

[MS윈도우11, WSL2, Ubuntu 22.04,  
CUDA 12.1, xrdp, IDE 환경 구축]

2024-01-29(월)

정도윤

rabbit.white@daum.net

# 목차

<b>제 1장 MS윈도11 환경에서 우분투 22.04 사용하기</b>	
1.1 WSL2과 우분투 설치	5
1.2 Cuda 설치하기	8
1.3 CuDNN 설치하기	14
<b>제 2장 OpenSSH Server 서버 구축과 서비스 운영</b>	
2.1 OpenSSH Server 우분투 설치	25
2.2 OpenSSH Server 서비스 재시작	29
<b>제 3장 xfce4 데스크톱 환경과 xrdp 구축</b>	
3.1 우분투 데스크톱 환경 설치	32
3.2 xfce4 설치	33
3.3 한글 폰트, 언어 설정	35
3.4 dbus 서비스 시작	37
3.5 xrdp - startwm.sh 환경설정	38
3.6 한글 언어팩 설치	39
3.7 ibus 한글 키보드 설정	40
3.8 xrdp 부팅 서비스 활성화	45
<b>제 4장 구글 크롬</b>	
4.1 크롬 설치	47
<b>제 5장 PyCharm Community</b>	
5.1 OpenJDK 설치하기	51
5.2 PyCharm Community Edition 설치하기	55
5.3 프로젝트 생성 및 파이썬 코드 실행하기	59

## 제 6장 Visual Studio Code

6.1 Visual Studio Code 설치하기	62
6.2 Visual Studio Code 실행 과정	65

## 제 7장 Miniconda 설치하기

7.1 Miniconda 설치하기	73
7.2 bashrc 수정하기	75
7.3 conda를 활용한 jupyter 설치하기	77
7.4 Zsh, Oh-my-zsh 설치하기	79
7.5 zshrc 수정하기	82
7.6 PyTorch 설치하기	83
7.7 PyCharm Community PATH 추가 반영하기	86
7.8 Visual Studio Code에서 torch 사용 확인하기	89

# 1장 MS윈도11 환경에서 우분투 22.04 사용하기

- 1.1 WSL2과 우분투 설치
- 1.2 Cuda 설치하기
- 1.3 Cudnn 설치하기

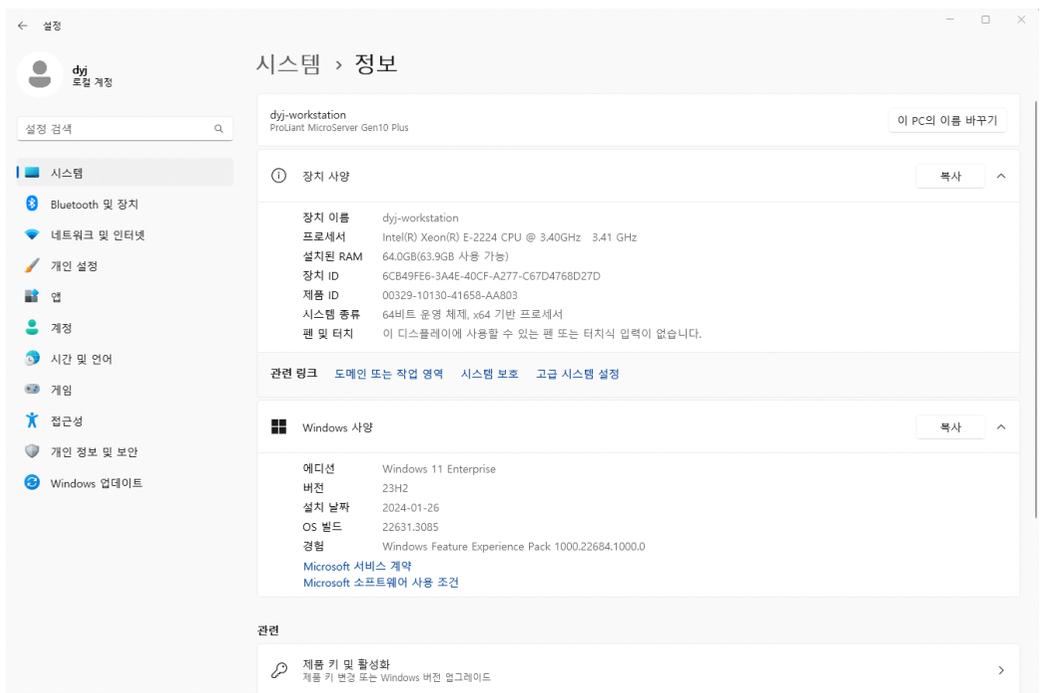
## 1.1 WSL2과 우분투 설치

### ◎ WSL2 개요

먼저 순수하게 MS윈도만 설치해서 사용하는 것과 순수하게 우분투 리눅스만을 설치해서 사용하는 것에서 벗어난 개념이라고 볼 수 있다.

지금 소개할 내용은 Windows 하위 시스템으로 리눅스를 두는 개념이라고 보면 된다. WSL2(Windows Subsystem for Linux 2)는 Linux 운영체제를 Windows에서 실행할 수 있게 해주는 기술이다. 이를 통해 개발자들은 Windows 환경에서 Linux 운영체제의 기능을 사용할 수 있다.

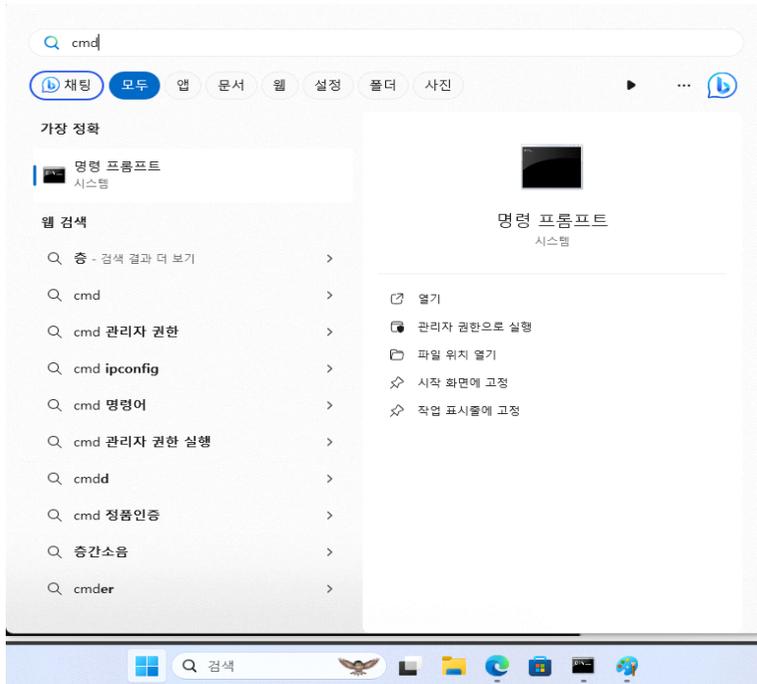
아래 그림처럼 시스템에서 정보를 클릭하면 MS윈도 운영체제 버전 정보를 확인할 수 있다.



[그림 1.1] MS윈도11 버전 확인하기

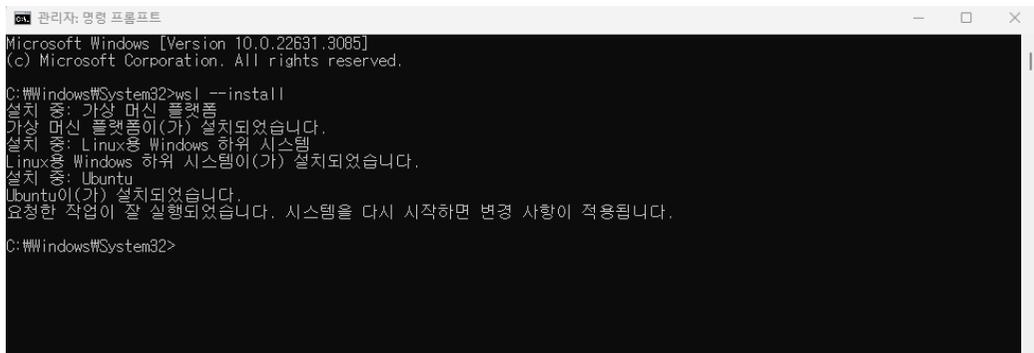
## ◎ WSL2를 명령 프롬프트로 설치하기

1단계로 시작에서 cmd를 관리자 권한으로 실행한다.

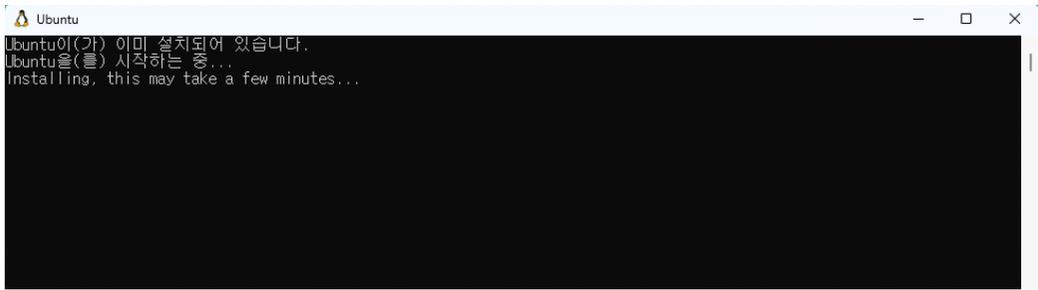


[그림 1.2] 명령 프롬프트

2단계로 `wsl --install` 명령어를 통해 WSL2를 MS윈도11에 설치할 수 있다.



[그림 1.3] WSL2 설치

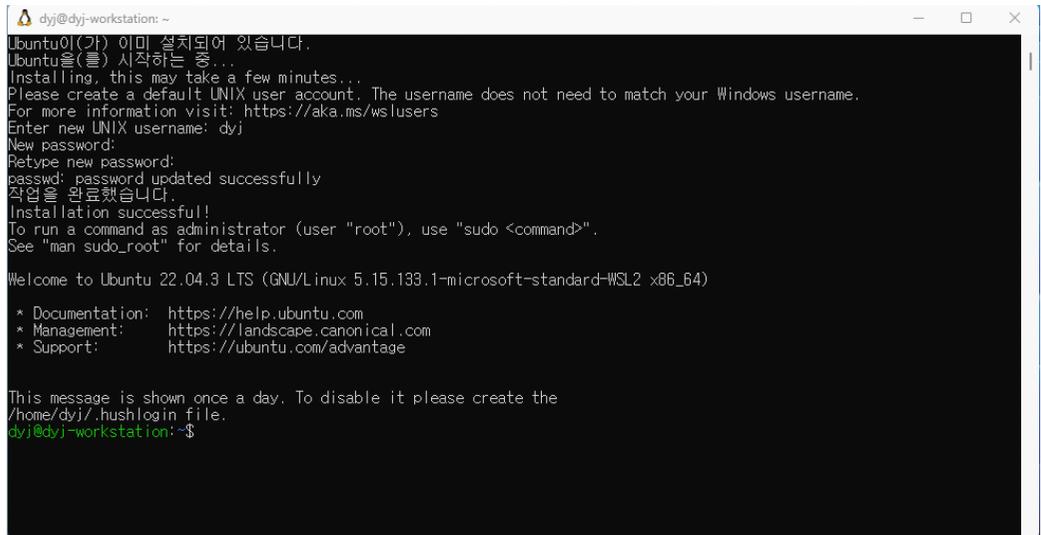


[그림 1.3] WSL2 - 우분투 22.04 시작 모습

3단계로 Enter new UNIX username이 뜨면, 계정명을 입력한다.

New password와 Retype new password를 순서대로 입력한다.

그러면, 'Installation successful!'이 뜨는 것을 볼 수 있다.

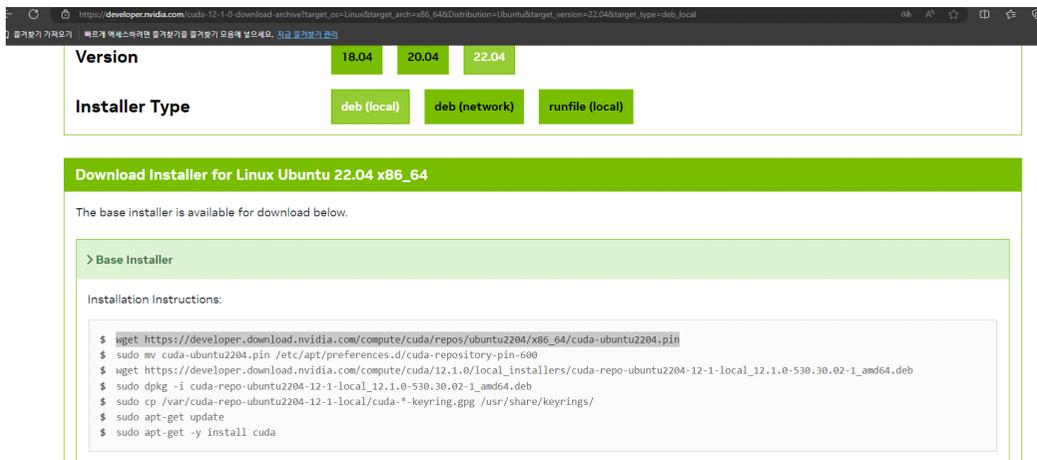


[그림 1.4] WSL2 - 우분투 22.04 시작 모습

## 1.2 Cuda 설치하기

### © CUDA 12.1 설치하기(1)

1단계로 “<https://developer.nvidia.com/cuda-downloads>”에 접속해서 Resources에서 ‘Archive of Previous CUDA Releases’를 클릭한다. Linux, x86\_64, WSL-Ubuntu, deb(local)을 클릭한다.



[그림 1.5] CUDA 12.1 공식 사이트

2단계에 설치를 따라할 때 주의할 점은 `sudo apt-get -y install cuda`에서 `sudo apt-get -y install cuda-12.1`로 버전을 명확하게 해줘야 한다.



[그림 1.6] WSL2에서 CUDA 설치 작업 모습(1)

```

TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

dyj@dyj-workstation: ~/download$ sudo apt-get install openssh-server
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  libwrap0 ncursec-term openssh-sftp-server ssh-import-id
Suggested packages:
  molly-guard monkeysphere ssh-askpass
The following NEW packages will be installed:
  libwrap0 ncursec-term openssh-server openssh-sftp-server ssh-import-id
0 upgraded, 5 newly installed, 0 to remove and 2 not upgraded.
Need to get 800 kB of archives.
After this operation, 6161 kB of additional disk space will be used.
Do you want to continue? [Y/n] Y
Get:1 http://mirror.kakao.com/ubuntu jammy-updates/main amd64 openssh-sftp-server amd64 1:8.9p1-3ubuntu0.6 [38.7 kB]
Get:2 http://mirror.kakao.com/ubuntu jammy/main amd64 libwrap0 amd64 7.6.q-31build2 [47.9 kB]
Get:3 http://mirror.kakao.com/ubuntu jammy-updates/main amd64 openssh-server amd64 1:8.9p1-3ubuntu0.6 [435 kB]
Get:4 http://mirror.kakao.com/ubuntu jammy-updates/main amd64 ncursec-term all 6.3-2ubuntu0.1 [267 kB]
Get:5 http://mirror.kakao.com/ubuntu jammy/main amd64 ssh-import-id all 5.11-0ubuntu1 [10.1 kB]
Fetched 800 kB in 1s (634 kB/s)
Preconfiguring packages ...
Selecting previously unselected package openssh-sftp-server.
(Reading database ... 24259 files and directories currently installed.)
Preparing to unpack .../openssh-sftp-server_1:8.9p1-3ubuntu0.6_amd64.deb ...
Unpacking openssh-sftp-server (1:8.9p1-3ubuntu0.6) ...
Selecting previously unselected package libwrap0:amd64.
Preparing to unpack .../libwrap0_7.6.q-31build2_amd64.deb ...
Unpacking libwrap0:amd64 (7.6.q-31build2) ...

```

[그림 1.7] WSL2에서 CUDA 설치 작업 모습(2)

```

256 SHA256: fmnTbn00ifjHexGawSFBk8m0ANSyY86bqot9P3tb0c root@dyj-workstation (E025519)
Created symlink /etc/systemd/system/ssh.service → /lib/systemd/system/ssh.service.
Created symlink /etc/systemd/system/multi-user.target.wants/ssh.service → /lib/systemd/system/ssh.service.
rescue-ssh.target is a disabled or a static unit, not starting it.
ssh.socket is a disabled or a static unit, not starting it.
Processing triggers for ufw (0.36.1-4ubuntu0.1) ...
Processing triggers for man-db (2.10.2-1) ...
Processing triggers for libc-bin (2.35-0ubuntu3.6) ...
/usr/lib/ldconfig.real: /usr/lib/wsl/lib/libcuda.so.1 is not a symbolic link

dyj@dyj-workstation: ~/download$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 172.29.54.41 netmask 255.255.240.0 broadcast 172.29.63.255
    inet6 fe80::215:5dff:fe48:8ef4 prefixlen 64 scopeid 0x20<link>
    ether 00:15:5d:48:8e:f4 txqueuelen 1000 (Ethernet)
    RX packets 73861 bytes 113649177 (113.6 MB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 36636 bytes 2468362 (2.4 MB)
    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 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

dyj@dyj-workstation: ~/download$

```

[그림 1.8] WSL2에서 CUDA 설치 작업 모습(3)

그림 1.8에서 ifconfig 명령어가 안 되는 경우에는

```

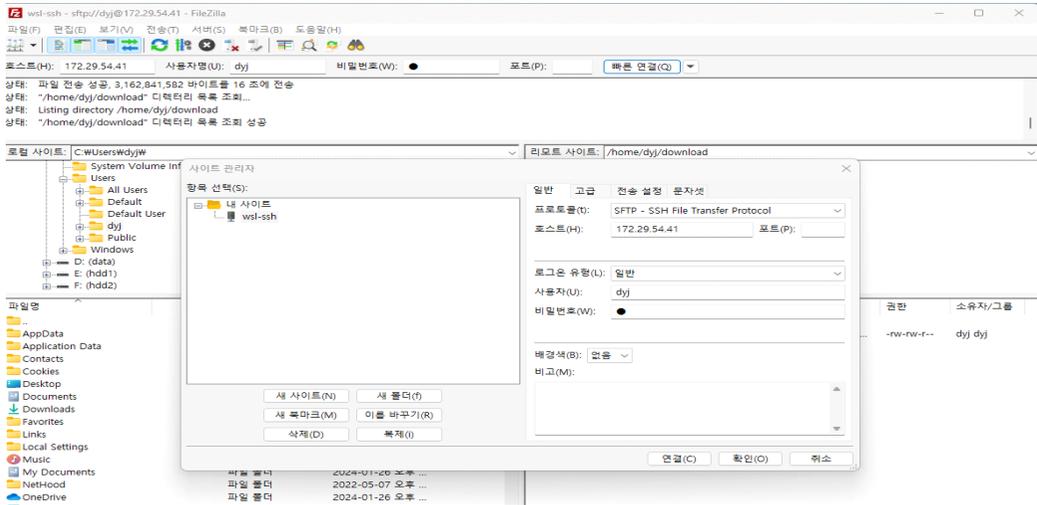
sudo apt-get install net-tools

```

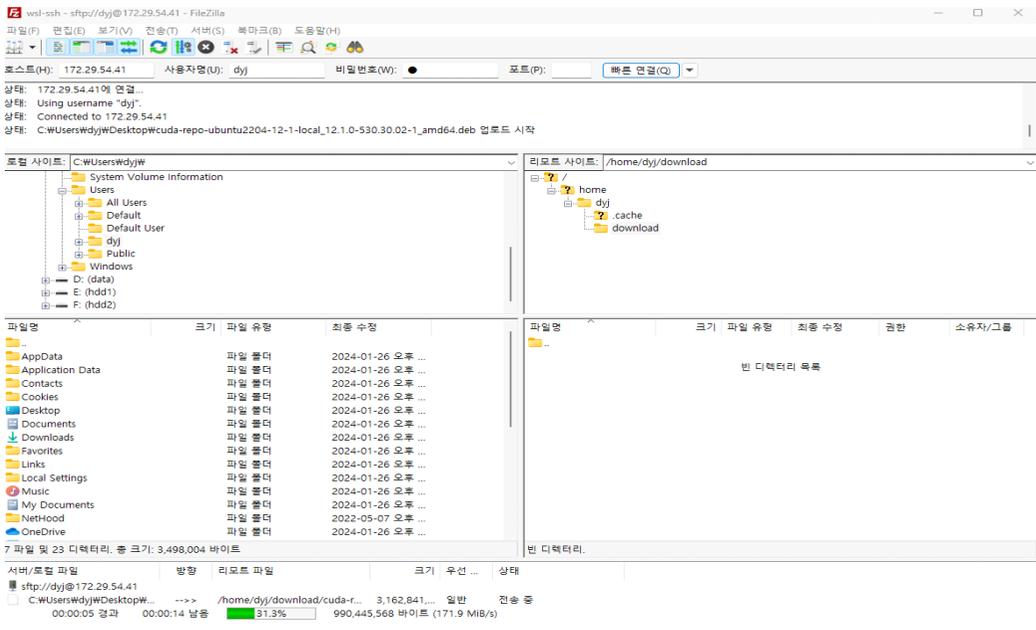
를 통해 설치할 수 있다.

## ▷ CUDA 12.1 - 팁(TIP)

네트워크 재배포가 가능한 패키지를 다운로드 받았기 때문에 SSH나 SFTP를 통해 Cuda 설치파일을 배포할 수 있다.

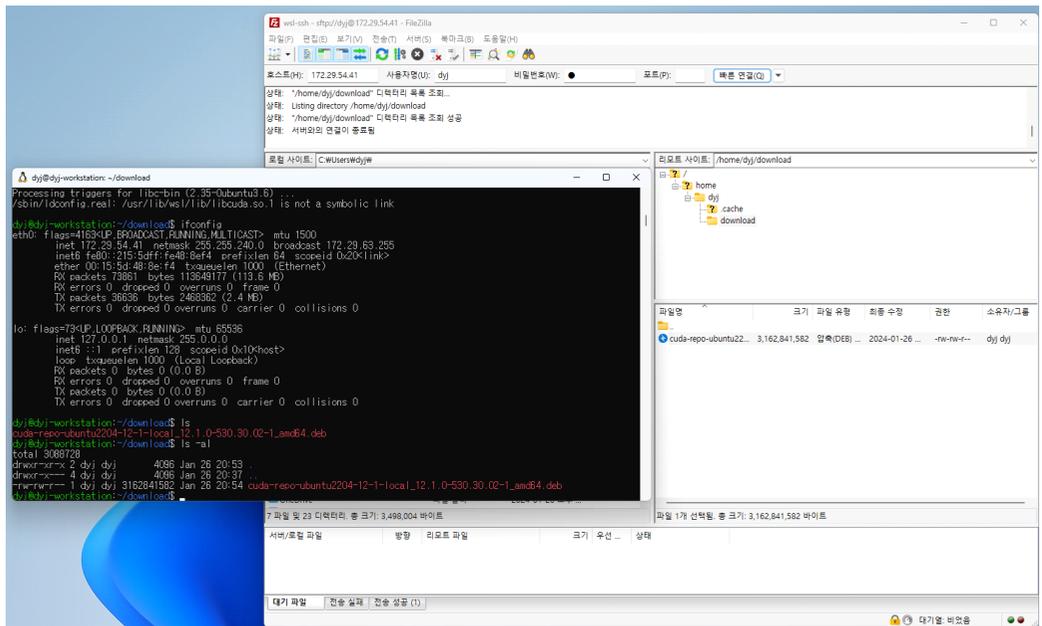


[그림 1.9] WSL2(SFTP)로 로컬 PC 통신하기(1)



[그림 1.10] WSL2(SFTP)로 로컬 PC 통신하기(2)

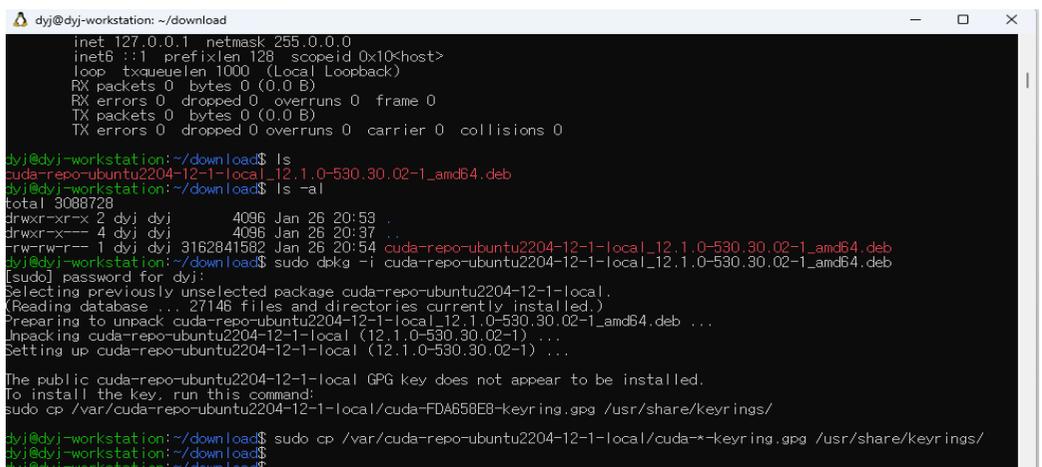
아래 그림은 cuda 설치파일이 전송이 완료된 모습을 살펴볼 수 있다.



[그림 1.11] WSL2(SFTP)로 로컬 PC 통신하기(3)

### ◎ CUDA 12.1 설치하기(2)

‘CUDA 12.1 설치하기(1)’에 이어서 설치 과정을 진행하는 모습이다.



[그림 1.12] WSL2에서 cuda 설치 작업 모습

아래 그림은 앞서 언급한 것처럼, cuda-12.1로 명확하게 버전을 명시한 것이다. 참고로 설치하는 데 오래 걸리므로 기다린다.

```

dyj@dyj-workstation: ~/download
Get:10 http://mirror.kakao.com/ubuntu jammy-updates/main amd64 cpp-11 amd64 11.4.0-1ubuntu1~22.04 [10.0 MB]
1% [10 cpp-11 6013 B/10.0 MB 0%] ^C
dyj@dyj-workstation: ~/download$ sudo apt-get install cuda-12.1
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Note, selecting 'cuda-12-1' for regex 'cuda-12.1'
The following additional packages will be installed:
  adwaita-icon-theme alsa-topology-conf alsa-ucm-conf at-spi2-core build-essential bzip2 ca-certificates-java cpp
  cpp-11 cpp-12 cuda-cccl-12-1 cuda-command-line-tools-12-1 cuda-compiler-12-1 cuda-cudart-12-1 cuda-cudart-dev-12-1
  cuda-cuobjdump-12-1 cuda-cupti-12-1 cuda-cupti-dev-12-1 cuda-cuxxfilt-12-1 cuda-demo-suite-12-1
  cuda-documentation-12-1 cuda-driver-dev-12-1 cuda-drivers cuda-drivers-530 cuda-gdb-12-1 cuda-libraries-12-1
  cuda-libraries-dev-12-1 cuda-nsight-12-1 cuda-nsight-compute-12-1 cuda-nsight-systems-12-1 cuda-nvcc-12-1
  cuda-nvdisasm-12-1 cuda-nvml-dev-12-1 cuda-nvprof-12-1 cuda-nvprune-12-1 cuda-nvrtc-12-1 cuda-nvrtc-dev-12-1
  cuda-nvtx-12-1 cuda-nvvp-12-1 cuda-opengl-12-1 cuda-opengl-dev-12-1 cuda-profiler-api-12-1 cuda-runtime-12-1
  cuda-sanitizer-12-1 cuda-toolkit-12-1 cuda-toolkit-12-1-config-common cuda-toolkit-12-config-common
  cuda-toolkit-config-common cuda-tools-12-1 cuda-visual-tools-12-1 dconf-gsettings-backend dconf-service dctrl-tools
  default-jre default-jre-headless dkms dkms-dev fakeroot fontconfig fontconfig-config fonts-dejavu-core
  fonts-dejavu-extra g++ g++-11 gcc-11 gcc-11-base gcc-12 gds-tools-12-1 gsettings-desktop-schemas
  gtk-update-icon-cache hicolor-icon-theme humanity-icon-theme java-common libalgorithm-diff-perl
  libalgorithm-diff-xs-perl libalgorithm-merge-perl libasan6 libasan8 libasound2 libasound2-data libatk-bridge2.0
  libatk-wrapper-java libatk-wrapper-java-jni libatk1.0-0 libatk1.0-data libatomic1 libatspi2.0-0 libavahi-client3
  libavahi-common-data libavahi-common3 libc-dev-bin libc-devtools libc6-dev libcairo-gobject2 libcairo2 libcc1-0
  libcolord2 libcrypt-dev libcublas-12-1 libcublas-dev-12-1 libcufft-12-1 libcufft-dev-12-1 libcufile-12-1
  libcufile-dev-12-1 libcups2 libcurand-12-1 libcurand-dev-12-1 libcusolver-12-1 libcusolver-dev-12-1 libcuspars-12-1
  libcuspars-dev-12-1 libdatrie1 libdconf1 libdeflate0 libdkg-perl libdrm-amdgpu libdrm-intel1 libdrm-nouveau2
  libdrm-radeon1 libegl-mesa0 libegl1 libepoxy0 libfakeroot libfile-fcntllock-perl libfontconfig1 libfontenc1
  libfreeptep6 libgbm1 libgcc-11-dev libgcc-12-dev libg3 libgdk-pixbuf-2.0-0 libgdk-pixbuf2.0-bin
  libgdk-pixbuf2.0-common libgl1 libgl1-amd-gli libgl1-mesa-dri libgl1-mesa-dri libglapi-mesa libgles2 libglvnd0
  libglx-mesa0 libglx0 libgomp1 libgraphite2-3 libatk-3-0 libatk-3-bin libatk-3-common libharfbuzz0b libice6 libis123

```

[그림 1.13] WSL2에서 cuda 설치 작업 모습

```

dyj@dyj-workstation: ~/download
Setting up cuda-visual-tools-12-1 (12.1.0-1) ...
Setting up cuda-tools-12-1 (12.1.0-1) ...
Setting up cuda-toolkit-12-1 (12.1.0-1) ...
Processing triggers for ca-certificates (20230811ubuntu0.22.04.1) ...
Updating certificates in /etc/ssl/certs...
0 added, 0 removed; done.
Running hooks in /etc/ca-certificates/update.d...
done.
done.
Processing triggers for dbus (1.12.20-2ubuntu4.1) ...
Processing triggers for libgl1-mesa0:amd64 (2.72.4-0ubuntu2.2) ...
Setting up libatk-3-0:amd64 (3.24.33-1ubuntu2) ...
Processing triggers for libc-bin (2.35-0ubuntu3.6) ...
/sbin/ldconfig.real: /usr/lib/wsl/lib/libcuda.so.1 is not a symbolic link

Setting up libatk-3-bin (3.24.33-1ubuntu2) ...
Processing triggers for man-db (2.10.2-1) ...
Setting up at-spi2-core (2.44.0-3) ...
Setting up nvidia-settings (530.30.02-0ubuntu1) ...
Setting up cuda-drivers-530 (530.30.02-1) ...
Setting up cuda-drivers (530.30.02-1) ...
Setting up cuda-runtime-12-1 (12.1.0-1) ...
Setting up cuda-demo-suite-12-1 (12.1.55-1) ...
Setting up cuda-12-1 (12.1.0-1) ...
Processing triggers for libgdk-pixbuf-2.0-0:amd64 (2.42.8+dfsg-1ubuntu0.2) ...
Processing triggers for libc-bin (2.35-0ubuntu3.6) ...
/sbin/ldconfig.real: /usr/lib/wsl/lib/libcuda.so.1 is not a symbolic link
dyj@dyj-workstation: ~/download$

```

[그림 1.14] WSL2에서 cuda 설치 작업 모습

설치 명령어를 정리하면 다음과 같다.

