

마이크로프로세서 및 응용설계 개발환경 구축 가이드

장 영준
uessie@naver.com

목 차

- 개발 환경 구축
 - VMware 환경 구축 (Linux, Ubuntu 12.04)
 - JDK, DevkitPro, Eclipse, Emulator, Git 설치
- 실습
 - 프로젝트 홈페이지 소개 및 활용 방법
 - Git을 사용하여 실습에 필요한 소스 코드 다운로드
 - 프로젝트 생성 및 빌드 (hello world 따라 하기 예제)
 - 다운로드 프로그램 사용
 - 실험장비(NDS)에 실행 이미지 다운로드 및 실행

개발 환경을 구축하는 우리의 자세

- 앞으로 설치하게 되는 프로그램들의 경로 및 설정들은

반드시 똑같이 따라하기

개발 환경 구축 목록

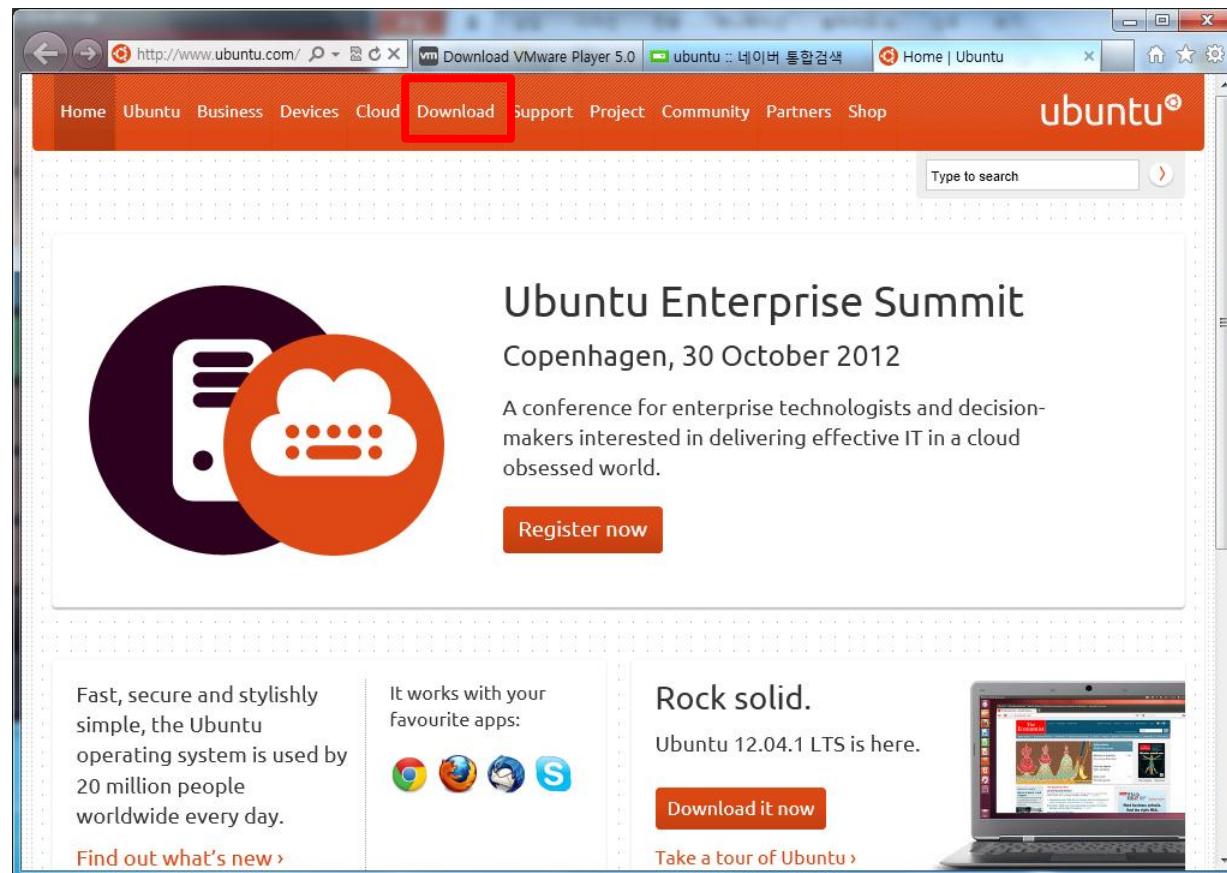
- **Virtual Box**
- **Linux (Ubuntu 12.04)**
- **JDK**
- **devkitPro**
- **Eclipse**
- **Emulator (Desmume)**
- **Git**

개발에 필요한
소프트웨어 및 라이브러리는
Home 디렉토리의
`ndedev`에 저장

ex) `/home/hansung/ndesdev`

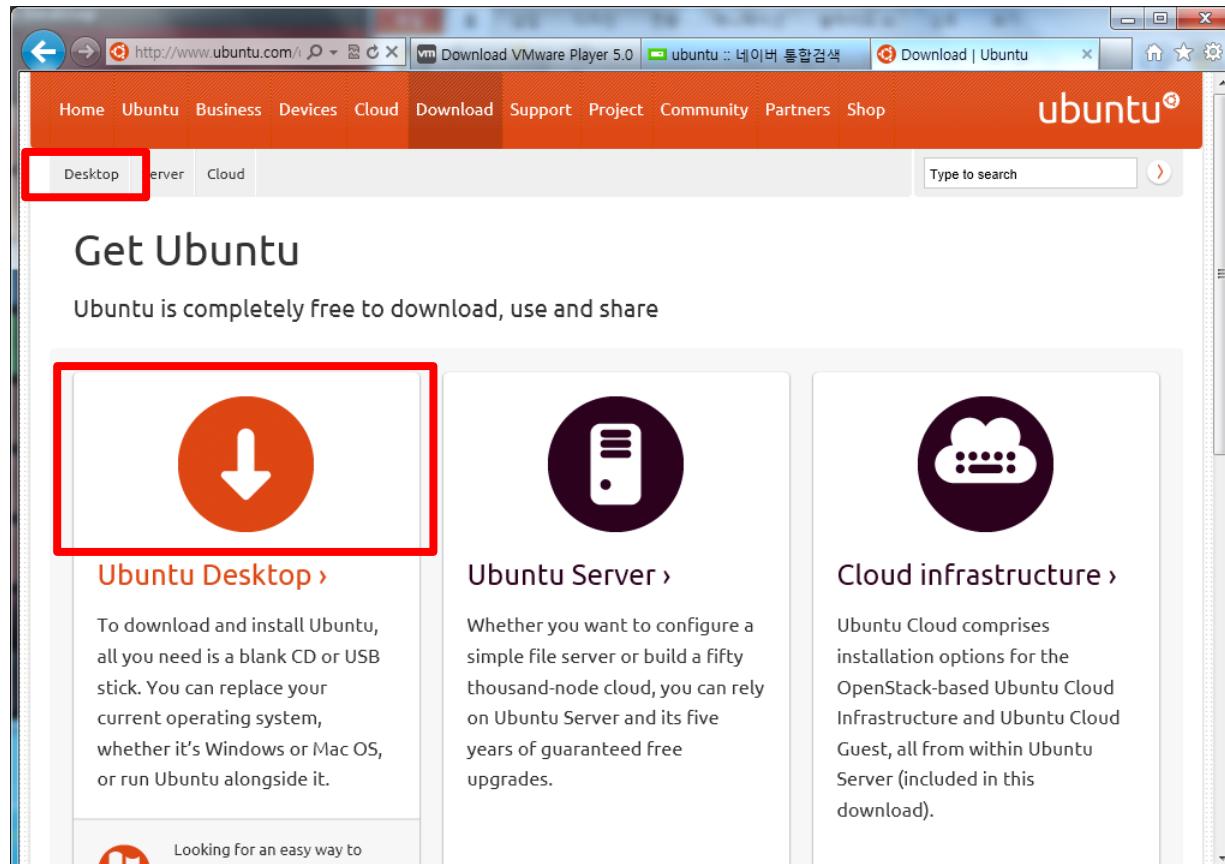
Linux(Ubuntu) 이미지 다운로드

- <http://www.ubuntu.com>에 접속하여, Download 메뉴를 선택



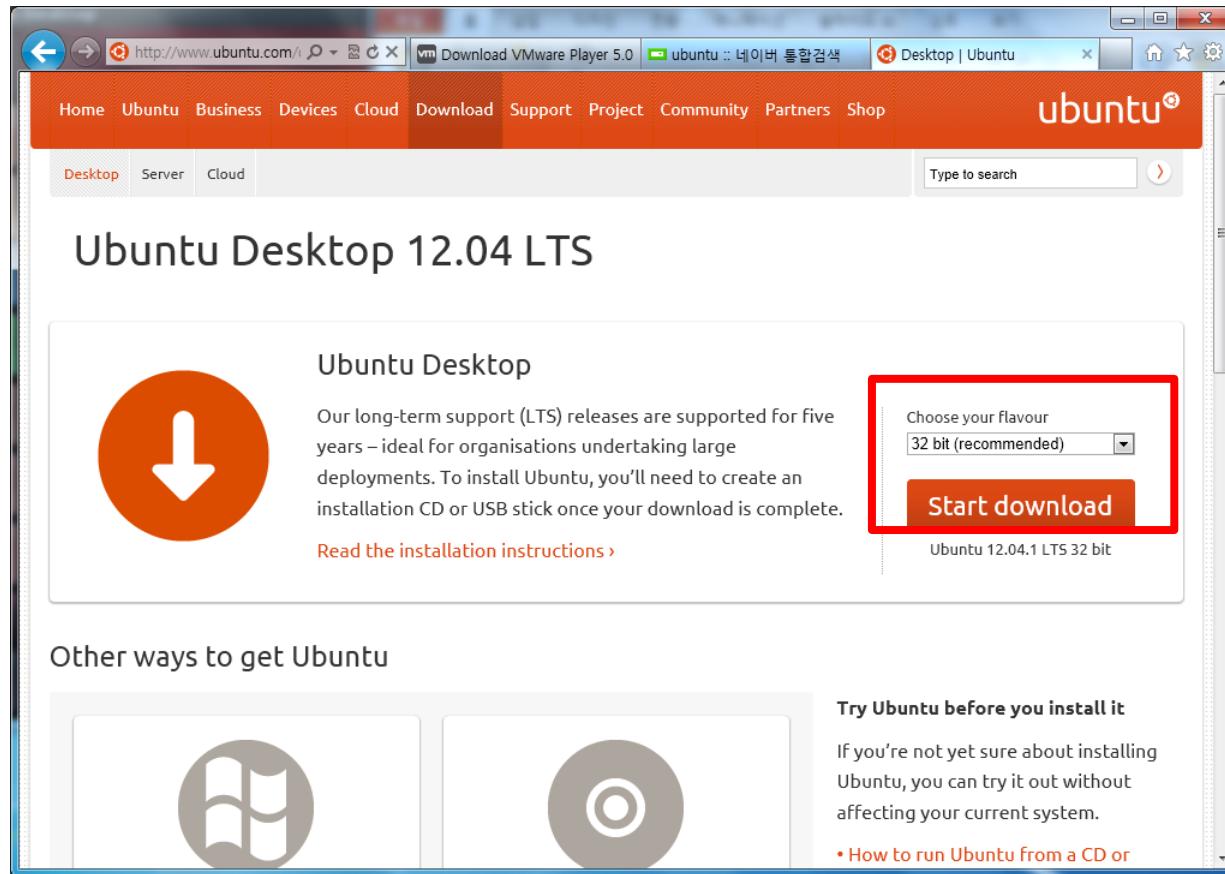
Linux(Ubuntu) 이미지 다운로드

- Desktop 메뉴 또는 아래의 아이콘을 클릭



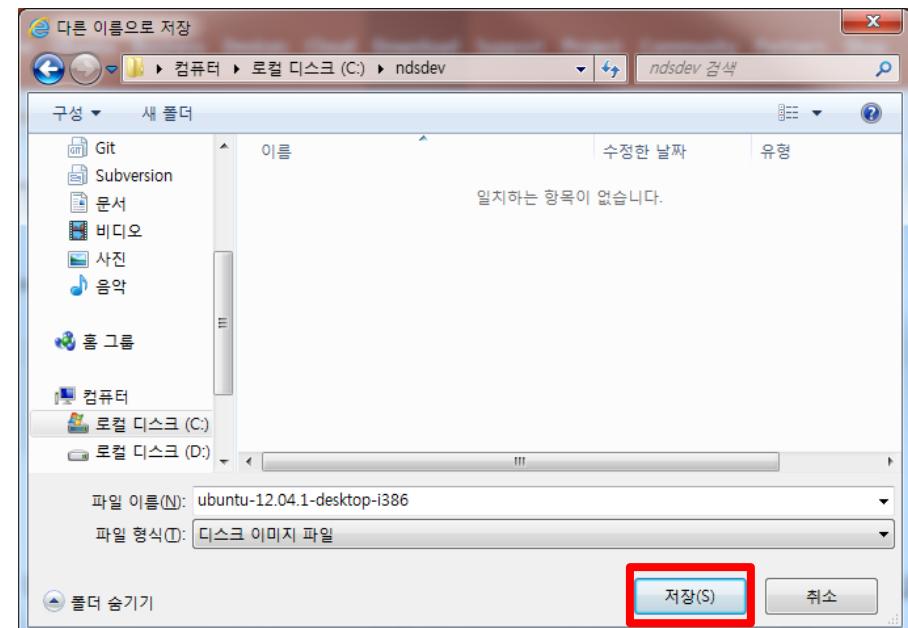
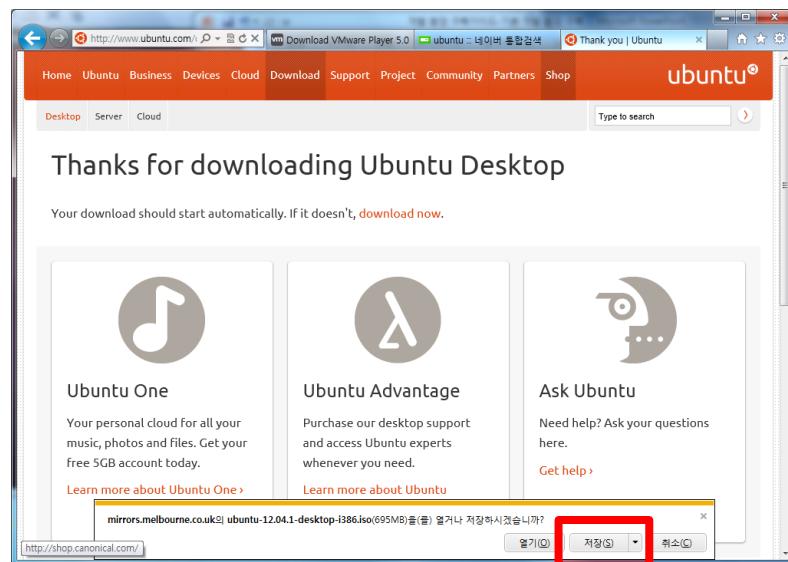
Linux(Ubuntu) 이미지 다운로드

- 다운로드 받을 Ubuntu 이미지를 자신의 환경에 맞게 선택한 뒤, Start download 버튼을 클릭



Linux(Ubuntu) 이미지 다운로드

- 다음과 같은 화면이 나오면 자동으로 다운로드가 시작됨
다운로드 창이 나타나면 적당한 위치에 다운로드



Virtual Box 설치

- <https://www.virtualbox.org>에 접속하여, Download 메뉴를 선택



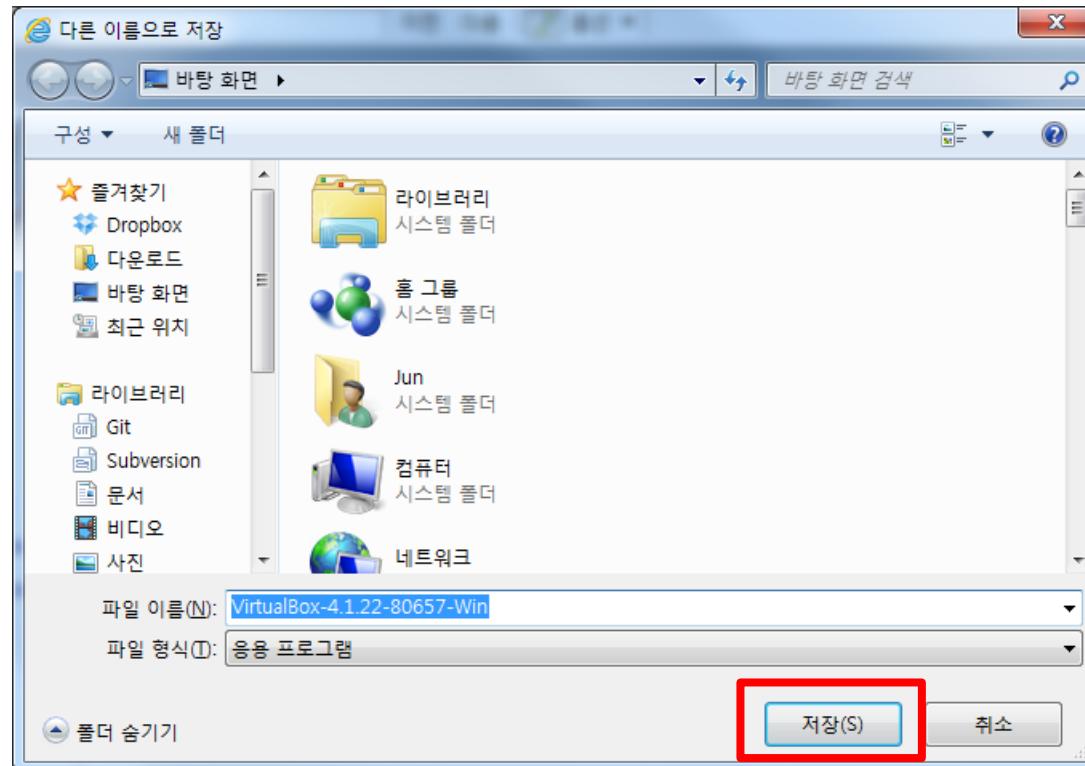
Virtual Box 설치

- VirtualBox 4.1.22 for Windows hosts 항목의 링크를 선택



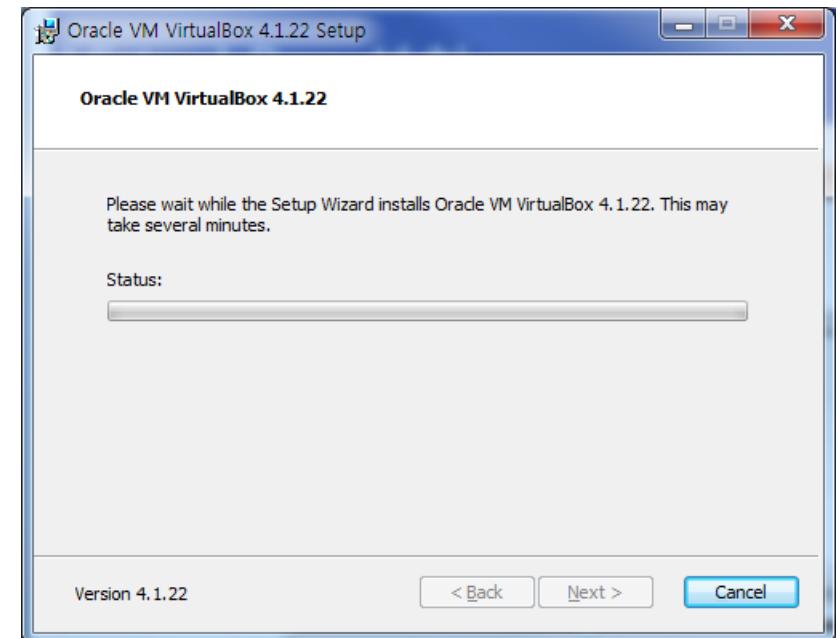
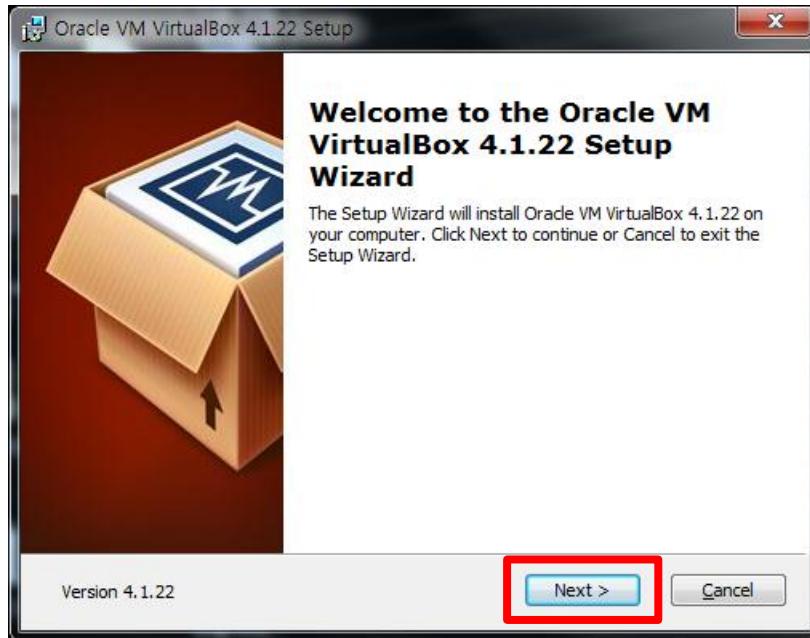
Virtual Box 설치

- 다운로드 창이 나타나면 적당한 위치에 다운로드



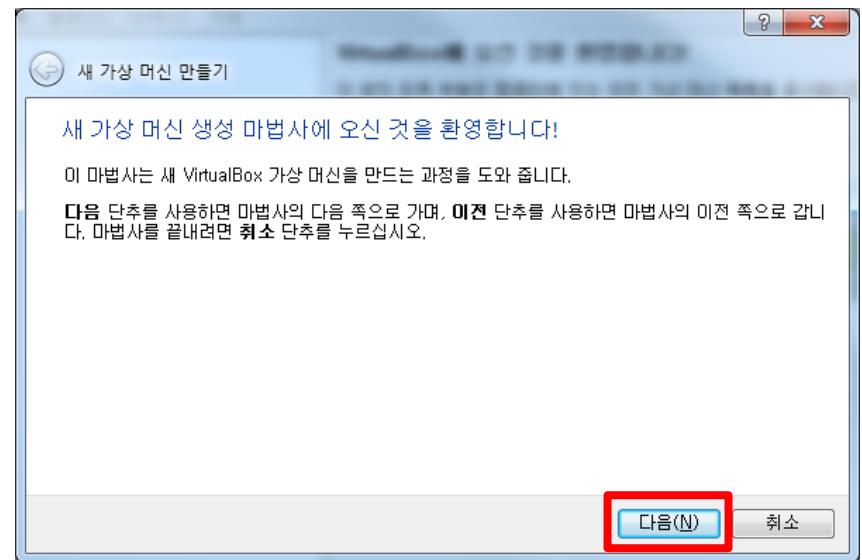
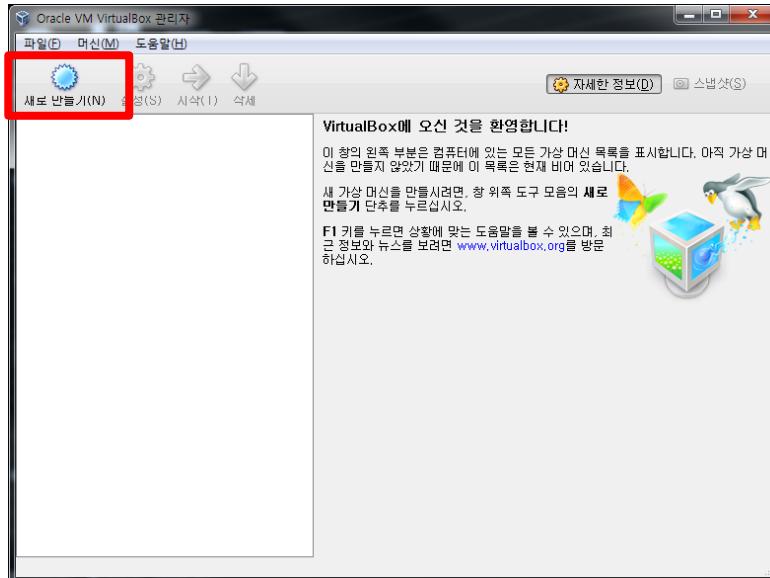
Virtual Box 설치

- 다운로드 받은 파일을 실행하여 Virtual Box를 설치
(Next등을 누르면 별 문제없이 설치 됨)



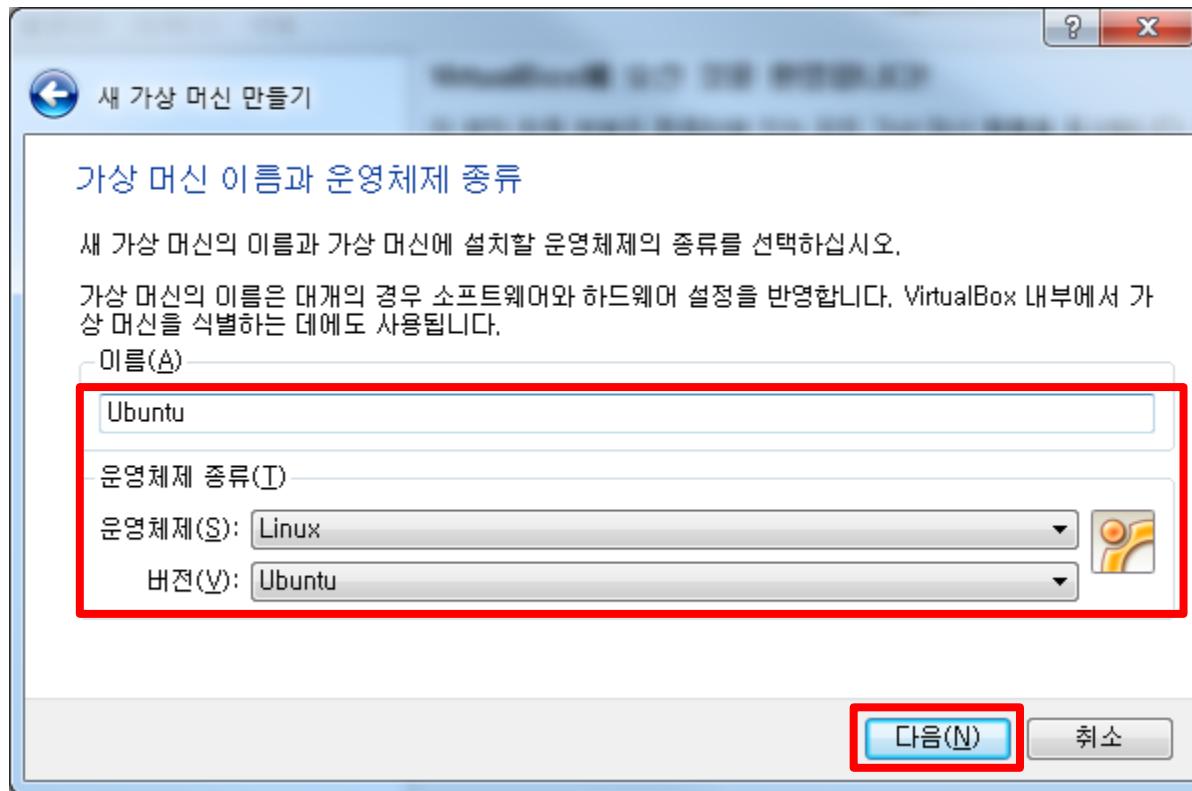
Linux(Ubuntu) 설치

- Virtual Box를 실행한 뒤, 새로 만들기 메뉴를 선택한 뒤 다음 버튼을 누름



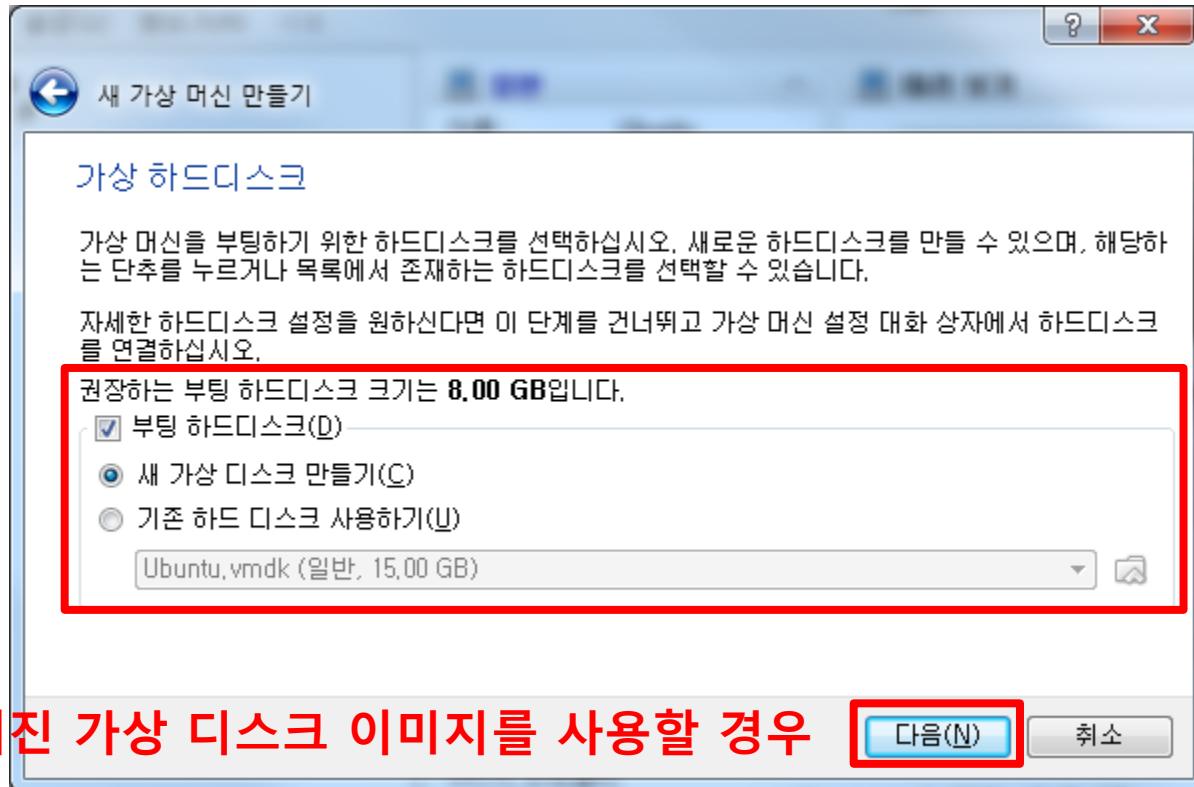
Linux(Ubuntu) 설치

- 다음과 같이 가상 머신의 이름을 입력하고, 운영체제를 Linux, Ubuntu로 선택한 뒤 다음 버튼을 누름



Linux(Ubuntu) 설치

- 다음과 같이 새 가상 디스크 만들기 항목을 선택하여 가상 머신 하드디스크를 설정한 뒤, 다음 버튼을 누름

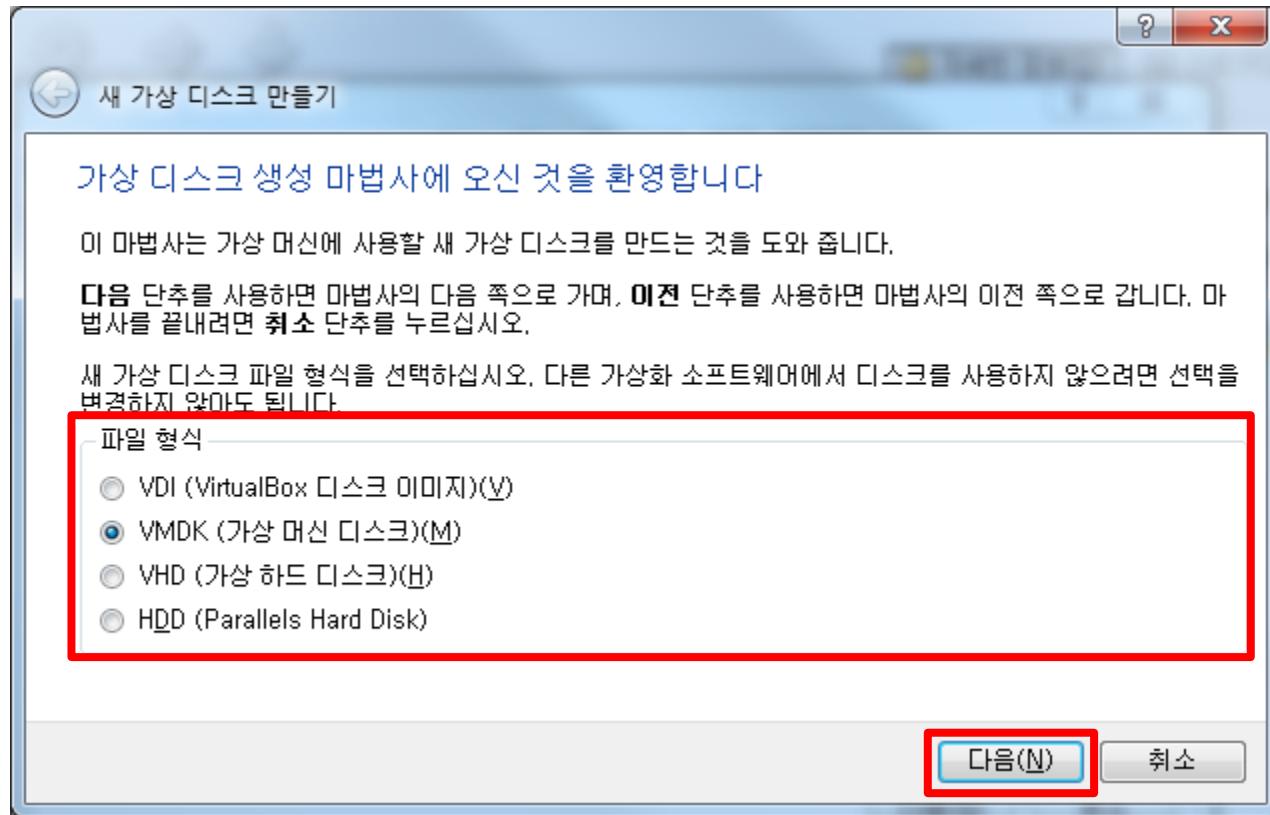


기존 하드 디스크 사용하기 메뉴를 통해 가상 하드디스크 만들기

앞으로 설치할 모든 소프트웨어 및 환경이 구성되어 있음 (50Page로 이동)

