

Lab 实验报告 VII

Lab: Multithreading

丁睿

dromniscience@gmail.com

更新: 2020 年 12 月 5 日

目录

1	任务完成清单	2
2	详细情况 & 困难和收获	2
2.1	Uthread: switching between threads	2
2.2	Using threads	2
2.3	Barrier	3
3	参考资料	3

1 任务完成清单

Subtask	Done?	Time
Uthread	Y	1h
Using threads	Y	20min
Barrier	Y	15min

Grade: 60/60

我的作业已开放在[这个网页](#)上。

2 详细情况 & 困难和收获

本次 lab 难度不大，代码量不多，调试以来非常轻松。由于实际上并未在 RISC-V xv6 里实现支持多处理器的进程-线程模型，所以对于原本的 xv6 的改动非常小。

2.1 Uthread: switching between threads

仅需要保存被调用者保存寄存器，因为余下的寄存器在调用本函数前已被调用者保存在栈中。于是可以照搬 `struct context` 的内容，而且上下文切换就如同 `swtch.S` 里那样操作 (其实也只有偏移的问题)。

注意到用户线程的函数原型总是 `void (*)(void)`，所以初始化时除了返回地址和栈顶位置，其余 `struct context` 中的字段都可以初始化为 0。返回地址将设成函数的起始地址，而栈将设成 `struct thread` 的 `stack` 数组的最高位¹。这样，一旦被调度，就会使用新的栈和代码。而此后，线程就如进程切换那样保存和恢复上下文。

源码使用的一个 trick 是总是用 `RUNNING` 标记一个执行完的线程或者不希望被调度的线程。例如 `thread_a`, `thread_b`, `thread_c` 执行完后仍然被标记，所以不会被调度，而 `main` 从不调用 `pthread_yield`，因此在 `pthread_schedule` 后再也不被调度。另外，`uthread` 仍然运行在单核上，没有实现真正的并行效果。

2.2 Using threads

这明显是一个开散列结构。每个桶指向一个链表头，里面保存了所有哈希值相同的元素。为了 `put` 和 `get` 能够以任意顺序被交错地调用，我们需要给每个链表一把锁。

注意不同的链表是不存在竞争问题的。而在同一个链表内，则应当在线程访问链表的整个过程中加锁。例如 `put`，如果在查找到相应节点和调用 `insert` 之间有窗口，那么可能出现重复插

¹因为栈总是向下增长，我第一次实现时就忽略了这个问题。

入一个键值的错误。而`get`是不需要保护的。因为链表头总是指向一个有效的后继链表，尽管可能不是最新的，但这对`get`的访问没有任何影响。

2.3 Barrier

条件变量这种机制类似 RISC-V xv6 的 `sleep/wakeup` 语义，同时也符合管程 (monitor) 里条件变量的语义。

我们应当保护对共享全局变量 `bstate` 的修改和访问，因此 `bstate.barrier_mutex` 专门作为它的互斥锁。其中对调用 `barrier` 的次数加加，并检查结果是否到 `nthread`。如果没有，线程休眠并释放对 `bstate` 的访问权。否则，应当唤醒睡在条件变量上的线程。由于该线程一直持有锁，所以在唤醒前或者唤醒后更新 `bstate.round` 都是可以的，只要在持有锁的区域内且由唤醒休眠线程的那个线程做。

代码附在下方:

```
1 static void
2 barrier()
3 {
4     pthread_mutex_lock(&bstate.barrier_mutex);
5     bstate.nthread += 1;
6     if(bstate.nthread == nthread){
7         bstate.nthread = 0;
8         bstate.round += 1;
9         pthread_cond_broadcast(&bstate.barrier_cond);
10    }
11    else{
12        pthread_cond_wait(&bstate.barrier_cond, &bstate.
13                           barrier_mutex);
14    }
15    pthread_mutex_unlock(&bstate.barrier_mutex);
16 }
```

3 参考资料

1. *xv6: a simple, Unix-like teaching operating system*
Russ Cox, Frans Kaashoek, Robert Morris August 31, 2020
2. *The RISC-V Reader: An Open Architecture Atlas*
David Patterson, Andrew Waterman 1st Edition
3. 现代操作系统 [M]
A.S.Tanenbaum, H. Bos 著, 陈向群 马洪兵 译, 北京: 机械工业出版社, 2011: 47-95