

3장

Linux 기본 환경

전체 내용

Terminal
실행하기

Terminal에
나타나는 Prompt
기호 이해하기

기본적인 명령어
사용법

RunLevel 이해
및 System
Shutdown 하기

리눅스 시스템의
Booting 과정

1 – Terminal 실행하기

Terminal 실행하기

Terminal 전환하기

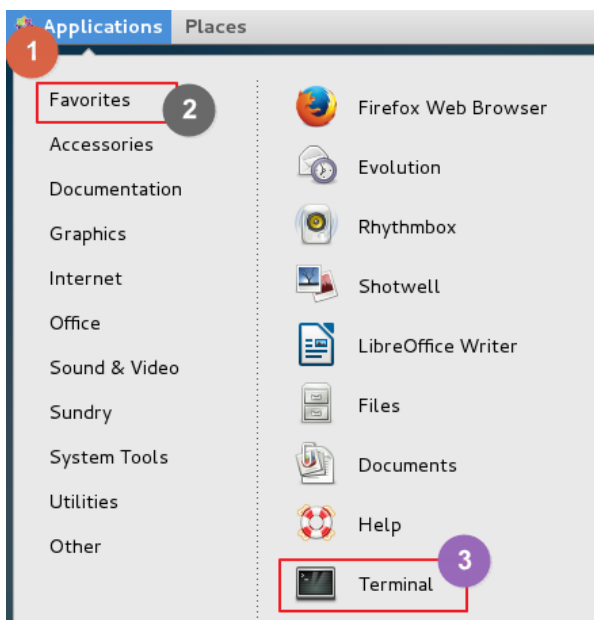
어떻게 Linux 시스템에 접속했는지 확인하기

TTY 및 PTS란 무엇인가?

1 – Terminal 실행하기

- Terminal 실행하기

- Linux는 명령어를 사용하여 관리하기 때문에 Terminal을 사용한다
- Terminal을 실행하기 위해서 다음과 같이 한다



```
root@centos1:~  
File Edit View Search Terminal Help  
[root@centos1 ~]# hostname  
centos1  
[root@centos1 ~]# ping -c 1 www.google.com  
PING www.google.com (203.233.126.44) 56(84) bytes of data.  
64 bytes from 203.233.126.44: icmp_seq=1 ttl=53 time=5.10 ms  
  
--- www.google.com ping statistics ---  
1 packets transmitted, 1 received, 0% packet loss, time 0ms  
rtt min/avg/max/mdev = 5.106/5.106/5.106/0.000 ms  
[root@centos1 ~]# nmcli dev  
DEVICE      TYPE      STATE      CONNECTION  
virbr0      bridge    connected  virbr0  
eth0        ethernet  connected  eth0  
lo          loopback  unmanaged  --  
virbr0-nic  tap       unmanaged  --  
[root@centos1 ~]#
```

1 – Terminal 실행하기

- Terminal 전환하기

- 여러 개의 Terminal을 Windows로 실행 수 있다(Virtual Terminal)
- 여러 개의 Terminal을 사용하면 화면 단위로 다른 작업을 할 수 있을 뿐 아니라 각 Terminal별로 다른 사용자 계정으로 로그인해서 각각 다른 작업을 할 수 있는 장점이 있다
- **Centos**에서는 최대 5개의 Terminal을 실행할 수 있다
 - CTRL + ALT + **F2 ~ F6**
 - Terminal에서 GUI로 Console을 전환: CTRL + ALT + **F1**

```
CentOS Linux 7 (Core)
Kernel 3.10.0-229.20.1.el7.x86_64 on an x86_64

centos1 login: adminuser
Password:
Last login: Fri Dec 4 12:23:20 on :0
[adminuser@centos1 ~]$ tty
/dev/tty2
[adminuser@centos1 ~]$ _
```

```
CentOS Linux 7 (Core)
Kernel 3.10.0-229.20.1
```

```
centos1 login: root
Password:
Last login: Fri Dec 4 18:47:26 on :0
[root@centos1 ~]# tty
/dev/tty3
[root@centos1 ~]#
```

```
File Edit View Search Terminal Help
[root@centos1 ~]# tty
/dev/pts/1
[root@centos1 ~]# who
root      :0                2015-12-04 18:47 (:0)
root      pts/0                2015-12-04 18:35 (192.168.219.103)
root      pts/1                2015-12-04 21:41 (:0)
adminuser tty2                2015-12-04 21:42
root      tty3                2015-12-04 21:43
[root@centos1 ~]#
```

1 – Terminal 실행하기

- 어떻게 Linux 시스템에 접속했는지 확인하기
 - 실행중인 Process 확인하기
 - 현재 자신의 세션에서만 실행한 프로세스 확인
 - **ps**
 - 자신의 ID로 접속한 모든 터미널에서 실행 중인 모든 프로세스 확인
 - **ps a**
 - 현재 시스템에 로그인한 모든 사용자가 실행한 전체 프로세스 확인
 - **ps aux**
 - 접속하여 작업하고 있는 터미널 확인하기
 - 접속한 작업 중인 모든 사용자가 사용하고 있는 터미널 확인하기
 - **who**
 - 자신이 접속하여 작업하고 있는 터미널 확인하기
 - **who am i**

```
[root@centos1 ~]# who
root      :0                2015-12-04 18:47 (:0)
root      pts/0            2015-12-04 18:35 (192.168.219.103)
root      pts/1            2015-12-04 21:41 (:0)
adminuser tty2              2015-12-04 21:42
root      tty3              2015-12-04 21:43
root      pts/2            2015-12-04 22:11 (:0)
root      pts/3            2015-12-04 22:11 (:0)
[root@centos1 ~]# who am i
root      pts/0            2015-12-04 18:35 (192.168.219.103)
[root@centos1 ~]#
```

1 – Terminal 실행하기

- TTY 및 PTS란 무엇인가?
 - TTY(**TeleTY**pewriter)
 - Early user terminals connected to computers were electromechanical teleprinters or teletypewriters (TeleTYpewriter, TTY), and since then TTY has continued to be used as the name for **the text-only console**
 - Teletypewriter originally and now also means any terminal on Linux/Unix systems. It also means any serial port on Unix/Linux systems.
 - PTS (**P**seudo(사칭의) **T**erminal **S**lave)
 - The difference between TTY and PTS is **the type of connection to the computer**.
 - TTY ports are **direct connections to the computer** such as a **keyboard/mouse or a serial connection** to the device.
 - PTS connections are **SSH connections or telnet connections**.
 - All of these connections can connect to a shell which will allow you to issue commands to the computer.
 - Windows인 Virtual Terminal은 PTS에 포함된다

