

# 1장 리눅스 시스템 명령어, 관리

## 1. 리눅스 시스템 디렉토리

- / : 루트 디렉토리로 서브디렉토리를 가지며, 리눅스 프로그램을 가진다.
- /home : 사용자의 홈디렉토리를 가진다.
- /bin : 기본 리눅스 프로그램에서 많이 실행되는 실행파일들이 들어 있다.
- /usr : 대부분의 배포판 소프트웨어가 설치되는 중요한 파티션.  
사용자 지향적인 디렉토리.
- /usr/bin : 사용자 지향적인 실행파일들을 가진다.
- /usr/sbin : 파일 관리 시스템을 가진다.
- /usr/local : 사용자가 스스로 컴파일하여 설치하는 프로그램은 관례상 이 디렉토리에 설치된다.  
Ex) /usr/local/apache, /usr/local/mysql
- /dev : 리눅스에서 사용하는 각 장치에 대한 Device 들이 위치하는 디렉토리. 가상디렉토리.
- /sbin : 자동적으로 사용되는 파일 시스템을 가진다.(주로 관리자용 명령어)
- /etc : 리눅스 시스템에서 사용되는 어플리케이션 및 리눅스 환경 설정 파일들이 있는 디렉토리. 리눅스 구성파일을 가지고 있다.
- /tmp : 임시 파일들이 생성되는 디렉토리.
- /boot : 커널 및 grub 관련 파일, 부팅에 관련된 파일.
- /lib : 필수적인 라이브러리.
- /proc : 프로세스, 시스템 정보를 위한 가상 디렉토리.
- /var : 시스템 운영중 생성되는 각종 로그, mail 등의 임시 보관 디렉토리
- /mnt : cd-rom, floppy 장치에 대한 pointer가 위치하는 디렉토리
- /root : system user의 홈디렉토리

## 2. 파일 / 디렉토리 관리 명령어

- shutdown : 시스템 종료 및 재부팅
  - shutdown -h now : 바로 종료
  - shutdown -r now : 시스템 재부팅
  - shutdown +3 "system is going down" : 로그인한 사용자에게 시스템 종료 메시지 뿌려주고 3분후 시스템 종료.

- man, help, whatis : 도움말, 매뉴얼 출력
  - whatis : makewhatis 로 db 생성후 사용

```
]# man ls
]# ls --help
```

- ls : 현재 디렉토리 내에 있는 파일 / 디렉토리 출력
  - (-l : 자세히 나열(파일정보-파일종류,퍼미션,i-node,소유자,소유그룹, 최종 변경날짜 표시),
  - a : 디렉토리 내의 모든 파일 출력)

drwxr-xr-x. 4 skykim skykim 4096 2014-08-13 01:23 sample							
↑	↑	↑	↑	↑	↑	↑	↑
파일종류	접근권한	링크	소유자	소유그룹	파일크기	최종변경날짜,시간	파일이름

. : 점(.) 한 개는 현재 디렉토리 의미  
 .. : 점(..) 두 개는 상위 디렉토리로 이동할 수 있음을 의미  
 .bash\_history : 입력한 명령어 내용이 저장되는 파일.  
 .bash\_profile : 도스의 autoexec.bat 파일과 유사한 기능을 가진 파일.  
                   로그인시 로그인 셸에 의해 불러워지는 환경설정파일  
 .bashrc : 알리아스(alias) 정보를 가지고 있는 파일

<참고>  
 i-node : 시스템에서 파일을 인식할 때 사용하는 고유 번호,  
           리눅스에서 모든 파일들은 고유의 i-node 번호를 가지고 있다.  
           같은 파티션 내에서만 같은 i-node 정보를 가진다.

<참고>  
 리눅스에서는 모든 것을 파일로 처리한다.  
 텍스트, 이미지, 영상등의 자료, 디렉토리 정보, 네트워크 소켓, 시스템 장치 등을 파일로 처리한다. 그래서 그래픽 환경이 아니더라도 텍스트 환경에서도 시스템을 제어할 수 있다.

- more : 파일의 내용을 화면 단위로 출력

- 한줄씩 출력 : Enter 키
- 한 화면씩 출력 : 스페이스 바
- 이전 화면 출력 : b
- 도움말 : h
- 종료 : q

옵션	설명
-d	[Press space to continue, 'q' to quit.] 메시지 보여준다
-p	스크롤 되지 않고 모든 화면 지우고 메시지 보여준다.
-s	여러줄의 빈 줄을 한줄로 보여준다.
-라인	화면에 출력될 수 지정
+라인	지정한 줄부터 출력

→ less : 파일의 내용 출력

→ sort : 텍스트 파일의 내용을 정렬시키는 명령어

```

]# sort /etc/inittab
]# sort -r /etc/inittab
]# ls -al | sort -r | more

```

→ head, tail : 파일의 앞(뒤) 부분 1page 정도를 보여준다.

```
- head(tail) -10 filename
```

→ pwd : 현재 디렉토리의 위치 출력, 절대 경로로 표시

→ cd : 디렉토리 변경

- . : 현재 디렉토리, .. : 부모디렉토리
- / : 최상위 디렉토리를 의미
- cd 에 문자를 붙이지 않으면 자신이 login 한 디렉토리로 이동
- ~ : 자신의 계정디렉토리로 이동한다.
- \$HOME : 변수에 지정된 자신의 디렉토리로 이동

→ cat : 임의의 파일 생성 및 파일 보기

```
Ex1) cat > test.html
```

내용

Ctrl + D : 저장하고 빠져나옴.

cat test.sh test3.sh var > test : test.sh, test3.sh, var 파일의 내용을 test 파일 복사.

cat cat.txt >> test : cat.txt 파일 내용을 test 파일에 추가

Ex2) man cat > cat.txt : cat 의 man 페이지를 cat.txt 파일에 저장

<참고> 리다이렉션

- < : 표준입력 (Std In)
- > : 표준출력(Std Out)
- 2> : 표준 에러 (Std Err)

→ touch : 파일사이즈가 '0' 인 파일 생성

→ mkdir : 디렉토리 생성

```
-p : 하위 디렉토리까지 생성
```

```
]# mkdir test/test1/test2/test3 //에러 발생
```

mkdir: 'test/test1/test2/test3' 디렉토리를 만들 수 없습니다: 그런 파일이나 디렉터리가 없습니다.

```
]# mkdir -p test/test1/test2/test3 // 디렉토리 생성
```

→ rmdir : 디렉토리 삭제

```
-p : 지정한 하위디렉토리까지 한번에 삭제
```

```
]# rmdir -p test/test1/test2/test3
```

→ cp : 파일 복사

```
-p : 파일의 내용뿐 아니라 속성도 그대로 복사
```

- r, -R : 서브 디렉토리, 즉 디렉토리 전체를 복사.
- a : 원본파일의 속성, 링크 정보들을 유지하면서 복사
- f : 복사할 파일이 존재할 때 삭제하고 복사

```
cp -f /dev/null /var/log/message → 파일 크기를 0 으로 만들 수 있다.
```

→ mv : 파일 이름변경, 이동

→ rm : 파일, 디렉토리 삭제  
 -f : 삭제 확인을 묻지 않고 바로 삭제.  
 -r : 디렉토리 삭제시 사용.

→ ln : 다른 파일 참조

- ln -F filename(경로) linkname : 하드링크
- ln -s filename(경로) linkname : 소프트링크, 원본파일 삭제시 링크파일 쓸모 없음.

→ find : 파일 찾기

```
Ex) find / -name perl
find /bin -name s* // * 사용시 ' ' 로 묶어주지 않으면 error 발생하는 경우 있음,
// find: paths must precede expression:
find /bin -name '[a,s][b,e]*' -print // 첫글자가 a 또는 b로 시작하고, 두번째 글자가 b 또는 e로 된파일
find /bin -name '[s-z]????' -print // s~z 로 시작하는 5자로 된 파일 출력
find /home -name *.bak -exec rm -f {} \;
find /home -perm 777 -print
find /home -empty -print
find / -size +102400k -print
find /home -user skykim -print
find /home -type d -print // 일반파일 :f , 디렉토리 : d , 링크파일 : l , 블록/문자디바이스 : b/c
find /home/skykim -mtime +7 -exec rm -f {} \; // 수정된지 7일이 지난 파일 삭제
```

- name filename : 가장 일반적으로 사용하는 옵션.
- perm mode : 주어진 액세스 모드를 가진 파일을 찾는다.
- type c : 지정된 타입에 관해서 파일을 찾는다.
- user name : name에 해당하는 사용자파일을 찾는다. Name 에는 사용자 id 뿐만아니라 UID 도 들어갈수 있다.
- group group명 : 그룹이 그룹인명.
- size n : 해당되는 사이즈의 파일을 찾는다. n 은 블록길이의 파일을 찾는다. 한 블록은 512와 같다. +n 이란 표시가 들어가면 n 블록보다 더 긴파일을 찾는다라는 뜻.
- atime n : n일전데 액세스한 파일들을 찾는다.
- mtime n : n일전데 마지막으로 파일내용이 수정된 파일들.
- ctime n : n일 전에 마지막으로 속성이 바뀌어진 파일들
- newer file : 주어진 file 보다 더 최근에 수정된 파일을 찾는다.
- exec command : 찾은 파일들을 command 로 처리한다. command에서 찾은 파일의 경로명을 포함시키려 면 중괄호를 사용한다. {} command 명령을 실행시키고 난 뒤에는 반드시 백슬래시와 세미콜론을 사용한다. (W;)

→ chmod : 파일 허가권, 파일 디렉토리 접근 권한 변경 명령어.

```
drwxrwxrwx 3 root root 4096 Jun 18 16:06 dir2
-R 옵션 : 변경 대상이 디렉토리 인 경우 하위의 모든 디렉토리, 파일 변경.
```

Ex) chmod -R 755 \*

읽기 (r)	쓰기 (w)	실행(x)	rwX
4	2	1	7

- 파일에 모든 사용자에게 읽기 권한만 부여하고 싶다면, 접근 권한은 r-r-r-로 수정해야 한다.

```
]# chmod 444 파일명
```

- 권한 기호 : u(소유자), g(그룹), o(타인), a(소유자,그룹,타인 모두)

```
chmod g+w 파일명
chmod g-wx+r 파일명
chmod o+rx 파일명
chmod o-rwx 파일명
chmod g-wx+r 파일명
chmod ugo+rx 파일명
chmod +x 파일명
chmod go=rw 파일명
chmod u+x,go=rw 파일명
chmod a=rw 파일명
```

```
chmod g-w 파일명
chmod g+w,o-x 파일명
```

-SetUID : 실행시 root의 권한을 잠시 가질 수 있다.

4 : SetUID                chmod 4755 file

2: SetGID                chmod 2755 file

→ SetGid, SetUID 는 파일을 실행할 때 그 파일의 소유자 또는 그룹의 권한으로 실행되도록하는 것이다. 루트 권한이 필요한 경우또는 시스템 자원을 이용하기 위한 경우에 필요.

```
[root@ns bin]# ls -al ping
-rwsr-xr-x 1 root root 35832  9월 27  2009 ping
```

```
[root@ns bin]# chmod 755 /bin/ping
[skykim@ns ~]# ping 168.126.63.1
ping: icmp open socket: Operation not permitted
```

1 : Sticky bit , other에 해당되는 사용자는 쓰기권한이 있어 파일을 만들수는 있지만 자기 파일만 지울 수 있다.즉, 모든 사용자가 쓰고 삭제할 수 있는 디렉토리에 적용. (/tmp , /var/tmp)

→ chown : 파일 및 디렉토리 소유권한 변경

-R 옵션 : 하위디렉토리나 디렉토리 하위의 모든 파일의 소유권 변경

-h 옵션 : 심볼릭 링크 자체의 소유권 변경

→ chgrp : 그룹 소유권 변경

-R 옵션, -h 옵션

#### \* umask

파일 또는 디렉토리 생성시에 기본적으로 설정되는 사용권한 값.

Umask 로 지정한 값은 기본값에서 제거

Umask 값 확인 : /etc/profile 에 umask 지정.

```
]# umask -S
u=rwx,g=rx,o=rx
```

```
]$ umask
0002
```

일반파일최대값 666(rw-rw-rw) - UMASK(022) = 디렉토리 기본값(755)

디렉토리 최대값 777(rwxrwxrwx) - UMASK(022) = 파일 기본값(755)

→ df : 하드디스크 사용 상태 확인

df -k : 단위 KB

df -m : 단위 MB

df -h : 단위 직접 표시

df -a : 모든 파일시스템 확인

df -T : 파일 시스템 확인

→ du : 지정한 디렉토리 사용량을 표시해준다.

-a(--all) : 디렉토리가 아닌 모든 파일에 대한 정보 출력

-h : 사용자가 이해하기 쉬운 용량의 단위 표시(KB, MB, GB)

-l : 하드링크의 용량까지 모두 계산

-s : 사용량의 총합계 출력

-S : 하위디렉토리 표시하지 않고 각각 나누어서 출력

```
]# du -sh
```

```
]# du -a
```

```
]# du -max-depth=1 -h //현재 위치에서 지정한 depth 까지의 용량을 표시할 경우 사용
```

→ mount : 새로운 시스템을 추가하기 위해 사용. 즉, 새로운 시스템을 디렉토리로 연결하여 사용.

- mount -t iso9660 -o ro /dev/cdrom /media

: ro (read-only)

: -t : 파일시스템 지정

- umount /media

Umount /media

- mount -t vfat /dev/hdc /win : 윈도우에서 사용하던 디스크를 리눅스에서 마운트하여 윈도우에서 사용하

던 하드 디스크 사용.

→ free : 메모리 상태 출력  
-b, -k, -m, -g

→ cron : 특정 시간에 명령어 실행 시켜 주는 데몬, 스케줄링  
/etc/crontab 파일과 관련 디렉토리(cron.d/ , cron.daily/ , cron.hourly/ , cron.monthly/ ,cron.weekly/)

```
]# crontab -e //수정
]# crontab -l //보기
]# crontab -r //삭제
```

분 : 0-59 , 시 : 0-23, 일 : 0-31, 월 : 1-12, 요일 : 0-7( 0,7은 일요일)  
Ex) 10 9 \* \* \* shutdown -r now  
\* 0-23/2 \* \* \* : 매 두시간마다.  
30 2 \* \* \* /root/bin/backup.sh : 매요일 매월 매일 2시 30분에 backup.sh 실행  
\*/05 \* \* \* \* : 5분마다

```
<cron 프로세스>
]# service crond stop [start | restart | status]
]# /etc/rc.d/init.d/crond stop [ start | restart | status ]
]# /etc/init.d/crond stop [ start | restart | status]
```

→ grep : 문자열 패턴 검색  
-r : 서브 디렉토리 포함하여 검색.  
-c : 문자열을 포함하는 행의 수 출력  
-n : 일치되는 행 번호 출력  
-v : 일치되지 않는 행만 출력  
-w : 문자열이 하나의 단어인 경우만 검색

Ex) grep root /etc/passwd // 'root'라는 문자열을 포함하는 행을 보여준다.  
grep -c root /etc/passwd // 'root' 라는 문자열이 포함되어 있는 행의 수를 보여준다.  
Grep -v root /etc/passwd //문자열과 일치하지 않는 행들 보여준다.

Ex) grep -n bind zshrc | less  
→ zshrc 내용중 'bind' 라는 단어를 포함하고 있는 라인을 파이프라인과 less 명령어를 이용하여 한페이지씩 출력.

