操作系统

Operating system

胡燕

大连理工大学

内容纲要

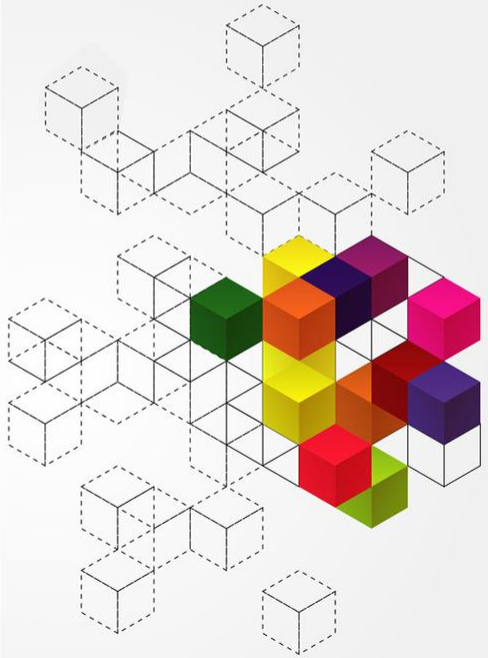
5.1CPU调度基本概念

一、调度基本概念

二、并行与并发

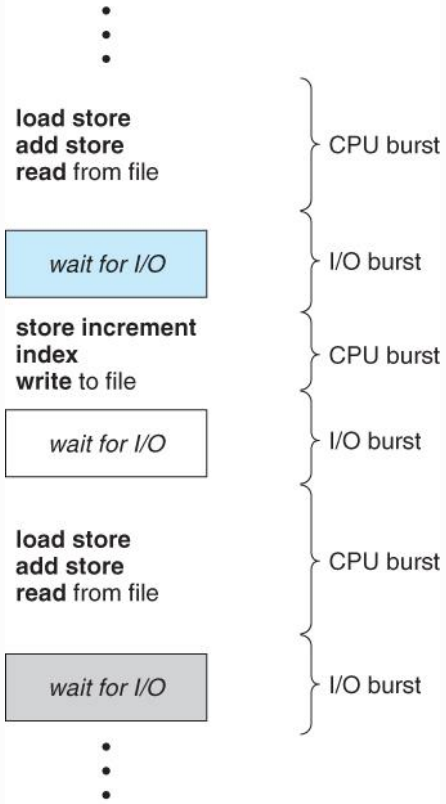
三、调度时机

四、CPU调度器构成

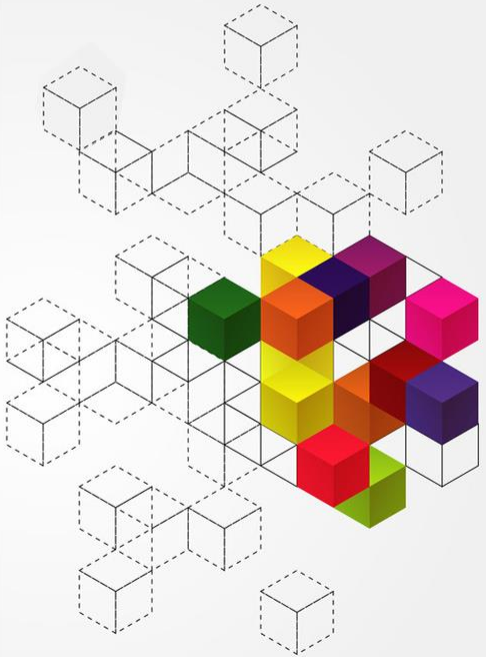


一、调度基本概念

为什么要进行调度

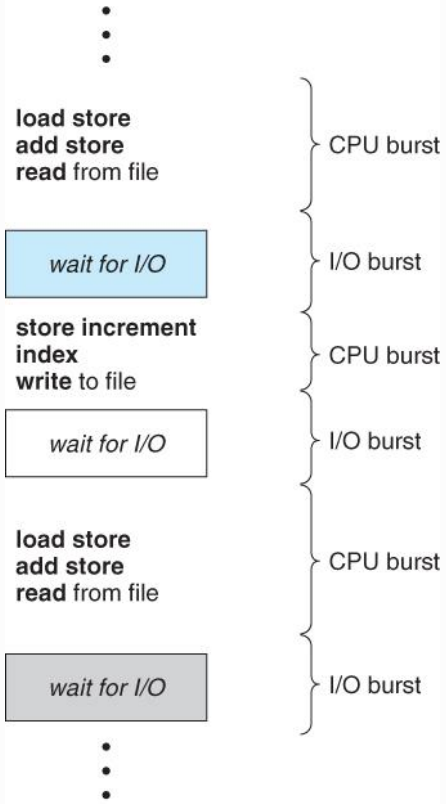


- 单任务系统
  - 程序执行场景：CPU burst与IO burst交替
  - CPU轻松，干一天休一天 （利用率不高）

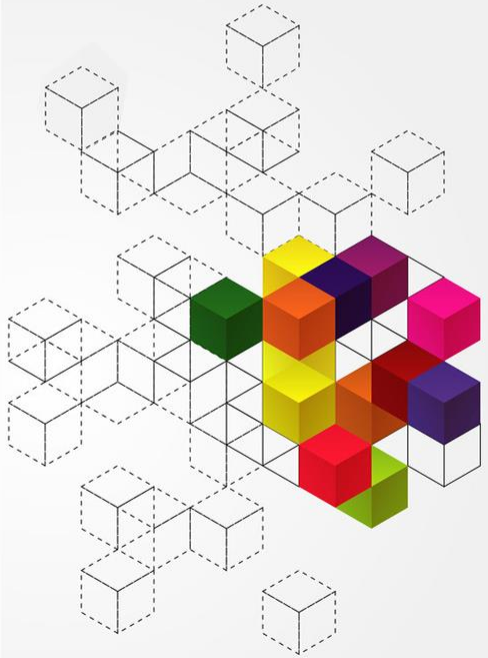


一、调度基本概念

为什么要进行调度 （基本场景）



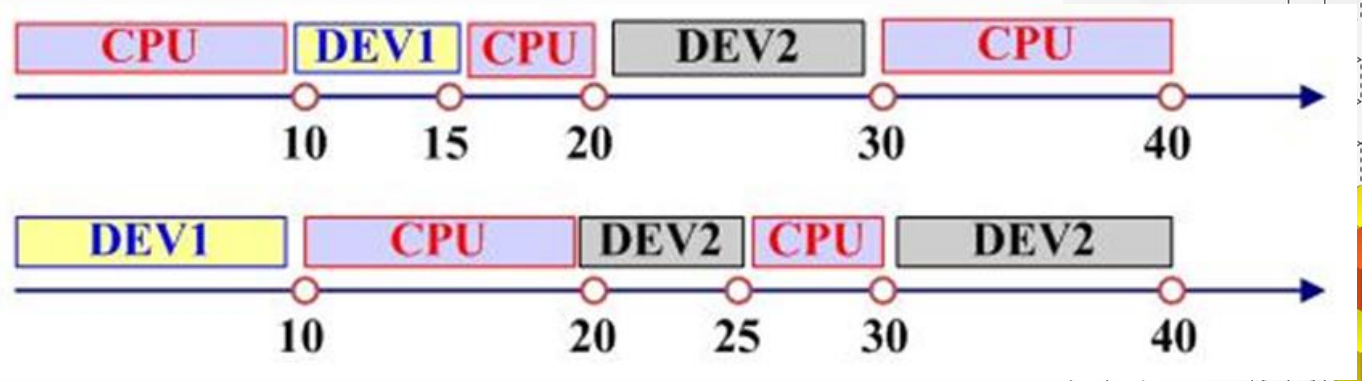
- 引入调度的基本动机
  - 多任务（Multitasking）
  - 程序执行特性：CPU burst与IO burst交替
  - 若仍采用串行执行的方式，则IO burst期间，CPU被限制（浪费CPU资源）
  - 需要更为精细的调度方法，使得多任务能够以高效的并发形式运作



一、调度基本概念

为什么要进行调度 （基本场景）

例：串行与并发

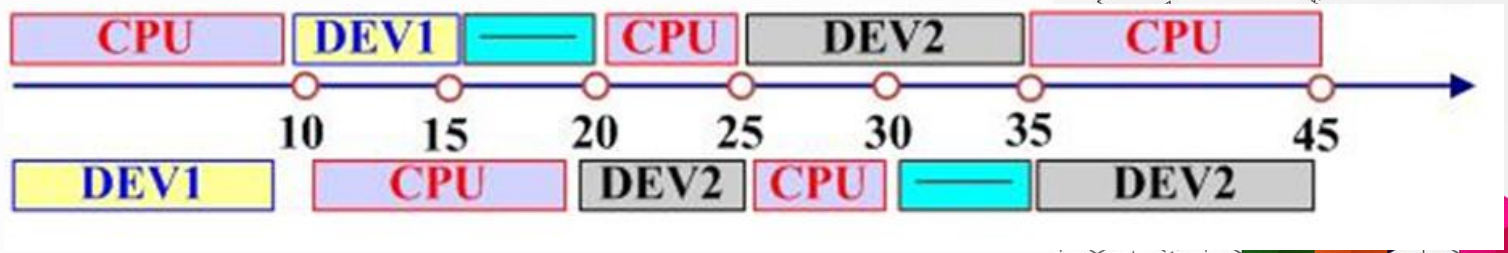


串行执行：A和B总周转时间=80

一、调度基本概念

为什么要进行调度 （基本场景）

例：串行与并发



并发执行：A和B总周转时间=45

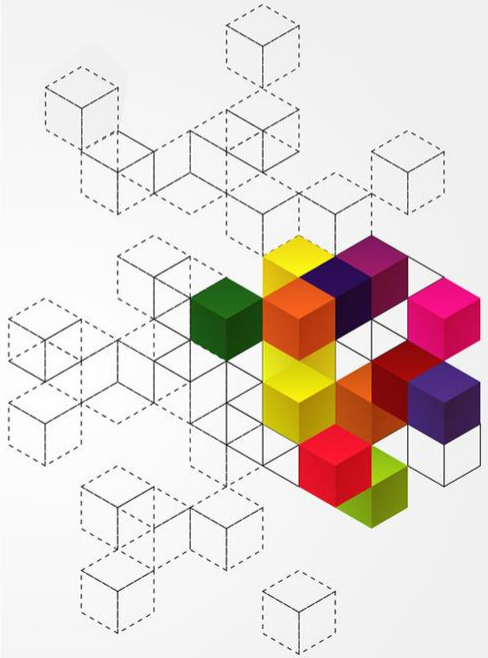
一、调度基本概念

sequential execution（串行执行结果）：

- Cumulative Time 周转时间：80
- CPU Efficiency：40/80 = 50%
- DEV1 Efficiency：15 / 80 = 18.75%
- DEV2 Efficiency：25 / 80 = 31.25%

