

제7장 리눅스 활용

1. 동영상 플레이어
2. LPP (Linux Progress Patch)
3. Webalizer 사용하기
4. 커널 컴파일 하기
5. LVM(Logical Volume Manager)을 이용한 동적 파티션 구현
6. APM(Apache + PHP + MySQL) 사용하기
7. MRTG(Multi Router Traffic Grapher) 사용하기
8. 파란 웹메일
9. CVS를 이용한 공동작업
10. 제로보드의 활용

1. 동영상 플레이어

(1) MPlayer

와우리눅스 7.1 Paran 에는 강력한 리눅스용 동영상 재생기 MPlayer 가 기본 설치되어 있어, 별도의 작업 없이, 바로, 동영상을 즐길 수 있습니다.

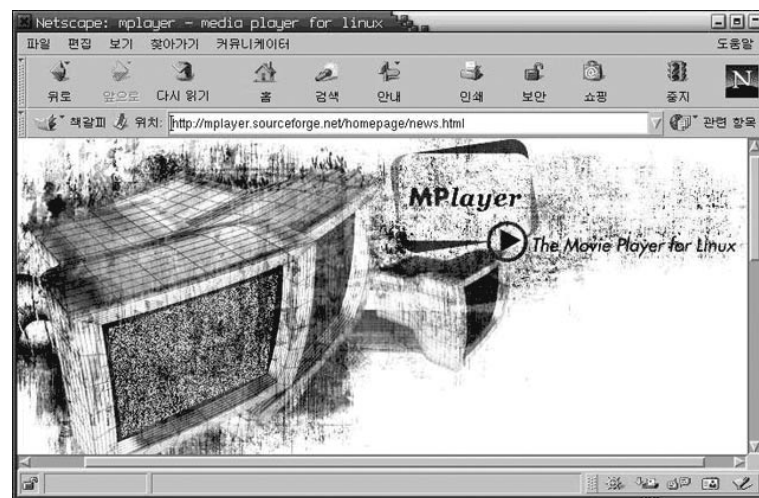
MPlayer 는 제작자가 기존의 재생기의 기능에 불만을 가지고 제작한 만큼 강력한 기능을 자랑합니다. 또한 국내 사용자에 의해 이 패키지의 SMI 자막 지원 패치까지 만들어져 더 강력한 기능의 패키지로 paran에 포함되었습니다.

MPlayer 의 홈페이지는 <http://mplayer.sourceforge.net> 입니다.

MPlayer 0.17 "The IdegCounter" 버전이 지원하는 것은 다음과 같습니다.

- DVD, VCD 지원
- OSD (On Screen Display) 지원
- 서브타이틀(자막) 지원
- DivX/AVI/ASF/MPEG1/MPEG2 지원
- 다른 15개의 비디오 출력 지원
- 리모트 콘트롤 지원

더 많은 정보는 홈페이지에서 찾아 볼 수 있습니다.



[그림 Ⅷ-1]

① MPlayer 환경 설정

MPlayer의 환경 설정 파일들은 각 유저의 홈디렉토리 안에 .mplayer 디렉토리 안에 들어 있습니다. 이것은 유저를 생성할 때, /etc/skel/.mplayer 디렉토리 안의 내용이 복사된 것입니다.

와우리눅스 7.1 Paran의 초기 인스톨 시에는 root로는 mplayer를 실행할 수 없도록 되어 있으므로, root로 실행시키기 원하는 사용자는 /etc/skel/.mplayer 디렉토리를 /root 디렉토리 아래로 복사해 넣은 후 실행합니다.

② MPlayer 실행하기

asf, avi, divx, mpeg1, mpeg2, vod 형식의 동영상을 준비합니다.

비디오 시디나 DVD 를 재생할 때는 마운트를 하지 않아도 됩니다. MPlayer가 직접 접근하여 재생합니다.

MPlayer는 아직 GUI 메뉴 방식을 지원하지 않으므로, 터미널 상에서, 아래와 같이 실행합니다.

```
# mplayer matrix.avi
```

MPlayer 는 동영상에 따라 별도로 옵션을 지정을 하지 않고, 동영상 형식을 자동으로 분석하여 알맞는 코덱으로 재생을 합니다.

또한, 동영상의 자막 파일이 디렉토리 내에 함께 존재하면, 자동으로 자막을 볼 수 있도록 패치 되어 있습니다.

MPlayer 의 종료는 ESC 키나 'q' 를 누르면 됩니다.



[그림 Ⅷ-2] matrix.avi 파일 재생 화면

③ MPlayer의 실행 옵션

MPlayer 에는 많은 옵션들이 있습니다. 그중에서 일반적으로 사용되는 옵션에 대해 설명하겠습니다.
자세한 옵션은 mplayer --help 를 하시거나, 메뉴얼 페이지(man mplayer)를 보시면 됩니다.

옵 션	설 명
-vo <드라이버>	출력 드라이버를 선택할수 있습니다. 사용할수 있는 드라이버는 -vo help 하면 됩니다. x11, sdl, x11/xv, dga, framebuffer 등을 지원합니다. ※ mplayer 의 동영상 재생 화면을 캡처하기 위해서는 출력 드라이버를 x11 로 설정 하여 화면을 캡처하면 됩니다. \$ mplayer -vo x11 <동영상 파일>
-vcd <트랙>	비디오 CD의 재생할 트랙을 지정합니다. 비디오 시디를 마운트하지 않아도 됩니다.
-dvd <장치>	DVD 를 재생할 때 사용됩니다. 단, 시스템에 DVD-ROM이 있어야 합니다.
-nosound	사운드를 재생하지 않습니다.
-fs	전체화면으로 재생합니다.
-vm	XF86VidMode 로 재생합니다.
-xy <비율>	<비율> 배로 재생합니다. 예를 들어 2배 3배...

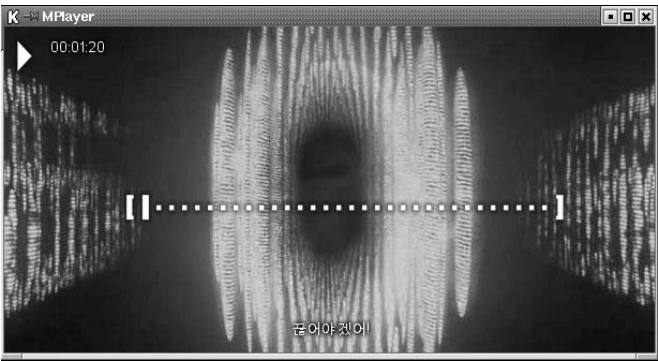
④ MPlayer 의 기능 키 사용

MPlayer 는 GUI 는 지원하지 않지만, 몇 가지 키로 기능을 제어 할 수 있습니다.

키	설 명
왼쪽 / 오른쪽 방향키	10 초 단위로 전진/후진 탐색합니다.
위 / 아래 방향키	1분 단위로 전진/후진 탐색합니다.
p / SPACE 키	동영상 재생을 잠시 정지합니다.
q / ESC 키	프로그램을 종료합니다.
+ / -	0.1초 단위로 오디오를 delay시킵니다.
/	볼륨을 줄입니다.
*	볼륨을 높입니다.
o	OSD 를 none / seek / seek + timer 로 표시합니다.
m	master/pcm 믹서장치를 선택합니다.



[그림 Ⅶ-3] 볼륨 조절 화면 (/, *)



[그림 Ⅶ-4] 탐색 화면 (방향키)



[그림 Ⅶ-5] OSD를 감춘 화면 (o)

④ 관련 링크

mplayer 홈페이지 :
<http://mplayer.sourceforge.net/>

안승님의 리눅스 동영상 프로그램 리뷰 페이지 :
<http://wxy69.pe.kr/video/video.html>

양선진님의 MPlayer SAMI 자막 패치와 한글 폰트 페이지 :
<http://realtime.ssu.ac.kr/~lethean/mplayer/>

MPlayer 사용자 메일링 리스트 :
<http://lists.sourceforge.net/lists/listinfo/mplayer-users>

MPlayer 개발자 영문 메일링 리스트 :
<http://lists.sourceforge.net/lists/listinfo/mplayer-dev-eng>

⑤ MPlayer가 지원하는 포맷

지원하는 입력 포맷
VCD (Video CD) directly from CD-ROM or from CDRwin's .bin image file
DVD, directly from your DVD disk, using optional libcss for decryption
MPEG 1/2 System Stream (PS) and Elementary Stream (ES) file formats
RIFF AVI file format
ASF v1.0 files

지원하는 오디오 & 비디오 포맷
/usr/share/doc/mplayer-0.17/CODECS, 혹은
<http://mplayer.sourceforge.net/DOCS/codecs.html>를 보세요.

(2) XINE

xine은 와우리눅스 7.1 Paran에 기본 설치되어 있는 또 다른 동영상 플레이어입니다.
xine은 mpeg, video 파일등을 재생할 수 있으며, 비디오 CD 혹은, DVD 등을 재생할 수 있습니다. 아래는 지원하는 포맷입니다.

- mpeg-1 mpeg-2 system (audio and video) streams,
 - mpeg elementary streams (video-only streams and mp3 audio streams)
 - AVI file (external win32 codecs을 사용하는).
 - DVD, VCD
- 보다 자세한 사항은 매뉴얼 페이지 (man xine) 혹은 홈페이지를 참고하세요.
xine 의 홈페이지는 <http://xine.sourceforge.net/> 입니다.

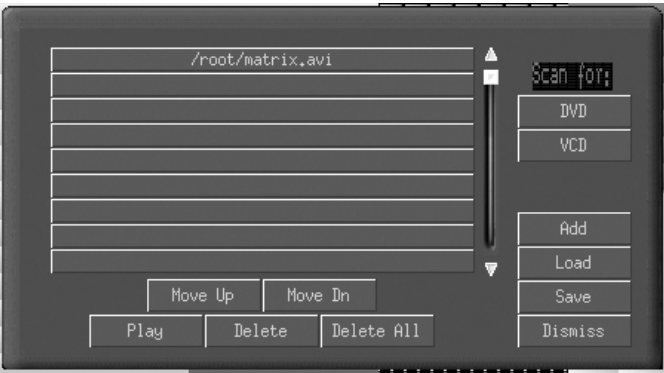
① XINE 의 실행

#xine 명령으로 xine을 실행 시킵니다.



[그림 VII-6] xine 실행 화면

플레이 하려는 동영상 파일을 추가 하기 위해서, Playlist 버튼을 누르면 아래와 같이 등록된 파일들이 나타납니다. 플레이를 원하는 파일을 선택한 후 Play 하거나, Dismiss로 창을 없앴 후 플레이 버튼 (▶)으로 플레이 합니다.



