



Unit 3

Linux Environment & Basic Information



<http://cafe.daum.net/bseosolaris>

단원 목표

- 실습 환경 만들기
- TUI/GUI 환경 전환
- 로그인/로그아웃
- 언어변경
- 관리자 암호 변경 및 복구 방법
- 제어문자
- 운영체제 셧다운과 재부팅

<http://cafe.daum.net/bssc-solaris>

1. 배시셸과 로그인 & 로그아웃

배시셸 소개(BASH Shell Overview)

■ 배시셸(bash shell)

운영체제의 기본 셸로 GNU Bourne-Again Shell(bash) 이다. bash 셸은 UNIX 계열 시스템에 사용되는 가장 성공적인 셸 중 하나인 Bourne Shell(sh)의 향상된 버전이다.

배시(bash) 셸은 MS 윈도우(Windows)의 `cmd.exe`의 최신 버전에 포함된 **명령줄 인터프리터(Command Line Interpreter)**와 유사한 개념이다. bash 셸이 더욱 정교한 스크립트 언어를 갖고 있다. 또한 MS Windows 및 Windows Server의 파워셸(PowerShell)과도 유사한다. bash 셸은 MacOS에서도 기본적으로 사용된다.

■ 셸 프롬프트(bash shell prompt)

기본 사용자 셸 프롬프트 : `[user@host ~]$`

기본 관리자 셸 프롬프트 : `[root@host ~]#`

* user: 사용자 이름

* host: 서버 이름(축약)

* \$: 일반사용자, #: 관리자(root)

2. 로그인 & 로그아웃(Login & Logout)

■ 로컬 컴퓨터에 로그인 & 로그아웃

■ 네트워크를 통한 로그인 & 로그아웃

(1) 로컬 컴퓨터에 로그인 & 로그아웃

일반적으로 셸을 실행하려면 컴퓨터에 터미널(terminal)로 로그인 해야 한다. 터미널(terminal)은 컴퓨터 시스템에 명령을 입력하고 컴퓨터 시스템에서 출력을 인쇄하는 데 사용되는 텍스트 기반 인터페이스이다.



■ 물리콘솔(Physical Console) versus 가상콘솔(Virtual Console)

- 시스템을 조작하는데 사용되는 하드웨어 키보드와 디스플레이를 물리콘솔(Physical Console)이라고 한다. 물리콘솔은 별개의 터미널처럼 실행할 수 있는데 여러개의 가상콘솔(Virtual Console)을 지원한다. 각각의 가상 콘솔은 독립적인 로그인 세션을 지원한다.
- 가상 콘솔은 일반적으로 <CTRL + ALT + F1> ~ <CTRL + ALT + F6>등의 키 조합으로 접근 가능한 텍스트 기반 로그인 세션이다. GUI 환경이 있는 경우 <CTRL + ALT + F7> 이상은 그래픽 콘솔(X11/Wayland)이 배정된다. 이러한 가상 콘솔은 텍스트 로그인 프롬프트를 제공하는 터미널을 실행하며, 사용자 이름과 암호를 입력하면 로그인이 되고 쉘 프롬프트가 표시된다.

■ 터미널(Terminal) & 터미널 에뮬레이터(Terminal Emulator)

- 터미널은 입력/출력 장치를 시뮬레이션 시켜 놓은 프로그램이다.
- 대표적인 터미널 에뮬레이터 프로그램 종류: putty, SecreCRT, Xshell, MobaXterm 등이 있다.

