

软件世界

SOFTWARE WORLD

(双月刊)

1984年9月创刊

社 长 周慕吕
主 编 周锡龄
副 主 编 应 明 孙毓林
主 办 机电部计算机与微电子发展研究中心
编辑出版 中国计算机报社《软件世界》编辑部
地 址 (100846)北京万寿路27号
印 刷 机电部电子情报所印刷厂
国内总发行 北京报刊发行局
订 购 全国各地邮局
邮发代号 82-469
刊 号 ISSN 1000-8926/CN11-2105
广告许可证号 京海工商广字004号
每期定价 1.50元 全年定价 9元
出版日期 1993年3月21日

MAIN CONTENTS

Open System and It's Evolution(2)
Virtual Reality Technology and It's Application Prospect(7)
A Practical Prototyping Support Environment(9)
The Practical Software Design for a Supervising System and a Control Terminal Office(12)
EIMS——IBM/PC Edit-directed Intelligent Transactions Management System(17)
Development of Graphic and Text Processing Software with SunCGI and Sunview(20)
The Implementation of User Exit in SQL * Forms(22)
Test and Evaluation of Borland C and Microsoft C(28)
Current State of Multimedia Operating System(32)
Software Reliability Engineering, It's Basic Concept, Tasks and Implementing Methods(34)
A Program's Complexity Metrics(37)
Directive to the Preparation of Software Level Test(42)
UNIX System V Programming Lecture
Chapter 4 Control Operation of UNIX files (50)

目 录

1993年第2期(总第78期)

技术研讨

开放系统及其发展……………王桂兰(2)
虚拟现实技术及其应用前景探讨……………张质洞(7)

开发与应用

一种实用的原型开发支持环境……………姚淑珍(9)
实用监控系统控制端局的软件设计……………宣东(12)
IBM/PC 编辑式智能事务管理系统 EIMS ……章登植(17)
SunCGI 和 SunView 组合开发图文处理软件的探讨……………江红(20)

实践与经验

SQL * FORMS 用户出口的实现过程……………刘先强 陈义强(22)
菜单设计方法与弹跳式菜单程序……………李桂根(22)
隐含方式加密子目录的解密程序……………邓波 蒲昌平(24)
认可方式的程序设计……………何昌波(25)
数据输入程序中编码的记忆录入……………毛奕(26)
dBASE III 应用中的两个小技巧……………李建春 王小青(27)

软件评测

Borland C 与 Microsoft C 的评测……………孟军(28)

技术与产品动态

多媒体操作系统进展近况……………陈少昭(32)

标准化与质量管理

软件可靠性工程的基本概念、任务与实施方法……………徐仁佐(34)
一个程序的复杂性度量……………弓惠生(37)

经营与管理

软件经营连载之二——软件开发的经营……………陈幼松(39)

软件水平考试

考生备考指南……………林欣(42)
离散数学学习指导与试题分析……………耿素云(44)

技术讲座

UNIX System V 程序设计讲座
第四讲 UNIX 文件控制操作……………周祖伦等(50)

软件市场

厂商志……………(55)

厂商志

中国长城计算机集团公司——高科技领域的一朵奇葩……………(57)

信息之窗……………(21,31,49,54)

计算机软件著作权登记公告……………(58)

开放系统及其发展

王 桂 兰

目前,工业界对开放系统的定义还没有形成共识,但人们普遍认为开放系统应当具有以下三个主要特征:

(1)可移植性(Portability)——操作系统或应用软件稍微修改或不做修改就可以在不同厂家的不同硬件平台上畅通无阻地运行;

(2)互操作性(Interoperability)——能在不同厂家的计算机之间进行联网,互相交换信息;

(3)可剪裁性(Scalability)——可在不同厂家、不同档次的计算机之间互相传递信息。

对于某一系统而言,要高水准地达到以上三点是很困难的,但是这些目标已经成了工业界共同的努力方向。

开放系统的由来及发展

开放系统的概念早在60年代初就已出现。那时主要表现为对程序可移植性的追求,希望为一台机器编写的程序不做或少做修改就可以在另一台机器上运行。为适应此形势要求而出现的系列机、兼容机为运行同一操作系统的硬件的更新换代提供了方便,但是用户却仍然局限于同一厂家的同类机型上,不能在不同厂商的产品间择优。

随着信息技术日新月异的发展,高速度、大容量、数据库共享、不同机种之间的网络连接、24小时连续操作、远距离维护等新技术、新功能的不断出现,用户要求打破一家技术的界限,实现数据共享、技术共享,以支持在市场竞争中的需求。

在这种形势下,AT&T首先顺应时代的要求,在1970年用与机器无关的C语言编写出了可供各种机型使用的UNIX操作系统。

自从UNIX在AT&T公司贝尔实验室的PDP-7型机上产生以来的二十多年中,产业界不断对它进行丰富和发展,使它成为成熟的开放系统领域的主流操作系统。

UNIX是在当时的硬件水平下,为了共享资源和提供字符型服务而设计的。它的出现使在各种硬件平台上建立一个通用的软件平台的愿望成了现实,使人们看到了开放系统给人类带来的好处,也认识到开放系统是不可逆转的潮流。目前,UNIX几乎成了开放系统的同义语和代名词。UNIX以其简练

高效、用户界面好、执行速度快、软件丰富、可移植性强、联网功能强、发展潜力大、廉价等优点征服了整个工业界。从八位微机到大型机、巨型机,从工程工作站到事务处理、办公自动化,UNIX之风吹遍了计算机世界的各个领域,直到如今,对UNIX的开发和研究仍然方兴未艾。UNIX的诸多优势吸引了工业界的注意,早在80年代中期,许多硬件厂商就开始意识到UNIX将成为操作系统的标准,1987年UNIX已经成为计算机领域里四项最热门的技术之一。1987年美国市场研究公司IDC把UNIX的活力增强评为该年度计算机工业的十项大事之一,并把对UNIX广为接受已成为定局列为1988年十项重要发展展望的第一项。工业界的许多专家预计UNIX在90年代将成为多用户、多任务应用领域中的主流操作系统。在这种形势下,由于UNIX的分时操作和多用户设计吸引了众多的厂家,于是便出现了各式各样的UNIX变种产品,现在,最流行的UNIX操作系统有:

USL: SVR4	OSF: OSF/1
SCO: XENIX	IBM: AIX
DEC: ULTRIX	SUN: SunOS
AT&T: SVR3	Apple: A/UX
HP: HP/UX	SCO UNIX

在应用软件方面,以UNIX操作系统为平台的应用软件也已经不下几千种。主要的PC机应用软件,诸如Lotus1-2-3、Wordperfect、dBASE、publisher等已经或者正在转向UNIX。由于OSF开发的OSF/Motif和AT&T及SUN开发的OpenLook两个标准图形用户界面(GUI)的出现,大大简化了UNIX的使用。

到目前为止,UNIX软件的世界市场已经达到80亿美元,用户已经超过1000万家。据估计,到1997年将增加到300亿美元。自1991年起,UNIX的应用从非商用的计算机向商业应用发展的趋势更加明显,在1991年交货的12.5万台工作站中用于商业领域的占1/4。据IDC市场研究人员估计,1992年UNIX-PC和UNIX工作站的数量约增加40%。

开放系统领域的两大联盟 ——OSF和UI

80年代末使人难以忘怀的UNIX大战促进了

UNIX产品的丰富和发展。当时,虽然 AT&T 首先研究出了 UNIX,但在计算机市场上,AT&T 的力量远不如 IBM 和 Apple 等公司强大,所以,人们对 UNIX 的前途有一种不踏实感。为了加强 UNIX 的形象,AT&T 与 Sun 微系统公司结为联盟,以利用 Sun 的 SunOS(SunOS 是 BSD 的商业版),使 UNIX System V 与 Berkely 软件发行版(BSD)合并,加强 UNIX System V 在市场上的竞争力。

AT&T 的这一举措使工业界许多有名的公司感到了威胁,它们认为 AT&T 与 Sun 的联盟结果是出现了不平衡的竞争态势。为了扭转自己在竞争中可能处于的劣势,从 AT&T 手中夺取对 UNIX 发展前途的控制权,IBM、DEC 和 HP 联合 Siemens、Apollo 等其他支持者于 1988 年 5 月成立了一个强大的联盟——开发软件基金会(Open Software Foundation),简称 OSF,并以该组织为基础开发与 AT&T 无关的 UNIX 操作系统。

作为对 OSF 联盟出现的一个反应,在 OSF 成立之后的几个月内,AT&T 则联合支持 System V 的厂商,例如 Sun、NCR、Unisys、Apple、日本电气等公司也成立了一个庞大的联盟——UNIX 国际(UNIX International),简称 UI。从此以后,开放系统运动在这两个机构的组织领导下,围绕两个 UNIX 版本——OSF/1 和 System V 展开了竞争。

但是, System V 和 OSF/1 都遵循 XPG3、POSIX 和其他重要的工业标准,都运行 Motif 图形用户接口,所以两个版本基本上是相似的。所不同的是 System V 在市场上占据了“先入为主”的优势,它利用了 UNIX 的过去,尤其是 SVR4(UNIX 系统 V 第四版本)把 System V、BSD 和 XENIX 的早期版本结合在一起,其中 SCO 和 Microsoft 共同为 PC 机开发的 XENIX 已经占据了广大的市场。UI 的这种作法使 SVR4 很快成为根基很牢的、能适应市场要求的技术。