2 - Terminal에 나타나는 Prompt 기호 이해하기

\$ 및 #의 의미(意味)

~의 의미(意味)

adminuser@centos1:~\$의 의미(意味)

장착한 Disk와 Partition

Linux 기본 디렉터리 이해하기

2 - Terminal에 나타나는 Prompt 기호 이해하기

- 터미널의 프롬프트에서 **\$** 및 **#**의 의미(意味)
 - **\$**는 Bash를 이용하는 “일반 사용자”를 말한다
 - **#**은 “**root**” 사용자임을 뜻한다
 - “관리자 권한을 가진 사용자”란 **[sudo 명령어]**를 실행하여 관리 작업을 할 수 있는 사용자
 - wheel 그룹의 구성원은 sudo 명령어를 사용할 수 있다
 - **sudo yum** install tree
 - wheel 그룹의 구성원 확인하기
 - **cat /etc/group | grep wheel**
 - **id --group wheel**
- **~**의 의미(意味)
 - 명령을 실행하는 사용자의 홈 디렉토리를 나타내는 위치

```
[root@centos1 ~]# cat /etc/group | grep wheel  
wheel:x:10:adminuser
```

```
[root@centos1 ~]# pwd  
/root  
[root@centos1 ~]# su adminuser  
[adminuser@centos1 root]$ cd ~  
[adminuser@centos1 ~]$ pwd  
/home/adminuser
```

2 - Terminal에 나타나는 Prompt 기호 이해하기

- **[adminuser@centos1 ~]의 의미(意味)**

```
[adminuser@centos1 ~]$ whoami  
adminuser  
[adminuser@centos1 ~]$ hostname  
centos1  
[adminuser@centos1 ~]$ pwd  
/home/adminuser
```

- adminuser: 명령을 실행할 수 있는 사용자 이름
 - **whoami**
- @centos1: 현재 로그인한 시스템 이름이 centos1이다
 - **hostname**
- ~: 현재 디렉터리 위치가 명령을 실행할 수 있는 자의 홈 디렉터리이다
 - **pwd**
- ~ 대신 디렉터리 이름이 있는 경우는 현재 위치를 나타내는 것이다

```
[root@centos1 adminuser]# pwd  
/home/adminuser  
[root@centos1 adminuser]# cd /tmp  
[root@centos1 tmp]# pwd  
/tmp
```

- \$: 일반 사용자 권한으로 작업하고 있다
- #: root 사용자로 작업하고 있다

3 - 기본적인 명령어 사용법

명령 행 편집하기

Keyboard Shortcut Key 사용하기

명령어 구문 이해하기

리눅스 명령어의 특징

리눅스 기본 명령어 사용하기

Terminal 종료하기

시스템 종료하기

3 - 기본적인 명령어 사용법

- 명령 행 편집하기

- Terminal에 명령어를 입력하다가 잘못 입력했을 때 수정할 일이 생긴다.
이 때 다양한 키보드 키를 조합하여 원하는 작업을 할 수 있다
- 커저 앞 문자 지우기: backspace
- 커저가 있는 문자 지우기: Delete
- 커저 앞 단어 지우기: ctrl + w (w는 word)
- 커저 앞 행 모두 지우기: ctrl + u (u는 upper)

3 - 기본적인 명령어 사용법

• Keyboard Shortcut Key 사용하기

Ctrl + A	Go to <u>the beginning of the line</u> you are currently typing on
Ctrl + E	Go to <u>the end of the line</u> you are currently typing on
Ctrl + L	<u>Clears the Screen</u> , similar to the clear command
Ctrl + U	<u>Clears the line before the cursor position</u> . If you are at the end of the line, clears the entire line.
Ctrl + H	Same as backspace
Ctrl + R	Let's you search through previously used commands
Ctrl + C	<u>Kill whatever you are running</u>
Ctrl + D	<u>Exit the current shell</u>
Ctrl + Z	Puts whatever you are running into <u>a suspended background process</u> . <u>fg restores it</u> .
Ctrl + W	Delete the word before the cursor
Ctrl + K	<u>Clear the line after the cursor</u>
Ctrl + T	Swap the last two characters before the cursor
Esc + T	Swap the last two words before the cursor
Alt + F	Move cursor forward one word on the current line
Alt + B	Move cursor backward one word on the current line
Tab	<u>Auto-complete files and folder names</u>

3 - 기본적인 명령어 사용법

- Keyboard Shortcut Key 사용법

CTRL+B	Moves the cursor backward one character.
CTRL+C	<u>Cancels the currently running command.</u>
CTRL+D	<u>Logs out of the current session.</u>
CTRL+F	Moves the cursor forward one character.
CTRL+H	Erase one character. Similar to pressing backspace.
CTRL+P	<u>Paste previous line(s).</u>
CTRL+S	Stops all output on screen (XOFF).
CTRL+Q	Turns all output stopped on screen back on (XON).
CTRL+U	<u>Erases the complete line.</u>
CTRL+W	Deletes the last word typed. For example, if you typed 'mv file1 file2' this shortcut would delete file2.
CTRL+Z	Cancels current operation, moves back a directory or takes the current operation and moves it to the background. See bg command for additional information about background.

3 - 기본적인 명령어 사용법

- Command line shortcut

<code>~</code>	Moves to the user's home directory.
<code>!!</code>	<u>Repeats the line last entered at the shell.</u> See history command for previous commands.
<code>!\$</code>	<u>Repeats the last argument</u> for the command last used. See history command for previous commands.
<code>reset</code>	Resets the terminal <u>if terminal screen is not displaying correctly.</u>
<code>shutdown -h now</code>	Remotely or locally shuts the system down.

<!`$` 사용법>

`date`

`ifconfig`

`cd ~`

`!d`

`!i`

3 - 기본적인 명령어 사용법

- 명령어 구문 이해하기

명령의 구조

형식 명령 [옵션] [인자...]

명령 리눅스를 사용하기 위해 사용자가 입력하는 다양한 명령은 `date`, `man`, `ls`, `cp`, `mv` 등 수백 가지가 있다.

