

제5장 네트워크 서비스

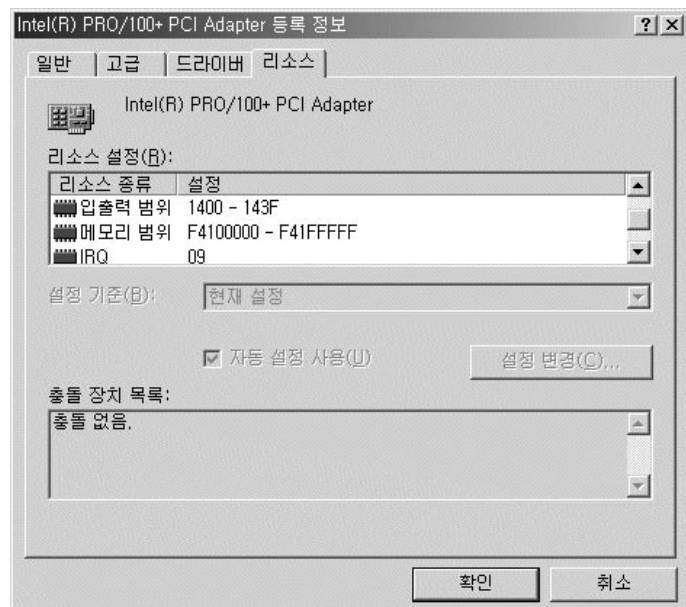
1. 네트워크 개요
2. 네트워크 환경 설정
3. 네트워크 관리도구
4. 네임서비스(DNS)
5. 파일 공유하기(NFS,Samba)
6. 웹 서비스(Apache, SSL,)
7. Mail Service(sendmail)

1. 네트워크 개요

리눅스는 기본적으로 인터넷이나 네트워크로 연결된 환경에서 작동되는 OS 라고 할 수 있습니다. 리눅스 네트워크 구성을 위해서 사용자들은 몇 가지 준비를 할 필요가 있습니다.

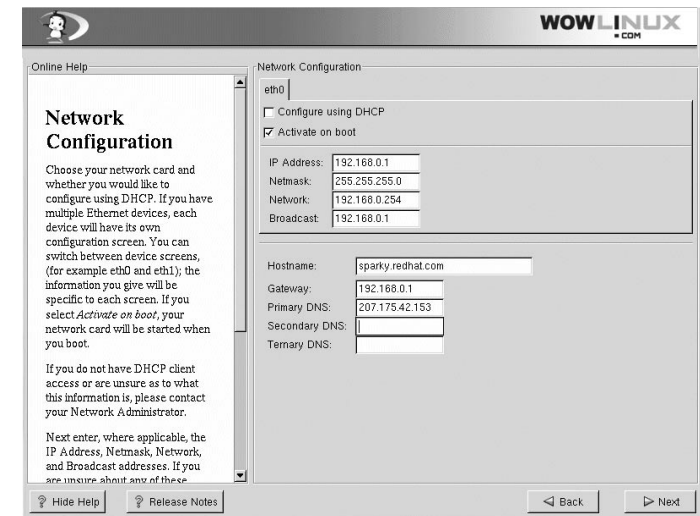
먼저 자신이 네트워크 장비로 활용하려는 장치(Device)의 정보를 알아야 할 것입니다. H/W의 IRQ, IO 포트, DMA 등을 알아두는 것입니다. 최근에 나오는 H/W들은 리눅스에서 대부분 자동으로 설정되지만(Kudzu 라는 툴을 통해서) 일부 장치들은 그렇지 못할 수도 있으므로 기록해 놓는 것입니다. 최근에 PC를 구입했다면 크게 걱정할 일은 아닙니다.

리눅스를 설치하기 전에 Windows 98/2000 등에서 제어판의 장치관리자를 참고 해도 됩니다.



2. 네트워크 환경 설정

랜카드(NIC)가 있다면 와우 리눅스 설치 시에 자동으로 설정화면이 나옵니다. 아래 그림은 설치시 나타나는 IP 어드레스 설정화면입니다. 랜카드가 1장 일 때는 eth0 로 표시되고 2장 이상 일때는 eth1, eth2...등으로 표시됩니다. DHCP는 서버에서 일정 범위의 IP를 가지고 있다가 접속하는 사람들에게 1개씩 자동으로 할당하는 방법입니다. 이것은 공공장소나 불특정 다수가 인터넷을 사용하는 곳에 적당한 방법입니다. Activate on boot 라는 옵션은 부팅 때 마다 랜카드가 작동할 수 있게 하는 옵션입니다. 리눅스에서는 네트워크 설정을 변경해도 리부팅 할 필요가 없습니다.



(1) 기본 설정

이제 부터는 로그인 후에 기본 설정을 변경하는 방법에 관해서 알아보겠습니다. 로그인 후에 설정을 변경하는 방법은 크게 3가지가 방법이 있습니다.

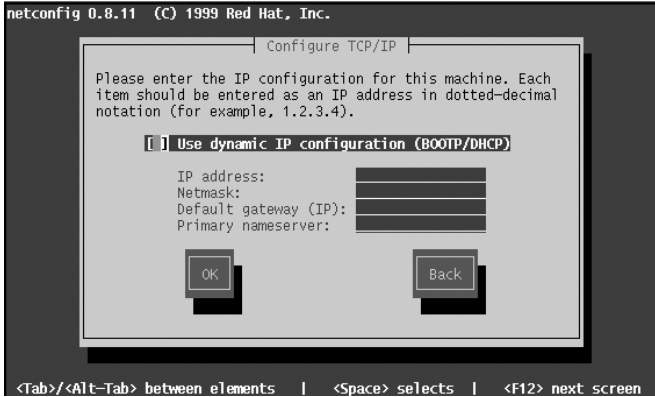
■ 콘솔에서 설정하는 방법-netconfig

이 명령은 간단히 로그인 후에 **netconfig** 라는 명령을 실행하는 것으로 가능합니다. 이때 주의할 점은 반드시 root 사용자의 자격으로 시작해야 한다는 것입니다. 리눅스에서 root 사용자는 모든 실행파일 및 설정 파일들을 조작할 수 있는 권한을 가지고 있습니다.

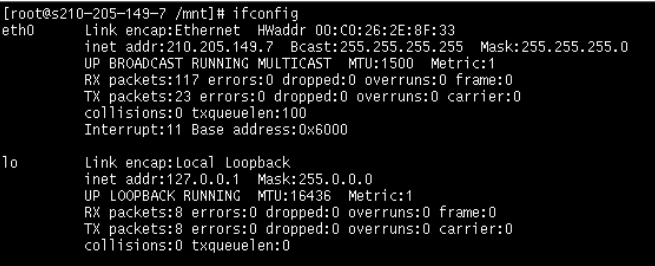
아래 그림은 netconfig를 실행한 후의 화면입니다. 이전의 설정을 무시하고 새로운 설정파일을 생성할 지를 묻는 창입니다. <Yes>를 선택하고 네트워크 설정을 계속합니다.



아래 그림은 새로운 IP 주소와 gateway 정보 등을 입력하는 창입니다. 설치할 때처럼 화면에 지시된 대로 자신이 사용할 PC의 IP 주소 관련 정보를 입력합니다. 만약 DHCP를 사용한다면(두루넷 등의 케이블 모뎀 사용자도 해당) BOOTP/DHCP 항목을 선택합니다. 이 설정이 끝나고 'OK' 버튼을 누르면 자동으로 shell 상태로 빠져 나옵니다.

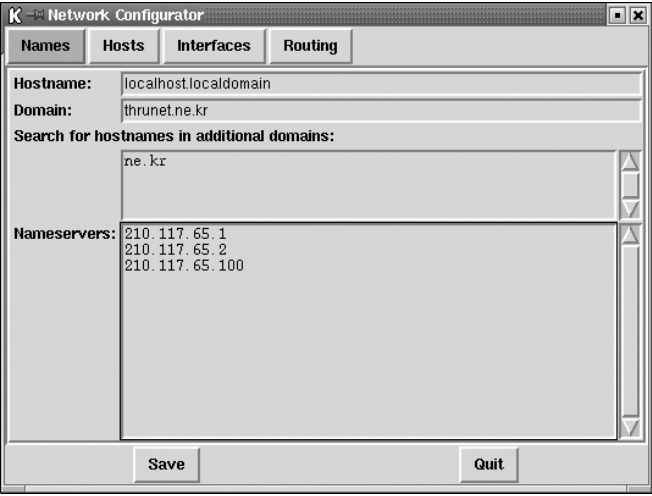


IP 설정이 끝난후에는 ifconfig 라는 명령으로 자신이 설정한 랜카드의 IP 어드레스를 확인 합니다. 아래 그림에서 IP 어드레스가 210.205.149.7로 지정된 것을 알 수 있습니다. 아래에 lo 라는 장치는 loopback 장치를 말하는 것으로 랜카드가 없어도 자신의 리눅스로 접속해서 테스트 할 수 있도록 리눅스에서 자동으로 설정해 두는 것입니다.

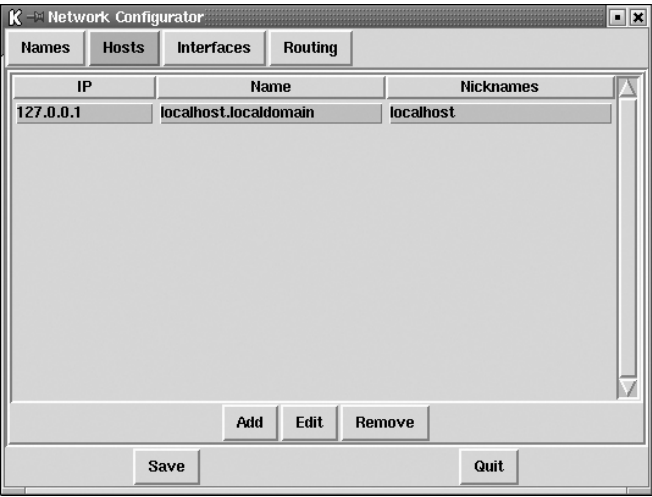


■ X window에서 설정하는 방법-netcfg

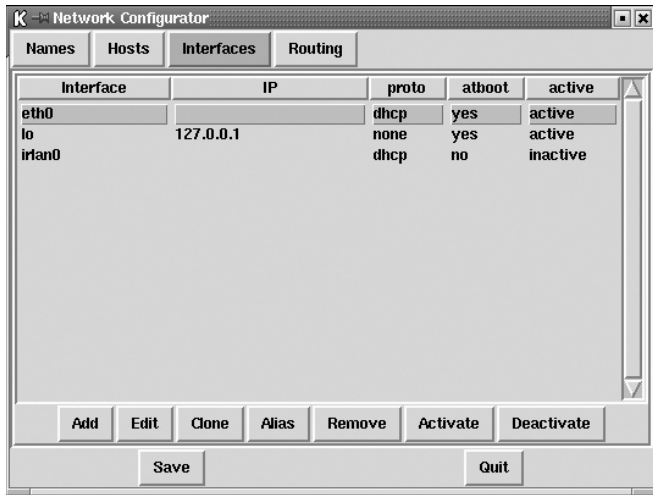
X window에서 네트워크를 설정하는 도구로 **netcfg** 라는 도구가 있습니다. X Window 터미널에서 netcfg 라는 명령을 실행하면 아래 그림과 같은 도구가 나타납니다. 아래의 그림은 두루넷 서비스를 받는 리눅스 PC에서 dhcp를 선택했을 경우 나타나는 화면입니다.



일단 dhcp를 선택하면 hosts 항목은 아무런 항목도 지정되지 않습니다. 아래 그림에서 사용자가 스스로 이름을 붙일수도 있습니다. 즉 다른 컴퓨터의 IP 주소를 자신의 PC에서 별명(Nicknames)을 붙이는 것입니다. 이 방법을 사용하면 다른 서버에 접속할 때 IP 대신 쉬운 이름을 사용할 수 있습니다. PC 통신 서비스들의 이름을 등록해 놓는 것도 괜찮을 것 같습니다.



Interfaces 옵션에서는 물리적으로 랜카드(eth0)의 IP 주소를 설정할 수 있습니다. 이것은 위의 netconfig 항목에서 입력한 IP주소를 입력하는 것과 같은 효과를 나타냅니다. 이 항목이 중요한 항목입니다. 옆에 있는 Routing 버튼은 리눅스를 다른 사용자에게 라우터로 제공할 때 신호(Packet)을 전달할 것인지를 결정하는 옵션입니다.

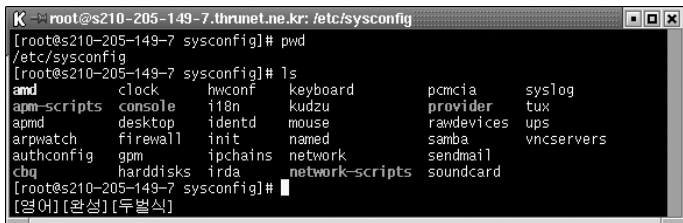


모든 설정이 변경된 후에는 저장(Save) 한 후에 종료(Quit) 하면 됩니다.

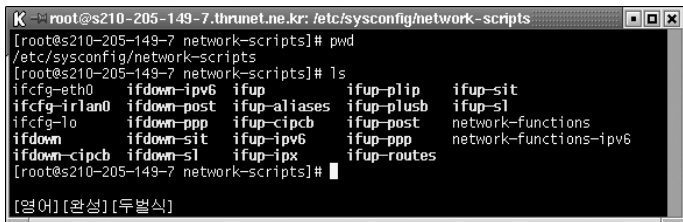
■ 세 번째로는 직접 IP 주소 등을 수정하는 방법

사실 위의 도구를 이용한 방법들도 모두 아래에 설명하는 파일들을 수정하는 결과를 가져옵니다. 리눅스의 장점인 직접적인 접근 방식이라고 할 수 있습니다.

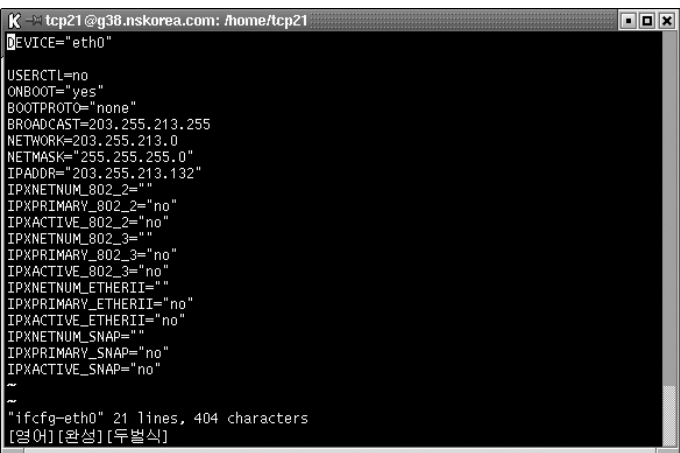
/etc/sysconfig/network-scripts 디렉토리에는 네트워크 설정파일이 들어 있습니다.



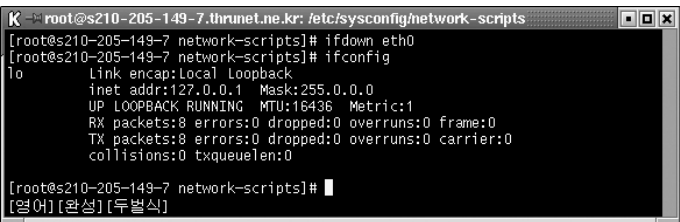
아래 그림에서 우리는 'ifcfg-eth0' 라는 파일을 수정하면 됩니다.



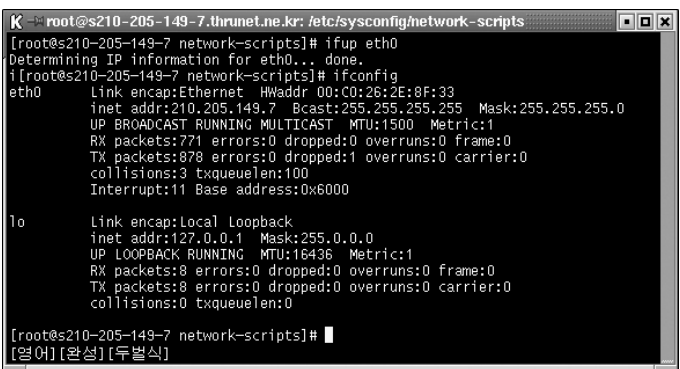
아래 그림은 'eth0' 라는 랜카드의 설정 정보입니다. IPADDR 이후의 정보는 IP-Masquerade를 위해서 설정한 내용들입니다. 신경쓰지 않아도 됩니다. 없어도 상관 없습니다.



ifdown 이라는 명령은 랜카드를 물리적으로 동작을 멈추게 하는 명령입니다. 아래 그림에서처럼 ifdown eth0 라는 명령을 실행하면 eth0 랜카드가 작동이 멈춥니다.



다시 ifup eth0 라는 명령으로 eth0 랜카드를 다시 살렸습니다. 정상적으로 작동하고 있는 것을 ifconfig 명령으로 확인하였습니다.

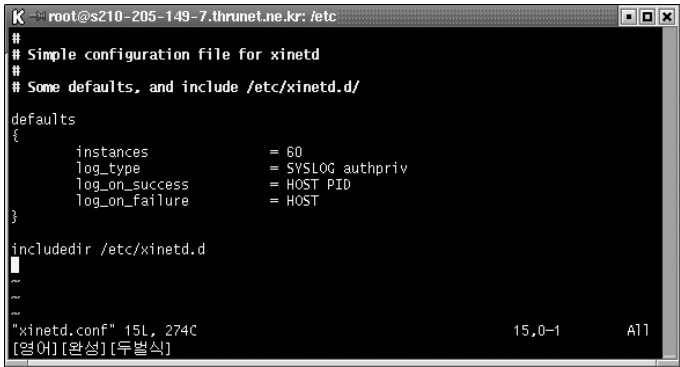


(2) xinetd와 ntsysv(ksysv, tksysv)

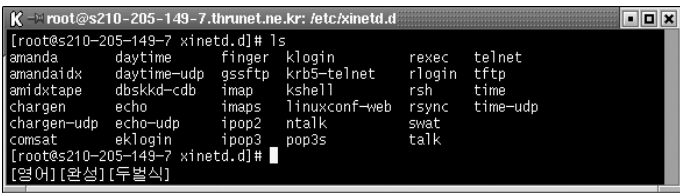
와우 리눅스 7.0 이전 버전에서는 inetd와 tcp-wrapper 등을 조합해서 네트워크 내부에 최소한의 보안 절차를 가졌습니다. 와우 리눅스 7.0 부터는 xinetd 라는 향상된 보안 도구를 사용하고 있습니다. 이번 절에서는 xinetd와 리눅스를 run level 편집기인 ntsysv, ksysv, tksysv 등에 대해서 알아보겠습니다.

① xinetd

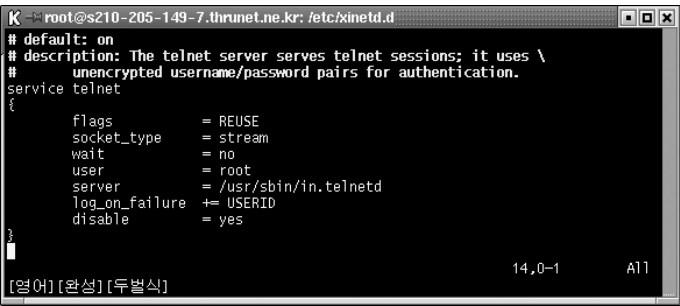
이것은 '/etc/' 디렉토리 아래에 **xinetd.conf** 라는 파일과 **/etc/ xinetd.d** 라는 디렉토리에 관련 파일이 존재 하게 됩니다. 이전 버전의 inetd 와 유사한 구조를 가지게 됩니다.
아래 그림은 xinetd.conf 파일의 내용입니다. xinetd.conf 파일은 /etc/xinetd.d 디렉토리에 들어 있는 여러 가지 설정 파일에 근본적(전역적)으로 영향을 미치게 됩니다.



