

# 【VSCode LLVM Compiler Explorer】VSCode LLVM编译器浏览器

码路工人 码路工人 2024年09月10日 01:06 山东

## VSCode LLVM Compiler Explorer

公众号 · 码路工人

### 项目简介

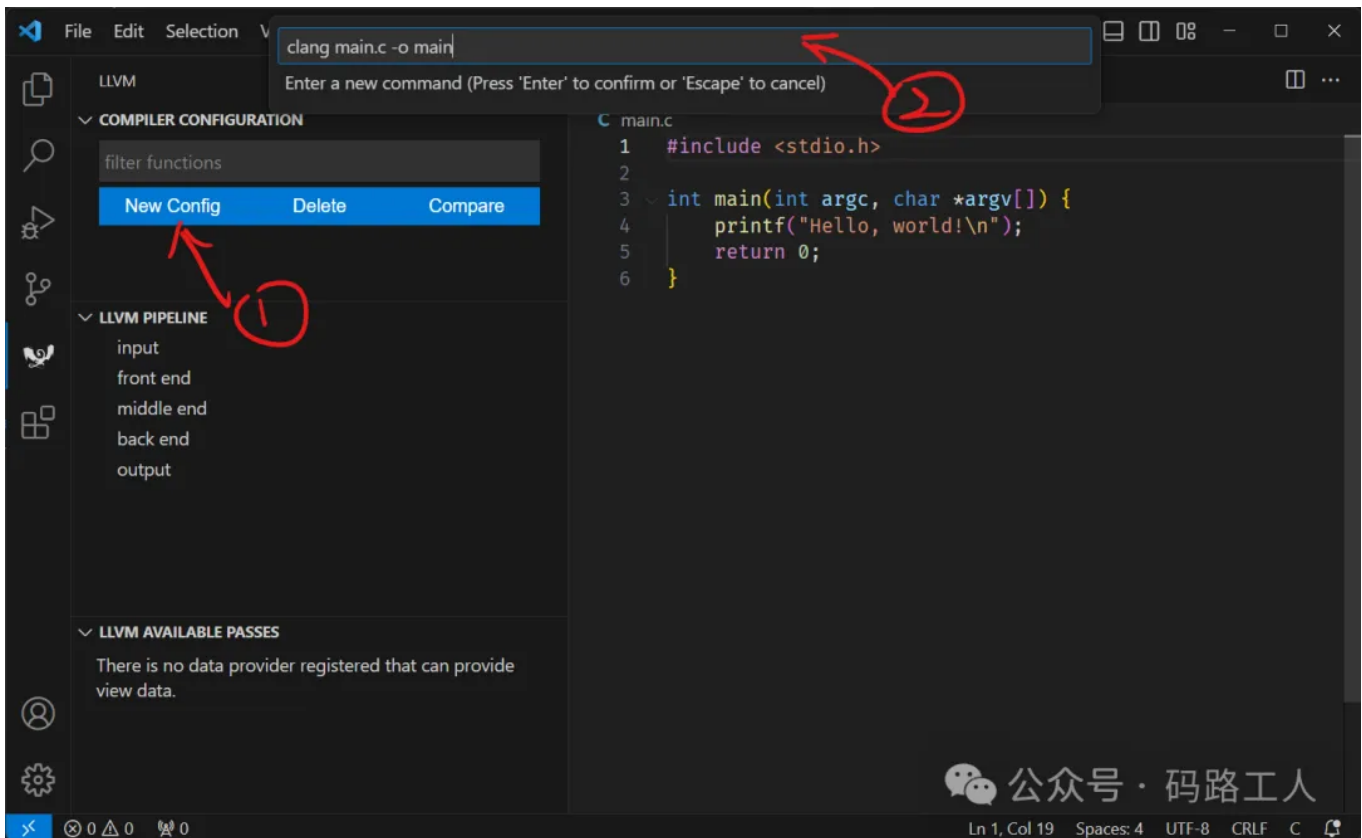
从 VSCode 市场下载本扩展工具

