

OpenCV를 활용한 이미지 처리

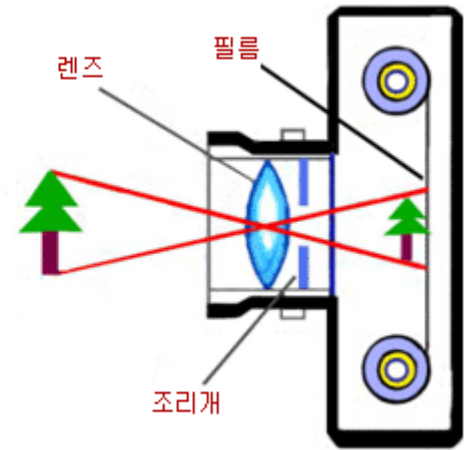
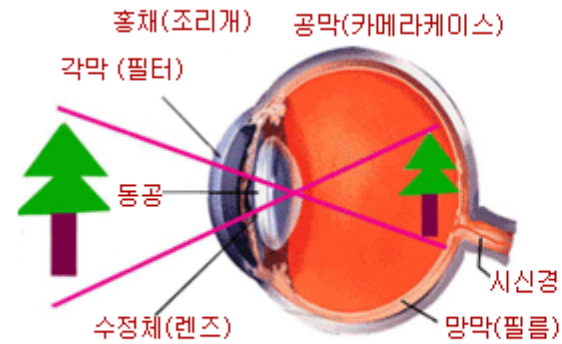
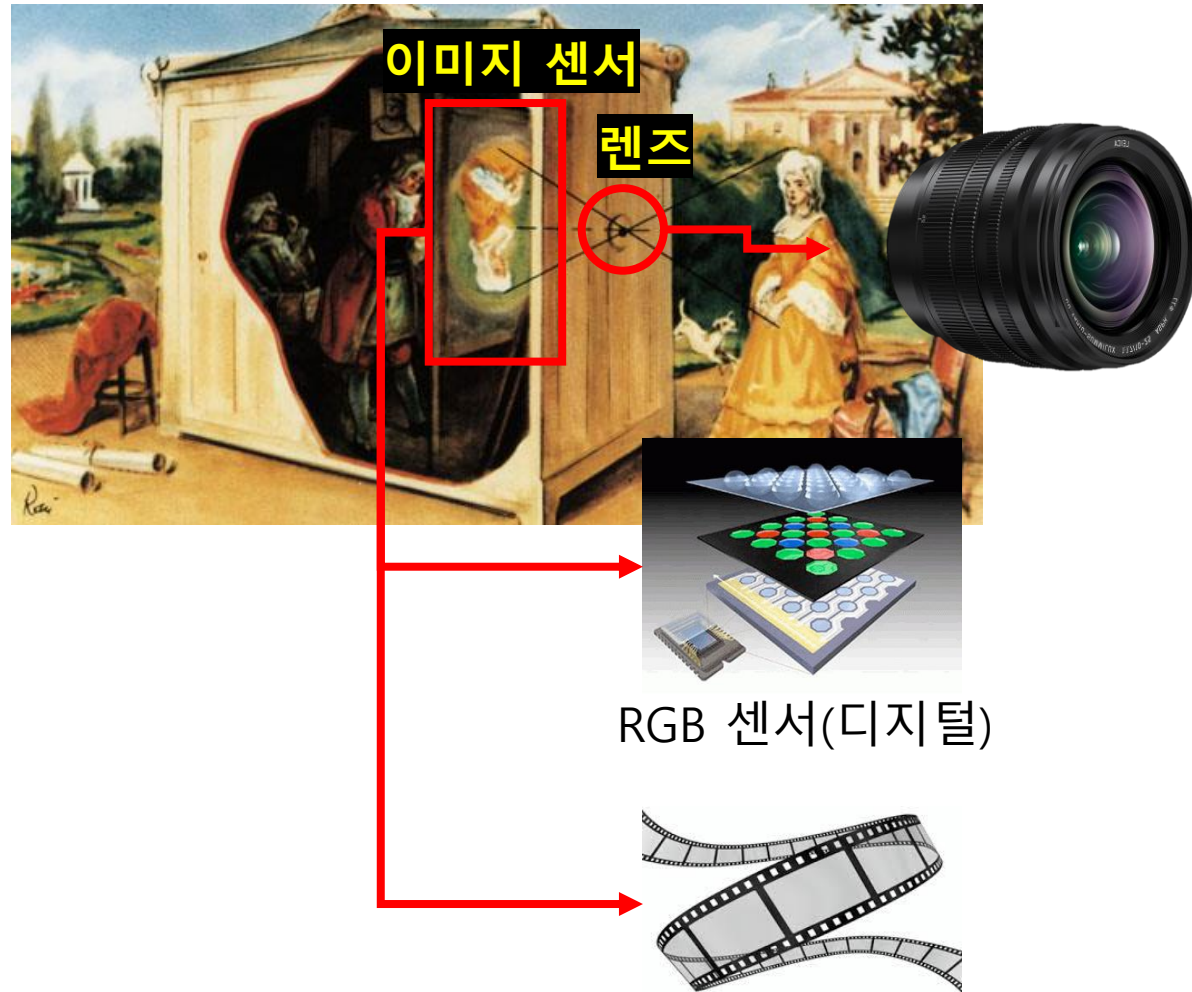


이미지를 어떻게 얻을까?

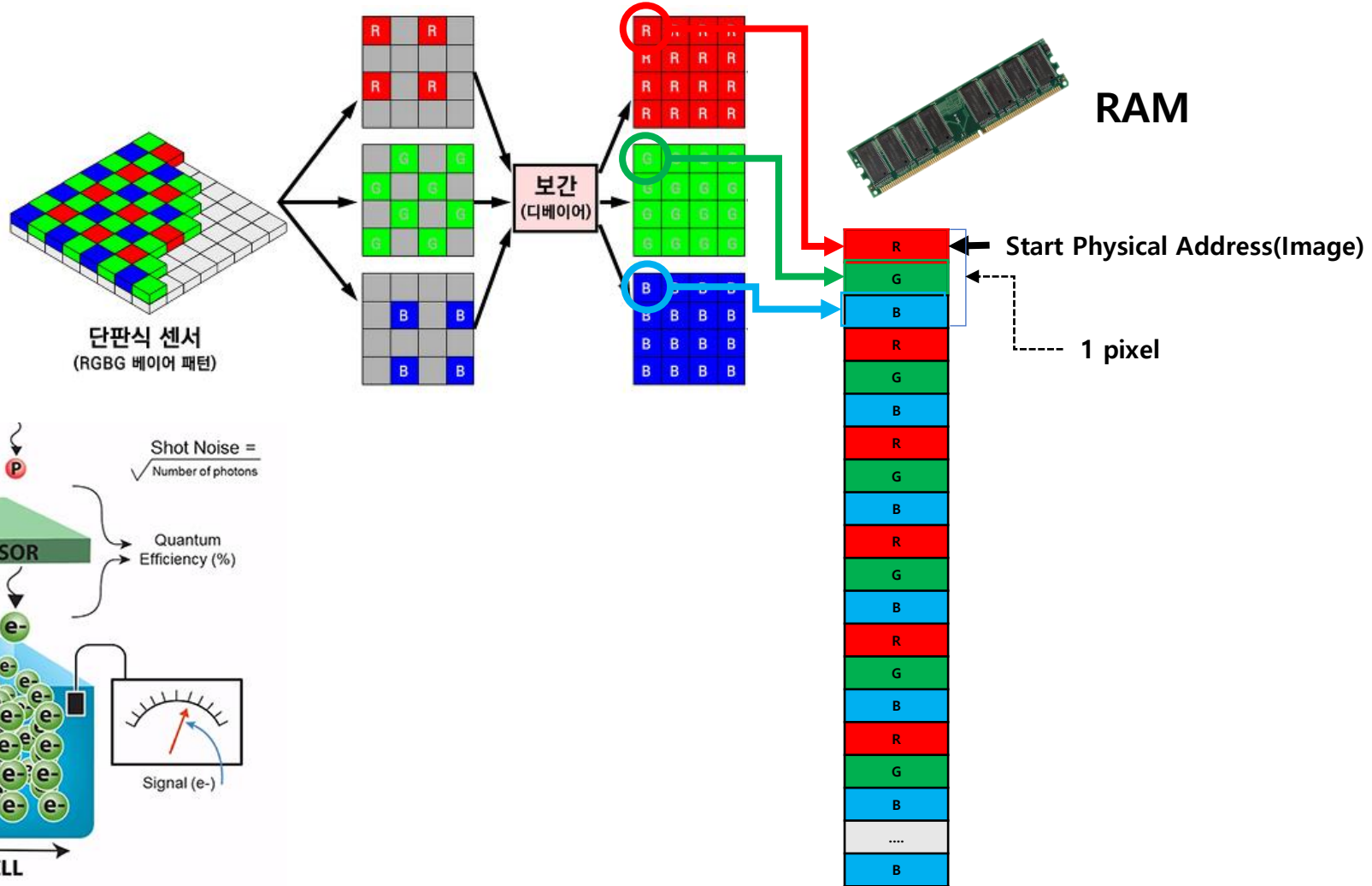
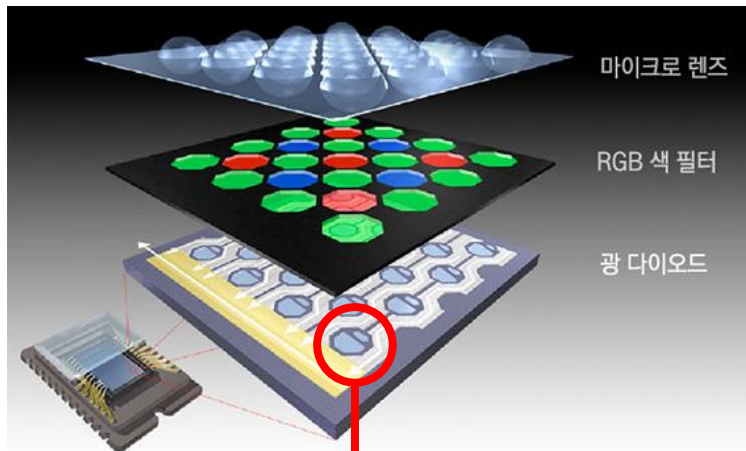


카메라 옵스큐라
(라틴어(camera obscura) 어두운방 이라는 뜻)

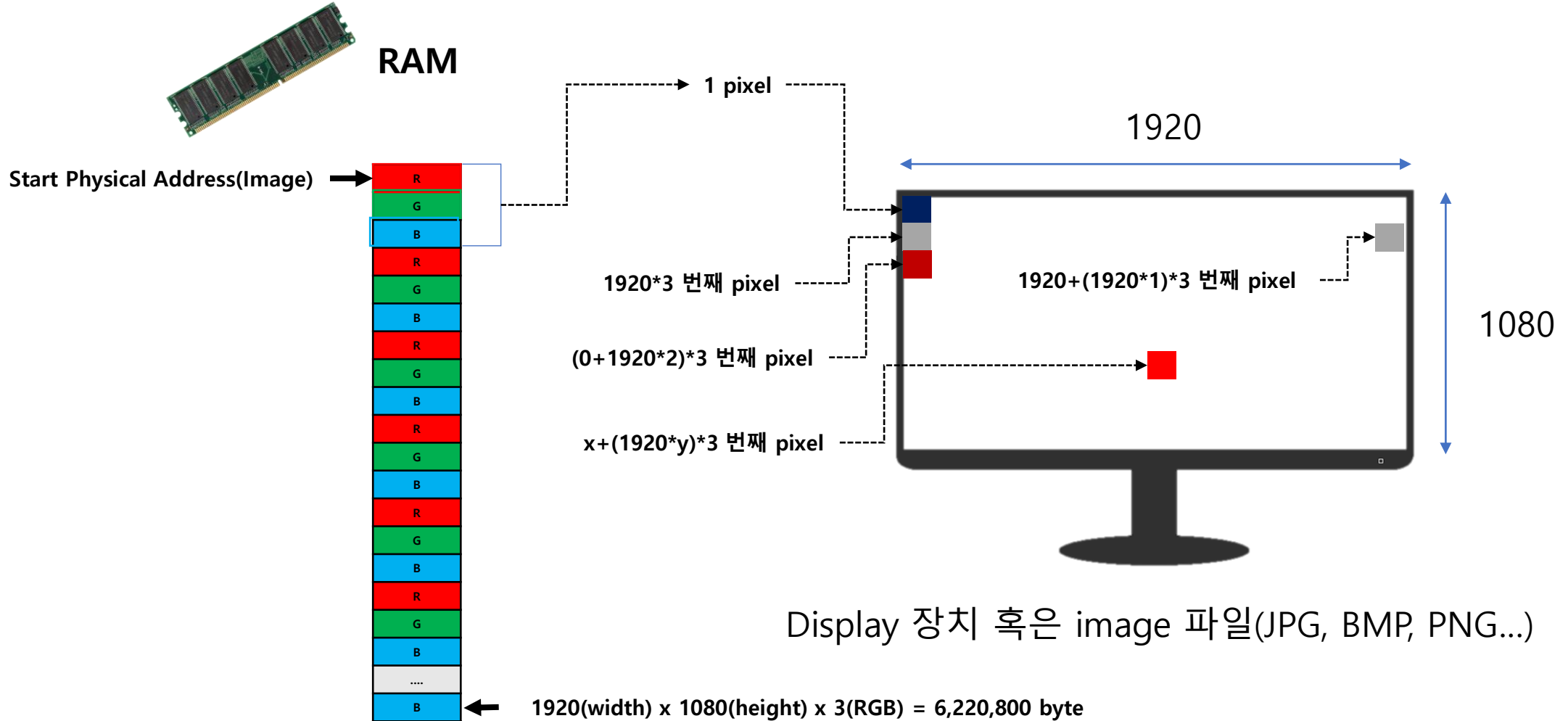
컴퓨터(카메라)는 어떻게 이미지를 얻을까?



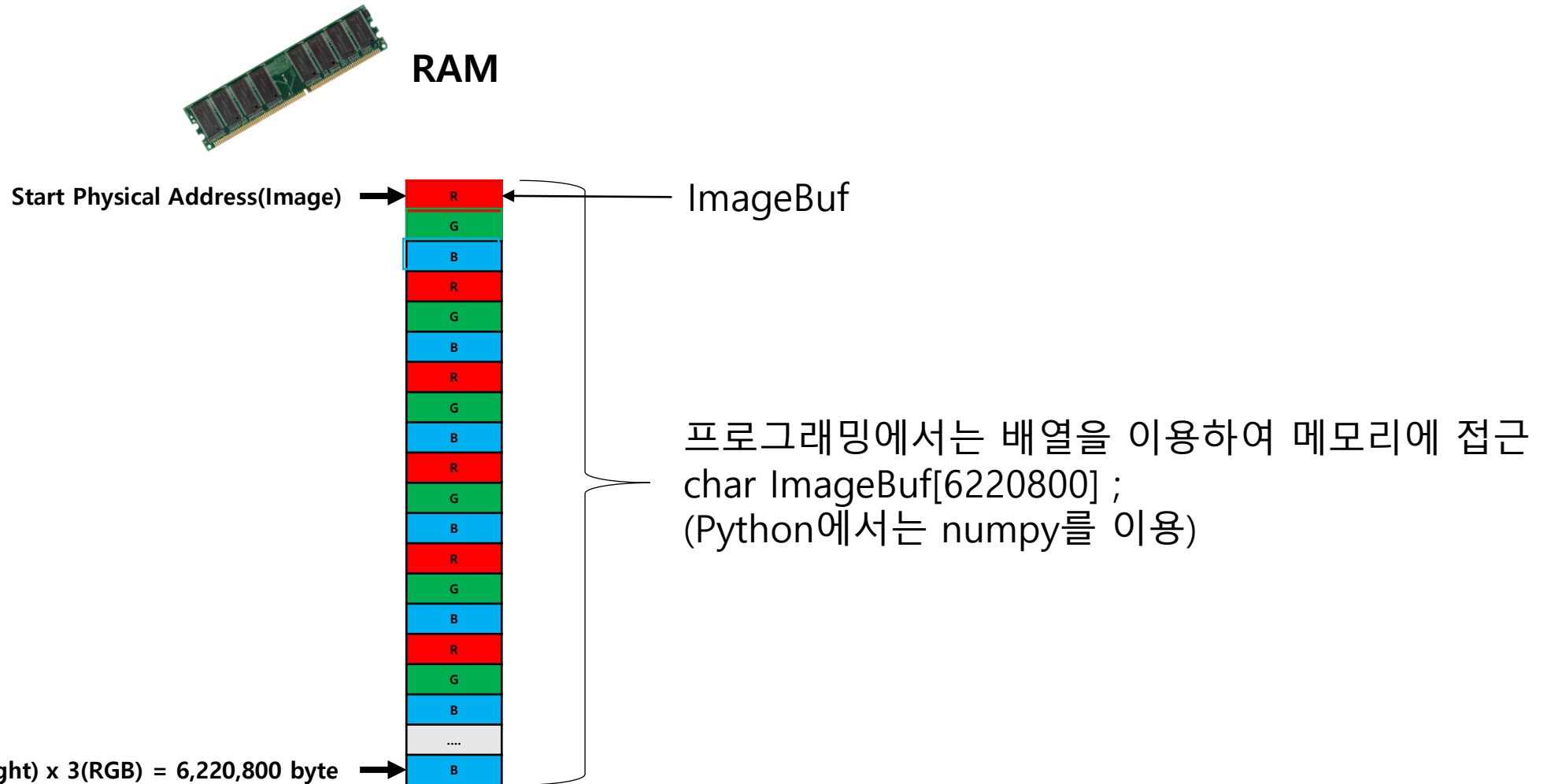
컴퓨터(카메라)는 어떻게 이미지를 얻을까?



