操作系统

Operating system

徐子川

大连理工大学

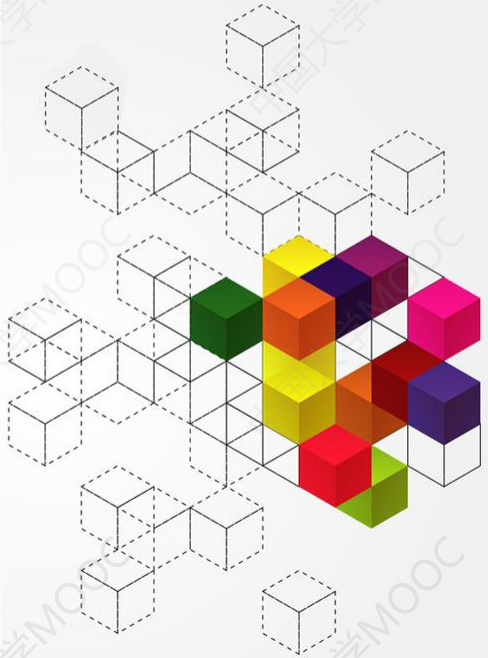
内容纲要

5.6 调度实现示例

一、 Linux调度算法

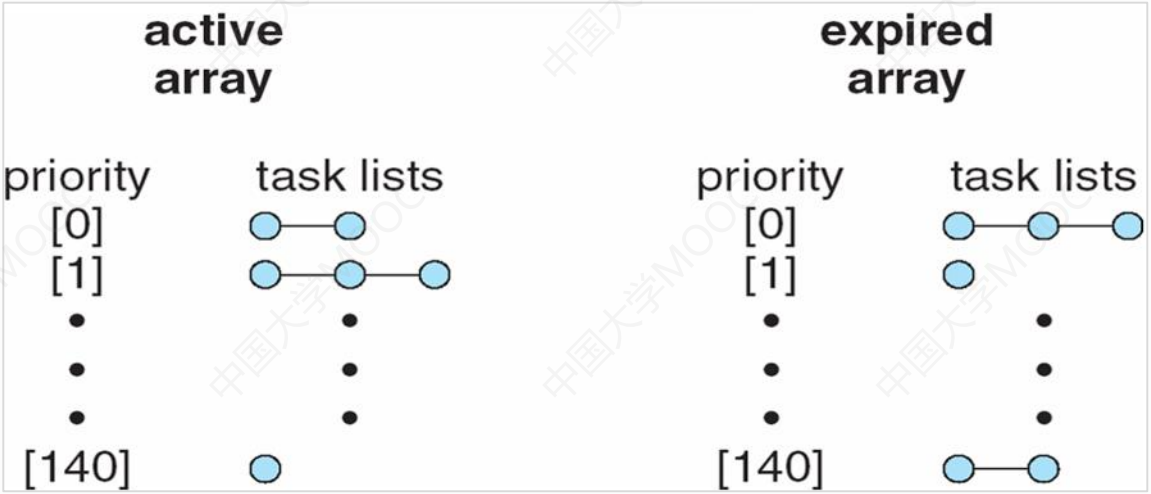
二、 Windows调度

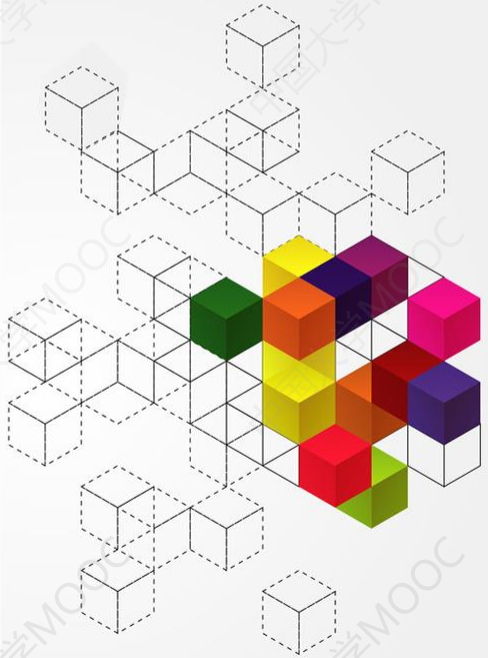
三、 Solaris调度



一、Linux调度

- Scheduler Data Structure (Linux 2.6+)
  - Expire Array and Active Array

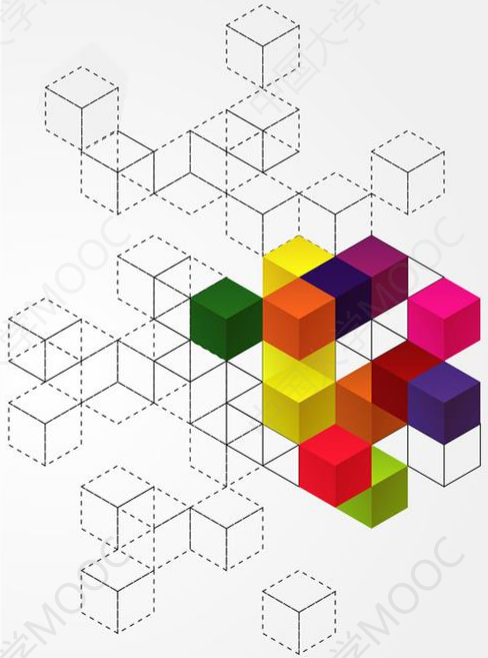




一、Linux调度

