

IBMPC/AT

硬件和XENIX系统资料汇编之十五

UNIFY 关系数据库管理系统

程序员手册

汪 农 周保刚

袁晓清 赵智敏

译

孙玉方 校

北京科海总公司培训中心
中国科学院软件研究所

IBM PC/AT硬件和XENIX系统资料汇编
(十五)

UNIFY 关系数据库管理系统 程序员手册

汪 农	周保刚	译
袁晓清	赵智敏	
	孙玉方	校

北京科海总公司培训中心
中国科学院软件研究所

编辑：科海培训中心教材部
发行：科海培训中心资料组
地址：北京2725信箱 科海培训中心
资料组

(北京海淀路332路黄庄站旁)

印刷：河北省蔚县印刷厂

编 者 序

IBM PC已从PC、PC/XT推进到PC/AT。PC/AT以Intel80286为主CPU,具有丰富的硬件资源。鉴于目前DOS系统基本上是一种单用户系统,许多硬件资源未得到充分利用,许多用户都要求在PC/AT上配备多用户多任务的XENIX系统。

XENIX系统是UNIX系统在以Intel为主CPU的微机上的实现,该系统由Microsoft公司开发。目前在PC/AT上运行的XENIX相当于UNIX的System III或System V。

为了更好地在国内推广PC/AT及其兼容机,中国科学院软件研究所在其雄厚的技术力量基础上,积多年研究、开发UNIX系统之经验,开发成功多种XENIX中西文信息处理系统并移植到几乎所有PC/AT的兼容机和部分386机上。为了更好地推广XENIX中英文信息处理系统,科海培训中心和中国科学院软件研究所组织了一批专家和技术人员,收集并编译整理了有关XENIX及IBM PC/AT的全部技术资料。

这些资料包括以下几类:

- I. IBM PC/AT硬件资料,包括硬件安装及组装手册、技术手册和维护手册。
- II. XENIX基本系统的安装、基本用户指南、命令参考手册、系统管理手册和直观 shell手册。
- III. XENIX开发系统方面的软件开发手册、库函数程序员手册、系统调用和子程序手册。
- IV. XENIX系统上运行的汇编语言和高级语言(C、Fortran、Cobol、Basic)的用户指南和参考手册。
- V. XENIX正文格式化手册。
- VI. XENIX系统上配备的最新版本INFORMIX和UNIFY数据库管理系统用户及参考手册。
- VII. 中西文兼容的C—XENIX系统安装和基本使用手册。

全套资料约400万字,分装成20本。

全书的主要译校任务由中国科学院软件研究所的专家、技术人员承担,科海培训中心负责编辑、印刷和发行工作。

由于时间仓促,本资料汇编中必有不少错漏之处,敬请读者批评指正,以便再版时更正。

主编 孙玉方
董洪皋

目 录

第一章 概述	(1)
1.1 怎样使用本手册	(1)
1.2 定义和概念	(2)
1.2.1 术语	(2)
1.2.2 UNIFY 文件	(3)
1.2.3 UNIFY 目录	(4)
1.3 如何编译并装入UNIFY C程序	(5)
1.4 内部及外部数据格式	(7)
1.5 UNIFY C 语言预处理程序	(9)
1.6 UNIFY 程序装入的进一步讨论	(10)
1.6.1 向系统说明程序	(13)
1.6.2 用shell装入程序	(14)
1.6.3 用菜单处理程序装入程序	(15)
第二章 设计一个数据库	(17)
2.1 概述	(17)
2.2 逻辑数据库设计	(18)
2.2.1 规范化的概念	(18)
2.2.2 范式定义	(22)
2.2.3 产生一个规范化的数据库设计	(23)
2.2.4 连接图表	(28)
2.3 物理数据库设计	(29)
2.3.1 UNIFY散列法	(29)
2.3.2 显式关系	(30)
2.3.3 B-树	(31)
2.3.4 缓冲顺序访问	(34)
2.3.5 磁盘存贮空间的考虑	(36)
2.3.6 如何选择访问方式	(38)
第三章 记录函数	(40)
3.1 概述	(40)
3.1.1 数据库函数	(40)
3.1.2 访问函数	(40)
3.1.3 创立/删除函数	(40)
3.2 记录操作	(41)
3.2.1 “当前记录”的概念	(41)

