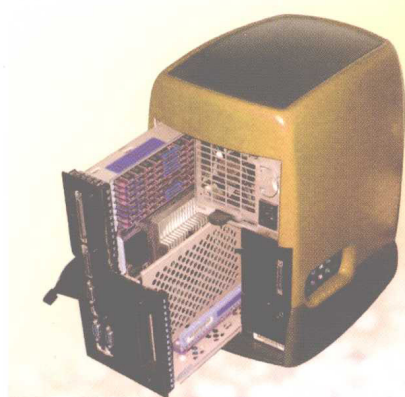




UNIX和 计算机软件技术基础

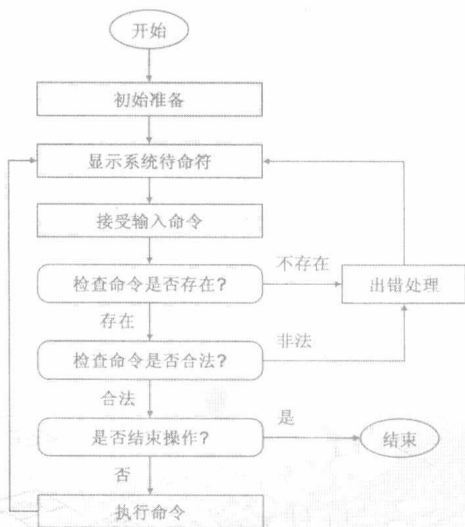
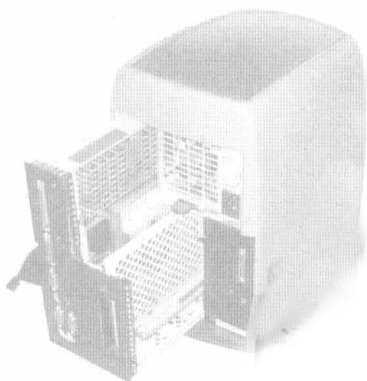
赵文庆 编 著





UNIX和 计算机软件技术基础

赵文庆 编 著



UNIX

图书在版编目(CIP)数据

UNIX 和计算机软件技术基础/赵文庆编著. —上海:复旦大学出版社,2011.5
(复旦博学·微电子学系列)
ISBN 978-7-309-08175-6

I. U… II. 赵… III. ①UNIX 操作系统②软件 IV. TP31

中国版本图书馆 CIP 数据核字(2011)第 106233 号

UNIX 和计算机软件技术基础

赵文庆 编著
责任编辑/张志军

复旦大学出版社有限公司出版发行
上海市国权路 579 号 邮编:200433
网址:fupnet@fudanpress.com <http://www.fudanpress.com>
门市零售:86-21-65642857 团体订购:86-21-65118853
外埠邮购:86-21-65109143
上海第二教育学院印刷厂

开本 787×1092 1/16 印张 21.5 字数 472 千
2011 年 5 月第 1 版第 1 次印刷

ISBN 978-7-309-08175-6/T·418
定价:42.00 元

如有印装质量问题,请向复旦大学出版社有限公司发行部调换。
版权所有 侵权必究

内 容 提 要

本书介绍计算机软件的基础知识和应用技术。全书共分四章。第一章以UNIX/Linux为例，简要地叙述了计算机操作系统的基本概念和基本原理，介绍了操作系统的各个功能模块，以及UNIX/Linux中B shell和C shell的用法。第二章介绍了UNIX/Linux中与文本编辑和字符处理有关的常用工具，如vi，ex，sed和awk，用于软件维护和管理工具make和SVN，以及编译器生成程序lex和yacc。第三章介绍了数据与存储单元的关系和构造性数据类型等C语言的一些重要特征以及编程技术。第四章介绍数据结构的基本内容、基本操作和应用，包括线性表与树的概念和实现、图的基本定义等概念。

本书没有采用计算机专业所需的完整的理论知识，而是从一个非计算机专业的软件工作者和计算机应用软件开发者的视角来叙述和介绍计算机软件的基础知识与应用技术，使得该书非常具有实用性，特别适合于电子类软件工程师的需求。

