**Skynet 远程调试笔记**

## 一．准备工作

1. Win10 下Vscode 安装，及相关插件安装

1） Remote-WSL

2） Remote-SSH

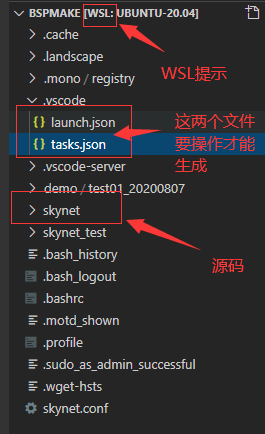
3) Remote Development

1. WSL 安装，及相关插件安装 （ubuntu 20.04）
2. sudo apt install git build-essential libreadline-dev autoconf
3. sudo apt install gdb
4. code . (看是否能从linux中打vscode,如果不能那就从windows中打开)
5. git clone <https://github.com/cloudwu/skynet.git>

cd skynet

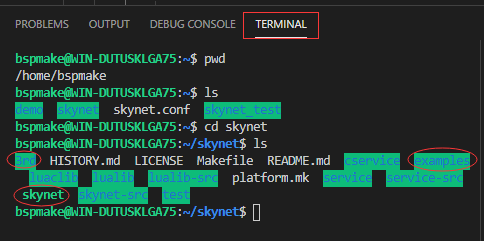
make linux

1. 效果：



说明：提示WSL 说明环境已经好了。打开终端就可以开始编译了。

1. 进入您下载的目录 make linux 就可以了生成skynet.
2. 菜单－〉 view ->Terminal



## 如何配置调试环境：

1. 添加编译任务（tasks.json），生成可执行文件。即类似于VS系统的建立工程，多个任务就相当于多个工程了。
2. 主是要生成tasks.json 文件 如下：

{

    "version": "2.0.0",

    "tasks": [

        {

            "type": "shell",

            "label": "build skynet",

            "command": "/usr/bin/make",

            "args": [

                "linux"

            ],

            "options": {

                "cwd": "${workspaceFolder}/skynet"

            },

            "problemMatcher": [

                "$gcc"

            ],

            "group": "build"

        }

    ]

}

1. 添加可调试环境配置：（launch.json）

{

    "version": "0.2.0",                 //launch.json文件的语言版本

    "configurations": [

               {

            "name": "Launch: skynet",

            "type": "cppdbg",

            "request": "launch",

            "program": "${workspaceFolder}/skynet/skynet",

            "args": ["${workspaceFolder}/skynet.conf"],

            "stopAtEntry": false,

            "cwd": "${workspaceFolder}",

            "environment": [],

            "externalConsole": false,

            "MIMode": "gdb",

            "miDebuggerPath": "/usr/bin/gdb",

            "setupCommands": [

                {

                    "description": "Enable pretty-printing for gdb",

                    "text": "-enable-pretty-printing",

                    "ignoreFailures": true

                }

            ]

