

适合开发者 项目经理 CTO&CIO 编程爱好者阅读收藏

**Broadview**  
WWW.BROADVIEW.COM.CN

**Csdn**.net

**增值**

# 程序员

## 2006精华本

### Programmer 上

《程序员》杂志社 编

**合订内容:**

《程序员》全年12期精华

**超大增值内容:**

开发者读库  
新工具推荐  
2006新技术辞典  
2006年开发图书TOP 100  
2006技术月历  
2006年软件业大事记  
2006年中国开发者大调查

**光盘(1 DVD):**

《程序员》全年电子刊  
CSDN《软件研发》全年电子刊  
2006年度优秀图书精选连载

**CSDN & 《程序员》名家访谈精彩视频**



电子工业出版社

Publishing House of Electronics Industry  
<http://www.phei.com.cn>



杂志频道: <http://www.programmer.com.cn>

# 程序员

## 2006 精华本(上)

《程序员》杂志社 编

总策划: 蒋 涛  
编委会: 唐 琦 韩 磊 孟 岩 闫 辉  
刘龙静 张浩祥 欧阳璟 霍泰稳  
常 政 刘洪洁 朱海燕



电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

# 程序员 2006 精华本 (上)

未经许可, 不得以任何方式复制或抄袭本书之部分或全部内容  
版权所有, 侵权必究

## 图书在版编目(CIP)数据

程序员 2006 精华本 / 《程序员》杂志社编. —北京: 电子工业出版社, 2006.12  
ISBN 7-121-03640-1

I. 程... II. 程... III. 程序设计—文集 IV. TP311.1-53

中国版本图书馆 CIP 数据核字 (2006) 第 152617 号

责任编辑: 孟迎霞

印 刷: 北京牛山世兴印刷厂  
装 订:

出版发行: 电子工业出版社

北京市海淀区万寿路 173 信箱 邮编: 100036

开 本: 850 × 1168 大 1/16 印张: 25.5 字数: 800 千字 彩插: 8

印 次: 2007 年 1 月第 1 次印刷

定 价: 45.00 元 (赠光盘 1 张)

读者订购或发现装订错误、缺损、破损, 请与《程序员》读者服务部联系。

联系地址: 北京市朝阳区酒仙桥路 14 号兆维工业园 B3 区 2 门 1 层 (100016)

电 话: 010-51661202-202 010-64351425

传 真: 010-64348545





**慧都科技**  
WWW.EVGET.COM

**选择慧都 · 选择专业**

采用**慧都控件**—为您的软件开发铸入源源不绝的动力

产品投放时间缩至原来的 **50%**

产品缺陷密度降至原来的 **35%**

维护成本降至原来的 **70%**

开发成本降至原来的 **30%**



**www.evget.com**

**全面**

超过1000个优秀控件

**专业**

专业的控件信息化平台

**时效**

与超过150个商家同步更新

**尽在慧都控件网**

**中国专业的控件产品代理服务商 助您打造成功软件**

- 慧都公司：专业的控件代理和技术支持的服务商，微软VS企业合作伙伴（VSIP），众多知名企业的首选。
- 慧都伙伴：遍及全球几十个国家和地区，是许多控件厂商在大中国地区的代理和指定的技术支持服务商。
- 慧都技术：专业的技术支持队伍，提供一整套完善的技术分析方案和解决方案，解决问题高效快捷周到。
- 慧都目标：全球优秀的控件代理和技术服务商！

慧都科技有限公司

重庆高新区帝豪名都7-8

电话：023-66090381

上海东昌路东安大楼1101

电话：021-58825726

Email: sales@evget.com

MSN: evgetsupport@hotmail.com

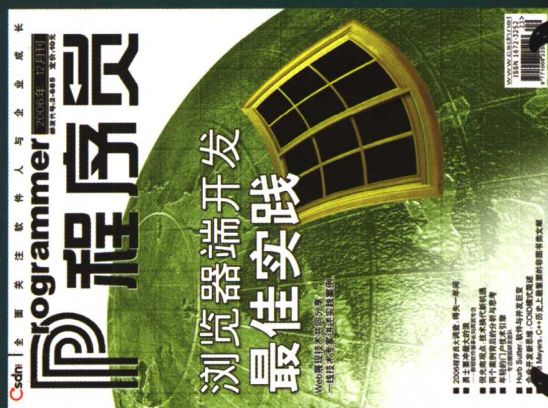
网址: www.evget.com

**让成功从控件起步**





# 极具 珍藏价值 的技术刊物



2007年《程序员》杂志征订现已开始

## 全面关注软件人与企业成长

网上订阅: [www.dearbook.com.cn](http://www.dearbook.com.cn) 电话: 010-64351425 010-51661202-2621176

E-mail: [reader@csdn.net](mailto:reader@csdn.net) 传真: 010-64348545

优惠订阅, 好礼多多! 详情敬请关注[www.csdn.net](http://www.csdn.net)

地址: 北京市朝阳区酒仙桥路14号兆维工业园B3区2门1层《程序员》读者服务部 邮编: 100016  
开户行: 中国银行 支行: 北京创新乐知广告有限公司 帐号: 802119627508091001



# 《程序员 2006 精华本》内容简介

《程序员》创刊6年来，每年末编辑部精心打造的“合订本”已经形成一个品牌，得到广大读者的认可和喜爱。2006年，《程序员》杂志对内容进行了优化整合，每期推出一个大型专题策划，各版块也纷纷推出新的主打栏目，精彩随处可见，另外杂志的版式、色彩方面也有了很大飞跃，给读者带来耳目一新的阅读体验。历时2个多月精心整理编制的《程序员2006精华本》，除荟集了12期杂志的精华，更新增开发者读库、程序员手册、2006年鉴等增值内容，全书共832页，分为上下2册，并附赠1张DVD光盘。相关具体内容介绍如下：

## ► 合订增值版

**人物 & 报道：**包括《程序员》的经典品牌栏目名人堂、软件产业、人物专访、软件创业及程序百味等，您既可以宏观地了解到软件产业的动态；又能体会到成功者的酸甜苦辣；聆听到专家的观点。

**管理 & 实践：**分为项目实践、团队故事、需求分析、管理圈、新视点几个部分，简单却清晰地说明了管理的几个重要方面。包含2006全年热点管理方法、最新的开发模式等。

**程序员手册：**这个栏目既是对原有栏目内容的整合，又包含了编辑策划的增值内容，如百本技术书评、各种实用技术文档、常用工具点评等。并调整了内容结构，更加突出“手册”的作用，方便读者查找。

**年鉴：**《2006年软件业大事记》里程碑式的介绍了2006年的业界事件，各种重大资讯尽在掌握；《技术月历》、《2006书市观察》中的专家分析，让行业发展趋势一目了然；《2006开发者大调查》更是不可多得的第一手数据分析。

## ► 技术专题版

**《程序员》特别策划：**集合全年12期特别策划，12个热点专题尽收其中，为开发者献上一份饕餮盛宴。

**技术专区：**包含了所有热点领域的优秀技术文章，Java、.NET、数据库、网络与安全、软件调试等，为《程序员》全年的技术精华所在。

**开发者读库：**为本书增值专题，也是本年度亮点。邀请技术高手、架构师、领域专家亲自为本书打造的精彩文章，就其所专长技术的实际应用、某项技术关键点的深入理解进行论述，深入浅出，值得期待！

## ► 光盘介绍

本DVD主要包括了2006全年杂志PDF文档、CSDN网站推出的《软件研发》、全年电子刊、CSDN&《程序员》名家访谈精彩视频、2006年度优秀图书精选试读、优秀开源代码库、热门软件集锦等，另外《程序员》编辑部还特别奉上了《编辑部故事（贺年版）》，特别适合开发者学习、查阅与收藏。



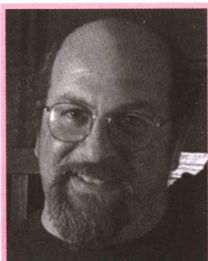
特别感谢

(排名不分先后, 以姓氏首字母为序,)

# Contributors



**Brian Behlendorf**  
Apache之父, CollabNet 公司首席技术官 Brian Behlendorf 先生。



**Marc Canter**  
多媒体之父, 公认的多媒体奠基人之一, Macromedia 公司的创始人。



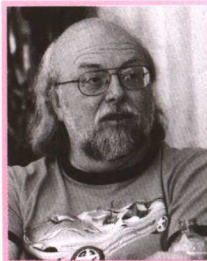
**陈黎夫**  
微软(中国)有限公司 Windows Live China 软件开发工程师。