[참고] CentOS 5, 6, 7, 8 가상콘솔 비교

CentOS 5

<CTRL + ALT + F1~F6> : 텍스트 가상콘솔
<CTRL + ALT + F7> : GUI 로그인 창

CentOS 6, 7

<CTRL + ALT + F1> : GUI 로그인 창
<CTRL + ALT + F2~F6> : 텍스트 가상콘솔

CentOS 8

<CTRL + ALT + F1> : 로그인 세션에 현재 사용되지 않는 첫번째 가상콘솔(tty1)
<CTRL + ALT + F2~F6> : 텍스트 가상콘솔
<CTRL + ALT + F7~F12> : 일반적인 로그인 세션에 사용되는 두번째 가상콘솔(tty2)

(2) 네트워크를 통한 로그인/로그아웃

일반적으로 관리자가 네트워크를 통해 연결하여 원격 시스템에 대한 쉘 액세스 권한을 얻어야 하는 경우 SSH(보안 쉘) 사용합니다. SSH 외에도 telnet, rlogin 등도 있습니다. 하지만 보안상 SSH 사용할 것을 권장합니다.

■ 일반적인 SSH 로그인

```
[user@host ~]$ ssh ruser@rhost
ruser@rhost's password: (암호입력)
[ruser@rhost ~]$
```

공개키 인증을 사용한 SSH 로그인

```
[user@host ~]$ ssh -i id_rsa ruser@rhost
[ruser@rhost ~]$
```

* ssh -i id_rsa ec2-user@30.23.33.23

■ 로그아웃

쉘 사용을 마치고 종료하는 경우 exit 명령어를 사용하여 현재 쉘 세션을 종료한다. 또는 <CTRL + D> 사용할 수 도 있다.

exit 명령어를 사용하여 SSH 세션 종료하기

```
[ruser@rhost ~]$ exit
logout
Connection to rhost closed.
[user@host ~]$
```

[실습] 로그인/로그 아웃

■ 사용시스템

- server1
- server2

■ 실습 목록

- 로컬 컴퓨터에 로그인/로그아웃
- 네트워크를 통한 로그인/로그아웃

[EX] 로컬 컴퓨터에 로그인/로그아웃

■ 실습 시나리오 - 물리콘솔(Physical Console)과 가상콘솔(Virtual Console) 실습

- 가상 콘솔을 사용하여 GUI 로그인 창과 로그인된 화면, 텍스트 로그인 화면등을 테스트 해 본다.

(전제조건) 서버가 재부팅 된 이후에 로그인 창이 떠 있는 상태이다.



① 다음과 같은 키를 입력 해 본다.

<CTRL + ALT + F1> 입력해 본다.

- 변화가 있는가?

<CTRL + ALT + F2> ~ <CTRL + ALT + F6> 입력해 본다.

- 변화가 있는가?
- 필요하다면, fedora 사용자로 로그인 해 본다.(CTRL + ALT + F2>)

<CTRL + ALT + F7> ~ <CTRL + ALT + F12> 입력해 본다.

- 변화가 있는가?

② GUI 로그인 창에서 root 사용자로 로그인하고 다음과 같은 키를 입력 해 본다.

<CTRL + ALT + F1> 입력하고 root 사용자로 로그인

- **목록에 없습니다?** 선택하고,
- **사용자이름**은 **root**, **암호**는 **centos** 입력한다.
- 터미널을 띄워서 id 명령어를 수행해 놓는다.

<CTRL + ALT + F1> ~ <CTRL + ALT + F6>

- 변화가 있는가?

<CTRL + ALT + F7> ~ <CTRL + ALT + F12>

- 변화가 있는가?

③ GUI 로그인 창에서 fedora 사용자로 로그인하고 다음과 같은 키를 입력 해 본다.

<CTRL + ALT + F1> 입력 후 fedora 사용자로 로그인

- **fedora** 사용자 아이콘을 선택하고
- 암호로 **fedora**를 입력한다.
- 터미널을 띄워서 **id** 명령어를 수행해 본다.

<CTRL + ALT + F1> ~ <CTRL + ALT + F6>

- 변화가 있는가?

<CTRL + ALT + F7> ~ <CTRL + ALT + F12>

- 변화가 있는가?

④ 서버 재부팅

- 서버를 재부팅한다.

```
# reboot
```

[EX] 네트워크를 통한 로그인/로그아웃

■ 실습시스템

- server1(**SSH 서버**)
- server2(**SSH 클라이언트**)

■ 실습 시나리오

- SSH 사용하여 원격 로그인/로그아웃을 테스트 해 본다.

