

多媒体技术

第七章 多媒体操作系统

1. 概述
2. 计算机操作系统
3. 操作系统对连续媒体的支持
4. 支持连续媒体的文件系统

第七章 多媒体操作系统

1. 概述

操作系统控制和管理计算机硬件和软件资源，合理地组织计算机工作流程，方便用户程序的执行。操作系统提供与计算机关键资源相关的各种服务。

目前普通操作系统在多媒体应用中的不足：

- 1) 缺乏操作系统实时的支持。
- 2) 缺乏基于QoS的资源管理。QoS管理应能够使系统对活动的应用具有服务质量的保证。在这种模式下，并不是仅仅需要在会话开始的时候指定一个服务质量，并在整个会话生命周期中都维持这个服务质量。而是需要一种能够适应不同的环境和应用的动态QoS控制模式。

第七章 多媒体操作系统

1. 概述

目前普通操作系统的不足：

3) 对系统的输入输出缺乏有效的管理和控制。计算机的设计强调处理器性能，而对多媒体系统来说，I/O系统必须支持视频、音频等连续媒体，这些媒体对系统的要求不同于文本之类的数据媒体。

4) 缺乏适合连续媒体的文件系统。连续的媒体文件因为受到严格的时间限制，对文件的布局、缓冲器的分配、许可控制及客户调度具有很高的要求。

第七章 多媒体操作系统

2. 计算机操作系统

2.1 操作系统的基本概念

2.1.1 操作系统定义

操作系统是计算机系统软件资源和硬件资源的管理中心，它以尽量合理有效的方法组织多个用户共享计算机的各种资源。

这个定义应用到多媒体系统中也是合适的，对多媒体系统来说，多媒体操作系统将负责管理系统中的各类资源，包括网络资源，提高资源的可用性和利用率，为用户提供与系统交互的人机界面。

第七章 多媒体操作系统

2. 计算机操作系统

2.1 操作系统的基本概念

2.1.2 操作系统分类

计算机的操作系统分为：单用户系统、分时系统、实时系统等多种形式。

分时系统：为了使一台机器可以同时为多个用户和多个任务服务。为了做到这一点，系统将CPU时间分成多个时间片，提供给多个用户或进程使用。这样，由于系统可以为多个用户或进程使用，提高系统的吞吐量，系统的利用率也大大提高。

分时系统以提高系统的整体利用率为目标。对某些特殊的应用，在时间响应上就难以满足要求。为了解决这个问题，产生了实时操作系统。

第七章 多媒体操作系统

2. 计算机操作系统

2.1 操作系统的基本概念

2.1.2 操作系统分类

实时操作系统的目标：对实时的应用提供及时的响应，同样可以支持多个用户和多个任务，但不是把吞吐率放在第一位。而是把在一定时间范围内对事件的响应放在第一位。

对多媒体系统来说，由于在系统中大量使用基于时间的连续媒体，所以对时间的要求就非常严格。很显然，使用分时系统难以满足时间的要求，实时系统为多媒体操作系统提供了对基于时间媒体进行管理的一条途径。

