

操作系统原理

Operating System Principle

田丽华

5-3 SJF (Shortest-Job-First)

Shortest-Job-First (SJF) Scheduling

Associate with each process the length of its next CPU burst. Use these lengths to schedule the process with the shortest time.

(关联到每个进程下次运行的CPU脉冲长度, 调度最短的进程)

Shortest-Job-First (SJF) Scheduling

Two schemes:

nonpreemptive – once CPU given to the process it cannot be preempted until completes its CPU burst (非抢占式调度 – 一旦进程拥有CPU, 它的使用权限只能在该CPU 脉冲结束后让出).

Preemptive – if a new process arrives with CPU burst length less than remaining time of current 法executing process, preempt. This scheme is know as the Shortest-Remaining-Time-First (SRTF). (抢占式调度 – 发生在有比当前进程剩余时间片更短的进程到达时, 也称为最短剩余时间优先调度)

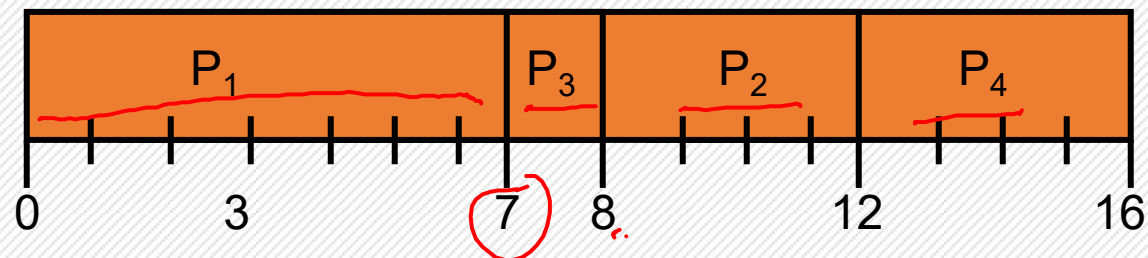
Shortest-Job-First (SJF) Scheduling

SJF is optimal – gives minimum average waiting time for a given set of processes. (SJF是最优的 – 对一组指定的进程而言, 它给出了最短的平均等待时间)

Example of Non-Preemptive SJF

Process	Arrival Time	Burst Time
P_1	0.0	7
P_2	2.0	4
P_3	4.0	1
P_4	5.0	4

01 SJF (non-preemptive)



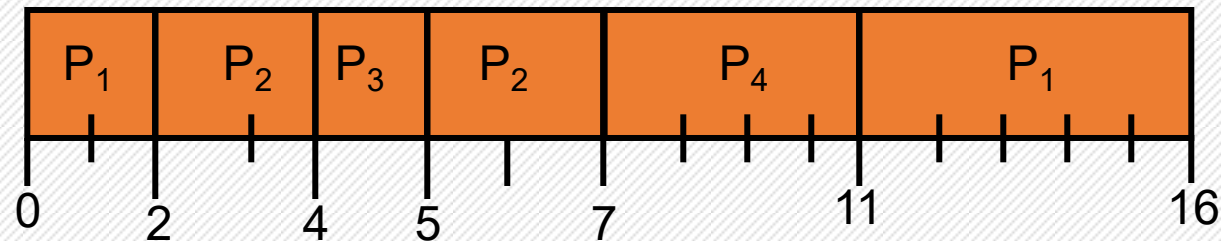
02

$$\text{Average waiting time} = (0 + 6 + 3 + 7)/4 = 4$$

Example of Preemptive SJF

Process	Arrival Time	Burst Time
P_1	0.0	7
P_2	2.0	4
P_3	4.0	1
P_4	5.0	4

01 SJF (preemptive)



02 Average waiting time = $(9 + 1 + 0 + 2)/4 = 3$

Determining Length of Next CPU Burst

- Although the SJF algorithm is optimal, it cannot be implemented at the level of short_term CPU scheduling.
- Can only estimate the length (其长度只能估计) .
- Can be done by using the length of previous CPU bursts, using exponential averaging (可以通过先前的CPU脉冲长度及计算指数均值进行) .

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

$$\tau_{n+1} = \alpha t_n + (1 - \alpha) \tau_n$$

Scheduling Algorithm

调度算法

Shortest-Job-First (SJF) Scheduling

01

采用SJF有利于系统减少平均周转时间,提高系统吞吐量。

02

一般情况下SJF调度算法比FCFS调度算法的效率要高一些,但实现相对要困难些。

03

如果作业的到来顺序及运行时间不合适,会出现饥饿现象,例如,系统中有一个运行时间很长的作业JN,和几个运行时间小的作业,然后,不断地有运行时间小于JN的作业的到来,这样,作业JN就因得不到调度而饿死。另外,作业运行的估计时间也有问题。