

Ucore 实验的运行和调试

1. shell 下的编译运行调试

先进入一个 lab（例如 `ucore_os_lab/labcodes_answer/lab1_result`），然后键入 `make`，就可以编译了，`make clean` 清除编译的结果，`make qemu` 可以运行代码。

调试呢稍微复杂一点，我们需要开两个 Terminal 窗口，都进入同一个 lab 目录下，其中一个使用 `qemu-system-i386 -S -s -hda ./bin/ucore.img -monitor stdio` 命名将虚拟机跑起来并等待调试操作。项目中有两个可执行文件（`obj/bootblock.o` `bin/kernel`），一个对应于 `bootLoader`，一个对应于操作系统内核，可以选择任意一个进行调试。在另一个窗口键入 `i386-elf-gdb + 需要调试的文件`，进入 `gdb` 后，键入 `target remote 127.0.0.1:1234`，使得 `gdb` 与 `qemu` 连接。然后就可以开始调试了。

`b + 函数名 或者 内存地址`：表示添加断点

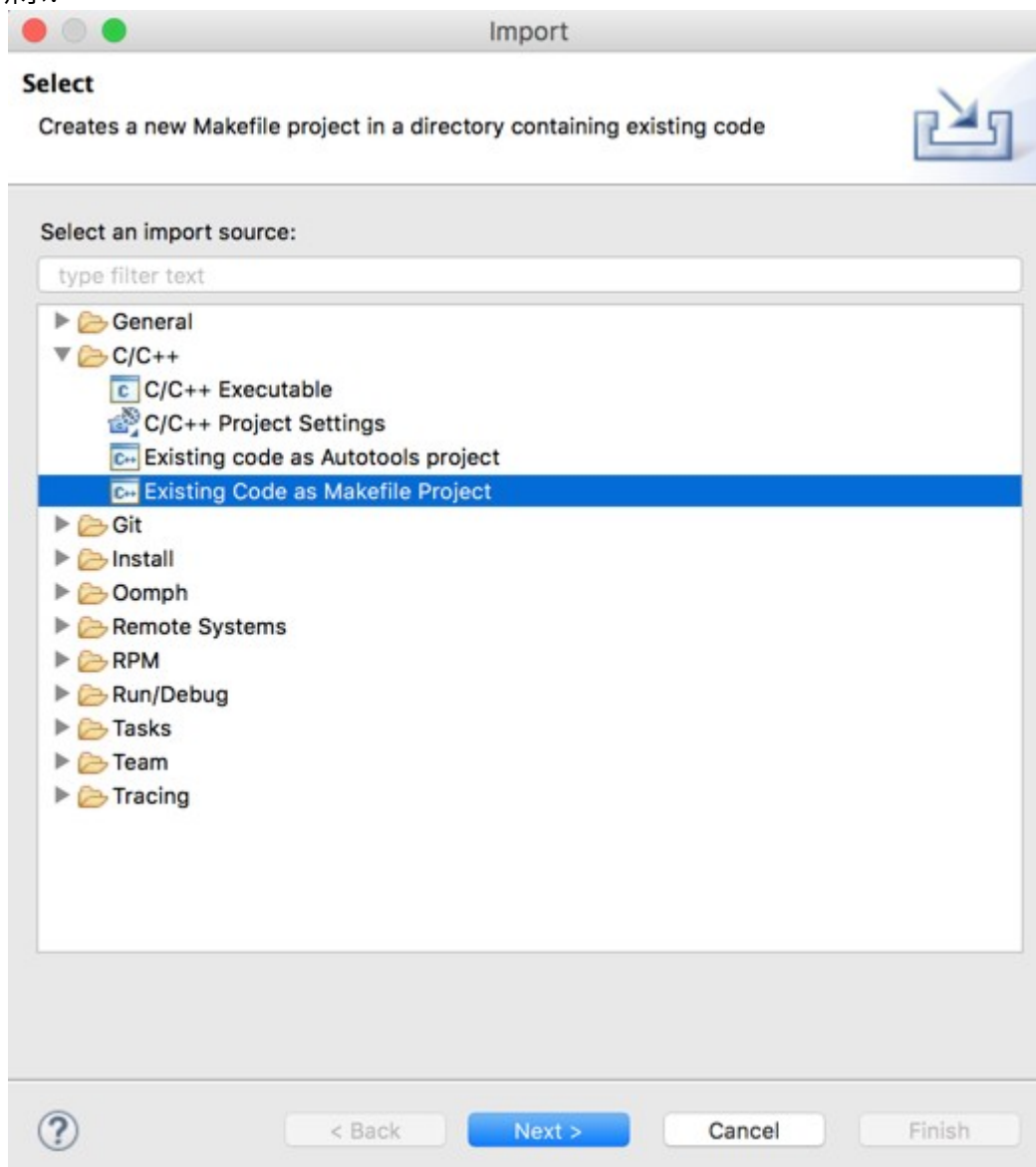
