本页不打印

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 开发/优化者 | 时间 | 审核人 | 开发类型（新开发/优化） | 更新说明 |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| 课程编码 | 适用产品 | 产品版本 | 课程版本ISSUE |
|  |  |  |  |

修订记录

C++编程实践环境搭建说明书



华为技术有限公司

目录

[1 编程实践环境要求 2](#_Toc177115279)

[1.1 编程实践环境要求 2](#_Toc177115280)

[1.1.1 开发环境要求 2](#_Toc177115281)

[1.1.2 开发工具列表 2](#_Toc177115282)

[2 编程环境搭建-windows 3](#_Toc177115283)

[2.1 g++安装 3](#_Toc177115284)

[2.1.1 mingw64下载 3](#_Toc177115285)

[2.1.2 安装 3](#_Toc177115286)

[2.1.3 安装验证 3](#_Toc177115287)

[2.2 cmake安装 4](#_Toc177115288)

[2.2.1 下载&安装 4](#_Toc177115289)

[2.2.2 验证 4](#_Toc177115290)

[2.3 git安装 4](#_Toc177115291)

[2.3.1 下载 4](#_Toc177115292)

[2.3.2 安装 4](#_Toc177115293)

[2.3.3 git配置检查与用户设置 5](#_Toc177115294)

[2.3.4 安装TortoiseGit[可选] 6](#_Toc177115295)

[2.4 Visual Studio Code安装 6](#_Toc177115296)

[2.4.1 下载与安装 6](#_Toc177115297)

[2.4.2 推荐插件列表 7](#_Toc177115298)

[2.4.3 推荐配置 7](#_Toc177115299)

[3 编程环境搭建-linux 10](#_Toc177115300)

[3.1 g++安装 10](#_Toc177115301)

[3.2 cmake安装 10](#_Toc177115302)

[3.3 git安装 11](#_Toc177115303)

[3.4 Visual Studio Code安装 11](#_Toc177115304)

[4 工程环境验证 12](#_Toc177115305)

[5 编程工具快速入门扩展学习资料 13](#_Toc177115306)

# 编程实践环境要求

## 编程实践环境要求

### 开发环境要求

操作系统：64位操作系统，以满足开发工具的64位环境要求。

### 开发工具列表

g++：支持C++17标准

cmake

git

visual studio code

# 编程环境搭建-windows

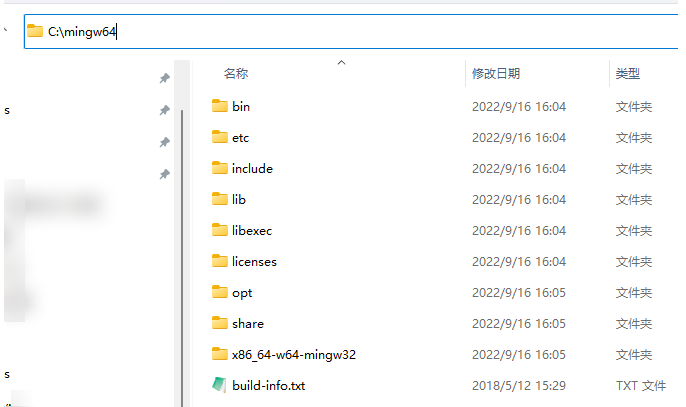
## g++安装

### mingw64下载

[点击下载](https://sourceforge.net/projects/mingw-w64/files/Toolchains%20targetting%20Win64/Personal%20Builds/mingw-builds/8.1.0/threads-posix/seh/x86_64-8.1.0-release-posix-seh-rt_v6-rev0.7z)

### 安装

下载成功后，解压x86\_64-8.1.0-release-posix-seh-rt\_v6-rev0.7z.rar 到c:\mingw64



### 安装验证

在命令行终端执行：

g++ --version

验证g++是否配置成功：

g++ (x86\_64-posix-seh-rev0, Built by MinGW-W64 project) 8.1.0

Copyright (C) 2018 Free Software Foundation, Inc.

