操作系统的目标

● **设置操作系统的目的：**

1、方便性:操作系统使计算机更易于使用

2、有效性:操作系统允许以更有效的方式使用计算机系统资源。

3、可扩充性:在操作系统中,允许有效地开发，测试和引进新的系统功能。

4、开放性:实现应用程序的可移植性和互操作性,要求具有统一的开放的环境。

OS 的作用

●用户和计算机硬件之间的接口

●操作系统作为计算机系统资源管理者。

●os 实现了对计算机资源的抽象

操作系统的功能：

1、处理机管理：分配和控制 CPU。

2、存储器管理：内存分配与回收。

3、I/O 设备管理：I/O 设备的分配与操纵。

4、文件管理：文件的存取、共享和保护。

5、用户接口：命令接口与程序接口。

●操作系统用作扩充机器功能，使其成为便于使用的机器，这种机器又称为虚拟机。

二、操 作 系 统 的 发 展

一．无操作系统时的计算机系统

1、 人工操作方式

一台计算机的所有资源由用户独占，降低了计算机资源利用率，人操作慢，出现了严重的人机矛盾。

缺点：1：用户独占全机 2：cpu 等待人工操作

2、 脱机输入输出方式

- **在外围计算机的控制下，实现输入输出。**
- **主要解决了 CPU 与设备之间不匹配的矛盾。**

二.单道批处理系统

1、在内存中仅存一道作业运行，运行结束或出错，才自动调另一道作业运行。

2、单道批处理系统主要特征：自动性、顺序性、单道性。

3、单道批处理系统主要优点：减少人工操作，解决了作业的自动续接。

4、单道批处理系统主要缺点：平均周转时间长，没有交互能力。

三.多道批处理系统

（一）多道程序的概念：

在内存中存放多道作业运行，运行结束或出错，自动调度内存中的另一道作业运行。

●多道程序带来的好处：

1、提高 CPU 的利用率。

2、提高内存和 I/O 设备利用率。

3、增加系统吞吐率。

（二）多道批处理系统主要特征：

多道性、宏观上并行、微观上串行。

（三）多道批处理的主要优点：提高了资源利用率和吞吐能力。

多道批处理的主要缺点：平均周转时间长，没有交互能力。

设计批处理多道系统时，首先要考虑的是 ：系统效率和吞吐量

四.分时系统

（一）分时系统的产生

用户需要：人-机交互、共享主机、便于用户上机

（二）分时系统实现的方法

具有“前台”和“后台”的分时系统

多道分时系统

（三）分时系统实现中的关键问题：

及时接收：实现多个用户的信息及时接收。

及时处理：及时控制作业的运行。

(1）作业应直接进入内存

(2）不允许一个作业长期占用处理机直至他运行结束或发生 I/O 请求后，才调度其他作业运行。

（四）分时系统的特征：

多路性：多个用户分时使用一台计算机，宏观上同时，微观上轮流。

独立性：独立运行，不混淆，不破坏。

及时性：系统能在很短的时间得到回答。

交互性：能实现人机对话。

（五）两个概念

时间片

响应时间

五.实时系统

● 所谓实时系统：是计算机及时响应外部事件的请求，在规定的时间内完成对该事件的处理，并控制所有实时设备和实时任务协调一致的运行。

（一）实时系统分为两类

1、实时控制系统

2、实时信息处理系统

例如：用于控制生产流水线，进行工业处理控制的操作系统是实时系统

（二）实时系统的特征

1、多路性：能对多个对象进行控制。

2、独立性：独立运行，不混淆，不破坏。

3、交互性：仅限于访问系统中某些特定的专用服务程序。

4、可靠性：高可靠性，应具有过载防护能力。

5、及时性：不同的系统要求不一样，控制对象必须在截止时间内完成。

三、操作系统的基本特征

- 现代 OS 的四个基本特征：
  1、并发
  2、共享
  3、虚拟
  4、异步
- 并发是最重要的特征，其它特征都以并发为前提。

1.并发

- 并发——并行性和并发性

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

2.共享

- 所谓共享是指系统中的资源可供内存中多个并发执行的进程共同使用。
- 1、互斥共享方式：

- 把在一段时间内只允许一个进程访问的资源，称为临界资源。
- 系统中的临界资源可以提供给多个进程使用，但一次仅允许一个进程使用，称为互斥共享方式。

- 2、同时访问方式：

- 从宏观上看，资源共享是指多个任务可以同时使用系统中的软硬件资源
- 从微观上看，资源共享是指多个任务可以交替互斥地使用系统中的某个资源。例如磁盘。

3.虚拟

- 所谓虚拟是指通过某种技术把一个物理实体变为若干个逻辑上的对应物。
- 虚拟处理机：分时实现
- 虚拟设备：SPOOLING 技术
- 虚拟存储器：虚拟存储管理实现

4.异步性

- 异步性—— 是指进程以异步的方式执行，进程是以人们不可预知的速度向前推进。

内存中每个进程在何时能获得处理机运行，何时又因提出资源请求而暂停，以及进程以怎样的速度向前推进，每到程序总共用多长时间才能完成等等，这些都是不可预知的。

第二章　　进程管理

一、相关概念

1、**前趋图**——有向无循环的图。

表示程序执行的偏序关系。

2、**程序的顺序执行**——严格按照程序给定的顺序执行，仅当前一个执行结束才执行后一个。

3、**程序的顺序的特征：**

① 顺序性
② 封闭性
③ 可再现性

4、**程序的并发执行**——是指两个或两个以上程序段在执行的时间上是重叠的，即使这种重叠只有一小部分，则称这些程序为并发执行。

5、**程序并发执行的特征：**

① 间断性
② 失去封闭性
③ 不可再现性

二、进程的基本概念

1、**进程的定义**——一个具有一定独立功能的程序在一个数据集合上的一次动态执行过程，是系统进行资源分配和调度的独立单位.

