

## 操作系统考研复习提纲

### 一、操作系统概述

#### (一) 操作系统的概念、特征、功能和提供的服务

- (1) 牢固掌握操作系统的定义：计算机操作系统是方便用户、管理和控制计算机软、硬件资源的系统（或程序集合）；清楚地了解操作系统所处的地位：是裸机之上的第一层软件，是建立其他所有软件的基础。
- (2) 记住操作系统的基本特征：并发、共享、虚拟和异步性。  
理解模拟：并发——“大家一起前进了”；  
共享——“一件东西大家用”；  
虚拟——“一个物理实体变成若干个逻辑事物”；  
异步性——“你走我停”。
- (3) 牢固掌握操作系统的五大主要功能：存储器管理、处理机管理、设备管理、文件管理、用户接口管理。
- (4) 操作系统的服务：1 公共服务类型。程序执行、I/O 操作、文件系统操作、通信、资源分配、统计、保护和差错检测。 2 系统调用。
- (5) 研究操作系统面对用户的服务观点和系统内部的管理观点。即为用户提供方便而安全的工作环境，体现“用户至上”、“服务至上”的原则；内部管理采用中断、通道、串行、并发、并行和本地远地通信等分设的各种管理部门机构；里外协作分工组织，力争达到“多快好省”。

#### (二) 操作系统的发展与分类

- (1) 操作系统的形成和五大类型（批处理操作系统、分时操作系统、实时操作系统、网络操作系统和分布式操作系统）还可以根据用户数分为单用户操作系统和多用户操作系统

#### (2) 计算机操作系统的 4 个发展阶段

手工操作阶段（对应第一代计算机），早期批处理和执行系统（对应第二代计算机），批量操作系统、分时操作系统（对应第三代计算机），个人计算机操作系统、网络操作系统、分布式操作系统（对应现在的第四代计算机）

批量操作系统的主要特征和优缺点：

批量操作系统的主要特征是“批量”，优点是系统吞吐率高，缺点是对用户的响应时间较长，用户不能及时了解和控制自己的程序的运行

分时系统的主要特点：

1 多路性 2 独占性 3 交互性 4 及时

实时操作系统的特点以及与分时系统的区别：

1 系统对外部实时信号必须能及时响应，响应的时间间隔要足以能够控制发出实时信号的那个环境

2 系统要求有高可靠性和安全性，效率则放在第二位

3 系统整体性强，要求所管理的联机设备和资源，必须按一定的时间关系和逻辑关系协调工作

4 实时操作系统没有分时操作系统那样强的交互会话能力，通常不允许用户通过实时终端设备去编写新的程序或修改已有的程序，终端设备只作为执行装置或询问装置。

计算机网络和网络操作系统：

网络的组成：若干主机、一个通信子网和一系列协议

网络操作系统除了具备一般操作系统应具有的功能模块外，还要增加网络通信模块。提供以下能力：1 允许用户访问网络主机中各种资源 2 对用户访问进行控制仅允许授权用户访问特定的资源 3 对远程资源的利用如同本地资源一样 4 提供全网统一的记帐方法 5 联机提供最近网络说明资料 6 提供比单机更可靠的操作

### (三) 操作系统的运行环境

#### 1、硬件环境

##### (1) 处理机的状态和特权指令

简单的系统可只分为管态和用户态，特权指令是指计算机中不允许用户直接使用的指令。特权指令包括下面功能：1 允许和禁止中断 2 在进程之间切换处理机 3 存取用于主存保护的寄存器 4 执行 I/O 操作 5 停止一个处理机的工作

##### (2) 存储器和存储器的组织

两种类型：ROM 和 RAM，三级组织体系：主存储器、辅助存储器和高速缓冲存储器

##### (3) 中断和中断系统。

中断的概念是指某个事件发生时，系统中止现程序的运行，引出处理事件程序对该事件进行处理，处理完毕后返回断点处继续执行。中断系统是指发现中断源而产生中断过程的设备，又称中断装置。

##### (4) 向量中断和中断向量

当中断发生时，由中断源自己引导处理机进入中断服务程序的中断过程称为向量中断。在向量中断中，系统为每个中断类型都设置一个中断向量，用来存放该类型中断的中断服务例行服务程序的入口地址和程序状态字。

#### 2、os 与其它软件的关系

#### 3、用户接口

了解现代操作系统为用户提供的三种使用界面：命令界面、系统调用界面和图形界面。一般用户通过命令、图形方式控制，编程人员通过系统调用方式控制计算机。

### 习题

#### 1、教材中介绍了学习“操作系统”要记住最精髓的两句话，是什么？

答：计算机操作系统使方便用户、管理和控制计算机软硬件资源的系统软件（或程序集合）；操作系统目前有五大类型（批处理、分时、实时、网络和分布式）和五大功能（作业管理、文件管理、存储管理、设备管理和进程管理）。

#### 2、分时系统和实时系统有什么不同？

答：分时系统通用性强，交互性强，及时响应性要求一般（通常数量级为秒）；实时系统往往是专用的，系统与应用很难分离，常常紧密结合在一起，实时系统并不强调资源利用率，而更关心及时响应性（通常数量级为毫秒或微秒）、可靠性等。

#### 3、多用户操作系统和网络操作系统有什么区别？

答：以单机多用户系统为例，说明它与网络操作系统的比较。

	单机多用户系统	网络操作系统
CPU	共用一个或几个	多个处理机
共享资源	共享主存	共享服务器

终端工作	分时方式	网址通信链接
客户端工作	不能独立工作	客户机能独立工作
操作系统	集中管理	客户机有独立的操作系统

4、操作系统负责管理计算机系统的（ ），其中包括处理机、存储器、设备和文件。

A. 程序 B. 文件 C. **资源** D. 进程

5、没有下列（ ）设备计算机无法工作。

A. 硬盘 B. 软盘 C. **内存** D. 打印机

6、操作系统是计算机系统的核心软件。按功能特征的不同，可把操作系统分为[1]、[2]、[3]、网络操作系统和分布式操作系统基本类型。其中[1]的主要目标是提高系统的吞吐率和效率，而[2]是一旦有处理请求和要求处理的数据时，CPU 就应该立即处理该数据并将结果及时送回，例如[4]等。

供选择的答案：

[1][2][3]

A、单用户系统 **B、批处理系统 C、分时系统**

D、微机操作系统 **E、实时系统**

[4]

A、计算机激光照排系统 B、办公自动化系统

C、计算机辅助设计系统 **D、航空订票系统**

7、操作系统是一种（ ）。

A. 应用软件 **B. 系统软件**

C. 通用软件 D. 工具软件

8、在下列性质中，哪一个不是分时系统的特征。（ ）

A. 交互性 B. 多路性 **C. 成批性** D. 独占性

9、实时操作系统追求的目标是（ ）。

A. 高吞吐率 B. 充分利用内存 C. **快速响应** D. 减少系统开销

10、操作系统是为了提高计算机的[1]和方便用户使用计算机而配置的基本软件。它负责管理计算机系统中的[2]，其中包括[3]，[4]，外部设备和系统中的数据。操作系统中的[3]管理部分负责对进程进行管理。操作系统对系统中的数据进行管理的部分通常叫做[5]。

供选择的答案：

[1] A、速度 **B、利用率** C、灵活性 D、兼容性

[2] A、程序 B、功能 **C、资源** D、进程

[3][4] **A、主存储器** B、虚拟存储器 C、运算器

D、控制器 E、微处理器 **F、处理机**

[5] A、数据库系统 **B、文件系统** C、检索系统

D、数据库 E、数据存储系统 F、数据结构

G、数据库管理系统

11、现代操作系统的两个基本特征是（ ） 和资源共享。

A. 多道程序设计 B. 中断处理

**C. 程序的并发执行** D. 实现分时与实时处理

12、以下（ ）项功能不是操作系统具备的主要功能。

- A. 内存管理 B. 中断处理 C. **文档编辑** D. CPU 调度
- 13、批处理系统的主要缺点是（ ）。
- A. CPU 的利用率不高 B. **失去了交互性**
- C. 不具备并行性 D. 以上都不是
- 14、引入多道程序的目的在于（ ）。
- A、充分利用 CPU，减少 CPU 等待时间**
- B. 提高实时响应速度
- C. 有利于代码共享，减少主、辅存信息交换量
- D. 充分利用存储器
- 15、（ ）没有多道程序设计的特点。
- A. **DOS** B. UNIX C. Windows D. OS/2
- 16、DOS 是**磁盘操作系统**的缩写。
- 17、下列四个操作系统中，是分时系统的为（ ）。
- A. CP/M B. MS-DOS
- C. **UNIX** D. WINDOWS NT
- 18、在分时系统中，时间片一定，（ ），响应时间越长。
- A. 内存越多 B. **用户数越多**
- C. 后备队列 D. 用户数越少
- 19、操作系统是一组（ ）。
- A. 文件管理程序 B. 中断处理程序
- C. **资源管理程序** D. 设备管理程序
- 20、（ ）不是操作系统关心的主要问题。
- A. 管理计算机裸机
- B. 设计、提供用户程序与计算机硬件系统的界面
- C. 管理计算机系统资源
- D、高级程序设计语言的编译器**

## 二、 进程管理

### （一） 进程与线程

1. 进程概念
2. 进程的状态与转换
3. 进程控制
4. 进程组织
5. 进程通信

共享存储系统；消息传递系统；管道通信。

### 6. 线程概念与多线程模型

#### （1）程序顺序执行的特点

1 顺序性 2 封闭性 3 可再现性

#### （2）程序并发执行的特点

1 失去程序的封闭性 2 程序与计算不再一一对应 3 程序并发执行的相互制约

## (3) 进程的定义、进程与程序的区别

进程即是一个具有独立功能的程序在某个数据集合的一次运行活动。

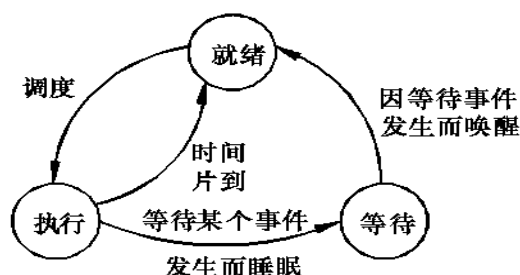
进程的基本特征：动态性，并发性，独立性，异步性，结构性。

## (4) 进程和程序是既有联系又有区别的两个概念：

1 程序是指令的有序集合，本身没有任何运行的含义是一个静态的概念。而进程是程序在处理机上的一次执行过程，是个动态的概念。2 进程是一个能独立运行的单位，能与其他进程并行活动3 进程是竞争计算机系统有限资源的基本单位4 同一程序同时运行若干不同的数据集合上将属于不同的进程。

## (5) 进程的基本状态及其变迁

三种基本状态：就绪、运行、等待，状态变迁是有方向的。



## (6) 进程的描述

用进程控制块（PCB）来描述一个进程。

从结构上来说，进程由一个程序段和一个进程控制块组成。

## (7) 进程控制

所谓进程控制，就是系统使用一些具有特定功能的程序段来创建、撤消进程以及完成进程各状态间的转换，从而达到多进程高效率并发执行和协调、实现资源共享的目的。

进程创建、进程撤消、进程等待和进程唤醒等。由操作系统内核来实现。

把系统态下执行的某些具有特定功能的程序段称为原语。原语可分为两类：一类是机器指令级的，其特点是执行期间不允许中断，正如在物理学中的原子一样，在操作系统中，它是一个不可分割的基本单位。另一类是功能级的，其特点是作为原语的程序段不允许并发执行。

## (8) 进程组织

常用的组织方式有两种：链接方式、索引方式。

## (9) 进程通信：指进程间的信息交换。

按通信内容可以划分为2种：低级通信，高级通信

高级通讯机制类型：

1 共享存储器系统(Shared-Memory System)：相互通讯的进程通过共享数据结构和存储区进行通讯

2 消息传递系统(Message passing System)：在消息传递系统中，进程间的数据交换是以消息(message，在计算机网络中又称报文)为单位。程序员直接利用系统提供的一组通讯命令（原语）来实现通讯。消息缓冲机制，信箱通信方式

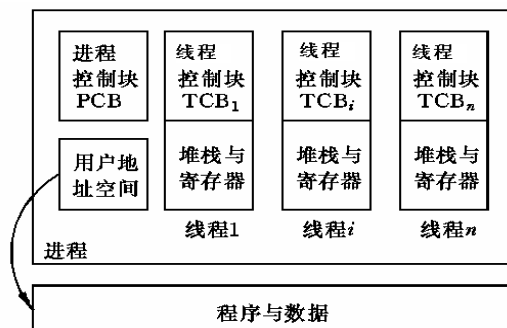
3 管道(pipe)通信系统

管道(pipe)通讯由 UNIX 首创的一种借助文件和文件系统形成的一种通信方式，。由于其有效性，一些系统继 UNIX 之后相继引入了管道技术，如 pc-dos，管道通信将成为进程通讯的一种重要方式。消息缓冲通信机构是以内存缓冲区为基础。

管道是以文件系统为基础。有名管道，无名管道

## (10) 线程概念与多线程模型

线程是进程中的一个实体，是被系统独立调度和分配的基本单位，故又称为轻权（轻型）进程 (Light Weight Process)，它由线程控制表、存储线程上下文的用户栈以及核心栈组成。{传统的进程称为重型进程 (Heavy Weight Process)。}



## 习题

## 1、什么是进程？

答：进程是一个具有一定独立功能的程序关于某个数据集合的一次运行活动。它是操作系统动态执行的基本单元，在传统的操作系统中，进程既是基本的分配单元，也是基本的执行单元。（在 Windows NT 等采用微内核结构的现代操作系统中，进程的功能发生了变化：它只是资源分配的单位，而不再是调度运行的单位，其调度运行的基本单位是线程。）

## 2、什么是线程？它与进程有什么关系？

答：线程是进程中执行运算的最小单位，即处理机调度的基本单位。它与进程的关系是：一个线程只能属于一个进程，而一个进程可以有多个线程；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；处理机分给线程，即真正在处理机上运行的是线程；线程在运行过程中，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。

特别注意的是：传统操作系统中的进程概念与现代操作系统中的进程概念不同——简单说，传统操作系统中进程具有分配资源、调度运行两大功能，而现代操作系统中进程只作为分配资源单位，线程才作为调度运行单位。

## 3、一个进程入睡时其断点落在何处？它被唤醒后从何处继续原来的执行？

答：一个进程入睡是指该进程由于缺乏资源不能占用 CPU，进入等待状态。一个进程由程序、数据集合和进程控制块（PCB）组成。PCB 是进程存在的唯一标志。PCB 中包括如下内容：进程标志号、进程状态（执行/就绪/等待）、进程标志、进程优先数、程序地址、现场保护区（通常被保护的信息有程序计数器、程序状态字、各个工作寄存器等）、通信机构、其他信息等。

处于睡眠状态的进程，它的断点在它的 PCB 中的现场保护区中保护起来。保存程序运行的 CPU 现场，以便在将来的某一时刻恢复并继续原来的运行。它被唤醒后，把它从等待进程队列中摘下，将进程状态改为就绪，然后将它插入到就绪队列中；等它占用 CPU 进入执行状态时，从断点处继续执行。

## 4、进程之间有哪些基本的通信方式？它们分别有什么特点？适用于哪些场合？

答：进程通信根据交换信息量的多少分为高级通信和低级通信。低级通信一般只传送一个或几个字节的信息，以达到控制进程执行速度的作用（如 PV 操作）；高级通信则要传送大量数据，目的不是为了控制进程的执行速度，而是为了交换信息。

高级进程通信方式有很多种，大致可归并为三类：共享存储器、管道文件和消息传递。

共享存储器：在内存中分配一片空间作为共享存储区。需要进行通信的进程把它附加到自己的地址空间中，不需要时则把它取消。

管道文件：它是连接两个命令的一个打开文件。一个命令向该文件中写入数据，为写者；另一个命令从该文件中读出数据，为读者。

消息传递：它以消息为单位在进程间进行数据交换。具体说明见教材“消息缓冲”。

5、程序的**并发**执行是现代操作系统的基本特征之一，为了更好地描述这一特征而引入了**进程**这一概念。

6、进程存在的标志是**进程控制块 PCB**。

7、进程的静态实体由**程序**、**数据集合**和**进程控制块 PCB**三部分组成。

8、用于进程控制的原语主要有**建立原语**、**撤消原语**、**挂起原语**和**激活原语**。

9、进程被创建后，最初处于**就绪**状态，然后经**进程调度程序**选中后进入**执行**状态。

10、进程创建工作主要完成的是创建进程控制块（PCB），并把它挂到**就绪**队列中。

11、如果系统中有  $n$  个进程，则在就绪队列中进程的个数最多为  $n-1$ 。

12、通常，线程的定义是**是进程中执行运算的最小单位**。在现代操作系统中，资源的分配单位是**进程**，而处理机的调度单位是**线程**，一个进程可以有**多个**线程。

13、计算机系统一般都设计有两种运行状态：**用户态**和**核心态**。

14、操作系统中，可以并行工作的基本单位是[1]，[1]也是系统核心调度及资源分配的基本单位，它是由[2]组成的，它与程序的主要区别是[3]。

供选择的答案：

[1]： A. 作业 B. 函数 C. **进程** D. 过程

[2]： A. **程序、数据和 PCB** B. 程序、数据和标识符

C. 程序、标识符和 PCB D. 数据、标识符和 PCB

[3]： A. 程序有状态，而它没有 B. **它有状态，而程序没有**

C. 程序可占用资源，而它不可 D. 它可占用资源，而程序不可

15、下列进程状态的转换中，哪一个是不正确的（ ）。

A. 就绪→执行 B. 执行→就绪

C. **就绪→等待** D. 等待→就绪

16、下列各项步骤中，哪一个不是创建进程所必须的步骤（ ）。

A. 建立一个进程控制块 PCB B. **由 CPU 调度程序为进程调度 CPU**

C. 为进程分配内存等必要的资源 D. 将 PCB 链入进程就绪队列

17、在下列特性中，哪一个不是进程的特性（ ）。

A. 异步性 B. 并发性 C. **静态性** D. 动态性

18、进程就是可与其他程序并行执行的程序段的一次执行过程，它是系统进行资源分配和调度的一个基本单位。进程具有[1]、[2]、调度性、异步性和结构性 5 个基本特征。进程是一次执行过程，具有生命期体现了进程的[1]特征。进程由程序段、[3]、[4]组成，其中[4]是进程在系统中存在的唯一标识。

供选择的答案：

[1][2] A、**动态性** B、静态性 C、并行性

D、**并发性** E、可执行性 F、易用性

[3] A、过程 B、**数据** C、进程标识符 D、函数

[4] A、FCB B、FIFO C、**PCB** D、JCB

19、进程执行时的间断性，决定了进程可能具有多种状态。进程的基本状态有三种，在分时系统中，当一个进程拥有的时间片到时，则该进程即由[1]进入[2]。



如果出现因某种原因使得处理机空闲时，则需要从就绪队列中选择一进程，并将处理机分配给它，此时该进程进入[3]，这个过程是由[4]来完成。

供选择的答案：

[1][2][3] A、**就绪状态** B、静止状态 C、等待状态 D、**执行状态**

[4] A、进程控制程序 B、资源分配程序

C、**进程调度程序** D、处理机分配程序

20、为了描述进程的动态变化过程，采用了一个与进程相联系的（ ）系统，根据它而感知进程的存在。

A. 进程状态字 B. 进程优先数 C. **进程控制块** D. 进程起始地址

21、已经获得除（ ）以外的所有运行所需资源的进程处于就绪状态。

A. 存储器 B. 打印机 C. **CPU** D. 磁盘空间

22、进程是（ ）。

A. 与程序等效的概念 B. **并发环境中程序的执行过程**

C. 一个系统软件 D. 存放在内存中的程序

23、进程具有并发性和（ ）两大重要属性。

A. **动态性** B. 静态性

C. 易用性 D. 封闭性

24、操作系统在控制和管理进程过程中，涉及到（ ）这一重要数据结构，这是进程存在的唯一标志。

A. FCB B. FIFO

C. FDT D. **PCB**

25、并发性是指若干事件在（ ）发生。

A. 同一时刻 B. **同一时间间隔内**

C. 不同时刻 D. 不同时间间隔内

26、顺序程序和并发程序的执行相比，（ ）。

A. 基本相同

B. 有点不同

C. **并发程序执行总体上执行时间快**

D. 顺序程序执行总体上执行时间快

27、在单一处理机上，将执行时间有重叠的几个程序称为（ ）。

A. 顺序程序 B. 多道程序

C. **并发程序** D. 并行程序

28、引入多道程序技术后，处理机的利用率（ ）。

A. 降低了 B. 有所改善

C. **大大提高** D. 没有变化，只是程序的执行方便了

29、在单一处理机上执行程序，多道程序的执行是在（ ）进行的。

A. 同一时刻 B. **同一时间间隔内**

C. 某一固定时刻 D. 某一固定时间间隔内

30、在单处理机系统中，处于运行状态的进程（ ）。

A. **只有一个** B. 可以有多个

C. 不能被挂起 D. 必须在执行完后才能被撤下

31、如果某一进程获得除 CPU 外的所有所需运行资源，经调度，分配给它 CPU，该进程将进入（ ）。

A. 就绪状态 B. **运行状态**



- C. 等待状态 D. 活动状态
- 32、如果某一进程在运行时，因某种原因暂停，此时将脱离运行状态，而进入（ ）。
- A. 自由状态 B. 停止状态  
C. **等待状态** D. 静止状态
- 33、一个进程被唤醒意味着（ ）。
- A. 该进程重新占有了 CPU B. **进程状态变为就绪**  
C. 它的优先权变为最大 D. 其 PCB 移至就绪队列的队首
- 34、进程从运行状态变为等待状态的原因是（ ）。
- A. **输入或输出事件发生** B. 时间片到  
C. 输入或输出事件完成 D. 某个进程被唤醒
- 35、在操作系统中同时存在多个进程，它们（ ）。
- A. 不能共享系统资源  
B. 不能调用同一段程序代码  
C. **可以共享允许共享的系统资源**  
D. 可以共享所有的系统资源
- 36、操作系统中有一组常称为特殊系统调用，它不能被系统中断，在操作系统中称为（ ）。
- A. 初始化程序 B. **原语**  
C. 子程序 D. 控制模块
- 47、进程和程序的本质区别是（ ）。
- A. 存储在内存和外存 B. 顺序和非顺序执行机器指令  
C. 分时使用和独占使用计算机资源 D. **动态和静态特征**

## （二）处理机调度

1. 调度的基本概念
  2. 调度时机、切换与过程
  3. 调度的基本准则
  4. 调度方式
  5. 典型调度算法
- 先来先服务调度算法；短作业（短进程、短线程）优先调度算法；时间片轮转调度算法；优先级调度算法；高响应比优先调度算法；多级反馈队列调度算法。

### 1 一般来说，计算机中的调度可分为 4 级

作业调度：又称为“宏观调度”、“高级调度”。按一定的原则对外存输入井上的大量后备作业进行选择，给选定的作业分配内存、输入输出设备等必要的资源，并建立相应的进程，使得该作业的进程获得竞争处理机的权利。

交换调度：又称为“中级调度”。按照给定的原则和策略，将处于外存交换区中的就绪状态或就绪准等待状态的进程调入内存，或把处于内存就绪或内存等待状态的进程交换到外存交换区。

