

**计算机与信息工程学院**

**课题名称：计算机系统及原理**

班 级：2014网络编程

学 号：20141105044

姓 名: 严伟

指导教师：朝力萌

完成日期：2015.12.20

Githup账号：yan786362806

Githup论文地址：

2014 级 《计算机系统》 课程论文 2015 年 12 月

深入理解计算机系统

严伟

摘要：本文就详细为大家解释了：计算机的组成部件，计算机如何工作，计算机的组成语言，计算机的内存和虚拟储存空间，数组超界。能让大家更深入的了解计算机到底是个什么，怎么工作的，用来干什么。

关键词：计算机的组成与工作原理；计算机语言；计算机的存储器

**Thorough understanding of computer systems**

YAN Wei

Abstract: In this paper, we explain in detail: the composition of the computer parts, how the computer works, the computer's composition language, the computer's memory and virtual storage space, the array of super. Can let us more in-depth understanding of the computer in the end is what, how to work, what to do.

Key words: the composition and working principle of computer; computer language; computer memory

**0 引言**

随着Internet技术的发展，人们的日常生活已经离不开网络。未来社会人们的生活和工作将越来越依赖于数字技术的发展，越来越数字化、网络化、电子化、虚拟化。计算机也随着网络的发展日益和人们的生活贴近。通过本文，用户可以了解到计算机的组成部件、如何实现工作、计算机的编程语言以及计算机的存储器。

# 1 计算机的组成部件

1.1 硬件系统

1. 控制器：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、数据、地址，协调计算机各部分工作及内存与外设的访问等。

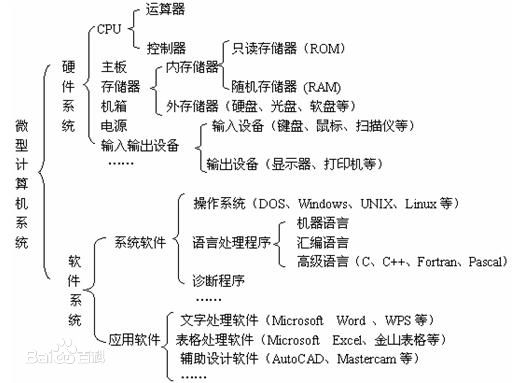
2. 运算器：运算器的功能是对数据进行各种算术运算和逻辑运算，即对数据进行加工处理。

3. 存储器：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。

4. 输入：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：输出设备与输入设备同样是计算机的重要组成部分，它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。微机常用的输出设备有显示终端CRT、打印机、激光印字机、绘图仪及磁带、光盘机等。

1.2 软件系统

软件由系统软件和应用软件组成。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。系统软件使得计算机使用者和其他软件将计算机当作一个整体而不需要顾及到底层每个硬件是如何工作的。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。计算机软件分为系统软件和应用软件两大类。应用软件是为满足用户不同领域、不同问题的应用需求而提供的那部分软件。 

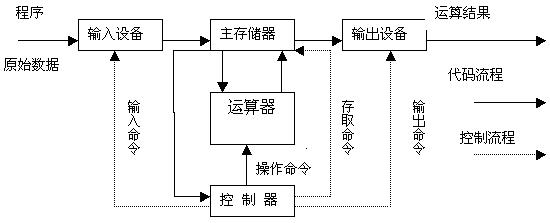
# 2 计算机是如何工作的

计算机的工作分为三个阶段:

(1)电源开启， 电流的进入，激活内部硬件设施启动,这是第一阶段工作。

(2)系统程序运行， 进行初始化，相当于所有数据归零的一个阶段，一切的指令计算等待主人的发布， 全部进入待命阶段， 这是第二个阶段工作。

(3)主人开始发布命令，例如鼠标的每一次移动,每当移动到桌面的任意一角，哪怕是一个像素点， 它都会进行计算，这种计算是靠系统程序和硬件设施的配合完成，其实并不是计算机在进行二进制的计算，而是程序进行计算后转化为一种指令， 这种指令其实已经提前被编辑程序的人员设置好。 比方说你的每次按键，它会出现你对应想要的命令，这个按键的命令背后都隐藏着数百串已经被写好的可选择或者不可选择的命令方式。而构架这种命令方式的东西就叫做“代码”， 计算机的代码都是由每一个二进制体系构成的语言。 从而我们可以获知计算机的工作原理是： 原始二进制体系（转译成）代码（编辑成）程序（发布给）硬件的动作（转化成）图文信息。



