在Windows使用VSCode搭建嵌入式Linux开发环境

百问网已经制作好了完备的Ubuntu镜像，可以从这里下载：

```
链接：https://pan.baidu.com/s/1vw4VUV_MvtOHXz8IC66ACg
提取码：iftb
```

如果网盘链接无效了，可以加QQ群联系我们：341014981

我们也正在(2022.10.17开始)使用纯粹的Ubuntu环境开始教驱动入门，免费的，感兴趣者也加上面的群。

1. Ubunt上的操作

1.1 安装基本开发工具

```
git clone https://e.coding.net/weidongshan/DevelopmentEnvConf.git
cd DevelopmentEnvConf
sudo ./Configuring_ubuntu.sh
```

1.2 安装bear

```
sudo apt install bear
```

1.3 下载和编译内核

1.3.1 下载内核

执行如下命令：

```
$ git clone https://e.coding.net/codebug8/repo.git
$ mkdir -p 100ask_imx6ull-sdk && cd 100ask_imx6ull-sdk
$ ../repo/repo init -u https://gitee.com/weidongshan/manifests.git -b linux-sdk
-m imx6ull/100ask_imx6ull_linux4.9.88_release.xml --no-repo-verify
$ ../repo/repo sync -j4
```

1.3.2 配置工具链

执行如下命令：

```
gedit  ~/.bashrc
```

在最后加入如下内容：

```
export ARCH=arm
export CROSS_COMPILE=arm-buildroot-linux-gnueabihf-
export PATH=$PATH:/home/book/100ask_imx6ull-sdk/ToolChain/arm-buildroot-linux-gnueabihf_sdk-buildroot/bin
```

重新关闭、打开终端。

1.3.3 编译内核

vscode的clangd插件使用compile_commands.json文件来生成索引文件，这样当我们点击某个函数时可以飞快跳转到它定义的地方。

compile_commands.json文件中记录的是每个文件的编译选项，样式如下：

```
        "arguments": [
            "arm-buildroot-linux-gnueabihf-gcc",
            "-c",
            "-Wp,-MD,init/.main.o.d",
            "-nostdinc",
            "-isystem",
            "/home/book/100ask_imx6ull-sdk/ToolChain/arm-buildroot-linux-gnueabihf_sdk-buildroot/bin/../lib/gcc/arm-buildroot-linux-gnueabihf/7.5.0/include",
            "-I./arch/arm/include",
            "-I./arch/arm/include/generated/uapi",
            "-I./arch/arm/include/generated",
            "-I./include",
            "-I./arch/arm/include/uapi",
            "-I./include/uapi",
            "-I./include/generated/uapi",
            "-include",
            "./include/linux/kconfig.h",
            "-D__KERNEL__",
            "-mlittle-endian",
            "-Wall",
            "-fno-dwarf2-cfi-asm",
            "-fno-omit-frame-pointer",
            "-o",
            "init/.tmp_main.o",
            "init/main.c"
        ],
        "directory": "/home/book/100ask_imx6ull-sdk/Linux-4.9.88",
        "file": "init/main.c"
    },
```

我们使用bear命令来生成compile_commands.json，它的用法如下：

```
bear make [其他make本身的参数]
```

它会记录make过程编译文件时用到的命令。

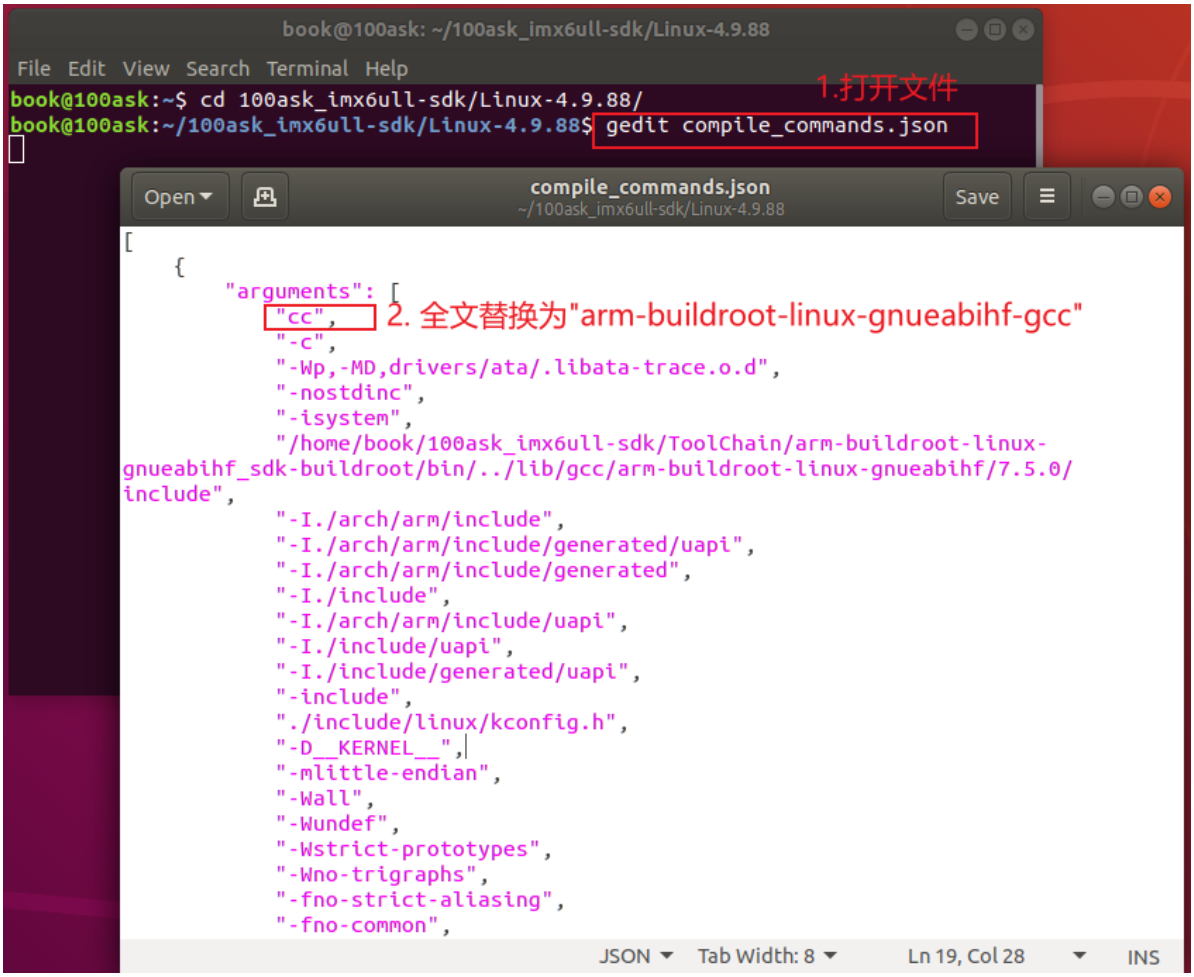
所以我们编译内核的目的是生成compile_commands.json，执行如下命令：

```
$ cd /home/book/100ask_imx6ull-sdk/Linux-4.9.88
$ make 100ask_imx6ull_defconfig
$ bear make zImage -j4
```

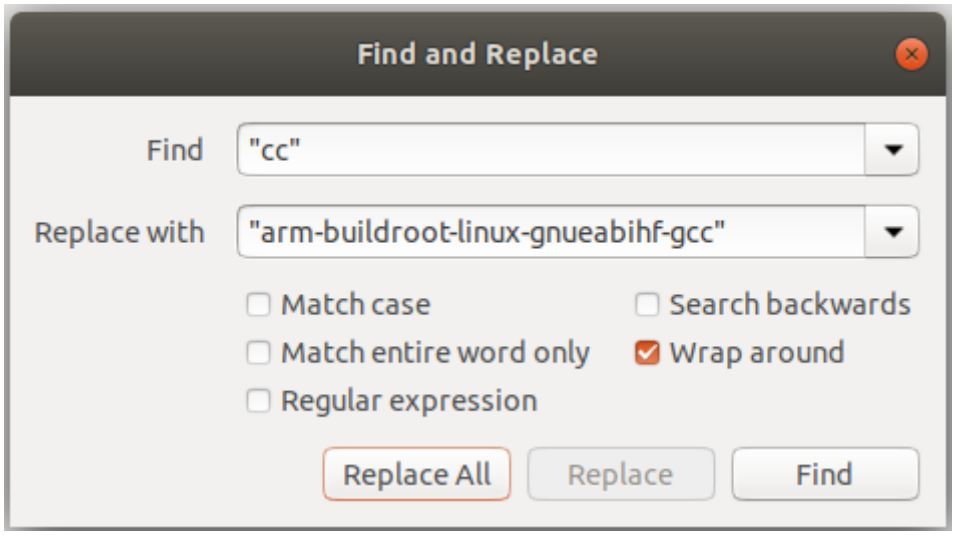
如果你之前曾经编译过内核但是没有在前面使用bear命令，那么需要重新编译：

```
$ make  clean
$ bear make zImage -j4
```

