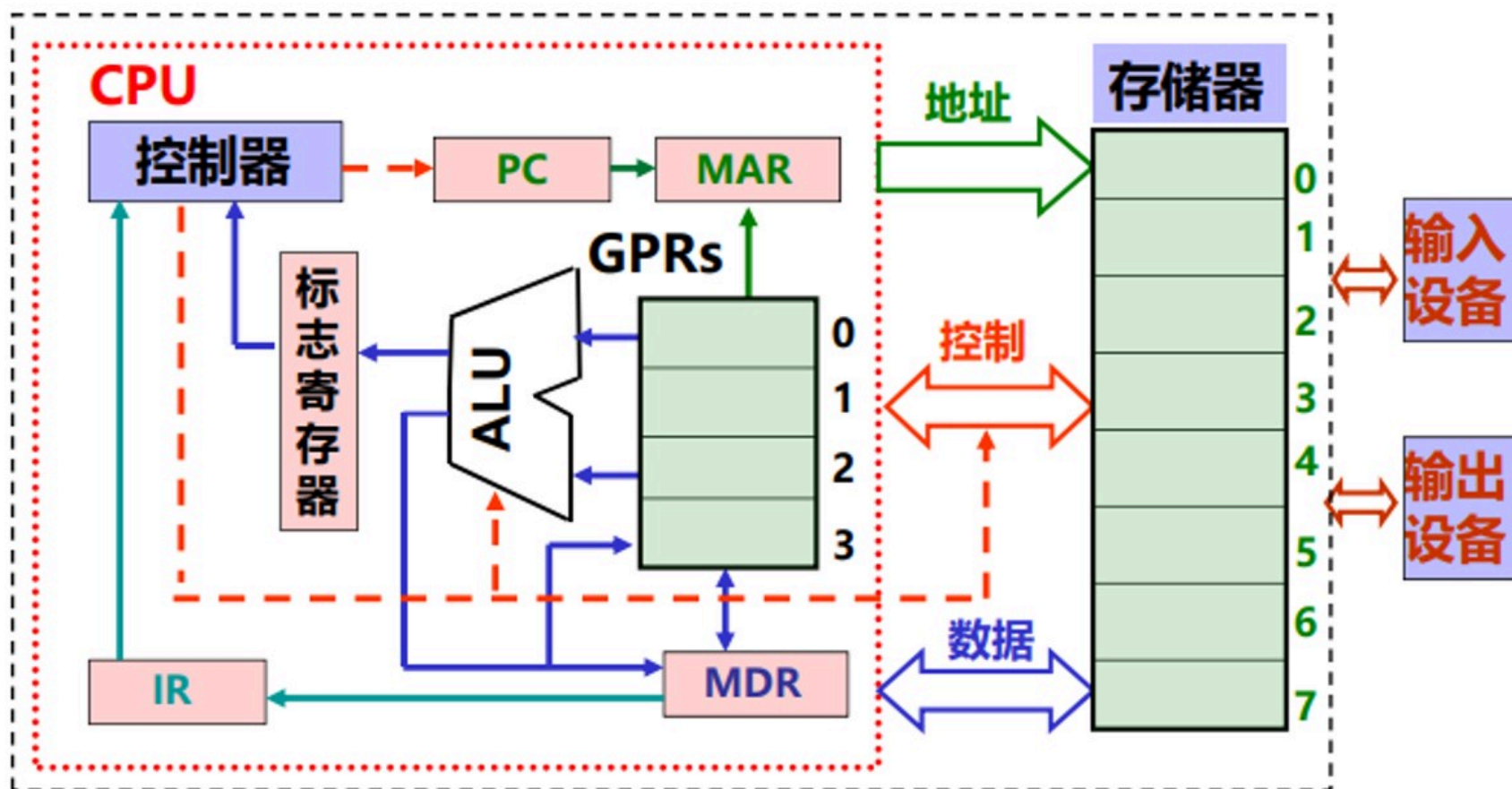


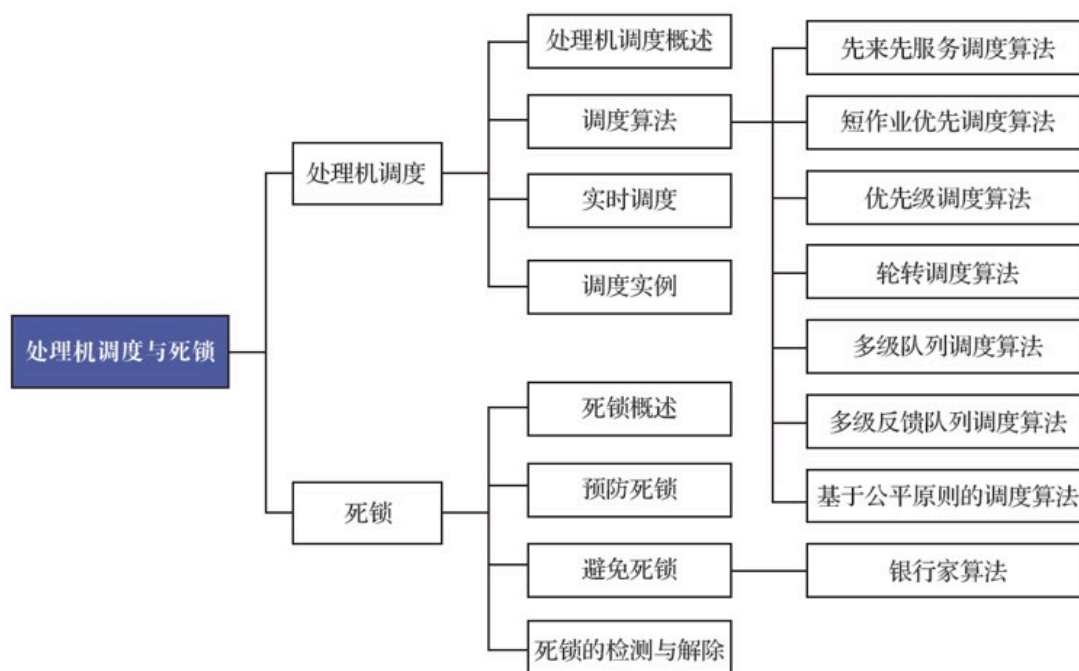
知识回顾





第3章知识导图

第1章	操作系统引论
第2章	进程的描述与控制
第3章	处理机调度与死锁
第4章	进程同步
第5章	存储器管理
第6章	虚拟存储器
第7章	输入/输出系统
第8章	文件管理
第9章	磁盘存储器管理
第10章	多处理机操作系统
第11章	虚拟化和云计算
第12章	保护和安全





Part 1 调度基本概念 (3.1)

Part 2 调度算法 (3.2)

Part 3 死锁概念 (3.5)

Part 4 死锁处理方法 (3.6, 3.7, 3.8)

