



高等学校计算机科学与技术教材

计算机系统理解

COMPUTER Science and Technology

□ 艾丽华 王志海 于双元 编著

- 原理与技术的完美结合
- 教学与科研的最新成果
- 语言精练，实例丰富
- 可操作性强，实用性突出



清华大学出版社

● 北京交通大学出版社

高等学校计算机科学与技术教材

计算机系统理解

艾丽华 王志海 于双元 编著



清华大学出版社
北京交通大学出版社

· 北京 ·

内 容 简 介

本书是北京交通大学教材出版基金资助项目。

全书共分5章,第1章介绍计算机系统软、硬件构成,系统层次和性能评测等;第2章介绍硬件数据表示对高级语言程序数据类型的支持及相关概念;第3章针对高级语言程序转化为机器可执行代码过程,介绍程序预处理、编译、解释、库和链接等技术,以及集成开发环境的配置;第4章围绕程序执行,介绍程序存储映像、指令流水化执行、多核处理器对线程级并行的支持,以及基于OpenMP的并行程序设计思想;第5章关于存储对计算机系统和程序性能的影响,介绍计算机存储资源的层次性,程序空间、内存空间和交换空间概念,内存碎片、内存泄露和垃圾回收管理,以及高级语言程序数据组织和访问优化思想。每章后配有习题,并为任课教师提供课件资源和代码资源。

本书适合作为高等院校计算机科学与技术、软件工程、电子信息工程、物联网等专业方向的本科生教材,也可作为计算机工程技术人员的参考书。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13501256678 13801310933

图书在版编目(CIP)数据

计算机系统理解/艾丽华,王志海,于二元编著. —北京:北京交通大学出版社:清华大学出版社,2018.9

ISBN 978-7-5121-3584-0

I. ①计… II. ①艾…②王…③于… III. ①计算机系统 IV. ①TP303

中国版本图书馆CIP数据核字(2018)第144369号

计算机系统理解

JISUANJI XITONG LIJIE

责任编辑:谭文芳

出版发行:清华大学出版社 邮编:100084 电话:010-62776969 <http://www.tup.com.cn>
北京交通大学出版社 邮编:100044 电话:010-51686414 <http://www.bjtup.com.cn>

印刷者:艺堂印刷(天津)有限公司

经 销:全国新华书店

开 本:185 mm×260 mm 印张:12.75 字数:326千字

版 次:2018年9月第1版 2018年9月第1次印刷

书 号:ISBN 978-7-5121-3584-0/TP303

印 数:1~2 000册 定价:31.00元

本书如有质量问题,请向北京交通大学出版社质监组反映。对您的意见和批评,我们表示欢迎和感谢。

投诉电话:010-51686043, 51686008; 传真:010-62225406; E-mail: press@bjtu.edu.cn。

前 言

在高校人才培养过程中,各个专业呈现出对计算机系统愈发依赖的趋势。应用级程序设计人员为了使工作更加顺利有效,需要对程序的运行环境有一个系统化的理解。本书以软、硬件结合为出发点,展现了一个适于初级编程人员理解的计算机系统。

本书内容包括计算机系统软硬件构成、编程语言使用的数据类型和机器级数据表示、高级语言程序的翻译和链接、程序的并行执行、计算机系统的存储层次和程序的数据访问优化。考虑到计算机软件和硬件的密切关联,本书内容组织的特点是结合程序案例采取软硬件概念捆绑式介绍。

本书内容能够引领学生从硬件层面去理解程序数据表达,弥补学生仅从高级语言编程规则去认识、编写代码的不足;加强学生对于程序集成开发环境配置的理解,使学生在程序构建和调试过程中思路更为流畅;深入理解程序运行时系统软件对内存的维护和管理,处理器和多级存储对于程序快速执行的支持;从而使学生认识一个立体化、软硬件综合的计算机系统,有助于编写更优化的程序代码。

本书共有5章,每章的教学目标如下。

第1章——以桌面台式机主板为案例,学习常用计算机部件特征参数,进一步了解计算机部件间的组织方式;通过学习开机启动到应用完成各阶段的软件类型,使学生理解计算机系统的层次关系;结合计算机系统性能度量参数和第三方测试程序,使学生掌握如何评价计算机系统;通过 Amdahl 定律的学习和基本计算机性能公式的分析,使学生深刻理解计算机系统设计决策的量化分析思想。

第2章——面向程序中的数和符号,介绍计算机使用的数制和字符编码;通过整型和浮点数据表示的介绍,使学生理解程序中整型和浮点变量长度(或位宽)以及能够表达的数值范围;通过捆绑指针、存储器编址以及寄存器三个概念,使学生深入理解指针类型变量的使用以及快速访问指针变量的方法;通过捆绑结构型变量、存储对齐及字节排序三个概念,使学生深入理解变量声明方式对存储空间占用的影响,以及硬件特性对于多字节数据存放排序的影响;通过介绍 SIMD 数据类型,使学生理解现代处理器对于加速多媒体应用中向量运算的支持。