Ex) grep -i ls ls.txt //대소문자 구분 없이 문자 검색

EX) ls -al /etc/ | grep init // 다른 명령어와 조합해서 사용

→ egrep : grep 확장된 정규표현식이나 메타 문자 사용  
EX) egrep 'root|bash' /etc/passwd // root 혹은 bash  
Egrep init+ /etc/inittab //init 문자가 한번 또는 이상 대응되는 행  
Egrep 's[h|u]' 파일명

→ 리다이렉션 : 명령의 출력을 변경할 수 있다. 기본적으로 명령어 결과는 화면에 출력되지만, 리다이렉션을 이용하면 파일에 기록.  
: 연산자 >, >>, <  
: 명령어 > 파일 - 파일이 없다면 생성하고, 있다면 기존의 내용을 지운다.  
: 명령어 >>파일 - 파일이 없다면 생성하고, 있다면 기존의 내용에 추가한다  
:명령어 < 파일 - 파일에서 표준입력 받는다.

```
[root@ns ~]# echo hello
hello
```

- 파일에 저장

```
[root@ns ~]# echo hell > hello.txt
[root@ns ~]# more hello.txt
hell
```

```
[root@ns ~]# echo hello >>hello.txt
[root@ns ~]# more hello.txt
hell
hello
```

### 3. 시스템 / 프로세스 명령어

- top : 실행 중인 프로그램 자원 할당 상태, 약 5초마다 갱신되며, cpu를 가장 많이 점유하는 프로세스부터 출력
  - q : 종료
  - cpu 사용률이 많은 프로세스 정렬  
: top 실행화면에서 “SHIFT+P”
  - 메모리 사용률이 높은 프로세스 정렬  
: SHIFT+ M
  - 실행시간이 오래된 프로세스 정렬  
: “SHIFT+T
  - idle 프로세스들을 감추고 간단히 확인  
: top 실행후 “i” 입력, i 상태에서 한번 더 누르면 원상태 복귀. 즉 running 상태의 프로세스들만 확인
  - 특정 사용자가 실행한 프로세스만 확인  
: top을 실행후 “u “ 입력, “u”를 입력한 후에 특정사용자 id 입력
- 첫번째줄 : 현재시간, 시스템 가동시간, 현재로그인수, 평균부하율(1분, 5분, 15분간)
- 두번째줄 : 프로세스 정보
- 세번째줄 : cpu 사용률(사용자, 시스템, NICE, 사용되지 않는 CPU, 입출력 대기 상태, 하드웨어 인터럽트, 소프트웨어 인터럽트 cpu 사용률, 가상화환경에서 cpu 손실율)
- 네번째줄 : memory 사용률
- 세부정보 필드 :
  - PID : 프로세스 ID
  - USER : 프로세스 소유자
  - PR : 프로세스 우선순위
  - NI : 우선순위를 지정하는 nice 값, -20~19 , - 가 우선순위 높음
  - VIRT : 프로세스가 점유하고 있는 가상메모리양(KB 단위)
  - RES : 스왑되지않는 물리메모리양 (KB 단위), 즉 현재페이지가 상주하고 있는 크기
  - SHR : 공유메모리양(KB 단위)
  - S : 프로세스 상태
  - %CPU : cpu 점유율
  - %MEM : memory 점유율
  - TIME+ : 프로세스가 cpu 를 사용한 시간,
  - COMMAND : 프로세스를 실행한 명령

→ ps : 현재 돌아가는 프로세스 보여줌.

```
[root@www1 ~]# ps
  PID TTY          TIME CMD
 29763 pts/0    00:00:00 su
 29764 pts/0    00:00:00 bash
 29796 pts/0    00:00:00 ps
```

PID : Process ID(작업번호)  
 TTY : 연결된 제어 터미널  
 TIME : 연결된 시간  
 CMD : command , 실행된 명령어

→ ps aux :

- a : 모든 프로세스 출력
- u : 사용자와 관련된 프로세스 출력, cpu, 메모리 점유율에 대한 정보도 출력.
- x : 제어터미널이 없는 프로세스도 출력
- r : 현재 수행중인 프로세스만 출력

```
[root@www1 ~]# ps uax
USER  PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root    1  0.0  0.0  2456  552 ?        S    2008   0:00 init [3]
```

USER : 사용자 계정, PID : Process ID(작업번호)  
 %CPU : CPU점유율, %MEM : 메모리점유율, VSZ : 가상 메모리양  
 RSS : 프로세스가 사용한 실제 메모리양  
 TTY : 연결된 제어터미널, ? 로 표시되는 프로세스들은 tty 와 상관없다.  
 즉, 사용자가 올린 프로세스가 아닌 리눅스가 올린 프로세스이다.  
 콘솔 로그인시에는 tty1 ~tty6 까지, 원격로그인시에는 pts/번호  
 STAT : 프로세스 상태( R(Runnable) : 실행중 또는 실행 대기인 프로세스  
 S(sleeping) : 수면상태  
 D(in Disk wait) : 입출력을 기다리는 상태

T(sTopped) : 정지되어 있는 상태(suspend)  
Z(Zombie) : 좀비 프로세스

START : 시작시간, TIME : 사용시간, COMMAND : 명령어

Ex) ps aux | grep httpd ( grep 를 이용해서 원하는 프로세스의 정보를 볼수 있다.)

<프로세스>

프로세스란 실행 중인 프로그램을 나타내는 개념이다.

리눅스 및 유닉스에서 프로세스들은 계층적 구조로 되어있으며, 각각의 프로세스들은 자신을 만들어준 부모프로세스를 가진다. 즉, 부모 프로세스에 의해 만들어진 프로세스를 자식프로세스라고 한다.

자식 프로세스는 부모 프로세스로부터 속성을 상속받는데, 자신의 프로세스 ID 인 PID를 가지며 부모 프로세스의 ID 인 PPID 를 갖고 있다.

부모 프로세스 까지 출력 : ps -ef

]# ps -ef | grep udev

```
root      497      1  0 14:56 ?          00:00:00 /sbin/udevd -d
root     1403    497  0 14:56 ?          00:00:00 /sbin/udevd -d
root     1404    497  0 14:56 ?          00:00:00 /sbin/udevd -d
```

프로세스는 포그라운드나 백그라운드 프로세스로 실행된다.

특정 작업을 장시간 오랫동안 실행해야 한다면 작업을 백그라운드로 처리하고 다른 작업을 할수 있다.

백그라운드 처리는 앰퍼센드(&) 기호를 붙여 실행한다.

]# top &

[1] 3367

]# jobs // 백그라운드에서 실행되고 있는 작업확인. + : 실행중 , - : 실행될 작업

[1]+ Stopped

top

]# fg %1 // 숫자 1은 작업번호, 백그라운드에서 포그라운드로 전환,  
포그라운드 전환 후 작업 일시 중단 : Ctrl + z

]# bg %1 // 다시 백그라운드로 전환

]# kill %1 // 작업 종료

[1]+ Stopped

top

]#

[1]+ Exit 143

top

]# jobs // 종료 확인

]#

→ kill : 프로세스 중지

- kill -9 30080 (PID 30080 을 강제로 종료)

- kill -HUP 30080 (PID 30080을 종료했다가 다시 실행)

→ 압축 명령어

- tar : 여러 개의 파일을 하나로 묶기

Ex) tar cvf test.tar /etc : etc 디렉토리를 test.tar 로 묶음.

tar xvf test.tar : 현재 방에 test.tar 파일 풀기

tar tvf test.tar : test.tar 에 들어 있는 내용보기

tar cvfz test.tar.gz /etc : /etc 디렉토리 묶고 압축

Ex) -r 옵션 TEST

```

]# touch a b c d e f
]# ls
a b c d e f
]# tar cvf test.tar *
a
b
c
d
e
f
]# ll
합계 12
-rw-r--r-- 1 root root    0 2015-07-04 00:24 a
-rw-r--r-- 1 root root    0 2015-07-04 00:24 b
-rw-r--r-- 1 root root    0 2015-07-04 00:24 c
-rw-r--r-- 1 root root    0 2015-07-04 00:24 d
-rw-r--r-- 1 root root    0 2015-07-04 00:24 e
-rw-r--r-- 1 root root    0 2015-07-04 00:24 f
-rw-r--r-- 1 root root 10240 2015-07-04 00:24 test.tar
[root@ora skykim]# touch g h i j
[root@ora skykim]# ll
합계 12
-rw-r--r-- 1 root root    0 2015-07-04 00:24 a
-rw-r--r-- 1 root root    0 2015-07-04 00:24 b
-rw-r--r-- 1 root root    0 2015-07-04 00:24 c
-rw-r--r-- 1 root root    0 2015-07-04 00:24 d
-rw-r--r-- 1 root root    0 2015-07-04 00:24 e
-rw-r--r-- 1 root root    0 2015-07-04 00:24 f
-rw-r--r-- 1 root root    0 2015-07-04 00:25 g
-rw-r--r-- 1 root root    0 2015-07-04 00:25 h
-rw-r--r-- 1 root root    0 2015-07-04 00:25 i
-rw-r--r-- 1 root root    0 2015-07-04 00:25 j
-rw-r--r-- 1 root root 10240 2015-07-04 00:24 test.tar
]# tar rvf test.tar g h i j          //아카이브 파일에 g h i j 파일 추가
g
h
i
j

]# ll
합계 12
-rw-r--r-- 1 root root    0 2015-07-04 00:24 a
-rw-r--r-- 1 root root    0 2015-07-04 00:24 b
-rw-r--r-- 1 root root    0 2015-07-04 00:24 c
-rw-r--r-- 1 root root    0 2015-07-04 00:24 d
-rw-r--r-- 1 root root    0 2015-07-04 00:24 e
-rw-r--r-- 1 root root    0 2015-07-04 00:24 f
-rw-r--r-- 1 root root    0 2015-07-04 00:25 g
-rw-r--r-- 1 root root    0 2015-07-04 00:25 h
-rw-r--r-- 1 root root    0 2015-07-04 00:25 i
-rw-r--r-- 1 root root    0 2015-07-04 00:25 j
-rw-r--r-- 1 root root 10240 2015-07-04 00:25 test.tar

]# tar tvf test.tar
-rw-r--r-- root/root    0 2015-07-04 00:24 a
-rw-r--r-- root/root    0 2015-07-04 00:24 b
-rw-r--r-- root/root    0 2015-07-04 00:24 c
-rw-r--r-- root/root    0 2015-07-04 00:24 d
-rw-r--r-- root/root    0 2015-07-04 00:24 e
-rw-r--r-- root/root    0 2015-07-04 00:24 f
-rw-r--r-- root/root    0 2015-07-04 00:25 g
-rw-r--r-- root/root    0 2015-07-04 00:25 h
-rw-r--r-- root/root    0 2015-07-04 00:25 i
-rw-r--r-- root/root    0 2015-07-04 00:25 j

```

```

]# ll // a 파일 내용 변경
합계 16
-rw-r--r-- 1 root root 25 2015-07-04 00:32 a
-rw-r--r-- 1 root root 0 2015-07-04 00:24 b
-rw-r--r-- 1 root root 0 2015-07-04 00:24 c
-rw-r--r-- 1 root root 0 2015-07-04 00:24 d
-rw-r--r-- 1 root root 0 2015-07-04 00:24 e
-rw-r--r-- 1 root root 0 2015-07-04 00:24 f
-rw-r--r-- 1 root root 0 2015-07-04 00:25 g
-rw-r--r-- 1 root root 0 2015-07-04 00:25 h
-rw-r--r-- 1 root root 0 2015-07-04 00:25 i
-rw-r--r-- 1 root root 0 2015-07-04 00:25 j
-rw-r--r-- 1 root root 10240 2015-07-04 00:25 test.tar

```

```

]# tar tvf test.tar
-rw-r--r-- root/root 0 2015-07-04 00:24 a
-rw-r--r-- root/root 0 2015-07-04 00:24 b
-rw-r--r-- root/root 0 2015-07-04 00:24 c
-rw-r--r-- root/root 0 2015-07-04 00:24 d
-rw-r--r-- root/root 0 2015-07-04 00:24 e
-rw-r--r-- root/root 0 2015-07-04 00:24 f
-rw-r--r-- root/root 0 2015-07-04 00:25 g
-rw-r--r-- root/root 0 2015-07-04 00:25 h
-rw-r--r-- root/root 0 2015-07-04 00:25 i
-rw-r--r-- root/root 0 2015-07-04 00:25 j

```

```

]# tar uvf test.tar *
a

```

tar: test.tar: file is the archive; not dumped

```

[root@ora skykim]# tar tvf test.tar // 중복된 파일이 있어도 해제시 최종 수정된 파일이 풀어진다.

```

```

-rw-r--r-- root/root 0 2015-07-04 00:24 a
-rw-r--r-- root/root 0 2015-07-04 00:24 b
-rw-r--r-- root/root 0 2015-07-04 00:24 c
-rw-r--r-- root/root 0 2015-07-04 00:24 d
-rw-r--r-- root/root 0 2015-07-04 00:24 e
-rw-r--r-- root/root 0 2015-07-04 00:24 f
-rw-r--r-- root/root 0 2015-07-04 00:25 g
-rw-r--r-- root/root 0 2015-07-04 00:25 h
-rw-r--r-- root/root 0 2015-07-04 00:25 i
-rw-r--r-- root/root 0 2015-07-04 00:25 j
-rw-r--r-- root/root 25 2015-07-04 00:32 a

```

옵션) -c : 새로운 아카이브 생성  
 -x : 압축 해제  
 -v : 작업내용 출력  
 -f : 아카이브 파일 사용. 즉 , tar 아카이브 파일 지정  
       압축 해제시 동일 옵션.  
 -z : gzip 형식으로 압축 시도.  
 -t : 파일 풀지않고 파일 안의 내용 확인  
 -p : 파일 퍼미션 보존  
 -u : 저장 파일에 저장된 사본보다 새로운 파일만을 덧붙인다.  
 -r : 저장 파일의 끝에 파일을 덧붙인다.

- gzip : tar 파일을 압축해서 tar.gz 생성

Ex) gzip test.tar : test.tar 를 압축해서 test.tar.gz 생성  
 gzip -d test.tar.gz : test.tar.gz 압축해제 test.tar 생성

gzip test : test 파일을 test.gz 형식으로 압축  
 gunzip test.gz : 압축된 파일을 해제

- compress

Ex) compress test.tar : test.tar 를 압축해서 test.tar.Z 를 생성  
 Umcompress test.tar.Z : test.tar.Z 를 압축해제 test.tar 생성

- bzip2

Ex) bzip2 test : test 파일을 test.bz2 로 압축

bunzip2 test.bz2 : bz2 파일을 압축 해제  
bzip2 -d test.gz2

→ date : 시스템 시간과 날짜 출력 및 설정

Ex) date 062012192007.30 : 6월 20일 12시 19분 2007년 30초  
MM : 월  
DD : 일  
hh : 시  
mm : 분  
CC : 연도의 처음 두 숫자  
YY : 연도의 나중 두 숫자  
ss : 초

ex) date -s mm:hh:ss → 시간 조정  
date -date '2 days ago' → 2일 이전의 날짜와 시간 정보 확인  
date -date '2 days' → 2일 후 날짜 시간정보  
date --date '3 months 1 day' → 3개월 1일이 지난 시간 정보

→ rdate : 지정된 원격지 time 서버로 부터의 시간 설정

Ex) rdate -s time.bora.net  
rdate -s time.kriss.or.kr && /sbin/clock -w

→ useradd(adduser) : 사용자 생성 명령어

Ex) adduser test → test 라는 사용자 생성  
passwd test → test 사용자 패스워드 생성

[ 참고 ] /etc/default/useradd 내용  
GROUP=100 : 새로 생성되는 사용자가 기본적으로 소속될 그룹의 GID 를 지정  
  
HOME=/home : 새로 생성되는 사용자의 홈디렉토리위치를 지정함!  
  
