第17 - 18讲 实时系统与实时任务调度

§2.5 Real-Time Scheduling

Real-Time Systems

- Correctness of the system depends not only on the logical result of the computation but also on the time at which the results are produced.
- Tasks or processes attempt to control or react to events that take place in the outside world.
- These events occur in “real time” and process must be able to keep up with them.

Real-Time Systems

- Control of laboratory experiments
- Process control plants
- Robotics
- Air traffic control
- Telecommunications
- Military command and control systems

Real-Time Systems

指能及时响应外部事件的请求，在规定的时间内完成对该事件的处理，并控制所有实时任务协调一致地运行的计算机系统。

Real-Time Systems

- ***实时控制系统***，指要求进行实时控制的系统。用于生产过程的控制，实时采集现场数据，并对所采集的数据进行及时处理 。如飞机的自动驾驶系统，以及导弹的制导系统等。
- ***实时信息处理系统***，指能对信息进行实时处理的系统。典型的实时信息处理系统有：飞机订票系统、情报检索系统等。

Real-time Task

- 按任务执行时是否呈现周期性来划分
  - periodic （周期性）实时任务
  - aperiodic （非周期性）实时任务，必须联系着一个 deadline
- 根据对截止时间的要求来划分
  - hard real-time task (硬实时任务)，系统必须满足任务对截止时间的要求，否则可能出现难以预测的结果
  - soft real-time task (软实时任务)

Characteristics of Real-Time Operating Systems

- **Deterministic(确定性)**
  - Operations are performed at fixed, predetermined times or within predetermined time intervals.
  - Concerned with how long the operating system delays before acknowledging an interrupt.

Characteristics of Real-Time Operating Systems

- Responsiveness(响应性)
  - How long, after acknowledgment, it takes the operating system to service the interrupt.
  - Includes amount of time to begin execution of the interrupt.
  - Includes the amount of time to perform the interrupt.

Characteristics of Real-Time Operating Systems

- User control
  - User specifies priority
  - Specify paging （存储分页）
  - What processes must always reside in main memory.
  - Disks algorithms to use
  - Rights of processes

Characteristics of Real-Time Operating Systems

- Reliability （可靠性）
  - Degradation(下降) of performance may have catastrophic consequences （灾难性的后果）．
  - Attempt either to correct the problem or minimize its effects while continuing to run.
  - Most critical, high priority tasks execute.

Features of Real-Time Operating Systems

- Fast context switch
- Small size
- Ability to respond to external interrupts quickly.
- Multitasking with interprocess communication tools such as semaphores, signals and events.
- Files that accumulate （存储） data at a fast rate.

Features of Real-Time Operating Systems

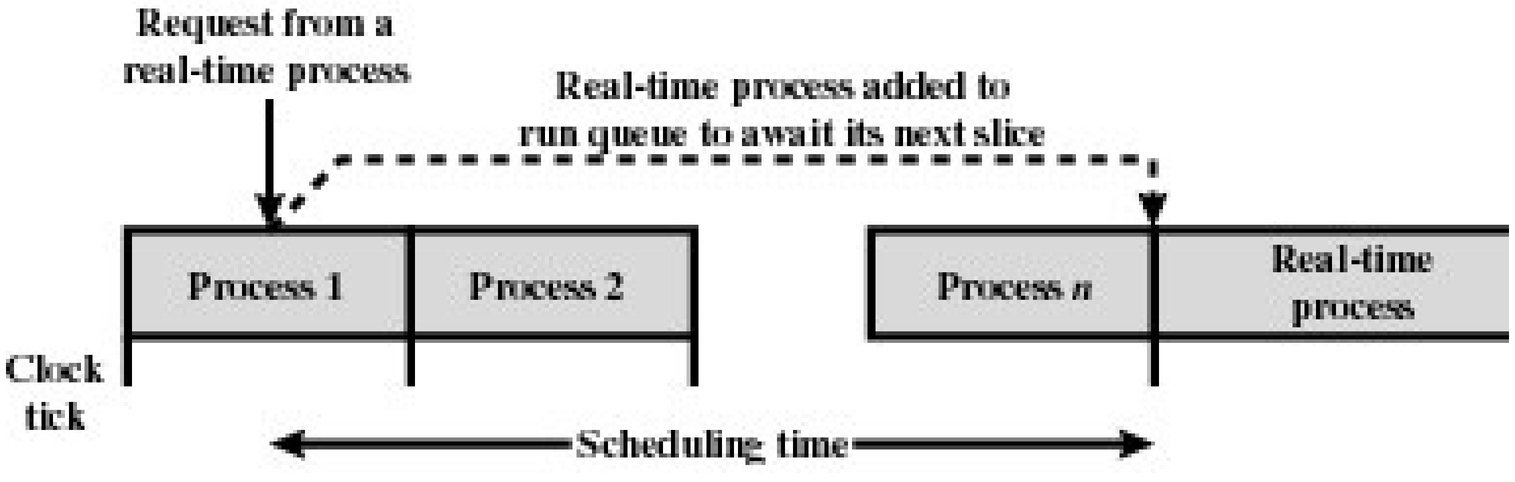
- Use of special sequential files that can accumulate data at a fast rate.
- Preemptive scheduling base on priority.
- Minimization of intervals during which interrupts are disabled.
- Delay tasks for fixed amount of time.
- Special alarms and timeouts （报警和超时处理 ）.

