

操作系统实验三

汪航 2021211114

2023 年 10 月 27 日

1 实验内容

本次实验共编写两道程序, 第一个程序使用 POSIX 标准的 pthread 库编写了多线程执行求平均数、求最大值、求最小值的程序; 第二个程序使用了 pthread 库编写了矩阵乘程序。

2 实验环境

Linux 操作系统,gcc 编译

3 程序一

3.1 主要内容

以多个数字为输入, 使用 pthread 库编写求这多个数字的平均数、最大值、最小值的多线程程序。

使用 **API:** pthread_attr_init, pthread_create, pthread_join

3.2 程序设计思路

开辟三个线程, 三个线程分别用于计算平均数、最大值、最小值, 平均数、最大值、最小值设置为全局变量, 父子线程之间通过全局变量传递信息, 最终结果由父线程打印。

程序实现:

```
1 #include <pthread.h>
2 #include <stdio.h>
3 #include <stdlib.h>
4 int minimum = __INT32_MAX__;
5 int maximum = -__INT32_MAX__ - 1;
6 int average = 0;
7 int numbercount = 0;
8 void *claverage(char *param[]);
9 void *clminimum(char *param[]);
10 void *clmaximum(char *param[]);
11 int main(int argc, char *argv[])
12 {
13     pthread_t workers[3];
14     pthread_attr_t attr;
15     numbercount = argc - 1;
16
17     pthread_attr_init(&attr);
18
19     pthread_create(&workers[0], &attr, claverage, argv);
20     pthread_create(&workers[1], &attr, clminimum, argv);
21     pthread_create(&workers[2], &attr, clmaximum, argv);
22     for (int i = 0; i < 3; i++)
23     {
24         pthread_join(workers[i], NULL);
25     }
26     printf("The average value is %d\n", average);
27     printf("The minumum value is %d\n", minimum);
28     printf("The maximum value is %d\n", maximum);
29     return 0;
30 }
31 void *claverage(char *param[])
32 {
33     for (int i = 1; i <= numbercount; i++)
34     {
35         average += atoi(param[i]);
36     }
37     average = average / numbercount;
38 }
39 void *clminimum(char *param[])
40 {
41     for (int i = 1; i <= numbercount; i++)
42     {
43         int k = atoi(param[i]);
44         if (k < minimum)
45         {
46             minimum = k;
```

```
47     }  
48 }  
49 }  
50 void *clmaximum(char *param[])  
51 {  
52     for (int i = 1; i <= numbercount; i++)  
53     {  
54         int k = atoi(param[i]);  
55         if (k > maximum)  
56         {  
57             maximum = k;  
58         }  
59     }  
60 }
```

3.3 程序测试结果及测试结果分析

输入: ./lab3 90 81 78 95 79 72 85

输出: The average value is 82

The minumum value is 72

The maximum value is 95

输入: ./lab3 29 281 4 2 44 2 45 31 563

输出: The average value is 111

The minumum value is 2

The maximum value is 563

输入: ./lab3 -2 3 -212 213 -2 -6

输出: The average value is -1

The minumum value is -212

The maximum value is 213

经分析, 该程序能够正常执行期望的求平均数、最大值、最小值功能。

```
wanghang@wanghangpc:~/oslab3$ ./lab3 90 81 78 95 79 72 85
The average value is 82
The minumum value is 72
The maximum value is 95
```

图 1: 运行结果

```
wanghang@wanghangpc:~/oslab3$ ./lab3 29 281 4 2 44 2 45 31 563
The average value is 111
The minumum value is 2
The maximum value is 563
wanghang@wanghangpc:~/oslab3$ ./lab3 -2 3 -212 213 -2 -6
The average value is -1
The minumum value is -212
The maximum value is 213
```

图 2: 运行结果

4 程序二

4.1 主要内容

给定矩阵 A,B, 求 $A*B$ 的结果 C.

使用 API: `pthread_attr_init`, `pthread_create`, `pthread_join`

4.2 程序设计思路

A 为 $M*K$ 矩阵, B 为 $K*N$ 矩阵, 则对每一个 $C_{i,j}$ 都可以开辟一个线程进行 $C_{i,j}$ 的计算, 计算 $C_{i,j}$ 的线程执行了 $\sum_{n=1}^k A_{i,n}B_{n,j}$ 的计算。