**Alistair Cockburn**  
两度 Jolt 生产力奖获得者, 软件开发大师 Alistair Cockburn。



**西摩·克雷(Seymour Cray)**  
超级计算机之父。上世纪 70 年代, 克雷先后创造了 CRAY-1、CRAY-2 等超级计算机, 速度达到当时市场上其它机型的 40 多倍。



**James Gosling**  
Java 之父, Sun 公司首席技术官 James Gosling 先生。



**David H. Hansson**  
创造了名为 Ruby on Rails 的 Web 开发框架, 并因此当选 2005 开源软件 OSCON 年度最佳黑客(Hacker of the Year at OSCON)。



**Renee James**  
英特尔公司全球副总裁 Renee James 女士。



**陆海滨**  
美国盛瑞纳软件有限公司中国区总经理陆海滨先生。



**陆首群**  
中国开源软件推进联盟主席陆首群先生。



**马宁**  
微软最有价值专家, Windows Mobile 开发者。



**Eben Moglen**  
GPL 的创造者之一, 自由软件基金会(FSF)的法律顾问。



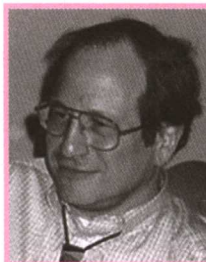
**Jonathan Pailey**  
Idaptive 公司 CTO 和创始人之一 Jonathan Pailey 先生。



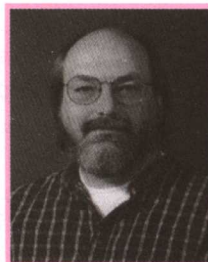
**潘加宇**  
UMLChina 首席专家, 潜心研究和实践 UML/UP 相关技术的应用。



**Robert Parker**  
微软工程院高级工程总监 Robert Parker 先生。



**Gerald Jay Sussman**  
人工智能专家, 麻省理工学院(MIT)教授, SICP 和 scheme 语言的发明人。



**Ken Thompson**  
UNIX 创始人之一, 1983 年图灵奖获得者。



**涂曙光**  
微软(中国)有限公司产品技术专家, 博客堂成员。



**王翔**  
全国海关信息中心开发部高级架构师。



**叶亮**  
系统分析员, SCJP、Java 工程师, 关注开源社区。



**袁传宽**  
中关村创新研修学院院长袁传宽博士。



**张宏江**  
微软工程院院长张宏江。



**郑雨林**  
用友软件副总裁、产品总监郑雨林先生。



**周立功**  
本土嵌入式领域著名企业家、周立功单片机发展有限公司创始人周立功先生。

## 人物&amp;报道

## 名人堂

|                                           |   |
|-------------------------------------------|---|
| 《设计模式》作者之一 John Vlissides 辞世 .....        | 1 |
| RSS 的守望者: Dave Winer.....                 | 1 |
| Craigs Newmark: Craigslist 背后的“书呆子” ..... | 2 |
| 影响算法世界的十位大师 .....                         | 2 |
| 搜索引擎的先驱者——Paul A. Flaherty.....           | 4 |
| 倡导自由软件运动的法律之手 .....                       | 4 |
| Gerald Jay Sussman: 计算机科学的引渡者 .....       | 5 |
| Seymour Cray: 隐居丛林的超级计算机之父.....           | 5 |
| Ken Thompson——UNIX 时代的开创者 .....           | 6 |
| Marc Canter——人生因为爱好而精彩 .....              | 7 |
| RoR 之父: David H. Hansson.....             | 7 |
| Jonathan Sachs: 平凡的程序员, 非凡的软件.....        | 8 |

## 报道

|                          |    |
|--------------------------|----|
| 共享软件第二次浪潮.....           | 9  |
| Flash Lite 挑战 J2ME ..... | 11 |
| 悼念王选: 技术创新树丰碑.....       | 12 |
| 谁会买下 Delphi? .....       | 14 |
| GPL 3.0 又一个霸权? .....     | 15 |
| 中国软件集体互联网攻略 .....        | 16 |
| 开源软件: 困境与出路? .....       | 19 |
| 程序员向流氓软件开枪 .....         | 20 |
| 有多少软件可以自由.....           | 21 |

## 人物专访

|                                             |    |
|---------------------------------------------|----|
| Michael Tiemann 专访.....                     | 23 |
| 是商业扼杀了语言的进步 .....                           | 25 |
| 微软 CTO Ray Ozzie 访谈.....                    | 27 |
| 自由软件的挑战和机遇 .....                            | 30 |
| 管理超大型服务的挑战 .....                            | 33 |
| 跨越鸿沟 .....                                  | 35 |
| 专访 Google 图像搜索创始工程师.....                    | 38 |
| 目的明确地进行系统开发 .....                           | 39 |
| JSF 与 JSP: Java EE5 的新特征 .....              | 41 |
| Java 前沿: 重构 EJB API Linda DeMichiel 专访..... | 43 |
| 网格: 海量存储的新秀 .....                           | 45 |
| 敏捷开发的本质 .....                               | 47 |
| 开源软件, 路在何方? .....                           | 48 |
| Java 将如何迎接新挑战.....                          | 50 |
| Robert Parker: 把微软软件开发方法带到中国.....           | 51 |
| 勇士要冲最大的浪 .....                              | 53 |

## 软件创业

|                         |    |
|-------------------------|----|
| Word 奇兵.....            | 55 |
| 网站商务通: 创业不负有心人 .....    | 56 |
| 与风险投资商第一次亲密接触 .....     | 57 |
| 软件创业的初期选型 .....         | 59 |
| i 桌面, 与微软 Live 争锋 ..... | 61 |
| 清华没有教我做婚介 .....         | 62 |
| 梦想的起点 .....             | 63 |
| 科技创新项目 .....            | 65 |



## 调查

|                          |    |
|--------------------------|----|
| 离职：面临尴尬的抉择 .....         | 68 |
| 程序员，你的 20% 时间在做什么？ ..... | 69 |
| 从华为员工劳累致死说起 .....        | 71 |
| 毕业了，向左走？向右走？ .....       | 72 |
| 蜕变，程序员转型的故事 .....        | 73 |
| 推开编程那道门 .....            | 74 |
| 从宾馆服务生到微软技术专家 .....      | 75 |
| 写诗机背后的思考 .....           | 75 |
| 2006 程序员大调查：得失一年间 .....  | 76 |

## 程序百味

|                                |    |
|--------------------------------|----|
| 开源：让自由为控制的欲望埋单 .....           | 78 |
| Ubuntu Linux 桌面 .....          | 78 |
| 承认多元化，保持宽容 .....               | 79 |
| 程序员创业三关 .....                  | 79 |
| 为什么 Linux 比 FreeBSD 更流行？ ..... | 80 |
| 太阳是否还会升起？ .....                | 82 |
| 迎接 P2P 分布式并行计算 .....           | 82 |
| 一切远未结束 .....                   | 83 |
| Java 开源说起 .....                | 84 |
| 技术换代新机遇 .....                  | 85 |
| 当我们不再年轻 .....                  | 86 |
| 程序员的臭毛病 .....                  | 86 |
| 让程序员成为正常人——加班篇 .....           | 87 |
| 时间缝里看技术发展 .....                | 88 |
| 最具行业渗透性的六种技术 .....             | 89 |
| 技术与市场的时差 .....                 | 91 |
| 透过职场看技术 .....                  | 95 |
| 从幻灯片到彩色电视机 .....               | 96 |
| 中国软件外包破局 .....                 | 97 |

## 最佳开源项目

|                                          |     |
|------------------------------------------|-----|
| 用程序解析天文 .....                            | 100 |
| 基于 Web 的灾难管理门户系统 .....                   | 101 |
| OpenQRM 第一个开源系统管理平台 .....                | 103 |
| sourceFORGE 最佳推荐项目 sourceFORGE.net ..... | 104 |
| OSDN 推荐项目 GForge.net.com.cn .....        | 105 |

## 声音&amp;幽默

|             |     |
|-------------|-----|
| 声音&幽默 ..... | 106 |
| 编程之道 .....  | 111 |

