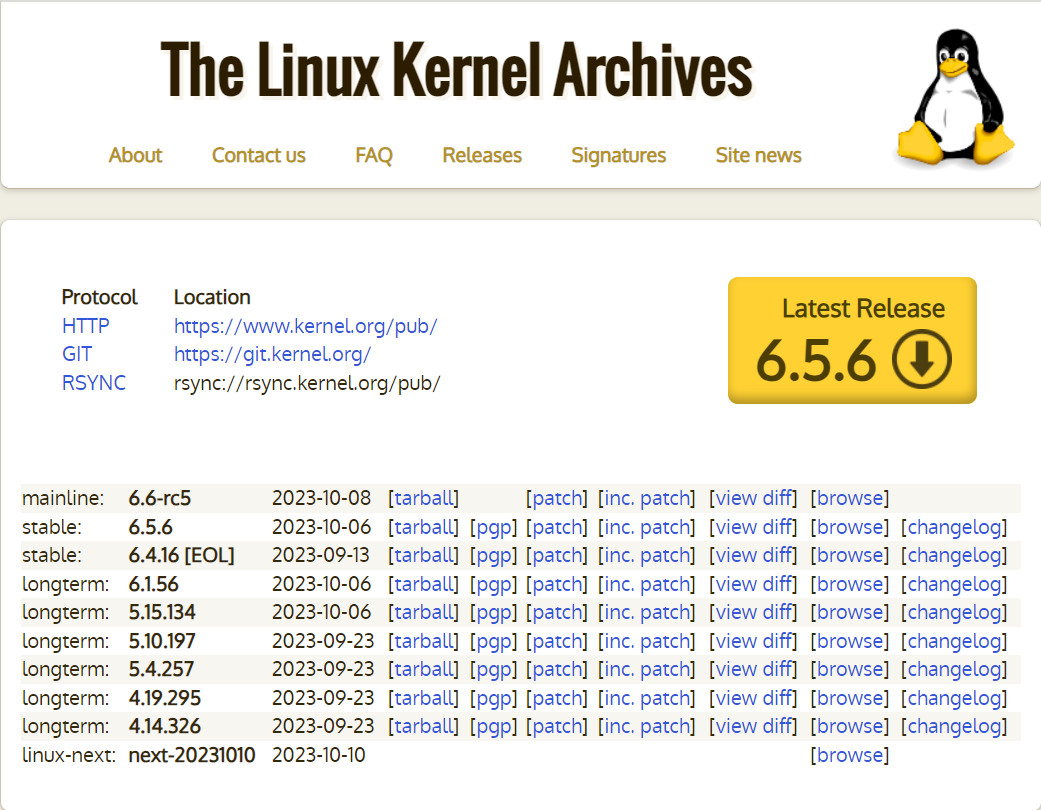
# **Lab02 李鹏2113850**

# **Write a c/c++ program to implement copy one directory**

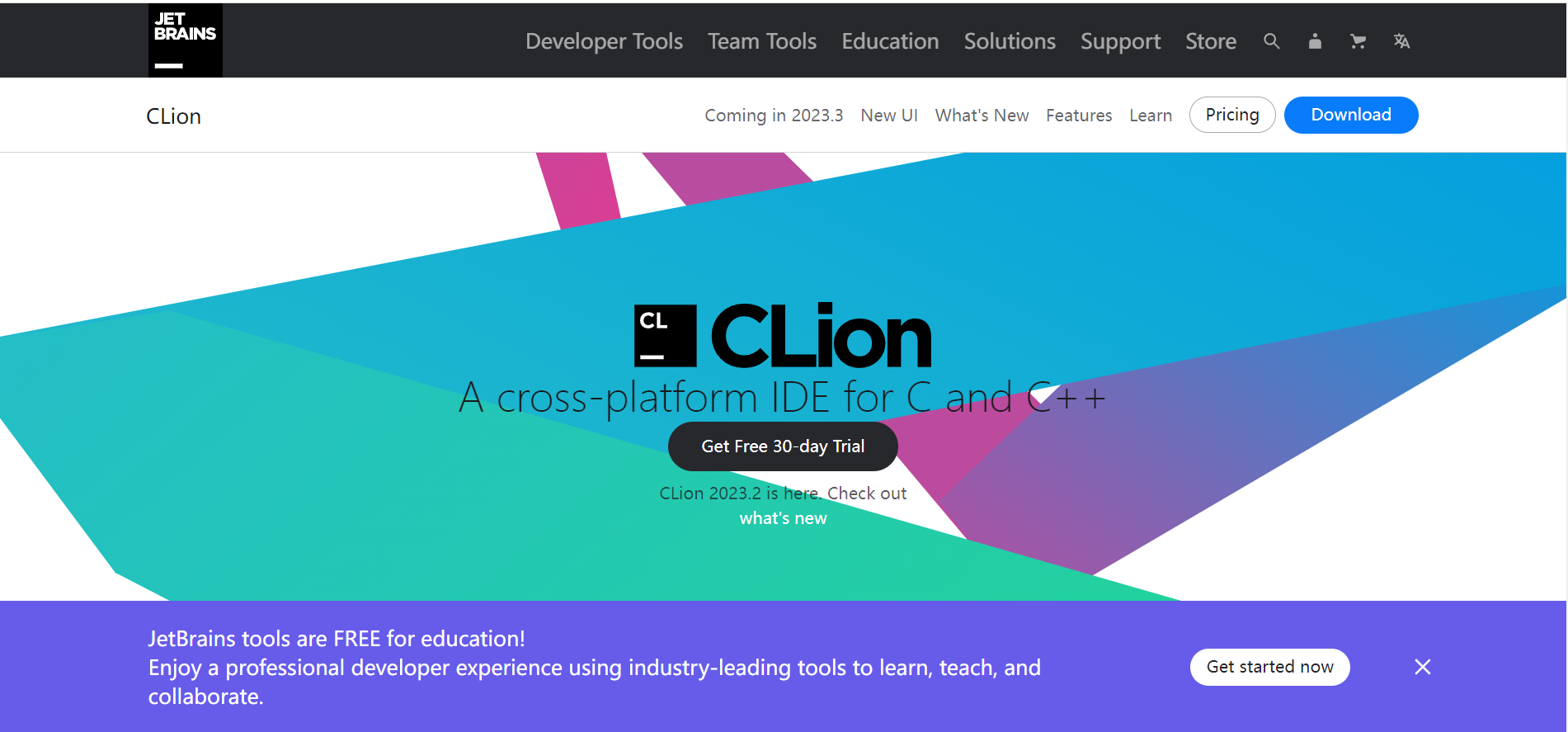