进程调度：又称为“微观调度”、“低级调度”。按照某种策略和方法选取一个处于就绪状态的进程占用处理机。

线程调度：类似进程调度

### 2 进程调度的时机

(1) 正在执行的进程执行完毕。这时，如果不选择新的就绪进程执行，将浪费处理机资源。

(2) 执行中进程自己调用阻塞原语将自己阻塞起来进入睡眠等状态。

(3) 执行中进程调用了 P 原语操作，从而因资源不足而被阻塞；或调用了 V 原语操作激活了等待资源的进程队列。

(4) 执行中进程提出 I/O 请求后被阻塞。

(5) 在分时系统中时间片已经用完。

(6) 在执行完系统调用等系统程序后返回用户进程时，这时可看作系统进程执行完毕，从而可调度选择一新的用户进程执行。

以上都是在可剥夺方式下引起进程调度的原因。在 CPU 执行方式是可剥夺时，还有

(7) 就绪队列中的某进程的优先级变得高于当前执行进程的优先级，从而也将引发进程调度。

### 3 进程的上下文切换过程：

用户态执行进程 A 代码——进入 OS 核心（通过时钟中断或系统调用）

保存进程 A 的上下文，恢复进程 B 的上下文（CPU 寄存器和一些表格的当前指针）

用户态执行进程 B 代码

注：上下文切换之后，指令和数据快速缓存 cache 通常需要更新，执行速度降低

### 4 调度的基本准则

1) 面向用户的准则：为了满足用户的需求所应遵循的一些准则

(1) 周转时间短（从提交到完成的时间间隔）

$$T_i = T_{ei} - T_{si}$$

作业平均周转时间为（有 n 个作业， $n \geq 1$ ）

$$T = 1/n \sum_{i=1}^n T_i$$

一个作业的周转时间说明了该作业在系统内停留的时间

包含两部分：一是等待时间；二为执行时间

$$T_i (\text{停留时间}) = T_{wi} + T_{ri}$$

带权周转时间  $W_i$ ： $W_i = T_i / T_{ri}$

平均带权周转时间为：

$$W = 1/n \sum_{i=1}^n W_i$$

(2) 响应时间快

分时系统，用户通过键盘提交一个请求开始，直至系统首次产生响应为止的时间。

(3) 截止时间的保证

实时系统

(4) 优先权准则

### 2) 面向系统的准则

(1) 系统的吞吐量高

(2) 处理机利用率好

(3) 各类资源的平衡利用

### 5 调度方式

是指当某一进程正在处理机上执行时，若有某个更为重要或紧迫的进程需要进行处理，此时如何分配处理机。通常有剥夺调度和非剥夺调度（抢占和非抢占）。

## 6 调度算法

### (1) 先来先服务（FCFS）调度算法

将用户作业和就绪进程按提交顺序或变为就绪状态的先后排成队列，并按照先来先服务的方式进行调度处理，是一种最普遍和最简单的方法。它优先考虑在系统中等待时间最长的作业，而不管要求运行时间的长短。

### (2) 最短作业优先法（SJF）

该算法总是优先调度要求运行时间最短的作业

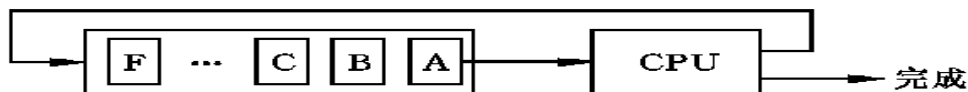
### (3) 最高响应比作业优先算法（HRN）

最高响应比作业优先算法是对 FCFS 方式和 SJF 方式的一种综合平衡响应比  $R$  定义为系统对作业的响应时间与作业要求运行时间的比值

$$\begin{aligned} R &= \text{响应时间} / \text{要求运行时间} \\ &= (\text{作业等待时间} + \text{需运行时间}) / \text{需运行时间} \\ &= 1 + \text{已等待时间} / \text{需运行时间} \\ &= 1 + W/T \end{aligned}$$

### (4) 轮转法（Round Robin）

采用此算法的系统，其程序就绪队列往往按进程到达的时间来排序。进程调度程序总是选择就绪队列中的第一个进程，也就是说按照先来先服务原则调度，但一旦进程占用处理机则仅使用一个时间片。在使用先一个时间片后，进程还没又完成其运行，它必须释放出处理机给下一个就绪的进程，而被抢占的进程返回到就绪队列的末尾重新排队等待在次运行。



### (5) 优先级调度算法

一种常用的进程调度算法是把处理机分配给具有最高优先数的进程（用于实时系统）

在这种算法中，首先考虑的问题是如何确定进程的优先数。一种是静态优先数，另一种是动态优先数。

### (6) 多级反馈队列

设置多个就绪队列，分别赋予不同的优先级，如逐级降低，队列 1 的优先级最高。每个队列执行时间片的长度也不同，规定优先级越低则时间片越长，如逐级加倍

新进程进入内存后，先投入队列 1 的末尾，按 FCFS 算法调度；若按队列 1 一个时间片未能执行完，则降低投入到队列 2 的末尾，同样按 FCFS 算法调度；如此下去，降低到最后的队列，则按“时间片轮转”算法调度直到完成。

仅当较高优先级的队列为空，才调度较低优先级的队列中的进程执行。如果进程执行时有新进程进入较高优先级的队列，则抢先执行新进程，并把被抢先的进程投入原队列的末尾。

## 习题

- 1、单道批处理系统，采用短作业优先调度算法。当第一个作业进入后开始调度，现有 3 个作业，进入和计算时间如图所示，求：（1）完成下表；（2）计算平均周转时间。

序号	进入时间	需计算时间	开始执行时间	完成时间	周转时间
1	9: 00	60 分钟			
2	9: 10	45 分钟			
3	9: 15	25 分钟			

答：平均周转时间 83 分钟 20 秒。（3 分）

序号	进入时间	需计算时间	开始执行时间	完成时间	周转时间
1	9: 00	60 分钟	9: 00	10: 00	60 分钟
2	9: 10	45 分钟	10: 25	11: 10	120 分钟
3	9: 15	25 分钟	10: 00	10: 25	70 分钟

- 2、某多道程序设计系统配有一台处理器和两台输入输出设备  $IO_1$  和  $IO_2$ 。现有优先数从高到低 3 个进程 P1、P2 和 P3 同时存在，他们使用资源的先后顺序和占用时间分别是：

进程 P1:  $IO_2$ (30ms), CPU(10ms),  $IO_1$ (30ms), CPU(10ms),  $IO_2$ (10ms)

进程 P2:  $IO_1$ (20ms), CPU(20ms),  $IO_2$ (40ms)

进程 P3: CPU(30ms),  $IO_1$ (20ms)

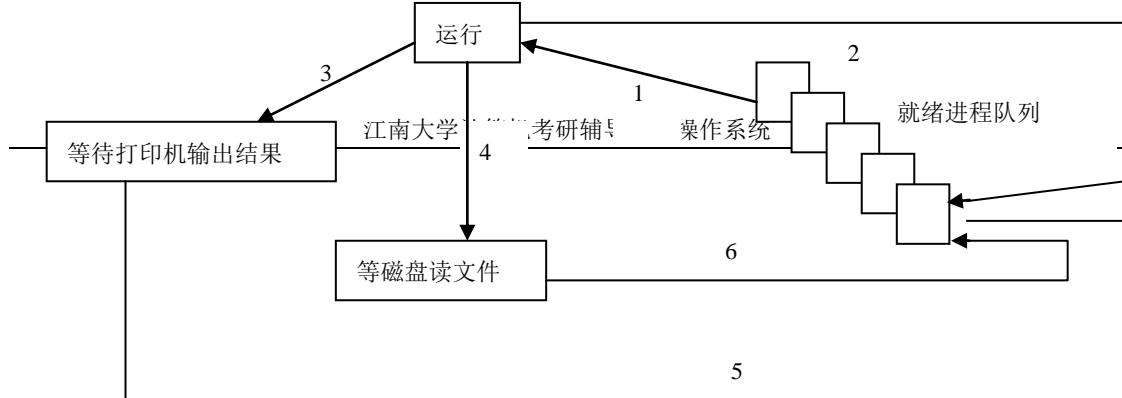
忽略调度等所需时间，回答下列问题：（1）进程调度采用非抢占式的优先数算法时 3 个进程从开始到完成所用的时间分别是多少？CPU 的利用率为多少？（2）进程调度采用可抢占的优先数算法时，3 个进程从开始到全部完成总共用了多少时间？CPU 的利用率为多少？

答：

（1）三个进程从开始到完成所用的时间分别为：进程 P1 110ms；进程 P2 100ms；进程 P3 50ms。处理器的利用率为  $(30+20+10+10)/110 = 63.6\%$ ；

（2）3 个进程从开始到全部完成总共用了 100ms 时间，此时 CPU 的利用率为  $(20+10+10+10+10+10)/100 = 70\%$ 。

- 3、某分时系统中的进程可能出现如图所示状态变化，回答下列问题：（1）根据图示，该系统采用的是什么进程调度策略？（2）写出图示中每一个状态变化的原因。



答：（1）分析图可知，图中 2 变化是时间片到的时候才会发生，又考虑到是分时系统，所以进程调度采用的是时间片轮转算法。（3 分）（2）变化原因如下表所示：

变化	原因
1	进程被选中运行
2	运行满一个时间片（时间片到）（1 分）
3	启动打印机工作后等待（1 分）
4	等待磁盘工作结束（1 分）
5	打印机工作结束（1 分）
6	磁盘传输信息结束（1 分）

4、5 个进程 P1、P2、P3、P4、P5 同时依次进入就绪队列，它们的优先数和需要的处理器时间如图所示：

进程	所需处理器时间	优先数
P1	10	3
P2	1	1
P3	2	3
P4	1	4
P5	5	2

忽略进程调度所花费时间，回答下列问题：

- （1）写出分别采用先来先服务和非抢占式的优先数调度算法选中进程执行的顺序
- （2）分别计算上述两种算法使各进程在就绪队列中的等待时间以及两种算法下的平均等待时间。

答：（1）先来先服务次序：P1、P2、P3、P4、P5；（2分）非抢占式优先数：P1、P4、P3、P5、P2；（2分）（2）各进程在就绪队列中的等待时间如下表：先来先服务算法的平均等待时间：9.6；（2分）非抢占式优先数算法的平均等待时间：10.4。（2分）

进程	处理器时间	优先数	等待时间（先来先服务）	等待时间（非抢占式优先数）
P1	10	3	0	0
P2	1	1	10	18
P3	2	3	11	11
P4	1	4	13	10
P5	5	2	14	13

5、一个具有两道作业的批处理系统中，作业调度采用先来先服务的调度算法，作业一旦被调度运行就不再退出，作业对应的进程调度采用内存中的就绪进程平分 CPU 时间，不计作业对换及其他系统开销。如下表的作业序列。

①计算平均周转时间和平均带权周转时间

作业号	进入时刻	估计运行时间
JOB1	8: 00	25 分钟
JOB2	8: 20	30 分钟
JOB3	8: 25	25 分钟
JOB4	8: 30	15 分钟

答：

提交时刻	运行时间	结束时刻	周转时间	带权周转时间
1	10	15	14	1.4
2	2	4	2	1
3	2	6	3	1.5
4	1	21	17	17
5	5	20	15	3

(1) 周转时间=14+2+3+17+15=51；平均周转时间=51/5=10.2 （4分）

(2) 带权周转时间=1.4+1+1.5+17+3=23.9；平均带权周转时间=23.9/5=4.78 （4分）

- 6、某系统采用不能移动正在主存中作业的可变分区方式管理主存，现有供用户使用的主存空间 100K，系统配有 4 台磁带机（静态分配），采用多道程序设计技术，忽略设备工作时间和系统进行调度所花时间，若对计算时间短的作业进程赋予高优先数，作业调度采用“可抢占的优先数”调度算法，则平均周转时间为多少。作业如下：

序号	入井时间	要求计算时间	需要主存量	申请磁带机数
1	10: 00	25 分钟	15 K	2 台
2	10: 20	30 分钟	60 K	1 台
3	10: 30	10 分钟	50 K	3 台
4	10: 35	20 分钟	10 K	2 台
5	10: 40	15 分钟	30 K	2 台

答：平均周转时间为 43 分钟（4 分）。列表如下：（4 分）

序号	选中次序	选中时间	开始执行时间	完成时间	周转时间
1	1	10: 00	10: 00	10: 25	25 分钟
2	2	10: 20	10: 25	11: 15 (10:35 暂停 20 分)	55 分钟
3	4	11: 15	11: 15	11: 25	55 分钟
4	3	10: 35	10: 35 (抢占)	10: 55	20 分钟
5	5	11: 25	11: 25	11: 40	60 分钟

请分别写出采用“先来先服务调度算法”和“计算时间最短优先算法”选中作业执行顺序以及它们的平均周转时间。

答：两种调度算法相同，平均周转时间为 44 分钟（4 分）。列表如下：（4 分）

作业序号	选中次序	选中时间	开始执行时间	完成时间	周转时间
1	1	10: 00	10: 00	10: 25	25 分钟
2	2	10: 20	10: 25	10: 55	35 分钟
3	5	10: 30	11: 30	11: 40	75 分钟
4	3	10: 35	10: 55	11: 15	40 分钟
5	4	10: 55	11: 15	11: 30	50 分钟



- 7、某单道系统中有 3 个作业正在输入井中等待处理，每个作业估计的计算时间均为 2 小时，那么这 3 个作业的平均周转时间至少为( )小时。
- A. 2                      B. 4                      C. 6                      D. 8
- 8、某分时系统采用时间片轮转调度算法分配处理器，对每个用户的请求需花费 200 毫秒可给出应答。那么，要保证响应时间不超过 4 秒，该系统最多可同时接收( )个终端用户。
- A. 5                      B. 20                      C. 50                      D. 200
- 9、( ) 算法综合考虑了作业的等待时间和计算时间。
- A. 先来先服务                      B. 计算时间短的优先  
C. 均衡调度                      D. 响应比最高者优先
- 10、 为了对紧急进程或重要进程进行调度，调度算法应采用( )。
- A. 先进先出调度算法    **B. 优先数法**  
C. 最短作业优先调度    D. 定时轮转法
- 11、 进程调度的关键问题是选择合理的( )，并恰当地进行代码转换。
- A. 时间片间隔    **B. 调度算法**  
C. CPU 速度    D. 内存空间
- 12、 采用时间片轮转法进行进程调度是为了( )。
- A. 多个终端都能得到系统的及时响应**  
B. 先来先服务  
C. 优先级较高的进程得到及时响应  
D. 需要 CPU 最短的进程先做

### (三) 进程同步

1. 进程同步的基本概念
2. 实现临界区互斥的基本方法  
软件实现方法；硬件实现方法。
3. 信号量
4. 管程
5. 经典同步问题  
生产者-消费者问题；读者-写者问题；哲学家进餐问题。

#### 1、进程同步的基本概念

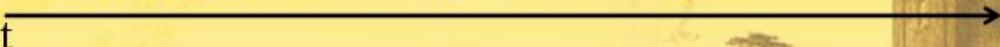
进程同步：它主要源于进程合作，是进程间共同完成一项任务时直接发生相互作用的关系。为进程之间的直接制约关系。在多道环境下，这种进程间在执行次序上的协调是必不可少的。

进程互斥：它主要源于资源共享，是进程之间的间接制约关系。在多道系统中，每次只允许一个进程访问的资源称为临界资源，进程互斥就是保证每次只有一个进程使用临界资源。

#### 2、临界区

每个进程中访问临界资源的那段程序称为临界区（临界资源是一次仅允许一个进程使用的共享资源）。每次只准许一个进程进入临界区，进入后不允许其他进程进入。

## 临界区(Critical Section)问题

- 当两个以上进程共享某个变量时，一次只能允许一个进程对共享变量进行写操作（即互斥地对共享变量进行访问）。
  - 如进程  $P_i: \dots; x := x + 1; \dots$  ( $i=1, 2$ )  
 处理机执行进程  $P_1$ , 有内部通用寄存器  $R_1$ ;  
 处理机执行进程  $P_2$ , 有内部通用寄存器  $R_2$ .
    - 该赋值语句由三个机器内部动作组成:  
 $R := x; R := R + 1; x := R;$
    - $P_1: \dots; [R_1 := x; R_1 := R_1 + 1; x := R_1;] \dots$   
 $P_2: \dots; [R_2 := x; R_2 := R_2 + 1; x := R_2;] \dots$
- t 

临界资源:一次仅允许一个进程使用的资源。

软件方法教材中提到四种算法。（西电的教程）

硬件法采用硬件指令的方式：（1）Test-and-Set（测试并设置指令） （2）Swap（交换指令）

3、信号量——用于表示资源数目或请求使用某一资源的进程个数的整型量。

- $S$  是与临界区内所使用的公用资源有关的信号量。
- $S \geq 0$  可供并发进程使用的资源数
- $S < 0$  正在等待使用临界区的进程数

4、P 原语操作的主要动作

- $S - 1$
- 如果  $S - 1$  以后仍大于等于零，则进程继续进行
- 如果  $S - 1$  以后小于零，则将该进程阻塞以后插入阻塞队列，然后转进程调度

5、V 原语操作的主要动作

- $S + 1$
- 如果相加后结果大于零，则继续进行
- 相加后结果小于等于零，则从该信号的等待队列中唤醒一个等待进程，然后返回原进程继续执行或者转进程调度。

6、管程的定义：管程是关于共享资源的数据结构及一组针对该资源的操作过程所构成的软件模块。1973 年，Hoare 和 Hanson 所提出；其基本思想是把信号量及其操作原语封装在一个对象内部。即：将共享变量以及对共享变量能够进行的所有操作集中在一个模块中。

7、用信号量解题的关键

- 步骤：（1）信号量的设置；（2）给信号量赋初值（常用的互斥和同步信号量值的大小）；（3）P、V 操作安排的位置（其中，P 的顺序不能颠倒，V 的顺序任意）
- 注意区分 1) 公用信号量, 互斥时使用的信号量（二元信号量）：它仅允许取值为“0”与“1”，用作互斥。它联系着一组互斥进程，初值为 1，每个进程均可对

之施加 P、V 操作。2) 私用信号量：一般信号量（资源信号量）：它联系着一组并发进程，但其初值为 0，或为某个正整数  $n$ ，表示资源的数目，主要用于进程同步。只允许拥有它的进程对之施加 P 操作。

### 习题

1、利用 P、V 原语实现进程互斥

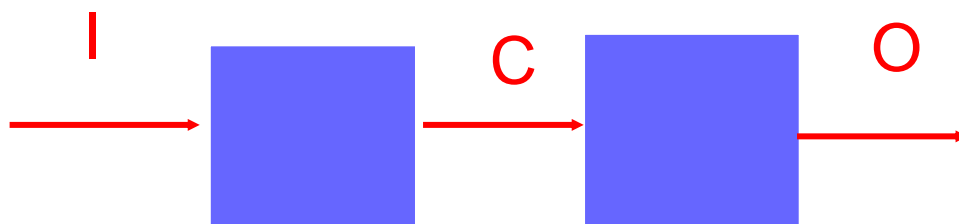
semaphore mutex=1; //用于互斥的信号量

cobegin

Pa: { while(true){ p(mutex) 临界区代码 v(mutex) ... } }	Pb: { while(true){ p(mutex) 临界区代码 v(mutex) ✓... } }
---	--

coend

2、用 P、V 操作描述以下三个进程共享两个缓冲区的问题。



3、利用信号量来描述前趋图关系

Struct semaphore a, b, c, d, e, f, g, h, i, j=0, 0, 0, 0, 0, 0, 0, 0, 0, 0

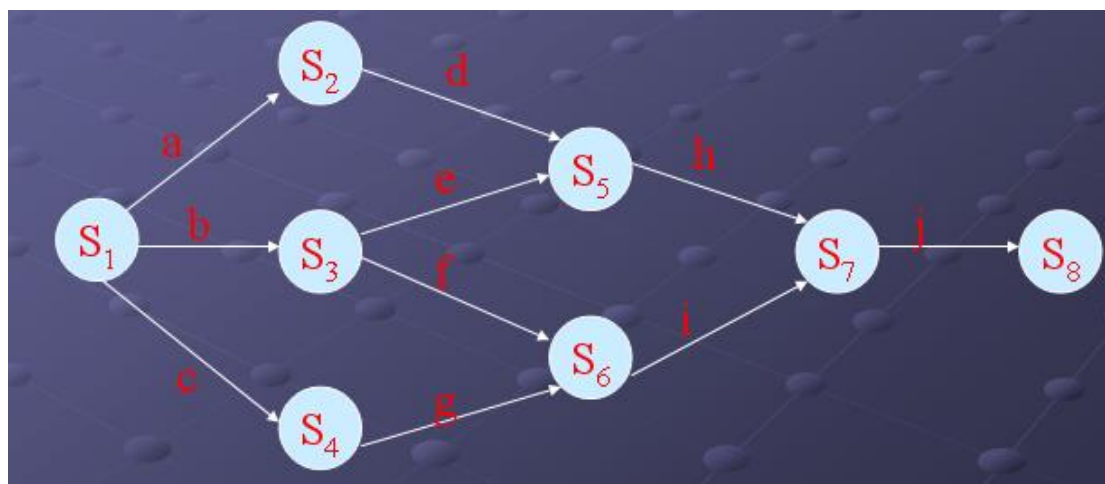
cobegin

```

{S1;V(a);V(b);V(c);}
{P(a);S2;V(d);}
{P(b);S3;V(e);V(f);}
{P(c);S4;V(g);}
{P(d);P(e);S5;V(h);}
{P(f);P(g);S6;V(i)}
{P(h);P(i);S7;V(j);}
{P(j);S8;}

```

coend



#### 4、生产者—消费者问题(the producer-consumer problem)

问题描述：若干进程通过有限的共享缓冲区交换数据。其中，“生产者”进程不断写入，而“消费者”进程不断读出；共享缓冲区共有 N 个；任何时刻只能有一个进程可对共享缓冲区进行操作。

full 是“满”数目，初值为 0，empty 是“空”数目，初值为 N。

mutex 用于访问缓冲区时的互斥，初值是 1

**Semaphore** full, empty;

**Semaphore** mutex;

Full=0;

Empty=n;

Mutex=1;

Cobegin

Producer:

P(empty);

P(mutex);

one unit  $\rightarrow$  buffer;

V(mutex);

V(full);

Consumer:

P(full);

P(mutex);

one unit  $\leftarrow$  buffer;

V(mutex);

V(empty);

Coend;

#### 5、读者—写者问题(the readers-writers problem)

一个文件可能被多个进程共享，为了保证读写的正确性和文件的一致性，系统要求，当有读者进程读文件时，不允许任何写者进程写，但允许多读者同时读；当有写者进程写时，不允许任何其它写者进程写，也不允许任何读者进行读。

● 为了解决读者和写者问题，需设置两个信号量：

- (1) 读互斥信号量 `rmutex`, 用于使读者互斥地访问共享变量 `readcount`, 这里 `readcount` 是记录有多少读者正在读;
- (2) 写互斥信号量 `wmutex`, 用于实现读写互斥和写写互斥地访问共享文件。读者—写者问题进行如下描述:

```
Semaphore rmutex, wmutex=1, 1;
int readcount:=0;
cobegin
vord readeri(vord) (i=1, 2, ..., k)
{
  while(true) {
    p(rmutex);
    if readcount=0 then
      p(wmutex);
      readcount:=readcount+1;
      v(rmutex);
      读文件;
      ...
    p(rmutex);
    readcount:=readcount-1;
    if readcount=0 then
      v(wmutex);
    v(rmutex);
  }
}
vord writerj(vord) (j=1, 2, ..., m)
{
  while(true) {
    p(wmutex);
    写文件;
    v(wmutex);
  }
}
Coend
```

第二类读者写者问题:

写者优先

条件:

- 1) 多个读者可以同时进行读 (N)
- 2) 写者必须互斥 (只允许一个写者写, 也不能读者写者同时进行)
- 3) 写者优先于读者 (一旦有写者, 则后续读者必须等待, 唤醒时优先考虑写者)

- `S, sn: semaphore;`
- `S:=1;` //用于写者与其他读者/写者互斥的访问共享数据
- `sn:=n;` //该信号量初始值设为 `n`, 表示最多允许 `n` 个读者进程同时进行读操作
- `Cobegin`
- `Reader I (I:=1, 2, ..., n);`
- `{`

```

● P(s); //读者、写者互斥
● P(sn);
● V(s); // 释放读写互斥信号量，允许其它读、写进程访问资源
● Read file f;
● V(sn);
● }
● Writer I (I=1,2,...,n);
● {
● P(s);
● For I:=1 to n do p(sn); //禁止新读者，并等待已进入的读者退出
● Write file f;
● For I:=1 to n do v(sn); // 恢复允许 rmutex 值为 n
● V(s);
● }
● Coend;

