

IT CookBook CentOS 리눅스

시스템 & 네트워크

[강의교안 이용 안내]

- 본 강의교안의 저작권은 한빛아카데미(주)에 있습니다.
- 이 자료를 무단으로 전제하거나 배포할 경우 저작권법 136조에 의거하여 최고 5년 이하의 징역 또는 5천만원 이하의 벌금에 처할 수 있고 이를 병과(併科)할 수도 있습니다.



CentOS 리눅스

시스템 & 네트워크

Chapter 08. 리눅스의 부팅과 종료

목차

00. 개요

01. 리눅스 시스템의 부팅

02. systemd 서비스

03. 리눅스 시스템의 종료

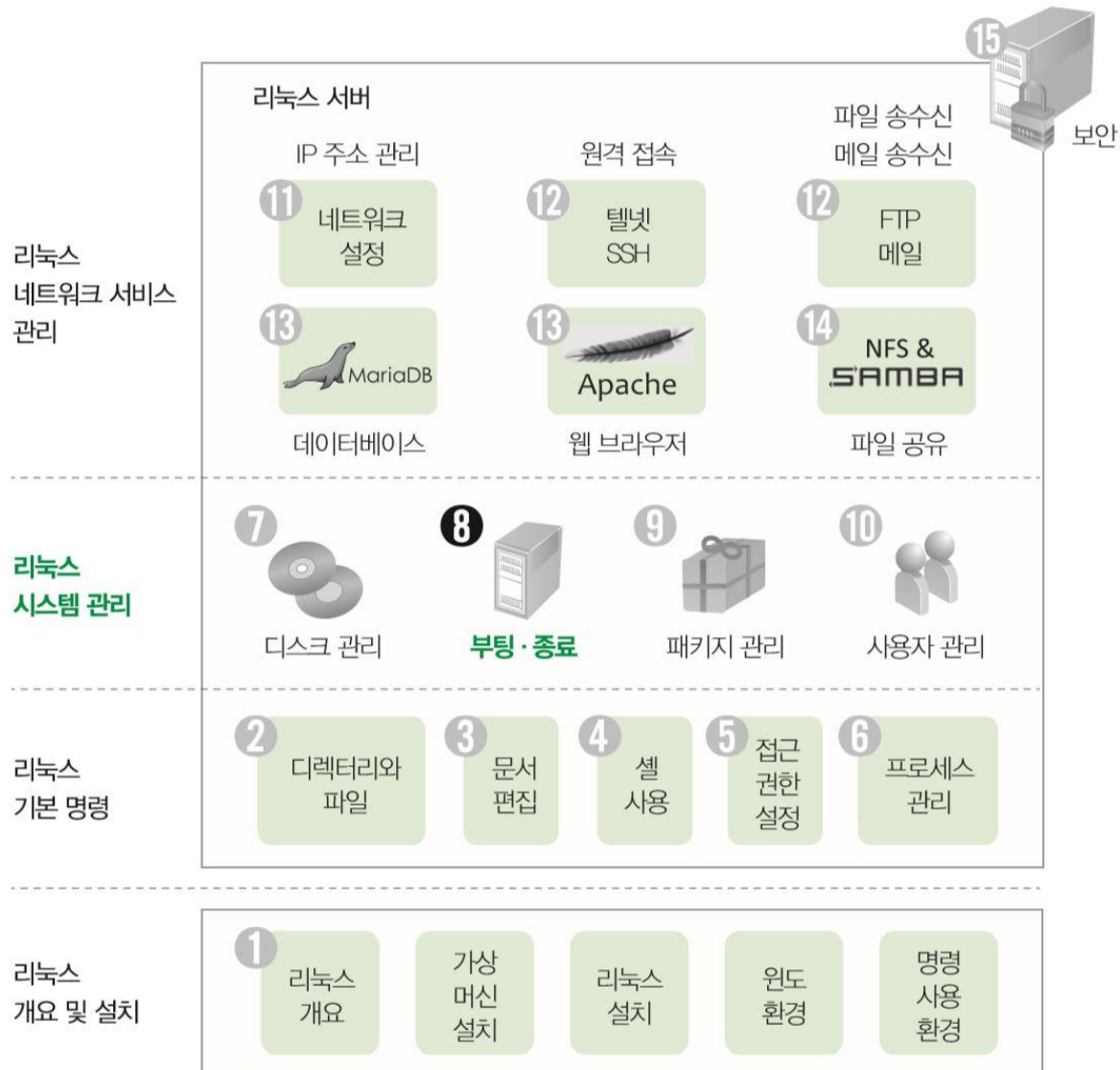
04. 데몬 프로세스

05. 부트 로더

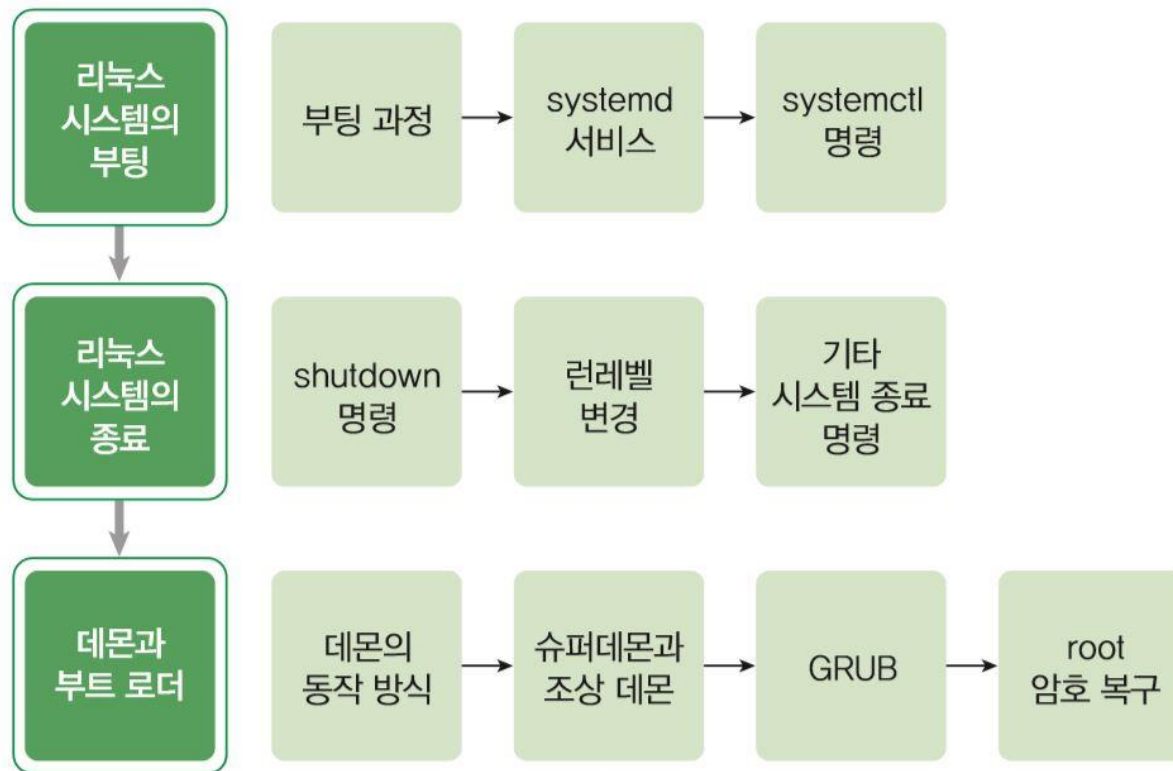
학습목표

- 리눅스 시스템의 부팅 과정을 이해하고 부트 로더의 역할을 설명할 수 있다.
- systemd의 기능을 이해하고 사용법을 설명할 수 있다.
- 런레벨을 이해하고 변경할 수 있다.
- 리눅스 시스템 종료 방법을 설명할 수 있다.
- 데몬을 이해하고 슈퍼데몬의 역할을 설명할 수 있다.
- root 계정의 암호를 복구할 수 있다.

00.개요



00.개요



01.리눅스 시스템의 부팅

■ 리눅스의 부팅

■ 리눅스의 부팅

- PC의 전원을 켜는 순간부터 리눅스가 완전히 동작하여 로그인 프롬프트가 출력될 때까지를 말한
- 크게 PC 부팅과 리눅스 부팅으로 나뉨

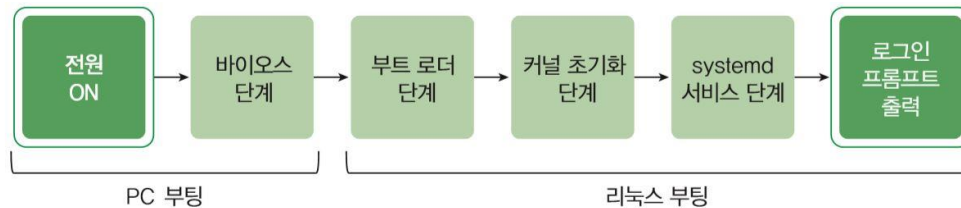


그림 8-1 리눅스의 부팅 과정

■ 바이오스 단계

■ 바이오스(BIOS)

- PC의 전원 스위치를 켜면 제일 먼저 동작
- 보통 ROM에 저장되어 있어 흔히 ROM-BIOS라고 부름
- PC에 장착된 기본적인 하드웨어(키보드, 디스크 등)의 상태를 확인한 후 부팅 장치를 선택하여 부팅 디스크의 첫 섹터에서 512바이트(마스터 부트 레코드, MBR)를 로딩
- 메모리에 로딩된 MBR은 부트 로더를 찾아 메모리에 로딩하는 작업까지 수행



