost /]# rsync -avz -e ssh 서버아이피:/home/hostway  /backup/hostway
```

4.2.3 rsync 로 로컬에서 백업하기
로컬 서버에서 하드디스크를 추가 하여 추가된 디스크를 백업용으로 사용할 경우, rsync 를 이용 하여 증분백업 및 풀 백업용으로 사용할 수 있습니다. /home 에 있는 자료를 /backup 폴더로 백업해 보도록 하겠습니다.

- 증분 백업

```
root@local:/
[root@localhost /]#  rsync -avz /home/ /backup/
```

- 풀 백업

```
root@local:/
[root@localhost /]#  rsync -avz --delete /home/ /backup/
```

5. nfs
NFS(Network File System)는 TCP 네트워크를 통하여 다른 서버의 파일 시스템을 마운트하여 서버의 자료 공유를 할 수 있게 제공하는 시스템으로 리눅스를 비롯한 동일 운영체제 간에 사용되는 프로토콜입니다.
nfs 서비스를 제공하기 위해서는 nfs 서버와 클라이언트에 연결을 위한 설정을 해주어야 합니다.
nfs 설치, 서버측 설정, 클라이언트 설정, 백업 방법에 관하여 알아보도록 하겠습니다.

5.1 nfs 설치

nfs 는 기본적으로 rpm 패키지 설치되며 호스트웨이 OS 설치시 기본으로 설치 됩니다.
해당 패키지의 설치 여부를 확인해 보신 후 설치가 되지 않은 경우 http://rpmfind.net 사이트에서 해당 패키지를 다운 받으시어 설치하시면 됩니다.

```
root@local:/
[root@localhost /]# rpm -qa | grep nfs
nfs-utils-lib-1.0.8-7.6.el5
nfs-utils-1.0.9-42.el5
system-config-nfs-1.3.23-1.el5
```

5.2 nfs 서버 설정

nfs 서버는 설정에서 허용해 놓은 클라이언트에서 요청시 즉각 마운트 될 수 있도록 사전 준비가 되어 있어야 합니다.
기본적으로 nfs 서버에서는 nfs, nfslock, portmap 데몬이 실행되고 있어야 정상 서비스가 가능 합니다.
단, 3 개의 데몬 중 portmap 데몬이 가장 먼저 실행되어야 정상적 서비스가 가능하며 종료 시에는 nfs 데몬이 가장 먼저 중지 되어야 합니다.
또한 nfs 는 tcp / 2049, portmap 은 tcp / 111 포트를 사용하므로 iptables 및 방화벽을 사용하시는 사용자들은 해당 포트를 허용해 주어야 합니다.

```
root@local:/
[root@localhost /]# /etc/rc.d/init.d/portmap start
portmapper (을)를 시작합니다: [  확인  ]
[root@localhost /]# /etc/rc.d/init.d/nfs start
Starting NFS services:  [  확인  ]
Starting NFS quotas: [  확인  ]
Starting NFS daemon: [  확인  ]
Starting NFS mountd: [  확인  ]
[root@localhost /]# /etc/rc.d/init.d/nfslock start
Starting NFS statd: [  확인  ]
```

nfs 설정 파일은 /etc/export 파일입니다. 형식은 아래와 같습니다.

마운트될디렉토리	클라이언트IP	옵션

옵션 설명

root_squash	클라이언트측 root를 mount 디렉토리에서 nobody 사용자로 인식
no_root_squash	클라이언트측 root를 mount 디렉토리에서 root 사용자로 인식
ro	파일시스템 읽기 전용
rw	파일시스템 읽기/쓰기 전용

예) /backup 이란 디렉토리에 10.20.30.40 IP 를 읽고 쓰기 권한으로 mount 허가 하며 디렉토리 권한에 root 권한을 부여하도록 하겠습니다.
해당 설정 모두 완료 후 nfs restart 혹은 nfs reload 를 통하여 설정 변경 적용해 주셔야 정상 작동 합니다.

```
root@local:/
[root@localhost /]# vi /etc/export
/backup          10.20.30.40(rw,no_root_squash)
```

5.3 nfs 클라이언트 설정
클라이언트 서버측에서는 별다른 설정이 필요 업으며 portmap 데몬만 정상적으로 실행시켜 주시면 됩니다.

5.4 nfs 마운트
nfs 서버와 클라이언트 간의 설정이 모두 완료된 상태 이 후 클라이언트 서버에서 nfs 를 통한 mount 를 하시면 됩니다.
형식은 아래와 같으며 파일 시스템 타입을 꼭 nfs 로 지정하셔야 하며 클라이언트 서버에 mount 될 디렉토리가 실제 존재하지 않는 경우 에러가 발생됨으로 이미 생성된 디렉토리 혹은 사용자가 임의로 만든 디렉토리로 지정해 주셔야 합니다.

```
mount -t nfs nfs서버:/디렉토리 클라이언트서버의mount위치
```

예) nfs 서버(10.10.10.10)의 /backup 디렉토리를 클라이언트 서버(10.20.30.40)의 /home/backup 으로 mount 하기(기본적으로 클라이언트에서 /home/backup 디렉토리가 이미 생성된 상태입니다.)

```
root@local:/
[root@localhost /]# mount -t nfs 10.10.10.10:/backup /home/backup
[root@localhost /]# df
Filesystem           1k-blocks      Used Available Use% Mounted on
/dev/sda1               961824    182712    730252  21% /
/dev/sda2              4806584    594404   3968020  14% /usr
/dev/sda5              2884176    126240   2611428   5% /var
/dev/sda6             28853948  17217712  10170508  63% /home
none                    256816         0    256816   0% /dev/shm
10.10.10.10:/backup
                      28176160  21163936   5580928  80% /home/backup
```

5.5 부팅시 자동 mount

서버내 디렉토리 관련 정보 파일은 /etc/fstab 이며 부팅시 해당 파일을 참조하여 파티션을 mount 하게 됩니다.

따라서 /etc/fstab 파일에 부팅시 자동 mount 될 수 있도록 설정해 놓으면 매번 mount 해야 하는 번거로움을 덜 수 있습니다.

자동 mount 하기 위한 etc/fatab 등록 형식(클라이언트 서버의 /etc/fstab 에 등록합니다.)

nfs서버:/마운트될디렉토리	클라이언트마운트디렉토리	nfs(파일시스템타입)	마운트옵션

예) nfs 서버(10.10.10.10)의 /backup 디렉토리를 클라이언트 서버(10.20.30.40)의 /home/backup 으로 부팅시 자동 mount 하기

```
root@local:/
[root@localhost ~]# vi /etc/fstab
10.10.10.10:/backup      /home/backup      nfs      defaults      0 0
```

5.6 nfs 를 통한 백업 활용하기

앞 절에서 설명 드렸던 압축 기법과 같이 활용하여 원격지 서버를 백업 하는 방법으로 많이 활용되고 있습니다.

원격지의 백업 서버를 해당 시간에 portmap 데몬 start 후 nfs 를 통하여 자신의 서버에 mount 합니다.

이 후 압축 기법을 통한 백업 방법을 활용하여 백업 받을 파일들을 압축하여 자신의 서버에 mount 된 원격의 디렉토리에 파일들을 백업 합니다.

백업 완료 후 mount 해제 후 portmap 을 종료하여 백업을 마무리 합니다.

단, 위와 같은 방법을 통한 백업은 서버 로컬에서 이루어지는 백업이 아닌 외부 백업이므로 최소 2 대 이상의 유닉스 계열의 서버가 필요 합니다. 또한 원격지 백업이므로 백업의 속도적인 측면은 로컬 백업보다 다소 떨어집니다.

예) nfs 서버(10.10.10.10)의 /backup 디렉토리를 클라이언트 서버(10.20.30.40)의 /home/backup 으로 mount 하여 /home 디렉토리 아래 디렉토리 백업하기

(클라이언트 서버의 데이터를 nfs 서버로 백업하는 경우이며 해당 스크립트는 클라이언트 서버에서 실행되어야 합니다.)

파일명 : nfs_backup.sh

```
#!/bin/sh
  service portmap start
  today=$(date +%m-%d)
  rmday=$(date +%m-%d --date '5 days ago')
  mount -t nfs 10.10.10.10:/backup /home/backup
  mkdir /home/backup/$today
  cd /home/backup/$today
  for backup in $(ls /home)
    do
       tar -zcvf $backup.tar.gz /home/$backup
    done
  echo " <<<<  backup info >>>>" > mail.txt
  echo "" >> mail.txt
  date >> mail.txt
  echo "" >> mail.txt
  echo " <<<< backup size >>>> " >> mail.txt
  du -sh >> mail.txt
  echo " " >> mail.txt
  echo " <<<< backup list >>>> " >> mail.txt
  ls -alh >> mail.txt
  echo " " >> mail.txt
  echo " <<<< used hard space >>>> "" >> mail.txt
  df -h >> mail.txt
  echo " " >> mail.txt
  mail -s " hostway.co.kr backup mail " hostway@hostway.co.kr < ./mail.txt
  umount /home/backup
  service portmap stop
```

위와 같은 방식으로 사용하시는 서버의 백업을 받으시고자 하는 디렉토리를 설정하신 후 cron 에 등록하시어 활용하시면 주기적인 백업이 가능하며 백업 내역을 mail 로 확인 가능하므로 손쉬운 관리가 가능 합니다.
실제 호스트웨이에서도 위와 비슷한 방법을 활용하여 고객 분들께 백업 서비스를 지원해 드리고 있습니다.

6. samba

서버를 운영하는 측면에서 유닉스 계열의 서버만 혹은 윈도우 서버만으로 운영하기 불가능한 경우가 많이 있습니다. 서비스의 측면에 따라 그렇고 이미 구축되어 있는 시스템에 추가 구성하는 경우, 이전 관리자가 특정 OS 를 선호하여 발생하는 경우 등 아주 다양한 경우로 여러 종류의 OS 를 혼용하여 서비스 하는 경우가 많습니다.
이렇게 이기종의 OS 를 사용하는 경우 운영체제 간에 데이터 및 하드웨어의 공유가 필요하게 됩니다.

이런 경우 samba 를 이용하면 리눅스 서버의 디렉토리를 윈도우 서버에서 C:, D: 드라이브와 같이 활용이 가능 합니다.

6.1 samba 설치

samba 는 기본적으로 rpm 패키지로 설치 됩니다.
해당 패키지의 설치 여부를 확인해 보신 후 설치가 되지 않은 경우 http://rpmfind.net 사이트에서 해당 패키지를 다운 받아서 설치하면 됩니다.

```
root@local:/
[root@localhost /]# rpm -qa | grep samba
samba-common-3.0.33-3.28.el5
samba-3.0.33-3.28.el5
```

6.2 samba 서버 설정

samba 가 기본적으로 설치된 설정 파일의 위치는 /etc/samba/smb.conf 파일이며 해당 파일에서 모든 설정을 할 수 있습니다.
또한 samba 의 서비스 시작을 위하여 /etc/rc.d/init.d/smb 실행 파일을 구동시키며 데몬 실행 시 삼바 데몬인 smbd 와 Netbiod 데몬인 nmbd 데몬이 구동 됩니다.
samba 의 사용 포트는 tcp / 137,138,139 포트를 이용합니다. iptables 나 방화벽을 사용자는 해당 포트를 오픈해 주셔야 정상적인 서비스가 가능합니다.

```
root@local:/
[root@localhost /]# /etc/rc.d/init.d/smb start
SMB 서비스를 시작하고 있습니다: [  확인  ]
NMB서비스를 시작하고 있습니다: [  확인  ]
```

samba 의 설정 파일인 /etc/samba/smb.conf 에서 설정은 전체 설정(Global Setting) 부분과 공유 정의(Share Definitions) 부분으로 나누어 집니다.
전체 설정 부분의 samba 서버의 전체적인 부분을 세팅하게 되며 공유 정의 부분에서는 각각의 공유 디렉토리의 설정을 세팅하게 됩니다.
일반적인 samba 사용시 전체 설정의 부분은 수정 않아도 되며 공유 정의 부분에서 사용하고자 하는 디렉토리의 일반적인 설정이 이루어 집니다.

6.3 전체 설정(Global Setting)

samba 서버의 전체적인 설정을 정의 하는 부분으로 일반적인 공유 디렉토리와 관련된 설정은 공유 정의 부분에서 정의하며 프린터와 관련된 부분의 설명은 생략 하도록 하겠습니다.

```
workgroup = MYGROUP
- 작업 그룹의 네임 정의
hosts allow = 192.168.1. 192.168.2.0/255.255.255.0
- samba 서버에 접근 호스트 정의
log file = /var/log/samba/%m.log
- samba 로그 파일 정의
max log size = 0
- samba 로그 파일 사이즈를 정의하는 것으로 0 은 무제한을 의미하며 Kbite 단위입니다. 사이즈 설정을 해 놓은 경우 사이즈 초과시 .old 파일로 변환하며 새로운 로그 파일로 재생성 됩니다.
security = user
- 클라이언트에서 접속시에 인증 레벨을 정의
encrypt passwords = yes
smb passwd file = /etc/samba/smbpasswd
- 두 옵션을 같이 사용하며 클라이언트 접속시 인증시 암호화 패스워드 사용을 정의하고 samba 패스워드 파일을 정의 합니다.
```

6.4 공유 정의(Share Definitions)

사용자 공유 디렉토리를 설정해 주는 부분으로 사용자 권한 등을 임의로 부여 가능 합니다. samba 서버에 기본으로 설정되어 있는 공유 디렉토리 부분의 설명입니다.

```
[homes]
- 공유 디렉토리 네임 정의
  comment = /home Directories
  - 공유 디렉토리의 설명
  path = /home
  - 공유 디렉토리의 경로 지정
  browseable = no
  - 공유 디렉토리 리스트 출력 정의
  writable = yes
  - 공유 디렉토리 쓰기 권한 정의
  valid users = %S
  - 공유 디렉토리 접근 가능 계정 리스트 정의
  create mode = 0664
  - 공유 디렉토리 파일 생성시 권한 부여 정의
  directory mode = 0775
  - 공유 디렉토리 디렉토리 생성시 권한 부여 정의
```

예) 10.10.10.0/24 IP 대역에서만 접근 허용, /home/samba 디렉토리를 samba 라고 정의, 공유 디렉토리 리스트를 출력 허용, hostway, admin 계정은 읽고/쓰기 권한을 test 계정은 읽기 권한만을 부여 다른 계정의 접근은 거부, 파일 생성 권한은 644, 디렉토리 생성 권한은 770 으로 정의