# 3 计算机的组成语言

3.1 机器语言

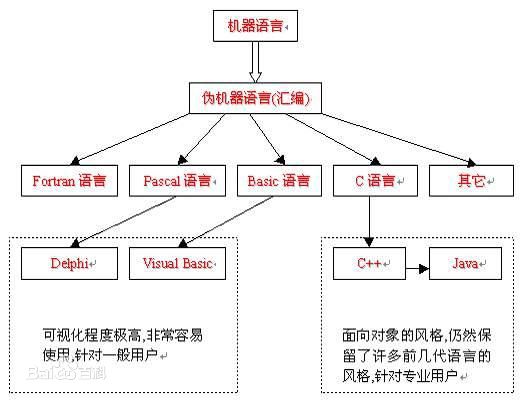
用机器语言编写程序，编程人员要首先熟记所用计算机的全部指令代码和代码的涵义。手编程序时，程序员得自己处理每条指令和每一数据的存储分配和输入输出，还得记住编程过程中每步所使用的工作单元处在何种状态。这是一件十分繁琐的工作，编写程序花费的时间往往是实际运行时间的几十倍或几百倍。而且，编出的程序全是些0和1的指令代码。直观性差，还容易出错。除了计算机生产厂家的专业人员外，绝大多数程序员已经不再去学习机器语言了。

3.2 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，汇编是一种用助记符表示的仍然面向机器的计算机语言。汇编语言亦称符号语言。汇编语言由于是采用了助记符号来编写程序，比用机器语言的二进制代码编程要方便些，在一定程度上简化了编程过程。汇编语言的特点是用符号代替了机器指令代码。而且助记符与指令代码一一对应，基本保留了机器语言的灵活性。使用汇编语言能面向机器并较好地发挥机器的特性，得到质量较高的程序。汇编语言像机器指令一样，是硬件操作的控制信息，因而仍然是面向机器的语言，使用起来还是比较繁琐费时，通用性也差。汇编语言是低级语言。但是，汇编语言用来编制系统软件和过程控制软件，其目标程序占用内存空间少，运行速度快,有着高级语言不可替代的用途。

3.3 高级语言

高级语言是绝大多数编程者的选择。和汇编语言相比，它不但将许多相关的机器指令合成为单条指令并且去掉了与具体操作有关但与完成工作无关的细节，例如使用堆栈、寄存器等，这样就大大简化了程序中的指令。由于省略了很多细节，所以编程者也不需要具备太多的专业知识。　高级语言主要是相对于汇编语言而言，它并不是特指某一种具体的语言，而是包括了很多编程语言，流行的VB、VC、Delphi等，这些语言的语法、命令格式都各不相同。



# 4计算机的存储器

在计算机的组成结构中，有一个很重要的部分，就是存储器。存储器是用来存储程序和数据的部件，对于计算机来说，有了存储器，才有记忆功能，才能保证正常工作。存储器的种类很多，按其用途可分为主存储器和辅助存储器，主存储器又称内存储器（简称内存），辅助存储器又称外存储器（简称外存）。外存通常是磁性介质或光盘，像硬盘，软盘，磁带，CD等，能长期保存信息，并且不依赖于电来保存信息，但是由机械部件带动，速度与CPU相比就显得慢的多。内存指的就是主板上的存储部件，是CPU直接与之沟通，并用其存储数据的部件，存放当前正在使用的（即执行中）的数据和程序，它的物理实质就是一组或多组具备数据输入输出和数据存储功能的集

4.1 虚拟存储空间

现代操作系统普遍采用虚拟存储器，说白了虚拟存储器是不真实存在的，是操作系统虚拟的，在磁盘上开辟的空间，该虚拟存储器可以在任何磁盘上。虚拟存储空间是通过硬件和软件的综合来扩大用户可存储空间，它在内存储器和外存储器（磁盘、光盘）之间增加一定的硬件和软件支持，使两者形成一个有机整体，支持运行比实际配置的内存容量大的多的大任务程序。程序预想放在外存储器中，在操作系统的统一管理和调度下，按照某种置换算法依次调入内存储器由CPU执行。这样，CPU看到的速度接近内存却具有外存容量的假想存储器。这个假想存储器成为虚拟存储器。具有保护模式的80286以上的CPU均支持虚拟存储空间。一般虚拟存储空间远大于实地址访存空间。

