

# 一、实验内容

结合数据结构的相关知识,编写通道处理过程模拟程序。

## 二、实验目的

通过模拟实现通道处理过程，掌握通道技术

## 三、实验要求

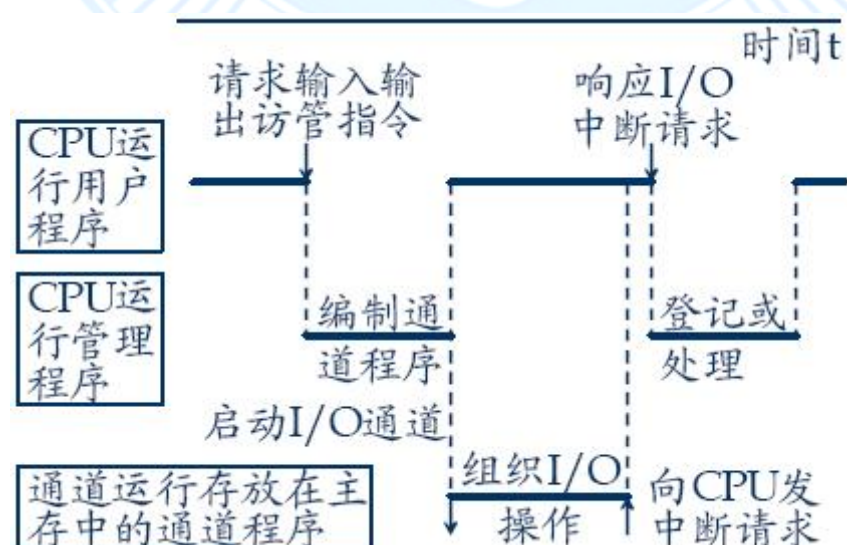
结合数据结构的相关知识,编写通道处理过程模拟程序。

1. 通道完成一次数据输入输出的过程需三步（如图一所示）：

(1) 在用户程序中使用访管指令进入管理程序，由 CPU 通过管理程序组织一个通道程序，并启动通道；

(2) 通道处理机执行通道程序，完成指定的数据输入输出工作；

(3) 通道程序结束后第二次调用管理程序对输入输出请求进行处理每完成一次输入输出工作，CPU 只需要两次调用管理程序，大大减少了对用户程序的打扰。



---

图 1 通道程序，管理程序 and 用户程序的执行时间关系

## 2. 通道的主要功能

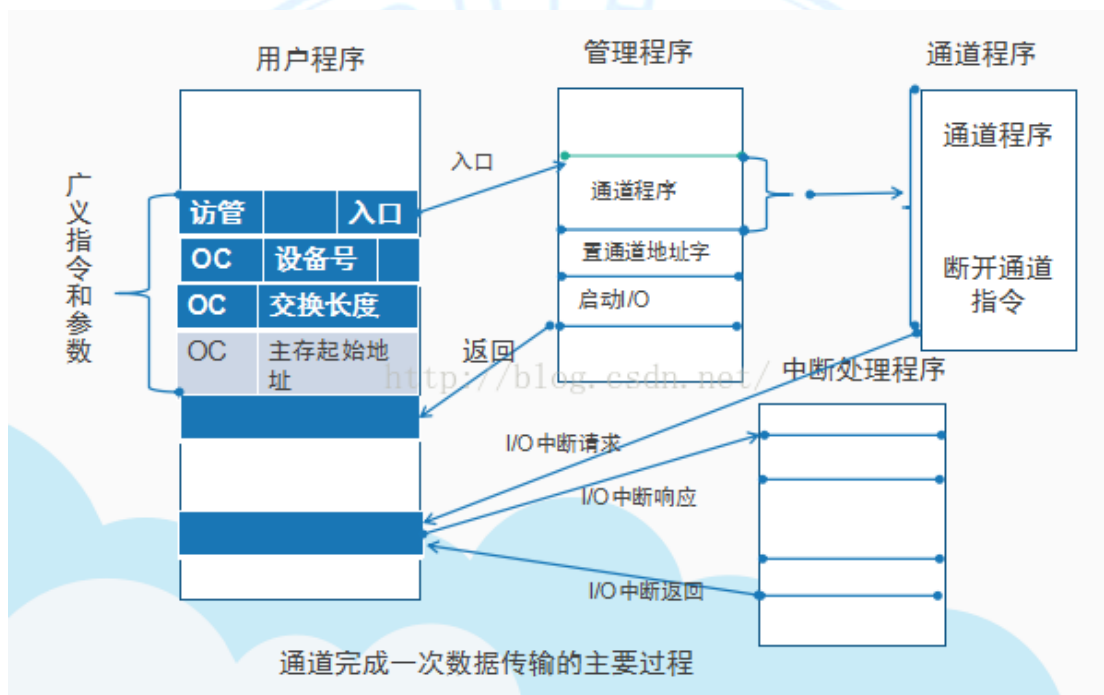
- (1) 接受 CPU 发来的指令，选择一台指定的外围设备与通道相连接
- (2) 执行 CPU 为通道组织的通道程序
- (3) 管理外围设备的有关地址
- (4) 管理主存缓冲区的地址
- (5) 控制外围设备与主存缓冲区间数据交换的个数
- (6) 指定传送工作结束时要进行的操作
- (7) 检查外围设备的工作状态，是正常或故障
- (8) 在数据传输过程中完成必要的格式的变换