```
root@local:/
[root@localhost /]# vi /etc/samba/smb.conf
[samba]
comment = home samba test
path = /home/samba
valid users = hostway admin test
writeble = no
write list = hostway admin
browseable = yes
host allow = 10.10.10.0/255.255.255.0
creat mode = 644
directory mode = 770
```

samba 를 정의한 후 데몬을 restart 해 주면 설정이 바로 적용 됩니다.
이 후 smbpasswd 명령을 통한 samba 계정 생성 및 패스워드를 설정합니다.

예) hostway 계정 생성

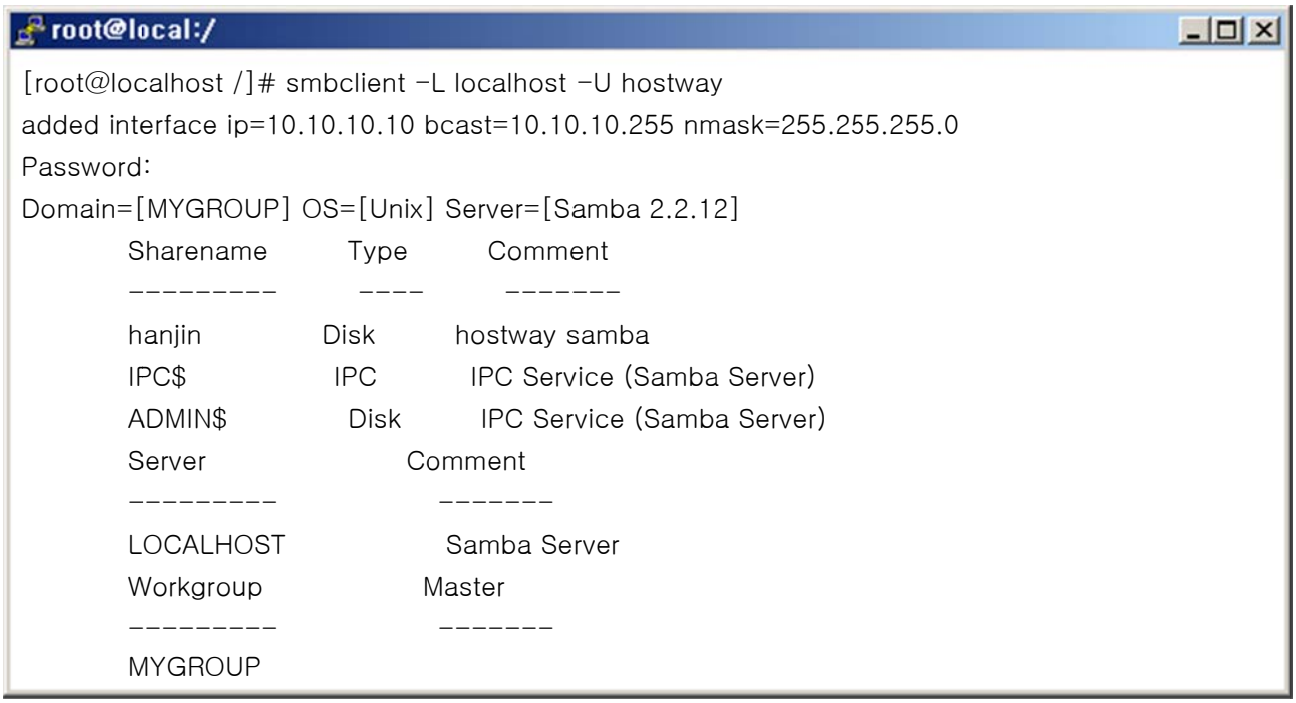
```
root@local:/
[root@localhost /]# smbpasswd -a hostway
New SMB password:
Retype new SMB password:
Added user hostway.
```

모든 설정이 완료 되었습니다. 윈도우 서버에서 네트웍 드라이브 연결 시도하여 사용하면 됩니다.

6.5 samba 설정 test
서버내 samba 를 통한 접속 상태 확인은 smbstatus 명령으로 확인할 수 있습니다.
아래는 smbstatus 명령으로 samba 상태를 확인한 것으로 접근 IP 10.10.10.10 인 hostway 라는 머신으로부터 samba 공유 디렉토리에 hostway 라는 계정으로 접속해 있는 것을 확인할 수 있습니다.

```
root@local:/
[root@localhost /]# smbstatus
Samba version 3.0.33-3.28.el5
Service      uid       gid       pid     machine
----------------------------------------------
samba        hostway   hostway   3104    hostway    (10.10.10.10) Thu Jun 12 10:07:58 2010
```

smbclient 명령을 통하여 samba 계정을 통한 정상 접근 여부를 확인할 수 있습니다.



6.6 윈도우 서버에서 samba 연결

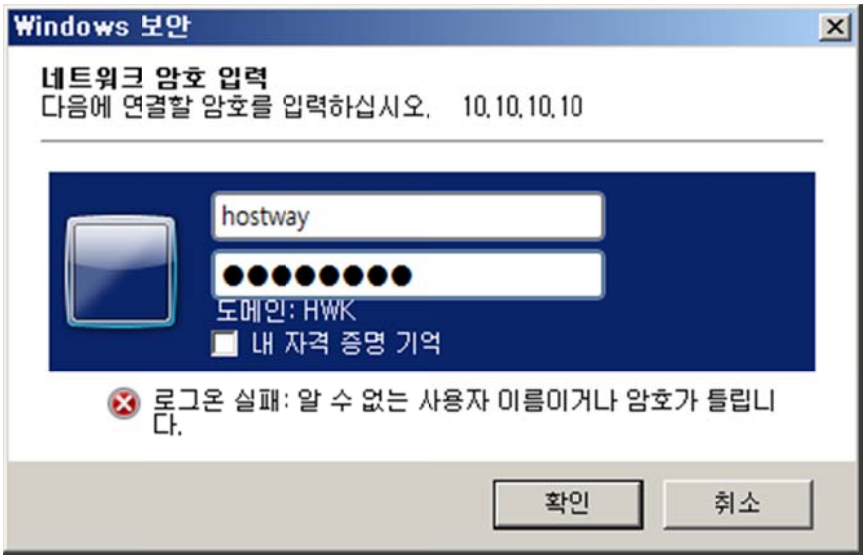
리눅스 서버에 세팅된 samba 를 통하여 윈도우 서버에서 연결 합니다.

연결 방법은 윈도우 하단의 [시작] 버튼을 클릭하여 [실행]을 클릭 합니다. 실행 창이 열리게 되며 열기 부분에 samba 가 세팅된 리눅스 서버의 IP 를 "₩₩리눅스 서버 IP" 의 형식으로 입력합니다.

[그림 10-1] Samba 접속하기

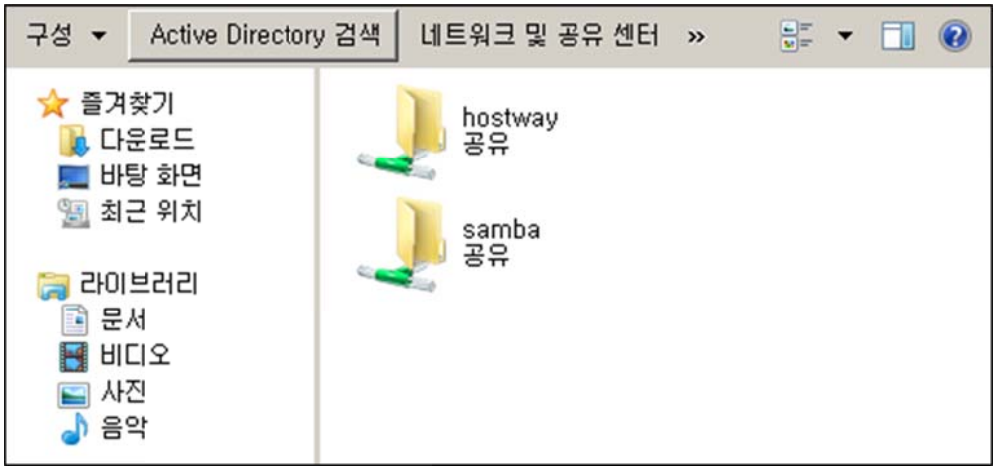
인증 창이 뜨며 사용자 이름과 암호의 입력을 대기 합니다.

리눅스 서버 samba 설정시 세팅한 ID 와 패스워드를 입력합니다.



[그림 10-2] Samba 인증

인증 절차를 거쳐 로그인이 되었습니다. samba 설정시 공유 정의 부분에 설정 하였던 samba 폴더가 보이며 samba 설정시 부여한 각 계정의 권한에 따른 작업이 가능 합니다.



[그림 10-3] Samba 폴더

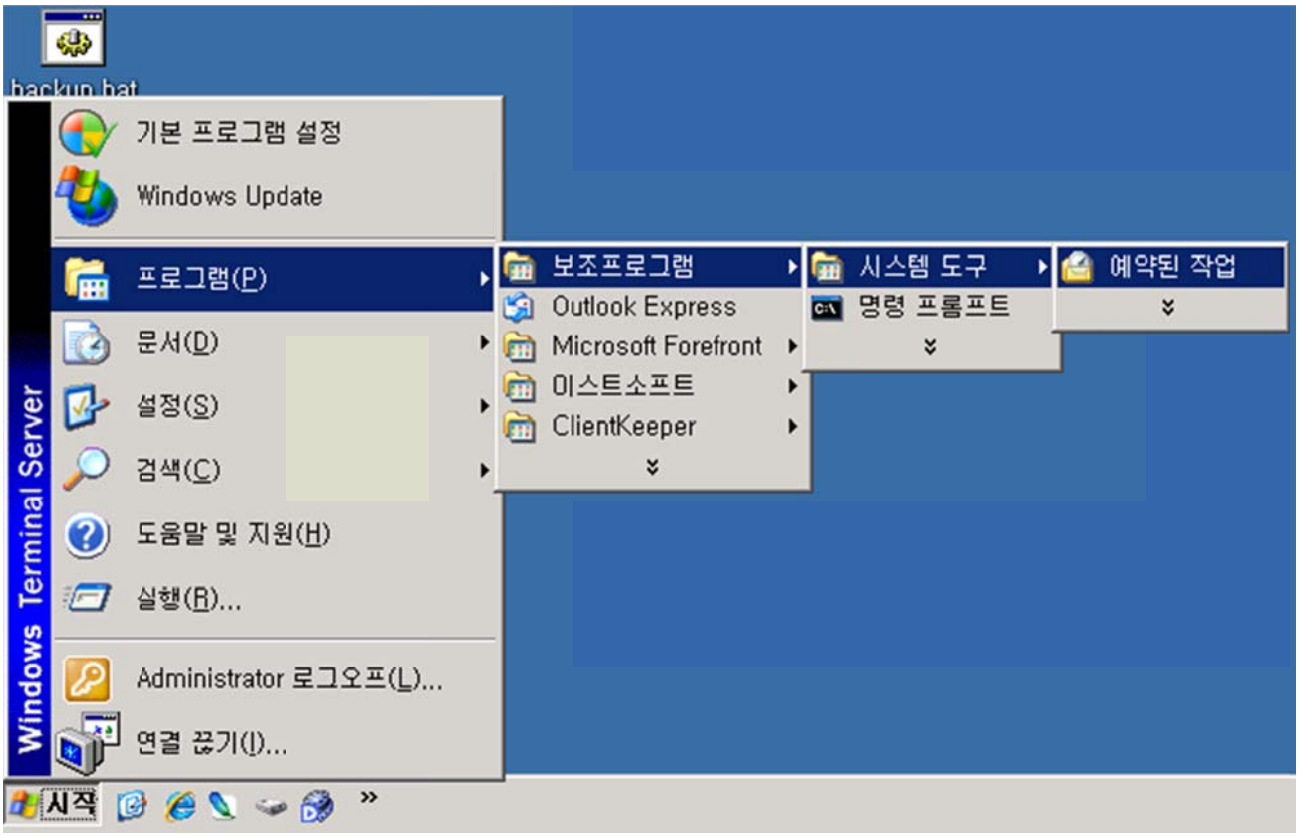
6.7 samba 를 활용한 백업

이기종 OS 간의 백업으로 해당 백업은 네트워크 드라이브를 통한 리눅스 서버의 데이터를 samba 를 이용하여 윈도우 서버에서의 예약 작업 방식으로 백업하게 됩니다.

우선 윈도우 서버에서 접근이 가능 하도록 리눅스 서버에서 samba 설정을 완료 하여야 합니다. 이 후 윈도우 서버에서 메모장 등을 통한 스크립트를 작성 후 확장자 .bat 파일로 스크립트를 저장 합니다.

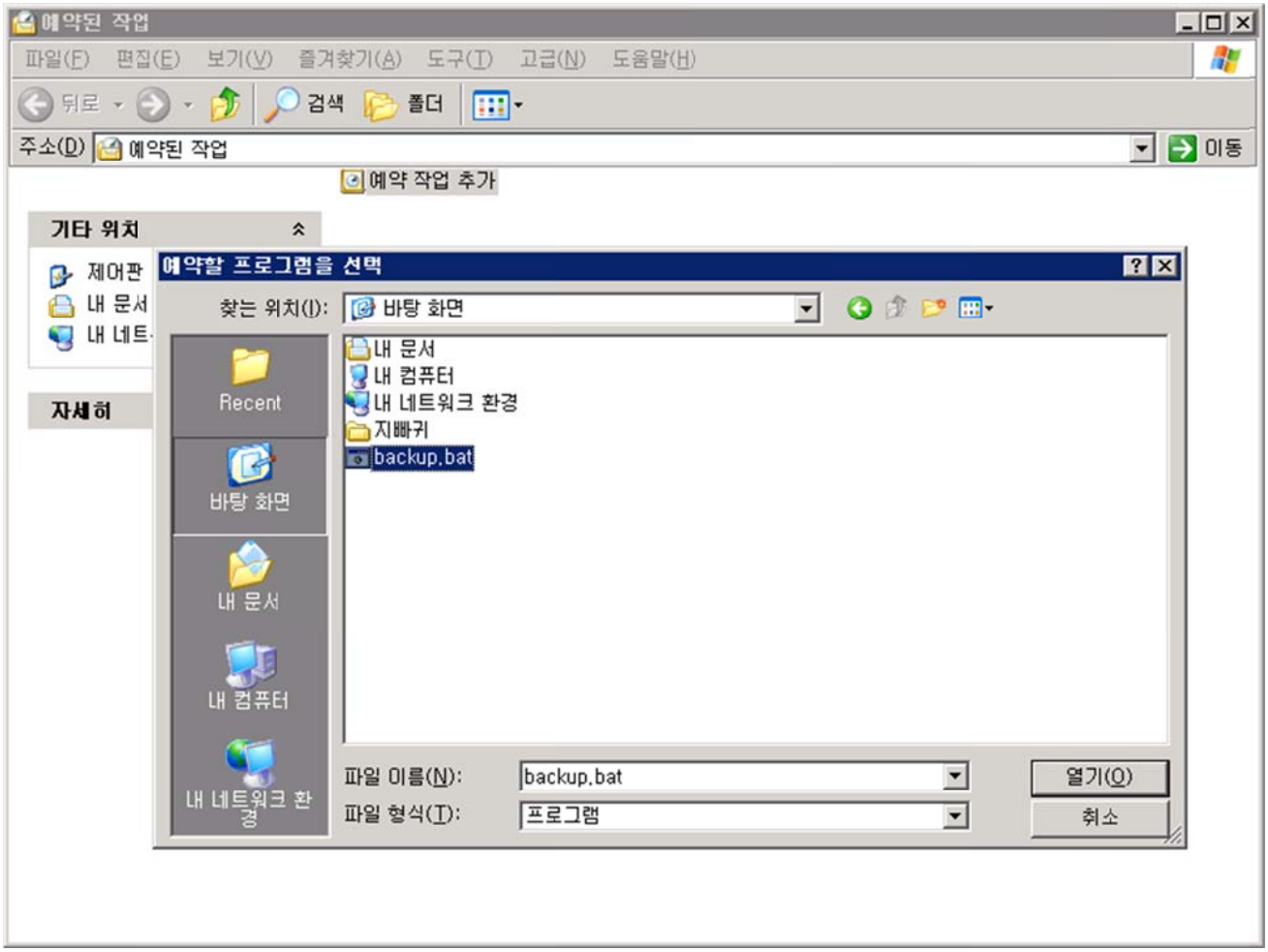
```
net use ₩₩10.10.10.10 /user:hostway hostway
rd e:₩linux_backup₩hostway2 /s /q
move e:₩linux_backup₩hostway1 f:₩linux_backup₩hostway2
xcopy ₩₩10.10.10.10₩samba₩* e:₩linux_backup₩hostway1₩ /e /d /y
/EXCLUDE:e:₩linux_backup₩exclude.txt
```

저장 한 백업 스크립트 파일을 시스템 도구의 [예약된 작업]에 등록하여 정해진 시간에 리눅스 서버의 samba 를 통한 백업을 이루어질 수 있도록 설정 합니다.



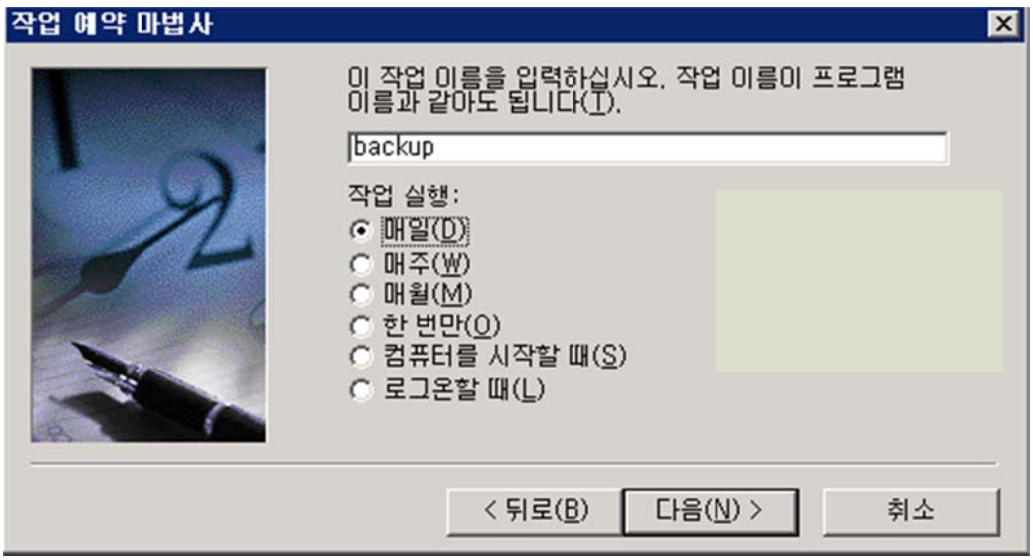
[그림 10-4] 예약된 작업

[예약 작업 추가]를 클릭한 후 [작업 예약 마법사] 찾아보기를 통하여 작성한 스크립트.bat 파일을 등록 합니다.



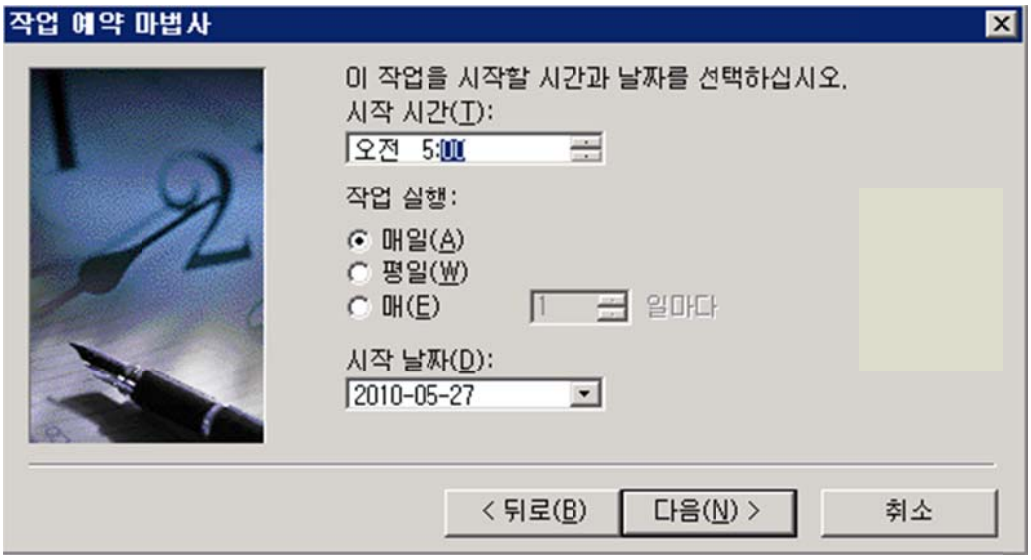
[그림 10-5] 예약할 프로그램 등록

백업 스케줄 중 백업 단위를 선택 체크 합니다.



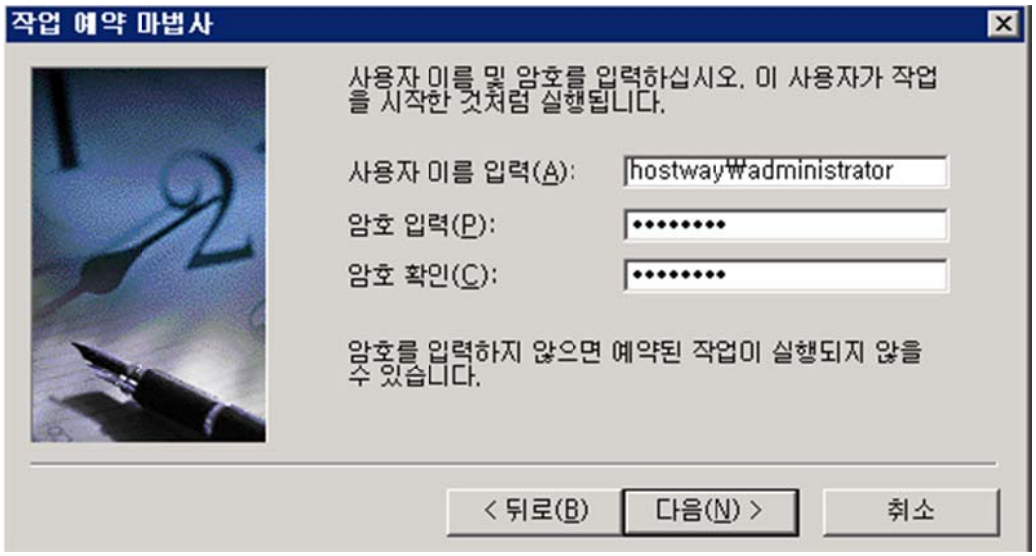
[그림 10-6] 백업 스케줄 설정

백업 될 시간 등을 설정 합니다.



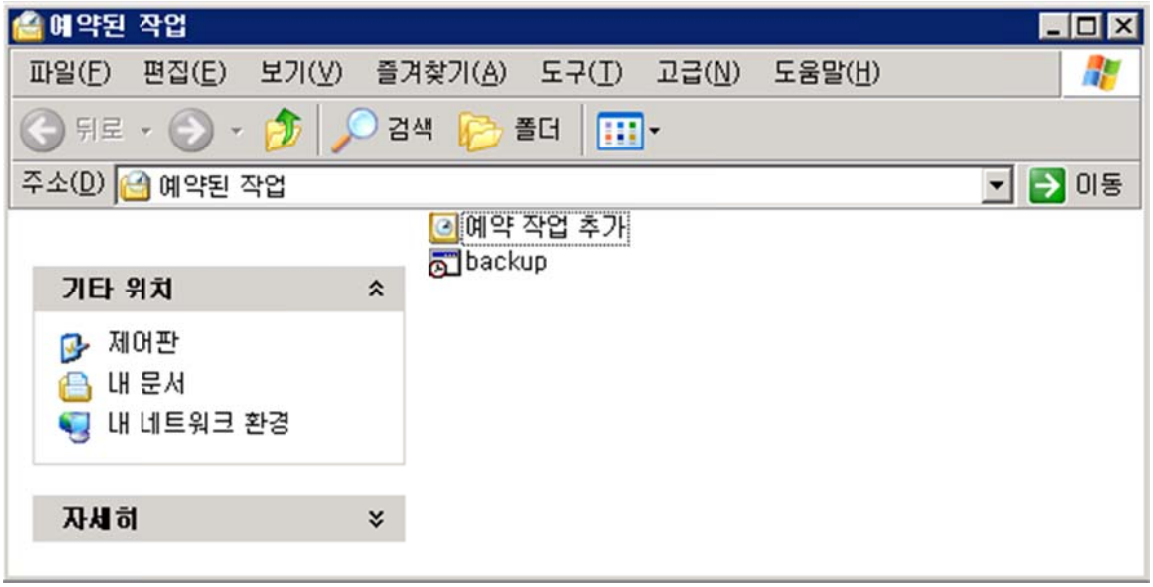
[그림 10-7] 백업 시간 설정

예약 작업 스케줄 등록 시 해당 작업에 필요한 권한 여부를 확인 합니다. 필요에 따른 계정 및 패스워드를 입력 합니다.



[그림 10-8] 백업 서버 인증 정보 입력

예약 작업이 완료 되었으며 예약된 작업에서 설정된 사항을 확인 할 수 있습니다.



[그림 10-9] 예약된 작업 확인

Chapter 11. application 설치

리눅스 서버에서 일반적으로 많이 사용하는 상용 application 설치 작업을 해 보도록 하겠습니다.
이번 장에서 설치할 프로그램은 phpMyAdmin, Zeroboard, Textcube 입니다. 이 세 가지 application 들은 홈페이지 내에서 운영되는 tool 이기 때문에 설치하기 전에 먼저, 가상호스트 설정, 데이터베이스 추가, apache module 확인, 등의 사전 작업이 필요 합니다.

1. 설치준비

mysql-5.0.51a, httpd-2.2.8, php-5.2.11, ZendOptimizer-3.3.3 버전을 설치하여 phpMyAdmin, Zeroboard, Textcube 를 설치하도록 하겠습니다. APM 은 아래와 같이 설치되었습니다.