```

#### 6、哲学家进餐问题

问题描述：（由Dijkstra首先提出并解决）5个哲学家围绕一张圆桌而坐，桌子上放着5支筷子，每两个哲学家之间放一支；哲学家的动作包括思考和进餐，进餐时需要同时拿起他左边和右边的两支筷子，思考时则同时将两支筷子放回原处。如何保证哲学家们的动作有序进行？如：不出现相邻者同时要求进餐；不出现有人永远拿不到筷子；

```

● Var S:array[0..4] of :semaphore;
● S[I]:=1;
● Repeat
● P(s[I]);
● P(s[(I+1) % 5]);
● Eat;
● v (s[I]);
● v (s[(I+1) %5]);
● Until false;

```

一种正确解法：

```

semaphore fork[5];
for (int i=0;i<5;i++)
    fork[i]= 1;

cobegin
process philosopher_i( )    {    /*i=0, 1, 2, 3, 4*/
    while(true) {
        think( );
    If (i<(i+1)%5 )
        {P(fork[i]);
        P(fork[(i+1)%5] ); }
    Else
        {P(fork [(i+1)%5] ;
        P(fork [i]) ; }

```

```

        eat( );
        V(fork[i]);
        V(fork[(i+ 1) % 5]);
    }
}
coend

```

## 7、图书馆阅览室问题

问题描述：假定阅览室最多可同时容纳 100 个人阅读，读者进入时，必须在阅览室门口的一个登记表上登记，内容包括姓名、座号等，离开时要撤掉登记内容。用 P、V 操作描述读者进程的同步算法。

```

#define CHAIRS 100
Semaphore mutex=1;
Int Readers=0;
Void Reader( )
{while (1) {
P(mutex);
If (readers< CHAIRS) {
    登记;
    Readers++;
    V(mutex);
    阅读;
    P(mutex);
    撤销登记;
    Readers--;
    V(mutex);
} else
V(mutex);
}
}

```

### 另一种解法：

答：读者的动作有两个，一是填表进入阅览室，这时要考虑阅览室里是否有座位；一是读者阅读完毕，离开阅览室，这时的操作要考虑阅览室里是否有读者。读者在阅览室读书时，由于没有引起资源的变动，不算动作变化。

算法的信号量有三个：seats——表示阅览室是否有座位（初值为 100，代表阅览室的空座位数）；readers——表示阅览室内的读者数，初值为 0；用于互斥的 mutex，初值为 1。

读者进入阅览室的动作描述 getin:

```

while(TRUE) {
P (seats); /*没有座位则离开*/
P (mutex) /*进入临界区*/
    填写登记表;
    进入阅览室读书;
    V (mutex) /*离开临界区*/
    V (readers)
}

```



```
}

```

读者离开阅览室的动作描述 getout:

```
while(TRUE) {
P (readers) /*阅览室是否有人读书*/
P (mutex) /*进入临界区*/
消掉登记;
离开阅览室;
V (mutex) /*离开临界区*/
V (seats) /*释放一个座位资源*/
}
```

**提问：这两种解法有什么不同。**

#### 8、理发师问题(Dijkstra 1965)

问题描述：一个理发店由一个有几张椅子的等候室和一个放有一张理发椅的理发室组成。若没有要理发的顾客，则理发师就去睡觉；若一顾客走进理发店且所有的椅子都被占用了，则该顾客就离开理发店；若理发师正在为人理发，则该顾客就找一张空椅子坐下等待；若理发师在睡觉，则顾客就唤醒他，设计一个协调理发师和顾客的程序。

```
#define CHAIRS 5
Semaphore
customers=0;barbers=0;mutex=1;
Int waiting=0;
Void barber()
{while (1){
P(customers);
P(mutex);
Waiting--;
V(barbers);
V(mutex);
Cut_hair();
}
}
Void customer()
{p(mutex);
If (waiting<CHAIRS){
Waiting++;
V(customers);
V(mutex);
P(barbers);
Get_haicut();
}else{
V(mutex);
}
```

```
}

```

### 9、吃水果问题

问题描述：桌上有一只盘子，每次只能放一个水果，爸爸专向盘中放苹果，妈妈专向盘中放桔子，儿子专等吃盘里的桔子，女儿专等吃盘里的苹果。只要盘子空，则爸爸或妈妈可向盘中放水果，仅当盘中有自己需要的水果时，儿子或女儿可从中取出，请给出四人之间的同步关系，并用 PV 操作实现四人正确活动的程序。

### 10、司机—售票员问题

❖ 设公共汽车上，司机和售票员的活动分别是：

司机：	售票员：
启动车辆	上下乘客
正常行车	关车门
到站停车	售票
	开车门
	上下乘客

在汽车不断到站，停车，行驶过程中，这两个活动的同步关系。

```

● semaphore s1=0;//是否启动车辆，0 表示不能启动
● s2=0;//是否可以开车门，0 表示不可以开
● Main()
● {cobegin
●     driver();
●     busman();
●     coend;}
● Driver()
● {while (1)
●     {p(s1);
●     启动车辆;
●     正常行车;
●     到站停车;
●     V(s2);
●     }
● }
● busman()
● {while (1)
●     {
●     关车门;
●     v(s1);
●     售票;
●     p(s2);
●     开车门;
●     上下乘客;
●     }
● }

```

- 11、 假设有一个成品仓库，总共能存放 8 台成品。生产者进程生产产品放入仓库，消费者进程从仓库中取出成品消费。为了防止积压，仓库满的时候就停止生产。由于仓库搬运设备只有一套，故成品的存入和取出只能分别执行，使用 PV 操作来实现方案。

```

begin
mutex , full , empty :semaphore; (1 分)
mutex:= 1 ; (1 分)
empty:= 8 ; (1 分)
full:= 0 ; (1 分)
cobegin
    processor producer (3 分)
    begin
        生产一个成品;
        P(empty);
        P(mutex);
        将成品存入仓库;
        V(mutex);
        V(full);
    end
    processor consumer (3 分)
    begin
        P(full);
        P(mutex);
        将成品从仓库取出;
        V(mutex);
        V(empty);
        消费成品;
    end
coend;
end.

```

- 12、生产围棋的工人不小心将相等数量的黑子和白子混合装在一个盒子里，现要用自动分拣系统把黑子和白子分开，该系统由两个并发执行的进程 PA 和 PB 组成，系统功能如下：PA 专拣黑子，PB 专拣白子；每次只拣一个子，当一个进程拣子时，不许另一个进程去拣子；当一个进程拣子后，必须让另一个进程拣子。

试回答：（1）两个进程间是同步还是互斥？（2）写出用 PV 定义的信号量及其初值。（3）写出 PV 管理的两个并发进程的程序。

答：（1）同步关系。（2 分）（2）两个信号量：S1 允许拣黑子；S2 允许拣白子；初值为：0、1 或 1、0；（2 分）（3）程序如下：

```

begin
    S1, S2 : semaphore;
    S1:=1;
    S2:=0;
    Cobegin

```

```

process PA          (3 分)
begin
L1: P(S1);
    拣黑子;
    V(S2);
    Goto L1;
end
process PB          (3 分)
begin
L2 : P(S2);
    拣白子;
    V(S1);
    goto L2;
end;
coend ;
end.

```

13、试说明进程互斥、同步和通信三者之间的关系。

答：进程的同步与互斥是指进程在推进时的相互制约关系。在多道程序系统中，由于资源共享与进程合作，这种进程间的制约称为可能。为了保证进程的正确运行以及相互合作的进程之间交换信息，需要进程之间的通信。

进程之间的制约关系体现为：进程的同步和互斥。

进程同步：它主要源于进程合作，是进程间共同完成一项任务时直接发生相互作用的关系。为进程之间的直接制约关系。在多道环境下，这种进程间在执行次序上的协调是必不可少的。

进程互斥：它主要源于资源共享，是进程之间的间接制约关系。在多道系统中，每次只允许一个进程访问的资源称为临界资源，进程互斥就是保证每次只有一个进程使用临界资源。

进程通信是指进程间的信息交换。PV 操作作为进程的同步与互斥工具因信息交换量少，效率太低，称为低级通信。而高级通信则以较高的效率传送大批数据。

14、 每执行一次 P 操作，信号量的数值 S 减 1。若  $S \geq 0$ ，则该进程 **继续执行**；若  $S < 0$ ，则该进程 **等待**。

15、 每执行一次 V 操作，信号量的数值 S 加 1。若  $S > 0$ ，则该进程继续执行；否则，从对应的 **等待**队列中移出一个进程并将 **就绪**状态赋予该进程。

16、 利用信号量实现进程的 **互斥**，应为临界区设置一个信号量 mutex，其初值为 1，表示该资源尚未使用，临界区应置于  $P(mutex)$  和  $V(mutex)$  原语之间。

17、 在操作系统中，对信号量 S 的 P 原语操作定义中，使进程进入相应等待队列等待的条件是（ ）。

A.  $S > 0$  B.  $S = 0$  C.  $S < 0$  D.  $S \neq 0$

18、 在一段时间内，只允许一个进程访问的资源称为（ ）。

A. 共享资源 B. 临界区  
C. **临界资源** D. 共享区

19、 并发性是指若干事件在（ ）发生。

A. 同一时刻 B. **同一时间间隔内**

- C. 不同时刻 D. 不同时间间隔内
- 20、 顺序程序和并发程序的执行相比，（ ）。
- A. 基本相同  
B. 有点不同  
C. **并发程序执行总体上执行时间快**  
D. 顺序程序执行总体上执行时间快
- 21、 在单一处理机上，将执行时间有重叠的几个程序称为（ ）。
- A. 顺序程序 B. 多道程序  
C. **并发程序** D. 并行程序
- 22、 在单一处理机上执行程序，多道程序的执行是在（ ）进行的。
- A. 同一时刻 B. **同一时间间隔内**  
C. 某一固定时刻 D. 某一固定时间间隔内
- 23、 进程间的基本关系为（ ）。
- A. 相互独立与相互制约 B. **同步与互斥**  
C. 并行执行与资源共享 D. 信息传递与信息缓冲
- 24、 进程间的同步与互斥，分别表示了各进程间的（ ）。
- A. 相互独立与相互制约 B. **协调与竞争**  
C. 不同状态 D. 动态性与独立性
- 25、 两个进程合作完成一个任务，在并发执行中，一个进程要等待其合作伙伴发来信息，或者建立某个条件后再向前执行，这种关系是进程间的（ ）关系。
- A. **同步** B. 互斥  
C. 竞争 D. 合作

26、 如何理解进程的顺序性与并发性？

答：进程的顺序性与并发性：

1、顺序性

顺序性包括两层含义：（1）内部顺序性，对于一个进程来说，它的所有指令是按序执行的；（2）外部顺序性，对于多个进程来说，所有进程是依次执行的。

例如，假如有 P1 和 P2 两个进程，其活动分别为：

P1 活动：a1 a2 a3 a4

P2 活动：b1 b2 b3 b4

顺序执行时，有如下两种情形：

情形 1：a1 a2 a3 a4 b1 b2 b3 b4

情形 2：b1 b2 b3 b4 a1 a2 a3 a4

2、并发性

并发性包括如下两层含义：（1）内部顺序性，对于一个进程来说，它的所有指令是按序执行的；（2）外部并发性，对于多个进程来说，所有进程是交叉执行的。

例如，对于上面 P1 和 P2 两个进程来说，并发执行有许多情形，如：

情形 1：a1 b1 b2 a2 a3 b3 a4 b4

情形 2：b1 b2 a1 a2 a3 b3 b4 a4

并发进程在其执行过程中，出现哪种交叉情形是不可预知的，这就是并发进程的不确定性，操作系统应当保证：无论出现何种交叉情形，每个进程运行的结果都应当是唯一的，正确的。

27、 什么是进程的同步与互斥？

答：进程的同步与互斥是指进程在推进时的相互制约关系。在多道程序系统中，由于进程合作与资源共享，这种进程间的制约称为可能。我们把前者称为进程同步，后者称为进程互斥。

进程同步是进程间共同完成一项任务时直接发生相互作用的关系。为进程之间的直接制约关系。在多道环境下，这种进程间在执行次序上的协调是必不可少的。同步进程之间的关系如同接力赛跑中的运动员，或生产流水线的每一道工序。

进程互斥是进程之间的间接制约关系。在多道系统中，每次只允许一个进程访问的资源称为临界资源，进程互斥就是保证每次只有一个进程使用临界资源。互斥进程之间的关系如同汽车在交叉路口争用车道，篮球比赛中双方争抢篮板球。

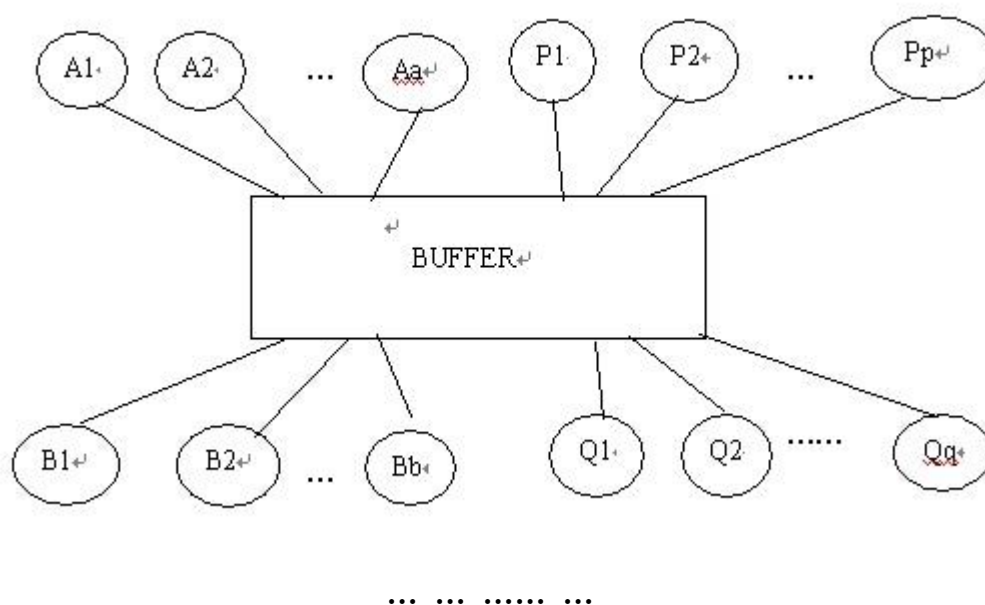
## 28、设系统中有两类生产者消费者进程

第一类：生产者 A 生产的数据类型为 AB，供对应的消费者 B 消费；

第二类：生产者 P 生产的数据类型为 PQ，供对应的消费者 Q 消费；

消费者 B 只能消费 AB 型数据，不能消费 PQ 型数据，同样，消费者 Q 只能消费 PQ 型数据，不能消费 AB 型数据。

系统仅使用唯一的一个公共缓冲区 BUFFER，用于 A→B 和 P→Q 的数据传送，并且该缓冲区仅能容纳一个 AB 型或 PQ 型数据。系统中有多个 A 类和多个 P 类生产者进程，以及多个 B 类和多个 Q 类消费者进程，如图所示：



用 P.V 操作描述出生产者 A 消费者 B，以及，生产者 P，消费者 Q 的算法。并阐述所设计的算法是如何达到系统要求的。

要求：由于系统者两类生产者-消费者进程都异常活跃，唯一的公共缓冲区，可能会成为瓶颈，因此，算法要避免类似下述情况发生，即：当某一生产者 A 将生产出的数据放入

buffer 后，由于所有的消费者 B 进程都忙于消费先前得到的数据而无法立即将数据从 buffer 中取出，从而导致 buffer 长期被占用，使得一个空闲的消费者 Q 进程无法收到生产者 P 已经生产出的数据。

#### （四）死锁

1. 死锁的概念
2. 死锁处理策略
3. 死锁预防
4. 死锁避免

系统安全状态：银行家算法。

5. 死锁检测和解除

1、死锁 Deadlock：是计算机系统中多道程序并发执行时，两个或两个以上的进程由于竞争资源而造成的一种互相等待的现象（僵局），如无外力作用，这些进程将永远不能再向前推进。

陷入死锁状态的进程称为死锁进程，所占用的资源或者需要它们进行某种合作的其它进程就会相继陷入死锁，最终可能导致整个系统处于瘫痪状态。

产生死锁的四个必要条件：

- (1)互斥条件:进程访问的是临界资源，那个资源一次只能被一个进程所使用。
- (2)不剥夺条件:一个资源仅能被占有它的进程所释放，而不能被其他进程剥夺。
- (3)部分分配:(请求和保持条件)一个进程在请求新的资源的同时，保持对某些资源的占有。
- (4)环路等待条件:存在一个进程的环路链，链中每一个进程占用有着某个或某些资源，又在等待链中的另一个进程占有的资源。

#### 2、死锁处理策略

- 其 1 鸵鸟算法（置之不理）
- 其 2 预防死锁
- 其 3 避免死锁
- 其 4 检测和解除死锁

#### 3、死锁预防

根据产生死锁的四个必要条件，只要使用其中之一不能成立，死锁就不会出现。但必要条件（1）是由设备的固有特性所决定的，不仅不能改变，相反还应加以保证，因此实际上只有三种方法。

预防死锁是一种较易实现的方法，已被广泛使用，但由于所施加的限制条件往往太严格，可能导致系统资源利用率和系统吞吐量降低。

##### （1）防止部分分配（摒弃请求和保持条件）

系统要求任一进程必须预先申请它所需的全部资源，而且仅当该进程的全部资源要求能得到满足时，系统才能给予一次性分配，然后启动该进程运行，但是在分配时只要由一种资源不满足，系统就不会给进程分配资源。进程运行期间，不会再请求新的资源，所以，再分配就不会发生（摒弃了部分分配）。

特点：资源严重浪费

进程延迟进行

##### （2）防止“不剥夺”条件的出现

- 其 采用的策略：一个已经保持了某些资源的进程，当它再提出新的资源要求而不能立



即得到满足时，必须释放它已经保持的所有资源，待以后需要时再重新申请。

- 其 实现比较复杂，且要付出很大代价；此外，还因为反复地申请和释放资源，而使程序的执行无限地推迟，延长了周转时间，增加了系统的开销，降低了系统吞吐量。（例如打印机打印的结果不连续）

### （3）防止“环路等待”条件的出现

采用资源顺序使用法，基本思想是：把系统中所有资源类型线性排队，并按递增规则赋予每类资源以唯一的编号。例如输入机=1，打印机=2，磁带机=3，磁盘=4 等等。进程申请资源时，必须严格按资源编号的递增顺序进行，否则系统不予分配。

优点：资源利用率和系统吞吐量与另两种方法相比有较明显的改善。

缺点：

- ①为系统中各种类型的资源所分配的序号必须相对稳定，这就限制了新设备类型的增加
- ②作业实际使用资源的顺序与系统规定的顺序不同而造成资源的浪费；

### 4、死锁避免

避免死锁与预防死锁的区别在于，预防死锁是设法至少破坏产生死锁的必要条件之一，严格地防止死锁的出现。

避免死锁，它是在进程请求分配资源时，采用银行家算法等防止系统进入不安全状态。

- 其 在资源的动态分配的过程中，使用某种方法去防止系统进入不安全状态，从而避免死锁的发生。
- 其 系统状态：（1）安全状态：指系统能按照某种顺序如 $\langle P_1, P_2, \dots, P_n \rangle$ （称为 $\langle P_1, P_2, \dots, P_n \rangle$ 序列为安全序列），为每个进程分配所需的资源，直至最大需求，使得每个进程都能顺利完成。（2）非安全状态：即在某个时刻系统中不存在一个安全序列，则称系统处于不安全状态或非安全状态。

例：假定系统有三个进程  $P_1$ 、 $P_2$ 、 $P_3$ ，共有 12 台磁带机。进程  $P_1$  总共要求 10 台磁带机， $P_2$  和  $P_3$  分别要求 4 台和 9 台。设在  $T_0$  时刻，进程  $P_1$ 、 $P_2$  和  $P_3$  已经获得 5 台、2 台和 2 台，还有 3 台空闲没有分配。

进	最 大 需	已 分	可
$P_1$	1	5	3
$P_2$	4	2	
$P_3$	9	2	

$T_0$  时刻系统时安全的。这时存在一个安全序列 $\langle P_2, P_1, P_3 \rangle$

- 其 虽然并非所有不安全状态都是死锁状态，但当系统进入不安全状态后，便有可能进入死锁状态；反之只要系统处于安全状态，系统便可避免进入死锁状态。因此，避免死锁的实质是如何使系统不进入不安全状态。

银行家算法：银行家算法是最有代表性的避免死锁算法，是 Dijkstra 提出的银行家算法。这是由于该算法能用于银行系统现金贷款的发放而得名。

设  $Request_i$  是进程  $P_i$  的请求向量，如果进程  $P_i$  需要  $K$  个  $R_j$  类资源，当  $P_i$  发出资源请求后，系统按下述步骤进行检查：

- 1 如果  $Request_i \leq Need_i$ ，则转向步骤 2；否则认为出错。（因为它所需要的资源数已超过它所宣布的最大值。

2 如果  $Request_i \leq Available$ , 则转向步骤 3; 否则, 表示系统中尚无足够的资源,  $P_i$  必须等待

3 系统试探把要求的资源分配给进程  $P_i$ , 并修改下面数据结构中的数值:

$Available := Available - Request_i$ ;

$Allocation := Allocation + Request_i$ ;

$Need_i := Need_i - Request_i$ ;

4 系统执行安全性算法, 检查此次资源分配后, 系统是否处于安全状态。若安全, 正式将资源分配给进程  $P_i$ , 以完成本次分配; 否则, 将试探分配作废, 恢复原来的资源分配状态, 让进程  $P_i$  等待。

系统所执行的安全性算法可描述如下:

1 设置两个向量

①工作向量  $Work$ . 它表示系统可提供给进程继续运行所需要的各类资源的数目, 它含有  $m$  个元素, 执行安全算法开始时,  $Work := Available$ 。

② $Finish$ . 它表示系统是否有足够的资源分配给进程, 使之运行完成。开始时先做  $Finish[i] := false$ ; 当有足够的资源分配给进程时, 令  $Finish[i] := true$ 。

2 从进程集中找到一个能满足下述条件的进程: ① $Finish[i] = false$ ; ② $Need_i \leq Work$ . 如找到, 执行步骤 3; 否则执行步骤 4。

3 当进程  $P_i$  获得资源后, 可顺利执行, 直至完成, 并释放出分配给它的资源, 故执行:

$Work := Work + Allocation$ ;

$Finish[i] := true$ ;

Goto step2;

4 如果所有进程的  $Finish[i] = true$ , 则表示系统处于安全状态; 否则, 系统处于不安全状态。

## 5、死锁检测和解除

死锁的检测和恢复技术是指定期启动一个软件检测系统的状态, 若发现有死锁存在, 则采取措施恢复之。

### (1) 死锁的检测

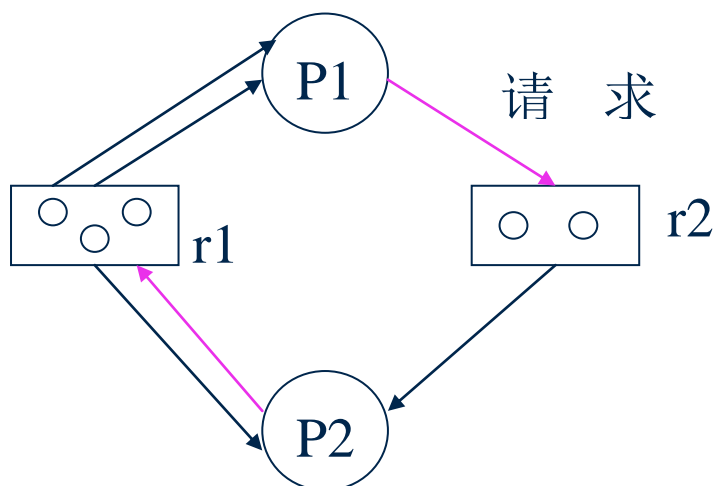
检查死锁的办法就是由软件检查系统中由进程和资源构成的有向图是否构成一个或多个环路, 若是, 则存在死锁, 否则不存在。

