

시작과 종료

- # 프롬프트는 root 사용자
- \$ 프롬프트는 일반 사용자

• 종료하는 방법

- ① 바탕 화면의 [▼ 아이콘] → [컴퓨터 끄기/로그아웃] → <컴퓨터 끄기>
- ② 터미널/콘솔에서 시스템 종료 명령 입력
`poweroff, shutdown -p now, halt -p, init 0`

• 시스템 재부팅

- ① 바탕 화면의 [▼ 아이콘] → [컴퓨터 끄기/로그아웃] → <다시 시작>
- ② 터미널/콘솔에서 시스템 재부팅 명령 입력
`shutdown -r now, reboot, init 6`

• 로그아웃

- ① 바탕 화면의 [▼ 아이콘] → [컴퓨터 끄기/로그아웃] → [로그아웃]
- ② 터미널/콘솔에서 시스템 종료 명령 입력
`logout 또는 exit`

가상 콘솔

- 쉽게 '가상의 모니터' 라 생각하면 됨. 우분투는 총 6개의 가상 콘솔을 제공.
- 각각의 가상 콘솔로 이동하는 단축키는 Ctrl+ Alt + F2 ~ F7 (Ctrl + Alt + F2은 X윈도우 모드)



<실습1> 여러 명의 사용자가 동시 접속

- 실습목표
 - 멀티유저의 환경을 실습해 본다.
 - 가상 콘솔 기능을 활용해 본다.

- 이번 실습은 Windows 버전/VMware 버전/키보드 종류에 따라서 잘 수행되지 않을 수 있음.
- 대신 cnvt 가상콘솔번호 명령을 사용해도 됨.

런 레벨(Runlevel)

- 'init' 명령어 뒤에 붙는 숫자를 런레벨(RunLevel)이라고 부른다.

런레벨	영문 모드	설명	비고
0	Power Off	종료 모드	
1	Rescue	시스템 복구 모드	단일 사용자 모드
2	Multi-User		사용하지 않음
3	Multi-User	텍스트 모드의 다중 사용자 모드	
4	Multi-User		사용하지 않음
5	Graphical	그래픽 모드의 다중 사용자 모드	
6	Reboot		

- 런레벨 모드를 확인하려면 `/lib/systemd/system` 디렉터리의 `runlevel?.target` 파일을 확인



<실습2> 시스템에 설정된 런레벨을 변경

- 실습목표
 - 런레벨의 개념을 파악한다.
 - 런레벨을 변경하는 방법을 파악한다.
- 실습화면(런레벨 3로 변경)

```
root@server: ~/바탕화면
root@server:~/바탕화면# ln -sf /lib/systemd/system/multi-user.target /lib/systemd/system/default.target
root@server:~/바탕화면#
root@server:~/바탕화면# ls -l /lib/systemd/system/default.target
lrwxrwxrwx 1 root root 37 5월 4 08:33 /lib/systemd/system/default.target -> /lib/systemd/system/multi-user.target
root@server:~/바탕화면#
```

자동 완성과 히스토리

- 자동 완성이란 파일명의 일부만 입력한 후에 Tab키를 눌러 나머지 파일명을 자동으로 완성하는 기능을 말함.

예) `cd /etc/NetworkManager/dispatcher.d/` 를 입력하려면
`cd /et[Tab키]Ne[Tab키]dis[Tab키]`

➤ 자동 완성기능은 빠른 입력효과도 있지만, 파일명이나 디렉터리가 틀리지 않고 정확하게 입력되는 효과도 있으므로 자주 활용된다.

- 도스 키란 이전에 입력한 명령어를 상/하 화살표 키를 이용해서 다시 나타내는 기능을 말함.

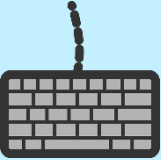


<실습3> 자동 완성과 도스키

- 실습목표
 - 자동 완성 기능과 도스 키 기능을 익힌다.
 - `history` 명령어의 기능을 확인한다.
- 실습화면(`history` 명령어)

```
root@server: ~/바탕화면
root@server:~/바탕화면# history
 1  passwd
 2  clear
 3  nano  /etc/gdm3/custom.conf
 4  nano  /etc/pam.d/gdm-password
 5  nano  /etc/pam.d/gdm-autologin
 6  reboot
 7  cd    /etc/apt/
 8  ls
 9  cat   sources.list.
10  cat   sources.list.bak
11  cd    /etc/apt
12  mv    sources.list sources.list.bak
```

에디터 사용



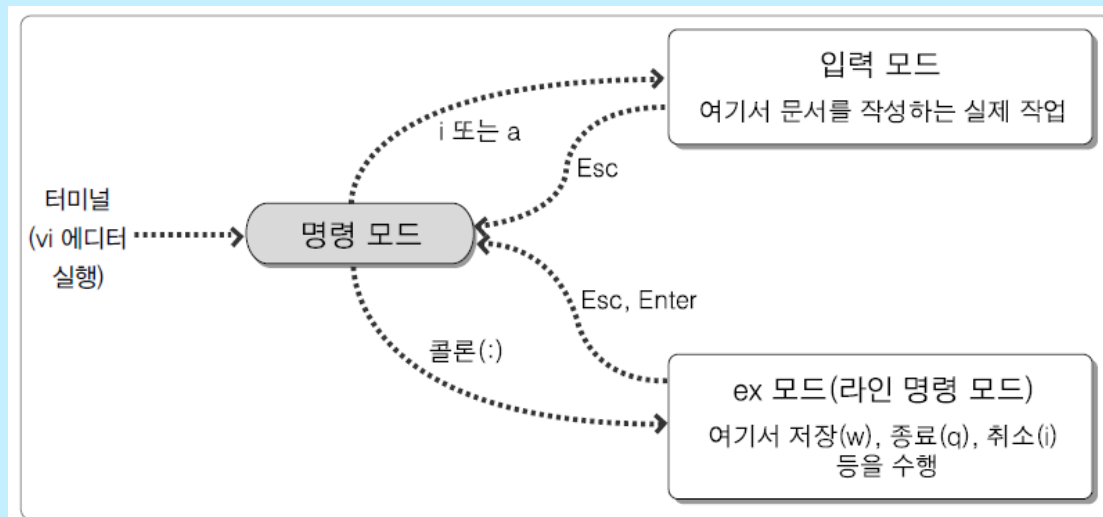
<실습4> 에디터를 사용하자

- 실습 목표

- `gedit`의 기본적인 사용법을 확인한다.
- `nano` 에디터의 사용법을 익힌다.
- `vi`의 사용법을 연습한다.

- `vi` 에디터 사용법 개요도

➤ `vi` 나 `nano`는 자주 사용해야 할 기능이므로 반드시 익혀야 한다.





<실습5> vi의 비정상 종료시 조치법

- 실습목표
 - vi 에디터가 비정상적으로 종료 시에 조치법을 익힌다.
- 실습 화면 (vi의 비정상 종료 후 다시 열었을 때)

```
root@server: ~  
  
E325: ATTENTION  
Found a swap file by the name ".new.txt.swp"  
    owned by: root   dated: 월  5월 04 17:31:24 2020  
    file name: ~root/new.txt  
    modified: YES  
    user name: root   host name: server  
to free process ID: changes (see "help" for "undo" and "recover")  
If you did this already, delete the swap file ".new.txt.swp"  
to avoid this message.  
"new.txt" 8 lines, 156 characters  
Press ENTER or type command to continue
```


vi 기능 요약

• 명령모드 → 입력모드

i	현재 커서의 위치부터 입력([I])	I	현재 커서 줄의 맨 앞에서부터 입력([Shift] + [I])
a	현재 커서의 위치 다음 칸부터 입력([A])	A	현재 커서 줄의 맨 마지막부터 입력([Shift] + [A])

• 명령 모드에서 커서를 이동

h	커서를 왼쪽으로 한 칸 이동([H]와 같은 의미, [H])	j	커서를 아래로 한 칸 이동([J]와 같은 의미, [J])
k	커서를 위로 한 칸 이동([K]와 같은 의미, [K])	l	커서를 오른쪽으로 한 칸 이동([L]와 같은 의미, [L])

• 명령 모드에서 삭제, 복사, 붙여넣기

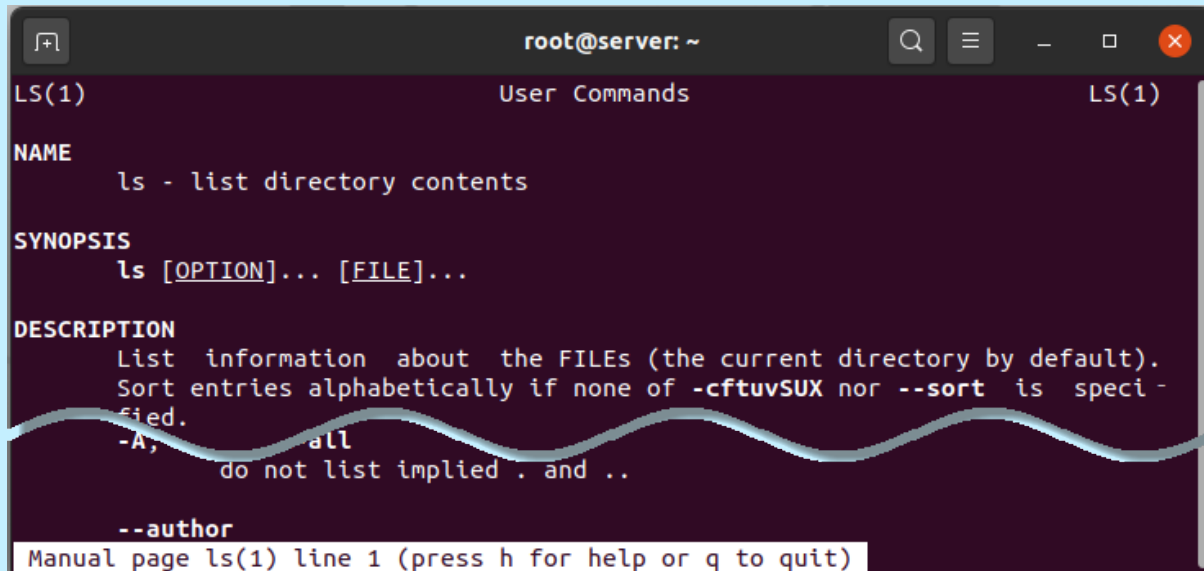
x	현재 커서가 위치한 글자 삭제([Del]과 같은 의미, [X])	X	현재 커서가 위치한 앞 글자 삭제([BackSpace]와 같은 의미, [Shift] + [X])
dd	현재 커서의 행 삭제([D] 연속 두 번 입력)	숫자 dd	현재 커서부터 숫자만큼의 행 삭제(숫자 다음 [D] 연속 두 번 입력)
yy	현재 커서가 있는 행을 복사([Y] 연속 두 번 입력)	숫자 yy	현재 커서부터 숫자만큼의 행을 복사(숫자 다음 [Y] 연속 두 번 입력)
p	복사한 내용을 현재 행 이후에 붙여 넣기([P])	P	복사한 내용을 현재 행 이전에 붙여 넣기([Shift] + [P])

- 문자열을 치환은 ":%s/기존문자열/새문자열"
- 행번호 표시는 ":set number"

