Operating Systems

Chapter 9 Uniprocessor Scheduling

Agenda

- 9.1 Type of Processor Scheduling
- 9.2 Scheduling Algorithms
- 9.3 Summary

9.1 Type of Processor Scheduling

- 9.1.0 Overview
- 9.1.1 Long-term Scheduling
- 9.1.2 Medium-term Scheduling
- 9.1.3 Short-term Scheduling

9.1 Type of Processor Scheduling(1/10)

- Why ： Aim of Scheduling(调度目标)
  - Assign processes to be executed by the processor(s)(处理器分配)
    - Response time (响应时间)
    - Throughput (吞吐率)
    - Processor efficiency (处理器效率)

9.1 Type of Processor Scheduling(2/10)

- When to Scheduling
  - Nonpreemptive 主动放弃 CPU
    - The running thread terminates.
    - The running thread is blocked.
  - Preemptive
    - The running thread is running out of time slice.
    - Interrupt service is completed.（ blocked to ready ）
    - New thread arrives

9.1 Type of Processor Scheduling(3/10)

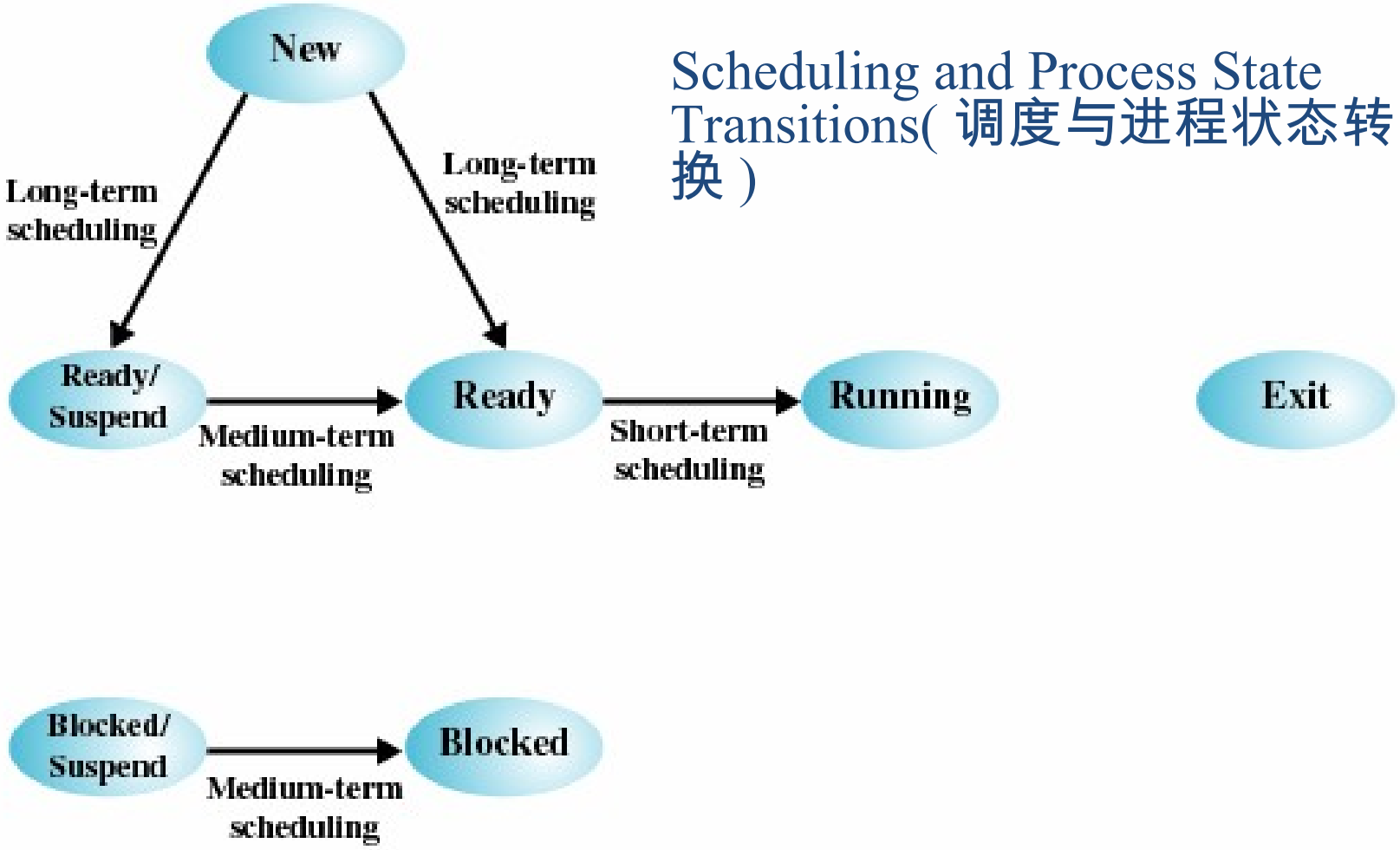
- How to Scheduling
  - Save context in PCB
  - Select a ready process
    - Criteria/evaluate
  - and load ready process's context (switch)

9.1 Type of Processor Scheduling(4/10)

Types of Scheduling

Long-term scheduling	The decision to add to the pool of processes to be executed
Medium-term scheduling	The decision to add to the number of processes that are partially or fully in main memory
Short-term scheduling	The decision as to which available process will be executed by the processor
I/O scheduling	The decision as to which process's pending I/O request shall be handled by an available I/O device

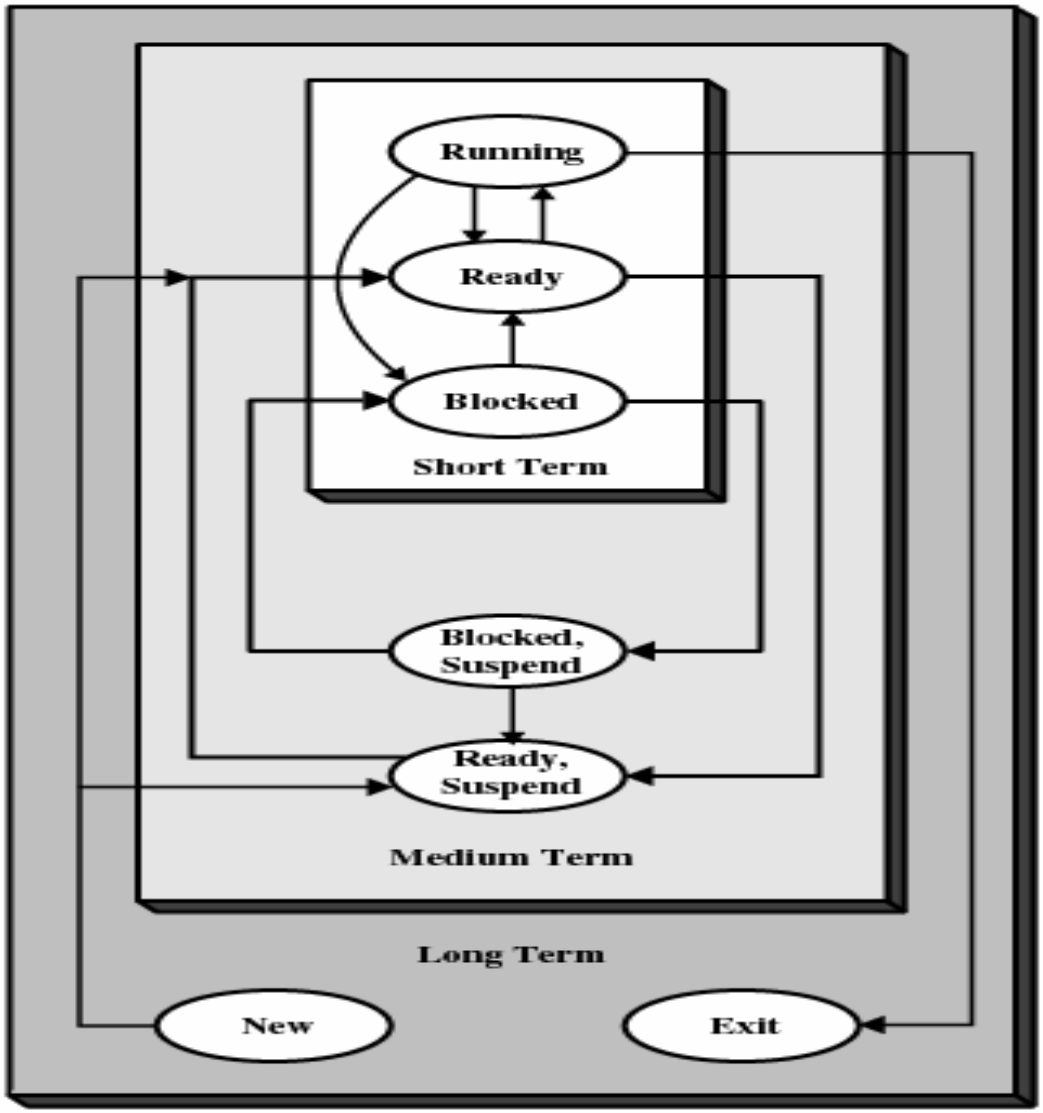
9.1 Type of Processor Scheduling(5/10)