Concurrent Execution（并发执行结果）

- Cumulative Time 周转时间：45
- CPU Efficiency：40 / 45 = 89%
- DEV1 Efficiency：15 / 45 = 33%
- DEV2 Efficiency：25 / 45 = 55.6%

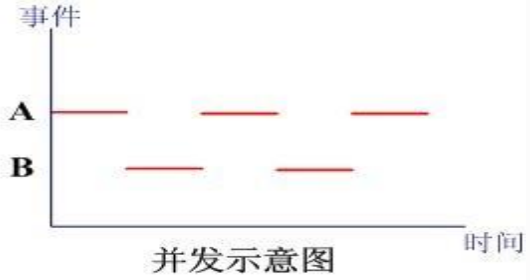


二、并行与并发

并行与并发的比较

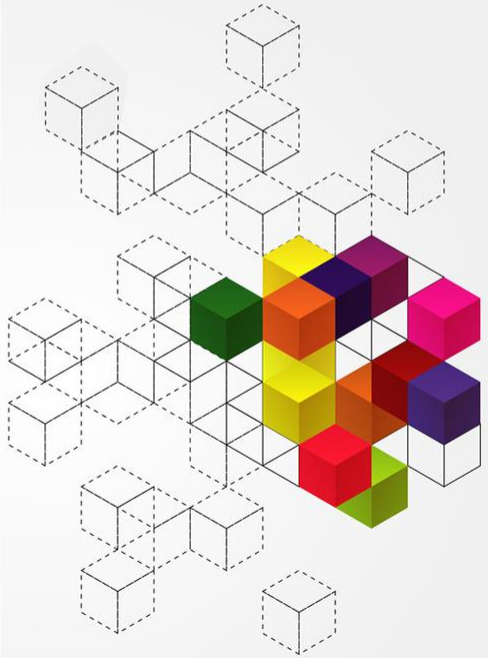
- 并行（Parallelism）
- 并发（Concurrency）

并行示意图



并发示意图

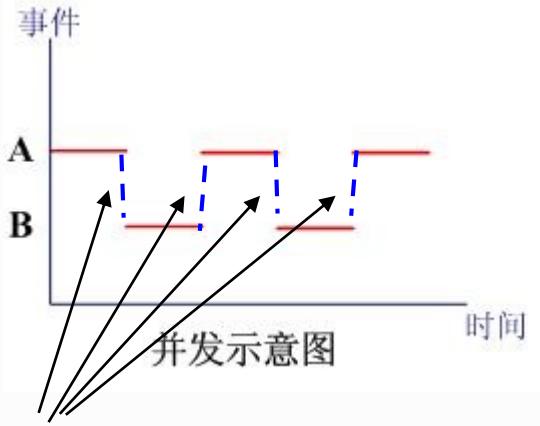
- 并发的概念 > 并行的概念
- 右侧的图说明的是并发概念中除了并行情况之外的另一种例外情况：在单处理机的机器上，也可以通过任务调度与切换来造成多任务同时执行的假象



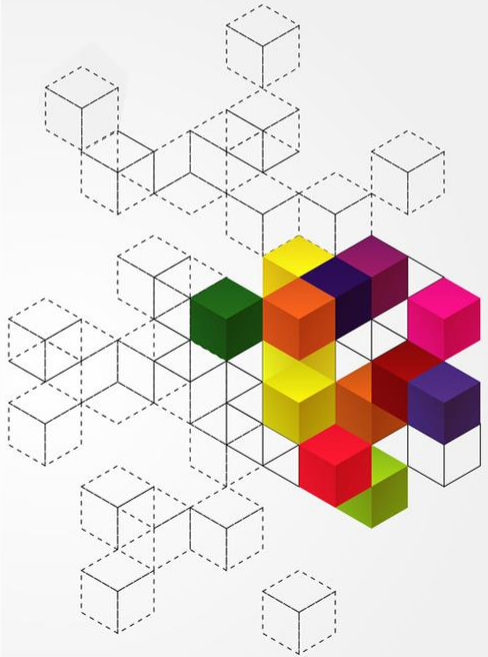
二、并行与并发

并发概念下的上下文切换

- 并发（Concurrency）

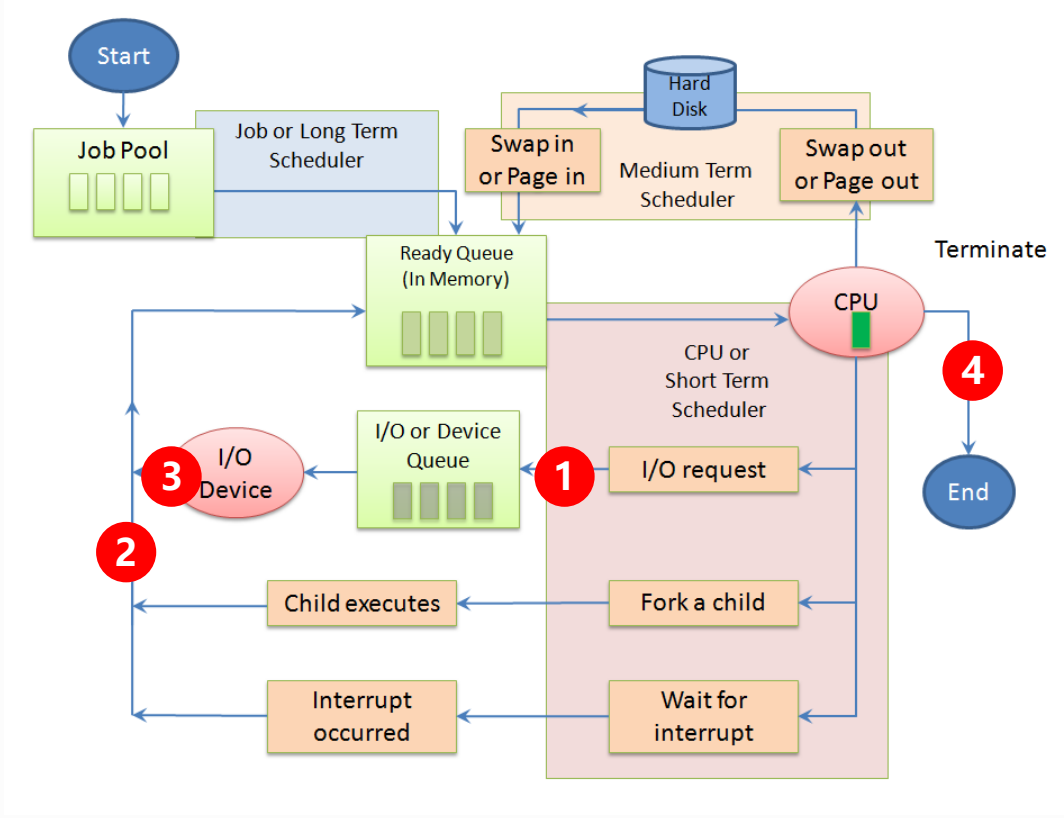


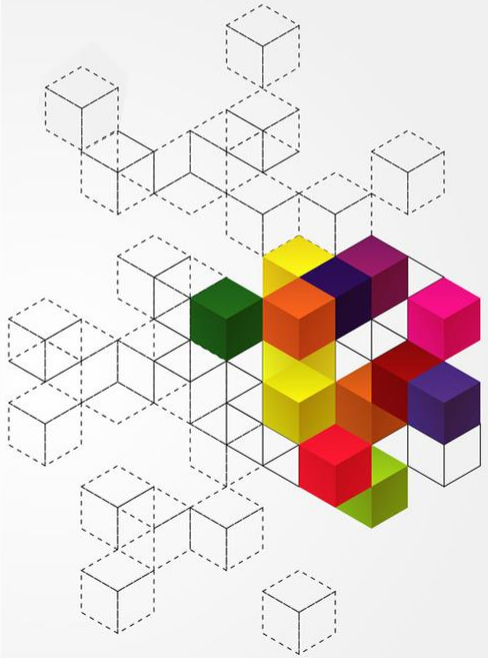
上下文切换，需要调度接入的位置



三、调度时机

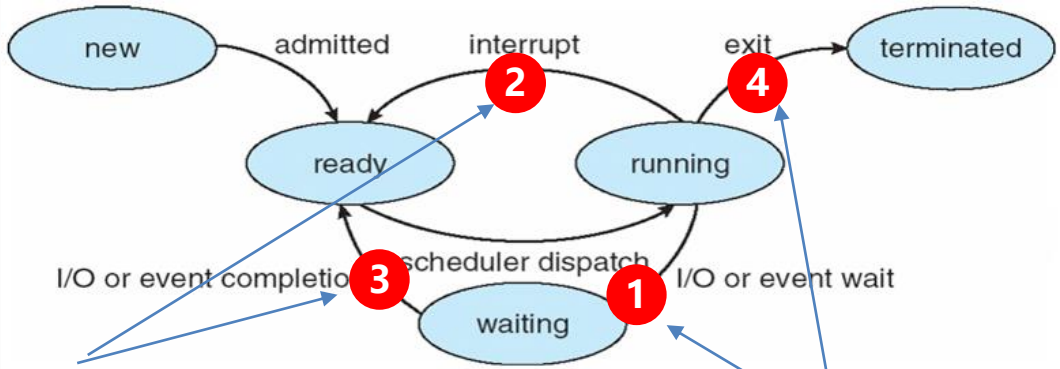
- 何时需要调度程序选择新的进程运行?