## 管理&amp;实践

## 项目实践

|                       |     |
|-----------------------|-----|
| 软件开发项目的需求和范围管理 .....  | 114 |
| 软件开发项目的计划制定 .....     | 116 |
| 微软软件设计规范书撰写技巧 .....   | 119 |
| 项目研发过程中的微软式管理控制 ..... | 121 |
| 软件开发过程中的测试管理 .....    | 124 |
| 敏捷模式在微软项目中的经验谈 .....  | 126 |
| TDD：金字塔上神像的光芒 .....   | 129 |
| 结构化和敏捷方法及过程修炼 .....   | 131 |
| 单元测试修炼 .....          | 135 |
| 单元测试中的框架与管理 .....     | 138 |
| 单元领域模型的描述与处理 .....    | 140 |

|                            |     |
|----------------------------|-----|
| 如何建设中小型 IT 企业的项目组织结构 ..... | 149 |
| 软件开发管理 .....               | 152 |
| IT 企业自主研发产品的立项管理方法 .....   | 154 |
| 一个项目经理的 2005 年工作总结 .....   | 157 |
| 从马岛战争看项目启动管理 .....         | 159 |
| 一个项目团队的敏捷之旅 .....          | 161 |
| 轻松实现软件产品的国际化 .....         | 163 |
| 中小型 IT 企业产品测试案例解析 .....    | 166 |
| 标签式界面风格的前世今生 .....         | 168 |
| 互联网企业的研发管理问题分析与对策 .....    | 171 |
| 让统一过程也敏捷 .....             | 174 |
| 软件架构概念思辨 .....             | 176 |
| 我的 Lean & Agile 项目经历 ..... | 178 |
| 软件界面设计中的 Fitts 定律 .....    | 180 |
| 交换编程 .....                 | 182 |
| 如何使界面支持多级用户 .....          | 184 |

### 团队故事

|                           |     |
|---------------------------|-----|
| Mop 研发团队：网络时代的技术偏执狂 ..... | 187 |
| Donews 游戏公会小团队的英雄梦想 ..... | 189 |
| 做第一门户背后的守护神 .....         | 191 |
| 浪潮楼上团队七日谈 .....           | 193 |
| 年轻的门户技术引擎 .....           | 194 |

### 需求分析

|                           |     |
|---------------------------|-----|
| 需求的“沙漏” .....             | 197 |
| 需求管理改进业务流程和产品开发过程 .....   | 199 |
| 走出需求分析的泥潭 .....           | 201 |
| 铅笔和纸最有效，而不是电脑 .....       | 203 |
| 从多角度看软件系统需求 .....         | 205 |
| 用 Use Case 表达架构观点 .....   | 208 |
| 用 Use Case 表达架构观点之二 ..... | 210 |
| 基于软件主板的系统整合之道 .....       | 212 |
| 结合用例，快速开发 J2EE 程序 .....   | 215 |
| 从用例到重构程序 .....            | 219 |
| 两个案例背后的分析与思考 .....        | 221 |
| 从涉众利益改善需求质量 .....         | 224 |
| 涉众需求的和谐获取之道 .....         | 226 |

### 管理圈

|                       |     |
|-----------------------|-----|
| 海滩上的思想者 .....         | 228 |
| “系统分析员”的去向 .....      | 229 |
| 萧规曹随也不错 .....         | 230 |
| 让客户来进行界面原型设计 .....    | 232 |
| 敏捷过程的三分之一 .....       | 235 |
| 福尔摩斯的有效决策 .....       | 236 |
| 激发团队的和谐特质 .....       | 237 |
| 高效开发的 6 个技巧 .....     | 239 |
| 阿司匹林 .....            | 240 |
| 从世界杯看软件团队建设和管理 .....  | 241 |
| 如何做一场成功的技术讲座？ .....   | 243 |
| 让我们 Pair 做任何一件事 ..... | 245 |
| 关注软件开发项目中的人 .....     | 246 |
| 外包测试中的管理学实践 .....     | 248 |
| “面霸”是这样炼成的 .....      | 250 |



## 产品&amp;应用

## 四书五经

|                       |     |
|-----------------------|-----|
| 软件开发项目管理指南 .....      | 252 |
| 移动开发图书概览 .....        | 252 |
| 底层技术开发 .....          | 253 |
| 算法之“四书五经” .....       | 255 |
| 敏捷软件开发图书概览 .....      | 257 |
| 搜索技术图书攻略 .....        | 258 |
| 嵌入式系统开发四书五经 .....     | 259 |
| World Cup .....       | 261 |
| Web 开发佳作点评 .....      | 263 |
| C++历史上最重要的图书 .....    | 263 |
| C++历史上最重要的非图书文献 ..... | 265 |
| 原版软件工程图书阅读指南 .....    | 266 |

## 经典工具推荐

|                                      |     |
|--------------------------------------|-----|
| 软件项目经理手中的利器 .....                    | 268 |
| 软件开发版本控制工具点评 .....                   | 270 |
| Java IDE: 苗条身材, 重在实用 .....           | 273 |
| 开源配置管理工具双杰 .....                     | 275 |
| 一个实战团队 Pragmatic 的项目工具集 .....        | 278 |
| 软件开发的“结对”助手 .....                    | 280 |
| XP 团队管理的利器 .....                     | 282 |
| 高手过招, 各有千秋 .....                     | 284 |
| 工具的力量 .....                          | 287 |
| 于精细处显真功夫 .....                       | 290 |
| 开源语法分析工具 .....                       | 292 |
| Open Workbench: MS Project 的杀手 ..... | 293 |
| 跟随 Teamwork 进入团队合作新境界 .....          | 294 |
| 浅谈工具在软件测试中的应用 .....                  | 296 |
| 用 Maven 简化你的项目管理 .....               | 297 |
| 防御 C++内存泄露的闸门 .....                  | 299 |
| GlassFish: Java 社区的新惊喜 .....         | 300 |
| 使用 Visual Studio 团队开发版进行项目管理 .....   | 303 |
| 为你的开发团队找个好管家 .....                   | 304 |
| wxWidgets: 全能的跨平台软件开发包 .....         | 305 |
| 初尝 Mylar .....                       | 306 |
| TestDriven.NET 2.0 .....             | 308 |
| 新产品、工具点评 .....                       | 310 |
| 项目管理软件的开发利器 .....                    | 311 |
| Web 报表工具评估宝典 .....                   | 312 |

## 程序员手册

|                          |     |
|--------------------------|-----|
| 2006 新技术辞典 .....         | 314 |
| 2006 年开发图书 TOP 100 ..... | 317 |
| 2006 国外优秀技术网站动态 .....    | 329 |
| CSDN 精彩 Blog 文摘 .....    | 333 |

## 年鉴

|                      |     |
|----------------------|-----|
| 2006 技术月历 .....      | 348 |
| 2006 年度书市观察 .....    | 375 |
| 2006 年软件业大事记 .....   | 388 |
| 2006 年中国开发者大调查 ..... | 393 |

## 附录

|                       |     |
|-----------------------|-----|
| 《程序员》2006 年全年目录 ..... | 403 |
|-----------------------|-----|

## 《设计模式》作者之一 John Vlissides 辞世

■ Mike / 文

2001年9.11事件发生的时候,《设计模式》(Design Patterns)作者之一 John Vlissides 在距离事发地点30公里的公司中听到这一消息,感到非常震惊。然而,仅仅几年之后,John Vlissides 在2005年11月24日在家中去世的消息传了出来,全世界各地的程序员也为此感到震惊,因为这时他年仅44岁。

John Vlissides 是 IBM's TJ Watson 研究中心职员,也是受人尊敬的软件专家。John Vlissides 最为人所知的贡献,无疑是他与 Eric Gamma、Richard Helm、Ralph Johnson 合写的经典著作《设计模式》。这本书于1994年由 Addison Wesley 公司出版,11年来一直是技术图书中一座难以逾越的丰碑。2005年7月30日,ACM组织的SIGPLAN协会基于 John Vlissides 写作《设计模式》一书的贡献,与其他几位作者一同获得了“2005年编程语言成就奖”。

John Vlissides 不但为这本书做出了重要的贡献,而且随后还撰写了《Pattern Hatching》,帮助读者更容易更深入地理解模式。此外,在学术性更浓的《Pattern Languages of Program Design》系列中,John 也担任了其中一本的主编。年纪轻轻的他有如此成就,归功于他的天才、经验和深刻的洞察力。早在1990年代初期,他就主持开发了 InterView 系统,在其中对于面向对象技术和设计模式的运用超越了当时的时代。John 认为软件技术仍然处于幼年期,他说:“看看传统工业发展起来需要多少时间,百年?千年?而我们才写了多少软件,50年?”

