深入理解计算机系统

丁杰

摘要：本文主要写了计算机组成部分，计算机启动过程，计算机系统的构成，还举出一些编程的小例子，让我们可以大致了解计算机的软硬件结合。

关键词：硬件设备，应用软件，操作系统，编写代码。

**Thorough understanding of computer systems**

**DINGJie**

**Abstract:** This paper mainly wrote the computer components, computer startup process, the composition of the computer system, also cited some examples of programming, so that we can understand the computer hardware and software combination.

**Key words:** hardware equipment, application software, operating system, writing code.

0 引言

计算机不但决定着货架上堆放着哪些产品，它们甚至能够在游戏节目中胜过真人。我们已经习惯了这样的观念，即认为电脑是需要我们用手来操作的机器，无论是使用键盘、鼠标还是触摸屏。这就是为什么非接触式人机界面是一种革命。在过去的几年里，计算机技术已经变得更加本地化、移动化，同时也更具有社交性。未来的数字化战场将转移到消费者的客厅里。一种新兴的战略是开发原创节目，以吸引和保持用户群。在过去的十年中，大型多人在线游戏——如《魔兽世界》——风靡一时。与传统的电脑游戏不同，多人在线游戏不是让你简单地与计算机比赛，而是与其他许多人在线PK。这种游戏非常引人入胜。现在，多人在线生活已经不止于游戏和聊天。美国在线教育网站Khan Academ提供成千上万的教育视频，任何入学年龄的孩子都可以在线学习各种学科的课程。该网站最近开发的“大型网上开放课程（MOOC）”向用户免费提供大学教育课程。物联网（Web of Things）可能是当前最普遍的趋势,它意味着我们接触的几乎任何物体都变成一个计算机终端。我们的房子、汽车，甚至在大街上的物体都将能够与我们的智能手机实现无缝连接，而且这些对象本身之间也是如此。在未来几年里推动这一趋势是两种互补的技术：近场通信（NFC）和超低功率芯片。近场通讯可以让互相靠近的设备进行双向的数据通信，超低功率芯片可以从周围环境中获得能量，它将能够让计算机终端变得无处不在。物联网的某些进程，如移动支付和IBM的智慧地球( Smarter Planet)计划，在短短的几年将普及开来。

1计算机组成部分



图1 计算机组成图

**1.1硬件主要分为五个部分：**

（1）控制器(Control)：是整个计算机的中枢神经，其功能是对程序规定的控制信息进行解释，根据其要求进行控制，调度程序[1]、数据、地址，协调计算机各部分工作及内存与外设的访问等。

（2）运算器(Datapath)：运算器的功能是对数据进行各种算术运算和逻辑运算，即对数据进行加工处理。

（3）存储器(Memory)：存储器的功能是存储程序、数据和各种信号、命令等信息，并在需要时提供这些信息。包括可读可写存储器RAM和只读存储器ROM。

（4）输入(Input system)：输入设备是计算机的重要组成部分，输入设备与输出设备合称为外部设备，简称外设，输入设备的作用是将程序、原始数据、文字、字符、控制命令或现场采集的数据等信息输入到计算机。常见的输入设备有键盘、鼠标器、光电输入机、磁带机、磁盘机、光盘机等。

（5）输出(Output system)：输出设备把外算机的中间结果或最后结果、机内的各种数据符号及文字或各种控制信号等信息输出出来。

**1.2软件主要分为两个部分：**

（1）系统软件是指控制和协调计算机及外部设备,支持应用软件开发和运行的系统，是无需用户干预的各种程序的集合，主要功能是调度，监控和维护计算机系统；负责管理计算机系统中各种独立的硬件，使得它们可以协调工作。

（2）应用软件是用户可以使用的各种程序设计语言，以及用各种程序设计语言编制的应用程序的集合，分为应用软件包和用户程序。应用软件包是利用计算机解决某类问题而设计的程序的集合，供多用户使用。

2计算机是如何运行的



图2计算机启动顺序图

**2.1第一阶段：BIOS**

计算机通电后，第一件事就是读取它。BIOS这块芯片里的程序叫做“基本输出输入系統”（Basic Input/Output System），简称为BIOS。

（1） 硬件自检

BIOS中主要存放的程序包括：自诊断程序（通过读取CMOS RAM中的内容识别硬件配置，并对其进行自检和初始化）、CMOS设置程序[2]（引导过程中，通过特殊热键启动）、系统自动装载程序，主要I/O驱动程序和中断服务（BIOS和硬件直接打交道，需要加载I/O驱动程序）。

