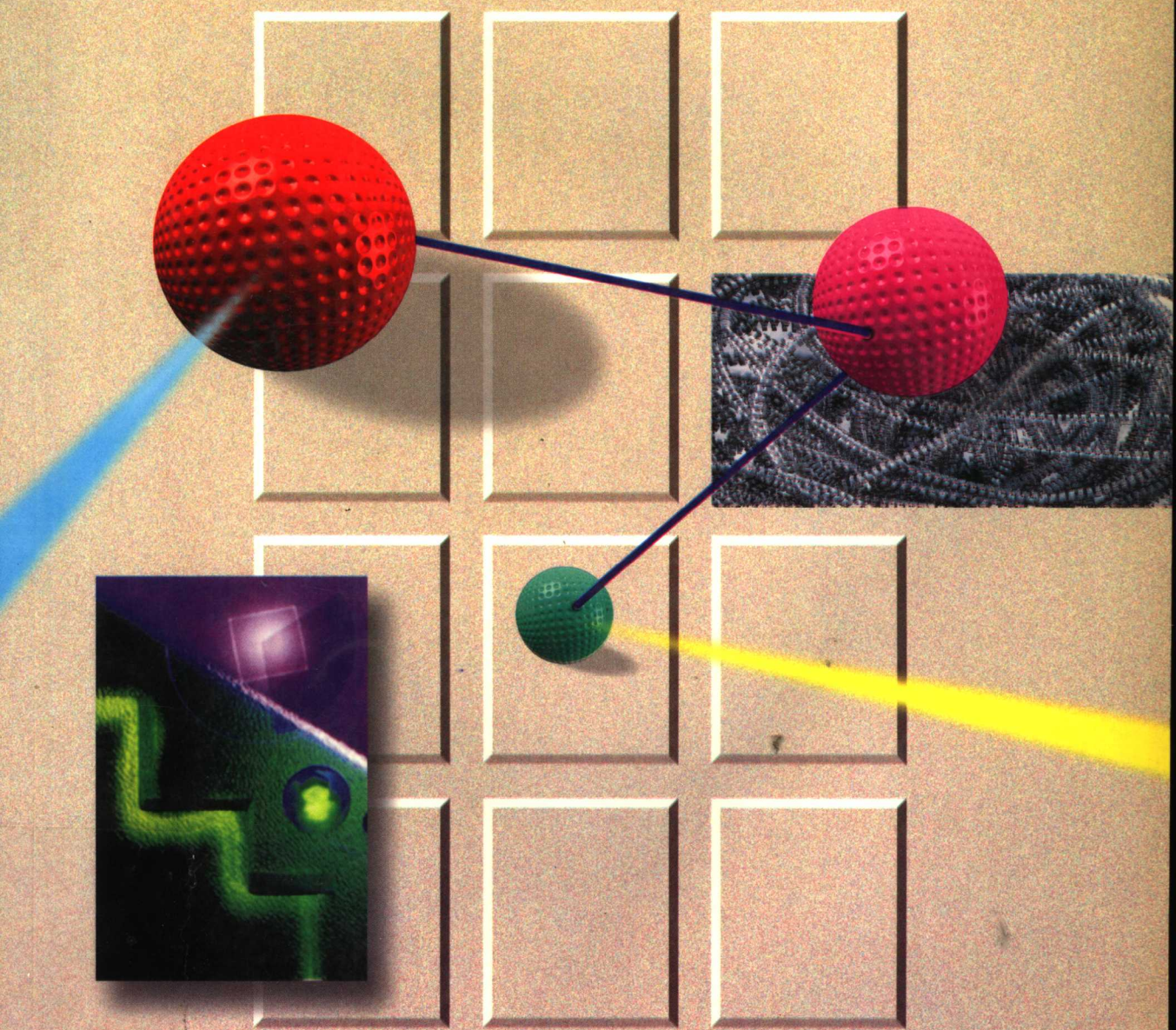
SYSTEM SOFTWARE DESIGN

系统软件设计

王开铸　　编著



哈尔滨工业大学出版社

系统软件设计

王开铸　编著

哈尔滨工业大学出版社
哈尔滨

内容简介

本书是为计算机科学与技术专业的本科生和大专生编写的系统软件设计的教材。前七章内容从概念上、方法上阐述了系统软件的基本功能、基本原理和基本方法，可作为该专业的大专生学习系统软件知识的基本教材；而对于该专业的本科生，可选用本书中第三章(汇编程序)、第四章(反汇编程序)和第八、九、十、十一章作为教材。本书也可作为非计算机专业的学生或科技人员快速了解系统软件知识的自学教材。

