Chapter 14

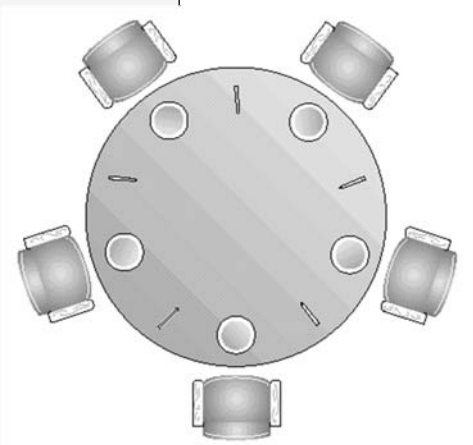
Synchronization

Standard synchronization problems

- Concurrent readers and writers
- Single writer multiple readers
- Consumer / Producer
- Sleeping barber
- Lock free circular buffer
- etc.

The dining philosophers

- It Illustrates the difficulty of allocating resources among process without deadlock and starvation
- n philosophers
- Each philosopher can have one of two states: Eating or thinking
- Each needs to use 2 forks for eating
- Only n forks are available



Elevated IRQLs and synchronization

- Elevating IRQL only does not synchronize
  - `KeRaiseIrql(...)` and `KeLowerIrql(...)` not usable for synchronization
  - Only works on single processor systems
  - Windows is a symmetric multiprocessor system
- Different multi processor safe synchronization mechanisms for elevated IRQLs
- Interlocked operations available on all IRQLs

WDM synchronization up to IRQL APC_LEVEL

- Dispatcher Objects
  - Similar objects as in Windows API
  - Event, Mutex and Semaphore
  - Thread object obtained by `ObReferenceObjectByHandle()` is waitable
- Wait functions
  - Similar functions as in Windows API
  - `KeWaitForSingleObject(...)`
  - `KeWaitForMultipleObjects(...)`
  - `KeWaitForMutexObject(...)`

WDM synchronization on IRQL APC_LEVEL

- Fast Mutexes
  - `ExInitializeFastMutex()`
  - `ExAcquireFastMutex()`
  - `ExReleaseFastMutex()`
  - Cannot be acquired recursively
- `ERESOURCE` locks
  - `ExAcquireResourceExclusive(...)`
  - `ExAcquireResourceSharedLite(...)`
  - Implementation of standard reader/writer synchronization
  - For performance critical operations

WDM synchronization up to IRQL DISPATCH_LEVEL

- Spin locks
  - Mutual exclusion
  - Busy waiting, CPU actively spinning at IRQL `DISPATCH_LEVEL`
  - No recursive acquisition
- Explicit programming
  - `KeInitializeSpinLock(...)`
  - `KeAcquireSpinLock(...)`
  - `KeReleaseSpinLock(...)`
- In stack queued spin locks
  - Better scaling on NUMA systems
  - Acquisition order preserved

WDM synchronization up to IRQL DIRQL

- Interrupt spin locks
  - Synchronization of Interrupt Service Routine
  - Up to IRQL `DIRQL`
- Implicit programming
  - No acquisition necessary in ISR
  - Implicit acquisition by `KeSynchronizeExecution(...)`
- Explicit acquisition and release
  - `KeAcquireInterruptSpinLock(...)`
  - `KeReleaseInterruptSpinLock(...)`

Chapter 14:
Synchronization

KMDF locks

- Framework wait locks
  below `IRQL_DISPATCH_LEVEL`
- Framework spin locks
  up to `IRQL_DISPATCH_LEVEL`
- Framework interrupt locks up to `IRQL_DIRQL`
  - Implicit acquisition
    `WdfInterruptSynchronize()` schedules
    `EvtInterruptSynchronize()` callback
  - Explicit acquisition (performance!)
    `WdfInterruptAcquireLock()`

Chapter 14:
Synchronization

KMDF automatic synchronization of PnP and Power Management

- Plug'n'Play and Power Managemen is always synchronized on device level
- `IRQL PASSIVE_LEVEL`
- Automatic device level PnP and Power Management synchronization includes
  - General device object event callbacks
  - Functional device object (FDO) event callbacks
  - Physical device object (PDO) event callbacks
- Exceptions:
  - `EvtDeviceQueryStop`
  - `EvtDeviceSurpriseRemoval`
  - `EvtDeviceQueryRemove`

Framework automatic synchronization of IO

- Optional automatic synchronization on device level or queue level
  - `WdfSynchronizationScopeDevice`
  - `WdfSynchronizationScopeQueue`
  - `WdfSynchronizationScopeNone`
- Optional IO synchronization includes
  - Queue object (Request handlers, `EvtIoQueueState`, `EvtIoResume`, `EvtIoStop`)
  - File object (all callback functions)
  - Request object (`EvtRequestCancel`)
- Object execution levels `WDF_EXECUTION_LEVEL`
  - `WdfExecutionLevelPassive`
  - `WdfExecutionLevelDispatch`

KMDF object presentation lock

- Object locks for automatic synchronization
  - For framework device object and queue object
  - Up to `WdfExecutionLevelDispatch` (WDM IRQL `DISPATCH_LEVEL`)
  - Synchronizes framework object callbacks
  - Can explicitly be acquired, too

Further information

- Scheduling, Thread Context, and IRQL
  - http://msdn.microsoft.com/en-us/windows/hardware/gg487402
- Summary of Windows Synchronization Primitives
  - http://msdn.microsoft.com/en-us/windows/hardware/gg463245
- Locks, Deadlocks, and Synchronization
  - http://www.microsoft.com/whdc/driver/kernel/locks.mspx