컴퓨터(카메라)는 어떻게 이미지를 표시 할까?

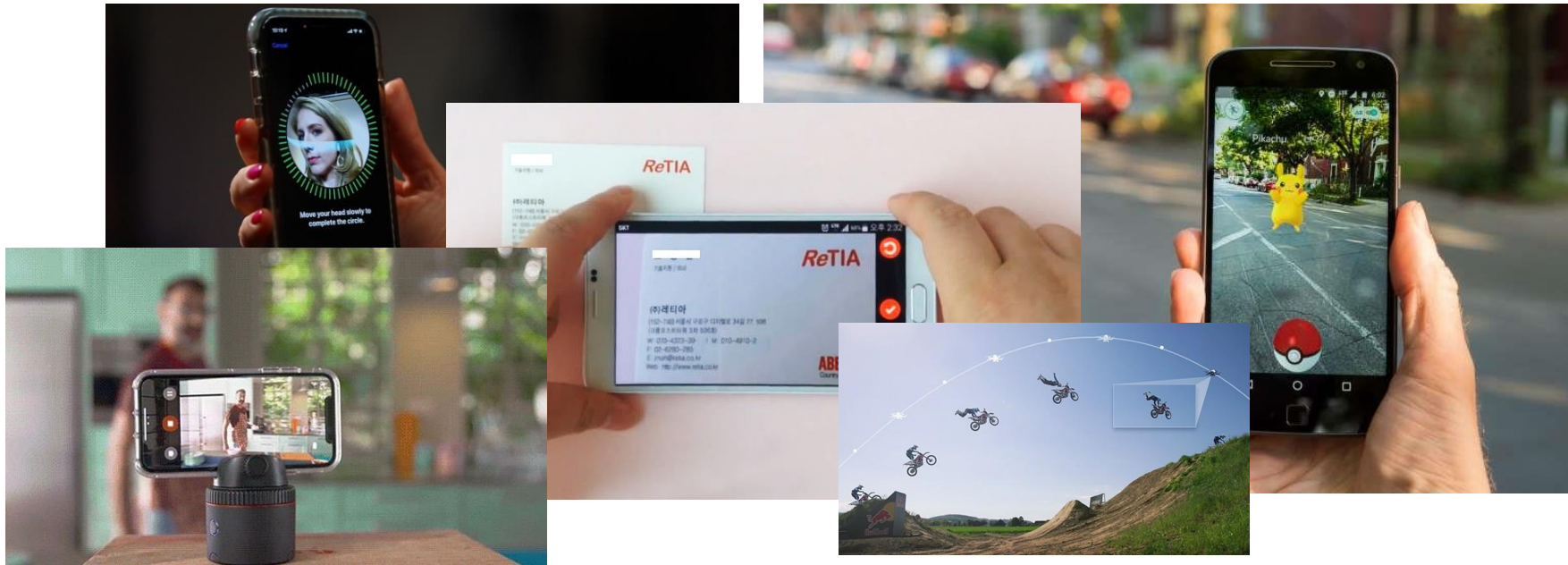


우리는 어떻게?



이제 무얼, 어떻게 하지?

- 영상처리를 이용한 서비스를 만들고 싶은데.



- 그럼 무엇보다 해야 하지?

OpenCV

- OpenCV : Opensource Computer Vision
 - <https://opencv.org/>
 - Open Source : OpenCV is open source and released under the BSD 3-Clause License. It is free for commercial use.
 - Optimized : OpenCV is a highly optimized library with focus on real-time applications.
 - Cross-Platform : C++, Python and Java interfaces support Linux, MacOS, Windows, iOS, and Android.
- Computer Vision 알고리즘을 오픈소스로 공개, 최신의 알고리즘을 가장 빠르게 적용
- 특별한 경우가 아니라면 거의 모든 Computer Vision의 기본
- OpenCV의 함수를 이용하여 편리하게 이미지 데이터에 접근하고 영상처리가 가능
 - 기본 데이터 컨테이너 : `cv::Mat`

Ubuntu + OpenCV 개발 환경 만들기

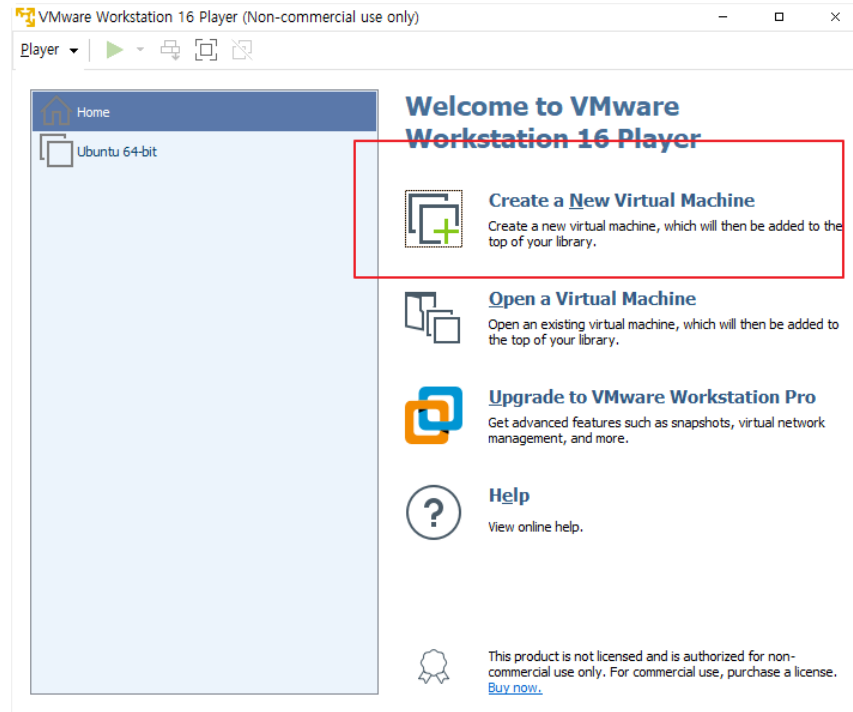
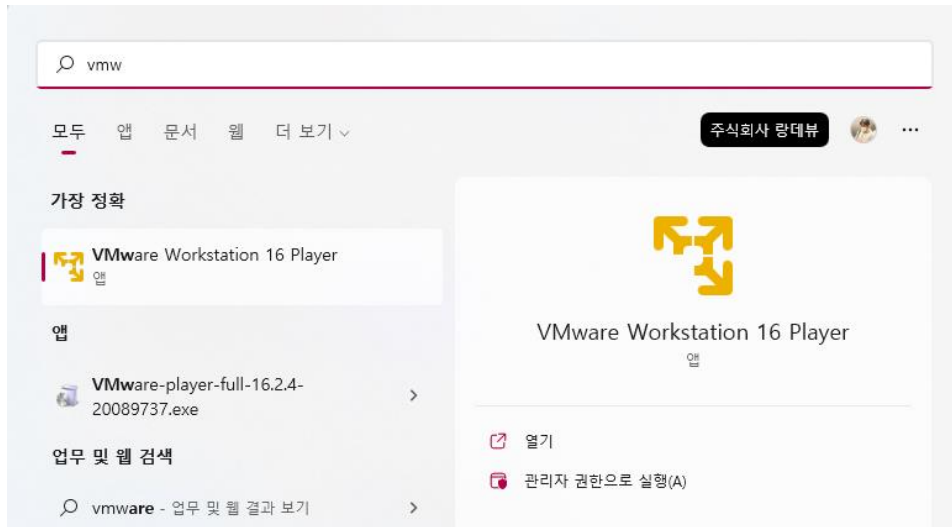
- 우리에게 익숙한 OS는 윈도우즈!
- 그러나 OpenCV개발환경으로는 Ubuntu를 추천!
- 윈도우에 가상 머신을 이용하여 Ubuntu를 설치하자.
- 윈도우에 가상의 컴퓨터는 만들어 주는 SW : VMWare

<https://www.vmware.com/products/workstation-player.html>

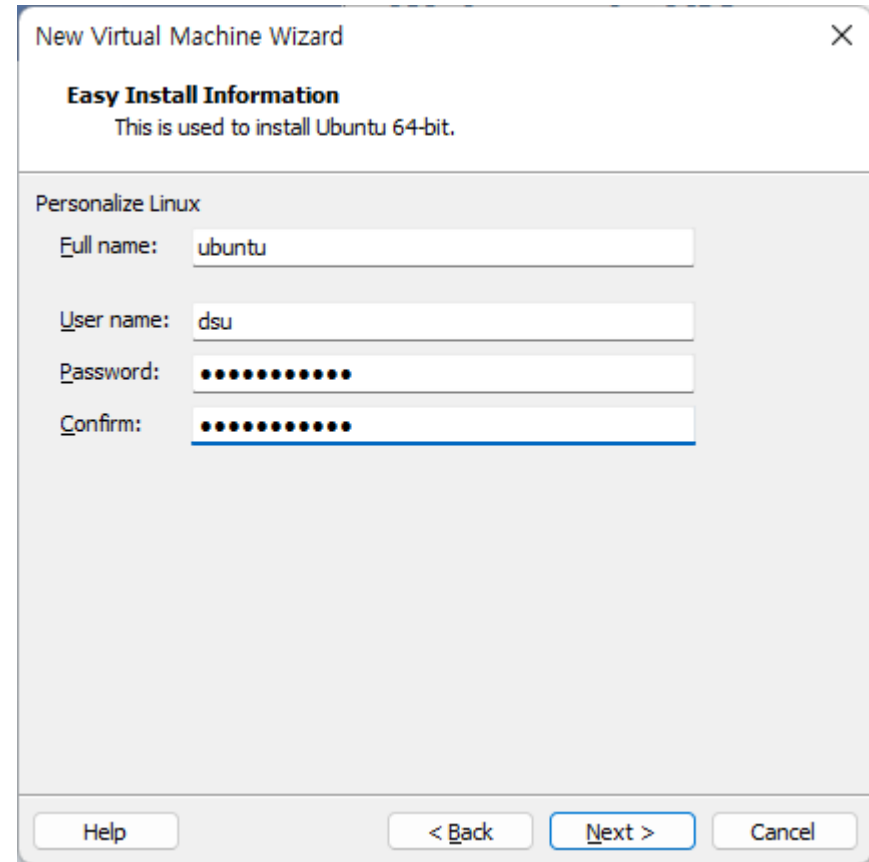
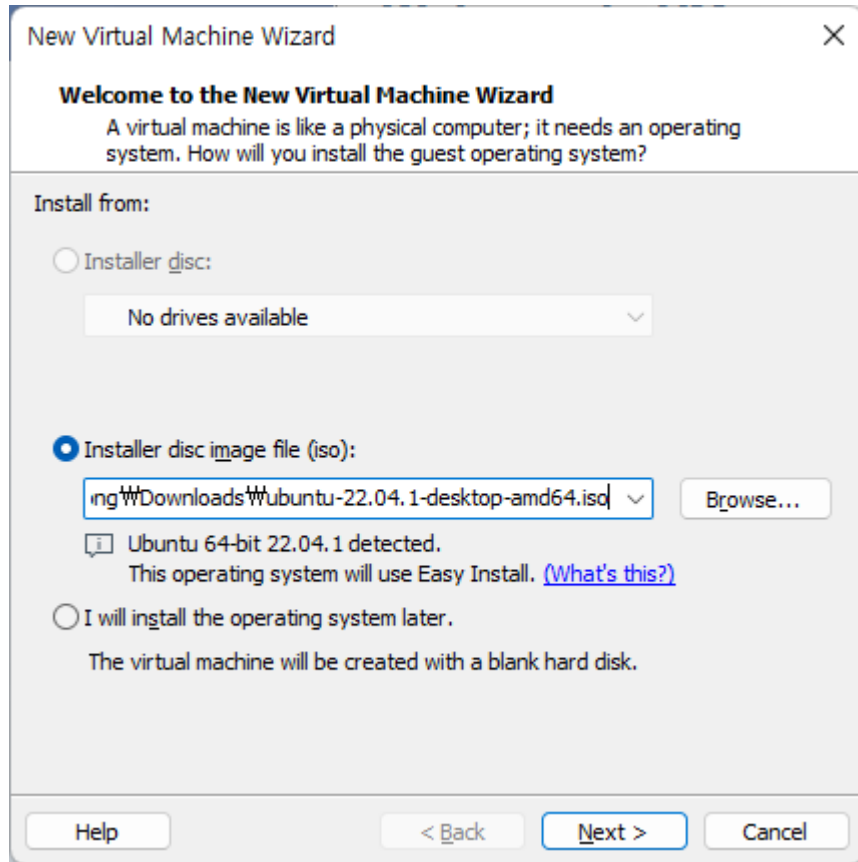
- VMWare에 Ubuntu를 설치

<https://ubuntu.com/download/desktop>

Vmware에 Ubuntu 설치



Vmware에 Ubuntu 설치



Vmware에 Ubuntu 설치

New Virtual Machine Wizard

Name the Virtual Machine
What name would you like to use for this virtual machine?

Virtual machine name:
Ubuntu 64-bit (2)

Location:
- 주식회사 랑데뷰\문서\Virtual Machines\Ubuntu 64-bit (2) Browse...

< Back Next > Cancel

New Virtual Machine Wizard

Specify Disk Capacity
How large do you want this disk to be?

The virtual machine's hard disk is stored as one or more files on the host computer's physical disk. These file(s) start small and become larger as you add applications, files, and data to your virtual machine.

