

이것이 리눅스다

개정판

with  RedHat CentOS 8

4장. 서버를 구축하는 데 알아야 할
필수 개념과 명령어



VMware
실습환경

동영상 강의
무료 제공

145

쪽

시작과 종료

• 종료하는 방법

- ① 바탕 화면의 [오른쪽 위] → [컴퓨터 끄기] → <컴퓨터 끄기>
- ② 터미널/콘솔에서 시스템 종료 명령 입력
“shutdown -P now”, “halt -p”, “init 0”

➤ # 프롬프트는 root 사용자
➤ \$ 프롬프트는 일반 사용자

• 시스템 재부팅

- ① 바탕 화면의 [오른쪽 위] → [컴퓨터 끄기] → <다시 시작>
- ② 터미널/콘솔에서 시스템 재부팅 명령 입력
“shutdown -r now”, “reboot”, “init 6”

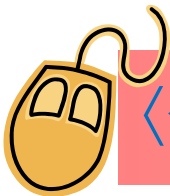
• 로그아웃

- ① 바탕 화면의 [오른쪽 위] → [사용자 이름] → [로그아웃]
- ② 터미널/콘솔에서 시스템 종료 명령 입력
“logout” 또는 “exit”

가상 콘솔

147
쪽

- 쉽게 ‘가상의 모니터’라 생각하면 됨. CentOS는 총 5~6개의 가상 콘솔을 제공.
- 각각의 가상 콘솔로 이동하는 단축키는 Ctrl+ Alt + F3 ~ F6 (Ctrl + Alt + F2는 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 'chvt 가상콘솔번호' 명령을 사용해도 됨.

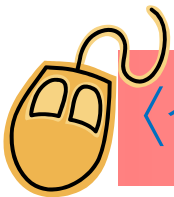
런 레벨(Runlevel)

151
쪽

- ‘init’ 명령어 뒤에 붙는 숫자를 런레벨RunLevel이라고 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



152

쪽

〈실습2〉 시스템에 설정된 런레벨을 변경

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

```
root@localhost:~  
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)  
[root@localhost ~]# cd  
[root@localhost ~]# ls -l /etc/systemd/system/default.target  
lrwxrwxrwx. 1 root root 36 1월 8 19:00 /etc/systemd/system/default.target -> /  
lib/systemd/system/graphical.target  
[root@localhost ~]#
```

자동 완성과 히스토리

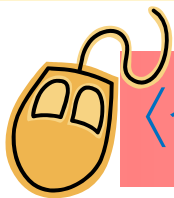
154
쪽

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

예) `cd /etc/sysconfig/network-scripts/` 를 입력하려면
`cd /et[Tab키]sysco[Tab키]net[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

155
쪽

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

```
root@localhost:~  
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)  
[root@localhost ~]# history  
 1  exit  
 2  su -  
 3  passwd  
 4  clear  
 5  gsettings set org.gnome.software download-updates false  
 6  systemctl disable dnf-makecache.service  
 7  systemctl disable dnf-makecache.timer  
 8  exit  
 9  cd /etc/yum.repos.d/  
10  cp backup
```

에디터 사용

158
쪽

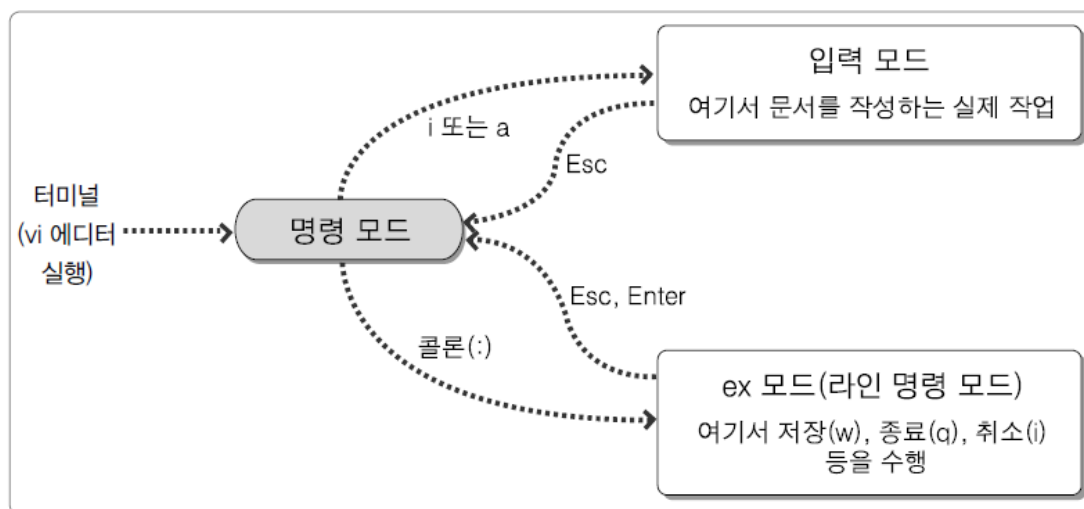
〈실습4〉 에디터를 사용하자

- 실습목표

- gedit의 기본적인 사용법을 익힌다.
- vi의 사용법을 연습한다.

➤ vi는 자주 사용해야 할 기능이므로 반드시 익혀야 한다.

- vi 에디터 사용법 개요도





〈실습5〉 vi의 비정상 종료시 조치법

163

쪽

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

```
root@localhost:~  
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)  
  
E325: ATTENTION  
Found a swap file by the name ".test1.txt.swp"  
  owned by: root   dated: Mon Nov 11 18:37:38 2019  
  file name: ~root/test1.txt  
  modified: YES  
  user name: root   host name: localhost.localdomain  
  process ID: 4070  
If you are using the file already, use ":swap!" to swap file  
to avoid this message.  
"test1.txt" 5L, 140C  
Press ENTER or type command to continue
```

vi 기능 요약

165
쪽

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

- 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([←]와 같은 의미, [H])	j	커서를 아래로 한 칸 이동([↓]와 같은 의미, [J])
k	커서를 위로 한 칸 이동([↑]와 같은 의미, [K])	l	커서를 오른쪽으로 한 칸 이동([→]와 같은 의미, [L])

- 명령 모드에서 삭제, 복사, 붙여넣기

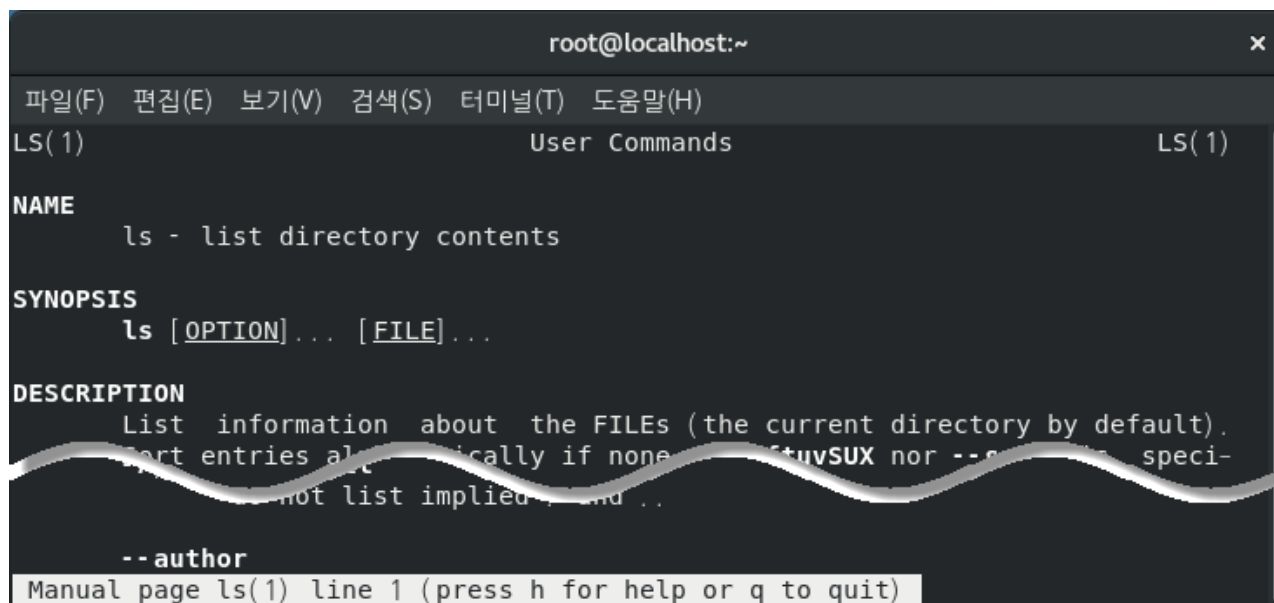
x	현재 커서가 위치한 글자 삭제([Del]과 같은 의미, [X])	X	현재 커서가 위치한 앞 글자 삭제([BackSpace]와 같은 의미, [Shift] + [X])
dd	현재 커서의 행 삭제([D] 연속 두 번 입력)	숫자 dd	현재 커서부터 숫자만큼의 행 삭제(숫자 다음 [D] 연속 두 번 입력)
yy	현재 커서가 있는 행을 복사([Y] 연속 두 번 입력)	숫자 yy	현재 커서부터 숫자만큼의 행을 복사(숫자 다음 [Y] 연속 두 번 입력)
p	복사한 내용을 현재 행 이후에 붙여 넣기([P])	P	복사한 내용을 현재 행 이전에 붙여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