Part 1 调度基本概念





学习目标


- 能够说出并区分处理机调度的三个层次
- 能够理解每一种处理机调度的任务
- 理解进程调度的实现机制和调度方式

Part 1 调度基本概念

1.1 处理机调度的层次

 **高级调度** (长程调度/**作业**调度)

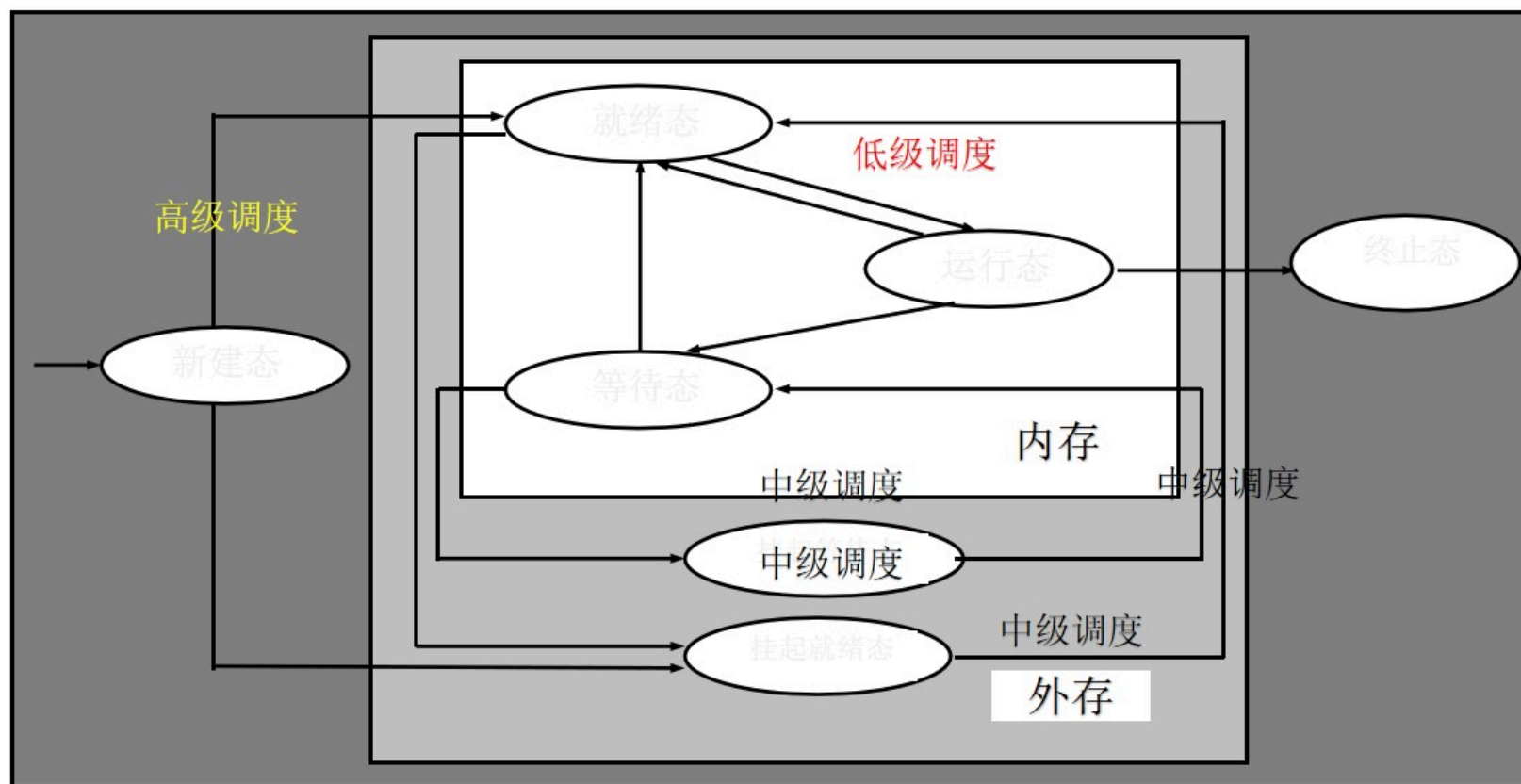
 **低级调度** (短程调度/**进程**调度)