[그림 VII-7]

플레이 리스트에 동영상 파일을 추가하려면, 우측의 Add 버튼을 눌러 아래 검색창을 띄워, 원하는 파일을 선택하여, Add 하면 됩니다. 상하위 디렉토리간의 이동은 'Down To', 'Up 1 Level' 버튼을 이용하여 합니다.



[그림 Ⅷ-8]

비디오 CD를 삽입하면, 별도로 마운트 할 필요 없이, 자동으로 CD 안의 플레이할 파일들의 리스트를 보여줍니다.



[그림 Ⅷ-9] xine으로 동영상을 실행한 모습

② XINE 실행 옵션

xine을 실행할 때 아래와 같은 옵션을 주어 실행할 수 있으며, 보다 자세한 옵션은 매뉴얼 페이지를 참고 하시기 바랍니다.

키	설 명
P	프로그램이 시작하면서 바로 플레이 함
F	풀스크린으로 보여줌
H	GUI 컨트롤러 숨김
Q	플레이가 끝나면 프로그램 종료
D	DVD로부터 플레이 리스트를 받을때
V	VCD로부터 플레이 리스트를 받을 때

③ XINE의 기능 키 사용

키	설 명
ENTER	플레이
SPACE	잠시 멈춤
F	풀스크린 모드 토글
G	GUI 보이기/감추기 토글
H	Video output window 보이기/감추기 토글
A	가로세로비율조정 (자동/16:9/4:3/DVB)
Ctrl	마우스 커서 보이기/감추기 (GUI가 감춰진 경우에만 유효)
PgUp	이전 챕터
PgDown	다음 챕터

※ xine는 자막 파일이 지원 가능하나, 아직까지는 완벽하게 지원되지 않습니다.

2. LPP (Linux Progress Patch)

LPP란 Linux Progress Patch 를 말합니다. Linux Progress Patch는 프레임버퍼를 이용하여 부팅할 때 보이는 메시지들을 감추고, 대신에 풀스크린 로고와 함께 약간의 정보를 제공하는 진행바와 텍스트를 표시하는 커널 패치입니다. 이것은 커널에서 표시하는 전형적인 부팅 메시지를 숨깁니다. 그래서, 이 용자들은 더 이상 복잡하고 혼란스러운 부트메시지에 신경을 쓰지 않아도 됩니다.

LPP 프로젝트 홈페이지 주소는 <http://lpp.freelords.org/>입니다.

(1) 요구사항

vesa 모드를 지원하고 640x480 해상도에서 256(8비트) 컬러를 표시할 수 있는 그래픽 카드와 이를 지원하는 모니터

(2) LPP 패치된 로고 화면 보기

와우리눅스 7.1 Paran에서는 LPP 패치를 기본적으로 제공하고 있어, 시스템 설치시에

□ 프레임 버퍼 사용 - for LPP

항목을 선택할 경우, 부팅시에 아래와 같은 LPP 패치된 부팅 화면을 볼 수 있습니다.



[그림 Ⅶ-10]와우리눅스 7.1 Paran의 부팅 시 LPP 패치가 적용된 화면

* 만약, 설치시

□ 프레임 버퍼 사용 - for LPP

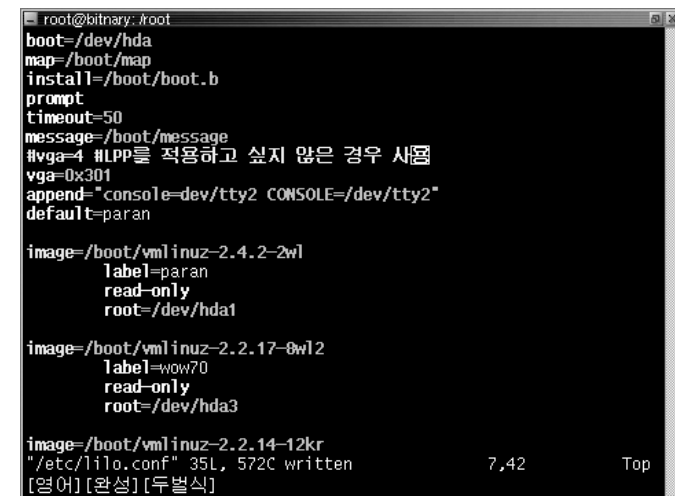
옵션을 선택했으나, 부팅시 LPP 적용된 화면을 볼 수 없다면, `/etc/lilo.conf` 파일에서, 다음의 두개 항목이 존재하는지 확인하십시오.

```
vga=0x301
```

```
append="console=/dev/tty2 CONSOLE=/dev/tty2"
```

부팅메시지를 tty2(가상콘솔2)에 보여줍니다.

부팅시 alt + tty2를 누르면 자세한 메시지를 볼 수 있습니다.



[그림 Ⅶ-11] LPP 적용시 /etc/lilo.conf의 내용

(3) 제거하는 방법

LPP가 맘에 들지 않거나, 잠시동안 LPP 부트화면을 보지 않으려면 `/etc/lilo.conf` 에서 `"vga=0x301"` 부분을 `"vga=4"` 으로 바꾸어 주면 됩니다.

(4) LPP 로고 이미지 바꾸기

와우리눅스 7.1 Paran에서는 기본 테마로 위의 LPP 화면을 사용합니다. 기본 로고가 맘에 들지 않는다면 LPP의 홈페이지에서 마음에 드는 테마를 받아서 바꿀 수 있습니다. 그러나, 이 작업을 하기 위해

서는 테마 적용을 위해 커널을 컴파일 해야 하므로, 커널 컴파일에 대한 지식이 있는 사용자가 수행할 것을 권합니다. 커널을 컴파일 하기 이전에 안전을 위하여 커널을 백업하도록 합니다.

먼저, LPP 홈페이지(<http://lpp.FreeLords.org>)로부터 LPP 패치 파일과 LPP 테마 파일을 다운로드 받아, 압축을 풉니다. lpp-0.4.0디렉토리로 이동하면, theme_install 스크립트가 있습니다. 이 스크립트를 이용하여, 다운 받은 LPP 테마를 설치합니다. 테마 파일은 tar.gz 혹은 tar.bz2 로 압축되어 있으며, lpp-0.4.0 디렉토리로 이동하여 다음과 같이 설치합니다.

```
# ./theme_install 테마이름.tar.gz /usr/src/linux-2.4. (커널 소스 디렉토리)
```

테마 파일의 압축 포맷 tar.gz이 아닌 경우, theme_install 파일을 vi로 열어, gzip 부분을 해당 압축을 푸는 명령으로 변경한 후 스크립트를 실행 하십시오.

이제 커널을 컴파일 합니다. 여기서는 커널 컴파일 방법을 별도로 설명하지 않습니다. 단 다음 옵션을 [*]체크를 하여 컴파일합니다.

Code maturity level options -> Prompt for development and/or incomplete

Console Drivers -> Video mode selection support

Frame-buffer support ->

-> Support for frame buffer devices (EXPERIMENTAL)

-> VESA Framebuffer support (RIVA/MATROX whatever should be ok)

-> Enable Linux Progress Patch (EXPERIMENTAL) (NEW)

커널 컴파일 후 lilo를 다시 실행하면, 새로운 부팅 화면을 즐길 수 있습니다.



[그림 Ⅶ-12] LPP 테마를 변경한 화면]

3. Webalizer 사용하기

(1) Webalizer 개요

Web Server를 운영하다 보면 Log 분석은 꼭 필요합니다.

Web Server의 가용성과 얼마나 많은 사용자가 Server를 사용하는지 Server 운영을 하는데 있어서 관리적인 측면이 많이 대두되고 있으므로 Log 분석기 또한 꼭 필요한 부분입니다.

Paran에는 공개용 Web Server Log 분석기인 Webalizer가 설치가 되어 있습니다.

Webalizer에 대한 최신 소식 및 자료는 <http://webalizer.dexa.org>에서 참조하시길 바랍니다.

Webalizer의 설정 및 사용방법에 대해서 자세히 살펴보기로 하겠습니다.

(2) Webalizer 설치

아래의 그림을 보면 Webalizer에 관련된 Package 및 관련된 파일의 위치를 확인할 수 있습니다.

```
peterpan@peterpan.wowlinux.com: /home/peterpan
[peterpan@peterpan peterpan]$ rpm -qa | grep webalizer
webalizer-2.01_06-8
[peterpan@peterpan peterpan]$ rpm -ql webalizer-2.01_06-8
/etc/cron.daily/00webalizer
/etc/webalizer.conf
/usr/bin/webalizer
/usr/bin/webalizer
/usr/share/doc/webalizer-2.01_06
/usr/share/doc/webalizer-2.01_06/README
/usr/share/man/man1/webalizer.1.gz
/var/www/html/usage
[peterpan@peterpan peterpan]$
```

[그림 Ⅶ-13]

■ webalizer Version 및 관련 Package → webalizer-2.01_06-8.rpm

■ Webalizer 설정파일 → /etc/webalizer.conf

■ Webalizer 실행파일 → /usr/bin/webalizer

■ Webalizer Cron Script File → /etc/cron.daily/00webalizer

■ Webalizer 실행 결과 파일 위치 → /var/www/html/usage

[주의] Paran의 Webalizer의 설정 및 파일의 위치는 다른 webalizer의 설정 값들과 차이가 있을 수 있습니다.

(3) Webalizer 설정

webalizer를 처음 설정하기 위한 절차는 다음과 같습니다.

- STEP 1. webalizer.conf 수정
- STEP 2. webalizer 실행
- STEP 3. webalizer cron script File 수정
- STEP 4. web browser 확인

■ webalizer.conf 수정

webalizer.conf File을 수정하기 전에 Web Server의 설정되어 있는 설정 값들을 참조하므로 꼭 미리 확인하고 설정하여야 합니다.
webalizer.conf File을 살펴보면 아래와 같은 부분에 주석 처리되어 있는 부분을 삭제하고 적당한 값을 입력한다.

LogFile /var/log/httpd/access_log
→ Apache Web Server의 Log File 위치
LogType web
→ ftp, web 중 하나를 선택한다. Web을 선택한다.
OutputDir /var/www/html/usage
→ webalizer가 만들어낸 html 문서 위치
HistoryName /var/www/html/usage/webalizer.hist
→ webalizer의 history File과 Name과 File의 위치 지정
Incremental yes
IncrementalName webalizer.current
Hostname wowlinux.com(예)
→ web server의 hostname을 적어 준다.
HTMLExtension html
TimeMe yes

이외의 설정값들은 자신의 Server 환경에 맞게 설정을 하면 됩니다.

