

(Operating System) Practice -3-

Vi Editor & GCC Basic



Index

I. Vi Practice

II. VS Code Setup

- VI improved (vim) extension
- C++ extension
- GCC

III. GCC Simple Example

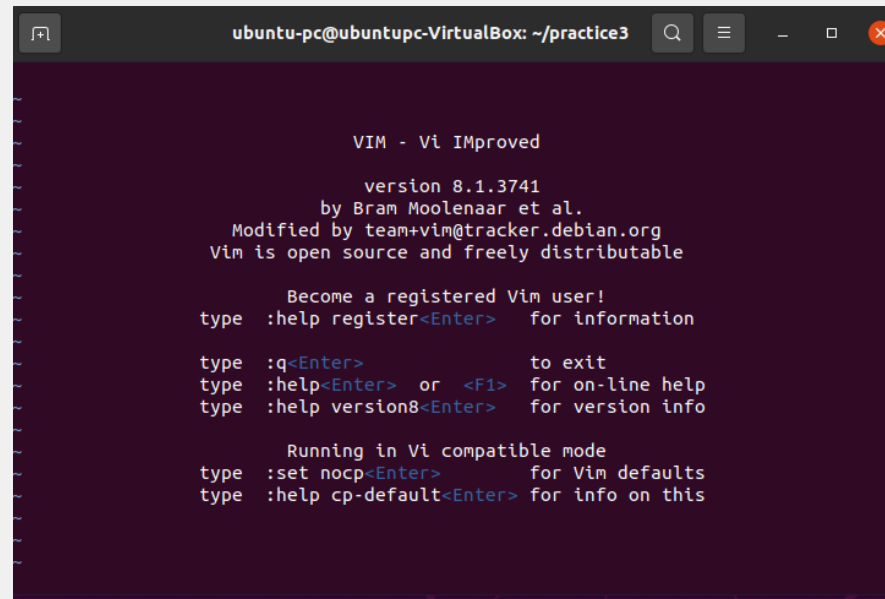


Vi Practice

- **Vi editor**

- 유닉스 환경에서 전통적으로 사용되온 텍스트 편집기
- 리눅스 환경에서 코딩할 때 아직도 많이 사용
- 모든 편집을 키보드로 할 수 있음, 정규식표현의 활용이 용이
→ 익숙해지면 코드 편집 효율이 높아짐

Example



```
ubuntu-pc@ubuntupc-VirtualBox: ~/practice3
VIM - Vi IMproved
      version 8.1.3741
      by Bram Moolenaar et al.
      Modified by team+vim@tracker.debian.org
      Vim is open source and freely distributable

      Become a registered Vim user!
      type  :help register<Enter>  for information

      type  :q<Enter>               to exit
      type  :help<Enter>  or  <F1>  for on-line help
      type  :help version8<Enter>  for version info

      Running in Vi compatible mode
      type  :set nocp<Enter>        for Vim defaults
      type  :help cp-default<Enter> for info on this
```

Vi Practice

- Vi Improved (VIM) 설치

- `sudo apt-get install vim` → Vi 최신버전인 VIM 설치

Example

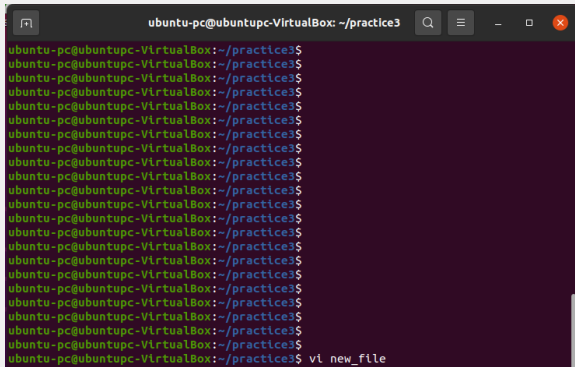
```
ubuntu-pc@ubuntu-pc-VirtualBox: ~/practice3
ubuntu-pc@ubuntu-pc-VirtualBox:~/practice3$ sudo apt-get install vim
[sudo] password for ubuntu-pc:
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  vim-runtime
Suggested packages:
  ctags vim-doc vim-scripts
The following NEW packages will be installed:
  vim vim-runtime
0 upgraded, 2 newly installed, 0 to remove and 89 not upgraded.
Need to get 7,110 kB of archives.
After this operation, 34.6 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://kr.archive.ubuntu.com/ubuntu focal-updates/main amd64 vim-runtime a
ll 2:8.1.2269-1ubuntu5.7 [5,872 kB]
Get:2 http://kr.archive.ubuntu.com/ubuntu focal-updates/main amd64 vim amd64 2:8
.1.2269-1ubuntu5.7 [1,238 kB]
Fetched 7,110 kB in 32s (222 kB/s)
Selecting previously unselected package vim-runtime.
(Reading database ... 184614 files and directories currently installed.)
Preparing to unpack .../vim-runtime_2%3a8.1.2269-1ubuntu5.7_all.deb ...
Adding 'diversion of /usr/share/vim/vim81/doc/help.txt to /usr/share/vim/vim81/d
```

