Preemption

A1

学　　号：	**12210158**
姓　　名：	吴同馨

目 录

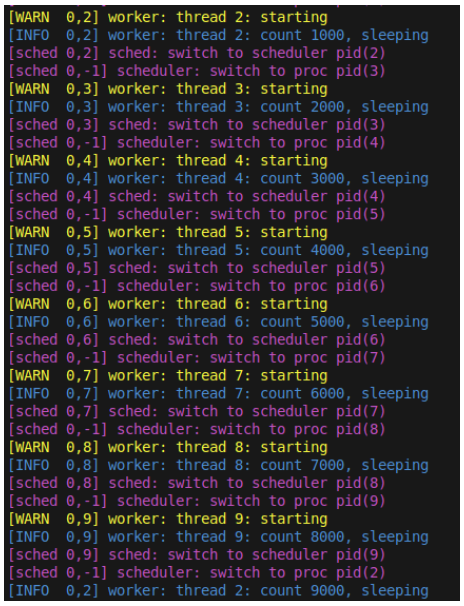
第1章 Checkpoint1

1.1 Code

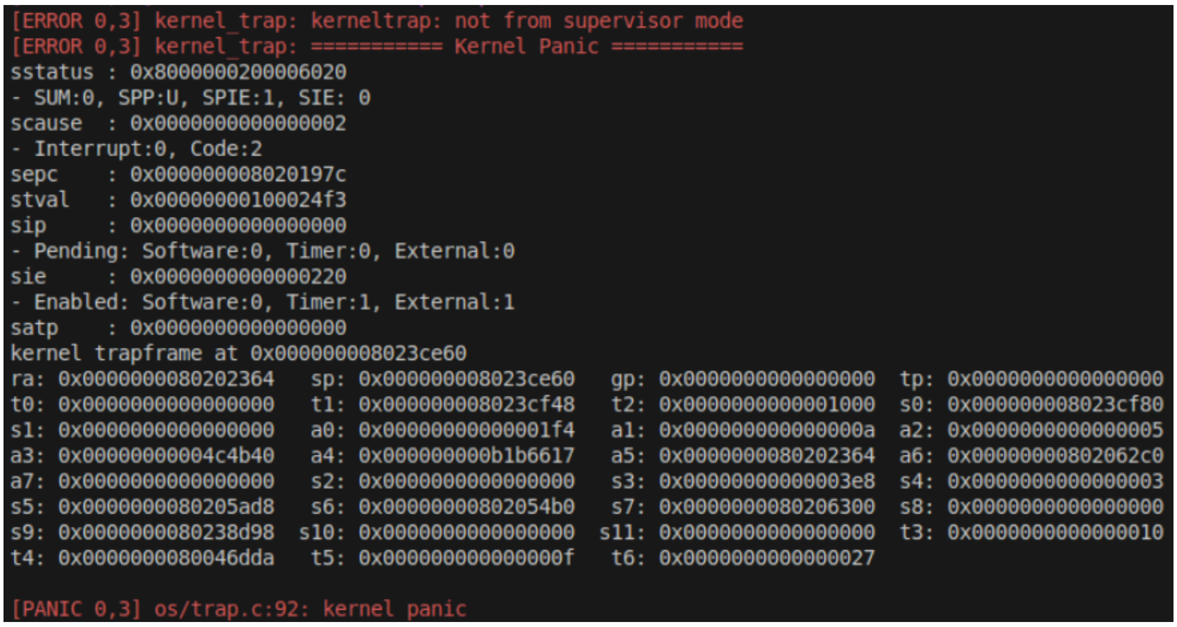
```
case SupervisorTimer:
            tracef("s-timer interrupt, cycle: %d", r_time());
            set_next_timer();
            // we never preempt kernel threads.
             struct proc* p = curr_proc();
             if (p != NULL) {
                    int saved_trap = mycpu()->inkernel_trap;
                    mycpu()->inkernel_trap = 0;
                     yield();
                     mycpu()->inkernel_trap = saved_trap;
            }
             break;
```

I add the code in trap.c, kernel_trap() function, following the checkpoint.

