

**计算机与信息工程学院**

**题目：深入理解计算机系统**

**学号：20141105072**

**班级：网络编程班**

**姓名：段静**

**计算机与信息工程学院**

**目录**

摘要 ……………………………………………………………………………2

Abstract ………………………………………………………………………2

引言 ……………………………………………………………………………2

1计算机的组成部件……………………………………………………………2

1.1计算机的硬件组成…………………………………………………………2

1.1.1中央处理器………………………………………………………………3

1.1.2存储器……………………………………………………………………4

1.1.3输入设备…………………………………………………………………4

1.1.4输出设备…………………………………………………………………4

1.2计算机的软件组成…………………………………………………………4

2计算机是如何工作的 …………………………………………………………5

2.1计算机简单工作原理…… …………………………………………………5

2.2进制…………………………………………………………………………5

2.2.1二进制… …………………………………………………………………5

2.2.2八进制 ……………………………………………………………………6

2.2.3十六进制 …………………………………………………………………6

2.2.4转换原则 …………………………………………………………………6

3计算机的程序语言 ……………………………………………………………6

3.1机器语言 ……………………………………………………………………6

3.2汇编语言 ……………………………………………………………………6

3.3高级语言 ……………………………………………………………………6

3.4三种语言的优缺点… ………………………………………………………7

4存储器 …………………………………………………………………………7

4.1存储器功能 …………………………………………………………………7

4.2存储器层次结构 ……………………………………………………………7

4.3虚拟内存技术 ………………………………………………………………8

5指针 ……………………………………………………………………………8

5.1指针 …………………………………………………………………………8

5.2地址 …………………………………………………………………………8

5.3链表 …………………………………………………………………………8

6动态存储分配 …………………………………………………………………9

6.1动态分配/撤销内存的运算符new和delete ………………………………9

6.2动态存储分配器………………………………………………………………9

7嵌入式系统 ……………………………………………………………………10

7.1简介 …………………………………………………………………………10

7.2特征 …………………………………………………………………………10

7.3实例与运用 …………………………………………………………………11

参考文献…………………………………………………………………………11

Github……………………………………………………………………………13

**深入理解计算机系统**

作者：段静

摘要：本论文介绍了计算机的硬件及软件组成，计算机是如何工作的，在运行中计算机所使用的不同进制数以及三种不同的程序语言，在编程工作中哪种语言更为有效；存储器类型及需内内存技术；简介了C语言中的灵魂“指针”，介绍了针对数组存储数据缺点的链表；在编程中用到的动态存储分配该如何使用及释放以及在嵌入式系统中所运用的实例。

关键词：程序语言；存储器；动态存储分配

**Thorough understanding of computer systems**

DUAN Jing

**Abstract:**This paper introduces the composition of the hardware and software of computer, computer is how to work, in the operation of the computer by the use of different binary number and three different programming languages, in programming, what kind of language is more effective; types of memory and memory technology; brief introduction of the C language of the soul "pointer", introduced the linked list for the storage of array data shortcomings; used in the programming of dynamic storage allocation an instance of how to use and release and used in embedded systems.

**Key words:** Programming language; memory; dynamic memory allocation

0引言

所有计算机系统都有相似的硬件和软件组件，它们执行着相似的功能。到目前为止，我们依赖于一个简单的计算机系统模型：CPU执行指令，而存储器系统为CPU存放指令和数据。

1. 计算机的组成部件[4]

