使用 VScode 调试反汇编代码

by Anxi

本文档介绍如何使用 VScode 调试反汇编代码。

首先是对于一般的 Windows 环境，我们可以直接在工作目录下通过配置.vscode 文件夹下的 launch.json 和 task.json 文件来，通过调试控制台和反汇编视图来完成对于反汇编代码的单步调试。

Step1．配置 launch.json 和 task.json 文件。

launch.json 如下：

```
{
    "version": "0.2.0",
    "configurations": [
        {
            "name": "Debug",
            "type": "cppdbg",
            "request": "launch",
            "program": "${fileDirname}\\exe_collect\\${fileBasenameNoExtension}.exe",
            "args": [],
            "stopAtEntry": false,
            "cwd": "${fileDirname}",
            "environment": [],
            "externalConsole": false,
            "MIMode": "gdb",
            "miDebuggerPath": "D:\\mingw64\\bin\\gdb.exe",
            "setupCommands": [
                {
                    "description": "为 gdb 启用整齐打印",
                    "text": "-enable-pretty-printing",
                    "ignoreFailures": true
                },
                {
                    "description": "将反汇编风格设置为 att",
                    "text": "-gdb-set disassembly-flavor att",
                    "ignoreFailures": true
                }
            ],
            "preLaunchTask": "C/C++: g++.exe 生成活动文件"
        },
    ]
}
```

task.json 如下：

```
{
    "tasks": [
        {
```

```
            "type": "cppbuild",
            "label": "C/C++: g++.exe 生成活动文件",
            "command": "D:\\mingw64\\bin\\g++.exe",
            "args": [
                "-fdiagnostics-color=always",
                "-g",
                "${file}",
                "-o",
                "${fileDirname}\\exe_collect\\${fileBasenameNoExtension}.exe",
                "-fexec-charset=GBK"
            ],
            "options": {
                "cwd": "${fileDirname}"
            },
            "problemMatcher": [
                "$gcc"
            ],
            "group": {
                "kind": "build",
                "isDefault": true
            },
            "detail": "调试器生成的任务。"
        }
    ],
    "version": "2.0.0"
}
```

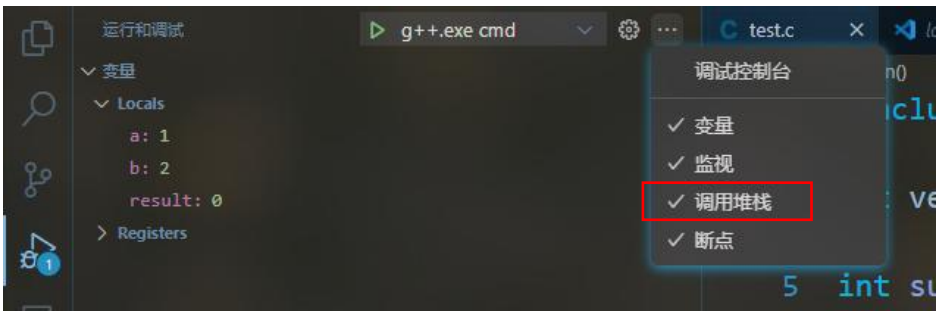
注意将 gdb.exe 和 g++.exe 的路径改为自己的安装路径（最好使用新版 MinGW）。