## 四、实验分析与设计

对于工作方式、工作速度、工作性质不同的外围设备，采用三种常用的三种不同的输入输出方式：程序控制输入输出方式、中断输入输出方式以及DMA方式。程序控制方式不论何时何地都受cpu的控制，不能与cpu并行工作；中断方式虽然可以克服不能并行的缺点，但数据输入输出都要经过cpu；DMA方式需要cpu在开始时对其进行初始化，结束时要向cpu申请中断；

所以，为了把对外围设备的管理工作从cpu分离出来，产生了通道处理机技术。

通道处理机一次数据传送过程，如下：



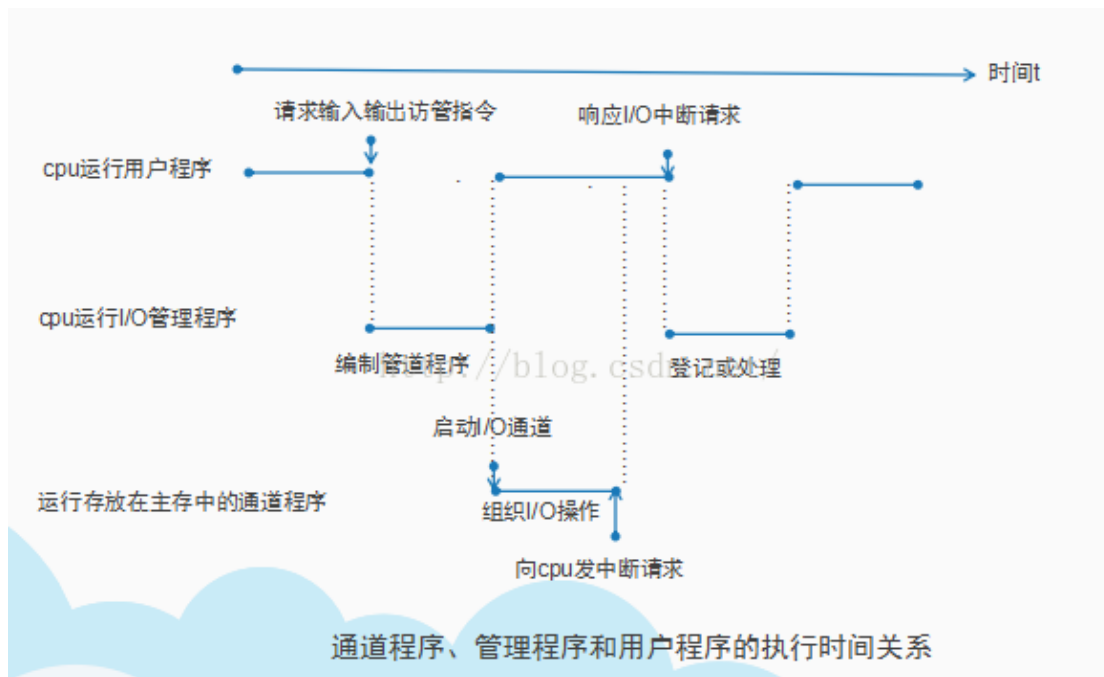
1、cpu执行到访管指令时，按其提供的入口地址，第一次进入管态(特权态)，将管理程序调出来执行。这个管理程序的任务是利用所带的参数来编制通道程序。

