



操作系统

第二十一讲

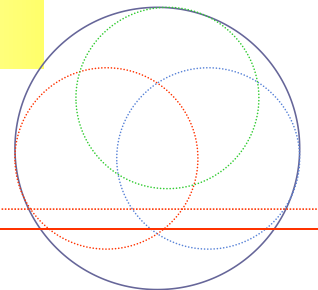
张涛

第六章 输入/输出系统

概述

I/O系统硬件特点

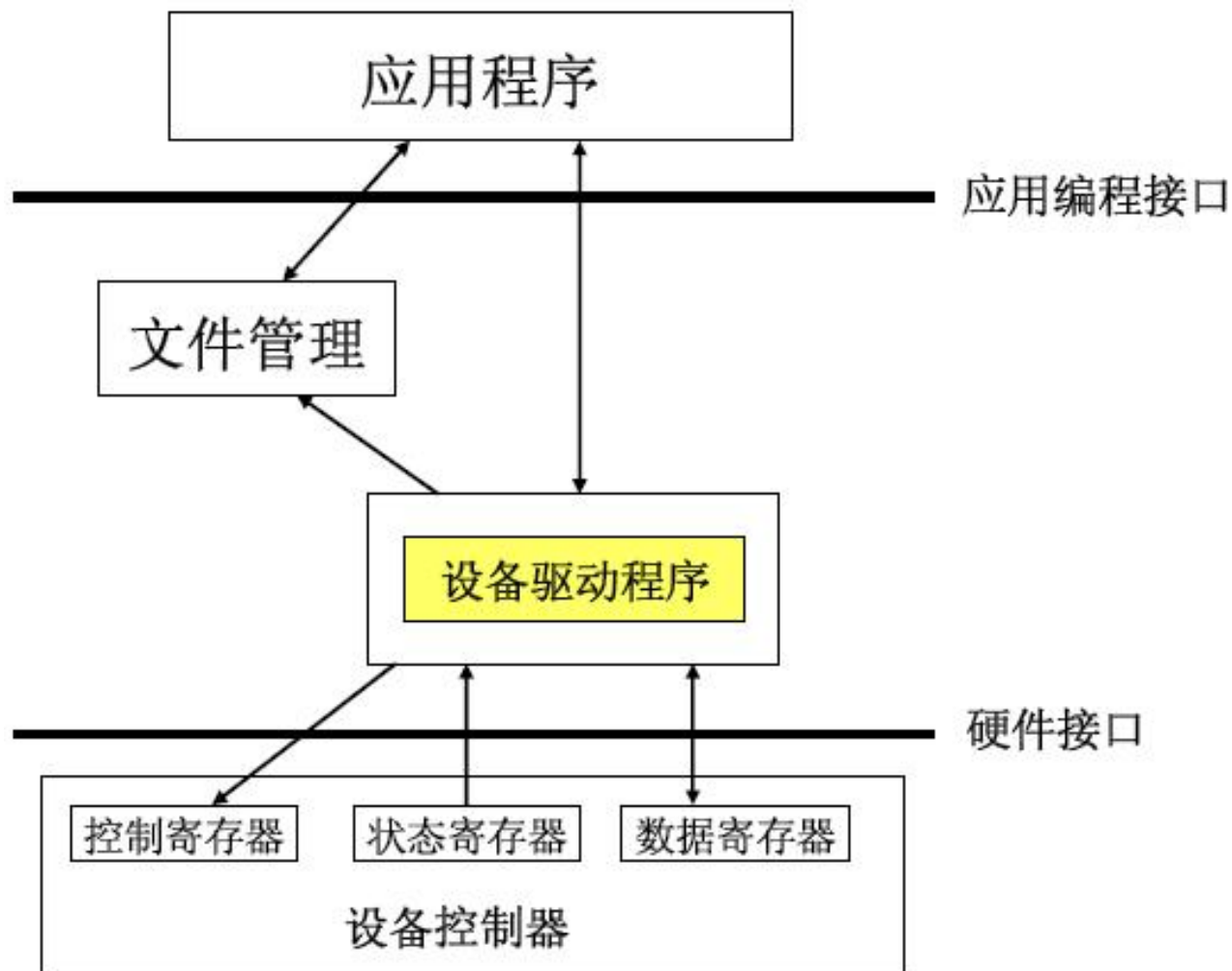
I/O控制方式



I/O的特点

- I/O性能经常成为系统性能的瓶颈
 - CPU性能不等于系统性能
 - CPU性能越高，与I/O差距越大
 - 进程切换多，系统开销大
- 操作系统庞大复杂：资源多、杂，并发，均来自I/O
- 与其他功能联系密切，特别是文件系统

I/O管理示意图



6.1 概述

- 设备的分类
- 设备管理的目标
- 设备管理的功能
- 设备管理数据结构

6.1.1 设备的分类

1、按传输速率分

- 低速设备
- 中速设备
- 高速设备

2、按信息交换的单位分类

- 字符设备：速率较低、中断驱动。
- 块设备：速率高（几兆）、可随机访问任一块、DMA方式驱动。

3、按资源管理方式分类

- 独占型设备
- 共享型设备
- 虚拟设备

4、按外部设备的从属关系分

- 系统设备
- 用户设备

6.1.2 设备管理的目标

- 1、实现设备独立性
- 2、提高设备利用率