아래 그림은 xinetd 파일을 모아 놓은 디렉토리 입니다. 아래 그림에서 보는 바와 같이 각각의 설정 파일들은 서비스에 영향을 미치게 됩니다.



telnet 서비스의 예를 들어 보겠습니다. 아래 그림과 같이 되어 있으면 서비스가 제대로 되지 않습니다. 서버를 재시작 할 수 있는 사용자로 root만을 허용했고 disable=yes 로 되어 있습니다. 이것을 disable=no 로 지정합니다. 그리고 나서 **'/etc/rc.d/init.d/xinetd restart'** 라는 명령으로 xinetd를 재시작해주면 telnet 서비스를 시작할 수 있습니다.(xinetd 설정 파일을 수정해 주지 않으면 해당 서비스를 사용하지 못합니다. default가 disable 로 되어 있습니다.)



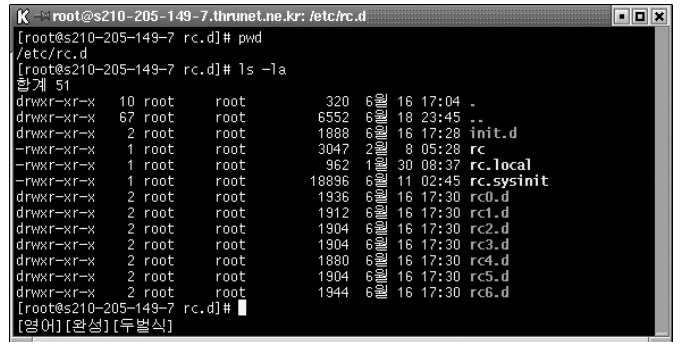
다른 서비스들도 이와 유사한 방법으로 사용이 가능합니다. 주의 해야 할 것은 반드시 xinetd 데몬을 restart 시켜 주어야 한다는 것입니다.

② ntsysv(ksysv,tksysv)

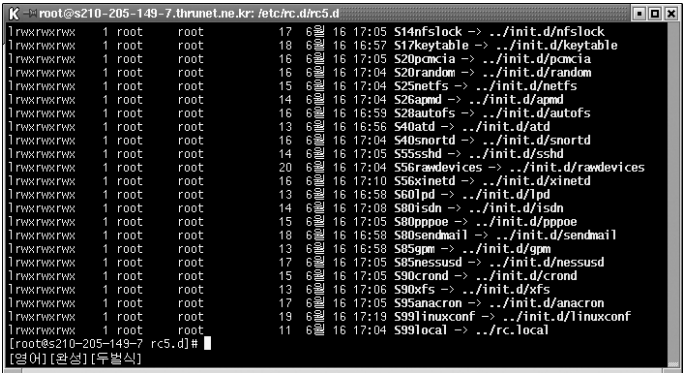
리눅스에는 run level 이라는 것이 있습니다. run level은 Windows 98로 말하자면 부팅모드를 설정해 놓은 것입니다. 리눅스에서는 이렇게 부팅 모드를 디렉토리에 모아 놓고 따로 관리 하고 있습니다. 참고로 부팅 모드는 아래와 같이 지정 되어있습니다. 부팅모드를 결정하는 파일은 /etc/inittab 파일입니다.

- 0 - halt (이것을 기본으로 지정하면 시스템을 켜자마자 전원을 차단해 버립니다.)
- 1 - Single user mode(패스워드를 잃어 버렸을때 슈퍼유저 모드로 부팅합니다.)
- 2 - Multiuser, without NFS (nfs를 지원하지 않는 다중 사용자 모드입니다.)
- 3 - Full multiuser mode(리눅스 네트워크의 모든 기능을 지원하는 모드입니다.)
- 4 - 사용하지 않습니다.
- 5 - X11(X Window로 부팅하는 모드입니다. 와우리눅스의 기본 설정모드입니다.)
- 6 - reboot (이것을 기본으로 지정하면 시스템을 켜자마자 리부팅 됩니다.)

와우 리눅스의 서버 서비스를 시작하게 하는 파일들은 **/etc/rc.d/** 디렉토리 아래에 있습니다. 아래그림에서 보는 것과 같은 구조로 되어 있습니다. init.d 라는 디렉토리에는 현재 설정된 run level의 시작 스크립트들이 들어 있습니다. rc, rc.local, rc.sysinit 파일에는 리눅스가 시작될 때 사용하는 batch 파일들이 들어 있습니다. 그 아래에 rc*.d 디렉토리들은 각각의 run level 에 해당되는 서비스들의 파일들을 포함하고 있습니다.



아래 그림은 rc5.d(run level 5) 디렉토리의 내용입니다. 여기서 보는 바와 같이 실제 실행 서비스 데몬 파일들이 심볼릭 링크되어 있습니다. S로 시작한다는 것에 주목하십시오.



이제 이러한 서비스들의 run level을 수동으로 설정하는 방법에 대해서 설명합니다. 크게 3가지 도구가 있습니다. 하는 역할들은 모두 같고, run level 실행 디렉토리들을 변경합니다.

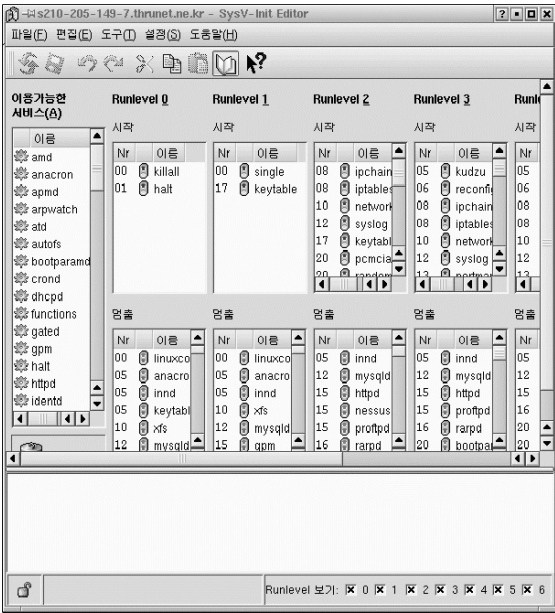
■ ntsysv

아래 그림에서 보는 바와 같이 ntsysv 는 콘솔이나 터미널용 프로그램입니다. 해당되는 서비스를 선택/해제 하려면 <Space Bar> 를 누르면 됩니다. <F1>을 누르면 해당되는 서비스의 간략한 설명을 볼 수도 있습니다. 가장 간편하게 서비스들을 실행/해제 할 수 있는 도구입니다.



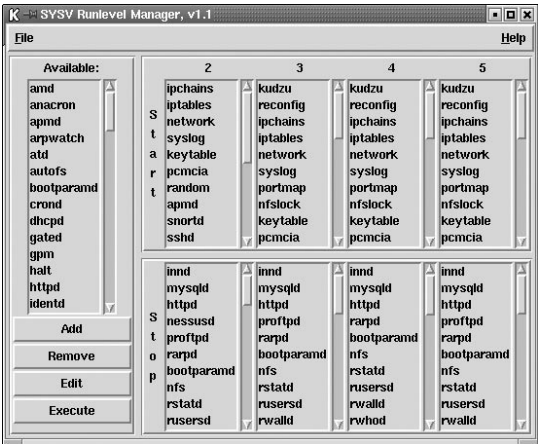
■ ksysv

이 프로그램은 X Window에서 사용할 수 있는 KDE용 run level 편집기입니다. 아래 그림에서 보는 바와 현재 이용가능한 서비스와 각 run level 에 해당되는 서비스들을 보여 줍니다. 시작과 정지된 서비스들을 한눈에 보여 줍니다.



■ tksysv

이 프로그램은 가장 오래된 X Window용 run level 편집기입니다. 가장 기본적인 형태를 보여줍니다. Tcl/tk 툴킷을 사용해서 제작된 도구입니다. 현재도 많은 사람들이 사용하고 있습니다.



(3) rp-pppoe를 이용한 ADSL 접속

한국통신 ADSL접속에 대한 자세한 사항은 별책으로 만들어진 Install Guide를 참고 하시기 바랍니다.

3. 네트워크 관리도구

(1) ifconfig

이 명령은 네트워크 카드에 부여된 주소를 확인하고 새롭게 IP 주소를 부여하거나 이미 할당된 IP 주소를 수정 제거 할 수 있습니다.

- ifconfig 명령을 옵션없이 수행하면 현재 설정되어 있는 랜카드의 설정을 보여 주게 됩니다. ifconfig 명령은 root 사용자만이 실행 할 수 있습니다. 아래 그림에서 eth0 랜카드의 설정을 보여주고 있습니다.

```
[root@210-205-149-7 /mnt]# ifconfig
eth0      Link encap:Ethernet  HWaddr 00:C0:26:2E:8F:33
          inet addr:210.205.149.7  Bcast:255.255.255.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:117 errors:0 dropped:0 overruns:0 frame:0
          TX packets:23 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:100
          Interrupt:11 Base address:0x6000

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:8 errors:0 dropped:0 overruns:0 frame:0
          TX packets:8 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
```

- 랜 카드에 IP 주소 부여하기

새로운 랜카드에 IP 주소를 부여하려면 아래와 같이 사용하면 됩니다.

```
ifconfig eth0 192.168.0.2 netmask 255.255.255.0 up
```

위의 명령으로 eth0 랜카드에 수동으로 IP 주소를 지정해 줄 수 있습니다. 만약 하나의 랜카드에 여러개의 IP 주소를 지정해야 한다면 아래와 같이 하면 됩니다.

```
ifconfig eth0:0 192.168.0.3 netmask 255.255.255.0 up
```

이렇게 하면 eth0 라는 물리적인 랜카드에 192.168.0.2 번 주소와 192.168.0.3 번 주소가 할당되게 됩니다.

- 랜카드의 IP 주소 정지하기

랜카드의 할당된 IP 주소를 정지하기 위해서는 아래와 같은 명령을 입력하면 됩니다.

```
ifconfig eth0 down
```

지금까지 살펴본 ifconfig 명령과 유사하게 아래와 같이 랜카드를 사용할 수도 있습니다.

```
ifup eth0-랜카드를 새로 시작할 경우
```

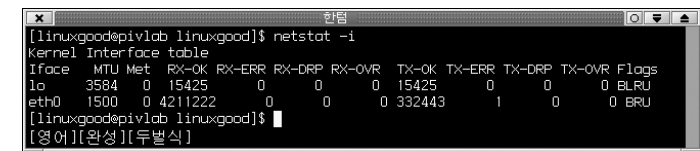
```
ifdown eth0-랜카드를 중지 시킬 경우
```

(2) netstat

이 명령은 자신의 리눅스 PC의 네트워크 상태를 알 수 있게 합니다. netstat 명령의 여러 가지 사용방법을 알아보도록 하겠습니다. netstat 명령은 network statics 라는 단어를 합성한 단어입니다. 이 유틸리티를 통해서 자신의 리눅스 서버를 좀더 세밀히 관찰 할 수 있습니다.

- 자신의 랜카드의 현재 작동 상황을 살펴보려면 아래그림과 같이 하면 됩니다. MTU는 한 패킷의 최대 전송 바이트수를 나타내고 Rx-OK는 에러없이 서로 전송받은 패킷을 나타냅니다. Rx-ERR은 에러가 생긴 패킷 신호의 개수를 나타냅니다. Rx-DRP 는 오다가 버려진 패킷신호를 나타냅니다. Rx-OVR은 패킷을 전송하는 도중에 Overrun 에러가 난 것을 뜻합니다. Tx 신호는 보내는 패킷에 대한 내용을 담고 있습니다.

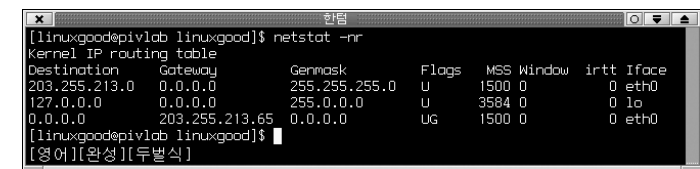
netstat -i



```
[linuxgoodepivlab linuxgood]$ netstat -i
Kernel Interface table
Iface  MTU Met  RX-OK RX-ERR RX-DRP RX-OVR  TX-OK TX-ERR TX-DRP TX-OVR  Flags
lo      3584  0   15425    0    0    0   15425    0    0    0  BLRU
eth0    1500  0  4211222    0    0    0   332443    1    0    0  BRU
```

- 현재 설정되어 있는 라우팅 테이블의 이름을 보려면 아래와 같이 합니다. 이 명령은 route 명령과 유사합니다. 현재 구성되어 있는 라우팅 테이블의 자세한 내용을 보여줍니다.

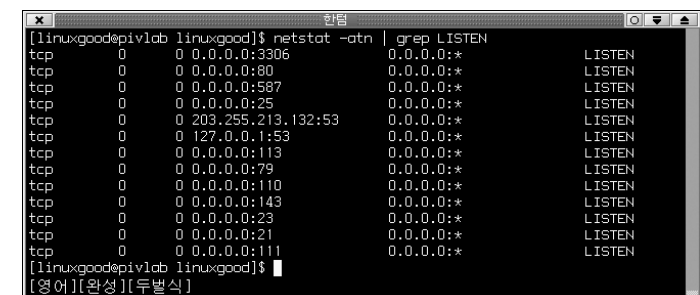
netstat -nr



```
[linuxgoodepivlab linuxgood]$ netstat -nr
Kernel IP routing table
Destination Gateway         Genmask         Flags MSS Window  irtt Iface
203.255.213.0  0.0.0.0         255.255.255.0   U      1500  0          0 eth0
127.0.0.0      0.0.0.0         255.0.0.0       U      3584  0          0 lo
0.0.0.0        203.255.213.65  0.0.0.0         UG     1500  0          0 eth0
```

- netstat 명령을 grep 이라는 유틸리티와 함께 사용해서 현재 서비스를 위해서 열려 있는 포트의 정보를 출력하게 합니다.

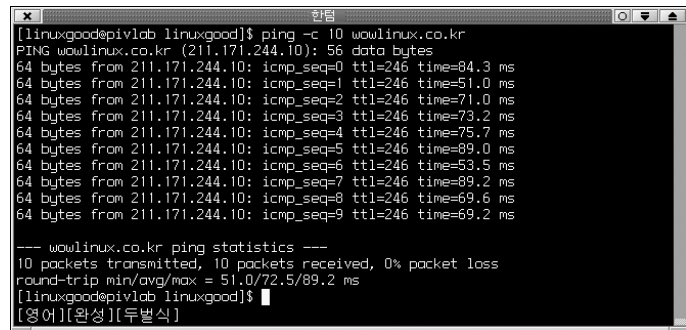
netstat -atn | grep LISTEN



```
[linuxgoodepivlab linuxgood]$ netstat -atn | grep LISTEN
tcp      0      0 0.0.0.0:3306      0.0.0.0:*        LISTEN
tcp      0      0 0.0.0.0:80        0.0.0.0:*        LISTEN
tcp      0      0 0.0.0.0:587       0.0.0.0:*        LISTEN
tcp      0      0 0.0.0.0:25        0.0.0.0:*        LISTEN
tcp      0      0 203.255.213.132:53 0.0.0.0:*        LISTEN
tcp      0      0 127.0.0.1:53      0.0.0.0:*        LISTEN
tcp      0      0 0.0.0.0:113      0.0.0.0:*        LISTEN
tcp      0      0 0.0.0.0:79        0.0.0.0:*        LISTEN
tcp      0      0 0.0.0.0:110      0.0.0.0:*        LISTEN
tcp      0      0 0.0.0.0:143      0.0.0.0:*        LISTEN
tcp      0      0 0.0.0.0:23        0.0.0.0:*        LISTEN
tcp      0      0 0.0.0.0:21        0.0.0.0:*        LISTEN
tcp      0      0 0.0.0.0:111      0.0.0.0:*        LISTEN
```

(3) ping

ping 은 packet internet gopher의 합성어입니다. ping 으로 우리는 원격지에 있는 서버의 동작상태를 점검 할 수 있습니다. 아래 그림은 ping을 사용해서 원격지에 있는 서버를 점검한 그림입니다. -c 10 옵션은 10번 ping 신호를 보내서 돌아오는 것을 보게 한다는 의미입니다. 만약 무제한으로 ping test를 하고자 한다면 'ping IP어드레스[도메인 네임]'을 실행 하면 됩니다. 빠져 나가려면 <Ctrl+C>를 누르면 됩니다.

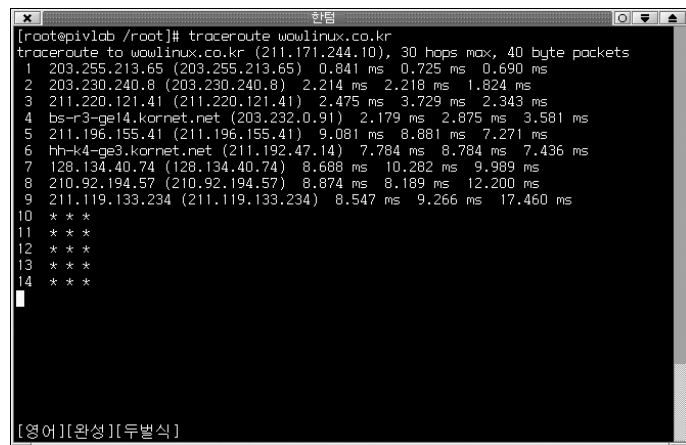


```
[linuxgood@pivlab linuxgood]$ ping -c 10 wowlinux.co.kr
PING wowlinux.co.kr (211.171.244.10): 56 data bytes
64 bytes from 211.171.244.10: icmp_seq=0 ttl=246 time=84.3 ms
64 bytes from 211.171.244.10: icmp_seq=1 ttl=246 time=51.0 ms
64 bytes from 211.171.244.10: icmp_seq=2 ttl=246 time=71.0 ms
64 bytes from 211.171.244.10: icmp_seq=3 ttl=246 time=73.2 ms
64 bytes from 211.171.244.10: icmp_seq=4 ttl=246 time=75.7 ms
64 bytes from 211.171.244.10: icmp_seq=5 ttl=246 time=89.0 ms
64 bytes from 211.171.244.10: icmp_seq=6 ttl=246 time=53.5 ms
64 bytes from 211.171.244.10: icmp_seq=7 ttl=246 time=89.2 ms
64 bytes from 211.171.244.10: icmp_seq=8 ttl=246 time=69.6 ms
64 bytes from 211.171.244.10: icmp_seq=9 ttl=246 time=69.2 ms

--- wowlinux.co.kr ping statistics ---
10 packets transmitted, 10 packets received, 0% packet loss
round-trip min/avg/max = 51.0/72.5/89.2 ms
[linuxgood@pivlab linuxgood]$
```

(4) traceroute

이 명령은 사용자가 어떤 경로를 통해서 호스트에 접속되는지를 보여주는 명령입니다. 이 명령도 역시 root 사용자만이 사용할 수 있는 명령입니다. 아래 그림은 현재 속해 있는 네트워크 에서 wowlinux.com 로 어떤 경로를 통해서 신호가 전달되는지를 보여주는 그림입니다. 이 도구를 사용해서 사용자는 네트워크의 어느 지점에서 정체현상(?)이 있는지를 알 수 있습니다.



```
[root@pivlab /root]# traceroute wowlinux.co.kr
traceroute to wowlinux.co.kr (211.171.244.10), 30 hops max, 40 byte packets
 1  203.255.213.65 (203.255.213.65)  0.841 ms  0.725 ms  0.690 ms
 2  203.230.240.8 (203.230.240.8)  2.214 ms  2.218 ms  1.824 ms
 3  211.220.121.41 (211.220.121.41)  2.475 ms  3.729 ms  2.343 ms
 4  bs-r3-ge14.kornet.net (203.232.0.91)  2.179 ms  2.875 ms  3.581 ms
 5  211.196.155.41 (211.196.155.41)  9.081 ms  8.881 ms  7.271 ms
 6  hh-k4-ge3.kornet.net (211.192.47.14)  7.784 ms  8.784 ms  7.436 ms
 7  128.134.40.74 (128.134.40.74)  8.688 ms  10.282 ms  9.989 ms
 8  210.92.194.57 (210.92.194.57)  8.874 ms  8.189 ms  12.200 ms
 9  211.119.133.234 (211.119.133.234)  8.547 ms  9.266 ms  17.460 ms
10  * * *
11  * * *
12  * * *
13  * * *
14  * * *
```

4. 네임서비스(DNS)

네임서비스 라는 것은 IP 주소로 된 서버의 주소를 일반적인 영문 이름으로 변경해 주는 서비스를 말합니다. 이 서비스에서 사용되는 유틸리티로 bind 라는 유틸리티와 named-cache 서버가 있습니다. bind 는 항상 최신 버전의 패치가 필요합니다. 현재 와우리눅스 7.1 에 포함된 버전은 bind-9.1.0-10 버전입니다. 최근에 DNS 서비스를 제공하는 bind 유틸리티의 보안 버그로 인해 해킹사건이 발생한 적이 있었습니다.