死锁的检测: 实质是确定是否存在环路等待现象, 一旦发现这种环路便认定死锁存在, 并识别出该环路所涉及的有关进程, 以供系统采取适当的措施来解除死锁。最常用的是一种基于资源分配图 RAG 和死锁定理的检测死锁算法。

资源分配图 (RAG)

系统死锁可用资源分配图来描述, 该图是由一组结点  $N$  和一组边  $E$  所组成的一对偶  $G = (N, E)$ 。(用圆圈代表一个进程, 用方框代表一类资源, 由于一种类型的资源可能有多个, 可用方框中的一个点代表一类资源中的一个资源)

几个概念: 请求边, 分配边



❏ 封锁进程：是指某个进程由于请求了超过了系统中现有的未分配资源数目的资源，而被系统封锁的进程。

❏ 非封锁进程：即没有被系统封锁的进程  
资源分配图的化简方法：假设某个 RAG 中存在一个进程  $P_i$ ，此刻  $P_i$  是非封锁进程，那么可以进行如下化简：当  $P_i$  有请求边时，首先将其请求边变成分配边（即满足  $P_i$  的资源请求），而一旦  $P_i$  的所有资源请求都得到满足， $P_i$  就能在有限的时间内运行结束，并释放其所占用的全部资源，此时  $P_i$  只有分配边，删去这些分配边（实际上相当于消去了  $P_i$  的所有请求边和分配边），使  $P_i$  成为孤立结点。（反复进行）

死锁定理：系统中某个时刻  $S$  为死锁状态的充要条件是  $S$  时刻系统的资源分配图是不可完全简化的。

在经过一系列的简化后，若能消去图中的所有边，使所有的进程都成为孤立结点，则称该图是可完全简化的；反之的是不可完全简化的。

❏ 死锁的恢复。是与检测死锁相配套的一种措施，用于将进程从死锁状态下解脱出来。常用的方法有：

❏ 1 撤消进程：最简单撤消进程的方法是使全部死锁的进程夭折掉；另一方法是按照某种顺序逐个地撤消进程，直至有足够的资源可用，死锁状态消除为止

❏ 2 挂起进程（剥夺资源）。使用挂起/激活机构挂起一些进程，剥夺它们的资源以解除死锁，待条件满足时，再激活进程。目前挂起法比较受到重视。

习题：

1、（北大 95）一个 OS 有 20 个进程，竞争使用 65 个同类资源，申请方式是逐个进行的，一旦某个进程获得它所需要的全部资源，则立即归还所有资源。每个进程最多使用三个资源。若仅考虑这类资源，该系统有无可能产生死锁，为什么？

2、在某系统中，三个进程共享四台同类型的设备资源，这些资源一次只能一台地为进程服务和释放，每个进程最多需要二台设备资源，试问在系统中是否会产生死锁？

3、某系统中有  $n$  个进程和  $m$  台打印机，系统约定：打印机只能一台一台地申请、一台一台地释放，每个进程需要同时使用的打印机台数不超过  $m$ 。如果  $n$  个进程同时使用打印机的总数小于  $m+n$ ，试讨论，该系统可能发生死锁吗？并简述理由。

4、一台计算机有 8 台磁带机。它们由  $N$  个进程竞争使用，每个进程可能需要 3 台磁带机。请问  $N$  为多少时，系统无死锁危险，并说明原因。

5、银行家算法是一种（ ）算法。

A. 死锁解除 B. 死锁避免 C. 死锁预防 D. 死锁检测

- 6、某系统有 3 个进程，都需要同类资源 4 个，试问该系统不会发生死锁的最少资源数。
- 7、当进程数大于资源数时，进程竞争资源时一定会发生死锁吗？
- 8、资源的按序分配策略可以破坏造成死锁的什么条件？
- 9、产生死锁的四个必要条件是什么？
- 10、假定系统中有五个进程 {P0、P1、P2、P3、P4} 和三种类型的资源 {A, B, C}，每一种资源的数量分别为 10、5、7，在 T0 时刻的资源分配情况如图
- 请找出该表中 T0 时刻以后存在的安全序列（至少 2 种）

资源情况 进程	Max			Allocation			Need			Available		
	A	B	C	A	B	C	A	B	C	A	B	C
P0	7	5	3	0	1	0	7	4	3	3	3	2
P1	3	2	2	2	0	0	1	2	2			
P2	4	4	3	3	0	2	6	0	0			
P3	2	2	2	2	1	1	0	1	1			
	4	3	3	0	0	2	4	3	1			

- 11、若系统有 A、B、C、D 四类资源供 5 个进程共享，进程对资源的需求和分配情况如下表所示：

进程	已占资源数				最大需求数			
	A	B	C	D	A	B	C	D
P1	0	0	1	2	0	0	1	3
P2	1	0	0	0	1	7	5	0
P3	1	3	5	4	2	3	5	6
P4	0	6	3	2	0	6	5	2
P5	0	0	1	4	0	6	5	6

现在系统还剩资源 A 类 1 个，B 类 5 个，C 类 2 个和 D 类 0 个，请按银行家算法回答下列问题：（1）现在系统是否处于安全状态？给出解释。（2）如果现在进程 P2 提出需要（0，4，2，0）个资源的请求，系统能否去满足它，为什么？

答：（1）系统处于安全状态。

将系统剩余资源如下次序分配：先为 P4 分配，后剩余（1，5，0，0），归还后剩余（1，11，5，2）；分配给 P2 和 P3，后剩余（0，4，0，0），归还后剩余（3，14，10，6）；最后分配给 P5，后剩余（3，8，6，4），归还后剩余（3，14，11，10）；P1 只需要 1 个资源，故每个进程都能在有限时间内得到所需全部资源，且 P1 归还后系统能收回全部资源。（4 分）

（2）当 P2 提出资源要求时，按银行家算法不能满足它。因为当前剩余资源数（1，5，2，0）预分配给 P2 后，找不到一个安全序列，不符合算法的执行条件。（4 分）

12、某系统中有 10 台打印机，有 3 个进程 P1、P2、P3 分别需要 8 台、7 台、4 台，若 P1、P2、P3 已经申请到 4 台、2 台和 2 台，问：按银行家算法能安全分配吗？请说明分配的过程。

答：系统是安全的，（4 分）因为系统已经分配了 8 台，剩余的 2 台可以先分配给 P3，这样，P3 得到所需的全部资源，能执行到结束并归还所占用的 4 台，再将其分配给 P1，使 P1 能执行结束并归还 7 台，最后将其中的 5 台给 P2，则 P2 也能执行到结束并归还资源。这样，每个进程都能得到所需的全部资源，系统是安全的。（4 分）

13、当前某系统有同类资源 10 个，进程 P、Q、R 所需资源总数分别为 8、4、9，它们向系统申请及原的次序和数量如表所示：

次序	进程	申请量	次序	进程	申请量
1	R	2	6	Q	2
2	P	4	7	R	1
3	Q	2	8	P	2
4	P	2	9	R	1
5	R	1			

回答：（1）系统采用银行家算法分配资源，请写出系统完成第 6 次分配各进程占用的资源量。

（2）在以后的各次申请中，哪次的申请要求可先得到满足？

答：完成第 6 次分配后各个进程的状态以及所占用的资源量为：

P：等待，所占资源量 4 个； Q：就绪或运行，所占资源量 4 个； R：等待，所占资源量 2 个。进程 Q 已经得到全部资源，能运行到结束并归还资源。（4 分）

若进程 Q 归还所有资源，则系统剩余资源 4 个，可满足 P 所需资源数，但不能满足 R 的尚需资源数，因而，第 8 次 P 的申请将先得到满足。（4 分）

14、系统出现死锁的原因是（ ）。

- A. 计算机系统发生了重大故障
- B. 有多个封锁的进程同时存在
- C. 若干进程因竞争资源而无休止地等待着，不释放已占有的资源
- D. 资源数大大少于进程数，或进程同时申请的资源数大大超过资源总数

15、两个进程争夺同一个资源（ ）。

- A. 一定死锁 B. 不一定死锁
- C. 不会死锁 D. 以上说法都不对

16、解决死锁的途径是（ ）。

- A. 立即关机排除故障
- B. 立即关机再重新开机
- C. 不要共享资源，增加独占资源
- D. 设计预防死锁方法，运行检测并恢复

17、进程 P1 使用资源情况：申请资源 S1，申请资源 S2，释放资源 S1；进程 P2 使用资源情况：申请资源 S2，申请资源 S1，释放资源 S2，系统并发执行进程 P1，P2，系统将（ ）。

- A. 必定产生死锁 B. 可能产生死锁
- C. 会产生死锁 D. 无法确定是否会产生死锁

### 三、内存管理

## （一） 内存管理基础

### 1. 内存管理概念

**程序装入与链接；逻辑地址与物理地址空间；内存保护。**

程序的装入就是把程序装入内存空间。

#### ● 采用三种方式

- （1）绝对装入方式：是由装入程序根据装入模块中的地址，将程序和数据装入主存。
- （2）可重定位方式：是由装入程序根据主存当前的实际使用情况，将装入模块装入到主存适当的地方。
- （3）动态运行时装入方式：动态运行时的装入程序，在把装入模块装入主存后，并不立即把装入模块中的相对地址转换为绝对地址，而是把这种地址转换推迟到程序要真正执行时才进行。

链接程序的功能是将经过编译或汇编后所得到的一组目标模块以及它们所需要的库函数，装配成一个完整的装入模块。

#### ● 实现链接的方法有三种

- 静态链接：事先进行链接，以后不再拆开的链接方式。
- 装入时动态链接：用户源程序经编译后所得到的目标模块，是在装入主存时，边装入边链接的。
- 运行时动态链接：可将某些目标模块的链接，推迟到执行时才进行。

逻辑地址（相对地址）——程序用来访问信息所用的一系列的地址单元。

物理地址（绝对地址）——主存中一系列存储物理单元。

逻辑地址空间是逻辑地址的集合。物理地址空间是物理地址的集合。一个是虚的概念，一个是实的物体。

重定位：当一个地址装入与其地址空间不一致的存储空间中，就得要地址变换。也就是说将虚地址映射为内存地址，把这种作法叫做地址重定位。

#### （1）静态地址重定位

在装入一个作业时，把作业中的指令地址全部转换为绝对地址（地址转换工作是在作业执行前集中一次完成的）在作业执行过程中就无须再进行地址转换工作。

#### （2）动态地址重定位

动态地址重定位是在程序执行过程中，在 CPU 访问内存之前，将要访问的程序或数据地址转换成内存地址。动态重定位依靠硬件地址变换机构完成。

地址重定位机构需要一个或多个基地址寄存器 BR 和一个或多个程序虚地址寄存器 VR。指令或数据的内存地址 MA 与虚地址的关系为： $MA = (BR) + (VR)$ ，其中，(x)表示寄存器 x 中的内容。

常用的内存信息保护方法有硬件法、软件法和软硬件结合三种。

1 上下界保护法：是一种常用的硬件保护法。

2 保护键法。

3 界限寄存器与 CPU 的用户态或核心态工作方式相结合的保护方式。

处理器在执行指令时，要检查其逻辑地址是否小于界限寄存器的值，

——若小于，则与重定位寄存器中的基址相加，产生物理地址，到主存中去执行。

——否则，产生一个“地址越界”中断信号，由操作系统进行处理，以达到存储保护的的目的。

### 2. 交换与覆盖

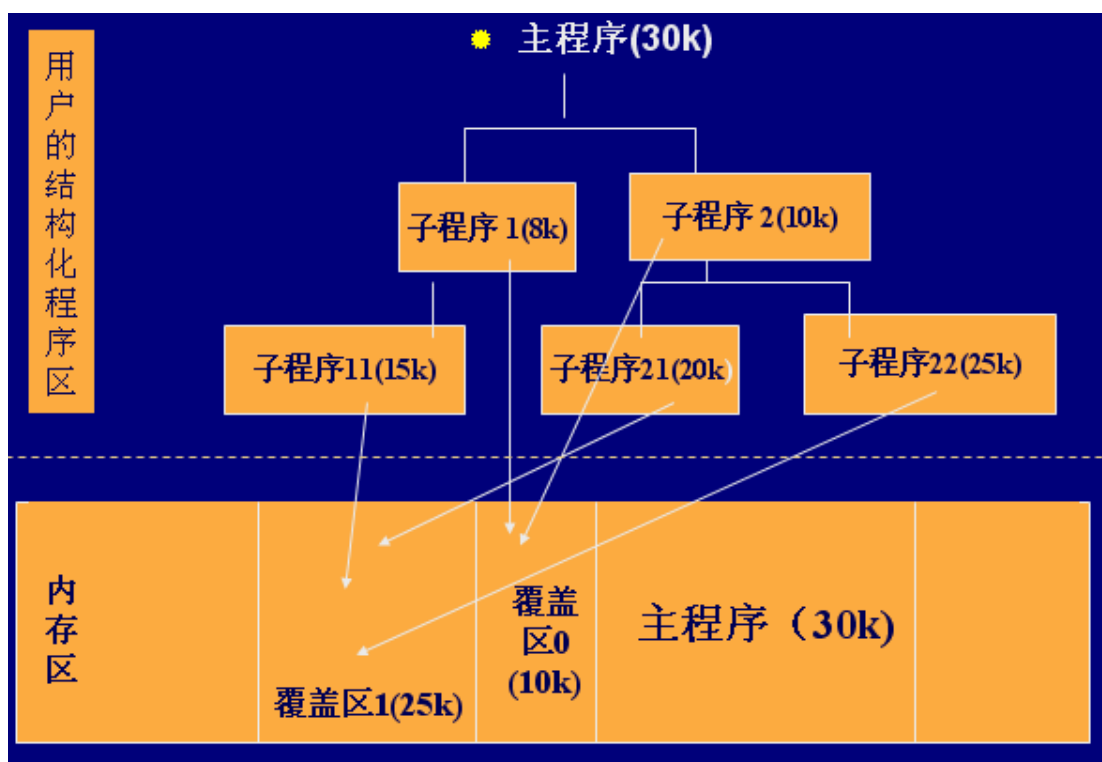
所谓交换，就是系统根据需把主存中暂时不运行的某个(或某些)作业部分或全部移到外存，而把外存中的某个(或某些)作业移到相应的主存区，并使其投入运行。

交换技术被广泛地运用于早期的小型分时系统的存贮管理中。其目的，一方面解决主存容量不够大的矛盾，一方面使各分时用户能保证合理的响应时间。

交换的时机通常在以下情况发生：①作业的进程用完时间片或等待输入输出；②作业要求扩充存贮而得不到满足时。

交换技术也是利用外存来逻辑地扩充主存。它的主要特点是打破了一个程序一旦进入主存便一直运行到结束的限制。

所谓覆盖，是指同一主存区可以被不同的程序段重复使用。通常一个作业由若干个功能上相互独立的程序段组成，作业在一次运行时，也只用其中的几段，利用这样一个事实，我们就可以让那些不会同时执行的程序段共用同一个主存区。因此，我们把可以相互覆盖的程序段叫做覆盖。而把可共享的主存区叫做覆盖区。



覆盖技术的主要特点是打破了必须将一个作业的全部信息装入主存后才能运行的限制。在一定程度上解决了小主存运行大作业的矛盾。

### 3. 连续分配管理方式：单一连续分配；分区分配。

单一连续分配：

- 这是最早出现的一种存储管理方式。
- 在主存中仅驻留一道程序，整个用户区为一用户独占。当用户作业空间大于用户区时，该作业不能装入。
- 这种分配方式仅能用于单用户、单任务的操作系统中，不能用于多用户系统和单用户多任务系统中。
- 1. 主存空间的分配

采用这种存储管理方式时，主存分为两个分区（系统区 和用户区）。

其分配过程是：首先，从作业队列中取出队首作业；判断作业的大小是否大于



用户区的大小，若大于则作业不能装入，否则，可以把作业装入用户区。它一次只能装入一个作业。

- 2. 主存空间的回收

作业一旦进入主存，就要等到它结束后才能释放主存，再装入第二个作业即可。

- 3. 地址转换

它采用静态分配方式。

处理器设置两个寄存器：界限寄存器和重定位寄存器。界限寄存器用来存放主存用户区的长度，重定位寄存器用来存放用户区的起始地址。

地址转换过程是：CPU 获得的逻辑地址首先与界限寄存器的值比较，若大于界限寄存器的值，产生“地址越界”中断信号，由相应的中断处理程序处理；若不大于界限寄存器的值，就与重定位寄存器中的基址相加，得到物理地址，对应于主存中的一个存储单元。

- 4. 存储保护

处理器在执行指令时，要检查其逻辑地址是否小于界限寄存器的值，

若小于，则与重定位寄存器中的基址相加，产生物理地址，到主存中去执行。

否则，产生一个“地址越界”中断信号，由操作系统进行处理，以达到存储保护的

目的。

- 5. 管理特点

(1) 管理简单。它把主存分为两个区，用户区一次只能装入一个完整的作业，且占用一个连续的存储空间。它需要很少的软硬件支持，且便于用户了解和使用。

(2) 在主存中的作业不必考虑移动的问题，并且主存的回收不需要任何操作。

(3) 资源利用率低。不管用户区有多大，它一次只能装入一个作业，这样造成了存储空间的浪费，使系统整体资源利用率不高。

(4) 这种分配方式不支持虚拟存储器的实现。

固定分区分配：

1. 原理：

把主存中可分配的用户区域预先划分成若干个固定大小的区域，每一个区域称为一个分区，每个分区中可以装入一个作业，一个作业也只能装入一个分区中，这样可以装入多个作业，使它们并发执行。当有一个空闲分区时，便可从外存的后备队列中，选择一个适当大小的作业装入该分区；当该作业运行完时，又可从后备队列中选择另一个作业装入该分区。

2. 分配和回收

(1) 采用的数据结构：为了记录各个分区的基本情况和使用情况，方便主存空间的分配与回收操作，设置了一张分区分配表。

分区分配表的内容包括：分区序号、起始地址、大小、状态。

状态栏的值为“0”表示分区空闲，可以装入作业；当装入作业后，其值改为作业名，表示这个分区被该作业占有。

区号	分区长度	起始地址	状态
1	8K	20K	已分配
2	32K	28K	已分配

3	64K	60K	已分配
4	132K	124K	已分配

(2) 主存空间的分配：在作业分配之前，根据主存分区的划分，在分区分配表填入每个分区的始址、大小，在状态栏中一律填入“0”，表示该分区可用，当作业装入时，填入作业名。

(3) 主存空间的回收：当作业运行结束时，根据作业名到分区分配表中进行检查，从状态栏的记录可知该作业占用的分区，把该分区的状态标志置成“0”，表示该分区就空闲了，可以用来装入新的作业。

3. 地址转换：采用静态重定位方式。

- 处理器设置两个寄存器：下限寄存器和上限寄存器。下限寄存器用来存放分区低地址，即起始地址；上限寄存器用来存放分区的高地址，即末址。
- 地址转换过程
  - CPU 获得的逻辑地址首先与下限寄存器的值相加，产生物理地址；
  - 然后与上限寄存器的值比较，若大于上限寄存器的值，产生“地址越界”中断信号，由相应的中断处理程序处理；若不大于界限寄存器的值，得到物理地址就是合法地址，它对应于主存中的一个存储单元。

4. 存储保护

- 系统设置了一对寄存器，称为“下限寄存器”和“上限寄存器”记录当前在 CPU 中运行作业在主存储器中的下限和上限地址。
- 当处理机执行该作业的指令时必须核对表达式“下限地址 $\leq$ 绝对地址 $\leq$ 上限地址”是否成立。
  - 若成立，就执行该指令，否则就产生“地址越界”中断事件，停止执行该指令。
- 运行的作业在让出处理器时，调度程序选择另一个可运行的作业，同时修改当前运行作业的分区分号和下限、上限寄存器的内容，以保证处理器能控制作业在所在的分区内正常运行。

5. 管理特点

- (1) 一个作业只能装入一个分区，不能装入两个或多个相邻的分区。一个分区只能装入一个作业，当分区大小不能满足作业的要求时，该作业暂时不能装入。
- (2) 通过对“分区分配表”的改写，来实现主存的分配与回收。作业在执行时，不会改变存储区域，所以采用静态地址重定位方式。此方法易于实现，系统开销小。
- (3) 当分区较大作业较小时，仍然浪费许多主存空间。并且分区总数固定，限制了并发执行的作业数目。

可变分区存储管理方式：

1. 基本原理：

可变分区存储管理方式是在作业要求装入主存时，根据作业的大小动态地划分分区，使分区的大小正好适应作业的要求。各分区的大小是不定的，主存中分区的数目也是不定的。

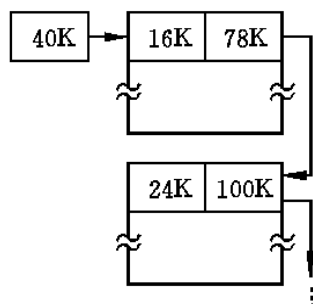
2. 主存空间的分配与回收

(1) 采用的数据结构

为了实现分区分配，系统中必须配置相应的数据结构，用来记录主存的使用情况，包括空闲分区的情况和使用分区的情况，为作业分配主存空间提供依据。为此，设置了两个表，即已分分区表和空闲分区表或自由链。

区号	分区长度	起始地址
1	16K	40K
3	24K	78K
5	9K	100K

(a) 可用表



(b) 自由链

作业(进程)号	请求长度
P <sub>1</sub>	13K
P <sub>2</sub>	20K
⋮	

(c) 请求表

## (2) 可变分区时的分配与回收

- ① 对于请求表中要求内存长度,从可用表或自由链中找出合适的空闲区。
- ② 分配空闲区之后,更新可用表或自由链。
- ③ 进程或作业释放内存资源时,和相邻的空间区进行链合并,更新可用表或自由链。

## (3) 分配算法

- ① 最先适应法: 按照某种次序依次检查各个空闲区, 把第一个找到能容纳申请要求的内存区分配给申请者。空闲区按地址顺序从小到大登记。
- ② 最佳适应算法: 按照某一从大到小次序依次检查所有的空闲区, 把能容纳申请要求的一个最接近尺寸且大于或等于作业大小的分区给申请的作业。
- ③ 最坏适应算法: 要求空闲区按其大小递减的顺序组成空闲区可用表或自由链当用户作业或进程申请一个空闲区时, 先检查空闲区可用表或自由链的第一个空闲可用区的大小是否大于或等于所要求的内存长度, 若可用表或自由链的第一个项所有空闲区长度小于申请, 则失败, 否则从空闲区可用表或自由链分配相应的空间给用户, 然后修改和调整空闲区可用表或自由链。

## 3. 地址转换

采用动态重定位方式装入作业。它需要设置硬件地址转换机构: 两个专用寄存器, 即基址寄存器和限长寄存器, 以及一些加法、比较电路。

## 4. 存储保护

- 系统设置了一对寄存器, 称为“基址寄存器”和“限长寄存器”记录当前在 CPU 中运行作业在主存中所占分区的始址和末址。
- 当处理机执行该作业的指令时必须核对表达式“始址 $\leq$ 绝对地址 $\leq$ 末址”是否成立。若成立, 就执行该指令, 否则就产生“地址越界”中断事件, 停止执行该指令。

## 5. 管理特点

- 一是, 分区的长度不是预先固定的, 而是按作业的实际需求来划分的。分区的个数也不是预先确定的, 而是由装入的作业数决定的。
- 二是, 分区的大小由作业的大小来定, 提高了主存的使用效率。
- 三是, 在主存分配过程中, 会产生许多主存“碎片”。主存“碎片”是指小的无法使用的主存空间。使主存空间仍有一定的浪费。

## 4. 非连续分配管理方式

分页管理方式; 分段管理方式; 段页式管理方式。

分页管理方式:

### 1. 基本原理:

- 将用户作业的地址空间分成若干个大小相同的区域, 称为页面或页, 并为每个页从

“0”开始编号；

- 相应地，主存空间也分成与页大小相同的若干个存储块，或称为物理块或页框(frame)，并且采用同样的方式为它们进行编号，从0开始：0块，1块，...，n-1块。
- 程序的逻辑地址由页号和页内地址组成，页号的长度决定了分页的多少，页内地址的长度决定了页面的大小。
- 在为作业分配主存时，以块为单位将作业中的若干页分别装入多个可以不相邻接的块中。作业执行时根据逻辑地址中的页号找到它所在的块号，再确定当前指令要访问的主存的物理地址。
- 它的地址转换属于动态重定位。

## 2. 主存空间的分配与回收

### (1) 采用的数据结构

页表：由页号和页面号组成，指出逻辑地址中页号与主存中块号（页面号）的对应关系。

页号——作业地址空间的页序号

页面号——内存空间的页面序号

页 号	页面号

请求表（主存分配表）：它记录主存中各作业的作业名、页表始址和页表长度，页表长度为页表中的最大序号。整个系统设置一个主存分配表。

进程号	请求页面数	页表地址	页表长度	状态
1	20	1024	20	已分
2	34	1044	34	已分
3	18	1078	18	已分
4	21	.....	.....	未分

存储页面表（位示图）：一个系统只有一张存储页面表。它指出内存各页面是否被分配,以及用来分配页面的总数。

19	18	3	2	1	0
		...			
		...			
		...			

