

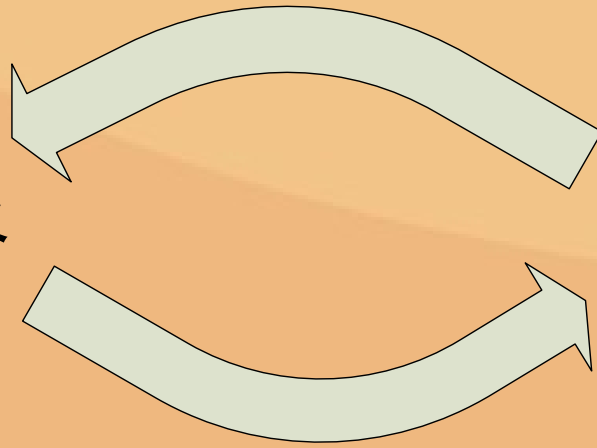
28001010 班

三大操作系统调度策略 分析与对比

主讲人：胡晓

策略

功能



CONTENT

1

调度算法评价准则

2

Unix 调度策略

3

Linux 调度策略

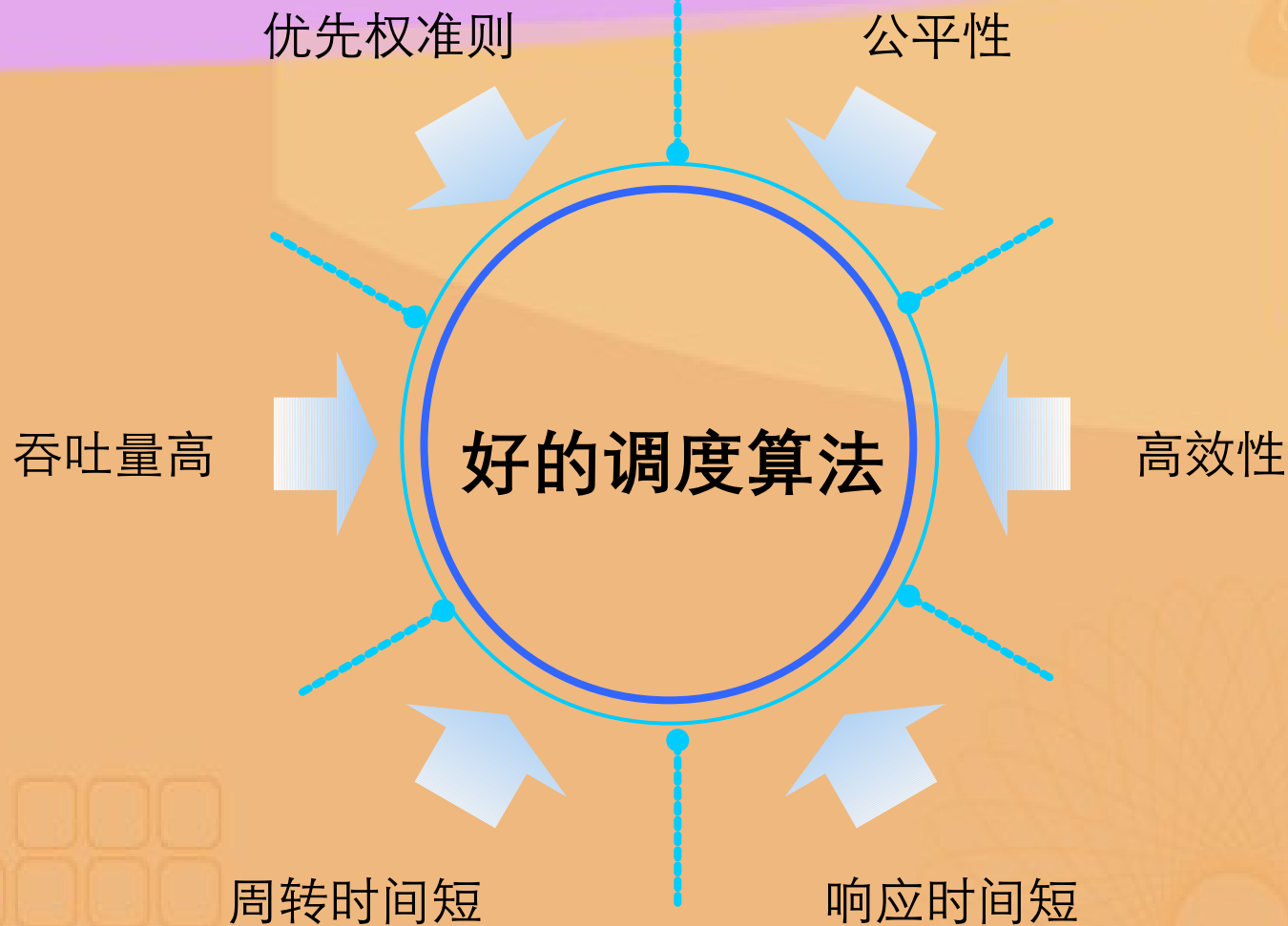
4

Windows 调度策略

5

三大操作系统调度策略对比

调度算法评价准则



基本调度算法

1

先来先服务算法（FCFS）

2

短进程优先

3

时间片轮转调度

4

最优比例优先法

四种基本算法的平均周转时间

```
22 scanf("%d",&n);
23 for(i=0;i<n;i++)
24     scanf("%d%d",&t[i].s,&t[i].len);
25 std::sort(t,t+n,cmp);
26 double res=0;
27 int pos=0;
28 int tm=0,cnt=0;
29 while(1)
30 {
31     if(front-back==0)
32     {
33         if(pos==n) break;
