

[이 문서는 Ubuntu 16.04 기준으로 작성되었습니다.]

1. 리눅스의 부팅과정

A. ROM-BIOS의 실행

- 컴퓨터 전원이 켜지고 나면 가장 먼저 ROM-BIOS가 작동한다. ROM-BIOS는 컴퓨터에 어떤 하드웨어가 설치되어 있는지를 확인하고 이를 인식할 준비를 한다.
- 어떤 프로그램이라도 실행되기 위해서는 메모리에 올라가야 한다(이를 적재라고 한다). 전원이 켜지면 ROM에 저장된 BIOS 프로그램이 메모리에 적재된다.
- 먼저 POST기능이 작동한다. 이 과정에서 컴퓨터에 연결된 하드웨어 장치들을 하나씩 인식하고 이상유무를 체크한다. 만약 이상이 없다면 다음 작업으로 넘어간다.
- POST기능이 정상 종료하면 부트로더를 로딩한다. ROM-BIOS는 설치된 하드웨어에서 부팅이 가능한 매체를 뒤지고 가장 먼저 검색된 장치의 부트로더를 메모리에 적재하고 부트로더에게 권한을 넘긴다.

B. 부트로더(GRUB) 실행

- 리눅스의 기본 부트로더는 GRUB이다. 부트로더는 리눅스 커널을 적재하고 swapper 프로세스를 호출하는 역할을 한다.
- 먼저 리눅스 커널을 적재한다. 우분투 같은 경우는 /boot/grub/grub.cfg 파일에 어떻게 부팅을 할지에 대한 정보가 적혀 있다. 리눅스 커널은 리눅스의 가장 기본적인 기능을 담고 있는 운영체제의 일부분으로 이 파일이 메모리에 적재 되면 본격적으로 리눅스가 실행된다.
- 그 다음 swapper 프로세스를 실행한다.

C. Swapper 프로세스 수행

- 이 프로세스는 먼저 POST과정에서 인식 했던 각 하드웨어들의 드라이브들을 초기화 한다. 그리고 init 프로세스를 실행한 후 종료한다. 리눅스에서는 프로세스마다 번호를 부여 받는데, swapper는 0번이고 이는 부팅이 완료되면 종료되고, init은 1번으로 모든 프로세스의 부모가 되는 프로세스이다. 그래서 부팅된 리눅스에서 프로세스를 확인하면 1번부터 나오는 것을 볼 수 있다(ps aux 명령어)

D. init 프로세스 실행

- /etc/inittab의 설정파일을 읽어서 어떤 과정으로 초기화를 진행할 것인지를 확인하고 그 설정대로 작업을 진행한다.

E. 부팅 레벨의 결정

- 리눅스에는 부팅 레벨(run level)이라는 것이 존재한다.
 - 0번 : 종료
 - 1번 : 시스템 복원 모드(Single user mode)
 - 2번 : 네트워크를 사용하지 않는 multiuser mode CLI환경
 - 3번 : multiuser mode CLI환경
 - 4번 : 사용하지 않음
 - 5번 : multiuser mode X11 GUI환경
 - 6번 : 재부팅
- /etc/inittab에서 부팅레벨을 바꿀 수 있다. 우분투를 기본으로 설치 했다면 default run level은 5이다.

F. /etc/rc.d/rc.sysinit 실행

- 모든 run level에서 기본적으로 적용되는 초기화 스크립트이다.

G. /etc/rc.d/rcN.d/ 디렉토리의 스크립트 실행(N은 run level)

- 각 run level 별로 폴더가 만들어져 있고, 해당하는 run level의 디렉토리 내부의 내용을 적용한다.

H. 시스템 매직키 설정(Ctrl + Alt + Del)

I. 시스템 전원공급 설정

J. 가상 터미널과 로그인 창 실행

K. X윈도우 실행(run level 5 에서만)

2. 리눅스의 유저 관리 방법

A. 리눅스의 유저

- 우리는 리눅스에 로그인할 때 이름을 보고 비밀번호를 입력하지만, 리눅스에서는 이름이 아닌 숫자로 사용자를 구분한다.
- 사용자는 기본적으로 자신과 똑 같은 이름의 그룹에 속해 있다.
- 사용자는 동시에 여러 개의 그룹에 속할 수 있다.

