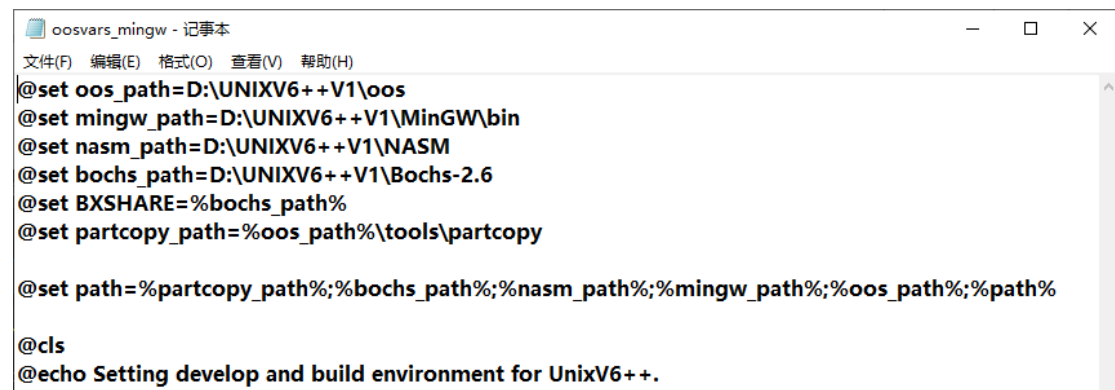


使用 VSCode 配置 GDB 的远程开发环境

By 邓蓉

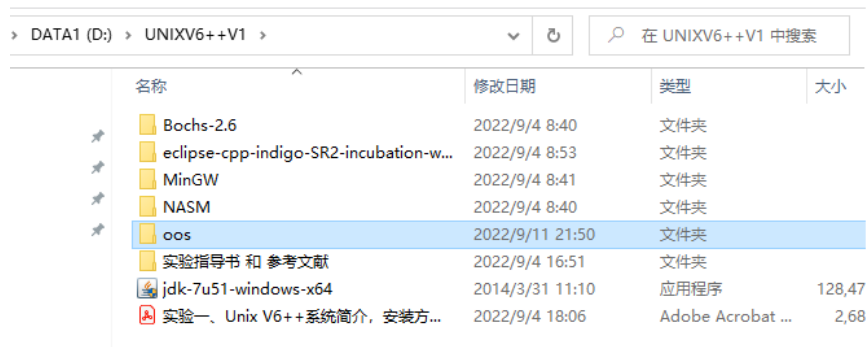
1、把路径名中的空格全部去掉



```
@set oos_path=D:\UNIXV6++V1\oos
@set mingw_path=D:\UNIXV6++V1\MinGW\bin
@set nasm_path=D:\UNIXV6++V1\NASM
@set bochs_path=D:\UNIXV6++V1\Bochs-2.6
@set BXSHARE=%bochs_path%
@set partcopy_path=%oos_path%\tools\partcopy

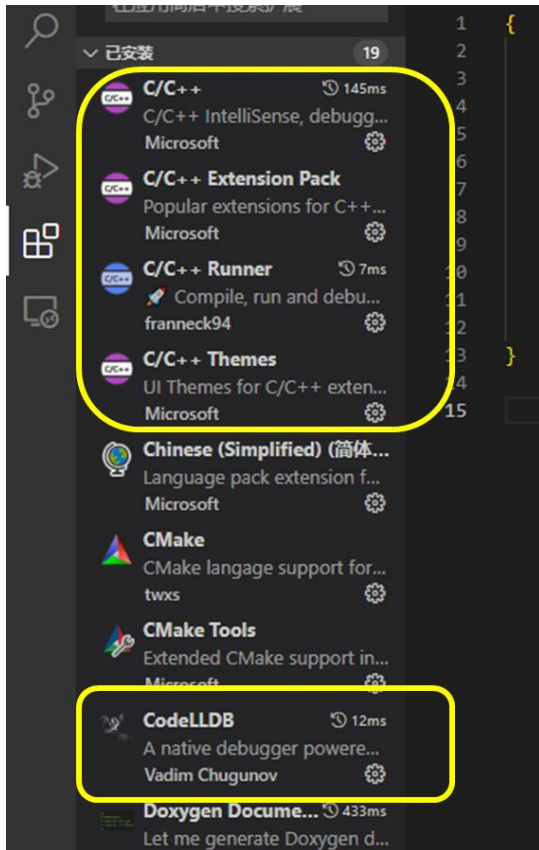
@set path=%partcopy_path%;%bochs_path%;%nasm_path%;%mingw_path%;%oos_path%;%path%

@cls
@echo Setting develop and build environment for UnixV6++.
```

