操作系统

2018年1月9日 12:07

考试题型

填空题：（每空1分，共10分）
单选题：（每题2分，共40分）
简答题：（4道，共 18 分）
算法设计填空题：（1道，8分）
综合题：（每题8分，共24分）

第一章

1.用户态、核心态的概念

核心态：操作系统程序，特权。

用户态：执行用户应用程序。

2.操作系统的定义、主要功能；操作系统的用户观点和系统观点

操作系统是控制和管理计算机系统内各种硬件和软件资源，有效地组织多道程序运行的系统软件（或程序集合），是用户与计算机之间的接口。

（1）存储管理功能

内存分配（策略）、地址映射（转换）、内存保护（不干扰不越界）、内存扩充（逻辑扩充：虚拟存储器与置换）。

（2）处理机管理功能

作业和进程调度（创建进程，算法）、进程控制（创建、撤销、唤醒、挂起等）和进程通信（依赖和制约：同步和互斥）。

（3）设备管理功能

缓冲区管理（速度匹配），设备分配（请求、策略），设备驱动（通道）和设备无关性（逻辑设备名、物理设备名）。

（4）文件管理功能

文件存储空间的管理（分配与回收），文件操作的一般管理（创建、删除、打开、关闭），目录管理，文件的读写管理和存取控制（防止越权与破坏）。

（5）用户接口

命令界面：

这是指由OS提供了一组联机命令(语言)， 用户可通过键盘输入有关命令，来

直接操纵计算机系统。

程序界面：

OS提供了一组系统调用，用户可在自己的应用程序中通过相应的系统调用，来操纵计算机。

图形界面：

用户通过屏幕上的窗口和图标来操纵计算机系统和运行自己的程序。

3.操作系统提供给用户的三种接口

命令行、系统调用、图形界面

4.批处理系统、分时系统、实时系统的概念及主要特征

（1）单道批处理系统：

程序一个一个不被打断的顺序执行。

主要特征如下：

自动性、顺序性、单道性。

（2）多道批处理系统：

在该系统中，用户所提交的作业都先存放在外存上并排成一个队列，称为“后备队列”；然后，由作业调度程序按一定的算法从后备队列中选择若干个作业调入内存，使它们共享CPU和系统中的各种资源。

主要特征如下：

多道性、无序性、 调度性、共享性。

多道程序设计技术可带来以下好处：

①提高CPU的利用率。

②可提高内存和I/O设备利用率。

③增加系统吞吐量。

多道程序设计技术明显缺点：

①用户作业的等待时间长

②没有交互能力

（3）分时系统指的是：

在这个操作系统下有多个用户终端，分时共享主机资源。

所谓分时，就是对时间的共享，主要是指若干并发程序对CPU时间的共享，分享的时间单位叫时间片。

所谓并行是指在同一时刻有两个或两个以上的活动发生。

分时系统的特征：

同时性：多用户同时上机使用，多终端。

交互性：人机交互。

独立性：各用户操作互不干扰。

及时性：很短时间相应。

分时系统的优点：

为用户提供友好接口；

一个系统可带多台终端；

便于资源共享和信息交换。

（4）实时系统的引入：

实时系统(Real-Time System)是指系统能及时(或即时)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。

对时间有严格的限制和要求：实时控制；实时信息处理。

实时系统现在有三种典型应用形式：

过程控制系统

工业自动控制，如温度压力控制、导弹发射等。

信息查询系统

配有大型数据库供查询，如仓储、医护信息系统等。

事务处理系统

数据更新频繁，如飞机票预定、银行财务往来。

5.区分并行与并发

并发是指两个或多个活动在同一给定的时间间隔中进行。宏观概念。如CPU共享。

并行性是指两个或多个事件在同一时刻发生；而并发性是指两个或多个事件在同一时间间隔内发生。

6.操作系统的4个基本特征

（1）并发

并发是指两个或多个活动在同一给定的时间间隔中进行。宏观概念。如CPU共享。

例如：

在多道程序环境下，并发性是指在一段时间内，宏观上有多个程序在同时运行，每一时刻却仅能有一道程序执行，故微观上这些程序只能是分时地交替执行。

倘若在计算机系统中有多个处理机，则这些可以并发执行的程序便可被分配到多个处理机上，实现并行执行，即利用每个处理机来处理一个可并发执行的程序，这样，多个程序便可同时执行。

（2）共享

共享是指计算机系统中的资源被多个进程所共用。如CPU、硬盘、内存、数据等。

共享分如下两种：

互斥地共享：某进程申请资源、若空闲、分配、运行，下一个进程只能等待，直到前一进程释放资源。

宏观上同时访问、微观上并发执行的共享：如硬盘上文件的访问。

例如：

系统中的某些资源，如打印机、磁带机，虽然它们可以提供给多个进程(线程)使用，但为使所打印或记录的结果不致造成混淆，应规定在一段时间内只允许一个进程(线程)访问该资源。

（3）不确定性

不确定性是指系统中各种事件发生顺序的不可预测性。

只有进程在获得所需的资源后方能执行，所以进程的执行通常都不是“一气呵成”，而是以“停停走走”的方式运行。

第二章

1.进程的概念、基本特征

进程是进程实体的运行过程，是系统进行资源分配和调度的一个独立单位

进程的基本特征：

（1）动态性

（2）并发性

（3）调度性：处理机调度的单位。

2.进程的三个基本状态及其转换关系

进程的基本状态

运行状态（Running）

已分配到CPU正在处理机执行的状态。

单CPU系统中，某时刻运行态进程最多只有一个。

就绪状态（Ready）

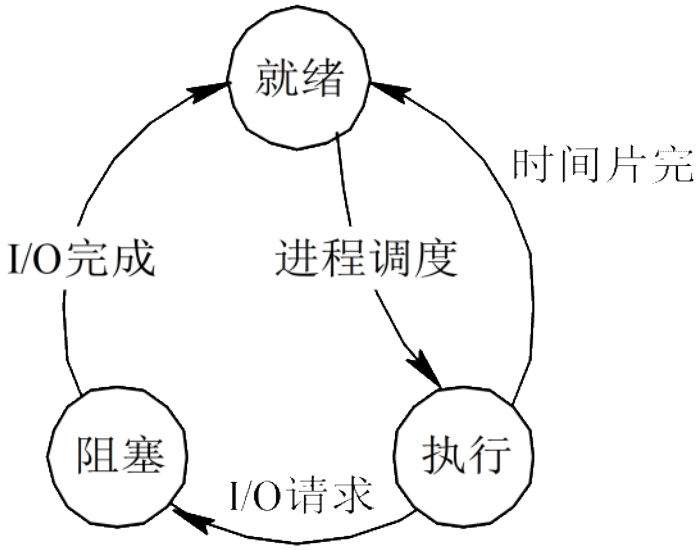
进程具备运行所有资源，只是等待分配CPU。

就绪队列中可有多个就绪态的进程。

阻塞状态（Blocked）

进程因等待某种事件发生（例如等待某个输入、输出操作完成，等待其他进程发来的信号等）而暂时不能运行的状态。（封锁或等待状态）。 阻塞队列中可有多个阻塞态的进程。

三种基本状态及其转换



3.进程的组成，进程控制块的作用

进程实体：

程序和数据。静态。

进程是动态的

需要一个数据结构：描述进程当前状态、本身特性、对资源的占用和调度等动态的信息。----进程控制块PCB。

进程控制块的作用

（1）每个进程有惟一的进程控制块。进程控制块是进程存在的唯一标识。

（2）进程控制块的作用是使一个在多道程序环境下不能独立运行的程序(含数据)，成为一个能独立运行的基本单位，一个能与其它进程并发执行的进程。

（3）或者说，OS是根据PCB来对并发执行的进程进行控制和管理的。

（4）P34段中例如。

4.进程与线程的关系，各自的分工

（1）一个进程可以有多个线程，但至少要有一个线程；而一个线程只能在一个进程的地址空间内活动。

（2）资源分配给进程，同一进程的所有线程共享该进程的所有资源。

（3）处理机分配给线程，即真正在处理机上运行的是线程。

（4）线程在执行过程中需要协作同步。不同进程的线程间要利用消息通信的办法实现同步。

5.创建进程的过程

创建新进程时要执行创建进程的系统调用（如UNIX/Linux系统中的fork），其主要操作过程有如下四步：

（1）申请一个空闲的PCB，

（2）为新进程分配资源，

（3）将新进程的PCB初始化，

（4）将新进程加到就绪队列中。

6.结构型信号量值的含义，P、V操作的含义

结构型信号量一般是由两个成员组成的数据结构。其中一个成员是整型变量，表示该信号量的值；另一个是指向PCB的指针。

```
typedef struct{
        int value;
        struct PCB *list;
    } semaphore;
```

P操作的定义如下：

```
  void P(semaphore S)
 {
      S.value--;
      if(S.value<0)
      {
         把这个进程加到S.list队列;
         block( )；
       }
 }
```

V操作的定义如下：

```
    void V(semaphore S)
   {
            S.value++；
            if(S.value<=0)
           {
              从S.list队列中移走进程Q；
              wakeup(Q)；
            }
    }
```

在具体实现时应注意，P, V操作都应作为一个整体实施，不允许分割或相互穿插执行。

7.熟练掌握：会用PV操作写出进程间的同步算法。类型题练习：书后第17题。

生产者-消费者问题

读者-写者问题

哲学家进餐问题

打瞌睡的理发师问题

第三章

1.死锁的概念，饥饿的概念

死锁：

所谓死锁，是指在一个进程集合中的每个进程都在等待仅由该集合中的另一个进程才能引发的事件而无限期地僵持下去的局面。

计算机系统产生死锁的根本原因就是资源有限且操作不当.

饥饿:

在某些策略下，系统会出现这样一种情况：在可以预计的时间内，某个或某些进程永远得不到完成工作的机会，因为它们所需的资源总是被别的进程占有或抢占。这种状况称做“饥饿”或者“饿死”（Starvation）。

饥饿不同于死锁

2.解决死锁的办法

3.产生死锁的4个必要条件及破坏必要条件的策略

（1）互斥条件

（2）占有且等待条件

（3）不可抢占条件

（4）循环等待条件

只要有一个必要条件不满足，则死锁就可以排除。

（1）破坏互斥条件

（2）破坏占有且等待条件

一种办法是预分资源策略：静态分配.

另一种办法是“空手”申请资源策略

（3）破坏非抢占条件

隐式抢占：如果要申请其他资源的进程，当前所占有的全部资源可以被抢占。

另一种方法是抢占等待者的资源。

（4）破坏循环等待条件

①一种方法是实行资源有序分配策略:

所有进程对资源的申请严格按照序号递增和资源顺序而依次序进行，避免出现环路。

②另一种申请办法也很简单：

先弃大，再取小。

4.死锁避免的基本思想

（1）安全状态

在避免死锁的方法中，允许进程动态地申请资源，但系统在进行资源分配之

前，应先计算此次资源分配的安全性。若此次分配不会导致系统进入不安全状态，则将资源分配给进程； 否则，令进程等待。

所谓安全状态，是指系统能按某种进程顺序(P1, P2, …， Pn)(称〈P1, P2, …, Pn〉序列为安全序列)，来为每个进程Pi分配其所需资源，直至满足每个进程对资源的最大需求，使每个进程都可顺利地完成。