① (**server2**) ssh 명령어를 이용한 원격 서버 로그인 시도

[참고] SSH 명령어 형식:

```
# ssh <사용자>@<호스트> (예) # ssh root@192.168.10.20
# ssh -i <사용자> <호스트> (예) # ssh -i root 192.168.10.20
```

```
[root@server2 ~]# ssh root@192.168.10.20
```

```
The authenticity of host '192.168.10.20 (192.168.10.20)' can't be established.
ECDSA key fingerprint is SHA256:iY95UhrNcQCum8goQiuYkKp4NDjekFs9Xd056nnHjBw.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.10.20' (ECDSA) to the list of known hosts.
root@192.168.10.20's password: centos
Activate the web console with: systemctl enable --now cockpit.socket

Last login: Wed Jan 27 14:35:35 2021
```

② (server2) 로그인 후 정보 확인

```
[root@server1 ~]# hostname
```

```
server1.example.com
```

```
[root@server1 ~]# id
```

```
uid=0(root) gid=0(root) groups=0(root)
```

```
[root@server1 ~]# pwd
```

```
/root
```

③ (server2) 로그 아웃

```
[root@server1 ~]# exit
```

```
[root@server2 ~]#
```

3. 데스크탑(Desktop) 사용

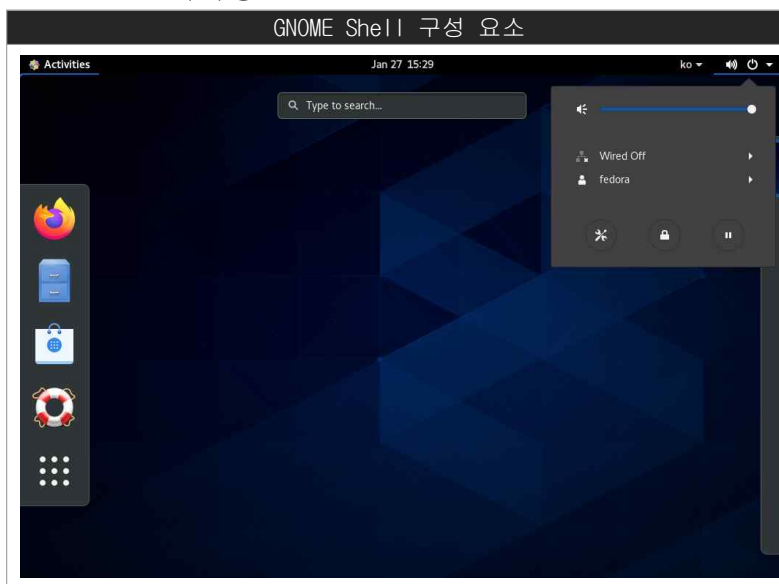
- * GNOME : GTK+ 라이브러리
- * KDE : QT 라이브러리

■ GNOME 3 데스크탑 환경 소개

데스크탑 환경은 리눅스 시스템의 그래픽 사용자 인터페이스이다. CentOS 8.X는 기본 데스크탑 환경으로 GNOME 3 제공한다. 사용자를 위한 통합 데스크탑과 Wayland(기본값) 또는 레거시 X Window System에서 제공하는 그래픽 프레임워크를 기반으로 하는 통합 개발 플랫폼을 제공한다.

GNOME Shell은 GNOME 데스크탑 환경을 위한 핵심 사용자 인터페이스 기능을 제공한다. GNOME Shell 애플리케이션은 사용자 지정 정의가 가능하다.

■ GNOME Shell의 구성 요소



4. 실습 환경 만들기

실습하기 좋은 환경으로 만들어 보자.

(주의) 일부 보안상 취약한 설정들도 존재하므로 실무에 적용시 주의해야 한다.

