

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

增值

程序员

2004合订本

Programmer 上

《程序员》杂志社 编

合订内容:

《程序员》全年12期精华
《开发高手》全年12期精华

超大增值内容:

新名词解析
流行工具手册
百本书评
程序员手册
2004开发年鉴
2004开发类网站精萃



盘(2CD):

《程序员》12期PDF及源码下载
《开发高手》12期PDF及源码下载
10本技术开发经典图书
编程规范
知名开源软件介绍与源码下载
《程序员》全年技术沙龙视频
Sun公司技术社区技术培训课程



Csdn.net



电子工业出版社
Publishing House of Electronics Industry
http://www.phei.com.cn

程序员

2004 合订本(上)

《程序员》杂志社 编

总策划: 蒋 涛

编委会: 唐 琦

程 琰

欧阳璟

刘彦博

韩 磊

霍泰稳

罗景文

孟 岩

常 可

刘 婧

闫 辉

张浩祥

赵建凯



电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

程序员 2004 合订本 (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 2004 合订本 (上) / 《程序员》杂志社编. —北京: 电子工业出版社, 2005.1
ISBN 7-121-00648-0

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 (2004) 第 125478 号

责任编辑: 孟迎霞 熊妍妍

印刷: 北京牛山世兴印刷厂

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

经销: 各地新华书店

开本: 850 × 1168 1/16 印张: 51 字数: 800 千字 彩插: 8

印次: 2005 年 2 月第 2 次印刷

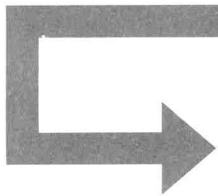
印数: 3900 册 定价: 45.00 元 (上、下册 + 2CD)

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: 北京市朝阳区北三环东路静安中心 26 层 (100028)

电话: 010-84480948/84540231~33 转 279

传真: 010-84540263



程序员增值合订本内容简介

本着精心组织、全新策划、超值合订、强力推出等四大原则,《程序员》杂志社编辑部组织最强阵容全心投入,历时两个月整理编制出《程序员2004合订本》这一众多软件开发爱好者期待的书籍。《程序员2004合订本》全套共832页,分手册增值版、技术专题版上下两册,并附送两张配套光盘。相关具体内容介绍如下:

► 手册增值版

人物 & 报道: 包括《程序员》的经典品牌栏目名人堂、特别策划、技术访谈、走向海外、软件创业及程序百味等。在这个栏目里可以品味软件业成功者的酸甜苦辣, 程序员的感悟与梦想。

管理与实践: 主要包括软件工程、交互设计、项目管理及企业应用等与软件开发相关的技术话题, 2004 年度最新的开发模式、测试模式及管理方法均集于此。

程序员手册: 在这个栏目里我们全新组织了2004 年最新名词, 如何利用多种技术混合进行软件开发及国内知名技术网站精粹与热贴索引等内容, 具有很高的实用价值。

2004 年鉴: 2004 年的中国软件业有复苏的景象, 在这一年中软件业究竟发生了哪些大事, 在这个栏目中有一个全面的检索。而2004 年百本好书与2004 程序员薪资报告相信会给关心软件的朋友在技术书籍与业界薪资方面一个很好的参考。

► 技术专题版

《程序员》技术专题: 每期的技术专题为把《程序员》打造成中国软件开发类最有影响力的专业杂志起了很大的作用, 在这本合订本里我们完全重现了2004 年《程序员》全年的技术专题, 以飨读者。

《开发高手》技术专题: 《开发高手》的定位是高校里热爱软件技术的学生, 所以其文章都很有实用性, 每期的技术专题也很有特色, 可限于篇幅我们只是选择了几期有代表性的文章, 但全部内容会在光盘里找到。

技术专区—专题: 这一部分包容了Java、Open Source、.NET、REBOL、XML及数据库等软件开发技术内容, 皆为《程序员》与《开发高手》的精华。而游戏开发、移动开发及网络与安全等三个小的专题栏目则将当前软件开发的热点和盘托出。

► 光盘介绍

CD-1:

本光盘内容主要包括了《程序员》12 期PDF 电子文件及全部源码下载, 《开发高手》12 期PDF 电子文件及全部源码下载, 还有CSDN网站推出的《Oracle 专刊》电子版, 常用开源软件介绍与安装文件等, 特别适合开发者学习、查阅和收藏。

CD-2:

本光盘内容涵盖了著名技术专家2004 年讲座实录及优秀技术培训课程实录, 大容量的视频、音频文件给辛苦一年的软件开发者带来一顿精美的盛宴。

首先感谢您购买和阅读《程序员2004 合订本(上、下)》, 在阅读过程中如果您有好的建议与想法可通过参与“《程序员2004 合订本(上、下)》有奖调查”活动反馈给我们, 您的意见是我们前进的最大动力。我们会从所有参与调查的朋友中挑选出10 名幸运读者, 并送上我们准备的精美礼品。

