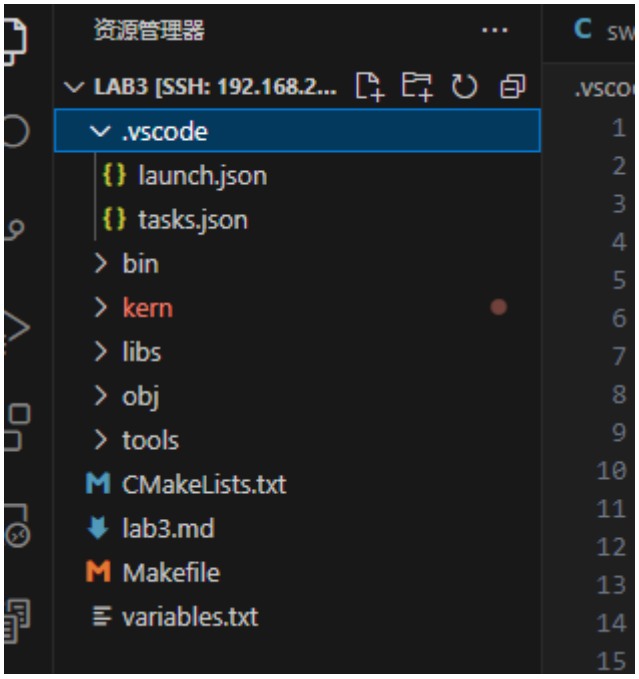
vscode可视化调试内核

前期准备

使用`vscode`远程连接虚拟机，打开任意一个实验的文件夹（如`lab3`），在该目录下新建文件夹`.vscode`，在该文件夹中新建两个文件`launch.json`，`tasks.json`，建立完毕之后的文件目录如下：



打开`tasks.json`文件，在其中输入

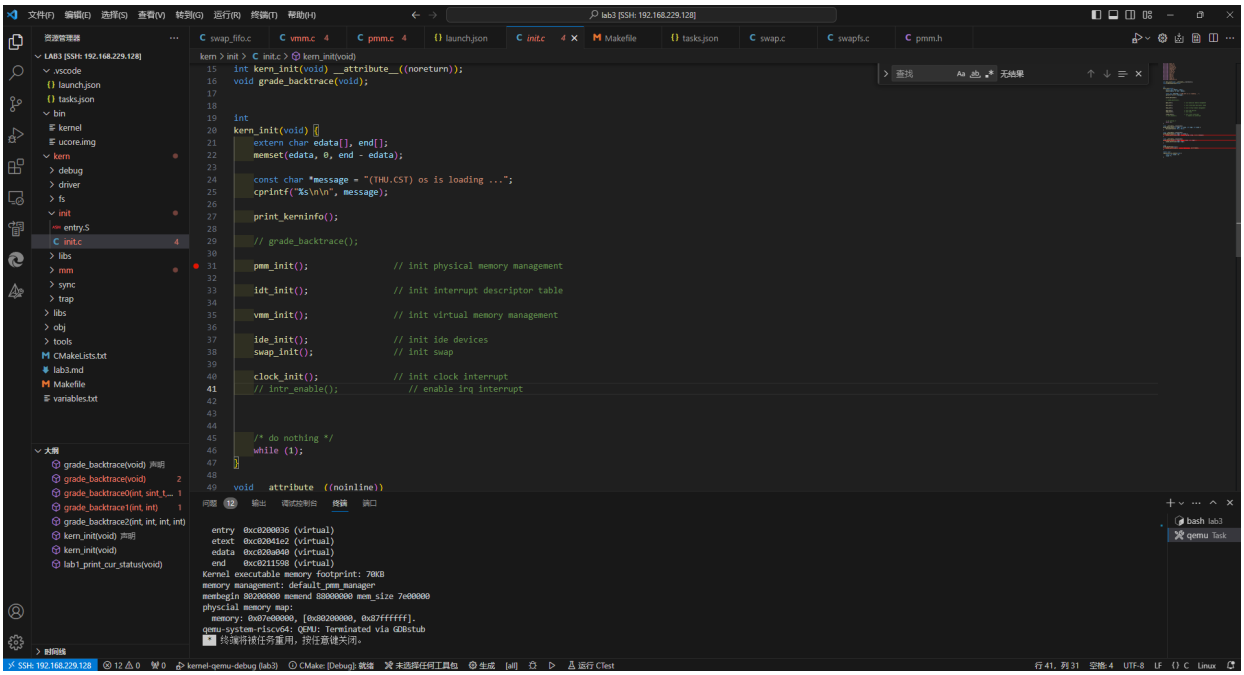
```
{
    // See https://go.microsoft.com/fwlink/?LinkId=733558
    // for the documentation about the tasks.json format
    "version": "2.0.0",
    "tasks": [
        {
            "label": "qemu",
            "type": "shell",
            "command": "make",
            "args": ["debug"],
            "isBackground": true,
            "problemMatcher": []
        }
    ]
}
```

