

제1장

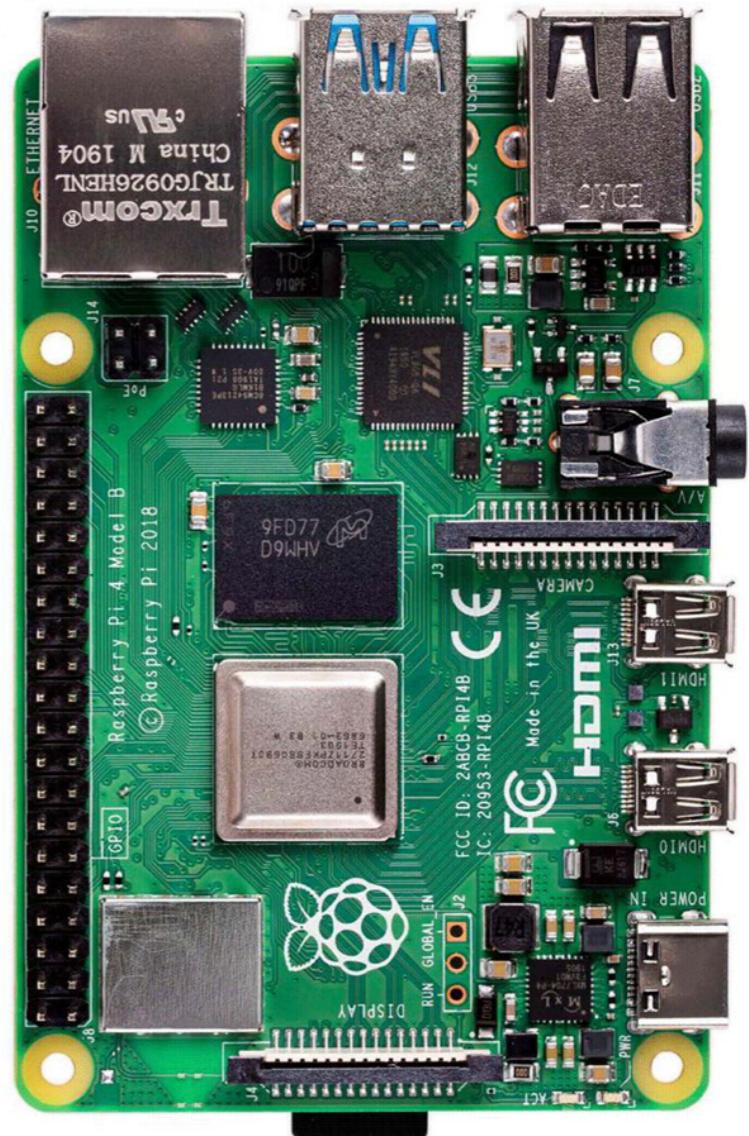
Raspberry Pi, Linux, and Python

Raspberry Pi

Raspberry Pi

작고, 저렴한, 오픈 소스 Linux 시스템

- 강력하고
 - 저렴하고 (\$45-\$55)
 - 외부 장치와의 다양한 표준 인터페이스를 지원하고 (GPIO, I²C, SPI, UART)
 - 저전력의
 - 확장이 용이하고
 - 다양한 커뮤니티 지원을 받을 수 있는
 - 오픈 하드웨어, 오픈 소프트웨어 시스템



When Raspberry Pi ?

- ⦿ 고 수준의 소프트웨어와 저 수준 장치의 통합이 필요한 응용

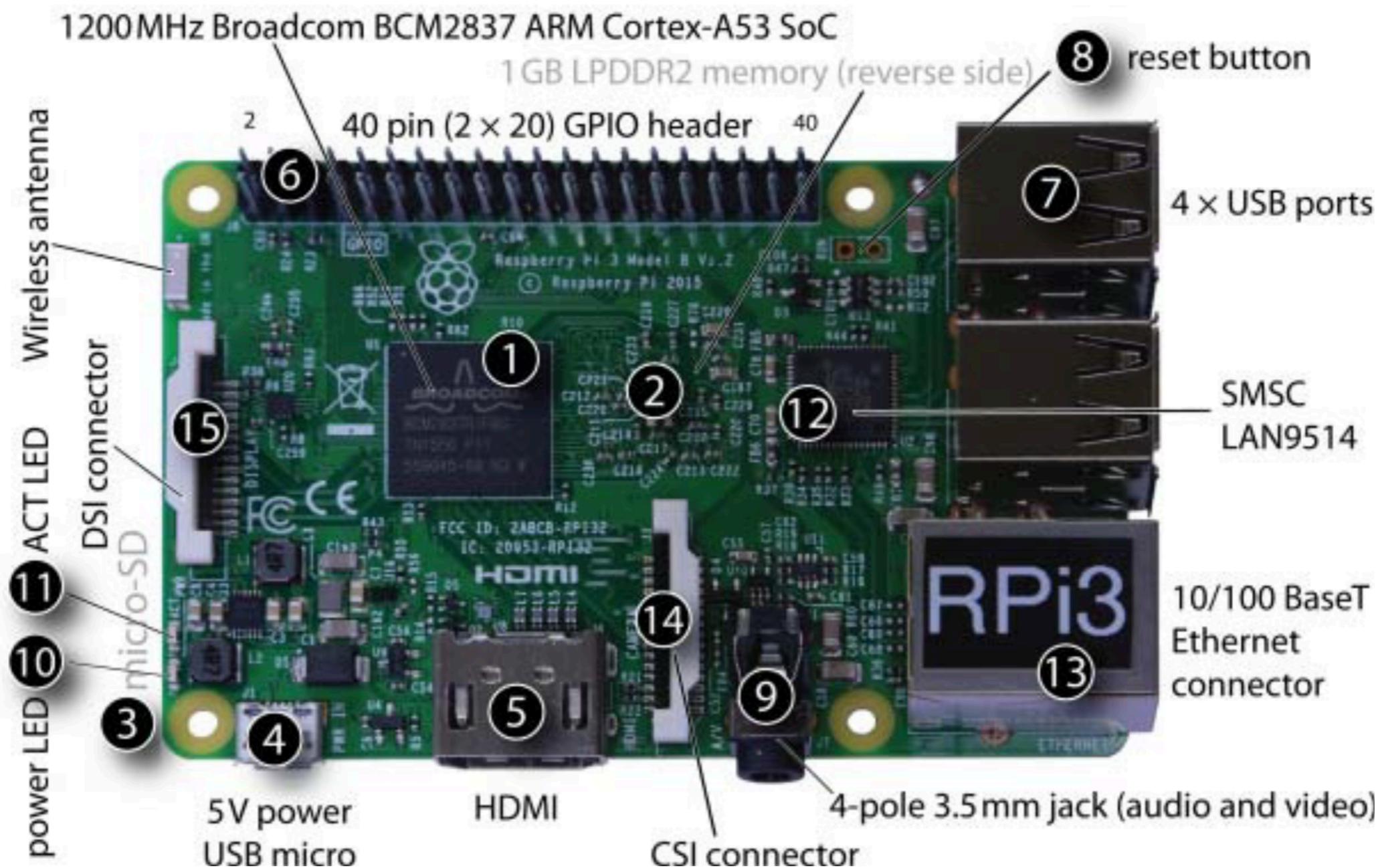
- ⦿ 홈 오토메이션 시스템
- ⦿ 로보틱스 (robotics)
- ⦿ 센서 네트워크 (sensor network)

- ⦿ Linux OS가 제공하는 강력한/다양한 기능이 필요할 때

- ⦿ Apache web server, secure shell (SSH)
- ⦿ 다양한 USB 주변장치 (cameras, Wi-Fi adapters, 등)에 대한 디바이스 드라이버 지원

Family	Model	Form Factor	Ethernet	Wireless	GPIO	Released	Discontinued	
Raspberry Pi 1	B	Standard (85.60 × 56.5 mm)	Yes	No	26-pin	2012	Yes	
	A		No			2013	Yes	
	B+		Yes			2014		
	A+	Compact (65 × 56.5 mm)	No			2014		
Raspberry Pi 2	B	Standard	Yes	No	40-pin	2015		
Raspberry Pi Zero	Zero	Zero (65 × 30 mm)	No	No		2015		
	W/WH			Yes		2017		
Raspberry Pi 3	B	Standard	Yes	Yes		2016		
	A+	Compact	No			2018		
	B+	Standard	Yes			2018		
Raspberry Pi 4	B (1GB)	Standard	Yes	Yes	2019 ^[26]	2019 ^[26]		
	B (2GB)							
	B (4GB)							

Raspberry Pi 3



Raspberry Pi 3

	Function	Physical	Details
1	Processor	BCM283x (CPU)	The RPi boards use the Broadcom BCM2835/BCM2836/BCM2837 processor. The different boards use slightly different processors that run between 700 MHz and 1.2 GHz and are based on ARMv6, ARMv7, ARMv11, and ARMv8 A53 processor cores.
2	Memory	Graphics Engine (GPU) 256 MB to 1 GB DDR	Broadcom VideoCore® IV 3D graphics subsystem with a OpenGL ES 1.1 and 2.0 driver. The amount of system memory affects performance and the use of the RPi as a general-purpose computing device. Memory is shared between the CPU and GPU.
3	Storage	micro-SD card	The RPi boards all boot from a micro-SD or SD card, with the exception of the Compute module. It has an on-board eMMC, which is effectively an SD card on a chip. The RPi 3 uses a friction-fit slot, rather than a click-in/click-out slot.
4	Power	micro-USB connector	A 5 V supply is required that should ideally deliver a current of at least 1.1 A and ideally 2.5 A for the RPi 3. There is over-current protection on this input. Be careful not to confuse the USB hub and USB power inputs on the RPi Zero.
5	Video Out	HDMI or mini-HDMI connector	Used to connect the RPi boards to a monitor or television. The RPi models support 14 output resolutions, including full-HD (1920 x 1080) and 1920 x 1200.
6	GPIOs	40 pin (or 26 pin) GPIO header	40 pins that are multiplexed to provide access to the features listed on the following table rows. Not all functionality is available at the same time. These inputs and outputs are described in detail in Chapter 6 and Chapter 8.
		26 x GPIOs	General purpose inputs outputs that are used for reading or writing binary data. The maximum number of GPIOs is 26 on the 40 pin RPi models. All GPIOs are 3.3V tolerant. Using buses and other interfaces reduces the number of available GPIOs.
		2 x I ² C bus	I ² C is a digital bus that allows you to connect several modules to each of the two-wire buses at the same time. One of these two buses is reserved for HAT support.
		SPI bus	Serial peripheral interface (SPI) provides a synchronous serial data link over short distances. It uses a master/slave configuration and requires 4 wires for communication. The RPi SPI bus has Linux support for two slave select lines.
		UART	Used for serial communication between two devices. The RPi typically (except the RPi 3) has one UART device that is allocated by default to providing a serial console connection.
		PWM	Pulse width modulation (PWM) outputs allow you to send a type of analog output that can be used to control devices (e.g., motors). There is at least one hardware PWM output on all RPi boards, and two on more recent boards.
		GPCLK	General purpose clocks (GPCLK) allow you to establish accurate timing signals.

Raspberry Pi 3

7	USB Hub	USB Connectors	There is an internal USB hub on RPi models with varying numbers of inputs. For example, the RPi 2/3 has five internal USB ports – one is connected to the Ethernet port and the other four are available for external connection.
8	Reset	Unpopulated RUN	Can be used as a reset button for the RPi. This topic is described later in Chapter 1.
9	Audio and Video	4-pole 3.5 mm jack	This provides composite video and stereo audio on more recent boards.
10	Power LED	PWR LED	Indicates that the board is powered (not on the RPi Zero).
11	Activity LED	ACT LED	Indicates that there is activity on the board (i.e., it flashes on SD card activity).
12	USB-to-Ethernet	SMSC LAN9514	This IC provides a USB 2.0 hub and a 10/100 Ethernet Controller. The RPi boards connect to the Internet via USB rather than an on-board Ethernet controller within the SoC.
13	Network	RJ-45 Ethernet	10/100 Mbps Ethernet via a RJ45 connector. The RPi 3 has on-board Wi-Fi and Bluetooth using a BCM43438. See the Optional Accessories section in this chapter.
14	Camera	CSI	The RPi has a Mobile Industry Processor Interface (MIPI) Camera Serial Interface (CSI), a 15-pin connector that can be connected to a special-purpose camera. See Chapter 15.
15	Display	DSI	The Display Serial Interface (DSI) is an interface that is typically used by mobile phone vendors to interface to a screen display. There are few displays available that support this interface – one example is the 7" Raspberry Pi Touchscreen (800x 480 display).