- Scheduling of a Real-Time Process
- Real-Time Scheduling
- Deadline Scheduling
- Rate Monotonic Scheduling

Scheduling of a Real-Time Process

- Round Robin Preemptive Scheduler （基于时间片的轮转调度法）
- Priority-driven Nonpreemptive Scheduler （基于优先级非剥夺调度法）
- Priority-driven Preemptive Scheduler （基于优先级的剥夺调度法）
- Immediate Preemptive Scheduler （立即剥夺调度法）

Round Robin Preemptive Scheduler
（基于时间片的轮转调度法）


Request from a real-time process
Real-time process added to run queue to await its next slice
Process 1
Process 2
Process *n*
Real-time process
Clock tick
Scheduling time

(a) Round-robin Preemptive Scheduler

Figure 10.4 Scheduling of Real-Time Process

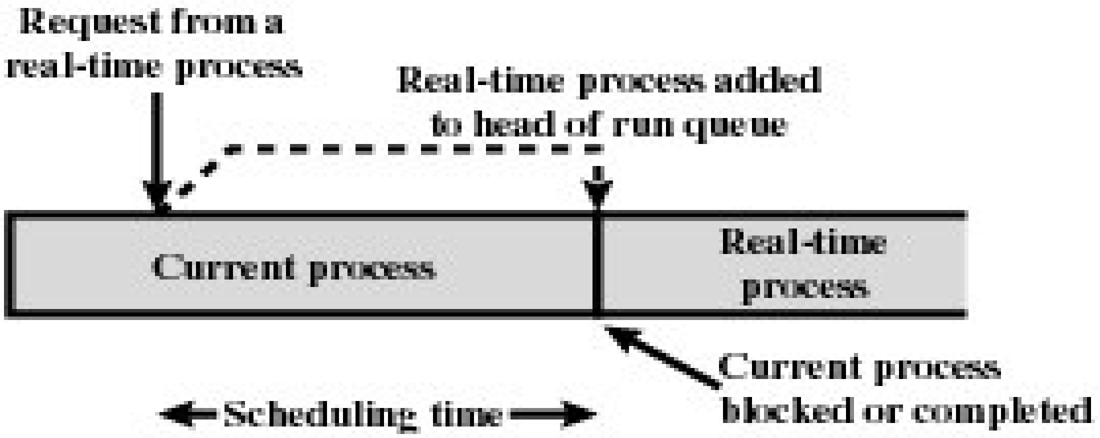
Round Robin Preemptive Scheduler （基于时间片的轮转调度法）

- 响应时间在<u>**秒级**</u>
- 广泛应用于分时系统，也可用于一般的实时信息处理系统
- 不适合于要求严格的实时控制系统。

Priority-driven Nonpreemptive Scheduler
（基于优先级非剥夺调度法）

为实时任务赋予较高的优先级，将它插入就绪队列队首，只要正在执行的进程释放Processor，则立即调度该实时任务执行。

Priority-driven Nonpreemptive Scheduler
（基于优先级非剥夺调度法）


Request from a real-time process
Real-time process added to head of run queue
Current process
Real-time process
Current process blocked or completed
Scheduling time