```
root@local:/
[root@localhost /]# ./configure --prefix=/usr/local/mysql --with-charset=euckr && make && make install
[root@localhost /]# ./configure --prefix=/usr/local/apache2 --enable-rewrite --enable-module=so --enable-shared=max && make && make install
[root@localhost /]# ./configure --prefix=/usr/local/php --with-mysql=/usr/local/mysql --with-apxs2=/usr/local/apache2/bin/apxs --enable-sysvshm=yes --enable-sysvsem=yes --enable-debug=no --enable-track-vars=yes --enable-url-fopen-wrapper=yes --with-ttf --with-png-dir=/usr --with-zlib-dir --with-jpeg-dir=/usr --with-gdbm=/usr --enable-ftp --with-tiff-dir=/usr --enable-memory-limit --enable-mbstring --with-expat-dir=/usr --enable-sockets --enable-wddx --with-freetype-dir=/usr --enable-bcmath --enable-mbstr-enc-trans --enable-mbregex --enable-exif --with-gd --enable-gd-native-ttf --enable-gd-imgstrttf --enable-calendar --with-openssl --with-libxml-dir=/usr/local/libxml2 && make && make install
[root@localhost /]# ./install.sh
```

1.1.1 apache module 확인

Tattertools 를 설치하기 위해서는 apache 의 mod_rewrite 와 mod_alias 모듈이 설치되어 있어야 합니다.
위에 설치한 apache 에서는 --enable-rewrite 옵션을 주었기 때문에 mod_rewrite 가 포함이 되었습니다. 추가된 모듈은 아래와 같이 httpd -l 명령으로 확인 가능합니다. 만약 apache 설치 시 모듈이 포함되어 있지 않다면, 모듈을 추가하거나, apache 를 재설치 한 후에 tattertools 를 설치해야 합니다.

```
root@local:/
[root@localhost /]# /usr/local/apache2/bin/httpd -l
Compiled in modules:
  ~
  mod_alias.c
  mod_rewrite.c
  ~
[root@localhost root]#
```

1.1.2 가상호스트 설정

apache2 에서의 가상호스트 설정은 /usr/local/apache2/conf/extra/httpd-vhosts.conf 파일에서 하게 됩니다.
/usr/local/apache2/conf/extra/httpd-vhosts.conf 파일을 Include 하기 위해서
/usr/local/apache2/conf/httpd.conf 파일에서 아래 부분을 찾아 Include 앞의 주석을 제거 합니다.

```
# Virtual hosts
Include conf/extra/httpd-vhosts.conf
```

아래는 /usr/local/apache2/conf/extra/httpd-vhosts.conf 파일에서 가상호스트 설정한 내용입니다.

```
NameVirtualHost 10.30.100.167
<VirtualHost 10.30.100.167>
    ServerAdmin admin@hostway.co.kr
    DocumentRoot /home/hostway/public_html
    ServerName hostway.co.kr
    ServerAlias www.hostway.co.kr
    ErrorLog logs/hostway.co.kr-error_log
    CustomLog logs/hostway.co.kr-access_log common
</VirtualHost>
```

1.1.3 데이터베이스 추가

Zeroboard 와 Tattertools 에서 사용할 데이터베이스를 추가하도록 하겠습니다.
Zeroboard 에서 사용할 데이터베이스는 bbs 로 추가하고, Tattertools 에서 사용할 데이터베이스는 ttools 로 추가 하도록 하겠습니다.

root@local:/

```
[root@localhost /]# /usr/local/mysql/bin/mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or ₩g.
Your MySQL connection id is 1
Server version: 5.0.51a-log Source distribution
Type 'help;' or '₩h' for help. Type '₩c' to clear the buffer.
mysql> use mysql;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A
Database changed
mysql> create database bbs;
Query OK, 1 row affected (0.02 sec)
mysql> create database ttools;
Query OK, 1 row affected (0.00 sec)
mysql> insert into user (host,user,password) values('localhost','bbs',password('bbs!#%'));
Query OK, 1 row affected, 3 warnings (0.00 sec)
mysql> insert into db values('localhost','bbs','bbs','y','y','y','y','y','y','y','y','y','y','y','y','y','y','y','y','y','y','y','y','y');
Query OK, 1 row affected (0.00 sec)
mysql> insert into user (host,user,password) values('localhost','ttools',password('ttools!#%'));
Query OK, 1 row affected, 3 warnings (0.00 sec)
mysql> insert into db values('localhost','ttools','ttools','y','y','y','y','y','y','y','y','y','y','y','y','y','y','y','y','y','y','y','y');
Query OK, 1 row affected (0.00 sec)
mysql> flush privileges;
Query OK, 0 rows affected (0.00 sec)
mysql> exit
Bye
```

root@local:/

```
[root@localhost /]# /usr/local/mysql/bin/mysql -u bbs -p bbs
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or ₩g.
Your MySQL connection id is 6
Server version: 5.0.51a-log Source distribution

Type 'help;' or '₩h' for help. Type '₩c' to clear the buffer.
mysql> exit
Bye
[root@localhost /]# /usr/local/mysql/bin/mysql -u ttools -p ttools
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or ₩g.
Your MySQL connection id is 6
Server version: 5.0.51a-log Source distribution

Type 'help;' or '₩h' for help. Type '₩c' to clear the buffer.
mysql> exit
Bye
```

2. phpMyAdmin 설치

2.1 source 파일 다운로드 및 압축 풀기

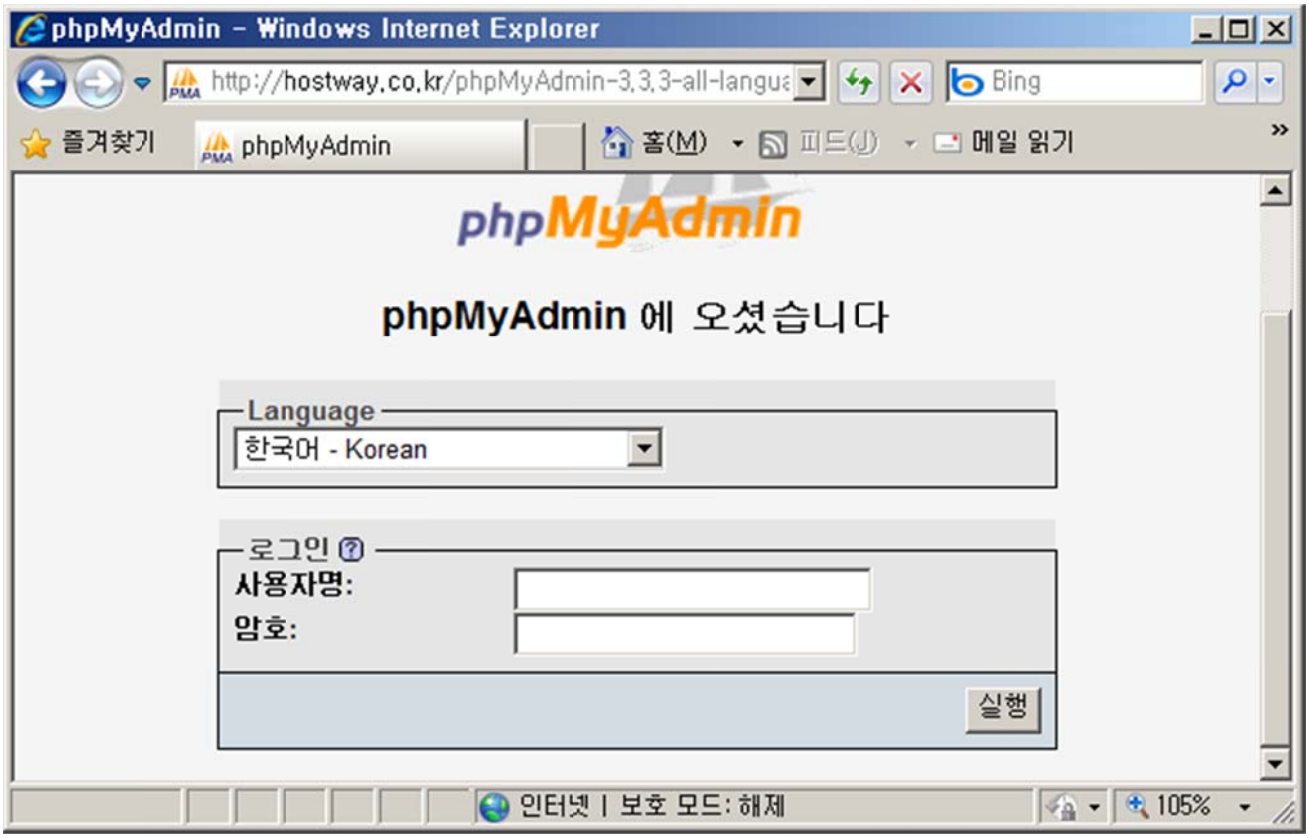
phpMyAdmin 의 최신 버전은 http://www.phpmyadmin.net 에서 다운받을 수 있습니다. 다운받은 압축 파일을 ftp 를 이용해서 홈 디렉토리 아래 public_html 디렉토리에 업로드 합니다.
이제부터는 홈디렉토리의 주인인 hostway 계정으로 ssh 접속을 하여 작업을 진행하게 됩니다.