바이오스 단계

그림 8-2 바이오스 단계의 세부 동작

01.리눅스 시스템의 부팅

■ 부트 로더 단계

- 바이오스 단계에서 MBR는 부트 로더를 찾아 메모리에 로딩
- 부트 로더는 여러 운영체제 중에서 부팅할 운영체제를 선택할 수 있도록 메뉴를 제공

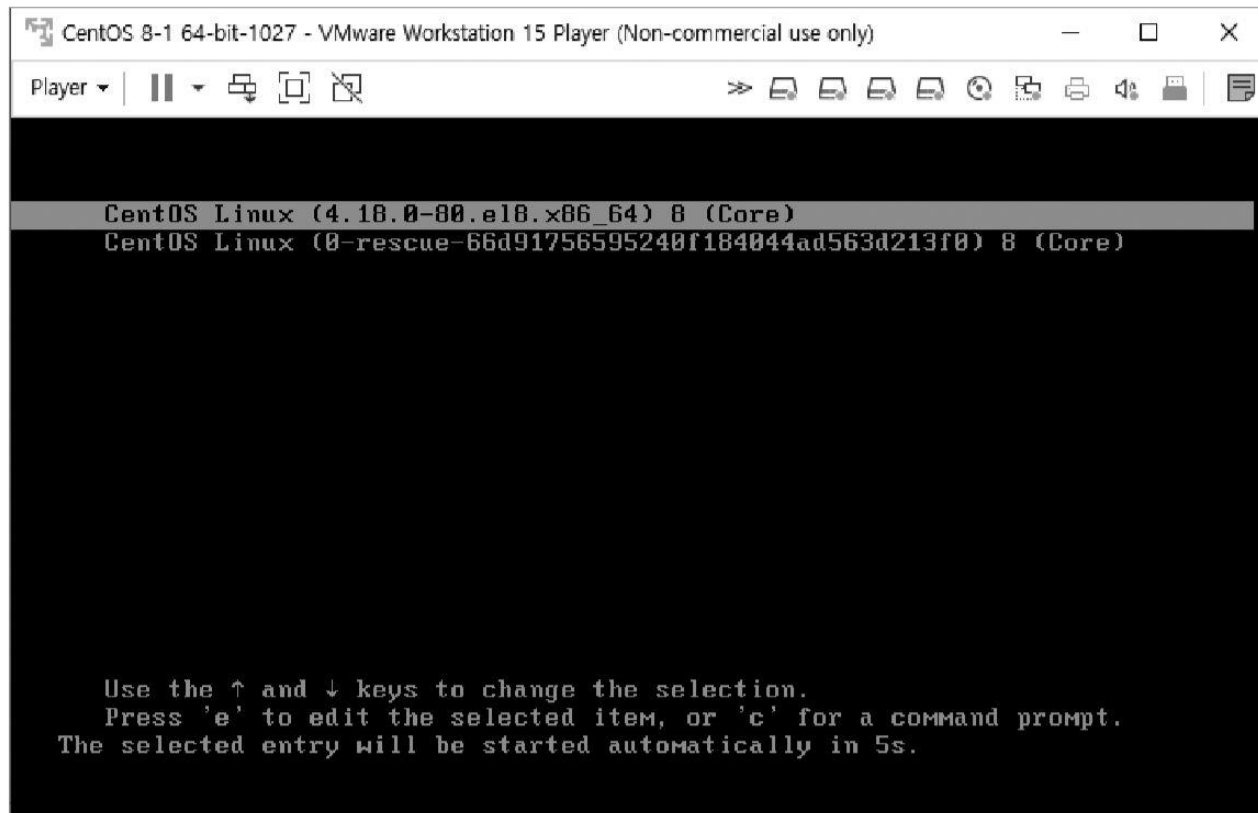


그림 8-3 부트 로더 시작 화면

01.리눅스 시스템의 부팅

■ 부트 로더 단계

- 부트 로더는 리눅스 커널을 메모리에 로딩하는 역할을 수행
- 리눅스 커널은 /boot 디렉터리 아래에 'vmlinuz-버전명'의 형태로 제공
- 리눅스의 대표적인 부트 로더는 GRUB
- rescue 버전은 응급 상황에서 시스템을 복구하기 위해 사용, 윈도우의 안전모드와 유사

```
[user1@localhost ~]$ ls /boot/vm*  
/boot/vmlinuz-0-rescue-66d91756595240f184044ad563d213f0  
/boot/vmlinuz-4.18.0-80.el8.x86_64
```

01.리눅스 시스템의 부팅

■ 커널 초기화 단계

- 커널은 가장 먼저 시스템에 연결된 메모리, 디스크, 키보드, 마우스 등 장치들을 검사
 - 장치 검사 등 기본적인 초기화 과정이 끝나면 커널은 fork를 사용하지 않고 프로세스와 스레드 생성
 - 이 프로세스들은 메모리 관리 같은 커널의 여러 가지 동작을 수행
 - 프로세스의 개수와 종류는 리눅스의 버전과 종류에 따라 다름
 - 이 프로세스는 일반적인 프로세스와 구분되도록 대괄호([])로 표시하며, 주로 PID 번호가 낮게 배정
-
- 커널 프로세스가 생성되면 커널이 수행할 작업이 끝남

```
[user1@localhost ~]$ ps -ef | more
UID          PID    PPID  C STIME TTY          TIME CMD
root          1        0  0 20:24 ?        00:00:04 /usr/lib/systemd/systemd --swi
tched-root --system --deserialize 17
root          2        0  0 20:24 ?        00:00:00 [kthreadd]
root          3        2  0 20:24 ?        00:00:00 [rcu_gp]
root          4        2  0 20:24 ?        00:00:00 [rcu_par_gp]
root          6        2  0 20:24 ?        00:00:00 [kworker/0:0H]
root          8        2  0 20:24 ?        00:00:00 [mm_percpu_wq]
root          9        2  0 20:24 ?        00:00:00 [ksoftirqd/0]
root         10        2  0 20:24 ?        00:00:00 [rcu_sched]
root         11        2  0 20:24 ?        00:00:00 [migration/0]
root         12        2  0 20:24 ?        00:00:00 [watchdog/0]
(생략)
```

01.리눅스 시스템의 부팅

■ systemd 서비스 단계

- systemd 서비스는 기존의 init 스크립트를 대체한 것으로 다양한 서비스를 동작시킴
- 각 서비스가 시작하는 과정은 화면에 메시지로 출력되나 CentOS에서는 메시지 대신 이미지를 출력
- 이 이미지를 부트 스플래시라고 하며 CentOS의 버전에 따라 다르게 출력

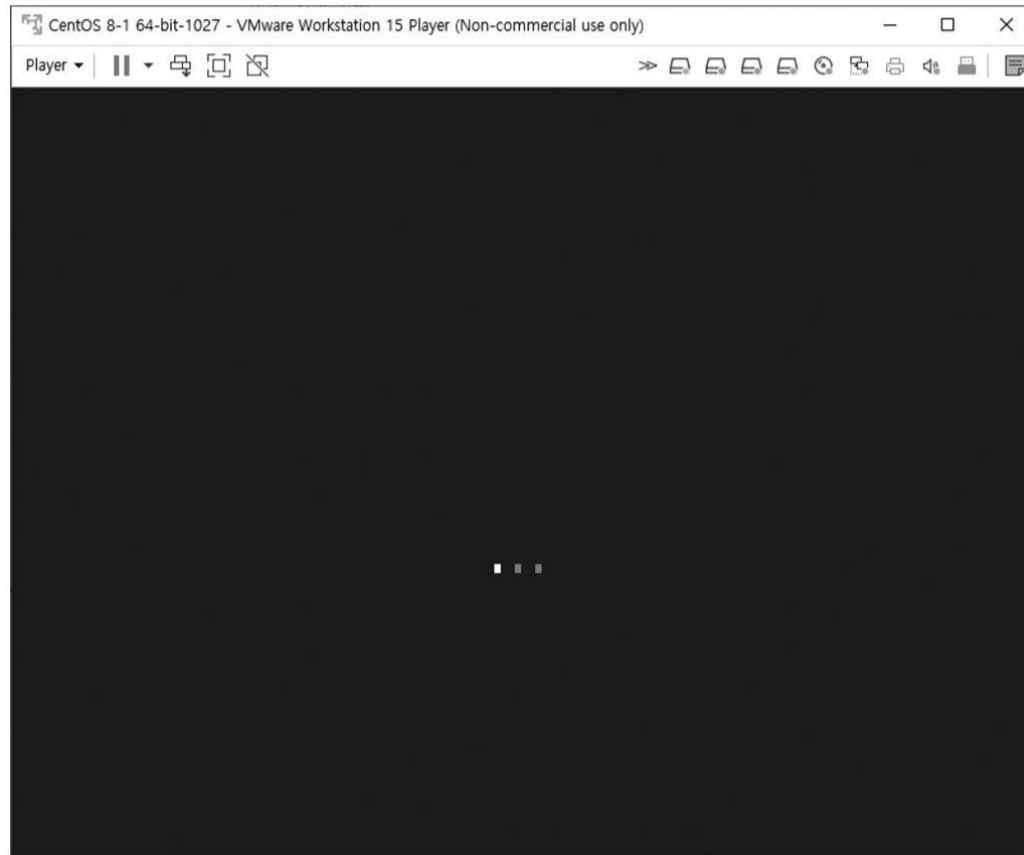
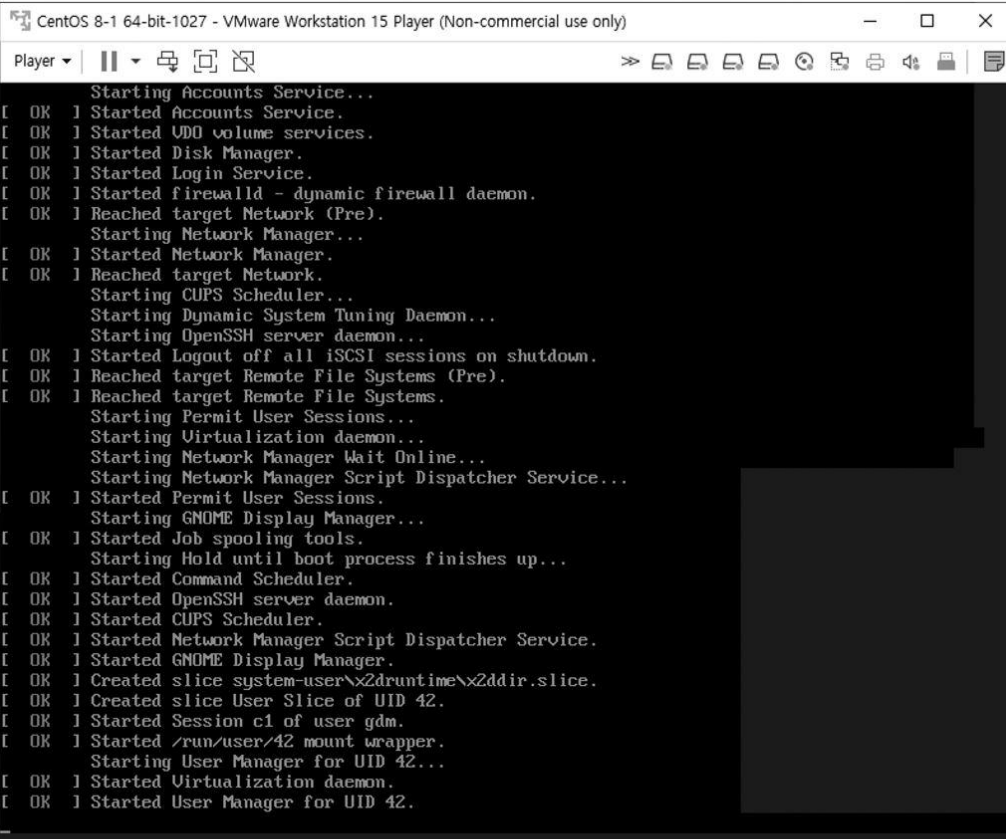