在通道程序编好并且存入主存中某个通道缓冲区时，置好相应的通道地址字（即就是在主存的地址），当管理程序执行到“启动I/O”的管态程序时，开始选择设备。

2、cpu选择指定的通道、子通道、设备控制器和设备后，发出启动命令，设备成功启动后，cpu就退出管态，返回目态（用户态）去执行其他的用户程序。

3、被启动的通道开始执行存放于通道缓冲区的通道程序具体组织I/O操作，直至执行到断开通道指令完成了通道程序后，又向cpu发出I/O中断请求。

4、cpu在响应I/O中断请求后，第二次进入管态，调出相应的管理程序，进行善后登记和处理后（正常结束则进行登记工作，异常情况则进行额外处理），重新返回目态。



其中，在cpu发中断请求和响应中断请求之间的时间是中断响应时间。

通道可划分为三类：字节多路通道，选择通道，数组多路通道。

对数据传送时间作以解释：

字节多路通道

$$np * (T_s + T_d)$$

字节多路通道，选择好设备后，传送一个字节就释放总线，所以传送n个字节，需要n次申请使用通道。

选择通道

$$p * (T_s + nT_d)$$

选择通道每选择好一台设备后，独占通道，n个字节传送完后才释放通道，所以每台设备传送n个字节只需一次申请使用通道总线

数组多路通道

$$\left\lceil \frac{n}{K} \right\rceil * p(T_s + KT_d)$$

数组多路通道选择好设备后，传送完固定k个字节的成组数据后，才释放总线，再去选择下一台设备，再传送k个字节，要传送n个字节，需要申请n/k次使用通道总线。

[Ts是设备选择时间，Td是传送一个字节所用的时间]

通道流量是指一个数据在传送期间，单位时间内能够传送的最大数据量，一般用字节个数表示。

传送过程说明：

通道在每个工作周期里，对设备发出的请求进行一次排队，选出优先级高的设备请求予以响应和处理，只有在此刻之前或与此刻同时发出的请求传送数据的设备，才能参加本次排队，错过这个时间，必须等到下一次设备选择期，开始参加排队。

---

流量设计的要求：要求实际流量接近极限流量，但可能会造成数据丢失，所以一般会增加通道最大流量，保证请求都可以被响应，或者动态改变优先级，增加数据缓冲寄存器



## 五、实验运行结果和相关源代码

### 1.实验结果

模拟通道处理机，将主存内的三个数据块computer architecture channelTest分别传送到三个外部设备。

在用户程序中使用访管指令进入管理程序，由CPU通过管理程序组织一个通道程序，并启动通道；

CPU [Java Application] /Library/Java/JavaVirtualMachines/jdk1.8

CPU中断，准备响应外部设备请求  
收到I/O指令，通道初始化...  
CPU忙...

CPU中断，响应通道请求，通道初始化后CPU继续完成之前操作。

此程序模拟的是字节多路通道，核心程序如下：

```
for (int i = 0; i < this.maxDevCap; i++)  
    for (int j = 0; j < this.Devices.size(); j++) {  
        if (i < this.memory.get(j).getDataLength()) {  
            this.Devices.get(j).setData(this.memory.get(j).getData().substring(0, i + 1));  
        }  
        System.out.println();  
        printDevice();  
    }
```

从下面程序运行结果可以看出，每次通道处理机从存储器取一个字节，输出到指定设备中，每次写入一个设备：



通道处理机连接设备数量: 3

c null null  
c a null  
c a c  
co a c  
co ar c  
co ar ch  
com ar ch  
com arc ch  
com arc cha  
comp arc cha  
comp arch cha  
comp arch chan  
compu arch chan  
compu archi chan  
compu archi chann  
comput archi chann  
comput archit chann  
comput archit channa  
compute archit channa  
compute archite channa  
compute archite channel  
computer archite channel  
computer architec channel  
computer architec channelT  
compute archite channel  
computer archite channel  
computer architec channel  
computer architec channelT  
computer architect channelT  
computer architect channelTe  
computer architect channelTe  
computer architectu channelTe  
computer architectu channelTes  
computer architectu channelTes  
computer architectur channelTes  
computer architectur channelTest  
computer architectur channelTest  
computer architecture channelTest