深入理解计算机系统

计算机与信息工程学院 2014级网络编程 卢佳明 20141105055

指导教师 朝力萌

摘要 本文主要研究了计算机基本组成部件知识，计算机的工作原理，其中包括了对指针，计算机的动态存储分配，嵌入式开发的理解和描述。

关键词 存储器，指针，动态存储分配

0 引言

本文主要研究计算机系统结构，在书写的过程中参考了相关书籍和网页资料，对于参考的部分或借鉴的部分有小标在文段末尾注释，并在参考文献中明确指出。

1.计算机的组成部件

指的是系统结构的逻辑实现，包括机器机内的数据流和控制流的组成及逻辑设计等。见图1.1

**1.1硬件组成**

（1）控制器：在整个计算机中起着举足轻重的作用，它担负着对整个计算机的各个功能的调控，例如对于应用程序的调度（何时工作，及其工作时间的调度），并协调其与外设和内存间的工作。

（2）运算器：对（二进制）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。

注：在计算机中实际完成以上两项工作的是CPU（CPU=运算器+控制器）[2]

（3）存储器[3]：存储各类数据和各类信号，命令。其中存储器分为内存储器和外存储器，内存储器包括只读和随机两类存储器。其英文简称分别为ROM,RAM,这两个英文简称在生活中很是常见，现也常见说手机内存储器的配置，其中，ROM只能读出信息，不能写入信息，计算机关闭电源后其内的信息仍旧保存，一般用它存储固定的系统软件等。RAM表示的是读写存储器，可对计算机的其中任一存储单元进行读或写操作，计算机关闭电源后其内的信息将不在保存，再次开机需要重新装入，通常用来存放操作系统，各种正在运行的软件、输入和输出数据、中间结果及与外存交换信息等。外存储器包括硬盘，光盘，软盘[6]等。

（4）总线：贯穿整个系统的一组电子管道，它携带信息字节并负责在各个部件间传递。一般总线被设计成传送定长的字节，字中的字节数是一个基本的系统参数，一般的大多数机器的字长为4个，即32位，有的是64位。

（5）输入输出设备（IS/OS）：具有将要计算的命令，数据，信号输入计算机并将计算后的结果（或中间结果）输出到计算机的显示终端上的作用。一般的IO设备有键盘，鼠标，扫描仪，显示器等。

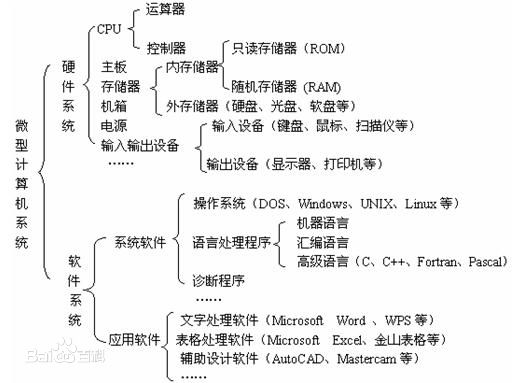


图1.1

**1.2计算机的软件组成**

（1）操作系统：是管理和控制计算机硬件与软件资源的计算机程序，是直接运行在“裸机”上的最基本的系统软件，任何其他软件都必须在操作系统的支持下才能运行[4]。操作系统的作用可以在计算机的各个方面，主要是控制计算机内各程序间的运行，管理计算机系统的数据资源，还可以同时管理计算机的软件和硬件。为用户提供一个更好的使用界面，并为其他类型的软件提供必要的服务和接口。并能够防止硬件被失控的应用程序滥用。一般情况下，操作系统在正常工作时是不需要外界干涉的，操作系统能够根据应用程序的需求合理划分并分配资源（储存空间，CPU的划分）现今较为实用的操作系统有：DOS,Windows,UNIX,Linux等。见图1.2