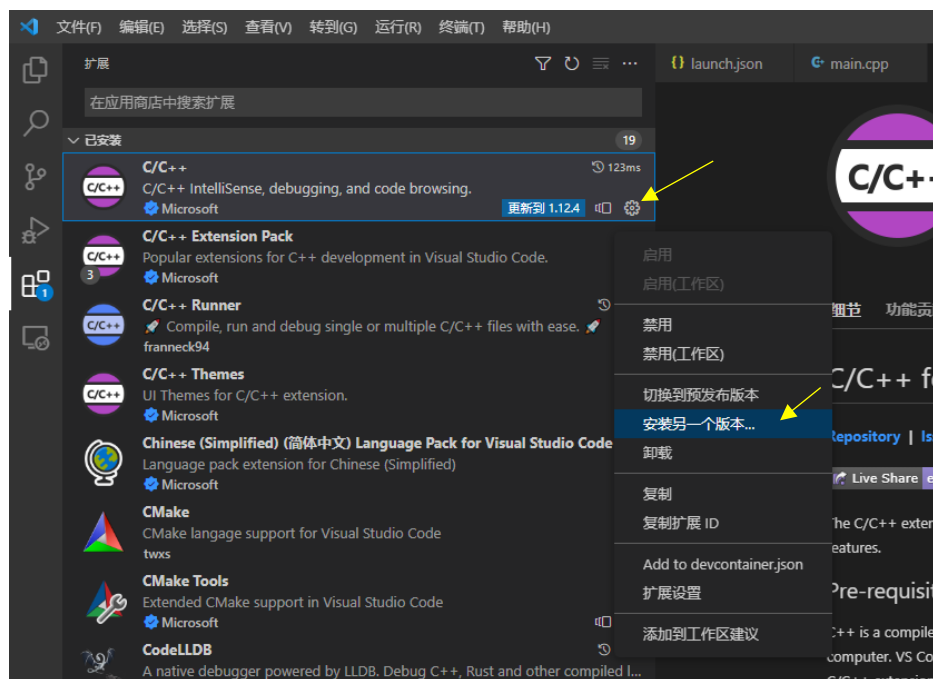


DATA1 (D:) > UNIXV6++V1 >					在 UNIXV6++V1 中搜索	
	名称	修改日期	类型	大小		
★	Bochs-2.6	2022/9/4 8:40	文件夹			
★	eclipse-cpp-indigo-SR2-incubation-w...	2022/9/4 8:53	文件夹			
★	MinGW	2022/9/4 8:41	文件夹			
★	NASM	2022/9/4 8:40	文件夹			
★	oos	2022/9/11 21:50	文件夹			
	实验指导书 和 参考文献	2022/9/4 16:51	文件夹			
	jdk-7u51-windows-x64	2014/3/31 11:10	应用程序	128,47		
	实验一、Unix V6++系统简介, 安装方...	2022/9/4 18:06	Adobe Acrobat ...	2,68		

