第一讲 简述 x86 寻址演变

我们知道，操作系统是一组软件的集合。但它和一般软件不同，因为它是充分挖掘硬件潜能的软件，也可以说，操作系统是横跨软件和硬件的桥梁。因此，要想深入解析操作系统内在的运作机制，就必须搞清楚相关的硬件机制——尤其是内存寻址的硬件机制。

操作系统的设计者必须在硬件相关的代码与硬件无关的代码之间划出清楚的界限，以便于一个操作系统很容易地移植到不同的平台。Linux 的设计就做到了这点，它把与硬件相关的代码全部放在 arch(architecture 一词的缩写，即体系结构相关)的目录下，在这个目录下，可以找到 Linux 目前版本支持的所有平台，例如，支持的平台有 arm、alpha,、i386、m68k、mips 等十多种。在这众多的平台中，大家最熟悉的就是 i386，即 Intel80386 体系结构。因此，我们所介绍的内存寻址也是以此为背景。

1、 鼻祖图灵

曾经有一个叫“阿兰.图灵”的天才（据说他 16 岁开始研究相对论，虽然英年早逝，但才气纵横逻辑学、物理学、数学等多个领域，尤其是数学逻辑上的所作所为奠定了现代计算技术的理论基础 。后来以他名字命名的“图灵奖”被看作计算机学界的最高荣誉）。他设想出了一种简单但运算能力几乎无限发达的理想机器，这不是一个具体的机械设备，而是一个思想模型，可以用来计算能想象得到的所有可计算函数。这个有趣的机器由一个控制器，一个读写头和一条假设两端无限长的带子组成。工作带相当于存储器，被划分成大小相同的格子，每格上可写一个字母，读写头可以在工作带上随意移动，而控制器可以要求读写头读取其下方工作带上的字母。

这听起来仅仅是纸上谈兵，但它却是当代冯.诺依曼计算机体系的理论鼻祖。它带来的“数据连续存储和选择读取思想”是目前我们使用的几乎所有机器运行背后的灵魂。计算机体系结构中的核心问题之一就是如何有效地进行内存寻址，因为所有运算的前提都是先要从内存中取得数据，所以内存寻址技术从某种程度上代表了计算机技术

2、 4 位和 8 位处理器及寻址

在微处理器的历史上，第一款微处理器芯片4004是由Intel推出的，那是一个4位的微处理器。在4004之后，Intel推出了一款8位处理器8080，它有1个主累加器（寄存器A）和6个次累加器（寄存器B,C,D,E,H和L），几个次累加器可以配对（如组成BC，DE或HL）用来访问16位的内存地址，也就是说8080可访问到64K内的地址空间。

3、 实模式－16位处理器及寻址

几年后，Intel开发出了16位的处理器8086，这个处理器标志着Intel x86王朝的开始，这也是内存寻址的第一次飞跃。之所以说这是一次飞跃，是因为8086处理器引入了一个重要概念—**段**。

8086处理器的寻址目标是1M大的内存空间，于是它的地址总线扩展到了20位。但是，一个问题摆在了Intel设计人员面前，虽然地址总线宽度是20位的，但是CPU中“算术逻辑运算单元（ALU）”的宽度，即数据总线却只有16位，也就是可直接加以运算的指针长度是16位的。如何填补这个空隙呢？可能的解决方案有多种，例如，可以像一些8位CPU中那样，增设一些20位的指令专用于地址运算和操作，但是那样又会造成CPU内存结构的不均匀。又例如，当时的PDP－11小型机也是16位的，但是其内存管理单元（MMU）可以将16位的地址映射到24位的地址空间。受此启发，Intel设计了一种在当时看来不失为巧妙的方法，即分段的方法。 在微处理器的历史上，第一款微处理器芯片4004是由Intel推出的，那是一个4位的微处理器。在4004之后，Intel推出了一款8位处理器8080，它有1个主累加器（寄存器A）和6个次累加器（寄存器B,C,D,E,H和L）,几个次累加器可以配对（如组成BC，De或HL）用来访问16位的内存地址，也就是说8080可访问到64K内的地址空间。另外，那时还没有段的概念，访问内存都要通过绝对地址，因此程序中的地址必须进行硬编码（给出具体地址），而且也难以重定位，这就不难理