系统软件设计

Xitong Ruanjian Sheji

王开铸　编著

*

哈尔滨工业大学出版社出版发行

黑龙江新华印刷厂印刷

*

开本 787×1092　1/16　印张 16.875　字数 385 千字

1999 年 9 月第 1 版　1999 年 9 月第 1 次印刷

印数 1—3 000

ISBN 7-5603-1400-7/TP·129　定价 19.80 元

前　言

通常所说的系统软件大多只是指操作系统和编译系统，因此，大学本科的系统软件课程及其教材，基本上是《操作系统原理》和《编译系统原理》这两本书。但实际上，系统软件所指范围很广。广义地讲，系统软件是指能扩充单机、多机或计算机网络的功能，方便用户使用，为用户编制和使用应用程序而提供的工具软件。例如，连接和装配程序，汇编语言和汇编程序，各种高级语言和对应的编译程序，机器语言级或源程序级的调试程序，单机、多机或计算机网络的操作系统和通讯系统程序，数据库语言，数据库管理系统程序……都是系统软件。狭义地讲，系统软件是指由厂家随计算机出售的软件。可见系统软件涉及的面是很广泛的。

作者在哈尔滨工业大学计算机系曾讲授过《操作系统原理》和《编译系统原理》这两门课程，也有幸在我国国产机上设计过操作系统和编译系统软件。总感到这两本书和具体系统的实现之间有一段差距，这就是编著《系统软件设计》这本教材的初衷。

学习系统软件设计的目的不外乎三个：第一，当我们为了某个目标选择计算机系统时，不仅要选购好的硬件系统，而且要选择相应的软件系统，只有当软硬件系统配备恰当时，计算机系统的效率才能得到最好的发挥；第二，我们能从系统软件中学到很多有用的知识和技术；第三，在原有计算机系统上扩充其功能或设计自己的系统软件时，能给你许多启迪。前二个目的，对大多数技术人员是有用的，而第三个目的，目前只发生在少数技术人员身上。

本书是为大学计算机应用专业的本科生和大专生编写的有关系统软件设计的教材。计算机科学与技术专业的大专生，在培养目标上，强调掌握系统软件中基本的知识、技能、方法和动手能力。而计算机科学与技术专业的本科生，不仅要掌握上述大专生所必须掌握的这些知识，而且还要求对系统软件设计的完整性、系统性和实现的方法与技术有所掌握。本书的前七章内容，强调系统软件的基本功能、基本原理和基本方法，从概念上、方法上去讲清这些问题，可作为计算机科学与技术专业的大专生学习系统软件知识的教材。同时，对计算机科学与技术专业的本科生，可选择前七章中的汇编程序、反汇编程序设计这两章和后四章内容来学习系统软件知识。本书也可作为非计算机专业的学生或科技人员泛读有关内容，快速了解系统软件的入门知识的自学教材。

由于本人知识有限，书中不当之处，在所难免，敬请批评指正。

作者

1999 年 6 月于哈尔滨

目　录

第一章 基 础

本章主要涉及一些计算机术语、硬件和软件发展的概况及操作系统的基本任务。

1.1 系统软件

如今，计算机的应用已渗透到人类社会的各个方面，正在影响着人类的生活方式。日后，计算机将会发展到具有一定的智能，将会影响人类的思维方式。你看，在棋艺方面，计算机已能战胜世界级棋艺大师；多媒体技术、灵境技术和信息高速公路技术将把世界一下子缩短在咫尺之内；一向被认为是人类智慧标志的“能说、会道、看得见”、“学习、理解、会推导”的特征，已成为世界上许多先进国家研究“智能计算机”的攻克目标。确实，计算机基本上是由金属构成的制造物，也就是说，计算机是只能执行一些很特殊的原始指令的制造物。它确实不懂

```
Y = 10
X = 30 * Y
```

