

操作系统

Operating Systems

L8. CPU管理的直 观想法

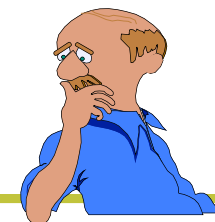
CPU?

授课教师：李治军

lizhijun_os@hit.edu.cn

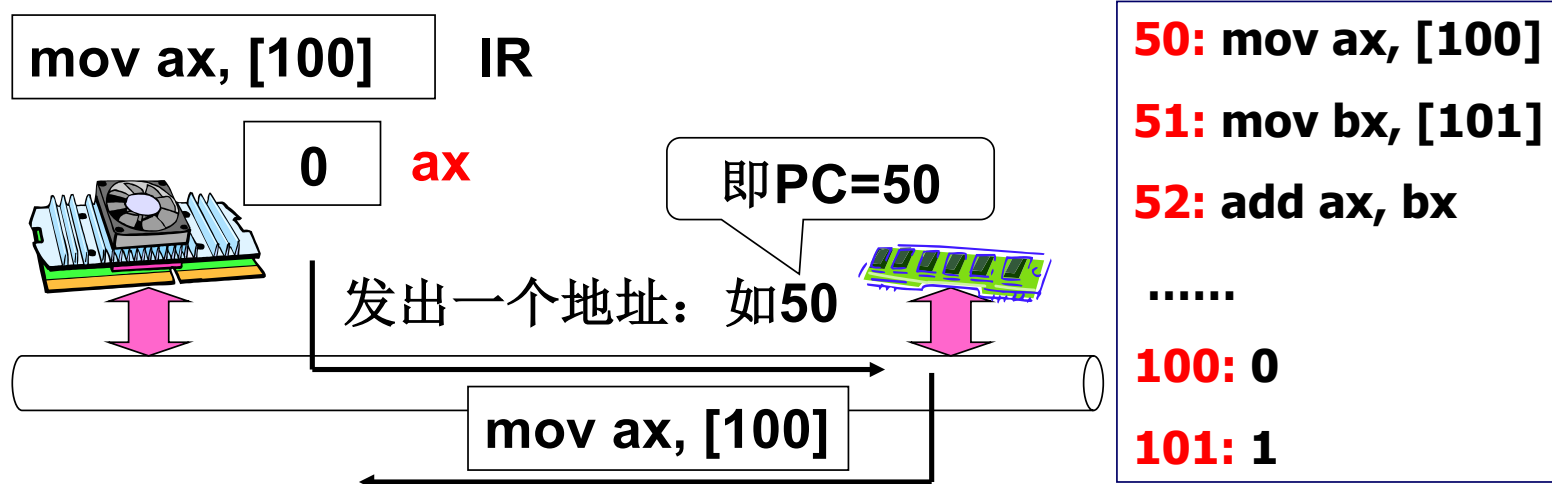
综合楼416室

管理CPU，先要使用CPU...