[실습] GUI 환경 설정

■ 사용시스템

- main.example.com
- server1.example.com
- server2.example.com

■ 실습 목록

- 화면 해상도 조정
- 화면 잠금 기능 끄기/절전 기능 끄기
- 절전 기능 끄기
- 터미널 폰트 조정
- 폰트 조정
- 바탕화면 아이콘 생성
- 방화벽 끄기
- SELinux 끄기

■ 화면 해상도 조정

- **[전원]** 아이콘 > **[설정]** 아이콘 > **[디스플레이]** > **[해상도]**
 - 적당한 해상도 선택: 1024x768

■ 화면 잠금 기능 끄기/절전 기능 끄기

- **[전원]** 아이콘 > **[설정]** 아이콘 > **[개인 정보]** > **[화면 잠금]**
 - 빈 화면 지연시간: 안함
 - 자동 화면 잠금: 끄
- **[전원]** 아이콘
 - Power Mode: **[v]** Performance
 - Power Saving Options:
 - * Screen Blank: **Naver**, Automatic Suspend: **Off**

```
# 자동 잠금 끄기(Authmatic Screen Lock)
gsettings set org.gnome.desktop.screensaver lock-enabled false
# 잠금 시간(idle delay) 설정
gsettings set org.gnome.desktop.session idle-delay 0
# 전원 연결 시 절전 끄기
gsettings set org.gnome.settings-daemon.plugins.power sleep-inactive-ac-type 'nothing'
# 배터리 사용 시 절전 끄기
gsettings set org.gnome.settings-daemon.plugins.power sleep-inactive-battery-type 'nothing'
```

■ gnome-tweaks 설치 및 설정

- **(주의)** 네트워크 통신(인터넷)이 가능해야 한다.

```
# yum -y install gnome-tweaks
# gnome-tweaks
```

■ 터미널 폰트 조정

- **(주의)** gnome-tweaks 설치된 이후에 작업해야 한다.
- (임시) 터미널에서 <CTRL + SHIFT + "+>, <CTRL + "->, <CTRL + O>
- (영구) gnome-tweaks > 글꼴(fonts) > 추천글꼴: Monospace

```
# 패키지 설치
yum -y install gnome-tweaks
# 폰트 조정
gsettings set org.gnome.desktop.interface monospace-font-name 'Monospace Bold 18'
```

■ GNOME Shell Extensions 활성화

```
# gnome-extensions list
```

```
# gnome-extensions enable <UUID> /* UUID: Universally Unique Identifier */
```

-> 모든 기능을 활성화(enable) 해서 사용한다.

다음 내용을 복사해서 사용할 수 있다.

```
# GNOME Shell Extension 모든 기능 활성화 2
for i in $(gnome-extensions list); do gnome-extensions enable ${i}; done
```

■ 바탕화면에 아이콘 생성

- /usr/share/applications/* (많은 아이콘이 존재)
(주의) 다른 위치로 복사(ex: 홈/바탕화면)
- 아이콘은 다음과 같은 것을 바탕화면에 복사한다.
이 름: 터미널(gnome-terminal)
이 름: 텍스트 편집기(gedit)

다음 내용을 복사해서 사용할 수 있다.

```
# 바탕화면 아이콘 복사
/bin/cp /usr/share/applications/org.gnome.{Terminal.desktop,gedit.desktop} ~/바탕화면
```

■ 방화벽 끄기

방화벽(Firewall)? 윈도우 방화벽

(부팅) # systemctl disable firewalld

(현재) # systemctl stop firewalld

다음 내용을 복사해서 사용할 수 있다.

```
# 방화벽 끄기
systemctl disable --now firewalld
```

■ SELinux(Secure Enhanced Linux) 끄기

SELinux? **프로세스**가 사용하는 **파일/디렉토리/포트**에 보안 레이블(Security Label)을 설정하여 접근 제어하는 기능이다. 이 기능을 사용하여 사용자와 프로세스에서 파일 및 장치 등을 무단 상호 작용을 하지 못하도록 할 수 있다.