在纪念 John Vlissides 的 wiki 页上,可以看到关于他的更多信息。许多人士也到该 wiki 页留下对 John Vlissides 的悼念。Wiki 发明者 Ward Cunningham 留言说:John 是天赋之人,他给我们中的许多人留下极深刻印象。请在本页写下他如何帮助、支持或鼓励你的故事,以慰亡灵。敏捷开发模式创始人之一 Martin Fowler 留言说:在参加各种会议时,John 是我最愿意会见的人,我很喜欢和他相处,而且发现他是个易于相处的人。有一次他邀请我到 IBM 研究中心做一日之游,带我四处参观,非常好客。他也是个专注而文采甚丰的作者。没能和他共事更久我很遗憾。想到再也不能见面,我悲不自禁。

John 一直是一位乐于助人的人,对于很多素不相识的读者和爱好者,他从来都热情帮助。国内程序员中,有不少曾经与 John 通过邮件。不知道是不是过度的劳累摧垮了他,当年非常健康的他,突然身染恶疾(据说是癌症)。几年来,他一直在与疾病积极地抗争,但最终病魔还是带走了他。有位程序员表示:“今天还在公司看了设计模式,便听到这样的消息,哀悼,痛惜。”

无论如何,John 为软件技术走向成熟做出了重大的贡献。John Vlissides 表示自己生前有三个愿望,希望能够成为一个了不起的作家,一个了不起的厨师和一个了不起的作曲家。现在,我们可以作证,他实现了至少他的梦想一个——一位了不起的作家。(摘自2006年1月刊) ◀

## RSS 的守望者: Dave Winer

■ Shirley / 文

Dave Winer 是一个备受争议的人。讨厌他的人说他极度迂腐、过于敏感;崇拜他的人则认为他是一个头脑清晰、目的明确的领袖人物。这个年龄“奔5”、留着满脸大胡子的男人,在议论纷纷中不仅是第一代博客、主导了 RSS 技术的诞生与发展,而且作为一个纯粹、资深的程序员,他还参与了 SOAP、XML-RPC 的创建,更推出过商业内容管理软件 Frontier 和在苹果电脑时代备受欢迎的 ThinkTank 等。不过, Dave Winer 的所做所为始终围绕着一个主题——大纲编辑器(outliner),贯穿着他一向的行事作风——随心所欲。

1970年代,在威斯康辛大学计算机系就读的 Dave Winer 痴迷于大纲处理与脚本。后来,他开发出一个集成大纲编辑器与数据库的高级文档处理系统,能让数据库建立大纲等字段,而大纲可以作为数据库的索引。毕业后, Winer 怀揣着这个程序想投奔苹果,可惜苹果只对数据库感兴趣。一气之下, Winer 转投 Personal Software 怀抱,这是当时硅谷最热门的软件公司。可惜 PS 管理层最终决定不把大纲编辑器推向市场, Winer 只好带着 PS 给的遣散费在家闲赋。这时候,他尝试把自己的大纲编辑器变成一个在线系统,名为 LBBS(即活的BBS),允许多个用户把消息链接到树形结构上。

Winer 的遣散费已花得七七八八,不得不认真考虑把自己的“玩”意之作商品化,计划先推单机版本的 Thank Tank,获得成功后再推网络版的 MORE。1983年,带着银行账户上剩下的4千多美元、一本用户指南、两张苹果 II 主盘, Winer 成立了 Living Videotext 公司,尽管

他对如何操作软件生意还毫无概念。没想到 Think Tank 的销量奇好,即使后来受苹果机销量的影响 Macintosh 版本的销售一落千丈,但是用于 IBM 电脑的那个版本却不断增长。

面对飞速成长的业务, Winer 心情很复杂。尽管新技术的开发能带来很大的乐趣,但是繁杂的公司事务让 Winer 精疲力尽,完全没了私人生活,而且也使他的身体变得臃肿不堪。于是 Winer 把 Living Videotext 卖给赛门铁克,一边狂滑一个月的雪、减掉了差不多80斤的赘肉,一边思考成立新公司、开发新产品。

1989年, Winer 成立 Userland Software 公司,推出脚本语言 Frontier。最初 Frontier 的定位是用于 Macintosh 系统的免费软件,但没想到苹果出尔反尔,在新系统中绑定他们自己的脚本语言 AppleScript, Winer 决定把 Frontier 移植到 Windows 平台并开始收费。现在 Frontier 的内核遵守 GPL 协议,对外开放源代码。

在网络兴盛的1990年代, Frontier 成为支撑 Manila 内容管理系统的主要技术,允许用户接管(host)网站并通过客户端浏览器编辑它们。UserLand 提供免费的 Manila 接管服务,很快这就被大多数的人用来管理网络日志(WebLog)。Userland 也是第一代网络聚合器的提供者之一,让用户通过 RSS 能在一个页面上获得无数网络日志的内容。

这里,不得不提到 Winer 对 RSS 的贡献。早在1997年, Winer 提出了站点聚合模式 ScriptingNews。这种简单的 XML 格式,能让网页在不同的浏览器中被阅读。两年后网景推出应用 RDF 技术的 RSS 0.9,



但这种 XML 格式很复杂。所以，很快网景就去掉 RDF，融入 ScriptingNews 格式，推出至今也很流行的简化版 0.91。Winer 全程参与了这个版本的开发，贡献巨大。后来网景因看不到 RSS 的商业前景而放弃掉这项技术，而 Winer 却一直致力于 RSS 的发展和推广，并最终制定了 RSS 2.0。可以说，如果没有 Winer，就不存在 RSS，因为这种格式就会像通道定义格式（Channel Definition Format，早期微软

提出的网站聚合格式）一样无人问津。

也许正是因为 Winer 特立独行、固执己见，才会经常遭到别人挖苦和嘲笑。但是，他的一位支持者却这样说道：“Winer 在多个交叉领域有着建树：博客、新闻业、软件开发、政治、商业、学术等等。没有他的聪明才智和个人领导，互联网可能就会变得非常不一样，就没有现在这样的影响力。”（摘自 2006 年 2 月刊）←



## Craigs Newmark: Craigslist 背后的“书呆子”

■ Shirley / 文

“我们的首要任务不是挣钱，服务社区是我们的首要任务”，Craigs Newmark 说。你也许会觉得这样的说法太天真，然而 Craigs 就是凭借着这份天真、这份“书呆子”（nerd）气，创建了如今美国最大的分类广告网站——Craigslist.com。

这个分类广告加 BBS 的混合型网站，每月流量达到 30 亿人次。上面没有图片，只有密密麻麻的文字，标着各种生活信息，看上去颇为简单，但是它却是美国人最喜欢的网站之一。有人在这里卖掉自己的旧车，有人在这里租到中意的房子，有人在这里找到工作，有人在这里找到帮自己修电脑的人，还有人在这里找到女朋友。然而，维护这个庞大无比网站的只有 Craigs 和他的 17 名员工。

1994 年，在嘉信（Charles Schwab）银行从事计算机安全工作的 Craigs Newmark 一直在寻找一种提高自己交际生活的方法，因为从小到大他总被冠以“不合群”的头衔，社交能力一向不高。于是，他设计了一种 E-mail 列表，通过这个列表，Craigs 可以通知他的朋友将要在旧金山举行的各种文化和艺术活动的消息。随着越来越多的人要求加入，这个 E-mail 列表系统就无法承受如此巨大的发送量。Craigslist.org 随之诞生，人们通过这个网站可以随时查阅感兴趣的消息，这比发送 E-mail 方便很多。不久，Craigslist 上的内容也得到扩展，加入了旧货买卖、应聘招聘等信息。但这一切信息的发布和使用都是免费的，整个 Craigslist Craigs 和几名志愿者共同维护。

但是，只把 Craigs 设定为一个追求理想不顾现实的“书呆子”，未免太简单。事实上，Craigs 不是对商业一窍不通或者彻头彻底地看不起挣钱，他有自己的想法。“我对商业没有任何偏见，也不反对商业。对我来说，