옵션 옵션을 사용하여 명령의 세부 기능을 선택할 수 있다. 리눅스의 기능을 풍부하게 하는 중요한 특징으로, 명령에 따라 있는 경우도 있고 없는 경우도 있다. 옵션은 `-` 기호로 시작하며 영문 소문자나 대문자로 구성된다. 명령에 따라 어떤 옵션이 있고 그 기능이 무엇인지는 해당 명령의 사용법을 참조해야 한다.

인자 인자는 명령으로 전달되는 값으로 주로 파일명이나 디렉터리명이 사용된다. 명령에 따라 없을 수도 있고 각 명령에 필요한 인자도 각기 다르므로 사용법을 참조해야 한다.

- 명령어만 사용한 경우

- **ls**

- 명령어와 옵션을 사용한 경우

- **ls -al**

- 명령어와 인자를 사용한 경우

- **ls /tmp**

- 명령어와 옵션 그리고 인자를 모두 사용한 경우

- **ls -al /tmp**

3 - 기본적인 명령어 사용법

- 리눅스 명령어의 **특징**

- 윈도우와는 다르게 대소문자를 구분한다
 - PasswD와 passwd를 다른 것이다
- Semicolon을 사용하여 한 라인에 여러 개의 명령어를 순차적으로 실행
 - **who ; date ; ifconfig**
 - 윈도우는 ; 대신 **&&**를 사용한다
hostname && ping 8.8.8.8 -n 2 && date

- 파일에 대한 사용 권한이 있다

- 리눅스는 여러 사용자가 동시에 접속해서 사용하므로 각 파일 및 디렉터리에는 사용 권한(Permission, ACL)이 설정되어 있어서 권한이 있는 사람만이 파일이나 디렉터리에 접근할 수 있다
 - **ls -la jesuswithme**
-rw-rw-r-- 1 adminuser adminuser 0 10월 9 23:27 jesuswithme

3 - 기본적인 명령어 사용법

- 리눅스 명령어의 **특징**

- 명령어 및 파일에 확장자가 없다

- 파일 및 명령어에 확장자가 없기 때문에 확장자를 보고서 파일의 종류를 알 수 없다

cp (copy.exe가 아니다)

mkdir (mkdir.exe가 아니다)

- 파일의 종류를 알기 위해서는 **[file 파일명]**을 사용한다

adminuser@centos1:~\$ **file** **jesuswithme**

jesuswithme: empty

adminuser@centos1:~\$ **file** **peace**

peace: ASCII text

adminuser@centos1:~\$ **file** **/bin/cp**

/bin/cp: ELF 64-bit LSB executable

3 - 기본적인 명령어 사용법

- 리눅스 기본 명령어 **사용하기**

- date:** 현재 시스템의 날짜와 시간을 확인하고 설정한다

- 날짜에 대한 도움말 확인하기

- date --help**

- ISO-8601의 날짜 표기 방식으로 출력(예: 2015-06-02)

- date -l** (대문자 아이)

- 세계 표준 시각 확인

- date -u** (Universal Time Coordinated)

- root 사용자가 시스템 시간을 변경하기

- date MMDDhhmmYY (월/일/시/분/년)

- date 060210002016**

3 - 기본적인 명령어 사용법

- 리눅스 기본 명령어 **사용하기**

- **who:** 현재 시스템에 누가 접속하고 있는지 확인하기
 - **whoami**
 - **who am i**
 - **who**
 - **who --help**
 - 로그인 한 사용자의 모든 정보를 확인한다
 - **who -aH**
 - 출력되는 정보의 Header(속성 이름)을 보여줄 때 -H를 사용한다
 - **who -H**
 - 접속한 사용자의 DNS hostname을 알려준다
 - **who -l** (소문자 L)
 - 터미널로 접속한 사용자 계정 모두와 수량을 보여준다
 - **who -q**

3 - 기본적인 명령어 사용법

- 리눅스 기본 명령어 **사용하기**

- **clear**: 현재 사용 중인 터미널의 내용을 지운다

- **clear**
- **Ctrl + I** (##소문자 l)

- **passwd**: 로그인한 사용자가 자신의 계정에 대한 암호만 재설정한다.
또는 관리자 권한을 가진 사용자가 다른 사람 계정에 대하여 암호를 재설정할 수 있다

- **passwd** (본인 것만 변경; 반드시 현재 사용중인 암호를 알고 있어야 한다)
- **passwd adminuser** (root 사용자가 adminuser의 암호를 변경한다)

3 - 기본적인 명령어 사용법

- 리눅스 기본 명령어 **사용하기**

- **man:** 명령어의 내용을 세세하게 기술하고 있는 설명서를 보는 것이다

- **man clear**
- **man who**
- **man who | grep \\-a**

- **--help:** 각 명령어의 도움말을 보는 것이다. man보다 양이 적지만 자주 활용한다

- **clear --help**
- **who --help**

- **Terminal 종료하기**

- 윈도우에서는 X표시를 클릭한다
- **Ctrl + d**
- **exit**

4 – RunLevel 이해 및 System Shutdown 하기

- RunLevel 이해하기

- centos7부터 runlevel 설정을 init 프로세스가 담당하지 않고 systemd가 담당하는 것으로 변경되었다
- runlevel이란 시스템을 어떤 수준으로 운영할 것인지를 결정하는 것으로서 다양한 runlevel이 있다
- runlevel 종류 및 목록 확인하기
 - **ls -l /lib/systemd/system | grep runlevel**

```
[root@centos1 system]# pwd
/lib/systemd/system
[root@centos1 system]# ls -l | grep runlevel
lrwxrwxrwx. 1 root root 15 Dec 3 16:38 runlevel0.target -> poweroff.target
lrwxrwxrwx. 1 root root 13 Dec 3 16:38 runlevel1.target -> rescue.target
drwxr-xr-x. 2 root root 49 Dec 3 16:38 runlevel1.target.wants
lrwxrwxrwx. 1 root root 17 Dec 3 16:38 runlevel2.target -> multi-user.target
drwxr-xr-x. 2 root root 49 Dec 3 16:38 runlevel2.target.wants
lrwxrwxrwx. 1 root root 17 Dec 3 16:38 runlevel3.target -> multi-user.target
drwxr-xr-x. 2 root root 49 Dec 3 16:38 runlevel3.target.wants
lrwxrwxrwx. 1 root root 17 Dec 3 16:38 runlevel4.target -> multi-user.target
drwxr-xr-x. 2 root root 49 Dec 3 16:38 runlevel4.target.wants
lrwxrwxrwx. 1 root root 16 Dec 3 16:38 runlevel5.target -> graphical.target
drwxr-xr-x. 2 root root 49 Dec 3 16:38 runlevel5.target.wants
lrwxrwxrwx. 1 root root 13 Dec 3 16:38 runlevel6.target -> reboot.target
-rw-r--r--. 1 root root 761 Sep 15 22:21 systemd-update-utmp-runlevel.service
```