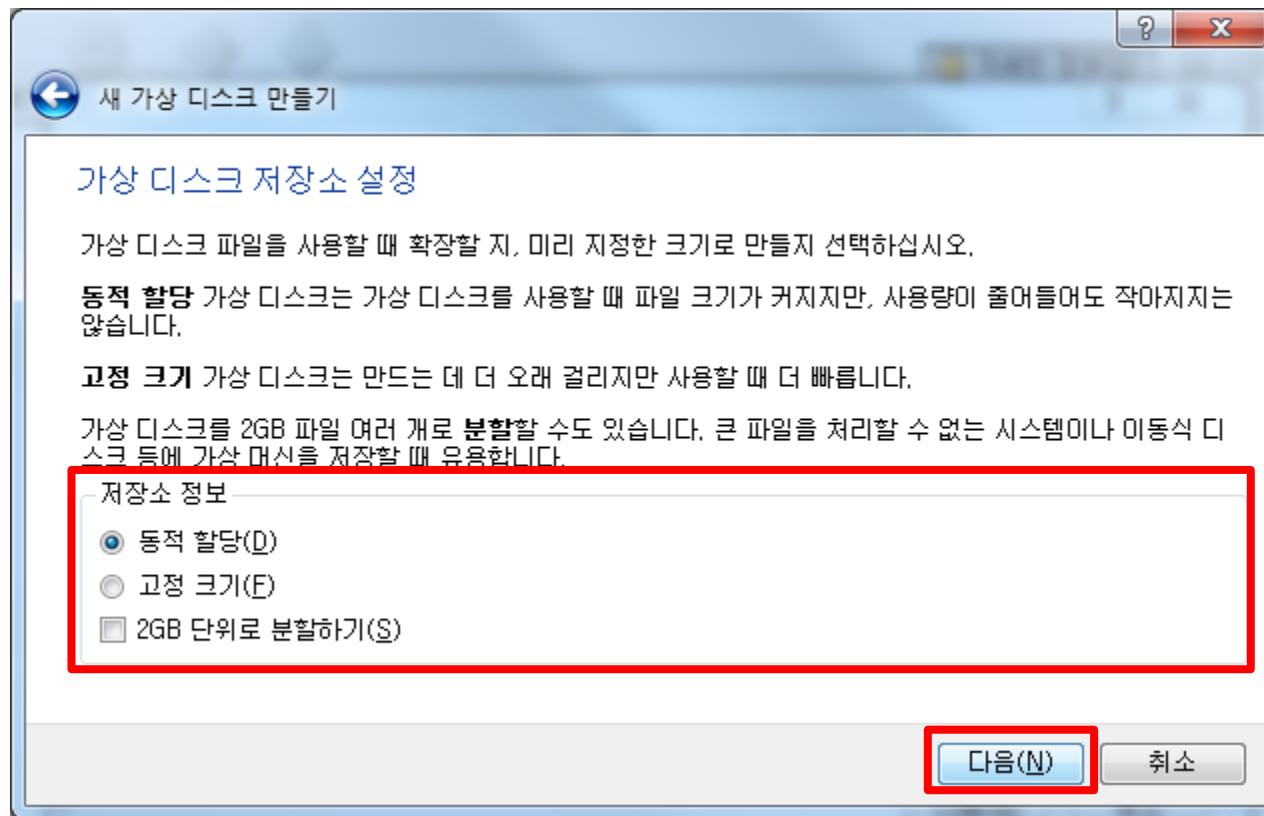
Linux(Ubuntu) 설치

- 가상 디스크의 형식을 선택한 뒤 다음 버튼을 누름
(여기에서는 VMware Workstation에서도 사용할 수 있도록 VMDK로 생성)



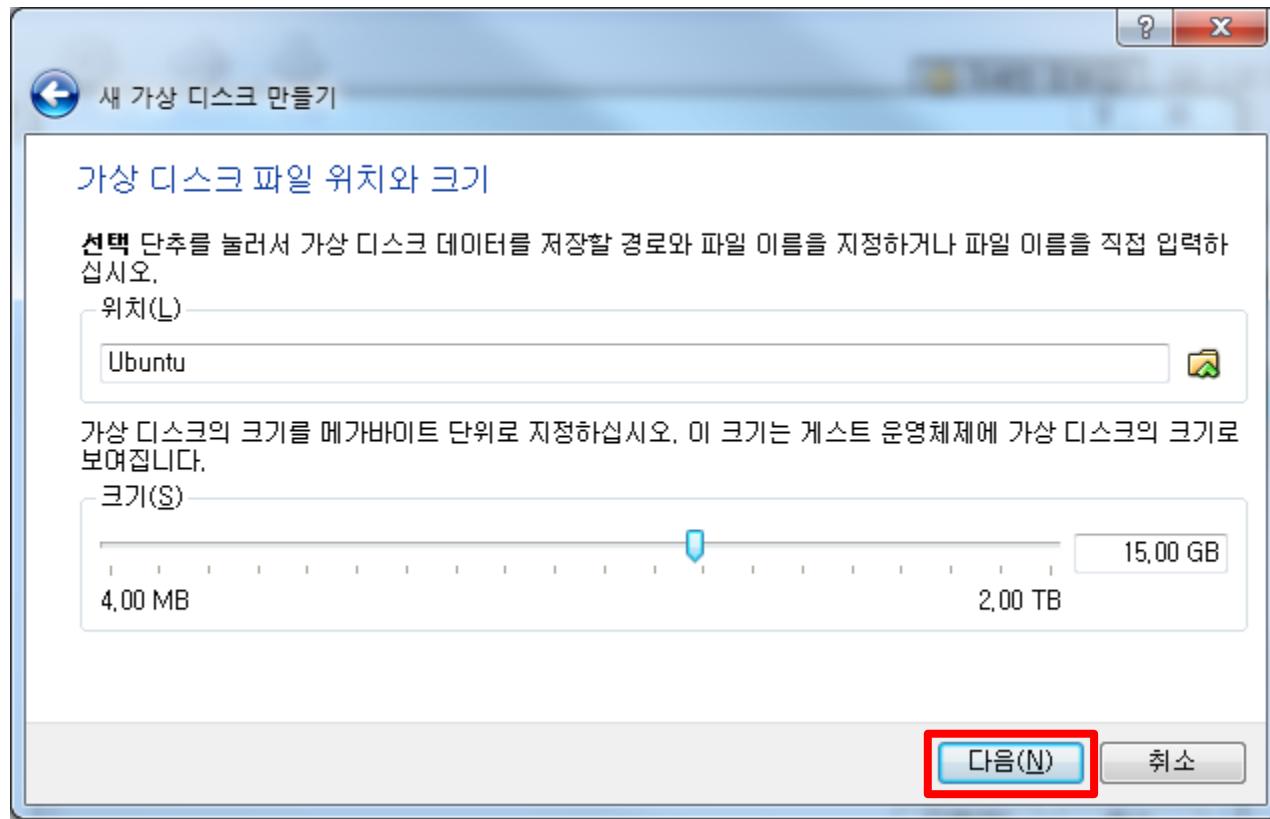
Linux(Ubuntu) 설치

- 가상 디스크 저장소를 설정한 뒤 다음 버튼을 누름



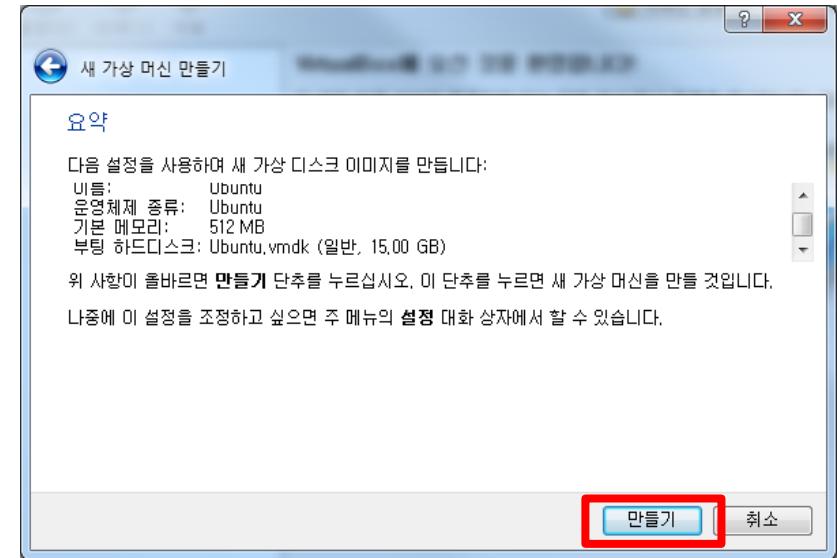
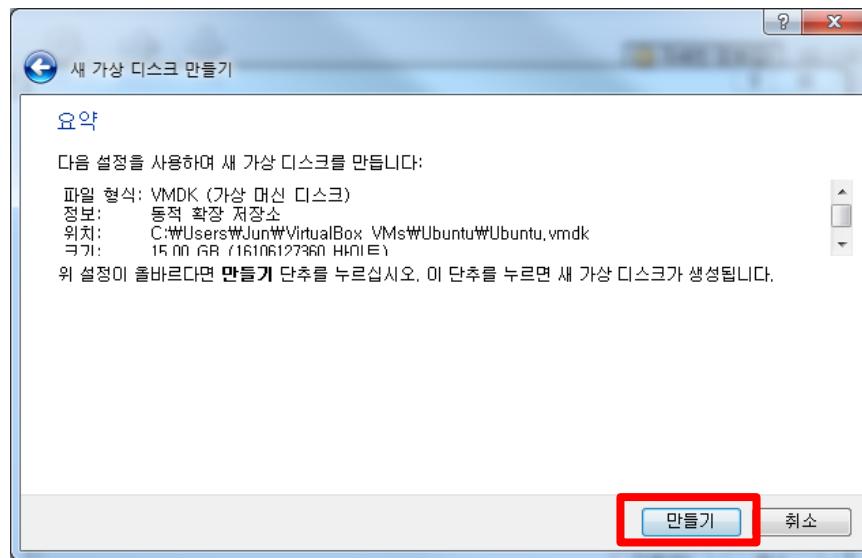
Linux(Ubuntu) 설치

- 가상 디스크 파일 위치와 크기를 설정한 뒤 다음 버튼을 누름



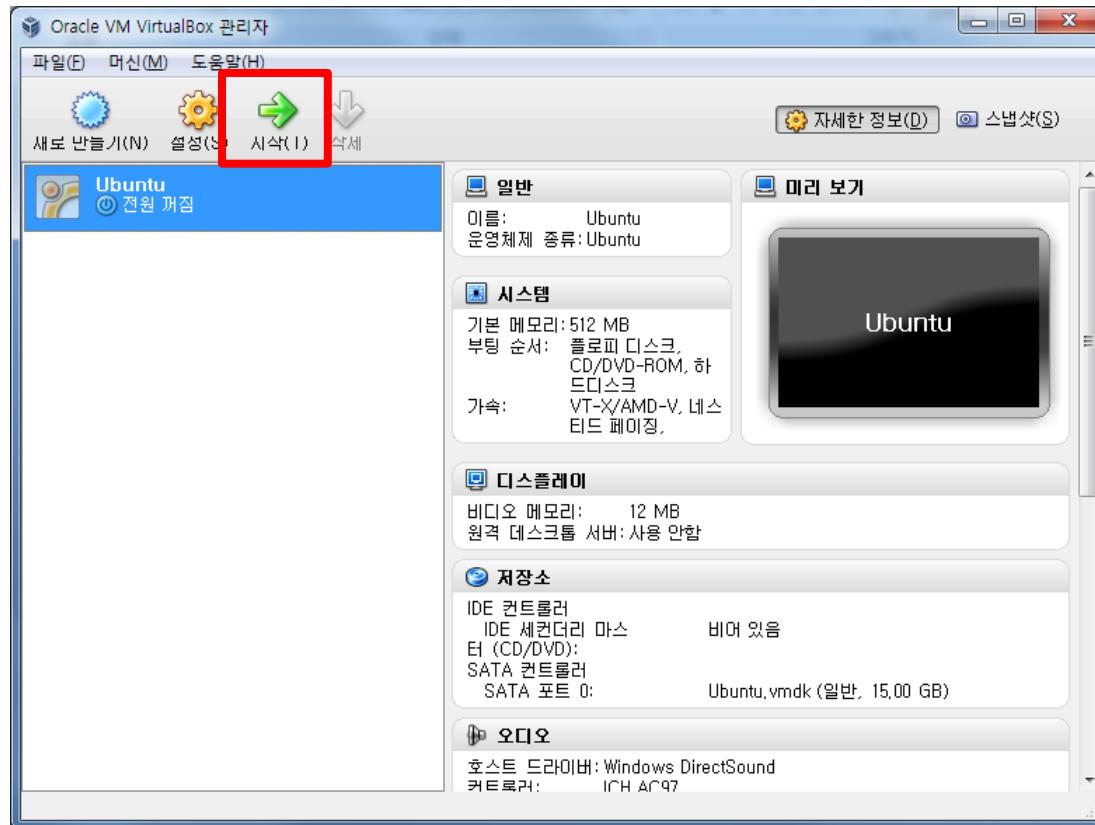
Linux(Ubuntu) 설치

- 만들기 버튼을 눌러 가상디스크 생성



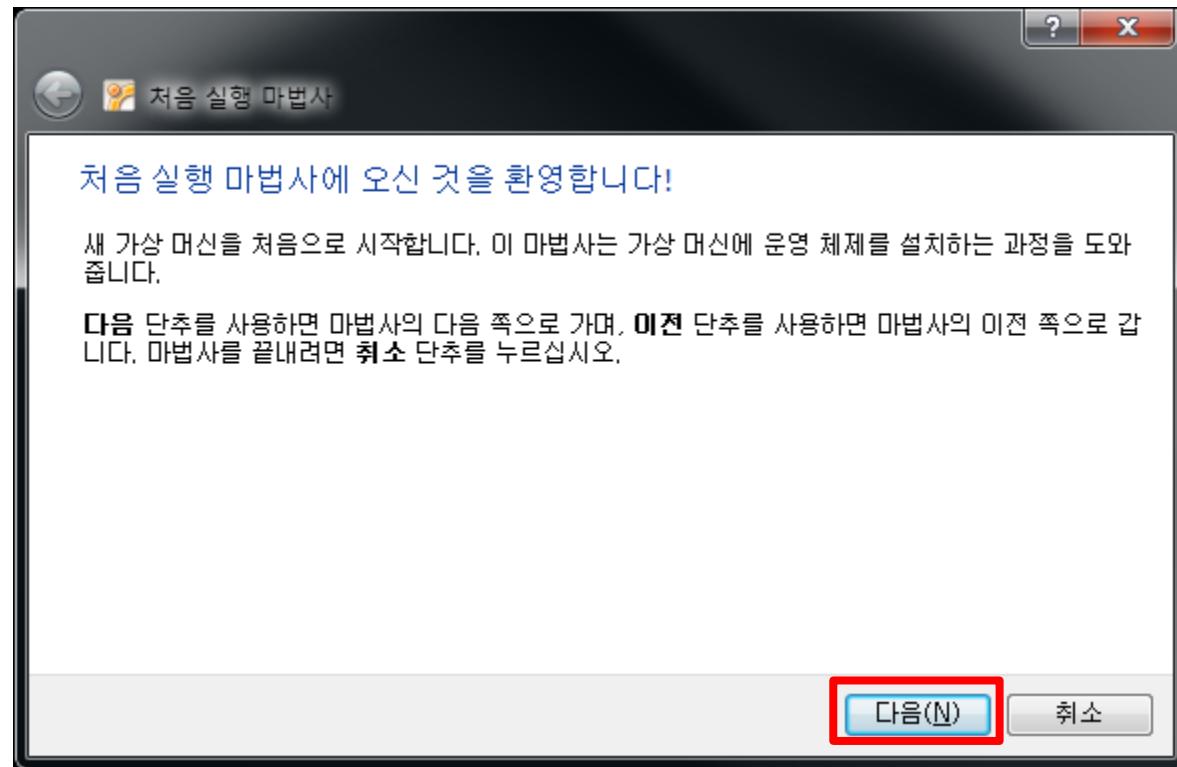
Linux(Ubuntu) 설치

- 시작 버튼을 눌러 Ubuntu 실행



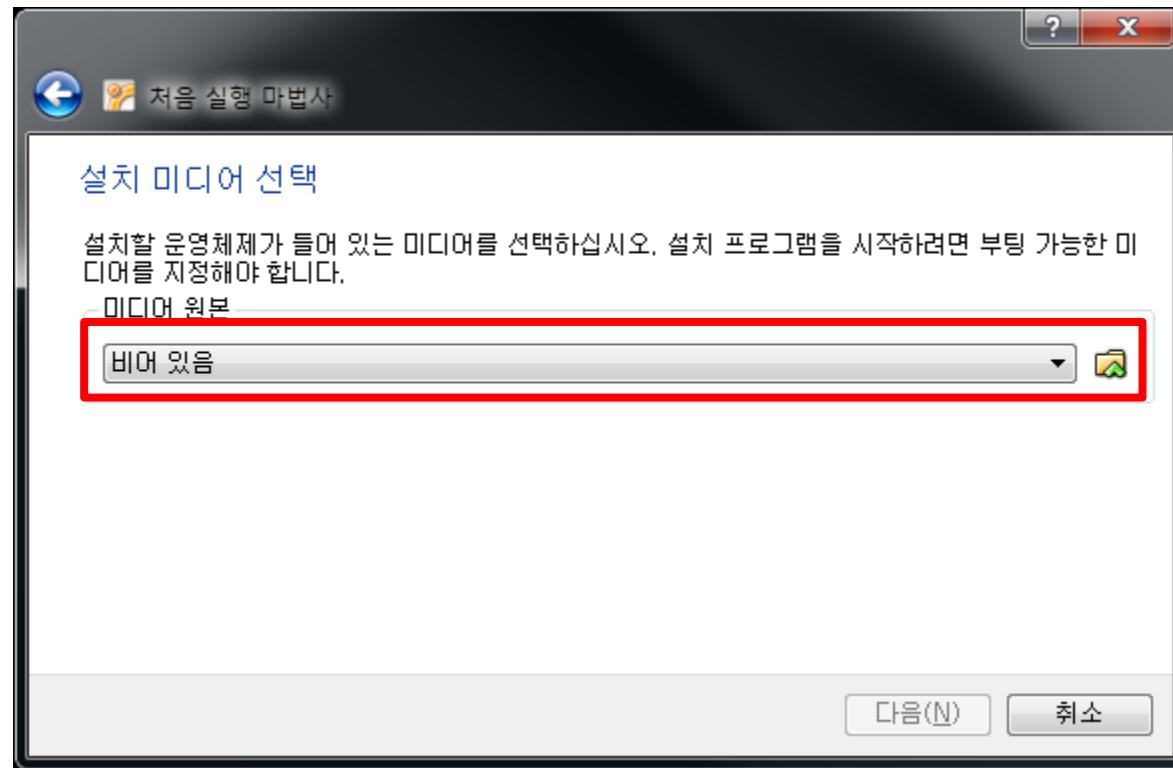
Linux(Ubuntu) 설치

- 처음 실행 마법사를 통하여 설치 시작



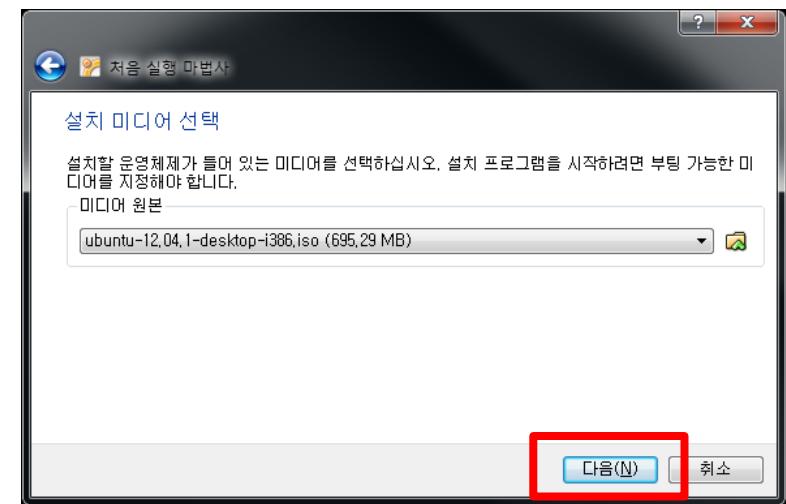
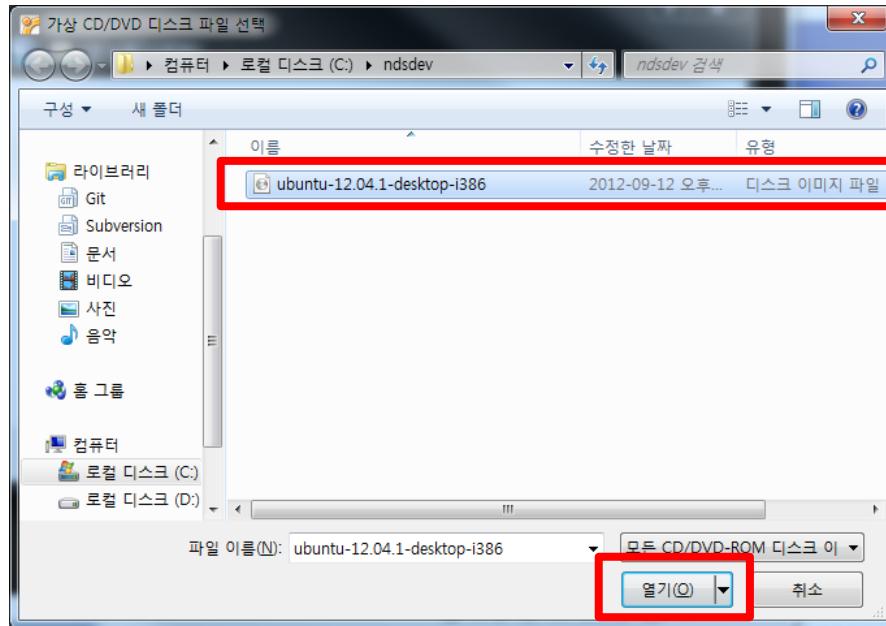
Linux(Ubuntu) 설치