**1.1计算机的硬件组成**

计算机的硬件组成部分分为中央处理器、存储器、输入、输出。基本功能就是接受计算机程序来实现数据输入、运算、数据输出等一系列根本性操作。

程序 存数 取数

运算器

存储器

输入设备

原始数据 存数

输 程 存 运

入 取 序 取 算

命 数 命 命 命

令 令 令 令

输出设备

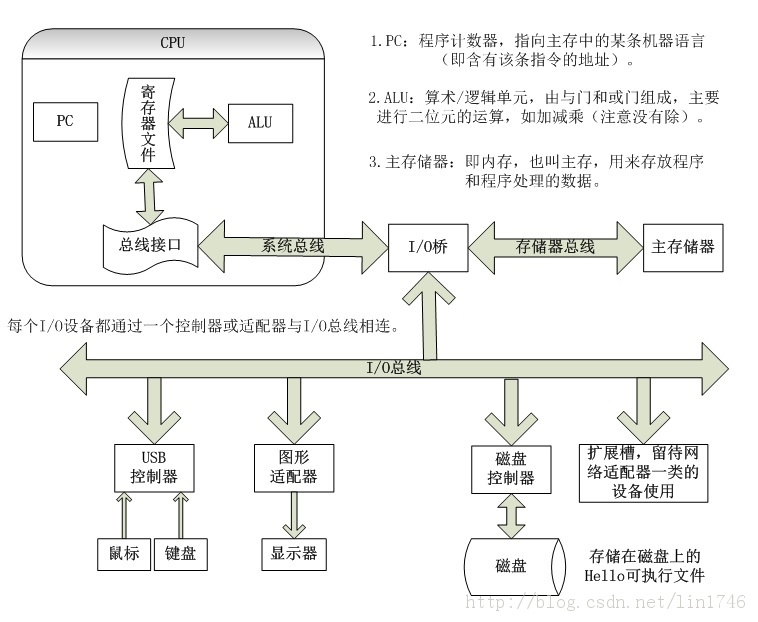
控制器

计算结果

输出命令

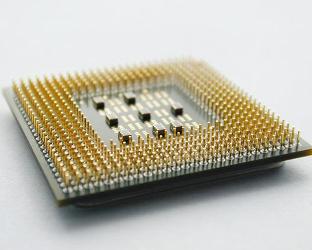
上图为一个计算机系统的基本硬件结构。图中实线代表数据流，虚线代表指令流，计算机各部件之间的联系就是通过这两股信息流动来实现的。

下图为一个典型系统的硬件组成



**1.1.1中央处理器**

中央处理器简称为CPU，是一块超大规模的集成电路。它是计算机系统的核心，包括运算器和控制器，与内部存储器和输入/输出设备合称为电子计算机三大核心部件。

[](http://baike.sogou.com/PicBooklet.v?relateImageGroupIds=&lemmaId=36424&now=&type=1)

CPU包括运算逻辑部件、寄存器部件和控制部件等

运算逻辑部件：可以执行定点或浮点算数运算操作、移位操作以及逻辑操作，也可以执行地址运算和转换。

寄存器：包括通用寄存器、段寄存器和控制寄存器。

控制部件：主要负责对指令译码，并且发出为完成每条指令所要执行的各个操作的控制信号。

计算机所发生的全部动作都受CPU的控制。其中，运算器主要完成各种算术运算和逻辑运算，是对信息加工和处理的部件，由进行运算的运算器件以及用来暂时寄存数据的寄存器、累加器组成。

CPU缓存越大，CPU运行速度越快。

CPU是计算机的心脏，其品质的高低直接决定了计算机系统的档次。CPU能够处理的数据位数是它的一个最重要的指标。

**1.1.2存储器**

存储器是计算机的记忆和存储部件，用来存放信息。对存储器而言，容量越大，存取速度则越快。计算机中的操作，大量的是与存储器交换信息，存储器的工作速度相对于CPU的运算速度要低得多，因此存储器的工作速度是制约计算机运算速度的主要因素之一。

一个存储器包含许多存储单元，每个存储单元可存放一个字节（按字节编址）。每个存储单元都一个编号，即地址，一般用十六进制表示。

1. 内存储器

内存储器（简称内存）又称为主存，它和CPU一起构成了计算机的主机部分。内存储器即主存储器，主存储器越大，计算机速度越快。读取速度介于外存储器和CPU，有电才能保存数据。

1. 外存储器

内存由于技术及价格上的原因，容量有限，不可能容纳所有的系统软件及各种用户程序，因此，计算机系统都要配置外存储器（简称外存）。磁盘即外存储器，平时的数据和程序都存在磁盘里，没电依然能保存数据。

常用外存有磁盘、光盘和磁带，磁盘又可以分为硬盘和软盘等。

**1.1.3输入设备**

输入设备是外界向计算机输入数据和传送信息的装置，是计算机与用户或其他设备通信的桥梁，是用户和计算机系统之间进行信息交换的主要装置之一。最常见的有鼠标、摄像头、扫描仪等。

**1.1.4输出设备**

输出设备是计算机硬件系统的终端设备，用于接收计算机数据的输出显示、打印、声音、控制外围设备操作等。它的作用是将计算机中的数据信息传送到外部媒介，并转化成某种为人们所认识的表示形式。最常用的有显示器和打印机。

**1.2计算机的软件组成**

计算机软件是指计算机系统中的软件及其文档。程序是计算任务的处理对象和处理规则的描述；文档是为了便于了解程序所需的阐述性资料。程序必须装入机器内部才能工作，文档一般是给人看的，不一定装入机器。软件是用户与硬件之间的接口界面，用户主要是通过软件与计算机进行交流。

计算机软件分为系统软件和应用软件，如果把计算机比喻为一个人的话，那么硬件就表示人的身躯，而软件则表示人的思想、灵魂。一台没有安装任何软件的计算机我们把它称之为“裸机”。