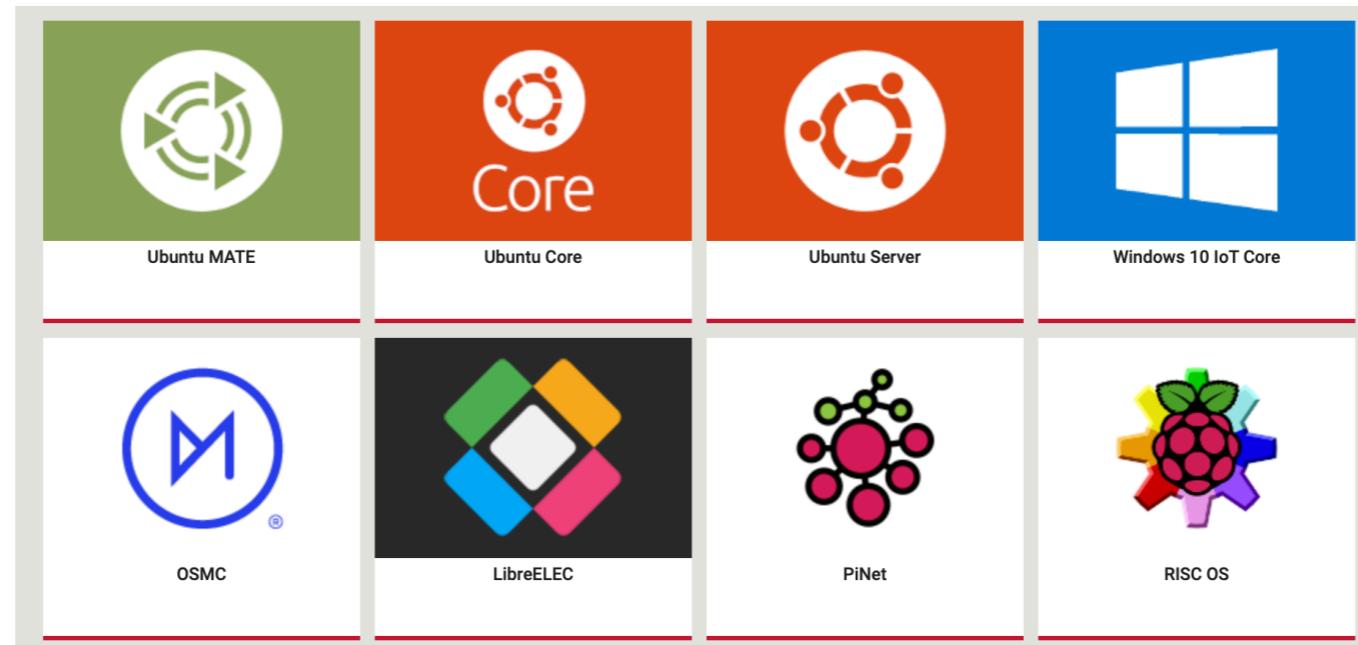
Linux Distributions for RPi

<https://www.raspberrypi.org/downloads/>

➊ Raspbian: RPi를 위해 배포된 Debian의 한 갈래

- ➏ Raspbian Buster: Debian Buster (Debian version 10.x)에 기반한 Raspbian 최신 버전.
Full Desktop 지원
- ➏ Raspbian Jessie, Wheezy, Stretch: Raspbian의 이전 버전들

➋ 다양한 Third Party OS 버전들



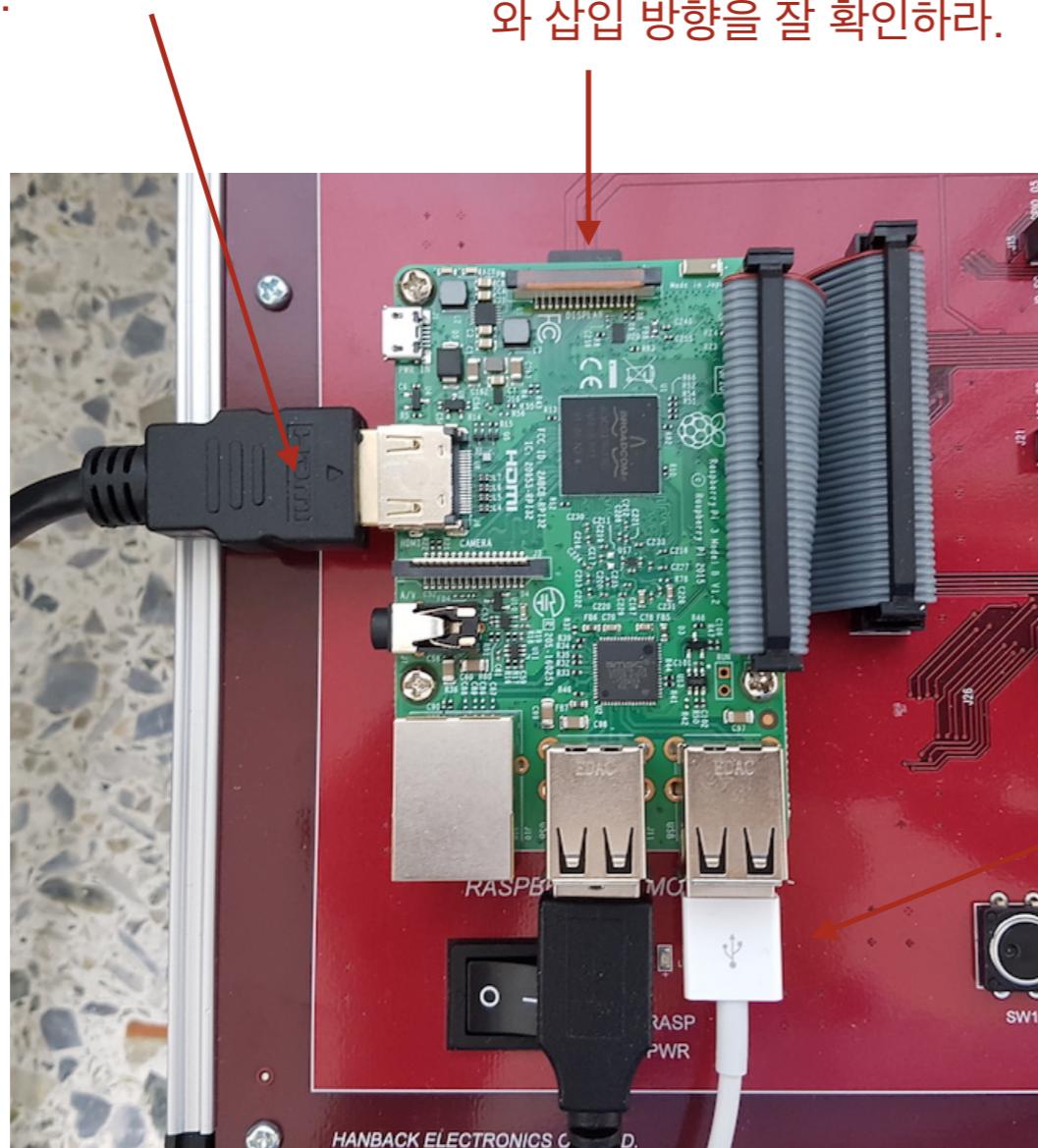
RPi 시작하기

RPi 부팅하기

- RPi에 Micro-SD 카드가 삽입되어 있는지 확인하고, HDMI 포트로 모니터를 연결, USB 마우스와 키보드를 연결, 전원을 연결한다.

HDMI 케이블로 모니터와 연결한다.

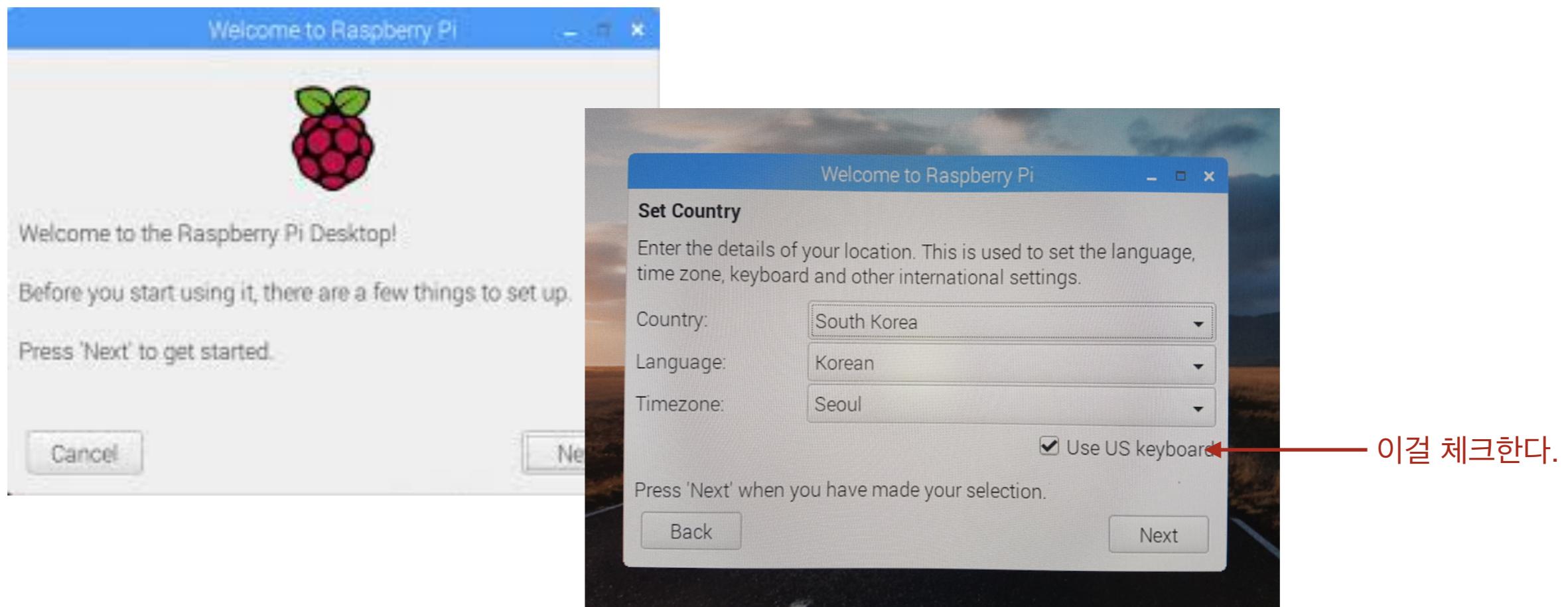
Micro-SD 카드 슬롯의 위치와 삽입 방향을 잘 확인하라.



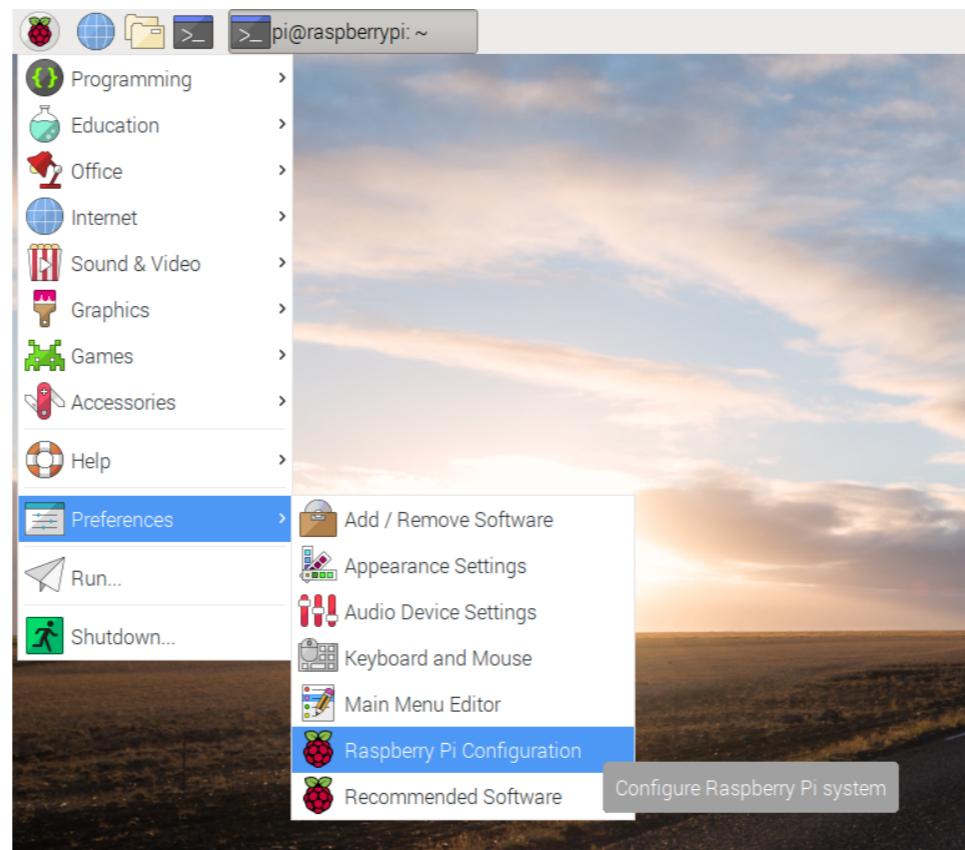
USB 마우스와 키보드를 연결한다.

RPi 설정하기

- 부팅이 완료되면 RPi 설정을 위한 대화상자가 나타난다. **Next**를 클릭하여 설정 사항으로 넘어간다.
- 국가를 설정하는 대화상자가 나타난다. South Korea로 설정하면 GUI 화면이 깨진다. 국가 설정은 미국(United States)으로 하고, “Use US Keyboard”를 클릭한다.