도움말 사용법

➤ man 페이지는 섹션 1~9까지로 나뉩니다

- "man 명령어" 를 사용하면 도움말 출력



```
root@server: ~
LS(1)                                User Commands                                LS(1)

NAME
  ls - list directory contents

SYNOPSIS
  ls [OPTION]... [FILE]...

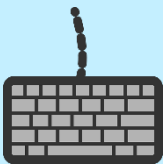
DESCRIPTION
  List information about the FILES (the current directory by default).
  Sort entries alphabetically if none of -cftuvSUX nor --sort is speci-
  fied.
  -A, --all                        list all, including hidden files
                                   do not list implied . and ..

  --author

Manual page ls(1) line 1 (press h for help or q to quit)
```

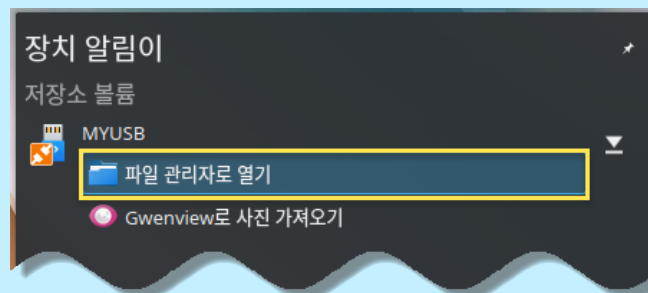
마운트와 CD/DVD, USB의 활용

- 물리적인 장치를 특정한 위치(대개는 디렉터리)에 연결 시켜 주는 과정



<실습6> CD/DVD 마운트

- 실습목표
 - X 윈도우에서 CD/DVD 와 USB 사용법을 익힌다.
 - 텍스트모드에서 CD/DVD 와 USB 사용법을 익힌다.
- 실습 화면 (X윈도우와 텍스트모드에서 USB 마운트)



```
tmpfs on /run/udev/tmpfs (rw,relatime,size=10M)
/var/lib/flatpak/snapshots/flatpak on /snap/flatpak/14954 type squashfs (ro,nodev,relatime,x-gdu.hide)
nsfs on /run/flatpak/nsfs/flatpak type nsfs (rw)
/dev/sr0 on /media/cdrom type iso9660 (ro,relatime,nojoliet,check=s,map=n,blocksize=2048)
/dev/sdb1 on /media/usb type vfat (rw,relatime,fmask=0022,dmask=0022,codepage=437,iocharset=iso8859-1,shortname=mixed,errors=remount-ro)
root@server-b:~#
root@server-b:~# ls /media/cdrom
boot  dists  install  md5sum.txt  pool  README.diskdefines
casper  EFI  isolinux  pics  preseed  ubuntu
root@server-b:~#
root@server-b:~# ls /media/usb
config-5.4.0-26-generic  sources.list
config-5.4.0-28-generic  'System Volume Information'
ntfs-3g_ntfsprogs-2017.3.23.tgz  VMware-workstation-full-15.5.0-14665864.exe
root@server-b:~#
```



<실습7> 리눅스에서 ISO 파일을 제작

- 실습 목표
 - CD 레코딩 명령어 `genisoimage` 사용법을 익힌다.
 - ISO 파일을 리눅스에서 마운트 해서 사용하는 방법을 익힌다.
- 실습 화면 (ISO 파일 생성)

```
root@server: ~/바탕화면# cd
root@server: ~#
root@server: ~# genisoimage -r -J -o boot.iso /boot
I: -input-charset not specified, using utf-8 (detected in locale settings)
Using MEMTE000.BIN;1 for /memtest86+.bin (memtest86+_multiboot.bin)
Using GFXTE000.MOD;1 for /boot/grub/i386-pc/gfxterm_menu.mod (gfxterm_background.mod)
Using USBSE000.MOD;1 for /boot/grub/i386-pc/usbserial_ftdi.mod (usbserial_sl2303.mod)
Total directory size: 44840
Path table size(bytes): 52
Max brk space used 44000
36222 extents written (70 MB)
root@server: ~#
```

리눅스 기본 명령어 (1)

- **ls**

Windows의 "dir" 과 같은 역할로, 해당 디렉터리에 있는 파일의 목록을 나열

예) # ls /etc/systemd

- **cd**

디렉터리를 이동

예) # cd ../etc/systemd

- . (현재 디렉터리)
- .. (현재의 상위 디렉터리)

- **pwd**

현재 디렉터리의 전체 경로를 출력

- **rm**

파일이나 디렉터를 삭제

예) # rm -rf abc

- 리눅스는 별도의 숨김 파일(Hidden File)이라는 속성이 존재하지 않는다, 파일명이나 디렉터리의 제일 앞 글자를 . 으로 하면 자동으로 숨김 파일이 된다.

리눅스 기본 명령어 (2)

- **cp**

파일이나 디렉터리를 복사

예) # `cp abc.txt cba.txt`

- **touch**

크기가 0인 새 파일을 생성, 이미 존재하는 경우 수정 시간을 변경

예) # `touch abc.txt`

- **mv**

파일과 디렉터리의 이름을 변경하거나 위치 이동 시 사용

예) `mv abc.txt www.txt`

- **mkdir**

새로운 디렉터리를 생성

예) # `mkdir abc`

리눅스 기본 명령어 (3)

- **rm**

디렉토리를 삭제, (단, 비어 있어야 함)

예) # `rm -r abc`

- **cat**

텍스트로 작성된 파일을 화면에 출력

예) # `cat a.txt b.txt`

- **head, tail**

텍스트로 작성된 파일의 앞 10행 또는 마지막 10행만 출력

예) # `head /etc/systemd/bootchart.conf`

- **more**

텍스트로 작성된 파일을 화면에 페이지 단위로 출력

예) # `more /etc/systemd/system.conf`

리눅스 기본 명령어 (4)

- **less**

more와 용도가 비슷하지만 기능이 더 확장된 명령
예) # less /etc/systemd/system.conf

- **file**

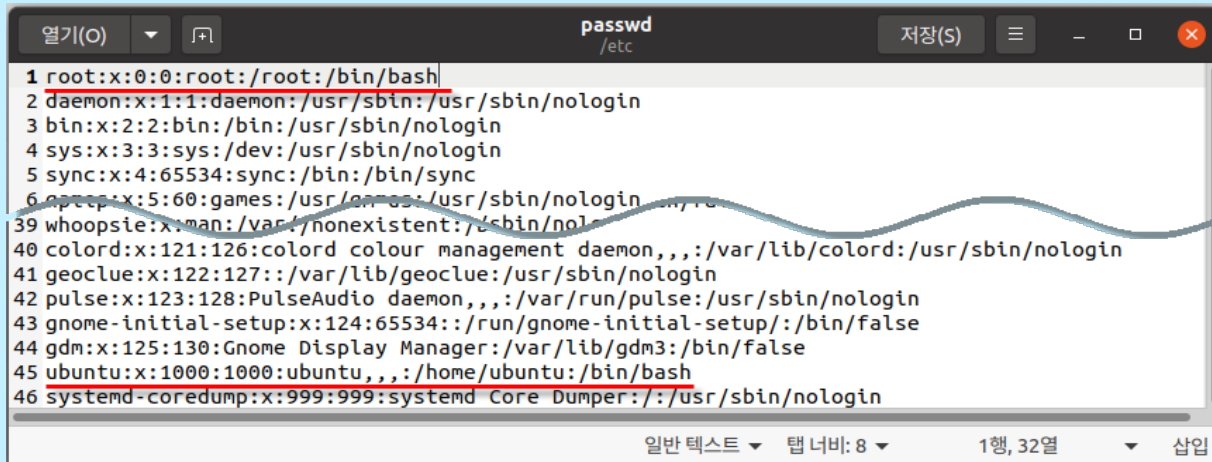
File이 어떤 종류의 파일인지를 표시
예) # file /etc/systemd/system.conf

- **clear**

명령창을 깨끗하게 지워줌
예) # clear

사용자와 그룹(1)

- 리눅스는 다중 사용자 시스템(Multi-User System) 임
- 기본적으로 root라는 이름을 가진 수퍼유저(Superuser)가 있으며, 모든 작업을 할 수 있는 권한이 있음
- 모든 사용자를 하나 이상의 그룹에 소속되어 있음
- 사용자는 /etc/passwd 파일에 정의되어 있음



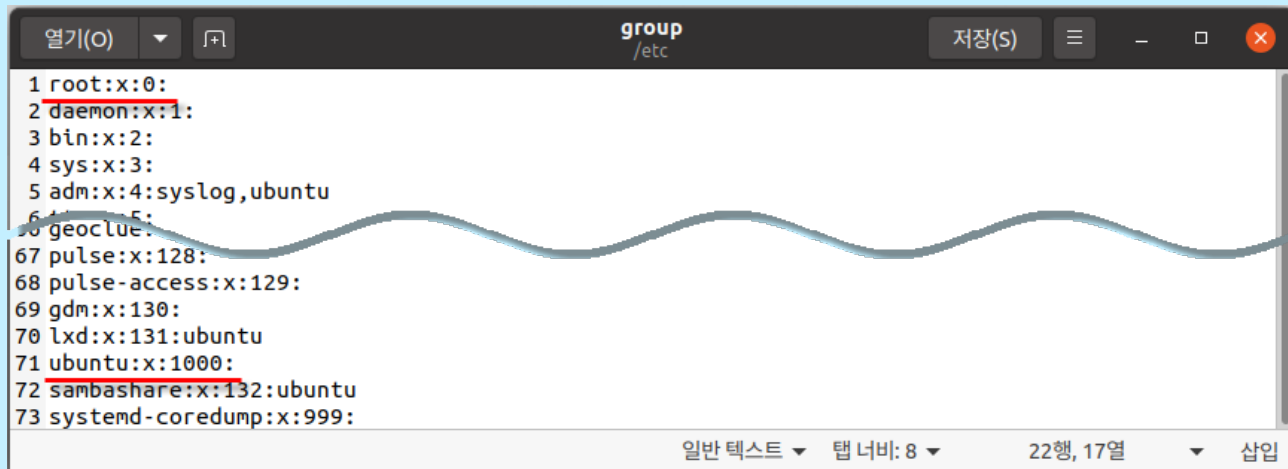
```
1 root:x:0:0:root:/root:/bin/bash
2 daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
3 bin:x:2:2:bin:/bin:/usr/sbin/nologin
4 sys:x:3:3:sys:/dev:/usr/sbin/nologin
5 sync:x:4:65534:sync:/bin:/bin/sync
6 apt:x:5:60:games:/usr/games:/usr/sbin/nologin
39 whoopsie:x:man:/var/lib/whoopsie:/usr/sbin/nologin
40 colord:x:121:126:colord colour management daemon,,,:/var/lib/colord:/usr/sbin/nologin
41 geoclue:x:122:127::/var/lib/geoclue:/usr/sbin/nologin
42 pulse:x:123:128:PulseAudio daemon,,,:/var/run/pulse:/usr/sbin/nologin
43 gnome-initial-setup:x:124:65534::/run/gnome-initial-setup:/bin/false
44 gdm:x:125:130:Gnome Display Manager:/var/lib/gdm3:/bin/false
45 ubuntu:x:1000:1000:ubuntu,,,:/home/ubuntu:/bin/bash
46 systemd-coredump:x:999:999:systemd Core Dumper:/:/usr/sbin/nologin
```

- 각 행의 의미는 다음과 같음

사용자 이름:암호:사용자 ID:사용자가 소속된 그룹 ID:추가 정보:홈 디렉터리:기본 셸

사용자와 그룹(2)

- 사용자의 비밀번호는 `/etc/shadow` 파일에 정의되어 있음
- 그룹은 `/etc/group` 파일에 정의되어 있음



A screenshot of a text editor window titled 'group /etc'. The window displays the contents of the `/etc/group` file. The text is as follows:

```
1 root:x:0:
2 daemon:x:1:
3 bin:x:2:
4 sys:x:3:
5 adm:x:4:syslog,ubuntu
6 lp:x:7:
67 pulse:x:128:
68 pulse-access:x:129:
69 gdm:x:130:
70 lxd:x:131:ubuntu
71 ubuntu:x:1000:
72 sambashare:x:132:ubuntu
73 systemd-coredump:x:999:
```

The text is displayed in a monospaced font. The window has a dark title bar with buttons for '열기(O)', '저장(S)', and window control icons. At the bottom, there is a status bar showing '일반 텍스트', '탭 너비: 8', '22행, 17열', and '삽입'.