Vi Practice

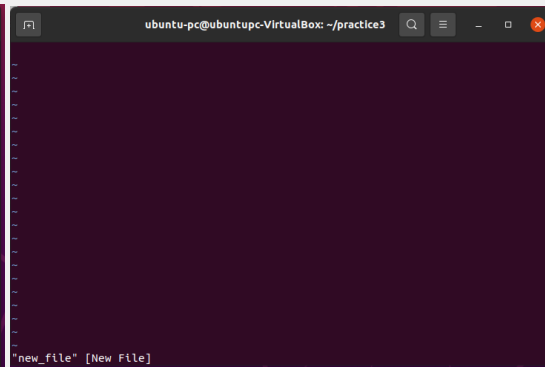
• Vi editor 실행과 종료

- vim [filename] → 입력한 file을 vi 편집기로 실행
- vim [new_filename] → Directory에 없는 file 이름을 입력하면, 새 파일 생성
- :q → 종료

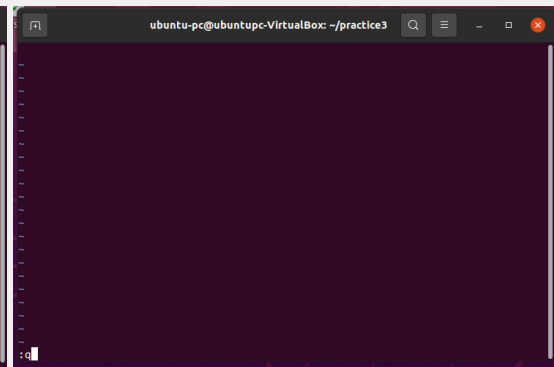
Example



새 파일 생성



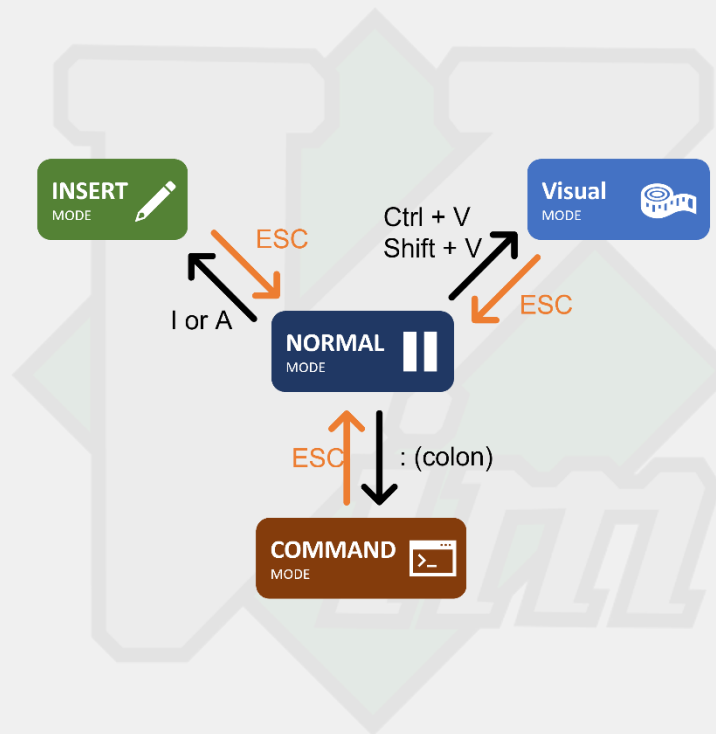
vi 실행상태

vi **종료**

Vi Practice

• Vi editor 모드

- Normal mode: 일반모드, 다른 모드로 진입
- Visual mode: 보기모드, 문서 내 범위 지정
- Insert mode: 편집모드, 문서 편집
- Command mode: 명령어모드, 저장 & 종료 등 편집 이외의 기능 실행