```
root@local:/
[hostway@localhost /]$ tar zxvf phpMyAdmin-3.3.3-all-languages.tar.gz
  ~
[hostway@localhost /]$ ls -l
total 11504
drwxr-xr-x 11 hostway hostway      4096 May 11 01:32 phpMyAdmin-3.3.3-all-languages
-rw-r--r--  1 hostway hostway 4698826 May 11 01:36 phpMyAdmin-3.3.3-all-languages.tar.gz
-rw-rw-r--  1 hostway hostway 4024932 Apr  6 04:08 textcube-1.8.3.1-core.tar.gz
-rw-r--r--  1 hostway hostway 3024379 Jun 14 11:06 xe.1.4.2.3.zip
[hostway@localhost /]$
```

위와 같이 압축을 풀고 나면, phpMyAdmin-3.3.3-all-languages 디렉토리가 생성 됩니다. phpMyAdmin 설치를 완료하기 위해서는 웹 브라우저를 통해 아래와 같이 phpMyAdmin-phpMyAdmin-3.3.3-all-languages 디렉토리로 접속을 합니다.

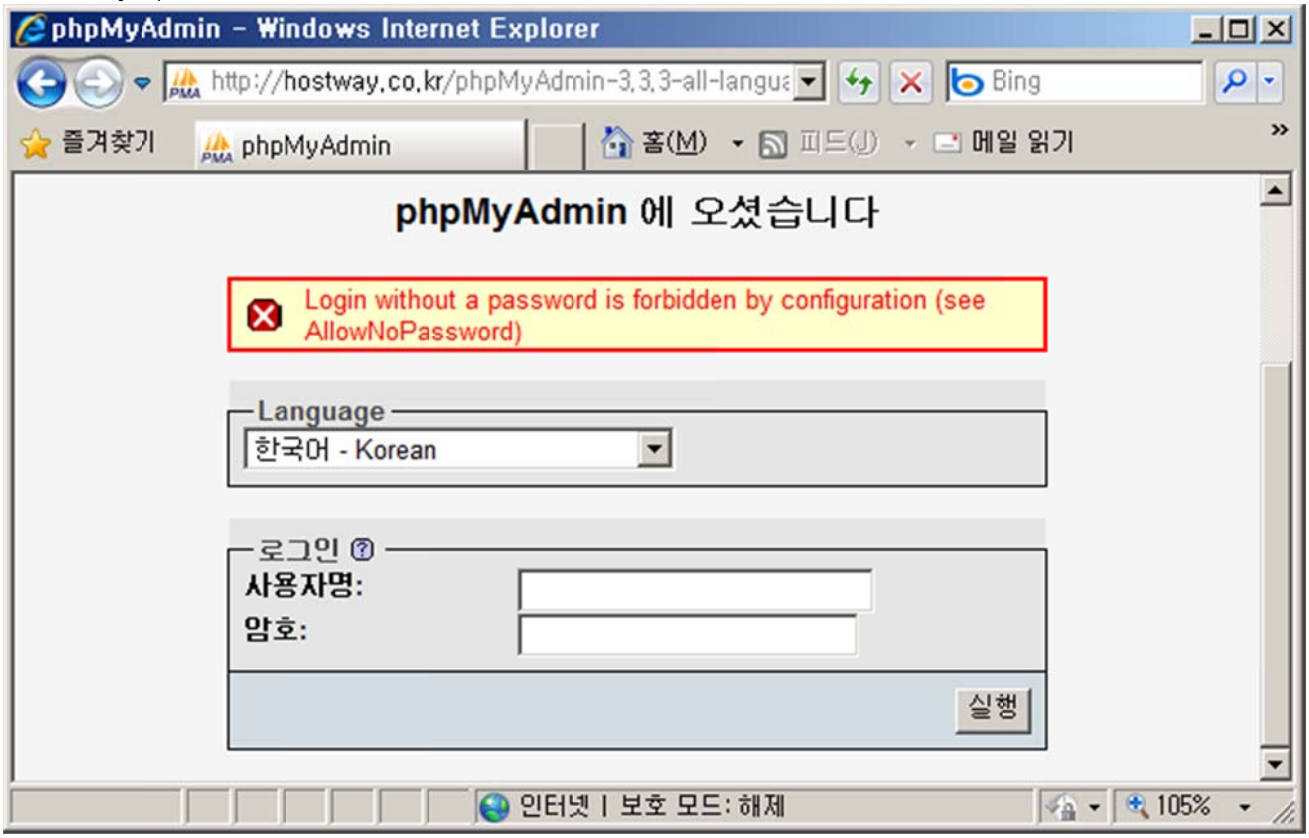
2.2 웹브라우저에서 phpMyAdmin 설정하기

phpMyAdmin 의 버전이 3.x 버전으로 높아지면서 기존 설치 방식과는 다소 달라진 부분이 있으니 이점 참고하시어 운영하시기 바랍니다.



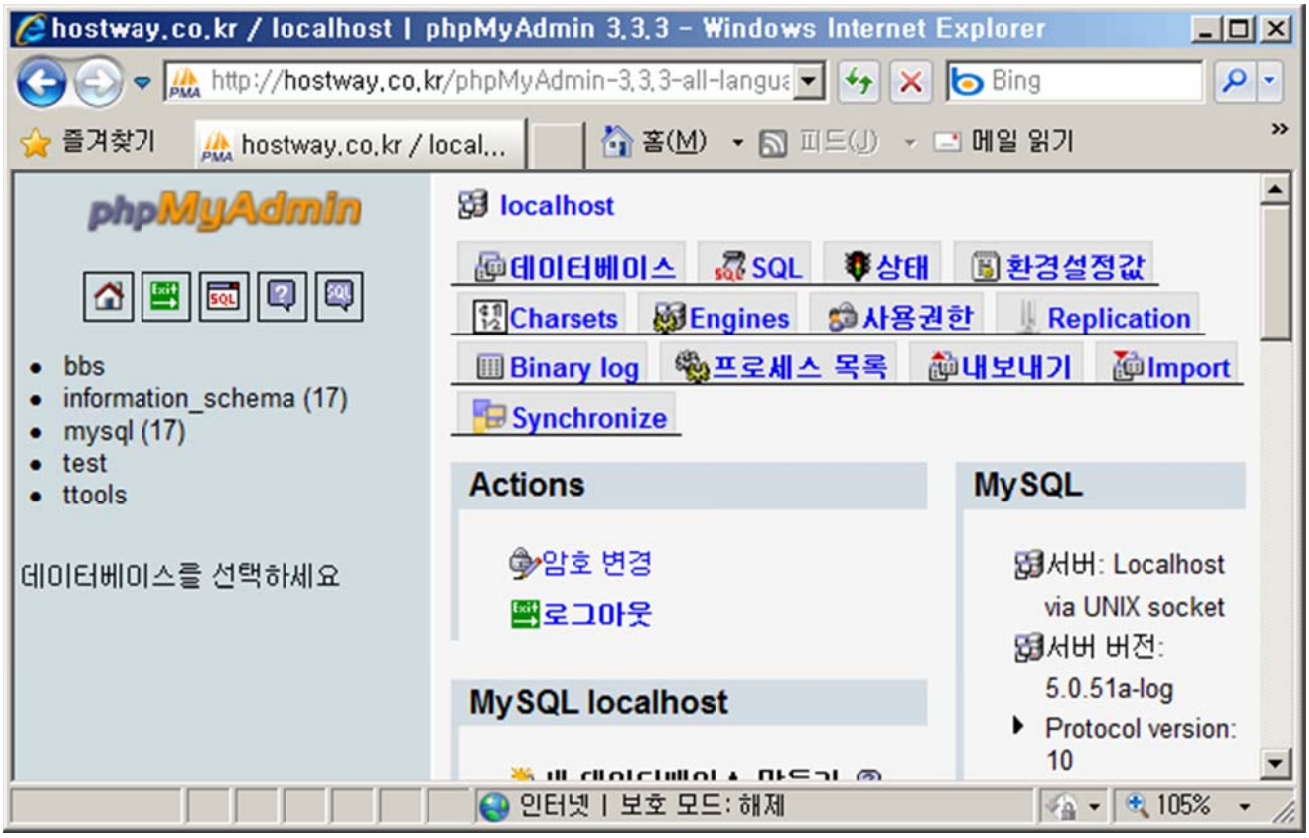
[그림 11-1] phpMyAdmin 설치

처음 설치 후 phpMyAdmin 에 접속을 하면, 위와 같은 페이지가 출력됩니다. 사용자 명과 암호 입력 부분에 mysql root 접속 정보를 입력합니다.
이때, mysql root 패스워드가 지정되지 않은 경우 아래와 같은 에러 메시지가 발생합니다.
따라서 mysql root 패스워드를 설정해 놓고 사용할 것을 권장합니다.



[그림 11-2] phpMyAdmin 접속

정상적으로 mysql root 패스워드가 세팅되어 있는 경우 로그인 완료 시 아래와 같이 정상 접속된 상태를 확인 할 수 있습니다.



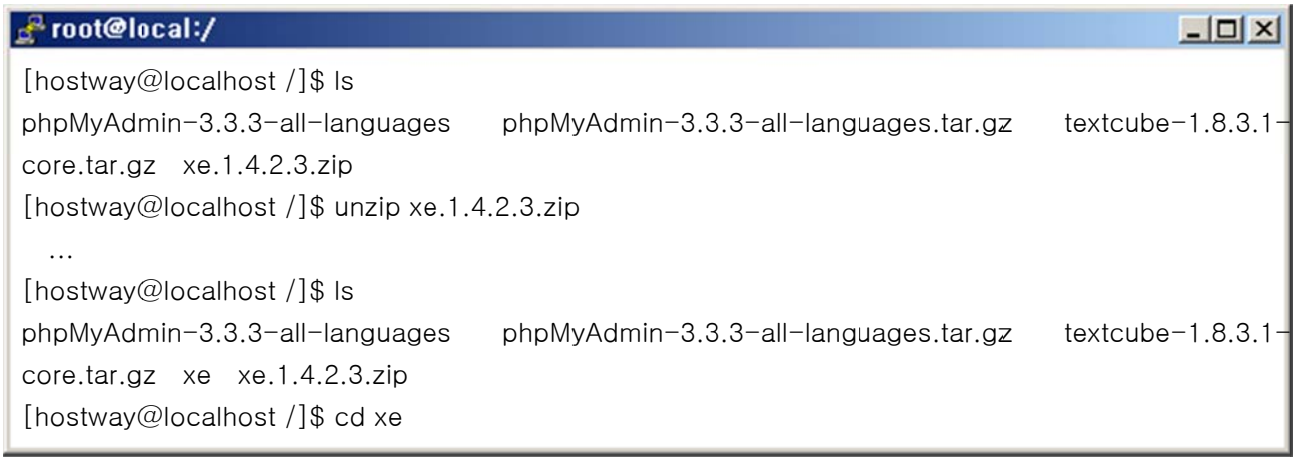
[그림 11-3] phpMyAdmin 설치 완료

이로써 phpMyAdmin 설치가 완료 되었습니다.
[그림]과 같이 익스플로러에서 http://도메인 phpMyAdmin 설치 경로의 주소로 접속하여 관리할 데이터베이스를 선택하고 사용하면 됩니다.

3. 제로보드 설치

3.1 Source 파일 다운로드 및 압축 풀기

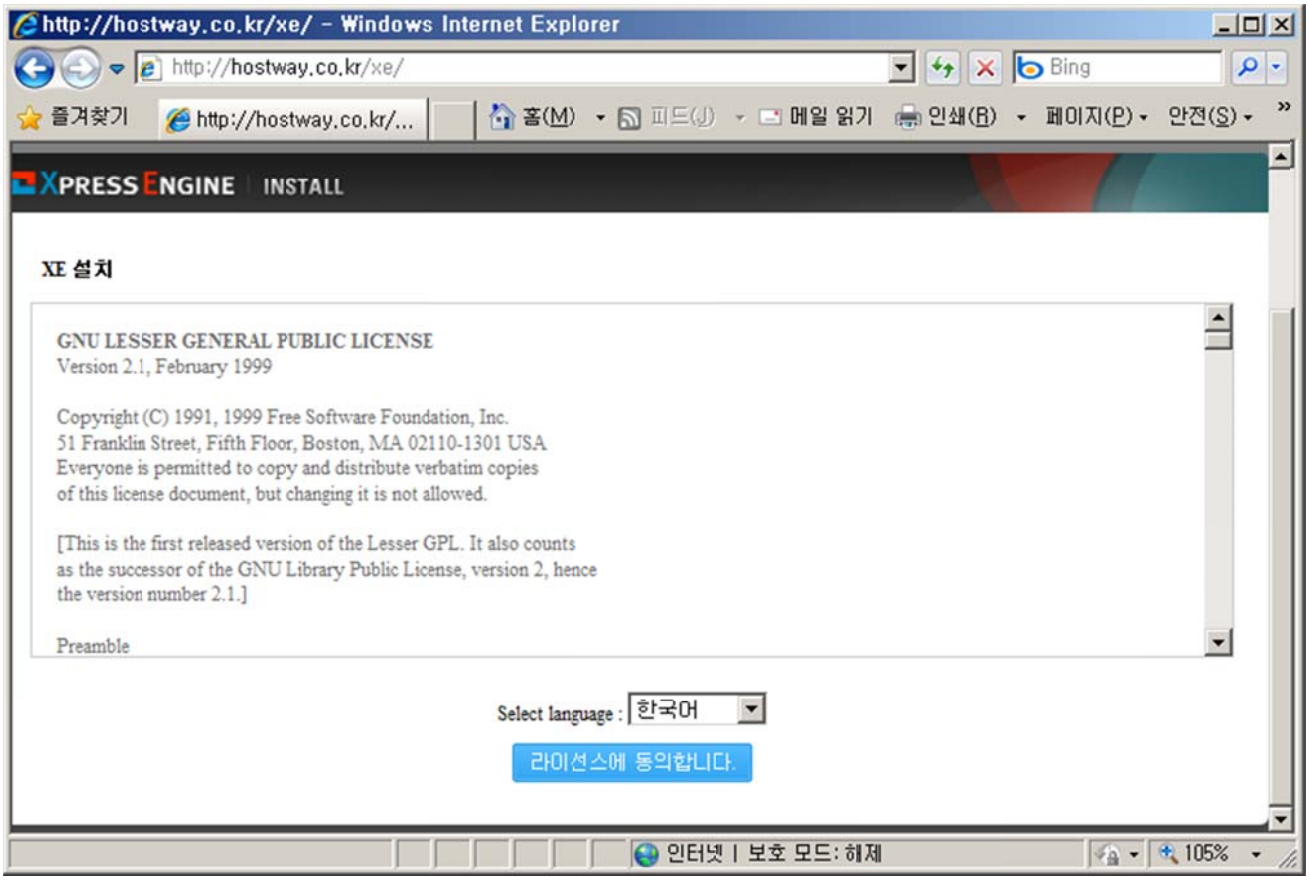
제로보드의 source 파일은 http://www.nzeo.com 사이트에서 다운로드 받을 수 있습니다. phpMyAdmin 과 마찬가지로 다운로드 받은 파일을 홈 디렉토리 하위의 public_html 디렉토리에 업로드 합니다.



위와 같이 압축을 풀고나면, xe 디렉토리가 생기게 됩니다.

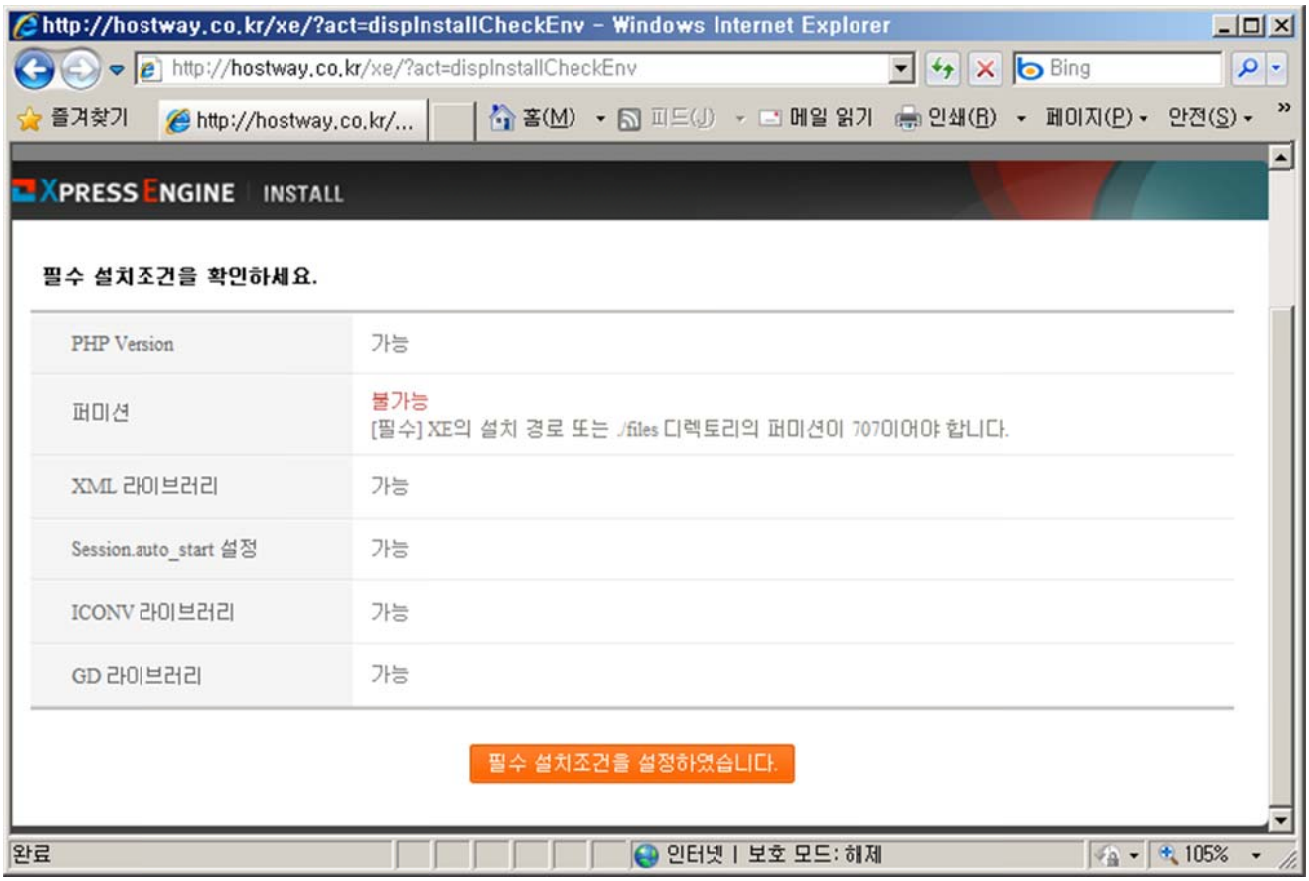
3.2 웹브라우저에서 제로보드 설정하기

홈디렉토리 하위에 xe 디렉토리가 생성되었기 때문에 아래와 같이 “http://hostway.co.kr/xe/” 로 웹 접속을 합니다.
아래는 제로보드 설치 초기화면 입니다.



[그림 11-4] 제로보드 설치

언어 선택 및 라이선스에 동의합니다.

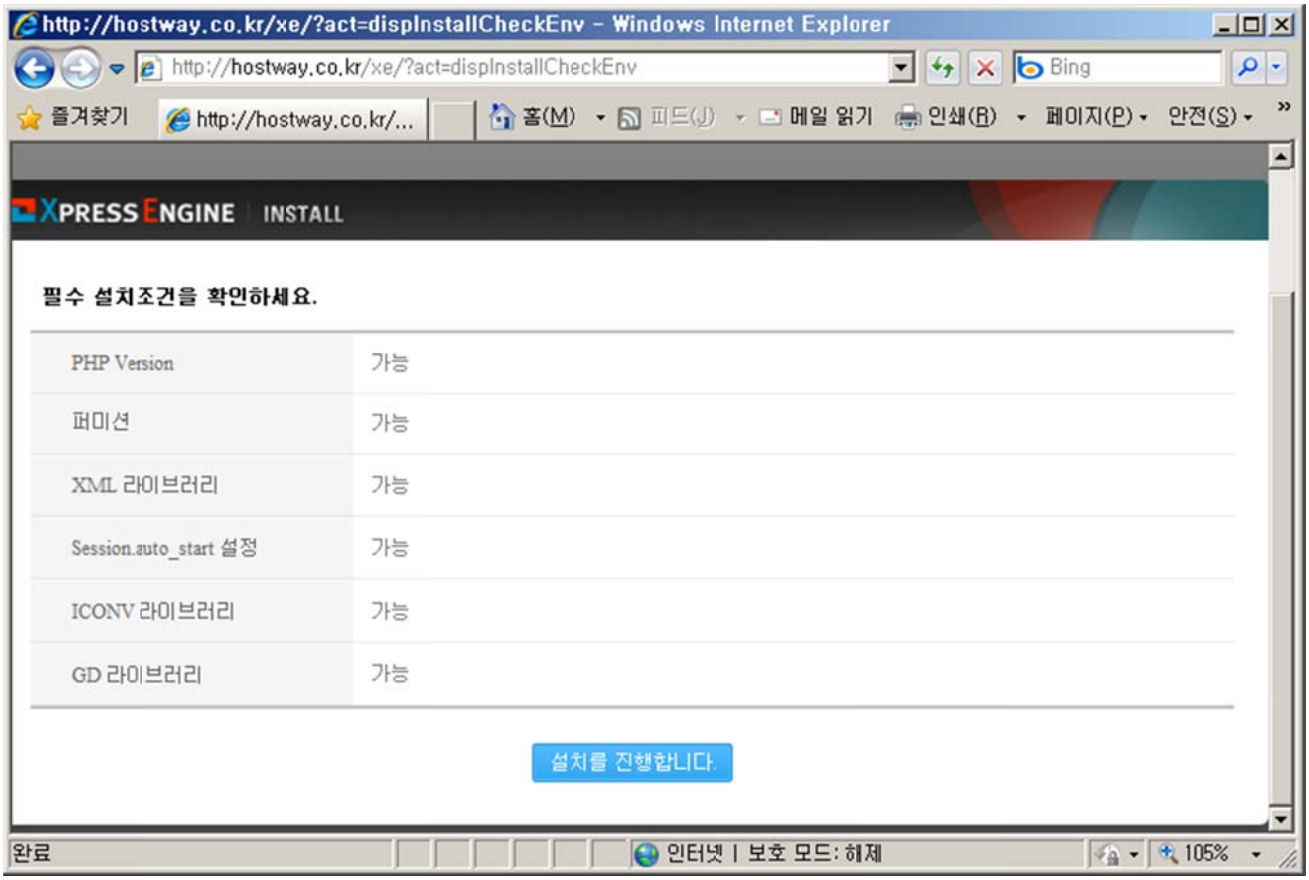


[그림 11-5] 제로보드 필수 설치 조건 확인

위와 같이 필수 설치 조건 중 설치 디렉토리의 퍼미션이 올바르지 않다면 shell 로 돌아가서 xe 디렉토리의 퍼미션을 707 로 변경하고 다음단계로 넘어갑니다.

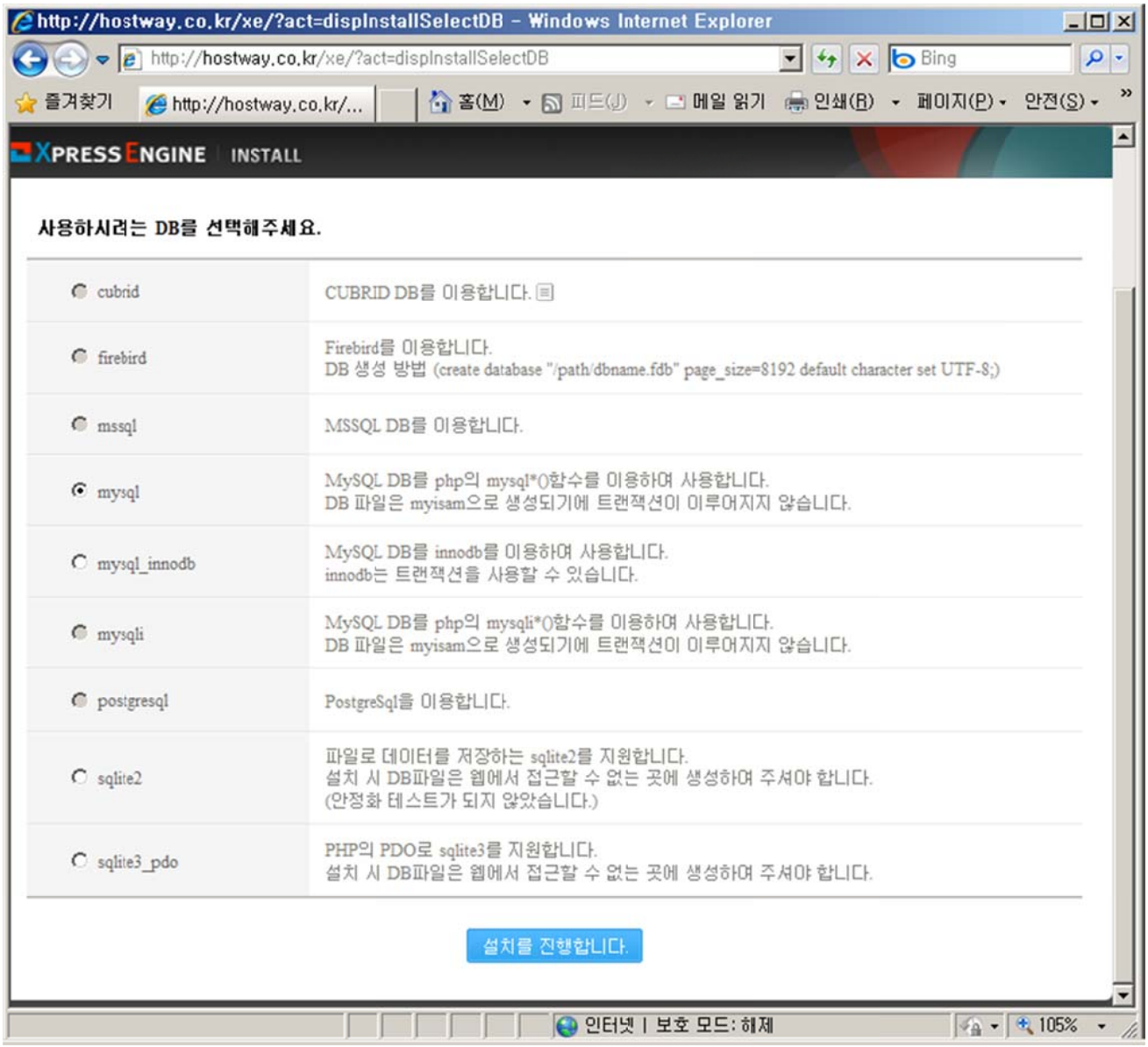
```
root@local:/
[hostway@localhost /]$ chmod 707 xe
[hostway@localhost ~]$ ls -l
total 11508
drwxr-xr-x 11 hostway hostway      4096 Jun 14 12:48 phpMyAdmin-3.3.3-all-languages
-rw-r--r--  1 hostway hostway 4698826 May 11 01:36 phpMyAdmin-3.3.3-all-languages.tar.gz
-rw-rw-r--  1 hostway hostway 4024932 Apr  6 04:08 textcube-1.8.3.1-core.tar.gz