Maximum disk size (GB): 100

Recommended size for Ubuntu 64-bit: 20 GB

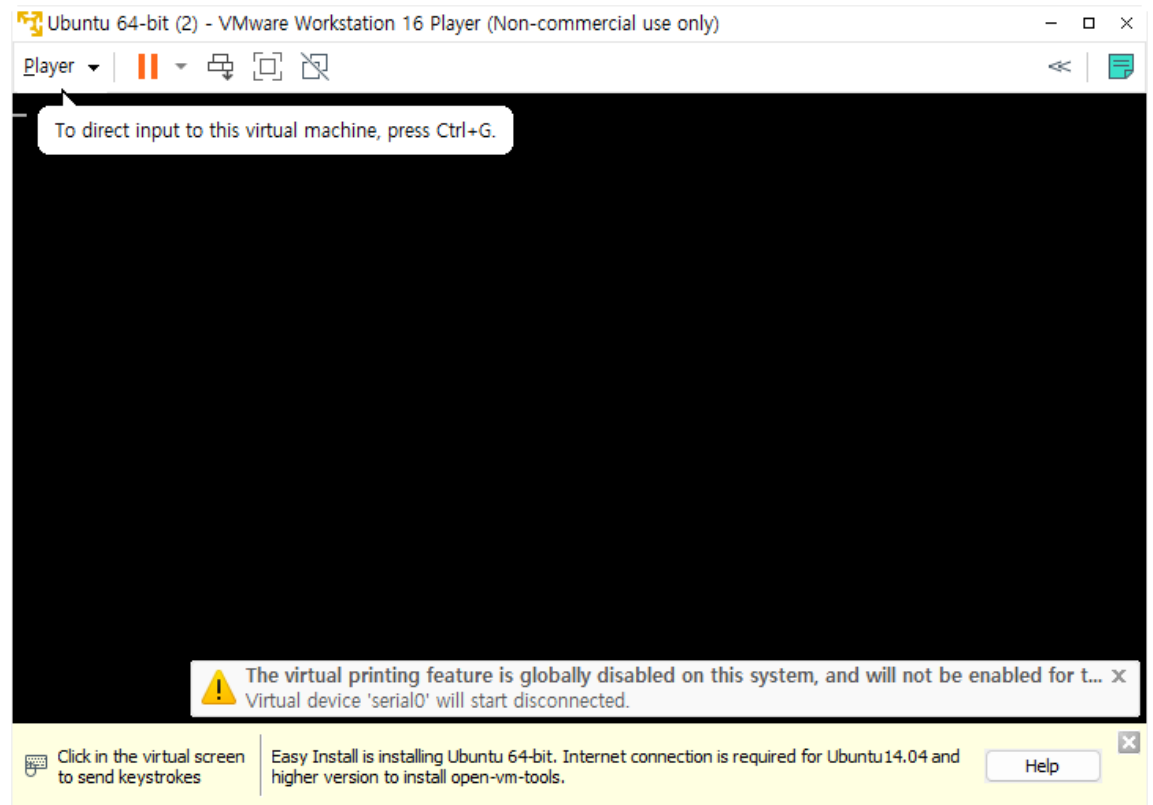
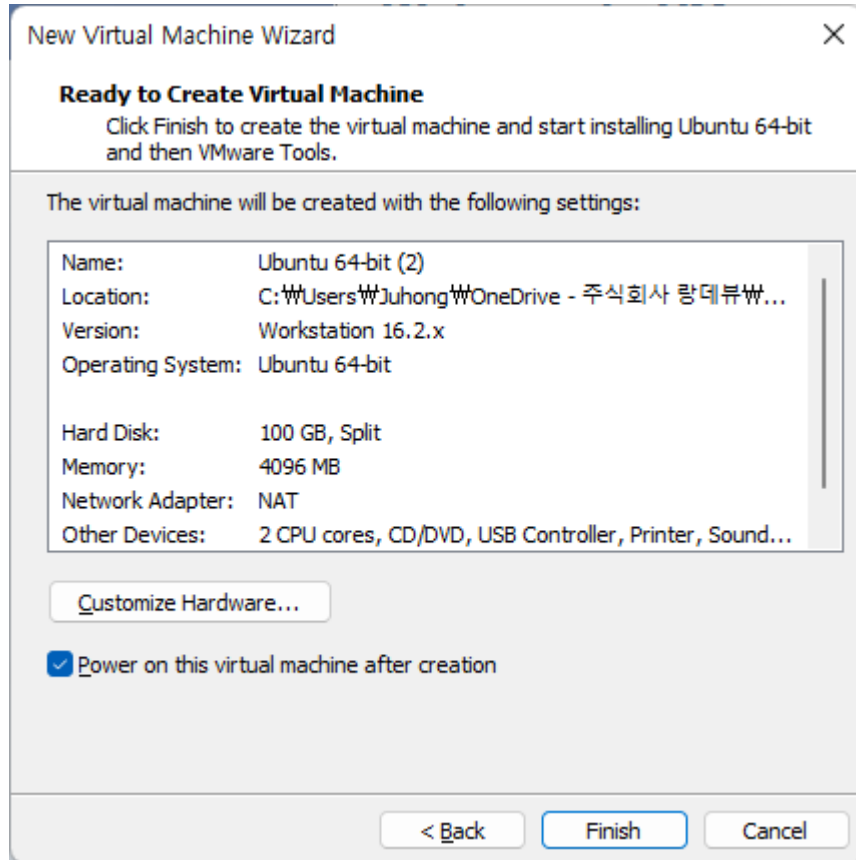
☐ Store virtual disk as a single file

☒ Split virtual disk into multiple files

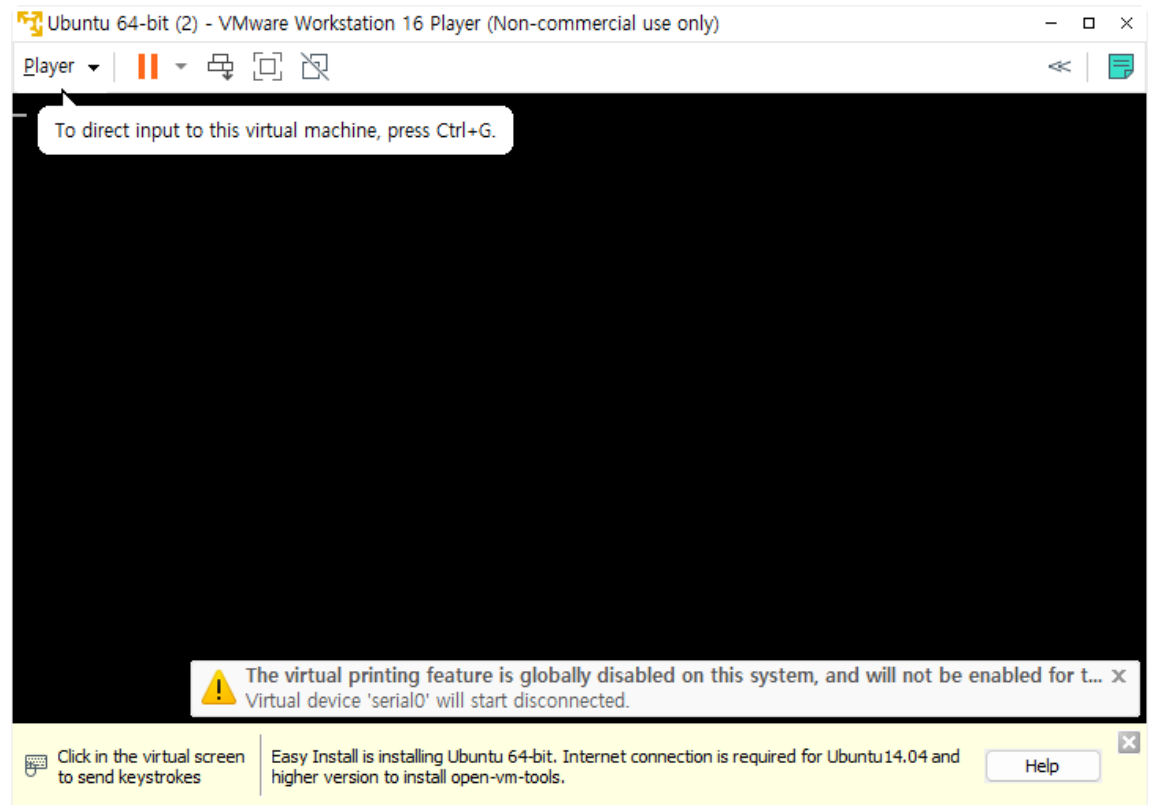
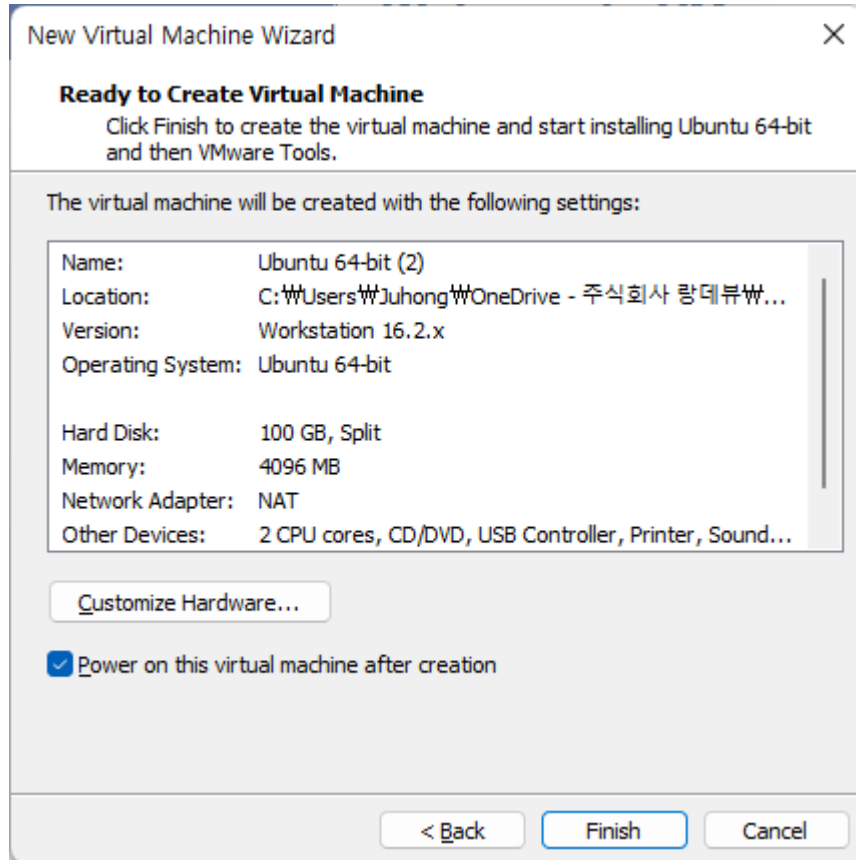
Splitting the disk makes it easier to move the virtual machine to another computer but may reduce performance with very large disks.

Help < Back Next > Cancel

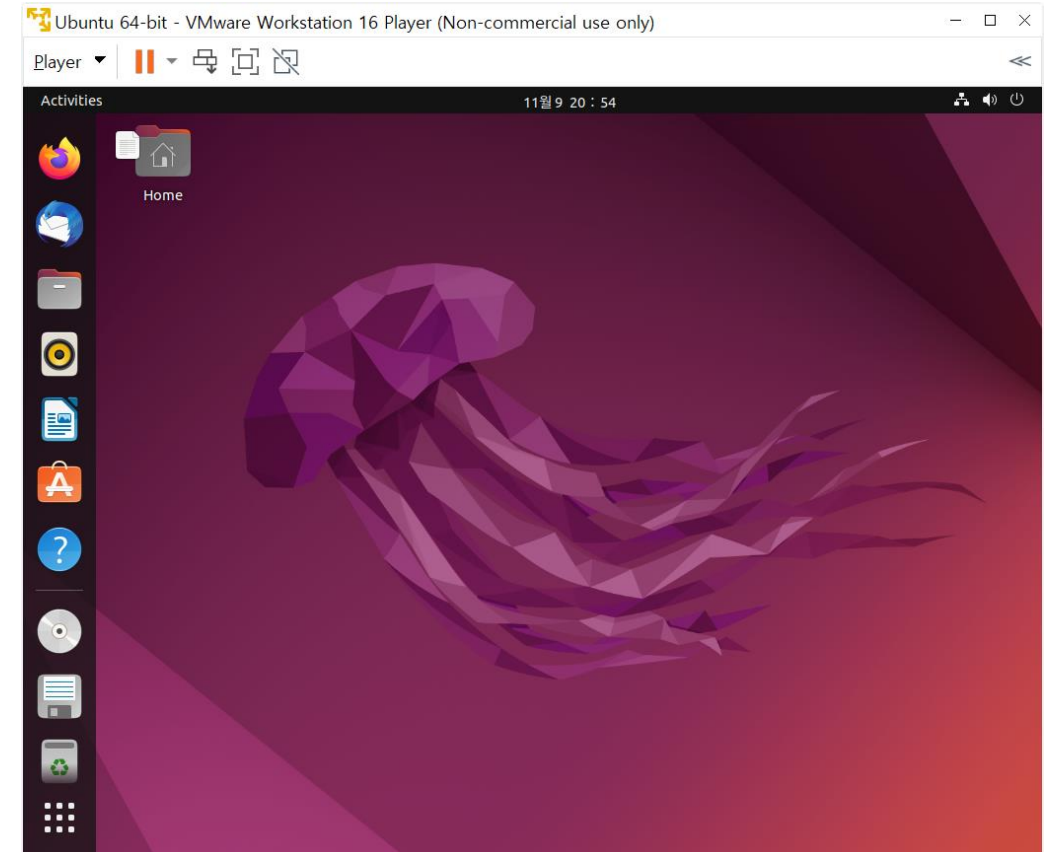
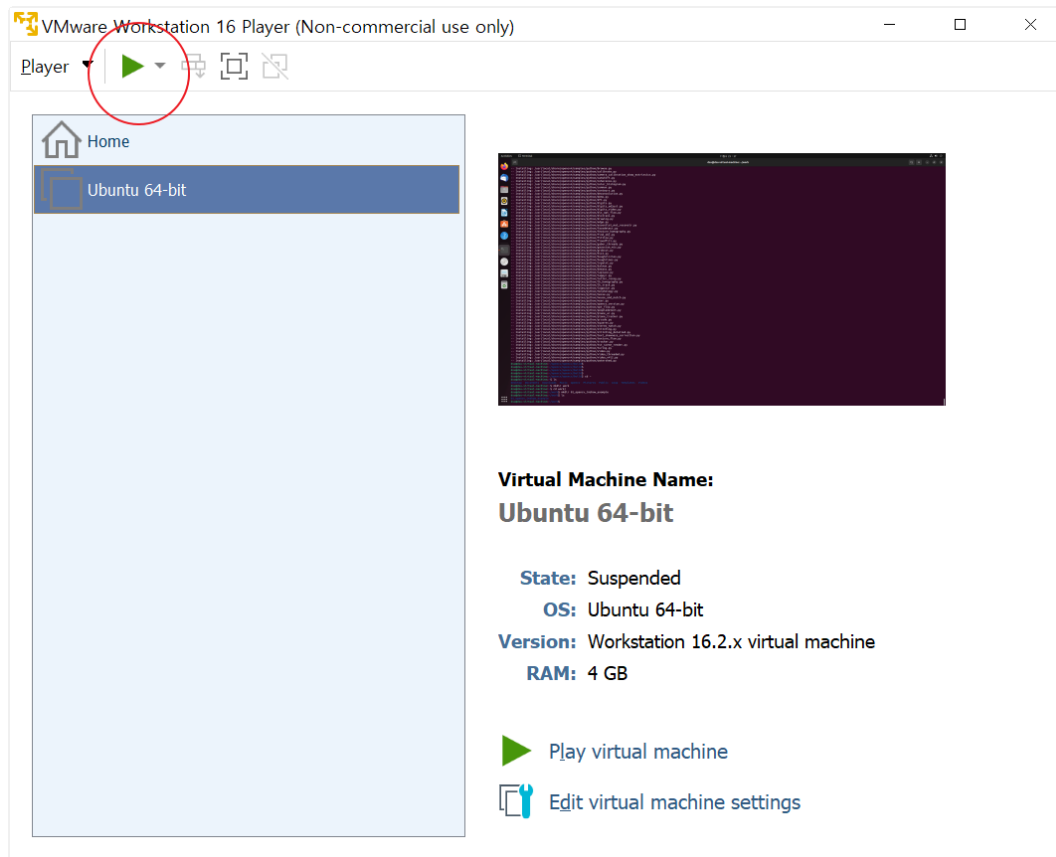
Vmware에 Ubuntu 설치



Vmware에 Ubuntu 설치



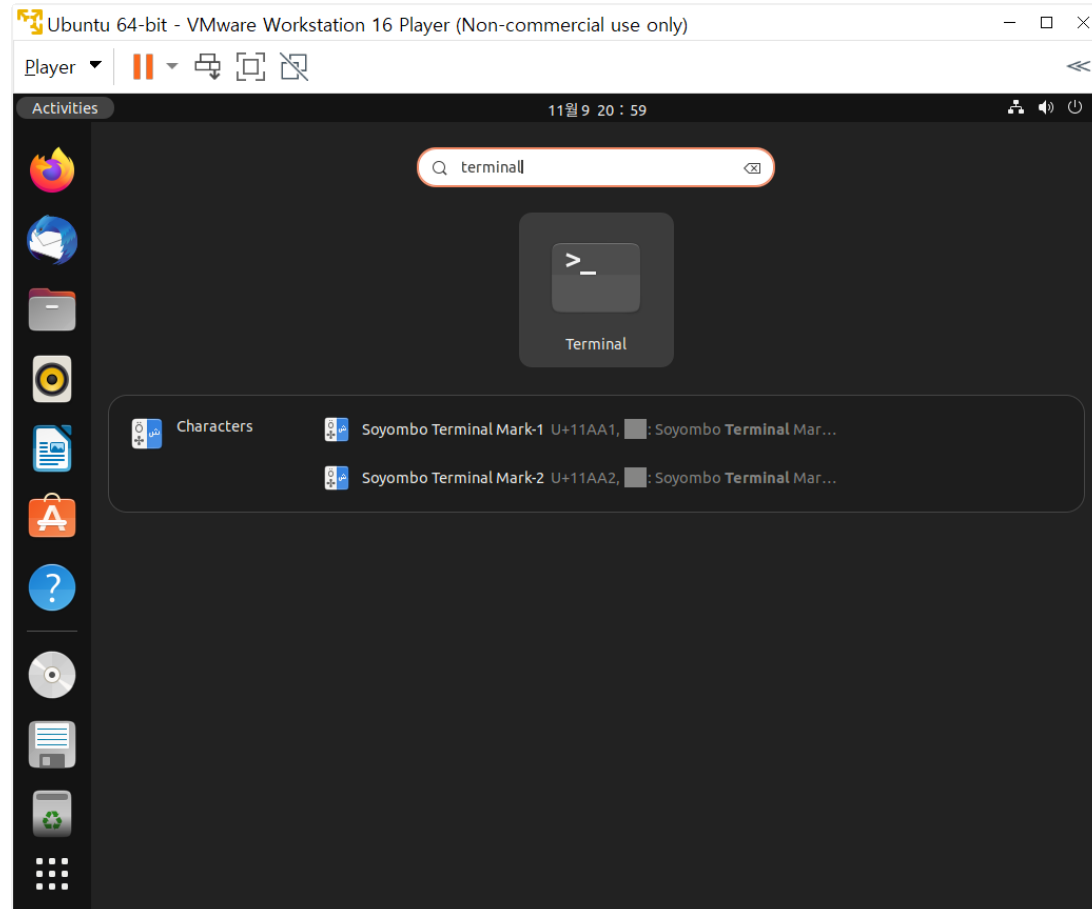
Hello Ubuntu!