그림 8-4 CentOS의 부트 스플래시 화면

01.리눅스 시스템의 부팅

■ systemd 서비스 단계

- 부트 스플래시 화면이 진행 중일 때 [Alt]+d(소문자 D)키를 누르면 메시지가 출력되는 화면으로 전환

A screenshot of a VMware Workstation 15 Player window titled "CentOS 8-1 64-bit-1027 - VMware Workstation 15 Player (Non-commercial use only)". The window displays a terminal window with a black background and white text showing the output of the systemd boot process. The messages include starting various services like Accounts Service, UDD volume services, Disk Manager, Login Service, firewalld, Network Manager, CUPS Scheduler, Dynamic System Tuning Daemon, OpenSSH server daemon, and logging off iSCSI sessions. It also shows reaching targets like Network (Pre), Remote File Systems (Pre), and Permit User Sessions, followed by starting GNOME Display Manager, job spooling tools, and the Command Scheduler. The process concludes with creating user slices and starting the User Manager for UID 42.

```
Starting Accounts Service...
[ OK ] Started Accounts Service.
[ OK ] Started UDD volume services.
[ OK ] Started Disk Manager.
[ OK ] Started Login Service.
[ OK ] Started firewalld - dynamic firewall daemon.
[ OK ] Reached target Network (Pre).
      Starting Network Manager...
[ OK ] Started Network Manager.
[ OK ] Reached target Network.
      Starting CUPS Scheduler...
      Starting Dynamic System Tuning Daemon...
      Starting OpenSSH server daemon...
[ OK ] Started Logout off all iSCSI sessions on shutdown.
[ OK ] Reached target Remote File Systems (Pre).
[ OK ] Reached target Remote File Systems.
      Starting Permit User Sessions...
      Starting Virtualization daemon...
      Starting Network Manager Wait Online...
      Starting Network Manager Script Dispatcher Service...
[ OK ] Started Permit User Sessions.
      Starting GNOME Display Manager...
[ OK ] Started Job spooling tools.
      Starting Hold until boot process finishes up...
[ OK ] Started Command Scheduler.
[ OK ] Started OpenSSH server daemon.
[ OK ] Started CUPS Scheduler.
[ OK ] Started Network Manager Script Dispatcher Service.
[ OK ] Started GNOME Display Manager.
[ OK ] Created slice system-user\x2druntime\x2ddir.slice.
[ OK ] Created slice User Slice of UID 42.
[ OK ] Started Session c1 of user gdm.
[ OK ] Started /run/user/42 mount wrapper.
      Starting User Manager for UID 42...
[ OK ] Started Virtualization daemon.
[ OK ] Started User Manager for UID 42.
```

그림 8-5 부팅 메시지 출력 화면

01.리눅스 시스템의 부팅

■ systemd 서비스 단계

- 부팅 시 출력되는 메시지는 각종 서비스가 정상적으로 시작하는지(OK) 아니면 실패인지(FAIL)를 나타냄
- 이 메시지는 부팅 후 `dmesg` 명령이나 `more /var/log/boot.log` 명령으로 확인할 수 있음
- `dmesg` 명령으로 출력되는 메시지에는 데몬의 시작과 관련된 메시지뿐만 아니라 하드웨어 검사와 관련된 메시지도 모두 포함

```
[user1@localhost ~]$ dmesg | more
[    0.000000] Linux version 4.18.0-80.el8.x86_64 (mockbuild@kbuilder.bsys.centos.org) (gcc version 8.2.1 20180905 (Red Hat 8.2.1-3) (GCC)) #1 SMP Tue Jun 4 09:19:46 UTC 2019
[    0.000000] Command line: BOOT_IMAGE=(hd0,msdos1)/vmlinuz-4.18.0-80.el8.x86_64 root=/dev/mapper/cl-root ro crashkernel=auto resume=/dev/mapper/cl-swap rd.lvm.lv=cl/root rd.lvm.lv=cl/swap rhgb quiet
[    0.000000] Disabled fast string operations
[    0.000000] x86/fpu: Supporting XSAVE feature 0x001: 'x87 floating point registers'
(생략)
```

01.리눅스 시스템의 부팅

■ systemd 서비스 단계

- 전통적으로 유닉스에서는 init 프로세스가 서비스를 실행하였으므로 init 프로세스는 PID가 1번이었음
- 그러나 CentOS는 버전 7부터 init 대신에 시스템과 서비스 관리자로 systemd를 사용하기 시작했고, systemd 프로세스가 1번 프로세스가 됨

```
[user1@localhost ~]$ ps -ef | more
UID          PID    PPID  C STIME TTY          TIME CMD
root          1        0  0  20:24 ?        00:00:04 /usr/lib/systemd/systemd --swi
tched-root --system --deserialize 17
root          2        0  0  20:24 ?        00:00:00 [kthreadd]
root          3        2  0  20:24 ?        00:00:00 [rcu_gp]
(생략)
```