本书可作为高等院校非计算机专业本科生的教学参考书，也可供从事信息技术等领域涉及软件开发和软件应用工作的科技人员阅读参考。

序

计算机软件是一个庞大的产业,已深透到各行各业。当今整个社会的运转均离不开软件产品,它影响着信息产业、高端制造业、宇航产业等的发展,是国民经济发展的重要支持力量。软件产业的发展需要大量人才,因此需要很好的教材和教学参考用书。

本书作者赵文庆教授长期从事电子设计自动化(EDA)的科研和教学工作,长期为大学本科生和研究生开设软件课程,积累了丰富的经验。在此基础上,于 1993 年编写了《计算机软件技术基础》一书。该书经过长期使用,效果很好。现今作者修改和扩充了该书的内容,编写了本书《UNIX 和计算机软件技术基础》。

本书重点介绍 UNIX/Linux 操作系统的各个功能模块及有关工具的使用、C 语言的基本内容与编程方法和编程中所需的数据结构技术。虽然本书的内容是普适性的,对任何一个专业软件的开发者均可参考,但是本书中的很多例子是结合集成电路设计中的问题来说明的,因此本书特别适用于电子设计自动化领域的读者。

目前,在信息技术领域的很多电子产品都是软硬件结合的,因此很多学生需要软件方面的基础知识。不像过去的电子产品只是硬件,设计硬件所需要的工具均由软件公司提供,设计者只需要会使用这些软件工具就能够完成设计任务。现今软硬件结合的产品,即产品本身就有软件,这些软件可能就需要由设计者自行设计和完成编程。对于大部分信息领域的电子产品设计者来说,仅仅使用软件工具开发产品已不能完成设计任务,还需要自己动手设计和开发软件。大部分学生必须具备开发软件的知识 and 能力,因此本书的读者面很广。

我认为本书对软件开发者(特别是 EDA 的软件开发者)是一本很好的参考书,并可作为信息学科有关专业的大学生或研究生的教材或教学参考书,特向读者推荐。

唐璞山
2011 年 3 月

前 言

随着包括计算机软件技术在内的信息技术的高度发展, 计算机软件开发所依托的计算机技术和软硬件平台在不断更新。30 多年来, 无论在科研工作还是在教学工作中, 从中学到大学, 计算机知识的普及以及学生拥有的计算机资源正在大幅度提高, 非计算机专业的工程技术人员对计算机软件知识和技术的需求也在大幅度提高。修读该课程的学生或者参加本领域科研工作的师生, 无论在学术界还是企业界, 纷纷为我们提供了许多有益的反馈意见, 推动我们不断更新自己的知识结构, 使得该课程的内容也一直在不断刷新。

作为教学参考书的《计算机软件技术基础》出书多年, 现在每年都需要附加一份不薄的补充教材, 令本课程的教学逊色。大学教学对一线教师的要求以及切身感受, 使作者觉得这是一种使命感, 应该为我们的学生以及读者提供知识和技术的食粮。但是, 如果只是再版此书, 已经不能够满足实际的需要。因此, 在复旦大学微电子学系领导和同事们的促进和推动下, 作者编写了作为更新版的《UNIX 和计算机软件技术基础》。

本书着重介绍涉及计算机软件技术的一些基础课题并强调实用性, 包括 3 个主要部分: UNIX 操作系统的原理、应用和编程介绍, C 语言编程进阶和规范, 数据结构基础。Linux 作为类 UNIX 的操作系统, 已经得到广泛使用。因此, 本书所称的 UNIX 操作系统, 实际是指 UNIX/Linux 操作系统, 敬请读者注意。