(b) Priority-Driven Nonpreemptive Scheduler

Figure 10.4 Scheduling of Real-Time Process

Priority-driven Nonpreemptive Scheduler
（基于优先级非剥夺调度法）

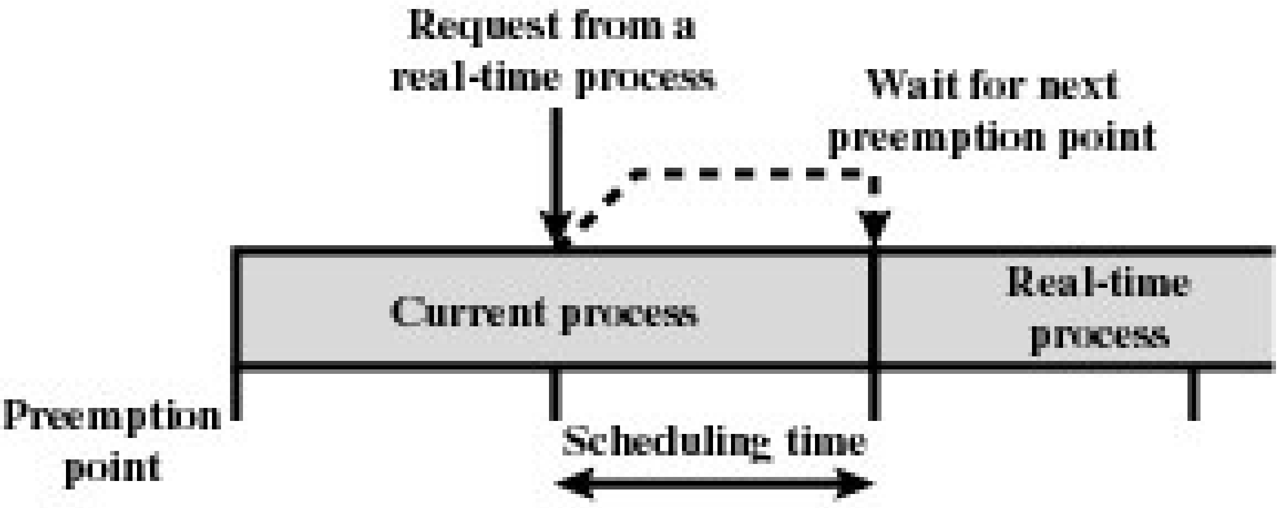
- 响应时间一般在*数百毫秒至数秒*范围。
- 多用于多道批处理系统，也可以用于要求不太严格的实时系统。

Priority-driven Preemptive Scheduler
（基于优先级剥夺调度法）

- 当实时任务到达后，可以在时钟中断时，剥夺正在执行的低优先级进程的执行，调度执行高优先级的任务

- 响应时间较短，一般在*几十毫秒或几毫秒*。

Priority-driven Preemptive Scheduler
（基于优先级剥夺调度法）



(c) Priority-Driven Preemptive Scheduler on Preemption Points

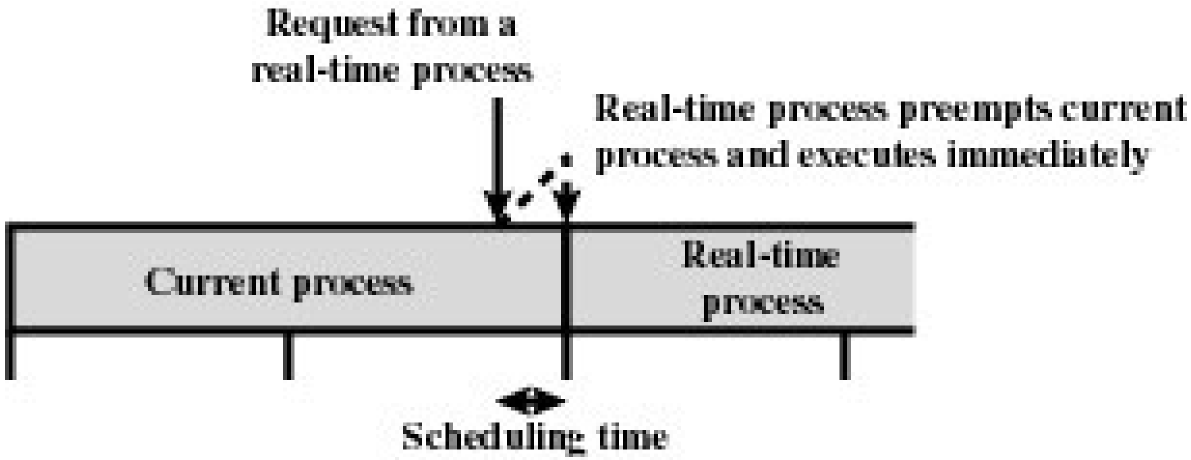
Figure 10.4 Scheduling of Real-Time Process

Immediate Preemptive Scheduler （立即剥夺调度法）

- 要求操作系统具有快速响应外部事件的能力。一旦出现外部中断，只要当前任务未处于临界区，便立即剥夺其执行，把处理机分配给请求中断的紧迫任务。

- 调度时延可以降至*100微秒，甚至更低。*

Immediate Preemptive Scheduler （立即剥夺调度法）


Request from a real-time process
Real-time process preempts current process and executes immediately
Current process
Real-time process
Scheduling time

(d) Immediate Preemptive Scheduler

Figure 10.4 Scheduling of Real-Time Process

- Scheduling of a Real-Time Process
- **Real-Time Scheduling**
- Deadline Scheduling
- Rate Monotonic Scheduling

Aim of Real-Time scheduling

- ***hard real-time task,*** 在其规定的截止时间内完成
- 尽可能使 ***soft real-time task*** 也能在规定的截止时间内完成。

公平性和最短平均响应时间等要求已不再重要。

但是，大多数现代实时操作系统无法直接处理任务的截止时间，它们只能尽量提高响应速度，以尽快地调度任务。

Real-Time scheduling

- Static table-driven （静态表驱动调度法）
  - Determines at run time when a task begins execution.
- Static priority-driven preemptive（静态优先级剥夺调度法）
  - Traditional priority-driven scheduler is used.
- Dynamic planning-based（动态计划调度法）
- Dynamic best effort （动态最大努力调度法）