**설치 명령어**

**Download Installer for Linux Ubuntu 22.04 x86\_64**

The base installer is available for download below.

> Base Installer

Installation Instructions:

```
$ wget https://developer.download.nvidia.com/compute/cuda/repos/ubuntu2204/x86_64/cuda-ubuntu2204.pin
$ sudo mv cuda-ubuntu2204.pin /etc/apt/preferences.d/cuda-repository-pin-600
$ wget https://developer.download.nvidia.com/compute/cuda/12.1.0/local_installers/cuda-repo-ubuntu2204-12-1-local_12.1.0-530.30.02-1_amd64.deb
$ sudo dpkg -i cuda-repo-ubuntu2204-12-1-local_12.1.0-530.30.02-1_amd64.deb
$ sudo cp /var/cuda-repo-ubuntu2204-12-1-local/cuda-*-keyring.gpg /usr/share/keyrings/
$ sudo apt-get update
$ sudo apt-get -y install cuda
```

The CUDA Toolkit contains Open-Source Software. The source code can be found [here](#).  
The checksums for the installer and patches can be found in [Installer Checksums](#).  
For further information, see the [Installation Guide for Linux](#) and the [CUDA Quick Start Guide](#).

[그림 1.15] CUDA 12.1 공식 사이트 설치 명령어 정리

wget

[https://developer.download.nvidia.com/compute/cuda/repos/ubuntu2204/x86\\_64/cuda-ubuntu2204.pin](https://developer.download.nvidia.com/compute/cuda/repos/ubuntu2204/x86_64/cuda-ubuntu2204.pin)

sudo mv cuda-ubuntu2204.pin

/etc/apt/preferences.d/cuda-repository-pin-600

wget

[https://developer.download.nvidia.com/compute/cuda/12.1.0/local\\_installers/cuda-repo-ubuntu2204-12-1-local\\_12.1.0-530.30.02-1\\_amd64.deb](https://developer.download.nvidia.com/compute/cuda/12.1.0/local_installers/cuda-repo-ubuntu2204-12-1-local_12.1.0-530.30.02-1_amd64.deb)

sudo dpkg -i

cuda-repo-ubuntu2204-12-1-local\_12.1.0-530.30.02-1\_amd64.deb

sudo cp /var/cuda-repo-ubuntu2204-12-1-local/cuda-\*-keyring.gpg

/usr/share/keyrings/

sudo apt-get update

sudo apt-get -y install cuda-12.1

### 1.3 CuDNN 설치하기

◎ CuDNN 8.96 - CUDA 12.1 설치하기(1)

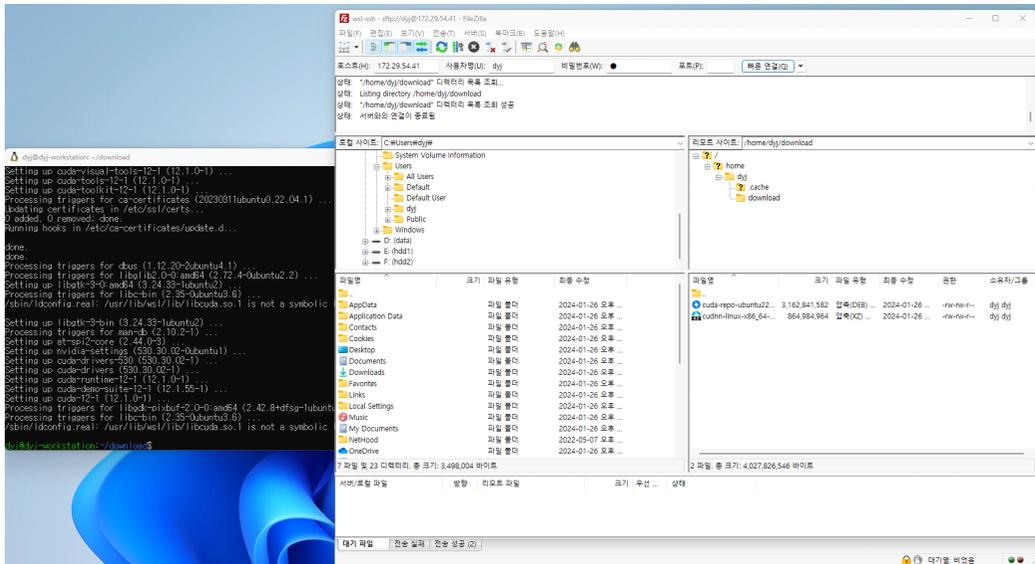
웹 브라우저에서 아래 사이트에 접속한다.

<https://developer.nvidia.com/cudnn>

로그인을 진행한다. 계정이 없는 경우라면, JOIN을 클릭해 회원가입을 한다. 이메일 계정으로 로그인을 시도하면, 보안 로그인이라고 해서 이메일 인증을 받으라고 한다. 보안 이메일 인증을 시도할 때 주의할 점은 로그인을 시도하는 이메일 계정과 같은 PC에서만 로그인 인증이 가능하다.

cuDNN을 Windows zip 파일이 아니라, “Local Installer for Linux x86\_64 (Tar)”을 클릭하여 다운로드한다. 그리고 파일질라(이하 “Filezilla”) 프로그램을 통해 SFTP로 WSL2에 접속한다.

파일을 업로드 해준다.



[그림 1.16] WSL2(SSH)에서 cudnn 설치하기(1)

다음 명령으로 압축을 풀어준다. 압축파일 이름은 글에 있는 이름과 다를 수 있으니 유의해야 한다.

tar xvf cudnn을 입력 후, tab키를 누르면 파일 이름이 완성되는 것을 사용하면 편하다.

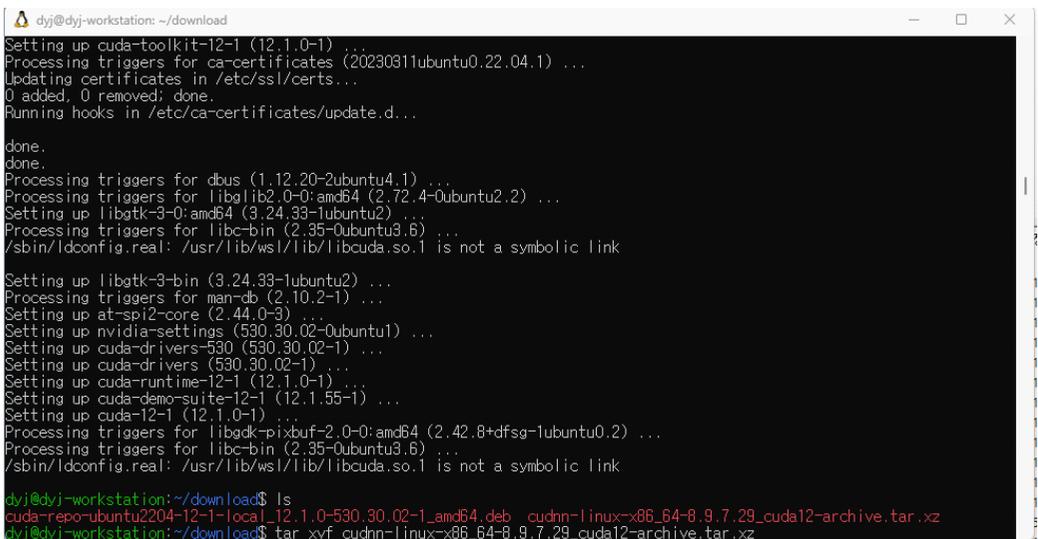
```
tar xvf cudnn-linux-x86_64-8.9.7.29_cuda12-archive.tar.xz
```

다음 명령으로 필요한 파일을 복사해준다.

```
sudo cp cudnn-*-archive/include/cudnn*.h /usr/local/cuda/include
```

```
sudo cp -P cudnn-*-archive/lib/libcudnn* /usr/local/cuda/lib64
```

```
sudo chmod a+r /usr/local/cuda/include/cudnn*.h  
/usr/local/cuda/lib64/libcudnn*
```



[그림 1.17] WSL2에서 cudnn 설치하기(2)

아래 그림은 설치 모습이다.

```

dyj@dyj-workstation: ~/download
cudnn-linux-x86_64-8.9.7.29_cuda12-archive/lib/libcudnn_cnn_train.so
cudnn-linux-x86_64-8.9.7.29_cuda12-archive/lib/libcudnn_cnn_train.so.8.9.7
cudnn-linux-x86_64-8.9.7.29_cuda12-archive/lib/libcudnn_cnn_train.so.8
cudnn-linux-x86_64-8.9.7.29_cuda12-archive/lib/libcudnn_ops_infer.so.8.9.7
cudnn-linux-x86_64-8.9.7.29_cuda12-archive/lib/libcudnn_ops_infer.so
cudnn-linux-x86_64-8.9.7.29_cuda12-archive/lib/libcudnn_ops_infer.so.8
cudnn-linux-x86_64-8.9.7.29_cuda12-archive/lib/libcudnn_ops_train.so.8.9.7
cudnn-linux-x86_64-8.9.7.29_cuda12-archive/lib/libcudnn_ops_train.so
cudnn-linux-x86_64-8.9.7.29_cuda12-archive/lib/libcudnn_ops_train.so.8
cudnn-linux-x86_64-8.9.7.29_cuda12-archive/include/
cudnn-linux-x86_64-8.9.7.29_cuda12-archive/include/cudnn_v8.h
cudnn-linux-x86_64-8.9.7.29_cuda12-archive/include/cudnn_adv_infer_v8.h
cudnn-linux-x86_64-8.9.7.29_cuda12-archive/include/cudnn_adv_train_v8.h
cudnn-linux-x86_64-8.9.7.29_cuda12-archive/include/cudnn_backend_v8.h
cudnn-linux-x86_64-8.9.7.29_cuda12-archive/include/cudnn_cnn_infer_v8.h
cudnn-linux-x86_64-8.9.7.29_cuda12-archive/include/cudnn_cnn_train_v8.h
cudnn-linux-x86_64-8.9.7.29_cuda12-archive/include/cudnn_ops_infer_v8.h
cudnn-linux-x86_64-8.9.7.29_cuda12-archive/include/cudnn_ops_train_v8.h
cudnn-linux-x86_64-8.9.7.29_cuda12-archive/include/cudnn_version_v8.h
cudnn-linux-x86_64-8.9.7.29_cuda12-archive/include/cudnn.h
cudnn-linux-x86_64-8.9.7.29_cuda12-archive/include/cudnn_adv_infer.h
cudnn-linux-x86_64-8.9.7.29_cuda12-archive/include/cudnn_adv_train.h
cudnn-linux-x86_64-8.9.7.29_cuda12-archive/include/cudnn_backend.h
cudnn-linux-x86_64-8.9.7.29_cuda12-archive/include/cudnn_cnn_infer.h
cudnn-linux-x86_64-8.9.7.29_cuda12-archive/include/cudnn_cnn_train.h
cudnn-linux-x86_64-8.9.7.29_cuda12-archive/include/cudnn_ops_infer.h
cudnn-linux-x86_64-8.9.7.29_cuda12-archive/include/cudnn_ops_train.h
cudnn-linux-x86_64-8.9.7.29_cuda12-archive/include/cudnn_version.h
cudnn-linux-x86_64-8.9.7.29_cuda12-archive/LICENSE
dyj@dyj-workstation: ~/download$

```

[그림 1.18] WSL2에서 cudnn 설치하기(3)

```

dyj@dyj-workstation: ~/download/cudnn-linux-x86_64-8.9.7.29_cuda12-archive
cudnn-linux-x86_64-8.9.7.29_cuda12-archive/include/cudnn.h
cudnn-linux-x86_64-8.9.7.29_cuda12-archive/include/cudnn_adv_infer.h
cudnn-linux-x86_64-8.9.7.29_cuda12-archive/include/cudnn_adv_train.h
cudnn-linux-x86_64-8.9.7.29_cuda12-archive/include/cudnn_backend.h
cudnn-linux-x86_64-8.9.7.29_cuda12-archive/include/cudnn_cnn_infer.h
cudnn-linux-x86_64-8.9.7.29_cuda12-archive/include/cudnn_cnn_train.h
cudnn-linux-x86_64-8.9.7.29_cuda12-archive/include/cudnn_ops_infer.h
cudnn-linux-x86_64-8.9.7.29_cuda12-archive/include/cudnn_ops_train.h
cudnn-linux-x86_64-8.9.7.29_cuda12-archive/include/cudnn_version.h
cudnn-linux-x86_64-8.9.7.29_cuda12-archive/LICENSE
dyj@dyj-workstation: ~/download$ ls -al
total 3933448
drwxr-xr-x 3 dyj dyj 4096 Jan 26 21:06 .
drwxr-xr-x 4 dyj dyj 4096 Jan 26 20:37 ..
-rw-rw-r-- 1 dyj dyj 3162841582 Jan 26 20:54 cuda-repo-ubuntu2204-12-1-local_12.1.0-530.30.02-1_amd64.deb
drwxr-xr-x 4 dyj dyj 4096 Nov 30 17:38 cudnn-linux-x86_64-8.9.7.29_cuda12-archive
-rw-rw-r-- 1 dyj dyj 864984964 Jan 26 21:00 cudnn-linux-x86_64-8.9.7.29_cuda12-archive.tar.xz
dyj@dyj-workstation: ~/download$ cd cudnn-linux-x86_64-8.9.7.29_cuda12-archive/
dyj@dyj-workstation: ~/download/cudnn-linux-x86_64-8.9.7.29_cuda12-archive$ ls
LICENSE include lib
dyj@dyj-workstation: ~/download/cudnn-linux-x86_64-8.9.7.29_cuda12-archive$ sudo cp include/cudnn*.h /usr/local/cuda/incl
ude
dyj@dyj-workstation: ~/download/cudnn-linux-x86_64-8.9.7.29_cuda12-archive$ sudo cp -P lib/libcudnn* /usr/local/cuda/lib64
dyj@dyj-workstation: ~/download/cudnn-linux-x86_64-8.9.7.29_cuda12-archive$ sudo chmod a+r /usr/local/cuda/include/cudnn*
.h /usr/local/cuda/lib64/
Display all 171 possibilities? (y or n)
dyj@dyj-workstation: ~/download/cudnn-linux-x86_64-8.9.7.29_cuda12-archive$ sudo chmod a+r /usr/local/cuda/include/cudnn*
.h /usr/local/cuda/lib64/libcudnn*
dyj@dyj-workstation: ~/download/cudnn-linux-x86_64-8.9.7.29_cuda12-archive$

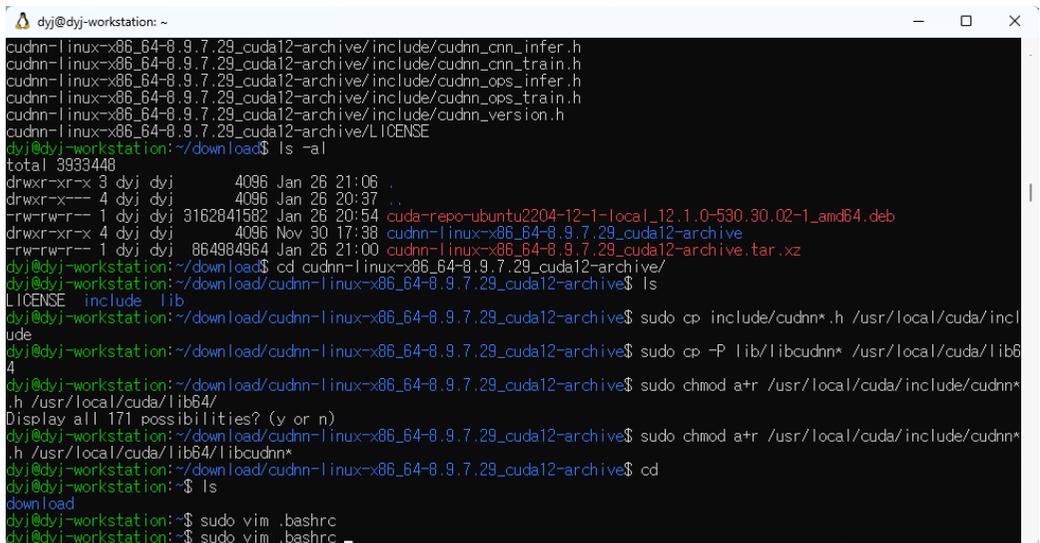
```

[그림 1.19] WSL2에서 cudnn 설치하기(4)

## ▷ Bash 셸 파일에 Cuda 추가하기

.bashrc 또는 .zshrc와 같은 사용자가 사용하는 셸 환경설정 파일에 cuda의 위치를 식별해주는 작업을 수행해야 한다. cd /usr/local/ 입력 후 tab키로 여러 번 눌러보면 cuda-12.1 폴더가 존재하는지 확인할 수 있다.

해당 경로를 바탕으로 홈 디렉터리(/home/사용자계정)에 있는 .bashrc 또는 .zshrc 파일의 하단에 수정하면 된다.



```
dyj@dyj-workstation: ~  
dyj@dyj-workstation:~/download$ ls -al  
total 3933448  
drwxr-xr-x 3 dyj dyj 4096 Jan 26 21:06 .  
drwxr-xr-x 4 dyj dyj 4096 Jan 26 20:37 ..  
-r--r--r-- 1 dyj dyj 3162841582 Jan 26 20:54 cuda-repo-ubuntu2204-12-1-local_12.1.0-530.30.02-1_amd64.deb  
drwxr-xr-x 4 dyj dyj 4096 Nov 30 17:38 cudnn-linux-x86_64-8.9.7.29_cuda12-archive  
-r--r--r-- 1 dyj dyj 864984964 Jan 26 21:00 cudnn-linux-x86_64-8.9.7.29_cuda12-archive.tar.xz  
dyj@dyj-workstation:~/download$ cd cudnn-linux-x86_64-8.9.7.29_cuda12-archive/  
dyj@dyj-workstation:~/download/cudnn-linux-x86_64-8.9.7.29_cuda12-archive$ ls  
LICENSE include lib  
dyj@dyj-workstation:~/download/cudnn-linux-x86_64-8.9.7.29_cuda12-archive$ sudo cp include/cudnn*.h /usr/local/cuda/incl  
ude  
dyj@dyj-workstation:~/download/cudnn-linux-x86_64-8.9.7.29_cuda12-archive$ sudo cp -P lib/libcudnn* /usr/local/cuda/lib6  
4  
dyj@dyj-workstation:~/download/cudnn-linux-x86_64-8.9.7.29_cuda12-archive$ sudo chmod a+r /usr/local/cuda/include/cudnn*  
.h /usr/local/cuda/lib64/  
Display all 171 possibilities? (y or n)  
dyj@dyj-workstation:~/download/cudnn-linux-x86_64-8.9.7.29_cuda12-archive$ sudo chmod a+r /usr/local/cuda/include/cudnn*  
.h /usr/local/cuda/lib64/libcudnn*  
dyj@dyj-workstation:~/download/cudnn-linux-x86_64-8.9.7.29_cuda12-archive$ cd  
dyj@dyj-workstation:~$ ls  
download  
dyj@dyj-workstation:~$ sudo vim .bashrc  
dyj@dyj-workstation:~$ sudo vim .bashrc
```

[그림 1.20] WSL2에서 CUDA 12.1 - 셸 파일에 한 줄 추가하기(1)



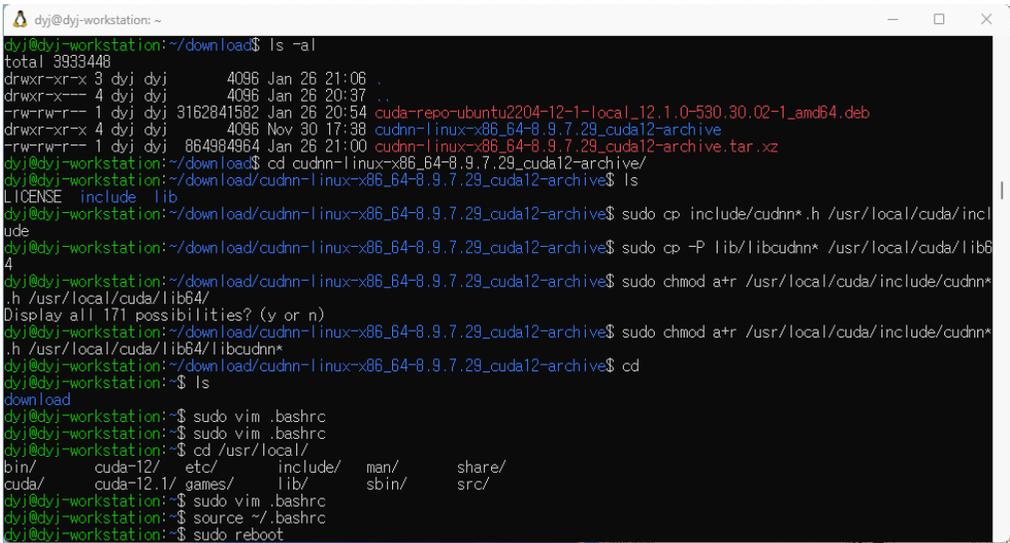
```
export
```

```
LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/usr/local/cuda-12.1/lib64/
```

입력 후, ESC키를 누른다.

:wq라고 입력 후 Enter 키를 누른다.

다음으로 source ~/.bashrc 또는 source ~/.zshrc 라고 입력한다.



```
dyj@dyj-workstation: ~  
dyj@dyj-workstation: ~/download$ ls -al  
total 3939448  
drwxr-xr-x 3 dyj dyj 4096 Jan 26 21:06 .  
drwxr-xr-x 4 dyj dyj 4096 Jan 26 20:37 ..  
-rw-rw-r-- 1 dyj dyj 3162841582 Jan 26 20:54 cuda-repo-ubuntu2204-12-1-local_12.1.0-530.30.02-1_amd64.deb  
drwxr-xr-x 4 dyj dyj 4096 Nov 30 17:38 cudnn-linux-x86_64-8.9.7.29_cuda12-archive  
drwxr-xr-x 1 dyj dyj 864984964 Jan 26 21:00 cudnn-linux-x86_64-8.9.7.29_cuda12-archive.tar.xz  
dyj@dyj-workstation: ~/download$ cd cudnn-linux-x86_64-8.9.7.29_cuda12-archive/  
dyj@dyj-workstation: ~/download/cudnn-linux-x86_64-8.9.7.29_cuda12-archive$ ls  
LICENSE include lib  
dyj@dyj-workstation: ~/download/cudnn-linux-x86_64-8.9.7.29_cuda12-archive$ sudo cp include/cudnn*.h /usr/local/cuda/incl  
ude  
dyj@dyj-workstation: ~/download/cudnn-linux-x86_64-8.9.7.29_cuda12-archive$ sudo cp -P lib/libcudnn* /usr/local/cuda/lib6  
4  
dyj@dyj-workstation: ~/download/cudnn-linux-x86_64-8.9.7.29_cuda12-archive$ sudo chmod a+r /usr/local/cuda/include/cudnn*  
.h /usr/local/cuda/lib64/  
Display all 171 possibilities? (y or n)  
dyj@dyj-workstation: ~/download/cudnn-linux-x86_64-8.9.7.29_cuda12-archive$ sudo chmod a+r /usr/local/cuda/include/cudnn*  
.h /usr/local/cuda/lib64/libcudnn*  
dyj@dyj-workstation: ~/download/cudnn-linux-x86_64-8.9.7.29_cuda12-archive$ cd  
dyj@dyj-workstation: ~$ ls  
download  
dyj@dyj-workstation: ~$ sudo vim .bashrc  
dyj@dyj-workstation: ~$ sudo vim .bashrc  
dyj@dyj-workstation: ~$ cd /usr/local/  
bin/      cuda-12/  etc/      include/  man/      share/  
cuda/     cuda-12.1/  games/   lib/      sbin/     src/  
dyj@dyj-workstation: ~$ sudo vim .bashrc  
dyj@dyj-workstation: ~$ source ~/.bashrc  
dyj@dyj-workstation: ~$ sudo reboot
```

[그림 1.23] WSL2에서 CUDA 12.1 - 셸 파일에 한 줄 추가하기(3)

### .zshrc에서 오류가 나는 경우

ZSH 셸을 설치하지 않은 경우에는 오류가 발생하므로 bashrc에 적용해도 된다.

◎ WSL2(Ubuntu 22.04)에서 Cuda 설치 확인하기

다음 명령어를 입력함으로써 cuda가 정상적으로 설치되었는지를 확인할 수 있다. nvidia-smi은 실제 물리적인 우분투 환경에서 Cuda를 설치한 경우라면 직관적으로 확인 가능한 명령어이다. WSL2에서는 Cuda를 설치하지 않은 경우에도 동작하기 때문에 큰 의미가 없다.

```
dyj@dyj-workstation: ~  
dyj@dyj-workstation:~$ nvidia-smi  
Fri Jan 26 21:14:49 2024  
+-----+  
| NVIDIA-SMI 530.30.02              Driver Version: 531.14      CUDA Version: 12.1     |  
+-----+-----+-----+-----+-----+-----+  
| GPU   Name                Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |  
| Fan  Temp  Perf            Pwr:Usage/Cap|     Memory-Usage | GPU-Util  Compute M. |  
|-----+-----+-----+-----+-----+-----+  
|    0   NVIDIA T1000 8GB           On          00000000:07:00:0   On      | 693MiB / 8192MiB      1%      Default  |  
|              N/A / 50W           693MiB / 8192MiB      |              N/A      N/A      |  
+-----+-----+-----+-----+-----+-----+  
| Processes:                                                                  |  
|  GPU   GI   CI        PID   Type   Process name                      GPU Memory |  
|  ID   ID   ID           |                 | Usage     |  
+-----+-----+-----+-----+-----+-----+  
|    0   N/A  N/A         253    G    /Xwayland                          N/A       |  
+-----+-----+-----+-----+-----+-----+  
dyj@dyj-workstation:~$
```

[그림 1.24] WSL2에서 CUDA 12.1 - 설치 확인하기(1)

다음 명령어를 통해 CUDA 디바이스를 쿼리로 조회해볼 수 있다.

```
/usr/local/cuda-12.1/extras/demo_suite/deviceQuery
```

```

djy@djy-workstation: ~
djy@djy-workstation:~$ /usr/local/cuda-12.1/extras/demo_suite/deviceQuery
/usr/local/cuda-12.1/extras/demo_suite/deviceQuery Starting...

CUDA Device Query (Runtime API) version (CUDA static linking)

Detected 1 CUDA Capable device(s)

Device 0: "NVIDIA T1000 8GB"
  CUDA Driver Version / Runtime Version      12.1 / 12.1
  CUDA Capability Major/Minor version number: 7.5
  Total amount of global memory:             2192 MBytes (8589606912 bytes)
  (14) Multiprocessors, ( 64) CUDA Cores/MP: 896 CUDA Cores
  GPU Max Clock rate:                       1395 Mhz (1.39 GHz)
  Memory Clock rate:                        5001 Mhz
  Memory Bus Width:                         128-bit
  L2 Cache Size:                            1048576 bytes
  Maximum Texture Dimension Size (x,y,z)    1D=(131072), 2D=(131072, 65536), 3D=(16384, 16384, 16384)
  Maximum Latched ID Texture Size, (num) layers 1D=(32768), 2048 layers
  Maximum Latched 2D Texture Size, (num) layers 2D=(32768, 32768), 2048 layers
  Total amount of constant memory:          65536 bytes
  Total amount of shared memory per block:  49152 bytes
  Total number of registers available per block: 65536
  Warp size:                                32
  Maximum number of threads per multiprocessor: 1024
  Maximum number of threads per block:      1024
  Max dimension size of a thread block (x,y,z): (1024, 1024, 64)
  Max dimension size of a grid size (x,y,z): (2147483647, 65535, 65535)
  Maximum memory pitch:                    2147483647 bytes
  Texture alignment:                        512 bytes

```

[그림 1.25] WSL2에서 CUDA 12.1 - 설치 확인하기(2)

```

djy@djy-workstation: ~
Maximum Latched 1D Texture Size, (num) layers 1D=(32768), 2048 layers
Maximum Latched 2D Texture Size, (num) layers 2D=(32768, 32768), 2048 layers
Total amount of constant memory:            65536 bytes
Total amount of shared memory per block:    49152 bytes
Total number of registers available per block: 65536
Warp size:                                  32
Maximum number of threads per multiprocessor: 1024
Maximum number of threads per block:        1024
Max dimension size of a thread block (x,y,z): (1024, 1024, 64)
Max dimension size of a grid size (x,y,z):  (2147483647, 65535, 65535)
Maximum memory pitch:                       2147483647 bytes
Texture alignment:                           512 bytes
Concurrent copy and kernel execution:        Yes with 2 copy engine(s)
Run time limit on kernels:                   Yes
Integrated GPU sharing Host Memory:          No
Support host page-locked memory mapping:     Yes
Alignment requirement for Surfaces:          Yes
Device has ECC support:                      Disabled
Device supports Unified Addressing (UVA):    Yes
Device supports Compute Preemption:          Yes
Supports Cooperative Kernel Launch:          Yes
Supports MultiDevice Co-op Kernel Launch:   No
Device PCI Domain ID / Bus ID / location ID: 0 / 7 / 0
Compute Mode:                                < Default (multiple host threads can use ::cudaSetDevice() with device simultaneously) >

deviceQuery, CUDA Driver = CUDART, CUDA Driver Version = 12.1, CUDA Runtime Version = 12.1, NumDevs = 1, Device0 = NVIDIA T1000 8GB
Result = PASS
djy@djy-workstation:~$

```

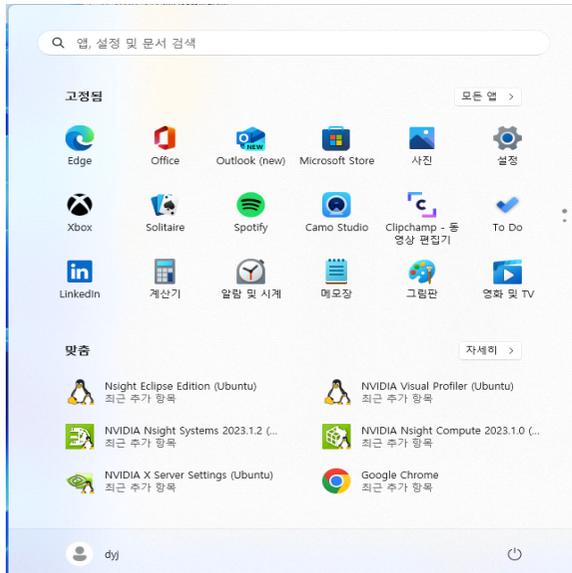
[그림 1.26] WSL2에서 CUDA 12.1 - 설치 확인하기(3)

실행 후, 마지막 줄에 다음처럼 PASS가 출력되면 잘 설치가 된 것이다.