도움말 사용법

167
쪽

➤ man 페이지는 섹션 1~9까지로 나뉨

- “man 명령어”를 사용하면 도움말 출력

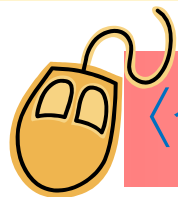


```
root@localhost:~
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)
LS(1) User Commands LS(1)
NAME
  ls - list directory contents
SYNOPSIS
  ls [OPTION]... [FILE]...
DESCRIPTION
  List information about the FILES (the current directory by default).
  Sort entries alphabetically if none of -t, -u, or -x is specified.
  --author
Manual page ls(1) line 1 (press h for help or q to quit)
```

마운트와 CD/DVD 및 USB 메모리의 활용

168
쪽

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



〈실습6〉 CD/DVD 마운트

168

쪽

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

```
현재 활동 (월) 22 : 34 EN [system icons]
CentOS-8-BaseOS-x86_64

mount -t iso9660 /dev/sr0 /media/cdrom
mount -t vfat /dev/sdb1 /media/usb

[root@localhost ~]# ls /media/cdrom/
AppStream BaseOS EFI images isolinux media.repo TRANS.TBL

[root@localhost ~]# ls /media/usb/
config-4.18.0-80.el8.x86_64 'System Volume Information'
python-3.8.0rc1.exe          vs_community_2017.exe
```

리눅스 기본 명령어 (1)

180~184쪽

- ls

Windows의 “dir”과 같은 역할로, 해당 디렉터리에 있는 파일의 목록을 나열
예) # ls /etc/sysconfig

- cd

디렉터리를 이동

예) # cd ../etc/sysconfig

- “.” (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)

180~184쪽

- 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)

180~184쪽

- rmdir

디렉토리를 삭제. (단, 비어 있어야 함)

예) # rmdir 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 anaconda-ks.cfg

- more

텍스트로 작성된 파일을 화면에 페이지 단위로 출력

예) # more anaconda-ks.cfg

리눅스 기본 명령어 (4)

180~184쪽

- less

more와 용도가 비슷하지만 기능이 더 확장된 명령

예) # less anaconda-ks.cfg

- file

File이 어떤 종류의 파일인지를 표시

예) # file anaconda-ks.cfg

- clear

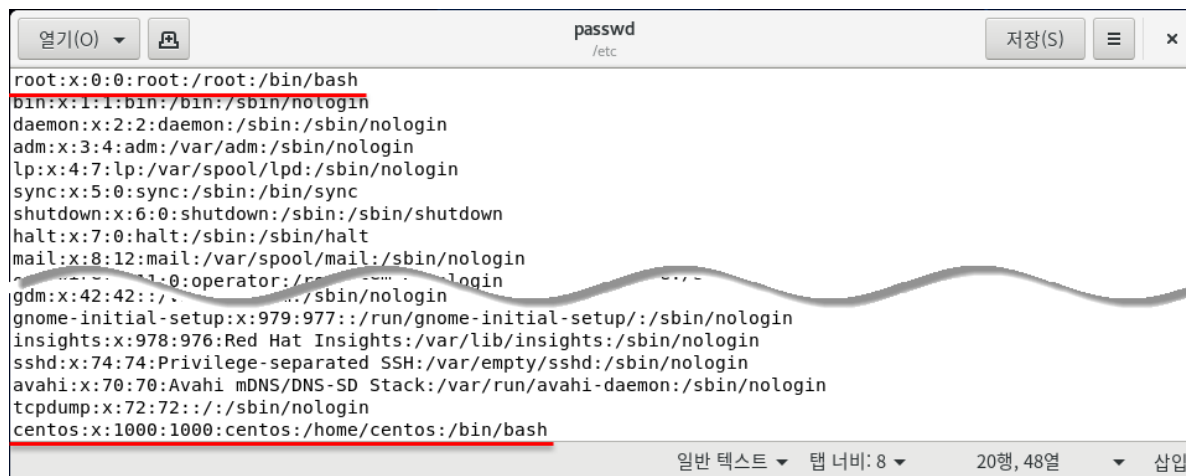
명령창을 깨끗하게 지워줌

예) # clear

사용자와 그룹(1)

184~186쪽

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



```
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
sync:x:5:0:sync:/sbin:/bin/sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
operator:x:11:0:operator:/root:/sbin/nologin
gdm:x:42:42:/:/sbin:/sbin/nologin
gnome-initial-setup:x:979:977:./run/gnome-initial-setup:/sbin/nologin
insights:x:978:976:Red Hat Insights:/var/lib/insights:/sbin/nologin
sshd:x:74:74:Privilege-separated SSH:/var/empty/ssh:/sbin/nologin
avahi:x:70:70:Avahi mDNS/DNS-SD Stack:/var/run/avahi-daemon:/sbin/nologin
tcpdump:x:72:72:./:/sbin/nologin
centos:x:1000:1000:centos:/home/centos:/bin/bash
```

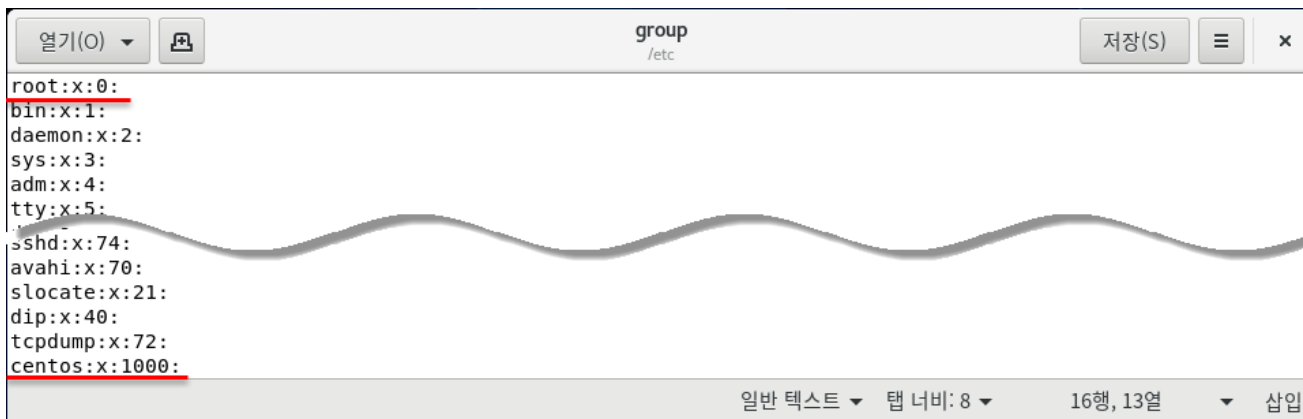
- 각 행의 의미는 다음과 같음

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

사용자와 그룹(2)

184~186쪽

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



```
root:x:0:
bin:x:1:
daemon:x:2:
sys:x:3:
adm:x:4:
tty:x:5:
ssh:x:74:
avahi:x:70:
slocate:x:21:
dip:x:40:
tcpdump:x:72:
centos:x:1000:
```

- 각 행의 의미는 다음과 같음
그룹명:비밀번호:그룹 id:그룹에 속한 사용자명

사용자와 그룹 관련 명령어(1)

186~188쪽

- useradd

새로운 사용자를 추가

예) # useradd newuser

➤ 사용자 생성시 옵션
-u : ID 지정
-g : 그룹 지정
-d : 홈 디렉터리 지정
-s : 셸 지정

- passwd

사용자의 비밀번호를 지정하거나 변경

예) # passwd newuser

- usermod

사용자의 속성을 변경

예) # usermod -g root newuser

- userdel

사용자를 삭제

예) # userdel newuser

사용자와 그룹 관련 명령어(2)

186~188쪽

- **chage**
사용자의 암호를 주기적으로 변경하도록 설정
예) # `chage -m 2 newuser`
- **groups**
현재 사용자가 속한 그룹을 보여줌
예) # `groups`
- **groupadd**
새로운 그룹을 생성
예) # `groupadd newgroup`
- **groupmod**
그룹의 속성을 변경
예) # `groupmod -n newgroup mygroup`

사용자와 그룹 관련 명령어(3)

186~188쪽

- groupdel
그룹을 삭제
예) # groupdel newgroup
- gpasswd
그룹의 암호를 설정하거나, 그룹의 관리를 수행
예) # gpasswd newgroup



