

Ubuntu 환경에 Java 프로젝트 개발 및 배포를 위하여 Java를 설치하고, 환경변수를 등록해보자.

버전 정보

OS : Ubuntu 22.04.3 LTS

Java : openjdk-11

설치

Java 다운로드

ubuntu 패키지 업데이트

업데이트

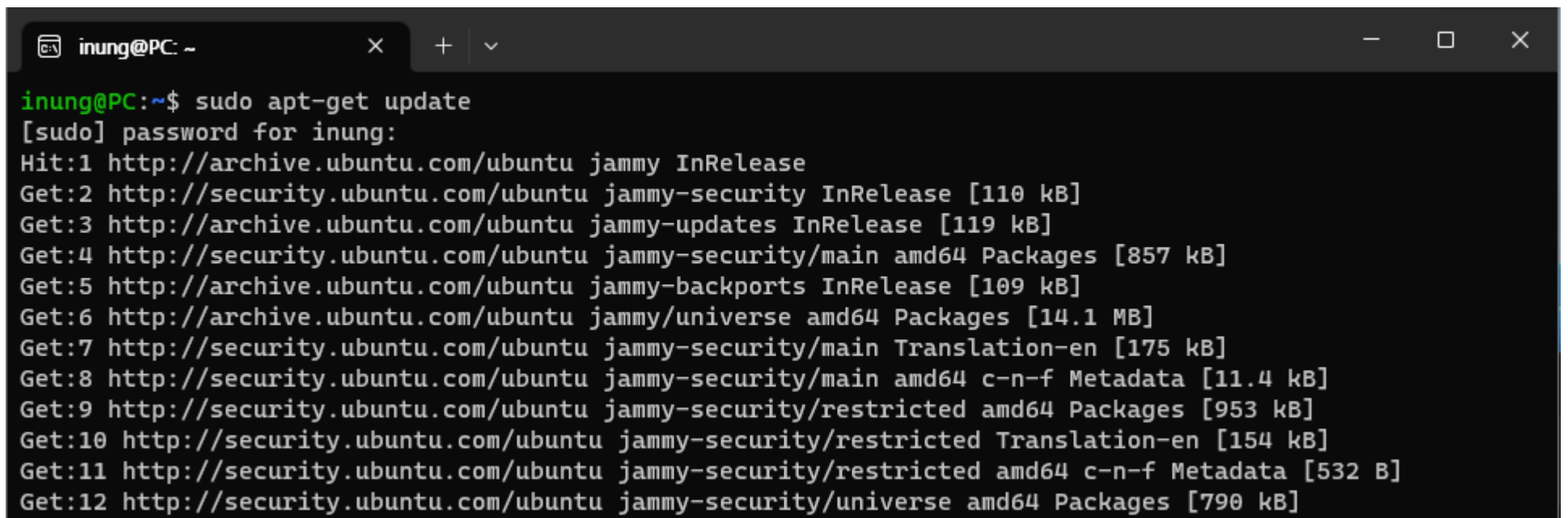
```
sudo apt update
```

업그레이드

```
sudo apt upgrade
```

여기서 잠시 궁금했던 점을 해결하고 넘어가려 한다. 우분투를 사용하면서 들었던 의문점은 update 와 upgrade 의 차이점이었다.

- update : 설치 가능한 패키지 리스트를 최신화

A terminal window titled 'inung@PC: ~' showing the output of the 'sudo apt-get update' command. The output lists various Ubuntu repositories and the packages being updated, including InRelease files and amd64 Packages for security, updates, backports, and universe repositories. The total size of the updates is 14.1 MB.

```
inung@PC:~$ sudo apt-get update
[sudo] password for inung:
Hit:1 http://archive.ubuntu.com/ubuntu jammy InRelease
Get:2 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
Get:3 http://archive.ubuntu.com/ubuntu jammy-updates InRelease [119 kB]
Get:4 http://security.ubuntu.com/ubuntu jammy-security/main amd64 Packages [857 kB]
Get:5 http://archive.ubuntu.com/ubuntu jammy-backports InRelease [109 kB]
Get:6 http://archive.ubuntu.com/ubuntu jammy/universe amd64 Packages [14.1 MB]
Get:7 http://security.ubuntu.com/ubuntu jammy-security/main Translation-en [175 kB]
Get:8 http://security.ubuntu.com/ubuntu jammy-security/main amd64 c-n-f Metadata [11.4 kB]
Get:9 http://security.ubuntu.com/ubuntu jammy-security/restricted amd64 Packages [953 kB]
Get:10 http://security.ubuntu.com/ubuntu jammy-security/restricted Translation-en [154 kB]
Get:11 http://security.ubuntu.com/ubuntu jammy-security/restricted amd64 c-n-f Metadata [532 B]
Get:12 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 Packages [790 kB]
```