## 下载最新的linux内核linux-5.19.10.tar.xz

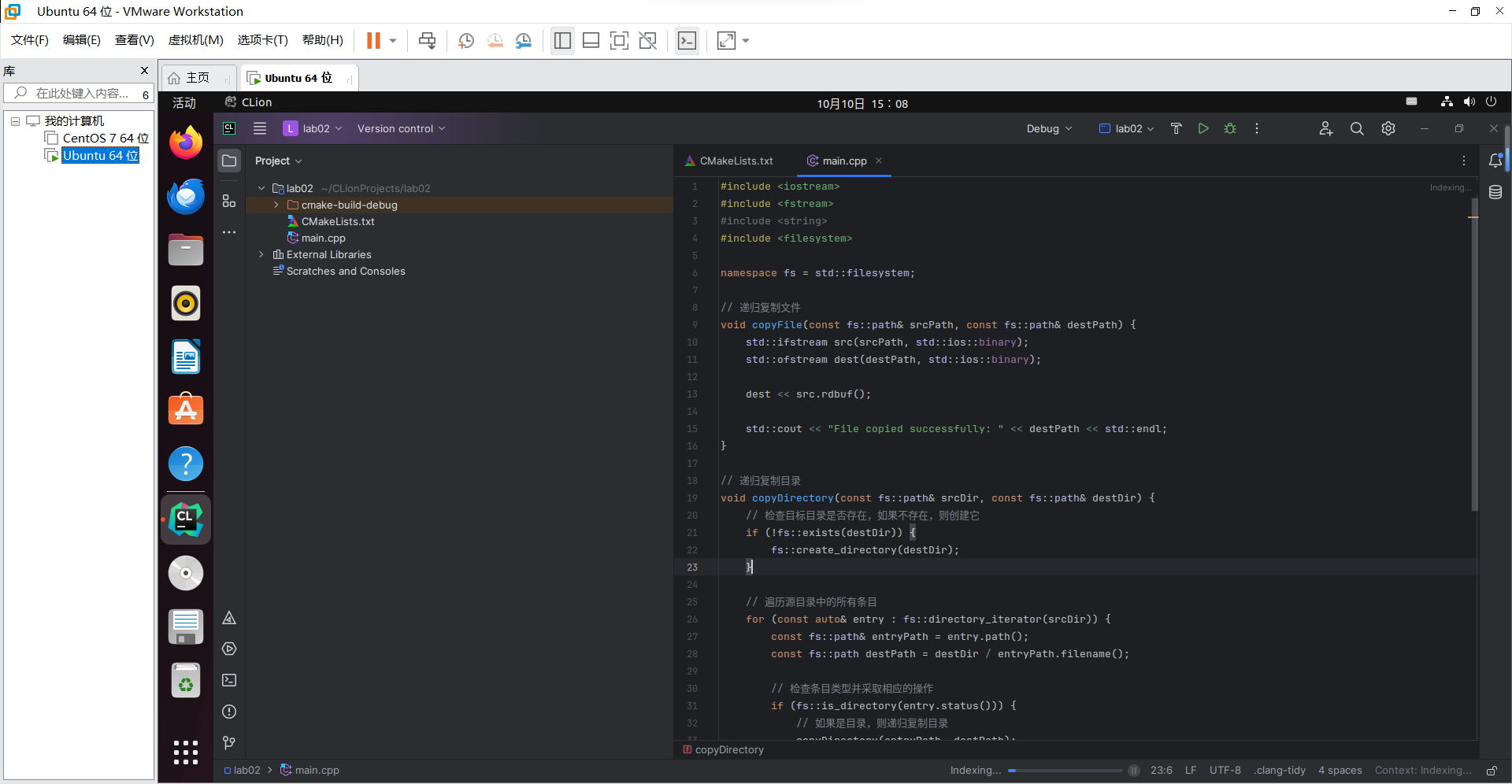
<www.kernel.org>



## 安装使用Clion编译器

<https://www.jetbrains.com/clion>





## 编写C++程序实现copydir

int main() {

// 源目录和目标目录的路径

const fs::path srcDir = "/home/roc/Desktop/jetbra";

const fs::path destDir = "/home/roc/Public";

// 复制目录及其子目录

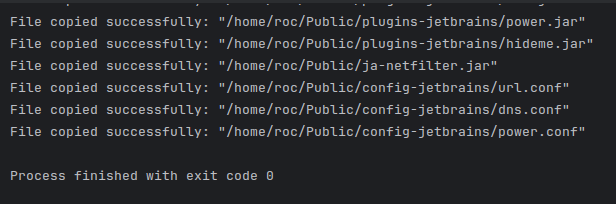
copyDirectory(srcDir, destDir);

return 0;

