

2.(라즈베리파이)

리눅스 기초

IoT 사물인터넷

학습목표

- 운영체제가 무엇인지 이해할 수 있다.
- VNC를 이용하여 원격 접속하는 방법
- 리눅스 파일시스템, 명령어에 대해서 이해할 수 있다.

1. 운영체제

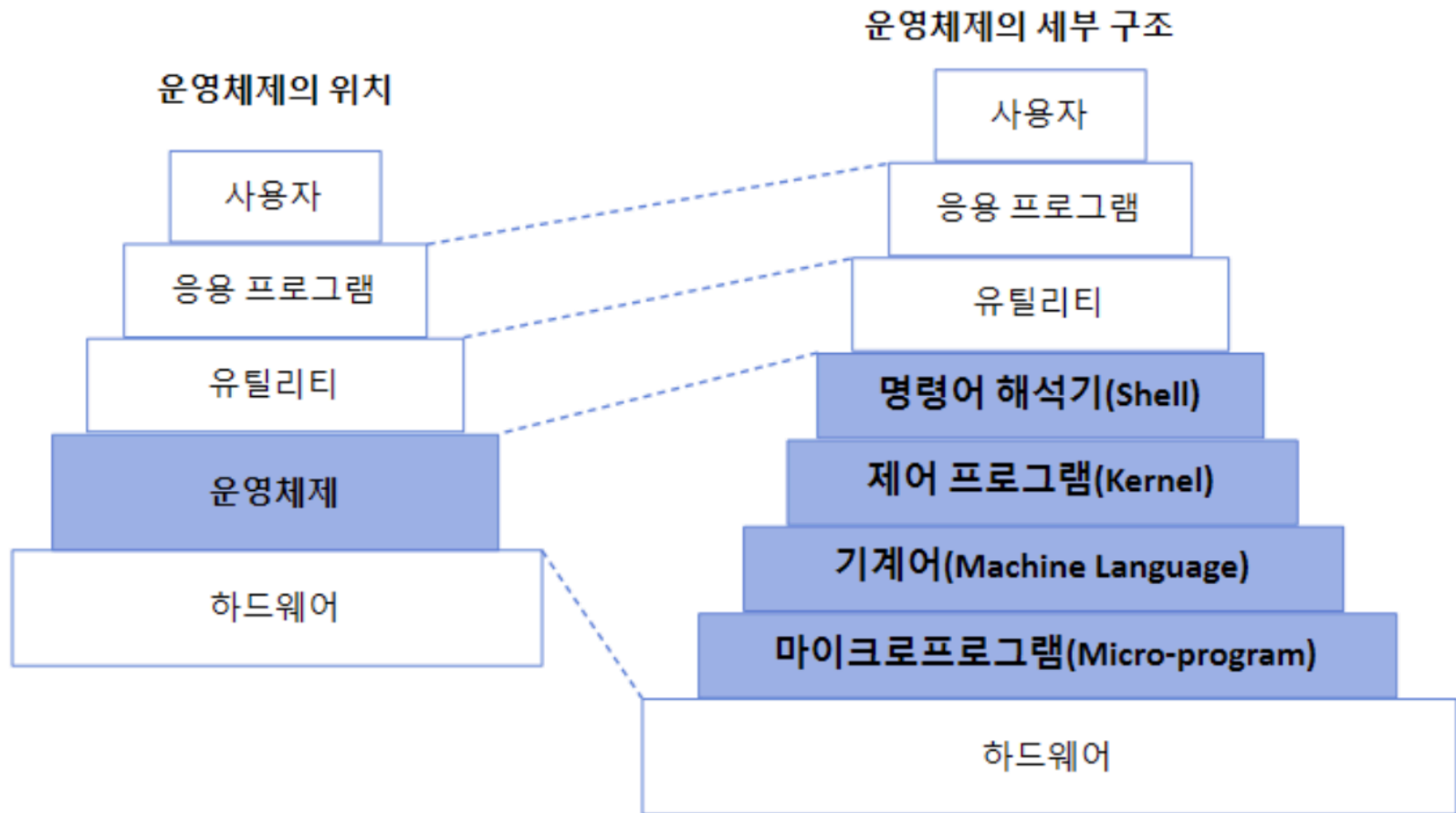
1. 운영체제란?

- 1) 시스템 하드웨어를 관리하고
- 2) 응용 소프트웨어를 실행하기 위한 추상화 플랫폼을 제공하는
시스템 소프트웨어

컴퓨터의 하드웨어 구조와 동작원리를 몰라도 사용 가능하도록 구성되어 있습니다.



1. 운영체제



자원(메모리, CPU, 보조기억장치, 입출력장치)을 관리,
프로세스를 관리하는 특별한 프로그램
운영체제는 프로그램을 위한 프로그램이다.

1. 운영체제

2. 리눅스 운영체제

- 1) 대형기종에 작업이 가능한 유닉스를 개인용 컴퓨터에서 작동할 수 있게 변형한 것
- 2) 리처드스톨먼의 자유소프트웨어 재단의 GNU정신을 준수
: GNU is Not Unix!
- 3) 대표적인 종류
 - : Debian 계열 => *.deb 패키지를 사용하여 소프트웨어 설치 / apt-get / 우분투
 - : Redhat 계열 => *.rpm 패키지를 사용하여 소프트웨어 설치 / yum / CentOS
 - : 라즈베리파이 운영체제인 라즈비안은 데비안 계열로 우분투와 비슷한 점이 많음



2. 라즈베리파이 외부에서 접속하기

1. VNC와 SSH

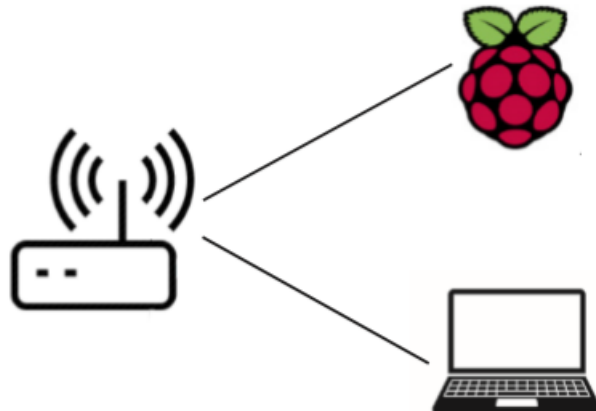
- 1) VNC : Virtual Network Computing
- 2) SSH : Secure Shell Protocol