4 – RunLevel 이해 및 System Shutdown 하기

- RunLevel 이해하기

- 현재 runlevel 상태 확인하기

- **who -r**

```
[root@centos1 ~]# who -r
      run-level 5    2015-12-05 10:21          last=3
[root@centos1 ~]#
```

- default runlevel 확인 및 설정하기

- default runlevel이란 시스템이 시작할 때 자동으로 설정되는 runlevel 값

- default runlevel **확인하기**

- **systemctl get-default**

- default runlevel **변경하기**

- **systemctl set-default multi-user.target**

- shutdown -r now

- systemctl get-default

```
$ ssh root@192.168.219.104
root@192.168.219.104's password:
Last login: Sat Dec  5 08:15:27 2015
[root@centos1 ~]# systemctl get-default
multi-user.target
[root@centos1 ~]#
```


4 – RunLevel 이해 및 System Shutdown 하기

- RunLevel 이해하기
 - systemd와 runlevel과의 관계

런레벨	target 파일(심벌릭 링크)	target 원본 파일
0	runlevel0.target	poweroff.target
1	runlevel1.target	rescue.target
2	runlevel2.target	multi-user.target
3	runlevel3.target	
4	runlevel4.target	
5	runlevel5.target	graphical.target
6	runlevel6.target	reboot.target

4 – RunLevel 이해 및 System Shutdown 하기

- RunLevel 이해하기

- 현재 runlevel 수정하기

- 이 작업은 시스템을 재부팅하지 않은 상태에서 runlevel만 변경하는 것이다
 - single-user mode runlevel (runlevel 1): root로만 접속 가능/문제 해결용
 - **systemctl isolate rescue.target** (## 또는 systemctl isolate **runlevel1.target**)
 - who -r
 - multi-user mode runlevel (runlevel 3): 다중 사용자 접속 허용/text mode 환경
 - **systemctl isolate multi-user.target** (## 또는 systemctl isolate **runlevel3.target**)
 - who -r
 - graphical mode runlevel (runlevel 5): 다중 사용자 접속 허용/graphic 환경
 - **systemctl isolate graphical.target** (## 또는 systemctl isolate **runlevel5.target**)
 - who -r

- 현재 runlevel 수정하기-이전 방식 그대로 사용 가능(**telinit #**)

- **telinit 1**
 - **telinit 3**
 - **telinit 5**

4 – RunLevel 이해 및 System Shutdown 하기

- System shutdown하기

- Shutdown 명령어를 실행하면 다음과 같은 단계를 진행한다

- ① Shutdown 하기 전에 sync 작업을 한다
- ② 접속해 있는 사용자들에게 시스템이 종료된다는 메시지를 전달한다
- ③ 새로운 사용자의 로그인을 금지한다
- ④ 지정된 시간 내에 종료되지 않은 프로세스를 강제 종료한다
- ⑤ 지정된 시간 내에 로그아웃하지 않은 사용자를 강제 로그아웃시킨다
- ⑥ 메모리에 남아 있는 데이터를 디스크에 저장한다
- ⑦ 시스템 종료와 관련된 정보를 시스템 로그 파일에 기록한다
- ⑧ 마운트 되어 있는 장치들을 언마운트(탈착)한다
- ⑨ 시스템을 종료한다

- Shutdown 명령어는 root만 실행할 수 있다

- 명령어 **visudo**를 실행하여 adminuser에게 root와 같은 계정을 부여하면, adminuser도 **sudo shutdown -h now**를 실행할 수 있다

4 – Run Level 이해 및 System Shutdown 하기

- System 종료하기

- Shutdown 명령어 사용법: shutdown [-krhfc][-t sec] time [경고메시지]

shutdown

기능 리눅스를 종료한다.

형식 shutdown [옵션] [시간] [메시지]

옵션 -k : 실제로 시스템을 종료하는 것이 아니라 사용자들에게 메시지만 전달한다.

-r : 종료 후 재시작한다.

-h : 종료하며 halt 상태로 이동한다.

-f : 빠른 재시작으로 이 과정에서 fsck를 생략할 수도 있다.

-c : 이전에 내렸던 shutdown 명령을 취소한다.

시간 : 종료할 시간(hh:mm, +m, now)

메시지 : 모든 사용자에게 보낼 메시지

사용 예 shutdown -h now shutdown -r +3 "System is going down" shutdown -c

- 지금 당장 시스템 종료하기

shutdown -h now

- 지금 시스템 재부팅하기

shutdown -r now

- 10분 후에 종료하기

shutdown -h +10

4 – Run Level 이해 및 System Shutdown 하기

- System 종료하기

- Shutdown 명령어 사용법: shutdown [-krhfc][-t sec] time [경고메시지]

- 3분 후에 메시지를 전달하고 종료하기

shutdown +3 “시스템을 3분 후에 종료합니다”

- 시스템을 종료한다는 가짜 메시지만을 보낸다

shutdown -k “시스템을 3분 후에 종료합니다”

- Single user mode로 바꾸기

shutdown now

```
[root@centos adminuser]# shutdown now
```

```
Broadcast message from adminuser@centos  
(/dev/pts/0) at 18:16 ...
```

```
The system is going down for maintenance NOW!
```

```
[root@centos adminuser]# █
```

```
Telling INIT to go to single user mode.  
init: rc main process (1915) killed by TERM signal  
[root@centos /]# runlevel  
1 S  
[root@centos /]# _
```

- **GUI**에서 shutdown now를 하면 “The system is going down for maintenance NOW!”라는 메시지를 던지고 아무 것도 할 수 없다
 - Console로 작업하다가 shutdown now를 하면 시스템이 재부팅되면서 runlevel 1로 시작함을 알 수 있다. 즉, root 만 접속할 수 있는 single user mode로서 시스템을 유지보수만 할 수 있는 환경이다