2、**进程的特征**

① 动态性：是最基本的特征。因为进程的实质是程序的一次执行过程，而且该特征还表现在进程由创建而产生，由调度而执行，由撤销而消亡，即进程具有一定的生命期。

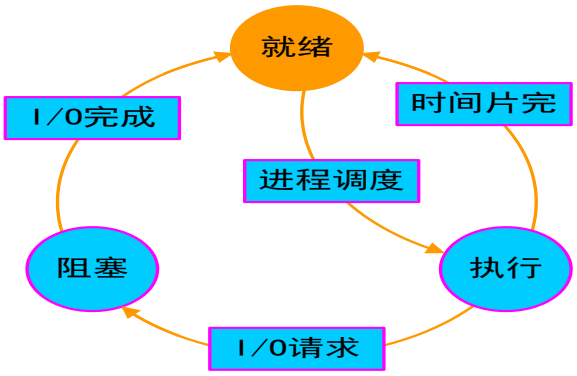
② 并发性

③ 独立性

④ 异步性

⑤ 结构特征

3、进程的三种基本状态及其转换原因



(1) 就绪态变为执行态

调度程序选择一个新的进程分配给其处理机CPU

(2) 执行态变为就绪态

执行进程用完了时间片

执行进程被中断，因为一高优先级进程处于就绪状态（系统原因）

(3) 执行态变为阻塞态

执行进程发生I/O请求或等待外部事件发生（自身原因）

(4) 阻塞态变为就绪态

当I/O完成或所等待的事件发生时；

4、 进程控制块——描述和控制进程运行，系统为每个进程定义的一个数据结构。

PCB 是进程存在的惟一标志。

PCB 常驻内存。

5、 进程控制块的组织方式

链接方式、索引方式

6. 进程与程序的区别

（1）程序是静态的，进程是动态的；

程序是有序代码的集合；进程是程序的执行。

（2）进程是暂时的，程序的永久的：

进程进程只是一次执行过程，有生命周期，有诞生有消亡，是暂时的；而程序可作为软件资源长期保存，相对长久的；

（3）进程与程序的组成不同：

进程是由程序、数据和 PCB 三部分组成的.

（4）进程与程序的对应关系：

通过多次执行，一个程序可对应多个进程；通过调用，一个进程可包括多个程序。

三、进程控制

1、进程管理

进程图：表明进程的创建关系，创建的进程和被创建的进程可以并发执行。

2、引起进程创建的原因

① 用户登录：为终端用户建立进程。

② 作业调度：选中的作业建立进程。

③ 提供服务：为用户提供的服务进程。

例如：I/O 进程等。

④ 应用请求：应用程序自己创建的进程。

3、原语：由若干条指令构成，用于完成一定功能的一个过程。

4、原子操作（原子性）：一个操作中的所有动作，要么全做，要么全不做。是一个不可分割的操作。

5、 线程的基本概念

（1） 线程：一个被调度和分派的基本单位并可独立运行的实体。

（2） 线程分类：

① 内核支持线程：依赖于内核进行控制和管理。

② 用户级线程：在用户级创建、撤消和切换。

（3） 在引入线程的 OS 系统中，则把线程作为调度和分派的基本单位，而把进程作为资源的拥有的基本单位。

（4） 在同一进程中的线程切换不会引起进程切换。

（5） 在不同进程中的线程切换会引起进程切换。

6.线程与进程的比较

1.调度性

在传统的操作系统中，作为拥有资源的基本单位和独立调度、分派的基本单位都是进程；在引入线程的 OS 系统中，则把线程作为调度和分派的基本单位，而把进程作为资源的拥有的基本单位。

并发性

引入线程后的 os 中，不仅进程之间可以并发执行，而且在一个进程的多个线程之间也可以并发执行，使 os 具有更好的并发性，从而能更有效地使用系统资源和提高系统吞吐量 。

2.拥有资源

不论是传统的操作系统，还是设有线程的操作系统，进程都是拥有资源的一个独立单位，它可以拥有自己的资源。一般地说，线程自己不拥有系统资源（也有一点必不可少的资源），但它可以访问其隶属进程的资源

系统开销

创建、撤销以至于切换进程时付出的系统开销显著大于创建、撤销、切换线程时的系统开销

进程同步的基本概念

1、 进程的相互制约

① 间接相互制约（进程互斥）——若干进程共享一资源时，任何时刻只允许一个进程使用。

② 直接相互制约（进程同步）——并发进程之间存在的相互制约和相互依赖的关系。

2、 临界资源：一次仅允许一个进程使用的资源称为临界资源。

3、 临界区：访问临界资源的那段代码称为临界区。

4、 同步机制应遵循的准则：

① 空闲让进 —— 充分利用资源

② 忙则等待 —— 保证同步与互斥

③ 有限等待 —— 防止陷入“死等”

④ 让权等待 —— 防止陷入“忙等”

信号量机制

1）信号量

信号量是一种只能进行 P 操作和 V 操作的特殊变量

2) 信号量的物理含义：

S>0 表示有 S 个资源可用.
S=0 表示无资源可用.
S<0 则| S |表示 S 等待队列中的进程个数.
P(S)：表示申请一个资源.
V(S)：表示释放一个资源.
思考：对于 N 个并发进程，信号量的取值范围是什么，有什么含义？
互斥信号量的值仅取 1、0、-1……、1-N 等:
①若 mutex=1，表示没有进程进入临界区；
②若 mutex=0，表示有一个进程进入临界区；
■ ③若 mutex=－1，表示一个进程进入临界区，另一个进程等待进入.
…….
④若 mutex=1-N，表示一个进程进入临界区，另有 N-1 个进程等待

- 进程同步应用示例讲解
  - 水果盘问题
- 题目：
  - 桌上有一个盘子，可以存放一个水果。父亲总是把苹果放在盘子中，母亲总是把香蕉放在盘子中；一个儿子专等吃盘中的香蕉，一个女儿专等吃盘中的苹果。

问题：
1）系统要设几个进程来完成这个任务？各自的工作是什么？
2）这些进程间有什么样的相互制约关系?
3）用 P，V 操作写出这些进程的同步算法(注：标明信号量的含义)。