本书的参考学时约为 15~16 周, 每周 2~3 学时。第一章和第二章的内容是 UNIX 操作系统的原理、应用和编程介绍, 以及 UNIX 操作系统的软件开发工具的使用(包括 Shell 编程, vi, sed, awk, make, SVN, lex 和 yacc 等), 参考学时约为 8~9 周。第三章讨论 C 语言, 参考学时约为 2~3 周。本书没有从一般程序设计语言的角度来介绍 C 语言的使用, 而是讨论了 C 语言中比较重要的一些课题, 以期引导读者理解和学会编制结构合理、风格良好的程序。第四章介绍数据结构的基本内容, 并用 C 语言来实现各种算法, 因为使用合理的数据结构对软件开发是至关重要的, 参考学时约为 4~5 周。配合课堂教学的上机实习很重要, 每次课可安排 2 学时的上机操作。书中尽量给出实例, 并且已在相应的操作系统等计算机环境中经过验证。

本书内容的积累与多年来复旦大学电子工程系/微电子学系 CAD 研究室的软件开发和算法研究等科研工作密切相关。曾璇教授和王伶俐教授在参与本课程的讲授过程中认真和严谨的治学态度给予作者精神力量。青年教师陶俊和朱恒亮在参与教学的过程中, 不仅得到自身能力的提升, 也用他们的科研经验帮助作者更新了教学的内容。我们的同事及百多位历届研究生的科研工作经验是本书得以完成的基础, 数十位担任助教的研究生为我们的教学进步提出了许多切实的意见, 二十几届学习本课程的本科生是令本书内容和作者本人保持活力的不可或缺的推动力。

在本书的编写过程中, 作者的同事帮助提供和核对各种参考资料, 难以枚举的学生协助我们制作文稿的电子版。但是在征求这些同事署名著作时, 竟然被一一婉拒, 作者为他们的

信任而骄傲,所得更多的是感动!

恩师唐璞山教授再次欣然为本书作序,令作者油然升起对他的无限敬意,把人做好,把书写好,把课教好才是报答师恩的实际行动。复旦大学微电子学系的周嘉教授给予作者的信任和支持也难以致谢。复旦大学出版社责任编辑范仁梅和张志军对本书的出版给予了大力支持。

为此,谨向以上各位表示诚挚的敬意和谢意!

