

I/O 管理

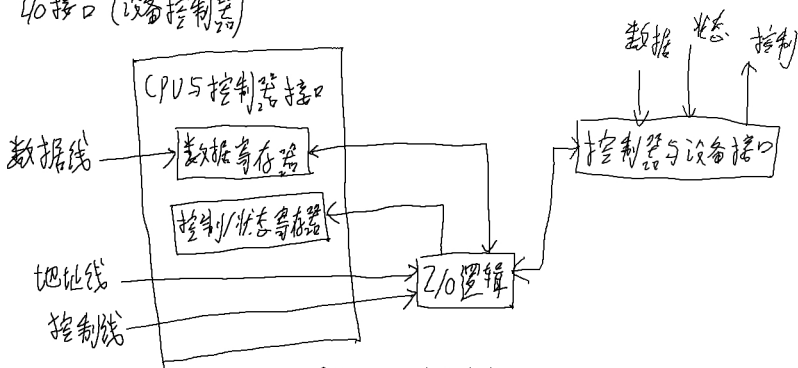
I/O 设备分类

按信息交换单位 { 块设备
字符设备

传输速率高、可寻址

速率低、不可寻址，常用中断 I/O 方式

I/O 接口 (设备控制器)



功能 { 接收和识别 CPU 发来的命令
数据交换
标识和报告设备状态
地址识别
数据缓冲
差错控制

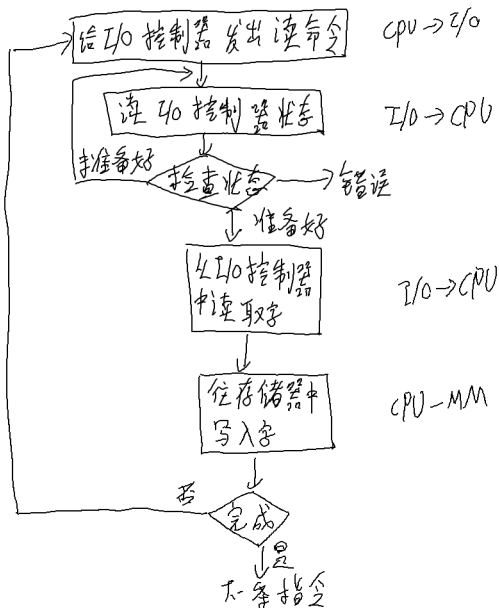
I/O 端口 (设备控制器中可被 CPU 直接访问的寄存器)

{ 数据寄存器
状态寄存器
控制寄存器

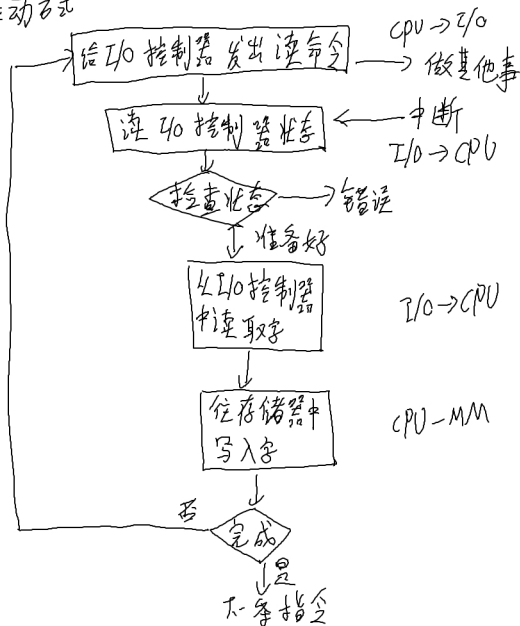
端口编址方法 { 独立编址
统一编址 (内存映射 I/O)

