

经典操作系统调度算法——对比演示

唐洋洋

2015-3-31

内容概要

- 回顾与铺垫
- 几个经典的调度算法
- 调度过程动态演示

回顾与铺垫

- 调度准则
 - 面向用户：周转时间 (Turnaround time)、响应时间 (Response time)、截至期限 (Deadlines)、可预测性 (Predictability)
 - 面向系统：吞吐量 (Throughput)、处理器利用率 (Processor utilization)、公平性 (Fairness)
- 决策模式
 - 非抢占模式
 - 抢占模式

回顾与铺垫

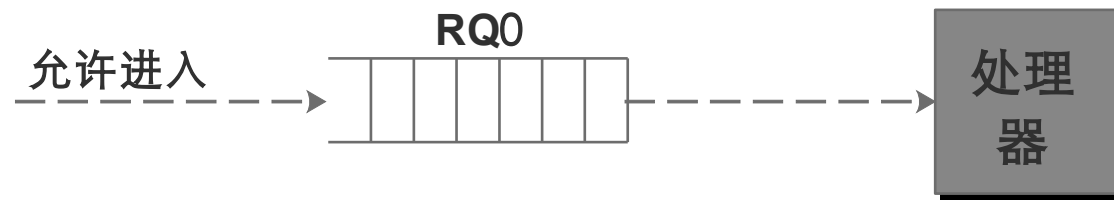
- 调度准则
 - 面向用户：周转时间 (Turnaround time)、响应时间 (Response time)、截至期限 (Deadlines)、可预测性 (Predictability)
 - 面向系统：吞吐量 (Throughput)、处理器利用率 (Processor utilization)、公平性 (Fairness)
- 决策模式
 - 非抢占模式
 - 抢占模式

w □□□□□□□□□□□□□□□□
 e □□□□□□□□□□□□□□
 s □□□□□□□□□□

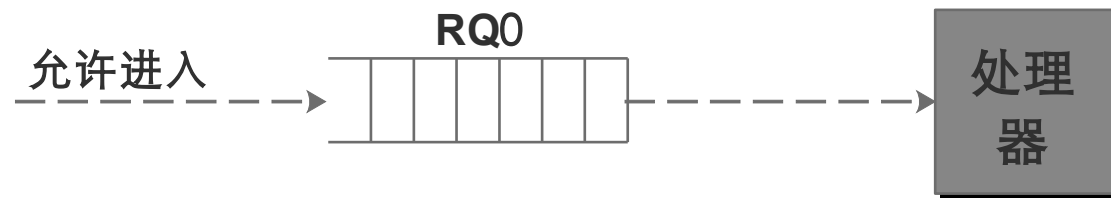
几个经典的调度算法

- 先来先服务 (FCFS)
- 轮转 (RR)
- 最短进程优先 (SPN)
- 最短剩余时间 (SRT)
- 最高响应比优先 (HRRN)
- 反馈 (Feedback)

几个经典的调度算法

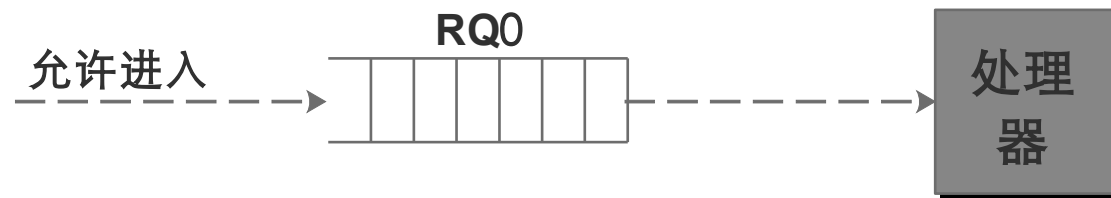


几个经典的调度算法



When to do it?
How to do it?

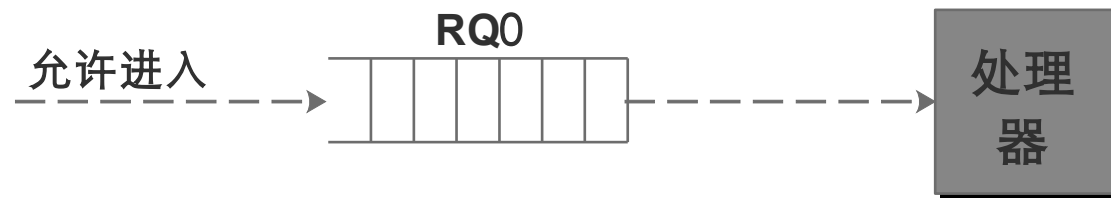
几个经典的调度算法



When to do it? When to schedule?
How to do it? How to schedule?

