

一个关于 test 分析的样例：

首先 test 在哪里？src\tests\threads

然后，见名知意

这里分析最简单的一个作为你们的参考：alarm-single

这个 test 不是单独存在的，它在 alarm-wait 里

首先看代码：

```
void
test_alarm_single (void)
{
    test_sleep (5, 1);
}
```

那么 test_sleep()是什么呢？

```
static void
test_sleep (int thread_cnt, int iterations)
```

代码太长，自己去看

创建 5 个子线程 Thread0-Thread5。线程运行函数为 sleeper，休眠时间是 10-50。每个 sleeper 线程都会迭代 iteration（1）次，以 Thread0 为例，每一次迭代过程中(这里面只有 1 次)，线程休眠 10 个 ticks (ticks 是 pintos 的计时单位；休眠的线程会进入 block 状态，退出 CPU)，每次休眠结束后被重新唤醒，会将当前的时间和上次记录的记录的时间之差写进 output 数组。然后从 CPU 退回到 ready 队列。这两个时间理论上需要相等。接下来的线程以此类推，

运行结果：可以看到 duration==product

```
Boot complete.
Executing 'alarm-single':
(alarm-single) begin
(alarm-single) Creating 5 threads to sleep 1 times each.
(alarm-single) Thread 0 sleeps 10 ticks each time,
(alarm-single) thread 1 sleeps 20 ticks each time, and so on.
(alarm-single) If successful, product of iteration count and
(alarm-single) sleep duration will appear in nondescending order.
(alarm-single) thread 0: duration=10, iteration=1, product=10
(alarm-single) thread 1: duration=20, iteration=1, product=20
(alarm-single) thread 2: duration=30, iteration=1, product=30
(alarm-single) thread 3: duration=40, iteration=1, product=40
(alarm-single) thread 4: duration=50, iteration=1, product=50
(alarm-single) end
Execution of 'alarm-single' complete.
```

写 test 分析的时候，要一步步分析，每一步在干什么，理论希望输出什么样的值，这些都要写。