- 3、设备的统一管理

- 速度
- 传递单位
- 操作方法和特性
- 出错条件

6.1.3 设备管理的功能

- 1、监视系统中所有设备的状态
- 2、设备的分配
- 3、I/O控制

6.1.4 设备管理数据结构

■ 设备控制块(DCB)

- 设备标识符
- 设备属性
- 设备I/O总线地址
- 设备状态
- 等待队列指针

DCB的结构

设备名↗
设备属性↗
设备状态↗
设备在 I/O 总线上的地址↗
等待队列指针↗

6.2 I/O系统硬件特点

■ 设备组成

- (1) 物理部分：物理设备机械部分，设备本身
- (2) 电子部分：设备控制器

■ 电子部分（设备控制器）完成的工作

- （端口）地址译码
- 接受主机发来的数据和控制信号，向主机发送数据和状态信号
- 将计算机的数字信号转换成机械部分能识别的模拟信号，或反之
- 实现设备内部硬件缓冲、数据加工等提高性能或增强功能

6.2.1 设备接口

- 操作系统将命令写入控制器的接口寄存器（或接口缓冲区）中，以实现输入 / 输出，并从接口寄存器读取状态信息或结果信息
- 接口内的组成部分：
 - 设备识别线路
 - 命令寄存器
 - 数据缓冲寄存器（输入/输出）
 - 控制寄存器
 - 状态寄存器