INACTIVE=-1 : 새로 생성되는 사용자의 패스워드 종료일수 이후의 유효기간  
  
EXPIRE= : 앞으로 추가 되는 계정들의 종료일수  
  
SHELL=/bin/bash : 새로 생성되는 사용자가 기본셸로 사용할 셸의 종류이다.  
  
SKEL=/etc/skel : 새로 생성되는 사용자의 홈디렉토리로 복사될 초기환경파일들이 저장된 디렉토리 지정하는곳

EX) adduser test -c "관리자계정" → 사용자 정보 추가

Ex) adduser -d /home2/test test → /home2 디렉토리에 test 사용자 생성  
passwd test → test 사용자 패스워드 설정

ex) adduser -d /data/test test -g sky → /data 디렉토리에 test 사용자 생성하고 그룹은 sky로 지정.

Ex) adduser -d /data/test2 test2 -g sky -s /bin/csh

Ex) adduser -d /data1/test6 test6 -e '2015-07-05' //유효기간 지정

확인 : grep test6 /etc/shadow

test6:\$6\$J7Y3mpDW\$V9oZr626x4Uj9V5XdwLj6CQs3mCRzwLiLRv5u6DPxCvCCZ6/32cD/5SG46ng9eAMAJBhBq3HF/X  
Vnj4YCPmM51:16619:0:99999:7::16621:

⇒ 계정생성과정

- ① adduser 명령
- ② /etc/default/useradd 파일과 /etc/login.defs 파일 참조
- ③ /etc/passwd 파일과 /etc/group 파일에 추가
- ④ 디렉토리에 계정 생성
- ⑤ /etc/skel 디렉토리에 있는 파일들 계정 디렉토리로 복사
- ⑥ /var/spool/mail 디렉토리에 계정 mail box 생성

⇒ 계정생성시 참조파일과 디렉토리

```
]# more /etc/default/useradd
# useradd defaults file
GROUP=100                //계정 생성시 기본적으로 속할 그룹 GID를 지정한 것.
HOME=/home               // 계정 생성이 기본 홈디렉토리 위치
INACTIVE=-1              // 패스워드 사용기간이 만기된 후 유효기간 여부 설정. (0 : 만기되면 사용불가능
                        -1 : 이 기능 사용하지 않음, 1이상: 종류일 이후의 유효기간
EXPIRE=                  // 계정 종료일자 지정, adduser -e 옵션 사용
SHELL=/bin/bash         // 기본 셸
SKEL=/etc/skel          // 계정 디렉토리로 복사될 환경 파일이 있는 디렉토리
CREATE_MAIL_SPOOL=yes   // 계정 생성시 메일 박스 생성여부

]# more /etc/login.defs
MAIL_DIR                /var/spool/mail    //메일 박스 디렉토리 위치

PASS_MAX_DAYS  99999    //패스워드 만료일. 즉, 동일 패스워드를 지속적으로 사용할 수 있는 최대일자
PASS_MIN_DAYS   0      //패스워드 변경 후 다시 변경할 수 있는 일자.
PASS_MIN_LEN    5      //패스워드 최소 길이
PASS_WARN_AGE   7      // 패스워드 종료일 7일전부터 사용자에게 알려준다

UID_MIN          500    // 사용자 UID 시작번호
UID_MAX          60000  // 사용자 UID 최대 번호

GID_MIN          500    //그룹의 GID 시작번호
GID_MAX          60000  // 그룹의 GID 최대 번호

CREATE_HOME      yes    //사용자 홈디렉토리를 생성할 것인지를 결정하는 값
UMASK            077    //UMASK 값, 지정하지 않으면 022 지정
USERGROUPS_ENAB yes    //사용자 생성시 그룹도 생성, 사용자 삭제시 그룹도 삭제
ENCRYPT_METHOD   SHA512 // 비밀번호 암호화 방법

]# ls -al /etc/skel/
합계 44
drwxr-xr-x.  4 root root  4096 2014-08-13 02:31 .
drwxr-xr-x. 137 root root 12288 2015-07-04 01:12 ..
-rw-r--r--.  1 root root   18 2013-02-22 06:05 .bash_logout
-rw-r--r--.  1 root root  176 2013-02-22 06:05 .bash_profile
-rw-r--r--.  1 root root  124 2013-02-22 06:05 .bashrc
drwxr-xr-x.  2 root root  4096 2010-11-12 09:54 .gnome2
-rw-r--r--.  1 root root  171 2014-08-06 18:38 .kshrc
drwxr-xr-x.  4 root root  4096 2014-08-13 01:23 .mozilla
-rw-r--r--.  1 root root   658 2012-06-23 08:14 .zshrc
```

→ userdel : 사용자 계정 삭제  
-r 옵션  
Ex) userdel test2

Userdel -r test2

→ usermod : 사용자 정보를 수정하는 명령어  
옵션) -c : 사용자의 설명을 수정  
-d : 홈디렉토리를 변경  
-m : 홈디렉토리 변경시 파일을 옮긴다.  
-L : 사용자 패스워드에 Lock을 걸어 로그인을 막는다  
-U : 사용자의 패스워드에 걸린 LOCK 을 푼다.

```
[root@ns ~]# usermod -c admin유저 skykim
skykim:x:504:504:admin유저:/home/skykim:/bin/bash
```

→ groupadd : 새로운 그룹 생성시 사용  
Ex ) groupadd dba  
groupadd -g 999 dba1

→ groupdel : 그룹 삭제  
Ex) groupdel dba1

→ su : 사용자 전환, 다른사용자의 권한으로 shell을 실행.  
[ 사용법 ] su [ 옵션 ] 사용자명  
[ 옵션 ] - : 사용자의 환경변수를 읽는다.

```
[skykim@db ~]$ su -  
Password:  
[root@db ~]#  
[root@db ~]# su - skykim  
[skykim@db ~]$
```

#### 4. vi 편집기

- vi 파일명

- 종료

:w - 데이터 저장  
:wq - 데이터 저장하고 종료.  
:wq! - 데이터 저장하고 강제로 종료.  
:q - 데이터 저장하지 않고 종료  
:q! - 데이터 저장하지 않고 강제 종료

- 커서 이동

h - 커서를 한 칸 왼쪽으로 이동  
j - 커서를 한 줄 아래로 이동  
k - 커서를 한 줄 위로 이동  
l - 커서를 한 칸 오른쪽으로 이동  
0 - 커서를 현재줄의 맨 앞으로 이동  
\$ - 커서를 현재 줄의 맨 끝으로 이동  
Ctrl + F - 한 화면 아래로 이동  
Ctrl + B - 한 화면 위로 이동  
gg - 파일의 처음으로 이동  
G - 파일의 끝으로 이동  
:1 - 1행으로 이동  
:\$= - 현재 편집중인 파일의 총 라인수  
:.= - 현재 커서가 놓인 행 번호

- 삽입

i - 입력 모드로 전환, 커서 위치 앞에서 삽입  
a - 입력모드로 전환, 커서 위치 뒤에서 삽입  
I - 입력모드로 전환, 현재줄의 앞에 삽입  
A - 입력모드로 전환, 현재줄의 끝에 삽입  
o - 입력모드로 전환, 현재 줄 아래에 전개  
O - 입력모드로 전환, 현재 줄 위에 전개

- 문자 삭제

x - 커서가 있는 문자 삭제  
X - 커서의 왼쪽 문자 삭제  
D - 커서부터 줄의 끝까지 삭제  
dd - 현재 줄 전체 삭제  
dG - 커서부터 편집 버퍼의 끝까지 삭제  
d1G - 커서부터 편집 버퍼의 맨 앞까지 삭제

- 취소

u - 편집 버퍼를 수정했던 마지막 명령을 취소  
U - 편집 버퍼를 수정했던 마지막 명령 반복

- 복사

yy - 한줄 복사  
5yy - 현재 커서부터 5줄 복사  
p - 현재 라인 밑에서부터 붙여넣기

- 탐색

- / - 전방 탐색
- ? - 후방탐색

- 정규식과 교체

- :! ls - 외부 명령어 ls 실행
- :sh - 잠깐 빠져나가기
- :r !ls - ls 결과를 불러 들이기
- :s/apple/banana/ - 현재행 첫번째 apple 를 banana 로 치환
- :s/apple/banana/g - 현재 행 모든 apple를 banana로 치환
- :%s/apple/banana/g - 모든 행의 apple를 banana로 치환
- :%s/[a-z]/WU&/g - 소문자를 대문자로 치환
- :g/^\$/d - 파일 전체 행의 공백 행 삭제
- :g/^M/s//g - ^M 문자 제거. ^M 은 ctrl + v 키 누르고 ctrl + M 키 누르면 표시된다.

### 5. RPM (RedHat Package Manager) 사용

- Redhat 사에서 만든 패키지 관리 도구,
- 리눅스 시스템을 패키지 형식으로 관리할 수 있는 관리 도구.
- RPM 파일은 크게 4개의 부분으로 이루어져 있다.  
패키지 명칭, 버전, 배포버전, 설치될 시스템(32비트 cpu인경우 i386,i586,i686, 64비트 CPU 는 x86\_64 로 표시된다)  
libxml2-2.6.16-6.i386.rpm

- 설치 및 업그레이드

```
]# rpm -ivh 패키지 이름
]# rpm -Uvh 패키지 이름
]# rpm -ivh libxml2-2.6.16-6.i386.rpm
Preparing... ##### [100%]
package libxml2-2.6.16-6 is already installed
→ 이미 설치되어 있는 경우
```

```
]# rpm -ivh --replacepkgs libxml2-2.6.16-6.i386.rpm
→ 이미 설치된 패키지에 덮어쓰기, 설치된 패키지에 문제가 있어 고쳐야 할 경우 사용한다.
```

```
]# rpm -ivh --force libxml2-2.6.16-6.i386.rpm
→ 패키지가 설치되어 있더라도 기존 패키지 무시하고 강제로 설치
```

```
]# rpm -ivh --nodeps xxx.rpm
→ 의존성 검사 무시하고 설치
```

- 패키지 제거

```
]# rpm -e 패키지 이름

]# rpm -e --nodeps curl-7.10.6-4.1
→ 의존성 검사 무시하고 제거
```

- 패키지 확인

```
]# rpm -q tar → 패키지 설치여부 확인
tar-1.13.25-13
]# rpm -qa | grep tar → 패키지 목록
setarch-1.3-1
startup-notification-0.5-1
redhat-config-kickstart-2.3.19-1
tar-1.13.25-13
star-1.5a08-4
]# rpm -qi sendmail → 패키지의 자세한 정보

]# rpm -qR sendmail → 의존 패키지를 보여준다.

]# rpm -qf /bin/ls → 특정 파일의 패키지 보여준다.

]# rpm -ql sendmail → 설치된 파일 목록

]# rpm -qc sendmail → 환경 설정파일에 해당되는 파일 보여준다.

]# rpm -Vf /bin/ls → 특정 파일의 설치후 변경상태 점검

]# rpm -Va sendmail → 전체 파일의 설치후 변경상태 점검
```

### 6. 커널(Kernel)

리눅스 운영체제는 커널(Kernel) 과 시스템 프로그램(System Program)으로 이루어져있다.

커널 - 파일들을 적절히 배치

프로그램들을 시동시켜 작업수행하게 함.

메모리와 같은 시스템자원을 각각의 프로세스에 할당.

네트워크를 통해 패킷을 주고 받을 수 있게 함.

파일 시스템 - 커널과 사용자 응용프로그램간의 연결다리 구실.

커널을 바탕으로 움직임.

→ 모든 프로그램들은 커널위에서 실행된다.

## 7. 셸(shell)

- 도스를 처음 사용할 때, 제일 먼저 필요한 파일이 command.com이다. Command.com은 운영체제를 사용하는데 있어서는 없어서는 안될 기본파일이다. 모든 명령어는 command.com을 통하여 해설을 거쳐야만 실행이 된다.

리눅스에서 이 command.com과 같은 역할을 하는 것이 셸(shell)이다.

리눅스에서 사용자가 입력한 문자열들을 해석하고 해당하는 명령어를 찾아 다시 셸로 보내면, 셸이 사용자에게 다시 나타내 주는 역할을 한다. 도스에서 command.com이 필요하듯이 리눅스에서 운영체제를 조작하기 위해서는 셸이라는 것이 필요하다.

- 셸선택 (로그인하고 나서 실행되는 기본 셸은 사용자 계정이 만들어 질 때 정해진다.)

```
[root@www1 ~]# echo $SHELL
/bin/bash
[root@www1 ~]#
```

→ bash : Bourne Again Shell-GNU 프로젝트의 결과물로 셸의 장점을 모두 수용하고 있으며, 스크립트는 Bourne Shell과 호환된다.

```
[skykim@www1 ~]$ chsh → 셸 바꿈
Changing shell for skykim.
Password:
New shell [/bin/bash]: /bin/csh
Shell changed.
```

- bash 설정 파일 : .bashrc, .bash\_profile

: 각 사용자의 디렉토리에 존재하며, 그사용자를 위한 고유 설정을 지정해 놓은 파일로서 셸이 작동할때 자동적으로 실행이 되는 파일.

: 이 파일에서 사용자가 사용하는 각 프로그램에서 사용할 환경변수들을 미리 지정해서 실행하도록 설정 할 수도 있고, 명령어의 기능들을 사용자의 취향에 맞게 지정해 놓을 수도 있다.

: 가장 먼저 읽혀지는 초기화 설정파일은 .bash\_profile이다.

: 터미널에 직접 연결되어 로그인되어 실행되는 셸은 대화형 셸이라 하며 이 경우의 초기화 파일은 .bashrc 이다.

