

信息科学与技术丛书

程序设计系列

# Icon与Unicon

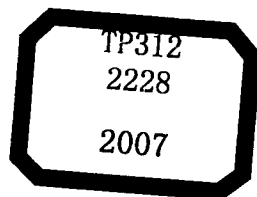
## ——更高级的程序语言

鲁 松 等编著

- ◎ Icon 与 Unicon 的基础应用
- ◎ Unicon 图形用户界面的开发
- ◎ Unicon 面向数据库的编程
- ◎ 网络 CGI 程序的开发
- ◎ 网站实时监视与 E-mail 报警系统
- ◎ 开发游戏软件：手写体、十字绣



机械工业出版社  
CHINA MACHINE PRESS



信息科学与技术丛书  
程序设计系列

# Icon 与 Unicon ——更高级的程序语言

鲁 松 等编著



机械工业出版社

本书深入浅出地介绍了一种更高级的通用型计算机程序语言 Icon/Unicon。全书分为三部分共 25 章,从程序语言的历史开始,详尽讲述了 Icon/Unicon 这一跨平台程序语言中的数据结构、控制结构、字符串处理、文件操作、数值计算、产生式机制、协表达式、计算机绘图,一直到图形界面开发、面向对象编程、数据库操作、Internet 编程和操作系统控制接口等高级技术,并给出了基于 Icon/Unicon 开发的一些有趣而实用的程序实例。本书内容丰富,构思严谨,举例充分,可以使读者在掌握语言的同时,感受到 Icon/Unicon 功能强大且简单易学的独特魅力,体验到其他语言编程中难以得到的乐趣。

本书适合作为计算机编程语言的人门级学习教程,不仅适用于计算机及其相关专业的大学生和研究生,也因 Icon/Unicon 的简单易学,同样适用于初级计算机编程爱好者和计算机程序语言的文科学生学习。

## 图书在版编目(CIP)数据

Icon 与 Unicon——更高级的程序语言/鲁松等编著. —北京:机械工业出版社, 2007.1

(信息科学与技术丛书·程序设计系列)

ISBN 7-111-20348-8

I .I... II .鲁... III .程序语言—程序设计 IV .TP312

中国版本图书馆 CIP 数据核字(2006)第 135329 号

机械工业出版社(北京市百万庄大街 22 号 邮政编码 100037)

责任编辑:丁 诚

责任印制:杨 曦

北京蓝海印刷有限公司印刷

2007 年 1 月第 1 版·第 1 次印刷

184mm×260mm·19.75 印张·485 千字

0001—3000 册

定价:34.00 元

凡购本图书,如有缺页、倒页、脱页,由本社发行部调换

本社购书热线电话:(010) 68326294

编辑热线电话:(010) 88379739

封面无防伪标均为盗版

## 出版说明

随着信息科学技术的迅速发展,人类每时每刻都会面对层出不穷的新技术、新概念。毫无疑问,在节奏越来越快的工作和生活中,人们需要通过阅读和学习大量信息丰富、具备实践指导意义的图书,来获取新知识和新技能,从而不断提高自身素质,紧跟信息化时代发展的步伐。

众所周知,在计算机硬件方面,高性价比的解决方案和新型技术的应用一直备受青睐;在软件技术方面,随着计算机软件的规模和复杂性与日俱增,软件技术受到不断挑战,人们一直在为寻求更先进的软件技术而奋斗不止。目前,计算机在社会生活中日益普及,随着因特网延伸到人类世界的层层面面,掌握计算机网络技术和理论已成为大众的文化需求。由于信息科学技术在电工、电子、通信、工业控制、智能建筑、工业产品设计与制造等专业领域中已经得到充分、广泛的应用,所以这些专业领域中的研究人员和工程技术人员越来越迫切需要汲取自身领域信息化所带来的新理念和新方法。

针对人们对了解和掌握新知识、新技能的热切期待,以及由此促成的人们对语言简洁、内容充实、融合实践经验的图书迫切需要的现状,机械工业出版社适时推出了“信息科学与技术丛书”。这套丛书涉及计算机软件、硬件、网络、工程应用等内容,注重理论与实践相结合,内容实用,层次分明,语言流畅,是信息科学与技术领域专业人员不可或缺的图书。

现今,信息科学技术的发展可谓一日千里,机械工业出版社欢迎从事信息技术方面工作的科研人员、工程技术人员积极参与我们的工作,为推进我国的信息化建设作出贡献。

机械工业出版社

## 前 言

虽然出身于计算机专业，但总是更乐于沉醉在程序算法的设计中，对动手直接编写程序却经常望而却步，不到万不得已，是不愿意出手的。原因很简单，C/C++ 等高级语言的繁琐，Perl 等脚本语言的晦涩，总让人感觉编程是件很麻烦的事情。