A red arrow points from the first column of text to the second column of text.

几个经典的调度算法

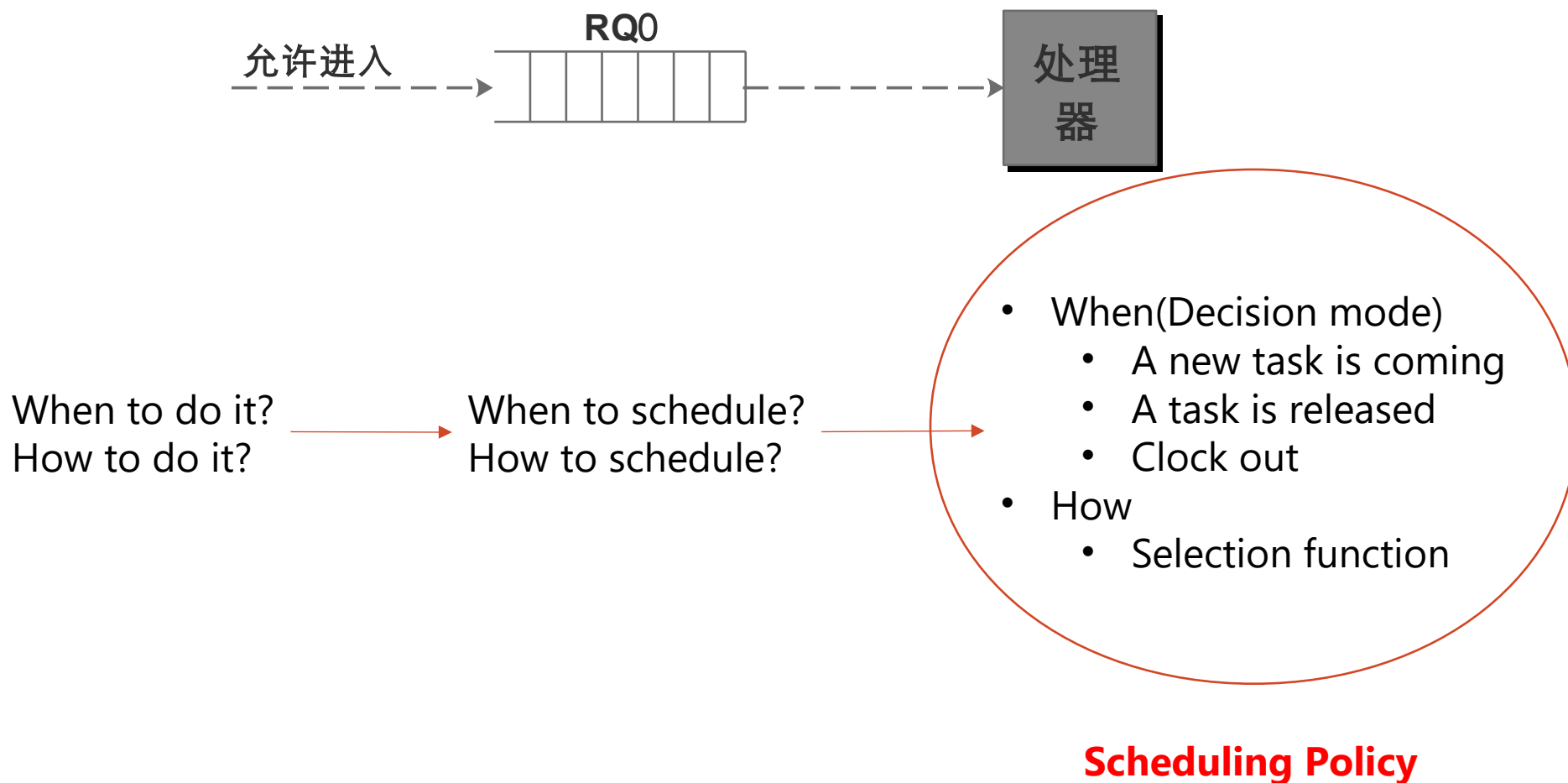


When to do it?
How to do it?