7

第七章 多媒体操作系统

2. 计算机操作系统

2.1 操作系统的基本概念

2.1.3 实时系统的有关概念

实时进程：是指在给定时间段内给出处理结果的进程。

实时系统：实时系统的一个永久任务是从环境中接收信息（信息可能是自发的或周期性发生的），在给定的时间限制内把处理结果返回给环境。

一个重要特征：系统工作的正确性不仅依赖于计算结果的逻辑正确性，而且还依赖于得出结果的时间。

也就是说，实时系统中的任务是具有时间约束的。

8

第七章 多媒体操作系统

2. 计算机操作系统

2.1 操作系统的基本概念

2.1.3 实时系统的有关概念

任务的时间约束多种多样。最普遍的时间约束是周期性（periodic）和非周期性（aperiodic）。

周期性：任务必须在每个周期内完成一次，可以在周期开始时启动，也可以在周期开始后启动，但都必须在周期结束前完成。

非周期性任务没有规定的时间要求，但必须有一个启动时间或结束时间，或二者兼而有之。

9

第七章 多媒体操作系统

2. 计算机操作系统

2.1 操作系统的基本概念

2.1.3 实时系统的有关概念

最后期限(deadline)：最后期限是指表现处理结果的最迟可以接受时间。用它来判断处理动作的正确与否。

其他的时间约束还包括：

任务执行时间：从任务启动到结束花费的时间。

任务余量：完成任务后离时限的时间余量。

紧时间约束：任务余量少；松时间约束：任务余量大。

10

第七章 多媒体操作系统

2. 计算机操作系统

2.1 操作系统的基本概念

2.1.3 实时系统的有关概念

时限粒度：任务从启动到时限之间的时间。

强实时：时限前必须完成，否则结果失去可用性。

弱实时：超过时限仍然可用，但可用性下降。

关键任务：重要性高，完不成会引起灾难性后果的任务。

11

第七章 多媒体操作系统

2. 计算机操作系统

2.1 操作系统的基本概念

2.1.4 实时系统的特征

实时系统必须满足逻辑正确和时间约束两个条件。总的看来：在时间约束内完成进程。实时系统应该具有如下的特征：

④ 能对限时事件进行预期的快速响应，并且有精确的时序信息；

④ 高度的可调度性；

④ 短暂超负荷情况下的稳定性。如果出现短暂的超负荷，系统应不会失败，并能保证一些重要的任务完成；

12

第七章 多媒体操作系统

2. 计算机操作系统

2.1 操作系统的基本概念

2.1.5 实时系统的性能

实时系统的性能的好坏主要从以下几个方面进行衡量：

📖 响应时间：包括中断响应时间、关联转换时间和任务等待时间等

📖 吞吐率

📖 可靠性

13

第七章 多媒体操作系统

2. 计算机操作系统

2.1 操作系统的基本概念

2.1.6 实时系统中的调度

调度的内容包括：CPU调度，资源分配，I/O调度，实时通信调度、分布式调度及关键任务调度等。

几种不同的调度方法：

基于优先级的抢占调度算法：任务具有静态或动态的优先级，当一个低优先级的任务正在运行时，如果高优先级任务到来，系统将抢占正在运行的任务的资源，让新到达的任务开始运行。

14

第七章 多媒体操作系统

2. 计算机操作系统

2.1 操作系统的基本概念

2.1.6 实时系统中的调度

不精确调度算法（Imprecise Computing, IC）：把任务分成两个部分，关键部分和可选部分。关键部分必须完成，但得到的结果只保证可用，不精确；可选部分的作用是使结果更加精确。系统保证关键部分满足时限要求，在允许的情况下尽可能地运行可选部分，在系统过载时不运行可选部分。

最早时限优先（Earliest Deadline First, EDF）：动态优先级调度算法，时限最早的任务优先级最高。由于没有任务的优先级会有变化，系统要经常计算每个用户的优先级。

15

第七章 多媒体操作系统

2. 计算机操作系统

2.1 操作系统的基本概念

2.1.6 实时系统中的调度

最关键任务优先（Most Critical First, MCF）算法：按照任务的重要性分配优先级。

关键性和时限优先（Criticalness and Deadline First, CDF）算法，同时考虑任务的重要性和时限。

最小余量优先（Minimum Laxity First, MLF）算法，按照任务的余量分配优先级。

16

第七章 多媒体操作系统

2. 计算机操作系统

2.2 多媒体的引入与操作系统的限制

2.2.1 实时服务与时间要求

多媒体系统中存在大量基于时间的连续媒体：要求操作系统具有对时间的复杂控制和维持特定QoS的机制，提供在系统短暂过载时进行保护和管理模式。

17

第七章 多媒体操作系统

2. 计算机操作系统

2.2 多媒体的引入与操作系统的限制

2.2.2 中断等待

在一些多媒体应用中，会产生比较频繁的中断请求，产生中断延时，对连续媒体来说，延时会影响它的使用，需要采取一些方法来改进。

18

第七章 多媒体操作系统

- 2. 计算机操作系统
- 2.2 多媒体的引入与操作系统的限制
- 2.2.3 优先级倒置管理

当一个实时程序在系统中与一个非实时程序共享一个资源时，可能会出现实时程序不得不等待非实时程序运行完成的情况。

```
Server(){  
    While(1){  
        Receive_mag();  
        Do_service();  
        Reply();  
    }  
}
```

