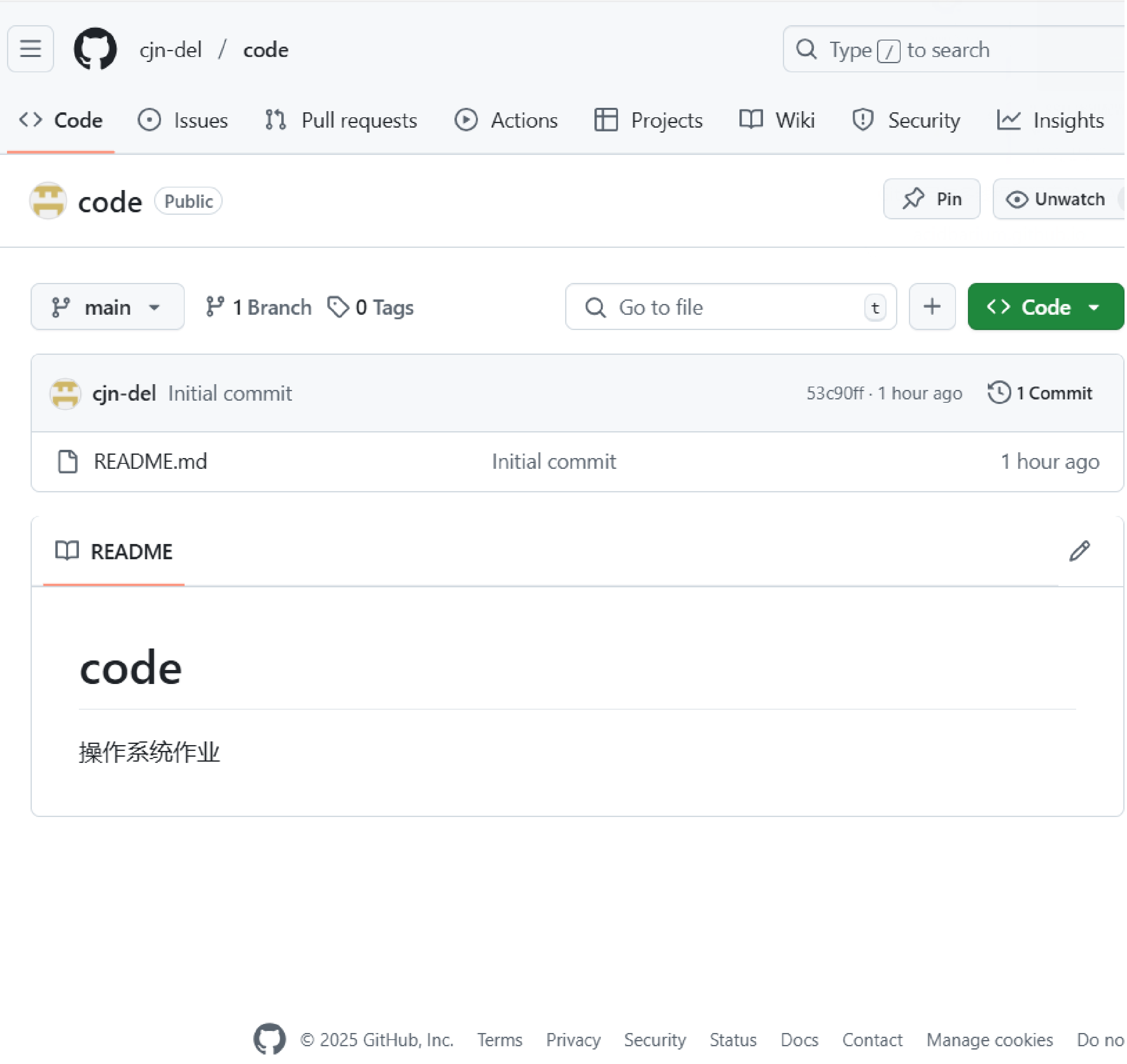
1. GitHub 仓库创建与管理

使用git上传文件到GitHub ![alt text](



code Public

cjn-del Initial commit 53c90ff · 1 hour ago 1 Commit

README.md Initial commit 1 hour ago

code

操作系统作业

© 2025 GitHub, Inc. Terms Privacy Security Status Docs Contact Manage cookies Do no

)