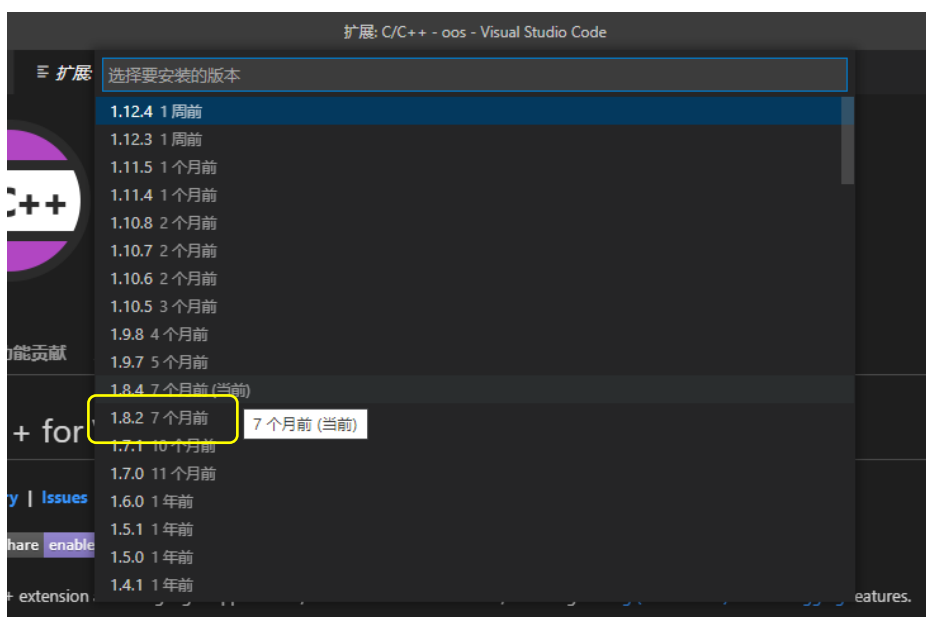
2、为 VSCode 装插件



3、新版的 C/C++ 插件不会自动生成 .vscode 文件夹和 launch.json 文件。我们退回旧版本。方法是：