虽然本人尽力,但书中难免存在不足和谬误之处,敬请读者不吝给以指正,以期共同提高。

作 者
2011 年 3 月

目 录 Contents

第一章 操作系统及 UNIX Shell / 1

1.1 什么是操作系统 / 1

1.2 操作系统的分类 / 2

1.2.1 实时操作系统 / 2

1.2.2 作业处理操作系统 / 4

1.2.3 批处理操作系统 / 6

1.2.4 分时操作系统 / 6

1.2.5 其他类型的操作系统 / 7

1.2.6 工业界的主流操作系统 / 7

1.3 UNIX 操作系统的发展史 / 9

1.3.1 UNIX 系统的发展历史和特点 / 9

1.3.2 Linux 系统的发展历史和特点 / 10

1.4 UNIX 操作系统的功能模块 / 11

1.4.1 处理机和进程管理模块 / 11

1.4.2 存储管理模块 / 15

1.4.3 设备管理模块 / 17

1.4.4 文件系统 / 19

1.4.5 用户界面 / 23

1.5 Shell 概述 / 26

1.5.1 shell 的种类 / 26

1.5.2 shell 程序的识别 / 27

1.5.3 shell 环境 / 28

1.5.4 UNIX 的用户管理 / 29

1.5.5 UNIX 命令和工具表达形式的约定 / 31

1.6 B shell / 32

1.6.1 shell 命令的主要特征 / 32

1.6.2 shell 程序 / 41

1.7 C shell / 48

1.7.1 C shell 变量 / 48

1.7.2 C shell 特征 / 51

1.7.3 控制结构 / 51

1.7.4 文件名补齐 / 53

1.7.5 假名 / 54

1.7.6 命令史 / 55

1.7.7 工作目录栈 / 58

1.8 Bash / 60

1.8.1 Bash 的增强特征 / 60

1.8.2 命令行扩展功能 / 62

1.8.3 函数 / 63

第二章 UNIX 系统的软件开发工具 / 68

2.1 软件开发工具概述 / 68

2.2 BNF 范式 / 70

2.2.1 BNF 范式的定义 / 70

2.2.2 BNF 范式的特征 / 71

2.2.3 BNF 范式示例 / 71

2.3 vi / 73

2.3.1 启动 vi 和 ex / 73

2.3.2 vi 的工作状态及其转换 / 74

2.3.3 编辑对象及定位 / 76

2.3.4 输入类命令 / 77

2.3.5 输出类命令 / 77

2.3.6 退出类命令 / 77

2.3.7 插入类命令 / 77

2.3.8 删除类命令 / 79

2.3.9 搜索命令 / 80

2.3.10 全局操作命令 / 80

2.3.11 块操作命令 / 83

2.3.12 设置命令 / 85

2.3.13 杂类命令 / 85

2.3.14 宏命令 / 86

2.3.15 vi 环境的定义 / 87

2.4 sed / 87

2.4.1 sed 命令 / 88

2.4.2 地址表达式和匹配字符串 / 89

2.4.3 删除命令 / 89

2.4.4 替代命令 / 89

2.4.5 输入输出命令 / 91

2.4.6 块操作命令 / 91

2.4.7 实例 / 92

2.5 awk / 96

2.5.1 awk 概述 / 96

2.5.2 记录、字符段和特殊变量 / 97

2.5.3 变量及运算 / 97

2.5.4 awk 程序的匹配模式 / 98

2.5.5 awk 程序的操作 / 100

2.5.6 数组 / 101

2.5.7 内部函数 / 101

2.5.8 控制结构 / 102

2.5.9 实例 / 103

2.6 make / 107

2.6.1 make 的功能 / 107

2.6.2 描述文件 / 109

2.6.3 后缀定义和传递规则 / 111

2.6.4 变量和宏定义 / 112

2.6.5 make 命令行 / 113

2.6.6 包含文件 / 114

2.6.7 控制结构 / 115

2.7 SVN / 118

2.7.1 SVN 的基本概念 / 119

2.7.2 SVN 的基本工作流程 / 121

2.7.3 SVN 帮助命令 / 122

2.7.4 创建版本库 / 122

2.7.5 初始化导入 / 122

2.7.6 导出工作拷贝 / 123

2.7.7 目录树修改命令 / 123

2.7.8 提交命令 / 125

2.7.9 拷贝-修改-合并机制 / 126

- 2.7.10 更新命令 / 126
- 2.7.11 检查状态命令 / 127
- 2.7.12 查看差异命令 / 128
- 2.7.13 放弃命令 / 129
- 2.7.14 删除冲突命令 / 129
- 2.8 lex 和 yacc / 130**
 - 2.8.1 lex 和 yacc 概述 / 130
 - 2.8.2 lex 的源程序结构 / 134
 - 2.8.3 词法规则 / 134
 - 2.8.4 词法规则的操作语句 / 135
 - 2.8.5 lex 的定义部分 / 137
 - 2.8.6 lex 的用户程序 / 139
 - 2.8.7 yacc 的源程序结构 / 140
 - 2.8.8 yacc 的说明部分 / 141
 - 2.8.9 语法规则 / 142
 - 2.8.10 lex 和 yacc 的界面函数 / 144
 - 2.8.11 实例 / 147

第三章 C 语言及编程规范 / 152

3.1 基本问题 / 152

- 3.1.1 数据类型 / 152
- 3.1.2 常数 / 154
- 3.1.3 运算符 / 154
- 3.1.4 存储类型 / 156
- 3.1.5 流程控制 / 158

3.2 指针 / 164

- 3.2.1 指针概念 / 164
- 3.2.2 指针和数组 / 166
- 3.2.3 指针和字符串 / 167
- 3.2.4 指针和动态内存单元 / 169
- 3.2.5 指针数组和 multidimensional array / 172
- 3.2.6 数组指针 / 174
- 3.2.7 多级指针 / 175
- 3.2.8 指针应用举例 / 176