编译成功后就会在当前目录下得到文件compile_commands.json，需要如下修改：



在gedit中使用快捷键"Ctrl+H"即可如下操作：



2. Windows上的操作

2.1 安装vscode

2.1.1 从官网下载安装

使用浏览器从从https://code.visualstudio.com/下载vscode安装包，双击安装。

2.1.2 在本地安装插件

我们的目的是在Windows上运行vscode，使用vscode阅读Linux服务器上的内核源码。

这需要安装很多插件，这些插件是安装在windows上还是Linux服务器上？

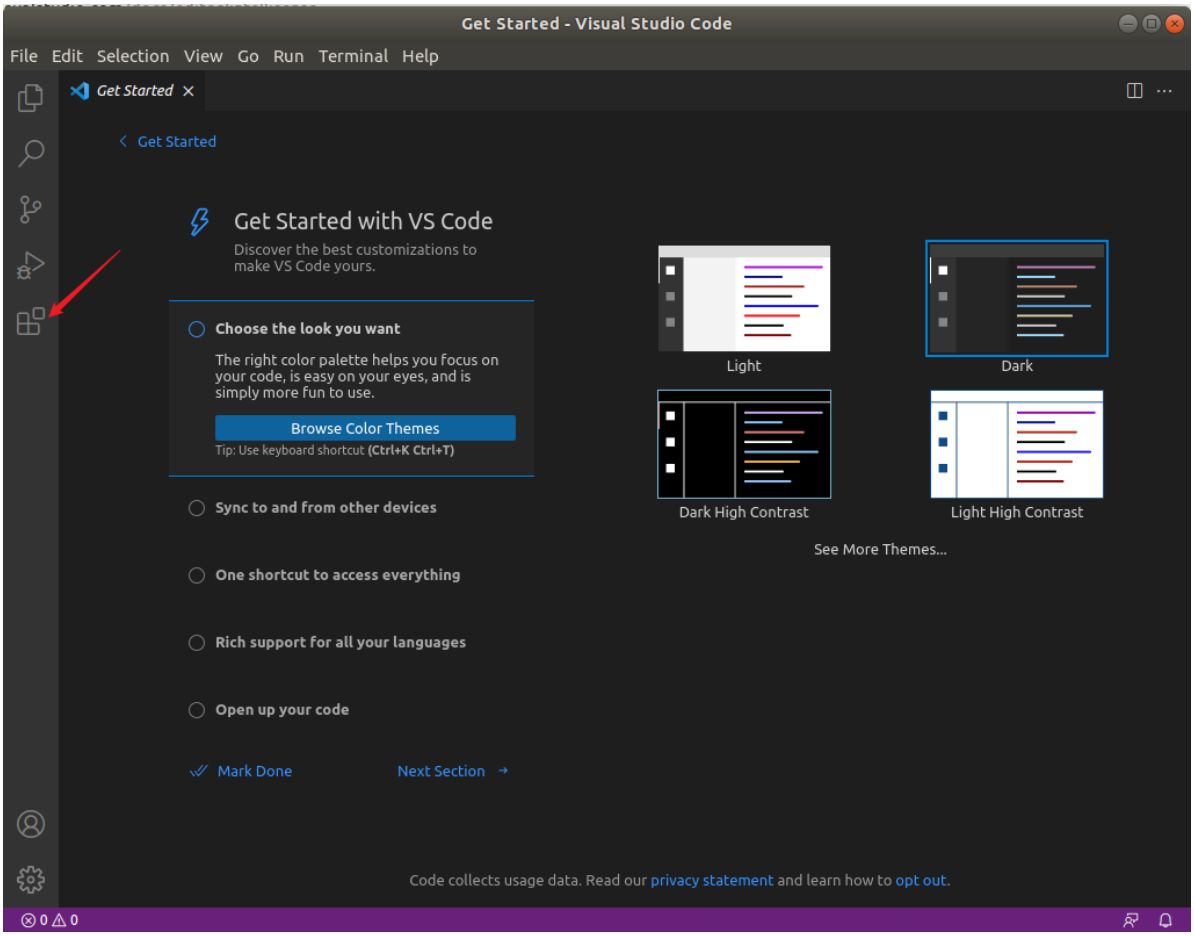
vscode的插件有两种类型：

- 全局插件：只需要安装在Windows上，打开远程服务器的代码后也可以使用这些插件
- 远程插件：即使在Windows上使用vscode，这类插件也必须安装在远程服务器上

但是我们并不知道插件属于全局插件还是远程插件，怎么办呢？

- 先在Windows安装所需的全部插件
- 以后打开远程服务器文件夹时，再查看已经安装的插件，它会有相应的提示。

打开vscode后，点击左侧图标：



依次输入下列插件名字，安装：

- C/C++
- C/C++ Extension Pack
- C/C++ Snippets
- Clangd
- Remote SSH
- Code Runner
- Code Spell Checker
- vscode-icons
- compareit
- DeviceTree
- Tabnine AI Autocomplete
- Bracket Pair Colorization Toggler
- Rainbow Highlighter
  - 高亮文字：shift + alt + z
  - 取消高亮：shift + alt + a
- Arm Assembly
- Chinese
- Hex Editor
- One Dark Pro
- Markdown All in One

- Markdown Preview Enhanced

我们已经安装的插件有这些：

File Edit Selection View Go Run Terminal Help

EXTENSIONS

Search Extensions in Marketplace

INSTALLED 16

Arm Assembly
Arm assembly syntax support for Visual Studio Co...
dan-c-underwood

Bracket Pair Colorization Toggler
Quickly toggle 'Bracket Pair Colorization' setting ...
Dzhavat Ushev

C/C++
C/C++ IntelliSense, debugging, and code browsing.
Microsoft

C/C++ Advanced Lint
An advanced, modern, static analysis extension fo...
Joseph Benden

C/C++ Snippets
Code snippets for C/C++
Harsh

Chinese (Simplified) (简体中文) Language Pack f...
中文(简体)
Microsoft

clangd
C/C++ completion, navigation, and insights
LLVM

Code Runner 29ms
Run C, C++, Java, JS, PHP, Python, Perl, Ruby, Go, ...
Jun Han

Code Spell Checker 69ms
Spelling checker for source code
Street Side Software

compareit 22ms
Compare files
in4margaret

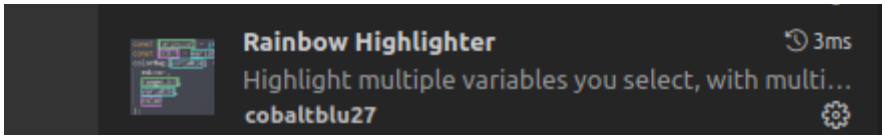
DeviceTree
DeviceTree Language Support for Visual Studio C...
Pietro Lorefice