这本是不应该的。

“工于善其事，必先利其器。”这把利器就是 Icon 和 Unicon。作为一种更高级的程序语言，用 Icon/Unicon 编写程序确实可以给人带来一种快乐。这种快乐来自于我们找到了一种以最简单的方式可以完成最复杂任务的工具，并由此享受着编写程序本来应有的快乐。越是老练的程序员就会越有这样的感觉。

所以说，Icon/Unicon 可以用更短的时间、更少的代码行和有限的精力，完成更多的工作。甚至有人感叹，两人写一年的 C++ 程序，改用 Icon/Unicon，一个人两个月就可以完成。

正是由于 Icon 和 Unicon，编程变得简单和快乐了。易用的东西通常难学，而在易用和好学上，Icon 和 Unicon 恰恰达到了统一。

本书试图以最简单和最直接的方式让人们尽快知道其中的原因。而更重要的是，要与大家分享 Icon 和 Unicon 可能带来的快乐。这也是作者三年来没有因为各种阻碍放弃完成这本书的真正初衷。

本书主要由鲁松编写，参与编写的人员还有邹昕、陈忠、肖颖和朱嵬。

最后，必须感谢我的家人默默地为做出的牺牲，使我节省了太多的业余时间来完成此书，必须感谢 Unicon 的缔造者 Jeffery Clinton 博士为我提供所需的技术文档，必须感谢国家计算机网络与信息安全实验室和先锋未来软件开发有限公司为我提供的不可或缺的环境支持，还必须感谢机械工业出版社计算机分社等所有为出版此书和发展 Icon/Unicon 做出贡献、提供帮助的人。

鲁 松

# 目 录

出版说明

前言

## 第 一 部 分

|                                |    |
|--------------------------------|----|
| 第 1 章 Icon 与 Unicon .....      | 1  |
| 1.1 程序语言的体系 .....              | 1  |
| 1.2 Icon 和 Unicon 的渊源 .....    | 3  |
| 1.3 Icon 和 Unicon 的用途 .....    | 3  |
| 1.4 Icon 和 Unicon 的特色 .....    | 3  |
| 1.5 Icon/Unicon 的限制 .....      | 6  |
| 第 2 章 Icon 及 Unicon 的启动 .....  | 7  |
| 2.1 Icon 与 Unicon 的获取 .....    | 7  |
| 2.2 Icon 及 Unicon 的安装与设置 ..... | 7  |
| 第 3 章 Icon 及 Unicon 程序 .....   | 11 |
| 3.1 Icon/Unicon 的程序结构 .....    | 11 |
| 3.1.1 过程的价值 .....              | 12 |
| 3.1.2 过程间数据交换 .....            | 13 |
| 3.2 运行 Icon/Unicon 程序 .....    | 14 |
| 3.2.1 Icon 程序的创建 .....         | 14 |
| 3.2.2 Icon 程序的编译链接 .....       | 14 |
| 3.3 运算符 .....                  | 16 |
| 3.4 数据类型 .....                 | 18 |
| 3.5 保留字 .....                  | 19 |
| 3.6 关键字 .....                  | 19 |
| 3.7 变量 .....                   | 20 |
| 3.7.1 变量的命名 .....              | 20 |
| 3.7.2 变量的定义 .....              | 21 |
| 3.7.3 变量的范围 .....              | 21 |
| 3.8 语句 .....                   | 22 |
| 3.8.1 注释语句 .....               | 22 |
| 3.8.2 语句分割 .....               | 22 |
| 第 4 章 控制结构 .....               | 24 |
| 4.1 选择结构 .....                 | 24 |

|                                      |           |
|--------------------------------------|-----------|
| 4.2 循环结构 .....                       | 25        |
| 4.3 退出与中止结构 .....                    | 26        |
| 4.3.1 自然退出 .....                     | 26        |
| 4.3.2 循环结构的退出和从头再来——break、next ..... | 26        |
| 4.3.3 程序的强制退出——stop、exit .....       | 28        |
| 4.3.4 过程的返回——return .....            | 28        |
| 4.3.5 调用过程的挂起——suspend .....         | 29        |
| <b>第5章 数据结构 .....</b>                | <b>31</b> |
| 5.1 表——list .....                    | 31        |
| 5.1.1 表结构的定义 .....                   | 31        |
| 5.1.2 表结构变量的创建 .....                 | 31        |
| 5.1.3 表结构变量的使用 .....                 | 32        |
| 5.1.4 表结构变量的运算 .....                 | 32        |
| 5.1.5 表结构变量的操作函数 .....               | 33        |
| 5.1.6 与表结构变量有关的函数 .....              | 33        |
| 5.1.7 表结构与数组 .....                   | 34        |
| 5.1.8 表结构中间元素的添加和删除 .....            | 35        |
| 5.2 索引表——table .....                 | 35        |
| 5.2.1 索引表结构的定义 .....                 | 35        |
| 5.2.2 索引表结构变量的创建和使用 .....            | 36        |
| 5.2.3 索引表结构变量的运算 .....               | 36        |
| 5.2.4 索引表结构变量的操作函数 .....             | 37        |
| 5.2.5 索引表结构变量的排序 .....               | 38        |
| 5.2.6 索引表结构变量的特殊情况 .....             | 38        |
| 5.2.7 索引表结构的嵌套 .....                 | 39        |
| 5.3 集合——set .....                    | 39        |
| 5.3.1 集合结构的定义 .....                  | 39        |
| 5.3.2 集合结构变量的建立 .....                | 40        |
| 5.3.3 集合间的运算 .....                   | 40        |
| 5.3.4 集合结构的函数 .....                  | 41        |
| 5.3.5 集合间关系的判定 .....                 | 42        |
| 5.4 记录——record .....                 | 42        |
| 5.4.1 记录结构的定义 .....                  | 42        |
| 5.4.2 记录变量的使用 .....                  | 43        |
| 5.4.3 记录数据结构的操作 .....                | 43        |
| 5.4.4 记录数据结构的函数 .....                | 44        |
| 5.5 数据结构的比较和总结 .....                 | 45        |
| <b>第6章 输入输出与文件操作 .....</b>           | <b>48</b> |
| 6.1 数据的输入输出体系 .....                  | 48        |