Static table-driven approaches

- 用于调度*周期性实时任务*。
- 按照任务周期到达的时间、执行时间、完成截止时间（ending deadline）以及任务的优先级，*制订调度表*，调度实时任务。
- 最早截止时间优先（EDF）调度算法即属于此类。
- 此类算法不灵活，任何任务的调度申请改动都会引起调度表的修改。

Static priority-driven preemptive approaches

- 此类算法***多用于非实时多道程序系统***。
- 优先级的确定方法很多，例如在分时系统中，可以对 I/O bound 和 processor bound 的进程赋予不同的优先级。
- 实时系统中一般根据对***任务的限定时间***赋予优先级，例如速度单调算法（ RM ）即是为实时任务赋予静态优先级。

Dynamic planning-based approaches

- 当实时任务到达以后，系统为新到达的任务和正在执行的任务*动态创建一张调度表*。
- 在当前执行进程不会错过其截止时间的条件下，如果也能使新到达任务在截止时间内完成，则立即调度执行新任务。

Dynamic best effort approaches

- 实现简单，广泛用于*非周期性实时任务调度*。
  - 当任务到达时，系统根据其属性赋予优先级，优先级高的先调度。例如最早截止时间优先 EDF 调度算法就采用了这种方法。这种算法总是*尽最大努力尽早调度紧迫任务*，因此称为“最大努力调度算法”。
- 缺点在于，当任务完成，或截止时间到达时，很难知道该任务是否满足其约束时间。

- Scheduling of a Real-Time Process
- Real-Time Scheduling
- Deadline Scheduling
- Rate Monotonic Scheduling

Deadline Scheduling

Information used

- Ready time
- Starting deadline
- Completion deadline
- Processing time
- Resource requirements
- Priority
- Subtask scheduler： 一个任务可以分解出*强制子任务 (mandatory subtask)* 和*非强制子任务 (optional subtask)*。只有强制子任务拥有硬截止时间 (hard deadline) 。

Deadline Scheduling

Which task to schedule next?

- Scheduling tasks with the *earliest deadline* minimized the fraction of tasks that miss their deadlines

Deadline Scheduling

What sort of preemption is allowed?

- When starting deadlines are specified, then a nonpreemtive scheduler makes sense. 在执行完强制子任务或临界区后，阻塞自己。

- For a system with completion deadlines, a preemptive strategy is most appropriate.

Earliest Deadline (最早截止时间优先， 简称 ED)

- 常用调度算法
- 若指定任务的 <u>***Starting deadlines***</u>， 则采用 <u>***Nonpreemption***</u>， 当某任务的开始截止时间到达时，正在执行的任务必须执行完其强制部分或临界区，释放 CPU， 调度开始截止时间到的任务执行
- 若指定任务的 <u>***Completion deadlines***</u>， 则采用 <u>***Preemption***</u>