引入优先级继承（Priority Inheritance）和优先级交出（Priority Hand_off）机制。

19

第七章 多媒体操作系统

- 2. 计算机操作系统
- 2.2 多媒体的引入与操作系统的限制
- 2.2.3 优先级倒置管理

引入优先级继承（Priority Inheritance）和优先级交出（Priority Hand_off）机制。

当高优先级的线程被封锁时，就把它的优先级传给对它进行封锁而优先级更低的线程；
若低优先级的线程停止封锁高优先级的线程时，低优先级的线程就恢复到它原来的优先级；

20

第七章 多媒体操作系统

- 2. 计算机操作系统
- 2.2 多媒体的引入与操作系统的限制
- 2.2.4 周期性活动

连续媒体为了传输和表现需要周期性的服务活动。

最简单的操纵周期性数据流的方法：使用循环结构。

21

第七章 多媒体操作系统

- 2. 计算机操作系统
- 2.2 多媒体的引入与操作系统的限制
- 2.2.5 时限和恢复管理

与传统的实时系统不同，很多连续媒体应用程序具有内在的“软”时限，也就是“弱实时”。

例如，在进行视频会议时，即使大多数的视频图形都没有能够按时处理完毕，我们还是可以把会议继续进行下去。

因此，当程序由于过载、硬件或软件错误而错过时限时，用户程序应能够决定相应的应对措施。

22

第七章 多媒体操作系统

- 2. 计算机操作系统
- 2.2 多媒体的引入与操作系统的限制
- 2.2.6 QoS管理和许可控制

两种管理模式：静态的和动态的。

静态的，会话开始之前，由用户指定一个QoS的级别。并在整个会话存在期间都维持这个级别。

23

第七章 多媒体操作系统

- 2. 计算机操作系统
- 2.2 多媒体的引入与操作系统的限制
- 2.2.6 QoS管理和许可控制

动态控制模式则允许用户或系统在会话存在期间改变QoS的初始值。可以用两种方法来改变，也就是分别由系统和用户来发起这种改变。

系统根据系统可用的资源来进行调整。

用户根据任务的需要而降低或改善QoS的级别。

24

第七章 多媒体操作系统

3. 操作系统对连续媒体的支持

对连续媒体应用的支持可以分为三个方面：

- 📖 体系结构的支持；
- 📖 资源管理的支持；
- 📖 程序设计的支持。

25

第七章 多媒体操作系统

3. 操作系统对连续媒体的支持

3.1 体系结构的支持

操作系统的基本结构一般是内核体系。进程管理、文件系统服务、网络管理、虚拟内存管理、设备管理等服务都建立在内核之中。最终导致了一个庞大的、笨重的、不灵活的内核。很难将它进行增强。

体系结构改变：微内核。

微内核的基本思想：尽量减少内核中的东西，把它们搬到高层，使得内核只留下很少的一部分。

26

第七章 多媒体操作系统

3. 操作系统对连续媒体的支持

3.1 体系结构的支持

微内核中一般包括线程和任务管理、IPC（Inter Process Communication，进程间通讯）管理、存储对象管理、虚拟内存服务、I/O和中断服务等。

将其他功能如文件管理、进程管理等移出内核，作为操作系统的服务器在用户空间运行。传统的设备驱动，也可作为用户级的任务在内核外实现。

内核中的服务通过微内核接口函数提供给用户级的进程或任务，或由用户要求某种服务，由内核中服务器完成并返回应答消息。

27

第七章 多媒体操作系统

3. 操作系统对连续媒体的支持

3.1 体系结构的支持

微内核的优势：

提供了一种简单的、高度模块化的可扩充体系结构。在这种体系结构下，对连续媒体的特殊功能可以根据需要进行添加和改造。

微内核结构易于移植，而且效率高。

28

第七章 多媒体操作系统

3. 操作系统对连续媒体的支持

3.2 资源管理的支持

采取新的资源管理模式来提供必需的系统资源。这样，用户就能够对所请求得到的QoS级别进行维护，从而在再现媒体时避免出现不可预见的延迟和抖动。

对连续媒体应用进行资源管理的技术主要有如下一些：

- 📖 基于QoS的资源控制
- 📖 实时调度
- 📖 基于QoS的内存管理
- 📖 基于时间的I/O管理

29

第七章 多媒体操作系统

3. 操作系统对连续媒体的支持

3.2 资源管理的支持

基于QoS的资源控制：

本地和分布式多媒体应用程序的传输和处理需求可以用如下一组QoS参数来说明：

- 1) 吞吐量：满足应用程序所需连接的数据位率；
- 2) 局部和全局的延迟，包括资源延迟和端到端延迟。“资源延迟”是指该资源完成某一任务所需要的最大时间跨度。“端一端延迟”是指一个数据单元从源传输到目标的总延迟。