- 각 행의 의미는 다음과 같음
그룹명:비밀번호:그룹 id:보조 그룹 사용자

사용자와 그룹 관련 명령어(1)

- **adduser**

새로운 사용자를 추가

예) # `adduser newuser1`

➤ 사용자 생성시 옵션
--uid : ID 지정
--gid : 그룹 지정
--home : 홈 디렉터리 지정
--shell : 셸 지정

- **passwd**

사용자의 비밀번호를 지정하거나 변경

예) # `passwd newuser1`

- **usermod**

사용자의 속성을 변경

예) # `usermod --groups ubuntu newuser1`

- **userdel**

사용자를 삭제

예) # `userdel newuser2`

사용자와 그룹 관련 명령어(2)

- `chage`

사용자의 암호를 주기적으로 변경하도록 설정

예) `# chage -m 2 newuser1`

- `groups`

현재 사용자가 속한 그룹을 보여줌

예) `# groups`

- `groupadd`

새로운 그룹을 생성

예) `# groupadd newgroup1`

- `groupmod`

그룹의 속성을 변경

예) `# groupmod --new-name mygroup1 newgroup1`

사용자와 그룹 관련 명령어(3)

- `groupdel`

그룹을 삭제

예) # `groupdel newgroup2`

- `gpsswd`

그룹의 암호를 설정하거나, 그룹의 관리를 수행

예) # `gpsswd mygroup1`



<실습8> 사용자와 그룹 관리 연습

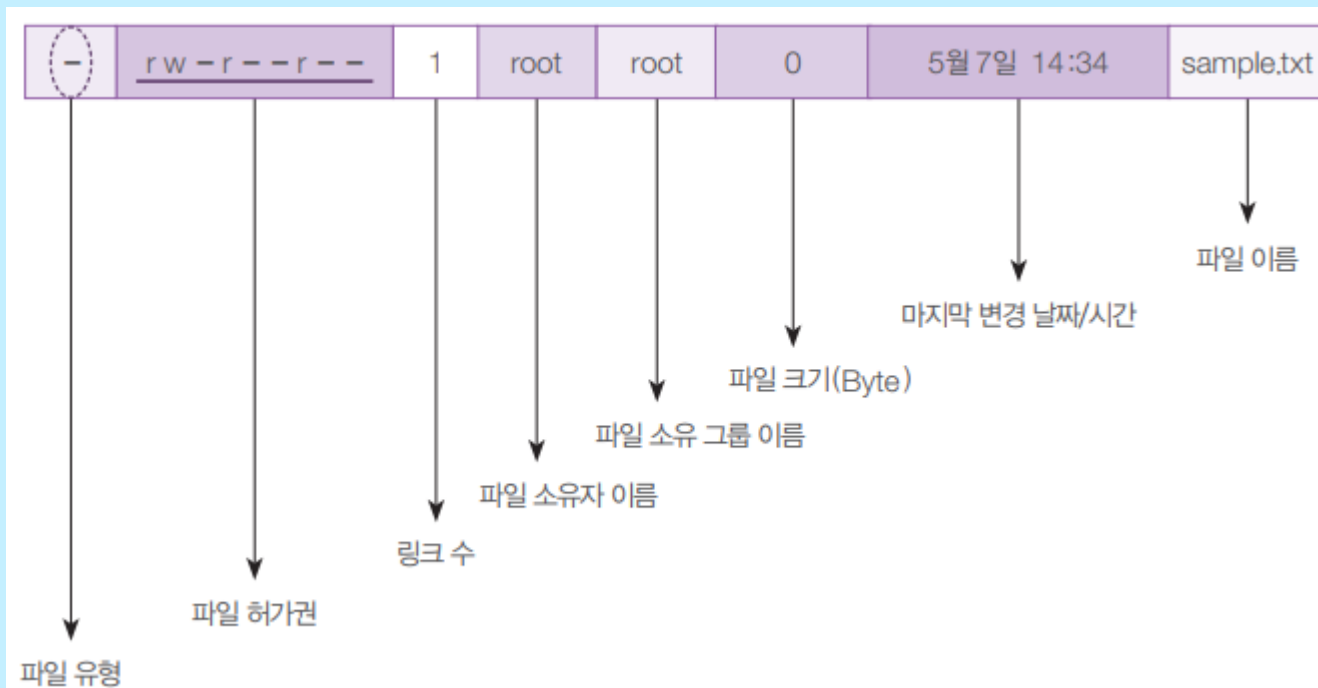
- 실습 목표
 - 사용자와 그룹의 생성/삭제/변경 명령어를 연습한다.
 - 사용자와 그룹의 관련 파일을 파악한다.
 - X윈도우에서 사용자 관리 툴을 확인한다.
- 실습 화면 (X 윈도에서 사용자 추가)



파일과 디렉터리의 소유와 허가권 (1)

- 파일의 리스트와 파일 속성

```
root@server: ~/바탕화면
root@server:~/바탕화면# touch sample.txt
root@server:~/바탕화면# ls -l sample.txt
-rw-r--r-- 1 root root 0  5월  7 14:34 sample.txt
root@server:~/바탕화면#
```



파일과 디렉터리의 소유와 허가권 (2)

- 파일 유형

- 디렉터리일 경우에는 d, 일반적인 파일일 경우에는 -가 표시

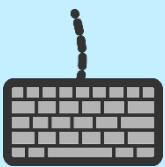
- 파일 허가권(Permission)

- "rw-" , " r--" , " r--" 3개씩 끊어서 읽음 (r은 read, w는 write, x는 execute의 약자)
- 첫 번째 "rw-" 는 소유자(User)의 파일접근 권한
- 두 번째의 "r--" 는 그룹(Group)의 파일접근 권한
- 세 번째의 "r--" 는 그 외의 사용자(Other)의 파일접근 권한
- 숫자로도 표시 가능 (8진수)

소유자(User)			그룹(Group)			그 외 사용자(Other)		
r	w	-	r	-	-	r	-	-
4	2	0	4	0	0	4	0	0
6			4			4		

파일과 디렉터리의 소유와 허가권 (3)

- `chmod` 명령
 - 파일 허가권 변경 명령어
 - 예) `# chmod 777 sample.txt`
- 파일 소유권 (Ownership)
 - 파일을 소유한 사용자와 그룹을 의미
- `chown/chgrp` 명령
 - 파일의 소유권을 바꾸는 명령어
 - 예) `# chown ubuntu.ubuntu sample.txt` 또는
`# chown ubuntu sample.txt` 및 `# chgrp ubuntu sample.txt`



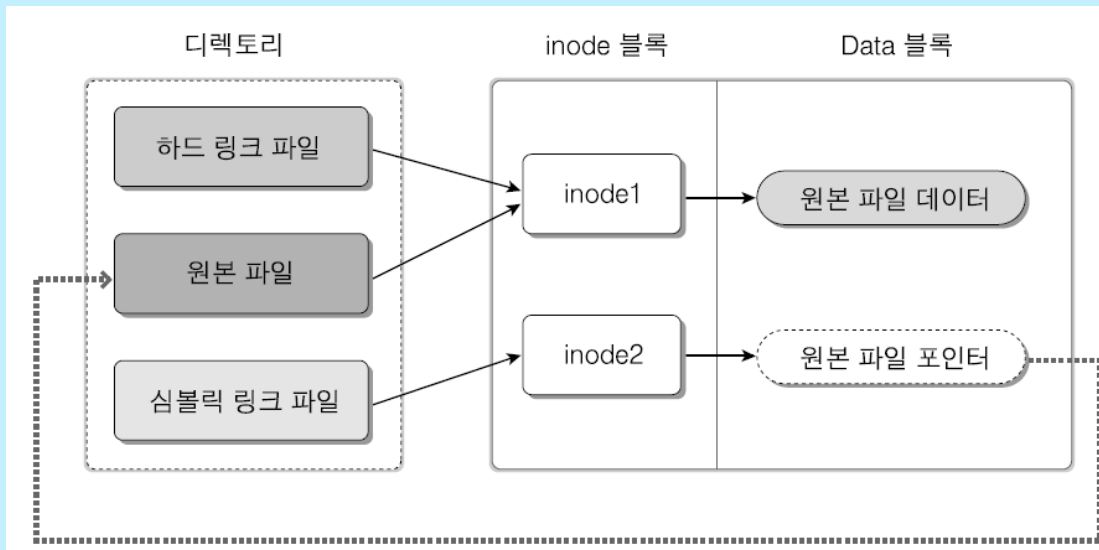
<실습9> 파일의 허가권 및 소유권 연습

- 실습목표
 - 파일의 허가권에 대한 개념과 명령어를 익힌다.
 - 파일의 소유권에 대한 개념과 명령어를 익히고, 보안에 관련된 내용도 확인해 본다.
- 실습 화면 (소유권 변경 실패)

```
ubuntu@server: ~  
ubuntu@server:~$ whoami  
ubuntu  
ubuntu@server:~$ chown root.root test  
chown: 'test'의 소유자 변경: 명령을 허용하지 않음  
ubuntu@server:~$
```

링크

- 파일의 링크(Link)에는 하드 링크(Hard Link)와 심볼릭 링크(Symbolic Link 또는 Soft Link) 두 가지가 있음



➤ 심볼릭 링크는 Windows의 바로가기 아이콘과 개념이 비슷하다.