3.2.2 记录封锁	(42)
3.2.3 字段级安全保护	(42)
3.3 记录函数例子	(43)
3.3.1 记录拷贝	(43)
3.3.2 置父亲或长辈记录为当前的	(48)
3.3.3 安全pfield	(52)
3.3.4 安全gfield	(54)
3.3.5 顺序访问记录	(56)
3.3.6 修改数据库记录	(58)
3.3.7 向数据库加入一个新记录	(63)
3.3.8 从数据库中删除一个记录	(66)
3.3.9 用组合键访问一个记录	(68)
第四章 选择函数	(71)
4.1 概述	(71)
4.1.1 选择文件	(72)
4.1.2 标识文件	(73)
4.2 例子	(73)
4.2.1 创建“快速”排序文件	(74)
4.2.2 用unisel创建选择文件	(78)
4.2.3 创建两个选择文件的逻辑与	(82)
4.2.4 创建两个选择文件的逻辑或	(85)
4.2.5 两个选择文件的逻辑与	(88)
4.2.6 创建排序标识文件	(96)
4.2.7 用选择函数联接两个记录类型	(101)
第五章 显式关系函数	(106)
5.1 概述	(106)
5.2 显式关系函数举例	(108)
5.2.1 父-子关系	(108)
5.2.2 祖父-孙子关系	(113)
5.2.3 调用unisort的一个简单例子	(116)
5.2.4 用unisort排序所有记录	(117)
5.2.5 用相关记录中的字段排序	(119)
5.2.6 使用unisort函数表	(122)
第六章 B-树函数	(128)
6.1 概述	(128)
6.2 B-树函数实例	(129)
6.2.1 按序检索记录	(129)
6.2.2 B-树的搜索	(132)
第七章 带缓冲的顺序存取函数	(137)

7.1 概述	(137)
7.2 带缓冲区的顺序存取函数实例	(137)
7.2.1 在一种类型上扫描全部记录	(137)
7.2.2 更新新选的记录	(140)
第八章 终端I/O函数	(143)
8.1 概述	(143)
8.1.1 屏幕表格	(143)
8.2 终端I/O函数实例	(144)
8.2.1 利用屏幕提示进行删除	(144)
8.2.2 利用屏幕表格进行显示	(147)
8.2.3 利用屏幕表格进行更新	(151)
8.2.4 ptube、gtube和数据库字段	(156)
第九章 打印机I/O函数	(158)
9.1 概述	(158)
9.2 打印机I/O函数实例	(158)
9.2.1 基本报表	(158)
9.2.2 打印来自若干记录类型的信息	(160)
9.2.3 pform的用法	(163)
9.2.4 用户函数与pform	(165)
第十章 实用函数	(169)
10.1 概述	(169)
10.1.1 字符串操纵函数	(169)
10.1.2 低层终端I/O函数	(169)
10.1.3 资源封锁函数	(169)
10.1.4 其它函数	(169)
10.2 实用函数实例	(170)
10.2.1 获取数据字典信息	(170)
10.2.2 另一个日期函数	(172)
10.2.3 字符串比较函数	(174)
10.2.4 用元字符比较字符串	(176)
10.2.5 动态地建立数据结构	(178)
第十一章 事务记录函数	(181)
11.1 概述	(181)
11.1.1 事务记录和用户程序	(181)
11.1.2 出错处理	(183)
11.2 事务记录函数实例	(183)
11.2.1 成组更新至一逻辑事务	(183)
11.2.2 读取事务记录	(188)
第十二章 UNIFY裁剪工具	(190)

