
云南大学数学与统计学实验教学中心

实验报告

课程名称：操作系统实验	学期：2015~2016 学年上学期	成绩：
指导教师： 彭程	学生姓名：金洋	学号：20131910023
实验名称：进程并发		
实验编号：五	实验日期： 12 月 1 日	实验学时： 1
学院： 数学与统计学院	专业： 信息与计算科学	年级： 2013 级

一、实验目的

了解系统中多线程的并发执行机制；

二、实验内容

1. 创建三个线程（2, 3, 4）来直观地模拟进程执行时的过程，显示在屏幕上。以此来分析进程的初始状态和进度都是随机的，不可控制的，即异步性；
2. `getCurrentTime()`：/*得到当前世界时间： 年/月/日 时:分:秒:毫秒 星期*/；
为增加运行时间，打印 20 次时间；
3. `getPrimer()` 得到 1000 到 1100 之间的素数；
4. `primMain()`：通过 Prim 算法，实现判断一个无向加权图是否连通，若连通，则求其一种最小支撑树；

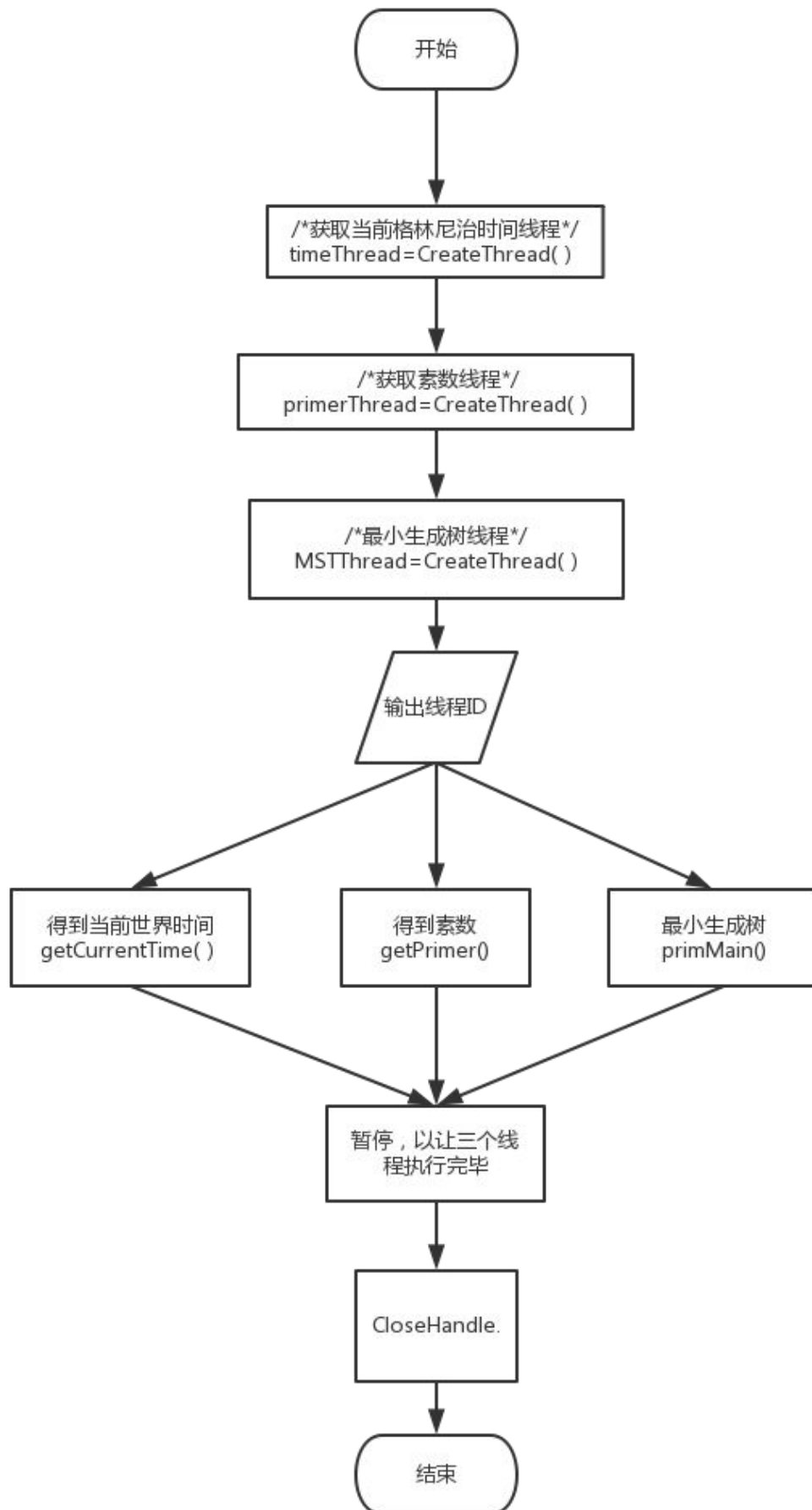
三、实验环境

平台：Visual C++ 6.0

语言：C 语言

四、算法介绍

- 1.流程图：



-
2. `getCurrentTime()` /*得到当前世界时间： 年/月/日 时:分:秒:毫秒 星期*/;
为增加运行时间，打印 20 次时间；
 3. `getPrimer()` 得到 1000 到 1100 之间的素数；
 4. `primMain()`: 通过 Prim 算法，实现判断一个无向加权图是否连通，若连通，则求其一种最小支撑树；
 5. 最小生成树(Minimal Spanning Trees, MST)算法：

Algorithm Prim(MST)

Input: the associated matrix M of a weighted graph $G=(V,E,W)$;

Output: a Minimal Spanning Tree of G or " G is not connected.";

Begin

Step1: $X=\{v_1\}, F=\emptyset$;

Step2: while $((X \neq V) \text{ and } (\delta(X) \neq \emptyset))$ do

1. choose an edge $e=uv \in \delta(X)$ which has the minimum

weight; $\delta(X)=\{e=uv \in R | u \in X \text{ and } v \in V-X\}$

2. $X=X \cup \{v\}$;

3. $F=F \cup \{(u,v)\}$;

Step3: if $(X=V)$ then output MST $T=(V,F)$

else output " G is not connected.";

End.

