

**计算机与信息工程学院学生会**

**工**

**作**

**总**

**结**

**计算机与信息工程学院**

**14级软件工程（网络编程）**

**梁爽 20141105040**

https://github.com/ls123456789

**2016.1.7**

深入理解计算机系统

梁爽

[摘要]：本论文主要讲述的为计算机系统。其中包括计算机系统的组成部件、工作原理、嵌入式系统及计算机语言。通过对不同部件的功能和工作方式的解释说明，来描述计算机的系统组成和工作方式。

[关键词]：计算机系统；工作原理；嵌入式系统

**Depth understanding of computer systems**

LIANG Shuang

[Abstract]: This paper describes the computer system. Including the components of a computer system, working principle, embedded systems and computer languages. By functions and working methods of the various components of explanation, to describe the system composition and working methods of the computer.

[Key words]: computer system; works; embedded systems

0 引言

本文分为四个大标题，分别描述了计算机的组成部件、计算机的编程语言、计算机是如何工作的以及嵌入式系统。通过对计算机的组成部件的描述，重点突出计算机的存储器在计算机系统中的重要性，以及存储器系统中的虚拟内存和指针。

1. 计算机的组成部件

计算机的组成部件基本可以分为三类：中央处理器（CPU）、输入\输出设备（I/O设备）、存储器。计算机的这三大组成部分都是由一个总线作桥梁连接在一起的。它们之间相互协作，分工明确，共同完成计算机的各项功能。

* 1. **CPU**

CPU既中央处理器，CPU是电脑的核心组成部件 ，它相当于计算机的大脑，主要进行运算和对计算机整体的控制，是执行存储在主存储器中的指令的引擎。CPU 还包括以下几个部件，如下图1.1.1，中央处理单元（ALU）、寄存器、程序计数器（PC）、总线接口。

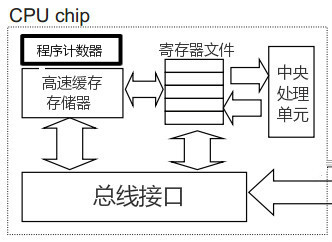


图1.1.1

程序计数器是用来存储数据的，它所存储的数据的类型为地址，既中央处理单元即将处理的下一条指令的地址。如下图1.1.2，这中的指令计数器，中央处理单元（运算器）每处理一条指令，PC中的地址或自动加1或由指针给出下一条指令的地址，由此可见，程序计数器也属于寄存器的一种。

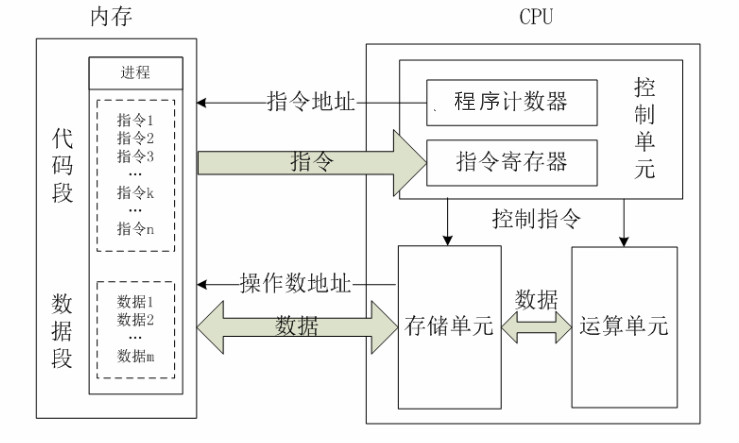


图1.1.2

* 1. **I/O设备**

i/o设备是指数据的输入输出设备，一般包括键盘鼠标，显示器，打印机等。其中磁盘驱动也属于i/o设备。i/o设备是计算机系统与系统外连接的通道。一般来讲，它们都是由一个控制器或适配器链接到i/o总线上来进行数据的传递的。适配器与控制器的区别在于它们的封装方式不同，控制器是I/O设备本身带有的或是计算机的主板上控制器中的一部分；而适配器则是一种接口转换器，是一个独立的硬件设备，一般插在计算机主板的扩展槽上。两者相同点就是都是为I/O总线和外部设备上信息传递。