这是一个供LLVM编译器开发人员使用的工具。此 vscode 扩展可以支持在每次传递后探索 LLVM IR 和机器 IR。

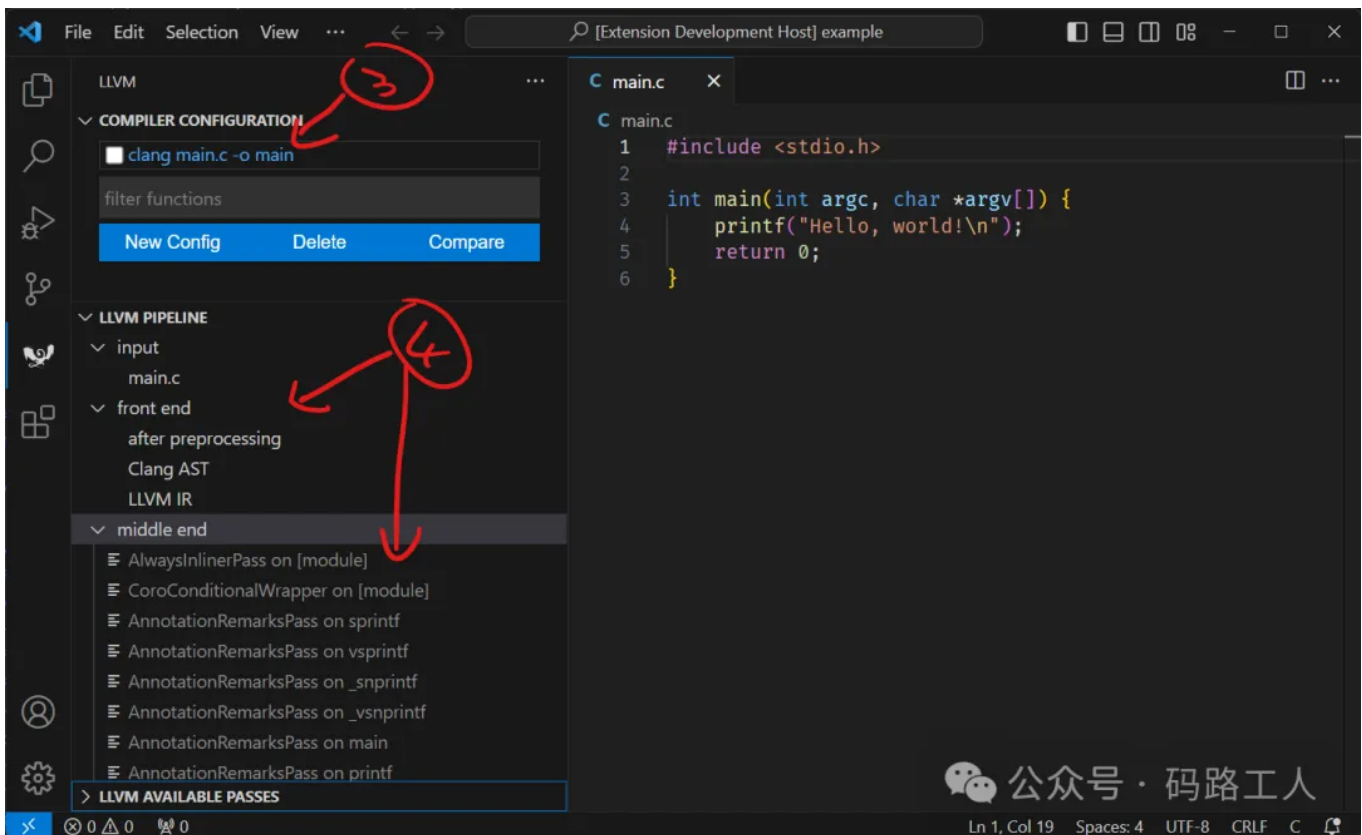
### 功能特性

1. 运行 clang 命令并探索预处理阶段、clang AST 构建阶段、LLVM 传递的每个阶段以及最终生成的汇编代码。
2. 比较运行一次之前和之后 IR 之间的差异。
3. 支持自定义 clang 或修改版本。