1. 计算机是如何工作的

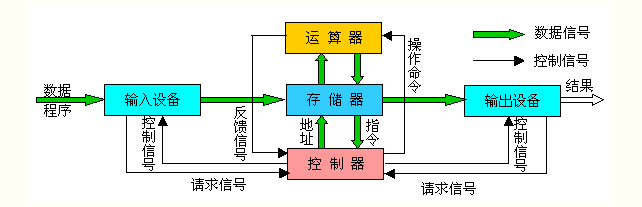
**2.1计算机简单工作原理[1]**

1.操作员通过输入设备将数据和程序送入存储器；

2.通过输入设备发出运行程序的命令；

3.系统接收到运行程序命令后，控制器便从存储器中取出第一条程序指令，进行分析，然后向受控对象发出控制信号，执行该指令；

4.控制器再从存储器中取出下一条指令，进行分析，执行该指令，周而复始重复取指令，分析指令---执行指令过程，直到程序中的全部指令执行完毕。

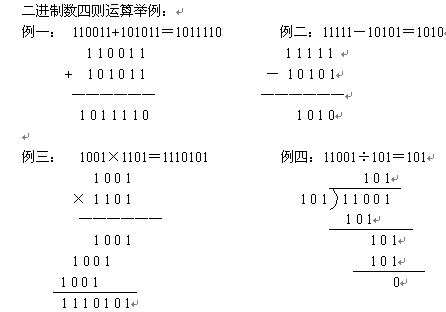


**2.2进制[4]**

**2.2.1二进制**

二进位计数制简称二进制，具有以下特点：

1. 有两个不同的数码符号0、1；
2. 每个数码符号根据它在数中的数位，按“逢二进一”来决定其实际数值。



**2.2.2八进制**

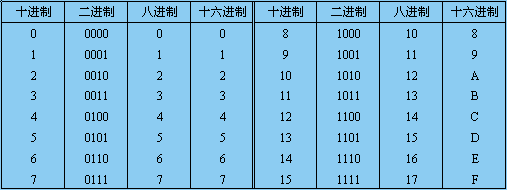
八进位计数制简称八进制，具有以下特点：

1. 有八个不同的数码符号0、1、2、3、4、5、6、7；
2. 每个数码符号根据它在这个数中的数位，按“逢八进一”来决定其实际的数值

**2.2.3十六进制**

十六进位计数制简称为十六进制，具有以下特点：

1. 它有十六个不同的数码符号0、1、2、3、4、5、6、7、8、9、A、B、C、D、E、F。由于数字只有0~9十个，而十六进制要使用十六个数字，所以用A~F六个英文字母分别表示数字10~15；
2. 每个数码符号根据它在这个数中的数位，按“逢十六进一”来决定其实际的数值。



**2.2.4转换原则**

不同进位计数制之间的转换，实质上是基数间的转换。一般转换的原则是：如果两个有理数相等，则两数的整数部分和小数部分一定分别相等。因此，各数制之间进行转换时，通常对整数部分和小数部分分别进行转换，然后将其转换结果合并即可。



1. 计算机的程序语言[5]

**3.1机器语言**

机器语言包含了计算机中CPU（中央处理器）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能够理解的，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。

**3.2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类似于英语单词（称为助记符）的东西所代替，即用助记符代替二进制指令的语言。因此，汇编语言编写的程序比较接近英语，编写和调试相对容易。

**3.3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。从应用角度来看，高级语言可以分为基础语言、结构化语言和专用语言。

从软件工程角度来看，只有在高级语言不能满足设计要求或不具备支持某种特定功能的技术性能（如特殊的输入输出）时，汇编语言才会被使用。

**3.4三种语言的优缺点**

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，指令代码不易记忆，需要用户直接对存储空间进行分配，编程效率极低。

汇编语言同样存在着难学难用、容易出错、维护困难等缺点，但是汇编语言也有自己的优点：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。

高级语言的一个命令可以代替几条、几十条甚至几百条汇编语言的指令，因此，高级语言易学易用、通用性强、应用广泛。

4.存储器[3]

**4.1存储器功能**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 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 |  |
| 只读存储器（ROM） | 读 | 随机寻址 | 数据不丢失 | 工作前写入数据 |
| 闪存 | 读写 | 随机寻址 | 数据不丢失 |  |
| 先进先出存储器（FIFO） | 读写 | 顺序寻址 | 数据丢失 |  |
| 先进后出存储器（FILO） | 读写 | 顺序寻址 | 数据丢失 |  |