Ubuntu에 OpenCV lib. 컴파일 및 설치

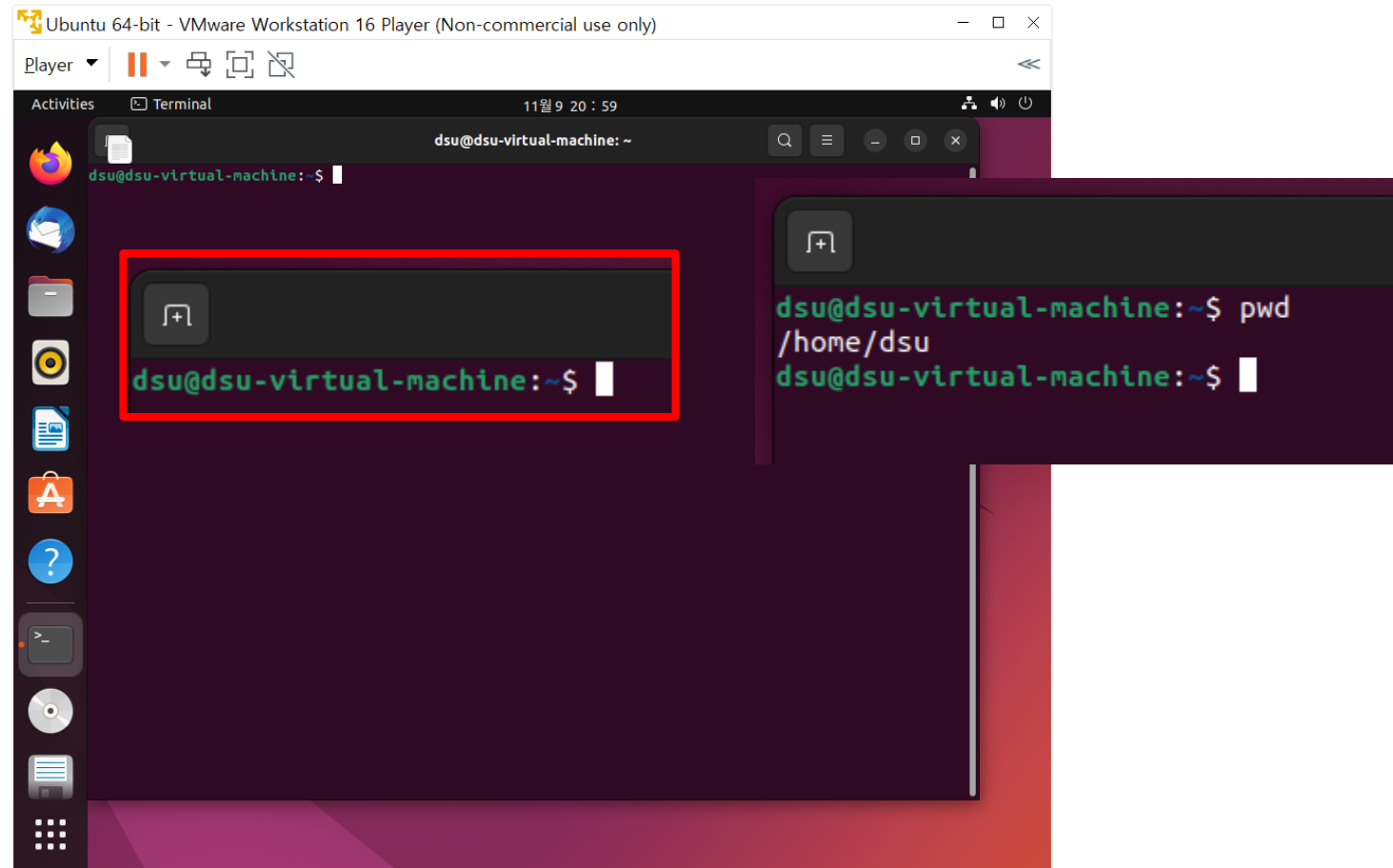
- OpenCV 컴파일에 필요한 유틸리티 및 의존 라이브러리 설치

명령실행을 위해서 terminal을 실행



Ubuntu에 OpenCV lib. 컴파일 및 설치

- OpenCV 컴파일에 필요한 유틸리티 및 의존 라이브러리 설치



Ubuntu에 OpenCV lib. 컴파일 및 설치 (소스코드)

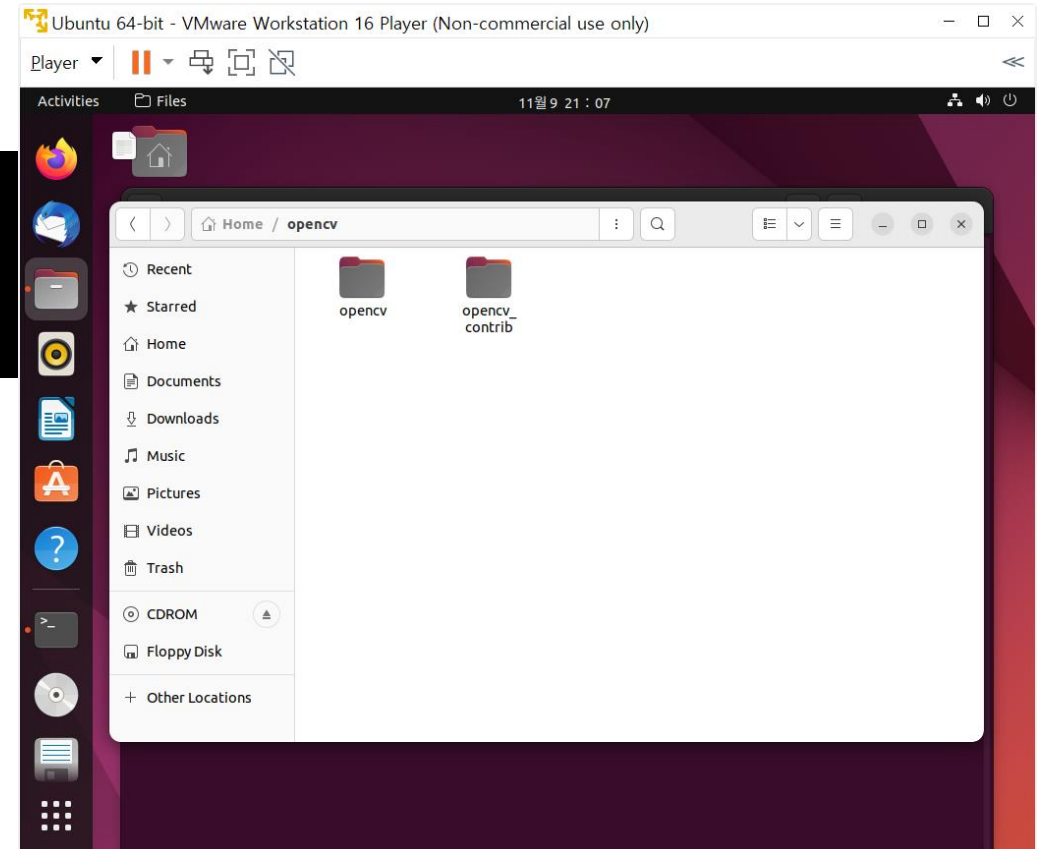
- OpenCV 컴파일에 필요한 유틸리티 및 의존 라이브러리 설치
- 터미널에 아래의 명령으로 필요한 라이브러리 모두 설치

```
~$ sudo apt-get update
~$ sudo apt-get install build-essential cmake git
~$ sudo apt-get install pkg-config
~$ sudo apt-get install libjpeg-dev libtiff5-dev libpng-dev
~$ sudo apt-get install ffmpeg libavcodec-dev libavformat-dev libswscale-dev libxvidcore-dev libx264-dev libxine2-dev
~$ sudo apt-get install libv4l-dev v4l-utils
~$ sudo apt-get install libgstreamer1.0-dev libgstreamer-plugins-base1.0-dev
~$ sudo apt-get install libgtk-3-dev
~$ sudo apt-get install mesa-utils libgl1-mesa-dri libgtkgl2.0-dev libgtkglext1-dev
~$ sudo apt-get install libatlas-base-dev gfortran libeigen3-dev
~$ sudo apt-get install python3-dev python3-numpy
```

Ubuntu에 OpenCV lib. 컴파일 및 설치 (소스코드)

- git을 이용하여 최신 OpenCV 라이브러리 소스코드 복사(clone)

```
~$ mkdir opencv
~$ cd opencv
~/opencv$ git clone https://github.com/opencv/opencv.git
~/opencv$ git clone https://github.com/opencv/opencv_contrib.git
```

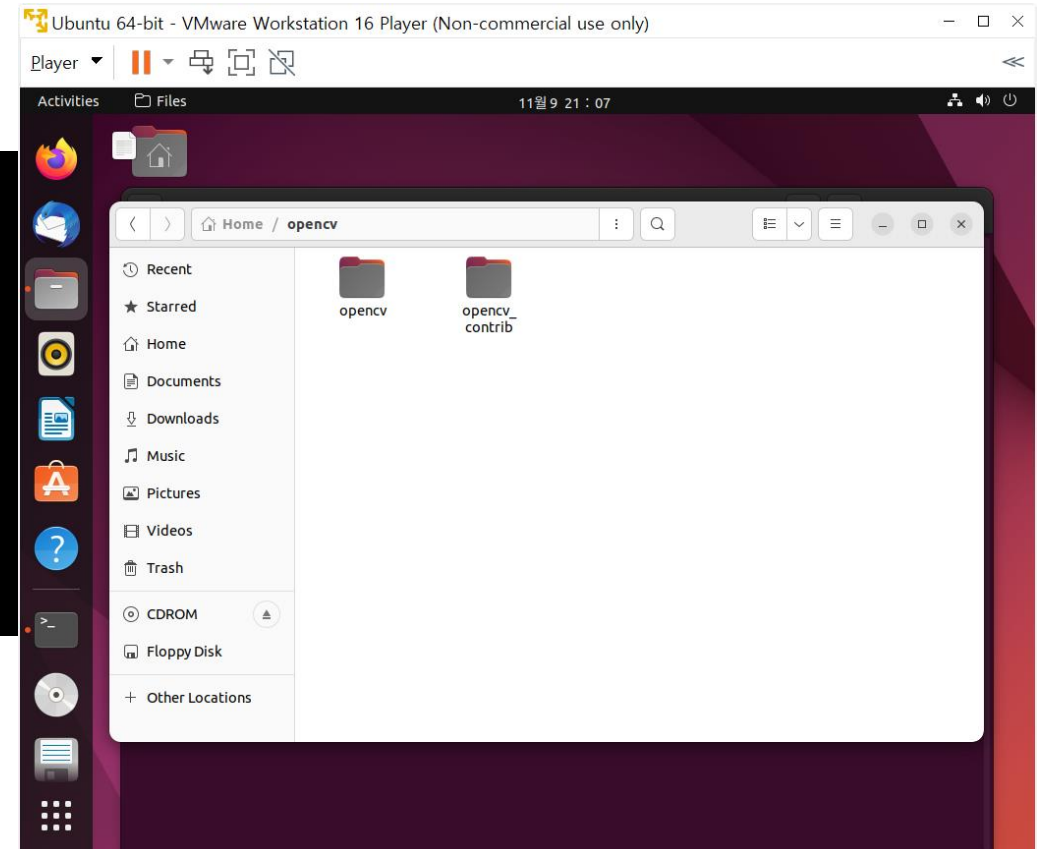


Ubuntu에 OpenCV lib. 컴파일 및 설치 (소스코드)

- git을 이용하여 최신 OpenCV 라이브러리 소스코드 최신 버전으로 변경(git checkout)

```
~$ mkdir opencv
~$ cd opencv
~/opencv$ cd opencv
~/opencv/opencv$ git checkout 4.6.0

~/opencv$ cd ../opencv_contrib
~/opencv/opencv_contrib$ git checkout 4.6.0
```



Ubuntu에 OpenCV lib. 컴파일 및 설치 (소스코드)

- opencv 소스코드를 다운로드 받았으니 opencv를 컴파일 하자!

```
#opencv 소스코드가 있는 폴더로 이동
```

```
~/opencv$ cd /home/hsu/opencv/opencv
```

```
~/opencv/opencv$
```

```
#opencv 컴파일 결과물을 저장해 놓을 폴더를 생성
```

```
~/opencv/opencv$ mkdir build
```

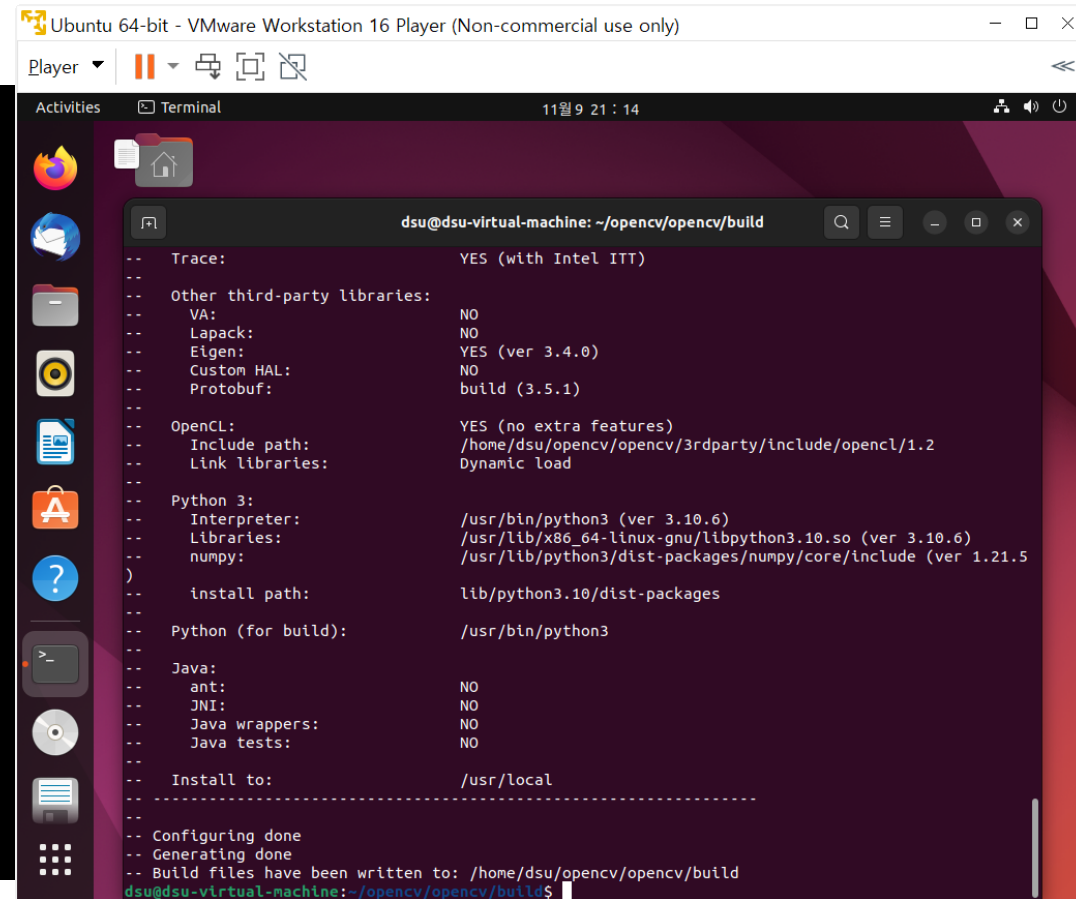
```
~/opencv/opencv$ cd build
```

```
~/opencv/opencv/build$
```

