

《操作系统》

进程与线程

首都师范大学
信息工程学院
霍其润

Processes and Threads

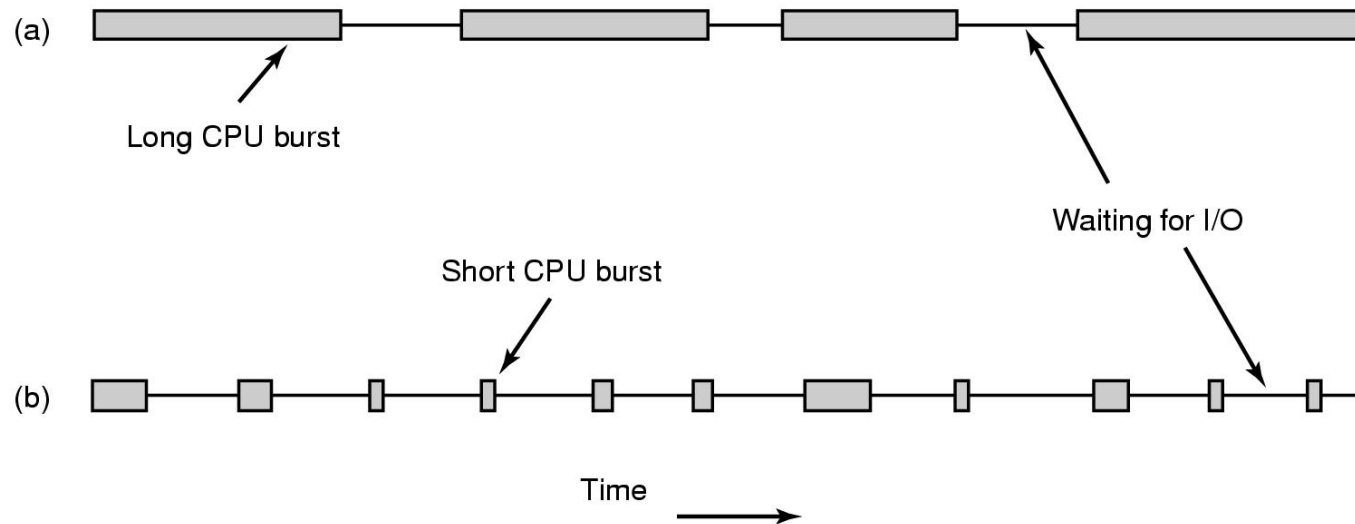
- ❖ Processes
- ❖ Threads
- ❖ Interprocess communication
- ❖ Classical IPC problems
- ❖ Scheduling



五 调度 (Scheduling)

1、调度简介

Introduction to Scheduling



- Bursts of CPU usage alternate with periods of I/O wait
 - a CPU-bound process
 - an I/O bound process

Introduction to Scheduling

- When to schedule
 - a new process is created
 - a process exits
 - a process blocks
 - an I/O interrupt occurs
 - preemptive/nonpreemptive
- Categories of Scheduling Algorithms
 - Batch
 - Interactive
 - Real-Time