30

第七章 多媒体操作系统

3. 操作系统对连续媒体的支持

3.2 资源管理的支持

基于QoS的资源控制

本地和分布式多媒体应用程序的传输和处理需求可以用如下一组QoS参数来说明：

3) 抖动：指的是数据到达目的地所需时间的最大偏差。

4) 可靠性：多媒体任务传输和处理过程中使用的错误检测和校正机制。在视频音频等时间相关媒体中，重传机制往往不适合。

31

第七章 多媒体操作系统

3. 操作系统对连续媒体的支持

3.2 资源管理的支持

基于QoS的资源控制

资源分配机制：资源分配可以采用悲观或乐观的方法。

悲观方法：按最坏的情况预留，以避免资源冲突。即预定一个任务所需要的最长处理时间和最高速率的资源带宽。导致资源利用率较低。

乐观的方法：资源仅根据平均工作负荷预留。QoS参数被尽可能的满足。资源的利用率很高。但在过载时会引起失败。为检测过载情况并对其进行相应的处理，可以实现一个监控进程，根据进程的重要性作抢占式中断处理。

32

第七章 多媒体操作系统

3. 操作系统对连续媒体的支持

3.2 资源管理的支持

实时调度

对具有时间约束的连续媒体应用程序，实时调度能够提供更好的处理器调度支持。

对基于QoS的模式来说，应该把传统的实时调度策略与许可控制和强制机制结合起来。

实时调度应该能够对连续媒体的最大和最小执行速率实行控制和强制，以便在会话过程中维护所请求的QoS级。

33

第七章 多媒体操作系统

3. 操作系统对连续媒体的支持

3.2 资源管理的支持

基于QoS的内存管理

在基于QoS的方法中，系统应该能够以时间为基础对连续媒体对象实行预留。

但若允许用户能对媒体流进行随机的位置指定，则对连续媒体寻找正确的存储页面将十分困难。

34

第七章 多媒体操作系统

3. 操作系统对连续媒体的支持

3.2 资源管理的支持

基于时间的IO管理

实时系统的设计者关心的主要是如何加快进入I/O事件的处理速度。但对多媒体系统来说，时态的正确性更加重要。因为系统必须对多重数据流的同步进行支持。

为了做到这一点，一般是给数据流加上时间标志。

35

第七章 多媒体操作系统

3. 操作系统对连续媒体的支持

3.3 程序设计的支持

实时线程：实时线程具有时间约束。

时限管理：对于时态正确性错误，必须提供一种和其他异常一样的捕捉计时错误的接口，以使得户能够以应用指定的方式进行处置。

实时同步：为了支持实时同步，系统至少应该提供快速事件标识机制和实时互斥机制。为了避免优先权倒置而采用的机制。

36

第七章 多媒体操作系统

3. 操作系统对连续媒体的支持

3.4 实时的进程间通信

在通信端口中提供下面的属性：

消息排队：指明消息排队的次序。可以使用FIFO及基于优先权的排序策略。

优先权交出：在消息传递时操纵接收者的优先权。如果设置了这个属性，接收者的优先权就将由发送者传递，或者根据所选定的策略给定优先权。

优先权继承：执行优先权继承协议。

消息分发：当两个或多个接收者在运行时，将选择合适的接收者线程。

37

第七章 多媒体操作系统

4. 支持连续媒体的文件系统

常规的文件系统，不能满足一些多媒体应用的需要，不能保证客户在正常缓冲条件下，以某种可达到传输极限的方式来访问连续媒体。

磁性存储设备、光存储设备具有大量的寻道和旋转等潜在因素，如果连续媒体流不能被及时地缓冲并且与其它请求相协调，就会导致上层的应用需要的数据不能够得到保障，产生暂停、断续等方面的问题。对文件系统来说，应该考虑解决以下一些问题：