做生意是基于一些价值观，我曾戏称它们为‘书呆子（nerd）价值观’。”

“书呆子的价值观非常简单。那就是，挣钱过上好生活是对的，对你的雇员好也是对的。我觉得一个人能做的最大的善事之一就是你能给别人创造一份工作。所以，你搞商业化运作可以，然而你还要有所不同。”

面对 Craigslist 来来去去的志愿者，Craigs 很明白人是无法长期坚持无偿工作的。1999 年，Craigs 决定停止志愿组织的形式，组建公司、聘用雇员。同时，Craigslist.org 更名为 Craigslist.com，并转为部分服务收费的网站：向旧金山及周边地区公司的招聘广告收取每条 25 美元的费用。这一收费后来涨到了 45 美元，并且最终涨到了目前的 75 美元。随后几年，Craigslist 逐渐扩展到纽约、伦敦等城市，成为当地人信息交流的重要方式。Craigslist 也准备向房屋买卖广告等内容收取费用。然而，Craigs 始终坚持一条原则：永远不对个人张贴的信息收费。

Craigslist 的走红也引起不少风险投资的青睐，很多人捧着上好的条件想要收购，但都被 Craigs 拒绝了，尽管他可以因此而成为千万富翁。“我承认当我想起这些钱能够让一个人享受些什么时，我的确有点痛苦。但是‘书呆子’价值观让我感到更开心：让自己生活得舒适就够了，然后该做一些事改变这个世界。”

2004 年 8 月，eBay 从 Craigslist 前雇员手中收购 Craigslist 25% 的股份。尽管 Craigs 不希望这事发生，但觉得是 eBay 而非其他公司收购还是让人安慰。他希望通过跟 eBay 的亲密接触能够帮助 Craigslist 更好得防范网络欺诈。对 Craigs 来说，能够取得经验，保护网络用户，维护“简单、温暖、相互信任”的网络社区，这才是他最关心的。（摘自 2006 年 3 月刊）←



## 影响算法世界的十位大师

■ 范凯 马林 / 文

### • Don E. Knuth

伟大的智者——Don E. Knuth，中文名：高德纳（1938-）算法和程序设计技术的先驱者。Oh, God! 一些国外网站这样评价他。一般说来，不知道此人的程序员是不可原谅的。其经典著作《计算机程序设计艺术》更是被誉算法中“真正”的圣经，像 KMP 和 LR (K) 这样令人不可思议算法，在此书比比皆是。难怪连 Bill Gates 都说：“如果能做对书里所有的习题，就直接到微软来上班吧！”

对于 Don E. Knuth 本人，一生中获得的奖项和荣誉不计其数，包括图灵奖、美国国家科学金奖、美国数学学会斯蒂尔奖（AMS Steel Prize），以及发明先进技术荣获的极受尊重的京都奖（Kyoto Prize）等等，写过 19 部书和 160 余篇论文，每一篇著作都能用影响深远来形容。Don E. Knuth 也被公认是美国最聪明的人之一。当年他上大学的时候，常写些各种各样的编译器来挣外快，只要有他参加的编程比赛，总

是第一名，同时也是世上少有的编程达到 40 年以上的程序员之一。他除了是技术与科学上的泰斗外，更是无可非议的写作高手，技术文章堪称一绝，文风细致，讲解透彻，思路清晰而且没有学究气，估计这也是《计算机程序设计艺术》被称为圣经的原因之一。

### • Edsger Wybe Dijkstra

谦逊的长者——Edsger Wybe Dijkstra，1930 年出生于荷兰阿姆斯特丹，2002 年逝世于荷兰纽南。他在祖国荷兰获得数据和物理学学士，理论物理博士学位，2000 年退休前一直是美国 Texas 大学的计算机科学和数学教授。以发现了图论中的最短路径算法（Dijkstra 算法）而闻名于世，1972 年因为 ALGOL 第二代编程语言而获图灵奖。“Go To Statement Considered Harmful”（EWD215）也是被广为传诵的经典之作。除了科学研究之外，他最喜欢做的事情就是教学，被人称作“一

天教学 24 小时”的教授。

且不说 Dijkstra 算法对计算科学、网络科学发展的深远影响,单从他在 1972 年获得图灵奖时的演讲“The Humble Programmer”,你就不得不肃然起敬,在获得计算科学中至高无上的奖项时,Edsger Wybe Dijkstra 仍然称自己不过是一个谦逊普通的程序员,何等胸襟,举世之中几人可比。

### • George Dantzig

运筹学大师——George Dantzig 可谓是由父亲一手培养出的天才。George 的父亲是俄国人,曾在法国师从著名的科学家 Henri Poincaré。他曾经这样回忆自己的父亲:“在我还是个中学生时,他就让我做几千道几何题……解决这些问题的大脑训练是父亲给我的最好礼物。这些几何题,在发展我分析能力的过程中,起了最重要的作用。”

在伯克利学习的时候,有一天 George 上课迟到,只看到黑板上写着两个问题,他只当是课堂作业,随即将问题抄下来并做出解答。六个月后,这门课的老师——著名的统计学家 Jerzy Neyman——帮助他答案整理了一下,发表为论文,George 这才发现自己解决了统计学领域中一直悬而未决的两个难题。

George 后来在运筹学建树极高,获得了包括“冯诺伊曼理论奖”在内的诸多奖项。他在 Linear programming and extensions 一书中研究了线性编程模型,为计算机语言的发展做出了不可磨灭的贡献。天妒英才,他于 2005 年 5 月 13 日去世。

### • James Cooley

推动时代前进的人——James Cooley (1926-) 美国数学家,哥伦比亚大学的数学博士,以他所创造的快速傅里叶变换(FFT)而著名,不能不说是意义极其重大,FFT 的数学意义不光在于使大家明白了傅里叶(Fourier)变换计算起来是多么的容易,而且使得数字信号处理技术取得了突破性的进展,对于现在的网络通信,图形图像处理等领域的发展与前进奠定了基础。Fourier 变换的意义在于将电能变为了工业的命脉,而 FFT 的意义更是在于做推动了整个社会信息化的进程。在 IBM 研究中心中主要从事数字信号处理的研究一直到 1992 年退休,同时他还是 IEEE 的数字信号处理委员会的成员。1980 年获得 ASSP's Meritorious Service Award, 1984 年获得 ASSP Society Award 以及 IEEE Centennial Medal。

### • John Backus

FORTRAN 之父——John Backus 早年在 Hill School 学习的时候因为讨厌学习,成绩一塌糊涂而不得不在暑假补课。1943 年,他在父亲的要求下到维吉尼亚大学学习化学,随后参军、照顾头部受伤的伤员、在医学学校学习治疗,可是最后又都放弃了。不过还好,战后 Backus 进入纽约哥伦比亚大学学习数学,并于 1949 年毕业。在毕业前夕,他跑到了麦迪逊大街的 IBM 计算机中心参观。事情凑巧,和导游聊天的时候 Backus 谈到自己正在找工作,在导游的鼓励下,他和中心一位主管的面谈,成为了一名 IBM 的程序员。

在 IBM, Backus 的才华得到了施展,发明了人类历史上第一个高级语言——FORTRAN。接着,又提出了规范描述编程语言语法的 Backus-Naur Form (BNF)。这位当年的“差生”终于被整个计算机业界肯定——美国计算机协会于 1977 年授予 John Backus 图灵奖。

### • Jon Bentley

实践探索先锋——Jon Bentley 1974 年获得了斯坦福大学的学士学位,1976 年获得北卡罗莱纳大学的硕士和博士学位。毕业后在卡内基梅隆大学教授了 6 年计算机科学课程,1982 年进入贝尔实验室。2001 年退休后加入了现在的 Avaya 实验室。他还曾作为访问学者在西点军校和普林斯顿大学工作。他的研究领域包括编程技术、算法设计、软件工具和界面设计等等。

他写作过三本编程书籍,其中最著名的就是涵盖从算法理论到软件工程各种主题的 Programming Pearls (《编程珠玑》),这其实是他发表过的文章的合集。在这些文章里,Jon 从工程实践的角度出发,为程序员们提供了一个个艰难问题的解决方案,犹如一颗颗闪闪发亮的珍