Ubuntu에 OpenCV lib. 컴파일 및 설치 (소스코드)

- opencv 소스코드를 다운로드 받았으니 opencv를 컴파일 하자!
- cmake를 이용한 configure

```
~/opencv/opencv/build $ cmake -D CMAKE_BUILD_TYPE=RELEASE -D CMAKE_INSTALL_PREFIX=/usr/local -D WITH_TBB=OFF -D WITH_IPP=OFF -D WITH_1394=OFF -D BUILD_WITH_DEBUG_INFO=OFF -D BUILD_DOCS=OFF -D INSTALL_C_EXAMPLES=ON -D INSTALL_PYTHON_EXAMPLES=ON -D BUILD_EXAMPLES=OFF -D BUILD_PACKAGE=OFF -D BUILD_TESTS=OFF -D BUILD_PERF_TESTS=OFF -D WITH_QT=OFF -D WITH_GTK=ON -D WITH_OPENGL=OFF -D BUILD_opencv_python3=ON -D OPENCV_EXTRA_MODULES_PATH=../opencv_contrib/modules -D WITH_V4L=ON -D WITH_FFMPEG=ON -D WITH_XINE=ON -D OPENCV_ENABLE_NONFREE=ON -D BUILD_NEW_PYTHON_SUPPORT=ON -D OPENCV_SKIP_PYTHON_LOADER=ON -D OPENCV_GENERATE_PKGCONFIG=ON ../
```

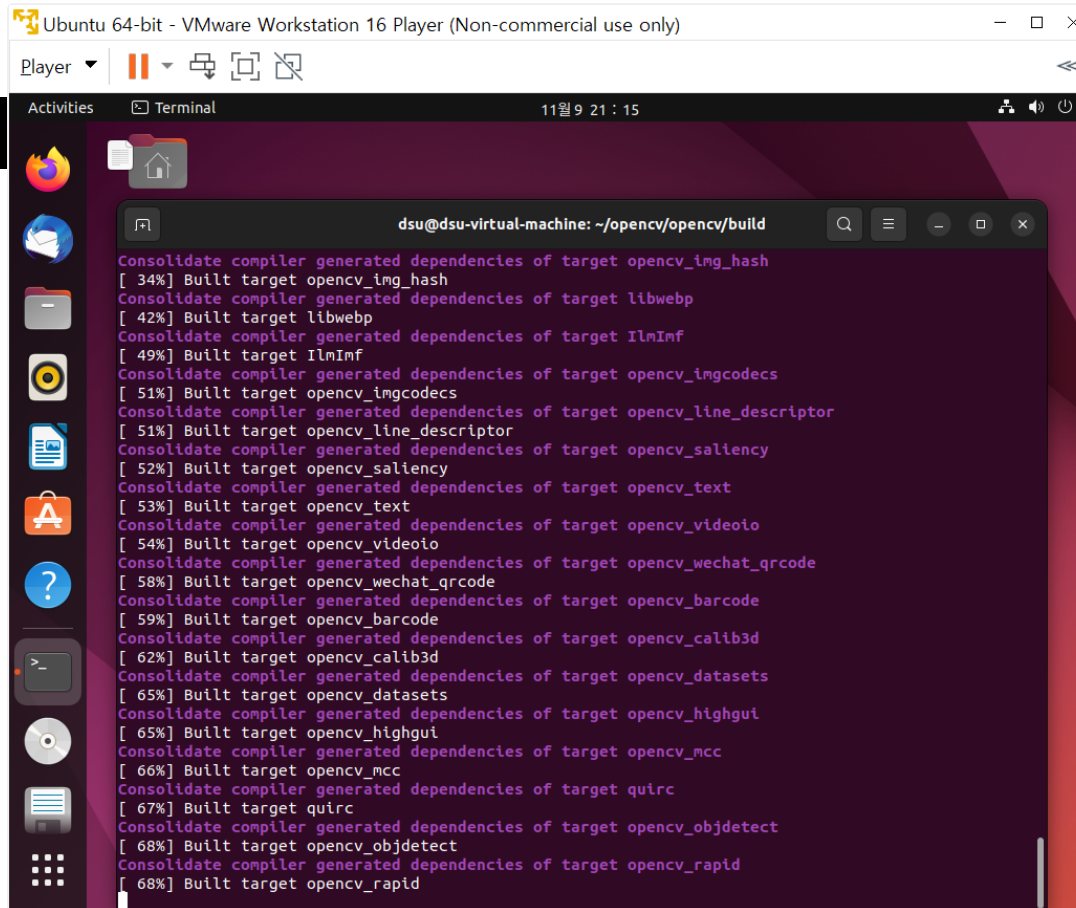


```
Ubuntu 64-bit - VMware Workstation 16 Player (Non-commercial use only)
Player
11월 9 21 : 14
dsu@dsu-virtual-machine: ~/opencv/opencv/build
-- Trace: YES (with Intel ITT)
-- Other third-party libraries:
--   VA: NO
--   Lapack: NO
--   Eigen: YES (ver 3.4.0)
--   Custom HAL: NO
--   Protobuf: build (3.5.1)
-- OpenCL: YES (no extra features)
--   Include path: /home/dsu/opencv/opencv/3rdparty/include/opencl/1.2
--   Link libraries: Dynamic load
-- Python 3:
--   Interpreter: /usr/bin/python3 (ver 3.10.6)
--   Libraries: /usr/lib/x86_64-linux-gnu/libpython3.10.so (ver 3.10.6)
--   numpy: /usr/lib/python3/dist-packages/numpy/core/include (ver 1.21.5)
--   install path: lib/python3.10/dist-packages
-- Python (for build): /usr/bin/python3
-- Java:
--   ant: NO
--   JNI: NO
--   Java wrappers: NO
--   Java tests: NO
-- Install to: /usr/local
--
-- Configuring done
-- Generating done
-- Build files have been written to: /home/dsu/opencv/opencv/build
dsu@dsu-virtual-machine: ~/opencv/opencv/build$
```

Ubuntu에 OpenCV lib. 컴파일 및 설치 (소스코드)

- opencv 소스코드를 다운로드 받았으니 opencv를 컴파일 하자!
- make!

```
~/opencv/opencv/build $ make
```



The screenshot shows a terminal window titled "dsu@dsu-virtual-machine: ~/opencv/opencv/build". The terminal output displays the progress of compiling various OpenCV targets, with each target preceded by a percentage indicating its completion status. The targets listed include opencv_img_hash, libwebp, IlmImf, opencv_imgcodecs, opencv_line_descriptor, opencv_saliency, opencv_text, opencv_videoio, opencv_wechat_qrcode, opencv_barcode, opencv_calib3d, opencv_datasets, opencv_highgui, opencv_mcc, opencv_quirc, opencv_objdetect, and opencv_rapid. The progress bar at the bottom of the terminal window is partially filled, indicating the overall progress of the compilation process.

```
dsu@dsu-virtual-machine: ~/opencv/opencv/build
Consolidate compiler generated dependencies of target opencv_img_hash
[ 34%] Built target opencv_img_hash
Consolidate compiler generated dependencies of target libwebp
[ 42%] Built target libwebp
Consolidate compiler generated dependencies of target IlmImf
[ 49%] Built target IlmImf
Consolidate compiler generated dependencies of target opencv_imgcodecs
[ 51%] Built target opencv_imgcodecs
Consolidate compiler generated dependencies of target opencv_line_descriptor
[ 51%] Built target opencv_line_descriptor
Consolidate compiler generated dependencies of target opencv_saliency
[ 52%] Built target opencv_saliency
Consolidate compiler generated dependencies of target opencv_text
[ 53%] Built target opencv_text
Consolidate compiler generated dependencies of target opencv_videoio
[ 54%] Built target opencv_videoio
Consolidate compiler generated dependencies of target opencv_wechat_qrcode
[ 58%] Built target opencv_wechat_qrcode
Consolidate compiler generated dependencies of target opencv_barcode
[ 59%] Built target opencv_barcode
Consolidate compiler generated dependencies of target opencv_calib3d
[ 62%] Built target opencv_calib3d
Consolidate compiler generated dependencies of target opencv_datasets
[ 65%] Built target opencv_datasets
Consolidate compiler generated dependencies of target opencv_highgui
[ 65%] Built target opencv_highgui
Consolidate compiler generated dependencies of target opencv_mcc
[ 66%] Built target opencv_mcc
Consolidate compiler generated dependencies of target opencv_quirc
[ 67%] Built target opencv_quirc
Consolidate compiler generated dependencies of target opencv_objdetect
[ 68%] Built target opencv_objdetect
Consolidate compiler generated dependencies of target opencv_rapid
[ 68%] Built target opencv_rapid
```

Ubuntu에 OpenCV lib. 컴파일 및 설치 (소스코드)

- opencv 소스코드를 다운로드 받았으니 opencv를 컴파일 하자!
- 컴파일된 opencvlib를 ubuntu에 설치!(실제로는 lib파일을
=/usr/local 폴더에 복사)

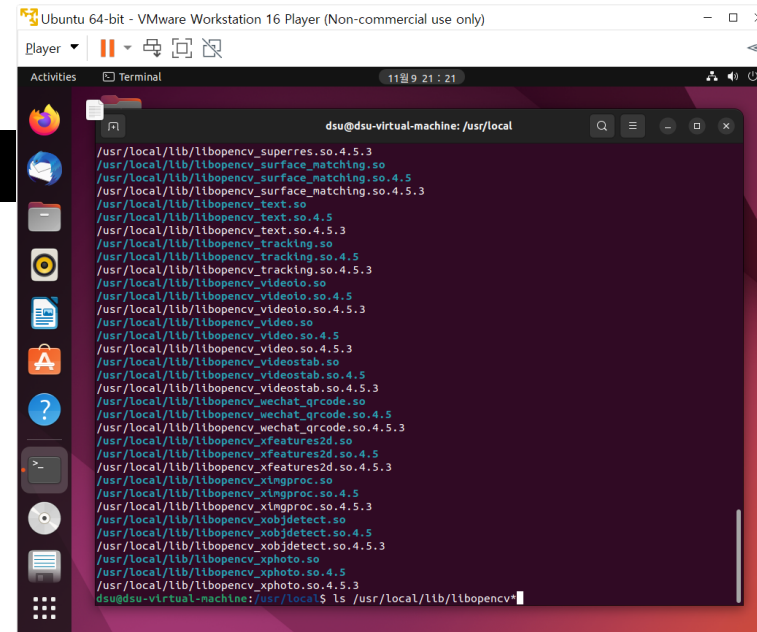
```
~/opencv/opencv/build $ sudo make install
```

```
~/opencv/opencv/build $ cmake -D CMAKE_BUILD_TYPE=RELEASE -D  
CMAKE_INSTALL_PREFIX=/usr/local -D WITH_TBB=OFF -D  
WITH_IPP=OFF -D WITH_1394=OFF -D  
BUILD_WITH_DEBUG_INFO=OFF -D BUILD_DOCS=OFF -D  
INSTALL_C_EXAMPLES=ON -D INSTALL_PYTHON_EXAMPLES=ON -D  
BUILD_EXAMPLES=OFF -D BUILD_PACKAGE=OFF -D  
BUILD_TESTS=OFF -D BUILD_PERF_TESTS=OFF -D WITH_QT=OFF -D  
WITH_GTK=ON -D WITH_OPENGL=OFF -D  
BUILD_opencv_python3=ON -D  
OPENCV_EXTRA_MODULES_PATH=../../opencv_contrib/modules -D  
WITH_V4L=ON -D WITH_FFMPEG=ON -D WITH_XINE=ON -D  
OPENCV_ENABLE_NONFREE=ON -D  
BUILD_NEW_PYTHON_SUPPORT=ON -D  
OPENCV_SKIP_PYTHON_LOADER=ON -D  
OPENCV_GENERATE_PKGCONFIG=ON ..
```

Ubuntu에 OpenCV lib. 컴파일 및 설치 (소스코드)

- opencv 소스코드를 다운로드 받았으니 opencv를 컴파일 하자!
- 컴파일된 opencvlib를 ubuntu에 설치!(실제로는 lib파일을
=/usr/local 폴더에 복사)
- 실제로 /usr/local/ 폴더에 OpenCV라이브러리가 있는지 확인해
보자!

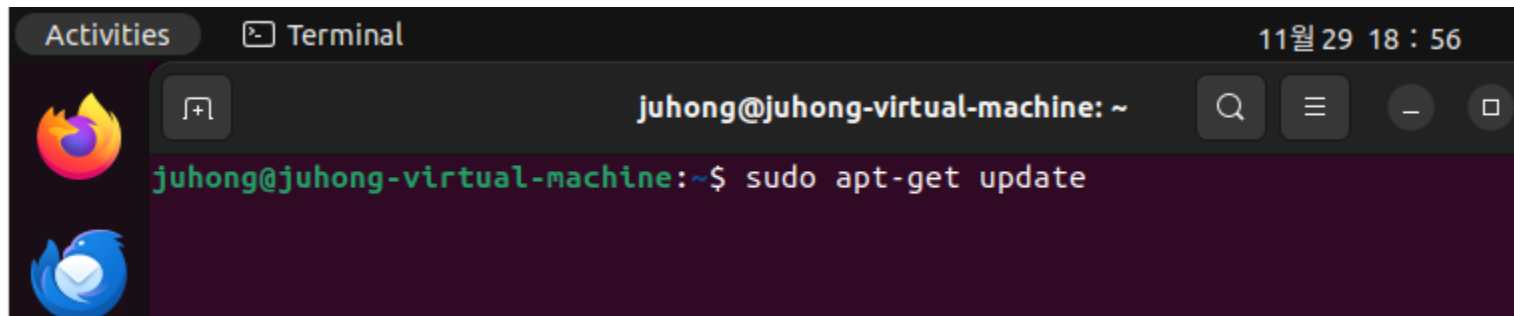
```
~/opencv/opencv/build $ ls /usr/local/lib/libopencv*
```



```
Ubuntu 64-bit - VMware Workstation 16 Player (Non-commercial use only)
Player
Activities
Terminal
11월 9 21:21
dsu@dsu-virtual-machine: /usr/local
/usr/local/lib/libopencv_superres.so.4.5.3
/usr/local/lib/libopencv_surface_matching.so
/usr/local/lib/libopencv_surface_matching.so.4.5
/usr/local/lib/libopencv_surface_matching.so.4.5.3
/usr/local/lib/libopencv_text.so
/usr/local/lib/libopencv_text.so.4.5
/usr/local/lib/libopencv_text.so.4.5.3
/usr/local/lib/libopencv_tracking.so
/usr/local/lib/libopencv_tracking.so.4.5
/usr/local/lib/libopencv_tracking.so.4.5.3
/usr/local/lib/libopencv_videot.so
/usr/local/lib/libopencv_videot.so.4.5
/usr/local/lib/libopencv_videot.so.4.5.3
/usr/local/lib/libopencv_video.so
/usr/local/lib/libopencv_video.so.4.5
/usr/local/lib/libopencv_video.so.4.5.3
/usr/local/lib/libopencv_videostab.so
/usr/local/lib/libopencv_videostab.so.4.5
/usr/local/lib/libopencv_videostab.so.4.5.3
/usr/local/lib/libopencv_wechat_qrcode.so
/usr/local/lib/libopencv_wechat_qrcode.so.4.5
/usr/local/lib/libopencv_wechat_qrcode.so.4.5.3
/usr/local/lib/libopencv_xfeatures2d.so
/usr/local/lib/libopencv_xfeatures2d.so.4.5
/usr/local/lib/libopencv_xfeatures2d.so.4.5.3
/usr/local/lib/libopencv_ximgproc.so
/usr/local/lib/libopencv_ximgproc.so.4.5
/usr/local/lib/libopencv_ximgproc.so.4.5.3
/usr/local/lib/libopencv_xobjdetect.so
/usr/local/lib/libopencv_xobjdetect.so.4.5
/usr/local/lib/libopencv_xobjdetect.so.4.5.3
/usr/local/lib/libopencv_xphoto.so
/usr/local/lib/libopencv_xphoto.so.4.5
/usr/local/lib/libopencv_xphoto.so.4.5.3
dsu@dsu-virtual-machine: /usr/local$ ls /usr/local/lib/libopencv*
```