Periodic tasks with completion deadlines

- 由于此类任务是周期性的、可预测的，可采用***静态表驱动之最早截止时间优先调度算法***，使系统中的任务都能按要求完成。

- 举例：

  周期性任务A和B，指定了它们的完成截止时间，任务A每隔20毫秒完成一次，任务B每隔50毫秒完成一次。任务A每次需要执行10毫秒，任务B每次需要执行25毫秒

Two Tasks

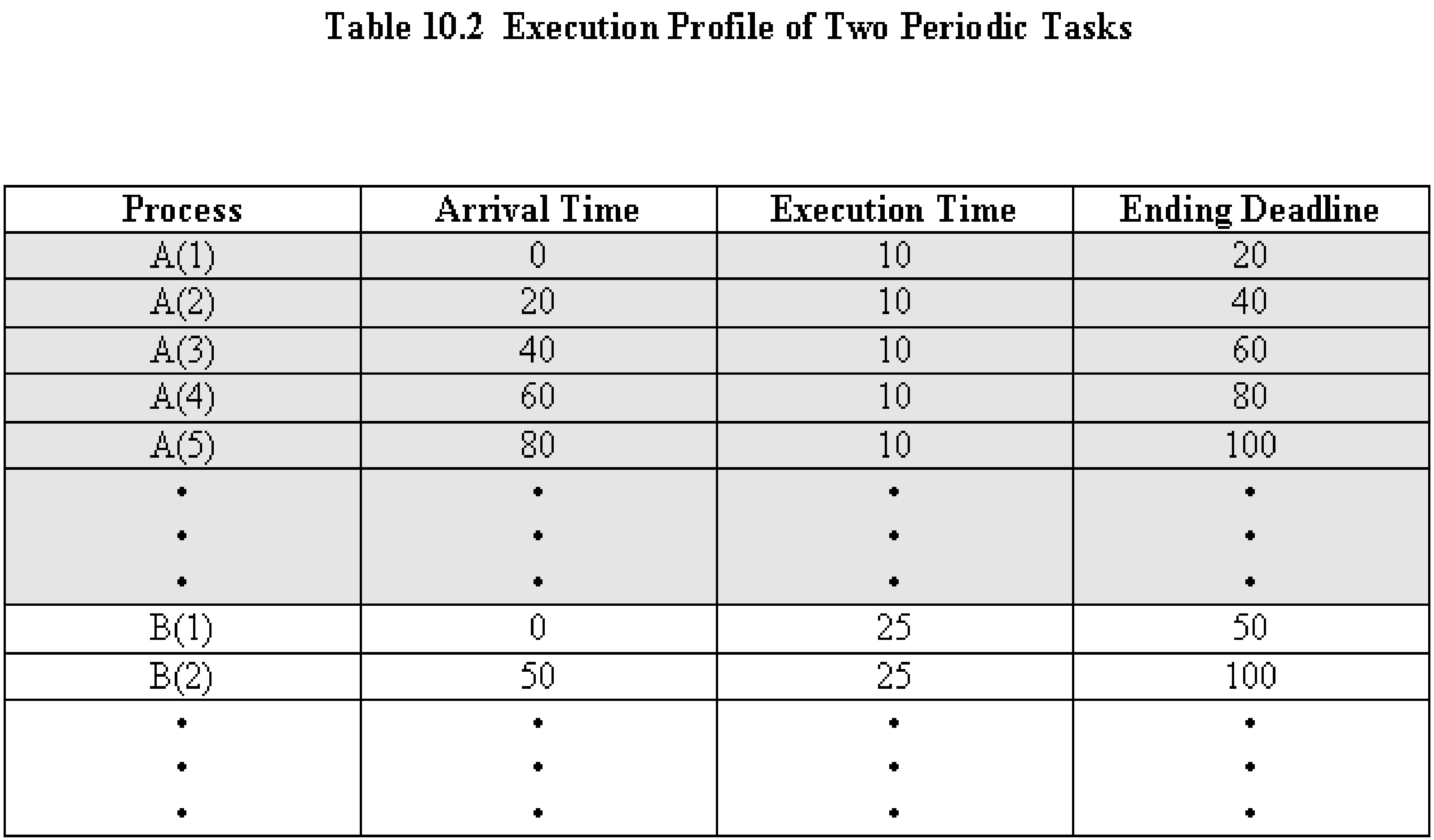


Table 10.2 Execution Profile of Two Periodic Tasks

Process	Arrival Time	Execution Time	Ending Deadline
A(1)	0	10	20
A(2)	20	10	40
A(3)	40	10	60
A(4)	60	10	80
A(5)	80	10	100
• • •	• • •	• • •	• • •
B(1)	0	25	50
B(2)	50	25	100
• • •	• • •	• • •	• • •