Hex Editor
Allows viewing and editing files in a hex editor
Microsoft

Markdown All in One
All you need to write Markdown (keyboard shortc...
Yu Zhang

Markdown Preview Enhanced
Markdown Preview Enhanced ported to vscode
Yiyi Wang

One Dark Pro 22ms
Atom's iconic One Dark theme for Visual Studio C...
binaryify



2.2 设置SSH

2.2.1 安装Git

vscode自带的ssh程序有Bug，我们需要替换ssh。

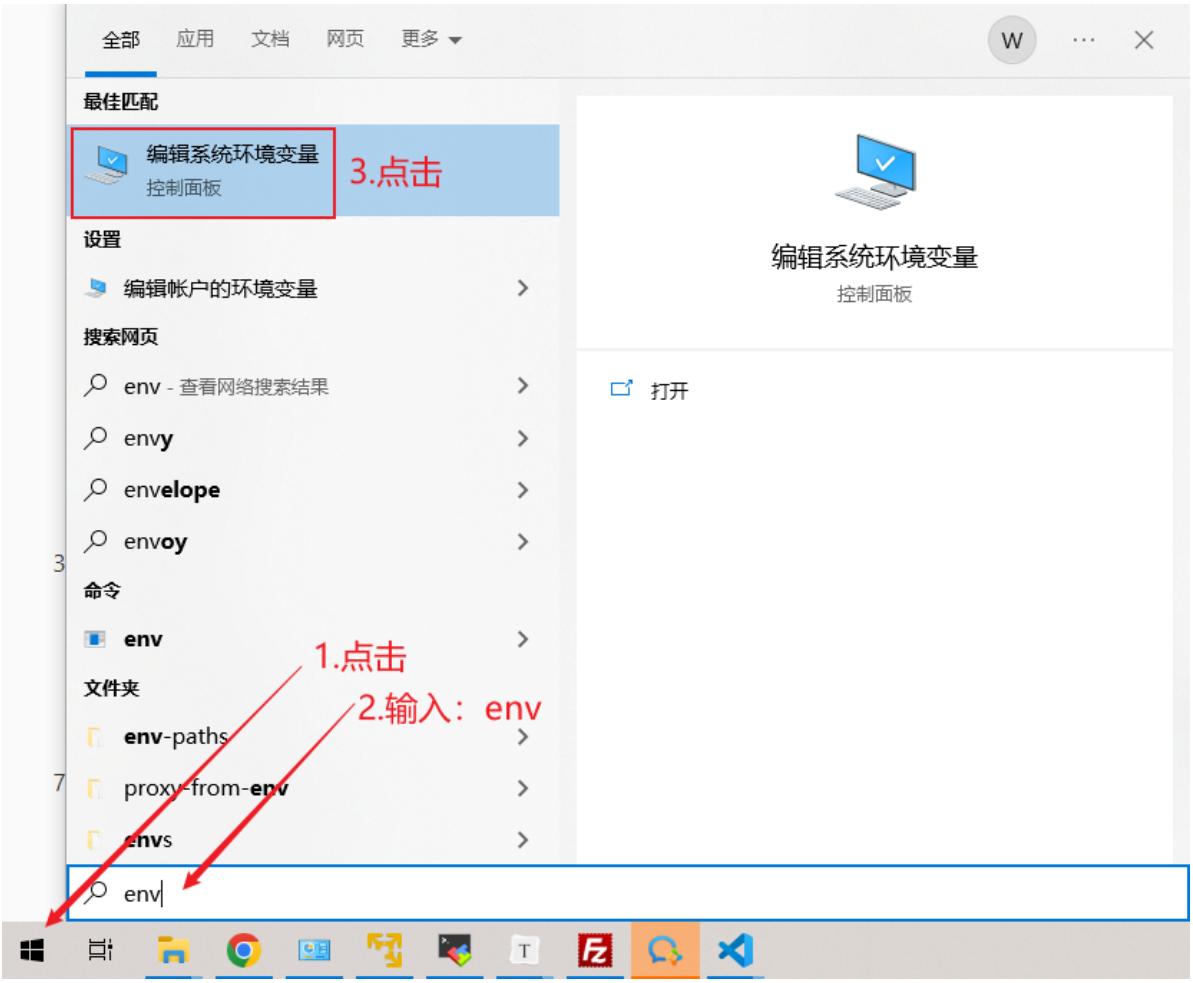
可以使用GIT工具自带的ssh，所以先安装Git：

- 下载：https://gitforwindows.org/
- 安装：双击即可

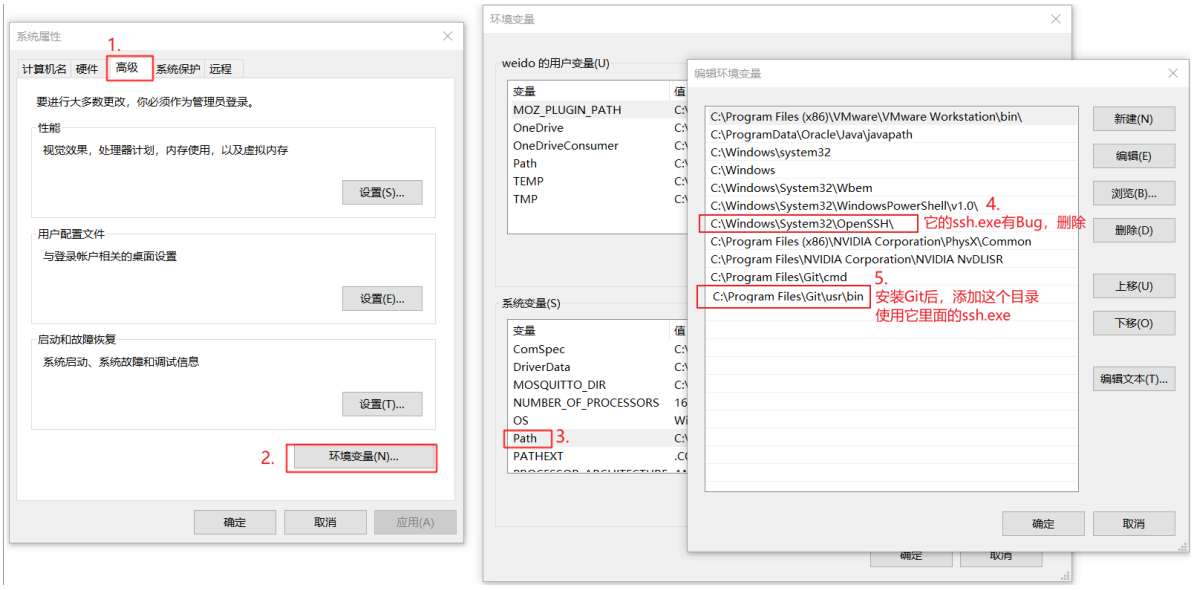
2.2.2 替换ssh

修改环境变量，替换Path中ssh的路径即可。

先打开"编辑系统环境变量"：



然后替换ssh，确保GIT工具的路径下有ssh.exe后，如下替换：



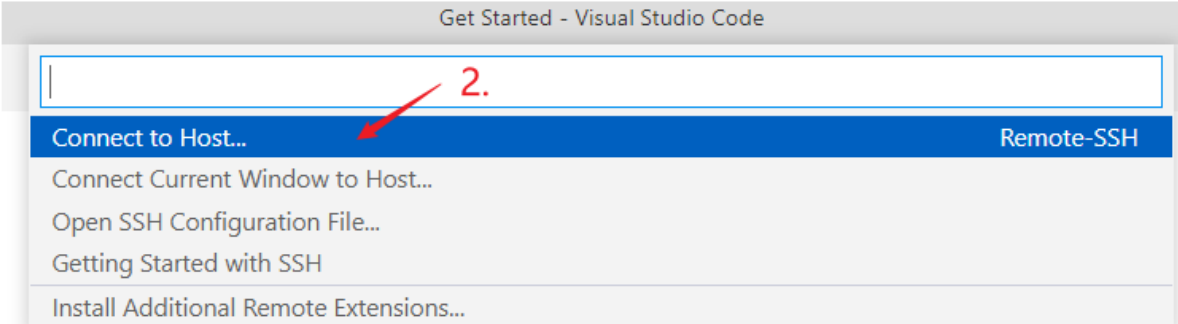
2.3 远程登录服务器

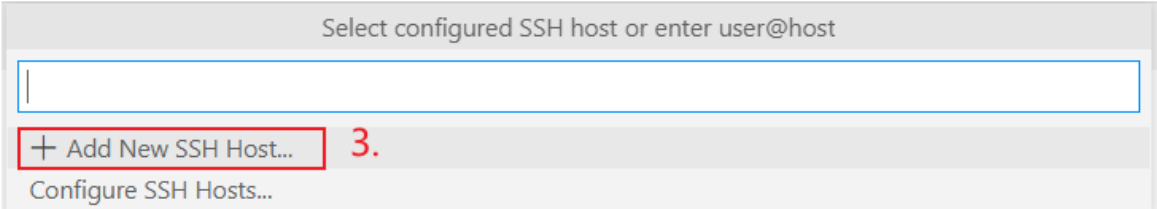
2.3.1 连接Ubuntu

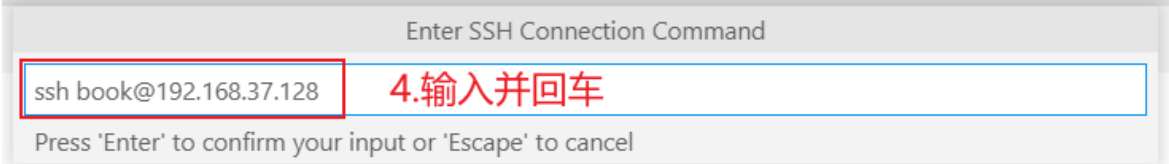
安装好插件后，即可远程登录服务器，如下操作：

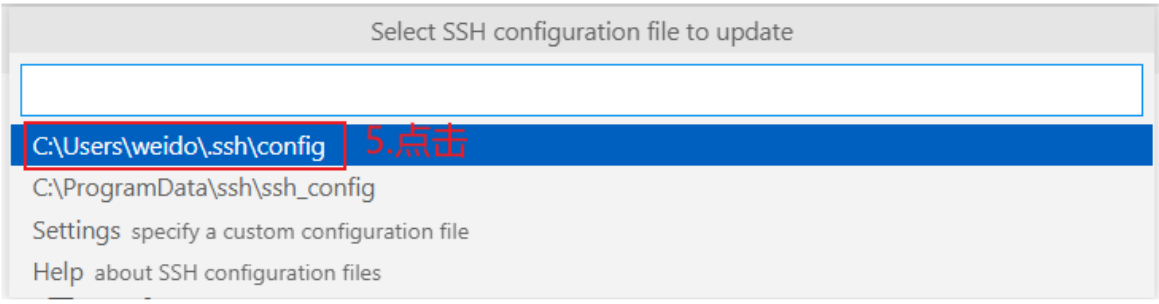
- 先增加Host
- 再连接Host


1.点击
0 0
tabnine starter

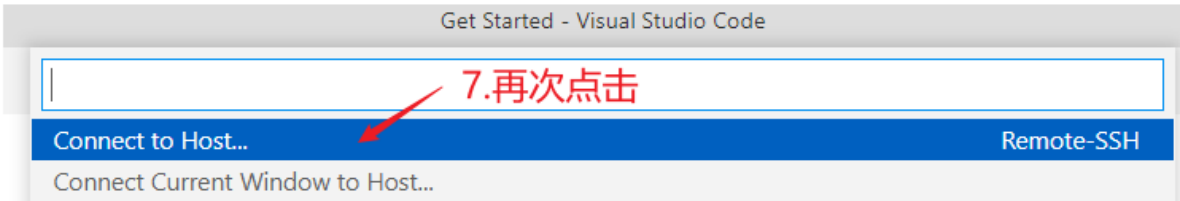

Get Started - Visual Studio Code
2.
Connect to Host...
Remote-SSH
Connect Current Window to Host...
Open SSH Configuration File...
Getting Started with SSH
Install Additional Remote Extensions...

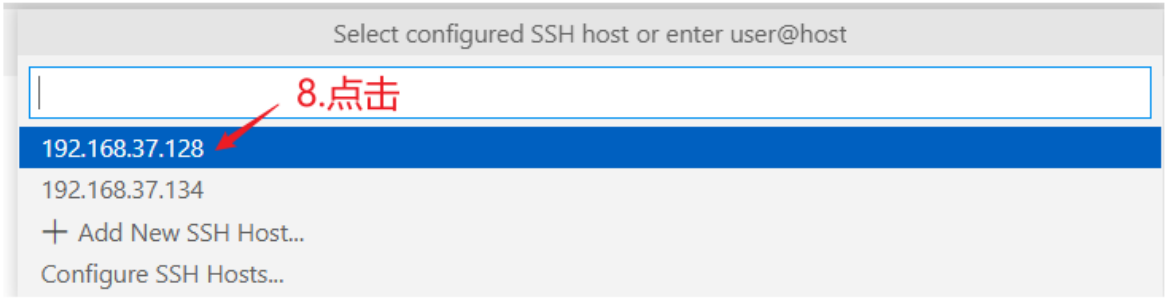

Select configured SSH host or enter user@host
Add New SSH Host...
3.
Configure SSH Hosts...

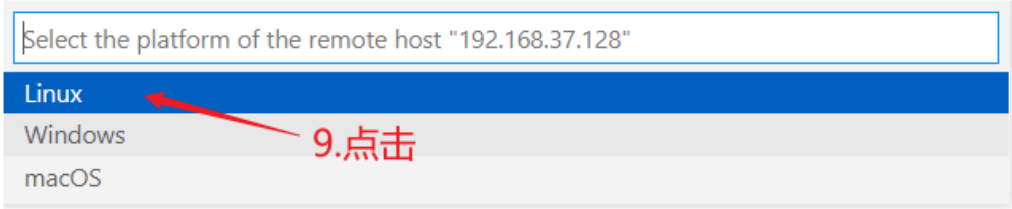

Enter SSH Connection Command
ssh book@192.168.37.128
4.输入并回车
Press 'Enter' to confirm your input or 'Escape' to cancel

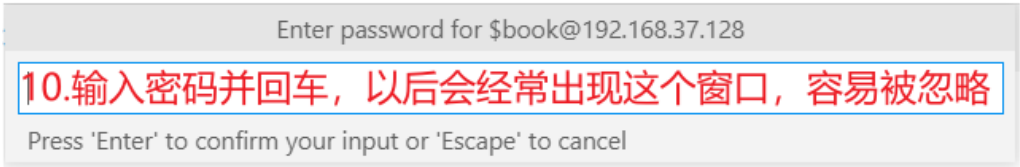

Select SSH configuration file to update
C:\Users\weido\.ssh\config
5.点击
C:\ProgramData\ssh\ssh_config
Settings specify a custom configuration file
Help about SSH configuration files


6.再次点击
0 0
tabnine starter


Get Started - Visual Studio Code
7.再次点击
Connect to Host...
Remote-SSH
Connect Current Window to Host...