 **中级调度** (中程调度/**内存**调度)

Part 1 调度基本概念

1.1 处理机调度的层次

● 三级调度图

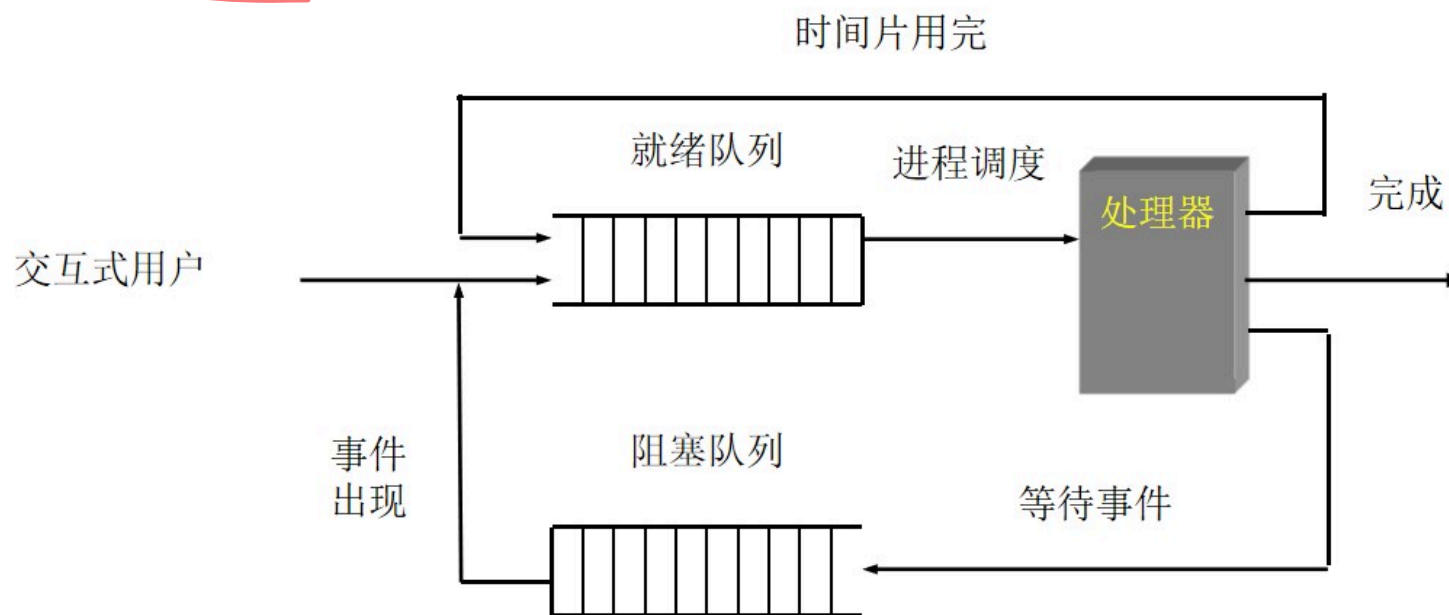


Part 1 调度基本概念

1.1 处理机调度的层次

● 调度队列模型

仅有进程调度的调度队列模型

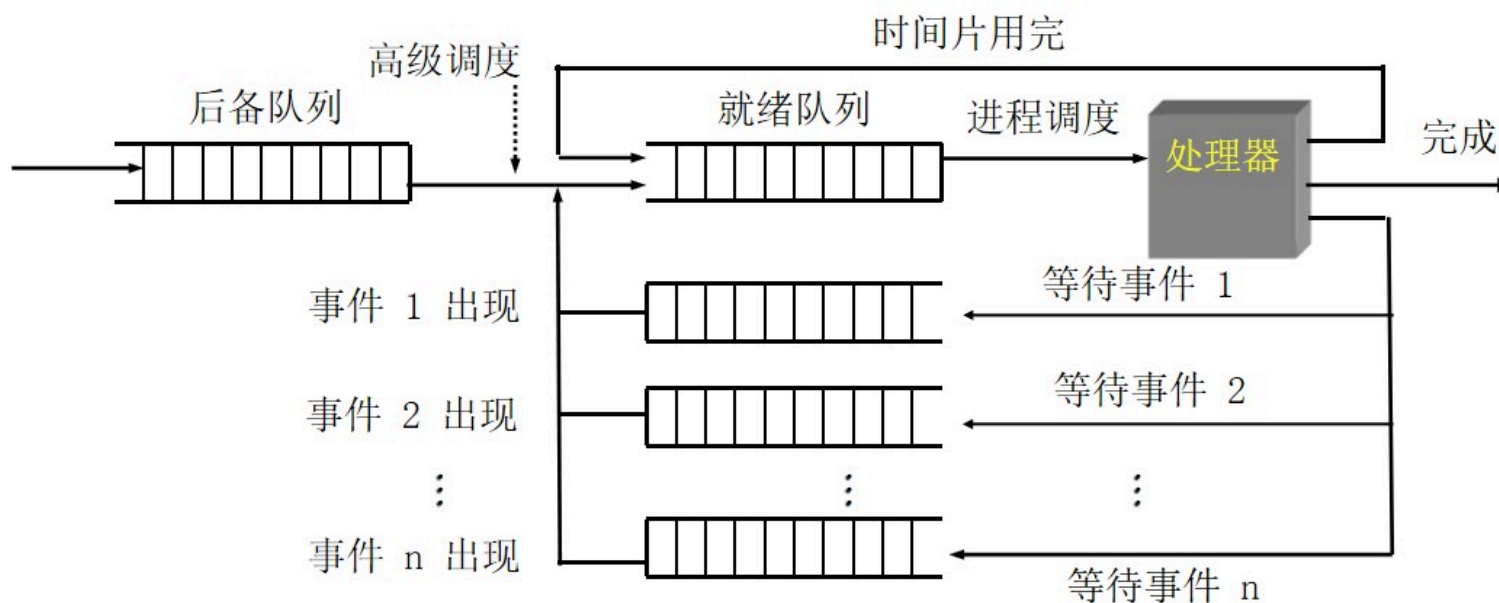


Part 1 调度基本概念

1.1 处理机调度的层次

● 调度队列模型

具有高级和低级调度的调度队列模型