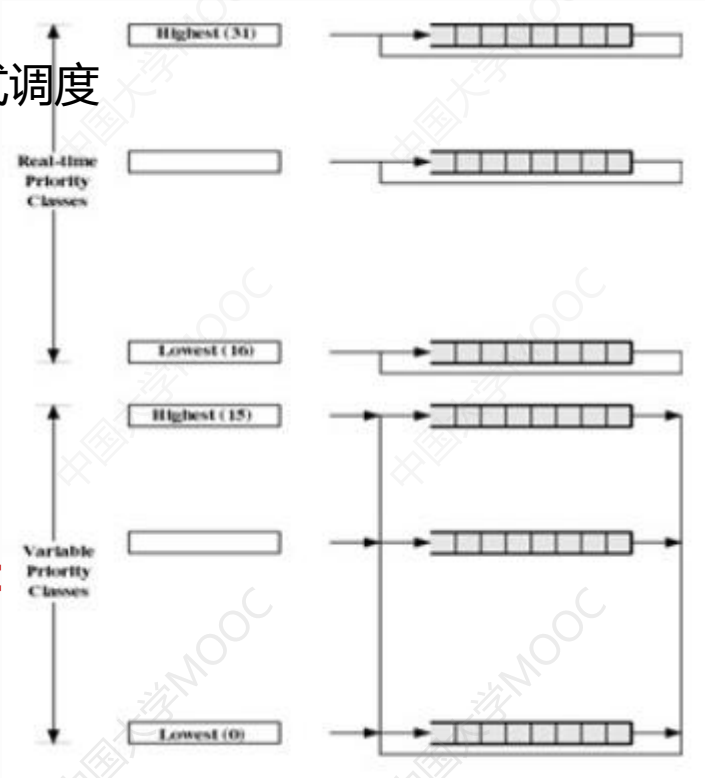
- Linux调度优先级
  - 用优先数0-140表示
  - 优先数越大，优先级越低

numeric priority	relative priority		time quantum
0	highest	real-time tasks	200 ms
•			
•			
•			
99			
100		other tasks	
•			
•			
•			
140	lowest		10 ms



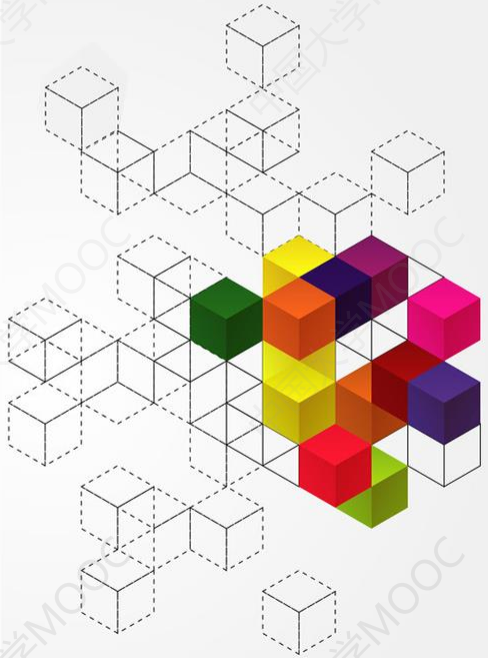
二、Windows调度

- Windows NT-based OS use a multilevel feedback queue.
  - 基于优先级的抢占式调度
  - 使用多级反馈队列