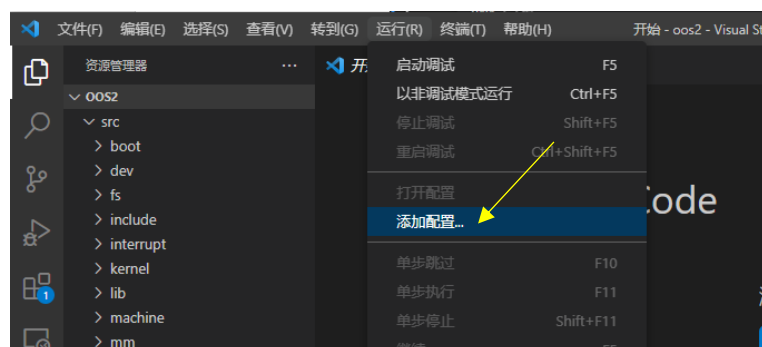
我装了 1.8.2。

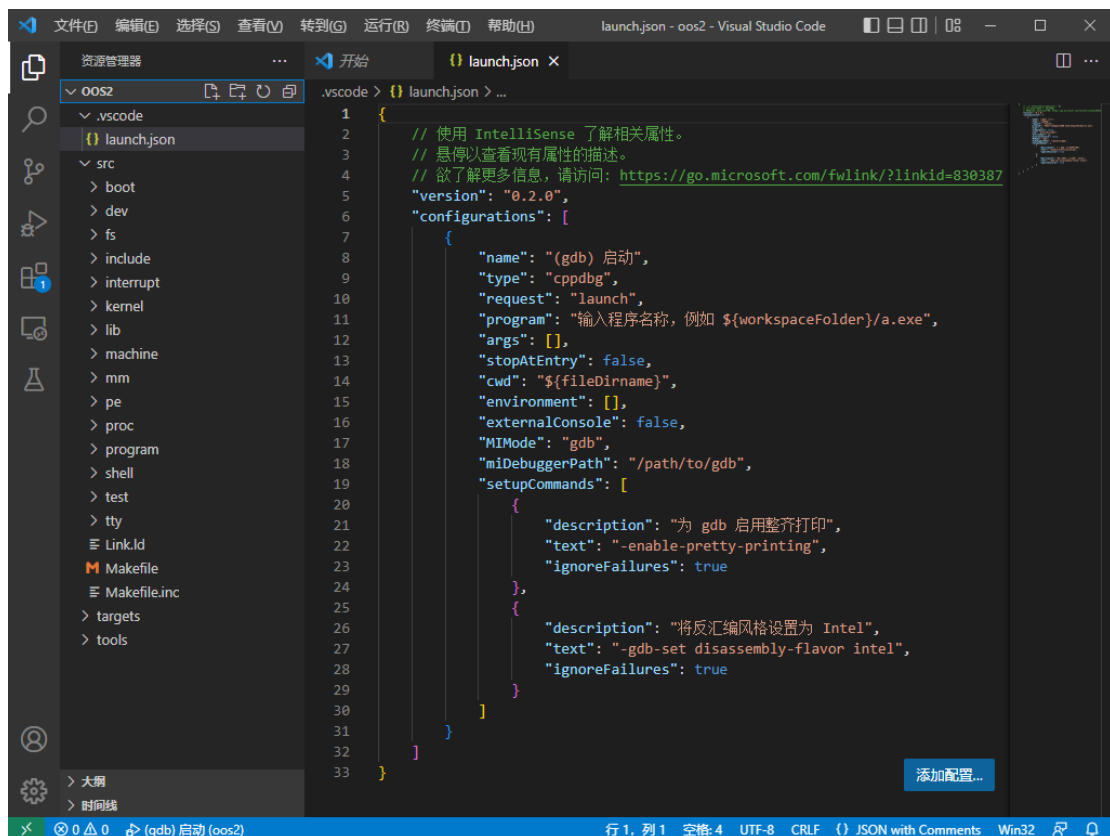
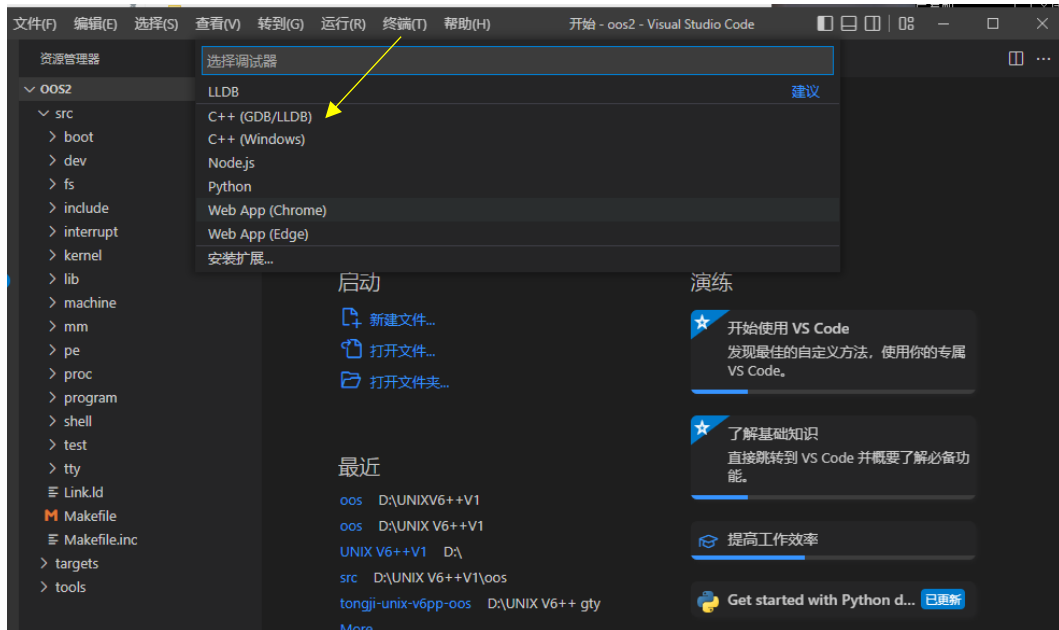