这两个句子。然而，当计算机武装了高级语言的编译系统时，计算机就能回答“X 等于 300”了。同样，装有高级语言的编译系统的计算机，确实不懂“当 Y 等于 10 时，30 乘上 Y 等于多少?”这个自然语言的句子。然而，当计算机武装了具有有限自然语言理解能力的智能软件时，计算机就能回答出“X 等于 300”了。

计算机之所以能力很强，就是因为有了系统软件和智能软件。关于智能软件是什么样的，这里不予叙述，而对什么是系统软件？系统软件的功能和设计的基本方法，本书却要涉及一番。

计算机系统包括硬件系统和软件系统。硬件与软件的关系就相当于皮和毛的关系、血液和躯干的关系。硬件相当于躯干和皮，软件相当于血液和毛，软件必须附着在硬件上，同时，硬件与软件必须相互匹配，才能导致计算机系统充分地发挥效能。

系统软件(System Software)是指发挥和扩充硬件系统(单机、多机或计算机网络)的功能，方便用户使用，为用户编制和使用应用程序而提供的一套工具软件。例如连接和装配程序，汇编语言和汇编程序，高级语言和编译程序，目标程序和源程序级的调试程序，单机、多机或计算机网络上的操作系统和通信软件，数据库语言和数据库系统软件……因此，编译程序是系统软件，它接受类似于人的语言的程序，并将它翻译成机器语言的程序。装配程序是系统软件，它把编译好的一个或多个目标程序装入并连接成可执行的机器语言程序。文件系统是允许其他软件能灵活地存储信息和恢复信息，也是系统软件。系统软件是介于人和计算机间的软件，人-系统软件-硬件的关系见图 1.1。

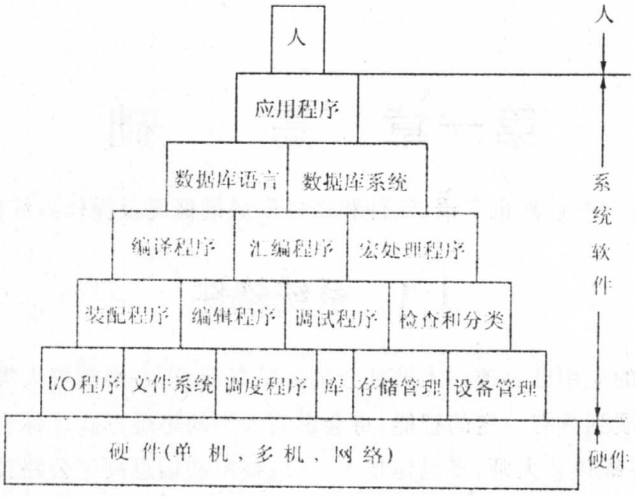


图 1.1 人-系统软件-硬件关系图

1.2 计算抽象模型

问题求解相当于求某个函数 $F(X)$的值,这里的 $F(X)$是问题的描述,X 是问题的初始输入数据 ,F 是问题求解的步骤,$Z = F(X)$是问题所要求的解(输出的结果数据)。X,Z 和 F 均可给予广义的理解。X 和 Z 可以代表数、词句、信息文件等等。而 F 可以是数值计算、信息管理、文字识别、语音识别和自然语言处理等的步骤。为了用一特定计算机求解 $F(X)$,我们必须善于把 F 分解成一串子函数的合成

$$F = F_1 \cdot F_2 \cdot F_3 \cdots F_N$$

这些子函数 F_i 可以用计算机上的指令(命令)来确定。因此,对 $F(X)$求解的程序,可写为下列形式

$$\begin{aligned} Y_1 &= F_1(X) \\ Y_2 &= F_2(Y_1) \\ &\vdots \\ Y_{N-1} &= F_{N-1}(Y_{N-2}) \\ Z &= F_N(Y_{N-1}) \end{aligned}$$

1936 年,英国数学家 Alan Turing(1912 ~ 1954 年)提出计算机抽象模型,称为 Turing 机器。图 1.2 指出了 Turing 机器的组成部分。一条无限长度的存储带,被纵向分成一个个方格。每个方格或是空白,或是有限个符号中的一个。一个有限状态的数字机器,它具有一个读写头,能读出带上任一方格的内容,或改变其内容,并将带从读写头的现行位置上向左或向右移动一个方格。