I/O 控制方式 { 程序直接控制
中断驱动方式
DMA 方式

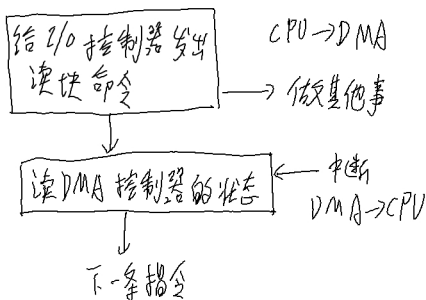
程序直接控制方式



中断驱动方式



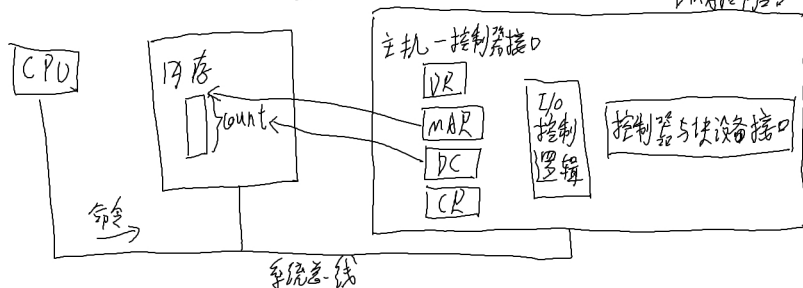
DMA 方式



DMA特点 {

- 基本单位是数据块
- 数据在设备与内存间直接传送
- 整块数据的传送在DMA控制器的控制下完成

DMA控制器



{

- CR 命令/状态寄存器
- MAR 内存地址寄存器
- DR 数据寄存器
- DC 数据计数器

I/O 软件层次结构

I/O 软件 {

- 用户层 I/O 软件
- 设备独立性软件
- 设备驱动程序
- 中断处理程序
- 硬件

应用程序 I/O 接口

- 字符设备接口
- 块设备接口
- 网络设备接口
- 阻塞/非阻塞

设备独立性软件

磁盘高速缓冲 { 在内存中开辟一个单独空间作 Disk Cache
缓冲池

缓冲区

引入目的

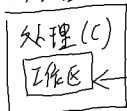
- 缓和 CPU 与 I/O 设备间的不匹配
- 减少对 CPU 的中断频率
- 解决数据粒度不匹配问题
- 提高 CPU 和 I/O 设备间的并行性

缓冲技术

- 单缓冲
- 双缓冲
- 循环缓冲
- 缓冲池

单缓冲

用户进程

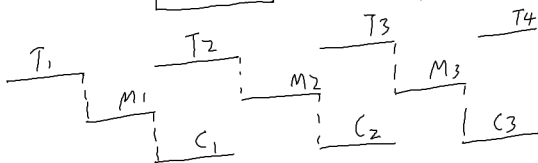


传送(M)

缓冲区

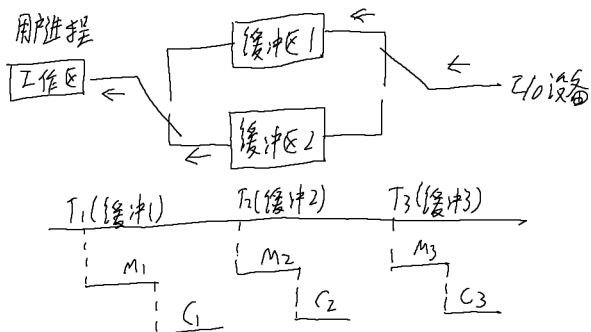
输入(T)