五、调试过程

1、程序代码

```
#include <stdio.h>
#include <windows.h>
#include <time.h> //获取时间的头文件

/*得到当前世界时间： 年/月/日 时:分:秒:毫秒 星期*/
/*LPVOID 是一个没有类型的指针，也就是说可以将任意类型的指针赋值给
LPVOID 类型的变量（一般作为参数传递），然后在使用的时候在转换回来。*/
DWORD WINAPI getCurrentTime(LPVOID param)
{
    SYSTEMTIME currentTime;
    for (int i=0;i<20;i++){
        GetSystemTime(&currentTime);

        printf("time%d: %u/%u/%u %u:%u:%u:%u %d\n",i,currentTime.wYea
r,currentTime.wMonth,currentTime.wDay,currentTime.wHour,curren
tTime.wMinute,currentTime.wSecond,currentTime.wMilliseconds,cu
rrentTime.wDayOfWeek);

    }

    return 0;
}

/*得到 standN 和 endN 之间的素数*/
DWORD WINAPI getPrimer(LPVOID param)
{

    int startN=1000,endN=1100;

    /*为了使运行时间适当增大，此处求素数的算法未做任何优化*/
    for (int k=startN;k<=endN;k++)
    {
```

```

        for(int i = 2; i < k; i++)
            if (k %i == 0) break;
        if (i==k) printf("%d is a primer.\n",k);

    }

    return 0;

}

```

/*通过 Prim 算法，实现判断一个无向加权图是否连通，若连通，则求其一种最小支撑树；*/

/*程序原理见算法介绍*/

```
#define maxnum 999999999
```

```
int X[100],F[2][100],Fnum,sum,n=8;
```

/*M[1*n,1*n]为无向加权图*/

/*此处为了减少输入工作量，提前将 M 指定*/

```
int M[9][9]={
```

```
    {0,0,0,0,0,0,0,0,0},
```

```
    {0,0,3,2,0,0,0,0,0},
```

```
    {0,3,0,3,6,0,0,0,0},
```

```
    {0,2,3,0,0,5,5,0,0},
```

```
    {0,0,6,0,0,2,0,0,2},
```

```
    {0,0,0,5,2,0,4,3,0},
```

```
    {0,0,0,5,0,4,0,4,0},
```

```
    {0,0,0,0,0,3,4,0,6},
```

```
    {0,0,0,0,2,0,0,6,0}
```

```
};
```

```
int equal()
{
    int i;
    for(i=1;i<=n;i++)
        if (X[i]==0) return(0);
    return(1);
}
```

```
int dota()
{
    int i,j;
    for(i=1;i<=n;i++)
        for (j=1;j<=n;j++)
            if ((i!=j)&&(X[i])&&(X[j]==0)&&(M[i][j]<maxnum))
return(1);
    return(0);
}
```

```
void prim()
{
    int i,j,min,v1=0,v2=0;
    min=maxnum;
    for(i=1;i<=n;i++)
        if (X[i])

            for(j=1;j<=n;j++)
                if ((!X[j])&&(i!=j))

                    if (M[i][j]<min)
                    {
                        min=M[i][j];
                        v1=i;
                        v2=j;
                    }
            }
```

```

        }
    if (min!=maxnum)
    {
        X[v2]=1;
        Fnum++;
        F[0][Fnum]=v1;
        F[1][Fnum]=v2;
        sum=sum+min;
    }
}

/*最下生成树算法主函数*/
DWORD WINAPI primMain(LPVOID param)
{
    int i,j;

    for (i=1;i<=n;i++)
        for (j=1;j<=n;j++)
        {

            if (M[i][j]==0) M[i][j]=maxnum;
        }

    for (i=1;i<=n;i++) X[i]=0;
    X[1]=1;

    while ((!equal())&&(dota())) prim();

    if (equal())
    {
        for(i=1;i<=Fnum;i++)
            printf("%c,%c\n",F[0][i]+64,F[1][i]+64);
        printf("The total weight is %d.",sum);
    }
}

```

```
    }  
    else printf("G is not connected.");  
  
    return 0;  
}
```

```
void main()  
{  
    HANDLE timeThread,primerThread,MSTThread;//定义三个线程  
    DWORD id1,id2,id3;//分别表示三个线程的 id  
  
    /*以下创建三个线程，且创建后先不进入执行队列*/  
  
    /*获取当前格林尼治时间线程*/  
    timeThread=CreateThread(NULL,0,getCurrentTime,NULL,CREATE_S  
USPENDED,&id1);  
  
    //getCurrentTime 函数名，需要定义其实体；要求第一个函数只打印一句  
    话，调用系统时间  
    //第二个分配大小，0 表示动态；第三个参数指向线程的函数指针；第四个  
    传递参数，第五个参数为 CREATE_SUSPENDED 表示将创建好的进程挂起，若为 0  
    则创建好后即可执行；最后一个线程的 ID 地址、ID 变量  
  
    /*获取素数线程*/  
    primerThread=CreateThread(NULL,0,getPrimer,NULL,CREATE_SUSP  
ENDED,&id2);  
  
    /*最小生成树线程*/  
    /*实现用 Prim 算法判断一个无向加权图是否连通，若连通，则求其一种最
```

小支撑树; */

```
    MSTThread=CreateThread(NULL,0,primMain,NULL,CREATE_SUSPENDED,&id3);
```

```
    printf("\n获取当前格林尼治时间\n线程 ID 为:%d\n",id1);
```

```
    printf("\n获取素数\n线程 ID 为:%d\n",id2);
```

```
    printf("\n最小生成树\n线程 ID 为:%d\n",id3);
```

```
    //system("pause");//主线程运行 DOS 命令 pause
```

```
    //printf("你按任意键后，三个线程将被恢复\n");
```

```
    ResumeThread(timeThread);//线程恢复函数
```

```
    ResumeThread(primerThread);
```

```
    ResumeThread(MSTThread);
```

```
    system("pause");
```

```
    ::CloseHandle(timeThread);
```

```
    ::CloseHandle(primerThread);
```