네임 서비스는 bind 유틸리티가 설치된 상황에서 named 라는 서비스를 시작하면 됩니다. 설정파일을 몇 가지 수정해 주어야 합니다.

(1) DNS 서버의 구축

만약 서버로 웹호스팅 서비스를 하고자 한다면 DNS(Domain Name Services : 도메인네임서비스) 서버를 잘 구축해야 합니다. 항상 출발을 시켜주는 곳은 DNS서버입니다. 이것은 1개의 IP어드레스에 여러 개의 도메인을 사용할 때도 중요하게 작용하는 설정에 해당됩니다.

DNS 서버 구축에 관한 글은 인터넷을 뒤져보면 매우 많이 나와 있습니다. 특히 <http://kldp.org>에서 관련 문서들을 찾아보고 한번 읽어 보시는 것도 괜찮을 것이라는 생각입니다.

리눅스 설치중에 네트워크 설정을 할 때 네임서버의 이름을 물어 보았을 것입니다. 여기서 Primary 와 Secondary 서버를 지정하였을 것입니다. Secondary를 두는 것은 첫 번째 네임서버가 잘못되었을 경우에도 계속서비스 할 수 있도록 해 주기 위해서입니다. 도메인 신청기관에 신청을 할 때 써 주어야 합니다.

보통은 네임서버의 이름으로 ns(primary), ns2(secondary)를 host name 으로 사용하고 1개의 IP를 라우터나 스위치 등의 장치에 사용하고, 나머지를 ns,ns3등의 이름에 할당하게 됩니다. 즉, ns.wowlinux.com / ns2.wowlinux.com 이라는 이름으로 네임서버를 구축하였습니다.

wowlinux.com 이라는 회사에서 설정한 DNS서버의 설정을 예로 설명하도록 하겠습니다.

(2) Bind(Berkeley Internet Name Daemon) 설치하기

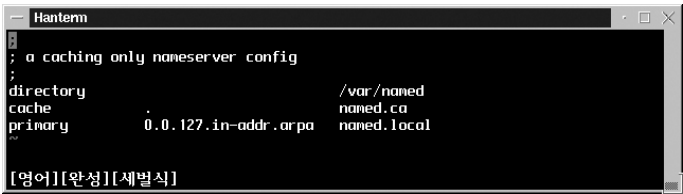
Bind 라는 것은 네임서버가 제대로 작동하기 위해서 필수적인 데몬(유틸리티)입니다. 아마도 리눅스가 설치될 때 자동으로 설치되었을 수도 있습니다. 일반적으로 배포되는 버전은 BIND Version 8.2.2 patchlevel 5 (Released November 12th, 1999)입니다.



- ① bind의 설치와 버전을 확인해 봅니다
- #rpm -qa | grep bind
- ② nameserver이름을 찾는데 시간을 절약해주는 caching-nameserver 라는 꾸러미의 설치 여부도 확인합니다.
- #rpm -qa | grep nameserver

(3) /etc/named.boot 파일 (BIND-4.X 대의 설정파일)

bind 와 caching-nameserver 의 설치가 완료되었다면 이제는 설정파일들을 수정합니다. /etc/named.boot 파일은 bind 가 시작될 때 참조되는 파일이며 환경정보와 서버로 운영할 도메인의 설정사항들이 기술됩니다. 내용중에서 ‘:’ 는 주석을 의미합니다. 설정을 변경해줄 필요는 없습니다.



(4) /etc/named.conf 파일 (BIND 8.X 이상에서의 설정파일)

이 파일을 수정하여서 네임서버의 설정을 합니다. C언어의 문법과 유사합니다. 아래의 내용은 ns.wowlinux.com(Primary 네임서버)을 위한 /etc/named.conf 파일입니다. Bind-8 에서는 //를 주석으로 사용합니다. 또, /var/named 디렉토리 아래에 Zone 파일을 첨가하고 수정해서 웹호스팅을 위한 네임 서버를 구성할 수 있습니다.

① 첫 번째 네임 서버를 위한 설정(Primary name server)

```
options {
    directory "/var/named";           // 웹호스팅을 위한 베이스 디렉토리
    dump-file "/var/tmp/named_dump.db"; // Dump 파일 디렉토리
    statistics-file "/var/tmp/named.stats"; // 접속통계를 담아두는 파일
    pid-file "/var/run/named.pid";    // 프로세서 ID 정보 기록경로
};
```

```
//필요하지 않은 로그파일의 삭제를 위한 설정
logging {
    category lame-servers { null; };
    category cname { null; };
    category response-checks { null; };
    category notify { null; };
};
```

// 캐쉬 파일을 위한 설정

```
zone "." IN {
    type hint;
    file "named.root";
};
```

// localhost(내부 접속용)를 위한 설정

```
zone "0.0.127.in-addr.arpa" IN {
    type master;
    file "zone-0.0.127.in-addr.arpa";
};
```

//도메인네임에 대한 설정

```
zone "wowlinux.com" IN {
    type master;
    file "zone-wowlinux.com";
};
```

② 두 번째 네임 서버를 위한 설정(Secondary name server)

```
options {
    directory "/var/named";
};
logging {
    category lame-servers { null; };
    category cname { null; };
};
```



```
zone "." IN {
    type hint;
    file "named.root";
};

zone "0.0.127.in-addr.arpa" IN {
    type master;
    file "zone-0.0.127.in-addr.arpa";
};

zone "213.255.203.ns2" IN {
    type slave;
    file "ns2";
    masters { 203.255.213.132; };
};

//ns2.wowlinux.com 에 대한 설정
zone "wowlinux.com" IN {
    type slave;
    file "ns2-wowlinux.com";
    masters { 203.255.213.134; };
};
```

이상의 예에서 보았듯이 //ns2.wowlinux.com을 설정하는 부분에서 추가로 다른 사람의 네임서버를 구성해 줄 수 있습니다. 이러한 형식으로 다른 사람이 원하는 서버를 구성해 주면 됩니다. 단, IP주소는 모두 같다는 것을 알 수 있습니다. 이제부터는 웹호스팅을 이용하는 사용자의 설정파일을 수정해 주는 부분이 남았습니다.

(5) /var/named/zone-wowlinux.com

이 파일은 네임서버의 전반적인 설정이 담겨 있는 파일입니다.

(6) SOA 레코드 (Start Of Authority)

Zone 파일은 항상 SOA 레코드로 시작하게 됩니다. SOA 레코드는 네임서버가 최적의 상태로 관리되도록 해줍니다.

```
wowlinux.com. IN SOA ns.wowlinux.com. hostmaster.wowlinux.com. (
                                2001060101 :Serial
                                21600      :Refresh ( 6 hours)
                                1800       :Retry  (30 minutes)
                                1209600    :Expire  (14 days)
                                86400      :Minimum ( 1 day)
IN      NS      ns.wowlinux.com.
IN      A      203.255.213.133
```

1 줄에는 도메인 명이 위의 형식과 같이 들어 가게 됩니다. 도메인 이름(wowlinux.com 대신에 @ 기호를 사용해도 됩니다.) 여기서 IN(Internet)은 클래스 이름입니다. 클래스는 여러 가지 이름들이 사용될 수 있습니다. SOA 에 관한 사항을 모두 기입했습니다.면 이번에는 Primary 네임서버와 관리자의 Email 주소가 들어가게 됩니다. 다음에는 Serial, Refresh, Retry, Expire, Minimum 라는 필드가 있습니다. 여기서 Minimum 만이 Primary 에 관련된 설정사항이고, 나머지는 Secondary 서버를 위한 설정입니다. 단 위는 '초'이며 M(Minute), H(Hour), D(Day), W(Week)를 붙여 날짜와 시간을 표시합니다.

(7) NS(Name Server) 레코드

위의 예에서 wowlinux.com. IN NS ns.wowlinux.com. 에 해당되는 부분입니다.

(8) A(Address) & CNAME(Canonical Name) 레코드

A 레코드는 도메인에 IP를 부여하는 역할을 합니다.

```
mail.wowlinux.com.      IN A      203.255.213.133
webservice              IN A      203.255.213.133
```

//www를 webservice.wowlinux.com 이라고 가정합니다.

www IN CNAME webservice.wowlinux.com.

CNAME 레코드는 같은 IP어드레스에 다른 이름을 부여해서 인식하도록 하는 것입니다. 위에서 www.wowlinux.com 이라고 쳐도 203.255.213.133 번에 해당되는 정보를 보여주고 webservice.wowlinux.com 이라고 쳐도 같은 역할을 하게됩니다.

(9) 웹호스팅 서비스를 받는 업체를 위한 Zone 데이터 베이스

아래의 예는 /var/named/zone-cactus2000.com 이라는 파일로 다른 업체의 서버설정을 해 준 것입니다. 아래와 같이 만들어 놓으면 www.cactus2000.com 이라는 이름과 ftp.cactus2000.com 이라는 서버이름의 설정이 완료된 것입니다.(서버가 설정된 것은 아닙니다.)

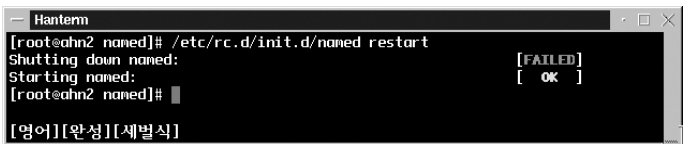
```
@      IN      SOA      ns.cactus2000.com. admin.cactus2000.com. (
                                2001000101 ; Serial
                                28800      ; Refresh
                                14400      ; Retry
                                3600000    ; Expire
                                86400 )    ; Minimum
IN      NS      ns.cactus2000.com.
IN      A       203.255.213.134
IN      HINFO   "x86 Pentium" "WOWLinux 7.1"
```

; IP에 해당되는 별명(alias)설정부분

```
ns      IN      CNAME @
www     IN      CNAME @
ftp     IN      CNAME @
```

Name Server 재시작

모든 설정이 끝났다면 이제는 아래 그림처럼 네임서버를 재시작 하면 됩니다.



네임 서버의 정보가 전 세계적으로 갱신 되려면 보통 3 ~ 7일정도의 시간이 소용됩니다. 그것은 전 세계적으로 운영되고 있는 name 서버의 DNS 정보를 갱신해야 하기 때문입니다.

5. 파일 공유하기

다른 시스템에 있는 파일을 공유하는 방법에는 두가지 방법이 있습니다. NFS 와 samba 가 그것입니다. NFS는 같은 리눅스 등의 유닉스 형태의 시스템 사이에서 파일을 공유하는 방법이고 samba는 Windows 운영체제등에서 파일을 공유하기 위한 방법입니다.

(1) NFS(Network File System) 서버 설치 및 환경설정

NFS는 파일 시스템을 제공해주는 서버와 서버의 파일 시스템을 이용하는 클라이언트로 이루어집니다. 여기서는 먼저 NFS 서버의 설치 및 환경설정에 대해 알아보겠습니다.

NFS 서버 설치하기

거의 모든 유닉스 시스템은 처음 설치할 때 NFS에 필요한 파일을 설치합니다. 따라서 별도의 NFS 설치 과정은 필요 없습니다. 리눅스도 초기 설치 과정에서 NFS가 기본적으로 선택되어 있으므로 대부분의 시스템에 설치되어있을 것입니다.

설치 여부를 확인하기 위해 아래와 같이 실행해 보면 됩니다.

rpm -qa | grep nfs

만약 설치되어 있지 않다면 관련 RPM파일을 설치하면 됩니다.

rpm -Uvh nfs-utils-0.3.1-5.rpm

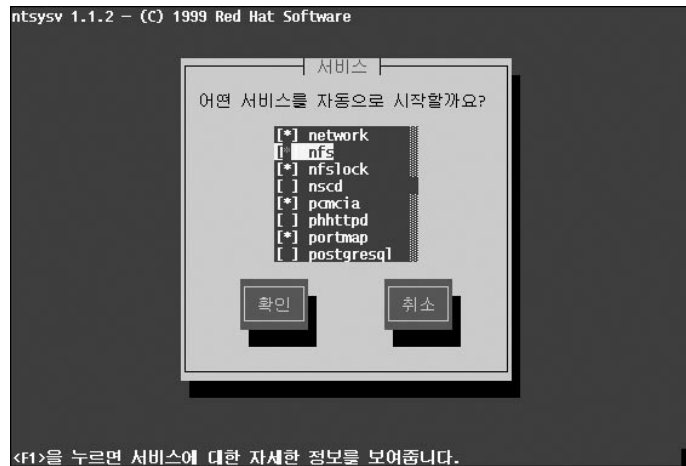
NFS 서버의 실행

아래 그림에서 보는 바와 같이 시작 스크립트를 수행하면 관련 daemon 들이 시작이 되면서 다른 클라이언트에게 서비스를 할 수 있도록 시작이 됩니다.

/etc/rc.d/init.d/nfs start

만약 NFS 서비스가 시스템이 시작될 때 자동으로 시작되게 하려면 Run Level에 추가해주어야 합니다. NFS관련 서비스를 시작해 주면 됩니다.





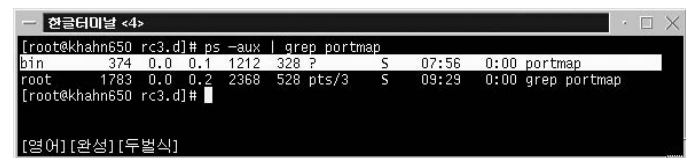
NFS 서버 데몬

NFS 서비스에 필요한 모든 설치를 마치고, 서비스가 시작된다는 것을 확인했다면, 이제는 서비스에 필요한 설정을 해야 합니다. NFS 서비스를 정상적으로 수행하기 위한 스크립트는 `/etc/rc.d/init.d/nfs` 입니다. (nfs 스크립트의 내용을 참조)

■ portmap : 이 서비스는 RPC 서비스를 TCP/UDP 포트에 연결하는 역할을 합니다.

NFS는 쉘의 XDR(아키텍처와 무관하게 데이터를 표현하는 방법)과 RPC(Remote Procedure Call) 인터페이스를 기반으로 설계되었습니다. RPC를 이용하는 프로그램이 시작되면 그 프로그램은 자신이 제공하는 서비스와 자신이 사용하는 포트를 portmap에 등록하게 됩니다. 그리고, 클라이언트는 portmap에 문의해 원하는 서버에 접근할 수 있는 방법을 알아내게 됩니다. 유닉스 시스템에 따라서는 portmap 대신에 rpc.portmap 또는 rpcbind라고도 부릅니다.

리눅스에서 돌아가고 있는 portmapper를 확인하려면 아래와 같이 하면 됩니다.



■ rpc.mountd : 이것은 외부에서 마운트 요청이 오면 응답해 주는 역할을 합니다.

NFS 클라이언트가 서버의 파일 시스템을 이용하기 위해서는 먼저 서버가 자신의 파일 시스템을 NFS를 이용해 다른 호스트가 공유할 수 있도록 설정해주어야 합니다. 리눅스에서는 `/etc/exports`라는 파일을 이용합니다. NFS 클라이언트가 공유된 NFS 서버의 파일 시스템을 이용하기 위해서는 반드시 서버의 파일 시스템을 마운트 해야 합니다. NFS클라이언트가 마운트를 요청해오면, rpc.mountd 데몬이

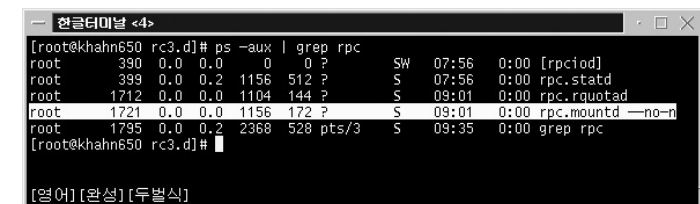
`/etc/exports` 파일의 설정에 따라 마운트 요청을 처리합니다.

클라이언트가 서버 파일 시스템을 마운트할 수 있는 권한을 획득한 이후에도 서버와 클라이언트간에 접속이 계속 유지되는 것은 아닙니다. 즉 http처럼 요청이 들어오는 경우에만 접속이 이루어졌다가 더 이상 필요가 없으면 접속이 해제되는 stateless 방식을 취하게 됩니다.

유닉스 시스템에 따라서는 rpc.mountd 대신에 mountd라는 용어를 쓰기도 합니다.

■ rpc.nfsd : 서버파일을 클라이언트에 제공하는 역할을 합니다.

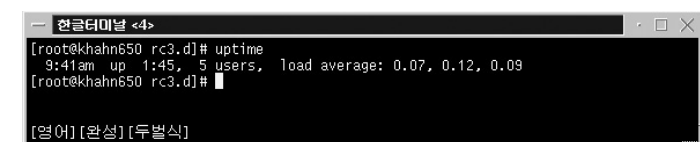
rpc.mountd가 클라이언트의 마운트 요청을 받아들이면, 클라이언트는 마운트된 파일 시스템에 대해 다양한 작업을 할 수 있습니다. 클라이언트가 작업을 수행하면서 서버 쪽 파일 시스템에 무언가를 요구하게 되면 rpc.nfsd 데몬이 이를 맡아서 처리하게 됩니다. rpc.mountd와 마찬가지로 자기 자신이 NFS 서버로서의 역할도 동시에 수행하고 있지 않다면 NFS 클라이언트가 rpc.nfsd를 실행시킬 필요는 없습니다. 아래 그림과 같이 하면 리눅스 시스템에서 돌아가는 rpc daemon 들을 확인할 수 있습니다.



rpc.nfsd는 포크(fork)를 통해 만들어내는 자기 복사본의 수를 몇개로 할 것인가를 지정하는 하나의 매개 변수만 받아들이게 됩니다. 리눅스를 제외한 대부분의 시스템에서 적절한 수의 rpc.nfsd를 설정하는 것은 매우 중요합니다. 하지만, 어떤 수가 적절한 숫자인지 알 수 있는 명확한 척도는 없고, 대부분의 경우 시행착오를 통해서 알아낼 수 밖에 없습니다. 이 숫자가 너무 많거나 적으면 NFS의 성능은 크게 영향을 받게 됩니다.