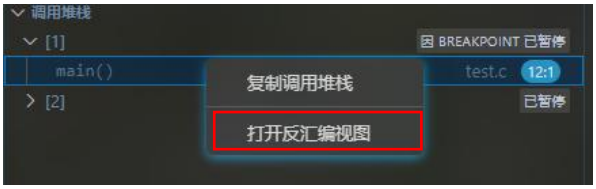
Step2. 按下“F5”开始调试。

Step3. 查看反汇编代码。

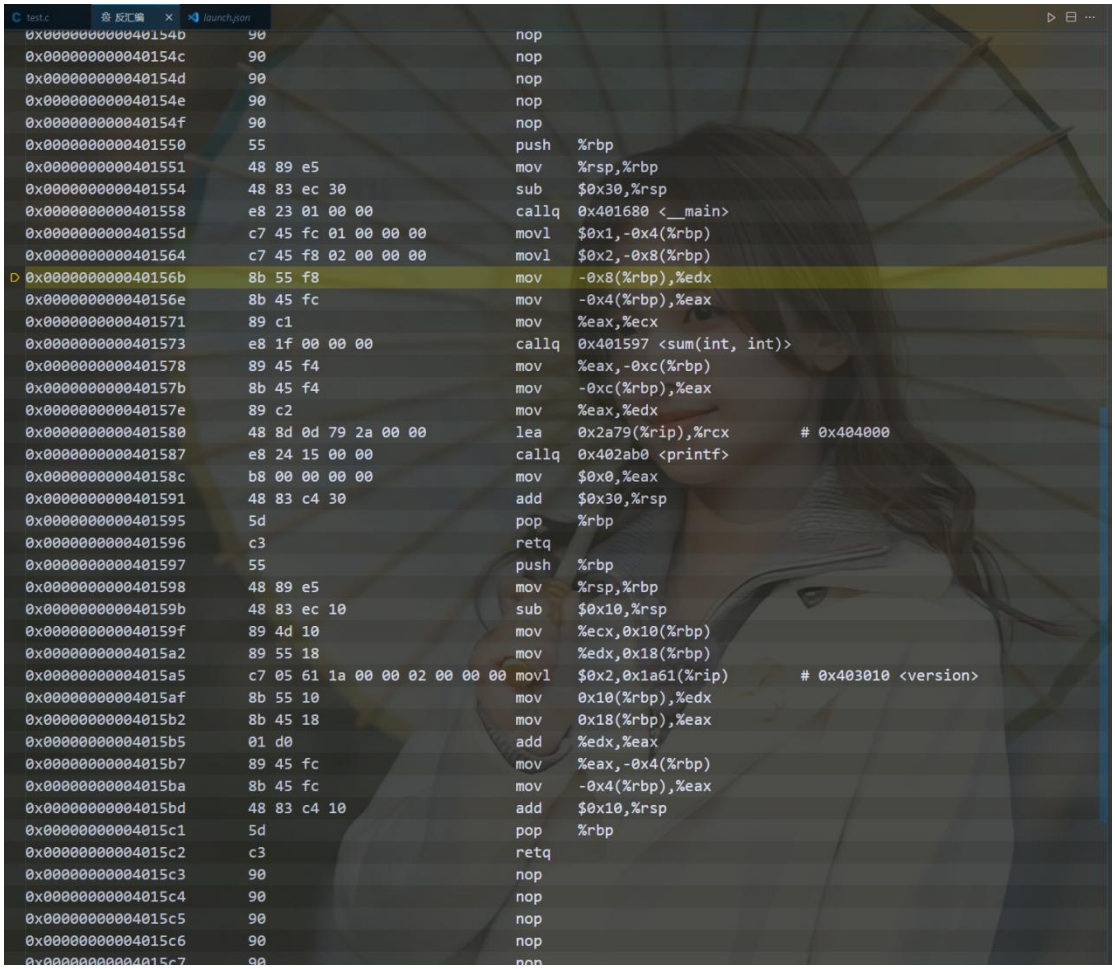
下面介绍三种查看反汇编代码的方式：

法 1. 确保打开了“调用堆栈”视图

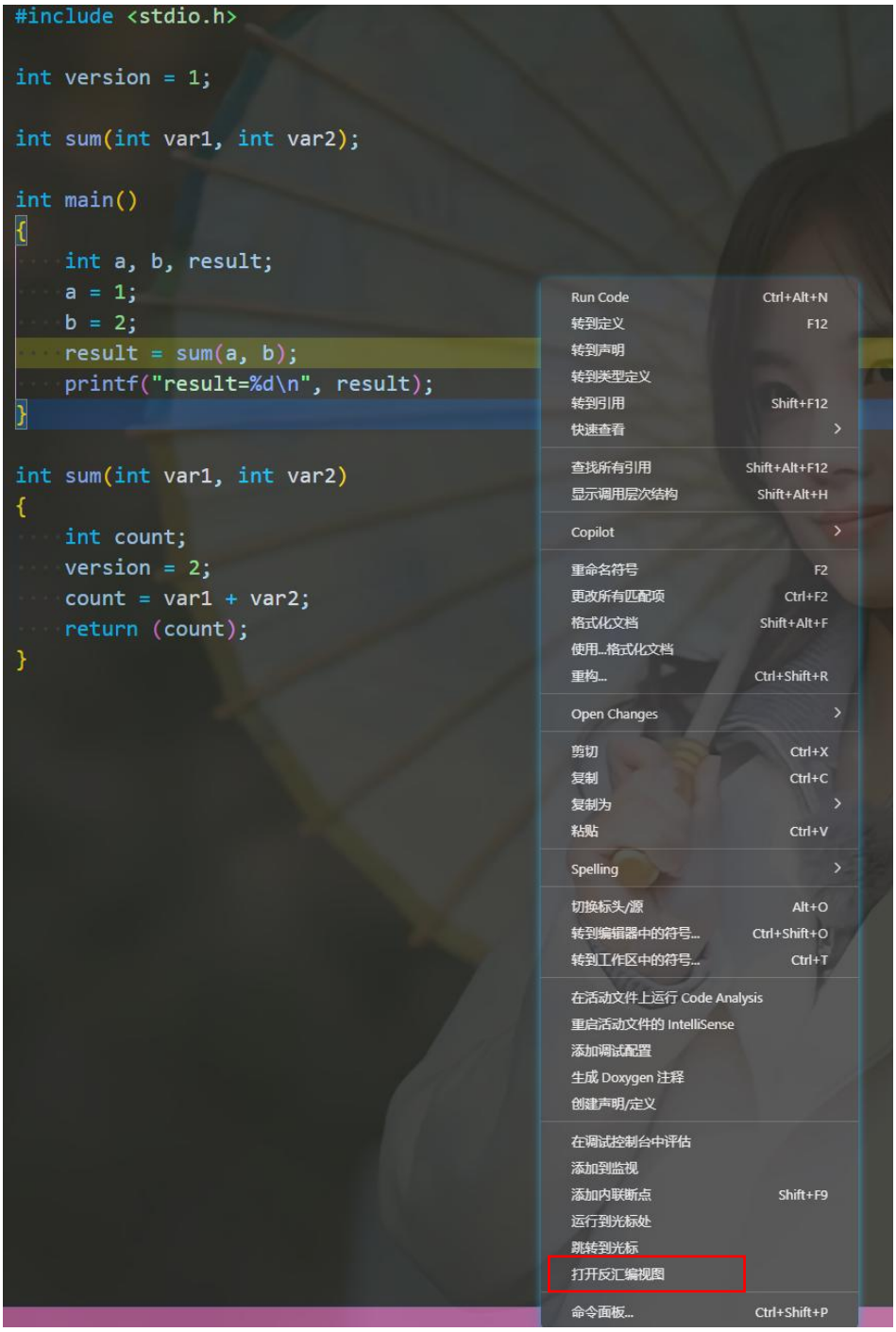




在“调用堆栈”处右键 main()，即调试断点所在的函数，点击“打开反汇编视图”即可查看反汇编代码。结果如下：



法 2．直接在编辑器中按右键，在弹出的菜单栏中点击“打开反汇编视图”也可正常打开