6.2.2 端口编址方法

- **内存映像编址（内存映像I/O模式）**：分配给系统中所有端口的地址空间与内存的地址空间统一编址

- **优点**

- 凡是可对存储器操作的指令都可对I/O端口操作
- 不需要专门的I/O指令
- I/O端口可占有较大的地址空间

- **缺点**：占用内存空间

- **I/O独立编址（I/O专用指令）**：分配给系统中所有端口的地址空间完全独立，主机使用专门的I/O指令对端口进行操作

- **优点**

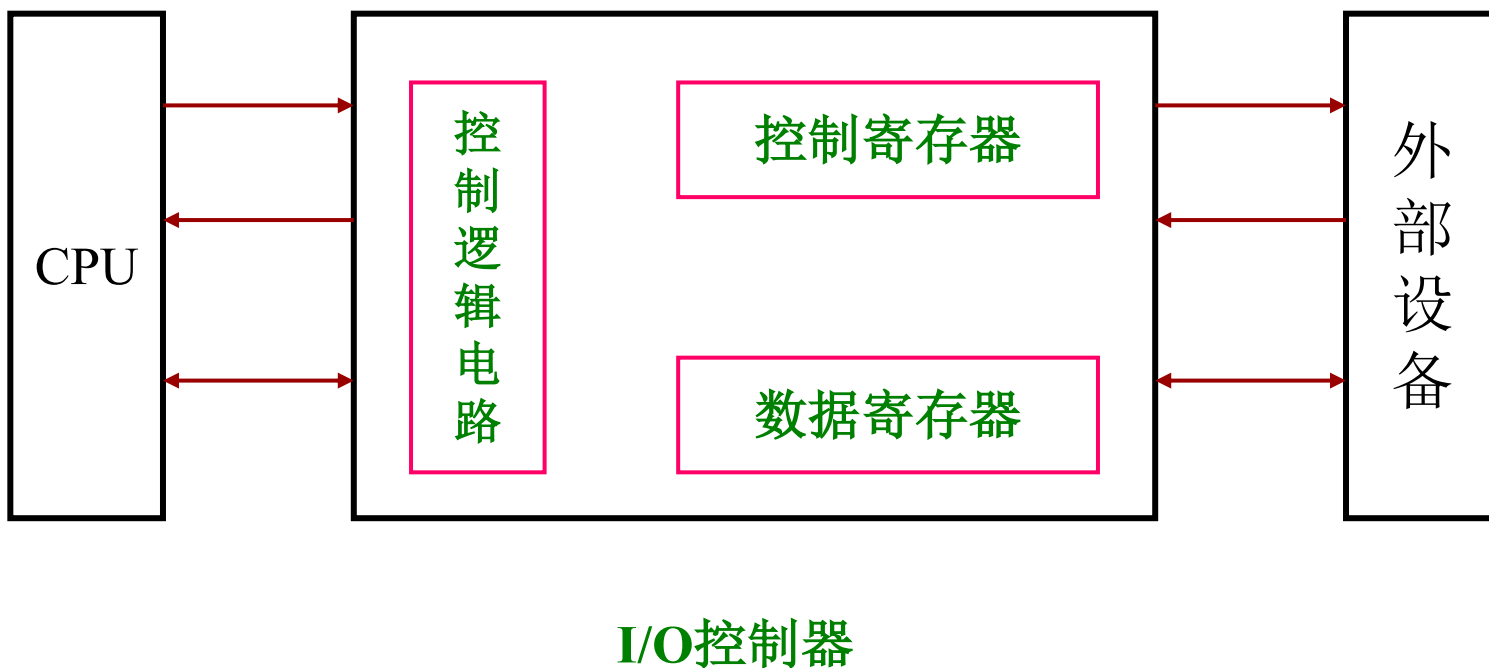
- 外部设备不占用内存的地址空间
- 程序设计时易于区分是对内存操作还是对I/O端口操作

- **缺点**：对I/O端口操作的指令类型少，操作不灵活

6.3 I/O控制方式

- (1) 程序直接控制方式(循环查询I/O方式)
- (2) I/O中断方式
- (3) 直接存储器存取方式(direct memory access, DMA)
- (4) I/O通道控制方式(I/O channel control)
- (5) 外围处理机输入输出方式(peripheral processor unit)