`next`：执行下一条语句

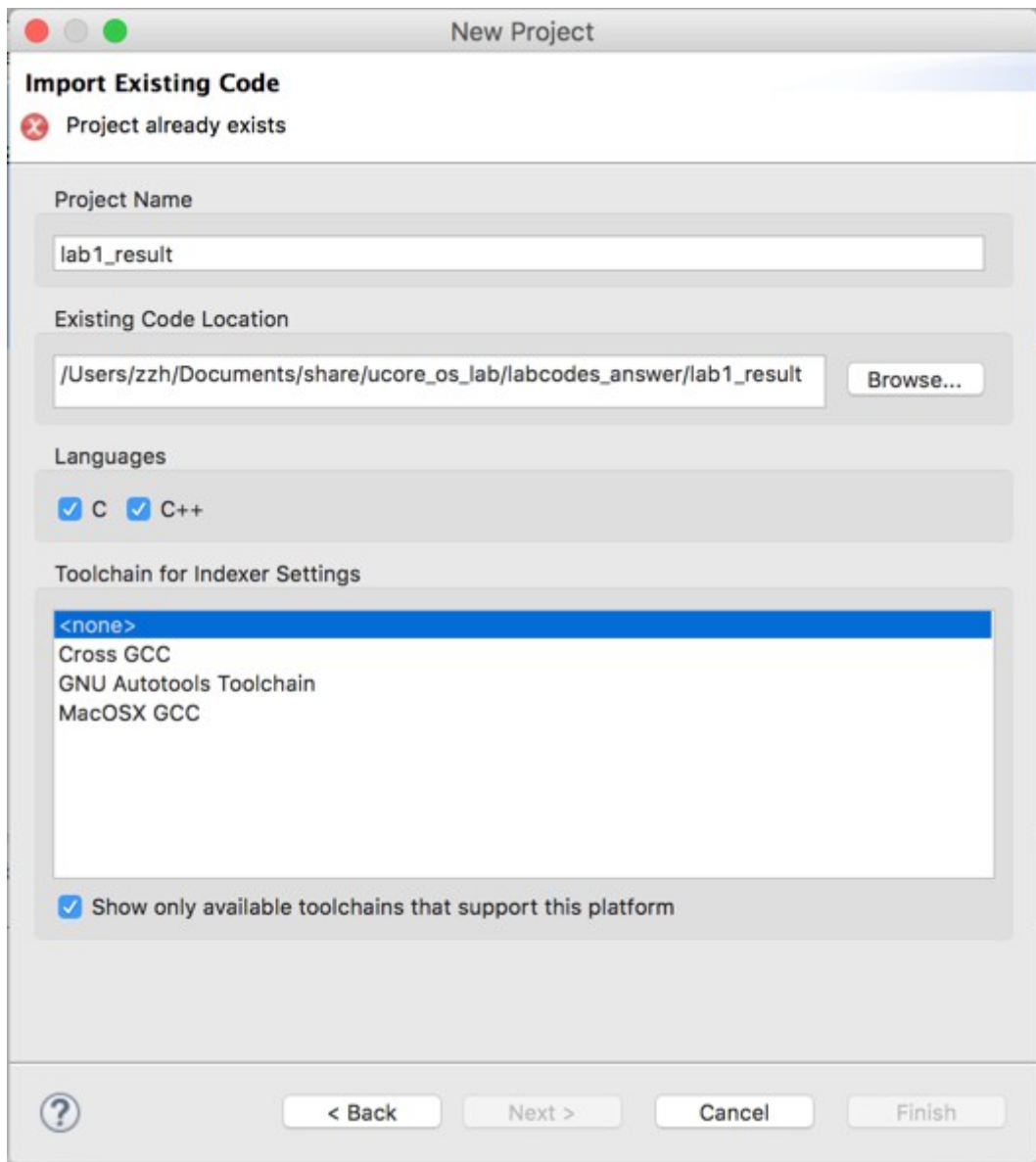
`nexti`：执行下一条汇编语句

`file + 文件名`：指定要调试的文件

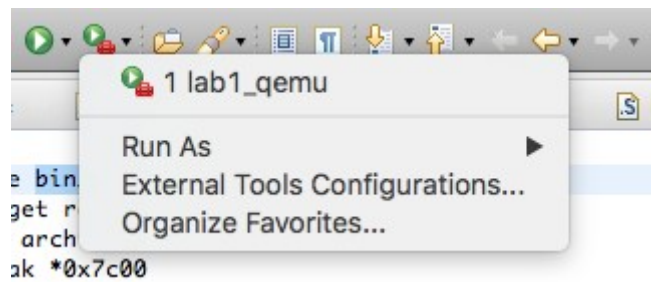
2. Eclipse 下调试代码

下面说说如何在 Eclipse 下调试代码，选择一个 lab（例如 `ucore_os_lab/labcodes_answer/lab1_result`）导入 Eclipse，然后在 Terminal 中 `make`。过程如下所示：