12.1 概述	(190)
12.2 裁剪ENTER	(190)
12.2.1 操作原理	(190)
12.2.2 裁剪ENTER的实例	(195)
12.2.2.1 裁剪ENTER函数	(197)
12.2.2.2 装入ENTER	(204)
12.2.2.3 多个ENTER可执行程序	(205)
12.3 裁剪RPT	(207)
12.3.1 操作原理	(208)
12.3.2 裁剪RPT的实例	(210)
第十三章 订单维护	(211)
13.1 概述	(211)
13.2 程序设计	(211)
13.3 源程序代码	(217)
第十四章 RM/COBOL接口	(230)
14.1 概述	(230)
14.1.1 记录函数	(230)
14.1.2 特殊的COBOL例程	(230)
14.1.3 内部和外部的数据格式	(230)
14.1.4 COBOL包含文件	(231)
14.2 COBOL例子	(231)
14.2.1 编译SETLOC和UACCESS	(231)
14.2.2 使用组合键访问记录	(237)
14.2.3 使用显式关系	(240)
14.2.4 增加记录	(245)
14.2.5 删除记录	(249)
14.2.6 修改记录	(253)
14.2.7 选择一个组记录	(259)
第十五章 出错消息和一般原因	(265)
15.1 概述	(265)
15.1.1 出错记录文件	(265)
15.1.2 致命错的类型	(266)
15.1.3 UNIFY错误捕捉函数	(267)
15.2 一般错误的类型	(268)
15.2.1 非当前记录	(268)
15.2.2 未打开数据库	(269)
15.2.3 错误的环境变量	(269)
15.3 出错消息汇总	(270)
15.3.1 存贮超限错	(270)

15.3.2 数据字典错	(271)
15.3.3 内部数据库错	(272)
15.3.4 用户错误	(279)
15.3.5 杂错	(280)
第十六章 C 语言接口参考	(282)
16.1 概述	(282)
16.2 宿生语言函数	(282)
第十七章 RM/COBOL 接口参考	(360)
17.1 概述	(360)
17.2 数据类型兼容性	(360)
17.3 用COPY把文件复制到程序中	(360)
17.3.1 HFILECC程序	(361)
17.4 COBOL调用序列	(361)
17.5 记录级I/O函数	(361)
17.6 其它UNIFY函数	(362)
17.7 安装指令	(369)
17.8 用菜单处理程序登记一个COBOL程序	(369)
17.9 实例程序	(370)

第一章 概 述

本手册介绍如何使用UNIFY宿主语言接口编制程序。为了成功地使用这些信息，读者需熟悉UNIFY、基本操作系统命令及程序设计的一些概念。

读者至少应该能使用UNIFY完成下列任务：

- (1) 输入一个数据库设计，理解UNIFY基本数据类型。
- (2) 建立一个数据库。
- (3) 使用PAINT设计屏幕。
- (4) 使用ENTER登记屏幕。
- (5) 使用SQL和ENTER对数据库查询和修改。

读者还应该熟悉《UNIFY参考手册》第一章的内容。若完成上述任务有困难，则应该先学习《UNIFY指导手册》和《UNIFY参考手册》，直到熟悉为止。

除需了解UNIFY以外，读者还应该会使用操作系统的基本命令并理解目录的工作原理。虽然可以边读本手册边补习那些知识，但这将大大影响你的进展速度。下面列出必须掌握的操作系统知识：

- (1) 什么是目录？怎样使用目录？
- (2) 建立、改名、删除、拷贝文件的方法。
- (3) 正文编辑程序的使用方法。
- (4) I/O改向及管道的使用。
- (5) 编译、存档及装入程序的方法。
- (6) 怎样使用打印机进行输出。
- (7) 怎样设置环境参数。
- (8) 使用操作系统的文档功能。

如果读者还不熟悉这些，则需学习一些介绍UNIX操作系统的书，边读边做些上机尝试。

最后，读者还需熟悉下述C语言程序设计概念：

- (1) 数据类型；
- (2) 数据定义文件(include文件)；
- (3) 算术逻辑表达式；
- (4) 函数调用与参数；
- (5) 数组、结构、指针；
- (6) 控制结构(if-then-else, 循环控制)；
- (7) 标准I/O程序包。

读者可以通过读一本介绍C语言的书来复习这些概念。

1.1 怎样使用本手册

UNIFY宿主语言接口包括九组相关的函数：

- (1) 记录函数;
- (2) 选择函数;
- (3) 显式关系函数;
- (4) B-树函数;
- (5) 缓冲顺序访问函数;
- (6) 终端I/O函数;
- (7) 打印机I/O函数;
- (8) 实用程序函数;
- (9) 事务记录函数。