OSF 则没有采用已经投入市场很久的技术,而是以卡内基·梅隆大学为多处理而研究的新技术——Mach 为基础,于 1991 年推出 OSF/1 操作系统。Mach 是以 Berkeley 大学的 4.2BSD 为基础、类似 UNIX、与 System V 兼容的一个操作系统。

OSF 的技术开发活动及 OSF/1

·OSF/1 是支持多处理器的 Mach 内核,并与 IBM 的操作系统相结合的产物,它与 System V 和 BSD 相比性能和功能更好,可靠性更高。

·OSF/1 的内核实现了模块化,因而增加了操作

系统的可扩展性和可剪裁性,便于系统的维护和移植。

·OSF/1 的文档系统的结构以 4.2BSD 的虚拟文档系统为基础,具有简明的用户界面,并与现有的 UNIX 操作系统保持兼容和互操作性。

·OSF/1 具有网络间的通信协议 TCP/IP 和与 NFS 兼容的分布式文档系统,具有很强的联网能力。

·OSF/1 的装载程序常驻用户空间,使 OSF/1 具有很强的扩充能力。

·OSF/1 可以用 C 编译程序进行编译。

·OSF/1 支持的 Motif 用户界面利用了 DEC、HP、Microsoft 的技术(如 DEC 的 XUI、HP 的三维外观等)和 Presentation Manager 的风格,采用单一的 API,得到全世界众多的计算机和软件供应商的支持,到目前已经在 120 多种硬件平台和 43 种操作系统上运行。

·在国际化方面,OSF/1 支持 XPG3 和 POSIX 规范。它支持 8 位的简单命令,因此可以进行多字节的数据交换,可以处理 ASCII 码以外的文字集,可以支持 15 个以上的国家扩充自己的文字功能。

·OSF/1 达到了 B1 级安全性。

·OSF/1 的系统管理功能是 4.3 BSD 的管理功能与 System V 的管理功能的结合,对商业系统还增加了逻辑卷号管理和磁盘镜像化功能。

OSF 成立之后即在全世界广泛征求技术。1989 年 4 月,它首先选择英国国防研究局的 TDF 技术作为 OSF 中性结构分布格式(ANDF)的核心。ANDF 是一种编译程序的中间语言技术,是能在各种硬件平台上运行的中间格式的、介于源代码和二进代码之间的中间代码。按这种独立于硬件的分布格式开发和分布的应用软件可以在各种不同的开放系统结构上安装和运行。软件开发人员利用 ANDF 技术在不同的平台之间移植应用软件时,不需对应用软件的二进制接口(ABI)进行再汇编。系统厂商在推出新产品时只需给硬件配上相应的 ANDF 安装器/翻译器就可以了。因为这项技术为多种平台提供一个一致的软件开发、分布环境,因此对软件的分布方式具有重要的影响。用户对它反应强烈,希望能早日推出,但由于厂商们认为工业界对推出 ANDF 技术的准备不足而延缓了它的开发进度。虽然 OSF 也曾承诺将于 1993 年广泛推出这种技术,但有可能要延缓推出的时间。

1991 年秋 OSF 推出的 DCE 支持异种网络环境中的互操作,提供综合式的分布式计算服务,如远程调用、分布式命名服务、网内时间服务、多线程服务、

安全性服务、分布式文件系统等。在 DCE 交货之后,几个大的系统厂商即着手把它与自己的产品集成。目前,DEC、Gradient Technologies、Groupe Bull 和 Transarc 等公司都已推出了这种产品。HP、Hitachi、IBM、Siemens Nixdorf 和其它厂商正在 DCE 产品上做工作。另外还有50多个机构也宣布支持这种产品。

在 DCE 上市的同时,OSF 宣布了对 DME 技术的选择,1992年秋天,OSF 会员得到了与 OSF/1 集成的第一个 DME “锥形”产品。1993年上半年将推出与 DCE 共同提供管理服务的 DME 的分布式服务部分。在服务层和操作系统之间提供内部应用通信的分布式结构将于1993年下半年推出。

SVR4及 UI 的规划图

越来越多的用户转向联机的、共享资源的环境,以求建立全球、全企业范围的计算机系统。他们要求一种开放式的、遵守各种标准的、适合于技术转让的操作系统。SVR4以用户的这种要求为动力,在世界上取得了极大的成功。

UNIX System V 是一个通用的多任务、多用户操作环境。它适用于各种不同的任务,从工厂自动化的简单的过程控制到用于支持终端服务器的个人工作站,都可以运行进行事务处理的应用软件。UNIX System V R4.0版及其后续版本及附加特性可以应用于各种环境,例如分布式计算、企业中心计算(Corporate Hub)和台式计算。

自1989年11月 UI 宣布 SVR4公开发售以来,许多大的系统供应商已经在十多种处理器上配置了以 SVR4为基础的产品。1990年底,已有300多个公司提供 SVR4平台上的产品,出现了硬件和软件供应商向 SVR4标准的工作环境发展的倾向。1991年之后,大的开放系统供应商中已有三分之二提供了基于 SVR4的产品。这种现象说明,SVR4已在市场上建立了牢固的基础,它的特性和功能适合最终用户的要求。

SVR4综合了 SVR3、BSD4.2/4.3、SunOS 和 XENIX 的功能,符合 POSIX 1003.1、X/Open、XPG3和 ANSIC 标准,并具有以下特点:

- 它的应用目标码接口(ABI)可以使不同机型上的应用软件直接运行。
- 它的存储映像文件、共享内存、普通文件对换、可移植硬件地址变换等新技术提高了程序的执行效率。
- 它的虚拟文件系统(VFS)比以前的文件系统的功能更强。

- 使用统一的通信接口协议,并支持 TCP/IP、RPC、XDR、NFS、RFS 等工业标准。

- 使用 OpenLook 作为它的图形用户接口。

- 以命令或菜单方式简化系统的管理。

- 实现了 B2级安全性。

- 支持实时应用,支持用户进行可控制的进程调度、高精度时钟和内存锁定等。

- 支持多字节字符,便于实现国际化。

- 其动态连接功能可节省硬盘空间。

- 提供与 BSD 和 XENIX 兼容的软件包。

为了使 UNIX System V 成为开放系统供应商和应用软件开发者的最佳平台,满足在分布式计算、企业中心计算和台式计算这三个关键领域中工业界提出的要求,UI 会员及 UNIX 系统实验室拟出了发展 System V 的计划。

1990年,UI 公布了 UNIX System V 规划的第一个版本,这是开放系统的主要组织机构首次公开披露长期的产品计划进度和要求。通过这个规划文件,UI 的会员公司向工业界指出了 UI 开发开放系统产品所沿袭的方向以及开发进度。规划以一种新的方式使工业界(制造商、独立软件商(ISV)、系统集成商和最终用户)深入了解 UI 的会员公司为了设计 UNIX System V 的未来版本和附加性能而确定的整个计划的全部要求。

在这个规划公布后的几个月内,UI 决策委员会收到了对该规划的各种意见,也收到了开放系统工业界的各个方面提供的规划中所描述的技术。UI 决策委员会的规划专门工作组对这些意见和市场上发生的变化进行了分析,对规划进行了适当的修改,形成了 System V 规划的1991年版本。

与第一个版本相比,1991规划的重要变化是,提前推出多处理版本和提前推出 OSI(开放系统互连)技术。

规划(1991年版)表明通过 UI 的程序开发和公布应用程序接口(API),System V 将获得全新的发展。

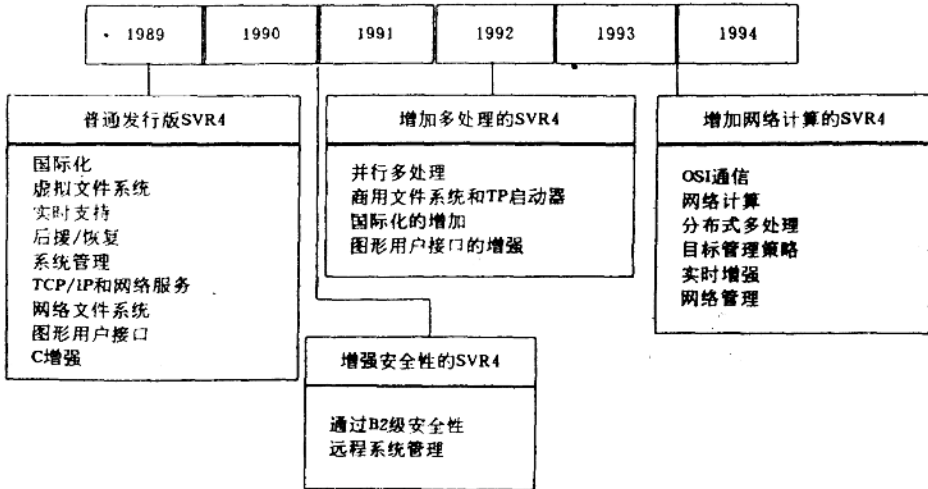
按照规划,UI 在1991年推出两种新版本:一种有寻址功能的增强安全性版本和一种对称多处理的基准实施版本。增强安全性版本(SVR4 ES)获得了国家计算机安全中心的 B2级安全性的官方认可,并具有某些 B3级安全性。SVR4 ES 具有高级别的安全性,具有远程系统管理功能和附加的国际化能力。这些特性使 SVR4既适合美国政府和商业界的要求,也适合国际市场的要求。

从1991年初开始,USL 分阶段推出多处理的增

强。对对称多处理功能的开发主要分在两个阶段。第一个阶段,即 SVR4 MP 阶段,提供一个高度平行的基础系统,重点在内核级的多线程结构和各种联网协议,例如 Streams、RFS、NFS 和 TCP/IP。以后推出的 SVR4 ES/MP 把安全性版和多处理版相结合加以增强,使现有的应用软件可以利用多处理系统的性能优势。第二个阶段,也即 SVR4 ES/MP 阶段,为