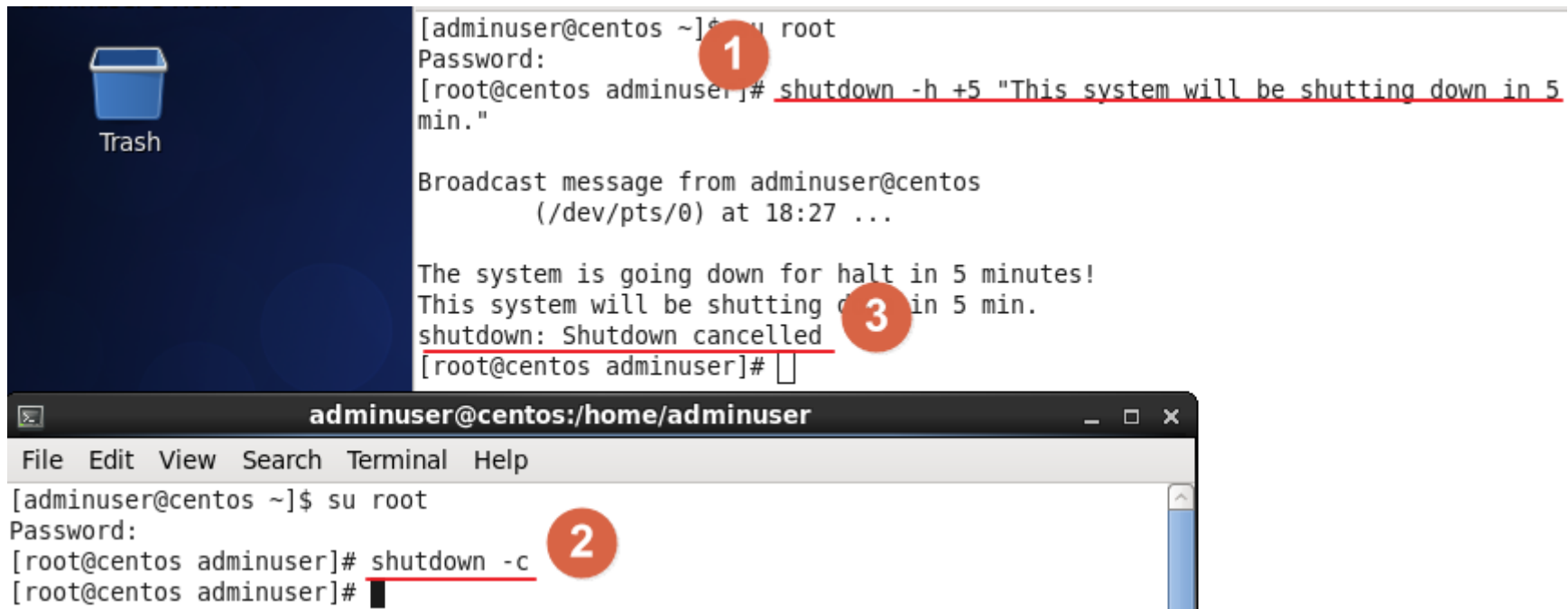
4 – Run Level 이해 및 System Shutdown 하기

- System 종료하기

- Shutdown 명령어 사용법: shutdown [-krhfc][-t sec] time [경고메시지]

- 시스템 종료 명령어 취소하기

shutdown -c



```
[adminuser@centos ~]$ su root
Password:
[root@centos adminuser]# shutdown -h +5 "This system will be shutting down in 5 min."

Broadcast message from adminuser@centos
(/dev/pts/0) at 18:27 ...

The system is going down for halt in 5 minutes!
This system will be shutting down in 5 min.
shutdown: Shutdown cancelled
[root@centos adminuser]#
```

adminuser@centos:/home/adminuser

File Edit View Search Terminal Help

```
[adminuser@centos ~]$ su root
Password:
[root@centos adminuser]# shutdown -c
[root@centos adminuser]#
```

4 – Run Level 이해 및 System Shutdown 하기

- System shutdown하는 명령어
 - **shutdown**
 - 가장 다양하게 시스템을 끄거나 재시작한다
 - Runlevel에는 전혀 영향을 끼치지 않는다
 - shutdown -r now로 재시작할 때는 시스템의 default runlevel로 시작한다
 - **telinit**
 - 시스템 종료: **telinit 0**
 - 시스템 재부팅: **telinit 6**
 - 시스템 종료
 - **halt**
 - **poweroff**
 - 시스템 재시작
 - **reboot**

5 – 리눅스 시스템의 Booting 과정

리눅스 시스템 Booting 개요

systemd 이해하기

리눅스 시스템 Booting 개요

리눅스 시스템 Booting이란?