本手册是围绕着这些函数组编写的。第一、二章是为了帮助读者入门。

第一章: 提供有关UNIFY文件的背景知识并给出两个如何编译并装入的简单例子 程序, 以使读者能立即开始运行数据库。

第二章: 讨论如何设计一个能适应需求变化的数据库(逻辑设计)及如何使用 UNIFY 提供的多种存取方式(物理设计)。读者可以在任何时候读这一章, 但每次在设计一个重大应用程序之前, 应仔细重读这一章。

第三——十一章: 讨论各类函数并给出了大量范例。遇到不熟悉的函数时可到这些章节来查阅其用法。

第十二章: 讨论用C函数说明ENTER和RPT的方法。用户可以在数据库的不同层次中加入自己编写的函数来扩展数据库标准版本的功能。

第十三章: 给出一个在单一屏幕上修改多个记录的完整实例。此例为本手册中最复杂的一个例子。

第十六章: 是一个简明手册, 按字母顺序列出了所有函数的名字。当用户已了解了函数的用法而只是想查阅一下其参数和返回值时, 可使用本手册。而第十四、十七章是有关RM-COBOL接口的简明手册。

第十五章: 给出用户可能遇到的典型出错信息及问题诊断方法。

手册最后有一个索引以供查阅手册内容。

1.2 定义和概念

这一节介绍手册中用到的术语以及与UNIFY使用的文件、目录有关的信息。这些内容已在《UNIFY参考手册》中讨论过, 但由于对阅读本手册至关重要, 故在此再重复一遍。

1.2.1 术语

数据库管理领域中有一些专用术语, 在此仅列出本手册用到的一部分常用术语。

字段——以字符形式存储于数据库中的人名、地名、事物、概念等。一个字段有一个名字。字段是数据存取的最小单位。例如, 一个有关人事情况的数据库中, 可能有一些存放社会保险号码、姓名、地址、工资、家庭经济负担等情况的字段。

UNIFY的字段拥有唯一的名字、数据类型和长度。基本数据类型有NUMBER、STRING、AMOUNT、DATE、TIME和FLOAT。还有一种特殊的字段类型COMB, 可用它来将一组字段定义为一个。

本手册中使用的“字段”相当于其它书中的“属性”, “列”之类。

记录类型——由一组逻辑上相关的字段组成并有一个名字。它用于描述那些你想要存放其有关信息的人、地方、事物、概念等。例如，人事数据库可能有雇员、部门、职位等记录类型。

每个UNIFY记录类型至少有一个名字、记录数的估值及一组字段。

本手册中的“记录类型”相当于其它书中的“表格”、“关系”、“文件”之类。本手册的某些地方（特别是第二章）也称记录类型为“表”。

记录——记录类型的一个例。例如记录类型“雇员”有字段“姓名”、“部门”、“工资”。而其一个记录则可能是〈“约翰”、“经营部”、“45000美元”〉。

本手册中的记录相当于别的书中的“元组”。

文件——平时驻于磁盘上的、逻辑上连续的字节序列，由操作系统管理。文件可用来存贮数据库、索引、程序、shell命令程序、文档等。

数据库——由一组记录类型组成，描述并存贮你在现实世界中的应用目标。UNIFY的数据库存放在操作系统的file.db文件中。只要操作系统有可存放的目录，就可以建立任意多个数据库。

数据库设计——说明数据库所存内容的逻辑模型。设计中必须包括对所有字段和记录类型的说明。此处的“设计”相当于其它书中的“模式”、“数据模型”。

UNIFY把数据库设计放在文件“unify.db”中，这个文件还包含屏幕格式、菜单输出、用户、安全保密信息、系统参数等。

相对记录号——在数据库内，记录是通过它在同类型记录中的相对位置而被访问的。记录的相对记录号就是一个记录在数据库中的这种相对位置。由于是逻辑位置，所以记录的实际物理存贮可由一处移往另一处而不会影响任何存取方式。

当前记录——通过UNIFY宿主语言接口，在某一时刻只可以对每一种记录类型中的一个记录操作，而此记录即称为当前的。当前记录是许多宿主语言函数的隐参数。因此控制和选择当前记录是应用程序的主要任务之一。