* 1. **存储器**

存储器系统是一个有不同容量、成本和访问时间的存储设备的层次结构。存储器根据其数据传输速度和存储字节量可分为很多的等级，如下图1.3，越靠近金字塔尖的存储器数据传输速度越快，但是每个字节的成本越高，所以在实际的计算机中可存储的数据就越少；而反之，越靠近金字塔底部的存储器数据传输速度越慢，但是每个字节的成本就越便宜，在实际的计算机中可存储的数据的量就越大。由于现阶段CPU处理的速度远远超过SRAM和DRAM传输的速度，所以我们需要速度更快的寄存器。如果用低速的存储器来为CPU提供数据，CPU只有小部分时间在工作，而大部分时间是闲置的，这样属于浪费CPU的资源，CPU不能被充分的利用，计算机的运算速度就不能提高；如果将寄存器这样的材料和工艺用作主存储器来存储数据，就需要大量的资金，这样会导致计算机造价太高，不能被普及。所以，计算机就需要像金字塔一样的存储器系统，用比寄存器慢，比主存储器快的高速缓存来做桥梁，使得寄存器和主存储器起到速度和大小的互补作用。

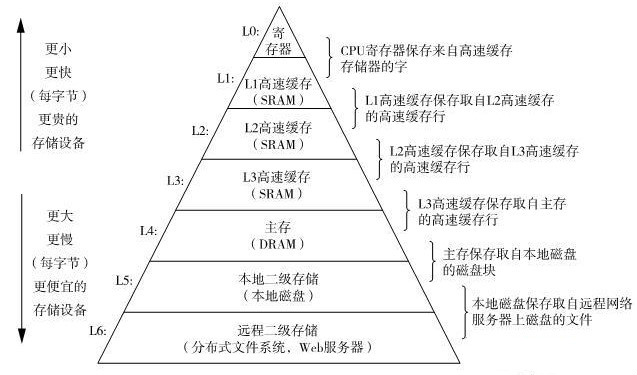


图1.3

* + 1. **寄存器**

寄存器既属于存储器系统，也属于CPU系统，一般用来存储指令、数据和地址。寄存器在CPU中不止一个，或者说分为很多种，例如在中央处理器的控制部件中，包含的寄存器有指令寄存器和程序计数器。在中央处理器的算术及逻辑部件中，寄存器有累加器。

* + 1. **高速缓存与主存**

高速缓存是作为CPU和主存之间的缓存区域，在访问的时候有差别，高速缓存为静态访问，既SRAM；主存为动态访问，既DRAM。。静态RAM比动态RAM的数据传输速度快，高速缓存是在CPU中的部件，而主存是插在主板的扩展槽上的。它们有一个共同点，当计算机断开电源后，SRAM和DRAM都会丢失数据。在对主存进行访问时，需要总线作桥梁来进行数据流通，而高速缓存则不需要。

* + 1. **虚拟内存**

在Windows的操作系统中，虚拟内存是作为内存使用的一部分硬盘空间，在硬盘上就是一个硕大无比的文件，它在运作的时候需要硬件和操作系统软件的紧密配合。它的工作方式可以通俗的理解为“循环使用”，将一段程序分为几个大小小于它容量的部分，在程序运行的过程中，通过软件的控制和硬件数据、程序的交换来实现，即将需要用到的或即将用到的程序导入虚拟内存，再将执行过的或暂时不需要的程序存入磁盘。

虚拟存储器中，有几个重要的区，每个区都有准确的定义和专门的功能。用Linux的虚拟地址空间为例，地址是由下向上增大的。在虚拟地址空间的底部是只读的代码和数据，这一部分是可执行的代码或数据，紧接着是可读可写的数据区域，这两部分都是在进城一开始运行时就被规定了大小和位置。