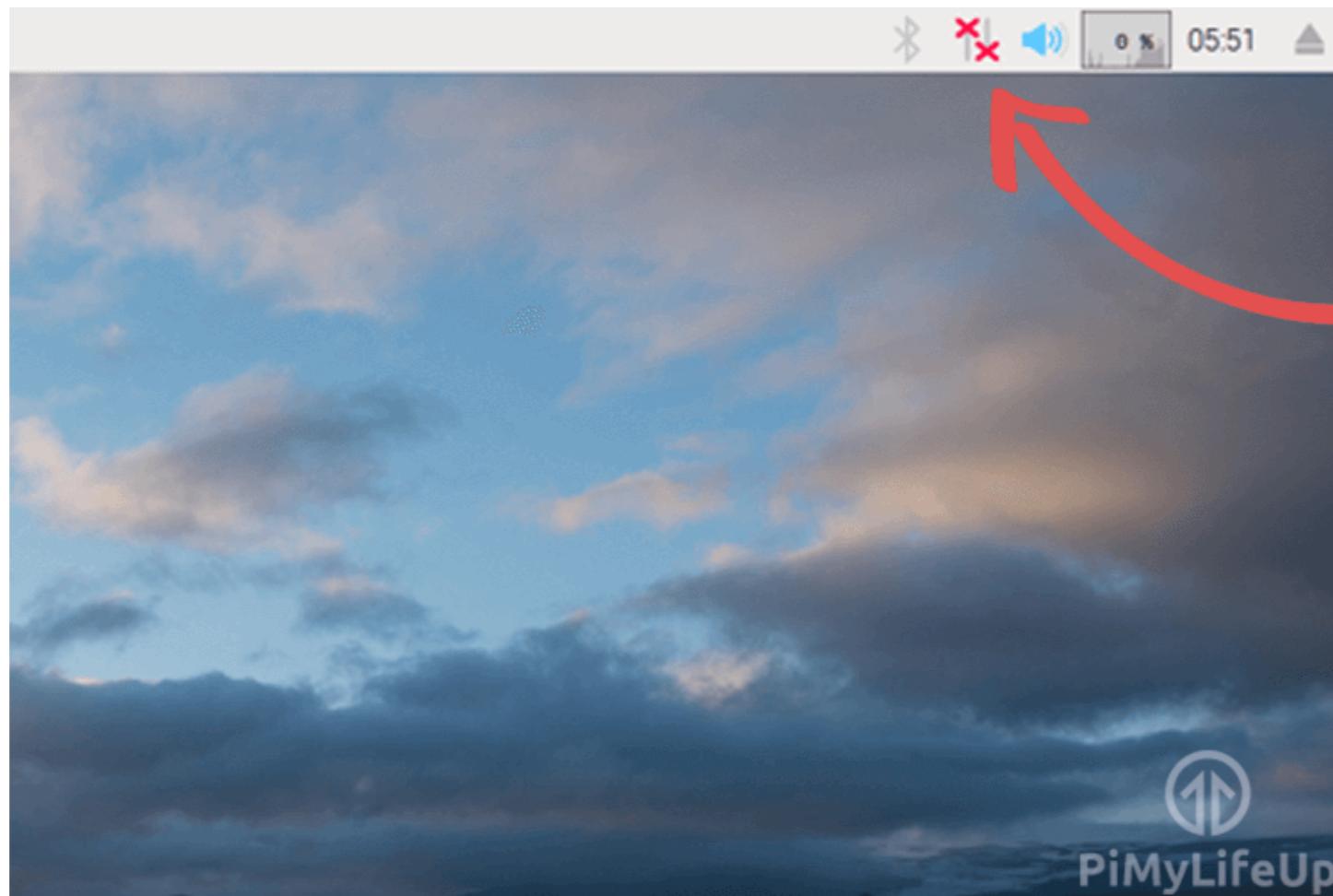


- 기본 계정의 id는 “pi”, 비번은 “raspberry”이다. 비밀번호는 변경하지 않는다.
- “Set Up Screen”, “Select Wifi Network”, “Update Software” 창은 그냥 Skip한다.
- 아래 그림과 같이 Raspberry Pi Configuration 메뉴를 선택하고 설정 창으로 들어갈 수 있다. 설정 창의 Interfaces 탭에서 Camera, SSH, SPI, I²C를 Enable한다.
- Localisation 탭에서는 WiFi Country를 미국으로 선택한다. 설정을 완료하면 시스템이 자동으로 재부팅 된다.



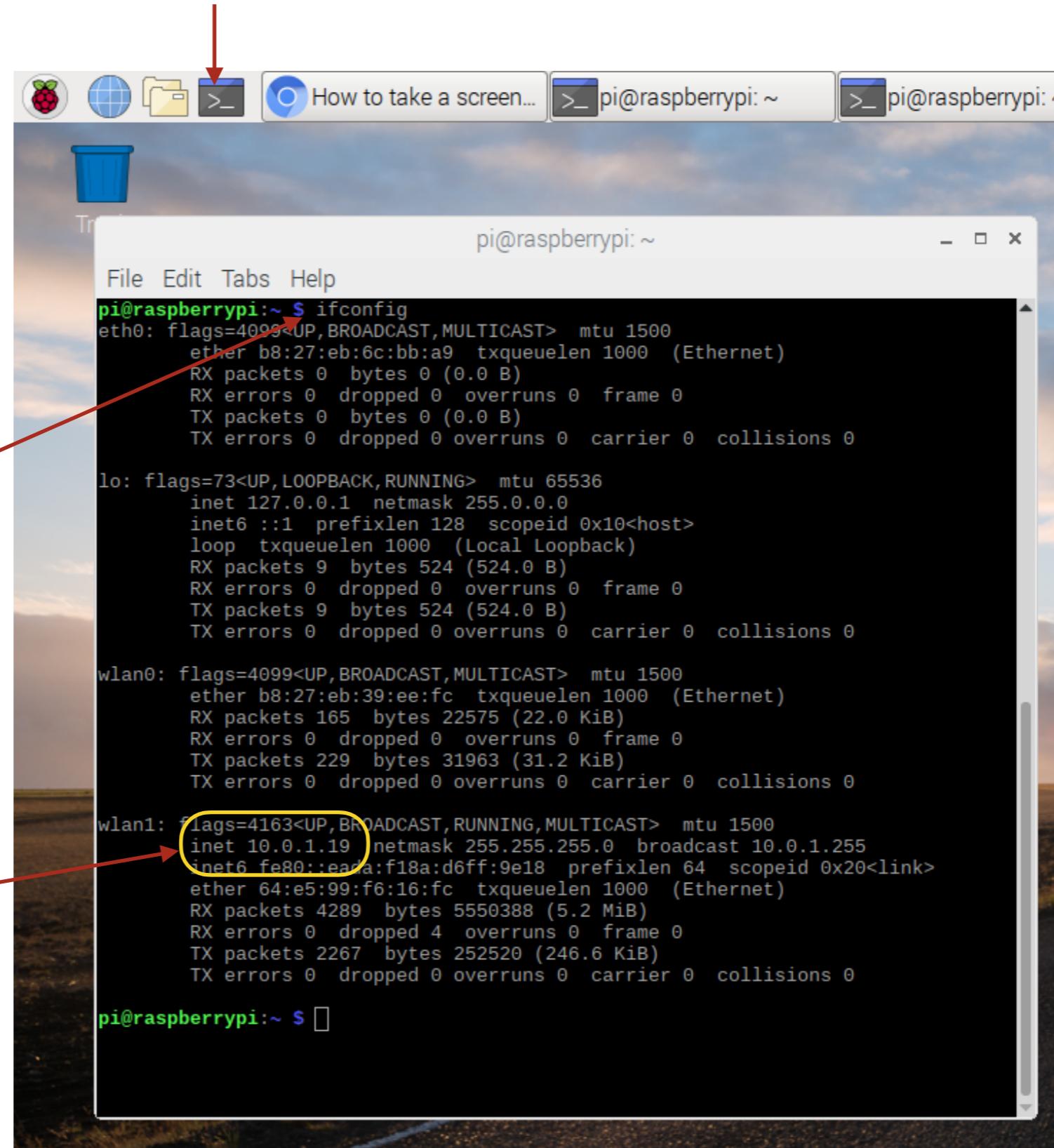
RPi WiFi 연결하기

- 화면 오른쪽 상단에서 WiFi를 선택하여 연결한다.



ip 주소 확인

1. 이걸 클릭하여 terminal을 실행한다.



2. ifconfig 명령을 실행한다.

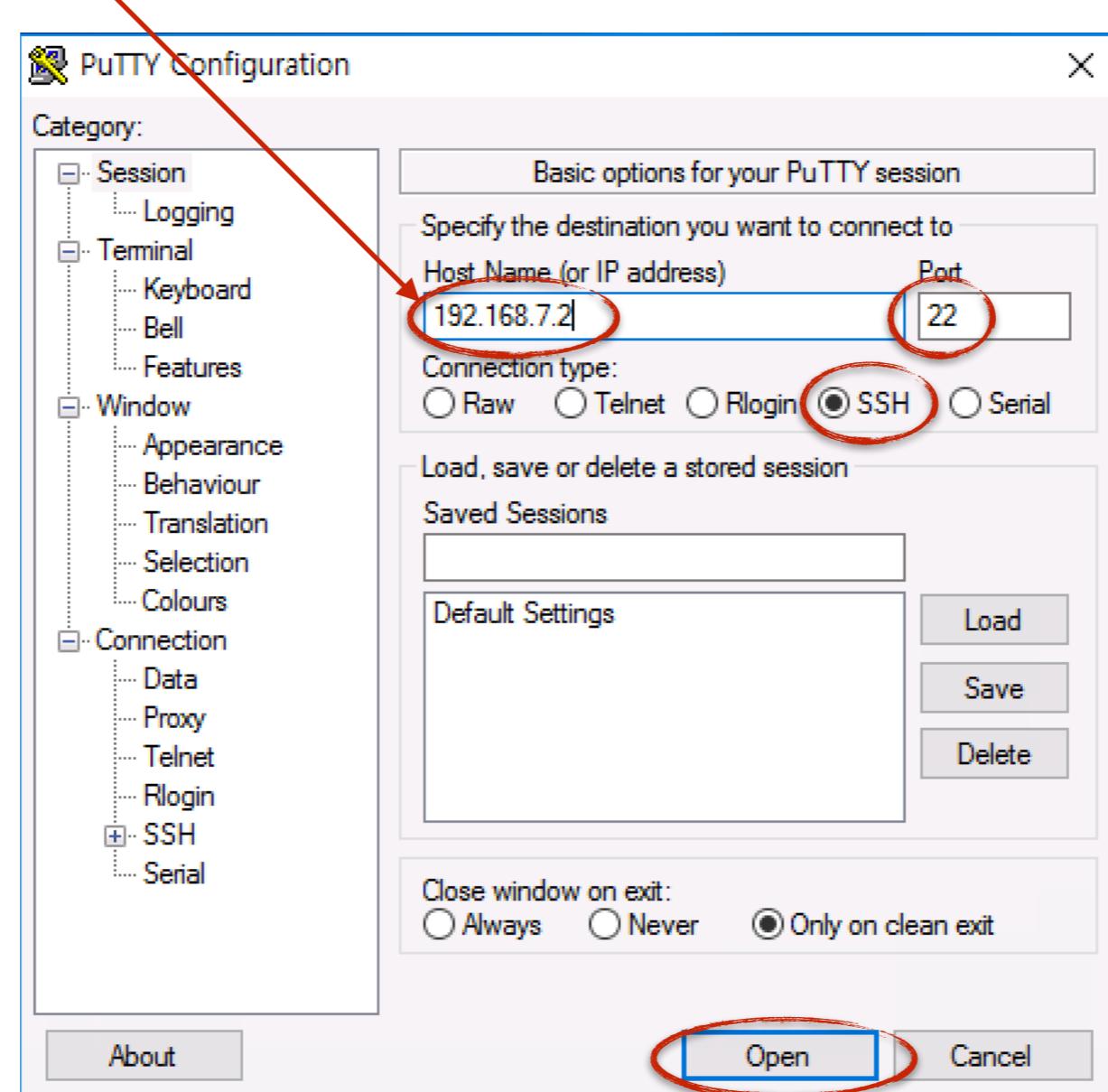
3. 이 부분에서 RPi에 할당된 ip address를 확인한다. (이 그림에서는 wlan1이지만 여러분의 경우에는 wlan0만 있을 것이다.)

PC에서 RPi에 SSH로 원격 접속하기

- ☞ PuTTY
 - ☞ Free, open-source **terminal emulator, serial console, and SSH client**
 - ☞ <http://www.putty.org>
- ☞ 다운로드 페이지에서 **putty-64bit-0.73-installer.msi**를 다운로드하여 설치한다. (그러면 putty와 함께 psftp등의 다른 프로그램이 함께 설치된다.)
- ☞ RPi가 WiFi에 연결된 상태에서 putty를 실행한다.