- 하드 링크를 생성하면 "하드링크파일" 만 하나 생성되며 같은 inode1을 사용 (명령 : # ln 링크대상파일이름 링크파일이름)
- 심볼릭 링크를 생성하면 새로운 inode2를 만들고, 데이터는 원본 파일을 연결하는 효과 (명령 : # ln -s 링크대상파일이름 링크파일이름)



<실습10> 하드 링크, 심볼릭 링크 연습

- 실습 목표
 - [그림 4-94]의 개념을 실습을 통해 익힌다.
- 실습 화면 (링크 실습 화면)

```
root@server: ~/linktest
root@server:~/linktest# ln basefile hardlink
root@server:~/linktest# ln -s basefile softlink
root@server:~/linktest# ls -il
합계 8
394289 -rw-r--r-- 2 root root 61  5월  7 16:06 basefile
394289 -rw-r--r-- 2 root root 61  5월  7 16:06 hardlink
394288 lrwxrwxrwx 1 root root  8  5월  7 16:11 softlink -> basefile
root@server:~/linktest# cat hardlink
파일 링크를 실습하기 위한 원본 파일입니다.
root@server:~/linktest# cat softlink
파일 링크를 실습하기 위한 원본 파일입니다.
root@server:~/linktest#
```

프로그램 설치를 위한 dpkg (1)

- `dpkg`(Debian Package)
 - Windows의 "`setup.exe`" 와 비슷한 설치 파일
 - 확장명은 `*.deb`이며, 이를 패키지(Package)라고 부름.
- 파일의 의미

```
root@server: ~/다운로드
root@server:~/다운로드# ls -l
합계 156
-rw-r--r-- 1 root root 159468 5월  8 09:01 calculator_2.1.4-1_amd64.deb
root@server:~/다운로드#
```

패키지이름_버전-개정번호_아키텍처.deb

- 패키지이름 : `calculator` → 패키지(프로그램)의 이름
- 버전 : `2.1.4` → 대개 3자리수로 구성. 주버전, 부버전, 패치버전
- 개정번호 : `1` → 문제점을 개선할 때마다 붙여지는 번호
- 아키텍처 : `amd64` → 64비트 CPU를 의미

➤ 아키텍처
`amd64` : 64비트 CPU
`i386` : 32비트 CPU
`all` : 모든 CPU

프로그램 설치를 위한 dpkg (2)

- 자주 사용하는 dpkg 명령어 옵션
 - 설치
 - `dpkg -i 패키지파일이름.deb`
 - 삭제
 - `dpkg -r 패키지이름`
 - `dpkg -p 패키지이름` → 설정파일까지 삭제
 - 패키지 조회
 - `dpkg -i 패키지이름` → 설치된 패키지에 대한 정보를 보여줌
 - `dpkg -L 패키지이름` → 패키지가 설치한 파일 목록을 보여줌
 - 아직 설치되지 않은 deb 파일 조회
 - `dpkg --info 패키지파일이름.deb` → 패키지 파일에 대한 정보를 보여줌

프로그램 설치를 위한 dpkg (3)

- dpkg 명령의 단점
 - '의존성' 문제
 - A패키지가 설치되기 위해서 B패키지가 필요할 경우, dpkg 명령으로는 해결이 까다로움.
 - 이를 해결하기 위해 apt가 등장함



<실습11> dpkg 명령 연습

- 실습목표
 - dpkg 명령어의 사용법을 익힌다.
 - deb 파일의 의존성 문제를 파악한다.
- 실습화면 (axel 패키지 설치)

```
root@server: ~/다운로드
root@server:~/다운로드# dpkg -i axel_2.17.5-1ubuntu1_amd64.deb
Selecting previously unselected package axel.
(데이터베이스 읽는중 ...현재 146363개의 파일과 디렉터리가 설치되어 있습니다.)
Preparing to unpack axel_2.17.5-1ubuntu1_amd64.deb ...
Unpacking axel (2.17.5-1ubuntu1) ...
axel (2.17.5-1ubuntu1) 설정하는 중입니다 ...
Processing triggers for man-db (2.9.1-1) ...
root@server:~/다운로드#
root@server:~/다운로드# dpkg -l axel
희망상태=알수없음(U)/설치(I)/지우기(R)/깨끗이(P)/고정(H)
| 상태=아님(N)/설치(I)/설정(C)/폴림(U)/절반설정(F)/일부설치(H)/트리거대기(W)/
| / 트리거미림(T)
|/ 오류?=(없음)/다시설치필요(R) (상태, 오류가 대문자=불량)
||/ 이름 버전 Architecture 설명
+++=====
ii axel 2.17.5-1ubuntu1 amd64 light command line download acc>
root@server:~/다운로드#
```


편리한 패키지 설치, apt (1)

- apt 명령

- "dpkg" 명령의 패키지 의존성 문제를 완전하게 해결됨.
- 인터넷을 통하여 필요한 파일을 저장소(Repository)에서 자동으로 모두 다운로드해서 설치하는 방식

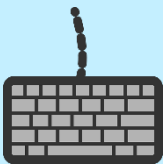
➤ 저장소의 URL은/etc/apt/sources.list 파일

- apt 기본적인 사용법

- 기본 설치 : `apt install 패키지이름`
 - 주로 "`apt -y install 패키지이름`" 으로 사용
 - "-y" 는 사용자의 확인을 모두 "yes" 로 간주하고 설치를 진행한다는 옵션
- 패키지 목록의 업데이트 : `apt update`
- 삭제 : `apt remove/purge 패키지 이름`
- 사용하지 않는 패키지 제거 : `apt autoremove`
- 내려 받은 파일 제거 : `apt clean` 또는 `apt autoclean`

편리한 패키지 설치, apt (2)

- `apt-cache`
 - 패키지를 설치하기 전에 패키지에 대한 정보나 의존성 문제를 미리 확인
 - 패키지 정보 보기
 - `apt-cache show` 패키지이름
 - 패키지 의존성 확인
 - `apt-cache depends` 패키지이름
 - 패키지 역의존성 확인
 - `apt-cache rdepends` 패키지이름



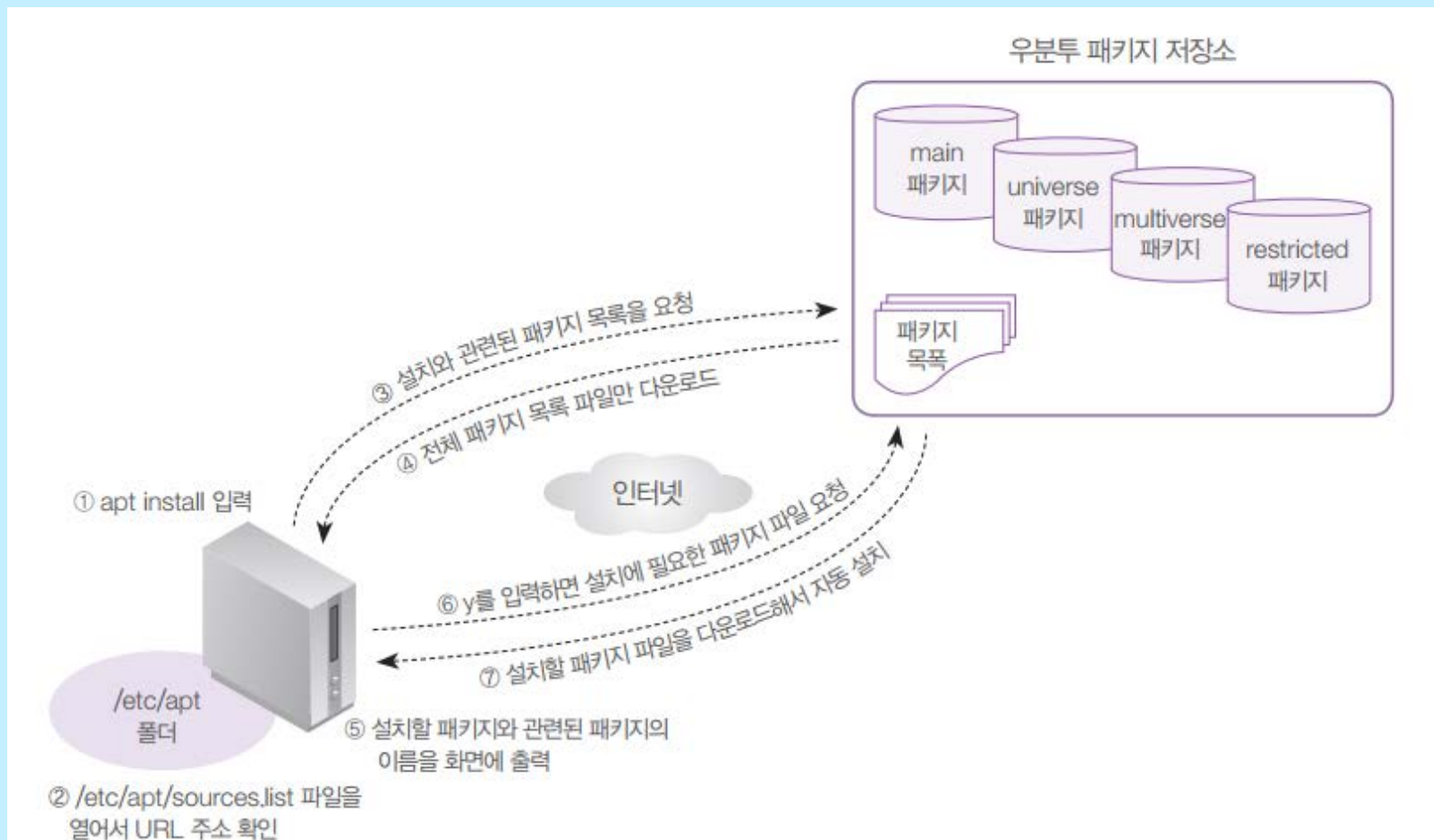
<실습12> apt 기본 연습

- 실습 목표
 - 의존성 문제가 있는 패키지를 "apt" 명령으로 설치해 본다.
- 실습 화면 (의존성 패키지 자동 다운로드)

```
root@server: ~/바탕화면
root@server:~/바탕화면# apt install calculator
패키지 목록을 읽는 중입니다... 완료
의존성 트리를 만드는 중입니다
상태 정보를 읽는 중입니다... 완료
다음의 추가 패키지가 설치될 것입니다 :
  libquadmath0
다음 새 패키지를 설치할 것입니다:
  calculator libquadmath0
0개 업그레이드, 2개 새로 설치, 0개 제거 및 0개 업그레이드 안 함.
308 k바이트 아카이브를 받아야 합니다.
이 작업 후 1,816 k바이트의 디스크 공간을 더 사용하게 됩니다.
계속 하시겠습니까? [Y/n]
```

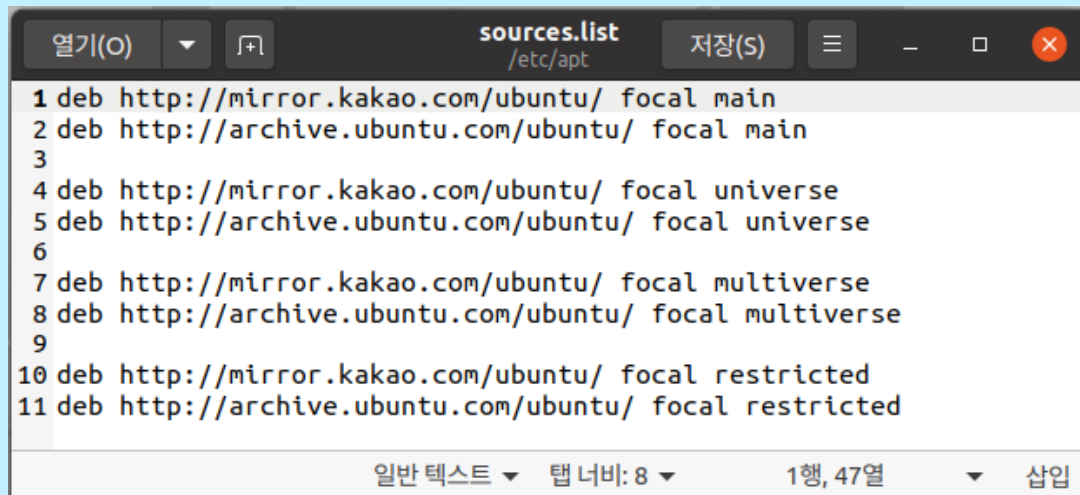
편리한 패키지 설치, apt (3)