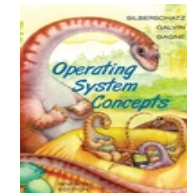


- CPU的工作原理
- CPU上电以后发生了什么？

自动的取指—执行

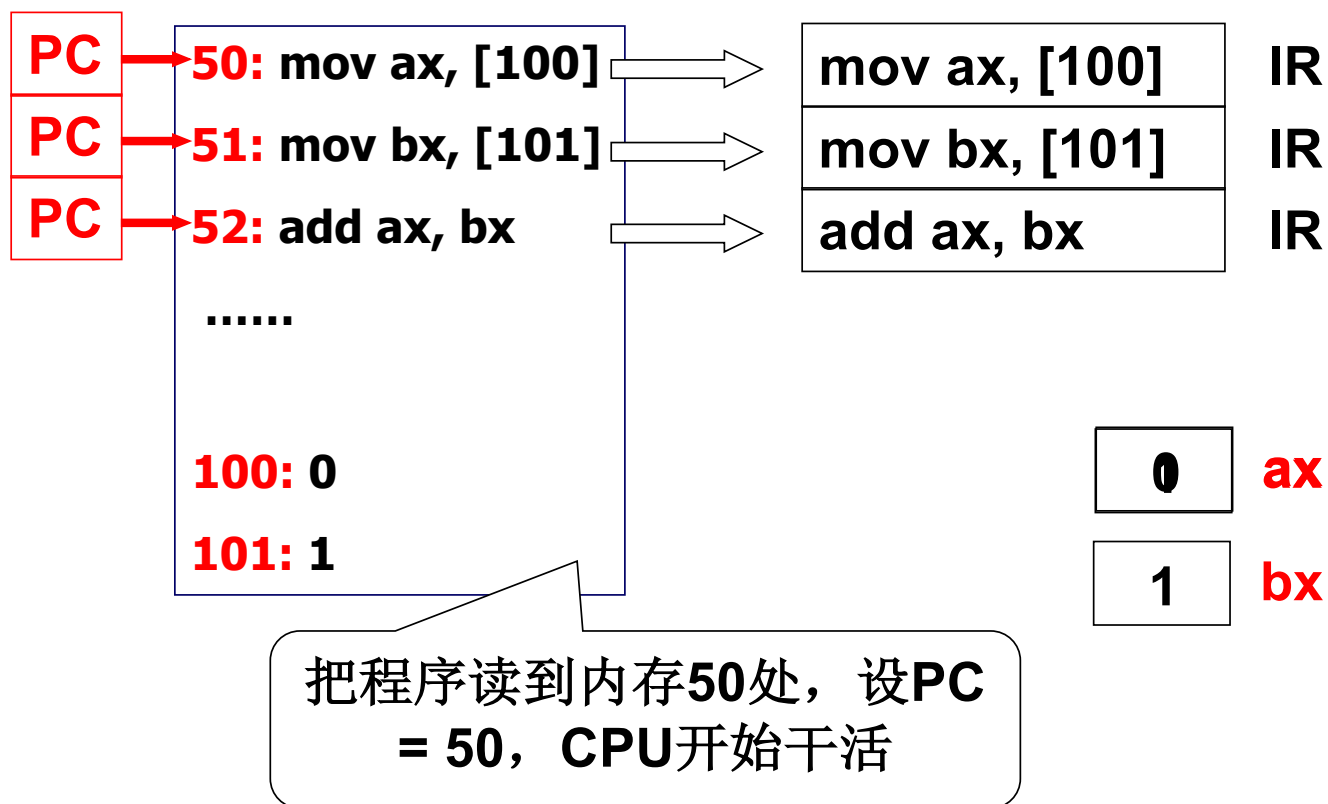


- CPU怎么工作？
- CPU怎么管理？



管理CPU的最直观方法

■ 设好PC初值就完事!



看看这样做有没有问题？ 提出问题

```
int main(int argc, char* argv[])
{
    int i, to, *fp, sum = 0;
    to = atoi(argv[1]);
    for(i=1; i<=to; i++)
    {
        sum = sum + i;
        fprintf(fp, "%d", sum);
    }
}
```