2.3.2 免密登录

这不是必须的，后续使用vscode访问远程服务器时，你可以一直使用密码登录。

如果想免密登录的话，需要生成ssh秘钥。

先在windows的命令行执行：

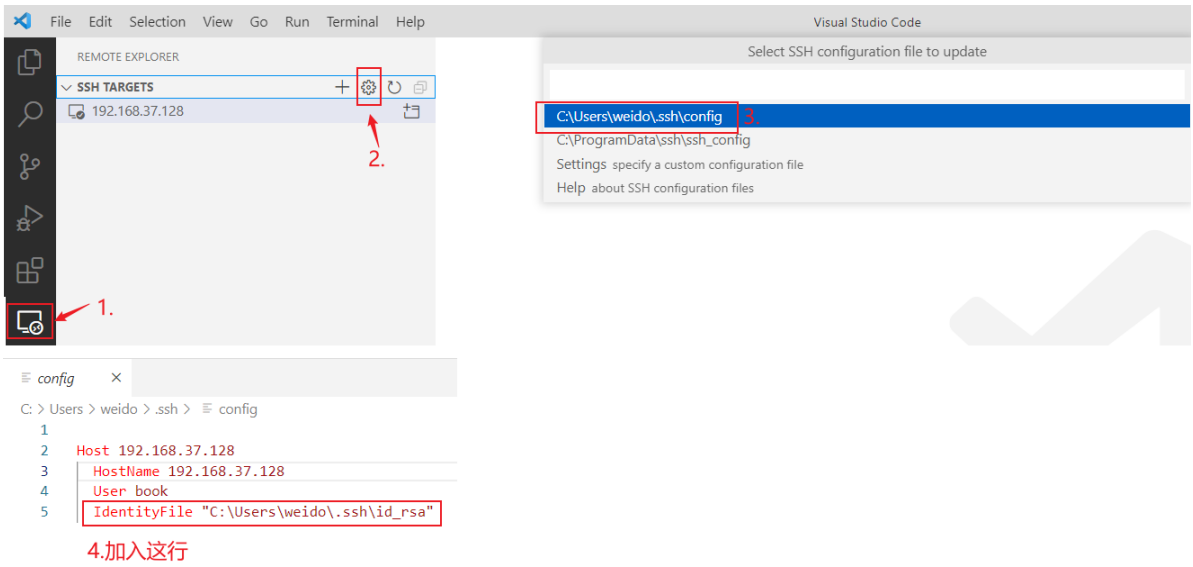
```
ssh-keygen
```

```
Microsoft Windows [版本 10.0.19044.2130]
(c) Microsoft Corporation。保留所有权利。

C:\Users\weido>ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/c/Users/weido/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /c/Users/weido/.ssh/id_rsa.
Your public key has been saved in /c/Users/weido/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:7AVJXYrCNW2wcNbH02v8TChSXo/nTDFOEWVI2JSxpV8 weido@LAPTOP-TTU4PSU2
The key's randomart image is:
+---[RSA 2048]----+
|       . B= o.*+*B|
|      . * ==.* *O |
|       o =..+ +==E|
|        o .. o *o*|
|         S .. o O.|
|        . .      =|
|         .        |
|                  |
|                  |
+----[SHA256]-----+

C:\Users\weido>
```

然后再修改vscode配置：

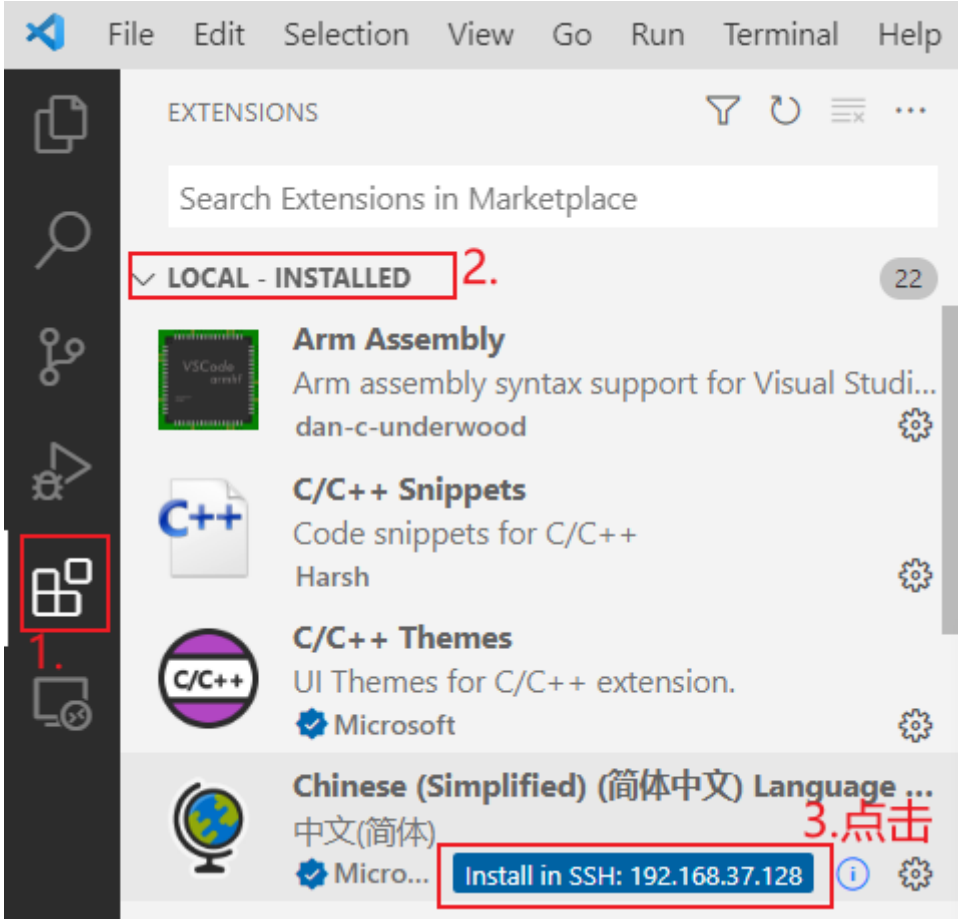


最后把前面生成的id_rsa.pub复制到Ubuntu目录/home/book：

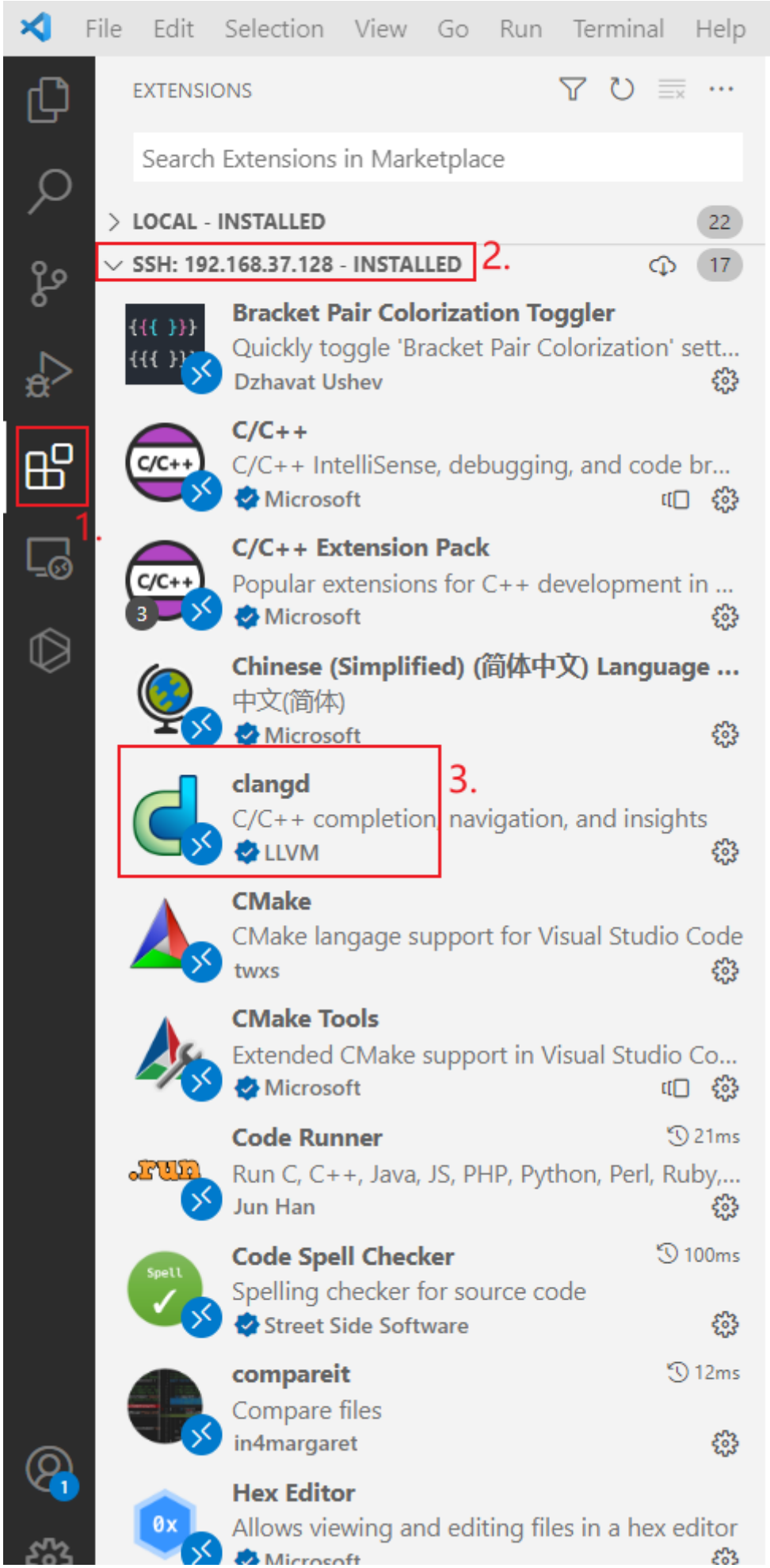
```
mkdir /home/book/.ssh
cat /home/book/id_rsa.pub >> /home/book/.ssh/authorized_keys
chmod 700 /home/book/.ssh
chmod 600 /home/book/.ssh/authorized_keys
sudo /usr/sbin/sshd restart
```

2.4 在服务器上安装插件

vscode连接上服务器后，查看本地插件，发现有如下字样的插件就点击"Install in SSH"：



安装完后，可以如下图查看，确保远程服务器上已经有了clangd插件：


File Edit Selection View Go Run Terminal Help
EXTENSIONS
Search Extensions in Marketplace
LOCAL - INSTALLED 22
SSH: 192.168.37.128 - INSTALLED 2.
17
Bracket Pair Colorization Toggler
Quickly toggle 'Bracket Pair Colorization' sett...
Dzhavat Ushev
C/C++
C/C++ IntelliSense, debugging, and code br...
Microsoft
1.
C/C++ Extension Pack
Popular extensions for C++ development in ...
Microsoft
Chinese (Simplified) (简体中文) Language ...
中文(简体)
Microsoft
clangd
3.
C/C++ completion, navigation, and insights
LLVM
CMake
CMake langage support for Visual Studio Code
twxs
CMake Tools
Extended CMake support in Visual Studio Co...
Microsoft
Code Runner
21ms
Run C, C++, Java, JS, PHP, Python, Perl, Ruby,...
Jun Han
Code Spell Checker
100ms
Spelling checker for source code
Street Side Software
compareit
12ms
Compare files
in4margaret
Hex Editor
Allows viewing and editing files in a hex editor
Microsoft