38

第七章 多媒体操作系统

4. 支持连续媒体的文件系统

- 1) 连续媒体文件的布局：即连续媒体文件数据块如何在磁盘上进行安排。
- 2) 对缓冲器的要求：即每个媒体流必须分配的最小的缓冲是多大？反之，如果缓冲器是固定的，能够支持多少并发的流？
- 3) 许可控制：在当前服务的客户集合中再增加一个客户，何时是安全的？(需要具有QoS保证时才能进行此项判断。)
- 4) 服务中客户请求的调度：对给定的一组客户请求，在一轮服务中应如何排序，才可以满足实时性的要求并且使缓冲器资源最小；

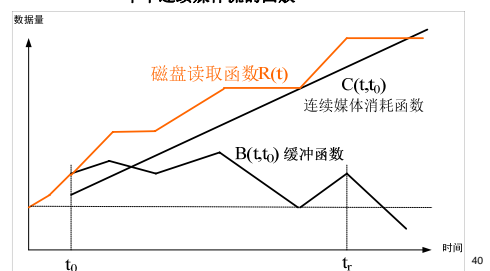
39

第七章 多媒体操作系统

4. 支持连续媒体的文件系统

4.1 单流情况

单个连续媒体流的重放



40

第七章 多媒体操作系统

4. 支持连续媒体的文件系统

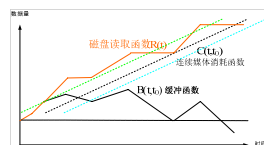
4.1 单流情况

最早起始时间、最少缓冲器

最小缓冲器与最早回放时间 t_0 相关。可以用图示法找到最小回放起始时间：改变 t_0 ，使得B最小且非负。

- 1) 首先考虑 $B(t, 0)$ ，如果此函数在 $0 < t \leq t_r$ 非负，则 t_0 最小值是0
- 2) 否则，在 $0 < t \leq t_r$ 之间假设于 t_{min} 得到 $B(t, 0)$ 的最小值 $-m$ ，则 $R(t)$ 与 $B(t, 0) + m$ 的交点就是 t_0 的最小值

41



第七章 多媒体操作系统

4. 支持连续媒体的文件系统

4.2 多流情况

多流情况要求考虑以下问题：

- 各通道有不同的消耗速率；
- 某些通道的回放可以暂停，而其他回放过程仍在继续；
- 每个周期中各个通道的读取总量不固定；

42

第七章 多媒体操作系统

4. 支持连续媒体的文件系统

4.2 多流情况

要求读操作读出的数据能够跟上数据的实时使用：

R_c 数据的实时使用速率， Δ_{max} 给定通道两个读操作之间最大的时间间隔

在读操作之间必须的缓冲存储量为 $R_c \Delta_{max}$

问题的关键：寻找最大的间隔时间

43

第七章 多媒体操作系统

4. 支持连续媒体的文件系统

4.3 介质调度

4.3.1 传统磁盘调度算法

传统文件系统的两个目标是：

📖 向用户提供访问文件的友好接口

📖 有效利用存储介质

44

第七章 多媒体操作系统

4. 支持连续媒体的文件系统

4.3 介质调度

4.3.1 传统磁盘调度算法

文件结构：

主要有顺序存储和非顺序存储两种结构。

在顺序结构中，每个文件组织成一个字节（或记录）的简单序列。彼此之间用一个预定义的“文件结束”（可以是某种特殊序列的字符）来分隔。文件描述符一般放在文件的开头。

在磁带类型的介质上，顺序存储是组织文件系统的唯一可行的方法。

45

第七章 多媒体操作系统

4. 支持连续媒体的文件系统

4.3 介质调度

4.3.1 传统磁盘调度算法

文件结构：

顺序存储的优点：连续存取时有较好的效率。（适合于大数据流的多媒体数据的存储）

顺序存储的缺点：但在某些应用中，文件建立、删除、改变大小等操作频繁，就会使效率大大的下降。

46

第七章 多媒体操作系统

4. 支持连续媒体的文件系统

4.3 介质调度

4.3.1 传统磁盘调度算法

文件结构：

在非顺序存储中，数据的放置主要有两种方法：

（1） 一种方法是用链接块，代表连续逻辑位置的物理块以指针链接。文件描述：文件所占的块数，指向第一块的指针。这种方法的最大缺点是在文件随机访问时，代价较高（必须把访问点之前的所有数据都读出）。