This is free software; see the source for copying conditions. There is NO

warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

## cmake安装

### 下载&安装

下载地址：https://cmake.org/download/ 下载64位版本。

•如果下载的是cmake-3.28.0-rc4-windows-x86\_64.msi，双击安装。

•如果下载的是 cmake-3.28.0-rc4-windows-x86\_64.zip，解压到任意目录（例如C:\Program Files\cmake）即可。然后将CMake安装目录下的bin目录加入PATH环境变量。

### 验证

在命令行终端执行：

cmake --version

验证cmake是否配置成功：

cmake version 3.28.0-rc4

CMake suite maintained and supported by Kitware (kitware.com/cmake)

## git安装

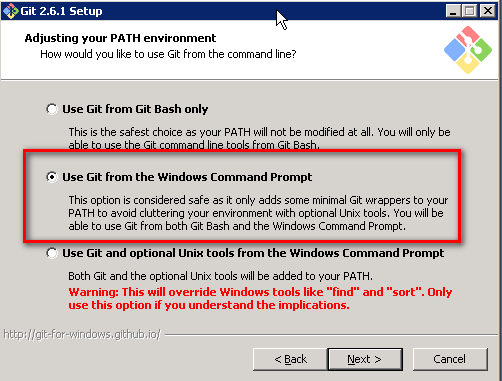
### 下载

官网： [https://git-scm.com/download](https://link.jianshu.com/?t=https%3A%2F%2Fgit-scm.com%2Fdownload)

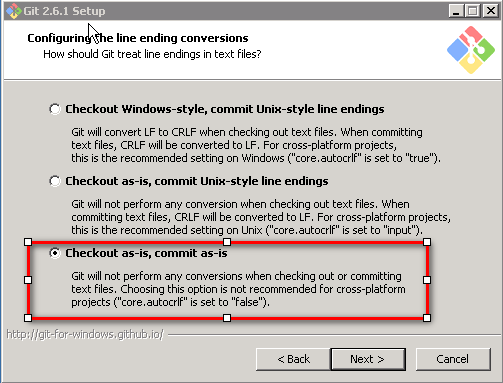
### 安装

安装过程有两步需要注意，其它步骤按默认安装即可。

修改系统的环境变量



配置行尾的结束符



### git配置检查与用户设置

在WINDOWS资源管理器任何一个目录下点击鼠标右键，选择Git Bash Here打开git命令行工具(后面的命令都是在Git Bash环境下执行)：



**检查用户名和邮箱是否配置**

$ git config --global --list

**如未配置，则执行以下命令进行配置：**

$ git config –global user.name "这里换上你的用户名"

$ git config --global user.email "这里换上你的邮箱"

### 安装TortoiseGit[可选]

官网：<https://tortoisegit.org/>

**按默认安装即可**

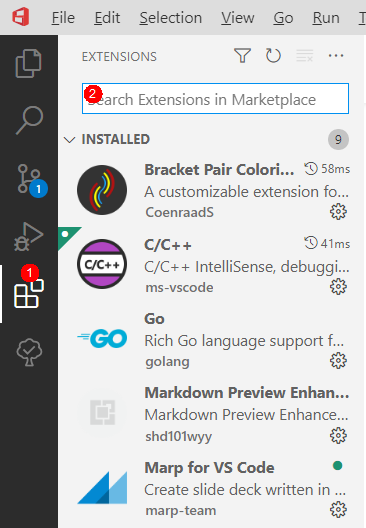
## Visual Studio Code安装

### 下载与安装

官网：<https://code.visualstudio.com/download>

默认步骤安装即可。

### 推荐插件列表



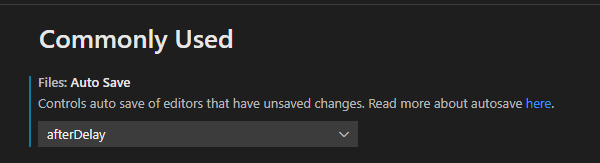
点开Extensions，在搜索栏搜索对应插件名称，点击Install即可。

* Bracket Pair Colorizer
* Todo Tree
* Visual Studio IntelliCode
* C/C++ for Visual Studio Code