- 1）需要四个进程
- 进程描述：
  - Father：父亲放置苹果的进程；
  - Mother：母亲放置香蕉的进程；
  - Son：儿子吃香蕉的进程；
  - Daughter：女儿吃苹果的进程。
- 分析：
  - 四人公用一个盘子；
  - 盘子每次只能放一个水果，当盘子为空时，父母均可尝试放水果，但一次只能有一人成功；
  - 盘中是香蕉，儿子吃，女儿等；
  - 盘中是苹果，女儿吃，儿子等。

- 2）进程之间既有互斥又有同步关系。
- Father 进程和 Mother 进程要互斥的向盘中放水果，应设置一互斥信号量 dish，初值为 1，表示盘子为空；
- Father 进程要设置同步信号量 apple，用于给 Daughter 进程传送消息，初值为 0，表示还没有消息产生，即没有放苹果；相应 Daughter 进程也要向父、母进程传送盘子为空的消息。
- Mother 进程要设置同步信号量 banana，用于给 Son 进程传送消息，初值为 0，表示还没有消息产生，即没有放香蕉。相应 Son 进程也要向父、母进程传送盘子为空的消息。
- 3）变量设置：
  - dish：表示盘子是否为空，初值=1；
  - apple：表示盘中是否有苹果，初值=0；
  - banana：表示盘中是否有香蕉，初值=0。

Father:
P (dish);//判断盘子是否空
将苹果放入盘中；
V (apple);//传消息给女儿进程
Mather:
P (dish);//判断盘子是否空
将香蕉放入盘中；
V (banana);//传消息给儿子进程
Son:
P (banana);//判断是否有香蕉
从盘中取出香蕉；
V (dish);//传送盘空消息
吃香蕉；
Daughter:
P (apple);//判断是否有苹果
从盘中取出苹果；
V (dish);//传送盘空消息
吃苹果；

- 题目：

生产者与消费者共用一个有 n 个缓冲单元的缓冲区，生产者向缓冲区中放产品，消费者从缓冲区中取产品消费。
问题：
1）系统要设几个进程来完成这个任务？各自的工作是什么？
2）这些进程间有什么样的相互制约关系?
3）用 P，V 操作写出这些进程的同步算法(注：标明信号量的含义)。

1）系统需设两个进程

producer 进程 ：生产产品并向缓冲区中放；

consumer 进程 ：从缓冲区取产品并消费；

- 1）分析：
  - 生产者与消费者共用一个缓冲区，但有 n 个缓冲单元；
  - 对于缓冲区同一时刻只能有一个进程使用；
  - 有空缓冲单元时，生产者可以放，否则等待；
  - 有满缓冲单元时，消费者可以取，否则等待。
- 2）两进程间有互斥和同步的关系

producer 进程和 consumer 进程要互斥的访问缓冲区，应设置一互斥信号量 mutex，初值为 1。

producer 进程和 consumer 进程有同步的关系，当没有满缓冲单元时，consumer 进程不能消费，当没有空缓冲单元时， producer 进程不能向缓冲区放。应为他们设置同步信号量 empty=n 和 full=0，用于互传消息。当 empty=0 时不能生产，当 full=0 时不能消费。

3）变量设置

mutex：保证对缓冲区的互斥访问，初值=1；

empty：表示缓冲区中是否有空缓冲单元，初值=n；

full： 表示缓冲区中是否有满缓冲单元，初值=0。

```
producer:
begin
    生产一个成品;
    P（empty）;
    P（mutex）;
    将产品存入缓冲区;
    V（mutex）;
    V（full）;
End
```

```
consumer:
begin
    P（full）;
    P（mutex）;
    将产品从缓冲区取出;
    V（mutex）;
    V（empty）;
    消费成品;
end
```

- 题目：

设有进程 A、B、C 分别对单缓冲区 S 和 T 进行操作，其中 A 进程负责从卡片上读取数据并把数据块输入到缓冲区 S，B 进程负责从缓冲区 S 中提出数据块并将数据块送到缓冲区 T 中，C 进程负责从缓冲区 T 中取出信息打印。

问题：

1）这些进程间有什么样的相互制约关系?

2）用 P，V 操作写出这些进程的同步算法(注：标明信号量的含义)。

- 分析：两个阶段的生产者和消费者问题。

只有 S 为空时，A 进程才能向 S 中放数据；
只有 S 中有数据时，B 进程才能取数据；
只有 T 为空时，B 进程才能向 T 中放数据；
只有 T 中有数据时，C 进程才能取数据。

- 1）三进程间有互斥和同步的关系

A 进程和 B 进程要互斥的访问缓冲区 S，B 进程和 C 进程要互斥的访问缓冲区 T。

A 进程和 B 进程有同步的关系，当缓冲区 S 空时，A 进程才能向 S 中放数据，只有 S 中有数据时，B 进程才能取数据。应为他们设置同步信号量 emptyS=1 和 fullS=0，用于互传消息。

B 进程和 C 进程有同步的关系，只有当缓冲区 T 空时，B 进程才能向 T 中放数据，只有 T 中有数据时，C 进程才能取数据。应为他们设置同步信号量 emptyT=1 和 fullT=0，用于互传消息。

3）变量设置

emptyS ：表示单缓冲区 S 是否为空，初值为 1；
emptyT ：表示单缓冲区 T 是否为空，初值为 1；
fullS ：表示单缓冲区 S 是否有数据可处理，其初值 0
fullT ： 表示单缓冲区 T 是否有数据可处理，其初值 0

```
A: begin
   repeat
   读卡片
   P(emptyS );
   将数据送入S;
   V(fullS ) ;
   until false;
    end
```

```
B: begin
   repeat
   P(fullS );
   从S中取数据;
   V(emptyS) ;
    ......
   P(emptyT);
   将数据送入T;
   V(fullT ) ;
   until false;
   end
```

```
C: begin
   repeat
   P(fullT);
   从T中取数据;
   V(emptyT) ;
   打印数据;
    until false;
   end
```

