

专业课

# 计算机

操作系统

袁礼

华图网校

版权所有 盗版必究

# 目录

操作系统.....	1
1. 操作系统概况.....	1
2. 用户界面.....	6
3.1 进程 .....	8
3.2 进程控制.....	13
4. 处理机调度.....	20
5. 存储管理.....	23
6. 文件系统.....	29
7. 设备管理.....	37

# 操作系统

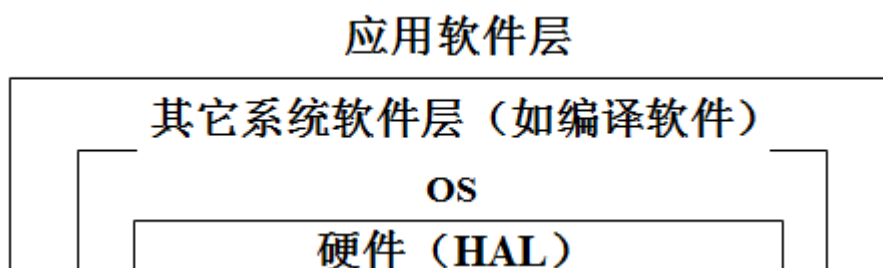
本章内容：

- 操作系统概况
- 用户界面
- 进程
- 处理机调度
- 处理机管理
- 文件系统
- 设备管理

## 1. 操作系统概况

### 一、操作系统地位

- 硬件抽象层（HAL）之上
- 所有其它软件层之下



### 二、操作系统特性

- 程序并发性
  - 多个程序在宏观上同时向前推进、微观上串行推进
  - 并发(concurrent) vs. 并行 (parallel)
- 资源共享性
  - 多个程序共用系统中的各种软硬件资源
  - 在操作系统的协调和控制下
- 虚拟性
  - 物理上的一台设备变成逻辑上的多台设备

- 不确定性

### 三、操作系统的基本类型

- 多道批处理操作系统(batch processing system)
- 分时操作系统(time-sharing system)
- 实时操作系统(real time system)
- 通用操作系统(multi-purpose system)
- 单用户操作系统(single user system)
- 网络操作系统(network operating system)
- 分布式操作系统(distributed operating system)
- 多处理机操作系统(multi-processor system)

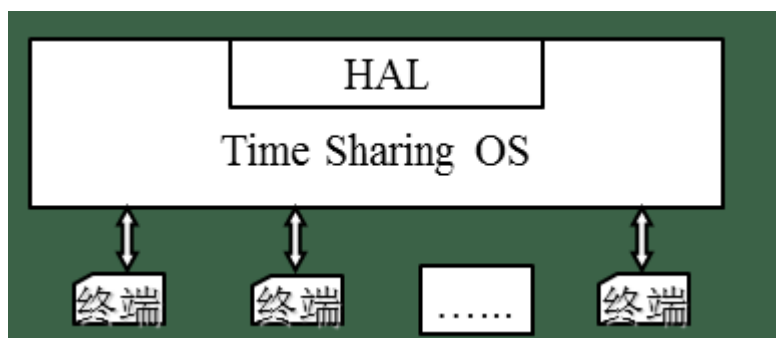
#### ①多道批处理系统

- 特点
  - 多道：系统中同时容纳多个作业
  - 成批：作业分批进入系统
  - 宏观上并行，微观上串行

多道批处理系统是以脱机为标志的操作系统，适用于处理运行时间比较长的程序。

- 主机中作业合理搭配
  - 目标 1：提高资源利用率
  - 目标 2：提高吞吐量(throughput)

#### ②分时操作系统（On-line）



界面 1：交互式命令语言(eg. shell, command)

界面 2：图形用户界面(GUI)

- 特点：

- 多路性：一个主机与多个终端相连；
- 交互性：以对话的方式为用户服务；
- 独占性：每个终端用户仿佛拥有一台虚拟机。

分时操作系统是以联机为标志的操作系统，特别适用于程序的动态调试与修改。

### ③实时操作系统

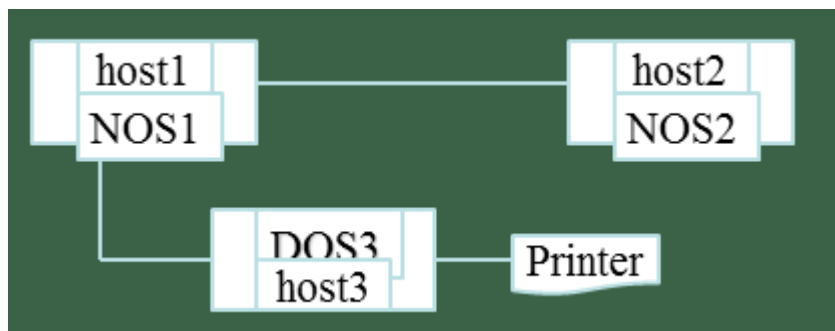
- 实时控制
  - 工业控制，军事控制，医疗控制，…….
- 实时信息处理
  - 航班定票，联机情报检索，…….

### ④通用操作系统(multi-purpose OS)

- 同时具有：分时、实时、批处理功能。
- 目标：
  - 提高处理能力；
  - 扩展应用领域。
- 常见模式：
  - 分时(前台)+批处理(后台)
  - 实时(前台)+批处理(后台)

### ⑤网络操作系统

建立在宿主操作系统之上，提供网络通讯、网络资源共享、网络服务。



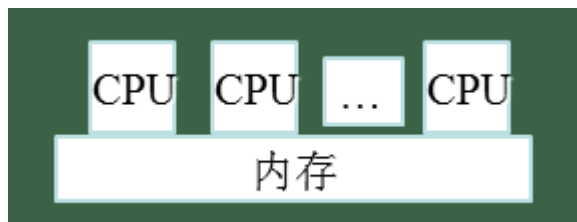
网络操作系统的目标

- 相互通讯
- 资源共享（信息，设备）
- 提供网络服务

- database server
- ftp server
- e-mail server
- telnet server
- etc.

## ⑥分布式操作系统

- 紧耦合: (tightly coupled)
  - 由多机系统发展而来 (多 CPU)
  - 有公共内存
  - 多处理机操作系统



- 松散耦合: (loosely coupled)
  - 由计算机网络发展而来 (多 Host)
  - 无公共内存, 无公共时钟
- 分布式操作系统特征:
  - 统一的操作系统
  - 资源的进一步共享
  - 可靠性
  - 透明性

## ⑦多处理机操作系统

- 多处理机系统
  - 具有公共内存的多 CPU 系统
- 对称多处理机系统(SMP)
  - 没有主从关系的多处理机系统
- 多处理机操作系统
  - 有效管理和使用多个 CPU 的操作系统

- 复杂性：多个主动体（CPUs）

- 如：UNIX, Linux, Windows

#### 四、操作系统的功能

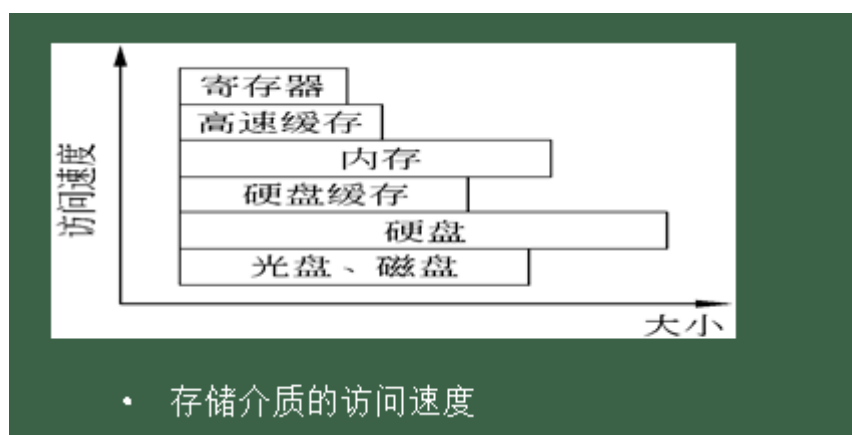
- 处理机管理
- 存储管理
- 设备管理
- 信息管理（文件系统管理）
- 用户接口

#### 五、与操作系统相关的几种主要寄存器

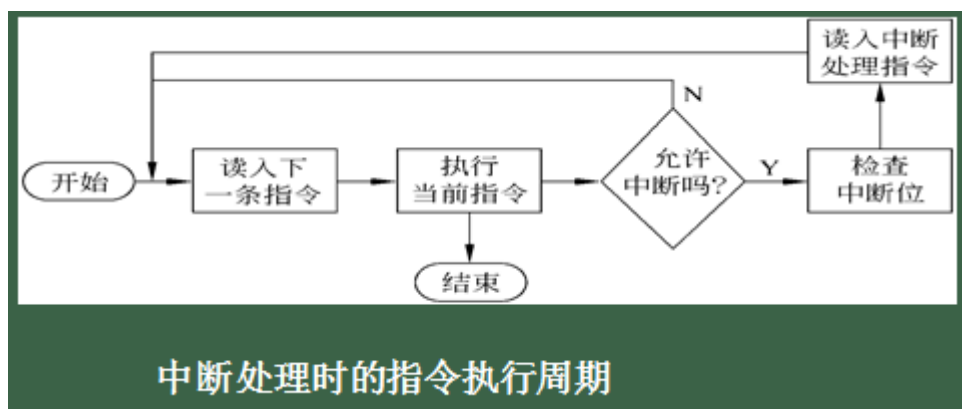
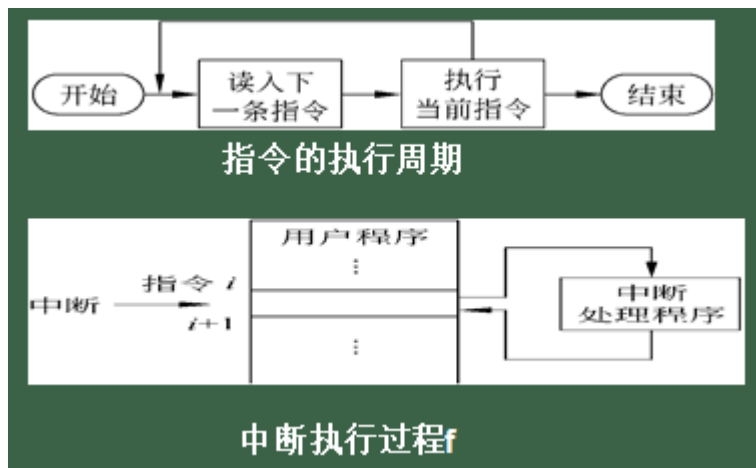
- 1. 数据寄存器
- 2. 地址寄存器
- 3. 条件码寄存器
- 4. 程序计数器 PC
- 5. 指令寄存器 IR
- 6. 程序状态字 PSW
- 7. 中断现场保护寄存器
- 8. 过程调用堆栈

