

내손으로 최신 커널 컴파일하여 업그레이드하기

초보 리눅서도 이번 호를 보고 그대로 따라하기만 하면 그 동안 어렵게만 생각해 오던 커널 컴파일 및 업그레이드를 내 손으로 직접 할 수 있다. 물론 커널 설정은 각 시스템의 하드웨어에 따라 다소 달라지지만 가장 일반적인 시스템을 기준으로 하였으므로 대부분 실제로 적용하는데 무리가 없을 것이다. 그 중 시스템 환경에 따라 특별히 신경 써야 할 부분은 별도로 설명을 하였으니 참고하기 바란다.

글 | 홍석범 오늘과 내일 넷센터(antihong@tt.co.kr)

커널(Kernel)에 대한 개념 및 2.4 버전 커널에 대한 설명은 자주 다루어 졌으므로 이번 호에서는 관련 설명은 제외하고, 실제로 리눅스를 이용하여 서버로 운영하고자 할 때 커널을 컴파일하여 업그레이드하는 방법에 대해서만 알아보도록 하겠다. 그럼, 실제로 커널 컴파일에 들어가기 전에 왜 커널을 컴파일하고 또 업그레이드해야 하는지 그 필요성에 대해 알아보자. 커널을 컴파일 및 업그레이드해야 하는 이유는 여러 가지로 설명 가능하지만 다음과 같이 크게 '보안'과 '성능향상'이라는 두 가지 이유로 설명할 수 있다.

(1) 시스템 보안을 위해

아래는 실제로 필자가 운영하는 리눅스 서버에서 테스트한 결과이다.

```
[user1 @www user1]$ id
uid=500(user1) gid=500(user1) groups=500(user1),10(wheel)
[user1 @www user1]$ gcc exploit.c -o exploit
[user1 @www user1]$ ./exploit
bug exploited successfully.
enjoy!
bash# id
uid=0(root) gid=0(root) groups=500(user1),10(wheel)
bash# uname -a
Linux kernel.tt.co.kr 2.2.18 #1 SMP Tue Nov 21 01:22:41 KST 2000 i686 unknown
bash# exit
```

이 버그는 리눅스 배포판과 관계없이 커널 버전 2.2.19 이전의 모든 시스템에 통용되는 버그로서 통상적으로 가장 많이 사용하는 레드햇 6.x 계열을 설치 후 커널 업그레이드를 하지 않았다면 초기 배포판 버전대로 커널이 2.2.16 이므로 위 버그가 바로 적용이 되어 일반 유저 권한만 있으면 소스를 컴파일 후 실행하면 바로 root 권한을 획득할 수 있게 되는 치명적인 문제가 있다. 위와 같이 리눅스 커널에서 치명적인 보안 문제가 발생하는 일은 그리 흔한 일은 아니지만 이러한 경우가 가끔 발생하므로 시스템의 보안을 위해 지속적으로 커널 패치 또는 업그레이드를 하여야 한다.

(2) 확장된 기능과 최적화된 성능을 위해

리눅스를 서버로 사용한다면 서버의 성능을 높이기 위해 CPU를 Dual 로 한다거나 메모리를 추가로 장착하기도 한다. 그러나 실제 물리적인 성능을 높이기 보다는 시스템의 핵심인 커널과 각종 데몬의 설정을 최적화함으로써 더욱 가시적인 향상을 기대할 수 있다. 일반적인 리눅스 배포판의 커널은 배포판을 이용하는 유저들이 어떤 하드웨어를 사용하는지 그리고 어떤 목적으로 사용할 것인지 알 수 없으므로 여러 다른 많은 종류의 하드웨어와 상당수의 설정들을 지원 하도록 설정되어 배포되므로 커널 이미지의 크기가 당연히 커질 수밖에 없고, 사용하지 않는 여타 기능들이 많이 추가 되었으므로 시스템에 최적화되지 못해

당연히 성능이 떨어지게 된다. 혹 이전에 커널 컴파일을 한 적이 없다면 /usr/src/linux 로 이동한 후 make menuconfig를 한 후에 Network device support ---> 선택후 Ethernet (10 or 100Mbit) ---> 를 선택해 보기 바란다. (CD 로 설치시 Kernel Source 를 선택하지 않았다면 linux 라는 디렉토리가 없을 것이다.) 실제 사용하지도 않는 모든 이더넷 카드가 모듈로 선택되어 있고 이것들이 현재 시스템의 커널에 올라가 있다. 이렇듯 불필요한 기능을 커널에 포함시키거나 모듈로 설정하였을 경우 시스템이 전반적으로 무거워지고 많은 메모리를 소모하게 된다.

또한 정작 나에게 필요한 기능은 포함되지 않는 경우가 있어 특수한 기능을 이용하거나 나에게 필요한 기능만을 선택하여 사용할 수 있도록 하기 위해 커널 컴파일이 필요하다. 이를테면 리눅스를 이용해 라우터로 이용하거나 클러스터링을 구성 하려면 반드시 커널 패치 및 컴파일이 필요하다.

이제, 커널 업그레이드 및 컴파일의 필요성을 느꼈다면 본격적으로 커널 업그레이드 및 컴파일을 시작해 보도록 하자. 사실 커널 컴파일을 통한 업그레이드를 하려면 시스템에 대한 많은 배경 지식도 필요하고 또한 복잡한 과정이 필요하기는 하지만 이번 호에는 그런 설명은 빼고 설명을 보고 그대로 따라 해서 일단 커널 업그레이드 및 컴파일에 성공하는 것만을 목적으로 하기로 하자. 실제로 한번 성공하고 나면 이후부터는 커널에 친숙해 지고 어렵지 않게 재컴파일을 할 수 있을 것이다.

1. 커널 컴파일 과정을 보다 쉽게 이해를 하기 위해 먼저 /etc/bashrc 를 열어 PS1="[Wu@Wh WW]WW\$"로 되어 있는 부분에서 대문자 W를 PS1="[Wu@Wh Ww]WW\$"와 같이 소문자로 변경한 후 다시 로그인 해보자. 이제부터는 이동하는 전체 경로가 모두 보이게 될 것이다. 작업을 위해 먼저 /usr/src 디렉토리로 이동하자.

```
[root@work /root]# cd /usr/src
[root@work /usr/src]#
```

2. 먼저 커널의 공식 사이트인 ftp.kernel.org 또는 한국 미러 사이트인 ftp.kr.kernel.org 에 ftp로 접속 후 /pub/linux/kernel/v2.4 디렉토리에서 8월 중순 현재 최신 리눅스 커널인 linux-2.4.8.tar.gz 파일을 다운로드 한다. (이 글을 마감할 때 썬이면 2.4.8 보다 업그레이드된 상위 버전이 발표되어 있을 것이다. 그러나 소스만 바뀔 뿐 메뉴의 내용과 컴파일 방식은 동일하므로 어떤 버전이라도 관계없으며 이 책에서는 커널 2.4.8 을 기준으로 설명한다.)

이때 /pub/linux/utils/kernel/modutils/v2.4 디렉토리에 있는 최신 버전의 modutils-2.4.6.tar.gz라는 프로그램도 같이 다운로드 한다. Modutils 버전은 Kernel 버전과는 관계 없으므로 kernel 이 2.4.8 이라고 해서 modutils 도 2.4.8 를 받을 필요는 없으며 단지 최신 버전을 다운로드하면 된다.

