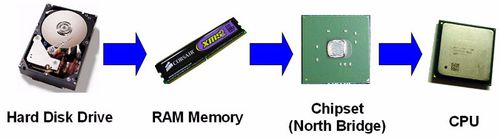
当双击系统桌面程序图标运行程序的时候，程序大多数存储在电脑的硬盘中，然后要加载入内存，最后处理器要在一个称为“内存控制器”的电路控制下读取相应的指令。

在因特尔处理器中，内存控制器装载在北桥芯片组上，AMD则直接装载入处理器中。下图是对上述过程的一个概括。

[](https://gss0.baidu.com/-Po3dSag_xI4khGko9WTAnF6hhy/zhidao/pic/item/ac4bd11373f08202a1b24c0f47fbfbedab641bda.jpg)

【题外话：

如果断电的话，内存中的指令就会丢失。因而内存归类为“易失性”介质。所以要把程序、数据存储在不易失性的介质中，比如硬盘和光盘。

内存有易失性，处理器为什么不直接从外存中取指令呢？

因为外存是机械结构，内存是电路结构的，数据的传输速率上来看内存要比外存快许多。从更好的发挥处理器的性能上看，要从内存中读取数据，越接近处理器传输速率，处理器的利用率就越高。】

内存存储区的大致划分

|  |  |  |
| --- | --- | --- |
| **堆**：由程序员分配释放 | **动态存储区** | <——*指针P*  < 哪个区  < 都可以  < 指向；  但常量  < 区、代码区内容不可修改 |
| **栈**：由编译器自动分配释放 |
| **全局区、静态区** | **静态存储区** |
| **常量区**：常量字符串就是放在这里的程序结束后由系统释放 |
| **程序代码区**：存放函数体的二进制代码。 | **程序区** |

void main()

{

char str[50],\*pt=str; //pt指向str，在动态区

\*pt=’a’; //是在动态存储区进行写操作

pt="abcde"; //pt指向发生变化，指向常量区

\*pt=’b’; //出错，不允许在常量区进行写操作 }

char str[]="123",\*pt="abcde";

str在动态区，这个赋值操作实现方式与strcpy(str, "123")相同，是将字符一个个拷入str对应的动态空间里；

而"abcde"是常量区里的内容，pt被赋值，指向了这个空间，\*pt访问的是常量区的内容，则不允许对之写操作。