CSE 451: Operating Systems
Hard Lessons Learned

Windows
Debugging Deadlocks

Gary Kimura

The usual suspects

- Spinlock
- Semaphore
- Mutex
- EResource (Reader/writer lock)

What's in our favor

- Once the system is deadlocked, it doesn't go away. That is, you can slowly and painfully walk through all the locks on the system and all the threads on the system and see each thread owns and what it is waiting on.
- This does require the ability to identify the owner(s) of a lock.
- Once you draw the graph. You have the deadlock.
- Often the harder part is figuring out how to avoid the deadlock.

What didn’t work well

- Mutex levels. In theory they avoided deadlocks but in practice they were too cumbersome to use, and deadlocks were still possible when mixed with other kinds of locks.

Summary

- In the Windows Kernel deadlock avoidance was the strategy taken.
- Slowly, most deadlocks have been eliminated, but there are probably still some unusual situations where deadlocks can still occur.
- Don’t confuse starvation with deadlocks