B. 현재 시스템에 로그인해 있는 자신을 확인하기 위한 방법

- id 명령어

```
ksh@ubuntu:/boot/grub$ id
uid=1000(ksh) gid=1000(ksh) groups=1000(ksh),4(adm),24(cdrom),27(sudo),30(dip),46(plugdev),113(lpadmin),128(sambashare)
ksh@ubuntu:/boot/grub$
```

- id 명령어는 자신이 누구인지 uid, gid, groups의 정보를 통해 알려준다.
- id는 현재 로그인한 사용자를 기준으로 알려준다. 내가 ksh였는데 su 명령어를 사용해 root로 변경했다면 id 명령을 다시 실행하면 root의 정보를 알려준다.
- 위의 예시에서 ksh의 uid는 1000, gid는 1000, 속한 그룹은 ksh, adm, cdrom, sudo, dip, plugdev, lpadmin, sambashare 인 것을 확인할 수 있다.

- who 명령어

```
root@ubuntu:/boot/grub# who
ksh      tty7      2017-09-20 15:13 (:0)
root@ubuntu:/boot/grub#
```

- who 명령어는 현재 시스템에 어떤 사용자들이 로그인 했는지를 알려준다.
- 위의 예시에서 현재 로그인 해 있는 사용자는 ksh 이고 tty7 터미널을 사용했으며, 2017년 9월 20일 15시 13분에 접속한 것을 확인할 수 있다.

C. 리눅스 계정 관리 파일

- 리눅스의 계정은 두 가지 파일에 의해 관리된다.
- /etc/passwd

```

games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin)/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-timesync:x:100:102:systemd Time Synchronization,,,:/run/systemd:/bin/false
systemd-network:x:101:103:systemd Network Management,,,:/run/systemd/netif:/bin/false
systemd-resolve:x:102:104:systemd Resolver,,,:/run/systemd/resolve:/bin/false
systemd-bus-proxy:x:103:105:systemd Bus Proxy,,,:/run/systemd:/bin/false
syslog:x:104:108:./home/syslog:/bin/false
_apt:x:105:65534:./nonexistent:/bin/false
messagebus:x:106:110:./var/run/dbus:/bin/false
uuidd:x:107:111:./run/uuidd:/bin/false
lightdm:x:108:114:Light Display Manager:/var/lib/lightdm:/bin/false
whoopsie:x:109:117:./nonexistent:/bin/false
avahi-autoipd:x:110:119:Avahi autoip daemon,,:/var/lib/avahi-autoipd:/bin/false
avahi:x:111:120:Avahi mDNS daemon,,:/var/run/avahi-daemon:/bin/false
dnsmasq:x:112:65534:dnsmasq,,:/var/lib/misc:/bin/false
colord:x:113:123:colord colour management daemon,,:/var/lib/colord:/bin/false
speech-dispatcher:x:114:29:Speech Dispatcher,,:/var/run/speech-dispatcher:/bin/false
hplip:x:115:7:HPLIP system user,,:/var/run/hplip:/bin/false
kernoops:x:116:65534:Kernel Oops Tracking Daemon,,:/bin/false
pulse:x:117:124:PulseAudio daemon,,:/var/run/pulse:/bin/false
rtkit:x:118:126:RealtimeKit,,:/proc:/bin/false
saned:x:119:127:./var/lib/saned:/bin/false
usbmux:x:120:46:usbmux daemon,,:/var/lib/usbmux:/bin/false
ksh:x:1000:1000:Ubuntu_16.04,,:/home/ksh:/bin/bash
ntp:x:121:129:./home/ntp:/bin/false
root@ubuntu:/boot/grub#