- env : 환경변수출력

```
[root@www1 ~]# env
SHELL=/bin/bash
TERM=vt100
HISTSIZE=1000
USER=root
LS_COLORS=no=00:fi=00:di=01;34:ln=01;36:pi=40;33:so=01;35:bd=40;33:01:cd=40;33:01:or=01;05;37;41:mi=01;05;37;41:ex=01;32:*.cmd=01;32:*.exe=01;32:*.com=01;32:*.bat=01;32:*.sh=01;32:*.csh=01;32:*.tar=01;31:*.tgz=01;31:*.arj=01;31:*.taz=01;31:*.lzh=01;31:*.zip=01;31:*.Z=01;31:*.gz=01;31:*.bz2=01;31:*.bz=01;31:*.tz=01;31:*.rpm=01;31:*.cpio=01;31:*.jpg=01;35:*.gif=01;35:*.bmp=01;35:*.xpm=01;35:*.png=01;35:*.tif=01;35:
MAIL=/var/spool/mail/root
PATH=/usr/kerberos/sbin:/usr/kerberos/bin:/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin:/usr/X11R6/bin:/root/bin
INPUTRC=/etc/inputrc
PWD=/root
LANG=ko_KR.eucKR
SHLVL=1
HOME=/root
LOGNAME=root
LESSOPEN=|/usr/bin/lesspipe.sh %s
G_BROKEN_FILENAMES=1
_=/bin/env
```

- 패스(PATH)지정

```
[root@www1 ~]# echo $PATH
/usr/kerberos/sbin:/usr/kerberos/bin:/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin:/usr/X11R6/bin:/root/bin
```

```
[root@www1 ~]#
```

```
: PATH=$PATH:추가할 디렉토리
export PATH
```

- Alias 지정 (별칭)

: 어렵거나 긴 명령어를 사용자가 자신만이 아는 명령어로 바꿔서 사용할 때 쓰인다.

```
: alias ll='ls -al'
```

: 재부팅하면 사라짐. 계속하고 싶다면 .bashrc에 적어준다.

```
[skykim@www1 ~]$ ll
합계 3712
drwxr-xr-x 3 root root 4096 2월 28 14:03 public_html
[skykim@www1 ~]$ alias ll='ls -al'
[skykim@www1 ~]$ ll
합계 3752
drwx-----x 4 skykim skykim 4096 3월 1 21:04 .
drwxr-xr-x 5 root root 4096 12월 29 08:30 ..
-rw----- 1 skykim skykim 2091 3월 1 21:05 .bash_history
-rw-r--r-- 1 skykim skykim 24 11월 10 14:17 .bash_logout
-rw-r--r-- 1 skykim skykim 191 11월 10 14:17 .bash_profile
-rw-r--r-- 1 skykim skykim 125 3월 1 21:04 .bashrc
-rw-r--r-- 1 skykim skykim 383 11월 10 14:17 .emacs
-rw----- 1 skykim skykim 1278 3월 1 21:04 .viminfo
-rw-r--r-- 1 skykim skykim 658 11월 10 14:17 .zshrc
drwxr-xr-x 3 root root 4096 2월 28 14:03 public_html
```

- | (파이프라인)

: 앞부분의 출력결과를 뒷부분의 입력값으로 지정해주는 역할.

- history : 지금까지 사용했던 모든 명령어를 보여준다.

```
[skykim@www1 ~]$ history
 1 ls -al
 2 exit
 3 vi .bash_profile
 4 exit

skykim@www1 ~]$ history 10
325 set
326 ll
327 ls -al
328 exit
329 ll
330 alias ll='ls -al'
331 ll
332 alias ll='ls -al'
333 history
334 history 10
```

- Ctrl + R 을 누르면 실행했던 명령어 중 원하는 명령어를 찾아 준다.

## 2장 리눅스 서버 관리 실무

### 1. 파일 시스템 설정

- 파일 시스템과 관련된 설정 파일

/etc/fstab : 부팅시 마운트 할 파일 시스템 설정

/etc/mntab : 마운트 상황

① /etc/fstab

LABEL=/1	/	ext3	defaults	1 1
none	/dev/pts	devpts	gid=5,mode=620	0 0
none	/dev/shm	tmpfs	defaults	0 0
none	/proc	proc	defaults	0 0
none	/sys	sysfs	defaults	0 0
LABEL=SWAP-hda3	swap	swap	defaults	0 0
/dev/hdc	/media/cdrom	auto	pamconsole,exec,noauto,managed	0 0



### 3. 배드블럭 점검

- 배드블럭 검사 명령어 : badblocks
- 옵션 ) -b 블록 크기 : 블록 크기를 바이트 단위로 나타낸다.
  - o 출력파일 : 배드블럭의 목록을 파일에 저장.
  - v : 상세한 내용을 출력
  - w : 쓰기모드를 테스트한다. 이 옵션을 선택하면 장치상의 모든 블록에 써보고 다시 읽어 내용을 비교함으로써 배드블럭을 찾아낸다.
- 예 ) ]# badblocks -v /dev/hda2  
]#badblocks -v -w /dev/had2
- 배드블럭 마킹 : fsck.ext3 -cvf /dev/hda2 → 이 옵션시 주의요망

### 4. 파일 시스템 작성

- mkfs 이용
- 예) ]# mkfs.ext3 /dev/hdb1  
mkfs -t ext3 /dev/hdb1

### 5. 파일 시스템 점검

- fsck 이용 : 손상된 파일 시스템 점검 및 복구
- 파일 시스템을 점검하는 경우 반드시 마운트 해제하고 실행.
- / 파티션의 경우에는 읽기전용으로 다시 마운트 한 다음 fsck를 실행
- 읽기전용으로 리마운트  
]# mount -o remount,ro /dev/hda2
- / , /home 과 같은 파일시스템은 사용자가 로그인을 하고 있다면 계속 사용되고 있기 때문에 마운트를 해제하거나 리마운트 할수 없다. 이런경우 /etc/fstab 파일을 수정하고 재부팅한다.
- /dev/hdb3 점검  
]# umount /dev/hdb3  
]# fsck.ext3 /dev/hdb3  
]# fsck.ext3 -a /dev/hdb3

### 6. 로그파일

- ① /var/log/maillog - 메일과 관련된 로그
- ② /var/log/message - 메일 뉴스등을 제외한 전체적인 로그를 기록하는 파일
- ③ /var/log/secure - 접속을 기록
- ④ /var/log/lastlog - 사용자들이 마지막으로 로그인한 정보를 기록
- ⑤ /var/log/boot.log - 부팅시에 데몬들의 실행 상태를 기록
- ⑥ /var/log/dmesg - 부팅시 시스템에 출력되는 메시지들이 기록
- ⑦ /var/log/cron - cron 과 관련된 메시지들이 저장
- ⑧ /var/log/wtmp - 사용자들이 접속한 정보를 기록 , last
- ⑨ /var/log/xferlog - ftp 서버의 데이터 전송관련 로그를 기록

### 7.로그관리를 위한 데몬

- 로그데몬 : 시스템과 관련된 대부분의 로그는 syslogd가 하고 추가로 klogd라는 데몬이 있다.
- syslogd : 커널과 여러가지 시스템 프로그램들은 각종 에러와 경고 메시지, 기타 일반적인 메시지들을 출력한다. 이런 메시지들을 파일로 기록하는 데몬이다.
- klogd : 부팅후에 부팅과 관련된 메시지를 보려면 dmesg라는 명령을 친다. 이 명령은 실제 /var/log/dmesg라는 파일의 정보를 가지고 오는 것이 아니라, 시스템에는 부팅과 관련된 메시지를 포함하여 커널관련 메시지를 담은 약 8196bytes의 버퍼크기를 갖고 메시지를 기록하는 데 이 메시지를 기록하는 데몬이 klogd이다.

#### - 관련파일

```
/etc/rc.d/init.d/syslog : 실행과 관련된 스크립트파일로 실질적인 실행파일이다. syslogd와 klogd 데몬을 모두 실행시켜 준다.  
/etc/syslog.conf : syslog데몬의 환경설정파일이다.  
/etc/sysconfig/syslog : syslogd 와 klogd의 시작과 관련된 스크립트의 환경파일이다
```

#### - syslogd

```
]# ps aux | grep syslogd
```

root	4555	0.0	0.2	2384	524	?	Ss	12:24	0:00	syslogd	-m 0
root	9888	0.0	0.2	5332	640	pts/0	R+	15:18	0:00	grep	syslogd

- /etc/syslog.conf - 환경설정파일

```
#kern.* /dev/console

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

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

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

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

# Everybody gets emergency messages
*.emerg *
➔emerg 이상의 에러가 발생하면 모든 유저들이 긴급메시지를 받을수 있게한다.
# Save news errors of level crit and higher in a special file.
uucp,news.crit /var/log/spooler
➔ 메일, 뉴스에 관한 에러메시지를 기록
# Save boot messages also to boot.log
local7.* /var/log/boot.log
➔ 시스템 부팅시 데몬메시지를 기록
```

- facility : 메시지를 발생시키는 유형

```
cron : cron, at과 같은 프로그램이 발생한 메시지
auth : login과 같이 인증프로그램 유형이 발생한 메시지
authpriv : 개인 인증을 요하는 프로그램 유형이 발생한 메시지
daemon : telnetd,ftpd등과 같이 daemon이 발생한 메시지
kern : 커널이 발생한 메시지
lpr : 프린트 유형의 프로그램이 발생한 메시지
mail : mail 시스템이 발생한 메시지
mark : syslogd에 의해 만들어지는 날짜유형
news : 유즈넷 뉴스 프로그램 유형이 발생한 메시지
syslog : syslog 프로그램이 유형이 발생한 메시지
user : 사용자 프로세스
uucp : UUCP 시스템이 발생한 메시지
local0-local7 : 여분으로 남겨둔 유형
```

- level : 위험 정도

```
Debug : 프로그램을 디버깅할 때 발생하는 메시지
info : 통계, 기본정보 메시지
notice : 특별한 주의를 요하나 에러는 아닌 메시지
warning : 주의를 요하는 경고 메시지
err : 에러가 발생하는 경우의 메시지
crit : 크게 급하지는 않지만 시스템에 문제가 생기는 단계의 메시지
alert! : 즉각적인 조정을 해야 하는 상황
emerg : 모든 사용자들에게 전달되어야 할 위험한 상황
```

- /etc/logrotate.conf

```
# see "man logrotate" for details
# rotate log files weekly
weekly

# keep 4 weeks worth of backlogs
rotate 4
➔ 순환될 파일 개수 , 0 부터 시작 주간유지
# create new (empty) log files after rotating old ones
create
```

```
# uncomment this if you want your log files compressed
#compress

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

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

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

- /etc/logrotate.d/

```
]# ls -al /etc/logrotate.d/
total 128
drwxr-xr-x  2 root root  4096 Dec  8 12:00 .
drwxr-xr-x 87 root root 12288 Dec  8 13:58 ..
-rw-r--r--  1 root root   144 Feb 22  2005 acpid
-rw-r--r--  1 root root   161 Jul 25 23:56 cups
-rw-r--r--  1 root root   571 Jul 26 05:43 mgetty
-rw-r--r--  1 root root   163 Jul 26 03:26 named
-rw-r--r--  1 root root    91 Jul 26 19:49 opensm
-rw-r--r--  1 root root   136 Feb 22  2005 ppp
-rw-r--r--  1 root root   319 Jan 18  2007 psacct
-rw-r--r--  1 root root    61 Jul 26 05:49 rpm
-rw-r--r--  1 root root    68 Jul 26 08:34 sa-update
-rw-r--r--  1 root root   228 Apr 11  2008 syslog
-rw-r--r--  1 root root    32 Dec  6  2006 up2date
-rw-r--r--  1 root root   92 Jul 25 21:14 vsftpd.log
-rw-r--r--  1 root root    89 Nov 19  2007 yum
```

- /etc/logrotate.d/yum

```
]# more yum
/var/log/yum.log {
    Missingok → 로그파일이 없더라도 에러를 내지 않는다.
    notifempty → 로그 파일이 비어 있는 경우 순환을 하지 않는다.
    size 30k → 파일의 크기를 30k 넘지 않도록 한다.
    create 0600 root root
}
```

## 8. Quota 설정

-quota 설치 확인

```
]# rpm -qa quota
quota-3.12-6.el4
]# dmesg | grep quota
VFS: Disk quotas dquot_6.5.1
```

- quota 설정

: /etc/fstab 파일에 적용할 파일 시스템에 usrquota 추가, 리마운트

```
LABEL=/data1          /data          ext3    defaults,usrquota    1 2

]# mount -o remount /data
]# mount
/dev/hda3 on /data type ext3 (rw,usrquota)
```

- quota 를 설정할 파티션의 최상의 디렉토리에 aquota.user 파일 생성, 권한 600으로 변경, quota 적용을 위해 현재 상태 체크

```
]# touch /data/aquota.user
]# chmod 700 aquota.user
]# quotacheck -avug
quotacheck: WARNING - Quotafile /data/aquota.user was probably truncated. Can't save quota settings...
```

```
quotacheck: Scanning /dev/hda3 [/data] quotacheck: Old group file not found. Usage will not be
substracted.
done
quotacheck: Checked 58 directories and 703 files
```

- quota 시작, 종료

```
]# quotaon -avug
/dev/hda3 [/data]: user quotas turned on

]# quotaoff -avug
/dev/hda3 [/data]: user quotas turned off
```

- quota의 현재 상황

```
]# repquota -a
*** Report for user quotas on device /dev/hda3
Block grace time: 7days; Inode grace time: 7days
```

User	Block limits				File limits			
	used	soft	hard	grace	used	soft	hard	grace
root	-- 86196	0	0		413	0	0	
nfsnobody	-- 5028	0	0		7	0	0	
ssr	-- 7052	0	0		335	0	0	
mysql	-- 20	0	0		1	0	0	
skykim	-- 12	0	0		3	0	0	

- used : 현재 사용량,
- soft : 현재 설정된 soft limit
- hard : 현재 설정된 hard limit
- grace :soft limit 에 지정된 용량 초과시 남은 유효기간

-특정사용자 quota 설정내역 확인

```
]# quota -v skykim
Disk quotas for user skykim (uid 504):
```

Filesystem	blocks	quota	limit	grace	files	quota	limit	grace
/dev/hda3	12	0	0		3	0	0	

- 유저 용량 제한

```
]# edquota -u skykim

Disk quotas for user skykim (uid 504):
```

Filesystem	blocks	soft	hard	inodes	soft	hard
/dev/hda3	16	50000	55000	4	0	0

- Filesystem : quota 를 적용할 파일 시스템
- blocks : 현재 유저의 총 사용량
- inodes : 현재 유저의 파일갯수
- soft : 일반적 의미의 디스크 제한용량
- hard : 절대로 넘을수 없는 용량 제한
- 설정단위 : kilobyte

-개별 계정사용자가 자기자신의 quota 설정내역을 확인

```
]# quota -u skykim
Disk quotas for user skykim (uid 504):
```

Filesystem	blocks	quota	limit	grace	files	quota	limit	grace
/dev/hda3	16	50000	55000		4	0	0	

- Filesystem : quota 를 적용할 파일 시스템
- blocks : 현재 유저의 총 사용량

## 9 .서버 관리 명령어

### ① free - 메모리 관리

- 옵션) -b : 메모리 양을 바이트로 표시  
 -k : 메모리 양을 Kb로 표시  
 -m : Mb로 표시  
 -g  
 -t : 총합을 표시  
 -o : 버퍼 정보 출력하지 않는다.  
 -s : 지정된 시간마다 계속해서 출력

-V : 버전정보출력

```
]# free -s 3 → 3초 단위
      total        used         free   shared    buffers     cached
Mem:   515540      313468      202072         0      182096     110880
-/+ buffers/cache:    20492      495048
Swap:   522072           0      522072

      total        used         free   shared    buffers     cached
Mem:   515540      313468      202072         0      182096     110880
-/+ buffers/cache:    20492      495048
Swap:   522072           0      522072

      total        used         free   shared    buffers     cached
Mem:   515540      313468      202072         0      182104     110880
-/+ buffers/cache:    20484      495056
Swap:   522072           0      522072
```

- total : 전체메모리 용량을 Kbyte 단위로 표시
- used : 시스템에서 사용중인 메모리 용량.
- free : 시스템에서 사용주이지 않은 메모리 용량
- shared : 시스템에서 공유한 메모리 용량
- buffers : 현재 시스템에서 버퍼링된 메모리 용량
- chched : 현재 시스템에서 캐싱된 메모리 용량.
- /+ buffers/cache : 현재 캐시 메모리에서 버퍼링된 사용량을 표시

② pstree - 프로세스 정보를 트리 형태로 출력