|                             |           |
|-----------------------------|-----------|
| 6.2 文件的读写 .....             | 48        |
| 6.2.1 文件设备的打开和关闭 .....      | 48        |
| 6.2.2 写入文件 .....            | 50        |
| 6.2.3 读取文件 .....            | 52        |
| 6.3 数据的键盘输入 .....           | 53        |
| 6.4 显示器和打印机输出 .....         | 54        |
| <b>第7章 字符、字符集与字符串 .....</b> | <b>57</b> |
| 7.1 字符 .....                | 57        |
| 7.1.1 定义 .....              | 57        |
| 7.1.2 转义字符 .....            | 57        |
| 7.1.3 字符函数 .....            | 59        |
| 7.2 字符集 .....               | 59        |
| 7.2.1 定义 .....              | 59        |
| 7.2.2 运算 .....              | 59        |
| 7.3 字符串 .....               | 60        |
| 7.3.1 定义 .....              | 60        |
| 7.3.2 字符串中大写字母的判断 .....     | 60        |
| 7.3.3 引用、运算和比较 .....        | 61        |
| 7.3.4 字符串函数 .....           | 63        |
| 7.3.5 字符串的分析 .....          | 65        |
| 7.3.6 汉字的处理 .....           | 71        |
| <b>第8章 数值计算 .....</b>       | <b>75</b> |
| 8.1 数学关键词 .....             | 75        |
| 8.2 算术运算符 .....             | 75        |
| 8.3 数值比较运算符 .....           | 76        |
| 8.4 数学计算函数 .....            | 76        |
| 8.4.1 转换函数 .....            | 76        |
| 8.4.2 常用计算函数 .....          | 77        |
| 8.4.3 三角函数和角度转换函数 .....     | 77        |
| 8.5 复数的计算 .....             | 77        |
| 8.6 整数按二进制位的计算 .....        | 78        |
| 8.7 伪随机数 .....              | 79        |
| 8.8 大数的问题及其他数学库函数 .....     | 80        |
| 8.8.1 大数的问题 .....           | 80        |
| 8.8.2 与数学有关的库函数 .....       | 80        |
| 8.9 均方差：成绩稳定性测试 .....       | 81        |
| <b>第9章 产生式与多值产生式 .....</b>  | <b>84</b> |
| 9.1 产生式 .....               | 84        |
| 9.2 多值产生式 .....             | 84        |



|                                   |     |
|-----------------------------------|-----|
| 9.2.1 多值产生式详细 .....               | 84  |
| 9.2.2 产生条件 .....                  | 86  |
| 9.2.3 多值表达式的执行顺序 .....            | 87  |
| 9.2.4 多值的强制数量限制 .....             | 89  |
| 9.3 HTML 页面中内容提取 .....            | 89  |
| <b>第 10 章 协表达式</b> .....          | 93  |
| 10.1 协表达式变量的创建 .....              | 93  |
| 10.2 协表达式的触发 .....                | 93  |
| 10.3 协表达式的触发计数 .....              | 93  |
| 10.4 协表达式的更新 .....                | 94  |
| 10.5 协表达式操作汇总 .....               | 94  |
| 10.6 协表达式变量作为参数 .....             | 95  |
| 10.7 利用协表达式统计与分析文本 .....          | 95  |
| <b>第 11 章 高级技术</b> .....          | 97  |
| 11.1 Icon 信息读取 .....              | 97  |
| 11.2 赋值和复制 .....                  | 97  |
| 11.3 关联与非关联 .....                 | 98  |
| 11.4 递归: 阶乘与汉塔 .....              | 99  |
| 11.5 正则表达式 .....                  | 101 |
| <b>第 12 章 预处理</b> .....           | 105 |
| 12.1 \$define/\$undef 预处理指令 ..... | 105 |
| 12.2 \$include 预处理指令 .....        | 107 |
| 12.3 条件编译预处理指令 .....              | 108 |
| 12.4 内置的预定义标示符 .....              | 109 |
| 12.5 \$error 预处理指令 .....          | 110 |
| 12.6 \$line 预处理指令 .....           | 111 |
| 12.7 预处理指令汇总 .....                | 111 |
| <b>第 13 章 程序的调试和排错</b> .....      | 112 |
| 13.1 错误排查关键字 .....                | 114 |
| 13.2 错误排查函数 .....                 | 116 |
| <b>第 14 章 绘图</b> .....            | 118 |
| 14.1 窗口 .....                     | 118 |
| 14.2 绘点线的操作函数 .....               | 122 |
| 14.3 填充图操作函数——填充图与太极图 .....       | 125 |
| 14.4 窗口中的字符操作 .....               | 127 |
| 14.5 区域处理 .....                   | 130 |
| 14.6 事件处理 .....                   | 130 |
| 14.6.1 事件处理函数 .....               | 130 |
| 14.6.2 事件处理关键字 .....              | 131 |