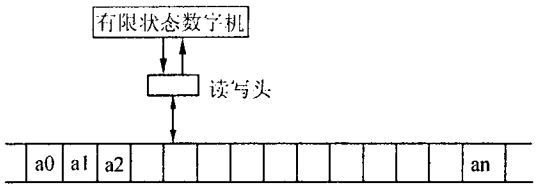


图 1.2 Turing 机器

Turing 机器可以看作有下列组成的有限指令集合。

$$S_h \quad T_i \quad O_j \quad S_k$$

其意义是当有限状态数字机 P 处于 S_h 状态，且符号 T_i 当时处于读写头之下，执行操作 O_j，且将 P 的状态变到 S_k。而操作 O_j 可以是以下四种的任意一种。

(1) $O_j = T_j$，意即将符号 T_j 写到带上(改变原有内容)。

(2) $O_j = R$，意即将读写头右移一个方格(即存储带 M 左移一格)。

(3) $O_j = L$，意即将读写头左移一个方格(即存储带 M 右移一格)。

(4) $O_j = H$，意即停止计算。

按下面的方法，用 Turing 机器可实现计算 $Z = F(X)$。首先，将输入数据 X，以适当的编码形式放在空白带上，然后，启动 Turing 机器，产生要完成的操作序列 $F_1, F_2 \cdots F_N$。在第 N 步操作结束时，机器停止，同时，带里应包含了结果 Z。

1945 年，匈牙利的数学家 John Von Neumann(1903 ~ 1957 年)首次提出一台计算机的设想，又称冯·诺依曼结构计算机模型，把 Turing 机模型具体化了，它有如下的功能。

(1)把要执行的程序和所需的数据送至计算机中；

(2)需要具有长期记忆输入的程序、数据、中间结果及最终结果的能力；

(3)能够完成程序指定的各种算逻运算和数据传送功能；

(4)能够根据运算结果控制程序的走向；

(5)能按人们的要求输出结果。

这三个计算抽象模型是计算机科学的本原问题。程序设计的目的就是问题求解，而求解的物质基础是计算机。

1.3 计算机发展史

计算机发展史，可分为一般意义上的计算机发展史(即传统计算机的发展史)和微型计算机的发展史。传统计算机的进展受到元器件技术改进的极大影响。往往一种新的设计思想必须等到相应的技术跟上以后才能付诸实现。表 1.1 对电子计算机四代不同的主要特征作了总结。请注意，这样按代来划分是有用的，但是划分所根据的标准不是很明确的。目前，世界各国都在研制新一代的计算机，有人将这种计算机归为第五代计算机，这一代的特点是采用更大规模集成电路，是非冯·诺伊曼体系结构、人工神经网络的智能计算机系统。

表 1.1　电子计算机发展的里程碑

代	工　艺	硬件特点	软件特点	代表性计算机	应用领域
第一代 1946 ~ 1954 年	电子管 延迟线存储器	定点 算术运算	机器语言 汇编语言	LAS UNIVAC	数值计算
第二代 1955 ~ 1964 年	分立晶体管 铁淦氧磁芯 磁盘	浮点算术运算 变址寄存器 IO 处理机	高级语言 子程序库 批处理管理 程序	IBM 7094 CDC1604	数值计算 数据处理
第三代 1965 ~ 1974 年	集成电路 (SSI 和 MSI)	微程序设计 流水线 高速缓冲 (Cache)存储器	多道程序设计 多处理 操作系统 虚拟存储器	IBM S/360 DEC PDP - 8	数值计算 数据处理 工业控制
第四代 1975 年至今	LSI 电路 半导体存储器	计算机网络	网络操作 系统	Amdah 1470 Intel 8748	增加计算机网络和综合信息处理 CAD/CAM, AI

大规模集成电路和超大规模集成电路的发展，使微型计算机迅猛发展。微电子技术可使一块芯片上集成上万个器件，这样的芯片就称为微处理器。微型计算机自 70 年代初问世以来，也是经历了四个发展年代。因为微型计算机的核心部件是微处理器，所以人们常以微处理器为依据来表达微型计算机的发展历史。表 1.2 对微型计算机各代主要特征做了总结。