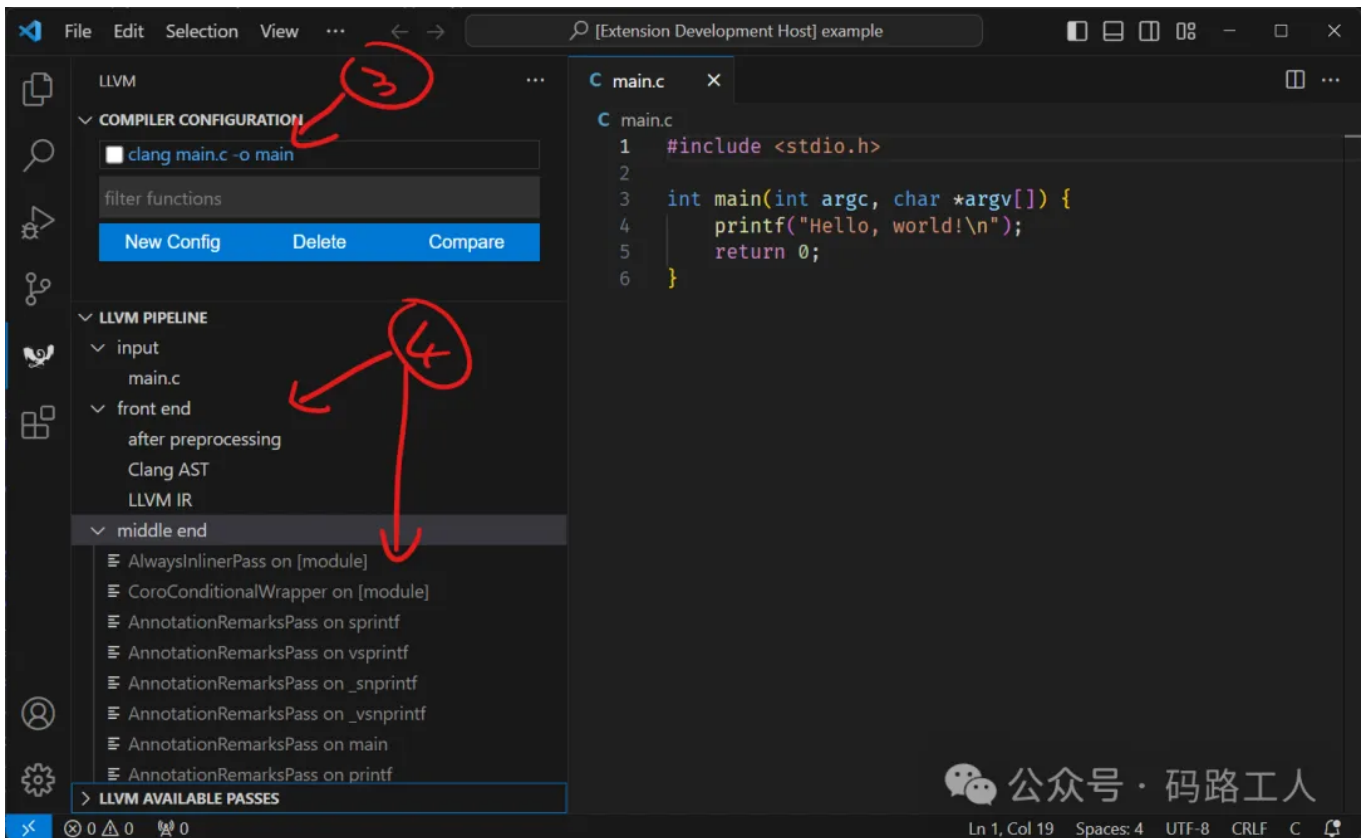
### 如何使用



1. 单击“新配置”以创建新配置
2. 输入命令来编译文件

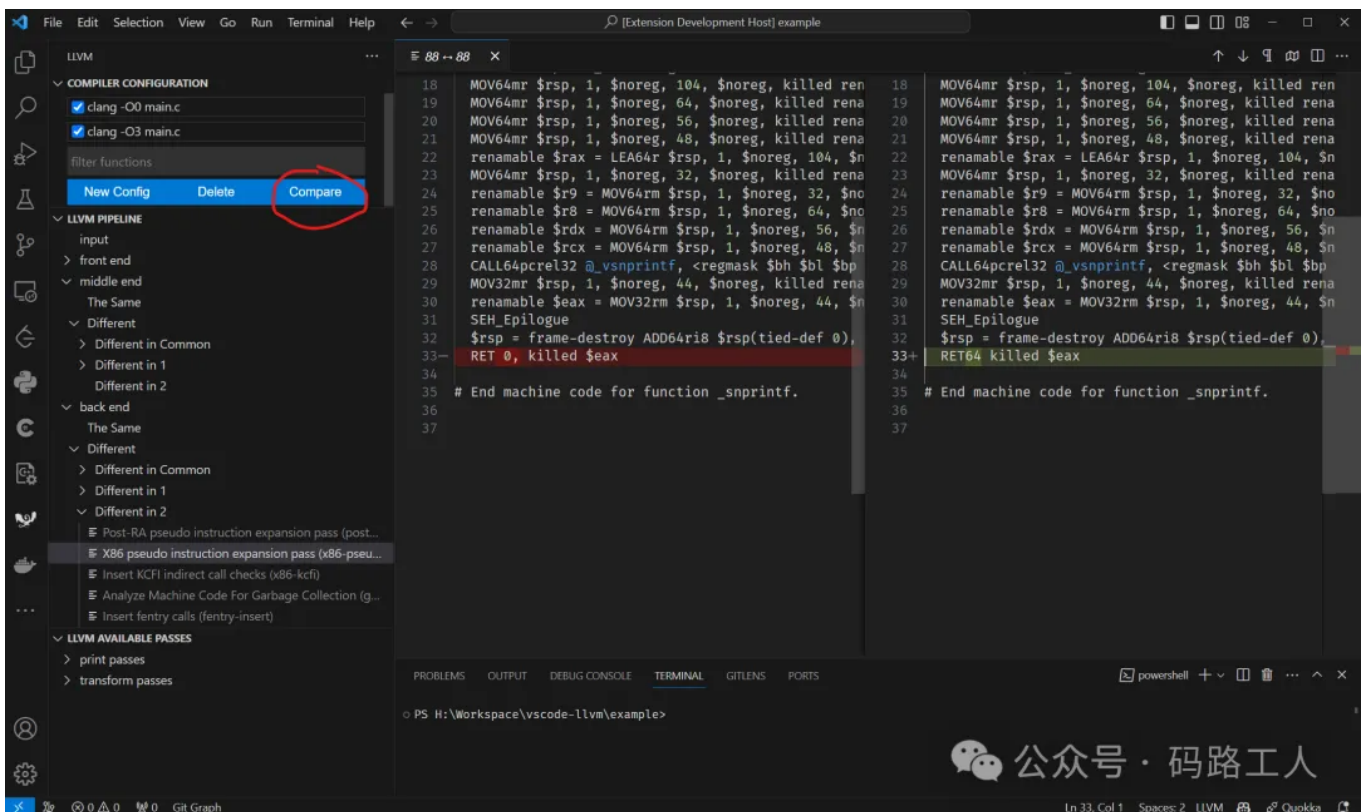


3. (可选) 如果您想专注于一项功能。您可以在“过滤函数”字段中输入函数名称。请注意，如果是 C++，这个函数名称应该是 mangle 名称。然后，单击命令名称来运行该命令。
4. 现在，您可以在每次通过后探索 IR。