Periodic tasks with completion deadlines

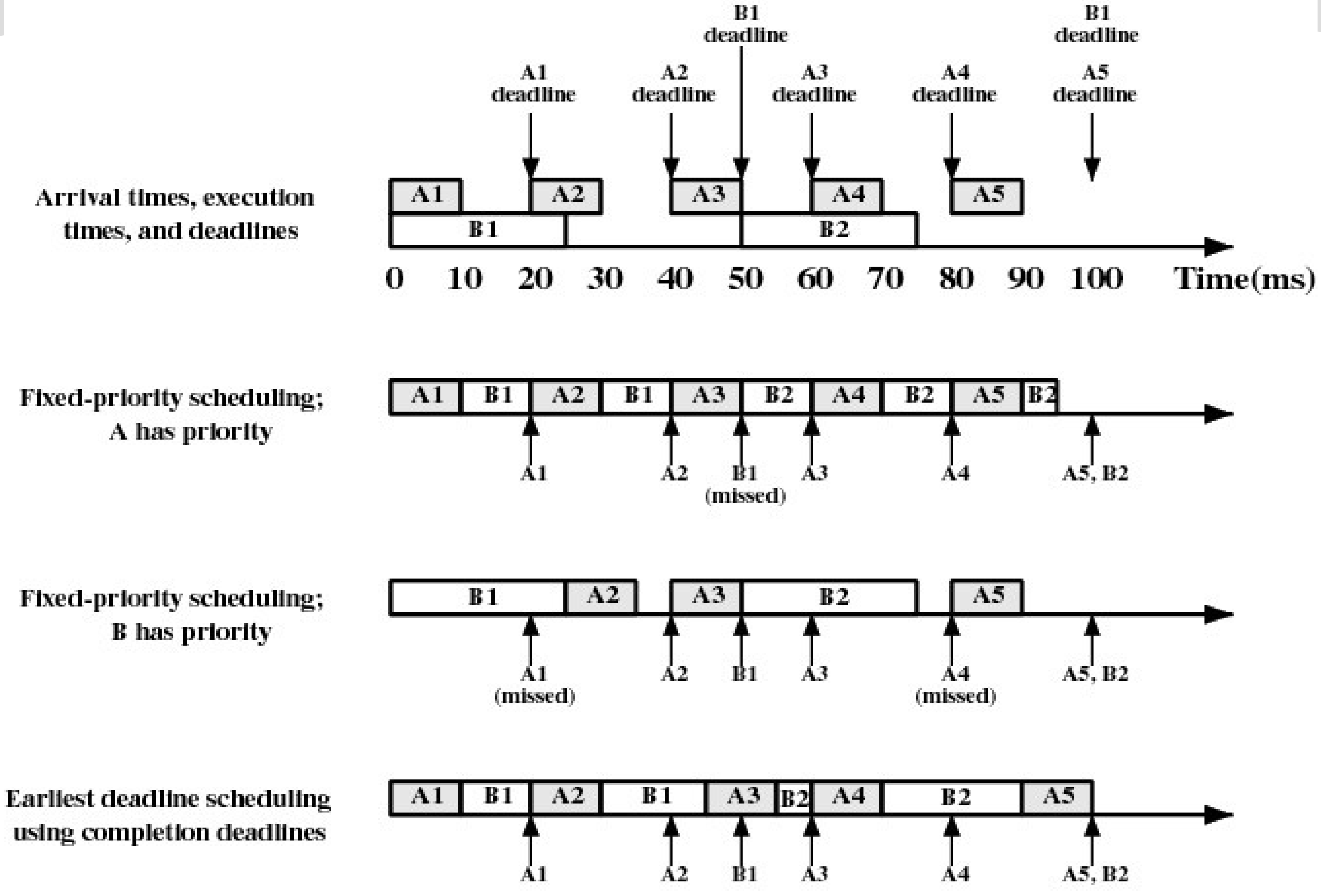

B1 deadline
B1 deadline
A1 deadline
A2 deadline
A3 deadline
A4 deadline
A5 deadline
Arrival times, execution times, and deadlines
A1
A2
A3
A4
A5
B1
B2
0
10
20
30
40
50
60
70
80
90
100
Time(ms)
Fixed-priority scheduling; A has priority
A1 B1 A2 B1 A3 B2 A4 B2 A5 B2
A1
A2
B1 (missed)
A3
A4
A5, B2
Fixed-priority scheduling; B has priority
B1 A2 A3 B2 A5
A1 (missed)
A2
B1
A3
A4 (missed)
A5, B2
Earliest deadline scheduling using completion deadlines
A1 B1 A2 B1 A3 B2 A4 B2 A5
A1
A2
B1
A3
A4
A5, B2

Figure 10.5 Scheduling of Periodic Real-time Tasks with Completion Deadlines

Aperiodic tasks with starting deadlines

- 可以采用*最早截止时间优先调度算法*或*允许 CPU 空闲的 EDF 调度算法*。
- *Earliest Deadline with Unforced Idle Times(允许 CPU 空闲的 EDF 调度算法）*，指优先调度最早截止时间的任务，并将它执行完毕才调度下一个任务。即使选定的任务未就绪，允许 CPU 空闲等待，也不能调度其他任务。尽管 CPU 的利用率不高，但这种调度算法可以保证系统中的任务都能按要求完成。

Aperiodic tasks with starting deadlines

Table 10.3

Execution profile of five aperiodic tasks

Process	Arrival Time	Execution Time	Starting Deadline
A	10	20	110
B	20	20	20
C	40	20	50
D	50	20	90
E	60	20	70

