

Interrupt Request Levels

主講人：虞台文

1

Content

- ✧ Multiprocessor Machine
- ✧ IRQs
- ✧ Functions for IRQs

2

Interrupt Request Levels

Multiprocessor Machine

3

Symmetric Multiprocessor Platforms

- ✧ The NT-based operating system is designed to run on **multiprocessor** machines, usually configured to be:
- ✧ Homogenous
- ✧ Closely coupled
- ✧ Symmetric

4

Symmetric Multiprocessor Platforms

- ✧ The NT-based operating system is designed to run on **multiprocessor** machines, usually configured to be:
- ✧ Homogenous
- ✧ Closely coupled
- ✧ Symmetric

All CPUs are identical, and either all have identical coprocessors or none has a coprocessor.

5

Symmetric Multiprocessor Platforms

- ✧ The NT-based operating system is designed to run on **multiprocessor** machines, usually configured to be:
- ✧ Homogenous
- ✧ Closely coupled
- ✧ Symmetric

All CPUs share memory and have uniform access to memory.

6

Symmetric Multiprocessor Platforms

- ✧ The NT-based operating system is designed to run on **multiprocessor** machines, usually configured to be:
- ✧ Homogenous
- ✧ Closely coupled
- ✧ Symmetric

Every CPU can

- access memory,
- handle any interrupt, and
- access I/O control registers just like every other CPU in the system.

7

Interrupt Request Levels

IRQs

8

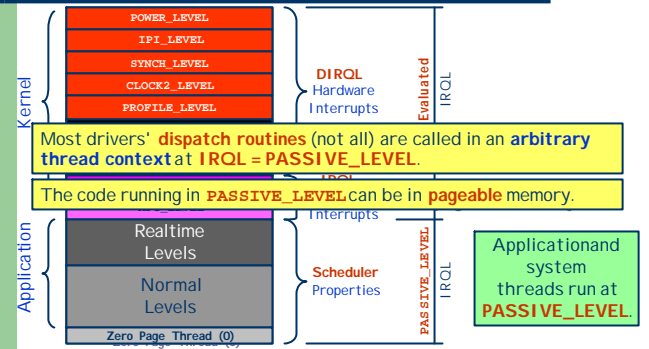
The Priority System

NT-based OS has a **two-level** priority model:

- Higher-Level
 - Controlled by **hardware** and **software interrupt**
- Lower-Level
 - Controlled by the **scheduler**