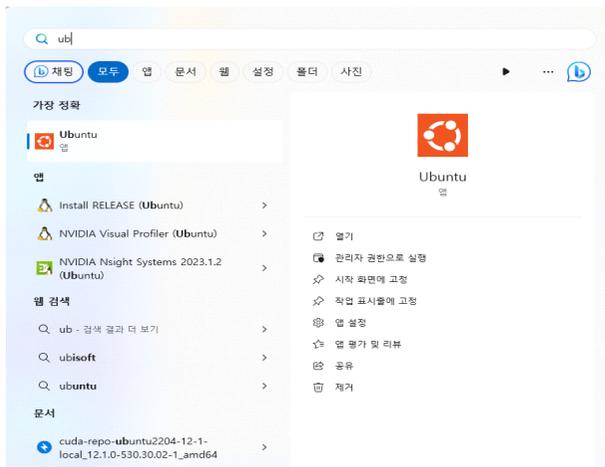
deviceQuery, CUDA Driver = CUDART, CUDA Driver Version = 12.1, CUDA Runtime Version = 12.1, NumDevs = 1, Device0 = NVIDIA T1000 8GB  
Result = PASS

참고로 MS윈도11에서도 그래픽카드 드라이버를 별도로 설치하고, CUDA 12.1을 따로 설치하는 것으로 생각할 수 있는데, 기존에 일반 그래픽카드 드라이버는 지우고, Cuda를 설치해야 한다. Cuda 설치 파일에도 그래픽카드 드라이버와 Nvidia HDMI 오디오 드라이버가 내장되어 있다.

◎ 참고사항 - MS윈도11에서 WSL2 - 우분투 실행하는 방법  
 시작 메뉴에서 “ub”라고 검색하면 Ubuntu가 검색된다.  
 클릭하면 실행된다.



[그림 1.27] 시작 메뉴 모습(1)



[그림 1.28] 시작 메뉴 모습(2)

## 참고 자료

1. 딥러닝에 목마른 사람들을 위한 PyTorch / 장지수 지음, BJ Public
2. “WSL2에 CUDA 설치하는 방법”, webnautes,  
<https://webnautes.tistory.com/1848>,  
accessed by 2024-01-26, last modified 2023-10-02.
3. “WSL2 ubuntu에 원격 데스크탑으로 접속하기..”, freetome,  
<https://freetome.tistory.com/517>,  
accessed by 2024-01-26, last modified 2022-02-03.
4. “WSL2 Ubuntu 한글화 설정”, 코딩하고분석하는돌스,  
<https://datanavigator.tistory.com/60>,  
accessed by 2024-01-26, last modified 2023-10-21
5. “wsl2 ubuntu 20.04 distro에서 snap 오류”, bitlog,  
<https://bitlog.tistory.com/135>,  
accessed by 2024-01-27, last modified 2020-07-20.
6. “[240127] WSL(Windows Subsystem for Linux) 종료 및 재부팅 방법 - JooTC”, JooTC,  
<https://jootc.com/p/202007093546>,  
accessed by 2024-01-27, last modified 2020-07-03.
7. “WSL - 2. Windows에서 WSL2 ubuntu 데스크톱 모드로 사용”, 동이,  
<https://guiyomi.tistory.com/113>,  
accessed by 2024-01-27, last modified 2022-03-24.

## 2장 OpenSSH Server 서버 구축과 서비스 운영

- 2.1 OpenSSH Server 우분투 설치
- 2.2 OpenSSH Server 서비스 재시작

## 2.1 OpenSSH Server 우분투 설치

### ◎ OpenSSH 개요

OpenSSH(Open Secure Shell)는 네트워크 통신을 암호화하여 보안을 강화한 프로토콜인 SSH(Secure Shell)를 구현한 소프트웨어이다. SSH는 원격으로 컴퓨터를 제어할 때 주로 사용되며, OpenSSH는 이를 오픈 소스로 구현한 것이다. OpenSSH Server는 이 중 SSH 서버 기능을 제공하는 컴포넌트를 말한다. OpenSSH Server가 설치된 컴퓨터는 SSH 클라이언트를 통한 원격 접속을 받아들일 수 있다. 즉, OpenSSH Server가 실행되고 있는 컴퓨터는 원격에서도 안전하게 컴퓨터를 제어하거나, 파일을 전송할 수 있다.

OpenSSH Server는 다양한 운영체제에서 사용될 수 있으며, 특히 리눅스나 유닉스 계열의 서버에서 많이 사용된다.

실질적으로 국내 웹호스팅의 경우, 현재는 FTP는 거의 사용되지 않으며, SSH를 이용한 SFTP를 주로 사용한다.

SFTP의 단점으로는 jail(이하 “교도소” 또는 “구류”)에 대한 대책을 강구해야 한다. chroot directory 셋팅부터 사용자 권한, 디렉터리 권한 등 해야 할 일이 있다. 해당 글에서 OpenSSH Server에 대해서 깊게 다루지 않도록 하겠다.

보안 문제가 가장 크다고 할 수 있으며, 서버를 운영하는 경우에는 SSH를 활용하여 VPN이나 VNC를 활용한 터널링을 개방할 수 있다.

우분투 22.04(이하 “Ubuntu 22.04”)에서 간단한 명령어를 통해 설치할 수 있다.

```
sudo apt-get install openssh-server
```

```

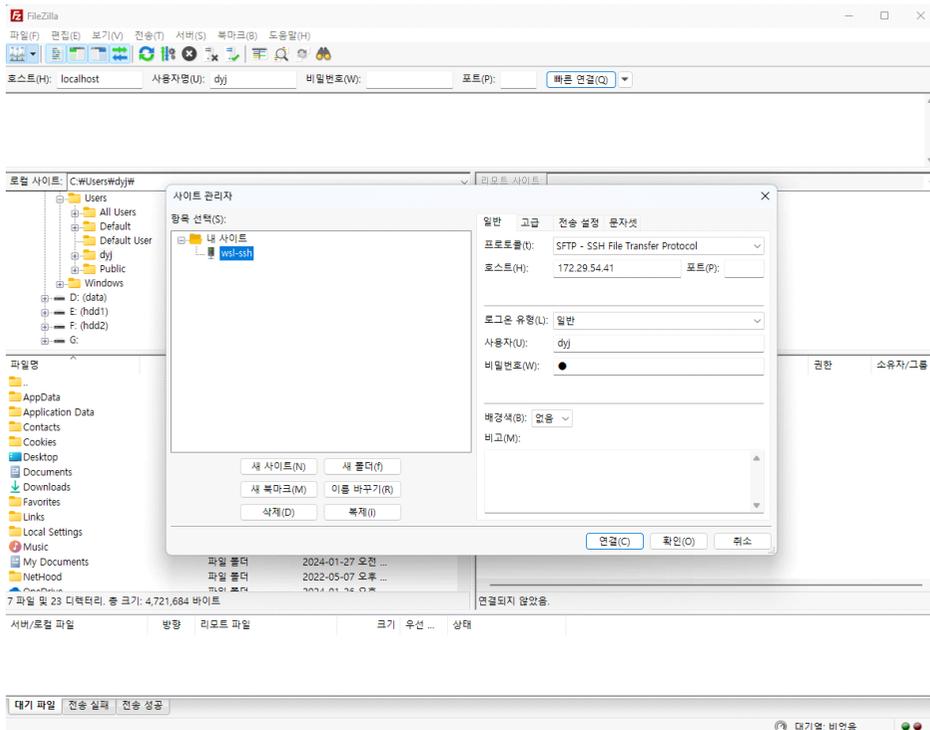
djy@djy-workstation:~$ sudo apt-get install openssh-server
[sudo] djy 암호:
패키지 목록을 읽는 중입니다... 완료
의존성 트리를 만드는 중입니다... 완료
상태 정보를 읽는 중입니다... 완료
다음의 추가 패키지가 설치될 것입니다:
ncurses-term openssh-sftp-server ssh-import-id
제안하는 패키지:
molly-guard monkeysphere ssh-askpass
다음 새 패키지를 설치할 것입니다:
ncurses-term openssh-server openssh-sftp-server ssh-import-id
0개 업그레이드, 4개 새로 설치, 0개 제거 및 0개 업그레이드 안 함.
752 k바이트 아카이브를 받아야 합니다.
이 작업 후 6,050 k바이트의 디스크 공간을 더 사용하게 됩니다.
계속 하시겠습니까? [Y/n] Y
받기:1 http://mirror.kakao.com/ubuntu jammy-updates/main amd64 openssh-sftp-server amd64 1:8.9p1-3ubuntu0.6 [38.7 kB]
받기:2 http://mirror.kakao.com/ubuntu jammy-updates/main amd64 openssh-server amd64 1:8.9p1-3ubuntu0.6 [435 kB]
받기:3 http://mirror.kakao.com/ubuntu jammy-updates/main amd64 ncurses-term all 6.3-2ubuntu0.1 [267 kB]
받기:4 http://mirror.kakao.com/ubuntu jammy/main amd64 ssh-import-id all 5.11-0ubuntu1 [10.1 kB]
내려받기 752 k바이트, 소요시간 1초 (598 k바이트/초)
패키지를 미리 설정하는 중입니다.

```

[그림 2.1] OpenSSH 설치 모습

◎ 파일질라 클라이언트를 통한 SFTP 접속하기(1)

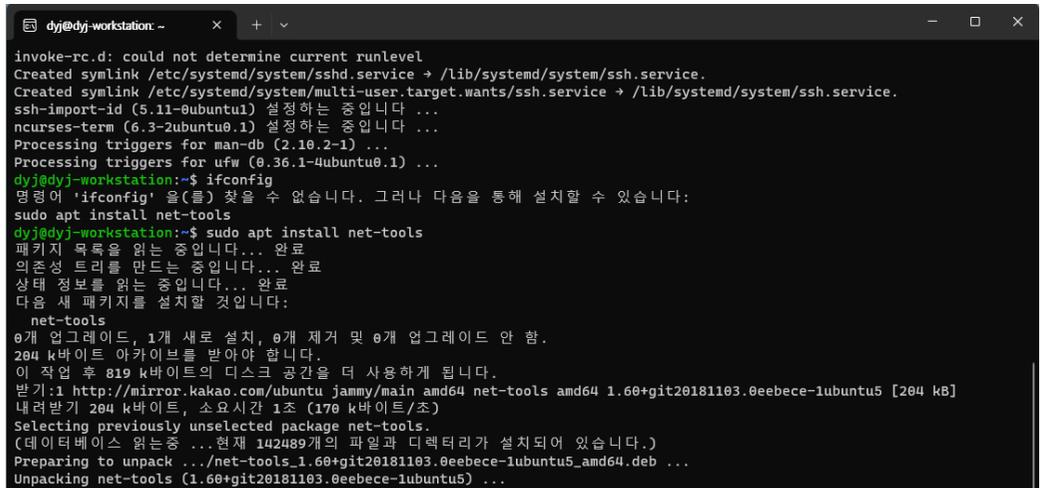
파일질라 클라이언트를 통해 WSL2의 서버에 접근할 수 있다.



[그림 2.2] 파일질라 클라이언트에서 SFTP 접속을 위한 설정 모습

▷ net-tools 설치 후 ifconfig로 아이피 주소 확인하기

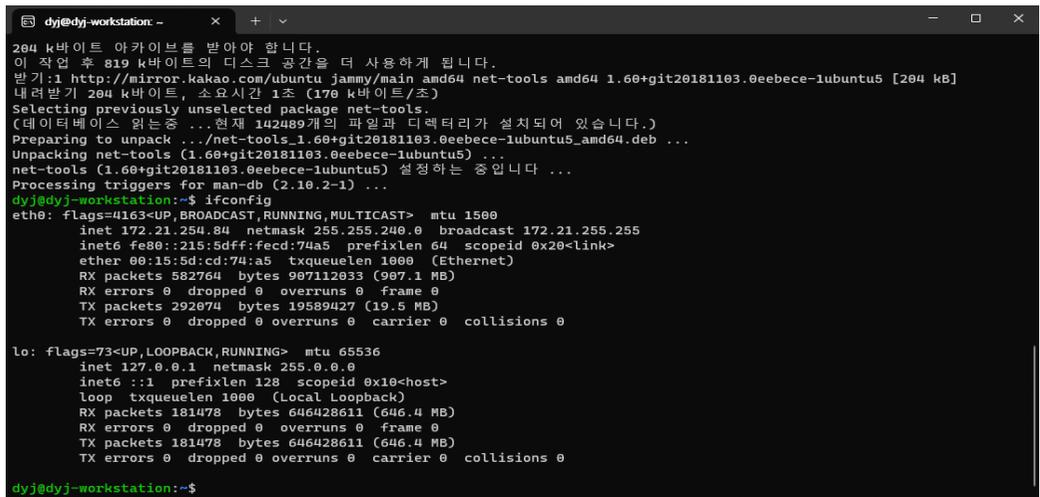
sudo apt-get install net-tools 라는 명령어로 net-tools 유틸리티를 우분투에 설치할 수 있다.



```
invoke-rc.d: could not determine current runlevel
Created symlink /etc/systemd/system/ssh.service → /lib/systemd/system/ssh.service.
Created symlink /etc/systemd/system/multi-user.target.wants/ssh.service → /lib/systemd/system/ssh.service.
ssh-import-id (5.11-0ubuntu1) 설정하는 중입니다 ...
ncurses-term (6.3-2ubuntu0.1) 설정하는 중입니다 ...
Processing triggers for man-db (2.10.2-1) ...
Processing triggers for ufw (0.36.1-4ubuntu0.1) ...
dyj@dyj-workstation:~$ ifconfig
명령어 'ifconfig' 을(를) 찾을 수 없습니다. 그러나 다음을 통해 설치할 수 있습니다:
sudo apt install net-tools
dyj@dyj-workstation:~$ sudo apt install net-tools
패키지 목록을 읽는 중입니다... 완료
의존성 트리를 만드는 중입니다... 완료
상태 정보를 읽는 중입니다... 완료
다음 새 패키지를 설치할 것입니다:
 net-tools
0개 업그레이드, 1개 새로 설치, 0개 제거 및 0개 업그레이드 안 함.
204 k바이트 아카이브를 받아야 합니다.
이 작업 후 819 k바이트의 디스크 공간을 더 사용하게 됩니다.
받기:1 http://mirror.kakao.com/ubuntu jammy/main amd64 net-tools amd64 1.60+git20181103.0eebece-1ubuntu5 [204 kB]
내려받기 204 k바이트, 소요시간 1초 (170 k바이트/초)
Selecting previously unselected package net-tools.
(데이터베이스 읽는중 ... 현재 142489개의 파일과 디렉터리가 설치되어 있습니다.)
Preparing to unpack .../net-tools_1.60+git20181103.0eebece-1ubuntu5_amd64.deb ...
Unpacking net-tools (1.60+git20181103.0eebece-1ubuntu5) ...
```

[그림 2.3] net-tools 설치 후 ifconfig로 아이피 주소 확인하기(1)

net-tools를 설치 후 ifconfig 명령어로 ipv4, ipv6 아이피 주소를 확인할 수 있다.



```
204 k바이트 아카이브를 받아야 합니다.
이 작업 후 819 k바이트의 디스크 공간을 더 사용하게 됩니다.
받기:1 http://mirror.kakao.com/ubuntu jammy/main amd64 net-tools amd64 1.60+git20181103.0eebece-1ubuntu5 [204 kB]
내려받기 204 k바이트, 소요시간 1초 (170 k바이트/초)
Selecting previously unselected package net-tools.
(데이터베이스 읽는중 ... 현재 142489개의 파일과 디렉터리가 설치되어 있습니다.)
Preparing to unpack .../net-tools_1.60+git20181103.0eebece-1ubuntu5_amd64.deb ...
Unpacking net-tools (1.60+git20181103.0eebece-1ubuntu5) ...
net-tools (1.60+git20181103.0eebece-1ubuntu5) 설정하는 중입니다 ...
Processing triggers for man-db (2.10.2-1) ...
dyj@dyj-workstation:~$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 172.21.254.84 netmask 255.255.240.0 broadcast 172.21.255.255
    inet6 fe80::215:5dff:fe0d:74a5 prefixlen 64 scopeid 0x20<link>
    ether 08:15:5d:cd:74:a5 txqueuelen 1000 (Ethernet)
    RX packets 582764 bytes 907112033 (907.1 MB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 292074 bytes 19589427 (19.5 MB)
    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 181478 bytes 646428611 (646.4 MB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 181478 bytes 646428611 (646.4 MB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

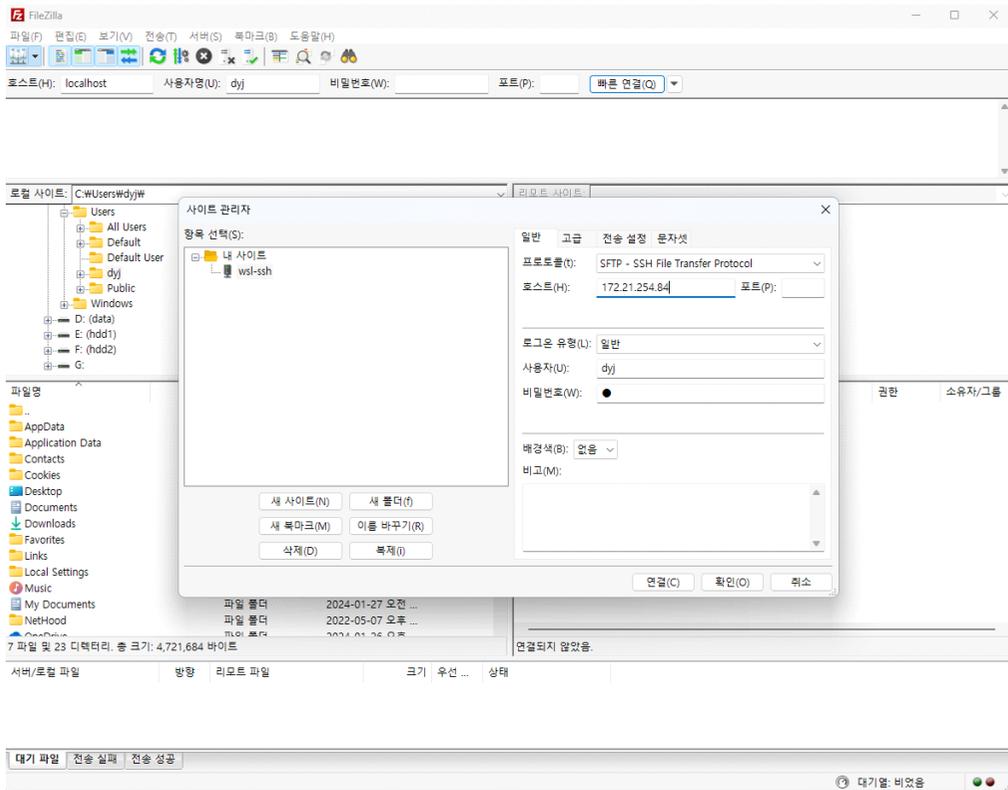
dyj@dyj-workstation:~$
```

[그림 2.4] net-tools 설치 후 ifconfig로 아이피 주소 확인하기(2)

## ◎ 파일질라 클라이언트를 통한 SFTP 접속하기(2)

앞서 확인한 ipv4 주소를 입력한다. OpenSSH는 기본 포트는 22번이므로 별도로 입력하지 않을 경우에는 22번이 자동으로 인식된다.

사용자 계정과 비밀번호를 입력한다. 그리고 WSL2으로 이동한다.



[그림 2.5] 파일질라 클라이언트에서 SFTP 접속하기(1)

## 2.2 OpenSSH Server 서비스 재시작

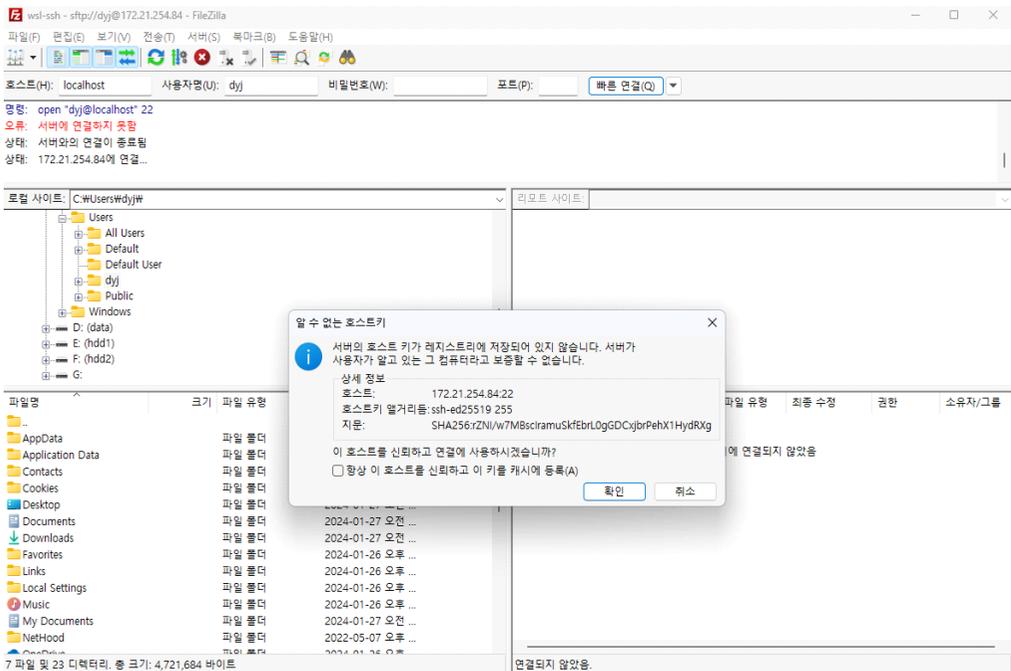
다음 명령어를 통해 OpenSSH Server를 재시작할 수 있다.

```
sudo service ssh restart
```



[그림 2.6] OpenSSH Server 재시작하기(1)

그리고 파일질라 클라이언트로 돌아가서 “연결(C)”를 클릭한다.



[그림 2.7] OpenSSH Server 재시작하기(2)

인증키에 대한 서명을 확인하면 SFTP를 사용할 수 있다.

## 참고 자료

1. “SSH 서버 Ubuntu 22.04 설치 - 시스템 관리”, linux-console.net,  
<https://ko.linux-console.net/?p=10414>,  
accessed by 2024-01-27, last modified 2023.

## 3장 xfce4 데스크톱 환경과 xrdp 구축

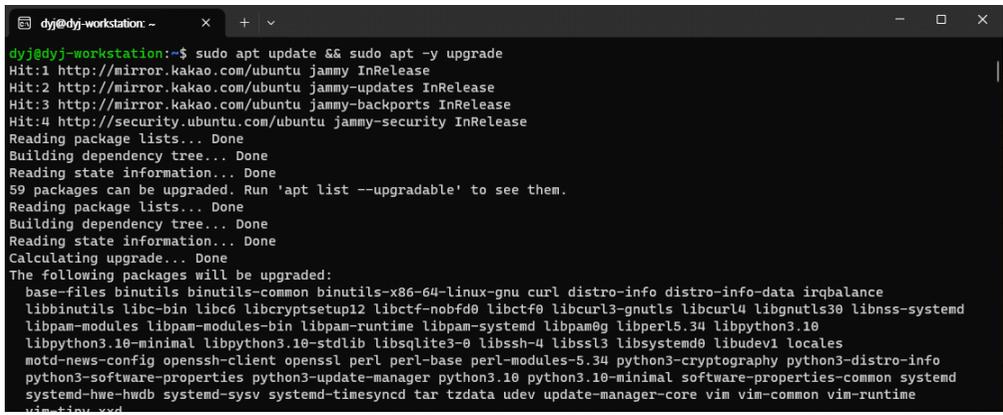
- 3.1 우분투 데스크톱 환경 설치
  - 3.2 xfce4 설치
  - 3.3 한글 폰트, 언어 설정
  - 3.4 dbus 서비스 시작
  - 3.5 한글 언어팩 설치
  - 3.6 ibus 한글 키보드 설정
  - 3.7 xrdp 부팅 서비스 활성화

### 3.1 우분투 데스크톱 환경 설치

다음과 같은 명령어로 apt 서버 업데이트 및 업그레이드를 진행할 수 있다.

apt update와 apt upgrade를 먼저 수행하는 이유는 ubuntu-desktop의 최신 버전이 설치될 수 있도록 하기 위함이다.

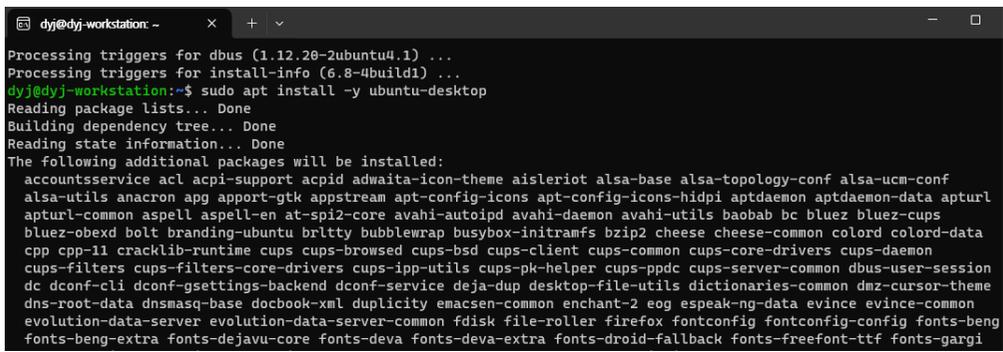
```
sudo apt update && sudo apt -y upgrade
```



[그림 3.1] apt update, apt upgrade 실행 모습

다음 명령어를 통해 우분투 데스크톱 환경(이하 “GNOME”)을 설치할 수 있다.

```
sudo apt install -y ubuntu-desktop
```



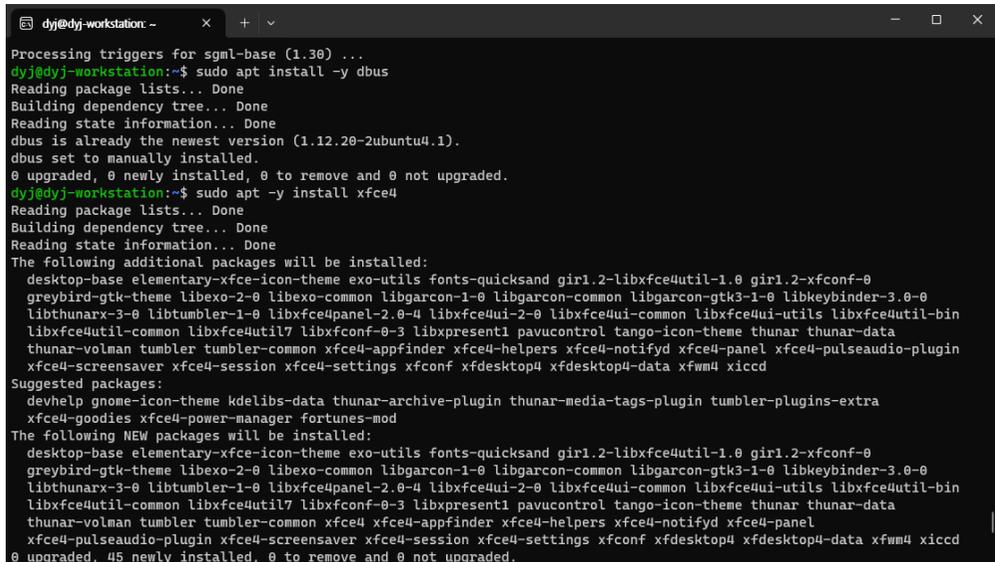
[그림 3.2] Ubuntu Desktop 환경

## 3.2 xfce4 설치

다음과 같은 명령어로 xfce4를 설치할 수 있다.