- 다음과 같은 화면이 나오면 파일 탐색기 아이콘을 누름



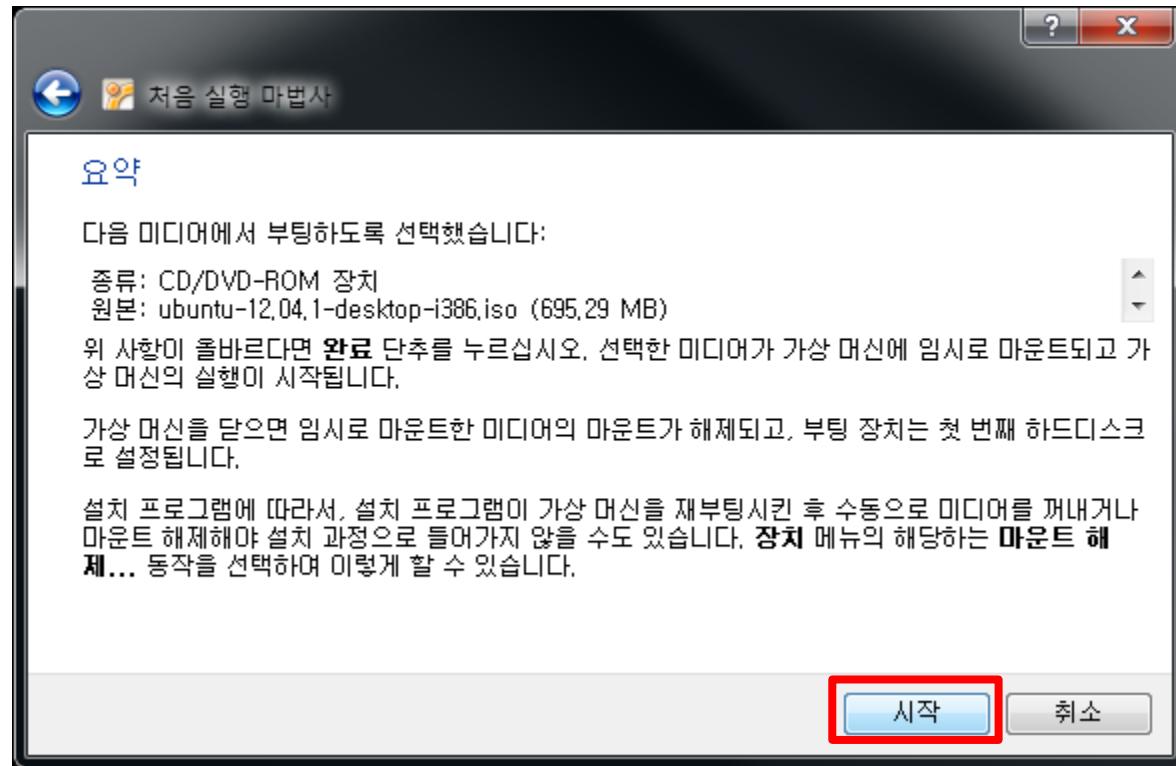
Linux(Ubuntu) 설치

- 설치할 Ubuntu 이미지를 선택한 뒤, 다음 버튼을 누름



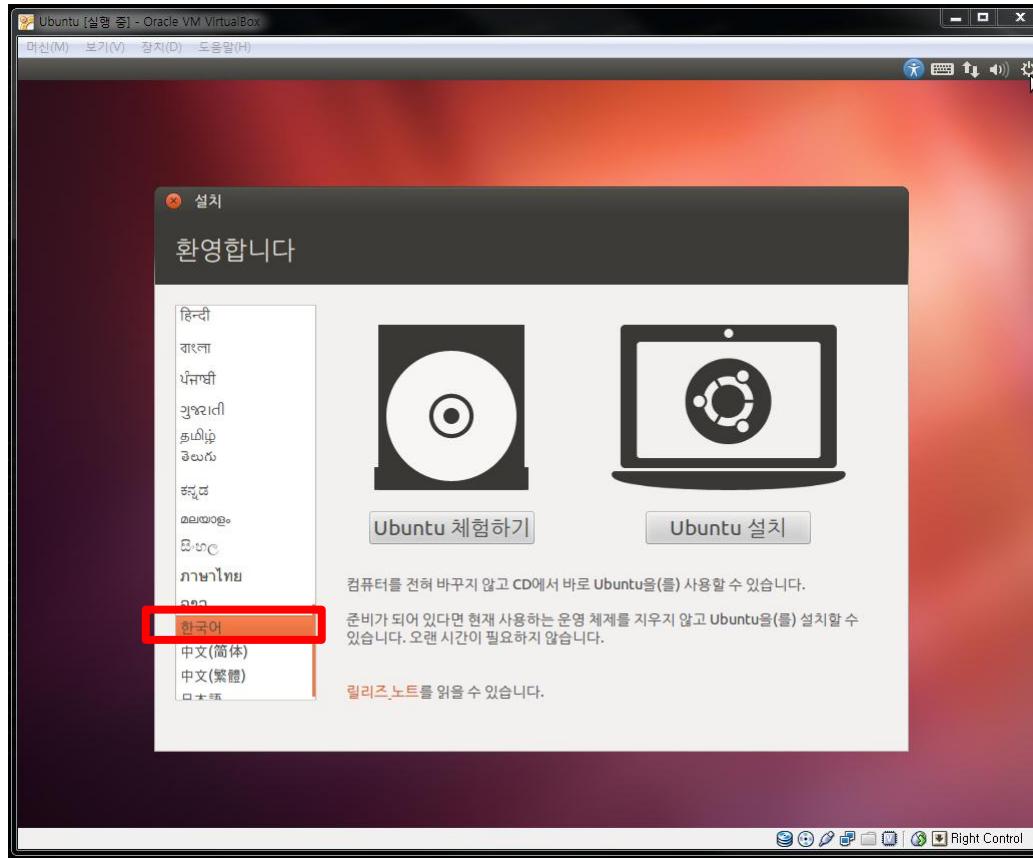
Linux(Ubuntu) 설치

- 시작 버튼을 눌러 Ubuntu를 설치



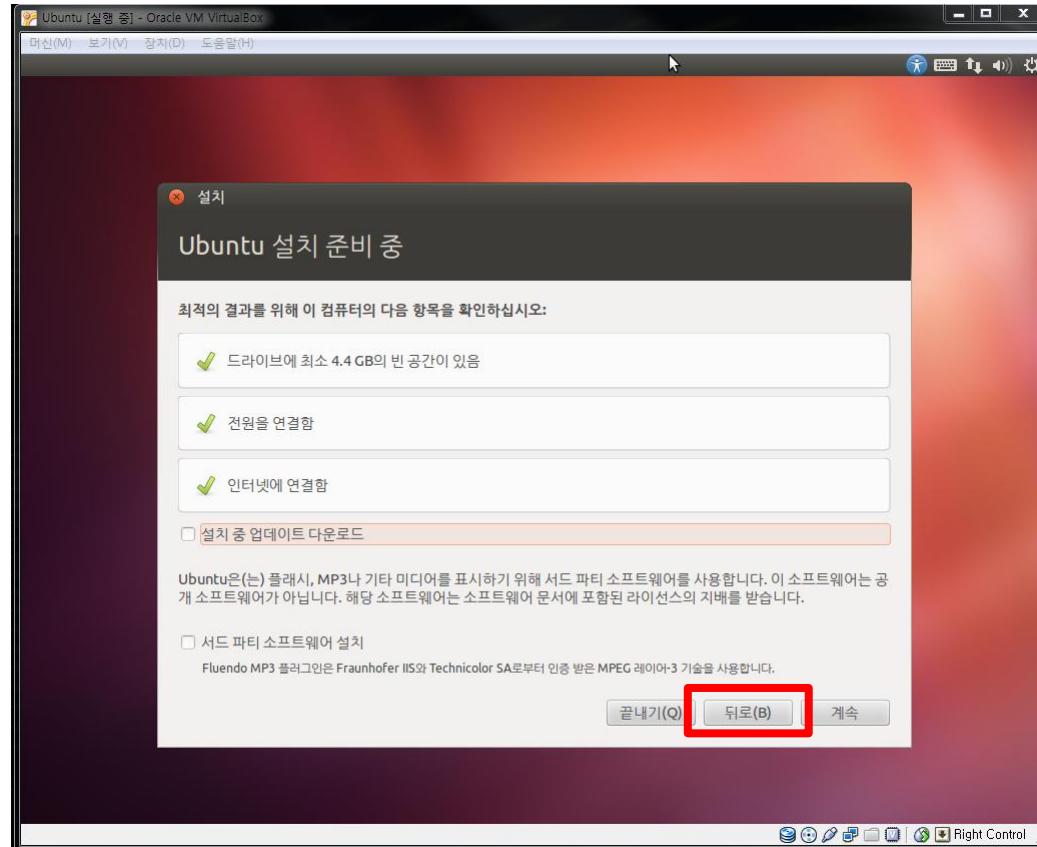
Linux(Ubuntu) 설치

- 언어 설정



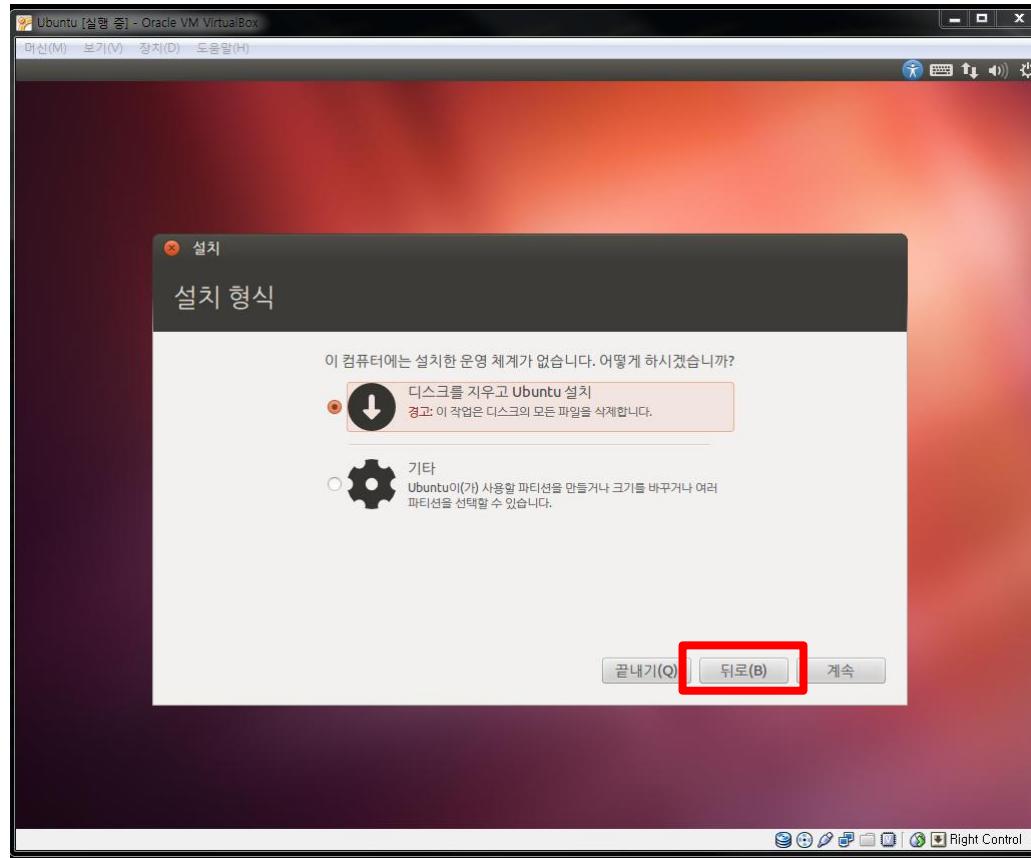
Linux(Ubuntu) 설치

- 업데이트, 기타 소프트웨어 설치 옵션 설정



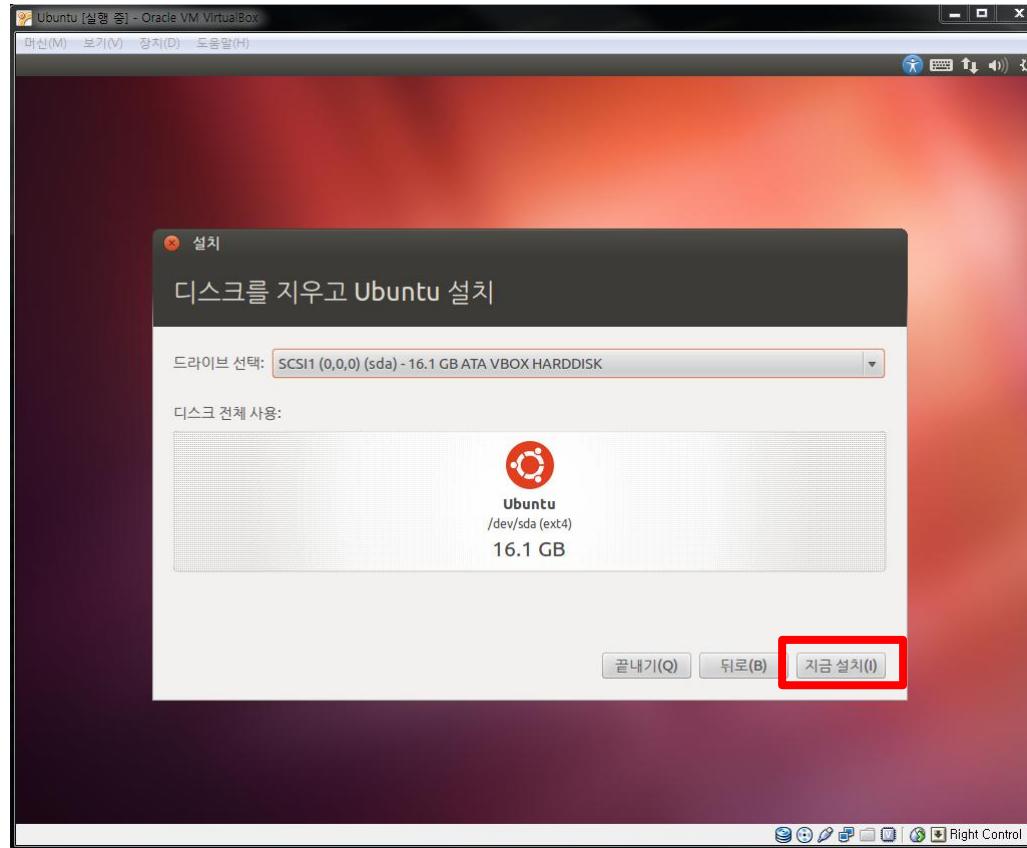
Linux(Ubuntu) 설치

- 설치 형식을 설정



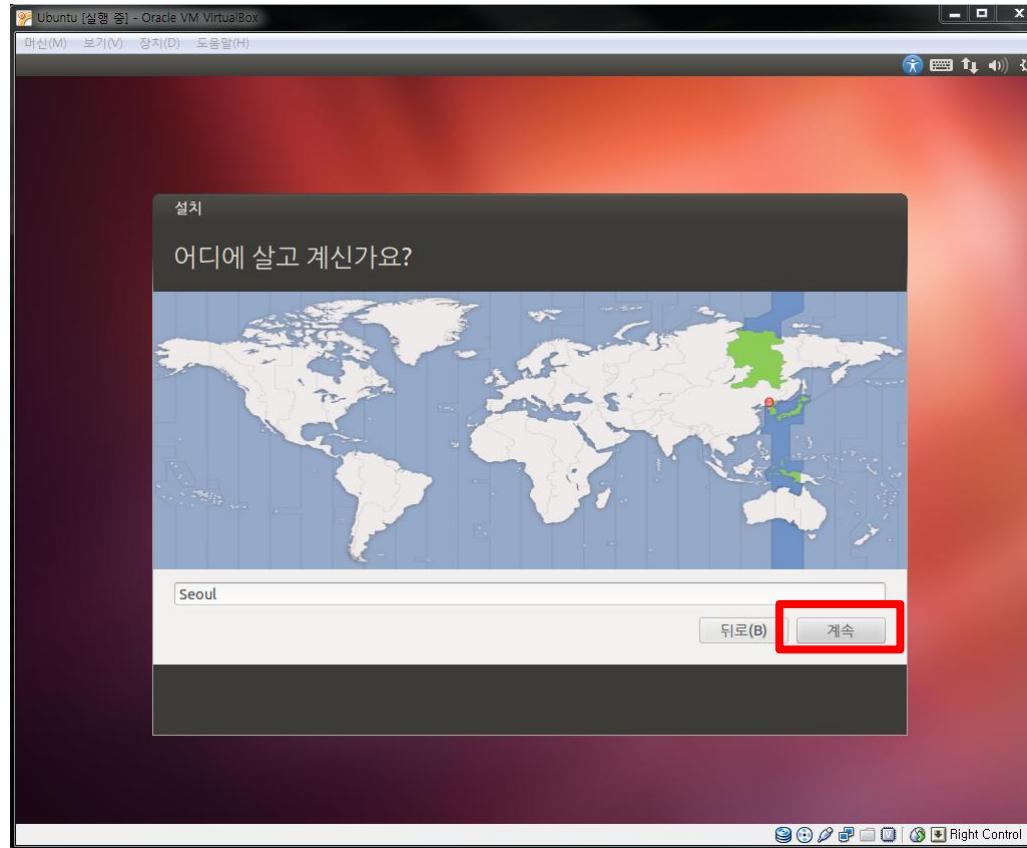
Linux(Ubuntu) 설치

- 설치 위치를 설정



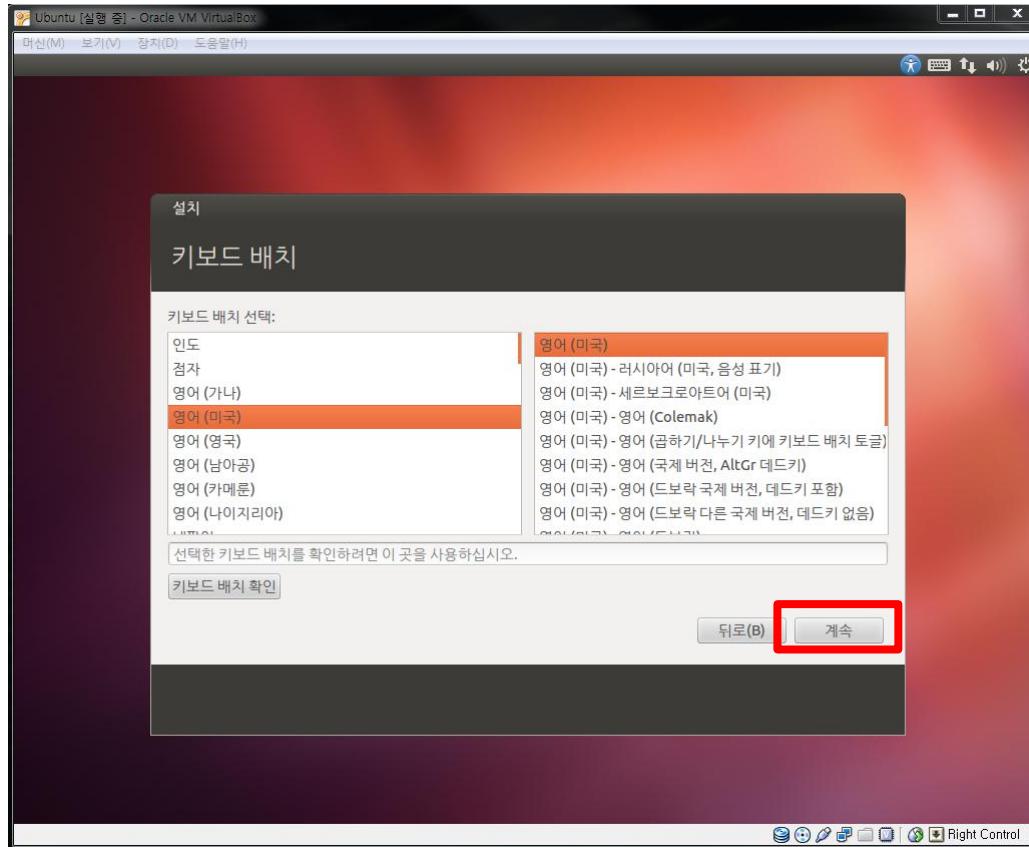
Linux(Ubuntu) 설치

- 위치 설정



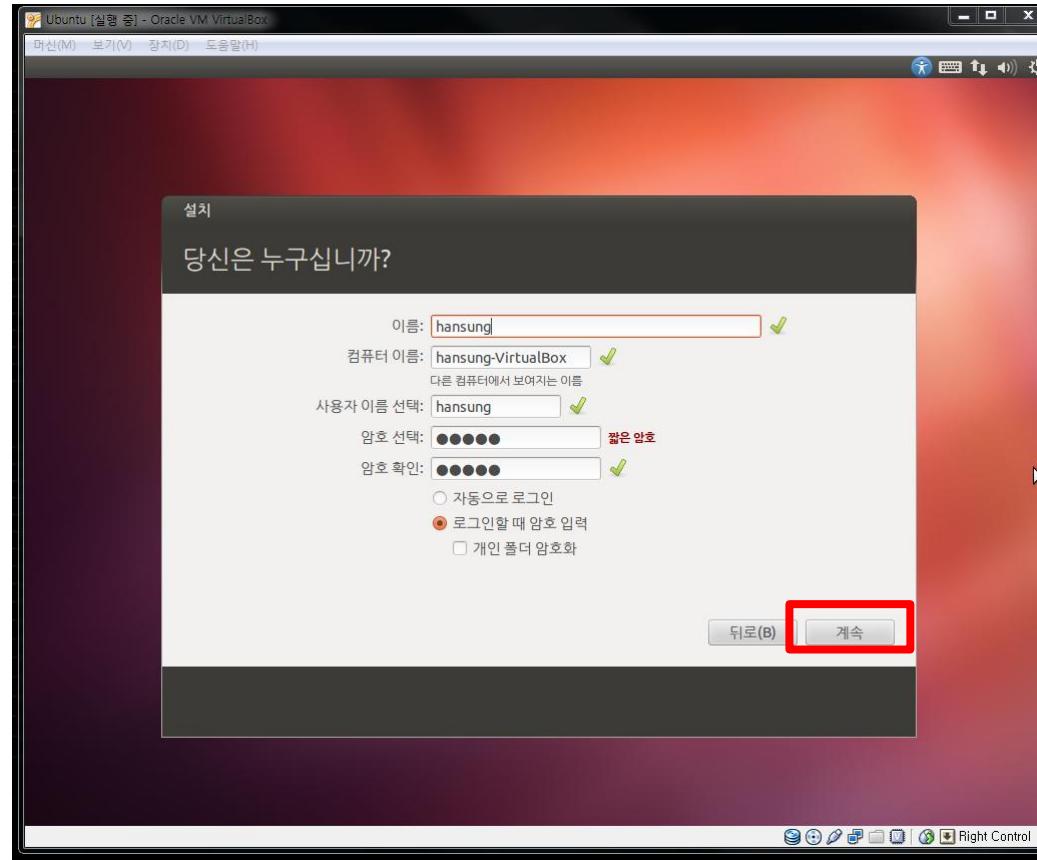
Linux(Ubuntu) 설치

- 키보드 배치 설정



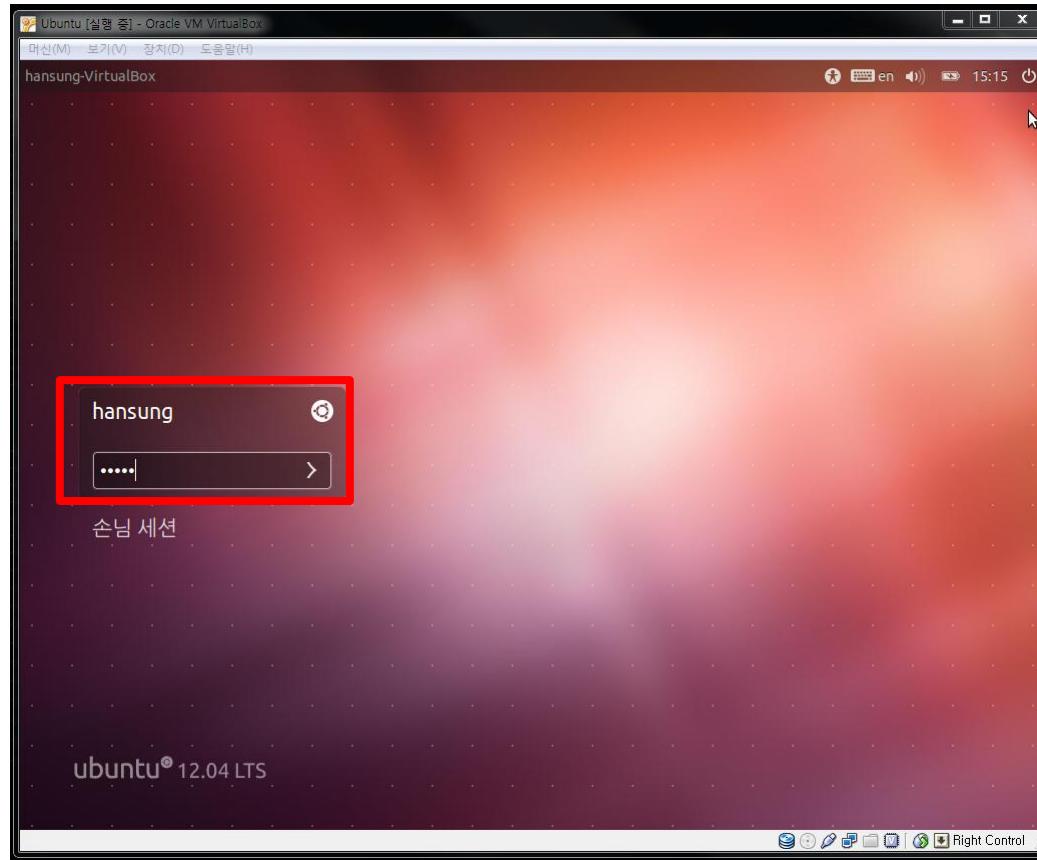
Linux(Ubuntu) 설치

- 사용자 정보 설정



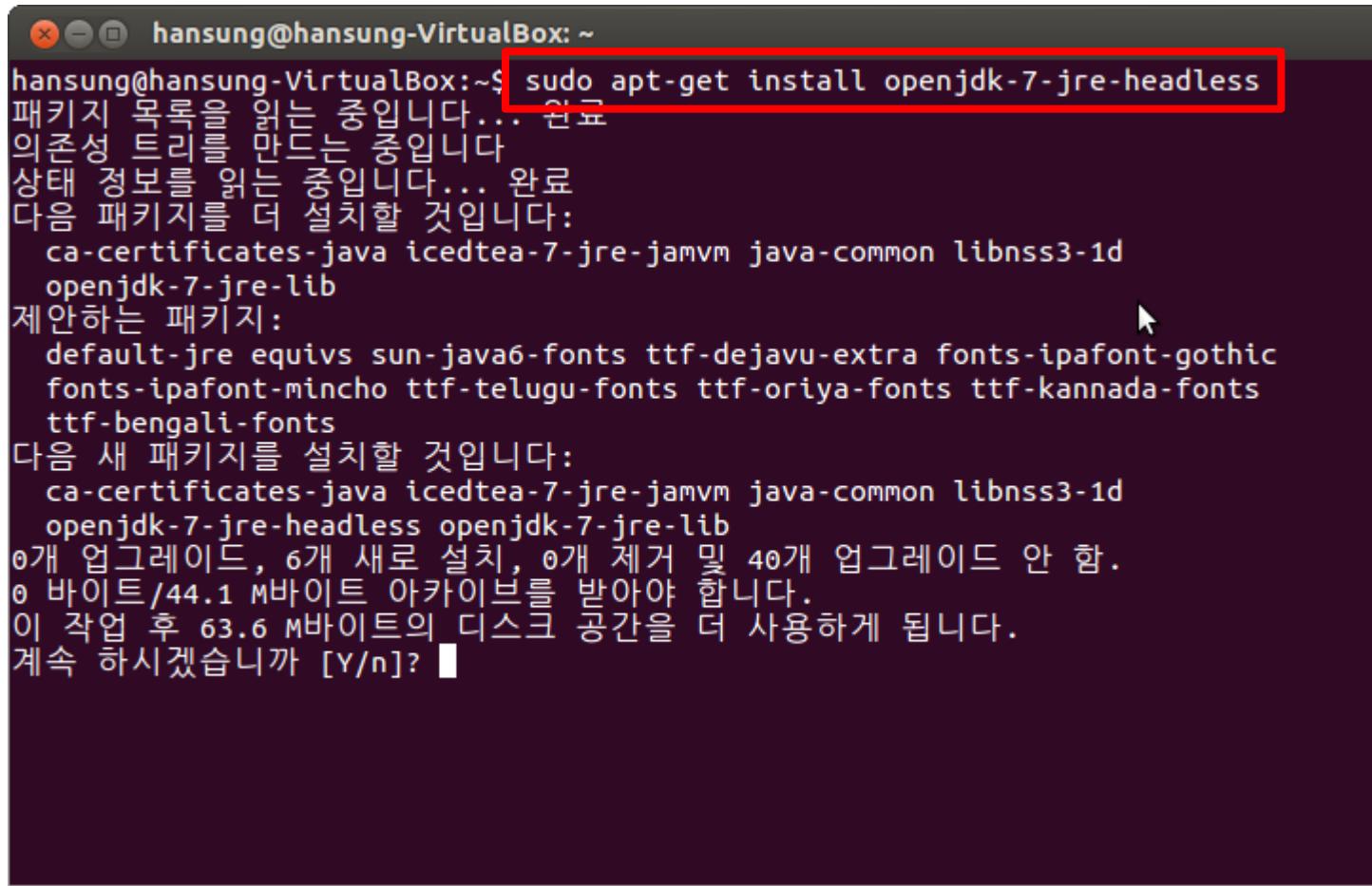
Linux(Ubuntu) 설치

- 다음과 같이 로그인 화면이 나오면 사용자 정보를 입력한 뒤 로그인



JDK 설치

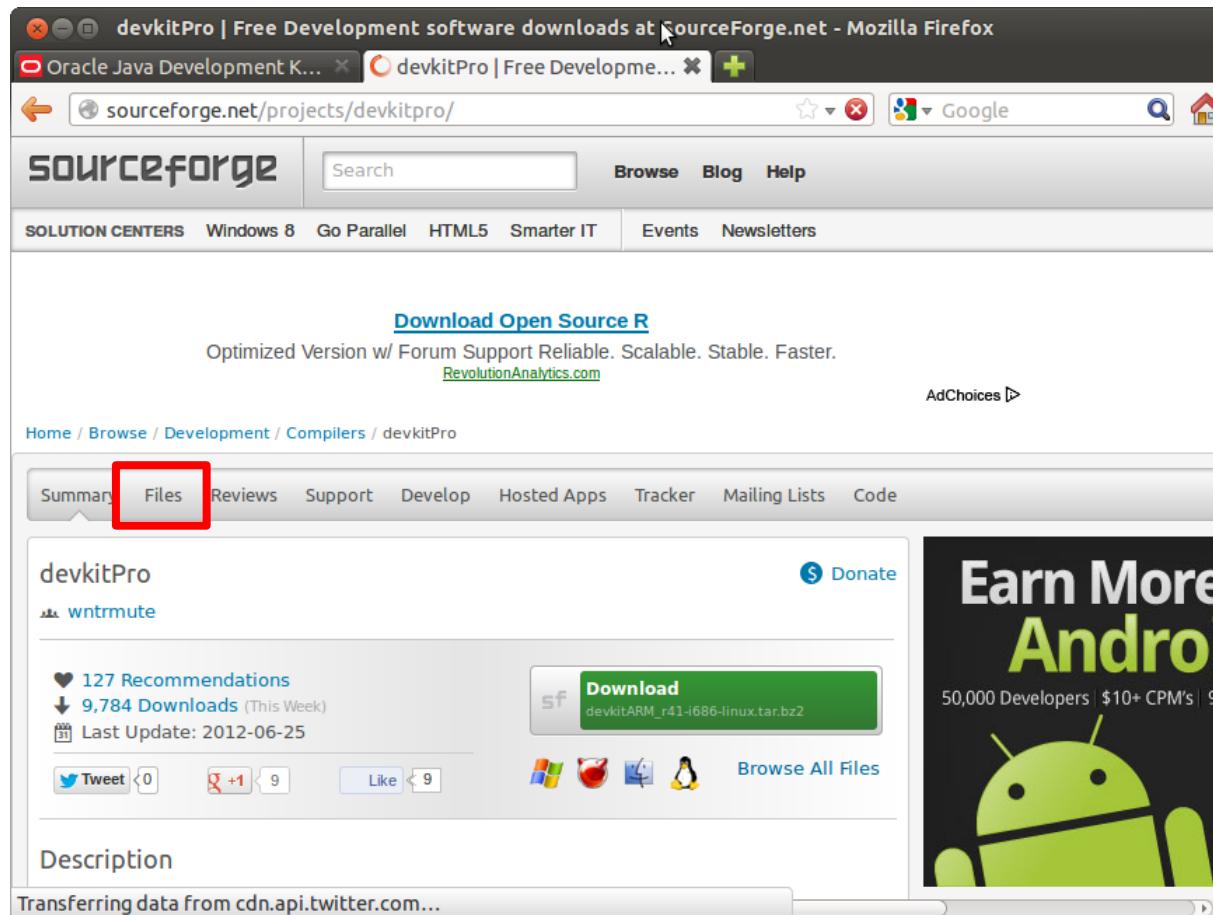
- 터미널에서 apt-get 명령어를 통해 JDK를 설치



```
hansung@hansung-VirtualBox:~$ sudo apt-get install openjdk-7-jre-headless
패키지 목록을 읽는 중입니다... 완료
의존성 트리를 만드는 중입니다
상태 정보를 읽는 중입니다... 완료
다음 패키지를 더 설치할 것입니다:
  ca-certificates-java icedtea-7-jre-jamvm java-common libnss3-1d
  openjdk-7-jre-lib
제안하는 패키지:
  default-jre equivs sun-java6-fonts ttf-dejavu-extra fonts-ipafont-gothic
  fonts-ipafont-mincho ttf-telugu-fonts ttf-oriya-fonts ttf-kannada-fonts
  ttf-bengali-fonts
다음 새 패키지를 설치할 것입니다:
  ca-certificates-java icedtea-7-jre-jamvm java-common libnss3-1d
  openjdk-7-jre-headless openjdk-7-jre-lib
0개 업그레이드, 6개 새로 설치, 0개 제거 및 40개 업그레이드 안 함.
0 바이트/44.1 M바이트 아카이브를 받아야 합니다.
이 작업 후 63.6 M바이트의 디스크 공간을 더 사용하게 됩니다.
계속 하시겠습니까 [Y/n]? █
```

