实验5——线程的创建与调度及竞争范围

1.实验内容

学会使用pthread_attr_init()初始化线程参数。学会使用pthread_create()函数创建线程，并使用其参数指定线程执行的函数。学会使用pthread_join()函数调整线程的执行次序，测验线程的调度。

了解线程竞争范围的基本知识。学会使用pthread_attr_getscope函数获取当前线程的竞争范围；学会使用pthread_attr_setscope函数尝试更改线程的竞争范围。

- 任务一：

```
void *runner01(void *param)
{
    int i, upper = atoi(param);
    for(i=1; i<=upper; i++) sum += i;
    printf("[thread 1] sum = %d\n", sum);
}
```

```
void *runner02(void *param)
{
    int i, lower = atoi(param);
    for(i=1; i<=lower; i++) sum -= i;
    printf("[thread 2] sum = %d\n", sum);
}
```

创建两个线程，线程1执行runner01函数，线程2执行runner02函数，并且要求运行参数采用命令行参数的方式从命令中获取。

仿照实验2的test脚本，编写脚本使程序执行100次，观察运行结果。

- 任务二：

将程序改写为线程1先运行，线程1运行结束后线程2才能运行。编写脚本使程序执行100次，观察运行结果。

针对于本实验中任务一、二出现的结果的区别请在实验报告中予以解释和说明。

- 任务三：

了解线程竞争范围的相关概念。线程一共有两种竞争范围，分别是进程竞争范围和系统竞争范围。实验探究本机线程之间的竞争范围。

首先使用pthread_attr_getscope函数获取当前线程的竞争范围，随后使用pthread_attr_setscope函数尝试更改线程的竞争范围，尝试更改后再次查看当前线程的竞争范围。根据实验结果得出当前机器支持的线程竞争范围是什么。

2.实验示例代码

任务一、二

- 需要用到的头文件

```
#include<pthread.h>
#include<stdio.h>
#include<stdlib.h>
```

- 创建一个线程

```
pthread_t tid01;
pthread_attr_t attr;
pthread_attr_init(&attr);// get default setting
pthread_create(&tid01,&attr,runner01,argv[1]);
pthread_exit(0);
```

- 命令行参数的获取

```
int main(int argc, char* argv[])
{
    if(argc != 2){
        fprintf(stderr,"usage:a.out<integer value>\n");
        return -1;
    }
    if(atoi(argv[1]) <= 0){
        fprintf(stderr,"%d must be > 0\n",atoi(argv[1]));
                return -1;
    }
}
```

任务三

- 需要用到的头文件

```
#include <pthread.h>
#include <stdio.h>
```

- 获取一个线程的竞争范围

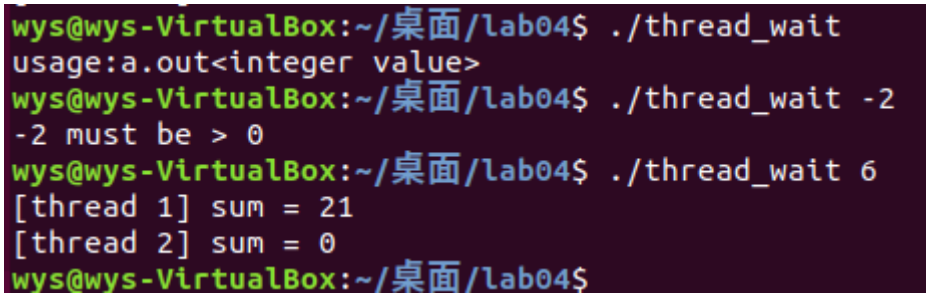
```
int scope;
pthread_attr_t attr;
/* get the default attributes*/
pthread_attr_init(&attr);
/* first inquire on the current scope*/
if(pthread_attr_getscope(&attr,&scope)!=0)
    fprintf(stderr,"Unable to get scheduling scope\n");
else{
    if(scope == PTHREAD_SCOPE_PROCESS)
        printf("PTHREAD_SCOPE_PROCESS\n");
    else if(scope == PTHREAD_SCOPE_SYSTEM)
        printf("PTHREAD_SCOPE_SYSTEM\n");
    else
        fprintf(stderr,"Illegal scope value.\n");
}
```

- 设置线程的竞争范围

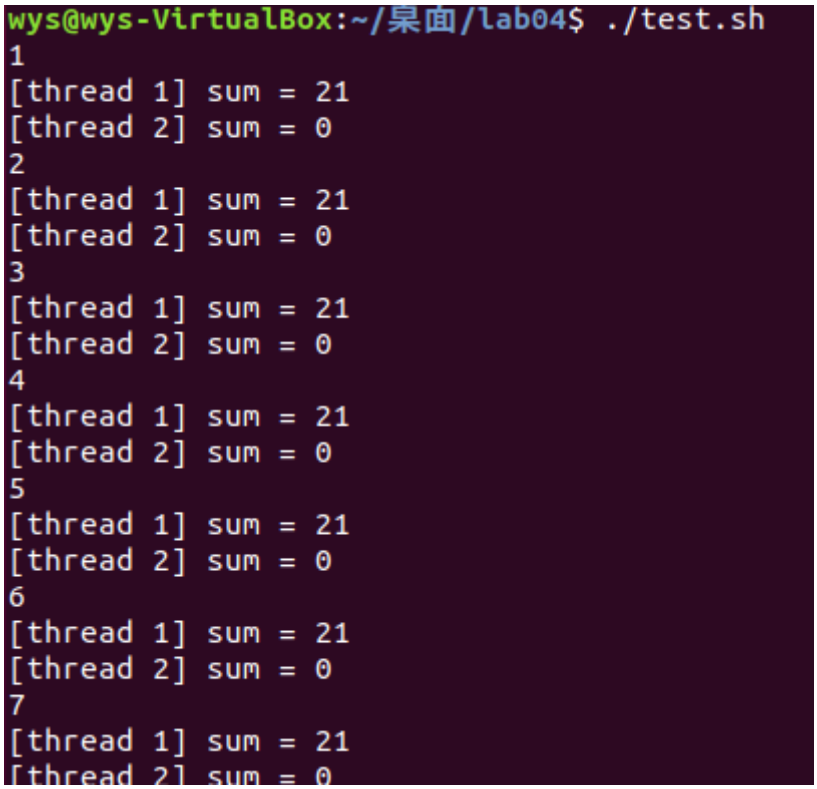
```
pthread_attr_setscope(&attr, PTHREAD_SCOPE_SYSTEM);
pthread_attr_setscope(&attr, PTHREAD_SCOPE_PROCESS);
```

3.实验结果示例

注意，本实验中用到了pthread库，需要在编译时指明链接库，例如“gcc -o xxx xxx.c -lpthread”



任务一、二运行示例



任务二示例

```
[thread 2] sum = 0
57
[thread 1] sum = 21
[thread 2] sum = 0
58
[thread 1] sum = 21
[thread 2] sum = 0
59
[thread 2] sum = -21
[thread 1] sum = 0
60
[thread 2] sum = 0
[thread 1] sum = 21
61
[thread 2] sum = -21
[thread 1] sum = 0
62
[thread 1] sum = 21
[thread 2] sum = 0
63
[thread 1] sum = 21
[thread 2] sum = 0
64
```

任务一示例

```
wys@wys-VirtualBox:~/桌面/lab05$ ./exam5_1
PTHREAD_SCOPE_SYSTEM
PTHREAD_SCOPE_SYSTEM
wys@wys-VirtualBox:~/桌面/lab05$
```

任务三示例