（2） 另一种方法是信息存储到映射表中。每个文件与一个表相连。表中除了文件块数以外，还有文件所有者，文件大小，建立时间，最后访问时间等信息。

47

第七章 多媒体操作系统

4. 支持连续媒体的文件系统

4.3 介质调度

4.3.1 传统磁盘调度算法

目录结构：

通常是树型目录

48

第七章 多媒体操作系统

4. 支持连续媒体的文件系统

4.3 介质调度

4.3.1 传统磁盘调度算法

在随机访问存储设备中，每个文件操作都需要移动读写头，这个操作称为“寻道”。而读写磁盘的实际时间由以下因素决定：

寻道时间（读写头移动所需的时间）

等待时间或称旋转延迟（在正确的块或扇区旋转到读写头之前传输无法进行的时间）

将数据从磁盘读出到主存所需的实际数据传输时间

49

第七章 多媒体操作系统

4. 支持连续媒体的文件系统

4.3 介质调度

4.3.1 传统磁盘调度算法

磁盘管理：磁盘的管理和调度与文件访问的效率密切相关。

在传统文件系统中，降低磁盘访问时间的技术是块高速缓存，预期将来的读或写操作会再次访问的数据块被保存在内存中以缩短访问时间，从而提高性能。

另一种提高性能的方法是减少磁盘臂运动（也就是读写磁头的径向运动），可能被顺序访问的数据块在同一柱面上放置在一起。例如，磁盘上映射表的放置，如果把它放置在磁盘的头上，则它们和块之间的距离平均为柱面的一半。一种改进的方法，将它们放在磁盘中间，则平均寻道时间大约降低一半。同样，相邻块应放在同一柱面，映射表和被索引块存储在同一个柱面也能提高性能。

50

第七章 多媒体操作系统

4. 支持连续媒体的文件系统

4.3 介质调度

4.3.1 传统磁盘调度算法

调度算法（1）先到先服务（FCFS）,P208

在该算法中，磁盘驱动器按达到顺序每一次接受一个请求服务。虽然实现比较简单，且比较公平，但是对读写头的移动来说不够优化。

这种算法没有考虑到队列中其他请求的位置，因而导致了很高的平均寻道时间。

51

第七章 多媒体操作系统

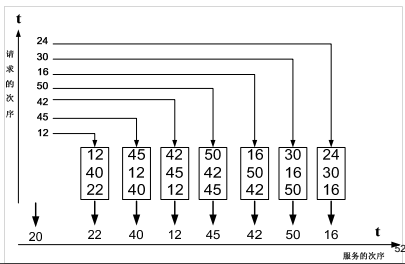
4. 支持连续媒体的文件系统

4.3 介质调度

4.3.1 传统磁盘调度算法

调度算法（1）

先到先服务（FCFS）



第七章 多媒体操作系统

4. 支持连续媒体的文件系统

4.3 介质调度

4.3.1 传统磁盘调度算法

调度算法（2）最短寻道时间优先（SSTF）P209, 图7.5

每一个时刻，当一个数据访问被请求，SSTF先在所有请求中选择距当前读写头位置具有最小寻道时间的请求，即读写头移动到请求队列中最近的磁道。

该算法用来使寻道时间最小化，在这一点上它是最优的。但有可能一些请求迟迟得不到满足。

53

第七章 多媒体操作系统

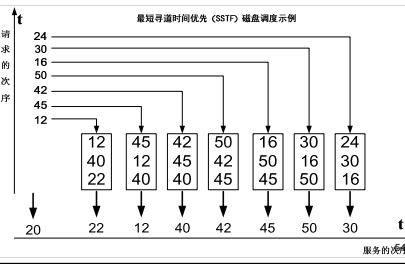
4. 支持连续媒体的文件系统

4.3 介质调度

4.3.1 传统磁盘调度算法

调度算法（2）

（SSTF）



第七章 多媒体操作系统

4. 支持连续媒体的文件系统

4.3 介质调度

4.3.1 传统磁盘调度算法

调度算法（3）扫描（SCAN）P209,图7.6

和SSTF一样，SCAN算法也对请求进行排序以降低寻道时间。但与SSTF不同的是，它考虑当前磁头移动方向。它先服务一个方向上的所有请求，从一端向另一端扫描过去，遇到请求便服务，没有请求则继续前进。运动到磁盘的边缘后在反向运动，继续扫描。