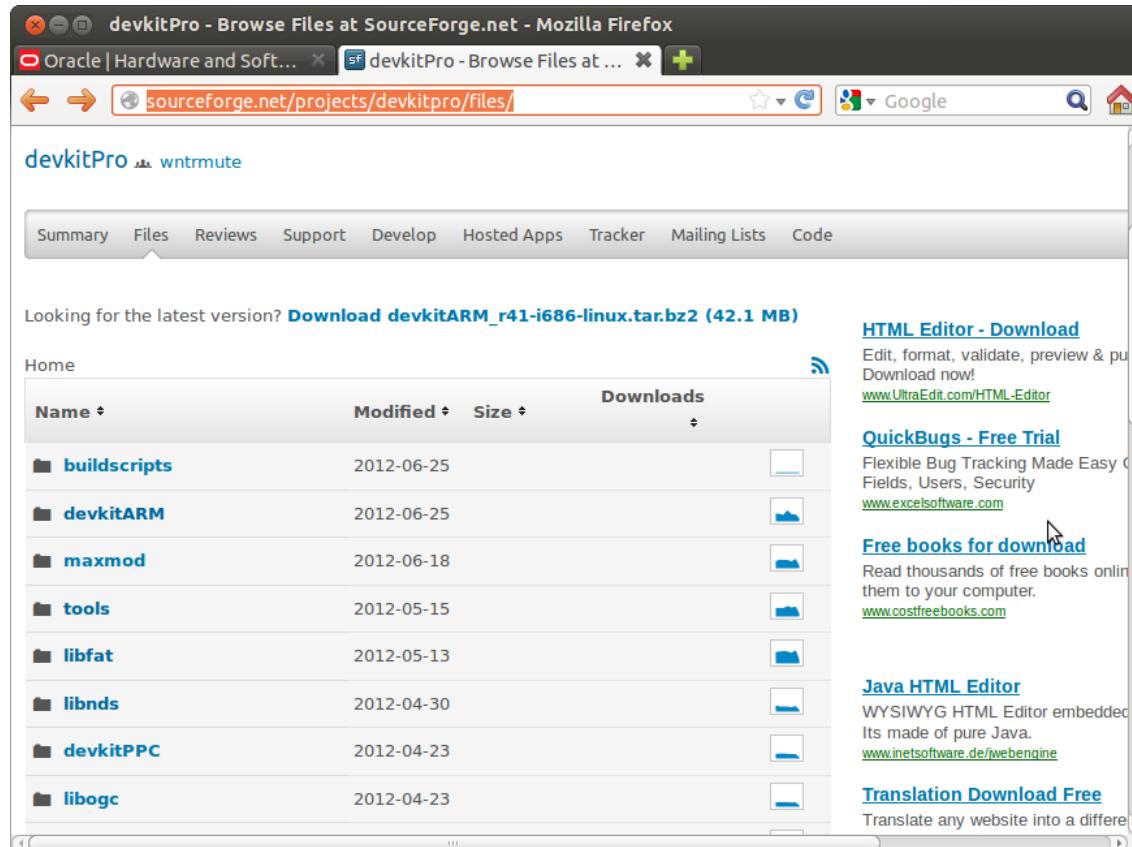
DevkitPro 설치

- <http://sourceforge.net/projects/devkitpro>에 접속하여, Files 항목을 선택



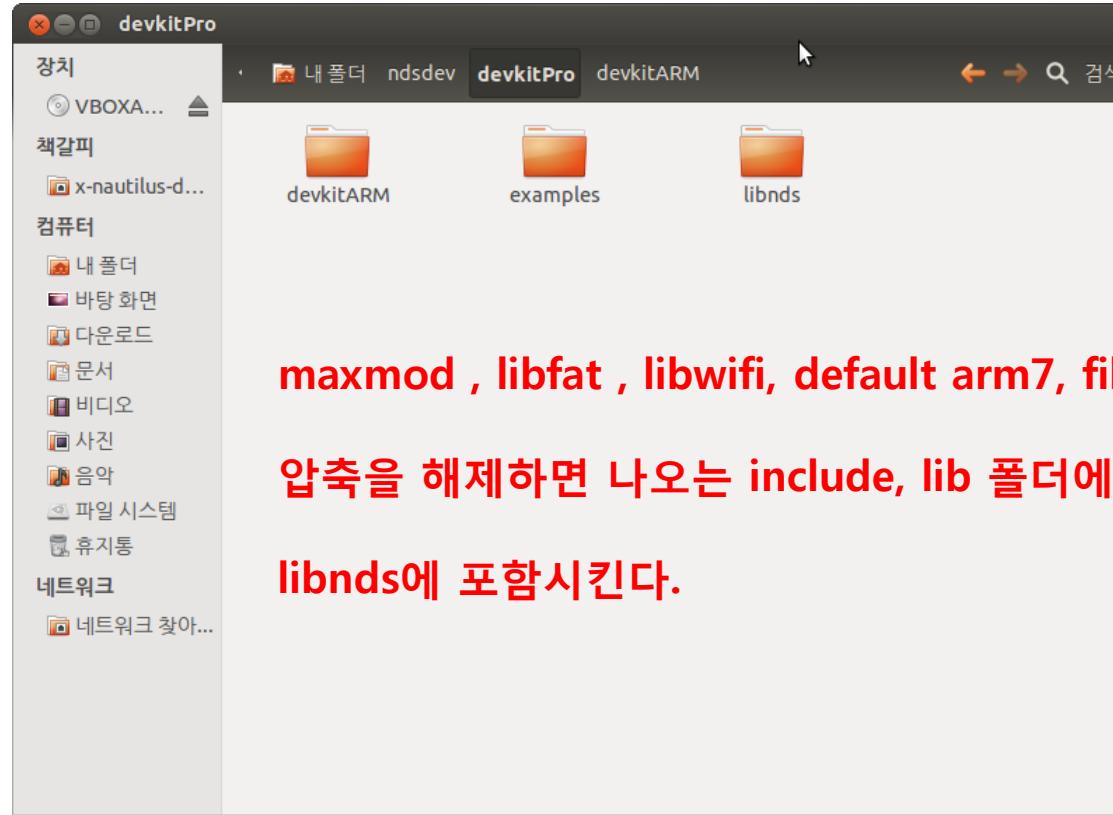
DevkitPro 설치

- 다음과 같은 화면이 나오면, 아래 표기된 항목들을 선택하여 다운로드
(devkitARM, maxmod, libfat, libnds, dswifi, default arm7, filesystem, examples)
(Linux 버전 및 Library를 선택하여 다운로드 받으면 됨)



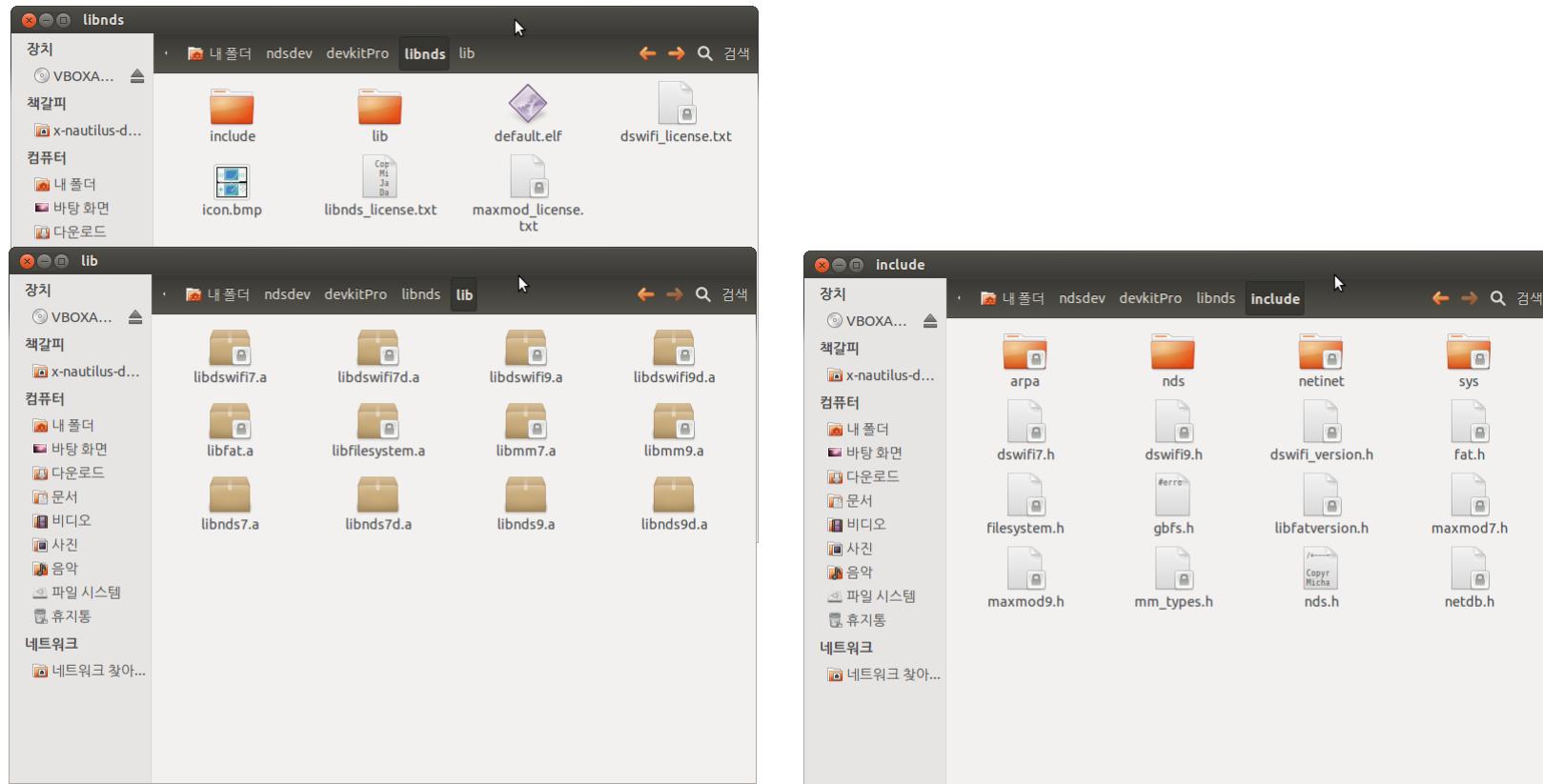
DevkitPro 설치

- devkitPro 프로젝트에서 다운로드 받은 파일을 모두 압축 해제한 뒤, 다음 화면과 같이 정리



DevkitPro 설치

- devkitPro 프로젝트에서 다운로드 받은 파일을 모두 압축 해제한 뒤, 다음 화면과 같이 정리



DevkitPro 설치

- 다음과 같이 환경 변수를 설정

The image shows two screenshots of a Linux terminal window. The top screenshot is from a root shell (root@hansung-VirtualBox: ~) and displays the command 'vi .bashrc'. The bottom screenshot is from a user shell (hansung@hansung-VirtualBox: ~) and shows the contents of the .bashrc file being edited. A red box highlights the line 'export PATH=\$PATH:/home/hansung/ndsdev/devkitPro/devkitARM/bin' in the file. The terminal window also shows the message '.bashrc' 110 lines, 3551 characters written at the bottom.

```
root@hansung-VirtualBox: ~
root@hansung-VirtualBox:~/ndstest# cd
root@hansung-VirtualBox:~# vi .bashrc

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

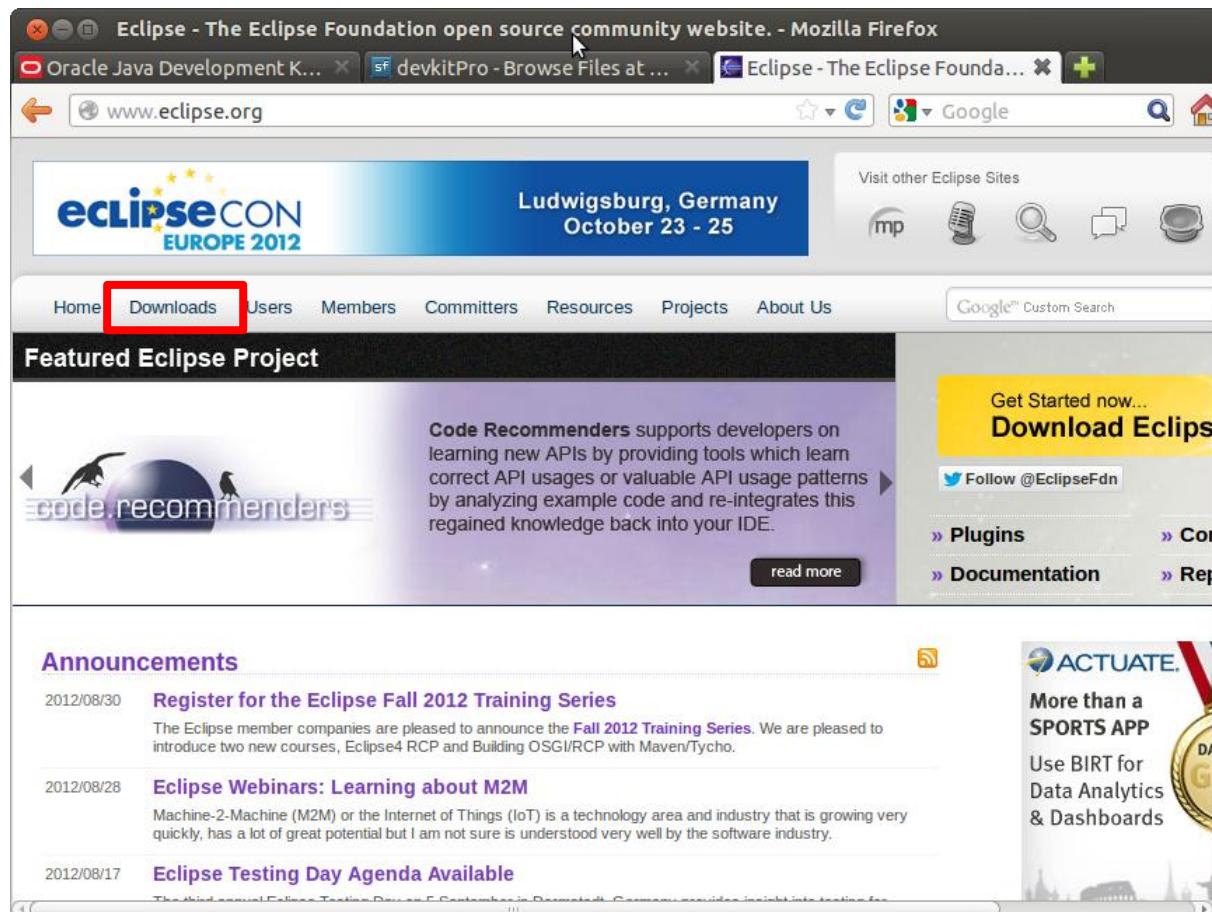
# enable programmable completion features (you don't need to enable
# this, if it's already enabled in /etc/bash.bashrc and /etc/profile
# sources /etc/bash.bashrc).
if [ -f /etc/bash_completion ] && ! shopt -oq posix; then
    . /etc/bash_completion
fi

export PATH=$PATH:/home/hansung/ndsdev/devkitPro/devkitARM/bin

.bashrc" 110 lines, 3551 characters written
```

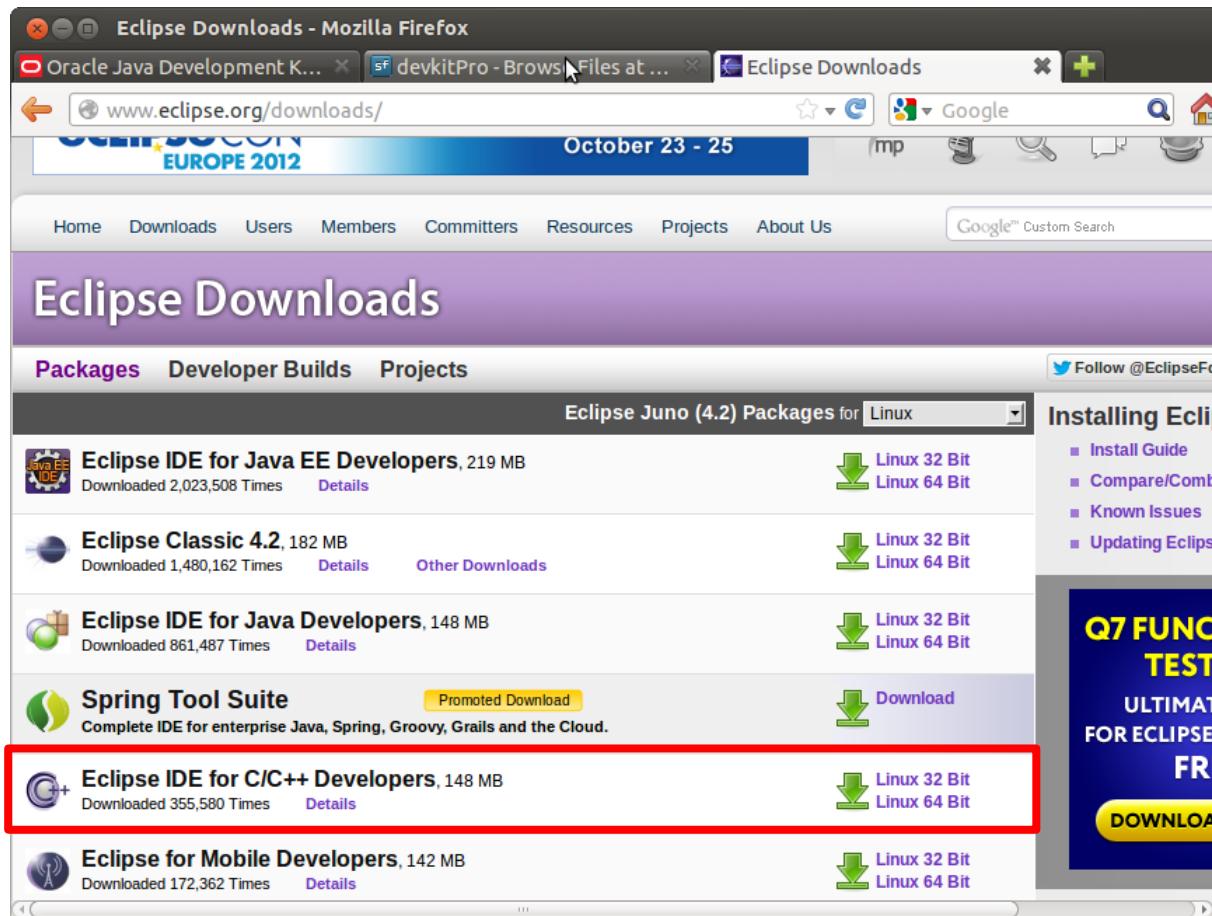
Eclipse 설치

- www.eclipse.org에 접속하여, Downloads 메뉴 클릭



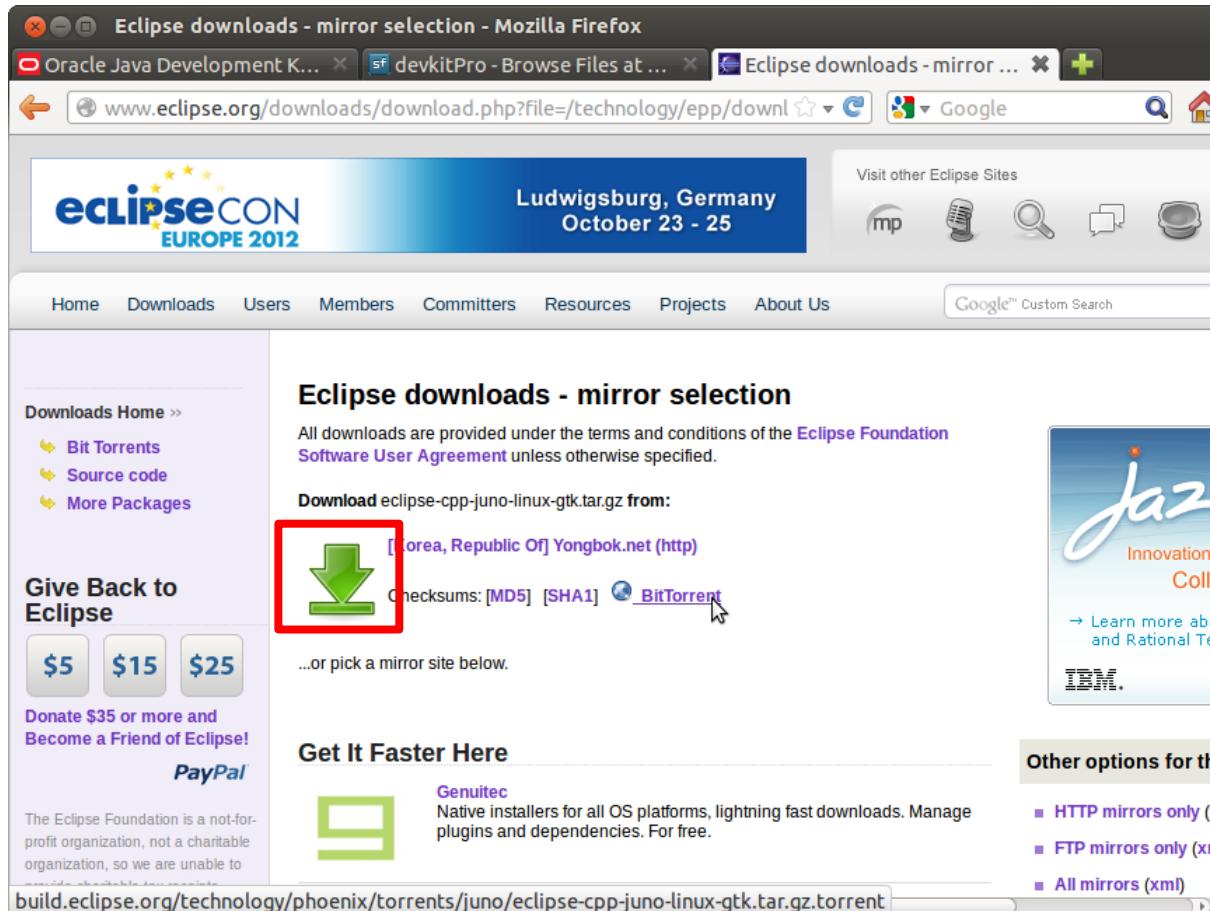
Eclipse 설치

- Eclipse IDE for C/C++ Developers의 Linux 32bit를 눌러 다운로드



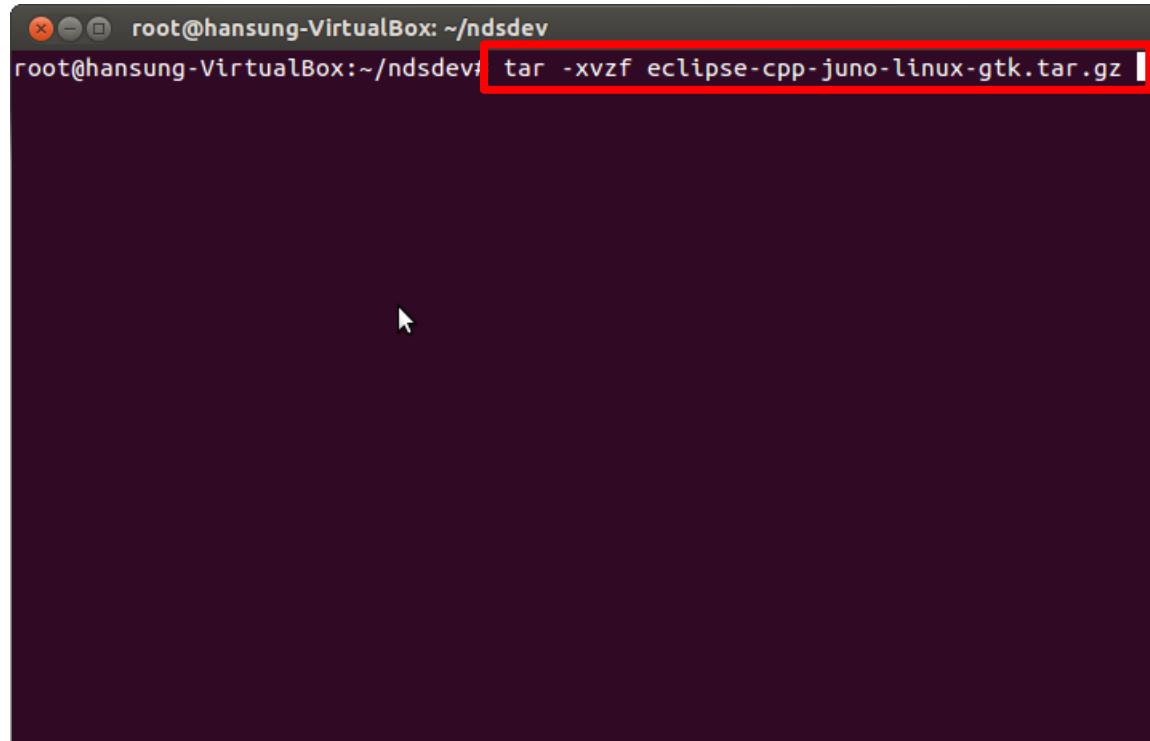
Eclipse 설치

- 녹색 화살표를 눌러 다운로드



Eclipse 설치

- 다운로드 받은 파일을 압축 해제
(Eclipse CDT는 앞에서 설치했던 JDK만 있으면 실행 가능함)



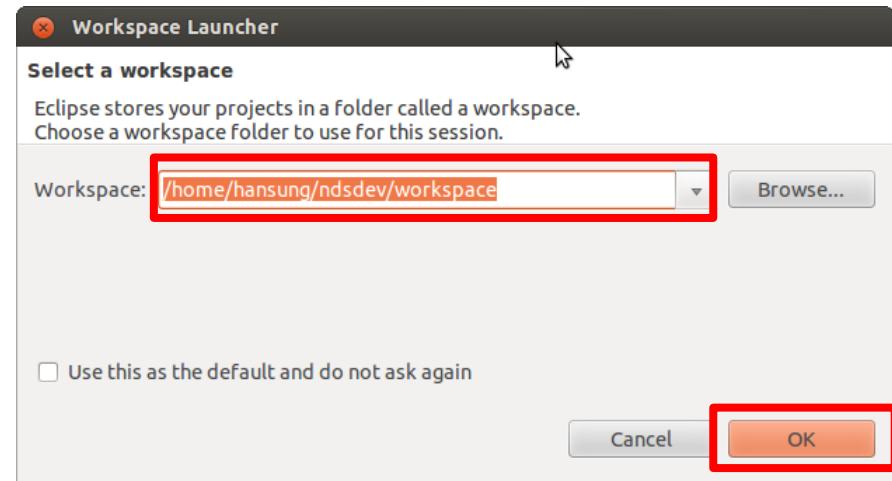
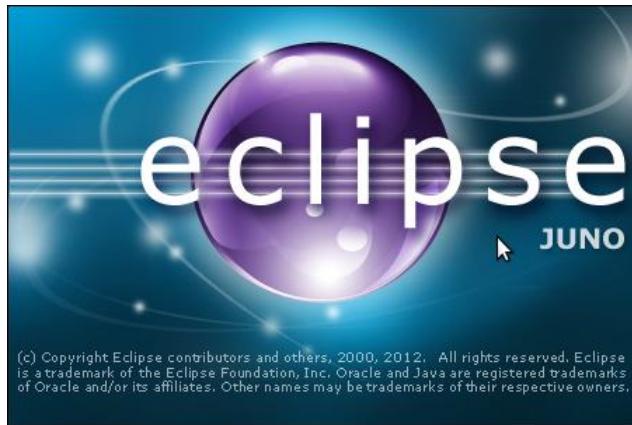
```
root@hansung-VirtualBox: ~/ndsdev
root@hansung-VirtualBox:~/ndsdev| tar -xvzf eclipse-cpp-juno-linux-gtk.tar.gz |
```

A screenshot of a terminal window with a dark background and white text. The window title bar says "root@hansung-VirtualBox: ~/ndsdev". In the main area, the command "tar -xvzf eclipse-cpp-juno-linux-gtk.tar.gz" is typed. A red rectangle highlights the command line. A cursor arrow is visible at the bottom center of the window.