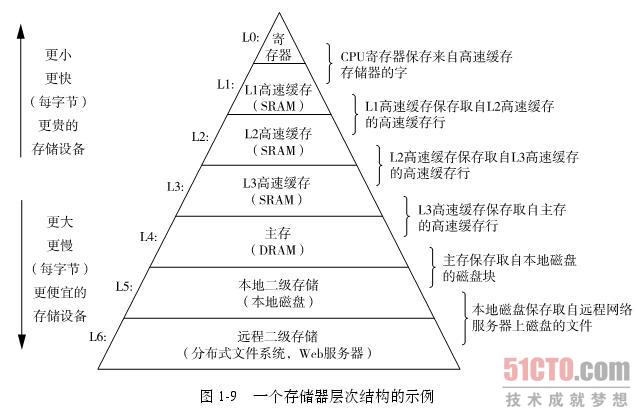
**4.2存储器层次结构**

如果你的程序需要的数据是存储在CPU寄存器中的，那么在指令的执行期间，在零个周期内就能访问到它们。如果存储在高速缓存中，需要1~30个周期。如果存储在主存中，需要50~200个周期。而如果存储在磁盘上，需要大约几千万个周期！

存储器层次结构是可行的，这是因为与下一个更低层次的存储设备相比来说，一个编写良好的程序倾向于更频繁地访问某一个层次上的存储设备。所以，下一层的存储设备可以更慢速一点，也因此可以更大，每个位更便宜。整体效果是一个大的存储器池，其成本与层次结构底层最便宜的存储设备相当，但是却已接近于层次结构顶部存储设备的高速率向程序提供数据。

一般而言，从高层往底层走，存储设备变得更慢、更便宜和更大。

下图为存储器层次结构



**4.3虚拟内存技术**

虚拟内存是计算机系统内存管理的一种技术，是硬盘空间做内存来弥补计算机RAM空间的缺乏。当实际RAM满时（实际上，在RAM满之前），虚拟内存就在硬盘上创建了。当物理内存用完后，虚拟内存管理器选择最近没有用过的，低优先级的内存部分写到交换文件上。这个过程对应用是隐藏的，应用把虚拟内存和实际内存看作是一样的。

虚拟内存〉物理内存

5.指针[2]

**5.1指针**

在IT界普遍认为C语言是集自由性与简练性于一身的编程语言，其中它的自由性的大部分就是通过指针来体现的。因此，人们认为指针是C语言的灵魂。  
 有了指针我们就可以直接访问计算机的内存地址，通过指针可以直接使用指针所指的地址空间的变量或函数。指针赋予C语言强大的编程自由度与计算机底层控制能力。

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。

**5.2地址**

平时程序保存在计算机的外存储器中，当程序需要运行时需要将程序和数据从外存储器中加载到主存储器（内存储器）中，然后CPU会从主存储器中把程序指令或数据逐条取回运行。

内存中的数据必须以某种有规则组织的形式存放。我们可以将内存看成以行列的形式排放的存储单元的有序集合。

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。

**5.3链表**

需要处理大量同类型数据时我们可以使用数组来存储数据，这样可以方便、迅速处理大量数据。可是使用数组有缺点，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存储空间；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，链表可以很好地克服利用数组存储数据的缺陷。

链表的原理其实比较简单，应用数据分别存储在叫做结点的存储块儿中，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。这样我们只要知道某个结点的地址，就可以通过当前这个结点，知道下个结点的地址。因此在一个链表中我们只需要记住第一个结点的地址，通过第一个结点的地址我们就可以访问整个链表的结点。

链表原理如下图所示。

数据4

数据3

数据2

数据1

指针

指针

指针

指针

6.动态存储分配[2]

**6.1动态分配/撤销内存的运算符new和delete**

在软件开发中，常常需要动态地分配和撤销内存空间。在C语言中是利用库函数malloc和free分配和撤销内存空间的。但是使用malloc函数时必须指定需要开辟的内存空间的大小。其调用形式为malloc（size）。size是字节数，需要人们事先求出或用sizeof运算符由系统求出。此外，malloc函数只能从用户处知道应开辟空间的大小而不知道数据的类型，因此无法使其返回的指针指向具体的数据。其返回值一律为void\*类型，必须在程序中进行强制类型转换，才能使其返回的指针指向具体的数据。

C++提供了较简便而功能较强的运算符new和delete来取代malloc和free函数。

new运算符使用的一般格式为：

**new类型[初值]；**

用new分配数组空间时不能制定初值。

delete运算符使用的一般格式为：

**delete[]指针变量**

delete用来将动态分配到的内存空间归还给系统。

new功能相当于一个函数，在内存开辟完空间后，返回这个空间的首地址，此时该地址必须用一个指针保存下来，才不会丢失。用new开辟的内存单元如果程序不“主动”收回，那么这段空间就会一直存在，直到重新开机为止。用new运算符分配的空间，不能在分配空间时进行初始化。