第3章——通过学习程序预处理内容,使学生深入理解程序头文件的作用和条件包含的用途;通过编译、解释及二者结合的翻译方式,使学生了解不同编程语言程序的翻译过程,以 HelloWorld.c 编译产生的目标代码为例,使学生深入理解高级语言语句与汇编和机器指令之间的对应关系;通过学习程序静态、动态链接过程,使学生深入理解集成开发环境关于库指向的配置;结合 HelloWorld.c 的目标代码,使学生理解静态库、动态库、导入库概念。

第4章——结合程序存储映像,使学生综合理解程序执行期间程序代码、全局变量、静态变量、局部变量、malloc 动态分配空间所处的存储位置,以及对于栈和堆的具体化认

识；通过学习指令级并行和线程级并行概念，使学生了解现代处理器并行执行程序的方式；通过学习 OpenMP 并行编程模型，使学生掌握基于共享存储的并程序序设计原理和方法。

第 5 章——通过多级存储资源的学习，使学生深入理解计算机系统的性能也受到存储层次设计的影响；将程序从虚拟空间到内存空间的映射与操作系统对内存空间的划分相结合，使学生充分理解操作系统管理交换空间和合并内存碎片的意义，程序编写时及时释放动态分配内存的重要性，以及程序运行时垃圾回收的作用；使学生掌握如何利用访问局部性改进高级语言（如 C 语言）程序代码的数据访问次序，从而缩短程序执行时间。

本书可作为计算机及相关专业学生计算机系统课程的教材，建议采用 32~64 个教学课时。对于本书读者，最好具有“高级程序设计语言”或“C 语言程序设计”知识背景，以便对计算机系统有更加全面、透彻的理解。

本书每章都配有一定数量的习题。这些习题的安排是对书中知识点应用的补充案例；也有一些习题是对正文知识点的扩展，以便在突出全书主题、保证思路流畅的同时能够为学生建立更加丰富的知识结构；习题中的代码可以直接运行，配合代码设计的问题能够开阔学生编程视野，从而进一步激发学生的兴趣。本书配有电子课件和相关软件资源，可与本书作者或责任编辑联系。

本书的编写得到了北京交通大学教材出版基金资助，也得到了北京交通大学教务处、计算机与信息技术学院领导的大力支持，在此一并表示衷心的感谢！

由于作者水平有限，书中难免有疏漏与不妥之处，还望广大读者和专家批评、指正。

作 者
于北京交通大学
2018 年 1 月

目 录

第1章 计算机系统构成	1
1.1 计算机组件与特征参数	1
1.2 软件	5
1.2.1 BIOS	5
1.2.2 操作系统	6
1.2.3 应用软件栈	7
1.2.4 设备驱动程序	8
1.3 计算机系统层次	8
1.4 计算机系统性能评测	8
1.4.1 性能度量参数与均值	9
1.4.2 评测程序	10
1.5 量化设计原则	16
1.5.1 加速经常性事件的原则	16
1.5.2 Amdahl 定律	17
1.5.3 CPU 性能公式	18
1.5.4 访问局部性	19
1.6 计算机分类	19
习题1	21
第2章 数据类型与数据表示	39
2.1 数制和编码	39
2.1.1 数制和数的机器表示	39
2.1.2 字符编码	40
2.2 整型数表示	45
2.2.1 Intel 体系结构整型数据表示	45
2.2.2 补码	46
2.3 实型数表示	47
2.3.1 Intel 体系结构浮点数据表示	47
2.3.2 移码	48
2.4 指针型变量表示	50
2.4.1 存储器编址	50
2.4.2 寄存器	52
2.5 结构型变量表示	53
2.5.1 存储对齐	54
2.5.2 字节排序	55

2.6 合成式 SIMD 数据类型	57
2.6.1 64 位 SIMD 合成式数据类型	57
2.6.2 128 位 SIMD 合成式数据类型	58
习题 2	60
第 3 章 程序翻译和链接	77
3.1 程序预处理	77
3.1.1 预处理制导类型	77
3.1.2 头文件的作用	80
3.1.3 条件包含的应用	83
3.2 程序翻译	84
3.2.1 编译	84
3.2.2 解释	88
3.2.3 双重翻译	89
3.3 链接	90
3.3.1 库的基本概念	91
3.3.2 静态链接	93
3.3.3 动态链接	97
3.4 集成开发环境的配置	101
3.4.1 关于头文件指向的配置	102
3.4.2 关于库指向的配置	103
3.4.3 关于平台选择的配置	104
习题 3	106
第 4 章 程序的并行执行	116
4.1 程序存储映像	116
4.1.1 代码段	117
4.1.2 数据段	121
4.1.3 栈	124
4.1.4 堆	125
4.2 指令并行执行	126
4.2.1 指令流水线	126
4.2.2 流水线性能	128
4.2.3 流水化并行的瓶颈	129
4.3 线程级并行执行	130
4.3.1 多核处理器	130
4.3.2 多核处理器对多线程同时执行的支持	131
4.3.3 Intel HT 技术	132
4.4 并行程序设计	135
4.4.1 加速 for 循环的并行构造	135
4.4.2 基于共享存储的并行编程模型	136