Eclipse 설치

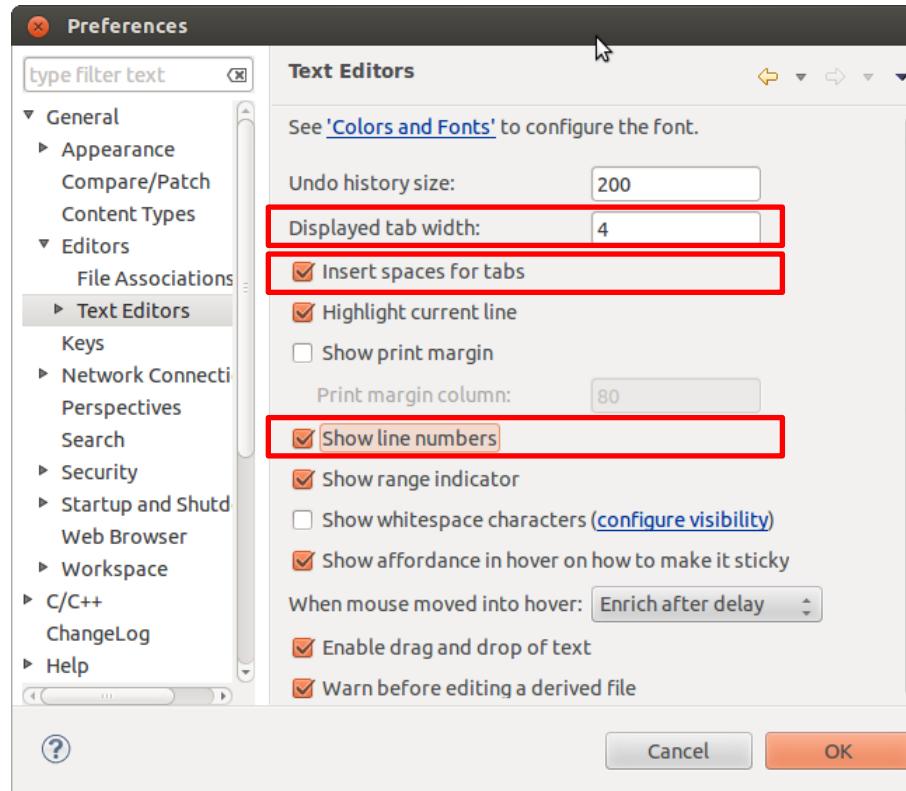
- Eclipse를 실행하면 다음 시작 화면이 나옴

/home/hansung/ndsdev/workspace로 프로젝트를 저장할 기본 폴더를 지정



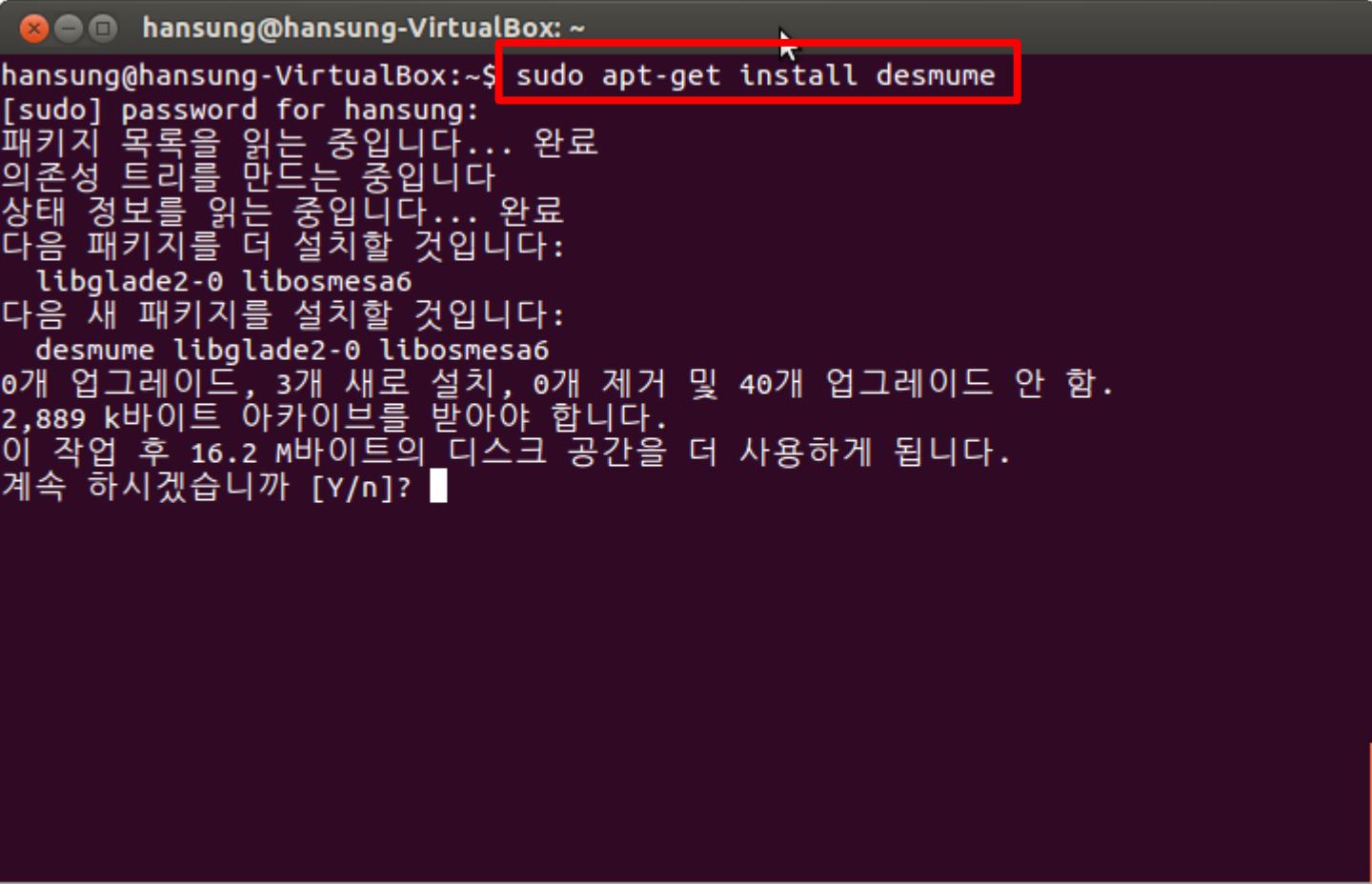
Eclipse 설치

- 소스 코드 일관성을 위해 <Windows> → <Preferences> 설정 창에서 Editors → Text Editors의 Displayed width는 4로, Insert spaces for tabs, Show line numbers를 체크



Emulator 설치

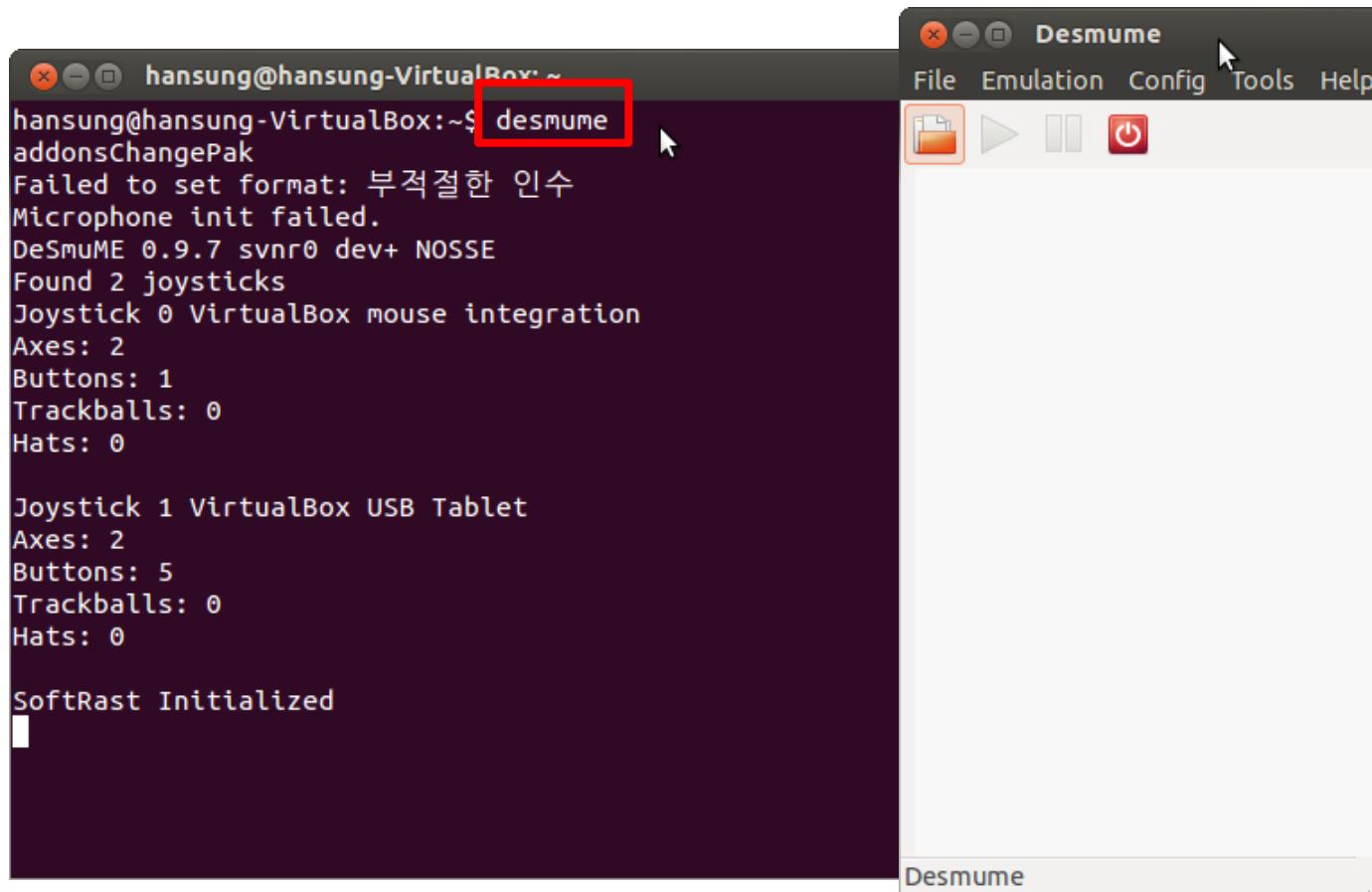
- 터미널에서 다음의 apt-get 명령어를 이용해 Emulator를 설치



```
hansung@hansung-VirtualBox:~$ sudo apt-get install desmume
[sudo] password for hansung:
패키지 목록을 읽는 중입니다... 완료
의존성 트리를 만드는 중입니다
상태 정보를 읽는 중입니다... 완료
다음 패키지를 더 설치할 것입니다:
  libglade2-0 libosmesa6
다음 새 패키지를 설치할 것입니다:
  desmume libglade2-0 libosmesa6
0개 업그레이드, 3개 새로 설치, 0개 제거 및 40개 업그레이드 안 함.
2,889 kB바이트 아카이브를 받아야 합니다.
이 작업 후 16.2 kB바이트의 디스크 공간을 더 사용하게 됩니다.
계속 하시겠습니까 [Y/n]? █
```

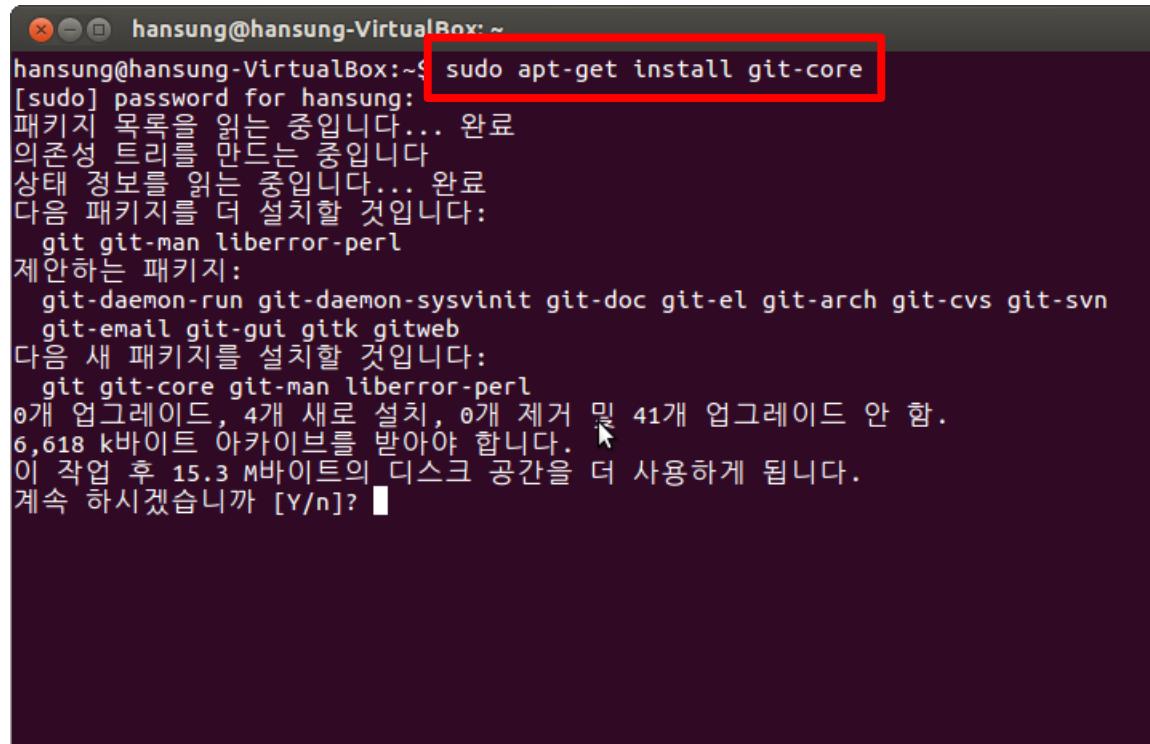
Emulator 설치

- 다음 화면과 같이 Emulator를 실행



Git 설치

- 터미널에서 다음의 apt-get 명령어를 이용해 Git 패키지를 설치

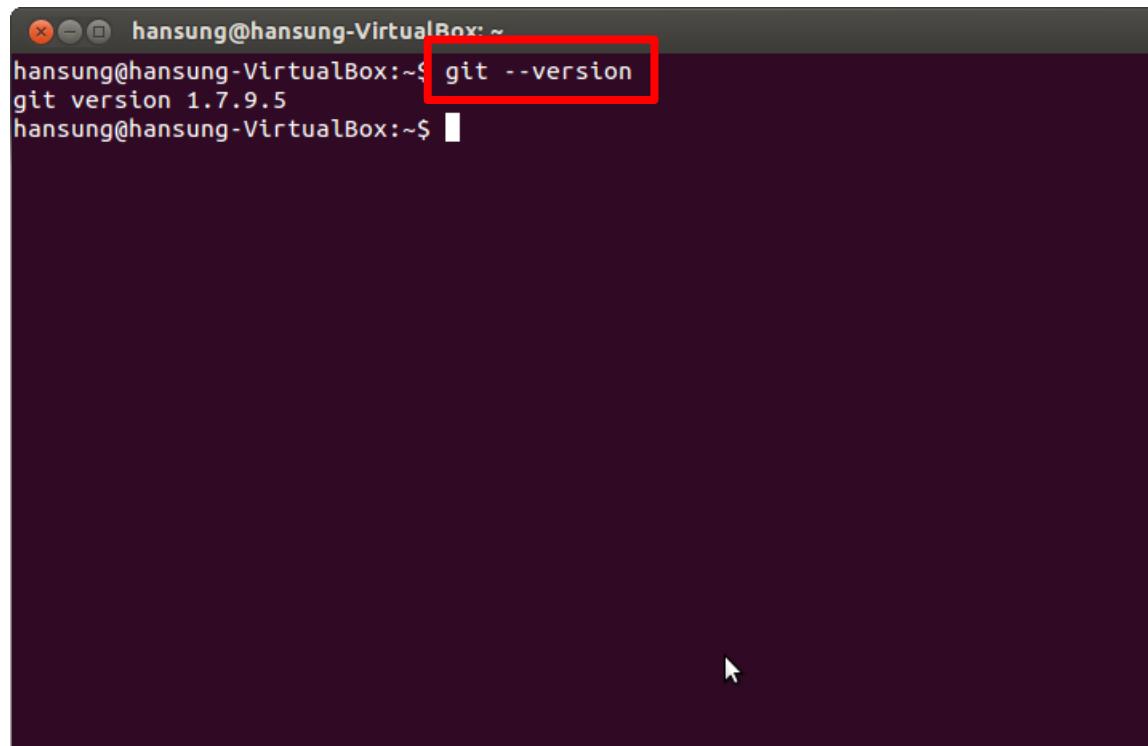


The screenshot shows a terminal window titled "hansung@hansung-VirtualBox: ~". The user has run the command "sudo apt-get install git-core". A red box highlights the command line. The terminal then prompts for a password, which is partially obscured. It continues to show the progress of the package installation, including dependency resolution and upgrade information. The user is asked if they want to proceed with the installation.

```
hansung@hansung-VirtualBox:~$ sudo apt-get install git-core
[sudo] password for hansung:
패키지 목록을 읽는 중입니다... 완료
의존성 트리를 만드는 중입니다
상태 정보를 읽는 중입니다... 완료
다음 패키지를 더 설치할 것입니다:
  git git-man liberror-perl
제안하는 패키지:
  git-daemon-run git-daemon-sysvinit git-doc git-el git-arch git-cvs git-svn
  git-email git-gui gitk gitweb
다음 새 패키지를 설치할 것입니다:
  git git-core git-man liberror-perl
0개 업그레이드, 4개 새로 설치, 0개 제거 및 41개 업그레이드 안 함.
6,618 kB바이트 아카이브를 받아야 합니다.
이 작업 후 15.3 kB바이트의 디스크 공간을 더 사용하게 됩니다.
계속 하시겠습니까 [Y/n]? █
```

Git 설치

- 설치가 완료되면, git --version 명령어를 입력하여 제대로 설치되었는지 확인



```
hansung@hansung-VirtualBox:~$ git --version
git version 1.7.9.5
hansung@hansung-VirtualBox:~$
```

A screenshot of a terminal window titled "hansung@hansung-VirtualBox: ~". The window contains the command "git --version" followed by its output "git version 1.7.9.5". The command itself is highlighted with a red rectangular box. The terminal has a dark background and a light-colored text area.

마이크로 프로세서 응용 및 설계 실습

- 수업시간에는 미리 만들어놓은 Linux(Ubuntu) 이미지를 이용하여 실습을 진행하므로 15Page의 단계에서 기존 하드 디스크 사용하기 항목을 선택한 뒤, Linux로 부팅

(닌텐도DS를 이용한 임베디드 소프트웨어 개발에 필요한 모든 환경이 갖추어져 있음)

프로젝트 홈페이지

- 프로젝트 홈페이지

<http://hl1itj.github.com/nds-ide>



[Source]

- Github Repository

[Game]

- 매년 개발된 팀 프로젝트 결과물

[Document]

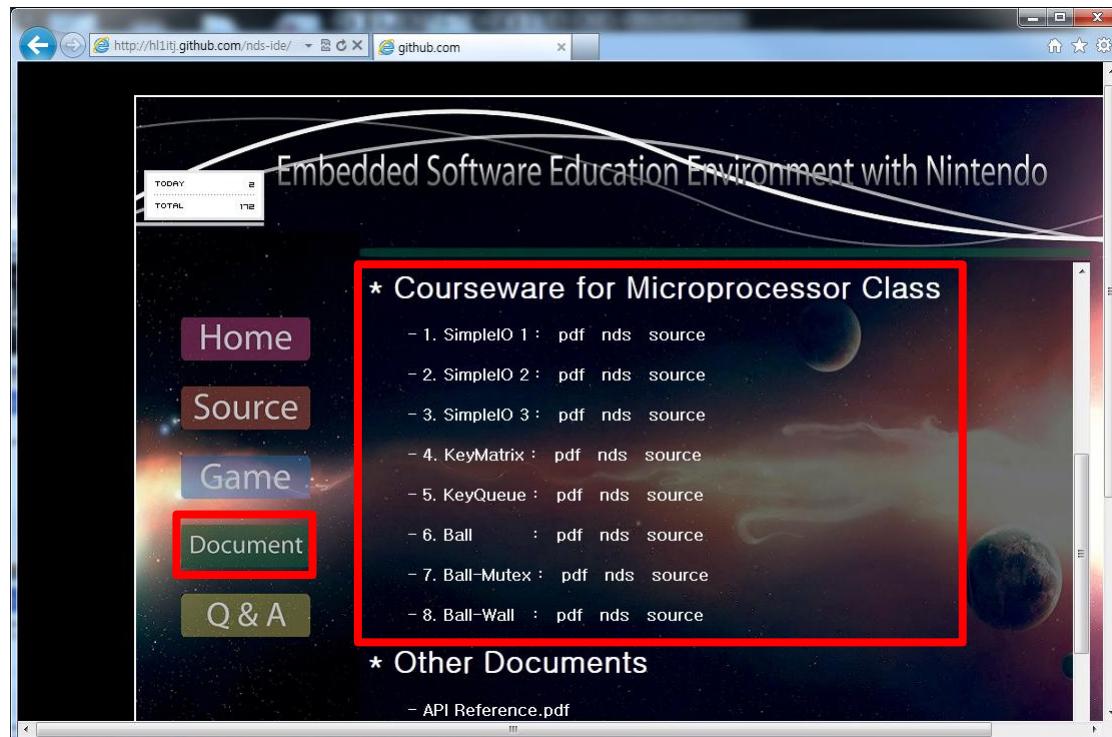
- 수업 자료, 매뉴얼 등

[Q & A]

- 질의 응답 게시판

프로젝트 홈페이지

- 프로젝트 홈페이지 (<http://hl1itj.github.com/nds-ide>)



수업자료는
Document 메뉴를 선택하면
확인할 수 있음
총 8개의 실습 과제로 구성

[PDF]
수업에 필요한 문서
(실습 내용 설명 및 과제)

[NDS]
실습 내용이 모두 구현된 실행
이미지

[SOURCE]
실습을 위한 템플릿 소스

프로젝트 홈페이지

- 실습에 필요한 소스 및 문서는 Github 사이트에 등록되어 있는 Repository를 통해 공유하고, 각자의 실습 결과 또한 해당 Repository에 등록하는식으로 진행

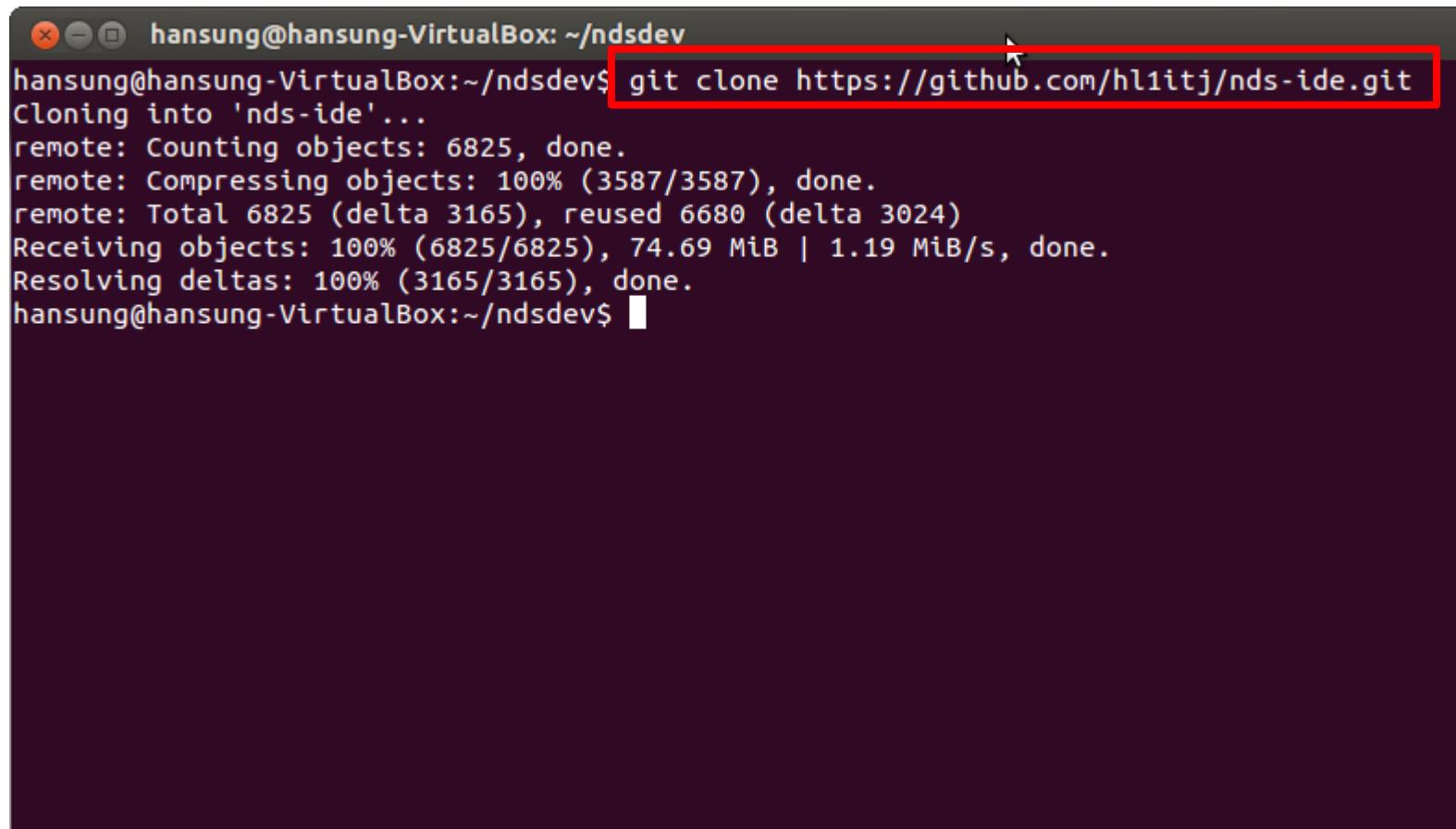
name	age	message
0892067	2 hours ago	0892067 Song tae woong [Song-tae-woong]
example	2 hours ago	This is helloworld example [Song-tae-woong]
.DS_Store	3 hours ago	Revert "clean .DS_Store to start new branch:microp... [Song-tae-woong]
README.txt	3 hours ago	clean contents in branch:master [TaeWoong-Song]

각자의 결과물은
Github 사이트의 Repository에
(<https://github.com/hl1itj/nds-ide>)
microprocessor 브랜치를 사용하여
학번으로 관리

- 수업을 듣는 모든 학생은 Github ID를 만든 뒤,
<https://github.com/hl1itj/nds-ide>에서 프로젝트 Watch하기

실습 및 과제 소스 다운로드

- 터미널에서 git clone 명령어를 이용하여 프로젝트를 다운로드



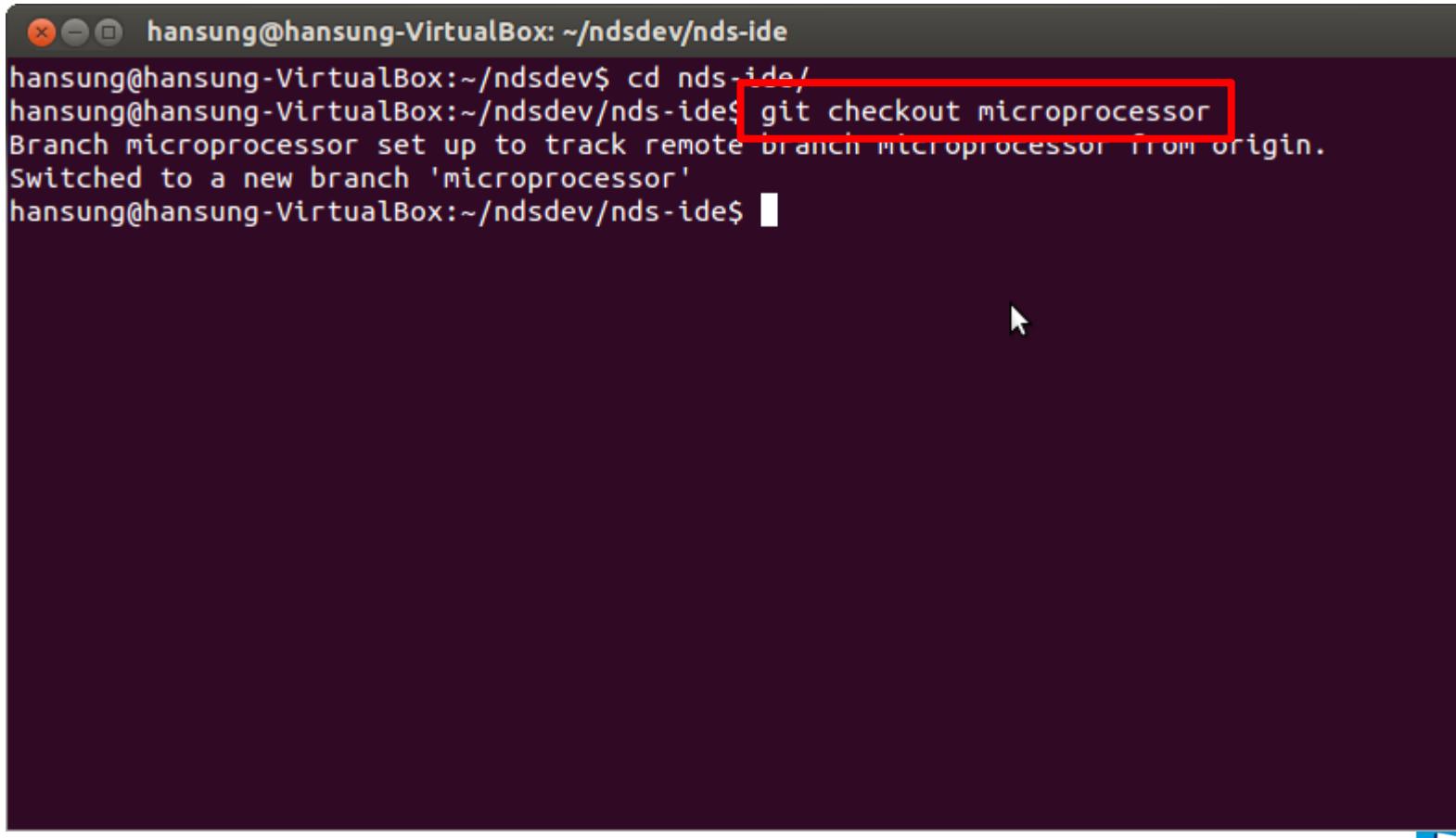
A screenshot of a terminal window titled "hansung@hansung-VirtualBox: ~/ndsdev". The window contains the following text:

```
git clone https://github.com/hl1itj/nds-ide.git
Cloning into 'nds-ide'...
remote: Counting objects: 6825, done.
remote: Compressing objects: 100% (3587/3587), done.
remote: Total 6825 (delta 3165), reused 6680 (delta 3024)
Receiving objects: 100% (6825/6825), 74.69 MiB | 1.19 MiB/s, done.
Resolving deltas: 100% (3165/3165), done.
hansung@hansung-VirtualBox:~/ndsdev$
```

The command "git clone https://github.com/hl1itj/nds-ide.git" is highlighted with a red rectangle.