|                             |     |
|-----------------------------|-----|
| 14.6.3 事件有关的属性 .....        | 132 |
| 14.6.4 事件处理举例：鼠标事件的记录 ..... | 132 |
| 14.7 对话框 .....              | 135 |
| 14.8 任意图形的绘制 .....          | 141 |
| 14.8.1 图素绘制函数 .....         | 142 |
| 14.8.2 调色板 .....            | 143 |
| 14.8.3 调色板函数 .....          | 145 |
| 14.8.4 八皇后问题的绘图显示 .....     | 145 |

## 第二部分 Unicon

|                                    |     |
|------------------------------------|-----|
| 第 15 章 函数库和应用程序库 .....             | 146 |
| 15.1 Unicon 的内置函数 .....            | 146 |
| 15.2 Unicon 的新增能力 .....            | 147 |
| 15.3 Icon 程序库 .....                | 147 |
| 第 16 章 面向对象、命名冲突及其 Unicon 实现 ..... | 153 |
| 16.1 面向对象 .....                    | 153 |
| 16.1.1 思想 .....                    | 153 |
| 16.1.2 价值 .....                    | 153 |
| 16.1.3 思想与语言 .....                 | 154 |
| 16.1.4 基本概念 .....                  | 154 |
| 16.2 面向对象与 Unicon .....            | 156 |
| 16.2.1 类的定义 .....                  | 156 |
| 16.2.2 对象的定义和使用 .....              | 157 |
| 16.2.3 类的初始化定义 .....               | 158 |
| 16.2.4 类的属性与成员函数的参数 .....          | 160 |
| 16.2.5 类的继承 .....                  | 160 |
| 16.2.6 类的关联 .....                  | 164 |
| 16.3 基于类的绘图与堆栈数据结构 .....           | 165 |
| 16.4 命名空间与打包 .....                 | 168 |
| 第 17 章 图形用户界面的开发 .....             | 172 |
| 17.1 概念 .....                      | 172 |
| 17.2 图形界面的开发工具——Ivib .....         | 173 |
| 17.2.1 启动 Ivib .....               | 173 |
| 17.2.2 常用组件详细 .....                | 174 |
| 17.2.3 Ivib 的应用详细 .....            | 176 |
| 17.3 Ivib 与 Unicon .....           | 206 |
| 17.4 组件、容器和窗口类中成员函数的调用 .....       | 206 |
| 17.5 图形开发界面库 gui.icn 详细 .....      | 218 |

|                                               |            |
|-----------------------------------------------|------------|
| 17.6 注意事项 .....                               | 218        |
| <b>第 18 章 面向数据库的编程 .....</b>                  | <b>220</b> |
| 18.1 DBM 数据库及其 Unicon 操作 .....                | 220        |
| 18.2 SQL 关系数据库及其 Unicon 操作 .....              | 221        |
| 18.2.1 SQL 关系数据库及 SQL 语言 .....                | 221        |
| 18.2.2 ODBC 接口的逻辑结构 .....                     | 222        |
| 18.2.3 Unicon 的 ODBC 接口 .....                 | 223        |
| 18.3 基于 mySQL 数据库的编程 .....                    | 228        |
| <b>第 19 章 Unicon 与 Internet 协议的控制接口 .....</b> | <b>233</b> |
| 19.1 Unicon 控制 Internet 协议的基本框架 .....         | 233        |
| 19.2 TCP 协议的控制 .....                          | 234        |
| 19.2.1 从服务器读取 TCP 数据 .....                    | 234        |
| 19.2.2 监听本地流入的 TCP 数据 .....                   | 235        |
| 19.2.3 互联网数据的读取 .....                         | 236        |
| 19.3 UDP 协议的控制 .....                          | 237        |
| 19.4 HTTP 协议的控制 .....                         | 238        |
| 19.5 POP 和 SMTP——电子邮件的收发 .....                | 239        |
| 19.5.1 POP .....                              | 239        |
| 19.5.2 SMTP .....                             | 242        |
| <b>第 20 章 Unicon 中操作系统的控制接口 .....</b>         | <b>245</b> |
| 20.1 文件与目录的控制 .....                           | 245        |
| 20.1.1 文件 .....                               | 245        |
| 20.1.2 目录 .....                               | 246        |
| 20.1.3 文件/目录的信息获取 .....                       | 247        |
| 20.1.4 目录的删除、大小统计和文件查找 .....                  | 248        |
| 20.2 程序、进程和信号的控制 .....                        | 251        |
| 20.2.1 程序的控制 .....                            | 251        |
| 20.2.2 信号的控制 .....                            | 252        |
| 20.2.3 进程的控制 .....                            | 254        |
| 20.2.4 等待函数 select () .....                   | 254        |