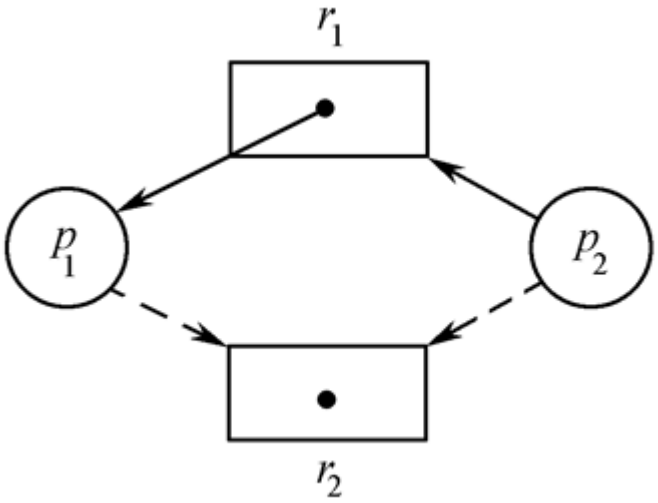
如果系统无法找到这样一个安全序列，则称系统处于不安全状态。

(2) 资源分配图算法

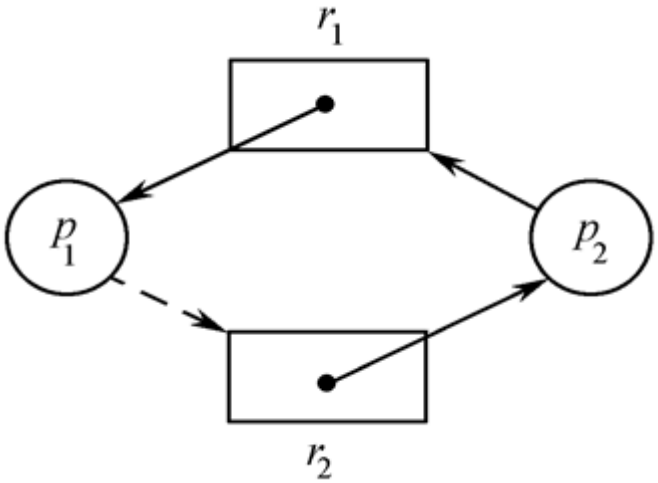
单体资源类

多体资源类

资源分配图示例



处于不安全状态的资源分配图



(3) 银行家算法

最著名的避免死锁的算法叫做“银行家算法”（Banker's Algorithm）。

银行家算法的设计思想是：当用户申请一组资源时，系统必须做出判断；如果把这些资源分出去，系统是否还处于安全状态。若是，就可以分出这些资源；否则，该申请暂不予满足。

5.熟练掌握：利用银行家算法避免死锁。类型题练习：书后第16题。

数据结构

令n表示系统中进程的数目，m表示资源分类数。

①Available是一个长度为m的向量，它表示每类资源可用的数量。Available[j]=k，表示rj类资源可用的数量是k。

②Max是一个n×m矩阵，它表示每个进程对资源的最大需求。Max[i, j]=k，表示进程pi至多可申请k个rj类资源单位。

③Allocation是一个n×m矩阵，它表示当前分给每个进程的资源数目。Allocation [i, j]=k，表示进程pi当前分到k个rj类资源。

④Need是一个n×m矩阵，它表示每个进程还缺少多少资源。Need [i, j]=k，表示进程pi尚需k个rj类资源才能完成其任务。

可以把矩阵Allocation和Need中的每一行当做一个向量，并分别写成Allocationi和Needi。Allocationi表示当前分给进程pi的资源。

（1）资源分配算法

令Requesti表示进程pi的申请向量。Requesti[j]= k，表示进程pi需要申请k个rj类资源。

当进程pi申请资源时，就执行下列动作：

① 若Requesti > Needi，表示出错，

②如果Requesti > Available，则pi等待。

③假设系统把申请的资源分给进程pi，则应对有关数据结构进行修改（试着分配）

Available：= Available – Requesti

Allocationi：= Allocationi + Requesti

Needi：= Needi – Requesti

④系统执行安全性算法，查看此时系统状态是否安全。如果是安全的，就实际分配资源，满足进程pi 的此次申请；否则，若新状态是不安全的，则pi等待，对所申请资源暂不予分配，并且把资源分配状态恢复成③之前的情况。

（2）安全性算法

①令Work和Finish分别表示长度为m和n的向量，最初，置Work：= Available，Finish[i]：=false，i=1, 2,…, n。

②搜寻满足下列条件的i值：

Finish[i]=false，且Needi≤Work。

若没有找到，则转向④。

③修改数据值：

Work：=Work+ Allocationi（pi释放所占的全部资源）

Finish[i]=true

转向②。

④若Finish[i]=true对所有i都成立（任一进程都可能是pi），则系统处于安全状态；否则，系统处于不安全状态。

第四章

1.处理机调度有几级，哪一级必不可少

处理机调度分为作业调度（高级调度）、进程挂起与对换（中级调度）和进程调度（低级调度）三级。

进程调度必不可少

2.高级调度与低级调度的主要功能，引入中级调度的目的

高级调度:

又称为“作业调度”。从用户工作流程的角度。从输入的一批作业中选出若干作业，为其分配必要的内存，建立相应的用户进程和系统进程，然后将程序和数据调入内存，等待进程调度。时间上通常是分钟、小时或天。

低级调度:

又称为“微观调度”、“进程调度”。从CPU资源的角度。时间上通常是毫秒。因为执行频繁，要求在实现时达到高效率。

它是指根据一定的算法，将CPU分派给就绪队列中的一个进程。这级调度是必须有的。

执行低级调度功能的程序称做进程调度程序。进程调度是操作系统中最基本的一种调度。调度策略的优劣直接影响系统的性能。

中级调度:

从存储器资源的角度。将进程的部分或全部换出到外存上，将当前所需部分换入到内存。(指令和数据必须在内存里才能被CPU直接访问。)

中级调度的功能是在内存使用情况紧张时，将一些暂时不能运行的进程从内存对换到外存上等待；(第5章介绍)

3.理解单处理机系统各调度算法的思想、调度方式和特点，包括：先来先服务（FCFS）、短作业优先法（SJF）、高响应比优先法（HRRF）、时间片轮转法（RR）、优先级法（SPF）、多级反馈队列法（MFQ）

（1）先来先服务法FCFS

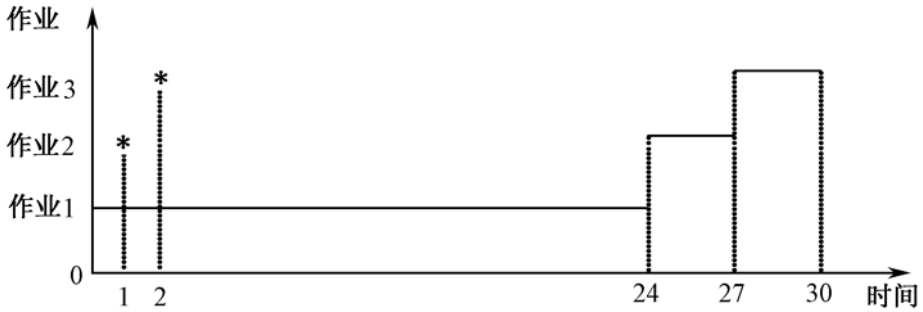
当每个进程创建完成后，该进程就进入就绪队列，并排到就绪队列的队尾。在当前正在运行的进程停止执行时，选择在就绪队列中存在时间最长的进程

运行，即从就绪队列的头部移走一个进程运行。

一个进程一旦分得处理机，便执行下去，直到该进程完成或阻塞时，才释放处理机。

例:

设有三个作业，编号分别为1, 2, 3。各作业分别对应一个进程。各作业依次到达，相差一个时间单位(即0、1、2)，需要运行时间分别为24、3、3。



*表示作业到达的时间，实线表示作业执行过程

作业	到达时间	运行时间	开始时间	完成时间	周转时间	带权周转时间
1	0	24	0	24	24	1
2	1	3	24	27	26	8.67
3	2	3	27	30	28	9.33
平均周转时间T=26			平均带权周转时间W=6.33			

非抢占式调度方式。

优点:

简单，易于理解，便于在程序中运用。

缺点:

有利于长进程，不利于短进程。

有利于CPU繁忙型作业，不利于I/O繁忙型作业。

效率较低。

即可用于作业调度，也可用于进程调度。在实际的OS中，FCFS算法与其他算法结合起来使用.

(2) 短作业优先法SJF

所谓作业的长短是指作业要求运行时间的多少。

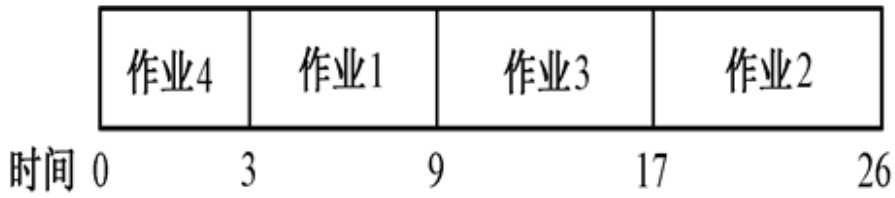
当分配CPU时，选择所需处理时间最短的进程。短进程将越过长进程，跳到队列头。一个进程一旦分得处理机，便执行下去，直到该进程完成或阻塞时，才释放处理机。

这是对FCFS算法的改进，其目标是减少平均周转时间。

例:

考虑表4-2给出的一组作业（它们同时提交到系统）。

作　业	运行时间
1	6
2	9
3	8
4	3



非抢占式

优点:

对于一组给定的作业，SJF能给出较小的平均等待时间，提高了系统的吞吐量。

缺点:

实现上有困难，需要知道或至少需要估计每个作业/进程所需要的处理时间。

对长作业不利。

不能保证及时处理紧迫作业。

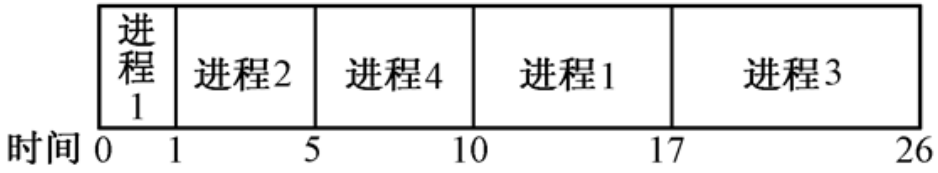
作业调度用的多，进程调度用的少。

(3) 最短剩余时间优先法SRTF

当新进程进入就绪队列时，如果它需要的运行时间比当前运行的进程所需的剩余时间还短，则执行切换，当前运行进程被剥夺CPU的控制权，调度新进程运行。

例:

进　程	到达时间	运行时间
1	0	8
2	1	4
3	2	9
4	3	5



抢占式

优点:

保证新的短作业一进入系统就能很快得到服务，平均等待时间短。

缺点：

为保存进程断点现场，统计进程剩余时间增加了系统开销。

不利于长作业。

作业调度用的少，进程调度用的多

（4）高响应比优先法HRRF

基于过去的历史或用户和配置管理员的某些输入值近似估计服务时间。为每个进程计算一个响应比RR=（w+s）/s。在调度进行时，以各进程的响应比作为其优先级，从中挑选级别最高的进程投入运行。

w是进程等待处理机所用的时间;

s是进程要求的服务时间。

例：

进程	到达时间	要求服务时间
P1	0	8
P2	5	4
P3	7	6
P4	9	3

请算出进程的执行序列。

P1,P2,P4,P3

特点：

高响应比优先法是一种非抢占方式。

兼顾短进程、长进程。

调度前需要计算进程的响应比，增加系统开销。

对实时系统无法做出及时反应。

（5）轮转法RR

时间片轮转法（Round-Robin）主要用于分时系统中的进程调度。

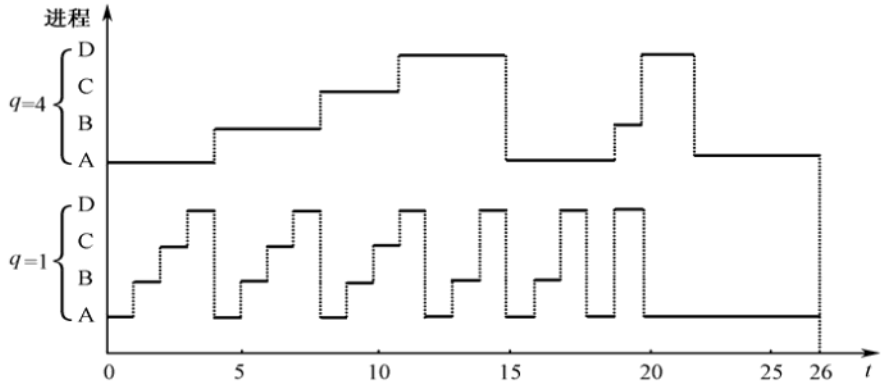
时间片是一个小的时间单位，通常为10～100 ms数量级。

将系统中所有的就绪进程按照FCFS原则，排成一个队列。每次调度时将CPU分派给队首进程，让其执行一个时间片。时间片的长度从几个ms到几百ms。在一个时间片结束时，发生时钟中断。

调度程序据此暂停当前进程的执行，将其送到就绪队列的末尾，并通过上下文切换执行当前的队首进程。进程可以未使用完一个时间片，就出让CPU（如阻塞）。

例：

A,B,C,D同时到达，需运行时间为12,5,3,6个时间单位，试求时间片分别为1个和4个时间单位时，进程的运行情况?