리눅스 부팅 과정

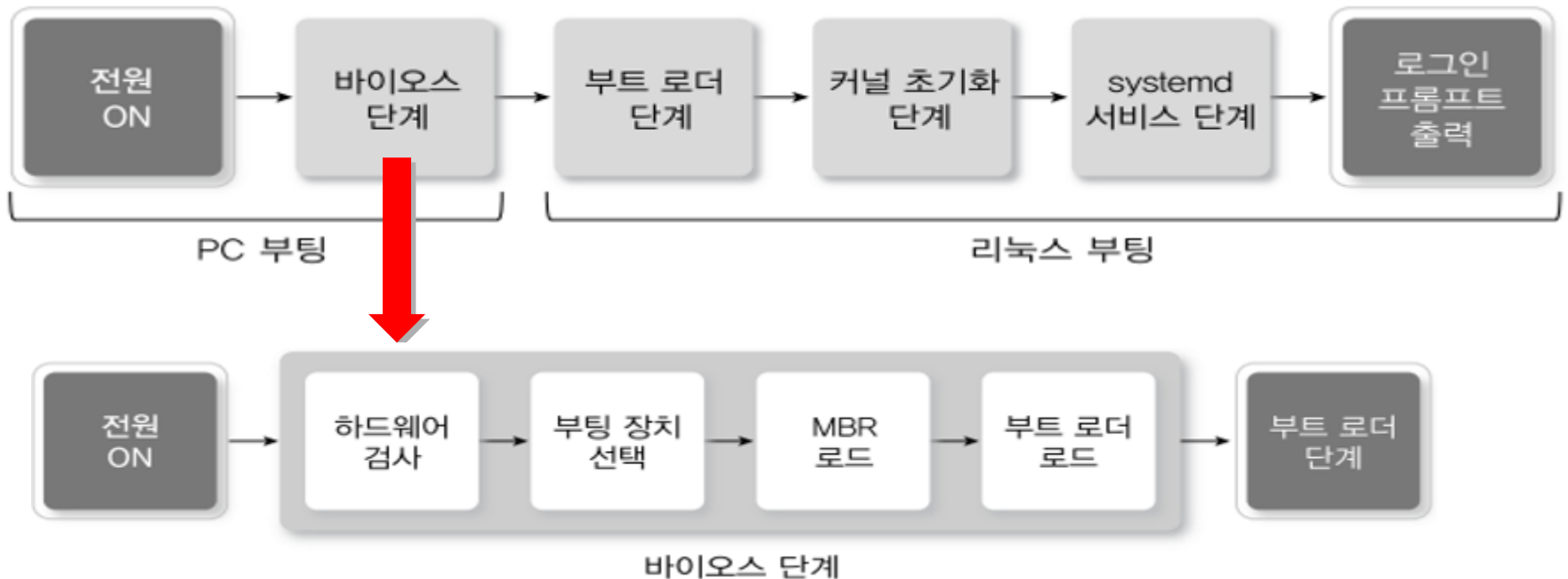
리눅스 시스템 Booting 개요

- 리눅스 시스템의 부팅이란?

- 컴퓨터에 전원을 공급하면 먼저 Mainboard의 BIOS를 검사한 후 Boot Loader 및 Kernel을 로딩하고 각종 서비스(데몬)를 시작하고 마지막 단계에 Login 프롬프트를 화면에 출력하게 된다
- 사용자는 콘솔 및 네트워크로 로그인을 하여 다양한 작업을 할 수 있다
- 부팅할 때 가장 먼저 실행되는 프로세스가 init인데, 이것은 다른 프로세스를 동작시키는 스크립트를 실행한다. 그래서 init 프로세스를 부모 프로세스라고 한다
- 리눅스의 대표적인 Boot Loader는 GRUB인데, root 계정에 대한 암호를 잃어버렸을 때는 [단일 사용자 모드]로 부팅하여 해결할 수 있다

리눅스 시스템 Booting 개요

- 리눅스 부팅 과정



1. BIOS 단계

- BIOS에서 CPU, RAM 등이 있는지 검사한 후 부팅 디스크를 찾아서 그것의 첫 번째 섹터에서 512byte(MBR)를 로딩하여 디스크의 어느 Partition에 2차 부팅 프로그램(Boot Loader)이 있는지 찾아서 Boot Loader를 메모리로 로딩한다

리눅스 시스템 Booting 개요

- 리눅스 부팅 과정

- 2. Boot Loader 단계

- BIOS 단계에서 MBR은 Boot Loader를 찾아서 메모리에 로딩한다
 - Boot Loader는 일반적으로 여러 OS 중에서 부팅할 OS를 선택할 수 있는 메뉴를 제공하는 것이다
 - 리눅스의 대표적인 Boot Loader는 GRUB와 LILO이다
 - 기본적으로 GRUB를 Boot Loader로 사용하는데, 하나의 운영체제만 설치되어 있으면 선택 메뉴가 없이 곧장 부팅 작업을 진행한다
 - Boot Loader는 **Linux Kernel**을 메모리로 로딩하는 역할을 한다
 - 설치된 리눅스 커널 버전을 확인하기 위해서는
 - **ls /boot/vm***
 - **uname -r**

```
[root@centos1 ~]# uname -r
3.10.0-229.20.1.el7.x86_64
[root@centos1 ~]# ls /boot/vm*
/boot/vmlinuz-0-rescue-93949fbd5064487bb7
/boot/vmlinuz-3.10.0-229.20.1.el7.x86_64
/boot/vmlinuz-3.10.0-229.el7.x86_64
[root@centos1 ~]#
```

리눅스 시스템 Booting 개요

• 리눅스 부팅 과정

3. 커널 초기화 단계

- Boot Loader에 의해 메모리에 로딩된 커널은 가장 먼저 리눅스 시스템에 연결된 RAM, Disk, Keyboard, Mouse 등의 장치들을 검사한다
- 리눅스를 처음 시스템에 설치할 때, 사용 가능한 Hardware 정보를 미리 확인했다가, 나중에 부팅할 때 이 장치들이 사용 가능한 상태로 유지되고 있는지 확인하는 것이다.
- 장치 검사 등 기본적인 초기화 과정이 끝나면 커널은 일반적으로 생성하는 방식인 fork를 사용하지 않고 생성되는 프로세스와 스레드를 생성한다
- 이 프로세스들은 메모리 관리와 같은 커널의 여러 가지 동작을 수행한다
- 이러한 프로세스들은 일반적인 프로세스와 구분하기 위해서 []로 표시하며, PID가 낮게 할당되어 있다(**kernel process**라고 함)

ps -ef | more

```
[root@centos1 ~]# ps -ef | more
UID          PID    PPID  C STIME TTY          TIME CMD
root           1        0  0  10:55 ?           00:00:01 /usr/lib/systemd/systemd
system --deserialize 21
root           2        0  0  10:55 ?           00:00:00 [kthreadd]
root           3        2  0  10:55 ?           00:00:00 [ksoftirqd/0]
root           5        2  0  10:55 ?           00:00:00 [kworker/0:0H]
root           6        2  0  10:55 ?           00:00:00 [kworker/u128:0]
root           7        2  0  10:55 ?           00:00:00 [migration/0]
root           8        2  0  10:55 ?           00:00:00 [rcu_bh]
```

리눅스 시스템 Booting 개요

- 리눅스 부팅 과정

- 4. systemd 서비스 단계

- kernel process들이 생성된 후에는 systemd 서비스 단계로 넘어간다
 - systemd 서비스 단계에 이르면 **리눅스가 본격적으로 동작하기** 시작한다
 - centos7에서의 systemd 서비스는 기존의 cenos7 이전 버전의 init script를 대체한 것으로 다양한 서비스를 동작시킨다
 - 각 서비스가 시작할 때 그 과정을 화면에 메시지로 출력하게 된다. 하지만 boot splash 이미지 때문에 과정이 안보인다. **[ALT+D]**를 눌러서 메시지를 보면 된다
 - 로그인한 후 다음 명령어를 실행하면 부팅할 때 실행된 메시지를 볼 수 있다
 - **more /var/log/boot.log**
 - **dmesg** (## 데몬의 시작과 관련된 것과 하드웨어 검사와 관련된 것)
 - 리눅스 커널이 제일 먼저 실행시킨 프로세스가 systemd다.
 - 따라서 systemd는 모든 프로세스의 [최상위 부모 프로세스]로서 PID가 1번이다.