- apt 작동 방식 설정 파일
 - 'apt install 패키지이름' 명령이 작동하는 방식



편리한 패키지 설치, apt (4)

- 우분투 패키지 저장소
 - `main` : 우분투에서 공식적으로 지원하는 무료(Free) SW
 - `universe` : 우분투에서 지원하지 않는 무료 SW
 - `restricted` : 우분투에서 공식적으로 지원하는 유료(Non-Free) SW
 - `multiverse` : 우분투에서 지원하지 않는 유료 SW
- 저장소가 기록된 파일
 - `/etc/apt/sources.list` 파일
 - 형식



```
sources.list
/etc/apt

1 deb http://mirror.kakao.com/ubuntu/ focal main
2 deb http://archive.ubuntu.com/ubuntu/ focal main
3
4 deb http://mirror.kakao.com/ubuntu/ focal universe
5 deb http://archive.ubuntu.com/ubuntu/ focal universe
6
7 deb http://mirror.kakao.com/ubuntu/ focal multiverse
8 deb http://archive.ubuntu.com/ubuntu/ focal multiverse
9
10 deb http://mirror.kakao.com/ubuntu/ focal restricted
11 deb http://archive.ubuntu.com/ubuntu/ focal restricted
```

일반 텍스트 ▼ 탭 너비: 8 ▼ 1행, 47열 ▼ 삽입

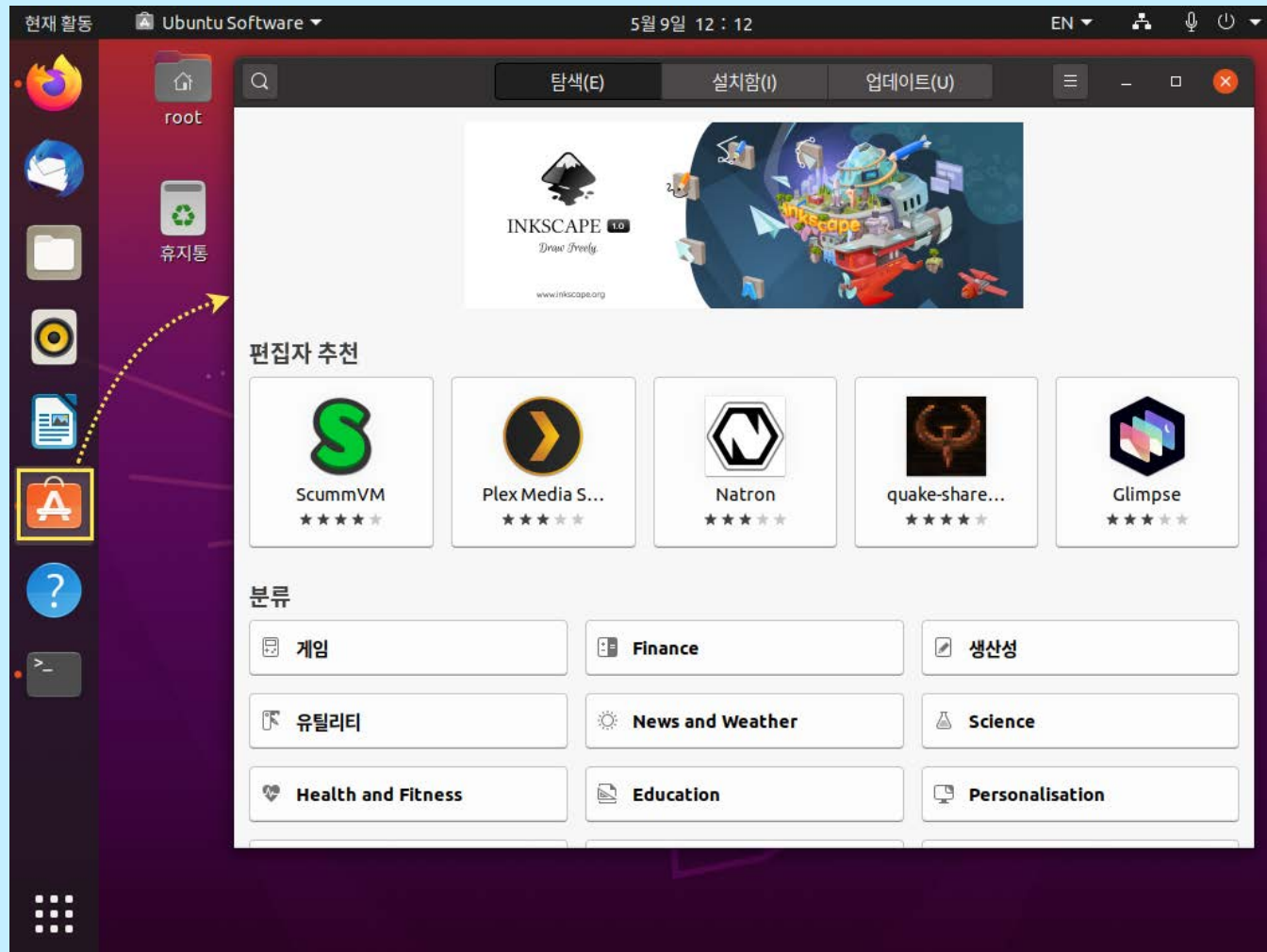


<실습13> apt 추가 실습

- 실습 목표
 - 저장소 URL을 변경하는 실습을 한다.
 - 업데이트된 패키지를 설치하는 방법을 확인한다.
- 실습 화면 (전체 패키지 업드레이드 화면)

```
root@server: ~/바탕화면
root@server:~/바탕화면# apt upgrade
패키지 목록을 읽는 중입니다... 완료
의존성 트리를 만드는 중입니다
상태 정보를 읽는 중입니다... 완료
업그레이드를 계산하는 중입니다... 완료
다음 새 패키지를 설치할 것입니다:
linux-headers-5.4.0-29 linux-headers-5.4.0-29-generic
linux-image-5.4.0-29-generic linux-modules-5.4.0-29-generic
linux-modules-extra-5.4.0-29-generic
다음 패키지를 업그레이드할 것입니다:
cups cups-bsd cups-client cups-common cups-core-drivers cups-daemon
cups-ipp-utils cups-ppdc cups-server-common distro-info-data file-roller
fox-locale-en firefox-locale-ko fonts-symbol
python3-myscripter python3-r1.2-ubuntu.8-minimal-ubuntu-shell
ubuntu-release-upgrader-core ubuntu-release-upgrader-gtk uno-libs-private
update-manager update-manager-core ure
78개 업그레이드, 5개 새로 설치, 0개 제거 및 0개 업그레이드 안 함.
209 M바이트 아카이브를 받아야 합니다.
이 작업 후 359 M바이트의 디스크 공간을 더 사용하게 됩니다.
계속 하시겠습니까? [Y/n]
```

Ubuntu Software



파일의 압축과 묶기 (1)

- 파일 압축

- 압축파일 확장명은 xz, bz2, gz, zip, Z 등
- xz나 bz2 압축률이 더 좋음

- 파일 압축 관련 명령

- xz : 확장명 xz로 압축을 하거나 풀어준다
예) xz 파일명
xz -d 파일명.xz
- bzip2 : 확장명 bz2로 압축을 하거나 풀어준다
예) bzip2 파일명
bzip2 -d 파일명.bz2
- gzip : 확장명 gz으로 압축을 하거나 풀어준다
예) gzip 파일명
gzip -d 파일명.gz
- zip/unzip : 확장명 zip으로 압축하거나 풀어준다,
예) zip 새로생성될파일이름.zip 압축할파일이름
unzip 압축파일이름.zip

파일의 압축과 묶기 (2)

- 파일 묶기

- 리눅스(유닉스)에서는 '파일 압축' 과 '파일 묶기' 는 원칙적으로 별개의 프로그램으로 수행
- 파일 묶기의 명령어는 'tar' 이며, 묶인 파일의 확장명도 'tar' 이다

- 파일 묶기 명령(tar)

- tar : 확장명 tar로 묶음 파일을 만들어 주거나 묶음을 풀어 준다

동작 : c(묶기), x(풀기), t(경로확인)

옵션 : f(파일), v(과정보이기), J(tar+xz), z(tar+gzip), j(tar+bzip2)

- 사용 예

tar cvf my.tar /etc/systemd/ → 묶기

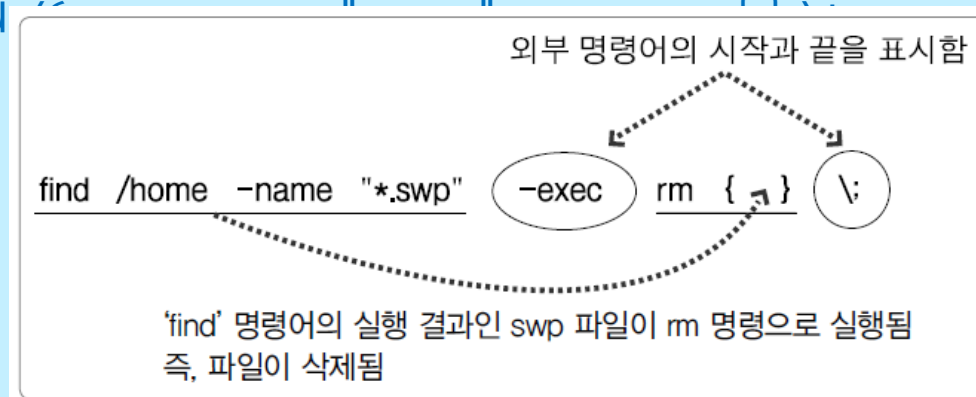
tar cvfJ my.tar.xz /etc/systemd/ → 묶기 + xz 압축

tar xvf my.tar → tar 풀기

tar xvfJ my.tar.xz /etc/systemd/ → xz 압축 해제 + tar 풀기

파일 위치 검색

- `find` [경로] [옵션] [조건] [action] : 기본 파일 찾기
 - [옵션] `-name`, `-user`(소유자), `-newer`(전,후), `-perm`(허가권), `-size`(크기)
 - [action] `-print`(디폴트), `-exec` (외부명령 실행)
 - 사용 예
 - # `find /etc -name "*.conf"`
 - # `find /bin -size +100k -size -1000k`
 - # `find / -name "*.swp" -exec rm {} \;`



- `which` 실행파일 이름 : PATH에 설정된 디렉터리만 검색
- `whereis` 실행파일 이름 : 실행 파일, 소스, man 페이지 파일까지 검색
- `locate` 파일 이름 : 파일 목록 데이터베이스에서 검색

시스템 설정

- 다양한 환경 설정 (`gnome-control-center`)



- 네트워크 설정 (`nmtui`)
- 방화벽 설정 (`ufw`, `gufw`)

CRON과 AT

- **cron**

- 주기적으로 반복되는 일을 자동적으로 실행될 수 있도록 설정
- 관련된 데몬(서비스)은 "crond" , 관련 파일은 "/etc/crontab"
- /etc/crontab 형식
 - 분 시 일 월 요일 사용자 실행명령
 - 예) 00 05 1 * * root cp -r /home /backup