fprintf用一条其他计算语句代替

```
C:\>sum 10000000
0.015000 seconds
```

0.015/10⁷

有fprintf

```
C:\>sum 1000
0.859000 seconds
```

0.859/10³

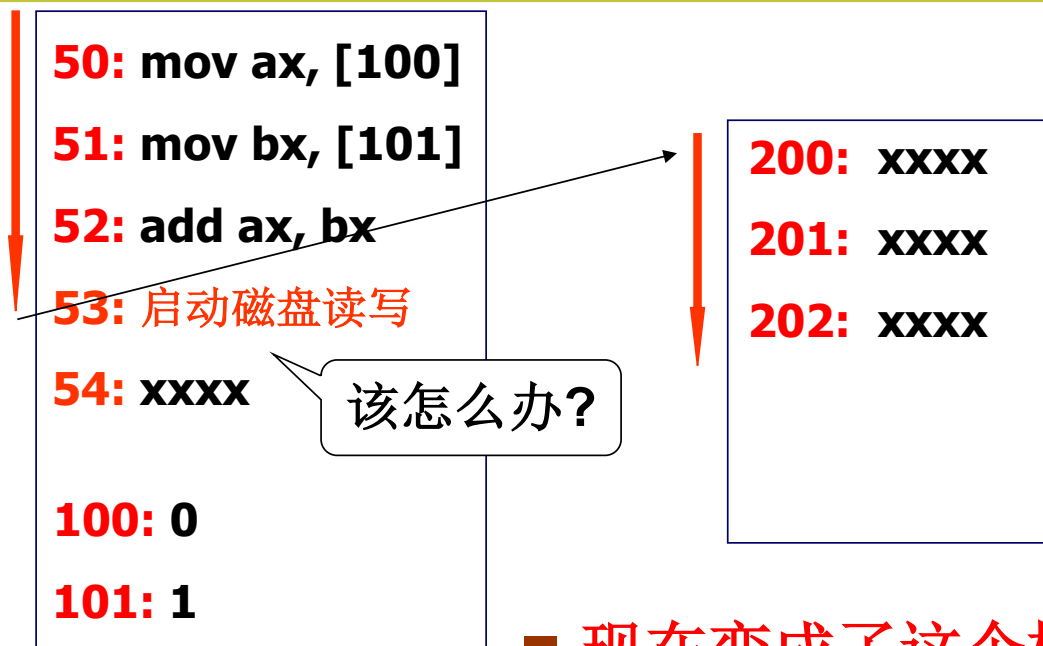
5.7×10⁵ : 1

IO指令比计算指令耗时非常多
IO特别慢，后面的指令可能依赖于IO的结果
所以必须让IO完成，当前任务才能够继续
在执行IO的时候可以让CPU执行其他的任务

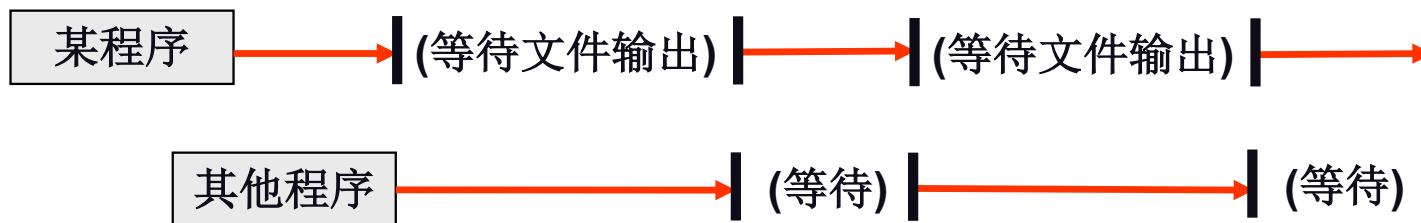




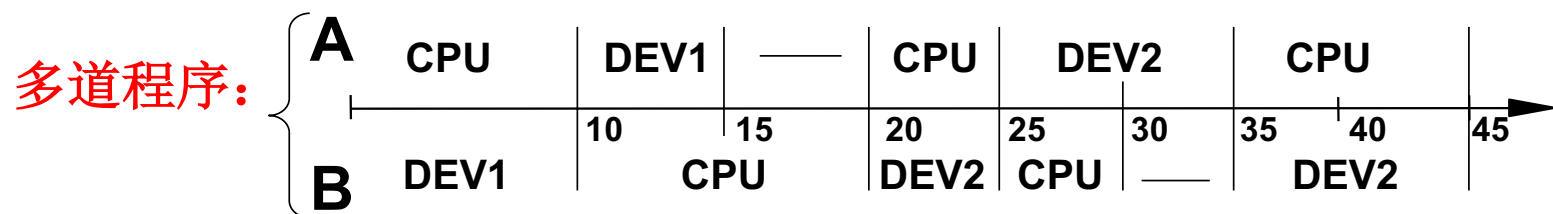
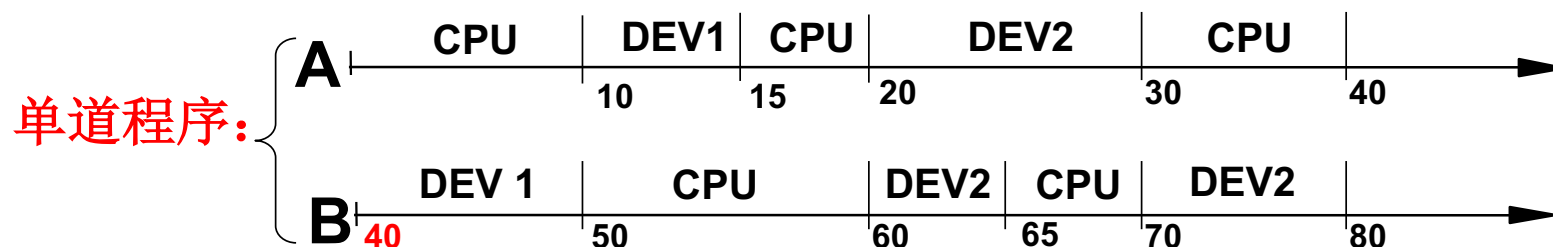
怎么解决?