进程通信

进程通信的类型：低级通信和高级通信

（1）高级通信方式：

① 共享存储器系统：共享数据结构、共享存储器区

② 消息传递系统：

直接通信方式——通过收发原语 Send，Receive

间接通信方式——通过信箱实现信息交换

③ 管道通信

（2）管道通信具有三方面的协调能力：

①双方同时存在 ②同步关系 ③互斥使用管道

第三章　处理机　调度与死锁

一、调度的类型（三个层次）

1、 高级调度（作业调度）

2、 低级调度（进程调度）：

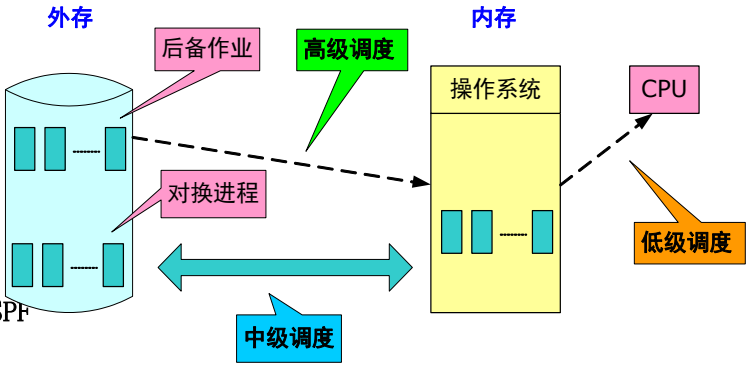
非抢占式、抢占式

抢占式：时间片原则、优先权原则、

短作业/进程优先原则

3、 中级调度（对换）

高级、中级和低级调度



二、调度的算法

1、 先来先服务调度的算法 FCFS

2、 短作业/进程优先调度的算法 SJF/SPF

3、 时间片轮转调度的算法（分时）RR

4、 优先权调度的算法：非抢占式、抢占式

① 静态优先权

② 动态优先权（高响应比）

5、 响应比高者优先调度的算法 HRN：既考虑作业等待时间，又考虑作业执行时间 。

RP=1+等待时间/ 服务时间

举例

设在批处理系统中有 4 道作业，他们进入系统的时刻及运行时间如下：设系统每次只选择一个作业，分别给出下列算法中这组作业的平均周转时间和带权周转时间：FCFS、SJF、HRN.

作业号	进入时刻	运行时间
1	8.00	2.00
2	8.50	0.50
3	9.00	0.10
4	9.50	0.20

- 完成时间＝开始时间＋服务时间
- 周转时间＝完成时间－到达时间
- 带权周转时间＝周转时间/服务时间

FCFS

作业名	提交时刻	要求运行时间	开始运行时刻	完成时刻	Ti	Wi
1	8.00	2.00	8.00	10.00	2	1
2	8.50	0.5	10.00	10.5	2	4
3	9.00	0.1	10.5	10.6	1.6	16
4	9.50	0.2	10.6	10.8	1.3	6.5

平均周转时间=（2+2+1.6+1.3）/4

平均带权周转时间=(1+4+16+6.5)/4

SJF

作业名	提交时刻	要求运行时间	开始运行时刻	完成时刻	Ti	Wi
1	8.00	2.00	8.00	10.00	2	1
3	9.00	0.1	10.00	10.1	1.1	11
4	9.50	0.2	10.1	10.3	0.8	4
2	8.50	0.5	10.3	10.8	2.3	4.6

HRN

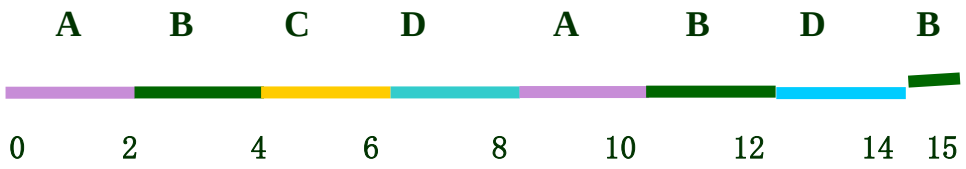
作业名	提交时刻	要求运行时间	开始运行时刻	完成时刻	Ti	Wi
1	8.00	2.00	8.00	10.00	2	1
3	9.00	0.1	10.00	10.1	1.1	11
2	8.50	0.5	10.1	10.6	2.1	4.2
4	9.50	0.2	10.6	10.8	1.3	6.5

T=6.5/4　　W=22.7/4

RR(时间片轮转)

时间片=2

进程名	到达时间	服务时间	开始时间	完成时间	周转时间	带权周转时间
A	0	4	0	10	10	2.5
B	0	5	2	15	15	3
C	0	2	4	6	6	3
D	0	4	6	14	14	3.7



死锁的基本概念

1、 何谓死锁？ （见P103）死锁是指多个进程因竞争资源而造成的一种僵局，当进程处于这种僵持状态时，若无外力作用，他们都无法再向前推进。

2、 产生死锁原因：

① 竞争资源

② 推进顺序不当

3、 产生死锁的必要条件

① 互斥

② 请求和保持

③ 不剥夺

④ 环路等待条件

4、 处理死锁的基本方法

①预防死锁：设置某些限制条件，破坏四个条件。

②避免死锁：资源动态分配，防止进入不安全状态。（银行家算法）

③检测死锁：设置检查机构，定时检查系统是否出现死锁。

④解除死锁：已出现死锁时。

银行家算法练习

①银行家算法中的数据结构

- Available：可利用资源向量

- 这是一个含有 m 个元素的数组，其中的每一个元素代表一类可利用的资源数目；
- 初始值是系统中所配置的该类全部可用资源的数目，其数值随该类资源的分配和回收而动态地改变；
- 如果 Available [j] =K，则表示系统中现有 Rj 类资源 K 个。