- **at**

- 일회성 작업을 예약
- 사용 예
 - 예약 : # at <시간>
 - 예) # at 3:00am tomorrow → 내일 새벽 3시
 - # at now + 1 hours → 1시간 후
 - at> 프롬프트에 예약 명령어 입력 후 [Enter]
 - 완료되면 [Ctrl] + [D]
 - 확인 : # at -l
 - 취소 : # atrm <작업번호>



<실습14> cron, at 실습

- 실습 목표
 - CRON을 활용하여 매월 15일 새벽 3시 1분에 /home 디렉터리와 그 하위 디렉터리를 /backup 디렉터리에 백업하는 방법을 익힌다.
 - AT의 사용법을 익힌다.
- 실습 화면 (백업 진행)

```
root@server: ~
root@server:~# date 011503002030
2030. 01. 15. (화) 03:00:00 KST
root@server:~# systemctl restart cron
root@server:~# ls -l /backup
합계 360
-rw-r--r-- 1 root root 366664 1월 15 03:01 backup-2030.01.15.tar.xz
root@server:~#
root@server:~# date 021503002030
2030. 02. 15. (금) 03:00:00 KST
root@server:~# systemctl restart cron
root@server:~# ls -l /backup
합계 720
-rw-r--r-- 1 root root 366664 1월 15 03:01 backup-2030.01.15.tar.xz
-rw-r--r-- 1 root root 366664 2월 15 03:01 backup-2030.02.15.tar.xz
root@server:~#
```

네트워크 관련 필수 개념 (1)

- TCP/IP
 - 컴퓨터끼리 네트워크 상으로 의사소통을 하는 “프로토콜” 중 가장 널리 사용되는 프로토콜의 한 종류
- 호스트 이름(Hostname)과 도메인 이름(Domain name)
 - 호스트 이름은 각각의 컴퓨터에 지정된 이름
 - 도메인 이름(또는 도메인 주소)는 nanbit.co.kr과 같은 형식
- IP 주소
 - 각 컴퓨터의 랜카드에 부여되는 중복되지 않는 유일한 주소
 - 4바이트로 이루어져 있으며, 각 자리는 0~255까지의 숫자
 - 예) Server의 IP 주소는 192.168.111.100
- 네트워크 주소
 - 같은 네트워크에 속해 있는 공통된 주소 (예 : 192.168.111.0)

네트워크 관련 필수 개념 (2)

- 브로드캐스트(Broadcast) 주소
 - 내부 네트워크의 모든 컴퓨터가 듣게 되는 주소
 - 현재 주소의 제일 끝자리를 255로 바꾼 주소(C클래스)
- 게이트웨이(Gateway), 라우터(Router)
 - 라우터 = 게이트웨이
 - 네트워크 간에 데이터를 전송하는 컴퓨터 또는 장비
 - VMware의 게이트웨이 주소는 192.168.111.2로 고정
- 넷마스크(Netmask) & 클래스(Class)
 - 넷마스크 : 네트워크의 규모를 결정(예:255.255.255.0-C클래스)
- DNS(Domain Name System) 서버 (= 네임 서버) 주소
 - URL을 해당 컴퓨터의 IP주소로 변환해 주는 서버
 - 설정 파일은 /etc/resolv.conf
 - VMware를 사용하면 VMware가 192.168.111.2번을 게이트웨이 및 DNS 서버로, 192.168.111.254를 DHCP 서버로 설정함.

네트워크 관련 필수 개념 (3)

- 리눅스에서의 네트워크 장치 이름
 - 우분투는 랜카드를 `ens32` 또는 `ens33`으로 할당함
 - 명령 예
 - `# ifconfig ens32 또는 ens33` → 네트워크 설정 정보를 출력
 - `# ifdown ens32 또는 ens33` → 네트워크 장치를 정지
 - `# ifup ens32 또는 ens33` → 네트워크 장치를 가동

중요한 네트워크 관련 명령어 (1)

- `nm-connection-editor` 또는 `nmtui`
 - 네트워크와 관련된 대부분의 작업을 이 명령어에서 수행
 - 자동 IP 주소 또는 고정 IP주소 사용 결정
 - IP주소, 서브넷 마스크, 게이트웨이 정보 입력
 - DNS 정보 입력
 - 네트워크 카드 드라이버 설정
 - 네트워크 장치(`ens32`)의 설정
- `systemctl <start/stop/restart/status> networking`
 - 네트워크의 설정을 변경한 후에, 변경된 내용을 시스템에 적용시키는 명령어

중요한 네트워크 관련 명령어 (2)

- `ifconfig <장치이름>`
 - 장치의 IP주소 설정 정보를 출력
- `nslookup`
 - DNS 서버의 작동을 테스트하는 명령어
- `ping <IP주소 또는 URL>`
 - 해당 컴퓨터가 네트워크상에서 응답하는지를 테스트하는 간편한 명령어

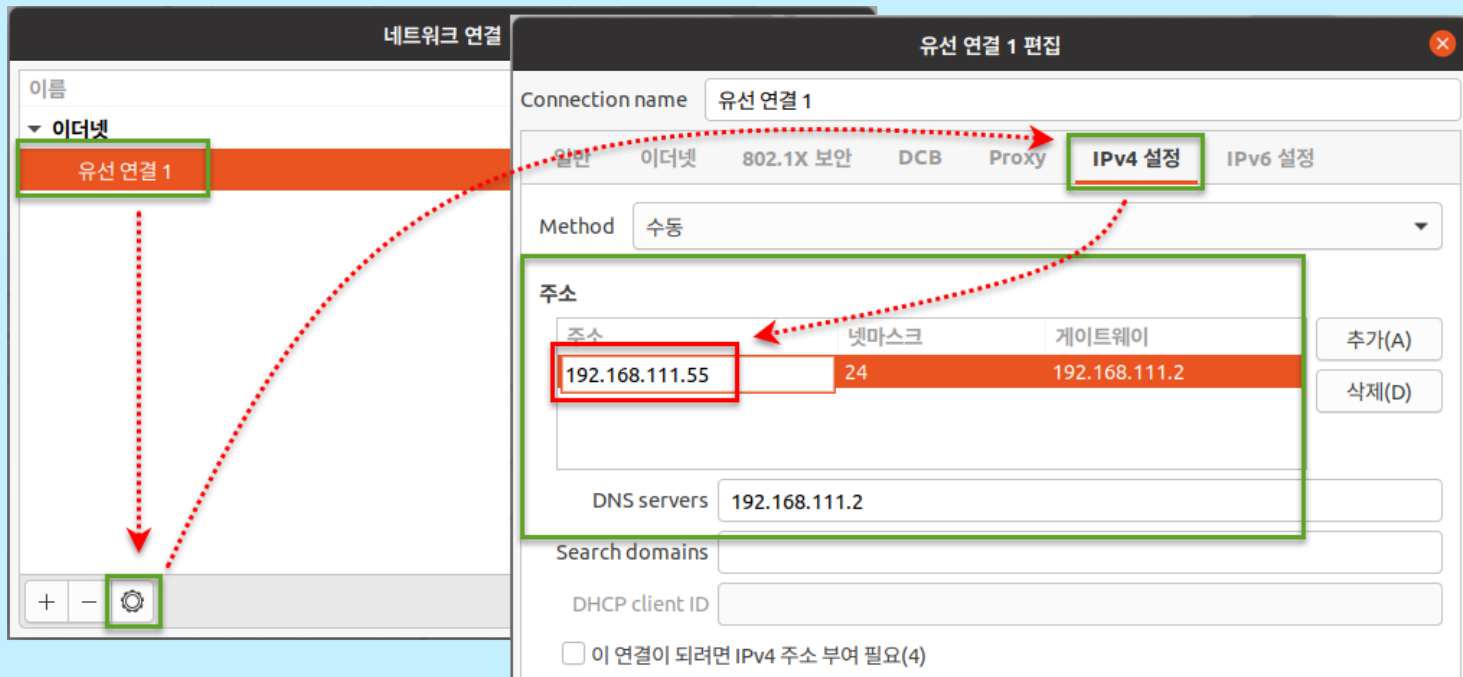
네트워크 설정과 관련된 주요 파일

- 네트워크 기본 정보가 설정된 파일
 - X 윈도 모드 : Server, Client
 - '/etc/NetworkManager/system-connections/유선 연결 1' 파일
 - 텍스트 모드 : Server(B)
 - /etc/netplan/*.yaml 파일
- /etc/resolv.conf
 - DNS 서버의 정보 및 호스트 이름이 들어 있는 파일
 - 영구적으로 DNS 서버 정보를 변경하려면 nm-connection-editor 명령이나 /etc/netplan/*.yaml 파일을 직접 편집해야 함
- /etc/hosts
 - 현 컴퓨터의 호스트 이름 및 FQDN이 들어 있는 파일



<실습15> nm-connection-editor 명령

- 실습 목표
 - nm-connection-editor 명령의 작동을 이해한다.
 - 네트워크 관련 파일들의 내용을 확인한다.
 - DNS의 작동을 /etc/resolv.conf 파일과 연관해서 이해한다.
- 실습 화면 (네트워크 설정)



파이프, 필터, 리디렉션

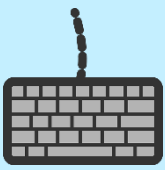
- 파이프(pipe)
 - 두 개의 프로그램을 연결해 주는 연결통로의 의미
 - "|" 문자를 사용함
 - 예) # ls -l /etc | more
- 필터(filter)
 - 필요한 것만 걸러 주는 명령어
 - grep, tail, wc, sort, grep, awk, sed 등
 - 주로 파이프와 같이 사용
 - 예) # ps -ef | grep bash
- 리디렉션 (redirection)
 - 표준 입출력의 방향을 바꿔 줌
 - 예) ls -l > list.txt
 - sort < list.txt > out.txt

프로세스, 데몬 (1)

- 정의
 - 하드디스크에 저장된 실행코드(프로그램)가, 메모리에 로딩되어 활성화된 것
- 포그라운드 프로세스(Foreground Process)
 - 실행하면 화면에 나타나서 사용자와 상호작용을 하는 프로세스
 - 대부분의 응용프로그램
- 백그라운드 프로세스(Background Process)
 - 실행은 되었지만, 화면에는 나타나지 않고 실행되는 프로세스
 - 백신 프로그램, 서버 데몬 등
- 프로세스 번호
 - 각각의 프로세스에 할당된 고유번호
- 작업 번호
 - 현재 실행되고 있는 백그라운드 프로세스의 순차번호

프로세스, 데몬 (2)

- 부모 프로세스와 자식 프로세스
 - 모든 프로세스는 부모 프로세스를 가지고 있음
 - 부모 프로세스를 kill 하면, 자식 프로세스도 자동으로 kill 됨
- 프로세스 관련 명령
 - **ps**
 - 현재 프로세스의 상태를 확인하는 명령어
 - "ps -ef | grep <프로세스 이름>" 을 주로 사용함
 - **kill**
 - 프로세스를 강제로 종료하는 명령어
 - "kill -9 <프로세스 번호>" 는 강제 종료
 - **pstree**
 - 부모 프로세스와 자식 프로세스의 관계를 트리 형태로 보여 줌



<실습16> 프로세스 연습

- 실습 목표
 - ▣ 포그라운드 프로세스와 백그라운드 프로세스의 상호 전환 연습을 한다.
- 실습 화면 (백그라운드 프로세스로 실행)