### (2) 主存空间的分配

- 主存空间的分配过程为：
  - 首先，系统要初始化位示图，即把位示图中的标志位全部置为“0”，空闲块数置为主存的块数。
  - 其次，在进行主存分配时，从作业队列中取出队首作业，计算该作业的页数，然后，与位示图中的空闲块数比较，若不能满足作业的要求，则作业不能装

入，显示主存不足的信息，把该作业放到队尾或删除该作业；若能满足作业的空间要求，则为该作业建立页表，并根据位示图中主存块的状态标志，找出标志为“0”的那些位，置上占用标志“1”，根据该位在位示图中的字号和位号，利用下列公式可以计算出该页所对应的块号。

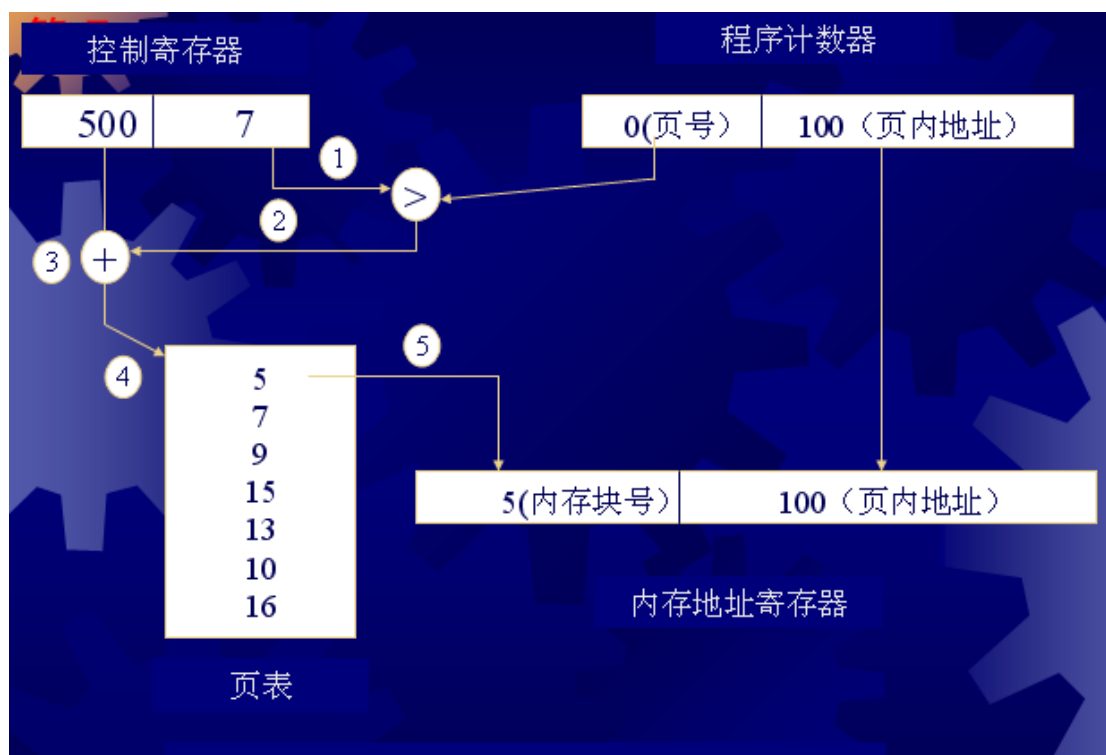
- 把作业页装入对应的主存块，并在页表中填入对应的块号。直到所有作业页全部装入。
- 最后，修改位示图中空闲块数，即原有空闲块数减去本次占用的块数(页数)，并在主存分配表中增加一条记录，登记该作业的作业名、页表始址和页表的长度。

### (3) 主存空间的回收

- 当一个作业执行结束，则应收回该作业所占用的主存块。
- 根据主存分配表中的记录，取出该作业的页表。从该作业的页表中取出每一个归还的块号计算出该块在位示图中的位置，将占用标志位清为“0”；
- 最后，把归还的块数加入到空闲块数中，删除该作业的页表，并把分区分配表中该作业的记录删除。

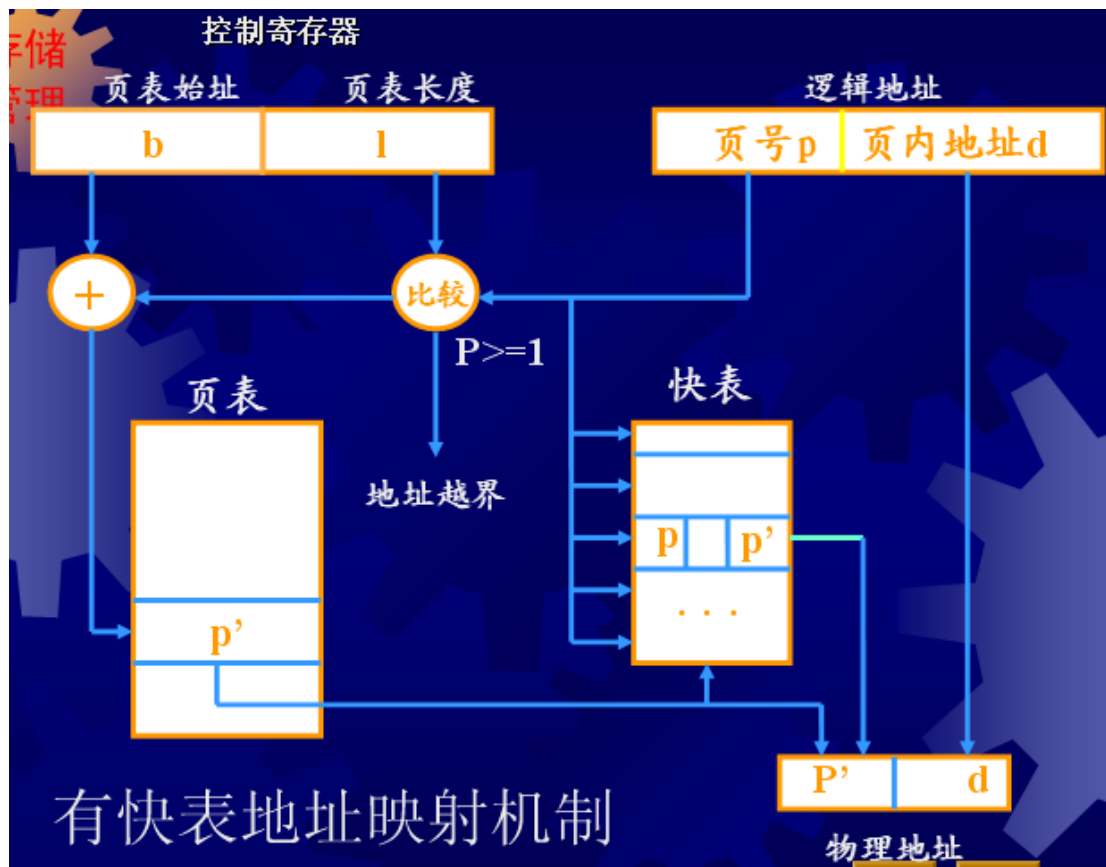
### 3. 地址转换：页式存储管理采用动态重定位方式装入作业。

- 页表的功能可以由一组专门的寄存器来实现。
- 在执行检索之前，先将页号与页表长度进行比较，如果页号大于页表长度，则表示本次所访问的地址已超越作业的地址空间。这一错误将被系统发现并产生“地址越界”中断。若未出现越界错误，则将页表始址与页号相加，便得到该表项在页表中的位置，可以从中得到该页的物理块号，将块号装入物理地址寄存器中。与此同时，再将逻辑地址寄存器中的页内地址，直接送入物理地址寄存器的块内地址字段中。经过硬件机制，把物理地址寄存器中的块号和块内地址，转换成物理地址。这样，便完成了从逻辑地址到物理地址的变换。



### 4. 快表和联想存储器

从上述地址转换过程可以看出，执行一次访内操作至少要访问主存两次。一次访页表，以确定所取数据或指令的物理地址；另一次是根据地址取数据或指令。这样就把程序的执行速度降低一倍。为了提高存取速度，通常设置一个专用的高速缓冲寄存器组，用来存放页表的一部分。我们把存放在高速缓冲寄存器中的页表叫快表，这个高速缓冲寄存器又叫联想存储器。



## 5. 管理特点

- 一是，可以使程序和数据存放在不连续的主存空间，不必象可变分区管理那样通过增加系统开销来解决“碎片”问题，它有效地解决了“碎片”多的问题。
- 二是，通过位示图和页表来记录主存的使用情况和每个作业的分配情况，当主存很大，并且作业也很大时，位示图和页表都有可能占用较大的存储空间。
- 另外，它要求有相应的硬件支持，从而增加了系统成本，也增加了系统开销。如需要地址变换机构、快表等。并且它仍然存在不可利用的空间，如最后一页的部分空间没有放满等。它还要求页的大小固定，不能随程序的大小而变，对程序的共享和使用造成了许多困难。

## 段式存储管理：

### 1. 基本原理

- 在段式存储管理方式中，作业的地址空间被划分为若干个段，每个段定义了一组逻辑信息。
- 它以段为单位分配主存，每段分配一个连续的主存空间，但各段之间不要求连续。
- 供用户使用的逻辑地址为段号+段内地址。在装入作业时，用一张段表记录每个分段在主存中的起始地址和长度。若装入作业的某段信息找不到足够大的空闲区，可采用移动技术，合并分散的空闲区。
- 主存的分配与回收类似于动态分区分配，采用动态重定位。



- 在段式存储管理方式中，作业的地址空间被划分为若干个段，每个段定义了一组逻辑信息。
- 它以段为单位分配主存，每段分配一个连续的主存空间，但各段之间不要求连续。
- 供用户使用的逻辑地址为段号+段内地址。在装入作业时，用一张段表记录每个分段在主存中的起始地址和长度。若装入作业的某段信息找不到足够大的空闲区，可采用移动技术，合并分散的空闲区。
- 主存的分配与回收类似于动态分区分配，采用动态重定位。

段号	段内地址
----	------

## 2. 主存空间的分配与回收

### (1) 采用的数据结构

- 为了记录主存中空闲分区的始址和大小，以及作业中每个段分配主存的情况，在段式存储管理方式下，设置了空闲分区表、段表和主存分配表（请求表）。
  - 空闲分区表：用于记录主存中空闲区的序号、起始地址和大小，整个主存设置一个。
  - 段表：系统为每个作业建一个段表，用于记录每个作业的每个段在主存中所占分区的起始地址和大小。
  - 主存分配表（请求表）。整个系统设置一个主存分配表，用于记录主存中各作业的作业名、段表始址和段表长度，段表长度为段表中的最大序号。

### (2) 主存空间的分配

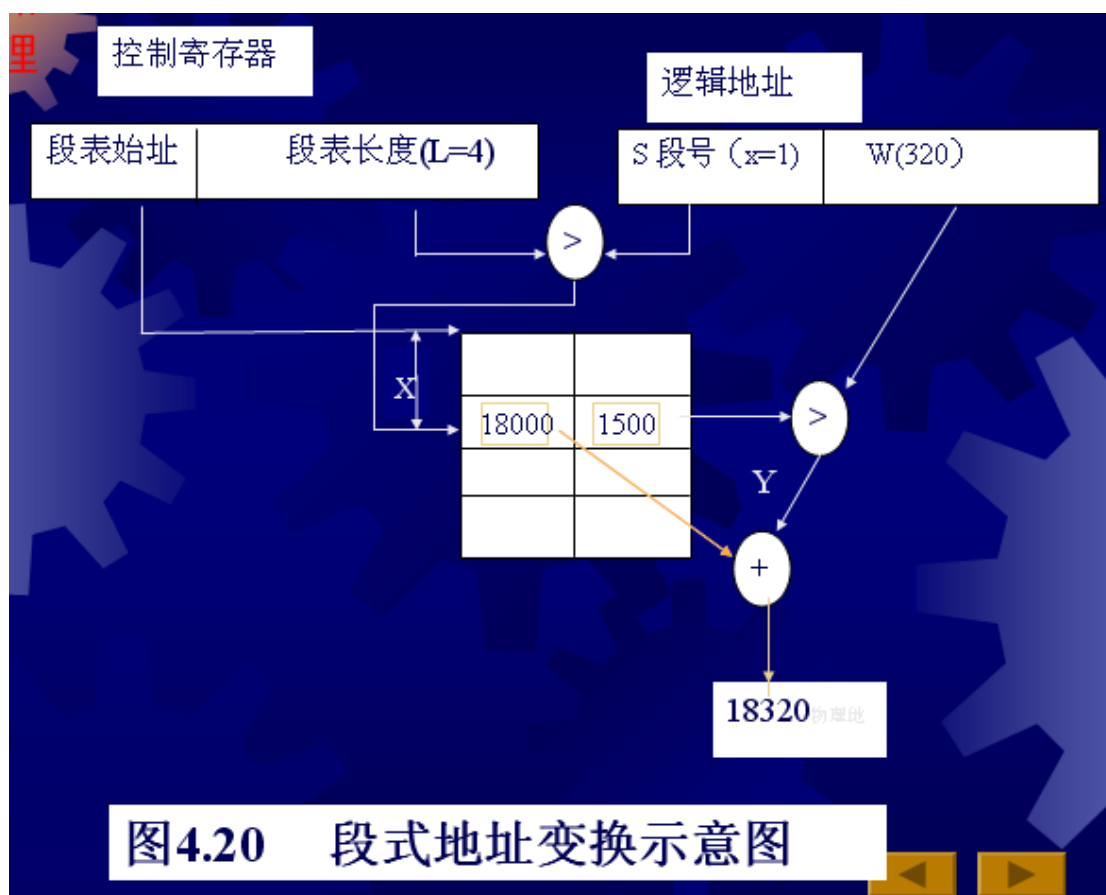
- 作业分配时，用作业的长度与空闲分区表的所有记录的长度之和进行比较。若大于则不能装入。否则，可以装入，为该作业创建一个段表。
- 根据作业段的大小在空闲分区表中查找满足其大小的空闲块，将该段装入，该块剩余部分仍作为空闲分区登记在空闲分区表中，并在段表中填入该段的段长和段的起始地址，直至所有段分配完毕。
- 若找不到足够大的空闲分区，可以采用移动技术，合并分散的空闲区后，再装入该作业段。最后，在主存分配表中，登记该作业段表的起始地址和段表的长度。

### (3) 主存空间的回收

- 当作业运行结束时，根据该作业段表的每一条记录，去修改空闲分区表。修改的方式与可变分区回收主存空间相同，根据回收区是否与空闲区相邻，分 4 种情况处理。
- 删除该作业的段表。
- 删除主存分配表中该作业的记录。

## 3. 地址转换与存储保护

### (1)



## (二) 虚拟内存管理

### 1. 虚拟内存基本概念

所谓虚拟存储器，是指具有请求调入功能和置换功能，能从逻辑上对内存容量加以扩充的一种存储器系统。其逻辑容量由内存容量和外存容量之和所决定，其运行速度接近于内存速度，而每位的成本却又接近于外存。

### 2. 请求分页管理方式

原理：分页式虚拟存储系统是将作业信息的副本存放在磁盘这一类辅助存储器中，当作业被调度投入运行时，并不把作业的程序和数据全部装入主存，而仅仅装入立即使用的那些页面，在执行过程中访问到不在主存的页面时，再把它们动态地装入。

当需要执行某条指令或使用某个数据，而发现它们并不在主存时，产生一个缺页中断，系统从辅存中把该指令或数据所在的页面调入内存。

优点：

作业的程序和数据可以按页分散存放在内存中，减少了移动的开销，有效地解决了碎片问题；由于采用请页式管理，用户可用的主存空间大大扩展，既有利于改进主存利用率，又有利于多道程序运行。

主要缺点是：

要有一定硬件支持，要进行缺页中断处理，因而机器成本增加，系统开销加大，此外页内会出现碎片，如果页面较大，则主存的损失仍然很大。

### 3. 页面置换算法



最佳置换算法 (OPT): 当要调入一页而必须淘汰一个旧页时, 所淘汰的页应该是以后不再访问的页或距现在最长时间后再访问的页。这样的调度算法使缺页中断率为最低。然而这样的算法是无法实现的因为在程序运行中无法对以后要使用的页面作出精确的断言。不过, 这个理论上的算法可以用来作为衡量各种具体算法的标准。这个算法是由 Belady 提出来的, 所以叫做 Belady 算法, 又叫做最佳算法 (Optimal)。

先进先出置换算法 (FIFO): 这种算法总是淘汰最先调入主存的那一页, 或者说在主存中驻留时间最长的那一页 (常驻的除外)。

最近最少使用置换算法 (LRU): 该算法淘汰的页面是在最近一段时间里较久未被访问的那一页。它是根据程序执行时所具有的局部性来考虑的, 即那些刚被使用过的页面, 可能马上还要被使用, 而那些在较长时间内未被使用的页面, 一般说可能不会马上使用到。

时钟置换算法 (CLOCK):

- 算法的实现要点如下:
- 一个页面首次装入主存时, 其“引用位”置 0。
- 在主存中的任何一个页面被访问时, 其“引用位”置 1。
- 淘汰页面时, 存储管理从指针当前指向的页面开始扫描循环队列, 把所迁到的“引用位”是 1 的页面的“引用位”清成 0, 并跳过这个页面; 把所迁到的“引用位”是 0 的页面淘汰掉, 指针推进一步。
- 扫描循环队列时, 如果迁到的所有页面的“引用位”为 1, 指针就会绕整个循环队列一圈, 把碰到的所有页面的“引用位”清 0; 指针停在起始位置, 并淘汰掉这一页, 然后, 指针推进一步。

#### 4. 页面分配策略

分配给一个进程多少页面。

固定数目分配 与 可变数目分配

##### 1) 平均分配

如内存 128 页, 进程 25 个, 每个进程 5 个页面

##### 2) 按进程长度比例分配

$p_i$  共  $s_i$  个页面;  $S = \sum s_i$ ; 内存共  $m$  个页面

$$a_i = (s_i / S) \times m$$

##### 3) 按进程优先级比例分配

##### 4) 按进程长度和优先级别比例分配

#### 5. 抖动, 抖动现象; 工作集。

抖动: 页面在内存与外存之间频繁换入换出。

原因: (1) 分给进程物理页架过少;

(2) 淘汰算法不合理。

处理: (1) 增加分给进程物理页架数;

(2) 改进淘汰算法。

工作集(working set): 进程在一段时间内所访问页面的集合。

- 如果在一段时间内, 作业占用的主存块数目小于工作集时, 运行过程中就会不断出现缺页中断, 从而导致系统的抖动。所以为了保证作业的有效运行, 在相应时间段内就应该根据工作集的大小分配给它主存块, 以保证工作集中所需要的页面能够进入主存。

- 推而广之, 为了避免系统发生抖动, 就应该限制系统内的作业数, 使它们的工作集

总尺寸不超过主存块总数。

#### 6. 请求分段管理方式

分段式虚拟存储系统把作业的所有分段的副本都存放在辅助存储器中,当作业被调度投入运行时,首先把当前需要的一段或几段装入主存,在执行过程中访问到不在主存的段时再把它们装入。

#### 7. 请求段页式管理方式

请求段页式存储管理方式是请求段式存储管理方式和请求页式存储管理方式的结合。它把内存划分为若干大小相等的帧,对用户程序按照其内在的逻辑关系划分成若干段,每一段划分成若干大小与内存帧相等的页面。内存以帧为单位分配给每个进程,在逻辑上相邻的页面在内存中不一定相邻,各段不要求全部装入内存,段的各页也不要求同时全部装入内存。

#### 习题:

1、某采用页式存储管理的系统,接收了一共7页的作业,作业执行时依次访问的页为:1、2、3、4、2、1、5、6、2、1、2、3、7、6、3、2、1、2、3、6。分别用FIFO与LRU页面调度算法,在得到4块主存空间情况下,分别列出各自的页面淘汰顺序和缺页中断次数,以及最后留驻主存4页的顺序。(假定前面四页1234已在主存)

答: FIFO: 1 2 3 4 5 6 2 1 3 7 10次 (3分) 1 3 6 2 (1分)

LRU: 3 4 5 6 1 7 6次 (3分) 6 3 2 1 (3分)

注: 假定前面四页1234已在主存

2、若在一分页存储管理系统中,某作业的页表如下所示,已知页面大小为1024字节,试将逻辑地址1011、2148、4000、5012转化为相应的物理地址。

页号	块号
0	2
1	3
2	1
3	6

答: 设页号为P, 页内位移为W, 则:

1011: P=0, W=1011, 在第2块, 物理地址为3059; (2分)

2148: P=2, W=100, 在第1块, 物理地址为1124; (2分)

4000: P=3, W=928, 在第6块, 物理地址为7072; (2分)

5012: P=4, W=916, 超过页表长度, 该逻辑地址非法。 (2分)

3、用可变分区方式管理主存, 假定主存中按地址顺序依次有5个空闲区, 其大小依次为32K、10K、5K、228K、100K。现有5个作业J1、J2、J3、J4、J5, 它们各自需要主存1K、10K、108K、28K、115K。若采用最先适应分配算法能把这5个作业按J1-J5的顺序全部装入主存吗? 你认为按怎样的次序装入这5个作业可使主存空间利用率最高?