程序实现:

```
1 #include<pthread.h>
2 #include<stdio.h>
3 #include<stdlib.h>
4 #define M 3
5 #define K 2
6 #define N 3
7 struct v
8 {
9     int row;
```

```
10     int column;
11 };
12
13 int A[M][K] = {{1, 4}, {2, 5}, {3, 6}};
14 int B[K][N] = {{8,7,6},{5,4,3}};
15 int C[M][N]={0};
16 void *runner(void * param)
17 {
18
19     int row = ((struct v *)param)->row;
20     int column = ((struct v *)param)->column;
21     for (int i = 0; i < K;i++)
22     {
23         C[row][column] += A[row][i] * B[i][column];
24     }
25 }
26 int main()
27 {
28     pthread_attr_t attr;
29     pthread_attr_init(&attr);
30     pthread_t workers[M * N];
31     for (int i = 0; i < M; i++)
32     {
33         for (int j = 0; j < N;j++)
34         {
35             struct v *data = (struct v *)malloc(sizeof (struct v));
36             data->row = i;
37             data->column = j;
38             pthread_create(&workers[i * N + j], &attr, runner, data);
39         }
40     }
41     for (int i = 0; i < M;i++)
42     {
43         for (int j = 0; j < N;j++)
44         {
45             pthread_join(workers[i * N + j], NULL);
46         }
47     }
48     for (int i = 0; i < M;i++)
49     {
50         for (int j = 0; j < N;j++)
51         {
52             printf("%d ", C[i][j]);
53         }
54         printf("\n");
55     }
```

56 }

4.3 程序测试结果及测试结果分析

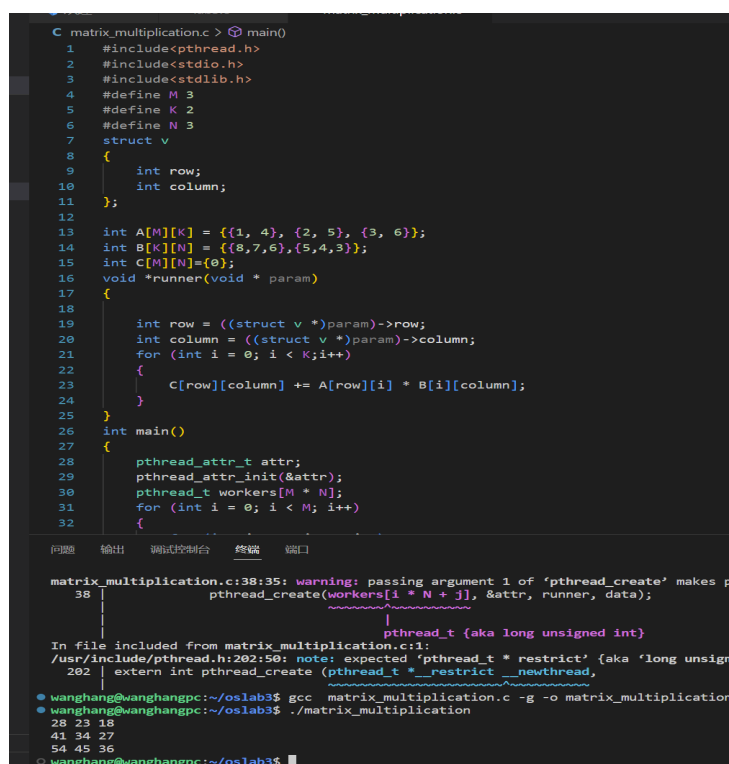
输入: A:{{1, 4}, {2, 5}, {3, 6}}, B:{{8,7,6},{5,4,3}};

输出: 28 23 18

41 34 27

54 45 36

经分析, 该程序能够正常执行期望的矩阵相乘功能。



```
C matrix_multiplication.c > main()
1 #include<pthread.h>
2 #include<stdio.h>
3 #include<stdlib.h>
4 #define M 3
5 #define K 2
6 #define N 3
7 struct v
8 {
9     int row;
10    int column;
11 };
12
13 int A[M][K] = {{1, 4}, {2, 5}, {3, 6}};
14 int B[K][N] = {{8,7,6},{5,4,3}};
15 int C[M][N]={0};
16 void *runner(void * param)
17 {
18
19     int row = ((struct v *)param)->row;
20     int column = ((struct v *)param)->column;
21     for (int i = 0; i < K;i++)
22     {
23         C[row][column] += A[row][i] * B[i][column];
24     }
25 }
26
27 int main()
28 {
29     pthread_attr_t attr;
30     pthread_attr_init(&attr);
31     pthread_t workers[M * N];
32     for (int i = 0; i < M; i++)
33     {
34         pthread_create(&workers[i * N + j], &attr, runner, data);
35     }
36
37     pthread_t {aka long unsigned int}
38
39     In file included from matrix_multiplication.c:1:
40     /usr/include/pthread.h:202:50: note: expected 'pthread_t * restrict' (aka 'long unsigned int * restrict') but argument 2 has type 'pthread_t * restrict'
41     202 | extern int pthread_create(pthread_t * restrict __newthread,
42
43 wanghang@wanghangpc:~/oslab$ gcc matrix_multiplication.c -g -o matrix_multiplication
44 wanghang@wanghangpc:~/oslab$ ./matrix_multiplication
45 28 23 18
46 41 34 27
47 54 45 36
48 wanghang@wanghangpc:~/oslab$
```

图 3: 运行结果