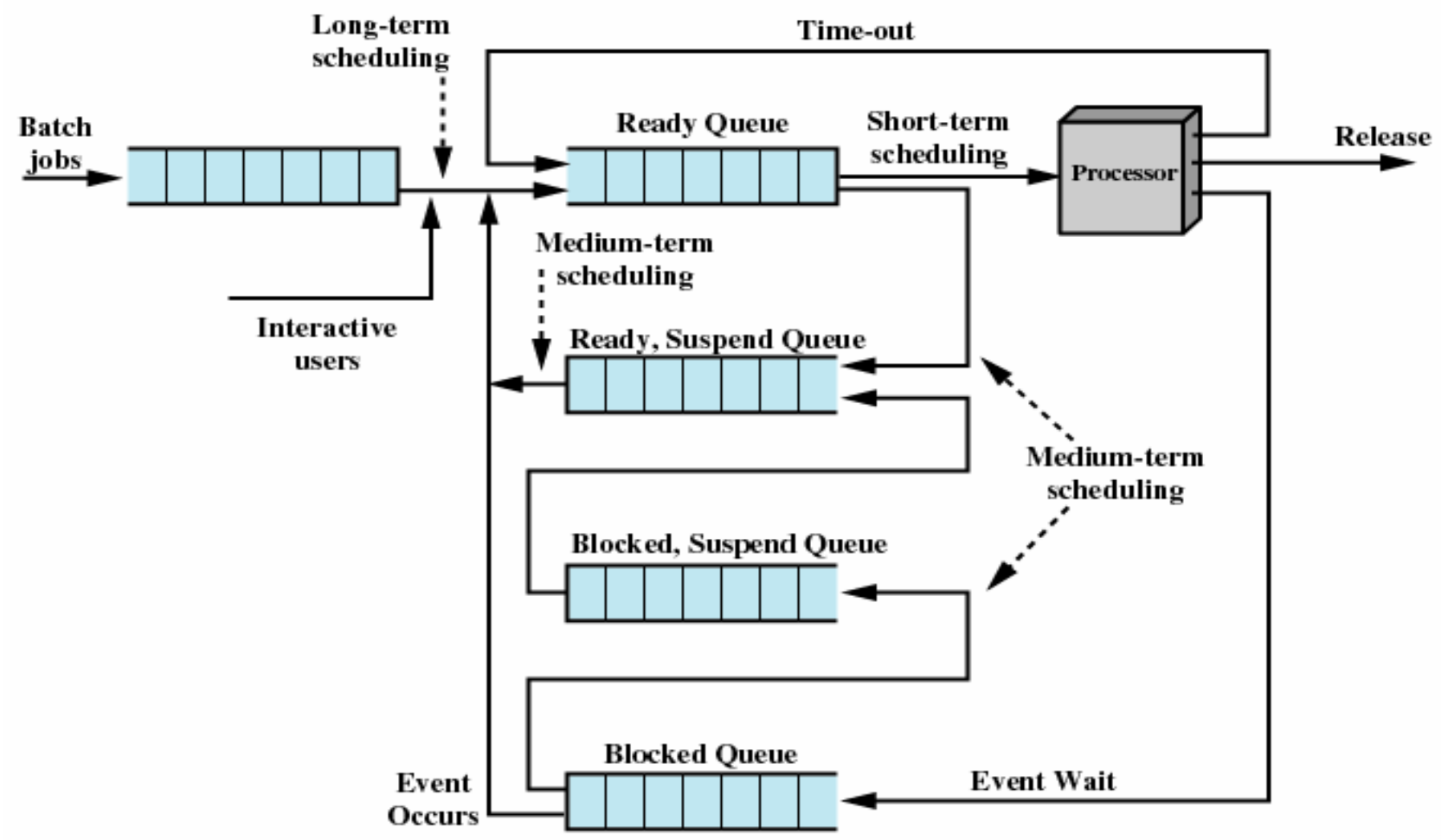
Scheduling and Process State Transitions(调度与进程状态转换)

9.1 Type of Processor Scheduling(6/10)

Levels of Scheduling (调度层次)



9.1 Type of Processor Scheduling(7/10)



Queuing Diagram for Scheduling 调度队列图

9.1 Type of Processor Scheduling(8/10)

- Long-Term Scheduling (长程调度)
  - Determines which programs are admitted to the system for processing
  - Controls the degree of multiprogramming
  - More processes, smaller percentage of time each process is executed
  - 增加新进程
    - 当一个进程终止
    - 当 CPU 空闲率超过某个阈值

9.1 Type of Processor Scheduling(9/10)

- Medium-Term Scheduling (中程调度)
  - Part of the swapping function 交换功能的一部分
  - Based on the need to manage the degree of multiprogramming (thrashing 抖动)

9.1 Type of Processor Scheduling(10/10)

- Short-Term Scheduling (短程调度)
  - Known as the dispatcher(分派器)
  - Executes most frequently
  - Invoked when an event occurs
    - Clock interrupts(时钟中断)
    - I/O interrupts(I/O 中断)
    - Operating system calls(操作系统调用)
    - Signals(信号)

Agenda

- 9.1 Type of Processor Scheduling
- 9.2 Scheduling Algorithms
- 9.3 Summary

9.2 Scheduling Algorithms

- 9.2.1 Short-term Criteria
- 9.2.2 The Use of Priorities
- 9.2.3 Alternative of Scheduling Policies
- 9.2.4 Fair-Share Scheduling

9.2.1 Short-term Criteria(1/4)

- Short-Tem Scheduling Criteria(短程调度准则)
  - User-oriented(面向用户) &&Performance-related(性能相关)
    - Behavior of OS as perceived by the user (用户感知到的系统行为)
      - **Turnaround time 周转时间** This is the interval of time between the submission of a process and its completion.
      - **Response Time(响应时间):**Elapsed time between the submission of a request until there is output.
      - **Deadlines**(终止时间) : maximizing the percentage of deadlines met

9.2.1 Short-term Criteria(2/4)

- Short-Tem Scheduling Criteria(短程调度准则)
  - User-oriented(面向用户)&& other
    - Qualitative 定性的 and Unmeasurable
      - predictability(可预测性)

9.2.1 Short-term Criteria(3/4)

- Short-Tem Scheduling Criteria(短程调度准则)
  - System-oriented(面向系统) &&Performance-related(性能相关)
    - **Processor utilization** ： Effective and efficient utilization of the processor(处理器的使用效率)
    - **Throughput**(吞吐量) ：单位时间完成进程数

9.2.1 Short-term Criteria(4/4)

- Short-Tem Scheduling Criteria(短程调度准则)
  - System-oriented(面向系统) &&other
    - **Fairness** ： no process starvation
    - **Enforcing priorities** ： scheduling policy favor higher-priority processes.
    - **Balancing resources(**also for medium-term and long-term **)** ： keep the resources busy. Processes requests underutilize stressed resources should be favored.

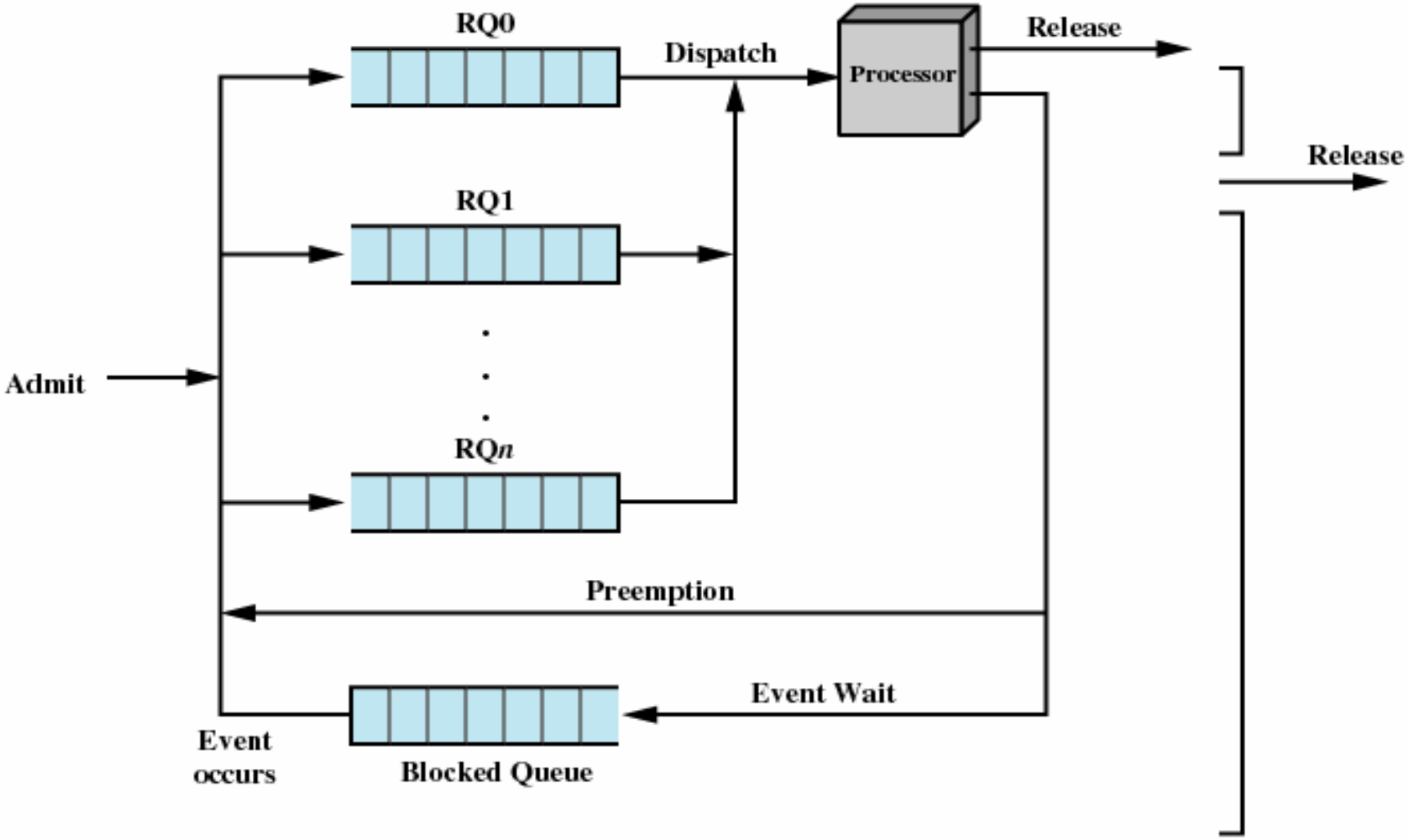
9.2 Scheduling Algorithms

- 9.2.1 Short-term Criteria
- 9.2.2 The Use of Priorities
- 9.2.3 Alternative of Scheduling Policies
- 9.2.4 Fair-Share Scheduling