### 第三部分 应用实例

|                                  |            |
|----------------------------------|------------|
| <b>第 21 章 网络 CGI 程序的开发 .....</b> | <b>256</b> |
| 21.1 CGI 技术 .....                | 256        |
| 21.2 CGI 程序——网上报刊订阅系统 .....      | 257        |
| 21.2.1 CGI 程序的接口 .....           | 257        |
| 21.2.2 数据传递 .....                | 263        |
| 21.2.3 环境变量 .....                | 264        |

|                                       |     |
|---------------------------------------|-----|
| 21.2.4 CGI 程序·····                    | 265 |
| 21.3 Icon/Unicon 的 CGI 程序·····        | 265 |
| 21.3.1 Icon/Unicon 的 CGI 程序框架 ·····   | 266 |
| 21.3.2 输入数据的读取 ·····                  | 266 |
| 21.3.3 处理结果的输出 ·····                  | 267 |
| 21.3.4 读取环境变量 ·····                   | 270 |
| 21.3.5 CGI 程序中的写文件 ·····              | 270 |
| 21.3.6 Icon/Unicon 编写 CGI 程序小结 ·····  | 271 |
| 21.4 CGI 程序的运行平台 ·····                | 271 |
| 第 22 章 Google 搜索及 Yahoo! 财经信息读取 ····· | 275 |
| 22.1 定制 Google 搜索 ·····               | 275 |
| 22.2 查询 Yahoo 财经消息 ·····              | 277 |
| 第 23 章 网站实时监视与 E-mail 报警系统·····       | 279 |
| 第 24 章 游戏软件：手写体、十字绣 ·····             | 283 |
| 24.1 手写体 ·····                        | 283 |
| 24.2 十字绣 ·····                        | 284 |
| 第 25 章 Icon/Unicon 的精进·····           | 290 |
| 参考文献 ·····                            | 292 |
| 索引表 ·····                             | 293 |

# 第一部分

## 第1章 Icon 与 Unicon

在这里 Icon 不是图标的意思。作为一种程序语言, Icon 和 Unicon 是当今世界上,语法形式最优美、功能最强大,也是最为好学易用的计算机程序语言。而我们喜爱 Icon 和 Unicon 的真正原因,是他们可以给我们带来意想不到的编程快乐。所以有人说, Icon 和 Unicon 是更高级的程序语言,而“更高级”则意味着 Icon/Unicon 可以使程序员以更快的速度和更简单的方式完成更多的事情。

### 1.1 程序语言的体系

人类的语言是人与人之间交流用的,我们称之为自然语言(natural language);计算机程序语言是人用来操纵计算机用的,通常被简称为程序语言(programming language)。两者之间有着有趣的联系和惊人的相似之处。

作为硬件的计算机只能处理二进制机器码。而早期对计算机的控制是用穿孔纸带以二进制方式实现的。要把一条计算机操作命令编写为一串由 01 构成的二进制代码,难度可想而知。于是计算机科学工作者设计了更易于人理解和编写的各种类型的低级或高级的程序语言,通过编译器(compiler)/解码器(interpreter)和链接器(linker)对源代码(source code)进行编译/解码、链接(link),直至转换成二进制代码,使计算机得以识别和执行。