적어도 4개의 rpc.nfsd를 실행시키는게 일반적입니다. NFS를 자주 이용하지 않는다면 4개 정도면 충분합니다. 이론적으로는 수백, 수천개의 rpc.nfsd를 실행하는 것도 가능하지만, 너무 많은 rpc.nfsd를 실행하는 것도 각각의 프로세스가 CPU를 두고 서로 경쟁하기 때문에 시스템 성능을 저하시키게 됩니다.

서버의 load average(uptime 명령으로 알 수 있습니다.)가 급격히 많아지는 시점까지 rpc.nfsd의 숫자를 늘려보면, 시스템이 감당할 수 있는 rpc.nfsd의 최대 값을 알 수 있습니다. 이 최대값에서 몇 개를 줄이면 안전한 값이 되지만, 이 값이 최선의 값이라는 보장은 없습니다. 단지 최대 값을 기준으로 한 것이기 때문입니다.



또 하나의 방법은 UDP 오버플로우의 개수로 파악하는 방법입니다. 부하가 많이 걸리는 NFS 서버의 경우, 모든 rpc.nfsd가 사용중일 때 또 다른 요청이 들어오면 UDP 소켓이 오버플로우를 일으킬 수 있습니다. 오버플로우의 개수는 netstat -s 명령으로 알아 볼 수 있는데, 이를 통해 UDP 소켓 오버플로우가 0으로 떨어지려면 대략 얼마나 많은 rpc.nfsd가 있어야 하는지 알아내는 것입니다. 이렇게 알아낸 rpc.nfsd의 최적값을 시스템의 rc 스크립트에 적용하면 됩니다.

```
한글터미널 <4>
[root@khahn650 rc3.d]# netstat -s
Ip:
 14628 total packets received
 0 forwarded
 0 incoming packets discarded
 14122 incoming packets delivered
 9262 requests sent out
 201 reassemblies required
 68 packets reassembled ok
Icmp:
 1 ICMP messages received
 0 input ICMP message failed.
 ICMP input histogram:
  destination unreachable: 1
 0 ICMP messages sent
 0 ICMP messages failed
 ICMP output histogram:
Tcp:
 13 active connections openings
 0 passive connection openings
 0 failed connection attempts
 0 connection resets received
 0 connections established
 373 segments received
 270 segments send out
 1 segments retransmitted
 0 bad segments received.
 3 resets sent
Udp:
 18910 packets received
 0 packets to unknown port received.
 0 packet receive errors
 8984 packets sent
TcpExt:
[root@khahn650 rc3.d]#
```

그러나, 리눅스의 경우 rpc.nfsd를 여러개 띄우는 것이 불가능하진 않지만, 아직 완전하게 테스트가 끝난 사항이 아니므로 권장 사항은 아닙니다. 유닉스 시스템에 따라서는 rpc.nfsd 대신에 nfsd라고도 합니다.

- rpc.lockd : 파일 잠금(lock)을 통해 여러 명이 동시에 한 파일을 수정하는 것을 막을 때 사용하지만, 리눅스에서는 실행되지 않습니다.
- rpc.statd : 파일 잠금의 해제와 복구를 담당합니다. 위의 데몬들이 제대로 실행되었는지를 확인하고 싶다면, 우선 rpcinfo -p 명령을 통해서 rpc.mountd와 rpc.nfsd가 portmap에 제대로 등록되었는지를 확인해 보면 됩니다. 이외에도 사용자마다 디스크사용량에 제한을 두는 quotad 등이 작동하게 됩니다.

```
한글터미널 <4>
[root@khahn650 rc3.d]# rpcinfo -p
프로그램 버전 원형 포트
100000 2 tcp 111 portmapper
100000 2 udp 111 portmapper
100021 1 udp 1024 nlockmgr
100021 3 udp 1024 nlockmgr
100024 1 udp 1000 status
100024 1 tcp 1002 status
100011 1 udp 617 rquotad
100011 2 udp 617 rquotad
100005 1 udp 625 mountd
100005 1 tcp 627 mountd
100005 2 udp 630 mountd
100005 2 tcp 632 mountd
100003 2 udp 2049 nfs
[root@khahn650 rc3.d]#
```

등록된 이름은 실제 데몬 이름과 약간 다를 수 있습니다. 이를테면 rpc.mountd가 mountd로, rpc.nfsd가 nfsd로 등록될 수도 있습니다. 위에서와 같이 ps -aux(AT&T 계열에선 ps -ef)명령을 통해서 해당 데몬이 실행되고 있는지 확인해 보면 됩니다. 때로는 해당 프로세스가 제대로 작동하지 않아도 portmap에 등록되어 있을 수 있기 때문입니다.

파일시스템 공유 설정하기

대부분의 유닉스에서는 /etc/exports 파일에 어느 디렉토리(또는 파일 시스템)를 어떤 제약을 두어 공유할 것인가하는 내용을 적어두게 됩니다. 아무에게나 자신의 파일 시스템을 마구 사용할 수 있게 하는 것은 마치 대문을 활짝 열어두고 사는 것과 같은 일이기 때문입니다. exports 파일의 형식은 공유하는 디렉토리를 왼쪽에 적고 그 디렉토리와 관련된 옵션이나 속성을 오른쪽에 적어주는 것으로 시작합니다. 아래의 예를 보겠습니다.

```
# vi /etc/exports

/home/ftp/dists khahn(rw,no_root_squash) kkb(ro)
/home/httpd khahn(ro) kkb(ro)
```

이 exports 파일은 /home/ftp/dists 라는 디렉토리를 khahn이라는 호스트가 읽고 쓸 수 있고 루트의 권한으로 접근하는 것도 허가한 것이며, kkb라는 호스트는 읽기 전용으로 마운트할 수 있게 설정한 것입니다. 그리고 /home/httpd 디렉토리에 대해서는 khahn, kkb 호스트가 읽기 전용으로 마운트할 수 있게끔 설정한 것입니다.

아래의 내용은 리눅스의 exports 옵션들입니다.

옵션	설 명
ro	읽기만 가능하도록 마운트
rw	읽고 쓰기가 가능하도록 마운트
no_root_squash	루트의 자격으로 파일 시스템에 접근할수 있도록 마운트
root_squash	루트의 자격으로 파일 시스템에 접근하면 anonymous uid/gid로 바꾸어서 허가
noaccess	디렉토리를 접근하지 못하게 합니다. 공유된 디렉토리의 특정 하위 디렉토리만을 접근하지 못하도록 제한하고 싶을 때 유용합니다.

리눅스의 /etc/exports파일 형식은 다른 유닉스 시스템에 비해 특이한 편입니다. 일반적인 유닉스 시스템의 exports 파일은 옵션 사이엔 쉼표(.)를 쓰고 호스트를 나열할 때는 콜론(:)을 써서 다음과 같이 나타냅니다.

```
/home/ftp/dists -access=khahn, rw=kkb, root=khahn
/home/httpd -access=khahn:kkb, ro=khahn:kkb
```

exports 파일의 옵션도 유닉스마다 약간 다른데, 자주 쓰이는 옵션을 정리하면 다음과 같습니다. NFS는 물리적 파일 시스템을 대상으로 하는 것이 아니라 논리적 파일 시스템을 대상으로 합니다. 즉, 물리적인 파일 시스템에 구애받지 않고 어떤 디렉토리도 공유할 수 있다는 말입니다. 하위 디렉토리에 다른 파티션이 존재하더라도, 이를 전혀 신경 쓰지 않아도 됩니다.

exports 파일에 아무런 호스트도 지정하지 않고 단순히 공유할 디렉토리만 적어주게 되면 그 디렉토리는 “모든”호스트가 마운트 할 수 있게 됩니다. 이 경우 보안상의 문제를 불러 일으킬 수 있으므로 특히 주의해야 합니다.

Solaris의 경우 /etc/exports 대신 /etc/dfs/dfstab파일을 씁니다. 이 파일은 환경 설정파일이라기 보다는 share명령을 실행하는 쉘 스크립트입니다. share명령에 쓰이는 옵션은 SunOs의 exports 옵션과 유사합니다. 예를 들면,

```
share -F nfs -o rw=khahn:kkb, root=khahn /home/ftp/dists
share -F nfs -s ro=khahn:kkb /home/httpd
```

exports 파일을 바꿔도 당장 그 효력을 발생하지는 않습니다. exports 파일을 바꾼 후에 rpc.mountd가 설정파일을 다시 읽도록 해주어야 합니다. /etc/exports 파일을 수정한 다음 이를 반영하는 절차는 유닉스 시스템에 따라 다릅니다.

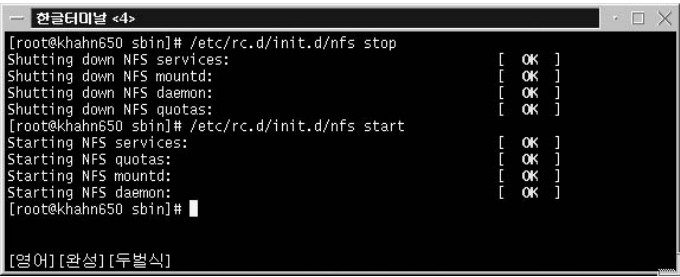
리눅스는 exports 명령이 없기 때문에 kill명령으로 rpc.mountd와 rpc.nfsd에 SIGHUP시스널을 보내야 합니다. 다음과 같은 스크립트를 /usr/sbin/exportfs라는 이름으로 저장하는 것도 한가지 방법이 될 수 있습니다.

```
# vi /usr/sbin/exportfs

killal -HUP /usr/sbin/rpc.mountd
killal -HUP /usr/sbin/rpc.nfsd
echo re-exported file system

# chmod 755 exportfs
```

와우리눅스 7.1의 경우 아래와 같이 실행하면 위와 같은 결과를 나타냅니다.



(2) SAMBA 서비스 활용하기



① Samba를 사용하는 이유

Samba는 리눅스와 다른 운영체제를 서로 연결해서 서로간의 파일이나 프린터 등을 공유할 때 사용됩니다. 그리고, Windows 98/NT/2000 등의 탐색기에서 리눅스에 있는 파일들을 보이게 하고 싶어질 때 사용할 수 있습니다. 이렇게 리눅스에서 Windows 운영체제를 위한 공유설정이나 프린터설정 등을 하고 싶을 때 , 우리는 삼바(samba)라는 프로그램을 사용하게 됩니다. 일반적으로 가장 많은 사용자를 가지고 있는 운영체제인 Windows 98/NT/2000등과 리눅스를 연결하는 방법을 설명해 보겠습니다.



참고로 1.9.x 대 버전의 samba는 /etc/smb.conf 라는 파일을 vi, pico, emacs와 같은 에디터로 직접 수정하면서 설정을 해주었지만 2.x.x 대 버전에서는 SWAT라는 웹브라우저에서 관리하는 툴을 이용해서 설정을 하게 됩니다. 따라서 복잡한 수동설정이 아닌 SWAT를 사용한 쉬운 방법을 소개하도록 하겠습니다.

② 삼바(Samba)의 최신버전 구하기

Samba 의 최신 버전을 구하는 방법은 http://www.samba.org 라는 홈페이지에서 다운로드 받는 방법입니다. 이곳에 가면 각 나라마다 ftp 사이트를 링크시켜 놓았습니다. 물론 한국도 있습니다. 와우리눅스 7.1 에는 samba-2.0.8-0.7.1이 기본 설치 됩니다.

이 파일들은 소스파일 형태로도 배포가 되고, 이미 컴파일된 바이너리 형태로도 배포가 됩니다. 소스형태로 배포되는 파일은 보통 tar 와 gz 라는 확장자를 가지고 있습니다. 또 한가지 방법은 각 PC통신사의 리눅스 동호회의 자료실이나, 공개자료실 같은 곳을 이용하는 방법입니다.

③ 바이너리 버전의 samba 설치하기

① samba 패키지 점검

rpm 버전의 패키지가 처음부터 설치되는 경우도 있지만, 그렇지 않은 경우도 있을 수 있습니다. 따라서 rpm 명령을 사용해서 먼저 점검을 해 봅니다. 혹은 X-Window 에서는 kpackage와 같은 유틸리티를 사용합니다.

아래명령으로 이전 버전의 samba 가 설치되어 있는지 점검할 수 있습니다.

```
[root@localhost /root]# rpm -q samba
samba-2.0.8-0.7.1
[root@localhost /root]#
```

만일 여러분의 시스템에 설치된 1.X 버전은 SWAT라는 웹브라우저를 이용한 관리툴을 포함하지 않은 버전입니다. 따라서 2.x대의 버전을 설치하도록 합니다. 먼저 제거를 합니다. 제거는 rpm -e samba-1.9.* 라는 명령으로 하면 됩니다.

② samba 바이너리 복사해오기

samba-2.0.8.tar.gz 버전이나 rpm 버전을 복사해 옵니다. 이 버전의 용량은 약 2.09 MB 정도 됩니다. 한국의 미러링 사이트는 ftp://cair-archive.kaist.ac.kr/pub/samba/ 라는 사이트입니다.

③ 설치된 패키지 확인

samba 가 제대로 설치되었는지를 확인합니다. samba 가 제대로 설치되었다면 웹브라우저로 관리하는 방식인 SWAT를 알아보도록 하겠습니다. 참고로 samba 와 swat가 위치한 디렉토리는 /usr/sbin 디렉토리입니다.

```
#rpm -qa | grep samba
```

④ 웹브라우저에서 samba 설정하기

samba 패키지가 설치되었다면 SWAT라는 관리툴이 설치되었다고 생각해도 됩니다. SWAT는 Samba Web Administration Tool 의 약자이며 /etc/smb.conf 라는 파일을 웹 브라우저에서 수정하고 설정에 따라서 실행 등을 할 수 있는 매우 유용한 툴입니다. SWAT를 사용해서 현재 리눅스서버에서 활동하고 있는 사람들을 모니터링 할 수 있으며, samba를 시작하고 정지, 재시동 시킬수 있는 역할도 해 줍니다. 웹서버를 운영하고 있지 않아도 SWAT는 사용할 수 있습니다. 보통은 901번 포트를 사용하게 됩니다.

① SWAT 조정하기

samba 패키지를 설치하면 SWAT패키지는 설치되었다고 할 수 있지만, SWAT를 위해서 사전에 아래와 같은 몇가지 사항들에 대해서 조치를 취해 놓아야 웹브라우저에서 설치가 가능합니다.

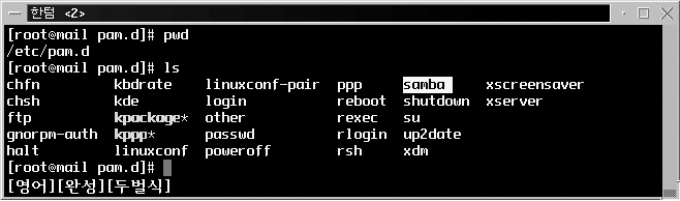
㉠ /etc/services 파일 수정

/etc/services파일은 리눅스에서 사용되는 여러 가지 서비스들을 정의해 놓은 파일입니다. 만약에 아래와 같은 내용이 없다면 한줄을 써 줍니다. 이것은 SWAT를 웹서버가동 여부에 상관없이 접근할 수 있도록 리눅스에서 901포트를 열어놓는 것입니다. /etc/services 파일에는 samba 뿐만 아니라, 다른 서비스- telnet, ftp, http-를 위한 포트도 설정되어 있습니다. 만약 901번 포트를 다른 서비스 프로그램에서 사용하고 있다면, 1024번 이후의 충돌되지 않는 번호로 변경해 주면 됩니다.

㉡ /etc/xinetd.d/swat 파일수정 파일에서 'disable=no' 로 변경합니다. 이 설정이 마쳐졌다면 이제 xinetd를 재 시작합니다.

㉢ PAM(Pluggable Authentication Module) 인증모듈의 사용

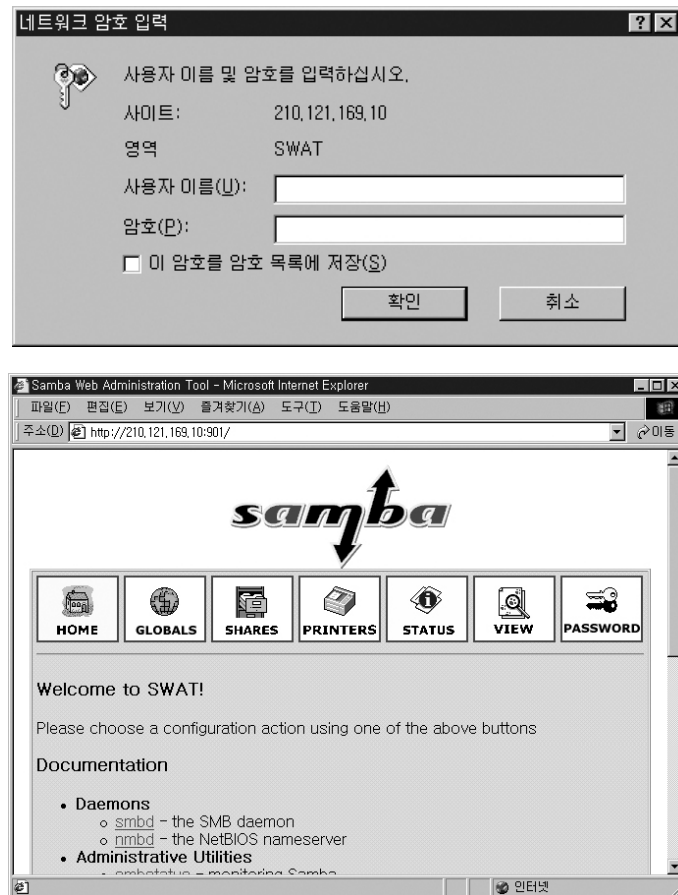
PAM 인증모듈 이라는 것은 /etc/pam.d 라는 설정파일에 어떠한 프로그램의 인증방식을 적어놓고 그것과 상호 교환하면서, 새로운 인증방식이 나왔을때도 프로그램의 인증부분을 유연하게 대처하도록 해주는 역할을 하도록 하는 것입니다. 보통은 /etc/pam.d 라는 디렉토리 밑에 해당되는 설정들을 넣어두게 됩니다. 우리가 사용할 SWAT는 /etc/pam.d/samba 라는 파일로 그 설정을 저장하고 있습니다. samba 이외에도 여러 가지 인증을 위한 설정파일들이 들어 있습니다.



samba를 위한 설정을 위해서 /etc/pam.d/samba 파일을 열어 봅니다.. 만약 이 파일이 없다면 root로 로그인할 수 없어서 SWAT로 samba 설정을 하는 것은 불가능하게 됩니다.

㉔ 웹 브라우저를 시작한 후 리눅스의 SWAT로 접속하기.

이제부터는 자주 사용하는 웹 브라우저로 설정에 들어가도록 하겠습니다. URL을 적어주는 곳에다가 `http://210.121.169.10:901/` (여러분의 IP) 이라고 적어 주었습니다. 접속하면 아래 그림과 같은 네트워크 암호를 입력하는 화면이 나옵니다. 여기서 사용자 이름(U)에 `root`를 넣고 암호(P)에는 리눅스 `root`의 로그인 암호를 적습니다. 인증을 통과하면 SWAT의 시작화면이 나옵니다.



여기서 주의할 점은 `/etc/samba.conf` 파일을 다른 곳에 미리 복사해 놓으라는 것입니다. 웹브라우저에서 SWAT로 설정들을 바꾸게 되면 `/etc/samba.conf` 라는 파일이 변경되게 됩니다. 비록 설정사항들을 SWAT로 웹브라우저에서 변경한다고 해도 `/etc/samba.conf` 파일이 변경되는 것입니다. 따라서 설정에 들어가기 전에 미리 변경해 놓는 것이 나중을 위해서 안전할 것입니다.