9.2.2 The Use of Priorities(1/2)

- Scheduler will always choose a process of higher priority over one of lower priority
- Have multiple ready queues to represent each level of priority
- Lower-priority may suffer starvation
  - Allow a process to change its priority based on its age or execution history

9.2.2 The Use of Priorities(2/2)



Priorities Queuing

9.2 Scheduling Algorithms

- 9.2.1 Short-term Criteria
- 9.2.2 The Use of Priorities
- 9.2.3 Alternative of Scheduling Policies
- 9.2.4 Fair-Share Scheduling

9.2.3 Alternative of Scheduling Policies

- 9.2.3.1 Terms 相关术语
- 9.2.3.2 FCFS 先来先服务
- 9.2.3.3 Round-Robin (RR 轮转)
- 9.2.3.4 Shortest Process Next (SPN 最短进程优先)
- 9.2.3.5 Shortest Remaining Time (SRT 最短剩余时间)
- 9.2.3.6 Highest Response Ratio Next (HRRN 最高响应比优先)
- 9.2.3.7 Feedback (反馈)

9.2.3.1 Terms (1/2) 相关术语

- Response time(响应时间)
- 服务时间 Ts ：进程单独执行需要的时间
- Throughput (吞吐量)
- Turnaround time(周转时间) Tr 或驻留时间
  - Tr= 完成时间 - 到达时间
  - 受并发的其它进程执行时间的影响
- Normalized turnaround time(归一化周转时间) =Tr/Ts

9.2.3.1 Terms (2/2) 相关术语

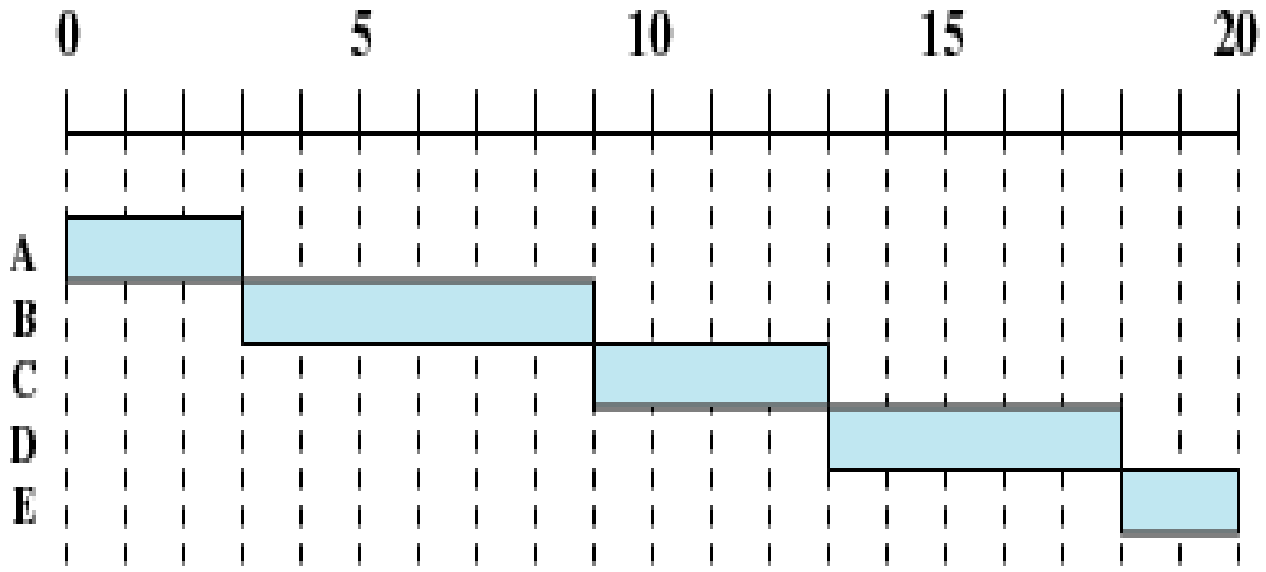
- Predictability(可预测性)
- Selection function(选择函数，确定在就绪进程中选择哪一个进程在下一次执行)
- Decision mode(决策模式，选择函数被执行瞬间的处理方式)
  - Preemptive (抢占，当前正在执行的进程可能被操作系统中断，并转移到就绪态)
  - Nonpreemptive(非抢占，一旦处于运行态就不断执行直到终止或者被阻塞)
    - 通常在上一个时间片结束时 / 也可在新进程到达时

Process Scheduling Example

Process	Arrival Time	Service Time
A	0	3
B	2	6
C	4	4
D	6	5
E	8	2

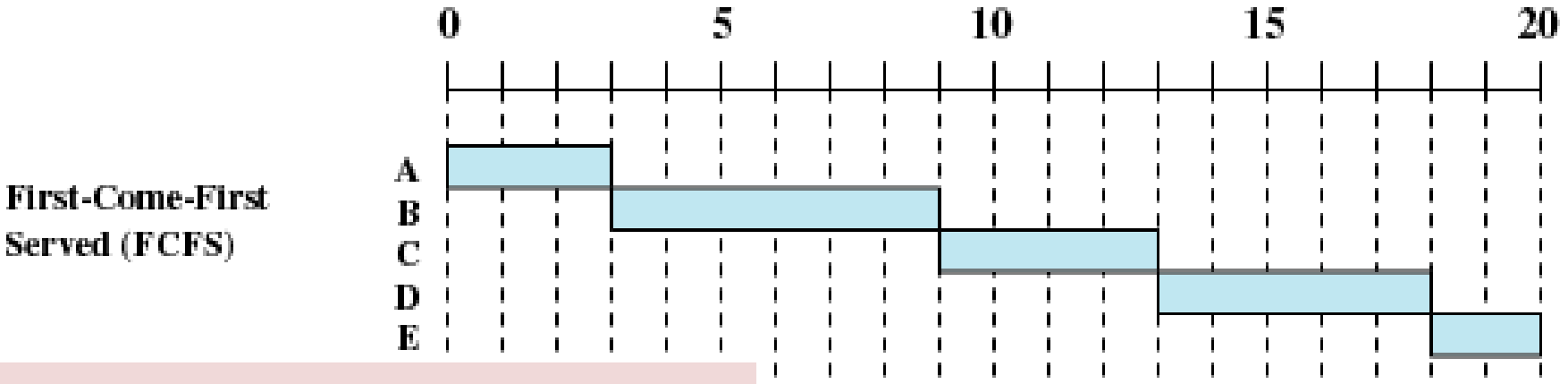
9.2.3.2First-Come-First-Served(1/3) (FCFS 先来先服务)

Process	Arrival Time	Service Time
A	0	3
B	2	6
C	4	4
D	6	5
E	8	2



- Decision mode: ***Nonpreemptive 非抢占***
- Dispatch time: the current process ceases to execute 结束切换
- Method: The oldest process in the Ready queue is selected

9.2.3.2First-Come-First-Served(2/3) (FCFS 先来先服务)



Tr= Finish time – arrival time

Process	A	B	C	D	E	
Arrival Time	0	2	4	6	8	
Service Time (T_s)	3	6	4	5	2	Mean
FCFS						
Finish Time	3	9	13	18	20	
Turnaround Time (T_r)	3	7	9	12	12	8.60
T_r/T_s	1.00	1.17	2.25	2.40	6.00	2.56