01.리눅스 시스템의 부팅

■ systemd 서비스 단계

- 마지막으로 systemd 서비스 단계에서는 데몬을 모두 실행한 뒤 그래픽 로그인 시스템인 GDM을 동작
- 로그인 프롬프트 화면을 출력

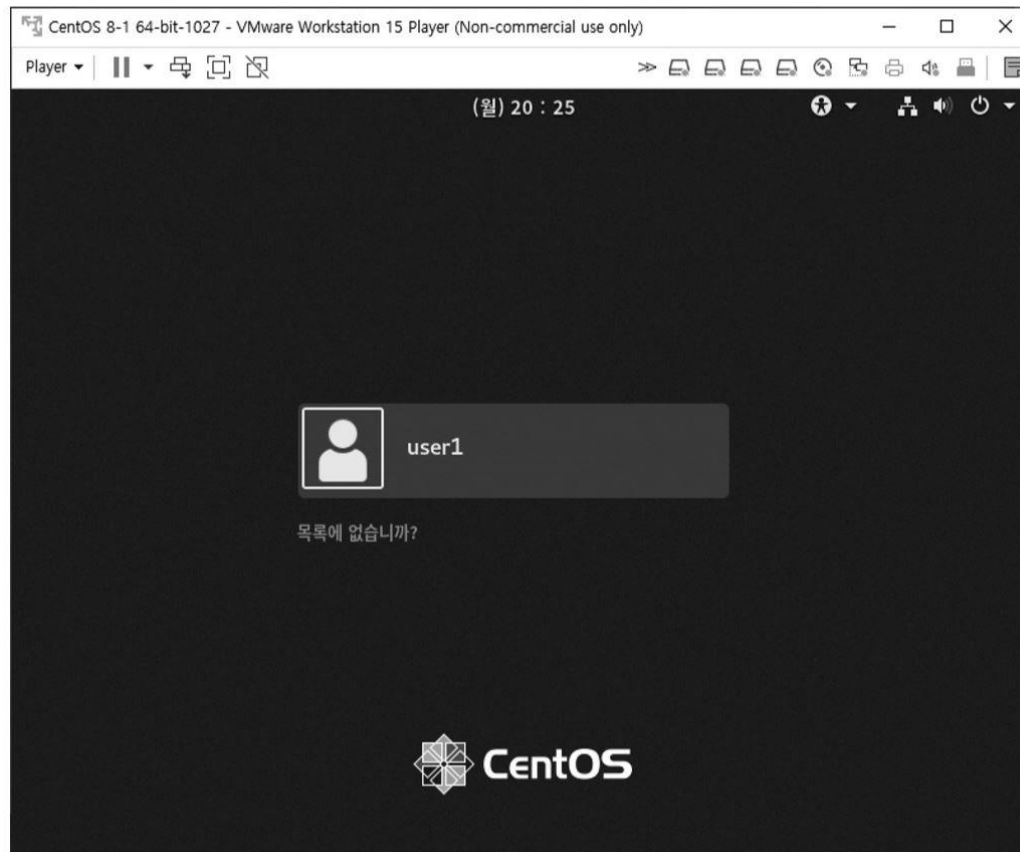


그림 8-6 로그인 프롬프트 화면

02.systemd 서비스

■ init 프로세스와 런레벨

■ init 프로세스

- PID가 1번인 프로세스로 모든 프로세스의 조상 역할을 수행
- 부팅 과정에서 각종 서비스를 제공하는 셸 스크립트 파일을 실행
- 관련된 설정 파일은 /etc/inittab
- 이젠 init를 사용하지 않고 systemd를 사용

```
[user1@localhost ~]$ cat /etc/inittab
# inittab is no longer used.
#
# ADDING CONFIGURATION HERE WILL HAVE NO EFFECT ON YOUR SYSTEM.
#
# Ctrl-Alt-Delete is handled by /usr/lib/systemd/system/ctrl-alt-del.target
#
# systemd uses 'targets' instead of runlevels. By default, there are two main
targets:
#
# multi-user.target: analogous to runlevel 3
# graphical.target: analogous to runlevel 5
#
# To view current default target, run:
# systemctl get-default
#
# To set a default target, run:
# systemctl set-default TARGET.target
```


02.systemd 서비스

■ init 프로세스와 런레벨

- init 프로세스가 실행하는 스크립트 파일은 /etc/rc.d/init.d 디렉터리에 위치
- /etc/rc*.d 디렉터리에 있는 파일들은 /etc/rc.d/init.d 파일에 대한 심볼릭 링크
- 현재는 systemd 서비스에 의해 대부분 대체되고 일부 파일만 유지

```
[user1@localhost ~]$ ls /etc/rc.d/init.d  
README  functions
```

- etc/rc.d/init.d 디렉터리의 README 파일에 전통적인 init 스크립트가 systemd 기반으로 바뀐 설명이 있음

```
[user1@localhost ~]$ cat /etc/rc.d/init.d/README  
You are looking for the traditional init scripts in /etc/rc.d/init.d,  
and they are gone?  
  
Here's an explanation on what's going on:  
  
You are running a systemd-based OS where traditional init scripts have  
been replaced by native systemd services files. Service files provide  
very similar functionality to init scripts. To make use of service  
files simply invoke "systemctl", which will output a list of all  
currently running services (and other units). Use "systemctl  
list-unit-files" to get a listing of all known unit files, including  
stopped, disabled and masked ones. Use "systemctl start  
foobar.service" and "systemctl stop foobar.service" to start or stop a  
service, respectively. For further details, please refer to  
systemctl(1).  
(생략)
```

02.systemd 서비스

■ init 프로세스와 런레벨

- 런레벨 : init는 시스템의 상태를 일곱 개로 정의하여 구분하고 각 상태에 따라 셸 스크립트를 실행
이 단계들을 런 레벨이라고 함

표 8-1 리눅스의 init 런레벨

런레벨	의미	관련 스크립트의 위치
0	시스템 종료	/etc/rc0.d
1, S	단일 사용자 모드	/etc/rc1.d
2	다중 사용자 모드(NFS를 실행하지 않음)	/etc/rc2.d
3	다중 사용자 모드(NFS 포함)	/etc/rc3.d
4	사용하지 않음(예비 번호)	/etc/rc4.d
5	X11 상태로 부팅	/etc/rc5.d
6	재시작	/etc/rc6.d

02.systemd 서비스

■ systemd 기본 개념

- CentOS는 버전 7부터 거의 대부분의 서비스가 systemd 기반으로 바뀜
- systemd의 장점
 - 소켓 기반으로 동작하여 inetd와 호환성을 유지
 - 셸과 독립적으로 부팅이 가능
 - 마운트 제어가 가능
 - fsck 제어가 가능
 - 시스템 상태에 대한 스냅샷을 유지
 - SELinux와 통합이 가능
 - 서비스에 시그널을 전달할 수 있음
 - 쉼다운 전에 사용자 세션의 안전한 종료가 가능

02.systemd 서비스

■ systemd 기본 개념

■ systemd 유닛

- systemd는 전체 시스템을 시작하고 관리하는 데 유닛이라 부르는 구성 요소를 사용
- systemd는 관리 대상의 이름을 '서비스명.유닛 종류'의 형태로 관리
- 각 유닛은 같은 이름과 종류로 구성된 설정 파일과 동일한 이름을 사용

표 8-2 systemd 유닛의 종류

유닛	기능	예
service	시스템 서비스 유닛으로 데몬을 시작 · 종료 · 재시작 · 로드한다.	atd.service
target	유닛을 그룹핑한다. (예 multi-user.target → 런레벨 5에 해당하는 유닛)	basic.target
automount	디렉터리 계층 구조에서 자동 마운트 포인트를 관리한다.	proc-sys-fs-binfmt_misc.automount
device	리눅스 장치 트리에 있는 장치를 관리한다.	sys-module-fuse.device
mount	디렉터리 계층 구조의 마운트 포인트를 관리한다.	boot.mount
path	파일 시스템의 파일이나 디렉터리 등 경로를 관리한다.	cups.path
scope	외부에서 생성된 프로세스를 관리한다.	init.scope
slice	시스템의 프로세스를 계층적으로 관리한다.	system-getty.slice
socket	소켓을 관리하는 유닛으로 AF_INET, AF_INET6, AF_UNIX 소켓 스트림과 데이터그램, FIFO를 지원한다.	dbus.socket
swap	스왑 장치를 관리한다.	dev-mapper-fedoraWx2dswap. swap
timer	타이머와 관련된 기능을 관리한다.	dnf-makecache.timer

02.systemd 서비스

■ systemd 관련 명령

- `systemctl` : `systemd` 기반으로 서비스를 시작하거나 종료할 때 사용하는 명령
유닛을 지정할 때 유닛의 종류는 제외해도 됨

systemctl

- **기능** `systemd` 서비스를 제어한다.
- **형식** `systemctl` [옵션] [명령] [유닛명]
- **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`: 유닛 상태를 출력한다.
`enable`: 부팅 시 유닛이 시작되도록 설정한다.
`disable`: 부팅 시 유닛이 시작하지 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`

02.systemd 서비스

■ systemd 관련 명령

■ 동작중인 유닛 출력하기

- 옵션이나 명령 없이 systemctl 명령만 사용하면 현재 동작 중인 유닛이 출력

```
[user1@localhost ~]$ systemctl
```

UNIT	LOAD	ACTIVE	SUB	DESCRIPTION
(생략)				
basic.target	loaded	active	active	Basic System
bluetooth.target	loaded	active	active	Bluetooth
cryptsetup.target	loaded	active	active	Local Encrypted Volumes
getty.target	loaded	active	active	Login Prompts
graphical.target	loaded	active	active	Graphical Interface
local-fs-pre.target	loaded	active	active	Local File Systems (Pre)
local-fs.target	loaded	active	active	Local File Systems
multi-user.target	loaded	active	active	Multi-User System
network-online.target	loaded	active	active	Network is Online
network-pre.target	loaded	active	active	Network (Pre)
network.target	loaded	active	active	Network
nfs-client.target	loaded	active	active	NFS client services
(생략)				

02.systemd 서비스

■ systemd 관련 명령

- 전체 유닛 출력하기 : -a
 - systemctl 명령에 -a 옵션을 지정하면 전체 유닛이 출력

```
[user1@localhost ~]$ systemctl -a
```

UNIT	LOAD	ACTIVE	SUB	DESCRIPTION
boot.automount	not-found	inactive	dead	boot.automount
proc-sys-fs-binfmt_misc.automount	loaded	active	waiting	Arbitrary Exec>
dev-block-8:17.device	loaded	active	plugged	LVM PV r930iM-f78e-EbB>
dev-block-8:18.device	loaded	active	plugged	LVM PV uP0uft-4dQ5-BD7>
dev-block-8:2.device	loaded	active	plugged	LVM PV u7eKdg-zNiu-kra>
dev-block-8:33.device	loaded	active	plugged	LVM PV l1foLX-g2dY-cmE>
dev-block-8:34.device	loaded	active	plugged	LVM PV o01pjh-nH7z-aoW>
dev-cdrom.device	loaded	active	plugged	VMware_Virtual_IDE_CDR>
dev-cl-root.device	loaded	active	plugged	/dev/cl/root
dev-cl-swap.device	loaded	active	plugged	/dev/cl/swap

(생략)

02.systemd 서비스

■ systemd 관련 명령

- 특정 유닛 출력하기 : -t
 - 특정 종류의 유닛만 출력하려면 -t 옵션을 사용

```
[user1@localhost ~]$ systemctl -t service
```