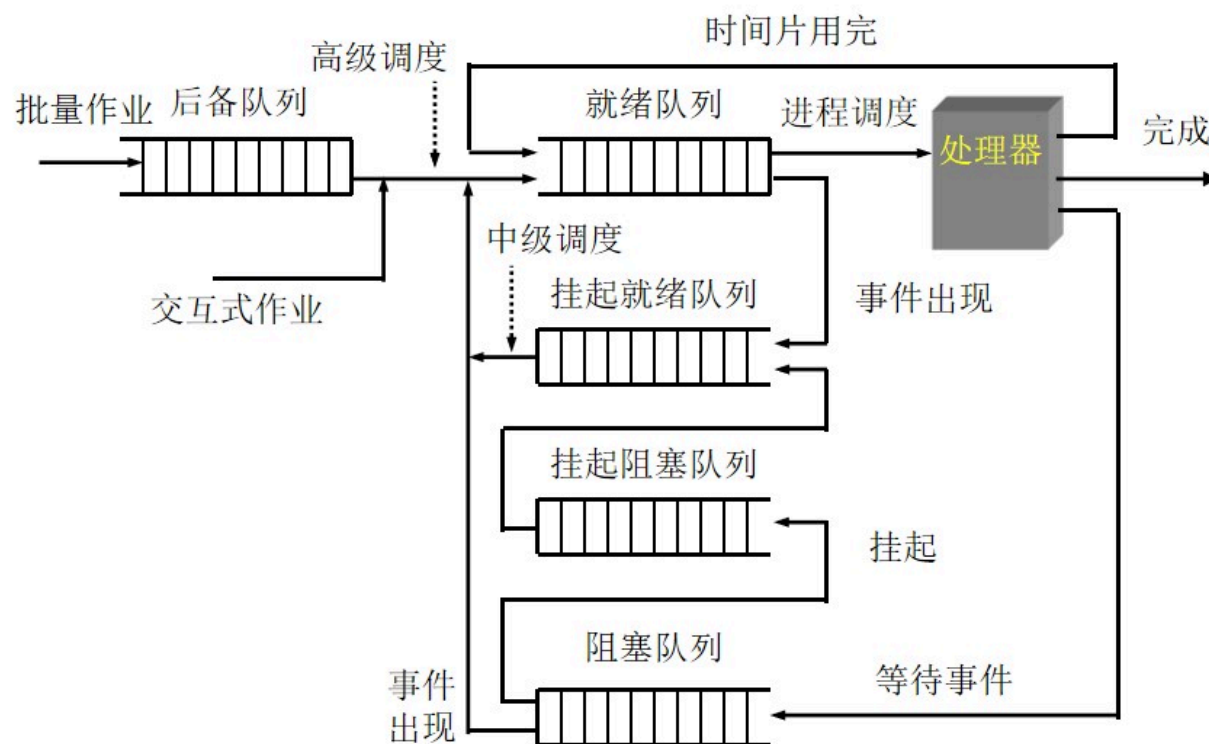


Part 1 调度基本概念

1.1 处理机调度的层次

● 调度队列模型

三级调度队列模型



Part 1 调度基本概念

1.1 处理机调度的层次

● 调度层次总结

- 高级调度：选择磁盘的作业进入内存；批处理系统
- 进程调度：选择就绪队列中的进程获得CPU；各种支持多道的操作系统
- 中级调度：选择内存中暂时不执行的进程换出外存，或者将外存可以执行的进程换入内存；支持对换的操作系统