② SWAT를 사용한 samba 설정

이제부터는 웹 브라우저로 samba 설정을 변경하는 절차를 알아보도록 하겠습니다. 많은 경우 디폴트로 설정된 사항들을 거의 그대로 두고 작동시키기도 하지만, 수정을 할 경우에는 자신이 무엇을 수정했는지는 정도는 알고 있어야 합니다. 무턱대고 아무 것이나 변경한다면 처음부터 다시 시작해야 할

수도 있습니다.

SWAT관리 화면은 마치 홈페이지의 한 화면과 같습니다. 크게 두 부분으로 나뉘어 있는데 첫 번째는 7개의 아이콘형태의 버튼이 있는 부분이고, 두 번째는 samba 설정을 위한 여러 가지 문서들과 mailing list 에 가입하라는 안내문이 있는 부분입니다. 우리가, 여기서 관심을 가져야 할 부분은 첫 번째 영역의 버튼 부분입니다. 버튼을 누르면 홈페이지에서처럼 링크된 설정영역으로 이동하게 됩니다. 이제부터는 버튼을 중심으로 각 설정영역을 설명하겠습니다.

① HOME 버튼

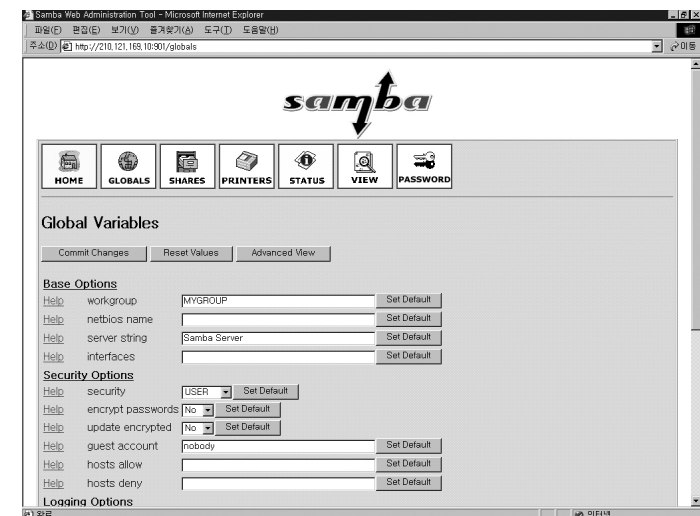
이 버튼을 클릭 하면 맨 처음 나타났던 화면들을 다시 볼 수 있습니다. samba의 설정과는 아무런 관련이 없다고 할 수 있습니다.



② Globals 버튼



이곳이 samba 서버의 설정에 있어서 가장 중요하다고 할 수 있습니다. `/etc/smb.conf` 파일의 여러 가지 설정들을 변경 할 수 있게 됩니다. 처음에 구동되면 아래 그림과 같은 화면을 보여주는데, 3개의 버튼과 여러 가지 항목들로 이루어져 있습니다. [Commit Changes] 버튼은 변경 후에 변경 사항들을 반영하도록 저장하는 버튼이고, [Reset Values] 버튼은 작성된 항목들을 초기화시키는 버튼입니다. 그리고, [Advanced View] 버튼을 누르게 되면 처음화면보다 더욱 확장된 설정화면을 보여주게 됩니다. 여기서는 basic view 옵션을 가지고 설명을 하겠습니다.



Advanced View를 눌렀을 경우 GLOBALS 항목은 크게 13개의 옵션영역으로 구성됩니다. 이제 부터는 각 영역에 해당되는 옵션을 살펴보도록 하겠습니다. 각 옵션에 해당되는 설정들은 기본설정 과 필자의 경우를 비교해서 써 놓았습니다. 독자들은 자신에게 맞도록 설정을 변경해 나가면 될 것 입니다.

☞ Base Options

WorkGroup - 탐색기에서 나타날 이름을 설정하는 부분입니다.

기본설정 = WORKGROUP | 이곳에서는 = wowlinux

netBIOS name - samba 서버에 알려질 NetBIOS의 이름을 설정하는 곳입니다. 기본설정은 호스트의 DNS명과 동일합니다. 만약 브라우징서버(서버의 이름을 다른 클라이언트들에게 서비스하는 컴퓨터) 또는 로그온 서버라면 그 이름을 써 주면 됩니다.

기본설정 = DNS 네임

netbios aliases - nmbd 는 samba 서버에 알려진 이름들에 의해서 NetBIOS 이름의 리스트들을 다른 곳에 알려주는데, 이때 하나의 클라이언트에서 여러 가지 다른 이름으로 나타날 수 있도록 해 줍니다. 만약 서버가 브라우징 서버나 로그온 서버로 사용된다면 이 이름들은 다른 서버에 알려질 때와는 달라야 될 것입니다.

기본설정 = 없음 | 예제 = Exam Exam1 Exam2

server String - Windows의 탐색기에서 설명부분에 나타날 samba 서버의 이름에 해당됩니다.

기본설정 = 없음

interfaces - 이 옵션은 사용자가 Lan 카드를 2개 꽂았을 경우에 multiple 네트워크 인터페이스를 사용할 수 있도록 samba 가 조절해 주는 옵션입니다. 이 옵션은 ip/netmask 의 쌍으로 이루어지게 됩니다. 사용자가 C 클래스 네트워크에 속해 있다면 /24를 입력해 주면 됩니다.

기본설정 = 없음

☞ Security Options

security - 이 옵션은 사용자가 samba 서버에게 어떻게 응답할 것인가를 설정하는 매우 중요한 옵션중의 하나입니다. SHARE, USER, SERVER, DOMAIN 등 4개의 기본설정이 있으며, Windows 98/NT/2000 등을 위해서는 USER 옵션이 기본설정으로 되어 있습니다. 이전 버전에서는 SHARE 옵션이 기본설정이었습니다. 하지만, 사용자를 인식하는 부분에서 약간의 버그가 있어서 지금은 USER로 기본설정이 변경되었습니다.

기본설정 = USER

encrypt passwords - Windows 98/2000, Windows NT(서비스팩3) 등에서 레지스트리에 별다른 변경을 하지 않았다면 암호화해서 신호를 주고 받는데, 이렇게 암호화해서 신호를 주고 받을 것인지 물어 보는 항목입니다. 더 자세한 내용은 samba 패키지 내의 ENCRYPTION.txt 라는 문서를 읽어 보면 됩니다.

기본설정 = No

update encrypted - 사용자가 samba 서버로 접근했을 경우 암호화된 비밀번호를 자동으로 갱신해 줄 것인지를 물어보는 옵션입니다.

기본설정 = No

guest account - samba 서버에 접근하는 사용자가 손님일 경우 손님의 ID를 nobody로 할 것인지 guest 로 할 것인지를 결정합니다.

기본설정 = nobody

hosts allow - 리눅스에서 외부의 접속에 대한 정책을 결정하는 파일과 같이 접속을 허용할 컴퓨터의 IP어드레스의 형식으로 써 줍니다.

기본설정 = 없음

hosts deny - 접속을 거부할 리스트를 적는 곳입니다.

기본설정 = 없음

☞ Logging Options

samba 서버에 사용자들이 접속했을 때 흔적들을 남기는데 이 영역에서는 그러한 흔적들을 추적하는 옵션들을 지정하는 영역입니다.

log level - debug level 과 유사한 역할을 하는 것으로 리눅스에서 유연하게 변경될 수 있습니다.

기본설정 = 1

log file - 사용자들이 접속하고 사용한 흔적들이 여기서 지정해준 파일에 남아있습니다. 누가 samba 서버에 접속했는지 알 수 있습니다. %m 은 machine name을 나타내는 변수입니다.

기본설정 = /var/log/samba/log.%m = /var/log/samba/log.%m

max log size - 이것은 log 파일의 사이즈를 결정하는 것으로 단위는 킬로바이트입니다. log 파일이 무한정 많이 생성된다면, 사람들의 빈번한 접속이 log 파일에 기록되어서 서버를 힘들게 할 것입니다. 가급적 100 KB 이내로 제한하는 것이 좋다.

기본설정 = 50

☞ Protocol Options

여기서는 samba 서버를 다른 Windows 클라이언트들에게 어떤 모습으로 보여지게 할지를 결정하는 옵션을 적는 곳입니다.

announce as - 기본설정은 NT로 되어있고, NT Server, NT workstation, Win95, WfW(윈도우 for 워크그룹 운영체제) 등의 옵션이 있습니다.

기본설정 = NT

☞ Tuning Options

socket options - 다른 클라이언트들과 통신을 할 때 소켓을 설정하는 것을 허용하는 옵션입니다. 자세한 도움말은 man setsockopt 이라는 명령으로 찾아 볼 수 있습니다. 여기서는 설정사항을 디폴트로 두고 변경하지 않았습니다.

☞ Browse Options

os level - samba 서버가 외부에 알려질 때 자신이 무엇으로 알려질지 순위를 정하는 옵션입니다. 이 설정으로 인해서 NT 서버의 접근처럼 허용될 수 있도록 다른 하위접속은 무시할 수 있도록 됩니다. 이 옵션은 같은 워크그룹에 있는 사용자에게 해당됩니다.

기본설정 = 0

preferred master - nmbd 가 자신의 워크그룹 내에서 마스터 브라우저로 작동할 것인지 아니지를 설정하는 옵션입니다. 마스터브라우저로 설정해 놓으면 컴퓨터를 찾느라고 전반적으로 접속이 느려질 수 있습니다.

기본설정 = NO

local master - 같은 서브넷에 있는 컴퓨터들의 마스터 브라우저로 작동하는 것을 도와줍니다. 그 자체가 마스터 브라우저가 된다는 말은 아닙니다.

기본설정 = YES

domain master - 위의 2가지 옵션과 비슷한 의미로, 도메인 내부에서 마스터 브라우저로의 작동을 nmbd 로 도와준다는 말입니다.

기본설정 = NO

☞ WINS Options

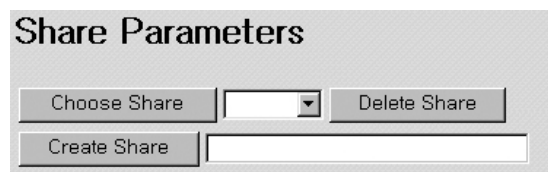
이 옵션은 Windows 시스템의 TCP/IP 설정부분에서 보았을 것입니다. WINS서버로 쓰고 있는 컴퓨터가 없으며 Proxy도 안 쓸 경우 수정할 필요가 없습니다. 만약 WINS 서버가 있는 사람이라면 사용한다고 samba 서버에 알려 주어야 할 것입니다.

이제 GLOBALs 아이콘에 대한 설명을 마쳤습니다. [Advanced View]라는 버튼을 누르게 되면 더욱 더 세세한 설정을 할 수가 있습니다. 하지만, 기본설정만으로 samba 서버를 이용하는데, 별 문제는 되지 않습니다.

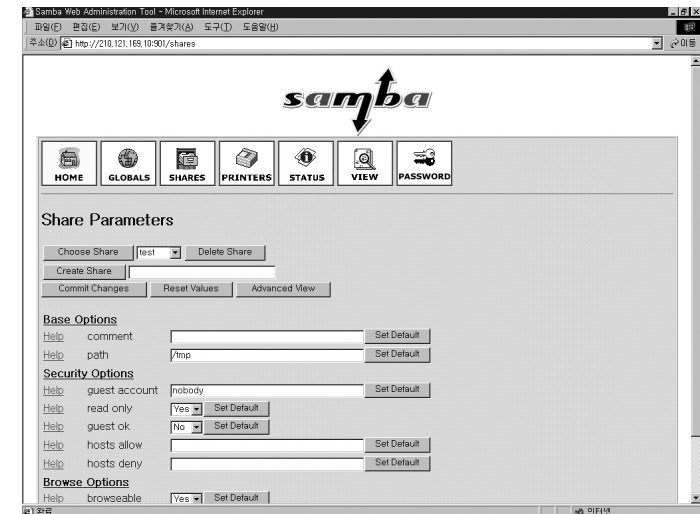
㉔ SHARES 버튼



이곳은 공유할 방을 만들고 관리하는 영역입니다. 처음 들어가면 그림과 같은 화면이 나타나게 됩니다. 여기서 [Create Share] 라는 버튼으로 test 라는 방을 하나 만들어 놓고 그것을 가지고 작업하도록 하겠습니다.



그림에서 보는 바와 같이 test라는 방을 위한 4개의 옵션영역이 생기게 됩니다.



☞ Base Options

comment - 이곳은 윈도우 탐색기에서 나타날 때 설명부분이 됩니다.

기본설정 = 없음

Path - 디렉토리의 경로입니다.

기본설정 = /tmp

☞ Security Options

guest account - 손님 자격으로 방을 열어볼 사람들의 기본 ID 입니다.

기본설정 = nobody

read only - 디렉토리를 읽기 전용으로만 둘 것인지를 결정합니다. 만약 NO 라고 써준다면 쓰기도 가능한 상태가 될 것입니다.

기본설정 = YES

guest ok - 손님 자격을 가진 사람이 들어오는 것을 허용할지 안 할지를 선택하는 옵션입니다.

기본설정 = NO

hosts allow, hosts deny - 이것은 GLOBALS 버튼에서 알아본 것과 같은 의미입니다.

☞ Browse Options

browseable - 다른 사용자에게 보여질지 아닐지를 결정하는 옵션입니다. 보여지는 것이 나올 것입니다.

기본설정 = YES



☛ Miscellaneous Options

available - 이 옵션은 잠시 사용을 중지시킬 때 사용하는 옵션입니다. 전원스위치로 치자면 전원을 ON/OFF시키는 것과 같은 이치입니다.

기본설정 = YES

모든 설정을 마친 후에는 [Commit Changes] 라는 버튼을 눌러서 정보를 갱신합니다.

Ⓜ PRINTERS 버튼

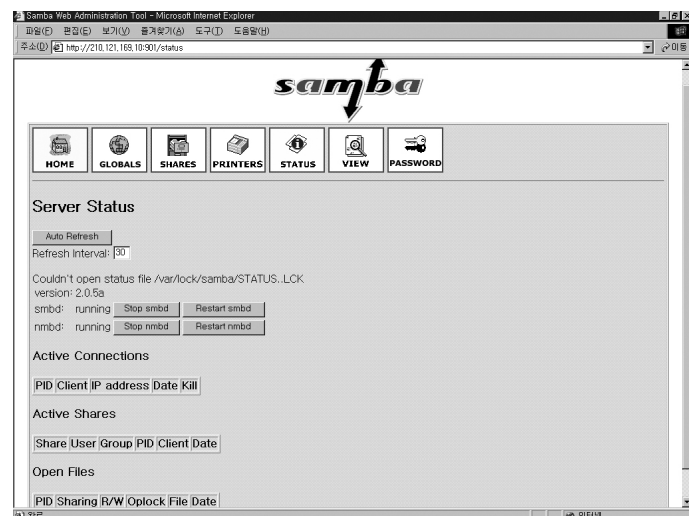


이곳은 프린터를 공유하기 위해서 설정하는 부분입니다. 프린터의 설정도 역시 [SHARES] 아이콘의 설정과 별로 다르지 않습니다. 여기서는 HP692K 프린터를 공유하기 위해서 HP692K라는 공유이름을 만들었습니다. 여기서도 모든 설정을 마친 후에는 [Commit Changes] 라는 버튼을 눌러서 정보를 갱신해 주어야 합니다.

Ⓜ STATUS 버튼



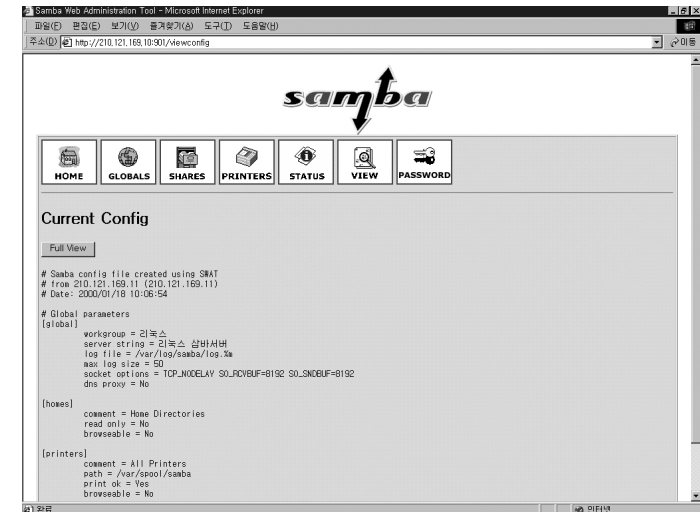
이제는 모든 설정이 마쳐진 것입니다. 아래 그림은 모든 설정이 마쳐진 상태에서 samba 서버가 운용되고 있는 상황을 표시하는 옵션입니다. 홈디렉토리로 ahn 이라는 ID로 접근하고 있다는 정보를 보여줍니다. 이와 같이 다른 사람이 접근할 때도 화면에 그 상황을 보여주게 됩니다. 그리고, [stop smbd] [Restart smbd] 라는 버튼이 있는데, 이것은 말 그대로 samba 서비스를 정지 할 때나 재시작 할 때 사용하는 옵션입니다.



Ⓜ View 버튼



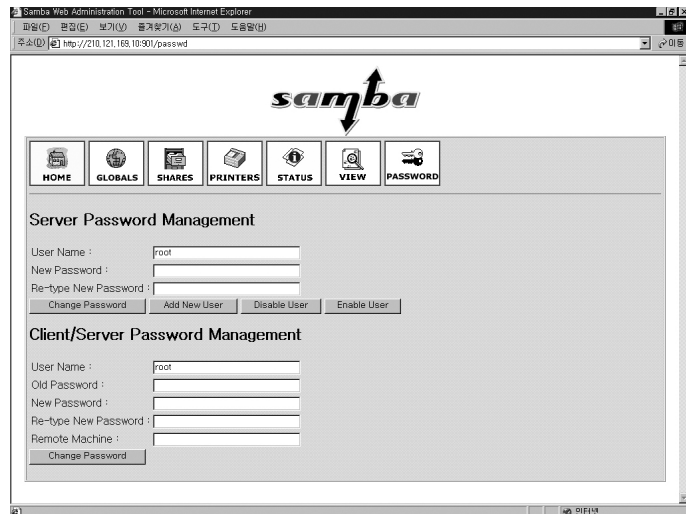
이것은 지금 까지 설정한 /etc/smb.conf 파일의 내용을 보여주는 것입니다. [Full View] 라는 버튼을 클릭하면 자세한 내용이 나오게 됩니다. 눈치가 빠른 사람은 알겠지만, SWAT라는 유틸리티로 설정하는 것은 /etc/smb.conf 라는 파일을 수정하는 것과 같은 효과가 나타난다는 것을 알 수 있을 것입니다.



Ⓜ PASSWORD 버튼



패스워드를 설정하는 영역입니다. 여러분들이 알고 있어야 할 점은 리눅스에 계정이 없으면 samba의 홈디렉토리를 사용할 수 없다는 것입니다. 즉, 패스워드 설정은 리눅스 사용자에게만 해당되는 내용이라는 것입니다. [GLOBAS] 영역에서 samba의 패스워드를 저장해 놓은 경로변경을 하지 않았다면 /etc/smbpasswd 라는 파일이 samba의 패스워드 파일입니다. 여기서 수정되는 내용들은 /etc/smbpasswd 라는 파일에 반영되게 됩니다. 그림에서 보는 것처럼, 새로운 사용자를 추가하고, 삭제하고, 사용을 일시 정지시킬 수도 있도록 되어 있습니다.