When to schedule?
How to schedule?

- When(Decision mode)
 - A new task is coming
 - A task is released
 - Clock out
- How
 - Selection function

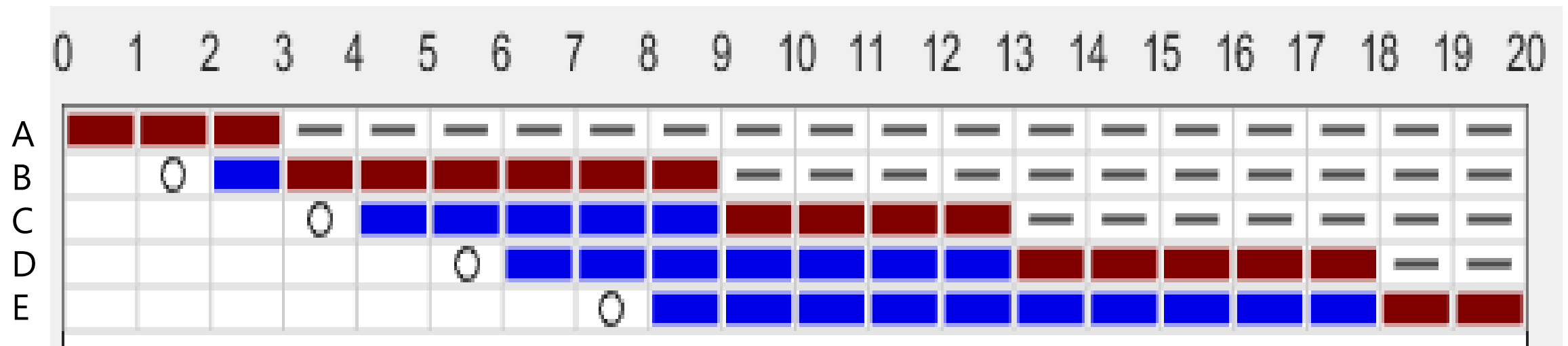
几个经典的调度算法



看图说话

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
A																					
B																					
C																					
D																					
E																					

看图说话



- New task is coming
- A task is release
- Clock out

When & How

- Selection function

1 先来先服务 (FCFS)

- 定义
 - 非抢占式，调度主要发生在进程释放时
 - $\max[w]$

1 先来先服务 (FCFS)

- 定义
 - 非抢占式，调度主要发生在进程释放时
 - $\max[w]$
- 评价
 - 简单，易实现
 - 归一化周转时间不合理，缺乏公平性，对短进程不利
 - 不适合 I/O 密集型进程

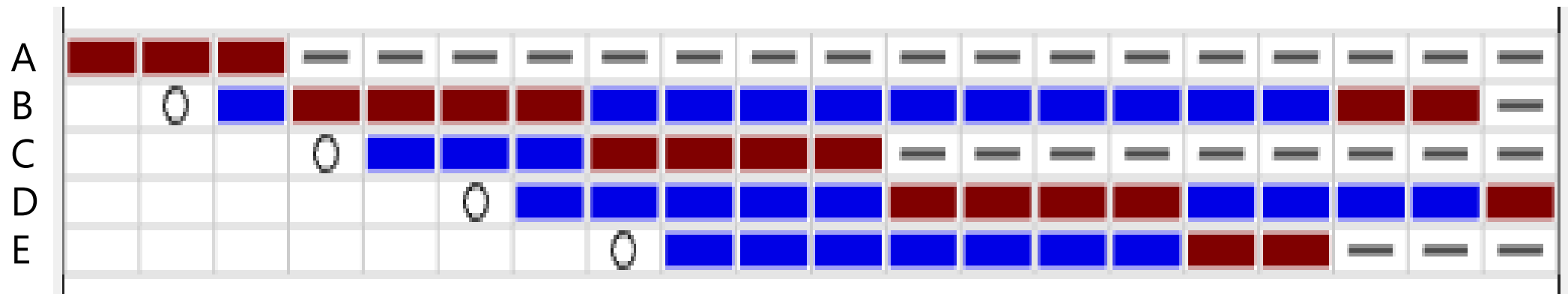
1 先来先服务 (FCFS)

- 定义
 - 非抢占式，调度主要发生在进程释放时
 - $\max[w]$
- 评价
 - 简单，易实现
 - 归一化周转时间不合理，缺乏公平性，对短进程不利
 - 不适合 I/O 密集型进程