- **Max：最大需求矩阵**
  - 这是一个 n×m 的矩阵，它定义了系统中 n 个进程中的每一个进程对 m 类资源的最大需求；
  - 如果 Max [i,j] =K，则表示进程 i 需要 Rj 类资源的最大数目为 K。
- **Allocation：分配矩阵**
  - 这也是一个 n×m 的矩阵，它定义了系统中每一类资源当前已分配给每一进程的资源数；
  - 如果 Allocation [i,j] =K，则表示进程 i 当前已分得 Rj 类资源的数目为 K。
- **Need：需求矩阵**
  - 这也是一个 n×m 的矩阵，用以表示每一个进程尚需的各类资源数；
  - 如果 Need [i,j] =K，则表示进程 i 还需要 Rj 类资源 K 个，方能完成其任务；
  - Need [i,j] =Max [i,j] -Allocation [i,j]
- **Available：可利用资源向量**
- **Max：最大需求矩阵**
- **Allocation：分配矩阵**
- **Need：需求矩阵**
- **②银行家算法**
- 假设：
  - Requesti 是进程 Pi 的请求向量，如果 Requesti [j] =K，表示进程 Pi 需要 K 个 Rj 类型的资源。当 Pi 发出资源请求后，系统按下述步骤进行检查：
- 步骤一：
  - 如果 Requesti [j] ≤Need [i,j]，便转向步骤二；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。
- 步骤二：
  - 如果 Requesti [j] ≤Available [j]，便转向步骤三；否则， 表示尚无足够资源，Pi 须等待。
- 步骤三：
  - 系统试探着把资源分配给进程 Pi，并修改下面数据结构中的数值：■
  - Available [j] ：=Available [j] -Requesti [j] ;■
  - Allocation [i,j] ：=Allocation [i,j] +Requesti [j]
  - Need [i,j] ：=Need [i,j] -Requesti [j]
- 步骤四：
  - 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，才正式将资源分配给进程 Pi，以完成本次分配；否则， 将本次的试探分配作废，恢复原来的资源分配状态，让进程 Pi 等待。

③安全性算法

- **步骤一：设置两个向量**

- 工作向量 Work：它表示系统可提供给进程继续运行所需的各类资源数目，它含有 *m* 个元素，在执行安全算法开始时，Work：=Available;
- 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; 当有足够资源分配给进程时， 再令 Finish［i］：=true。

▪ **步骤二：**
- 从进程集合中找到一个能满足下述条件的进程：
- ① Finish［i］=false; ② Need［i,j］≤Work［j］；
- 若找到， 执行步骤三， 否则，执行步骤四；

▪ **步骤三：**
- 当进程 Pi 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，故应执行：■
  - Work［j］：= Work［i］+Allocation［i,j］；
  - Finish［i］：= true;■
  - go to step 2;

▪ **步骤四**
- 如果所有进程的 Finish［i］=true 都满足， 则表示系统处于安全状态；否则，系统处于不安全状态。

④银行家算法实例之一

▪ **有三类资源 A(17)、B(5)、C(20)。有 5 个进程 P1—P5。T0 时刻系统状态如下页表所示：**

▪ **问：**

- **⑴T0 时刻是否为安全状态，给出安全系列。**
- **⑵T0 时刻，P2：Request(0,3,4)，能否分配，为什么？**
- **⑶在⑵的基础上 P4：Request(2,0,1)，能否分配，为什么？**
- **⑷在⑶的基础上 P1：Request(0,2,0)，能否分配，为什么？**

T0 时刻的系统状态

进程标识	最大需求	已分配
P1	5 5 9	2 1 2
P2	5 3 6	4 0 2
P3	4 0 11	4 0 5
P4	4 2 5	2 0 4
P5	4 2 4	3 1 4

解：(1) T0 时刻的安全系列

先求出 Need 和 Work

资源情况 进程	最大需求	已分配	Need
	A B C	A B C	A B C
P1	5 5 9	2 1 2	3 4 7
P2	5 3 6	4 0 2	1 3 4

P3	4 0 11	4 0 5	0 0 6
P4	4 2 5	2 0 4	2 2 1
P5	4 2 4	3 1 4	1 1 0

A(17)、B(5)、C(20)

Work=2 3 3

	Work	Allocation	Need	W+A	Finish
P5	2 3 3	3 1 4	1 1 0	5 4 7	T
P4	5 4 7	2 0 4	2 2 1	7 4 11	T
P3	7 4 11	4 0 5	0 0 6	11 4 16	T
P2	11 4 16	4 0 2	1 3 4	15 4 18	T
P1	15 4 18	2 1 2	3 4 7	17 5 20	T

(2) P2: Request(0,3,4)

因：（ Available =2 3 3）< Request(0,3,4) 所以不能分配

(3) P4：Request(2,0,1) Available =2 3 3

	Allocation	Need	Available
P1	2 1 2	3 4 7	2 3 3
P2	4 0 2	1 3 4	
P3	4 0 5	0 0 6	
P4	2 0 4	2 2 1	
P5	3 1 4	1 1 0	

	Allocation	Need	Available
P1	2 1 2	3 4 7	0 3 2
P2	4 0 2	1 3 4	
P3	4 0 5	0 0 6	
P4	4 0 5	0 2 0	
P5	3 1 4	1 1 0	

(3) P4：Request(2,0,1)

有安全序列

P4 P5 P3 P2 P1 可以分配

	Work	Allocation	Need	W+A	Finish
P4	0 3 2	4 0 5	0 2 0	4 3 7	T
P5	4 3 7	3 1 4	1 1 0	7 4 11	T
P3	7 4 11	4 0 5	0 0 6	11 4 16	T
P2	11 4 16	4 0 2	1 3 4	15 4 18	T
P1	15 4 18	2 1 2	3 4 7	17 5 20	T

(4) P1：Request(0,2,0)

	Allocation	Need	Available

P1	2 1 2	3 4 7	0 3 2
P2	4 0 2	1 3 4	
P3	4 0 5	0 0 6	
P4	4 0 5	0 2 0	
P5	3 1 4	1 1 0	

	Allocation	Need	Available
P1	2 3 2	3 2 7	0 1 2
P2	4 0 2	1 3 4	
P3	4 0 5	0 0 6	
P4	4 0 5	0 2 0	
P5	3 1 4	1 1 0	