轮转法*q*=1和*q*=4时进程运行情况

到达时间	名	到达时间	运行时间	开始时间	完成时间	周转时间	带权周转时间
时间片 q=1	A	0	12	0	26	26	2.17
	B	0	5	1	17	17	3.4
	C	0	3	2	11	11	3.67
	D	0	6	3	20	20	3.33
	平均周转时间T=18.5			平均带权周转时间 W=3.14			
时间片q =4	A	0	12	0	26	26	2.17
	B	0	5	4	20	20	4
	C	0	3	8	11	11	3.67
	D	0	6	11	22	22	3.67
	平均周转时间 T=19.75			平均带权周转时间 W=3.38			

特点:

抢占式调度。

简单易行，平均响应时间短。

时间片的大小直接影响轮转法的性能。

不利于处理紧急作业。

(6) 优先级法HPF

优先级调度算法是从就绪队列中选出优先级最高的进程，让它在CPU上运行。

可用于作业调度或进程调度。

优先数:优先级一般用某个固定范围内的整数表示。此种整数称为优先数。

本书采用“优先数小、优先级高”的表示方式。

进程优先级可由系统内部定义或由外部指定

确定进程优先级的方式有静态方式和动态方式两种。

静态优先级是在创建进程时就确定下来的，而且在进程的整个运行期间保持不变。

进程类型（系统进程优先级较高）

对资源的需求（对CPU和内存需求较少的进程，优先级较高）

用户要求（紧迫程度和付费多少）

动态优先级随进程的推进而不断改变。

在就绪队列中，等待时间延长则优先级提高。(解决饥饿问题)

进程每执行一个时间片，就降低其优先级。(实现负反馈，防止长期占用CPU)

例:

进程名	到达时间	处理时间	优先级
A	0	3	1
B	1	5	3
C	3	4	2
D	7	8	4
E	12	5	5

优先级1<2<3<4<5（数越大，优先级越大），忽略进程切换的开销。

1、采用非抢占的优先级调度算法，执行顺序？

A3,B5,D8,E5,C4, 周转时间：[3+(8-1)+(16-7)+(21-12)+(25-3)]/5

2、采用抢占的优先级调度算法，执行顺序？

A1,B5,C1,D5,E5,D3,C3,A2

(7) 多级反馈队列法MFQ

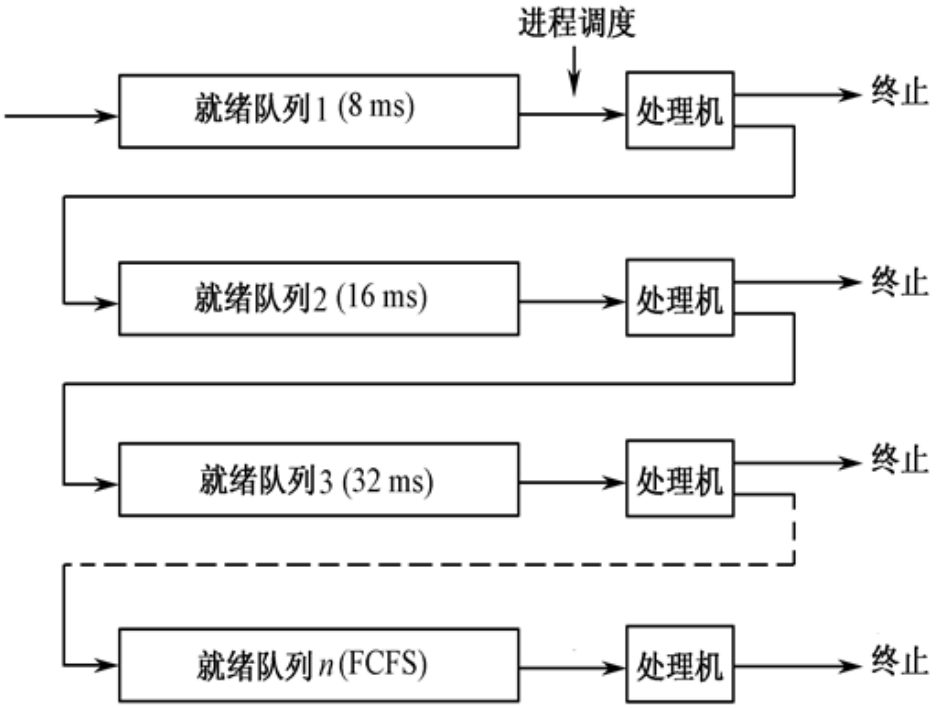
MFQ实现思想:

①系统中设置多个就绪队列，每个队列对应一个优先级；

②各就绪队列中进程的运行时间片不同，高优先级队列的时间片小，低优先级队列的时间片大；

③新进程进入系统后，先放入第1个队列的末尾，如果在时间片内工作未完成，则转入下一个队列尾，依此类推；

④系统先运行第1个队列中的进程,若第1队列为空,才运行第2队列，依次类推；



这种调度算法基于抢占式，使用动态优先级机制。

长作业自动沉到下面低优先级的队列中，如果之后进入系统的都是短作业，并形成稳定的作业流，则长作业可能一直等待，处于“饥饿”状态；

解决办法是提升等待时间长的进程的优先级；

特点：

抢占式调度

动态优先级

需要解决“饥饿”问题

是最通用的CPU调度算法,也最复杂.

4.熟练掌握：对于给定的一组作业或进程，能使用以上算法给出调度过程，并会计算周转时间和平均周转时间＼平均带权周转时间。类型题练习：见ppt。

第五章

1.物理地址、逻辑地址、物理地址空间和逻辑地址空间的概念

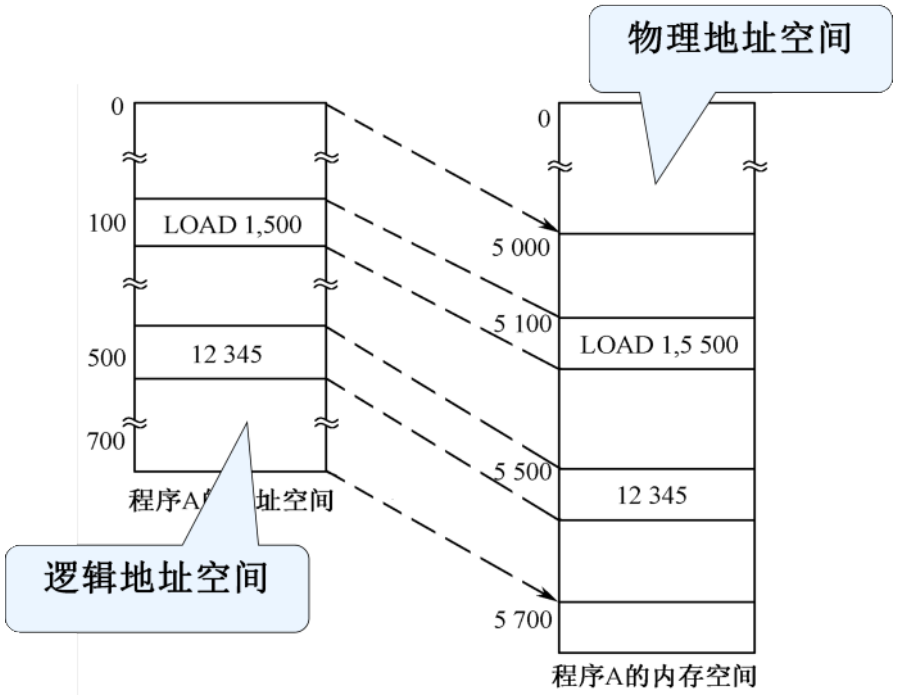
物理地址：

内存中各物理存储单元的地址是从统一的基地址开始顺序编址的，这种地址称为绝对地址或物理地址。

逻辑地址：

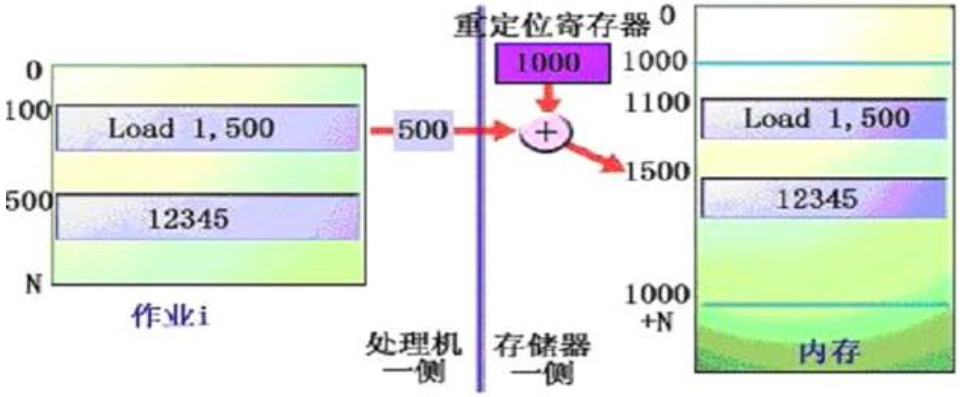
用户程序经编译之后的每个目标模块都以0为基地址顺序编址，其余指令中的地址都相对于首地址而编址。这种地址称为相对地址或逻辑地址；

物理地址空间和逻辑地址空间



2.动态重定位实现

动态重定位:在程序运行过程中要访问数据时再进行地址变换。由地址变换机构进行的地址变换，硬件上需要重定位寄存器的支持。



优点:

OS可以将一个程序分散存放于不连续的内存空间，可以移动程序。

有利于实现共享。它是虚拟存储的基础。

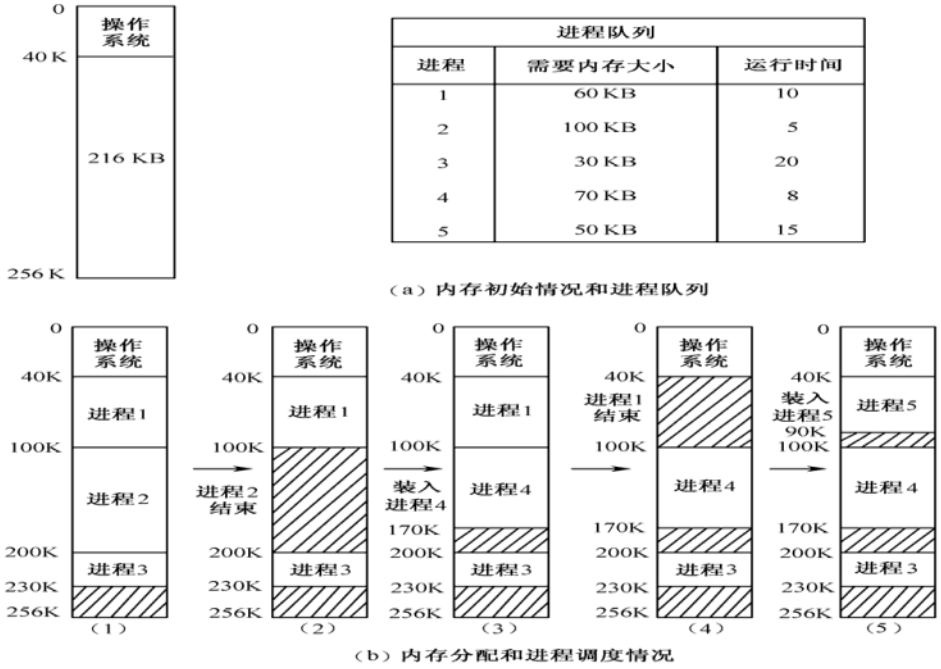
缺点:

需要硬件支持（通常是CPU），OS实现较复杂。

3.掌握动态分区分配算法的基本思想，包括最先适应算法、最佳适应算法、循环适应算法，注意空闲分区表的排列方式，会解题。

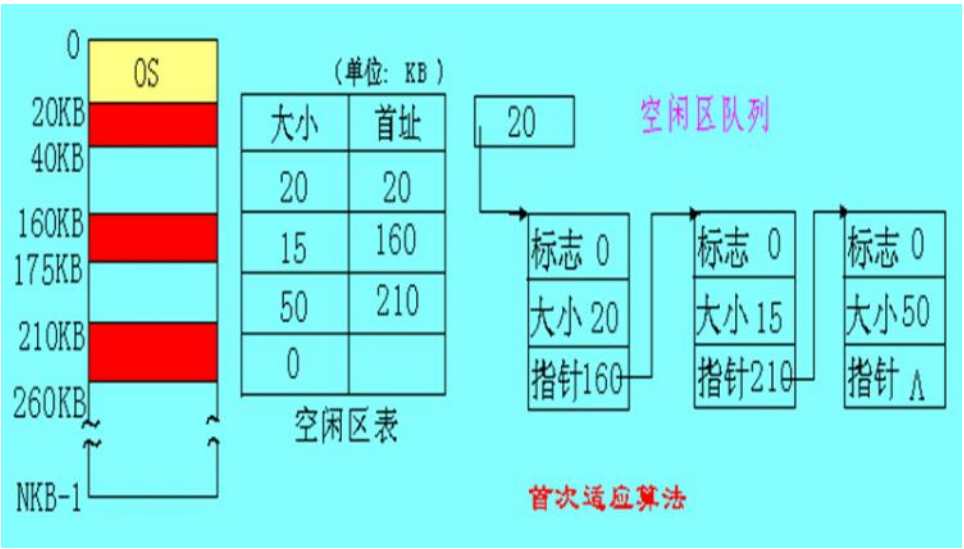
基本思想：

内存不是预先划分好的，而是当进程装入时，根据进程的需求和内存空间的使用情况来决定是否分配。若有足够的空间，则按需要分割一部分分区给该进程；否则令其等待主存空间。



最先适应法分配算法

空闲表按位置排列（空闲块地址小，在表中的序号也小）。
分配内存时，在各空闲分区中查找满足大小要求的可用块。找到第一个可满足要求的空闲块就停止查找，并进行分配。
若空闲空间全部占用则从空闲表取消该项，若有剩余，则余下部分仍保留在空闲表中，并更新分区大小和分区地址。