1 先来先服务 (FCFS)

- 定义
 - 非抢占式，调度主要发生在进程释放时
 - $\max[w]$
- 评价
 - 简单，易实现
 - 归一化周转时间不合理，缺乏公平性，对短进程不利
 - 不适合 I/O 密集型进程
- 改进方案
 - 时间轮转，可以抢占（轮转）
 - 短进程优先

看图说话



- New task is coming
- A task is release
- Clock out

When & How

- Selection function

2 轮转 (RR)

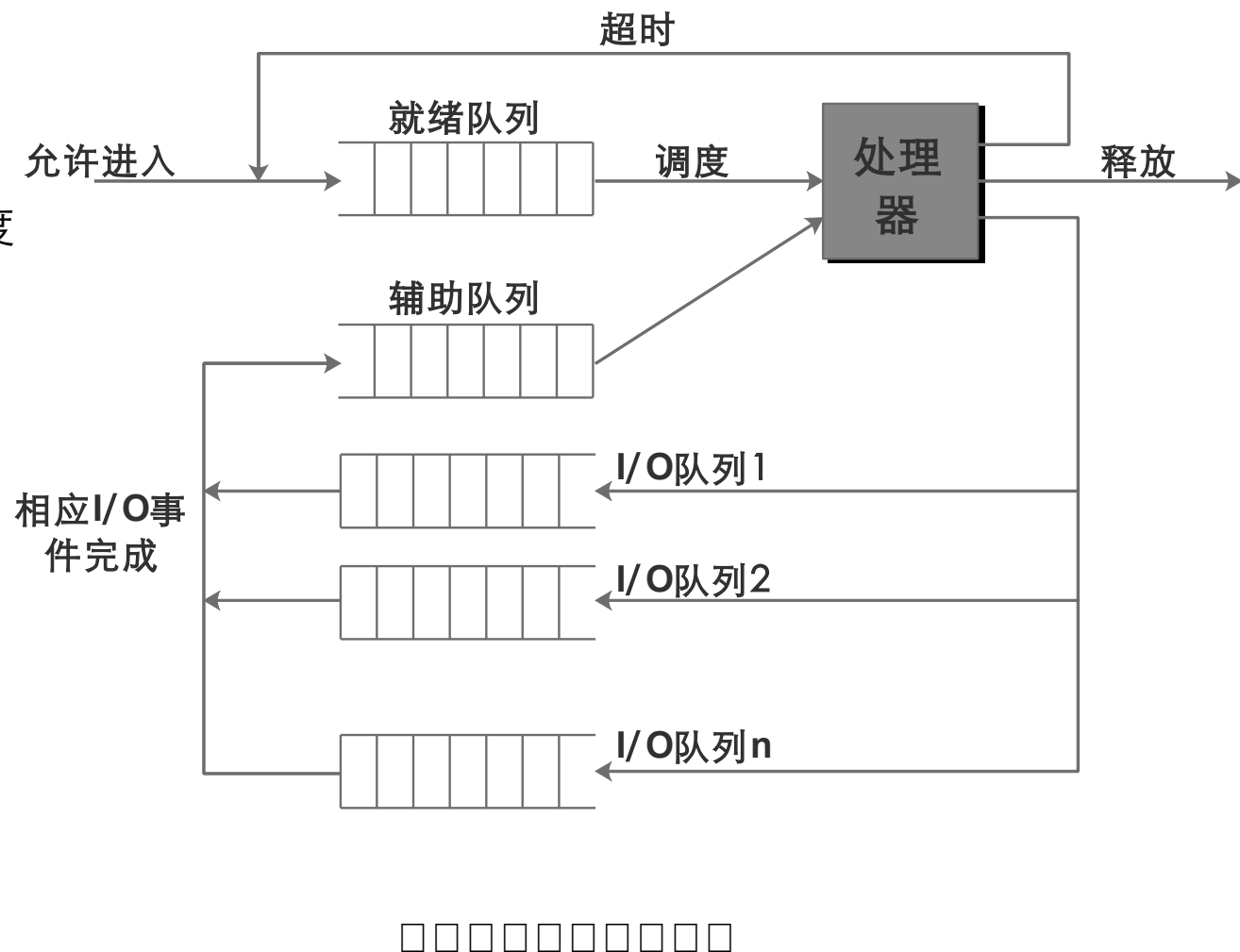
- 定义
 - 抢占式，时间片或进程结束时发生调度
 - 时间片长度影响算法性能
- 评价
 - I/O 密集型进程性能低
- 改进方案
 - 虚拟轮转法 (VRR)