실습 및 과제 소스 다운로드

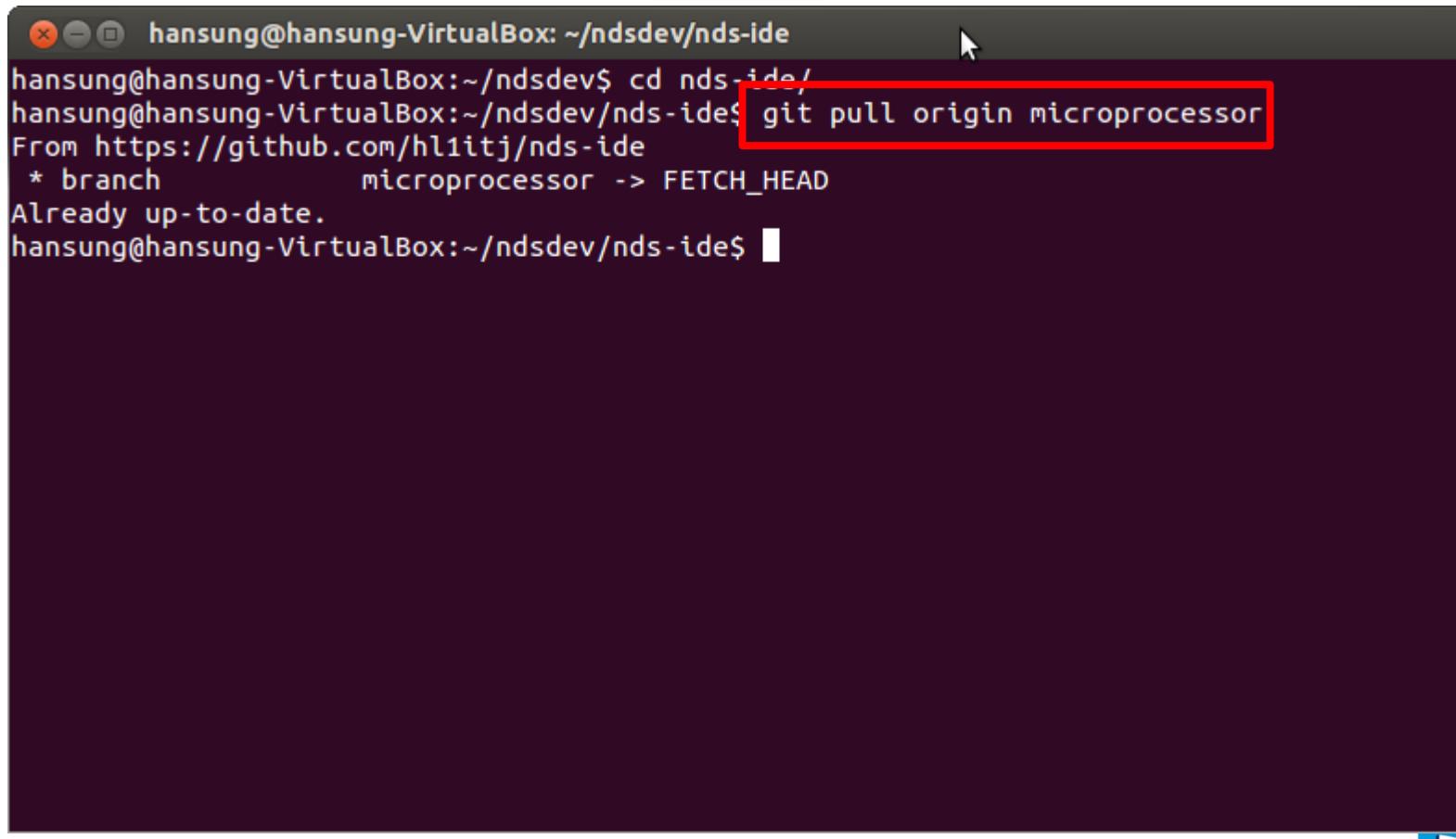
- 터미널에서 `git checkout` 명령어를 이용하여 브랜치 변경
(실습을 위한 소스 코드는 microprocessor를 이용)



```
hansung@hansung-VirtualBox: ~/ndsdev/nds-ide
hansung@hansung-VirtualBox:~/ndsdev$ cd nds-ide/
hansung@hansung-VirtualBox:~/ndsdev/nds-ide$ git checkout microprocessor
Branch microprocessor set up to track remote branch microprocessor from origin.
Switched to a new branch 'microprocessor'
hansung@hansung-VirtualBox:~/ndsdev/nds-ide$
```

실습 및 과제 소스 다운로드

- 터미널에서 git pull 명령어를 이용하여 업데이트
(원격 저장소의 변경 사항을 로컬 저장소의 이력과 합침)



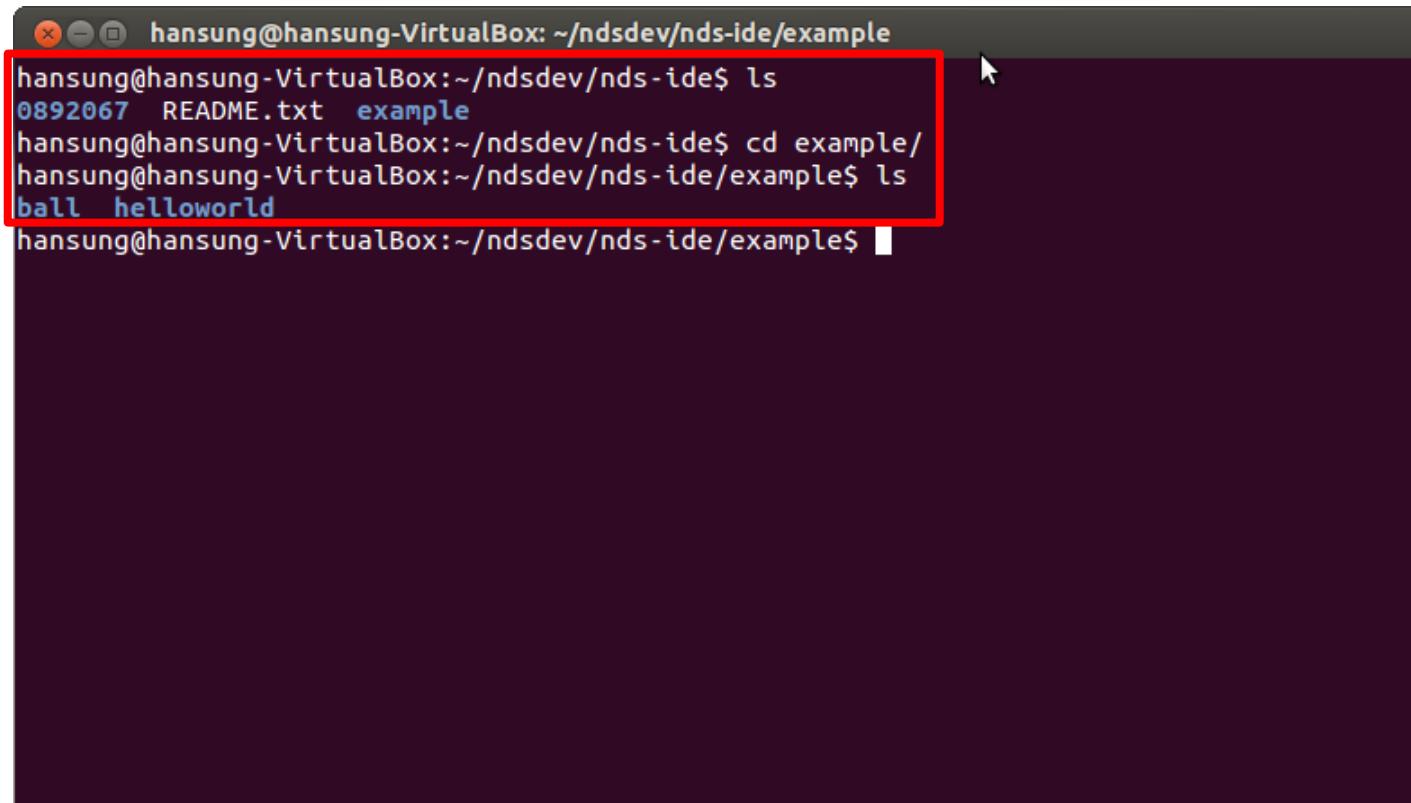
A screenshot of a terminal window titled "hansung@hansung-VirtualBox: ~/ndsdev/nds-ide". The window contains the following text:

```
hansung@hansung-VirtualBox:~/ndsdev/nds-ide$ cd nds-ide/
hansung@hansung-VirtualBox:~/ndsdev/nds-ide$ git pull origin microprocessor
From https://github.com/hl1itj/nds-ide
 * branch            microprocessor  -> FETCH_HEAD
  Already up-to-date.
hansung@hansung-VirtualBox:~/ndsdev/nds-ide$ █
```

The command "git pull origin microprocessor" is highlighted with a red rectangle.

실습 및 과제 소스 다운로드

- Git을 통해 원격 저장소에서 소스를 다운로드 받으면, 다음과 같이 helloworld 예제가 있는 것을 확인할 수 있음



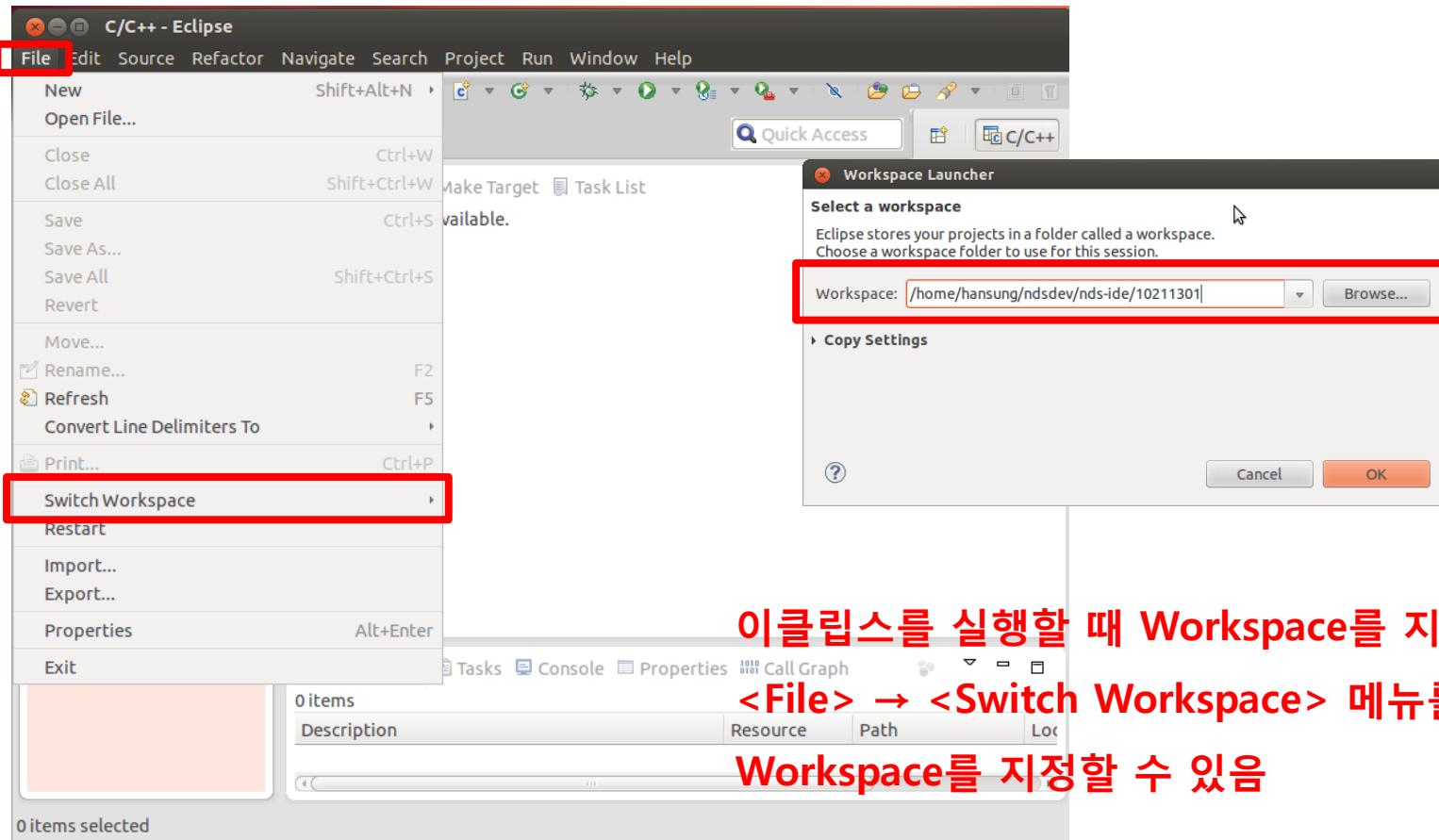
The screenshot shows a terminal window with the following command history:

```
hansung@hansung-VirtualBox: ~/ndsdev/nds-ide/example
hansung@hansung-VirtualBox:~/ndsdev/nds-ide$ ls
0892067 README.txt example
hansung@hansung-VirtualBox:~/ndsdev/nds-ide$ cd example/
hansung@hansung-VirtualBox:~/ndsdev/nds-ide/example$ ls
ball helloworld
hansung@hansung-VirtualBox:~/ndsdev/nds-ide/example$
```

The output of the first `ls` command is highlighted with a red rectangle.

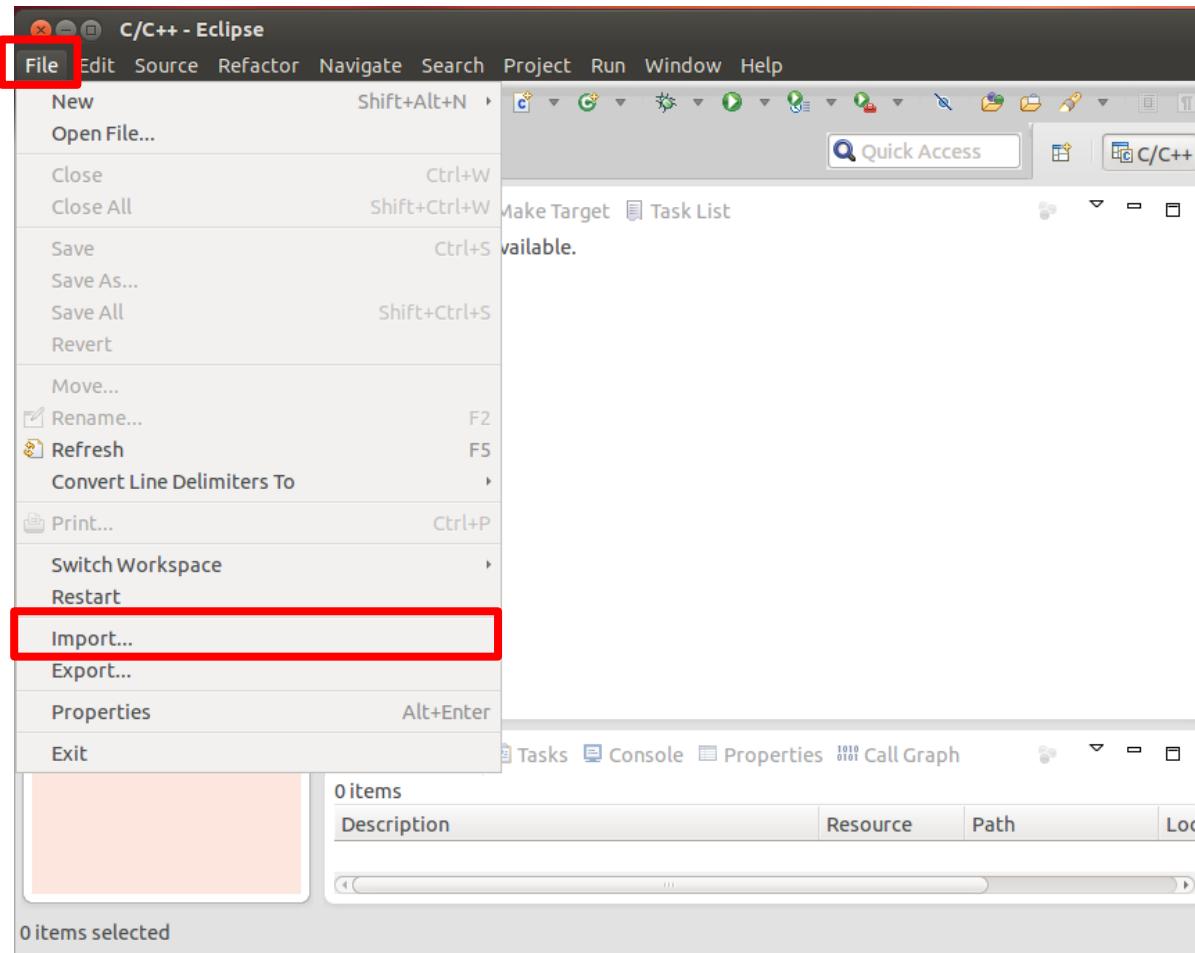
프로젝트 생성 (helloworld 예제 실습)

- 모든 실습 결과물은 Github Repository에 자신의 학번으로 관리하기 때문에, 이클립스의 Workspace를 /home/hansung/ndsdev/nds-ide/[학번] 으로 지정



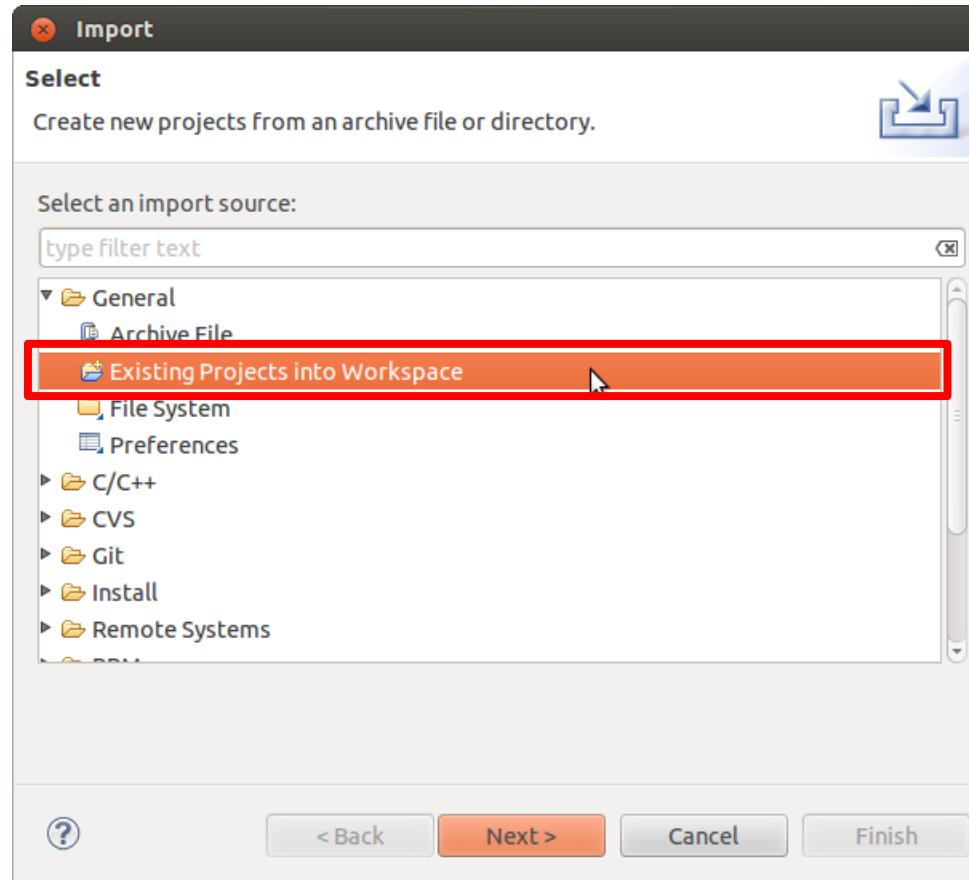
프로젝트 생성 (helloworld 예제 실습)

- <File> → <Import>를 클릭하여 hello world 예제를 불러옴



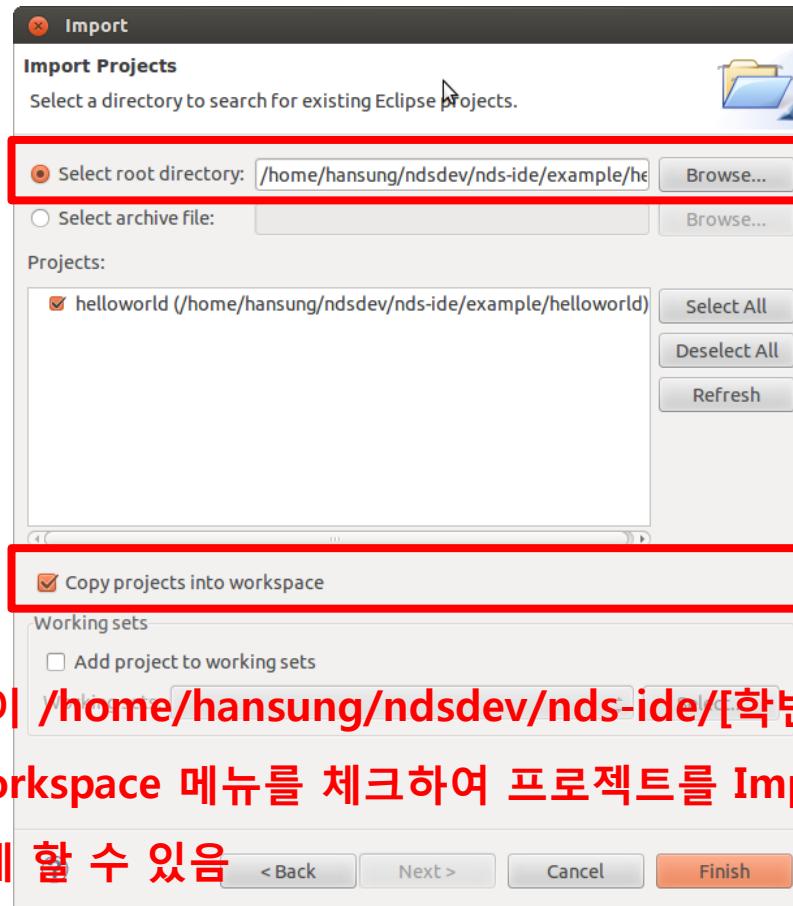
프로젝트 생성 (helloworld 예제 실습)

- General 항목의 Existing Projects into Workspace를 선택



프로젝트 생성 (helloworld 예제 실습)

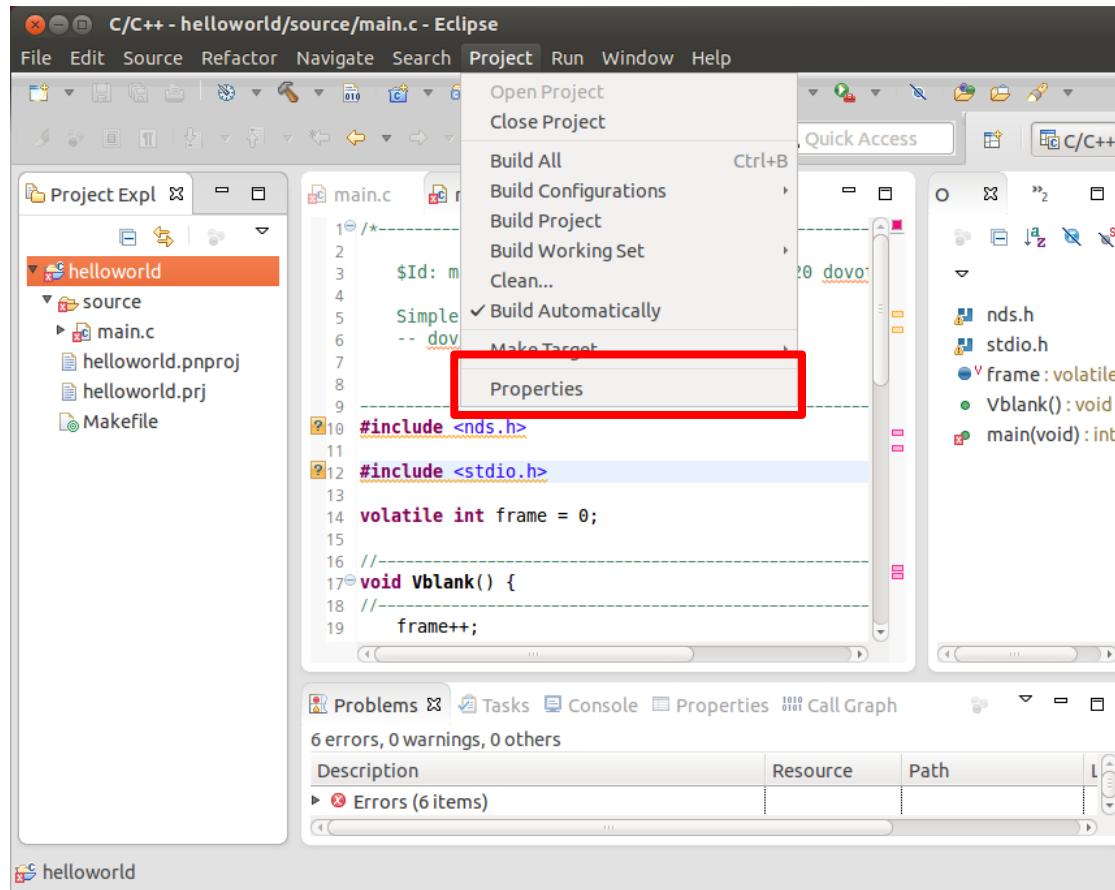
- Select root directory 항목의 Browse 버튼을 클릭하여 Git을 통해 다운로드 받은 helloworld 예제의 경로를 선택 (본 과정에서는 /home/hansung/ndsdev/nde-ide/에 다운로드 되어 있음)



Workspace를 앞과 같이 /home/hansung/ndsdev/nde-ide/[학번]으로 지정한 경우,
Copy projects into workspace 메뉴를 체크하여 프로젝트를 Import 하면
프로젝트 관리를 편하게 할 수 있음

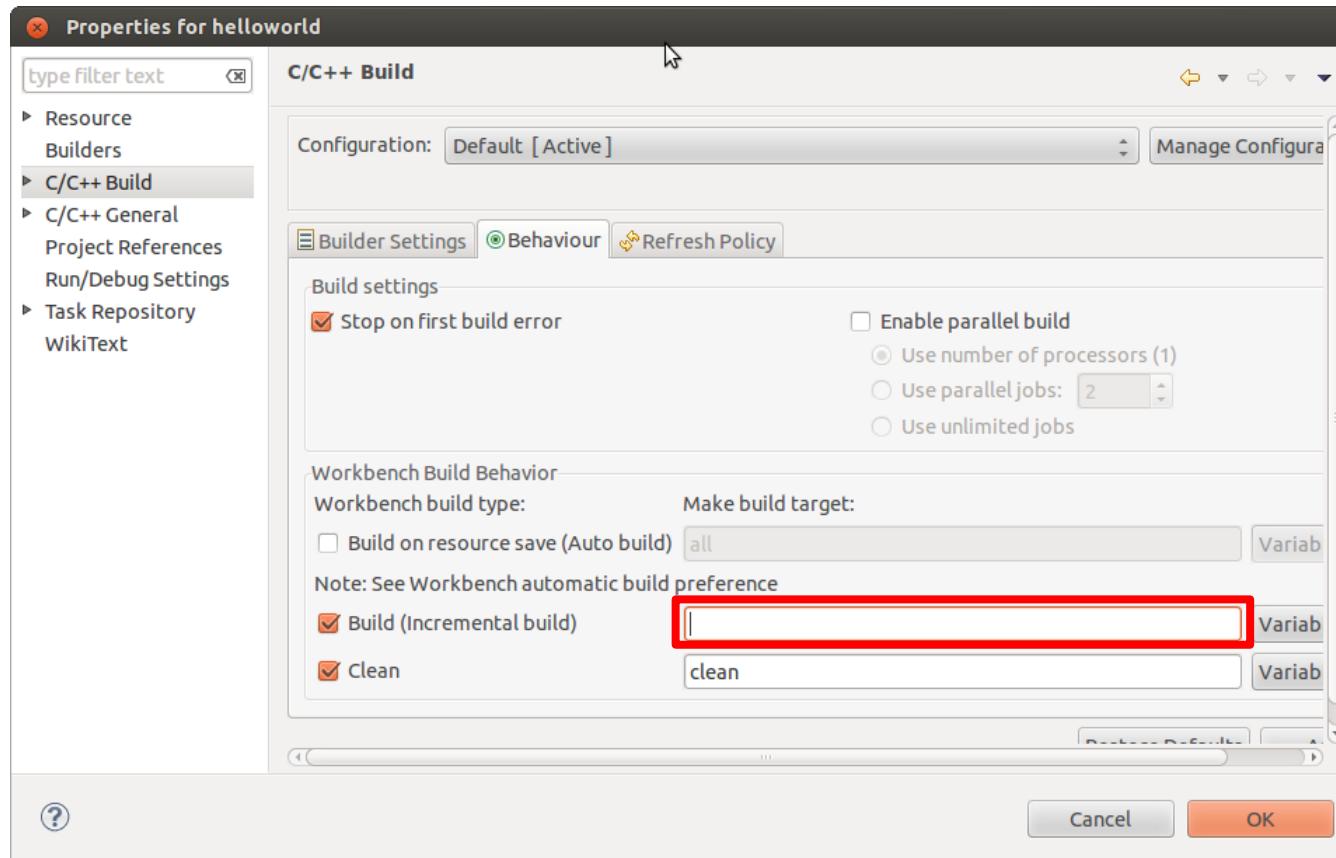
프로젝트 설정 (helloworld 예제 실습)

- <Project> → <Properties> 메뉴를 선택



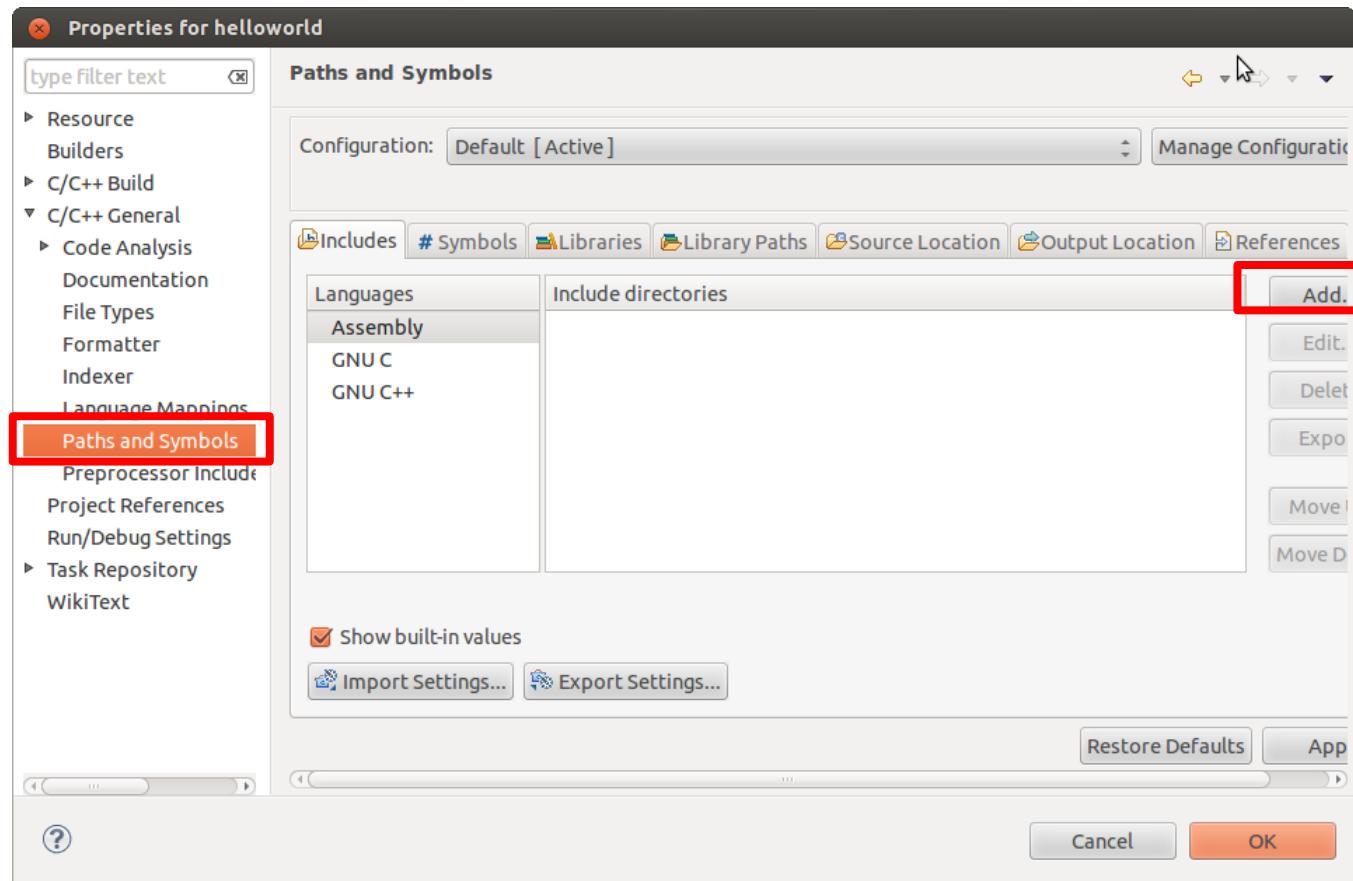
프로젝트 설정 (helloworld 예제 실습)

- <Project> → <Properties>를 클릭하고, C/C++ Build의 Behaviour 탭에서 Build(Incremental build)를 비움



