

多线程编程实践报告

一、实验目的

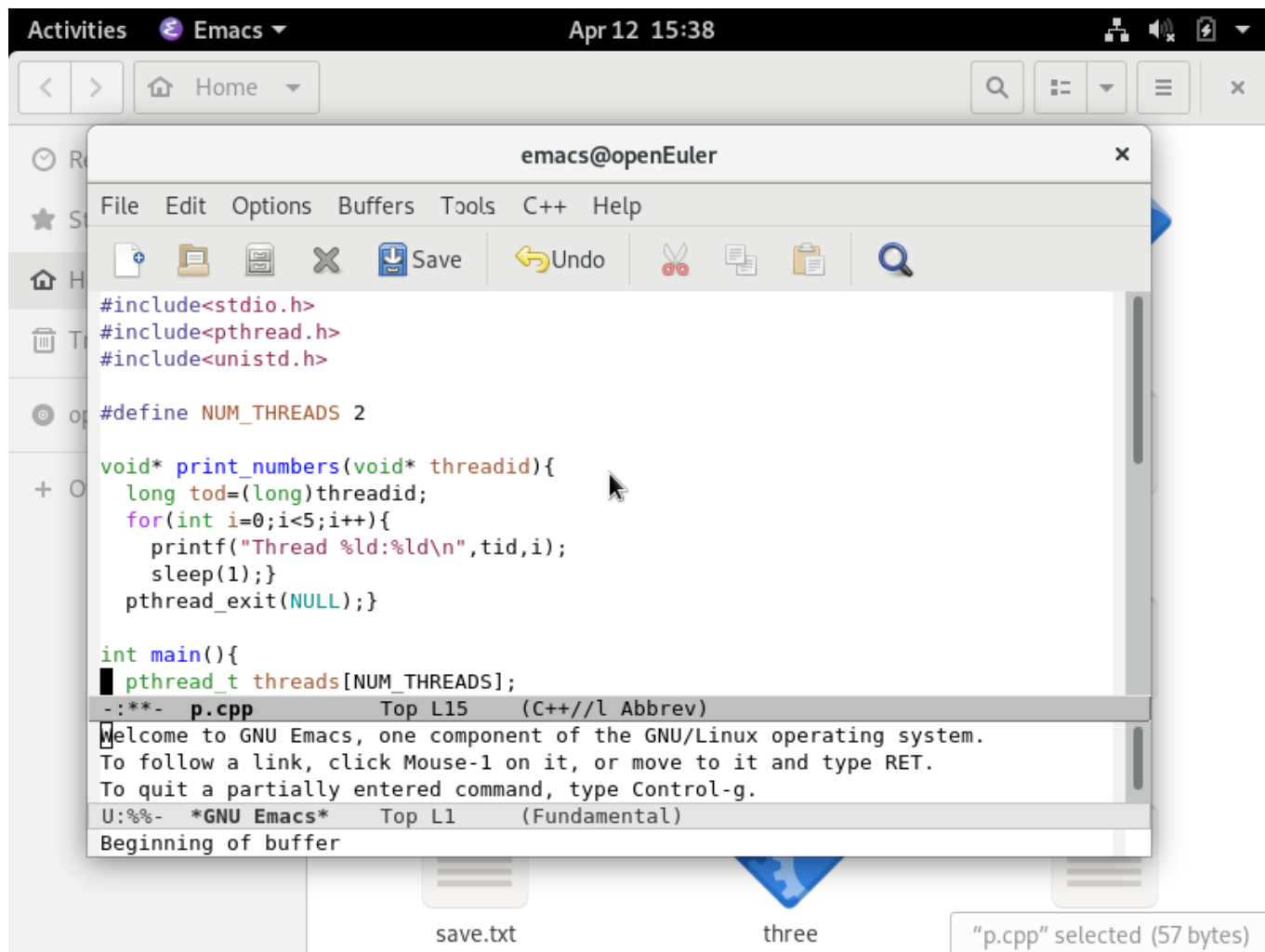
学习使用 pthread 库进行多线程编程 熟悉 Git 工具的使用

二、实验环境

- 操作系统: Linux
- 编程语言: C++

三、实验内容

1.使用 pthread 库编写多线程程序



emacs@openEuler

File Edit Options Buffers Tools C++ Help

Save Undo

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

#define NUM_THREADS 2

void* print_numbers(void* threadid){
    long tod=(long)threadid;
    for(int i=0;i<5;i++){
        printf("Thread %ld:%ld\n",tid,i);
        sleep(1);}
    pthread_exit(NULL);}