程序语言的所处地位如图 1-1 所示。

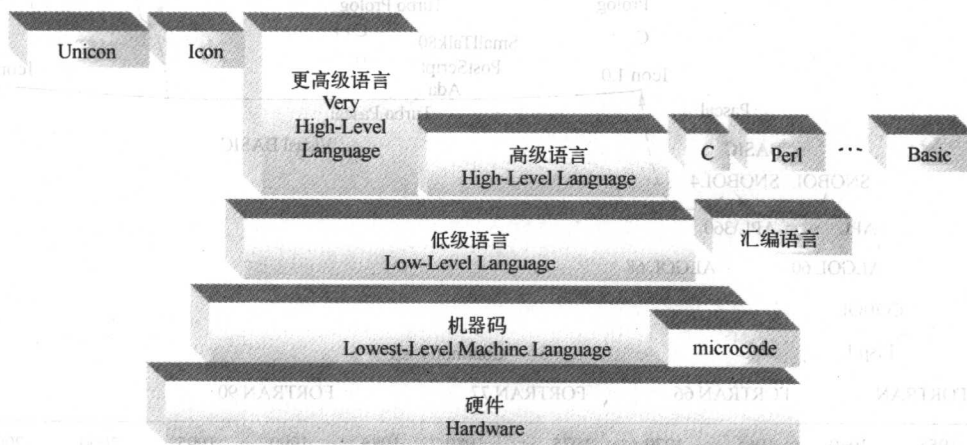


图 1-1 程序语言的层次

对于人来讲,易写、好读而且维护简便的高级语言源代码(source code)需要经过编译器转换为与机器码类似的目标码(object code),再用连接器或汇编器(assembler)对目标码处理生成机器码(machine language)。编译链接流程如图 1-2 所示。

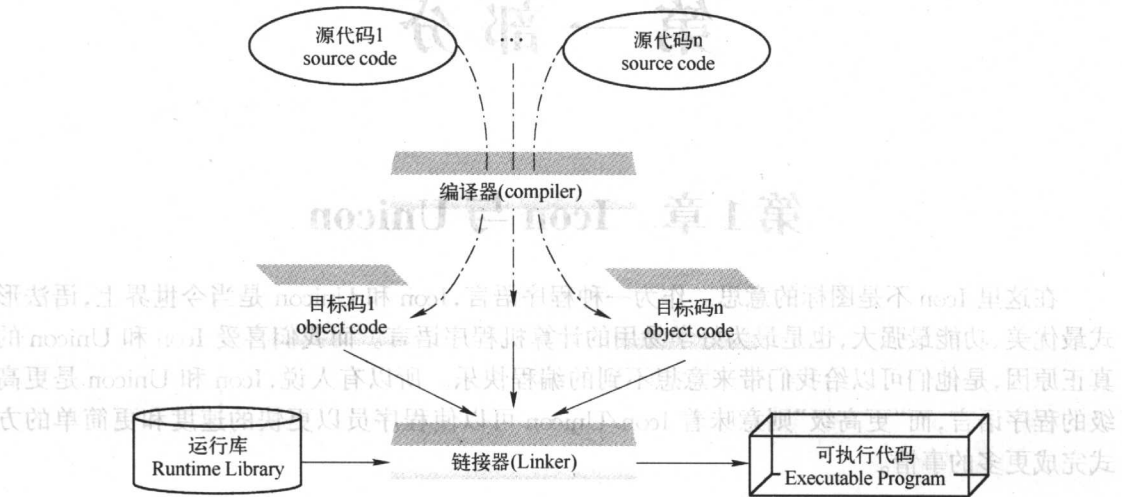


图 1-2 编译链接流程

从第一个高级程序语言 FORTRAN 诞生以来,多达 2500 种以上的各种专用或通用的程序语言被设计出来,其中,能流传至今,并被继续发扬光大的则是凤毛麟角。一种编程语言是发展还是灭亡,就像人类自然语言的发展史一样,遵循着自然淘汰的法则。