三、调度时机

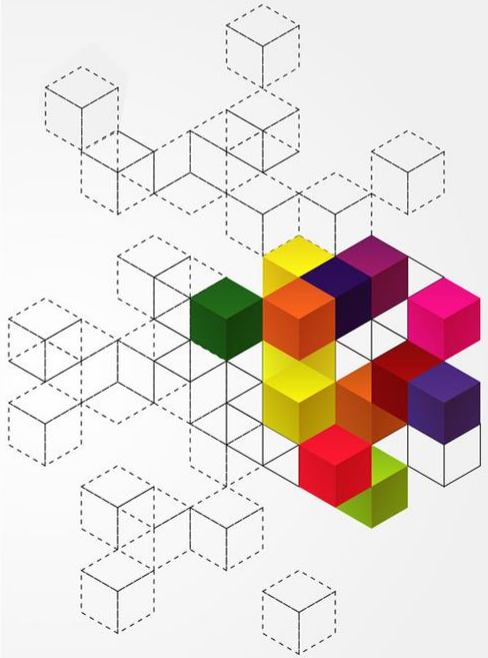
- 何时需要调度程序选择新的进程运行?
  - 从进程状态变换的视角看调度时机



可抢占调度点
(preemptive)

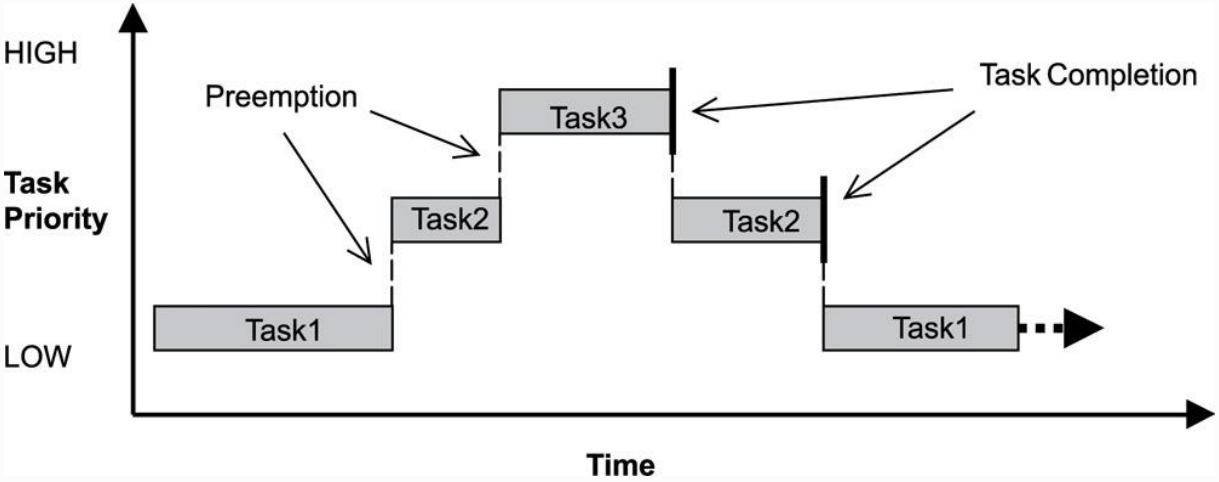
-允许新进程抢占当前进程执行

非抢占调度点
(non-preemptive)

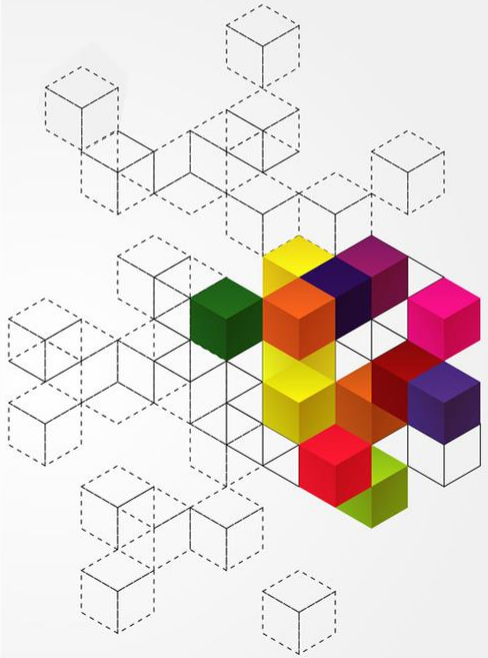


三、调度时机

- 何时需要调度程序选择新的进程运行?
  - Preemption in reatime OS



高优先级任务抢占低优先级任务的时间(preemption)