表 1.2　微型计算机发展的里程碑

代	字长	集成度(万/片)	时钟频率(MHz)	运算速度(MIPS)	代表机型
第一代 1971 ~ 1973 年	4 和 8 低档	0.2		< 1	MCS - 4
第二代 1974 ~ 1978 年	8 中档	0.5			Z - 80
第三代 1979 ~ 1980 年	16	6.8	6 ~ 12	< 1	MC 6800
第四代 1981 年至今	32	17 27.5	16 ~ 33	6 ~ 12	MC 68021 Intel 80386

对 32 位微处理器以后的划代，无法再用简单的字长和集成度来加以衡量。事实上，自 Intel 80386 问世后，又推出了许多高性能的 32 位微处理器，如 MC 68030, Intel 80486, MC 68040以及 Intel 的 Pentium 等，其中后三种微处理器的集成度均超过了 100 万元器件/片，时钟频率也达到 25 ~ 300 MHz。

1.4 计算机组织结构进展

计算机发展到如今，不仅是元器件的变化，在体系结构上也发生了巨大的变化。通常，电子计算机是由中央处理机、主存储器、外部设备（包括接口）及总线等部分组成。不同的计算机，总线的设置方法可能是不同的。

1.4.1 以 CPU 为中心的双总线结构

以 CPU 为中心的双总线结构如图 1.3 所示。

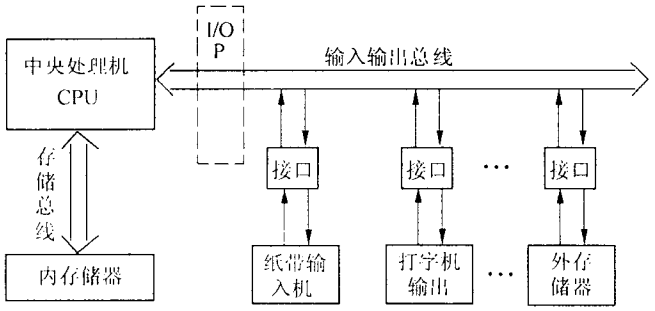


图 1.3 以 CPU 为中心的双总线结构

基本的小型机是用两簇总线来组织系统的，第一簇是中央处理机和主存储器之间交换信息的通路，称为存储总线（简称 M 总线）。第二簇是中央处理机和外部设备之间交换信息的通路，称为输入输出总线（简称 I/O 总线）。主存储器要与外部设备交换信息，必须经过中央处理机。因此，这样的体系结构称为以 CPU 为中心的双总线结构或面向 CPU 的结构。在大中型计算机中增添了 IOP（输入输出处理器），分担 CPU 的功能。