1.2.2 UNIFY文件

为了编译、装入、运行、调试应用程序，必须知道UNIFY所需的程序文件、数据文件及库文件在哪里。UNIFY用文件名模式来区分应用程序所需的不同种类的文件，文件名形如“名.后缀”。“名”和“后缀”的长度可变，只要总长（包括点.）不超过14个字符。后缀决定文件类型（这与操作系统一样）。

以下列出各种UNIFY文件名后缀及其含义，同时列出一些重要的文件和UNIFY文件所在的目录。

.a——档案文件。由ar (1) 命令生成，以一种适合于ld (1) 命令装配的格式存放可重定位的二进制数据。UNIFY在系统lib目录下有一些系统档案文件。用户也可以自己建立档案文件，一般放在src子目录下。

.c——C语言源程序文件。由用户建立的应用程序，存放在一个src子目录下。

.db——数据库文件。每一个应用有两个数据库文件，一个是数据字典unify.db，另一个是应用数据库文件file.db。当用户首次启动UNIFY时，UNIFY从系统lib下的unify.db原型文件中拷贝一份unify.db。“建立数据库”程序按照用户打入数据字典的数据库设计建立file.db文件。一般这两个文件都放在数据库的bin目录下。

.dbr——“原始”数据库文件，名为file.dbr。也是由“建立数据库”程序建立的。这一文件在每一应用中都有一个。它们或是联到file.db上（若是一个普通操作系统文件被用作数据库），或是指向同一磁盘延展file.db（若是一个文件系统被作为数据库）。这只是访问应用数据库文件的另一种方法。

.h——include文件，与C语言源程序联用，通常在def或include目录下。

UNIFY建立两种.h文件。第一种是file.h，由“建立数据库”或“数据库重整”程序建立。这个文件包含一系列的字段名和记录类型名。这些东西将被C程序用来访问数据库。第二种.h文件是screen.n文件，由“构造屏幕格式”或“编译屏幕”程序建立。它们包含一些屏幕字段名，供使用这种屏幕格式的C程序使用。

.idx——B-树索引文件，是放在bin目录下的二进制文件。它们由“加入、删除B树索引”程序建立，而由那些把数据存入索引字段的程序修改。名为bt_{nnn}.idx（nnn是一个数字）的文件为实际的B-树文件，而文件field.idx是索引文件的目录。

.ld——shell命令文件，它使用操作系统的ld(1)命令生成可执行文件，这些文件通常存放在用户的build目录下。

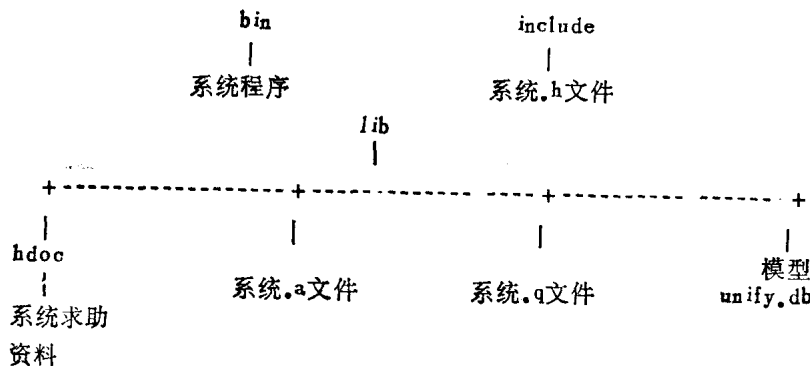
.o——可重定位的二进制文件。由命令ucc-c或cc-c产生，这种文件通常放在档案文件内。

.q——UNIFY屏幕格式文件。文件内容为屏幕格式的二进制形式，供运行时的快速读取。它们由“构造屏幕格式”程序或“编译屏幕”程序建立。UNIFY系统屏幕格式存放在系统lib目录下，而用户的屏幕格式通常存放在数据库bin目录下。

1.2.3 UNIFY目录

UNIFY有两种基本目录类型。一类是不可改变的目录，由UNIFY的所有用户使用，是版本的标准内容。

第二类是每一个应用的（或说用户的）私有目录。这些目录中存放着属于某一特定应用的文件。用户可通过设置shell环境变量而使两类目录都能方便地访问（《UNIFY参考手册》中对环境变量有详细讨论）。以下列出UNIFY系统目录：



