### **移植步骤**

#### **1. 复制必要的文件**

将以下文件复制到你的项目中：

* thrd\_pool.h：头文件，包含线程池的接口定义。
* thrd\_pool.c：源文件，实现线程池的具体功能。
* spinlock.h：自旋锁的实现，线程池内部使用。
* atomic.h：原子操作的实现，线程池内部使用。

#### **2. 编译依赖**

确保你的项目能够正确编译这些文件。由于代码中使用了 POSIX 线程库（pthread），你需要在编译时链接该库。如果你使用的是 GCC 编译器，可以在编译命令中添加 -lpthread 选项。

#### **3. 包含头文件**

在你的代码中包含 thrd\_pool.h 头文件：

#include "thrd\_pool.h"

#### **4. 使用线程池**

以下是一个简单的示例，展示了如何在你的代码中使用线程池：

#include <stdio.h>#include "thrd\_pool.h"

// 任务处理函数void task\_handler(void \*arg) {

int \*num = (int \*)arg;

printf("Processing task with number: %d\n", \*num);

// 释放内存

free(num);}

int main() {

// 创建线程池，指定线程数量为 4

thrdpool\_t \*pool = thrdpool\_create(4);

if (pool == NULL) {

perror("Failed to create thread pool");

return 1;

}

// 添加任务到线程池

for (int i = 0; i < 10; i++) {

int \*num = (int \*)malloc(sizeof(int));

\*num = i;

if (thrdpool\_post(pool, task\_handler, num) != 0) {

perror("Failed to post task");

free(num);

}

}

// 等待所有任务完成

thrdpool\_waitdone(pool);

return 0;}

#### **5. 编译和运行**

编译你的代码，确保链接 pthread 库：

gcc your\_file.c thrd\_pool.c -o your\_program -lpthread

运行生成的可执行文件：

./your\_program