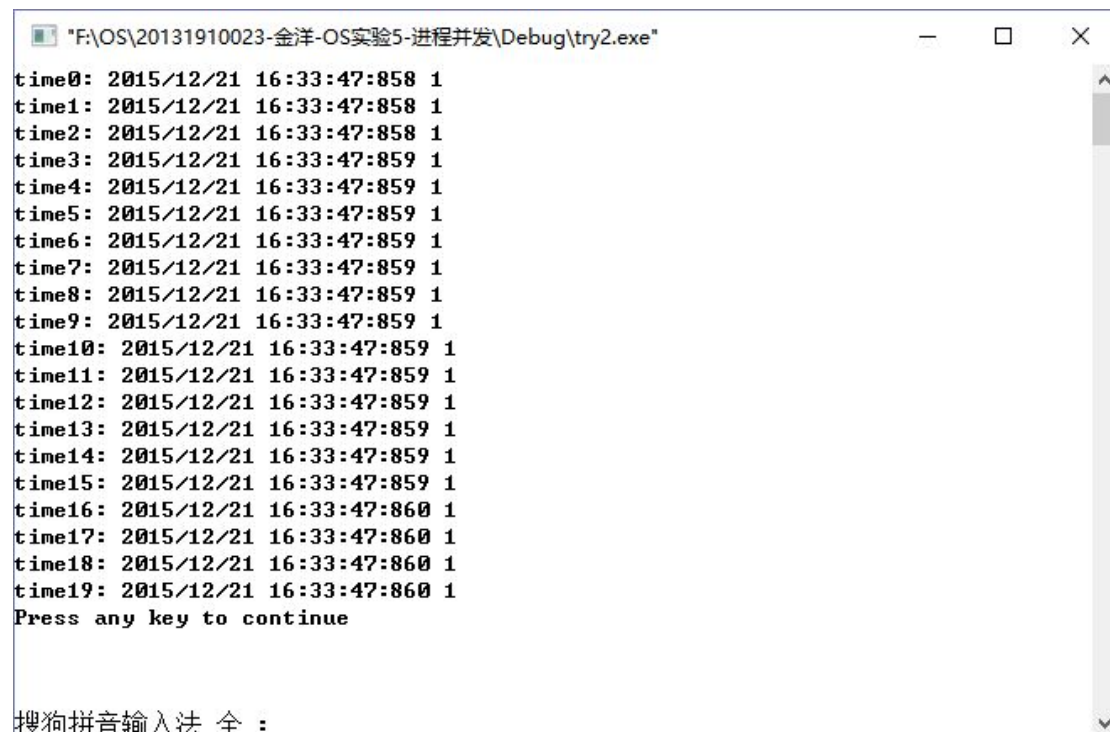
```
    ::CloseHandle(MSTThread);
```

```
}
```