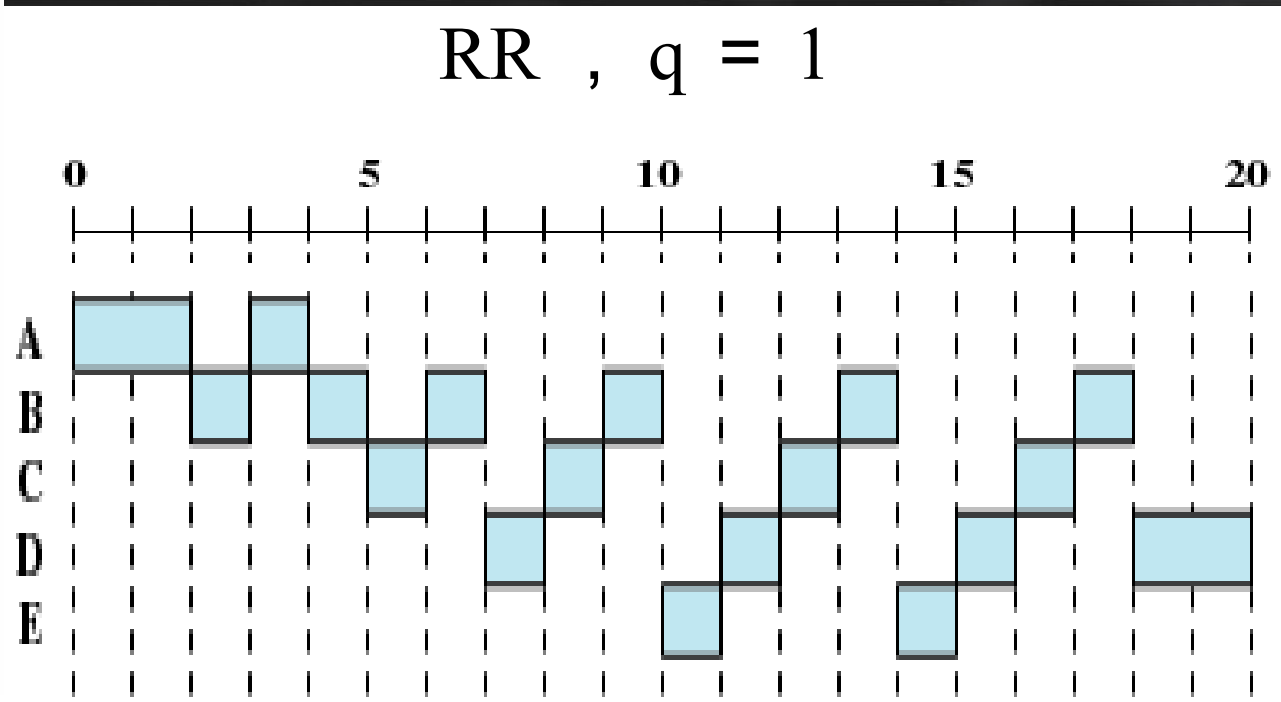
前 5 项的平均

9.2.3.2First-Come-First-Served(3/3) (FCFS 先来先服务)

- Disadvantageous to short process
  - A short process may have to wait a very long time before it can execute
- Favors CPU-bound processes(适合处理器密集型进程)
  - I/O processes have to wait until CPU-bound process completes
  - combined with a priority scheme to provide an effective scheduler.

9.2.3.3 Round-Robin (1/6)(RR 轮转)

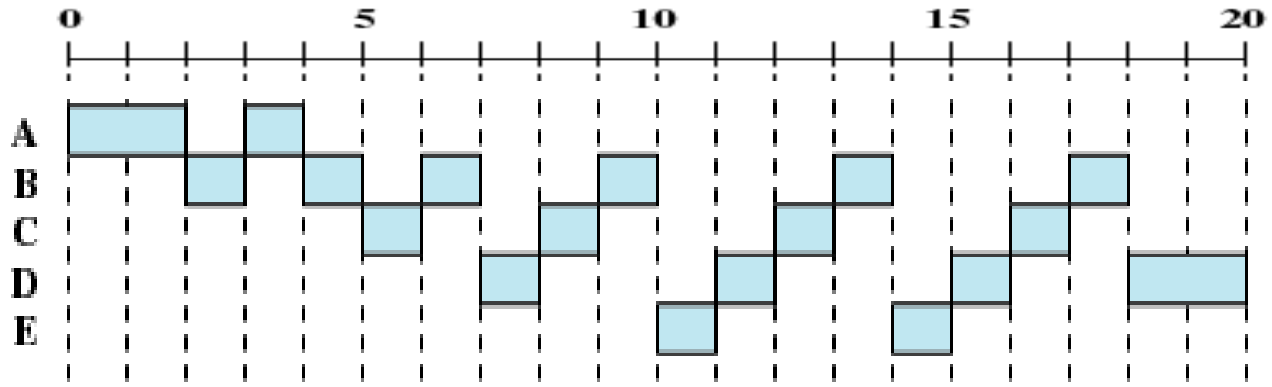
Process	Arrival Time	Service Time
A	0	3
B	2	6
C	4	4
D	6	5
E	8	2



- Decision mode: ***preemption***
- Dispatch time: period q based on a clock
- Method ： select next process based on FCFS

9.2.3.3 Round-Robin (2/6)(RR 轮转)

Round-Robin
(RR), $q = 1$



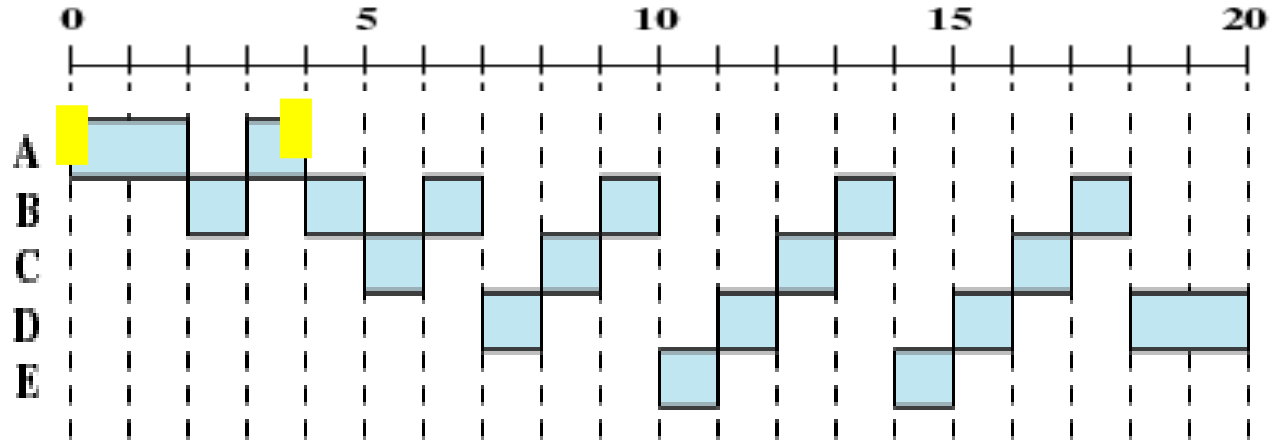
时刻	运行
0	A
1	A
2	B
3	A
4	B
5	C
6	B
7	D
8	C
9	B

(头) 就绪队列 (尾)			
A			
B			
C			
B			
D	C		
C	B		
B	E	D	
E	D	C	

时刻	运行
10	E
11	D
12	C
13	B
14	E
15	D
16	C
17	B
18	D
19	D

(头) 就绪队列 (尾)			
D	C	B	
C	B	E	
B	E	D	
E	D	C	
D	C	B	
C	B		
B	D		
D			

9.2.3.3 Round-Robin (3/6)(RR 轮转)



Round-Robin (RR), $q = 1$

Process	A	B	C	D	E	
Arrival Time	0	2	4	6	8	
Service Time (T_s)	3	6	4	5	2	Mean

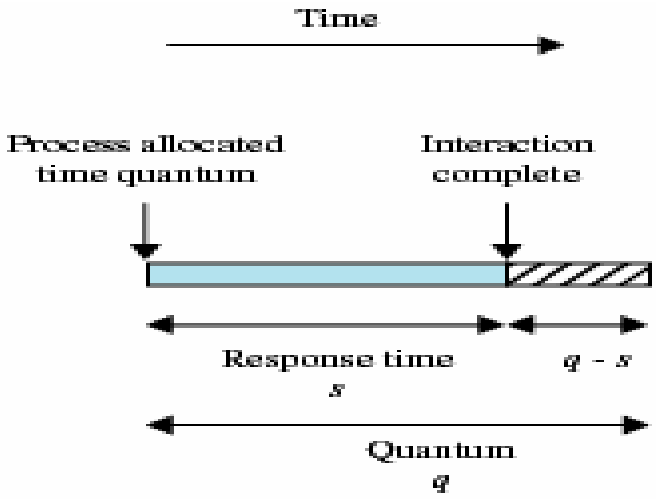
RR $q = 1$

Finish Time	4	18	17	20	15	
Turnaround Time (T_r)	4	16	13	14	7	10.80
T_r/T_s	1.33	2.67	3.25	2.80	3.50	2.71

9.2.3.3 Round-Robin (4/6)(RR 轮转)

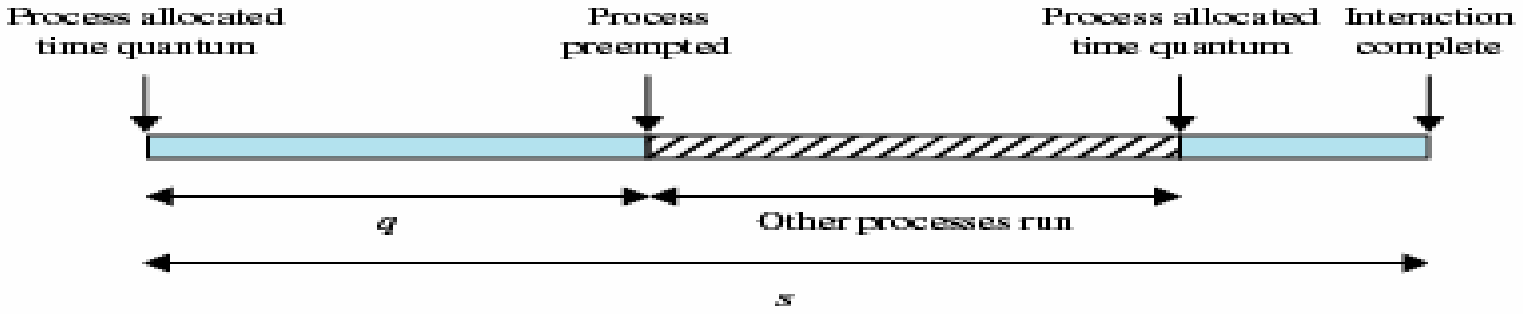
- Clock interrupt is generated at periodic intervals
- When an interrupt occurs, the currently running process is placed in the ready queue
  - **Next ready job is selected based on FCFS**
- Known as time slicing (时间片)
  - Size?
- Disadvantageous to I/O-bound processes .Thus, processor-bound processes receive major portion of CPU time.(better than FCFS)