Part 1 调度基本概念

1.1 处理机调度的层次

● 进程调度方式



非抢占

进程获得CPU会一直使用直到阻塞或完成



抢占

按照一定的原则抢占CPU

- 优先级
- 时间片
- 短进程优先

Part 1 调度基本概念

1.1 处理机调度的层次

● 处理机调度目标



共同目标:

- 资源利用率
- 公平性
- 平衡性
- 策略强制执行

$$\text{CPU利用率} = \frac{\text{CPU有效工作时间}}{\text{CPU有效工作时间} + \text{CPU空闲等待时间}}$$



批处理系统的目标:

- 平均周转时间短、系统吞吐量高、处理机利用率高



分时系统的目标:

- 响应时间快、均衡性



实时系统的目标:

- 截止时间的保证、可预测性

Part 1 调度基本概念

1.1 处理机调度的层次

● 处理机调度评价指标



周转时间:

- 从作业提交给系统开始, 到作业完成为止的这段时间间隔。
- 平均周转时间
- 带权周转时间: 权值为作业周转时间 T 与系统为之服务时间 T_s 之比。
- 平均带权周转时间

$$T = \frac{1}{n} \left(\sum_{i=1}^n T_i \right)$$



响应时间:

- 从用户通过键盘提交请求开始, 直到系统首次显示出处理结果为止的一段时间。

$$W = \frac{1}{n} \left(\sum_{i=1}^n \frac{T_i}{T_{s_i}} \right)$$



等待时间 (进程调度) :