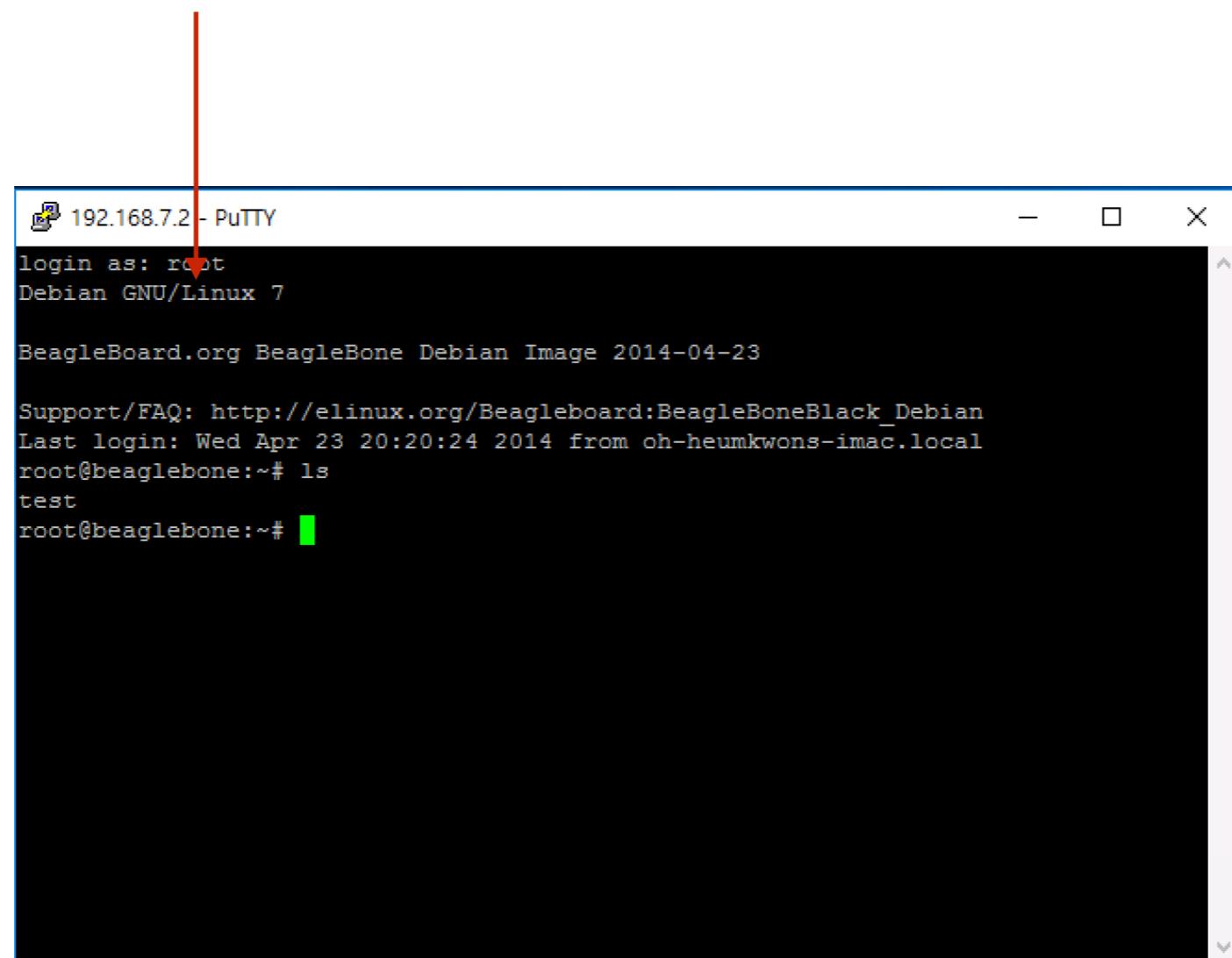
putty를 이용하여 SSH로 RPi에 접속하기

RPi에 할당된 IP 주소를 적는다.



SSH 연결

login id는 pi이고, password는 raspberry이다.



```
192.168.7.2 - PuTTY
login as: root
Debian GNU/Linux 7

BeagleBoard.org BeagleBone Debian Image 2014-04-23

Support/FAQ: http://elinux.org/Beagleboard:BeagleBoneBlack_Debian
Last login: Wed Apr 23 20:20:24 2014 from oh-heumkwons-imac.local
root@beaglebone:~# ls
test
root@beaglebone:~#
```

Power Shell로 접속

- Windows에 Power Shell이 설치되어 있는 경우 Power Shell로 접속하면 된다.

"\$ ssh pi@RPi의 IP주소"
로 접속한다.

```
PS C:\Users\ohheum> ssh pi@10.0.1.19
The authenticity of host '10.0.1.19 (10.0.1.19)' can't be established.
ECDSA key fingerprint is SHA256:ApELI0a3DiEzd15vKCdKCA/d5AkTFDDBxHg0+06s2M0.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '10.0.1.19' (ECDSA) to the list of known hosts.
pi@10.0.1.19's password:
Linux raspberrypi 4.14.98-v7+ #1200 SMP Tue Feb 12 20:27:48 GMT 2019 armv7l

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Thu Oct  3 13:22:26 2019 from 10.0.1.2
Linux raspberrypi 4.14.98-v7+ #1200 SMP Tue Feb 12 20:27:48 GMT 2019 armv7l

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Thu Oct  3 13:22:26 2019 from 10.0.1.2

SSH is enabled and the default password for the 'pi' user has not been changed.
This is a security risk - please login as the 'pi' user and type 'passwd' to set a new password.

pi@raspberrypi:~ $ ls
2019-05-18-172436_1824x984_scrot.png  2019-05-18-172742_1824x984_scrot.png  camera.py  Downloads  Public
2019-05-18-172514_1824x984_scrot.png  2019-05-18-172812_1824x984_scrot.png  chap04    image.jpg   __pycache__
2019-05-18-172629_1824x984_scrot.png  2019-05-18-172836_1824x984_scrot.png  Desktop   MagPi     Templates
2019-05-18-172640_1824x984_scrot.png  2019-10-03-131100_1824x984_scrot.png  Dev      Music     venv
2019-05-18-172705_1824x984_scrot.png  2019-10-03-131154_1824x984_scrot.png  Documents Pictures  Videos
pi@raspberrypi:~ $
pi@raspberrypi:~ $
```

psftp로 PC와 RPi간에 파일 전송하기

// PC의 C드라이브의 temp 폴더에 test.txt라는 이름의 파일을 하나 만든다. 내용은 아무렇게나...
// putty와 함께 설치된 sftp 프로그램을 실행하거나 PowerShell에서 "sftp pi@RPi의 IP주소" 명령을 // 실행한다.

psftp: no hostname specified; use "open host.name" to connect

psftp> **open pi@10.0.1.19** ← pi로 접속한다.

Using username "pi". Debian GNU/Linux 7

Remote working directory is /home/pi

psftp> **lcd c:\temp** ← PC 쪽에서 전송할 파일이 있는 디렉토리로 이동하는 명령은 lcd (local change directory)이다.

New local directory is c:\temp

psftp> **lls** ← PC의 현재 디렉토리에 있는 파일이나 서브 디렉토리의 이름을 나열한다.

test.txt

psftp> **pwd** ← RPi에서의 현재 디렉토리를 알려준다.

/home/pi

psftp> **mkdir test** ← RPi의 현재 디렉토리에 test라는 이름의 디렉토리를 만든다.

mkdir /home/pi/test: OK

psftp> **cd test** ← RPi에서 /home/pi/test 디렉토리로 이동한다.

psftp> **pwd** ← RPi에서 현재 디렉토리를 알려준다.

/home/pi/test

psftp> **put test.txt** ← put은 파일을 전송하고, get은 파일을 전송 받는 명령이다.

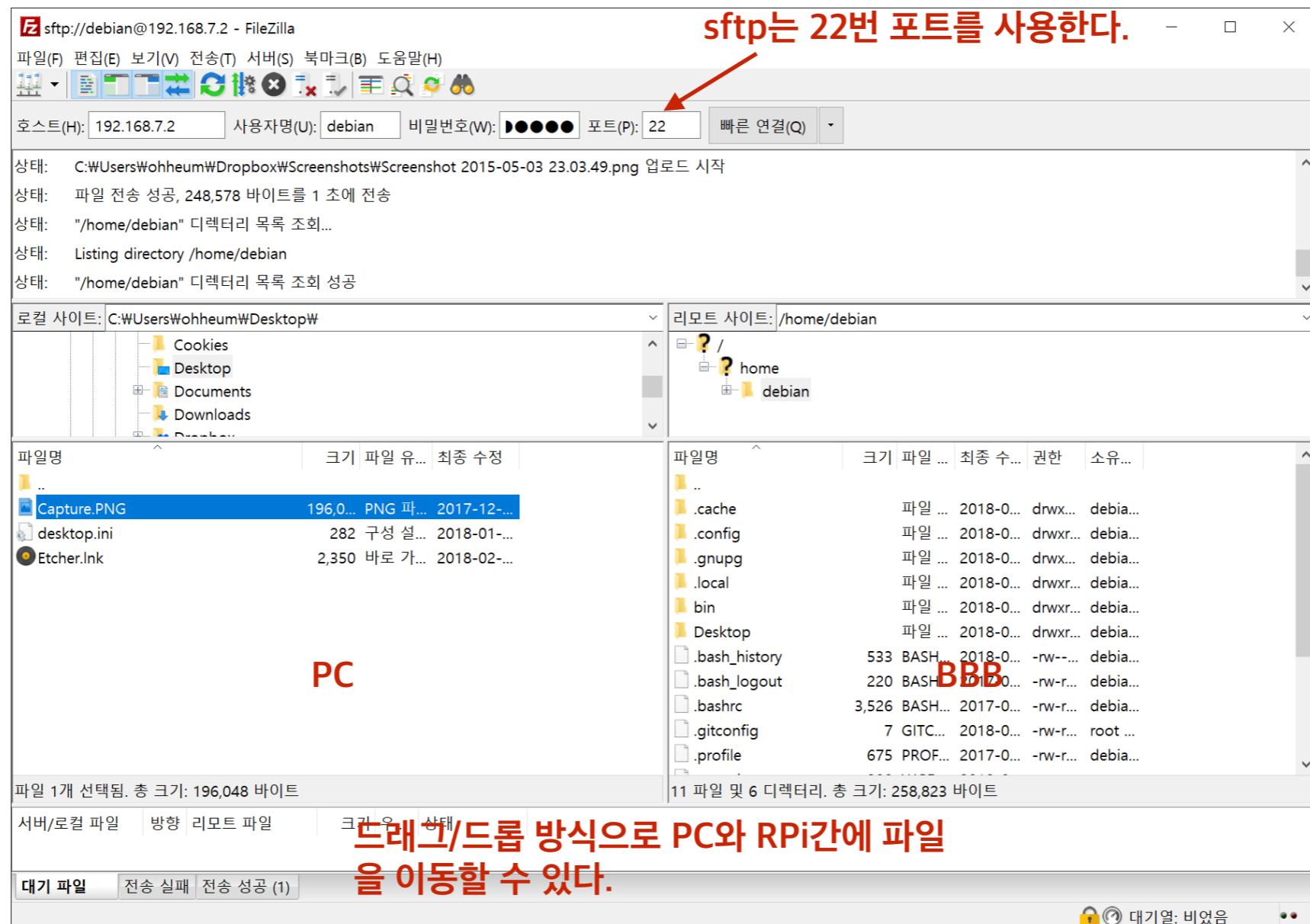
local:test.txt => remote:/home/pi/test.txt

psftp> **quit** ← 종료한다.

유용한 sftp 명령들

- ☞ 소문자 'l'로 시작하는 명령들은 PC쪽에 적용되는 명령들이다.
 - ☞ lcd (local change directory), lpwd (local print working directory).
- ☞ 하나의 파일을 전송하는 명령은 put, 하나의 파일을 역으로 전송받는 명령은 get이다. 동시에 여러 개의 파일을 전송하거나 전송 받을 때는 mput과 mget 명령을 사용한다.
- ☞ 그 외에도 다양한 명령들이 있음

- FTP와 SFTP 를 지원하는 free application
- <https://filezilla-project.org> 에서 Filezilla Client를 다운로드/설치하고 실행한다.



- Linux를 먼저 shutdown하지 않고 RPi의 전원을 차단하는 것은 SD 카드의 file system이나 전원 관리 칩을 손상할 수 있음
- 리눅스 shutdown 혹은 reboot하기 (root로 로그인한 경우에는 sudo 불필요)

```
# sudo shutdown -h now      /* 즉시 shutdown하기 */  
# sudo shutdown -h +5       /* 5분 후에 shutdown하기 */  
# sudo halt                 /* 즉시 shutdown하기 */  
# sudo reboot               /* 재부팅 하기 */
```

- USB 케이블을 연결했다 뺏다 하는 방식으로 ON/OFF 하지 말것

Linux 환경 구축

Linux 실습 환경 구축하기

- ☞ RPi가 있는 경우 RPi를 사용
- ☞ VMware player를 이용하여 가상머신으로 Linux 설치
 - ☞ 메모리 4G 이상이면 가상머신으로 설치해도 무방함
 - ☞ VMware Workstation player 15 (혹은 VirtualBox) 사용: [VMware 설치 참고 동영상](#)
 - ☞ Mac 사용자는 Parallels 혹은 VMware Fusion 최신버전 사용
- ☞ Windows 10 pro 혹은 edu 버전의 경우 WSL (Windows Subsystem for Linux) 사용
- ☞ Cloud server 사용
 - ☞ <https://repl.it/languages/bash>

Linux 소개

Kernel

- 1991년 Linus Tovalds가 개발 시작
- 운영체제의 핵심 구성요소

Shared Libraries

- 컴파일 타임(static library) 혹은 런 타임(dynamic library)에 응용 프로그램과 링크(link)되는 오브젝트 코드들

Application Programs

- 수많은 프로그램들.
- 파일조작관련, 텍스트 에디터, 웹 브라우저, 오디오 및 비디오, 프로그래밍, 워드 프로세서, 암호화 등

Shell & GUI 사용자 인터페이스

- 쉘은 명령어 방식의 사용자 인터페이스를 제공
- KDE, GNOME 등의 GUI

Linux 배포판 (distributions)

- ☞ 수 많은 배포판이 있음
- ☞ 크게 보면 Debian 계열과 Red Hat 계열이 지배적
- ☞ High-end Server
 - ☞ Red Hat Enterprise, Debian, OpenSUSE 등의 주로 사용됨
- ☞ Desktop
 - ☞ Ubuntu, Debian, Fedora, Linux Mint 등
- ☞ RPi에는 Debian 8(Jessie), 9(Stretch), 10(Buster)에 기반한 Raspbian 버전 8, 9, 혹은 10이 설치되어 있음