4.4.3 fork-join 并行执行模型 139

习题 4 141

第 5 章 存储层次与访问优化 155

5.1 存储资源的层次结构 155

5.1.1 存储层次的平均访问时间 156

5.1.2 存储体系的构建基础 156

5.1.3 多级 cache 157

5.2 内存空间 158

5.2.1 内存与程序空间 158

5.2.2 虚拟地址到内存地址的映射 159

5.3 交换空间 160

5.3.1 Linux 对交换空间的管理 162

5.3.2 Windows 对交换空间的管理 163

5.4 内存维护管理 164

5.4.1 内存碎片 164

5.4.2 内存泄露 165

5.4.3 垃圾回收 167

5.5 高级语言程序数据组织与访问的优化 168

5.5.1 数据结构的声明 168

5.5.2 针对访问性能的循环变换 170

习题 5 172

参考文献 196

第 1 章 计算机系统构成

计算机系统由软件和硬件综合而成。本章介绍计算机系统的基本部件及其组织、开机启动程序、操作系统、应用软件栈及设备驱动程序、从用户视角认识计算机系统的层次，以及通过执行测试程序进行的计算机系统性能评测。

1.1 计算机组件与特征参数

我们今天使用的计算机仍然沿用了存储程序（stored-program）的思想。即将计算或者处理过程描述为由许多命令按照一定顺序组成的程序，然后将程序和数据一起输入计算机，计算机对已存入的程序和数据处理后，输出结果。

英国科学家艾伦·图灵（Alan Turing）在 1936 年提出了“图灵机”（Turing machine）的设想。他将抽象的数字计算机描绘为具有无限的存储；读写器可以在存储区域来回移动，并能够读或写；读写器的行为受指令序列控制；程序指令保存在存储区中。这是图灵关于存储程序的思想。

匈牙利科学家冯·诺依曼在 1945 年参与 ENIAC（electronic numerical integrator and computer）项目过程中，进一步强调并发布了存储程序指令的思想。在其之后设计的 EDVAC（electronic discrete variable automatic computer）是第一个采用存储程序思想的计算机。

EDVAC 计算机包括 5 个组件，即中央运算单元（central arithmetic）、中央控制单元（central control）、存储器（memory）、输入（input）/输出（output）设备，如图 1.1 所示。

这是一个以运算器为核心的结构，存储器和输入/输出设备（简称外设）之间的数据交换都要通过运算器，这种单一的通路成为制约系统性能的瓶颈。

目前的计算机在组件连接方面借助系统总线（bus）构成了更灵活的方式，运算器和控制单元都被集成在中央处理器（central processing unit, CPU）中。如图 1.2 所示，这是一个以主存储器为核心的结构，外设可以和主存储器直接交换数据，而不再需要运算器的参与。总线是各个部件之间信息交换的公共通路，并且按照传输信息的类别将总线分组为数据总线（data bus）、地址总线（address bus）和控制总线（control bus）。总线不仅降低了组件之间连接的复杂性，而且有助于推进标准化接口的形成。实际上，中央处理器、主存储器和输入/输出

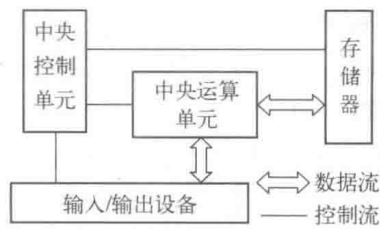


图 1.1 EDVAC 计算机组件

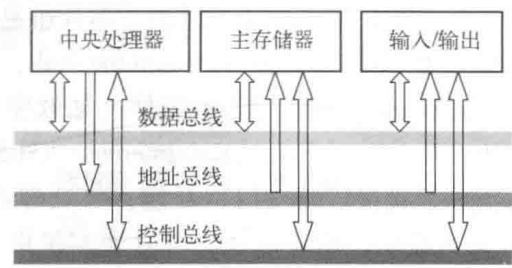


图 1.2 单一总线方式连接各组件

设备在处理数据的速度上差异很大，这种单一总线的连接方式又会产生一种新的瓶颈。为了克服这种新的瓶颈，需要按照组件之间的速度等级设立多级总线（即多总线）连接系统。

下面结合一款 Intel 台式机主板（motherboard）DB85FL 介绍计算机组成部件及其特征参数。DB85FL 主板如图 1.3 所示。在图 1.3 右侧轮廓图中的字母分别代表放置在主板中的器件的位置。