```

- 각 필드에 대한 설명(각 필드는 : 으로 구분한다)

기준은 ksh로 한다.

ksh : x : 1000 : 1000 : Ubuntu_16.04, , , /home/ksh : /bin/bash

- 첫 번째 필드 (ksh) : 사람이 인식하는 계정의 이름이다.
- 두 번째 필드 (x) : 원래는 계정의 비밀번호가 적혀 있는 필드인데 /etc/passwd라는 파일이 모든 유저에게 읽기 권한이 있어 여기에 비밀번호를 저장하면 보안상 취약하기 때문에 원래의 비밀번호는 암호화 하여 /etc/shadow에 저장하고 여기에는 x 표시로 필드만 구분해 둔다.
- 세 번째 필드 (1000) : uid를 말한다. 리눅스는 이 번호로 사용자가 ksh임을 확인한다.
- 네 번째 필드 (1000) : gid를 말한다. 리눅스는 이 번호로 그룹이 ksh임을 확인한다.
- 다섯 번째 필드 (Ubuntu_16.04) : 그 사용자에 대해 설명하는 comment이다.
- 여섯 번째 필드 (/home/ksh) : 해당 사용자의 홈 디렉터리의 경로를 말한다.
- 일곱 번째 필드 (/bin/bash) : 해당 사용자가 사용할 기본 셸의 경로를 지정한다.

- /etc/shadow

```
games:*:17379:0:99999:7:::
man:*:17379:0:99999:7:::
lp:*:17379:0:99999:7:::
mail:*:17379:0:99999:7:::
news:*:17379:0:99999:7:::
uucp:*:17379:0:99999:7:::
proxy:*:17379:0:99999:7:::
www-data:*:17379:0:99999:7:::
backup:*:17379:0:99999:7:::
list:*:17379:0:99999:7:::
irc:*:17379:0:99999:7:::
gnats:*:17379:0:99999:7:::
nobody:*:17379:0:99999:7:::
systemd-timesync:*:17379:0:99999:7:::
systemd-network:*:17379:0:99999:7:::
systemd-resolve:*:17379:0:99999:7:::
systemd-bus-proxy:*:17379:0:99999:7:::
syslog:*:17379:0:99999:7:::
_apt:*:17379:0:99999:7:::
messagebus:*:17379:0:99999:7:::
uidd:*:17379:0:99999:7:::
lightdm:*:17379:0:99999:7:::
whoopsie:*:17379:0:99999:7:::
avahi-autoipd:*:17379:0:99999:7:::
avahi:*:17379:0:99999:7:::
dnsmasq:*:17379:0:99999:7:::
colord:*:17379:0:99999:7:::
speech-dispatcher:!:17379:0:99999:7:::
hplip:*:17379:0:99999:7:::
kernoops:*:17379:0:99999:7:::
pulse:*:17379:0:99999:7:::
rtkit:*:17379:0:99999:7:::
saned:*:17379:0:99999:7:::
usbmux:*:17379:0:99999:7:::
ksh:$1$chnSRP8C$eTcfLtKu0MCHW8L.xm4m1.:17416:0:99999:7:::
ntp:*:17423:0:99999:7:::
root@ubuntu:/boot/grub#
```

- 각 필드에 대한 설명(필드 구분은 : 이다)

기준은 ksh로 한다.

ksh : \$1\$chnSRP8C\$eTcfLtKu0MCHW8L.xm4m1. : 17416 : 0 : 99999 : 7 :::

- 첫 번째 필드 (ksh) : 사용자의 이름이다.
- 두 번째 필드 (\$로 시작하는 문자열) : 비밀번호를 hash 방식으로 암호화 한 결과이다.
- 세 번째 필드 (17416) : 1970년 1월 1일로부터 패스워드 파일이 최종 수정된 날짜의 일수