Linux 배포판 (distributions)

Table 1.1: Most popular general-purpose Linux distributions

Distribution	Web site	Comments
Arch	archlinux.org	For those who fear not the command line
CentOS	centos.org	Free analog of Red Hat Enterprise
CoreOS	coreos.com	Containers, containers everywhere
Debian	debian.org	Free as in freedom, most GNUish distro
Fedora	fedoraproject.org	Test bed for Red Hat Linux
Kali	kali.org	For penetration testers
Linux Mint	linuxmint.com	Ubuntu-based, desktop-friendly
openSUSE	opensuse.org	Free analog of SUSE Linux Enterprise
openWRT	openwrt.org	Linux for routers and embedded devices
Oracle Linux	oracle.com	Oracle-supported version of RHEL
RancherOS	rancher.com	20MiB, everything in containers
Red Hat Enterprise	redhat.com	Reliable, slow-changing, commercial
Slackware	slackware.com	Grizzled, long-surviving distro
SUSE Linux Enterprise	suse.com	Strong in Europe, multilingual
Ubuntu	ubuntu.com	Cleaned-up version of Debian

“[Unix and Linux System Administration Handbook, 5th Edition](#)”에서 인용

For complete list of linux distributions, <http://static.lwn.net/Distributions/>

접근 제어(Access Control)

User, Superuser

- ☞ Linux는 다중 사용자(multi-user) 운영체제. 각각의 사용자는 유일한 username을 가짐.
- ☞ superuser라고 불리는 특권을 가진 사용자: root
- ☞ superuser는 시스템의 거의 모든 자원에 대한 권한을 가짐
 - ☞ 일반 사용자는 자신의 home directory에 대해서만 완전한 권한을 가지며,
 - ☞ 소프트웨어 패키지의 설치/삭제, 시스템 설정, 네트워크 설정 등은 superuser권한이 있을 때만 가능
- ☞ RPi는 superuser인 root와 일반 사용자인 pi가 미리 설정된 상태
- ☞ root 계정으로 리눅스 명령어를 연습하는 것은 위험하므로 일반 사용자 계정인 pi로 로그인한다. 앞으로 특별한 언급이 없으면 항상 이 계정으로 로그인한다.

login

pi는 RPi에 미리 설정되어 있는 사용자 계정의 id이다.



login as: pi

pi@10.0.1.19's password: ← 초기 password는 raspberry이다.

Linux raspberrypi 4.14.98-v7+ #1200 SMP Tue Feb 12 20:27:48 GMT 2019 armv7l

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/*copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.

Last login: Thu Oct 3 13:37:00 2019 from 10.0.1.8

SSH is enabled and the default password for the 'pi' user has not been changed.
This is a security risk - please login as the 'pi' user and type 'passwd' to set a new
password.

pi@raspberrypi:~\$ ←

prompt의 형태는 “user_id@raspberrypi:현재 디렉토리\$”이다.

~은 사용자의 home directory를 표시하는 기호이다.

root계정에서는 \$대신 #이 표시된다. 물론 prompt의 형태는 바꿀 수 있다.

Linux Version 확인하기

여러 개의 명령어를 `&&`로 연결하여 한 번에 순차적으로 실행할 수 있다

(단 `&&`이므로 선행 명령이 실패하면 다음 명령들은 실행되지 않는다. 각각의 명령을 따로 실행해보자.)

```
pi@raspberrypi:~ $ uname -a && cat /etc/os-release && cat /proc/version  
Linux raspberrypi 4.19.75-v7+ #1270 SMP Tue Sep 24 18:45:11 BST 2019 armv7l GNU/Linux  
PRETTY_NAME="Raspbian GNU/Linux 10 (buster)"  
NAME="Raspbian GNU/Linux"  
VERSION_ID="10"  
VERSION="10 (buster)"  
VERSION_CODENAME=buster  
ID=raspbian  
ID_LIKE=debian  
HOME_URL="http://www.raspbian.org/"  
SUPPORT_URL="http://www.raspbian.org/RaspbianForums"  
BUG_REPORT_URL="http://www.raspbian.org/RaspbianBugs"  
Linux version 4.19.75-v7+ (dom@buildbot) (gcc version 4.9.3 (crosstool-NG crosstool-  
ng-1.22.0-88-g8460611)) #1270 SMP Tue Sep 24 18:45:11 BST 2019
```

Linux Kernel 버전

Raspbian 10 (buster)가 설치되어 있다.

- ▶ `uname -r` 혹은 `uname -a` 은 커널 버전을 알아내는 명령임
- ▶ `cat` 명령은 텍스트 파일을 화면에 디스플레이할 때 사용한다.
- ▶ `/etc/os-release` 파일과 `/proc/version` 파일은 설치된 Linux의 버전 정보를 저장한 파일이다.

superuser 권한 획득하기

1. root로 login

- 처음부터 root로 login: id는 “root”, 보통 패스워드도 “root”로 초기 설정되어 있음
- 보안 때문에 비추천: Ubuntu와 Rasbian 10의 경우 default로 disable되어 있음
- 보안 문제로 ssh로 접속시에는 root login이 막혀있는 경우가 많음

2. su 명령을 사용하여 root로 전환하는 방법

- su (switch user, substitute user) 명령으로 root 계정으로 전환 가능 (역시 보안 문제로 막혀있는 경우가 많음)
- su 명령을 입력하면 root의 password를 물어봄.
- su 명령으로 root가 아닌 다른 사용자 계정으로도 전환 가능

3. sudo 명령을 사용하는 방법

- root계정으로 전환하지 않고 일반 사용자가 sudo 명령으로 privileged 명령 수행 가능
- 각각의 사용자에 대해서 sudo 명령을 수행할 권한 여부를 지정
- 일반적으로 권장되는 방법

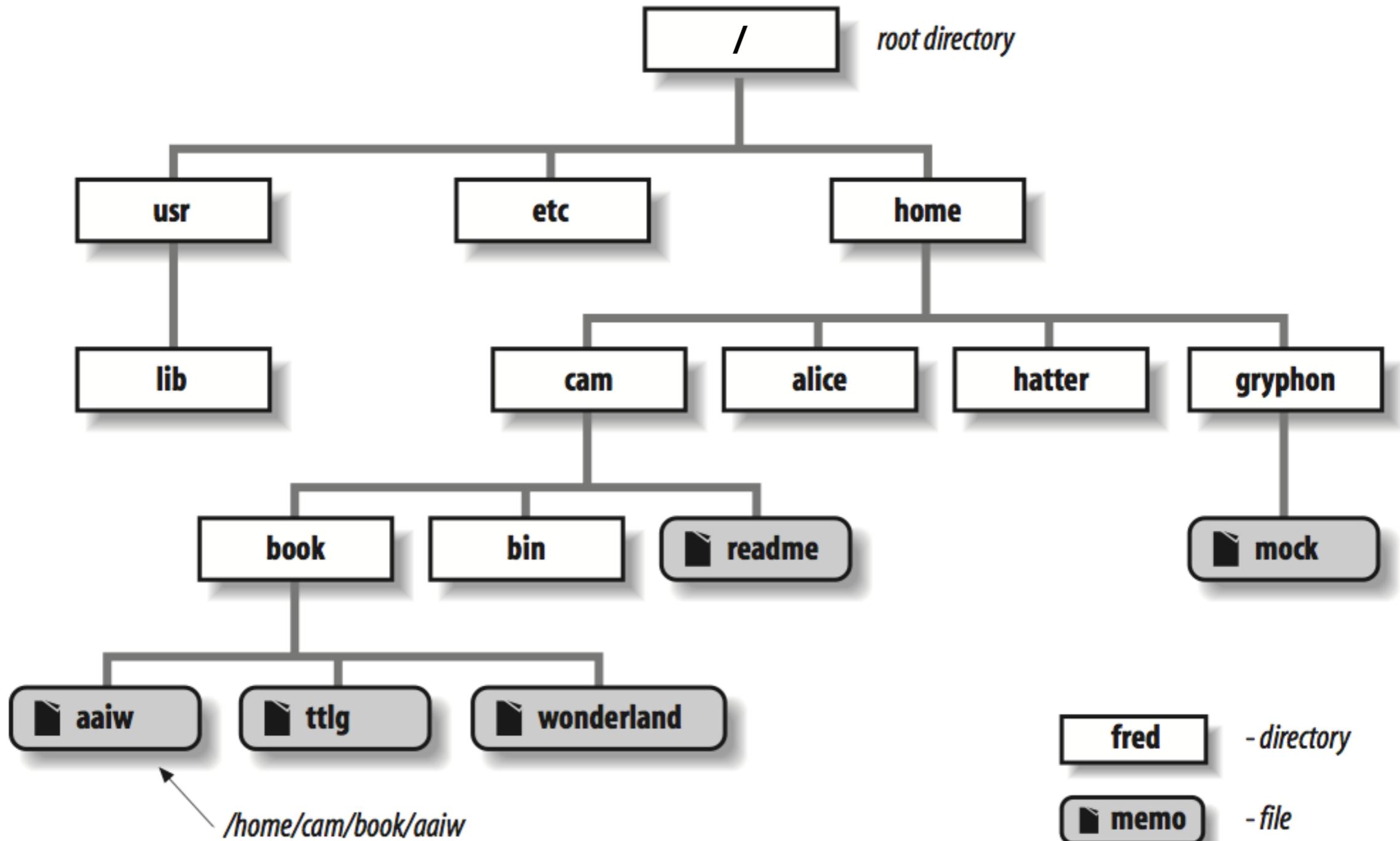
su / sudo

- ❶ superuser가 되기 위해서는 su 명령을 사용 (Ubuntu와 RPi에서는 root 계정으로 switch하는 것은 기본적으로는 막혀있음)
- ❷ 혹은 “sudo” 명령을 사용

```
pi@raspberrypi:~$ sudo any-privileged-command-you-want  
Password: (root 패스워드가 아닌 자신의 password를 입력)
```

파일 시스템 사용하기

파일과 디렉토리



- ⦿ 루트 디렉토리
 - ⦿ 최상위 디렉토리, /로 표시
- ⦿ 절대 경로
 - ⦿ /home/cam/book/wonderland
- ⦿ 현재(작업) 디렉토리와 상대경로
 - ⦿ 사용자가 현재 위치한 디렉토리 (pwd 명령은 현재 디렉토리의 절대 경로명을 보여줌)
 - ⦿ 상대 경로: book/wonderland

- ☞ 각각의 사용자에게 할당된 디렉토리
- ☞ Debian/Ubuntu 등에서는 “/home/username”
- ☞ 자신의 홈 디렉토리 찾기/이동하기:
 - ☞ \$ cd
 - ☞ 매개변수 없는 cd 명령은 자신의 홈 디렉토리로 이동
 - ☞ HOME 환경 변수는 자신의 홈 디렉토리 이름을 값으로 가짐
 - ☞ 예: \$ echo \$HOME
 - ☞ ~
 - ☞ 디렉토리명이 들어갈 자리에 있는 ~는 자신의 홈 디렉토리명으로 대체됨
 - ☞ 예: \$ echo ~

▶ echo 명령은 입력된 값을 "그대로" 화면에 출력하는 명령임
pi@raspberrypi:~\$ echo hello world
hello world

▶ 단, echo 다음에 어떤 변수가 나오면 그 변수의 값을 대입하여 출력함. 즉 HOME, ~
등은 변수임
pi@raspberrypi:~\$ echo \$HOME
/home/pi
pi@raspberrypi:~\$ echo ~
/home/pi

Special Directories

.

- 항상 현재 디렉토리를 나타냄
- 따라서 ./readme.txt와 readme.txt는 동일

..