〈실습7〉 사용자와 그룹 관리 연습

188
쪽

- 실습목표
 - 사용자와 그룹의 생성/삭제/변경 명령어를 연습한다.
 - 사용자와 그룹의 관련 파일을 파악한다.
- 실습 화면 (/etc/passwd 파일)

```

root@localhost:~
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)
[root@localhost ~]# tail /etc/passwd
setroubleshoot:x:981:979::/var/lib/setroubleshoot:/sbin/nologin
pipewire:x:980:978:PipeWire System Daemon:/var/run/pipewire:/sbin/nologin
gdm:x:42:42::/var/lib/gdm:/sbin/nologin
gnome-initial-setup:x:979:977::/run/gnome-initial-setup:/sbin/nologin
insights:x:978:976:Red Hat Insights:/var/lib/insights:/sbin/nologin
sshd:x:74:74:Privilege-separated SSH:/var/empty/sshd:/sbin/nologin
avahi:x:70:70:Avahi mDNS/DNS-SD Stack:/var/run/avahi-daemon:/sbin/nologin
tcpdump:x:72:72:::/sbin/nologin
centos:x:1000:1000:centos:/home/centos:/bin/bash
user1:x:1001:1001:~/home/user1:/bin/bash
[root@localhost ~]#
  
```

파일과 디렉터리의 소유와 허가권 (1)

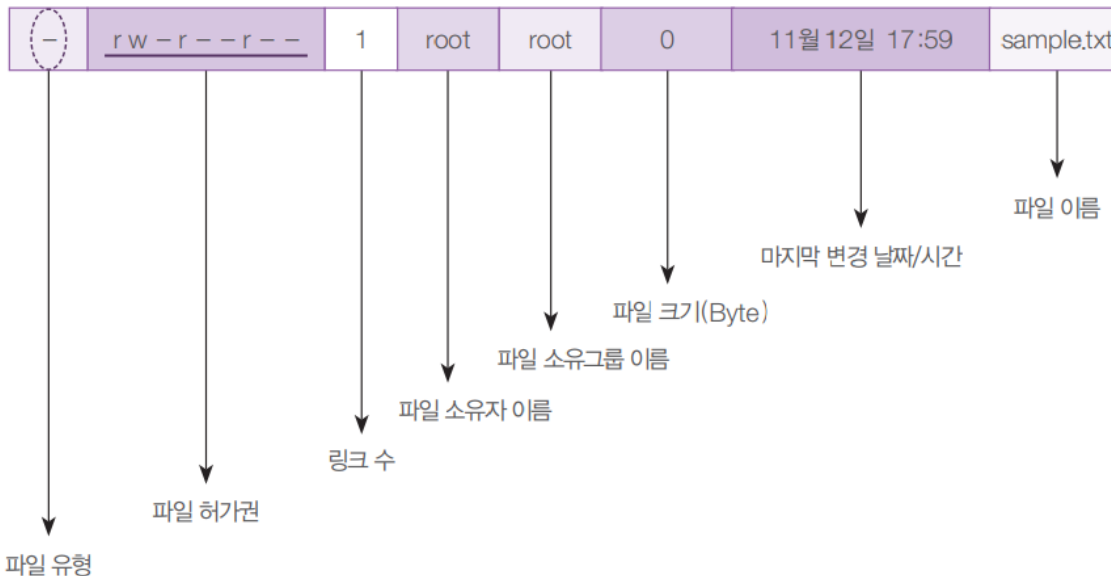
193~195쪽

• 파일의 리스트와 파일 속성

```

root@localhost:~
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)
[root@localhost ~]# touch sample.txt
[root@localhost ~]# ls -l
합 계 8
-rw-----. 1 root root 1493 11월 11 20:02 anaconda-ks.cfg
-rw-r--r--. 1 root root 1648 11월 11 22:08 initial-setup-ks.cfg
-rw-r--r--. 1 root root 0 11월 12 17:59 sample.txt
drwxr-xr-x. 2 root root 6 11월 11 22:09 공개
drwxr-xr-x. 2 root root 6 11월 11 22:09 다운로드
drwxr-xr-x. 2 root root 6 11월 11 22:09 음악

```



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 centos.centos sample.txt 또는
chown centos sample.txt 및 # chgrp centos sample.txt



〈실습8〉 파일의 허가권 및 소유권 연습

196

쪽

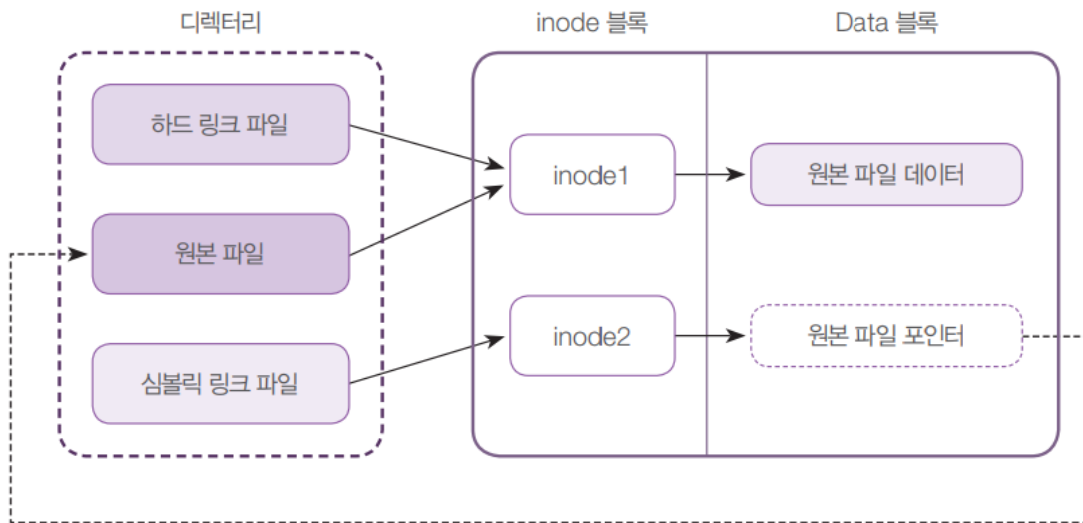
- 실습목표
 - 파일의 허가권에 대한 개념과 명령어를 익힌다.
 - 파일의 소유권에 대한 개념과 명령어를 익히고, 보안에 관련된 내용도 확인해 본다.
- 실습 화면 (소유권 변경 실패)

```
centos@localhost:~  
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)  
[centos@localhost ~]$ whoami  
centos  
[centos@localhost ~]$ chown root.root test  
chown: changing ownership of 'test': 명령을 허용하지 않음  
[centos@localhost ~]$
```

링크

200
쪽

- 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)



〈실습9〉 하드 링크, 심볼릭 링크 연습

201
쪽

- 실습목표
 - 하드 링크, 심볼릭 링크의 개념을 실습을 통해 익힌다.
- 실습 화면 (링크 실습 화면)

```
root@localhost:~/linktest
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)
[root@localhost linktest]# ln basefile hardlink
[root@localhost linktest]# ln -s basefile softlink
[root@localhost linktest]# ls -il
합계 8
1573326 -rw-r--r-- 2 root root 61 2월 1 20:49 basefile
1573326 -rw-r--r-- 2 root root 61 2월 1 20:49 hardlink
1573324 lrwxrwxrwx 1 root root 8 2월 1 20:51 softlink -> basefile
[root@localhost linktest]#
[root@localhost linktest]# cat hardlink
파일 링크를 실습하기 위한 원본 파일입니다.
[root@localhost linktest]# cat softlink
파일 링크를 실습하기 위한 원본 파일입니다.
[root@localhost linktest]#
```

203~206쪽

프로그램 설치를 위한 RPM (1)

- RPM(Redhat Package Manager)
 - Windows의 “setup.exe”와 비슷한 설치 파일
 - 확장명은 *.rpm이며, 이를 ‘패키지(Package)’라고 부름.
- 파일의 의미

```
root@localhost:/run/media/root/CentOS-8-BaseOS-x86_64/BaseOS/
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)
[root@localhost Packages]# pwd
/run/media/root/CentOS-8-BaseOS-x86_64/BaseOS/Packages
[root@localhost Packages]# ls -l gzip*
-r--r--r-- 1 root root 169968 7월  2 22:22 gzip-1.9-4.el8.x86_64.rpm
[root@localhost Packages]#
```

- x86_64 : 64비트 CPU
- i386~686 : 32비트 CPU
- src : 소스
- noarch : 모든 CPU

패키지이름-버전-릴리즈번호.CentOS버전.아키텍처.rpm

- 패키지이름 : gzip → 패키지(프로그램)의 이름
- 버전 : 1.9 → 대개 3자리수로 구성. 주버전, 부버전, 패치버전
- 릴리즈번호 : 4 → 문제점을 개선할 때마다 붙여지는 번호
- CentOS 버전 : el8 → CentOS에서 배포할 경우에 붙여짐
- 아키텍처 : x86_64 → 64비트 CPU를 의미

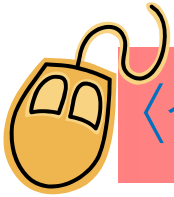
프로그램 설치를 위한 RPM (2)