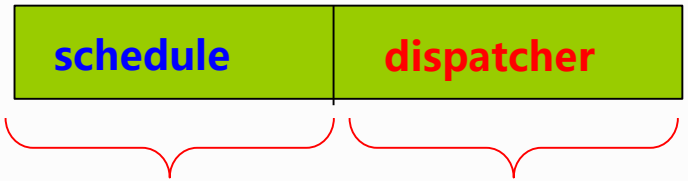


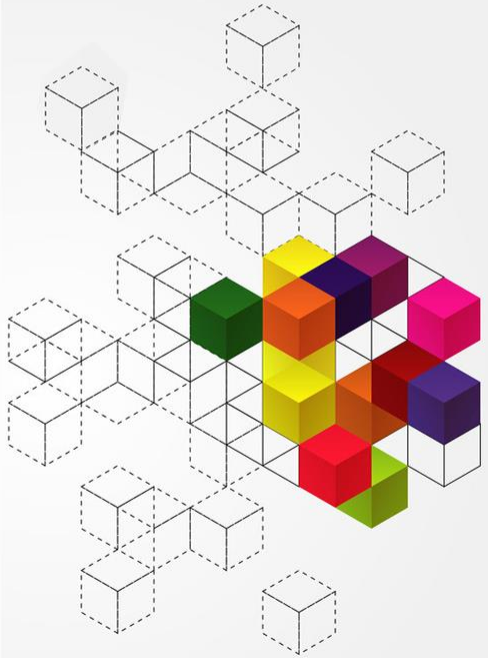
四、CPU调度器构成

调度器由两个主要部件组成：

- 调度(schedule)：在调度点，从就绪队列选择新的进程

- 派遣(dispatch)：将新的进程安排到CPU上运行
- 操作系统通过派遣程序将CPU的控制权交给新选中的进程





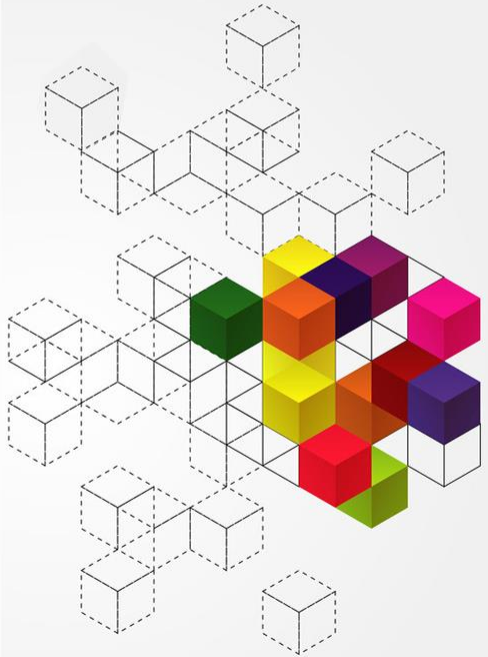
四、CPU调度器构成

派遣（Dispatch）的功能

- 1.切换上下文:切换到新被选中的进程的上下文
- 2.将进程从内核态切换到用户态
- 3.跳转到进程上次执行到的PC位置，重新开始执行

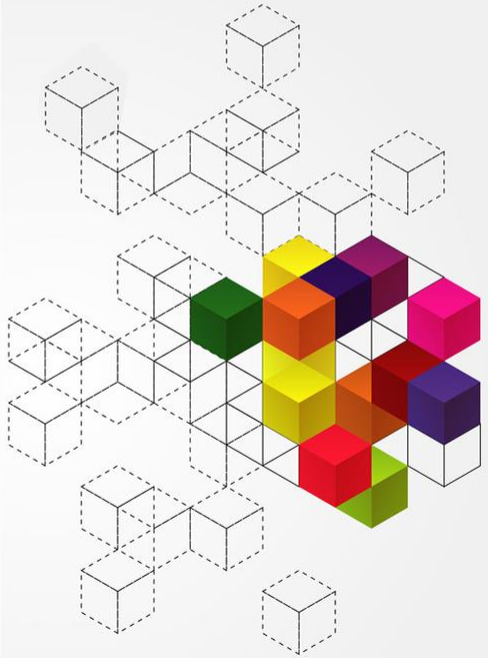
派遣延迟：派遣过程所耗费的时间

调度器执行一次调度的开销
=调度算法执行时间+派遣延迟



本讲小结

- 调度基本概念
- 并行与并发
- 调度时机
- CPU调度器构成



内容纲要

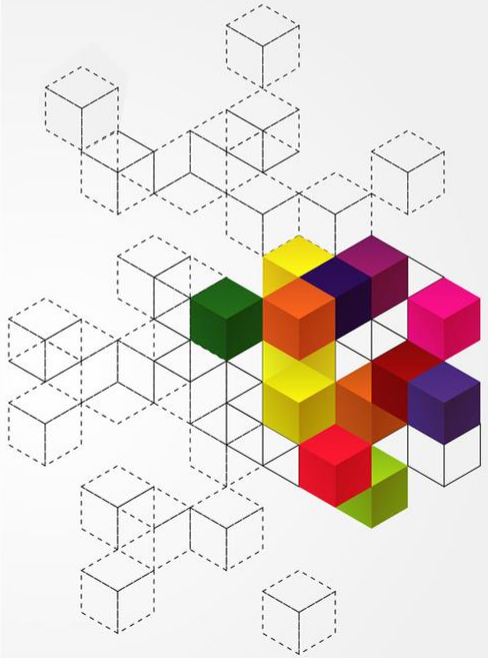
5.2 调度算法评价指标

一、调度算法评价

二、吞吐率

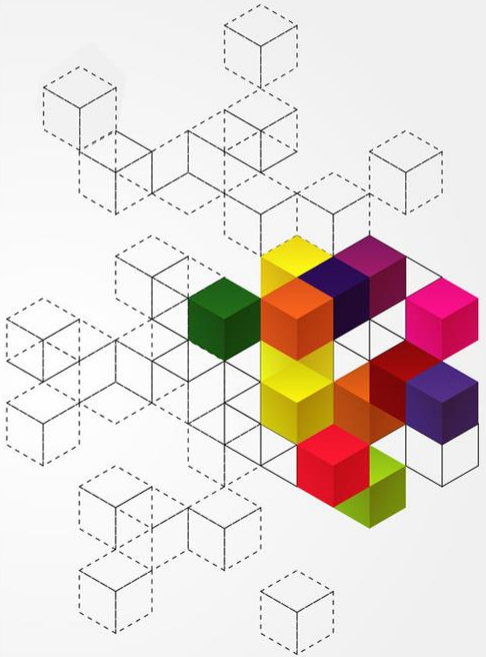
三、等待时间与周转时间

四、响应时间



一、调度算法评价

- **调度算法多种多样，需要特定的标准对其评价**
  - CPU利用率
  - 吞吐率
  - 平均周转时间
  - 平均等待时间
  - 周转时间

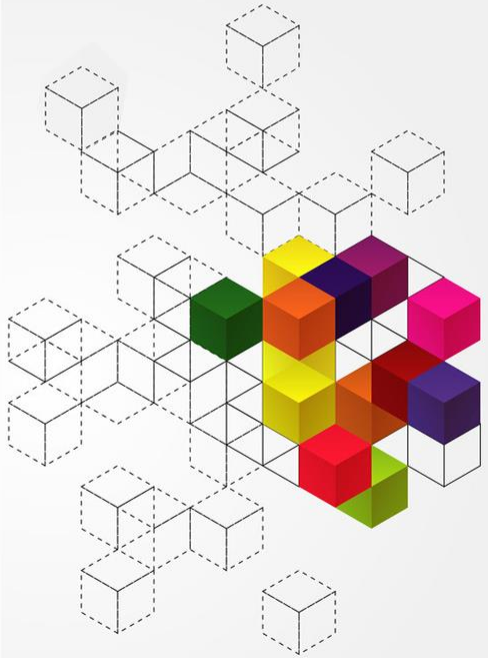


二、吞吐率

- **Throughput**
  - 单位时间内完成的进程（任务）数
  - 在这种标准下，调度算法优先选择短进程，保证完成任务的数量
  - 算法实例：短作业优先调度算法（后续有专题具体讨论）

$$Throughput = \frac{number\ of\ processes\ finished}{total\ execution\ time}$$

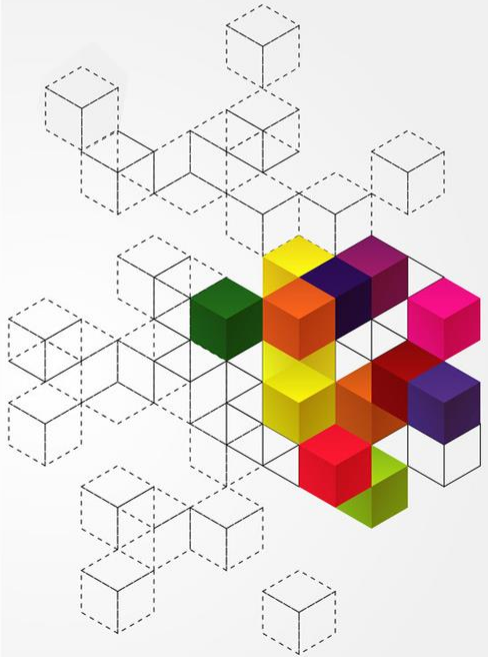
例如：某计算机完成了10道作业，共用了100秒，则系统吞吐量为 10/100 = 0.1道/秒



三、周转时间与等待时间

- **周转时间 （Turnaround Time）**
  - 进程从被创建，到执行完毕退出的时间跨度长度
  - 若以此为标准，调度算法会考虑进程的平均周转时间
  - 平均周转时间越短，说明进程的总体执行效率较高，也就表明调度算法较优

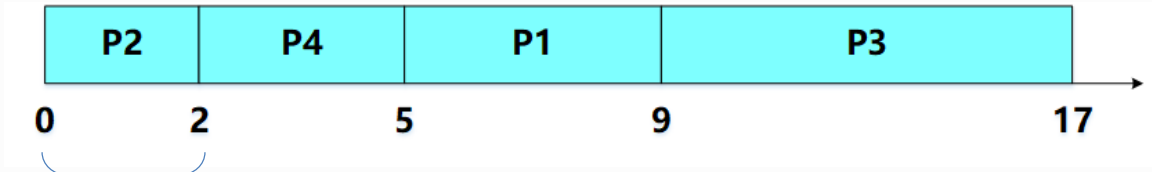
周转时间 = 进程完成时间 – 进程到达时间



三、周转时间与等待时间

- 周转时间 示例

- 调度甘特图示例：P1,P2,P3,P4进程，到达时间均为0

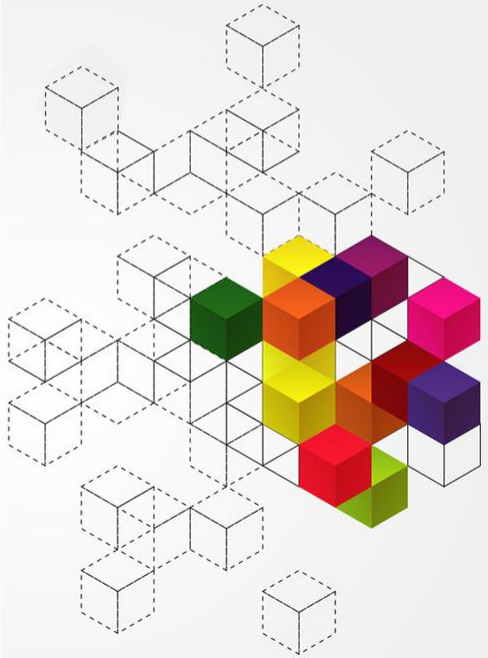


2-0=0

5-0=5

9-0=9

17-0=17

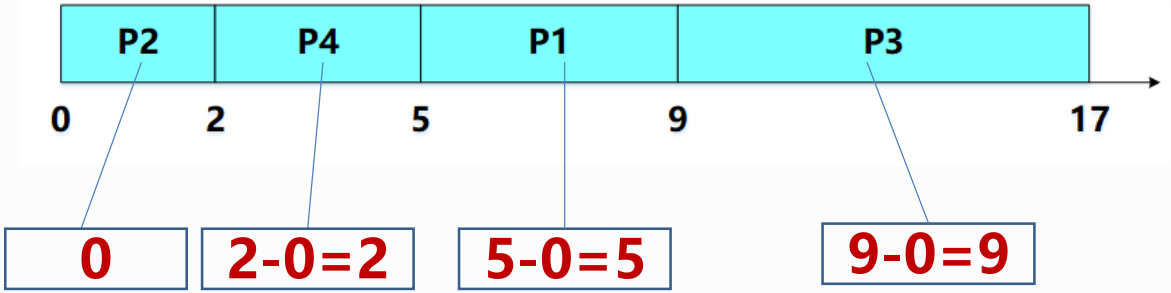


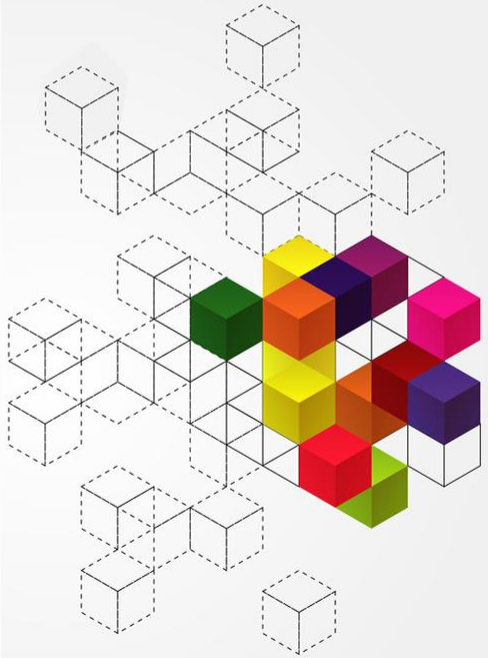
三、周转时间与等待时间

- **等待时间 （Waiting Time）**
  - 进程建立后在就绪队列内等待被服务的时间
  - 调度算法可以将进程平均等待时间作为标准
  - 等待时间越短，调度算法效果越好