- 네 번째 필드 (0) : 패스워드를 변경하기 위한 최소 일 수(0일이면 이 설정을 하지 않는다)
 - 다섯 번째 필드 (99999) : 패스워드 사용 만기일(99999는 사실상 설정하지 않은 것이다)
 - 여섯 번째 필드 (7) : 패스워드 사용 만기일 전에 경고 메시지를 제공하는 일 수. 이 경우 7일 전부터 경고하게 된다.
 - 일곱 번째 필드 (미설정) : 패스워드가 만료되면 로그인 접속을 차단할 일 수
 - 여덟 번째 필드 (미설정) : 계정 만료 일
 - 아홉 번째 필드 (미설정) : 사용하지 않는다.
- 사용자 계정에 대한 기본적인 정보
- 일반 사용자 계정은 특별히 지정하지 않는 이상 1000번부터 uid가 부여된다. 1000번 미만은 시스템에 필요한 계정들이 할당 받는다.
 - 리눅스가 사용자를 uid로 인식한다는 특성 때문에 일반 사용자의 uid를 root의 uid인 0으로 변경하면 리눅스는 일반 사용자를 root로 인식한다.
 - 해시 암호 방식은 어떤 문자열을 해시 함수에 집어 넣으면 항상 고정된 길이의 암호화된 문자열을 출력하는 방식을 말한다.
 - 셸은 사용자가 명령어를 입력했을 때 이를 해석할 수 있는 프로그램을 말한다.
 - 위의 /etc/passwd 파일에서 셸이 nologin, false인 경우는 로그인 할 수 없는 계정이라고 생각하면 된다.

D. 셸(Shell)

- 사용자가 입력하는 명령어를 해석하여 요청한 작업을 처리하는 프로그램이다. 셸의 내부 명령어라면 자기가 실행하고, 자신의 명령어가 아니라면 PATH 환경변수에 지정된 경로에서 해당 명령어의 파일을 찾아 실행해준다.
- 셸은 스크립트 언어(명령어를 입력하면 컴파일 과정 없이 이를 실행)이기 때문에, 셸에서 사용되는 명령어들과 함수를 이용해 간단한 프로그램을 짤 수도 있다.
- 환경변수
 - 셸에 기본적으로 설정된, 혹은 사용자가 설정한 변수들을 말한다.
 - 셸에 설정된 환경변수를 확인하기 위해서는 env 명령(bash 셸의 경우)을 입

력하면 된다.

- 셸의 종류

- 본 셸(/bin/sh) : AT&T의 벨 연구소의 스티브 본이 개발하였다.
- 배시 셸(/bin/bash) : 브라이언 폭스가 개발한 셸로 현재는 리눅스에 기본 셸이 되었다. 본 셸의 많은 기능이 호환된다.
- C 셸(csh) : 빌 조이가 개발한 셸이다.
- 콘 셸(ksh) : 벨 연구소의 데이비드 콘이 개발한 셸이다. C 셸의 많은 기능이 호환된다.

E. 계정의 전환

- su 명령어는 로그오프를 하지 않고도 다른 계정으로 로그인할 수 있는 기능을 제공한다.

- 사용법

■ su

- ◆ 루트의 비밀번호를 입력하면 루트로 로그인할 수 있다. 환경변수는 su를 입력한 사용자의 것을 적용한다.

■ su -

- ◆ 루트의 비밀번호를 입력하면 루트로 로그인할 수 있다. 환경변수는 루트의 것을 적용한다.

■ su - username

- ◆ username 사용자로 로그인한다. 환경변수는 username의 것을 적용한다.

■ su -c '명령어' - root

- ◆ 루트의 권한으로 해당 명령어를 실행한다. 계정을 변경하지는 않는다.

■ su -s '셸 경로' - username

- ◆ username으로 로그인하는데 셸은 '셸 경로'를 사용한다.

