

本节内容

I/O控制方式

王道考研/CSKAQYAN.COM

知识总览

I/O控制方式

即：用什么样的方式来控制I/O设备的数据读/写

程序直接控制方式

中断驱动方式

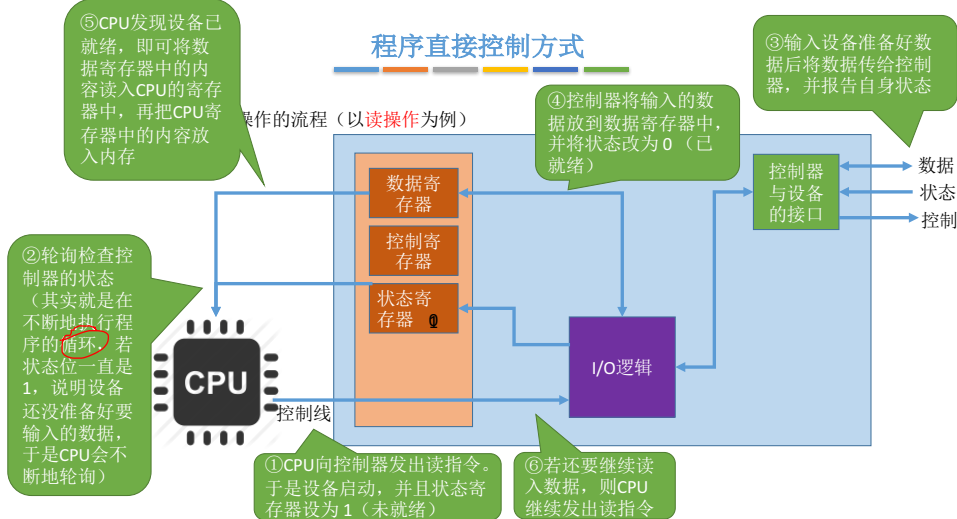
DMA方式

通道控制方式

- 需要注意的问题：
1. 完成一次读/写操作的流程；
 2. CPU干预的频率；
 3. 数据传送的单位；
 4. 数据的流向；
 5. 主要缺点和主要优点。

王道考研/CSKAQYAN.COM

程序直接控制方式



王道考研/CSKAQYAN.COM

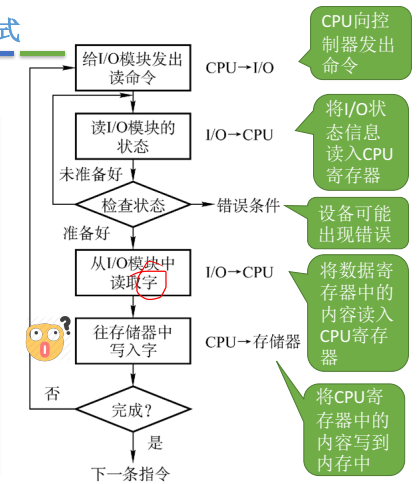
程序直接控制方式

1. 完成一次读/写操作的流程（见右图，Key word: 轮询）

```
01. #include <stdio.h>
02. #include <stdlib.h>
03. int main()
04. {
05.     int a, b, c, d;
06.     scanf("%d", &a); //输入整数并赋值给变量a
07.     scanf("%d", &b); //输入整数并赋值给变量b
08.     printf("a+b=%d\n", a+b); //计算a+b的值
09.     scanf("%d %d", &c, &d); //输入两个整数并分别赋值给c、d
10.     printf("c*d=%d\n", c*d); //计算c*d的值
11.
12.     system("pause");
13.     return 0;
14. }
```