타이핑을 따라 할 수 있도록 명령어를 정리하였다.

```
sudo apt -y install xfce4
sudo apt-get install xrdp
sudo cp /etc/xrdp/xrdp.ini /etc/xrdp/xrdp.ini.bak
sudo sed -i 's/3389/3390/g' /etc/xrdp/xrdp.ini
sudo sed -i 's/max_bpp=32/#max_bpp=32nmax_bpp=128/g' /etc/xrdp/xrdp.ini
sudo sed -i 's/xserverbpp=24/#xserverbpp=24nxserverbpp=128/g'
/etc/xrdp/xrdp.ini
```



```
dj@dj-workstation: ~$ sudo apt install -y dbus
Processing triggers for sgml-base (1.30) ...
dj@dj-workstation:~$ sudo apt install -y dbus
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
dbus is already the newest version (1.12.20-2ubuntu4.1).
dbus set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
dj@dj-workstation:~$ sudo apt -y install xfce4
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
 desktop-base elementary-xfce-icon-theme exo-utils fonts-quicksand gir1.2-libxfce4util-1.0 gir1.2-xfconf-0
 greybird-gtk-theme libexo-2-0 libexo-common libgarcon-1-0 libgarcon-common libgarcon-gtk3-1-0 libkeybinder-3.0-0
 libthunarx-3-0 libtumbler-1-0 libxfce4panel-2.0-4 libxfce4ui-2-0 libxfce4ui-common libxfce4ui-utils libxfce4util-bin
 libxfce4util-common libxfce4util7 libxfconf-0-3 libxpresent1 pavucontrol tango-icon-theme thunar thunar-data
 thunar-volman tumbler tumbler-common xfce4-appfinder xfce4-helpers xfce4-notifyd xfce4-panel xfce4-pulseaudio-plugin
 xfce4-screensaver xfce4-session xfce4-settings xfconf xfdesktop4 xfdesktop4-data xfm4 xiccd
Suggested packages:
 devhelp gnome-icon-theme kdelibs-data thunar-archive-plugin thunar-media-tags-plugin tumbler-plugins-extra
 xfce4-goodies xfce4-power-manager fortunes-mod
The following NEW packages will be installed:
 desktop-base elementary-xfce-icon-theme exo-utils fonts-quicksand gir1.2-libxfce4util-1.0 gir1.2-xfconf-0
 greybird-gtk-theme libexo-2-0 libexo-common libgarcon-1-0 libgarcon-common libgarcon-gtk3-1-0 libkeybinder-3.0-0
 libthunarx-3-0 libtumbler-1-0 libxfce4panel-2.0-4 libxfce4ui-2-0 libxfce4ui-common libxfce4ui-utils libxfce4util-bin
 libxfce4util-common libxfce4util7 libxfconf-0-3 libxpresent1 pavucontrol tango-icon-theme thunar thunar-data
 thunar-volman tumbler tumbler-common xfce4-appfinder xfce4-helpers xfce4-notifyd xfce4-panel
 xfce4-pulseaudio-plugin xfce4-screensaver xfce4-session xfce4-settings xfconf xfdesktop4 xfdesktop4-data xfm4 xiccd
0 upgraded, 45 newly installed, 0 to remove and 0 not upgraded.
```

[그림 3.3] xfce4 설치 모습(1)

```

Processing triggers for man-db (2.10.2-1) ...
dyj@dyj-workstation:~$ sudo apt-get install xrdp
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  libfuse2 xorgxrdp
Suggested packages:
  guacamole
The following NEW packages will be installed:
  libfuse2 xorgxrdp xrdp
0 upgraded, 3 newly installed, 0 to remove and 0 not upgraded.
Need to get 672 kB of archives.
After this operation, 3710 kB of additional disk space will be used.
Do you want to continue? [Y/n] Y
Get:1 http://mirror.kakao.com/ubuntu jammy/universe amd64 libfuse2 amd64 2.9.9-5ubuntu3 [90.3 kB]
Get:2 http://mirror.kakao.com/ubuntu jammy/universe amd64 xrdp amd64 0.9.17-2ubuntu2 [517 kB]
Get:3 http://mirror.kakao.com/ubuntu jammy/universe amd64 xorgxrdp amd64 1:0.2.17-1build1 [65.3 kB]
Fetched 672 kB in 1s (539 kB/s)
Selecting previously unselected package libfuse2:amd64.
(Reading database ... 138178 files and directories currently installed.)
Preparing to unpack .../libfuse2_2.9.9-5ubuntu3_amd64.deb ...
Unpacking libfuse2:amd64 (2.9.9-5ubuntu3) ...
Selecting previously unselected package xrdp.
Preparing to unpack .../xrdp_0.9.17-2ubuntu2_amd64.deb ...
Unpacking xrdp (0.9.17-2ubuntu2) ...
Selecting previously unselected package xorgxrdp.
Preparing to unpack .../xorgxrdp_1%3a0.2.17-1build1_amd64.deb ...
Unpacking xorgxrdp (1:0.2.17-1build1) ...
Setting up libfuse2:amd64 (2.9.9-5ubuntu3) ...

```

[그림 3.4] xfce4 설치 모습(2)

```

/sbin/ldconfig.real: Can't link /usr/lib/wsl/lib/libnvoptix_loader.so.1 to libnvoptix.so.1
/sbin/ldconfig.real: /usr/lib/wsl/lib/libcuda.so.1 is not a symbolic link

dyj@dyj-workstation:~$ sudo cp /etc/xrdp/xrdp.ini /etc/xrdp/xrdp.ini.bak
dyj@dyj-workstation:~$ sudo sed -i 's/3389/3390/g' /etc/xrdp/xrdp.ini
dyj@dyj-workstation:~$ sudo sed -i 's/max_bpp=32/#max_bpp=32max_bpp=128/g' /etc/xrdp/xrdp.ini
dyj@dyj-workstation:~$ sudo sed -i 's/xserverbpp=24/#xserverbpp=24nxserverbpp=128/g' /etc/xrdp/xrdp.ini
dyj@dyj-workstation:~$ sudo /etc/init.d/xrdp start
* Starting Remote Desktop Protocol server
dyj@dyj-workstation:~$ echo xfce4-session > ~/.xsession
dyj@dyj-workstation:~$ sudo apt-get install fonts-nanum*
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Note, selecting 'fonts-nanum-extra' for glob 'fonts-nanum*'
Note, selecting 'fonts-nanum-coding' for glob 'fonts-nanum*'
Note, selecting 'fonts-nanum-eco' for glob 'fonts-nanum*'
Note, selecting 'fonts-nanum' for glob 'fonts-nanum*'
The following NEW packages will be installed:
  fonts-nanum fonts-nanum-coding fonts-nanum-eco fonts-nanum-extra
0 upgraded, 4 newly installed, 0 to remove and 0 not upgraded.
Need to get 46.0 MB of archives.
After this operation, 177 MB of additional disk space will be used.
Get:1 http://mirror.kakao.com/ubuntu jammy/universe amd64 fonts-nanum all 20200506-1 [10.3 MB]
Get:2 http://mirror.kakao.com/ubuntu jammy/universe amd64 fonts-nanum-coding all 2.5-3 [4988 B]
Get:3 http://mirror.kakao.com/ubuntu jammy/universe amd64 fonts-nanum-eco all 1.000-7 [14.7 MB]
Get:4 http://mirror.kakao.com/ubuntu jammy/universe amd64 fonts-nanum-extra all 20200506-1 [21.0 MB]
Fetched 46.0 MB in 5s (8954 kB/s)
Selecting previously unselected package fonts-nanum.
(Reading database ... 138318 files and directories currently installed.)

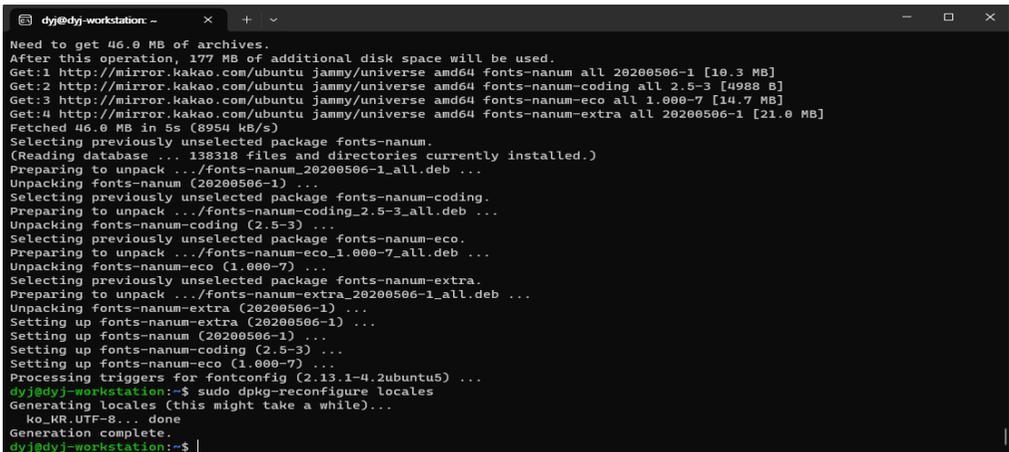
```

[그림 3.5] xfce4 설치 모습(2)

### 3.3 한글 폰트, 언어 설정

다음과 같은 명령어로 한글 폰트(이하 “나눔 폰트”)를 설치할 수 있다.

```
sudo apt-get install fonts-nanum*
```

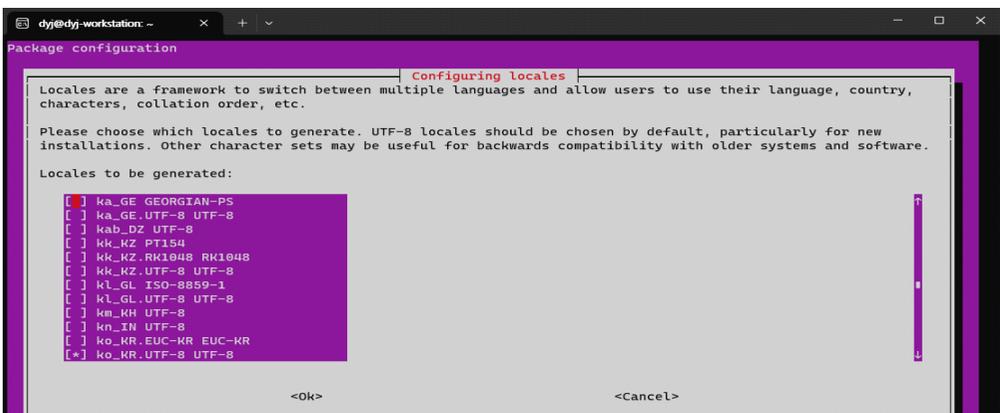


[그림 3.6] 나눔 고딕 폰트 설치 모습

다음과 같은 명령어로 로케일을 설정할 수 있다.

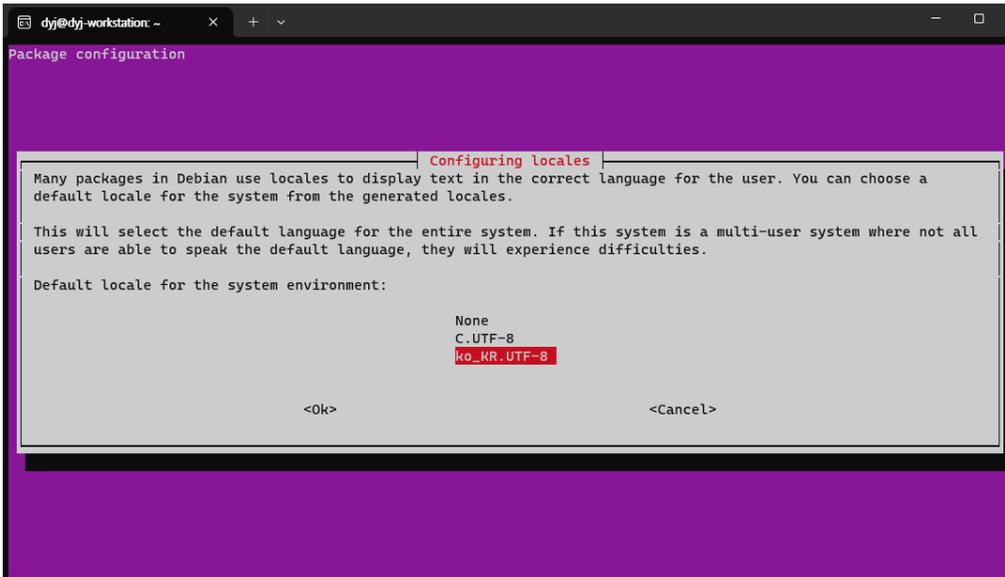
```
sudo dpkg-reconfigure locales
```

아래처럼 ko\_KR.UTF-8 UTF-8을 선택하면 된다. 그리고 OK를 선택한다.



[그림 3.7] locale 설정하기(1)

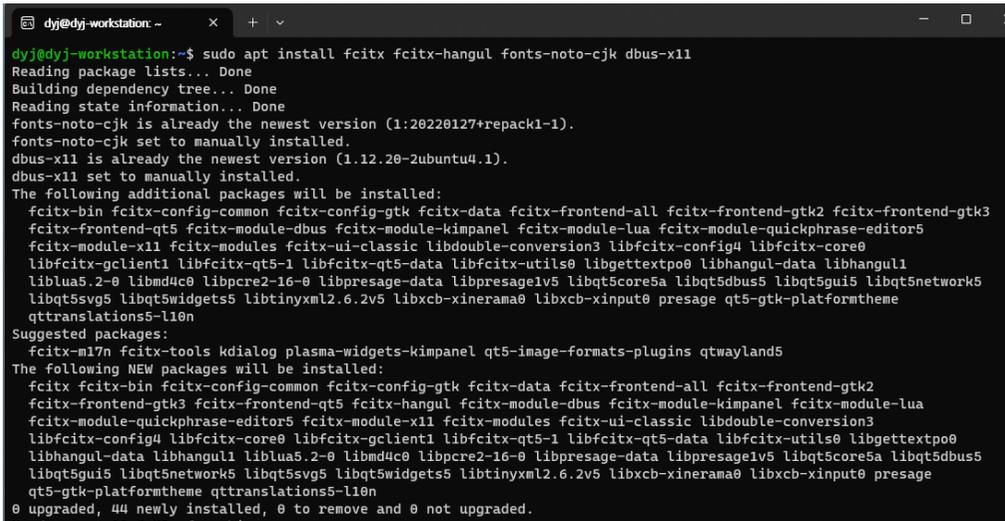
ko\_KR.UTF-8을 선택한 후 OK를 선택한다.



[그림 3.8] locale 설정하기(2)

fcitx 키보드, fcitx 한글키보드, noto-cjk 폰트, dbus-x11을 설치할 수 있다.

```
sudo apt install fcitx fcitx-hangul fonts-noto-cjk dbus-x11
```



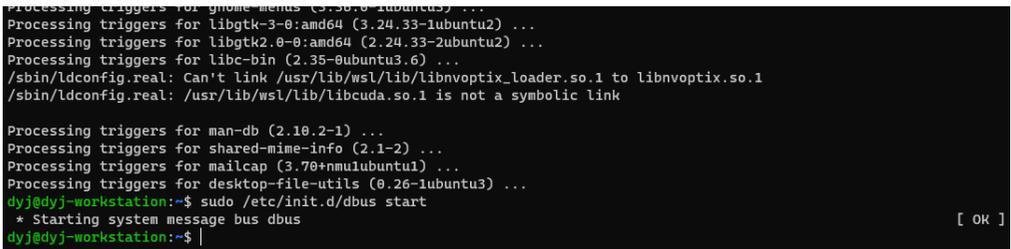
[그림 3.9] fcitx 키보드 설치, 폰트 설치, dbus-x11 설치

### 3.4 dbus 서비스 시작

다음과 같이 dbus 서비스를 시작할 수 있다.

dbus 서비스를 시작하지 않으면 GUI의 환경설정 프로그램을 사용할 수 없다.

```
sudo /etc/init.d/dbus start
```



```
Processing triggers for gnome-menus (3.36.0-1ubuntu1) ...
Processing triggers for libgtk-3-0:amd64 (3.24.33-1ubuntu2) ...
Processing triggers for libgtk2.0-0:amd64 (2.24.33-2ubuntu2) ...
Processing triggers for libc-bin (2.35-0ubuntu3.6) ...
/sbin/ldconfig.real: Can't link /usr/lib/wsl/lib/libnvoptix_loader.so.1 to libnvoptix.so.1
/sbin/ldconfig.real: /usr/lib/wsl/lib/libcuda.so.1 is not a symbolic link

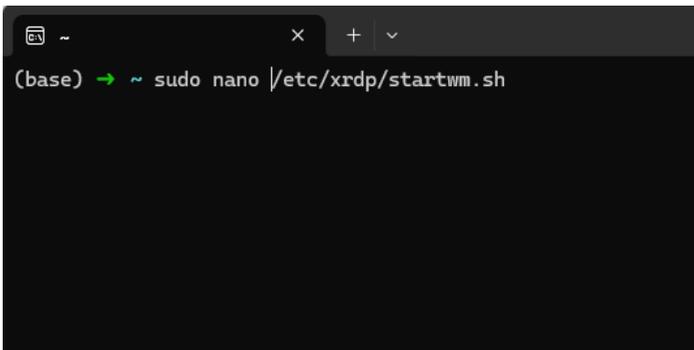
Processing triggers for man-db (2.10.2-1) ...
Processing triggers for shared-mime-info (2.1-2) ...
Processing triggers for mailcap (3.70+nmulubuntu1) ...
Processing triggers for desktop-file-utils (0.26-1ubuntu3) ...
dyj@dyj-workstation:~$ sudo /etc/init.d/dbus start
 * Starting system message bus dbu
dyj@dyj-workstation:~$
```

[그림 3.10] dbus-x11 서비스 시작

### 3.5 xrdp - startwm.sh 환경설정

xrdp의 xfce4에서 한글 키보드 인식, 튕김 현상 제거 등을 수행하려면 다음 작업을 진행해줘야 한다.

```
sudo nano /etc/xrdp/startwm.sh
```



```
(base) → ~ sudo nano /etc/xrdp/startwm.sh
```

[그림 3.11] startwm.sh 관리자 권한으로 열기

```

GNU nano 6.2 /etc/xrdp/startwm.sh
#!/bin/sh
# xrdp X session start script (c) 2015, 2017, 2021 mirabilos
# published under The MirOS Licence

# Rely on /etc/pam.d/xrdp-sesman using pam_env to load both
# /etc/environment and /etc/default/locale to initialise the
# locale and the user environment properly.

if test -r /etc/profile; then
    . /etc/profile
fi

unset DBUS_SESSION_BUS_ADDRESS
unset XDG_RUNTIME_DIR

test -x /etc/X11/Xsession && exec /etc/X11/Xsession
exec /bin/sh /etc/X11/Xsession

```

[그림 3.12] startwm.sh 수정하기

fi 아래와 test -x 중간 사이에 다음과 같이 입력한다.

```

unset DBUS_SESSION_BUS_ADDRESS
unset XDG_RUNTIME_DIR

```

Ctrl + X를 눌러서 저장을 하고 나간다.

다음의 명령어로 서비스를 재시작한다.

```

sudo service xrdp restart

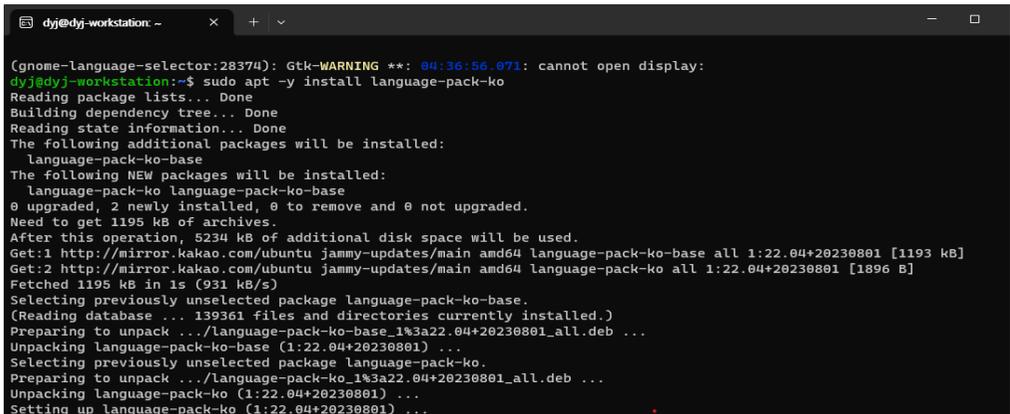
```

### 3.6 한글 언어팩 설치

한글 언어팩을 사용하려면, 언어팩을 설치해줘야 한다.

다음 명령어를 통해 설치할 수 있다.

```
sudo apt -y install language-pack-ko
```



```
dyj@dyj-workstation: ~  
(gnome-language-selector:28374): Gtk-WARNING **: 04:36:56.071: cannot open display:  
dyj@dyj-workstation:~$ sudo apt -y install language-pack-ko  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
The following additional packages will be installed:  
  language-pack-ko-base  
The following NEW packages will be installed:  
  language-pack-ko language-pack-ko-base  
0 upgraded, 2 newly installed, 0 to remove and 0 not upgraded.  
Need to get 1195 kB of archives.  
After this operation, 5234 kB of additional disk space will be used.  
Get:1 http://mirror.kakao.com/ubuntu jammy-updates/main amd64 language-pack-ko-base all 1:22.04+20230801 [1193 kB]  
Get:2 http://mirror.kakao.com/ubuntu jammy-updates/main amd64 language-pack-ko all 1:22.04+20230801 [1896 B]  
Fetched 1195 kB in 1s (931 kB/s)  
Selecting previously unselected package language-pack-ko-base.  
(Reading database ... 139361 files and directories currently installed.)  
Preparing to unpack .../language-pack-ko-base_1%3a22.04+20230801_all.deb ...  
Unpacking language-pack-ko-base (1:22.04+20230801) ...  
Selecting previously unselected package language-pack-ko.  
Preparing to unpack .../language-pack-ko_1%3a22.04+20230801_all.deb ...  
Unpacking language-pack-ko (1:22.04+20230801) ...  
Setting up language-pack-ko (1:22.04+20230801) ...
```

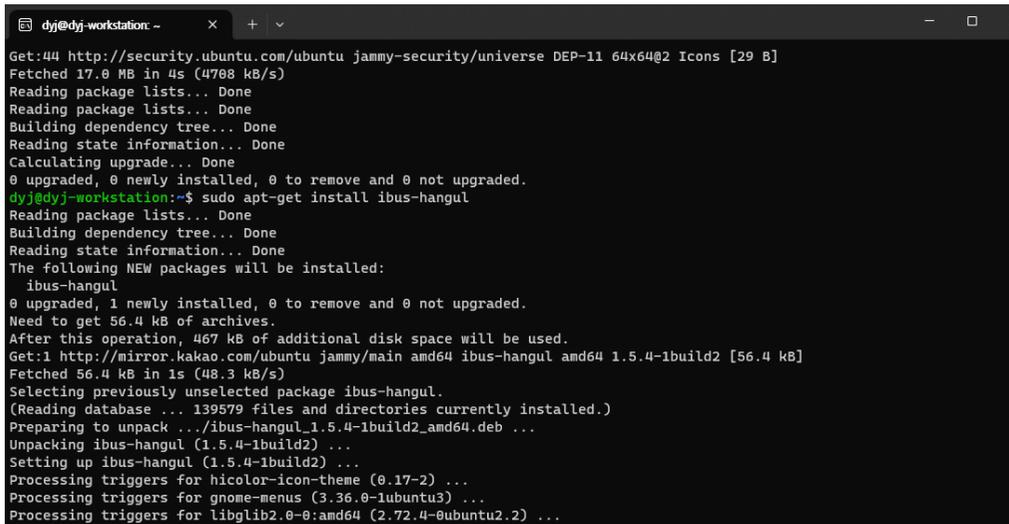
[그림 3.13] language-pack 한글 설치하기

### 3.7 ibus 한글 키보드 설정

우분투 데스크톱 환경에는 기본적으로 ibus 키보드가 설치되어 있다.

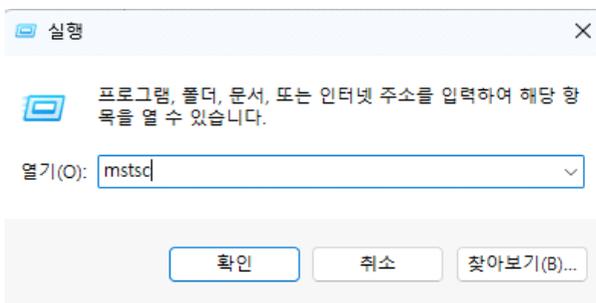
ibus-hangul이 설치되어 있지 않다면, 다음 명령어를 통해 설치할 수 있다.

```
sudo apt-get install ibus-hangul
```



[그림 3.14] ibus-hangul 설치하기

MS윈도에서 ‘윈도+R’키를 누른다. ‘mstsc’라고 입력한 후 확인을 누른다.



[그림 3.15] 실행에 mstsc 명령어 입력하기



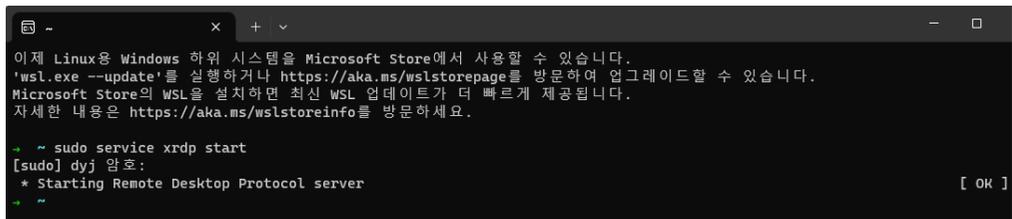
[그림 3.16] 원격 데스크톱 연결

컴퓨터(C)에 'localhost:3390'을 입력한다.

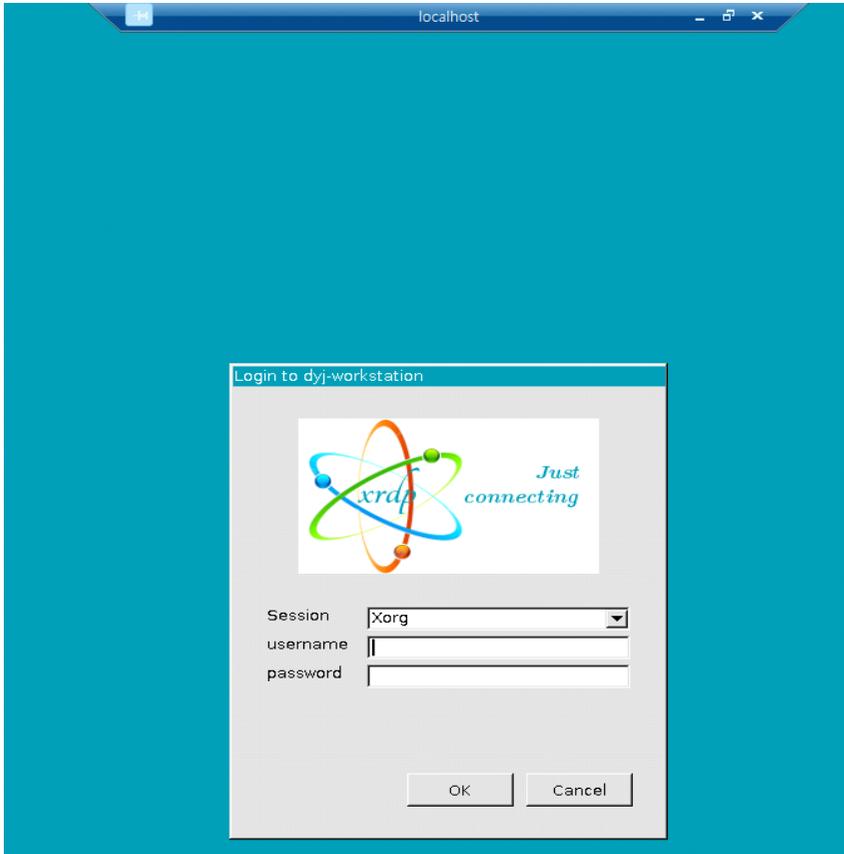
'연결(N)'을 클릭한다. 사용자 계정과 비밀번호를 입력한 후 확인을 누른다.

원격 데스크톱 연결이 안 되는 경우라면, 'Ubuntu' 앱을 먼저 부팅 후 실행해서 `sudo service xrdp start` 명령어를 입력해줘야 한다.

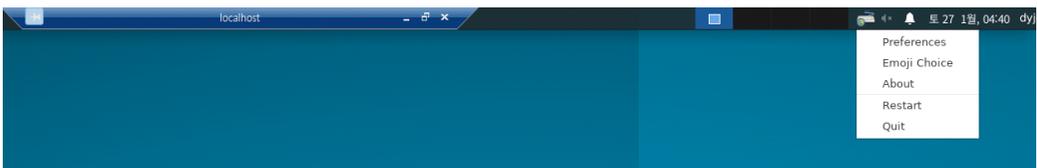
```
sudo service xrdp start
```



[그림 3.17] WSL2에서 XRDP 서비스 실행하기

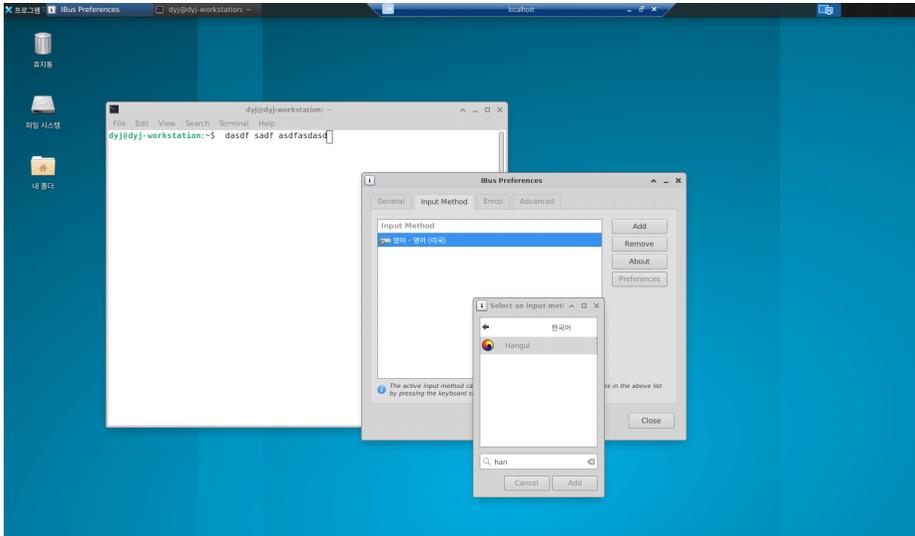


[그림 3.18] 원격 데스크톱 연결에서 사용자 계정 입력 모습  
 아래 그림은 상단에 키보드 설정을 마우스 오른쪽 버튼으로 클릭한 모습이  
 다. 'Preferences'를 클릭한다.