drwx---rwx 11 hostway hostway      4096 Jun 14 13:12 xe
-rw-r--r--  1 hostway hostway 3024379 Jun 14 11:06 xe.1.4.2.3.zip
[hostway@localhost /]$
```

퍼미션을 조정한 후 페이지 새로 고침을 하면 아래와 같이 필수 설치 조건이 모두 가능으로 변경됩니다. 이제 설치를 진행합니다.



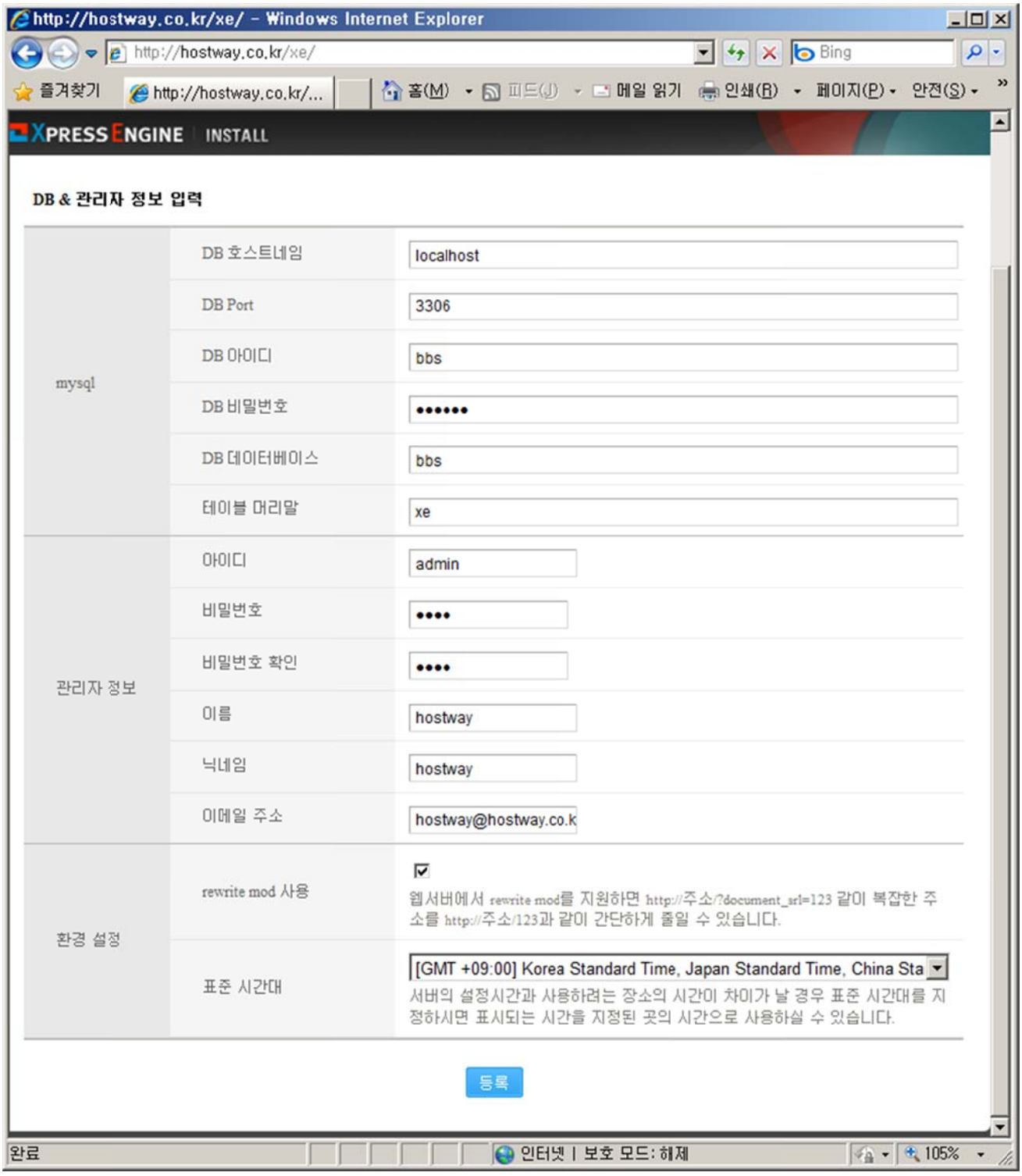
[그림 11-6] 제로보드 필수 설치 조건 확인 완료

사용하려는 DB를 선택 후 설치를 진행합니다. 이때는 사전에 설치 및 계정 등 설정 해 둔 Default Mysql 로 선택합니다.



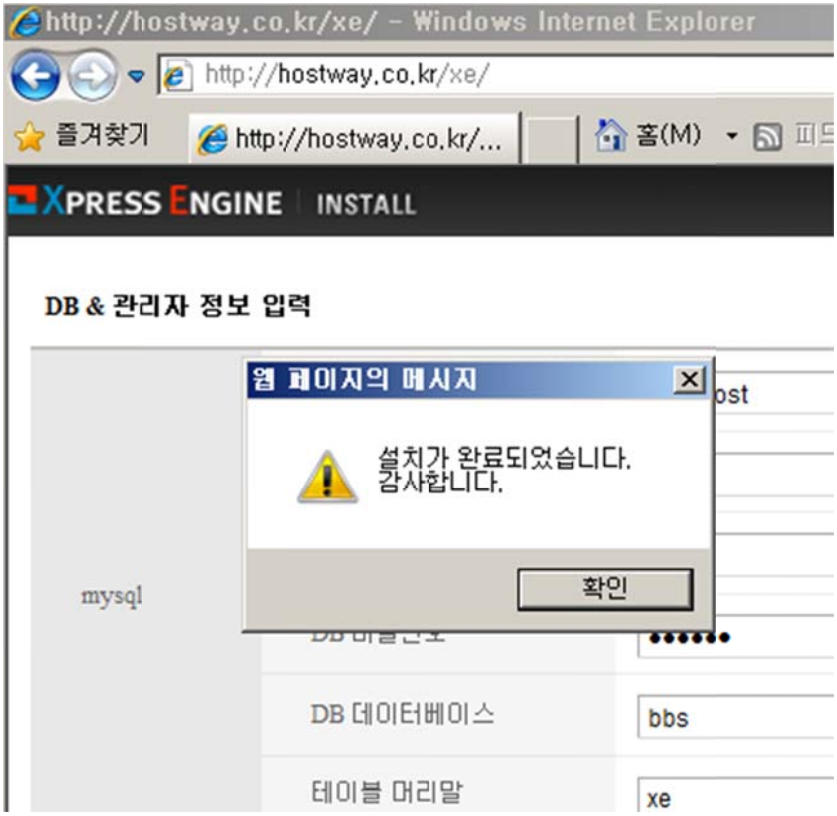
[그림 11-7] 제로보드 데이터베이스 선택

아래의 페이지는 제로보드에서 사용할 DB 및 관리자 정보를 입력하는 페이지 입니다.
빠짐 없이 올바르게 입력한 후 [등록]을 클릭합니다.



[그림 11-8] 제로보드 DB & 관리자 정보 입력

이제 설치가 모두 완료 되었습니다.



[그림 11-9] 제로보드 설치 완료

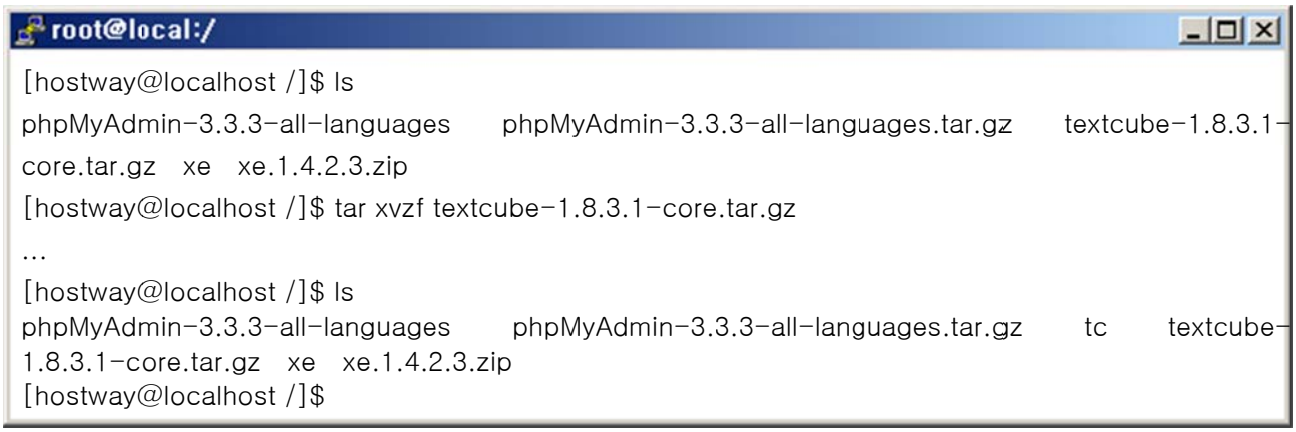
이제 완료 팝업의 확인을 클릭한 후 페이지를 새로 고침하면 관리자 페이지 로그인 화면이 출력됩니다.
제로보드 설치가 완료 되었으므로 제로보드 사용 매뉴얼을 확인 하고 그룹 생성 및 게시판을 생성하여 사용하면 됩니다.

4. Textcube

4.1 Source 파일 다운로드 및 압축 풀기

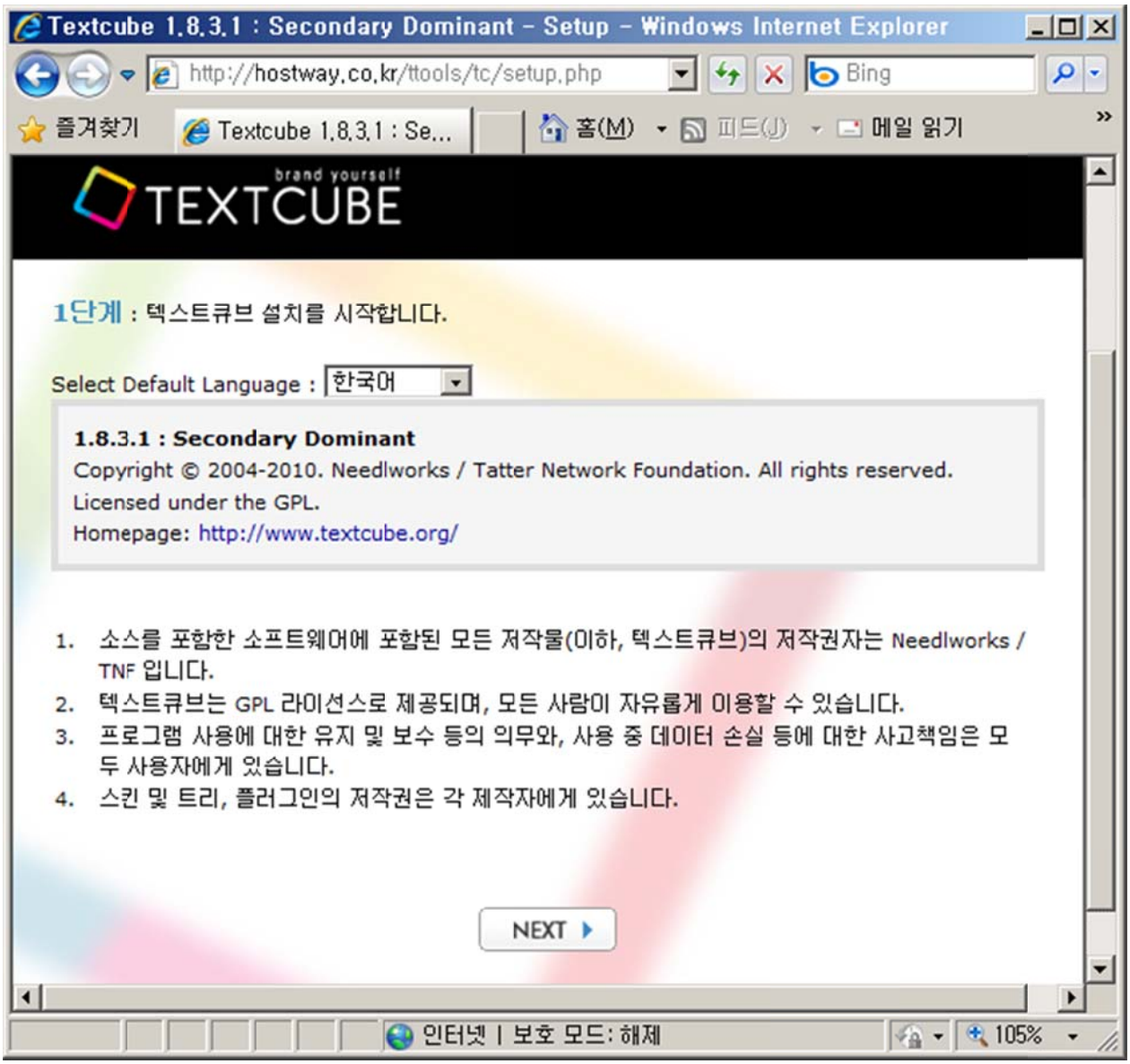
textcube 설치방법도 제로보드 설치방법과 거의 비슷합니다. 우선 http://tattertools.com 에서 textcube 의 소스파일을 다운로드 받고, ftp 를 이용하여, 홈 디렉토리내의 public_html 디렉토리에 업로드 합니다.

파일 업로드가 완료되면 ssh 로 접속하여 아래와 같이 textcube 디렉토리에서 사용할 디렉토리를 생성 후 압축을 풀고 웹상에서 설치를 진행하게 됩니다.



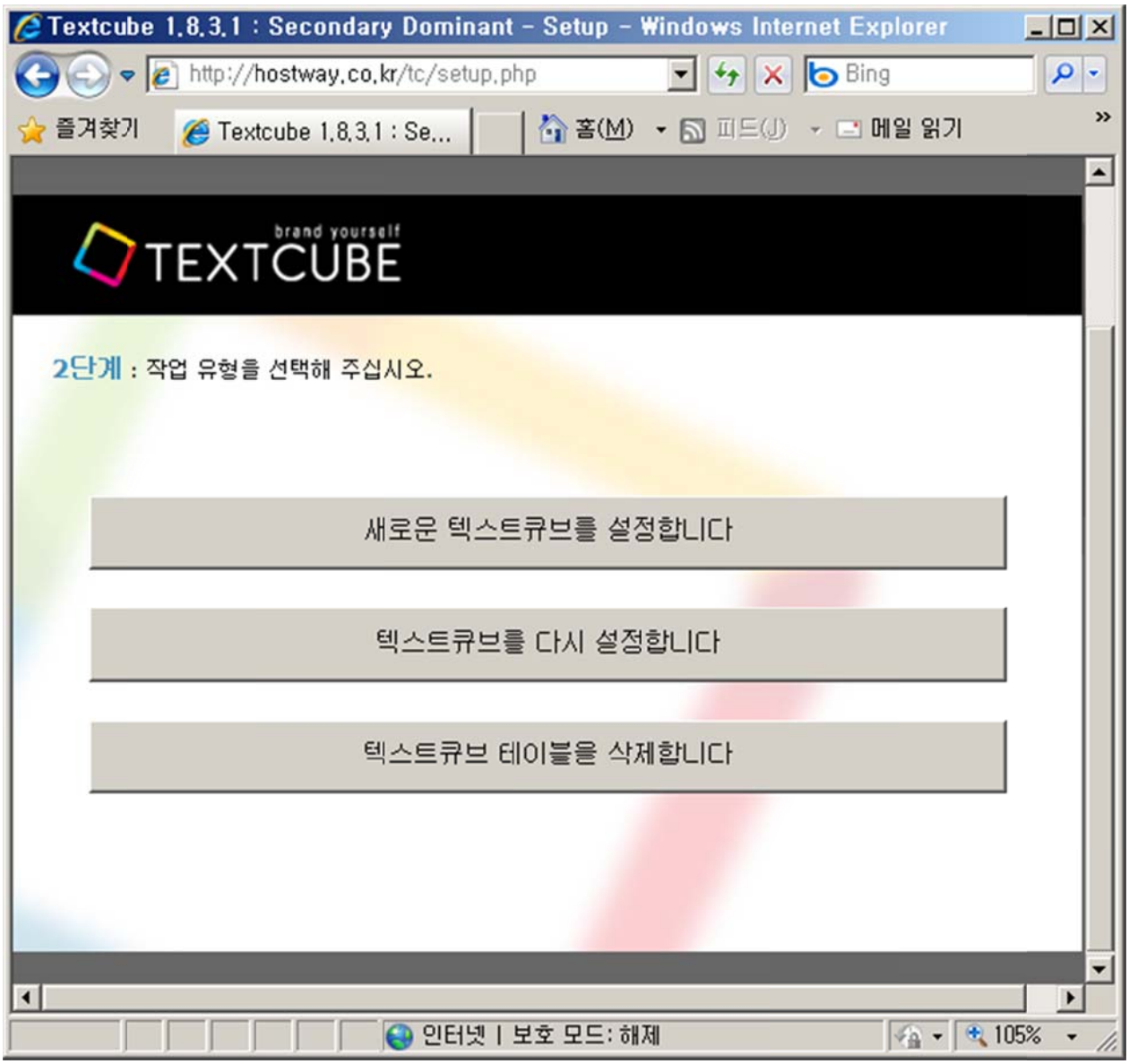
4.2 웹브라우저에서 textcube 설정하기

웹브라우저를 이용해서 textcube 소스 디렉토리에 접속합니다.
http://hostway.co.kr/tc/ 로 접속하면 아래와 같이 textcube 1단계 설치 화면이 출력됩니다.



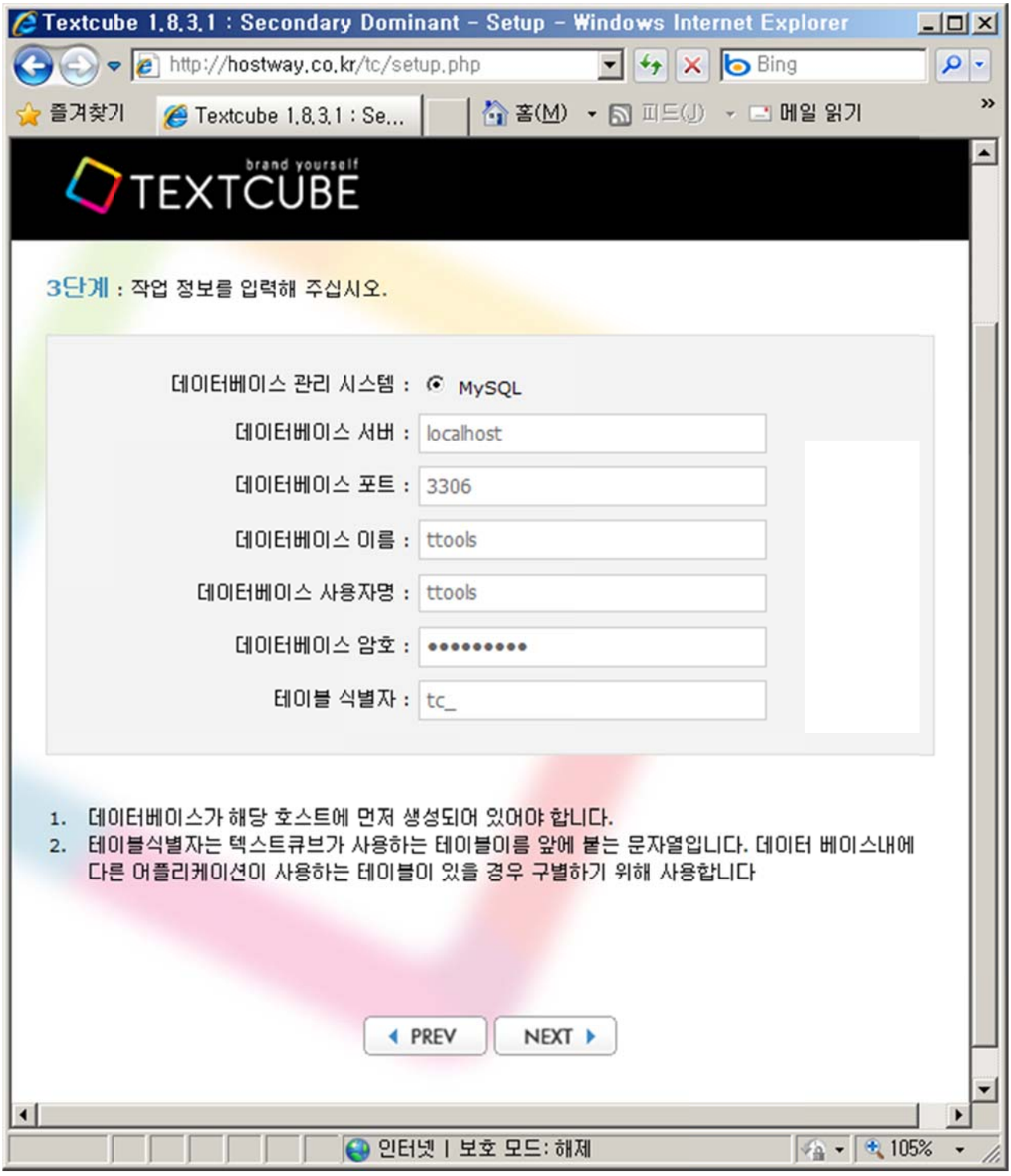
[그림 11-10] 텍스트큐브 설치

1단계 확인 후 [Next] 클릭하면 작업 유형 선택 페이지가 출력됩니다.
Textcube 를 초기 설치하고 있으므로 [새로운 텍스트큐브를 설정합니다]를 클릭하여 다음 페이지로 이동합니다.



[그림 11-11] 텍스트큐브 작업 유형 선택

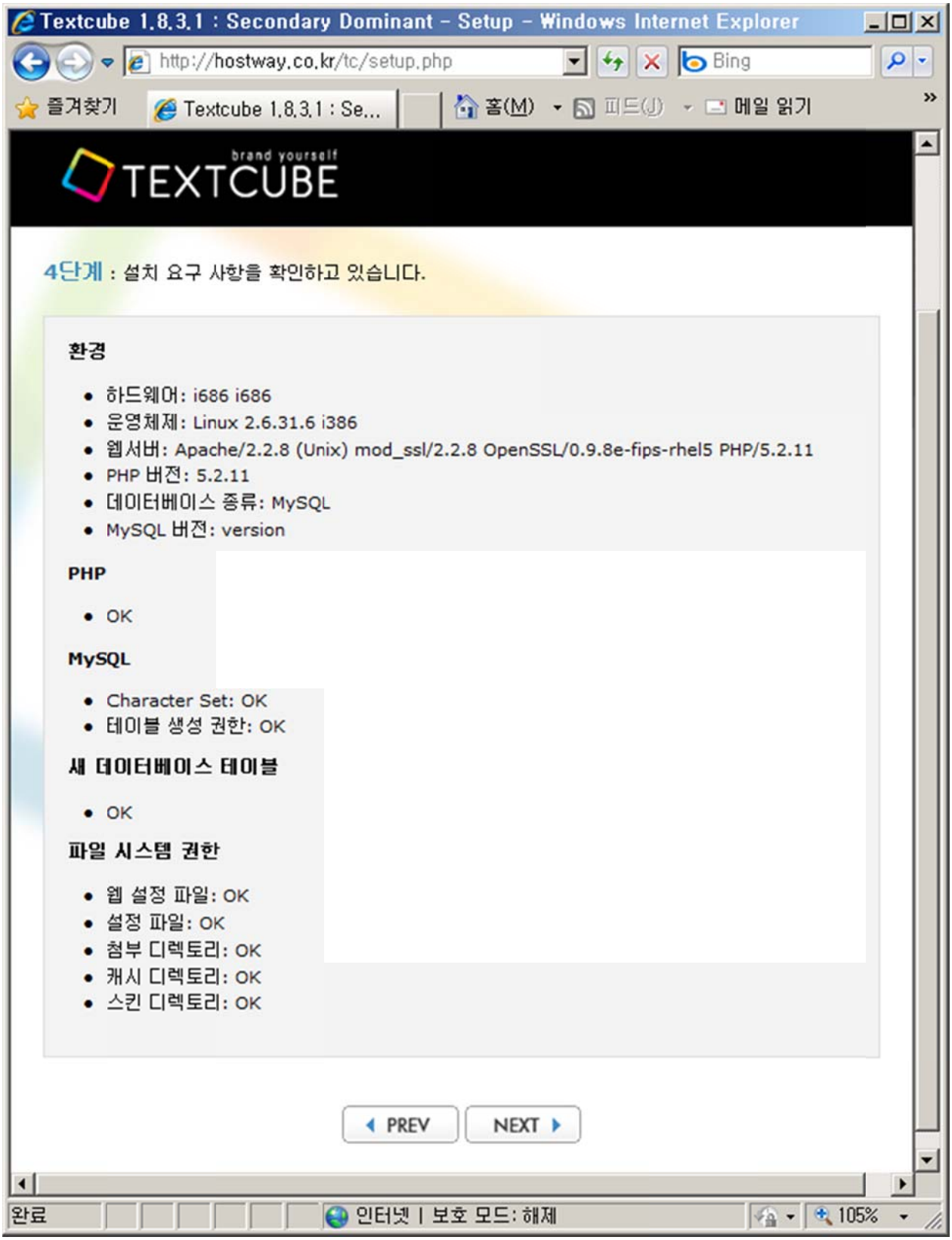
3 단계 textcube 에서 사용할 작업 정보를 입력합니다.
초기 생성해 놓은 DB 접속 정보를 입력한 후 [Next]를 클릭합니다.



[그림 11-12] 텍스트큐브 작업 정보 입력

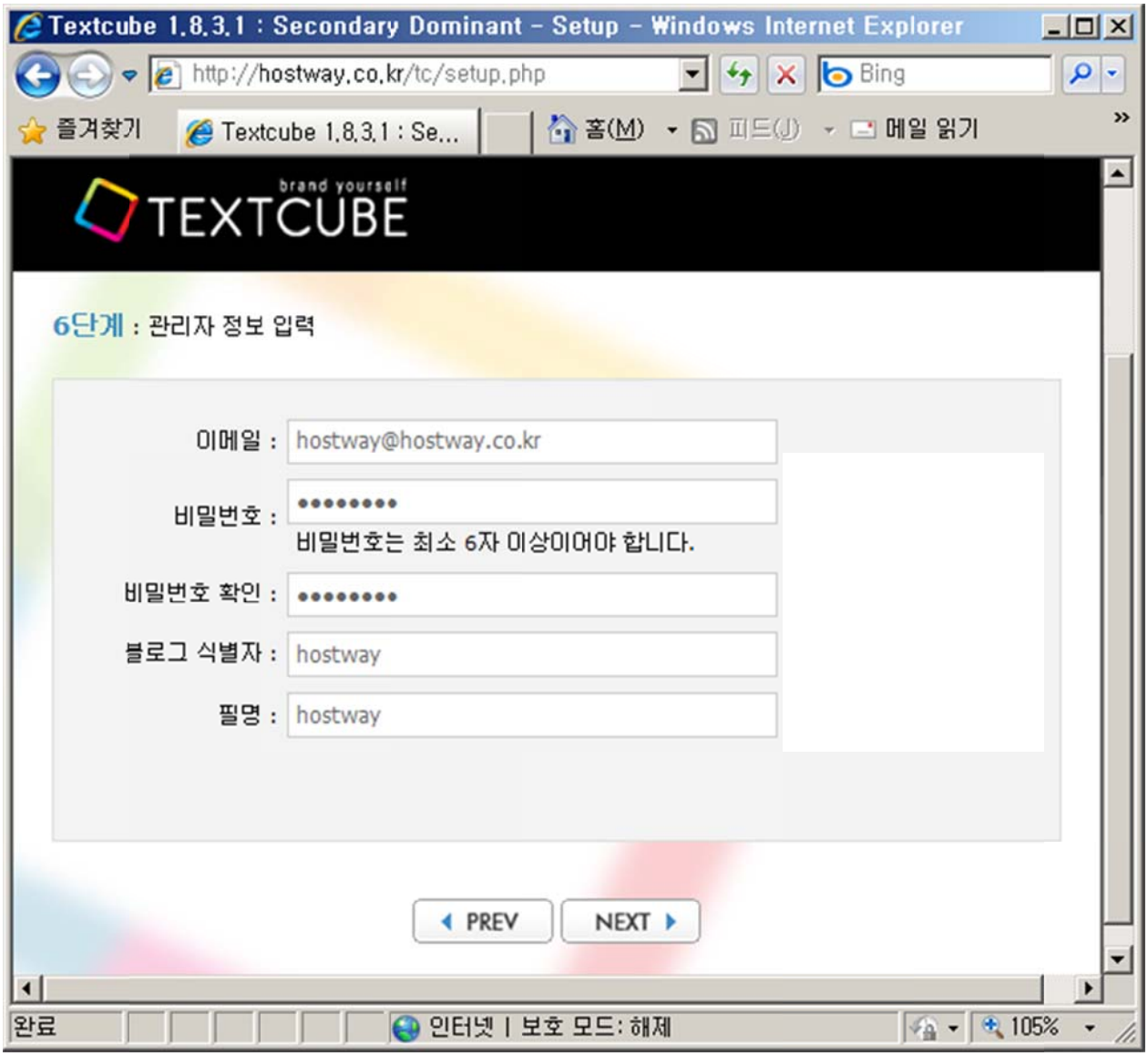
4 단계 설치 요구 사항 확인 단계에서는 하드웨어와 설치 환경 그리고 php 버전, 데이터 베이스, 설치 디렉토리의 퍼미션 등을 확인합니다.
이 때 tc 등의 디렉토리가 퍼미션이 777 이 아닌 경우 에러가 발생할 수 있습니다.
출력되는 메시지를 확인하여 shell 모드에서 설정해 주면 쉽게 설치가 가능합니다.

또한 apache 의 rewrite 모듈 설정이 되어 있지 않은 경우는 이번 단계에서 진행이 되지 않게 됩니다. 그러나 해당 모듈 기능 비활성화를 통해 무시 하겠습니다. 설치 후 사용 가능하도록 설정이 가능하며 빠른 설치를 위해서 입니다.



[그림 11-13] 텍스트큐브 설치 요구 사항 확인

4 단계 설치 요구 사항이 모두 확인되고 Next 를 클릭하면 5 단계 사용 가능한 운영 방법 페이지가 출력됩니다.
잘 읽어 보고 [Next]를 클릭합니다.



[그림 11-14] 텍스트큐브 관리자 정보 입력

이제 설치가 거의 다 완료 되었습니다. 관리자 정보를 입력하신 후 [Next]를 클릭합니다.
단, 패스워드가 6 자리 이상으로 입력해야만 다음으로 진행이 가능합니다. 이점 주의해 주시기 바랍니다.

[그림 11-15] 텍스트큐브 설치 완료

이제 설치가 모두 완료 되었습니다.

Chapter 12. 장애복구

1. web 서버 장애

- 홈페이지 속도가 느려지거나, 접속이 되지 않는 경우