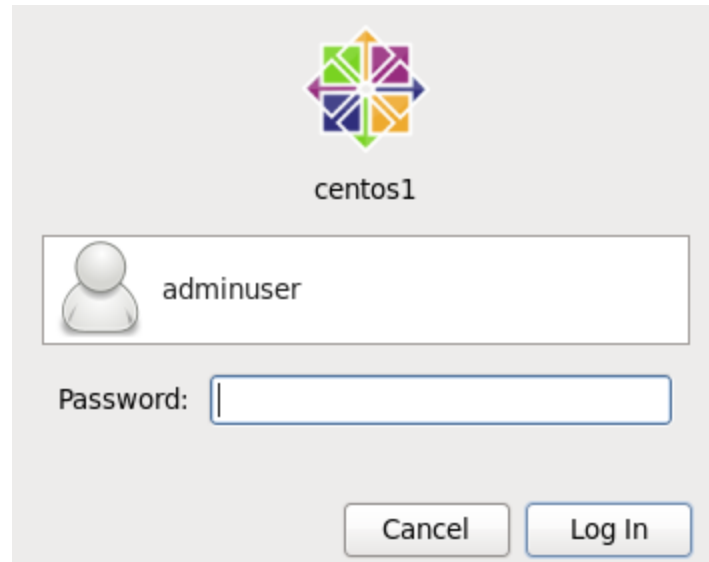
```
[root@centos1 ~]# pstree
systemd─ModemManager─2*[{ModemManager}]
        │
        └─NetworkManager─dhclient
                        └─3*[{NetworkManager}]
        └─2*[abrt-watch-log]
```

리눅스 시스템 Booting 개요

- 리눅스 부팅 과정

- 5. Login Prompt 출력 단계

- systemd 실행 단계에서 데몬을 모두 실행시킨 뒤 마지막으로 그래픽 로그인 시스템을 동작시킨다
 - 이것으로서 리눅스 부팅 과정이 완료되었다
 - 사용자는 사용자 계정과 암호를 입력하여 로그인하면 된다



리눅스 시스템 Booting 개요

- Single User mode(단일 사용자 모드)로 부팅하기
 - Root 암호를 잃어버렸을 때 Single User mode로 접속하여 다른 사용자들의 접속을 제한한 후 root의 암호를 복구할 수 있다
 - Root 계정의 암호를 복구하는 방법
 - 리눅스 시스템을 재시작할 때 GRUB Boot Menu가 출력될 때 신속하게 **e**를 눌러서 **GRUB 편집모드**로 진입해야 한다

```
CentOS Linux (3.10.0-229.20.1.el7.x86_64) 7 (Core)
CentOS Linux (3.10.0-229.20.1.el7.x86_64) 7 (Core) with debugging
CentOS Linux 7 (Core), with Linux 3.10.0-229.el7.x86_64
CentOS Linux 7 (Core), with Linux 0-rescue-93949fbd5064487bb73e2a24758f3-
```

- 아래의 **rhgb quiet**를 찾아서 **init=/bin/bash**로 변경한다

```
linux16 /vmlinuz-3.10.0-229.20.1.el7.x86_64 root=/dev/mapper/centos-ro\
ot ro rd.lvm.lv=centos/root rd.lvm.lv=centos/swap crashkernel=auto rhgb quiet \
LANG=en_US.UTF-8 systemd.debug
initrd16 /initramfs-3.10.0-229.20.1.el7.x86_64.img
```

- 그 다음 **ctrl+x**를 눌러서 bash로 부팅한다-**Single User mode 상태**

```
[ OK ] Reached target Initrd Default Target.
[ 8.420374] systemd-journald[211]: Received SIGTERM
bash-4.2#
```


리눅스 시스템 Booting 개요

- Single User mode(단일 사용자 모드)로 부팅하기

- Root 계정의 암호를 복구하는 방법(계속)

- root 파티션 상태 검사해 보면 read only permission으로 되어 있다

- **mount | grep root**

```
bash-4.2# mount | grep root
/dev/mapper/centos-root on / type xfs (ro,relatime,attr2,inode64,noquota)
bash-4.2#
```

- root 파티션을 read only에서 **read/write**로 변경한 후 root 계정의 암호를 reset한다

- **mount -o remount,rw /**

- **mount | grep root**

- **passwd root**

새로운 암호 2번 입력하기

```
bash-4.2# mount | grep root
/dev/mapper/centos-root on / type xfs (rw,relatime,attr2,inode64,noquota)
bash-4.2#
```

- SELinux context를 relabel하여 암호를 사용하여 로그인하도록 설정하기

- **touch /.autorelabel**

- 다음 명령어를 입력하여 변경한 root 암호를 가지고 로그인 하기

- **exec /sbin/init**

systemd 이해하기

systemd의 특징

systemd 명령어 사용하기

systemd 이해하기

- systemd의 특징

- systemd는 Unix계열 운영체제의 PID 1이었던 init(System V Init)을 교체한 것이며 초기화 스크립트 관리자이고 로그시스템 관리자이다
- **시스템 부팅 속도를 획기적으로 개선하기** 위해 부팅시에 띄워야 하는 서비스 수를 최대한 줄였다
 - bluetoothd 데몬은 Bluetooth dongle 장치가 접속되지 않은 때에는 꼭 실행될 필요가 없기 때문에 bluetoothd는 부팅할 때는 실행하지 않게 한다
 - **systemd-analyze** (##부팅 시간 확인하기)
 - **systemd-analyze blame** (##부팅시 각 서비스별로 걸린 시간 확인하기)
- **서비스 간의 의존성을 없애서** 서비스들을 병렬로 실행하도록 한다
 - 보통 서비스들은 의존하고 있다. 즉, D-Bus가 실행되려면 로그시스템인 syslog가 먼저 준비되어 있어야 한다
 - serviceA의 선행 서비스가 serviceB인 경우, serviceA는 serviceB 전체가 필요한 것이 아니라 serviceB의 socket만 필요하다
 - systemd는 이러한 점을 이용하여 해당 소켓을 먼저 생성해 두고, 경우에 따라 들어오는 시그널은 소켓 버퍼에 쌓아두었다가 해당 서비스가 준비 완료되면 그것을 처리하게 해주는 방식으로 의존성을 낮추었다

systemd 이해하기

- systemd의 특징

- systemd는 전체 시스템을 시작하고 관리하는데 **units**를 사용한다
 - systemd는 관리 대상의 이름을 [서비스이름.유닛종류] 형태로 관리한다
- systemd의 유닛 종류