输入的数据最终要放到内存中（a/b/c/d变量存放在内存中）

同理，输出的数据也存放在内存中，需要从内存取出



(a) 程序直接控制方式
王道考研/CSKAQYAN.COM

程序直接控制方式

1. 完成一次读/写操作的流程（见右图，Key word: 轮询）

2. CPU干预的频率
很频繁，I/O操作开始之前、完成之后需要CPU介入，并且在等待I/O完成的过程中CPU需要不断地轮询检查。

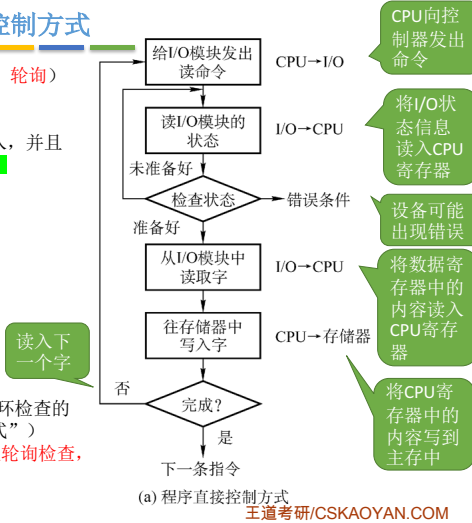
3. 数据传送的单位
每次读/写一个字节

4. 数据的流向
读操作（数据输入）：I/O设备→CPU→内存
写操作（数据输出）：内存→CPU→I/O设备
每个字的读/写都需要CPU的帮助

5. 主要缺点和主要优点

优点：实现简单。在读/写指令之后，加上实现循环检查的一系列指令即可（因此才称为“程序直接控制方式”）

缺点：CPU和I/O设备只能串行工作，CPU需要一直轮询检查，长期处于“忙等”状态，CPU利用率低。



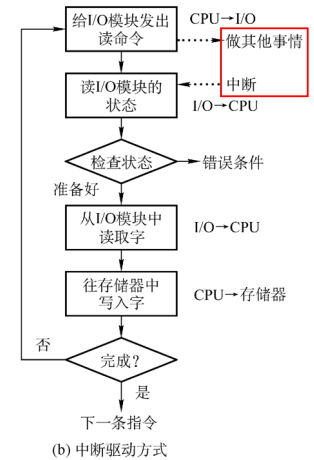
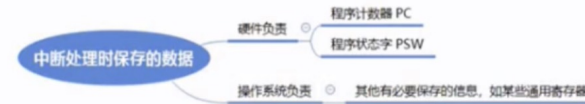
(a) 程序直接控制方式
王道考研/CSKAQYAN.COM

中断驱动方式

引入中断机制。由于I/O设备速度很慢，因此在CPU发出读/写命令后，可将等待I/O的进程阻塞，先切换到别的进程执行。当I/O完成后，控制器会向CPU发出一个中断信号，CPU检测到中断信号后，会保存当前进程的运行环境信息，转去执行中断处理程序处理该中断。处理中断的过程中，CPU从I/O控制器读一个字的数据传送到CPU寄存器，再写入主存。接着，CPU恢复等待I/O的进程（或其他进程）的运行环境，然后继续执行。

注意：①CPU会在每个指令周期的末尾检查中断；
②中断处理过程中需要保存、恢复进程的运行环境，这个过程是需要一定时间开销的。可见，如果中断发生的频率太高，也会降低系统性能。

原则：保存中断处理程序结束返回之后（在原程序中）还需使用且不可恢复的内容



(b) 中断驱动方式
王道考研/CSKAQYAN.COM

中断驱动方式

1. 完成一次读/写操作的流程（见右图，Key word: 中断）

2. CPU干预的频率
每次I/O操作开始之前、完成之后需要CPU介入。等待I/O完成的过程中CPU可以切换到别的进程执行。

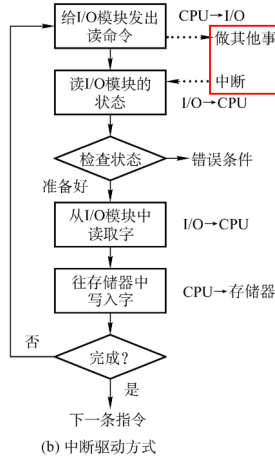
3. 数据传送的单位
每次读/写一个字节

4. 数据的流向
读操作（数据输入）：I/O设备→CPU→内存
写操作（数据输出）：内存→CPU→I/O设备

5. 主要缺点和主要优点

优点：与“程序直接控制方式”相比，在“中断驱动方式”中，I/O控制器会通过中断信号主动报告I/O已完成，CPU不再需要不停地轮询。CPU和I/O设备可并行工作，CPU利用率得到明显提升。