UNIT	LOAD	ACTIVE	SUB	DESCRIPTION
accounts-daemon.service	loaded	active	running	Accounts Service
alsa-state.service	loaded	active	running	Manage Sound Card State (>
atd.service	loaded	active	running	Job spooling tools
auditd.service	loaded	active	running	Security Auditing Service
avahi-daemon.service	loaded	active	running	Avahi mDNS/DNS-SD Stack
bluetooth.service	loaded	active	running	Bluetooth service
bolt.service	loaded	active	running	Thunderbolt system service
chronyd.service	loaded	active	running	NTP client/server
colord.service	loaded	active	running	Manage, Install and Gener>
crond.service	loaded	active	running	Command Scheduler
cups.service	loaded	active	running	CUPS Scheduler
(생략)				

02.systemd 서비스

■ systemd 관련 명령

- 유닛의 상태 확인하기 : status
 - 유닛의 상태를 확인하려면 status 명령을 사용

```
[user1@localhost ~]$ systemctl status crond.service
● crond.service - Command Scheduler
   Loaded: loaded (/usr/lib/systemd/system/crond.service; enabled; vendor prese>
   Active: active (running) since Mon 2019-11-11 21:00:06 KST; 56min ago
 Main PID: 1139 (crond)
    Tasks: 1 (limit: 11362)
   Memory: 2.1M
    CGroup: /system.slice/crond.service
           └─1139 /usr/sbin/crond -n
lines 1-8/8 (END)
```

02.systemd 서비스

■ systemd 관련 명령

- 유닛 서비스 정지하기 : stop
 - 유닛 서비스를 정지하려면 stop 명령을 사용
 - 유닛 목록의 아무 서비스나 정지하지 말도록 주의

```
[user1@localhost init.d]$ su -  
암호:  
[root@localhost ~]# systemctl stop crond.service  
[root@localhost ~]# systemctl status crond  
● crond.service - Command Scheduler  
   Loaded: loaded (/usr/lib/systemd/system/crond.service; enabled; vendor prese>  
   Active: inactive (dead) since Mon 2019-11-11 21:58:55 KST; 7s ago  
   Process: 1139 ExecStart=/usr/sbin/crond -n $CRONDARGS (code=exited, status=0/>  
   Main PID: 1139 (code=exited, status=0/SUCCESS)
```

```
11월 11 21:00:09 localhost.localdomain crond[1139]: (CRON) INFO (running with i>  
11월 11 21:01:01 localhost.localdomain CROND[2390]: (root) CMD (run-parts /etc/>  
11월 11 21:01:01 localhost.localdomain anacron[2399]: Anacron started on 2019-1>  
11월 11 21:01:01 localhost.localdomain anacron[2399]: Will run job `cron.daily`>  
11월 11 21:01:01 localhost.localdomain anacron[2399]: Jobs will be executed seq>  
11월 11 21:46:01 localhost.localdomain anacron[2399]: Job `cron.daily' started  
11월 11 21:46:01 localhost.localdomain anacron[2399]: Job `cron.daily' terminat>  
11월 11 21:46:01 localhost.localdomain anacron[2399]: Normal exit (1 job run)  
11월 11 21:58:55 localhost.localdomain systemd[1]: Stopping Command Scheduler...  
11월 11 21:58:55 localhost.localdomain systemd[1]: Stopped Command Scheduler.  
lines 1-16/16 (END)
```

02.systemd 서비스

■ systemd 관련 명령

- 유닛 서비스 시작하기 : start
 - 유닛 서비스를 시작하려면 start 명령을 사용

```
[root@localhost ~]# systemctl start crond.service
[root@localhost ~]# systemctl is-active crond.service
active
```

02.systemd 서비스

■ systemd와 런레벨

- 런레벨은 현재 시스템의 상태를 나타내는 한 자리 숫자(문자 S, s 포함)
- 이에 대응하는 systemd의 target 유닛은 /usr/lib/systemd/system 디렉터리에 있음

표 8-3 런레벨과 target 유닛의 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

02.systemd 서비스

■ systemd와 런레벨

■ 현재 target과 런레벨 확인하기

- Ex) 현재 target을 확인

```
[root@localhost ~]# systemctl get-default  
graphical.target
```

- 런레벨을 확인하려면 runlevel 명령을 사용

```
[root@localhost ~]# runlevel  
N 5
```

02.systemd 서비스

■ systemd와 런레벨

■ 기본 target 지정하기

- 기본 런레벨은 예전에는 /etc/inittab 파일에 지정
- 지금은 default.target이 가리키는 target 유닛으로 바뀜
- default.target은 /etc/systemd/system 디렉터리 아래에 있으며 심볼릭 링크 파일

```
[root@localhost ~]# ls -l /etc/systemd/system/de*  
lrwxrwxrwx. 1 root root 36 10월 27 20:53 /etc/systemd/system/default.target -> /lib/  
systemd/system/graphical.target
```

- Ex) default.target이 가리키는 기본 target은 다음과 같은 형식으로 지정

```
systemctl set-default <name of target>.target
```

- 이 명령은 /etc/systemd/system 디렉터리 아래의 심볼릭 링크인 default.target이 가리키는 target 파일을 변경
- Ex) 현재 target인 graphical.target에서 multi-user.target으로 변경

```
[root@localhost ~]# systemctl set-default multi-user.target  
Removed /etc/systemd/system/default.target.  
Created symlink /etc/systemd/system/default.target -> /usr/lib/systemd/system/multi-  
user.target.  
[root@localhost ~]# ls -l /etc/systemd/system/de*  
lrwxrwxrwx. 1 root root 41 11월 11 22:14 /etc/systemd/system/default.target -> /usr/  
lib/systemd/system/multi-user.targe
```

02.systemd 서비스

■ systemd와 런레벨

■ 기본 target 지정하기

- 런레벨 0이나, 런레벨 6에 해당하는 target을 기본으로 지정해서는 안됨
- 기본 target은 graphical.target(런레벨 5)으로 해놓는 것이 좋음

```
[root@localhost ~]# systemctl set-default graphical.target
Removed /etc/systemd/system/default.target.
Created symlink /etc/systemd/system/default.target → /usr/lib/systemd/system/graphical.target.
```

02.systemd 서비스

■ systemd와 런레벨

■ target 변경하기

- isolate 명령으로 systemd에서 런레벨을 변경하는 것도 가능
- Ex) multi-user.target(런레벨 3)으로 변경

```
systemctl isolate multi-user
```

```
systemctl isolate runlevel3
```

- Ex) graphical-user.target(런레벨 5)으로 변경

```
systemctl isolate graphical
```

```
systemctl isolate runlevel5
```


02.systemd 서비스

■ systemd와 런레벨

■ 런레벨 변경하기 : telinit, init

- init은 1번 프로세스의 이름이기도 하지만 init 프로세스의 런레벨을 바꿀 때 사용

```
[root@localhost ~]# ls -l /sbin/init
lrwxrwxrwx. 1 root root 22 5월 23 23:48 /sbin/init → ../lib/systemd/systemd
```

- Ex) init 명령만 입력할 경우

```
[root@localhost ~]# init
init: required argument missing.
```

- Ex) init -help로 사용법을 알아볼 경우

```
[root@localhost ~]# init --help
init [OPTIONS...] {COMMAND}
Send control commands to the init daemon.

    --help      Show this help
    --no-wall   Don't send wall message before halt/power-off/reboot

Commands:
  0              Power-off the machine
  6              Reboot the machine
  2, 3, 4, 5     Start runlevelX.target unit
  1, s, S       Enter rescue mode
  q, Q           Reload init daemon configuration
  u, U           Reexecute init daemon
```

02.systemd 서비스

■ systemd와 런레벨

■ 런레벨 변경하기 : telinit, init

- 런레벨을 바꾸는 명령으로 telinit도 있음

```
[root@localhost ~]# ls -l /sbin/telinit
lrwxrwxrwx. 1 root root 16 5월 23 23:48 /sbin/telinit -> ../bin/systemctl
```

- Ex) telinit 명령을 실행하면 init 명령과 같은 결과가 출력

```
[root@localhost ~]# telinit --help
telinit [OPTIONS...] {COMMAND}

Send control commands to the init daemon.

    --help      Show this help
    --no-wall   Don't send wall message before halt/power-off/reboot

Commands:
    0           Power-off the machine
    6           Reboot the machine
    2, 3, 4, 5  Start runlevelX.target unit
    1, s, S     Enter rescue mode
    q, Q        Reload init daemon configuration
    u, U        Reexecute init daemon
```

02.systemd 서비스

■ systemd와 런레벨

- 단일 사용자 모드로 전환하기 : rescue.target(런레벨 1)
 - 시스템에 문제가 있을 경우 시스템을 rescue.target 유닛(런레벨 1, 런레벨 S)으로 변경하여 점검
 - 윈도의 안전 모드 같은 것으로, 다중 사용자 모드에서 시스템 관리자만 사용할 수 있는 단일 사용자 모드로 전환
 - 이 모드로 변환하기 전에 다른 사용자들은 로그아웃을 해야함

```
systemctl isolate rescue
```

```
systemctl isolate runlevel1
```

```
init 1
```

```
telinit S
```

- 단일 사용자 모드로 전환하면 그래픽 환경이었던 리눅스가 텍스트 모드로 바뀜
- 단일 사용자 모드에서 다중 사용자 모드로 전환하려면 reboot 명령이나 systemctl default 명령을 사용

03.리눅스 시스템의 종료

■ shutdown 명령

- shutdown 명령 : 리눅스 시스템을 가장 정상적으로 종료할 때 사용
- 다른 명령들과 달리 shutdown 명령은 다양한 종료 방법을 제공
- 시스템 종료 외에 런레벨을 바꿀 때도 사용

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 시간이다(hh:mm, +m, now).