具有 SVR4 ES 安全性的现有应用软件提供同样的优势。除此之外,它还有并行的编程接口,这样,新的应用软件的设计就可以利用并行化的优点,产生出更多的应用软件和使用软件的可剪裁性更强。以上两种多处理实施均具有二进制兼容性和源码兼容性。目前 SVR4 MP 产品仍然作为单个产品发行。到 1993 年年底,USL 还要推出一系列 OSI 产品。

UI 规划(第一版)的计划进度



1992年6月,USL 推出了 UNIX System V 的最新版本——R4.2(又称 Destiny 或 UNIX Lite)。SVR4.2能与 Berkeley 和 SCO 的各种 UNIX 版本兼容。但 SVR4.2实际上并没有改变系统 V 的基本设计体系,只是 SVR4基础版本与增强安全性版本的集成。

在 AT&T 把 UNIX 的商标权转让给 USL 之后,在产品的计划和开发方面 USL 和 UI 密切协作。由于 USL 的努力促进以及 Univel 和 Sunsoft 在工业标准计算机上提供基于 SVR4 的操作系统,USL 的 SVR4 UNIX操作系统技术在工业界得到更广泛的注意,使人们更加认为 SVR4既是一个统一 System V、Berkeley、SCO UNIX 产品的 UNIX 版本,又是一个已推向市场的最先进的 UNIX 技术。

USL 的这些策略对市场产生了重大的影响。现在越来越多的人认为 SVR4是 UNIX 系统开发的最高技术,而 SVR4.2是台式机最好的 UNIX 系统,也是 Microsoft 产品的 UNIX 替换系统。另外,为了加入 Univel 的 UnixWare 产品系列,现在 USL 也开始开发和促进台式机市场所用的最新版本 SVR4.2的 Intel 实施。

有人认为,1992年11月份推向市场的 UnixWare

是市场上的第一个 SVR4.2 的实施版本,它承担了第一个版本的全部风险,它的质量是一个未知数,需要用用户的实践来证明。

开放系统 UNIX 的标准化

开放系统必须符合许多工业标准,没有标准就无法实现开放,只有各个厂家依据标准开发出的软件,才能真正具备开放所要求的可移植性、互操作性和可剪裁性。

目前 UNIX 的标准化工作已经不仅仅限于对 UNIX操作系统,而是对整个软件运行环境进行全面的标准化。现在,从事 UNIX 标准化的有关组织主要有 POSIX 委员会和 X/Open。

POSIX 委员会是 IEEE 计算机学会下设的标准化活动理事会所属的许多技术委员会之一,受专门负责制订操作系统和应用可移植性标准的 TCOS 委员会直接领导,从事 UNIX 的标准化工作是它的专门任务。该委员会要制定 POSIX 所涉及的四个方面(基础类标准、开放系统环境类标准、语言联编类标准和相关类标准)的25项内容的标准。由于 IEEE 是一个民间机构,它需将 POSIX 的成果报 ANSI(美国

国家标准所),经审查批准后才能成为国家标准。国家标准报 ISO(国际标准组织),经审查批准后才成为国际标准。

X/Open 是一个计算机制造商的行业组织,是为了在同行业之间建立一致的、能满足用户要求的软件运行环境而建立的。X/Open 已按照 XPG 规范建立了一套评测机构,对各厂家的产品进行 XPG 规范验证,经过验证的产品即发给 X/Open 商标。凡具有 X/Open 商标的产品都符合这种规范。X/Open 从现有产品中选择要用的标准,经该组织成员投票通过后登载在《X/Open 可移植性指南》(XPG)上推广。

虽然 X/Open 不是一个官方机构,但由于该组织的成员几乎覆盖了世界上所有的名牌计算机厂商,所以它所建议的标准成为事实上的标准,例如 OSF 的 Motif 已经被工业界公认为一种 API 标准。

如上所述,一般情况下,对 X/Open 标准的增加必须经过 X/Open 中的公司会员(例如 Hitachi、NEC、DEC、IBM、UNIX 国际)的代表所组成的许多工作委员会的批准。但最近 X/Open 决定仿效 1991 年在华盛顿举行的 Xtra 大会上由用户组织直接提供技术的做法,加快纳入 DCE 的批准速度。

最终用户因需要与定时、安全和命名等有关的一套核心系统服务而使从多个厂家得到的开放式系统交互操作时,会遇到许多问题,DCE 就是为解决这些问题而设计的一个环境。一旦批准之后,DCE 就成了 X/Open 可移植性指南(XPG)的一部分,从而扩大开放系统环境的范围。

X/Open 在 1989 年公布了开放式计算系统规范 XPG3,最近又公布了它的新版本 XPG4。

除了 POSIX 和 X/Open 制订的标准之外,在工业界还有一些产品虽然没有被某个标准机构确认为标准,但却已经被工业界广泛认可,被视为事实上的标准。例如由 ACE 的主要厂商 SCO、USL、Compaq 等共同制定的各种 UNIX 应用软件的公共的编程接口(API),通过它,用户可以将 OSF、AT&T、DEC 和 SCO 的各种 UNIX 版本联系起来,使各家的成千上万的应用软件不加修改就能直接在 ACE 的 UNIX 中运行,最大限度地达到资源共享。

目前 SVR4 和 OSF/1 所支持的标准有 Xenix、Streams、POSIX、XPG3、SVID、BSD、FIPS、VFS、TCP/IP、TLI/S、NFS、X/Open 和 Motif 等。正因它们支持众多标准,所以才成为当前最流行的 UNIX 产品。

开放系统的统一趋势

联合、统一、国际标准是人心所向、众望所归。

OSF 和 UI 两大集团在竞争的同时也一直在寻求联合的契机,为此曾多次协商。但终因集团的利益等原因而未能联合,但双方一直未放弃联合的意向。

1992 年 6 月,USL 在旧金山金门公园公布了 UNIX SVR4.2(也称 Destiny 或 UNIX Lite)。这一事件不仅仅是一个操作系统的产品发布会,它的重要意义还在于发布会上所表现出来的开放系统两大派别之间出现的统一迹象。

人们注意到在发布会主席台上就座的人中有 UI 的竞争对手——OSF 的总裁 David Tory。另外,OSF 的发起厂家 DEC、HP 和其他一些大的软件公司,如 Adobe System、Banyan System、Borland International、Computer Associates International、Lotus Development、Novell 和 Wordperfect 等公司也都出席了发布会表示支持。

在发布会上,USL 的总裁 Roel Pieper 表示:公布 SVR4.2 的目的是用它来统一 UNIX 和开放系统市场。会上做出的几项声明也说明工业界已经放弃了派别斗争,如 USL 和 OSF 已同意成立一个特别的工作部门以快速、及时地开发急需的 UNIX 技术,例如激烈竞争的技术领域——图形用户接口。使 OSF 打入竞争圈的 Motif 一直比 Sun 和 AT&T 开发的 Open Look 更通用。由于在 Sparcstation 上 Motif 的实施已经近 10 个,连 Sun 的用户也得到了这个 GUI,所以 SVR4.2 与 Motif 和 OpenLook 两者兼容。USL 的 MoOLit(Motif Open Look 接口工具包)支持两种 GUI。这样,在 Destiny 的环境下可以支持两种现有的 GUI。

在分布式计算技术方面,USL 则支持 OSF 的分布式计算环境(DCE),另外由于 OSF 的发起厂家向 ISV 投资支持 ISV 开发 OSF/1 平台上的应用程序的很少,OSF 就把接力棒交给了 USL,让 USL 开发中性结构的分布格式(ANDF)技术。

许多客户希望 SVR4.2 成为台式机的 UNIX 版本,而 USL 提供的是一个可剪裁、分层次的产品,它既可以适用于转向 UNIX 的 PC 用户,在经过剪裁之后,又可以适用于专用平台。

因此,SVR4.2 有潜力成为市场最大的操作系统,在 Intel 386 和 486 上的 SVR4.2 供货之后,它就成为了事实上的工业标准。1993 年分别以 IBM RS/6000、DEC Alpha 和 HP-Precision 为硬件平台的几种 SVR4.2 版本将陆续出台。

围绕 SVR4.2 出现的这种团结现象为工业界指出了一条通向标准化的道路。

虚拟现实技术及其应用前景探讨

张 质 洞

一、虚拟现实技术概念

虚拟现实(Virtual reality)技术是一种让使用者通过其感官与计算机交互的接口技术。它能让用户进入计算机所产生的世界,并通过其视觉、听觉、触觉三维地与该世界沟通。用户利用交互式计算机程序,可以如同真实的一样处理计算机系统所生成的物体。用户穿上特殊的服装——一种带有表明人体位置的光纤传感器的衣服或手套和戴上一付带有视觉屏幕和音频装置的头盔,便可置身于三维模拟空间或真实世界的模型中。虚拟现实有各种各样同义名词,如远距临场(telepresence)、人工世界(artificial world)、多感官输入/输出(Multisensory I/O),甚至机控空间(cyberspace)。

在虚拟现实下,人机的交互是通过使用双向数据传输来完成的:与虚拟现实设备连接的光纤和电缆记录用户的动作,并将这些动作信息作为运动数据传送给计算机,计算机则根据这些运动数据修改模型中的图形。例如,用户把头转向右边,计算机便产生一种景象左移的情景。这个新的情景信息又立即返回到用户的头盔中,使在他面前呈现出与其动作同步变化的景象和声音的世界。这种人机交互的信息将持续不断地变换更新。