关闭 VSCode。

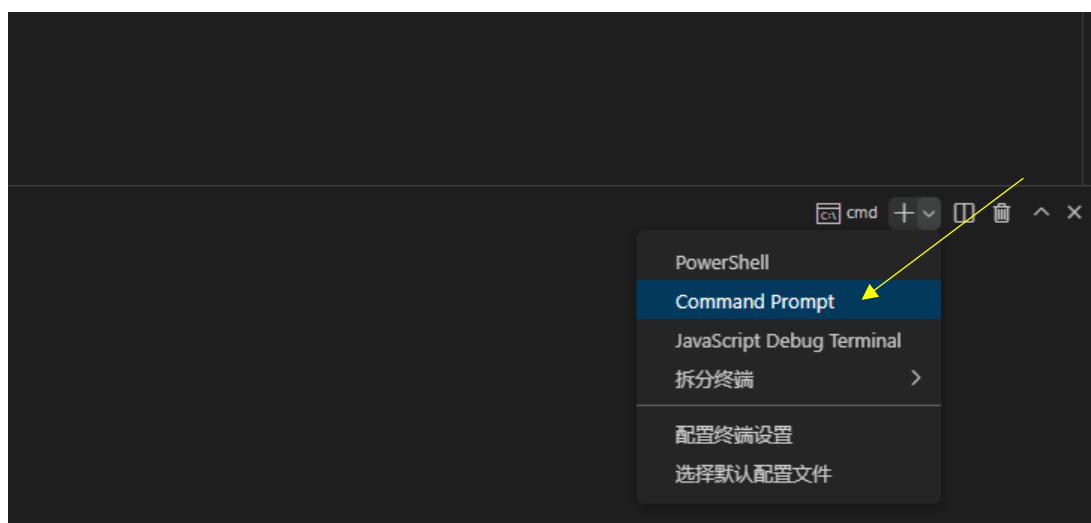
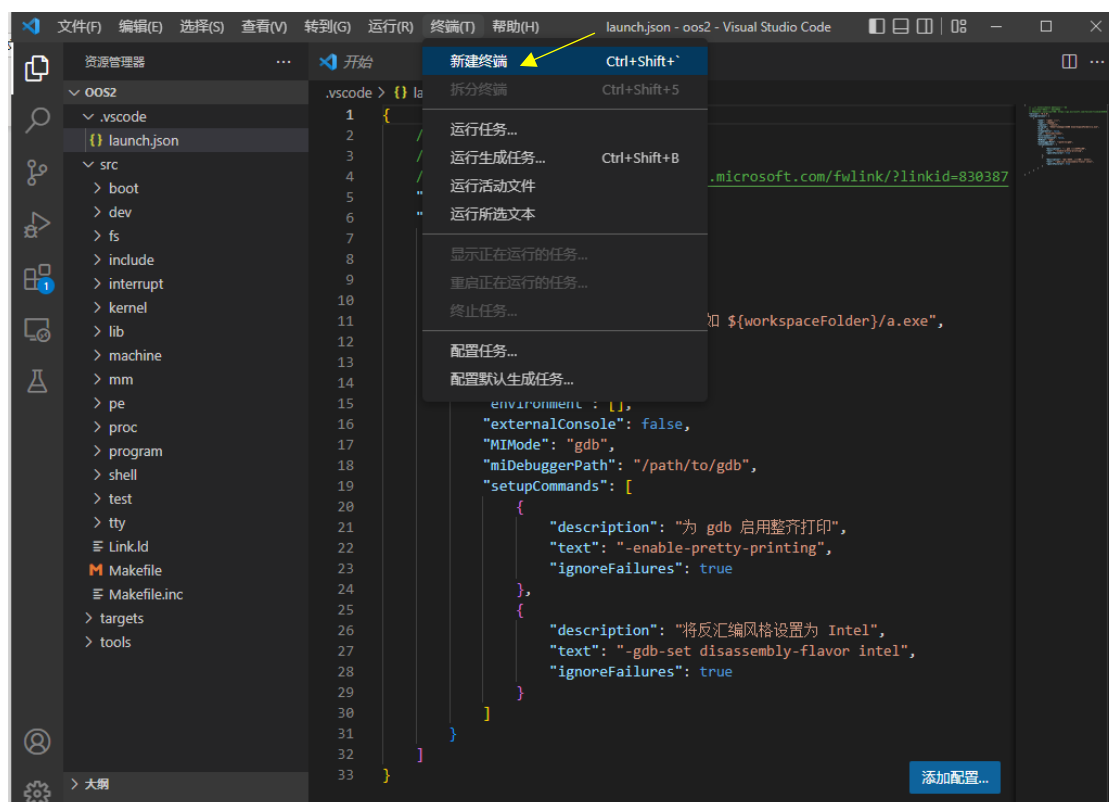
4、用 code 打开文件夹 oos。

