

Homework: Threads and Locking

刘恒星 2022229044

March 30, 2023

1 实验

这次实验目的主要是了解多线程和临界区处理。这次的代码是实验提供的，如果我们直接运行，会看到如下结果。

```
1: put time = 0.003338
0: put time = 0.003389
0: get time = 7.684335
0: 17480 keys missing
1: get time = 7.684335
1: 17480 keys missing
completion time = 7.687856
```

但是单线程运行会出现一下结果

```
$ ./a.out 1
0: put time = 0.004073
0: get time = 6.929189
0: 0 keys missing
completion time = 6.933433
```

在多线程运行下，会出现 key 丢失的情况，我们分析代码，每一个线程执行以下代码

```
1 static void
2 insert(int key, int value, struct entry **p, struct entry *n)
3 {
4     struct entry *e = malloc(sizeof(struct entry));
5     e->key = key;
6     e->value = value;
7     e->next = n;
8     *p = e;
9 }
10
11 static void put(int key, int value)
12 {
13     int i = key % NBUCKET;
14     pthread_mutex_lock(&mutex);
15     insert(key, value, &table[i], table[i]);
16     pthread_mutex_unlock(&mutex);
17 }
18
19 static struct entry *
20 get(int key)
21 {
22     struct entry *e = 0;
23     for (e = table[key % NBUCKET]; e != 0; e = e->next)
24     {
25         if (e->key == key)
26             break;
27     }
28     return e;
29 }
30
31 static void *
32 thread(void *xa)
33 {
```

```

34     long n = (long)xa;
35     int i;
36     int b = NKEYS / nthread;
37     int k = 0;
38     double t1, t0;
39
40     // printf("b = %d\n", b);
41     t0 = now();
42     for (i = 0; i < b; i++)
43     {
44         // printf("%d: put %d\n", n, b*n+i);
45         put(keys[b * n + i], n);
46     }
47     t1 = now();
48     printf("%ld: put time = %f\n", n, t1 - t0);
49
50     // Should use pthread_barrier, but MacOS doesn't support it ...
51     __sync_fetch_and_add(&done, 1);
52     while (done < nthread)
53         ;
54
55     t0 = now();
56     for (i = 0; i < NKEYS; i++)
57     {
58         struct entry *e = get(keys[i]);
59         if (e == 0)
60             k++;
61     }
62     t1 = now();
63     printf("%ld: get time = %f\n", n, t1 - t0);
64     printf("%ld: %d keys missing\n", n, k);
65     return NULL;
66 }

```

导致 key 丢失的原因就是 *get()* 函数找不到对应的 key, 而 *get()* 找不到对应的 key 的原因是因为不同线程在调用 *put()* 的时候没有做好同步而修改其他线程的信息导致的, 解决这个问题我们就要考虑在 *get()* 和 *put()* 做修改, 考虑到只有 *put()* 会对 *table[]* 做修改, 所以在 *put()* 代码中添加同步锁

```

1 static void put(int key, int value)
2 {
3     int i = key % NBUCKET;
4     pthread_mutex_lock(&mutex);
5     insert(key, value, &table[i], table[i]);
6     pthread_mutex_unlock(&mutex);
7 }

```

效果如下:

```
xteam@xtaem:~/Calculus/os/xv6-public$ ./a.out 2
0: put time = 0.019085
1: put time = 0.019160
0: get time = 5.152239
0: 0 keys missing
1: get time = 5.395675
1: 0 keys missing
completion time = 5.415034
```