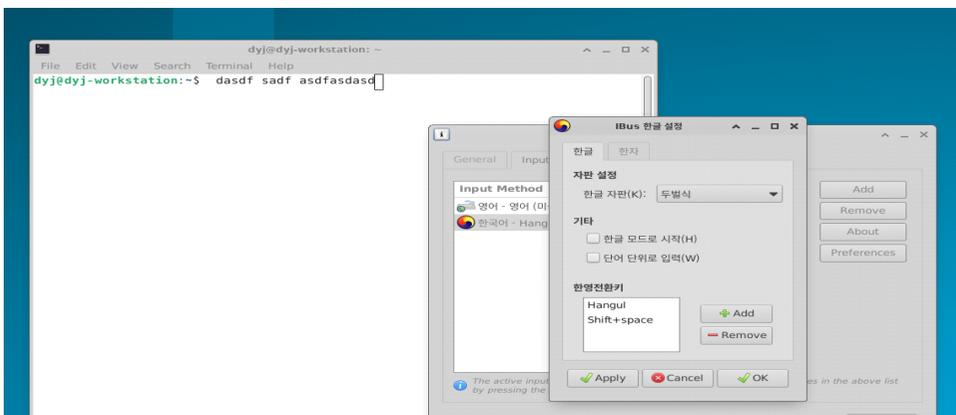


[그림 3.19] 상단 키보드 설정 - 오른쪽 메뉴 모습

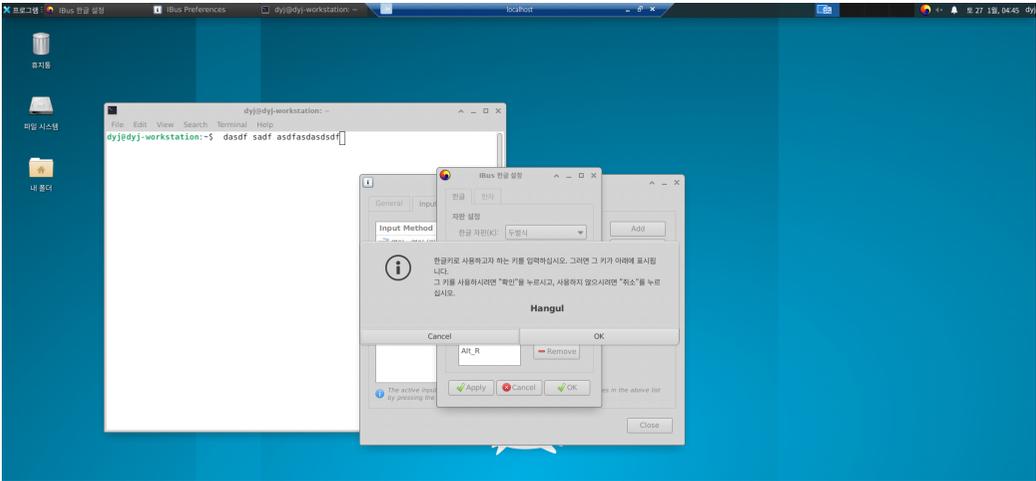
Input Method 탭에서 Add를 클릭한다.  
 한국어말고, h를 입력해서 Hangul을 찾는다.  
 Hangul을 선택한다. Add를 누른다.



[그림 3.20] 상단 키보드 설정 - 오른쪽 메뉴 모습  
 ‘한국어 - Hangul’을 선택한 후 ‘Preferences’를 클릭한다.  
 그리고 ‘Add’를 누른다. 다음에 한글키를 누른다.  
 ‘OK’를 누른다.

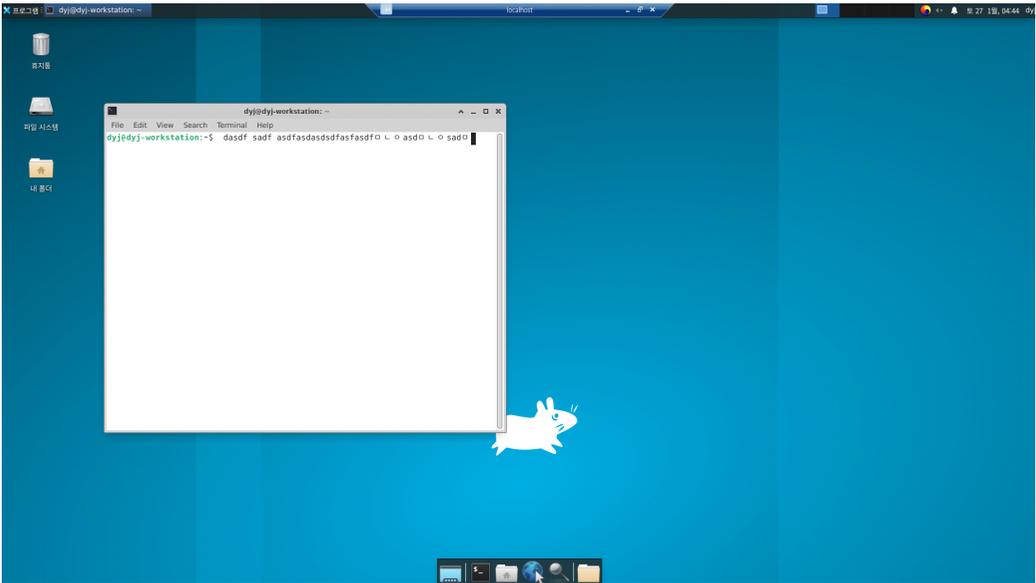


[그림 3.21] ibus 설정 모습(1)



[그림 3.22] ibus 설정 모습(2)

몇 번 반복하면, 한영 키가 동작하는 ‘한글 키보드’ 사용이 가능하게 된다. MS윈도 운영체제와 리눅스 운영체제는 이러한 점에서 차이점이 될 수 있다.

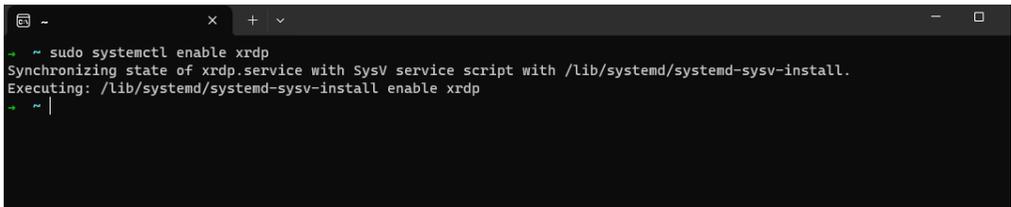


[그림 3.23] ibus 설정 모습(3)

### 3.8 xrdp 부팅 서비스 활성화

우분투에서 부팅 시 xrdp 서비스를 활성화하도록 하는 방법이다.

```
sudo systemctl enable xrdp
```



```
~ -  
x + v  
- □  
-> ~ sudo systemctl enable xrdp  
Synchronizing state of xrdp.service with SysV service script with /lib/systemd/systemd-sysv-install.  
Executing: /lib/systemd/systemd-sysv-install enable xrdp  
-> ~ |
```

[그림 3.24] xrdp 서비스 활성화

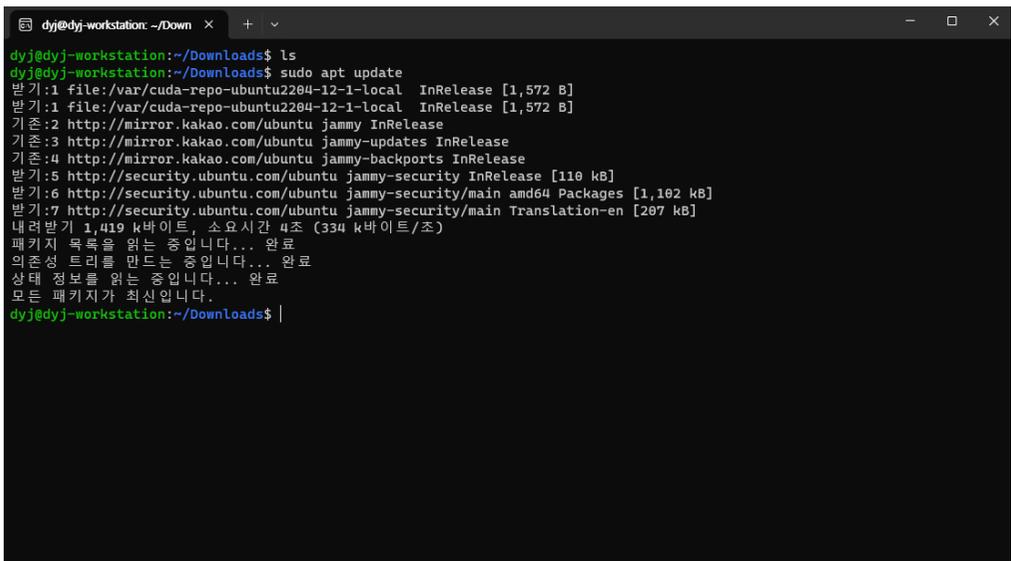
## 4장 구글 크롬

### 4.1 크롬 설치

## 4.1 크롬 설치

우분투 22.04의 xrdp 환경(이하 “원격 데스크톱”)에서 구글 크롬을 설치하는 방법을 소개하겠다. 명령어를 정리하면 다음과 같다.

sudo apt update
sudo apt install wget -y
wget
<a href="https://dl.google.com/linux/direct/google-chrome-stable_current_amd64.deb">https://dl.google.com/linux/direct/google-chrome-stable_current_amd64.deb</a>
sudo dpkg -i ./google-chrome-stable_current_amd64.deb

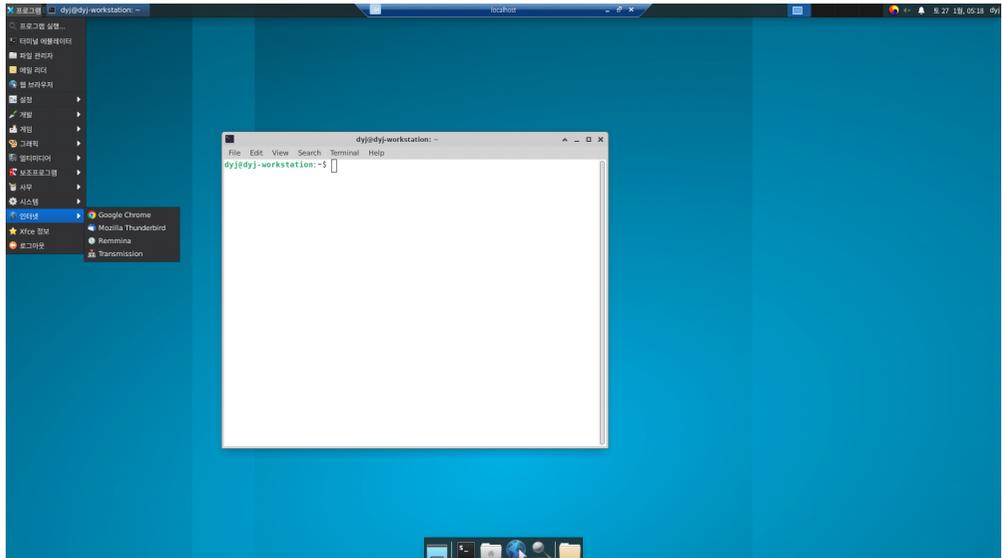


```
djy@djy-workstation: ~/Downloads$ ls
djy@djy-workstation:~/Downloads$ sudo apt update
받기:1 file:/var/cuda-repo-ubuntu2204-12-1-local InRelease [1,572 B]
받기:1 file:/var/cuda-repo-ubuntu2204-12-1-local InRelease [1,572 B]
기존:2 http://mirror.kakao.com/ubuntu jammy InRelease
기존:3 http://mirror.kakao.com/ubuntu jammy-updates InRelease
기존:4 http://mirror.kakao.com/ubuntu jammy-backports InRelease
받기:5 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
받기:6 http://security.ubuntu.com/ubuntu jammy-security/main amd64 Packages [1,102 kB]
받기:7 http://security.ubuntu.com/ubuntu jammy-security/main Translation-en [207 kB]
내려받기 1,419 k바이트, 소요시간 4초 (334 k바이트/초)
패키지 목록을 읽는 중입니다... 완료
의존성 트리를 만드는 중입니다... 완료
상태 정보를 읽는 중입니다... 완료
모든 패키지가 최신입니다.
djy@djy-workstation:~/Downloads$ |
```

[그림 4.1] 구글 크롬 설치(1)

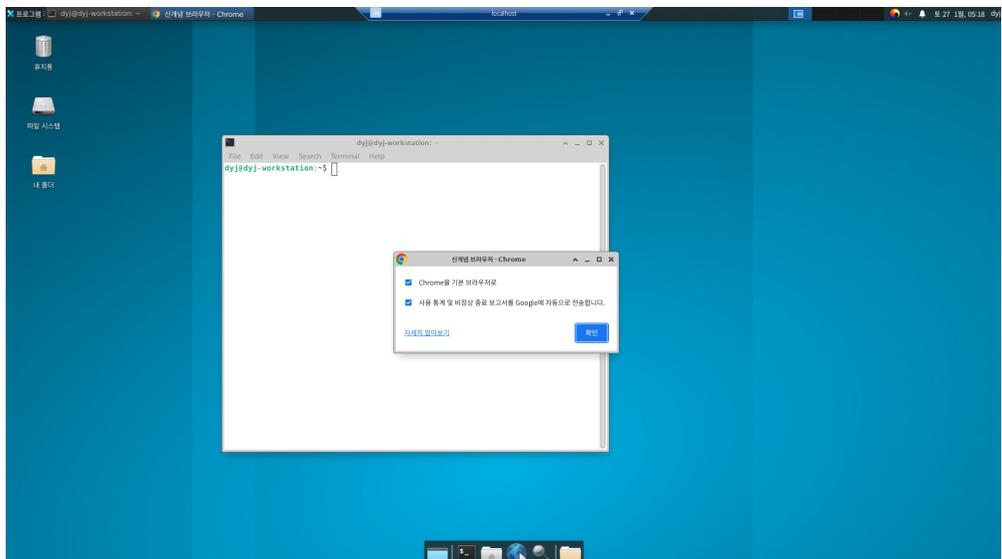


상단의 ‘프로그램’ 메뉴에서 ‘인터넷’ 그 다음에 ‘Google Chrome’을 클릭한다.



[그림 4.4] 구글 크롬 실행(1)

‘확인’을 클릭하면 실행된다.



[그림 4.5] 구글 크롬 실행(2)

## 5장 PyCharm Community

5.1 OpenJDK 설치하기

5.2 PyCharm Community Edition 설치하기

5.3 프로젝트 생성 및 파이썬 코드 실행하기

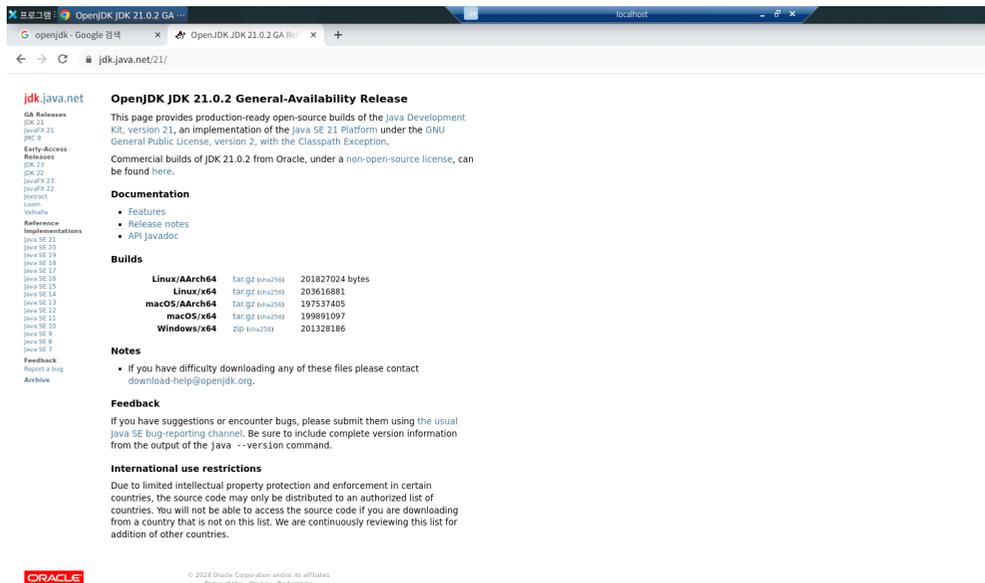
## 5.1 OpenJDK 설치하기

Java Development Kit는 다운로드 받아서 설치하는 가능하나 라이선스 문제로 오픈소스로 공개된 JDK를 사용할 수 있다. 오픈소스로 공개되고 있는 JDK 배포판 중 하나가 AmazonJDK, OpenJDK 등이 있다.

본 실습에서는 오라클에서 오픈소스로 공개하고 있는 OpenJDK 21(2024-01-27 기준, 최신 버전 중 하나)을 설치하는 방법을 소개하겠다.

1단계로 <https://jdk.java.net/21>에 접속한다.

Linux/x64의 tar.gz를 클릭해서 다운받는다.



[그림 5.1] OpenJDK 다운로드 받기 - java.net

2단계로 터미널을 연다.

tar xvf openjdk\*.tar.gz을 입력한다.

```
djy@djy-workstation: ~/Down x + v
jdk-21.0.2/lib/libnio.so
jdk-21.0.2/lib/libprefs.so
jdk-21.0.2/lib/librmi.so
jdk-21.0.2/lib/libsaproc.so
jdk-21.0.2/lib/libscjp.so
jdk-21.0.2/lib/libsplashscreen.so
jdk-21.0.2/lib/libsyslookup.so
jdk-21.0.2/lib/libverify.so
jdk-21.0.2/lib/libzip.so
jdk-21.0.2/lib/modules
jdk-21.0.2/lib/psfont.properties.ja
jdk-21.0.2/lib/psfontj2d.properties
jdk-21.0.2/lib/security/blocked.certs
jdk-21.0.2/lib/security/cacerts
jdk-21.0.2/lib/security/default.policy
jdk-21.0.2/lib/security/public_suffix_list.dat
jdk-21.0.2/lib/server/classes.jsa
jdk-21.0.2/lib/server/classes_nocoops.jsa
jdk-21.0.2/lib/server/libjsig.so
jdk-21.0.2/lib/server/libjvm.so
jdk-21.0.2/lib/src.zip
jdk-21.0.2/lib/tzdb.dat
jdk-21.0.2/release
djy@djy-workstation:~/Downloads$ ls
google-chrome-stable_current_amd64.deb  jdk-21.0.2  openjdk-21.0.2_linux-..._bin.tar.gz
djy@djy-workstation:~/Downloads$ cd jdk-21.0.2/
djy@djy-workstation:~/Downloads/jdk-21.0.2$ sudo mkdir -p /usr/local/jdk-21
[sudo] djy 암호:
djy@djy-workstation:~/Downloads/jdk-21.0.2$ sudo mv * /usr/local/jdk-21
djy@djy-workstation:~/Downloads/jdk-21.0.2$
```

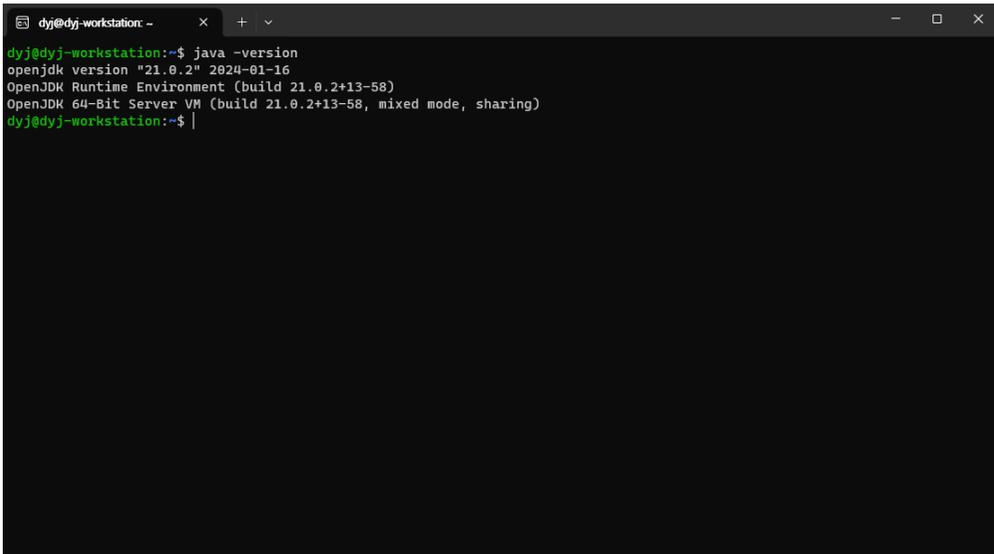
[그림 5.2] OpenJDK 압축 해제하기

3단계는 아래의 정리된 명령어로 openjdk를 설치할 수 있다.

cd jdk-21.0.2/
sudo mkdir -p /usr/local/jdk-21
sudo mv * /usr/local/jdk-21
cd
vim .bashrc 또는 vim .zshrc
2줄 복사 붙여넣기
export JAVA_HOME=/usr/local/jdk-21
export PATH=\$JAVA_HOME/bin:\$PATH
저장 후
source ~/.bashrc 또는 source ~/.zshrc
java --version



4단계로 `java --version`을 입력하여 자바 버전을 확인할 수 있다.

A terminal window titled 'dyj@dyj-workstation: -' with standard window controls. The terminal shows the command 'java -version' being executed. The output is: 'openjdk version "21.0.2" 2024-01-16', 'OpenJDK Runtime Environment (build 21.0.2+13-58)', and 'OpenJDK 64-Bit Server VM (build 21.0.2+13-58, mixed mode, sharing)'. The prompt 'dyj@dyj-workstation:~\$' is visible at the end of the output.

```
dyj@dyj-workstation:~$ java -version
openjdk version "21.0.2" 2024-01-16
OpenJDK Runtime Environment (build 21.0.2+13-58)
OpenJDK 64-Bit Server VM (build 21.0.2+13-58, mixed mode, sharing)
dyj@dyj-workstation:~$ |
```

[그림 5.5] OpenJDK 설치하기(3)

## 5.2 PyCharm Community Edition 설치하기

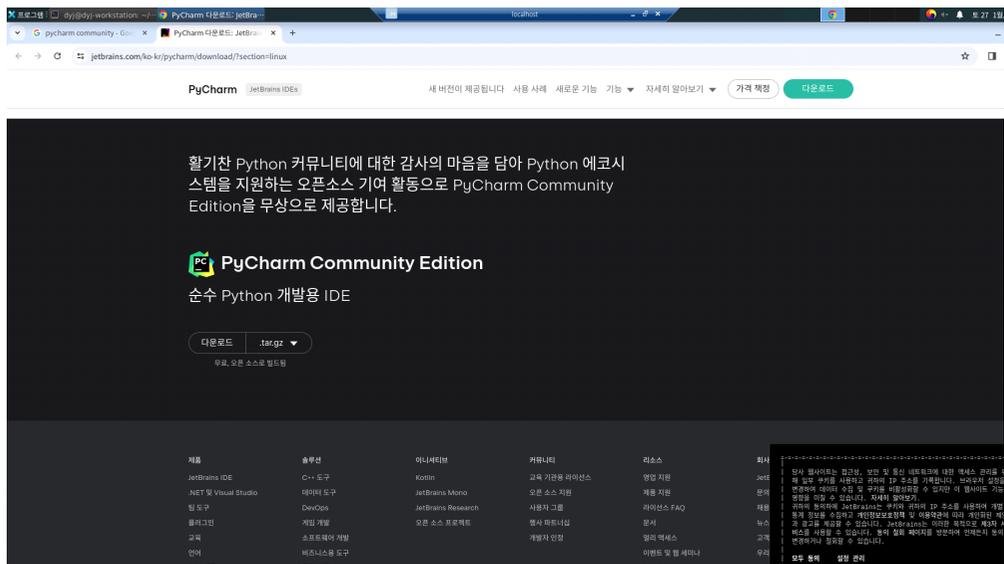
Jetbrain 공식 사이트에 접속한다.

<https://jetbrains.com/ko-kr/pycharm/download/?section=linux>

하단으로 스크롤을 내리면, PyCharm Community Edition을 볼 수 있다.

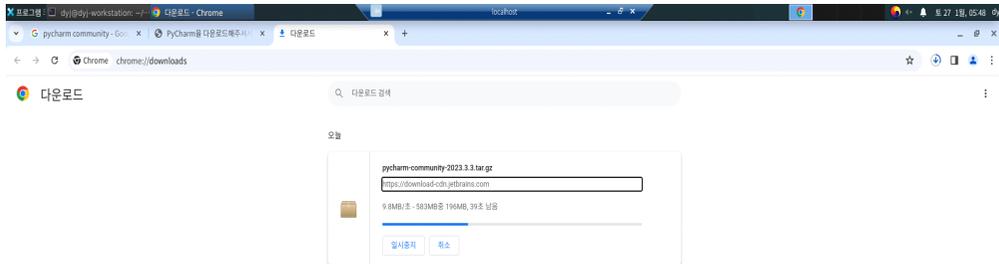
‘다운로드’를 클릭한다.

WSL2 Ubuntu 22.04에서는 apt는 지원하지만, snap 어플리케이션을 지원하지 않는다. 수동으로 작업을 해주어야 한다.



[그림 5.6] PyCharm Community - 웹사이트 접속하기

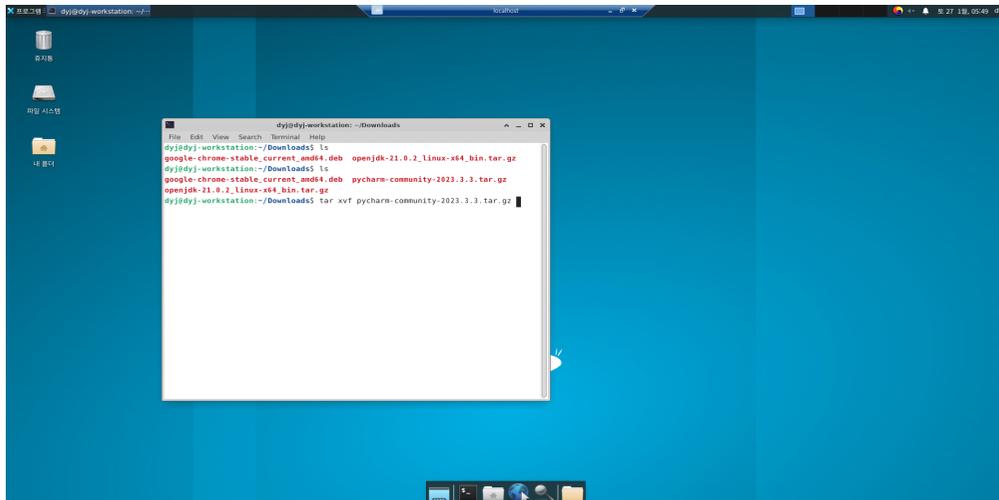
다운로드 과정을 기다린다.



[그림 5.7] PyCharm Community - 다운로드 받기

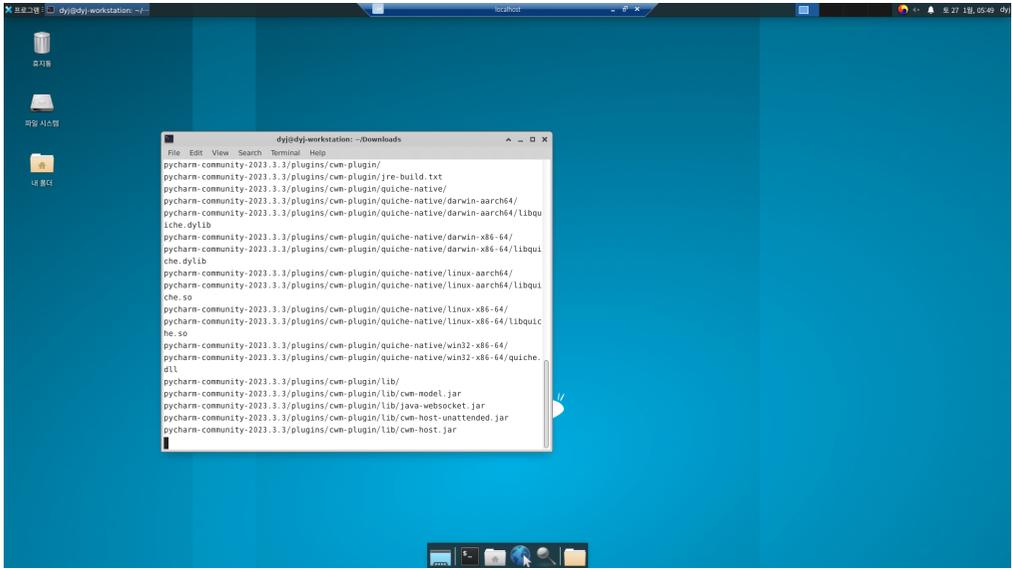
아래의 명령어로 압축을 해제한다.

```
tar xvf pycharm-community*.tar.gz
```



[그림 5.8] PyCharm Community 압축해제(3)

아래 그림은 압축해제 과정이다.



[그림 5.9] PyCharm Community 압축해제(4)

다음 명령어로 압축해제한 PyCharm을 정리 및 실행할 수 있다.

