深入了解计算机系统

15-网编 钟洋洋

摘要：通过学习计算机系统这门课程。是我学习到了许多有关计算机系统、编程、以及Arduino等知识。通过老师的讲解让我对计算机组成、计算机工作原理、计算机语言、Arduino运用等知识有了一定的深入了解。

关键字：计算机组成、计算机语言、Arduino

Deep understdanding of computer systems

Zhong yang yang

Abstract: through the course of learning computer system. I learned a lot about computer systems, programming, as well as Arduino and other knowledge. Through the teacher's explanation let me to the computer composition, the computer working principle, the computer language, the Arduino application and so on knowledge has the certain thorough understanding.

Keywords: computer composition, computer language, Arduino.

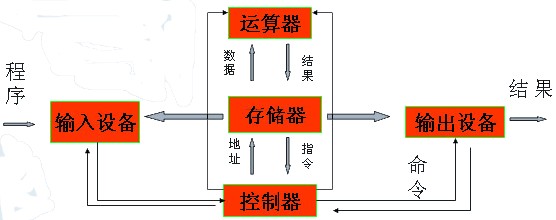
0.引言：通过一个学期的学习。使我深入的了解到计算机系统这门课程。也让我对这门课程产生了极大的兴趣。这次论文主要就计算机系统、编程中的链表以及Arduino基础知识等向大家介绍一下。希望通过本次论文可以更加深入的了解计算机系统。

一：计算机系统：

1.1计算机系统是由硬件和软件组成的。共同工作使应用程序运行。系统的硬件由总线、I/O设备、主存以及处理器组成。总线：是贯穿整个系统的一组电子管道，它负责将信息字节在每个部件中传递。I/O设备：是将系统与外部世界连接起来。简单的I/O设备有键盘、鼠标、显示器以及磁盘驱动器。主存：是一组动态随机存取存储器［注1］。主存在系统中起到的作用是在处理器执行程序时，用来存放程序和程序处理的数据。处理器：是中央处理单元,又叫CPU.处理器是按照一个简单的指令执行模型来操作的。CPU又是由PC 、ALU、寄存器文件以及总线接口组成。ALU：算术／逻辑单元。PC:程序计数器［注2］。

1.2操作系统有两个基本功能：1.防止硬件被失控的应用程序滥用。2.向应用程序提供简单一致的机制来控制复杂而又通常大相径庭的低级硬件设备［注3］。

计算机硬件组成图：



1.3计算机系统软件：

软件是计算机用户与硬件之间的接口界面。用户主要通过软件与计算机交流。

软件的含义：1.运行时，能够提供要求的功能和性能的指令或程序结合。

2.程序能够处理信息的数据结构。

3.描述程序功能需求以及程序如何操作和所使用所要求的文档。

特点：1.与硬件的交互性。

2.能对资源共享进行调度管理。

3.能解决并发操作处理中存在的协调问题。

4.其中的数据结构复杂，外部接口多样化，便于反复使用。

软件的分类：系统软件是各类的操作系统，如windows、linux、UNIX等，还包括操作系统的补丁程序以及硬件驱动程序。

操作系统能控制所有计算机运行的程序并管理整个计算机资源。操作系统是计算机系统的控制和管理中心，从资源角度来看，它具有处理机、存储器管理、设备管理、文件管理等功能。

操作系统在计算机系统中主要有两方面：对内，操作系统管理计算机系统的各种资源，扩充硬件。对外，操作系统能提供良好的人机界面，方便用户使用计算机。

四个基本特征：并发性、共享性、虚拟性和不确定性。

二：计算机语言：

2.1：机器语言：

机器语言包含了计算机中CPU的指令集，是由二进制0、1代码指令构成，指令集包含的指令是CPU能够理解，不同的CPU具有不同的指令系统。用机器指令编写的程序通常称为机器代码。机器语言程序的效率是最高的，但是机器语言程序难编写、难修改、难维护和难调试，需要用户直接对存储空间进行分配，编程效率极低。因此现在的程序很少用机器语言编写。

机器语言是计算机唯一能接受和执行的语言。机器语言由二进制码组成，每一串二进制码叫做一条指令。一条指令规定了计算机执行的一个动作。一台计算机所能懂得的指令的全体，叫做这个计算机的指令系统。不同型号的计算机的指令系统不同。 指令通常由几个字节组成，第一个字节是操作码，它规定了计算机要执行的基本操作；后面的字节是操作数，它规定了操作对象或操作对象的地址。 机器语言是直接用二进制代码指令表达的计算机语言，指令是用0和1组成的一串代码，它们有一定的位数，并分成若干段，各段的编码表示不同的含义。【注4】