- 자주 사용하는 RPM 명령어 옵션
 - 설치 : `rpm -Uvh 패키지파일이름.rpm`
 - U → (대문자) 패키지가 설치/업그레이드
 - v → 설치과정의 확인
 - h → 설치진행과정을 "#"마크로 화면에 출력
 - 삭제 : `rpm -e 패키지이름`
 - 이미 설치된 패키지 질의
 - `rpm -qa` 패키지 이름 → 패키지가 설치되었는지 확인
 - `rpm -qf 파일의절대경로`
→ 파일이 어느 패키지에 포함된 것인지 확인
 - 아직 설치되지 않은 rpm 파일에 대한 질의
 - `rpm -qlp 패키지파일이름.rpm`
→ 패키지 파일에 어떤 파일들이 포함되었는지 확인
 - `rpm -qip 패키지파일이름.rpm` → 패키지 파일의 상세정보

프로그램 설치를 위한 RPM (3)

203~206쪽

- RPM 단점
 - ‘의존성’ 문제
 - A패키지가 설치되기 위해서 B패키지가 필요할 경우, RPM으로는 해결이 까다로움.
 - 이를 해결하기 위해 DNF(또는 YUM)가 등장함



〈실습10〉 rpm 패키지 설치 연습

208
쪽

- 실습목표
 - RPM 명령어의 사용법을 익힌다.
 - RPM 파일의 의존성 문제를 파악한다.
- 실습화면 (mc 패키지 설치)

```
root@localhost:/run/media/root/CentOS-8-BaseOS-x86_64/AppStream/Packages x
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)
[root@localhost Packages]# rpm -Uvh mc-4.8.19-9.el8.x86_64.rpm
경고: mc-4.8.19-9.el8.x86_64.rpm: Header V3 RSA/SHA256 Signature, key ID 8483c65
d: NOKEY
Verifying... ##### [100%]
준비 중... ##### [100%]
Updating / installing...
 1:mc-1:4.8.19-9.el8 ##### [100%]
[root@localhost Packages]#
```


편리한 패키지 설치, DNF (1)

210~212쪽

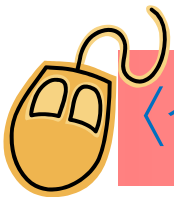
- DNF(Dandified dnf) 개념

- “rpm”명령의 패키지 의존성 문제를 완전하게 해결됨.
- 인터넷을 통하여 필요한 파일을 저장소(Repository)에서 자동으로 모두 다운로드해서 설치하는 방식
- CentOS 7은 YUM, CentOS 8은 YUM이 개선된 DNF 명령을 사용.

- DNF 기본적인 사용법

➤ 저장소의 URL은 “/etc/yum.repos.d/” 디렉터리

- 기본 설치 : `dnf install 패키지이름`
 - 주로 “`dnf -y install 패키지이름`”으로 사용
 - “-y”는 사용자의 확인을 모두 “yes”로 간주하고 설치를 진행한다는 옵션
- RPM 파일 설치 : `dnf install rpm파일이름.rpm`
- 업데이트 가능한 목록 보기 : `dnf check-update`
- 업데이트 : `dnf update 패키지이름`
- 삭제 : `dnf remove 패키지이름`
- 정보 확인 : `dnf info 패키지이름`



〈실습11〉 dnf 기본 연습

212

쪽

- 실습목표
 - 의존성 문제가 있는 패키지를 “dnf”명령으로 설치해 본다.
- 실습 화면 (의존성 패키지 자동 다운로드)

```
root@localhost:~  
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)  
[root@localhost ~]# dnf install mysql-errmsg  
마지막 메타 데이터 만료 확인 : 0:01:15 전에 2019년 11월 13일 (수) 오후 12시 06분  
46초.  
종속성이 해결되었습니다.  
=====
```

구러미	아키텍처	버전	리포지토리	크기
=====				
Installing:				
mysql-errmsg	x86_64	8.0.17-3.module_el8.0.0+181+899d6349	AppStream	557 k
종속성 설치:				
mariadb-connector-c-config				
	noarch	3.0.7-1.el8	AppStream	13 k
mysql-common	x86_64	8.0.17-3.module_el8.0.0+181+899d6349	AppStream	143 k
Enabling module streams:				
mysql		8.0		
거래 요약				
=====				
설치 3 구러미				
=====				
총 다운로드 크기 : 713 k				
설치 크기 : 8.3 M				
이게 괜찮습니까 [y / N] :				

편리한 패키지 설치, DNF (2)

214~219쪽

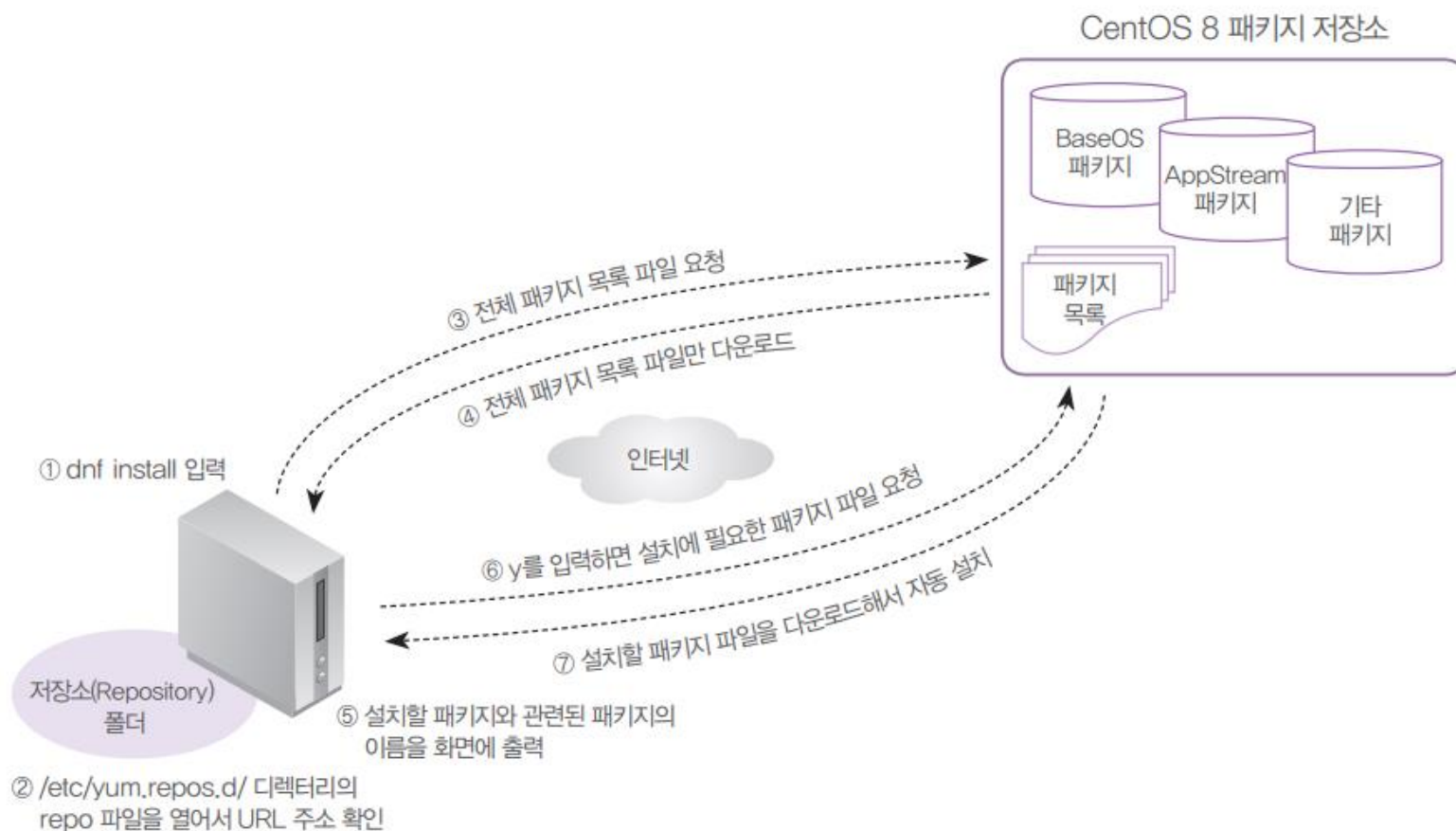
- DNF 고급 사용법

- 패키지 그룹 설치
 - `dnf groupinstall` “패키지그룹이름”
- 패키지 리스트 확인
 - `dnf list` 패키지이름
- 특정 파일이 속한 패키지 이름 확인
 - `dnf provides` 파일이름
- GPG 키 검사 생략
 - `dnf install --nogpgcheck rpm파일이름.rpm`
 - CentOS 8에서 인증되지 않은 패키지를 강제로 설치할 때 사용
- 기존 저장소 목록 지우기
 - `dnf clean all`

편리한 패키지 설치, DNF (3)

214~219쪽

- DNF 작동 방식 설정 파일
 - ‘dnf install 패키지이름’ 명령이 작동하는 방식



편리한 패키지 설치, DNF (4)

