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。

1. 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。

对于该项目，计算每一个是一个独立的工作线程，一次它将会涉及生成个线程。主线程将初始化矩阵和，并分配足够的内存给矩阵以容纳矩阵和的积。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、和。

本实验在Linux环境下以Pthread库进行。

1. 实验步骤
   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

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++)

{

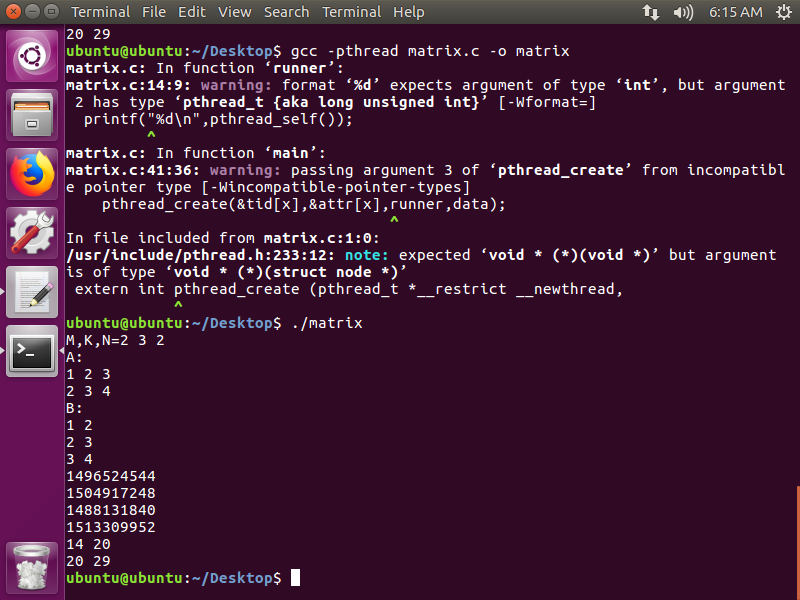
for (j=0;j<m;j++) printf("%d ", c[i][j]);

printf("\n");

}//输出结果

}

* 1. 测试



测试数据中，为矩阵，为矩阵。计算，得到矩阵：。四个子线程的tid如上图所示。

1. 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。