Ubuntu에 OpenCV lib. 컴파일 및 설치 (패키징)

- OpenCV를 소스코드가 아닌 대표 패키징 라이브러리로 설치
- "sudo apt install libopencv-dev" 명령으로 필요한 의존 라이브러리와 함께 OpenCV 최신버전 설치



The screenshot shows a terminal window titled "Terminal" with the date and time "11월 29 18 : 56" in the top right corner. The terminal prompt is "juhong@juhong-virtual-machine: ~". The command "sudo apt-get update" has been entered and is being executed. The terminal window is part of a desktop environment with a taskbar at the bottom showing icons for Firefox and a mail client.

```
juhong@juhong-virtual-machine: ~$ sudo apt-get update
```


Ubuntu에 OpenCV lib. 컴파일 및 설치 (패키징)

의존 패키지 먼저 설치

```
$ sudo apt-get update
```

```
$ sudo apt-get install pkg-config
```

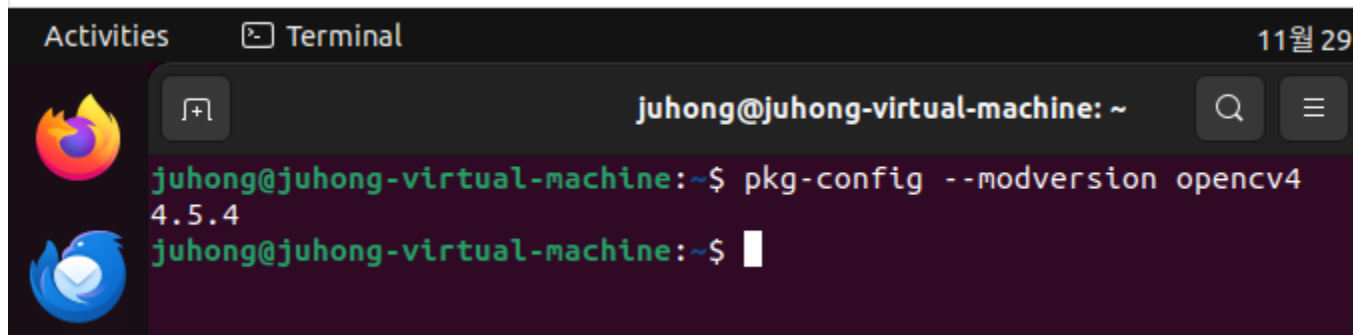
```
$ sudo apt-get install cmake
```

OpenCV 패키지 설치

```
$ sudo apt-get install libopencv-dev
```

Ubuntu에 OpenCV lib. 컴파일 및 설치 (패키징)

- OpenCV 설치 확인(버전 확인)
 - `pkg-config --modversion opencv4`

A screenshot of a Linux terminal window. The title bar shows 'Activities' and 'Terminal' on the left, and '11월 29' on the right. The terminal content shows a user prompt 'juhong@juhong-virtual-machine: ~' followed by the command 'pkg-config --modversion opencv4' and its output '4.5.4'.

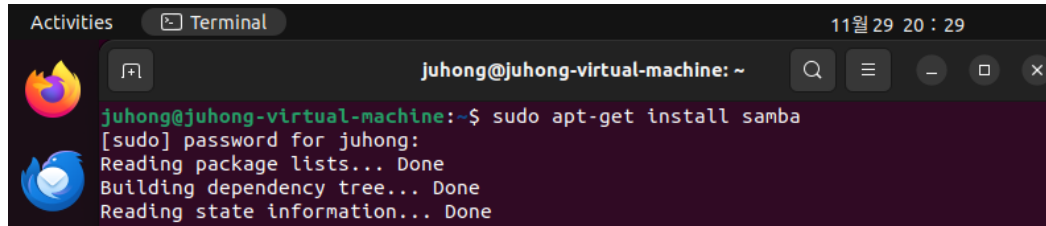
```
juhong@juhong-virtual-machine: ~  
juhong@juhong-virtual-machine:~$ pkg-config --modversion opencv4  
4.5.4  
juhong@juhong-virtual-machine:~$
```

Window에서 Ubuntu폴더에 접근 : Samba



Window에서 Ubuntu폴더에 접근 : Samba

- Ubuntu에 Samba 설치
 - 설치 명령 : `sudo apt-get install samba`

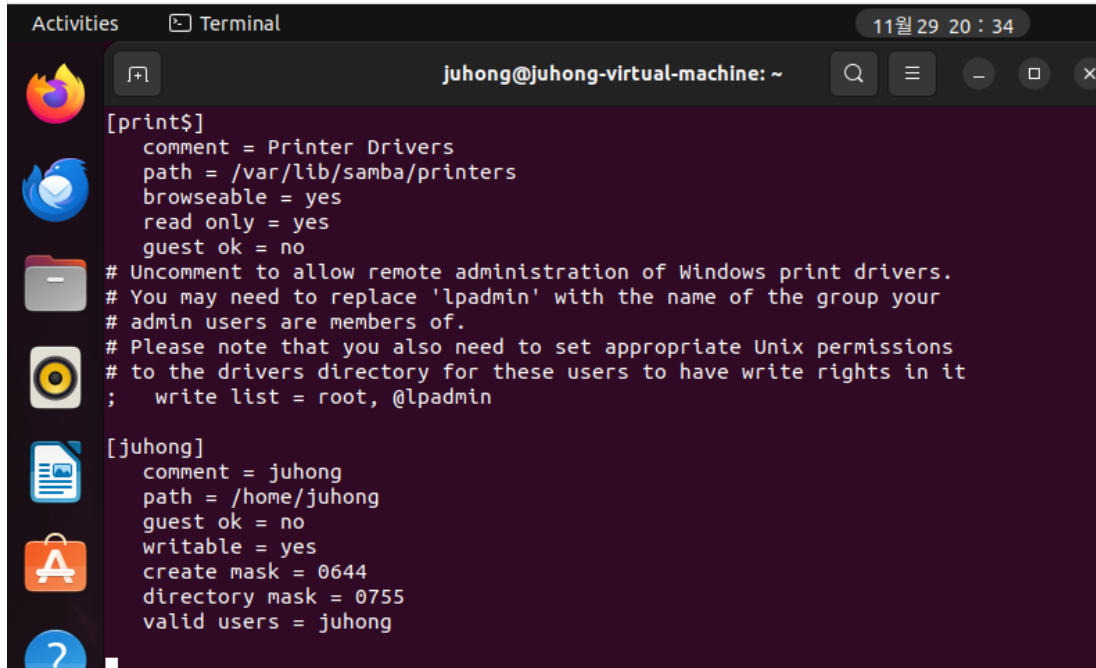
A screenshot of a Linux terminal window. The title bar shows 'Activities' and 'Terminal' on the left, and the date and time '11월 29 20 : 29' on the right. The terminal prompt is 'juhong@juhong-virtual-machine: ~'. The user has entered the command 'sudo apt-get install samba'. The terminal output shows the password prompt '[sudo] password for juhong:', followed by 'Reading package lists... Done', 'Building dependency tree... Done', and 'Reading state information... Done'.

```
juhong@juhong-virtual-machine: ~  
juhong@juhong-virtual-machine:~$ sudo apt-get install samba  
[sudo] password for juhong:  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done
```

- Samba 설정
 - 설정 파일(smb.conf)의 내용 수정
 - `sudo vim /etc/samba/smb.conf`

Window에서 Ubuntu폴더에 접근 : Samba

- Samba 설정
 - 설정 파일(smb.conf)의 내용 수정
 - `sudo vi /etc/samba/smb.conf`



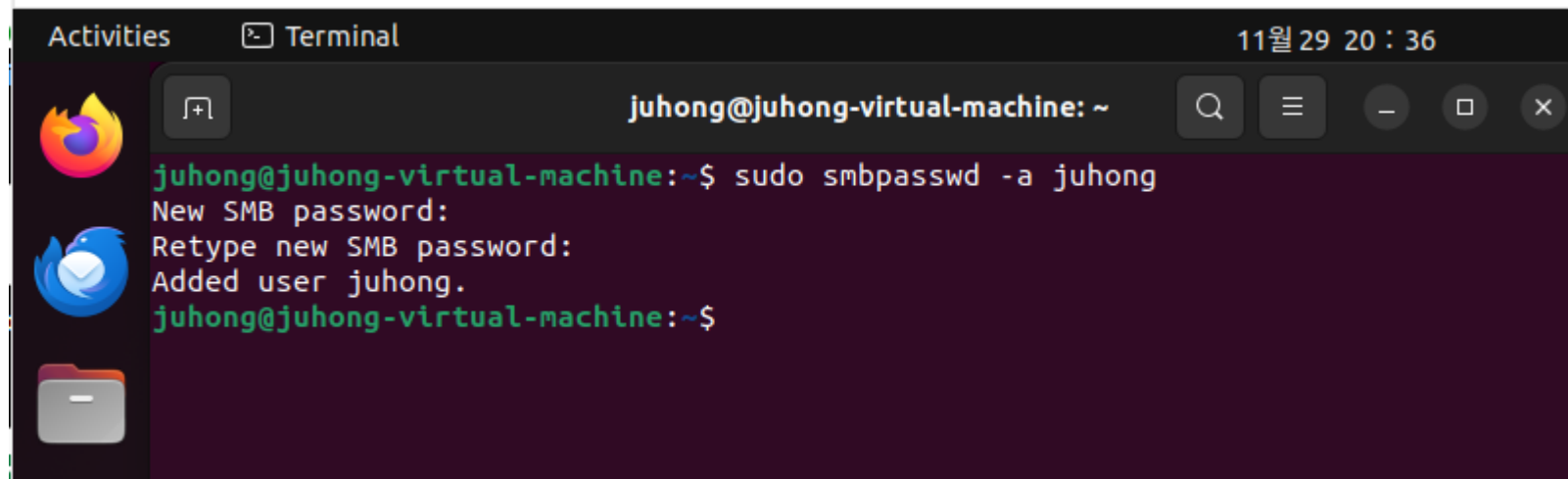
```
Activities  Terminal  11월 29 20 : 34
juhong@juhong-virtual-machine: ~

[print$]
comment = Printer Drivers
path = /var/lib/samba/printers
browseable = yes
read only = yes
guest ok = no
# Uncomment to allow remote administration of Windows print drivers.
# You may need to replace 'lpadmin' with the name of the group your
# admin users are members of.
# Please note that you also need to set appropriate Unix permissions
# to the drivers directory for these users to have write rights in it
; write list = root, @lpadmin

[juhong]
comment = juhong
path = /home/juhong
guest ok = no
writable = yes
create mask = 0644
directory mask = 0755
valid users = juhong
```

Window에서 Ubuntu폴더에 접근 : Samba

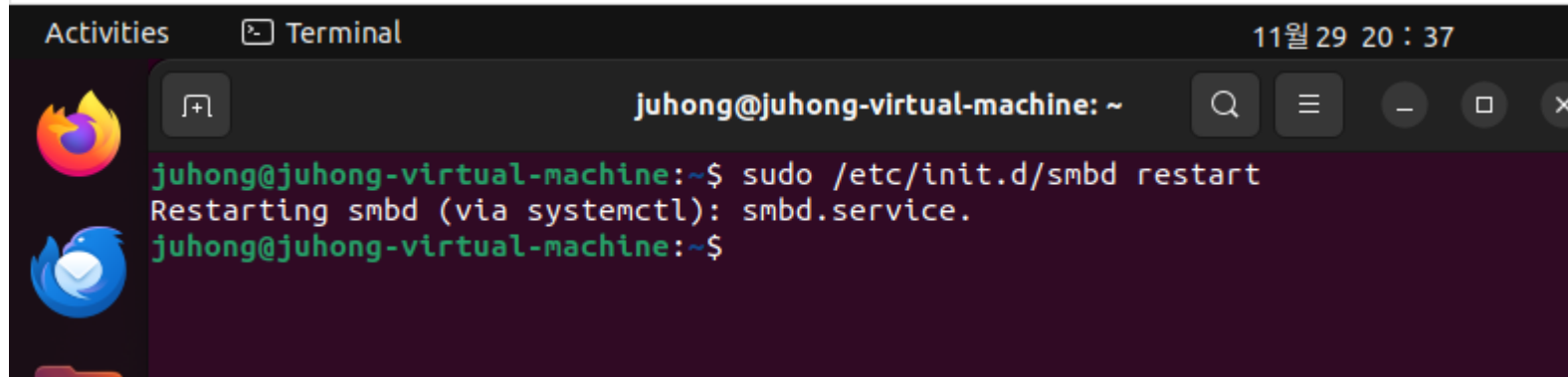
- Samba 설정
 - Samba 사용자 추가
 - 명령 : `sudo smbpasswd -a (username)`

A screenshot of a Linux terminal window. The window title bar shows 'Activities', 'Terminal', and the date/time '11월 29 20 : 36'. The terminal prompt is 'juhong@juhong-virtual-machine: ~'. The user has entered the command 'sudo smbpasswd -a juhong'. The terminal output shows 'New SMB password:', 'Retype new SMB password:', and 'Added user juhong.' followed by the prompt 'juhong@juhong-virtual-machine:~\$'. On the left side of the terminal window, there is a vertical sidebar with icons for Firefox, a mail client, and a file manager.

```
Activities Terminal 11월 29 20 : 36
juhong@juhong-virtual-machine: ~
juhong@juhong-virtual-machine:~$ sudo smbpasswd -a juhong
New SMB password:
Retype new SMB password:
Added user juhong.
juhong@juhong-virtual-machine:~$
```


Window에서 Ubuntu폴더에 접근 : Samba

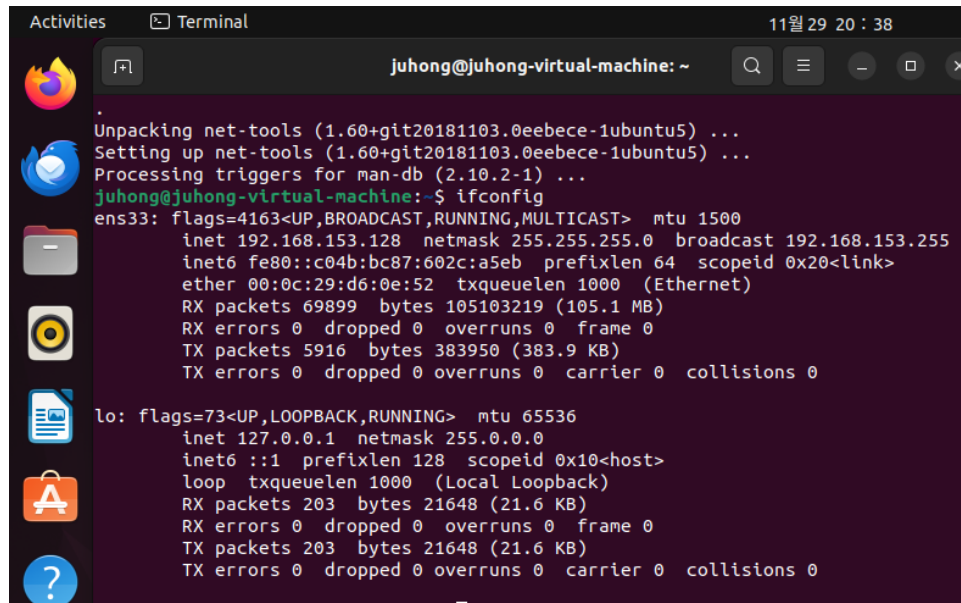
- Samba 설정을 모두 마쳤으면 Samba 서비스 재시작
 - 명령 : `sudo /etc/init.d/smbd restart`

A screenshot of a Linux terminal window. The title bar shows 'Activities' and 'Terminal' on the left, and the date and time '11월 29 20 : 37' on the right. The terminal prompt is 'juhong@juhong-virtual-machine: ~'. The user has entered the command 'sudo /etc/init.d/smbd restart'. The output of the command is 'Restarting smbd (via systemctl): smbd.service.' followed by a new prompt 'juhong@juhong-virtual-machine: ~\$'.

```
juhong@juhong-virtual-machine:~$ sudo /etc/init.d/smbd restart
Restarting smbd (via systemctl): smbd.service.
juhong@juhong-virtual-machine:~$
```