How much time processes spend in the ready queue waiting their turn to get on the CPU

- **调度甘特图示例：P1,P2,P3,P4进程，到达时间均为0**



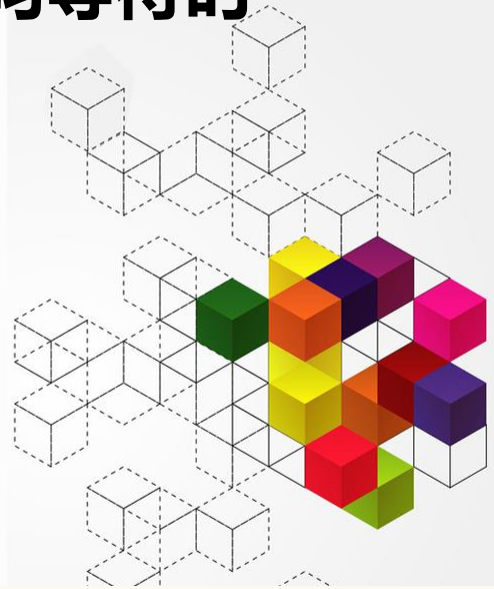


填空题 5分

设置

若进程按照ABCDE的次序被调度，则该调度中各个进程的平均周转时间= [填空1]，平均等待时间= [填空2]。

进程	到达时间	执行时间
A	0	3
B	2	6
C	4	4
D	6	5
E	8	2

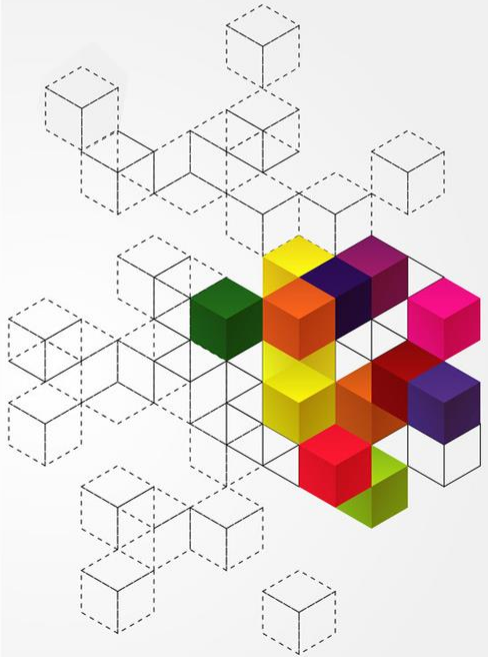


作答

四、响应时间

- **响应时间 （Response Time）**
  - 进程从发出服务请求，到请求被满足的时间跨度
  - 该标准在强调交互性的分时系统中较为常用
  - 通过控制响应时间在较小的范围内，从而保证不同用户的交互操作能够及时得到响应，保证用户公平地分享分时操作系统所管理的软硬件资源

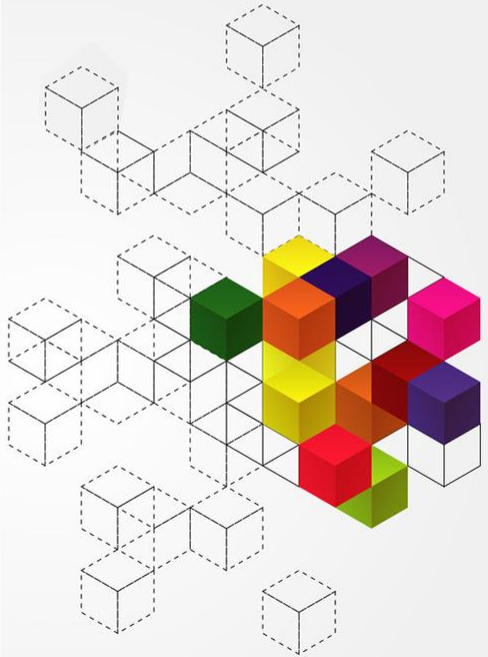
例如：浏览器中输入一个网址后，等了1秒钟，网页内容渲染完毕用户可以开始浏览，系统对此浏览事件的响应时间=1秒



四、响应时间

常见的系统操作响应时间

操作	响应时间
打开一个网站	几秒
在数据库中查询一条记录（有索引的情况下）	十几毫秒
机械磁盘的一次寻址	约4毫秒
机械磁盘顺序读取1MB的数据	约2毫秒
读取Redis服务器的一条数据	0.5毫秒
网络中传输2KB的数据	1毫秒



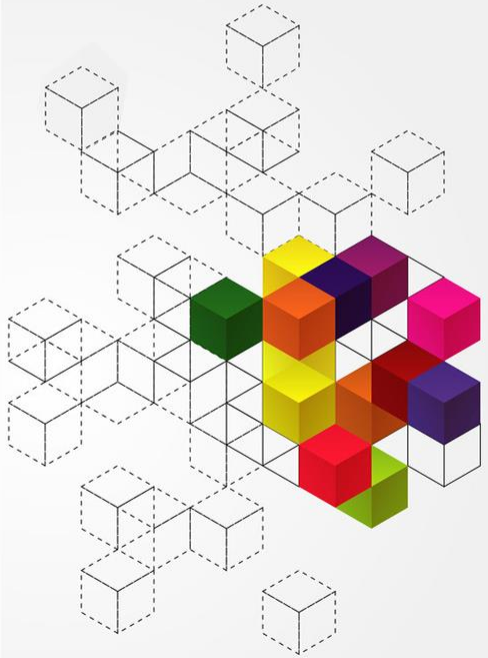
四、响应时间

响应时间对于不同类型操作系统的重要性

OS类型	重要程度
批处理系统	不重要
分时系统	重要
实时系统	非常重要

分时系统的响应时间期待：
毫秒级（尽量让人感知不到操作的粘滞）

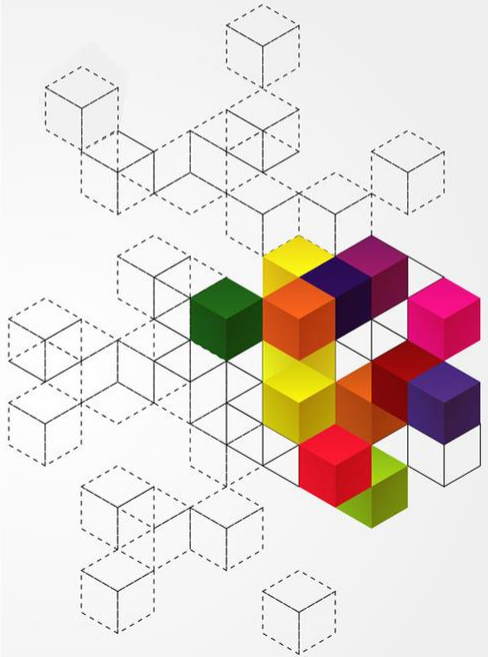
实时系统的响应时间期待：
微秒级（取决于系统的实际deadline需求）



四、响应时间

响应时间重要性案例：波音737MAX事故

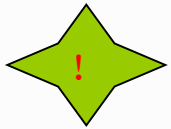
时间	事件简述
2019.3.10上午	埃塞俄比亚航空的737Max飞机失事坠毁
2012年早期	波音公司的一名试飞员对该系统错误激活的反应时间超过了10秒。报告里提到，该飞行员认为这种情况可能是“灾难性的”。
2012.11.1	一名波音员工通过电子邮件对另一名员工说：“（飞行员对该系统错误激活的）反应时间太长了。”这名未透露姓名的员工曾经询问是否应该提高该系统的风险评级，接到这一警告后，波音本应进行更彻底的安全检查。
	波音公司的最终结论是，机组人员对机动特性增强系统故障的反应要快得多，一般在4秒内。