2. 运行结果

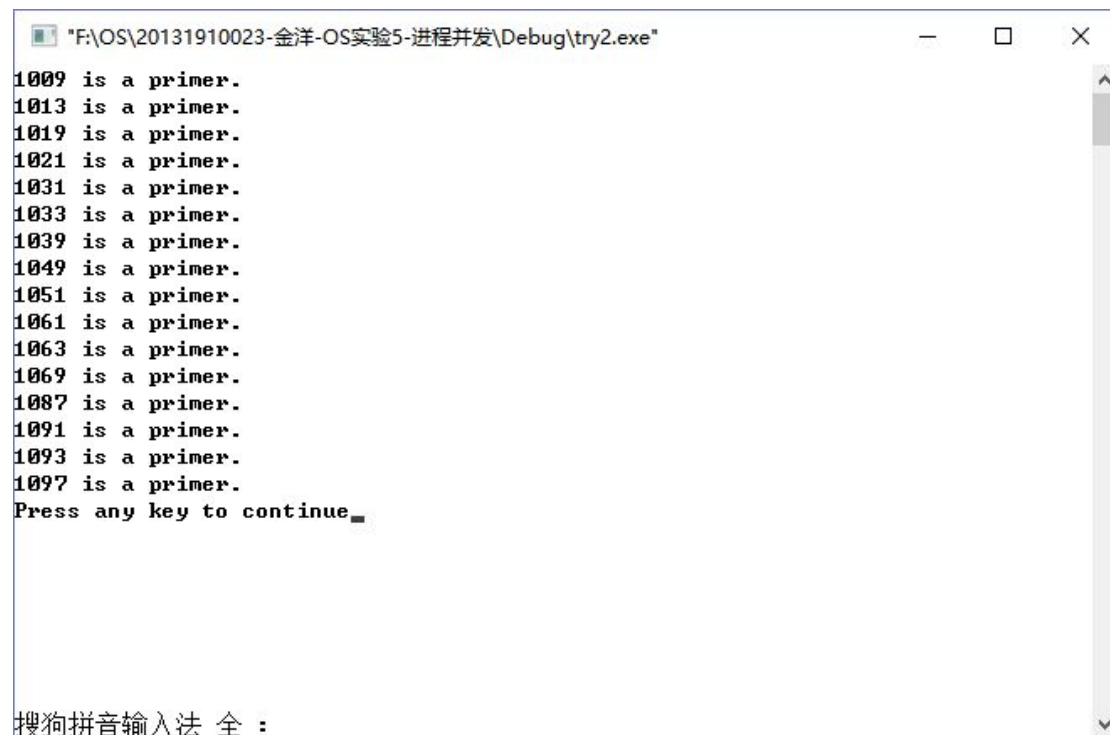
三个线程各自执行时结果如下

timeThread:




```
"F:\OS\20131910023-金洋-OS实验5-进程并发\Debug\try2.exe"
time0: 2015/12/21 16:33:47:858 1
time1: 2015/12/21 16:33:47:858 1
time2: 2015/12/21 16:33:47:858 1
time3: 2015/12/21 16:33:47:859 1
time4: 2015/12/21 16:33:47:859 1
time5: 2015/12/21 16:33:47:859 1
time6: 2015/12/21 16:33:47:859 1
time7: 2015/12/21 16:33:47:859 1
time8: 2015/12/21 16:33:47:859 1
time9: 2015/12/21 16:33:47:859 1
time10: 2015/12/21 16:33:47:859 1
time11: 2015/12/21 16:33:47:859 1
time12: 2015/12/21 16:33:47:859 1
time13: 2015/12/21 16:33:47:859 1
time14: 2015/12/21 16:33:47:859 1
time15: 2015/12/21 16:33:47:859 1
time16: 2015/12/21 16:33:47:860 1
time17: 2015/12/21 16:33:47:860 1
time18: 2015/12/21 16:33:47:860 1
time19: 2015/12/21 16:33:47:860 1
Press any key to continue
搜狗拼音输入法 全 :
```

primerThread:



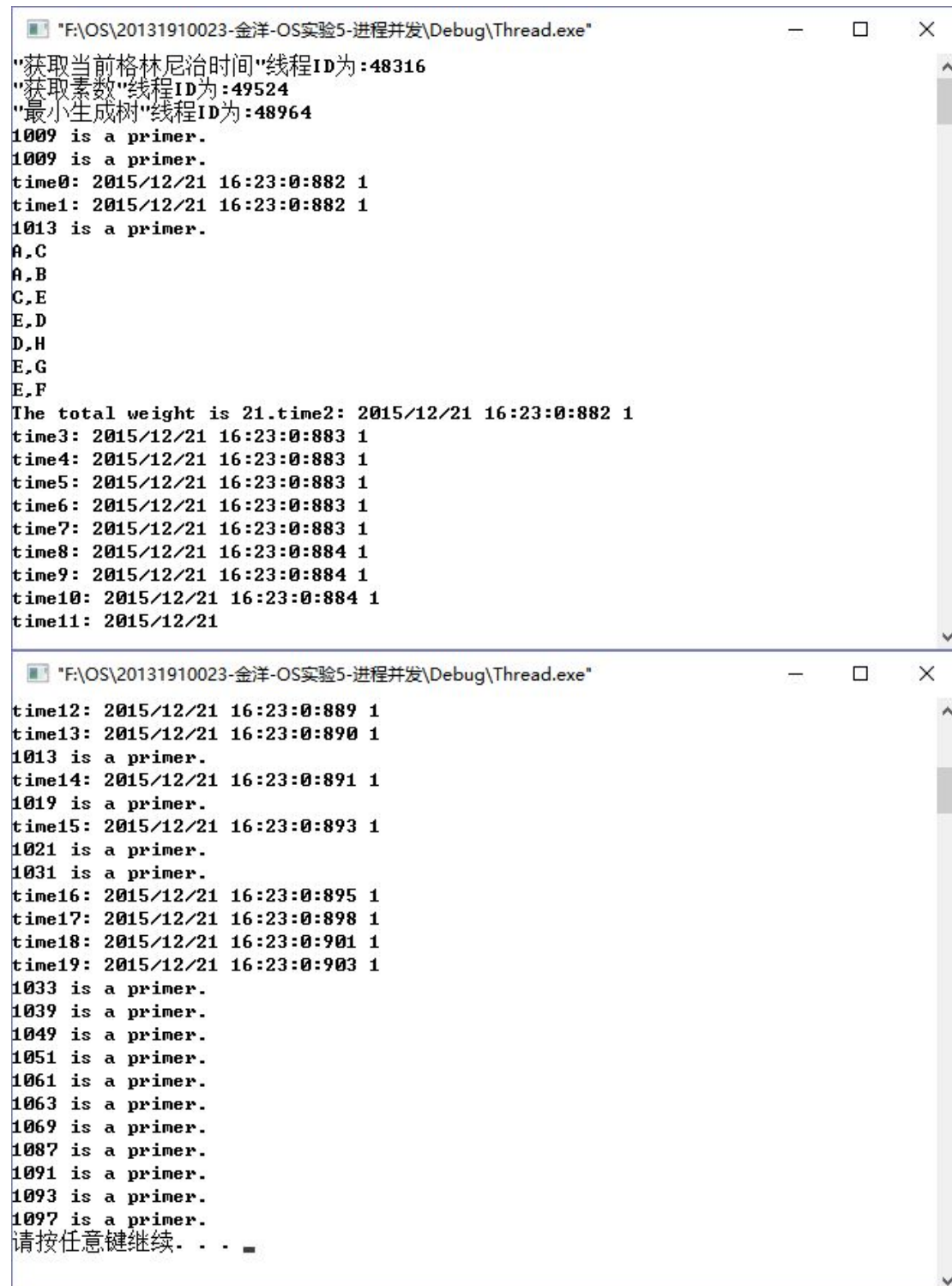
```
"F:\OS\20131910023-金洋-OS实验5-进程并发\Debug\try2.exe"
1009 is a primer.
1013 is a primer.
1019 is a primer.
1021 is a primer.
1031 is a primer.
1033 is a primer.
1039 is a primer.
1049 is a primer.
1051 is a primer.
1061 is a primer.
1063 is a primer.
1069 is a primer.
1087 is a primer.
1091 is a primer.
1093 is a primer.
1097 is a primer.
Press any key to continue
搜狗拼音输入法 全 :
```

MSTThread:



```
"F:\OS\20131910023-金洋-OS实验5-进程并发\Debug\try2.exe"  
A,C  
A,B  
C,E  
E,D  
D,H  
E,G  
E,F  
The total weight is 21.Press any key to continue_  
搜狗拼音输入法 全 :
```