9.2.3.3 Round-Robin (5/6)(RR 轮转)



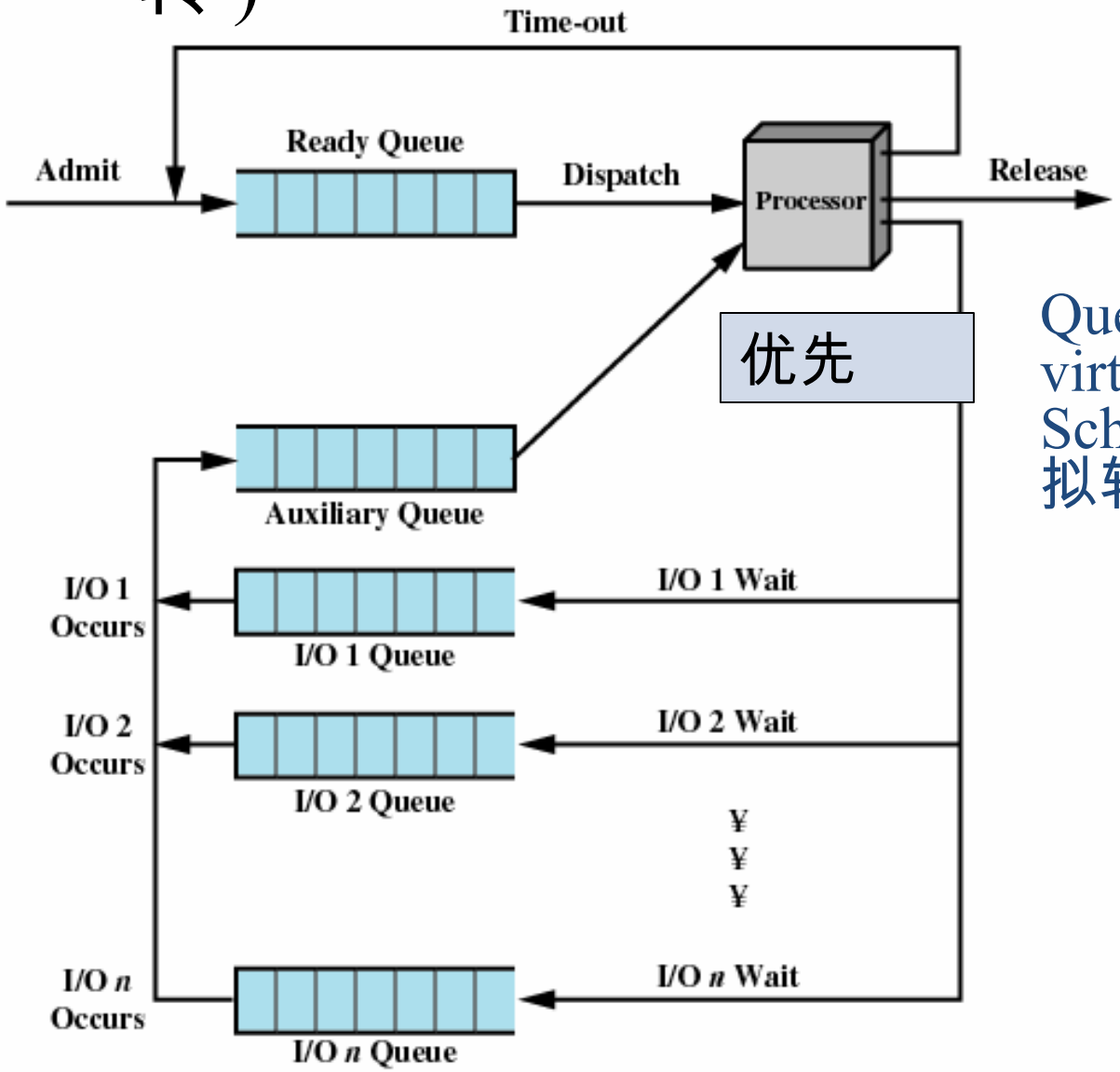
Effect of Size of Preemption Time Quantum (时间片大小的影响) : 略大于一次典型交互时间为好

(a) Time quantum greater than typical interaction



(b) Time quantum less than typical interaction

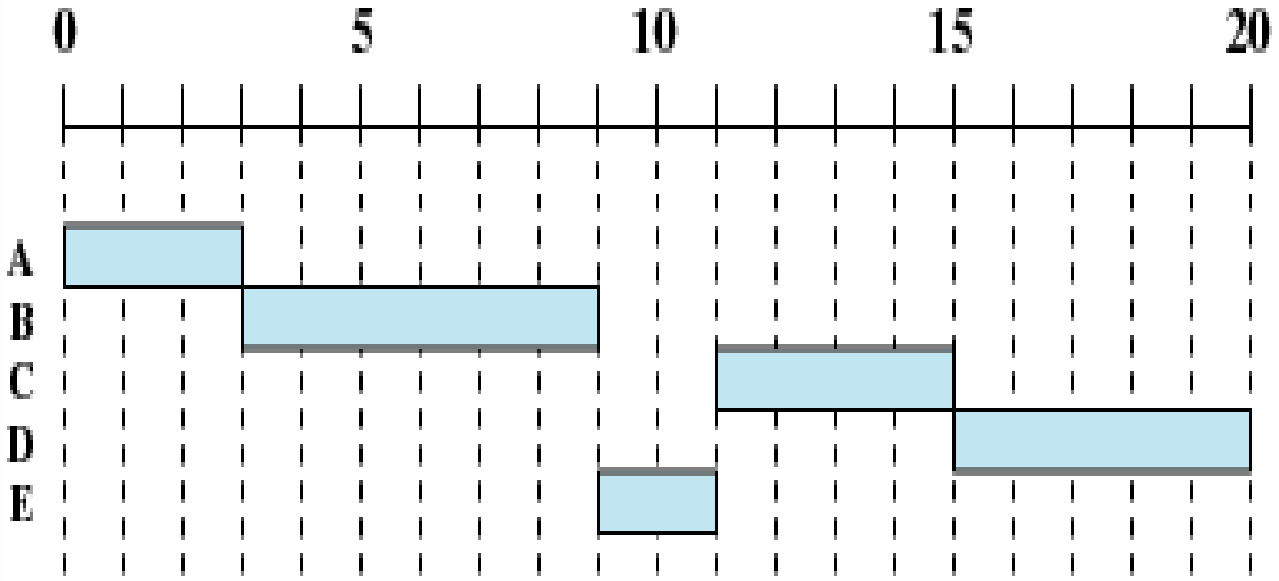
9.2.3.3 Round-Robin (6/6)(RR 轮转)



Queuing Diagram for virtual Round-Robin Scheduler (VRR ，虚拟轮转法)

9.2.3.4Shortest Process Next(1/2) (SPN 最短进程优先)

Process	Arrival Time	Service Time
A	0	3
B	2	6
C	4	4
D	6	5
E	8	2



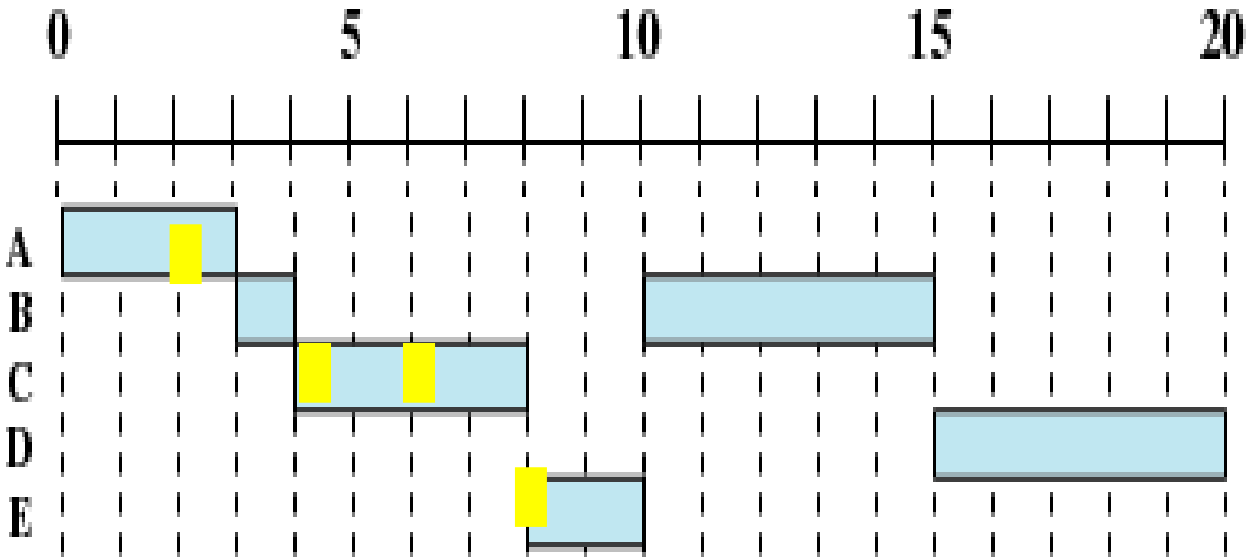
- Decision mode: ***Nonpreemptive***
- Dispatch time: the current process ceases to execute
- Method: Process with shortest expected processing time is selected next