3. 만약 현재의 시스템에 linux 라는 링크가 있으면 rm -f linux로 링크를 삭제한다. 그리고 다운로드 한 파일을 /usr/src 디렉토리에서 압축을 해제한다. 만약 bz2 형식일 경우에는 bzip2 -d 로 압축을 해제한 후 tar 로 압축을 해제하면 된다.

```
[root@work /usr/src]# rm -f linux
[root@work /usr/src]# tar zxvfp linux-2.4.8.tar.gz
```

4. 압축을 해제하면 /usr/src/linux와 같이 압축이 풀리는데, 압축이 모두 풀리고 난 후에는 /usr/src 디렉토리에서 linux 디렉토리를 linux2.4.8 으로 rename 하고 난 후 ln -s linux linux-2.4.8과 같이 linux라는 이름으로 링크한다.

```
[root@work /usr/src]# mv linux linux-2.4.8
[root@work /usr/src]# ln -s linux-2.4.8 linux
```

링크를 한 후 ls -al로 확인해 보면 아래와 같이 보일 것이다.

```
lrwxrwxrwx    1 root    root          11  8월 1 17:55 linux -> linux-2.4.8
drwxr-xr-x   18 root    root        4096  8월 1 19:45 linux-2.4.8
```

5. /usr/include 디렉토리로 이동한 후 다음과 같이 기존의 디렉토리를 일부 삭제하고 커널 소스를 설치한 경로로 다시 링크를 한다.

```
[root@work /usr/src]# cd /usr/include/
[root@work /usr/include]# rm -rf asm linux scsi
[root@work /usr/include]# ln -s /usr/src/linux/include/asm-i386    asm
[root@work /usr/include]# ln -s /usr/src/linux/include/linux    linux
[root@work /usr/include]# ln -s /usr/src/linux/include/scsi     scsi
```

위와 같이 /usr/include 에서 압축을 해제한 커널 소스에 있는 디렉토리로 링크를 하는 이유는 소스내에 #include 를 하는 파일들은 기본적으로 /usr/include 디렉토리에서 파일을 찾기 때문이다. 예를 들어 커널 소스내에 #include <linux/config.h>와 같이 파일을 include 하고 있다면 /usr/include/linux 디렉토리에서 config.h라는 헤더 파일을 찾게 된다. 실제로 커널 소스의 헤더 파일들은 커널 소스의 include 디렉토리 이하에 있다.

6. 함께 다운로드한 modutils-2.4.6 도 컴파일하여 설치한다.
이 유틸리티는 모듈의 추가 및 삭제등 모듈 관련 작업을 수행하기 위한 패키지로 커널 옵션 선택중 모듈을 선택하지 않는다면 필요가 없지만 대부분 모듈을 사용하므로 필요하다.

```
[root@work /usr/include]# cd /usr/src/
[root@work /usr/src]# rpm -e --nodeps modutils
[root@work /usr/src]# tar zxvfp modutils-2.4.6.tar.gz
[root@work /usr/src]# cd modutils-2.4.6
[root@work /usr/src/modutils-2.4.6]# ./configure
[root@work /usr/src/modutils-2.4.6]# make
[root@work /usr/src/modutils-2.4.6]# make install
```

modutils 패키지는 이전 커널 버전의 경우 /etc/conf.modules 파일에 설정되어 있는 내용을 읽어 들여 부팅때마다 자동으로 필요한 모듈을 로드(load)하였으나, 커널 2.4 버전부터는 파일 이름이 기존의 conf.modules 에서 modules.conf로 변경되었으므로 아래와 같이 파일 이름을 변경한다.

```
[root@work /usr/src/modutils-2.4.6]# mv /etc/conf.modules /etc/modules.conf
```

그리고 부팅시 이 파일을 읽어들이도록 설정되어 있는 rc.sysinit 파일도 함께 변경한다.
[root@work /usr/src/modutils-2.4.6]# vi /etc/rc.d/rc.sysinit로 파일을 읽어
:%s/conf.modules/modules.conf/g 로 conf.modules로 설정된 부분을 모두
modules.conf로 수정한다.

7. 커널 2.4 에서는 공유 메모리를 사용하기 위해 아래와 같이 /var/shm 디렉토리를 생성하고 부팅시 자동으로 마운트 될 수 있도록 fstab에 추가해 준다.

```
[root@work /usr/src/modutils-2.4.6]# mkdir /var/shm
```

```
[root@work /usr/src/modutils-2.4.6]# vi /etc/fstab로 파일을 열어 제일 하단에
```

```
tmpfs /var/shm tmpfs defaults          0 0
```

라인을 추가한다.

이제 본격적으로 커널 컴파일을 하기 전에 선택할 옵션을 설정하도록 하자.

8. /usr/src/linux 로 이동한 후 커널 옵션을 설정하기 위해 make menuconfig 를 한다.

```
[root@work /usr/src/modutils-2.4.6]# cd /usr/src/linux
```

```
[root@work /usr/src/linux]# make menuconfig
```

이제 아래와 같은 화면이 뜨게 될 것이다.

그림 1.

Spacebar, Tab 키, Enter 키 및 상,하,좌,우 화살표키를 적절히 이용하여 아래와 같이 설정을 해 보자. Kernel 2.4.8 의 경우 30개의 메인 메뉴가 있는데, 메인 메뉴에서 Spacebar 나 Enter 키를 이용하면 하위 메뉴로 이동할 수 있으며 하위 메뉴 선택 항목에서는 Spacebar를 한 번씩 선택할 때마다 선택된 항목을 해제하거나 선택하며, 모듈이 존재할 경우에는 모듈로 선택할 수 있도록 해 준다. [*] 로 설정되면 해당 메뉴가 선택되어 선택된 기능을 커널에 포함시키는 것이며(이를 '커널에 정적으로 또는 Static 하게 포함시킨다' 라고 표현한다.) [] 와 같이 공란일 경우에는 메뉴를 선택하지 않아 해당 기능을 커널에서 제외하는 것이며 <M>은 모듈로 설정하여 커널에 모듈로 설정하는 것이다.

그리고 하위 메뉴로 이동하여 메뉴를 모두 설정한 다음에는 Tab 이나 화살표키를 이용해 Exit를 선택하면 상위 메뉴로 이동하고, Help를 선택하면 해당 메뉴에 대한 설명을 볼 수 있다.

일반적으로 아래에서 지정한 설정을 따라하기만 하면 무난히 커널 컴파일이 되나 자신의 시스템 환경에 맞추어야 하는 부분은 별도로 설명을 하였으니 자신의 시스템에 맞도록 설정을 하기 바란다.