缺点：每个字在I/O设备与内存之间的传输，都需要经过CPU。而频繁的中断处理会消耗较多的CPU时间。



(b) 中断驱动方式
王道考研/CSKAQYAN.COM

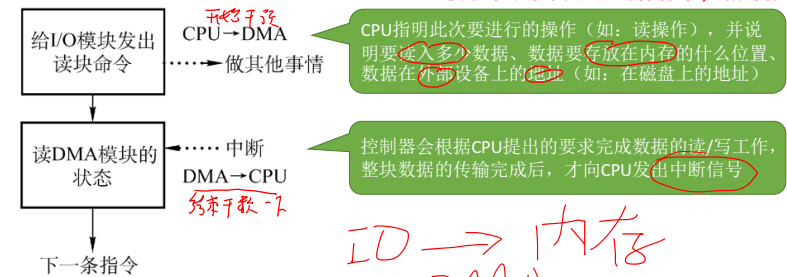
传一个字中断一次

DMA方式

与“中断驱动方式”相比，DMA方式（Direct Memory Access，直接存储器存取。主要用于块设备的I/O控制）有这样几个改进：

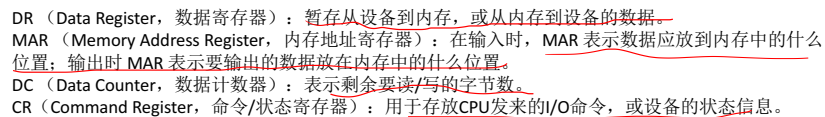
①数据的传送单位是“块”。不再是一个字、一个字的传送；
②数据的流向是从设备直接放入内存，或者从内存直接到设备。不再需要CPU作为“快递小哥”。
③仅在传送一个或多个数据块的开始和结束时，才需要CPU干预。

Cou写好本次要读的数据清单，然后交给dma去做



IO → 内存
DMA

王道考研/CSKAQYAN.COM



The diagram illustrates the Channel Program (通道程序) architecture and its execution flow.

Architecture:

- CPU:** Interacts with the Channel Program via I/O instructions (① I/O指令) and interrupts (③ 中断).
- 内存 (Memory):** Contains the Channel Program (任务清单) and Data (数据).
- 通道 (硬件) (Channel Hardware):** Contains the Channel Program (通道程序) and its starting address (首地址). It interacts with I/O devices (I/O设备1, I/O设备2, I/O设备3).

Execution Flow:

- ① I/O指令:** CPU sends I/O instructions to the Channel Program.
- ② 通道执行内存中的通道程序:** The Channel Program executes in memory, specifying data to be read/written and its location.
- ③ 中断:** After completing the task, the Channel Program sends an interrupt signal to the CPU, which then processes the interrupt.

Channel Program Execution Details:

- 请求I/O访问指令:** Request for I/O access instruction.
- 响应I/O中断请求:** Response to I/O interrupt request.
- 编制通道程序:** Channel program compilation.
- 启动I/O通道:** Start I/O channel.
- 通道控制工作:** Channel control work.
- 组织I/O操作:** Organize I/O operation.
- 向CPU发中断请求:** Send interrupt request to CPU.
- 登记或处理:** Register or process.

Channel Control: A single channel controls multiple devices, and only one device can work at any given time.

王道考研/CSKAOYAN.COM

知识点回顾与重要考点

	完成一次读/写的过程	CPU干 预频率	每次I/O的数 据传输单位	数据流向	优缺点
程序直接控制方式	CPU发出I/O命令后需要不断轮询	极高	字	设备→CPU→内存 内存→CPU→设备 <i>CPU忙于来回传送</i>	每一个阶段的优点都是解决了上一阶段的 最大缺点。
中断驱动方式	CPU发出I/O命令后可以 做其他事，本次I/O完成后 设备控制器发出中断信号	高	字	设备→CPU→内存 内存→CPU→设备	总体来说，整个 发展过程就是 要尽量减少
DMA方式	CPU发出I/O命令后可以 做其他事，本次I/O完成后 DMA控制器发出中断信号	中	块	设备→内存 内存→设备	CPU对I/O过程 的干预，把CPU 从繁杂的I/O控 制事务中解脱
通道控制方式	CPU发出I/O命令后可以 做其他事。通道会执行通道 程序以完成I/O，完成后通 道向CPU发出中断信号	低	一组块	设备→内存 内存→设备	出来，以便更 多地去完成数 据处理任务。

难点理解：
通道=弱鸡版CPU
通道程序=任务清单

从哪读，读多少，放到哪里