- 进程在就绪队列中等待调度的所有时间之和。

Part 2 调度算法

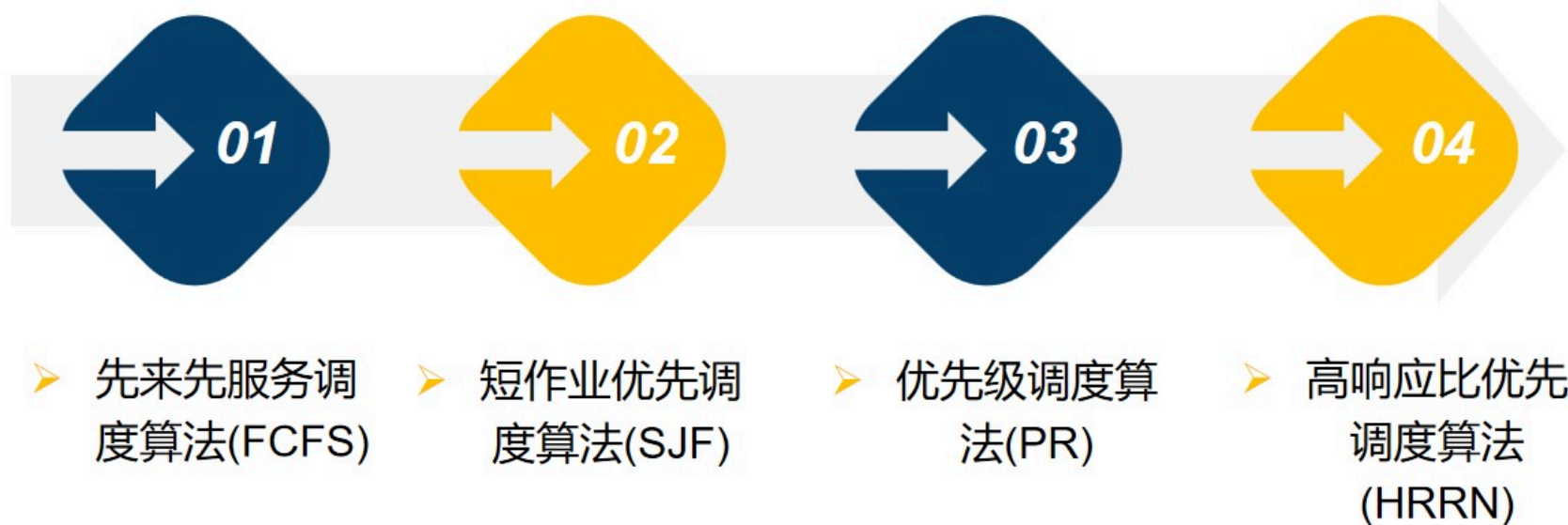


学习目标

- 能够说出调度算法的评价指标
- 能够理解每种调度算法的主要思想和特点
- 能够应用调度算法分析和解决应用问题

Part 2 调度算法

作业调度算法




FCFS、SJF、PR既可用于作业调度，也可用于进程调度


Part 2 调度算法

进程调度算法


 先来先服务调度算法(FCFS)


 多级队列调度算法

 短作业优先调度算法(SJF)

 多级反馈队列调度算法

 优先权调度算法(PR)

 基于公平原则的调度算法

 时间片轮转调度算法(RR)

Part 2 调度算法

2.1 先来先服务调度算法

00

按照作业到达的先后次序来进行调度

作业	运行时间
J1	24
J2	3
J3	3

00

假定作业到达顺序如下: J1, J2, J3
该调度的甘特图(Gantt)为:



➤ 平均等待时间 = $(0 + 24 + 27)/3 = 17$ ➤ 平均周转时间 = $(24 + 27 + 30)/3 = 27$

Part 2 调度算法

2.1 先来先服务调度算法

■ Q1: 先来先服务算法有何优缺点?

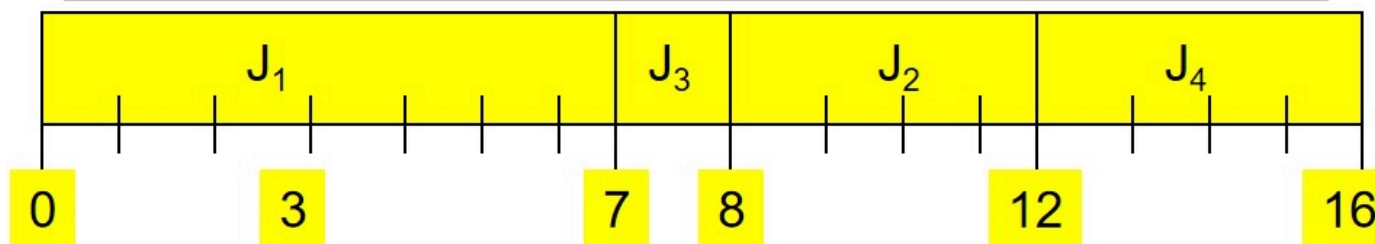
FCFS 对短作业不利（相较于 PR 和 SJF）；有利于 CPU 繁忙型作业，不利于 I/O 繁忙型作业。

Part 2 调度算法

2.2 短作业优先调度算法

非抢占式

进程	到达时间	运行时间
J_1	0	7
J_2	2	4
J_3	4	1
J_4	5	4



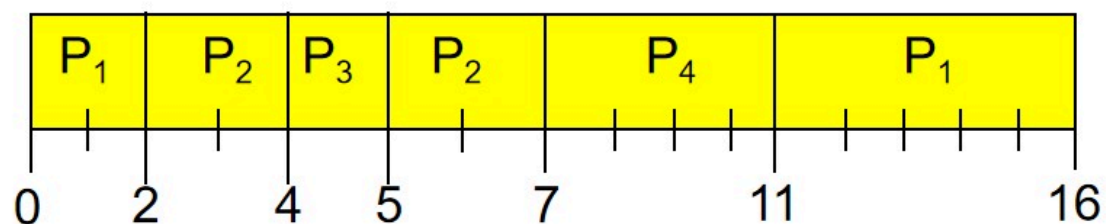
➤ 平均等待时间 = $(0 + 6 + 3 + 7)/4 = 4$ ➤ 平均周转时间 = $(7 + 10 + 4 + 11)/4 = 8$