그리고 각 옵션에서 1번부터 30 번까지의 일련 번호는 실제로 메뉴에는 없으나 편의를 위해 임의로 설정한 것임을 참고하기 바란다.

```
#####
```

```
1. Code maturity level options --->
```

```
#####
```

```
 [*] Prompt for development and/or incomplete code/drivers
  # 커널 코드중 개발중인 부분을 메뉴에 나타낼 것인지 결정한다.
  # 단지 메뉴만 추가되므로 선택하는 것이 좋다.
```

```
#####
```

```
2. Loadable module support --->
```

```
#####
```

```
 [*] Enable loadable module support
 [ ] Set version information on all module symbols
 [*] Kernel module loader
  # 모듈 기능을 이용하려면 위와 같이 선택하는 것이 좋다.
```

```
#####
```

```
3. Processor type and features --->
```

```
#####
```

```
# CPU에 대한 전반적인 설정을 하는 부분이다.
```

```

(Pentium-III/Celeron(Coppermine)) Processor family
# 시스템에 맞는 CPU를 선택해야 한다. 만약 이 메뉴에서 386을 사용하면 모든 CPU
에서
# 사용 가능하나 Pentium-III 를 선택하면 펜티엄이나 486에서는 사용할 수 없으니
# 주의하기 바란다. 즉, 실제 하드웨어 사양보다 하위 사양을 선택해도 되나 상위 사양을
# 선택하면 안 된다는 것이다. 그러나 시스템 최적화를 위해 자신의 실제 사양에 맞는
# 사양을 선택하는 것이 좋다. 자신의 시스템에서 어떤 CPU 를 사용하는지 잘 모르겠으면
# cat /proc/cpuinfo 로 확인하면 된다.
# "Pentium-Classic" 인텔 펜티엄의 경우 선택
# "Pentium-MMX" 인텔 펜티엄 MMX의 경우 선택
# "Pentium-Pro" 인텔 펜티엄 프로/셀러론/펜티엄II의 경우 선택
# "Pentium-III" 인텔 펜티엄 III 의 경우 선택.

```

```

< > Toshiba Laptop support
< > /dev/cpu/microcode - Intel IA32 CPU microcode support
< > /dev/cpu/*/msr - Model-specific register support
< > /dev/cpu/*/cpuid - CPU information support
(off) High Memory Support
    # 시스템의 RAM 이 1GB 미만이라면 off를 선택하고 1GB 에서 4GB 사이라면 4GB
    를
    # 선택한다. 만약 4GB 이상일 경우에는 64GB를 선택하고 만약 시스템에서 물리적인
    # 메모리가 모두 인식되지 않으면 /etc/lilo.conf에 "mem=256M" 와 같이 추가하면
    # 된다.
[ ] Math emulation
[ ] MTRR (Memory Type Range Register) support
    # AGP 그래픽 카드를 사용하여 이미지 쓰기시 속도를 증가시켜준다.
[ ] Symmetric multi-processing support
    # CPU가 1개라면 선택하지 않고 2개 이상이라면 SMP를 선택하고 아울러 Character
    # devices --> "Enhanced Real Time Clock Support" 옵션에도 Y를 선택해
    # 주어야 한다.
[ ] APIC and IO-APIC support on uniprocessors

```

```
#####
```

```
4. General setup --->
```

```
#####
```

```

# 네트워크 및 전원 관리등 시스템 전반적인 설정을 위한 메뉴이다.
[*] Networking support
# 설사 인터넷에 연결되지 않더라도 반드시 선택한다.
[ ] SGI Visual Workstation support
[*] PCI support
# 최근에는 거의 PCI 를 사용하므로 선택한다.
(Any) PCI access mode
    # 리눅스에서는 커널이 직접 하드웨어에 접근할 수 있는데, Any는 커널이 직접
    접근해
    # 보고 동작하지 않으면 BIOS로 돌아간다. Any 가 기본값이며 권장 사항이다.
[*] PCI device name database
    #/proc/pci나 /proc/ioprots 에 pci 에 대한 정보를 저장하게 한다. 선택한다.

```

```

[ ] EISA support
[ ] MCA support
[ ] Support for hot-pluggable devices
PCMCIA/CardBus support --->
# 노트북등에서 PCMCIA 를 사용시 선택한다.
[*] System V IPC
# 동기화를 위해 이 기능을 사용하므로 공유 메모리 사용등을 위해 반드시 설정한다.
[*] BSD Process Accounting
# 프로세스가 끝날 때 커널에 의해 프로세스 정보가 파일에 추가된다.
[*] Sysctl support
# 커널을 다시 컴파일 하거나 재부팅 할 필요 없이 커널 파라미터나 변수를
# 바꿀 수 있도록 해 주며 반드시 선택한다.
(ELF) Kernel core (/proc/kcore) format
<*> Kernel support for a.out binaries
<*> Kernel support for ELF binaries
<*> Kernel support for MISC binaries
[ ] Power Management support

```

```

#####
5. Memory Technology Devices (MTD) ---> N
#####
# 임베디드 시스템에서 사용된다. N 을 선택한다.

```

```

#####
6. Parallel port support ---> N
#####
# 프린터 및 플로터 등 병렬포트와 관련된 부분이다. N 을 선택한다.

```

```

#####
7. Plug and Play configuration --->
#####
# pnp 를 지원할 지 여부에 대한 설정이다.
<*> Plug and Play support
< > ISA Plug and Play support

```

```

#####
8. Block devices --->
#####
<*> Normal PC floppy disk support
# 일반적인 PC 내 Floppy에 대한 지원 여부를 선택한다.
< > XT hard disk support
< > Compaq SMART2 support
< > Compaq Smart Array 5xxx support
< > Mylex DAC960/DAC1100 PCI RAID Controller support
<M> Loopback device support
# 파일을 블록 디바이스처럼 사용 가능한 옵션으로 유용한 기능이므로 모듈로 설정한다.
# loop 디바이스를 이용하면 iso이미지등을 쉽게 마운트 할 수 있다.
# 네트워크에서 흔히 사용되는 loopback (127.0.0.1)과는 다른 것이므로

```

```

# 주의하기 바란다.
< > Network block device support
< > RAM disk support
# 메모리를 디스크처럼 사용하는 방법으로 대부분 이 기능은 필요 없다.
#####
9. Multi-device support (RAID and LVM) ---> N
#####
# RAID 나 LVM 등 다중 저장 장치들을 지원한다

#####
10. Networking options --->
#####
# 매우 중요한 네트워크 설정 부분이다.
<*> Packet socket
# tcpdump 등 네트워크 장치와 직접 통신하기를 원할 때 선택한다.
[ ] Packet socket: mmaped IO
[ ] Kernel/User netlink socket
[*] Network packet filtering (replaces ipchains)
# ipchains 를 대체하는 iptables 등 패킷 필터링을 지원하는지 여부를 설정한다.
# 리눅스를 라우터나 게이트웨이 또는 방화벽등의 용도로 사용하려면 반드시
# 설정하여야 하며 보안을 위해 설정하는 것을 권장한다. 이 옵션을 선택하면 하단의
# IP: Netfilter Configuration 설정이 추가된다.
[ ] Network packet filtering debugging
[ ] Socket Filtering
<*> Unix domain sockets
# 유닉스 도메인 소켓 지원에 필요하다.
# 네트워크에 연결되지 않아도 필요하므로 반드시 선택한다.
[*] TCP/IP networking
# 네트워크를 사용하지 않아도 반드시 선택하여야 한다.
[ ] IP: multicasting
# Mbone 방식의 인터넷 방송등에 사용된다.
[ ] IP: advanced router
# 리눅스를 라우터의 용도로 사용할 때 필요한 메뉴이다.
# 특별한 목적이 아니면 선택하지 않는다.
[ ] IP: kernel level autoconfiguration
< > IP: tunneling
< > IP: GRE tunnels over IP
[ ] IP: multicast routing
# Mbone 라우터에 사용된다.
[ ] IP: TCP Explicit Congestion Notification support
[* ] IP: TCP syncookie support (disabled per default)
# 서비스 거부 공격의 일종인 TCP SYN Flooding 공격을 차단하기 위해 설정한다.
# 이 기능을 설정한 후에라도 sysctl -w net.ipv4.tcp_syncookies=1 를 실행해
주어야
# syncookies 기능 사용이 가능하다. 기본적으로는 선택되어 있지 않지만 서비스 거부
# 공격을 차단하기 위해서라도 반드시 설정할 것을 권장한다.
IP: Netfilter Configuration --->
# 패킷 필터링에 대한 세부 설정 항목이다.