二、虚拟现实领域研制概况

目前,美国等国家的许多公司、计算机厂商和研究机构都在虚拟现实领域从事研制工作,其中主要的有:

·Crystal River 工程公司(美国加州)——利用三维 Convolutron 头戴耳机研制了一种先进的虚拟现实声音再生设备。该耳机能精确地全方位地播放录制的声音和音乐。

·Fake 空间实验室(美国加州)——利用其全方位双目监视器研制了一种不同于密封式头盔视频系统的视频装置。

·Telepresence 研究公司(美国加州)——它自己并不开发硬件或软件产品,只是根据客户要求为其做虚拟现实系统的集成工作。

·VPL 研究公司(美国加州)——开发全套硬件系统(包括头盔、手套、Convolutron 耳机以及

Macintosh 工作站等),售价从6万至25万美元不等。

·W 工业公司(英国)——研制了一种称做为虚拟拱廊(Virtuality Arcade)的设备,可用来玩 Dactyl 魔鬼的游戏,这是第一个在市场上大批量销售的虚拟现实设备。

·Apple 计算机公司——正利用其 Macintosh 计算机进行虚拟现实的模型建立和成像的研制工作,其主要兴趣放在接口的改进上。

·DEC 公司——正在从事利用软件实现虚拟世界原型的研制工作。

·IBM ——正从事一项称为“真实用户接口”(Veridical User Interface)的计划,进行接口和程序方面的研制,以实现虚拟世界。

·麻省理工学院——主要从事硬、软件开发和触觉反馈的研究。它正在 Minsky 的媒介实验室全力制造“虚拟砂纸”,这实际上是一种触觉,好像带在手套内的手摸到了粒状的粗糙表面。

·西雅图华盛顿大学——从事软件与硬件各个方面的研制工作。

·北卡罗来纳大学——受投资者委托进行软件建模方面的研究

三、虚拟现实技术的应用

虚拟现实技术的实际应用目前主要在电视游戏等娱乐业中,在工商业中的应用则主要在故障检测、产品设计和数据模型化方面进行尝试。然而从目前的研究与发展看,在工商业中也有着光辉的应用前景。下面是一些组织在进行的若干应用开发工作。

1. 信息系统方面

目前虚拟现实技术及其能让使用者以多种方式观察数据的能力正在传统计算环境下的数据库和网络领域中进行试验。

随着数据系统的不断发展,像窗口、图符和菜单等这样一些现有界面技术将远不能满足用户需求。目前来说,只通过计算机的打印输出和显示屏来了解数据库,是很难了解字段间的相互关系,尤其是在有多种关系和需要互相参照的场合。为此,1991年,一些组织如 VPL 研究公司、华盛顿大学的人机接口实验室(Human Interface Technology Laboratory)和 DEC 公司着手研究对数据世界的“物理”交互作用。

他们的想法是,如果用户能如同与日常事物那样与数据进行交互,则可以减少培训、加深对数据结构的认识和简化对特殊应用的操作。

研究工作主要集中在通过使用虚拟现实头盔与数据产生物理连接,使数据库管理人员(或其他人员)把数据库看成为一个三维模型,以这种方式来了解数据库中数据的内部联系。管理人员不必具备数据库应用方面的专门知识。例如,他可以掌握对某项工作至关重要的、具有一定经验的职员的数据,把它们分类存储在数据库的一个区段中。

VPL 研究公司和 Telepresence 研究公司的研究人员正在探索如何利用虚拟现实技术来保障数据的质量与完整性,如把数据和数据流同色彩、声音和运动(转动或跳动)等联系起来,实现向管理人员的故障报警。如,把跳动编制成程序,使它表示不仅在数据中而且在数据结构中出现的异常现象。这样,经过快速查找,管理人员如发现某个文件或存储区内出现闪光一样的跳动,他便知道该区域出现了故障。

2. 网络领域方面

在网络领域,东京大学正在开发一种虚拟现实系统的样机,以帮助东京电力公司监视其大规模计算机网络上的信息流。这项工作将延续到21世纪。

他们正在探索在计算机生成的网络上通过色彩和声音来向用户提供故障报警的可能性,而不必在实际的电缆网络上和在程序代码中查找故障。

美国西部通信公司及其他通信公司也在用类似的网络管理方法进行试验。他们试图对整个通信链路用计算机三维图形模拟显示出来以进行跟踪,一旦发现问题采取相应措施。

3. 产品设计方面

在产品设计领域,利用模拟环境来工作的想法是有吸引力的。在这个领域设计和建立模型要受到空间、时间和经费的限制,因此目前有多家公司在研究采用虚拟现实技术。

SUN 微系统公司(SUN Microsystem Inc.)和波音公司(Boeing CO.)已向位于西雅图的人机接口技术实验室投资,以研究远程设计的可能性,试图实现在不同地点的工程师们通过计算机网络在同一个三维空间共同从事设计。SUN 公司研制高分辨率三维图像,使人机接口更实用,波音公司则设计飞机座舱。

在评审一项新的设计时,连接在同一虚拟现实系统中的一批用户甚至可跨越不同国家同时审查某个计算机生成的产品,各人只要戴上特殊眼镜和手套,便身临其境地对产品设计进行分析、评审。

但目前的技术尚未达到能实时地改变设计计划

的程度。例如,当设计师审查一个程序时,如对设计模型提出修改,这些修改意见必须编入该软件的后续版本中。将来利用虚拟现实技术,这样的修改便可体现在设计中。

对于制药行业,建立模型也很重要。他们在合成产品时,经常要和分子的微观世界打交道。位于 Chapel Hill 的北卡罗来纳大学运用虚拟现实技术帮助制药公司建立产品的化学模型。研究人员配戴手套后,可在计算机生成的分子世界中物理地操作分子结构,将它们或结合或分开,以观察其化学键合,用这种方式开发新的分子实体。

4. 建筑业方面

在建筑行业,虚拟现实技术有其独到之处。HP 公司、北卡罗来纳大学和德国柏林都在从事这项工作。他们在项目动工之前先把平面建筑设计图转换成能修改的三维计算机辅助设计软件,然后,建筑师与客户戴上手套和头盔一起审查建筑模型,对建筑计划作全面的修改。

HP 公司正在使用 VPL 研究公司的虚拟现实设备设计欧洲综合办公大楼;柏林城使用同样的设备设计连接东西柏林的新地铁;北卡罗来纳大学则使用他们自己的程序设计校园内雪铁龙大厦(Sitteron Hall)的计算机科学楼。

虚拟现实技术在建模领域的优越性同样吸引了松下电器公司的一个百货商场。他们利用虚拟现实技术来扩大销售。他们编制了一个虚拟厨房应用程序,它能让顾客在商店为其厨房组合和搭配各种厨具和设施。商店职工将厨房布置图拷贝进计算机系统,然后顾客戴上手套和头盔,走进模拟的厨房,按他们自己的喜好改变颜色、大小,增加设施用具等,直至配置成一个令他满意的厨房为止。这样,对顾客来说,好处是在还没有正式安装之前就可以知道自己是否喜欢;而对松下来说,则可做到现配现售。

四、虚拟现实技术应用中 目前存在的问题

以上只是列举了虚拟现实技术的若干应用尝试,但就目前来说,虚拟现实技术仍很不成熟,其应用中,还存在不少缺点,如:

(1)费用昂贵。一个单人系统配戴一个头盔、一付光纤手套、一个动作传感器和一个带有三维建模软件的工作站,其费用一般至少在5万美元,要想功能更强、设备更优,则其费用还需增加5倍。

(2)设备累赘。目前所用的元器件比较累赘。传

(下转第36页)

一种实用的原型开发支持环境

北京航空航天大学计算机系 姚淑珍

摘要 本文提出了一种实用的原型开发支持环境 PSE (Prototyping Support Environment) 的设计方案。PSE 将重用构件库技术、面向对象技术、联想查找与组装技术结合起来, 提供了从软件需求描述到原型生成的自动化过程。开发者可利用 PSE 快速构造出各个应用领域的软件原型。用户可借助 PSE 提供的模糊评价机制对软件原型进行评价, 提出改进意见。开发者据此不断提高原型版本, 直到满足用户要求为止。

一、前言

目前, 有多种快速原型开发技术, 例如, 可执行规格说明、超高级语言、应用程序生成器、重用构件库技术和面向对象技术。其中, 重用构件库技术是指用已有构件来快速构造出系统原型的一种技术。领域知识以构件形式存放在构件库中, 它们有简单和清晰的界面。并且有高内聚、低耦合的特点。随着所开发软件数目的增加, 领域知识得到不断积累和完善。这种方法中, 领域知识的表示是一个很关键的问题, 面向对象方法为此提供了很好的描述手段。在面向对象方法中, 表示单个实体的对象由数据和操作这些数据的过程组成, 对象间通过消息进行通讯, 每个对象都属于一个类, 类有继承性质。面向对象方法描述实体时所体现的自然性、模块化、共享性和信息隐蔽性, 使得用类表示知识成为可能。在类的等级结构中, 用子类表示较特殊的知识, 用父类表示较一般的知识。子类可以继承父类的所有特征, 也可以定义自己的特征, 子类又可将这些特征传递到它的子类中。用不同但相关的类的共性可构造超类。这样就可以从已存在的较一般的知识得到较特殊的知识, 将较特殊的知识归纳成较一般的知识。本文提出的原型开发环境 PSE 将重用构件库技术和面向对象技术结合起来, 充分利用以面向对象方法表示的领域知识快速生成符合要求的软件原型, 有效地解决了快速原型开发方法中诸如知识表示、知识重用、方案择优等关键问题。