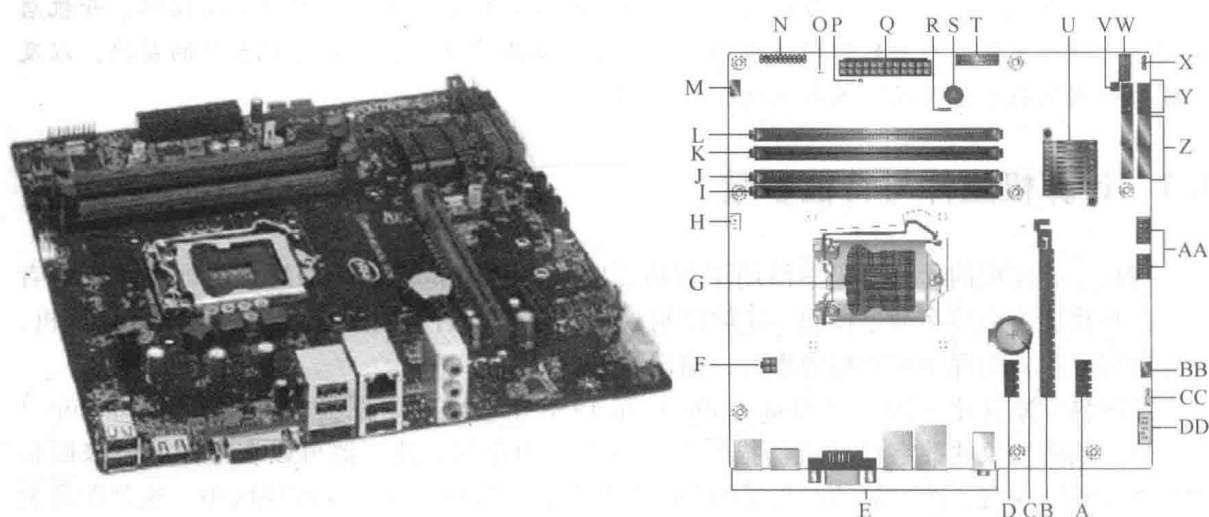


图 1.3 DB85FL 主板

Intel 台式机 DB85FL 主板由以下部件组成。

(1) 中央处理器 CPU（插座）

由于大规模集成电路的发展，芯片制作可以将运算器和控制器集成在一个芯片之内，这就是我们常说的中央处理器（CPU），也称为处理器（processor），它是计算机的核心组件，负责程序的执行。它由完成算术及逻辑运算的运算单元（arithmetic logic unit, ALU）以及控制程序执行的控制单元（control unit, CU）构成。图 1.3 中，线 G 指示的是主板上的 CPU 插座（socket），可插入 Intel 第四代多核处理器如 Intel Core i7-4790s，该处理器主频高达 4.40 GHz，具有 4 个核心（core），可支持 8 线程（thread），处理器内部缓存（Cache）容量为 8 MB。

截至 2016 年第一季度，Intel 已经发布第六代桌面多核处理器系列，例如，Core i7-6700K。第六代处理器可以支持更高速的内存和高性能的显示功能。

(2) 芯片组（Chipset）

图 1.3 中，线 U 指示的是芯片组。芯片组是一个连接主板 I/O 通路的中心控制器，主要包括主板控制集线器（platform controller hub, PCH）。PCH 连接 USB 总线、SATA 总线、LPC（low pin count）传统计算机部件（如软驱、串/并口、等）连接总线、LAN 局域网总线以及 PCI（Peripheral Connecting Interface）外部设备接口。PCH 与处理器通过 DMI（direct media interface, 直接媒体接口）互连，与处理器集成图形加速器通过 FDI（flexible display interface, 柔性显示接口）互连。PCH 还连接集成声卡。这实际上就构成了前面提到的多总线结构！该芯片组可支持高达 1066 MHz 的系统总线。

(3) 主存储器 (插槽)

图 1.3 中, 线 I、J、K、L 指示的是主存储器插槽, 主存储器将插入这些插槽。主存储器保存正在执行的程序及所用的数据。该主板支持 4 个 240 引脚的 DDR3 SDRAM DIMM (Double Data Rate-Synchronous Dynamic Random Access Memory-Dual Inline Memory Module) 插槽, 访问频率可达 1600 MHz, 可支持最大为 4 GB 的内存容量, 用于存储在线处理数据。

DB85FL 主板支持双通道 (dual-channel)、单通道 (single-channel) 和灵活通道 (flex-channel) 三种内存组织形式, 其中双通道内存组织对于实际应用提供的吞吐率最高。在双通道模式下, 该主板上的内存插槽需要成对使用, 即 I 和 K、J 和 L, 或者说同一颜色插槽需要同时使用; 这是典型的双通道内存总线 (dual-channel memory bus), 也称为双通道结构 (dual-channel architecture)。单通道所提供的带宽低于双通道; 使用灵活通道通常混合使用单通道和双通道。

(4) PCI Express×16 显卡连接器

图 1.3 中, 线 B 指示的是 PCI Express×16 显卡连接器, PCI-Express 又可简称为 PCI-E 或 PCIe, 是高速串行总线和接口标准。PCI-Express×16 适于连接高端图形应用的显示部件, 显示信息传输带宽可以达到 32 GBps, 也记为 PCI Express 3.0; PCI-Express×16 也向下支持 16 GBps 传输带宽, 记为 PCI Express 2. x; PCI-Express×16 也支持 8 GBps 传输带宽, 记为 PCI Express 1. x。

(5) 2 个 PCI Express×1 连接器

图 1.3 中, 线 A 和 D 指示的是 2 个 PCI Express×1 连接器, 传输带宽可达 500 MBps, 可以连接较低速的 PCI 总线设备。

(6) 6 个 Serial ATA 接口

图 1.3 中, 线 Z 和 Y 指示的是 6 个 Serial ATA (即 SATA) 接口, 可用于连接硬盘、固态硬盘。其中线 Z 指示的 4 个端口的数据传输率可达 6.0 Gbps; 线 Y 指示的 2 个端口的数据传输率为 3.0 Gbps。