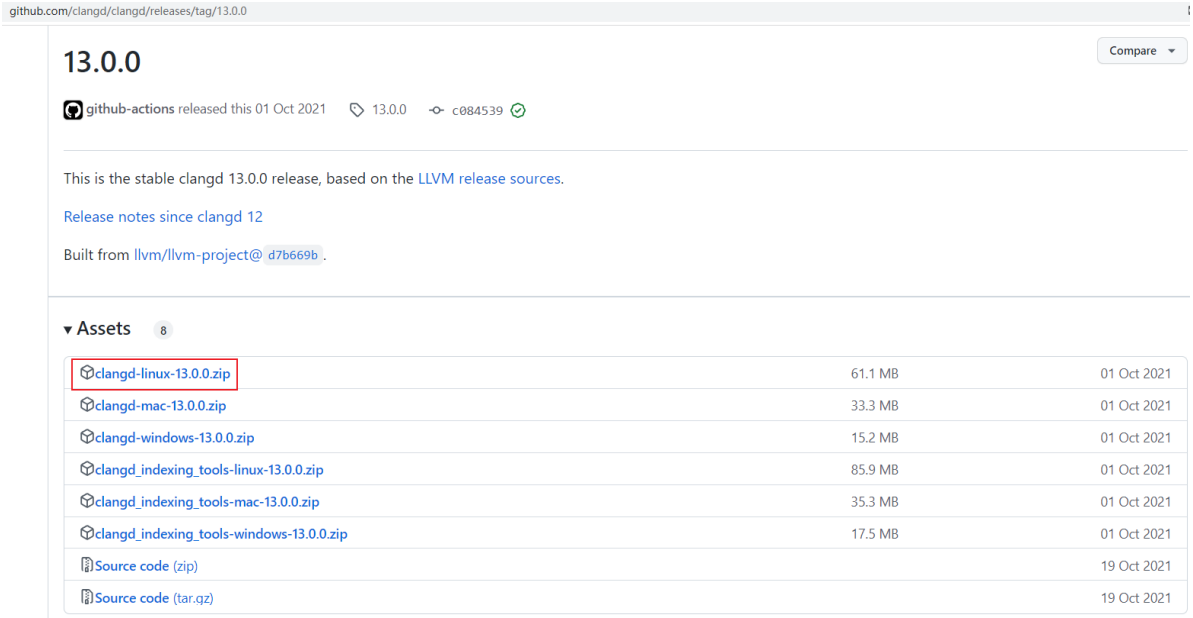
2.5 配置clangd

2.5.1 下载clangd

前面只是安装clangd插件，它的使用还需要一个运行在Linux服务器上的clangd程序。

我们以后使用vscode打开C文件时，会提示你安装clangd程序，它会安装最新版本(版本15)，但是这个版本有一些Bug，所以我们手工安装版本13。

在Ubuntu中使用浏览器打开https://github.com/clangd/clangd/releases/tag/13.0.0，下载Linux安装包：

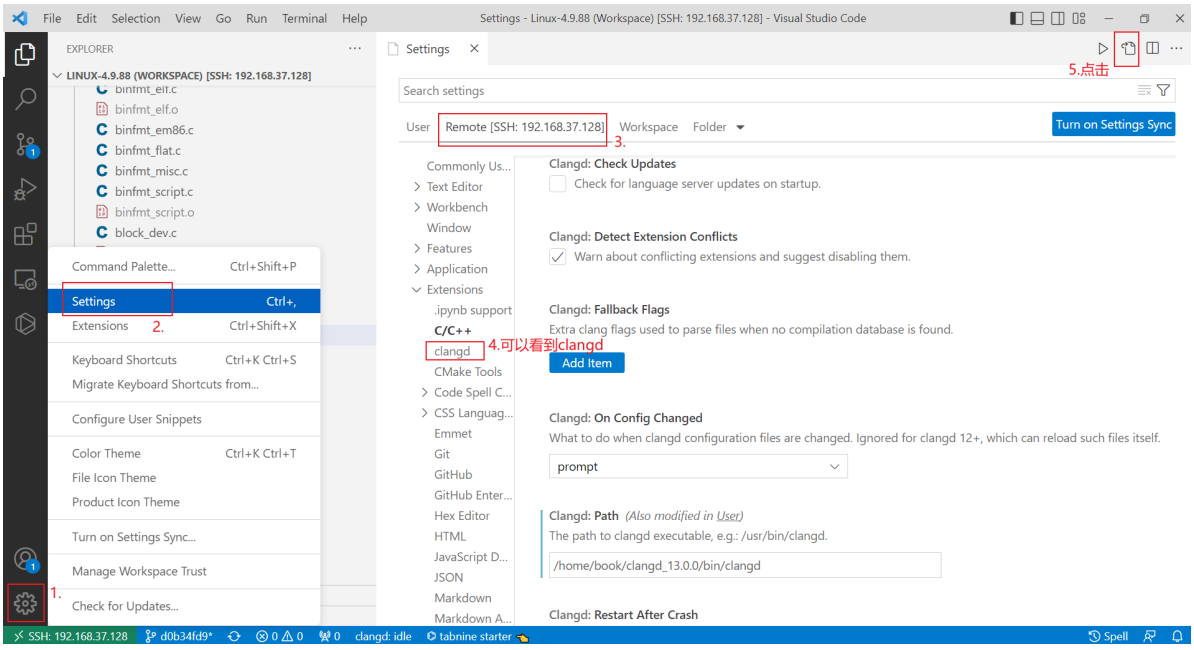


把下载到的clangd-linux-13.0.0.zip放到/home/book目录下，执行解压命令：

```
cd /home/book
unzip clangd-linux-13.0.0.zip
```

2.5.2 配置clangd

在Windows的vscode界面按下图步骤打开setting.json文件:



在setting.json中写入如下内容(我们第1次打开源码目录后，这个文件可能被自动修改，你需要再次修改它)：

```
{
    "C_Cpp.default.intelliSenseMode": "linux-gcc-arm",
    "C_Cpp.intelliSenseEngine": "Disabled",
    "clangd.path": "/home/book/clangd_13.0.0/bin/clangd",
    "clangd.arguments": [
        "--log=verbose",
    ],
}
```

C/C++插件里的intellisense和clangd是冲突的，如果我们没有手工设置setting.json，当使用vscode打开C文件时也会提示禁止intellisense，点击鼠标即可禁止。它的本质也是修改setting.json，它会写入如下文字：

```
"C_Cpp.intelliSenseEngine": "disabled",
```

上面文件有Bug，其中的"disabled"应该改为"Disabled"。

2.6 常用快捷键

打开C文件后，在文件里点击右键就可以看到大部分快捷键。

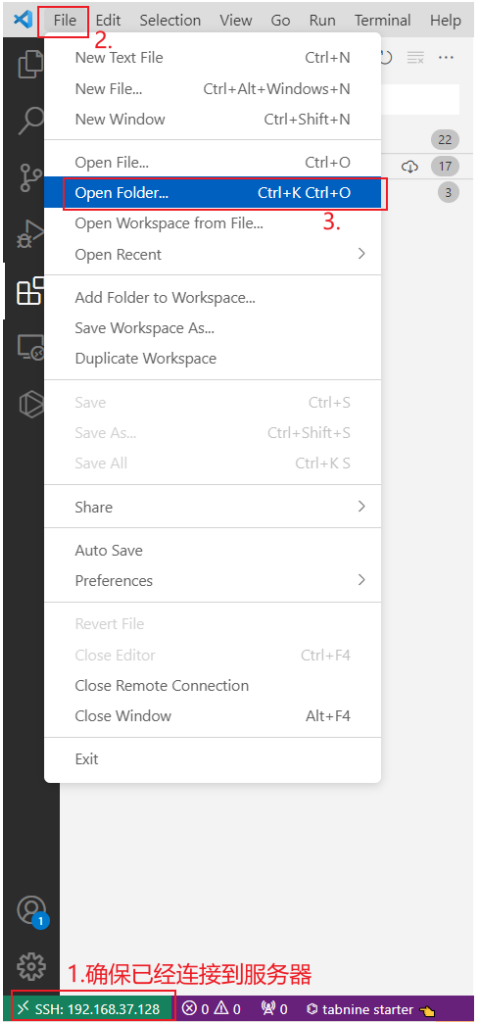
```
输入文件名打开文件: Ctrl + P
跳到某行: Ctrl + G + 行号
打开文件并跳到某行: Ctrl + p 文件名:行号
列出文件里的函数 : Ctrl + Shift + O，可以输入函数名跳转
函数/变量跳转: 按住Ctrl同时使用鼠标左键点击、F12
前进: Ctrl + Shift + -
后退: Ctrl + Alt + -
列出引用 : Shift + F12
查找所有引用 : Alt + Shift + F12
切换侧边栏展示/隐藏: Ctrl + B
打开命令菜单: Ctrl + Shift + P
手动触发建议: Ctrl + Space
手动触发参数提示: Ctrl + Shift + Space
打开/隐藏终端: Ctrl + `(Tab上方的那个键)
重命名符号: F2
当前配置调试: F5
上/下滚编辑器: Ctrl + ↑/↓
搜索/替换 : Ctrl + F/H
高亮文字：shift + alt + z
取消高亮：shift + alt + a
```