```
[root@www sysconfig]# pstree
init--+-crond
  |--events/0--+-aio/0
  |   |--ata/0
  |   |--ata_aux
  |   |--kacpid
  |   |--kauditd
  |   |--kblockd/0
  |   |--khelper
  |   |--2*[pdflush]
  |--khubd
  |--7*[kjournald]
  |--klogd
  |--kseriod
  |--ksoftirqd/0
  |--kswapd0
  |--6*[mingetty]
  |--sshd---sshd---bash---pstree
  |--syslogd
  |--udev
  |--xfs
  `--xinetd
```

③ setup - 설정관리 프로그램

```
[root@www sysconfig]# setup
```

텍스트 모드 설정 유틸리티 1.17

(c) 1999-2004 Red Hat, In

```

  |-----|
  | 도구들 선택해 주십시오 |
  |-----|
  | X 설정 |
  | 네트워킹 설정 |
  | 마우스 설정 |
  | 방화벽 설정 |
  | 시간대 설정 |
  | 시스템 서비스 |
  | 인종 설정 |
  | 키보드 설정 |
  | 프린터 설정 |
  |-----|
  | 실행 도구 | | 종료 |
  |-----|

```

<Tab>/<Alt-Tab> 구성요소 이동 | <Enter>를 사용하여 선택 사항을 편집

Ready

ssh2: AFS-12f |

④ ntsysv : 시스템 서비스 관리 도구

⑤ chkconfig : 실행 레벨 관리해주는 명령어

```
]# chkconfig --help
```

```

사용법:  chkconfig [--list] [--type <종류>] [이름]
         chkconfig --add <이름>
         chkconfig --del <이름>
         chkconfig --override <이름>
         chkconfig [--level <레벨>] [--type <종류>] <이름> <on|off|reset|resetpriorities>

```

```
]# chkconfig --list
```

```
]# chkconfig --list | grep sendmail
```

```

sendmail      0:해제  1:해제  2:활성  3:해제  4:활성  5:해제  6:해제
→ 0,1,2,3,4,5,6, 은 runlevel 을 의미, 현재 sendmail 은 runlevel 2번과 4번에 활성화

```

```

]# chkconfig --del sendmail           // sendmail 시스템 서비스에서 삭제
]# chkconfig --list | grep sendmail   //확인

```

```

]# ls -al /etc/rc.d/init.d/sendmail   // /etc/rc.d/init.d에 sendmail 구동 스크립트가 있어야 한다.
-rwxr-xr-x. 1 root root 3702 2010-02-16 00:36 /etc/rc.d/init.d/sendmail

```

```

]# chkconfig --add sendmail           // sendmail 시스템 서비스에 추가
]# chkconfig --list | grep sendmail   //확인

```

```

sendmail      0:해제  1:해제  2:활성  3:활성  4:활성  5:활성  6:해제

```

```
]# chkconfig --level 24 sendmail off
```

```
]# chkconfig --list | grep sendmail
```

```

sendmail      0:해제  1:해제  2:해제  3:활성  4:해제  5:활성  6:해제

```

⑥ uptime

- 시스템 부하율 확인.
- 1분, 5분, 15분간의 시스템 평균 부하율을 각각 출력.
- 현재 시간
- 시스템이 부팅후에 종료없이 얼마나 운영되었는지에 대한 시간.
- 현재 시스템에 로그인된 사용자 수

```
[root@www sysconfig]# uptime
15:10:35 up 1:33, 1 user, load average: 0.00, 0.00, 0.00
```

⑦ w 또는 who - 시스템 사용자 확인

```

]# w
23:50:34 up 54 min, 3 users, load average: 0.00, 0.00, 0.00
USER TTY FROM LOGIN@ IDLE JCPU PCPU WHAT
root tty1 - 23:30 20:19 0.00s 0.00s -bash
root pts/0 192.168.0.3 23:06 24:44 0.16s 0.16s -bash
root pts/1 192.168.0.3 23:30 0.00s 0.04s 0.00s w

```

→ 서버의 현재시간, 부팅 후 작동시간, 접속자수, 평균부하율  
→ 사용자, 접속TTY 명, 접속IP 주소, 로그인시각정보, idle 타임, 접속자별 CPU 사용정보(JCPU,PCPU), 명령어  
→ JCPU : TTY 장치명에서 사용하는 모든 프로세스의 CPU 사용시간. 지난 백그라운드 잡은 포함하지 않고 현재 운영중인 백그라운드 잡만 포함한다.  
PCUP : WHAT 필드에 있는 현재 프로세스 명에서 사용하는 cpu 시간

```

]# who
root pts/0 2015-07-03 23:06 (192.168.0.3)

```

]# who -u // 그인 후 사용자가 얼마의 시간, 분 동안 한가하게 있었는지 출력해준다. '.' 은 바로 방금 전 까지도 활동을 하고 있다는 의미이다. 'old' 는 24 시간 이상 한가하게 있었음을 의미한다.

```

root tty1 2015-07-03 23:30 00:07 2345
root pts/0 2015-07-03 23:06 00:11 2813 (192.168.0.3)
root pts/1 2015-07-03 23:30 . 3025 (192.168.0.3)

```

]# who -m // 'who am i'와 같음, who 명령을 실행한 사용자 표시

```

root pts/1 2015-07-03 23:30 (192.168.0.3)

```

]# who -q // 로그인명과 로그인 수를 출력한다.

```

root root root
# users=3

```

10. 한글 설정

```

[root@www sysconfig]# cd /etc/sysconfig/
[root@www sysconfig]# pwd
/etc/sysconfig
[root@www sysconfig]# vi i18n
LANG="ko_KR.eucKR"
SUPPORTED="en_US.UTF-8:en_US:en:ko_KR.UTF-8:ko_KR:ko"
SYSFONT="latarcyrheb-sun16"
[root@www sysconfig]# source /etc/sysconfig/i18n
[root@www sysconfig]# env | grep LANG
LANG=ko_KR.eucKR

```

11. 사용자 계정에 대한 파일 - /etc/passwd, /etc/shadow, /etc/group

- /etc/passwd

```

]# more /etc/passwd
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin

```

→ (Login name : Password : UID : GID : Comment : Home Directory : Login Shell)

- /etc/shadow

```

]# more /etc/shadow
root:$1$j1kafBXM$tA9Gd1VBFZSACEe8fYEIa.:14099:0:99999:7:::
bin:!:14099:0:99999:7:::
daemon:!:14099:0:99999:7:::

```

- (사용자 이름 : 암호화된 비밀번호 : 최종 변경일 : 최대 변경일 : 경고 일수 : 최대값 비활성일수 : 만기 날짜)
- 최종변경일 : 기준날짜 "1970년 1월 1일"로부터 명시된 날이 지나고 변경
- 최소 값 변경일수 : 명시된 날짜 이전에는 비밀번호 변경이 안된다.
- 최대 값 변경일 수 : password Aging 을 구현할 때 기준, 명시된 일 이후에는 passwd가 강제적으로 실행 되어 비밀번호를 무조건 바꿔야 한다.

- 경고 일수 : 최대값 변경일수 7일전부터 passwd 변경 경고
- 최대 값 비활성 일 수 : 명시된 날 이상 로그인 하지 않을 시 로그인 불가능
- 만기 날짜 : 기준 날짜를 기점으로 설정된 날 이후로는 로그인 불가능

- /etc/group

```

]# more /etc/group
root:x:0:root
bin:x:1:root,bin,daemon
  
```

→ (그룹이름 : 암호화된 그룹 비밀번호 : GID : 사용자 목록)

### 제 3장 리눅스 네트워크 관리

#### 1. TCP/IP 프로토콜

- TCP 프로토콜과 IP 프로토콜
- IP(Internet Protocol) 프로토콜 : 데이터를 어떻게 전송하는가를 다루는 프로토콜
- TCP(Transmission Control Protocol) 프로토콜 : 데이터를 어떻게 안정적으로 보낼것인가를 정의하고 있는 프로토콜.

#### 2. IP 주소

- 0.0.0.0 을 2진수로 표기

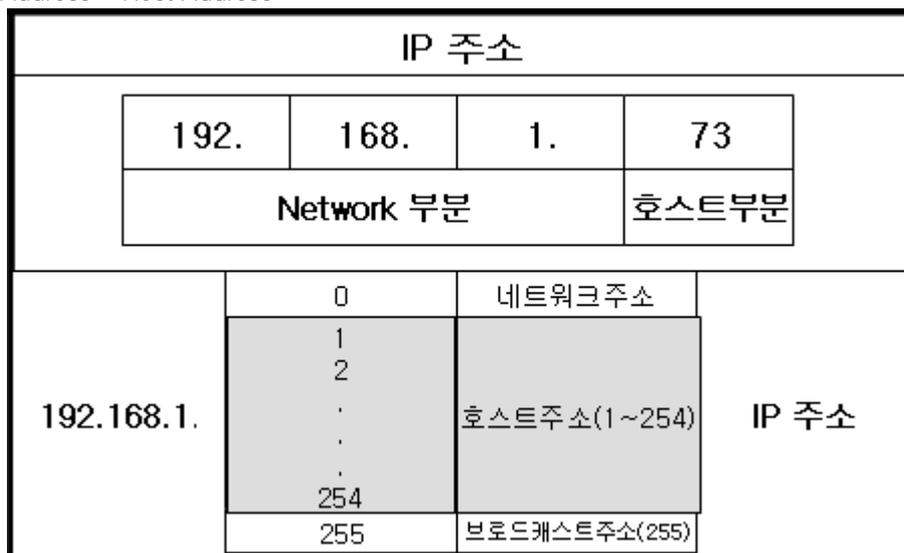
0	.	0	.	0	.	0
00000000 . 00000000 . 00000000 . 00000000						

- 255.255.255.255 를 2진수로 표기

255	.	255	.	255	.	255
11111111 . 11111111 . 11111111 . 11111111						

#### 3. IP 주소 및 범위

- IP 주소는 Network Address + Host Address



- 데이터 전송을 위한 라우팅은 IP 어드레스 부분중 네트워크 주소부분을 근거로 하여 이루어진다.
- ip 주소의 범위

구분	IP 범위	네트워크	호스트	네트워크 수	IP 수
A Class	0.0.0.0~ 127.255.255.255	8 bit	32bit	128	2 <sup>32</sup>
B Class	128.0.0.0 ~191.255.255.255	16 bit	16 bit	16,384	2 <sup>16</sup>
C Class	192.0.0.0 ~223.255.255.255	24 bit	8 bit	2,097,152	2 <sup>8</sup>

D Class	224.0.0.0 ~239.255.255.255	멀티캐스팅을 위해 사용되는 특별 주소
E Class	240.0.0.0 ~255.255.255.255	예비 주소로 남겨둔 주소

#### A 클래스

- 첫번째 숫자가 1 ~ 127 사이의 값을 가진다.
- 32bit 중 맨 첫번째 bit 가 0 이다
- 첫번째 1byte(8bit)가 네트워크 주소이며 나머지 3byte(24bit)가 호스트 주소이다.
- $2^7$  즉 128개의 네트워크 주소가 존재
- 사실 IP : 10.0.0.0 ~ 10.255.255.255
- Loopback 주소 : 127.0.0.0 ~ 127.255.255.255

#### B 클래스

- 첫번째 숫자가 128 ~ 191사이의 값을 가진다.
- 전체 IP 주소 32 Bit 중 10 으로 시작한다.
- 첫번째 2byte 가 네트워크 주소이며 나머지 2byte 가 호스트 주소
- $2^{14}$  즉, 16,384 개의 네트워크가 존재.
- 사실 IP : 172.16.0.0 ~ 172.31.0.0

#### C 클래스

- 첫번째 숫자가 192 ~ 233 사이의 값
- 전체 IP 주소 32 bit 중 첫번째가 110 으로 시작
- 첫번째 3byte 가 네트워크 주소이며 나머지 1byte 가 호스트주소이다.
- $2^{21}$  즉, 2,097,152개의 네트워크가 존재.
- 사실 IP : 192.168.0.0 ~ 192.168.255.255

#### D 클래스

- 첫번째 숫자가 224 ~ 239사이의 값을 가진다.
- 이 클래스의 ip 주소들은 모두 호스트 주소로 사용하지 않는 IP 주소들이다.
- IP 주소 32 bit 중 첫번째가 1110 으로 시작

#### E 클래스

- 첫번째 숫자가 240 ~ 255 사이의 값
- 전체 IP 주소 32 bit 중 첫부분이 11110 로 시작

#### NetMask

- 하나의 네트워크를 2개이상의 네트워크로 나누어 사용할 때에 각각의 네트워크로 구분 할 수 있도록 해주는 것, Subnet Mask

#### 네트워크 주소

- 하나의 네트워크를 통칭하기 위해 사용하는 주소
- C클래스의 경우 위 표에서 보는 바와 같이 192.168.1.0 이 네트워크 주소가 된다. 만약 서브 넷이 이루어져 네트워크를 2개로 나누면 192.168.1.0 과 192.168.1.128 이 네트워크 주소가 된다.

#### 브로드캐스트

- 특정 네트워크에 속하는 모든 호스트들이 듣게 되는 주소.
- 특정 네트워크의 맨 마지막 주소를 사용.
- c 클래스인경우 끝자리 주소 255 가 되지만 서브넷으로 나눈 경우 두개의 네트워크 주소가 존재하므로 127 과 255 의 브로드캐스트 주소가 존재한다.

#### 4. ifconfig 활용

- 네트워크 인터페이스 정보 결과를 나타낸다.
- ifconfig 를 이용하여 IP 설정 삭제

```
]# ifconfig eth0 down
```

- eth0 다시 사용

```
]# ifconfig eth0 up
```

- eth0에 대한 네트워크 설정

```
]# ifconfig eth0 192.168.0.200 netmask 255.255.255.0 broadcast 192.168.0.255 up
```

#### 5. network 설정 파일들을 이용한 네트워크 설정

- 설정 파일 : /etc/sysconfig/network, /etc/sysconfig/network-scripts/ifconfig-eth0
- 네트워크 스크립트 : /etc/rc.d/init.d/network start | stop | restart
- /etc/sysconfig/network 파일

```
]# more network
NETWORKING=yes
HOSTNAME=www.test.com
```

```
GATEWAY=192.168.0.1
```

- /etc/sysconfig/network-scripts/ifconfig-eth0

```
]# more ifcfg-eth0
DEVICE=eth0
ONBOOT=yes
BOOTPROTO=static
IPADDR=192.168.0.204
NETMASK=255.255.255.0
GATEWAY=192.168.0.0
```

## 6. 여러 개의 IP 주소 할당

- 현재 시스템 인터페이스

```
]# ifconfig
eth0      Link encap:Ethernet  HWaddr 00:50:8D:36:10:B0
          inet addr:192.168.0.204  Bcast:192.168.0.255  Mask:255.255.255.0
          inet6 addr: fe80::250:8dff:fe36:10b0/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:650 errors:0 dropped:0 overruns:0 frame:0
          TX packets:373 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:59145 (57.7 KiB)  TX bytes:94689 (92.4 KiB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:6 errors:0 dropped:0 overruns:0 frame:0
          TX packets:6 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:448 (448.0 b)  TX bytes:448 (448.0 b)
```

- 물리적인 랜카드 1개에 IP 주소 할당