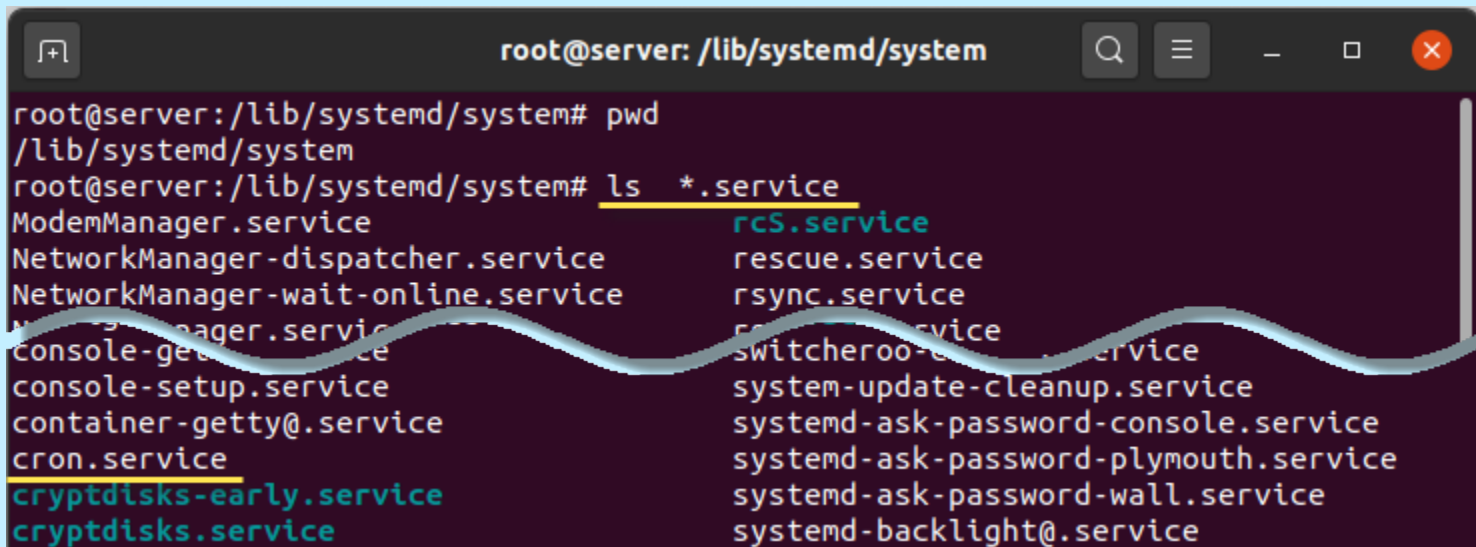
A screenshot showing a terminal window and a gedit editor window. The terminal window has a title bar 'root@server: ~/바탕화면' and shows the command 'gedit &' being executed, with the output '[1] 4860'. The gedit window is titled '이름 없는 문서 1' and shows a blank document with line number 1 visible.

```
root@server: ~/바탕화면
root@server:~/바탕 화면# gedit &
[1] 4860
root@server:~/바탕 화면#
```


서비스와 소켓(1)

- 서비스

- 시스템과 독자적으로 구동되어 제공하는 프로세스를 말한다, 예로 웹 서버, DB 서버, FTP 서버 등이 있다.
- 실행 및 종료는 대개 'systemctl start/stop/restart 서비스이름' 으로 사용된다.
- 서비스의 실행 스크립트 파일은 /lib/systemd/system/ 디렉터리에 '서비스이름.service' 라는 이름으로 확인할 수 있다, 예로 Cron 서비스는 cron.service라는 이름의 파일로 존재한다



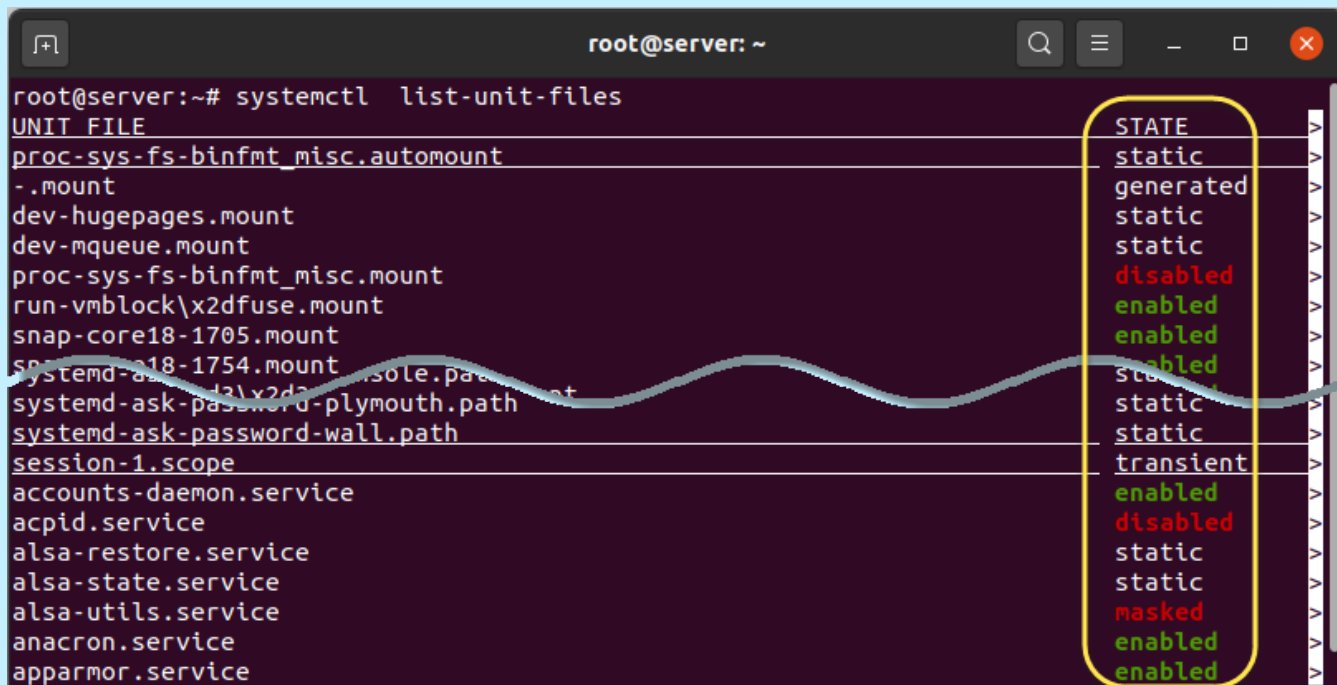
A terminal window titled 'root@server: /lib/systemd/system' showing the output of the 'ls *.service' command. The terminal has a dark background with light-colored text. A blue wavy line is drawn across the middle of the terminal output. The file names are listed in two columns, with some in red and some in green.

```
root@server:/lib/systemd/system# pwd
/lib/systemd/system
root@server:/lib/systemd/system# ls *.service
ModemManager.service          rcS.service
NetworkManager-dispatcher.service  rescue.service
NetworkManager-wait-online.service  rsync.service
NetworkManager.service         rsyslog.service
console-getty.service          Switcheroo-control.service
console-setup.service          system-update-cleanup.service
container-getty@.service       systemd-ask-password-console.service
cron.service                   systemd-ask-password-plymouth.service
cryptdisks-early.service       systemd-ask-password-wall.service
cryptdisks.service             systemd-backlight@.service
```

서비스와 소켓(2)

- 서비스

- 부팅과 동시에 서비스의 자동 실행 여부를 지정할 수 있는데, 터미널에서 `systemctl list-unit-files` 명령을 실행하면 현재 사용(enabled)과 사용 안 함(disabled)을 확인할 수 있다.



```
root@server: ~  
root@server:~# systemctl list-unit-files  
UNIT FILE  
proc-sys-fs-binfmt_misc.automount  
  .mount  
dev-hugepages.mount  
dev-mqueue.mount  
proc-sys-fs-binfmt_misc.mount  
run-vmblock\x2dfuse.mount  
snap-core18-1705.mount  
systemd-ask-password-console.path  
systemd-ask-password-plymouth.path  
systemd-ask-password-wall.path  
session-1.scope  
accounts-daemon.service  
acpid.service  
alsa-restore.service  
alsa-state.service  
alsa-utils.service  
anacron.service  
apparmor.service
```

UNIT FILE	STATE
proc-sys-fs-binfmt_misc.automount	static
.mount	generated
dev-hugepages.mount	static
dev-mqueue.mount	static
proc-sys-fs-binfmt_misc.mount	disabled
run-vmblock\x2dfuse.mount	enabled
snap-core18-1705.mount	enabled
systemd-ask-password-console.path	static
systemd-ask-password-plymouth.path	static
systemd-ask-password-wall.path	static
session-1.scope	transient
accounts-daemon.service	enabled
acpid.service	disabled
alsa-restore.service	static
alsa-state.service	static
alsa-utils.service	masked
anacron.service	enabled
apparmor.service	enabled

서비스와 소켓(3)

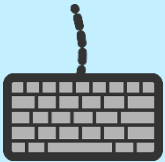
- 소켓

- 서비스는 항상 가동되지만, 소켓은 외부에서 특정 서비스를 요청할 경우에 `systemd`가 구동시킨다, 그리고 요청이 끝나면 소켓도 종료된다.
- 그래서 소켓으로 설정된 서비스를 요청할 때는 처음 연결되는 시간이 앞에서 설명한 서비스에 비교했을 때 약간 더 걸릴 수 있다, 왜냐하면 `systemd`가 서비스를 새로 구동하는 데 시간이 소요되기 때문이다, 이와 같은 소켓의 대표적인 예로 텔넷 서버를 들 수 있다.
- 소켓과 관련된 스크립트 파일은 `/lib/systemd/system/` 디렉터리에 소켓이름.`socket`라는 이름으로 존재한다

- 예전 우분투에서는 소켓과 비슷한 개념으로 `xinetd` 데몬이 주로 사용되었다,
- Ubuntu 20.04 LTS도 `xinetd` 데몬을 지원한다, 하지만 많은 서비스가 `xinetd` 대신에 소켓으로 사용된다,

응급 복구

- 시스템이 부팅이 되지 않을 경우에 수행



<실습17> root 비밀번호 분실 시

- 실습목표
 - root 비밀번호를 분실시, 응급 복구하는 방법을 익힌다.
 - GRUB에서 부팅시 설정을 변경하는 방법을 알아 본다.
- 실습 화면 (GRUB 편집)

```
GNU GRUB  version 2.04

insmod part_gpt
insmod ext2
set root='hd0,gpt1'
search --no-floppy --fs-uuid --set=root 554f406f-1937-4658-8457-59c56229c800
linux /boot/vmlinuz-5.4.0-28-generic root=UUID=554f406f-1937-4658-8457-59c56229c800 ro maybe-ubiquity init=/bin/bash
initrd /boot/initrd.img-5.4.0-28-generic
```

GRUB 부트로더(1)

- GRUB 부트로더의 특징

- 부트 정보를 사용자가 임의로 변경해 부팅할 수가 있다. 즉, 부트 정보가 올바르게 없더라도 수정하여 부팅할 수 있다.
- 다른 여러 가지 운영체제와 멀티부팅을 할 수 있다.
- 대화형 설정을 제공해줘서, 커널의 경로와 파일 이름만 알면 부팅이 가능하다.