3.3 结构 / 180

- 3.3.1 结构的定义方式 / 181

3.3.2	结构成员的引用	/ 183
3.3.3	链表	/ 184
3.3.4	插入链表结点	/ 187
3.3.5	删除链表结点	/ 197
3.4	联合	/ 202
3.5	函数	/ 204
3.5.1	函数的定义和调用	/ 204
3.5.2	函数之间的通信	/ 206
3.5.3	指向函数的指针	/ 209
3.5.4	递归函数	/ 211
3.6	文件操作	/ 212
3.6.1	文件的存取	/ 212
3.6.2	低级输入输出	/ 214
3.6.3	文件的随机存取	/ 216
3.7	C 语言和 shell 的通信	/ 217
3.7.1	命令行参数	/ 217
3.7.2	命令行可选项	/ 218
3.7.3	环境变量	/ 220
3.8	C 语言编程规范示例	/ 221
3.8.1	数据类型和运算符	/ 221
3.8.2	命名规则	/ 222
3.8.3	变量、指针和初始化	/ 222
3.8.4	函数	/ 223
3.8.5	合理使用空格	/ 223
3.8.6	排版	/ 224
3.8.7	注释	/ 226
3.8.8	文件	/ 227

第四章 数据结构 / 228

4.1	基本概念	/ 228
4.1.1	数据结构的定义	/ 228
4.1.2	结点和存储单元	/ 228
4.1.3	数据结构的存储方式	/ 230
4.1.4	数据结构的分类	/ 231
4.1.5	数据结构的基本操作	/ 232

4.2 线性表 / 233

- 4.2.1 基本问题 / 233
- 4.2.2 查找结点 / 234
- 4.2.3 添加结点和删除结点 / 236
- 4.2.4 线性表查找算法 / 238
- 4.2.5 线性表排序算法 / 240
- 4.2.6 线性表基本操作算法的时间复杂度 / 247

4.3 栈和队列 / 248

- 4.3.1 栈 / 248
- 4.3.2 顺序栈 / 249
- 4.3.3 链接栈 / 251
- 4.3.4 队列 / 255
- 4.3.5 顺序队列 / 256
- 4.3.6 环形队列 / 258
- 4.3.7 链接队列 / 259
- 4.3.8 双向栈 / 260

4.4 树 / 261

- 4.4.1 树的定义 / 261
- 4.4.2 树的括号表示 / 262
- 4.4.3 树的基本操作 / 262
- 4.4.4 树的存储形式 / 263
- 4.4.5 树的遍历 / 268

4.5 二叉树 / 274

- 4.5.1 二叉树的定义 / 274
- 4.5.2 二叉树的括号表示 / 274
- 4.5.3 二叉树的存储 / 275
- 4.5.4 二叉树的遍历 / 276
- 4.5.5 二叉树的顺序存储 / 277
- 4.5.6 二叉树的层号表示 / 282
- 4.5.7 二叉树的生成算法 / 284
- 4.5.8 任意次树到二叉树的转换 / 294
- 4.5.9 穿线树 / 296

4.6 图 / 299

- 4.6.1 图的定义 / 299
- 4.6.2 图的基本概念 / 300
- 4.6.3 图的基本操作 / 301

4.6.4 图的存储形式 / 301

4.6.5 图的遍历 / 306

4.6.6 图的应用举例 / 310

附录 A ASCII 字符集 / 314

附录 B B Shell, Bash 和 C Shell 比较一览表 / 317

附录 C C 语言标准库函数 / 322

参考资料 / 328

1

Chapter

2

3

4

第一章

操作系统及 UNIX Shell

1.1 什么是操作系统

计算机^①系统由硬件和软件两大部分组成,如图 1-1 所示。在计算机技术高度发展的过程中,硬件和软件的关系日益密切。硬件是指计算机系统中的各种物理资源,如中央处理器(CPU),存储器(Memory 和 Storage),其他各种外部设备(简称外设)等。它们与外部的界面是指令系统。软件通常分为系统软件和应用软件。