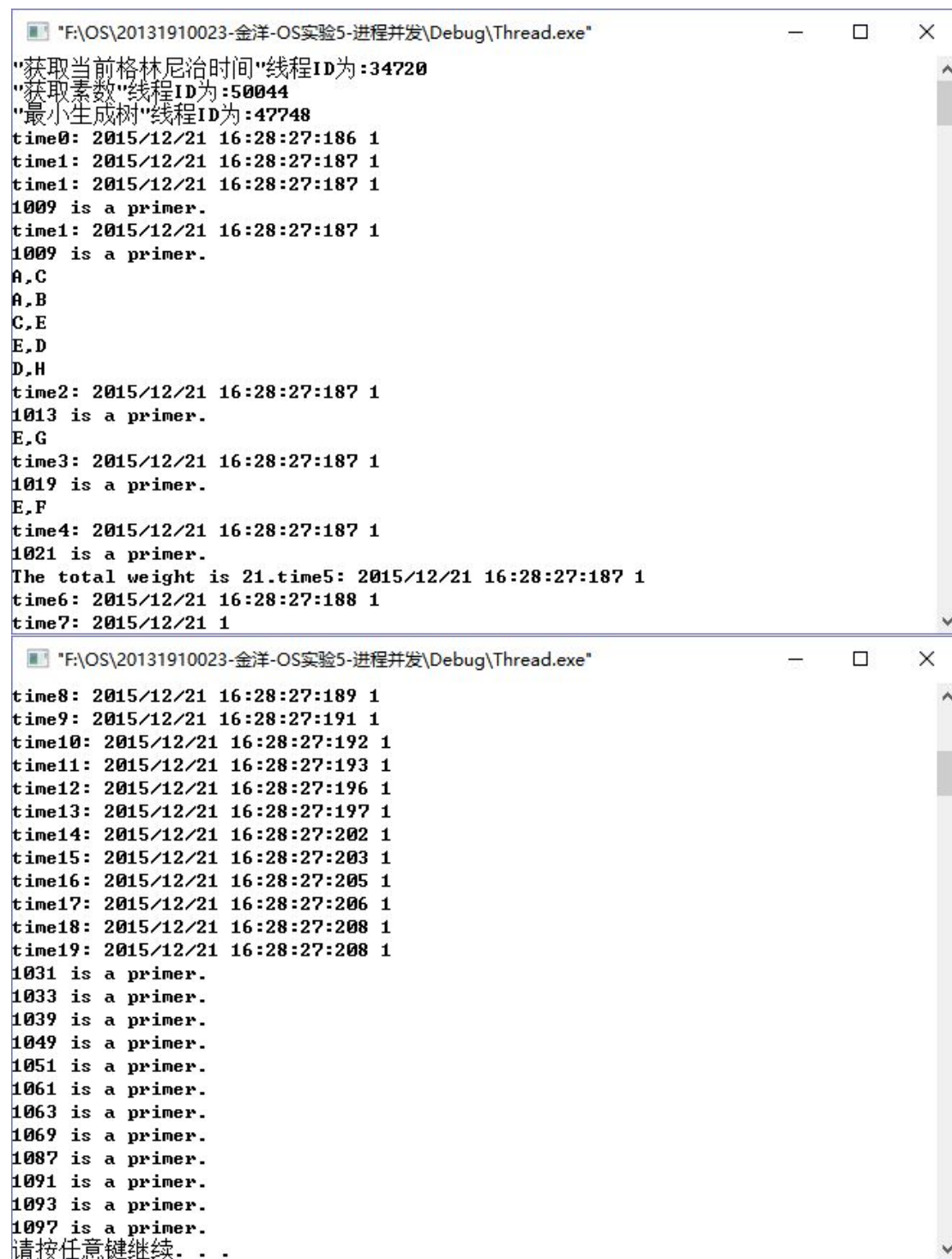
并发时运行结果①



```
"F:\OS\20131910023-金洋-OS实验5-进程并发\Debug\Thread.exe"
"获取当前格林尼治时间"线程ID为:48316
"获取素数"线程ID为:49524
"最小生成树"线程ID为:48964
1009 is a primer.
1009 is a primer.
time0: 2015/12/21 16:23:0:882 1
time1: 2015/12/21 16:23:0:882 1
1013 is a primer.
A,C
A,B
C,E
E,D
D,H
E,G
E,F
The total weight is 21.time2: 2015/12/21 16:23:0:882 1
time3: 2015/12/21 16:23:0:883 1
time4: 2015/12/21 16:23:0:883 1
time5: 2015/12/21 16:23:0:883 1
time6: 2015/12/21 16:23:0:883 1
time7: 2015/12/21 16:23:0:883 1
time8: 2015/12/21 16:23:0:884 1
time9: 2015/12/21 16:23:0:884 1
time10: 2015/12/21 16:23:0:884 1
time11: 2015/12/21

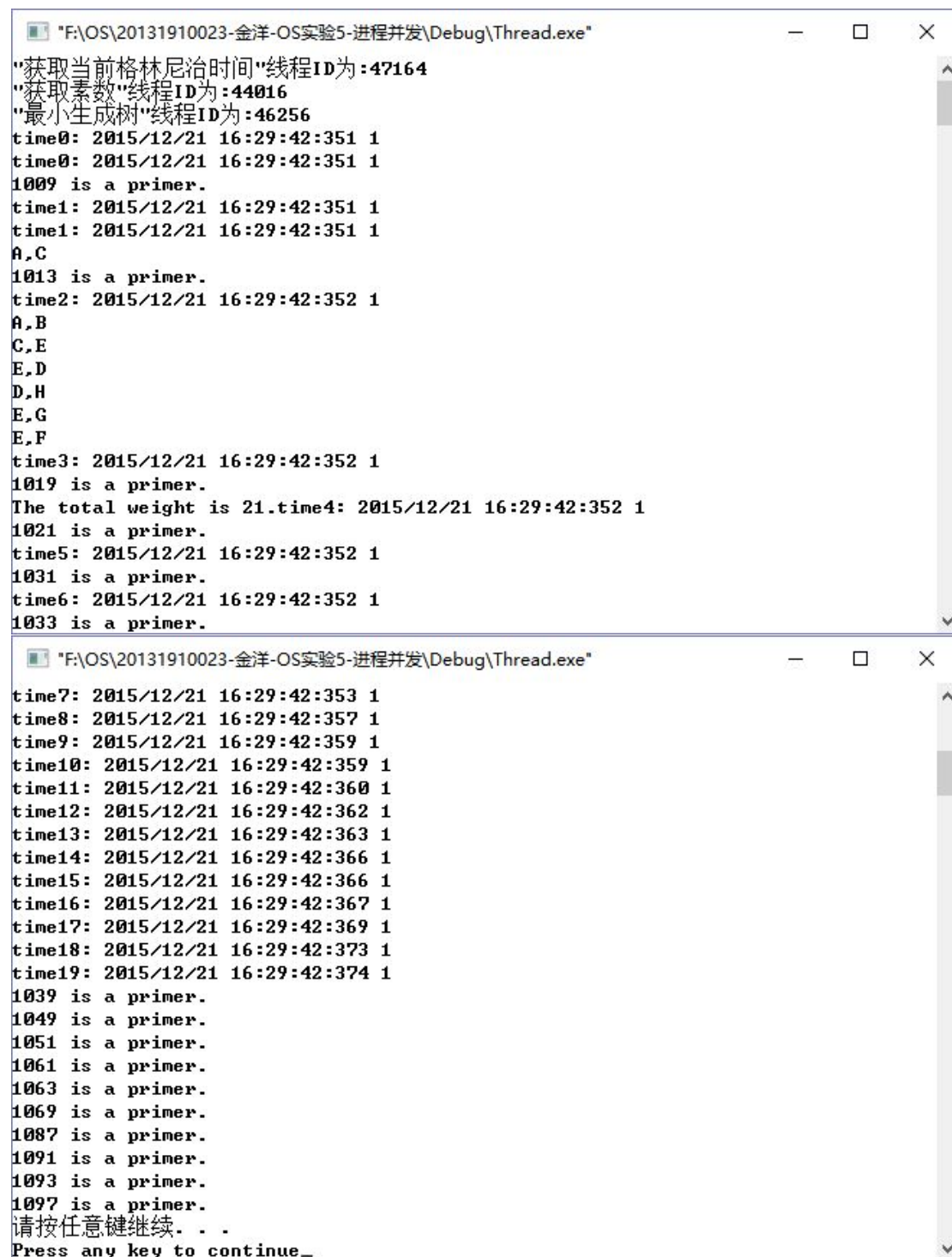
time12: 2015/12/21 16:23:0:889 1
time13: 2015/12/21 16:23:0:890 1
1013 is a primer.
time14: 2015/12/21 16:23:0:891 1
1019 is a primer.
time15: 2015/12/21 16:23:0:893 1
1021 is a primer.
1031 is a primer.
time16: 2015/12/21 16:23:0:895 1
time17: 2015/12/21 16:23:0:898 1
time18: 2015/12/21 16:23:0:901 1
time19: 2015/12/21 16:23:0:903 1
1033 is a primer.
1039 is a primer.
1049 is a primer.
1051 is a primer.
1061 is a primer.
1063 is a primer.
1069 is a primer.
1087 is a primer.
1091 is a primer.
1093 is a primer.
1097 is a primer.
请按任意键继续. . .
```