#### 六、存储器的访问速度

一般来说，速度高的存储介质，成本高，容量小；容量大的存储介质，速度慢，成本低。



#### 七、指令的执行与中断



【练习】(ABD) 是分时系统的基本特征：

- A、同时性
- B、独立性
- C、实时性
- D、交互性

【练习】

( D ) 不是基本的操作系统。

- A、批处理操作系统
- B、分时操作系统
- C、实时操作系统
- D、网络操作系统

## 2. 用户界面

### 一、一般用户的输入输出界面

- 作业的定义



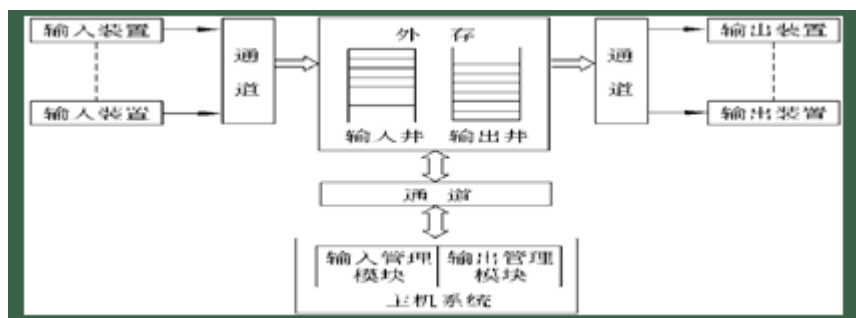
- 作业组织
- 一般用户的输入输出方式

### ①作业的定义

在一次应用业务处理过程中，从输入开始到输出结束，用户要求计算机所做的有关该次业务处理的全部工作称为一个作业。作业由不同的顺序相连的作业步组成。

### ②一般用户的输入输出方式

- 1. 联机输入输出方式
- 外围设备直接和主机相连，速度慢。
- 2. 脱机输入输出方式
- 外围机进行联机输入输出处理，通过外围机的后援存储来实现和主机的连接。速度快。
- 3. 直接耦合方式
- 主机和外围机通过一个公共外存直接连接。速度快，人工不用干预
- 4. SPOOLING 系统
- 低速输入输出设备与主机交换的一种技术，通常也称为“假脱机真联机”，核心思想是以联机的方式得到脱机的效果。
- 外围设备通过通道或 DMA 器件与主机和外存相连。



- 5. 网络联机方式

### ③命令控制界面

用户使用命令控制界面的方式：

#### 1、脱机方式

填写作业说明书，用户不能干预作业执行。

#### 2、联机方式

不用填写作业说明书，用户能够干预作业执行。

### ④系统调用

系统调用分为 6 类：

- 1 设备管理
- 2 文件管理
- 3 进程控制
- 4 进程通信
- 5 存储管理
- 6 线程管理

【练习】以下不属于系统调用的是（C）。

- A.设备管理
- B.文件管理
- C.操作控制
- D.进程通信
- E.线程管理

## 3.1 进程

### 一、程序的并发执行

#### （1）程序(program)

用来描述计算机所要完成的独立功能，并在时间上严格地按前后次序相继地进行计算机操作序列集合，是一个静态的概念。

#### （2）程序的顺序执行（sequence)

程序顺序执行的概念

一个具有独立功能的程序独占处理机直至最终结束的过程称为程序的顺序执行。

程序顺序执行的特征

- 顺序性
- 封闭性
- 可再现性

#### (3)程序的并发(concurrent)执行

程序的并发执行：宏观上同时向前推进，微观上同一时刻只有一个程序运行。

程序并发执行分为两种：一种是程序间的并发。另一种是同一程序内部多条指令的并发。

程序并发执行的特性：

交叉性、非封闭性、不可再现性

## 二、进程的定义

- 定义：

并发执行的程序在执行过程中分配和管理资源的基本单位。

- 定义强调两个方面：
  - 动态：执行中的程序；
  - 并发：可与其他进程同时执行。

### ①并发与并行

- 并发：concurrent
  - 宏观同时，“交替执行”，不要求多个 CPU
- 并行：parallel
  - 微观同时，要求多个 CPU
  - “并行算法”

### ②进程的定义

进程与程序的区别与联系：

- 进程是一个动态概念，程序是一个静态概念。
- 进程具有并发特征，而程序没有。
- 进程是竞争计算机系统资源的基本单位。
- 不同的进程可以包含同一程序，只要该程序所对应的数据集不同。（例如多个浏览器进程）

### ③进程控制块 PCB

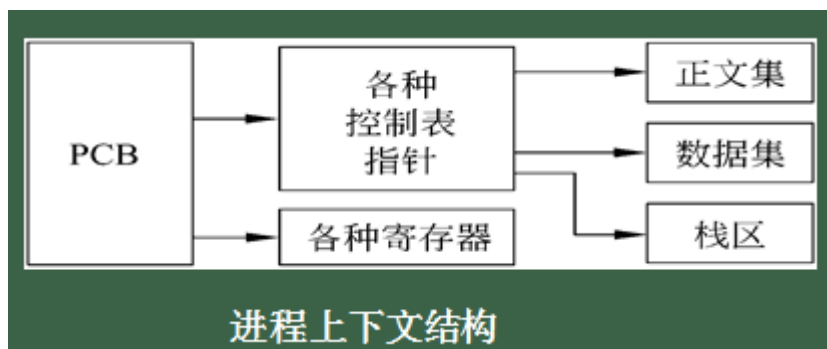
（Process Control Block）

- 定义：标志进程存在的数据结构，其中保存系统管理进程所需的全部信息。
- PCB 的内容：(不同系统不尽相同)
  - 1. 描述信息
  - 2. 控制信息
  - 3. 资源管理信息
  - 4. CPU 现场保护结构

## 三、进程的组成与上下文

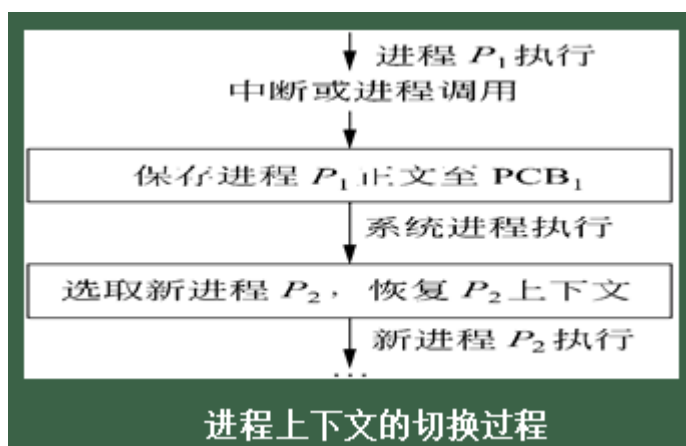
- 进程的组成

- 进程控制块(process control block)
  - 建立进程⇒建立 PCB
  - 撤销 PCB⇒撤销进程
- 程序
  - 代码(code)
  - 数据(data)
  - 堆栈(stack+heap)
- 进程上下文 (process context)
  - 进程的物理实体与支持进程运行的物理环境统称为进程上下文。
    - PCB+程序
    - 系统环境：地址空间，系统栈，打开文件表，...



#### 四、进程上下文切换

- 上下文切换 (context switch)
  - 由一个进程的上下文转到另外一个进程的上下文
- 系统开销 (system overhead)
  - 运行操作系统程序完成系统管理工作所花费的时间和空间



## 五、进程空间与大小

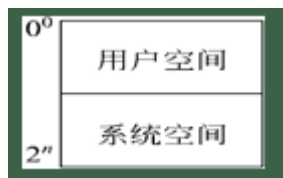
进程在内存中自己拥有的一个地址空间称为进程空间。

进程空间的大小只与处理机的位数有关。

进程空间可以分为：用户空间与系统空间。

用户程序在用户空间中运行。

操作系统内核程序在系统空间中运行



## 六、进程的类型

- 进程类型
  - 系统进程
    - 运行操作系统程序，完成系统管理(服务)功能。
  - 用户进程
    - 运行用户(应用)程序，为用户服务。

## 七、进程间相互联系与相互作用

- 相互联系
  - 相关进程
    - 同一家族的进程
    - 可以共享文件，需要相互通讯，协调推进速度...
    - 父进程可以监视子进程，子进程完成父进程交给的任务。
  - 无关进程
    - 没有逻辑关系、同时执行的进程。
    - 有资源竞争关系，互斥、死锁、饿死。