```

```

<*> Connection tracking (required for masq/NAT)
# NAT방식이나 IP 마스크레이딩등에 필요한 설정이다. 선택한다.
<*> FTP protocol support
# ftp 연결을 추적한다. 선택한다.
  <*> IP tables support (required for filtering/masq/NAT)
# 기존 2.2버전대의 ipchains 를 대체하는 iptables 를 지원할 것인지를
# 묻는다.NAT, IP 마스크레이딩, 방화벽 설정등에 필요하므로 선택한다.
< > limit match support
# iptables 에서 매칭되는 규칙이 있을 때 속도등을 제어할 수 있다.
# iptables 를 많이 사용할 경우에만 선택한다.
< > MAC address match support
# MAC 주소에 기반한 필터링을 할 때 필요하다.
< > netfilter MARK match support
< > Multiple port match support
# 규칙 적용시 여러 포트를 명시할 것인지 여부를 설정한다.
< > TOS match support
< > tcpmss match support
< > Connection state match support
< > Unclean match support (EXPERIMENTAL)
< > Owner match support (EXPERIMENTAL)
# 규칙 적용시 패킷을 생성한 사용자, 소유자등에 기반하여 필터링을 할 때
필요하다.
<*> Packet filtering
< > REJECT target support
< > MIRROR target support (EXPERIMENTAL)
<*> Full NAT
# 마스크레이딩, 포트 포워딩등을 사용할 때 필요하다.선택한다.
< > MASQUERADE target support
# NAT 의 특별한 케이스로 유동 IP를 사용하는 다이얼업 사용자에게
유용하다.
< > REDIRECT target support
# 투명한 프록시에 사용된다.
< > Packet mangling
< > TOS target support

< > MARK target support
< > LOG target support
# 규칙에 일치하는 패킷에 대해 로그를 남길 것인지를 설정한다.
< > TCPMSS target support
< > The IPv6 protocol (EXPERIMENTAL)
< > Kernel httpd acceleration (EXPERIMENTAL)
# 커널에 내장된 웹데몬인 KHTTPD 를 사용하고자 할 때 선택한다.
# 일반적으로 HTML 이나 이미지등 정적인 파일들을 담당하며 CGI 등은 처리하지
않는다.
# 아파치 웹서버에 비해 4-6배정도 속도가 빠른 것으로 알려져 있다.
[ ] Asynchronous Transfer Mode (ATM) (EXPERIMENTAL)
# ATM 을 사용하고자 할 때 필요하다.

```

```

< > The IPX protocol
# 노벨 네트워크와 연결시 필요하다.
< > Appletalk protocol support
# 애플의 맥킨토시와 연결할 때 필요하다.
< > DECnet Support
< > 802.1d Ethernet Bridging
< > CCITT X.25 Packet Layer (EXPERIMENTAL)
< > LAPB Data Link Driver (EXPERIMENTAL)
[ ] 802.2 LLC (EXPERIMENTAL)
[ ] Frame Diverter (EXPERIMENTAL)
< > Acorn Econet/AUN protocols (EXPERIMENTAL)
< > WAN router
[ ] Fast switching (read help!)
# LAN 구간에서 인터페이스끼리 직접 연결하여 데이터를 전송할 때 필요하며
# Tulip 등 일부 인터페이스 카드에서만 지원된다.
# Network packet filtering 을 선택하였으면 반드시 선택하지 않아야 한다.
# 주로 사용되지 않는 기능이므로 선택하지 않는다.
[ ] Forwarding between high speed interfaces
    QoS and/or fair queueing --->

```

```
#####
```

```
11. Telephony Support ---> N
```

```
#####
```

```
# 음성전송을 지원하는 Quicknet 의 telephony 카드를 사용하지 않는다면 선택하지
# 않는다.
```

```
#####
```

```
12. ATA/IDE/MFM/RLL support --->
```

```
#####
```

```
# SCSI 만 사용하고 IDE 를 사용하지 않으면 N 을 선택한다.
# SCSI 일 경우에는 df 시 파티션 정보가 sdx 와 같이 나타나고 IDE 일 경우에는
# hdx와 같이 나타난다. 중요한 설정이므로 주의있게 설정하기 바란다.
```

```
<*> ATA/IDE/MFM/RLL support
```

```
IDE, ATA and ATAPI Block devices --->
```

```
<*> Enhanced IDE/MFM/RLL disk/cdrom/tape/floppy support (NEW)
```

```
--- Please see Documentation/ide.txt for help/info on IDE drives
```

```
# 일반 PC에서 사용하는 E-IDE 장치 설정으로 IDE면 반드시 설정한다.
```

```
[ ] Use old disk-only driver on primary interface (NEW)
```

```
# 구형의 XT 하드 사용시 선택한다.
```

```
<*> Include IDE/ATA-2 DISK support (NEW)
```

```
[*] Use multi-mode by default (NEW)
```

```
# hda: set_multmode: status=0x51 { DriveReady SeekComplete Error }
```

```
# hda: set_multmode: error=0x04 { DriveStatusError } 와 같은 에러 메시지가
```

```
# 나오면 선택한다.
```

```
< > PCMCIA IDE support (NEW)
```

```
<*> Include IDE/ATAPI CDROM support (NEW)
```

```
# IDE 에 연결하는 CD 드라이버를 사용하면 선택한다.
```

```
< > Include IDE/ATAPI TAPE support (NEW)
```

```
< > Include IDE/ATAPI FLOPPY support (NEW)
```

```

< > SCSI emulation support (NEW)

--- IDE chipset support/bugfixes

[*] CMD640 chipset bugfix/support (NEW)
# CMD 640 칩을 사용하면 선택한다. 잘 모르겠으면 선택한다.
[ ] CMD640 enhanced support (NEW)
[*] RZ1000 chipset bugfix/support (NEW)

# RZ1000 을 지원하는지 여부를 묻는다. 잘 모르겠으면 선택한다.
[*] Generic PCI IDE chipset support (NEW)

[*] Sharing PCI IDE interrupts support (NEW)
[*] Generic PCI bus-master DMA support (NEW)
# 대부분의 펜티엄 보드가 해당되므로 선택한다.
[ ] Boot off-board chipsets first support (NEW)
[*] Use PCI DMA by default when available (NEW)
[ ] ATA Work(s) In Progress (EXPERIMENTAL) (NEW)
[ ] AEC62XX chipset support (NEW)
[ ] ALI M15x3 chipset support (NEW)
[ ] AMD Viper support (NEW)
[ ] CMD64X chipset support (NEW)
[ ] CY82C693 chipset support (NEW)
[ ] Cyrix CS5530 MediaGX chipset support (NEW)
[ ] HPT34X chipset support (NEW)
[ ] HPT366 chipset support (NEW)
[ ] Intel PIIXn chipsets support (NEW)
[ ] PIIXn Tuning support (NEW)
[ ] NS87415 chipset support (EXPERIMENTAL) (NEW)
[ ] OPTi 82C621 chipset enhanced support (EXPERIMENTAL) (NEW)
[ ] PROMISE PDC20246/PDC20262/PDC20267 support (NEW)
[ ] ServerWorks OSB4 chipset support (NEW)
[ ] SiS5513 chipset support (NEW)
[ ] SLC90E66 chipset support (NEW)
[ ] Tekram TRM290 chipset support (EXPERIMENTAL) (NEW)
[ ] VIA82CXXX chipset support (NEW)
[ ] Other IDE chipset support (NEW)

[ ] IGNORE word93 Validation BITS (NEW)