运行结果②



```
"F:\OS\20131910023-金洋-OS实验5-进程并发\Debug\Thread.exe"
"获取当前格林尼治时间"线程ID为:34720
"获取素数"线程ID为:50044
"最小生成树"线程ID为:47748
time0: 2015/12/21 16:28:27:186 1
time1: 2015/12/21 16:28:27:187 1
time1: 2015/12/21 16:28:27:187 1
1009 is a primer.
time1: 2015/12/21 16:28:27:187 1
1009 is a primer.
A,C
A,B
C,E
E,D
D,H
time2: 2015/12/21 16:28:27:187 1
1013 is a primer.
E,G
time3: 2015/12/21 16:28:27:187 1
1019 is a primer.
E,F
time4: 2015/12/21 16:28:27:187 1
1021 is a primer.
The total weight is 21.time5: 2015/12/21 16:28:27:187 1
time6: 2015/12/21 16:28:27:188 1
time7: 2015/12/21 1
time8: 2015/12/21 16:28:27:189 1
time9: 2015/12/21 16:28:27:191 1
time10: 2015/12/21 16:28:27:192 1
time11: 2015/12/21 16:28:27:193 1
time12: 2015/12/21 16:28:27:196 1
time13: 2015/12/21 16:28:27:197 1
time14: 2015/12/21 16:28:27:202 1
time15: 2015/12/21 16:28:27:203 1
time16: 2015/12/21 16:28:27:205 1
time17: 2015/12/21 16:28:27:206 1
time18: 2015/12/21 16:28:27:208 1
time19: 2015/12/21 16:28:27:208 1
1031 is a primer.
1033 is a primer.
1039 is a primer.
1049 is a primer.
1051 is a primer.
1061 is a primer.
1063 is a primer.
1069 is a primer.
1087 is a primer.
1091 is a primer.
1093 is a primer.
1097 is a primer.
请按任意键继续. . .
```

运行结果③



```
"F:\OS\20131910023-金洋-OS实验5-进程并发\Debug\Thread.exe"
"获取当前格林尼治时间"线程ID为:47164
"获取素数"线程ID为:44016
"最小生成树"线程ID为:46256
time0: 2015/12/21 16:29:42:351 1
time0: 2015/12/21 16:29:42:351 1
1009 is a primer.
time1: 2015/12/21 16:29:42:351 1
time1: 2015/12/21 16:29:42:351 1
A,C
1013 is a primer.
time2: 2015/12/21 16:29:42:352 1
A,B
C,E
E,D
D,H
E,G
E,F
time3: 2015/12/21 16:29:42:352 1
1019 is a primer.
The total weight is 21.time4: 2015/12/21 16:29:42:352 1
1021 is a primer.
time5: 2015/12/21 16:29:42:352 1
1031 is a primer.
time6: 2015/12/21 16:29:42:352 1
1033 is a primer.

time7: 2015/12/21 16:29:42:353 1
time8: 2015/12/21 16:29:42:357 1
time9: 2015/12/21 16:29:42:359 1
time10: 2015/12/21 16:29:42:359 1
time11: 2015/12/21 16:29:42:360 1
time12: 2015/12/21 16:29:42:362 1
time13: 2015/12/21 16:29:42:363 1
time14: 2015/12/21 16:29:42:366 1
time15: 2015/12/21 16:29:42:366 1
time16: 2015/12/21 16:29:42:367 1
time17: 2015/12/21 16:29:42:369 1
time18: 2015/12/21 16:29:42:373 1
time19: 2015/12/21 16:29:42:374 1
1039 is a primer.
1049 is a primer.
1051 is a primer.
1061 is a primer.
1063 is a primer.
1069 is a primer.
1087 is a primer.
1091 is a primer.
1093 is a primer.
1097 is a primer.
请按任意键继续. . .
Press any key to continue
```