All systems

Fairness - giving each process a fair share of the CPU

Policy enforcement - seeing that stated policy is carried out

Balance - keeping all parts of the system busy

Batch systems

Throughput - maximize jobs per hour

Turnaround time - minimize time between submission and termination

CPU utilization - keep the CPU busy all the time

Interactive systems

Response time - respond to requests quickly

Proportionality - meet users' expectations

Real-time systems

Meeting deadlines - avoid losing data

Predictability - avoid quality degradation in multimedia systems

Scheduling Algorithm Goals

平均周转时间

$$T = \frac{1}{n} \sum_{i=1}^n (t_{fi} - t_{bi})$$

平均带权周转时间

$$W = \frac{1}{n} \sum_{i=1}^n (t_{fi} - t_{bi}) / t_{si}$$

其中：

n 为单位时间内的作业数量。

t_{fi} 为作业 i 的完成时间。

t_{bi} 为作业的开始时间。

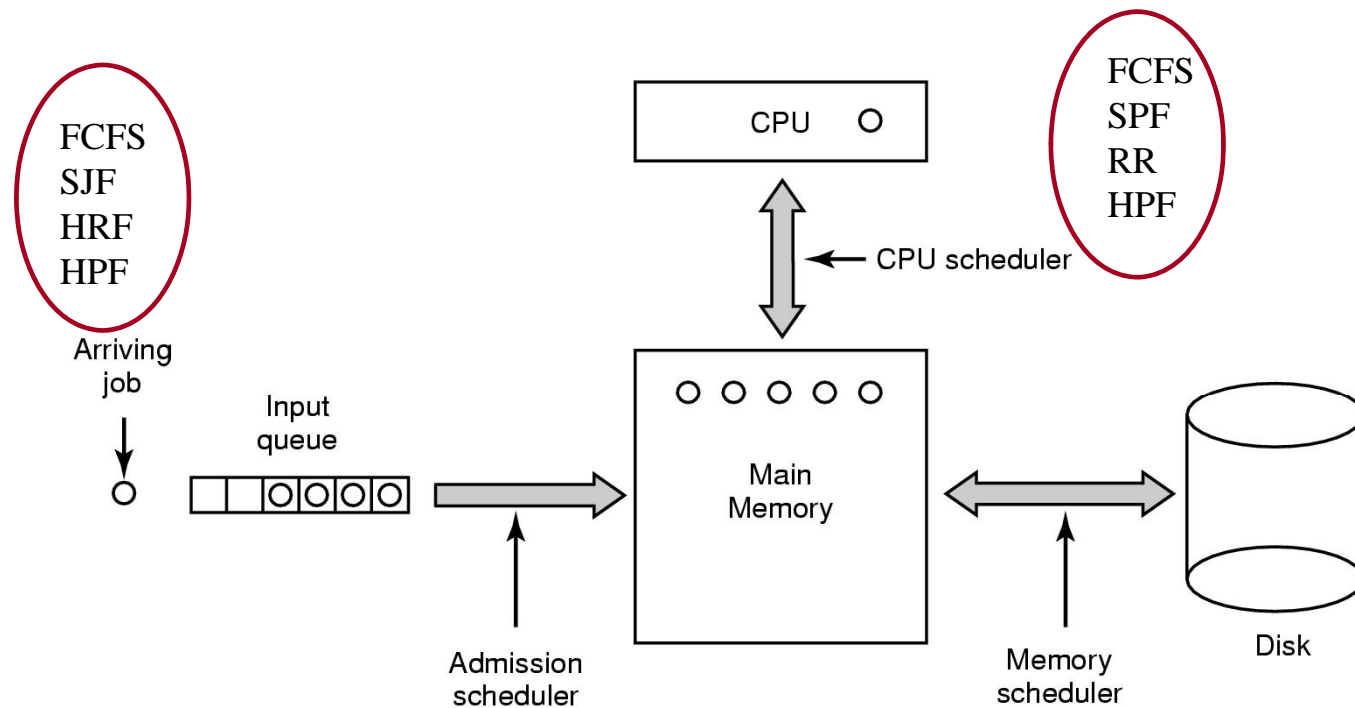
t_{si} 为作业 i 的运行时间。



五 调度 (Scheduling)

2、批处理系统调度

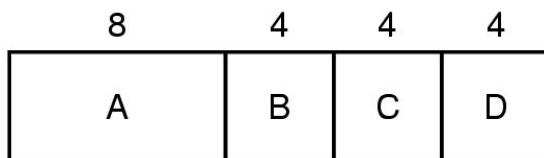
Scheduling in Batch Systems



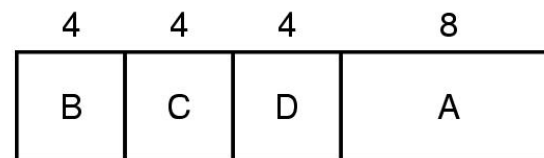
Three level scheduling

Scheduling in Batch Systems

- FCFS
- SJF
- HRF 响应比= (等待时间+估计运行时间) / 估计运行时间
- HPF
- SRF



(a)