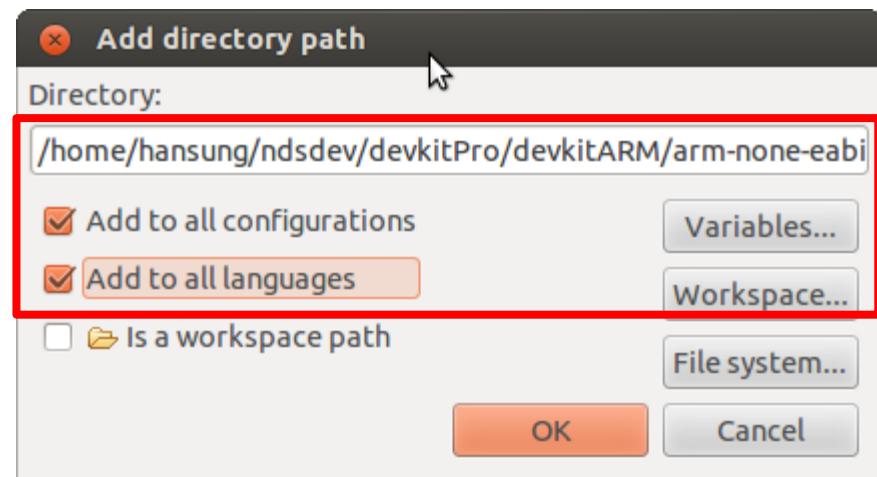
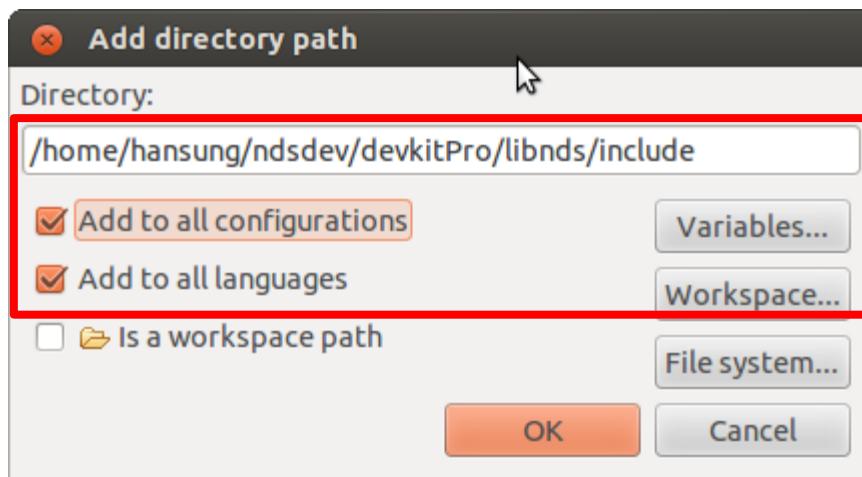
프로젝트 설정 (helloworld 예제 실습)

- <Project> → <Properties>의 C/C++ General의 Paths and Symbols 항목 선택하고, Include 탭에서 Add 버튼을 클릭



프로젝트 설정 (helloworld 예제 실습)

- 아래 그림과 같이 닌텐도 DS Library, ARM Compiler의 Include 경로를 추가
(Is a workspace path를 제외한 위의 두 체크 박스를 체크)

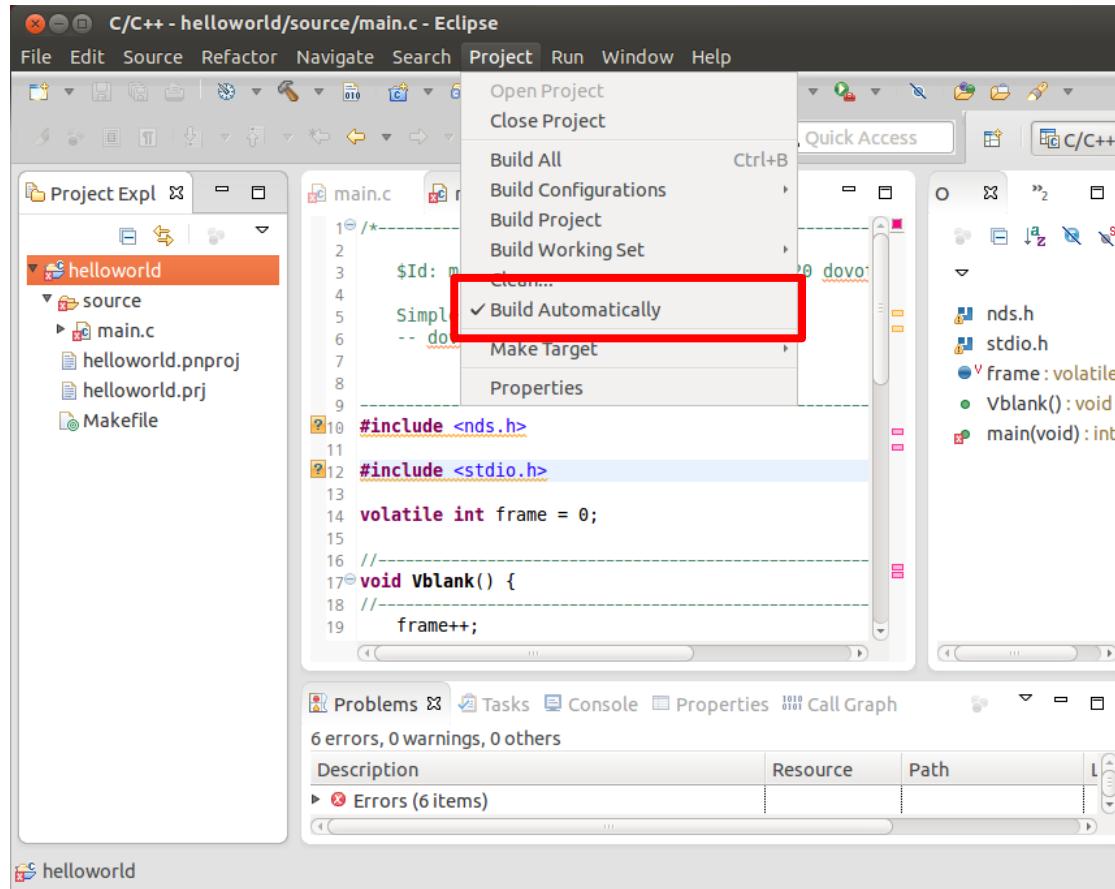


`/home/hansung/ndsdev/devkitPro/libnds/include`

`/home/hansung/ndsdev/devkitPro/devkitARM/arm-none-eabi/include`

프로젝트 설정 (helloworld 예제 실습)

- <Project> → <Build Automatically>를 클릭하여 해제



프로젝트 설정 (helloworld 예제 실습)

- helloworld는 매우 간단한 프로그램이라 nds 라이브러리만 필요로 하지만, 무선랜, 파일 입출력을 위해서는 Makefile을 해당 라이브러리가 추가되도록 수정하여야 함
Makefile의 “LIBS := -lnds9” 을 “**LIBS := -lswifi9 -lfat -lnds9**”로 수정



프로젝트 설정 (helloworld 예제 실습)

- 다음과 같이 환경 변수를 설정

The screenshot shows two terminal windows. The top window is run as root, with the command `vi .bashrc` highlighted with a red box. The bottom window is run as a regular user (hansung), showing the contents of the .bashrc file. A large red box highlights the last two lines of the file:

```
# 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 [ -f /etc/bash_completion ] && ! shopt -oq posix; then
    . /etc/bash_completion
fi

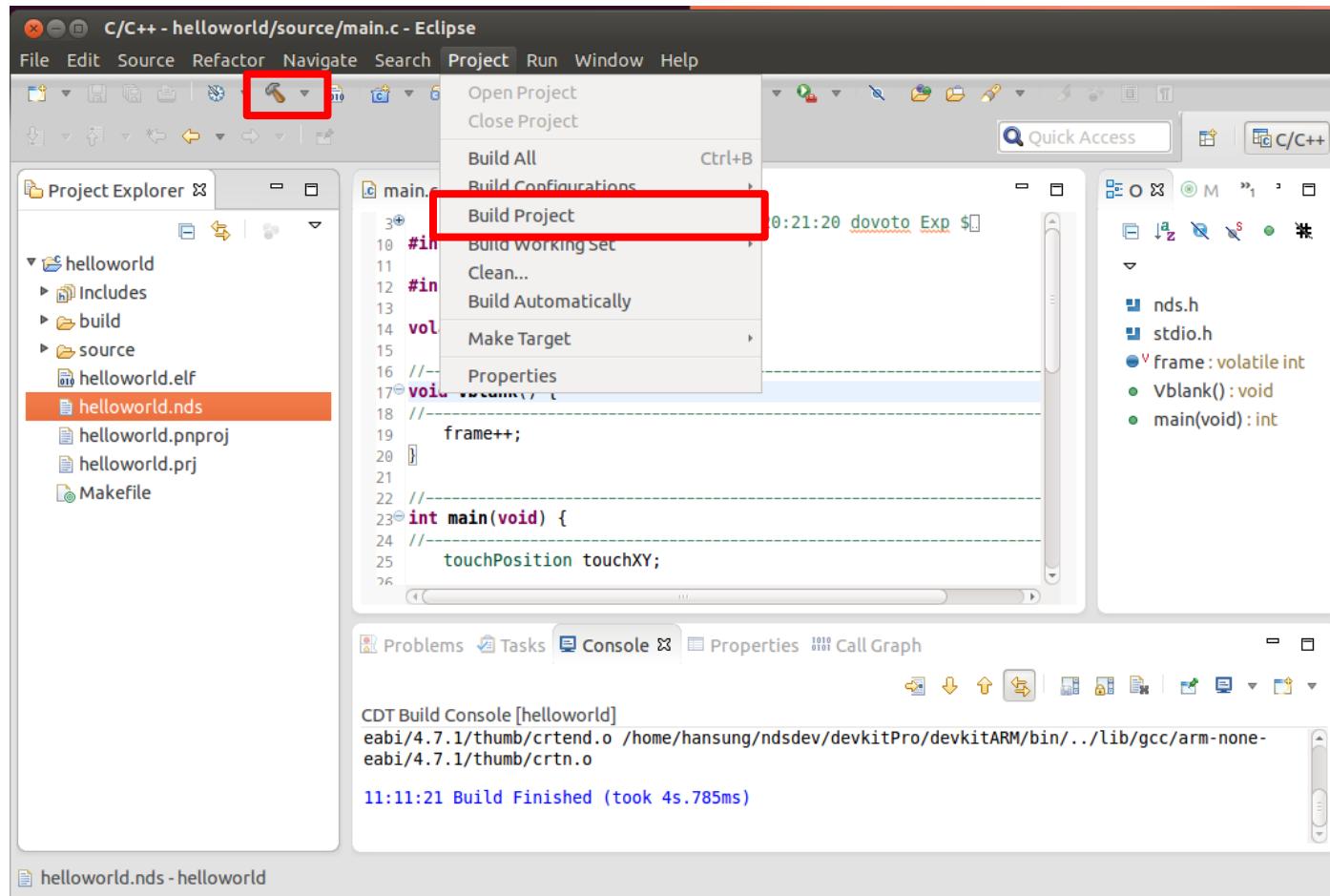
export PATH=$PATH:/home/hansung/ndsdev/devkitPro/devkitARM/bin:/home/hansung/nds
dev/eclipse

export DEVKITPRO=/home/hansung/ndsdev/devkitPro
export DEVKITARM=$DEVKITPRO/devkitARM
```

The message at the bottom of the terminal indicates 113 lines and 3668 characters were written to the file.

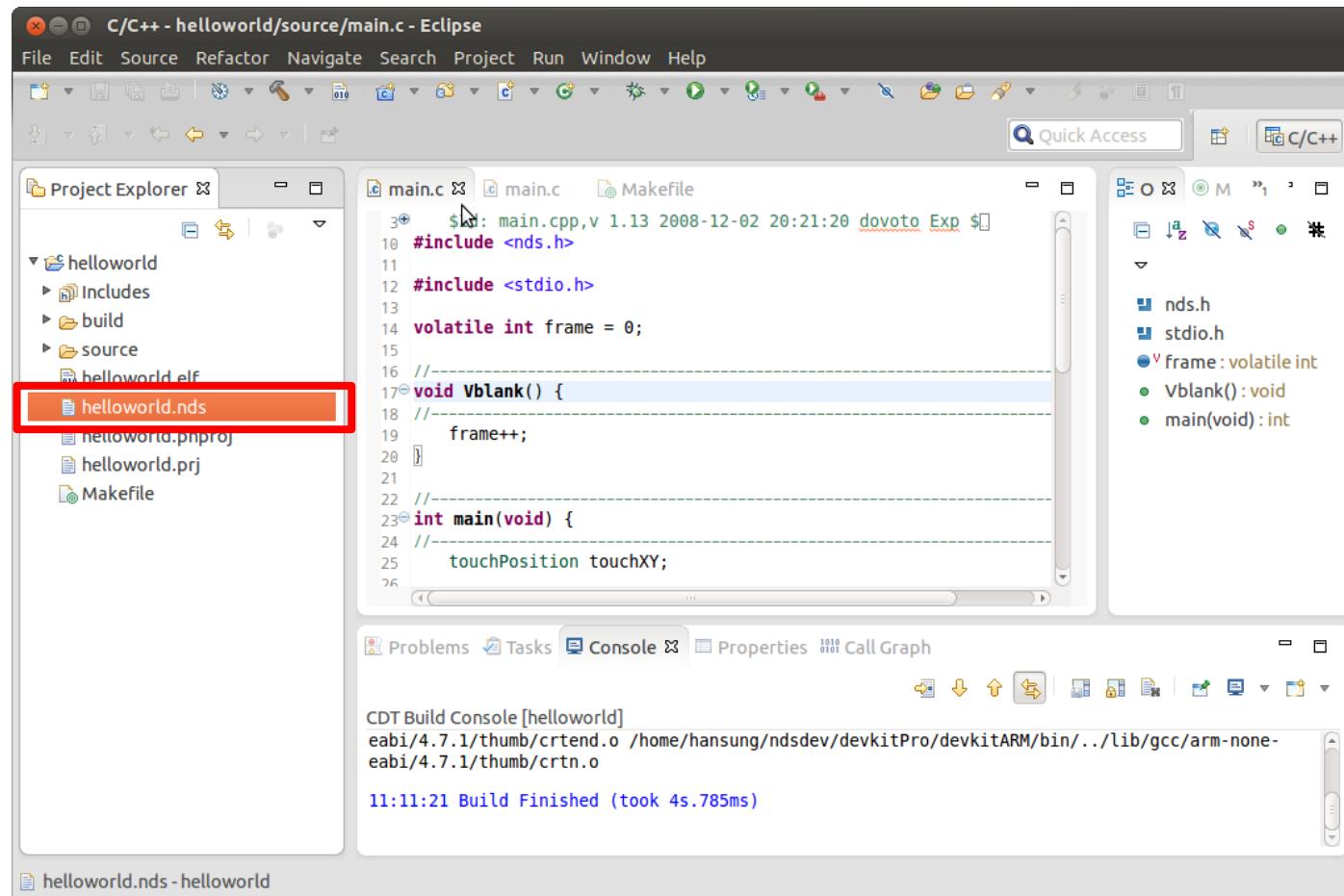
프로젝트 빌드 (helloworld 예제 실습)

- <Project> → <Build Project>를 클릭하거나 툴바를 이용하여 프로젝트를 빌드



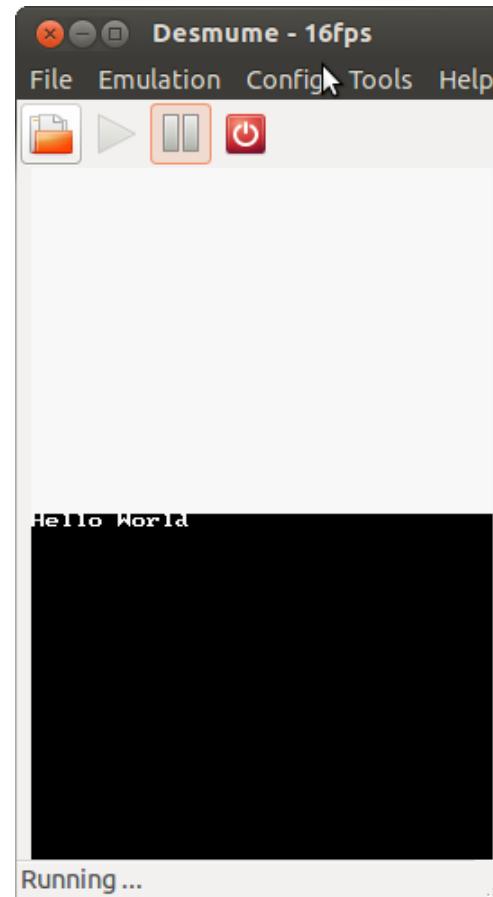
프로젝트 빌드 (helloworld 예제 실습)

- 다음과 같이 성공적으로 빌드 되어, helloworld.nds 파일이 생성되면 빌드 완료



에뮬레이터를 사용하여 실행

- 에뮬레이터를 이용하여 프로젝트를 빌드하여 생성된 파일을 실행



다운로드 및 실행

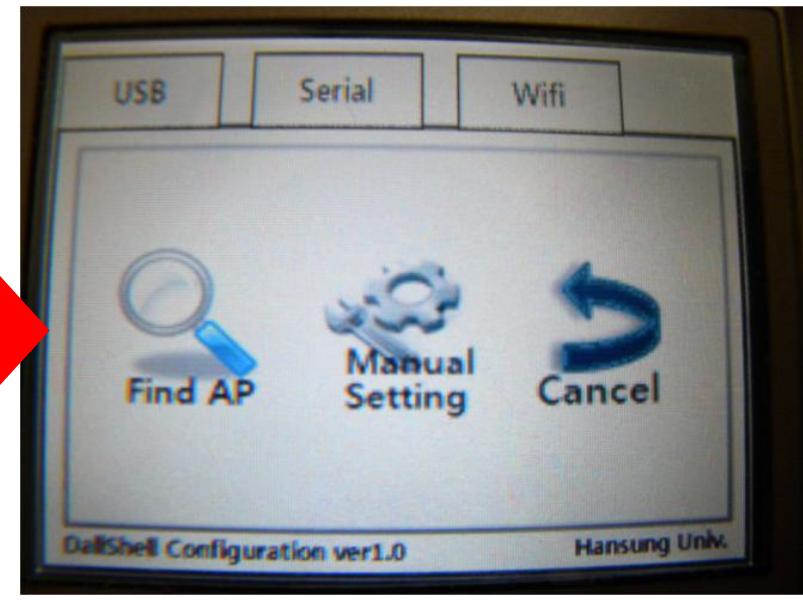
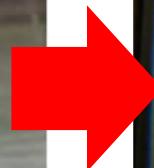
- 다운로드 프로그램이 현재 2개 버전으로 제공되고 있음
 - PC에서 서버를 거쳐서 닌텐도에 파일을 전송할 수 있는 버전 (PC → Server → NDS)
 - bin/down_pc : PC에서 서버로 파일을 전송하는 프로그램
 - bin/down_nds : 닌텐도DS에서 서버로부터 파일을 전송받는 프로그램
 - PC에서 직접 닌텐도에 파일을 전송할 수 있는 버전 (PC → NDS)
 - bin/wifidown_pc : PC에서 닌텐도DS로 파일을 전송하는 프로그램
 - bin/wifidown_nds : 닌텐도DS에서 PC로부터 파일을 전송받는 프로그램

[학교에서 실습할 경우 이 프로그램을 사용]

※ PC와 닌텐도가 같은 AP상에 있어야 함

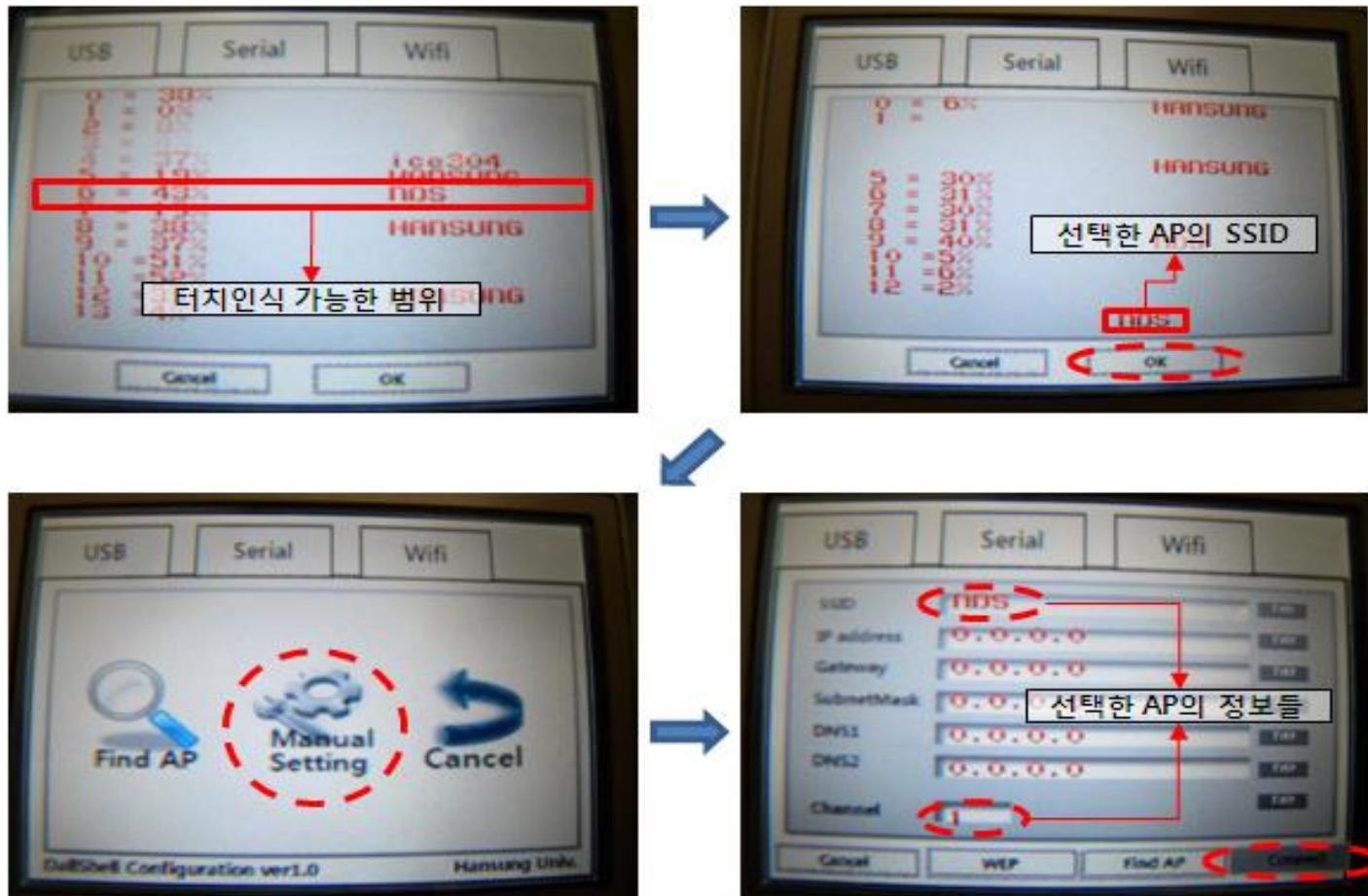
다운로드 설정 프로그램 사용 방법

- 다운로드 설정 프로그램을 사용하여 무선 네트워크 환경 설정



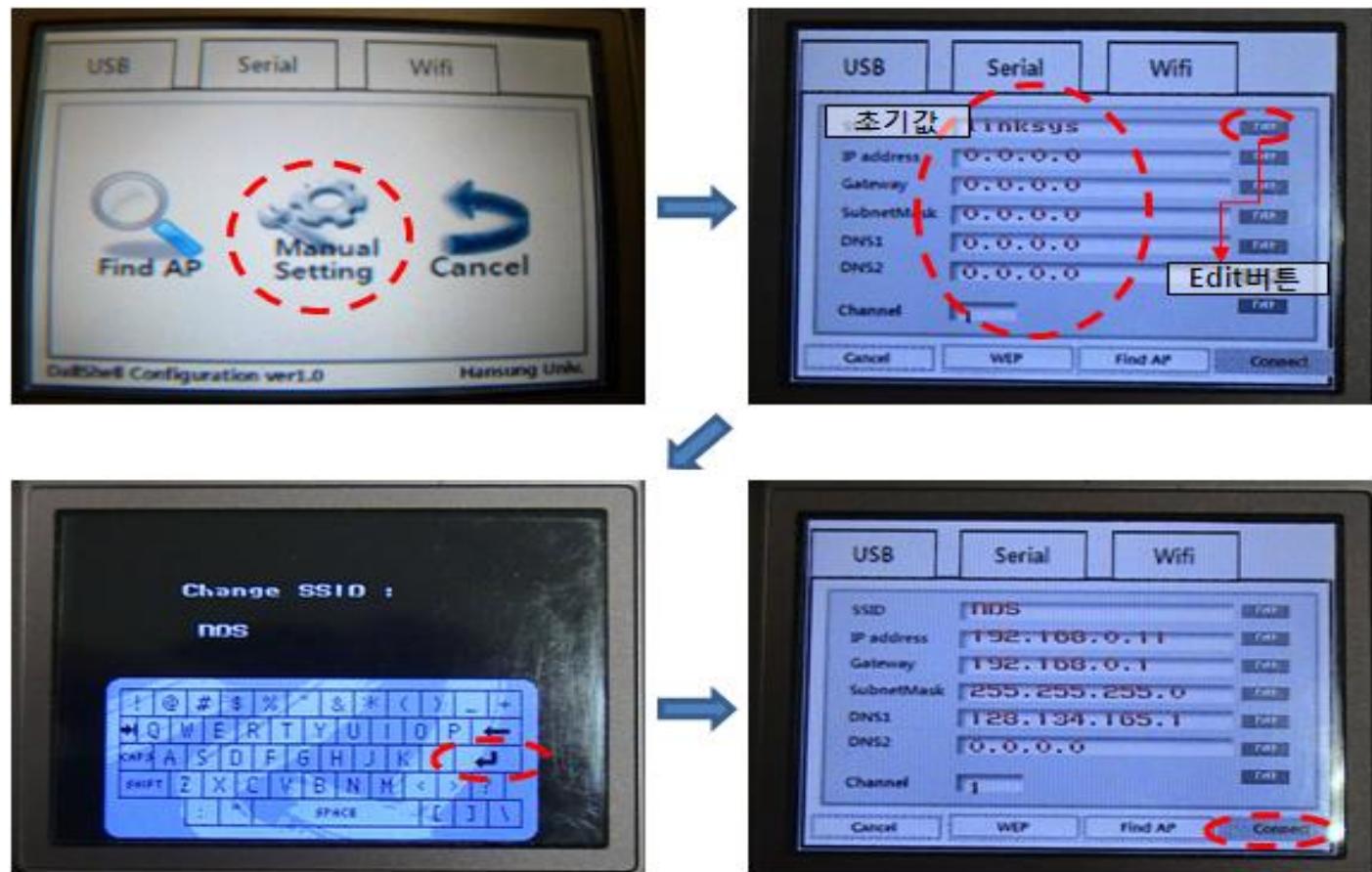
다운로드 설정 프로그램 사용 방법

- 자동 네트워크 검색 (Find AP)을 통한 설정 방법



다운로드 설정 프로그램 사용 방법

- 수동 네트워크 설정 (Manual Setting)을 통한 설정 방법



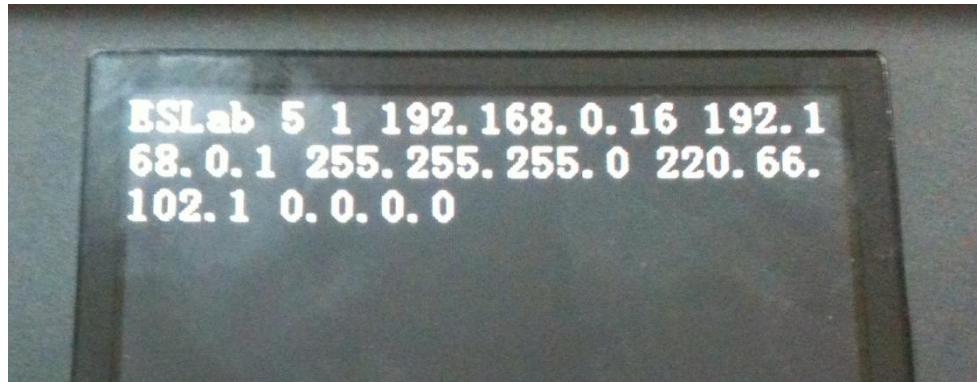
다운로드 설정 프로그램 사용 방법

- 자동 또는 수동으로 설정한 뒤, Connect 버튼을 누르면 다음과 같은 화면이 나옴
Save Setting을 눌러 설정 사항을 저장한 뒤, 닌텐도 DS를 다시 구동



다운로드 설정 프로그램 사용 방법

- 무선 네트워크 정보는 DallShell_Conf.txt에 저장되며, 저장된 정보는 AP, SSID, SSID길이, 채널, IP, Gateway, Subnet Mask, DNS1, DNS2 순서로 저장된다.
(닌텐도 DS를 구동한 뒤, Select 버튼을 눌러 파일 탐색기 화면의 모드를 바꿔 해당 파일을 찾을 수 있음)



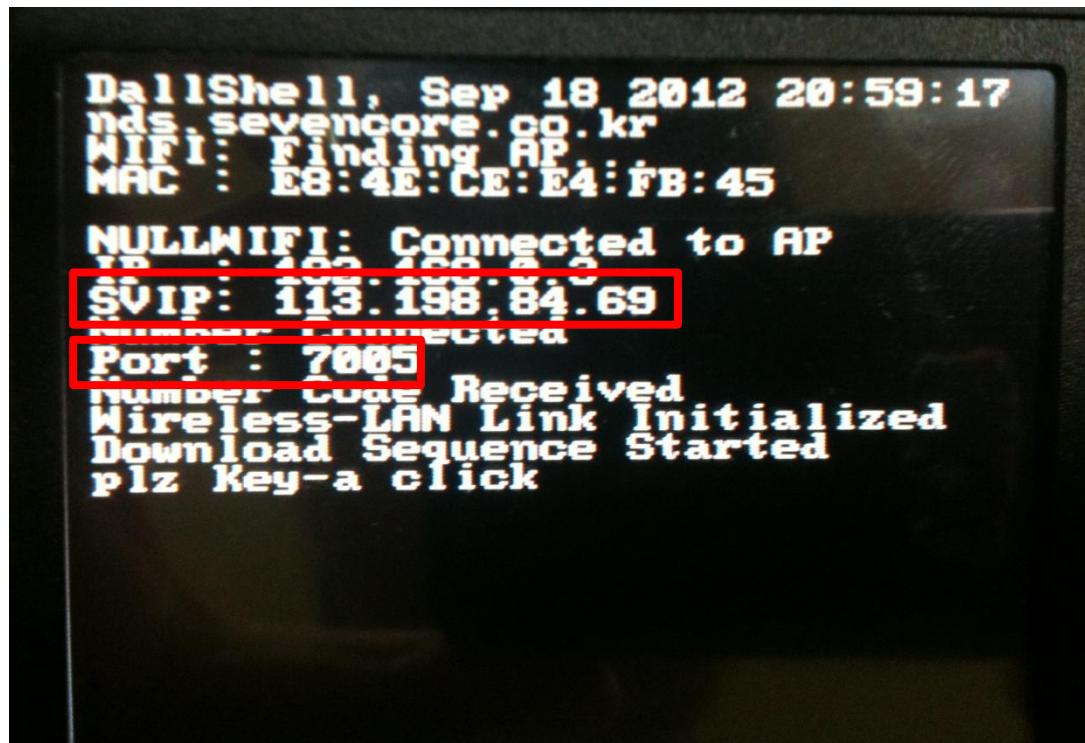
※ 그림과 같이 위에 설명된 순서에 맞게 네트워크 정보가 제대로 입력되었다면, 다음 슬라이드의 다운로드 프로그램 사용에 대한 실습을 진행하고, IP가 0.0.0.0 또는 255.255.255.255 와 같이 잘못 저장되었다면, 다운로드 설정 프로그램을 다시 실행하여 무선 네트워크를 설정 한다.