214~219쪽

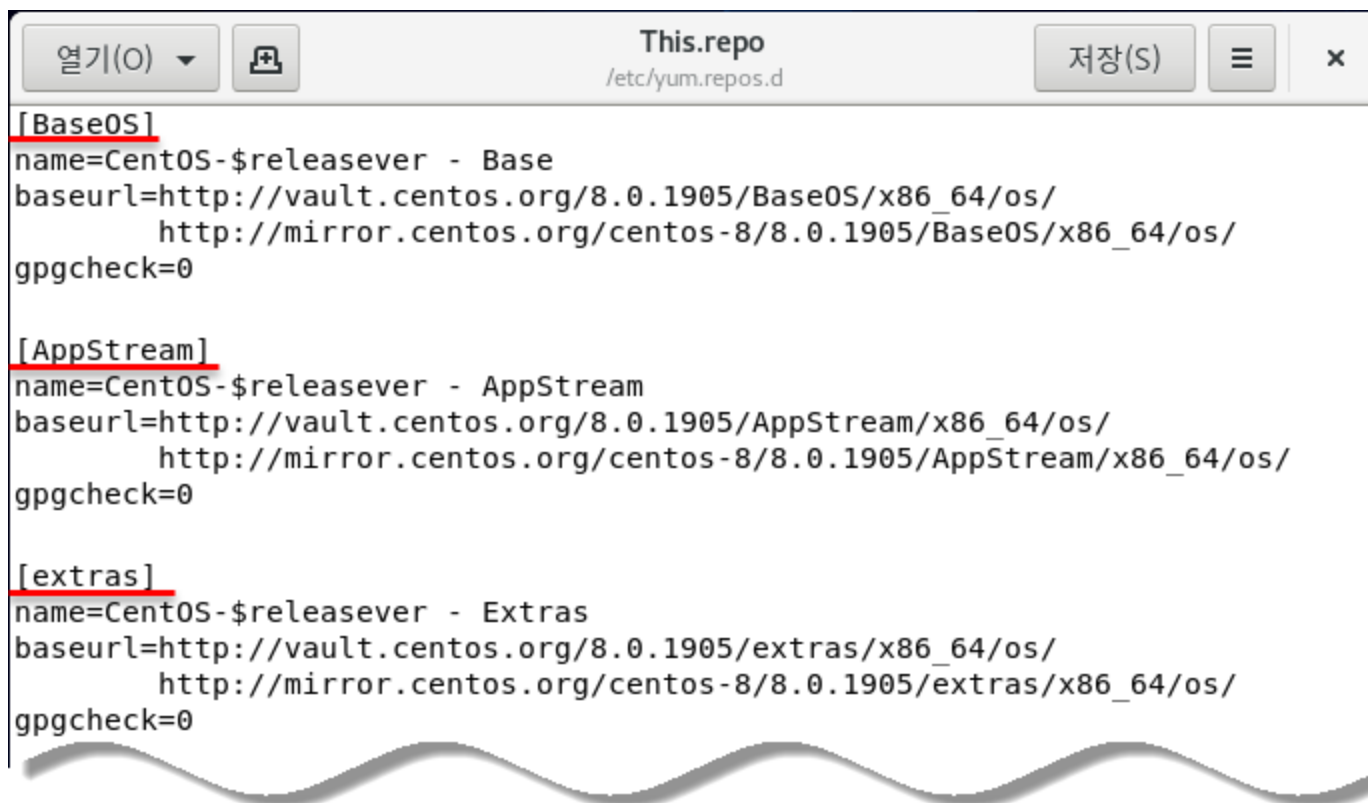
- DNF 작동 방식 설정 파일

- /etc/yum.conf 파일 : 특별히 변경할 필요 없음
- /etc/yum.repos.d/ 디렉터리
 - dnf 명령을 입력했을 때 검색하게 되는 네트워크의 주소가 들어 있는 여러 개의 파일이 있음
- /etc/yum.repos.d/ 디렉터리의 *.repo 파일
 - CentOS 8의 1905버전만 설치되도록 하기 위해 /etc/yum.repos.d/ 폴더의 내용을 모두 삭제하고 This.repo 파일을 새로 생성하여 남겨둌.
그리고 This.repo 파일에 BaseOS, AppStream 등의 저장소를 별도로 적어놓음.

편리한 패키지 설치, DNF (5)

214~219쪽

- This.repo 파일



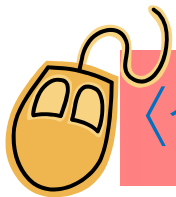
```

This.repo
/etc/yum.repos.d

[BaseOS]
name=CentOS-$releasever - Base
baseurl=http://vault.centos.org/8.0.1905/BaseOS/x86_64/os/
        http://mirror.centos.org/centos-8/8.0.1905/BaseOS/x86_64/os/
gpgcheck=0

[AppStream]
name=CentOS-$releasever - AppStream
baseurl=http://vault.centos.org/8.0.1905/AppStream/x86_64/os/
        http://mirror.centos.org/centos-8/8.0.1905/AppStream/x86_64/os/
gpgcheck=0

[extras]
name=CentOS-$releasever - Extras
baseurl=http://vault.centos.org/8.0.1905/extras/x86_64/os/
        http://mirror.centos.org/centos-8/8.0.1905/extras/x86_64/os/
gpgcheck=0
```



〈실습12〉 DNF 고급 실습

219

쪽

- 실습목표

- CentOS DVD에서 직접 설치하기 위한 방법을 알아 본다.
- *.repo 파일을 직접 편집해 본다.
- 네트워크와 DVD를 동시에 사용하는 효율적인 방법을 알아 본다.
- ‘패키지 그룹’을 설치하는 방법을 익힌다.

- 실습 화면 (네트워크와 DVD를 동시 사용)

```
root@localhost:/etc/yum.repos.d
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)
[root@localhost yum.repos.d]# dnf install mc help2man
CentOS DVD AppStream          4.2 MB/s | 4.3 kB    00:00
CentOS DVD BaseOS             3.8 MB/s | 3.9 kB    00:00
CentOS-8 - PowerTools         240 kB/s | 1.8 MB    00:07
종속성이 해결되었습니다.
=====
꾸러미      아키텍처  버전              리포지토리      크기
=====
Installing:
mc           x86_64          1:4.8.19-9.el8    DVD-AppStream    1.9 M
help2man     noarch          1.47.6-1.el8      Network-PowerTools 186 k
=====
거래 요약
=====
설치 2 꾸러미

총합 크기: 2.1 M
      크기: 10
```

비타민 퀴즈 4-1

226

쪽

- ▶ 힌트 1 : Server(B)를 초기화한 후, 메모리를 1.5GB (=1536MB) 이상으로 올리고 부팅한다.
- ▶ 힌트 2 : 패키지 환경 그룹 중에서 `-x rpm` 옵션을 추가해서 Workstation을 설치하면 된다
- ▶ 힌트 3 : 설치 완료 후 재부팅하고 `startx` 명령으로 X 윈도를 시작한다

파일의 압축과 묶기 (1)

226~229쪽

- 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
 - bunzip2 : “bzip2 -d” 옵션과 동일한 명령어
 - gzip : 확장명 gz으로 압축을 하거나 풀어준다
예) gzip 파일명
gzip -d 파일명.gz
 - gunzip : “gzip -d” 옵션과 동일한 명령어

파일의 압축과 묶기 (2)

226~229쪽

- 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/sysconfig/ → 묶기
```

```
# tar cvfJ my.tar.xz /etc/sysconfig/ /etc/sysconfig/ → 묶기 + xz 압축
```

```
# tar xvf my.tar → tar 풀기
```

```
# tar xvfJ my.tar.xz /etc/sysconfig/ → xz 압축 해제 + tar 풀기
```

파일 위치 검색

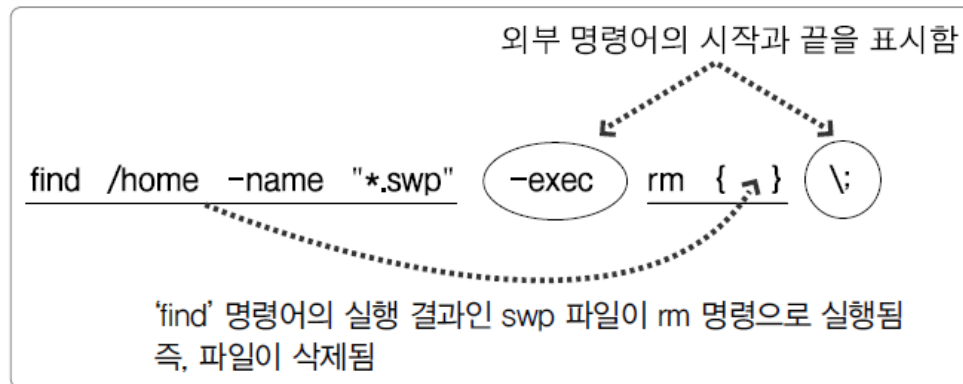
229~230쪽

- 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 +10k -size -100k
```