F. 계정의 관리

- 계정의 생성을 위해 해야 하는 일

- /etc/passwd 파일에 사용자의 계정 생성

- /etc/shadow 파일에 계정의 비밀번호 등록
 - /etc/group 파일에 사용자의 그룹 생성
 - /home에 사용자의 홈 디렉터리 생성
 - 그 외 서비스를 이용하기 위한 서비스 설정(FTP, Mail 등)
 - 사용자의 로그인정보 및 환경설정
- useradd
- 사용자의 계정을 생성하는 명령어이다.
 - 구체적 사용 법
 - ◆ useradd icewall : icewall 계정을 만든다.
 - ◆ passwd icewall : icewall 계정의 비밀번호를 설정한다.
 - ◆ 위와 같이 사용하면 아마 홈 디렉터리와 로그인 셸이 설정되지 않을 것이다(원래 리눅스에서는 default로 홈 디렉터리를 /home/icewall, 로그인 셸을 /bin/bash로 정해주는데 그렇지 않을 것이다. 그 이유는 뒤에서 확인한다).
 - 옵션을 적용하여 계정 생성하기
 - ◆ 옵션의 종류
 - -u uid(숫자) : uid의 설정
 - -g gid(숫자) : gid의 설정
 - -G groupname: 추가적으로 속할 그룹 설정
 - -e 2017-10-31 : 계정의 사용종료일자 설정
 - -m -d /home/icewall : -d는 홈 디렉토리의 지정. -m은 그 홈 디렉토리를 실제로 만드는 옵션
 - -s /bin/bash : 로그인 셸의 임의 지정
 - -c comment : 계정에 대한 간단한 설명 설정
 - user -D 옵션


```
#
# Defines whether the mail spool should be created
# creating the account
# CREATE_MAIL_SPOOL=yes

root@ubuntu:/etc/skel# useradd -D
GROUP=100
HOME=/home
INACTIVE=-1
EXPIRE=
SHELL=/bin/sh
SKEL=/etc/skel
CREATE_MAIL_SPOOL=no
root@ubuntu:/etc/skel#
```

- ◆ /etc/default/useradd의 내용을 출력하여 보여준다.
- ◆ 위의 파일은 useradd가 사용자를 만들 때 기본적으로 적용하는 옵션을 설정해 놓은 파일이다.
- ◆ useradd -D -s /bin/bash 로 하면 위의 파일의 SHELL을 /bin/bash로 변경해준다. 그러면 다음에 사용자를 만들면 기본적으로 /bin/bash의 로그인 셸을 갖게 된다.
- ◆ 같은 방법으로 -D -g gid 는 기본 그룹을 바꾸고, -D -b '경로'는 홈 디렉터리가 만들어질 폴더를 바꾼다.

■ /etc/skel

- ◆ useradd로 사용자를 만들 때 사용자의 홈 디렉터리에 기본적으로 포함될 폴더 및 파일들이 들어있는 폴더이다.

■ /etc/default/useradd

- ◆ useradd가 사용자를 생성할 때 참조하는 파일
- ◆ GROUP : 기본등록 그룹의 GID
- ◆ HOME : 홈 디렉터리가 생성될 위치
- ◆ INACTIVE : 패스워드 종료일 이후의 유효여부 설정. 이 값이 -1이면 기능을 비활성화 하고(패스워드 종료 후에도 패스워드를 무효화 하지 않는다), 0이면 패스워드 종료일이 되면 바로 계정을 잠그고, 1 이상의 값을 가지면 그 값의 일자 만큼 패스워드를 유효하게 한다.