(7) 10/100/1000 网络端口

该主板集成了 Intel 10/100/1000 Mbps 以太网卡。图 1.3 并没有具体标出网络端口, 线 E 指示的背板连接器中包括了 1 个 RJ-45 网络接口, 如图 1.4 所示。

(8) 12 个 USB 端口

该主板可连接 12 个 USB (universal serial bus) 设备。图 1.3 中, 线 E 指示的背板连接器中包括了 4 个 USB 2.0 设备接口, 支持的数据传输速率可达 480 Mbps; 2 个 USB 3.0 设备接口, 支持的数据传输速率可达 5 Gbps。图 1.3 中, 线 T 和 AA 指示的分别是前面板连接器中 2 个 USB 3.0 设备接口和 4 个 USB 2.0 设备接口, 如图 1.5 所示。



图 1.4 RJ-45 网络接口

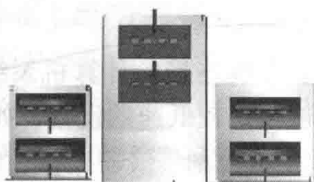


图 1.5 USB 接口

(9) 音频接口

DB85FL 主板集成了 Intel 高保真声音 (high definition audio) 子系统, 支持 8 通道 (5.1+2) 声音效果, 即包括 6 个数模转换通道、2 个独立立体声通道。其中 5.1 环绕声由 6 个数模转换通道支持, 即全频带的左、右、中置、左环绕、右环绕, 再加上一个超低音 (120 Hz 以下) 的声道 (又称做 0.1 声道); 所有这些声道合起来就是所谓的 5.1 环绕声。图 1.3 中, 线 DD 指示的是前面板音频连接器, 用于与主板音频输出端口进行跳线连接; 线 CC 指示的是前面板数字音频输出连接器; 线 E 指示的背板连接器中包括了图 1.6 所示的模拟音频输出端口。

(10) 主板电池

图 1.3 中, C 是主板上的电池, 为实时时钟和 CMOS 存储器 (保存基本输入/输出系统) 供电, 如图 1.7 所示。计算机接通电源时, 电池寿命得以延长。如果计算机不接通电源, 主板电池约有三年的寿命, 可以更换。

(11) 散热风扇连接器

图 1.3 中, H 是处理器风扇接口, M 和 BB 分别是机箱前、后部的风扇接口, 如图 1.8 所示。处理器风扇一定要与处理器风扇接口连接, 不能与机箱风扇接口相连。



图 1.6 模拟音频输出端口



图 1.7 电池

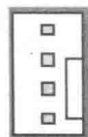


图 1.8 风扇接口

(12) 电源及其他连接器

图 1.3 中, Q 是主电源连接器, F 是处理器电源连接器, W 和 E 分别是前后面板连接器。X、O、P 分别是电源指示灯、电源故障指示灯、待机状态指示灯。该主板提供 +5V 待机电源信号, 服务于网络唤醒等功能。

(13) 可信平台模块 (TPM) 连接器

图 1.3 中, N 是 TPM (trusted platform module) 连接器。TPM 是一个硬件模块, 用来确保开机启动过程的安全性。TPM 模块直接连在主板上, 提供密钥和口令等的安全存储。

实际上, Intel 台式机主板 DB85FL 由芯片组将 CPU、主存储器以及各类输入/输出设备连接在一起, 如图 1.9 所示。

显然, 这是一个多总线连接的计算机系统, 并且针对不同速度的组件进行了层次化连接。主板控制集线器连接比较低速的设备, 如硬盘、USB 设备、网卡、声卡等。DMI 以点对点互连方式将 CPU 和主板控制集线器连接; FDI 将高清图形控制器 (集成在 CPU 中) 与主板控制集线器连接。