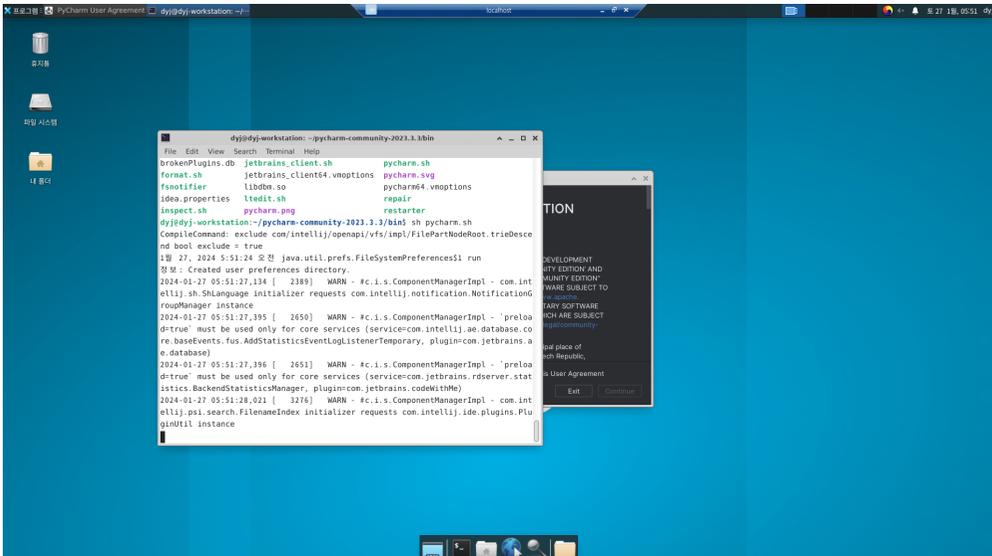
\$ mv pycharm-community-2023.3.3 ~
\$cd
~\$ls
~\$cd pycharm-community-2023.3.3
pycharm-community-2023.3.3\$ ls
pycharm-community-2023.3.3\$ cd bin
bin\$ ls
bin\$ sh pycharm.sh

```

dyj@dyj-workstation: ~/pycharm-community-2023.3.3/bin
File Edit View Search Terminal Help
pycharm-community-2023.3.3/jbr/legal/java.sql.rowset/LICENSE
pycharm-community-2023.3.3/jbr/legal/java.sql.rowset/ADDITIONAL_LICENSE_INFO
pycharm-community-2023.3.3/
pycharm-community-2023.3.3/product-info.json
dyj@dyj-workstation:~/Downloads$ ls
google-chrome-stable_current_amd64.deb  pycharm-community-2023.3.3
openjdk-21.0.2_linux-x64_bin.tar.gz      pycharm-community-2023.3.3.tar.gz
dyj@dyj-workstation:~/Downloads$ mv pycharm-community-2023.3.3 ..
dyj@dyj-workstation:~/Downloads$ cd
dyj@dyj-workstation:~$ ls
Desktop  Downloads  Pictures  Templates  pycharm-community-2023.3.3
Documents  Music  Public  Videos  thinclient_drives
dyj@dyj-workstation:~$ cd pycharm-community-2023.3.3/
dyj@dyj-workstation:~/pycharm-community-2023.3.3$ ls
Install-Linux-tar.txt  build.txt  jbr  license  plugins
bin                    help      lib  modules  product-info.json
dyj@dyj-workstation:~/pycharm-community-2023.3.3$ cd bin
dyj@dyj-workstation:~/pycharm-community-2023.3.3/bin$ ls
brokenPlugins.db  jetbrains_client.sh  pycharm.sh
format.sh         jetbrains_client64.vmoptions  pycharm.svg
fsnotifier        libdbm.so            pycharm64.vmoptions
idea.properties  ltedit.sh            repair
inspect.sh        pycharm.png          restarter
dyj@dyj-workstation:~/pycharm-community-2023.3.3/bin$

```

[그림 5.10] PyCharm 폴더 이동 및 실행 디렉터리 접근  
 'sh pycharm.sh'을 입력하여 PyCharm Community를 실행한다.

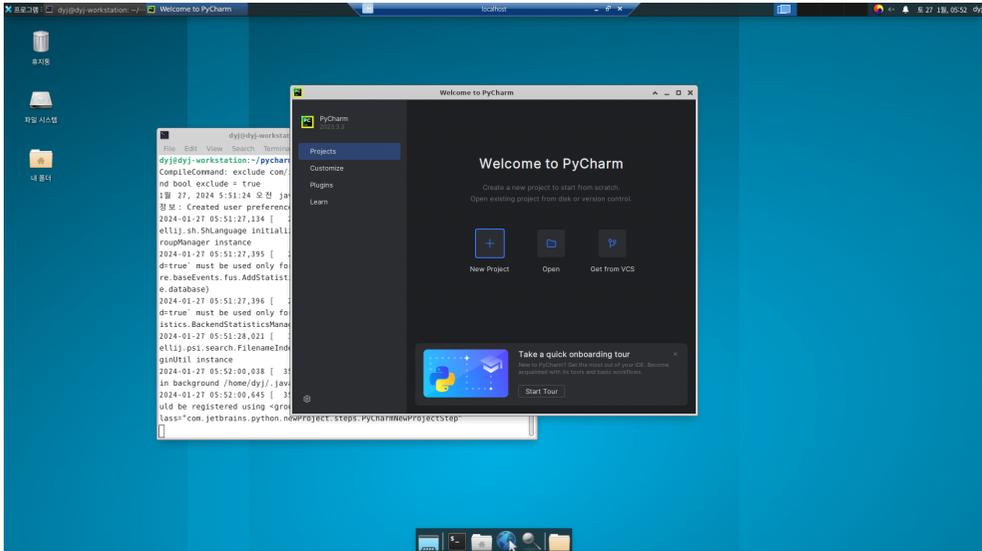


[그림 5.11] PyCharm Community 실행하기

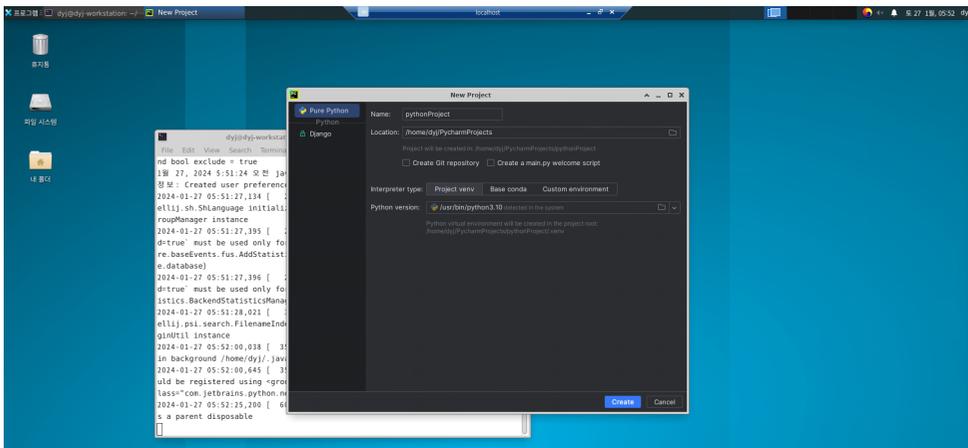
### 5.3 프로젝트 생성 및 파이썬 코드 실행하기

PyCharm이 정상적으로 작동하는 것을 확인할 수 있다.

‘New Project’를 클릭한다.



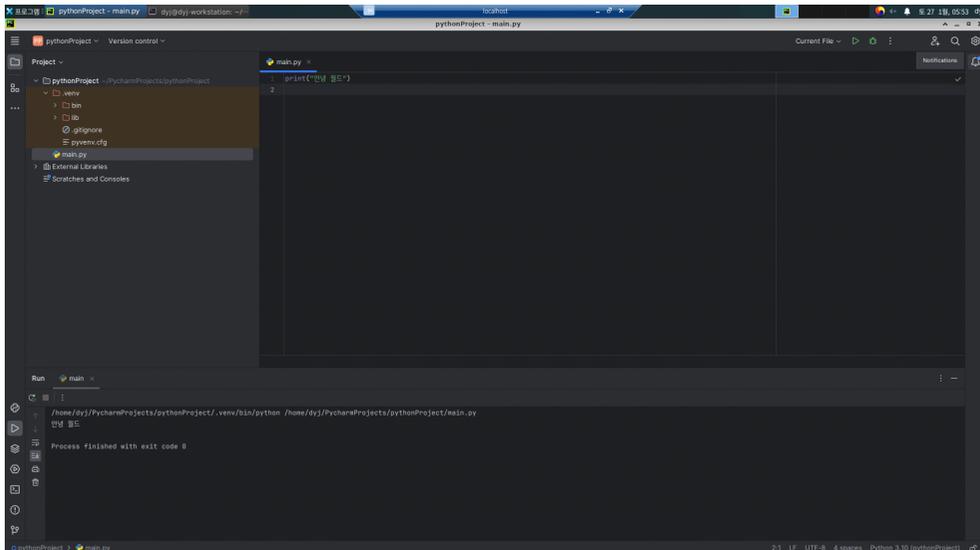
[그림 5.12] PyCharm Community - 새 프로젝트 작업하기(1)  
프로젝트 명, 인터프리터 등의 정보를 입력 후 Create를 누른다.



[그림 5.13] PyCharm Community - 새 프로젝트 작업하기(2)

파이썬 코드를 하나 생성해서, `print("안녕 월드")`라고 작성하고 실행하면 다음과 같은 화면이 나온다.

참고로 MS윈도에서 개발할 때와 리눅스에서 개발할 때는 상대경로, 절대경로 등이 다르다. 이점 참고하면서 개발하기 바란다.



[그림 5.14] PyCharm Community - 파이썬 print함수 사용하기

## 6장 Visual Studio Code

6.1 Visual Studio Code 설치하기

6.2 Visual Studio Code 실행 과정

## 6.1 Visual Studio Code 설치하기

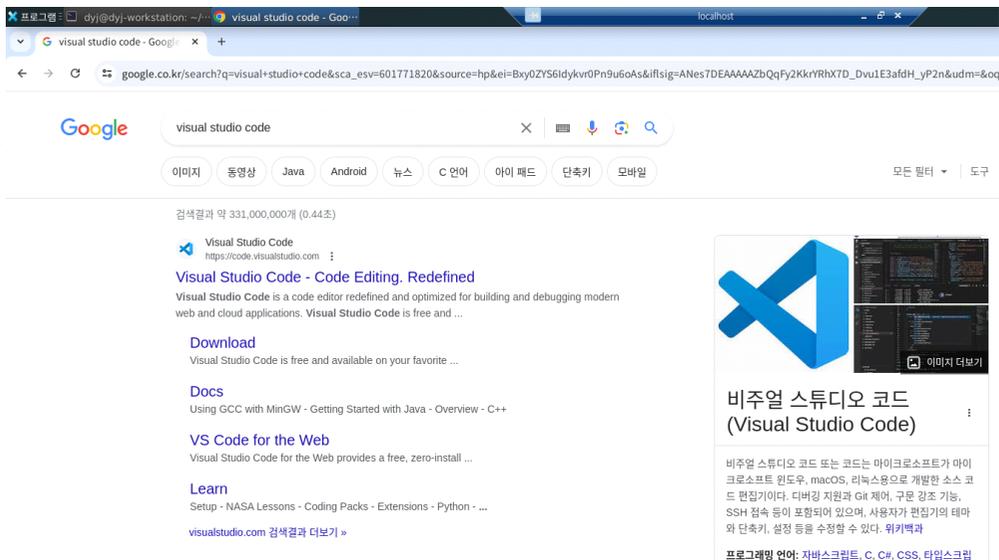
MS윈도에서 Visual Studio Code를 사용해본 사람이 있다면, 굉장히 심플하면서 Visual Studio의 고유 기능을 최소의 용량으로 출시되어서 IDE로 사용하는데 막강하다는 것을 경험해봤을 것이다. 자바, PHP, HTML, Javascript, CSS, C++, Python, C# 등 다양한 개발 언어를 지원한다.

IDE를 소개하는 장이 아니므로 생략은 줄이도록 하겠다.

설치 과정은 매우 간단하다.

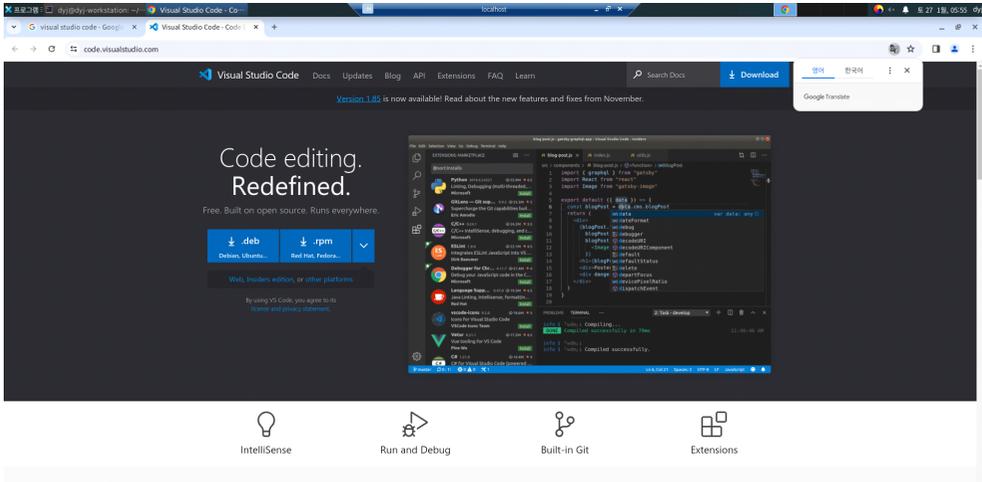
1단계로 검색엔진에서 'visual studio code'를 검색한다.

또는 <https://code.visualstudio.com>에 접속한다.

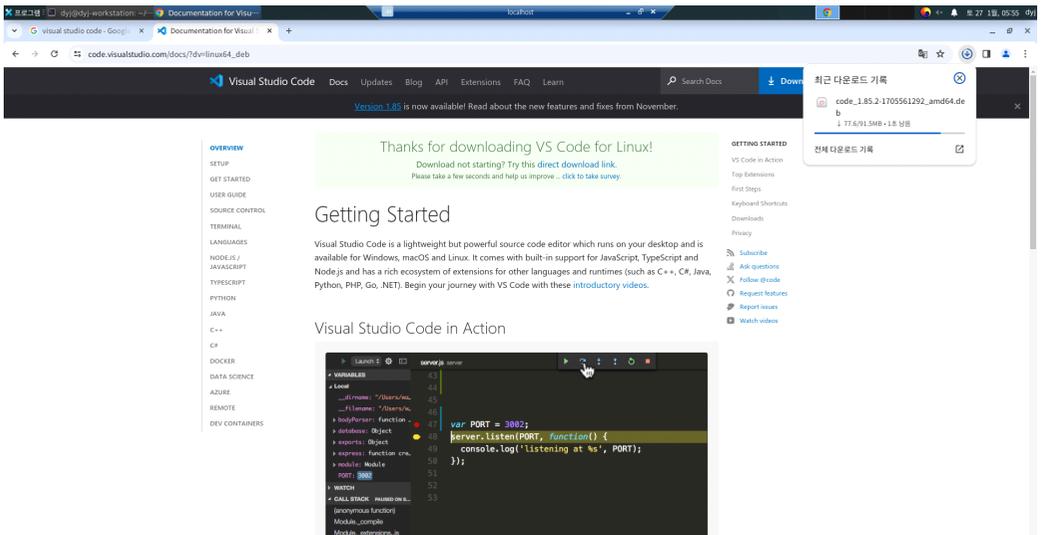


[그림 6.1] 우분투 22.04의 xfce4에서 visual studio code 웹사이트 검색

홈페이지에 접속하면, deb(데비안, 우분투)와 rpm(레드햇, 페도라) 계열을 지원하는 것을 알 수 있다. CentOS 사용자의 경우 rpm을 클릭해서 설치하면 된다. 여기에서는 우분투 22.04를 다루고 있기 때문에 deb를 클릭한다.

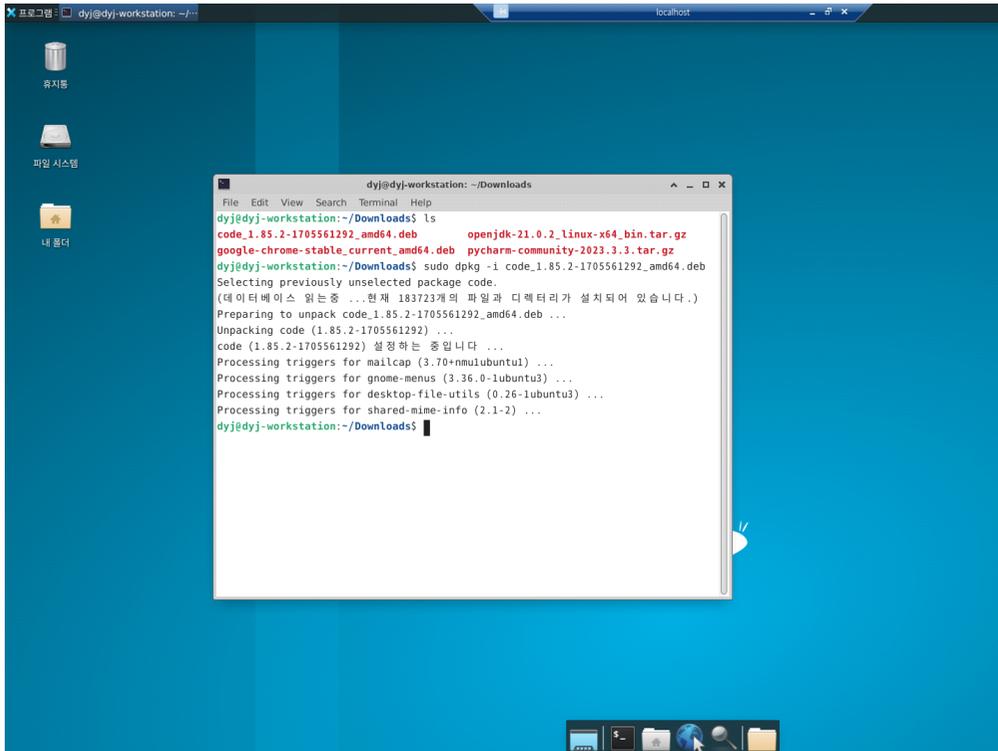


[그림 6.2] visual studio code 리눅스에서의 웹사이트 접속 ‘direct download link’를 클릭한다. 그리고 기다린다.



[그림 6.3] visual studio code 다운로드를 기다리는 모습

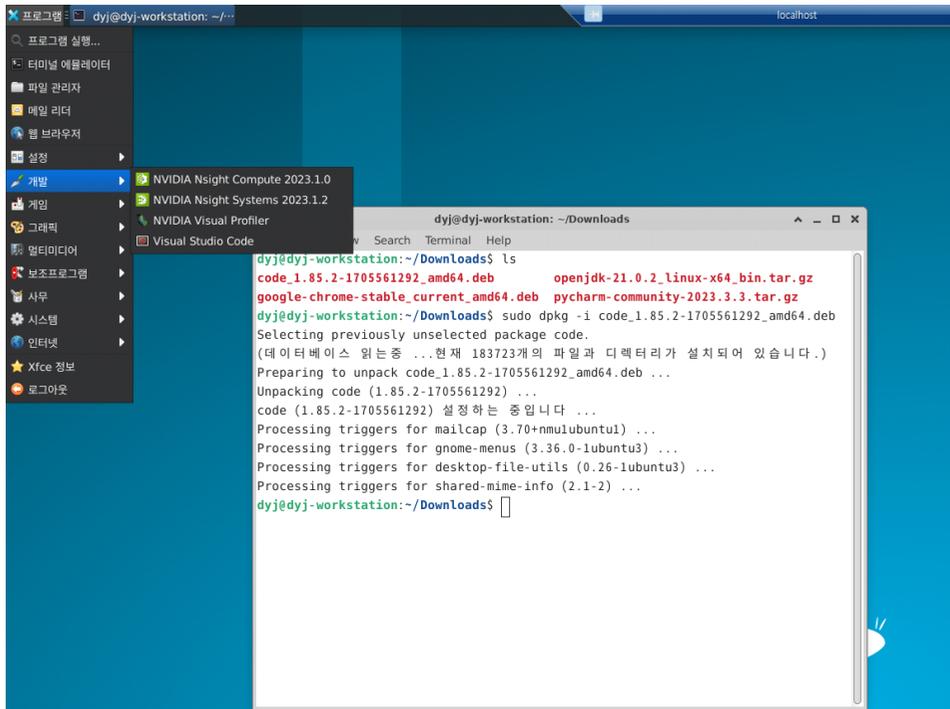
'sudo dpkg -i code\_1.85\*.deb'라고 입력하면 설치할 수 있다.



[그림 6.4] visual studio code 설치

## 6.2 Visual Studio Code 실행 과정

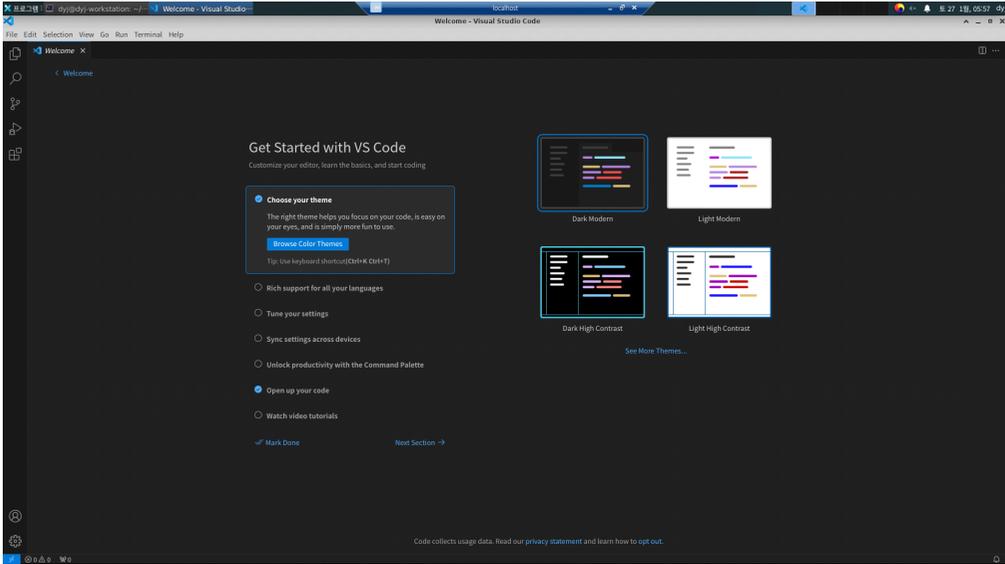
1단계로 상단의 왼쪽 ‘프로그램’에서 ‘개발’ 그 다음에 ‘Visual Studio Code’를 클릭한다.



[그림 6.5] visual studio code 실행하기

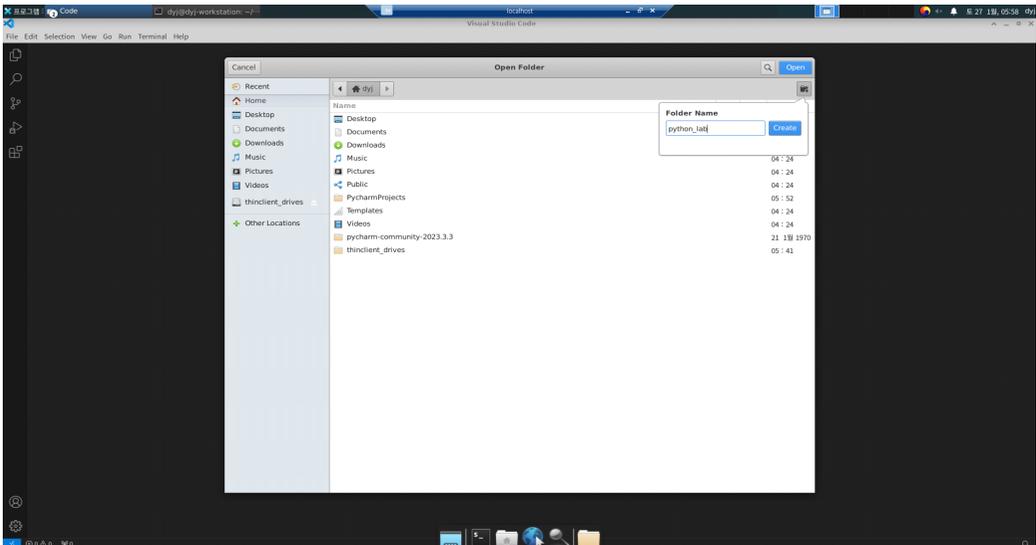
잠시 후 화면이 열린다.

2단계로 File에서 Open Folder를 누른다.



[그림 6.6] visual studio code - 신규 프로젝트 폴더 만들기(1)

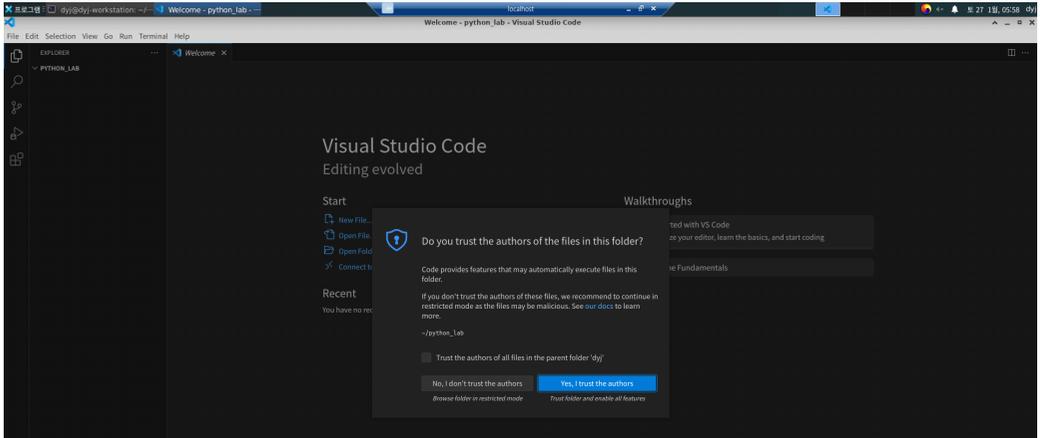
3단계로 '새 폴더'를 만든다. 그리고 폴더를 선택한 후 'Open'을 클릭한다. 본 예제에서는 'python\_lab'으로 폴더를 만들었다.



[그림 6.7] visual studio code - 신규 프로젝트 폴더 만들기(2)

4단계로 프로젝트 폴더에 대해서 신뢰하는지를 묻는 창이다.

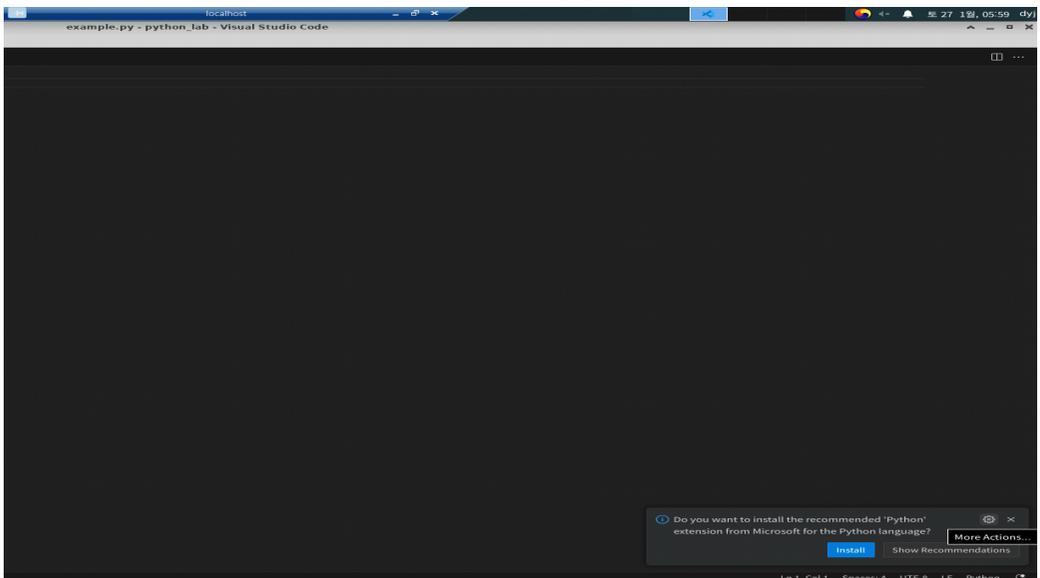
‘Yes, I trust’를 클릭한다.



[그림 6.7] visual studio code - 프로젝트 폴더 신뢰 여부

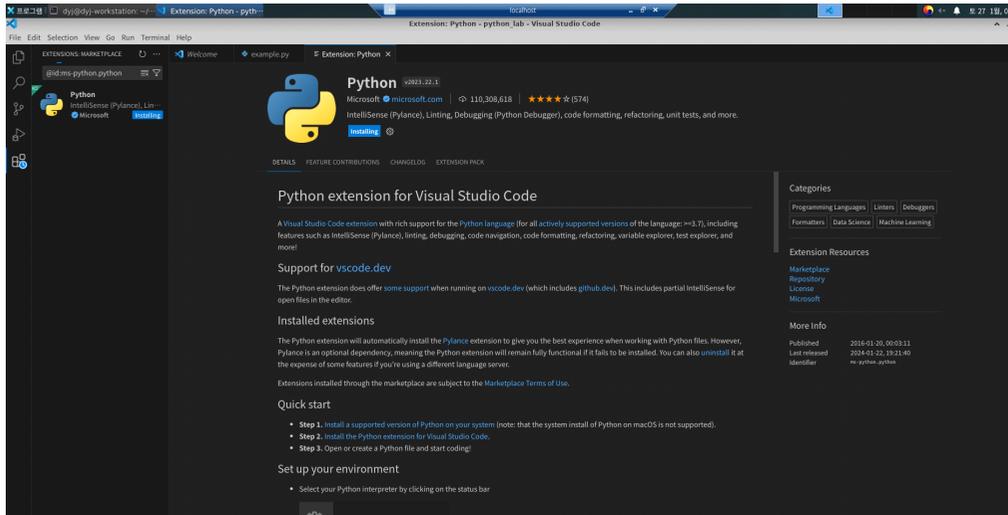
5단계로 오른쪽 하단에 ‘python ~~~ install’이라는 창이 뜬다.

‘install’을 클릭한다.



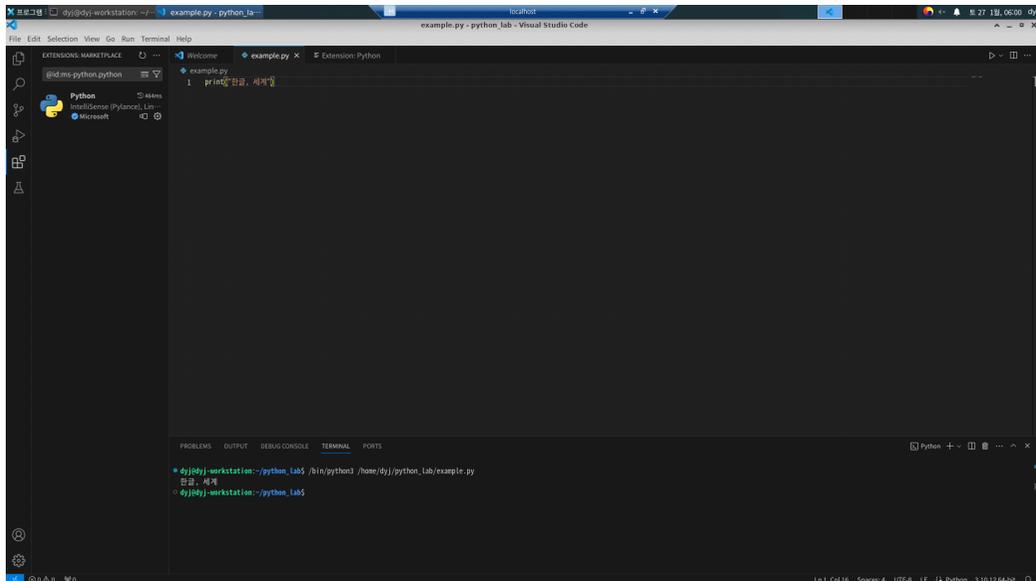
[그림 6.8] visual studio code - python 설치

6단계로 Visual Studio Code에서 Python이 설치되기를 기다린다.