这三个目录在你安装UNIFY的那个目录下。运行时只用到bin和lib目录。而include下的文件则在编译程序时使用。UNIFY的启动程序unify不在这个结构内，可以选择一个方便的目录存放这个文件以便你能在不修改环境或.profile文件的情况下启动UNIFY。

下面是一个典型的用户目录结构的例子（xxx, yyy是src子目录名）。

下一节讨论数据库与程序之间数据类型的匹配问题。

1.4 内部与外部数据格式

当你执行上节的start程序时，终端上将打出14对数字，它们是数据库中所有item记录的编号与售价字段。程序用gfield函数从当前item记录中读取字段值。用seqacc函数遍历所有item记录。

如果用SQL显示这个清单，数字将以另一种格式输出。以下是SQL显示结果：

```

+-----+
| sql> select serial_number, sales_price |
| sql> from item/                        |
+-----+
| serial_number| sales_price |
+-----+
| 1001| 10.24 |
| 1002| 10.24 |
| 1003| 10.24 |
| 1004| 10.76 |
| 1005| 11.29 |
| 1006| 7.23 |
| 1007| 9.19 |
| 1008| 2.47 |
| 1009| 2.82 |
| 1010| 3.03 |
| 1011| 3.35 |
| 1012| 5.78 |
| 1013| 9.97 |
| 1014| 15.75 |
+-----+

```

从上述两个屏幕显示情况可以看到数据的内部存贮形式与外部显示的区别。内部格式非常接近于物理存贮形式，它决定了数据的实际存贮方式；外部格式则决定了数据的显示形式。

UNIFY指导数据库定义了一个名为isal的字段，它表示售价，数据类型为AMOUNT，LEN=5，这意味着UNIFY显示isal数据时，小数点左边有五位（右边两位）

LEN说明外部格式。UNIFY中isal的内部格式为单精度整数。当用户要从C程序中显示isal数据时，必须先把它由内部格式转化为外部格式。下面的表格列出了数据库中所有六种数据类型的内部和外部格式。

数据库 数据类型	C数据 类型	外部格式
=====	=====	=====
NUMERIC (1-4)	short	9999
(5-9)	long	999999999
FLOAT	double	variable
STRING	char	xxxxxxxx.....
AMOUNT (1-7)	long	999999.99
(8-11)	double	99999999999.99
DATE	short	MM/DD/YY
TIME	short	HH:MM

因此，当你在程序中定义一个变量以存放来自数据库的数据时，必须使变量的类型与上述表格中数据库字段和类型一致。在程序start中，把对应于sno (NUMERIC 9数据库字段) 和对应于isal (AMOUNT 5字段) 的变量都定义为long型。

DATE字段初始化为空，所有其它类型的字段初始化为零（以上均指缺省初始化值）。但缺省初始化值可以被重置（请参阅《UNIFY参考手册》）。UNIFY空值的内部形式为-32768而外部格式为**/**/**。若要置AMOUNT字段为空，只要将其高位字置为-32768，低位字置为零即可。被置为空的AMOUNT字段外部显示为空格串。

UNIFY完全按内部格式存放NUMERIC，FLOAT及STRING数据类型。AMOUNT字段内部存储为美分值，而不是带小数的美元值，输出程序用mod和div计算得到小数点。TIME字段用来存放“时刻”，而不是存放一段时间。它可以表示小时和分，但12:59的内部格式是1259。输出时也要用mod和div来处理。

再看一个有关DATE字段的内部与外部数据格式关系的例子。这个程序显示数据库中所有商品的购入时间。

```
/******  
*  
*   unidate.c  
*  
*   UNIFY Functions Used:  
*       acckey  
*       gfield  
*  
*   Author: Digital Analysis Corporation  
*  
*****/  
$include `../../def/file.h`  
main( )  
{  
    int arr[3];      /* Buffers to hold month, day, year      */  
    long key;        /* Buffer for key field in item record      */  
    short day;       /* Buffer for julian day stored in db      */  
    int status;      /* Return status from db acckey call      */  
    key = 1000;  
    while (key <= 1013)  
    {  
        ++key;  
        if ((status = acckey(item, &key)) != 0)  
        {  
            printf("Status = %d returned from acckey for key = %ld\n",  
                status, key);  
            continue;  
        }  
    }  
}
```