2. 그래픽 모드 접속

- 1) 윈도우용(외부에서 사용할 OS) VNC Viewer 설치

: <https://www.realvnc.com/en/connect/download/viewer/>

- 2) 접속할 라즈베리파이와 원격 접속하려는 PC가 **같은 네트워크**에 있는지 확인



2. 라즈베리파이 외부에서 접속하기

라즈베리아이콘 - 기본설정 - Raspberry Pi Configuration
: Interfaces탭



Raspberry Pi Configuration

System	Display	Interfaces	Performance	Localisation
Camera:	<input checked="" type="radio"/> Enable	<input type="radio"/> Disable		
SSH:	<input checked="" type="radio"/> Enable	<input type="radio"/> Disable		
VNC:	<input checked="" type="radio"/> Enable	<input type="radio"/> Disable		
SPI:	<input checked="" type="radio"/> Enable	<input type="radio"/> Disable		
I2C:	<input checked="" type="radio"/> Enable	<input type="radio"/> Disable		
Serial Port:	<input type="radio"/> Enable	<input checked="" type="radio"/> Disable		
Serial Console:	<input checked="" type="radio"/> Enable	<input type="radio"/> Disable		
1-Wire:	<input type="radio"/> Enable	<input checked="" type="radio"/> Disable		
Remote GPIO:	<input checked="" type="radio"/> Enable	<input type="radio"/> Disable		

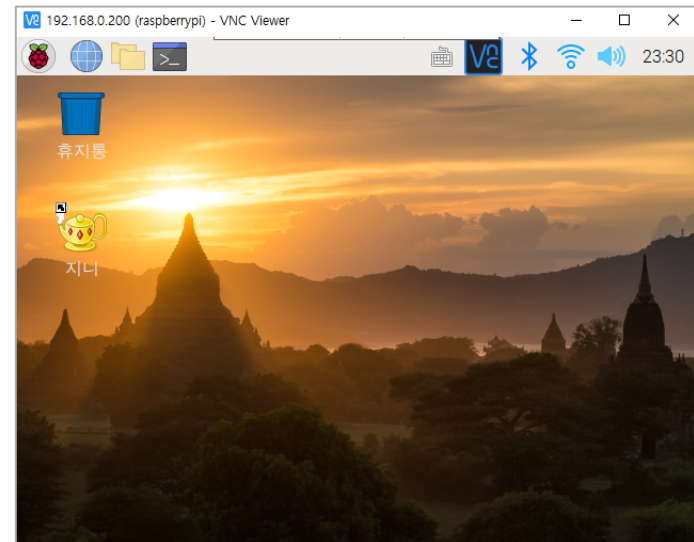
Cancel OK

2. 라즈베리파이 외부에서 접속하기

[아침퀴즈]

VNC(Virtual Network Computing, 가상 네트워크 컴퓨팅)는 컴퓨터 환경에서 **RFB 프로토콜**을 이용하여 원격으로 다른 **컴퓨터**를 제어하는 그래픽 **데스크톱 공유** 시스템이다.

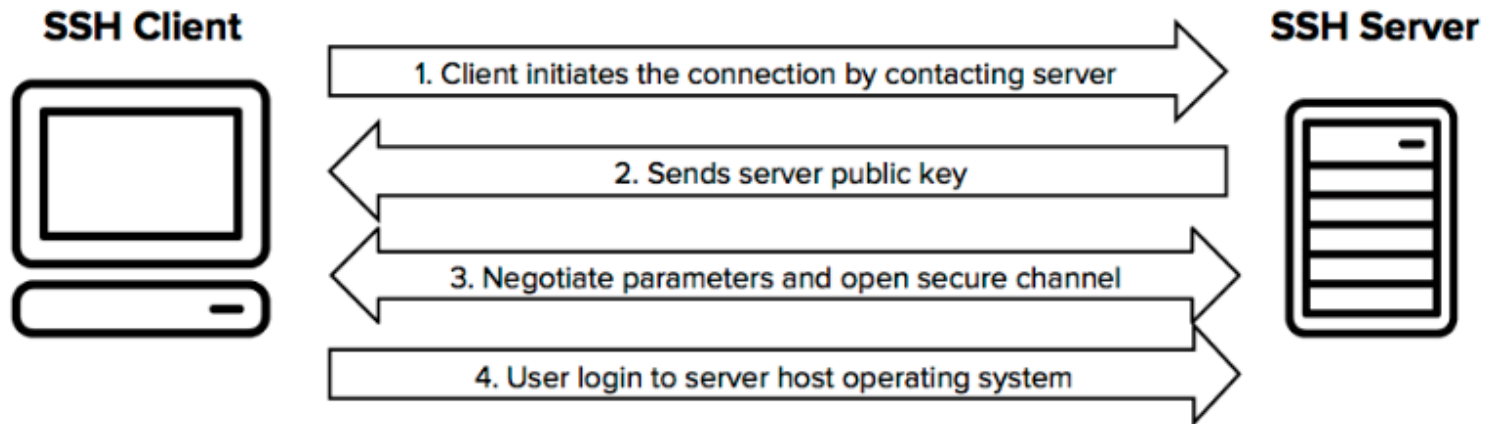
키보드와 **마우스** 이벤트를 한 컴퓨터에서 다른 컴퓨터로 전송시켜서 **네트워크**를 거쳐 그래픽 **화면**을 갱신하는 방식을 제공한다.



2. 라즈베리파이 외부에서 접속하기

[아침퀴즈]

SSH(Secure Shell)는 원격지 호스트 컴퓨터에 접속하기 위해 사용되는 인터넷 프로토콜이다. 기본 포트는 22번이다.



2. 라즈베리파이 외부에서 접속하기

2. 그래픽 모드 접속

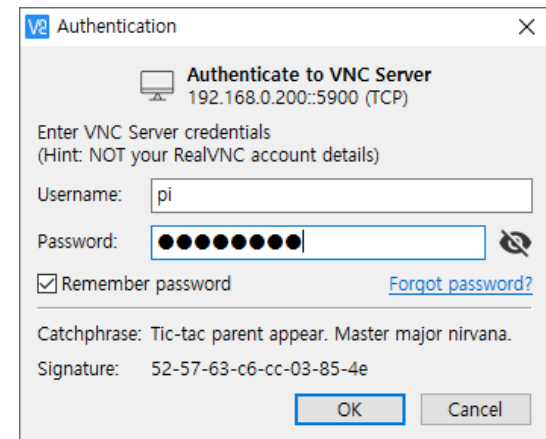
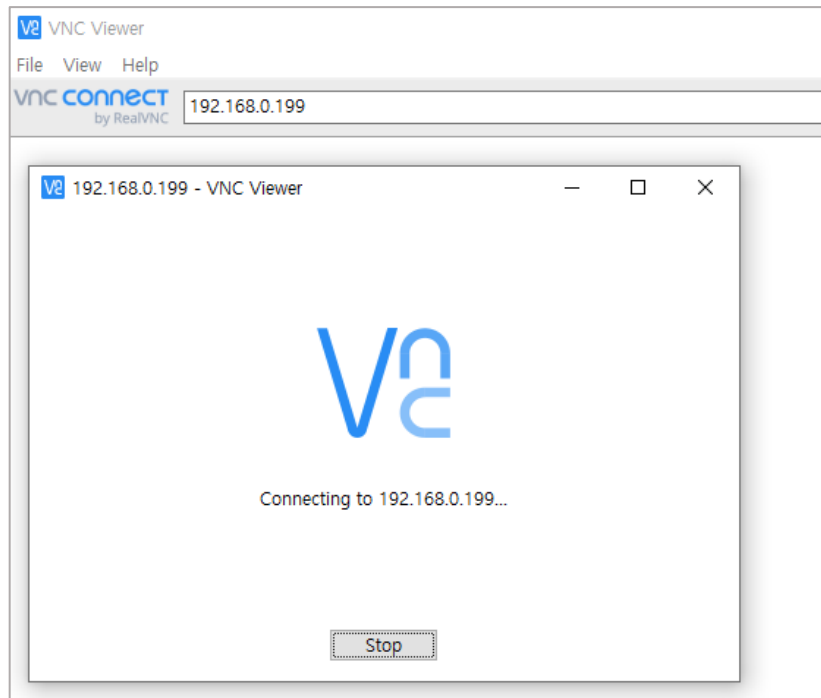
3) 터미널에서 IP확인