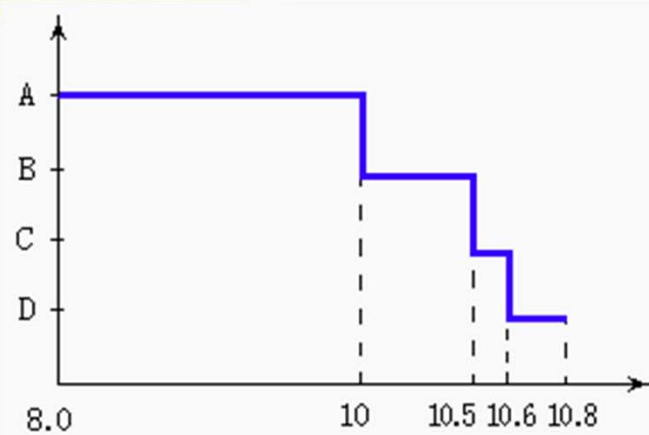
(b)

An example of shortest job first scheduling

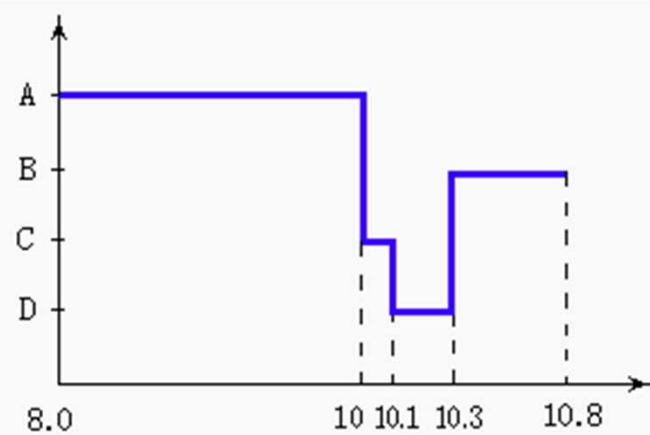
例子：

通过作业平均周转时间 T 和平均带权周转时间 W 来分析一下作业调度中的4种算法的性能。为了便于分析，我们设想的是一个单道批处理系统。选用的例子中有4个作业，按照下图所示。

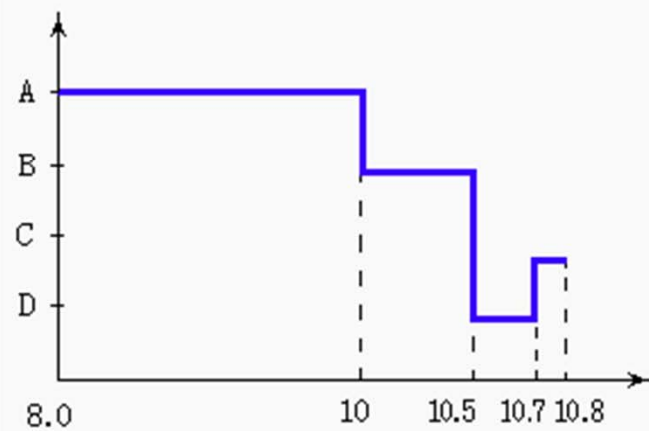
其中，作业的到达顺序为A, B, C, D。时间为：8, 8.5, 9, 9.5。运行时间：2, 0.5, 0.1, 0.2。优先级为：10, 25, 7, 18。