```
]# cd /etc/sysconfig/network-scripts/
]# pwd
/etc/sysconfig/network-scripts
]# cp ifcfg-eth0 ifcfg-eth0:1
]# vi ifcfg-eth0:1
DEVICE=eth0:1
ONBOOT=yes
BOOTPROTO=static
IPADDR=192.168.0.205
NETMASK=255.255.255.0
GATEWAY=192.168.0.0
→ 저장하고 빠져나옴

]# /etc/rc.d/init.d/network restart
인터페이스 eth0 (을)를 종료함: [ 확인 ]
loopback 인터페이스를 종료함: [ 확인 ]
IPv4 패킷 포워딩을 종료하고 있습니다: [ 확인 ]
네트워크 매개 변수를 설정하고 있습니다: [ 확인 ]
loopback 인터페이스 활성화중 입니다: [ 확인 ]
eth0 인터페이스 활성화중 입니다: [ 확인 ]

]# ifconfig → 할당된 IP 주소 확인
eth0      Link encap:Ethernet  HWaddr 00:50:8D:36:10:B0
          inet addr:192.168.0.204  Bcast:192.168.0.255  Mask:255.255.255.0
          inet6 addr: fe80::250:8dff:fe36:10b0/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:1067 errors:0 dropped:0 overruns:0 frame:0
          TX packets:682 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:94768 (92.5 KiB)  TX bytes:131623 (128.5 KiB)

eth0:1    Link encap:Ethernet  HWaddr 00:50:8D:36:10:B0
```

```
inet addr:192.168.0.205 Bcast:192.168.0.255 Mask:255.255.255.0
UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1

lo      Link encap:Local Loopback
        inet addr:127.0.0.1  Mask:255.0.0.0
        inet6 addr: ::1/128 Scope:Host
        UP LOOPBACK RUNNING MTU:16436 Metric:1
        RX packets:6 errors:0 dropped:0 overruns:0 frame:0
        TX packets:6 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:0
        RX bytes:448 (448.0 b) TX bytes:448 (448.0 b)
```

### 7. HostName 변경하기

- 호스트 네임 수정 파일 : /etc/sysconfig/network
- 호스트네임 확인

```
]# hostname
www.test.com
```

- 호스트네임 수정

```
]# cd /etc/sysconfig/
]# pwd
/etc/sysconfig

]# vi network
NETWORKING=yes
HOSTNAME=ns.test.com
GATEWAY=192.168.0.1

→ 저장하고 빠져나옴.
```

→ 로그아웃 후 로그인 적용

- 재부팅 하지 않고 일시적으로 적용

```
]# hostname
www.test.com
[root@www sysconfig]# hostname ns.test.com
[root@www sysconfig]# hostname
ns.test.com
```

→ 재부팅 후 다시 돌아감.

### 8. ping - 시스템 동작 확인

- 옵션) -s : 패킷사이즈 지정
- q : 종합 결과 보연준다
- i : 지연시간을 설정
- c : 보낼 패킷수를 지정해준다.

```
[root@www modules]# ping 192.168.0.20
PING 192.168.0.20 (192.168.0.20) 56(84) bytes of data.
64 bytes from 192.168.0.20: icmp_seq=0 ttl=128 time=0.182 ms
64 bytes from 192.168.0.20: icmp_seq=1 ttl=128 time=0.221 ms
64 bytes from 192.168.0.20: icmp_seq=2 ttl=128 time=0.237 ms
64 bytes from 192.168.0.20: icmp_seq=3 ttl=128 time=0.233 ms
64 bytes from 192.168.0.20: icmp_seq=4 ttl=128 time=0.215 ms

--- 192.168.0.20 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 3998ms
rtt min/avg/max/mdev = 0.182/0.217/0.237/0.025 ms, pipe 2
```

→기본적으로 아무 옵션없이 사용하면 계속 패킷을 보냄, 멈추고 싶으면 Ctrl+C를 입력

```
[root@www modules]# ping -c 3 192.168.0.20
PING 192.168.0.20 (192.168.0.20) 56(84) bytes of data.
64 bytes from 192.168.0.20: icmp_seq=0 ttl=128 time=0.195 ms
64 bytes from 192.168.0.20: icmp_seq=1 ttl=128 time=0.221 ms
64 bytes from 192.168.0.20: icmp_seq=2 ttl=128 time=0.215 ms

--- 192.168.0.20 ping statistics ---
```

```
3 packets transmitted, 3 received, 0% packet loss, time 2000ms
rtt min/avg/max/mdev = 0.195/0.210/0.221/0.016 ms, pipe 2
```

→보낼 횟수 3회로 지정

```
[root@www modules]# ping -s 1000 192.168.0.20
PING 192.168.0.20 (192.168.0.20) 1000(1028) bytes of data.
1008 bytes from 192.168.0.20: icmp_seq=0 ttl=128 time=0.677 ms
1008 bytes from 192.168.0.20: icmp_seq=1 ttl=128 time=0.695 ms
1008 bytes from 192.168.0.20: icmp_seq=2 ttl=128 time=0.687 ms
1008 bytes from 192.168.0.20: icmp_seq=3 ttl=128 time=0.698 ms
```

--- 192.168.0.20 ping statistics ---

```
4 packets transmitted, 4 received, 0% packet loss, time 2999ms
rtt min/avg/max/mdev = 0.677/0.689/0.698/0.020 ms, pipe 2
```

→패킷 크기 지정, 기본적으로 패킷의 크기는 64byte(56byte + 8byte(헤더))

→ ping 에 응답하지 않게 설정

```
]# pwd
/proc/sys/net/ipv4
]# echo 1 > net.ipv4.icmp_echo_ignore_all
```

### 9. traceroute 활용

- 대상 호스트까지 어떤 경로를 거쳐 통신이 이루어 지는지 확인.

```
]# traceroute yjc.ac.kr
traceroute to yjc.ac.kr (210.101.228.138), 30 hops max, 38 byte packets
 1  211.233.5.81 (211.233.5.81)  0.479 ms  0.489 ms  0.438 ms
 2  210.124.146.65 (210.124.146.65)  0.542 ms  0.555 ms  0.411 ms
 3  tgug12bb2-pos1-2.rt.bora.net (210.120.196.253)  0.482 ms  210.108.232.141 (210.108.232.141)
0.629 ms tgug12bb2-pos1-2.rt.bora.net (210.120.196.253)  0.584 ms
 4  210.92.194.185 (210.92.194.185)  5.529 ms  203.233.17.121 (203.233.17.121)  5.993 ms
203.233.17.125 (203.233.17.125)  5.612 ms
 5  210.120.102.202 (210.120.102.202)  5.487 ms  5.324 ms  203.233.17.202 (203.233.17.202)
5.478 ms
 6  210.92.194.254 (210.92.194.254)  5.235 ms  5.903 ms  5.551 ms
 7  203.229.222.225 (203.229.222.225)  11.308 ms  10.649 ms  10.419 ms
 8  59.18.34.113 (59.18.34.113)  17.905 ms  20.585 ms  17.858 ms
 9  218.145.32.226 (218.145.32.226)  14.745 ms  11.444 ms  17.648 ms
10  218.145.40.194 (218.145.40.194)  10.397 ms  10.241 ms  10.583 ms
11  121.181.235.198 (121.181.235.198)  17.336 ms  16.382 ms  16.809 ms
12  121.181.236.2 (121.181.236.2)  11.355 ms  10.806 ms  10.690 ms
13  211.199.18.182 (211.199.18.182)  12.868 ms  13.149 ms  12.592 ms
14  * * *
```

### 10.route 활용

- 라우팅 테이블은 요청된 데이터 패킷들을 어디로 보낼 것인가를 결정하는 역할
- 현재 서버의 라우팅 경로를 설정하기 위한 것.
- 리눅스 서버를 라우터로 사용하려면 두개 이상의 인터페이스가 필요.

```
]# route
Kernel IP routing table
Destination Gateway Genmask Flags Metric Ref Use Iface
192.168.0.0 * 255.255.255.0 U 0 0 0 eth0
169.254.0.0 * 255.255.0.0 U 0 0 0 eth0
default 192.168.0.1 0.0.0.0 UG 0 0 0 eth0
```

→ 규칙

- ① 위의 첫번째 행부터 다음행으로 차례대로 한 행씩 읽어 들어서 처리할 패킷을 보낼 수 있는가를 확인.
- ② default 가 서버의 기본 게이트웨이 설정행의 의미, 그 행의 Gateway 항목값이 기본게이트웨이의 IP 주소이다. Default 는 0.0.0.0 을 의미
- ③ 보낼 데이터의 목적지 IP 주소와 Genmask값(SubnetMask) 과의 AND 연산을 해서 그결과가 해당행의 Destination 항목과 동일 할 경우에 해당행의 Iface 에 있는 인터페이스로 해당 패킷을 전송하게 된다.
- ④ Iface 항목의 값이 lo 인 행은 루프백 데이터를 처리하기 위한 설정

→ Destination : 목적지

Gateway : 게이트웨이 주소

Genmask : 목적지 네트워크의 넷마스크 주소,

255.255.255.255 로 지정 되어 있으면 목적지 호스트 주소,  
 0.0.0.0 으로 지정되어있으면 게이트웨이 주소

Flags : 해당경로에 대한 정보를 알려준다.  
 U 는 경로가 살아있는 상태, G 는 게이트웨이, H 는 목적지 호스트

Metric : 목적지 까지 거리, 최소 경로로 경로 설정

Ref : 라우터에서 참조한 값, 리눅스커널에서는 사용하지 않음.

Use : 라우터에서 경로를 탐색한 수

Iface : 패킷이 오고갈 인터페이스

- route 대표적인 예

형식 : route add default gw 게이트웨이IP주소 dev 네트워크인터페이스장치명

```
]# route add default gw 192.168.0.1 dev eth0
```

- 현재 시스템과 통신이 가능하도록 네트워크 통신 경로 추가

```
route add -net 192.168.1.0 netmask 255.255.255.0 dev eth0
```

## 11. netstat 활용

- 네트워크 연결, 라우팅 테이블, 네트워크 인터페이스상태등을 종합적으로 확인 할 수 있는 필수 네트워크 도구.

옵션) -a : --all과 같으며 listen되는 소켓정보와 listen 되지 않은 소켓정보 모두를 보여준다.

-n : --numeric 과 같으며 10진수의 수치정보로 결과를 보여준다.

-r : --route 와 같으면 설정된 라우팅 정보를 보여준다.

-p : --program 과 같으며 실행되고 있는 각 프로그램과 PID 정보를 출력.

-i : 모든 네트워크 인터페이스 정보를 출력.

-c : netstat 결과를 연속적으로 출력.

-l : 현재 listen 되고 있는 소켓정보를 출력.

-s : --statistics 와 같으며 각 프로토콜에 대한 통계정보를 출력

- 라우팅 테이블 정보확인

```
[root@www modules]# netstat -nr
```

Kernel IP routing table

Destination	Gateway	Genmask	Flags	MSS Window	irtt Iface
192.168.0.0	0.0.0.0	255.255.255.0	U	0 0	0 eth0
169.254.0.0	0.0.0.0	255.255.0.0	U	0 0	0 eth0

- 현재 열려 있는 포트 확인

```
root@www modules]# netstat -an | grep LISTEN
```

```
tcp      0      0 0.0.0.0:901          0.0.0.0:*          LISTEN
tcp      0      0 0.0.0.0:199         0.0.0.0:*          LISTEN
tcp      0      0 0.0.0.0:139        0.0.0.0:*          LISTEN
tcp      0      0 0.0.0.0:23         0.0.0.0:*          LISTEN
tcp      0      0 0.0.0.0:445        0.0.0.0:*          LISTEN
tcp      0      0 :::127.0.0.1:8005   :::*              LISTEN
tcp      0      0 :::8009            :::*              LISTEN
tcp      0      0 :::80              :::*              LISTEN
tcp      0      0 :::22             :::*              LISTEN
unix    2      [ ACC ] STREAM LISTENING 5595 /tmp/.font-unix/fs7100
```

```
[root@www modules]# netstat -an
```

Active Internet connections (servers and established)

Proto	Recv-Q	Send-Q	Local Address	Foreign Address	State
tcp	0	0	0.0.0.0:901	0.0.0.0:*	LISTEN
tcp	0	0	0.0.0.0:199	0.0.0.0:*	LISTEN
tcp	0	0	0.0.0.0:139	0.0.0.0:*	LISTEN
tcp	0	0	0.0.0.0:23	0.0.0.0:*	LISTEN
tcp	0	0	0.0.0.0:445	0.0.0.0:*	LISTEN
tcp	0	0	192.168.0.204:139	192.168.0.20:1927	ESTABLISHED
tcp	0	0	:::ffff:127.0.0.1:8005	:::*	LISTEN
tcp	0	0	:::8009	:::*	LISTEN
tcp	0	0	:::80	:::*	LISTEN
tcp	0	0	:::22	:::*	LISTEN
tcp	0	0	:::ffff:192.168.0.204:22	:::ffff:192.168.0.20:1101	ESTABLISHED
udp	0	0	192.168.0.204:137	0.0.0.0:*	
udp	0	0	0.0.0.0:137	0.0.0.0:*	
udp	0	0	192.168.0.204:138	0.0.0.0:*	
udp	0	0	0.0.0.0:138	0.0.0.0:*	
udp	0	0	0.0.0.0:161	0.0.0.0:*	

Active UNIX domain sockets (servers and established)						
Proto	RefCnt	Flags	Type	State	I-Node	Path
unix	2	[ ACC ]	STREAM	LISTENING	5595	/tmp/.font-unix/fs7100
unix	2	[ ]	DGRAM		3147	@udev
unix	6	[ ]	DGRAM		5324	/dev/log
unix	2	[ ]	DGRAM		59115	
unix	2	[ ]	DGRAM		5543	
unix	2	[ ]	DGRAM		5524	
unix	2	[ ]	DGRAM		5491	
unix	2	[ ]	DGRAM		5332	

- ➔ LISTEN : 연결이 가능하도록 데몬이 떠있으며 연결이 가능.
- SYN-SENT : 연결을 요청한 상태
- SYN\_RECEIVED : 연결요구에 대한 응답을 준후에 확인메시지를 기다리고 있는 상태.
- ESTABLISHED : 앞의 3단계 연결과정이 모두 종료된 후에 연결이 완료된상태.
- FIN-WAIT, CLOSE-WAIT, FIN-WAIT2 : 연결 종료 요청을 받은후의 종료과정
- CLOSING : 전송된 메시지가 유실된 상태.
- TIME-WAIT : 연결종류후에 한동안 유지하고 있는 상태.
- CLOSED : 연결이 완전히 종료됨.