1) (enforcing -> disabled)

```
# gedit /etc/selinux/config
```

```
[수정전]
SELINUX=enforcing
[수정후]
SELINUX=disabled
```

```
# reboot
```

다음 내용을 복사해서 사용할 수도 있다.

```
sed -i 's/^SELINUX=enforcing/SELINUX=disabled/' /etc/selinux/config
reboot
```

2) (enforcing -> permissive)

```
# setenforce 0
```

```
# gedit /etc/selinux/config
```

```
[수정전]
SELINUX=enforcing
[수정후]
SELINUX=permissive
```

다음 내용을 복사해서 사용할 수 있다.

```
# SELinux 끄기
sed -i 's/^SELINUX=enforcing/SELINUX=permissive/' /etc/selinux/config
setenforce 0
```

[참고] 환경 작업 한번에 수행하기

다음 내용을 복사하여 실행한다.

다음은 각 서버에 맞게 실행한다.

* **(주의)** 반드시 VMware의 VMnet8 스위치의 대역이 192.168.10.0/24로 되어 있어야 한다.

```
# (main)
# ■ connection 이름 설정: eth0
NICNAME=$(nmcli device | tail -n +2 | grep -v '^lo' | awk '{print $1}')
OLDCON=$(nmcli device show "$NICNAME" | grep GENERAL.CONNECTION: | awk '{print $2}')
nmcli connection modify "$OLDCON" connection.id eth0
nmcli connection up eth0
```

```
# ■ 호스트 이름 설정 & IP 설정
hostnamectl set-hostname main.example.com
nmcli connection modify eth0 connection.autoconnect yes ipv4.method manual W
    ipv4.addresses 192.168.10.10/24 ipv4.gateway 192.168.10.2 ipv4.dns 8.8.8.8 W
    ipv4.dns-search example.com
nmcli connection up eth0
```

```
# (server1)
# ■ connection 이름 설정: eth0
NICNAME=$(nmcli device | tail -n +2 | grep -v '^lo' | awk '{print $1}')
OLDCON=$(nmcli device show "$NICNAME" | grep GENERAL.CONNECTION: | awk '{print $2}')
nmcli connection modify "$OLDCON" connection.id eth0
nmcli connection up eth0
```

```
# ■ 호스트 이름 설정 & IP 설정
hostnamectl set-hostname server1.example.com
nmcli connection modify eth0 connection.autoconnect yes ipv4.method manual W
    ipv4.addresses 192.168.10.20/24 ipv4.gateway 192.168.10.2 ipv4.dns 8.8.8.8 W
    ipv4.dns-search example.com
nmcli connection up eth0
```

```
# (server2)
# ■ connection 이름 설정: eth0
NICNAME=$(nmcli device | tail -n +2 | grep -v '^lo' | awk '{print $1}')
OLDCON=$(nmcli device show "$NICNAME" | grep GENERAL.CONNECTION: | awk '{print $2}')
nmcli connection modify "$OLDCON" connection.id eth0
nmcli connection up eth0
```

```
# ■ 호스트 이름 설정 & IP 설정
hostnamectl set-hostname server2.example.com
nmcli connection modify eth0 connection.autoconnect yes ipv4.method manual W
    ipv4.addresses 192.168.10.30/24 ipv4.gateway 192.168.10.2 ipv4.dns 8.8.8.8 W
    ipv4.dns-search example.com
nmcli connection up eth0
```