9.2.3.4Shortest Process Next(2/2) (SPN 最短进程优先)

- Predictability of longer processes is reduced(长进程的可预测性降低了)
- Possibility of starvation(饥饿) for longer processes

9.2.3.5Shortest Remaining Time(1/1) (SRT 最短剩余时间)

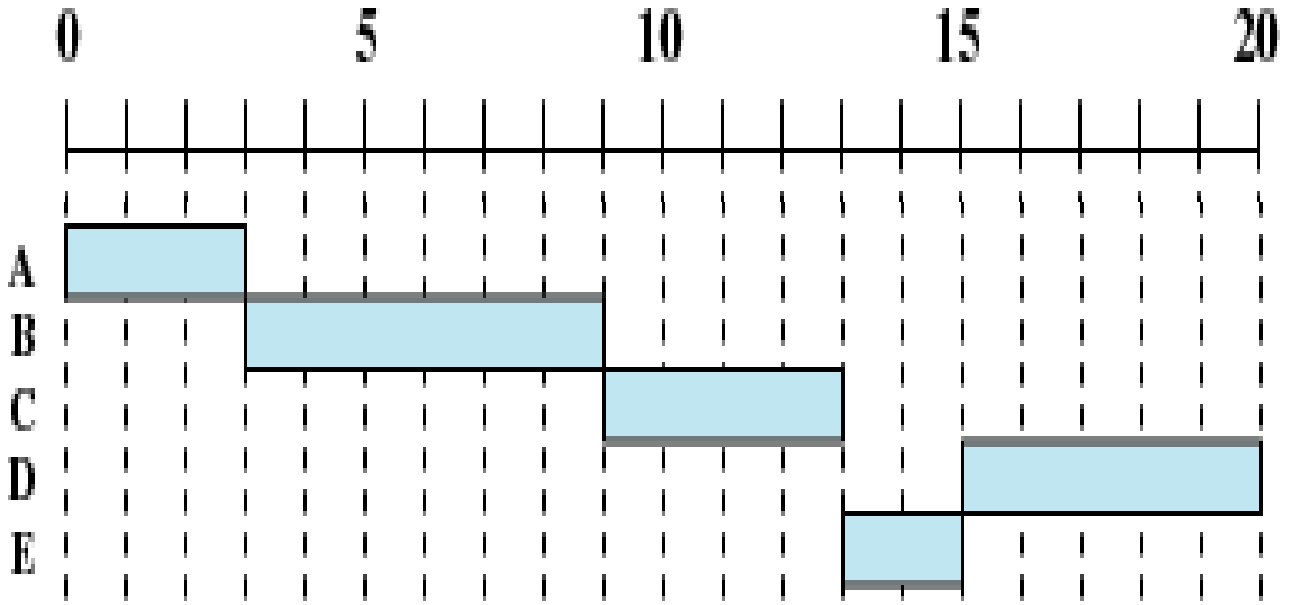
Process	Arrival Time	Service Time
A	0	3
B	2	6
C	4	4
D	6	5
E	8	2



- Decision mode: ***Preemptive*** version of shortest process next
- Dispatch time: decision is made when a new process arrives
- Method: Process with shortest remaining time is selected next

9.2.3.6Highest Response Ratio Next (1/2)(HRRN 最高响应比优先)

Process	Arrival Time	Service Time
A	0	3
B	2	6
C	4	4
D	6	5
E	8	2

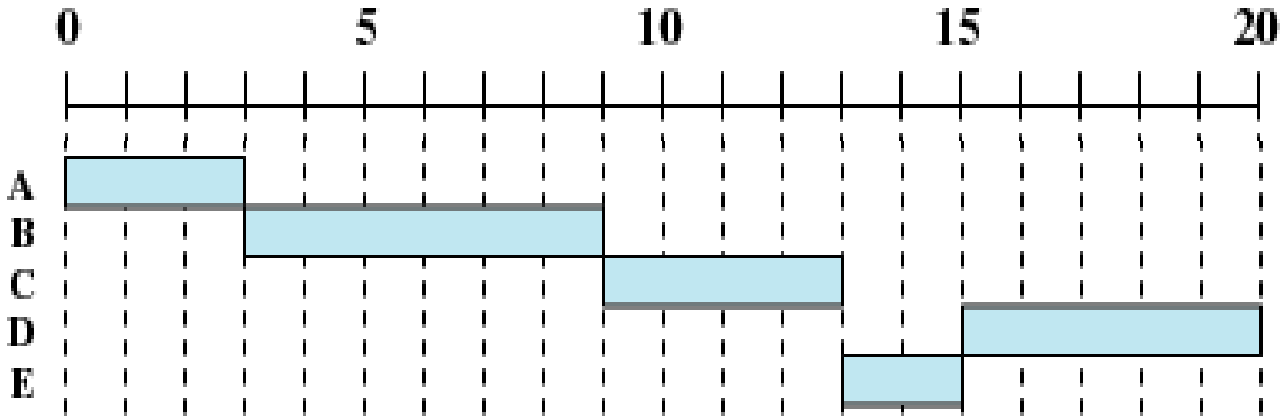


- Decision mode: ***Nonpreemptive***
- Dispatch time: the current process ceases to execute
- Method: Choose next process with the greatest response ratio

$$R = \frac{\text{time spent waiting} + \text{expected service time}}{\text{expected service time}}$$

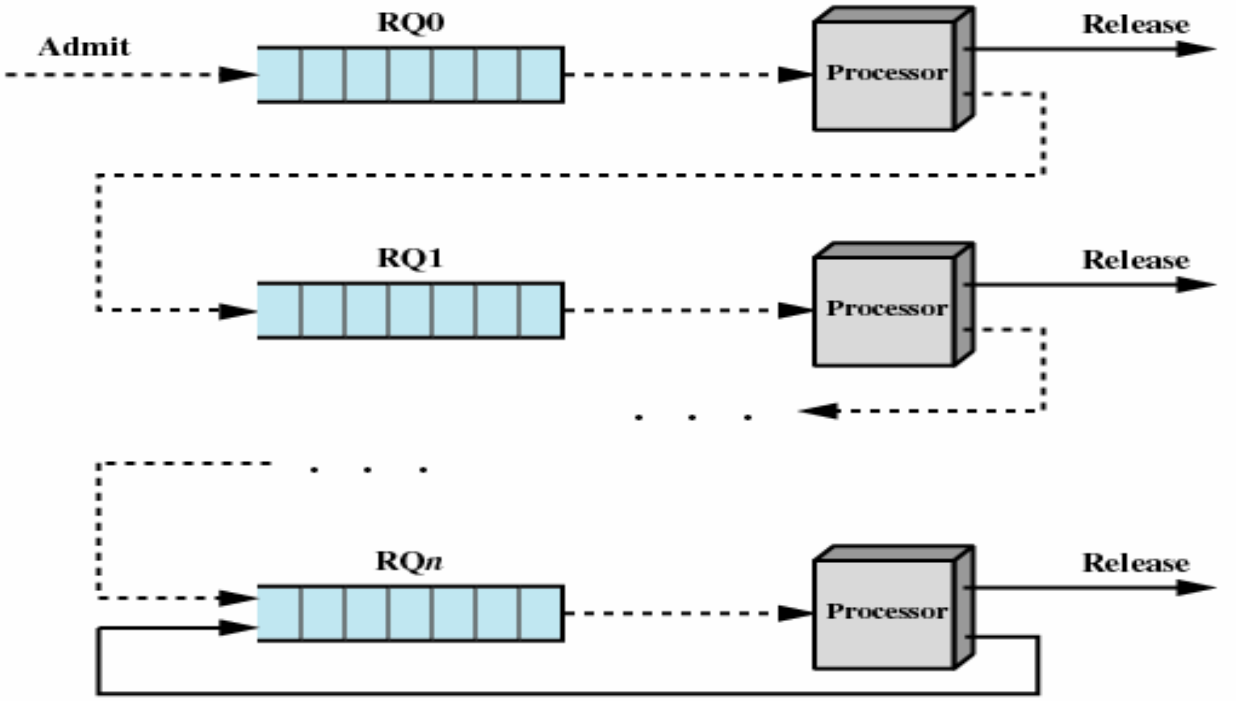
9.2.3.6Highest Response Ratio Next (2/2)(HRRN 最高响应比优先)

Highest Response Ratio Next (HRRN)