■ webalizer 실행

[root@localhost usage]# webalizer
→ webalizer를 처음 실행
[root@localhost usage]# ls
→ webalizer.conf에 설정되어 있는 OutputDir에서 파일 생성 확인

■ webalizer cron script File 수정

/etc/cron.daily/00webalizer
위의 File은 하루에 한번 am 04:00에 자동으로 webalizer를 실행 시키는 파일입니다.
만약 webalizer를 한시간에 한번 실행시키려면 00webalizer File을 /etc/cron.hourly Directory 밑에 Copy를 해두면 됩니다.
빈번하게 분 단위로 Log 분석을 하는 경우에는 간단하게 Script File을 만들어 crontab에 등록을 시켜 사용하시면 됩니다.
아래는 10분에 한번씩 webalizer가 Log 분석을 하도록 하는 Script 내용입니다.

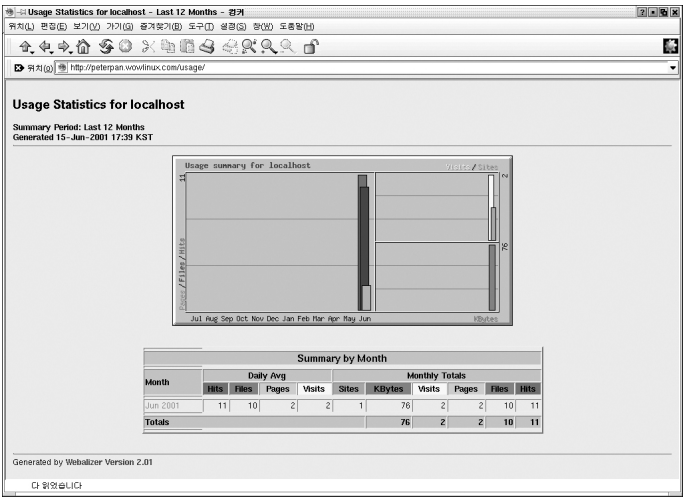
/usr/bin 밑에 weblog라는 File를 만듭니다.
weblog File 내용은 아래와 같습니다.
#!/bin/sh
/usr/bin/webalizer -c /etc/webalizer.conf
killall -HUP httpd
weblog File에 소유주만 실행할 수 있도록 권한을 부여합니다.
(예 : chmod 700 weblog)
crontab을 이용하여 cron에 등록을 합니다.
[root@localhost root]# crontab -e
*/10 * * * * /usr/bin/weblog

■ web browser 확인

이제 webalizer가 올바르게 설정이 되었다면 webalizer.conf에 설정해 놓은 Hostname을 이용하여 browesr에서 확인해 보면 아래와 같은 그림들을 볼 수 있습니다. 아래와 같은 화면이 출력되면 성공적으로 설정된 것입니다.

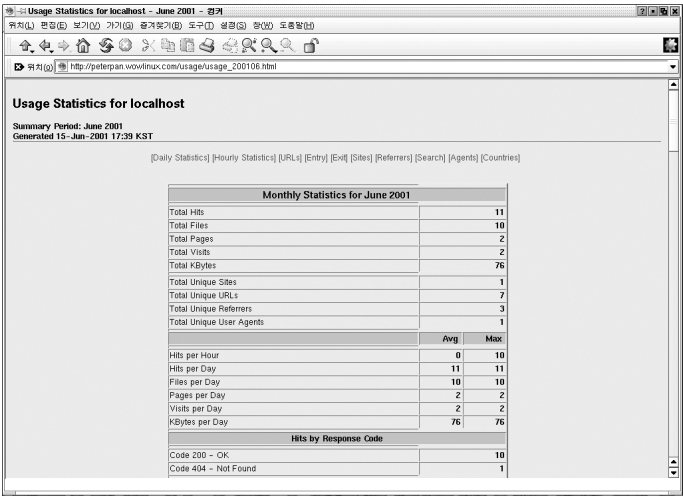
설치 URL : http://Hostname/usage

■ webalizer가 만들어내 Log 분석 초기 화면



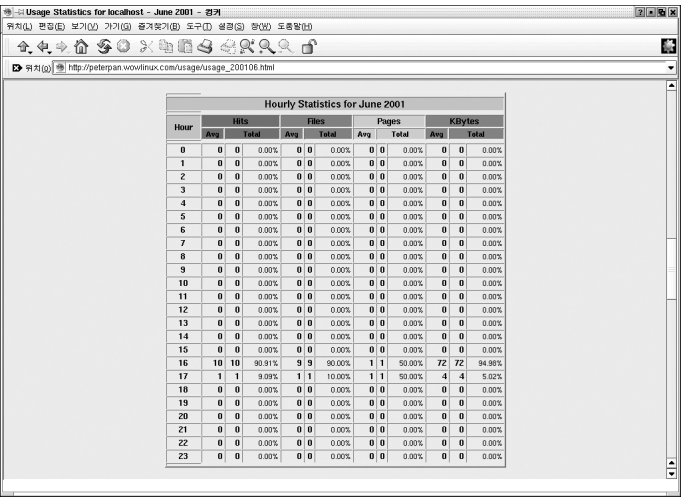
[그림 Ⅷ-14]

■ webalizer가 만들어낸 Log 분석 화면1.



[그림 Ⅷ-15]

■ webalizer가 만들어낸 Log 분석 화면2.



[그림 Ⅷ-16]

■ webalizer을 이용하며 설치 및 사용방법을 모두 끝마쳤습니다.

4. 커널 컴파일 하기

(1) 커널이란 무엇인가?

커널은 OS(Operating System)의 가장 중요한 핵심으로 사전적 의미는 핵심, (곡식의) 낱알, 요점 등의 뜻을 담고 있습니다.

커널은 CPU, Memory, 프로세스 등과 같은 시스템의 자원(Resource)을 관리하며 모든 시스템이 원활히 돌아갈 수 있도록 제어를 합니다.

우리가 현재 '리눅스' 라고 하는 것도 사실은 커널부분을 의미하며 리눅스에서의 커널이란 리눅스 그 자체를 의미한다고 볼 수 있습니다. 즉, 리눅스의 역사는 바로 커널의 역사 라고 볼 수 있습니다.

리눅스만이 커널을 가지고 있는 것은 아니며 모든 운영체제가 커널을 가지고 있습니다. 그러나 커널 소스가 공개되어 있는 것은 리눅스와 같은 공개형 운영체제로 제한 됩니다.

리눅스의 장점 중 하나는 실행 도중에 커널 코드를 확장할 수 있다는 것입니다. 이 말은 시스템 동작 중에도 커널에 기능을 추가할 수 있음을 의미합니다. 이때 커널에 추가할 수 있는 각각의 코드 부분을 모듈(module)이라고 합니다.

리눅스 커널은 여러 가지 타입의 모듈을 지원하며 디바이스 드라이버도 제한 없이 포함 가능합니다. 또한 각 모듈은(완전한 실행파일로 링크 되지 않은)오브젝트 코드들입니다.

(2) 커널 버전

커널은 안정버전(stable version)과 개발버전(beta version)으로 나뉘는데 새로운 기능들이 실험되는 개발과정의 커널을 개발 버전이라 부르며, 테스터, 개발자를 위한 버전입니다. 개발버전은 많은 테스트와 패치 과정을 거쳐 안정버전으로 발전되면 일반 사용자들에게 배포됩니다. 여러분이 사용하고 계신 와우리눅스 7.1 판의 커널 2.4.2도 2.3.x의 개발 버전이 여러 번의 패치가 가해져 충분히 안정성을 검증하여 2.4.x 의 안정버전으로 바뀐 것입니다.

커널 개발자가 아닌 이상 안정버전 커널 즉 '2.썩수.x' 버전을 사용하면 됩니다.

만일 도전정신이 충분하다면 가장 최근버전의 커널을 사용해 보는 것도 좋습니다. 그러나 그에 따른 위험은 충분히 감수할 준비가 되어 있어야 합니다.

[TIP]최신 버전의 커널 알아 내기

finger @finger.kernel.org 명령을 통해 현재의 커널 버전을 알 수 있습니다.

```
root@loveme.wowlinux.com: /root
[root@loveme /root]# finger @finger.kernel.org
[zeus.kernel.org]

The latest stable version of the Linux kernel is:      2.4.5
The latest prepatch (alpha) version *appears* to be:  2.4.6-pre3

[root@loveme /root]#
```

[그림 ㉗-17] 최신 커널버전 알아내기

[참고] Patch 에 대하여...

■ ac patch (알렌 콕스 패치)

알렌 콕스는 핵심적인 커널 개발자 중의 한 명으로 리누스 토발즈를 커널의 아버지, 알렌 콕스는 어머니라는 이야기가 있을 정도 입니다.

메일링 리스트에 보내는 수많은 커널 해커들의 패치와 알렌 콕스 자신이 만든 많은 부분의 실험적인 것을 포함하고 있는 자신만의 패치도 같이 묶어서 ac 패치로 발표하고 있습니다. 보안과 관련된 부분에 탁월한 능력을 가지고 있습니다.고 알려져 있습니다.

■ arca 패치 (Andrea Patch)

안드레이어에 의해 개발된 패치로 파격적인 내용의 패치를 적용하는 것이 특징입니다. 다른 코드에 대해 영향을 미치는 범위까지도 실험적인 내용을 포함해서 특별한 환경에서 탁월한 성능을 보여주기도 합니다.

<ftp://e-mind.com/pub/Adrea/kernel> 에서 구할 수 있습니다.

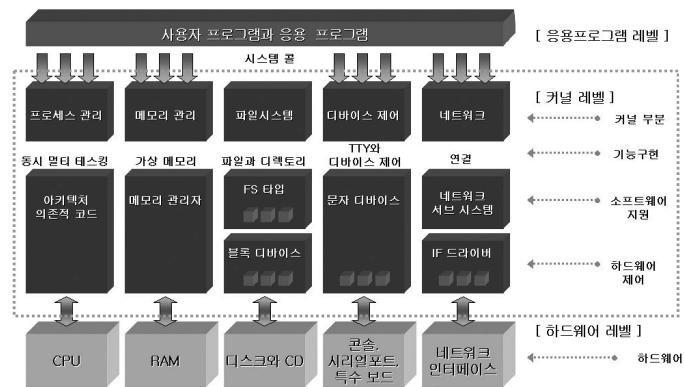
■ Pre patch (Pre-release 패치)

이 패치 들은 리누스에 의해 발표되는 다음 버전의 공식 커널에 대한 pre-release 입니다. 리누스가 다음 버전을 발표하기 전에 실제로 잘 동작하는지 테스트해 보려는 기능들이 포함됩니다.

예를 들어 2.2.7 pre에서는 일부에서 문제가 되었던 TCP/IP 네트워킹에 대한 업데이트가 포함되었습니다. Pre patch 는 더 이상 문제가 발생하지 않는다는 것을 확실하기 위해 리누스는 가능한 한 많은 사람들에 의해 그것이 테스트 되기를 원합니다.

(3) 커널의 역할

커널의 역할은 다음과 같이 크게 5가지로 나눌수 있습니다.