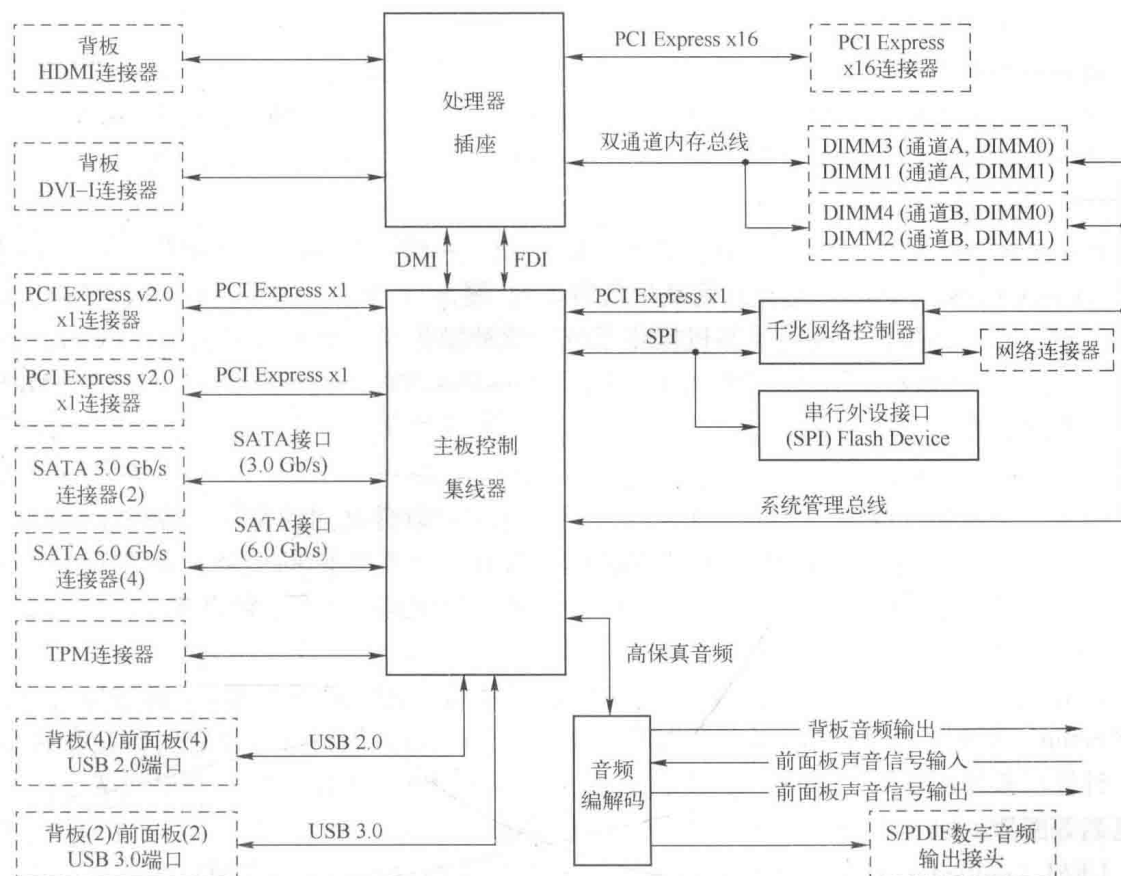


图 1.9 DB85FL 组件连接

1.2 软件

计算机需要软件来指挥控制其工作。完成此类工作的基础软件是基本输入/出系统和操作系统。基本输入/出系统保存在主板的只读存储芯片内，而操作系统通常保存在硬盘上。如果让计算机完成某类指定的功能，还必须在操作系统基础之上安装相应的软件，也将一组实现某个应用目标的软件称为软件栈（software stack），或者应用软件栈。

1.2.1 BIOS

基本输入/出系统 BIOS（basic input/output system）是一个底层软件（low level software），也是计算机系统加电后第一个运行的软件，用于计算机系统初始启动。BIOS 是小程序，占用的存储空间较小。Intel DB85FL 主板上的 BIOS 保存在 SPI Flash 中。BIOS 指令用于加载基本的计算机硬件，以及加电自测试即 POST（power on self test），因此 BIOS 也称为固件（firmware）。如果在开启的时候没有通过 POST 测试，计算机也会发出鸣响声音，用于指示该机目前的错误。POST 测试之后，BIOS 将加载操作系统（operating system, OS），并将控制交给操作系统。

BIOS 包括下述 4 个主要功能。

(1) POST

计算机开机即会跳转到 BIOS 的位置进行硬件检测,如处理器、内存、显示器、键盘等的测试。如果存在问题,例如内存没有插好,测试不通过就会有一些声音提示。通过测试之后,即将控制转交给启动加载器 (bootstrap loader)。

(2) Bootstrap Loader

Bootstrap loader 根据 BIOS 配置信息确定启动盘,并将控制转交给启动盘的主引导记录 (master boot record, MBR, 这里认为从硬盘启动)。硬盘启动扇区中的 bootloader 进一步加载操作系统,然后将控制权交给计算机操作系统。每种操作系统都有自己的 bootloader,例如 Windows 10/8/7/Vista 操作系统的 BOOTMGR、Windows NT/2000/XP 操作系统的 NTLDR、Ubuntu 操作系统的 GRUB 等。

(3) BIOS drivers

设备驱动程序 (device driver) 如同设备和使用它的软件之间的翻译,提供了计算机系统中硬件与软件的连接。每种设备都需要驱动程序,并且不同的设备驱动程序也不同。BIOS drivers 只包括对于计算机基本硬件的控制驱动,即键盘、内存、硬盘等。

(4) BIOS Setup

计算机开机时刻,如果按下 Del 键 (具体参考主板厂商的用户手册),即进入 BIOS 设置 (Setup) 页面。BIOS Setup 为用户提供了硬件设置接口,用户可以设置引导盘的启动次序、计算机系统的口令、时间、日期等信息。有些高级 BIOS 还提供 CPU 频率或者电压、内存延迟等配置。BIOS 配置信息保存在 CMOS 存储器中。

UEFI (unified extensible firmware interface) 是一个新的规范,与 BIOS 的作用相当,解决了 BIOS 对硬盘容量和分区数量的限制,能够加快系统的启动和关闭时间,被预计取代 BIOS。目前的操作系统和新款 PC 已经提供了对 UEFI 的支持!