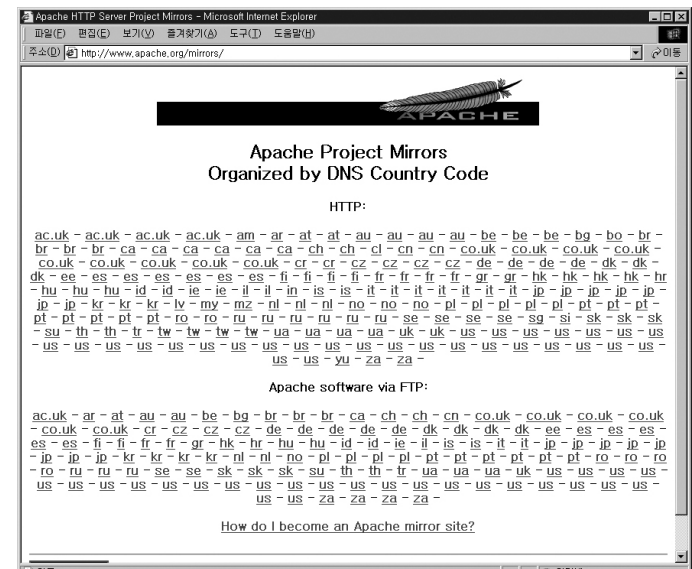
이제는 리눅스에서 할 수 있는 samba의 모든 설정을 마쳤습니다. 마지막으로 [STATUS] 버튼을 눌러서 Restart 버튼을 클릭하면 설정사항들이 즉시 반영됩니다.

6. 웹 서비스

(1) 아파치 서버에 관한 소개

우리가 리눅스를 통해서 서비스할 수 있는 것 중에서 웹서비스가 있습니다. 최근에 대형 ISP들의 웹서비스가 해커들의 DoS(서비스 거부) 공격으로 인해서 마비가 된 적이 있는 그러한 서비스입니다. 이러한 웹서버로 가장 많은 사용자를 확보하고 있는 것이 아파치 웹서버입니다. 아파치 이외에도 다른 상용 서버 프로그램들이 있지만, 아파치가 가장 많은 사용자를 확보하고 있습니다. 또한 상대적이기는 하지만, 무료로 얻을 수 있는 정보도 상당수 가지고 있습니다.

아파치는 1995년에 나왔습니다. 그 당시에 가장 많은 홈페이지에서 사용되고 있던 NCSA HTTPD 1.3 버전을 모체로 하여서 만들어 졌습니다. 현재에도 많은 부분이 유사하게 동작되고 있습니다. 97년 4월에 NetCraft라는 곳에서 조사한 바에 의하면 전세계의 웹서버 가운데, 45 % 이상의 웹서버가 아파치로 운영되고 있다고 합니다. 아파치는 이러한 현상을 반영하듯 전세계에 미러링되고 있습니다. 아래 그림은 아파치서버가 미러링 되고 있는 나라들입니다.



아파치는 현재까지 1.3.20 까지 나와있으며, 1.2 버전들은 안정화를 거친 안정버전들입니다. 만약 기존에 버전이 낮은 아파치서버를 운영하고 있다면, 보안문제나 여러 가지 약점들이 있을 수 있으므로 업데이트를 검토해 보는 것이 바람직할 것입니다.

아파치 서버는 대부분의 운영체제에 맞도록 바이너리와 소스 코드를 배포하고 있습니다.

(2) HTTP 와 TCP/IP

HTTP(Hyper Text Transfer Protocol)는 WWW(World Wide Web)에서 사용하고 있는 데이터 송수신 프로토콜로서 OSI 7계층 중에서 4번째 트랜스포트 계층에 UDP와 같이 자리하고 있는 프로토콜입니다. IP는 3번째 계층인 네트워크계층에 속하고 있습니다.

웹 에서는 TCP가 중요한 역할을 하는데, 그것은 사진이나 압축파일과 같은 이진파일들(바이너리파일들)을 8-bit 이진모드로 전송할 수 있다는 것입니다. 데이터를 전송할 때 이진파일은 완전전송모드(8-bit를 기본으로 전송함을 말함)로 전송하고 텍스트 파일들은 7-bit 아스키모드로 전송을 하게 됩니다. 그래서 실행파일이나, 그림 파일과 같은 바이너리는 반드시 8-bit모드로 전송을 해야 되는데, 그러한 역할을 도와주는 것이 아파치와 같은 웹서버가 하는 일입니다.

이렇게 전송하는 방법에 관한 규칙을 담고 있는 것이 HTTP규약인데, 현재는 HTTP/1.1 규약까지 나와 있습니다. TCP/IP 프로토콜에서는 TCP 전송 프로토콜을 이용하는 응용 프로토콜들 사이의 구별을 port 번호를 통해서 해결하고 있습니다. 보통 아파치 서버 같은 웹서버에서는 80번 TCP 포트를 HTTP를 위해서 디폴트로 할당하고 있습니다. HTTP/1.0 은 IETF (Internet Engineering Task Forces)에 의해 RFC 1945로 정식 등록되었으며, HTTP/1.1은 RFC2068에 등록되어 있습니다.

(3) 아파치서버의 설치

① 최신버전의 아파치서버 프로그램 다운로드

최신버전의 아파치 프로그램은 <http://www.apache.kr.net/dist>에서 받아 오면 됩니다.
<http://www.apache.kr.net/dist>

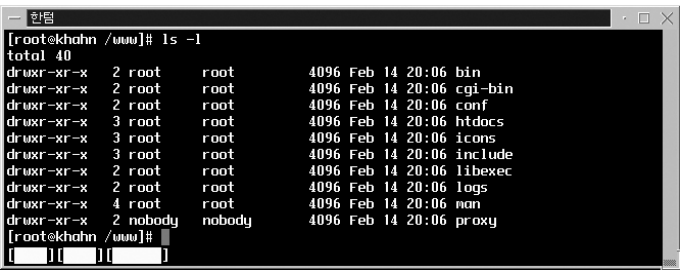
② 압축의 해제

`tar xzvf apache_1.3.20.tar.gz`
라는 명령으로 압축을 해제하고 나서 소스가 풀린 디렉토리로 이동합니다. 이곳에서 약간의 수정을 가해서 컴파일하게 되면 자신만의 아파치서버를 가질 수가 있습니다.

③ 아파치가 설치될 디렉토리를 지정하는 명령과 모듈소스 동시 컴파일

여기서는 /www 라는 디렉토리를 아파치의 기본디렉토리로 설정하였으나, 여러분들은 임의로 설정해도 됩니다. 참고로 와우리눅스 7.1 배포판에 있는 아파치 서버를 그대로 설치했다면 메인 디렉토리는 /var/www이 되고, 설정파일들은 /etc/httpd 아래에 위치하게 됩니다.
하지만, 임의로 디렉토리를 지정해 주었을 경우에는 아래 그림과 같은 디렉토리 구조를 가진다. 가급적 기존에 설치된 아파치 서버는 지우고 새로 컴파일 해서 최신 버전을 사용하도록 합니다.

```
# ./configure --prefix=/www
```



여기예다가 사용자가 사용하기를 원하는 모듈을 같이 컴파일 하게끔 모듈소스가 있는 위치를 지정해 줍니다.

```
# ./configure --prefix=/www/ --add-module=[모듈소스 위치]
```

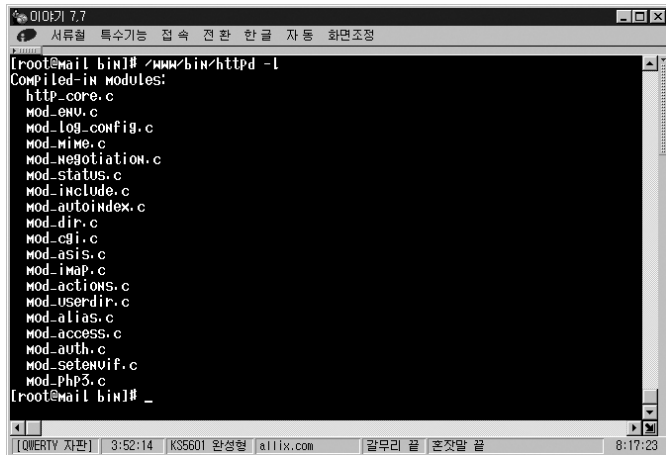
이렇게 하면 모듈소스가 같이 컴파일 될 수 있도록 위치를 잡아 줍니다.
만약 여러개의 모듈을 같이 첨가 하려면 add-module을 계속 반복해서 원하는 만큼 아파치 서버 모듈을 추가할 수 있습니다.

모듈이란?

아파치에서 뿐만 아니라 컴퓨터에서 사용되는 모듈이라는 단어의 의미는 메인 프로그램이나 하드웨어에 보조적으로 쓰이는 라이브러리나 프로그램 집합을 말합니다. 우리가 이번 글에서 살펴보게 될 아파치 모듈이라는 것도 이러한 의미와 크게 다르지 않다고 보면 되겠습니다.
즉, 아파치 서버에 보조적으로 추가되어서 일반사용자들에게 서비스나 응답을 해준다는 것입니다. 가장 대표적으로 쓰이는 모듈로써는 PHP 모듈을 들 수 있겠습니다. PHP 모듈은 아파치 서버에 최적화된 스크립트 웹프로그래밍 언어입니다. Windows 시리즈에는 ASP 라는 것이 있습니다. 또, 이들 두 언어를 서로 변환해주는 툴도 있든 것을 보면 PHP 가 얼마나 많은 장점들을 가지고 있는지 알 수 있습니다.
그 외에도 perl, ssl, ApacheJServ, MySQL, oas, AuthPostgreSQL 등의 모듈등이 아파치 서버와 함께 많이 사용되고 있습니다.

④ 이제 컴파일을 하는 과정입니다. 컴파일은 설치에 필요한 목적 파일들을 생성하는 단계입니다. 간단히 아래와 같이 make 명령을 내리면 됩니다.
`# make`
configure과정에서 add-module 로 모듈을 같이 컴파일 한 경우라면 /www/src/httpd 에 해당 모듈이 추가 되었는지를 확인합니다. 아래 그림은 여러 가지 모듈들을 포함시킨 예입니다.





⑤ 이제는 마지막으로 실행파일들을 생성시키는 과정입니다.

```
# make install
```

⑥ 디렉토리 점검

만약 /var/www 라는 디렉토리에 아파치 서버의 프로그램들을 설치했다면 아래와 같이 설치되었다는 것을 확인할 수 있습니다.
각 디렉토리를 설명하자면 다음과 같습니다.

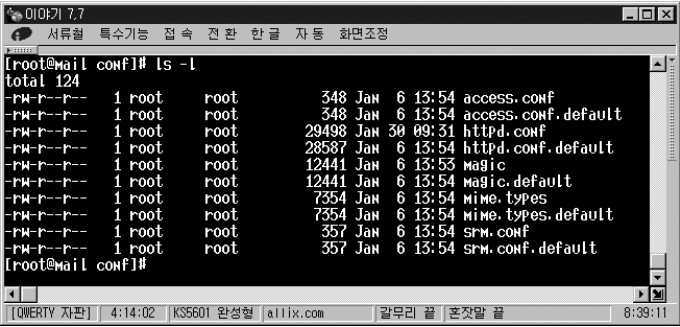
디렉토리	내 용
/www/bin	아파치 실행 파일 디렉토리
/www/cgi-bin	CGI Script 디렉토리
/www/conf	설정파일 디렉토리
/www/htdocs	웹서버 메인 디렉토리
/www/icons	아파치 서버 고유의 아이콘 디렉토리
/www/include	서버 소스파일 디렉토리
/www/libexec	실행가능한 라이브러리 디렉토리
/www/logs	웹서버 실행중에 만들게될 로그파일 디렉토리
/www/proxy	프록시 설정관련 파일 디렉토리

⑦ 아파치서버의 설정파일 수정하기

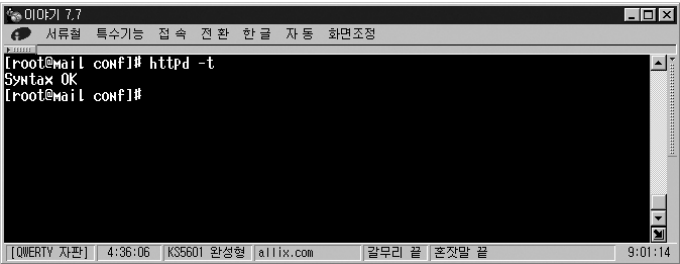
이제는 아파치 서버에서 자신이 관리하는 서버에 적합하도록 설정파일을 수정하는 일이 남아있습니다. 이 과정은 많은 사람들이 언급해 놓았으므로 자세한 사항들은 다른 문서들을 참조하기 바랍니다. 또한 여러분들이 메시지를 한글로 바꾼 아파치서버를 컴파일 하였다면 한글로 번역된 설정파일을 볼 수 있을 것입니다.
사전에 아파치 설정을 위해서 webmaster 라는 사용자를 한 명 만들어 두는 것이 좋을 것입니다.
adduser webmaster

■ /www/conf/httpd.conf 파일 수정하기

아래 그림은 아파치의 /www/conf/ 디렉토리의 내용입니다. 여기서 우리가 수정해야 할 파일은 httpd.conf 파일입니다. access.conf 파일과 srm.conf 파일은 httpd.conf 파일을 수정하면 아무것도 따로 수정할 것이 없는 파일들입니다. mime.types 파일에는 웹서버에서 반응하게 될 여러 가지 파일의 확장자 형식을 지정해 놓은 파일입니다. 초기에는 이 mime.types 가 제대로 설정이 안되어 있어서 zip 파일과 같은 확장자를 가진 파일을 홈페이지에 올려 놓으면 그대로 바이너리의 내용을 보여줘서 사용자들을 귀찮게 하곤 했습니다.



아파치 설정 파일은 전통적으로 httpd.conf, srm.conf, access.conf 이렇게 3 개로 구성되어 있습니다. 그러나 내부적으로는 3 개의 파일에 대한 구별을 따로 하지 않고 순차적으로 읽어옵니다. 새로운 버전의 아파치에서는 모든 설정파일은 httpd.conf 에 들어 있습니다.
만약 문법에 오류가 있는지 없는지 살펴보려면 -t 옵션을 주어서 문장을 검색해 볼 수 있습니다.



서버의 루트디렉토리 위치를 지정합니다.

ServerRoot “/www/htdocs”

서버를 접속한 후 얼마간 그 접속을 유지할 것인가를 지정하는 옵션

Timeout 300



사용자의 요구를 처리할 수 있는 횟수. 이것을 높여 놓으면 많은 사용자가 접속해도 서버에 무리가 가지 않습니다. 무한대로 설정해 놓으려면 0를 입력하면 됩니다.

MaxKeepAliveRequests 100

동시 접속 가능한 클라이언트의 개수를 지정합니다.

MaxClients 150

아파치에서 사용할 여러 가지 유용한 모듈을 적재하는 부분입니다. 이곳에서 우리는 여러 가지 모듈들을 적재합니다.

```
#LoadModule mmap_static_module modules/mod_mmap_static.so
LoadModule env_module      modules/mod_env.so
LoadModule config_log_module modules/mod_log_config.so
LoadModule agent_log_module modules/mod_log_agent.so
LoadModule referer_log_module modules/mod_log_referer.so
#LoadModule mime_magic_module modules/mod_mime_magic.so
LoadModule mime_module     modules/mod_mime.so
LoadModule negotiation_module modules/mod_negotiation.so
LoadModule status_module   modules/mod_status.so
LoadModule info_module     modules/mod_info.so
LoadModule includes_module modules/mod_include.so
LoadModule autoindex_module modules/mod_autoindex.so
LoadModule dir_module      modules/mod_dir.so
LoadModule cgi_module      modules/mod_cgi.so
LoadModule asis_module     modules/mod_asis.so
LoadModule imap_module     modules/mod_imap.so
LoadModule action_module   modules/mod_actions.so
#LoadModule speling_module modules/mod_speling.so
LoadModule userdir_module  modules/mod_userdir.so
LoadModule proxy_module    modules/libproxy.so
LoadModule alias_module    modules/mod_alias.so
LoadModule rewrite_module  modules/mod_rewrite.so
LoadModule access_module   modules/mod_access.so
LoadModule auth_module     modules/mod_auth.so
LoadModule anon_auth_module modules/mod_auth_anon.so
```

```
#LoadModule dbm_auth_module modules/mod_auth_dbm.so
LoadModule db_auth_module   modules/mod_auth_db.so
LoadModule digest_module    modules/mod_digest.so
#LoadModule cern_meta_module modules/mod_cern_meta.so
LoadModule expires_module   modules/mod_expires.so
LoadModule headers_module   modules/mod_headers.so
LoadModule usertrack_module modules/mod_usertrack.so
#LoadModule example_module  modules/mod_example.so
#LoadModule unique_id_module modules/mod_unique_id.so
LoadModule setenvif_module  modules/mod_setenvif.so
```

application/x-httpd-php3 .php3 이라는 PHP 스크립트를 위한 확장모듈의 설치 AddType 부분을 찾아서 주석을 해제 시켜 주어야 정상적으로 작동합니다.

아파치서버에서 Perl 모듈을 지원하기 위해서 아래와 같은 설정을 해줍니다.

LoadModule perl_module modules/libperl.so
위의 LoadModule 부분과 조합을 이루면서 사용되는 부분입니다. 위에서 Loadmodule부분과 쌍을 이루는 것을 알 수 있을 것입니다.

```
ClearModuleList
#AddModule mod_mmap_static.c
AddModule mod_env.c
AddModule mod_log_config.c
AddModule mod_log_agent.c
AddModule mod_log_referer.c
#AddModule mod_mime_magic.c
AddModule mod_mime.c
AddModule mod_negotiation.c
AddModule mod_status.c
AddModule mod_info.c
AddModule mod_include.c
AddModule mod_autoindex.c
AddModule mod_dir.c
AddModule mod_cgi.c
AddModule mod_asis.c
AddModule mod_imap.c
AddModule mod_actions.c
```

```
#AddModule mod_speling.c
AddModule mod_userdir.c
AddModule mod_proxy.c
AddModule mod_alias.c
AddModule mod_rewrite.c
AddModule mod_access.c
AddModule mod_auth.c
AddModule mod_auth_anon.c
#AddModule mod_auth_dbm.c
AddModule mod_auth_db.c
AddModule mod_digest.c
#AddModule mod_cern_meta.c
AddModule mod_expires.c
AddModule mod_headers.c
AddModule mod_usertrack.c
#AddModule mod_example.c
#AddModule mod_unique_id.c
AddModule mod_so.c
AddModule mod_setenvif.c
# Extra Modules
#AddModule mod_php.c
#AddModule mod_php3.c
#AddModule mod_perl.c
```

서버의 포트번호를 지정하는 부분입니다. 포트번호는 1024번이상의 다른 번호로 대체되기도 합니다. 80번이 아닌 다른 포트로는 8080등이 많이 사용되고 있습니다. 참고로 samba 서비스는 901번 포트를 사용하고 있습니다.

Port 80

서버의 관리자의 E-mail 주소를 적어주는 곳입니다.

ServerAdmin webmaster@[메일서버이름]

서버의 웹문서 루트 디렉토리를 지정합니다.

DocumentRoot "/www/htdocs"

아파치 서버가 접근할 수 있는 디렉토리에 관한 기본정책을 표시하는 곳입니다.

```
<Directory />
    Options FollowSymLinks
    AllowOverride None
</Directory>
```

사용자의 홈디렉토리를 지정하는 역할을 합니다. 아래와 같이 지정해 놓으면 사용자는 자신의 디렉토리에 public_html 이라는 디렉토리를 만들어 두고, 그 안에 index.html 이라는 파일을 넣어 두어야 합니다.