简单地讲,这个法则就是三条:强大,易学,好用。

主流程序语言发展的年鉴略图如图 1-3 所示。

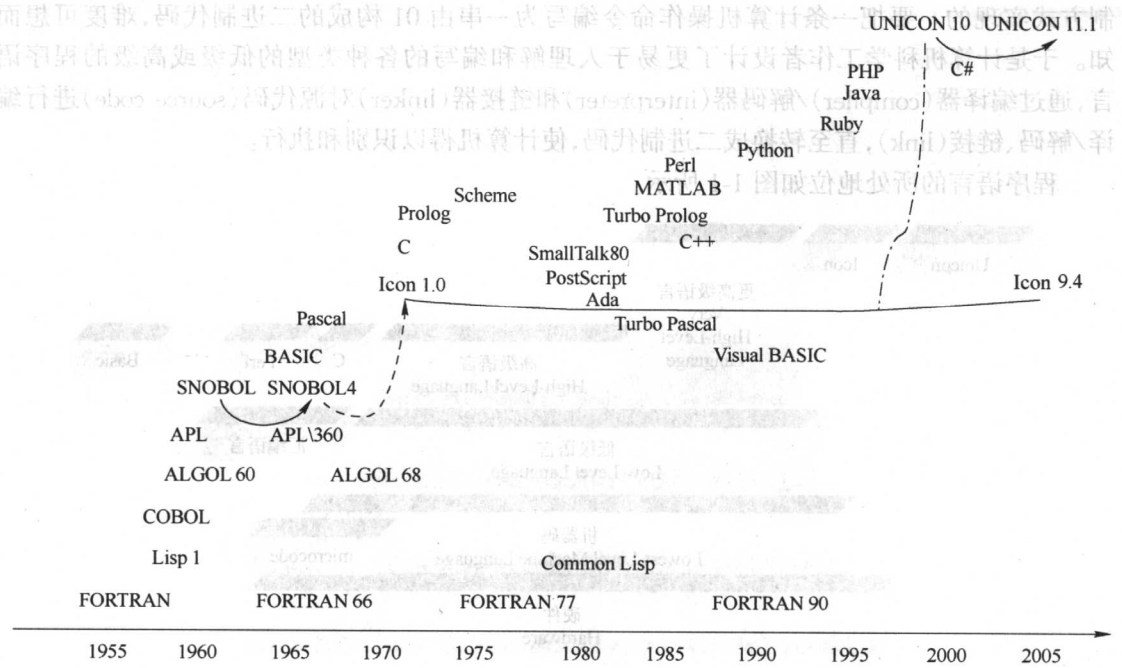


图 1-3 计算机程序语言的发展

## 1.2 Icon 和 Unicon 的渊源

说到 Icon 和 Unicon,不能不提及其前身 SNOBOL(StriNg-Oriented symBolic Language)。1962 年 Bell Lab(贝尔实验室)的三名计算机科学家(D.J. Farber, R.E. Griswold, F.P. Polensky)创造了长于字符/字符串处理的专用程序语言 SNOBOL。尽管极具特色,但发展到 SNOBOL4 后,终因它的动态特性(dynamic)和解释执行效率低下的原因终止了进一步的发展。SNOBOL 创始人之一的 R.E. Griswold 博士到 The University of Arizona(亚利桑那大学)后,在保持了 SNOBOL 的突出特点的基础上,于 1970 年改造 SNOBOL 而创造出了 1.0 版的 Icon 程序语言,直到今天的 9.4 版。Icon 不仅保持了字符(串)高效处理的传统特色,而且发展成为功能强大、易学易用的通用型程序语言。

在 Icon 继续发展的同时,为了满足当前技术发展的需求,Icon 的开发之一 Jeffery Clinton 博士组织了在 Icon 中融入面向对象、数据库操作、Internet 编程和系统控制等新技术的开放源代码项目,将其命名为 Unicon(UNified extended Icon)。Unicon 保持了与 Icon 的高度兼容性,在继承 Icon 所有优点的同时发展成为一种功能更为强大的程序语言。

## 1.3 Icon 和 Unicon 的用途

作为通用型程序语言,Icon 和 Unicon 可以胜任任何任务,但最擅长的任务是:

- 程序的快速开发;
- 原型系统的快速建模和实现;
- 网络管理员的日志分析;
- 人工智能、专家系统、计算语言学、符号计算等科学研究领域的计算机实现;
- 字符/字符串复杂高效的计算机处理;
- 文本的编辑和分析;
- 系统管理员的系统编程和操作;
- 图形程序的快速开发;
- 计算机程序语言的教学与研究;
- 跨平台的程序开发;
- 解决数据间存在复杂关系的问题。

## 1.4 Icon 和 Unicon 的特色

Icon/Unicon 令实现了 Icon 千方百计地降低编程复杂性的设计初衷,也为其强大功效提供了根本保障。

### (1) 易学

尽管理解和体会 Icon/Unicon 的精髓需要些时日,但如果有一点点程序语言学习基础,几天时间加上函数参考手册,就足以对 Icon/Unicon 得心应手。平滑的学习曲线使 Icon/Unicon 成为最易学的一种编程语言。

## (2) 强大的字符串处理能力

高效复杂的字符串处理能力是 Icon/Unicon 最具特色的闪光点。大家都知道,在 C 语言中定义和操作字符串不是件容易的事。但在 Icon/Unicon 中实现字符串的定义和字符的添、删、改以及灵活多样地索引都相当轻松。

## (3) 特有的数据类型

Icon 内置了三种高级的数据类型:集合(set)、列表(list)和哈希表(table),并针对他们定义了相应和全面的操作函数。

set 数据类型是离散数学中的基本概念,它的不可重复性以及针对集合的交、并、补和差运算为系统管理员统计和分析系统信息提供了其他编程语言难得一见的工具。

Icon 中没有数组(array)的概念,但却提供了一个比数组功能更为强大和方便的数据结构 list。list 可以表示为 数组(array)、向量(vector),以及后进先出的栈(stack)和先进先出的队(queue),并为此提供了相应的多种访问和存取方式。list 是 Lisp 语言最基本的数据类型,在此基础上形成各种针对 list 操作的方言系统。Icon 语言中的 list 数据类型及其相应的操作能力与 Lisp 语言中的 list 相比是有过之而无不及的。

table 是一种极有用的数据类型。虽然在 C 和 Pascal 等高级语言中没有,但 Perl, Python 和 Java 中可以看到它。虽然 table 不是 Icon 独有的,但极易实现的无限制多层 table 嵌套,以及构成 2 维、3 维等高维 table 却是 Icon 的过人之处。