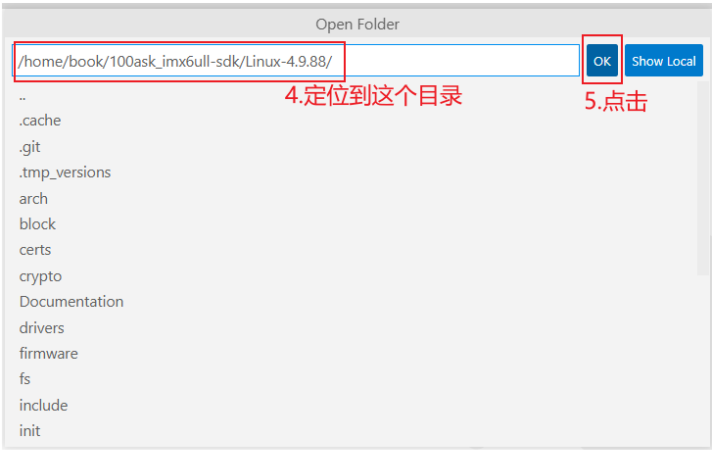
3. 使用vscode阅读内核源码

确保Ubuntu上Linux内核源码目录下已经有了文件compile_commands.json。

3.1 打开目录

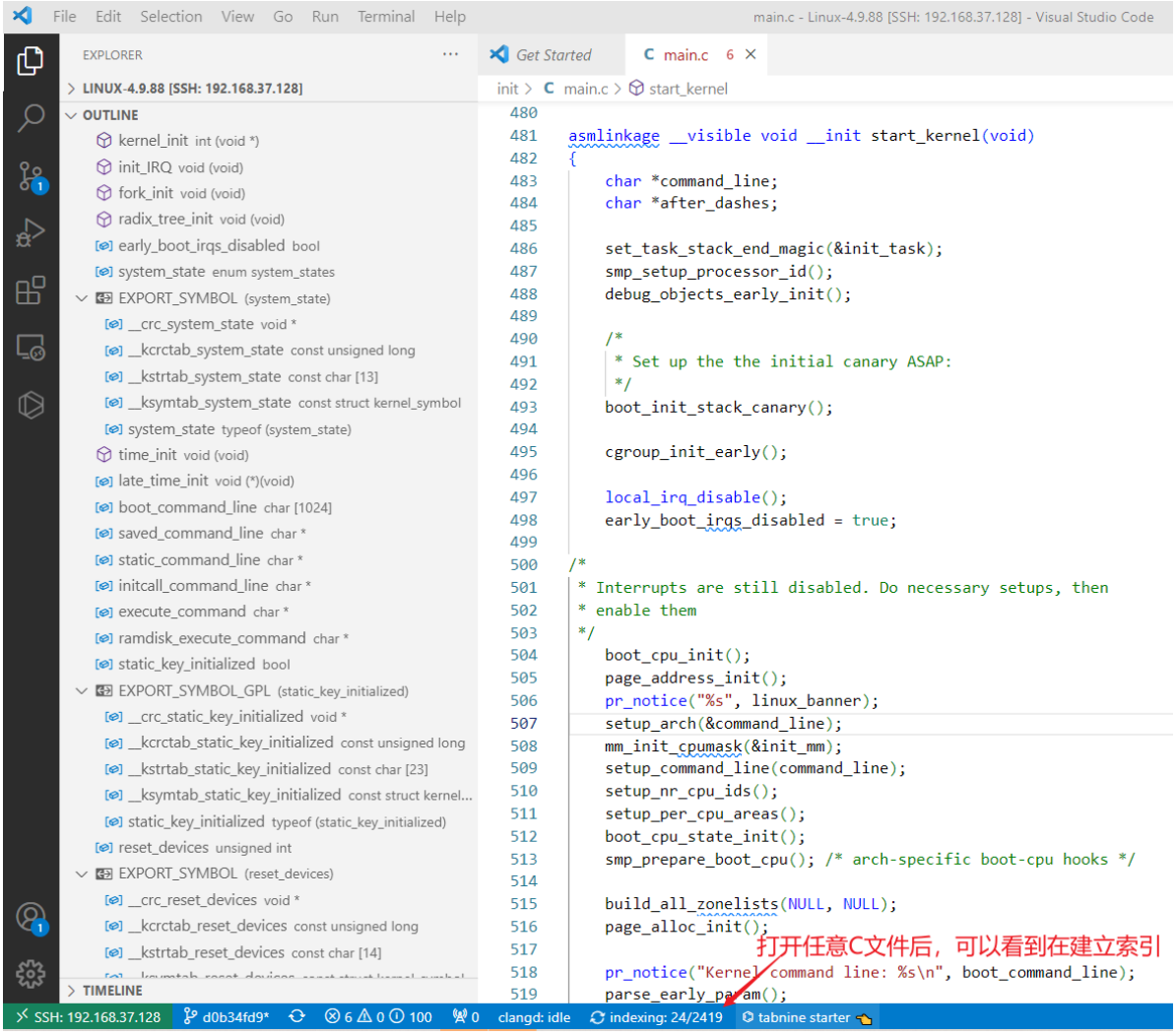
vscode已经连接到Ubuntu后，如下操作：





3.2 触发clangd建立索引

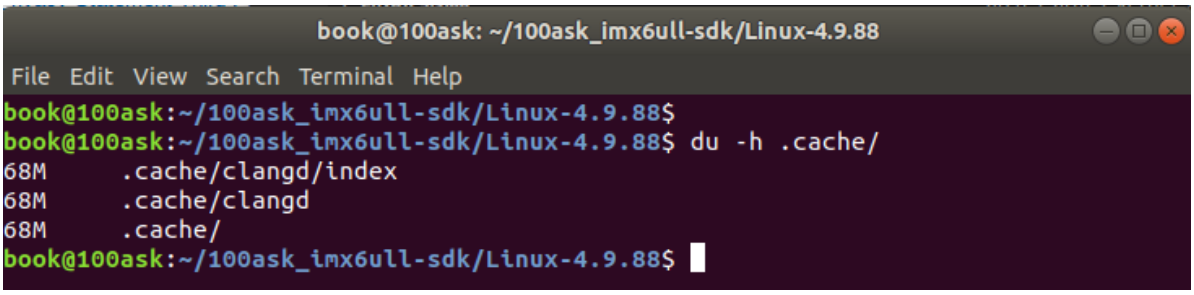
在vscode里打开任意一个C文件，就会触发clangd建立索引：



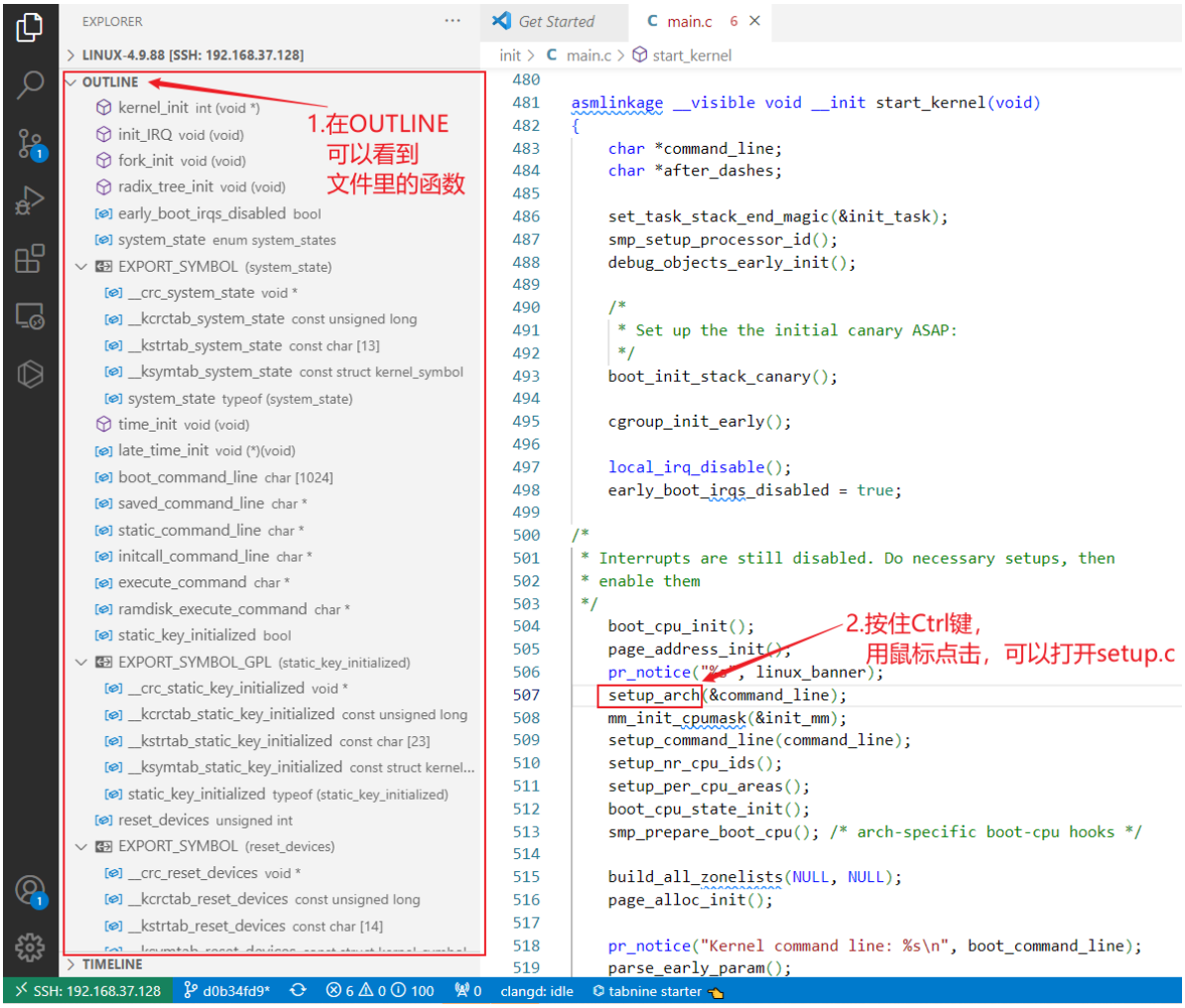
如果没有看到上述状态，可以如下处理：

- 按照《2.5.2 配置clangd》重新编辑setting.json
- 重新启动vscode、重新打开内核源码目录、重新打开C文件

在创建索引的过程中，可以使用如下命令查看.cache目录，它会不断变大(最终大小在60M左右)：



3.3 验证



4. 使用vscode阅读内核外部的源码

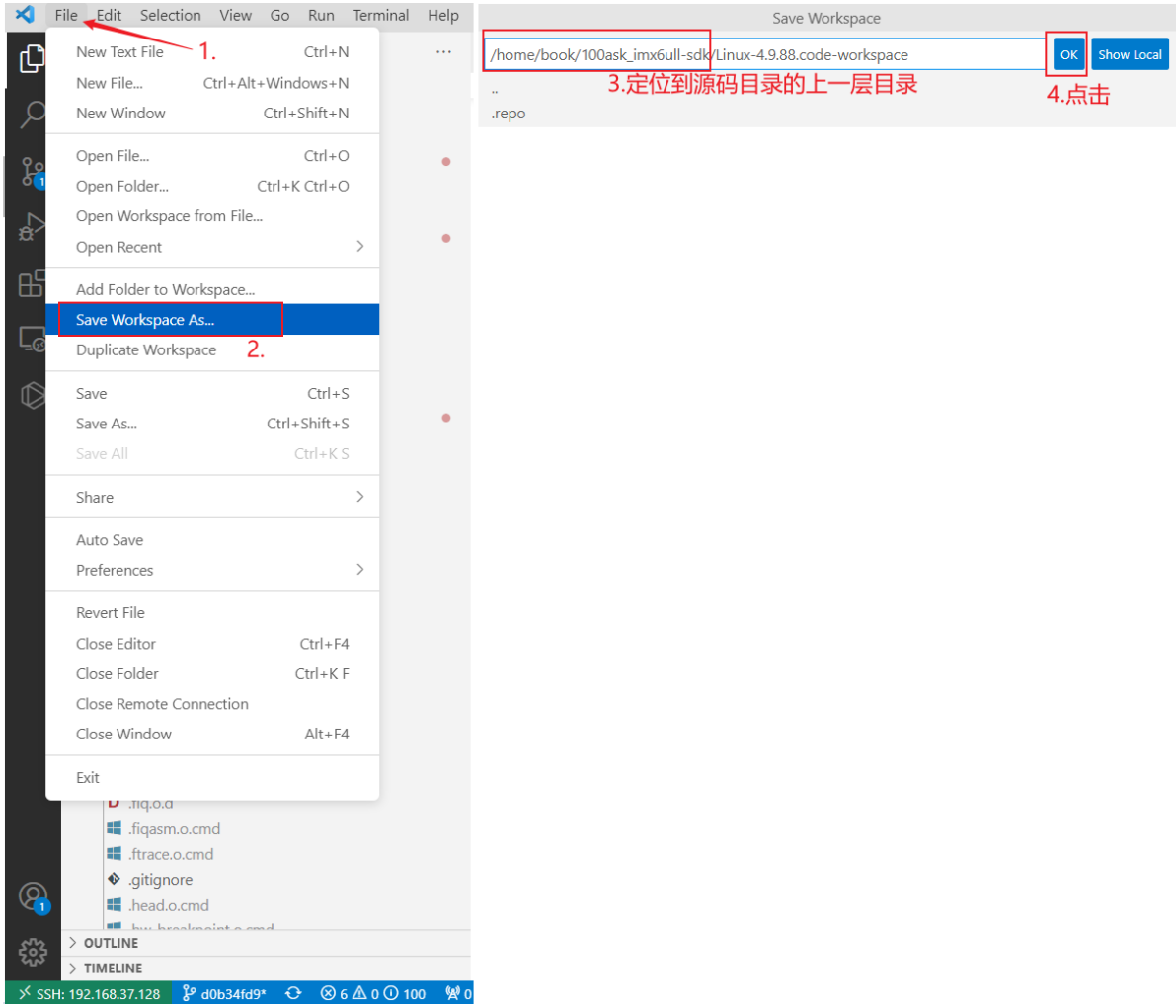
比如我们编写了hello驱动程序，它用到内核里的头文件、函数，我们点击hello驱动里的函数时，想打开内核的文件。

需要创建一个workspace：

- 里面含有内核目录、hello驱动源码目录
- 内核目录下有compile_commands.json
- hello驱动源码目录下有compile_commands.json

4.1 创建workspace

使用vscode打开内核目录，然后保存为WorkSpace，如下操作：



4.2 把驱动目录加入workspace

假设驱动程序位于这个目录：`/home/book/nfs_rootfs/drivers_projects/01_hello_drv/`。

4.2.1 编译驱动

使用如下命令编译，它会生成compile_commands.json：

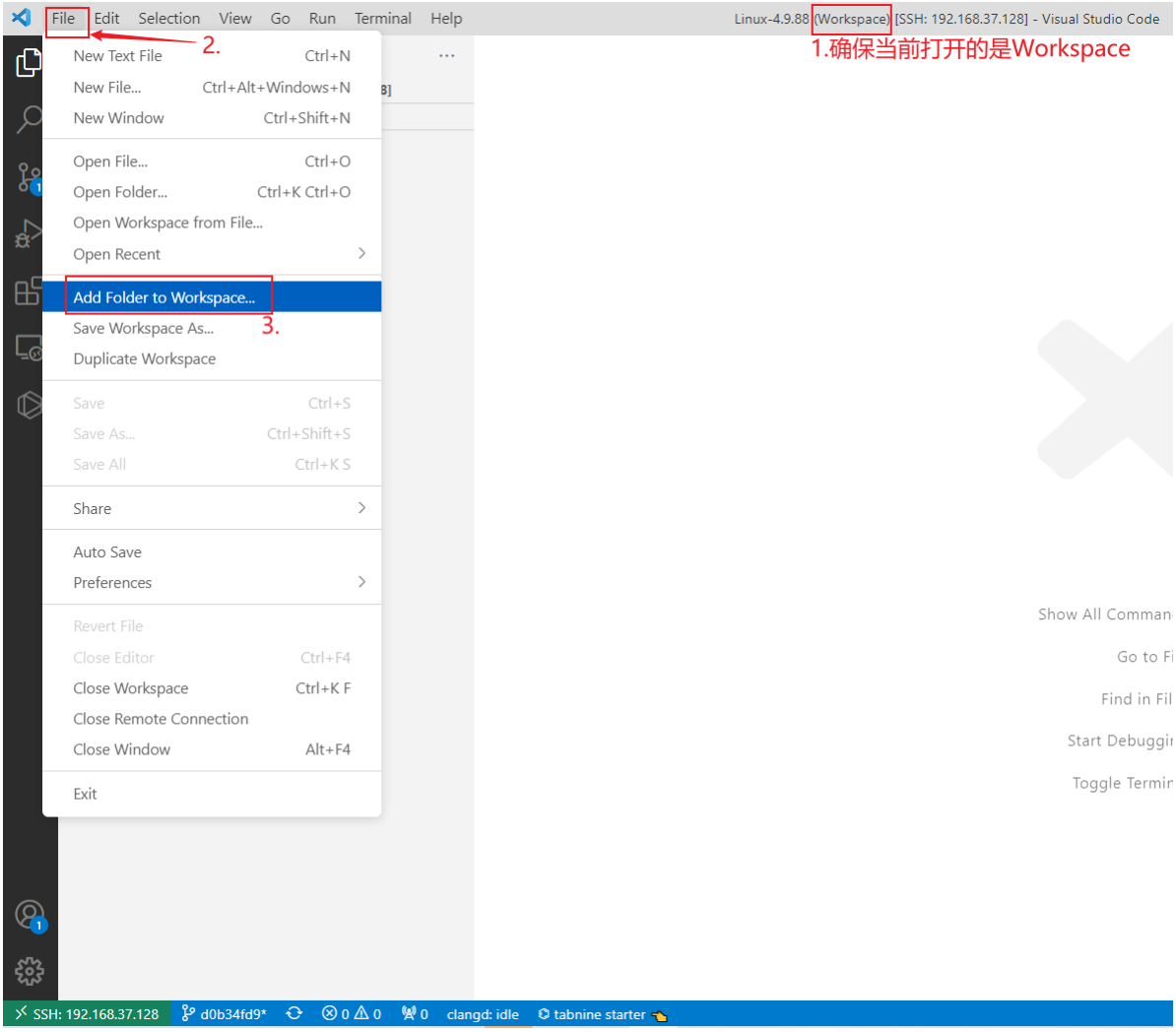
```
cd /home/book/nfs_rootfs/drivers_projects/01_hello_drv/