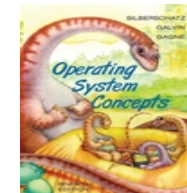
■ 现在变成了这个样子



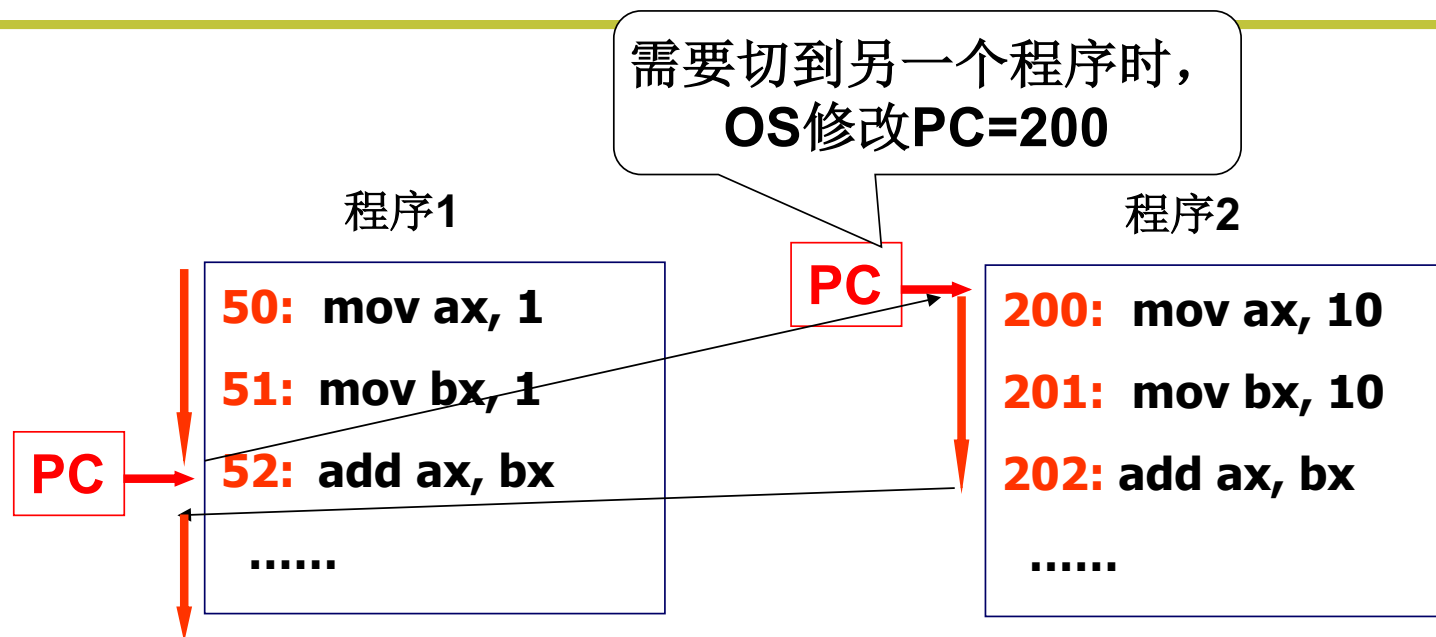
多道程序、交替执行，好东西啊！



	单道程序	多道程序
CPU利用率	$40/80=50\%$	$40/45=89\%$
DEV1利用率	$15/80=18.75\%$	$15/45=33\%$
DEV2利用率	$25/80=31.25\%$	$25/45=56\%$



