



北京邮电大学

Beijing University of Posts and Telecommunications

# 操作系统 实验报告

[进程控制实验]

学院：计算机学院  
2020211376 马天成  
2022 年 10 月 15 日

## 北京邮电大学《操作系统》课程实验报告

[illegible]

注：评语要体现每个学生的工作情况，可以加页。

## 目录

1. 实验目的和要求.....	4
2. 实验内容.....	4
2.1 准备工作.....	4
2.3 process.....	4
3. 实验总结.....	6

# 1. 实验目的和要求

Write a multithreaded program that calculates various statistical values for a list of numbers. This program will accept a series of numbers from the keyboard and will then create three separate worker threads. One thread will determine the average of the numbers, the second will determine the maximum value, and the third will determine the minimum value.

翻译一下就是输入一串数字，然后进行计算这个数组的最大值。最小值和平均值。

## 2. 实验内容

### 2.1 准备工作

首先准备好 linux 平台。我这里使用的是 CentOS-7-x86\_64-DVD-2009  
此外，也要准备好 gcc 环境。这里使用 yum 直接一键解决了。

代码编写的思路也很简单。照着 PPT 上的思路复现一遍基本上就可以了。

### 2.3 process

基本上就是如何使用 thread 函数。

```
...  
  
void* cal_maximum(void*paragram);  
void* cal_minimun(void*paragram);  
void* cal_average(void*paragram);  
  
int main(void)  
{  
    ...  
    // start thread  
    pthread_t pid1;  
    pthread_t pid2;  
    pthread_t pid3;  
    pthread_create(&pid1, NULL, cal_maximum, datas);  
    pthread_create(&pid2, NULL, cal_minimun, datas);  
    pthread_create(&pid3, NULL, cal_average, datas);  
    pthread_join(pid1, NULL);  
    pthread_join(pid2, NULL);
```

```

    pthread_join(pid3, NULL);
    ...
}

void* cal_maximum(void* paragram)
{
    ...
    pthread_exit(0);
}

void* cal_minimun(void* paragram)
{
    ...
    pthread_exit(0);
}

void *cal_average(void* paragram)
{
    ...
    pthread_exit(0);
}

```

上述代码展现了如何使用 `thread` 函数的框架：

- 先写出三个函数指针对应的代码空间，写出代码运行的功能。
- 主函数里申请三个线程，然后分别跑三个函数指针对应的代码块，分别计算出最大值最小值平均值。
- 最后在主函数里输出结果。

执行结果：

```

[spike@localhost Lab3]$ gcc pthread.c -o pthread -lpthread -std=c99
[spike@localhost Lab3]$ ./pthread
101
minimum=101
maximum=101
average=101.000000
[spike@localhost Lab3]$ ./pthread
13 14 141 5425 45 425 543
minimum=13
maximum=5425
average=943.714294
[spike@localhost Lab3]$

```

### 3. 实验总结

在这个实验里，我们主要进行了代码的编写。在这个过程中，显示的理解了代码的运作原理和设计思路。但是我们在深究的过程中，发现操作系统对于这一系列操作的支持是有迹可循的：

- **thread**

一个线程，可以在一个进程里申请。这里是一个多（多个线程）对一（一个进程）的关系。线程共享进程的地址空间，共享进程的全局变量。所以在这里线程的操作也很简单，单纯就是一个代码块的运行单元（实质上也是这样）。