珠。Bentley 的珍珠超出了可靠工程学的范畴、利用他的洞察力和创造力为那些恼人的问题提供了独特而巧妙的解决方案。

### • Nicklaus Wirth

PASCAL 之父——Nicklaus Wirth,如果说有一个人因为一句话而得到了图灵奖,那么这个人应该就是 Nicklaus Wirth,这句话就是他提出的著名公式“算法+数据结构=程序”。这个公式对计算科学的影响程度足以类

比于物理学中爱因斯坦的“ $E=MC^2$ ”——一个公式展示出了程序的本质。Nicklaus Wirth, 1934 年出生于瑞士,1963 年在加州大学伯克利分校取得博士学位。取得博士学位后直接被以高门槛著称的斯坦福大学聘到刚成立的计算机科学系工作。在斯坦福大学成功的开发出 Algol W 以及 PL360 后,爱国心极强的 Nicklaus Wirth 于 1967 年回到祖国瑞士,第二年在他的母校苏黎世工学院他创建与实现了 PASCAL 语言——当时世界上最受欢迎的语言之一。后来他的学生 Philippe Kahn 毕业后和 Anders Hejlsberg (Delphi 之父)创办了 Borland 公司靠 Turbo PASCAL 起家,很快成为了将 Borland 发展成为全球最大的开发工具厂商,这一切都不得不说是归功于 PASCAL 语言的魅力。PASCAL 已经影响了整整几代的程序员, Nicklaus Wirth 的思想还将会继续指引现在和以后的程序员前进的方向。

### • Robert Sedgewick

算法的讲解者——Robert Sedgewick 是普林斯顿大学的计算机科学教授,他还是 Adobe Systems 的一名主管,也曾作为访问学者在 Xerox PARC、IDA 和 INRIA 工作。他在斯坦福大学获得博士学位。他的著作包括 Algorithms in C、Algorithms in C++、Algorithms in Java 等系列书籍,这些都再版多次。“没有人能够将算法和数据结构解释得比 Robert Sedgewick 更清楚易懂了!”很多读过他著作的程序员这样说。

目前 Robert 正在研究算法设计、数据结构、算法分析等方面的基础理论。他善于通过数学方法评估和预测算法性能,设法发现算法、数据结构的通用机制,例如使用逼近方法寻找更快速更高效的算法。另外,他还将算法和图形学结合起来,例如使用可视化方法评估算法效率,算法的图形化模拟,用于出版物的高质量算法表现方法等等。

### • Tony Hoare

计算机领域的爵士——Tony Hoare, 1934 年出生于英国,1959 年博士毕业于俄罗斯莫斯科国立大学,获得语言机器翻译专业学士学位。1960 年发布了使他闻名于世的快速排序算法(Quick Sort),这个算法也是当前世界上使用最广泛的算法之一。

Tony Hoare 在取得博士学位后,就职于 Elliott Brothers,领导了 Algol 60 第一个商用编译器的设计与开发,由于其出色的成绩,最终成为该公司首席科学家。从 1977 年开始, Tony Hoare 博士任职于牛津大学,投身于计算系统的精确性的研究、设计及开发。因其对 Algol 60 程序设计语言理论、互动式系统及 APL 的贡献,1980 年被美国计算机协会授予“图灵奖”。

1999 年在牛津大学退休后, Tony Hoare 博士被微软剑桥研究院聘请担任高级研究员,从事微软剑桥研究院研究成果的工业化应用的工作,以及协助其他研究人员进行服务于软件产业及用户的长期基础研究项目。2000 年因为其在计算机科学与教育上做出的贡献被封为爵士。

### • Udi Manber

首席算法官——世界上还有如此奇怪的职位?但是对于 Amazon 乃至 Google 来说,就一点也不奇怪。Udi Manber, 这位前 Amazon 的“首席算法官”,现在是 Google 负责工程事务的副总裁。他研究 WWW 的应用程序、搜索以及隐藏在这背后的算法设计。在此期间,他与其他人共同开发了 Agrep、Glimpse 和 Harvest 等 UNIX 上的搜索软件。1998 年, Udi 成为了 Yahoo! 的首席科学家。2002 年, Amazon 创造性地给了 Udi“首席算法官”的职位,和 Udi 为 Amazon 的“Search Inside the Book”搜索项目所做的工作相得益彰。

Udi 还因为他所著的 Introduction to Algorithms——A Creative Approach 而被大家称道。(摘自 2006 年 4 月刊) ◀



# 搜索引擎的先驱者——Paul A. Flaherty

■ 倪志刚 / 文

又是阳春三月，又有一位天才的程序员离我们而去了。今年3月16日，Paul A. Flaherty，搜索引擎的先驱者，AltaVista的创造者，猝于心脏病突发，享年42岁。

Paul A. Flaherty，1964年，出生于美国密尔沃基，1994年在斯坦福取得电机工程的博士学位。1995年在他31岁时，和大家一起创造出了世界上第一个搜索引擎——AltaVista。此后的十年中，虽有Yahoo，Google等搜索引擎先后崛起，但是AltaVista一直保持搜索引擎领域中的长青树的地位。Paul Andrew Flaherty这个名字也随着AltaVista被永久载入了IT史册。

时间转回到1995年的春天，Paul离开斯坦福，加入DEC的网络系统实验室。当时的Internet还只是新生事物，整个网络上不过只有区区1000万张网页，那时的Yahoo还只不过是放了一组链接在主页上，Google这个Idea还只存在于某些人的潜意识中。但是凭着对网络与技术的深刻认识，Paul意识到：随着Internet上的Web页面越来越多，信息的查找会变得越来越困难。正在此时，DEC推出了新的机型Alpha 8400 (TurboLaser)，这个机型太过优秀，运算能力之强超过了当时的应用需要，以致于不知道该如何去推广它。正是在这种情况下，一个新的Idea就在Paul脑海里形成了：为什么不将Internet网络上的关键字与链接都存到一个巨大数据库中，然后再构建一个程序（即搜索引擎）来提供搜索服务呢？这样既可以解决网络的搜索问题，也正好可以用来显示Alpha机型的优越性。

这个好主意得到了公司上下的赞同，于是在1995年夏天为此成立了一个名为AltaVista的项目小组，由Paul出任项目组的技术主管。经过两个月的内部测试后，AltaVista项目建立了一个拥有约16,000,000个页面的索引，12月份AltaVista正式公布于众。AltaVista确实是太优

秀了，深受网民的喜欢。三周之内，AltaVista站点的访问请求量就达到两亿次。到1996年5月，网页索引量达到了30,000,000个页面，每天有1200万次的访问请求量。到11月时，日点击量更是超过了2200万次。

随着AltaVista的成功，Paul也成为了AltaVista公司的CTO一直到2000年离开。后来AltaVista项目的同事回忆说，“Paul是我所见到的少数几个不可思议的人之一，他能将他的聪明与技术敏感结合起来，可以推动一个想法直到实现。他的活力与鼓励也是推动着整个AltaVista项目组前进的力量之一。这是真正创造者的本色。”

可惜AltaVista算是生不逢时，做为一个商业项目AltaVista并没有取得成功。在互联网的泡沫时期，DEC曾经考虑让其上市，却在投资银行家的批评下只得作罢。更有甚者，在其巅峰时期，竟然连自己的域名也没有，AltaVista的所有权也是三年之内三易其主。虽然如此，大家都公认AltaVista是.com时代的创造者之一，由此可见其历史地位之重要。

不论AltaVista是否是成功的商业项目，但Paul Flaherty一直被大家公认是一位伟大的软件工程师，深受家人，朋友的爱戴。同时Paul还是一个兴趣和爱好非常广泛的人，在上学期间，他就积极参加各种演讲和辩论活动，充分锻炼各方面的能力。对计算机，无线电科学也一直保持着相当浓厚的兴趣。他还是一个不折不扣的铁路迷，一生中拍过许多的铁轨照片。除此外，他还酷爱户外运动，喜欢飞碟射击，婚后更是和夫人充分享受周游世界的乐趣。