■ /etc/login.defs

- ◆ useradd가 사용자를 생성할 때 참조하는 파일 2

◆ 내용

- MAIL_DIR : 메일 디렉터리를 지정한다.
- PASS_MAX_DAYS : 마지막으로 비밀번호를 변경한 뒤 비밀번호가 유효한 기간을 설정한다.
- PASS_MIN_DAYS : 마지막으로 비밀번호를 변경한 뒤 다시 변경할 수 있는 기간을 설정한다.
- PASS_MIN_LEN : 인정될 수 있는 최소 비밀번호 길이를 말한다.
- PASS_WARN_AGE : 비밀번호 만료 며칠전부터 경고를 할 지 설정한다.
- UID_MIN : 새로 생성되는 사용자에게 할당할 UID 시작번호를 설정한다.
- UID_MAX : 새로 생성되는 사용자에게 할당할 UID의 마지막 번호를 설정한다.
- GID_MIN : 새로 생성되는 그룹의 GID 시작번호를 설정한다.
- GID_MAX : 새로 생성되는 그룹의 GID 마지막 번호를 지정한다.
- CREATE_HOME : 홈 디렉터리를 만들 것인지의 여부를 결정한다.
- UMASK : umask 값을 지정한다. 지정하지 않으면 기본 022이다.
- USERGROUPS_ENAB : yes이면 userdel 실행 시 멤버가 없는 그룹도 같이 삭제된다.
- MD5_CRYPT_ENAB : 패스워드 암호화를 위하여 MD5를 사용한다.
- ENCRYPT_METHOD SHA512 : 암호화 방법으로 SHA512를 사용한다.

- passwd

- 사용자의 비밀번호를 설정하고 변경하는 명령어이다. root 사용자는 모든 사용자의 비밀번호를 변경할 수 있고, 각 사용자는 자신의 비밀번호만을 변경할 수 있다.
- 사용례
 - ◆ passwd icewall : root가 icewall 사용자의 비밀번호를 변경하는 명령이다.
 - ◆ passwd : 각 사용자가 자신의 비밀번호를 변경하는 명령이다.

- ◆ `passwd -S icewall` : root만 가능한 명령으로, icewall 계정의 비밀번호에 대한 정보를 확인할 수 있다.
- ◆ `passwd -l icewall` : icewall 계정의 비밀번호를 잠근다. 이 명령 이후 icewall 사용자는 로그인 할 수 없다.
- ◆ `passwd -u icewall` : icewall 계정에 걸려있는 lock을 푼다.

```
root@ubuntu:/home/ksh# passwd -l ksh
passwd: password expiry information changed.
root@ubuntu:/home/ksh# grep ksh /etc/passwd
ksh:x:1000:1000:Ubuntu_16.04,,,:/home/ksh:/bin/bash
root@ubuntu:/home/ksh# grep ksh /etc/shadow
ksh:!!$1$chnSRP8C$eTcfLtKu0MCHW8L.xm4m1.:17416:0:99999:7:::
root@ubuntu:/home/ksh#
```

- ◆ lock이 걸리면 shadow파일의 비밀번호 앞에 !가 붙게 된다. 물론 unlock을 하면 !는 사라진다.
- ◆ `passwd -d icewall` : icewall 계정의 비밀번호를 삭제한다. 해당 사용자는 root가 비밀번호를 재 설정 할 때 까지 로그인 할 수 없다.

- chage

- 사용자의 aging 정보란 비밀번호에 관련된 일자들을 말한다. 그 종류로는 계정의 사용기간, 비밀번호 유효기간, 경고날짜, 종료일, 길이 등이 있다.
- 사용자 패스워드의 각종 aging 정보(만기일 같은)를 관리하는 명령어이다. chage로 aging정보를 바꾸면 /etc/shadow 파일에 반영된다.
- 옵션
 - ◆ `-d 18000` : 1970년 1월 1일로부터 측정한 패스워드를 마지막으로 변경한 일자를 변경
 - ◆ `-E 2017-10-31` : 계정 사용 종료일자 설정
 - ◆ `-I(대문자 i) 20` : 패스워드 유효기간 종료 이후 계정이 비활성화 될 일 수
 - ◆ `-m 10` : 패스워드를 변경하기 위한 최소 일 수(한 번의 패스워드 변경 이후 10일 간은 패스워드 변경이 불가하다)
 - ◆ `-M 30` : 마지막 변경일 이후 패스워드 유효일 수(마지막 패스워드 변경 후 30일간 유효하다)
 - ◆ `-W 7` : 패스워드 만료일 전에 경고 메시지를 띄울 날 수(만료 7일 전부터 경고 메시지를 띄운다)

◆ -(소문자 L) icewall : icewall의 aging정보를 출력한다.