메시지: 모든 사용자에게 보낼 메시지다.
- **사용 예**

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

03.리눅스 시스템의 종료

■ shutdown 명령

■ 시스템 즉시 종료하기

- 시스템을 즉시 종료하려면 -h 옵션과 함께 현재 시간(now)을 지정

```
[root@localhost ~]# shutdown -h now
```

■ 섯다운 메시지 보내고 종료하기

- 시간을 now로 지정하면 안 되고 특정 시간을 지정
- Ex) 2분 후에 시스템이 종료

```
[root@localhost ~]# shutdown -h +2 "System is going down in 2 min"
```

03.리눅스 시스템의 종료

■ shutdown 명령

■ 시스템 재시작하기

- shutdown 명령으로 시스템을 재시작하려면 -r 옵션을 사용
- Ex) 3분 후에 시스템을 재시작

```
[root@localhost ~]# shutdown -r +3
```

■ 명령 취소하기

- shutdown 명령을 취소하려면 -c 옵션을 사용

```
[root@localhost ~]# shutdown -c
```

■ 메시지만 보내기

- 실제로 shutdown 명령을 실행하지는 않고 사용자들에게 메시지만 보내려면 -k 옵션을 사용

```
[root@localhost ~]# shutdown -k 2
```

03.리눅스 시스템의 종료

■ 런레벨 변경 명령

- 시스템을 종료하는 다른 방법은 런레벨을 변경
- telinit 명령으로 런레벨을 0으로 변경하면 시스템이 종료되고, 6으로 변경하면 재시작
- systemd 기능을 사용할 때는 target을 변경

■ 런레벨 변경하기

- 런레벨 0은 시스템 종료 상태이므로 런레벨을 0으로 바꾸면 시스템이 종료

```
[root@localhost ~]# telinit 0
```

- 재시작하려면 런레벨을 6으로 변경

```
[root@localhost ~]# telinit 6
```

03.리눅스 시스템의 종료

■ 런레벨 변경 명령

■ systemd로 종료하기

- systemd에서 target 유닛을 변경하면 시스템을 종료하거나 재시작할 수 있음

```
[root@localhost ~]# systemctl isolate poweroff.target
```

```
[root@localhost ~]# systemctl isolate runlevel0.target
```

```
[root@localhost ~]# systemctl isolate reboot.target
```

```
[root@localhost ~]# systemctl isolate runlevel6.target
```


03.리눅스 시스템의 종료

■ 기타 시스템 종료 명령

- 시스템을 종료하거나 재시작하기 위해 사용할 수 있는 명령으로 halt, poweroff, reboot등이 있음
- 이 명령들은 모두 systemctl 명령의 심볼릭 링크

```
[root@localhost ~]# ls -l /sbin/halt
lrwxrwxrwx. 1 root root 16  5월 23 23:48 /sbin/halt -> ../bin/systemctl
[root@localhost ~]# ls -l /sbin/poweroff
lrwxrwxrwx. 1 root root 16  5월 23 23:48 /sbin/poweroff -> ../bin/systemctl
[root@localhost ~]# ls -l /sbin/reboot
lrwxrwxrwx. 1 root root 16  5월 23 23:48 /sbin/reboot -> ../bin/systemctl
```

- halt, reboot, poweroff 명령: /var/log/wtmp 파일에 시스템 종료 기록을 남기고 시스템을 종료하거나 재시작
- 이 명령들은 런레벨이 1~5일 때 내부적으로 shutdown 명령을 호출
- 사용할 수 있는 옵션
 - -n : 재시작이나 종료 전에 sync를 호출하지 않는다.
 - -w : 실질적으로 재시작하거나 종료하지는 않지만 wtmp 파일에 기록을 남김
 - -d : wtmp 파일에 기록을 남기지 않음 -n 옵션은 -d 옵션을 포함
 - -f : 강제로 명령을 실행하며 shutdown을 호출하지 않음
 - -p : 시스템의 전원을 종료

04.데몬 프로세스

■ 데몬

- 리눅스의 백그라운드에서 동작하면서 특정한 서비스를 제공하는 프로세스
- 리눅스 시스템에서 동작하는 웹 서버, 데이터베이스 서버, 원격 접속 서버 등 각종 서비스를 제공하는 프로세스

■ 데몬의 동작 방식

- 데몬 혼자서 스스로 동작하는 독자형과 데몬을 관리하는 슈퍼데몬에 의해 동작하는 방식이 있음

■ 독자형

- 시스템의 백그라운드에서 서비스별로 항상 동작
- 자주 호출되는 데몬이 아니라면 시스템의 자원을 낭비할 우려

■ 슈퍼데몬에 의한 동작 방식

- 평소에는 슈퍼데몬만 동작하다가 서비스 요청이 오면 슈퍼데몬이 해당 데몬을 동작 시킴
- 독자형보다는 서비스에 응답하는 데 시간이 약간 더 걸릴 수 있지만 자원을 효율적으로 사용한다는 장점.

■ 슈퍼데몬

- 데몬의 종류가 늘어나자 이를 관리하기 위한 슈퍼데몬이 등장
- 유닉스의 슈퍼데몬은 이름이 inetd였으나 CentOS에서는 보안 기능이 포함된 xinetd를 사용
- 슈퍼데몬은 네트워크 서비스를 제공하는 데몬만 관리

04.데몬 프로세스

■ 데몬의 조상

■ systemd 데몬

- systemd는 init를 대체한 데몬
- systemd는 1번 프로세스로서 프로세스 대부분의 조상 프로세스
- 시스템의 상태를 종합적으로 관리하는 역할을 수행
- Ex) pstree 명령으로 프로세스의 실행 구조를 확인

```
[root@localhost ~]# pstree
systemd├─ModemManager──2*[{ModemManager}]
      │├─NetworkManager──2*[{NetworkManager}]
      │├─VGAuthService
      │├─accounts-daemon──2*[{accounts-daemon}]
      │├─alsactl
      │├─atd
      │├─auditd├─sdispatch
      │      │└─2*[{auditd}]
      │├─avahi-daemon──avahi-daemon
      │├─bluetoothd
      │├─boltd──2*[{boltd}]
      │├─chronyd
      │├─colord──2*[{colord}]
      │├─crond
      │├─cupsd
      │├─dbus-daemon──{dbus-daemon}
      │├─dnsmasq──dnsmasq
      └─firewalld──{firewalld}
```

(생략)

04.데몬 프로세스

■ 데몬의 조상

■ 커널 스레드 데몬

- 커널 데몬 : 커널의 일부분을 프로세스처럼 관리하는 데몬
- ps 명령으로 확인했을 때 대괄호([])에 들어 있는 프로세스
- 예전에는 대부분 k로 시작했으나 요즘에는 이를 반드시 준수하지 않음
- 대부분 입출력이나 메모리 관리, 디스크 동기화 등을 수행하며 대체로 낮은 PID가 할당
- 커널 데몬을 동작시키는 조상 데몬은 커널 스레드 데몬

- Ex) ps 명령으로 확인해보면 모든 커널 데몬의 PPID가 2번임을 확인가능

```
[root@localhost sbin]# ps -ef | more
UID          PID    PPID  C STIME TTY          TIME CMD
UID          PID    PPID  C STIME TTY          TIME CMD
root          1        0  0 21:20 ?        00:00:02 /usr/lib/systemd/systemd --swi
tched-root --system --deserialize 18
root          2        0  0 21:20 ?        00:00:00 [kthreadd]
root          3        2  0 21:20 ?        00:00:00 [rcu_gp]
root          4        2  0 21:20 ?        00:00:00 [rcu_par_gp]
root          5        2  0 21:20 ?        00:00:00 [kworker/0:0-xfs-buf/dm-0]
root          6        2  0 21:20 ?        00:00:00 [kworker/0:0H]
root          7        2  0 21:20 ?        00:00:00 [kworker/u256:0-events_unbound
]
(생략)
```

04.데몬 프로세스

■ 데몬의 조상

■ 주요 데몬

표 8-4 리눅스의 주요 데몬

데몬	기능	데몬	기능
atd	특정 시간에 실행하도록 예약한 명령을 실행한다(at 명령으로 예약).	popd	기본 편지함 서비스를 제공한다.
crond	주기적으로 실행하도록 예약한 명령을 실행한다.	routed	자동 IP 라우터 테이블 서비스를 제공한다.
dhcpcd	동적으로 IP 주소를 부여하는 서비스를 제공한다.	smb	삼바 서비스를 제공한다.
httpd	웹 서비스를 제공한다.	syslogd	로그 기록 서비스를 제공한다.
lpd	프린트 서비스를 제공한다.	sshd	원격 보안 접속 서비스를 제공한다.
nfs	네트워크 파일 시스템 서비스를 제공한다.	in.telnetd	원격 접속 서비스를 제공한다.
named	DNS 서비스를 제공한다.	ftpd	파일 송수신 서비스를 제공한다.
sendmail	이메일 서비스를 제공한다.	ntpd	시간 동기화 서비스를 제공한다.
smtpd	메일 전송 데몬이다.		