六、总结

1. 句柄，是整个 Windows 编程的基础。一个句柄是指使用的一个唯一的整数值，即一个 4 字节 (64 位程序中为 8 字节) 长的数值，来标识应用程序中的不同对象和同类中的不同的实例，诸如，一个窗口，按钮，图标，滚动条，输出设备，控

件或者文件等。应用程序能够通过句柄访问相应的对象的信息，但是句柄不是一个指针，程序不能利用句柄来直接阅读文件中的信息。

windows 之所以要设立句柄，根本上源于内存管理机制的问题——虚拟地址，简而言之数据的地址需要变动，变动以后就需要有人来记录管理变动，（就好像户籍管理一样），因此系统用句柄来记载数据地址的变更。

数据对象加载进入内存中之后即获得了地址，但是这个地址并不是固定的，数据对象会根据需要在内存与硬盘之间游弋移动（例如不常用的数据会为常用数据让出其占用的内存空间进而被淘汰进硬盘中的虚拟内存之中以优化配置整体系统的资源进而提升效率性能），因此其物理地址总是变动的，那么作为管理者则必须对 管理对象所发生的变化了如指掌才行，因此系统为进程分配固定的地址（句柄）来存储进程下的数据对象变化后的地址也就是当前的地址，其实设计机制很简单：系统的某个部门移动了对象的地址后，同时上报给句柄所属部门管理者，管理者将改动写入句柄即可。该数据被重新起用时去其所属句柄内按内容存取即可；

2. 创建线程：创建一个线程可以有两种方法：一种是使用 Win32API；一种是使用 MFC。Win32API 编程创建线程，使用的函数是 `CreateThread()`。

线程终止可以采用两种方式。一种方式是当一个工作者线程的线程函数执行一个返回语句或者调用 `AfxEndThread()` 时，这个工作者线程就终止。另一种方式适合于 UI 线程，当消息 `WM_QUIT` 被发送到线程的消息队列，或者 `AfxEndThread()` 函数被调用时，线程结束；

3 假设 `h1` 是线程，则恢复 `h1` 为 `ResumeThread(h1)`、挂起 `h1` 为 `SuspendThread(h1)`、终止 `h1` 为 `TerminateThread(h1,1)`；

4. `::` 是运算符中等级最高的，它分为三种：全局作用域符，类作用域符，命名空间作用域符。他们的作用都是为了更明确的调用自己的对象或者函数；

5. `LPVOID` 是一个没有类型的指针，也就是说可以将任意类型的指针赋值给 `LPVOID` 类型的变量（一般作为参数传递），然后在使用的時候在转换回来；

6. `system("pause")` 就是从程序里调用 “pause” 命令；“pause” 这个系统命令的功能很简单，就是在命令行上输出一行类似于 “Press any key to exit” 的字，等待用户按一个键，然后返回；

7. 在 `CreateThread` 中的第五个参数为 `CREATE_SUSPENDED` 表示将创建好的进程挂起，若为 0 则创建好后即可执行；

8. 通过对运行结果的分析，三个线程并发执行的结果与三线程各自独立执行的结果有了很大区别，前者显示的结果呈现出 “杂乱性”，如同后者结果的混合。

9. 程序并发执行时具有间断性、失去封闭性、不可再现性。在这里间断性和不可再现性很明显。结果的显示中并不是把每一个进程依次执行完，而是交叉着进行，处理机在被一线程占用，则其他两者必须先等待——间断性；而很容易看出三次