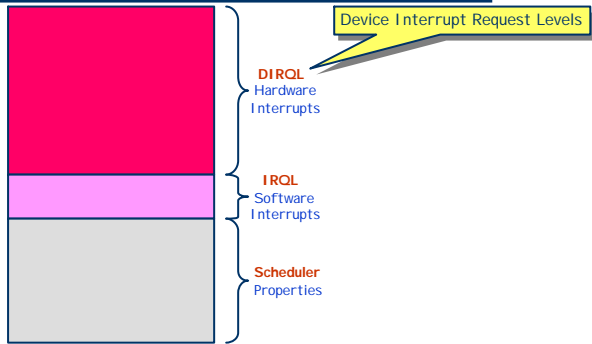
9

PASSIVE_LEVEL IRQL



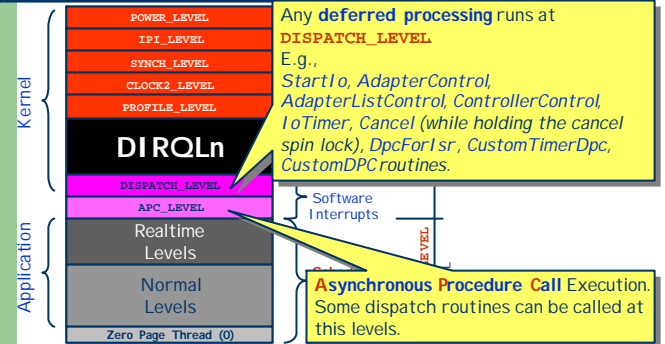
13

The Priority Levels



10

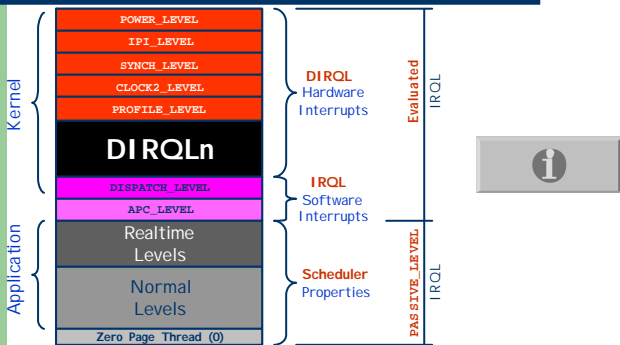
Evaluated IRQLs



14

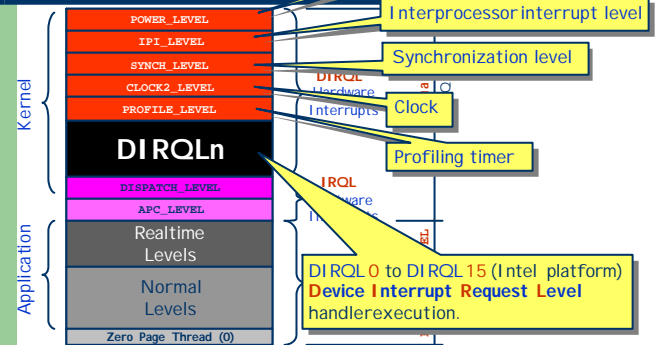
A routine can be **preempted** by an interrupt with a higher IRQL.

The Priority Levels



11

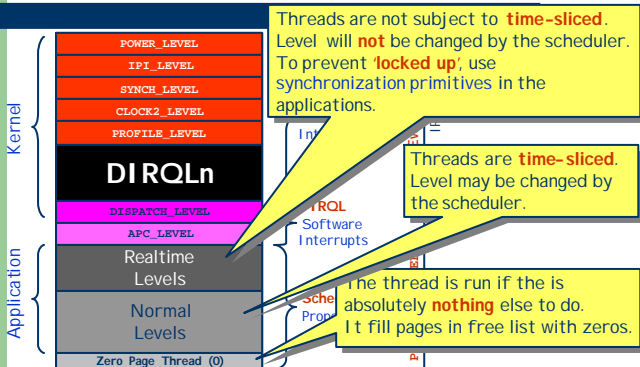
Evaluated IRQLs



15

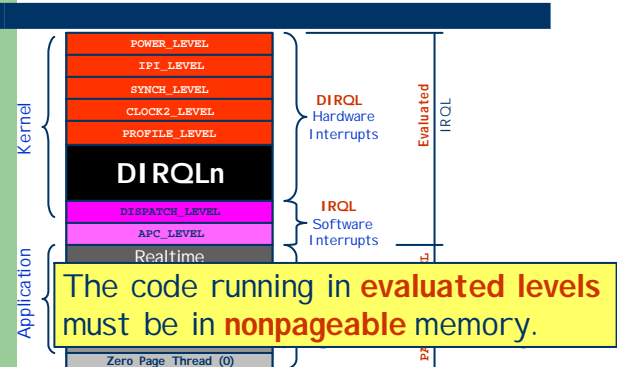
DDK calls IRQL ? DISPATCH_LEVEL as **raised IRQL**.

PASSIVE_LEVEL IRQL



12

Evaluated IRQLs



16

Interrupt Request Levels

Functions for IRQLs

17

Data Type of IRQL

⚡ KIRQL ?

the typedef name for an integer that holds an IRQL value.

18

Kernel Routines for IRQLs

⚡ KeGetCurrentIrql

- returns the current IRQL

⚡ KeRaiseIrql

- raises the hardware priority to a given IRQL

⚡ KeLowerIrql

- restores the IRQL on the current processor to its original value

19

Bug Check 0xA: IRQL_NOT_LESS_OR_EQUAL

⚡ This bug check is issued if paged memory (or invalid memory) is accessed when the IRQL is too high.



20

Example: Explicitly Controlling IRQL

```
KIRQL oldirql;  
. . . . .  
ASSERT(KeGetCurrentIrql() <= DISPATCH_LEVEL);  
KeRaiseIrql(DISPATCH_LEVEL, &oldirql);  
. . . . .  
KeLowerIrql(oldirql);  
. . . . .
```

21