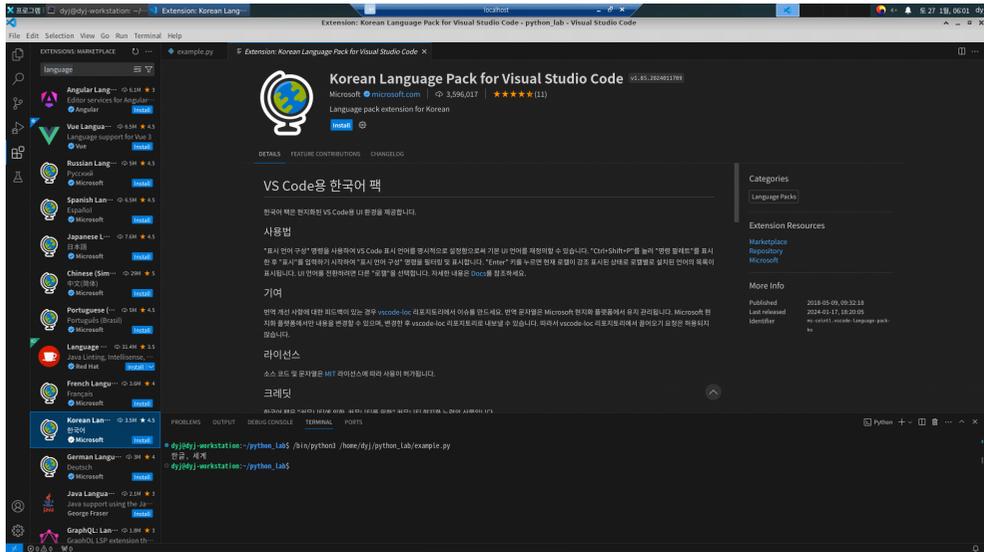
[그림 6.9] visual studio code - python 설치

7단계로 파이썬 파일을 하나 만든다. 그리고 `print("한글, 세계")`라고 입력 후 실행한다.

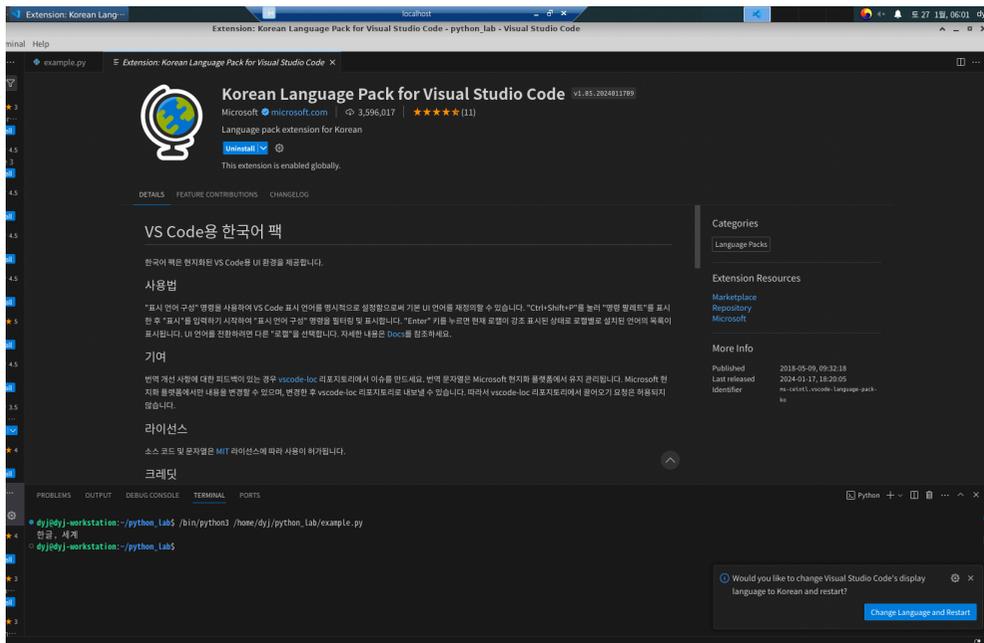


[그림 6.10] visual studio code - python 실행

8단계로 추가 확장팩으로 한글 언어팩을 설치하여 사용할 수 있다.



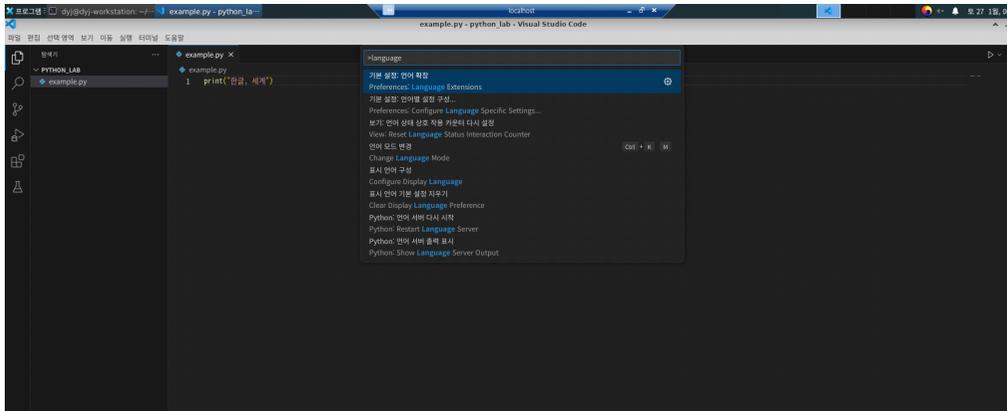
[그림 6.11] visual studio code - 한글 언어팩 설치(1)



[그림 6.12] visual studio code - 한글 언어팩 설치(2)

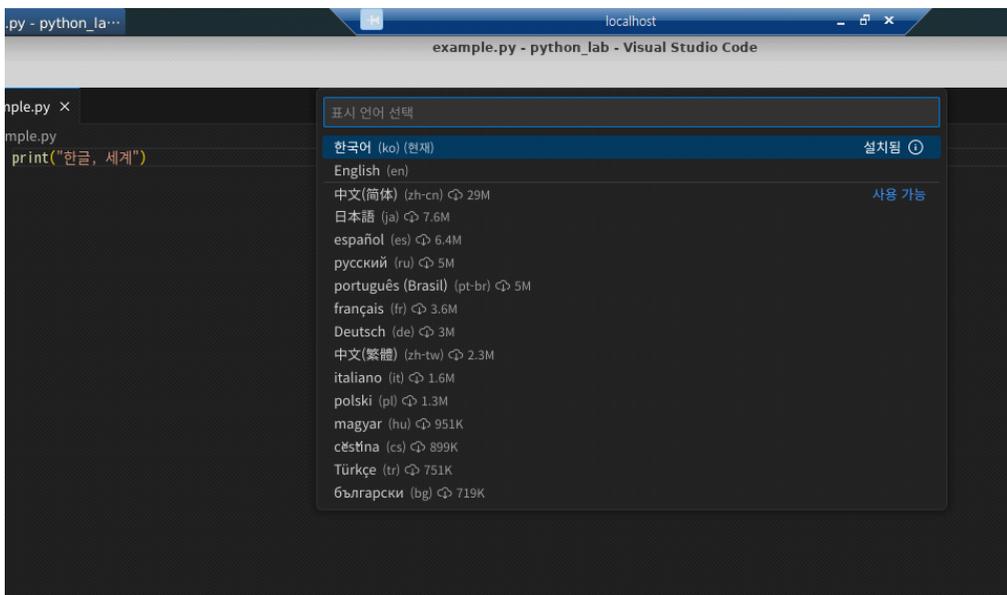
9단계로 한글 언어팩에서 영어 언어팩으로 변경하는 방법이다.

ctrl+shift+p 키를 누른 다음에 'language'라고 입력한 후 '표시 언어 구성'을 클릭한다.



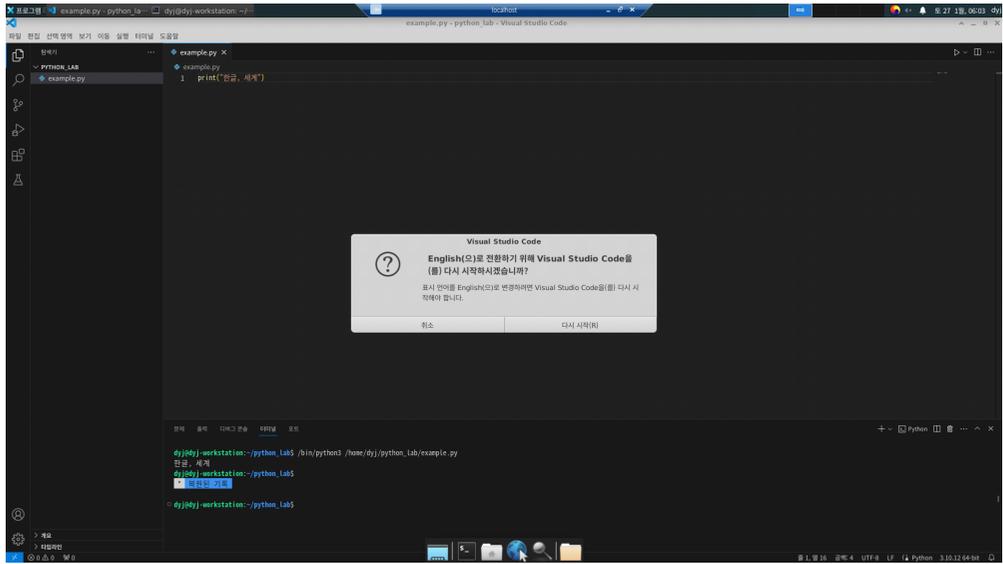
[그림 6.13] visual studio code - 표시 언어 구성(1)

10단계로 'English'를 클릭하면 바꿀 수 있다.



[그림 6.14] visual studio code - 표시 언어 구성(2)

11단계로 ‘다시 시작’을 클릭하면 언어 팩이 변경된다.



[그림 6.15] visual studio code - 표시 언어 구성(3)

## 7장 Miniconda 설치하기

7.1 Miniconda 설치하기

7.2 bashrc 수정하기

7.3 conda를 활용한 jupyter 설치하기

7.4 Zsh, Oh-my-zsh 설치하기

7.5 zshrc 수정하기

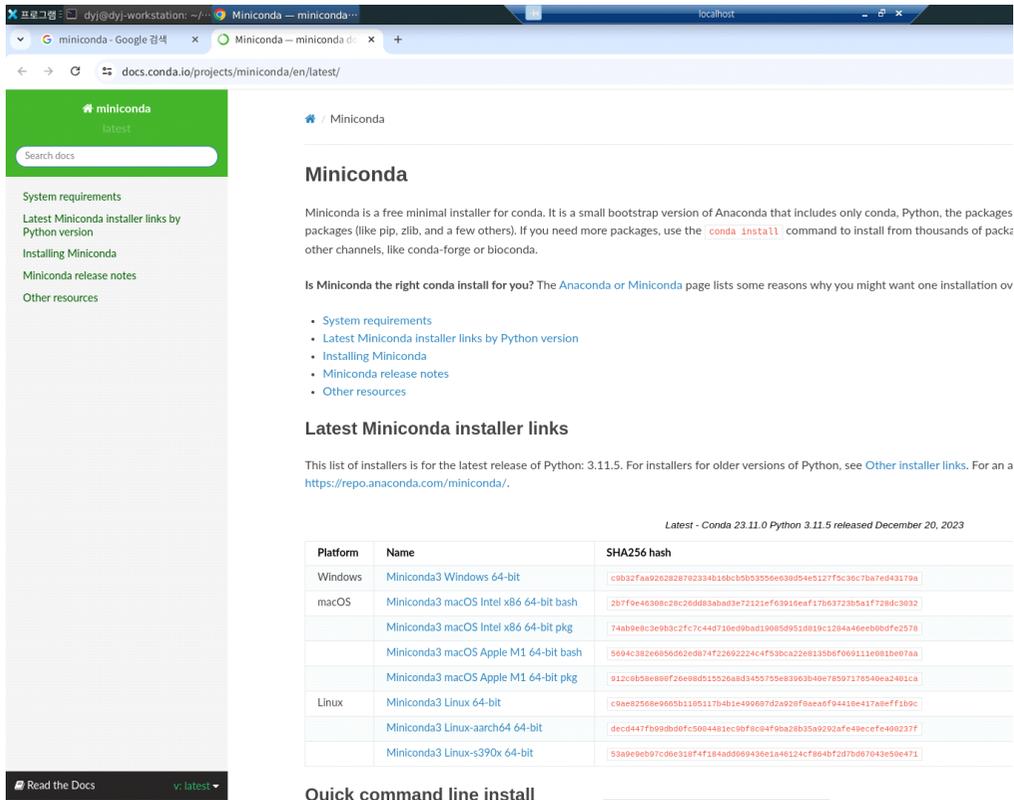
7.6 PyTorch 설치하기

7.7 PyCharm Community PATH 추가 반영하기

7.8 Visual Studio Code에서 torch 사용 확인하기

## 7.1 Miniconda 설치하기

리눅스에서 경량화된 python 개발 패키지 중 하나가 miniconda이다.  
1단계로 <https://docs.conda.io/projects/miniconda/en/latest>에 접속한다.



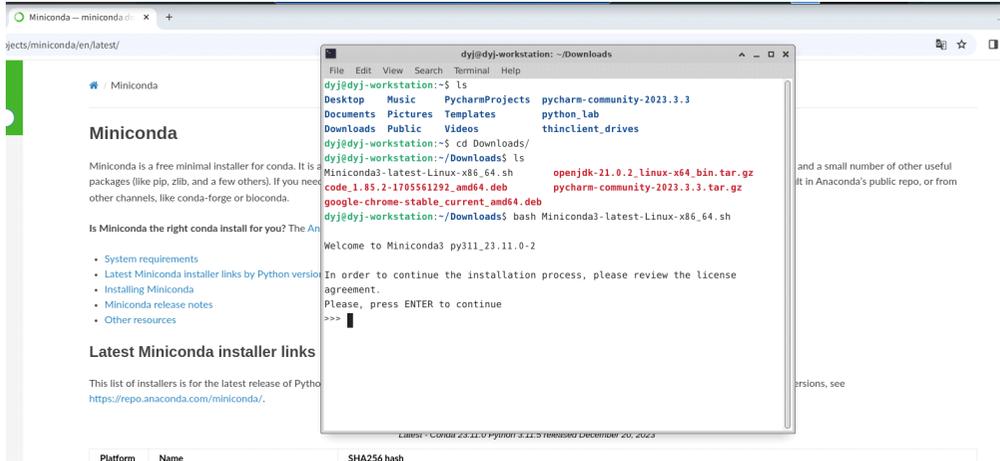
[그림 7.1] Miniconda3 - docs 홈페이지

2단계로 위에 그림에서 'Miniconda3 Linux 64-bit'를 마우스 오른쪽 버튼에서 '링크 복사'를 클릭한다.

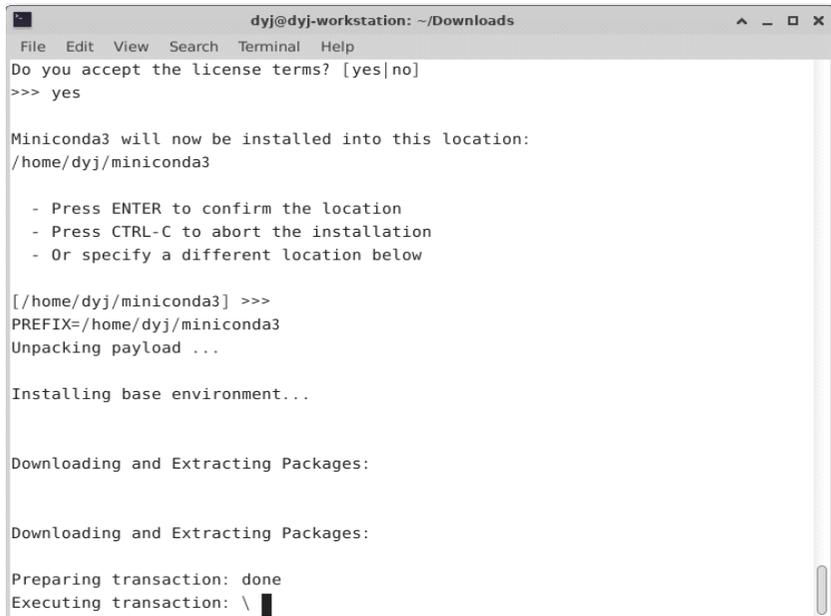
3단계로 터미널을 연다.

```
bash Miniconda3-latest-Linux-x86_64.sh
```

명령어를 입력한다. ENTER키를 누르고 space바를 눌러 설치를 진행한다.



[그림 7.2] Miniconda3 - 설치 진행 모습(1)

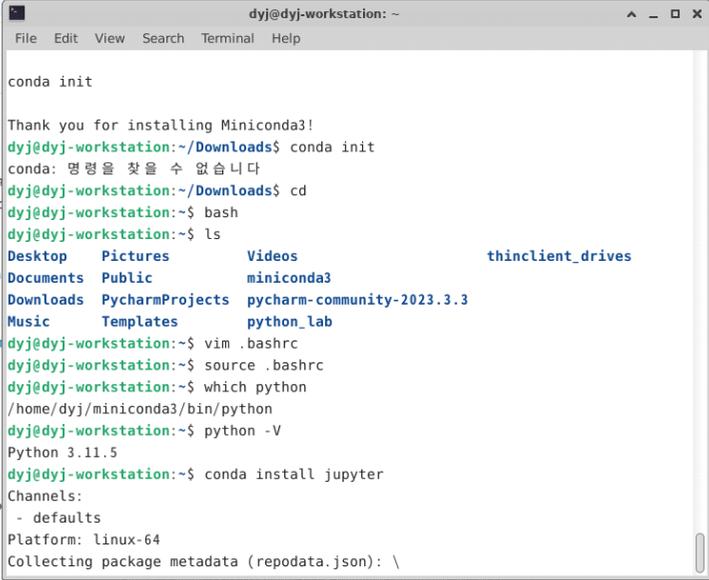


[그림 7.3] Miniconda3 - 설치 진행 모습(2)

## 7.2 bashrc 수정하기

사용한 명령어는 다음과 같다.

conda init
cd
bash
ls
vim .bashrc
마지막 줄에 한 줄 추가
<pre>export PATH="\$HOME/miniconda3/bin:\$PATH"</pre>
ESC키 :wq로 저장
source .bashrc
which python
python -V
conda install jupyter



```
dyj@dyj-workstation: ~
File Edit View Search Terminal Help

conda init

Thank you for installing Miniconda3!
dyj@dyj-workstation:~/Downloads$ conda init
conda: 명령을 찾을 수 없습니다
dyj@dyj-workstation:~/Downloads$ cd
dyj@dyj-workstation:~$ bash
dyj@dyj-workstation:~$ ls
Desktop Pictures Videos thinclient_drives
Documents Public miniconda3
Downloads PycharmProjects pycharm-community-2023.3.3
Music Templates python_lab
dyj@dyj-workstation:~$ vim .bashrc
dyj@dyj-workstation:~$ source .bashrc
dyj@dyj-workstation:~$ which python
/home/dyj/miniconda3/bin/python
dyj@dyj-workstation:~$ python -V
Python 3.11.5
dyj@dyj-workstation:~$ conda install jupyter
Channels:
- defaults
Platform: linux-64
Collecting package metadata (repodata.json): \
Latest - Conda 23.11.0 Python 3.11.5 released December 20, 2023
```

SHA256 hash

[그림 7.4] Miniconda3 - 설정 작업

```
vim
# Add an "alert" alias for long running commands.  Use like so:
#   sleep 10; alert
alias alert='notify-send --urgency=low -i "${[ $? = 0 ] && echo terminal || echo error}" "${history|tail -n1|sed -e '\''
s/\s*[0-9]\+\s*//;s/[:;&]\s*alert$/'\''"'

# Alias definitions.
# You may want to put all your additions into a separate file like
# ~/.bash_aliases, instead of adding them here directly.
# See /usr/share/doc/bash-doc/examples in the bash-doc package.

if [ -f ~/.bash_aliases ]; then
    . ~/.bash_aliases
fi

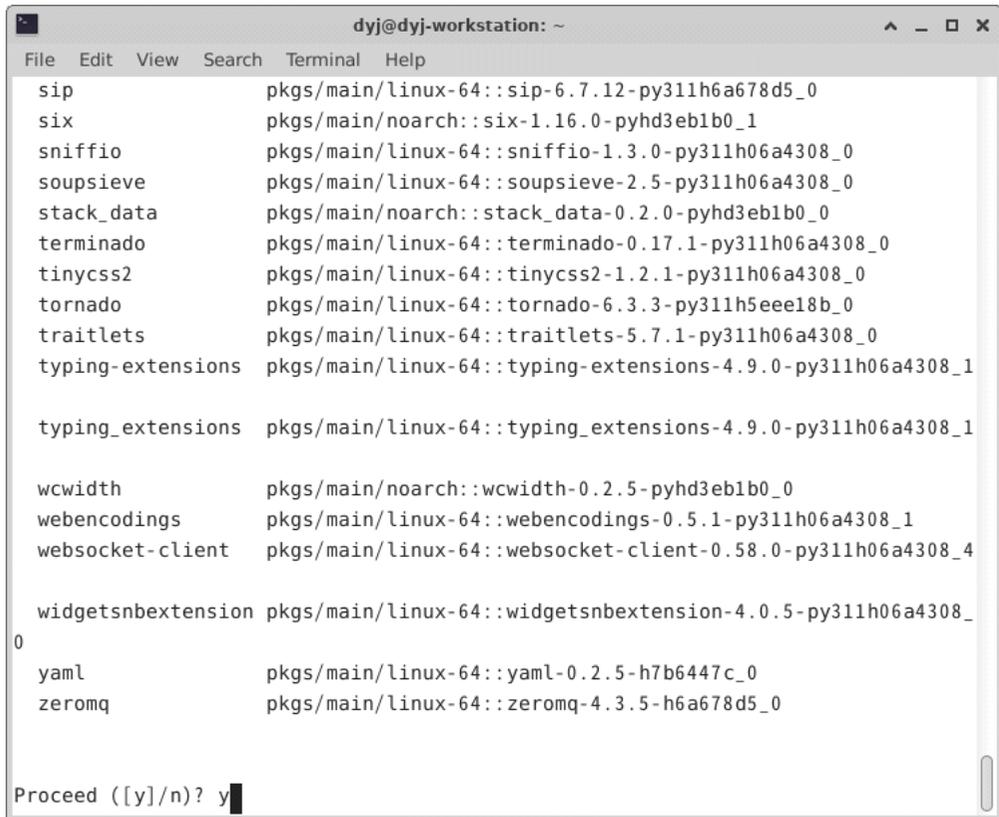
# enable programmable completion features (you don't need to enable
# this, if it's already enabled in /etc/bash.bashrc and /etc/profile
# sources /etc/bash.bashrc).
if ! shopt -oq posix; then
    if [ -f /usr/share/bash-completion/bash_completion ]; then
        . /usr/share/bash-completion/bash_completion
    elif [ -f /etc/bash_completion ]; then
        . /etc/bash_completion
    fi
fi

export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/usr/local/cuda-12.1/lib64/
export JAVA_HOME=/usr/local/jdk-21
export PATH=$JAVA_HOME/bin:$PATH
export PATH=$HOME/miniconda3/bin:$PATH
".bashrc" 122L, 3949B 122,40 박막
```

[그림 7.5] Miniconda3 - 설정 작업

### 7.3 conda를 활용한 jupyter 설치하기

1단계로 7.2에서 conda install jupyter를 입력하면 다음과 같은 메시지가 뜬다. y를 입력 후 ENTER키를 누른다.

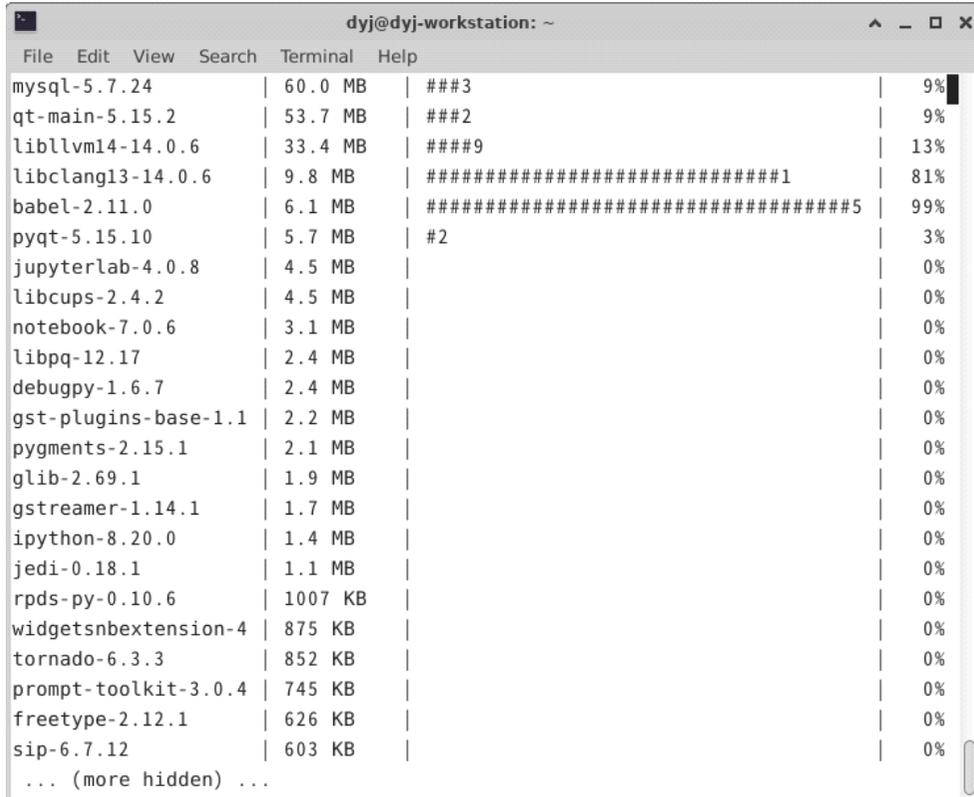


```
dyj@dyj-workstation: ~  
File Edit View Search Terminal Help  
sip                pkgs/main/linux-64::sip-6.7.12-py311h6a678d5_0  
six                pkgs/main/noarch::six-1.16.0-pyhd3eb1b0_1  
sniffio            pkgs/main/linux-64::sniffio-1.3.0-py311h06a4308_0  
soupsieve          pkgs/main/linux-64::soupsieve-2.5-py311h06a4308_0  
stack_data         pkgs/main/noarch::stack_data-0.2.0-pyhd3eb1b0_0  
terminado          pkgs/main/linux-64::terminado-0.17.1-py311h06a4308_0  
tinycss2           pkgs/main/linux-64::tinycss2-1.2.1-py311h06a4308_0  
tornado            pkgs/main/linux-64::tornado-6.3.3-py311h5eee18b_0  
traitlets          pkgs/main/linux-64::traitlets-5.7.1-py311h06a4308_0  
typing-extensions  pkgs/main/linux-64::typing-extensions-4.9.0-py311h06a4308_1  
  
typing_extensions pkgs/main/linux-64::typing_extensions-4.9.0-py311h06a4308_1  
  
wcwidth            pkgs/main/noarch::wcwidth-0.2.5-pyhd3eb1b0_0  
webencodings       pkgs/main/linux-64::webencodings-0.5.1-py311h06a4308_1  
websocket-client   pkgs/main/linux-64::websocket-client-0.58.0-py311h06a4308_4  
  
widgetsnextension  pkgs/main/linux-64::widgetsnextension-4.0.5-py311h06a4308_0  
yaml                pkgs/main/linux-64::yaml-0.2.5-h7b6447c_0  
zeromq             pkgs/main/linux-64::zeromq-4.3.5-h6a678d5_0  
  
Proceed ([y]/n)? y
```

[그림 7.6] Miniconda3 - conda를 활용하여 jupyter 설치하기(1)

2단계로 다운로드 작업 및 설치 작업이 진행된다.

잠시 기다린다. 인터넷 속도 사정에 따라서 시간이 달라질 수도 있다.



A terminal window titled 'dyj@dyj-workstation: ~' showing the output of a conda environment list command. The window has a menu bar with 'File', 'Edit', 'View', 'Search', 'Terminal', and 'Help'. The output is a table with columns for package name, size, channel, and progress percentage. The progress bars are represented by vertical bars of varying lengths. The packages listed include mysql-5.7.24, qt-main-5.15.2, libllvml4-14.0.6, libclang13-14.0.6, babel-2.11.0, pyqt-5.15.10, jupyterlab-4.0.8, libcups-2.4.2, notebook-7.0.6, libpq-12.17, debugpy-1.6.7, gst-plugins-base-1.1, pygments-2.15.1, glib-2.69.1, gstreamer-1.14.1, ipython-8.20.0, jedi-0.18.1, rpds-py-0.10.6, widgetsnbextension-4, tornado-6.3.3, prompt-toolkit-3.0.4, freetype-2.12.1, and sip-6.7.12. At the bottom, it says '... (more hidden) ...'.