## 八、进程状态及其转换

- 进程的状态
- 进程状态的转换
- 进程队列

### ①进程状态：

初始态（Initial）：进程刚被创建，其他进程正占据处理器而得不到执行。

运行态 (Run): 占有 CPU 正在向前推进。

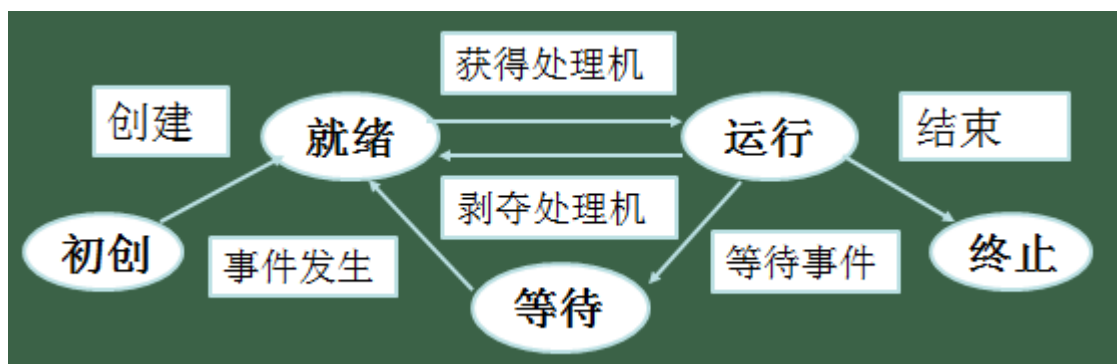
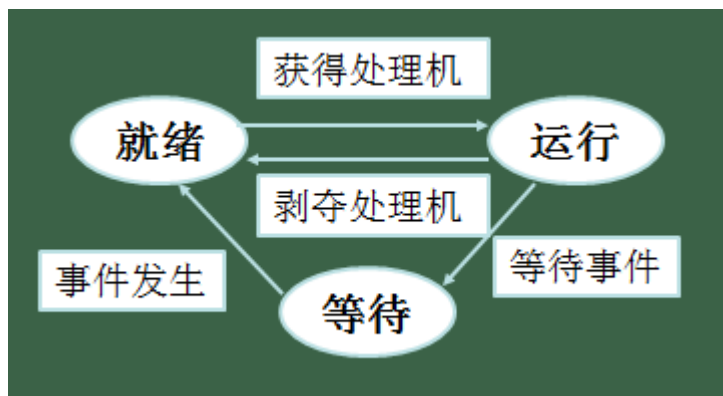
就绪态 (Ready): 可以运行, 但未得到 CPU。

等待态 (Wait): 等待某一事件发生。

终止态 (Stop): 进程执行结束, 将退出执行而被终止。

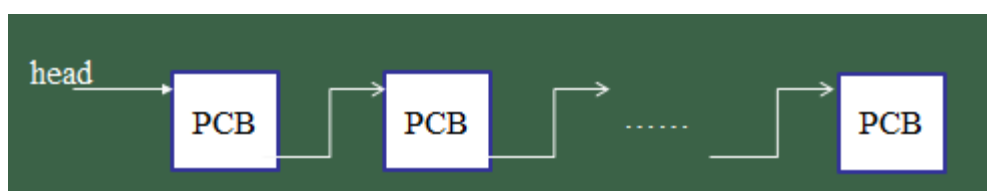
## ②进程状态转换

- 状态转换
  - 就绪⇒运行: 获得处理机
  - 运行⇒就绪: 剥夺处理机
  - 运行⇒等待: 申请资源未得到, 启动 IO
  - 等待⇒就绪: 得到资源, IO 中断
- 进程状态转换图



## ③进程队列

PCB 构成的队列: (不一定 FIFO)



1. 就绪队列：系统一个或若干个（根据调度算法确定）
2. 等待队列：每个等待事件一个
3. 运行队列：每个处理机一个

【练习】进程所请求的一次打印输出结束后，将使进程状态从（D）

- A、运行态变为就绪态
- B、运行态变为等待态
- C、就绪态变为运行态
- D、等待态变为就绪态

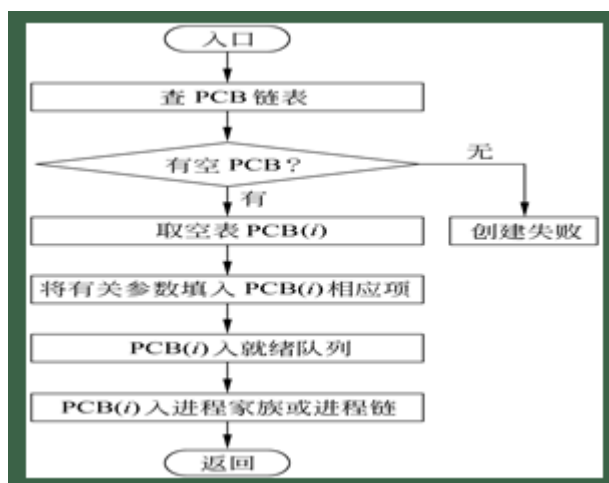
## 3.2 进程控制

进程控制：系统使用一些具有特定功能的程序段来创建、撤销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。

- 原语：系统态下执行的某些具有特定功能的程序段。
- 原语的分类：
  - 机器指令级：不许中断
  - 功能级：不许并发

### 一、进程创建与撤销

- ①进程创建方式：
- 系统创建、父进程创建

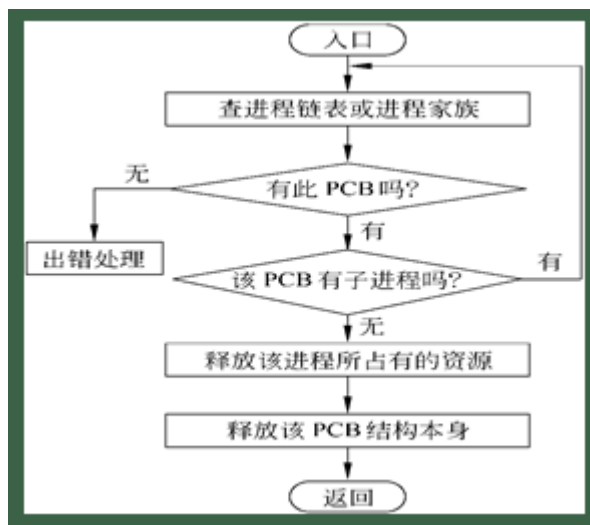


### ②进程撤销的方式：

正常终止

非正常终止

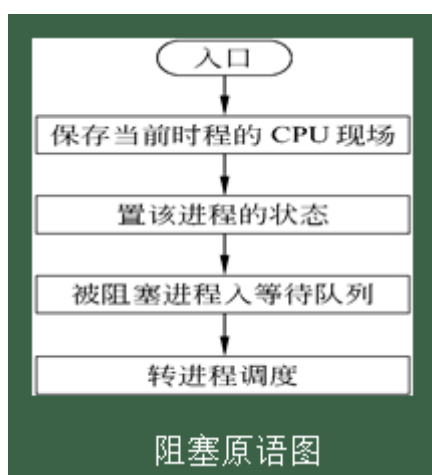
## 祖先进程撤销



## 二、进程的阻塞与唤醒

阻塞原语实现了进程由执行状态到等待状态的转变。

阻塞原语是由进程自己调用的。



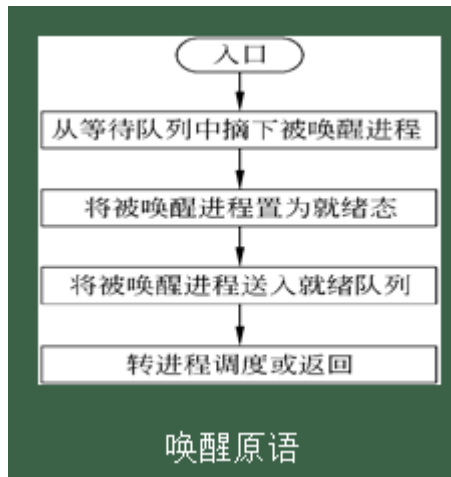
唤醒原语实现了进程由等待状态到就绪状态的转变。

唤醒的方式:

系统进程唤醒

事件发生进程唤醒





### 三、与时间有关的错误

错误原因之 1:

进程执行交叉(interleave);

错误原因之 2:

涉及公共变量(x)。

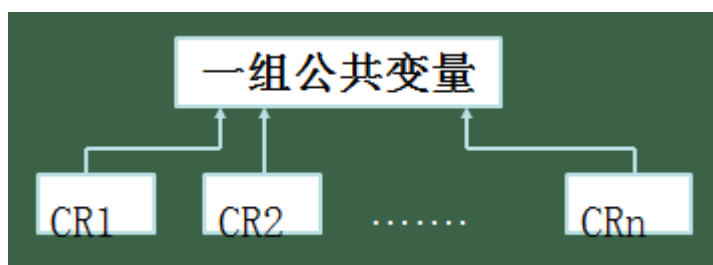
注意:

某些交叉结果不正确;

必须去掉导致不正确结果的交叉。

### 四、共享变量与临界区域

- 共享变量 (shared variable)
  - 多个进程都需要访问的变量。
- 临界区域 (critical region)
  - 访问共享变量的程序段。



### 五、进程互斥

定义: 多个进程不能同时进入关于同一组共享变量的临界区域, 否则可能发生与时间有关的错误, 这种现象称为进程互斥。

二层涵义:

- (1) 任何时刻最多只能有一个进程处于同一组共享变量的相同的临界区域；
- (2) 任何时刻最多只能有一个进程处于同一组共享变量的不同的临界区域。

注意：互斥是相对于公共变量而言的。

- 并发进程互斥执行必须满足 4 条准则：
  - 不能假设并发进程的相对执行速度。
  - 某进程不在临界区，不能阻止其他进程进入临界区。
  - 若干进程申请进入临界区，只能允许一个进程进入。
  - 某进程申请进入临界区，应该在有限的时间内得以进入临界区。

## 六、进程互斥的实现

进程互斥的实现有两种方法：

### ①软件方法实现：

完全用程序实现，不需特殊硬件指令支持。

可用于单 CPU 和多 CPU 环境中。

有忙式等待问题

### ②硬件方法实现：

- 硬件提供“测试并建立”指令、“交换”指令
- 硬件提供“关中断”和“开中断”指令

关中断

{ Critical Region }

开中断

注意：

- (1) 开关中断只用在单 CPU 系统
- (2) 影响并发性。

例：对临界区加锁实现互斥：

当某个进程进入临界区之后，它将锁上临界区，直到它退出临界区时为止。

lock( key [s] )