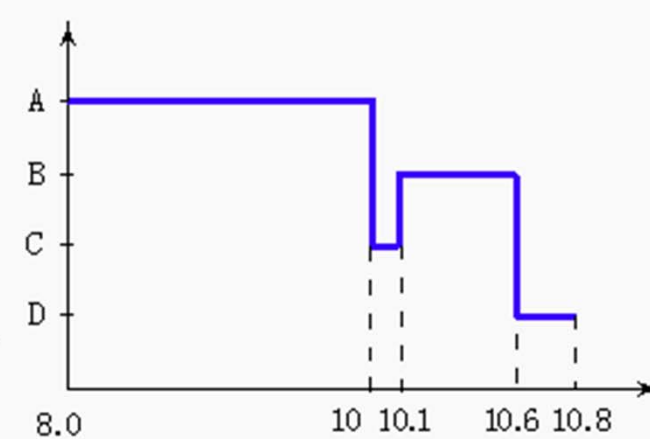
(a) FCFS 算法



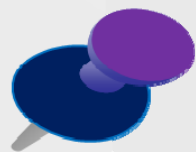
(b) SJF 算法



(c) HPF 算法



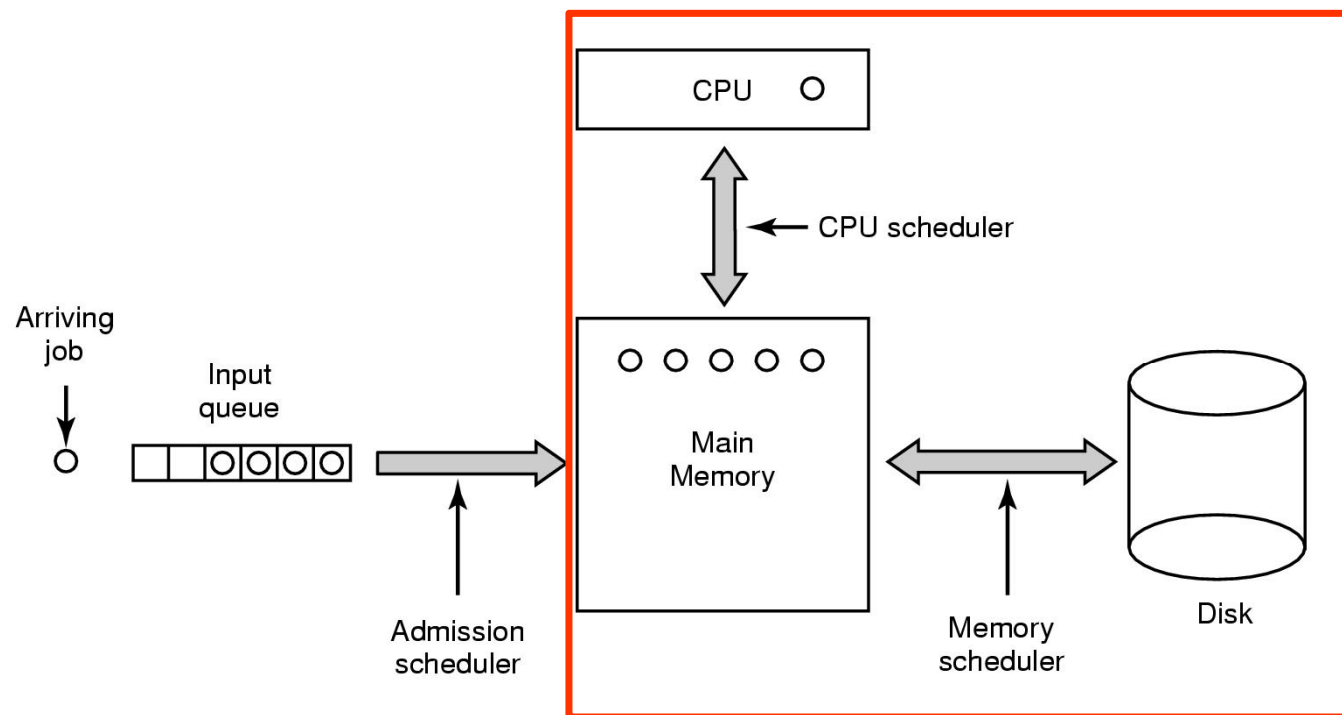
(d) HRF 算法



五 调度 (Scheduling)