bear make
```

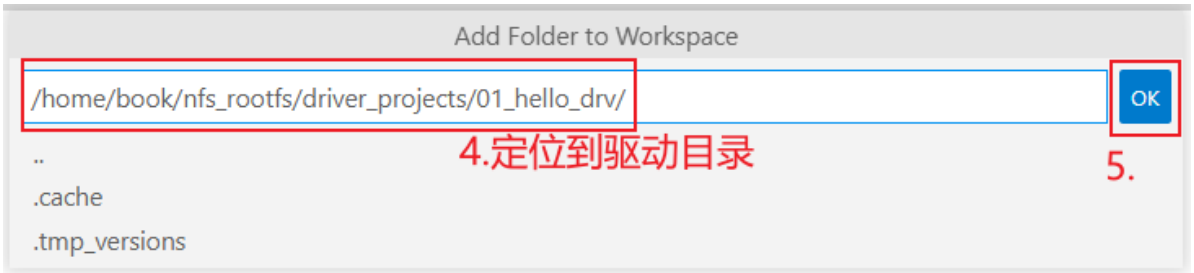
```
book@100ask: ~/nfs_rootfs/drivers_projects/01_hello_drv
File  Edit  View  Search  Terminal  Help
book@100ask:~/nfs_rootfs/drivers_projects/01_hello_drv$ bear make
make -C /home/book/100ask_imx6ull-sdk/Linux-4.9.88 M=`pwd` modules
make[1]: Entering directory '/home/book/100ask_imx6ull-sdk/Linux-4.9.88'
  CC [M]  /home/book/nfs_rootfs/drivers_projects/01_hello_drv/hello_drv.o
  Building modules, stage 2.
  MODPOST 1 modules
  CC      /home/book/nfs_rootfs/drivers_projects/01_hello_drv/hello_drv.mod.o
  LD [M]  /home/book/nfs_rootfs/drivers_projects/01_hello_drv/hello_drv.ko
make[1]: Leaving directory '/home/book/100ask_imx6ull-sdk/Linux-4.9.88'
arm-buildroot-linux-gnueabihf-gcc -o hello_drv_test hello_drv_test.c
book@100ask:~/nfs_rootfs/drivers_projects/01_hello_drv$
```

4.2.2 修改compile_commands.json

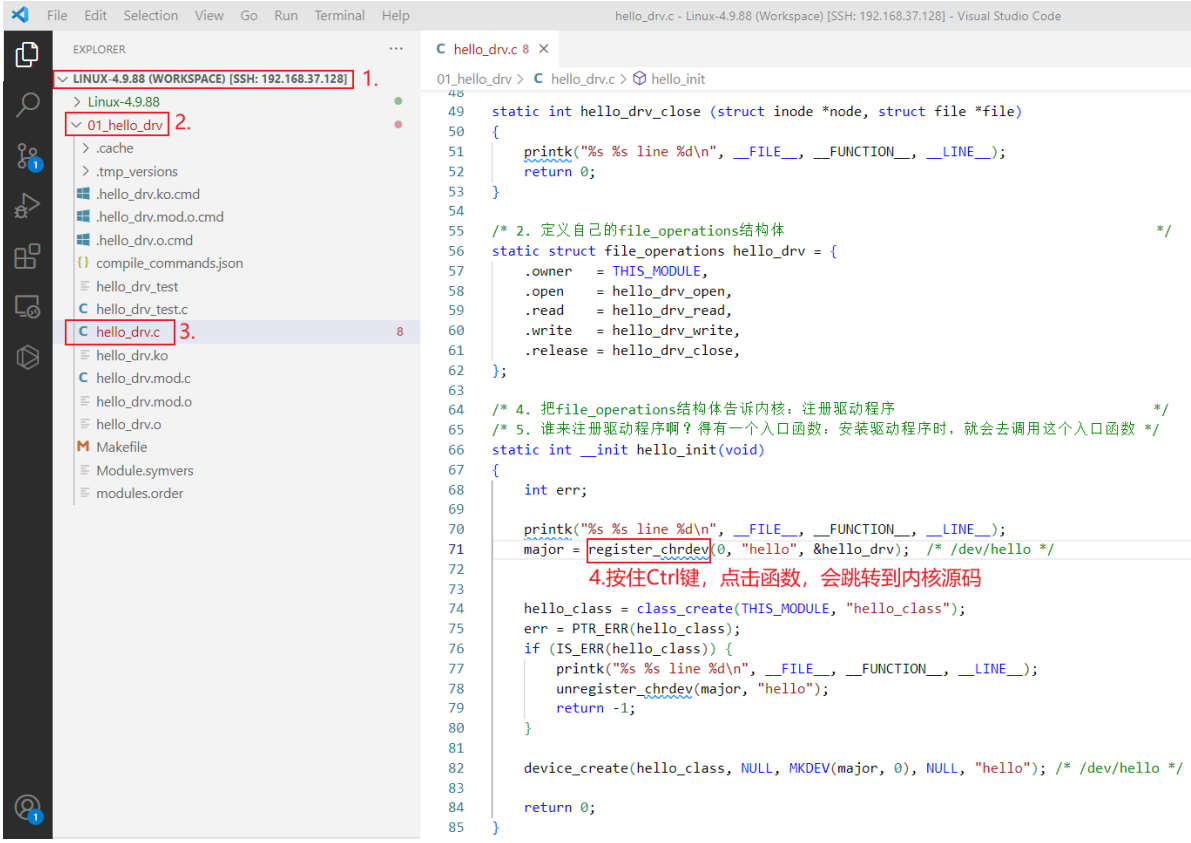
把里面的"cc"全部修改为"arm-buildroot-linux-gnueabihf-gcc"。