int main(){
    pthread_t threads[NUM_THREADS];
```

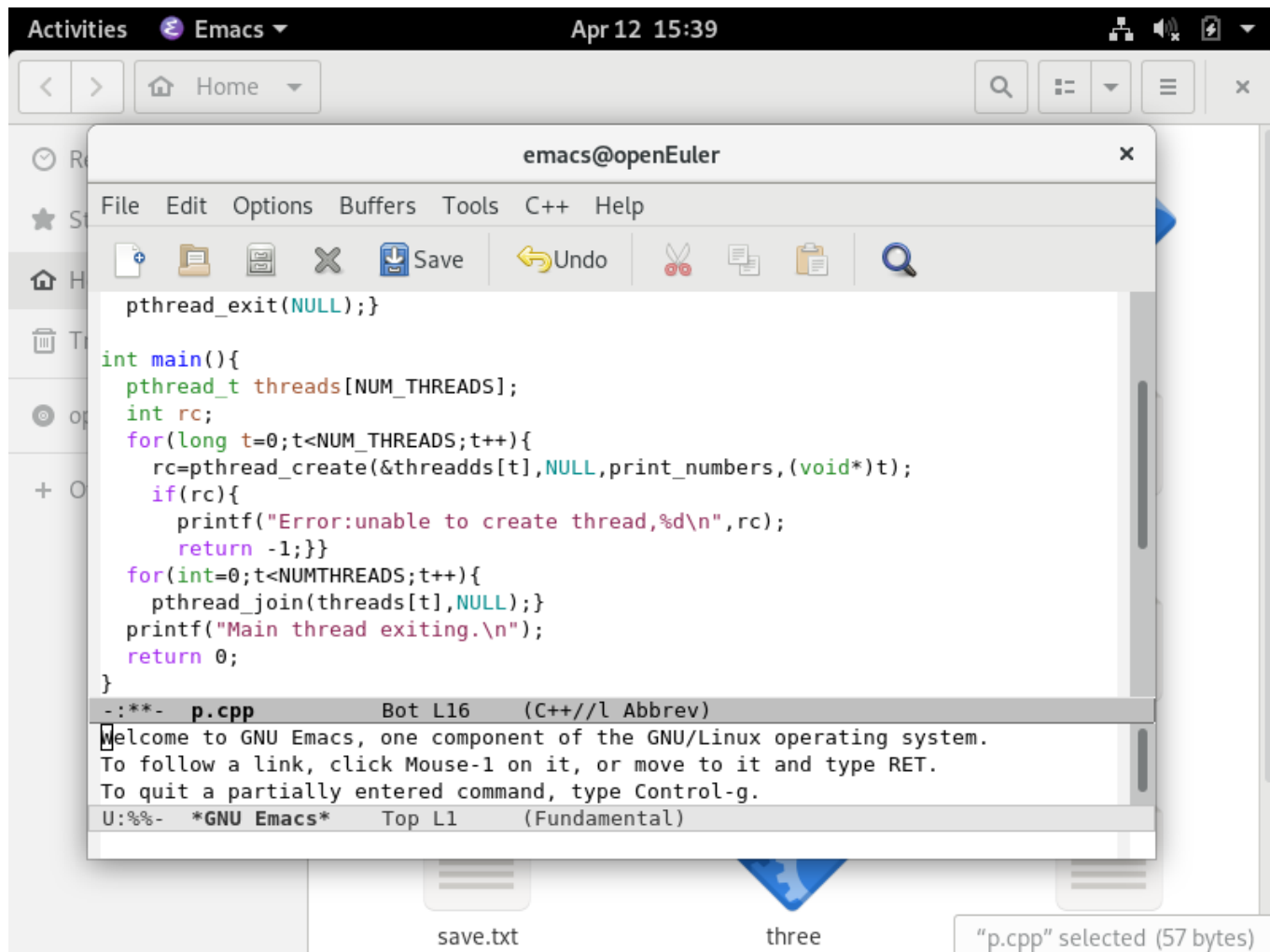
Top L15 (C++//l Abbrev)

Welcome to GNU Emacs, one component of the GNU/Linux operating system.
To follow a link, click Mouse-1 on it, or move to it and type RET.
To quit a partially entered command, type Control-g.

U:%%- *GNU Emacs* Top L1 (Fundamental)

Beginning of buffer

save.txt three "p.cpp" selected (57 bytes)



emacs@openEuler

File Edit Options Buffers Tools C++ Help

Save Undo