**6.2动态存储器分配器[3]**

虽然可以使用低级的mmap和munmap函数来创建和删除虚拟存储器的区域，但当运行时需要额外虚拟存储器时，用动态存储器分配器更方便，也有更好的移植性。

动态存储器分配器维护着一个进程的虚拟存储器区域，称为堆。我们假设堆是一个请求二进制的区域，它紧接在未初始化的bss区域后开始，并向上生长（向更高的地址）。对于每个进程，内核维护着一个变量brk（读作“break”），它指向堆的顶部。

分配器将堆视为一组不同大小的块（block）的集合来维护。每个块就是一个连续的虚拟存储器片（chunk），要么是已分配的，要么是空闲的。

下图为堆的示意图

用户栈

共享库的存储器映射区域

堆顶

堆

（brk指针）

未初始化的数据（.bss）

已初始化的数据（.data）

程序文件（.text）

7.嵌入式系统[1]

**7.1简介**

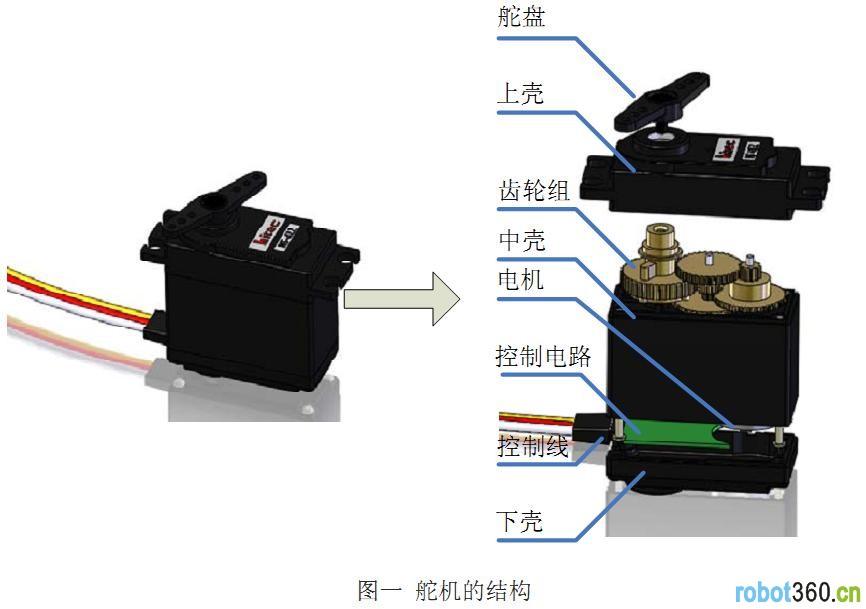
嵌入式系统通常执行的是带有特定要求的预先定义的任务。由于嵌入式系统只针对一项特殊的任务，所以设计人员能够对它进行优化，同时降低成本。由于嵌入式系统通常进行大量生产，所以单个的成本节约，能够随着产量进行成百上千的放大。

**7.2特征**

1. 系统内核小；
2. 专用性强；
3. 系统精简；
4. 高实时性的系统软件（OS）；
5. 多任务的操作系统；
6. 需要开发工具和环境；
7. 与具体应用有机结合在一起
8. 嵌入式系统的软件一般都固化在存储器芯片中。[1]

**7.3实例与运用**

嵌入式系统具有非常广阔的应用前景，其应用领域可以包括：工业控制、交通管理、信息家电、家庭智能管理系统、POS网络及电子商务、环境工程与自然、机器人等。



参考文献

[1]百度百科http://baike.baidu.com/

[2]谭浩强.C++面向对象程序设计（第二版）.清华大学出版社，2014.7

[3] Randal E.Bryant David R.O’Hallaron.深入理解计算机系统.机械工业出版社，2010.11

[4]赵希武，刘东升.大学计算机基础.高等教育出版社，2012.8

[5]张丽萍，孟繁军.C语言程序设计基础教程.清华大学出版社，2014.11

Github

网址：https://github.com/20141105072?tab=repositories

帐号：20141105072

上传代码：

1. array-bounds（数组越界）
2. Pointer-variable(指针)
3. Dynamic(动态存储分配)
4. steering（舵机）
5. car----flash lights（嵌入式系统LED 小灯闪烁）
6. linked-list（链表）
7. sort(排序)
8. Print address（地址打印）