유닛의 종류	기능	예
service	가장 명백한 유닛으로 데몬을 시작, 종료, 재시작, 로딩한다.	atd.service
socket	소켓을 관리하는 유닛으로 AF_INET, AF_INET6, AF_UNIX 소켓 스트림과 데이터그램, FIFO를 지원한다.	dbus.socket
device	리눅스 장치 트리에 있는 장치를 관리한다.	dev-sda.device
mount	디렉터리 계층 구조의 마운트 포인트를 관리한다.	boot.mount
automount	디렉터리 계층 구조에서 자동 마운트 포인트를 관리한다.	proc-sys-fs-binfmt_misc.automount
target	유닛들을 그룹핑한다(예 : multi-user.target → 런레벨 5에 해당하는 유닛).	default.target runlevel0.target
snapshot	다른 유닛을 참조하기 위한 유닛이다.	foo.snapshot
swap	스왑 장치를 관리한다.	foo.swap
path	경로를 관리한다.	cups.path
timer	타이머와 관련된 기능을 관리한다.	systemd-readahead-done.timer

systemd 이해하기

- The architecture of systemd

systemd Utilities

systemctl journalctl notify analyze cglsg cgtop loginctl nspawn

systemd Daemons

systemd
journald networkd
logind user session

systemd Targets

bootmode basic multi-user graphical user-session
shutdown reboot dbus telephony user-session display service
dlog logind tizen service

systemd Core

manager unit login namespace log
systemd service timer mount target multiseat inhibit
snapshot path socket swap session pam cgroup dbus

systemd Libraries

dbus-1 libpam libcap libcryptsetup tcpwrapper libaudit libnotify

Linux Kernel

cgroups autofs kdbus

systemd 이해하기

- systemd 명령어 사용하기
 - systemctl 구문

systemctl

기능 systemd를 제어한다.

형식 systemctl [옵션] [명령] [유닛 이름]

옵션 -a : 상태와 관계없이 유닛 전체를 출력한다.
-t 유닛 종류 : 지정한 종류의 유닛만 출력한다.

명령 start : 유닛을 시작한다.
stop : 유닛을 정지한다.
reload : 유닛의 설정 파일을 다시 읽어온다.
restart : 유닛을 재시작한다.
status : 유닛의 상태를 출력한다.
enable : 부팅 시 유닛이 시작되도록 설정한다.
disable : 부팅 시 유닛이 시작되지 않도록 설정한다.
is-active : 유닛이 동작하고 있는지 확인한다.
is-enabled : 유닛이 시작되었는지 확인한다.
isolate : 지정한 유닛 및 이와 관련된 유닛만 시작하고 나머지는 정지한다.
kill : 유닛에 시그널을 전송한다.

사용 예 systemctl systemctl -a systemctl start atd.service

systemd 이해하기

- systemd 명령어 사용하기
 - 현재 동작중(active)이거나 오류(failed) 난 유닛 확인하기
 - **systemctl**
 - 현재 동작중(active)이거나 동작 중이지 않는 모든 유닛 확인하기
 - **systemctl -a**
 - 시스템 시작시 오류 난 것만 확인하기
 - **systemctl --failed**
 - 특정한 유닛(units)만 확인하기
 - **systemctl -t service**
 - **systemctl -t device**
 - **systemctl -t socket**
 - **systemctl -t mount**
 - **systemctl -t target**

systemd 이해하기

- systemd 명령어 사용하기
 - 시스템이 시작할 때 각 유닛이 어떤 상태(state)에 있는지 확인하기
 - **systemctl list-unit-files**
 - 특정한 유닛인 service의 상태 확인
 - **systemctl list-unit-files | grep service**

systemd 이해하기

- systemd 명령어 사용하기
 - Service 관리하기
 - 특정한 서비스의 실행 여부 확인하기
 - **systemctl is-active crond**
 - **systemctl status crond**
 - 실행중인 서비스 중지하기
 - **systemctl stop crond**
 - 중지된 서비스 시작하기
 - **systemctl start crond**
 - 실행중인 서비스의 설정 변경 후 다시 적용하기(서비스 재시작하기)
 - **systemctl restart crond**
 - 부팅시 서비스 자동 시작 여부 확인
 - **systemctl is-enabled crond**
 - 서비스 자동 시작하도록 서비스 활성화
 - **systemctl enable crond**
 - 서비스 자동 시작 못하도록 서비스 비활성화
 - **systemctl disable crond**

systemd 이해하기

- systemd 명령어 사용하기

- Log 관리하기

- 로그는 **systemd-journald**를 통해서 관리된다
 - 로그 관리를 위해서 **journalctl** 명령어를 사용한다
 - 단순히 전체 이벤트를 로그를 확인하기
 - **journalctl**
 - 프로세스로 실행이 가능한 특정 바이너리에 대한 이벤트 확인하기
 - **journalctl /sbin/crond**
 - 특정 날짜부터 지금까지 기록된 이벤트 로그를 확인하기
 - **journalctl --since=today**
 - today, yesterday, tomorrow 같은 단어도 가능하다.
 - "YYYY-MM-DD HH:MM:SS" 형태의 시간 값을 이용해서 구간 별 조회가 가능
 - 특정 구간의 이벤트 로그 확인하기
 - **journalctl --since=2015-10-21 --until=2015-10-30**
 - 부팅 이후의 모든 이벤트 로그 확인하기
 - **journalctl -b**

systemd 이해하기

- systemd 명령어 사용하기
 - Log 관리하기(계속)
 - syslog에서 지정하는 debug, info, err와 같은 속성별로 조회 가능
 - **journalctl -p err**
 - 실시간으로 이벤트 로그를 콘솔 화면에 띄우기
 - **journalctl -f**
 - **tail -f /var/log/messages**와 같은 기능
 - 이벤트 로그 포맷 변경하기
 - **journalctl -p err -o json-pretty**