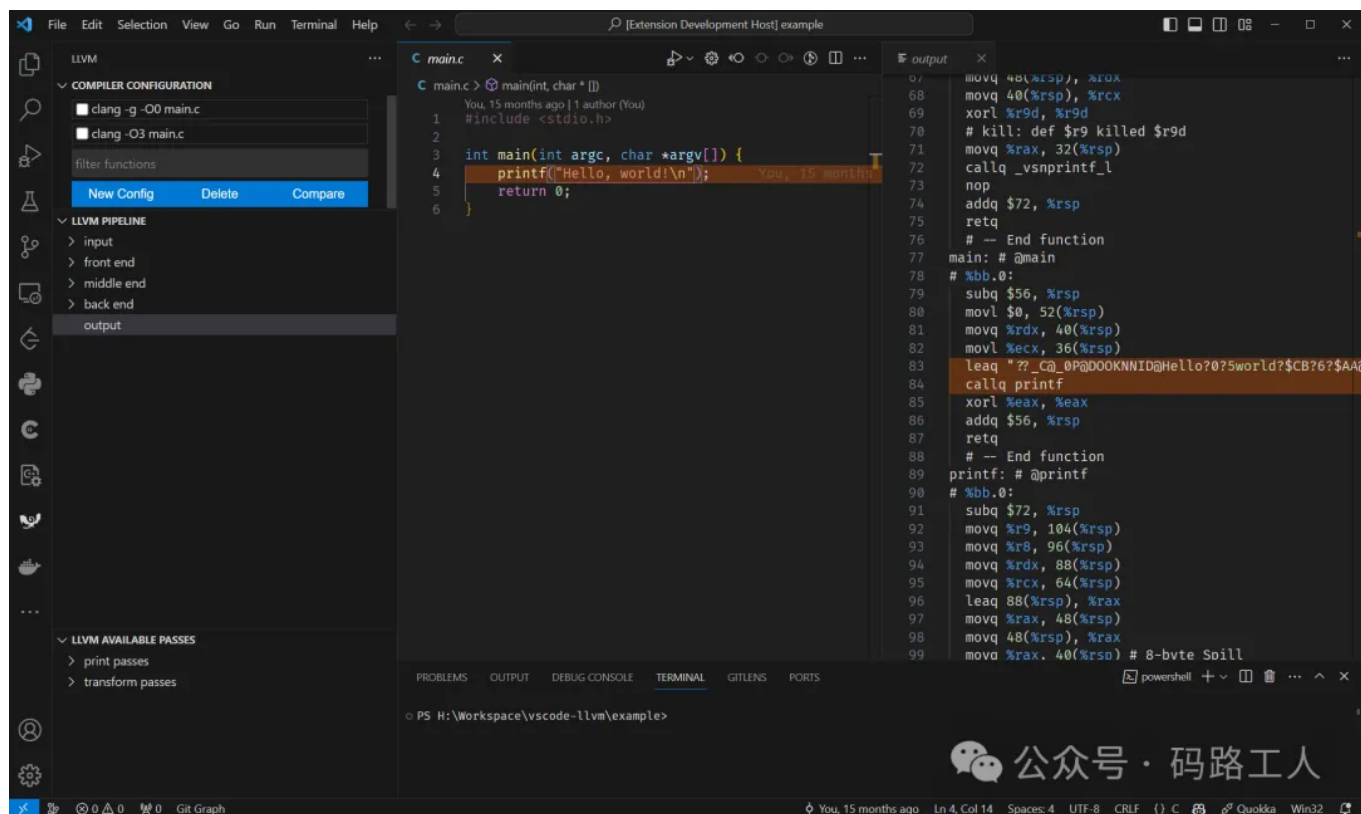
## 比较模式

要调试 pass，您可能需要比较带有和不带有 pass 的 clang 命令。或者您想用稳定版本调试不同版本的 clang。此扩展可以帮助您在每次通过后比较两个 IR 之间的差异。