0 1 2 已不能满足任何进程的需要，不能分配

某系统有同类互斥资源 m 个，供 n 个进程共享使用。如果每个进程最多申请 x 个资源（1≤x≤m），试证明：当 n(x-1)+1≤m 时，系统不会发生死锁。

证明：因为每个进程最多申请 x 个资源，所以最坏情况是每个进程都得到了（x-1）个资源，并且现在均需申请最后一个资源。此时，系统剩余资源数为 m-n(x-1)，于是只要系统中至少还有一个资源可供使用，就可以使这 n 个进程中某个进程得到其所需要的全部资源，并能够继续执行到完成，归还资源可供其他进程使用。因而不会发生死锁。即只要 m-n(x-1)≥1 时，系统就一定不会发生死锁。亦即当 n(x-1)+1≤m 时，系统不会发生死锁。

为了解决进程间的同步和互斥问题，通常采用一种称为___(21)___机制的方法。若系统中有 5 个进程共享若干个资源 R，每个进程都需要 4 个资源 R，那么使系统不发生死锁的资源 R 的最少数目是___(22)___。

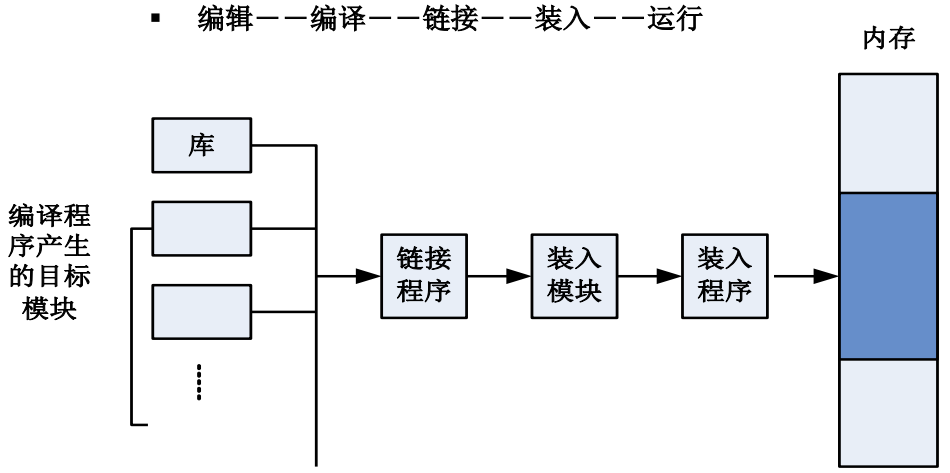
(21)A. 调度 B. 信号量 C. 分派 D. 通讯(22)A. 20 B. 18 C. 16 D. 15

答案：BC

- 若系统中有五台打印机，有多个进程均需要使用两台，规定每个进程一次仅允许申请一台，则至多允许（ 4 ）个进程参于竞争，而不会发生死锁。

第四章 存储器管理

- 编辑－－编译－－链接－－装入－－运行



一、程序的装入

1、绝对装入方式

直接用物理地址编制程序。

2、可重定位装入方式（静态重定位）

重定位——把在装入时对目标程序中的指令和数据地址的修改过程称为重定位。

3、动态运行时装入方式（动态重定位）

作业在存储空间的位置，也是装入时确定的，但在作业运行过程中，每次存访内存之前，将程序的地址（逻辑地址）变为内存的物理地址。这种变换是依靠硬件地址变换机构、自动实施，这样程序在内存的地址是可变的，可申请临时空间。

把用户空间的逻辑地址转变为内存中物理地址的过程称做重定位或地址映射。

设基址（重定位）寄存器内容为1000，在采用动态重定位的系统中，当执行指令“LOAD A,2000”时，操作数的实际地址是3000 。

二、程序的链接

1、静态链接——事先进行链接，以后不再拆开的链接方式，称为静态链接。

2、装入时动态链接——编译后的目标模块，是在装入内存时，边装入，边链接的。

3、运行时动态链接——将某些目标模块的链接推迟到执行时才进行，即在执行过程中，若发现一个被调用模块尚未装入内存时，再由操作系统去找到该模块，将它装入内存，并把它链接到调用者模块上。

4、静态链接需要共享目标模块的拷贝，而动态链接不需要共享目标模块的拷贝。

三、连续分配存储管理方式

1、 固定式分区

2、 动态分区分配——根据用户实际需要，动态的分配连续空间。

3、 动态重定位分区分配——采用动态重定位技术的分区分配。目的是解决碎片问题 。

紧凑技术：可以集中小的空闲区。

四、分区管理

分区的分配算法：

1、首次适应算法：每个空白区按地址递增顺序链接在一起，表头指向第一个空白区。

2、循环首次适应算法：将空白区构成循环链表。表头指向当前开始查找的第一个空白区。

3、最佳适应算法：空白区按容量大小递增顺序构成队列。表头指向第一个空白区。

4、最坏适应算法：空白区按容量大小递减顺序构成队列。表头指向第一个空白区。

分区回收：

回收区上临空闲区：合并，起始地址为上临空闲区的首地址，大小是两者之和。

回收区下临空闲区：合并，起始地址为回收区地址，大小是两者之和。

回收区上下均临空闲区：合并，起始地址为上临空闲区的地址，大小是三者之和。

回收区上下均不临空闲区：作为单独的一项插入分区表或分区链。

五、对换技术（交换技术）

将阻塞进程，暂时不用的程序和数据换出。将具备运行条件的进程或进程所需的程序和数据换入。

六、分页存储器管理

1、在分页存储管理方式中，一个进程的逻辑地址空间分成若干个大小相等的片，称为页面。内存空间也分成与页相同大小的存储块，并将进程的每一个页面离散地存储在内存的任一物理块中，建立相应的页表，由系统实现进程的正确运行。

2、快表——为了提高地址变换速度，可在地址变换机构中，增设一个具有并行查寻能力的特殊高速缓冲存储器，称为快表。

七、分段存储管理

① 在分段存储管理方式中，一个作业的地址空间分成若干个段，每一段定义了一组逻辑信息，则为每个段分配连续的分区，而进程中的各段可以离散地存储在内存中不同的分区中，建立相应的段表，由系统实现进程的正确运行。