UserDir public_html

디렉토리에서 맨 처음 찾게 되는 문서들의 이름을 써 놓습니다. 즉, 어떠한 디렉토리이든지 간에 아래의 확장자를 가진 문서가 찾아지면 그것을 초기 페이지로 인식하고 웹브라우저에 보여주는 역할을 합니다.

DirectoryIndex index.php3 index.html index.shtml index.cgi
DirectoryIndex index.html index.shtml index.cgi

아파치 인증에 관련된 부분을 적어 놓는 파일의 이름입니다. htpasswd 유틸리티를 사용해서 인증을 하고자 할 때 .htpasswd 라는 파일과 같이 사용합니다.

AccessFileName .htaccess

아파치 사용중에 만나게 되는 에러 메시지들을 저장하는 파일을 지정하는 곳입니다.

ErrorLog logs/error_log

사용자 접근에 대한 로그파일의 위치지정. 이곳에 사용자가 웹서버(아파치서버)에 접속했던 정보들이 기록됩니다. 나중에 자신의 홈페이지에 얼마만큼의 접속이 있었는지 간접적으로 확인 할 수도 있습니다.

CustomLog logs/access_log common

Alias /www/icons라는 디렉토리를 /icons 디렉토리가 대체합니다.
Alias /icons/ "/www/icons/"


```
<Directory "/www/icons">
    Options Indexes MultiViews
    AllowOverride None
    Order allow,deny
    Allow from all
</Directory>
```

cgi 스크립트등이 실행되는 디렉토리를 지정합니다. 이곳이 중요한 설정입니다. 만약 사용자 디렉토리에서 모두 CGI가 가능하게 하려면 아래와 같이 사용자 디렉토리에 대해서도 cgi 설정을 해 주어야 합니다.

```
ScriptAlias /cgi-bin/ "/www/cgi-bin/"
<Directory "/www/cgi-bin">
    AllowOverride None
    Options ExecCGI
    Order allow,deny
    Allow from all
</Directory>
```

보통은 사용자의 홈디렉토리가 /home/ 디렉토리 아래에 위치하고 있으므로 아래와 같이 /home/디렉토리 아래를 CGI 디렉토리로 운영하겠다는 것을 알려 주어야 합니다.

```
<Directory /home*>
    Options ExecCGI Includes
    AllowOverride All
</Directory>
```

각 언어에 대해서 어떤 접미어를 가질지를 설정해 줍니다.

```
AddLanguage ko .ko
AddLanguage en .en
AddLanguage fr .fr
AddLanguage de .de
AddLanguage da .da
AddLanguage el .el
AddLanguage it .it
```

웹 문서를 만났을 경우에 어떤 언어를 먼저 선택해서 보여줄지를 결정하는 옵션입니다. 한국어, 영어, 불어 등의 순서로 서버에서 문서를 해석하려고 시도합니다.

```
LanguagePriority ko en fr de
```

원래 웹서버에서 사용되는 파일들의 확장자를 mime.type 파일에서 정의해 주지만, httpd.conf 파일에 직접 써 줄 수도 있는데, 아래와 같은 형식으로 사용됩니다. 특히 PHP3 모듈에 대해서 아래와 같은 설정을 해 주어야 합니다.

```
AddType application/x-httpd-php3 .php3
AddType application/x-httpd-php3-source .phps
```

PHP4(Zend) 에 관해서는 아래와 같은 옵션을 줍니다.

```
AddType application/x-httpd-php .php
AddType application/x-httpd-php-source .phps
```

tgz 파일에 관해서는 아래와 같이 써줍니다. 다른 형식들도 이와 유사하게 써 주면 됩니다. 예를 들면, 자신만의 독특한 확장자를 가진 파일을 작성했을 때, 이것에 관한 type을 mime.types 에 써 주든지 httpd.conf 에 써 주든지 해야합니다.

```
AddType application/x-tar .tgz
```

지정된 디렉토리 외에 위치한 cgi 스크립트를 실행하기 위해서 아래와 같은 옵션을 써줍니다.

```
AddHandler cgi-script .cgi
```

perl 모듈을 사용하려면 아래와 같은 옵션을 주어야 합니다.

```
Alias /perl/ /www/perl/
<Location /perl>
    SetHandler perl-script
    PerlHandler Apache::Registry
    Options +ExecCGI
</Location>
```

사용자가 잘못된 페이지나 에러메시지를 보게 될 때 안내 페이지를 어떤 것으로 할 지 지정하는 부분입니다. 아파치의 에러메시지에 관해서 아래에 적어 놓았습니다.

1) 보통의 텍스트

ErrorDocument 500 서버가 정상적으로 작동하지 않습니다.

2) 지역적인 방향 전환(missing.html이라는 파일이 404 에러가 발생할 때 보여지는 웹페이지가 됩니다.)

ErrorDocument 404 /missing.html

ErrorDocument 404 /cgi-bin/missing_handler.pl

3) 외부 방향 전환

ErrorDocument 402 http://some.other_server.com/subscription_info.html

HTTP 반응방식에 관한 옵션입니다. 첫 번째 줄은 netscape 2.0 대와 그와 유사한 기능을 하는 브라우저들에 대한 설정이고, 아래는 마이크로소프트 Internet Explore 4.0을 위한 설정입니다. HTTP/1.1 규약을 제대로 처리하지 못하는 것에 대한 설정입니다. 이 후 버전에서는 개선 되었습니다.

BrowserMatch "Mozilla/2" nokeepalive

BrowserMatch "MSIE 4\..0b2;" nokeepalive downgrade-1.0 force-response-1.0

HTTP/1.1 규약을 제대로 지원하지 못하는 부분들에 대해서 잘못된 동작을 하지 못하게 하는 설정입니다.

BrowserMatch "RealPlayer 4\..0" force-response-1.0

BrowserMatch "Java/1\..0" force-response-1.0

BrowserMatch "JDK/1\..0" force-response-1.0

프록시 모듈을 사용하려고 할 때 아래와 같이 써 주면 됩니다. 프록시는 경우에 따라서 매우 유용할때가 있습니다.

<IfModule mod_proxy.c>

ProxyRequests On

<Directory proxy:*>

Order deny,allow

Deny from all

Allow from .your_domain.com

</Directory>

프록시 서버를 운영할때는 서버에서 캐쉬 기능을 이용하는 것이 중요하다. 아래의 내용은 캐쉬 기능을 사용하기 위한 설정입니다. 만약에 CacheRoot가 없으면 서버에서 프록시를 위해서 캐쉬를 하지 않습니다.

CacheRoot "/www/proxy"

CacheSize 5

CacheGcInterval 4

CacheMaxExpire 24

CacheLastModifiedFactor 0.1

CacheDefaultExpire 1

NoCache a_domain.com another_domain.edu joes.garage_sale.com

</IfModule>

(4) 아파치 가상호스트 운영(VirtualHost)

아파치의 기능중에서 가장 돋보이는 기능이 여러개의 호스트를 하나의 서버로 운영할 수 있다는 것입니다. 여러개의 호스트를 하나의 서버상에서 운영하면 서버를 여러개 운영하지 않아도 되는 장점이 있고, 하나의 IP어드레스에 여러개의 도메인을 인식시켜서 IP 어드레스를 절약할 수 있다는 이점도 있습니다. 반면 서버에 문제가 생기면 동시에 많은 서버가 다운된다는 단점도 있습니다. 주로 웹호스팅 업체에서 이 방법을 많이 사용하고 있습니다.

① VirtualHost의 설정

아파치 서버의 설정파일 맨 아래에 써 줍니다. /www/conf/httpd.conf 파일의 맨 아래 부분에 가상호스트 설정관련 파일들이 주석 처리되어 있을 것입니다. 크게 2가지 방식으로 설정을 할 수 있습니다. IP기반의 설정과 도메인 기반의 설정이 그것입니다. IP기반의 가상호스트 서비스는 도메인 이름이 들어갈 자리에 IP어드레스를 적어주고 IP alias 처리를 해준다는 점이 도메인 기반의 서비스와 다른 점입니다.

NameVirtualHost 라는 항목을 추가 시켜 주어야 합니다. 이것은 가상서비스를 담당할 서버를 적어 주는 것입니다. 만약 적어 주지 않으면 서비스가 안될 수도 있으니 반드시 적어주어야 합니다. IP Alias로 여러개의 IP를 운영하고 있다면 여러개를 적어 줄 수도 있습니다.

NameVirtualHost 211.50.61.10:80

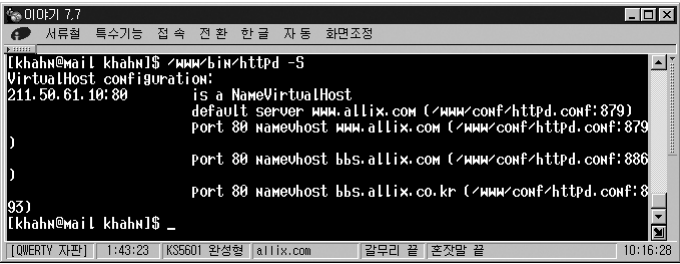
아래와 같은 양식으로 적어 줍니다.

```
<VirtualHost [도메인네임] 또는 [IP-어드레스]>
    ServerAdmin webmaster@[도메인네임]    <- 서버를 관리하는 관리자의 메일 주소
    ServerName [서버의 이름]              <- 서버의 이름
    DocumentRoot /www/htdocs/             <- 서버의 루트 디렉토리
    ErrorLog logs/[서버의 이름]-error_log   <- 에러메시지를 기록할 에러파일 이름
    ScriptAlias /cgi-bin/ /www/cgi-bin      <- cgi 스크립트가 가능하도록 하는 설정
</VirtualHost>
```

② name서버에 설정사항을 알려줌
이것이 중요한 부분인데, 반드시 네임서버에도 가상호스트의 이름을 알려 주어야 합니다. 만약 네임 서버를 자체적으로 운영하고 있지 않다면 네임서버 관리를 담당하는 서비스 업체에 자신이 설정한 가상호스트의 내용을 알려 주어야 합니다. 네임서버를 운영하고 있다면 아래와 같은 사항을 기록해 주어야 할 것입니다. 보통은 zone-[도메인네임] 형식의 파일에 이러한 정보를 담고 있습니다. DNS에 관한 문서를 참조해 보기 바랍니다.

```
@      IN      SOA      ns.wowlinux.com. hostmaster.wowlinux.com.
                                2000130000      ; Serial
                                3600              ; Refresh ( 6 hours )
                                1800              ; Retry ( 30 minutes)
                                1209600           ; Expire ( 14 days )
                                86400             ; Minimum ( 1 day )
      IN      NS       ns.wowlinux.com.
      IN      A        211.50.61.10
      IN      MX 10    mail
; Host addresses
ns      IN      A      211.50.61.10
wowlinux.com.  IN    A    211.50.61.10
; Aliases
mail     IN      CNAME  ns
www      IN      CNAME  ns
bbs      IN      CNAME  ns
[가상호스트서버의 도메인네임]  IN    CNAME  ns
```

③ 현재 운영중인 가상호스트의 리스트를 보여줍니다. httpd -S 옵션으로 확인 가능합니다.



(5) 아파치 인증모듈의 사용

아파치 서버에서는 웹 상에서 인증을 할 수 있도록 정보를 제공합니다. .htpasswd 라는 파일과 .htaccess 라는 파일을 가지고 설정을 하게 됩니다. 이 방법을 사용하기 전에 cgi를 사용가능하게 설정이 되어 있는지를 점검합니다.

- htaccess - 사용자의 ID를 등록해 두는 곳입니다.
- htpasswd - 사용자에게 해당하는 비밀번호를 등록하는 파일입니다.
- htaccess 파일에는 사용자인증을 실행할 디렉토리나 사용자를 적어줍니다. 여기서 .htpasswd 라는 파일은 관리자가 만들어 주어야 하는 파일입니다. 사용자에게 해당되는 패스워드가 htpasswd 라는 프로그램에 의해서 기록됩니다.

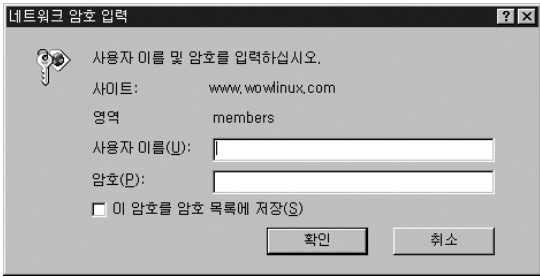
```
AuthName "User Auth"
AuthType Basic
AuthUserFile /home/[사용자ID]/.htpasswd
<Limit GET>
    require user [사용자ID]
</Limit>
Options Indexes FollowSymLinks Includes ExecCGI
AddType application/x-httpd-cgi .cgi

AddType text/x-server-parsed-html .html
```

이제는 등록된 ID에 관해서 패스워드를 입력해 줍니다.

```
htpasswd-c/home/[사용자ID]/ .htpasswd [사용자ID]
```

이제는 웹브라우저 상에서 확인하면 됩니다. 아래와 같은 대화 상자가 나타나면 성공입니다.



(6) 아파치 웹서버 에러메시지 일람

클라이언트 에러 메시지(웹브라우저상에서 확인되는 메시지)

400 Bad Request

서버에 보낸 요청 메시지 형식을 서버가 이해 할 수 없다는 응답메시지 입니다. HTML 문법을 확인 합니다.

401 Unauthorized

사용자 인증을 서버에서 이해하지 못했을 경우에 해당됩니다. 잘못된 사용자 인증요청이 있었을 경우 입니다.

403 Forbidden

서버가 인증을 거절했을 경우에 나타나는 메시지입니다. 이러한 메시지가 나오면 관리자에게 문의해서 인증이 되도록 권한을 변경해 달라고 합니다.

404 Not Found

클라이언트가 요청한 문서가 서버에 존재하지 않을 경우에 나타나는 메시지 입니다.

서버에서의 에러메시지 분석

500 Internal Server Error

사용자의 요구사항에 하자가 없지만, 서버에서 사용자의 요구를 처리할 준비가 되지 않았을 경우에 나타나는 옵션입니다.

501 Not Implemented

사용자 요구 가운데에서 일부의 명령을 제대로 수행할 수 없을 경우에 나타나는 메시지입니다. 일부는 실행이 될 수도 있습니다.

502 Bad Gateway

게이트웨이 경로를 잘못 지정해 주어서 나타나는 메시지입니다. 아파치서버의 프록시 관련 옵션을 수정해 줍니다.

503 Service Unavailable

서버에서 일시적으로 과부하가 걸렸을 경우에 나타나는 메시지입니다.

7. Mail Service(sendmail)



메일서버로 가장 많은 사람들이 사용하는 sendmail 서버에 관해서 알아보도록 하겠습니다. E-mail을 실제로 내부에서 메일을 전달해 주는 메일관리자(MDA:Mail Delivery Agent)로는 procmail, mail 등이 사용되고 있습니다. sendmail을 만든사람인 에릭앨먼은 procmail을 사용하도록 권장하고 있습니다. 또 외부로 나갈 필요가 있는 메일을 관리하는 메일전송관리자(MTA:Mail Transfer Agent)로는 sendmail, smail, Zmail, MMDF 등의 프로그램이 사용되고 있습니다.

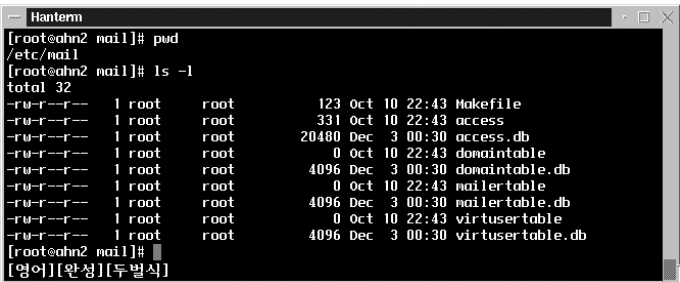
(1) sendmail 설치

① sendmail 이 설치되어 있는지 확인합니다. 또한 8.8.x 이하의 sendmail을 사용하고 있다면 삭제하고 새로운 버전을 설치할 것을 권장합니다. 스팸메일이나 보안상의 문제가 있습니다. 와우리눅스 7.1 에는 sendmail-8.11.3 버전이 포함 되어 있습니다.

```
[root@localhost /root]# rpm -qa | grep sendmail
sendmail-cf-8.11.3-1wl
sendmail-8.11.3-1wl
[root@localhost /root]#
```

② sendmail 에 관련된 설정파일 확인

sendmail을 설치하고 나면 아래 그림과 같이 /etc/mail 이라는 디렉토리에 여러 가지 설정파일들이 생기게 됩니다. 이것들을 수정해서 우리는 메일전송에 관련된 관리할 수 있습니다. 또, /etc/sendmail.cf 라는 파일이 매우 중요한 역할을 합니다.



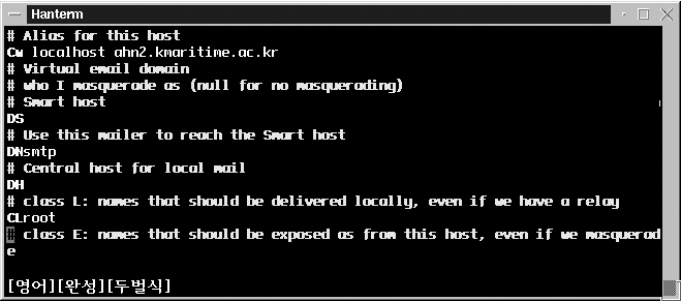
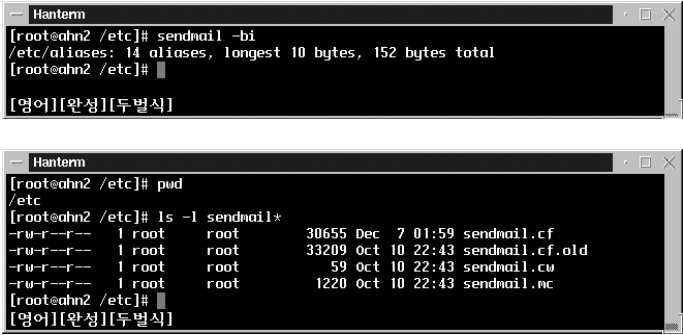
여기서 sendmail.cf 은 config file 이고, sendmail.cf.old 파일은 백업파일에 해당되며 sendmail.cw

파일은 메일을 처리해주는 Host의 이름을 적어주는 곳입니다. 이곳에 적어놓은 Host 명으로 메일이 전달되면 다른 곳으로 전달하지 않고 MDA로 메일을 보내줍니다. 보통 FQDN(Fully Qualified Domain Name)을 적고 만일 여러개의 호스트 이름을 사용한다면 모두 적어줘야 합니다.

sendmail.mc는 매크로를 정의해 놓은 파일들입니다. 이러한 파일들을 조절해서 우리는 메일을 보내고 받을 수 있습니다.

③ sendmail 설치 테스트

아래 그림과 같은 화면이 나오면 설치가 제대로 된 것입니다. 만약 제대로 나오지 않는다면 /etc/hosts 파일에 있는 도메인 네임이 잘 적혀 있는지 확인해야 합니다.