答：若采用最先使用分配算法装入，J1 和 J2 会被装入长度为 32K 的第一个空闲区，总共占用 11K，作业 J3 和 J4 装入长度为 228K 的第四个空闲区，剩余 92K，则剩下的任何一个空闲区都不能满足 J5 的需要。因此，5 个作业不能全部装入主存。（4 分）

采用最先适应分配算法，但按作业对主存的需求量从大到小的次序装入，则可使主存利用率最高。作业 J5 和 J3 占用 228K 空闲区中的 223K，作业 J4 和 J1 占用第一个空闲区中的 29K，作业 J2 正好占用长度为 10K 的第二个空闲区。于是，5 个作业全部被装入主存，空闲区充分利用。（4 分）

4、一个采用页式虚拟管理的系统中，有一用户作业，依次访问字地址序列为：115、228、120、88、446、102、321、432、260、167，若该作业的第 0 页已经装入主存，分配给该作业的主存共 300 字，页的大小为 100 字，计算 FIFO 和 LRU 调度的缺页中断次数、淘汰页和中断率。

答：FIFO 共产生 5 次缺页中断（2 分），淘汰了 0、1、2 页，中断率 50%（2 分）；LRU 共产生 6 次缺页中断（2 分），淘汰了 2、0、1、3 页，中断率 60%（2 分）

5、页式存储管理中，主存按页分配，可用一张“位示图”构成主存分配表。假设主存的容量为 2M 字节，页面长度为 512 字节，若用字长为 32 位的字作主存分配的“位示图”需要多少字？如页号从 1 开始，字号和字内位号（从高位到低位）均从 0 开始，试问第 2999 块对应与何字何位；99 字 19 位又对应于第几块？

答：128 个字（字数 =  $2^{21} \div 2^9 \div 2^5 = 128$ ）（2 分）

93 字 22 位（字 =  $2999 \text{DIV} 32 = 93$ ；位号 =  $2999 \text{MOD} 32 - 1 = 22$ ）（3 分）

第 3188 块（块号 =  $99 \times 32 + 19 + 1 = 3158 + 20 = 3188$ ）（3 分）

6、某系统采用页式存储管理策略，拥有逻辑空间 32 页，每页 2KB，拥有物理空间 1MB。（1）写出逻辑地址格式。（2）若不考虑访问权限，进程的页表有多少项？每项至少有多少位？（3）如果物理空间减少一半，页表结构应相应作怎样的调整？

答：（1）该系统拥有逻辑空间 32 页，故逻辑地址中页号必须用 5 位来描述，而每页 2KB，因此页内地址必须用 11 位来描述。（2）每个进程最多有 32 个页面，因此进程的页表项最多为 32 项。若不考虑访问权限，则页表项中只需给出页所对应的物理块号，1MB 的物理空间可分成  $2^9$  个内存块，故每个页表至少有 9 位。（3）如果物理空间减少一半，则页表中页表项数仍不变，但每项的长度可减少 1 位。

7、假定某页式管理系统，主存为 64KB，分成 16 块，块号为 0、1、2……15。设某作业有 4 页，其页号为 0、1、2、3，被分别装入主存的 2、4、1、6 块，试问：（1）该作业的总长度是多少字节？（按十进制）（2）写出该作业每一页在主存中的起始地址。（3）若给出逻辑地址 [0, 100]、[1, 50] [2, 0] [3, 60]，请计算相应的内存地址。

答：（1）每块的长度为  $64\text{KB}/16=4\text{KB}$ ，作业的总长度为 16KB。

(2) PMT 如下表所示:

页号	块号
0	2
1	4
2	1
3	6

所以该作业的: 第 0 页在主存中的起始地址为 8K; 第 1 页在主存中的起始地址为 16K; 第 2 页在主存中的起始地址为 4K; 第 3 页在主存中的起始地址为 24K。

(3) 逻辑地址 [ 0, 100] 的内存地址为  $4K \times 2 + 100 = 8292$ ; 逻辑地址 [ 1, 50] 的内存地址为  $4K \times 4 + 50 = 16434$ ; 逻辑地址 [ 2, 0] 的内存地址为  $4K \times 1 + 0 = 4096$ ; 逻辑地址 [ 3, 60] 的内存地址为  $4K \times 3 + 60 = 24636$ ;

8、对于如图所示的段表, 请将逻辑地址[0, 137], [1, 4000], [2, 3600], [5, 230]转换成物理地址。

段号	内存地址	段长/B
0	50K	10K
1	60K	3K
2	70K	5K
3	120K	8K
4	150K	4K

分析: 在分段系统中进行地址转换时, 先将逻辑地址中的段号与段表长度作比较, 如果段号超长, 则产生越界中断, 否则使用段号为索引去检索段表, 从中得到段在内存的起始地址和段长, 然后再将逻辑地址中的段内地址和段长作比较, 若不越界, 则由段的起始地址和段内地址相加, 形成物理地址。

答: (1) 段号、段内地址均合法, 物理地址为:  $50K + 137 = 51337$ 。

(2) 段号合法、段内地址超过段长, 产生越界中断。

(3) 段号、段内地址均合法, 物理地址为:  $70K + 360 = 75280$ 。

(4) 段号不合法, 产生越界中断。

9、已知页面走向为: 1、2、1、3、1、2、4、2、1、3、4, 且开始执行时主存中没有页面, 若只给该作业分配 2 个物理块, 当采用 FIFO 页面淘汰算法时缺页率为多少? 假定现有一种淘汰算法, 该算法淘汰页面的策略为当需要淘汰页面时, 就把刚使用过的页面作为淘汰对象, 试问就相同的页面走向, 其缺页率又为多少?

答: 采用 FIFO 时的页面置换情况如下表所示:

页面走向	1	2	1	3	1	2	4	2	1	3	4
物理块 1	1	1	3	3	2	2	1	1	4		
物理块 2		2		2	1	1	4		4	3	3

缺页	缺	缺		缺	缺	缺	缺		缺	缺	缺
----	---	---	--	---	---	---	---	--	---	---	---

页面的引用次数为 11 次，缺页次数为 9 次，缺页率为 9/11。

采用后一种时的页面置换情况如下表所示：

页面走向	1	2	1	3	1	2	4	2	1	3	4
物理块 1	1	1		3	1		1	1		3	4
物理块 2		2		2	2		4	2		2	2
缺页	缺	缺		缺	缺		缺	缺		缺	缺

页面的引用次数为 11 次，缺页次数为 8 次，缺页率为 8/11。

10、考虑一个请求调页系统，它采用全局置换策略和平均分配内存块的算法（即若有  $M$  个内存块和  $N$  个进程，则每个进程分得  $M/N$  个内存块）。若在该系统中测得如下 CPU 和对换内存利用率，请问能否增加多道程序的度数来增加 CPU 的利用率？为什么？

- (1) CPU 的利用率 13%，内存利用率 97%；(2) CPU 的利用率 87%，内存利用率 3%；  
(3) CPU 的利用率 13%，内存利用率 3%；

答：(1) 这种情况表示系统在进行频繁的置换，绝大部分时间被用在页面置换上。此时，增加多道程序的度数会进一步增加缺页率，使系统性能进一步恶化。

(2) 此时，CPU 的利用率相当高，但对换内存的利用率却相当低，表示，运行进程的缺页率很低，可适当增加多道程序的度数来增加 CPU 的利用率。

(3) 此时，CPU 的利用率和对换内存的利用率都相当低，表示内存中可运行的程序数不足，应增加多道程序的度数来增加 CPU 的利用率。

11、假如一个程序的段表如图所示，其中存在位为 1 表示段在内存，存取控制字段中 W 表示可写，R 表示可读，E 表示执行。对下面的指令，在执行时会产生什么样的结果？

段号	存在位	内存始址	段长	存取控制	其他信息
0	0	500	100	W	
1	1	1000	30	R	
2	1	3000	200	E	
3	1	8000	80	R	
4	0	5000	40	R	

- (1) STORE R1, [0, 70]      (2) STORE R1, [1, 20]      (3) LOAD R1, [3, 20]  
(4) LOAD R1, [3, 100]      (5) JMP[2, 100]

分析：在执行指令的过程中，如果指令中包含有地址部分，则先必须进行逻辑地址到物理地址的转换，在地址转换过程中还要进行越界检查和存取控制权限的检查。只有在地址不越界访问方式也合法，并形成物理地址后，才能去完成指令规定的操作。

答：(1) 从段表的第 0 项可读出第 0 段的存在位为 0，表示相应段未装入内存，以那次地址变换机构将产生一缺段中断，以请求 OS 将其调入内存。

(2) 从段表的第 1 项可以看出，虽然指令中的逻辑地址合法，段也在内存，但本指令对内

存的访问方式（写）与控制字段（只读）不符，故硬件将产生保护性中断信号。

（3）从段表的第3项可读出第3段的存在位为1，内存起始地址为8000，段长为80，存取控制为R，因此，逻辑地址合法，访问方式也合法，形成物理地址8020后，指令将把该单元的内容读到寄存器R1中。

（4）从段表的第3项可读出第3段的存在位为1，内存起始地址为8000，段长为80，存取控制为R，因此，指令的逻辑地址中段内地址超过了段长，地址变换机构将产生越界中断信号。

（5）从段表的第2项可读出第2段的存在位为1，内存起始地址为3000，段长为200，存取控制为E，因此，因此，逻辑地址合法，访问方式也合法，形成物理地址3100，指令执行后，将跳转到内存单元3100处继续执行。

12、请画出请求页式存储管理的系统工作流程图，即给出硬件及软件（缺页中断）处理过程的流图。

#### 四、文件管理

##### （一）文件系统基础

##### 1. 文件概念

文件是具有符号名的信息（数据）项的集合。

UNIX 系统文件分类方法：

- 1) 普通文件：普通文件是无结构的字符的集合。
- 2) 目录文件：目录文件是由文件的目录项组成的文件。
- 3) 特别文件：特别文件就是设备。

文件系统是操作系统中负责管理和存取文件信息的软件机构，它是由管理文件所需的数据结构和相应的管理软件以及访问文件的一组操作组成。

##### 2. 文件的逻辑结构

顺序文件；索引文件；索引顺序文件。

文件的逻辑结构：文件的逻辑结构是指用户思维中文件的结构。文件的逻辑结构：

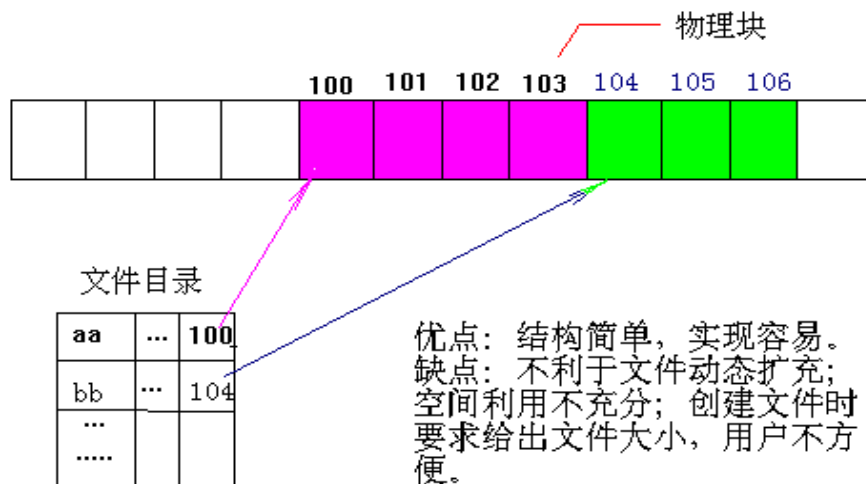
结构文件——记录式文件。

无结构文件——流式文件

文件的物理结构：文件的物理结构是指文件在存储介质上的结构（或称组织）。在当代，文件的存储介质是磁盘，包括软盘、硬盘和光盘、磁带，早期还有磁鼓。由于目前的磁带是模拟磁盘的结构，所以文件的物理结构主要是指磁盘上文件的结构。

## 1. 连续文件

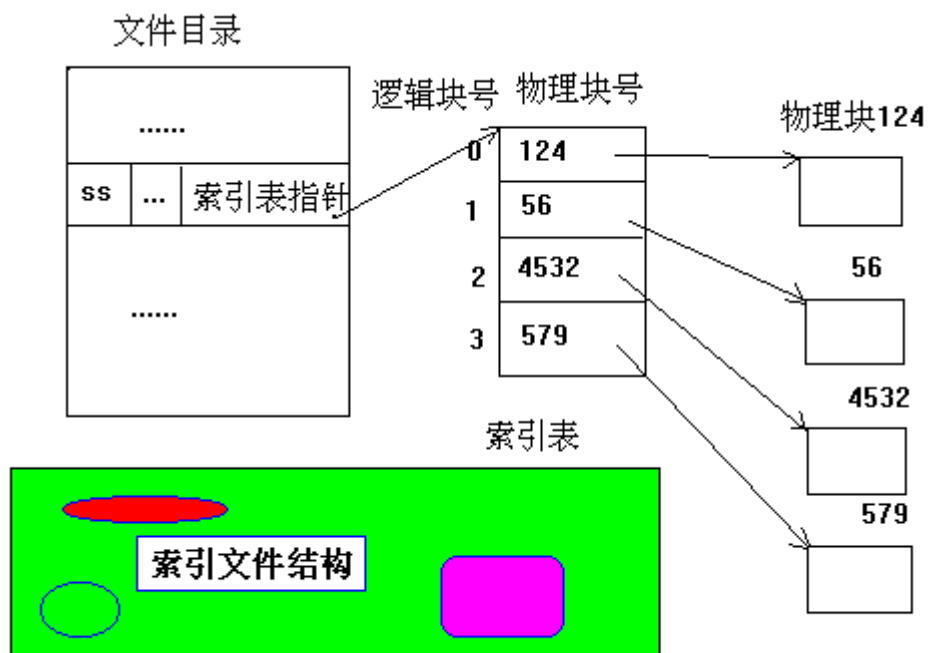
一个连续文件是由一组分配在磁盘连续区域的物理块组成的，



2) 索引文件结构 这种文件结构的数据结构是文件的索引表，每个文件有一个索引表，表中每个表目包括：逻辑块号，物理块号。

索引表位置：文件目录中，文件的开头等。

索引表大小：固定大小，非固定大小。



3) 索引顺序文件是顺序文件的扩展，其中各记录本身在介质上也是顺序排列的，它包含了直接处理和修改记录的能力。索引顺序文件能象顺序文件一样进行快速顺序处理，既允许按物理存放次序（记录出现的次序）；也允许按逻辑顺序（由记录主键决定的次序）进行处理

### 3.目录结构

文件控制块和索引节点；单级目录结构和两级目录结构；树形目录结构；图形目录结构。

**文件控制块（FCB）：**文件控制块是操作系统为管理文件而设置的数据结构，存放了为管理文件所需的所有有关信息。文件控制块是文件存在的标志。

**文件目录：**把所有的 FCB 组织在一起，就构成了文件目录，即文件控制块的有序集合。

**目录文件：**为了实现对文件目录的管理，通常将文件目录以文件的形式保存在外存，这个文件就叫目录文件。

**目录项：**构成文件目录的项目（目录项就是 FCB），文件目录项由文件名和文件 I 节点（索引节点）号组成。

UNIX 系统中文件目录项只包括文件名和 I 节点号，把通常目录项中的信息放在一个叫索引节点（I 节点）的结构中，有两个优点：其一是便于文件的共享，即不同的用户可以通过不同的文件路径名来共享一个文件；其二是加快文件检索的速度。

**一级目录结构：**为所有文件建立一个目录文件(组成一线性表)。

优点：简单，易实现

缺点：限制了用户对文件的命名

文件平均检索时间长

限制了对文件的共享

#### 二级目录结构

为改变一级目录文件目录命名冲突，并提高对目录文件检索速度而改进。

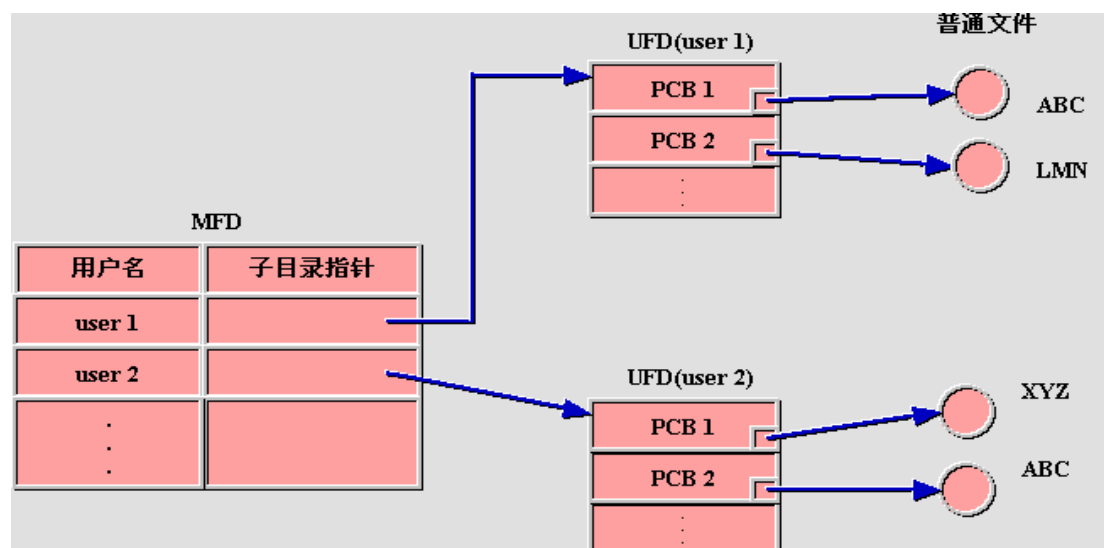
目录分为两级：一级称为主文件目录，给出用户名，用户子目录所在的物理位置；二级称为用户文件目录（又称用户子目录），给出该用户所有文件的 FCB。

优点：解决了文件的重名问题和文件共享问题

用户名 ^ 文件名

查找时间降低

缺点：增加了系统开销



#### 多级目录结构（树型目录）

优点：层次结构清晰，便于管理和保护，解决重名问题，查找速度加快。

缺点：查找一个文件按路径名逐层检查，由于每个文件都放在外存，多次访盘影响速度。



#### 4.文件共享

~~共享动机；共享方式；共享语义。~~

1) 定义：一个文件被多个用户或程序使用。

2) 共享形式：

被多个用户使用，由存取权限控制

被多个程序使用，但各用自己的读写指针

被多个程序使用，但共享读写指针

多个用户用相同或不同的名字来访问同一文件。

3) 动机：节省时间和存储空间，减少了用户工作量；进程间通过文件交换信息。

#### 5.文件保护

访问类型；访问控制。

文件保护：用于提供安全性的特定的操作系统机制。

❖ 对拥有权限的用户，应该让其进行相应操作，否则，应禁止

❖ 防止其他用户冒充对文件进行操作

❖ 实现：

\* 用户验证：当用户登录时，检验其身份

\* 存取控制：审查用户的权限；审查本次操作的合法性

方法一：文件的二级存取控制

第一级：对访问者的识别

对用户分类：

a) 文件主 (owner)

b) 文件主的同组用户 (group)

c) 其它用户 (other)

第二级：对操作权限的识别

对操作分类：

a) 读操作 (r)

b) 写操作 (w)

c) 执行操作 (x)

d) 不能执行任何操作 (-)

方法二：存取控制矩阵

用户	文件			
	A	B	C	.....
User1	rw	r	w	
User2	e	-	-	.....

#### (二) 文件系统实现

## 6. 文件系统层次结构



## 7. 目录实现

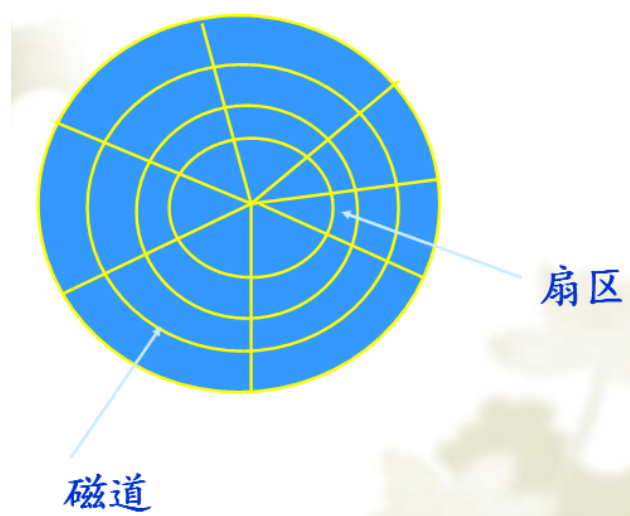
(1) 线性列表：用一张表格实现从存储文件名到数据块指针的映射。这个方法编程简单，但效率很低。

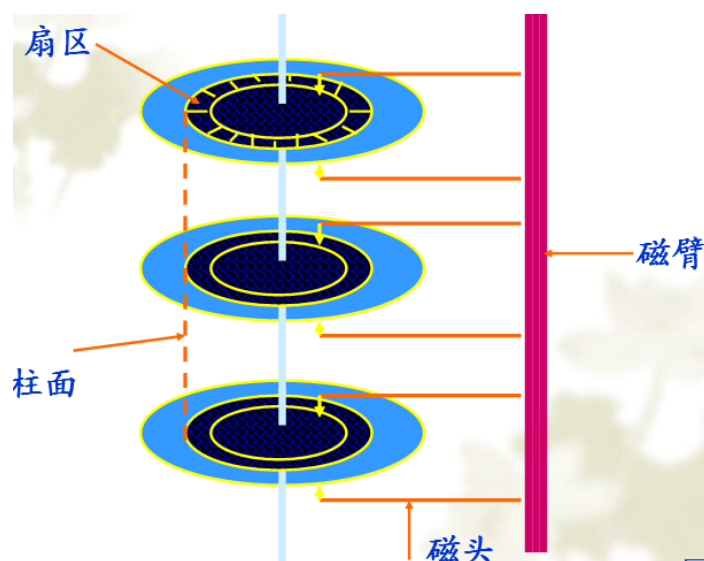
(2) 哈希表：用哈希函数来实现从存储文件名到数据块指针的映射。大大降低搜索时间，也提高了插入和删除操作的效率，但会带来冲突处理问题。

8. 文件实现：要考虑到文件的逻辑结构、考虑到文件的各种操作的实现、考虑到文件的共享方式实现、考虑到文件在磁盘上的物理块的分配方法、考虑到磁盘空闲空间的管理、考虑到数据不一致时的恢复和文件保护等等。

## (三) 磁盘组织与管理

## 1. 磁盘的结构





- ❖ 信息记录在磁道上，多个盘片，正反两面都用来记录信息，每面一个磁头
- ❖ 所有盘面中处于同一磁道号上的所有磁道组成一个柱面
- ❖ 物理地址形式：
  - 磁头号（盘面号）
  - 磁道号（柱面号）
  - 扇区号

## 2. 磁盘调度算法

一次访盘时间 = 寻道时间 + 旋转延迟时间 + 存取时间

磁盘调度的目标：减少寻道时间，减少延迟时间。

（1）先来先服务：按访问请求到达的先后次序服务。

优点：简单，公平；

缺点：效率不高，相临两次请求可能会造成最内到最外的柱面寻道，使磁头反复移动，增加了服务时间，对机械也不利。

（2）最短寻道时间优先：优先选择距当前磁头最近的访问请求进行服务，主要考虑寻道优先。

优点：改善了磁盘平均服务时间；

缺点：造成某些访问请求长期等待得不到服务

（3）扫描算法（电梯算法）

当设备无访问请求时，磁头不动；当有访问请求时，磁头按一个方向移动，在移动过程中对遇到的访问请求进行服务，然后判断该方向上是否还有访问请求，如果有则继续扫描；否则改变移动方向，并为经过的访问请求服务，如此反复。

（4）单向扫描调度算法

- 总是从 0 号柱面开始向里扫描
- 按照各自所要访问的柱面位置的次序去选择访问者
- 移动臂到达最后一个柱面后，立即带动读写磁头快速返回到 0 号柱面
- 返回时不为任何的等待访问者服务
- 返回后可再次进行扫描

（5）旋转调度算法：根据延迟时间来决定执行次序的调度

## 3. 磁盘的管理

习题：

1、假定磁盘的移动臂现处于 8 号柱面，列出如下访问者访问磁盘时的最省时响应次序。

序	柱面号	磁头号	扇区号
1	9	6	3
2	7	5	6
3	15	20	6
4	9	4	4
5	20	9	5
6	7	15	2

答：能使寻找时间和延迟时间最短的响应次序是最省时间的，还应考虑旋转调度，故响应次序为：6、2、1、4、3、5。（8 分）

2、某磁盘有 200 个柱面，编号为 0—199。若为 143 柱面的请求者服务后，正在访问 125 柱面的请求服务，同时有若干请求者在等待服务，柱面号依次为：86、147、91、177、94、150、102、175、130，请回答下列问题：

（1）分别用先来先服务、最短寻道时间优先算法来确定实际的服务顺序。（2）分别计算上述算法的移臂距离。

答：先来先服务：86、147、91、177、94、150、102、175、130，（2 分）共移动 547 个柱面（2 分）；最短寻道：130、147、150、175、177、102、94、91、86，（2 分）共移动 143 个柱面。