② 分页与分段存储管理的区别？

答：P138

分页与分段存储管理的区别？

- 页是信息的物理单位，段是信息的逻辑单位；
- 页的大小是由系统固定的，段的长度因段而异，由用户决定；
- 分页的作业地址空间是一维的，分段的作业地址空间是二维的。

八、段页式管理

（1） 基本思想（见 P140）

（2） 地址变换机构 （见 P141）

- 某虚拟存储器中的用户空间共有 32 个页面，每页 1KB,主存 16KB。假定某时刻系统为用户的第 0、1、2、3 页分别分配的物理块号为 5、10、4、7，虚拟地址 0A6FH 对应的物理地址为【 】
- 解：虚空间共 32 个页面，页大小为 1KB，所以虚

地址共 15 位二进制长度

0A6F 对应的二进制数为：000 1010 0110 1111

虚页号为 00010，即第二页，对应的物理块号为 4，即 0100，因此，相应的物理地址为

000100 10 0110 1111

126FH

虚拟存储管理

1、 虚拟存储器的概念？

虚拟存储器：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进行扩充的一种存储器系统。实际上，用户所看到的大容量只是一种感觉，是虚的，故而得名虚拟存储器。

特点：离散性；多次性；对换性；虚拟性

虚拟存储管理系统的基础是程序的局部性理论。

2、 请求分页存储管理方式

(1) 基本思想

在请求分页存储管理方式中，不限定把进程的整个地址空间全部装入主存，而只要求把当前需要的一部分装入主存，由系统实现进程的正确运行，其它的页面当需要时才去调用。这样实现了主存的“扩充”。

(2) 地址变换机构（见 132）

在请求页式存储管理中，当查找的页不在内存中时，要产生缺页中断。这时可以根据页面置换算法选择一些暂时不用的页面置换出去，把需要的页面调入内存。

页面置换算法：

●FIFO 例如,P150

●最近最久不用页面置换算法 LRU 例如,(P151)

●简单的 Clock 置换算法例如,(P153)

3、系统抖动？

- 系统抖动是指被调出的页面又立刻被调入所形成的频繁调入调出现象 。
- 系统“抖动”现象的发生是由置换算法选择不当引起的。

1. 设某作业占有 7 个页面，如果在主存中只允许装入 4 个工作页面（即工作集为 4），作业运行时，实际访问页面的顺序是 1，2，3，6，4，7，3，2，1，4，7，5，6，5，2，1 试用 FIFO 与 LRU 页面调度算法，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。（假设开始的 4 个页面为空）

1. 答：FIFO

页面	1	2	3	6	4	7	3	2	1	4	7	5	6	5	2	1
缺	*	*	*	*	*	*		*	*			*	*			
M	1	1	1	1	4	4	4	4	4	4	4	5	5	5	5	5
M		2	2	2	2	7	7	7	7	7	7	7	6	6	6	6
M			3	3	3	3	3	2	2	2	2	2	2	2	2	2
M				6	6	6	6	6	1	1	1	1	1	1	1	1

中断次数=10　　缺页率：10/16 =62.5%

答：LRU

页	1	2	3	6	4	7	3	2	1	4	7	5	6	5	2	1
标					*	*		*	*	*	*	*	*		*	*
M	1	1	1	1	4	4	4	4	1	1	1	1	6	6	6	6
M		2	2	2	2	7	7	7	7	4	4	4	4	4	2	2
M			3	3	3	3	3	3	3	3	7	7	7	7	7	1
M				6	6	6	6	2	2	2	2	5	5	5	5	5

中断次数=14　　缺页率：14/16=87.5%

3、请求分段存储管理方式

（1） 基本思想

在请求分段存储管理方式中，把作业的所有分段的副本保存在辅存中，当其运行时，只要求把当前需要的一段或数段装入主存，其它的段当需要时才装入，由系统实现进程的正确运行。这样实现了主存的“扩充”。

（2） 地址变换机构 （见P138）

●分段保护：

越界检查（段长值）

存取控制检查

第五章　　设备管理

一、 I/O系统的组成

1、I/O系统的结构

（1） 设备

① 独占设备——在一段时间内只允许一个用户访问的设备。

② 共享设备——在一段时间内允许多个用户访问的设备。

③ 虚拟设备——将一台独占物理设备变换为若干台逻辑设备。

（2） 设备控制器

① 是CPU与I/O设备之间的接口，它接收从CPU发来的命令，并控制I/O设备工作。

② 设备控制器是可编址设备。当用于控制多台设备时，则具有多个地址。

I/O通道分类：

① 字节多路通道

② 数组选择通道——按数组方式进行数据传送，但在一段时间内只能为一台设备占用，执行一道通道程序。

③ 数组多路通道——按数组方式进行数据传送，但能为多台设备占用，高速的进行数据传送。

二、 I/O控制方式

1、程序I/O方式（查询方式） (P167)

2、中断驱动方式 (P168)

3、DMA方式(P169)