05.부트 로더

■ GRUB의 개요

■ GRUB

- 'grand unified bootloader'의 약자
- 리눅스의 전통적인 부트 로더인 LILO의 단점을 보완하여 GNU 프로젝트의 일환으로 개발
- GRUB는 LILO에 비해 다음과 같은 장점을 가지고 있음
 - LILO는 리눅스에서만 사용이 가능하지만 GRUB는 윈도우에서도 사용할 수 있음
 - LILO에 비해 설정과 사용이 편리
 - 부팅할 때 명령을 사용하여 수정이 가능
 - 멀티 부팅 기능을 지원
- GRUB의 최신 버전은 GRUB2로 CentOS 8은 GRUB2를 기본 부트 로더로 사용

05.부트 로더

■ GRUB2 관련 디렉터리와 파일

■ /boot/grub2/grub.cfg 파일

- 기존의 menu.lst 파일을 대체하는 기본 설정 파일로 menu.lst 파일과 달리 사용자가 직접 수정할 수 없음
- 이 파일의 내용을 수정하려면 /etc/default/grub 파일과 /etc/grub.d 디렉터리 아래에 있는 스크립트를 수정

```
[root@localhost sbin]# more /boot/grub2/grub.cfg
#
# DO NOT EDIT THIS FILE
#
# It is automatically generated by grub2-mkconfig using templates
# from /etc/grub.d and settings from /etc/default/grub
#

### BEGIN /etc/grub.d/00_header ###
set pager=1

if [ -f ${config_directory}/grubenv ]; then
  load_env -f ${config_directory}/grubenv
elif [ -s $prefix/grubenv ]; then
  load_env
fi
if [ "${next_entry}" ] ; then
  set default="${next_entry}"
  set next_entry=
  save_env next_entry
  set boot_once=true
else
  set default="${saved_entry}"
fi
--More--(9%)
```

05.부트 로더

■ GRUB2 관련 디렉터리와 파일

■ /etc/grub.d 디렉터리

- GRUB 스크립트를 가지고 있으며, GRUB의 명령이 실행될 때 순서대로 실행되어 grub.cfg 파일이 생성

```
[root@localhost ~]# ls /etc/grub.d
00_header          01_users          20_ppc_terminfo   40_custom
00_tuned           10_linux          30_os-prober      41_custom
01_menu_auto_hide  20_linux_xen      30_uefi-firmware  README
```

■ /etc/default/grub 파일

- GRUB 메뉴 설정 내용이 저장되어 있으며, GRUB 스크립트가 이 파일을 읽어서 grub.cfg에 기록
- 기존의 menu.lst 파일과 비슷한 역할을 수행

```
[root@localhost ~]# cat /etc/default/grub
GRUB_TIMEOUT=5
GRUB_DISTRIBUTOR="$(sed 's, release .*$,,g' /etc/system-release)"
GRUB_DEFAULT=saved
GRUB_DISABLE_SUBMENU=true
GRUB_TERMINAL_OUTPUT="console"
GRUB_CMDLINE_LINUX="crashkernel=auto resume=/dev/mapper/cl-swap rd.lvm.lv=cl/root
rd.lvm.lv=cl/swap rhgb quiet"
GRUB_DISABLE_RECOVERY="true"
GRUB_ENABLE_BLSCFG=true
```