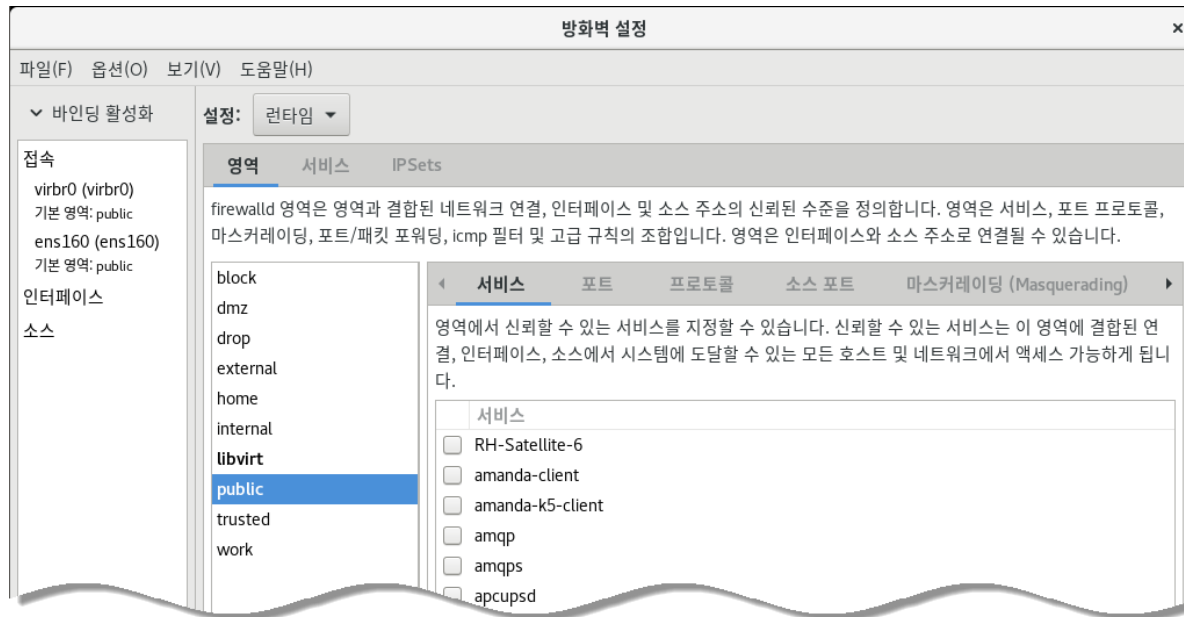
```
# find /home -name "*.swp" -exec rm { } \;
```



- which 실행파일이름 : PATH에 설정된 디렉터리만 검색
- whereis 실행파일이름 : 실행 파일, 소스, man페이지 파일까지 검색
- locate 파일이름 : 파일 목록 데이터베이스에서 검색

시스템 설정 231~232쪽

- 표준 시간대 변경(설정 >> [날짜 및 시각])
- 네트워크 설정 (nmtui)
- 방화벽 설정(firewall-config)



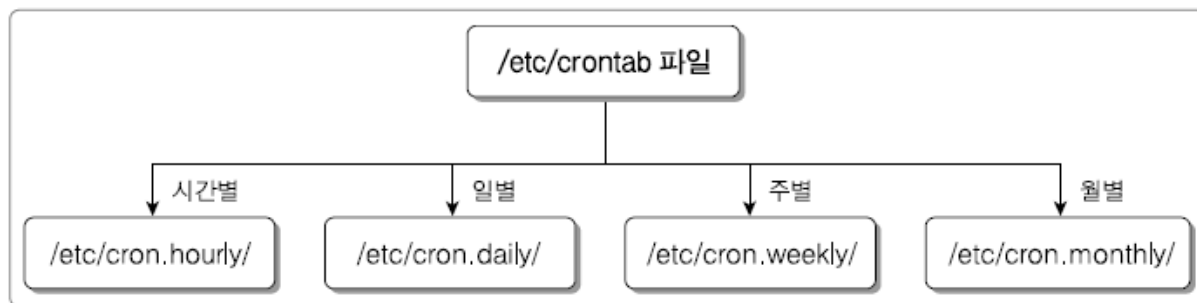
- 서비스 설정(ntsystsv)

CRON과 AT (1)

233~234쪽

- cron

- 주기적으로 반복되는 일을 자동적으로 실행될 수 있도록 설정
- 관련된 데몬(서비스)은 “crond”, 관련 파일은 “/etc/crontab”



- /etc/crontab 예

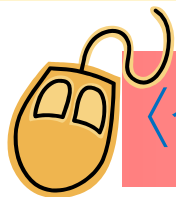
- 01 * * * * root run-parts /etc/cron.hourly
- 02 4 * * * root run-parts /etc/cron.daily
- 03 4 * * 0 root run-parts /etc/cron.weekly
- 42 4 1 * * root run-parts /etc/cron.monthly

- 첫 줄은 매시간 1분에 /etc/cron.hourly 디렉터리 안에 있는 명령들을 자동으로 실행한다

CRON과 AT (2)

233~234쪽

- at
 - cron은 주기적으로 반복되는 작업을 예약하는 것이지만, at는 일회성 작업을 예약
 - 사용 예
 - 예약 : # at <시간>
 - 예) # at 3:00am tomorrow → 내일 새벽 3시
 - # at 11:00pm January 30 → 1월 30일 오후 11시
 - # at now + 1 hours → 1시간 후
 - at> 프롬프트에 예약 명령어 입력 후 [Enter]
 - 완료되면 [Ctrl] + [D]
 - 확인 : # at -l
 - 취소 : # atrm <작업번호>



〈실습13〉 cron, at 실습

234
쪽

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

```
root@localhost:~  
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)  
[root@localhost ~]# date 011503002027  
2027. 01. 15. (금) 03:00:00 KST  
[root@localhost ~]# systemctl restart crond  
[root@localhost ~]# ls -l /backup  
합계 976  
-rw-r--r-- 1 root root 996929 1월 15 03:01 backup-01.15.tar.xz  
[root@localhost ~]# date 021503002027  
2027. 02. 15. (월) 03:00:00 KST  
[root@localhost ~]# systemctl restart crond  
[root@localhost ~]# ls -l /backup  
합계 1952  
-rw-r--r-- 1 root root 996929 1월 15 03:01 backup-01.15.tar.xz  
-rw-r--r-- 1 root root 996929 2월 15 03:01 backup-02.15.tar.xz  
[root@localhost ~]#
```

비타민 퀴즈 4-2

238

쪽

▶ 힌트 없음

네트워크 관련 필수 개념 (1)

238~242쪽

- 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 hanbit.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)

238~242쪽

- 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)

238~242쪽

- 리눅스에서의 네트워크 장치 이름
 - CentOS 8은 랜카드를 ens160 인식함
 - 이전버전에서는 eth0, eth1, ens32, ens33 등으로 인식했음.
 - VMware에 CentOS를 설치할 경우에는 VMware 버전에 따라서 완전히 다른 이름으로 인식할 수도 있음.

중요한 네트워크 관련 명령어 (1)

242~244쪽

- 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(ens160)의 설정
 - 텍스트 기반으로 작동함
- systemctl <start/stop/restart/status> NetworkManager
 - 네트워크의 설정을 변경한 후에, 변경된 내용을 시스템에 적용시키는 명령어

➤ X 윈도의 그래픽 모드를 사용하려면 'gnome-control-center network' 명령어를 사용한다.

중요한 네트워크 관련 명령어 (2)

242~244쪽

- ifup <장치이름> 및 ifdown <장치이름>
 - 네트워크 장치를 On 또는 Off 시키는 명령어
- ifconfig <장치이름>
 - 장치의 IP주소 설정 정보를 출력
- nslookup
 - DNS 서버의 작동을 테스트하는 명령어
- ping <IP주소 또는 URL>
 - 해당 컴퓨터가 네트워크상에서 응답하는지를 테스트하는 간편한 명령어

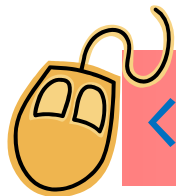
244

쪽

네트워크 설정과 관련된 주요 파일

- /etc/sysconfig/network
 - 네트워크의 기본적인 정보가 설정되어 있는 파일
- /etc/sysconfig/network-scripts/ifcfg-ens160
 - ens32 장치에 설정된 네트워크 정보가 모두 들어 있는 파일
- /etc/resolv.conf
 - DNS 서버의 정보 및 호스트 이름이 들어 있는 파일
- /etc/hosts
 - 현 컴퓨터의 호스트 이름 및 FQDN이 들어 있는 파일