可惜，天妒英才，这么优秀的一个却过早的离开了我们，真是令大家感到扼腕叹息。

最后以一首他做过的诗来表达对其的缅怀之情。（摘自2006年5月刊）←

## 倡导自由软件运动的法律之手

■ 倪志刚 / 文

“我是一名历史学家，一名计算机程序员，但是我主要是一个律师。我主要做的事情是：通过消除在知识产权保护法庇护下非法力量从而创建自由。”Eben Moglen——GPL的创造者之一，自由软件基金会（FSF）的法律顾问，在“Free Software, Not the Other Thing”中这样写道。

Eben Moglen的一生也可以算得上是具有传奇色彩了。在他12岁时就对计算机编程产生了浓厚的兴趣，随后在1973年，年仅14岁的他开始了职业程序员的生涯。到了1979年，他就能独当一面。在此后的5年中，他供职于IBM的实验室和研究中心，致力于高级语言的设计与开发。在此后10多年的程序员生涯中，他积累了足够多的软件开发经验，更重要的是，他为另一个目标——成为一名律师，积累了足够多的学费。

1980年，Eben Moglen在Swarthmore College取得了法律学士学位，并在此期间获得了文艺评论的“Hicks Prize”。5年后他又从耶鲁大学取得哲学的硕士和法学博士学位。以后的几年里他在纽约联邦地方法院和美国最高法院担任法官助理。1987年他在哥伦比亚法律学校开

始任教，担当副教授教法律学。到了1993年，他从耶鲁又取得了一个历史学博士学位，随后一直在哥伦比亚法律学校担当法律和法律的教授直到今天。Eben Moglen回忆说，在耶鲁期间是人生一段很长的黑暗时期，对人生感觉到迷茫。

时间转到1991年，在哥伦比亚工作的时候，Moglen处理了人生中第一个关于自由软件的重要案件。这个让Eben Moglen一直引以为豪的案例，也是成为他一生中最重要的一个转折点。当时他在阅读BBS时，发现了Pretty Good Privacy，一个由Phil Zimmerman编写的电子邮件加密程序，程序员出身的Moglen对这个软件有非常深刻的印象，同时，作为一名专业的法律工作者，他立即意识到Phil Zimmerman可能会在法律方面遇到麻烦，于是他主动给Zimmerman发邮件，表示愿意无偿提供法律帮助。果然，不久后美国政府开始控告Zimmerman，认定他将PGP软件在互联网上公开的做法是违反美国法律的。Eben Moglen为了大众的利益而帮助了Zimmerman，使得美国政府最终撤销了这一诉讼。

正是这个事件，让Moglen意识到，自由软件的存在和发展以后将

会越来越强大，并且将会是以后软件最重要的存在形式之一。也因为 Zimmerman 事件的影响，当时 Richard Stallman (FSF 的创始人，自由软件之父)，联系到 Moglen，希望他能提供法律上的帮助。Moglen 再一次无偿的帮助了 Stallman，协助他起草并创建了 GPL。至此为止，Moglen 找到了他人生中最重要的一个方向。随后 Moglen 应 Richard Stallman 的邀请加入 FSF 担任法律顾问，开始了为自由软件开发者提供法律帮助的事业。2005 年 2 月 Moglen 协助创建了“自由软件法律中心”(Software Freedom Law Center)，并在其中担任 Public Patent Foundation 的一名主管，这个法律中心无偿为自由软件开发者提供需要的法律援助。

基于其程序员与律师的背景，Moglen 开启了他自由软件法律援助之门，他说：“我所付出的时间不会比自由软件开发者为了改进程序所付

出的时间更多，我之所以花时间帮助他们是因为：有些工作必须由程序员来完成，而有些工作却只能由律师来完成。但是愿意无偿工作的律师比无偿工作的程序员少得太多了。”在自由软件法律中心中，Moglen 和他们的同事一起提供的都是无偿的服务，对于能进入法律中心的律师要求也比较特殊，除了法律知识外，同时还要求具备一些程序员的知识，因为他们这些律师需要和程序员打交道，而 Moglen 认为，只有对代码有着相同理解的人，才会不厌其烦的为软件开发者提供更优质的服务。

也正是由于 Eben Moglen 在自由软件上所做的贡献，2003 他被授予了 EFF Pioneer Award。他也将继续为了捍卫自由软件而高举自己的法律之手。(摘自 2006 年 6 月刊) ◀

## Gerald Jay Sussman: 计算机科学的引渡者

■ 倪志刚 / 文

“如果你能够下载脑中一切的内容，到一台机器里，那台机器，就变成了你。让你身体其他的部分死去吧！反正那些部分并不那么的有趣。从此，那机器（你）可以长生不死，就算硬体不能达到永恒，你总是可以把所有资料备份下来，在第一台机器坏掉的时候，转载到新机器上……每个人都希望能达到永生的境界，不幸的，我想我是属于最后一个还会死亡的时代。”

看完这段话，很多人也许会以为出自于某科幻小说，实际上这段话是出自于 Gerald Jay Sussman——人工智能专家，麻省理工学院 (MIT) 教授。

Sussman (苏斯曼)，1968 年从 MIT 取得计算机学士学位，五年之后，他又从 MIT 取得了博士学位。1964 年开始投身到 MIT 的人工智能研究当中，在从事人工智能的研究的这 40 多年中，他为整个计算机科学界贡献了无数妙想与成果，很多知名计算机科学家都是他的学生。所有这些成果中，最被大家关注的是 SICP 与 scheme 语言。

《structure and interpretation of computer programs》简称 SICP，中文名为《计算机程序的构造与解释》，Sussman 是该书的作者之一。这本书是如此的优秀，自出版以来世界各地 100 多所院校包括斯坦福、普林斯顿大学、牛津大学、东京大学等许多著名高校都采用该书做教材。这本书还被翻译为多种语言，包括法文、德文、中文、日文等。MIT 更是从 1981 年开始将 SICP 的课程作为计算机学科就业前的必修培训课程之一，1986 年在 Sussman 给 HP 做培训时，HP 的电视台将其拍摄下来，制作成为教学录像，以后该教学录像一直广泛被 HP 及其他公司、高校及 MIT 作为企业培训使用。

正是因为本书，以及 Sussman 在计算机教育上的贡献，1990 年他被 ACM 授予 Karl Karlstrom Outstanding Educator Award (杰出教育者奖)。不少人在评价这本书时说道：SICP 展示了计算机科学的核心所在，它是用来教你如何成为一个计算机科学家，而不是程序员。每一

位严肃的计算机科学家都应该阅读本书，它适合所有希望深刻理解计算机科学的人们。

Sussman 对计算机科学另一个贡献是 scheme 语言。scheme 作为 Lisp 语言的一个分支，于 1975 年由 Sussman 和他的学生 Guy Steele, Jr 一起完成。Scheme 的理念是程序语言不是拿来“学”的，而是拿来“用”的。scheme 诞生后，由于其具有所有 Lisp 的优点，而且干净、强大、稳定、特别适合于用于描述算法，以至于 Scheme 在教育界被广泛的使用，新一代优秀的计算机科学家中很多人的“母语”就是 Scheme。

由于使用这种语言教学，学生可以学会“解决现实世界的问题”而不是困惑于“电脑自己的问题”。学生会很快的理解到“计算”的实质内涵，深刻体会到计算科学是什么，所以 Scheme 被很多大学用于各个领域的计算机教学，甚至一些中学和小学也开始利用 Scheme 来讲授计算科学。令人惊奇的是一些 10 岁小孩可以用 Scheme 写出 Don Woods 的 adventure 游戏。

在人工智能的研究中，Sussman 看到了人工智能的一些思想能被用于计算机辅助设计中，于是和他的另一个学生一起开发了用于 VLSI (超大规模集成电路) 设计的复杂计算机辅助设计工具。Sussman 还是数字太阳系仪的主要设计者，这个太阳系仪主要用于轨道力学实验来做一些高精度的积分。由于采用了人工智能的思想与工具，这个太阳系仪只用了几个人，几个月的时间就建成了。使用这个数字太阳系仪，Sussman 与 Jack wisdom 一起发现了好几个外太空无序运动的证据。

Sussman 是 IEEE, NAE, AAAS, AAAI, ACM, NYAS 等的会员，除了这些与科学相关的团体外，他还是一个专业的锁匠，钟表制造协会的会员，业务的望远镜制造者以及无线电协会的会员。无可争议，Sussman 确实是个多才多艺的天才。(摘自 2006 年 7 月刊) ◀