[그림 Ⅶ-18] 커널의 동작

① 프로세스 관리

커널은 프로세스를 생성, 제거하고 외부와 연결(입출력)하는 책임을 가지고 있습니다. 시그널, 파이프 등을 통해서 각각의 프로세스 사이의 통신은 전체 시스템 기능의 바탕이 되며 마찬가지로 커널이 관리 합니다. 게다가 운영체제 전체에서 가장 중요한 루틴인 스케줄러도 프로세스 관리의 일부분입니다.

② 메모리 관리

컴퓨터의 메모리는 가장 중요한 자원이며 이것을 다루는 정책은 시스템 성능을 결정하는 중요한 요소입니다. 커널은 제한적인 가용 자원을 가지고 모든 프로세스에게 가상 주소 공간을 제공합니다. 커널의 다른 부분은 간단한 malloc/free 루틴에서부터 훨씬 더 복잡한 기능을 가진 일련의 평선콜을 이 용해서 메모리 관리 하부 시스템과 상호 작용을 하게 됩니다.

③ 파일 시스템

유닉스는 거의 모든 것을 파일시스템 개념에 기초로 해서 처리합니다. 커널은 비구조적인 하드웨어 위에 구조적인 파일시스템을 생성합니다. 리눅스는 가장 많은 파일시스템 타입을 지원합니다.

[참고] SWAP 이란?

프로세스의 가상 페이지를 물리적 메모리에 저장하는 과정에서 물리적 메모리의 공간이 충분 하지 않을 때 가상메모리의 페이지를 저장할 수 있는 공간에 이를 저장하는 일련의 과정입니다. 리눅스 커널이 사용하는 스와핑 방식은 가장 최근에 사용된 페이지를 선택하는 LRU(Latest Recently Used)방식을 쓴다.

④ 디바이스 제어

프로세서,메모리,그리고 극히 일부의 다른 자원을 제외하고는 모든 디바이스 컨트롤 작업은 지정된 디 바이스에 따른 코드에 의해서 수행됩니다. 이 코드는 디바이스 드라이버를 가지고 있어야 하며 커널이 이 부분의 역할을 수행합니다.

⑤ 네트워크

대부분의 네트워크작업은 하나의 프로세스에 국한되지 않으므로 네트워크는 반드시 운영체제가 관리해 야 합니다. 운영체제는 프로그램과 네트워크 인터페이스 사이에서 데이터 패킷을 전달할 책임이 있으며 네트워크로부터 데이터가 전달되기를 기다리는 프로그램을 정확히 sleep상태로 만들고 깨울 수 있어야 합니다. 또한 모든 라우팅 테이블과 주소 분석에 관한 일도 커널 안에 구현 됩니다.

Parameter	Linux 2.4 limit
File Size	2 ⁶⁴ bytes, or 16,777,216 terabytes
File system Size	for ext2fs, 4 TB by default, up to 32 TB on some architectures, based on file system block size defined at creation: multiply block size by 4 GB
Logical Volume Management	Yes
Number of Open Files	8,192 by default, dynamically allocated in /proc/sys/fs/file-max, theoretical maximum of 1,048,576
Number of Partitions Mounted	256 by default, dynamically allocated in /proc/sys/fs/super-max, theoretical maximum 2,147,483,647
Number of Running Processes	Dynamically allocated in /proc/sys/kernel/threads-max, theoretical maximum 32,768
Total Memory Supported	64 GB

표. LINUX KERENL 2.4 특징

(4) 커널 컴파일 하기

커널 컴파일을 하게 되면 시스템에 가장 적합한 작은 커널을 만들어 시스템 성능을 최적화 할 수 있습니다. 또한 최신의 하드웨어 드라이버를 지원하고 가볍고 안정적이며 결함이 없는 시스템을 구축할 수 있습니다. 또한 지적 호기심의 만족시키거나 재미나 새로운 것에 대한 도전 등의 이유로 커널 컴파일을 하는 경우도 있습니다.

커널 컴파일 단계는 다음과 같습니다.

- 1. 커널 소스 받기
- 2. 커널 소스 풀기
- 3. make mrproper
- 4. 컴파일 옵션 설정
- 5. make dep; make clean
- 6. make bzlilo
- 7. 모듈 컴파일
- 8. lilo 설정

① 커널 소스 받기

커널은 ftp.kernel.org에 anonymous로 접속하여 /pub/linux/kernel 디렉토리에서 가장 최신의 커널 소스를 받을 수 있습니다.
또한 아래의 미리 사이트에서도 받을 수 있습니다.
- ftp://ftp.kr.kernel.org/pub/linux/kernel/
- http://www.kr.kernel.org

② 커널 소스 풀기

일반적으로 커널 소스는 /usr/src 디렉토리 아래 설치합니다. /usr/src 아래에는 실제 헤더, 커널이 설치된 디렉토리 와 그 링크가 있습니다. 이전 커널이 들어 있는 디렉토리에 대한 링크를 삭제하고 새 커널 소스가 설치된 디렉토리를 linux 라는 이름으로 다시 링크합니다.
가장 최신의 커널 소스를 /tmp 디렉토리에 다운로드 받았다고 가정합니다.

```
# cp /tmp/linux-2.4.2.tar.gz /usr/src
# cd /usr/src
# rm -f linux
```

```
# tar xvfz linux-2.4.2.tar.gz

# mv linux linux-2.4.2
# ln -s linux-2.4.2 linux
```

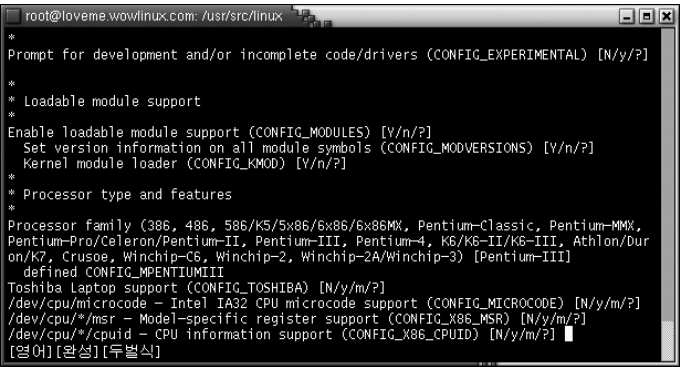
③ make mrproper

이전 커널을 컴파일할 때 만들어진 오브젝트 파일과 의존성 설정, 컴파일 환경 설정값, 버전 정보 등 새로 시작하는 컴파일에 영향을 주는 이전 정보들을 삭제합니다.

```
# cd /usr/src/linux
# make mrproper
```

④ 컴파일 옵션 설정

커널의 컴파일 방법에는 3가지 방법이 있습니다.
▶ make config : 커맨드 형식으로 옵션을 간단한 인터페이스를 지원하고 잘못 선택하면 처음부터 다시 선택해야 하는 단점이 있습니다.



[그림 Ⅷ-19] make config 실행화면

▶ make menuconfig : 가장 추천하는 방법입니다. (ncurses 라이브러리가 설치되어 있어야 합니다.)
메뉴방식의 화면으로 방향키로 메뉴를 선택할 수 있습니다.
[F1]을 누르면 도움말을 볼 수 있으며 "-" 표시가 있는 곳에서 엔터키를 누르면 하위 메뉴가 열립니다.
<Exit>를 선택하면 상위 메뉴로 올라갑니다.
설정 상태는 [] 또는 < > 안에 *(선택), M(모듈), 빈칸(제외)로 표시되며 모듈기능(M)은 "< >"으로 표시된 질문에만 설정할 수 있습니다.

㉓ 모듈 컴파일

커널 옵션에서 모듈(M)로 선택한 것을 컴파일 하기 위해 다음과 같은 명령을 합니다. 모듈이 생성되는 디렉토리는 '/lib/modules/2.4.2' 이고, 만약 현재 커널의 모듈을 다시 생성하려고 한다면 현재 커널의 모듈 디렉토리를 지우고 다시 컴파일 합니다.

커널 2.4.x 버전에서는 모듈의 디렉토리 구조가 많이 바뀌었습니다.

make modules은 커널 환경설정에서 모듈로 설정한 기능들을 컴파일하며 make modules_install 은 컴파일 된 모듈을 /lib/modules 아래에 설치합니다.

마지막으로 depmod 은 모듈 사이의 의존성을 검사하여 /lib/modules 디렉토리 아래 modules.dep 파일을 만듭니다.

```
# cd /usr/src/linux
# make modules
# make modules__install
# depmod -a 2.4.2
```

㉔ lilo 설정

```
# vi /etc/lilo.conf
boot = /dev/hda
map = /boot/map
install = /boot/boot.b
prompt
timeout = 50
message = /boot/message
vga = 0x301
append = "console = dev/tty2 CONSOLE = /dev/tty2"
default = linux

image = /boot/vmlinuz-2.4.2-2wl
    label = linux
    read-only
    root = /dev/hda1

image = /boot/vmlinuz-2.4.2
    label = linux242
    read-only
    root = /dev/hda1
```

커널 업그레이드 과정의 마지막으로 새롭게 만들어진 커널을 사용하기 위해 /etc/lilo.conf 를 수정하고 lilo를 갱신 시켜 줍니다.

이렇게 구 버전 커널과 정보를 남겨두면 새로운 커널에 부팅이 안 되는 것과 같은 문제가 생겼을 때 이전의 커널로 부팅한 다음, 문제를 조사해 볼 수 있습니다. 일종의 안전 장치인 셈입니다.

```
# /sbin/lilo
```

lilo의 갱신이 끝나면 이제 커널이 업그레이드가 된 것입니다.

이제 리부팅 후 LILO에서 새로운 커널이미지를 선택할 수 있습니다.

만일 문제가 생겼을 경우에는 이전 커널로 부팅 후 커널 옵션을 변경해 가며 새로이 컴파일 해야만 할 수도 있습니다.

마지막으로 컴파일 순서를 정리하면 다음과 같습니다.

```
# make mrproper
# make xconfig ( 또는 make menuconfig )
# make bzlilo
# mv /vmlinuz /boot/vmlinuz-2.4.2
# mv /System.map /boot/System.map-2.4.2
# make modules
# make modules__install
# depmod -a 2.4.2
# vi /etc/lilo.conf
# /sbin/lilo
```



5. LVM(Logical Volume Manager)을 이용한 동적 파티션 구현

(1) LVM의 개요

로지컬 볼륨 매니저(Logical Volume Manager, 이하 lvm)는 기존에 있던 파티션을 합쳐 새롭게 구성하는 기능입니다.

컴퓨터를 사용하다 보면 데이터가 쌓이고, 새로운 프로그램을 설치하다 보면 하드 디스크 공간이 부족한 경우가 생길 수 있습니다. fdisk(또는 cfdisk)를 사용해 한번 리눅스 파티션의 크기를 정한 후 파티션을 확장하려면 어쩔 수 없이 내용을 삭제하고, 크기를 늘려야 합니다.