SCAN算法对寻道时间进行了优化，但是还是存在不均匀的现象。当读写头改变方向时，它首先服务最近被服务过的磁道，假定数据请求是一致分布，则这时请求最密的地方是磁盘的另一端。增加判断以提高效率：在同一个方向上已没有请求时，不再移动到边缘，而是改变方向向回移动。

55

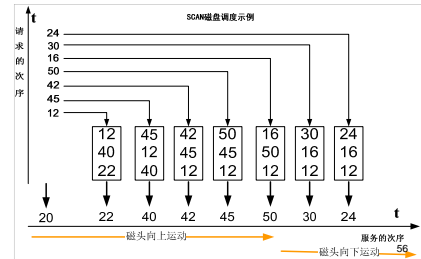
第七章 多媒体操作系统

4. 支持连续媒体的文件系统

4.3 介质调度

4.3.1 传统磁盘调度算法

调度算法（3）
扫描（SCAN）



第七章 多媒体操作系统

4. 支持连续媒体的文件系统

4.3 介质调度

4.3.1 传统磁盘调度算法

调度算法（4）C-SCAN算法，P209,图7.7

C-SCAN算法也是按照一个方向移动读写头，但是它提供了较为一致的等待时间的公平服务。与SCAN不同的是，它有一个从一边到另一边的读写头空移动。C-SCAN的性能比SCAN略低。

57

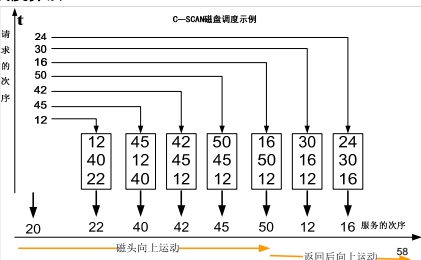
第七章 多媒体操作系统

4. 支持连续媒体的文件系统

4.3 介质调度

4.3.1 传统磁盘调度算法

调度算法（4）
C-SCAN算法



第七章 多媒体操作系统

4. 支持连续媒体的文件系统

4.3 介质调度

4.3.1 传统磁盘调度算法

总结：

传统的文件系统不是为多媒体系统设计的，其中没有考虑对于检索和存储音频及视频时很重要的因素：如实时性的要求等。

为满足多媒体应用要求，文件的结构和组织以及从磁盘上检索数据都必须采取新的策略。

59

第七章 多媒体操作系统

4. 支持连续媒体的文件系统

4.3 介质调度

4.3.2 连续媒体调度

连续媒体数据在以下几个方面与离散数据不同：

实时特征：检索、显示具有时间依赖性。数据必须在一个明确给出的最后期限前显示。因此，存储和检索这类数据的算法必须考虑时间约束，必须提供附加的缓冲器以平滑数据流。

文件大小：与文本和图形相比，视频音频需要很大的存储空间。因为文件系统必须存储从小的非结构性的文件到大的高结构性的数据单元（如视频及相关音频）等各种信息，所以它必须以高效利用有限存储的方式组织磁盘上的数据。

60

第七章 多媒体操作系统

4. 支持连续媒体的文件系统

4.3 介质调度

4.3.2 连续媒体调度

连续媒体数据在以下几个方面与离散数据不同：

多重数据流：多媒体系统必须同时支持不同的媒体，它不但必须保证它充分地共享资源，还必须考虑来自源的数据流之间的紧密相关性。例如，电影的检索就需要视频、音频、字幕的同步操作。

61

第七章 多媒体操作系统

4. 支持连续媒体的文件系统

4.3 介质调度

4.3.2 连续媒体调度

文件系统中用不同的方法来支持连续媒体。

1、 磁盘上的文件的组织和传统系统相同，而必要的实时支持则通过特别的磁盘调度算法来提供，并且提供充足的缓冲以避免抖动。

2、 磁盘上视频、音频文件的组织针对其在多媒体系统中的应用进行优化。多重数据流的调度可以采用实时系统中的一些调度算法。

62

第七章 多媒体操作系统

4. 支持连续媒体的文件系统

4.3 介质调度

4.3.2 连续媒体调度

调度算法——最早时限优先EDF (Earliest Deadline First) P210, 图7.8

如果请求的服务时间可以预知，最早时限优先法是可行的。在该算法中，系统调度器在每一个就绪状态时进行调度，将选择一个已经就绪但仍未处理完毕但时限最早的任务，并将其请求的资源赋给它。