1.2.2 操作系统

操作系统是计算机系统最重要的系统级软件,其本身是用于管理计算机硬件资源并且为应用软件提供服务的一组程序。操作系统属于中间层软件 (middle level software),通过驱动程序 (driver) 和 BIOS 与硬件设备交互。操作系统管理硬件和软件资源,涉及内存分配与控制、辨识外部设备输入、将输出传给计算机显示、管理硬盘文件,以及控制外部设备,如打印机和扫描仪等。一个大型计算机的操作系统还要负责监视不同的程序和用户,确保每个程序平稳运行、互不干扰,并且防止未授权用户访问计算机系统。

操作系统有多用户 (multiuser)、多处理 (multiprocessing)、多任务 (multitasking)、多线程 (multithreading) 及实时 (real-time) 之分。一个多用户操作系统能够使得多个用户同时运行程序,此类操作系统可能支持几个、成百甚至是上千用户同时运行程序。多处理操作系统允许一个程序在多个处理器上运行,亦即一个系统使用多个 CPU,并且共享主存储器资源。多任务操作系统允许同期运行多个程序,可以有一个 CPU 采用分时方式支持,也可以由多个 CPU 提供同时支持;多线程操作系统允许一个程序的不同部分同时运行,为并行程序设计提供进一步支持。实时操作系统能使计算机即时处理和响应输入请求。一般来说,通用操作系统不强调实时性。

当前的操作系统趋向于图形化用户界面,利用鼠标、触摸屏等点击设备进行输入。常用

操作系统包括 Microsoft Windows, MacOS X 和 Linux。

Windows 是微软公司的操作系统，主要定位于个人用户，当然也有服务器版本。微软操作系统经历了 MS-DOS、Windows 1.0、Windows 2.0、Windows 3.0、Windows 3.1、Windows 95、Windows 98、Windows ME、Windows NT 3.1-4.0 (Server)、Windows 2000 (商用桌面和服务器版)、Windows XP、Windows Vista、Windows 7、Windows 8 以及 Windows 10。操作系统的不断更新为其上运行的软件兼容性提出了很高的要求。

MacOS X 是 Apple 公司的操作系统，它以出色的图形用户界面 (graphical user interface, GUI) 著称。MacOS 的定位目标是为非技术类用户提供客户端操作系统。这意味着最终用户可以对界面复杂性的容忍度很低，但是 MacOS 的设计者却要成为最擅长设计简洁界面的技术牛人。

与其他操作系统不同，Linux 是免费、开源的操作系统，支持个人用户和服务器用途。Linux 有多种版本，如 Ubuntu、CentOS、Debian、openSUSE 和 Fedora 等。尽管 Linux 用户和开发者都已经适应了它的命令行界面 (command line interfaces, CLIs)，与商用 UNIX 操作系统相比，近年来的 Linux 版本也不断在 GUIs 和 GUI 工具方面有了很大的改观。

大多数应用程序都是针对某种操作系统而设计的，所以应用软件通常要指明它所赖以支持的操作系统。例如一个视频播放软件会要求它在什么操作系统下运行。软件开发者也通常会发布一些针对不同操作系统的应用软件。

1.2.3 应用软件栈

我们日常所使用的浏览器 (browser)、媒体播放器、办公系列化软件都是应用软件，也称为高层软件 (high level software)，它们都通过操作系统使用硬件资源。

软件栈是指为了实现某种功能所需的一套软件子系统或组件。在强调某种用途时，也会称其为应用软件栈 (application stack)。例如，一个典型的企业应用软件栈包括基本的办公功能 (如文字处理软件、表单、数据库)，Web 浏览器，电子邮件，以及即时消息。选择什么样的软件栈取决于项目的应用需求。

目前比较流行的应用软件栈有：LAMP 栈、WISA 栈、MAMP 栈及 Java Web 应用栈。LAMP 栈是一个流行的开源 Web 平台，一般用于运行动态 Web 站点，LAMP 软件栈包括 Linux 操作系统、Apache Web 服务器软件、MySQL 数据库、PHP 服务器端编程语言。WISA 软件栈是一个基于 Windows 操作系统的 Web 应用平台，使用微软公司的 Web 服务器软件 IIS，SQL Server 数据库，以及 ASP.NET 开发框架。MAMP 栈也用于构建 Web 服务应用，它包括 MacOS X 操作系统、Apache Web 服务器软件、MySQL 数据库、PHP 服务器端编程语言。Java Web 应用栈可以采用 Linux 或者 Solaris 操作系统、Tomcat 应用服务器软件、MySQL 数据库、JSP 编程技术。上面列举的软件栈都在服务器端，包括现成的软件产品 (如操作系统、数据库、Web 服务) 和编程语言，服务器端编程语言通常有 PHP，Python，Java，和 .NET 框架所支持的语言等；数据库也有若干选择，如 Oracle，MySQL，SQL Server 等。