```
root@local:/
[root@localhost /]# ls -alh /usr/local/apache2/logs
합계 2G
drwxr-xr-x     2 root      root          4.0K  6 월  26 18:02 .
drwxr-xr-x    15 root      root          4.0K  6 월  12 19:03 ..
-rw-r--r--     1 root      root          1.5K  6 월  14 13:33 access_log
-rw-r--r--     1 root      root          4.5K  6 월  26 19:16 error_log
-rw-r--r--     1 root      root            2G   6 월  26 19:16 hostway.co.kr-access_log
-rw-r--r--     1 root      root          3.7K  6 월  16 17:54 hostway.co.kr-error_log
-rw-r--r--     1 root      root             0  6 월  26 19:16 httpd.pid
[root@localhost /]# pkill httpd
[root@localhost /]# cat /dev/null > /usr/local/apache2/logs/hostway.co.kr-access_log
[root@localhost /]# /usr/local/apache2/bin/apachectl start
```

➢ 위와 같이 로그파일의 사이즈가 2G 이상이 되면, apache 데몬이 동작하지 않거나, 홈페이지 접속이 느려지게 됩니다. 홈페이지 속도가 느려지거나, 동작하지 않는 경우는 제일먼저 apache 데몬이 떠 있는지를 확인하고, 다음으로 로그파일의 사이즈를 확인 합니다. 파일 사이즈가 2G 이상일 경우는 로그파일을 삭제한 후 apache 데몬을 리스타트 합니다.

- php 소스가 정상적으로 동작하지 않고, 홈페이지 접속 시 index 파일 다운로드 장애가 발생할 때는 httpd.conf 파일에서 php 소스 인식 부분이 정상적으로 추가 되었는지를 확인 합니다.