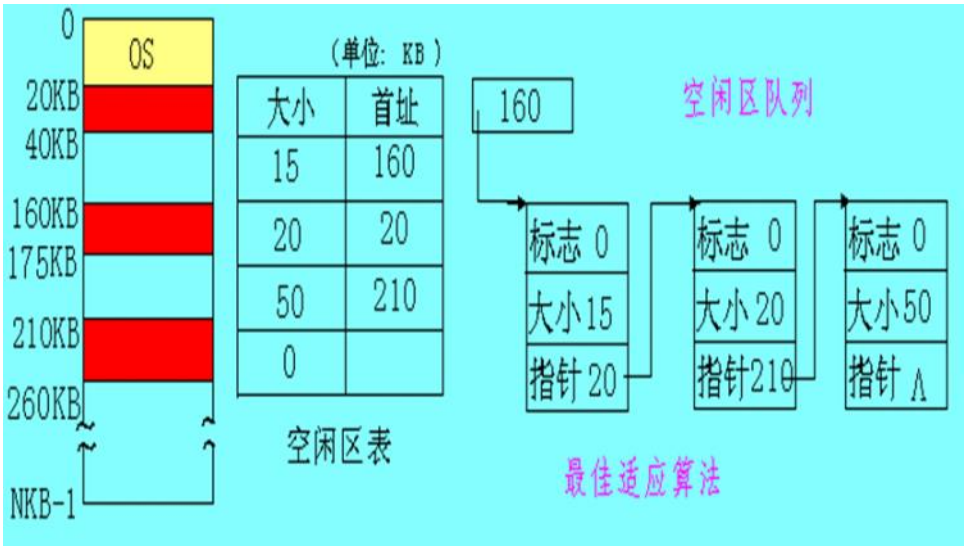
特点:

该算法的分配和释放的时间性能较好，较大的空闲分区可以被保留在内存高端。
但随着低端分区不断划分而产生较多小分区，每次分配时查找时间开销会增大。

最佳适应算法

空闲表以空闲块大小为序，按增量形式排列。接到内存申请时，在空闲块表中找到一个不小于请求的最小空闲块进行分配。
为作业选择分区时总是寻找其大小最接近于作业所要求的存储区域。
特点：

用最小空间满足要求。



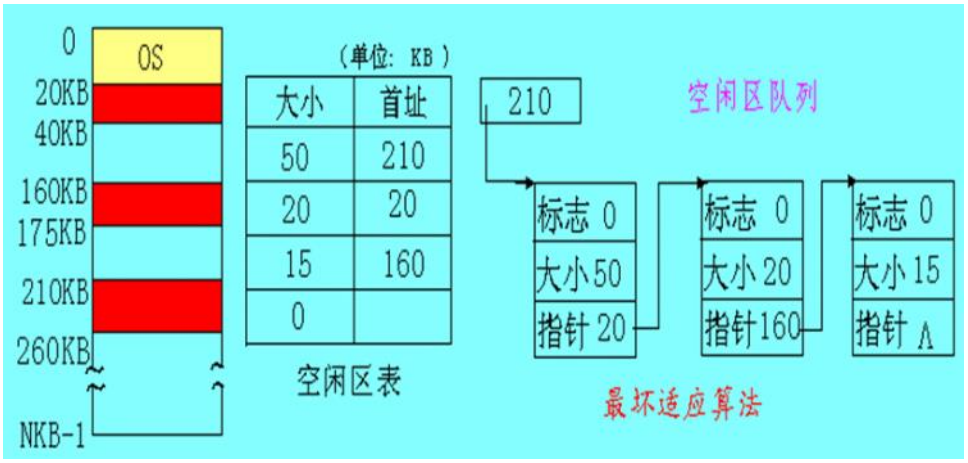
特点：

个别来看，碎片较小，但从整体来看，会形成较多外碎片。较大的空闲分区可以被保留

循环适应算法

接到内存申请时，在空闲块表中，上次找到的可用分区的下一个空闲分区开始查找可满足大小要求的第一个空闲分区。

该算法的分配和释放的时间性能较好，使空闲分区分布得更均匀，但较大的空闲分区不易保留。



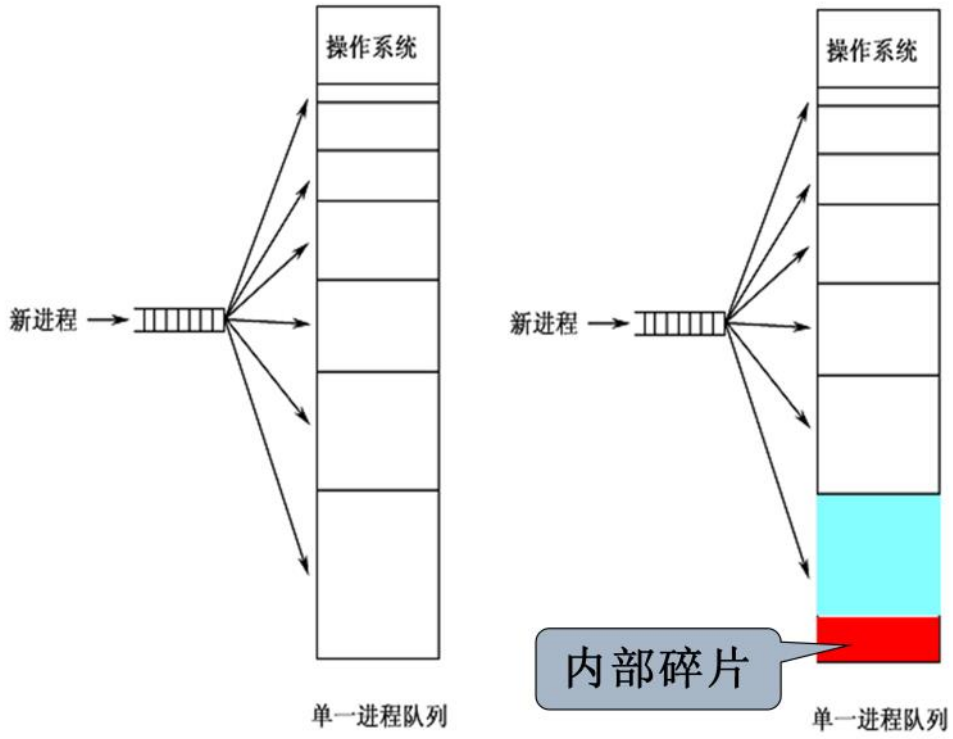
特点：

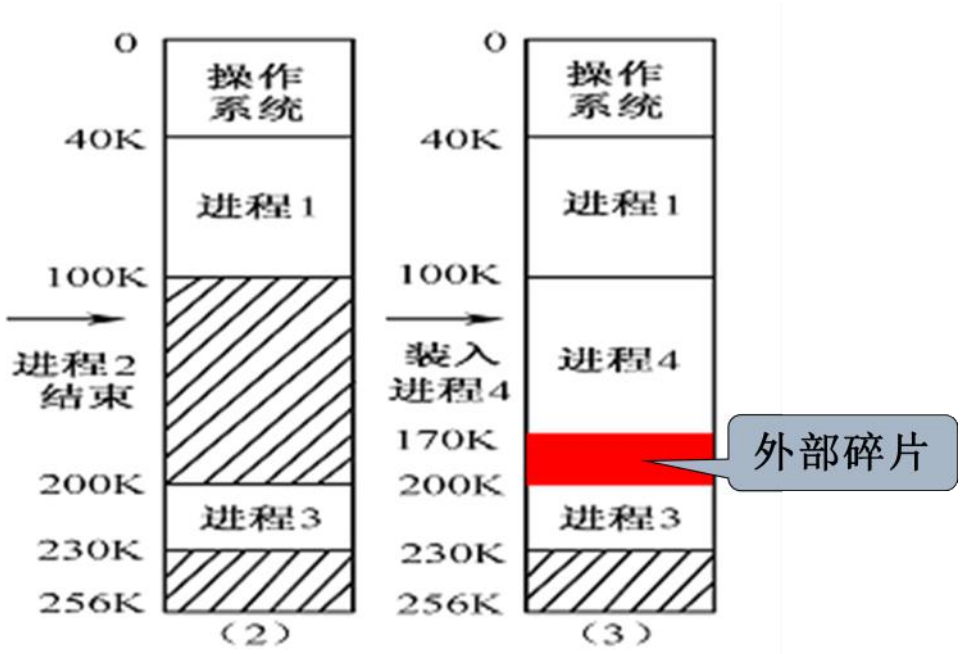
较大的空闲分区不被保留。

4.区分内部碎片和外部碎片

在一个分区内部出现的碎片（即被浪费的空间）称做内部碎片，如固定分区法会产生内部碎片。

在所有分区之外新增的碎片称做外部碎片。





5.虚拟存储器的定义及特征

定义：

借助于外存空间，允许一个进程在其运行过程中部分装入内存。虚拟存储系统将内存和外存有机结合在一起，从而得到一个容量相当于外存，速度接近于内存的存储体系

特征：

① 虚拟扩充。

② 部分装入。

③ 离散分配。

④ 多次对换。

6.系统抖动的概念和产生的原因

抖动

在虚存中，页面在内存与外存之间频繁调度，以至于调度页面所需时间比进程实际运行的时间还多，此时系统效率急剧下降，甚至导致系统崩溃。这种现象为“抖动或颠簸（Thrashing）”。

产生抖动的原因：

页面淘汰算法不合理。

分配给进程的物理页面数太少。

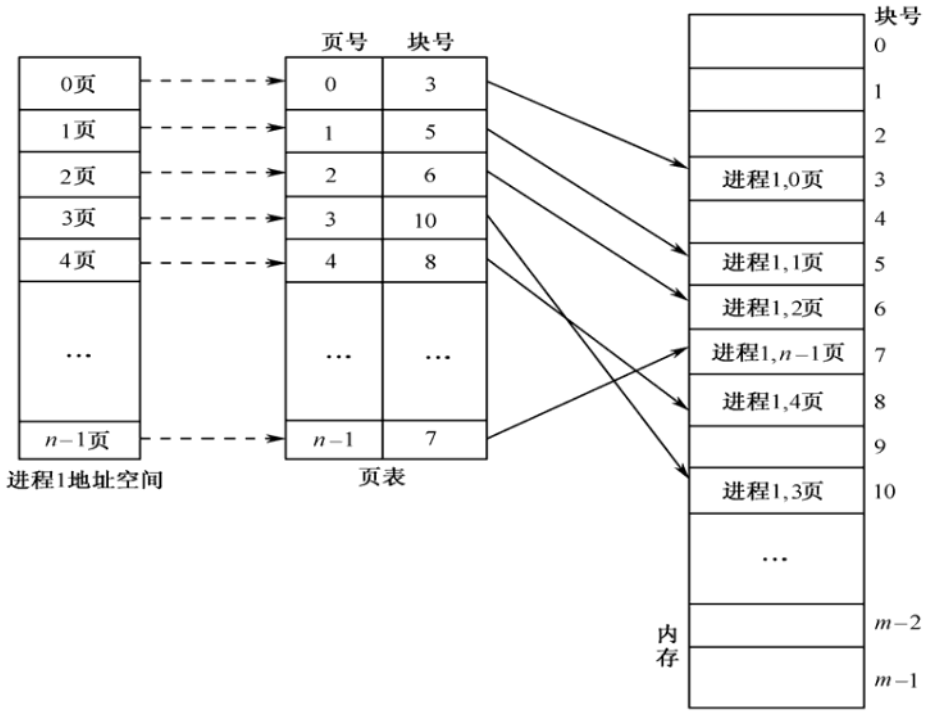
7.理解分页技术、分段技术、段页式技术、请求分页技术和请求分段技术的基本原理，页表和段表的作用，地址转换过程，会简单计算。