在这两部分的上面是“堆”区，运行时堆上面的是一部分是给堆预留的一部分空间，因为运行时堆是在运行动态存储创建时用到的，所在堆的大小可以随着程序的运行而变化。在堆的顶端存在一个brk指针，使得新分配的空间占据合理的位置，在释放或者访问时也起到很重要的作用。

在运行时堆的上面是共享库，再上一层为栈。栈位于虚拟地址空间的顶部，编译器主要用它来实现函数的调用，同样的栈的大小也会随着程序的运行而改变，调用函数时，栈就会增大，从函数返回时，栈就会缩小。栈有相应的栈帧结构，栈用来传递过程参数、存储返回信息、保存寄存器用于以后恢复，以及本地存储。为其中每个过程分配的栈的空间称为栈帧，既栈主要由多个栈帧组成。在栈中也存在指针，分别为寄存器%ebp(帧指针)和寄存器%esp(栈指针)。早大多数信息的访问都是相对于帧指针的。

虚拟存储器地址空间的顶部区域是留给内核的，内核是操作系统的一部分，该部分程序受到操作系统的保护，不会被应用程序改变或占用。

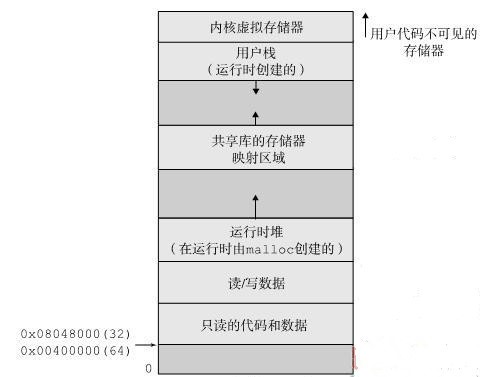


图1.3.3

* + 1. **动态存储分配**

动态存储分配用到的空间为上文讲述的虚拟内存中的运行时堆。动态存储分配需要用到一个动态存储分配器来控制运行时堆。分配器将堆中的空间分为很多的块来管理，每个块都是一个连续的被使用或已释放的空间。

分配器有两种基本风格，一种为显式分配器，另一种为隐式分配器。显示分配器既按照合法的分配或释放要求来控制管理堆的空间。一般有标准C中的malloc函数和free函数、C++中的new操作符和delete操作符。隐式分配器主要通过检测识别来判断数据是否有用，如果不再被程序使用，就会自动释放该空间。这个过程也叫做垃圾收集，隐式分配器也叫做垃圾收集器。

在堆中，引入一个碎片的概念。碎片指较小的，不能单独存储数据的，分散的块。当堆中的空间被多次分配使用又释放之后，会出现数据占用的空间比原来可分配的连续的空间略小的情况，没有被利用的空间太小，又不与其他空闲空间连续，被已经占用的空间隔开，这样的小的不连续的空间会随着程序的运行越来越多，导致虚拟存储空间的速度变慢。因为就需要定期整理存储器中的碎片，挪动数据的位置，使得小的不连续的空间连续在一起。

* + 1. **指针**

指针是一个变量或是一个寄存器。指针变量或寄存器中存储的都为地址，指针可以进行数据或程序的访问、参数的传递等。

* 1. **总线**

总线是用来在各部件之间传递信息的，是数据传输的通道。地址的总线宽度决定着内存的大小，数据总线的宽度决定着计算机的快慢。

1. 计算机的编程语言
   1. **分类**

现有的计算机语言大致可以分为三类：高级语言、汇编语言、机器语言。高级语言是目前程序员使用的最多的，最普及的一类计算机语言，例如常用的C语言，C++，Java，HTML5等，都为高级语言，高级语言更便于理解记忆编写，但是运行速度远远比不上汇编语言和机器语言。

* 1. **机器语言**

机器语言指计算机中的CPU可以直接识别解读运算的一种编程语言。机器语言都为0、1代码。为了安全、可靠、便捷，计算机的加法器只识别在规定范围内的高低两种电平，所以机器可识别的语言只有两种字符，我们以二进制的0和1来代表。早期的计算机输入依靠在纸带上打眼，让计算机读纸带来区分高低电平，后期才出现了与现在接近的输入、输出设备。