34         queue[back++]=t[pos];
35         pos++;
36     }
37     while(pos<n&&t[pos].s<=tm)
38     {
39         queue[back++]=t[pos];
40         pos++;
41     }
42     if(queue[front].s<=tm)
43     {
44         tm+=queue[front].len;
45         res+=tm-queue[front].s;
46     }
47     else
48     {
49         tm=queue[front].s+queue[front].len;
50         res+=queue[front].len;
```

"E:\OS project\FCFS\bin\Debug\FCFS.exe"

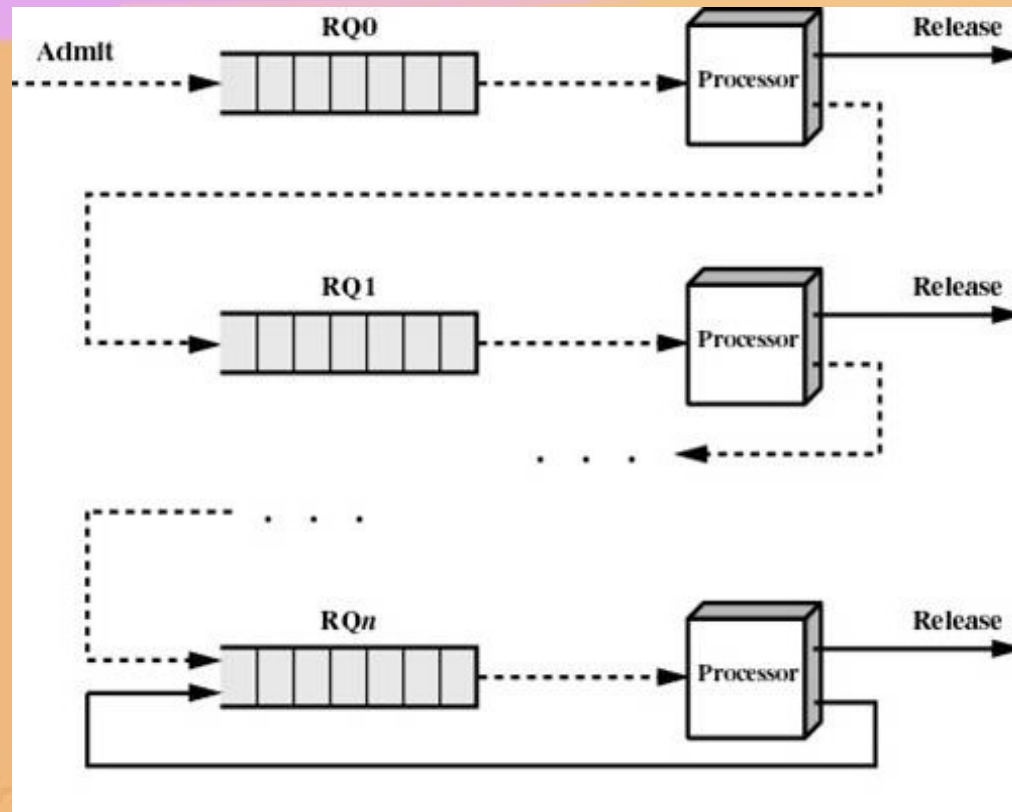
先进先出算法对10000个进程模拟运行得到平均周转时间为
24987213.714<ms>