4、I/O通道控制方式（P170）

通道是通过执行通道程序，并与设备控制器共同实现对Ｉ/Ｏ设备的控制的。通道程序是由一系列通道指令（或称为通道命令）所构成的 。 通道又称为特殊的处理机 。

三、 缓冲管理

引入缓冲原因（见P172）

（1）缓和CPU与Ｉ/Ｏ设备间速度不匹配的矛盾。

（2）减少对CPU的中断频率，放宽对CPU中断响应时间的限制。

（3）提高CPU和Ｉ/Ｏ设备之间的并行性。

例如： 单缓冲、 双缓冲、 循环缓冲、 缓冲池

缓冲技术是以空间换取时间，而且只能在设备使用不均衡时起到平滑作用。

四、 设备分配

设备管理是通过一些数据结构来实现对其设备进行管理和控制的。

1、 设备控制表、通道控制表、系统设备表、控制器控制表

2、 设备分配中应考虑的若干因素

（1）设备的固有属性：独享设备、共享设备、虚拟设备

（2）设备的分配算法：FIFO、优先级高者优先

（3）设备分配中的安全性

3、设备固有属性不同，其分配算法不同

4、SPOOLING 技术可将一台物理设备虚拟为多台逻辑设备，可为多个用户所共享。

SPOOLing 技术的核心思想是：在快速辅助存储设备中建立 I/O 缓冲区用于缓存从慢速输入设备流入内存的数据或缓存从内存流向慢速输出设备的数据。

- spooling 技术是对脱机 I/O 系统的模拟，该技术是建立在多道程序的环境上、可以提高独占设备的利用率。

五、设备处理

1、设备处理程序又称为设备驱动程序，它是 I/O 进程与设备控制器之间的通信程序。

2、用户请求设备使用的是逻辑设备名。

由系统通过逻辑设备表实现逻辑设备到物理设备的映射。当更换物理设备时，用户的程序不用改，仅修改逻辑设备表。

用户程序应与实际使用的物理设备无关，这种特性就称作 设备无关性

磁盘存储器管理

一、磁盘访问时间

磁盘的访问时间分成以下三个部分寻道时间、旋转延迟时间和传输时间。

二、 早期磁盘调度算法

1、 先来先服务（见 P194）

2、 最短寻道时间优先（见 P95）

3、 扫描法（见 P195-196）

4、 循环扫描法（见 P96）

若有磁盘共有 200 个柱面，其编号为 0-199，假定磁头刚完成 56 号磁道的访问，磁头正在 98 号磁道上，现有一个请求队列在等待访问磁盘，访问的磁道号分别为 190，97，90，45，150，32，162，108，112，80。请写出分别采用最短寻道时间优先和电梯调度算法处理上述服务请求的次序和移动的总磁道数。

190，97，90，45，150，32，162，108，112，80

SSTF		电梯
97	1	108
90	7	112
80	10	150
108	28	162
112	4	190
150	38	97
162	8	90
190	28	80
45	145	45
32	13	32

第六章　　文件系统

文件管理实际上是管理对辅助存储空间

一、文件逻辑结构（从用户角度看到的文件组织形式）

1、流式文件

2、记录式文件

二、文件的物理结构（从系统角度看到的文件组织形式）。

1、顺序结构

2、链式结构

3、索引结构

三、　　外存分配方法

1. 连续分配——将文件信息存放在连续编号的物理块中。

（见 P213　图 6-7）

优点：结构简单，存取速度快。

缺点：长度事先确定，随后不允许增加长度。

2、链接分配——将文件信息存放在非连续编号的物理块中。（见 P215　图 6-8）

优点：插入、删除方便，文件长度可变。

缺点：查找困难。

3、索引分配（见 P221、　图 6-12）

优点：可以随机存取。

缺点：增加空间的开销。

单级索引、多级索引。

注意 UNIX 系统中的文件分配方式：混合索引分配

- I-addr 索引区共有 13 项，其中前 10 项直接登记了存放文件信息的物理块号(即 I-addr(0)到 I-addr(9))，这称为直接寻址。0 到 9 可以看成是逻辑块号。如果一个文件大于 10 块，则利用 I-addr(10)作一次间接寻址，即 I-addr(10)指向一个物理块，其中最多可存放 128 个存放文件信息的物理块号。如果文件更大，还可以分别用 I-addr(11)和 I-addr(12)作两次甚至三次间接寻址。

四级间接寻址示意图

目录表　索引结点表　间接索引寻址　　数据块

文件系统采用多重索引结构搜索文件内容。设块长为 512 字节，每个块号长 3 字节，如果不考虑逻辑块号在物理块中所占的位置，分别求二级索引和三级索引时可寻址的文件最大长度。

解析：一个索引块中可放的盘块号：512/3~170

二级索引：170*170*512=7225K

三级索引：170*170*170*512=2456500KB

存放在某磁盘上的文件系统，采用混合索引分配方式，其 FCB 中共有 13 个地址项，地 0-9 个地址项为直接地址，第 10 个地址项为一次间接地址，第 11 个地址项为二次间接地址，第 12 个地址项为三次间接地址.如果每个盘块的大小为 4KB,若盘块号需要 4 个字节来描述，请问该系统中允许文件的最大长度为多少？

4K*(10+1K+1K*1K+1K*1K*1K)

目录管理

1、 对文件目录管理要求（见 P224 ）

2、 文件控制块与文件目录（见 P224-225）

3、 单级文件目录（见 P226）

缺点：查找速度慢、不允许重名、不便于实现文件共享

4、 两级目录和多级目录（见 P227-228）

优点：① 检查速度快

② 不同目录可以重名

③ 不同用户可使用不同名字，来访问系统中的同一个共享文件。

索引结点的引入

在检索目录文件的过程中，只用到了文件名，仅当找到一个目录项（即其中的文件名与指定要查找的文件名相匹配）时，才需从该目录项中读出该文件的物理地址。而其它一些对该文件进行描述的信息，在检索目录时一概不用，显然，这些信息在检索目录时，不需调入内存。为此，在有的系统中，如 UNIX 系统，便采用了把文件名与文件描述信息分开的办法，亦即，使文件描述信息单独形成一个称为索引结点的数据结构，简称为 i 结点。在文件目录中的每个目录项，仅由文件名和指向该文件所对应的 i 结点的指针所构成。

五、 空闲存储空间的管理方法

1、空闲表法（见 P231）

2、空闲链表法（见 P232）

3、位示图法（见 P232）

在采用位示图管理文件存储空间时，一个二进制位对应一个物理块。

4、成组链接法 （见 P233）

UNIX 文件系统对盘空间的管理采用成组链接法

某操作系统的磁盘文件空间共有 500 块，若用字长为 32 位的位示图管理盘空间，试问：

（1）位示图需多少个字？

（2）第 i 字第 j 位对用的块号是多少？

(设位示图中字、位均从 1 开始编址，磁盘存储空间也是从 1 编址)

字：500/32=16

块号： 32*（i-1）+j

系统态（管态、核心态）：只能运行 os 的程序。

用户态（目态）：运行用户的程序。

特权指令：只允许操作系统使用。

非特权指令：一般用户使用。