c++编译：clang++

系统默认编译文件：/usr/bin/clang++
自己安装了homebrew，且环境配置也用了homebrew，环境主要是基于cern root和geant4配置的："/opt/homebrew/opt/llvm/bin/clang++"

// 看目前环境编译器是系统还是homebrew
which clang++
clang++ --version

终端编译

> 用系统自带的话，首先终端临时环境改一下，要不然报错找不到头文件

- unset CC
- unset CXX
- unset CPPFLAGS
- unset LDFLAGS
- export PATH="/usr/bin:/bin:/usr/sbin:/sbin:/usr/local/bin:$PATH"
- which clang++
- clang++ --version

– /usr/bin/clang++
– Apple clang version 17.0.0 (clang-1700.0.13.3)
– Target: arm64-apple-darwin24.4.0
– Thread model: posix
– InstalledDir: /Library/Developer/CommandLineTools/usr/bin

- clang++ -std=c++17 -g main.cpp -o main
- ./main
- rm main // 删除可执行文件 或者运行 clang++ -std=c++17 -g main.cpp 时 删除临时可执行文件 rm a.out

> 用homebrew，即和cernroot和geant4环境一致

- which clang++
- clang++ --version

– /opt/homebrew/opt/llvm/bin/clang++
– Homebrew clang version 20.1.2

  - Target: arm64-apple-darwin24.4.0
  - Thread model: posix
  - InstalledDir: /opt/homebrew/Cellar/llvm/20.1.2/bin
  - Configuration file: /opt/homebrew/etc/clang/arm64-apple-darwin24.cfg
- clang++ -std=c++17 -g main.cpp -o main
- ./main

VS Code 编译

- cpp-dual-compiler
  - Tasks.json

```
{
  "version": "2.0.0",
  "tasks": [
    {
      "label": "clang++ build (system)",
      "type": "shell",
      "command": "/usr/bin/clang++", //主要变路径
      "args": [
        "-std=c++17",
        "-g",
        "main.cpp",
        "-o",
        "main"
      ],
      "group": {
        "kind": "build",
        "isDefault": true
      },
      "problemMatcher": [
        "$gcc"
      ]
    },
    {
      "label": "clang++ build (homebrew)",
      "type": "shell",
      "command": "/opt/homebrew/opt/llvm/bin/clang++", //主要变路径
      "args": [
        "-std=c++17",
        "-g",
        "main.cpp",
        "-o",
        "main"
      ],
      "group": "build",
      "problemMatcher": [
        "$gcc"
      ]
    }
  ]
```

— launch.json

```
{
"version": "0.2.0",
  "configurations": [
    {
      "name": "Run (system clang++)",
      "type": "cppdbg",
      "request": "launch",
      "program": "${workspaceFolder}/main",
      "args": [],
      "stopAtEntry": false,
      "cwd": "${workspaceFolder}",
      "externalConsole": false,
      "MIMode": "lldb",
      "preLaunchTask": "clang++ build (system)"
    },
    {
      "name": "Run (homebrew clang++)",
      "type": "cppdbg",
      "request": "launch",
      "program": "${workspaceFolder}/main",
      "args": [],
      "stopAtEntry": false,
      "cwd": "${workspaceFolder}",
      "externalConsole": false,
      "MIMode": "lldb",
      "preLaunchTask": "clang++ build (homebrew)"
    }
  ]
}
```

- 步骤

— **构建：**

— Cmd + Shift + P → Tasks: Run Task → 选择 clang++ build (system) 或 clang++ build (homebrew)

— **调试：**

— 点击左侧调试图标 ▶️ → 下拉选择你要用的版本 → F5 运行

```
.DSYM文件
```

- 源代码和编译产物的映射，源码指南针，调试文件
- rm -rf main.dSYM //删除
- du -sh main.dSYM // 看大小
- 避免生成

— 不加-g参数

— clang++ -g0 main.cpp -o main // 或者显示关闭调试信息