（2）启动顺序

硬件自检完成后，BIOS把控制权转交给下一阶段的启动程序。这时，BIOS需要知道，”下一阶段的启动程序”具体存放在哪一个设备。也就是说，BIOS需要有一个外部储存设备的排序，排在前面的设备就是优先转交控制权的设备。这种排序叫做”启动顺序”（Boot Sequence）。打开BIOS的操作界面，里面有一项就是”设定启动顺序”。

**2.2第二阶段：主引导记录**

BIOS按照”启动顺序”，把控制权转交给排在第一位的储存设备。即根据用户指定的引导顺序从软盘、硬盘或是可移动设备中读取启动设备的MBR，并放入指定的内存的位置中。

这时，计算机读取该设备的第一个扇区，也就是读取最前面的512个字节。就叫做”主引导记录”（Master boot record，缩写为MBR）。

**2.3第三阶段：硬盘启动**

这时，计算机的控制权就要转交给硬盘的某个分区了。计算机会读取激活分区的第一个扇区，叫做”卷引导记录”（Volume boot record，缩写为VBR）。“卷引导记录”的主要作用是，告诉计算机，操作系统在这个分区里的位置。然后，计算机就会加载操作系统了。

**2.4第四阶段：操作系统**

控制权转交给操作系统后，操作系统的内核首先被载入内存。跳出登录界面，等待用户输入用户名和密码。至此，全部启动过程完成。

3计算机系统

**3.1计算机系统的概念**

计算机系统是由硬件和系统软件组成的，它们共同工作来运行应用程序。虽然系统的具体实现方式随着时间不断变化，但是系统内在的概念却没有改变。

**3.2计算机编程语言（见图3）**

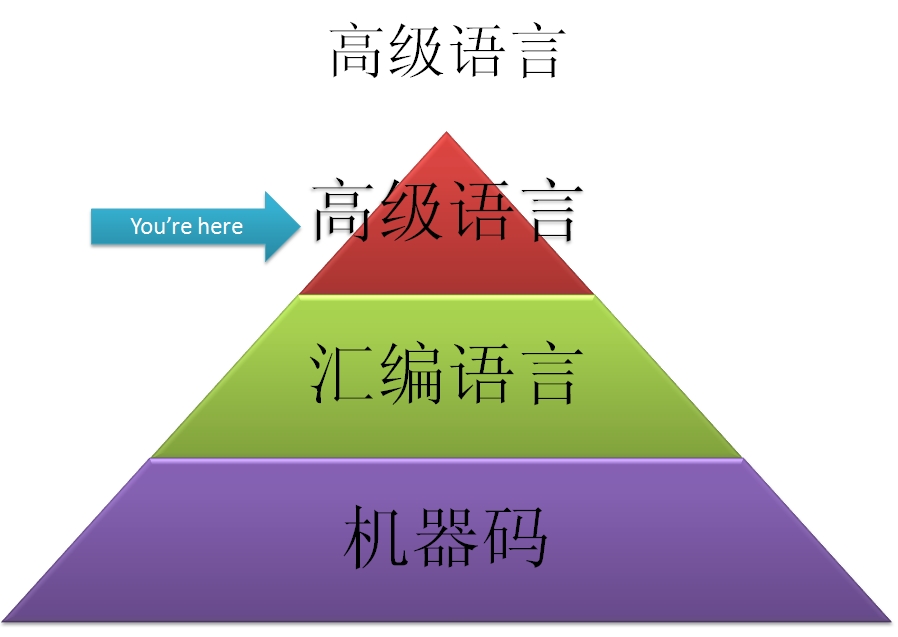


图3计算机语言

（1）机器语言

用机器语言编写程序，编程人员要首先熟记所用计算机的全部指令代码和代码的涵义。而且，编出的程序全是些0和1的指令代码。直观性差，还容易出错。

（2）汇编语言

为了克服机器语言难读、难编、难记和易出错的缺点，人们就用与代码指令实际含义相近的英文缩写词、字母和数字等符号来取代指令代码（如用ADD表示运算符号“+”的机器代码），于是就产生了汇编语言。所以说，汇编语言是一种用助记符表示的仍然面向机器的计算机语言，用汇编语言编制的程序送入计算机，计算机不能像用机器语言编写的程序一样直接识别和执行，必须通过预先放入计算机的“汇编程序“的加工和翻译，才能变成能够被计算机识别和处理的二进制代码程序。