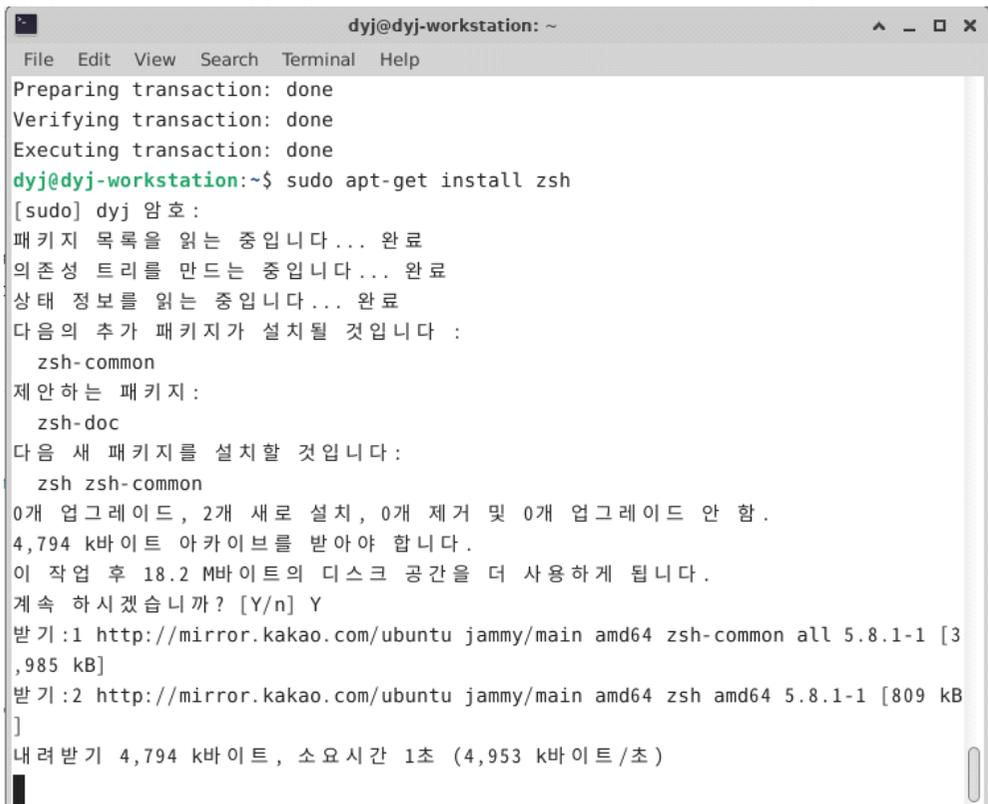
Package Name	Size	Channel	Progress
mysql-5.7.24	60.0 MB	###3	9%
qt-main-5.15.2	53.7 MB	###2	9%
libllvml4-14.0.6	33.4 MB	####9	13%
libclang13-14.0.6	9.8 MB	#####1	81%
babel-2.11.0	6.1 MB	#####5	99%
pyqt-5.15.10	5.7 MB	#2	3%
jupyterlab-4.0.8	4.5 MB		0%
libcups-2.4.2	4.5 MB		0%
notebook-7.0.6	3.1 MB		0%
libpq-12.17	2.4 MB		0%
debugpy-1.6.7	2.4 MB		0%
gst-plugins-base-1.1	2.2 MB		0%
pygments-2.15.1	2.1 MB		0%
glib-2.69.1	1.9 MB		0%
gstreamer-1.14.1	1.7 MB		0%
ipython-8.20.0	1.4 MB		0%
jedi-0.18.1	1.1 MB		0%
rpds-py-0.10.6	1007 KB		0%
widgetsnbextension-4	875 KB		0%
tornado-6.3.3	852 KB		0%
prompt-toolkit-3.0.4	745 KB		0%
freetype-2.12.1	626 KB		0%
sip-6.7.12	603 KB		0%
... (more hidden) ...			

[그림 7.7] Miniconda3 - conda를 활용하여 jupyter 설치하기(2)

## 7.4 Zsh, Oh-my-zsh 설치하기

1단계로 설치에 사용한 명령어는 다음과 같다.

```
sudo apt-get install zsh
chsh -s /usr/bin/zsh
재부팅 한다.
sh -c "$(curl -fsSL
https://raw.githubusercontent.com/robbyrussell/oh-my-zsh/master/tools/
install.sh)"
```



```
dyj@dyj-workstation: ~
File Edit View Search Terminal Help
Preparing transaction: done
Verifying transaction: done
Executing transaction: done
dyj@dyj-workstation:~$ sudo apt-get install zsh
[sudo] dyj 암호:
패키지 목록을 읽는 중입니다... 완료
의존성 트리를 만드는 중입니다... 완료
상태 정보를 읽는 중입니다... 완료
다음의 추가 패키지가 설치될 것입니다 :
  zsh-common
제안하는 패키지:
  zsh-doc
다음 새 패키지를 설치할 것입니다:
  zsh zsh-common
0개 업그레이드, 2개 새로 설치, 0개 제거 및 0개 업그레이드 안 함.
4,794 k바이트 아카이브를 받아야 합니다.
이 작업 후 18.2 M바이트의 디스크 공간을 더 사용하게 됩니다.
계속 하시겠습니까? [Y/n] Y
받기:1 http://mirror.kakao.com/ubuntu jammy/main amd64 zsh-common all 5.8.1-1 [3,985 kB]
받기:2 http://mirror.kakao.com/ubuntu jammy/main amd64 zsh amd64 5.8.1-1 [809 kB]
]
내려받기 4,794 k바이트, 소요시간 1초 (4,953 k바이트/초)
```

[그림 7.8] zsh 설치(1)

```
dyj@dyj-workstation: ~
File Edit View Search Terminal Help
다음 새 패키지를 설치할 것입니다:
  zsh zsh-common
0개 업그레이드, 2개 새로 설치, 0개 제거 및 0개 업그레이드 안 함.
4,794 k바이트 아카이브를 받아야 합니다.
이 작업 후 18.2 M바이트의 디스크 공간을 더 사용하게 됩니다.
계속 하시겠습니까? [Y/n] Y
받기:1 http://mirror.kakao.com/ubuntu jammy/main amd64 zsh-common all 5.8.1-1 [3
,985 kB]
받기:2 http://mirror.kakao.com/ubuntu jammy/main amd64 zsh amd64 5.8.1-1 [809 kB
]
내려받기 4,794 k바이트, 소요시간 1초 (4,953 k바이트/초)
Selecting previously unselected package zsh-common.
(데이터베이스 읽는중 ...현재 185167개의 파일과 디렉터리가 설치되어 있습니다.)
Preparing to unpack .../zsh-common_5.8.1-1_all.deb ...
Unpacking zsh-common (5.8.1-1) ...
Selecting previously unselected package zsh.
Preparing to unpack .../archives/zsh_5.8.1-1_amd64.deb ...
Unpacking zsh (5.8.1-1) ...
zsh-common (5.8.1-1) 설정하는 중입니다 ...
zsh (5.8.1-1) 설정하는 중입니다 ...
Processing triggers for man-db (2.10.2-1) ...
dyj@dyj-workstation:~$ chsh -s /usr/bin/zsh
암호:
dyj@dyj-workstation:~$
```

[그림 7.9] zsh 설치(2)

2단계로 재부팅 후에 새로운 메시지를 볼 수 있다. ENTER 키를 입력한다.

```
Ubuntu
This is the Z Shell configuration function for new users,
zsh-newuser-install.
You are seeing this message because you have no zsh startup files
(the files .zshenv, .zprofile, .zshrc, .zlogin in the directory
~). This function can help you with a few settings that should
make your use of the shell easier.

You can:

(q) Quit and do nothing. The function will be run again next time.

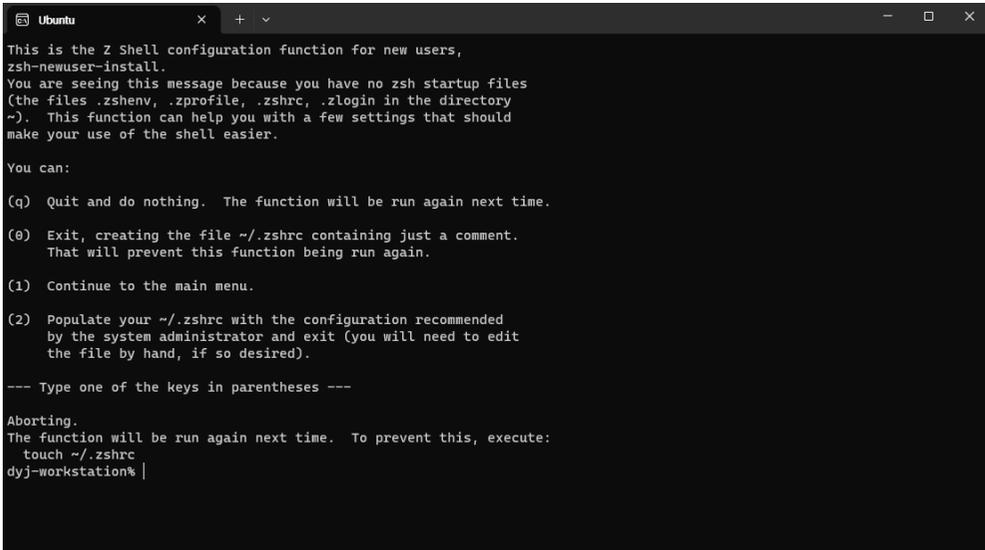
(0) Exit, creating the file ~/.zshrc containing just a comment.
That will prevent this function being run again.

(1) Continue to the main menu.

(2) Populate your ~/.zshrc with the configuration recommended
by the system administrator and exit (you will need to edit
the file by hand, if so desired).

--- Type one of the keys in parentheses ---
```

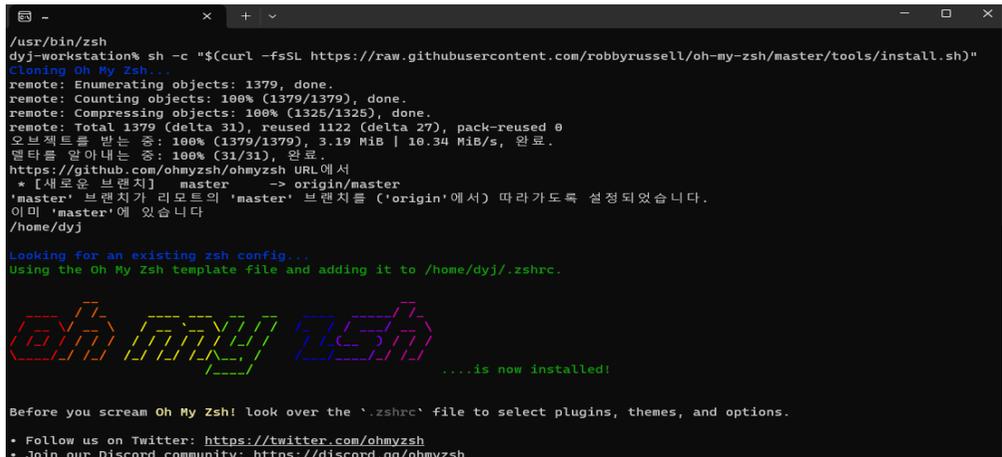
[그림 7.10] zsh 설치(3)



[그림 7.11] zsh 설치(4)

3단계로 oh-my-zsh를 아래의 명령어로 설치한다.

```
sh -c "$(curl -fsSL
https://raw.githubusercontent.com/robbyrussell/oh-my-zsh/master/tools
/install.sh)"
```



[그림 7.12] oh-my-zsh 설치(1)

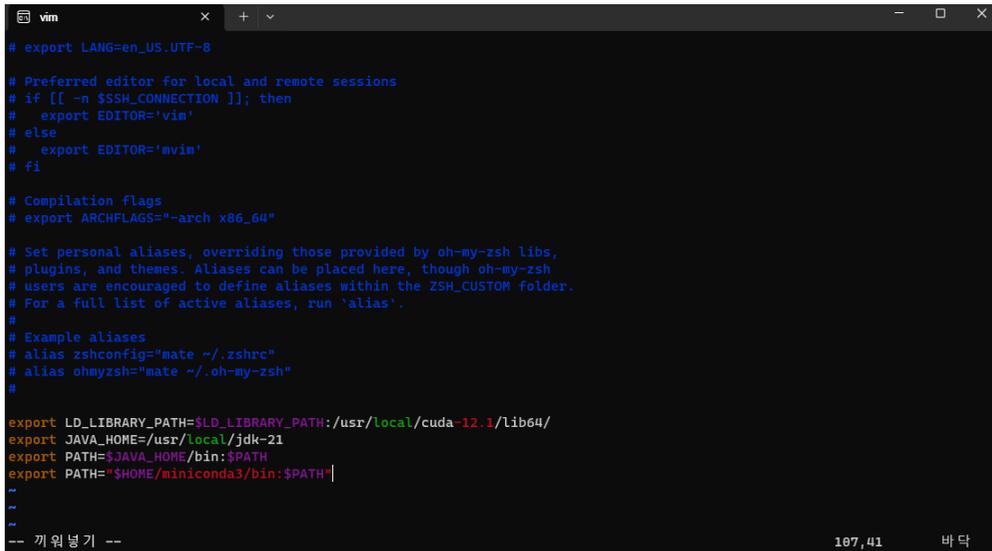
4단계로 oh-my-zsh가 설치되었다. pip, pip3 명령어가 동작하지 않는다.

## 7.5 zshrc 수정하기

pip, pip3이 동작하지 않는 이유는 .zshrc 파일을 수정하지 않았기 때문이다.  
vim 명령어로 .bashrc의 하단에 있는 내용을 복사한다.

그리고 아래의 정리된 명령어 순서대로 파일을 수정한다.

\$ vim .bashrc
하단 복사하기
\$ vim .zshrc
복사한 하단의 export 내용 붙여넣기
\$ source .zshrc



```
vim
# export LANG=en_US.UTF-8

# Preferred editor for local and remote sessions
# if [[ -n $SSH_CONNECTION ]]; then
#   export EDITOR='vim'
# else
#   export EDITOR='mvim'
# fi

# Compilation flags
# export ARCHFLAGS="-arch x86_64"

# Set personal aliases, overriding those provided by oh-my-zsh libs,
# plugins, and themes. Aliases can be placed here, though oh-my-zsh
# users are encouraged to define aliases within the ZSH_CUSTOM folder.
# For a full list of active aliases, run `alias`.
#
# Example aliases
# alias zshconfig="mate ~/.zshrc"
# alias ohmyzsh="mate ~/.oh-my-zsh"
#

export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/usr/local/cuda-12.1/lib64/
export JAVA_HOME=/usr/local/jdk-21
export PATH=$JAVA_HOME/bin:$PATH
export PATH="$HOME/miniconda3/bin:$PATH"
~
~
-- 끼워 넣기 --
```

[그림 7.13] .zshrc 파일 수정 모습

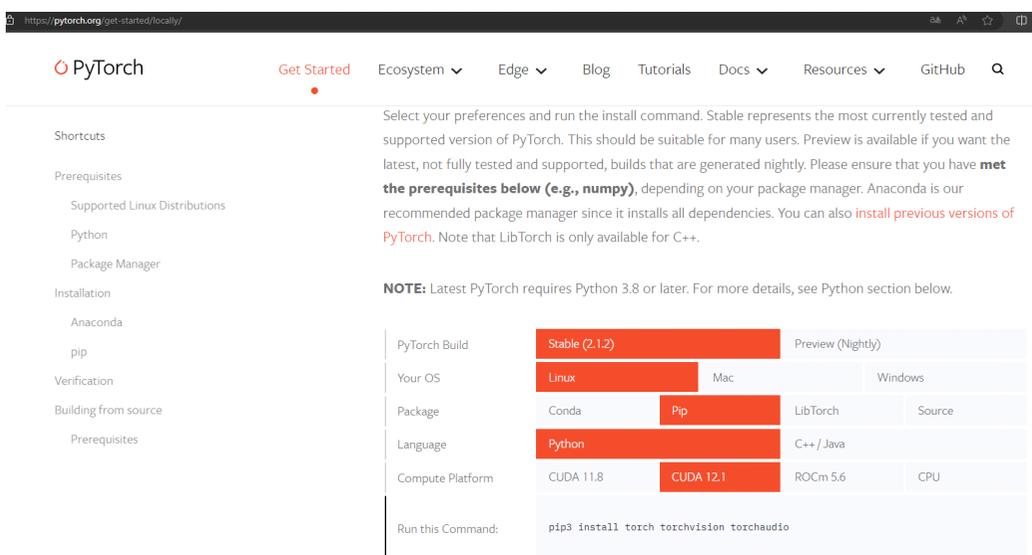
## 7.6 PyTorch 설치하기

1단계로 파이토치 공식 홈페이지에 접속한다.

<https://pytorch.org/get-started/locally>

그리고 Your OS는 Linux, Package는 Pip, Language는 Python, Compute Platform은 CUDA 12.1을 선택한다.

Run this Command의 내용을 복사한다.



The screenshot shows the PyTorch website's 'Get Started' page. It features a navigation menu with 'Get Started' highlighted. The main content area includes a sidebar with links like 'Shortcuts', 'Prerequisites', and 'Installation'. The main text explains how to select preferences and run the install command. A table below lists various options for PyTorch Build, Your OS, Package, Language, and Compute Platform. The 'Run this Command' section displays the command: `pip3 install torch torchvision torchaudio`.

PyTorch Build	Stable (2.1.2)	Preview (Nightly)
Your OS	Linux	Mac, Windows
Package	Conda, Pip	LibTorch, Source
Language	Python	C++ / Java
Compute Platform	CUDA 11.8, CUDA 12.1	ROCm 5.6, CPU

Run this Command: `pip3 install torch torchvision torchaudio`

[그림 7.14] PyTorch 공식 사이트(PyTorch 2.12, 최신 기준: 2024-01-27)

2단계로 pip 명령어를 입력한다.

```
pip3 install torch torchvision torchaudio
```

```

* [새로운 브랜치] master -> origin/master
'master' 브랜치가 리모트의 'master' 브랜치를 ('origin'에서) 따라가도록 설정되었습니다.
이미 'master'에 있습니다
/home/dyj

Looking for an existing zsh config...
Using the Oh My Zsh template file and adding it to /home/dyj/.zshrc.

oh my zsh
...is now installed!

Before you scream Oh My Zsh! look over the '.zshrc' file to select plugins, themes, and options.

• Follow us on Twitter: https://twitter.com/ohmyzsh
• Join our Discord community: https://discord.gg/ohmyzsh
• Get stickers, t-shirts, coffee mugs and more: https://shop.planetargon.com/collections/oh-my-zsh

+ ~ pip3 install torch torchvision torchaudio
zsh: command not found: pip3
+ ~ pip
zsh: command not found: pip
+ ~ vim .bashrc
+ ~ vim .zshrc
+ ~ source .zshrc
+ ~ pip3 install torch torchvision torchaudio

```

[그림 7.15] WSL2에서 PyTorch 2.12 설치하기(1)

3단계로 설치를 기다린다. 인터넷 속도에 따라 오래 걸릴 수도 있다. 다운로드 용량은 대략 3~4GB 이상을 요구한다.

```

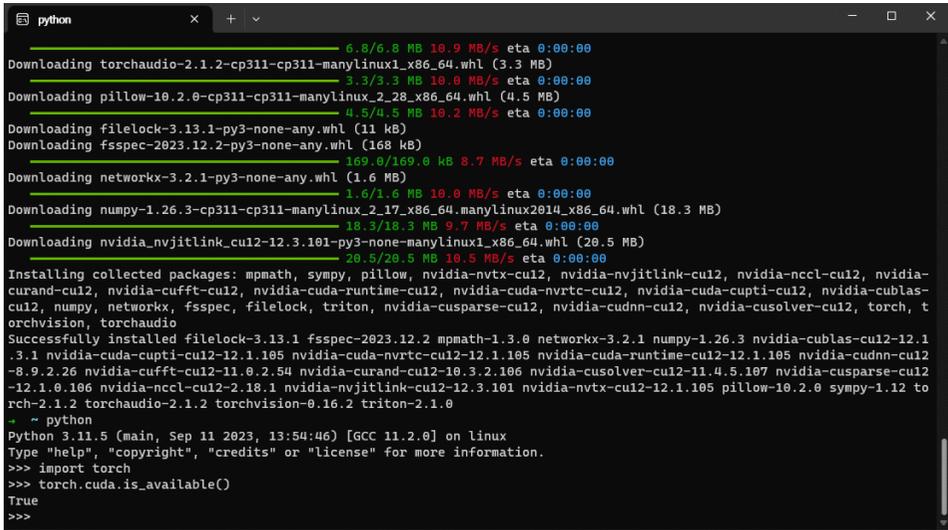
pip3
Downloading torch-2.1.2-cp311-cp311-manylinux1_x86_64.whl.metadata (25 kB)
Collecting torchvision
  Downloading torchvision-0.16.2-cp311-cp311-manylinux1_x86_64.whl.metadata (6.6 kB)
Collecting torchaudio
  Downloading torchaudio-2.1.2-cp311-cp311-manylinux1_x86_64.whl.metadata (6.4 kB)
Collecting filelock (from torch)
  Downloading filelock-3.13.1-py3-none-any.whl.metadata (2.8 kB)
Requirement already satisfied: typing-extensions in ./miniconda3/lib/python3.11/site-packages (from torch) (4.9.0)
Collecting sympy (from torch)
  Downloading sympy-1.12-py3-none-any.whl (5.7 MB)
 5.7/5.7 MB 10.7 MB/s eta 0:00:00
Collecting networkx (from torch)
  Downloading networkx-3.2.1-py3-none-any.whl.metadata (5.2 kB)
Requirement already satisfied: Jinja2 in ./miniconda3/lib/python3.11/site-packages (from torch) (3.1.2)
Collecting fsspec (from torch)
  Downloading fsspec-2023.12.2-py3-none-any.whl.metadata (6.8 kB)
Collecting nvidia-cuda-nvrtc-cu12==12.1.105 (from torch)
  Downloading nvidia_cuda_nvrtc_cu12-12.1.105-py3-none-manylinux1_x86_64.whl (23.7 MB)
 23.7/23.7 MB 9.4 MB/s eta 0:00:00
Collecting nvidia-cuda-runtime-cu12==12.1.105 (from torch)
  Downloading nvidia_cuda_runtime_cu12-12.1.105-py3-none-manylinux1_x86_64.whl (823 kB)
 823.6/823.6 kB 10.3 MB/s eta 0:00:00
Collecting nvidia-cuda-cupti-cu12==12.1.105 (from torch)
  Downloading nvidia_cuda_cupti_cu12-12.1.105-py3-none-manylinux1_x86_64.whl (14.1 MB)
 14.1/14.1 MB 10.5 MB/s eta 0:00:00
Collecting nvidia-cudnn-cu12==8.9.2.26 (from torch)
  Downloading nvidia_cudnn_cu12-8.9.2.26-py3-none-manylinux1_x86_64.whl.metadata (1.6 kB)
Collecting nvidia-cublas-cu12==12.1.3.1 (from torch)
  Downloading nvidia_cublas_cu12-12.1.3.1-py3-none-manylinux1_x86_64.whl (410.6 MB)
 175.2/410.6 MB 11.8 MB/s eta 0:00:20

```

[그림 7.16] WSL2에서 PyTorch 2.12 설치하기(2)

4단계로 python 명령어를 실행한다.

```
$ python
>>> import torch
>>> torch.cuda.is_available()
```



[그림 7.17] WSL2에서 Python 실행으로 PyTorch Cuda 사용여부 확인 결과값이 True가 나오면, Cuda가 지원되는 것이다.

## 7.7 PyCharm Community PATH 추가 반영하기

다음 명령어로 진행할 수 있다.

cd
vim .zshrc
맨 마지막에 한 줄 추가
<code>export PATH="\$HOME/pycharm-community-2023.3.3/bin:\$PATH"</code>
vim .bashrc
맨 마지막에 한 줄 추가
<code>export PATH="\$HOME/pycharm-community-2023.3.3/bin:\$PATH"</code>
source .zshrc
source .bashrc
zsh
bash
zsh

```

dyj@dyj-workstation:~$ source .bashrc
dyj@dyj-workstation:~$ ./pycharm.sh
bash: ./pycharm.sh: 그런 파일이나 디렉터리가 없습니다
dyj@dyj-workstation:~$ pycharm.sh
CompileCommand: exclude com/intellij/openapi/vfs/impl/FilePartNodeRoot.trieDescend bool exclude = true

Start Failed
Unable to detect graphics environment
dyj@dyj-workstation:~$ sh pycharm.sh
sh: 0: cannot open pycharm.sh: No such file
dyj@dyj-workstation:~$ ls
Desktop Music PycharmProjects miniconda3 thinclient_drives
Documents Pictures Templates pycharm-community-2023.3.3
Downloads Public Videos python_lab
dyj@dyj-workstation:~$ zsh
~$ ls
Desktop Music PycharmProjects miniconda3 thinclient_drives
Documents Pictures Templates pycharm-community-2023.3.3
Downloads Public Videos python_lab
~$ vim .zshrc
~$ source .zshrc
~$ vim .bashrc
~$ vim .zshrc
~$ |

```

[그림 7.18] WSL2에서 PyCharm Community 경로 수정하기(1)

```
vim
# ~/.bash_aliases, instead of adding them here directly.
# See /usr/share/doc/bash-doc/examples in the bash-doc package.

if [ -f ~/.bash_aliases ]; then
  . ~/.bash_aliases
fi

# enable programmable completion features (you don't need to enable
# this, if it's already enabled in /etc/bash.bashrc and /etc/profile
# sources /etc/bash.bashrc).
if ! shopt -oq posix; then
  if [ -f /usr/share/bash-completion/bash_completion ]; then
    . /usr/share/bash-completion/bash_completion
  elif [ -f /etc/bash_completion ]; then
    . /etc/bash_completion
  fi
fi

export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/usr/local/cuda-12.1/lib64/
export JAVA_HOME=/usr/local/jdk-21
export PATH=$JAVA_HOME/bin:$PATH
export PATH="$HOME/miniconda3/bin:$PATH"
export PATH="$HOME/pycharm-community-2023.3.3/bin:$PATH"
".bashrc" 123L, 4006B 122,40 바닥
```

[그림 7.19] WSL2에서 PyCharm Community 경로 수정하기(2)

```
vim
# export EDITOR='vim'
# else
# export EDITOR='mvim'
# fi

# Compilation flags
# export ARCHFLAGS="-arch x86_64"

# Set personal aliases, overriding those provided by oh-my-zsh libs,
# plugins, and themes. Aliases can be placed here, though oh-my-zsh
# users are encouraged to define aliases within the ZSH_CUSTOM folder.
# For a full list of active aliases, run `alias`.
#
# Example aliases
# alias zshconfig="mate ~/.zshrc"
# alias ohmyzsh="mate ~/.oh-my-zsh"
#

export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/usr/local/cuda-12.1/lib64/
export JAVA_HOME=/usr/local/jdk-21
export PATH=$JAVA_HOME/bin:$PATH
export PATH="$HOME/miniconda3/bin:$PATH"
export PATH="$HOME/pycharm-community-2023.3.3/bin:$PATH"
".zshrc" 108L, 4103B 108,40 바닥
```

[그림 7.20] WSL2에서 PyCharm Community 경로 수정하기(3)

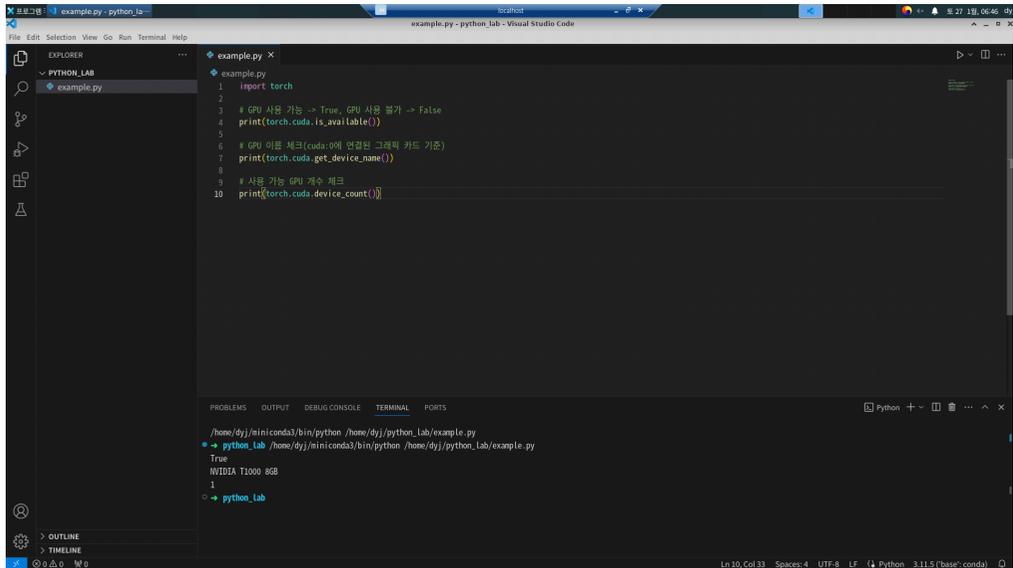
```
dyj@dyj-workstation: ~$ ls
Desktop Music PycharmProjects miniconda3 thinclient_drives
Documents Pictures Templates pycharm-community-2023.3.3
Downloads Public Videos python_lab
dyj@dyj-workstation:~$ zsh
~$ ls
Desktop Music PycharmProjects miniconda3 thinclient_drives
Documents Pictures Templates pycharm-community-2023.3.3
Downloads Public Videos python_lab
~$ vim .zshrc
~$ source .zshrc
~$ vim .bashrc
~$ vim .zshrc
~$ source .zshrc
~$ source .bashrc
.bashrc:16: command not found: shopt
.bashrc:24: command not found: shopt
.bashrc:111: command not found: shopt
/usr/share/bash-completion/bash_completion:45: command not found: shopt
/usr/share/bash-completion/bash_completion:1596: parse error near `|'
\[\e]0;\u@h: \w\|a\|[\033[01;32m\|u@\h\[\033[00m\]:\[\033[01;34m\|w\[\033[00m\]$ bash
dyj@dyj-workstation:~$ zsh
~$ bash
dyj@dyj-workstation:~$ |
```

[그림 7.21] WSL2에서 PyCharm Community 경로 수정하기(4)

## 7.8 Visual Studio Code에서 torch 사용 확인하기

Visual Studio Code에서 Python 파일을 생성한 후 아래의 코드를 입력하여 PyTorch 지원 여부를 확인할 수 있다.

```
import torch
print(torch.cuda.is_available())
print(torch.cuda.get_device_name())
print(torch.cuda.device_count())
```



The screenshot shows the Visual Studio Code interface. The Explorer view on the left shows a project named 'PYTHON\_LAB' with a file 'example.py'. The Editor view shows the following code in 'example.py':

```
1 import torch
2
3 # GPU 사용 가능 -> True, GPU 사용 불가 -> False
4 print(torch.cuda.is_available())
5
6 # GPU 이름 체크(cuda에 연결된 그래픽 카드 기준)
7 print(torch.cuda.get_device_name())
8
9 # 사용 가능 GPU 개수 체크
10 print(torch.cuda.device_count())
```

The Terminal view at the bottom shows the execution output:

```
python /home/dyjin/miniconda3/bin/python /home/dyjin/python/python_lab/example.py
python /home/dyjin/miniconda3/bin/python /home/dyjin/python/python_lab/example.py
True
NVIDIA T1000 8GB
1
python_Lab
```

[그림 7.22] xrdp에서 visual studio code의 PyTorch에서의 cuda 사용