结果都不相同——不可再现性；

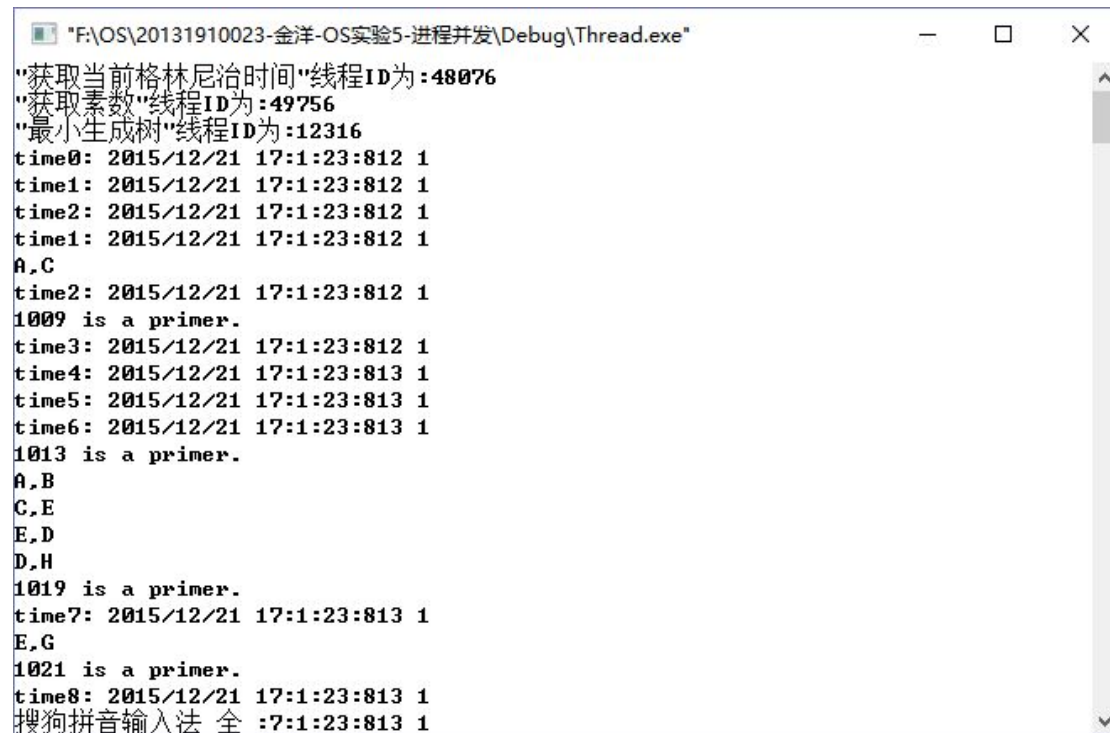
10. 在程序的结果中，发现有些结果被重复显示，如：

```
time0: 2015/12/21 16:29:42:351 1
time0: 2015/12/21 16:29:42:351 1
1009 is a primer.
time1: 2015/12/21 16:29:42:351 1
time1: 2015/12/21 16:29:42:351 1

time1: 2015/12/21 16:28:27:187 1
time1: 2015/12/21 16:28:27:187 1
1009 is a primer.
time1: 2015/12/21 16:28:27:187 1
1009 is a primer.

1009 is a primer.
1009 is a primer.
```

猜测为三个线程实体中循环变量名称相同所致，修改循环变量名后，情况依旧：



```
"F:\OS\20131910023-金洋-OS实验5-进程并发\Debug\Thread.exe"
"获取当前格林尼治时间"线程ID为:48076
"获取素数"线程ID为:49756
"最小生成树"线程ID为:12316
time0: 2015/12/21 17:1:23:812 1
time1: 2015/12/21 17:1:23:812 1
time2: 2015/12/21 17:1:23:812 1
time1: 2015/12/21 17:1:23:812 1
A,C
time2: 2015/12/21 17:1:23:812 1
1009 is a primer.
time3: 2015/12/21 17:1:23:812 1
time4: 2015/12/21 17:1:23:813 1
time5: 2015/12/21 17:1:23:813 1
time6: 2015/12/21 17:1:23:813 1
1013 is a primer.
A,B
C,E
E,D
D,H
1019 is a primer.
time7: 2015/12/21 17:1:23:813 1
E,G
1021 is a primer.
time8: 2015/12/21 17:1:23:813 1
搜狗拼音输入法 全 :7:1:23:813 1
```



```
"F:\OS\20131910023-金洋-OS实验5-进程并发\Debug\Thread.exe"
"获取当前格林尼治时间"线程ID为:46108
"获取素数"线程ID为:50784
"最小生成树"线程ID为:50528
time0: 2015/12/21 17:2:42:431 1
time0: 2015/12/21 17:2:42:431 1
1009 is a primer.
time1: 2015/12/21 17:2:42:431 1
time1: 2015/12/21 17:2:42:431 1
A,C
A,B
time2: 2015/12/21 17:2:42:431 1
1013 is a primer.
1019 is a primer.
1021 is a primer.
1031 is a primer.
1033 is a primer.
C,E
E,D
D,H
E,G
E,F
The total weight is 21.1039 is a primer.
1049 is a primer.
1051 is a primer.
1061 is a primer.
```

猜测是有关内存空间的互斥使用存在问题?

11.在一开始写完程序运行时,经常出现某一线程实体的内容还未运行完,而整个程序已经结束,如:

```
"F:\OS\20131910023-金洋-OS实验5-进程并发\Debug\Thread.exe"
A,B
time2: 2015/12/21 17:2:42:431 1
1013 is a primer.
1019 is a primer.
1021 is a primer.
1031 is a primer.
1033 is a primer.
C,E
E,D
D,H
E,G
E,F
The total weight is 21.1039 is a primer.
1049 is a primer.
1051 is a primer.
1061 is a primer.
1063 is a primer.
1069 is a primer.
Press any key to continue
```

```
"F:\OS\20131910023-金洋-OS实验5-进程并发\Debug\Thread.exe"
"获取当前格林尼治时间"线程ID为:28776
"获取素数"线程ID为:22304
"最小生成树"线程ID为:48372
1009 is a primer.
1013 is a primer.
1013 is a primer.
A,C
A,B
C,E
E,D
D,H
E,G
E,F
The total weight is 21.1019 is a primer.
time0: 2015/12/21 17:7:16:857 1
1019 is a primer.
time1: 2015/12/21 17:7:16:857 1
Press any key to continue_
```

搜狗拼音输入法 全 :

```
"F:\OS\20131910023-金洋-OS实验5-进程并发\Debug\Thread.exe"
"获取当前格林尼治时间"线程ID为:49756
"获取素数"线程ID为:37616
"最小生成树"线程ID为:47340
1009 is a primer.
1009 is a primer.
A,C
A,B
C,E
E,D
D,H
1013 is a primer.
1013 is a primer.
time0: 2015/12/21 17:9:0:986 1
E,G
E,F
The total weight is 21.1019 is a primer.
Press any key to continue_
```

搜狗拼音输入法 全 :

观察每个结果，未完成的都是时间线程和素数线程，而 MST 每次都已经运行完，猜测是 MST 的执行完毕影响了其他两者。在 CloseHandle 之前加上语句 system("pause") 是系统暂停等待用户的输入任意键，此问题不再有。

七、参考文献

- [1]汤小丹，梁红兵，哲凤屏，汤子瀛. 计算机操作系统[M]（第三版）. 西安：西安电子科技大学出版社，2007 年 5 月；
- [2]谭浩强著. c 程序设计[M]（第三版）. 北京：清华大学出版社. 2005. 7；

八、教师评语