이와 같은 구조는 서버에서는 큰 문제가 됩니다. 서버는 절대 정지해서는 안되는 컴퓨터이기 때문입니다.

이런 문제점을 해결하기 위해 유닉스에서 일찍부터 도입했던 것이 바로 lvm으로 리눅스 역시 커널 2.4부터 엔터프라이즈 기능 강화 차원에서 lvm을 새롭게 도입한 것입니다.

lvm은 이렇게 쉽게 파티션을 늘였다 줄였다 할 수 있으므로 데이터 사이즈의 변화가 급속히 일어 나는 대형 서버에서 아주 유용한 기능입니다.

Lvm은 IBM에 의해 최초 개발되었으며 OSF에서 최초로 적용, HP-UX와 Digital UNIX의 LVM에 근간이 되었습니다.

리눅스의 LVM은 HP-UX의 LVM 구현방식과 유사하며 1999년 3월 LVM 0.6이 발표된 이 후 커널 2.2와 2.3에서 꾸준한 테스트를 거쳐 커널 버전 2.4 에서 LVM을 손쉽게 사용할 수 있게 되었습니다. 현재 LVM 0.9.1-Beta7 이 릴리즈 된 상태입니다.

▶ 웹사이트 : <http://linux.msede.com/lvm/>

(2) 관련 용어

lvm을 사용하기 위해 알아야 할 기본적인 용어는 다음과 같습니다.

- ▶ PE(physical media) : 일반적인 물리적 하드 디스크를 말합니다.
예) /dev/hda, /dev/sda
- ▶ PV(Physical Volume) : 각각의 파티션을 나눈 것을 말합니다.
예) /dev/hda1, /dev/hda2
이 때 파일 타입은 lvm을 사용할 수 있게 0x 8e로 설정해야 합니다.
- ▶ PE(Physical Extents) : 하드 디스크를 제어할 때 블럭(PE) 단위로 제어합니다. 블럭 하나는 대개 MB 단위의 크기를 갖습니다. 예를 들면 볼륨그룹 크기가 184MB이고, 기본 PE가 4MB일 때 PE의

개수는 46개가 됩니다.

- ▶ VG(Volume Group) : PV로 되어 있는 파티션을 그룹으로 설정합니다. /dev/hda1을 하나의 그룹으로 만들 수도 있고, /dev/hda1 + /dev/hda2처럼 파티션 두 개를 하나의 그룹으로 만들 수 있습니다. 그밖에도 다양하게 그룹 설정을 할 수 있습니다.
- ▶ LV(Logical Volume) : 마운트 포인트로 사용할 실질적인 파티션입니다. 크기를 바꿀 수 있습니다.

파일 시스템은 ext2, reiserfs 등의 리눅스에서 사용하는 모든 파일시스템을 사용할 수 있습니다.

Logical Volume은 여러 개의 disk를 걸쳐서 묶을 수 있으며사이즈를 쉽게 변경할 수 있습니다. 또한 미러링이 가능하고 시스템에 하드디스크를 쉽게 추가 장착할 수 있습니다.

(3) lvm 이용 하기

① 커널 확인

lvm은 리눅스 커널 2.3 개발 버전 이후부터 사용이 가능해 졌습니다.

다음과 같은 부팅 메시지가 나오면 lvm이 정상적으로 설치된 것입니다.

```
LVM version 0.8final by Heinz Mauelshagen (15/02/2000)
lvm -- Driver successfully initialized
```

② lvm 유틸리티 설치

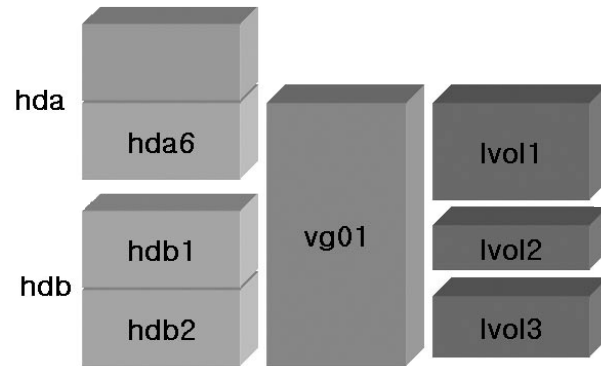
lvm을 사용하려면 먼저 lvm 관련 패키지를 설치해야 합니다.

와우리눅스 7.1 버전에 이미 포함되어 있으며 최신 버전은 0.9 입니다.

```
# tar xvfz lvm__0.9.tar.gz
# ./configure
# make ; make install
```

이제부터 6G 의 IDE HDD 2개를 이용해 LVM을 이용해 보려고 합니다.

아래의 구성도를 참고 하시기 바랍니다.



[그림 ㉓-23] 실습해 볼 LVM 구성도

HDD의 파티션 정보

Device	Boot	Start	End	Blocks	Id	System
/dev/hda1	*	1	407	3076888 +	83	Linux
/dev/hda2		408	776	2789640	5	Extended
/dev/hda5		408	425	136048 +	82	Linux swap
/dev/hda6		426	696	2048728 +	8e	Linux LVM

Device	Boot	Start	End	Blocks	Id	System
/dev/hdb1		1	434	3281008 +	8e	Linux LVM
/dev/hdb2		435	776	2585520	8e	Linux LVM

③ HDD 준비 및 파티션 변경

fdisk를 실행하고 파일시스템 타입을 8e번 Linux LVM 으로 변경합니다.
w로 저장하고 빠져 나옵니다.

④ PV(physical volume)만들기

파티션 설정을 한 다음 PV를 만들어야 합니다. pvcreate를 사용해 다음과 같이 만들어 줍니다. lvm을 사용하고자 하는 파티션 모두를 다음과 같이 해주면 사용할 수 있습니다.

```
[root@loveme /root]# pvcreate /dev/hda6
pvcreate -- physical volume "/dev/hda6" successfully created
```

```
[root@loveme /root]# pvcreate /dev/hdb1
pvcreate -- physical volume "/dev/hdb1" successfully created
```

```
[root@loveme /root]# pvcreate /dev/hdb2
pvcreate -- physical volume "/dev/hdb2" successfully created
```

㉓ VG(volume group) 만들기

다음으로 볼륨 그룹을 만들어 줍니다. vgcreate 명령을 사용해 /dev/hdb1와 /dev/hdb2를 vg01 그룹으로 만듭니다. /dev/hda6 는 나중 사용을 위해 남겨둡니다.

```
[root@loveme /root]# vgcreate vg01 /dev/hdb1 /dev/hdb2
vgcreate -- INFO: using default physical extent size 4 MB
vgcreate -- INFO: maximum logical volume size is 255.99 Gigabyte
vgcreate -- doing automatic backup of volume group "vg01"
vgcreate -- volume group "vg01" successfully created and activated
```

볼륨 그룹 vg01에 대한 정보는 vgdisplay를 이용해 알아볼 수 있습니다.

vgdisplay를 실행하면 볼륨 그룹 이름은 vg01로 정의하고, 읽고 쓰기와 크기 변환을 할 수 있고, 현재 5.59GB를 사용할 수 있음을 알 수 있습니다.

```
[root@loveme /root]# vgdisplay
--- Volume group ---
VG Name vg01
VG Access read/write
VG Status available/resizable
VG # 0
MAX LV 256
Cur LV 0
Open LV 0
MAX LV Size 255.99 GB
Max PV 256
```



```
Cur PV 2
Act PV 2
VG Size 5.59 GB
PE Size 4 MB
Total PE 1431
Alloc PE / Size 0 / 0
Free PE / Size 1431 / 5.59 GB
VG UUID 9W0Mhr-pYVf-lloR-VRd8-RKXI-4M95-LSU3Q8
```

㉔ 로지컬 파티션의 생성

다음으로는 로지컬 파티션을 만들어야 합니다. 이곳에 마운트 포인터를 지정, 마운트해 데이터를 저장할 수 있습니다.

[lvcreate 사용법]

lvcreate <크기> <로지컬 볼륨 이름> <볼륨 그룹>

- L 옵션은 크기를 MB 단위로 입력
- l은 PE 단위로 입력
- n은 로지컬 볼륨 이름

```
[root@loveme /root]# lvcreate -L 1500M -nlvol1 vg01
lvcreate -- doing automatic backup of "vg01"
lvcreate -- logical volume "/dev/vg01/lvol1" successfully created

[root@loveme /root]# lvcreate -L 500M -n lvol2 vg01
lvcreate -- doing automatic backup of "vg01"
lvcreate -- logical volume "/dev/vg01/lvol2" successfully created

[root@loveme /root]# lvcreate -L 2000M -n lvol3 vg01
lvcreate -- doing automatic backup of "vg01"
lvcreate -- logical volume "/dev/vg01/lvol3" successfully created
```

정상적으로 만들어졌는지 알아보려면 lvscan을 사용하면 알 수 있습니다. lvol1에 1.46GB, lvol2에 500MB, lvol3에 1.95GB가 만들어졌음을 알 수 있습니다.

```
[root@loveme /root]# lvscan
lvscan -- ACTIVE "/dev/vg01/lvol1" [1.46 GB]
lvscan -- ACTIVE "/dev/vg01/lvol2" [500 MB]
lvscan -- ACTIVE "/dev/vg01/lvol3" [1.95 GB]
lvscan -- 3 logical volumes with 3.91 GB total in 1 volume group
lvscan -- 3 active logical volumes
```

㉕ 파일시스템 생성

로지컬 볼륨을 생성한 후 파일을 저장할 수 있도록 파일시스템을 만들어야 합니다. 로지컬 볼륨을 하나의 파티션이라 생각하면 됩니다. 기존의 파티션이랑 다른 점은 크기를 변경할 수 있다는 것입니다.

```
# mkfs /dev/vg01/lvol1
# mkfs /dev/vg01/lvol2
# mkfs /dev/vg01/lvol3

mke2fs 1.19, 13-Jul-2000 for EXT2 FS 0.5b, 95/08/09
Filesystem label =
OS type: Linux
Block size = 4096 (log = 2)
Fragment size = 4096 (log = 2)
256000 inodes, 512000 blocks
25600 blocks (5.00%) reserved for the super user
First data block = 0

16 block groups
32768 blocks per group, 32768 fragments per group
16000 inodes per group
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376, 294912

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

⑧ 디렉토리 만들기

기존의 꼭 찬 디렉토리나 여분의 디렉토리 등, lvm으로 바꿀 디렉토리를 다음과 같이 이름을 바꿔주고 이름을 바꾼 디렉토리를 다음과 같이 새로 만들어 줍니다.

