2023/4/28 调试c语言.md

## ###在vscode上调试C语言

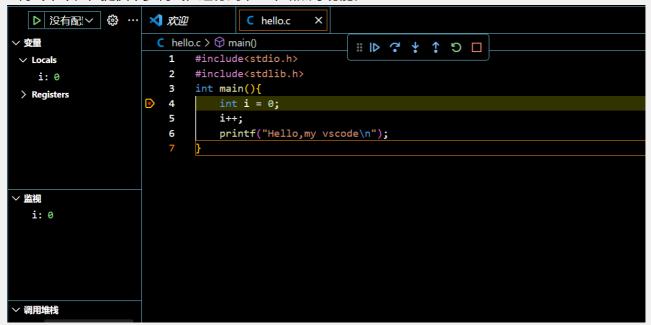
## 利用一个简单的C程序演示在vscode上进行简单的C语言调试 1.添加断点 类似于dev c++中的操作,在右边行号处添加断点即可 #include<stdio.h> 2 #include<stdlib.h> int main(){ int i = 0; 4 5 i++; 6 printf("Hello,my vscode\n"); 2.使用安装的mingw开始调试 下图右上角运行符号右边有个小三角,点开后选择调试试 X 调试c语言.md \$ ₩ ... C hello.c 调试 C/C++ 文件 C hello.c > .. #include<stdio.h> 运行 C/C++ 文件 #include<stdlib.h> int main(){ int i = 0; 5 i++; 6 printf("Hello,my vscode\n"); 选择调试配置,这里选择第一个(之后这个文件的调试会默认选择此项) \$ ₩ ... 资源管理器 C/C++: gcc.exe 生成和调试活动文件 preLaunchTask: C/C++: gcc.exe 生成活动文件 ✓ 打开的编辑器 1 个未保存 检测到的任务 (编译器: C:\mingw64\bin\gcc.exe) 🗙 欢迎 ₽ (gdb) 启动 • C hello.c (Windows) 启动 品 ✓ FOR FUN 2 int i = 0; C hello.c printf("Hello,my vscode\n");

调试开始,这里有三个地方需要关注一下

a.最左边的变量区

调试c语言.md 2023/4/28

b.变量区下面的监视区,可以添加需要监视的变量。添加方法有两个:一是在"监视"两个字后面有一个加号(这里没有显示出来);二是在代码区双击选中变量后右键,下拉菜单会有"添加到监视选项" c.调试命令栏,提供单步调试,运行到下一个断点等功能。



另外,输出结果会在终端显示交互

如果有什么差错,别问我,我也不会,我只是刚刚发现了这个很简单的东西,可恶啊,被这个很简单的信息差卡了一个学期。