1.4.2 单总线结构

从提高系统结构灵活性出发，某些小型计算机和绝大多数微型计算机采用了以单总线来组织系统的单总线结构，或面向系统的结构，如图 1.4 所示：

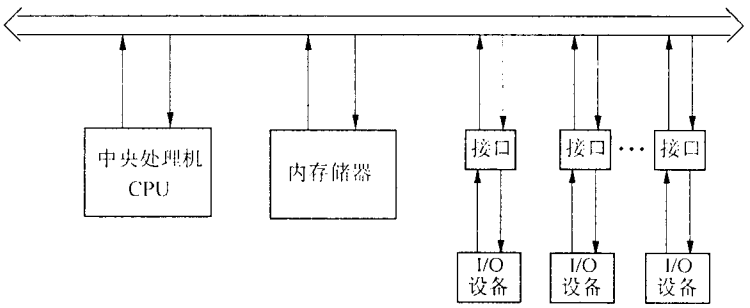


图 1.4 单总线结构系统组织

优点：接口简单，I/O 指令统一；一簇公共信号线实现五大部件间的互联，信息交换方

式相同。

缺点:总线负载很重,当 I/O 任务大时,系统性能下降。

1.4.3 以存储器为中心的双总线结构

为了解决单总线存在的缺点,在单总线结构系统组织的高档机中,在内存和中央处理机之间开辟了一簇高速内存总线,形成了一种新的双总线结构。如图 1.5 所示。

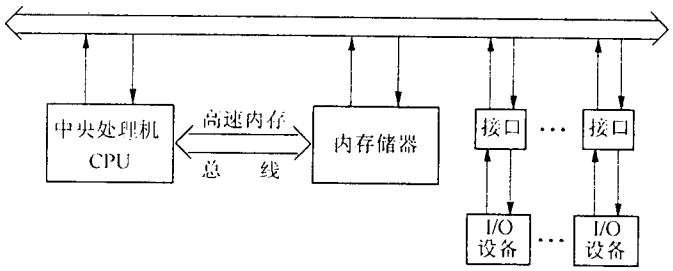


图 1.5 面向存储器的双总线结构

在以存储器为中心的总线结构中,中央处理机和存储器的连接及数据传输,仍由存储器总线实现。这时,输入输出设备能直接和存储器进行传送,而存储器并没有控制传送的机构,所以必须建立所谓通道或传输控制器,由通道或传输控制器来管理输入输出控制。这样,中央处理机就把输入输出控制的大部分工作交给通道或传输控制器去完成,它只执行启动和结束输入输出等少数控制任务,从而使输入输出设备和中央处理机并行工作,提高了中央处理机的效率。

1.4.4 标准总线结构

计算机的发展,几乎使所有设备都可成为它的外部设备,小到一根针,大到一台锅炉。许多厂家生产 OEM(初始设备制造厂家)产品,与计算机的连接也从近距离到达远距离,这就要求建立标准的总线结构。图 1.6 说明了这类总线的典型使用方法。

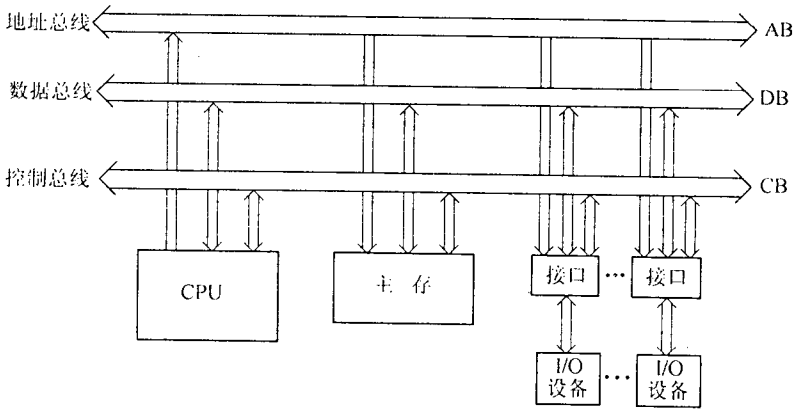


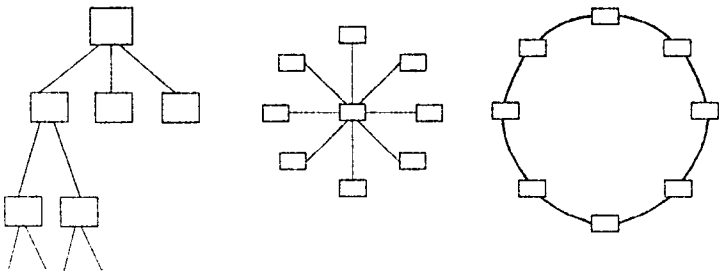
图 1.6 典型的标准总线结构

特别是微型计算机采用了标准总线结构后,不仅可以提高系统的工作效率和处理速

度，简化微机的系统结构，使系统易于扩充，而且可以大大简化系统硬件的设计过程，减轻了软件的设计和调试工作量。

1.4.5 多机系统体系结构

不论是近距离的或是远距离的都采用互联结构使许多计算机互相连接，而被连接的部件可以是处理机、存储器等等。用一个结点表示一个部件，则图 1.7 表示互联的多机结构。



a. 树形结构　　b. 星形结构　　c. 环形结构

图 1.7　多机互联体系结构

1.5 系统软件的演变史

系统软件的演变史是随着计算机的演变和计算机体系结构的变化而变化的。可以这样说，没有硬件谈不上系统软件，当然，系统软件的发展又反过来推进了硬件的发展。

1.5.1 汇编程序

1946 年完成的第一台电子计算机 ENIAC 是美国宾夕法尼亚大学制造的，但是具有冯·诺依曼体系结构的计算机在 1949 ~ 1951 年间批量问世。在这样的计算机上编程序，完全是用二进制代码的机器语言的指令，如