다운로드 및 실행 (PC → Server → NDS)

- 닌텐도 DS에서 down_nds.nds 파일을 실행

(아래 그림에서 표시된 부분의 서버 IP와 Port를 이용하여 다운로드를 진행)

※ 프로그램을 실행시킬 때마다 Port가 다르게 설정되니, 각자의 LCD 화면에 표시된 포트를 이용해야 함

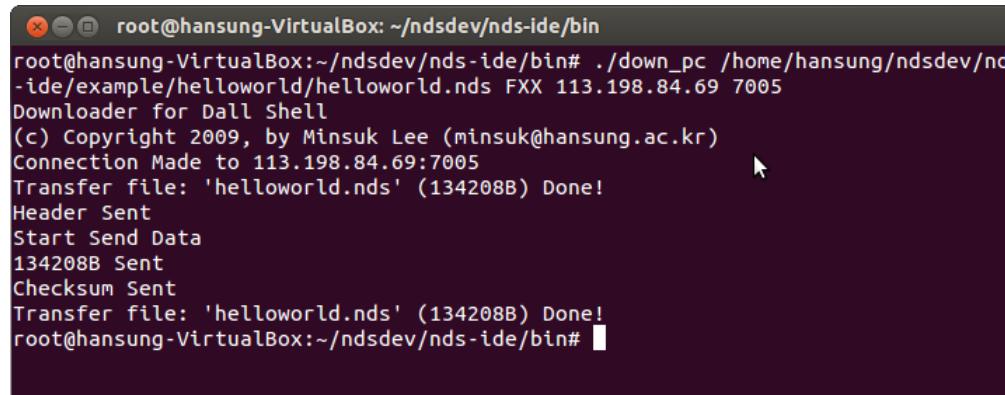


다운로드 및 실행 (PC → Server → NDS)

- 터미널에서 down_pc 파일을 실행하여 서버로 파일을 전송

[사용방법 : down_pc | 파일경로 | FXX | IP | Port]

down_pc /home/hansung/ndsdev/nds-ide/10211301/helloworld/helloworld.nds
FXX 113.198.84.69 7005



```
root@hansung-VirtualBox:~/ndsdev/nds-ide/bin
root@hansung-VirtualBox:~/ndsdev/nds-ide/bin# ./down_pc /home/hansung/ndsdev/nds-ide/example/helloworld/helloworld.nds FXX 113.198.84.69 7005
Downloader for Dell Shell
(c) Copyright 2009, by Minsuk Lee (minsuk@hansung.ac.kr)
Connection Made to 113.198.84.69:7005
Transfer file: 'helloworld.nds' (134208B) Done!
Header Sent
Start Send Data
134208B Sent
Checksum Sent
Transfer file: 'helloworld.nds' (134208B) Done!
root@hansung-VirtualBox:~/ndsdev/nds-ide/bin#
```

※ 학교에서는 서버를 연구실에 항상 실행시켜놓기 때문에, 서버 IP 주소를 위 주소 그대로 사용하고, 포트는 프로그램이 실행되는 기기마다 다르게 부여되기 때문에 닌텐도 DS의 LCD화면에 표시되는 포트를 사용하도록 함

다운로드 및 실행 (PC → Server → NDS)

- PC에서 서버로 파일 전송이 끝났다면,
닌텐도 DS에서 A 버튼을 누르면 다운로드가 진행 됨

```
root@hansung-VirtualBox: ~/ndsdev/nds-ide/bin
root@hansung-VirtualBox:~/ndsdev/nds-ide/bin# ./down_pc /home/hansung/ndsdev/nds-ide/example/helloworld/helloworld.nds FXX 113.198.84.69 7005
Downloader for Dall Shell
(c) Copyright 2009, by Minsuk Lee (minsuk@hansung.ac.kr)
Connection Made to 113.198.84.69:7005
Transfer file: 'helloworld.nds' (134208B) Done!
Header Sent
Start Send Data
134208B Sent
Checksum Sent
Transfer file: 'helloworld.nds' (134208B) Done!
root@hansung-VirtualBox:~/ndsdev/nds-ide/bin#
```

PC에서 서버로 파일 전송 완료

```
DallShell, Sep 18 2012 20:59:17
nds sevencore.co.kr
WIFI: Finding AP
MAC : E8:4E:CE:E4:FB:45
NULLWIFI: Connected to AP
IP : 192.168.0.3
SVIP: 113.198.84.69
Number Connected
Port : 7005
Number Code Received
Wireless-LAN Link Initialized
Download Sequence Started
plz Key-a click
```

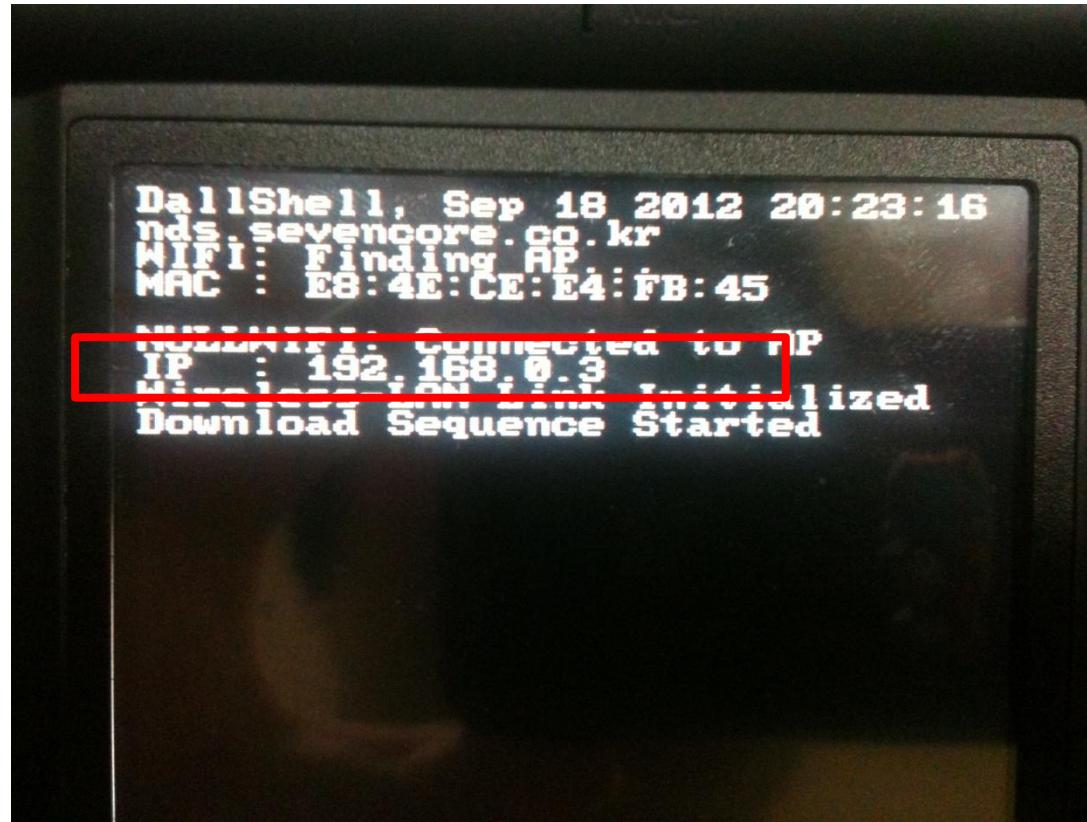
A 버튼을 누름

서버에서 닌텐도 DS로 파일 전송 완료

```
WIFI: Finding AP
MAC : E8:4E:CE:E4:FB:45
NULLWIFI: Connected to AP
IP : 192.168.0.3
SVIP: 113.198.84.69
Number Connected
Port : 7005
Number Code Received
Wireless-LAN Link Initialized
Download Sequence Started
Host Connected
Magic Code Received
Header Received
helloworld.nds (134208B)
Download (Byte) : 134208
Download All File Data
Download Checksum OK
Download Done
Download Successfully Done
```

다운로드 및 실행 (PC → NDS)

- 닌텐도 DS에서 wifidown_nds.nds 파일을 실행
(아래 그림에서 표시된 부분의 IP를 이용하여 다운로드를 진행)



다운로드 및 실행 (PC → NDS)

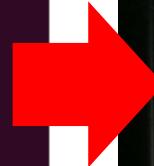
- 터미널에서 wifidown_pc 파일을 실행하여 서버로 파일을 전송

[사용방법 : wifidown_pc | t | 파일경로 | FXX | IP | Port] [Default Port : 3389]

wifidown_pc t /home/hansung/ndsdev/nds-ide/example/helloworld/helloworld.nds

FXX 192.168.0.3 3389

```
root@hansung-VirtualBox: ~/ndsdev/nds-ide/bin
root@hansung-VirtualBox:~/ndsdev/nds-ide/bin# ./wifidown_pc t /home/hansung/ndsdev/nds-ide/example/helloworld.nds FXX 192.168.0.3 3389
Downloader for Dall Shell
(c) Copyright 2009, by Minsuk Lee (minsuk@hansung.ac.kr)
socket successMagic Download Code Sent
Header Sent
Start Send Data
134208B Sent
checksum: 10119648
Checksum Sent
Magic Cleanup Code Sent
Transfer file: 'helloworld.nds' (134208B) Done!
root@hansung-VirtualBox:~/ndsdev/nds-ide/bin#
```



```
DallShell, Sep 18 2012 20:23:16
nds.sevencore.co.kr
WIFI: Finding AP
MAC : E8:4E:CE:E4:FB:45

NULLWIFI: Connected to AP
IP : 192.168.0.3
Wireless-LAN Link Initialized
Download Sequence Started
Host Connected
Magic Code Received
Header Received
helloworld.nds (134208B)
Download (Byte) : 134208
Download All File Data
Download Checksum OK
Download Done
Download Successfully Done
Now What to do ? - LOOP !
Wireless-LAN Link Initialized
Download Sequence Started
```

PC에서 닌텐도 DS로 파일 전송 완료

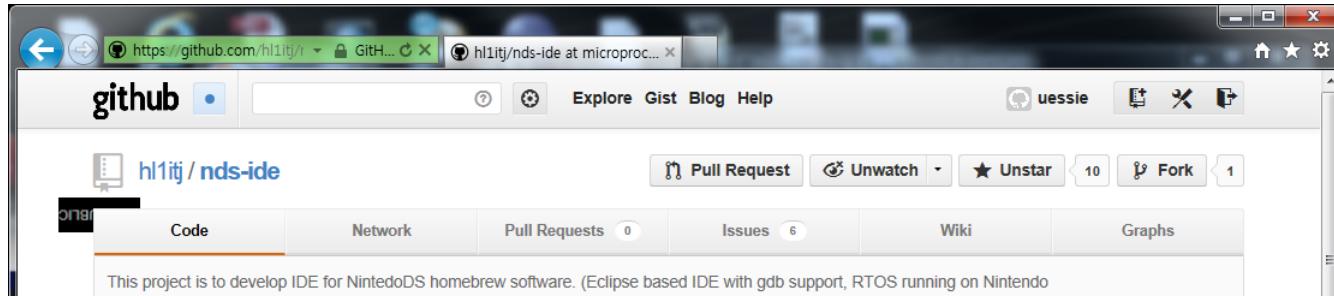
다운로드 및 실행

- 다운로드가 완료되었다면 닌텐도 DS를 다시 구동 하여, 닌텐도 DS에 다운로드 된 helloworld.nds 파일을 실행



실습 및 과제 소스 업로드

- 실습 및 과제 소스, 결과보고서는 Github의 저장소에 올립
(실습에 대한 평가는 해당 저장소에 업로드 된 제출물로 이루어짐)



각자의 결과물은 Github 사이트의 Repository에

(<https://github.com/hl1itj/nds-ide>)

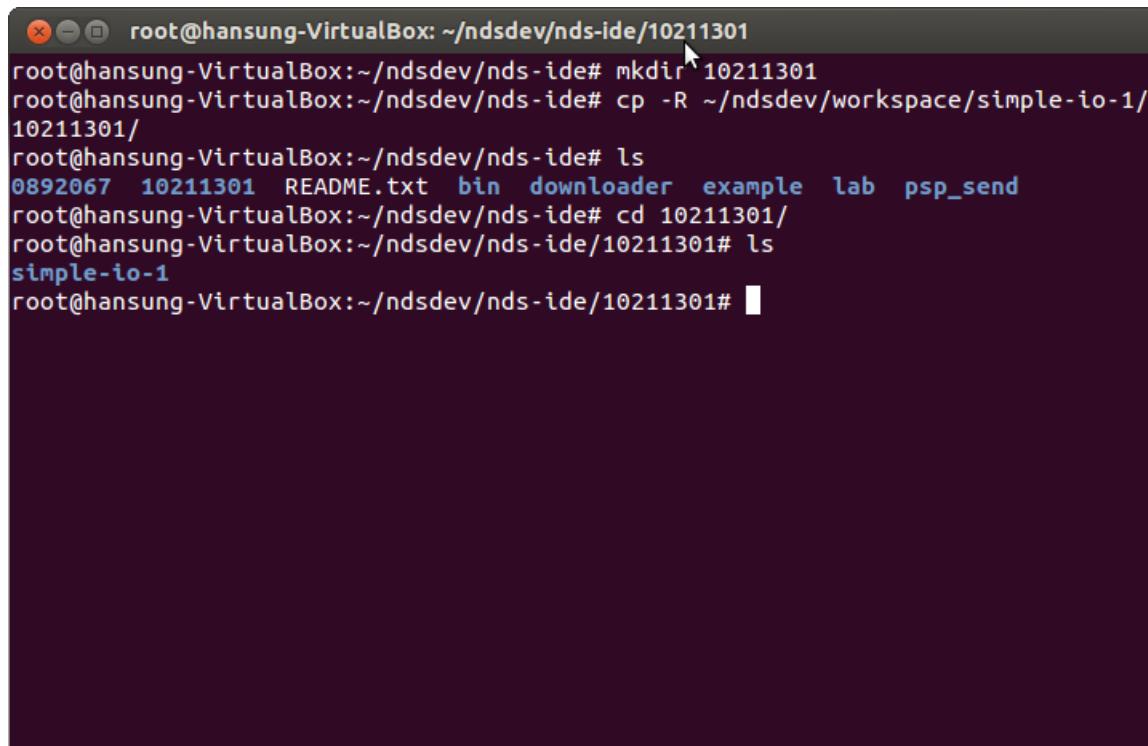
microprocessor 브랜치를 사용하여

아래 그림과 같이 학번/simple-io-1, 학번/simple-io-2 와 같이 업로드 해야 함



실습 및 과제 소스 업로드

- Github 저장소를 복사(Clone)한 뒤, microprocessor 브랜치로 변경하여, 각자의 학번 디렉토리를 만들고 해당 학번 디렉토리에 각 실험의 소스를 복사



```
root@hansung-VirtualBox: ~/ndsdev/nds-ide/10211301
root@hansung-VirtualBox:~/ndsdev/nds-ide# mkdir 10211301
root@hansung-VirtualBox:~/ndsdev/nds-ide# cp -R ~/ndsdev/workspace/simple-io-1/
10211301/
root@hansung-VirtualBox:~/ndsdev/nds-ide# ls
0892067 10211301 README.txt bin downloader example lab psp_send
root@hansung-VirtualBox:~/ndsdev/nds-ide# cd 10211301/
root@hansung-VirtualBox:~/ndsdev/nds-ide/10211301# ls
simple-io-1
root@hansung-VirtualBox:~/ndsdev/nds-ide/10211301#
```

현재 **nds-ide/10211301/simple-io-1**에 첫번째 실험의 결과물이 포함되어 있는 상태

실습 및 과제 소스 업로드

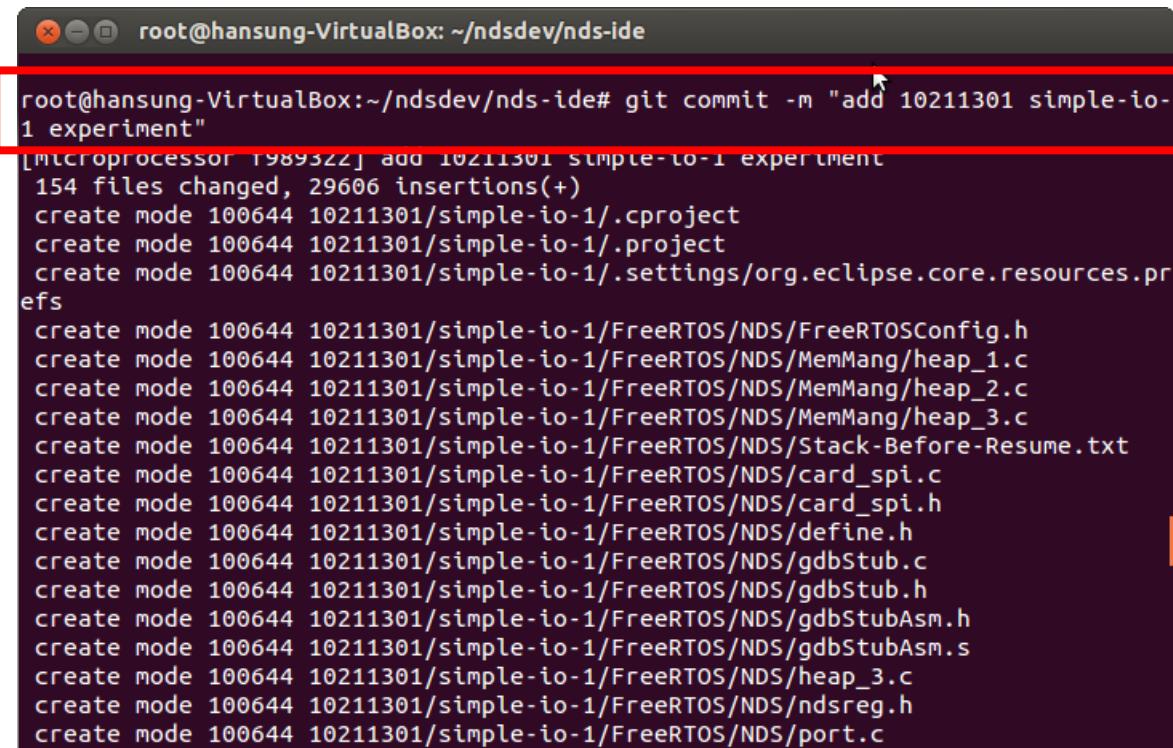
- git add 명령어를 이용하여 실습 결과물을 추가
git status 명령어를 이용하여 변경사항을 확인할 수 있음

The screenshot shows a terminal window titled "root@hansung-VirtualBox: ~/ndsdev/nds-ide". The user has navigated to the directory containing a project named "10211301". They run "git add 10211301/" followed by "git status". The output shows a list of new files to be committed, which are all files within the "10211301/simple-io-1" directory, including C source files like FreeRTOSConfig.h, heap_1.c, and card_spi.c, as well as configuration and header files.

```
root@hansung-VirtualBox:~/ndsdev/nds-ide/10211301# cd ..
root@hansung-VirtualBox:~/ndsdev/nds-ide# ls
0002027_10211301 README.txt bin_downloader example.lib psp_send
root@hansung-VirtualBox:~/ndsdev/nds-ide# git add 10211301/
root@hansung-VirtualBox:~/ndsdev/nds-ide# git status
# On branch master
# Changes to be committed:
#   (use "git reset HEAD <file>" to unstage)
#
#       new file:   10211301/simple-io-1/.cproject
#       new file:   10211301/simple-io-1/.project
#       new file:   10211301/simple-io-1/.settings/org.eclipse.core.resources.pr
efs
#       new file:   10211301/simple-io-1/FreeRTOS/NDS/FreeRTOSConfig.h
#       new file:   10211301/simple-io-1/FreeRTOS/NDS/MemMang/heap_1.c
#       new file:   10211301/simple-io-1/FreeRTOS/NDS/MemMang/heap_2.c
#       new file:   10211301/simple-io-1/FreeRTOS/NDS/MemMang/heap_3.c
#       new file:   10211301/simple-io-1/FreeRTOS/NDS/Stack-Before-Resume.txt
#       new file:   10211301/simple-io-1/FreeRTOS/NDS/card_spi.c
#       new file:   10211301/simple-io-1/FreeRTOS/NDS/card_spi.h
#       new file:   10211301/simple-io-1/FreeRTOS/NDS/define.h
#       new file:   10211301/simple-io-1/FreeRTOS/NDS/gdbStub.c
#       new file:   10211301/simple-io-1/FreeRTOS/NDS/qdbStub.h
```

실습 및 과제 소스 업로드

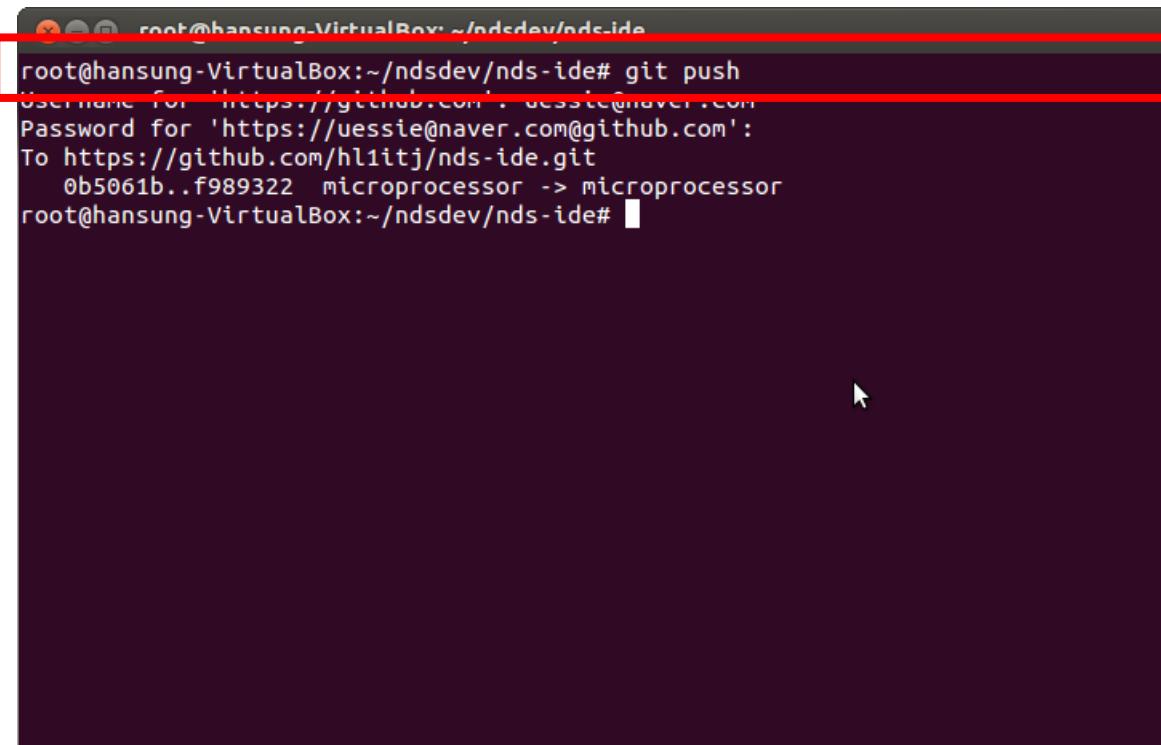
- git commit 명령어를 이용하여 변경사항 커밋



```
root@hansung-VirtualBox:~/ndsdev/nds-ide# git commit -m "add 10211301 simple-io-1 experiment"
[microprocessor 1989322] add 10211301 simple-io-1 experiment
 154 files changed, 29606 insertions(+)
  create mode 100644 10211301/simple-io-1/.cproject
  create mode 100644 10211301/simple-io-1/.project
  create mode 100644 10211301/simple-io-1/.settings/org.eclipse.core.resources.pr
efs
  create mode 100644 10211301/simple-io-1/FreeRTOS/NDS/FreeRTOSConfig.h
  create mode 100644 10211301/simple-io-1/FreeRTOS/NDS/MemMang/heap_1.c
  create mode 100644 10211301/simple-io-1/FreeRTOS/NDS/MemMang/heap_2.c
  create mode 100644 10211301/simple-io-1/FreeRTOS/NDS/MemMang/heap_3.c
  create mode 100644 10211301/simple-io-1/FreeRTOS/NDS/Stack-Before-Resume.txt
  create mode 100644 10211301/simple-io-1/FreeRTOS/NDS/card_spi.c
  create mode 100644 10211301/simple-io-1/FreeRTOS/NDS/card_spi.h
  create mode 100644 10211301/simple-io-1/FreeRTOS/NDS/define.h
  create mode 100644 10211301/simple-io-1/FreeRTOS/NDS/gdbStub.c
  create mode 100644 10211301/simple-io-1/FreeRTOS/NDS/gdbStub.h
  create mode 100644 10211301/simple-io-1/FreeRTOS/NDS/gdbStubAsm.h
  create mode 100644 10211301/simple-io-1/FreeRTOS/NDS/gdbStubAsm.s
  create mode 100644 10211301/simple-io-1/FreeRTOS/NDS/heap_3.c
  create mode 100644 10211301/simple-io-1/FreeRTOS/NDS/ndsreg.h
  create mode 100644 10211301/simple-io-1/FreeRTOS/NDS/port.c
```

실습 및 과제 소스 업로드

- **git push 명령어를 이용하여 변경사항 푸싱하기**
Github에 등록한 아이디와 패스워드를 입력하면 Github 저장소에 저장 됨

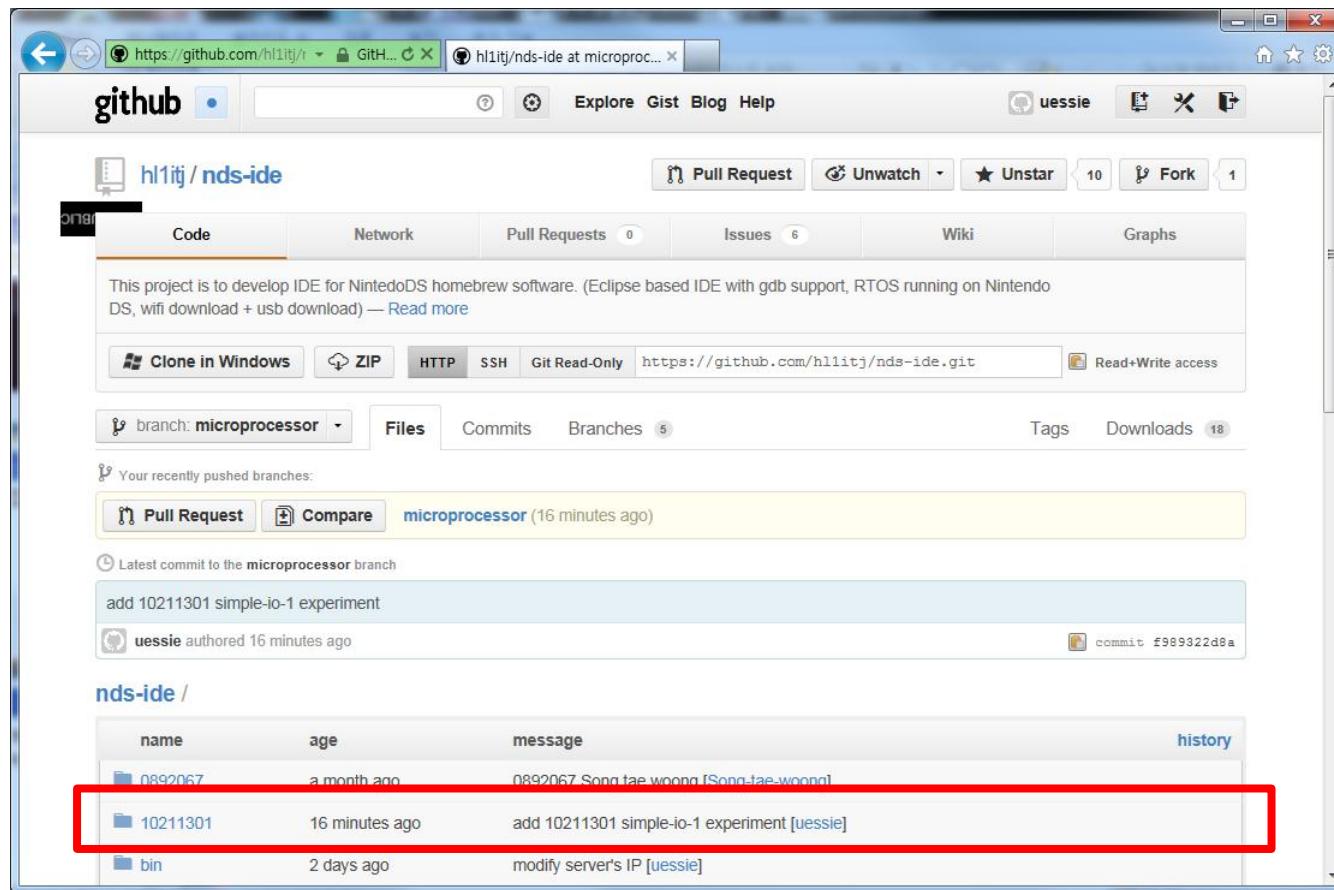


A screenshot of a terminal window on a Linux system. The window title is "root@hansung-VirtualBox:~/ndsdev/nds-ide". The user has run the command "git push". A red rectangle highlights the command line and the resulting output. The output shows the user being prompted for a GitHub username and password, and then pushing changes to a repository named "microprocessor" on GitHub.

```
root@hansung-VirtualBox:~/ndsdev/nds-ide# git push
Username for 'https://github.com': uessie@naver.com
Password for 'https://uessie@naver.com@github.com':
To https://github.com/hl1itj/nds-ide.git
  0b5061b..f989322  microprocessor -> microprocessor
root@hansung-VirtualBox:~/ndsdev/nds-ide#
```

실습 및 과제 소스 업로드

- 다음 그림과 같이 Github 저장소에 결과물이 올라간 것을 확인할 수 있음



Git 사용 관련 이슈

- Git에 대한 설명은 다음 문서에서 자세히 확인할 수 있음
 - <http://dogfeet.github.com/articles/2012/progit.html>
- Github의 저장소에서 복사한 뒤, microprocessor 브랜치로 변경 후 사용
- 여러 사람이 같이 사용하기 때문에, 최신 상태가 유지 되지 않은 상태에서 푸싱(push)하려고 하면, 푸싱 되지 않는 문제가 발생할 수 있음
그럴 경우, Github 저장소에서 변경 사항을 가져오거나(fetch) 풀링(pull) 한 뒤, 변경 사항을 푸싱하도록 한다.

Q & A

- 과제, 프로젝트 진행 시 질문 사항은
프로젝트 홈페이지의 포럼을 적극적으로 이용