图1.2 操作系统图例

（2）应用软件：文字处理软件：Microsoft Word ，WPS

表格处理软件：Microsoft Excel ，金山表格

辅助设计软件：AutoCDA等

2.计算机工作原理

计算机的基本原理是存储程序（数据）和程序控制[5]。

见图2

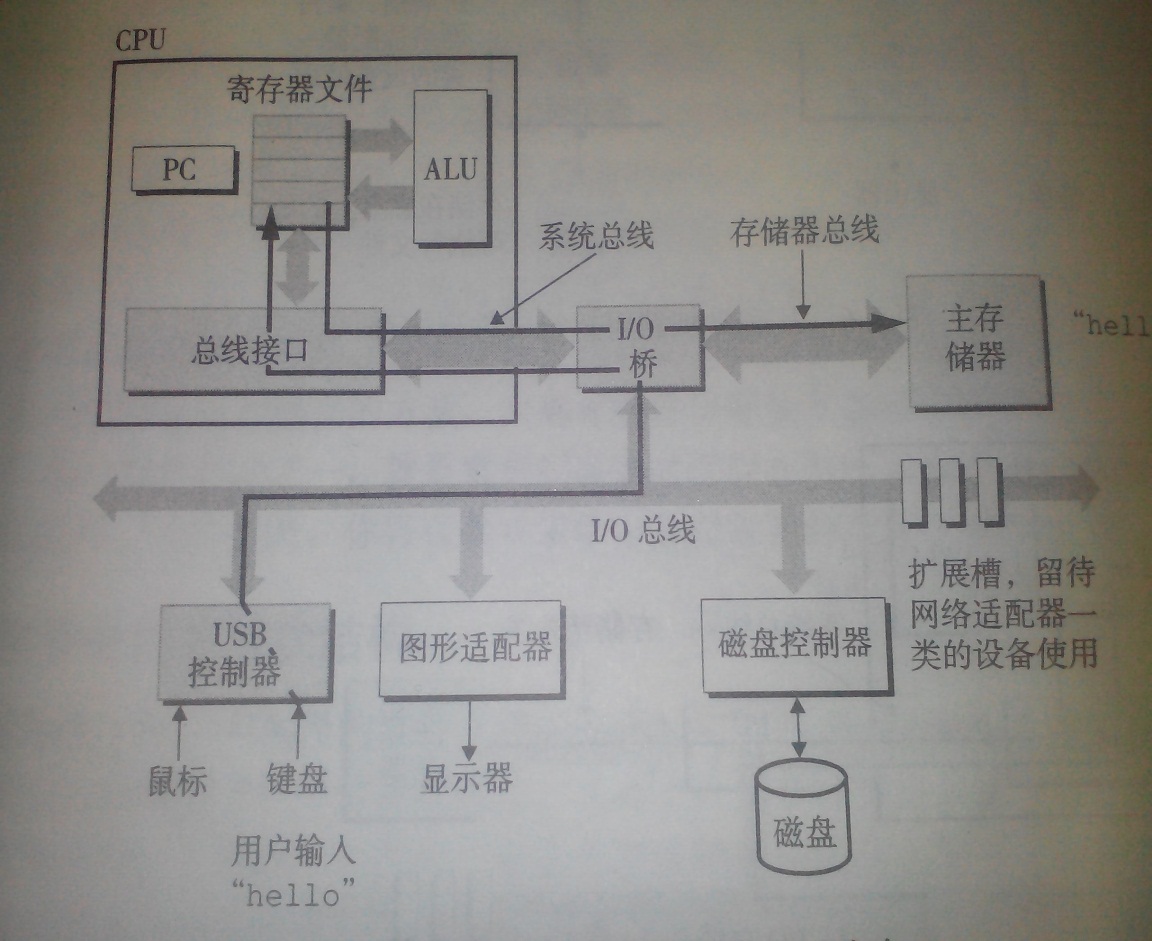


图2 计算机工作原理

**2.1输入设备的数据输入**

我们通过在输入设备（多用为接口USB）上输入数据（信号，命令）后，计算机把输入信息转换成能为计算机处理数据的形式。计算机输入的信息有数字、模拟量、文字符号、语声和图形图像等形式。对于这些信息形式，计算机往往无法直接处理，必须把它们转换成相应的数字编码后才能处理，然后将处理好的数字编码通过I/O总线输入到计算机的其他部件内进行进一步处理。