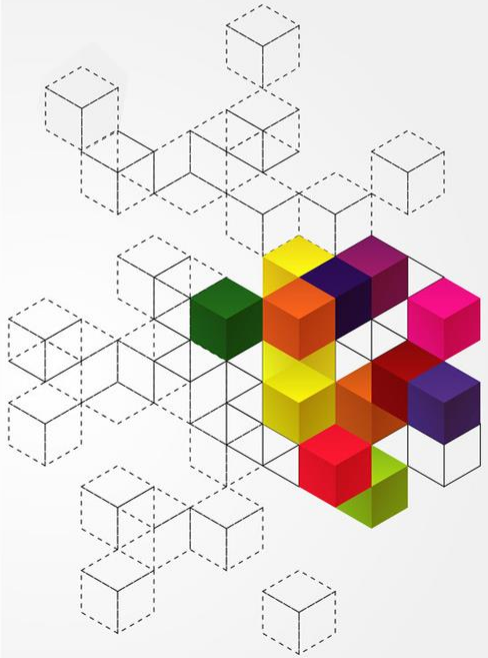


四、响应时间

调度目标：响应时间与吞吐率存在冲突

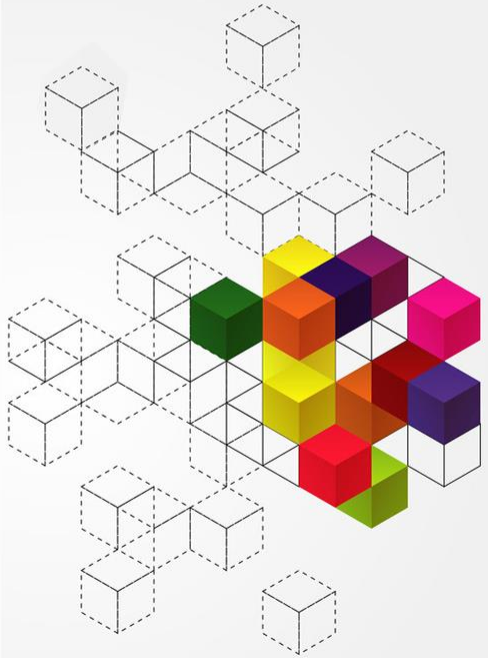
- **Max** CPU utilization
- **Max** throughput
- **Min** turnaround time
- **Min** waiting time
- **Min** response time





本讲小结

- 调度算法评价
- 吞吐率
- 周转时间与等待时间
- 响应时间



内容纲要

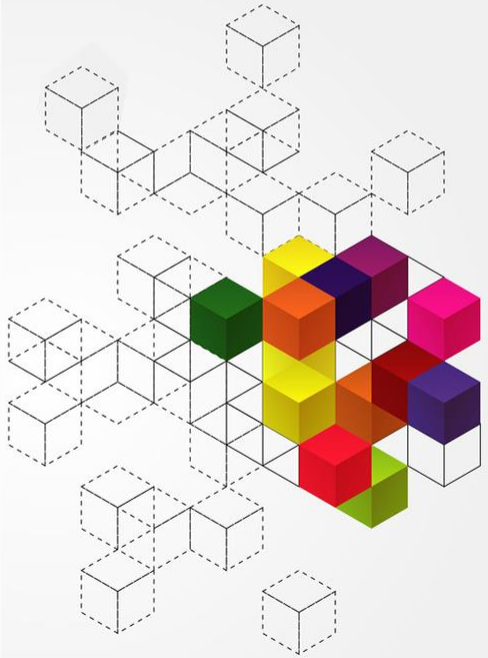
5.3 调度算法Part1

一、FCFS调度算法

二、SJF

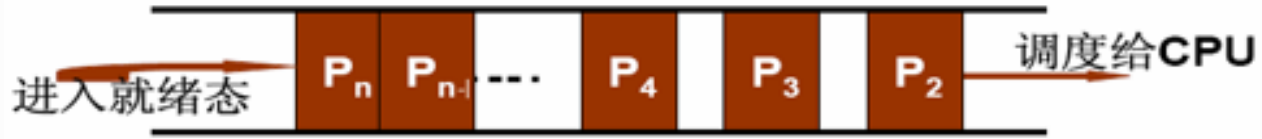
三、SRTF

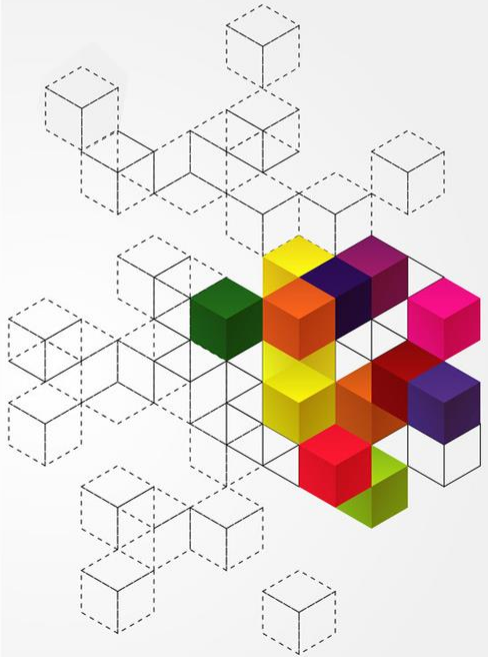
四、优先级调度



一、FCFS

- 先来先服务（First Come, First Serve）
- FCFS调度实现方式
  - 用一个FIFO队列来维护就绪进程
  - 每次从FIFO队列取队首进程，将其投入运行
  - 新进入就绪态的进程放入队尾





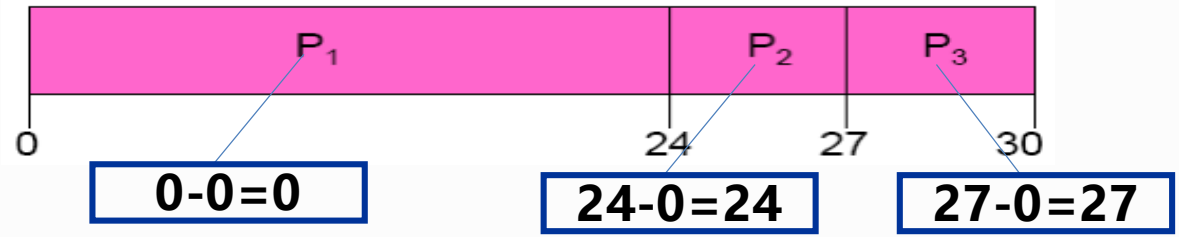
一、FCFS

- FCFS算法调度示例

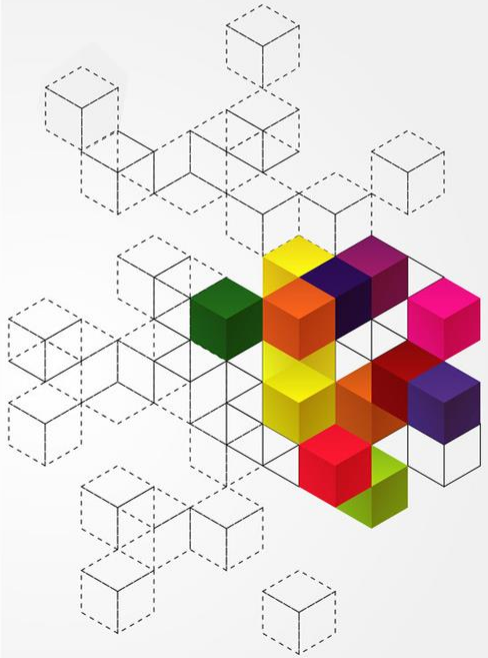
进程P1,P2,P3的顺序在时刻0依次到达

进程	CPU Burst time
P1	24
P2	3
P3	3

通过FCFS算法得到的Gantt图



平均等待时间: (0 + 24 + 27)/3 = 17



一、FCFS

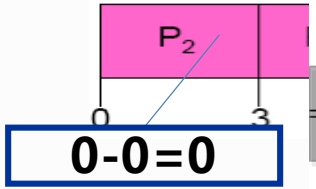
- FCFS算法调度示例

进程	CPU Burst time
P1	24
P2	3
P3	

Convoy Effect:
FCFS算法不稳定，长进程先于短进程到达，会导致平均等待时间拉长

假设3个进程

通过FCFS算



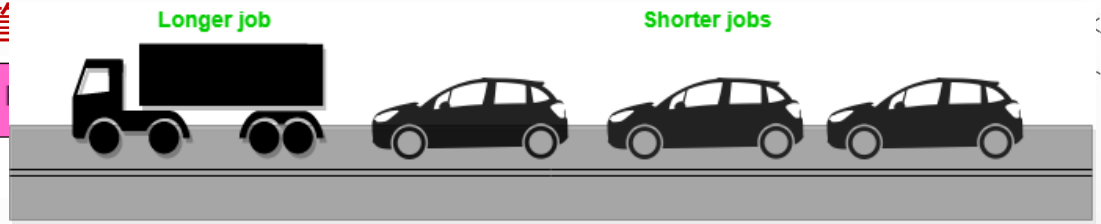
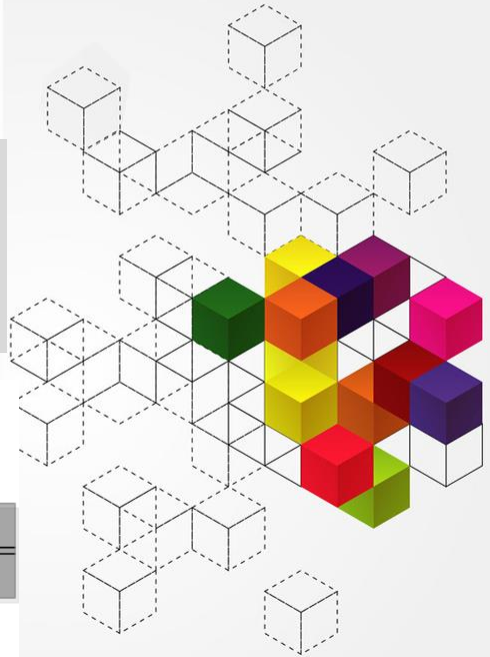