```
[root@www modules]# netstat -lnp
```

Active Internet connections (only servers)							
Proto	Recv-Q	Send-Q	Local Address	Foreign Address	State		
PID/Program name							
tcp	0	0	0.0.0.0:901	0.0.0.0:*	LISTEN		
2706/xinetd							
tcp	0	0	0.0.0.0:199	0.0.0.0:*	LISTEN		
2654/snmpd							
tcp	0	0	0.0.0.0:139	0.0.0.0:*	LISTEN		
28217/smbd							
tcp	0	0	0.0.0.0:23	0.0.0.0:*	LISTEN		
2706/xinetd							
tcp	0	0	0.0.0.0:445	0.0.0.0:*	LISTEN		
28217/smbd							
tcp	0	0	::ffff:127.0.0.1:8005	:::*	LISTEN	2724/java	
tcp	0	0	:::8009	:::*	LISTEN		
2724/java							
tcp	0	0	:::80	:::*	LISTEN		
2777/httpd							
tcp	0	0	:::22	:::*	LISTEN		
2694/sshd							
udp		0	0	192.168.0.204:137	0.0.0.0:*		
28222/nmbd							
udp		0	0	0.0.0.0:137	0.0.0.0:*		
28222/nmbd							
udp		0	0	192.168.0.204:138	0.0.0.0:*		
28222/nmbd							
udp		0	0	0.0.0.0:138	0.0.0.0:*		
28222/nmbd							
udp		0	0	0.0.0.0:161	0.0.0.0:*		
2654/snmpd							
Active UNIX domain sockets (only servers)							
Proto	RefCnt	Flags	Type	State	I-Node	PID/Program name	Path
unix	2	[ ACC ]	STREAM	LISTENING	5595	2767/xf	/tmp/.font-unix/fs7100

```
[root@www modules]# netstat -atp
```

Active Internet connections (servers and established)						
Proto	Recv-Q	Send-Q	Local Address	Foreign Address	State	
PID/Program name						
tcp	0	0	*:swat	:::*	LISTEN	
2706/xinetd						
tcp	0	0	*:smux	:::*	LISTEN	
2654/snmpd						
tcp	0	0	*:netbios-ssn	:::*	LISTEN	
28217/smbd						
tcp	0	0	*:telnet	:::*	LISTEN	

2706/xinetd					
tcp	0	0	*:microsoft-ds	*:*	LISTEN
28217/smbd					
tcp	0	0	www.test.com:netbios-ssn	192.168.0.20:1927	ESTABLISHED
28227/smbd					
tcp	0	0	localhost.localdomain:8005	*:*	LISTEN
2724/java					
tcp	0	0	*:8009	*:*	LISTEN
2724/java					
tcp	0	0	*:http	*:*	LISTEN
2777/httpd					
tcp	0	0	*:ssh	*:*	LISTEN
2694/sshd					
getnameinfo failed					
tcp	0	0	www.test.com:ssh	[UNKNOWN]:1101	ESTABLISHED
3403/0					

## 12. whois활용

- 특정 도메인 정보나 특정 IP Address, 특정 네트워크를 사용하고 있는 기관정보를 조회할 수 있다.
- 국내도메인 조회

```

]# whois yjc.ac.kr@whois.nic.or.kr
[Querying whois.nic.or.kr]
[whois.nic.or.kr]

query: yjc.ac.kr

# KOREAN

도메인이름       : yjc.ac.kr
등록인          : 영진전문대학
등록인 주소     : 대구 북구 복현2동 영진전문대학 218
등록인 우편번호 : 702721
책임자         : 이상용
책임자 전자우편 : sylee@yjc.ac.kr
책임자 전화번호 : +82.539405161
등록일         : 2000. 08. 07.
최근 정보 변경일 : 2008. 01. 11.
사용 종료일     : 2009. 08. 07.
정보공개여부   : Y
등록대행사     : (주)아이네임즈(http://www.inames.co.kr)

1차 네임서버 정보
  호스트이름     : ns.yjc.ac.kr
  IP 주소       : 210.101.225.1

2차 네임서버 정보
  호스트이름     : ns2.yjc.ac.kr
  IP 주소       : 220.89.62.156

네임서버 이름이 .kr이 아닌 경우는 IP주소가 보이지 않습니다.

# ENGLISH

Domain Name       : yjc.ac.kr
Registrant       : YEUNGJIN JUNIOR COLLEGE
Registrant Address : Bokhyeon 2(i)-dong, Buk-gu, Daegu, Republic of Korea, Bokhyeon 2-
dong, Buk-gu Daegu, KR
Registrant Zip Code : 702721
Administrative Contact(AC): Sang-yong, Lee
AC E-Mail        : sylee@yjc.ac.kr
AC Phone Number  : +82.539405161
Registered Date  : 2000. 08. 07.
Last updated Date : 2008. 01. 11.
Expiration Date  : 2009. 08. 07.
Publishes       : Y
Authorized Agency : Inames Co., Ltd.(http://www.inames.co.kr)

```

Primary Name Server  
Host Name : ns.yjc.ac.kr  
IP Address : 210.101.225.1

Secondary Name Server  
Host Name : ns2.yjc.ac.kr  
IP Address : 220.89.62.156

]# whois 211.199.18.186@whois.nic.or.kr  
[Querying whois.nic.or.kr]  
[whois.nic.or.kr]

query: 211.199.18.186

# KOREAN

조회하신 IPv4주소는 ISP가 아직 할당하지 않은 주소이거나 고객(End-User)에게 IPv4주소를 할당한 후 할당내역을 한국인터넷진흥원에 등록하지 않은 주소공간입니다.

따라서, 조회하신 IPv4주소에 대한 문의는 아래의 ISP 담당자에게 문의하시기 바랍니다.

[ ISP의 IPv4주소 관리 기관 정보 ]

기관명 : 주식회사 케이티  
서비스명 : KORNET  
기관 주소 : 성남시 분당구 정자동  
기관 상세 주소: 206 한국통신 e-Biz본부 기획팀

[ ISP의 IPv4주소 책임자 정보 ]

이름 : IP주소관리자  
전화번호 : +82-2-3674-5708  
전자우편 : kornet\_ip@kt.com

[ ISP의 IPv4주소 담당자 정보 ]

이름 : IP주소담당자  
전화번호 : +82-2-3674-5708  
전자우편 : kornet\_ip@kt.com

[ ISP의 Network Abuse 담당자 정보 ]

이름 : 스팸/해킹담당  
전화번호 : +82-2-100-0000  
전자우편 : abuse@kornet.net

# ENGLISH

KRNIC is not an ISP but a National Internet Registry similar to APNIC.  
The IPv4 address is allocated and still held by the following ISP,  
or its Whois information is not updated after assigned to end users.

Please contact following ISP for further information.

[ ISP Organization Information ]

Org Name : Korea Telecom  
Service Name : KORNET  
Org Address : Jungja-dong, Bundang-gu, Sungnam-ci  
Org Detail Address: 206

[ ISP IPv4 Admin Contact Information ]

Name : IP Administrator  
Phone : +82-2-3674-5708  
E-Mail : kornet\_ip@kt.com

[ ISP IPv4 Tech Contact Information ]

Name : IP Manager  
Phone : +82-2-3674-5708  
E-mail : kornet\_ip@kt.com



```
Atlanta, GA 30348
US
4048275000 fax: 4048271593
```

```
Record expires on 23-Sep-2017.
Record created on 22-Sep-1993.
Database last updated on 18-Jan-2009 23:40:06 EST.
```

Domain servers in listed order:

```
TWDNS-01.NS.AOL.COM
TWDNS-02.NS.AOL.COM
TWDNS-03.NS.AOL.COM
TWDNS-04.NS.AOL.COM
```

### 13. modprobe를 이용한 랜카드 모드 설정 및 변경

- modprobe : 리눅스 커널 모듈 확인 설정 명령어
- lsmod 로 현재 로드되어있는 커널 모듈 확인

```
]# lsmod
Module                Size  Used by
soundcore              10017  1 snd
e100                   33861  0
mii                   5313  1 e100
floppy                 58609  0
ext3                   117833  7
jbd                    72153  1 ext3
ata_piix               15173  0
libata                 111645  1 ata_piix
sd_mod                 17345  0
scsi_mod               125517  2 libata,sd_mod
```

→ e100 이라는 NIC 모듈 사용.

- 위 모듈명은 /etc/modprobe.conf 파일 참조

```
]# more /etc/modprobe.conf
alias eth0 e100
alias scsi_hostadapter ata_piix
alias snd-card-0 snd-intel8x0
options snd-card-0 index=0
install snd-intel8x0 /sbin/modprobe --ignore-install snd-intel8x0 && /usr/sbin/alsactl restore
>/dev/null 2>&
1 || :
remove snd-intel8x0 { /usr/sbin/alsactl store >/dev/null 2>&1 || : ; }; /sbin/modprobe -r --ignore-
remove snd
-intel8x0
alias usb-controller uhci-hcd
```

## 제 4장 XINETD

### 1. xinetd 이란 ?

- 인터넷 슈퍼데몬을 의미.
- 데몬이란? 커널 상에서 백그라운드 모드로 작동하여 비활성화 상태에서 요청이 있을때만 동작하는 프로세스
- 리눅스 서버에서 서비스되는 여러 가지 데몬들을 제어하면서 각각의 서비스들의 연결을 담당.
- 대표적 서비스 : telnet 등
- 각각의 서비스별로 별도의 파일에 설정이 가능
- xinetd에서 가지고 있던 접근 제어 기능을 가지고 있음.
- tcp\_wrapper를 내장하고 있기 때문에 접근 제어를 할 수 있음.
- timeout 설정으로 서비스를 접근제어를 할 수 있음.
- 접속 시도 횟수로 접근 제어를 할 수 있으므로 무차별 서비스 거부 공격을 방지 할 수 있음.
- 동일한 IP를 가진 호스트에서 동시 접속자수를 제어하여 접근 제어를 할 수 있음.
- 로그 파일의 크기를 제한 할 수 있음.
- xinetd에서 제어되는 각 서비스들에 대한 syslog 로깅 레벨 설정가능
- 접속하는 클라이언트들의 서비스 이용시간을 기록할 수 있음.
- 서비스를 거부하거나 서비스 접근 제어가 되었을 경우에 상세로그를 기록함.

### 2. 데몬실행 두가지 방법(standalone과 xinetd)

구분	Xinetd 환경서비스	Standalone서비스
의미	Xinetd 에 의해 제어되는 서비스들의 데몬	독립적으로 실행되는 서비스데몬
실행방법	필요할 때 xinetd 에 의해 수행됨	독립적 서비스를 위하여 항상 독립적으로 수행
데몬상주여부	<b>Xinetd에 의해 불리워진후에 서비스, 완료후에는 종료됨</b>	<b>항상 메모리에 독립적으로 상주</b>
기타	“/etc/xinetd.d/서비스명” 으로 된 xinetd 제어파일이 존재함	
접근제어	Tcpd에 의해 접근제어	Tcpd 와는 무관하지만 자체 설정파일에 의해 접근제어를 할 수도 있음
참고	Standalone에서 Xinetd환경으로 변경, xinetd환경에서 Standalone로 변경	

### 3. xinetd.conf파일

- xinetd 의 최상위 설정 파일
- /etc/xinetd.d/ 디렉토리내의 각 서비스 설정 파일들에 대하여 .global 하게 영향을 준다.
- xinetd 서비스에 관련된 파일  
( /etc/xinetd.conf, /etc/xinetd.d/서비스명, /usr/sbin/xinetd, /etc/services, /etc/protocols, /etc/syslog.conf, /usr/sbin/tcpd, /etc/hosts.allow, /etc/hosts.deny, /var/log/secure, /etc/rc.d/init.d/xinetd)

### 4. /etc/xinetd.d/ 디렉토리내의 각 설정 파일내에 설정가능한 지시자들

```

service telnet
{
    flags          = REUSE
    socket_type    = stream
    wait           = no
    user           = root
    server         = /usr/sbin/in.telnetd
    log_on_failure += USERID
    disable        = yes
}

```

- service : 서비스 이름, /etc/services 파일에 등록된 서비스이름과 동일
- disable : 해당 서비스 서비스 여부.
- Flag : 재사용 가능
- socket\_type : tcp 일경우 stream, udp일경우 dgram 으로 명시
- wait : xinetd 가 서비스 요청을 받은 경우, 이후에 즉시 또 다른 요청을 처리할것인지(no) 아닌지(yes)의 여부를 결정, stream 일 경우 반드시 no.
- user : 서비스 사용자 권한.
- Server : telnet 서비스 실행시 어느 위치에 있는 프로그램(데몬)을 불러와 연결할 것인가를 명시
- Log\_on\_failure : 서버 접속에 성공하지 못하였을 때 로그 파일에 기록하는 내용들을 설정.  
+= 는 /etc/xinetd.conf 파일의 기본설정 항목에 추가할 항목  
-+ 는 /etc/xinetd.conf 파일의 기본 설정 항목에 뺄 항목
- Log\_on\_success : 서버 접속에 성공하였을 경우에 기록할 내용 설정.
- Only\_from : “ only\_from = 192.168.1.0/24” 와 같이 설정 가능,
- No\_access : “no\_access = 192.168.1.100” 과 같이 설정가능.
- Instances : telnet 서비스로 접속할 수 있는 총접속자수.
- Access\_times : “access\_times = 08:00-18:00” 설정, 해당 서비스 이용가능시간을 오전 8시부터 오후 6시까지로
- Per\_source : “per\_source = 5” , 동일 호스트에서 해당 서비스로 접속시 동시에 5번 이상 접속할 수 없다.

### 5. xinetd 의 시작과 종료

```

]# /etc/rc.d/init.d/xinetd stop
xinetd 를 정지함: [ 확인 ]

]# /etc/rc.d/init.d/xinetd start
xinetd (을)를 시작합니다: [ 확인 ]

]# /etc/rc.d/init.d/xinetd restart
xinetd 를 정지함: [ 확인 ]
xinetd (을)를 시작합니다: [ 확인 ]

]# /etc/rc.d/init.d/xinetd status
xinetd (pid 3719) 가 실행하고 있습니다..

```

### 6. xinetd에 의한 접속 제어(tcpd - tcp\_wrapper)

- 특정 ip나 도메인으로부터 서버로의 telnet, ftp, pop 등의 접속 차단.
- 접속 기록이나 접속 시도기록을 특정한 파일에 로그로 기록

- xinetd 는 tcp\_wrapper 를 내장하고 있기 때문에 xinetd 모드에서 실행되는 서비스들(ftp, telnet, ssh 등)은 거의 대부분 tcpd 라는 tcp\_wrapper 의 데몬에 의해 접속제어를 받게 된다.
- /etc/hosts.allow : 접속 허용 설정  
/etc/hosts.deny : 접속 허용 거부
- hosts.allow와 hosts.deny 에 중복 설정될 경우 hosts.allow 우선
- hosts.deny 예

```
ALL : ALL
```

→ xinetd에 의해 서비스 되는 모든 서비스들을 거부

- hosts.allow

```
sshd : 192.168.0.20
in.telnetd : 192.168.0.20 192.168.0.21 192.168.0.22
ipop3d : ALL
```

## 7. /etc/services 파일

- 리눅스 에서 사용하는 모든 포트들에 대한 정의 설정.