分页存储管理的基本概念

（1）页面：将一个进程的逻辑地址空间分成若干个大小相等的部分，称为页面或页。编号如第0页，第1页等。

（2）内存块：把内存划分成与页面大小相等的存储块，称为内存块或页框。编号如0#，1#等

（3）内存分配：把一个进程页面分别装入到一些物理上不连续的内存块中。进程的每个页面对应某一个内存块。



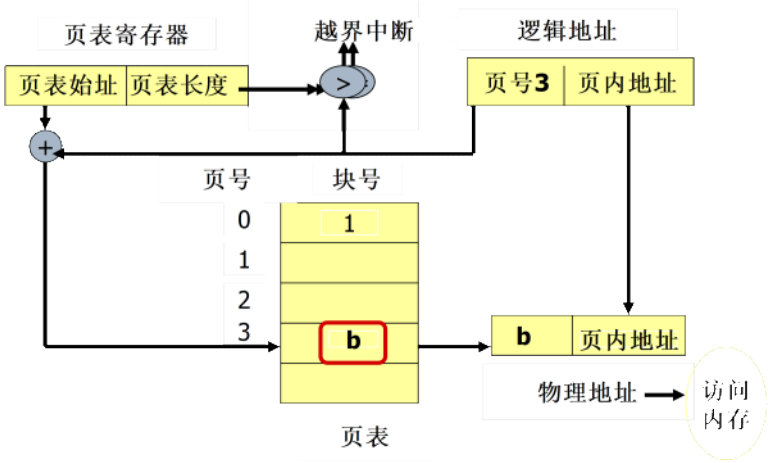
页表

系统为每一个进程在内存中建立了一张页面映像表；

作用:

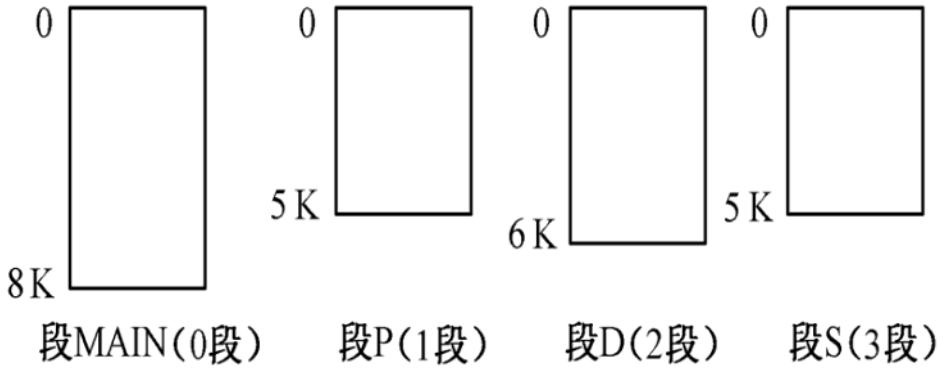
表示页号到物理块号的地址映射。

地址转换



分段存储管理的基本概念

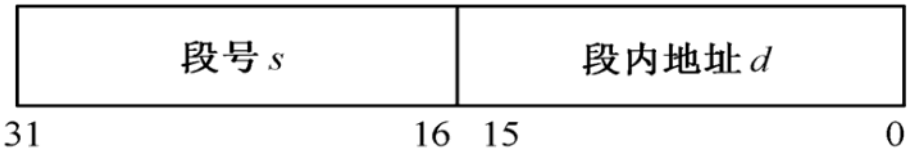
(1）分段：每段都有自己的名字和长度。



(2）程序的地址结构

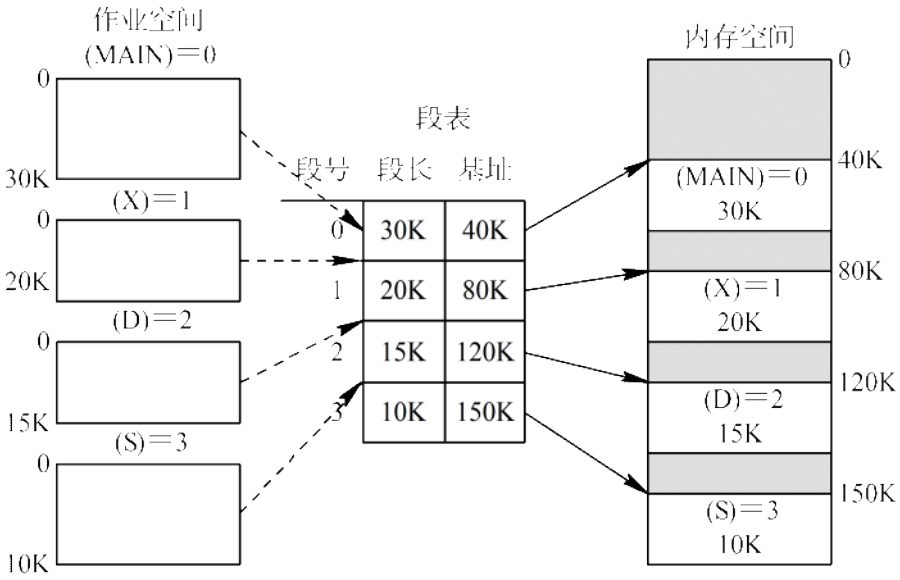
逻辑地址要用两个成

分来表示：段号s和段内地址d。



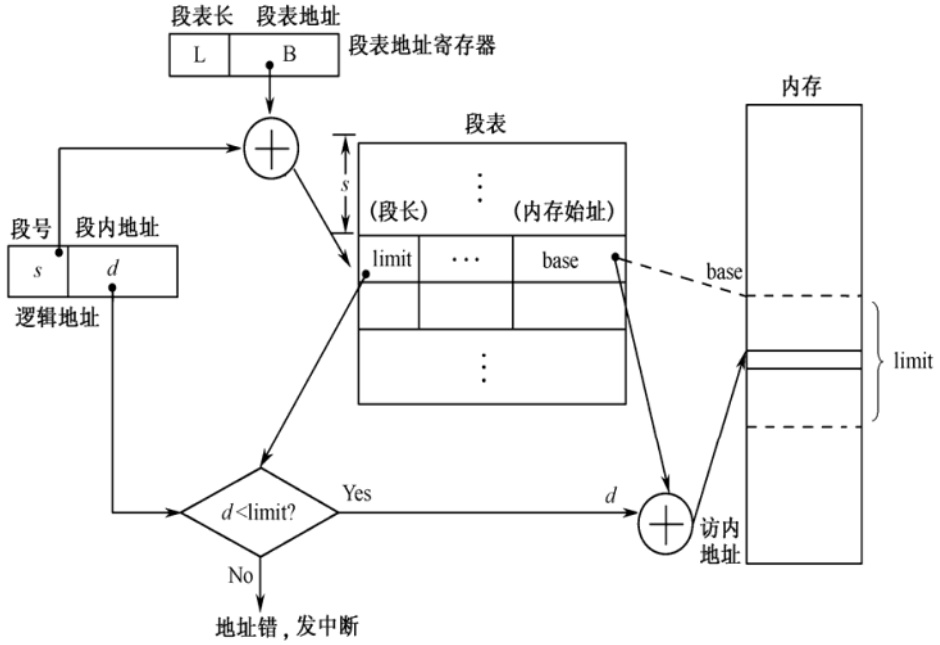
(3）内存分配

内存以段为单位进行分配，每段单独占用一块连续的内存分区。各分区的大小由对应段的大小决定。



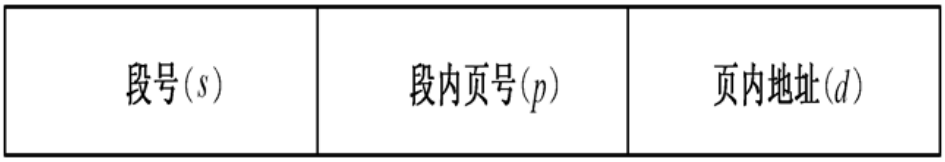
系统为每个进程建立一个段映射表，简称“段表”。每个段在段表中占有一项，段表项中包含段号、段长和段起始地址（又称“基址”）等。

地址转换



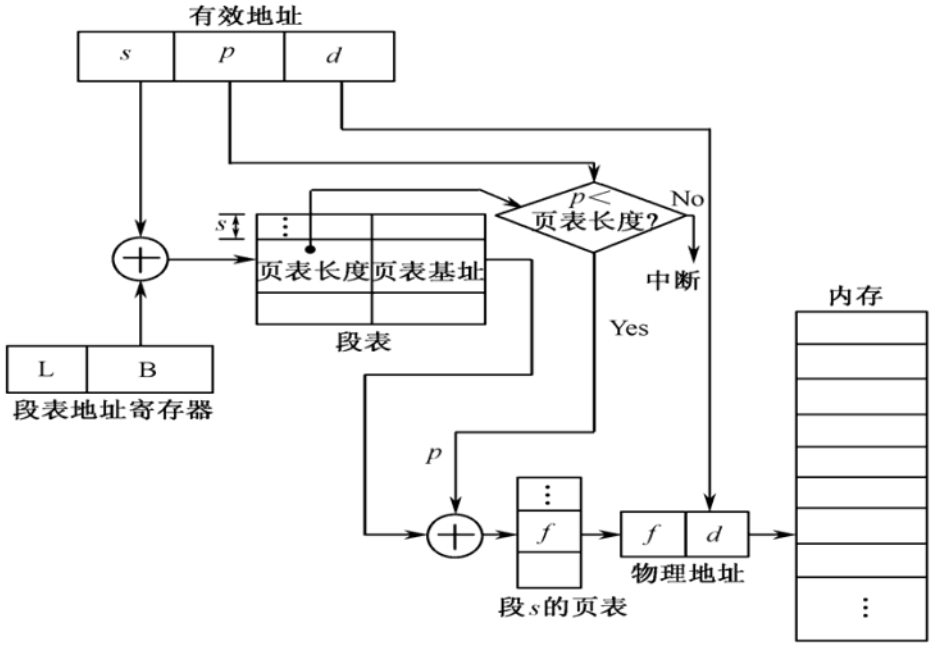
段页式存储管理的基本原理

把逻辑段再分成物理页



以上两种技术的结合

地址转换

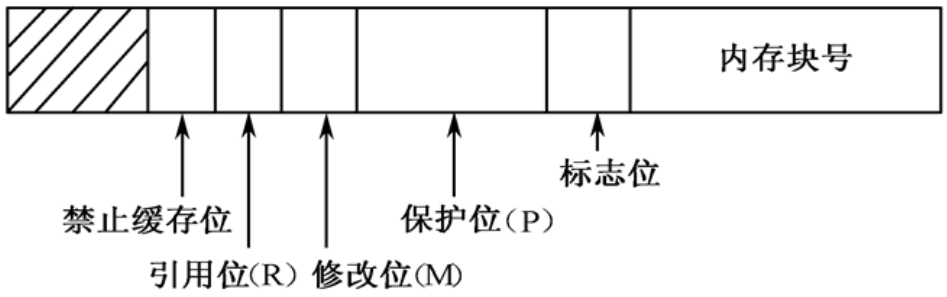


请求分页存储管理的基本思想

为了标示进程的页面是否已在内存，在每个页表项中增加一个标志位。

页表机制

页表项通常包含下列5种信息：



① 内存块号。

② 标志位。这一位用来标示对应的页面是否已装入内存。

如果该位是1，表示该表项是有效的，可以使用，即该页在内存中；

如果该位是0，则表示该表项对应的页面目前不在内存中，访问该页会引起缺页中断。

③ 保护位：访问权限：读、写、执行。

④ 修改位和引用位。这两位用来记录该页的使用状况。

⑤ 禁止缓存位。该位用于禁止该页被缓存。

请求分段技术

带虚存的段式存储管理技术,一个进程只有部分分段放在内存,仅用到某个段时才进行链接。

请求分段存储管理的硬件支持

各段表项中要增加一位，以表明该段的存在状态(内存\外存)。

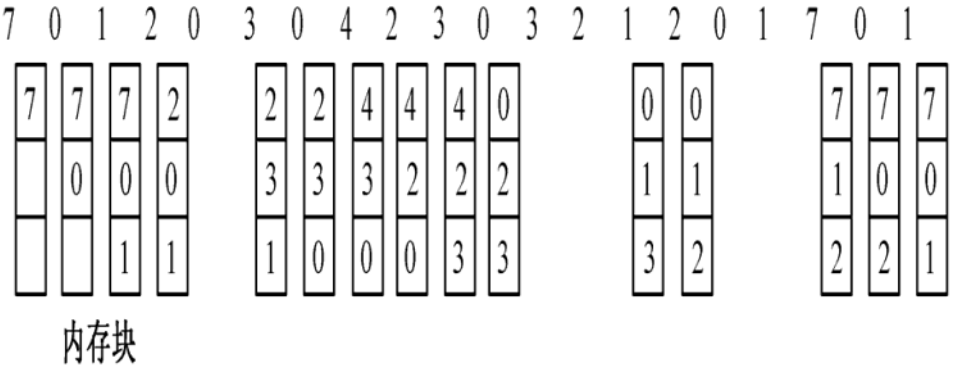
还要增加另外一些控制位，如修改位\保护位\共享位。

8.对于以上内存管理技术，给出逻辑地址，会计算物理地址。类型题练习：书后第10、13题。

9.理解各种页面置换算法的基本思想。熟练掌握FIFO、LRU、OPT三种页面置换算法，对于给定的地址访问序列能求出简化的页面走向，能画出不同置换算法的页面调度情况，并会计算缺页次数。类型题练习：书后第15、16题。