0011 1011 0000 0000 0111

即将存储单元 7 的内容加到累加器中。当时的程序员创造了一种编程序的方法，即用符号编程序，编好后，对符号程序进行内存分配，确定程序、数据和工作单元的内存位置。而后，进行内存代真，即将符号程序翻译成机器语言的程序。这些工作是由程序员本人进行的，如计算 5 + 3⇒Y，用机器语言编写的过程如下。

符号程序设计的程序：

地址	操作	操作数	意义
A1	⇐	〈5〉	将数据 5 放入累加器中
A2	+	〈3〉	取数据 3 与累加器内容相加
A3	⇒	〈Y〉	将累加器结果 8 放入结果单元 Y 中

数据：　　5

　　3

结果单元：　　Y

分配内存：使程序、数据和工作单元(包括结果单元)与内存的绝对地址相对应，即分配内存

内存地址	符号地址
0001	A1
0002	A2
0003	A3
0010	5
0011	3
0012	Y

内存代真：即翻译成机器指令

内存地址	OP 地址
0001	06 0010
0002	01 0011
0003	07 0012
0004	77 0000
⋮	⋮
0010	00 0005
0011	00 0003
0012	00 0000

其中 OP 为操作码，06 表示取指令，07 表示送指令，01 表示加法指令，77 表示停机指令。对代真好的程序进行穿孔、上机调试。到 1951 年时，世界上才出现了第一个称得上是系统软件的汇编程序。

汇编程序是一种翻译程序，它模拟人工用机器语言编程序中的“内存分配”和“内存代真”的全过程。即由软件将汇编语言的源程序自动进行内存分配和代真，并将汇编语言指令翻译成对应的机器语言指令。因此，汇编程序是一种将汇编语言的源程序翻译成用机器语言表示的目标程序。

这时期的系统软件有子程序库(即 10→2，2→10 的翻译程序，三角函数等)和标准程序库(如高斯消去法、龙格－库塔微分方程求解的标准程序)。另有一些服务性的程序，如

双倍字长运算程序，定点机上仿真执行浮点机上的程序。

这一时期的计算机组织结构是以 CPU 为中心的双总线结构。

1.5.2 装配程序

手编机器语言程序是二进制形式的绝对地址程序。所谓绝对地址程序是指目标程序中的程序存放的地址，数据存放的地址和工作区存放的地址，与目标程序在机内的真实地址一致。如何将这样的目标程序放入内存的真实地址内呢？

在第一代计算机上，一般都有一排到几排字长的开关钮，它直接与内存的地址和内容相连。程序员设计了一种既简单又简短的手拨 13 条指令，它是一种装入程序或称引导程序(BOOT)。程序员通过控制台上的开关钮把手拨 13 条引导程序拨入内存地址中，并直接执行这个引导程序。这个引导程序就把用穿孔纸带或卡片表示的目标程序从纸带机上或卡片机上输入到内存，即把目标程序装入到固定的内存。因为装入程序都是把目标程序和数据装入在绝对地址中的，故称为绝对地址装入程序。

已知汇编程序是一种翻译程序，它接受的是用汇编语言编制出的源程序，它输出的是用机器语言表示的功能等价于源程序的目标程序。通常，这样的目标程序存放在外存上，而且目标程序是以浮动地址形式编制的。所谓浮动地址程序是指目标程序中的程序、数据和工作区的地址，是以基址加位移量的形式出现的。一般基址的内容就是程序的首址，这就保证了浮动地址程序可装入在内存的任何可容纳的地方。同样也需要一个装入程序把浮动地址形式的目标程序装入在内存任意指定的地址中，故称为相对地址装入程序，或称为浮动地址装入程序。

