Introduction to Operating System

HW3 Synchronization Report

0610862 游祖霖

Language: Assembly

Spinlock implementation:

```
spin_lock:
    xorl %ecx, %ecx                 ;Set ECX to 1 by first zero it out and then increment.
    incl %ecx                       ;(more efficient than simply moving 1 into ECX)
spin_lock_retry:
    rep; nop                        ;This command is explained further below.
    xorl %eax, %eax                 ;Zero out EAX, because cmpxchg compares against EAX.
    lock cmpxchgl %ecx, (%rdi)      ;If mutex is 0, write ECX(1) to it. (Atomically with “lock” prefix)
    jnz spin_lock_retry             ;Mutex is not 0, spin until lock is unlocked.
    ret                             ;Mutex was 0 and is now set to 1, lock is acquired.

spin_unlock:
    movl $0, (%rdi)                 ;Set mutex to 0.
    ret                             ;Lock is released.
```

Further informations:

- ***rep; nop***: [rep; nop] is the same as the pause instruction (opcode F390).
  It is important for performance on certain Intel CPU (particularly with HyperThreading).

From Intel ISA reference page 233: ***link***

> *Improves the performance of spin-wait loops. When executing a “spin-wait loop,” processors will suffer a severe performance penalty when exiting the loop because it detects a possible memory order violation. The PAUSE instruction provides a hint to the processor that the code sequence is a spin-wait loop. The processor uses this hint to avoid the memory order violation in most situations, which greatly improves processor performance. For this reason, it is recommended that a PAUSE instruction be placed in all spin-wait loops.*

- ***lock***: [lock] prefix ensures cmpxchg executes atomically.

From x86 ref: ***link***

> *Causes the processor’s LOCK# signal to be asserted during execution of the accompanying instruction (turns the instruction into an atomic instruction). In a multiprocessor environment, the LOCK# signal ensures that the processor has exclusive use of any shared memory while the signal is asserted.*