先进先出法（FIFO）

总是淘汰在内存中停留时间最长的一页，即先进入内存的页，先被换出。



优点:

容易理解且方便程序设计。

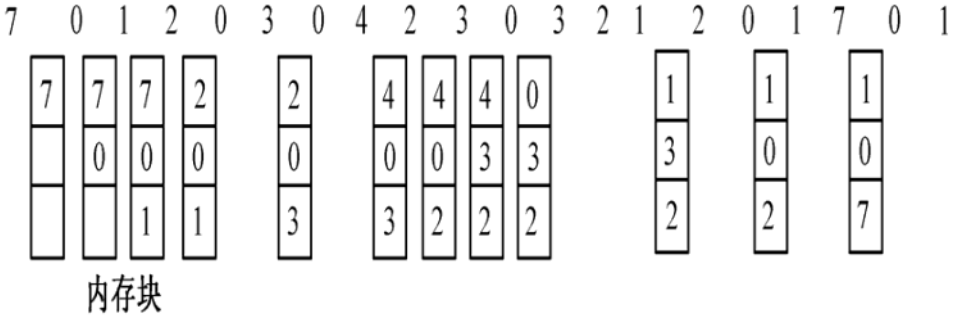
缺点:

性能并不很好。

存在Belady异常现象，即缺页率随内存块增加而增加。

最近最少使用置换法(LRU)

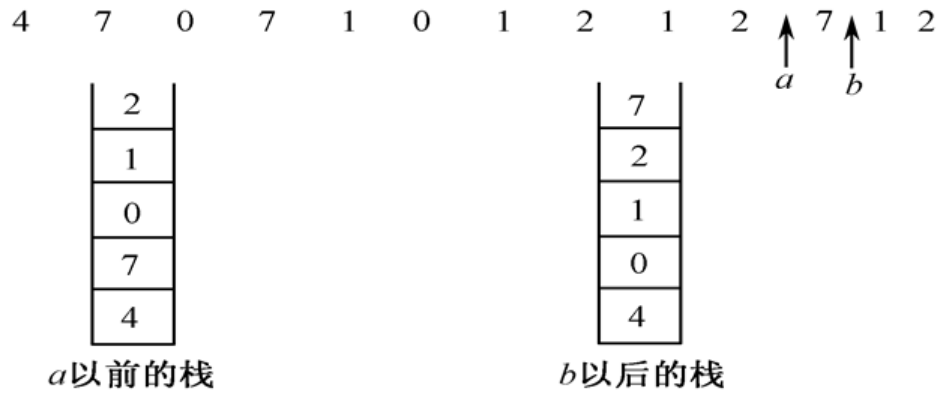
思想：当需要置换一页时，选择在最近一段时间里最久没有使用过的页面予以淘汰。



LRU算法需要实际硬件的支持。实现时的问题是：怎样确定最后访问以来所经历时间的顺序。有以下两种可行的办法：

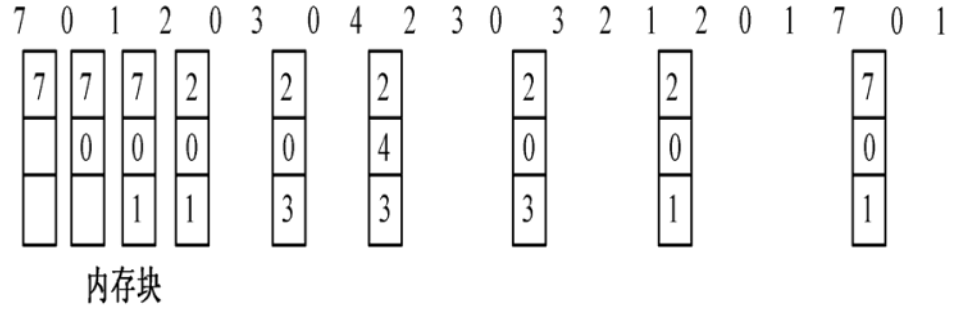
① 计数器。

② 栈。



最佳置换法（OPT）

实质：为调入新页面而必须预先淘汰某个老页面时，所选择的老页面应在将来不被使用，或者是在最远的将来才被访问。



第六章

1.文件系统的概念及主要功能

文件管理系统，简称文件系统。

就是操作系统中负责操纵和管理文件的一整套设施，实现文件的共享和保护，方便用户按名存取。

一般来说，文件系统应具备以下5种功能：

①文件管理。创建、删除、读写、执行等。

② 目录管理。每个文件创建一个目录项，若干目录项构成一个目录文件。可进行文件检索和权限限制。

③文件存储空间管理。对文件存储空间的分配与回收、文件逻辑地址与物理地址的映射。

④文件的共享和保护。共享、保护、转储、恢复。

⑤提供方便的接口。用户观点（方便、可靠、共享、保护）、系统的观点（存储空间组织、分配、信息传输等）。

2.区分流式文件和记录式文件

无结构文件

无结构文件是指文件内部不再划分记录，是由一组相关信息组成的有序字符流，即流式文件。在UNIX和Windows系统中文件都是有序字符流，长度直接按字节计算。为操作系统带来灵活性。

有结构文件

有结构文件又称记录式文件。它在逻辑上可被看成一组连续记录的集合，即文件是由若干相关记录组成，且对每个记录编上号码 。

①定长记录文件。

②变长记录文件。

3.目录文件、目录项、文件控制块的概念，三者的关系

为了加快对文件的检索，往往将文件控制块集中在一起进行管理。

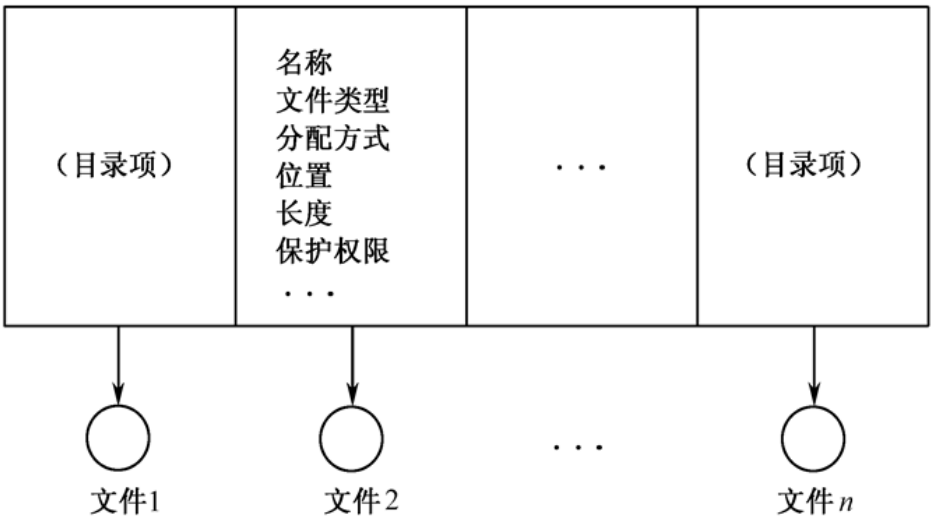
这种文件控制块的有序集合称为文件目录。

文件控制块就是其中的目录项。
完全由目录项构成的文件称为目录文件。

4.单级目录结构、二级目录结构、树型目录结构、非循环图目录结构的特点

单级目录结构

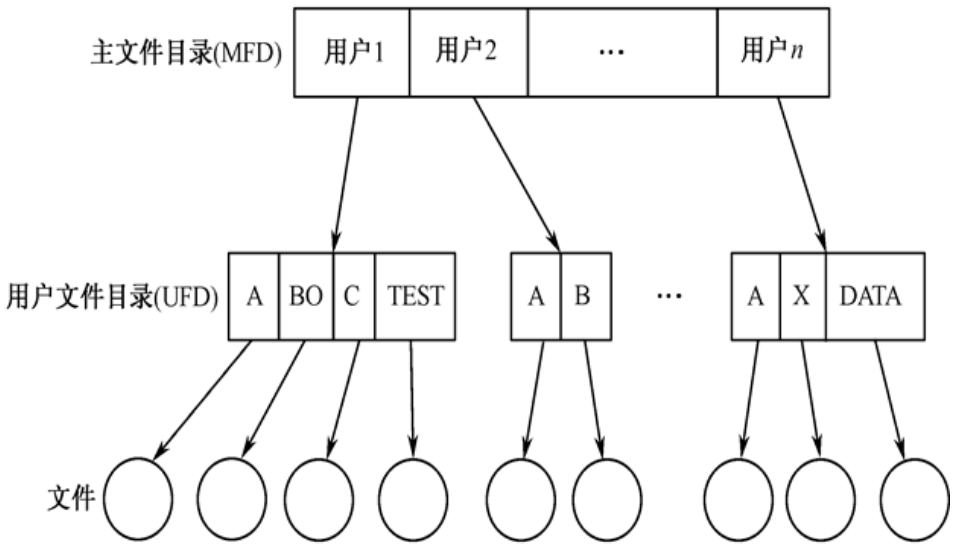
在这种组织方式下，全部文件都登记在同一目录中。每个文件对应一个目录项。



优点是简单，能够实现按名存取。
单级目录结构有以下三个缺点：
①查找速度慢。
②不允许重名。
③不便于共享。

二级目录结构

每个用户有自己的用户文件目录。

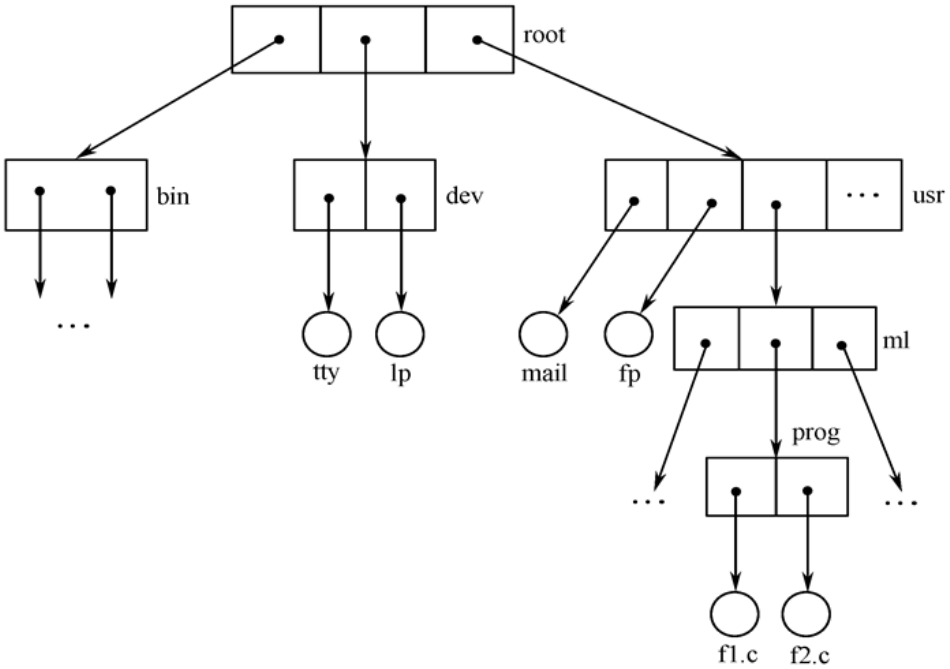


优点是：
不同用户可有相同的文件名；提高了检索目录的速度；不同用户可用不同的文件名访问系统中同一文件。
缺点是这种结构仍不利于文件共享。

树型目录结构

从根目录开始，一层一层地扩展下去，形成一个树形层次结构，每个目录的

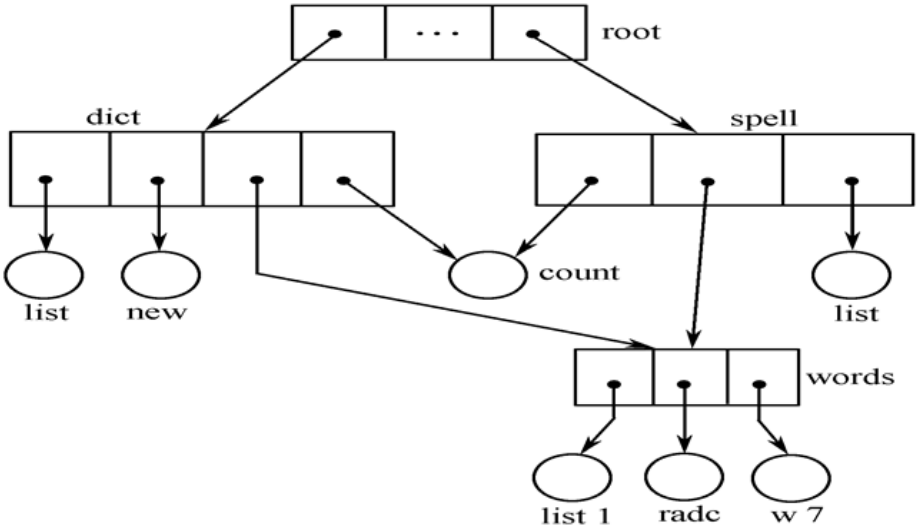
直接上一级目录称做该目录的父目录，而它的直接下一级目录称做子目录。系统中的每个文件都有唯一的路径名。