- 현재 디렉토리의 부모 디렉토리
- cd ..
- 예: 만약 현재 디렉토리가 /one/two/three라면 .는 이 디렉토리 자체를, 그리고 .. 는 /one/two를 의미

~

- 쉘(shell)은 ~를 항상 사용자의 홈 디렉토리명으로 대체한다.

```
$ cd /
$ pwd
/
$ cd ~
$ pwd
/home/pi
$ cd ..
$ pwd
/home
$ cd
```

Basic File System Commands

NAME	COMMAND	OPTIONS AND FURTHER INFORMATION	EXAMPLE(S)
List files	ls	<ul style="list-style-type: none">-a shows all (including hidden files)-l displays long format-R gives a recursive listing-r gives a reverse listing-t sorts last modified-S sorts by file size-h gives human readable file sizes	ls -al
Current directory	pwd	<ul style="list-style-type: none">Print the working directory-P prints the physical location	pwd -P
Change directory	cd	<ul style="list-style-type: none">Change directorycd then Enter or cd ~/ takes you to the home directorycd / takes you to the file system rootcd .. takes you up a level	cd /home/root cd / cd ..
Make a directory	mkdir	Make a directory	mkdir test
Delete a file or directory	rm	<ul style="list-style-type: none">Delete a file-r recursive delete (use for directories)-d remove empty directories	rm bad.txt rm -r test

Basic File System Commands (계속)

NAME	COMMAND	OPTIONS AND FURTHER INFORMATION	EXAMPLE(S)
Copy a file or directory	cp	-r recursive copy -u copy only if the source is newer than the destination or the destination is missing -v verbose copy (i.e., show output)	cp a.txt b.txt cp -r test testa
Move a file or directory	mv	-i prompts before overwrite No -r for directory. Moving to the same directory performs a renaming.	mv a.txt c.txt mv test testb
Create an empty file	touch	Create an empty file or update the modification date of an existing file.	touch d.txt
View content of a file	more	View the contents of a file. Use the Space key for the next page.	more d.txt
Get the calendar	cal	Display a text-based calendar.	cal 01 2015

- ⦿ **cat file1 file2 file3 ... (concatenate)**

- ⦿ 주어진 파일들을 concatenate해서 표준 출력으로 출력
 - ⦿ 결국 화면으로 파일의 내용을 보여줌

- ⦿ **more file1 file2 file3 ...**

- ⦿ 페이지 단위로 보여주며, 각 페이지마다 키보드 입력시 다음 페이지로
 - ⦿ 검색 기능 제공 (/ command)

- ⦿ **less file1 file2 file3 ...**

- ⦿ Does more than “more” with “less” !
 - ⦿ 역 방향 이동 가능 (? command).

head 와 tail

- ⦿ **head [<n>] <file>**
 - ⦿ 파일의 처음 <n> 라인을 출력
- ⦿ **tail [<n>] <file>**
 - ⦿ 파일의 마지막 <n> 라인을 출력
 - ⦿ `tail -f <file>`
처음 10 라인을 보여주고, 새로운 라인이 파일에 추가되면 계속해서 디스플레이 해줌. 로그 파일을 볼때 편리함.
- ⦿ **예:**
 - \$ `head windows_bugs.txt`
 - \$ `tail -f outlook_vulnerabilities.txt`

- ⦿ **grep <pattern> <files>**

- ⦿ 파일을 스캔하여 주어진 패턴을 포함하는 라인들을 찾아서 출력

- ⦿ **grep error *.log**

- ⦿ 현재 디렉토리에서 확장자가 .log인 모든 파일들에서 error라는 단어가 포함된 라인들을 화면에 출력

- ⦿ **grep -i error *.log**

- ⦿ 대소문자 구분하지 않음

- ⦿ **grep -ri error .**

- ⦿ 현재 디렉토리와 그 하위 디렉토리의 모든 파일에 대해서 recursive하게 검색

```
▶ /var/log 디렉토리에서 error에 대한 log를 검색해보자.  
pi@raspberrypi:~$ grep error /var/log/*.log  
pi@raspberrypi:~$ sudo grep error /var/log/*.log  
pi@raspberrypi:~$ sudo grep -ri error /var/log/  
▶ /etc 디렉토리에서 검색해보자.  
pi@raspberrypi:~$ sudo grep -ri 2226 /etc  
pi@raspberrypi:~$ sudo grep -ri stfp /etc
```

```
pi@raspberrypi:~$ cd ← 홈 디렉토리로 이동
pi@raspberrypi:~$ pwd ← 현재 디렉토리를 출력
/home/pi
pi@raspberrypi:~$ mkdir test ← 현재 디렉토리에 새로운 디렉토리 test를 생성
pi@raspberrypi:~$ ls ← 현재 디렉토리에 포함된 파일과 서브디렉토리의 목록을 표시
Desktop bin test
pi@raspberrypi:~$ cd test ← 디렉토리 test 내부로 이동
pi@raspberrypi:~/test$ touch hello.txt ← 빈(empty) 파일 hello.txt 생성
pi@raspberrypi:~/test$ ls
hello.txt
pi@raspberrypi:~/test$ cd .. ← 상위 디렉토리로 이동
pi@raspberrypi:~$ mkdir tmp ← 새로운 디렉토리 tmp 생성
```

```
pi@raspberrypi:~$ ls -l ← 파일에 대한 자세한 목록을 출력
total 16
drwxr-xr-x 2 pi pi 4096 Apr 23 2014 Desktop
drwxr-xr-x 2 pi pi 4096 Apr 23 2014 bin
drwxr-xr-x 2 pi pi 4096 Jan 2 05:36 test
drwxr-xr-x 2 pi pi 4096 Jan 2 05:37 tmp
pi@raspberrypi:~$ cp test/hello.txt tmp
pi@raspberrypi:~$ ls -al tmp/ ← tmp디렉토리의 내용을
total 8                                         숨김파일 포함해서 자세하게 출력
drwxr-xr-x 2 pi pi 4096 Jan 2 05:37 .
drwxr-xr-x 11 pi pi 4096 Jan 2 05:36 ..
-rw-r--r-- 1 pi pi 0 Jan 2 05:37 hello.txt
```

SHORTCUT	DESCRIPTION
Up arrow (repeat)	Gives you the last command you typed, and then the previous commands on repeated presses
Tab key	Auto-completes the file name, the directory name, or even the executable command name. For example, to change to the Linux /tmp directory you can type <code>cd /t</code> and then press Tab, which will auto-complete the command to <code>cd /tmp/</code> . If there are many options, press the Tab key again to see all of the options as a list
Ctrl+A	Brings you back to the start of the line you are typing
Ctrl+E	Brings you to the end of the line you are typing
Ctrl+U	Clears to the start of the line. Ctrl+E and then Ctrl+U clears the line
Ctrl+L	Clears the screen
Ctrl+C	Kills whatever process is currently running
Ctrl+Z	Puts the current process into the background. Typing <code>bg</code> then leaves it running in the background, and <code>fg</code> then brings it back to the foreground. This is discussed under Linux Processes in the next chapter

파일명 자동 완성

- 파일명을 입력하는 도중에 TAB키를 누르면 일치하는 이름을 찾아 자동으로 완성해주는 기능
- 2개 이상의 파일명과 일치할 경우 경고음이 들리고, TAB키를 두번 연속해서 누르면 일치 되는 파일명들만 보여줌
- 예:

```
$ cd /usr/bin  
$ ls un<TAB><TAB>
```

Symbolic Link

- ⦿ 원도우즈의 단축 아이콘과 유사
- ⦿ 다른 파일이나 디렉토리의 이름에 대한 참조
- ⦿ 동일한 파일이 디렉토리 구조상 2곳 이상에서 필요할 경우 사용
- ⦿ 심볼릭 링크 만들기:
 - ⦿ `ln -s file_name link_name`
- ⦿ 다른 디렉토리에 있는 파일의 심볼릭 링크를 동일한 이름으로 만들기:
 - ⦿ `ln -s ../README.txt`
- ⦿ 링크 삭제하기:
 - ⦿ `rm link_name`
 - ⦿ 물론 링크된 실제 파일은 삭제되지 않음

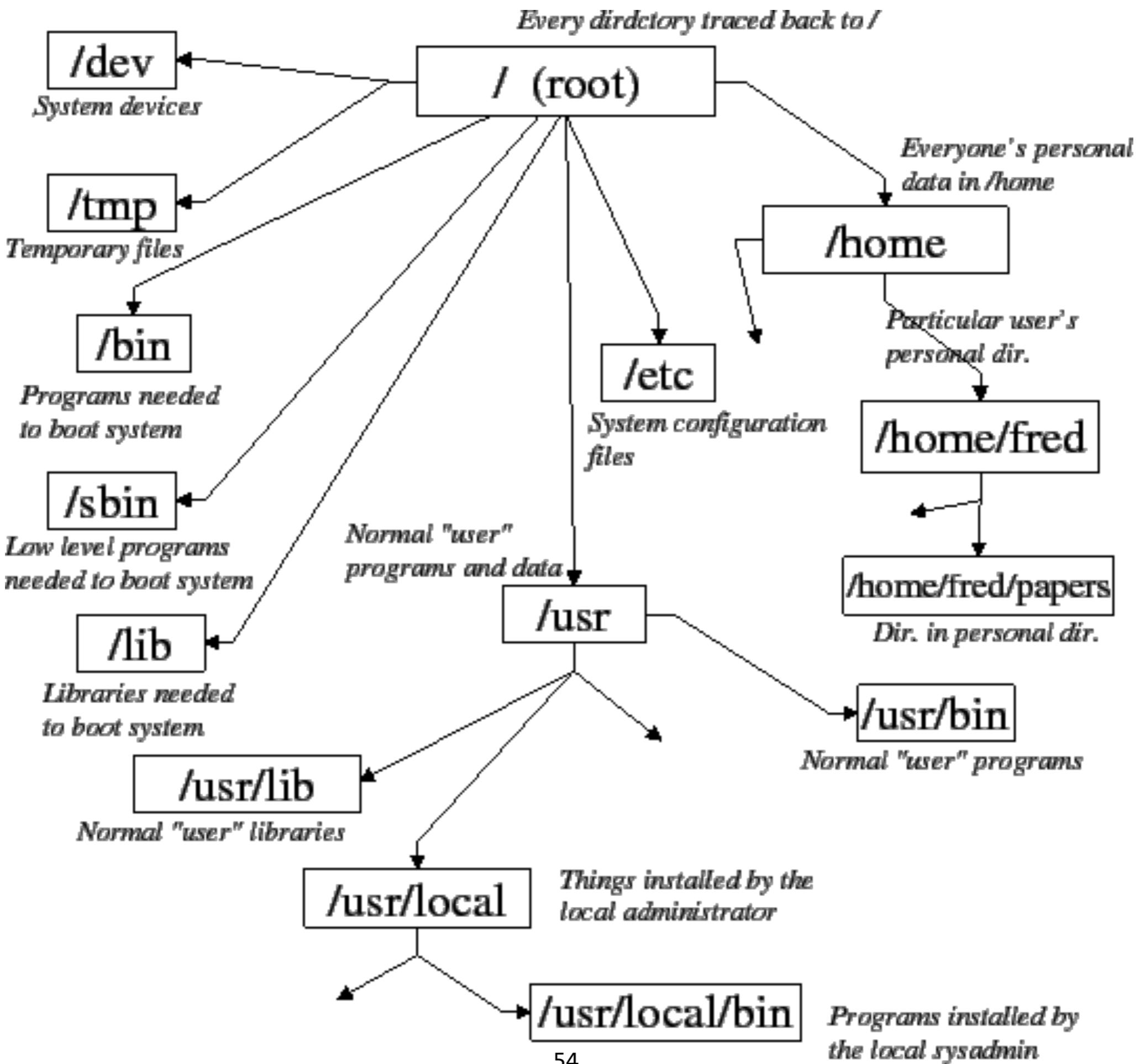
Hard Link

- -s 옵션 없이 명령어 `ln` 을 실행하면 하드 링크가 만들어짐
- 하드 링크는 파일에 대하여 원래의 이름과 완벽히 동일한 자격을 가진 또 하나의 이름이 됨. 즉 원래의 파일 이름 또한 그 파일에 대한 하나의 하드링크임
- 원본 파일은 그 파일에 대한 모든 하드링크가 삭제될 경우에만 실제로 삭제됨

Symbolic/Hard Link

```
pi@raspberry:~$ touch file.txt
pi@raspberry:~$ ln -s file.txt slink1
pi@raspberry:~$ ln file.txt hlink1
pi@raspberry:~$ ls -ail
total 50
 2005 drwxr-xr-x  12 pi      pi          4096 Jan  7 05:28 .
     16 drwxr-xr-x   5 root    root        4096 Jan  2 04:35 ..
 2380 -rw-r--r--   1 pi      pi          675 Jan  1 2013 .profile
36440 -rw-r--r--   2 pi      pi           0 Jan  7 05:28 file.txt
36440 -rw-r--r--   2 pi      pi           0 Jan  7 05:28 hlink1
52083 lrwxrwxrwx   1 pi      pi          8 Jan  7 05:28 slink1 -> file.txt
```

원본 파일과 하드 링크는 동일한 inode index를 가짐



Linux 기본 명령어 연습

- ☞ <https://www.youtube.com/watch?v=DkpmcTRGmt4>
- ☞ https://www.youtube.com/watch?v=3WdpHN3T_LE
- ☞ <https://www.youtube.com/watch?v=bOIboAsdgAw>

파일 접근 권한

• 누가 ?

- Owner (u): 파일의 소유자
 - Group (g): 파일의 소유 그룹
 - Other (o): 그 외의 모든 사용자

어떤 권한 ?

- Read (r)
 - Write (w)
 - Execute (x)

- 따라서 각각의 파일에 대해서 $3 \times 3 = 9$ 종류의 권한

▶ 시스템에 등록된 사용자 알아보기:

```
$ cat /etc/passwd
```

root:x:0:0:root:/root:/bin/bash

daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin

1

pi:x:1000:1000:,,,:/home/pi:/bin/bash...

