

使用 VSCode 进行 OOS GDB 调试 V2.0

By 刘吉加 陈元哲

1 简介

担心早古版本的 MinGW 污染了环境变量？

调试 OOS 却需要命令行输入繁琐的指令？

想要体验 Ctrl+ Shift +B 一键编译 / F5 一键调试吗？

下一个 Eclipse，何必是 Eclipse。

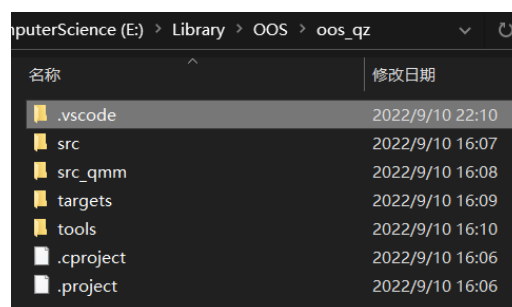
2 配置教程

注：无需配置环境变量，无需按原本教程配置，直接按照下列步骤进行即可

(1) 编辑 oos/targets/UNIXV6++/bochsrc.bxrc，下方语句用于开启 GDB 服务器，取消注释将开启 GDB，注释将关闭 GDB

```
#####  
gdbstub: enabled=1, port=1234, text_base=0, data_base=0, bss_base=0
```

(2) 在 oos 根目录下新建 .vscode 文件夹，将 launch.json 和 tasks.json 复制进去



(3) 修改 tasks.json 中的路径，可直接搜索 !修改!

a) 1 处 MingW

b) 1 处 NASM

c) 2 处 Bochs

```
19 // !修改! (1) 请修改为 : MingW 所在位置  
20 // !修改! (2) 请修改为 : NASM 所在位置  
21 "Path": "E:\\Library\\OOS\\MingW\\bin;E:\\Library\\OOS\\NASM;${env:Path};",  
22 // 注意: 1. 别把 ${env:Path}; 删了  
23 // 2. 别把分号 ; 删了
```

```

46 // !修改!(3) 请修改为 : Bochs所在位置
47 "BXSHARE": "E:\\Library\\OOS\\Bochs-2.6",
48 // !修改!(4) 请修改为 : Bochs所在位置
49 "Path": "E:\\Library\\OOS\\Bochs-2.6;${env:Path};"
50 // 注意: 1. 别把 ${env:Path}; 删了
51 //      2. 别把分号 ; 删了

```

(4) 修改 **launch.json** 中的路径，可直接搜索 **!修改!**

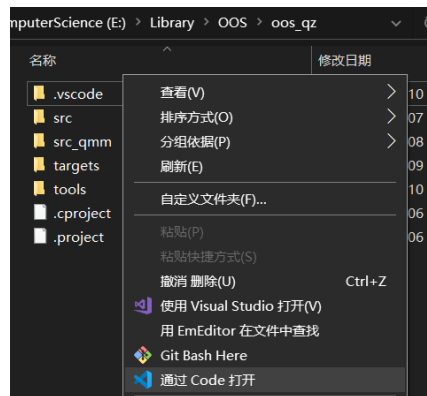
a) 1 处 MinGW GDB

```

23 // !修改!(5) 请修改为 : MinGW GDB 所在位置
24 "miDebuggerPath": "E:\\Library\\OOS\\MinGW\\bin\\gdb.exe",