3、交互式系统调度

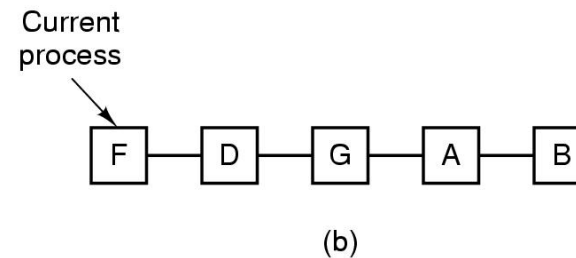
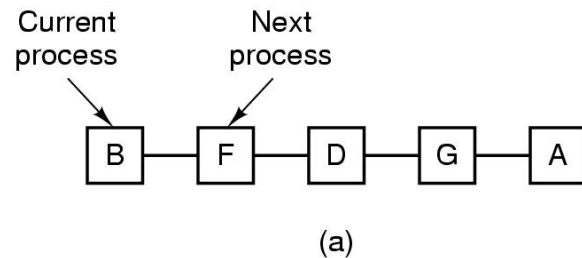
Scheduling in Interactive Systems



Scheduling in Interactive Systems

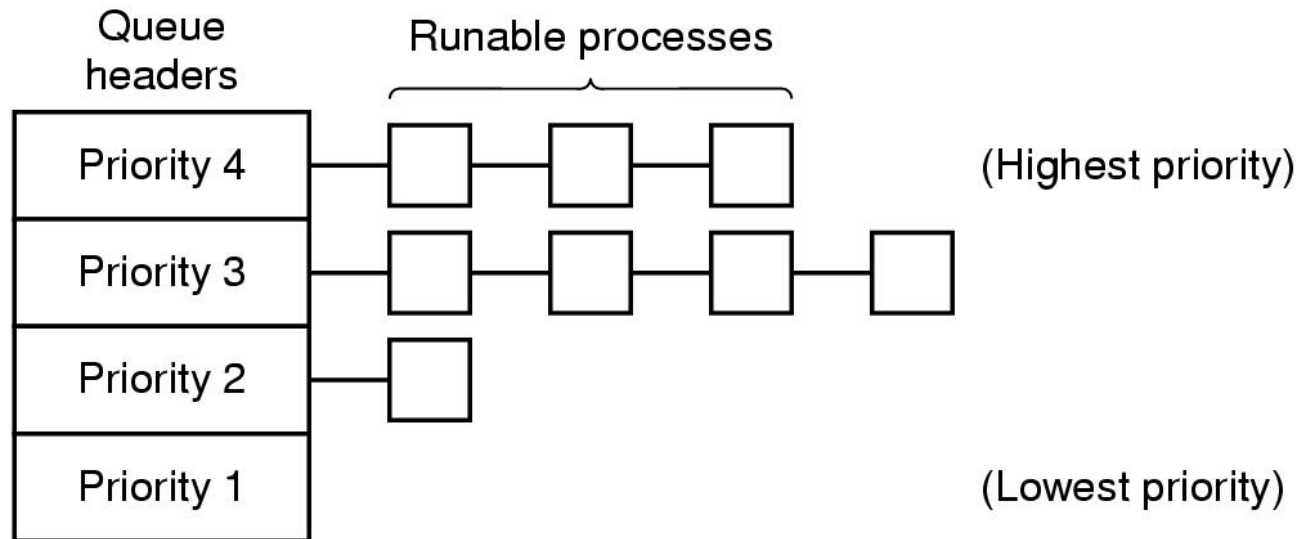
- Round-robin scheduling
- Priority scheduling
- Multiple queues
- Shortest process next
- Guaranteed scheduling
- Lottery scheduling
- Fair-share scheduling