运行，添加配置。选 C++(GDB/LLDB)。之后，系统会自动生成.vscode 文件夹和一个 launch.json 文件。

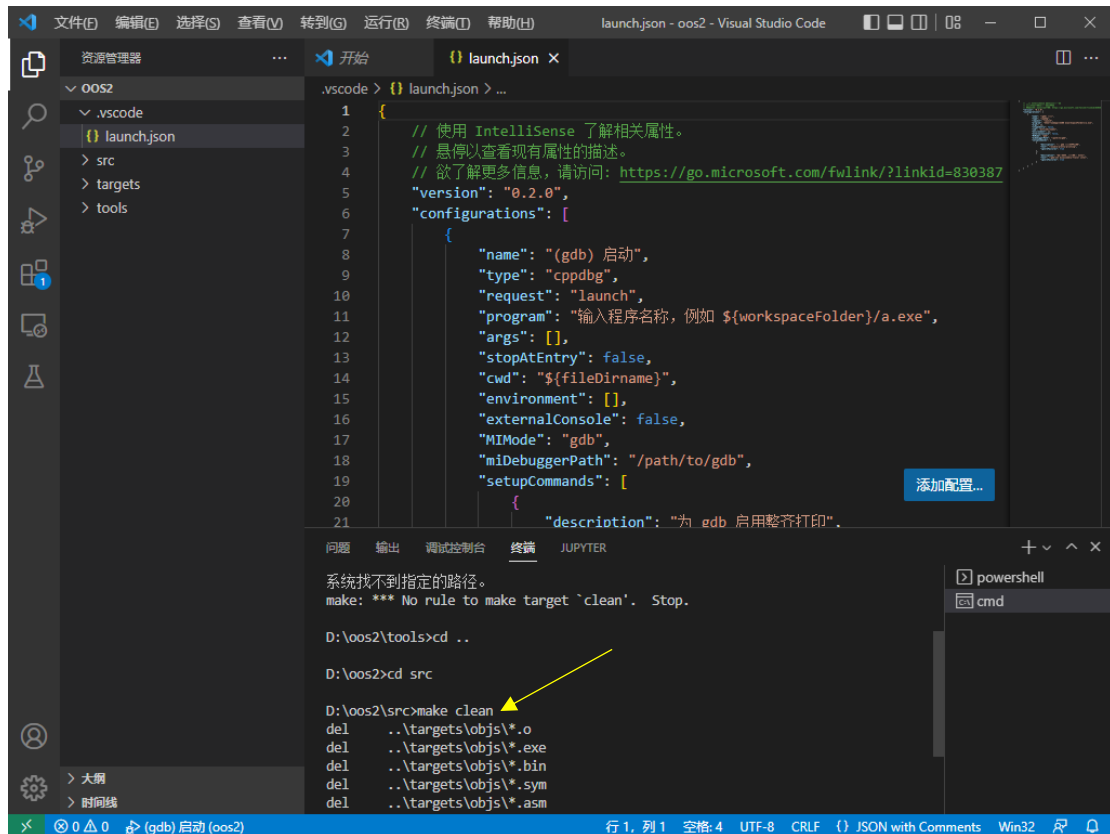




5、新建一个终端。选 cmd



尝试在终端 make 内核



make all, 可以成功。

- 6、将刘吉加的 task.json 和 launch.json 复制进.vscode 目录, 修改。
删掉系统自动生成的 launch.json。

6.1 配 make all

我的 task.json:

```
{
  "tasks": [
    { // 生成任务, 使用 Ctrl + Shift + B 编译
      "type": "shell",
      "label": "build all",
      "command": "cd ${workspaceFolder}\\src;make clean;make all",
      "args": [],
      "detail": "调用 Make 编译 OOS",
      "group": "build"
    },
  ],
}
```

```

"version": "2.0.0"
}

```

6.2 配置 gdb 远程调试

放弃 debug 的前置任务，oos run。

刘吉加的 launch.json 删掉选中的部分。

```

launch - 记事本
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)
// 以下采用陈元哲同学的方法
{
  "version": "0.2.0",
  "configurations": [
    {
      "name": "Debug",
      "type": "cppdbg",
      "request": "launch",
      "miDebuggerServerAddress": "localhost:1234",
      "program": "${workspaceFolder}/targets/objs/kernel.exe",
      "args": [],
      //"stopAtEntry": true,
      "cwd": "${workspaceFolder}",
      "externalConsole": false,
      "MIMode": "gdb",
      "preLaunchTask": "OOS run", // 前置任务: 启动Bochs
    }
  ]
}

```

7、重新编译内核的方法

资源管理器选中 src。ctrl+shift+b。

有 bug。系统重新编译内核后会执行 run 命令，终端按任意键结束即可。

8、调试内核的方法

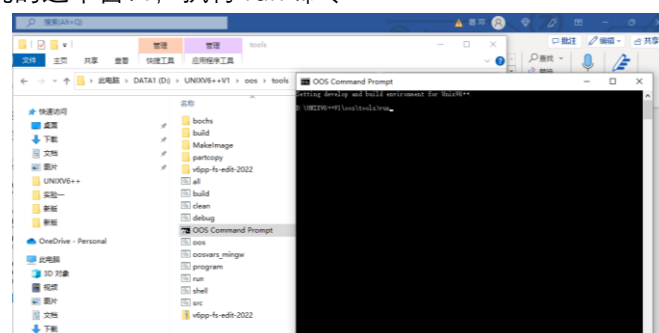
去掉 gdbstub 之前的 #号。变成这样

```

bochsrc - 记事本
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)
#####
# bochsrc.txt file for DLX Linux disk image.
#####
gdbstub: enabled=1, port=1234, text_base=0, data_base=0, bss_base=0

```

回到基础开发环境的这个窗口，执行 run 命令



好啦，bochs 虚拟机在 1234#端口等调试命令。

VSCode 里下断点。F5，开始调试。