```
pthread_exit(NULL);}

int main(){
    pthread_t threads[NUM_THREADS];
    int rc;
    for(long t=0;t<NUM_THREADS;t++){
        rc=pthread_create(&threads[t],NULL,print_numbers,(void*)t);
        if(rc){
            printf("Error:unable to create thread,%d\n",rc);
            return -1;}}
    for(int=0;t<NUMTHREADS;t++){
        pthread_join(threads[t],NULL);}
    printf("Main thread exiting.\n");
    return 0;
}
```

Bot L16 (C++//l Abbrev)

Welcome to GNU Emacs, one component of the GNU/Linux operating system.
To follow a link, click Mouse-1 on it, or move to it and type RET.
To quit a partially entered command, type Control-g.

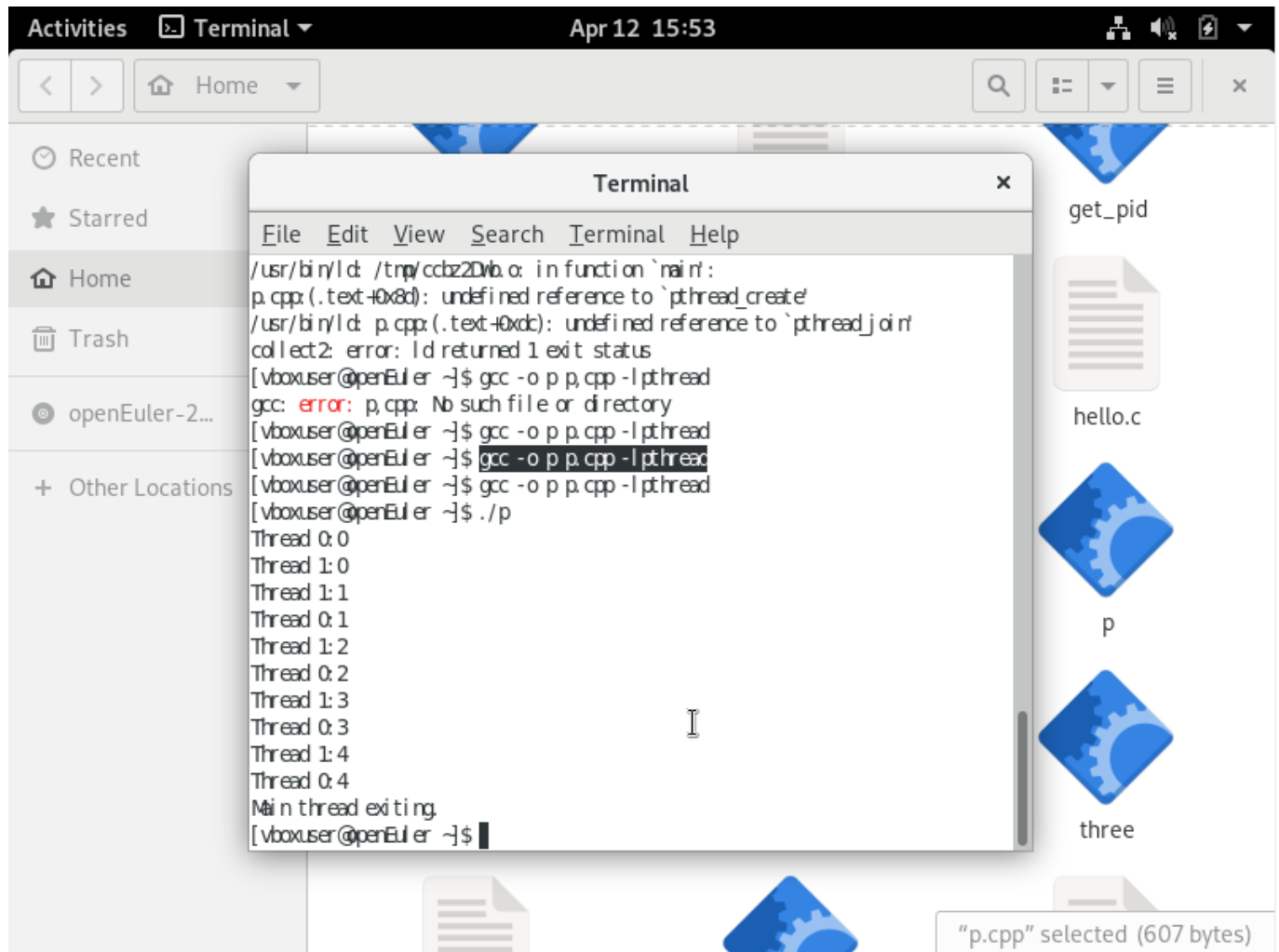
U:%%- *GNU Emacs* Top L1 (Fundamental)

save.txt three "p.cpp" selected (57 bytes)

2.编译和运行程序

使用以下命令编译和运行上述程序：`gcc -o p p.cpp -lpthread`
`./p`

运行结果：