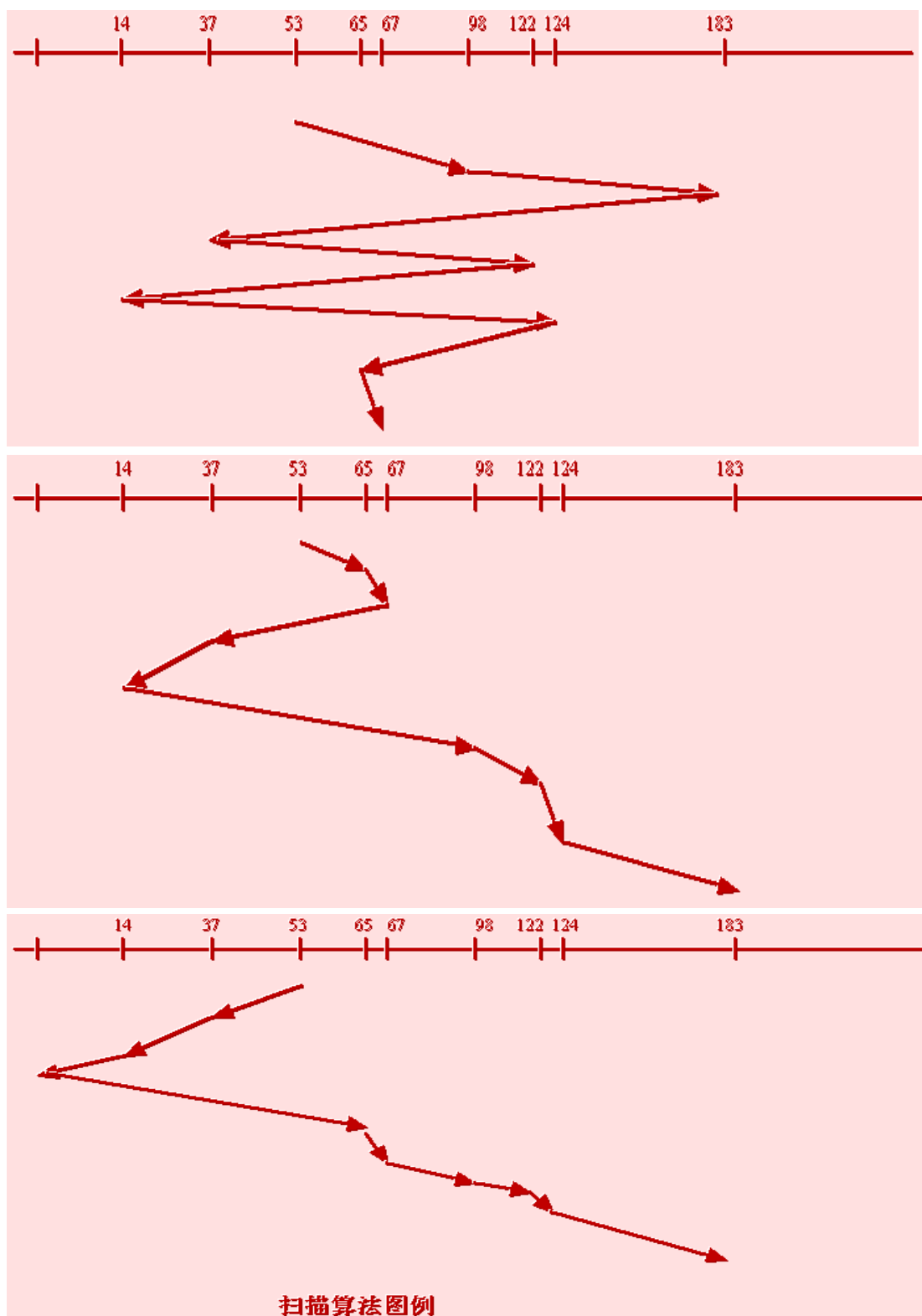
（1）电梯调度和单向扫描算法来确定实际的服务顺序。（2）分别计算上述算法的移臂距离。

答：电梯调度：102、94、91、86、130、147、150、175、177，（2 分）共移动 130 个柱面；（2 分）单向扫描：130、147、150、175、177、86、91、94、102，（2 分）共移动 159 个柱面。（2 分）

3、若干个等待磁盘的访问者依次要访问的柱面为：20、44、40、4、80、12、76，假设每移动一个柱面需要 3 毫秒时间，移动臂当前处于 40 号柱面，按先来先服务和最短寻道时间优先算法，计算完成上述访问所需时间。

答：先来先服务：移动 20、24、4、36。76、68、64，292 个柱面，876 毫秒；（4 分）最短寻道时间：移动 0、4、24、8、8、72、4，120 个柱面，360 毫秒。（4 分）

4、假设磁盘访问序列：98，183，37，122，14，124，65，67，读写头起始位置：53，分别用 FIFO、最短寻道时间优先和电梯算法，安排磁头服务序列，计算磁头移动总距离（道数）。



5、某磁盘的旋转速度为 20 毫秒/圈，每个盘面被分成 10 个扇区，现有 10 个逻辑记录存放在同一磁道上，安排如下：

扇区号	逻辑记录	扇区号	逻辑记录
1	A	6	F
2	B	7	G
3	C	8	H
4	D	9	I
5	E	10	J

处理程序要顺序处理这些记录，每读出一个记录后处理程序要花 4 毫秒的时间进行处理，然后顺序读下一个记录并处理，直到处理完这些记录，回答：（1）顺序处理完这些记录总共花了多少时间？（2）请给一种记录优化分布的方案，使处理程序能在最短时间内处理完这些记录，并计算优化分布后处理所需时间。

答：（1）读一个记录花 2 毫秒，处理花 4 毫秒，顺序的下一个记录需等待 16 毫秒才能再次旋转到磁头下。所以，除第一个记录外，其它记录都需花 16 秒读出，总时间为 204 毫秒。

（4 分）（2）优化要求前一个记录处理完，下一个记录正在磁头下，所费时间最少。则，总时间仅为读出和处理记录时间之和，共 60 毫秒。（4 分）分布如下图所示：

扇区号	逻辑记录	扇区号	逻辑记录
1	A	6	F
2	H	7	C
3	E	8	J
4	B	9	G
5	I	10	D

6、设磁盘有 200 个磁道，磁盘请求队列中是一些随机请求，他们按照到达的次序分别处于 55、58、39、18、90、160、150、38、184 号磁道上，，当前磁头在 100 号磁道上，并向磁道号增加的方向上移动，请给出按 FCHS、SSTF、SCAN 以及 CSCAN 算法进行磁盘调度时满足请求的次序，并计算他们的平均寻道长度。

分析：以上各算法分别指：按进程请求访问磁盘的先后顺序服务；最近距离请求优先服务；当前移动方向上距离最近请求优先服务和单向移动下当前移动方向上距离最近请求优先服务。

答：磁盘调度的次序以及他们的平均寻道长度，如下表所示：

FCFS		SSTF		SCAN		CSCAN	
访问的下一个磁道	移动的磁道数	访问的下一个磁道	移动的磁道数	访问的下一个磁道	移动的磁道数	访问的下一个磁道	移动的磁道数
55	45	90	10	150	50	150	50
58	3	58	32	160	10	160	10
39	19	55	3	184	24	184	24
18	21	39	16	90	94	18	166

90	72	38	1	58	32	38	20
160	70	18	20	55	3	39	1
150	10	150	132	39	16	55	16
38	112	160	10	38	1	58	3
184	146	184	24	18	20	90	32
平均寻道长度: 55.3		平均寻道长度 27.6		平均寻道长度: 27.8		平均寻道长度: 35.8	

7、设磁盘有 200 个磁道，磁盘请求队列中是一些随机请求，他们按照到达的次序分别处于 98、183、37、122、14、160、124、65、67 号磁道上，，当前磁头在 53 号磁道上，并向磁道号减小的方向上移动，请给出按 FCHS、SSTF、SCAN 以及 CSCAN 算法进行磁盘调度时满足请求的次序，并计算他们的平均寻道长度。

答：磁盘调度的次序以及他们的平均寻道长度，如下表所示：

FCFS		SSTF		SCAN		CSCAN	
访问的下一个磁道	移动的磁道数	访问的下一个磁道	移动的磁道数	访问的下一个磁道	移动的磁道数	访问的下一个磁道	移动的磁道数
98	45	65	12	37	16	37	16
183	85	67	2	14	23	14	23
37	146	37	30	65	51	183	169
122	85	14	23	67	2	124	59
14	108	98	84	98	31	122	2
124	110	122	24	122	24	98	24
65	59	124	2	124	2	67	31
67	2	183	59	128	59	65	2
平均寻道长度: 80		平均寻道长度 29.5		平均寻道长度: 26		平均寻道长度: 40.75	

8、有 5 个记录 A、B、C、D、E，存放在某磁盘的某磁道上，假定这个磁道划分成 5 块，每块存放一个记录，安排如下表所示。现在要顺序处理这 5 个记录，如果盘旋转一周需 20ms，处理程序读出一个记录后要花 6ms 进行处理，试问：（1）处理完这 5 个记录所需的总时间是多少？（2）为减少磁盘旋转的周数，应如何安排这 5 个记录，并计算所需的总时间。

块号	1	2	3	4	5
记录号	A	B	C	D	E

答：这是一个旋转调度问题。（1）所需的总时间=20\*5+6=206ms 因为，每转过一个记录需要 4ms，每读一个记录后要 6ms 处理时间，等处理完再读下一个记录时，需要再等下一周，当读出第 5 个记录后还需另外加 6ms 处理时间。

（2）将记录安排改进为：块号 1-5 分别存放记录 A、C、E、B、D。改进后，所需总时间为 20ms\*3=60ms 。

9、旋转型存储设备上信息的优化分布能减少输入输出服务的总时间，例如，有 10 个记录 A、B、……J 存放在某磁盘的一磁道上，假定这个磁道划分为十块，每块存放一个记录，如图。

则：顺序处理这些记录，若磁盘旋转速度为 20ms/r，处理程序每读出一个记录后花 4ms 进行处理。问：处理完这 10 个记录的总时间是多少？为缩短处理的总时间应进行怎样的安排以优化其分布？并计算优化处理后的总时间。

块号	1	2	3	4	5	6	7	8	9	10
记录号	A	B	C	D	E	F	G	H	I	J

答：（1）由题意磁盘旋转一周时间为 20ms，读取一个记录时间为 2ms，处理一个记录时间为 4ms。处理完前 9 个并到达各自下一个记录起始位置的时间为：22ms/个；处理完记录 J 所需时间为 6ms，综上，总时间为 204ms。

（2）优化分布后安排如下表：

块号	1	2	3	4	5	6	7	8	9	10
记录号	A	H	E	B	I	F	C	J	G	D

每处理一个的时间是 6ms，总时间是 60ms。

10、假设磁盘共有 200 个柱面，编号从 0~199，当前存取臂在 120 号柱面上服务，并刚刚完成了 105 号柱面的请求。如果现有进程 P1、P2、P3、P4 分别请求的柱面号为：186、158、115、90。按照下列三种调度算法时，试问：系统调度的次序是什么？存取臂的移动总量为多少？（1）先来先服务 （2）最短查找时间优先 （3）电梯调度算法

答：FCFS： P1、P2、P3、P4 186、158、115、90 移动总量 162；

SSTF： P3、P4、P2、P1 115、90、158、186 移动总量 126；

电梯： P2、P1、P3、P4 158、186、115、90 移动总量 159；

## 五、输入输出（I/O）管理

### （一）I/O 管理概述

1. I/O 设备
2. I/O 管理目标
3. I/O 管理功能
4. I/O 应用接口
5. I/O 控制方式

### （二）I/O 核心子系统

1. I/O 调度概念
2. 高速缓存与缓冲区
3. 设备分配与回收
4. 假脱机技术（SPOOLing）
5. 出错处理

### （一）I/O 管理概述

1. I/O 设备：在计算机系统中除 CPU 和内存储外所有的设备和装置称为计算机外部设备（外围设备、I/O 设备）。

设备特性：外存/辅存，IO 设备

设备分类：

- 从属关系：



- 系统设备：OS 生成时已配置于系统的各种标准设备
- 用户设备：用户后来自己提供，由系统管理，非标准
- 信息组织和处理方式：
  - 块设备：信息按字符块组织和处理（面向块的设备）
  - 字符设备：信息按字符组织和处理（面向字符的设备）
- 资源分配方式：
  - 独占设备：一旦分配给某进程或用户就一直占用直至用完。
  - 共享设备：多个进程或用户可以交替使用的设备。
  - 虚拟设备：用软件技术（SPOOL）把慢速独占设备变成共享设备。一般是通过借用大容量共享设备的一部分空间来充当缓冲而实现的。把这部分空间称为“虚拟设备”。（Spooling 技术）

## 2. I/O 管理目标

（1）方便性：用户无须了解物理设备的细节，也能从不熟练设备的使用中解脱出来而备感灵活和方便。

（2）并行性：CPU 和 I/O 设备同时工作，提高设备利用率。

（3）均衡性：能够使 CPU 和 I/O 操作的忙闲程序保持相对平衡。

（4）独立性：即设备的物理特性相对于用户来说是透明的。

## 3. I/O 管理功能

- 提供和进程管理系统的接口。
- 进行设备分配。
- 实现设备和设备、设备和 CPU 等之间的并行操作。
- 进行缓冲区管理。

## 4. I/O 应用接口

可以从不同的 I/O 设备中抽象出一些通用类型。每个类型都可以通过一组标准函数（接口）来访问。具体的差别被内核模块（设备驱动）所封装，这些设备驱动程序一方面可以定制，以适合各种设备，另一方面也提供一些标准接口。

## 5. I/O 控制方式

- 按照 I/O 控制器功能的强弱，及和 CPU 间联系方式的不同，对 I/O 设备的控制方式分类。
- 主要差别在于：中央处理器和外围设备并行工作的方式不同，并行工作的程度不同。

### 四种 I/O 方式：

#### (1) 程序直接控制（轮询）方式

- 由用户程序来直接控制内存或 CPU 和外设之间的信息传送。
- 用户程序通过 CPU 发“启动”命令      用户程序等待、CPU 循环测试      若空闲则设置“忙”      开始数据传送
- 缺点：
  - 1、CPU 和外设只能串行工作。
  - 2、无法发现和处理由于设备或其它硬件产生的错误。
- 注：控制状态寄存器只有表示设备状态的位

#### (2) 中断方式

- 为了提高 CPU 和设备的利用率，就应使 CPU 与设备并行工作，采用 I/O 中断方式。采用这种方式要求控制寄存器中有一个中断位。

- 在 I/O 中断方式下，数据的输入（或输出）步骤如下：
- 设备驱动程序工作：
- 1、要求输入数据的进程把一个启动命令和允许中断位“1”写入相应设备的控制状态寄存器中，从而启动了该设备；
- 2、该进程因等待输入的完成进入睡眠状态。
- 3、当输入完成后，输入设备向 CPU 发出完成中断请求信号；设备中断处理程序工作：
- 4、处理机响应中断，处理该中断，并唤醒等待输入完成的进程；
- 5、在以后的某个时期，该程序被调度到后，继续运行。
- 这种方式的优点是大大地提高了 CPU 的利用效率，缺点是每次 I/O 都要 CPU 的干预，如果系统中配备了多台（套）设备时，CPU 的利用率也会降低。解决的方法是采用通道技术。

### (3) DMA 方式

- 数据在内存与 I/O 设备间的直接成块传送的方式。I/O 控制器比程序直接控制方式和中断方式更强，增加传送字节计数器、和内存地址寄存器等。CPU 在开始时向设备发“传送一块”命令，结束时进行相应处理，实际操作由 DMA 硬件直接完成
- DMA 能够通过系统总线代替 CPU 管理数据的存入或取出
  - 当 CPU 不需要系统总线时可以使用总线
  - DMA 可以强迫 CPU 暂时延迟其他操作，获取一个总线周期（周期窃取）

### (4) 通道方式

- 通道是独立于 CPU 的专门负责数据输入/输出传输工作的处理机，对外部设备实现统一管理，代替 CPU 对输入/输出操作进行控制，从而使输入，输出操作可与 CPU 并行操作。

## （二）I/O 核心子系统

### 1. I/O 调度概念

调度一组 I/O 请求就是确定一个好的顺序来执行这些请求。应用程序所发布的系统调用的顺序并不一定总是最佳选择。调度能改善系统的整个性能，能在进程之间公平地共享设备访问，能减少 I/O 完成所需要的平均等待时间。安排服务顺序就是 I/O 调度的核心。

### 2. 高速缓存与缓冲区

高速缓存是可以保留数据拷贝的高速内存。

缓冲区是用来保存在两设备之间或在设备和应用程序之间所传输数据的内存区域。

### 3. 设备分配与回收

设备分配：当某进程向系统提出 I/O 请求时，设备分配程序按一定策略分配设备、控制器和通道，形成一条数据传输通路，以供主机和设备间信息交换。

#### 数据结构：

- 1、设备控制块 DCB（设备控制表 DCT）
- 2、系统设备表 SDT
- 3、控制器控制块 COCB（控制器控制表 COCT）
- 4、通道控制块 CHCB（通道控制表 CHCT）

**分配原则：**由于在多道程序系统中，进程数多于资源数，引起资源的竞争。因此，要有一套合理的分配原则

考虑的因素：

- I/O 设备的固有属性

- I/O 设备的分配算法
- 设备分配的安全性
- 与设备的无关性（设备独立性）

#### ● 独占设备的分配

要考虑充分发挥效率，避免由于不合理的分配策略造成死锁

#### ● 共享设备分配

由于同时有多个进程同时访问，且访问频繁，就会影响整个设备使用效率，影响系统效率。因此要考虑多个访问请求到达时服务的顺序，使平均服务时间越短越好

- **静态分配：**在进程运行前，完成设备分配；运行结束时，收回设备。

缺点：设备利用率低

- **动态分配：**在进程运行过程中，当用户提出设备要求时，进行分配，一旦停止使用立即收回。

优点：效率好

缺点：分配策略不好时，产生死锁

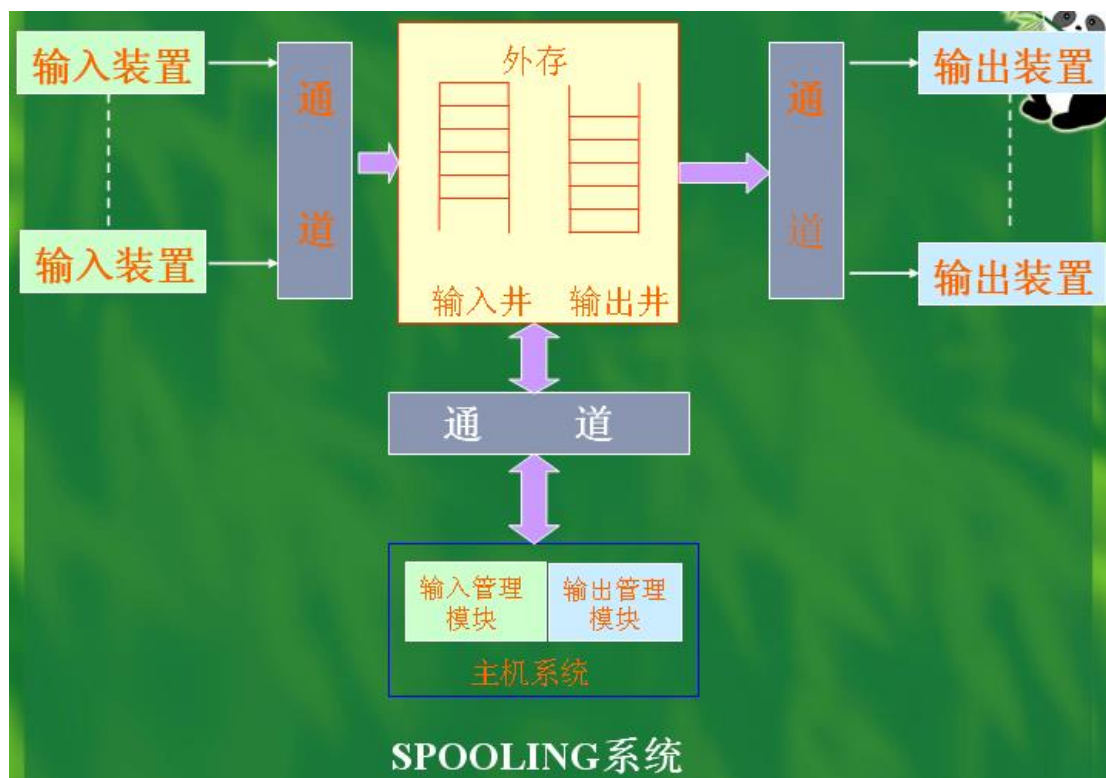
#### 4. 假脱机技术（SPOOLing）

##### ❖ SPOOLING 系统(假脱机操作 )Simultaneous Peripheral Operation On Line

- 它使用直接存取的大容量磁盘作为缓冲，将一个可共享的磁盘空间改造成若干个输入设备和输出设备，并使得 I/O 设备和 CPU 并行操作。（在联机情况下实现的同时外围操作）

##### ❖ SPOOLING 系统的组成

- 输入井和输出井
- 输入缓冲区和输出缓冲区
- 输入进程和输出进程（输入管理模块、输出管理模块）



## ❖ SPOOLING 系统的工作原理

当用户提交一批作业后，操作员键入“预输入命令”启动预输入程序工作，预输入程序启动输入机读出作业信息，并把它们存放到输入井中。当主存储器可以装入作业时就从输入井中选择若干作业装入主存储器。被装入主存储器中的作业在执行中可请求井管理程序从输出井读需处理的信息或把处理结果写到输出井中。缓输出程序利用处理器空闲时间把作业执行结果在打印机上输出

## ❖ SPOOLING 系统的特点

- 提高了 I/O 速度
- 将独占设备改造为共享设备
- 实现了虚拟设备功能

## 5. 出错处理

## 设备管理

系统中与一台由非分配性通道支持的磁盘机，在通道与请求者进程之间只设置了一个磁盘驱动程序来完成请求该进程与设备之间的通信，假设请求者进程首次请求读某逻辑文件的第 N 号记录，请把下列语句进行重新排列，简要描述从请求到完成所经历的如下 5 个步骤：（1）请求者进程发出对文件第 N 号记录的请求。（2）磁盘驱动进程运行。（3）I/O 操作完成，CPU 响应通道发来的完成中断请求。（4）磁盘驱动进程运行。（5）请求者进程运行。

注意：请从下列语句中挑选合适的语句描述上面的 5 个步骤：

- A. 组织通道程序
- B. 申请分配输入缓冲区
- C. 唤醒磁盘驱动程序
- D. 请求者把 I/O 参数通知磁盘驱动进程
- E. 阻塞请求者进程
- F. 求逻辑记录 N 所在的块号
- G. 根据物理块得到三维物理地址
- H. 启动通道与设备
- I. 磁盘驱动进程自行阻塞
- J. 唤醒磁盘驱动程序
- K. 分析中断原因，进行中断处理
- L. 返回被中断的进程继续执行
- M. 把正常完成的信息通知磁盘驱动进程
- N. 把输入缓冲区中的第 N 号记录分离出来并传输到请求者进程的数据区
- O. 磁盘驱动进程自行阻塞，等待新的请求唤醒
- P. 唤醒请求者进程
- Q. 对输入的数据加工

答：(1) F\B\D\C\E (2) A\G\H\I (3) K\J\M\L (4) N\P\O (5) Q

## 一、进程同步与互斥

1、一个仓库，可以存放 A 和 B 两种产品，但要求：

- (1) 每次只能存入一种产品 (A 或 B)
- (2) A 产品数量-B 产品数量<M
- (3) B 产品数量-A 产品数量< N

其中 M、N 是正整数，试用 P、V 操作描述产品 A、B 的入库过程。

分析：A、B 之间存在同步、互斥的关系。且：若只放 A 产品，不放 B 产品，A 最多可放 M-1 次，每多放一个 B，根据条件 2，还可多放一个 A；若只放 B 产品，不放 A 产品，B 最多可放 N-1 次，每多放一个 A，根据条件 3，还可多放一个 B；

答：为此，设信号量 mutex 用于互斥，SA 表示可放入产品 A 的个数，SB 表示可放入产品 B 的个数。

begin

mutex, SA, SB: Semaphore ;

mutex : = 1; SA : = M-1; SB : =N-1;