I/O 设备



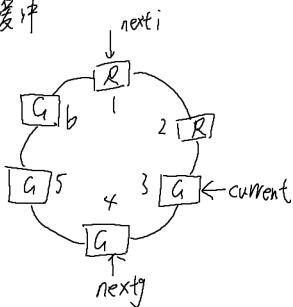
处理每块
数据时间
 $\max\{C, T\} + M$

双缓冲



处理一块数据时间 $\max\{C+M, T\}$

循环缓冲



R 空缓冲区
G 满缓冲区
C 现行工作缓冲区

缓冲池



设备分配

{ 独占式 使用设备 独占设备
 分时式 共享使用设备 共享设备
 以 Spooling 方式使用外部设备 虚拟设备

设备分配的数据结构

DCT	设备控制表
COCT	控制器控制表
CHCT	通道控制表
SDT	系统设备表

一个 DCT 对应 一个设备

DCT 集合

DCT1
DCT2
⋮
DCTn

设备类型 type

设备标识符 deviceid

设备状态 等待/不等待 忙/闲

指向控制器表的指针

重复执行次数或时间

设备队列的队首指针

COCT

$n:1$

CHCT

控制器标识符

controllerid

状态 忙/闲

与控制器连接的 CHCT 表指针

控制器队列的队首指针

控制器队列的队尾指针

通道标识符

channelid

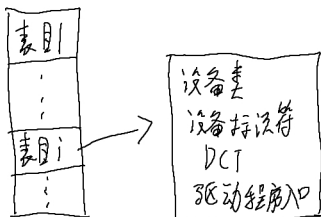
状态 忙/闲

与通道连接的 COCT 表首址

通道队列的队首指针

通道队列的队尾指针

SDT 整个系统只有一个



设备分配方式 { 静态分配 独占设备
动态分配 共享设备

动态设备分配算法 { 先请求先分配
优先级高者优先

安全性 { 安全分配 CPU与I/O串行
不安全分配 可能造成死锁

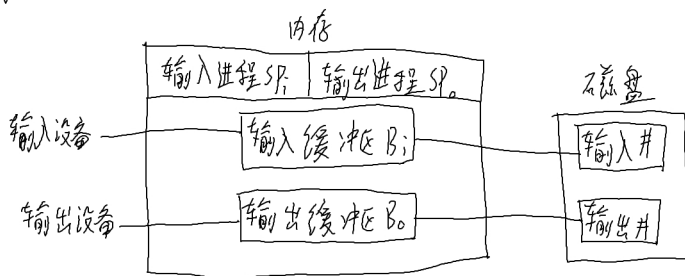
逻辑设备名 到 物理设备名 映射

LUT 逻辑设备表 { 整个系统设置一张 LUT
每个用户设置一张 LUT

LUT 表项

逻辑设备名	物理设备名	设备驱动程序入口地址
-------	-------	------------

Spooling 技术 (假脱机)



实例 共享打印机

当用户进程请求打印输出时

Spooling 管理进程完成两项任务

1. 在磁盘缓冲区中申请一个空闲盘块, 将要打印的数据送入其中暂存
2. 为用户进程申请一张空白的用户请求打印表, 并将用户打印的要求填入其中, 将该表挂到假脱机文件队列表上

完成任务后, 对用户进程而言打印任务已完成

真实打印操作是在打印机空闲且该打印任务在等待队列队首时进行

Spooling 特点 {

- 提高 I/O 速度
- 将独占设备改造为共享设备
- 实现虚拟设备功能

磁盘

磁盘地址 := <柱面号 · 盘面号 · 扇区号>

磁盘初始化

将磁盘分为扇区 \Rightarrow 低级格式化 (物理格式化)

OS { 将磁盘分为柱面组成的分区

逻辑格式化 (创建文件系统)

磁盘调度算法

一次磁盘读写操作时间 = 寻道时间 T_s + 旋转延迟时间 T_r + 传输时间 T_t

$T_s = m \times n + t_s$ 跨越 n 条磁道时间 + 启动磁头时间

$T_r = \frac{1}{2r}$ 旋转速度 r

$T_t = \frac{b}{rN}$ 每次读写字节数 b
磁盘每旋转数 r
一个磁道上的字节数 N

磁盘调度算法

{ FCFS
SSTF 最短寻找时间优先 可能饥饿
SCAN (电梯调度算法) > 需要到达端点
C-SCAN
Look
C-Look > 改进 SCAN、C-SCAN
不要到达端点

默认 SCAN、C-SCAN 是 Look、C-LOOK

优点

缺点

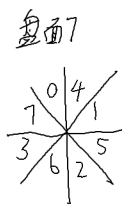
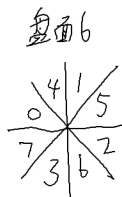
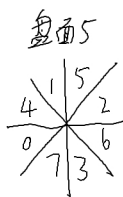
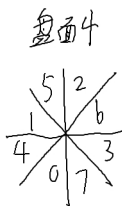
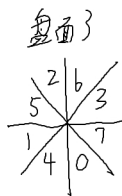
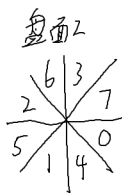
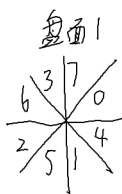
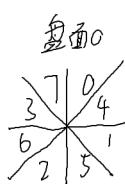
FDFS	公平、简单	平均寻道距离大
SSTF	性能好于FDFS	可能饥饿，不保证平均寻道时间最短
SCAN	寻道性能好，无饥饿	不利于远离磁道的一端
C-SCAN	消除对磁道两端请求的不公平	

减少旋转延迟时间方法

对扇区进行交替编号

对磁盘分组的的不同盘面错位命名

例



固态硬盘 SSD

基于闪存技术，需要擦写

数据以页为单位读写

磨损均衡技术 { 动态
静态

提高磁盘 I/O 速度方法 { 提前读
延迟写
虚拟盘