```

```
#####
```

```
13. SCSI support --->
```

```
#####
```

```
SCSI 컨트롤러를 사용하지 않고 IDE 만 사용한다면 선택하지 않는다.
```

```
<*> SCSI support
```

```
--- SCSI support type (disk, tape, CD-ROM)
```

```
<*> SCSI disk support
```

```
# 스카시 디스크가 부팅 디스크일 경우 반드시 커널에 포함시킨다.
```

```

# 모듈로 설정하지 말고 정적으로 포함시키기 바란다.
(40) Maximum number of SCSI disks that can be loaded as modules
< > SCSI tape support
< > SCSI OnStream SC-x0 tape support
< > SCSI CD-ROM support
< > SCSI generic support
# CD-Writer나 스캐너등의 장치가 있을 경우 선택한다.
---- Some SCSI devices (e.g. CD jukebox) support multiple LUNs
[*] Enable extra checks in new queueing code
# 이 옵션은 시스템의 성능을 저하시키기는 하지만 문제가 발생시 시스템이
# 크래쉬되지 않고 커널 패닉이 일어나게 해 준다. 선택하기 바란다.
[ ] Probe all LUNs on each SCSI device
# 한 개의 scsi ID 에 여러 논리 장치를 갖는 SCSI 를 사용시 필요하다.
# 선택하지 않는 것이 좋고, 또한 안전하므로 선택하지 않는다.
[*] Verbose SCSI error reporting (kernel size +=12K)
# 커널의 크기가 다소 커지기는 하지만 문제발생시 상세한 에러 정보를 제공하므로
# 선택하는 것이 좋다.
[ ] SCSI logging facility
    SCSI low-level drivers ---->
< > 3ware Hardware ATA-RAID support
< > 7000FASST SCSI support
< > ACARD SCSI support
< > Adaptec AHA152X/2825 support
< > Adaptec AHA1542 support
< > Adaptec AHA1740 support
<*> Adaptec AIC7xxx support
(253) Maximum number of TCQ commands per device (NEW)
        (15000) Initial bus reset delay in milli-seconds (NEW)
# 자신의 SCSI 컨트롤러에 맞는 메뉴를 선택한다.
# 필자의 시스템은 AIC7xxx 를 사용하여 이 메뉴를 선택한 것이므로 lsmod(커널
# 컴파일을 처음하는 경우)나 dmesg | grep scsi 또는 ls -la /proc/scsi/로
# 현재의 시스템에서 어떤 SCSI 를 사용하는지 확인 후 선택하면 된다.
< > Old Adaptec AIC7xxx support
< > AdvanSys SCSI support
< > Always IN2000 SCSI support
< > AM53/79C974 PCI SCSI support
< > AMI MegaRAID support
< > BusLogic SCSI support
< > Compaq Fibre Channel 64-bit/66Mhz HBA support
< > DMX3191D SCSI support
< > DTC3180/3280 SCSI support
< > EATA ISA/EISA/PCI (DPT and generic EATA/DMA-compliant boards) support
< > EATA-DMA [Obsolete] (DPT, NEC, AT&T, SNI, AST, Olivetti, Alphasronix)
support
< > EATA-PIO (old DPT PM2001, PM2012A) support
< > Future Domain 16xx SCSI/AHA-2920A support
< > GDT SCSI Disk Array Controller support
< > Generic NCR5380/53c400 SCSI support

```

```

< > IBM ServeRAID support
< > Initio 9100U(W) support
< > Initio INI-A100U2W support
< > NCR53c406a SCSI support
< > NCR53c7,8xx SCSI support
< > NCR53C8XX SCSI support
< > SYM53C8XX SCSI support
[ ] enable profiling
[ ] use normal IO
[ ] include support for the NCR PQS/PDS SCSI card
[ ] assume boards are SYMBIOS compatible (EXPERIMENTAL)
< > PAS16 SCSI support
< > PCI2000 support
< > PCI2220i support
< > PSI240i support
< > Qlogic FAS SCSI support
< > Qlogic ISP SCSI support
< > Qlogic ISP FC SCSI support
< > Qlogic QLA 1280 SCSI support
< > Seagate ST-02 and Future Domain TMC-8xx SCSI support
< > Simple 53c710 SCSI support (Compaq, NCR machines)
< > Symbios 53c416 SCSI support
< > Tekram DC390(T) and Am53/79C974 SCSI support
< > Trantor T128/T128F/T228 SCSI support
< > UltraStor 14F/34F support
< > UltraStor SCSI support
< > SCSI debugging host simulator (EXPERIMENTAL)

```

```
PCMCIA SCSI adapter support --->
```

```
#####
```

```
14. Fusion MPT device support ---> N
```

```
#####
```

```
# 2.4.8에서 처음 소개되는 메뉴로 Fusion MPT 장치를 지원하면 선택한다.
```

```
#####
```

```
15. IEEE 1394 (FireWire) support ---> N
```

```
#####
```

```
# Firewire나 I링크등 IEEE1394를 지원할 경우 선택한다.
```

```
#####
```

```
16. I2O device support ---> N
```

```
#####
```

```
# 입출력을 CPU 에서 하는 것이 아니라 장치에 있는 프로세서가 하는 I2O 장치가 지원될
# 경우에 선택한다.
```

```
#####
```

```
17. Network device support --->
```

```
#####
```

```
# 이더넷 카드등 네트워크 장비에 대한 설정으로 중요한 부분이다.
```

```
[*] Network device support
```

```
ARCnet devices --->
```

```
# 랜방식의 일종인 ARCnet을 사용할 때 지정한다.
```

```

<M> Dummy net driver support
# ADSL 등 SLIP/PPP를 사용할 때 선택한다.
# 모듈로 설정하는 것이 좋다.
< > Bonding driver support
# 이더넷 카드를 두 장 연결하여 속도를 증속시키고자 할 때 사용하는 것으로
# Cisco 에서는 채널링이라 하며 리눅스에서는 본딩이라 한다.
# 이를 설정했다 하더라도 연결되는 스위치에서도 지원하여야 제 속도를 낼 수 있다.
< > EQL (serial line load balancing) support
# PPP연결을 여러 개 연결하여 하나의 라인으로 속도를 향상시키는 기술이다.
< > Universal TUN/TAP device driver support
< > Ethertap network tap (OBSOLETE)
Ethernet (10 or 100Mbit) --->
[*] Ethernet (10 or 100Mbit)
# 일반적인 네트워크 장치는 거의 이더넷을 사용하므로 선택한다.
# 이하의 메뉴에서 자신에게 맞는 이더넷 장치(랜카드)를 선택하면 된다.
[ ] ARM EBSA110 AM79C961A support
< > Sun Happy Meal 10/100baseT support
< > Sun GEM support
[ ] 3COM cards
# 3Com 일 경우 선택 후 아래 메뉴에서 해당하는 카드의 종류를 선택한다.
# 최근에는 3Com 계열중 3c590/3c900 를 많이 사용한다.
< > 3c501 "EtherLink" support (NEW)

< > 3c503 "EtherLink II" support (NEW)

< > 3c505 "EtherLink Plus" support (NEW)

< > 3c507 "EtherLink 16" support (EXPERIMENTAL) (NEW)

< > 3c509/3c529 (MCA)/3c579 "EtherLink III" support (NEW)

< > 3c515 ISA "Fast EtherLink" (NEW)

< > 3c590/3c900 series (592/595/597) "Vortex/Boomerang" support
(NEW)
< > AMD LANCE and PCnet (AT1500 and NE2100) support
[ ] Western Digital/SMC cards
[ ] Racal-Interlan (Micom) NI cards
< > AT1700/1720 support (EXPERIMENTAL)
< > DEPCA, DE10x, DE200, DE201, DE202, DE422 support
< > HP 10/100VG PCLAN (ISA, EISA, PCI) support
[ ] Other ISA cards
# ISA 는 요즘 대부분 사용되지 않는다.
[*] EISA, VLB, PCI and on board controllers
# 대부분 PCI를 사용하므로 선택한다.3Com 을 선택했으면 선택할 필요 없다.
< > AMD PCnet32 PCI support
< > Adaptec Starfire support (EXPERIMENTAL)
< > Ansel Communications EISA 3200 support (EXPERIMENTAL)
< > Apricot Xen-II on board Ethernet