〈临界区〉

unlock( key [s] )

## 七、进程同步

- ①进程同步的概念

例：司机-乘客问题

司机活动：(P1)

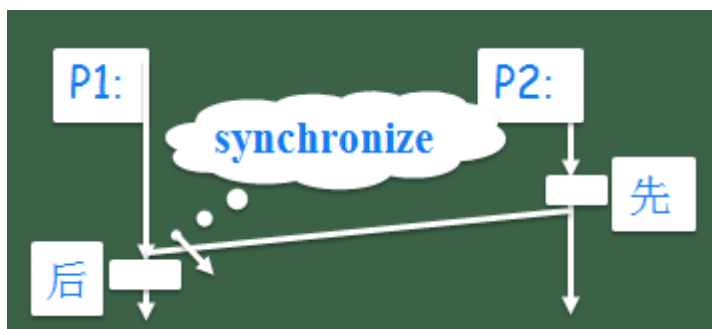
```
do{
  启动车辆
  正常行驶
  到站停车
while(1)
```

乘客活动：(P2)

```
do{
  上车
  投币
  下车
while(1)
```



- 定义：一组进程，为协调其推进速度，在某些关键点处需要相互等待与相互唤醒，进程之间这种相互制约的关系称为进程同步。



## ②进程同步机制

- 定义：用于实现进程同步的工具称为同步机制（synchronization mechanism）

- 同步机制要求：

- 描述能力够用；
- 可实现；
- 高效；
- 使用方便。

## ③进程通信

- 进程通信：进程之间的数据传送。

- 低级通讯（简单信号）

- 进程互斥
- 进程同步

- 高级通讯（大宗信息）

单机系统中进程通信可以分为 4 种形式：

### 1、主从式

- 2、会话式
- 3、消息或邮箱机制
- 4、共享存储方式

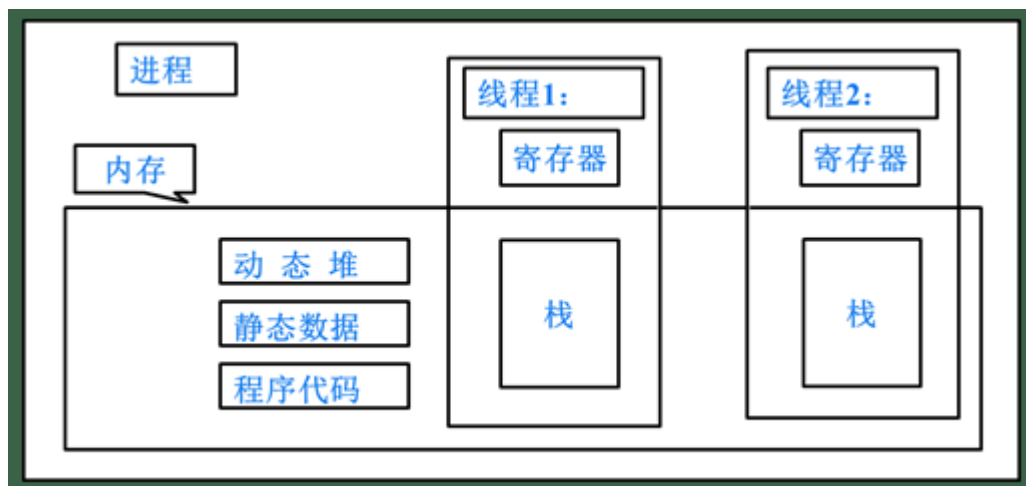
## 八、死锁概念

- ① 有并发进程  $P_1, P_2, \dots, P_n$ ，它们共享资源  $R_1, R_2, \dots, R_m$  ( $n > 0, m > 0, n \geq m$ )。其中，每个  $P_i$  ( $1 \leq i \leq n$ ) 拥有资源  $R_j$  ( $1 \leq j \leq m$ )，直到不再有剩余资源。同时，各  $P_i$  又在不释放  $R_j$  的前提下要求得到  $R_k$  ( $k \neq j, 1 \leq k \leq m$ )，从而造成资源的互相占有和互相等待。在没有外力驱动的情况下，该组并发进程停止住往前推进，陷入永久等待状态。把这种现象称为死锁。
- 从概念而来的结论：
  - 参与死锁的进程至少有二个；
  - 每个参与死锁的进程均等待资源；
  - 参与死锁的进程中至少有两个进程占有资源；
  - 死锁进程是系统中当前进程集合的一个子集。
- ②死锁预防
  - 对进程有关资源的活动加限制，所有进程遵循这种限制，即可保证没有死锁发生。
    - 优点：简单，系统不需要做什么。
    - 缺点：对进程的约束，违反约束仍可能死锁。
  - 预防方法：
    - 预先分配法；
    - 有序分配法。
- ③死锁的恢复
  - 1. 重新启动
    - 简单，代价大，涉及未参与死锁的进程。
  - 2. 终止进程(process termination)
    - 环路上占有资源的进程。
    - (1) 一次性全部终止；(2) 逐步终止（优先级，代价函数）
  - 3. 剥夺资源(resource preemption)+进程回退(rollback)
    - (1) select a victim; (2) rollback.

## 九、线程的引入

- 进程切换
  - 上下文涉及内容多，开销大，“笨重”
  - 相关进程之间耦合关系差
- 解决方案
  - Multi-threading
  - 同一进程中包含多个线程
  - 上下文只涉及寄存器和用户栈，切换速度快
  - 相关线程之间通讯方便、快捷
- ①线程的概念
- 进程中一个相对独立的执行流。
- 进程 vs.线程
  - 进程是资源分配单位
  - 线程是执行单位
- 多线程优点
  - 切换速度快（地址空间不变）(light weighted)
  - 系统开销小
  - 通讯容易（共享数据空间）
- ②线程控制块
- TCB（Thread control block）
  - 标志线程存在的数据结构，其中包含对线程管理需要的全部信息。
- 内容
  - 线程标识
  - 线程状态
  - 调度参数
  - 现场(通用寄存器,PC,SP)
- 存放位置
  - 用户级线程：目态空间（运行系统）
  - 核心级线程：系统空间

### ③多线程结构（用户视图）



【练习】产生系统死锁的原因可能是由于（C）。

- A、进程释放资源
- B、一个进程进入死循环
- C、多个进程竞争，资源出现了循环等待
- D、多个进程竞争共享型设备

【练习】线程是操作系统的概念，具有线程管理的操作系统有（ABCE）

- A、Windows 7
- B、OS/2
- C、Ubuntu
- D、DOS
- E、Cent OS

## 4、处理机调度

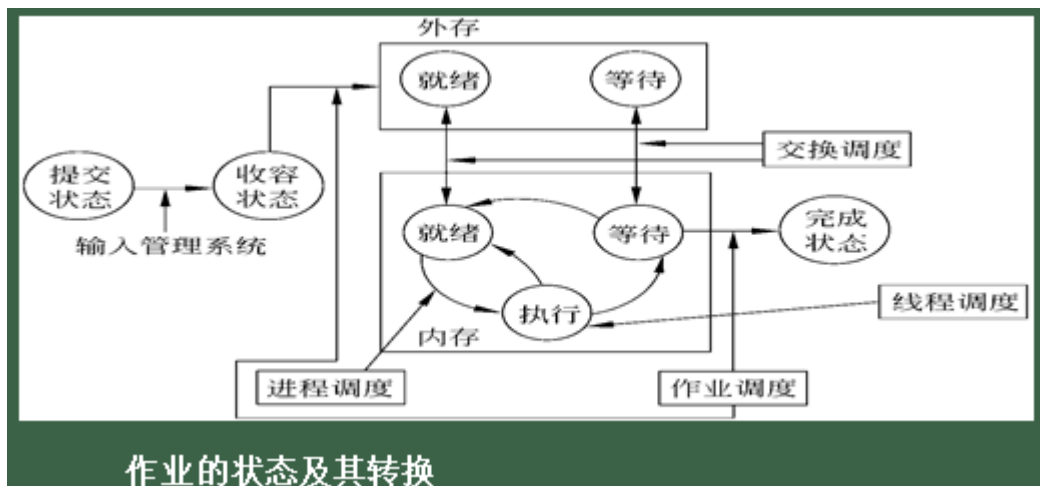
- 处理机调度是指处理机在可运行实体上的分配。批处理系统、分时系统、实时系统不同类型的操作系统采用不同的处理机的调度策略。

### 一、处理机调度包括：

- 分级调度
- 作业调度
- 进程调度
- 调度算法
- 实时调度算法

### 二、作业的状态及其转换

- 一个作业从用户提交开始到占有处理机被执行，要由系统经过多级调度才能实现。
- 一个作业从提交给计算机系统到执行结束退出系统，一般都要经历提交、收容、执行和完成等四个状态。



- 作业的提交状态：从输入设备进入外部存储设备的过程。
- 作业的后备状态：作业的全部信息输入至输入井。
- 作业的执行状态：被作业调度程序选中的作业所处的状态。
- 作业的完成状态：作业运行完毕，所占用的资源尚未全部被系统回收。

### 三、调度的层次

处理机调度分为 4 级：

- 作业调度（高级调度）
- 交换调度（中级调度）
- 进程调度（进程调度）
- 线程调度

### 四、作业与进程的关系

- 作业是用户向计算机提交任务的任务实体
- 进程是为了完成用户任务实体而设置的执行实体。
- 一个作业由一个以上的进程组成。

### ①作业调度

- 1) 记录系统中各作业的状况。
- 2) 从后备队列中挑选出一部分作业投入执行。
- 3) 为被选中作业做好执行前的准备工作。
- 4) 在作业执行结束时善后处理工作。

### ②作业调度目标与性能衡量

作业调度目标:

- 1) 对所有作业应该是公平合理的
- 2) 应使设备有高的利用率
- 3) 每天执行尽可能多的作业
- 4) 有快的响应时间

性能衡量指标:

周转时间

带权周转时间

### ③进程调度

- 1. 记录系统中所有进程的执行情况
- 2. 选择占有处理机的进程
- 3. 进行进程上下文切换

### ④调度算法

- 先到先服务算法(FCFS)
- 短作业优先(SJF)
- 最高优先数算法(HPF)
- 循环轮转算法(RR)
- 多级队列算法(MLQ)
- 反馈排队算法(FB)

## 五、实时任务分类

- 硬实时与软实时\_
  - 硬实时 :必须满足任务截止期要求 .
  - 软实时 :期望满足截止期要求 .
- 周期性与随机性\_
  - 周期性 :每隔固定时间发生一次



- 随机性:由随机事件触发,其发生时刻不确定

- 几个术语
- Ready time: 就绪时间
- Starting deadline: 开始截止期
- Processing time: 处理时间
- Completion deadline: 完成截止期
- Occurring frequency: 发生频率

### 【练习】

作业调度是从输入井中处于（B）状态的作业中选取作业调入主存运行。

- A、运行
- B、收容
- C、输入
- D、就绪

## 5、存储管理

### 一、存储管理的功能

#### 虚拟存储器

实际上,进程是一部分放在内存,另一部分放在外存。

系统为进程安排地址的方法有两种:

一种是按照物理存储器中的位置赋予实际物理地址;

另一种是编译链接程序把用户源程序编译后链接到一个以 0 为始地址的线性或多维虚拟地址空间。



虚拟空间: 每个进程都拥有一个由存储管理方式决定的空间。

物理地址 (内存地址): 物理存储器的实际地址。

物理地址空间 (内存空间): 内存地址的集合。

虚拟地址: 虚拟空间中的地址。

地址转换: 虚拟地址到物理地址的转变。

虚拟存储器: 进程中的目标代码、数据等虚拟地址组成的虚拟空间。

## 二、地址转换

地址映射：建立虚拟地址与内存地址的映射关系。

实现地址重定位的方法有两种：静态重定位 动态重定位

### 1、静态地址重定位

定义：静态地址重定位是在虚拟空间程序执行之前由装配程序完成地址映射工作。

优点：不需要硬件支持

缺点：无法实现虚拟存储器、必须占用连续的内存空间。

### 2、动态地址重定位

定义：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。

优点：可以对内存进行非连续分配；实现虚拟存储器；有利于程序段的共享。

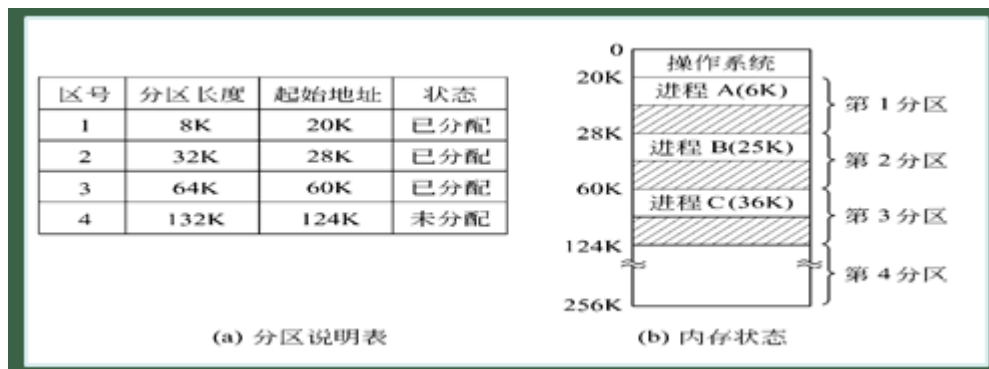
内存的分配与回收

一个作业进入内存时操作系统需将其变为进程，并为进程分配存储空间。进程运行结束时，操作系统应将其所占有的存储空间回收。

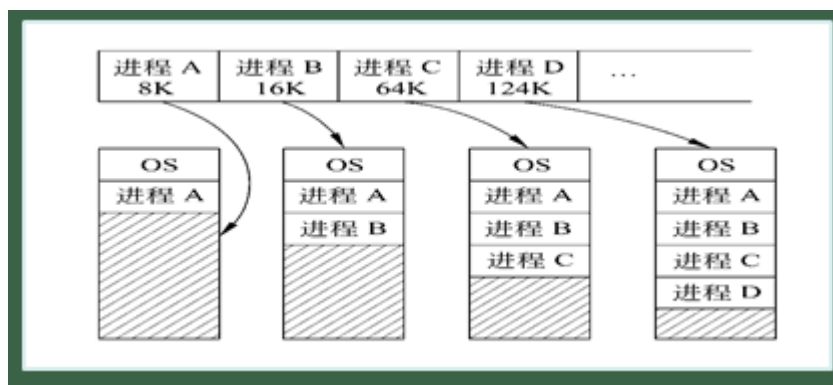
- 分配去配对象
- --内存、外存(相同方法)
- 分配去配时刻
- --进程创建、撤销、交换、长度变化

## 三、分区管理基本原理

- 分区管理的基本原理：
- 给每一个内存中的进程划分一块适当大小的存储区，以连续存储
- 各个进程的程序和数据，使进程得以并发执行。
- 分区管理的方法
- 固定分区法（静态）：在作业或进程执行之前就把内存区域固定地划分为若干个大小不等的区域。



动态分区法：在作业或进程执行过程中把内存划分为大小不等的区域。空间大小可变，可提高资源利用率。



## 分区的分配与回收

### 1 固定分区时的分配与回收

基本思路：

分配：当用户程序要装入执行时，通过请求表提出内存分配要求和所要求的内存空间大小。存储管理程序根据请求表查询分区说明表，从中找出一个满足要求的空闲分区，并将其分配给申请者。

回收：进程执行完毕时，管理程序将对应的分区状态置为未使用。

特点：简单

### 2 动态分区时的分配与回收

动态分区时的分配方法：

最先适应法

最佳适应法

最坏适应法

## 四、页式管理的基本原理

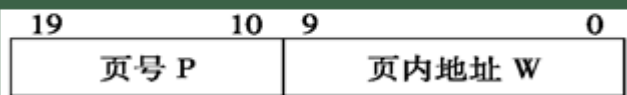
进程的虚拟空间被划分成若干个长度相等的页

内存空间也按页的大小划分为片或页面

页式管理实现了内外存存储器的统一管理

分页管理的重点在于页划分后的地址变换以及页面的调入调出技术

**页内地址连续，页面可以不连续**



- ①静态页面管理
- 静态页面管理方式：在作业或进程开始执行之前，把该作业或进程的程序段和数据全部装入内存的各个页面中，并通过页表和硬件地址交换机构实现虚拟地址到内存物理地址的地址映射。
- 1 内存页面分配与回收
- (1) 页表
- 页表由页号与页面号组成
- 页表在内存中占有一块固定的存储区
- 页表的大小由作业或进程的大小决定

页号	页面号

- (2) 请求表
- 整个系统一张，内容包括：进程号、请求页面数、页表开始地址、页表的长度和状态。

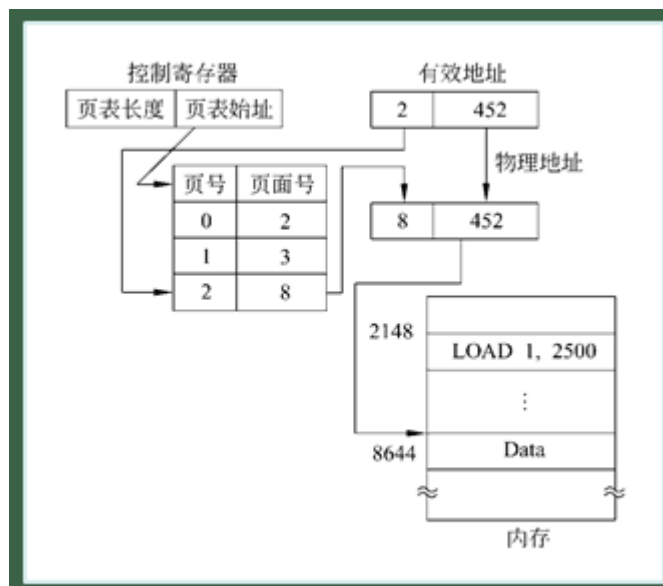
进程号	请求页面数	页表始址	页表长度	状态
1	20	1024	20	已分配
2	34	1044	34	已分配
3	18	1078	18	已分配
4	21	⋮	⋮	未分配
⋮	⋮			⋮

- (3) 存储页面表
- 整个系统一张，指出内存各页面是否已被分配出去，以及未分配页面的总数。

19	18	17	16	15	...	4	3	3	2	0
0	1	1	1	1	...	1	1	0	1	1
0	0	0	1	1	...	0	0	1	1	0
0	0	1	1	1	...	0	0	0	0	0

## 地址变换

- (1) 系统从请求表中取出所调度执行的进程页表开始地址和页表长度放入控制寄存器中。
- (2) 由控制寄存器得到页表开始地址，根据虚拟地址得到物理地址。



## ②动态页面管理

动态页面管理方式：在作业或进程开始执行之前，不是把作业或进程一次性地全部装入内存，而只装入一部分，其他部分则在执行过程中动态装入。包括请求页式管理和预调入页式管理。

为了配合动态页面管理，需要加入中断处理后的页表；加入改变位后的页表。

页号	页面号	中断位	外存始址
0			
1			
2			
3			

页号	页面号	中断位	外存始址	改变位

## 页式管理的优缺点

### 优点

- 不用连续存放，解决了碎片问题
- 实现了内外存的统一管理

### 缺点

- 要求硬件支持
- 增加系统开销
- 请求调页的算法如不当选择，可能抖动
- 最后一页有浪费。

## 五、段式与段页式管理

### 段式存储管理

1. 内存空间划分：动态异长，每区一段。
2. 进程空间划分：若干段，每段一个程序单位。

物理地址 = **段首址 + 段内地址**

### 所需表目

- (1) 段表：每进程一个

段号	段首址	段长度
0:	100k	40k
1:	200k	60k
2:	320k	80k
3:	300k	20k

(2) 空闲表：系统一个

array of (addr,size)

所需寄存器

(1) 段表首址寄存器：  
b 系统一个

(2) 段表长度寄存器：  
l 系统一个

(3) 快表：系统一组：

段号	段首址	段长度
...	...	...
s	b'	l'
...	...	...