```
[root@loveme /root]# mv /home/ /home.old
[root@loveme /root]# mv /var/ /var.old
[root@loveme /root]# mv /data/ /data.old
[root@loveme /root]# mkdir /home
[root@loveme /root]# mkdir /var
[root@loveme /root]# mkdir /data
```

⑨ LV(logical volume) 마운트 및 확인

새로 만든 디렉토리를 다음과 같이 마운트한 후 df 명령으로 확인합니다.

```
[root@loveme /root]# mount /dev/vg01/lvol1 /home
[root@loveme /root]# mount /dev/vg01/lvol2 /var
[root@loveme /root]# mount /dev/vg01/lvol3 /data
ot@loveme /root]# df
Filesystem 1k-blocks Used Available Use% Mounted on
/dev/hda1 3028552 2405092 469616 84% /
/dev/vg01/lvol1 1511856 20 1435036 1% /home
/dev/vg01/lvol2 495844 13 470231 1% /var
/dev/vg01/lvol3 2015824 20 1913404 1% /data
```

이상 없이 마운트 했으면 old로 바꾸어 놓은 내용을 다시 복사해 원상 복귀시킵니다.

```
[root@loveme /root]# cp -dpR /home.old/* /home
[root@loveme /root]# cp -dpR /var.old/* /var
[root@loveme /root]# cp -dpR /data.old/* /data
```

⑩ /etc/fstab 등록

부팅할 때마다 마운트를 할 필요없게 하려면 다음과 같이 /etc/fstab 에 아래 내용을 추가 해줍니다.

```
/dev/vg01/lvol1 /home ext2 defaults 1 2
/dev/vg01/lvol2 /data ext2 defaults 1 2
/dev/vg01/lvol3 /var ext2 defaults 1 2
```

이렇게 현재 시스템을 새로 포맷하지 않고 꼭 찬 디렉토리를 lvm으로 만들 수 있습니다.

⑪ 볼륨 그룹 바꾸기

볼륨 그룹을 변경하는 명령어는 vgrename입니다.

```
# vgrename vg01 loveme__vg
vgrename -- doing automatic backup of volume group "loveme__vg"
vgrename -- volume group "vg01" successfully renamed to "loveme__vg"
```

⑫ 기존 볼륨 그룹에 파티션 추가하기

기존에 있던 볼륨 그룹의 다른 하드 디스크에 있는 파티션을 추가할 때는 vgextend를 사용해, 하드 디스크나 파티션을 추가할 수 있습니다.

loveme_vg라는 볼륨그룹에 /dev/hda6의 비어 있는 lvm 파티션을 합쳐 보도록 하겠습니다. PV의 상태를 살펴보고 vgextend를 사용해 loveme_vg에 /dev/hda6을 다음과 같이 추가해 줍니다..

```
# pvscan
pvscan -- reading all physical volumes (this may take a while...)
pvscan -- inactive PV "/dev/hda6" is in no VG [1.95 GB]
pvscan -- ACTIVE PV "/dev/hdb1" of VG "vg01" [3.12 GB / 0 free]
pvscan -- ACTIVE PV "/dev/hdb2" of VG "vg01" [2.46 GB / 1.68 GB free]
pvscan -- total: 3 [7.55 GB] / in use: 2 [5.59 GB] / in no VG: 1 [1.95 GB]

[root@loveme /]# vgextend loveme__vg /dev/hda6
vgextend -- INFO: maximum logical volume size is 255.99 Gigabyte
vgextend -- doing automatic backup of volume group "loveme__vg"
vgextend -- volume group "loveme__vg" successfully extended
```

정상적으로 됐는지 확인하기 위해 vgdisplay을 실행해보면 같은 볼륨 그룹으로 돼 있는 것을 알 수 있습니다.

크기를 확인하면 다음과 같이 7.54GB로 확장됐음을 알 수 있습니다.

```
[root@loveme /]# vgdisplay loveme__vg
--- Volume group ---
VG Name loveme__vg
VG Access read/write
VG Status available/resizable
VG # 0
MAX LV 256
Cur LV 3
Open LV 3
MAX LV Size 255.99 GB
Max PV 256
Cur PV 3
Act PV 3
VG Size 7.54 GB
PE Size 4 MB
Total PE 1931
Alloc PE / Size 1000 / 3.91 GB
Free PE / Size 931 / 3.64 GB
VG UUID 9W0Mhr-pYVf-lIoR-VRd8-RKXI-4M95-LSU3Q8
```

⑬ 로지컬 볼륨(파티션) 사이즈 변경

로지컬 볼륨의 크기를 늘리는 방법으로는 다음과 같이 -L+100을 해주는 방법이 있습니다. 이렇게 하면 다음과 같이 기존의 크기에 100이 추가됩니다.

```
[root@loveme /]# lvextend -L +100 /dev/loveme__vg/lvol2
lvextend -- extending logical volume "/dev/loveme__vg/lvol2" to 600 MB
lvextend -- doing automatic backup of volume group "loveme__vg"
lvextend -- logical volume "/dev/loveme__vg/lvol2" successfully extended
```

다음과 같이 + 기호를 사용하지 않을 경우 해당 숫자의 크기만큼 증가합니다.

```
[root@loveme /]# lvextend -L900 /dev/loveme__vg/lvol2
lvextend -- extending logical volume "/dev/loveme__vg/lvol2" to 900 MB
lvextend -- doing automatic backup of volume group "loveme__vg"
lvextend -- logical volume "/dev/loveme__vg/lvol2" successfully extended
```

크기를 줄일 때는 lvreduce를 사용해 줄일 만큼의 숫자를 지정하면 그만큼 줄어들게 됩니다. 크기를 줄일 때는 기존의 데이터가 저장돼 있을 경우를 대비, 정말로 크기를 줄일 것인지를 확인하고 줄이게 되어 있습니다. 만약 데이터가 저장돼 있는 이상만큼 줄어들게 되면, 기존의 데이터는 당연히 삭제됩니다. 크기를 줄일 때는 주의해야 합니다.

```
[root@loveme /]# lvreduce -L-300 /dev/loveme__vg/lvol2
lvreduce -- WARNING: reducing active and open logical volume to 600 MB
lvreduce -- THIS MAY DESTROY YOUR DATA (filesystem etc.)
lvreduce -- do you really want to reduce "/dev/loveme__vg/lvol2"? [y/n]: y
lvreduce -- doing automatic backup of volume group "loveme__vg"
lvreduce -- logical volume "/dev/loveme__vg/lvol2" successfully reduced
```

⑭ 로지컬 볼륨(파티션) 이름 바꾸기

로지컬 볼륨의 이름은 다음과 같이 변경합니다.

```
[root@loveme /]# lvrename /dev/loveme__vg/lvol1 /dev/loveme__vg/home
lvrename -- doing automatic backup of volume group "loveme__vg"
lvrename -- logical volume "/dev/loveme__vg/lvol1" successfully renamed to
"/dev/loveme__vg/home"
```

확인해보면 다음과 같이 바뀐 것을 알 수 있습니다.

```
[root@loveme /]# lvscan
lvscan -- ACTIVE "/dev/loveme__vg/home" [1.46 GB]
lvscan -- ACTIVE "/dev/loveme__vg/lvol2" [600 MB]
lvscan -- ACTIVE "/dev/loveme__vg/lvol3" [1.95 GB]
lvscan -- 3 logical volumes with 4 GB total in 1 volume group
lvscan -- 3 active logical volumes
```

지금까지 LVM에 대해 간략히 살펴보았습니다.

더 자세한 것은 <http://linux.msede.com/lvm/>과 <http://linuxdoc.org>를 참고하면 알 수 있습니다.

6. APM(Apache + PHP + MySQL) 사용하기

(1) 개요

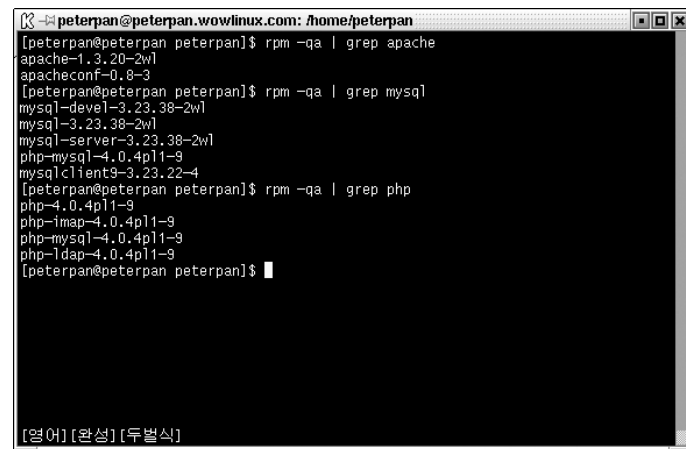
wowlinux 7.1 Paran를 설치하고 나면 Apache, PHP, MySQL은 자동으로 설치가 됩니다.

흔히 말하는 APM 연동을 위한 번거로운 작업을 다시 하실 필요가 없이 바로 강력한 APM의 성능을 사용할 수 있습니다.

wowlinux 7.1 Paran에서 APM이 어떻게 구성되어 있고 초기 설정에 대한 내용 및 주의사항을 알아보겠습니다.

(2) APM 구성

아래의 그림을 보면 APM에 관련된 모든 패키지들이 설치 되어 있는 것을 확인할 수 있을 겁니다.



```
[peterpan@peterpan wowlinux.com: /home/peterpan]
[peterpan@peterpan peterpan]$ rpm -qa | grep apache
apache-1.3.20-2w1
apacheconf-0.8-3
[peterpan@peterpan peterpan]$ rpm -qa | grep mysql
mysql-devel-3.23.38-2w1
mysql-3.23.38-2w1
mysql-server-3.23.38-2w1
php-mysql-4.0.4pl1-9
mysqlclient9-3.23.22-4
[peterpan@peterpan peterpan]$ rpm -qa | grep php
php-4.0.4pl1-9
php-imap-4.0.4pl1-9
php-mysql-4.0.4pl1-9
php-ldap-4.0.4pl1-9
[peterpan@peterpan peterpan]$
```

[그림 Ⅷ-24]

Paran에 설치되어 있는 Version은 아래와 같습니다.

Apache Version: apache-1.3.20 (<http://www.apache.org>)

MySQL Version: MySQL-3.23.38 (<http://www.mysql.com>)

PHP Version: 4.0.4pl1(<http://www.php.net>)

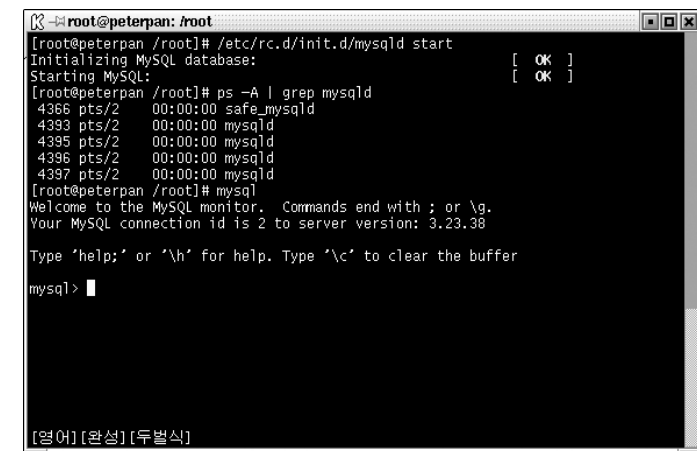
최신 Version에 대해서는 각 사이트에서 참조하시길 바랍니다.

(3) MySQL 설정 및 구동

Paran을 설치 하고 처음으로 MySQL를 실행하면 MySQL 초기에 생성되는 Database가 자동으로 설치가 됩니다.

[주의] MySQL의 초기화 값들은 /etc/rc.d/init.d/mysqld에 설정되어 있습니다.

반드시 처음에 MySQL를 초기화 하실 때 위의 스크립트를 제일 먼저 실행 시켜야 제대로 MySQL이 작동을 합니다.