Vi Practice

- Vi editor 모드

- `ESC` → Normal 모드로 전환
- `i / a` → Insert 모드로 전환
- `v / CTRL + v` → Visual 모드로 전환
- `:` → Command 모드로 전환

Example



Vi Practice

- **Vi editor 커서이동 (Normal 모드)**

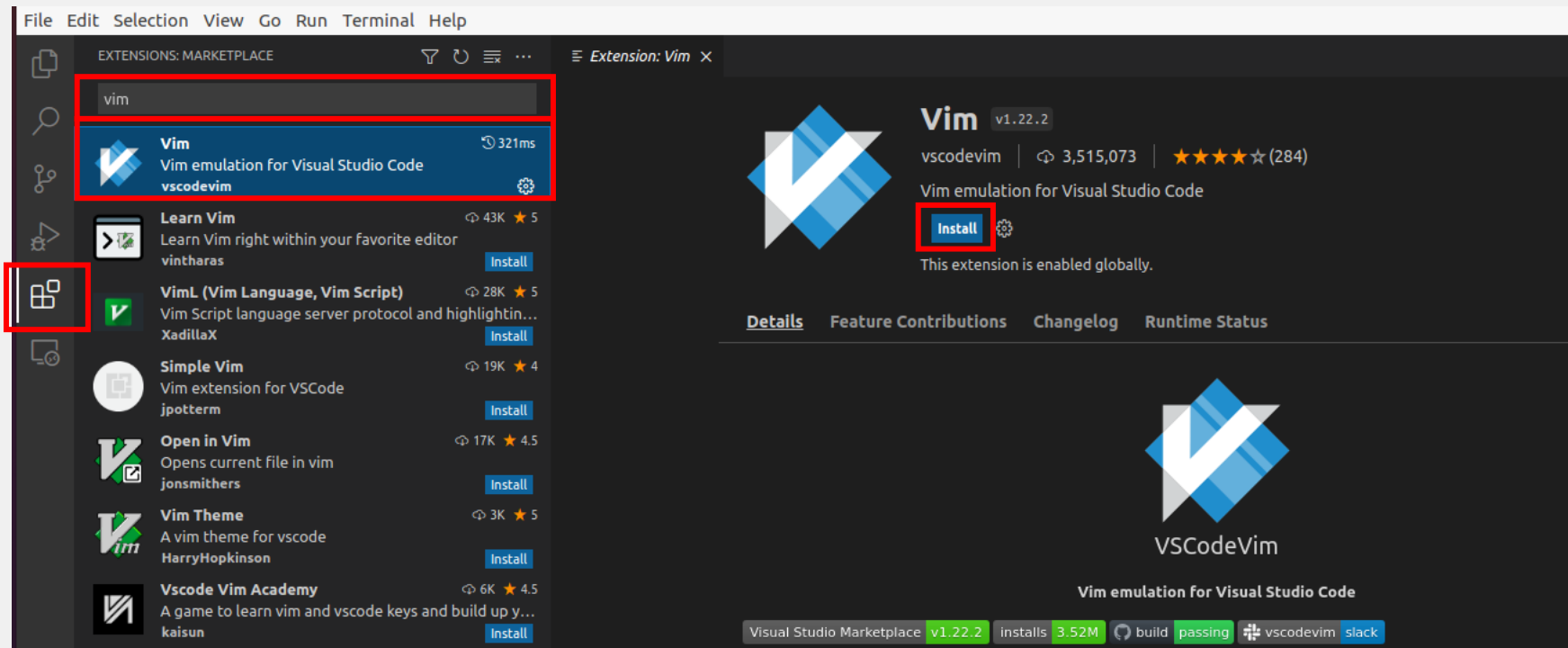
- `h` → 왼쪽으로 이동
- `j` → 아래로 이동
- `k` → 위로 이동
- `l` → 오른쪽으로 이동
- `Shift + $` → 현재 라인 가장 마지막으로 이동
- `Shift + ^` → 현재 라인 가장 처음으로 이동