32个优先级（1-15：variable class, 16-31:real time）

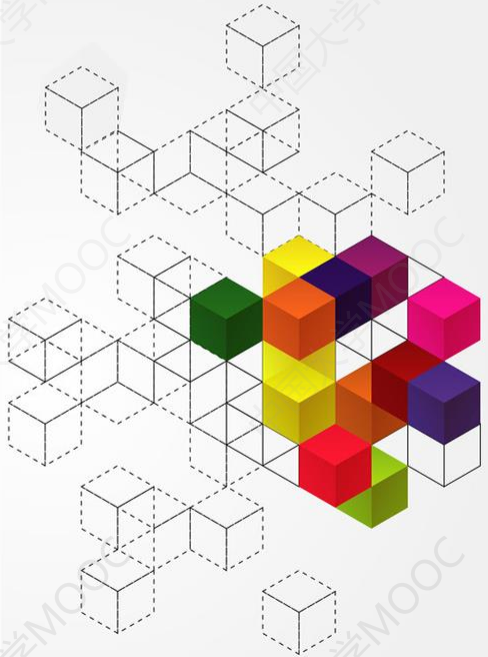
优先级0：赋予memory-management thread



二、Windows调度

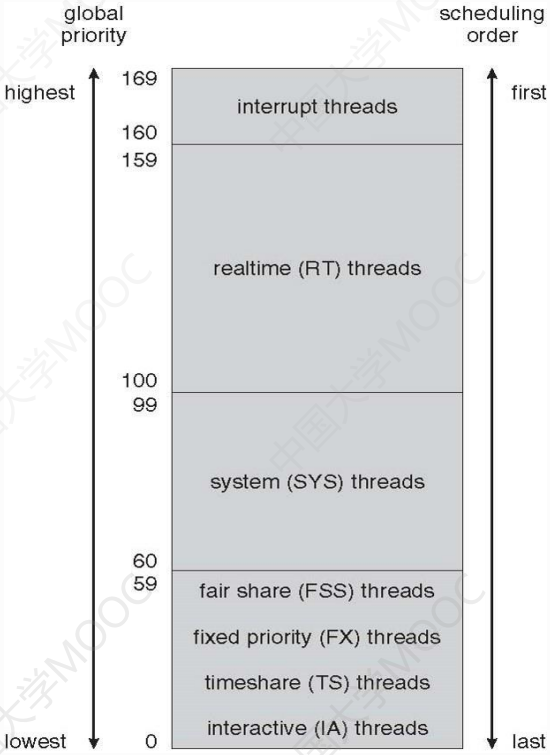
- Windows NT内核的优先级设定
  - 两个维度：base priority and relative priority

	real-time	high	above normal	normal	below normal	idle priority
time-critical	31	15	15	15	15	15
highest	26	15	12	10	8	6
above normal	25	14	11	9	7	5
normal	24	13	10	8	6	4
below normal	23	12	9	7	5	3
lowest	22	11	8	6	4	2
idle	16	1	1	1	1	1

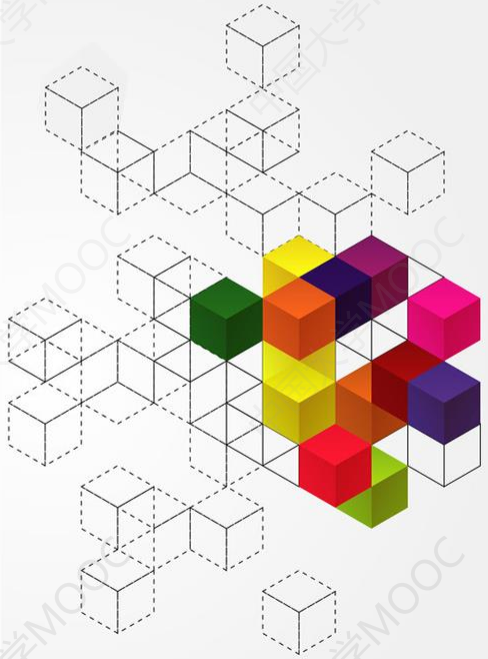


三、Solaris调度

- Solaris采用多级反馈队列调度



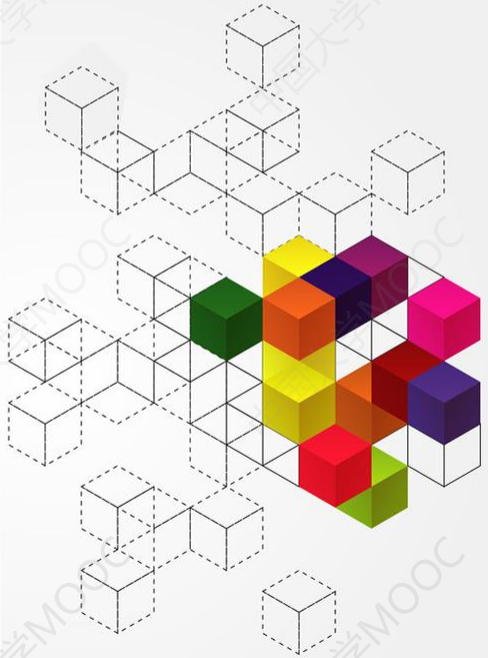
- **6类线程**
  - **Time sharing (default) 分时**
  - **Interactive 交互**
  - **Real time 实时**
  - **System 系统**
  - **Fair Share 公平轮转**
  - **Fixed priority 固定优先级**



三、Solaris调度

- 进程优先级的动态调整

priority	time quantum	time quantum expired	return from sleep
0	200	0	50
5	200	0	50
10	160	0	51
15	160	5	51
20	120	10	52
25	120	15	52
30	80	20	53
35	80	25	54
40	40	30	55
45	40	35	56
50	40	40	58
55	40	45	58
59	20	49	59



本讲小结

- Linux调度实现
- Windows调度
- Solaris调度