6.3.1 循环查询I/O方式



浪费大量CPU时间

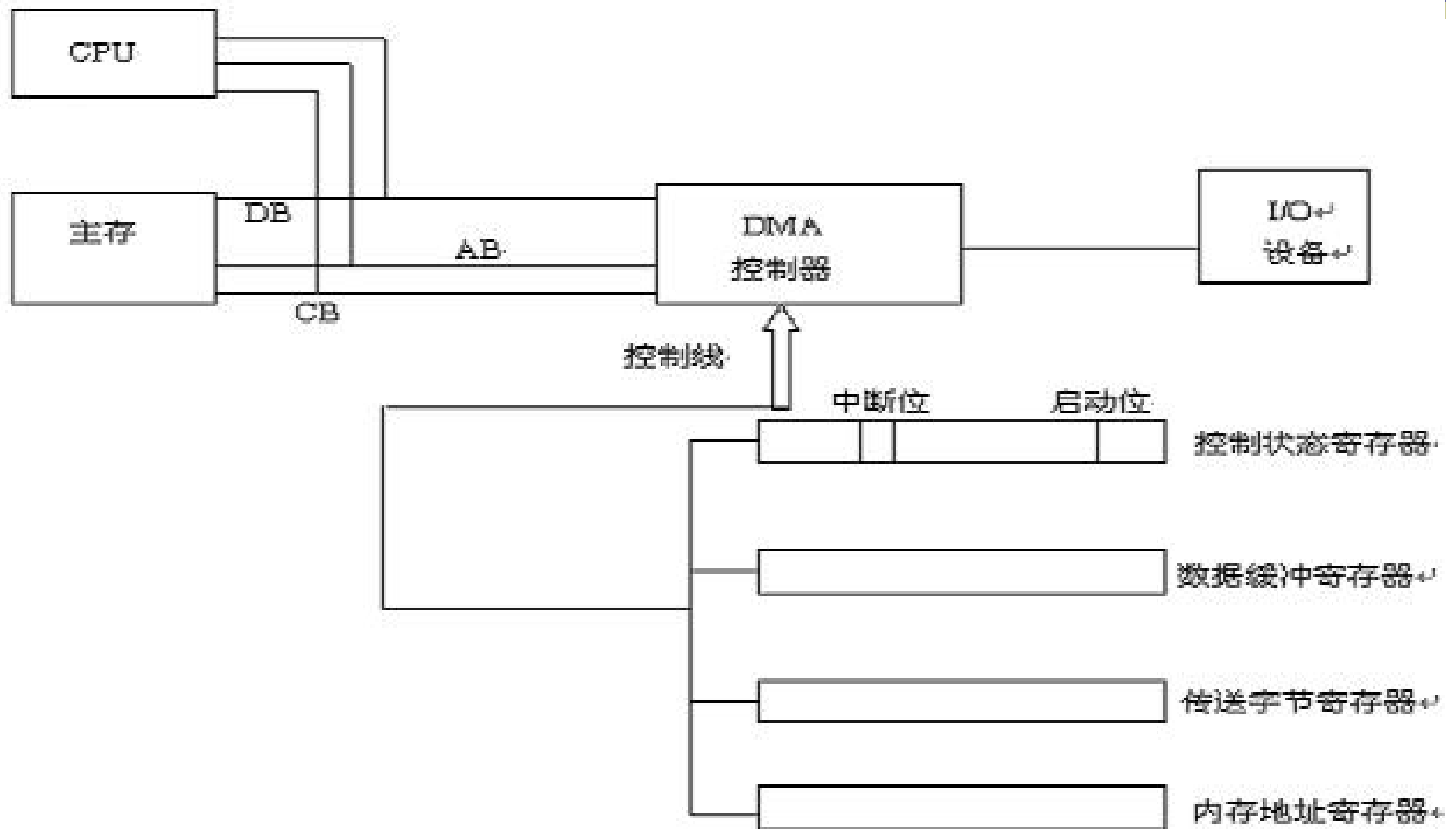
6.3.2 I/O中断方式

- I/O控制器能发中断
- 工作过程

同前相比，CPU利用率大大提高。

缺点：数据缓冲寄存器每满一次都要中断一次，如果设备较多时，中断次数会很多，使CPU的计算时间大大减少。

6.3.3 DMA方式



工作过程

- 1、当进程要求输入时，把要求传送的内存始址（M）和要传的字节数送入DMA的内存地址寄存器和传送字数寄存器
- 2、把启动位置1。设备开始工作。进程（A）挂起。调度另一进程（B）
- 3、一批数据输入完成后，DMA中断B，转向中断处理程序。
- 4、中断处理程序唤醒A，返回B的断点继续执行。
- 5、以后OS调度A运行时，A从M处取数据处理。