05.부트 로더

■ root 암호 복구하기

① 시스템 재시작하기

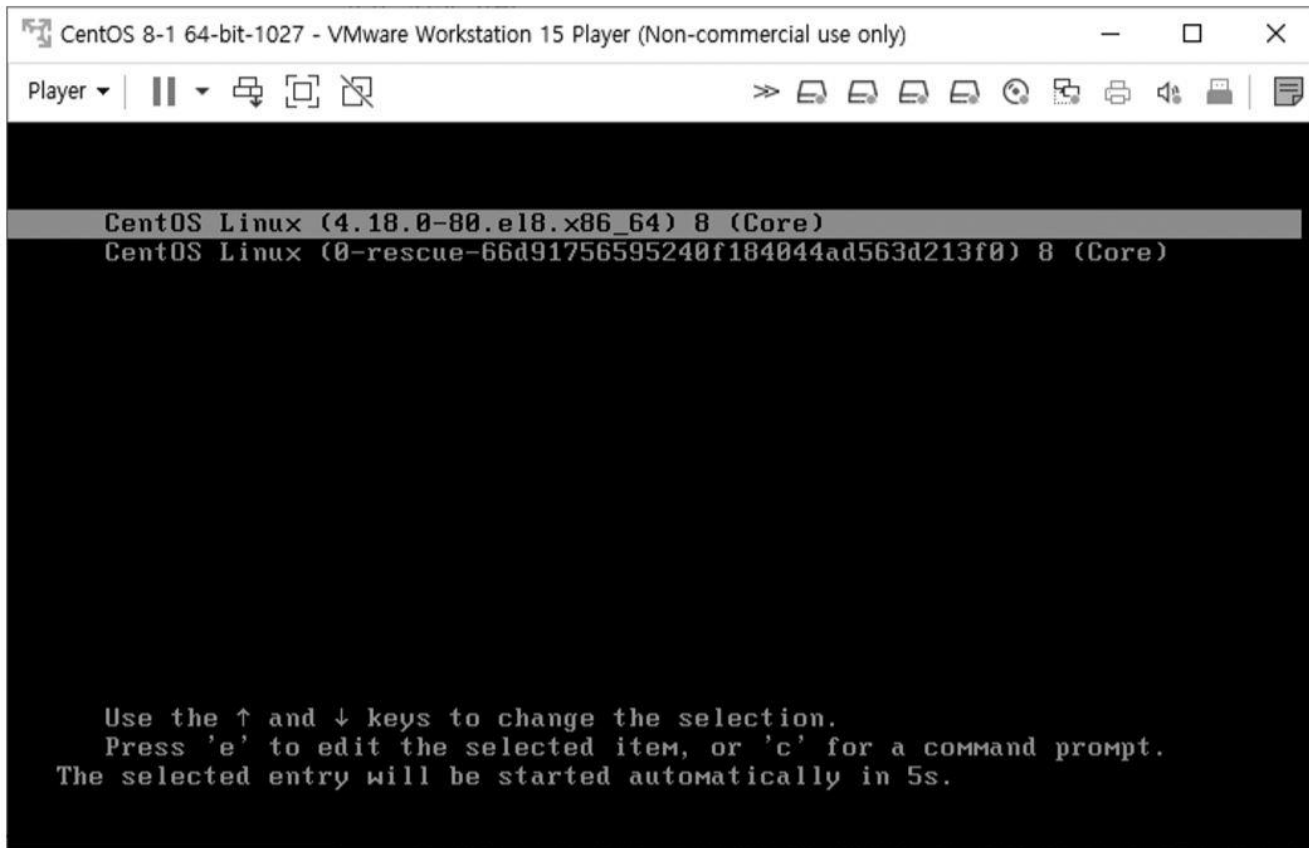


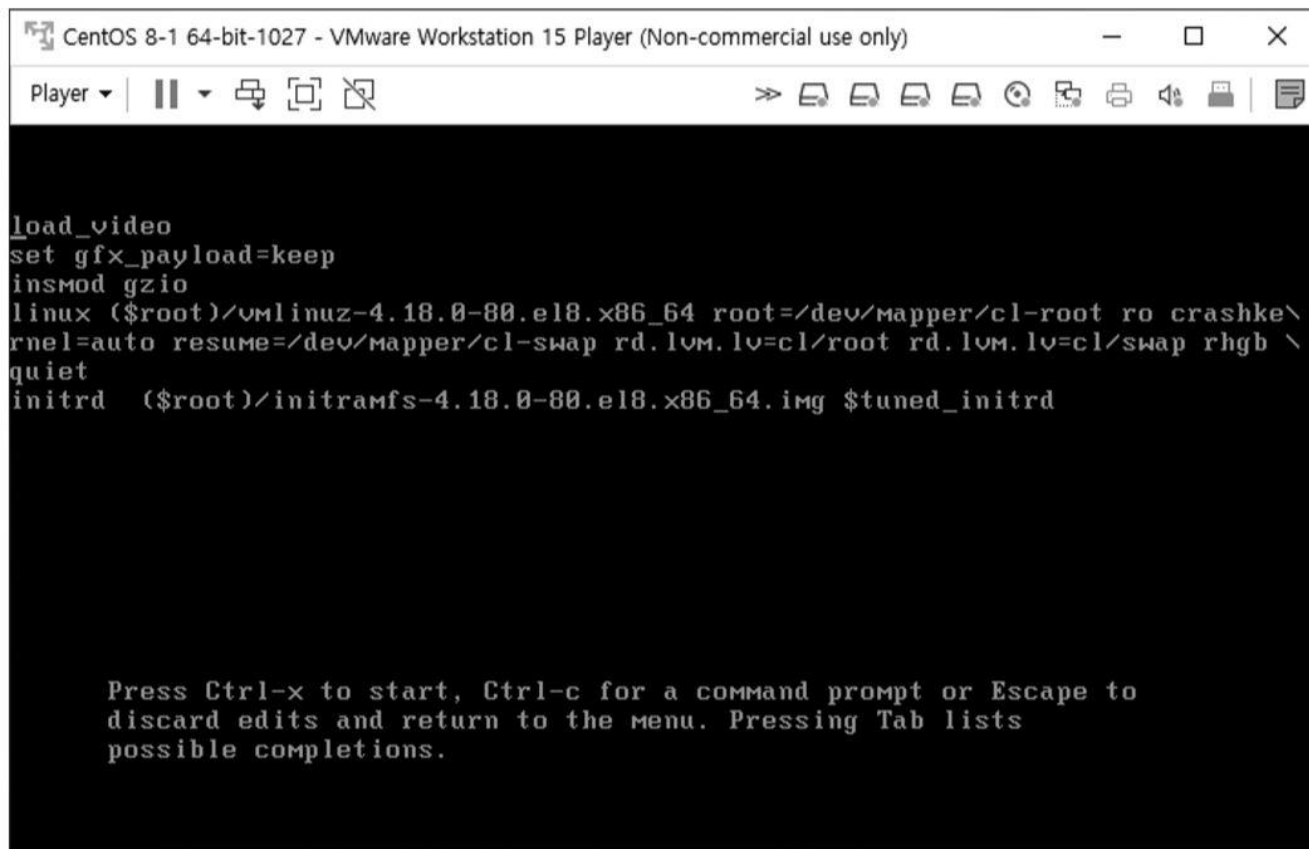
그림 8-7 GRUB 메뉴 초기 화면

05.부트 로더

■ root 암호 복구하기

② GRUB 편집 모드로 전환하기

- GRUB Boot Menu가 출력될 때 신속하게 'e' 키를 눌러서 편집 모드로 전환



```
CentOS 8-1 64-bit-1027 - VMware Workstation 15 Player (Non-commercial use only)
Player
load_video
set gfx_payload=keep
insmod gzio
linux ($root)/vmlinuz-4.18.0-80.el8.x86_64 root=/dev/mapper/cl-root ro crashke\
rnel=auto resume=/dev/mapper/cl-swap rd.lvm.lv=cl/root rd.lvm.lv=cl/swap rhgb \
quiet
initrd ($root)/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.
```

그림 8-8 GRUB 편집 화면

05.부트 로더

■ root 암호 복구하기

③ 단일 사용자 모드로 수정하기

- 리눅스 커널 정보가 있는 행에서 ro crach부터 quiet까지 삭제하고 rd.break enforcing=0를 추가

The screenshot shows a VMware Workstation window titled "CentOS 8-1 64-bit-1027 - VMware Workstation 15 Player (Non-commercial use only)". The interface includes a top toolbar with icons for player controls, a file explorer, and system settings. The main display area shows a black terminal window with white text representing a bootloader menu. The menu options are: load_video, set gfx_payload=keep, insmod gzio, linux (\$root)/vmlinuz-4.18.0-80.el8.x86_64 root=/dev/mapper/cl-root rd.break e_, enforcing=0_, and initrd (\$root)/initramfs-4.18.0-80.el8.x86_64.img \$tuned_initrd. At the bottom of the terminal window, there is instructional text: "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."

CentOS 8-1 64-bit-1027 - VMware Workstation 15 Player (Non-commercial use only)

Player ▾ | || ▾ | [] [X]

load_video
set gfx_payload=keep
insmod gzio
linux (\$root)/vmlinuz-4.18.0-80.el8.x86_64 root=/dev/mapper/cl-root rd.break e\
nforcing=0_
initrd (\$root)/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.

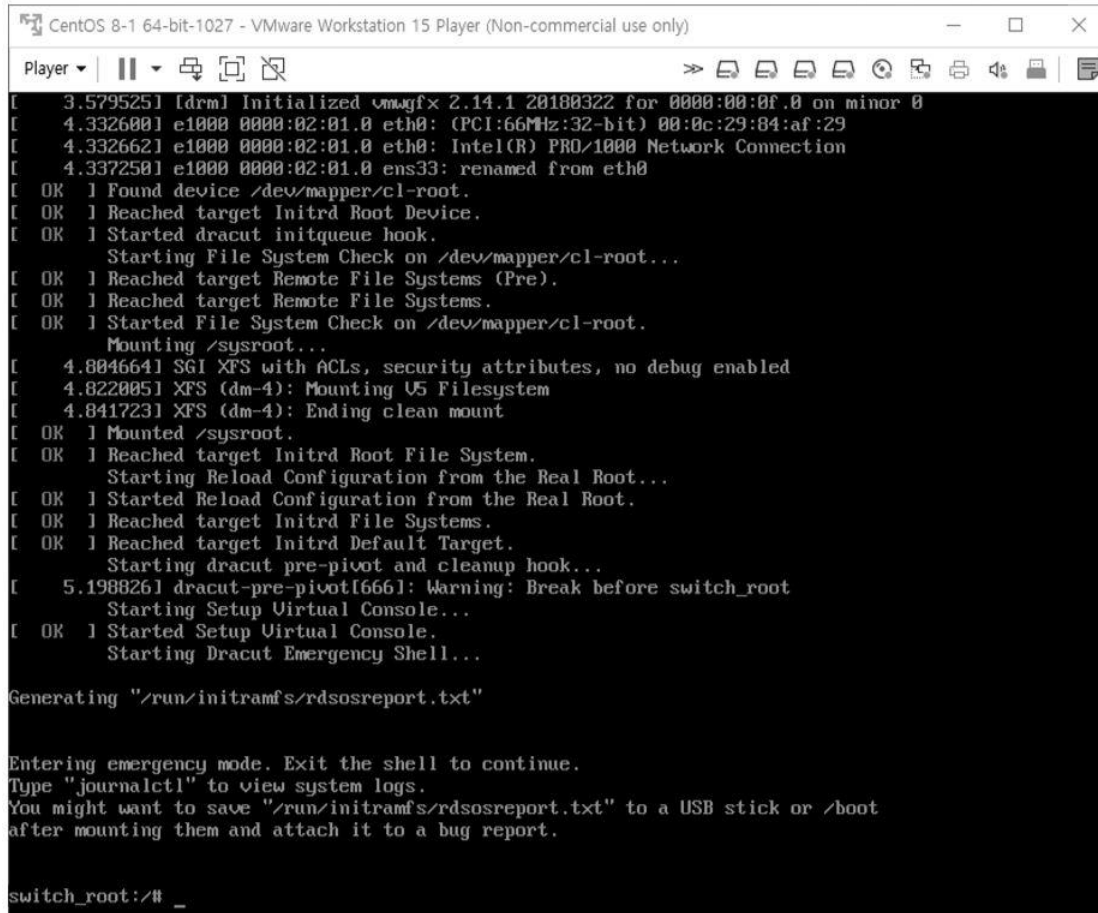
그림 8-9 root 암호 복구를 위한 리눅스 커널 항목 수정 화면

05.부트 로더

■ root 암호 복구하기

④ 재시작하기

- [Ctrl]+x를 눌러 재시작



```
CentOS 8-1 64-bit-1027 - VMware Workstation 15 Player (Non-commercial use only)
Player | [Icons]
[ 3.579525] [drm] Initialized vmwgfx 2.14.1 20180322 for 0000:00:0f:0 on minor 0
[ 4.332600] e1000 0000:02:01:0 eth0: (PCI:66MHz:32-bit) 00:0c:29:84:af:29
[ 4.332662] e1000 0000:02:01:0 eth0: Intel(R) PRO/1000 Network Connection
[ 4.337250] e1000 0000:02:01:0 ens33: renamed from eth0
[ OK ] Found device /dev/mapper/cl-root.
[ OK ] Reached target Initrd Root Device.
[ OK ] Started dracut initqueue hook.
Starting File System Check on /dev/mapper/cl-root...
[ OK ] Reached target Remote File Systems (Pre).
[ OK ] Reached target Remote File Systems.
[ OK ] Started File System Check on /dev/mapper/cl-root.
Mounting /sysroot...
[ 4.804664] SGI XFS with ACLs, security attributes, no debug enabled
[ 4.822005] XFS (dm-4): Mounting U5 Filesystem
[ 4.841723] XFS (dm-4): Ending clean mount
[ OK ] Mounted /sysroot.
[ OK ] Reached target Initrd Root File System.
Starting Reload Configuration from the Real Root...
[ OK ] Started Reload Configuration from the Real Root.
[ OK ] Reached target Initrd File Systems.
[ OK ] Reached target Initrd Default Target.
Starting dracut pre-pivot and cleanup hook...
[ 5.198826] dracut-pre-pivot[6661]: Warning: Break before switch_root
Starting Setup Virtual Console...
[ OK ] Started Setup Virtual Console.
Starting Dracut Emergency Shell...

Generating "/run/initramfs/rdsosreport.txt"

Entering emergency mode. Exit the shell to continue.
Type "journalctl" to view system logs.
You might want to save "/run/initramfs/rdsosreport.txt" to a USB stick or /boot
after mounting them and attach it to a bug report.

switch_root:/# _
```

그림 8-10 root 암호 복구를 위한 화면

05.부트 로더

■ root 암호 복구하기

④ 재시작하기

- 루트 파일 시스템이 읽기 전용으로 마운트되므로 읽기/쓰기 모드로 다시 마운트해야 함
- 시스템의 메시지가 기본적으로 한글로 출력되는데 콘솔 모드에서는 한글이 출력되지 않으므로 언어를 영어로 바꿔야 함
- passwd 명령으로 암호 변경 가능

```
switch_root:/# mount -o remount,rw /sysroot → /sysroot를 rw모드로 다시 마운트
switch_root:/# chroot /sysroot → /sysroot 디렉터리를 root 디렉터리로 설정
sh-4.4# LANG=en_US.UTF-8 → 언어 변경
sh-4.4# passwd → 암호 재설정
Changing password for user root.
New password:
Retype new password:
passwd: all authentication tokens updated succesfully.
sh-4.4# touch /.autorelabel → SELinux enable
sh-4.4# exit
exit
switch_root:/# exit
```

Thank you