: \$ ifconfig

: 앞에서 설정한 고정IP값이 맞는지 확인

: 와이파이 연결 - wlan0, 유선 연결 - eth0

4) 원격 접속을 하려는 PC에서 라즈베리파이의 IP주소를 입력



2. 라즈베리파이 외부에서 접속하기

2. 그래픽 모드 접속

5) 접속이 되지 않는다면 **VNC 포트가 개방되어 있는지 확인**

6) Username과 Password가 기억나지 않으면

: 라즈베리파이 기본 사용자명 - **pi**

: '기본설정 - Raspberry Pi Configuration - System - Change Password'

7) 라즈베리파이에 모니터를 연결하지 않고 부팅 후 VNC를 이용하여 원격접속
할때 Cannot currently show the desktop 문제 발생



Cannot currently show the desktop

: 터미널에서 수동으로 해상도를 설정

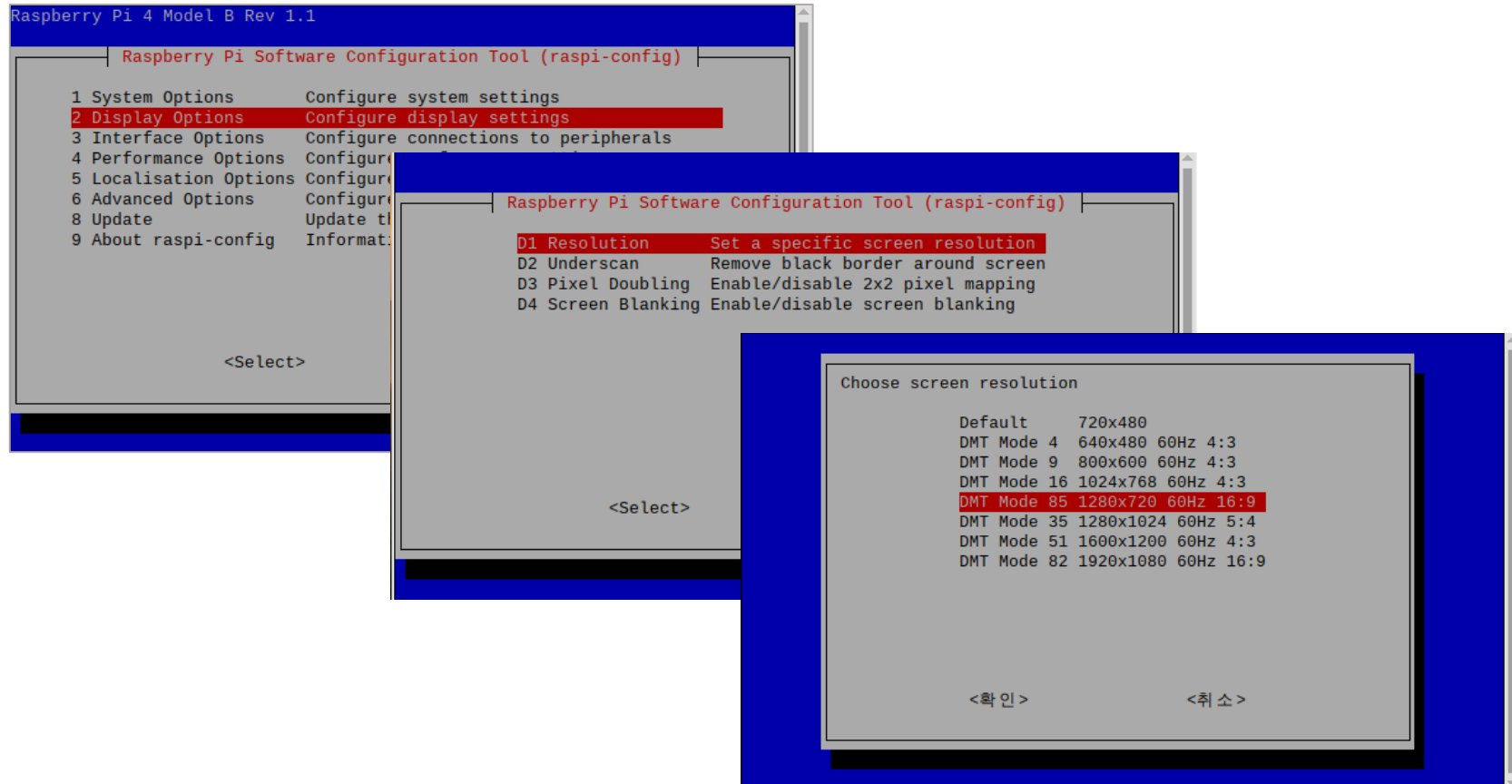
: `$ sudo raspi-config`

: 해상도 설정하고 재부팅

: `$ sudo reboot`

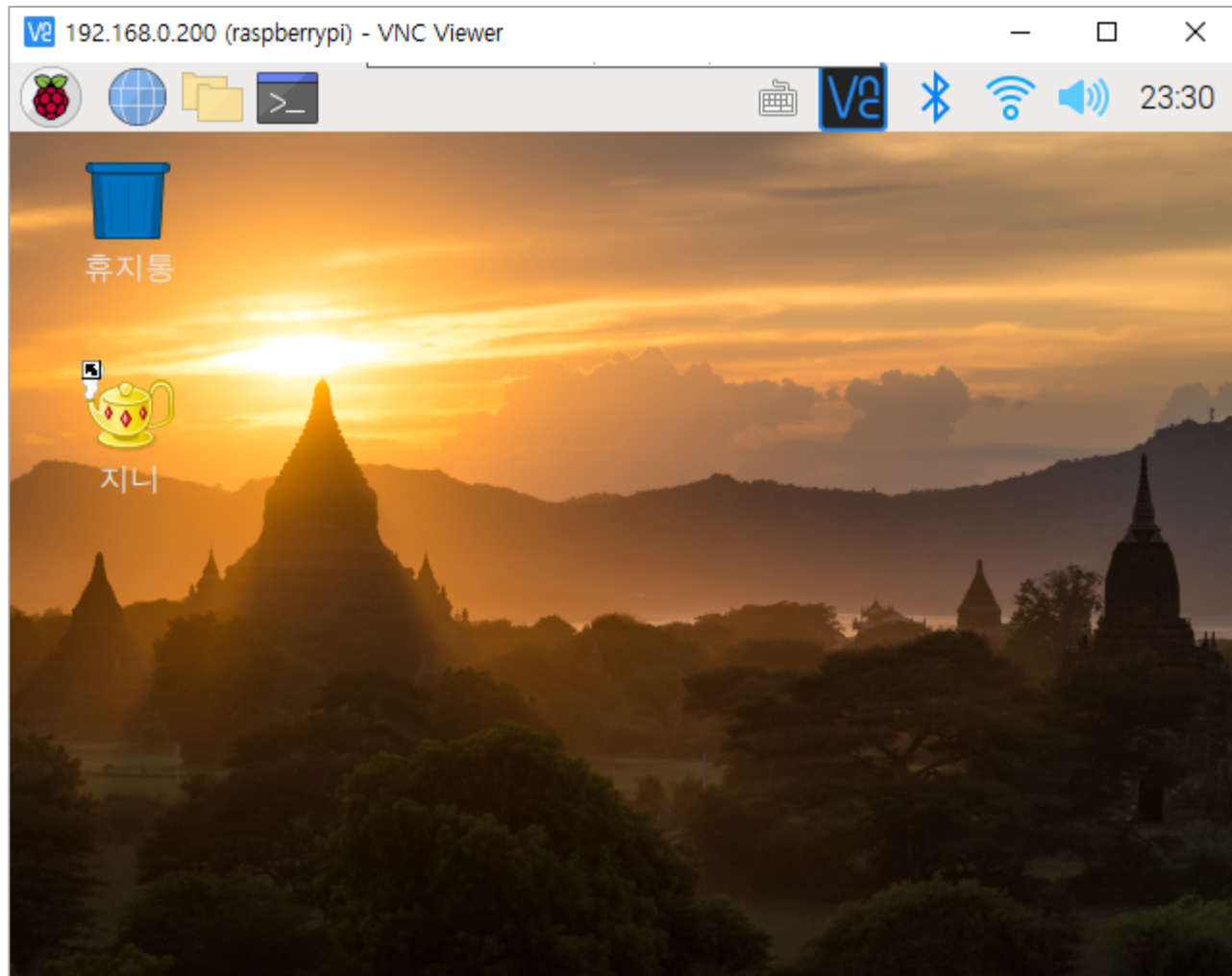
2. 라즈베리파이 외부에서 접속하기

2. 그래픽 모드 접속



2. 라즈베리파이 외부에서 접속하기

2. 그래픽 모드 접속



3. 리눅스 파일 시스템

1. 리눅스 디렉토리 구조

- 1) 리눅스의 기본 디렉토리 구조는 트리 구조를 하고 있음
- 2) 리눅스는 표준 파일 시스템 계층을 사용하기 때문에 같은 목적의 파일들은 같은 장소에 일관되게 모아서 관리함.
- 3) 리눅스 디렉토리 구조

```
pi@raspberrypi: /  
pi@raspberrypi:/ $ ls -l  
합계 64  
lrwxrwxrwx 1 root root 7 1월 11 21:50 bin -> usr/bin  
drwxr-xr-x 4 root root 3584 1월 1 1970 boot  
drwxr-xr-x 4 root root 4096 1월 1 1970 boot.bak  
drwxr-xr-x 17 root root 3840 2월 8 11:28 dev  
drwxr-xr-x 119 root root 4096 2월 8 21:35 etc  
drwxr-xr-x 3 root root 4096 1월 11 21:52 home  
lrwxrwxrwx 1 root root 7 1월 11 21:50 lib -> usr/lib  
drwx----- 2 root root 16384 1월 11 22:14 lost+found  
drwxr-xr-x 3 root root 4096 1월 24 13:48 media  
drwxr-xr-x 2 root root 4096 1월 11 21:50 mnt  