时刻	A	B	C	D	E
3	N	(1+4)/4	N	N	N
9	N	N	(9-4+4)/4	(9-6+5)/5	(9-8+2)/2
13	N	N	N	(13-6+5)/5	(13-8+2)/2

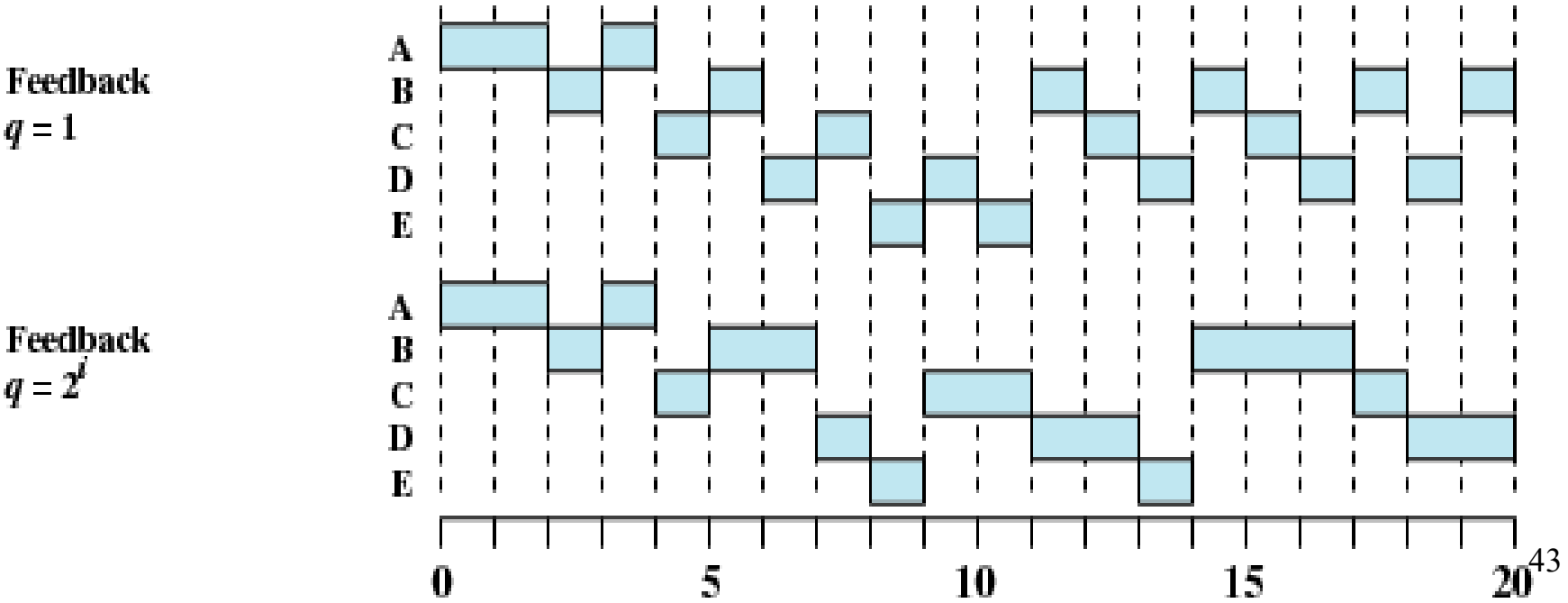
9.2.3.7Feedback(1/4) (反馈)



- Feedback doesn't know process service time while SPN/SRT/HRRN need
- ***Preemptive***
- Penalize(惩罚) jobs that have been running longer

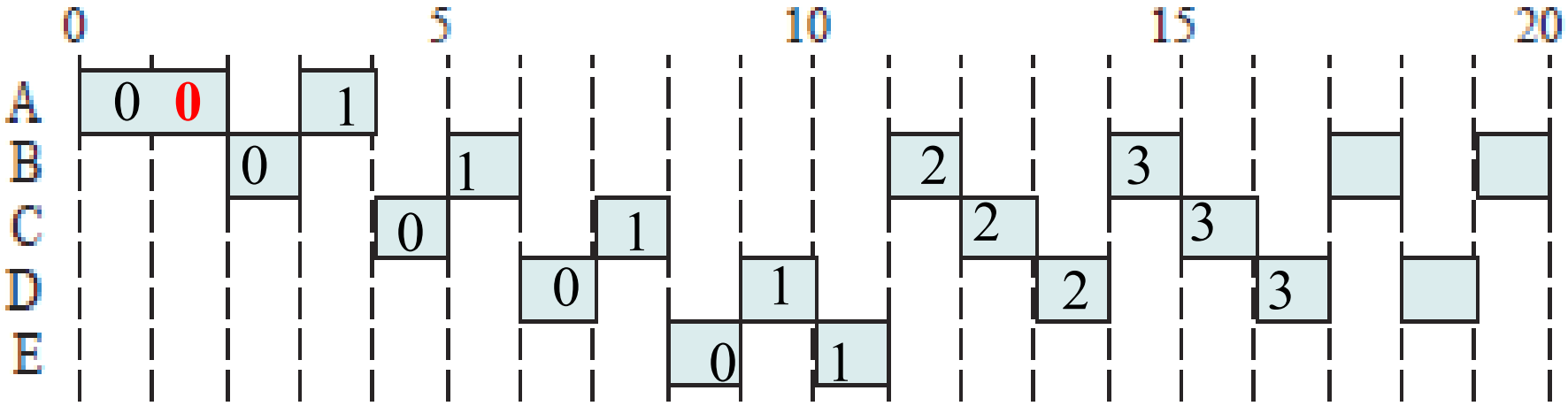
9.2.3.7Feedback(2/4) (反馈)

Process	Arrival Time	Service Time
A	0	3
B	2	6
C	4	4
D	6	5
E	8	2



9.2.3.7Feedback(3/4)（反馈）

Process	A	B	C	D	E
Arrival Time	0	2	4	6	8
Service Time (T_s)	3	6	4	5	2



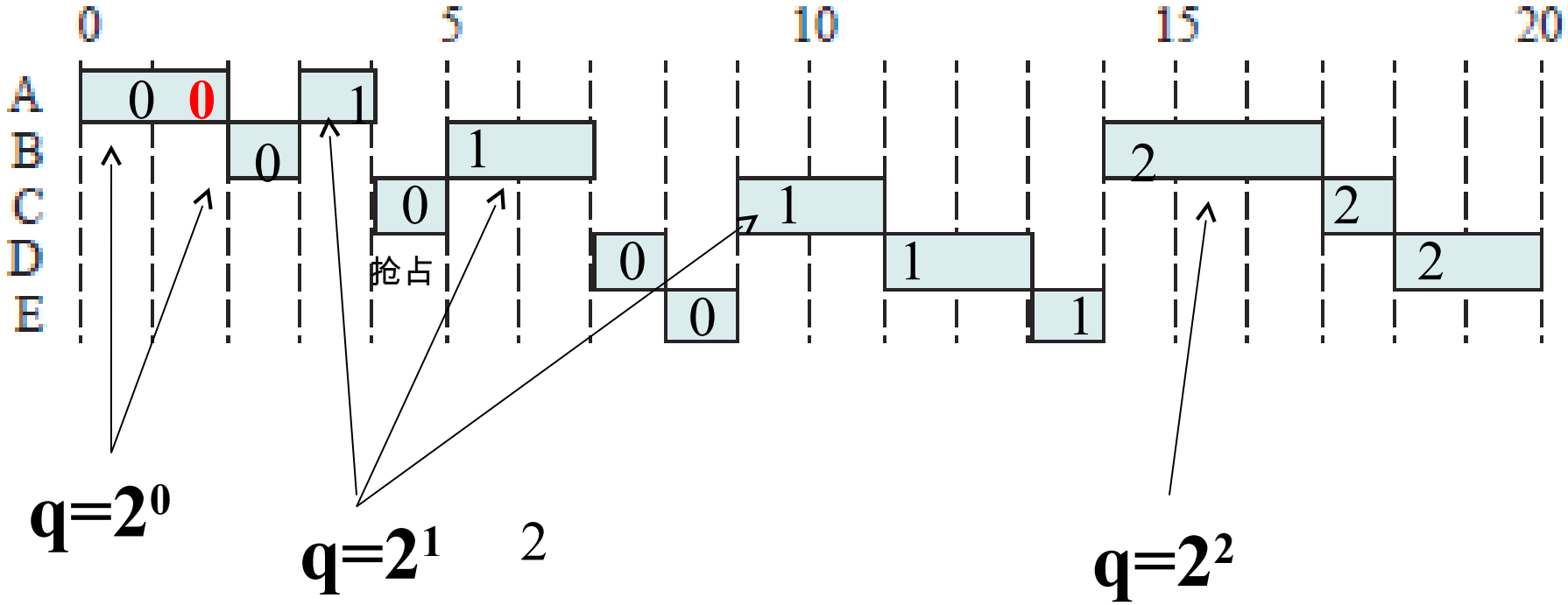
q=1

分配时间片	时刻	11			
1	RQ0				
1	RQ1				
1	RQ2	Brun	C	D	

9.2.3.7Feedback(4/4)（反馈）

$q=2^i$

Process	A	B	C	D	E
Arrival Time	0	2	4	6	8
Service Time (T_s)	3	6	4	5	2



9.2.3.8 comparision (1/2)