* 1. **汇编语言**

汇编语言是由机器语言进化而来的，是一种低级语言，也是符号语言。在汇编语言中，利用一些助记符和十六进制代码代替了原先的0、1代码，使得程序辨识度更高，更容易记忆。汇编语言保持了机器语言的优点，具有直接和简捷的特点，可有效地访问、控制计算机的各种硬件设备。所以汇编语言是相比较其他语言更快速的程序设计语言。

**2.3 高级语言**

高级语言是现在最常用的，最普及的程序设计语言。这类语言利用字符来表示指令，是高度封装过的编程语言，这种语言远离了对硬件的直接控制，所以降低了编程的难度，但同时因为运行时需要通过编译器将其编译成处理器可直接识别的汇编语言，导致高级语言的速度远远不及低级语言。

* 1. **三种语言之间的区别和优缺点**

从速度上来讲，最快的无异于机器可以直接运算的机器语言；汇编语言所操作的对象不是具体的数据,而是寄存器或者存储器，也就是说它是直接和寄存器和存储器打交道，所以汇编语言的执行速度要比其它高级语言快；与其他低级语言相比，高级语言最大的缺点就在于速度慢，在高级语言中，C语言和C++相比较其他的高级语言要快很多，其中最重要的原因在于C语言和C++中都有指针，指针可以直接指向数据或程序所在的存储空间，相当于通过一个变量，间接与存储器建立连接。

从编写的难易上来讲，对于程序员来说，高级语言比汇编语言和机器语言更容易记忆、编写和识别。因为汇编语言是机器语言的符号化、抽象化，而高级语言又是汇编语言的符号化、抽象化。高级语言接近算法语言，易学、易掌握，有已经提供的结构化程序设计的环境和工具，与硬件的直接接触少，程序的可移植性高，因此高级语言更为普遍，更易被接受，反之，汇编语言和机器语言就更不容易编写和使用了。

1. 计算机的工作原理
   1. **进程**

进程是正在运行的程序的一种抽象。一台计算机的CPU在一个时间点只能运行一个程序，既一个时间点只能进行一个进程，但是我们的计算机却可以同时运行多个程序，其原因在于操作系统为我们制造了同时运行多个程序的假象。操作系统制造这种假象的机制叫做：上下文切换。

上下文切换的过程如下图3.1中所示，操作系统将一个很小的时间段又分为很多个更小的时间段。先运行进程A，一小段时间后，并运行一小段操作系统内核代码，使得A运行的结果储存，并读取进程B的上一次的运行结果；再B运行一小段时间后，运行一小段操作系统内核代码，使得B的运行结果保存，读取A上次的运行结果，再运行进程A，以此类推。

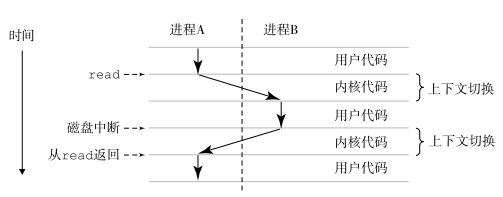


图3.1

但是需要注意的是，鼠标对进程的影响都很大，鼠标产生消息时，会截断对其他进程的运行，来处理鼠标移动或点击时产生的消息。

* 1. **文件**

为了便捷的对各类硬件设备进行管理和运用，引入文件的概念。每一个I/O设备都可被视为一个文件，这样就方便了用户对设备的使用，也大大减小了程序员的工作量。

1. 嵌入式系统

嵌入式系统也是一种完整的计算机系统，它具有与常见PC机所具有的各种部件，常用的为Arduino单片机。以Arduino单片机为例，如下图4，单片机具有相对简单的处理器和存储器，可以处理简单的程序；具有很多的接口作为输入输出的接口，在程序中这些接口的作用可以按照程序员的想法来设计；还有一个接通电源的电源接口。一个真正的嵌入式系统往往还需要配合其他的各类传感器，如下图4.1到图4.5。