一个CPU面对多个程序？



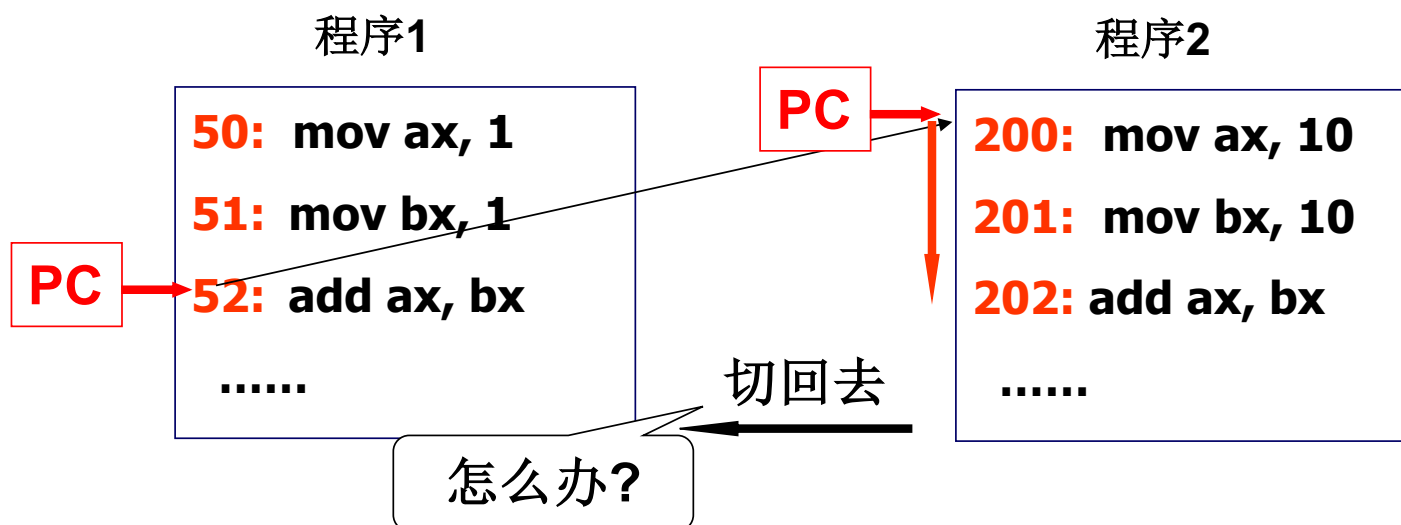
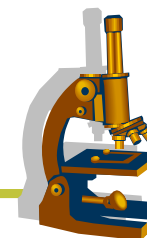
■ 一个CPU上交替的执行多个程序：并发

将多个程序放入到内存中

■ 怎么做到的？



修改寄存器PC就行了吗？



- 运行的程序和静态程序不一样了...

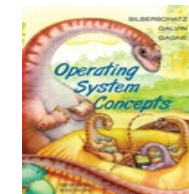
每个运行的程序
有很多需要记录的状态、便于切换之后能够恢复现场。如PC指针，各个寄存器等。
而静态程序就是在磁盘上的代码

程序1信息

2	ax
1	bx
53	PC

- 要记录返回地址，要记录ax...

每个程序有了一个存放信息的结构: **PCB**





引入“进程”概念

■ 运行的程序和静态程序不一样!

- 需要描述这些不一样...

- 程序 + 所有这些不一样 → 一个概念

这些不一样就成了
进程概念的外延

所有的不一样都表
现在**PCB**中...

■ 进程是进行（执行）中的程序

- 进程有开始、有结束，程序没有

- 进程会走走停停，走停对程序无意义

- 进程需要记录ax,bx,...，程序不用

-