```
[root@peterpan /root]# /etc/rc.d/init.d/mysqld start
Initializing MySQL database: [ OK ]
Starting MySQL: [ OK ]
[root@peterpan /root]# ps -A | grep mysqld
4366 pts/2 00:00:00 safe_mysqld
4393 pts/2 00:00:00 mysqld
4395 pts/2 00:00:00 mysqld
4396 pts/2 00:00:00 mysqld
4397 pts/2 00:00:00 mysqld
[root@peterpan /root]# mysql
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 2 to server version: 3.23.38

Type 'help;' or '\h' for help. Type '\c' to clear the buffer

mysql>
```

[그림 Ⅷ-25]

위의 그림과 같이 MySQL Databases와 MySQL Daemon이 실행되는 것을 볼 수 있습니다.

/etc/rc.d/init.d/mysqld start → MySQL 초기화 및 구동

ps -A | grep mysqld → MySQL이 정상적으로 구동 되었는지 ps 명령으로 확인

mysql → MySQL에 접속

앞으로는 MySQL을 다음과 같이 실행 시킬 수 있습니다.

■ MySQL Start / Stop

/etc/rc.d/init.d/mysqld start

/etc/rc.d/init.d/mysqld stop

■ MySQL을 실행할 때에는 Root 권한으로 실행을 시켜야 합니다.

- MySQL의 Data Directory는 /var/lib/mysql 입니다.
- MySQL의 설정 파일은 /etc/my.cnf 입니다.

[Tip] MySQL 환경설정 파일 수정

/usr/share/mysql/ 밑에 my-huge.cnf, my-large.cnf, my-medium.cnf, my-small.cnf File이 있습니다.
위의 File들은 MySQL 환경설정 File인 /etc/my.cnf의 시스템 사양에 따라 설정값들을 다르게 구성해
놓은 Sample File 입니다.

자신의 시스템에 가장 적합한 것을 선택하여 /etc/my.cnf File로 Copy을 한 뒤에 MySQL를 재시동하
시면 됩니다.

my-huge.cnf : System Memory가 1 2Gbyte 정도며 MySQL을 Main Server로 사용하는 경우

my-large.cnf : System Memory가 512Mbyte 이상이며 MySQL이 주용한 역할을 하는 경우

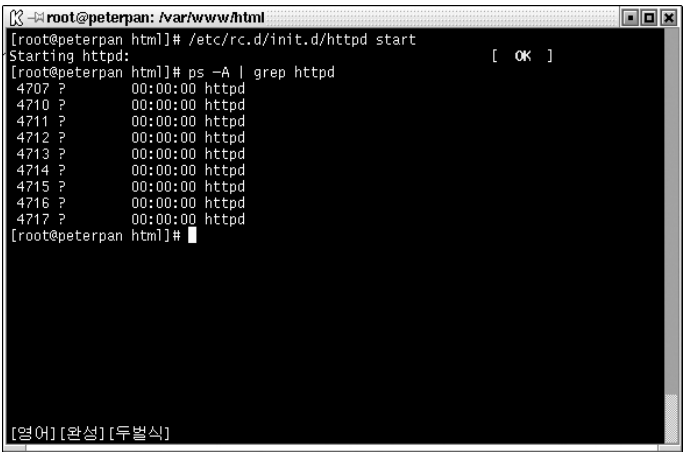
my-medium.cnf : System Memory가 32 64Mbyte 정도며 Web Server등의 다른 프로그램들과 함께
사용되는 경우

my-small.cnf : System Memory가 64Mbyte 이하에서 MySQL이 많은 자원들을 사용하지 않는 경우

- 설정 파일인 my.cnf를 수정한 뒤에는 반듯이 MySQL를 재가동시켜야 합니다.
바뀐 설정 값들을 확인하시려면 mysqladmin 명령으로 확인하실 수 있습니다.
mysqladmin variables

(4) Apache 설정 및 구동

- Apache를 처음 구동 시키는 방법은 아래의 그림과 같습니다.



[그림7-26]

/etc/rc.d/init.d/httpd start → Apache 실행

ps -A | grep httpd → Apache가 정상적으로 구동 되었는지 ps 명령으로 확인
/etc/rc.d/init.d/httpd stop → Apache 정지

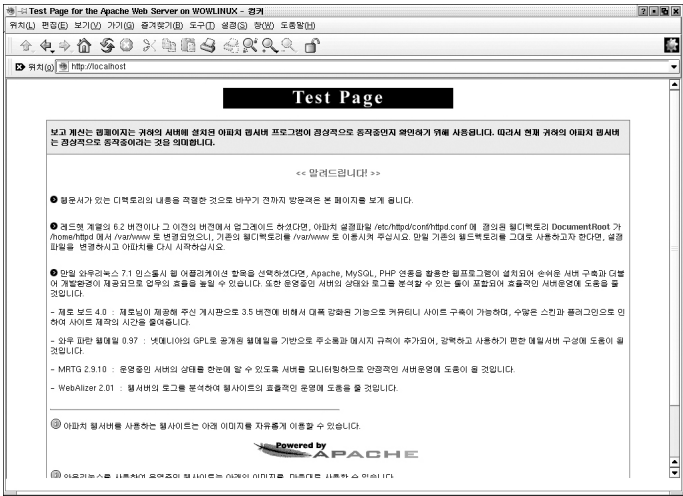
- Apache를 실행할 때에는 Root 권한으로 실행을 시켜야 합니다.

- Apache 설정파일인 /etc/httpd/conf/http.conf

[참고] Paran에 설치되어 있는 Apache와 관련해서 유용한 Package들이 포함되어 있습니다.

- zero board 4.0
- wow Paran webmail 0.97
- MRTG 2.9.10
- WebAlizer 2.01

- 정상적으로 Apache가 실행되었다면 아래와 같은 Apache의 Test Page를 볼 수 있습니다.



[그림 Ⅷ -27]

Paran에서의 Apache 초기 설정 값들은 아래와 같습니다.

[주의] Paran에서의 초기 설정 값들은 다른 Apache 설정과는 다를 수 있습니다.
Paran에서 Apache를 처음 설치할 때 적용되는 값들입니다.

- Apache Root → /etc/httpd
- Apache 설정파일 → /etc/httpd/conf/httpd.conf
- Apache Home Directory → /var/www/html
- Apache Log Directory → /var/log/httpd

자세한 설정 내용들은 /etc/httpd/conf/httpd.conf를 참조하시길 바랍니다. 또한 Apache 설정에 대한 자세한 내용들은 <http://www.apache.org>, <http://www.apache.kr.net>을 참조해 주시길 바랍니다.

(5) PHP (PHP : Hypertext Preprocessor)

웹 프로그래밍 개발 언어로 많이 사용되고 있는 언어중 한가지로 현재 많은 사이트들이 PHP을 이용해 서 개발을 하고 있습니다.

처음 PHP를 사용하려는 많은 사람들을 위하여 또는 웹 프로그래밍을 배우려는 학생 및 사람, PHP 개발을 처음 하는 개발자들을 위해서 Paran에서는 기본적으로 모든 개발환경을 설정해 놓았습니다.

Paran에서의 PHP 설정과 구성 요소들을 알아 보겠습니다.

[주의] Paran에서의 PHP 초기 설정 값들은 다른 PHP의 설정과는 다를 수 있습니다.

Paran에서 PHP 설치 옵션은 아래와 같이 많은 옵션들을 미리 설정하여 설치를 하였습니다. 가급적이면 PHP에서 사용할 수 있는 모든 환경설정을 해두어 사용시 불편함이 없도록 하였습니다.

사용자가 필요한 설정값들만 사용을 원하신다면 Paran에 설치되어 있는 PHP를 삭제하시고 다시 설치를 하셔야 합니다.

■ Paran에서의 PHP 설치 옵션값

```
./configure --prefix=/usr --with-config-file-path=/etc --disable-debug --enable-pic --enable-shared --enable-inline-optimization --with-apxs=/usr/sbin/apxs --with-exec-dir=/usr/bin --with-regex=system --with-gettext --with-gd --with-jpeg-dir=/usr --with-png --with-zlib --with-db2 --with-db3 --with-gdbm --enable-debugger --enable-magic-quotes --enable-safe-mode --enable-sockets --enable-sysvsem --enable-sysvshm --enable-track-vars --enable-yp --enable-ftp --enable-wddx --without-mysql --without-oracle --without-oci8 --with-xml
```

■ 설치 옵션에서 확인하실 수 있겠지만 PHP의 설치 위치와 설정파일의 위치는 아래와 같습니다.

- PHP Base Directory : /usr
- PHP config file : /etc/php.ini

각 옵션에 대한 자세한 내용은 <http://www.php.net>에서 참조하시길 바랍니다. 또한 <http://www.phpschool.com>에서도 자세한 정보를 얻을 수 있습니다.

■ PHP가 제대로 동작을 하는지 간단한 스크립트를 이용하여 확인을 해보겠습니다.

test.php라는 파일을 아래와 같은 경로에 만들어 둡니다.

→ /var/www/html/test.php

test.php 파일에 phpinfo() 함수를 호출하도록 스크립트를 적어 둡니다.