文件的层次和隶属关系很清晰，便于实现不同级别的存取保护和文件系统的动态装卸。但是，在上述纯树形目录结构中，只能在用户级对文件进行临时共享。

非循环图目录结构

它允许一个文件或目录在多个父目录中占有项目，但并不构成环路。这种结构方式叫做链接（Link）。

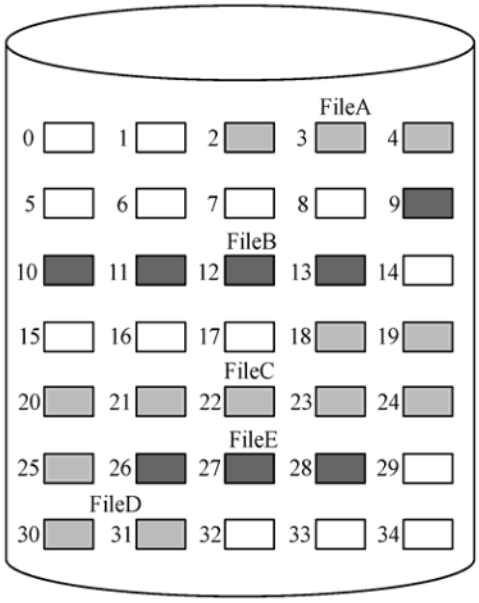


5.常用的文件分配方法有哪几种

常用文件分配方法有:

连续分配、链接分配、索引分配三种。

连续分配



文件分配表

文件名	起始块	长度
FileA	2	3
FileB	9	5
FileC	18	8
FileD	30	2
FileE	26	3

优点：

在顺序存取时速度较快，一次可以存取多个盘块，改进了I/O性能。另外，也很容易直接存取文件中的任意一块。

连续分配也存在如下缺点：

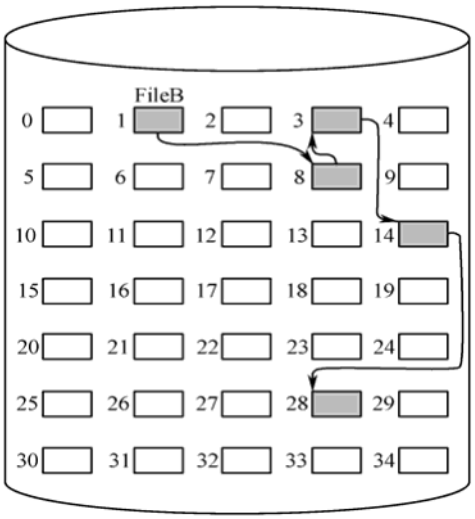
①要求建立文件时就确定它的长度，依此来分配相应的存储空间，这往往很难实现。

②它不便于文件的动态扩充。

③可能出现外部碎片。

链接分配

把一个逻辑上连续的文件分散存放在不同的物理块中，这些物理块不要求连续，也不必规则排列。FAT表，每个文件在表中占一项。



文件分配表

文件名	起始块	最后块
…	…	…
FileB	1	28
…	…	…

优：

①采用链接分配不会产生磁盘的外部碎片。

②文件可以动态增长。

③不需要紧缩磁盘空间。

缺：

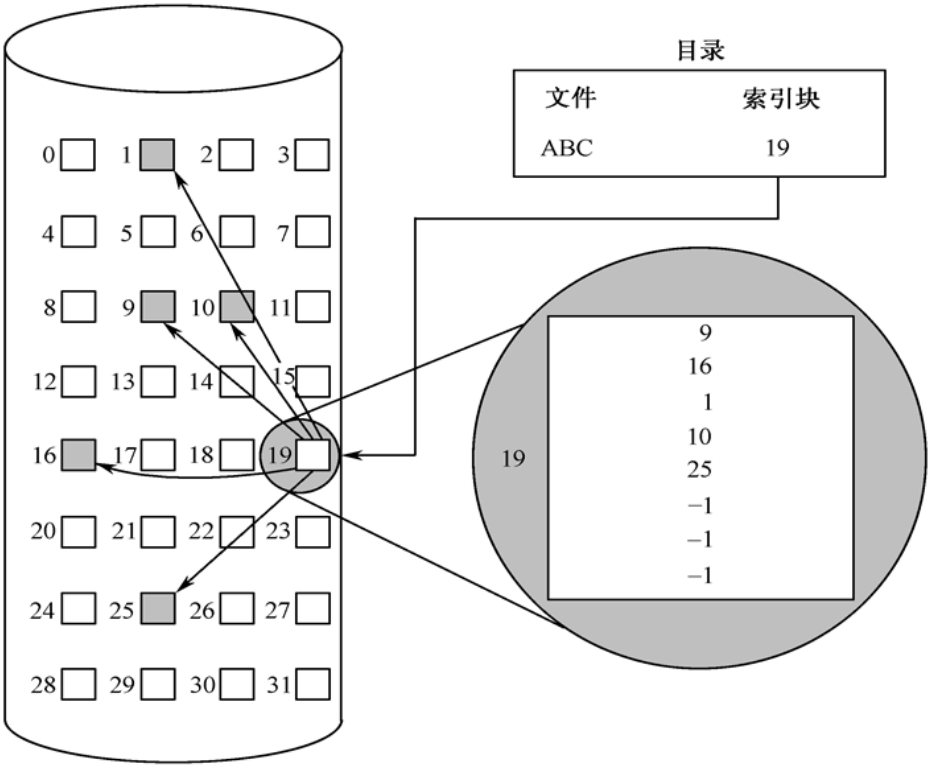
①一般仅适于对信息的顺序访问，而不利于对文件的随机存取。

②每个物理块上增加一个链接字。

③可靠性。丢失、故障。

索引分配：

系统为每个文件建立一个索引表，其中的表项指出存放该文件的各个物理块号。



优：

索引文件除了具备链接文件的优点外，还克服了它的缺点。它可以方便地进行随机存取。但是这种组织形式需要增加索引表带来的空间开销。

缺：

存取文件的速度有影响。

6.课后题13题，理解含义，会计算。

在实现文件系统时，为加快文件目录的检索速度，可利用“文件控制块分解法”。假设目录文件存放在磁盘上，每个盘块512字节。文件控制块占64字节，其中文件名占8字节。通常将文件控制块分解成两部分，第1部分占10字节（包括文件名和文件内部号），第2部分占54字节（包括文件内部号和文件其他描述信息）。

（1）假定某一目录文件共有254个文件控制块，试分别给出采用分解法前和分解法后，查找该目录的某一个文件控制块的平均访问磁盘次数。

（2）一般地，若目录文件分解前占用n个盘块，分解后改用m个盘块存放文件名和文件内部号，请给出访问磁盘次数减少的条件。

答：

*（1）采用分解法前，**对于254个文件控制块的目录文件，其一个控制块的平*

均查找次数为254/2，每个控制块为64B，所需访问磁盘的内容为64

**（254/2），磁盘每个盘块512B，所以，访问磁盘次数为64*

**（254/2）/512。分解法后，访问磁盘次数为10*（254/2）/512+1*

（2）访问磁盘次数减少的条件为：（n＋1）/2＞（m＋1）/2＋1 即m＜n－2

网上答案:

（1）采用分解法前，一个盘块存放[5I2/64]=8目录项，254个目录项需要32个盘块，查找一个文件的平均访问的盘块数：(1+32)/2=16.5次； 采用分解法后，一个盘块存放[5I2/10]=51目录项，254个目录项需要5个盘块，查找一个文件的第1部分平均访问的盘块数：(1+5)/2=3次；查找第2部分需要访问磁盘1次，故查找一个文件控制块的平均访问磁盘次数是3＋1＝4次。

（2）访问磁盘次数减少的条件为：（n＋1）/2＞（m＋1）/2＋1 即m＜n－2

7.文件控制块分解法概念。

利用“文件控制块分解法”可以加快文件目录的检索速度。在UNIX系统中就采用类似方法。其原理是减少因查找文件内部号而产生的访问磁盘次数。因此在查找文件内部号的过程中不需要把文件控制块的所有内容都写入内存，只要把文件名和文件内部号这一部分写入内存即可，从而减少所需读入的存储块，就有可能减少访问磁盘的次数。当找到所需的文件控制块后，要把该文件控制块的全部内容读入内存，它还需要访问一次磁盘。因而，在一定条件下采用这种方法并不能减少访问磁盘的次数。

第七章

1.六种I/O控制方式是什么，理解中断控制方式的工作过程、DMA方式的特点、通道的作用。

程序控制直接传递方式

程序查询方式

忙坏了CPU，浪费。

中断控制方式

其基本工作过程是：

①CPU执行设备驱动程序，发出启动I/O设备的指令，使外设处于准备工作状态。然后，CPU继续运行程序，进行其他信息的处理。

②I/O控制器按照I/O指令的要求，启动并控制I/O设备的工作。

③当输入就绪、输出完成或发生错误时，I/O控制器便向CPU发送一个中断信号。

④CPU接收到中断信号后，保存少量的状态信息。

然后将控制传送给中断处理程序。

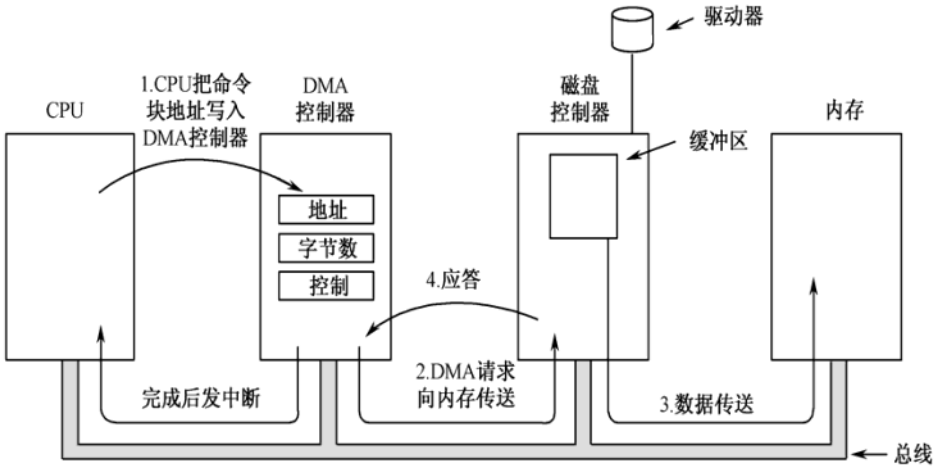
⑤ 中断处理程序确定中断原因，执行相应的处理工作，最后退出中断，返回中断前的执行状态。