Window에서 Ubuntu폴더에 접근 : Samba

- 이제 네트워크(폴더)로 Window와 Ubuntu와 파일을 공유
 - Ubuntu의 IP 주소를 확인
 - 명령 : ifconfig
 - ifconfig 명령으로 확인한 Ubuntu의 IP 주소 : **192.168.153.128**

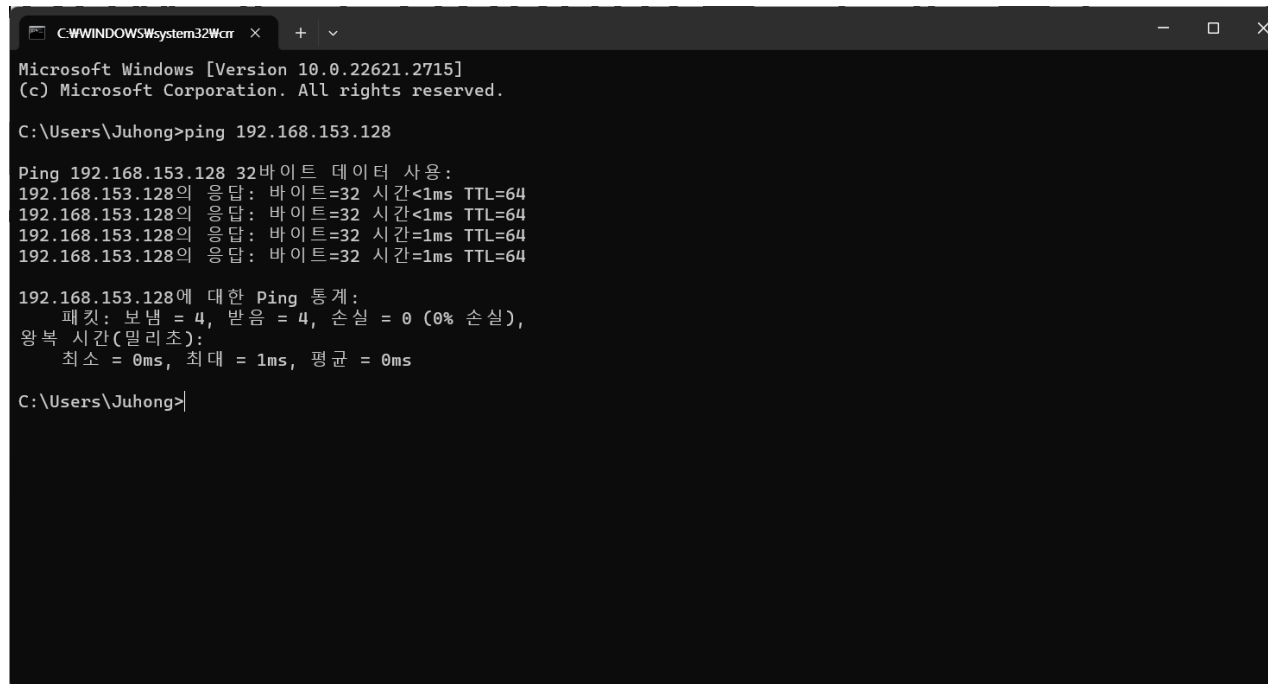


```
Activities Terminal 11월 29 20 : 38
juhong@juhong-virtual-machine: ~
.
Unpacking net-tools (1.60+git20181103.0eebece-1ubuntu5) ...
Setting up net-tools (1.60+git20181103.0eebece-1ubuntu5) ...
Processing triggers for man-db (2.10.2-1) ...
juhong@juhong-virtual-machine:~$ ifconfig
ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.153.128 netmask 255.255.255.0 broadcast 192.168.153.255
    inet6 fe80::c04b:bc87:602c:a5eb prefixlen 64 scopeid 0x20<link>
    ether 00:0c:29:d6:0e:52 txqueuelen 1000 (Ethernet)
    RX packets 69899 bytes 105103219 (105.1 MB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 5916 bytes 383950 (383.9 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 203 bytes 21648 (21.6 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 203 bytes 21648 (21.6 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

Window에서 Ubuntu폴더에 접근 : Samba

- 이제 네트워크(폴더)로 Window와 Ubuntu와 파일을 공유
 - 윈도우에서 Ubuntu와의 네트워크 연결 확인
 - 윈도우 터미널에서 명령 : ping **192.168.153.128**

A screenshot of a Windows Command Prompt window. The title bar shows 'C:\WINDOWS\system32\cmd.exe'. The window content displays the output of a 'ping' command. It shows four successful replies from 192.168.153.128 with a time of less than 1ms and TTL of 64. A summary at the bottom indicates 0% packet loss and a round-trip time of 0ms.

```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows [Version 10.0.22621.2715]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Juhong>ping 192.168.153.128

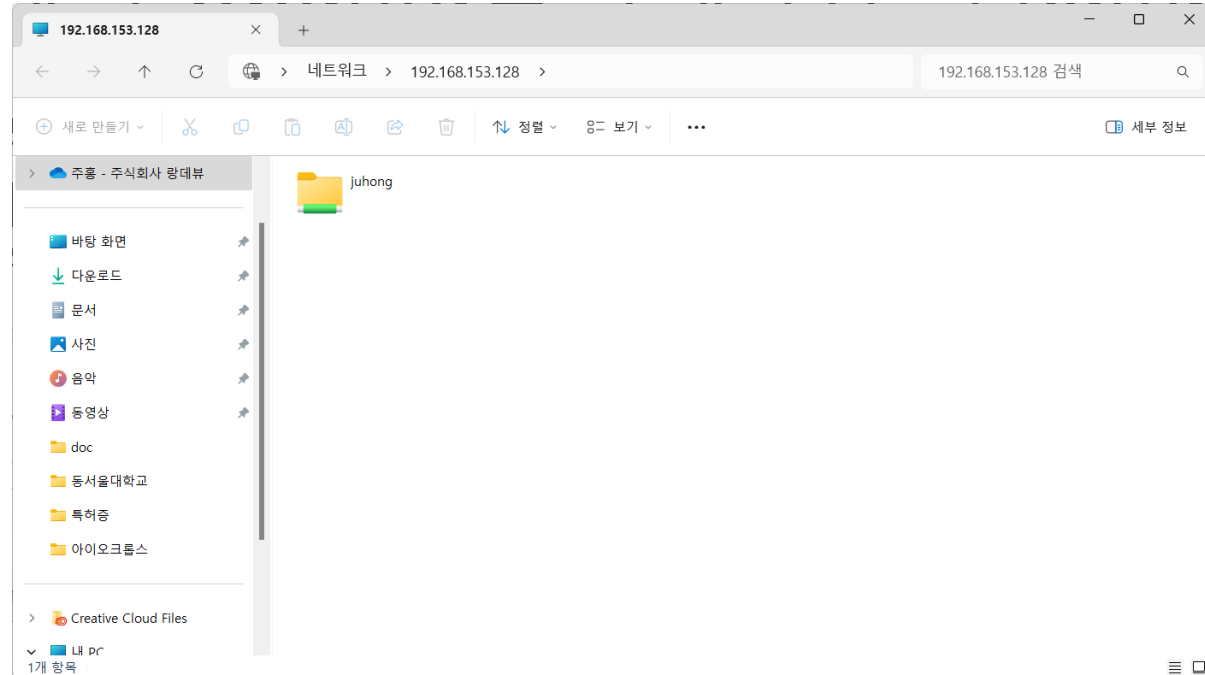
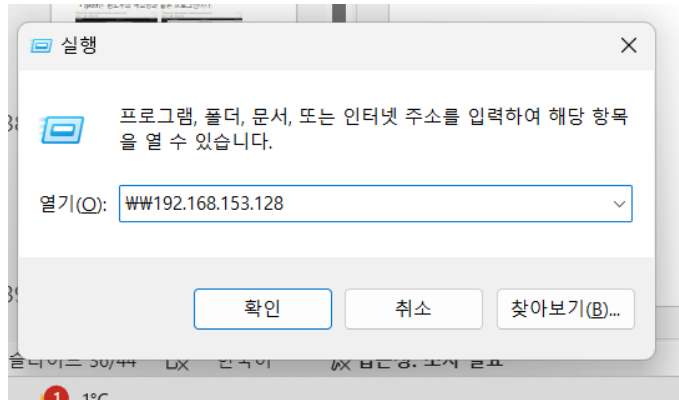
Ping 192.168.153.128 32바이트 데이터 사용:
192.168.153.128의 응답: 바이트=32 시간<1ms TTL=64
192.168.153.128의 응답: 바이트=32 시간<1ms TTL=64
192.168.153.128의 응답: 바이트=32 시간=1ms TTL=64
192.168.153.128의 응답: 바이트=32 시간=1ms TTL=64

192.168.153.128에 대한 Ping 통계:
패킷: 보냄 = 4, 받음 = 4, 손실 = 0 (0% 손실),
왕복 시간(밀리초):
최소 = 0ms, 최대 = 1ms, 평균 = 0ms

C:\Users\Juhong>
```

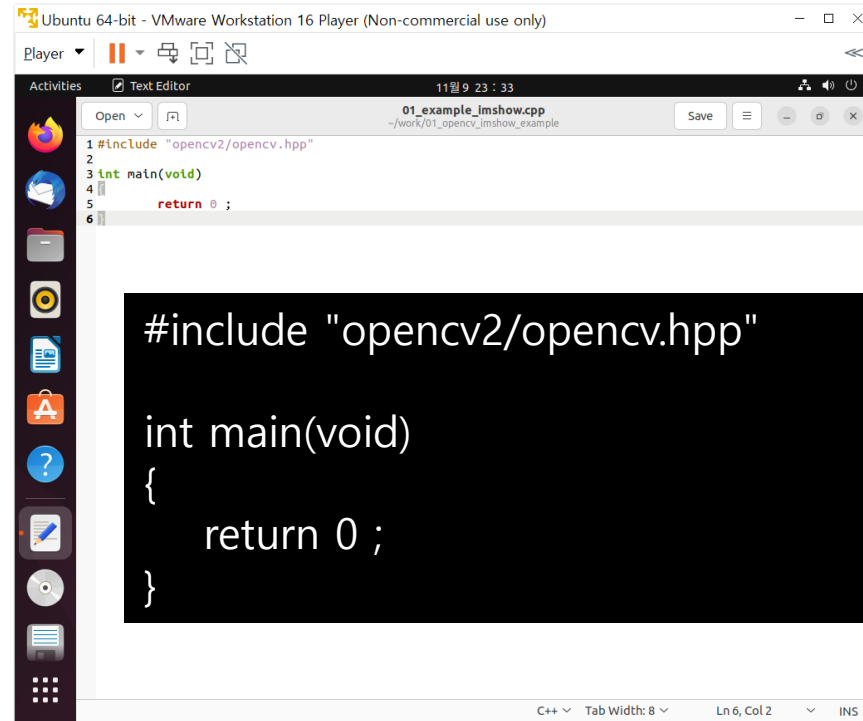
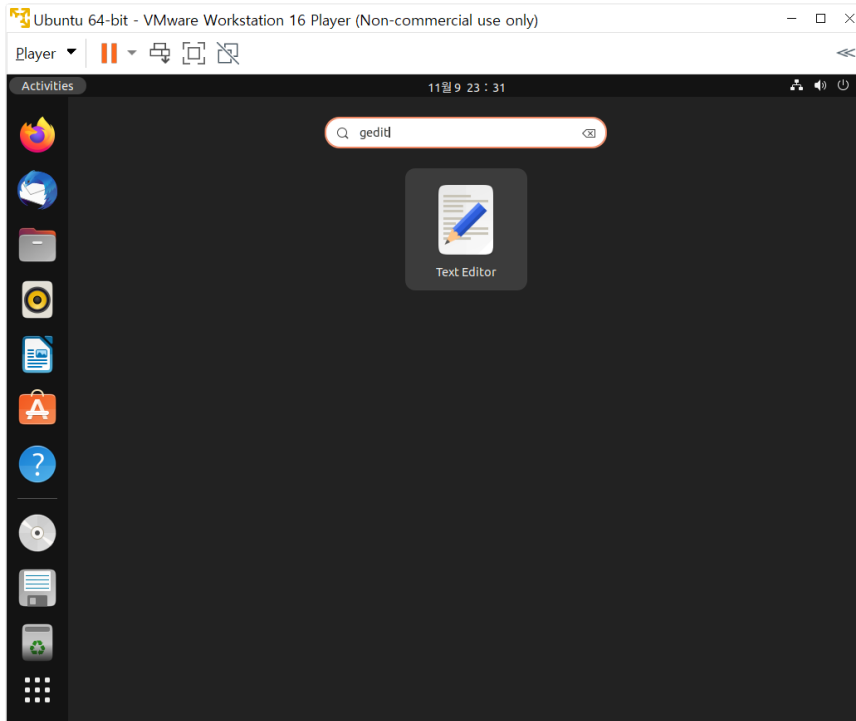
Window에서 Ubuntu폴더에 접근 : Samba

- 이제 네트워크(폴더)로 Window와 Ubuntu와 파일을 공유
 - 윈도우에서 Ubuntu의 (네트워크)폴더에 접근
 - 윈도우 실행 명령 : **\\192.168.153.128**



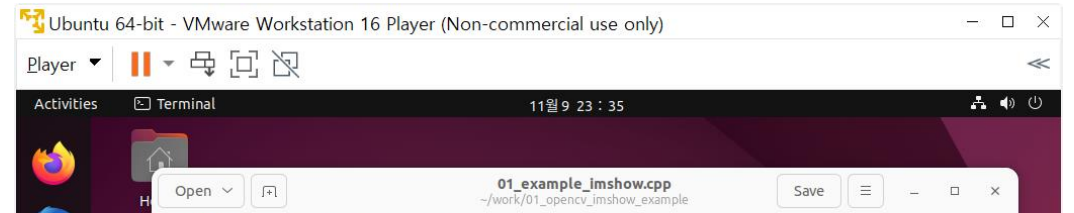
OpenCV 예제 프로그램 컴파일

- OpenCV를 이용한 simple example
- gedit를 이용하여 아래의 cpp(소스코드)파일을 만든다.
 - gedit는 윈도우의 메모장과 같은 프로그램이다.

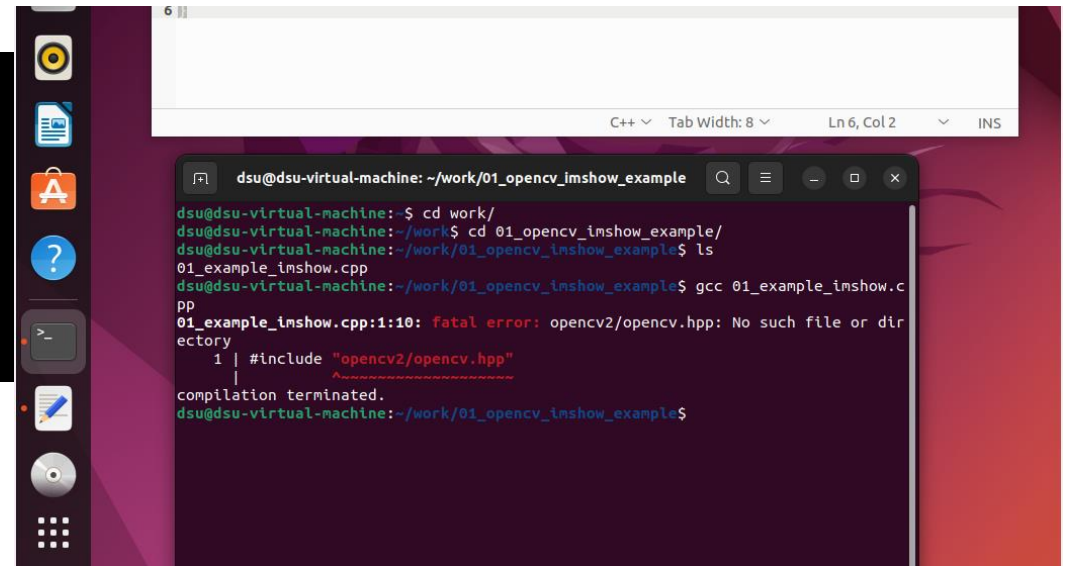
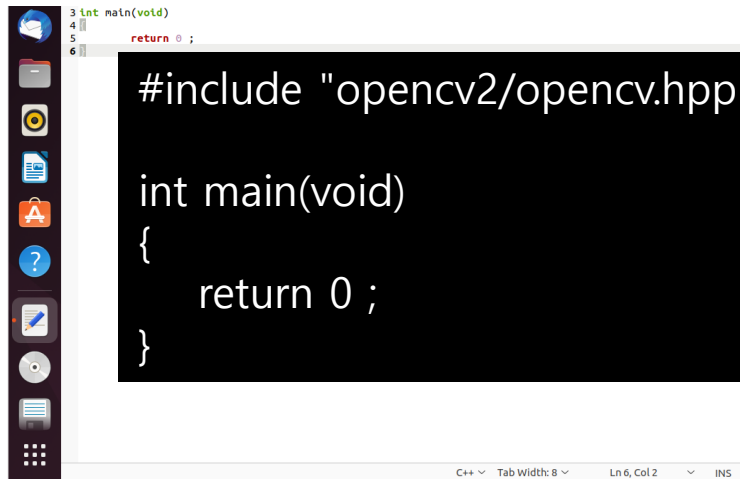


OpenCV 예제 프로그램 컴파일

- OpenCV를 이용한 simple example
- gedit를 이용하여 아래의 cpp(소스코드)파일을 만들고 저장
- GCC를 이용하여 cpp 소스코드를 컴파일



gcc test.cpp -o test

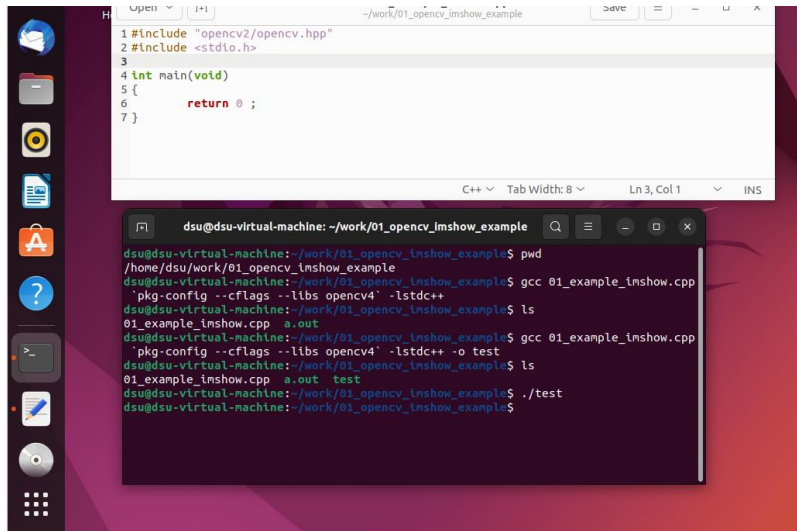


OpenCV 예제 프로그램 컴파일

- OpenCV 헤더를 추가만 하였는데 컴파일 에러가 발생한다.
- OpenCV 헤더와 라이브러리를 사용하기 위한 컴파일 옵션을 추가해야 한다.