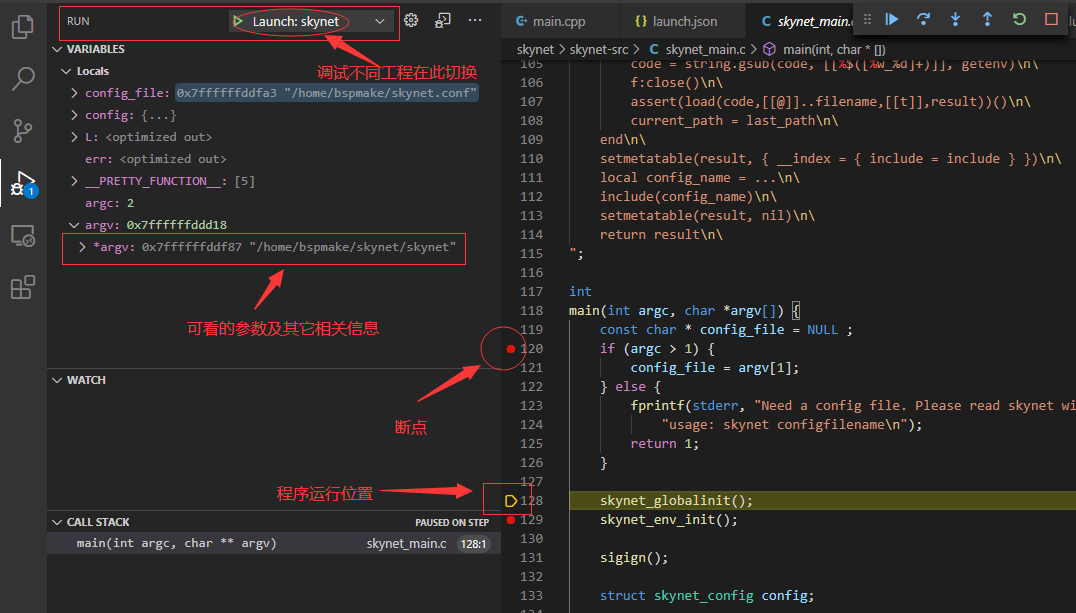
        }

    ]

}

说明：如果您的系统中没有上面的两个文件，刚开始也是一定没有。那么通过如下操作添加文件。 1） 添加tasks.json 菜单 －〉terminal -> configure tasks

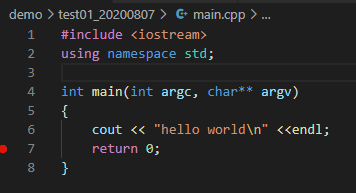
1. 添加launch.json 菜单 －〉Run -> add configuration
2. 复制上面相应的配置即可。
3. 运行效果：



## 感谢MARK老师：

感谢Mark老师指导并远程帮我完成第二步生成配置文件的步骤。作为新人即使知道步骤，也会错在细节上。可能是因为VS 系统IDE 的影响，喜欢那种调试风格，GDB感觉是大神的专属。好了，配置完成，举一反三的时候到了。

## 举一反三 C++ Hello world 配置：



{

    "version": "2.0.0",

    "tasks": [

        {

            "type": "shell",

            "label": "build skynet",

            "command": "/usr/bin/make",

            "args": [

                "linux"

            ],

            "options": {

                "cwd": "${workspaceFolder}/skynet"

            },

            "problemMatcher": [

                "$gcc"

            ],

            "group": "build"

        },

        {

            "type": "shell",

            "label": "build: c++ hello world",

            "command": "/usr/bin/g++",

            "args": [

                "-g",

                "${file}",

                "-o",

                "${fileDirname}/${fileBasenameNoExtension}"

            ],

            "options": {

                "cwd": "${workspaceFolder}"

            },

            "problemMatcher": [

                "$gcc"

            ],

            "group": "build"

        }

    ]

}

{

    "version": "0.2.0",                 //launch.json文件的语言版本

    "configurations": [

        {

            "name": "Launch: hello world",

            "type": "cppdbg",

            "request": "launch",

            "program": "${workspaceFolder}/demo/test01\_20200807/main",

            "args": [],

            "stopAtEntry": false,

            "cwd": "${workspaceFolder}",

            "environment": [],

            "externalConsole": false,

            "MIMode": "gdb",

            "setupCommands": [

                {

                    "description": "Enable pretty-printing for gdb",

                    "text": "-enable-pretty-printing",

                    "ignoreFailures": true

                }

            ]

        },

        {

            "name": "Launch: skynet",

            "type": "cppdbg",

            "request": "launch",

            "program": "${workspaceFolder}/skynet/skynet",

            "args": ["${workspaceFolder}/skynet.conf"],

            "stopAtEntry": false,

            "cwd": "${workspaceFolder}",

            "environment": [],

            "externalConsole": false,

            "MIMode": "gdb",

            "miDebuggerPath": "/usr/bin/gdb",

            "setupCommands": [

                {

                    "description": "Enable pretty-printing for gdb",

                    "text": "-enable-pretty-printing",

                    "ignoreFailures": true

                }

            ]

        }

    ]

}