在新的任务到达时，EDF就必须抢占目前运行的任务；立即计算最早时限并重新排序，新的任务将在这次排序中被考虑。如果新的任务比打断的任务时限还要早，新的任务立即执行，被打断的任务将根据EDF算法在后面重新调度。

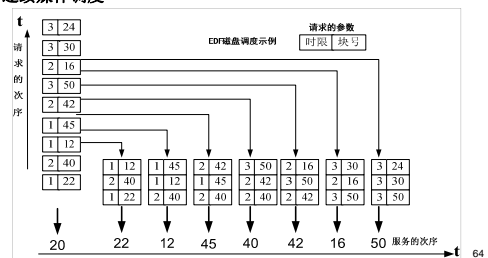
63

第七章 多媒体操作系统

4. 支持连续媒体的文件系统

4.3 介质调度

4.3.2 连续媒体调度



第七章 多媒体操作系统

4. 支持连续媒体的文件系统

4.4 调度算法——SCAN-EDF算法

两个算法相结合。P 212, 图7.9。对请求的服务按照其EDF顺序，如果几个请求的时限相同，那么就根据它们对应的介质位置来选择，即第一个符合扫描方向的最先服务。

如何提高这种算法的效率：因为只是对具有相同时限的请求进行优化，其效率就取决于有多少请求具有相同或者近似的时限。为提高时限相同的概率，可以采用所有请求的时限都是周期P的倍数。这样请求就可以按此分组并按扫描算法被服务。

65

第七章 多媒体操作系统

4. 支持连续媒体的文件系统

4.4 调度算法——SCAN-EDF算法

对于具有不同的数据传输速率要求的请求，除了SCAN-EDF外，还可以采用一种周期填充策略使所有的请求都具有相同的时限。所有的请求以循环方式被服务，每次循环中，各请求都获得与其所需数码率成正比的服务时间。

66

第七章 多媒体操作系统

4. 支持连续媒体的文件系统

4.4 调度算法——SCAN-EDF算法

SCAN-EDF的实现:

如果任务*i*的时限为 D_i , N_i 是其磁道的位置, 可以将时限改为 $D_i + f(N_i)$ 。

其中函数*f*将*i*的磁道号变成时限*i*的一个小扰动。这种小扰动应该足够小, 使得对于所有 $D_i < D_j$, 有 $D_i + f(N_i) < D_j$ 。 $f(N_i)$ 可采用以下定义:

$$f(N_i) = N_i / N_{\max}$$

其中, N_{\max} 是盘上的最大磁道号。

67

第七章 多媒体操作系统

4. 支持连续媒体的文件系统

4.4 调度算法——SCAN-EDF算法

对更细致的算法:

(1)如果读写头向 N_{\max} 移动, 则:

对于所有位于实际位置 N 和 N_{\max} 之间的块 N_i ,

$$f(N_i) = (N_i - N) / N_{\max} \quad (\text{对于所有 } N_i \geq N)$$

对于所有位于实际位置 N 和第一块(0号)之间的块 N_i ,

$$f(N_i) = (N_{\max} - N_i) / N_{\max} \quad (\text{对于所有 } N_i < N)$$

68

第七章 多媒体操作系统

4. 支持连续媒体的文件系统

4.4 调度算法——SCAN-EDF算法

对更细致的算法:

(2)如果读写头向 N_0 移动, 则:

对于所有位于实际位置 N 和 N_{\max} 之间的块 N_i ,

$$f(N_i) = N_i / N_{\max} \quad (\text{对于所有 } N_i \geq N)$$

对于所有位于实际位置 N 和第一块(0号)之间的块 N_i ,

$$f(N_i) = (N - N_i) / N_{\max} \quad (\text{对于所有 } N_i \leq N)$$

69

第七章 多媒体操作系统

小结

本章主要讨论多媒体特别是连续媒体对操作系统带来的影响, 要求掌握支持连续媒体操作的方法。

70