## Source2Asm 映射视图

受 Compiler Explorer 的启发，该扩展可以突出显示源代码和汇编代码之间的映射。

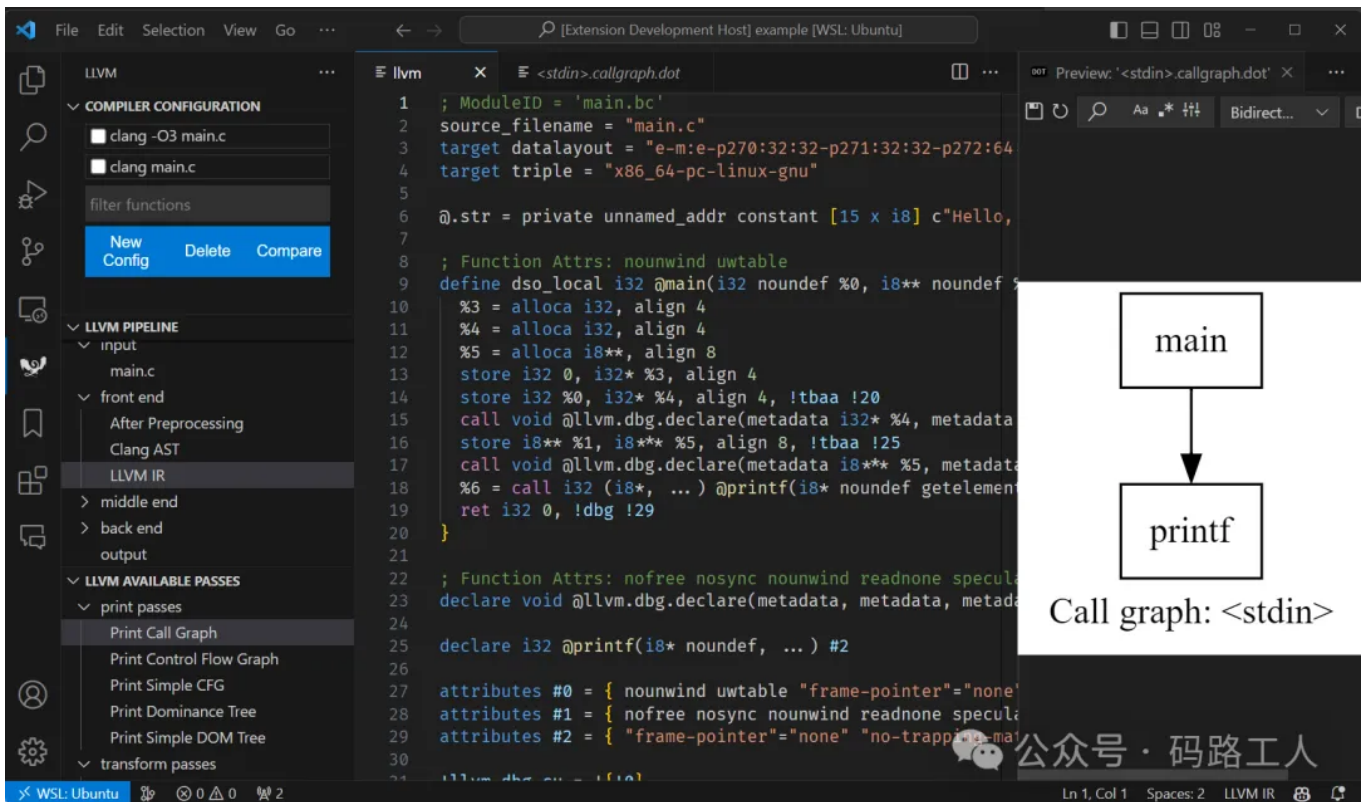


## 运行 Print Pass 来查看 CallGraph/CFG/DOMTree

该插件中有一个可用的实用程序通行证列表。您可以运行这些遍来查看 IR 文件的 CallGraph、CFG 和 DOMTree。

打开 IR 文件时，可以单击“打印调用图”按钮来生成 callgraph.dot 文件。然后，您可以使用 Graphviz 插件来查看调用图。

注意：您需要在 PATH 中包含“opt”工具（或在设置中指定路径）并安装 Graphviz 预览插件。



## 语法高亮

该插件为 Clang AST 和汇编语言提供语法高亮显示。您可能需要安装额外的 LLVM 语法高亮插件，有一些选项：

- RReverser.llvm
- colejcummins.llvm-syntax-highlighting

## 项目链接

<https://github.com/sunxfancy/vscode-llvm>