drwxr-xr-x 4 root root 4096 1월 11 21:59 opt  
dr-xr-xr-x 216 root root 0 1월 1 1970 proc  
drwx----- 6 root root 4096 1월 31 15:44 root  
drwxr-xr-x 30 root root 900 2월 9 00:38 run  
lrwxrwxrwx 1 root root 8 1월 11 21:50 sbin -> usr/sbin  
drwxr-xr-x 2 root root 4096 1월 11 21:50 srv  
dr-xr-xr-x 12 root root 0 1월 1 1970 sys  
drwxrwxrwt 18 root root 4096 2월 9 00:18 tmp  
drwxr-xr-x 11 root root 4096 1월 11 21:58 usr  
drwxr-xr-x 11 root root 4096 1월 11 22:15 var
```

3. 리눅스 파일 시스템

1. 리눅스 디렉토리 구조

4) 리눅스 디렉토리 내용 및 기능

디렉토리	내용
/	루트 디렉토리라고 부르는 리눅스 시스템에서 가장 최상위 디렉토리며 디렉토리 구조의 시작입니다 시스템 관리자의 홈인 /root와는 다르며, / 디렉토리 아래에 /bin, /etc, /boot, /mnt, /usr, /lib, /home, /dev, /proc, /var, /sbin, /tmp, /root, /lost+found 등의 디렉토리가 존재합니다
/bin	binaries의 약어로 이진 파일들이며 리눅스에서 가장 기본이 되는 명령어들이 모여 있는 디렉토리입니다 디렉토리의 파일들을 보면 대부분이 실행 파일임을 알 수 있다. 또한 이곳에는 부팅에 필요한 명령어들이 위치하여 부팅후에 시스템의 계정 사용자들이 사용할 수 있는 일반적인 명령어들도 위치 하고 있습니다
/etc	이 디렉토리는 리눅스 시스템에 관한 각종 환경 설정에 연관된 파일들과 디렉토리들을 가진 디렉토리입니다 대부분의 이 디렉토리의 파일들은 시스템 관리자에 의해 관리되는 파일들입니다 웹 서버 환경 설정, 시스템 계정 사용자 정보, 패스 워드 관리, 시스템의 파일 시스템 관리 파일, 여러가지 시스템 보안에 관련된 파일들, 시스템 초기화 설정 파일, TCP/IP 설정 파일 등 시스템 전반에 걸친 거의 모든 환경 설정 파일들이 모두 이 디렉토리에 있습니다
/boot	리눅스 커널이 저장되어 있는 디렉토리로서 각종 리눅스 boot에 필요한 booting지원 파일들이 저장되어 있는 디렉토리입니다
/mnt	외부 장치인 플로피 디스크, 시디롬, 삼바등을 마운트하기 위해서 제공되는 디렉토리입니다 임시로 사용되는 디렉토리이므로 프로그램들은 /mnt에 어떤 파일 시스템이 마운트 되었는지 자동으로 인식하지 않습니다 또한 /mnt는 보통 여러 개의 하위 디렉토리로 나누어 사용되고, 평소에는 각 디렉토리들은 비어 있습니다