**2.2数据的缓存**

通过输入设备输入的数据通过I/O总线，经过I/O桥[7]将数据存在主存储器中，再通过存储器总线将数据传输到总线接口，经此再将数据存入寄存器中，最后将数据交由中央处理器处理计算。其中主存储器是一个临时存储设备，在执行程序时，用来存放程序和程序处理后的数据。一个典型的寄存器文件只存储几百个字节的信息，而主存储器里可存放几亿字节，但是处理器从寄存器读取的速度是从主存储器上读取的速度的一百多倍。因而针对这样的差异,产生了“高速缓存储存器”，即高速缓存。高速缓存通过一条特殊的总线连接到处理器，见图2.2。

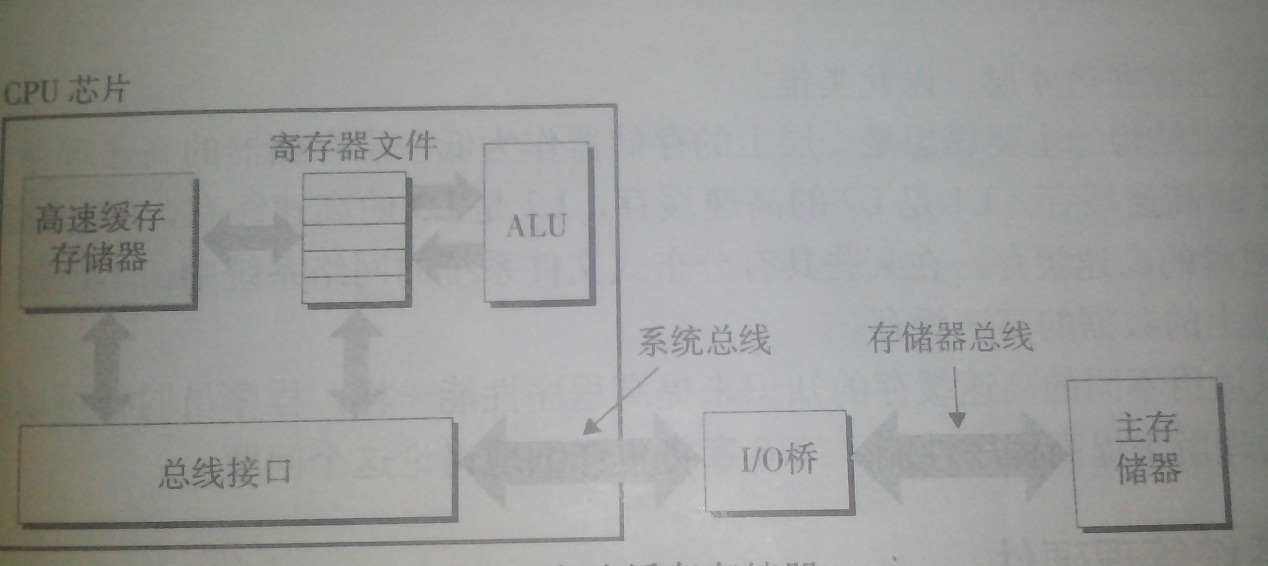


图2.2 高速缓存

**2.3数据的处理**

中央处理单元（CPU）,是执行储存在主存储器中的指令的引擎，主要负责计算机的数据处理和程序控制功能[8]。

3．程序语言

**3.1机器语言**

机器语言是一种指令集的体系。这种指令集，称机器码（machine code），是电脑的CPU可直接解读的数据。[9]机器代码用字节编码低级的操作，包括处理数据，管理存储器，读取存储设备上的数据，一般为二进制数，0，1表示。

特别注释：机器语言是计算机唯一可识别的语言，是最基础的语言。

**3.2汇编语言**

汇编语言（assembly language）是一种用于[电子计算机](http://baike.baidu.com/view/6373.htm)、[微处理器](http://baike.baidu.com/view/1125.htm)、[微控制器](http://baike.baidu.com/view/368297.htm)或其他可编程器件的低级语言，亦称为符号语言。是一种指令集的体系。这种指令集，称机器码（machine code）[10]。见图3.2