```

```

< > CS89x0 support
< > DECchip Tulip (dc21x4x) PCI support
< > Generic DECchip & DIGITAL EtherWORKS PCI/EISA
< > Digi Intl. RightSwitch SE-X support
< > Davicom DM910x/DM980x support
<*> EtherExpressPro/100 support
# 시스템에서 사용중인 이더넷 카드를 선택한다.
# 필자의 경우에는 eepr100 을 사용하므로 이를 선택하였다.
# 시스템에서 어떤 이더넷 카드를 사용하는지 아는 방법은 lsmod(커널 컴파일을
# 처음하는 경우)나 dmesg|grep eth0으로 확인하면 된다. 모듈로 지정할 수도
# 있으나 커널에 직접 포함시켰을 때 더욱 향상된 성능을 기대할 수 있으므로
# 가급적 커널에 포함시키는 것이 좋다.
< > Myson MTD-8xx PCI Ethernet support
< > National Semiconductor DP8381x series PCI Ethernet support
< > PCI NE2000 and clones support (see help)
# PCI 중 리얼텍 8019나 8029는 NE2000으로 선택한다.
< > RealTek RTL-8139 PCI Fast Ethernet Adapter support
# 리얼텍 8139 사용시 선택한다.
< > SiS 900/7016 PCI Fast Ethernet Adapter support
< > SMC EtherPower II
< > Sundance Alta support
< > TI ThunderLAN support
< > VIA Rhine support
< > Winbond W89c840 Ethernet support
[ ] Pocket and portable adapters
Ethernet (1000 Mbit) --->
# 기가비트 이더넷 카드 사용시 선택한다.
[ ] FDDI driver support
# 광케이블 이용시 선택한다.
[ ] HIPPI driver support (EXPERIMENTAL)
< > PPP (point-to-point protocol) support
# 케이블 모뎀이나 ADSL 등을 통해 인터넷을 이용하거나 이더넷 카드를 가지고
# 있다면 선택할 필요 없다. 전화선을 이용한다면 선택하고 아래의
# "PPP support for async serial ports" 도 선택한다.
< > SLIP (serial line) support
# PPP 이전에 주로 사용되었으나 요즘은 거의 사용되지 않는다.
Wireless LAN (non-hamradio) --->
# 무선 랜카드를 이용시 사용한다.
Token Ring devices --->
[ ] Fibre Channel driver support
< > Red Creek Hardware VPN (EXPERIMENTAL)
< > Traffic Shaper (EXPERIMENTAL)
Wan interfaces --->
# X.25나 프레임 릴레이등을 연결하여 WAN 라우터로 이용시 선택한다.
#####
18. Amateur Radio support ---> N
#####
# 아마추어 라디오로 통신할 때 선택한다.

```

```

#####
19. IrDA (infrared) support ---> N
#####
# 무선 네트워크등에 사용되는 적외선 프로토콜을 사용시 선택한다.
#####
20. ISDN subsystem ---> N
#####
# ISDN 카드를 연결하여 통신할 때 필요하다.
#####
21. Old CD-ROM drivers (not SCSI, not IDE) N
#####
# 구형 드라이브를 사용할 경우 선택한다.
#####
22. Input core support ---> N
#####
# USB 키보드,마우스 및 조이스틱을 사용할 경우 선택한다.
#####
23. Character devices --->
#####
# 터미널, 프린터등 다양한 장치를 지원한다.
[*] Virtual terminal
# 콘솔에서 ALT키와 F1,F2등과 같이 펑션키를 이용하여 가상 터미널을
사용시 선택한다.
[*] Support for console on virtual terminal
# 가상 터미널을 콘솔로 사용시 선택한다.
<*> Standard/generic (8250/16550 and compatible UARTs) serial support
# 표준 시리얼 포트를 사용할 때 필요하며 선택하는 것이 좋다.
[ ] Support for console on serial port
# VGA 카드가 없는 리눅스를 가능하게 하며 잘 모르겠으면 선택하지 않는다.
[ ] Extended dumb serial driver options
[ ] Non-standard serial port support
[*] Unix98 PTY support
# telnet,ssh, rlogin등으로 접속시 가상 터미널이 사용되는데,
# /dev/pts/번호 장치를 가상 터미널로 사용시 선택한다. 반드시 선택한다.
(256) Maximum number of Unix98 PTYs in use (0-2048)
# 최대 가상 터미널 개수를 설정한다.
I2C support --->
# 필립스에서 개발한 시리얼 버스 프로토콜로 리눅스에서 비디오나
오디오
# 캡처시 사용된다.
Mice --->
# 마우스를 사용하기 위한 설정이다.
Joysticks --->
# 조이스틱을 사용하기 위한 설정이다.
< > QIC-02 tape support
Watchdog Cards --->
# 와치독이란 시스템이 다운되어 작동하지 않을 때 자동으로 리부트하는
# 기능으로 하드웨어/소프트웨어 와치독을 지원한다.