3. 리눅스 파일 시스템

1. 리눅스 디렉토리 구조

4) 리눅스 디렉토리 내용 및 기능

디렉토리	내용
/usr	시스템에 사용되는 각종 프로그램들이 설치되는 디렉토리입니다 프로그램과 관련된 명령어 미치 라이브러리들이 이 디렉토리에 위치 하게 됩니다 또한 X 시스템관련 파일들과 리눅스 커널 소스, 각종 C언어 과련 해더 파일 등도 이 디렉토리 안에 저장되어 있습니다
/lib	프로그램들의 각종 라이브러리들이 존재합니다 대부분 공유 라이브러리로 더 편하게 사용할 수 있으며,파일의 크기를 줄여서 실행할 때 불러 사용하게 됩니다 /lib/modules 디렉토리에는 커널로 로딩 가능한 커널 모듈들이 저장되어 있습니다
/home	시스템 계정 사용자들의 홈 디렉토리와 ftp,www,등과 같은 서비스 디렉토리들이 저장됩니다 이곳의 디렉토리와 파일들은 시스템에서 상용되지 않습니다 단지 리모트상에서 시스템으로 접속하는 사용자들을 위한 공간입니다
/dev	디렉토리에는 시스템의 각종 디바이스들에 접근하기 위한 디바이스 드라이버들이 자장되어 있는 디렉토리입니다 이 디렉토리는 물리적인 용량은 갖지 않는 가상 디렉토리입니다 대표적으로는 하드 드라이브,플로피, 씨디롬 그리고 루프팩장치 등이 존재합니다 리눅스 시스템은 윈도우와 달리 각종 디바이스 장치들을 하나의 파일로 취급합니다 따라서 시스템은 각각의 장치들로부터의 정보를 /dev 디렉토리에 존재하는 해당 장치 파일로 부터 가지고 옵니다
/proc	시스템의 각종 프로세서, 프로그램 정보 그리고 하드웨어적인 정보들이 저장된다. 이 디렉토리는 가상 파일 시스템으로 가상 파일 /dev와 마찬가지로 하드 디스크상에 물리적인 용량을 갖지 않습니다 즉 디렉토리에 존재하는 파일들은 실제 하드 디스크에 저장되지 않고 커널에 의해 메모리에 적재 됩니다 디렉토리 안의 파일들은 현재의 시스템 설정을 보여 주는 것입니다

3. 리눅스 파일 시스템

1. 리눅스 디렉토리 구조

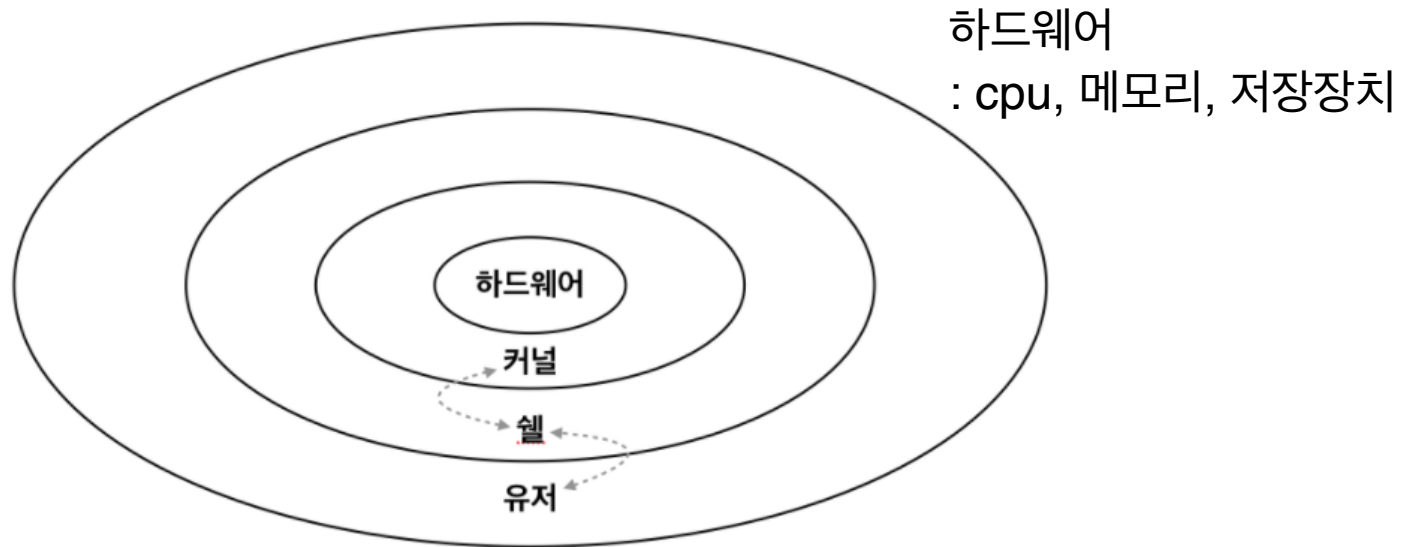
4) 리눅스 디렉토리 내용 및 기능

디렉토리	내용
/var	시스템에서 사용되는 동적인 파일들이 저장됩니다 각종 시스템 로그 파일, 사용자 로그인에 대한 보안기록, 메일서버를 운영한다면 사용자들에게 전송된 메일들을 임시로 저장합니다
/tmp	이름에도 알 수 있듯이 임시 파일들을 위한 디렉토리입니다
/root	시스템 관리자의 홈 디렉토리입니다

4. 리눅스 명령어

1. 커널과 셸 (Kernel, Shell)

- 1) 사용자가 입력한 명령어는 일련의 과정을 거쳐 하드웨어를 제어
- 2) 우리가 명령어를 입력하게 되면 컴퓨터 내부에서는 셸이 받아들이고 명령어를 해석하여 커널을 통해 하드웨어를 조작



4. 리눅스 명령어

2. 커널 (Kernel)

- 1) 소프트웨어와 하드웨어간의 커뮤니케이션을 관리하는 프로그램으로 운영체제의 핵심
- 2) 소프트웨어로부터의 요청을 컴퓨터에 있는 하드웨어(CPU, 메모리, 저장장치 등)가 처리할 수 있도록 요청을 변환하는 역할
- 3) 컴퓨터 부팅시 부트로더에 의해 로드 됨. 항상 메모리에 상주

