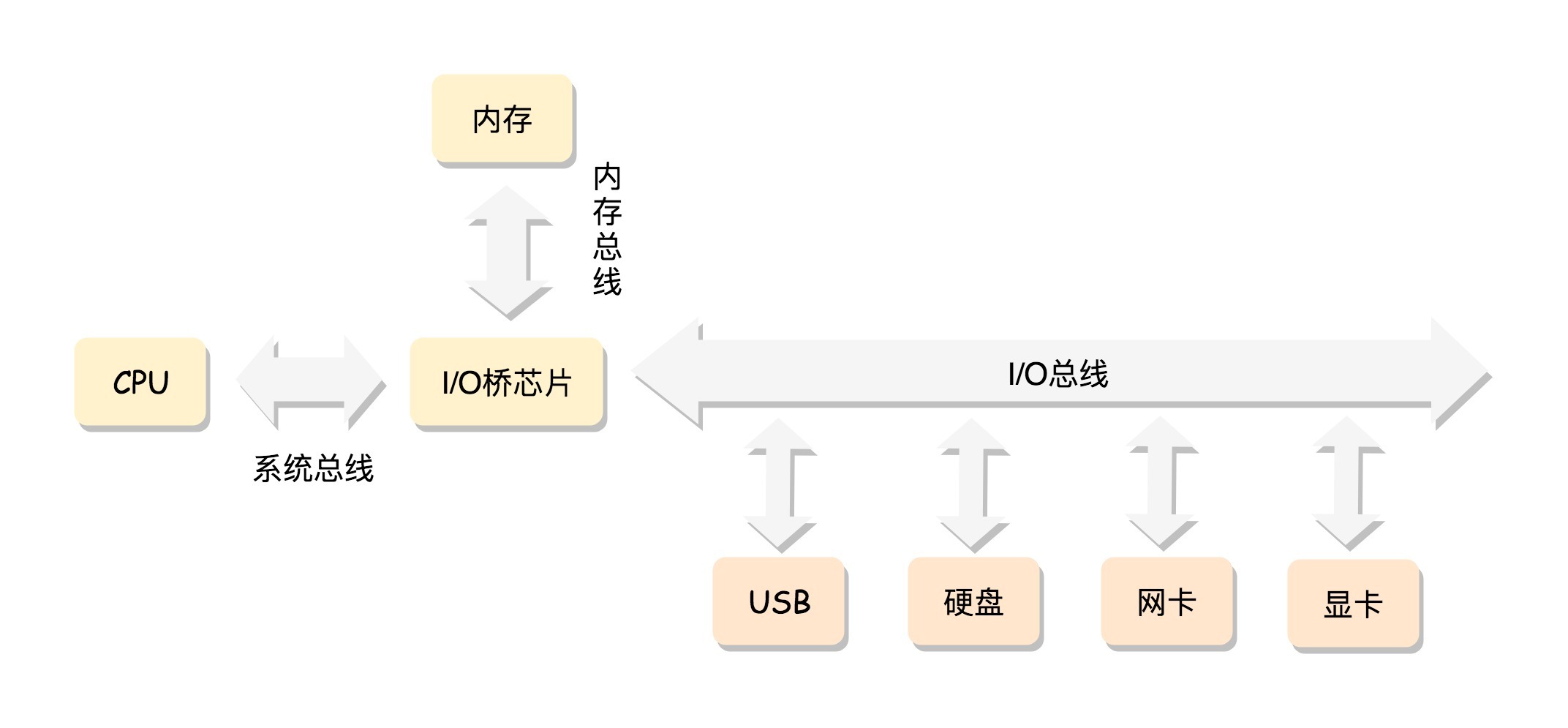
## 计算机组成概述

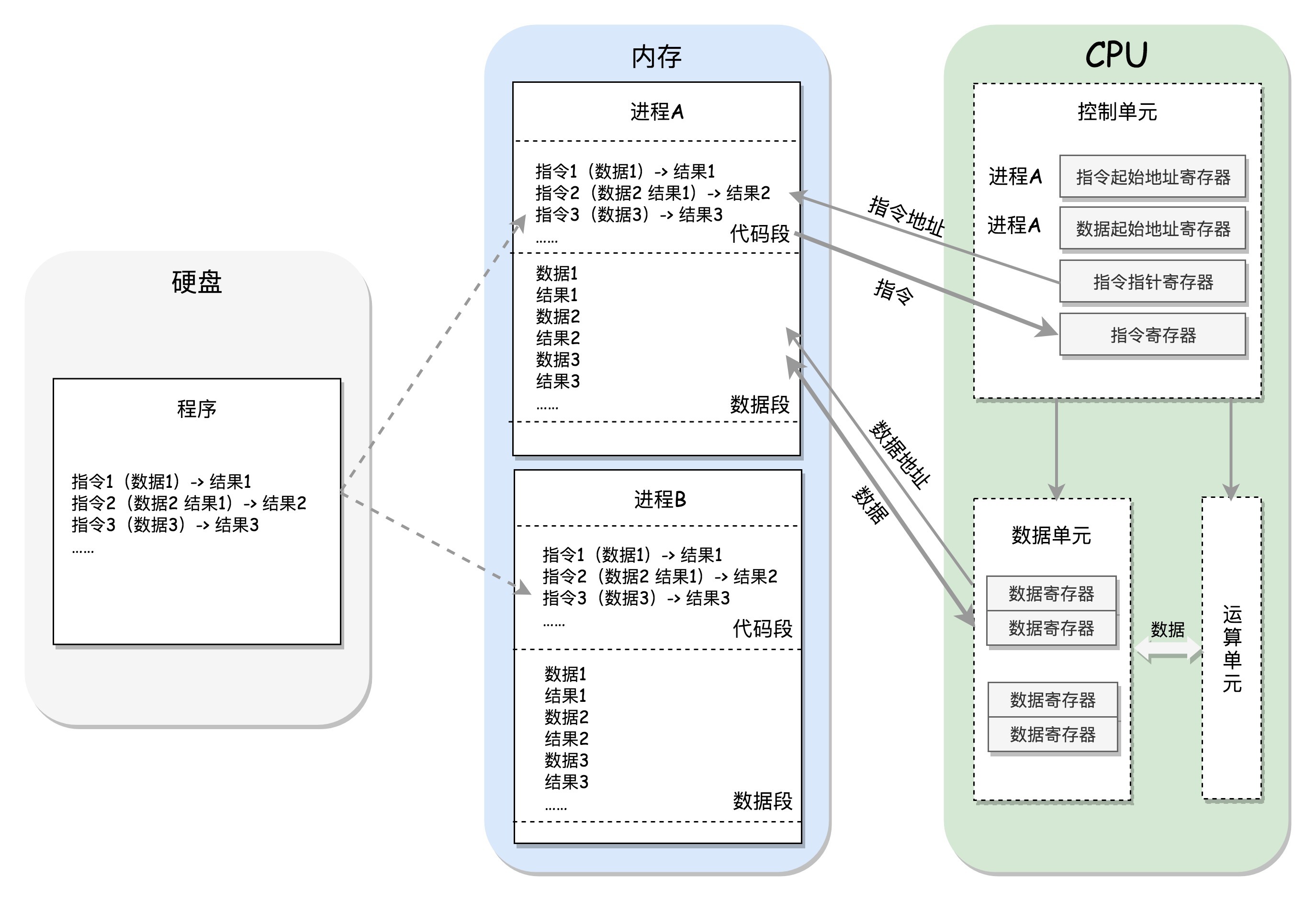


计算机来讲，最核心的就是 CPU（Central Processing Unit，中央处理器）；

CPU 和其他设备连接，要靠一种叫做总线（Bus）的东西，其实就是主板上密密麻麻的集成电路，这些东西组成了 CPU 和其他设备的高速通道；

在这些设备中，最重要的是内存（Memory）。因为单靠 CPU 是没办法完成计算任务的，很多复杂的计算任务都需要将中间结果保存下来，然后基于中间结果进行进一步的计算。CPU 本身没办法保存这么多中间结果，这就要依赖内存了。