파일의 권한 확인하기

```
$ ls -l filename
```

```
-rw-r--r-- 1 smith smith 7384 Jan 04 22:40 filename
```

or

```
$ ls -l
```

가장 왼쪽의 10 문자가 파일의 권한을 표시

Position	Meaning
1	File type: - = file, d = directory, l = symbolic link, p = named pipe, c = character device, b = block device
2–4	Read, write, and execute permissions for the file's owner
5–7	Read, write, and execute permissions for the file's group
8–10	Read, write, and execute permissions for all other users

- ⦿ **chmod <permissions> <files>**
- ⦿ <permissions>를 지정하는 두 가지 방법이 있음
- ⦿ 8진수 표현
 - ⦿ 예: chmod 644 <file>
 - ⦿ 여기서 644 = 110 100 100 즉, rw for u, r for g and o를 의미
- ⦿ 기호를 이용한 표현
 - ⦿ chmod go+r : 읽기 권한을 group과 others에 추가
 - ⦿ chmod u-w : 쓰기 권한을 user로 부터 제거
 - ⦿ chmod a-x: 실행 권한을 모두(a: all)로 부터 제거

File Ownership 변경

- ☞ 임베디드 시스템 개발에 유용
- ☞ **chown -R sco /home/linux/src**
 - ☞ 사용자 sco를 /home/linux/src에 있는 모든 파일들의 소유자로
- ☞ **chgrp -R empire /home/askywakler**
 - ☞ empire를 /home/askywakler 에 있는 모든 것들의 소유 group으로
- ☞ 혹은 한 번에
 - ☞ **chown -R ohheum:ohheum testdir**

기타 Shell 기능

예:

```
$ ls a*
```

쉘은 먼저 `a*`를 현재 디렉토리 내에서 ‘`a`’로 시작하는 모든 파일들의 이름으로 대체한 후 명령어를 실행함

Wildcard Meaning

*	Zero or more consecutive characters
?	Any single character
[<i>set</i>]	Any single character in the given <i>set</i> , most commonly a sequence of characters, like [aeiouAEIOU] for all vowels, or a range with a dash, like [A-Z] for all capital letters
[^ <i>set</i>]	Any single character <i>not</i> in the given <i>set</i> (as in the earlier example)
[! <i>set</i>]	Same as ^

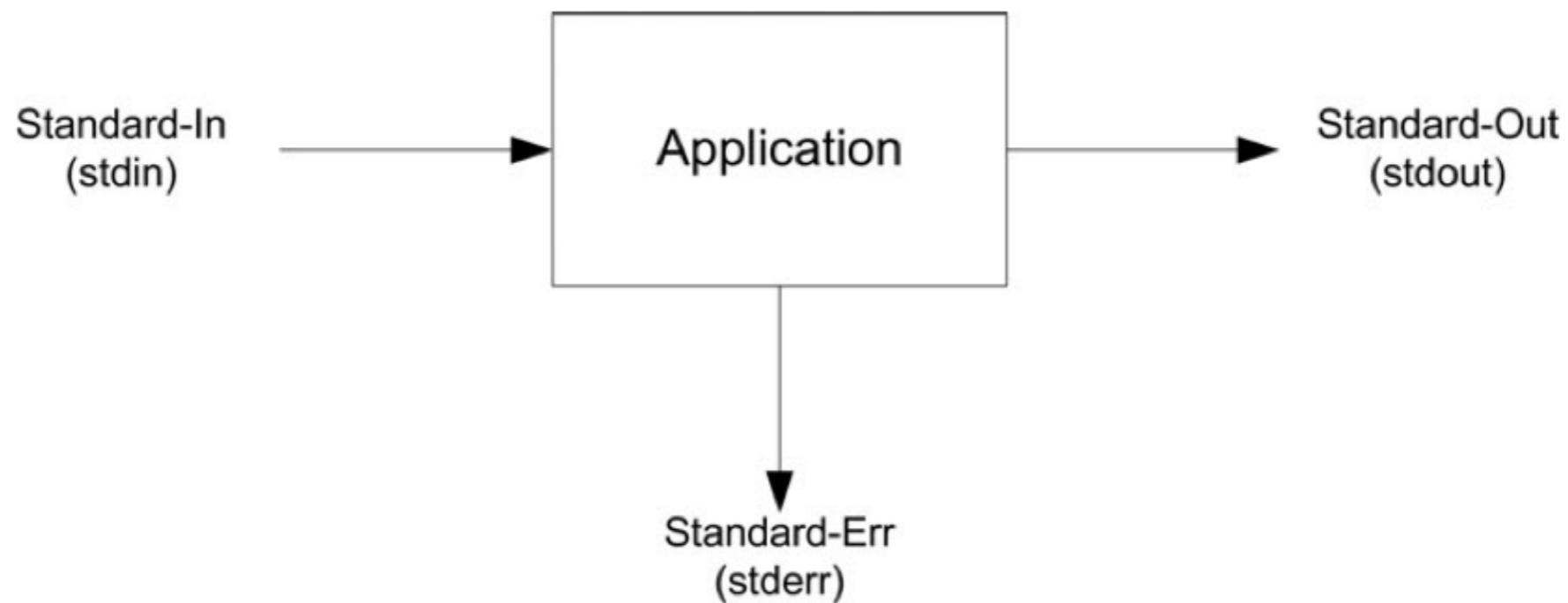
예 1:

- \$ ls /bin/[a-c]*
a, b, 혹은 c로 시작하는 모든 파일명을 나열

예 2:

- \$ ls /bin/[!a-o]*
\$ ls /bin/[^a-o]*

표준 입력, 출력, 에러



Descriptor	Description
0	Standard input (stdin)
1	Standard output (stdout)
2	Standard error (stderr)

입출력 redirection

- 표준 입력, 출력, 에러를 파일로부터 혹은 파일로 redirect.

입력 redirection

- 표준 입력으로부터 입력을 받는 명령어(혹은 프로그램)가 파일로 부터 입력을 받도록
 - \$ mycommand < infile

출력 redirection

- 표준 출력으로 출력하는 명령어(혹은 프로그램)가 파일로 출력하도록
 - \$ mycommand > outfile Create/overwrite outfile
 - \$ mycommand >> outfile Append to outfile

예:

- \$ ls -l > list
- \$ cat list
- \$ sort < list

▶ 키보드로부터 하나의 문자열을 입력받아 그대로 화면에 출력하는 프로그램 myecho.c를 작성하고 다음과 같이 컴파일한다.
pi@raspberrypi:~\$ cat myecho.c
#include <stdio.h>
int main() {
 char buf[20]; scanf("%s", buf); printf("%s\n", buf);
}
pi@raspberrypi:~\$ gcc -o myecho myecho.c
▶ 하나의 문자열이 저장된 input.txt 파일을 만든 후 다음과 같이 실행 해본다.
pi@raspberrypi:~\$./myecho < input.txt > output.txt

입출력 redirection

- 표준 에러 redirection

- \$ mycommand 2> errorfile

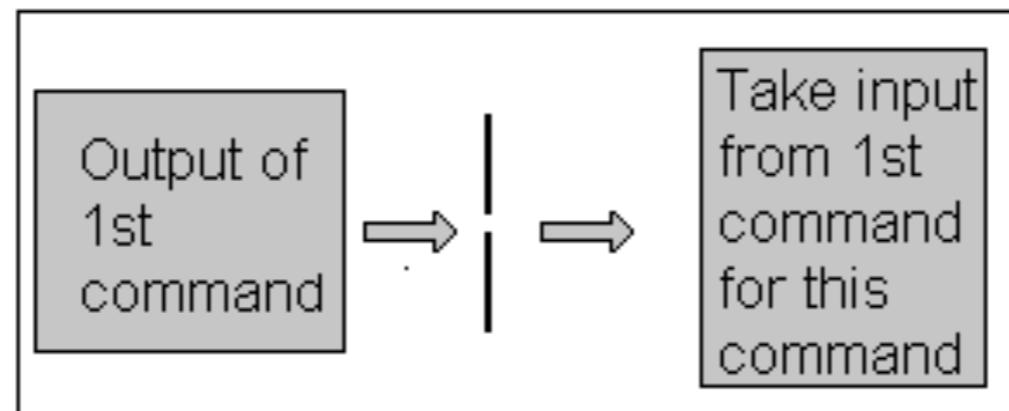
- 표준 출력과 표준 에러를 파일로 출력

- \$ mycommand > outfile 2> errorfile 별개의 파일로
 - \$ mycommand > outfile 2>&1 하나의 파일로

입출력 redirection

```
$ cat > sname  
vivek  
ashish  
zebra  
babu  
Press CTRL+D to save.  
$ sort < sname > sorted_names  
$ cat sorted_names  
ashish  
babu  
vivek  
zebraIn
```

- 파이프(|) 연산자를 이용해 한 명령어(프로그램)의 표준 출력을 다른 명령어(프로그램)의 표준 입력으로 redirect.



- 예:

```
$ who | sort
```

Pipe

Command using Pipes	Meaning or Use of Pipes
\$ ls more	Output of ls command is given as input to more command So that output is printed one screen full page at a time.
\$ who sort	Output of who command is given as input to sort command So that it will print sorted list of users
\$ who sort > user_list	Same as above except output of sort is send to (redirected) user_list file
\$ who wc -l	Output of who command is given as input to wc command So that it will number of user who logon to system
\$ ls -l wc -l	Output of ls command is given as input to wc command So that it will print number of files in current directory.
\$ who grep raju	Output of who command is given as input to grep command So that it will print if particular user name if he is logon or nothing is printed (To see particular user is logon or not)

▶ my echo 프로그램으로 파이프를 테스트해본다.

```
pi@raspberrypi:~ $ echo HelloRaspberryPi! | ./myecho  
HelloRaspberryPi!
```

- ⦿ 여러 개의 명령어를 순차적으로 실행
 - ⦿ \$ command1; command2; command3
 - ⦿ \$ date; who
- ⦿ 여러 명령어를 순차적으로 실행하되 하나라도 실패할 경우 중단하거나,
 - ⦿ \$ command1 && command2 && command3
- ⦿ 혹은 반대로 하나라도 성공할 경우 중단
 - ⦿ \$ command1 || command2 || command3

Text Editor: nano

```
pi@raspberrypi:~$ nano hello.txt
```



환경변수와 login script

- 쉘에서는 사용자가 변수를 정의하고 사용할 수 있다. 일반적으로 소문자로.

```
$ projdir=/home_marshall/coolstuff  
$ ls -la $projdir  
$ cd $projdir
```

- 변수는 현재 쉘에서만 사용가능
- 정의한 변수를 현재 쉘뿐 아니라 현재 쉘의 모든 서브 프로세스에서 사용가능하게 하려면 `export` 명령으로 환경변수로 만들어야

```
$ export DEBUG=1
```

- `env` 명령

- `env` 현재 정의되어 있는 모든 환경변수와 그 값을 나열

환경변수 (Environment Variables)

```
pi@raspberrypi:~$ env
...
PWD=/home/pi
HOME=/home/pi
LC_CTYPE=UTF-8
SSH_CLIENT=10.0.1.2 56896 22
SSH_TTY=/dev/pts/0
INFINALITY_FT_BRIGHTNESS=0
MAIL=/var/mail/pi
TERM=xterm-256color
SHELL=/bin/bash
LANGUAGE=en_US.UTF-8
LOGNAME=pi
DBUS_SESSION_BUS_ADDRESS=unix:path=/run/user/1000/bus
XDG_RUNTIME_DIR=/run/user/1000
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/local/games:/usr/games
...
```

❶ PATH

- 사용자가 실행한 명령어의 위치를 찾아볼 디렉토리들을 순서대로 명시

- \$ echo \$PATH

```
/home/acox/bin:/usr/local/bin:/usr/kerberos/bin:/usr/bin:/bin:/usr/X11R6/bin:/bin:/usr/bin
```

```
pi@raspberrypi:~ $ which gcc
/usr/bin/gcc
pi@raspberrypi:~ $ ls -l /usr/bin/*gcc*
lrwxrwxrwx 1 root root      5 Mar 18 2019 /usr/bin/arm-linux-gnueabihf-gcc -> gcc-8
-rwxr-xr-x 1 root root 1004100 Apr 15 2019 /usr/bin/arm-linux-gnueabihf-gcc-8
lrwxrwxrwx 1 root root      5 Mar 18 2019 /usr/bin/gcc -> gcc-8
lrwxrwxrwx 1 root root     25 Apr 15 2019 /usr/bin/gcc-8 -> arm-linux-gnueabihf-gcc-8
...
```

~/.profile

```
pi@raspberrypi:~$ cat .profile
# ~/.profile: executed by the command interpreter for login shells.
# This file is not read by bash(1), if ~/.bash_profile or ~/.bash_login
# exists.
# see /usr/share/doc/bash/examples/startup-files for examples.
# the files are located in the bash-doc package.

# the default umask is set in /etc/profile; for setting the umask
# for ssh logins, install and configure the libpam-umask package.
#umask 022

# if running bash
if [ -n "$BASH_VERSION" ]; then
    # include .bashrc if it exists
    if [ -f "$HOME/.bashrc" ]; then
        . "$HOME/.bashrc"
    fi
fi

# set PATH so it includes user's private bin if it exists
if [ -d "$HOME/bin" ] ; then
    PATH="$HOME/bin:$PATH"
fi
```

\$HOME/bin 디렉토리가 존재하면
환경변수 PATH에 \$HOME/bin을 추가하라.

Package Management

패키지 관리 (Package Management)

- ❶ 패키지 관리란 새로운 소프트웨어를 설치, 업데이트, 삭제하는 일을 말한다.
- ❷ 소프트웨어는 **소스코드의 형태**로 배포되는 경우와 **바이너리 패키지의 형태**로 배포되는 경우가 있다.
- ❸ 소스코드의 경우 보통 하나의 아카이브 파일(tar파일)로 묶은 후 압축하여 배포한다.
- ❹ 바이너리 패키지에는 응용 프로그램, 라이브러리 파일, 버전 정보나 의존성 등의 메타 정보 파일등이 포함된다.
- ❺ 패키지는 저장소(repository)에 저장되어 있으며, HTTP 혹은 FTP 서버를 통해 다운로드 한다.
- ❻ 하나의 패키지가 다른 패키지나 공유 라이브러리 등을 필요로 하는 관계를 패키지 간의 **의존성(dependency)**이라고 부른다.
- ❼ 의존성을 해결하는 기능을 제공하는 것이 리눅스의 패키지 관리 시스템의 주요 역할이다.