* 다음을 모든 서버에서 실행 한다. (실행 후 자동으로 재부팅된다.)

```
# (main/server1/server2)
# ■ 화면 잠금 기능 끄기/절전 기능 끄기
# 자동 잠금 끄기(Authmatic Screen Lock)
gsettings set org.gnome.desktop.screensaver lock-enabled false
# 잠금 시간(Idle delay) 설정
gsettings set org.gnome.desktop.session idle-delay 0
# 전원 연결 시 절전 끄기
gsettings set org.gnome.settings-daemon.plugins.power sleep-inactive-ac-type 'nothing'
# 배터리 사용 시 절전 끄기
gsettings set org.gnome.settings-daemon.plugins.power sleep-inactive-battery-type 'nothing'
```

```

# ■ gnome-tweaks 설치 및 설정
# (주의) 반드시 인터넷이 되어야 한다.
# 패키지 설치
yum -y install gnome-tweaks
# 폰트 조정(Monospace 18)
gsettings set org.gnome.desktop.interface monospace-font-name 'Monospace Bold 18'

# ■ GNOME Shell Extensions 활성화
for i in $(gnome-extensions list); do gnome-extensions enable ${i}; done

# ■ 바탕화면에 아이콘 생성
# (주의) 한국어로 되어 있는 경우에는 "바탕화면" 디렉토리에 만든다.
if [ -d ~/바탕화면 ] ; then
    /bin/cp /usr/share/applications/org.gnome.{Terminal.desktop,gedit.desktop} ~/바탕화면
elif [ -d ~/Desktop ] ; then
    /bin/cp /usr/share/applications/org.gnome.{Terminal.desktop,gedit.desktop} ~/Desktop
fi

# ■ 방화벽 끄기
systemctl disable --now firewallld

# ■ SELinux(Secure Enhanced Linux) 끄기
sed -i 's/^SELINUX=enforcing/SELINUX=disabled/' /etc/selinux/config
reboot

```

[참고] 그놈 셸 확장(GNOME Shell extensions)

[참고] Chapter 15. Managing GNOME Shell extensions via Command Line
https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/9/html/customizing_the_gnome_desktop_environment/managing-gnome-shell-extensions-via-command-line_customizing-the-gnome-desktop-environment

[참고] Chapter 16. Enabling and enforcing GNOME Shell extensions
https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/9/html/customizing_the_gnome_desktop_environment/enabling-and-enforcing-gnome-shell-extensions_customizing-the-gnome-desktop-environment

To list the installed GNOME Shell extensions, use:

```
$ gnome-extensions list
```

To install a GNOME Shell extension, use:

```
$ gnome-extensions install <UUID> /* UUID: Universally Unique Identifier */
```

To enable a GNOME Shell extension, use:

```
$ gnome-extensions enable <UUID>
```

To show information about a GNOME Shell extension, use:

```
$ gnome-extensions info <UUID>
```

To disable a GNOME Shell extension, use:

```
$ gnome-extensions disable <UUID>
```

To remove a GNOME Shell extension, use:

```
$ gnome-extensions uninstall <UUID>
```

5. TUI/GUI 환경 전환

■ 용어(Terms)

- TUI(Text User Interface), CLI(Command Line Interface)
- GUI(Graphic User Interface)

■ 동작수준/동작레벨(Runlevel)

- 런레벨(Runlevel)? 시스템 관리를 위해 서비스의 실행을 단계별로 구분하여 적용하는 것을 말한다.
- 시스템이 특별한 상태로 초기화되어 있는 상태를 런레벨, 이런 상태를 숫자에 매핑 시켜 놓았다.

[참고] CentOS 6 이하 versus CentOS 7 이상
 (CentOS 6 이하) 런레벨(Runlevel)? 동작 수준/동작 레벨
 * Runlevel 3 : TUI
 * Runlevel 5 : GUI
 (CentOS 7 이상) 대상(Target)? 동작 수준/동작 레벨
 * multi-user.target: TUI
 * graphical.target : GUI

- 대상(Target)? 관련된 서비스 UNIT의 집합(.service, .socket 등)이다.

```
# systemctl -t help
# systemctl -t service
```

(TUI) Runlevel 3 == multi-user.target
 (GUI) Runlevel 5 == graphical.target

■ 대상(Target) 확인

```
(현재) # who -r
(부팅) # systemctl get-default
```

■ 대상(Target) 전환 방법

```
(현재) # systemctl isolate multi-user.target|graphical.target
(부팅) # systemctl set-default multi-user.target|graphical.target
```

[실습] TUI/GUI 환경 전환

■ 사용시스템

- server1

■ 실습 시나리오

- TUI <=> GUI 환경 전환에 대한 실습을 진행한다.