自身特点：它是一种面向机器的低级语言，通常是为特定的计算机或系列计算机专门设计的。因为它是机器指令符号化的表示，所以不同的机器就会有不同的汇编语言。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。因为针对不同的机器汇编语言有其自身的特异性，所以它相对与高级语言汇编语言对特定的机器有着特定的联系。而在使用汇编语言时，也有其自身的优点，因为汇编语言代码组成简单，所占空间小，所以由汇编语言编辑的程序运行时间相对与高级语言下的程序快得多。但又是因为其特异性，限制了其在多处理器上的执行。

**3.3高级语言**

高级语言相对于机器语言。（是一种指令集的体系。这种指令集，称机器码，是电脑的CPU可直接解读的数据[11]

自身特点：高级语言是基于自然语言的一类编程语言，而且高级语言与计算机的硬件结构及指令系统无关，它有更强的表达能力，可方便地表示数据的运算和程序的控制结构，能更好的描述各种算法，而且容易学习掌握。现在使用较多的高级语言有java，c，C++，C#，pascal，python，lisp，prolog，FoxPro等语言。

**3.4不同程序语言间的联系与区别**

机器语言相对汇编语言和其他高级语言来说，有着其不可取代的地位，因为它是最为基础的语言，任何语言的使用都基于机器语言，或换言之都最终要转化机器语言来最终实行，但机器语言在日常生活中我们并不会直接接触，我们是通过高级语言或者汇编语言进行语句的书写，在经过编译器转化成机器可识别的机器语言。因为机器语言进行程序设计的思维和表达方式与人们的习惯大相径庭，只有经过较长时间职业训练的程序员才能胜任，所以如今已经很少有直接使用。汇编语言与高级语言相比而言，汇编语言是高级语言的低级表达。相反的，无论什么样的高级语言都得汇编并经编译器编译成机器可直接识别的机器语言，三者相辅相成相互转化，各有优缺点。它们分别尤其是由它们分别有自己使用的相应领域。

有其存在的价值所在。

**4.存储器**

其概念在上文已做过相应的介绍，即存储器的主要功能是存储程序和各种数据，并能在计算机运行过程中高速，自动地完成程序或数据的存取。存储器在各种不同的需求下，衍生出很多种类，见图4

表4 存储器分类

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 存储器 | 功能 | 寻址方式 | 掉电后 | 说明 |
| 随机存取存储器（RAM） | 读、写 | 随机寻址 | 数据丢失 | 工作时写入数据 |
| 只读存储器（ROM） | 读 | 随机寻址 | 数据不丢失 | 工作前写入数据 |

**4.1数组越界**

先从数组数据存储上浅理解数据的保存，定义两个数组，一个是有四个元素的数组，另一个是有五个元素的数组，并给它们数值并将它们的数值和地址以十六进制的形式输出，在多次编译下可以看出，有时会发生a数组的末尾值和b数组的顶端的值发生覆盖，（实例见例4.1）在地址的输出也会出现相同的状况，这其实就是空间上的不合法利用和分配，因为计算机在一个相同的应用程序下，应用程序所定义的变量会随机存储到一个指定区域内，但在这个空间内数据的存储是随机可变的，元素个数的数组内变量的存储地址一般是间断的，在每一段数据的储存中会有一些“碎片”，术语中将其称之为“磁盘碎片”。而之所以发生了数据覆盖，正因为其存储末端和存储起始点重叠，非法使用了空间。

int main(int argc, const char \* argv[])