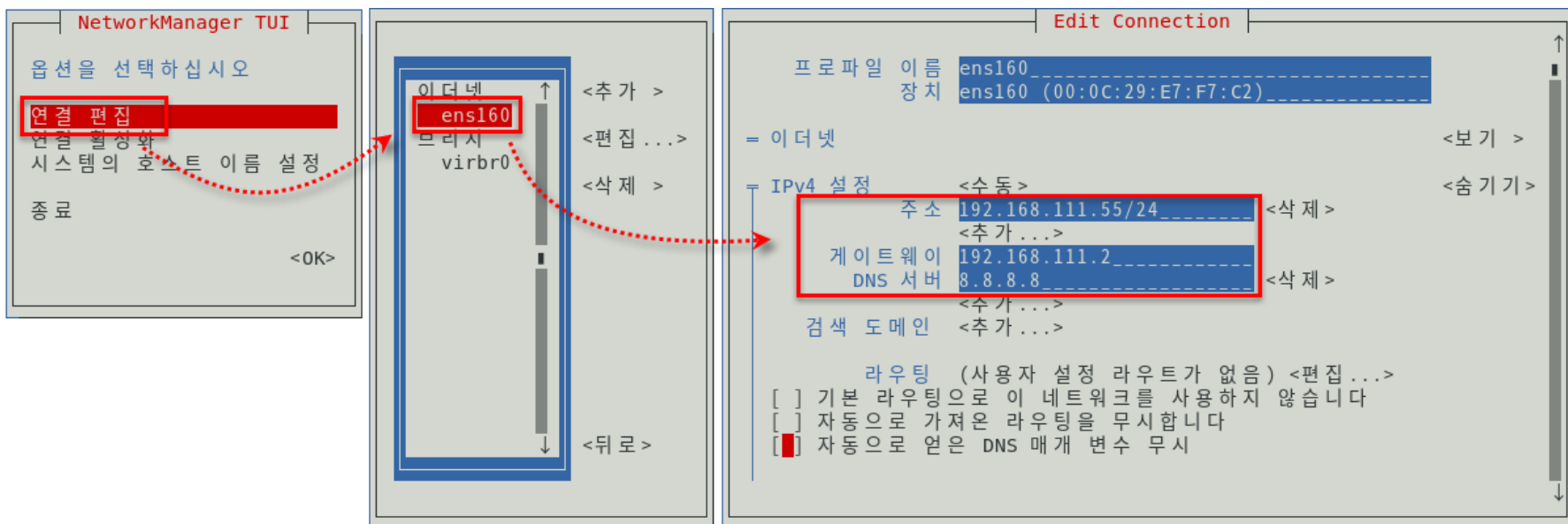
➤ 위 4개 파일을 직접 편집하면 "nmcli"를 사용하지 않아도 됨.



<실습14> nmtui 명령

244
쪽

- 실습목표
 - nmtui 명령의 작동을 이해한다.
 - 네트워크 관련 파일들의 내용을 확인한다.
 - DNS의 작동을 /etc/resolv.conf 파일과 연관해서 이해한다.
- 실습 화면 (네트워크 설정)



252

쪽

네트워크 보안을 위한 SELinux

- SELinux(Security Enhanced Linux)는 보안에 취약한 리눅스를 보호하기 위해 탄생
- 강제(Enforcing), 허용(Permissive), 비활성(Disabled) 세 가지 레벨
- 설정 파일인 /etc/sysconfig/selinux를 편집하거나, system-config-selinux 명령으로 설정
- 작동 방식
 - ‘강제(Enforcing)’는 시스템 보안에 영향을 미치는 기능이 감지되면 아예 그 기능이 작동되지 않도록 시스템에서 막아줌.
 - ‘허용(Permissive)’은 시스템 보안에 영향을 미치는 기능이 감지되면 허용은되지만 그 내용이 로그에 남음.
 - ‘비활성(Disabled)’은 SELinux를 사용하지 않음

파이프, 필터, 리디렉션

253~254쪽

- 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)

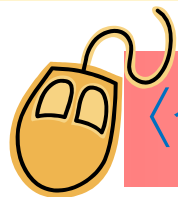
254~255쪽

- 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)

254~255쪽

- 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



〈실습15〉 프로세스 연습

256

쪽

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

```
root@localhost:~
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)
[root@localhost ~]# gedit
[root@localhost ~]# gedit &
[1] 7428
[root@localhost ~]#
```

서비스와 소켓(1)

259~261쪽

• 서비스

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

```
root@localhost:/usr/lib/systemd/system
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)
[root@localhost system]# pwd
/usr/lib/systemd/system
[root@localhost system]# ls *.service
ModemManager.service
NetworkManager-dispatcher.service
NetworkManager-wait-online.service
NetworkManager.service
acpid.service
analog-daemon.service
console-getty.service
container-getty@.service
cpupower.service
crond.service
cups-browsed.service
cups.service
dbus-org.freedesktop.hostname1.service
nis-domainname.service
numad.service
ostree-finalize-staged.service
ostree-prepare-root.service
oslinux-mount.service
k.service
selinux-autoflask.service
serial-getty@.service
smartd.service
speech-dispatcherd.service
spice-vdagentd.service
sshd-keygen@.service
sshd.service
```

서비스와 소켓(2)

259~261쪽

- 소켓

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

- 예전 CentOS에서는 소켓과 비슷한 개념으로 xinetd 데몬이 주로 사용되었다.
- CentOS 8도 xinetd 데몬을 지원한다. 하지만 많은 서비스가 xinetd 대신에 소켓으로 사용된다.

응급 복구

256
쪽

- 시스템이 부팅이 되지 않을 경우에 수행

〈실습16〉 root 비밀번호 분실 시

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

```
load_video
set gfx_payload=keep
insmod gzio
linux ($root)/boot/vmlinuz-4.18.0-80.el8.x86_64 root=UUID=314207a5-5f68-405d-a\
787-7cd8cd8a66f6 ro crashkernel=auto resume=UUID=dad90950-01b5-4d05-b363-87480\
5a29721 vga=773 init=/bin/sh
initrd ($root)/boot/initramfs-4.18.0-80.el8.x86_64.img $tuned_initrd
```

Press Ctrl-x to start, Ctrl-c for a command prompt or Escape to
discard edits and return to the menu. Pressing Tab lists
possible completions.

GRUB 부트로더(1)

265~267쪽

- GRUB 부트로더의 특징

- 부트 정보를 사용자가 임의로 변경해 부팅할 수 있다. 즉 부트 정보가 올바르지 않더라도 수정하여 부팅할 수 있다.
- 다른 여러 가지 운영체제와 멀티부팅을 할 수 있다.
- 대화형 설정을 제공하므로 커널 경로와 파일 이름만 알면 부팅이 가능.

- GRUB2의 장점

- 셸 스크립트를 지원함으로써 조건식과 함수를 사용할 수 있다.
- 동적 모듈을 로드할 수 있다.
- 그래픽 부트 메뉴를 지원하며 부트 스플래시 성능이 개선되었다.
- ISO 이미지를 이용해서 바로 부팅할 수 있다.
- 설정 파일의 형식이 변경되었지만 더 향상된 내용을 포함할 수 있다.

GRUB 부트로더(2)

265~267쪽

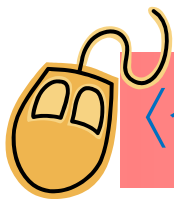
- GRUB2 설정 방법

- /boot/grub2/grub.cfg 설정파일 (직접 변경하면 안됨)
- /etc/default/grub 파일과 /etc/grub.d/ 디렉터리의 파일을 수정한 후에 'grub2-mkconfig' 명령어를 실행해 설정함.

- /etc/default/grub 파일

➤ 파일의 설명은 267쪽 참조

1	GRUB_TIMEOUT=5
2	GRUB_DISTRIBUTOR="\$(sed 's, release .*\$,g' /etc/system-release)"
3	GRUB_DEFAULT=saved
4	GRUB_DISABLE_SUBMENU=true
5	GRUB_TERMINAL_OUTPUT="console"
6	GRUB_CMDLINE_LINUX="crashkernel=auto resume=UUID=장치코드고유번호 rhgb quiet"
7	GRUB_DISABLE_RECOVERY="true"
8	GRUB_ENABLE_BLSCFG=true



〈실습17〉 GURB 부트로더 변경

267

쪽

- 실습목표
 - 부트로더를 변경하는 방법을 익힌다.
 - GRUB에 비밀번호를 지정하는 방법을 익힌다.
- 실습 화면 (대기 시간이 변경된 GRUB)

```
CentOS Linux (4.18.0-80.el8.x86_64) 8 (Core)
CentOS Linux (0-rescue-5866bf09495b4feb9c141b603ae87873) 8 (Core)

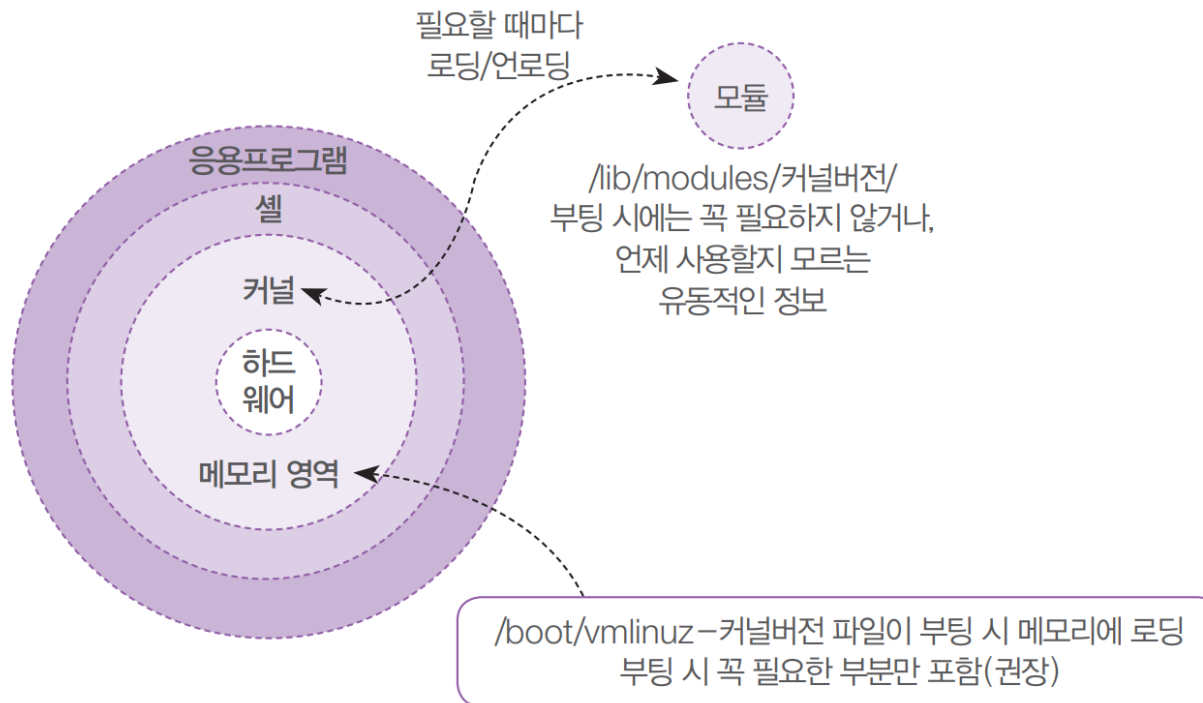
Use the ↑ and ↓ keys to change the selection.
Press 'e' to edit the selected item, or 'c' for a command prompt.
The selected entry will be started automatically in 19s.
```