Scheduling in Interactive Systems (1)



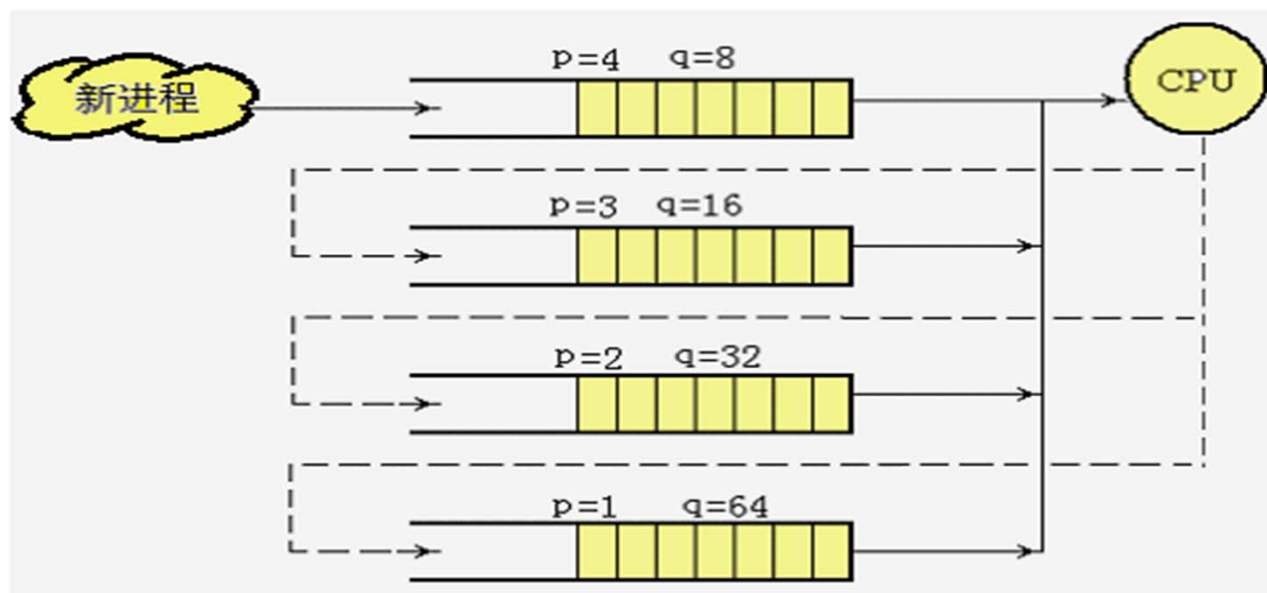
- Round Robin Scheduling
 - list of runnable processes
 - list of runnable processes after B uses up its quantum

Scheduling in Interactive Systems (2)



A scheduling algorithm with four priority classes

Scheduling in Interactive Systems (3)



Multiple Queues

Scheduling in Interactive Systems

- Round-robin scheduling
- Priority scheduling
- Multiple queues
- **Shortest process next**
- **Guaranteed scheduling**
- **Lottery scheduling**
- **Fair-share scheduling**



五 调度 (Scheduling)