A Comparison of Scheduling policies

	Process	A	B	C	D	E	
	Arrival Time	0	2	4	6	8	
	Service Time (T_s)	3	6	4	5	2	Mean
FCFS	Finish Time	3	9	13	18	20	
	Turnaround Time (T_r)	3	7	9	12	12	8.60
	T_r/T_s	1.00	1.17	2.25	2.40	6.00	2.56
RR $q = 1$	Finish Time	4	18	17	20	15	
	Turnaround Time (T_r)	4	16	13	14	7	10.80
	T_r/T_s	1.33	2.67	3.25	2.80	3.50	2.71
RR $q = 4$	Finish Time	3	17	11	20	19	
	Turnaround Time (T_r)	3	15	7	14	11	10.00
	T_r/T_s	1.00	2.5	1.75	2.80	5.50	2.71
SPN	Finish Time	3	9	15	20	11	
	Turnaround Time (T_r)	3	7	11	14	3	7.60
	T_r/T_s	1.00	1.17	2.75	2.80	1.50	1.84
SRT	Finish Time	3	15	8	20	10	
	Turnaround Time (T_r)	3	13	4	14	2	7.20
	T_r/T_s	1.00	2.17	1.00	2.80	1.00	1.59
HRRN	Finish Time	3	9	13	20	15	
	Turnaround Time (T_r)	3	7	9	14	7	8.00
	T_r/T_s	1.00	1.17	2.25	2.80	3.5	2.14
FB $q = 1$	Finish Time	4	20	16	19	11	
	Turnaround Time (T_r)	4	18	12	13	3	10.00
	T_r/T_s	1.33	3.00	3.00	2.60	1.5	2.29
FB $q = 2^i$	Finish Time	4	17	18	20	14	
	Turnaround Time (T_r)	4	15	14	14	6	10.60
	T_f/T_s	1.33	2.50	3.50	2.80	3.00	2.63

9.2.3.8 comparision (2/2)

	Selection Function	Decision Mode	Throughput	Response Time	Overhead	Effect on Processes	Starvation
FCFS	max[w]	Nonpreemptive	Not emphasized	May be high, especially if there is a large variance in process execution times	Minimum	Penalizes short processes; penalizes I/O bound processes	No
Round Robin	constant	Preemptive (at time quantum)	May be low if quantum is too small	Provides good response time for short processes	Minimum	Fair treatment	No
SPN	min[s]	Nonpreemptive	High	Provides good response time for short processes	Can be high	Penalizes long processes	Possible
SRT	min[$s - e$]	Preemptive (at arrival)	High	Provides good response time	Can be high	Penalizes long processes	Possible
HRRN	max $\left(\frac{w+s}{s}\right)$	Nonpreemptive	High	Provides good response time	Can be high	Good balance	No
Feedback	(see text)	Preemptive (at time quantum)	Not emphasized	Not emphasized	Can be high	May favor I/O bound processes	Possible

w = time spent waiting
e = time spent in execution so far
s = total service time required by the process, including e

Characteristics of Various Scheduling Policies

拓展： OpenEuler 多处理器调度 (1/3)

- openEuler 调度策略
  - 基于优先级 0~99
    - 为每个优先级维护一个进程链表，从最高优先级选择一个进程占用 CPU
    - 用位图表示某个优先级是否有进程排队
  - 要求每个进程都有一个调度策略
    - 若选中的进程使用 FIFO ，则该进程一直占用 CPU ，直到结束 / 被高优先级抢占 / 自己阻塞
    - 若选中的进程使用 RR ，则该进程直到时间片用完 / 提前结束 / 自己阻塞

拓展： OpenEuler 多处理器调度 (2/3)

- openEuler 调度策略
  - 核心基础调度策略：标准轮流分时调度策略，其采用的是 CFS(Completely Fair Scheduler ，完全公平调度)
  - 为满足不同应用场景，对多种类别基础调度（限期，实时和普通）实现多种调度策略。每种进程对应不同策略。
    - 限期：限期调度策略，选截至实践距当前时间点最近的进程
    - 实时： FIFO 和 RR
    - 普通：标准轮流分时， CFS(根据优先级分配不同时间片，一个调度时延内所有进程都有机会被调度一次)
    - 其它：停机 (0 号线程)，空闲进程

拓展： OpenEuler 多处理器调度 (3/3)

- openEuler 调度策略
  - 每个处理器都有一个调度队列

```
// 源代码： kernal/sched/sched.h
struct rq{
        unsigned int nr_running;// 进程总数
        struct cfs_rq  cfs;//CFS
        struct rt_rq  rt;//real time
        struct dl_rq  dl;//dead line
        struct task_struct *idle;
        struct task_struct *stop;
        int cpu;
};
```

9.2 Scheduling Algorithms

- 9.2.1 Short-term Criteria
- 9.2.2 The Use of Priorities
- 9.2.3 Alternative of Scheduling Policies
- 9.2.4 Fair-Share Scheduling

9.2.4 Fair-Share Scheduling(1/3)(公平共享调度)

- User's application runs as a collection of processes (threads)
- User is concerned about the performance of the application
- Need to make scheduling decisions based on process sets

9.2.4 Fair-Share Scheduling(2/3)(公平共享调度)

- Scheduling is done on the basis of
  - under-lying priority of the process 进程基本优先级
  - recent processor usage of the process 进程使用 cpu
  - recent processor usage of the group 用户组使用 cpu
- The higher the numerical value of the priority, the lower the priority. The following formulas apply for process j in group k:

$$CPU_j(i) = \frac{CPU_j(i-1)}{2}$$

$$GCPU_k(i) = \frac{GCPU_k(i-1)}{2}$$

$$P_j(i) = Base_j + \frac{CPU_j(i)}{2} + \frac{GCPU_k(i)}{4 \times W_k}$$

$$CPU_j(i) = \frac{CPU_j(i-1)}{2}$$

$$GCPU_k(i) = \frac{GCPU_k(i-1)}{2}$$

$$P_j(i) = Base_j + \frac{CPU_j(i)}{2} + \frac{GCPU_k(i)}{4 \times W_k}$$

	Process A			Process B			Process C		
	Priority	Process CPU count	Group CPU count	Priority	Process CPU count	Group CPU count	Priority	Process CPU count	Group CPU count
	60	0 1 2 • • 60	0 1 2 • • 60	60	0	0	60	0	0
1	90	30	30	60	0 1 2 • • 60	0 1 2 • • 60	60	0	0 1 2 • • 60
2	74	15 16 17 • • 75	15 16 17 • • 75	90	30	30	75	0	30
3	96	37	37	74	15	15 16 17 • • 75	67	0 1 2 • • 60	15 16 17 • • 75
4	78	18 19 20 • • 78	18 19 20 • • 78	81	7	37	93	30	37
5	98	39	39	70	3	18	76	15	18

90←60+30/2+30/2

90←60+30/2+30/2

90←60+0+30/2

96←60+37/2+37/2

Group 1 Group 2

Colored rectangle represents executing process

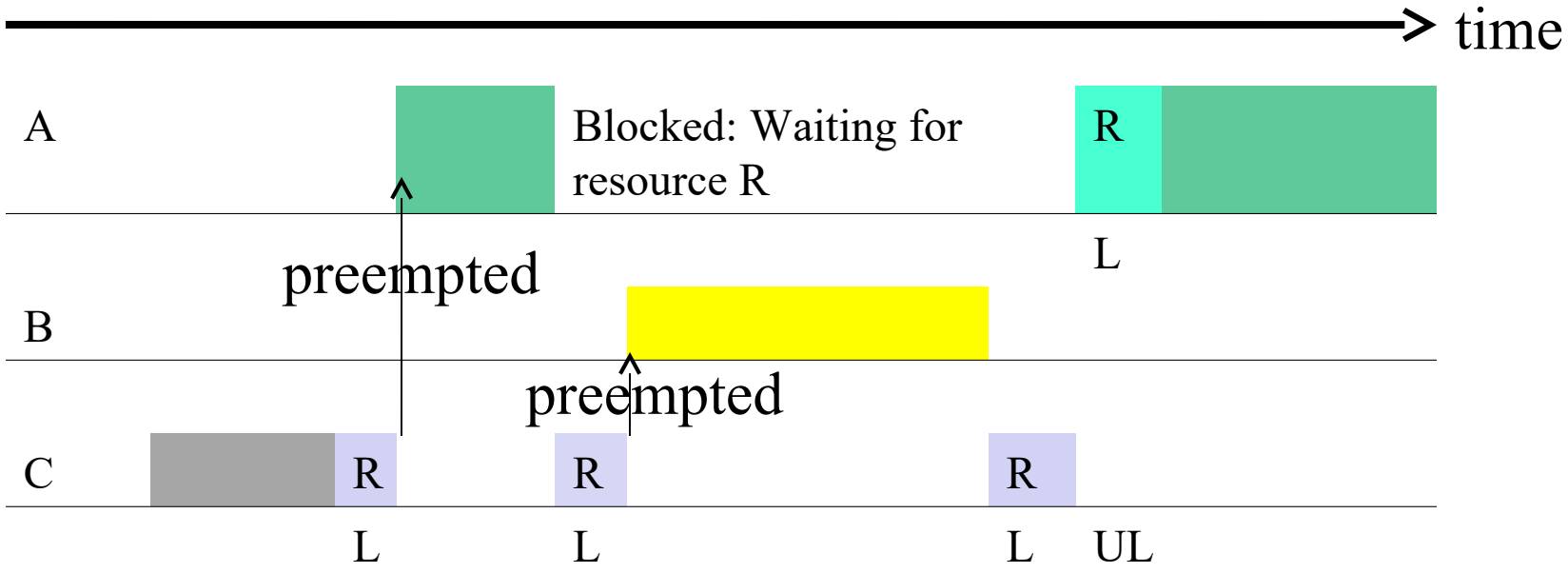
Figure 9.16 Example of Fair-Share Scheduler—Three Processes, Two Groups

Agenda

- 9.1 Type of Processor Scheduling
- 9.2 Scheduling Algorithms
- 9.3 Summary

An interesting problem

- 到目前为止我们讨论了各种调度方案，但有一个小问题我们没有分析，如果低优先级的进程占用了某个高优先级需要的资源，会怎么样？优先级 A>B>C,



如何设计一个调度策略解决这个问题？