打开`launch.json`文件，在其中输入

```
{
    // 使用 IntelliSense 了解相关属性。
    // 悬停以查看现有属性的描述。
    // 欲了解更多信息，请访问: https://go.microsoft.com/fwlink/?linkid=830387
    "version": "0.2.0",
    "configurations": [
        {
            "name": "kernel-qemu-debug",
            "type": "cppdbg",
            "request": "launch",
            "miDebuggerServerAddress": "127.0.0.1:1234",
            "miDebuggerPath": "riscv64-unknown-elf-gdb",
            "program": "${workspaceFolder}/bin/kernel",
            "args": ["-ex", "'set arch riscv:rv64'"],
            "preLaunchTask": "qemu",
            "stopAtEntry": true,
            "cwd": "${workspaceFolder}",
            "environment": [],
            "externalConsole": false,
            "logging": {
                "engineLogging": false
            },
            "MIMode": "gdb"
        }
    ]
}
```

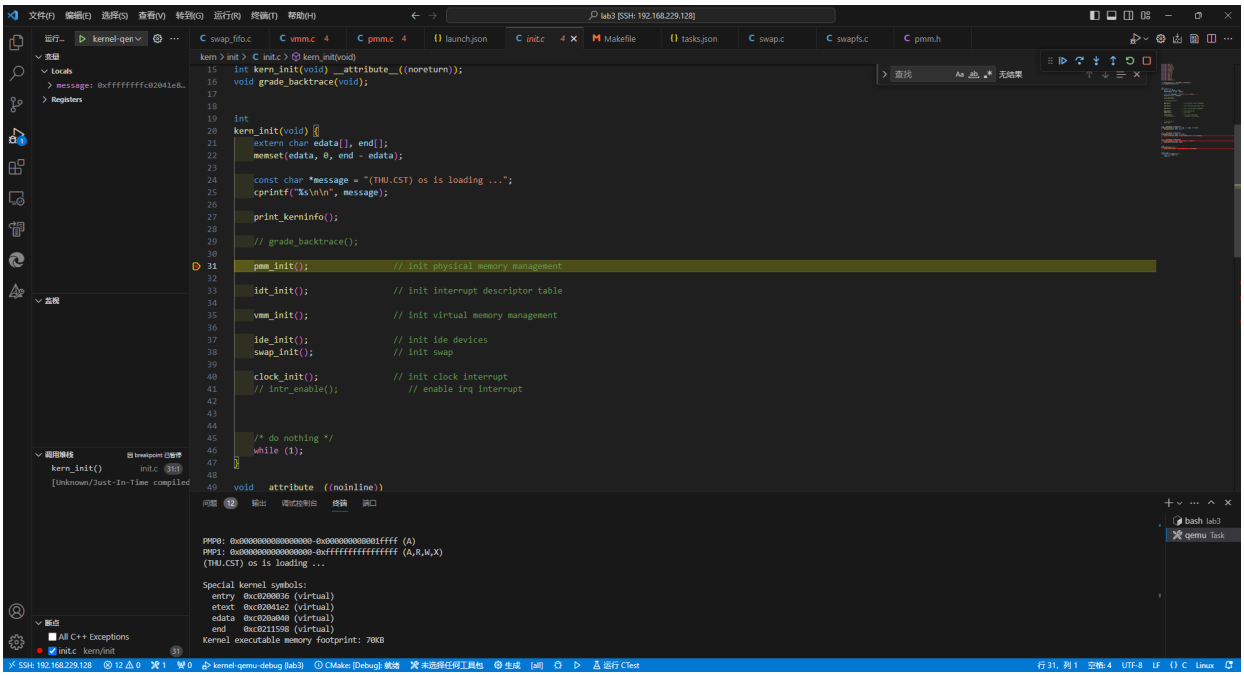
开始调试

打开文件`kern/init/init.c`，在其中打断点



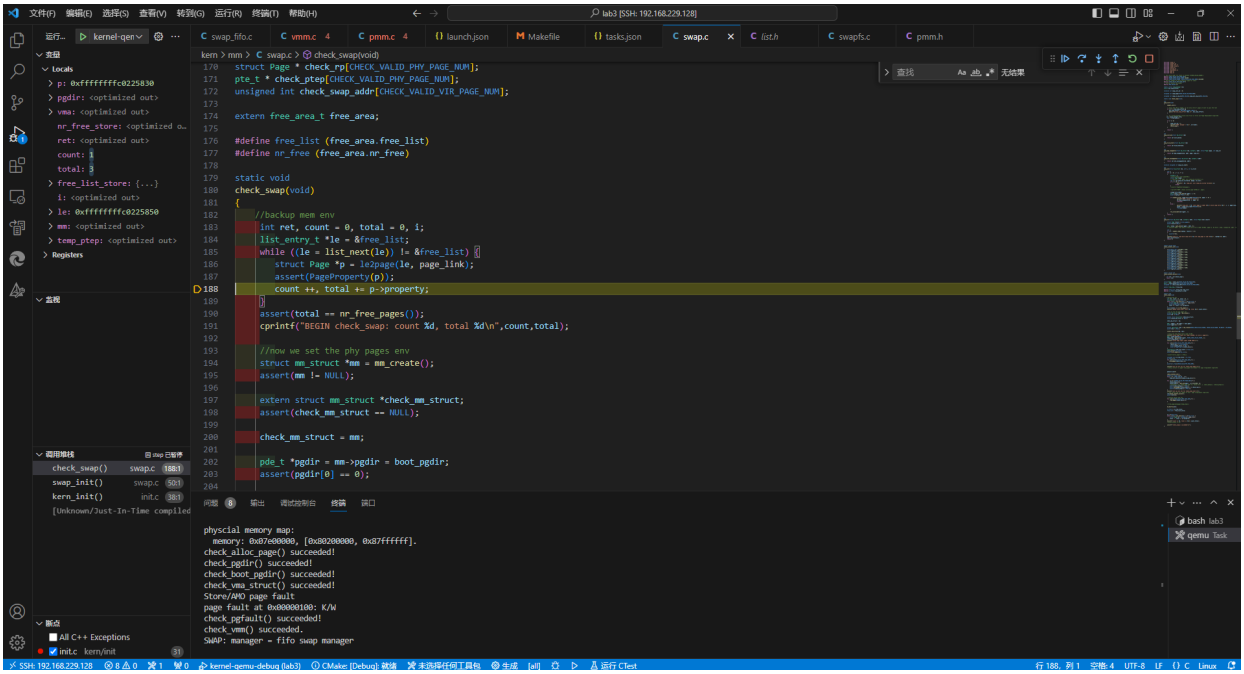
按F5，需要等待一会儿，然后会出现如下提示

这里勾选记住选择，点击仍要调试，之后就可以成功调试了



调试方式与`vscode`调试`C++`或`python`（或其他的语言）类似，F10单步执行，F11转到函数内部单步执行。

下方选择终端页面，可以看到每一个函数的输出，左侧也可以看到变量。



总结

占据空白页