段式与段页式管理

段式优于页式

- 便于共享和保护

页式优于段式

- 消除“碎片”问题

段页式：结合二者优点

- 每个进程包含若干段
- 每个段包含若干页

【练习】分页式存储管理中，地址转换工作是由（A）完成的。

- A、硬件
- B、地址转换程序
- C、用户程序
- D、装入程序

## 6、文件系统

- 文件系统的概念
- 文件的逻辑结构与存取方法

- 文件的物理结构与存储设备
- 文件存储空间管理
- 文件目录管理
- 文件系统的概念

## 一、文件与文件系统的概念

**文件：**一组赋名的相关联字符流的集合，或相关联记录的集合。

**文件系统：**操作系统中与管理文件有关的软件与数据。它负责用户建立、撤销、读写、修改和复制文件，复制按名存取和存取控制

## 文件的分类

按文件的性质和用途分类

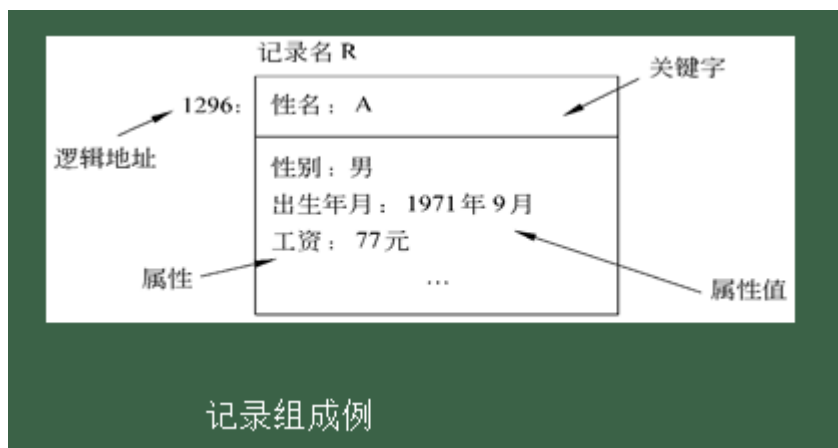
- 系统文件    —系统调用来使用；不可读写和修改
- 库文件       —可读写，执行；不可修改
- 用户文件    —授权使用

按组织形式分类

- 普通文件     —一般格式的文件
- 目录文件     —文件的目录信息构成的特殊文件
- 特殊文件

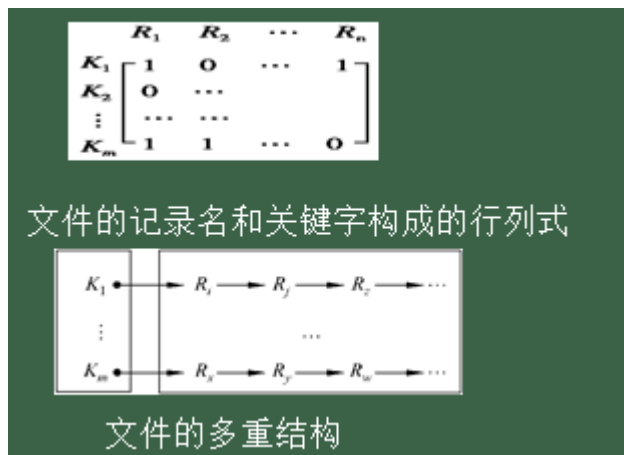
## 二、文件的逻辑结构

- 文件的组织称为文件的结构。文件的逻辑结构是指外部组织形式，亦是用户所看到的组织形式。可分为两类：
- 字符流式的无结构文件
- 访问以字节为基本单位。查找基本单位困难，文件管理简单
- 记录式的有结构文件
- 访问以记录为单位，记录可按不同的方式排列

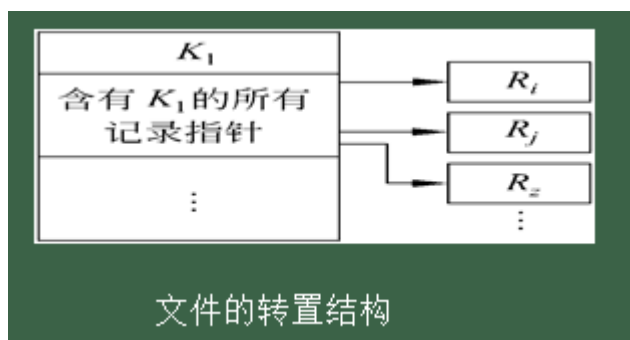




- 1. 连续结构：记录按生成的先后顺序连续排列
- 2. 多重结构：记录按关键字和记录名排列成行列式



- 3. 转置结构：记录按关键字和记录指针锁定
- 4. 顺序结构：记录按关键字的先后顺序排列



### 三、文件的存取方法

- 文件的存取是指文件的访问方式，就是要找到文件内容所在的逻辑地址。即用户使用文件时按何种次序存取文件的各个信息项。一般文件的访问方法如下：
  - 顺序存取
  - 按文件的逻辑地址顺序存取文件各个信息项
  - 随机存取
  - 按记录编号存取，或按存取指令把读写指针移到欲读写处
  - 按关键字存取
  - 常用于 DBMS 中，按照关键字和记录名存取

### 四、文件的物理结构

- 文件的物理结构是指文件的内部组织形式，是指文件在物理存储设备上的组织形式。

- 常用的文件物理结构：
- 连续文件
- 逻辑上连续的文件依次放入连续的物理块中
- 串联文件
- 逻辑上连续的文件放入不连续的物理块中
- 索引文件
- 增加索引表，说明逻辑块与物理块的对应，文件说明
- 信息项给出索引表的物理地址。

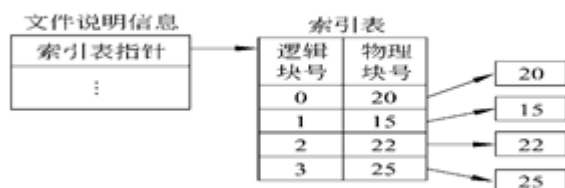
### 连续结构



### 串联结构



### 索引结构

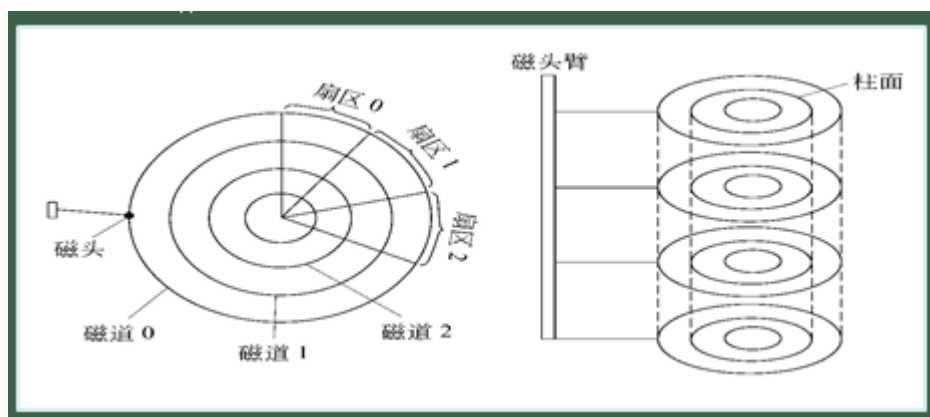


## 五、文件存储设备

- (1) 顺序存取设备

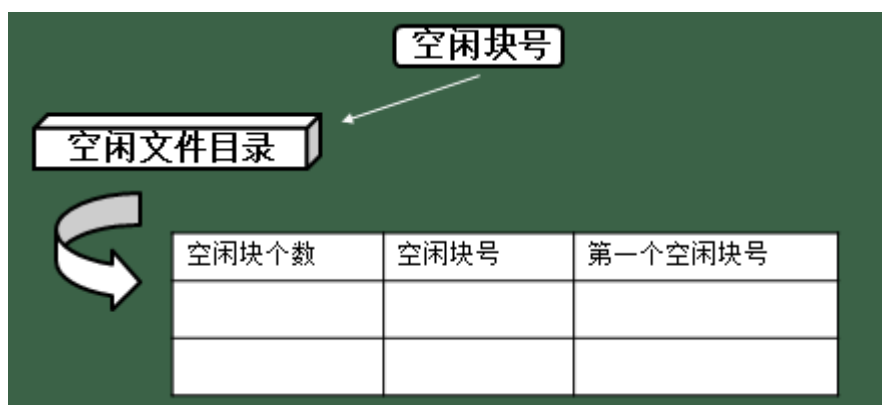


- (2) 直接存取设备
- 磁盘



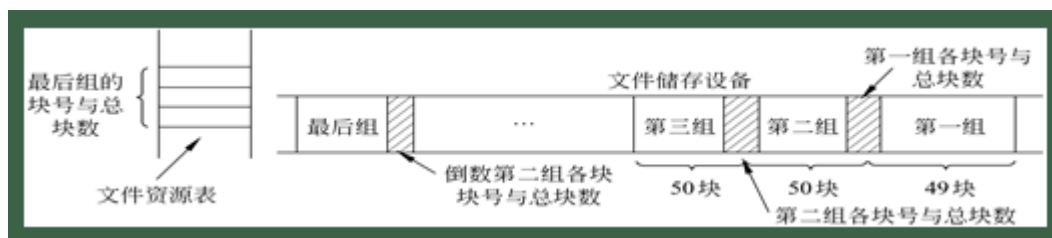
## 六、文件存储空间管理

- 文件存储设备是分成若干个大小相等的物理块，并以块为单位来交换信息的，因此，文件存储空间的管理实质是一个空闲块的组织和管理问题，它包括空闲块的组织，空闲块的分配与回收等问题。
- 空闲块管理的方法有 3 种：
- (1) 空闲文件目录