① (server1) 부팅시 적용될 target 확인

```
[root@server1 ~]# systemctl get-default
```

```
graphical.target
```

- 출력내용: 부팅시에 GUI 모드로 부팅한다.

```
(현재) # who -r
(부팅) # systemctl get-default
```

② (server1) 현재 target을 multi-user.target 변경

```
[root@server1 ~]# systemctl isolate multi-user.target
```

③ (server1) 로그인 창에서 root 사용자로 로그인한다.

```
server1 login: root
Password: (centos)
Last login: Sat Feb 6 03:48:18 on tty2
```

```
[root@server1 ~]#
```

```
(현재) # who -r
(부팅) # systemctl get-default
```

④ (server1) 현재는 어떤 상태인가요?

```
[root@server1 ~]# systemctl get-default
```

```
graphical.target
```

```
(현재) # who -r
(부팅) # systemctl get-default
```

⑤ (server1) 가상 콘솔 확인

가상 콘솔을 확인하기 위해서 다음과 같은 키조합을 사용하여 본다.

```
* <CTRL + ALT + F1>
* <CTRL + ALT + F2>
* <CTRL + ALT + F3>
* <CTRL + ALT + F4>
* <CTRL + ALT + F5>
* <CTRL + ALT + F6>
```

```
* <CTRL + ALT + F1>
```

⑥ (server1) 다음번째 부팅시 target 변경

```
[root@server1 ~]# 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@server1 ~]# systemctl get-default
```

```
multi-user.target
```

```
(현재) # who -r
(부팅) # systemctl get-default
```

⑦ (server1) 재부팅

```
[root@server1 ~]# reboot
```

⑧ (server1) 부팅이 된 이후에 root 사용자로 로그인

```
CentOS Linux 8
Kernel 4.18.0-240.el8.x86_64 on an x86_64

Activate the web console with: systemctl enable --now cockpit.socket

server1 login: root
Password: (centos)
Last login: Sat Feb 6 10:47:04 on tty1
```

```
[root@server1 ~]#
```

⑨ (server1) 현재 target을 graphical.target으로 변경

```
[root@server1 ~]# systemctl isolate graphical.target
```

⑩ (server1) root 사용자로 로그인

"목록에 없습니까?" 선택하고

(↵)사용자 이름으로 **root** 입력하고,
(↵)암호는 **root 사용자 암호(ex: centos)**를
입력한다.

⑪ (server1) 터미널을 띄우고, 현재 default target 확인 및 변경

```
[root@server1 ~]# systemctl get-default
```

```
multi-user.target
```

```
[root@server1 ~]# systemctl set-default graphical.target
```

```
Removed /etc/systemd/system/default.target.
```

```
Created symlink /etc/systemd/system/default.target → /usr/lib/systemd/system/graphical.target.
```

⑫ 결론

■ 환경 변경 방법(예: **GUI → TUI**)

(부팅) # systemctl set-default multi-user.target

(현재) # systemctl isolate multi-user.target

[추가적인 실습]

■ 다음과 같이 서버들을 적당한 GUI/TUI 환경으로 설정한다.(main, server1, server2 설정)

- (TUI) **main**.example.com → multi-user.target (runlevel 3) : Text Mode
- (GUI) **server1**.example.com → graphical.target (runlevel 5) : GUI Mode
- (TUI) **server2**.example.com → multi-user.target (runlevel 3) : Text Mode

```
# (main)
systemctl set-default multi-user.target
systemctl isolate multi-user.target

# (server1)
systemctl set-default graphical.target
systemctl isolate graphical.target

# (server2)
systemctl set-default multi-user.target
systemctl isolate multi-user.target
```

[추가적인 실습]