2. 多线程编程实验

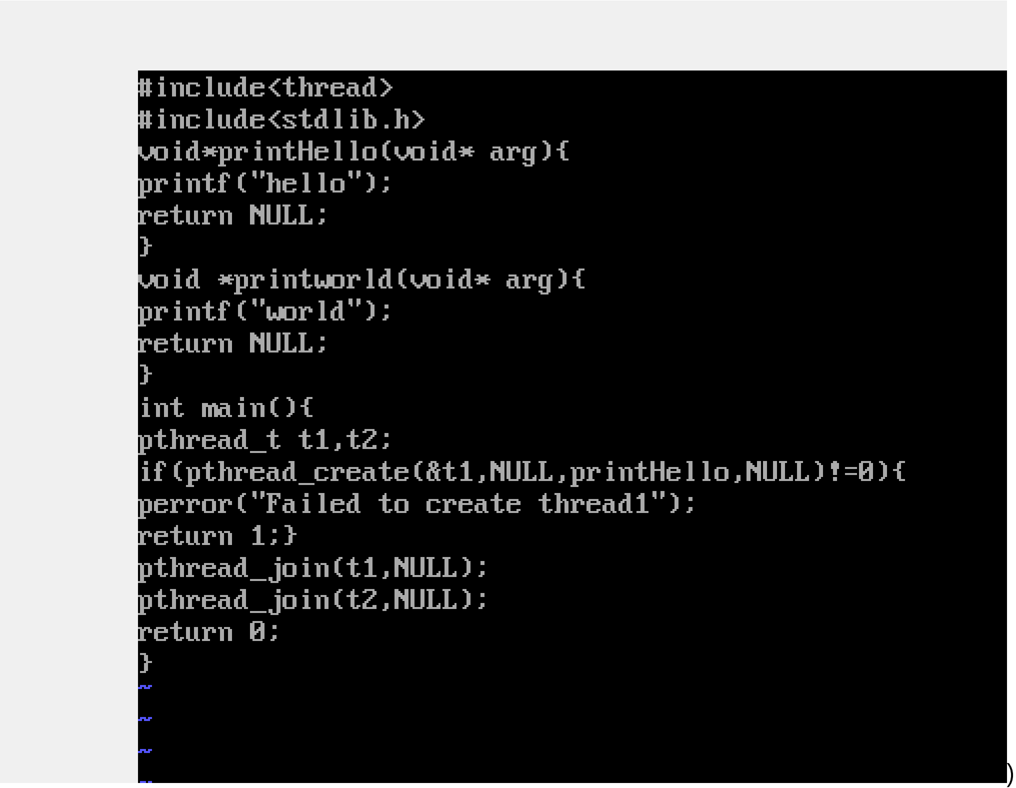
2.1 基本多线程程序设计

首先编写一个简单的多线程程序，该程序使用 `pthread` 库创建一个线程并输出 "helloworld"。以下是代码示例：

```
#include <stdio.h>
#include <pthread.h>

// 线程函数1
```

```c
void* printHello(void* arg) {
    printf("Hello");
    return NULL;
}

// 线程函数2
void* printWorld(void* arg) {
    printf("World");
    return NULL;
}

int main() {
    pthread_t t1, t2;

    // 创建两个线程
    if (pthread_create(&t1, NULL, printHello, NULL) != 0) {
        perror("Failed to create thread 1");
        return 1;
    }
    if (pthread_create(&t2, NULL, printWorld, NULL) != 0) {
        perror("Failed to create thread 2");
        return 1;
    }

    // 等待线程结束
    pthread_join(t1, NULL);
    pthread_join(t2, NULL);

    return 0;
}
```

如下图所示

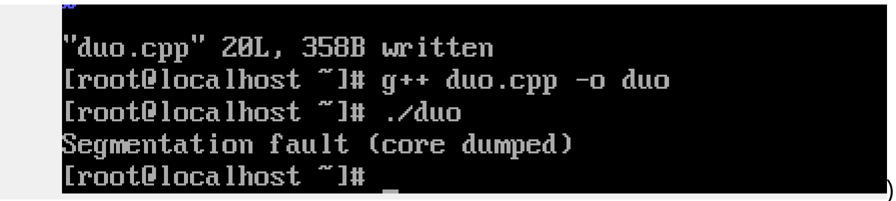
![alt text](

```
#include<thread>
#include<stdlib.h>
void*printHello(void* arg){
printf("hello");
return NULL;
}
void *printworld(void* arg){
printf("world");
return NULL;
}
int main(){
pthread_t t1,t2;
if(pthread_create(&t1,NULL,printHello,NULL)!=0){
perror("Failed to create thread1");
return 1;}
pthread_join(t1,NULL);
pthread_join(t2,NULL);
return 0;
}
~
~
~
```

)

运行结果如下

```
helloworld
```

如下图所示 ![alt text](

```
"duo.cpp" 20L, 358B written
[root@localhost ~]# g++ duo.cpp -o duo
[root@localhost ~]# ./duo
Segmentation fault (core dumped)
[root@localhost ~]# _
```

)