- **Vi editor 편집 명령어 (Visual 모드)**

- `y` → 복사
- `p` → 붙여넣기
- `d` → 삭제
- `u` → 이전 명령 취소

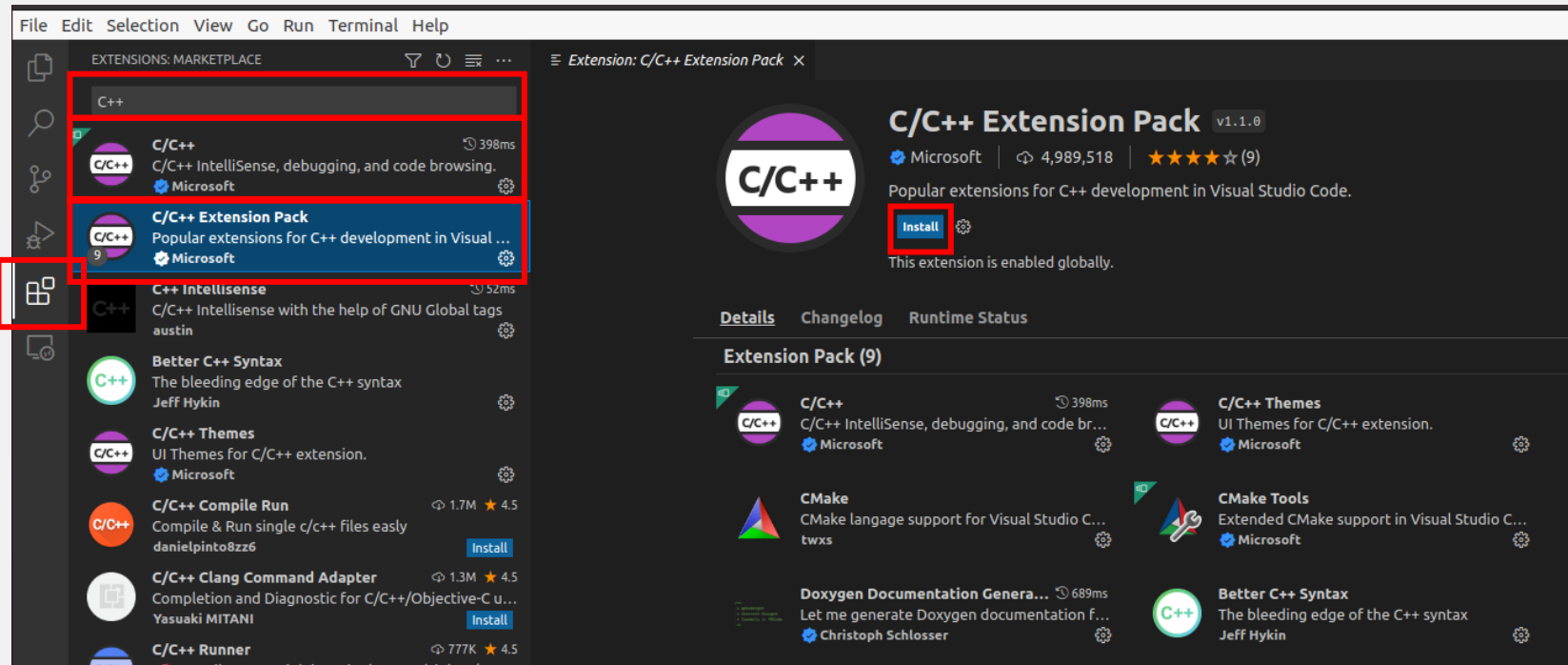
VS Code Setup

- VI improved (vim) extension



VS Code Setup

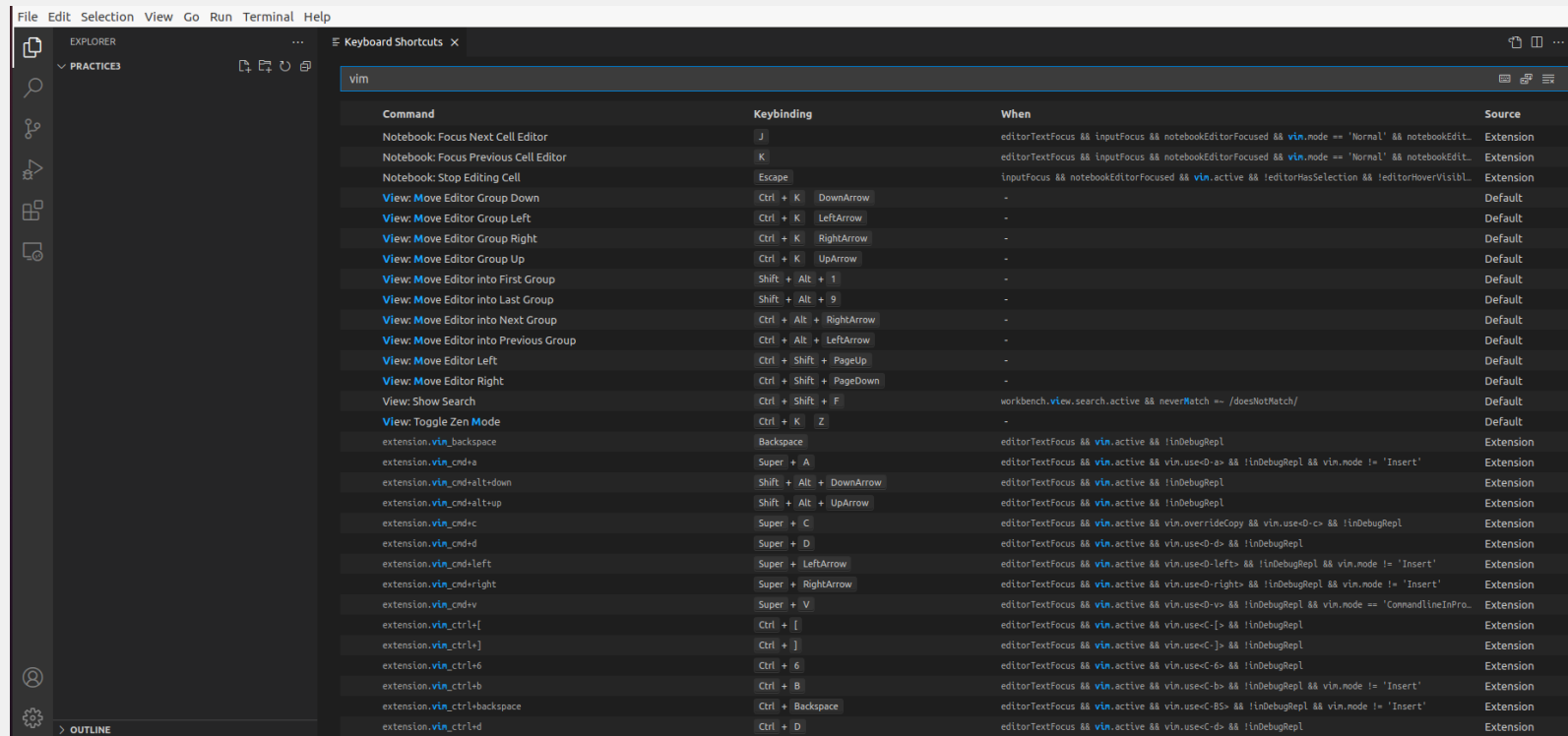
- C++ extension



VS Code Setup

- **VI improved (vim) extension**

- **CTRL + k + s** → VS code 단축키 목록
- “vim” 검색
- “cntr + c”, “cntr + c”, “cntr + c”, “cntr + c” 삭제 → 윈도우와 같은 명령어 사용 가능



VS Code Setup

- **GCC**

- `sudo apt-get update` → apt 업데이트
- `sudo apt-get install build-essential` → 컴파일 도구 설치
- `gcc --version` → GCC 설치 확인