## (4) 数据类型的动态定义

写过 C 语言程序的人都知道,其中的数据类型定义尽管严格,但编译后满篇的 error 和 warning,总给人一种倍受束缚的感觉。在 C 语言中人为了适应机器,必须事无巨细地遵守规则。这种束缚在 Icon 中得到了极大的改善。除了上述三种内置的高级数据类型由于其特殊性需要声明数据类型外,其他就再也没有任何限制了。更方便的是,各种数据类型可以无需声明的转换。

## (5) 多值表达式

Icon/Unicon 中的表达式(expression)求值功能强大而有趣。他可以产生一个值,也可以不产生任何值(failure),还可以产生多个值。这一设计与回溯机制结合,可以极大简化代码量,为求值提供一种更灵活的操控方式。

## (6) 优美的语言形式

Perl 语言的流行不是偶然,正像它的代表图标“骆驼”一样,它丑陋但耐劳和稳健。它的强大体现在它出色的字符串处理和 UNIX 下丰富的系统函数的调用上,这也是 Perl 能够取代 UNIX shell 在系统管理员甚至黑客中广为流传的关键。但作为一种高级的编程语言,它确实是丑陋的。Perl 代码给人的感觉更像是机器码,有时简洁到了怪异的程度,可读性极差。对于 Perl 的新手,稍不小心就会陷入莫名其妙的错误中。

Icon/Unicon 不一样,用 Icon/Unicon 写出的程序会给人一种清新优美的感觉。用过的程序语言越多,就越能体会这一点;编程经验越丰富,才越能感到良好可读性的价值。

举个例子,在 Perl 中,若有数组变量:

```
@b=([(7,8,9)],2,3,4)
```

如果想得到(7,8,9),用 \$b[0]不行,用@b[0]也不行,Perl 将会输出:



ARRAY(0xb75eb0)

必须要用 `@{ $b[0]}`。而在 Icon/Unicon 语言中,

```
b := [[7,8,9],2,3,4]
```

要想得到 `[7,8,9]`,则用 `b[1]` 即可。可见 Icon 的统一和简洁。

#### (7) 内存的自动管理

人们常问 `set`、`table`、`list` 不需要像 C 语言中的数组那样定义大小么? 这是个好问题,由此也引出了 Icon/Unicon 的另一优点:变量的大小是由 Icon/Unicon 动态管理的,硬件内存的大小是其惟一的限制。尽管许多人认为,这种不加预计的变量大小管理方式会造成程序异常错误,但绝大多数人会喜欢的,因为它带来的方便要远远大于由它引来的麻烦。

#### (8) GUI

图形用户界面(GUI)的设计在 IDE 中不是新鲜玩意儿,但在 Perl 和 Python 这样的新兴且时尚的编程语言中却是罕见的。但 Icon/Unicon 具备这一功能,其中的 Icon VIB(Visual Interfaces for Icon)和 Unicon Ivis 使得用 Icon 编写的程序可以摆脱命令行的枯燥和繁琐,实现像 Windows 程序那样简便轻快的鼠标操作。

#### (9) CGI

系统管理员(system administrator)毕竟是少数,更多的人了解 Perl 是从 CGI 的 Perl 实现开始的。Icon 也能做到,而且做得更好。CGI 的一个重要作用就是实现 client 和 server 之间的交互,且这种交互多以字符串为主,而这恰恰是 Icon/Unicon 的强项。Icon/Unicon 提供了一个 CGI 接口函数,可以用一种最为简便的方式完成 CGI 实现。由于 Icon/Unicon 的 CGI 程序是可执行的二进制代码,所以代码的安全性和版权问题无需担心。

#### (10) 绘图

Icon/Unicon 提供了一个丰富的绘图函数库,这为它实现数据可视化和制作有趣的电脑游戏提供了可能和一个十分简单易用的工具。这是 Icon/Unicon 最为与众不同的地方。

#### (11) 跨平台

无论是程序员还是使用者,谁不希望程序可以“一次书写,到处运行”。Icon 在不修改任何代码的情况下可以运行于加载 Icon 解释器的 MS-DOS、Macintosh/MPW、Linux、MS Windows 3.1/95/NT/98/Me/2000/XP/2003、VAX/VMS 和各种各样的 UNIX 平台上。

除了上述从 Icon 中继承的特色,Unicon 进一步丰富了如下功能:

##### (1) 面向对象(Object-Oriented)

Icon 中提供了一个附件 `idol` (Icon-Derived Object Language)可以方便地支持面向对象的思想。

Unicon 作为 Icon 99.9%兼容的继承者,放弃了附件的思想,干脆将面向对象技术集成到语言中。如此一来,Unicon 成为了地地道道的面向对象的编程语言,为 Unicon 适用于大型软件系统的开发奠定了基础。

面向对象并不代表一切,要具体问题具体分析。所以,面向对象只是 Unicon 编程的一种选择,但不是惟一的选择。

##### (2) 数据库连接

Icon 不支持数据库连接,但 Unicon 支持。通过标准的 ODBC 接口,Unicon 实现了对本地