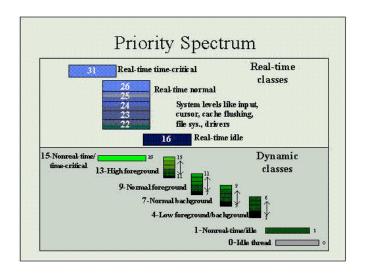
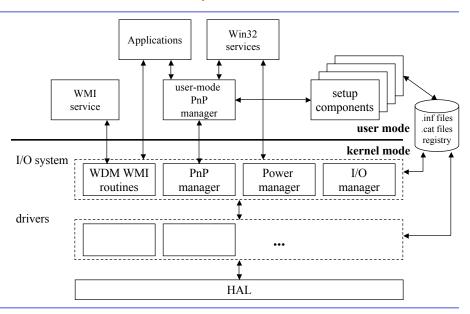
Windows NT and Real-Time?

- Reading: **"Inside Microsoft Windows 2000"**, (Solomon, Russinovich, Microsoft Programming Series)
- "Real-Time Systems and Microsoft Windows NT" (MSDN Library)
- "Windows XP with RTX The off-the-shelf platform for Integrated Communication Equipment" (www.venturcom.com)

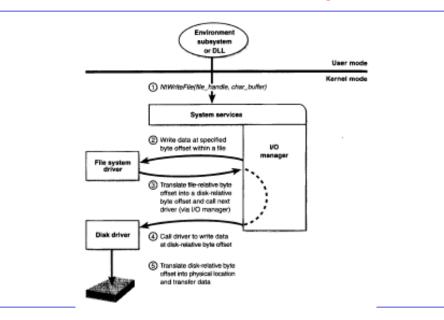
Priorities in Windows NT/2000

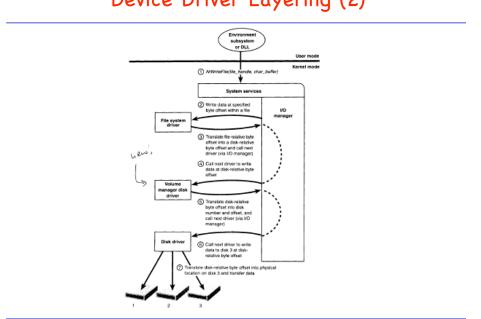




IO System Components (Windows 2k)

Device Driver Layering

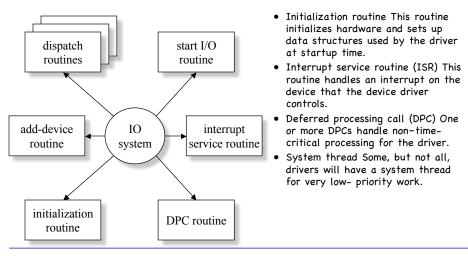


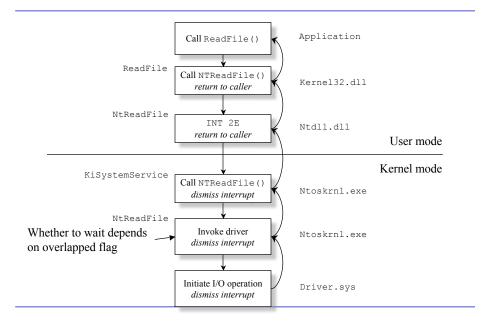


Device Driver Layering (2)

Primary Device Driver Routines

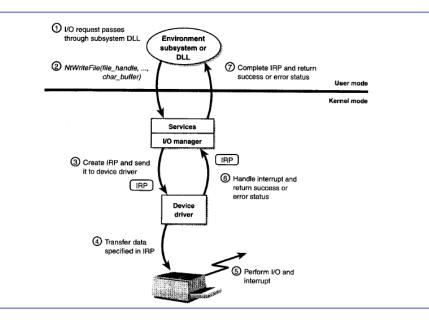
• NT/2000 device drivers run entirely within the system process and have access to all hardware through the HAL. A typical device driver will have several components:

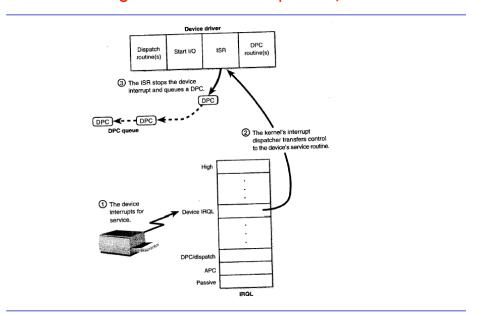




Control Flow for an IO Operation

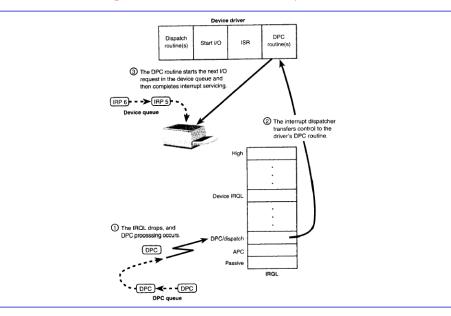
Queueing and Completing a Synchronous Request

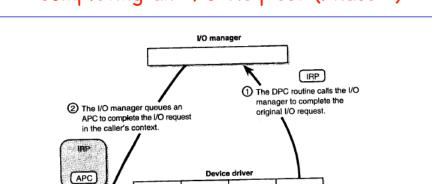




Servicing a Device Interrupt (only Phase I)

Servicing a Device Interrupt (Phase II)





Start I/O

IRP

(APC)

Dispatch

routine(s)

Thread's APC queue

1RP

(APC)

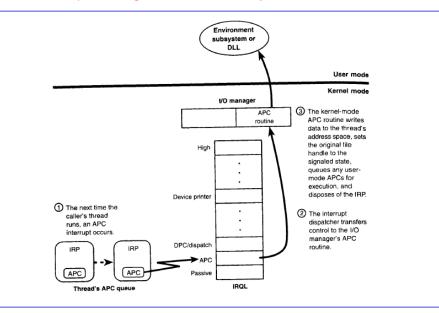
DPC

routine(s)

ISR

Completing an I/O Request (Phase I)

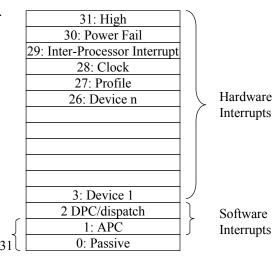
Completing an I/O Request (Phase II)



Priority Levels vs. Interrupt Levels

- The HAL maps hardwareinterrupt numbers to IRQLs.
- IRQLs are not the same as IRQs in x86.
- Scheduling priority is attribute of thread, while IRQL is attribute of an interrupt source.
- Lazy IRQL management for slow PICs.
- Code running at DPC/dispatch level or above can't wait on object if so would necessitate scheduler to invoke another thread.

Thread Priorities 0-31



Memory Management

- Paging I/O occurs at a lower priority level than the real-time priority process levels. Paging within the real-time process is still free to occur, but this really ensures that background virtual memory management won't interfere with processing at real-time priorities.
- Windows NT permits an application to lock itself into memory so that it is not affected by paging within its own process. This allows even very large processes (such as raster image processing, where some processes are over 100MB) to lock all their memory down into physical memory and avoid the overhead of paging, while allowing the rest of the system to function normally.
- Windows NT memory management allows for memory mapping, which permits multiple processes—even device drivers and user applications—to share the same physical memory. This results in very fast data transfers between cooperating processes or between a driver and an application. Memory mapping can be used to dramatically enhance real-time performance.

Windows 2000/NT and Real-Time Processing

- Windows 2000/NT does not prioritize device IRQs in controllable way.
- User-level applications execute only when a processor's IRQL is at passive level.
- System's devices and device drivers not the OS ultimately determine the worst-case delay.
- This is a problem with off-the-shelf hardware and drivers.
- System designer must bound the length of device's ISR and DPC in the worst case.
- Embedded versions of Windows NT/2000 provide control over memory footprint etc, but are not real-time capable.
- Extensions of real-time kernels can be provided through custom extensions of the HAL.

VenturCom RTX Architecture

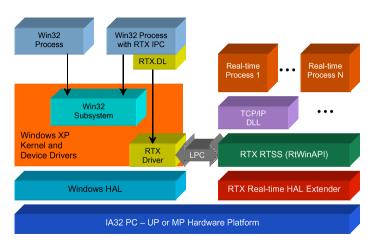


Figure 1: RTX Architecture