270

쪽

모듈의 개념과 커널 컴파일의 필요성

- 모듈 : 필요할 때마다 호출하여 사용되는 코드

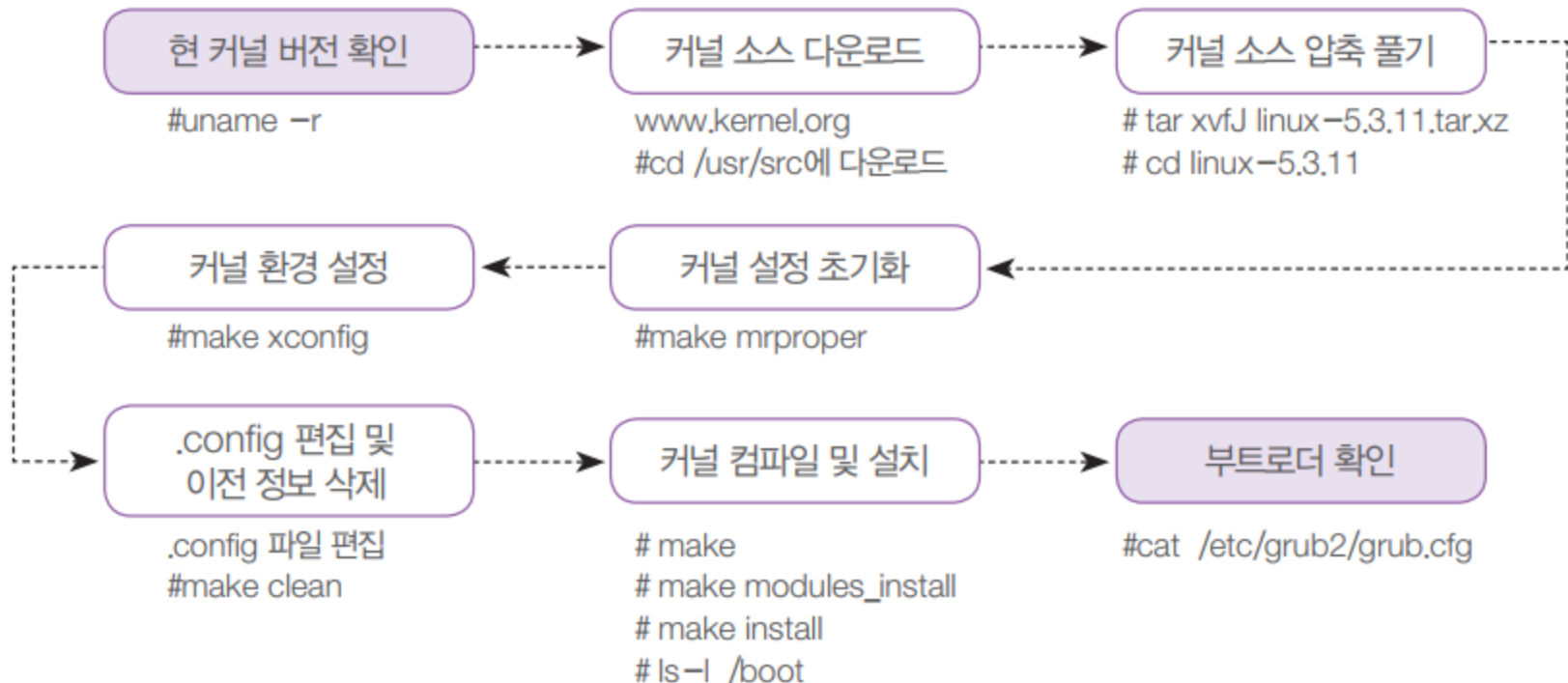


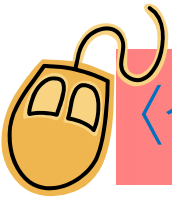
272

쪽

커널 컴파일(업그레이드)

- 커널 컴파일 순서

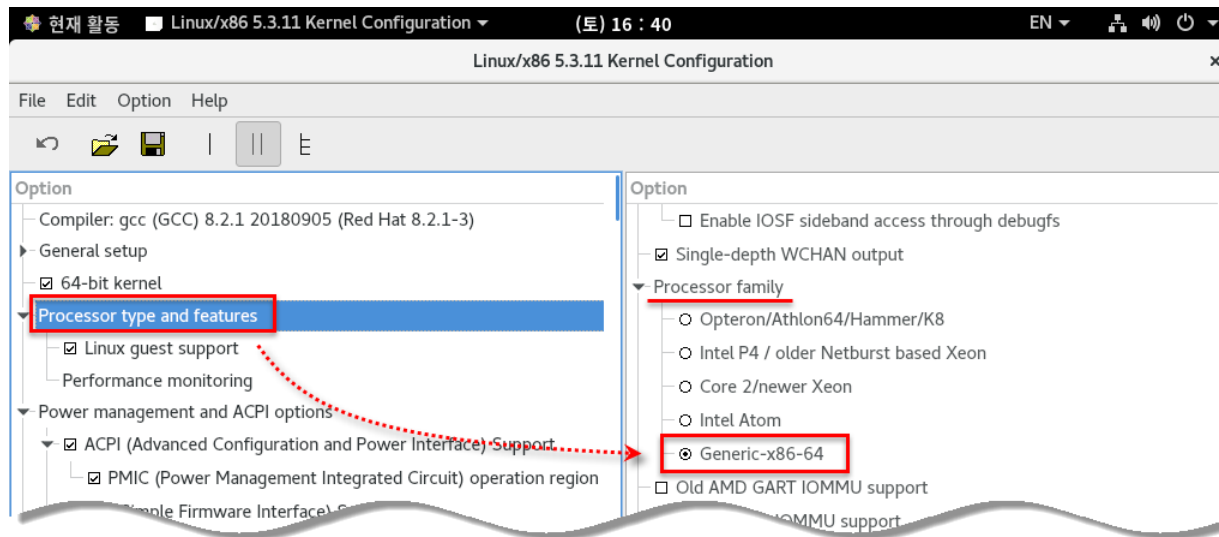




<실습18> 커널 업그레이드

273
쪽

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



```
CentOS Linux (5.3.11) 8 (Core)
CentOS Linux (4.18.0-80.el8.x86_64) 8 (Core)
CentOS Linux (0-rescue-5866bf09495b4feb9c141b603ae87873) 8 (Core)
```

비타민 퀴즈 4-3

280

쪽

▶ 힌트 1 : 추가 설치할 패키지 → bc, gcc, gcc-c++, make, ncurses-devel, perl, bison, flex, elfutils-libelf-devel, openssl-devel

▶ 힌트 2 : 다운로드 는 wget

<https://mirrors.edge.kernel.org/pub/linux/kernel/v5.x/linux-5.3.11.tar.xz>

명령을 사용.

▶ 힌트 3 : make xconfig 명령 대신 make menuconfig 명령을 사용한다.

▶ 힌트 4 : .config 파일을 수정한다.

비타민 퀴즈 4-4

280

쪽

▶ 힌트 0 : Client의 메모리를 2GB(=2048 MB) 정도로 올려서 작업하자. 우선 make 패키지를 설치한다.

▶ 힌트 1 : 다운로드는 <https://www.tuxera.com/community/ntfs-3g-download/>에서 소스파일을 다운로드한다. (파일명 : ntfs-3g_ntfsprogs-xxxx.xx.xx.tgz)

▶ 힌트 2 : 소스를 컴파일 한다.

`./configure; make; make install`

▶ 힌트 3 : 마운트는 다음 명령으로 한다. (필자와 동일하게 Client를 설치했다면 USB 장치는 /dev/sda1 이다)

`mount -t ntfs-3g USB장치이름 마운트할폴더`