1.2 Result



```
[WARN  0,2] worker: thread 2: starting
[INFO  0,2] worker: thread 2: count 1000, sleeping
[sched 0,2] sched: switch to scheduler pid(2)
[sched 0,-1] scheduler: switch to proc pid(3)
[WARN  0,3] worker: thread 3: starting
[INFO  0,3] worker: thread 3: count 2000, sleeping
[sched 0,3] sched: switch to scheduler pid(3)
[sched 0,-1] scheduler: switch to proc pid(4)
[WARN  0,4] worker: thread 4: starting
[INFO  0,4] worker: thread 4: count 3000, sleeping
[sched 0,4] sched: switch to scheduler pid(4)
[sched 0,-1] scheduler: switch to proc pid(5)
[WARN  0,5] worker: thread 5: starting
[INFO  0,5] worker: thread 5: count 4000, sleeping
[sched 0,5] sched: switch to scheduler pid(5)
[sched 0,-1] scheduler: switch to proc pid(6)
[WARN  0,6] worker: thread 6: starting
[INFO  0,6] worker: thread 6: count 5000, sleeping
[sched 0,6] sched: switch to scheduler pid(6)
[sched 0,-1] scheduler: switch to proc pid(7)
[WARN  0,7] worker: thread 7: starting
[INFO  0,7] worker: thread 7: count 6000, sleeping
[sched 0,7] sched: switch to scheduler pid(7)
[sched 0,-1] scheduler: switch to proc pid(8)
[WARN  0,8] worker: thread 8: starting
[INFO  0,8] worker: thread 8: count 7000, sleeping
[sched 0,8] sched: switch to scheduler pid(8)
[sched 0,-1] scheduler: switch to proc pid(9)
[WARN  0,9] worker: thread 9: starting
[INFO  0,9] worker: thread 9: count 8000, sleeping
[sched 0,9] sched: switch to scheduler pid(9)
[sched 0,-1] scheduler: switch to proc pid(2)
[INFO  0,2] worker: thread 2: count 9000, sleeping
```



```
[ERROR 0,3] kernel_trap: kerneltrap: not from supervisor mode
[ERROR 0,3] kernel_trap: =========== Kernel Panic ===========
sstatus : 0x8000000200006020
- SUM:0, SPP:U, SPIE:1, SIE: 0
scause  : 0x0000000000000002
- Interrupt:0, Code:2
sepc    : 0x000000008020197c
stval   : 0x00000000100024f3
sip     : 0x0000000000000000
- Pending: Software:0, Timer:0, External:0
sie     : 0x0000000000000220
- Enabled: Software:0, Timer:1, External:1
satp    : 0x0000000000000000
kernel trapframe at 0x000000008023ce60
ra: 0x0000000080202364   sp: 0x000000008023ce60   gp: 0x0000000000000000   tp: 0x0000000000000000
t0: 0x0000000000000000   t1: 0x000000008023cf48   t2: 0x0000000000001000   s0: 0x000000008023cf80
s1: 0x0000000000000000   a0: 0x00000000000001f4   a1: 0x000000000000000a   a2: 0x0000000000000005
a3: 0x00000000004c4b40   a4: 0x000000000b1b6617   a5: 0x0000000080202364   a6: 0x00000000802062c0
a7: 0x0000000000000000   s2: 0x0000000000000000   s3: 0x00000000000003e8   s4: 0x0000000000000003
s5: 0x0000000080205ad8   s6: 0x00000000802054b0   s7: 0x0000000080206300   s8: 0x0000000000000000
s9: 0x0000000080238d98  s10: 0x0000000000000000  s11: 0x0000000000000000   t3: 0x0000000000000010
t4: 0x0000000080046dda   t5: 0x000000000000000f   t6: 0x0000000000000027

[PANIC 0,3] os/trap.c:92: kernel panic
```

第2章　Checkpoint2

2.1 Code

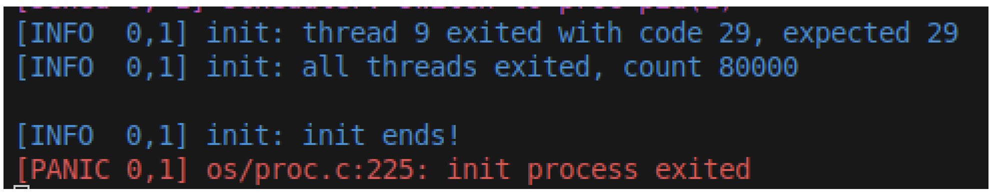
From checkpoint1 I found that SPP:U, means that before entering this Trap, the CPU is in U-mode.

When CPU execute the ‘sret’, it will clear the **sstatus.SPP** and return to U-mode.

To solve the problem, I add the codes in sched.c as follows.

```
uint64 sstatus, sepc;
sstatus = r_sstatus();
sepc = r_sepc();


swtch(&p->context, &mycpu()->sched_context);


w_sstatus(sstatus);
w_sepc(sepc);
```

2.2 Result



```
[INFO  0,1] init: thread 9 exited with code 29, expected 29
[INFO  0,1] init: all threads exited, count 80000

[INFO  0,1] init: init ends!
[PANIC 0,1] os/proc.c:225: init process exited
```

第3章 Checkpoint3

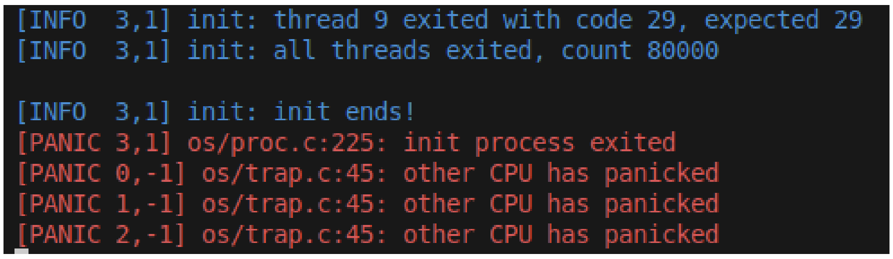
3.1 Code

To solve the problem, I add the codes in sched.c as follows. I save restored a register ra in yield().

```
void yield() {
    struct proc *p = curr_proc();
    debugf("yield: (%d)%p", p->pid, p);

    register unsigned long old_ra asm("ra");
    unsigned long ra_ = old_ra;

    acquire(&p->lock);
    p->state = RUNNABLE;
    sched();

    old_ra = ra_;
    release(&p->lock);
}
```

3.2 Result

I use make clean and make runsmp to run the code for 10 times, and each time there is the info as follows:



```
[INFO  3,1] init: thread 9 exited with code 29, expected 29
[INFO  3,1] init: all threads exited, count 80000

[INFO  3,1] init: init ends!
[PANIC 3,1] os/proc.c:225: init process exited
[PANIC 0,-1] os/trap.c:45: other CPU has panicked
[PANIC 1,-1] os/trap.c:45: other CPU has panicked
[PANIC 2,-1] os/trap.c:45: other CPU has panicked
```

第4章　Conclusion

I finish this task using about 5 hours.