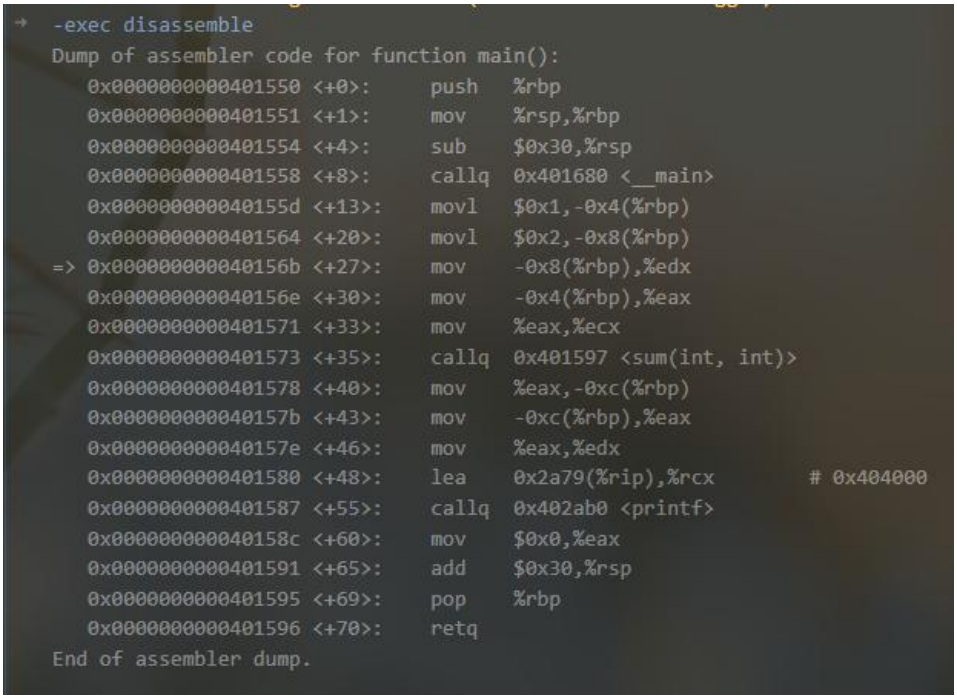


法 3. 在调试控制台中执行 GDB 命令”disassemble”

一般在 VScode 中运行或调试程序时都会自动跳出一些选项卡，默认是跳转到终端界面，但其实在终端左边还有一个调试控制台，在该控制台中可以执行一系列的 GDB 命令。

要是没有自动打开，可以通过“查看->打开视图->调试控制台”唤出。

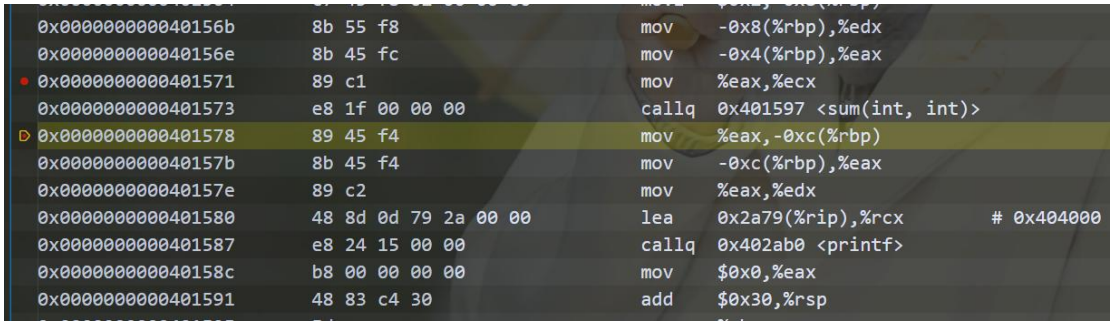
在给出的输入框中输入“-exec disassemble”命令即可输出当前断点所在函数的反汇编代码。



但是该种方法并不能对反汇编代码进行单步调试，不过在 Unix V6++系统中，不知道由于什么原因，反汇编视图无法显示正常的反汇编代码，可以通过该方法对照查看当前正在执行的反汇编代码。

Step4．在反汇编视图中通过单步执行方式调试代码，可以实现对于反汇编代码的单步执行任务。

使用该方法也可以在反汇编视图中对反汇编代码进行断点调试，例如：



在 Unix V6++环境中，通过查看给出的 launch.json 文件不难发现，其中表示要调试的程序的 program 属性对应的路径为 kernel.exe 导致只能调试内核程序。在 VScode 中，不像 eclipse 那么复杂，可以直接修改该路径从而实现对于自己编写的程序的调试。