2 轮转 (RR)

- 定义
 - 抢占式，时间片或进程结束时发生调度
 - 时间片长度影响算法性能
- 评价
 - I/O 密集型进程性能低
- 改进方案
 - 虚拟轮转法 (VRR)

2 轮转 (RR)

- 定义
 - 抢占式，时间片或进程结束时发生调度
 - 时间片长度影响算法性能
- 评价
 - I/O 密集型进程性能低
- 改进方案
 - 虚拟轮转法 (VRR)



轮转 (RR)—— 先来先服务 (FCFS)

- 关系
 - 时间片长于运行时间最长的进程时，轮转算法退化为先来先服务算法

□□	A	B	C	D	E
□□□□	0	2	4	6	8
□□□□	3	6	4	5	2
FCFS	1.00	1.17	2.25	2.40	6.00
RR(q = 1)	1.33	2.67	3.25	2.80	3.50
RR(q = 4)	1.00	2.50	1.75	2.80	5.50

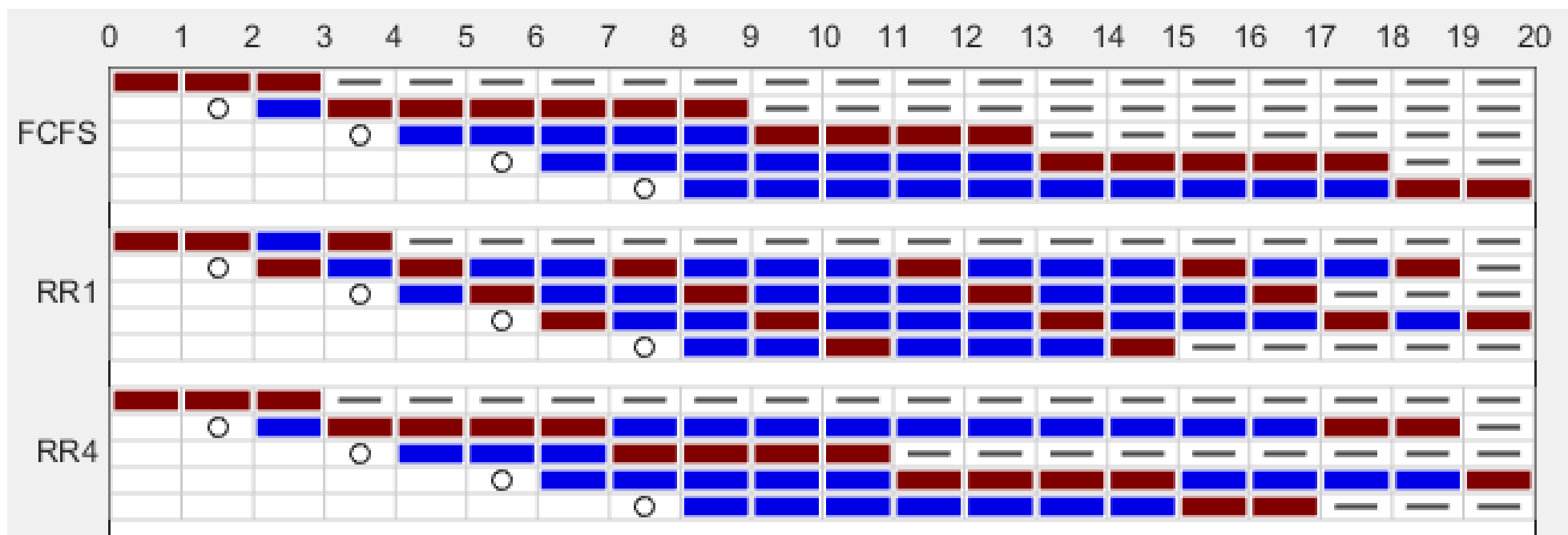
轮转 (RR)—— 先来先服务 (FCFS)

- 关系
 - 时间片长于运行时间最长的进程时，轮转算法退化为先来先服务算法

□□	A	B	C	D	E
□□□□	0	2	4	6	8
□□□□	3	6	4	5	2
FCFS	1.00	1.17	2.25	2.40	6.00
RR(q = 1)	1.33	2.67	3.25	2.80	3.50
RR(q = 4)	1.00	2.50	1.75	2.80	5.50