DMA方式与中断的主要区别

■ 中断时机

- 中断方式是在数据缓冲寄存器满后，发中断请求，CPU进行中断处理
- DMA方式则是在所要求传送的数据块全部传送结束时要求CPU进行中断处理

■ 数据传输

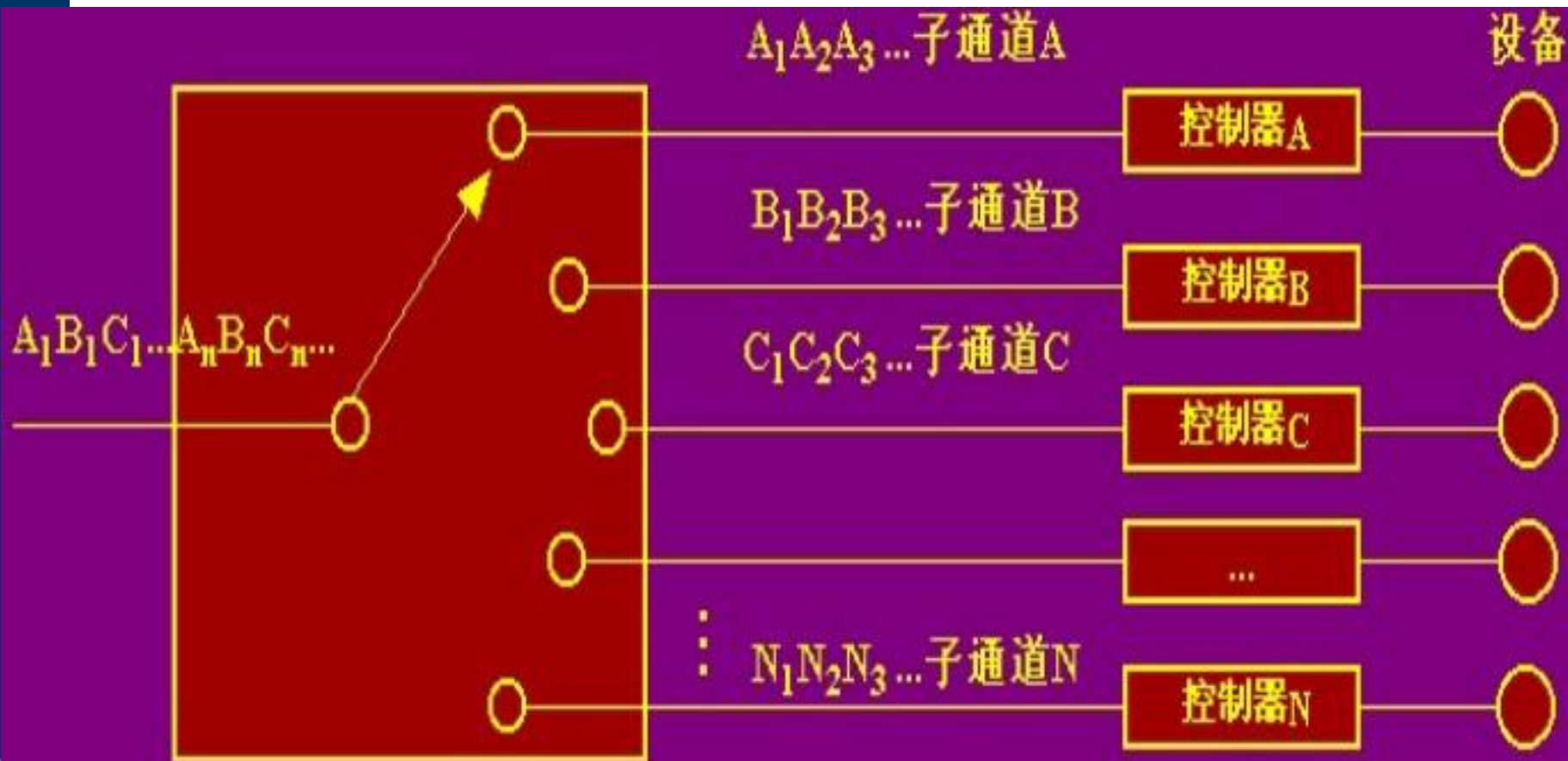
- 中断方式的数据传送由CPU控制完成
- DMA方式是在DMA控制器的控制下不经过CPU控制完成的