短进程优先算法对10000个进程模拟运行得到平均周转时间为
16765544.355<ms>

RR算法对10000个进程模拟运行得到平均周转时间为
33535509.359<ms>

最优比率优先算法对10000个进程模拟运行得到平均周转时间为
16765574.435<ms>

Unix 多级反馈轮转调度描述



优先级 = 最近使用 CPU 的时间 / 2 + 基本优先级

Unix 调度策略优点浅析

多级反馈轮转调度

= 反馈法

+ 时间片轮转法

+ 优先级法

满足了响应时间的要求

对批量处理有利

公平

Unix 调度策略缺点浅析

系统开销大

为不同优先级都要分配队列
系统每隔 1s 都要重新计算进程的优先级

使用时间片轮转的方式进程切换时会增加系统额外的开销

实时性不好

基本目标是提供了一个分时服务环境，未单独划分实时与非实时进程，实时任务处理能力较低

Linux 调度策略描述



Linux 调度优点浅析

优点

实时优先级高于普通进程，保证了实时进程的快速响应，**具有实时性**

对普通进程采用动态优先级算法，保证了**普通进程间的公平性**

调度算法**多样**，可用以处理不同问题

Linux 调度缺点浅析

实时进程调度采用**先进先出**的算法：

对短进程不公平
不利于 I/O 型进程
增加批处理系统中作业的平均周转时间

A B

在实时进程和普通进程都有采用**时间片轮转**

进程切换增加开销
如果进程数目增多，
进程排队等待时间会变长，响应时间会很长

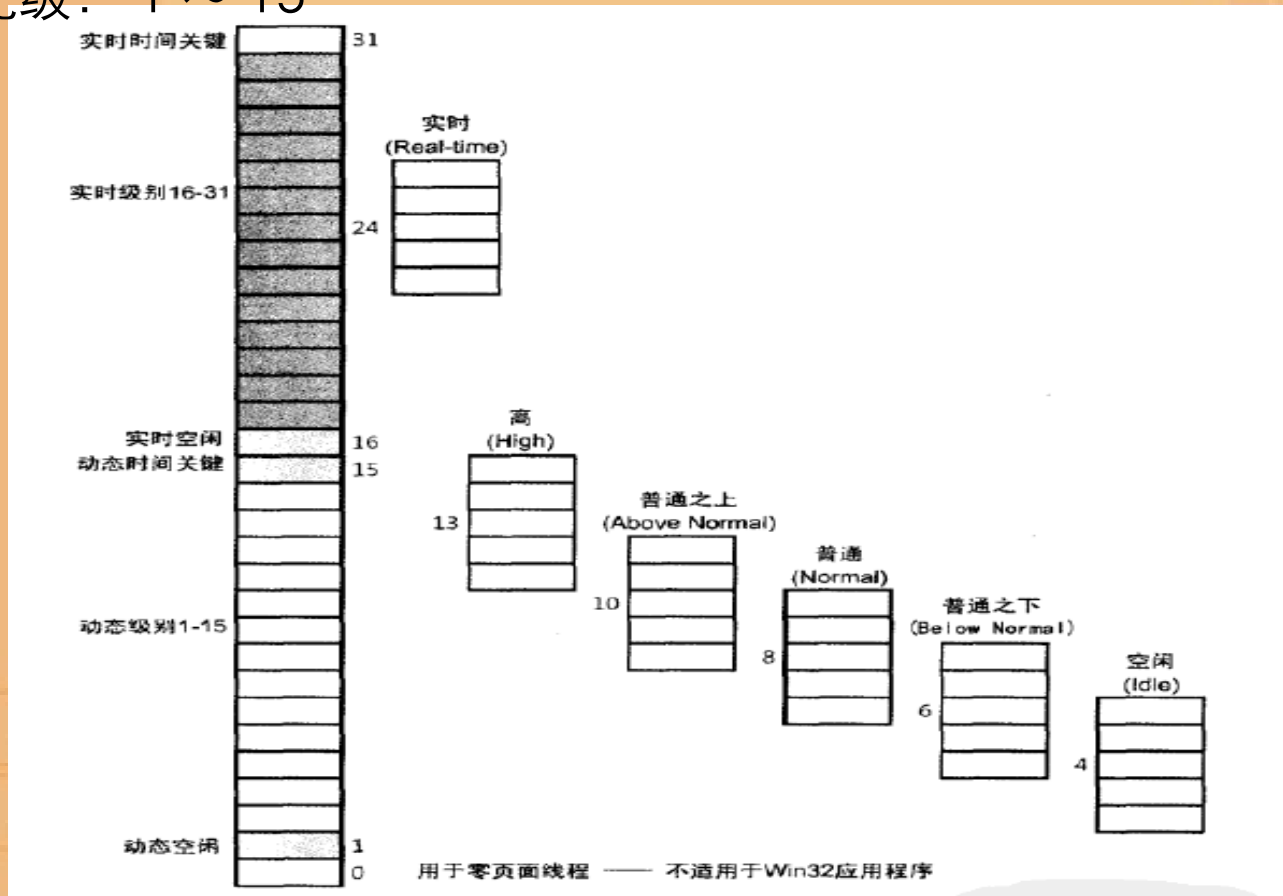
Windows NT 基于线程的优先级的调度策略

调度原则

- 每个优先级的就绪进程排成一个先进先出队列；
- 在同一优先级的各线程按时间片轮转算法进行调度；
- 当一个线程状态变成就绪时，它可能立即运行或排到相应优先级队列的尾部；
- 总运行优先级最高的就绪线程。
- 线程在给定时间配额内运行时可能被高优先级的线程抢先执行而被中断。

Windows 内部使用 32 个线程优先级，范围从 0 到 31，数值越大，优先级越高。

- 实时线程优先级：16 ~ 31
- 可变线程优先级：1 ~ 15



Windows 调度优点浅析

并发性好： 不仅进程之间可以并发执行，同属一个进程的多个线程之间也可并发执行

灵活性强： 线程是独立调度的基本单位，同一进程中的多个线程之间切换时不会引起进程切换