```

(5) 在 **oos** 根目录下打开 VSCode

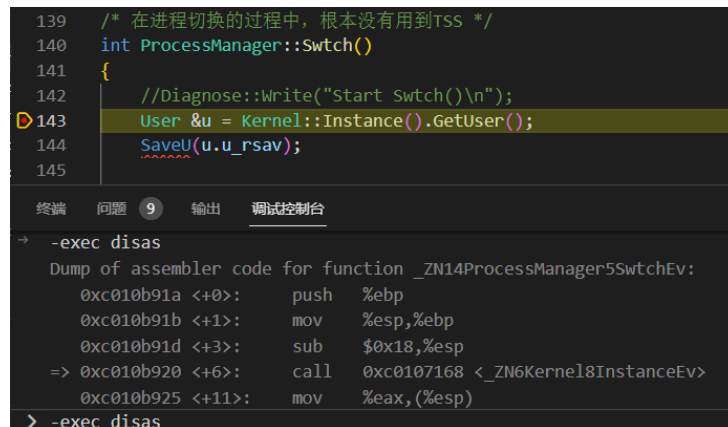


(6) 操作说明

a) 编译: Ctrl+Shift+B

b) 调试: F5

c) 反汇编: 停在断点后，在 VSCode 下方调试控制台输入 **-exec disas**



(7) 如果按照上述步骤操作，**oos** 文件夹位置可以任意移动；MinGW、NASM 和 Bochs 的位置如果改变，则需要按照上述步骤再次修改配置文件

3 常见问题

3.1 无法追踪 OOS Run 任务

问题：首次调试时，会遇到“无法追踪 OOS Run 任务”



解决方法：

- (1) 勾选“记住我对此任务的选择”
- (2) 点击“仍要调试”

原因：为了实现一键调试，目前方案是让 VSCode 先运行 OOS Run 任务启动 Bochs，随后连接 GDB；如果跟踪 Bochs 任务，将会阻塞 GDB 的运行，因此只能禁止对 OOS Run 任务的跟踪。

3.2 编译时卡住

问题：make 编译到 build.exe 这一句时卡住

```
copy ..\targets\img\c.img ..\tools\MakeImage\bin\Debug\c.img
1 file(s) copied.
cd ..\tools\MakeImage\bin\Debug && build.exe c.img boot.bin kernel.bin programs
```

解决方法：

- (1) 关闭 Windows 实时防护



原因：未知

3.3 系统找不到指定的文件

问题：编译时提示 process_begin ... 系统找不到指定的文件

```
make: Entering directory `E:/Library/OOS/oos_qz/src'
del    ..\targets\objs\*.o
process_begin: CreateProcess(NULL, del ..\targets\objs*.o, ...) failed.
make (e=2): 系统找不到指定的文件。
make: *** [clean] Error 2
make: Leaving directory `E:/Library/OOS/oos_qz/src'
```

解决方法：采用下列方法之一

(1) 按照上述步骤操作后，VSCode 不会遇到此问题

(2) 卸载 Git

(3) 在每一个 Makefile 最前面加入一行：SHELL=cmd

原因：如果电脑中存在 Git，make.exe 会首先尝试使用 Git Bash 执行 make 命令，而不是使用 cmd.exe，这会造成 del 语句执行出错。VSCode 解决方案中 task.json 中在执行 make 时加入了参数 "SHELL=cmd"，因此不会遇到此问题，如果使用 Eclipse 或命令行则可能会遇到。

3.4 “还没有为文档加载任何符号”类似问题

问题：调试过程中，提示“还没有为文档加载任何符号”相关错误

解决方法：

(1) 在 launch.json 中，取消下面这段代码的注释

```
16      // "setupCommands": [
17      //      {
18      //          "text": "directory 'E:/Project/unix v6++/oos_qz3/src/boot' 'E:/Project/unix v6++/oos_qz3",
19      //      },
20      // ]
```

(2) 按照其中的说明修改路径

(3) 若 OOS 的文件夹路径发生了变化，上述路径需要对应修改

原因：早古版本的 GDB 由于不明原因会找不到符号文件，加载符号文件后找不到对应的源文件。目前的解决方法并不优雅，考虑使用相对路径，或自动填充路径。

4 TODO

4.1 OOS Makefile

OOS 的 Makefile 存在一些问题，在部分源文件修改后，`make all` 可能并不会将其重新编译，需要先执行 `make clean`（上述配置中，编译前会自动 `make clean`），但是这会增加编译时间，考虑修改 OOS 的 Makefile。

4.2 VSCode 反汇编

相比于 Eclipse，VSCode 对远古版本 GDB 的反汇编支持并不好，反汇编查看较为繁琐，考虑修改 GDB 版本/调整 Bochs 等方法允许 VSCode 使用反汇编视图。