```
~/opencv/opencv/build $ gcc 01_example_imshow.cpp `pkg-config --cflags --libs opencv4` -lstdc++
```

```
gcc test.cpp -o test `pkg-config --cflags --libs opencv4` -lstdc++
```



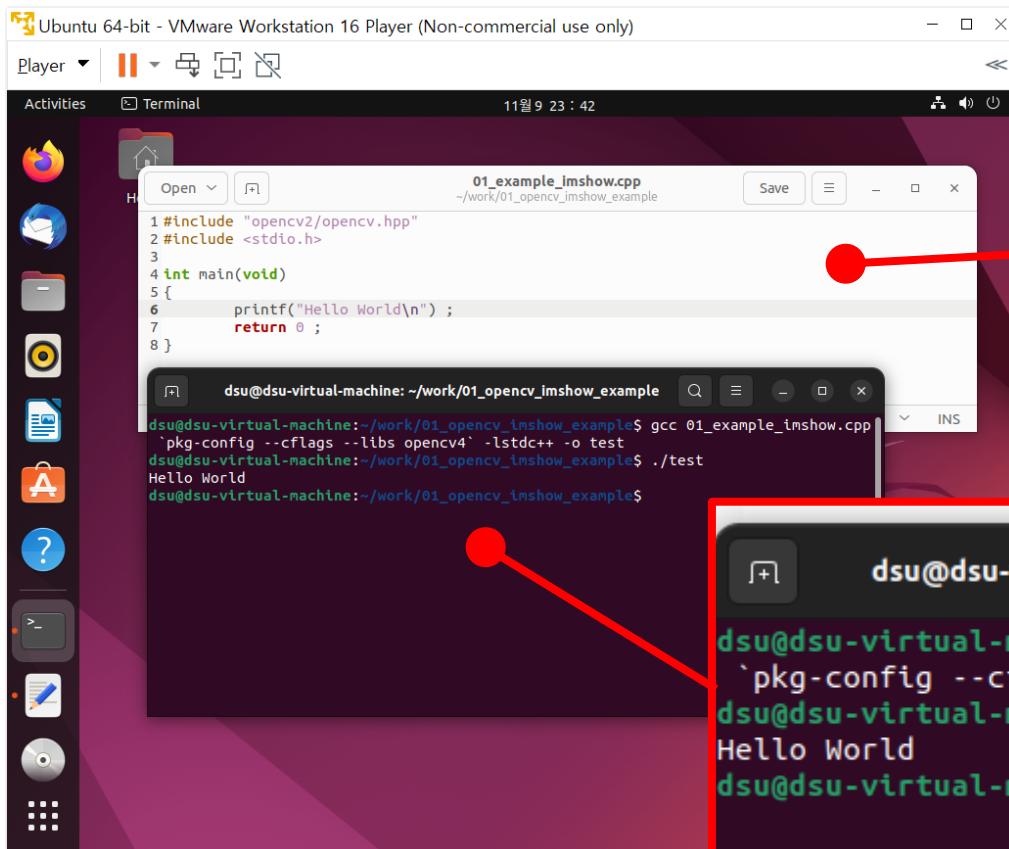
```

/home/dsu/work/01_opencv_imshow_example
dsu@dsu-virtual-machine:~/work/01_opencv_imshow_example$ gcc 01_example_imshow.cpp
`pkg-config --cflags --libs opencv4` -lstdc++
dsu@dsu-virtual-machine:~/work/01_opencv_imshow_example$ ls
01_example_imshow.cpp  a.out
dsu@dsu-virtual-machine:~/work/01_opencv_imshow_example$ gcc 01_example_imshow.cpp
`pkg-config --cflags --libs opencv4` -lstdc++ -o test
dsu@dsu-virtual-machine:~/work/01_opencv_imshow_example$ ls
01_example_imshow.cpp  a.out  test
dsu@dsu-virtual-machine:~/work/01_opencv_imshow_example$ ./test
dsu@dsu-virtual-machine:~/work/01_opencv_imshow_example$

```

OpenCV 예제 프로그램 컴파일

- 컴파일이 완료 되어 실행 파일을 만들었으니 printf를 이용하여 프로그램이 동작하는 정보를 출력해 보자.



```
1 #include "opencv2/opencv.hpp"
2 #include <stdio.h>
3
4 int main(void)
5 {
6     printf("Hello World\\n");
7     return 0;
8 }
```

```
dsu@dsu-virtual-machine: ~/work/01_opencv_imshow_example
dsu@dsu-virtual-machine:~/work/01_opencv_imshow_example$ gcc 01_example_imshow.cpp
dsu@dsu-virtual-machine:~/work/01_opencv_imshow_example$ ./test
Hello World
dsu@dsu-virtual-machine:~/work/01_opencv_imshow_example$
```

```
#include "opencv2/opencv.hpp"
#include <stdio.h>

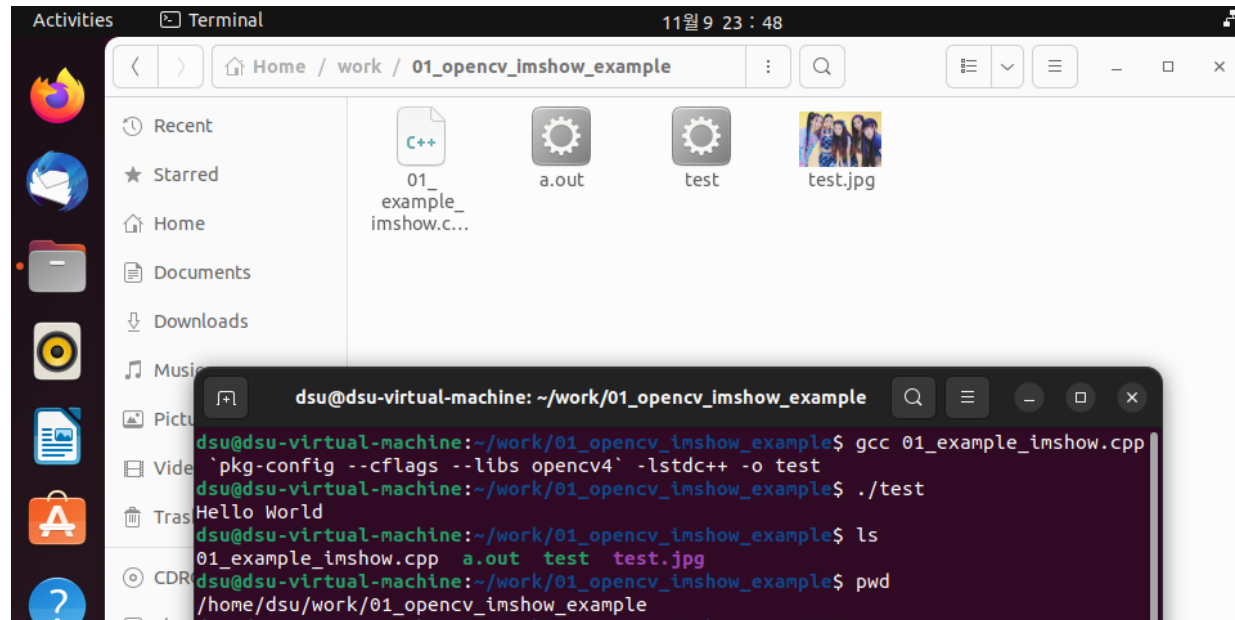
int main(void)
{
    printf("Hello World\\n");
    return 0;
}
```

```
dsu@dsu-virtual-machine: ~/work/01_opencv_imshow_example

dsu@dsu-virtual-machine:~/work/01_opencv_imshow_example$ gcc 01_example_imshow.cpp
dsu@dsu-virtual-machine:~/work/01_opencv_imshow_example$ ./test
Hello World
dsu@dsu-virtual-machine:~/work/01_opencv_imshow_example$
```

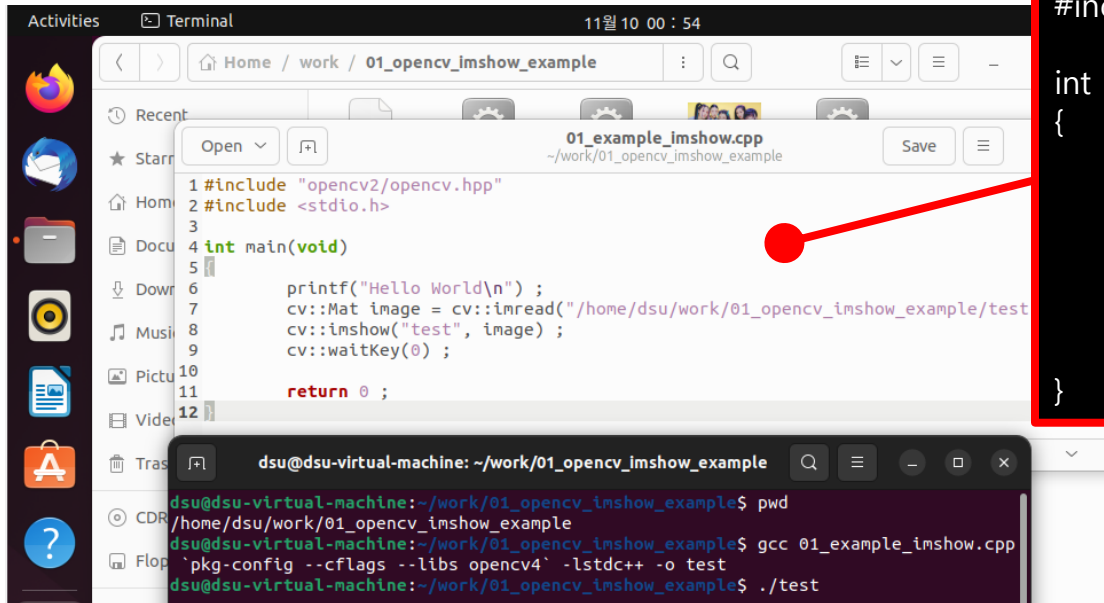

OpenCV 예제 프로그램 컴파일

- OpenCV를 이용하여 이미지 파일을 읽어 -> 출력하자!
- 샘플 이미지를 테스트 코드 경로에 복사
- 예제코드경로 : /home/dsu/work/test.jpg



OpenCV 예제 프로그램 컴파일

- OpenCV를 이용하여 이미지 파일을 읽어 -> 출력하자!
- 이미지 경로 : /home/dsu/work/test.jpg
- 위의 경로의 이미지를 Open -> Show!



```
#include "opencv2/opencv.hpp"
#include <stdio.h>

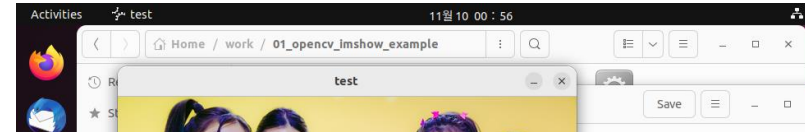
int main(void)
{
    printf("Hello World\n");
    cv::Mat imae = cv::imread("/home/dsu/work/01_opencv_imshow_example/test.jpg");
    cv::imshow("test", image);
    cv::waitKey(0);

    return 0;
}
```

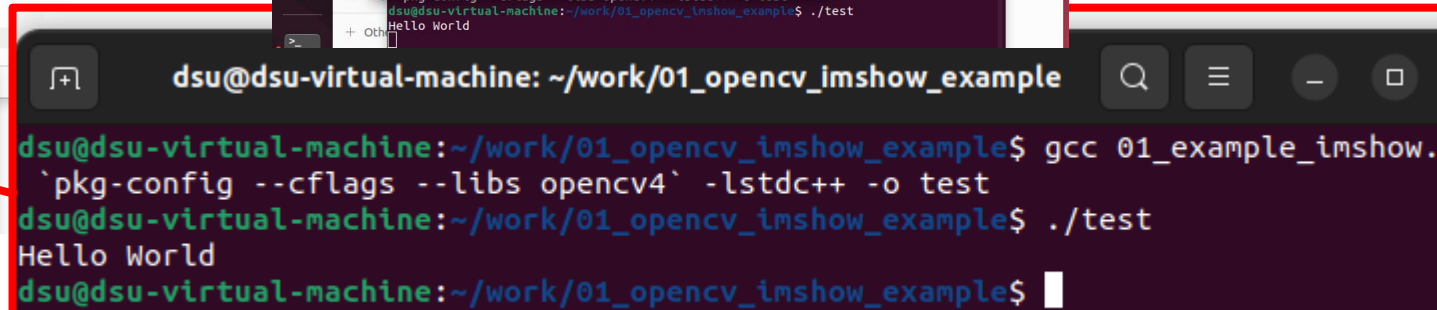
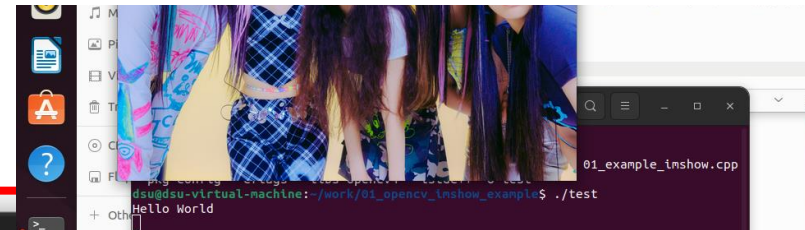
예제) 09_opencv/00_image_show.cpp

OpenCV 예제 프로그램 컴파일

- OpenCV를 이용하여 이미지 파일을 읽어 -> 출력하자!
- 이미지 경로 : /home/dsu/work/test.jpg
- 위의 경로의 이미지를 Open -> Show!



gcc 01_image_show.cpp -o test `pkg-config --cflags --libs opencv4` -lstdc++



OpenCV 예제 프로그램 컴파일

- 환영합니다! 이제 여러분은 OpenCV를 사용 가능합니다!
- 기본 적으로 제공 하는 예제 프로그램만 잘 활용해도 훌륭한 영상처리 알고리즘을 만들 수 있습니다!