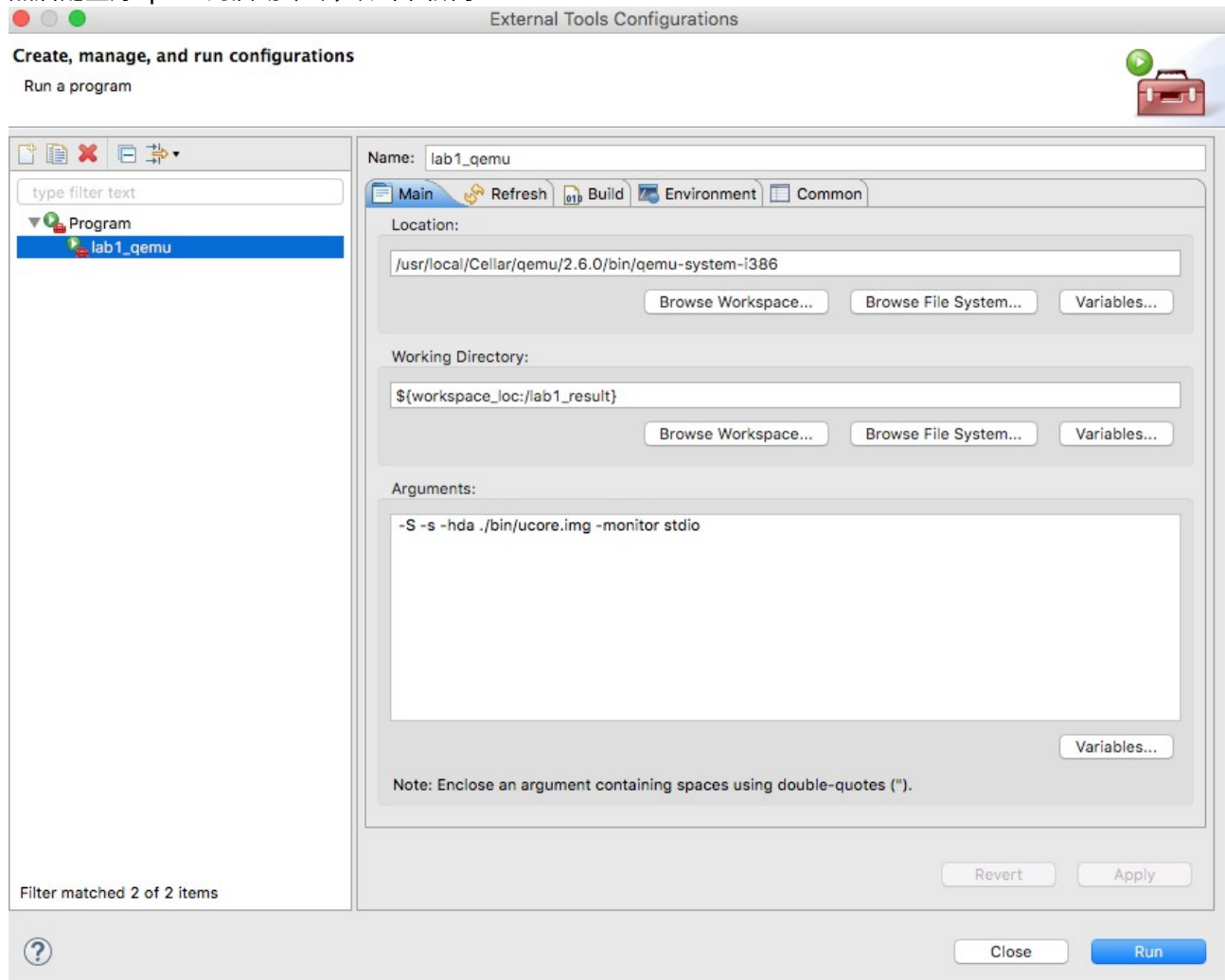




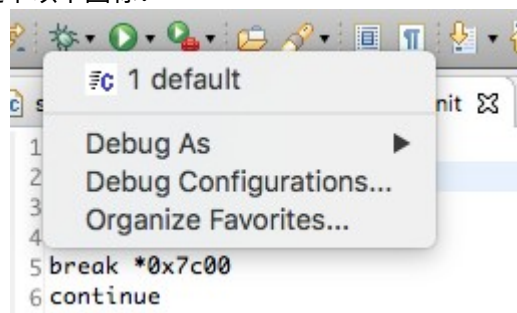
然后配置 qemu 的启动，在工具栏中点击图中按钮，然后选择 External 这个选项。



然后配置好 qemu 的启动命令，如下图所示：



最后配置 gdb 命名，在工具类选中以下图标：



然后选择 Debug Configuration 选项。在左边可以看到我们安装的 Zylin, 在 Native 那个选项上双击，就可新建一个 Zylin 选项，选项的配置如下图所示：

Name:

Main Debugger Commands Source Environment Common

Debugger:

☐ Stop on startup at:

Debugger Options

Main

GDB debugger:

GDB command file:

(Warning: Some commands in this file may interfere with the startup operation of the debugger, for example "run".)

GDB command set:

Protocol:

☐ Verbose console mode

☐ Use full file path to set breakpoints

Name:

Main Debugger Commands Source Environment Common

[Help/tips on how to setup GDB init script](#)

'Initialize' commands

```
file bin/kernel
target remote :1234
set architecture i8086
break *0x7c00
continue
x /2i $pc
```

'Run' commands

好了，先启动 qemu ,然后点那个 debug 就可以 debug 了，Console 中可以键入 gdb 命令。

祝各位顺利！