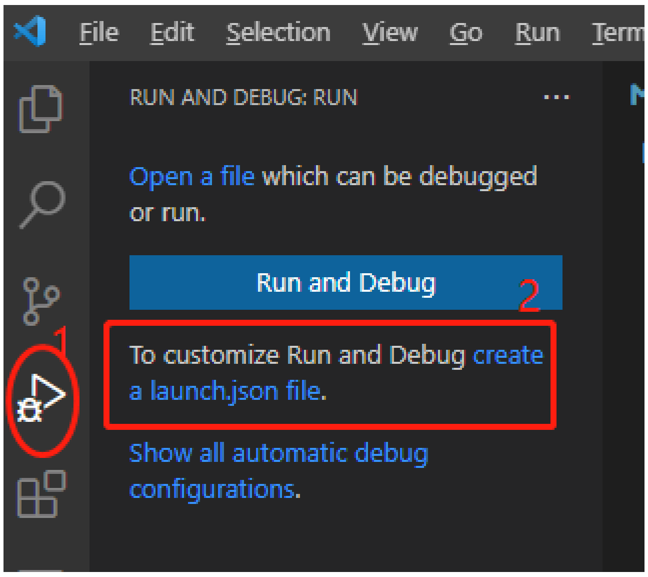
Ubuntu 下用 VSCode 调试 C++

1　准备工作

- 在 Ubuntu 系统中安装以下软件：gcc/g++、VSCode、cmake、gdb
- 在 VSCode 中安装以下插件：C/C++、CMake、CMake Tools

2　配置 launch.json 文件

在 VSCode 中点击左侧运行与调试按钮，添加 launch.json 文件


RUN AND DEBUG: RUN

Open a file which can be debugged or run.

To customize Run and Debug create a launch.json file.

Show all automatic debug configurations.

在 launch.json 文件中复制替换为以下内容：

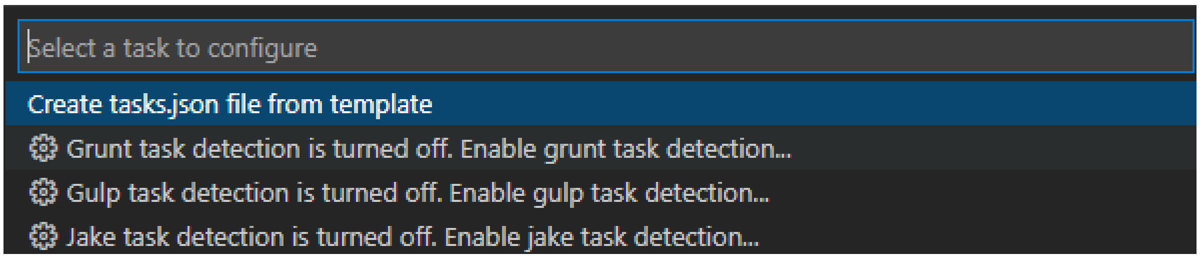
```
{
    // Use IntelliSense to learn about possible attributes.
    // Hover to view descriptions of existing attributes.
    // For more information, visit: https://go.microsoft.com/fwlink/?linkid=830387
    "version": "0.2.0",
    "configurations": [
        {
            "name": "g++ - 生成和调试活动文件",
            "type": "cppdbg",
            "request": "launch",
                "program": "${workspaceFolder}/build/my_cmake_exe",
                "args": [],
                "stopAtEntry": false,
                "cwd": "${workspaceFolder}",
                "environment": [],
                "externalConsole": false,
                "MIMode": "gdb",
                "setupCommands": [
                    {
                        "description": "为 gdb 启用整齐打印",
                        "text": "-enable-pretty-printing",
                        "ignoreFailures": true
                    }
                ],
                "preLaunchTask": "Build",
                "miDebuggerPath": "/usr/bin/gdb"
            }
        ]
}
```

其中红色标记的内容需要额外的关注：

- "program"通过绝对路径指定需要进行调试的文件，必要时进行修改
- "cwd"指定工作路径
- "preLaunchTask"指定调试前的预编译过程

3 配置 tasks.json 文件

在顶部点击 Terminal 选择 Configure Default Build Task …，随后选择 create tasks.json file from template


Select a task to configure

Create tasks.json file from template

Grunt task detection is turned off. Enable grunt task detection...

Gulp task detection is turned off. Enable gulp task detection...

Jake task detection is turned off. Enable jake task detection...

在默认创建的 tasks.json 文件中复制替换为以下内容：

```
{
  "version": "2.0.0",
  "options": {
    "cwd": "${workspaceFolder}/build"
  },
  "tasks": [
    {
      "type": "shell",
      "label": "cmake",
      "command": "cmake",
      "args": [
        ".."
      ]
    },
    {
      "label": "make",
      "group": {
        "kind": "build",
        "isDefault": true
      },
      "command": "make",
      "args": [


      ]
```

```
            },
            {
                "label": "Build",
                        "dependsOrder": "sequence", // 按列出的顺序执行任务依赖项
                "dependsOn":[
                    "cmake",
                    "make"
                ]
            }
        ]

}
```

该文件的本质就是将 cmake ..和 make 的预编译操作通过 tasks.json 文件交给 VSCode 进行处理。接下来可以按 F5 进行调试了，调试前别忘记打断点。

4 配置 CMakeLists.txt 文件

当然，也需要配置 CMakeLists.txt 文件，相关模板如下：

```
cmake_minimum_required(VERSION 3.0)

project(SOLIDERFIRE)

set(CMAKE_CXX_FLAGS "${CMAKE_CXX_FLAGS} -Wall")

set(CMAKE_BUILD_TYPE Debug)

include_directories(${CMAKE_SOURCE_DIR}/include)

add_executable(my_cmake_exe main.cpp src/Gun.cpp src/Solider.cpp)
```

在发行版中需要将 Debug 改为 Release