```
Activities Terminal Apr 12 15:53
Recent
Starred
Home
Trash
openEuler-2...
+ Other Locations

Terminal
File Edit View Search Terminal Help
/usr/bin/ld: /tmp/ccbz2DVo.o: in function `main':
p.cpp(.text+0x8d): undefined reference to `pthread_create'
/usr/bin/ld: p.cpp(.text+0xdc): undefined reference to `pthread_join'
collect2: error: ld returned 1 exit status
[vboxuser@openEuler ~]$ gcc -o p p.cpp -lpthread
gcc: error: p.cpp: No such file or directory
[vboxuser@openEuler ~]$ gcc -o p p.cpp -lpthread
[vboxuser@openEuler ~]$ gcc -o p p.cpp -lpthread
[vboxuser@openEuler ~]$ gcc -o p p.cpp -lpthread
[vboxuser@openEuler ~]$ ./p
Thread 0:0
Thread 1:0
Thread 1:1
Thread 0:1
Thread 1:2
Thread 0:2
Thread 1:3
Thread 0:3
Thread 1:4
Thread 0:4
Main thread exiting.
[vboxuser@openEuler ~]$
```

3.使用 Git 管理项目

创建项目目录并进入：`mkdir os_practice`

`cd os_practice`

初始化 Git 仓库：`git init` 创建子目录并添加代码：`mkdir hw3_pthread`

`cp path_to_your_code/p.cpp hw3_pthread/` 添加所有文件并提交：`git add .`

`git commit -m "Add pthread example program"`