轮转 (RR)—— 先来先服务 (FCFS)

- 关系
 - 时间片长于运行时间最长的进程时，轮转算法退化为先来先服务算法



3 最短进程优先 (SPN)

- 定义
 - 非抢占式，在进程结束时发生调度
 - $\min[s]$

3 最短进程优先 (SPN)

- 定义
 - 非抢占式，在进程结束时发生调度
 - $\min[s]$

□□□□□□□□□□□□□□

$$S_{n+1} = \frac{1}{n} \sum_{i=1}^n T_i = \frac{1}{n} T_n + \frac{n-1}{n} S_n$$

$$S_{n+1} = \alpha T_n + (1-\alpha) S_n = \alpha T_n + (1-\alpha) \alpha T_{n-1} + \cdots + (1-\alpha)^i \alpha T_{n-i} + \cdots + (1-\alpha)^n S_1$$

3 最短进程优先 (SPN)

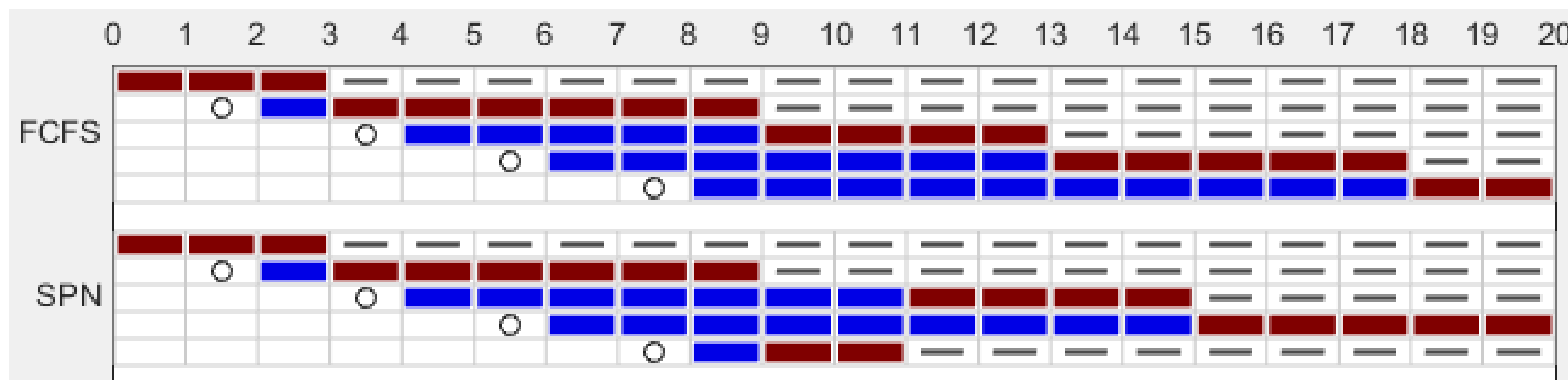
- 定义
 - 非抢占式，在进程结束时发生调度
 - $\min[s]$
- 评价
 - 长进程有可能出现饥饿
 - 缺少抢占机制，当短进程来的次序靠后时，该算法无法发挥作用

3 最短进程优先 (SPN)

- 定义
 - 非抢占式，在进程结束时发生调度
 - $\min[s]$
- 评价
 - 长进程有可能出现饥饿
 - 缺少抢占机制，当短进程来的次序靠后时，该算法无法发挥作用

3 最短进程优先 (SPN)

SPN □□□□□□□ FCFS □□□□□□□□□



3 最短进程优先 (SPN)

- 定义
 - 非抢占式，在进程结束时发生调度
 - $\min[s]$
- 评价
 - 长进程有可能出现饥饿
 - 缺少抢占机制，当短进程来的次序靠后时，该算法无法发挥作用
- 改进方案
 - 最短剩余时间 (SRT) —— 添加抢占机制的 SPN

