

Project 3: 多线程矩阵乘法计算器

516030910259 刘欣鹏

1. 实验目的

通过编写多线程矩阵乘法计算器，加深对线程的认识与理解。

2. 实验原理

给定两个矩阵， A 为 M 行 K 列， B 为 K 行 N 列，则 A 和 B 的乘积为一个 M 行 N 列的

矩阵。对于矩阵 C ，有 $C_{i,j} = \sum_{k=1}^K A_{i,k} \times B_{k,j}$ 。

对于该项目，计算每一个 $C_{i,j}$ 是一个独立的工作线程，一次它将会涉及生成 $M \times N$ 个线程。主线程将初始化矩阵 A 和 B ，并分配足够的内存给矩阵 C 以容纳矩阵 A 和 B 的积。这些矩阵将声明为全局数据，以使每个工作线程都能访问矩阵 A 、 B 和 C 。

本实验在 Linux 环境下以 Pthread 库进行。

3. 实验步骤

3.1. 源代码

```
#include <pthread.h>
#include <stdio.h>
#include <stdlib.h>
int n,k,m;
int a[50][50],b[50][50],c[50][50];
struct node{
    int x;
    int y;
};//用于传递参数的数据结构，表示当前线程需计算的值为  $C_{i,j}$ 

void *runner(struct node * param){
    c[param->x][param->y]=0;
    int i;
    for (i=0;i<k;i++)
        c[param->x][param->y]+=a[param->x][i]*b[i][param->y];
    printf("%d\n",pthread_self());//输出当前线程 tid 以供调试
    pthread_exit(0);
}//子线程执行的函数

int main()
{
    int i,j,x;
    scanf("%d%d%d",&n,&k,&m);
    for (i=0;i<n;i++)
        for (j=0;j<k;j++)
            scanf("%d",&a[i][j]);
    for (i=0;i<k;i++)
```

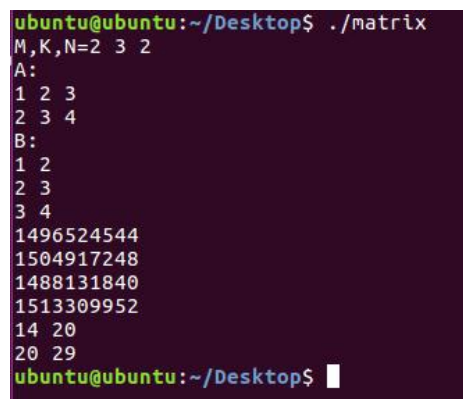
```

        for (j=0;j<m;j++)
            scanf("%d",&b[i][j]);
//读入数据
pthread_t tid[2501];
pthread_attr_t attr[2501];

for (i=0;i<n;i++)
    for (j=0;j<m;j++){
        x=i*m+j;
        pthread_attr_init(&attr[x]);
        struct node *data=(struct node *) malloc(sizeof(struct
node));
        data->x=i;
        data->y=j;
        pthread_create(&tid[x],&attr[x],runner,data);
        //创建用于计算的  $M \times N$  个线程
    }
for (i=0;i<m*n;i++)
    pthread_join(tid[i],NULL);
//等待这些线程结束
for (i=0;i<n;i++)
{
    for (j=0;j<m;j++) printf("%d ", c[i][j]);
    printf("\n");
} //输出结果
}

```

3.2. 测试



```

ubuntu@ubuntu:~/Desktop$ ./matrix
M,K,N=2 3 2
A:
1 2 3
2 3 4
B:
1 2
2 3
3 4
1496524544
1504917248
1488131840
1513309952
14 20
20 29
ubuntu@ubuntu:~/Desktop$

```

测试数据中, A 为 2×3 矩阵 $\begin{matrix} 1 & 2 & 3 \\ 2 & 3 & 4 \end{matrix}$, B 为 3×2 矩阵 $\begin{matrix} 1 & 2 \\ 2 & 3 \\ 3 & 4 \end{matrix}$ 。计算, 得到 2×2

矩阵 C : $\begin{matrix} 14 & 20 \\ 20 & 29 \end{matrix}$ 。四个子线程的 tid 如上图所示。

4. 心得与体会

通过本实验，我对 linux 下中用于线程管理的 pthread 库有了初步的认识。

同时，结合上次实验中所涉及的进程，我也更直观的认识到了进程和线程的区别，尤其是它们在共享资源上的不同：子线程与父线程共享全局变量，但子进程与父进程不。