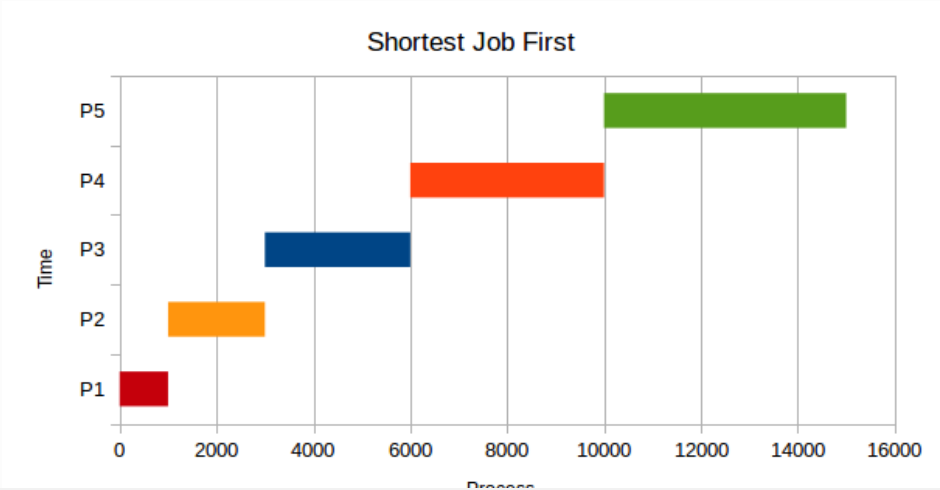
Figure - The Convey Effect, Visualized

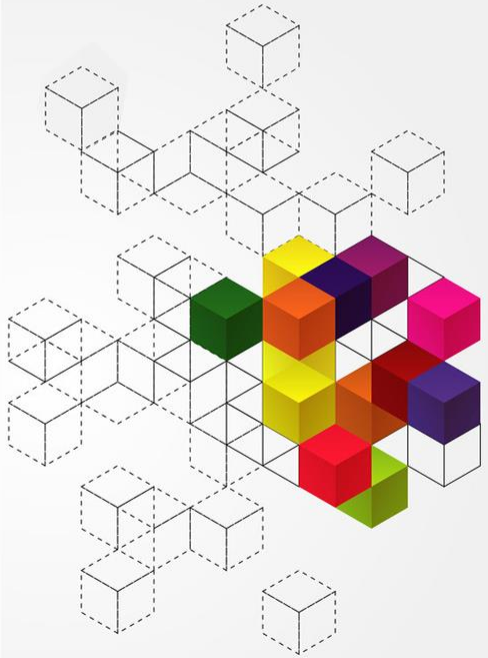
平均等待时间: (0 + 0 + 3)/3 = 3



二、SJF

- **短作业优先（Shortest Job First）**
  - 每次进行调度时，优先选择下一个CPU周期最短的进程
  - 调度重要信息：每个进程的下一个CPU周期长度
  - 按平均等待时间为指标，SJF是最优的调度



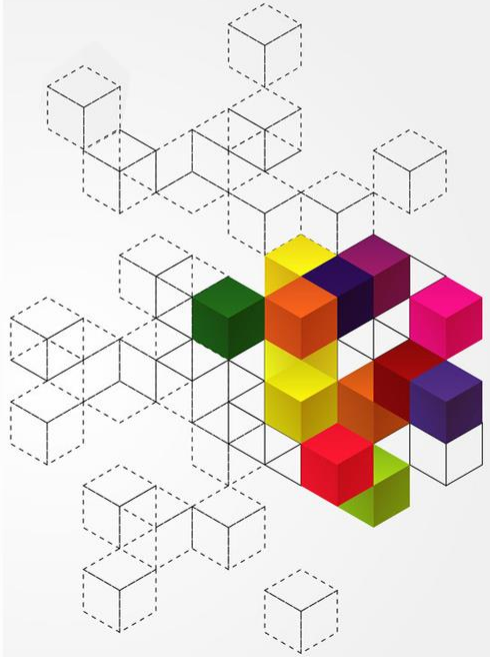


SJF调度目标：maximize throughput

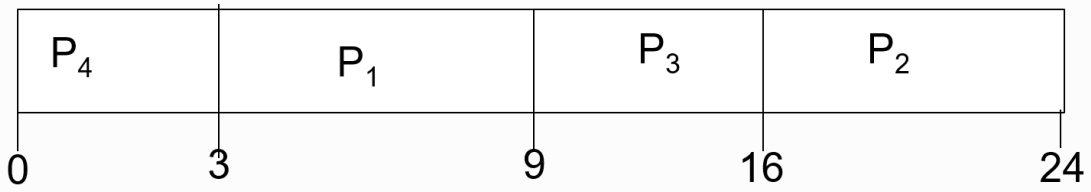
二、SJF

实例:

Process	Burst Time
P1	6
P2	8
P3	7
P4	3



SJF调度甘特图



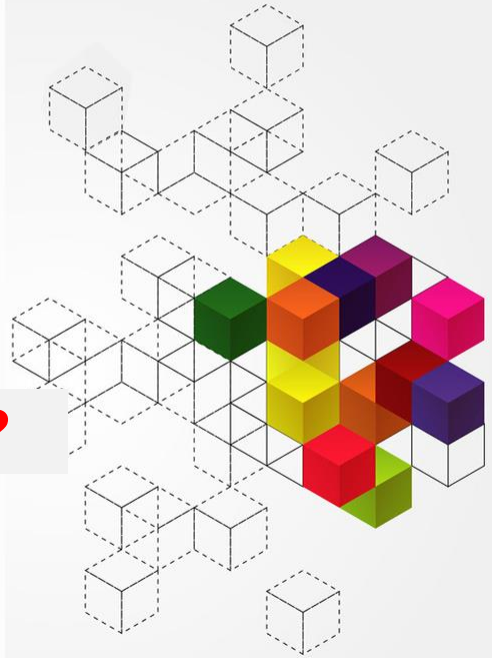
平均等待时间 = (3 + 16 + 9 + 0) / 4 = 7

二、SJF

练习:

Process	Arrival Time	Burst Time
P1	0	6
P2	2	8
P3	4	7
P4	5	3

SJF调度甘特图应为？平均周转时间=？平均等待时间=？



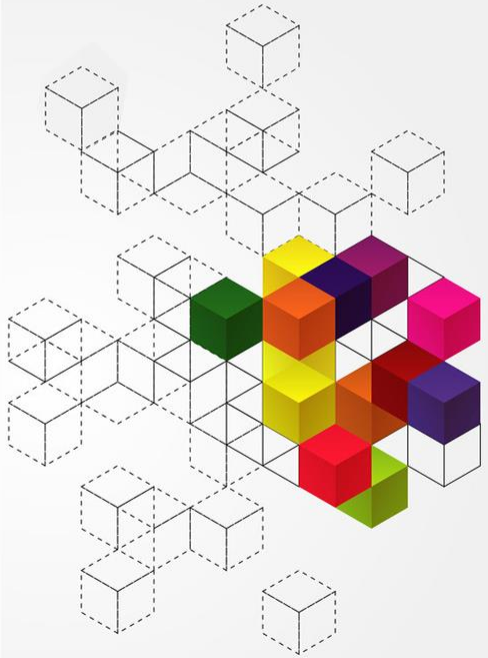
二、SJF

SJF算法实现问题:

下一个CPU Burst长度无法准确知道（还未执行）

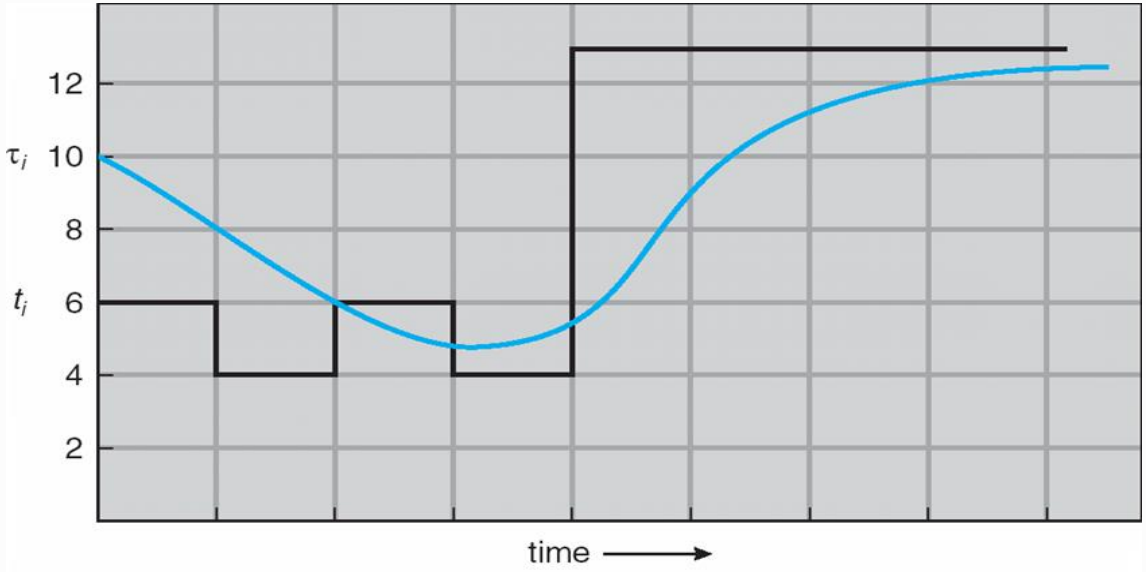
一种可选解决方案：预测

1. t_n = actual length of n^{th} CPU burst
2. τ_{n+1} = predicted value for the next CPU burst
3. $\alpha, 0 \le \alpha \le 1$
4. Define: $\tau_{n+1} = \alpha\, t_n + (1-\alpha)\tau_n$.