6.3.4 通道方式

- **通道**：在CPU的控制下独立地执行通道程序，对外部设备的I/O操作进行控制，以实现内存与外设之间成批的数据交换。
- 通道程序是由通道指令组成，一个通道可以分时的方式执行几道程序。每道程序控制一台外部设备，因此**每道通道程序称为子通道**。

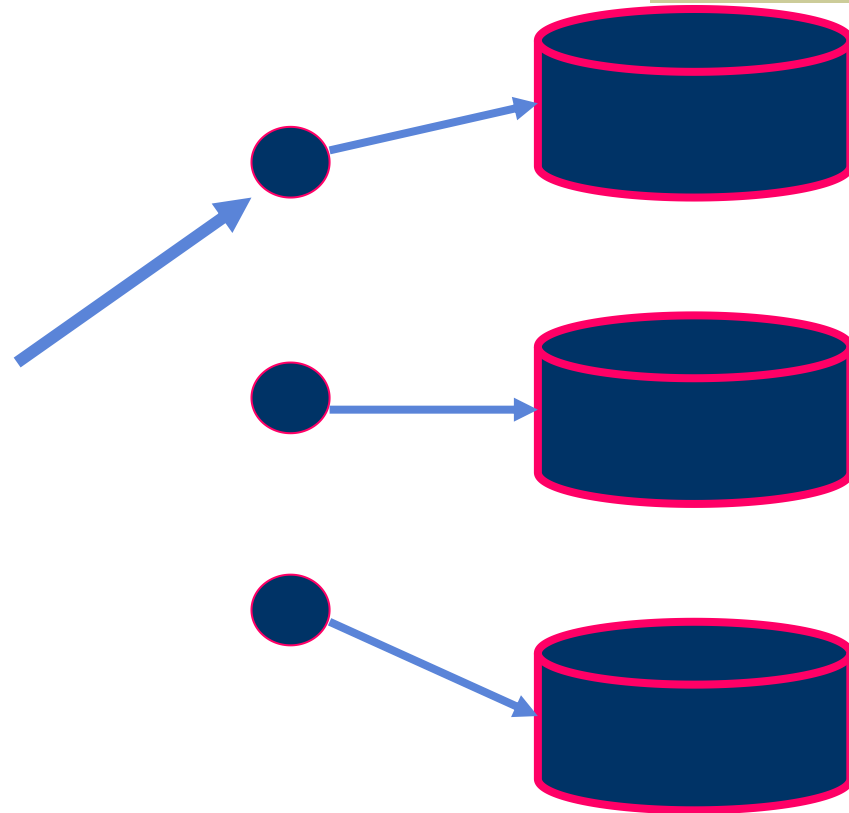
通道的种类

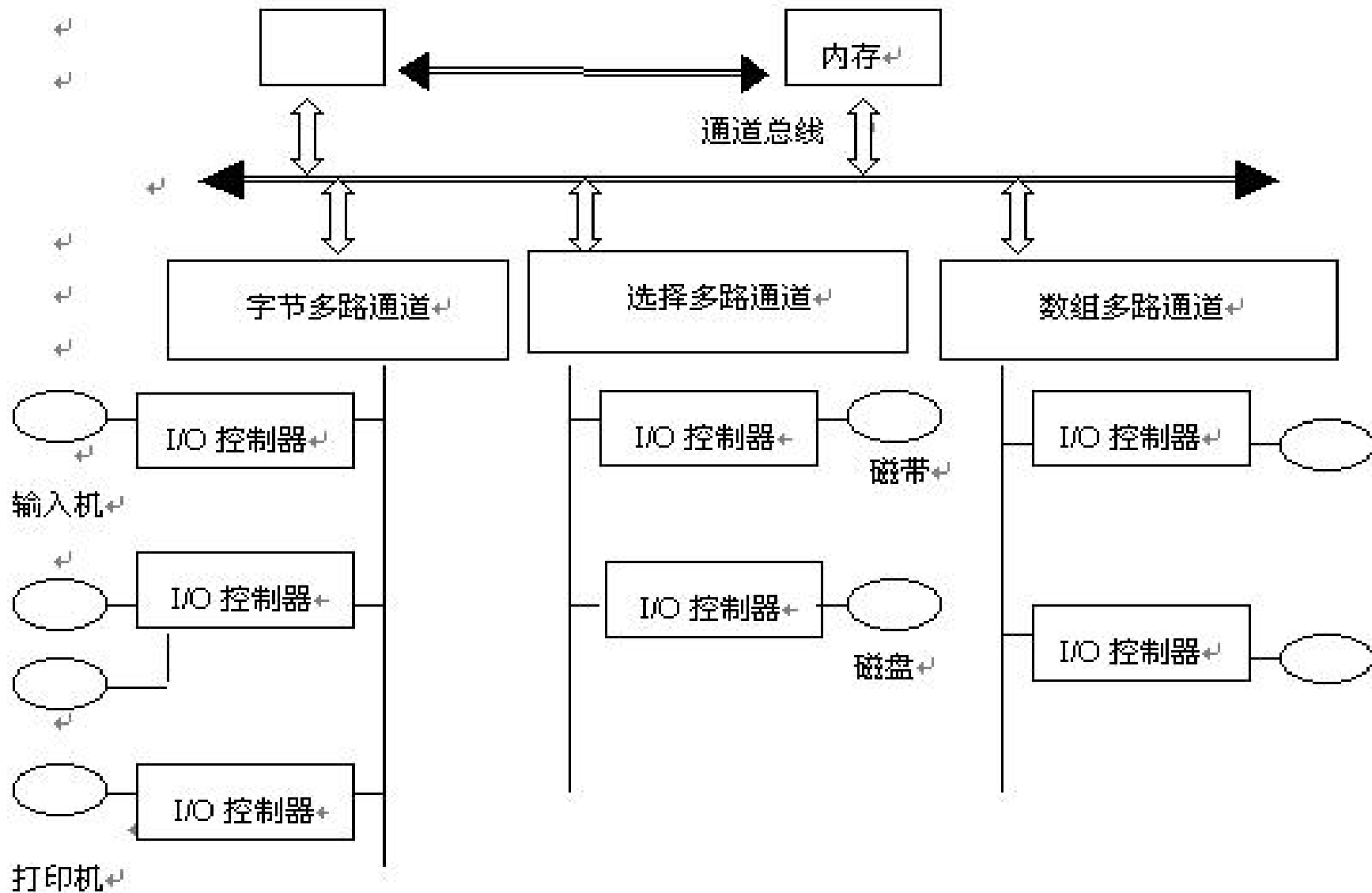
- 字节多路通道
- 数据选择通道
- 数据多路通道
- 在一大型系统中可以同时存在这三种类型的通道以便控制各种不同类型的设备。



字节多路通道的工作原理

选择通道





I/O系统结构

通道的工作过程

- 某进程在运行过程中，若提出了I/O请求，则通过系统调用进入操作系统。系统首先为I/O操作分配通道和外设，然后按I/O请求生成通道程序并存入内存，把起始地址送入通道的首地址寄存器（CAW），接着CPU发出启动通道的指令。
- 中央处理机启动通道后，通道的工作过程为：
 - 根据CAW，从内存取出通道指令，送入通道控制字寄存器（CCW），并修改CAW，使其指向下一条通道指令。
 - 执行CCW中的通道指令，进行实际的I/O操作，执行完毕后如果还有下一条指令，则返回前一步，否则转下一步。
 - 发出中断信号通知CPU通道程序已执行完成。

通道的发展

- 新的通道思想综合了许多新的技术
- 在个人计算机中，芯片组中有专门的I/O处理芯片，称为IOP (IO Processor)，发挥通道的作用
- IBM 390 中，沿用了输入输出通道概念
- IBM于1998年推出光纤通道技术（称为FICON），可通过 FICON 连接多达127个大容量I/O设备。传输速度是333MHz / s，未来将达到1GHz / s。

What you need to do?

- 课后3、4题

See you next time!