二、PSE 的总体设计方案

PSE 是基于重用构件库技术的通用的原型开发支持环境, 利用 PSE 开发者可快速构造任何应用领域的原型。该环境包括以下四个功能:

1. 接收以原型语言描述的软件需求;
2. 按需求到构件库中找出构件集;

3. 选择最优的构件集并组装成原型;

4. 评价原型的静态或动态特征。

当用户对所开发的原型不满意而需要修改或完善时, 开发者可根据用户意见是针对需求的还是设计的, 来选择返回到功能 1 或功能 3。

以 PSE 的原型语言描述的软件需求包括两部分: 声明部分和操作部分。声明部分包括对象声明部分和变量声明部分, 对象声明部分用来声明对象的类和应用领域; 操作部分由编程语句和原型语句组成, 原型语句用来实施对象的操作。PSE 的核心是重用构件库。重用构件库方法大体上分为两类: 白盒子方法和黑盒子方法。前者通过修改构件的内部结构来满足需求, 这种构件使用起来灵活性强, 应用范围广, 但开发者必须了解构件的实现部分。后者利用构件外部结构来选择和组装构件, 与前者相比较, 这种方法适于开发者比较熟悉构件功能和对外接口的情况。我们在 PSE 的查找功能中采用黑盒子方法, 原型组装时借以白盒子方法作局部调整。PSE 的构件库中的构件用面向对象中的类来表示, 它包括描述部分和本体部分。描述部分指出表示构件的类的名字、功能说明、子类、属性、操作和约束。操作部分是操作描述字的集合, 它包括操作名、功能说明、参数个数和类型以及操作的返回类型。构件描述部分是构件库的对外接口, 只当组装时才引入构件体。

除构件库外, 系统还有一个同义词典库, 用来存放对象和操作的一系列同义词。PSE 按对象声明部分和原型语句从构件库中特定应用领域的无环有向图中查找构件, 若没找到, 则用同义词替换继续查找。在查找与组装过程中, PSE 提供了联想查找机制和最优组合算法。

三、构件库的组织

为了存储和重用构件, 必须清楚定义每个构件的内部和外部说明, 还必须提供一个构件使用环境。

PSE 构件库的组织方法基于以下考虑:

- 易于描述构件的层次结构;
- 易于维护构件库;
- 易于进行联想查找;
- 易于组装所收集的构件。

PSE 的构件库存储的是各个应用领域构件的等级结构。每一应用领域构件的等级结构构成一个无环有向图。在无环有向图中,两个表示构件的类之间的关系程度用密切量来表示。假定,在无环有向图中,父类到子类边上的加权表示两者间的密切程度,加权越小,密切程度越高。把无环有向图中任意两类间的密切量定义为两类在图中的最小加权距离和。例如,无环有向图中 F_1 有两个子类 S_1 和 S_2 , W_1 为 $\langle f_1, S_1 \rangle$ 上的权, W_2 为 $\langle f_1, S_2 \rangle$ 上的权, 则 $D(S_1, S_2)$ 为 $W_1 + W_2$ 。若 f_1 的父类还有一个子类 f_2 , W_3, W_4 分别为 $\langle e, f_1 \rangle, \langle e, f_2 \rangle$ 的加权且从 f_2 到 s_1 只能通过 $\langle f_2, e \rangle, \langle e, f_1 \rangle, \langle f_1, s_1 \rangle$, 则 $D(f_2, s_1)$ 为 $W_4 + W_3 + W_1$ 。这种定义可扩展到 n 个类 $a_1, a_2 \dots a_n$, 规定 $D(a_1, a_2 \dots a_n) = \sum_{i=1}^n \sum_{j=1}^n D(a_i, a_j)$, 其中 $D(a_i, a_i) = 0$, 这种构件库组织形式有利于构件的联想查找和构件的择优。

四、构件的查找、组装与原型生成

对于在对象声明部分没找到的类,开发者可以自己建立。第一种方法是通过指定新类的父类和局部方法来建立新类。若在库中没有指定的父类,则用同样方法建立父类,一直到所有父类都在构件库中找到为止,否则用第二种方法编写类。对于局部方法开发者可自行设计。当找到的类的功能说明与要求不符时,PSE 联想查找与该类最密切的另一类作为选择项供开发者选择。第二种方法是开发者通过指定类中的数据和函数集自己编写类。PSE 按方法集到库中查找包含指定方法的类。所查到的每一个类可能包含方法集中的几个方法,将包含整个方法集的类归为一组。当库中可能存在不止一组类包含所指定的方法集时,可利用密切量概念选择关系最密切的一组类。假如新类中有方法 $X_1, X_2 \dots X_n$, 构件库中包含有 X_1 的类构成集合 A_1 , 包含有 X_2 的类构成集合 A_2, \dots , 包含有 X_n 的类构成集合 A_n , 则最佳一组父类为 $a_1, a_2 \dots a_n, a_1, a_2 \dots a_n$ 分别属于 $A_1, A_2 \dots A_n$, 且对于分别属于 $A_1, A_2 \dots A_n$ 中的任何类 $b_1, b_2 \dots b_n, D(a_1, a_2 \dots a_n) \leq D(b_1, b_2 \dots b_n)$ 。若这组父类不完全满足开发者要求,PSE 则选择其余组中密切量最小的一组,此过程一直进行下去,直到类中数据和操作都满足开发者要求为止。当方法集中至少有一个方

法在库中没找到相应的类,则开发者可将这些方法作为局部方法自行设计。新类建立完并经过系统测试后,PSE 将新类自动插入到无环有向图中作为所选择父类的子类,并指定父类到子类边上的加权。

开发者利用上面方法声明或建立完所有类及对象以后,就要用对象间发消息完成该软件的功能。此时还会遇到在所声明的类中没有所用方法的情况。有两种解决方法:第一,利用上面求密切量的机制择优查找含有所用方法的一组类连同所声明的类一起生成新类;第二,开发者自行设计所用方法。在选择好构件集后,就要对构件集进行组装,生成原型。对于操作部分生成的新类,应加入到声明部分。在构件组装过程中,对于所查找到的构件,开发者可决定是全部重用还是部分重用,即开发者可将构件看成一个白盒子进行局部调整。然后进行编译,运行原型。最后用户可模糊评价最终原型的动态或静态特征。

五、模糊评价

人类的思维活动具有两方面特征:其一是直觉和严密性的有机结合,决定了它具有模糊性的一面;其二是推理过程,其形式化的特点使得计算机处理这类问题成为可能。但人类智慧与计算机功能间有着本质的区别,人们借助数字计算机用以往的经典数学来评价某一事物就显得力不从心。与思维特征相对应,人的评价条件往往具有模糊性。例如:“软件内聚强”,“软件规模适中”等。其中“强”与“适中”就是模糊概念。PSE 为此设置了模糊评价机制。根据评价人数的不同可选用单用户模糊评价方法和多用户模糊评价方法。

1. 单用户模糊评价方法

如果用集合论来描述概念,将集合的定义视为内涵,组成集合的所有元素视为外延,则对于诸如“强”、“适中”等模糊概念没有明确的内涵和外延,也就没法用普通集合论来加以描述,为此我们引入了模糊集合概念。

设给定论域 U, U 到闭区间 $[0, 1]$ 的任一映射 $\mu_A: \mu_A(u)$ 都确定 U 的一个模糊子集 A, μ_A 叫 A 的隶属函数, μ_A 叫作 u 对 A 的隶属度,它表示元素 u 属于模糊集合 A 的程度。集合 X 到集合 Y 的一个模糊关系 R 是直积空间 X 到 Y 的一个模糊子集。被定义了隶属函数运算规则的否定词、联结词和程度副词叫作模糊算子。其运算规则定义如下:

- 作用了否定词“非”之后的隶属函数为: $\mu_{\bar{A}} = 1 - \mu_A$
- 作用了联结词“或”、“与”之后的隶属函数分别

为:

$$\mu_{A \cup B} = \max(\mu_A, \mu_B) \quad \mu_{A \cap B} = \min(\mu_A, \mu_B)$$

·作用了程度副词“很”、“比较”、“稍微”之后的隶属函数分别为:

$$\mu_{\text{很}A} = (\mu_A)^2 \quad \mu_{\text{比较}A} = (\mu_A)^{0.75} \quad \mu_{\text{稍微}A} = (\mu_A)^{0.25}$$

明确了以上定义后,我们可以将模糊语言变量定义为六元组(x,T(x),F(x),U,G,M),其中:x是语言变量,T(x)是用模糊集合表示的模糊量的集合,F(x)是用模糊关系表示的模糊符的集合,U是论域,指定语言变量的取值范围,G是语法规则,指定模糊评价条件的构成,M是语义规则,指定T(x)及F(x)的隶属函数.PSE建立一个包括模糊语言变量和模糊算子的模糊语言库,用户以此给出模糊评价条件,PSE将评价条件转换成语言变量(x,T(x),F(x),U,G,M)和模糊算子的联合表示,进行模糊评价.例如,对软件规模进行评价时,x为软件规模,T(x)={大,中,小},F(x)={远大于,远小于,近似于},U={i|1000000>i>0,i为整数}.按语法规则G构成评价条件“软件规模不很大”,这样系统就可根据语义规则M中指定的“大”的隶属函数求出评价值.