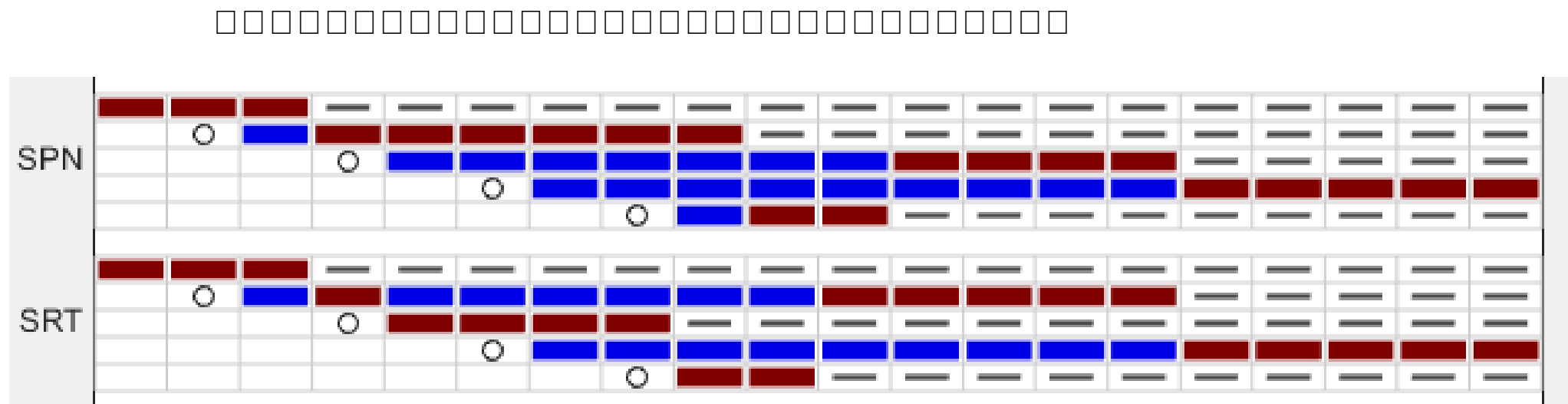
4 最短剩余时间 (SRT)

- 定义
 - 抢占式，在有新进程加入时发生调度
 - $\min[s - e]$ (如何预计每个进程的处理时间)

4 最短剩余时间 (SRT)

- 定义
 - 抢占式，在有新进程加入时发生调度
 - $\min[s - e]$ (如何预计每个进程的处理时间)
- 评价
 - 没有偏向长进程，也不像轮转那样产生额外的中断（只在有必要的时候中断）
 - 长进程有可能出现饥饿
 - 任何长进程都可以被新加入的短进程中断，过于偏向短进程

最短进程优先 (SPN)——最短剩余时间 (SRT)



4 最短剩余时间 (SRT)

- 定义
 - 抢占式，在有新进程加入时发生调度
 - $\min[s - e]$ (如何预计每个进程的处理时间)
- 评价
 - 没有偏向长进程，也不像轮转那样产生额外的中断（只在有必要的时候中断）
 - 长进程有可能出现饥饿
 - 任何长进程都可以被新加入的短进程中断，过于偏向短进程

4 最短剩余时间 (SRT)

- 定义
 - 抢占式，在有新进程加入时发生调度
 - $\min[s - e]$ (如何预计每个进程的处理时间)
- 评价
 - 没有偏向长进程，也不像轮转那样产生额外的中断（只有在有必要的时候中断）
 - 长进程有可能出现饥饿
 - 任何长进程都可以被新加入的短进程中断，过于偏向短进程
- 改进方案
 - 最高响应比优先（HRRN □ ——把长短进程同等对待

5 最高响应比优先 (HRRN)

- 定义
 - 抢占式，在有新进程加入时发生调度
 - R （如何预计每个进程的处理时间）

$$R = \frac{w + s}{s}$$

R □□□□
 w □□□□□□□□
 s □□□□□□□□

5 最高响应比优先 (HRRN)

- 定义
 - 抢占式，发生在有新进程加入的时刻
 - R （如何预计每个进程的处理时间）
- 评价
 - 目前为止同等考虑了长进程和短进程，达到了一个比较好的平衡
 - 如何预计每个进程的处理时间

5 最高响应比优先 (HRRN)