{

int a[4];

int b[5];

a[0]=12;

a[1]=15;

a[2]=25;

a[3]=55;

b[0]=2;

b[1]=3;

b[2]=4;

b[3]=5;

b[4]=6;

printf("a[0]=%d b[0]=%d a[1]=%d b[1]=%d a[2]=%d b[2]=%d a[3]=%d b[3]=%d ",a[0],b[0],a[1],b[1],a[2],b[2],a[3],b[3]);

printf("&a[0]=%x &a[1]=%x &a[2]=%x &a[3]=%x &b[0]=%x &b[1]=%x &b[2]=%x，&b[3]=%x&b[4]=%x\n" ,&a[0],&a[1],&a[2],&a[3],&b[0],&b[1],&b[2],&b[3],&b[4]);

printf("&a[7]=%x &b[1]=%x &b[2]=%x &b[3]=%x &b[4]=%x\n",&b[1],&b[2],&b[3],&b[4]);

return 0;

}

例4.1 数组越界

**4.2动态存储分配**

在4.1中的数组越界描述中，也隐含的体现了存储分配的理念，但与数组的静态分配不同，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，这也是动态存储分配的优势所在，分配的空间可以根据程序的需要扩大或缩小，避免了储存空间的浪费，极高的实现了空间的利用率。下面就

一些实例来做具体的解释。

int main(int argc, const char \* argv[]) {

// insert code here...

std::cout << "Hello, World!\n";

int \*a;

int N;

int i,j,temp;

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

for (i=0;i<N;i++)

scanf("%d",&a[i]);

for (i=0;i<N;i++)

printf("%d",&a[i]);

free (a)；

return 0;

}

例4.2 动态存储分配实例

在上面的例子中，程序的目的是实现几个数字的大小排序，在数据的存储中用到了动态存储的方法a=(int \*)malloc(N\*4);即分配n个可连续分配的整型空间，每个空间占4个字节。

**4.3指针**

指针是一个用来指示一个内存地址的变量，指针一般指向一个函数或一个变量。在使用一个指针时，一个程序既可以直接使用这个指针所储存的内存地址，又可以使用这个地址里储存的函数的值。在计算机语言中，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。下面举一些实例来说明，见例4.3。（这是上课是老师讲到的关于指针的第一个实例）

#import <Foundation/Foundation.h>

int main(int argc, const char \* argv[])

{

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;

}

例4.3

在例4.3中，定义了一个整形a，和整形变量型指针，将值100赋予a，将a的地址赋给p，之后将a的值还有a的地址，p的值和p指针指向的值以十六进制形式输出，输出结果正好应证了指针的意义，即指针是一个用来指示一个内存地址的变量。(输出结果\*p=a，a的地址的十六进制表现形式与p的值相同)。

**4.4虚拟存储技术**

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。若计算机运行程序或操作所需的随机存储器(RAM)不足时，则 Windows 会用虚拟存储器进行补偿。它将计算机的RAM和硬盘上的临时空间组合。当RAM运行速率缓慢时，它便将数据从RAM移动到RAM和硬盘上的临时空间组合中。将数据移入分页文件可释放RAM，以便完成工作。（12）

**4.5嵌入式系统**

嵌入式系统，用于控制、监视或者辅助操作机器和设备的装置，，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。嵌入式系统通常进行大量生产，所以单个的成本节约，能够随着产量进行成百上千的放大。

[参考文献]

[1] 张海藩. 计算机基础导论. 北京：清华大学出版社， 2003.

[2] 周佩德.计算机原理及应用〔M〕.北京：电子工业出版社，2004.

[3] 刘炳文等.VISUAL BASIC 程序设计——数据库篇〔M〕.北京：人民邮电出版社，1999.

[4] 李光明.Visual Basic 6.0 编程实例大制作〔M〕.北京：冶金工业出版社，2002.

[5] 王兴晶，赵万军等.Visual Basic 软件项目开发实例[M].北京：电子工业出版社,2004.

[6] 陈艳峰，高文姬等.Visual basic 软件项目案例导航[M].北京:清华大学出版社,2004.

[7] 李红等.管理信息系统开发与应用〔M〕.北京：电子工业出版社，2003.

[8] 周之英.现代软件工程〔M〕.北京：科学出版社，2000.

Ｇｉｔｈｕｂ　用户名　２０１４１１０５０５５