2. 多用户模糊评价方法

在这种方法中,首先确定评价指标 $U = \{u_1, u_2, \dots, u_n\}$ 和评语集 $V = \{V_1, V_2, \dots, V_m\}$,然后用模糊矩阵

$$R = \begin{bmatrix} r_{11} & r_{12} & \dots & r_{1m} \\ r_{21} & r_{22} & \dots & r_{2m} \\ \dots & \dots & \dots & \dots \\ r_{n1} & r_{n2} & \dots & r_{nm} \end{bmatrix}$$

来表示多用户对各评价指标的每一评语的百分比.利用模糊合成运算 $A * R$ 求出 $B = (b_1, b_2, \dots, b_m)$, $b_j = \sum_{i=1}^n a_i r_{ij} (j=1, 2, \dots, m)$ 作为多用户评价结果.例如, $U = \{\text{质量, 规模, 运行效率}\}$, $V = \{\text{很好, 较好, 一般, 差}\}$, 对质量评价时,有50%的人认为很好,40%的人认为较好,10%的人认为一般,没有人认为差,则对质量的评价为(0.5, 0.4, 0.1, 0.0),类似地得到对规模的评价为(0.4, 0.3, 0.2, 0.1),对运行效率的评价为(0.0, 0.1, 0.3, 0.6),这样就得到一个评价的模糊矩阵:

$$R = \begin{bmatrix} 0.5 & 0.4 & 0.1 & 0 \\ 0.4 & 0.3 & 0.2 & 0.1 \\ 0 & 0.1 & 0.3 & 0.6 \end{bmatrix}$$

在指定了评价中三个评价指标的权向量 $A = (0.5, 0.2, 0.3)$ 之后就可以求出向量B,归一后得(0.33,

0.29, 0.18, 0.20),这表明有33%的用户认为软件很好,29%的用户认为较好,18%的用户认为一般,20%的用户认为差.

在利用以上两种方法进行评价后,用户认为原型不满足要求时,开发者可对其做局部调整或重新设计.此时可借助PSE自动选择除本方案以外的另一最佳方案.

六、PSE 的实现考虑

纵观编程语言,C++体现了面向对象的思想.C++的类包括三个部分:私有的、保护的和公有的.私有部分说明的数据和函数在类的外面不能被访问,只能在类中被说明函数访问.保护部分说明的数据和函数除了从这个类派生的子类外,其它类不能访问.公有部分说明的数据和函数在这个类的外面可被访问.C++类的这种表示形式使得领域知识组织形式更为灵活.可将局部知识说明为私有类型,领域的内部知识说明为保护类型,领域的对外知识说明为公有类型.C++的虚拟函数和友元机制也有利于知识的表达.同时C++又是一个混合语言,它既溶合了面向对象的能力,又保留了传统的结构化语言特征.C++丰富的描述和处理能力使它比其它语言更适于PSE的实现.所以PSE选用C++作为PSE开发语言和原型设计语言,这样将更好地发挥PSE在设计上的独特之处而且可以保证原型生成环境与运行环境的统一.

七、结 论

软件生存期的提出旨在解决80年代出现的软件危机.目前,软件界普遍认为生存期正面临着新的挑战,并试图用原型模型来取代传统生存期模型,但多侧重于理论探讨.本文所提出的原型开发环境PSE是一个实用的原型开发支持环境.由于它易于进行重新设计和局部调整,所以适于开发任何应用领域的抛弃式原型、演化式原型和递增式原型.对于PSE,领域知识的整理和领域知识库的构造是一个长期的反复测试、扩充和修改的过程.如何合理地管理领域知识是使用PSE的关键.建议对构件库的维护要有严格的把关手段,例如质量保证、宏观预测等技术.构件库中构件间密切量的确定、单用户模糊评价中隶属函数的设置和多用户模糊评价中评价指标权向量的选定含有大量主观因素,如何发挥经验数据的指导作用还有待于进一步研究.

实用监控系统控制端局的软件设计

上海交通大学 宣东

摘要 本文是在笔者最近编制的一个实用软件基础上整理而成的,主要涉及了当前比较流行的下拉式菜单设计,计算机间通信,ISR等问题,提供了在西文DOS下适于菜单图像的汉字显示程序,在图形方式下类似于文本方式下工作的scanf()的g-scanf()程序,通信程序以及ISR实例,以上程序均用Turbo C 2.0编制。

最近,笔者为本市某重点工程的一个配套项目——路面应力监控系统编制了一个实用软件。这个所谓的“路面应力监控系统”由几十个监控站(CPU为8031)和一个控制端局(AST286)组成,呈总线型。各监控站负责监测所属路段的应力参数,控制端局负责管理整个系统。笔者负责编制控制端局软件。

为了具体实现控制端局的系统管理功能,笔者编制了系统复位、循环采集各站、若干站采集三个功能块。各功能块又包括确认、打印、出错告警等功能。这三个功能块都是建立在控制端局与各监控站通信的基础之上,所以本软件的通信程序块是整个软件功能的基础。因为是面向一般的操作人员,软件提供了一个操作提示一律汉字显示的菜单系统作为用户界面。为了能在异常情况下紧急退出,本软件引入ISR技术。

本软件在西文DOS下,用Turbo C 2.0编制。在具体的编制过程中,主要面临怎样编写一个面向一般操作人员的友好用户界面,怎样与几十个监控站进行可控通信以及编写可靠的ISR等问题。下面就这三部分加以阐述。

一、用户界面——菜单系统设计

作为一个管理软件,有一个友好的用户界面非常重要。尤其是在本软件的应用场合,软件的功能执行需由操作人员按监控系统的实际工作情况而选择。为此,笔者仿Window Turbo C的集成环境编制了一个滑块式菜单系统作为用户界面。

这种界面格式在屏幕顶部有一个水平菜单,中央是显示区,底部是功能键。顶部每项有一个对应功能的子菜单。操作人员移动顶部的一个滑动矩形光标进行选择,确认后对应选择块加亮,弹出相应子菜单。这种菜单技术的好处是它占用最小的屏幕空间,而使应用程序显示的绝大部分保留在操作人员的视野中。

1. 菜单系统结构

典型的下拉式菜单系统由四部分组成:菜单初始化、菜单系统驱动函数、菜单显示读取选择函数、具体应用函数。另外还定义一个结构数组,存放各菜单的有关信息,例如菜单图像内容、存放的内存地址、激活标志、热键定义等。菜单驱动函数是整个菜单系统的核心,它按照用户选择决定弹出哪个子菜单以及调用哪个应用函数。菜单显示读取选择函数用于显示指定编号的菜单、读取用户选择及返回选择项编号。整个菜单系统的工作过程就是驱动函数的反复调用、显示读取选择函数、按返回值具体执行应用函数的过程。

这种结构特别适于选择项较多、各级子菜单深度整齐的大型菜单系统。当然在实际菜单系统编程时,如果菜单系统选择项不多,各级子菜单深度参差不齐,就不一定要定义统一的结构数组,也不一定把菜单驱动和菜单显示读取选择分开。在本软件中,笔者把菜单驱动和菜单显示读取选择并在一起,放在主程序中。软件执行时,首先弹出主菜单,接收用户选择。主菜单有三个选择项,分别对应复位、循环采集各站和若干站采集。菜单驱动程序开辟了一个一维数据存放选择块标号,将三个选择块构成一条链。操作人员选择后,根据标号转入相应的功能块,弹出相应子菜单,或执行相应应用函数。

2. 菜单图像设计

为了在菜单系统中用汉字表示选择项以及显示采集数据的需要,本软件在图形方式下工作。Turbo C 2.0提供了丰富的图形库函数。下面先介绍在本软件菜单图像制作中用到的图形库函数,然后介绍显示中文和图形方式下交互的实用函数:

(1) Turbo C 图形库函数

·initgraph(), graphresult(), setgraphmode(), closegraph()等。

用于图形初始化、模式选择以及图形库关闭。本

程序在配备了 SVGA 的 286 上运行,故选择 VGA 方式 2,即像数为 640×480,调色板有 16 种颜色。

·setfillstyle(), bar() 等

用于在指定位置、指定的颜色及填充模式画条块。可用于制作选择块及菜单图像背景、菜单黑影。

·setcolor(), vrectangle(), setlinestyle(), line(), line to() 等

用于按指定位置、指定颜色及画线模式画框及线。可用来画选择框、菜单边框及数据显示表格。

·outtextxy() putpixel()

outtextxy() 和 sprintf() 结合,可以进行一般字符显示。用 putpixel() 显示特殊字符。下面要介绍的汉字显示实用函数就用到了这个函数。

·image_size(), getimage(), putimage()

image_size() 与 malloc() 合用可以用来申请被保存图像的所需内存大小。getimage() 和 putimage() 用于屏幕图像保存、重现,以达到良好的屏幕视觉效果。

(2) 两个实用函数

·write——chinese()

用户使用以上罗列的库函数已经可以编制出非常漂亮的图像,但不能直接用来显示汉字。现在汉字显示的方法很多,有一种方法是在中文 DOS 下,用 Turbo C 编程,通过设置 conio.h 中 directvideo=1,就可以在图形方式下显示汉字了。但这样做要进入中文状态,何况汉字间距、显示位置、大小较难控制。为了在西文状态下显示菜单提示的汉字,本软件编制了一个能在指定位置、间距、颜色等参数下显示汉字块的实用程序。用本函数在本软件中制作选择条块、操作提示块、数据显示块等。源程序如下(程序 1):