CPU 其实也不是单纯的一块，它包括三个部分，**运算单元、数据单元和控制单元**。



（1）运算单元只管算，例如做加法、做位移等等。但是，它不知道应该算哪些数据，运算结果应该放在哪里。

1. 运算单元计算的数据如果每次都要经过总线，到内存里面现拿，这样就太慢了，所以就有了数据单元。数据单元包括 CPU 内部的缓存和寄存器组，空间很小，但是速度飞快，可以暂时存放数据和运算结果。
2. 控制单元是一个统一的指挥中心，它可以获得下一条指令，然后执行这条指令。这个指令会指导运算单元取出数据单元中的某几个数据，计算出个结果，然后放在数据单元的某个地方。

进程一旦运行，比如图中两个进程 A 和 B，**会有独立的内存空间，互相隔离，程序会分别加载到进程 A 和进程 B 的内存空间里面，形成各自的代码段**。

CPU的数据单元根据数据的地址，从数据段里读到数据寄存器里，就可以参与运算了。运算单元做完运算，产生的结果会暂存在数据单元的数据寄存器里。最终，会有指令将数据写回内存中的数据段。

当前执行的是进程 A 的指令，等切换成进程 B，就会执行 B 的指令了，这个过程叫作进程切换（Process Switch）。

到这里，你会发现，CPU 和内存来来回回传数据，靠的都是总线。其实总线上主要有两类数据，一个是地址数据，也就是我想拿内存中哪个位置的数据，这类总线叫**地址总线**（Address Bus）；另一类是真正的数据，这类总线叫**数据总线**（Data Bus）。

地址总线的位数，决定了能访问的地址范围到底有多广。例如只有两位，那 CPU 就只能认 00，01，10，11 四个位置，超过四个位置，就区分不出来了。位数越多，能够访问的位置就越多，能管理的内存的范围也就越广。

而数据总线的位数，决定了一次能拿多少个数据进来。例如只有两位，那 CPU 一次只能从内存拿两位数。要想拿八位，就要拿四次。位数越多，一次拿的数据就越多，访问速度也就越快。