- usermod

- 기본적인 옵션은 useradd와 같다. 다만, usermod는 그 정보를 변경하는 역할을 한다.

- userdel

- 사용자를 삭제하는 명령이다.

- 사용 예

◆ userdel icewall : icewall 계정을 삭제한다. 단, 홈 디렉터리, 메일 파일 등은 그대로 남아있다.

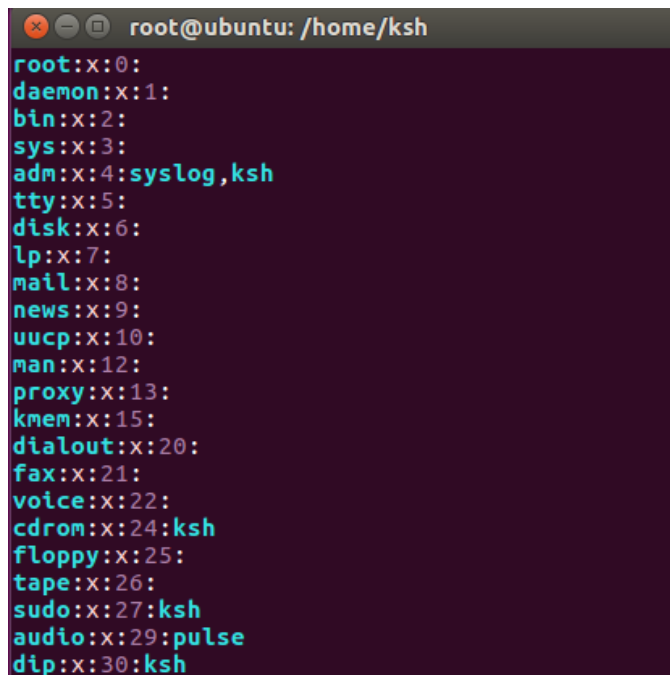
◆ userdel -r icewall : icewall 계정을 삭제한다. 홈 디렉터리, 메일 파일 등을 모두 함께 지운다.

G. 그룹 관리

- groupadd

- 새로운 그룹을 생성하는 명령어이다.

- /etc/group



```
root@ubuntu: /home/ksh
root:x:0:
daemon:x:1:
bin:x:2:
sys:x:3:
adm:x:4:syslog,ksh
tty:x:5:
disk:x:6:
lp:x:7:
mail:x:8:
news:x:9:
uucp:x:10:
man:x:12:
proxy:x:13:
kmem:x:15:
dialout:x:20:
fax:x:21:
voice:x:22:
cdrom:x:24:ksh
floppy:x:25:
tape:x:26:
sudo:x:27:ksh
audio:x:29:pulse
dip:x:30:ksh
```

◆ 생성된 그룹들을 관리하는 파일이다.

root : x : 0 :

첫 번째 필드 (root) : 계정의 이름이다

두 번째 필드 (x) : 계정의 비밀번호이다. 거의 사용하지 않는다

세 번째 필드 (0) : gid이다.

네 번째 필드 (미 설정) : 해당 그룹에 속한 유저들이다.

■ 명령어 사용 예

- ◆ groupadd -r icewall : 기본적으로 부여되는 gid를 역순으로 부여하여 icewall 그룹을 만든다.

- 그룹에 새로운 사용자 추가하기

■ /etc/group 파일을 편집하기

- ◆ /etc/group파일에 원하는 그룹의 네 번째 필드에 추가할 사용자를 적어주면 추가가 완료된다.

■ gpasswd 명령어 사용하기

- ◆ gpasswd -a icewall hanyang
hanyang 그룹에 icewall이라는 사용자를 추가한다.

- 그룹에서 사용자 삭제하기

■ /etc/group 파일을 편집하기

■ gpasswd 명령어 사용하기

- ◆ gpasswd -d icewall hanyang
hanyang 그룹에서 icewall 사용자를 제거한다.