例如，我想要对 ls 指令进行调试，只需要将 launch.json 文件修改为如下代码：

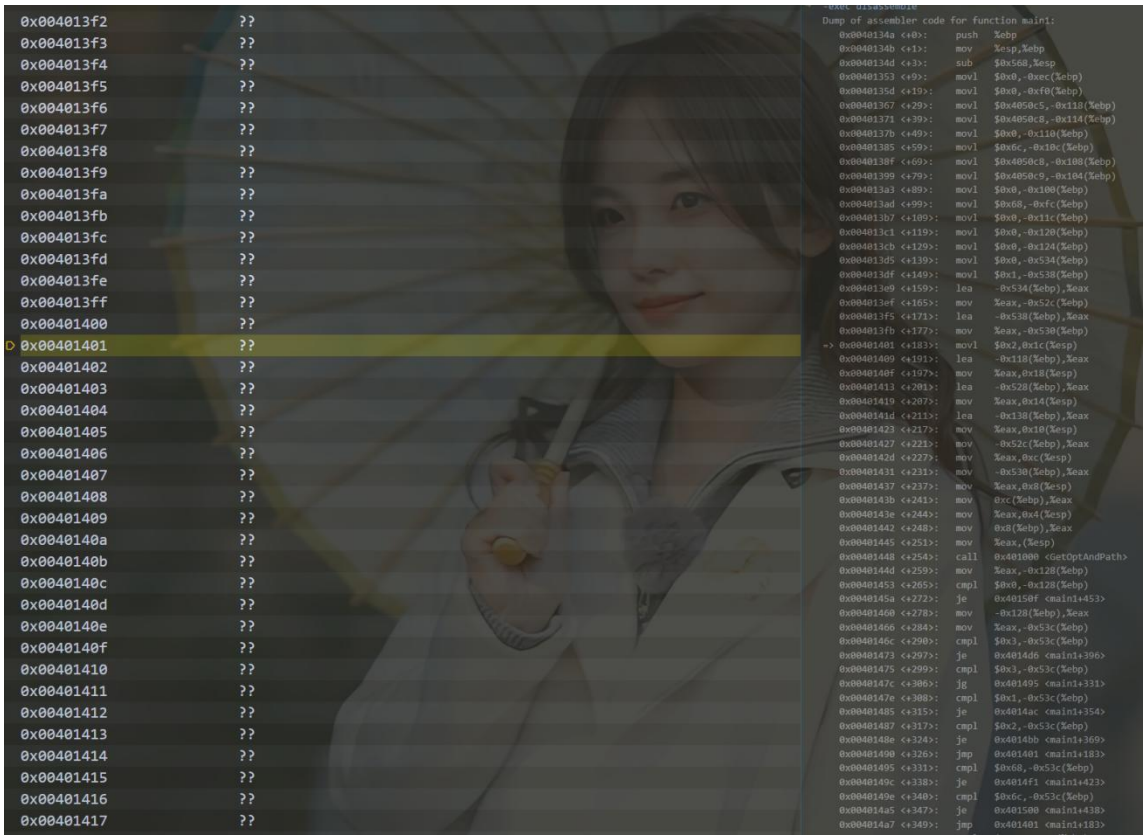
```
{
    "name": "Debug test",
    "type": "cppdbg",
    "request": "launch",
    "cwd": "${workspaceFolder}/src",
    "miDebuggerServerAddress": "localhost:1234",
    "miDebuggerPath": "E:/Unix_V6++/MinGW/bin/gdb.exe",
    "setupCommands": [
        {
            "description": "为 gdb 启用整齐打印",
            "text": "-enable-pretty-printing",
            "ignoreFailures": true
        },
        {
            "description": "将反汇编风格设置为 att",
            "text": "-gdb-set disassembly-flavor att",
            "ignoreFailures": true
        }
    ],
    "program": "${workspaceFolder}/src/program/objs/ls.exe",
    "args": [],
    "externalConsole": false,
    "MIMode": "gdb",
    "preLaunchTask": "OOS Run" // 前置任务：启动 Bochs，运行 OOS
}
```

注意，这里相较于原始的 launch.json 还添加了一些对于 gdb 的控制命令。其次，还需注意将 GDB 路径修改为自己的安装路径。

修改成上述代码之后就可以对 ls 程序进行正常的调试工作。但是这里会出现的一个问题就是，对于一些命令，有可能使用 cd 命令也会导致断点被触发，这里暂时并未想明白原因。不过我们可以不用 cd 进入 bin 文件夹，直接运行相应命令也可以 debug。

使用上面提供的方法进入反汇编视图会发现，视图中无法正常显示反汇编代码，这时我们可以在调试控制台中执行”-exec disassemble”命令来查看正确的反汇编代码，然后在反汇编视图中单步执行反汇编代码对照调试控制台的结果进行调试。这里暂未找出解决方法。

以 ls 为例，通过上述方法进行调试的示例如下图：



注意，在调试控制台输出的反汇编代码中，”=>”指向的是正在执行的汇编指令。

提示：有时候 VScode 的反汇编视图无法正常显示，这种情况只需要重启 VScode 就能解决！！！