之所以称其为软件栈，是因为这组软件之间存在相互支持的关系以及需要遵照一定的安装次序，以 LAMP 软件栈为例，其安装次序和依赖关系如图 1.10 所示。

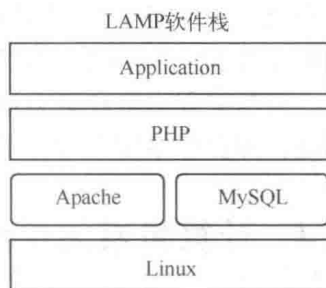


图 1.10 LAMP 软件栈

图 1.10 表示在建立 LAMP 软件栈时,首先要安装 Linux 操作系统;然后是 Apache Web 服务器和 MySQL 数据库的安装,两者之间没有先后次序限制,但是具体的软件版本(例如 32 位版本还是 64 位版本)与底层的 Linux 版本有关;最后安装应用软件编程语言 PHP,并加载整个应用。PHP 应用代码由 Web Server 解释执行,读写数据库中的信息,并返回结果页面。

1.2.4 设备驱动程序

设备驱动程序(device driver)是挂接在计算机上的某种设备的控制程序。设备驱动程序为硬件设备配备了一个软件接口,使得操作系统和应用程序能够使用该设备功能,而不必过度关注其硬件细节,如图 1.11 所示。



图 1.11 设备驱动层

设备驱动程序允许操作系统和应用程序与系统硬件交互。没有设备驱动程序,计算机就不能与这个设备进行数据通信。例如,我们将一个新型号的打印机或者扫描仪等设备连入计算机时,就需要安装它的驱动程序。不过,大多数情况下操作系统都能够自动检测与判断设备型号,并自动为其选择适合的驱动程序,即我们常说的即插即用(plug and play)设备,这表明设备驱动程序也是操作系统的一部分。只是有些设备厂商单独提供了驱动程序,

并且这些设备驱动程序没有被捆绑在现有的操作系统中,需要用户在使用的操作系统上额外安装该设备驱动程序。

1.3 计算机系统层次

计算机系统可以简单地表示为三层结构,即计算机硬件、系统程序和应用程序,如图 1.12 所示。

计算机系统最内层核心是硬件,最外层是应用软件。系统程序也称为系统软件,为应用程序提供具体编程环境和硬件资源使用环境。操作系统是一个主要的系统程序,起到了应用软



图 1.12 计算机系统层次

件与硬件资源之间的桥梁作用。集成开发工具作为系统软件为程序的编辑、翻译、调试等提供了支持。

处于硬件之上的操作系统,一方面保护硬件免于被应用程序误用,另一方面为应用程序提供了一个简洁而统一的硬件资源抽象,从而避免繁杂的硬件管理。要运行一个 64 位的操作系统,底层需要 64 位处理器支持;要运行一个 64 位的应用程序,就需要有一个 64 位的操作系统和一个 64 位的处理器支持。

虽然编程语言在努力地独立于底层平台,但是应用程序的开发仍然需要考虑将来的运行环境,即操作系统和硬件平台。

1.4 计算机系统性能评测

无论对于计算机厂商还是用户,计算机系统性能评测都是非常重要的。计算机组件的特

征参数可以作为评测的一个指标,但对系统的综合测试还是要通过测试程序进行。通过执行测试程序,可以获得反映系统性能的度量参数,进而对系统做出综合评价。

1.4.1 性能度量参数与均值

1. 参数

一些评测程序通常使用如下的性能度量参数。

(1) 程序执行时间

程序执行时间是指程序运行所花费的时间。该参数主要用于度量计算机系统自身性能。

(2) 响应时间

响应时间是指从用户向系统提交任务到系统完成任务所需的时间,这是计算机系统用户关心的参数。实际上,当只有一个用户提交一个任务时,响应时间就是程序执行时间。

(3) 吞吐率

吞吐率是指单位时间内完成的任务量,这是大规模数据处理中心系统管理员关心的参数。

(4) MIPS

MIPS (million instructions per second) 每秒执行的百万条指令数,反映 CPU 的性能。

(5) FLOPS/MFLOPS

FLOPS/MFLOPS (floating-point operations per second/mega FLOPS) 每秒执行的浮点操作数量/每秒执行的百万条浮点操作数量,反映 CPU 的浮点运算性能。

MIPS 与 FLOPS 常被硬件厂商使用,反映其处理器的运算能力或更进一步的浮点运算能力。这两个参数属于吞吐率类别的参数。

2. 均值

对系统进行多次测试后,需要将每次得到的测试结果进行综合,以便给出简洁的评测结果。下面给出常用的求均值的方法。

(1) 算术平均值 (arithmetic mean)

如果每次的测试结果为 t_i (即程序的执行时间,以下的 t_i 含义同此),则 n 次测试的算术平均值为

$$\frac{1}{n} \sum_{i=1}^n t_i$$

算术平均不适用于彼此相差很大的数据,这样反映的均值是不准确的。

(2) 加权算术平均值 (weighted arithmetic mean)

如果每个测试结果为 t_i ,而这个测试在整个测试中的权重是 w_i ,则 n 个测试的加权算术平均值为

$$\sum_{i=1}^n w_i t_i, \text{ 并且 } \sum_{i=1}^n w_i = 1$$

权重是一个相对的概念,是针对某一指标而言。某一指标的权重是指该指标在整体评价中的相对重要程度。当前性能评测中的指标是指测试程序的执行时间。当使用多个不同的程序来评测系统时,应该考虑系统日常应用所处理的程序类别的权重,这会给出一个比较合理的均值。