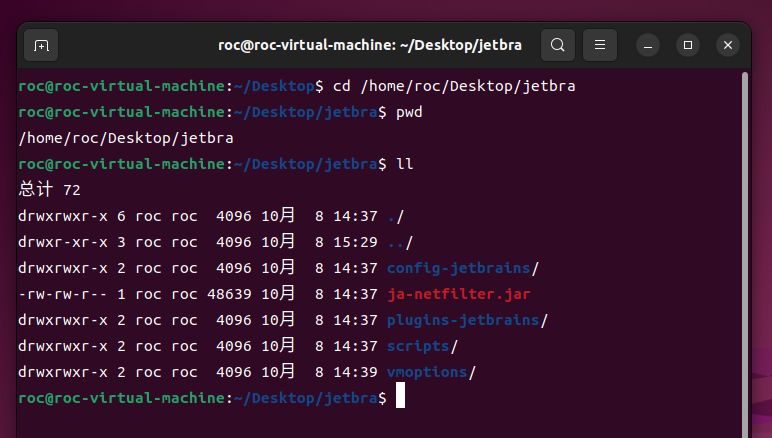
}

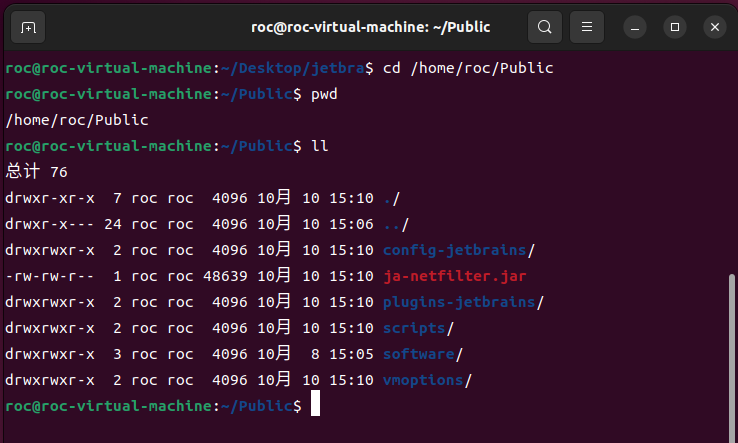
将srcDir = "/home/roc/Desktop/jetbra"复制到destDir = "/home/roc/Public"

运行程序：



## 验证copy的结果





原本以为出了bug，文件数目不对，检查过后发现是之前在public文件夹下装了个软件software。经过进一步验证，copydir没问题。

**附录：源代码**

#include <iostream>

#include <fstream>

#include <string>

#include <filesystem>

namespace fs = std::filesystem;

// 递归复制文件

void copyFile(const fs::path& srcPath, const fs::path& destPath) {

std::ifstream src(srcPath, std::ios::binary);

std::ofstream dest(destPath, std::ios::binary);

dest << src.rdbuf();

std::cout << "File copied successfully: " << destPath << std::endl;

}

// 递归复制目录

void copyDirectory(const fs::path& srcDir, const fs::path& destDir) {

// 检查目标目录是否存在，如果不存在，则创建它

if (!fs::exists(destDir)) {

fs::create\_directory(destDir);

}

// 遍历源目录中的所有条目

for (const auto& entry : fs::directory\_iterator(srcDir)) {

const fs::path& entryPath = entry.path();

const fs::path destPath = destDir / entryPath.filename();

// 检查条目类型并采取相应的操作

if (fs::is\_directory(entry.status())) {

// 如果是目录，则递归复制目录

copyDirectory(entryPath, destPath);

} else if (fs::is\_regular\_file(entry.status())) {

// 如果是普通文件，则复制文件

copyFile(entryPath, destPath);

}

}

}

int main() {

// 源目录和目标目录的路径

const fs::path srcDir = "/home/roc/Desktop/jetbra";

const fs::path destDir = "/home/roc/Public";

// 复制目录及其子目录

copyDirectory(srcDir, destDir);

return 0;

}

#include <iostream>

#include <fstream>

#include <string>

#include <filesystem>

namespace fs = std::filesystem;

// 递归复制文件

void copyFile(const fs::path& srcPath, const fs::path& destPath) {

std::ifstream src(srcPath, std::ios::binary);

std::ofstream dest(destPath, std::ios::binary);

dest << src.rdbuf();

std::cout << "File copied successfully: " << destPath << std::endl;

}

// 递归复制目录

void copyDirectory(const fs::path& srcDir, const fs::path& destDir) {

// 检查目标目录是否存在，如果不存在，则创建它

if (!fs::exists(destDir)) {

fs::create\_directory(destDir);

}

// 遍历源目录中的所有条目

for (const auto& entry : fs::directory\_iterator(srcDir)) {

const fs::path& entryPath = entry.path();

const fs::path destPath = destDir / entryPath.filename();

// 检查条目类型并采取相应的操作

if (fs::is\_directory(entry.status())) {

// 如果是目录，则递归复制目录

copyDirectory(entryPath, destPath);

} else if (fs::is\_regular\_file(entry.status())) {

// 如果是普通文件，则复制文件

copyFile(entryPath, destPath);

}

}

}

int main() {

// 源目录和目标目录的路径

const fs::path srcDir = "/home/roc/Desktop/jetbra";

const fs::path destDir = "/home/roc/Public";

// 复制目录及其子目录

copyDirectory(srcDir, destDir);

return 0;

}