程序 1

```
#include <stdio.h>
#include <graphics.h>
#include <bios.h>
#include <string.h>
extern long word[50][30];
write_chinese(number, topx, topy, widthx, heightx,
              wordx, wordy, farx, bcolor, bstyle, wcolor)
int number, topx, topy, widthx, heightx, wordx, wordy,
    farx, bcolor, bstyle, wcolor;
{
FILE * stream;
unsigned char buf[16][2]
int i, j, k, n, volx, liney, size;
if ((stream=fopen("hzk16", "r+b")) == 0){
    printf("can't fopen hzk16");
    exit(1);
}
setfillstyle(bstyle, bcolor);
```

```
bar(topx, topy, topx + widthx, topy + heightx);
for(k=0; word[number][K] != 0; k=k+2){
if (fseek(stream, (94 * (word[number][k] - 1) +
    word[number][k+1] - 1) * 32,
    SEEK SET) != 0){
    printf("fseek call fail");
    exit(1);
}
volx = topx + wordx + (16 + farx) * k/2;
liney = topy + wordy;
for(i=0; i<16; i++){
for(j=0; j<2; j++){
    buf[i][j] = fgetc(stream);
for(n=0; n<=7; n++){
    if(buf[i][j]&128)
        putpixel(volx+n, liney, wcolor);
    buf[i][j] = buf[i][j]<<1;
    volx = volx + 8;
}
    liney = liney + 1;
    volx = volx + 16;
}
}
fclose(stream);
return;
}
```

主要编程思想:是根据 16×16 汉字库点阵存放规律编写汉字在汉字库的位置,称为区位码。每个汉字的点阵信息占 32 个字节。本软件中,选择条块的显示汉字的区位码存放在 word[][] 中,函数 write_chinese() 的参数 number 为欲显示选择块汉字在 word[][] 的编号。函数用 fseek() 根据汉字区位码定位,用 fgetc() 读取一字节点阵信息,然后对 8 个 bit 逐位判别,若为 1 用 putpixel() 在相应位置画一个像素。通过两个循环完成一个汉字的显示。最外层循环用于显示汉字块。

不足之处是事先需要查得欲显示的汉字的区位码放在 word[][] 中,这对于显示较少汉字菜单并不非常苛刻。

·g_scanf()

前面已经提到图形方式下一大缺陷是没有文本方式下如 printf(), scanf() 的丰富输入输出函数。本软件的若干站采集功能块弹出交互窗口,要求操作人员输入起始、终止监控站号。为了实现这个功能,编制了类似 scanf() 的 g_scanf() 源程序如下(程序 2):

程序 2

```
#include <stdio.h>
#include <stdlib.h>
#include <graphics.h>
#include <bios.h>
#include <string.h>
#include <stdarg.h>
```

```

#define ENTER 0x1c0d
#define LARROW 0x4b00
int g__Scanf(int x,int y,int bcolor,int wcolor,int wfar,
             int len,char *format,...)
{
    Va__list varPP;
    int *k;
    float *f;
    char *c,*s
    int i=0,key;
    char *p,msg[80],show[2];
    char *int__s="+-01234567890",*float__s="+-0123456789.";
    va__start(varPP,format);
    setfillstyle(1,bcolor);
    setcolor(wcolor);
    if((p= strchr(format,'%'))==NULL) return(-1);
    for(;;){
        key= bioskey(0);
        if(key==ENTER)break;
        if(key== LARROW&&j!=0){
            bar(x+(i-1)*(8+wfar),y,x+i*(8+wfar),y+8);
            i--;
        }
        if((key&0x0ff)>0x20 &&(key&0x0ff)<0x7e&&(i<len-1)){
            msg[i]=key&0x0ff;
            sprintf(show,"%C",msg[i]);
            outtextxy(x+i*(8+wfar),y,show);
            i++;
        }
        msg[i]='\0';
        switch(* (P+1)){
            case'd':if(strspn(msg,int__s)!=i){
                bar(x,y,x+(8+wfar)*len,y+8);
                return(-1);
            }
            k=va__arg(VarPP,int*);
            *k=atoi(msg);
            break;
            case'f':if(strspn(msg,float__s)!=i){
                bar(x,y,x+(8+wfar)*len,y+8);
                return(-1);
            }
            f=va__arg(VarPP,float*);
            *f=atof(msg);
            break;
            case'c':c=va__arg(varPP,char*);
            *c=msg[0];
            break
            case's':s=va__arg(varPP,char*);
            strcpy(s,msg);
            break;
            default:break;}
        va__end(varPP);
        return(0);
    }
}

```

编程思想:在实际工作环境中,输入显示作为屏幕图像的一部分,所以必须把显示的位置,字符颜色、间距、字符的最大个数和类型说明,返回地址一

起传递给本函数。函数首先用 bioskey() 读取显示字符,放入 msg[] 中,并显示。然后按 format 的类型定义转换成相应类型,返回。

为了传递不定类型的参数,函数引入了 va__list 定义及相关的 va__start(),va__arg(),va__end() 函数(在 stdarg.h 中)。本函数还用到了 strchr() (在字符串中寻找指定字符)、strspn()(s₁,s₂)(判别 s₁ 中有否不包含于 s₂ 的字符)以及 strcpy()(用于字符串之间的拷贝)等库函数(在 string.h 中)。

不足之处是每次只能单个输入,而且限于四种类型,功能不如文本方式的 scanf(),当然经过一些改进,功能很容易加强。

3. 其他

(1) 键盘操作·bioskey()

本软件读取键盘一律用 bioskey(),在菜单系统中常定义一些特殊键作为功能键。不能用 getch() 读取,只能用 bioskey() 的 0 号功能读取键值的扩展码。另外,可以用 bioskey() 的 1 号功能检测是否有输入键,这在程序自动执行中 useful。

(2) 定时·biostime()

本函数读或设置 BIOS 计时器,该计时器以每秒 18.2 次滴嗒计时。本软体中功能块自动反复执行时,为了给操作人员干预提供机会以及在下面提到的通信程序中为了防止线路中断引起的长时间等待,都要使用这个函数。具体使用参见后面的通信程序。

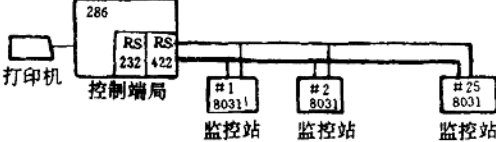
另外,可用 sound(),fprintf() 等告警和制作打印文件。

二、通信程序设计

引言中已提及,通信程序是本软件实现三个功能块的基础。它负责与各监控站进行通信,发布各种命令,接收各站采集数据。

1. 通信协议

本软件在一个由几十个单片机(8031)和 AST286 组成的监控系统中工作,系统组成如图:



作为一个实用的微机系统,本系统采用主从式通信,即只有控制端局有权发送命令。因为考虑到系

统工作环境比较恶劣,系统端间距离为2km,故采用RS422作为异步串行通信口,来提高可靠性。通信协议主要内容如下:

- 系统通信媒介为同轴电缆;通信方式为异步通信;通信速率为1200波特,数据格式为1位起始位,8位数据位,1位停止位。

- 系统暂定为25个监控站,每站数据为51个字节,FF作为起始标志,采用BCD码。

- 系统各站编号,兼作数据发送命令,另设置复位、初启命令。

- 控制端局呼叫监控站,监控站以本站地址原码作为回应,然后作出相应动作,若超时,控制端局应重复三次呼叫,无原码响应,表明系统出错,出示告警信息。

2. 通信程序编制

控制端局和各监控站为通信的双方,各有自己的通信程序。监控站通信程序主要通过串行通信控制位SM₂进行呼叫地址判别,相符作出相应动作。下面主要介绍一下本软件的通信程序。

从监控系统示意图中可以看出,对于AST286来说可以把RS-232和RS-422看成是一个RS-232。RS-232的输入、输出用于与8031通信,另外线接适当电平。

Turbo C 2.0提供了bioscom(),具体执行各种Rs-232通作功能。利用这个函数编制的通信程序如下:

程序3

```
#include<stdio.h>
#include<bios.h>
#define COM1 0
extern unsigned char data __buf[51]
int communication(current __station)
int current __station;
{
int i=0,j;
long time;
register int in,nodata=1,error __rcv=0,error __snd=0;
unsigned char out;
out=(unsigned char)current __station;
for(j=0;j<3;j++,error __rcv=0,error __snd=0){
if(!bioscom(1,out,COM1)&0x8e00){
if(out==0x2f)return(1);
time=biostime(0,0L);
for(;(biostime(0,0L)-time)<36,){
if(bioscom(3,0,COM1)&0x0100){
if(nodata==1){
in=bioscom(2,0,COM1);
if(in&0x8e00)break;
if((in&0x0ff)==out)nodata=0;
else break;}
```

```
else{
in=bioscom(2,0,COM1);
if(in&0x8e00)error __rcv=1;
data __buf[i]=in&0x0ff;
if(i==50){
if((data __buf[0]!=0x0ff)||(data __buf
[50]!=0x0ff)
||error __rcv)
break;
else return(1);}
i++;}
}
}
else error __snd=1;
}
if(error __snd||error __rcv)return(-2);
else return(1);
}
```

编程主要思想:本程序的入口参数是AST286与通信的监控站站号。在主程序中用bioscom()的0号子功能对串行口初始化。即bioscom(0,0×83,COM1),在本程序中,用1号子功能发送数据,并检测。发完数据,调用biostime()开启定时器,然后进入接收段。用bioscom()3号子功能,检测有无数据。用2号子功能接收数据,接收过程中如出错置标志。按照通信协议,第一个接收字节为监控站站号的原码。判别后,若是,就请nodata标志,准备接收数据;否则重新呼叫。接收的一站数据放在data__buf[]中,接收完判定头尾是否是FF。是,返回1;否,则重新呼叫,接收过程超时,重新呼叫,三次呼叫结束返回出错标志。