发展到第四代计算机上的连接装配程序时，它已具有如下四个功能的程序。

(1)为程序分配内存(分配)；

(2)解决目标重叠之间的符号访问(链接)；

(3)调整所有与地址有关的单元，例如地址常数，以使其对应于分配的空间(重定位)；

(4)将机器指令和数据具体地置入存储器(装入)。

在真实的计算机上的装配程序可能是上述四个功能的某一组合，因此，存在多种功能的装配程序，如“编译并运行”型、绝对型、重定位型、直接链接型、动态装配型和动态链接型的装配程序。

1.5.3 编辑程序

计算机发展到二三代时，人机交互式的 BASIC 语言的出现，键盘输入和 CRT 输出设备的出现，改善了人机关系。这主要取决于编辑程序。编辑程序也是系统软件之一，它是帮助用户输入源程序的程序。因此，文本编辑程序是用来建立和修改 ASCII 码源文件的一个实用程序。它处理的对象主要是字符串，它的主要功能是增、删、改。它的最大特点是人－机交互性强，使用方便。它主要是通过设立一个文本缓冲区，将输入的文本放在这个缓冲区中，实现增、删、改功能。它还要建立一个命令缓冲区，实现接收命令、分析命令和执行各种命令的功能。

1.5.4 调试程序

程序编出来以后，用较短的时间很快调试好一个程序，这是程序员的本领，一般的程序可能存在三种类型的错误：语法错误、语义错误和运行错误。一个程序的语法错误通过编译系统或汇编系统的执行能指出错误位置和性质。而一个程序的语义（逻辑）错误和运行错误，编释和汇编程序就无法查出，一个好的调试程序，就是帮助用户调试目标程序或源程序中的三种类型错误的工具软件，调试程序设计的主要思想是能在程序执行的任意一点上，为程序拍个快照，保持住机器现场和程序现场。机器现场是指计算机在那一点时的各个寄存器的内容；程序现场是指程序执行到那一点时各个变量的内容。用户通过分析这两类内容来断定错误与否，并决定要采取的步骤。因此，联机调试程序，可以在被调试程序中设置断点（拍快照的那一点），观察机器现场和程序现场，分析寄存器内容和变量内容，控制程序的执行、打印或修改等功能。一个调试程序主要由命令解释程序、命令执行程序和各种公用的程序组成。

1.5.5 编译程序

前面讲到的那些软件，是系统软件之一。但是真正称得起是系统软件的要算是编译程序了。1955 年出现了世界上第一个高级语言 FORTRAN 语言。随后，FORTRAN 语言发展壮大，同时也诞生出上百种的各种高级语言，如 BASIC，COBOL，Ada，C，C^{++}，VC^{++} 等。用户想用这些语言编写程序，必须配有相应的编译程序。所以，编译程序也是一种系统软件。它接受的是用高级语言编写的源程序。它输出的是把源程序翻译成功能等价的目标程序。一个编译程序在翻译源程序时，首先对源程序进行句法检查、语法检查，然后进行语义检查。只有当这三方面的检查都通过后，才进行真正的翻译（代码生成），生成目标程序。当前一个好的编译程序能生成两种目标程序，一种是未优化的目标程序，另一种是优化的目标程序。

一种高级语言的编译程序，随着时间的推移，它的功能不断完善，同时，计算机的体系结构发生重大变革时，编译程序会发生变化，如串行的编译程序发展到并行的编译程序，静态分配内存的编译系统发展到动态分配内存的编译系统，单任务的编译系统发展到多任务的编译系统。或者是具有动态分配、多任务的并行编译系统。

由于语言有高低之分，语言之间的翻译便引导出各种翻译程序，如

- 同一机器上的汇编语言到机器语言的翻译，称为汇编程序。
- 不同机器上的汇编语言到机器语言的翻译，称为交叉汇编程序。
- 同一机器上的高级语言到机器语言的翻译，称为编译程序。
- 不同机器上的高级语言到机器语言的翻译，称为交叉编译程序。
- 同一机器（或不同机器）上的高级语言到高级语言的翻译，称为转换程序。
- 同一机器上的机器语言到汇编语言的翻译，称为反汇编程序。
- 同一机器上的机器语言到高级语言的翻译，称为反编译程序。

1.5.6 操作系统

操作系统是与硬件发展密切不可分的系统软件。因此，不同体系的计算机结构就有