```

```

< > Intel i8x0 Random Number Generator support
    # 인텔 보드에 내장된 난수 발생 기능을 지원한다.
< > /dev/nvram support
< > Enhanced Real Time Clock Support
    # 실시간 시계기능을 사용할 수 있도록 한다.
    # SMP(Dual CPU) 를 사용한다면 선택할 것을 권장한다.
< > Double Talk PC internal speech card support
    # 음성 합성장치 드라이버이다.
< > Siemens R3964 line discipline
    # 지멘스 R3964 프로토콜을 지원할 것인지 선택한다.
< > Applicom intelligent fieldbus card support
< > Sony Vaio Programmable I/O Control Device support
Ftape, the floppy tape device driver --->
< > /dev/agpgart (AGP Support)
    # AGP 기능이 지원되는 비디오 카드 사용시 선택한다.
[ ] Direct Rendering Manager (XFree86 DRI support)
    # Xfree80 에서 Direct Rendering 사용시 선택한다.
PCMCIA character devices --->

```

```
#####
```

```
24. Multimedia devices ---> N
```

```
#####
```

```
    # 오디오/비디오 캡처 드라이브 사용시 선택한다.
```

```
#####
```

```
25. File systems --->
```

```
#####
```

```
    # 리눅스에서 사용 가능한 파일 시스템을 선택한다.
```

```
    [*] Quota support
```

```
        # 유저나 그룹별로 사용 용량을 제한할 수 있는 쿼터 기능을 사용할
```

```
        # 것인지 선택한다. 많은 유저가 사용하는 시스템이라면 선택한다.
```

```
    < > Kernel automounter support
```

```
        # 원격의 파일 시스템을 자동으로 마운트시 선택한다.
```

```
        # 특별한 이유가 없으면 선택하지 않는다.
```

```
    < > Kernel automounter version 4 support (also supports v3)
```

```
        # 위 기능보다 향상된 기능을 제공한다.
```

```
    < > Reiserfs support
```

```
        # 저널링 파일 시스템인 라이저 파일 시스템을 지원하고자 할 때 선택한다.
```

```
        # 저널링 파일 시스템은 별도의 로그가 있으므로 정전등 비정상적인
```

```
종료시에도
```

```
        # 안전하게 부팅할 수 있도록 해 준다.일반적으로 라이저 파일 시스템은 EXT2
```

```
        # 보다 큰 디렉토리에 작은 파일들이 많은 경우 매우 효율적인 것으로 알려져
```

```
        # 있다. 참고 : http://www.reiserfs.org/
```

```
    < > ADFS file system support
```

```
        # Risc 에서 사용하는 파일 시스템이다.
```

```
    < > Amiga FFS file system support (EXPERIMENTAL)
```

```
        # 아미가 시스템에서 사용하는 파일 시스템이다.
```

```
    < > Apple Macintosh file system support (EXPERIMENTAL)
```

```
        # 맥킨토시 형식의 파일을 읽고 쓸 수 있다.
```

```
    < > BFS file system support (EXPERIMENTAL)
```



```

    # Unixware 에서 사용된다.
< > DOS FAT fs support
    # MSDOS 나 Windows 9x 형식의 파일 시스템을 사용할 수 있도록 한다.
< > EFS file system support (read only) (EXPERIMENTAL)
    # IRIX 에서 사용되는 파일 시스템이다.
< > Compressed ROM file system support
    # 임베디드 시스템에서 압축된 파일 시스템 이용시 선택한다.
[*] Virtual memory file system support (former shm fs)
    # tmpfs 는 가상 메모리에 파일을 저장하는 파일 시스템이다.
    # 이를 선택하면 mkdir /var/shm 으로 디렉토리 생성후 /etc/fstab 에
    # tmpfs          /var/shm          tmpfs          defaults          0 0
    # 과 같이 한 줄을 추가한다.
< > Simple RAM-based file system support
<*> ISO 9660 CDROM file system support
    # 일반적인 cdrom 에 사용되는 표준 파일 시스템이다.
[ ] Microsoft Joliet CDROM extensions
< > Minix fs support
< > FreeVxFS file system support (VERITAS VxFS(TM) compatible)
< > NTFS file system support (read only)
    # Windows NT 에서 사용되는 파일 시스템을 읽을 때 필요하다.
< > OS/2 HPFS file system support
[*] /proc file system support
    # /proc 는 시스템의 각종 상태에 대한 정보를 실시간으로 기록하는 파일
    # 시스템으로 매우 중요하다. 반드시 선택한다.
[ ] /dev file system support (EXPERIMENTAL)
[*] /dev/pts file system for Unix98 PTYs
    # Unix98 PTY 를 선택했다면 반드시 선택한다.
< > QNX4 file system support (read only) (EXPERIMENTAL)
< > ROM file system support
<*> Second extended fs support
# Second extend fs 은 리눅스에서 주로 사용해 온 ext2 파일 시스템을 뜻한다.
# 대부분이 이 파일 시스템을 사용하나 와우 리눅스 7.1이나 수세 리눅스등 일부
# 시스템에서는 ext2 대신 Reiser 라는 파일 시스템을 사용하므로 Reiser 를 사용할
# 경우에는 이 설정을 해제하고 대신 상단의 Reiserfs support 를 선택한다.
# 자신이 현재 어떤 파일 시스템을 사용하고 있는 지는 /etc/fstab 파일을 보면
# 된다.
< > System V/Xenix/V7/Coherent file system support
< > UDF file system support (read only)
< > UFS file system support (read only)
Network File Systems --->
    # 네트워크를 통해 원격지의 파일 시스템을 마운트하는 NFS 를 사용시 선택한다.
    # 그러나 보안상 좋지 않으므로 가급적 사용하지 않는 것이 좋다.
Partition Types --->

```

```
#####
```

```
26.      Console drivers --->
```

```
#####
```

```
[*] VGA text console
```

```
# VGA를 지원하는 그래픽 카드에서 텍스트 모드를 지원한다. 반드시 선택한다.
[ ] Video mode selection support
< > MDA text console (dual-headed) (EXPERIMENTAL)
Frame-buffer support --->
```

```
#####
```

```
27. Sound ---> N
```

```
#####
```

```
# PC등에서 사운드 카드를 사용할 때 자신에게 맞는 카드를 선택한다.
# 물론 서버로 사용한다면 불필요한 기능이다.
```

```
#####
```

```
28. USB support ---> N
```

```
#####
```

```
# 기존의 시리얼 포트보다 빠른 속도와 다양한 기능을 지원하는 USB를 사용하려면
# 이 옵션을 선택한다. 스캐너,키보드,마우스등 최대 127개의 주변기기를 하나의
# USB 포트에 연결할 수 있다.
```

```
#####
```

```
29. Bluetooth support ---> N
```

```
#####
```

```
#####
```

```
30. Kernel hacking --->
```

```
#####
```

```
[*] Magic SysRq key
# Magic SysRq key는 시스템의 제어가 불가능한 상태(일반적으로 '다운'되었다고
한다.)
# 예서도 제어를 가능하게 해주므로 Magic SysRq 를 이용하면 비정상적인 종료시
# 안전하게 재부팅 할 수 있다.
# 이 메뉴를 선택했다 하더라도 echo 1 > /proc/sys/kernel/sysrq 를 실행해 주어야 이
# 기능이 활성화된다.
# 만약 시스템이 다운 되었다면 같은 다음과 순서의 조합으로 시스템에 손상을 # 주지
않고 안전하게 재부팅 할 수 있다. # Alt+SysRq+S : 버퍼에 있는 것들을 디스크로
sync(동기화)시킨다. # Alt+SysRq+E : init 프로세스를 제외한 모든 프로세스를
종료시킨다. # Alt+SysRq+U : 파일 시스템을 unmount 시킨다. # Alt+SysRq+B :
시스템 재부팅
```

Kernel Hacking 설정이 끝난 후

```
Load an Alternate Configuration File
```

```
Save Configuration to an Alternate File 메뉴는 설정할 필요 없으며 Exit를 선택하면
```

```
Do you wish to save your new kernel configuration?
```

```
< Yes > < No >
```

와 같이 묻는데, 이때 Yes를 선택하면 각 메뉴에 대해 지정한 사항이 .config 파일에 저장되게 된다. 만약 No 를 선택하면 방금 선택한 사항이 make menconfig 를 선택할 때와 같은 초기화 상태로 돌아가게 되므로 주의하기 바란다.