2.2：汇编语言：

汇编语言指令是机器指令的符号化，与机器指令存在着直接的对应关系，也就是说机器指令被助记符所替代。因此，汇编语言编写的程序比较接近英语，编写和调试比较容易。但这些程序在执行前被转化为CPU能理解的机器语言。所以汇编语言存在着难学难用容易出错等缺点。优点在于可直接访问系统接口，汇编程序翻译成机器语言后程序的效率高。

2.3：高级语言：

高级语言形式上接近于算数语言和自然语言，概念上接近于人们通常使用的概念。高级语言的一个命令可以代替很多条汇编语言的指令。因此，高级语言易学易用，应用广泛。

2.4：指针

2.4.1：地址：在计算机运行程序的过程中，计算机是通过内存的编号也就是内存地址找到数据存储的位置，从而对数据进行存取操作的。计算机程序指令从内存中存取每一条数据都需要明确指出数据存储的内存地址才可以。

指针的概念：指针包含有指示器的含义，指针就是内存地址。在C语言中内存地址就是指针。

2.4.2：指针变量的定义：在C语言中规定变量使用前必须先定义，通过变量的定义说明变量所能储存的数据类型及变量在内存中所占用存储空间大小。指针变量用于存储内存地址。

指针变量的引用：有了指针变量后，指针变量中我们可以存储数据在内存中的地址，这样我们通过变量的值也就是数值的地址可以直接访问数据。

有两个与指针相关的运算符&和\*。其中&是取地址运算符，\*是指针运算符。

2.4.3：指针与数组：指针与数组有密切的联系，数组的名字其实就是数组在内存中所占存储空间的起始地址。因此我们可以认为数组名就是数组的指针。数组名是个常量，数组名也就是个常量指针，一旦定义了一个数组，数组的起始指针就是固定不变的常量值，直到数组销毁释放存储空间为止。

指向数组元素的指针：数组中的数据元素是在内存中占有连续的存储空间，因此只要知道一个数组的第一个数据元素的内存地址，我们就可以根据这个数组数据元素的数据类型大小，能够计算出这个数组的每一个数据元素在内存中的存储地址。这样我们就可以通过数组的第一个元素的地址访问数组的其他任何元素。

将数中第一个元素的地址存储到一个指针变量中，我们就称指针指向了这个数组。

通过指针引用数组元素：将一个数组的起始地址保存到一个指针变量后，我们可以通过这个指向数组的指针来访问数组的任一元素。

用数组名作函数参数：在C语言中数组名其实就是一个常量指针，这个常量指针的值就是数组的起始地址。

2.4.4：在编写函数的时候，有时需要向函数传递多个数值或从函数返回多个值，用一般简单变量传递数据时传递过去的数据量是非常有限的，那么我们可以通过向函数传递数组的起始地址的方法，向函数内传递过去数组起始地址后，函数可以根据数组的起始地址访问数组的所有元素，这样可以达到向函数传递多个数值或从函数返回多个数值的目的。

字符串与指针：字符串的值就是字符串的起始地址。

2.4.5：在程序中使用数组有个缺点，第一，数组需要连续的存储空间，当内存存储空间碎片较多时需要对碎片进行清理整合后才可以使用较大的存储空间。第二，数组的大小必须事先定义好元素个数，不能在程序运行过程中根据需要随时增减数组存储空间的大小。针对数组的这些缺陷，链表完全克服利用数组存储数据的缺陷。

下面让我们了解了解链表的知识。

链表的原理：应用数据分别存储在叫作结点的存储块儿中，每一个结点中除了存储应用数据以外还要存储下一个结点的地址，如此形成一个链式存储结构。因此在一个链表中我们只需记住第一个结点的地址，通过第一个结点的地址，通过第一个结点的地址我们就可以访问整个链表的所有的结点。

链表中的头一个结点称为头结点，链表中最后一个结点称为尾结点。

对于一个链表来说，链表的头指针非常重要，只要我们知道链表头指针，就可以通过头指针访问链表中的所有结点。

2.4.6：指针和函数：在C语言中指针可以成为函数的参数，通过使数据的指针作为函数的参数，可以向函数内部传递大量数据或从函数返回大量数据，这种地址传递只是把数据的地址传递过去，从而节省了因值传递时复制数据参数所耗费的时间。在C语言中指针也可以指向函数，通过指向函数的指针就可以调用函数，在这程序中使用静态链接库时经常用到。

指向函数的指针：在C语言中，如果我们知道函数的地址，即使不知道函数的名字，也可以通过函数的入口地址调用此函数。我们只需将函数的入口地址赋给一个函数指针变量，那么通过函数指针变量就可以调用这个函数。函数指针变量的一般定义形式为：

函数类型 （\*变量名）（函数形式参数表）。

返回指针值的函数：在某些应用场合，我们需要函数返回某个存储空间的地址，也就是函数返回指针值。函数返回指针值的一般形式为：

类型名 \* 函数名（函数参数表列）。