系统开销较小： 操作系统在创建或撤销进程以及进程切换时将显著大于线程

Windows 调度策略缺点浅析

NT

NT

WindowsNT

三大操作系统调度策略对比 (1)

Unix 是以进程为调度对象

Linux 是以进程为调度对象

调度对象对比

WindowsNT 中是以**线程**为
基本调度单位和调度对象

总结

线程的引入减少了程序并发
执行时候所付出的时空开销使
得 Windows 拥有更好的并发性

三大操作系统调度策略对比 (2)

Unix 未**对实时与非实时进程进行划分**。

Linux 对任务进行了划分
对实时和非实时作了不同优先级的区分，实时高于非实时的优先级。

优先级划分对比

Windows NT 对任务进行了划分对实时和非实时作了不同优先级的区分，实时高于非实时的优先级。

总结

三个操作系统调度策略中最不适合实时的是 unix 操作系统。因为处理实时有两套策略所以相对比较适合实时的是 Linux 操作系统。

三大操作系统调度策略对比 (3)

Unix 调度都是基于时间片的轮转，但是时间片多级
时间片长短不一

Linux 系统在基于时间片轮转的时间片长短一致，也可以使用先入先出策略

时间片划分

WindowsNT 的调度都是基于时间片的轮转，时间片长短一致

总结

根据多级划分不同长度时间片有利于不同进程且满足调度公平型，但是实时性差。

三大操作系统调度策略对比 (4)



Unix 是基于多级反馈循环调度兼而有时间片轮转，先进先出，优先级等策略，公平性好，适合各类进程（长，短），也适合分时批处理

Linux 将进程分为实时和非实时，也会根据不同优先级做多样的策略，例如先进先出，时间片轮转等，适合实时的短进程，

WindowsNT 调度策略简单只有基于优先级的时间片轮转一种，较为单一，不适合处理多种类型进程。

综合分析： Unix 最适合批处理实时性差，linux 适合实时或分时，批处理性较差。Windows 适合分时系统，实时性差，且不适合批处理场合

Windows

优先级
交互式操作调度灵活
简单

适用于PC机

Linux

实时性
长进程不会产生饥饿
多样

多功能多用途
嵌入式系统

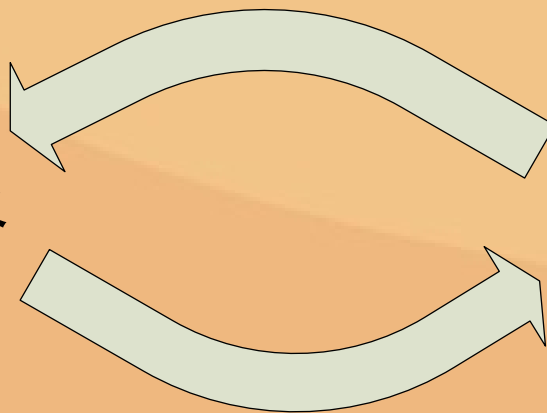
Unix

公平性

适用于批处理，银行、
网络大型机

策略

功能





Thank You!

感谢 28001010 班全体同学的大力支持

