# BARRIERS作业报告

软工2班 3017218092 侯雨茜

## 一、准备工作

下载barrier.c并在电脑上进行编译：

```
(base) houyuqiandeMacBook-Pro:barriers yuqianhou$ gcc -g -O2 -pthread barrier.c
barrier.c:46:1: warning: control may reach end of non-void function
      [-Wreturn-type]
}
^
1 warning generated.
(base) houyuqiandeMacBook-Pro:barriers yuqianhou$ ./a/out 2
-bash: ./a/out: No such file or directory
(base) houyuqiandeMacBook-Pro:barriers yuqianhou$ ./a.out 2
Assertion failed: (i == t), function thread, file barrier.c, line 42.
Abort trap: 6
```

2指定在barrier上同步的线程数（ `barrier.c` 中的 `nthread` ）。每个线程都处于紧密的循环中。在每个循环迭代中，线程都会调用 `barrier()` ，然后在随机的微秒数内休眠。断言触发，因为一个线程在另一线程到达屏障之前就离开了屏障。理想的行为是所有线程都应阻塞，直到 `nthread` 调用 `barrier` 为止。

目标是实现所需的行为。除了以前看到的锁原语之外，您还将需要以下新的pthread原语：

```
1 pthread_cond_wait(&cond, &mutex);  // go to sleep on cond, releasing
  lock mutex
2 pthread_cond_broadcast(&cond);     // wake up every thread sleeping on
  cond
3 pthread_cond_wait releases the mutex when called, and re-acquires the
  mutex before returning.
```

## 二、实现barrier()

```
1 static void
2 barrier()
3 {
```

```
    pthread_mutex_lock(&bstate.barrier_mutex);
    bstate.nthread++;
    printf("in round %d as %d\n", bstate.round, bstate.nthread);
    if (bstate.nthread != nthread)
    {
        pthread_cond_wait(&bstate.barrier_cond,
&bstate.barrier_mutex);
    }
    else
    {
        bstate.round++;
        bstate.nthread = 0;
        pthread_cond_broadcast(&bstate.barrier_cond);
    }
    pthread_mutex_unlock(&bstate.barrier_mutex);
}
```

## 三、测试结果

使用一个、两个和多个线程测试代码。

### 1个线程

```
(base) houyuqiandeMacBook-Pro:barriers yuqianhou$ gcc -g -O2 -pthread barrier.c
barrier.c:59:1: warning: control may reach end of non-void function
      [-Wreturn-type]
}
^
1 warning generated.
(base) houyuqiandeMacBook-Pro:barriers yuqianhou$ ./a.out 1
in round 0 as 1
in round 1 as 1
in round 2 as 1
in round 3 as 1
in round 4 as 1
in round 5 as 1
in round 6 as 1
```

……

```
in round 19995 as 1
in round 19996 as 1
in round 19997 as 1
in round 19998 as 1
in round 19999 as 1
OK; passed
```

2个线程

```
(base) houyuqiandeMacBook-Pro:barriers yuqianhou$ ./a.out 2
in round 0 as 1
in round 0 as 2
in round 1 as 1
in round 1 as 2
in round 2 as 1
in round 2 as 2
in round 3 as 1
in round 3 as 2
in round 4 as 1
in round 4 as 2
in round 5 as 1
in round 5 as 2
in round 6 as 1
```

……

```
in round 19995 as 1
in round 19995 as 2
in round 19996 as 1
in round 19996 as 2
in round 19997 as 1
in round 19997 as 2
in round 19998 as 1
in round 19998 as 2
in round 19999 as 1
in round 19999 as 2
OK; passed
```

3个线程

```
(base) houyuqiandeMacBook-Pro:barriers yuqianhou$ ./a.out 3
in round 0 as 1
in round 0 as 2
in round 0 as 3
in round 1 as 1
in round 1 as 2
in round 1 as 3
in round 2 as 1
in round 2 as 2
in round 2 as 3
in round 3 as 1
in round 3 as 2
in round 3 as 3
in round 4 as 1
```

……

```
in round 19995 as 3
in round 19996 as 1
in round 19996 as 2
in round 19996 as 3
in round 19997 as 1
in round 19997 as 2
in round 19997 as 3
in round 19998 as 1
in round 19998 as 2
in round 19998 as 3
in round 19999 as 1
in round 19999 as 2
in round 19999 as 3
OK; passed
```