H. 루트 권한의 일시적 부여. sudo

- sudo '명령어'

- 위의 명령을 터미널에 입력하면 현재 사용자의 비밀번호를 물어보고 맞다면 해당 명령어를 root권한으로 실행시켜 준다.
- useradd로 새로운 user를 만들면 이 user는 sudo 명령어를 사용할 수 없다. 왜냐하면 sudo 관리 파일에 등록이 되어있지 않기 때문이다.
- sudo를 사용할 수 있는 유저, 그룹, 혹은 sudo를 통한 명령어의 제한 등은 /etc/sudoers라는 파일에서 하게 된다.

- /etc/sudoers

```

root@ubuntu: /home/ksh
#
Defaults    env_reset
Defaults    mail_badpass
Defaults    secure_path="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/
sbin:/bin:/snap/bin"

# Host alias specification

# User alias specification

# Cmnd alias specification

# User privilege specification
root    ALL=(ALL:ALL) ALL
# Members of the admin group may gain root privileges
%admin  ALL=(ALL) ALL

# Allow members of group sudo to execute any command
%sudo   ALL=(ALL:ALL) ALL

# See sudoers(5) for more information on "#include" directives:

#includedir /etc/sudoers.d

```

- /etc/sudoers 파일은 sudo 권한을 부여하는 방법을 지정하는 파일로서 기본 퍼미션이 440으로 되어있다. 따라서 root또한 이를 변경할 수 없다.
- 따라서 파일을 변경하고자 한다면 먼저 `sudo su` 명령으로 자신의 비밀번호를 치고 들어가서(시스템 설치 시에 제일 처음 만들어진 계정은 기본적으로 `sudoers` 파일에 포함 되어있다), `chmod 640 /etc/sudoers`를 입력하여 `sudoers`파일을 쓸 수 있게 바꾼다.
- 권한 부여 예시(각 필드는 탭으로 구분한다).
 - ◆ `%admin ALL=(ALL) ALL`
admin 그룹에 `sudo` 로 모든 명령을 실행할 권한을 부여한다.
 - ◆ `icewall ALL=/bin/cat /etc/shadow`
icewall 유저에게 `sudo` 명령으로 `cat /etc/shadow`를 할 권한만 부여한다. 다른 명령어는 쓸 수 없다.
 - ◆ `%hanyang ALL=(ALL) NOPASSWD: ALL`
hanyang 그룹에 `sudo`로 모든 명령을 실행할 권한을 부여하되 비밀번호를 입력하지 않아도 된다.
- `sudoers` 파일의 변경을 마쳤으면 `chmod 440 /etc/sudoers` 를 입력하여 다시 권한을 변경한다.
- `su` 명령과 `sudo` 명령의 차이
 - `su` 명령은 현재 터미널에서 해당 사용자로 변경하는 것으로서 변경하고자 하는 사용자의 비밀번호를 필요로 한다.
 - `sudo` 명령을 통하면 현재 접속하고 있는 계정의 비밀번호로 root 권한의 명

령어를 실행할 수 있다.

- su 명령으로 root에 접속하면 root의 모든 권한을 누릴 수 있다.
 - sudo 명령은 root 권한으로 명령어만 실행할 수 있다.
- sudo 명령을 쓰는 이유
- 어떤 사용자에게는 root만 사용할 수 있는 명령어의 일부를 사용할 수 있도록 해줄 필요가 있다. 그렇다고 해서 root의 모든 권한을 다 주게 되면 시스템 보안상 문제가 발생한다.
예를 들면, 사용자 계정 관리를 계정 관리자를 따로 두어 하게 하는 경우, 계정 관리자는 useradd usermod passwd를 사용할 수 있어야 한다. 대신 다른 명령어는 막아 두어야 한다.
 - su 명령어로 이를 사용하게 하면 root의 비밀번호가 노출될 위험이 있다.
 - 따라서 root의 비밀번호를 보호하고, 시스템의 제한된 권한을 부여하기 위해서 sudo를 사용한다.