Example

```
ubuntu-pc@ubuntupc-VirtualBox:~$ sudo apt-get update
[sudo] password for ubuntu-pc:
Get:1 http://security.ubuntu.com/ubuntu focal-security InRelease [11.1 kB]
Hit:2 http://kr.archive.ubuntu.com/ubuntu focal InRelease
Get:3 http://kr.archive.ubuntu.com/ubuntu focal-updates InRelease [11.1 kB]
Get:4 http://security.ubuntu.com/ubuntu focal-security/main amd64 DEB [40.6 kB]
Get:5 http://security.ubuntu.com/ubuntu focal-security/universe amd64 DEB [66.4 kB]
Get:6 http://kr.archive.ubuntu.com/ubuntu focal-backports InRelease [11.1 kB]
Get:7 http://security.ubuntu.com/ubuntu focal-security/multiverse amd64 DEB [2,464 B]
```

```
khuzdar@virtualbox:~$ sudo apt install build-essential
R UbuntuSoftware lists... Done
Building dependency tree
Reading state information... Done
build-essential is already the newest version (12.8ubuntu1).
The following packages were automatically installed and are no longer required:
  gconf-service gconf-service-backend gconf2-common libcairo2 libnode64 libqt5designer5 libqt5help5 libqt5sql5 libqt5test5 libqt5xml5 nodejs nodejs-doc python3-pyqt5
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 408 not upgraded.
```

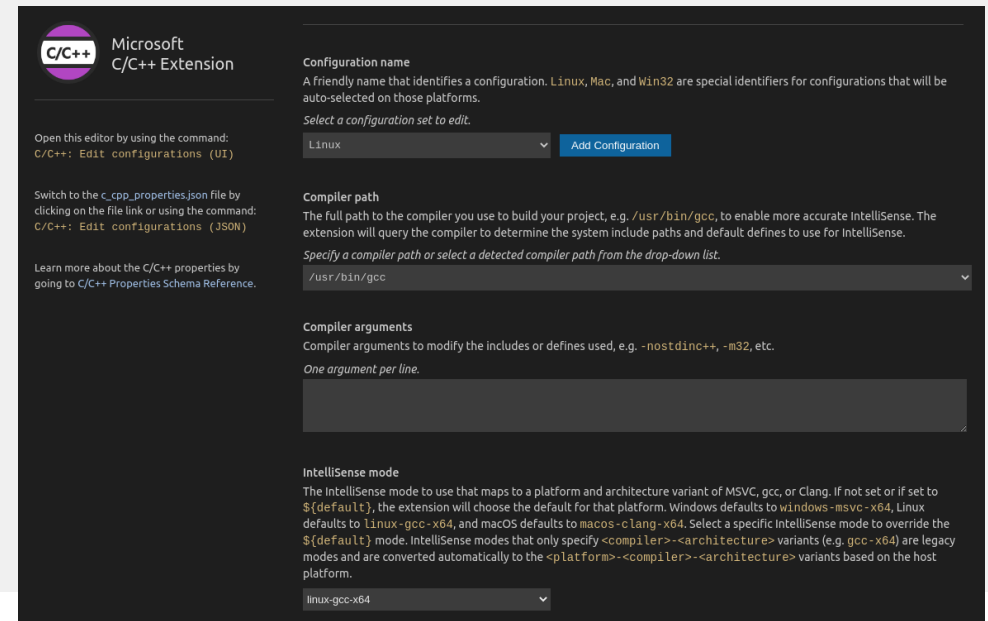
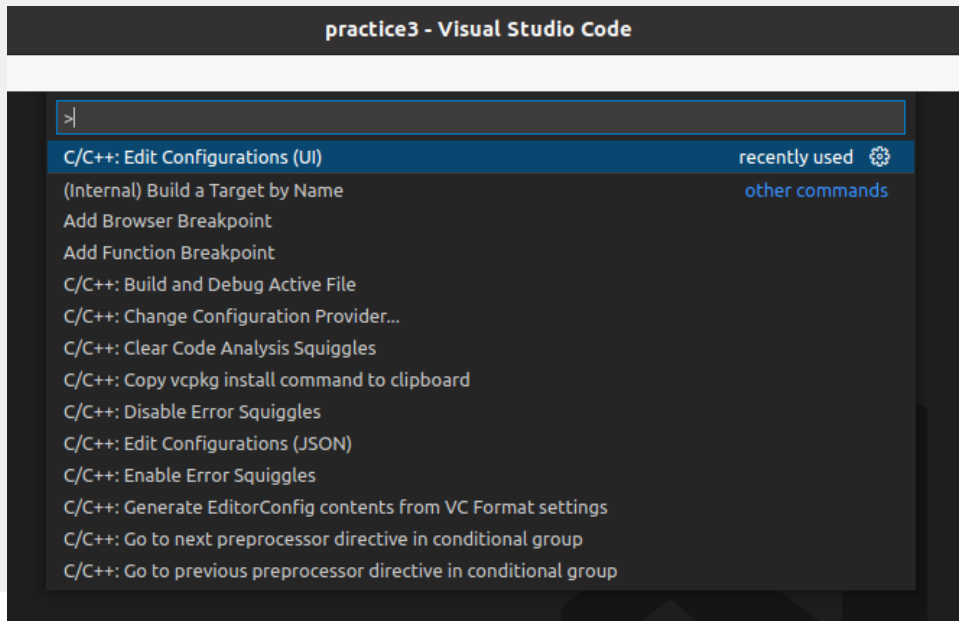
```
ubuntu-pc@ubuntupc-VirtualBox:~$ gcc --version
gcc (Ubuntu 9.4.0-1ubuntu1~20.04.1) 9.4.0
Copyright (C) 2019 Free Software Foundation, Inc.
This is free software; see the source for copying conditions. There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
```