High and low-level package tools

- 두 유형의 패키지 관리 도구가 사용됨

- low-level tools: 실제 패키지의 설치, 업그레이드, 삭제 등을 수행
- high-level tools: 의존성의 해결, 패키지 검색 등의 기능을 제공

DISTRIBUTION	LOW-LEVEL TOOL	HIGH-LEVEL TOOL
Debian and derivatives	dpkg	apt-get / aptitude
CentOS	rpm	yum
openSUSE	rpm	zypper

High and low-level package tools

❶ dpkg

- ❷ Debian-기반의 리눅스에서 사용되는 low-level package manager
- ❸ .deb 패키지의 설치와 삭제
- ❹ 하지만 자동으로 패키지를 다운로드하거나 의존성을 해결해주지는 않음

❷ apt-get / apt-cache / apt

- ❷ Debian-기반의 리눅스에서 사용되는 high-level package manager
- ❸ 패키지를 검색, 다운로드, 설치, 의존성 해결
- ❹ 최근의 Debian기반의 리눅스 배포판에는 apt-get과 apt-cache의 기능을 통합한 apt 명령이 설치되어 있다.

❸ aptitude

- ❷ Debian-기반의 리눅스의 또 다른 high-level package manager
- ❸ apt-get 보다 좀 더 개선된 기능을 제공

Software Repositories

- Ubuntu, Debian, CentOS, RHEL 등의 주요 리눅스 배포판들은 각각 자신의 공식 저장소를 유지하고 있으며 거의 매일 업데이트 한다.
- 따라서 리눅스 시스템에 새로운 프로그램을 설치하고자 하면 이 공식 저장소에 필요한 프로그램이 있는지 물어보고 있으면 저장소로 부터 프로그램을 다운로드한다.
- 공식 저장소에 없는 프로그램은 신뢰할 수 있는 사설(third-party) 저장소로부터 다운로드하여 설치할 수도 있다.
- Raspbian의 공식 저장소

저장소의 url은 /etc/apt/sources.list 파일에 저장되어 있다.

Raspbian의 버전명

```
pi@raspberrypi:~ $ cat /etc/apt/sources.list
deb http://raspbian.raspberrypi.org/raspbian/ buster main contrib non-free rpi
# Uncomment line below then 'apt-get update' to enable 'apt-get source'
#deb-src http://raspbian.raspberrypi.org/raspbian/ buster main contrib non-free rpi
pi@raspberrypi:~ $
```

저장소에 패키지들은 main, contrib, non-free 등의 카테고리로
분류되어 있다.

Searching packages

- APT는 키워드를 이용하여 패키지를 검색하는 기능을 제공한다. 예를 들어 다음과 같이 python과 관련이 있는 패키지들을 검색할 수 있다.
(\$ apt-cache search KEYWORD 명령 역시 유사한 기능을 제공한다.)

```
pi@raspberrypi:~ $ apt search python
Sorting... Done
Full Text Search... Done
2to3/stable 3.7.3-1 all
  2to3 binary using python3

accerciser/stable 3.22.0-7 all
  interactive Python accessibility explorer for the GNOME desktop

acme-tiny/stable 1:4.0.4-1+deb10u1 all
  letsencrypt tiny Python client

acr/stable 1.7.2-1 all
  autoconf like tool

adequate/stable 0.15.2 all
  Debian package quality testing tool
  ...
```

Package에 관한 정보 보기

- 특정 패키지에 관한 정보를 볼 수 있다. 가령 gcc 패키지에 관한 자세한 추가 정보를 알고 싶다면:

```
pi@raspberrypi:~ $ apt show gcc
Package: gcc
Version: 4:8.3.0-1+rpi2
Priority: optional
Section: devel
Source: gcc-defaults (1.181+rpi2)
Maintainer: Debian GCC Maintainers <debian-gcc@lists.debian.org>
Installed-Size: 46.1 kB
Provides: c-compiler, gcc-arm-linux-gnueabihf (= 4:8.3.0-1+rpi2)
Depends: cpp (= 4:8.3.0-1+rpi2), gcc-8 (>= 8.3.0-1~)
Recommends: libc6-dev | libc-dev
Suggests: gcc-multilib, make, manpages-dev, autoconf, automake, libtool, flex,
bison, gdb, gcc-doc
Conflicts: gcc-doc (<< 1:2.95.3)
Download-Size: 5,200 B
APT-Manual-Installed: no
APT-Sources: http://raspbian.raspberrypi.org/raspbian buster/main armhf Packages
Description: GNU C compiler
  This is the GNU C compiler, a fairly portable optimizing compiler for C.
  .
  This is a dependency package providing the default GNU C compiler.
pi@raspberrypi:~ $
```

Updating APT Cache

- apt search 혹은 apt show 명령에 의해 제공되는 패키지에 관한 정보들은 실제로는 로컬 캐시에 저장되어 있다.
- 따라서 캐시에 저장된 이 정보들은 apt update 명령으로 주기적으로 업데이트 되어야 한다.
- 이 명령은 캐시에 저장된 패키지에 관한 정보를 업데이트하는 것이며 프로그램 자체를 업데이트하는 것은 아니다. 패키지 자체를 전체적으로 업데이트 하기 위해서는 apt upgrade 명령을 수행해야 한다.

```
pi@raspberrypi:~ $ sudo apt update
Hit:1 http://archive.raspberrypi.org/debian buster InRelease
Get:2 http://raspbian.raspberrypi.org/raspbian buster InRelease [15.0 kB]
Fetched 15.0 kB in 2s (9,729 B/s)
Reading package lists... Done
Building dependency tree
Reading state information... Done
23 packages can be upgraded. Run 'apt list --upgradable' to see them.
pi@raspberrypi:~ $
```

새로운 패키지 설치하기

- 사용중인 리눅스 배포판의 공식 저장소에 저장된 패키지는 `apt install` 명령으로 설치할 수 있다.

```
pi@raspberrypi:~ $ sudo apt install git
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  git-man liberror-perl
...
0 upgraded, 3 newly installed, 0 to remove and 23 not upgraded.
Need to get 5,853 kB of archives.
After this operation, 32.3 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Processing triggers for man-db (2.8.5-2) ...
pi@raspberrypi:~ $
```

- 공식 저장소가 아닌 사설 저장소의 패키지를 설치하기 위해서는 저장소의 url을 `/etc/apt/sources.list.d` 디렉토리에 등록해 주어야 한다.

Package Management

COMMAND	ÅNGSTRÖM	DEBIAN/UBUNTU
Install a package.	opkg install nano	sudo apt-get install nano
Update the package index.	opkg update	sudo apt-get update
Upgrade the packages on your system.*	opkg upgrade	sudo apt-get upgrade
Is nano installed?	opkg list-installed grep nano	dpkg-query -l grep nano
Is a package containing the string nano available?	opkg list grep nano	apt-cache search nano
Get more information about a package.	opkg info nano	apt-cache show nano apt-cache policy nano
Get help.	opkg	apt-get help
Download a package to the current directory.	opkg download nano	sudo apt-get download nano
Remove a package.	opkg remove nano	sudo apt-get remove nano
Clean up old packages.	Nontrivial. Search for "opkg-clean script"	sudo apt-get clean

패키지 이름을 안다면
dpkg -s package

Ubuntu와 Debian의 최신 배포판에서는 apt-get과 apt-cache 대신 그냥 apt 명령을 사용하면 된다.

Python and Virtual Environment

Virtual Environment

- Python은 현재 버전 2와 3가 동시에 사용된다. 또한 Python 응용 프로그램들은 종종 특정한 버전의 라이브러리를 필요로 하는 경우가 있다. 그러므로 시스템에 설치된 하나의 python 환경이 모든 응용의 요구사항을 충족하지 못할 수도 있다.
- 이런 문제를 해결하기 위해 Python 가상환경(virtual environment)을 사용한다. 각각의 가상환경은 특정한 버전의 Python과 추가 패키지들을 포함한다.

Creating Virtual Environments

- ➊ 가상환경을 위치시킬 디렉토리를 찾아가서 다음의 명령을 실행한다.

```
pi@raspberrypi:~ $ python3 -m venv test-env ← 현재 디렉토리에 test-env라는 이름의 가상환경을 생성한다.  
pi@raspberrypi:~ $ cd test-env ← 현재 디렉토리에 test-env라는 이름의 디렉토리가 생성된다.  
pi@raspberrypi:~/test-env $ tree ← test-env 디렉토리에 있는 파일과 서브 디렉토리를 살펴본다.  
.  
├── bin  
│   ├── activate  
│   ├── activate.csh  
│   ├── activate.fish  
│   ├── easy_install  
│   ├── easy_install-3.7  
│   ├── pip  
│   ├── pip3  
│   ├── pip3.7  
│   ├── python → python3 ← test-env 가상환경에서 사용될 Python interpreter이다.  
│   └── python3 → /usr/bin/python3  
├── include  
└── lib  
    └── python3.7  
        └── site-packages ← test-env 가상환경에서 사용될 패키지들은 여기에 설치된다.  
    └── pyvenv.cfg  
    └── share
```

- ➋ 보통 가상 환경은 개발중인 프로젝트 디렉토리 내에 .venv라는 이름으로 생성하는 경우가 많다.

Activating and Deactivating Virtual Environment

가상환경을 활성화하면 프롬프트 앞에 이렇게 가상환경 이름이 표시된다.

Linux 혹은 Mac-OS에서는 이렇게 가상환경을 활성화한다.
(Windows에서는 test-env\Scripts\activate.bat을 실행한다.)

```
pi@raspberrypi:~ $ source test-env/bin/activate
(test-env) pi@raspberrypi:~ $ pip list ← test-env 가상환경 내에 설치된 패키지 목록을 보여준다.
Package           Version
-----
pip                18.1
pkg-resources      0.0.0
setuptools         40.8.0
(test-env) pi@raspberrypi:~ $ python
Python 3.7.3 (default, Dec 20 2019, 18:57:59)
[GCC 8.3.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import sys
>>> sys.path ← 패키지를 검색할 위치들을 저장한다.
['', '/usr/lib/python37.zip', '/usr/lib/python3.7', '/usr/lib/python3.7/
lib-dynload', '/home/pi/test-env/lib/python3.7/site-packages']
>>> quit()
(test-env) pi@raspberrypi:~ $ deactivate
pi@raspberrypi:~ $
pi@raspberrypi:~ $ rm -rf test-env ↑ 가상환경 내의 site-packages 디렉토리가
                                         패키지 검색 경로에 포함된다.

↑ 가상환경을 비활성화한다.
```

가상환경을 제거하려면 이렇게 가상환경 디렉토리를 통채로 삭제한다.

Managing Packages with pip

- ☞ pip이라는 프로그램을 이용하여 Python 패키지를 설치, 업그레이드, 삭제할 수 있다.

```
(test-env) pi@raspberrypi:~ $ pip search astronomy
astronomy (0.0.1)      - Astronomy!
catastropy (0.0dev)    - (cat)astronomy
gastropy (0.0dev)      - (g)astronomy
...
novas (3.1.1.5)        - The United States Naval Observatory NOVAS astronomy library
...
(test-env) pi@raspberrypi:~ $ pip install novas
Looking in indexes: https://pypi.org/simple, https://www.piwheels.org/simple
Collecting novas
  Downloading https://www.piwheels.org/simple/novas/novas-3.1.1.5-cp37-cp37m-
  linux_armv7l.whl (163kB)
    100% |██████████| 163kB 259kB/s
Installing collected packages: novas
Successfully installed novas-3.1.1.5
```

Managing Packages with pip

- 다음과 같이 패키지의 특정 버전을 설치할 수 있다.

```
(test-env) pi@raspberrypi:~ $ pip install requests==2.6.0
Looking in indexes: https://pypi.org/simple, https://www.piwheels.org/simple
Collecting requests==2.6.0
  Downloading https://files.pythonhosted.org/packages/73/63/
b0729be549494a3e31316437053bc4e0a8bb71a07a6ee6059434b8f1cd5f/requests-2.6.0-py2.py3-
none-any.whl (469kB)
    100% |████████████████████████████████| 471kB 409kB/s
Installing collected packages: requests
Successfully installed requests-2.6.0
```

- pip uninstall 명령으로 패키지를 삭제한다.

```
(test-env) pi@raspberrypi:~ $ pip uninstall requests
Uninstalling requests-2.6.0:
  Would remove:
    /home/pi/test-env/lib/python3.7/site-packages/requests-2.6.0.dist-info/*
    /home/pi/test-env/lib/python3.7/site-packages/requests/*
Proceed (y/n)? y
  Successfully uninstalled requests-2.6.0
```

Managing Packages with pip

- pip show 명령으로 설치된 특정 패키지에 관한 정보를 보거나, pip list 명령으로 설치된 패키지 목록을 볼 수 있다.

```
(test-env) pi@raspberrypi:~ $ pip show novas
Name: novas
Version: 3.1.1.5
Summary: The United States Naval Observatory NOVAS astronomy library
Home-page: http://www.usno.navy.mil/USNO/astromonical-applications/software-products/novas
Author: Eric G. Barron
Author-email: eric.barron@usno.navy.mil
License: UNKNOWN
Location: /home/pi/test-env/lib/python3.7/site-packages
Requires:
Required-by:
(test-env) pi@raspberrypi:~ $ pip list
Package      Version
-----  -----
novas        3.1.1.5
pip          18.1
pkg-resources 0.0.0
setuptools   40.8.0
(test-env) pi@raspberrypi:~ $
```