■ 언어 변경 방법

LANG 변수? 언어 지정 변수

- ko_KR.UTF-8 : 한국어 ⇒ export LANG=ko_KR.UTF-8
- en_US.UTF-8 : 영어 ⇒ export LANG=en_US.UTF-8

임시적인 방법	영구적인 방법
<pre># echo \$LANG # export LANG=ko_KR.UTF-8 # gnome-tweaks # export LANG=en_US.UTF-8 # gnome-tweaks</pre>	<pre># localectl # localectl set-locales LANG=en_US.UTF-8 # localectl # localectl set-locales LANG=ko_KR.UTF-8 => 적용하기 위해서 GUI 환경에서 [로그아웃/로그인] => !!! (주의) 스펠링 주의 !!!</pre>
<pre># (main/server1/server2) localectl set-locales LANG=en_US.UTF-8 echo ; echo "===== 언어 변경 =====" localectl</pre>	

6. 제어문자(CTRL + ?, Control Character)

컴퓨팅 및 전자통신 환경에서 **제어 문자(control character)**는 **문자 집합의 특별한 문자**이다. 자세한 내용은 다음은 참고한다. <https://ko.wikipedia.org/wiki/제어문자>

이 중에서 다음과 같은 몇가지 제어문자(Control Character)에 대해서 배워보자.

<CTRL + C>, <CTRL + D>

■ **<CTRL + C>** : 프로그램 실행 중간에 인터럽트(interrupt) 걸어서 종료 하는 역할

```
[root@server1 ~]# sleep 600
<CTRL + C>
```

```
[root@server1 ~]# ping 8.8.8.8
<CTRL + C>
```

```
[root@server1 ~]# top
<CTRL + C>
```

(주의) "WIN <CTRL + C>" vs "LINUX <CTRL + C>"

- "WIN <CTRL + C>" : 내용을 복사하는 단축키
- "LINUX <CTRL + C>" : 프로그램 실행 중간에 인터럽트 종료하는 역할

■ **<CTRL + D>** : (↵) 파일의 끝(EOF), (␣) 현재 셸(ex: bash) 종료

(↵) 파일의 끝(EOF, End of File)

```
[root@server1 ~]# mkdir /test
[root@server1 test]# cd /test
[root@server1 test]# cat > file1.txt
hello world
<CTRL + D>
[root@server1 test]# cat file1.txt
```

(␣) 현재 셸 종료

```
[root@server1 ~]# ps
[root@server1 ~]# bash
<CTRL + D>
<CTRL + D>
```

```
[root@server1 ~]# ssh 192.168.10.30
root 사용자로 로그인
[root@server2 ~]# ps
<CTRL + D>
```

<CTRL + W>, <CTRL + U>

■ **<CTRL + W>** : 마지막 단어(word) 삭제

```
[root@server1 ~]# find / -name core -type f <CTRL + W> <CTRL + W>
```

■ **<CTRL + U>** : 라인 전체 삭제(undo)

```
[root@server1 ~]# find / -name core -type f <CTRL + U>
```

<CTRL + S>, <CTRL + Q>

■ <CTRL + S> <CTRL + Q> : S(Stop) 화면 멈춤, Q(Quit) 빠져나가기

[root@server1 ~]# du -a /

<CTRL + S>	# 화면 멈춤
<CTRL + Q>	# 다시 실행
<CTRL + S>	# 화면 멈춤
<CTRL + Q>	# 다시 실행
<CTRL + C>	# 종료

7. 운영체제 셧다운과 재부팅

운영체제를 셧다운하거나 재부팅하는 방법은 여러가지가 있으며, 아래와 같은 방법으로 수행이 가능하다. (다양한 방법이 더 존재한다.)

■ 운영체제 전원 끄기(Power OFF)

```
# poweroff
# init 0          /* 0: Runlevel 0 = Power Off */
# shutdown -h now /* -h: halt */
```

■ 운영체제 재부팅(Reboot)

```
# reboot
# init 6          /* 6: Runlevel 6 = reboot */
# shutdown -r now /* -r: reboot */
```