3. 셸 (Shell)

- 1) 사용자와 운영체제간에 대화를 가능하게 해주는 명령어 해석기
- 2) 하드웨어를 직접적으로 제어하는 것은 어렵기 때문에 하드웨어 제어는 운영체제가 하고, 우리는 운영체제에게 명령을 내림

4. 리눅스 명령어

4. 기본 명령어

1) date

: 현재 날짜 및 시간을 확인

2) hostname

: 호스트의 이름을 확인

3) ls

: 현재 디렉토리안에 존재하는 모든 것을 확인

: \$ ls -l (파일을 상세하게 볼 수 있음)

: \$ ls -a (숨겨진 폴더와 파일 리스트를 확인)

: \$ ls -al (숨겨진 폴더와 파일 리스트를 상세히 확인)

4) clear

: 현재 화면을 깨끗하게 함

5) passwd

: 사용자 비밀번호를 변경

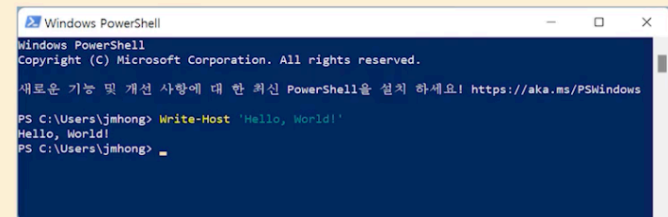
GUI vs CLI

Graphical User Interface



Command Line Interface

- 윈도우: 명령 프롬프트, 파워셸
- 맥OS, 리눅스: 터미널



VSCode의 터미널(Terminal)이 CLI 입니다.

4. 리눅스 명령어

4. 기본 명령어

6) 디렉토리 이동

: \$ cd webapps

: \$ cd ..

7) home 디렉토리 이동

: \$ cd

: \$ cd \$HOME

: \$ cd /home/pi

: \$ cd ~/

8) pwd

: 현재 작업 디렉토리 확인

9) mkdir

: 새로운 디렉토리를 만들

: \$ mkdir movie

4. 리눅스 명령어

4. 기본 명령어

10) touch

: 빈 파일생성

: \$ cd ~/webapps

\$ mkdir ch02

\$ cd ch02

\$ touch read.txt

\$ sudo nano read.txt

Hello word 입력후 (여러줄 입력해 주세요. 20줄이상)

(컨트롤 + X)

A terminal window titled 'pi@raspberrypi: ~/webapps/ch01' with standard menu options (파일(F), 편집(E), 탭(T), 도움말(H)). The prompt is 'pi@raspberrypi:~/webapps/ch01 \$'. The command 'cat led_ex01.py' has been entered, and the output is displayed: 'import RPi.GPIO as GPIO', 'import time', 'GPIO.setmode(GPIO.BOARD)', 'LED = 8', and 'GPIO.setup(LED, GPIO.OUT, initial=GPIO.LOW)'.

```
pi@raspberrypi: ~/webapps/ch01
파일(F) 편집(E) 탭(T) 도움말(H)
pi@raspberrypi:~/webapps/ch01 $ cat led_ex01.py
import RPi.GPIO as GPIO
import time

GPIO.setmode(GPIO.BOARD)

LED = 8

GPIO.setup(LED, GPIO.OUT, initial=GPIO.LOW)
```

11) cat

: 파일의 내용을 화면에 바로 출력

: \$ cd /home/pi/webapps/ch02

\$ cat read.txt

12) head

: 파일의 앞 10줄을 출력

: \$ head read.txt

4. 리눅스 명령어

4. 기본 명령어

13) tail

: 파일의 맨 뒤 5줄을 출력

: `$ tail -5 read.txt`

(마지막 5줄을 출력)


: `$ tail -f /home/pi/webapps/ch02/read.txt` (read.txt파일의 실시간 변경 확인)

: 터미널 2개 열어 Test



The image shows two terminal windows side-by-side. The top window has the title bar 'pi@raspberrypi: ~' and a menu bar with '파일(F) 편집(E) 탭(T) 도움말(H)'. The command prompt shows 'pi@raspberrypi:~ \$ tail -f /home/pi/webapps/ch02/read.txt' with a cursor at the end. The bottom window also has the title bar 'pi@raspberrypi: ~' and the same menu bar. Its command prompt shows 'pi@raspberrypi:~ \$ nano /home/pi/webapps/ch02/read.txt' with a cursor at the end.

: nano 에디터에서 "안녕하세요." 입력하고 저장하기



The image shows a terminal window with the title bar 'pi@raspberrypi: ~' and a menu bar with '파일(F) 편집(E) 탭(T) 도움말(H)'. The command prompt shows 'pi@raspberrypi:~ \$ tail -f /home/pi/webapps/ch02/read.txt'. Below the command, the output '안녕하세요 .' is displayed, followed by a cursor on a new line.

4. 리눅스 명령어

4. 기본 명령어 - 파일 복사, 이동, 삭제 명령어

14)cp

: 파일을 복사해주는 명령어

: \$ cd ~/webapps/ch02

\$ mkdir test

\$ ls

read.txt test

\$ cp read.txt test

(read.txt파일을 test폴더에 복사)

\$ cd test

\$ ls

read.txt

: 복사할 때 다른 이름으로 복사하고 싶으면

: cp 복사할대상 복사될위치/다른파일명

: \$ cd ~/webapps/ch02

\$ ls

read.txt test

\$ cp read.txt test/rename.txt

\$ cd test

\$ ls

read.txt rename.txt

4. 리눅스 명령어

4. 기본 명령어 - 파일 복사, 이동, 삭제 명령어

15) mv

: 파일을 이동하는 명령어

: \$ cd ~/webapps/ch02

\$ touch myfile.txt

\$ mv myfile.txt test (myfile.txt파일을 test폴더로 이동)

\$ cd test

\$ ls

myfile.txt read.txt rename.txt

: mv명령어를 통해 다른 이름으로 이동하고 싶으면

: mv 이동할대상 이동할위치/다른파일명

: \$ cd ~/webapps/ch02/test

\$ ls

myfile.txt read.txt rename.txt

\$ mv myfile.txt /home/pi/webapps/ch02/myfile2.txt

\$ cd ..