- upgrade : 실제 업데이트

```
inung@PC:~$ sudo apt upgrade
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Calculating upgrade... Done
The following packages will be upgraded:
  apport apt apt-utils base-files bind9-dnsutils bind9-host bind9-libs binutils binutils-common
  binutils-x86-64-linux-gnu ca-certificates curl distro-info distro-info-data dpkg file gawk
  gcc-12-base git git-man iptables irqbalance libapt-pkg6.0 libbinutils libc-bin libc6 libcap2
  libcap2-bin libctf-nobfd0 libctf0 libcurl3-gnutls libcurl4 libgcc-s1 libglib2.0-0 libglib2.0-bin
  libglib2.0-data libgssapi-krb5-2 libip4tc2 libip6tc2 libjson-c5 libk5crypto3 libkrb5-3
  libkrb5support0 libldap-2.5-0 libldap-common libmagic-mgc libmagic1 libncurses6 libncursesw6
  libnetplan0 libnss-systemd libpam-cap libpam-systemd libperl5.34 libpython3.10
  libpython3.10-minimal libpython3.10-stdlib libssh-4 libssl3 libstdc++6 libsystemd0 libtinfo6
  libudev1 libunwind8 libx11-6 libx11-data libxtables12 locales motd-news-config ncurses-base
  ncurses-bin netplan.io openssh-client openssl perl perl-base perl-modules-5.34 python-apt-common
  python3-apport python3-apt python3-distro-info python3-distupgrade python3-gdbm
  python3-problem-report python3-software-properties python3.10 python3.10-minimal snapd
  software-properties-common systemd systemd-sysv systemd-timesyncd tzdata ubuntu-advantage-tools
  ubuntu-minimal ubuntu-release-upgrader-core ubuntu-standard ubuntu-wsl udev ufw vim vim-common
  vim-runtime vim-tiny xxd
```

update 는 말 그대로 설치가 가능한 패키지 리스트만을 최신화 하는 것이다. 즉, 실제 버전을 업데이트하는 것은 아니다. update 를 통해 패키지 목록을 최신화하고, 현재 upgrade 가 필요한 목록을 조회한 뒤 upgrade 를 통해서 패키지를 최신 버전으로 업그레이드 하는 것이다.

순서를 명령어를 통해 알아보면 다음과 같다.

- sudo apt update : 패키지 목록 최신화
- sudo apt list --upgradable : 업그레이드 가능한 목록 확인
- sudo apt upgrade : 패키지 업그레이드

이제 update 와 upgrade 의 차이점을 알았으니 다음으로 넘어가자.

openjdk 설치

먼저 openjdk를 설치하기 위해 어떤 버전의 openjdk가 있는지 확인해보자.

```
# 모든 버전 목록 조회
sudo apt list openjdk*

# 특정 버전 목록 조회
sudo apt list openjdk-[버전]*
```

위 명령어를 입력하면 아래 이미지와 같이 openjdk의 목록이 출력된다.

```
inung@PC:~$ apt list openjdk-11*
Listing... Done
openjdk-11-dbg/jammy-updates,jammy-security 11.0.20.1+1-0ubuntu1~22.04 amd64
openjdk-11-demo/jammy-updates,jammy-security 11.0.20.1+1-0ubuntu1~22.04 amd64
openjdk-11-doc/jammy-updates,jammy-security 11.0.20.1+1-0ubuntu1~22.04 all
openjdk-11-jdk-headless/jammy-updates,jammy-security 11.0.20.1+1-0ubuntu1~22.04 amd64
openjdk-11-jdk/jammy-updates,jammy-security 11.0.20.1+1-0ubuntu1~22.04 amd64
openjdk-11-jre-dcevm/jammy 11.0.12+7-1 amd64
openjdk-11-jre-headless/jammy-updates,jammy-security 11.0.20.1+1-0ubuntu1~22.04 amd64
openjdk-11-jre-zero/jammy-updates,jammy-security 11.0.20.1+1-0ubuntu1~22.04 amd64
openjdk-11-jre/jammy-updates,jammy-security 11.0.20.1+1-0ubuntu1~22.04 amd64
openjdk-11-source/jammy-updates,jammy-security 11.0.20.1+1-0ubuntu1~22.04 all
inung@PC:~$ |
```