### 推荐配置

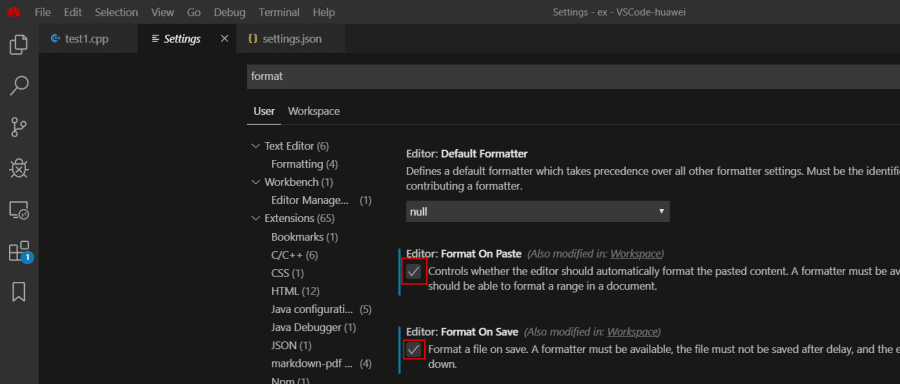
#### Auto Save

先打开首选项设置（ctrl + ,），搜索“Auto Save”。

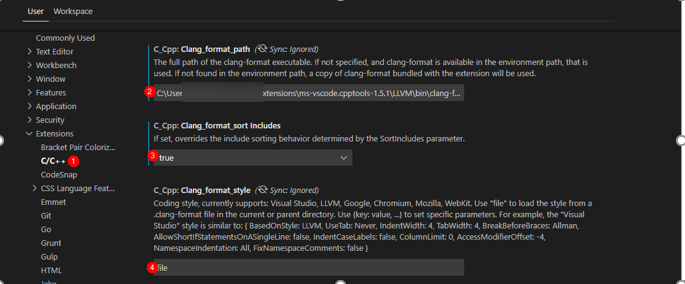


#### Auto format

先打开首选项设置（ctrl + ,），搜索“format”，可勾选“Editor: Format On Paste”，“Editor: Format On Save”，以在粘贴和保存文件时自动格式化。



#### Clang\_format\_style



**注：Clang\_format\_path中，填入**C/C++ for Visual Studio Code中，clang-format工具的安装路径，一般在C盘个人目录下：

C:\Users\{你的用户名} \.vscode\extensions\ms-vscode.cpptools-{版本号}\LLVM\bin\ clang-format.exe

# 编程环境搭建-linux

以ubuntu为例(建议使用ubuntu 20.04之后的版本)

## g++安装

通过apt安装g++

# 打开终端（Ctrl + Alt + T）

# 运行软件包列表更新命令

sudo apt update

# 运行g++安装命令

sudo apt install g++

# 安装完成后，运行版本查询命令检查g++版本，确保g++版本号>8.0

g++ --version

# C++17支持验证，如下代码保存为test.cpp，运行g++ -std=c++17 -o test test.cpp，如能编译通过，表明g++支持C++17

#include <string\_view>

#include <iostream>

int main(void)

{

auto func = [](std::string\_view str)

{

std::cout << str << std::endl;

};

func("hello");

return 0;

}

## cmake安装

通过apt安装cmake

# 打开终端（Ctrl + Alt + T）

# 运行软件包列表更新命令

sudo apt update

# 运行cmake安装命令

sudo apt install cmake

# 安装完成后，运行版本查询命令检查cmake版本

cmake --version

## git安装

通过apt安装git

# 打开终端（Ctrl + Alt + T）

# 运行软件包列表更新命令

sudo apt update

# 运行git安装命令

sudo apt install git

# 验证git安装

git –version

# 配置用户信息

git config –global user.name "这里换上你的用户名"

git config --global user.email "这里换上你的邮箱"

## Visual Studio Code安装

详见<https://code.visualstudio.com/docs/setup/linux>

# 工程环境验证



保存cpp-training-start.zip到本地，直接解压后进入到cpp-training目录中

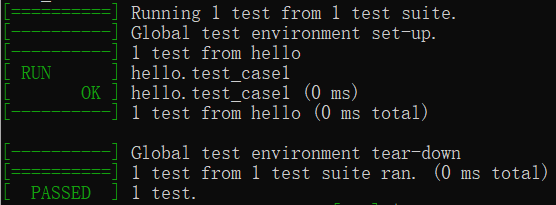
环境验证  
# windows下执行

script\build\_and\_run.bat

# linux下执行

bash script/build\_and\_run.sh

# 出现如下信息，说明环境满足实践开发要求



# 编程工具快速入门扩展学习资料

Quickstart: Building with CMake:

<https://google.github.io/googletest/quickstart-cmake.html>

GoogleTest FAQ:

<https://google.github.io/googletest/faq.html>