4.2 动态存储空间与堆栈和数组超界

所谓动态内存分配就是指在程序执行的过程中动态地分配或者回收存储空间的分配内存的方法。动态内存分配不像数组等静态内存分配方法那样需要预先分配存储空间，而是由系统根据程序的需要即时分配，且分配的大小就是程序要求的大小。

堆的概念：

通常定义变量（或对象），编译器在编译时都可以根据该变量（或对象）的类型知道所需内存空间的大小，从而系统在适当的时候为他们分配确定的存储空间。这种内存分配称为静态存储分配；有些操作对象只在程序运行时才能确定，这样编译时就无法为他们预定存储空间，只能在程序运行时，系统根据运行时的要求进行内存分配，这种方法称为动态存储分配。所有动态存储分配都在堆区中进行。

当程序运行到需要一个动态分配的变量或对象时，必须向系统申请取得堆中的一块所需大小的存贮空间，用于存贮该变量或对象。当不再使用该变量或对象时，也就是它的生命结束时，要显式释放它所占用的存贮空间，这样系统就能对该堆空间进行再次分配，做到重复使用有限的资源。

在C++中，申请和释放堆中分配的存贮空间，分别使用new和delete的两个运算符来完成：指针变量名=new 类型名(初始化式)；delete 指针名; 例如： int \*pi=new int(0);注意在堆中申请一块存储空间，当再使用该变量或对象时，也就是它的生命结束时，一定要释放它所占用的存储空间，这样系统就能再对该堆空间进行再次分配，否则便会出现BUG，程序可运行，但是时间长了会死机，崩溃等。

**举例**：我们在定义一个数组时，应该定义为多大？定义不同的数组需要根据各自的实际情况，不知道我们要定义的这个数组到底有多大，那么你就要把数组定义得足够大。这样，你的程序在运行时就申请了固定大小的你认为足够大的内存空间。即使你知道你想利用的空间大小，但是如果因为某种特殊原因空间利用的大小有增加或者减少，你又必须重新去修改程序，扩大数组的存储范围。这种分配固定大小的内存分配方法称之为静态内存分配。但是这种内存分配的方法存在比较严重的缺陷，特别是处理某些问题时：在大多数情况下会浪费大量的内存空间，在少数情况下，当你定义的数组不够大时，可能引起下标越界错误，甚至导致严重后果。

#include <iostream.h>

#include<string.h>

Void main()

{

Int n;

char \*pc;

cout<<”请输出动态数组的个数”<<endl;

cin>>n;

pc=new char[n];

strcpy(pc,”堆内存的动态分配”);

cout<<pc<<endl;

delete [ ] pc;

return 0;

}

数组超界例子：

Void test  
 {

char string[10];

char \*str1= "0123456789";

strcpy(string, str1);

}

str1是一个字符串，C风格的字符串以‘\0’结束，所以str1虽然看上去只有10个字符，但实际上应加上‘\0’,即应为11个字符，而string数组只有10个元素，而strcpy()函数的实现是遇'\0'而停止拷贝，所以拷贝时实际上是把‘\0’也进行了拷贝，这就导致string数组越界拷贝。

类似这样的错误还有进行浮点运算时，因为浮点溢出，付出了高昂的代价。Ariane 5火箭在1996年6月4日初次航行，因为犯了将大的浮点数转换成整数这一常见的程序性错误，便产生了灾难性的后果，火箭发射后仅仅37秒，火箭偏离了他的飞行路径，解体并且爆炸，损失了价值5亿美元的通信卫星。



参考文献

[1]Google—http://windows.microsoft.com/zh-cn/windows/computer-parts#1TC=windows-7

[2] Randal E.Bryant David R.O’Hallaron . 《深入理解计算机系统》 2014，3

[3] 百度文库

<http://baike.baidu.com/link?url=4qz1otOG3NuPvkbbcIndWKRi_MjZuugmr3Iw79ncDoiHTOU085b6mKkIjdSKOibYWMzFDZq3PdxSAOwnlhYBiK>.

<http://baike.baidu.com/link?url=urz0_cP86tqaMeTuD_RjY9hxZZ6p9xfQz7eTcnGKjyBvXgCJst099E-1TAkywZBK8p-V4noCkdX1EPCeplLF8K>