이제 본격적으로 커널 컴파일을 하도록 하자.

```
9. [root@work /usr/src/linux]# make dep (의존성 검사)
```

```
10. [root@work /usr/src/linux]# make clean (찌꺼기를 없앤다)
```

```
11. [root@work /usr/src/linux]# make bzlilo (커널 이미지를 압축하여 생성한다.)
```

12. [root@work /usr/src/linux]# make modules (모듈을 만든다.)
13. [root@work /usr/src/linux]# make modules_install (모듈을 설치한다.)
14. [root@work /usr/src/linux]# depmod -a 2.4.8 (커널 버전을 지정해 준다.)

또는 아래와 같이 각각의 명령어에 ';'로 구분하여 한 번에 실행할 수도 있다.

```
[root@work /usr/src/linux]# make dep; make clean; make bzlilo; make modules;
make modules_install ; depmod -a 2.4.8
```

에러 없이 커널 컴파일이 끝난 후에는 / 디렉토리에 커널 이미지 등 두개의 파일이 생기게 된다. 아래와 같이 두 파일을 /boot 디렉토리로 옮긴다. 두 파일 중 System.map 파일은 커널 디버깅을 위해 필요한 파일로 어떤 부분에서 에러가 발생했는지 등의 정보를 알 수 있게 해 주는 텍스트 파일로서 부팅시에는 사용되지 않고 부팅 후 디버깅등에만 사용된다. 그리고 vmlinuz가 바로 커널 컴파일을 통해 생성된 커널 이미지 파일로 가장 중요한 파일이다.

```
[root@work /usr/src/linux]# cd /boot
[root@work /boot]# mv /System.map System.map_2.4.8
[root@work /boot]# mv /vmlinuz vmlinuz_2.4.8
```

15. 기존의 vmlinuz와 System.map의 링크를 삭제한 후 새로 컴파일하여 생성된 파일을 가리키도록 링크한다.

```
[root@work /boot]# rm f System.map vmlinuz
[root@work /boot]# ln -s vmlinuz_2.4.8 vmlinuz
[root@work /boot]# ln -s System.map_2.4.8 System.map
```

16. 이제 lilo의 설정 파일인 /etc/lilo.conf 를 편집할 차례이다. vi /etc/lilo.conf 로 리로 설정파일을 읽어 아래와 같이 설정을 추가한다. 혹 어떠한 이유로 새롭게 컴파일한 커널로 정상적인 부팅이 되지 않을 수도 있으니 기존의 설정을 삭제하지 말고 기존의 설정을 그대로 복사하여 하단에 추가 하기 바란다.

```
lba32
boot=/dev/sda
map=/boot/map
install=/boot/boot.b
prompt
timeout=50
default=v248      <-- default 를 기존의 linux 에서 새로운 커널 이미지 label 로
수정한다.
```

```
image=/boot/vmlinuz-2.2.16-3kr2smp      (기존 커널 이미지 설정)
    label=linux
    initrd=/boot/initrd-2.2.16-3kr2smp.img
    read-only
    root=/dev/sda8
```

```
image=/boot/vmlinuz-2.2.16-3kr2      (기존 커널 이미지 설정)
    label=linux-up
    initrd=/boot/initrd-2.2.16-3kr2.img
```

```
read-only
root=/dev/sda8
```

```
image=/boot/vmlinuz_2.4.8          (새로운 커널 이미지 설정)
label=v248
read-only
root=/dev/sda8
```

새로운 커널 이미지 설정에서는 SCSI를 커널에 포함시켰으므로 initrd 는 필요 없다.

17. /sbin/lilo를 실행하여 변경한 lilo.conf 의 내용을 적용한다.

```
[root@work /boot]# /sbin/lilo
Added linux
Added linux-up
Added v248 *
```

위와 같이 에러 없이 메시지가 나오면 이제 재부팅을 하여 새로운 커널을 적용하기만 하면 된다. 만약 에러가 나면 어떤 이유 때문에 에러가 나는지 확인 가능하므로 lilo 실행시 에러 메시지를 주목하기 바란다.

```
[root@work /boot]# reboot
```

18. 문제 없이 부팅이 되면 로그인 후 새로운 커널로 적용이 되었는지 확인해 보도록 한다.

```
[root@work /root]# uname -r
2.4.8
```

참고 1. 커널 컴파일을 여러 시스템에서 해야 할 경우-----

만약 커널 컴파일을 하여야 할 시스템이 1 대가 아니라 여러 대라면 일일이 위와 같은 과정을 거치는 것이 여간 번거롭지 않을 수 없다. 그러한 경우에는 시스템의 하드웨어 구성이 완전히 똑같은 경우에는 커널 컴파일 과정없이 vmlinuz 와 System.map 파일만 그대로 복사해서 /etc/lilo.conf 에 설정하여 부팅해도 가능하다.

make menuconfig 에서 설정한 커널 옵션의 설정 내용은 /usr/src/linux 에 .config 라는 파일에 저장되므로 하드웨어 구성이 다른 시스템이 있다면 다른 리눅스 시스템의 커널 컴파일시 기존의 서버에서 이 파일을 복사하여 컴파일 하려는 시스템의 /usr/src/linux 디렉토리에 복사하여 make menuconfig 를 하면 기존의 선택 사항이 그대로 적용되므로 새롭게 커널을 컴파일할 시스템에 해당하는 몇 가지 옵션만 변경하여 그대로 사용하는 방법을 이용할 수도 있다.

참고 2. 만약 커널 컴파일후 커널 옵션을 변경하여야 할 경우-----

커널 컴파일을 하여 사용하다가 이더넷 카드를 변경하거나 또는 커널 옵션을 변경하여야 할 필요가 있을 경우 부득이 재컴파일을 하여야 하는데, 위와 같은 과정을 처음부터 할 필요 없이 make menuconfig 에서 적절히 옵션 변경후 make bzlilo 를 하여 11번 이후의 과정만 반복하면 된다.

참고 3. 커널 컴파일을 했는데 예전 버전으로 부팅될 경우-----
이러한 경우는 /etc/lilo.conf 를 변경후 /sbin/lilo 를 실행하지 않았기 때문이다.
/sbin/lilo로 LILO 정보를 갱신해 주었는지 확인하고 아울러 /etc/lilo.conf 설정에 이상이
없는지도 확인해 보기 바란다.

참고 4. 부팅시 Kernel Panic: VFS: Unable to mount fs와 같은 에러 메시지가 뜨는
경우--
커널 컴파일을 한 후 재부팅시 Kernel Panic: VFS: Unable to mount fs 와 같은
메세지가
나면서 부팅이 되지 않는다면 부팅 과정에서 루트 파일 시스템(/)을 마운트 하지 못했기
때문이다. 이러한 경우에는 대부분이 두 가지 이유인데, 일단 기존의 커널 이미지(linux)
로 다시 부팅한 후 lilo.conf 에서 지정한 root= 부분이 실제 시스템의 루트 파티션과
맞는지 그리고 사용하는 시스템의 하드 디스크(IDE 또는 SCSI) 를 커널 컴파일 옵션에서
선택하였는지 그리고 파일시스템(ext2나 Reiser등)도 제대로 선택하였는지 확인해 보기
바란다.

참고 사이트
http://kldp.org/리눅스_커널/커널_컴파일/
<http://kernel.pe.kr/>

진행 | 김태진 기자 | jiny@pserang.co.kr