（3）高级语言

高级语言是面向用户的语言。如今被广泛使用的高级语言有BASIC、PASCAL、C、COBOL、FORTRAN、LOGO以及VC、VB等。计算机并不能直接地接受和执行用高级语言编写的源程序，源程序在输入计算机时，通过“翻译程序”翻译成机器语言形式的目标程序，计算机才能识别和执行。这种“翻译”通常有两种方式，即编译方式和解释方式。每一种高级（程序设计）语言，都有自己人为规定的专用符号、英文单词、语法规则和语句结构（书写格式）。高级语言与自然语言（英语）更接近，而与硬件功能相分离（彻底脱离了具体的指令系统[3]），便于广大用户掌握和使用。高级语言的通用性强，兼容性好，便于移植。

**3.3 Malloc函数**

malloc的全称是memory allocation，中文叫动态内存分配,malloc 向系统申请分配指定size个字节的内存空间。返回类型是 void\* 类型。void\* 表示未确定类型的指针。C,C++规定，void\* 类型可以强制转换为任何其它类型的指针。下面举一个简单的C语言例子：

int main()

{

int \*a;

a=(int \*)malloc(4);

if(a==0)

{

exit(0);

}

\*a=100;

printf(“a=%x \*a=%x\n”,a,\*a)

free(a);

return 0;

}

首先定义一个int型指针a，由于一个int的大小等于4个字节，所以使用malloc函数给a分配4个字节的存储单元并强制转化为int型，这里需要注意的是，a本身已经占用一定的存储空间，a中保存的是动态分配的新存储单元的首地址。

**3.4数组下标越界**

int a[10];这是一个整型的数组a，有10个元素：a[0]-a[9]，因为正确的下标应该是从0开始，到9结束，与生活习惯中的1-10不一样，于是产生一个a[10]的错误，即数组下标越界。下面我举一个简单的小例子：

int main()

{  
int a[2];

int b[3];

a[0]=12;

a[1]=15;

a[2]=8;

b[0]=3;

b[1]=45;

b[2]=27;

printf(“a[0]=%d a[1]=%d a[2]=%d b[0]=%d

b[1]=%d b[2]=%d”\n,a[0] ,a[1] ,a[2] ,b[0] ,b[1] ,b[2]);

printf(“&a[0]=%x &a[1]=%x &a[2]=%x& b[0]=%x

&b[1]=%x &b[2]=%x”,&a[0] ,&a[1] ,&a[2] ,&b[0] ,&b[1] ,&b[2]);

return 0;

}

很明显，上面的程序会报错，如果幸运的编译通过，那也会在运行时出错。

**3.5虚拟内存**

电脑中所运行的程序均需经由内存执行，若执行的程序占用内存很大或很多，则会导致内存消耗殆尽。为解决该问题，Windows中运用了虚拟内存技术，即匀出一部分硬盘空间来充当内存使用。当内存耗尽时，电脑就会自动调用硬盘来充当内存，以缓解内存的紧张。

**3.6指针**

在计算机科学中，指针是编程语言中的一个对象，利用地址，它的值直接指向存在电脑存储器中另一个地方的值。由于通过地址能找到所需的变量单元，可以说，地址指向该变量单元。因此，将地址形象化的称为“指针”。意思是通过它能找到以它为地址的内存单元。

int main()

{

int a,\*p;

a=100;

p=&a;

printf("a=%x &a=%x &p=%x p=%x \*p=%x\n",a,&a,&p,p,\*p);

return 0;

}

作个比喻，假设将电脑存储器当成一本书，一张内容记录了某个页码加上行号的便利贴，可以被当成是一个指向特定页面的指针；根据便利粘贴面的页码与行号，翻到那个页面，把那个页面的那一行文字读出来，就相当于是对这个指针进行反参考的动作。

**3.7嵌入式系统**

嵌入式系统，是一种“完全嵌入受控器件内部，为特定应用而设计的专用计算机系统”，根据英国电气工程师协会的定义，嵌入式系统为控制、监视或辅助设备、机器或用于工厂运作的设备。与个人计算机这样的通用计算机系统不同，嵌入式系统通常执行的是带有特定要求的预先定义的任务。由于嵌入式系统只针对一项特殊的任务，设计人员能够对它进行优化，减小尺寸降低成本。

4结论

本文给出了计算机的基本组成结构，计算机启动顺序，计算机系统的基本概念和作用。我们还了解了计算机的发展趋势，具有相当大的前景。

【参考文献】

[1] 肖建编. 编程实例与技巧集粹. 上海：上海希望电子出版社，2007.

[2] 巴兹拉等. ASP.NET 安全性高级编程. 北京：清华大学出版社，2013.

[3] 张海藩. 软件工程导论. 北京：清华大学出版社， 2009.

githup账号:20141105043 网址：https://github.com/20141105043/