4.2.3 加入workspace


File
Edit
Selection
View
Go
Run
Terminal
Help
Linux-4.9.88 (Workspace) [SSH: 192.168.37.128] - Visual Studio Code
1.确保当前打开的是Workspace
2.
New Text File Ctrl+N
New File... Ctrl+Alt+Windows+N
New Window Ctrl+Shift+N
Open File... Ctrl+O
Open Folder... Ctrl+K Ctrl+O
Open Workspace from File...
Open Recent
Add Folder to Workspace...
Save Workspace As...
3.
Duplicate Workspace
Save Ctrl+S
Save As... Ctrl+Shift+S
Save All Ctrl+K S
Share
Auto Save
Preferences
Revert File
Close Editor Ctrl+F4
Close Workspace Ctrl+K F
Close Remote Connection
Close Window Alt+F4
Exit
Show All Comman
Go to Fi
Find in Fil
Start Debuggi
Toggle Termir
SSH: 192.168.37.128
d0b34fd9*
0 0
0
clangd: idle
tabnine starter


Add Folder to Workspace
/home/book/nfs_rootfs/driver_projects/01_hello_drv/
OK
4.定位到驱动目录
5.
..
.cache
.tmp_versions

4.3 验证

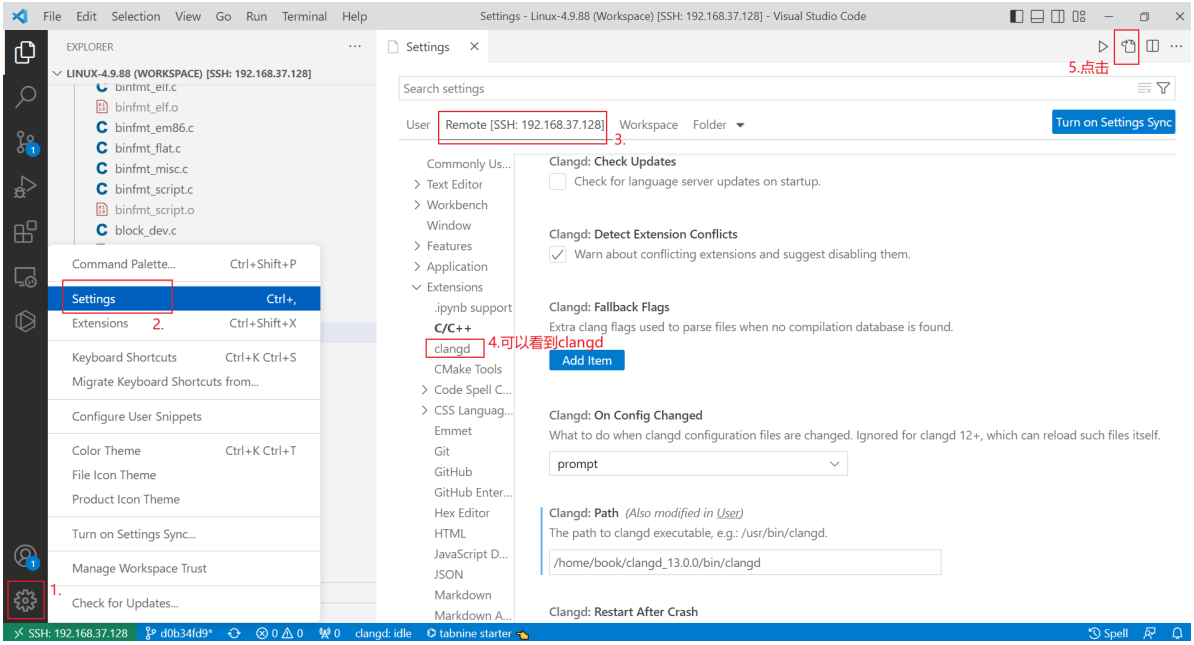


5. 常见错误

5.1 无法跳转

第1步，确认已经关闭intellisense：

在Windows的vscode界面按下图步骤打开setting.json文件:



在配置文件中：

Settings　{} settings.json ×

home > book > .vscode-server > data > Machine > {} settings.json > ...

```
1  {
2      "C_Cpp.default.intelliSenseMode": "linux-gcc-arm",
3      "C_Cpp.intelliSenseEngine": "Disabled",
4      "clangd.path": "/home/book/clangd_13.0.0/bin/clangd",
5      "clangd.arguments": [
6          "--log=verbose",
7      ],
8  }
```

确认首字母大写，
而不是"disabled"

第2步，跟第1步一样打开配置文件后，确认Ubuntu中有clangd：

Settings　{} settings.json ×

home > book > .vscode-server > data > Machine > {} settings.json > ...

```
1  {
2      "C_Cpp.default.intelliSenseMode": "linux-gcc-arm",
3      "C_Cpp.intelliSenseEngine": "Disabled",
4      "clangd.path": "/home/book/clangd_13.0.0/bin/clangd",
5      "clangd.arguments": [
6          "--log=verbose",
7      ],
8  }
```

确认服务器上有这个文件

第3步，确认源码目录下有compile_commands.json，并且文件里面记录有验证用的C文件、"cc"被改成了"arm-buildroot-linux-gnueabihf-gcc"：

File Edit View Search Terminal Help

```
[
    {
        "arguments": [
            "arm-buildroot-linux-gnueabihf-gcc",
            "-c",
            "-Wp,-MD,/home/book/nfs_rootfs/drivers_projects/01_hello_drv/.hello_drv.o.d",
            "-nostdinc",
            "../../nfs_rootfs/drivers_projects/01_hello_drv/hello_drv.c"
        ],
        "directory": "/home/book/100ask_imx6ull-sdk/Linux-4.9.88",
        "file": "../../nfs_rootfs/drivers_projects/01_hello_drv/hello_drv.c"
    },
```

第4步，在vscode里打开C文件后，确认.cache目录生成了：

```
book@100ask:~/100ask_imx6ull-sdk/Linux-4.9.88$
book@100ask:~/100ask_imx6ull-sdk/Linux-4.9.88$ du -h .cache/
68M     .cache/clangd/index
68M     .cache/clangd
68M     .cache/
book@100ask:~/100ask_imx6ull-sdk/Linux-4.9.88$
```

```
book@100ask:~/nfs_rootfs/drivers_projects/01_hello_drv$
book@100ask:~/nfs_rootfs/drivers_projects/01_hello_drv$ du -h .cache/
16K     .cache/clangd/index
20K     .cache/clangd
24K     .cache/
book@100ask:~/nfs_rootfs/drivers_projects/01_hello_drv$
book@100ask:~/nfs_rootfs/drivers_projects/01_hello_drv$
```

5.2 Ubuntu IP变化

Ubuntu中的网卡IP会发生变化，如果发现无法连接服务器后，需要确认IP是否发生了变化，然后按照《2.3.1 连接Ubuntu》重新连接。

如果想那么麻烦，可以设置vmware让NAT的固定下来，如下图操作：


虚拟网络编辑器
名称 类型 外部连接 主机连接 DHCP 子网地址
VMnet0 自定义... - - - 192.168.226.0
VMnet1 自定义... - - - 192.168.174.0
VMnet8 NAT 模式 NAT 模式 已连接 已启用 192.168.37.0
5.
添加网络(E)...
移除网络(O)
重命名网络(W)...
VMnet 信息
桥接模式(将虚拟机直接连接到外部网络)(B)
已桥接至(G):
自动设置(U)...
NAT 模式(与虚拟机共享主机的 IP 地址)(N)
NAT 设置(S)...
仅主机模式(在专用网络内连接虚拟机)(H)
将主机虚拟适配器连接到此网络(V)
主机虚拟适配器名称: VMware 网络适配器 VMnet8
6.
使用本地 DHCP 服务将 IP 地址分配给虚拟机(D)
DHCP 设置(P)...
子网 IP (I): 192 . 168 . 37 . 0
子网掩码(M): 255 . 255 . 255 . 0
还原默认设置(R)
导入(T)...
导出(X)...
确定
取消
应用(A)
帮助


虚拟网络编辑器
DHCP 设置
网络: vmnet8
子网 IP: 192.168.37.0
子网掩码: 255.255.255.0
起始 IP 地址(S): 192 . 168 . 37 . 128
结束 IP 地址(E): 192 . 168 . 37 . 254
广播地址: 192.168.37.255
7.改为最大值63
天:
小时:
分钟:
默认租用时间(D): 63 0 30
最长租用时间(M): 63 2 0
确定
取消
帮助
子网地址
192.168.226.0
192.168.174.0
192.168.37.0
重命名网络(W)...
自动设置(U)...
NAT 设置(S)...
仅主机模式(在专用网络内连接虚拟机)(H)
将主机虚拟适配器连接到此网络(V)
主机虚拟适配器名称: VMware 网络适配器 VMnet8
使用本地 DHCP 服务将 IP 地址分配给虚拟机(D)
DHCP 设置(P)...
子网 IP (I): 192 . 168 . 37 . 0
子网掩码(M): 255 . 255 . 255 . 0
还原默认设置(R)
导入(T)...
导出(X)...
确定
取消
应用(A)
帮助