比如，在 32 位处理器中，有 32 根地址总线，可以访问 2^32=4G 的内存

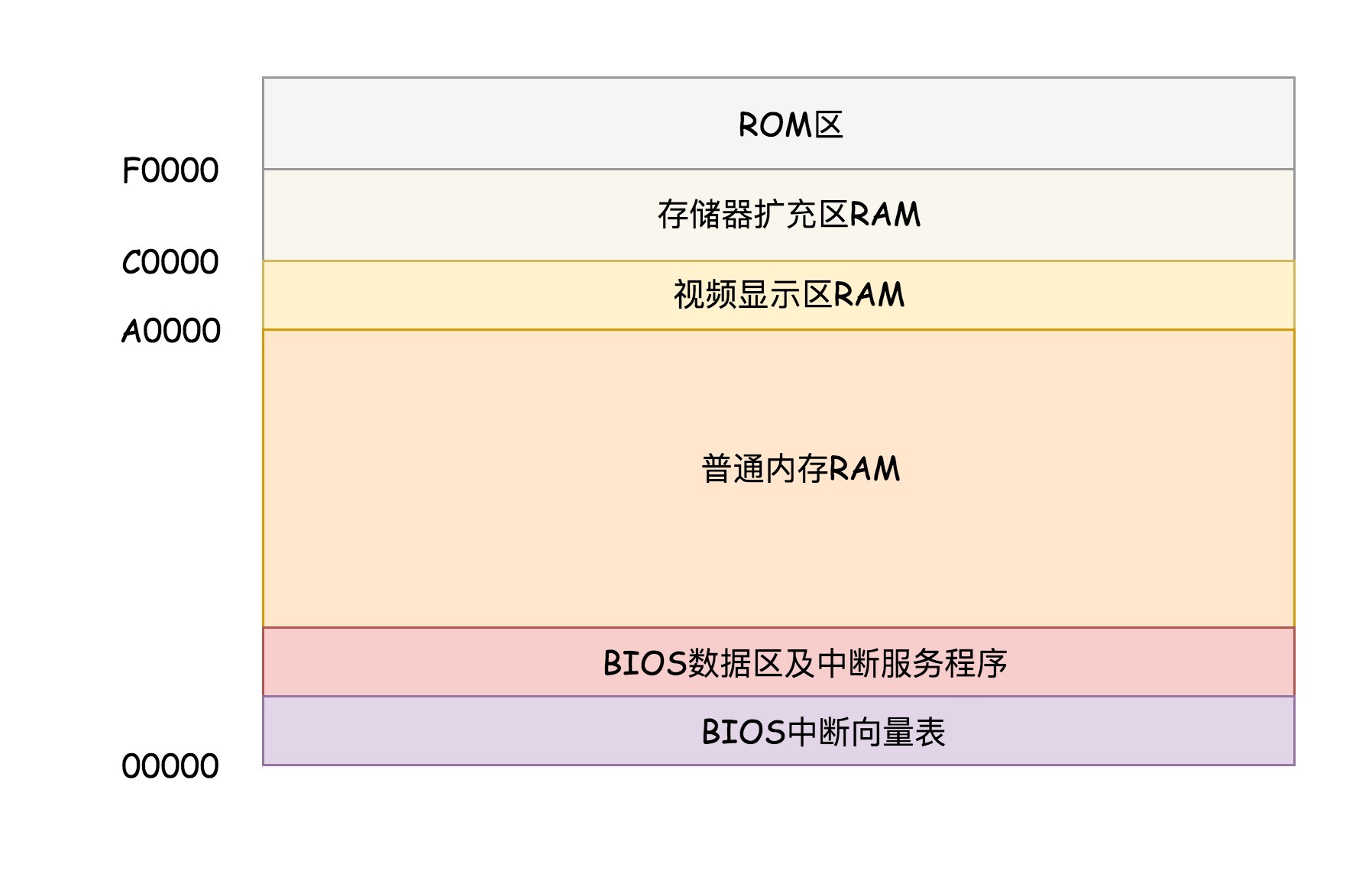
### 计算机启动

当你轻轻按下计算机的启动按钮时，你的主板就加上电了。这时候你的 CPU 应该开始执行指令了。

在主板上，有一个东西叫 ROM（Read Only Memory，只读存储器）。这和常说的内存 RAM（Random Access Memory，随机存取存储器）不同。

咱们平时买的内存条是可读可写的，这样才能保存计算结果。而 ROM 是只读的，上面早就**固化了一些初始化的程序，也就是 BIOS**（Basic Input and Output System，基本输入输出系统）。

刚启动的时候，按某个组合键，显示器会弹出一个蓝色的界面。能够调整启动顺序的系统，就是我说的 BIOS。



当电脑刚加电的时候，会做一些重置的工作，将 CS 设置为 0xFFFF，将 IP 设置为 0x0000，所以第一条指令就会指向 0xFFFF0，正是在 ROM 的范围内。在这里，有一个 JMP 命令会跳到 ROM 中做初始化工作的代码，于是，BIOS 开始进行初始化的工作。

BIOS 要检查一下系统的硬件是不是都好着的。

这个时候，要建立一个中断向量表和中断服务程序，因为现在你还要用键盘和鼠标，这些都要通过中断进行的。还会在内存空间映射显存的空间，在显示器上显示一些字符等等。

操作系统在哪儿呢？一般都会在安装在硬盘上，在 BIOS 的界面上。你会看到一个启动盘的选项。启动盘有什么特点呢？它一般在第一个扇区，占 512 字节，而且以 0xAA55 结束。这是一个约定，当满足这个条件的时候，就说明这是一个启动盘，在 512 字节以内会启动相关的代码。

这些代码是谁放在这里的呢？在 Linux 里面有一个工具，叫 Grub2，全称 Grand Unified Bootloader Version 2。顾名思义，就是搞系统启动的。

结果就是我们下面常见的图：

