

# 【OS】 Day3

▼ Class	现代操作系统
📅 Date	@December 7, 2021

## 【Ch1】 引论

### 1.3.1 处理器

除了用来保存变量和临时结果的通用寄存器之外，多数计算机还有一些对程序员可见的专用寄存器。其中之一是**程序计数器**，它保存了**将要取出的下一条指令的内存地址**。在指令取出之后，程序计数器就被更新以便指向后继的指令

另一个寄存器是**堆栈指针**，它**指向了内存中当前栈的顶端**。该栈包含了每个执行过程的栈帧。一个过程的栈帧中保存了有关的**输入参数、局部变量以及那些没有保存在寄存器中的临时变量**

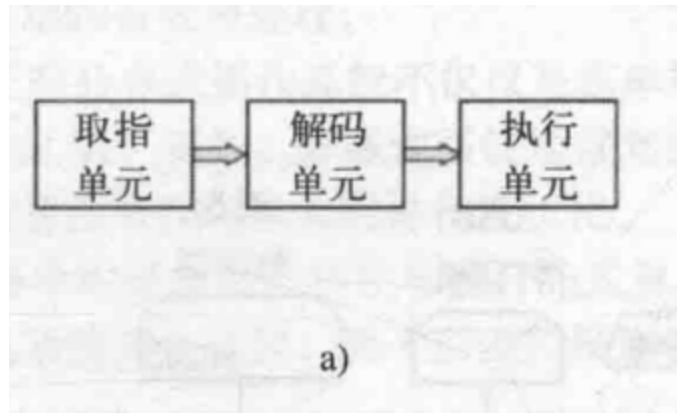
还有**程序状态字 (Program Status Word, PSW) 寄存器**。这个寄存器包含了**条件码位**（由比较指令设置）、**CPU优先级、模式（用户态或内核态）**，以及各种其他控制位。

操作系统必须知晓所有的寄存器。在**时间多路复用(time multiplexing)CPU**中，操作系统经常会**终止正在运行的某个程序并启动（或再启动）另一个程序**。每次停止一个运行着的程序时，操作系统必须**保存所有的寄存器值**，这样在稍后该程序被再次运行时，可以把这些寄存器重新装入

### 流水线

为了改善性能，CPU设计师放弃了同时读取、解码和执行一条指令的简单模型。许多现代CPU具有**同时取出多条指令的机制**。

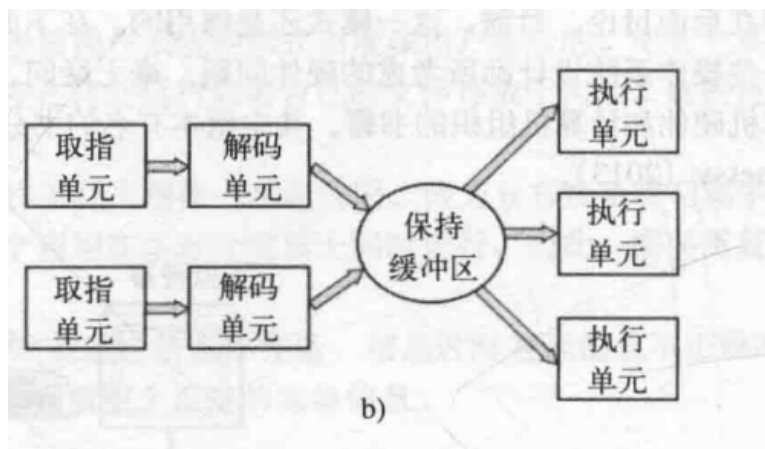
例如，一个CPU可以有单独的取指单元、解码单元和执行单元，于是当它执行指令 $n$ 时，还可以对指令 $n+1$ 解码，并且读取指令 $n+2$ 。



这样的机制称为**流水线(pipeline)**。

### 超标量CPU

比流水线更先进的设计是**超标量CPU**。在这种设计中，**有多个执行单元**，例如，一个CPU用于整数计算，一个CPU用于浮点算术运算，一个CPU用于布尔运算。



两个或更多的指令被同时取出、解码并**装入暂存缓冲区**中，直至它们执行完毕。

只要有一个**执行单元空闲**，就检查保持缓冲区中是否还有可处理的指令，如果有，就把**指令从缓冲区中移出并执行**。

这种设计有一种隐含的作用，**即程序的指令经常不按顺序执行**。在大多数情况下，硬件负责保证这种运算的结果与顺序执行指令时的结果相同。但是，仍然有部分工作交给操作系统处理。

## 内核态和用户态

多数CPU都有两种模式，即前面已经提到的内核态和用户态。

通常，PSW中有一个二进制位控制这两种模式。当在内核态运行时，CPU可以执行指令集中的每一条指令。

在台式机和服务器上，操作系统一般在内核态下运行

在大多数嵌入式系统中，一部分操作系统运行在内核态，其余的部分运行在用户态

相反，用户程序在用户态下执行。仅允许执行整个指令集的一个子集和访问所有功能的一个子集。

一般而言，在用户台中有关I/O和内存保护的所有指令是禁止的。当然，将PSW中的模式位设置成内核态也是禁止的

为了从操作系统中获得服务，用户程序必须使用系统调用(system call)以陷入内核并调用操作系统。TRAP指令把用户态切换成内核态，并启用操作系统。

当有关工作完成之后，在系统调用后面的指令把控制权返回给用户程序

## 多线程和多核芯片

Intel Pentium 4引入了被称为多线程(multithreading)的特性，x86处理器和其他一些CPU芯片就是这样做的。

多线程允许CPU保持两个不同的线程状态，然后在纳秒级的时间尺度来回切换（线程是一种轻量级的进程，即一个运行中的程序）

例如，如果某个进程需要从内存中读出一个字，多线程CPU则可以切换至另一个线程

### 1.3.4 I/O设备

I/O设备一般包括两个部分：设备控制器和设备本身。

控制器是插在电路板上的一块芯片或一组芯片，这块电路板物理地控制这些设备。它从操作系统接收命令

在许多情况下，对这些设备的控制是非常复杂和具体的，所以，控制器的任务是为操作系统提供一个简单的接口。