```
<?
    phpinfo();
?>
```

■ 이제 test.php 파일을 Brower를 통해 결과를 확인합니다.



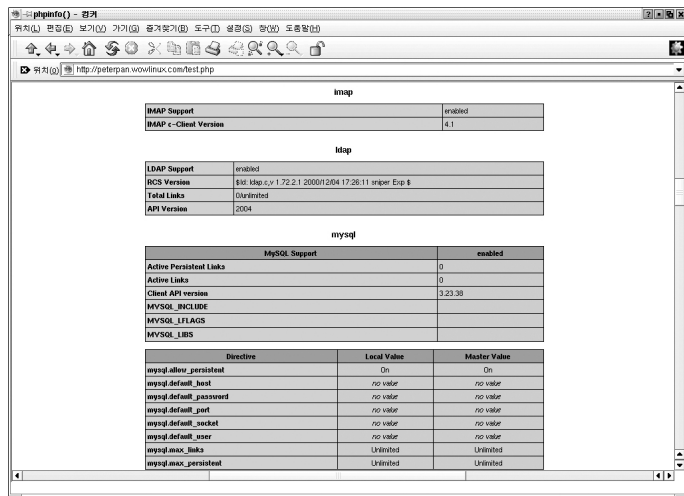
[그림 Ⅶ -28]

■ 위의 그림처럼 PHP의 기본 정보를 Brower에서 확인할 수 있습니다.

■ PHP에서 지원하는 기능에 대한 정보가 자세히 화면에 출력이 됩니다.

■ 기타 PHP 설정 정보를 알려주는 것이므로 자세히 확인하시길 바랍니다.

■ 아래 그림에서 MySQL과 PHP가 제대로 연동되어 있는지 볼 수 있습니다.



[그림 Ⅷ -29]

- Paran의 PHP에서는 php4, php3, phtml, php, html 확장자를 가진 파일들만이 PHP Script로 인식됩니다. 만약 inc, htm등의 다른 확장자를 더 추가하여 사용하고 싶다면 Apache 설정 파일인 httpd.conf에 아래와 같이 파일 타입을 추가로 정의하면 됩니다.

```
Addtype application/x-http-php .inc
Addtype application/x-http-php .htm
```

만약 여러 줄에 적는 것이 귀찮다면 다음과 같이 기존에 설정해 놓은 곳에 다음과 같이 적어 두면 됩니다.

```
<IfModule mod_php4.c>
    Addtype application/x-http-php .php4 .php3 .phtml .php .html .inc
    Addtype application/x-http-php-source .phps
</IfModule>
```

7. MRTG(Multi Router Traffic Grapher) 사용하기

(1) 개요

MRTG는 SNMP를 지원하는 Network 장비가 발생하는 Traffic을 Monitoring 해 주는 소스가 공개된 프로그램입니다.

지정한 시간(Default는 5분)마다 Monitoring 한 결과값을 GIF(또는 PNG)이미지로 생성하여 HTML 페이지로 생성해 주기 때문에 누구나 쉽게 Traffic 현황을 볼 수 있습니다

Paran 설치한 Server는 기본적으로 MRTG를 지원하고 있으며 간단하게 초기 세팅만 해주면 MRTG를 사용할 수 있습니다.

Paran에 설치되어 있는 MRTG는 SNMP를 이용하여 사용하는 것이 아니라 Perl Script를 이용하여 System을 Monitoring을 하는 방법입니다.

SNMP를 이용하여 다른 Network 장비등을 Monitoring을 하고자 한다면 다음 사이트들을 참조하시기 바랍니다.

→ <http://people.ee.ethz.ch/~oetiker/webtool/mrtg>
 → <http://www.mrtg.co.kr>

(2) MRTG 설치 및 사용

- mrtg가 설치가 되어 있는지 확인합니다.

```
[root@localhost /root]# rpm -qa | grep mrtg
mrtg-2.9.10-5w1
```

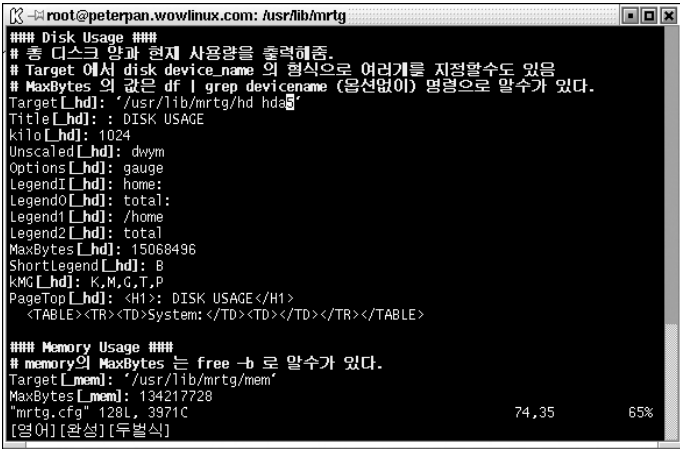
- Paran에 설치되어 있는 mrtg Version은 2.9.10 입니다.

- mrtg 설치 구성은 아래와 같습니다.

→ Base Directory : /usr/lib/mrtg
 → Config File : /usr/lib/mrtg/mrtg.cfg
 → HTML 생성기 : /usr/lib/mrtg/indexmaker
 → mrtg shell script : /usr/bin/mrtg.sh

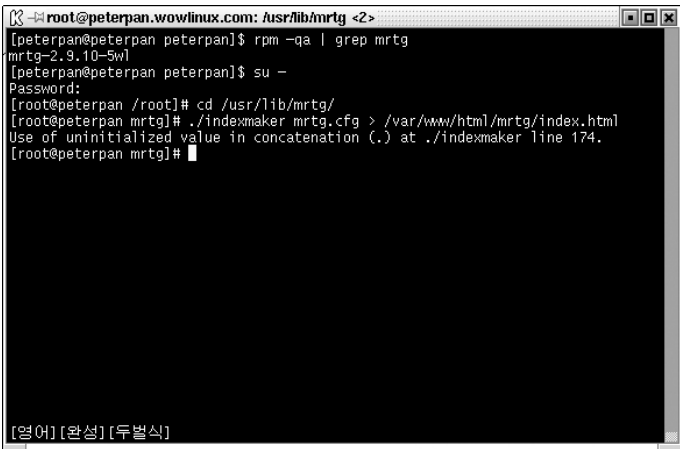
- mrtg가 기본으로 설치되어 있는 Directory는 /usr/lib/mrtg 입니다.
Paran의 mrtg에서는 Server의 HardDisk의 사용량을 주기적으로 점검 하도록 설정이 되어 있습니다.
그러므로 모니터링 하고자 하는 Device Name을 반드시 자신의 System에 맞게 수정을 해야 합니다.

- 아래의 그림처럼 /usr/lib/mrtg/mrtg.cfg File에 HardDisk 설정 부분에 Monitoring 하고자 하는 Device Name을 수정합니다.



[그림 Ⅶ -30]

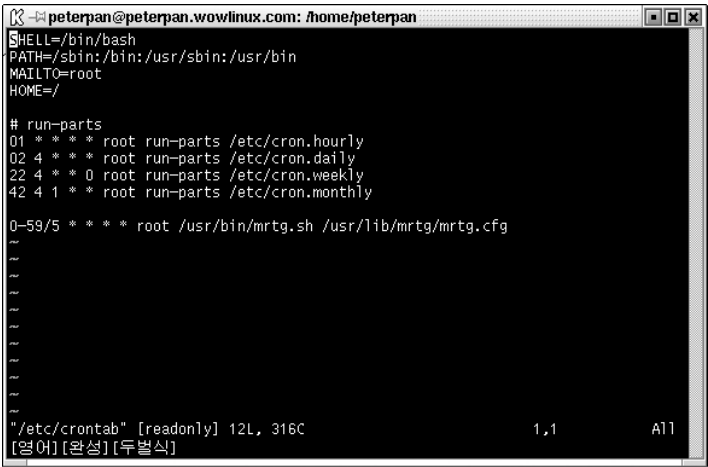
- 아래의 그림처럼 indexmaker를 이용하여 mrtg가 System을 Monitoring한 결과를 HTML File로 만들어 줍니다.



[그림 Ⅶ -31]

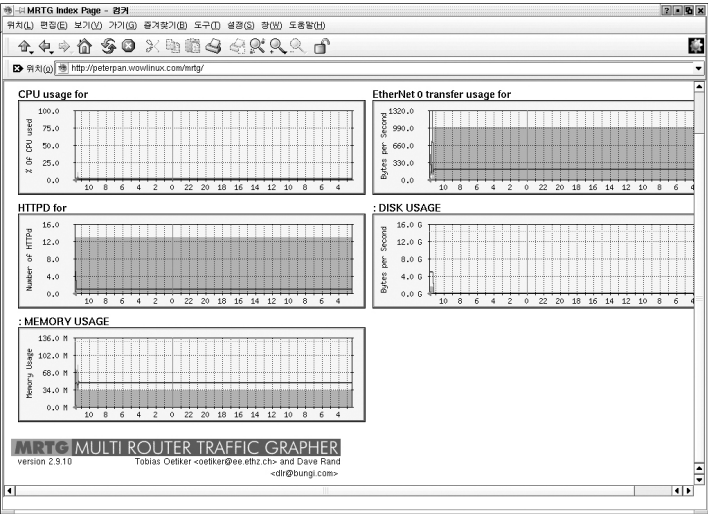
- 만약 위의 그림처럼 처음 indexmaker를 실행할 때 error가 나타난다면 무시하고 넘어가시길 바랍니다.

- /etc/crontab을 확인하시면 mrtg shell script가 5분마다 mrtg를 갱신하는 것을 보실 수 있습니다.



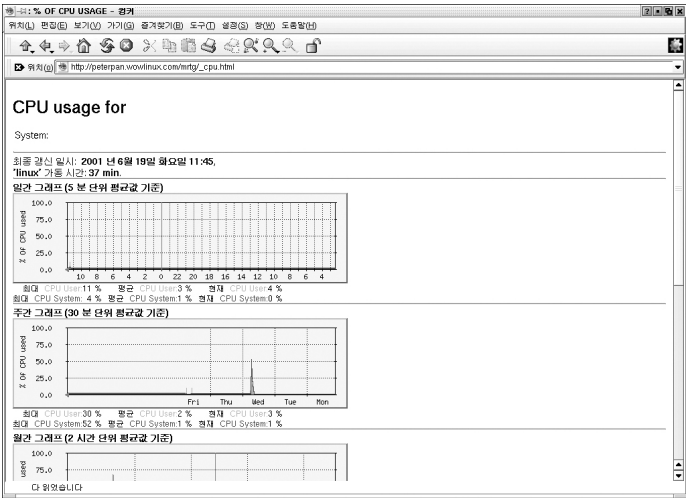
[그림 Ⅶ -32]

- 모든 작업이 끝났다면 mrtg가 제대로 수행하고 있는지 확인을 해봅니다.
(결과 화면 : <http://localhost/mrtg>)



[그림 Ⅶ -33]

■ Paran에서의 mrtg는 CPU, Ethernet, HTTPD, DISK, MEMORY 만을 Monitoring 합니다. mrtg 초기 화면에서 각 Monitoring 이미지를 누르면 아래와 같이 상세 정보를 확인할 수 있습니다.



[그림Ⅶ-34]