系统软件是计算机系统必须配备的,对各种领域的用户都通用的软件。它包括操作系统和系统实用程序。操作系统(Operating System,简称 OS)是计算机软件系统中最重要的系统软件。系统实用程序是指除了操作系统之外的系统软件,例如,各种程序设计语言的处理程序(编译、链接、解释等)、标准程序库、软件维护程序等。最早的时候,不包含任何软件的计算机称为裸机。人们使用裸机是很麻烦的,有了系统软件,使得计算机容易操作,从此也促进了计算机硬件和软件的发展。

应用软件是针对某些特定应用领域配置的软件。例如:用于计算机辅助设计、计算机辅助工程、计算机辅助制造、计算机辅助教学、工业控制、信息处理、数据处理等用途的软件。它们往往与某些计算机外围设备一起实现某方面的应用。

本书要介绍的是操作系统以及系统实用程序中的一些软件工具。

不论是系统软件(除操作系统以外)还是应用软件,都必须在操作系统的管理和支持下运行。除了早期的计算机外,现代的计算机都配有一个或几个操作系统,而且操作系统的性能在很大程度上决定了计算机系统工作的优劣。在计算机系统上配置操作系统,首先是将裸机改造成一台功能更强、服务质量更高、用户使用起来更加灵活方便、更加安全可靠的虚拟机,其次是利用操作系统的功能,充分合理地使用计算机系统包含的各种软件、硬件资源,提高整个系统的使用效率和经济效率。

^① John Von Neumann(约翰·冯·诺依曼,1903—1957年),美籍匈牙利人,被誉为计算机之父。1945年发明了第一台现代意义的通用计算机 EDVAC(Electronic Discrete Variable Automatic Computer,离散变量自动电子计算机)。奠定了由5个基本部分组成的现代计算机体系结构,首次使用二进制的运算机制。

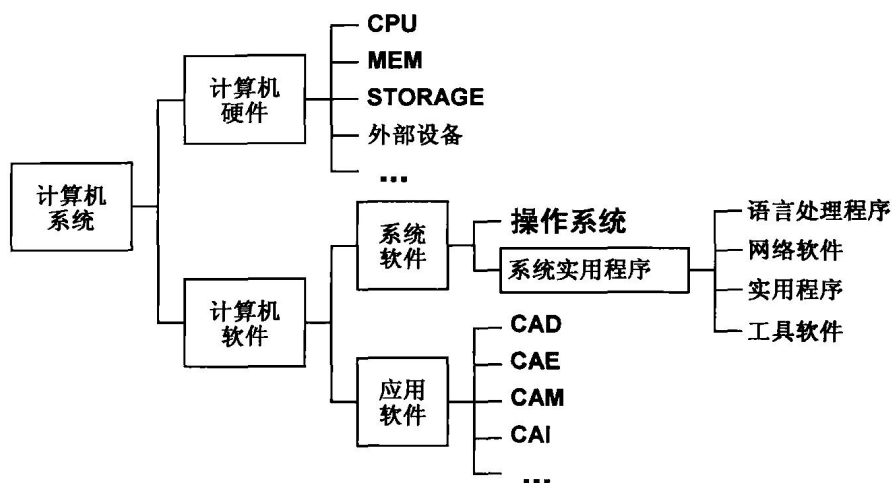


图 1-1 计算机系统的硬件和软件

简单地说,操作系统是计算机系统(硬件和软件)和用户之间的中介(interface, 界面、接口、管理员)。用户要使用计算机,必须学会与操作系统打交道。本章的目的旨在使读者在了解操作系统原理的基础上,通过学习操作系统和某些系统实用程序(在很多情况下,我们把它们合在一起称为操作系统)的使用来了解计算机的各种功能。

对一个用户来说,他面对的是计算机的各种硬件资源和软件资源。软件资源就是上面提到的操作系统、系统实用程序和与它有关的领域中的应用资源。他将通过与操作系统打交道来使用各种计算机资源,如图 1-2 所示。