- (2) 空闲块链
- 把文件存储设备上的所有空闲块链接在一起。

- 空闲区大小顺序链接；空闲区释放先后
- 顺序链接；成组链接。



### (3) 位示图

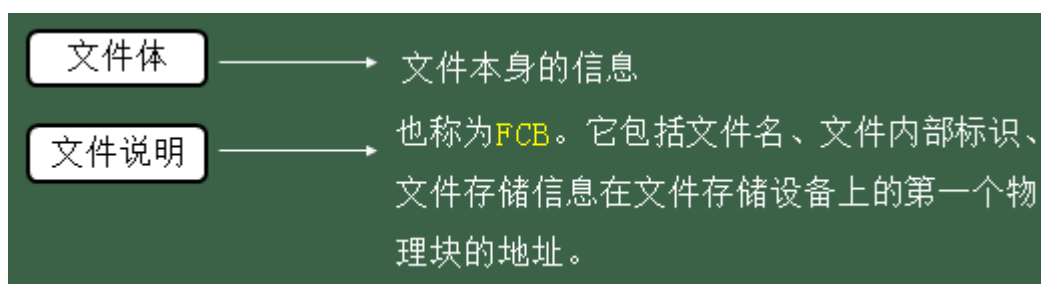
每个文件存储设备建立一张位示图。

用一个 bit 代表一个块的状态，0 表空闲，1 表占用。(多单元)



## 七、文件目录管理

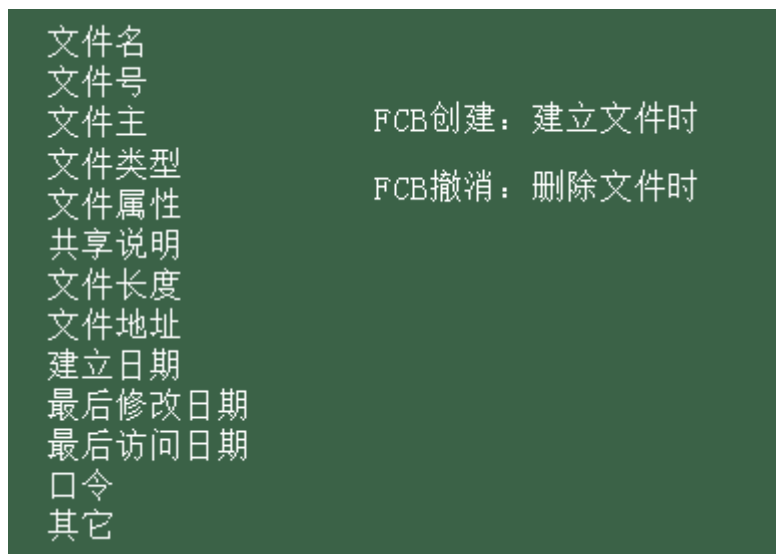
- 文件名和对该文件实施控制管理的控制管理信息称为该文件的文件说明，并把一个文件说明按一定的逻辑结构存放到物理存储块的一个表目中。
- 文件的组成



- 文件目录
- 文件控制块与目录项
- 文件控制块(FCB)
- 文件存在的标志，保存系统管理文件需要的全部信息
- 目录项
- 目录文件中的一项，内容为 FCB

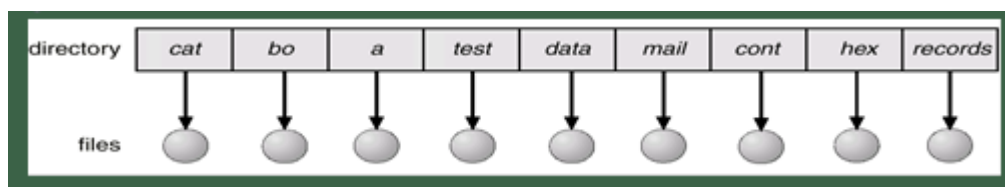
- 文件目录与目录文件
- 文件目录
- 用于检索文件的目录
- 目录文件
- 内容为目录项的文件

### FCB (File Control Block)



### 单级目录(Single-Level Directory)

A single directory for all users.

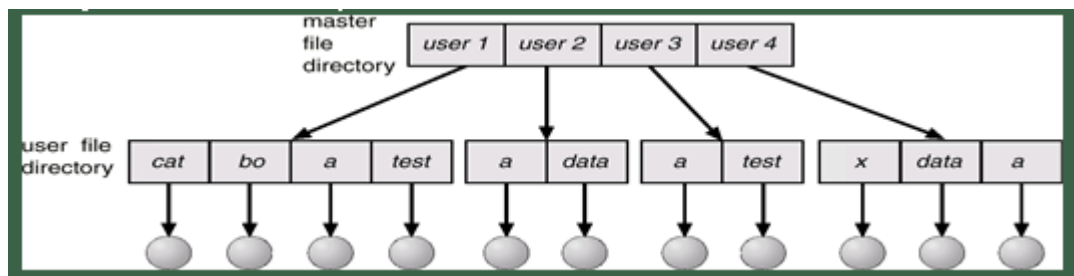


缺点:

1. 命名冲突问题
2. 搜索效率问题

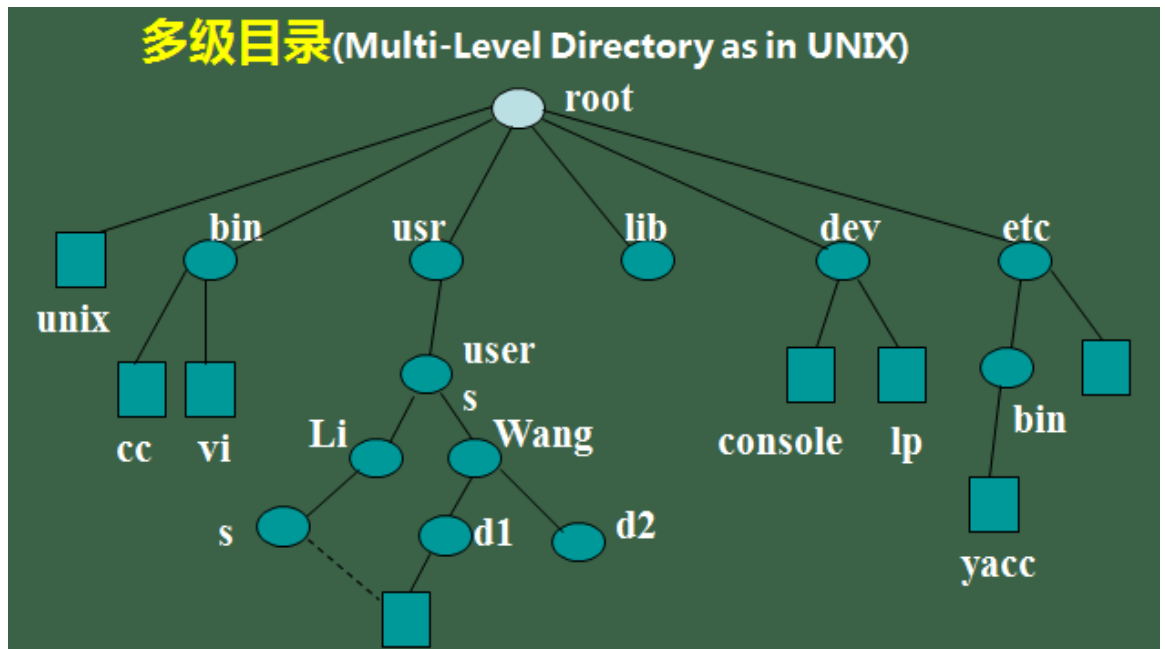
### 两级目录(Two-Level Directory)

Separate directory for each user.



特点:

1. 路径问题
2. 实现了文件重名和共享问题
3. 搜索效率高



## 目录管理

把当前正在使用的那些文件的目录表目复制到内存中。系统提供有关的目录文件复制到内存的指定区；以及当用户不再访问有关信息文件时删去有关目录文件的内存副本。

【练习】对记录式文件，操作系统为用户存取文件信息的最小单位是（C）。

- A、字符
- B、数据项
- C、记录
- D、文件

【练习】UNIX 系统中，文件存贮器的管理采用的是：（C）。

- A、位图法
- B、空闲块表法
- C、成组连接法
- D、单块连接法

## 7、设备管理

### 一、设备管理的概述

设备指：除 CPU 和内存之外的设备。包括输入输出设备、外存设备、终端设备和脱机设备。

#### 1、设备的分类

按设备的从属关系分类

系统设备：

在操作系统生成时已登记于系统中的各种标准设备。（如终端、打印机、磁盘机等）

用户设备：

由用户提供设备及其处理程序，并通过适当的手段把它们纳入系统，由系统实施管理。（如 A/D，D/A 转换器，CAD 所用专用设备）

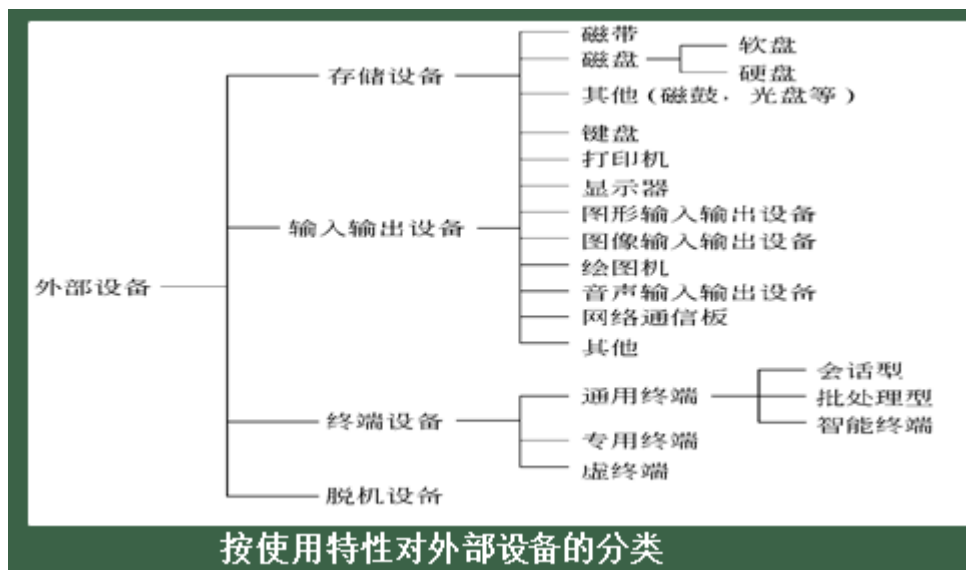
按设备的数据组织方式分类

#### 1)块设备：

以数据块为单位来组织和传送数据的设备，如磁盘、磁带等。

#### 2) 字符设备：

以单个字符为单位来传送信息的设备，如终端、打印机等。



- 2 设备管理的目标与功能
- 操作系统设备管理的目标
- 1) 提高外围设备的使用效率
- 2) 为用户提供方便、统一的界面
- 设备管理的功能
- 1) 设备分配
- 2) 缓冲区管理
- 3) 实现物理 I/O 操作
- 4) 提供进程管理系统的接口