Part 2 调度算法

2.2 短作业优先调度算法

抢占式

进程	到达时间	区间时间
P1	0	7
P2	2	4
P3	4	1
P4	5	4



➤ 平均等待时间 = $(9 + 1 + 0 + 2)/4 = 3$ ➤ 平均周转时间 = $(16 + 5 + 1 + 6)/4 = 7$



Part 2 调度算法

2.2 短作业优先调度算法

- Q2: 短作业优先算法有何优缺点?

Part 2 调度算法

2.3 优先级调度算法

-  既可用于作业调度，也可用于进程调度。
-  基于作业/进程的紧迫程度，由外部赋予作业相应的优先级，调度算法根据优先级进行调度。
 - 每个进程都有一个优先数，优先数为整数。
 - 默认：小的优先数具有高优先级。

Part 2 调度算法

2.3 优先级调度算法



优先级调度算法的类型

- 非抢占式
- 抢占式



优先级类型

- 静态优先级
- 动态优先级

Part 2 调度算法

2.3 优先级调度算法



➤ 平均等待时间 = $(0 + 1 + 6 + 16 + 18) / 5 = 8.2$

➤ 平均周转时间 = $(16 + 1 + 18 + 19 + 6) / 5 = 12$

Part 2 调度算法

2.3 优先级调度算法



优点

- 实现简单，考虑了进程的紧迫程度
- 灵活，可模拟其它算法



存在问题

- 饥饿 —— 低优先级的进程可能永远得不到运行



解决方法

- 老化 —— 视进程等待时间的延长提高其优先数

Part 2 调度算法

2.4 高响应比优先调度算法

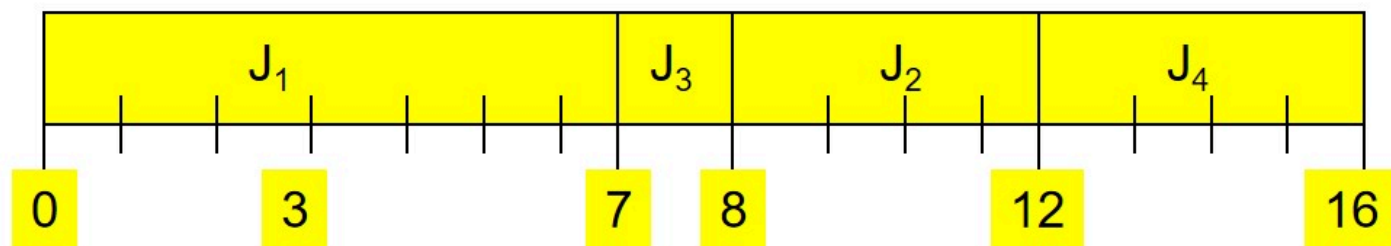
响应比

$$R_p = \frac{\text{等待时间} + \text{要求服务时间}}{\text{要求服务时间}} = \frac{\text{响应时间}}{\text{要求服务时间}}$$

Part 2 调度算法

2.4 高响应比优先调度算法

进程	到达时间	运行时间
J_1	0	7
J_2	2	4
J_3	4	1
J_4	5	4



➤ 平均等待时间 = $(0 + 6 + 3 + 7)/4 = 4$ ➤ 平均周转时间 = $(7 + 10 + 4 + 11)/4 = 8$

Part 2 调度算法

2.4 高响应比优先调度算法

■ Q3: 高相应比优先算法有何优缺点?

- 如等待时间相同, 取决于运行时间, 类似于SJF
- 如运行时间相同, 取决于等待时间, 类似于FCFS
- 长作业可随其等待时间的增加而提高, 也可得到服务
- 缺点: 每次调度之前, 都需要计算响应比, 增加系统开销