\$ ls

myfile2.txt read.txt test

4. 리눅스 명령어

4. 기본 명령어 - 파일 복사, 이동, 삭제 명령어

16)rm

: 파일을 삭제하는 명령어

: \$ cd ~/webapps/ch02

\$ ls

myfile2.txt read.txt test

\$ rm myfile2.txt

: 삭제할때 삭제확인 과정을 거치지 않음

\$ rm -f myfile2.txt

\$ ls

read.txt test

: 디렉토리를 삭제하고 싶을때 -r 옵션 사용

: \$ cd ~/webapps/ch02

\$ ls

read.txt test

\$ rm -r test

\$ ls

read.txt

4. 리눅스 명령어

4. 기본 명령어 - 파일 찾기

17) which

: 명령어의 위치를 찾아주는 명령어

: \$ which python
/usr/bin/python

: \$ which ls
/usr/bin/ls

```
pi@raspberrypi:~ $ whereis python
python: /usr/bin/python3.7-config /usr/bin/python /usr/bin/python3.7m /usr/bin/python2.7 /usr/bin/python3.7 /usr/bin/python3.7m-config /usr/bin/python2.7-config /usr/lib/python2.7 /usr/lib/python3.7 /etc/python /etc/python2.7 /etc/python3.7 /usr/local/lib/python2.7 /usr/local/lib/python3.7 /usr/include/python3.7m /usr/include/python2.7 /usr/include/python3.7 /usr/share/python /usr/share/man/man1/python.1.gz
```

18) whereis

: 명령어의 바이너리(실행파일), 소스, 매뉴얼 파일의 위치를 찾아주는 명령어

: \$ whereis python

: -b 바이너리 파일만 출력

: -m 매뉴얼 파일만 출력

: -s 소스 파일만 출력

: \$ whereis -m python

```
pi@raspberrypi:~ $ whereis -m python
python: /usr/share/man/man1/python.1.gz
```

4. 리눅스 명령어

4. 기본 명령어 - 파일 찾기

19) find

: 파일 및 디렉토리를 검색할 때 사용하는 명령어

: -name 옵션 - 지정된 문자열 패턴에 해당하는 파일 검색

: `$ sudo find / -name read.*` (최상단 '/' 디렉토리부터 파일명 read를 검색)

: `$ sudo find / -name python`

: sudo를 붙이는 이유는 pi사용자가 아닌 root사용자의 권한을 대행 받아 모든 폴더를 탐색하기 위함 (실습 - sudo 안 붙이고 실행)

20) grep

: grep: 파일 내에서 특정 패턴을 검색하는 명령어입니다.

: `$ grep 'hello' file.txt`

: 'file.txt' 파일 내에서 "hello"라는 패턴을 검색합니다.

: `$ grep 'hello' *` (현재 디렉토리내에 있는 모든 파일에서 'hello' 검색)

4. 리눅스 명령어

4. 기본 명령어 - 파일 찾기

21) top

: 현재 실행 중인 프로세스를 모니터링하는 명령어입니다. "top" 명령어를 실행하면 CPU, 메모리 등의 자원 사용 상태를 실시간으로 확인할 수 있습니다.

```
top - 16:54:16 up 5:35, 2 users, load average: 0.21, 0.22, 0.20
Tasks: 185 total, 1 running, 184 sleeping, 0 stopped, 0 zombie
%Cpu(s): 0.7 us, 6.3 sy, 0.0 ni, 93.0 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
MiB Mem : 3660.3 total, 2810.0 free, 296.7 used, 553.7 buff/cache
MiB Swap: 100.0 total, 100.0 free, 0.0 used. 3210.8 avail Mem
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
205	root	20	0	0	0	0	S	29.7	0.0	23:55.21	w1_bus_master1
4107	root	20	0	0	0	0	I	1.0	0.0	0:00.25	kworker/3:2-events
2982	pi	20	0	235804	48772	30020	S	0.7	1.3	0:07.53	node
4161	pi	20	0	8164	3160	2588	R	0.7	0.1	0:00.66	top
484	nobody	20	0	5312	2244	2060	S	0.3	0.1	0:00.27	thd
2600	root	20	0	0	0	0	I	0.3	0.0	0:01.79	kworker/1:2-events
3808	root	20	0	0	0	0	I	0.3	0.0	0:00.47	kworker/0:0-events

4. 리눅스 명령어

5. 권한 설정

- 1) 리눅스는 하나의 컴퓨터를 여러 사람이 사용할 수 있는 멀티유저 운영체제. 때문에 특정 파일이나 디렉토리를 생성한 뒤 그 파일에 접근할 수 있는 권한을 설정하여 다른 유저가 접근할 수 없게 할 수 있음.
- 2) file1.txt, file2.txt 파일을 생성하여 살펴보기

```
$ cd ~/webapps/ch02
$ sudo touch file1.txt
$ touch file2.txt
$ mkdir folder
$ ls -l
-rw-r--r-- 1 root root 0  9월  8 11:27 file1.txt
-rw-r--r-- 1 pi   pi   0  9월  8 11:27 file2.txt
drwxr-xr-x 2 pi   pi   4096 9월  8 11:29 folder
```

4. 리눅스 명령어

5. 권한 설정

```
-rw-r--r-- 1 root root 0  9월  8 11:27 file1.txt  
-rw-r--r-- 1 pi   pi   0  9월  8 11:27 file2.txt  
drwxr-xr-x 2 pi   pi   4096 9월  8 11:29 folder
```

2) file1.txt, file2.txt 파일을 생성하여 살펴보기

① -

• 파일의 타입

- 일반파일	d 폴더	l 링크파일
--------	------	--------

② rw-r--r--

• 퍼미션 정보

r	read 권한	4
w	write 권한	2
x	execute 권한	1
-	권한없음	0

소유자	rw-	6	읽고 쓰는 권한
소유그룹	r--	4	읽기 권한
모든유저	r--	4	읽기 권한

4. 리눅스 명령어

```
-rw-r--r-- 1 root root 0  9월  8 11:27 file1.txt  
-rw-r--r-- 1 pi   pi   0  9월  8 11:27 file2.txt  
drwxr-xr-x 2 pi   pi   4096 9월  8 11:29 folder
```

- rwx rw- r--

소유자

소유그룹

게스트
(기타사용자)

파일타입

- normal file
- d** directory
- l** link
- p** named pipe
- s** socket
- c** character device
- b** block device

r - 읽기 (4)
w - 쓰기 (2)
x - 실행 (1)

$rw x = 4 + 2 + 1 = 7$
 $rw - = 4 + 2 = 6$
 $r - - = 4$

4. 리눅스 명령어

5. 권한 설정

```
-rw-r--r-- 1 root root 0  9월  8 11:27 file1.txt  
-rw-r--r-- 1 pi   pi   0  9월  8 11:27 file2.txt  
drwxr-xr-x 2 pi   pi   4096 9월  8 11:29 folder
```

2) file1.txt, file2.txt 파일을 생성하여 살펴보기

- : 소유자 및 소유그룹이 root이고, 다른 모든 유저는 읽기 권한만 있음 (r--)
- : pi 사용자로 해당파일을 쓰는 것은 불가능
- : nano 에디터를 이용해 file1.txt를 읽고 수정해보기
- : `$ cd ~/webapps/ch02`
- `$ nano file1.txt`
- : "hello Pi"를 작성하고 파일저장 가능 ??

3) chmod 명령어를 이용하여 file1.txt 권한 변경하기

- : chmod 수정번호 파일명 (예: `$ chmod 646 file1.txt`)
- : 수정 권한이 없을 땐 앞에 sudo를 붙여준다.
- : 현재 사용자는 pi임으로 sudo를 붙여 수정
- : `$ sudo chmod 646 file1.txt`
- `$ ls -l`
- ```
-rw-r--rw- 1 root root 0 9월 8 11:27 file1.txt
```

## 4. 리눅스 명령어

## 5. 권한 설정