## Seymour Cray: 隐居丛林的超级计算机之父

■ 倪志刚 / 文

一个国家超级计算机的研制水平，是国际上用来衡量其科技水平最重要的标志之一。超级计算机一直以来应用于工业、科研、国防中

最尖端的一些领域。在所有超级计算机的研制者中，西摩·克雷 (Seymour Cray) 被誉为是无可非议的“超级计算机之父”。



1925 年克雷出身于美国一个工程师家庭,他从小喜爱电子学,中学期间更是长时间泡在电气实验室,对摆弄各种电气设备、收音机、电动机以及各种电路乐此不疲,以至于极少参加各种社会活动。小时候的经历似乎已经预示了他将来一定会走出一条出色的技术专家道路。

中学毕业后,18 岁的克雷加入部队,当时正值二战末期,他到过世界上不少的战区。二战结束后,克雷回到美国,在威斯康星大学上学,不久,又转到明尼苏达大学,1951 年取得了该大学的电气工程学士学位,由于感到课程太缺乏数学内容,其后又花了一年时间学习数学,1952 年他拿到了数学硕士学位。这时老师给了他一个建议说,如果我是你,我会沿着这条路走下去,一直走进工程研究协会(ERA)。克雷听从了老师的建议,从此也奠定了其一生事业的发展方向。

随后克雷加入了 ERA,负责建造密码设置。当时的克雷对还对计算机一无所知,为了完成目标,他一头扎进图书馆,开始吸收各种计算机的知识。不久后,他设计出他的第一台计算机 EAR 1103。但没过多久,ERA 被收购,克雷发现新公司与他自己的目标——研制第一流的超级计算机相差太远,最后克雷选择了离开。

1957 年,克雷和其他几位 ERA 的同事辞职后,创办了 CDC。在克雷刚开始 CDC 1604 的设计时,电子管问世了,不过价格很高。面对资金短缺的问题,克雷不得不在 CDC 1604 的建造中使用了不少的废弃的二手晶体管。尽管如此,CDC 1604 还是成为了当时最好的计算机,也是独一无二的晶体管超级计算机,不能不说,这是一次伟大的成功。随着 CDC 1604 的商业成功,克雷的声望与日俱增,过多的行政事务与社会活动使他不能专心于他的目标。于是克雷和公司达成协议,在他老家威斯康星的丛林中建立一个实验室,在此后的整整 4 年中,克雷谢绝一切社交往来,潜心于超级计算机的研究,为此别人送他一个“丛林隐士”的外号。他的隐居没有白费,1963 年 CDC 6600

的推出,使得 CDC 公司成为了市场上真正的主导者,再到后来 CDC 7600 的推后,这台每秒运行 1000 万次的机器被公认是当时第一台真正意义上的超级计算机,克雷也成了举世闻名的超级计算机专家,CDC 公司从此开始独霸整个超级计算机市场。

CDC 在当时的成功,也导致了其将目光转移到了计算机的商业应用,不再重视超级计算机的研制。这使得坚持自己信念的克雷第二次的选择了离开。1972 年,克雷自立门户,创立了克雷研究公司,公司只生产超级计算机。此后的十余年中,克雷先后创造了 CRAY-1, CRAY-2 等机型,他亲手设计了 Cray 机型的全部硬件与操作系统,其中的作业系统更是用机器码编写完成。在 CRAY-1,他第一次采用了集成电路来制造,速度达到了每秒钟 2.4 亿次,是当时市场上其他机型的 40 多倍速,价格却相差无几,这使 CRAY-1 取得了空前成功,其后推出的 CRAY-2 更是比 CRAY-1 功能强大了 4~6 倍,运行速度达到了每秒 12 亿次的浮点运算,以前需几年运算才能完成的工作在 CRAY-2 上只需要 1 秒钟。到了 1984 年时,克雷公司占据了超级计算机市场 70% 的份额。

但好景不长,20 世纪 80 年代后期,PC 的发展如日中天,大型机和超级计算机都受到了较大的冲击,最终 CRAY-3 在商业市场上惨淡收场。此时克雷再一次与公司的意见出现分歧,1989 年他退出了自己一手创办的克雷研究公司,另行成立了克雷计算机公司,全力研究 CRAY-4,但是这个设计目标为每秒 1000 亿次的机型最终没有完成,1995 年克雷计算机公司被迫宣布破产。

面对事业的几次大起大落,克雷一直没有放弃过,到 1996 年,70 余岁的克雷再一次创办了 SRC 公司,希望再一次能为世界创造出奇迹,可惜天不作美,厄运突然降临,1996 年 9 月,一场意外的交通事故让克雷永远的停止了呼吸,享年 71 岁。(摘自 2006 年 8 月刊) ◀

## Ken Thompson——UNIX 时代的开创者

■ 倪志刚 / 文

Ken Thompson, 1943 年出生于美国新奥尔良。1960 年,进入加州大学伯克利分校主修电气工程。1965 年毕业后,又花了一年时间在该校取得了电子工程硕士学位。在 Ken 读书期间,正好是计算机时代的起步阶段,自小喜欢电气的 Ken 接触到计算机后,立即沉迷其中,从 1962 年开始,就在学校计算机中心找到份工作,专门负责程序编写。为其后他一手开创的 UNIX 时代奠定了良好基础。

1966 年离开校园的 Ken 加入了贝尔实验室。那时的计算机系统还是批处理的天下,程序员只能在又慢又笨重的大型机上工作。应市场需要,当时贝尔实验室与麻省理工学院以及通用电气公司联合开发了一个多用户分时操作系统,取名为 Multics(多路信息计算系统),Ken 当时就是这个系统的开发人员之一,在开发 Multics 的期间,Ken 创造出了名为 Bon 的编程语言。可惜因为这个系统不但开发周期长,成本高,而且庞大而缓慢,最后贝尔实验室从这个项目中撤了出来。

面对此情此景,Ken 作为一个创造者的本性立即体现了出来,Ken 决定要开发一个全新的操作系统。利用 PDP-7 上的汇编语言,Ken 只花了一个月就编写完了操作系统的内核,在这个一个月中,他一周一个内核,一个文件系统,一个编辑器和一个编译程序的完成。做完这个系统后,Ken 将其命名为 UNiplexed Information and Computing System,缩写为 UNICS,后来做了一下改动,称为 UNIX,在开发第一版 UNIX 的过程中,Ken 还开发出一种新的语言,即 C 语言的前身——B 语言,这种语言简洁明了,接近于硬件语言,第一版 UNIX 就是基于 B 语言来开发的。

UNIX 的出现并不为大家所看好,但是却引起了贝尔实验室另一位同事的注意,这就是 Dennis M. Ritchie,于是 Dennis 主动加入了进来共同完善这个系统。至此一场轰轰烈烈的 UNIX 的传奇时代才真正的拉开了序幕。1972 年,他们联手将 UNIX 移植到当时最先进的大型机 PDP-2 上,由于 UNIX 的简洁、稳定与高效,以至于当时大家都放弃了 PDP-2 上自带的 DEC 操作系统,完全改用 UNIX。在 1973 年之前 UNIX 还不太为外界所知,到同年 10 月,UNIX 在 IBM 举办的操作系统原理专题研讨会上被提及,当 Ken 和 Dennis 在会上宣读论文并展示 UNIX 后,整个会场轰动了,大家都立即涌上来索取这种新型的操作系统的程序。随着 UNIX 的需求量的日益增加,Ken 与 Dennis 决定将 UNIX 进一步改写,以便可以移植到各种不同的硬件系统,由于 UNIX 的原码中不少是用汇编完成,不具备良好的移植性,正好 Dennis 在 1973 年在 B 语言的基础上开发出了 C 语言,C 语言灵活,高效性,与硬件无关,并且不失其简洁性,正是 UNIX 移植所需要的法宝,于是旧版的 UNIX 与 C 语言完美结合在一起产生了新的可移植的 UNIX 系统。随着 UNIX 的广泛使用,C 语言也成为了当时最受欢迎的编程语言一直到延续至今。

说到 UNIX 与 C 语言,还有一段小故事,当时安装了 UNIX 的 PDP-11 被放在贝尔实验室供大家使用,有一天大家发现 Ken 总是可以得到最高的权限轻松进入他们的帐户,于是有若干高人跳了出来,仔细分析 UNIX 代码,找到后门,修改后再重新编译整个 UNIX,却发现 Ken 还是很容