对于复位命令29H,各监控站收到后将重启,故不进行原码确认,通信函数直接返回。

在编制通信程序以前,应将RS-232的20,8,6线接高电平,否则bioscom()的1号和2号子功能将失效。这是因为bioscom()实际调用中断14。中断14在接收和发送数据前总要调用等待响应段监测6,8线是否置高电平,超时返回出错信息,串行口就不能接收和发送数据。如想绕开20,8,6线高电平的麻烦,可用outputb(),inputb()读取接串行口的状态寄存器及向输入输出读写,而避开中断14的对6,8线的监测,同样能实现可靠的通信。

三、中断服务例程(ISR)设计

本软件提供两个出口,一种是在主菜单接ALT-F₁退出,一种是在任何情况下按ALT-F₂异常退出。为了设置第二个出口,本软件引入3ISR。

在Turbo C 2.0中提供了编制ISR的机制。编程

人员可以全部用 C 语言编制成运行情况可靠的 ISR。Turbo(包含特殊改进类型 interrupt, 如想把一个函数充作 ISR, 可用如下说明:

void interrupt newkb(); (newkb() 为欲当作 ISR 函数)

另外, Turbo C 2.0 还提供了 getvect(), setvect(), 用于取中断向量和设置中断向量。本软件中涉及的 ISR 程序段罗列如下:

程序4

```
void interrupt (* oldtimer)();
void interrupt (* oldkb)();
void interrupt (* olddisk)();
void interrupt newtimer();
void interrupt newkb();
void interrupt newdisk();
unsigned char scan __code = 60;
unsigned char key __mask = 8;
static unsigned dos __seg; dosbusy;
static union REGS rg;
static int active = 0, ready = 0, disk __busy = 0;
unsigned char code; /* 类型说明 */
    rg.h.ah = 0x34;
    intdos(&rg, &rg);
    dos __seg = __ES; /* DOS 忙标志 */
    dosbusy = rg.x.bx;
    oldtimer = getvect(0x8);
    oldkb = getvect(9);
    olddisk = getvect(0x13);
    setvect(0x8, newtimer);
    setvect(9, newkb);
    setvect(0x13, newdisk); /* 置中断向量 */
void interrupt newkb()
{
    if(inportb(0x60) == scan __code){
        code = peekb(0, 0x417);
        if(((code & key __mask) ^ key __mask) == 0){
            code = inportb(0x61);
            outportb(0x61, code | 0x80);
            outportb(0x61, code);
            outportb(0x20, 0x20);
            if(! active)
                ready = 1;
            return;
        }
    }
    (* oldkb)();
}
void interrupt newtimer()
{
    (* oldtimer)();
    if(ready && (peekb(dos __seg, dosbusy) == 0))
        if(disk __busy == 0){
            outportb(0x20, 0x20);
            ready = 0;
            goodbye();
        }
}
```

```
}
void interrupt newdisk()
{
    disk __busy++;
    (* olddisk)(); /* 修改 int8, int9, int13 */
    --disk __busy;
}
setvect(0x8, oldtimer);
setvect(9, oldkb);
setvect(0x13, olddisk); /* 恢复中断向量 */
```

编程主要思想:是修改读键盘中断9, 使得先测试键值与已定义热键是否相同。如相同置 ready 返回。否则执行原中断9。然后修改中断8, 使先执行中断8, 测试相应标志, 满足条件, 执行退出函数。为了避免中断磁盘操作, 本软件修改了中断13, 加上了忙标志。这样在任何情况下, 一旦按一下热键, 中断9就置标志, 系统每 $\frac{1}{18.2}$ 秒调用中断8检测标志, 若标志已置位, 即执行退出函数。

此外, 我们知道, DOS 是不可重入的, 即: DOS 被一个程序使用时, 不能被另一个程序使用。如果能避免使用 DOS 中断最好。但本软件涉及内存分配, 文件管理等都要用于 DOS 中断, 所以只能在 DOS 运行时, 避免中断。DOS 功能 0x34 向 DOS 维持一个特殊标志, DOS 忙时置位, 否则清零。在中断8, 检测 DOS 忙否。这也是为什么不在中断9调用 goodbye, 而放在中断8调用的原因, 否则将导致热键按下无效。

本软件的 ISR 涉及了中断8、中断9、中断13和 DOS 中断 0x34, 对于编制 ISR 程序已经足够了。但是如果想把 ISR 升级为 TSR, 这又涉及到另外一些问题, 因为 TSR 往往在 DOS 状态下激活, 而 DOS 一般来说是不可重入的。随着 DOS 一些原未公开的资料的不断出现, 人们发现: DOS 并不是完全不可重入的, 当 DOS 执行 0~12 低号功能时, 使用其他 DOS 功能是安全的。反过来, 当 DOS 执行其它高功能时可以调回 0~12 功能。所以在有些场合虽然 DOS 忙, 但可中断。这里还用到中断 0x28, 这就是 DOS 编制假脱机打印程序的原理。这一机制对编制终止并驻留程序 (TSR) 有用。因为本软件 ISR 不必驻留, 故没有用到这一机制。

本软件涉及现在比较流行的菜单, ISR 设计, 也涉及到计算机通信, 而且本软件在常用的 Turbo C 2.0 上编制, 没有用到其他工具, 具有开发平台简单的优点。对于编制类似软件的读者或许会有一些参考价值。

但是, 作为一个实用软件, 有其许多不足, 笔者在各部分已有提及, 有待改进。

IBM/PC 编辑式智能事务管理系统 EIMS

石油物探职工大学 章登植

摘要 本文从用户观点分析当前流行的 dBASE 数据库管理系统的不足,提出一种设计数据库管理系统的新方案,并介绍据此编制成的一种新的单用户数据库管理系统——编辑式智能事务管理系统(简称 EIMS)。该系统是一种智能型的第四代语言,基本上可使用户摆脱繁琐复杂的编程工作,其编程难度仅与编制 DOS 操作系统的批处理文件相当,不存在复杂的算法问题,操作方式则与全屏编辑相似,因此极易为非计算机专业的一般管理人员所掌握。在 EIMS 功能所及的范围内,使用该系统编制一个专用事务管理系统,其效率比用当前流行的 dBASE 提高 10 倍以上。

编辑式智能事务管理系统(EIMS)是作者从用户观点分析当前流行的 dBASE 数据库管理系统的不足而编制成的一种新的单用户数据库管理系统。它总共只有 20 条用于事务管理方面的命令(不包括全屏编辑命令)和 3 条用于编程需要的命令。前者,每条都有强大的功能和良好的通用性,能独立解决事务管理中一个方面的问题,再通过命令文件将带有不同参数的有关命令组织起来(显然,这样的命令文件是很短的,少则几句,多则几十句就能解决一个相当复杂的问题)。然后,再通过菜单文件将有关命令文件组织起来就成为一个专用的事务管理系统。

EIMS 可以通过间接和直接两种方式建立命令程序。所谓间接方式就是常规的程序编写方式,所谓直接方式就是用户根据问题的要求,直接一条接一条地键入并执行该命令。EIMS 能把这些执行过的命令经过优化(滤去那些对命令文件来说无用的命令)后保存起来,最后再执行一条特定的存盘命令就可生成命令程序。

一、EIMS 的功能与特点

EIMS 具有如下功能与特点:

1) 是一个具有功能可与 WORDSTAR 编辑程序相媲美而在使用上要方便得多的中西文兼用的全屏编辑程序。

2) 事务管理功能以全屏编辑为基础,表格数据文件通过编辑方式形成,每个文件由记录构成,每个记录占表格一行,行中字段宽度和字段个数可根据需要随意改变,字段位置也可根据需要相互进行交换,记录长度可达 2160 个字符,字段个数最多为 100,每个文件长度仅受磁盘容量限制,但一次调入

内存部分最多为 32K 个字符或 1000 个记录。

3) 对表格数据文件具有排序、检索、求频度分布、求和及平均值、统计、汇总、联结、制表、报表、过帐、转置、添加、加密、代号、计算等处理功能。它对以记录为单位的行中数据有很强的计算能力,除了能进行代数及四则运算外,还能利用 IBM-BASIC 语言所能提供的许多内部函数。其计算精度达 16 位有效数字。此外,它还能根据表格原始文件构成索引文件和摘要文件。

4) 具有灵活方便的表格打印输出功能,也能将记录行中各字段的数据,按用户要求插入到预先编写好的报表文件中的指定点。

5) 具有批作业的功能,键入一条命令就能连续处理编了号的且文件结构基本相同的一批文件。

6) 运行速度能满足实际需要。以费时较长的“建立索引文件”为例,对 IBM-PC-286 来说,用 55 秒时间就可对 1000 个记录完成操作。

7) 采用了内外排序、二级索引的 B 树技术,在上万个记录中查找一个记录只需三秒钟左右。

8) 对表格数据文件具有按关键字段对记录进行加密、解密的功能。

9) 能自动生成命令文件,调入该文件就能自动且连续地执行文件中所包含的全部命令。

二、EIMS 的基本设计思想

EIMS 之所以有上述特点是因其设计思想突破了 dBASE 的框框。数据处理的特点是:数据量大,时间性强,但计算简单,处理类型不多。尽管事务管理的对象千差万别,但从用 dBASE 编制好的一些应用程序的结构来分析,无非由信息的输入、输出、查询、