```
-rw-r--rw- 1 root root 0 9월 8 11:27 file1.txt
-rw-r--r-- 1 pi pi 0 9월 8 11:27 file2.txt
drwxr-xr-x 2 pi pi 4096 9월 8 11:29 folder
```

### 3) chmod 명령어를 이용하여 file1.txt 권한 변경하기

: file1.txt 파일 수정하기

: \$ nano fil1.txt

: "hello Pi"를 작성하고 파일저장 가능 ??

### 4) 파일권한 살펴보기

: --- (0)

: --x (1)

: -w- (2)

: r-- (4)

: 예) r-x - (5), rw - (6), rwx - (7)

## 4. 리눅스 명령어

---

### 5. 권한 설정

#### 5) chown 소유자, 소유그룹 변경 명령어

: chown 소유자:소유그룹 파일명or디렉토리명

: \$ sudo chown root:pi file1.txt

\$ ls -l

-rw-r--rw- root pi 0 9월 8 11:27 file1.txt

```
-rw-r--rw- 1 root pi 0 9월 8 11:27 file1.txt
-rw-r-xr-- 1 pi pi 0 9월 8 11:27 file2.txt
drwxr-xr-x 2 pi pi 4096 9월 8 11:29 folder
```

# 4. 리눅스 명령어

---

## 6. 입출력 지정 IO Redirection

- 1) 리다이렉션은 표준 입출력의 방향을 바꿈.
- 2) 일반적으로 표준입력 - 키보드, 표준출력 - 모니터
- 3) 파일로 출력이 필요할 때 사용
- 4) ' >> '

: 파일에 내용을 출력

: \$ cd ~/webapps/ch02

\$ ls >> readme.txt

\$ nano readme.txt

: \$ ls -al >> readme.txt

\$ nano readme.txt

: 기존에 출력된 내용 밑으로 추가 출력

- 5) ' > '

: \$ ls -a > readme.txt

\$ nano readme.txt

: 기존 내용을 삭제하고 새롭게 내용 작성

## 4. 리눅스 명령어

### 7. 파일 다운로드(및 설치)

1) 파일을 다운 받고 설치를 하려면 자신의 아키텍처를 파악해야 한다.

2) 데비안 GUN/리눅스가 지원하는 주요 아키텍처

: 라즈베리파이 4 Model B는 armhf

: 아키텍처 확인 / `$ dpkg --print-architecture`

: CPU 정보확인 / `$ cat /proc/cpuinfo`

: OS 정보확인 / `$ grep . /etc/*-release`

| 아키텍처             | 데비안의 명칭 | 서브 아키텍처               | 기종         |
|------------------|---------|-----------------------|------------|
| 인텔 x86 기반        | i386    |                       |            |
| AMD64 및 인텔 64    | amd64   |                       |            |
| ARM              | armel   | 인텔 IXP4xx             | ixp4xx     |
|                  |         | Marvell Kirkwood      | kirkwood   |
|                  |         | Marvell Orion         | orion5x    |
|                  |         | Versatile             | versatile  |
| ARM, 하드웨어 FPU 포함 | armhf   | 멀티플랫폼                 | armmp      |
|                  |         | LPAE 가능 시스템에 대한 멀티플랫폼 | armmp-lpae |
| 64비트 ARM         | arm64   |                       |            |

### 7. 파일 다운로드(및 설치)

#### 3) wget을 이용한 다운로드

: GUI기반의 웹탐색기를 사용하지 않고 터미널을 이용하여 파일을 다운받는 방법

: Light Table 이라는 코드 편집기 다운 받기

: `$ wget http://github.com/Automattic/simplenote-electron/releases/download/`

`v1.1.7/Simplenote-1.1.7-amd64.deb`

: 다운로드 실행한 폴더에서 확인

## 7. 파일 다운로드(및 설치)

### 4) apt-get을 이용한 다운로드 및 설치

- : apt-get이란 프로그램 패키지를 다운로드 하고 설치하는 명령어
- : 데비안이 제공하는 공식 저장소를 사용
- : 저장소 주소 확인
- : cat /etc/apt/sources.list
- : 브라우저에서 <http://raspbian.raspberrypi.org/raspbian/> 으로 이동
- : 예) scrot (화면 캡처 프로그램) 저장소 경로 확인 및 설치
- : <http://raspbian.raspberrypi.org/raspbian/pool/main/s/scrot/>
- : apt-get을 이용하여 다운 및 설치
- : \$ sudo apt-get install scrot
- : scrot 사용방법
- : \$ scrot -s
- : 명령어 입력한 뒤 드래그를 하면 home/pi경로에 자동으로 사진이 저장



## 4. 리눅스 명령어

---

### 7. 파일 다운로드(및 설치)

#### 5) git clone을 이용한 다운로드

- : GitHub는 버전 제어 및 공동 작업을 위한 코드 호스팅 플랫폼
- : 오픈소스 및 라이브러리 제공. GitHub를 통해 많은 소스 다운
- : `$ git --version` (Git 설치확인)  
`git version ****`
- : 만약 없으면 다운로드 및 설치
- : `$ sudo apt-get install git`
- : git clone을 이용한 다운로드
- : 예) 온도센서(DHT11)를 구동하는 소스코드 다운
- : `$ git clone https://github.com/szazo/DHT11_Python.git`
- : clone을 실행한 폴더에서 확인

## 8. 파일 압축

### 1) tar, tar.gz

: tar - 파일과 폴더를 하나의 파일로 묶어주는 유틸. 압축은 아님.

: tar.gz - 파일과 폴더를 묶어서 압축

: tar 묶음

```
$ cd ~/webapps/ch02
```

```
$ mkdir aaa
```

```
$ tar -cvf aaa.tar aaa
```

(aaa 폴더를 aaa.tar로 묶어 준다.)

: tar 해제

```
$ tar -xvf aaa.tar
```

(현재 경로에 aaa.tar파일을 풀어준다.)

: tar.gz 압축

```
$ cd ~/webapps/ch02
```

```
$ mkdir ccc
```

```
$ tar -zcvf ccc.tar.gz ccc
```

(ccc 폴더를 ccc.tar.gz로 압축)

: tar.gz 압축해제

```
$ tar -zxvf ccc.tar.gz
```

(현재 경로에 ccc.tar.gz파일 압축 풀기)