⑥CPU恢复对被中断任务的处理工作。

直接存储器访问方式

①DMA控制方式的引入。原来的中断每个字节需要CPU干预，DMA每个数据块才需要CPU干预。

②DMA的传送操作。



独立通道方式

通道的引入

为使CPU摆脱繁忙的I/O事务，现代大、中型计算机都设置了专门处理I/O操作的机构，这就是通道。

通道程序由通道执行的指令组成。相当于一台小型的通道处理机。

通道类型

①字节多路通道。它以字节作为信息输送单位，服务于多台低速I/O设备。

②选择通道。它在同一时间里只能为一台设备服务。主要用于连接高速外部设备。

③成组多路通道。它结合字节多路通道分时操作和选择通道高速传送的优点，广泛用于连接高速和中速设备。

I/O处理器方式：

大中型机器专用处理器。

2.SPOOLing系统的组成，理解采用SPOOLing技术将打印机变成共享设备的过程

系统组成：

输入井和输出井

输入井：虚拟低速输入设备，暂存从输入设备预输入的信息；

输出井：虚拟低速输出设备，暂存要缓输出到输出设备的信息。

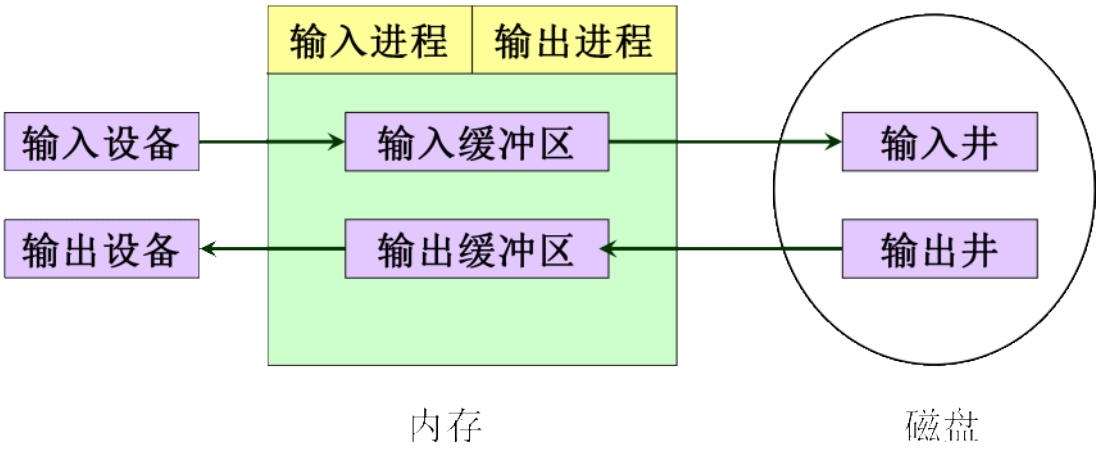
输入缓冲区和输出缓冲区

输入缓冲区：用作输入设备和磁盘输入井之间的中转站；

输出缓冲区：用作磁盘输出井和输出设备之间的中转站。

输入进程和输出进程

输入进程：模拟脱机输入时的卫星输入机，将数据从输入设备经过输入缓冲区送到输入井。当CPU需要读取数据时，直接从输入井中提取数据到内存；
输出进程：模拟脱机输出时的卫星输出机，将应用进程要输出的数据送到输出井，当输出设备空闲时，将输出井中的数据经过输出缓冲区送到输出设备输出。



利用SPOOLing技术共享打印机

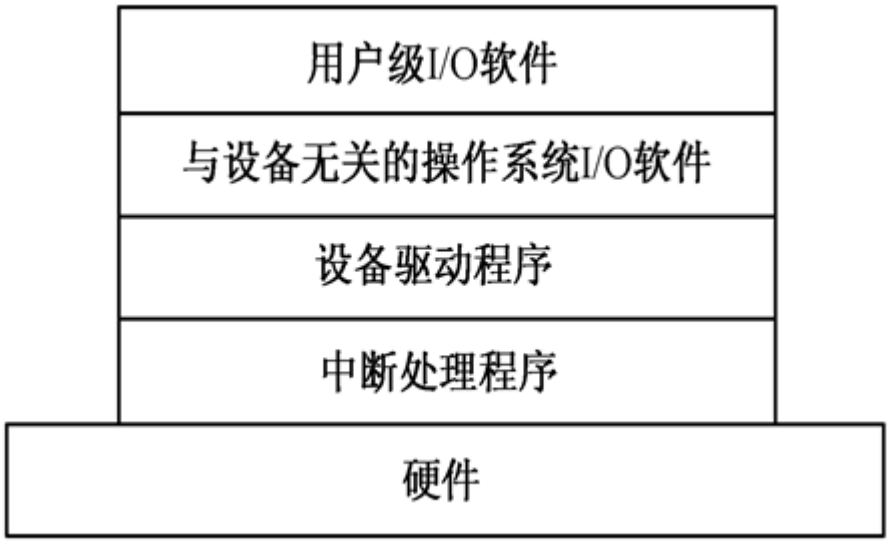
当用户进程请求打印输出时，SPOOLing系统立即同意为它打印输出，但并不真正把打印机分配给该用户进程。

由输出进程在输出井中为之申请一个空闲的磁盘块区，并将要打印的数据送入其中。

输出进程再为用户进程申请一张空白的用户请求打印表，并将用户的打印要求填入其中，然后将该表挂到打印机的请求打印队列上 。

如果打印机空闲，输出进程将从请求队列的队首取出一张请求打印表，根据表中的要求将要打印的数据从输出井传送到内存缓冲区，再由打印机进行打印。打印完后，输出进程将再检查请求队列中是否还有待打印的请求表，若有继续打印，否则便将自己阻塞起来，并在下次再有打印请求时被唤醒。

3.I/O软件有哪几层



4.在设备管理中引入缓冲技术的作用

引入缓冲技术的主要目的是：

① 缓解CPU与I/O设备间速度不匹配的矛盾。

② 提高它们之间的并行性。

③减少对CPU的中断次数，放宽CPU对中断响应时间的要求。

5.磁盘服务的时间由哪几部分组成？

6.位示图法作用。？

7.掌握磁盘调度算法：先来先服务（FCFS）、最短寻道时间优先(SSTF)、扫描法（SCAN）、循回扫描法（C-SCAN），对于给定的磁盘访问序列，能使用以上调度算法给出磁盘调度顺序，并会计算平均寻道长度。类型题练习：书上例题，书后第11题。

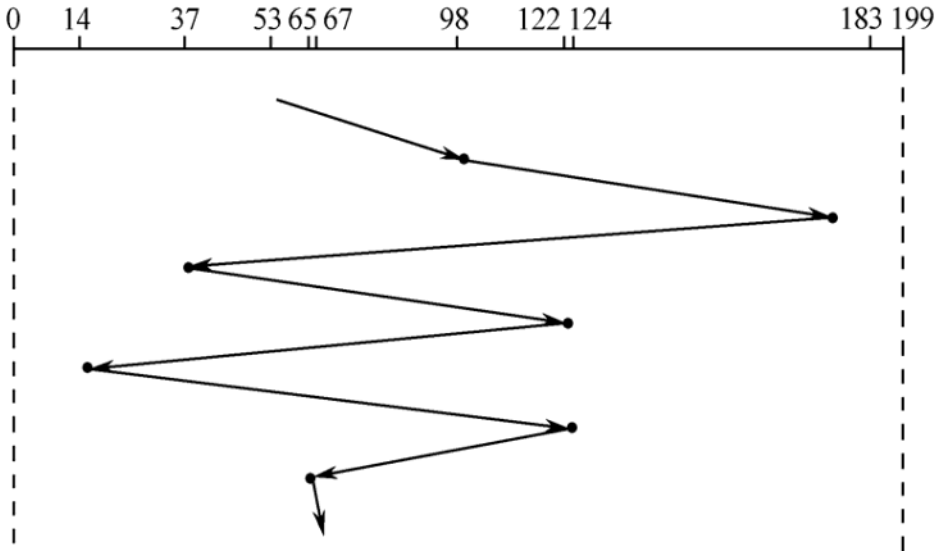
题:

要访问的磁道分别是：

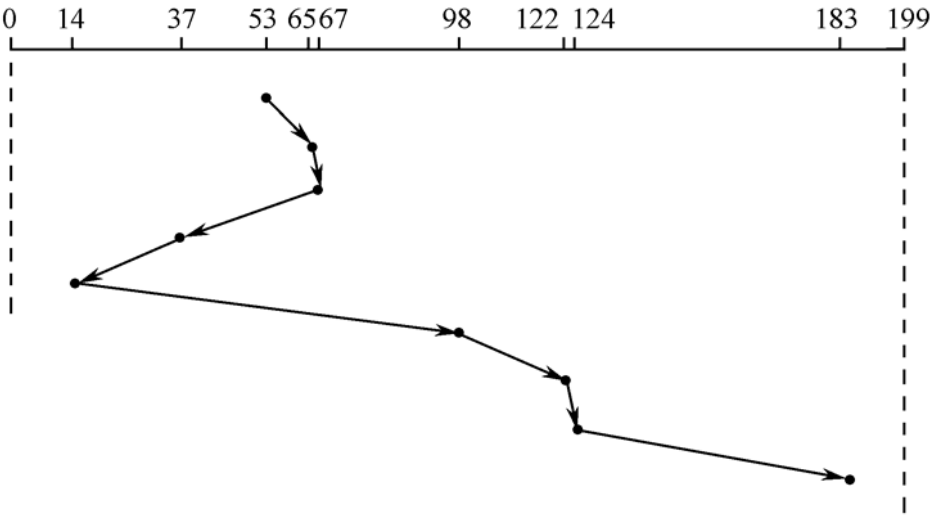
98，183，37，122，14，124，65，67

最早来的请求是访问98道，最后一个是访问67道。设磁头最初在53道上。

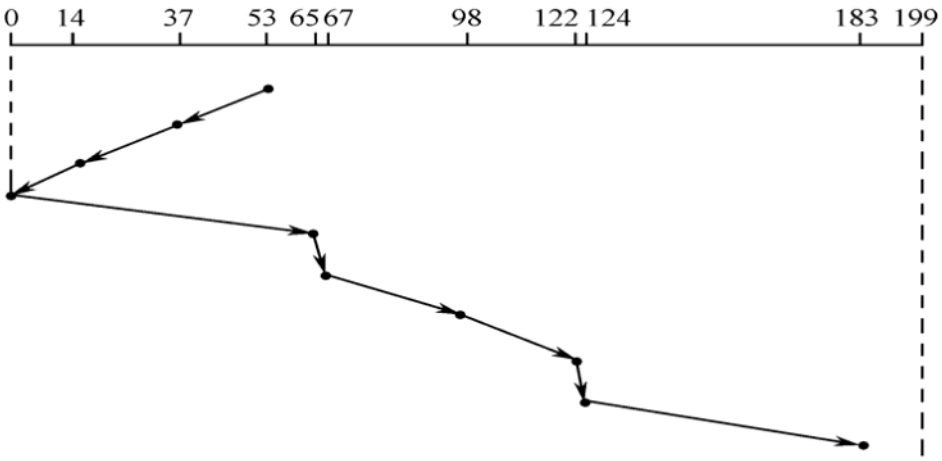
先来先服务法（FCFS）



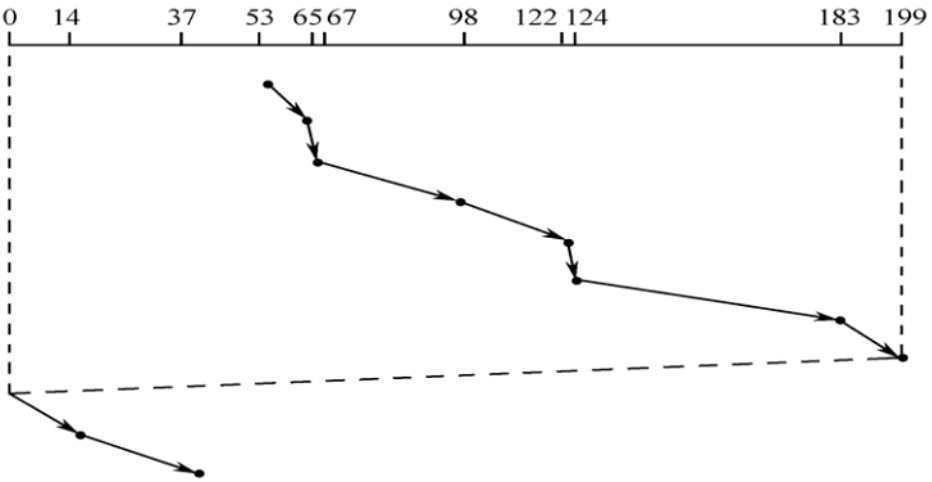
最短寻道时间优先法（SSTF）



扫描法（SCAN）


0 14 37 53 65 67 98 122 124 183 199

巡回扫描法（C-SCAN）


0 14 37 53 65 67 98 122 124 183 199