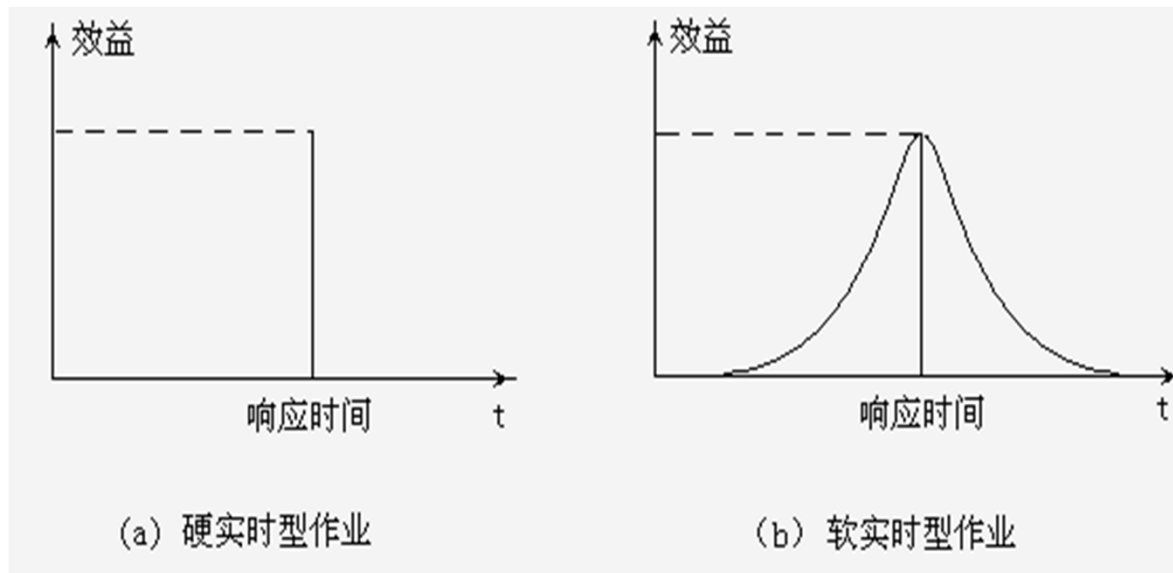
4、实时系统的调度

Scheduling in Real-Time Systems

- 要求更详细的调度信息：如，就绪时间、开始或完成截止时间、处理时间、资源要求、绝对或相对优先级（硬实时或软实时），优先级用户可控。
- 采用抢先式调度
- 快速中断响应：在中断处理时（硬件）关中断的时间尽量短。
- 快速上下文切换：相应地采用较小的调度单位（如线程）。提高响应时间。

Scheduling in Real-Time Systems

1、硬实时和软实时



2、周期性和非周期性

Scheduling in Real-Time Systems

周期性任务调度

- 在一些信号检测和过程控制系统中，有许多任务呈现周期性的运行规律。比如，气象信息检测中每隔2小时要读取一次数据，窑炉控制中每隔5分钟需检测一次炉温。这些任务的共同特点是，周期性强，而且有固定的时间间隔和相同的工作流程。通常，我们将这类任务称为周期性任务。目前，生产过程中的大多数数据信号采集系统，精密的过程检测与控制系统都属于此类。
- 一个周期性任务进入时，需要向系统提交的信息有：代码长度和资源需求，还要包括间隔周期和每个周期内的执行时间等。

Scheduling in Real-Time Systems

Schedulable real-time system

- Given
 - m periodic events
 - event i occurs within period P_i and requires C_i seconds
- Then the load can only be handled if

$$\sum_{i=1}^m \frac{C_i}{P_i} \leq 1$$

Scheduling in Real-Time Systems

非周期性任务

- 紧迫型实时任务调度
 - 立即抢占式优先级调度
- 普通型的实时任务调度
 - 基于时钟中断的抢占式优先级调度
- 宽松型的实时任务调度
 - 非抢占的HPF调度算法
 - RR算法

Scheduling in Real-Time Systems

最早截止时间优先调度算法

根据任务的截止时间来确定任务的优先级，截止时间越早，其优先级越高，称为最早截止时间优先算法。

在系统中保持一个实时任务就绪队列，该队列按各任务截止时间的早晚排序，具有最早截止时间的任务排在队列的最前面。调度程序在选择任务时，总是选择队列中的第一个任务，为之分配处理机，使之投入运行。