解为什么当时的软件大都是些可控性弱，结构简陋，数据处理量小的工控程序了。

Intel 开发出了 16 位的处理器 8086，这个处理器标志着 Intel x86 王朝的开始。

为了支持分段，Intel 在 8086 CPU 中设置了四个段寄存器：CS、DS、SS 和 ES，分别用于可执行代码段、数据段、堆栈段及其他段。每个段寄存器都是 16 位的，对应于地址总线中的**高 16 位**。每条“访内”指令中的内部地址也都是 16 位的，但是在送上地址总线之前，CPU 内部自动地把它与某个段寄存器中的内容相加。因为段寄存器中的内容对应于 20 位地址总线中的高 16 位(也就是把段寄存器左移 4 位)，所以相加时实际上是内存总线中的高 12 位与段寄存器中的 16 位相加，而低 4 位保留不变，这样就形成一个 20 位的实际地址，也就实现了从 16 位内存地址到 20 位实际地址的转换，或者叫“**映射**”。

段式内存管理带来了显而易见的优势，程序的地址不再需要硬编码了，调试错误也更容易定位了，更可贵的是支持更大的内存地址。程序员开始获得了自由。

4、 保护模式－24 位及 32 位寻址

技术的发展不会就此止步。Intel 的 80286 处理器于 1982 年问世了，它的地址总线位数增加到了 24 位，因此可以访问到 16M 的内存空间。更重要的是从此开始引进了一个全新理念—**保护模式**。这种模式下内存段的访问受到了限制。访问内存时不能直接从段寄存器中获得段的起始地址了，而需要经过额外转换和检查（从此你不能再随意存取数据段，具体保护和实现我们后面讲述）。

为了和过去兼容，80286 内存寻址可以有两种方式，一种是先进的保护模式，另一种是老式的 8086 方式，被成为**实模式**。系统启动时处理器处于实模式，只能访问 1M 空间，

经过处理可进入保护模式，访问空间扩大到16M，但是要想从保护模式返回到实模式，你只有重新启动机器。还有一个致命的缺陷是80286虽然扩大了访问空间，但是每个段的大小还是64k，程序规模仍受到限制。因此这个先天低能儿注定命不会很久。很快它就被天资卓越的兄弟——80386代替了。

80386是一个32位的CPU，也就是它的ALU数据总线是32位的，同时它的地址总线与数据总线宽度一致，也是32位，因此，其寻址能力达到4GB。对于内存来说，似乎是足够了。从理论上说，当数据总线与地址总线宽度一致时，其CPU结构应该简洁明了。但是，80386无法做到这一点。作为x86产品系列的一员，80386必须维持那些段寄存器的存在，还必须支持实模式，同时又要能支持保护模式，这给Intel的设计人员带来很大的挑战。

Intel选择了在段寄存器的基础上构筑保护模式，并且保留段寄存器16位。在保护模式下,它的段范围不再受限于64K，可以达到4G（参见段机制一讲）。这一下真正解放了软件工程师，他们不必再费尽心思去压缩程序规模，软件功能也因此迅速提升。

从8086的16位到80386的32位处理器，这看起来是处理器位数的变化，但实质上是处理器体系结构的变化，从寻址方式上说，就是从“实模式”到“保护模式”的变化。从80386以后，Intel的CPU经历了80486、Pentium、PentiumII、PentiumIII等型号，虽然它们在速度上提高了好几个数量级，功能上也有不少改进，但基本上属于同一种系统结构的改进与加强，而无本质的变化，所以我们把80386以后的处理器统称为**IA32**（32 Bit Intel Architecture）。