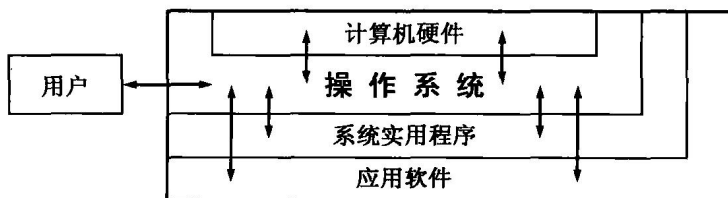


图 1-2 计算机资源与用户的关系

对于计算机系统来说,它将面对的是各种领域中的用户。因而,从计算机系统管理人员的观点来说,引入操作系统的目的是为了合理地组织计算机的工作流程,分配计算机资源为用户享用,以便更好地发挥计算机的效率。

1.2 操作系统的分类

根据不同的观点,对操作系统可以有不同的分类法。本节将讨论从功能出发的分类,并将操作系统分成实时操作系统和作业处理操作系统两大类。

1.2.1 实时操作系统

实时操作系统(Real Time Operating System)简称为实时系统。所谓“实时”,就是“立

即”或“及时”的响应。实时系统的含义是指系统能够及时响应随机发生的外部事件,随时采集现场信息,加以分析处理,并在限定的时间内完成对外部事件的处理。目前有 3 种比较典型的实时系统:过程控制系统(Process Control System)、信息查询系统(File Interrogation System)和事务处理系统(Transaction Processing System)。

(一) 过程控制系统

过程控制系统主要应用于工业过程控制(例如炼钢)、军事预警(例如导弹控制)和太空技术等。该系统的特点是,与被控制过程的变化速度相比,系统的反应速度足够快;工作安全可靠,并在某些软件或硬件发生故障的情况下不妨碍正常工作;极少需要人工干预,即使干预,其操作也很简单方便。

1991 年 1 月,海湾战争爆发。美国的“爱国者”导弹成功地拦截并摧毁了伊拉克的“飞毛腿”导弹,引起了全世界的瞩目。“爱国者”地空导弹系统便是一个实时系统的例子。当伊拉克的“飞毛腿”导弹发射后几秒钟,预警飞机、地面雷达和侦察卫星上的计算机都证实了“飞毛腿”导弹发射无疑,并算出了发射地点,及时将数据传送给“爱国者”导弹系统。指挥控制中心的计算机算出“飞毛腿”导弹的飞行速度、方向和弹着区域,并制订出“爱国者”导弹的拦截弹道,仅有的两名指挥人员按下了控制台上的发射钮。“爱国者”导弹根据自身的制导系统和地面指挥中心发来的修正指令,直向“飞毛腿”导弹飞去。当测出两颗导弹相距 20 m 时,引爆所携的 45 g 炸药。在两弹相碰的一瞬间,引爆了“飞毛腿”导弹携带的 1000 kg 炸药。从“飞毛腿”导弹发射到高空被摧毁,前后历时 5 min 35 s。

(二) 信息查询系统

信息查询系统通常配有大型文件系统或数据库,事先存有经过合理组织的大量数据,能及时响应用户的查询要求,迅速给出答复。多用于情报检索、库存查询和电话问讯业务等。

(三) 事务处理系统

事务处理系统类似于信息查询系统,事先备有大量的信息和数据。它根据用户的申请,及时地将文件或数据库中的信息实施无差错的更新,例如,银行业务处理系统、订票业务处理系统等。这样的系统要求具有一定的保密措施,以及相当高的可靠性,并能进行并发处理。

综上所述,实时系统有以下特点。

1. 及时性强

不同的实时系统对及时响应的速度有不同的要求,但必须根据应用的要求而确定。

2. 高可靠性

可靠性对实时系统极为重要,因此,实时系统往往具有容错管理能力。一些重要的实时系统还采用双机系统。

3. 简单的交互能力

实时系统大多是专用系统,其交互功能要求不高,仅提供一些简单的操作命令,并限制操作时不能修改现有的程序。