该算法既可用于抢占式调度，也可用于非抢占式调度。

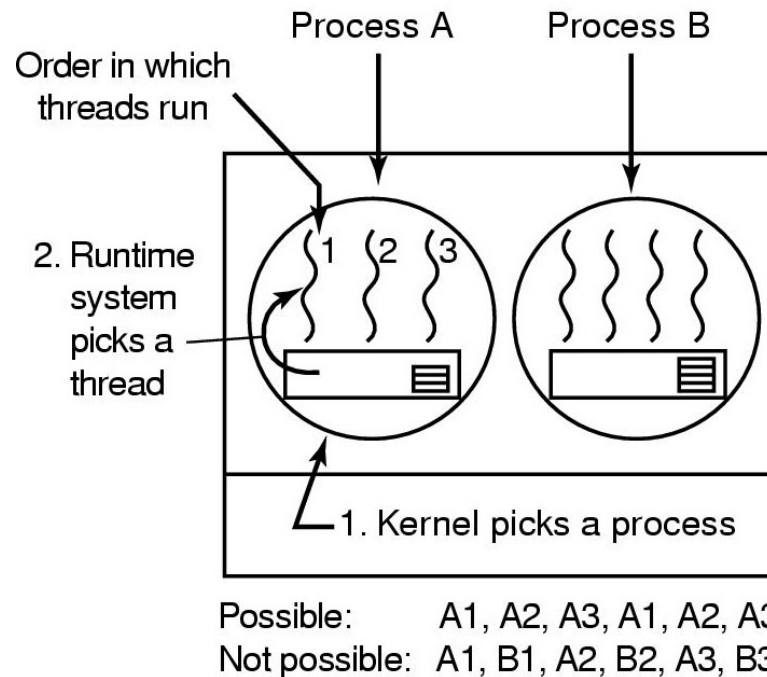


五 调度 (Scheduling)

5、线程调度

Thread Scheduling (1)

- 内核对线程不感知，故选择进程分配时间片；
- 用户空间的线程调度程序选择进程中某一线程执行；
- 无时钟中断，该线程可长期执行或执行完自动放弃，调度程序再选进程内其他线程；
- 线程调度算法可灵活设定；
- 线程阻塞——整个进程阻塞。

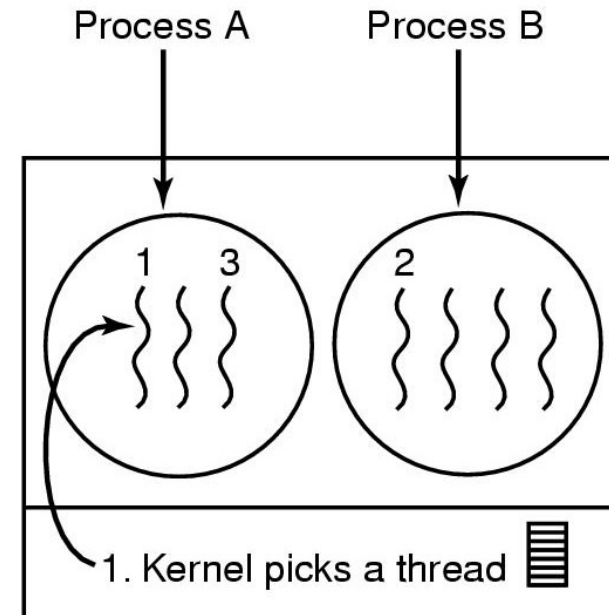


Possible scheduling of user-level threads

- 50-msec process quantum
- threads run 5 msec

Thread Scheduling (2)

- 内核选择线程分配时间片；
- 有时钟中断，时间片结束，该线程被切换；
- 线程结束/阻塞，可能调度相同/不同进程中的线程；
- 考虑到线程切换的开销，系统同等条件下会优先选择同一进程的线程；
- 线程阻塞——该进程不会阻塞。



Possible: A1, A2, A3, A1, A2, A3
Also possible: A1, B1, A2, B2, A3, B3

Possible scheduling of kernel-level threads

- 50-msec process quantum
- threads run 5 msec

小 结

- 调度的分类和目标
- 调度的层次
- 常用的调度算法
- 线程调度