## 二、数据传输控制方式

- 外围设备与内存或 CPU 之间的常用数据传送控制方式：
  - (1) 程序直接控制方式
  - (2) 中断控制方式
  - (3) DMA 方式
  - (4) 通道方式
- ① 程序直接控制方式
- 优缺点：
  - 优点：控制简单、硬件支持少
  - 缺点：串行、无法发现错误、只能和一台外围设备交换。



- 适用于 CPU 速度慢、外围设备少的系统
- ② 中断方式
- 优缺点：
  - 优点：CPU 利用率高、实现并行操作
  - 缺点：消耗 CPU 的处理时间数据丢失
- ③DMA 方式
- 优缺点：
  - 优点：克服了中断方式存在的问题
  - 缺点：CPU 对外围设备的管理趋向复杂、不是很经济。
- ④通道方式
- 独立于 CPU 的负责输入输出控制的处理机，控制设备与内存直接进行数据交换。

### 三、中断技术

- 中断的概念
- 中断装置
- 中断处理程序

#### ①中断的概念

处理机在运行过程中，出现了某一事件，必须中止正在运行的程序，转去处理这个事件，然后再返回原来运行的程序，这一过程称为中断。

中断系统：

- 中断装置(硬件)
- 中断处理程序(软件)

#### ②中断装置

发现并响应中断的硬件机构

- 识别中断源，当有多个中断源时，按紧迫程度排队；
- 保存现场；
- 引出中断处理程序。

#### (1) 中断源与中断字

- 中断源
  - 引起中断的事件。
- 中断寄存器

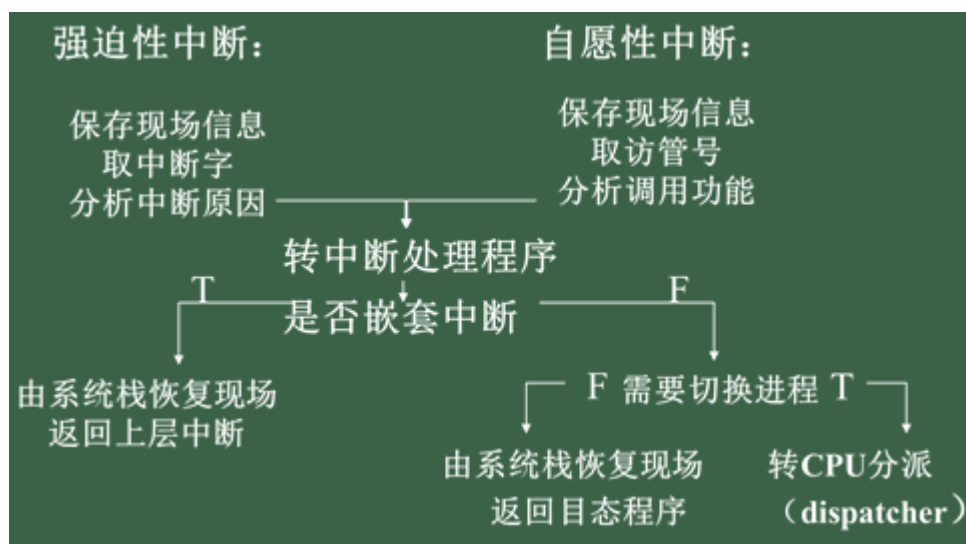
- 保存与中断事件相关信息的寄存器。

- 中断字
  - 中断寄存器的内容。
- 例：IO 中断：设备状态寄存器。

## (2) 中断类型与中断向量

- |  |   |
|--|---|
| <ul style="list-style-type: none"> <li>• 强迫性中断               <ul style="list-style-type: none"> <li>- 运行程序不期望的                   <ul style="list-style-type: none"> <li>• 时钟中断</li> <li>• IO中断</li> <li>• 控制台中断</li> <li>• 硬件故障中断                       <ul style="list-style-type: none"> <li>- power failure</li> <li>- 内存校验错</li> </ul> </li> </ul> </li> <li>• 程序性中断                   <ul style="list-style-type: none"> <li>- 越界，越权，缺页</li> <li>- 溢出，除0</li> <li>- 非法指令</li> </ul> </li> </ul> </li> </ul> | <ul style="list-style-type: none"> <li>• 自愿性中断               <ul style="list-style-type: none"> <li>- 运行程序期望的                   <ul style="list-style-type: none"> <li>• 系统调用</li> <li>• 访管指令</li> </ul> </li> <li>- 系统调用                   <ul style="list-style-type: none"> <li>• <code>fd=open(fname, mode)</code></li> </ul> </li> <li>- 访管指令                   <ul style="list-style-type: none"> <li>• 准备参数</li> <li>• <code>svc n</code></li> <li>• 取返回值</li> </ul> </li> </ul> </li> </ul> |
|--|---|

## ③中断处理程序



## (1) 中断类型与中断向量

- 中断向量：中断处理程序的运行环境与入口地址（PSW，PC）
  - 每类中断事件有一个中断向量，

- 中断向量的存放位置是由硬件规定的,
- 中断向量的内容是 OS 在系统初始化时设置好的。

## (2) 中断嵌套与系统栈

- 一般原则:
  - 高优先级别中断可以嵌入低优先级中断
- 实现方法:
  - 中断响应后立即屏蔽不高于当前中断优先级的中断源。
- 进入中断后一般需要进一步保存现场
- 关中断（屏蔽所有中断）
- 进一步保存现场（地址寄存器，通用寄存器等）
- 开中断（或开放高优先级中断）
- .....
- 中断处理
- .....
- 恢复现场
- 中断返回

## (3) 中断优先级与中断屏蔽

- 中断优先级:
  - 硬件规定的中断响应次序，依据:
    - 紧迫程度；
    - 处理时间。
- 中断屏蔽:
  - 高优先级中断事件处理不受低优先级中断打扰；
  - 程序调整中断响应次序。

## (4) IO 中断处理

- 正常结束
  - 继续传输；
  - 唤醒相关进程。
- 传输错误

- 复执 (eg. 3 次);
- 报告系统操作员。

#### (5) 程序性中断的处理

- 只能由操作系统处理的中断
  - 影响系统或其它进程
    - 越界, 非法指令, (处理: 终止进程、调试)
  - 需要系统管理或协助
    - 页故障, 缺段, (处理: 动态调入)
- 可以由用户自己处理的中断
  - 不影响系统和其它进程
    - 除 0, 溢出, (处理: 用户处理, 或 OS 处理)

#### 【练习】

磁盘属于\_\_\_\_ (1) \_\_\_\_, 其信息的存取是以\_\_\_\_ (2) \_\_\_\_为单位的; 磁盘的 I/O 控制主要采取\_\_\_\_ (3) \_\_\_\_方式, 打印机的 I/O 控制主要采取\_\_\_\_ (4) \_\_\_\_方式。

A: (1) 字符设备 (2) 独占设备 (3) 块设备 (4) 虚拟设备

B: (1) 位 (2) 字节 (3) 帧 (4) 固定长数据块

C: (1) 程序 I/O 方式 (2) 程序中断 (3) DMA (4) SPOOLing

D: (1) 程序 I/O 方式 (2) 程序中断 (3) DMA (4) SPOOLing

## ■ 华图网校介绍

华图网校（V.HUATU.COM）于2007年3月由华图教育投资创立，是华图教育旗下的远程教育高端品牌。她专注于公职培训，目前拥有遍及全国各地500万注册用户，已成为公职类考生学习提高的专业门户网站。

华图网校是教育部中国远程教育理事单位。她拥有全球最尖端高清录播互动技术和国际领先的网络课程设计思想，融汇华图教育十余年公职辅导模块教学法，凭借强大师资力量与教学资源、利用教育与互联网的完美结合，真正为考生带来“乐享品质”的学习体验，通过“高效学习”成就品质人生。

华图网校课程丰富多元，涵盖公务员、事业单位、招警、法院、检察院、军转干、选调生、村官、政法干警、三支一扶、乡镇公务员、党政公选等热门考试、晋升及选拔。同时，华图网校坚持以人为本的原则，不断吸引清华、北大等高端人才加入经营管理，优化课程学习平台，提升用户体验，探索网络教育新技术和教学思想，力争为考生提供高效、个性、互动、智能的高品质课程和服务。

华图网校将秉承“以教育推动社会进步”的使命，加快网站国际化进程，打造全球一流的网络学习平台。

我们的使命：以教育推动社会进步

我们的愿景：德聚最优秀人才，仁就基业长青的教育机构

我们的价值观：诚信为根、质量为本、知难而进、开拓创新。

- 咨询电话：400-678-1009
- 听课网址：v.huatu.com（华图网校）