이제 본인이 설치하고 싶은 openjdk를 선택하여 설치 명령어를 입력하면된다.

```
# 설치 명령어(11버전)
sudo apt install openjdk-11-jdk
```

```
inung@PC:~$ sudo apt install openjdk-11-jdk
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  alsa-topology-conf alsa-ucm-conf at-spi2-core ca-certificates-java dconf-gsettings-backend
  dconf-service fontconfig-config fonts-dejavu-core fonts-dejavu-extra gsettings-desktop-schemas
  java-common libasound2 libasound2-data libatk-bridge2.0-0 libatk-wrapper-java
  libatk-wrapper-java-jni libatk1.0-0 libatk1.0-data libatspi2.0-0 libavahi-client3
  libavahi-common-data libavahi-common3 libcups2 libdconf1 libdrm-amdgpu1 libdrm-intel1
  libdrm-nouveau2 libdrm-radeon1 libfontconfig1 libfontenc1 libfreetype6 libglib2.0-0 libglib2.0-data
```

설치 확인

모든 설치가 완료되었으니 제대로 설치가 된 것인지 확인해보자.

```
java --version
```

위 명령어를 입력하면 아래 이미지와 같이 버전이 출력될 것이다.

```
inung@PC:~$ java --version
openjdk 11.0.20.1 2023-08-24
OpenJDK Runtime Environment (build 11.0.20.1+1-post-Ubuntu-0ubuntu122.04)
OpenJDK 64-Bit Server VM (build 11.0.20.1+1-post-Ubuntu-0ubuntu122.04, mixed mode, sharing)
inung@PC:~$ |
```

Java의 설치가 완료되었다.

환경변수 설정

이제 Java의 설치를 완료했으니 환경변수 설정에 대해서 알아보자.

설정 이유

환경변수를 설정하기 전에 왜 해야하는지 알아보자. 기본적으로 Java를 실행하거나 참조하기 위해서는 Java가 설치된 디렉토리로 이동해서 동작을 호출해야한다. 이렇게되면 매번 사용할 때 마다 디렉토리 위치를 이동시켜서 Java가 존재하는 위치에서 호출해야하는 번거로움이 발생한다.

이러한 번거로움을 줄여주고 어떠한 위치에서라도 전역적으로 호출하여 사용 할 수 있도록하기 위해서 환경변수를 설정하는 것이다.

설정하기

먼저 환경변수를 설정하기 위해 설정 파일을 열어주자.

```
sudo vim /etc/profile
```

```
# /etc/profile: system-wide .profile file for the Bourne shell (sh(1))
# and Bourne compatible shells (bash(1), ksh(1), ash(1), ...).

if [ "${PS1}" ]; then
  if [ "${BASH}" ] && [ "$BASH" != "/bin/sh" ]; then
    # The file bash.bashrc already sets the default PS1.
    # PS1='\h:\w\$ '
    if [ -f /etc/bash.bashrc ]; then
      . /etc/bash.bashrc
    fi
  else
    if [ "$(id -u)" -eq 0 ]; then
      PS1='# '
    else
      PS1='$ '
    fi
  fi
fi

if [ -d /etc/profile.d ]; then
  for i in /etc/profile.d/*.sh; do
    if [ -r $i ]; then
      . $i
    fi
  done
unset i
fi
```

위와 같은 화면이 나오면 아래의 내용을 파일 하단에 입력하자.

```
export JAVA_HOME=/usr/lib/jvm/java-11-openjdk-amd64 # 본인의 Java 설치경로
export PATH=$JAVA_HOME/bin:$PATH
export CLASSPATH=$CLASSPATH:$JAVA_HOME/lib/ext:$JAVA_HOME/lib/tools.jar
```

이후 :wq! 를 입력하여 저장 후 종료해주고, 설정 파일의 변경사항을 저장해주자.

```
source /etc/profile
```

이렇게 입력하고 아무런 문구가 나오지 않으면 저장이 완료된 것이다. 이제 환경변수가 제대로 설정이 되었는지 호출하여 확인해보자.

```
echo $JAVA_HOME
```

결과 : /usr/lib/jvm/java-11-openjdk-amd64

위와 같이 확인된다면 환경변수 설정이 완료된 것이다.