VS Code Setup

- **GCC**

- Folder에 C 파일 생성 이후 → **CTRL + Shift + p** → C/C++ Edit Configuration(UI)
- 설정 확인
 - ✓ Configure path → /usr/bin/gcc
 - ✓ IntelliSense mode → linux-gcc-x64

Example



GCC Simple Example

• GCC 컴파일러 구성

- cpp: 전처리기 → #include, #define 등의 헤더부분 전처리
- cc1: 컴파일러 → 어셈블리어로 된 파일 생성
- as: 어셈플러 → 어셈블리어를 기계어로 바꾼 오브젝트 파일 생성
- ld: 링커 → 라이브러리 함수와 오브젝트 파일을 연결하고, 실행파일 생성



GCC Simple Example

- GCC 옵션

Option	Description
-E	전처리만 수행, i확장자 파일 출력
-S	컴파일까지 수행, s확장자 파일 출력
-c	어셈블까지 수행, o확장자 파일 출력
-o	링커까지 수행, 바이너리 형식의 실행파일 출력



GCC Simple Example

- **Example code**

- 예제코드 작성 및 실행
- `gcc -E hello.c -o hello.i` → 전처리까지 수행
- `gcc -S hello.c` → 컴파일러까지 수행
- `gcc -c hello.c` → 어셈블러까지 수행
- `gcc -E hello.c -o hello` → 실행파일 생성 → `./hello` → 실행

Example

```
C hello.c x
C hello.c > main(void)
1  #include <stdio.h>
2
3  int main(void){
4      printf("Hello World!");
5      return 0;
6  }

ubuntu-pc@ubuntupc-VirtualBox:~/practice3$ ls
hello.c
ubuntu-pc@ubuntupc-VirtualBox:~/practice3$ gcc -E hello.c -o hello.i
ubuntu-pc@ubuntupc-VirtualBox:~/practice3$ ls
hello.c hello.i
ubuntu-pc@ubuntupc-VirtualBox:~/practice3$ gcc -S hello.c
ubuntu-pc@ubuntupc-VirtualBox:~/practice3$ ls
hello.c hello.i hello.s
ubuntu-pc@ubuntupc-VirtualBox:~/practice3$ gcc -c hello.c
ubuntu-pc@ubuntupc-VirtualBox:~/practice3$ ls
hello.c hello.i hello.o hello.s
ubuntu-pc@ubuntupc-VirtualBox:~/practice3$ gcc hello.c -o hello
ubuntu-pc@ubuntupc-VirtualBox:~/practice3$ ls
hello hello.c hello.i hello.o hello.s
ubuntu-pc@ubuntupc-VirtualBox:~/practice3$ ./hello
Hello Operating systems
ubuntu-pc@ubuntupc-VirtualBox:~/practice3$
```