Aperiodic tasks with starting deadlines

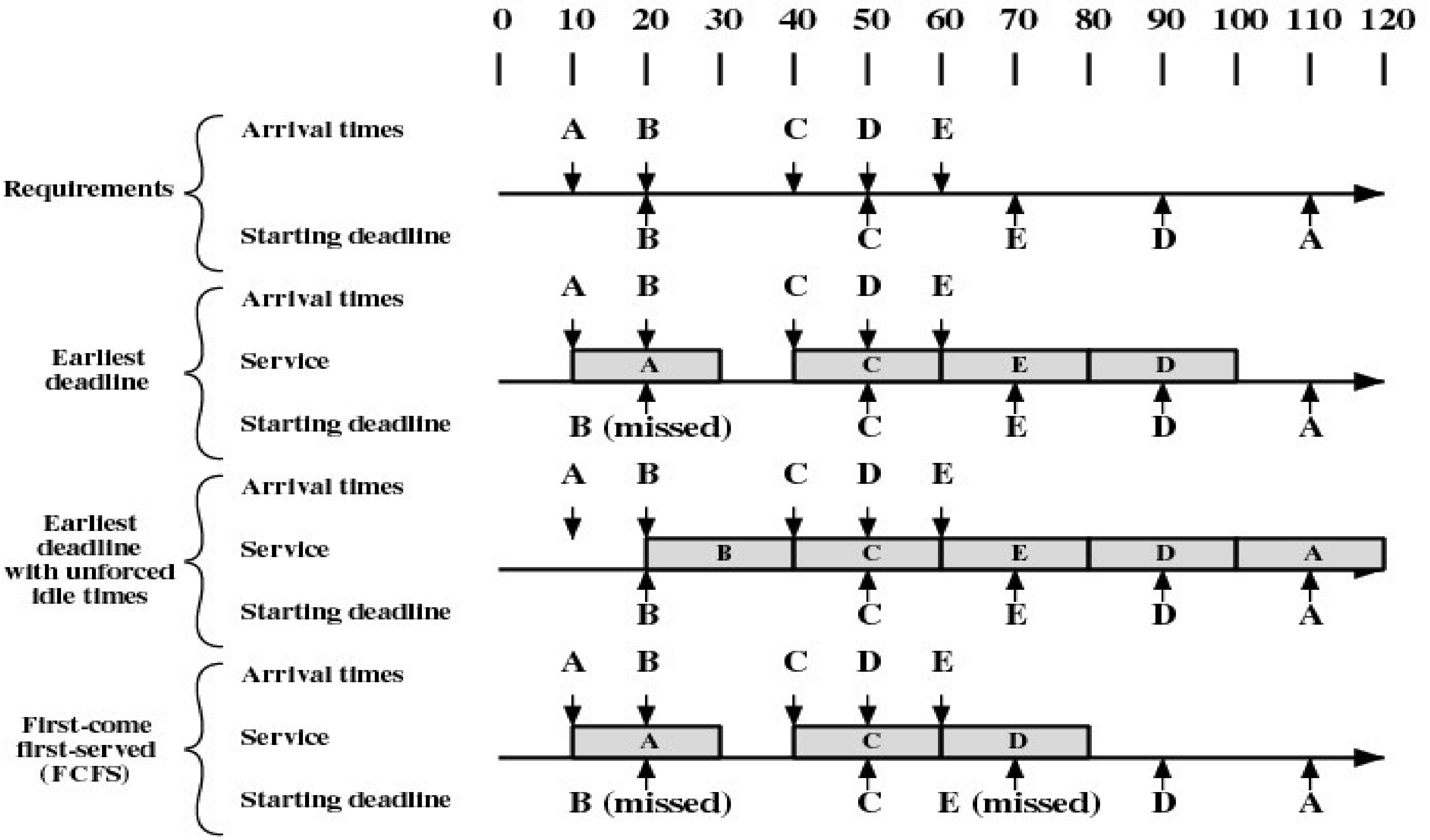


Figure 10.6 Scheduling of Aperiodic Real-time Tasks with Starting Deadlines

- Scheduling of a Real-Time Process
- Real-Time Scheduling
- Deadline Scheduling
- Rate Monotonic Scheduling

Rate Monotonic Scheduling (速度单调调度算法)

- Assigns priorities to tasks on the basis of their periods.
- Highest-priority task is the one with the shortest period.
- **Period （任务周期）， 指一个任务到达至下一任务到达之间的时间范围。**
- **Rate （任务速度）， 即周期（以秒计）的倒数,以赫兹为单位。**

Rate Monotonic Scheduling (速度单调调度算法)

- **任务周期的结束，**表示任务的硬截止时间。任务的执行时间不应超过任务周期
- CPU 的利用率 = 任务执行时间 / 任务周期
- 在 RMS 调度算法中，如果以任务速度为参数，则优先级函数是一个单调递增的函数