```
root@local:/
[root@localhost /]# cat /usr/local/apache2/conf/httpd.conf
~
    LoadModule php5_module          modules/libphp5.so
~
    AddType application/x-httpd-php .php .php3 .ph .inc .html .htm
    AddType application/x-httpd-php-source .phps
```

➢ 아래와 같이 httpd.conf 파일에서 LoadModule, AddType Application 부분이 정상적으로 추가되었는지 확인 합니다.

- apache 데몬실행시 에러 메시지 없이 데몬이 동작하지 않고, 에러로그에 아래와 같은 메시지 발생 할때

```
root@local:/
[root@localhost /]# /usr/local/apache2/bin/apachectl start
[root@localhost /]# ps ax | grep httpd
29474 pts/1    S      0:00 grep httpd
[root@localhost /]# tail /usr/local/apache2/logs/error_log
[Mon Jun 26 21:28:49 2006] [error] (13)Permission denied: could not create
/usr/local/apache2/logs/httpd.pid
[Mon Jun 26 21:28:49 2006] [error] httpd: could not log pid to file
/usr/local/apache2/logs/httpd.pid
[Mon Jun 26 21:28:53 2006] [warn] pid file /usr/local/apache2/logs/httpd.pid overwritten --
Unclean shutdown of previous Apache run?
```

➢ apache 데몬이 종료될 때, 정상적으로 종료되지 않고 httpd.pid 파일이 삭제되지 않고, 남아 있는 경우, apache 데몬을 재시작할 때 에러가 발생 합니다. 이런 경우는 아래와 같이,/usr/local/apache2/logs 디렉토리에서 httpd.pid 파일을 강제로 삭제한 후 데몬을 리스타트 합니다.