Parbegin

Process PA

Begin

Loop:

P (SA);

P (mutex);

<A 入库> ;

V (mutex);

V (SB);

Goto loop ;

End;

Process PB

Begin

Loop:

P (SB);

P (mutex);

<B 入库> ;

V (mutex);

V (SA);

Goto loop ;

End;

Parend;

End;

2、设有 4 个进程 A、B、C、D 共享一个缓冲区，进程 A 负责循环地从文件读出一个整数并放入缓冲区，进程 B 从缓冲区中循环读入 MOD 3 为 0 的整数并累计求和，进程 C 从缓冲区中循环读入 MOD 3 为 1 的整数并累计求和，进程 D 从缓冲区中循环读入 MOD 3 为 2 的整数并累计求和。请用 P、V 操作写出能够正确执行的程序。

分析：本题主要考查信号量机制的应用。实际上是一个生产者和一个消费者通过单缓冲同步的推广，可描述为：进程 A 读入一个数，若这个数 MOD 3 为 0，那么通知 B 去取，信号量为 SB；若这个数 MOD 3 为 1，那么通知 C 去取，信号量为 SC；若这个数 MOD 3 为 2，那么通知 D 去取，信号量为 SD；

答：

begin

    Semaphore, SB, SC, SD: Semaphore = 1, 0, 0, 0;

    Num: integer;

    Parbegin

        Process PA

        Begin

        P (Semaphore);

        <从磁盘读入一个整数到 Num> ;

        <Num 放入缓冲区> ;

        if (Num MOD 3 = 0)

            V (SB);

        if (Num MOD 3 = 1)

            V (SC);

        if (Num MOD 3 = 2)

            V (SD);

        End;

    Process PB

    Begin

        P (SB);

        <从缓冲区中取数求累加和> ;

        V (Semaphore);

    End;

Process PC

Begin

P (SC);

<从缓冲区中取数求累加和> ;

V (Semaphore);

End;

Process PD

Begin

P (SD);

<从缓冲区中取数求累加和> ;

V (Semaphore);

```

    End;
Parent;
End;

```

3、假定系统有三个并发进程 read, move 和 print 共享缓冲器 B1 和 B2。进程 read 负责从输入设备上读信息，每读出一个记录后把它存放到缓冲器 B1 中。进程 move 从缓冲器 B1 中取出一记录，加工后存入缓冲器 B2。进程 print 将 B2 中的记录取出打印输出。缓冲器 B1 和 B2 每次只能存放一个记录。要求三个进程协调完成任务，使打印出来的与读入的记录个数，次序完全一样。 请用 PV 操作，写出它们的并发程序。

```

答: begin  SR, SM1, SM2, SP: semaphore;
    B1, B2: record;
    SR:=1; SM1:=0; SM2:=1; SP:=0      (2 分)
    cobegin
    process read                        (3 分)
    X: record;
    begin R: (接收来自输入设备上一个记录)
    X:=接收的一个记录;
    P(SR);
    B1:=X;
    V(SM1);
    goto R;
    end;
    Process move                        (3 分)
    Y: record;
    begin
    M: P(SM1);
    Y:=B1;
    V(SR)
    加工 Y
    P(SM2);
    B2:=Y;
    V(SP);
    goto M;
    end;
    Process print                       (2 分)
    Z: record;
    begin

```

```

P:P(SP);
Z:=B2;
V(SM2)
打印 Z
goto P;
end;
coend;
end;

```

## 二、进程调度算法：

4、假设一个计算机系统具有如下性能特征：处理一次中断，平均耗时 1ms；一次进程调度，平均耗时 2ms；将 CPU 分配给选中的进程，平均耗时 1ms；如果操作系统采用轮转法调度，且每 10 个时钟中断会产生一次进程调度，则 CPU 用于进程调度的比率为多少？再假设其定时器芯片每秒产生 100 次中断，则 CPU 用于时间中断处理的比率为多少？

答：（1）一秒钟内进行中断处理的时间为 100ms；一秒钟内进程调度的次数为 10 次，进程调度的耗时为 20 ms，一秒内进程切换的时间为 10 ms 则 CPU 用于进程调度的比率为  $(100+20+10)/1000=13\%$ ；

（2）定时器每秒产生 100 次中断，则至少需要进行中断处理的时间为 100ms，则 CPU 用于时间中断处理的比率  $=100\text{ms}/1\text{s}=10\%$ 。

5、某多道程序设计系统配有一台处理器和两台输入输出设备  $IO_1$  和  $IO_2$ 。现有优先数从高到低 3 个进程 P1、P2 和 P3 同时存在，他们使用资源的先后顺序和占用时间分别是：

进程 P1:  $IO_2(30\text{ms})$ , CPU(10ms),  $IO_1(30\text{ms})$ , CPU(10ms),  $IO_2(10\text{ms})$

进程 P2:  $IO_1(20\text{ms})$ , CPU(20ms),  $IO_2(40\text{ms})$

进程 P3: CPU(30ms),  $IO_1(20\text{ms})$

忽略调度等所需时间，回答下列问题：（1）进程调度采用非抢占式的优先数算法时 3 个进程从开始到完成所用的时间分别是多少？CPU 的利用率为多少？（2）进程调度采用可抢占的优先数算法时，3 个进程从开始到全部完成总共用了多少时间？CPU 的利用率为多少？

答：（1）三个进程从开始到完成所用的时间分别为：进程 P1 110ms；进程 P2 100ms；进程 P3 50ms（2 分）。处理器的利用率为  $(30+20+10+10)/110=63.6\%$ ；（2 分）

（2）3 个进程从开始到全部完成总共用了 100ms 时间（2 分），此时 CPU 的利用率为  $(20+10+10+10+10+10)/100=70\%$ 。（2 分）

## 三、作业调度

6、某系统采用短作业优先的调度算法。现有作业序列：作业 1（提交时间 8:00，运行时间 1.50）；作业 2（提交时间 8:30，运行时间 0.80）；作业 3（提交时间 9:00，运行时间 0.10）；作业 4（提交时间 9:30，运行时间 0.30）；单位：小时，以十进制计算，则其平均带权周转时间为？

答：按照短作业优先算法，作业的执行顺序是 1、3、4、2。则作业 1 的周转时间为：1.50，



作业 3 的周转时间为 0.60，作业 2 的周转时间为：0.60，作业 4 的周转时间为 2.40，所以带权周转时间为  $(1.5/1.5+2.4/0.8+0.6/0.1+0.6/0.3)/4=3.00$ 。

7、有 3 个作业：作业 1（提交时间 8：50，运行时间 1.50）；作业 2（提交时间 9：00，运行时间 0.40）；作业 3（提交时间 9：30，运行时间 1.00）；，当作业全部到达后，批处理单道系统按照相应比高者优先算法进行调度，则作业被选中执行的次序是？

答：作业 1 首先被选中运行，在 10：20 完成运行，此时计算作业 2 和 3 响应比，作业 2 的响应比为： $(80+24)/24$ （分钟）=4.333，作业 3 的响应比为： $(50+60)/60$ （分钟）=1.833。

## 四、磁盘调度

8、假设一个磁盘有 100 个柱面，编号为 0~99，在完成了磁道 25 处的请求后，磁头当前正在磁道 43 处服务。磁盘请求的柱面案 38、6、40、2、20、22、10 的次序到达磁盘驱动器，寻道时每移动一个柱面需要 10ms，计算以下算法的总寻道时间：（1）先来先服务算法（2）最短寻道时间优先算法（3）电梯调度算法。

答：（1）先来先服务算法：移动顺序为：43、38、6、40、2、20、22、10，共移动的柱面数为： $5+32+34+38+18+2+12=141$ ，总寻道时间为： $141*10ms=1410ms$ 。

（2）最短寻道时间优先算法：移动顺序为：43、40、38、22、20、10、6、2、，共移动的柱面数为： $3+2+16+2+10+4+4=31$ ，总寻道时间为： $31*10ms=310ms$ 。

（3）电梯调度算法：移动顺序为：43、40、38、22、20、10、6、2、，共移动的柱面数为： $3+2+16+2+10+4+4=31$ ，总寻道时间为： $31*10ms=310ms$ 。

9、某磁盘的旋转速度为 20 毫秒/圈，每个盘面被分成 10 个扇区，现有 10 个逻辑记录存放在同一磁道上，安排如下：

扇区号	逻辑记录	扇区号	逻辑记录
1	A	6	F
2	B	7	G
3	C	8	H
4	D	9	I
5	E	10	J

处理程序要顺序处理这些记录，每读出一个记录后处理程序要花 4 毫秒的时间进行处理，然后顺序读下一个记录并处理，直到处理完这些记录，回答：（1）顺序处理完这些记录总共花了多少时间？（2）请给一种记录优化分布的方案，使处理程序能在最短时间内处理完这些记录，并计算优化分布后处理所需时间。

答：（1）读一个记录花 2 毫秒，处理花 4 毫秒，顺序的下一个记录需等待 16 毫秒才能再次旋转到磁头下。所以，除第一个记录外，其它记录都需花 16 秒读出，总时间为 204 毫秒。

（4 分）（2）优化要求前一个记录处理完，下一个记录正在磁头下，所费时间最少。则，总时间仅为读出和处理记录时间之和，共 60 毫秒。（4 分）分布如下图所示：

扇区号	逻辑记录	扇区号	逻辑记录
1	A	6	F
2	H	7	C
3	E	8	J
4	B	9	G
5	I	10	D

## 五、死锁

10、考虑  $N$  个进程共享的具有  $M$  个同类资源的系统。证明：如果对  $I=1, 2, \dots, N$ ，其中  $N > 0$  且所有最大需求量之和小于  $M+N$ ，那么该系统是死锁无关的。

答：设所有进程对资源的总需求量是  $X$ ，每个进程  $I$  对资源的需求量是  $X_I$ ， $X_I$  在 0 到  $M$  之间。因此有  $X_1 + X_2 + \dots + X_N = X$ 。假设发生死锁，即每个参加资源竞争的进程都在申请最后一个资源，而此时系统中的资源已经全部分配完毕，即满足： $(X_1 - 1) + (X_2 - 1) + \dots + (X_N - 1) = M$ ，即  $X = M + N$ ，所以，当  $X \leq M + N$  时，系统是不会发生死锁的。

银行家算法

11、设系统中有 4 种资源  $R_1, R_2, R_3, R_4$  且可用资源量为  $(2, 1, 0, 0)$ ，有 5 个进程  $P_1, P_2, P_3, P_4, P_5$ ，当前最大需求量及各进程已得到的资源量如下表所示。

进程	已拥有量	总需求量
P1	(0, 0, 1, 2)	(0, 0, 1, 2)
P2	(2, 0, 0, 0)	(2, 7, 5, 0)
P3	(0, 0, 3, 4)	(6, 6, 5, 6)
P4	(2, 3, 5, 4)	(4, 3, 5, 6)
P5	(0, 3, 3, 2)	(0, 6, 5, 2)

问：（1）当前系统是否安全？（2）若进程  $P_3$  发出请求为  $(0, 1, 0, 0)$ ，系统能否将资源分配给它？

答：（1）每个进程仍需资源量如下： $P_1$  (0,0,0,0)、 $P_2$  (0,7,5,0)、 $P_3$  (6,6,2,2)、 $P_4$  (2,0,0,2)、 $P_5$  (0,3,2,0) 又可用资源为  $(2, 1, 0, 0)$ ，则存在安全序列  $P_1, P_4, P_5, P_2, P_3$ ，因此系统是安全的。（2）如果系统接受  $P_3$  的请求，那么系统的可用资源为  $(2, 0, 0, 0)$ 。即使  $P_1$  因运行结束释放了它的资源，此时可用资源为  $(2, 0, 1, 2)$ 。不能满足其他任意进程的最大需求，虽然死锁不一定会发生，但此时系统将处于不安全状态。

## 六、可变分区中对内存的分配算法

12、当某操作系统采用分区存储管理技术，操作系统在低地址占用了 100KB 的空间，用户区主存从 100KB 开始占用了 512KB。初始时，用户区全部为空闲，分配时截取空闲区的低地址部分作为已分配区。在执行了如下申请、释放操作序列后：req(300KB)、req(100KB)、release(300KB)、req(150KB)、req(50KB)、req(90KB)。（1）采用首次适用算法，主存中有哪些空闲区？要求画出主存分布图，并指出空闲区的首地址和大小。（2）采用最佳适用算法，主存中有哪些空闲区？要求画出主存分布图，并指出空闲区的首地址和大小。（3）若随后又要申请 80KB，针对上述两种情况会产生什么后果？说明了什么问题？

答：首次：空闲区：1：首地址 390KB，大小 10KB；2：500KB，大小 112KB

最佳：空闲区：1：首地址 340KB，大小 60KB；2：550KB，大小 62KB

若再申请 80KB，对首次算法可以分配成功，对最佳算法则不行。其结果说明最佳适配算法因倾向于深成大量没用的小空间，从而不仅慢于首次适配算法，而且导致比首次更多的内存浪费。

## 七、页面替换算法

13、现有一请求分页的虚拟存储器，内存最多容纳 4 个页面，对于下面的引用串：1, 2, 3, 4, 2, 1, 5, 6, 2, 1, 2, 3, 7, 6, 3, 2, 1, 2, 3, 6 应用下面的页面替换算法，计算各会出现多少次缺页中断。（1）LRU （2）FIFO （3）Optimal 注意，所给的页面初始均为空，因此，首次访问一页时就会发生缺页中断。

答：LRU 算法是选择淘汰最近一段时间内最久未使用的页面。内存块数为 4 时，缺页中断次数为 10 次。

页面引用序列	1	2	3	4	2	1	5	6	2	1	2	3	7	6	3	2	1	2	3	6
	1	2	3	4	2	1	5	6	2	1	2	3	7	6	3	2	1	2	3	6
		1	2	3	4	2	1	5	6	2	1	2	3	7	6	3	2	1	2	3
			1	2	3	4	2	1	5	6	6	2	2	3	7	6	3	3	2	2
				1	1	3	4	2	1	5	5	6	1	2	2	7	6	6	6	1
缺页	+	+	+	+			+	+				+	+	+			+			

FIFO 算法：内存块数为 4 的时候，缺页中断的次数为 14 次

页面引用序列	1	2	3	4	2	1	5	6	2	1	2	3	7	6	3	2	1	2	3	6
	1	2	3	4	4	4	5	6	2	1	1	3	7	6	6	2	1	1	3	3
		1	2	3	3	3	4	5	6	2	2	1	3	7	7	6	2	2	1	2
			1	2	2	2	3	4	5	6	6	2	1	3	3	7	6	6	2	2
				1	1	1	2	3	4	5	5	6	2	1	1	3	7	7	6	6
缺	+	+	+	+			+	+	+	+		+	+	+		+	+		+	

页																			
---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

OPT 算法：内存块数为 4 的时候，缺页中断的次数为 8 次

页面引用序列	1	2	3	4	2	1	5	6	2	1	2	3	7	6	3	2	1	2	3	6
	1	2	3	4	4	4	5	6	6	6	6	6	6	6	6	6	6	6	6	6
		1	2	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
			1	2	2	2	2	2	2	2	2	2	2	2	2	2	2	22	2	2
				1	1	1		1	1	1	1	1	7	7	7	7	1	1	1	1
缺页	+	+	+	+			+	+					+				+			

## Belady 现象

14、考虑如下页面走向：1, 2, 3, 4, 1, 2, 5, 1, 2, 3, 4, 5，假设系统采用 FIFO 算法，分析内存块数为 3 和 4 的情况下，缺页次数为多少？并以此说明什么是 Belady 现象。  
内存块数为 3，缺页次数为 9 次。

页面走向	1	2	3	4	1	2	5	1	2	3	4	5
	1	2	3	4	1	2	5	1	2	3	4	4
		1	2	3	4	1	2	2	2	5	3	3
			1	2	3	4	1	1	1	2	5	5
缺页	+	+	+	+	+	+	+			+	+	

内存块数为 4，缺页次数为 10 次。

页面走向	1	2	3	4	1	2	5	1	2	3	4	5
	1	2	3	4	1	2	5	1	2	3	4	5
		1	2	3	3	3	4	5	1	2	3	4
			1	2	2	2	3	4	5	1	2	3
				1	1	1	2	3	4	5	1	2
缺页	+	+	+	+			+	+	+	+	+	+

说明：在进程没有得到所需的全部页面时，如果采用 FIFO 算法，就有可能出现：随着分配给用户的页面数上涨，缺页中断次数也随之增加的现象，称为：Belady 现象。

## 八、地址变换

15、假定某页式管理系统，主存为 64KB，分成 16 块，块号为 0、1、2……15。设某作业有 4 页，其页号为 0、1、2、3，被分别装入主存的 2、4、1、6 块，试问：（1）该作业的总长度是多少字节？（按十进制）（2）写出该作业每一页在主存中的起始地址。（3）若给出逻辑地址 [0, 100]、[1, 50] [2, 0] [3, 60]，请计算相应的内存地址。

答：（1）每块的长度为  $64\text{KB}/16=4\text{KB}$ ，作业的总长度为 16KB。

（2）PMT 如下表所示：

页号	块号
0	2
1	4
2	1
3	6

所以该作业的：第 0 页在主存中的起始地址为 8K；第 1 页在主存中的起始地址为 16K；第 2 页在主存中的起始地址为 4K；第 3 页在主存中的起始地址为 24K。

(3)逻辑地址[ 0, 100]的内存地址为  $4K*2+100=8292$ ；逻辑地址 [ 1, 50]的内存地址为  $4K*4+50=16434$ ；逻辑地址 [ 2, 0]的内存地址为  $4K*1+0=4096$ ；逻辑地址 [ 3, 60]的内存地址为  $4K*3+60=24636$ ；

16、某用户文件共 10 个逻辑记录，每个逻辑记录的长度为 480 个字符，现把该文件存放到磁带上，若磁带的记录密度为 800 字符/英寸，块与块之间的间隙为 0.6 英寸，回答下列问题：(1) 不采用记录成组操作时磁空间的利用率为多少。(2) 采用记录成组操作且块因子为 5 时，磁带空间的利用率为多少。(3) 当按上述方式把文件存放到磁带上后，用户要求每次读一个逻辑记录存放到他的工作区。当对该记录处理后，又要求把下一个逻辑记录读入他的工作区，直至 10 个逻辑记录处理结束。系统应如何为用户服务？

答：(1) 利用率为 50% (3 分) (2) 利用率为 83% (3 分) (3) 设置长度为 2400 字符的主存缓冲区；找到该文件的存放位置，启动磁带机读出第一块内容存入主存缓冲区；进行记录分解，按用户要求依次把主存缓冲区中的五个记录传送到用户工作区；启动磁带机读第二块内容存入主存缓冲区，把第 6 至 10 个逻辑记录按用户要求依次传送到用户工作区。(2 分)

17、页式存储管理中，主存按页分配，可用一张“位示图”构成主存分配表。假设主存的容量为 2M 字节，页面长度为 512 字节，若用字长为 32 位的字作主存分配的“位示图”需要多少字？如页号从 1 开始，字号和字内位号（从高位到低位）均从 0 开始，试问第 2999 页对应与何字何位；99 字 19 位又对应于第几页？

答：128 个字（字数  $= 2^{21} \div 2^9 \div 2^5 = 128$ ）(2 分)  
 93 字 22 位（字  $= 2999 \text{DIV} 32 = 93$ ；位号  $= 2999 \text{MOD} 32 - 1 = 22$ ）(3 分)  
 第 3188 页（页号  $= 99 \times 32 + 19 + 1 = 3158 + 20 = 3188$ ）(3 分)

18、一个有快表的请页式虚存系统，设主存访问周期为 1 微秒，内外存传送一个页面的平均时间为 5 毫秒。如果快表的命中率为 75%，缺页中断率为 10%。忽略快表访问时间，试求主存的有效存取时间。

答：快表的命中率为 75%，缺页中断率为 10%，所以，主存命中率为 15%。故主存的有效存取时间  $= 1 * 75\% + 2 * 15\% + (5000 + 2) * 10\% = 501.25$  微秒。

19、虚存系统，主存访问时间是 2ms，平均缺页中断处理时间 25ms，缺页中断率为 5%，试计算平均有效访问时间是多少？

答： $4\text{ms} \times (1-5\%) + 32\text{ms} \times 5\% = 5.35\text{ms}$

## 九、工作集 (WORKING SET, WS)

20、进程工作集指“在某一段时间间隔内进程运行所需访问的页面集合”。用  $W(t, \Delta)$  表示在时刻  $t-\Delta$  到时刻  $t$  之间所访问的页面集合，它就是进程在时刻  $t$  的工作集。

- 实现思想：

工作集模型用来对局部最佳页面替换算法进行模拟实现，不向前查看页面引用串，而是基于程序局部性原理向后看，在任何给定时刻，一个进程不久的将来所需主存页框数，可通过考查其过去最近的时间内的主存需求做出估计。

例：

设进程的页面引用串为 P3, P3, P4, P2, P3, P5, P3, P5, P1, P4, 工作集窗口尺寸  $\Delta=3$ ,  $t=0$  时刻初始工作集为 (P1, P4, P5) (表示 P1 在  $t=0$  时刻被引用, P4 在  $t=-1$  时刻, P5 在  $t=-2$  时刻)

时刻t	0	1	2	3	4	5	6	7	8	9	10
引用串	p1	p3	p3	p4	p2	p3	p5	p3	p5	p1	p4
p1	✓	✓	✓	✓	—	—	—	—	—	✓	✓
p2	—	—	—	—	✓	✓	✓	✓	—	—	—
p3	—	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
p4	✓	✓	✓	✓	✓	✓	✓	—	—	—	✓
p5	✓	✓	—	—	—	—	✓	✓	✓	✓	✓
In t		p3			p2		p5			p1	p4
OUT t			p5		p1			p4	p2		

此算法总的缺页次数为 5 次，工作集尺寸在 2~4 个页框之间波动。

十、旋转型存储设备上信息的优化分布能减少输入输出服务的总时间，例如，有 10 个记录 A、B、……J 存放在某磁盘的一磁道上，假定这个磁道划分为十块，每块存放一个记录，如图。则：顺序处理这些记录，若磁盘旋转速度为  $20\text{ms/r}$ ，处理程序每读出一个记录后花  $4\text{ms}$  进行处理。问：处理完这 10 个记录的总时间是多少？为缩短处理的总时间应进行怎样的安排以优化其分布？并计算优化处理后的总时间。

块号	1	2	3	4	5	6	7	8	9	10
记录号	A	B	C	D	E	F	G	H	I	J

十一、3. 设内存中有三道程序 A、B、C，它们按 A、B、C 的优先次序执行。它们的计算和 I/O 操作的时间如表所示（单位：ms）。

程序 操作	A	B	C
计算	30	60	20
I/O 操作	40	30	40
计算	10	10	20

假设三道程序使用相同设备进行 I/O 操作，即程序以串行方式使用设备，试画出单道运行和多道运行的时间关系图（调度程序的执行时间忽略不计）。在两种情况下，完成这三道程序各要花多少时间？

十二、2. 有一计算机系统利用如下图所示的位示图(行号、列号都从 1 开始编号)管理磁盘空间，并且在该位示图中，“0”表示对应的盘块空闲，“1”表示对应的盘块已分配。如果盘块从 1 开始编号，每个盘块的大小为 1KB。

1) 现要为文件分配一个盘块，试说明分配过程。

2) 若要释放磁盘的第 200 块，应如何处理？

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
3	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1
4	1	1	1	1	1	1	0	1	1	1	1	0	1	1	1	1
5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6																
...																
16																

### 操作系统应用题的范围

- 1、进程的状态变换
- 2、用 P、V 操作解决进程的互斥和同步问题
- 3、银行家算法、RAG 图的化简
- 4、作业调度、进程调度
- 5、逻辑地址到物理地址的转换
- 6、页面淘汰置换算法
- 7、可变分区的内存分配
- 8、磁臂调度算法
- 9、磁盘空间管理（位示图）
- 10、记录存储优化
- 11、单道、多道程序运行时间