

网址：www.icourses.cn，主页搜索“苏曙光”即可进入MOOC课堂

第6章 进程调度

-  6.1进程调度概念
-  6.2典型调度算法
-  6.3Linux进程调度

华中科技大学.苏曙光老师.《操作系统原理》MOOC课程组版权所有

网址：www.icourses.cn，主页搜索“苏曙光”即可进入MOOC课堂

《操作系统原理》

6.2 典型调度算法

教师：苏曙光

华中科技大学软件学院



华中科技大学.苏曙光老师.《操作系统原理》MOOC课程组版权所有

网址：www.icourses.cn，主页搜索“苏曙光”即可进入MOOC课堂

1.先来先服务调度 (First Come First Serve)



版权所有

网址：www.icourses.cn，主页搜索“苏曙光”即可进入MOOC课堂

1.先来先服务调度 (First Come First Serve)

■ 算法

- 按照作业进入系统的时间先后来挑选作业。先进入系统的作业优先被运行。

■ 特点

- 容易实现，效率不高
- 只考虑作业的等候时间，而没考虑运行时间的长短。因此一个晚来但是很短的作业可能需要等待很长时间才能被运行，因而本算法不利于短作业。

华中科技大学.苏曙光老师.《操作系统原理》MOOC课程组版权所有

网址：www.icourses.cn，主页搜索“苏曙光”即可进入MOOC课堂

1.先来先服务调度 (First Come First Serve)



版权所有

网址：www.icourses.cn，主页搜索“苏曙光”即可进入MOOC课堂

2. 短作业优先调度算法 (Short Job First)

■ 算法

- 参考运行时间，选取运行时间最短的作业投入运行。

■ 特点

- 易于实现，效率不高。
- 忽视了作业等待时间，一个早来但是很长的作业将会在很长时间得不到调度，易出现资源“饥饿”的现象。

网址：www.icourses.cn，主页搜索“苏曙光”即可进入MOOC课堂

3. 响应比高者优先调度算法

■ 响应比定义

- 作业的响应时间和与运行时间的比值
- 响应比 = 响应时间/运行时间
= (等待时间 + 运行时间) / 运行时间
= 1 + 等待时间 / 运行时间

■ 算法

- 计算每个作业的响应比，选择响应比最高的作业优先投入运行。

网址：www.icourses.cn，主页搜索“苏曙光”即可进入MOOC课堂

3.响应比高者优先调度算法

特点

- $\text{响应比} = 1 + \frac{\text{等待时间}}{\text{运行时间}}$
- 如果作业等待时间相同，则运行时间越短的作业，其响应比越高，因此越容易被调度。因而有利于短作业。
- 如果作业运行时间相同，则等待时间越长的作业，其响应比越高，因此越容易被调度。因而有利于等候长的作业。
- 对于运行时间长的作业，其优先级可以随等待时间的增加而提高，当其等待足够久的时候，也有可能获得CPU。

网址：www.icourses.cn，主页搜索“苏曙光”即可进入MOOC课堂

4.优先数调度算法

■ 算法

- 根据进程优先数，把CPU分配给最高的进程。
- 进程优先数 = 静态优先数 + 动态优先数

■ 静态优先数

- 进程创建时确定，在整个进程运行期间不再改变。

■ 动态优先数

- 动态优先数在进程运行期间可以改变。

网址：www.icourses.cn，主页搜索“苏曙光”即可进入MOOC课堂

4.优先数调度算法

■ 静态优先数的确定

- 基于进程所需的资源多少
- 基于程序运行时间的长短
- 基于进程的类型[IO/CPU,前台/后台,核心/用户]

■ 动态优先数的确定

- 当使用CPU超过一定时长时；
- 当进行I/O操作后；
- 当进程等待超过一定时长时；

网址：www.icourses.cn，主页搜索“苏曙光”即可进入MOOC课堂

5.循环轮转调度法 (ROUND-ROBIN)



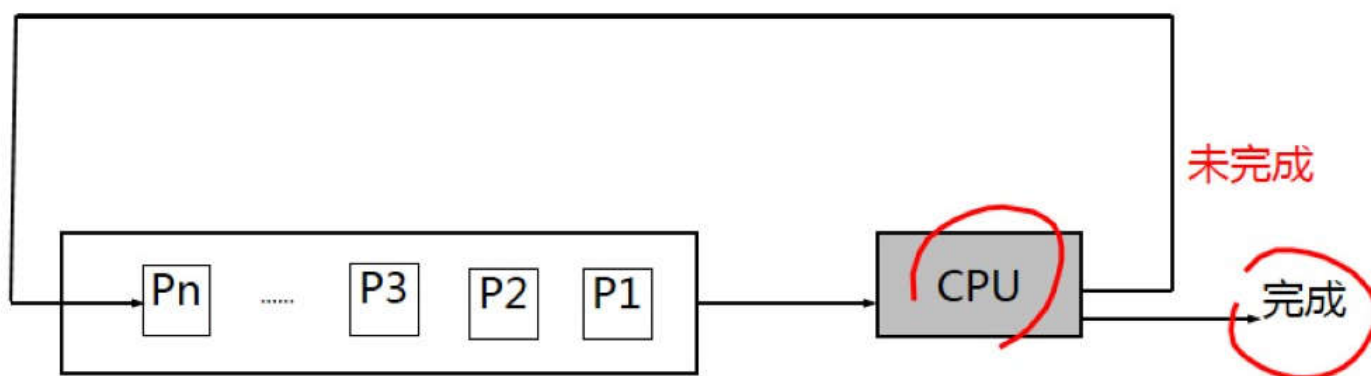
Round robin, too....

华中科技大学.苏曙光老师.《操作系统原理》MOOC课程组版权所有

网址：www.icourses.cn，主页搜索“苏曙光”即可进入MOOC课堂
循环轮转调度法 (ROUND-ROBIN)

概念

- 把所有就绪进程按先进先出的原则排成队列。新来进程加到队列末尾。
- 进程以时间片 q 为单位轮流使用CPU。刚刚运行一个时间片的进程排到队列末尾，等候下一轮运行。
- 队列逻辑上是环形的。



网址：www.icourses.cn，主页搜索“苏曙光”即可进入MOOC课堂

循环轮转调度法 (ROUND-ROBIN)

优点

- 公平性：每个就绪进程有平等机会获得CPU
- 交互性：每个进程等待 $(N-1) * q$ 的时间就可以重新获得CPU

时间片 q 的大小

- 如果 q 太大

- ◆ 交互性差
- ◆ 甚至退化为FCFS调度算法。

- 如果 q 太小

- ◆ 进程切换频繁，系统开销增加。

网址：www.icourses.cn，主页搜索“苏曙光”即可进入MOOC课堂
循环轮转调度法 (ROUND-ROBIN)

改进

- 时间片的大小可变
- 组织多个就绪队列