```
root@local:/
[root@localhost /]# ls /usr/local/apache2/logs
access_log  error_log  hostway.co.kr-access_log  hostway.co.kr-error_log  httpd.pid
[root@localhost /]# rm -rf /usr/local/apache2/logs/httpd.pid
[root@localhost /]# /usr/local/apache2/bin/apachectl start
[root@localhost /]# ps ax | grep httpd
29498 ?        S      0:00 /usr/local/apache2/bin/httpd -k restart
29499 ?        S      0:00 /usr/local/apache2/bin/httpd -k restart
29500 ?        S      0:00 /usr/local/apache2/bin/httpd -k restart
29501 ?        S      0:00 /usr/local/apache2/bin/httpd -k restart
29502 ?        S      0:00 /usr/local/apache2/bin/httpd -k restart
29503 ?        S      0:00 /usr/local/apache2/bin/httpd -k restart
29505 pts/1    S      0:00 grep httpd
[root@localhost /]#
```

- 가상호스트를 추가하고 홈페이지에 접속 했을때, 아래와 같이 forbidden 에러가 발생하는 경우

```
root@local:/
[root@localhost /]# ls -l /home
합계 32
drwxr-xr-x     6 root       root           4096  6월 26 12:53 data
drwx------     5 hostway    hostway        4096  6월 19 07:43 hostway
drwx------     2 root       root          16384  5월 15 02:25 lost+found
[root@localhost /]# chmod 701 /home/hostway
[root@localhost /]#
```

  - 이 경우는 useradd 명령으로 계정을 생성하고 가상호스트 설정을 했을 때, 자주 발생하는 에러입니다. useradd 명령으로 계정을 생성하면, 기본적으로 홈디렉토리의 퍼미션이 700 으로 생성되게 됩니다. 웹상에서 접근을 하려면 other 권한으로 홈디렉토리 안으로 이동할 수 있어야 하는데, 권한이 없기 때문에 에러가 발생하게 됩니다. 아래와 같이 chmod 명령으로 홈 디렉토리의 접근 권한을 755 나 751, 701 등으로 변경하여, other 에 실행권한을 부여하면 정상적으로 접속이 가능하게 됩니다.

2. mail 서버 장애

메일사용시의 장애는 주로 아웃룩을 이용해서 메일을 송 수신할 때, 발생하게 됩니다. 메일서버의 장애 유형은 아웃룩의 에러메시지로 언급 하도록 하겠습니다.

- 에러 문구

> 메일 서버에 로그온하는 데 문제가 있습니다. 지정한 암호가 거부되었습니다. 계정: 'hostwaykorea', 서버: 'mail.hostway.co.kr', 프로토콜: POP3, 서버 응답: '-ERR Bad login', 포트: 110, 보안(SSL): 아니오, 서버 오류: 0x800CCC90, 오류 번호: 0x800CCC92

  - 메일을 수신 하려는 계정의 패스워드가 틀려서 발생하는 에러입니다. 서버상에서 계정의 패스워드를 변경하거나, 올바른 패스워드를 입력해서 메일을 수신 합니다.

- 에러 문구

> 'mail.hosway.co.kr' 호스트를 찾을 수 없습니다. 서버 이름을 올바르게 입력했는지 확인하십시오. 계정: 'webmaster', 서버: 'mail.hosway.co.kr', 프로토콜: POP3, 포트: 110, 보안(SSL): 아니오, 소켓 오류: 11001, 오류 번호: 0x800CCC0D
> 서버에 연결할 수 없습니다. 계정: 'mail.hostway.co.kr.'

  - 아웃룩상에 계정을 설정할 때, pop3 이름에 오타가 발생해서 나타나는 에러입니다. 아웃룩 상의 설정을 확인해서 수정합니다.

- 에러문구

받는 사람 중 한 사람이 서버에서 거부되었으므로 메시지를 보낼 수 없습니다. 거부된 전자 메일 주소는 'hostway@hanmail.net. 제목 '테스트 메일', 계정: 'hostway', 서버: 'mail.hostway.co.kr', 프로토콜: SMTP, 서버 응답: '553 RCPT <hostway@hanmail.net> ERROR. Relaying not allowed', 포트: 25, 보안(SSL): 아니오, 서버 오류: 553, 오류 번호: 0x800CCC79

➢ 메일버서에서 릴레이 허용이 되어 있지 않아서 발생하는 에러입니다. smtp 서버 설정을 확인해서 auth-smtp 설정을 완료한 후에 메일을 발송합니다.

- 에러문구

"Returned mail: Service unavailable"이면서 메일내 "Can't create output: Error 0" 또는 "user_id: mbox is full!!"

➢ 보내려는 계정의 메일박스가 가득차서 더 이상 메일을 수신할수 없어서 반송된 메시지입니다. 수신하려는 계정의 메일박스를 비워서 메일저장공간을 확보해야 합니다.

- 에러문구

"Returned mail: User unknown" 또는 "Returned mail: no such user"

➢ 서버상에 계정이 존재하지 않아서, 반송된 메시지입니다. 계정이름을 정확히 입력했는지 확인하고, 서버상에 계정이 생성되지 않았다면, 계정을 생성 합니다.

3. mysql 장애

- /usr/local/mysql/var 디렉토리의 소유권이 mysql 로 설정되어 있지 않아서, 데몬이 실행되지 않을때.

➢ 위의 에러는 mysql 의 데이터가 저장되는 /usr/local/mysql/var 디렉토리에 pid 파일을 생성하지 못해서 발생하는 에러입니다. /usr/local/mysql/var 디렉토리의 소유권이 root 로 되어 있거나, /usr 파티션의 사용량이 100%는 아닌지 확인 합니다. var 디렉토리의 소유권이 root 로 되어 있을 경우는 아래와 같이 소유권을 변경한 후에 데몬을 실행 합니다.

```
root@local:/
[root@localhost /]# ls -l /usr/local/mysql
합계 40
drwxr-xr-x     2 root      root          4096  6월 12 18:56 bin
drwxr-xr-x     3 root      root          4096  6월 12 18:55 include
drwxr-xr-x     2 root      root          4096  6월 12 18:55 info
drwxr-xr-x     3 root      root          4096  6월 12 18:55 lib
drwxr-xr-x     2 root      root          4096  6월 12 18:57 libexec
drwxr-xr-x     3 root      root          4096  6월 12 18:56 man
drwxr-xr-x     7 root      root          4096  6월 12 18:57 mysql-test
drwxr-xr-x     3 root      root          4096  6월 12 18:56 share
drwxr-xr-x     5 root      root          4096  6월 12 18:56 sql-bench
drwx------     6 root      root          4096  6월 26 18:11 var
[root@localhost /]# chown -R mysql.mysql /usr/local/mysql/var
[root@localhost /]# /usr/local/mysql/bin/mysqld_safe &
[1] 966
```

● rpm 버전의 mysql 이 설치되어있어서 mysql 데몬이 실행되지 않을때.

```
root@local:/
[root@local /]# /usr/local/mysql/bin/mysqld_safe &
[1] 5058
[root@local /]# Starting mysqld daemon with databases from /var/lib/mysql
STOPPING server from pid file /var/run/mysqld/mysqld.pid
100503 21:42:04  mysqld ended
[1]+  Done                    /usr/local/mysql/bin/mysqld_safe
```

➢ 위의 경우는 mysql 데몬 실행시 pid 파일을 /var/lib/mysql 디렉토리에 생성하려고 하다가 실패한 모습입니다. rpm 버전의 mysql 이 설치되어 있는 경우 이러한 에러가 발생합니다. 아래와 같이 rpm -qa | grep mysql 로 rpm 버전의 mysql 을 검색하여, rpm -e --nodeps 명령으로 mysql 과 관련된 rpm 패키지를 모두 삭제한 후에 mysql 데몬을 실행 합니다.

```
root@local:/
[root@localhost /]# rpm -qa | grep mysql
mysql-5.0.77-4.el5_5.3
[root@localhost /]# rpm -e --nodeps mysql-5.0.77-4.el5_5.3
[root@localhost /]# /usr/local/mysql/bin/mysqld_safe &
[1] 917
[root@localhost /]# Starting mysqld daemon with databases from /usr/local/mysql/var
[root@localhost /]#
```

- mysql root 패스워드를 분실했을 때

```
root@local:/
[root@localhost /]# /usr/local/mysql/bin/mysqladmin -u root -p shutdown
Enter password:
STOPPING server from pid file /usr/local/mysql/var/localhost.pid
100520 02:03:01  mysqld ended
[1]+  Done                    /usr/local/mysql/bin/mysqld_safe
```

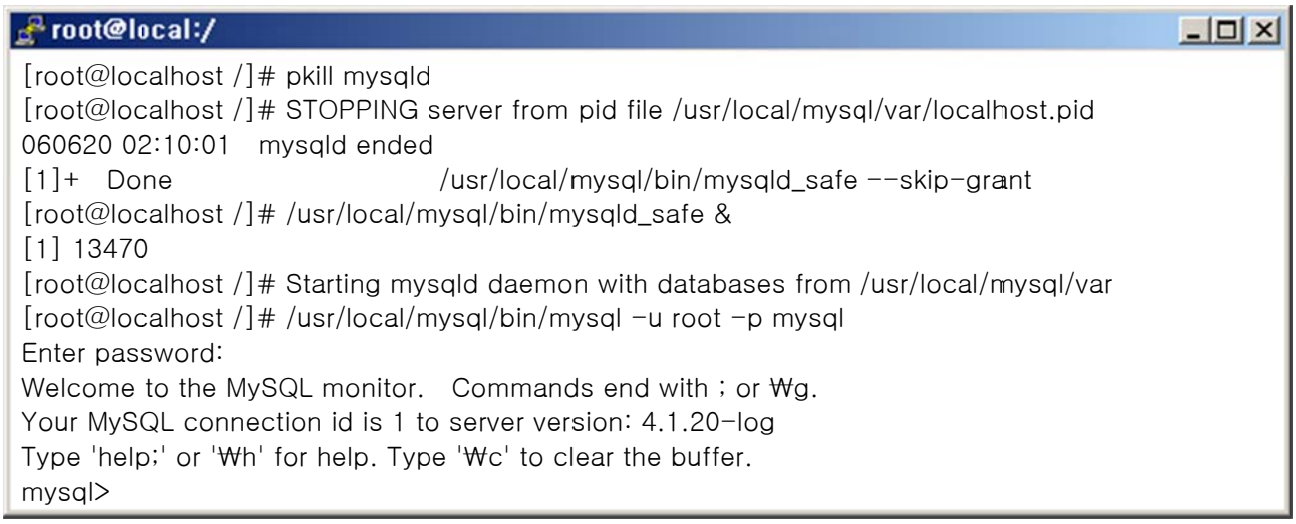
➢ mysql root 계정의 패스워드를 분실 했을 경우는 아래와 같은 순서로 변경할 수 있습니다.
mysql 데몬을 중지 시킵니다.
권한 테이블을 사용하지 않는 옵션으로(--skip-grant) 데몬을 실행시킵니다.
이 상태에서는 권한 테이블을 사용하지 않으므로 어떤 호스트에서도 아무 사용자로 모든 DB 에 접속이 가능하므로 빨리 작업을 끝내고 권한을 설정해서 mysql 데몬을 재시작 하는 것이 좋습니다.

```
root@local:/
[root@localhost /]# /usr/local/mysql/bin/mysqld_safe --skip-grant &
[1] 13418
[root@localhost /]# Starting mysqld daemon with databases from /usr/local/mysql/var
```

➢ 권한 설정 제한이 없으니 간단하게 root 로 로그인을 한 후 mysql root 패스워드를 설정 합니다.

```
root@local:/
[root@localhost /]# /usr/local/mysql/bin/mysql -u root mysql
Welcome to the MySQL monitor.  Commands end with ; or ₩g.
Your MySQL connection id is 1 to server version: 4.1.20-log
Type 'help;' or '₩h' for help. Type '₩c' to clear the buffer.
mysql> update user set password = password('new-password') where user = 'root';
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0
mysql> flush privileges;
Query OK, 0 rows affected (0.00 sec)
```

➢ 패스워드 변경을 완료한 후 mysql 데몬을 재 실행 합니다. mysql 재실행 후에 변경한 패스워드로 mysql 접속이 가능 합니다.



tip) mysql 패스워드 분실 했을 때, 변경할 수 잇는 스크립트.

```
#!/bin/sh
echo "mysql 데몬을 중지 합니다"
pkill mysqld
sleep 4
echo "mysql 데몬을 시작 합니다"
/usr/local/mysql/bin/mysqld_safe --skip-grant &
sleep 4
echo " "
echo "mysql 패스워드를 변경합니다."
echo " "
echo "변경할 mysql root 패스워드를 입력하시오"
read newpassword
echo "use mysql;
update user set password=password('$newpassword') where user='root';
flush privileges;" >insert_query
echo " "
/usr/local/mysql/bin/mysql -u root <insert_query
rm -rf insert_query
echo "mysql 데몬을 중지 합니다"
pkill mysqld
sleep 4
echo "mysql 데몬을 시작 합니다"
/usr/local/mysql/bin/mysqld_safe &
sleep 4
echo " "
echo "mysql 패스워드가 $newpassword로 변경되었습니다"
echo " "
```

- mysql 을 통한 웹서비스 중 “Warning...too many connections...” 라는 메시지 출력시
  - 이런 경우는 mysql 의 접속량이 많아서 발생하는 경우입니다. ps ax | grep mysql 로 확인 하면, mysql 데몬의 갯수가 많이 늘어나 홈페이지 속도가 느려지게 됩니다. 최대 실행 가능한 mysql 데몬의 갯수를 늘려서 데몬을 다시 띄워주면 됩니다.

아래는 최대 데몬갯수, 테이블 캐쉬사이즈, timeout 등을 설정하여 데몬을 다시 실행한 것입니다.

```
root@local:/
[root@localhost /]# /usr/local/mysql/bin//mysqld_safe -O max_connections=1000 ₩
-O table_cache=256 -O wait_timeout=300 &
```

4. nfs 장애

- 클라이언트 서버에서 nfs 를 통한 mount 시도시 곧바로 mount 되지 않으며 계속해서 응답이 없으며 클라이언트 서버 /var/log/messages 파일에서 아래와 같은 메시지 확인시

```
root@local:/
[root@localhost /]# vi /var/log/messages
Jun 14 18:09:35 localhost kernel: portmap: server localhost not responding, timed out
Jun 14 18:09:35 localhost kernel: RPC: failed to contact portmap (errno -5).
```

➢ 클라이언트 서버에서 portmap 데몬이 실행 중인지 확인 합니다.
클라이언트 서버에서 portmap 데몬이 정상적으로 실행되지 않은 상태로 mount 시 발생되는 메시지입니다.

- 클라이언트 서버에서 nfs 를 통한 mount 시도시 아래와 같은 메시지 발생시

```
root@local:/
[root@localhost /]# mount -t nfs 10.10.10.10:/backup /home/backup
mount: RPC: Program not registered
```

➢ nfs 서버에서 nfs 데몬이 정상적으로 실행중인지 확인합니다.
nfs 서버측 데몬이 정상적으로 실행되지 않을 때 발생되는 메시지입니다.

- 클라이언트 서버에서 nfs 를 통한 mount 시도시 아래와 같은 메시지 발생시

```
root@local:/
[root@localhost /]# mount -t nfs 10.10.10.10:/backup /home/backup
mount: RPC: Port mapper failure - RPC: Unable to receive
```

➢ nfs 서버에서 portmap 데몬이 정상적으로 실행중인지 확인합니다.
nfs 서버측 데몬이 정상적으로 실행되지 않을 때 발생되는 메시지입니다.

5. samba 장애

- samba 설정시 /etc/samba 디렉토리 아래 samba 패스워드 파일이 없는 경우

```
root@local:/
[root@localhost /]# ls -l /etc/samba
합계 44
-rw-r--r--     1 root      root         11400  6월 15 05:37 smb.conf
-rw-------     1 root      root           521  6월 15 05:46 smbpasswd
-rw-r--r--     1 root      root            38  2월  3 05:46 smbusers
```

- samba 패키지가 모두 설치되지 않은 상태입니다. 패키지가 모두 설치 된 상태에서는 아래와 같이 samba 설정 파일, samba 계정 파일, samba 패스워드 파일이 존재합니다.

- samba 계정 추가시 아래와 같은 메시지가 발생하며 계정이 추가 되지 않는 경우

```
root@local:/
[root@localhost /]# /etc/samba/smbpasswd -a hostway
New SMB password:
Retype new SMB password:
User hostway does not exist in system password file (usually /etc/passwd). Cannot add account
without a valid local system user.
Failed to modify password entry for user hostway
```

- samba 계정 파일은 시스템 계정 파일을 참조 합니다. system 내 계정이 존재하지 않아 발생되는 메시지입니다.

- 공유 정의에서 허용 IP 리스트에 등록하였으나 정상 접속이 되지 않는 경우, iptables 나 방화벽 설정은 이상 없음
  - 허용 IP 리스트의 경우 전체 설정 부분과 공유 정의 부분 모두에서 설정이 가능 합니다. 따라서 공유 정의에서 이상 없이 허용하였어도 전체 설정에서 허용 리스트에 등록되지 않는 경우 허용되지 않습니다.