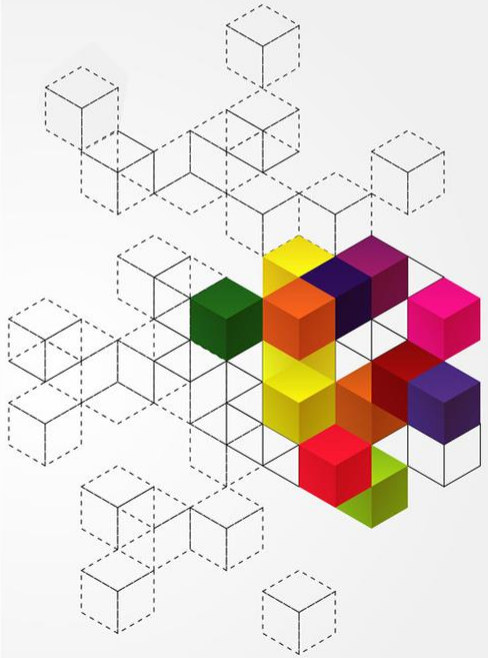


二、SJF

Predicting with exponential averaging



CPU burst (t_i)		6	4	6	4	13	13	13	...
"guess" (τ_i)	10	8	6	6	5	9	11	12	...

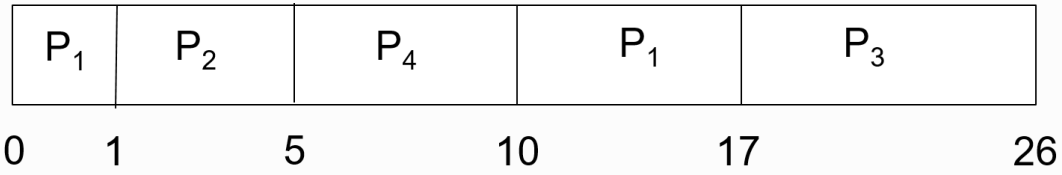


三、SRTF

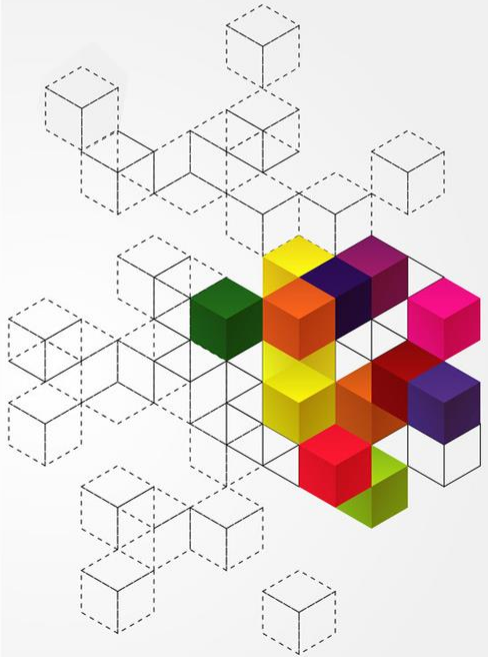
最短剩余时间优先（Shortest Remaining Time First）

Process	Arrival Time	Burst Time
P1	0	8
P2	1	4
P3	2	9
P4	3	5

SRTF的调度甘特图



平均等待时间= [(10-1)+(1-1)+(17-2)+5-3)]/4
= 26/4 = 6.5

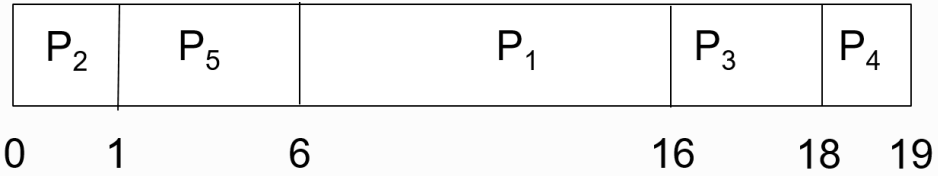


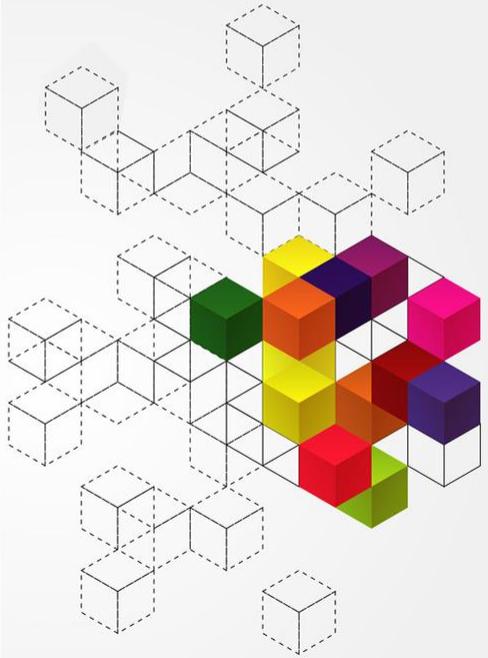
四、优先级调度

实例

Process	Burst Time	Priority
P1	10	3
P2	1	1
P3	2	4
P4	1	5
P5	5	2

优先级调度的甘特图





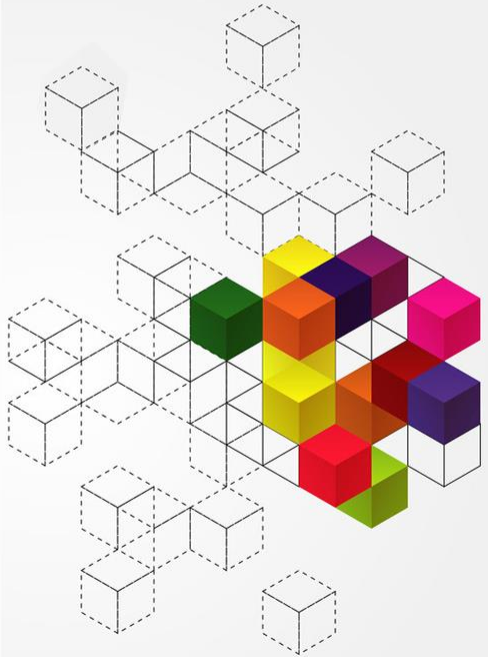
四、优先级调度

静态优先权 v.s. 动态优先权

静态优先权	各个进程的优先权在创建进程时所赋予，此后固定不变。
动态优先权	各个进程的优先权在创建进程时所赋予，随着进程的推进或其等待时间的增加而改变。

优先级通常被定为一个非负整数，称为优先数

- 有的系统优先数越小，优先级越高， 如Unix
- 有的系统优先数越大，优先级越高，如windows

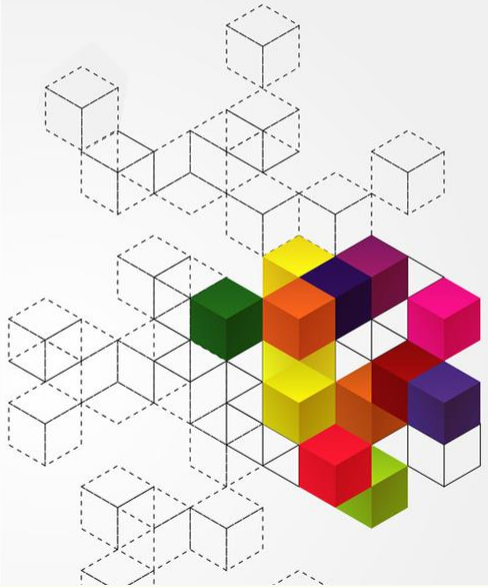


主观题 5分

设置

如果按照Windows的优先数设定规则，其调度的甘特图应为?

Process	Burst Time	Priority
P1	10	3
P2	1	1
P3	2	4
P4	1	5
P5	5	2

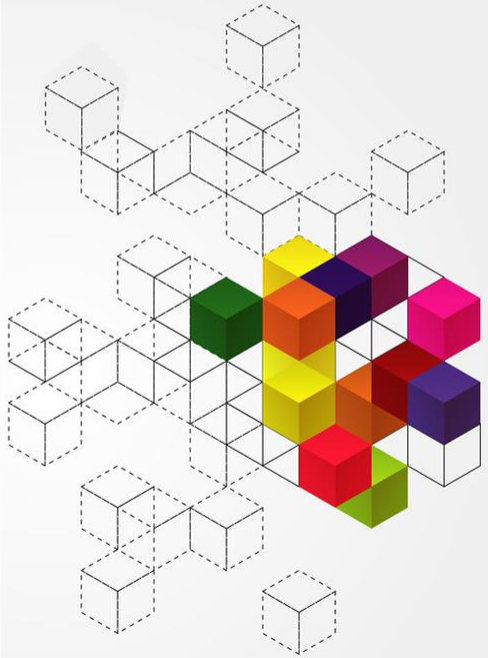


正常使用主观题需2.0以上版本雨课堂

作答

四、优先级调度

思考题：优先级调度存在何种问题，如何解决?



本讲小结

- FCFS
- SJF
- SRTF
- 优先级调度