- GRUB2의 장점

- 셸 스크립트를 지원함으로써 조건식과 함수를 사용할 수 있다.
- 동적 모듈을 로드할 수 있다.
- 그래픽 부트 메뉴를 지원하며, 부트 스플래시(boot splash) 성능이 개선되었다.
- ISO 이미지를 이용해서 바로 부팅할 수 있다.

GRUB 부트로더(2)

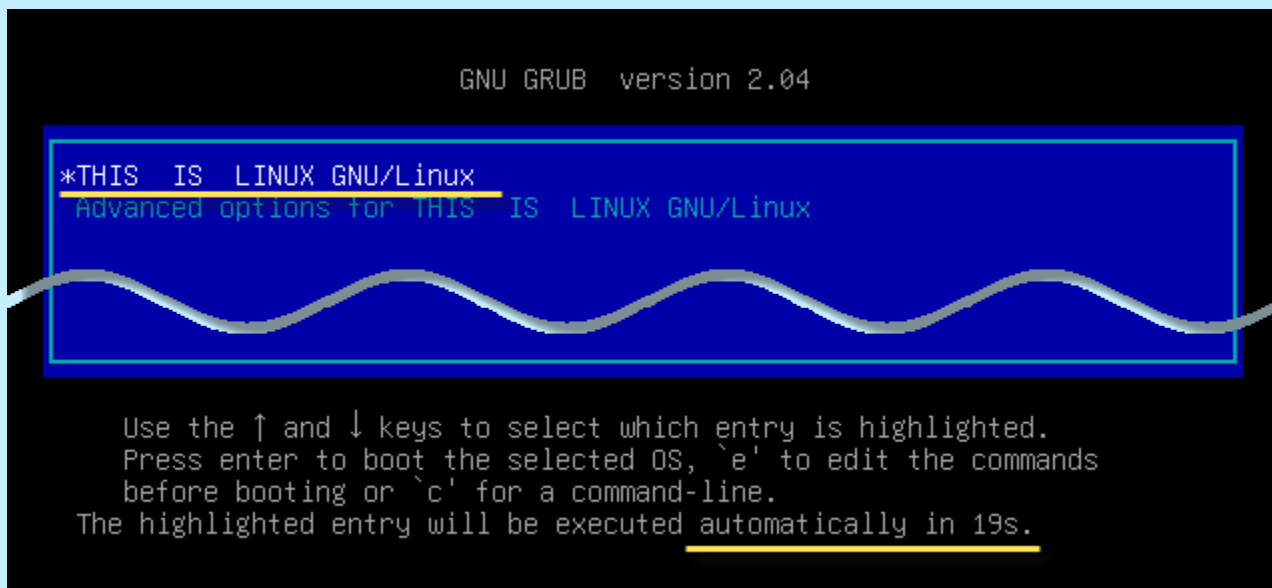
- GRUB2 설정 방법
 - `/boot/grub/grub.cfg` 설정파일 (직접 변경하면 안됨)
 - `/etc/default/grub` 파일과 `/etc/grub.d/` 디렉터리의 파일을 수정한 후에 `'update-grub'` 명령어를 실행해 설정함.
- `/etc/default/grub` 파일

1	<code>GRUB_DEFAULT=0</code>
2	<code>GRUB_TIMEOUT_STYLE=hidden</code>
3	<code>GRUB_TIMEOUT=0</code>
4	<code>GRUB_DISTRIBUTOR='lsb_release -i -s 2> /dev/null echo Debian'</code>
5	<code>GRUB_CMDLINE_LINUX_DEFAULT="maybe-ubiquity"</code>
6	<code>GRUB_CMDLINE_LINUX=""</code>



<실습18> GURB 부트로더 변경

- 실습 목표
 - 부트로더를 변경하는 방법을 익힌다.
 - GRUB에 비밀번호를 지정하는 방법을 익힌다.
- 실습 화면 (제목이 변경된 GRUB)

A screenshot of the GNU GRUB version 2.04 boot screen. The background is black. At the top, the text "GNU GRUB version 2.04" is displayed in white. Below this, a blue rectangular box with a thin blue border contains the text "*THIS IS LINUX GNU/Linux" in white, with a yellow underline under "THIS". Below that, in a smaller font, is "Advanced options for THIS IS LINUX GNU/Linux". A white wavy line separates this box from the text below. At the bottom, white text provides instructions: "Use the ↑ and ↓ keys to select which entry is highlighted. Press enter to boot the selected OS, 'e' to edit the commands before booting or 'c' for a command-line. The highlighted entry will be executed automatically in 19s." The last line is underlined in yellow.

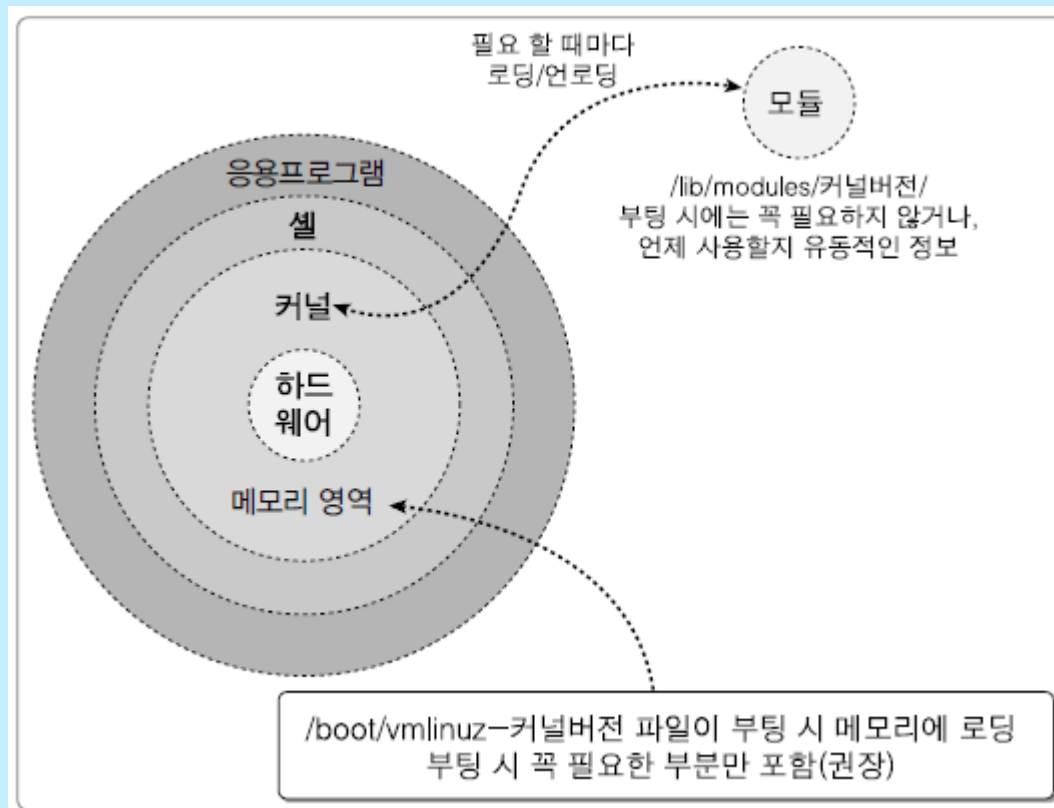
```
GNU GRUB version 2.04

*THIS IS LINUX GNU/Linux
Advanced options for THIS IS LINUX GNU/Linux

Use the ↑ and ↓ keys to select which entry is highlighted.
Press enter to boot the selected OS, 'e' to edit the commands
before booting or 'c' for a command-line.
The highlighted entry will be executed automatically in 19s.
```

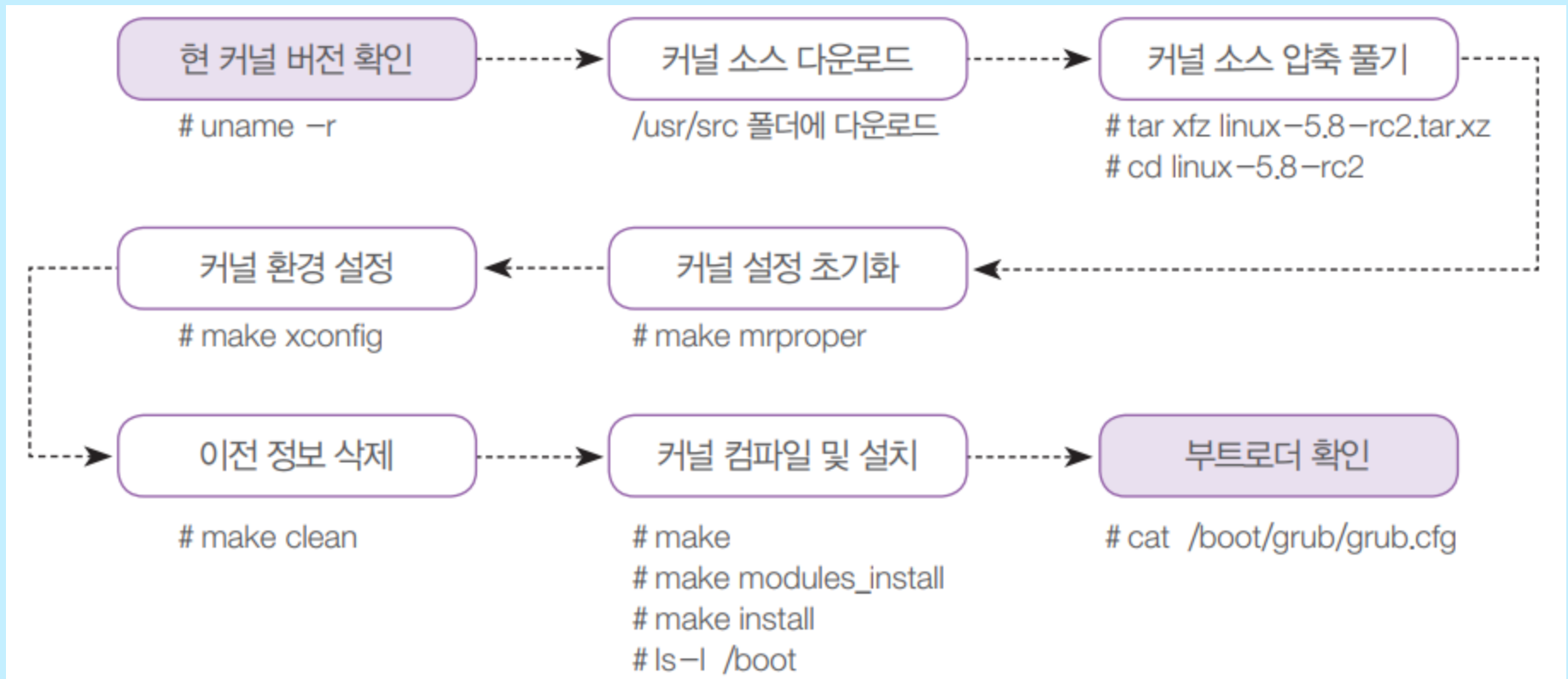
모듈의 개념과 커널 컴파일의 필요성

- 모듈 : 필요할 때마다 호출하여 사용되는 코드



커널 컴파일

- 커널 컴파일 순서





<실습19> 커널 업그레이드

- 실습 목표
 - 최신의 커널로 업그레이드 하는 방법을 익힌다.
- 실습 화면 (컴파일 환경 설정 및 컴파일 결과)