Periodic Task Timing Diagram

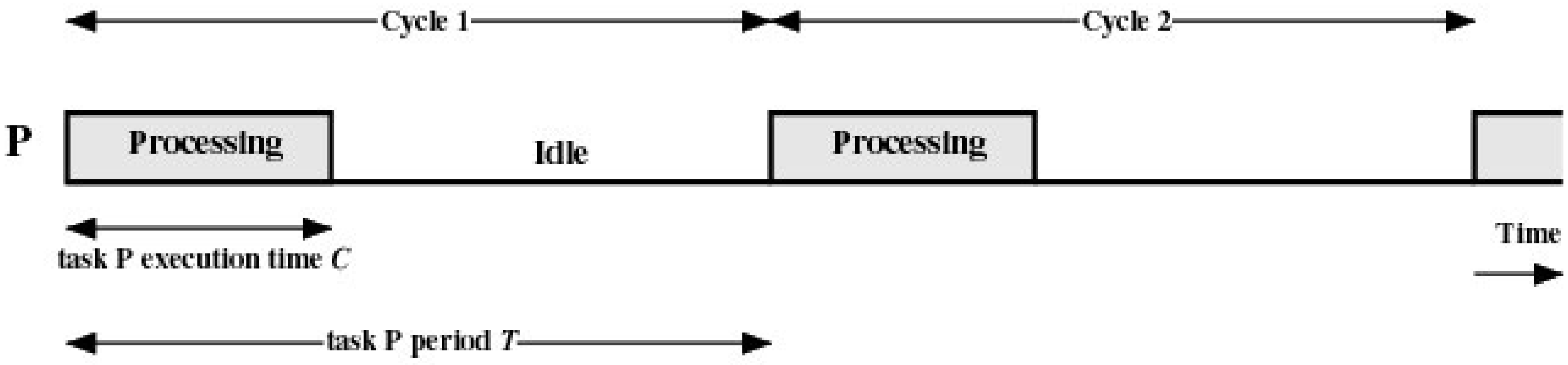


Figure 10.7 Periodic Task Timing Diagram

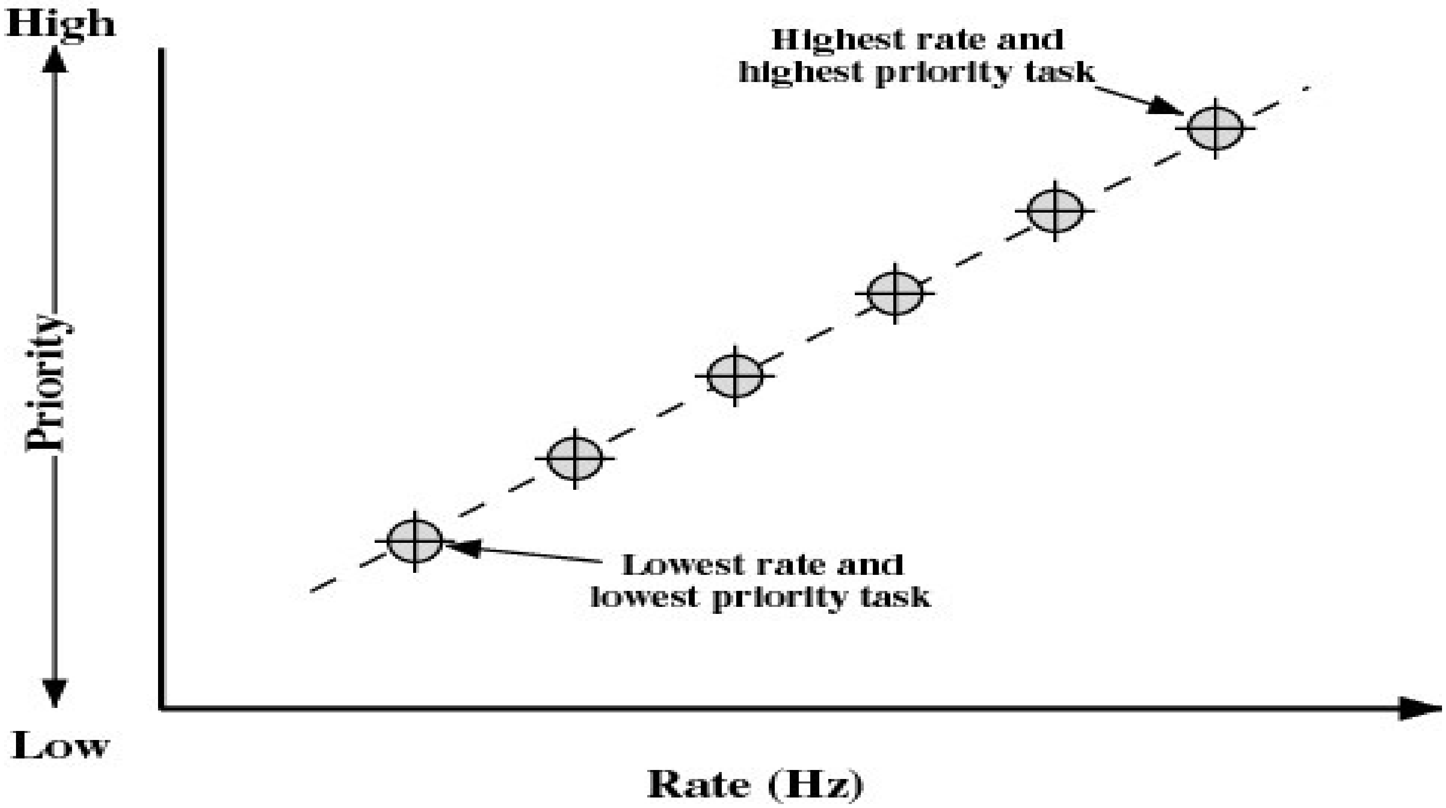


Figure 10.8 A Task Set with RMS [WARR91]

调度实例

Examples for Scheduling:

UNIX, Windows NT