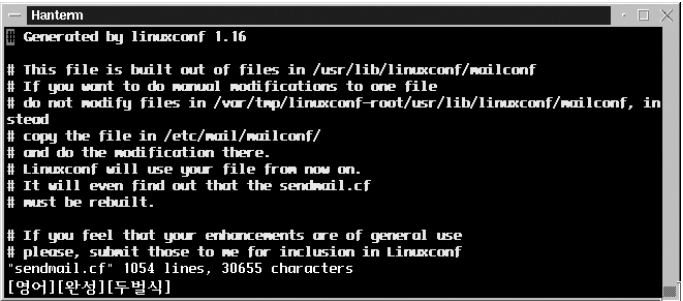


② 설정에 사용되는 명령어

- V sendmail.cf 에 사용된 파일의 버전
- M MDA(Mail Delivery Agent) 지정
- D 메일전송자 이름 지정
- R 다시쓰기 지정(Rewrite)
- S Rule-set 정의 시작하기
- C Class의 값을 특정값으로 정의한다
- F Class 파일로부터 읽어 들입니다.
- O Option 설정
- H Header의 구성방식 정의
- P 배달의 우선순위 설정
- T 승인된 사용자 지정하기(스팸방지)
- K Key설정관련 Database
- E 환경변수 지정(v8.7 이상에서)

③ 주석문 #

주석문은 # 문자로 시작되고 명령의 중간에 있어도 무관합니다. sendmail.cf는 약 1500 줄에 해당되는 내용을 가지고 있습니다.



(2) sendmail.cf 파일 관리하기

sendmail.cf 파일은 sendmail에서 가장 중요한 역할을 하는 설정파일입니다. 너무나 많은 내용들이 있어서 하나하나 살펴보려면 책이 한 권이 됩니다.

sendmail에서 사용되는 규칙

① 전반적인 규칙 알아보기

sendmail.cf 파일은 사용자가 보기에 편하게 만들어진 파일이 아니고 기계적으로 읽기 편하도록 만들어진 파일입니다. 아마도 이 파일을 만든 사람은 머리가 매우 좋든지 아니면 옆에 산더미처럼 메모를 해 놓고 작업했을 것이라 생각됩니다.

sendmail은 line 단위로 실행해 나가며, 설정에 관련된 명령어는 하나의 문자로 나타냅니다. 따라서 line의 처음에 공백이 있으면 안되고, 명령은 한줄에 하나씩 입력해야 합니다. 만약 line의 처음에 탭이나 공백문자가 있으면 바로 위의 줄에서 계속된 명령으로 인식하게 됩니다.

세부설정

아래의 내용은 필자의 sendmail.cf 설정입니다. 사용자 여러분에 맞게 고쳐서 사용하여야 합니다. 중요한 변경내용만을 표시해 보겠습니다. sendmail.cf 는 기본적인 설정과 각종 Rule로 설정되어 있습니다.

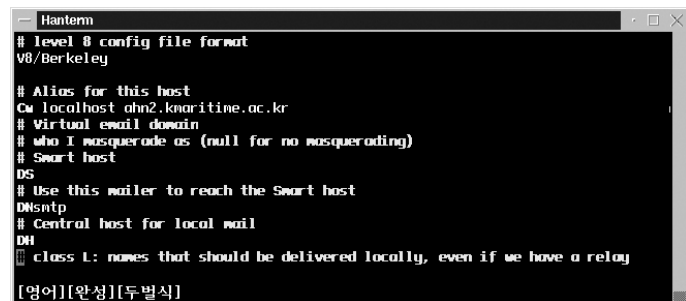
① 처음 부분

이곳에서는 sendmail.cf 파일을 설정하기 위해 사용한 툴과 권리에 대한 옵션들이 적혀 있습니다.

```
#
# Copyright (c) 1983 Eric P. Allman
# Copyright (c) 1988, 1993
# The Regents of the University of California. All rights reserved.
...중략
#
#####
#####
#####
#####
##### SENDMAIL CONFIGURATION FILE
#####
#####
#####
##### @(#)cfhead.m4 8.9 (Berkeley) 1/18/97 #####
##### @(#)cf.m4 8.24 (Berkeley) 8/16/95 #####
...중략
```

② 버전의 표시

sendmail.cf 의 내용들은 패치되거나 새로 판올림이 될 때 마다 여러 가지 기능이나, 파일의 설정이 변경되어 왔습니다. 따라서, 아래의 내용은 그러한 변형에 대한 버전을 표시해 줍니다. 아래의 내용은 Cw 라는 명령으로 localhost 가 ahn2.kmaritime.ac.kr 이라고 표시되어서 메일전달을 담당하게 되는 서버를 설정 하였습니다.

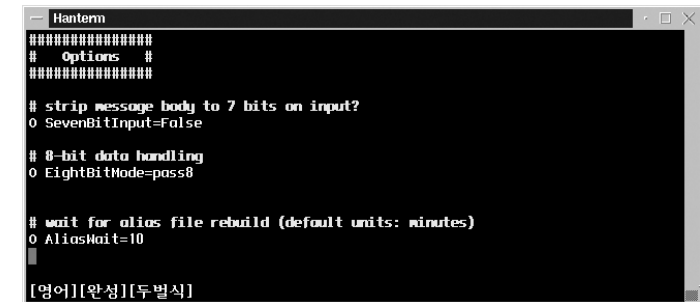


```
Hantenn
# level 8 config file format
V8/Berkeley

# Alias for this host
Cw localhost ahn2.kmaritime.ac.kr
# Virtual email domain
# who I masquerade as (null for no masquerading)
# Smart host
DS
# Use this mailer to reach the Smart host
Dnsnsp
# Central host for local mail
DH
class L: names that should be delivered locally, even if we have a relay
[영어][완성][두벌식]
```

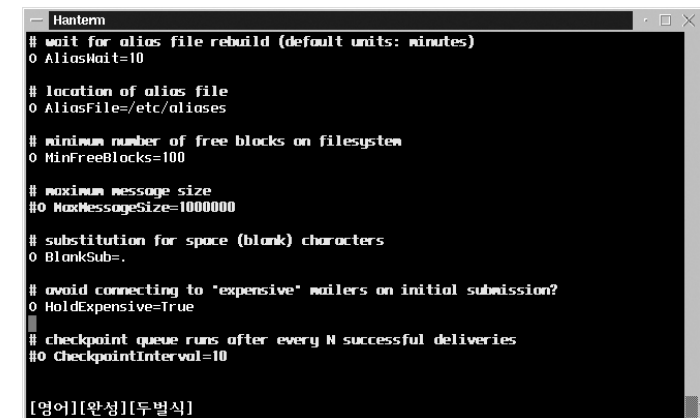
③ 옵션들의 설정

옵션들은 O 라는 명령으로 시작되고, O 와 명령사이에 공백이 한 칸 있습니다. 그리고, 중요한 설정은 한글 관련 부분입니다. 현재 배포되고 있는 배포판들에서 한글설정을 위해서 ServerBitInput=False 라고 되어있을 것입니다. 또, 한글이나 2바이트 문자권을 위해서 EightBitMode=pass8 이라고 지정되어 있는 것을 알 수 있습니다. /etc/aliases 라는 파일을 읽어서 sendmail을 실행하는데, 참고 하는데, 이 alias 파일을 읽기 위한 대기 시간을 표시해 줍니다.



```
Hantenn
#####
# Options
#####
# strip message body to 7 bits on input?
O SevenBitInput=False
# 8-bit data handling
O EightBitMode=pass8
# wait for alias file rebuild (default units: minutes)
O AliasWait=10
[영어][완성][두벌식]
```

아래 그림에서 /etc/aliases 라는 파일의 위치를 표시해 주는 옵션이 있으며, 전달된 메일을 관리하는데 사용할 최소 공간을 표시해 주고 있습니다.(MinFreeBlocks)



```
Hantenn
# wait for alias file rebuild (default units: minutes)
O AliasWait=10
# location of alias file
O AliasFile=/etc/aliases
# minimum number of free blocks on filesystem
O MinFreeBlocks=100
# maximum message size
#O MaxMessageSize=1000000
# substitution for space (blank) characters
O BlankSub=.
# avoid connecting to "expensive" mailers on initial submission?
O HoldExpensive=True
# checkpoint queue runs after every N successful deliveries
#O CheckpointInterval=10
[영어][완성][두벌식]
```

④ 메일 포워딩 옵션

서버로 메일이 전달되면 이것을 다른 서버로 곧바로 복사해서 보내 주도록 하는 옵션이 ForwardPath 옵션입니다. 보편적으로 자신의 홈디렉토리 아래에 .forward 라는 파일이 있고 그 안에 전달될 메일주소를 적어주게 됩니다.



```
Hantenn
# Forward file search path
O ForwardPath=$z/.forward.$u:$z/.forward
VISUAL
[영어][완성][두벌식]
```

⑤ 메일처리를 위한 임시 디렉토리(메일 큐 디렉토리)

메일이 전달되면 일단 임시로 머무는 장소가 있는데 그곳이 메일 큐 디렉토리입니다. 이곳에 전달된 메일들을 쌓아 놓았다가 서버가 한가한 시간에(거의 0.1초만에 전달됩니다.) 메일을 각자 사용자에게 보내 주는 역할을 하게 됩니다.

```
Hanterm
# queue directory
0 QueueDirectory=/var/spool/nqueue
- VISUAL -
[영어][완성][두벌식]
```

⑥ 메일전달 상태저장파일

메일이 입출력된 사실을 저장하는 정보 파일의 위치를 지정해 줍니다.

```
Hanterm
# status file
0 StatusFile=/var/log/sendmail.st
- VISUAL -
[영어][완성][두벌식]
```

⑦ 메일 헤더파일 설정

우리가 메일을 주고 받을 때 맨 앞쪽에 메일을 보낸 시스템의 정보나, 메일을 처리한 MTA등의 정보를 표시해 주는 곳을 볼 수 있습니다. 이러한 정보를 메일을 보낼 때 보내도록 해주는 옵션입니다.

```
Hanterm
#####
# Format of headers #
#####

H?P?Return-Path: <$g>
H?Received: $?sfrom $s $.$? ($?$s$|from $.$ )
    $$.by $j ($v/$z)?r with $r$. id $i$?u
    for $u; $j;
    $.$b
H?D?Resent-Date: $a
H?D?Date: $a
H?F?Resent-From: $?x$x <$g>$|$g$.
H?F?From: $?x$x <$g>$|$g$.
H?x?Full-Name: $x
# H?Posted-Date: $a
# H?Received-Date: $b
H?M?Resent-Message-Id: <$t.$ie$j>
H?M?Message-Id: <$t.$ie$j>

[영어][완성][두벌식]
```

⑧ 각종 Rule 들

이제 그 아래에는 여러 가지 운영에 관련된 규칙들을 정의해 놓은 영역이 나타납니다. 기본으로 설정된 상태로 사용해도 별로 지장이 없습니다.

```
Hanterm
#####
### Ruleset 98 -- local part of ruleset zero (can be null) ###
#####

$98

# addresses sent to foohost.REDIRECT will give a 551 error code
R$* < e $i .REDIRECT. > $: $1 < @ $2 . REDIRECT. > < &{opMode} >
R$* < e $i .REDIRECT. > <i> $: $1 < @ $2 . REDIRECT. >
R$* < e $i .REDIRECT. > < $- > $## error $e 5.1.1 $: "551 User has moved; please t
ry " <$!e$2>

Scheck_mail
# don't check these
R<$+e$-u> $e ok shortcut
# idea from Steven Schultz
<<> $: <$n @ $(dequote "" ${client_name} $) >

[영어][완성][두벌식]
```

(3) 여러개의 메일 계정 관리 하기

만약 root 사용자가 ahn 같은 일반계정도 가지고 있다고 생각합니다. 이 2개의 계정으로 오는 메일을 모두 ahn이라는 사용자가 받아 볼 수 있게 하기 위해서 /etc/aliases 파일을 수정해 줍니다.

① 파일위치 확인하기

```
Hanterm
[root@ahn2 /etc]# pwd
/etc
[root@ahn2 /etc]# ls -la aliases
-rw-r--r-- 1 root root 732 Oct 10 22:43 aliases
[root@ahn2 /etc]#
```

② 설정사항의 수정

vi /etc/aliases 명령으로 파일을 편집합니다. 여러 가지 변경사항이 있을 수 있겠지만, 아래 그림에서 보는 바와 같이 설정하면 root 로 오는 메일들은 모두 ahn@pivlab.kmaritime.ac.kr 에게로 전달하게 됩니다.

```
Hanterm
# Person who should get root's mail
#root:
root : ahn@pivlab.kmaritime.ac.kr
~

[영어][완성][두벌식]
```

③ newaliases 명령 실행

이 명령을 실행해야 비로소 /etc/aliases의 내용이 적용됩니다.

```
Hanterm
[root@ahn2 /etc]# newaliases
/etc/aliases: 14 aliases, longest 10 bytes, 152 bytes total
[root@ahn2 /etc]#
```



(4) 웹호스팅을 위한 가상메일 설정부분

웹호스팅을 하다보면 여러 도메인에 대한 메일처리를 담당해야 할 때가 있을 것입니다. 이러한 설정을 하는 방법을 알아보도록 하겠습니다.

wowlinux.com 이라는 도메인과 linuxuser.co.kr 이라는 도메인을 호스팅 해주고 있다고 가정을 합니다. 이때 wowlinux.com 의 관리자가 webadmin@wowlinux.com 이라는 메일주소를 달라고 하고, linuxuser.co.kr 의 사용자도 webadmin@linuxuser.co.kr의 메일주소를 달라고 합니다. 이때

- ① /etc/sendmail.mc 파일을 vi 에디터로 편집합니다.
sendmail.mc파일에 FEATURE(virtusertable, hash /etc/mail/virtusertable)을 추가하고 sendmail.cf를 다시 생성합니다.

```
Hanterm
define('confAUTO_REBUILD')
define('confFIO_CONNECT', 'In')
define('confFIRY_NULL_MX_LIST', true)
define('confDONT_PROBE_INTERFACES', true)
define('PROCMAIL_MAILER_PATH', '/usr/bin/procmail')
define('snrsh', '/usr/sbin/snrsh')
FEATURE('virtusertable', 'hash -o /etc/mail/virtusertable')
FEATURE(redirect)
FEATURE(always_add_domain)
FEATURE(use_cw_file)
FEATURE(local_procmail)
MAILER(procmail)
MAILER(sntp)
FEATURE('access_db')
-- VISUAL --
[영어][완성][두벌식]
```

sendmail.cf를 다시 생성하는 방법은 아래 그림과 같습니다. m4 라는 유틸리티를 사용합니다. 이 명령에서 cf.m4 파일은 반드시 포함되어져야 합니다.

```
Hanterm
[root@ahn2 /etc]# m4 [cf.m4 파일의 경로] sendmail.mc > sendmail.cf
[영어][완성][두벌식]
```

- ② web1 이라는 계정을 webadmin@wowlinux.com을 위해서 만들어 주고 web2 라는 계정을 webadmin@linuxuser.co.kr을 위해서 만들어 줍니다. adduser 명령을 사용했습니다.

```
Hanterm
[root@ahn2 ~]# adduser web1
[root@ahn2 ~]# adduser web2
[root@ahn2 ~]# ls /home/
ahn ftp samba tcp21 web1 web2
[root@ahn2 ~]#
[영어][완성][두벌식]
```

- ③ /etc/mail/virtusertable 파일에서 아래의 내용과 같이 수정합니다.

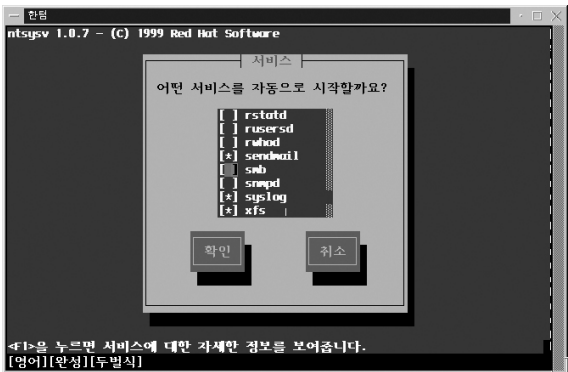
```
Hanterm
webadmin@tcp21.co.kr web1
webadmin@linuxuser.co.kr web2
-- INSERT --
[영어][완성][두벌식]
```

- ④ makemap으로 virtusertable.db 파일을 갱신해 줍니다.

```
Hanterm
[root@ahn2 mail]# makemap hash /etc/mail/virtusertable.db < /etc/mail/virtusertable.db
[영어][완성][두벌식]
```

sendmail 의 시작

- ① sendmail -bd -q1h
이 명령이 sendmail을 시작하는 가장 보편적인 명령입니다. -bd 라는 것은 sendmail 서비스를 background daemon 명령으로 실행하라는 것이고 - q1h 라는 명령은 1시간마다 메일큐 디렉토리를 점검하라는 옵션입니다. 만약 1시간 30분마다 점검하려면 - q1h30m 이라고 써주면 됩니다. 초단위 지정도 할 수 있습니다.(s 명령 사용)
- ② ntsysv에서 sendmail 항목을 체크하는 것으로 지정할 수도 있습니다.



mailq로 메일처리상황 확인

메일이 오고 가는 상황을 알 수 있습니다. 대기중인 메일이 없을 경우와 1개의 메시지가 있을 경우를 표시해 줍니다.

```
Hanterm
[root@pivlab ahn]# mailq
Mail queue is empty
[root@pivlab ahn]#
[영어][완성][두벌식]
```

```
Hanterm
[tcp21@un4 tcp21]$ mailq
Mail Queue (1 request)
--Q-ID-- --Size-- --Q-Time-- --Sender/Recipient--
LAA21594 1115 Thu Dec 9 11:59 webmaster@edulink.co.kr
(Deferred: connection refused by knail.com.)
finsider2@knail.com
[tcp21@un4 tcp21]$
[영어][완성][두벌식]
```


(6) POP3 서버의 설치

pop3 서버는 리눅스에서 다른 클라이언트 프로그램들에게 (Outlook Express, 유도라등.) 서버의 메일을 전달해 주가 위해서 필요한 서비스입니다. 이것은 imap 이라는 패키지가 설치되어 있으면 가능합니다. 하지만, 배포버전들에서 설정이 되지 않을 경우가 많이 있습니다. 리눅스에서 사용할 수 있는 POP3 서버로는 imap 프로그램 이외에도 qpopper 라는 프로그램이 있습니다.

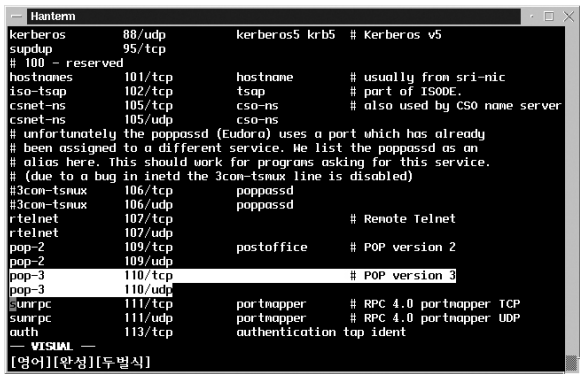
와우리눅스 7.1에는 imap-2000-9.rpm이 설치되어 있습니다.

① imap 설치

```
#rpm -Uvh imap-2000-9.rpm
```

② /etc/services 파일 점검하기

/etc/services 파일에서 pop3 에 해당되는 내용들을 수정합니다.110번 포트로 pop3 서비스가 실행되고 있음을 알 수 있습니다.



③ /etc/xinded.d/pop3 파일 수정하기

와우리눅스7.0 이하 버전에서는 inetd.conf 파일을 수정하였지만 그 이후 버전 부터는 xinetd 파일을 수정하여야 합니다. ‘disable=no’ 옵션으로 만들어 줍니다.(xinetd)

③ 아래 그림은 정상적인 작동상태를 확인하기 위해서 telnet 으로 110 포트를 점검한 것입니다. OK 라는 명령을 볼 수 있다면 성공적으로 설치가 된 것입니다.

