

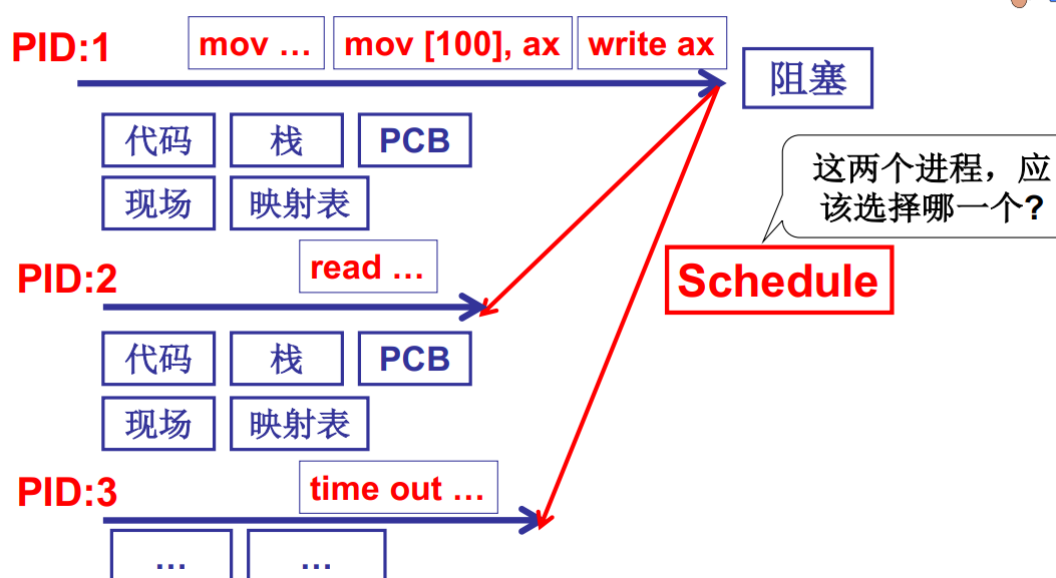
# 哈工大操作系统-L14CPU调度策略

## 哈工大操作系统-L14CPU调度策略

- 0.为何需要调度?
- 1.如何评价调度算法的好坏
  - 1.1吞吐量和响应时间的矛盾
  - 1.2前台任务和后台任务的关注点不同
- 2.调度算法0--先来先服务(FCFS: First Come, First Served)
- 3.调度算法1--短作业优先(SJF)--平均周转时间最短
- 4.调度算法2--时间片轮转(RR:round robin)--考虑响应时间
- 5.很多问题需要折中

## 0.为何需要调度?

### 多进程图像与CPU调度



- 有一堆进程可选的情况下，选哪一个就成了问题

## 1.如何评价调度算法的好坏

- **周转时间**--从任务被提交到系统到这个任务被执行完毕
  - 等待进入内存的时间,在就绪队列中等待的时间,在 CPU中执行的时间和I/O操作的时间的总和
- **响应时间**--从操作开始到响应
- **吞吐量**--系统完成的任务量(系统的内耗即消耗在调度上的时间小，吞吐量大)

但是指标需要折中....因此需要综合考量

## 1.1 吞吐量和响应时间的矛盾

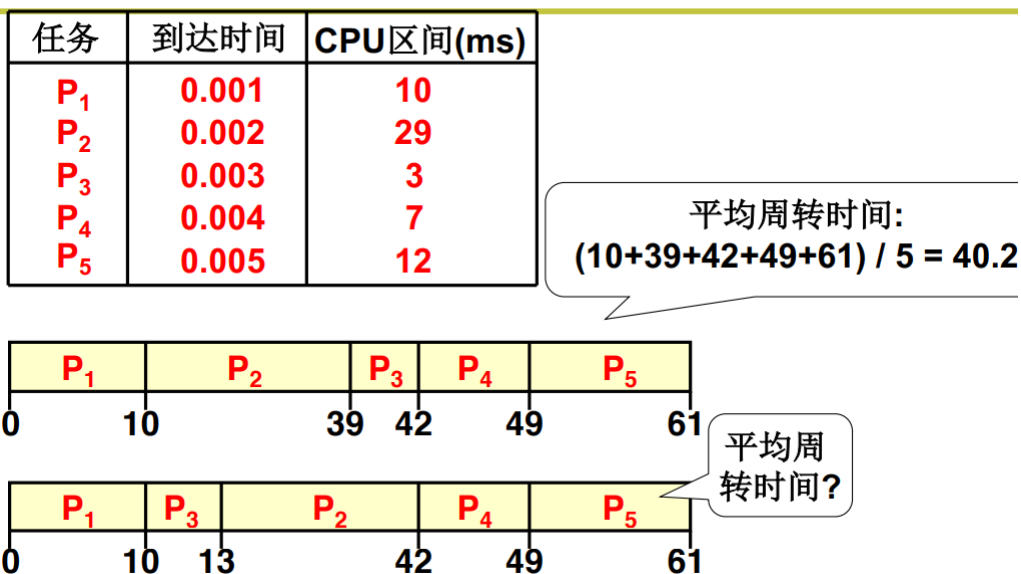
响应时间小---导致-->切换次数多---导致-->系统内耗大---导致-->吞吐量小