```
]# more /etc/services
# /etc/services:
# $Id: services,v 1.40 2004/09/23 05:45:18 notting Exp $
#
# Network services, Internet style
#
# Note that it is presently the policy of IANA to assign a single well-known
# port number for both TCP and UDP; hence, most entries here have two entries
# even if the protocol doesn't support UDP operations.
# Updated from RFC 1700, "Assigned Numbers" (October 1994). Not all ports
# are included, only the more common ones.
#
# The latest IANA port assignments can be gotten from
# http://www.iana.org/assignments/port-numbers
# The Well Known Ports are those from 0 through 1023.
# The Registered Ports are those from 1024 through 49151
# The Dynamic and/or Private Ports are those from 49152 through 65535
#
# Each line describes one service, and is of the form:
#
# service-name port/protocol [aliases ...] [# comment]

tcpmux      1/tcp          # TCP port service multiplexer
tcpmux      1/udp          # TCP port service multiplexer
rje         5/tcp          # Remote Job Entry
rje         5/udp          # Remote Job Entry
echo        7/tcp
echo        7/udp
discard     9/tcp          sink null
discard     9/udp          sink null
systat      11/tcp         users
systat      11/udp         users
daytime     13/tcp
daytime     13/udp
qotd        17/tcp          quote
qotd        17/udp          quote
msp         18/tcp          # message send protocol
msp         18/udp          # message send protocol
chargen     19/tcp          ttytst source
chargen     19/udp          ttytst source
ftp-data    20/tcp
ftp-data    20/udp
# 21 is registered to ftp, but also used by fsp
ftp         21/tcp
ftp         21/udp          fsp fspd
ssh         22/tcp          # SSH Remote Login Protocol
ssh         22/udp          # SSH Remote Login Protocol
telnet      23/tcp
telnet      23/udp
# 24 - private mail system
lmtp        24/tcp          # LMTP Mail Delivery
```

```

lmtpp      24/udp      # LMTTP Mail Delivery
smtp      25/tcp      mail
smtp      25/udp      mail
time      37/tcp      timserver
time      37/udp      timserver
rlp       39/tcp      resource      # resource location
rlp       39/udp      resource      # resource location
nameserver 42/tcp      name          # IEN 116
nameserver 42/udp      name          # IEN 116
nicname   43/tcp      whois
nicname   43/udp      whois
tacacs    49/tcp      # Login Host Protocol (TACACS)
tacacs    49/udp      # Login Host Protocol (TACACS)
re-mail-ck 50/tcp

```

~~~~~

~~~~~

~

~~~~~

## 8. /etc/protocols 파일

- /etc/services 파일에서 정의되는 프로토콜의 정의가 되어있는 파일.

```

]# more /etc/protocols
# /etc/protocols:
# $Id: protocols,v 1.3 2001/07/07 07:07:15 nalin Exp $
#
# Internet (IP) protocols
#
#      from: @(#)protocols      5.1 (Berkeley) 4/17/89
#
# Updated for NetBSD based on RFC 1340, Assigned Numbers (July 1992).
#
# See also http://www.iana.org/assignments/protocol-numbers

```

|         |    |             |                                                  |
|---------|----|-------------|--------------------------------------------------|
| ip      | 0  | IP          | # internet protocol, pseudo protocol number      |
| #hopopt | 0  | HOPOPT      | # hop-by-hop options for ipv6                    |
| icmp    | 1  | ICMP        | # internet control message protocol              |
| igmp    | 2  | IGMP        | # internet group management protocol             |
| ggp     | 3  | GGP         | # gateway-gateway protocol                       |
| ipencap | 4  | IP-ENCAP    | # IP encapsulated in IP (officially ``IP'')      |
| st      | 5  | ST          | # ST datagram mode                               |
| tcp     | 6  | TCP         | # transmission control protocol                  |
| cbt     | 7  | CBT         | # CBT, Tony Ballardie <A.Ballardie@cs.ucl.ac.uk> |
| egp     | 8  | EGP         | # exterior gateway protocol                      |
| igp     | 9  | IGP         | # any private interior gateway (Cisco: for IGRP) |
| bbn-rcc | 10 | BBN-RCC-MON | # BBN RCC Monitoring                             |
| nvp     | 11 | NVP-II      | # Network Voice Protocol                         |
| pup     | 12 | PUP         | # PARC universal packet protocol                 |
| argus   | 13 | ARGUS       | # ARGUS                                          |
| emcon   | 14 | EMCON       | # EMCON                                          |
| xnet    | 15 | XNET        | # Cross Net Debugger                             |
| chaos   | 16 | CHAOS       | # Chaos                                          |
| udp     | 17 | UDP         | # user datagram protocol                         |
| mux     | 18 | MUX         | # Multiplexing protocol                          |
| dcn     | 19 | DCN-MEAS    | # DCN Measurement Subsystems                     |
| hmp     | 20 | HMP         | # host monitoring protocol                       |
| prm     | 21 | PRM         | # packet radio measurement protocol              |
| xns-idp | 22 | XNS-IDP     | # Xerox NS IDP                                   |
| trunk-1 | 23 | TRUNK-1     | # Trunk-1                                        |
| trunk-2 | 24 | TRUNK-2     | # Trunk-2                                        |
| leaf-1  | 25 | LEAF-1      | # Leaf-1                                         |
| leaf-2  | 26 | LEAF-2      | # Leaf-2                                         |
| rdp     | 27 | RDP         | # "reliable datagram" protocol                   |
| irtp    | 28 | IRTP        | # Internet Reliable Transaction Protocol         |
| iso-tp4 | 29 | ISO-TP4     | # ISO Transport Protocol Class 4                 |
| netblt  | 30 | NETBLT      | # Bulk Data Transfer Protocol                    |

|            |    |            |                                        |
|------------|----|------------|----------------------------------------|
| mfe-nsp    | 31 | MFE-NSP    | # MFE Network Services Protocol        |
| merit-inp  | 32 | MERIT-INP  | # MERIT Internodal Protocol            |
| sep        | 33 | SEP        | # Sequential Exchange Protocol         |
| 3pc        | 34 | 3PC        | # Third Party Connect Protocol         |
| idpr       | 35 | IDPR       | # Inter-Domain Policy Routing Protocol |
| xtp        | 36 | XTP        | # Xpress Transfer Protocol             |
| ddp        | 37 | DDP        | # Datagram Delivery Protocol           |
| idpr-cmtp  | 38 | IDPR-CMTP  | # IDPR Control Message Transport Proto |
| tp++       | 39 | TP++       | # TP++ Transport Protocol              |
| il         | 40 | IL         | # IL Transport Protocol                |
| ipv6       | 41 | IPv6       | # IPv6                                 |
| sdrp       | 42 | SDRP       | # Source Demand Routing Protocol       |
| ipv6-route | 43 | IPv6-Route | # Routing Header for IPv6              |
| ipv6-frag  | 44 | IPv6-Frag  | # Fragment Header for IPv6             |
| idrp       | 45 | IDRP       | # Inter-Domain Routing Protocol        |
| rsvp       | 46 | RSVP       | # Resource ReSerVation Protocol        |
| gre        | 47 | GRE        | # Generic Routing Encapsulation        |
| mhrp       | 48 | MHRP       | # Mobile Host Routing Protocol         |
| bnr        | 49 | BNA        | # BNA                                  |
| ipv6-crypt | 50 | IPv6-Crypt | # Encryption Header for IPv6           |
| ipv6-auth  | 51 | IPv6-Auth  | # Authentication Header for IPv6       |
| i-nlsp     | 52 | I-NLSP     | # Integrated Net Layer Security TUBA   |
| swipe      | 53 | SWIPE      | # IP with Encryption                   |
| narp       | 54 | NARP       | # NBMA Address Resolution Protocol     |

~~~~~  
~  
~  
~

### 9. 호스트 이름 table

- /etc/hosts
- 네임서버를 거치기 전의 가장 간단한 형태의 호스트이름 -> ip 주소로 변환

```

]# more /etc/hosts
# Do not remove the following line, or various programs
# that require network functionality will fail.
127.0.0.1          localhost.localdomain localhost
192.168.0.204     ns.test.com ns
192.168.0.203     ns2.test.com ns2

```

### 제 5장 리눅스 부트 과정

- 전원 ON → ROM-BIOS 실행(POST → MBR에 있는 boot loader를 loading 함) → GRUB 실행 → GRUB에 의해 커널 이미지 파일 loading → GRUB에 의해 로딩된 커널이 swapper를 loading 함 → Swapper 에 의해 init실행 → init에 의해 /etc/inittab 불러워짐 → /etc/inittab 파일내의 실행내용 실행 → /etc/inittab 파일내의 /etc/rc.d/rc.sysinit 로딩되어 실행

#### 1. 부트 과정

- ① 전원 ON
- ② ROM-BIOS 실행
  - ROM-BIOS 의 BIOS 프로그램 실행
- ③ ROM-BIOS의 POST(자체 진단 기능) 수행
  - 자체 진단 기능
  - post 과정이 아무런 이상 없이 수행되었을 때 부트로드를 불러들여 시스템 부팅 시작
- ④ ROM-BIOS 가 부트로드(GRUB) 불러 들임.
  
- ⑤ ROM-BIOS 의 실행 종료와 GRUB 실행시작
  - 검색된 부팅 매체인 0번섹터(MBR)에서 Boot Program을 읽어들여 메모리로 적재한다.
  - Boot Program 에 리눅스의 Boot Loader 인 GRUB 가 들어 있다.
  - ROM-BIOS는 이 GRUB 을 메모리에 적재 한후에 모든 제어권을 GRUB에 넘겨 주고 사라진다.
  - 시스템의 모든 제어권을 GRUB가 가지고 부팅과정을 이어간다.
  
- ⑥ GRUB의 실행 초기 단계
  - GRUB의 부팅 화면
- ⑦ GRUB에 의한 부팅 메뉴 선택

- GRUB 명령어 모드로 들어가서 여러가지 작업을 할 수 있다.

- ⑧ Grub 커널 이미지를 메모리로 적재와 swapper 호출
  - 메모리로 로딩된 커널은 swapper를 호출
- ⑨ swapper의 장치드라이브 초기화
  - grub에 의해 swapper라는 PID 0번인 프로세스가 실행되며, swapper 는 각 장치 드라이브들을 초기화한다.
- ⑩ swapper의 init 프로세스 실행.
- ⑪ init 프로세스에 의해 /etc/inittab 파일 호출
  - /etc/inittab 파일

```
]# more /etc/inittab
#
# inittab      This file describes how the INIT process should set up
#              the system in a certain run-level.
#
# Author:      Miquel van Smoorenburg, <miquels@drinkel.nl.mugnet.org>
#              Modified for RHS Linux by Marc Ewing and Donnie Barnes
#
# Default runlevel. The runlevels used by RHS are:
#  0 - halt (Do NOT set initdefault to this)
#  1 - Single user mode
#  2 - Multiuser, without NFS (The same as 3, if you do not have networking)
#  3 - Full multiuser mode
#  4 - unused
#  5 - X11
#  6 - reboot (Do NOT set initdefault to this)
#
id:3:initdefault:

# System initialization.
si::sysinit:/etc/rc.d/rc.sysinit

l0:0:wait:/etc/rc.d/rc 0
l1:1:wait:/etc/rc.d/rc 1
l2:2:wait:/etc/rc.d/rc 2
l3:3:wait:/etc/rc.d/rc 3
l4:4:wait:/etc/rc.d/rc 4
l5:5:wait:/etc/rc.d/rc 5
l6:6:wait:/etc/rc.d/rc 6

# Trap CTRL-ALT-DELETE
#ca::ctrlaltdel:/sbin/shutdown -t3 -r now

# When our UPS tells us power has failed, assume we have a few minutes
# of power left.  Schedule a shutdown for 2 minutes from now.
# This does, of course, assume you have powerd installed and your
# UPS connected and working correctly.
pf::powerfail:/sbin/shutdown -f -h +2 "Power Failure; System Shutting Down"

# If power was restored before the shutdown kicked in, cancel it.
pr:12345:powerokwait:/sbin/shutdown -c "Power Restored; Shutdown Cancelled"

# Run gettys in standard runlevels
1:2345:respawn:/sbin/mingetty tty1
2:2345:respawn:/sbin/mingetty tty2
#3:2345:respawn:/sbin/mingetty tty3
#4:2345:respawn:/sbin/mingetty tty4
#5:2345:respawn:/sbin/mingetty tty5
#6:2345:respawn:/sbin/mingetty tty6

# Run xdm in runlevel 5
x:5:respawn:/etc/X11/prefdm -nodaemon
```

→ inittab 파일 변경후 “init q” 실행 - 변경된 내용을 적용하게 하기 위해서

```
]# init q  
[root@www ~]#
```

- ⑫ init에 의해 /etc/rc.d/rc.sysinit 실행
  - /etc/rc.d/rc.sysinit 스크립트 : 시스템 초기화 스크립트
  - /etc/sysconfig/network 파일 실행
  - 커널 모듈 로드 등
  - fsck 로 파일 시스템 점검(/etc/fstab 파일 참조)
  - 호스트이름, 도메인 네임 설정
  - swapon 명령으로 스왑 작동
  - pnp 하드웨어 점검
  - /proc 파일 시스템 마운트
  - 기타 등등
- ⑬ init 에 의해 각 부팅 레벨별 스크립트 실행
  - 실행 레벨이 3번일 경우 /etc/rc.d/rc3.d/ 디렉토리
- ⑭ init 에 의해 /etc/rc.d/rc.local 파일 실행

## 2. GRUB (Grand Unified Boot Loader) 이해

- GRUB 은 다른 부트로더와 마찬가지로 하드디스크의 첫번째 부트 섹트인 MBR(Master Boot Record)라는 곳에 저장.
- grub 설치 확인

```
]# rpm -qa | grep grub  
grub-0.95-3.8
```

- grub 버전 확인

```
]# grub --version  
grub (GNU GRUB 0.95)
```

- grub.conf

```
]# more /boot/grub/grub.conf  
# grub.conf generated by anaconda  
#  
# Note that you do not have to rerun grub after making changes to this file  
# NOTICE: You have a /boot partition. This means that  
# all kernel and initrd paths are relative to /boot/, eg.  
# root (hd0,0)  
# kernel /vmlinuz-version ro root=/dev/hda2  
# initrd /initrd-version.img  
#boot=/dev/hda  
default=0  
timeout=5  
splashimage=(hd0,0)/grub/splash.xpm.gz  
hiddenmenu  
title Red Hat Enterprise Linux ES (2.6.9-55.EL)  
root (hd0,0)  
kernel /vmlinuz-2.6.9-55.EL ro root=LABEL=/1 rhgb quiet  
initrd /initrd-2.6.9-55.EL.img
```

→ initrd : scsi 디스크를 사용하는 경우에는 반드시 지정, 해당 메뉴로 부팅할 때에 사용할 initrd 이미지를 지정

Scsi 하드 디스크를 사용할 경우 initrd 항목이 설정되지 않으면 kernel panic 에 빠짐.

- grub 패스워드 설정
  - : c 키입력
  - : grub> md5crypt 입력

Password: \*\*\*\*\*

Encrypted: \$1\$EmStA0\$jaByn#gbwgaYlPlq0DMSw0

- 위와 가이 암호화된 문자들을 복사 하여 /etc/grub/grub-conf에 복사해서 넣음

```
]# more /boot/grub/grub.conf  
# grub.conf generated by anaconda  
#  
# Note that you do not have to rerun grub after making changes to this file  
# NOTICE: You have a /boot partition. This means that  
# all kernel and initrd paths are relative to /boot/, eg.
```

```
# root (hd0,0)
# kernel /vmlinuz-version ro root=/dev/hda2
# initrd /initrd-version.img
#boot=/dev/had
password -md5 $1$EmStA0$jiaByn#gbwgaYlpLq0DMSw0
default=0
timeout=5
splashimage=(hd0,0)/grub/splash.xpm.gz
hiddenmenu
title Red Hat Enterprise Linux ES (2.6.9-55.EL)
    root (hd0,0)
    kernel /vmlinuz-2.6.9-55.EL ro root=LABEL=/1 rhgb quiet
    initrd /initrd-2.6.9-55.EL.img
```

→ 리눅스 초기화면에서 “p”키를 눌러 패스워드 입력

- 리눅스 싱글 부팅