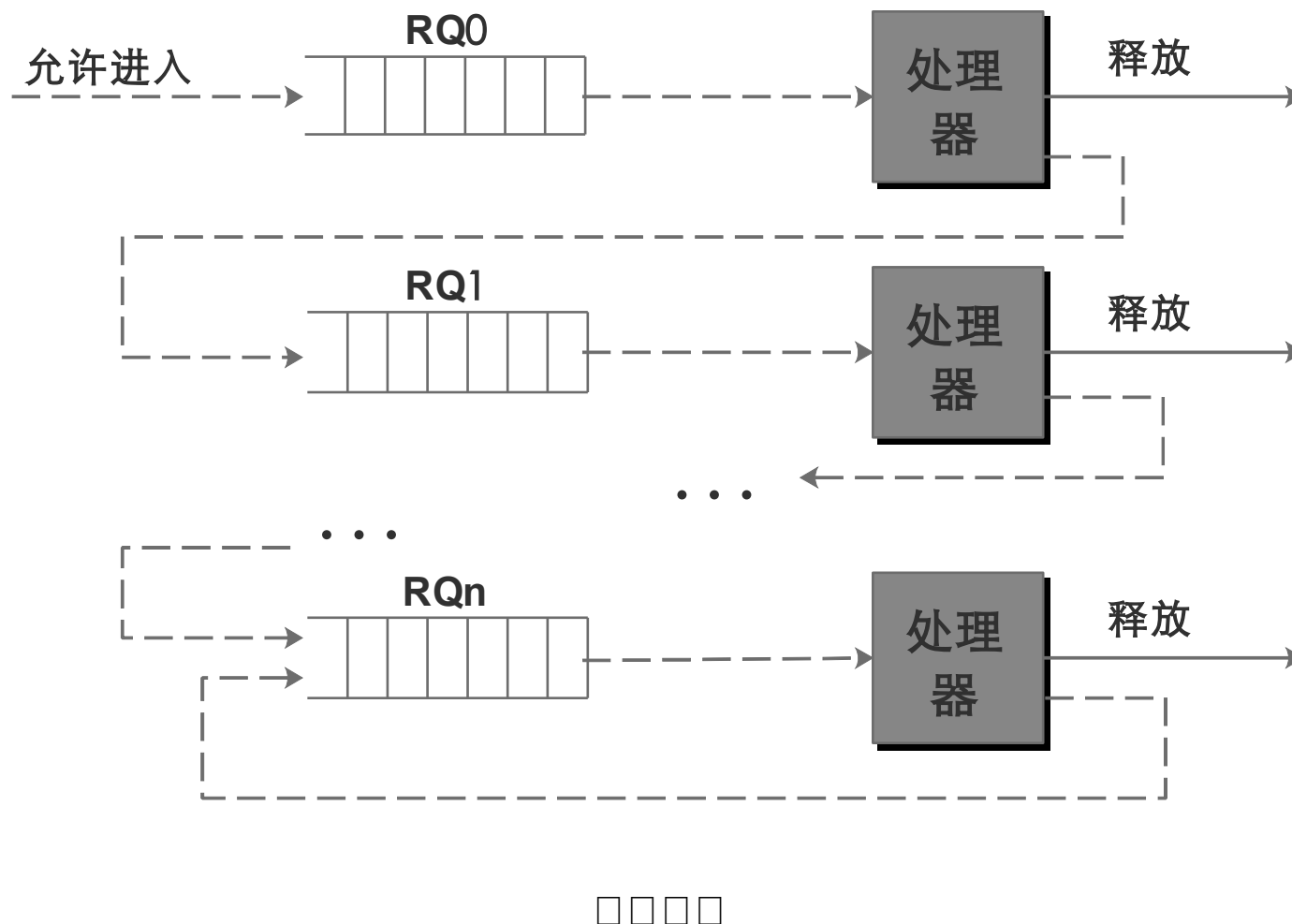
- 定义
 - 抢占式，发生在有新进程加入的时刻
 - R （如何预计每个进程的处理时间）
- 评价
 - 目前为止同等考虑了长进程和短进程，达到了一个比较好的平衡
 - 如何预计每个进程的处理时间
- 改进方案
 - 反馈（Feedback）

6 反馈 (Feedback)

- 定义
 - 抢占式
 - 按时间片
 - 动态优先级机制
 - FCFS

6 反馈 (Feedback)

- 定义
 - 抢占式
 - 按时间片
 - 动态优先级机制
 - FCFS



6 反馈 (Feedback)

- 定义
 - 抢占式
 - 按时间片
 - 动态优先级机制
 - FCFS
- 优化
 - 时间片长度
 - 设置最长等待时间

- □□□□
 - □□□ & □□□
 - □□□□□□ & I/O □□□

总结

谢谢

唐洋洋

2015-3-31