## 1.2 前台任务和后台任务的关注点不同

- 前台任务关注响应时间
- 后台任务关注周转时间

## 2. 调度算法0--先来先服务(FCFS: First Come, First Served)

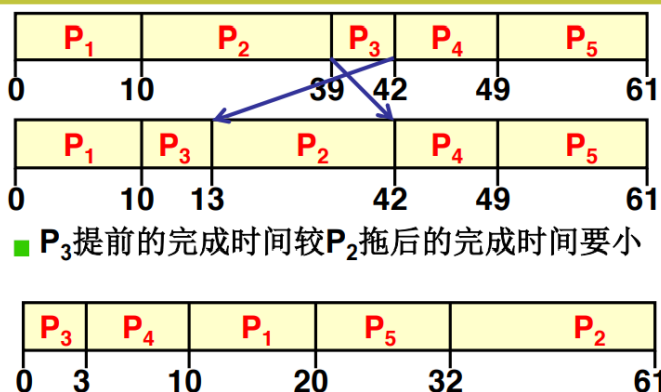
### First Come, First Served (FCFS)



## 3. 调度算法1--短作业优先(SJF)--平均周转时间最短

上述我们可以发现, 先来先服务的周转时间太长, 如果我们把短的服务提前, 我们可以得到一个较小的平均周转时间。因此引申出了SJF短作业优先。

### 如何缩短周转时间? SJF: 短作业优先



- $P_3$ 提前的完成时间较 $P_2$ 拖后的完成时间要小

- 如果调度结果是 $p_1p_2...p_n$ , 则平均周转时间为:  
$$p_1 + p_1 + p_2 + p_1 + p_2 + p_3 + \dots = \sum (n+1-i)p_i$$
  
如果存在 $i < j$ 而 $p_i > p_j$ , 交换 $p_i, p_j$ 会怎么样?

等待时间

可以证明, 短作业优先会使得平均周转时间最小  
因为如果长作业先, 会使得后面的每个短作业都要等待这个长作业的时间, 会导致周转时间急剧上升

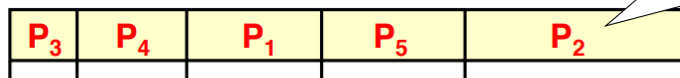


## 4. 调度算法2--时间片轮转(RR:round robin)--考虑响应时间

SJF会导致有些作业的响应时间无法接受, 因此需要考虑响应时间。

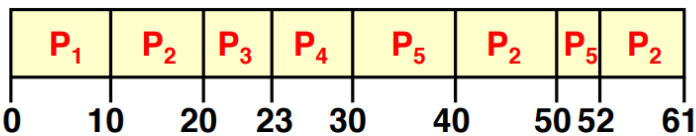
# 响应时间该怎么办？

P<sub>2</sub>用户的操作...



■ 如何解决？开动脑筋...

■ **RR: 按时间片来轮转调度**



■ 时间片大: 响应时间太长；时间片小: 吞吐量小

■ 折衷: 时间片**10-100ms**，切换时间**0.1-1ms(1%)**

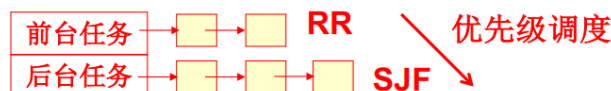
■ 如果**CPU**更快，时间片会怎么样？

## 5.很多问题需要折中

■ **Word**很关心响应时间，而**gcc**更关心周转时间，两类任务同时存在怎么办？

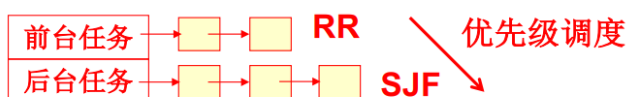
■ 一个调度算法让多种类型的任务同时都满意，怎么办？

■ **直观想法**: 定义前台任务和后台任务两队列，前台**RR**，后台**SJF**，只有前台任务没有时才调度后台任务



■ 但是这会产生很多问题？想一想会出现什么问题？

如果一直有前台任务...



一个故事: 1973年关闭的MIT的IBM 7094时，发现有一个进程在1967年提交但一直未运行

■ 后台任务可能一直得不到运行

■ **后台任务优先级动态升高**，但后台任务(用**SJF**调度)一旦执行，前台的响应时间...

■ **前后台任务都用时间片**，但又退化为了**RR**，后台任务的**SJF**如何体现？前台任务如何照顾？

我们知道RR可以有满意的响应时间，因此适合前台任务；而SJF可以有最短的周转时间，因此适合后台任务。

- 但如果前台RR后台SJF，先完成前台再完成后台；会导致后台任务饥饿(即一直得不到执行)
- 如果我们设计优先级，后台不执行的优先级可以慢慢升高，但后台任务使用SJF，一旦执行时间长，前台任务的响应时间就爆炸了。所以还是需要为后台任务设计时间片。
- 因此，如何基于RR设计适合前台后台任务的调度，又如何区分任务是前后台，成为问题

问题在下节课的调度算法会讲到