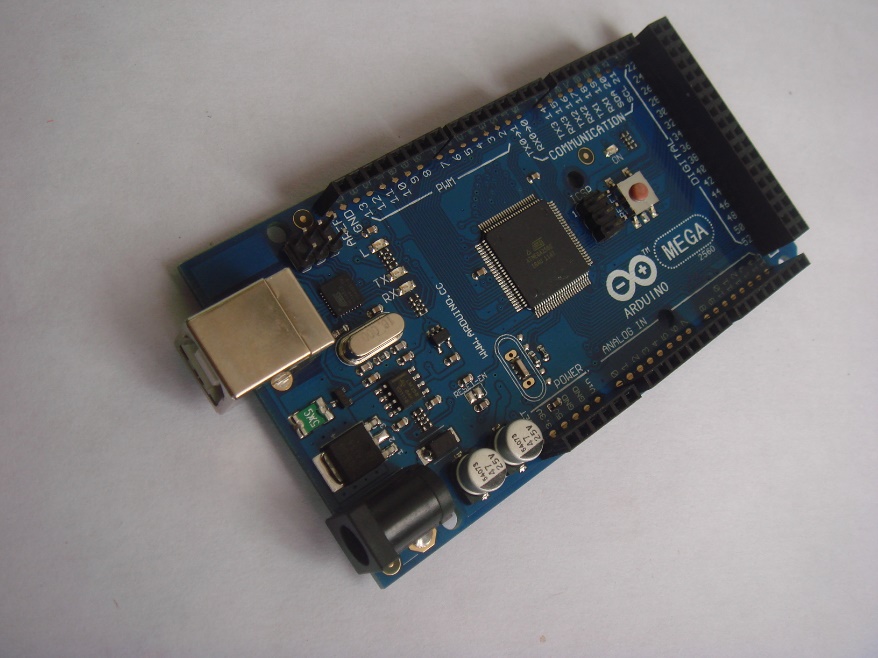


图4

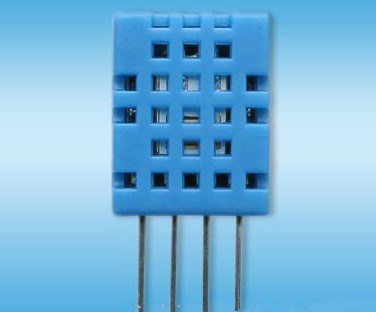


图4.1 数字温湿度传感器



图4.2 红外传感器



图4.3 球面红外传感器



图4.4 弧面远红外传感器



图4.5 舵机

1. 结论

本文描述了计算机系统的基本组成，突出讲解了计算机的存储器系统。通过对计算机硬件系统的叙述，更加贴切，准确的描述了计算机的部分工作原理和计算机的三大类编程语言。最后简单叙述了关于嵌入式系统的基本应用。

[参考文献]

[1]Randal E.Bryant , David R.O`Hallaron . 深入理解计算机系统 . 机械工业出版社，2010

[2]张尧学等．计算机操作系统教程．清华大学出版社，2002

[3] 杨振山，龚沛曾．大学计算机基础（第四版）．北京：高等教育出版社，2004

[4]http://baike.baidu.com/link?url=7NirDk6w0gfImauMT9htpr3JRLOgr4D6fCSBWX4MebNdkO4DsUqC6Asenicu3JPKDmL-eaY7y2bjp\_FjgLp0i747sIb\_ZchfX-z3mHJTPkK

[5]http://baike.baidu.com/item/%E5%AD%98%E5%82%A8%E5%99%A8%E7%BB%93%E6%9E%84

[6] http://baike.baidu.com/view/30741.htm

Github：

账号：ls123456789

[1]指针：ls01、ls02、pointer

[2]数组越界：Array

[3]动态存储分配：malloc

链表：link

[4]嵌入式：Sweep、Blink