调查链接: <http://survey.csdn.net/inquiry/viewinquiry.asp?id=40>

读者来信

刘海涛（学生）

确实《程序员增值合订本》是我见到的最好的软件方面的书籍，希望多介绍一些开发实例，以便广大初级读者学习进步。

徐强（学生）

本人是本刊（注：《程序员》杂志）的忠实读者，怀着对知识的渴望和追求订阅贵刊。贵刊内容充实、新颖，希望《程序员》杂志继续求新、求变、求好，使它更好地受益于读者。

陈永祥（在读研究生）

作为程序员最担心的是失去斗志，失去方向。《程序员》不断地激发着我（我不是软件专业，也不是计算机专业）的理想，让我懂得当现实与理想相违背时我该怎么做。《程序员》做到了。李开复先生说的很好，坚持理想，但也许我们还要把自己的事做的足够好！

段俊（软件工程师）

说真心话，《程序员增值合订本》和《程序员》杂志都是非常出色的刊物，对于我们学习编辑，做系统开发都有很多的帮助。我感觉贵刊考虑的很周到，《程序员增值合订本》非常超值。

黄隽（软件工程师）

合订本的内容和结构都十分满意，是一本值得放在案头查阅的必不可少的工具书，光盘内容也不错，继续努力。

王毅（采购经理）

总的来说《程序员》杂志很出色，我从试刊开始看，中间只漏过两期（2002.05与2003.06），合订本也买了三套了，觉得贵刊应该是国内软件技术类专业期刊的NO.1。

杨科（军人）

合订本将技术、工具介绍、规范集合安排，如同手册一般，给我提供了很大的方便，非常感谢。

严志强（数据库管理员）

《程序员增值合订本》和《程序员》杂志总体看上去很不错，文章基本个个精典，封面设计有了很大的提高，不再让人感到压抑。文章语言精炼，不失深度，横贯程序员关心的方方面面，这些都是我想得到的，现在已经得到了，当然，只有在《程序员》。希望你们继续努力，再创辉煌。

刘波（系统架构师）

能够继续紧跟技术潮流的发展方向，知识的介绍也很系统化，有很多叫得响的精品栏目。

张亮（项目经理）

《程序员》杂志是我们公司每年都要订阅的杂志，因为从这本杂志中从事软件开发的人员可以开阔自己的视野，了解业界最新动态，也可以对当前的先进技术有所把握。

人物&报道

名人堂

3D 游戏之神: John Carmack	1
最热衷编程的 CEO: 吉姆·古德奈特	1
俄罗斯杀毒大王 Eugene Kaspersky	2
剑桥情缘	2
BEA 首席软件架构师 Adam Bosworth	3
Ray Ozzie: 网络协作的天才骑士	3
Niklas Zennstrom: P2P 会战中的维京战士	4
Bill Joy: 技术天才+公共知识分子	5
操作系统“天神” David Cutler	6
Kevin Lynch: 穿梭在图形界面与应用程序中的探索者	6
盘点 Rob Pike	7

特别策划

中国程序员, 昂首走过 2003	8
网络游戏, 程序员的新天堂?	18
软件学院露锋芒	21
Windows 源代码泄露全景报道	24
日本软件工程的魔方	29
论托普的“倒掉”	32
中国开源的冰与火	38
中国优秀软件架构师感悟录	44
在中国如何卖企业软件	48
迫在眉睫的职业规划	50

技术访谈

Wiki 发明人 Ward Cunningham 访谈	53
.NET 架构内幕——Eric Gunnerson 专访	57
BitTorrent 之父的故事	59
再访契约式设计之父——故障、预处理和重构	60
来自巴黎的华人软件架构师	61
与 Luke Hohmann 对话软件架构师	63
系统设计原则与程序设计风格——Bjarne Stroustrup 访谈录	65
实用主义程序员访谈录	68
契约式设计之父——Bertrand Meyer 访谈	70
超越版本控制——解析本土化软件配置管理	73
超越灭蟑仪——和 Gerald Weinberg 谈“探索需求”	76
PHP 创始人——Rasmus Lerdorf: 我不喜欢编程	79
James Rumbaugh 面对面	80

软件创业

银弹, 中国造——面向构件, 面向未来的编程?	82
开发管理的 QMR 复方	84
激情开发中国的 IDE 平台——Magic C++ 背后的故事	85
网络游戏创业从足球开始	86
汉化中领悟开源	87
独行五年开发交互图形控件	88

走向海外

做合约程序员	90
美国之行见闻录	91

班加罗尔之行	92
程序百味	
叛逆的创新者和忠诚的追随者	95
IT 核心技术的肉搏战——IT 的水晶球这次又会透露什么样的讯息?	95
Vertigo 软件与选择	98
源代码和魔术	98
成为更好的程序员	99
奶酪	100
价值	100
程序员的生涯规划	100
MSN 昵称与审美疲劳	101
技术的文字	102
学校里没教的事	102
感悟两则	103
技术的风险	103
人类与面向对象语言	104
垃圾场	104
饱蘸生命的程序	105
古代最成功的项目团队	105
修车师傅与软件项目管理者	106
程序员的乌托邦	106
打磨软件 Positive Thinking!	107
软件是门艺术	108
3G 的 Killer Application?	108

管理与实践

软件工程

敏捷开发生态式培训设计及实践	109
拥抱变化敏捷设计从理论到实践	111
J2EE 企业架构的参考模型和实现策略	114
代码生成、产生式编程与 MDA	118
Ivar Jacobson 谈面向对象实例研究	128
迎接 UML 2.0	131

CMM 专栏

解惑 CMM/CMMI	135
人、过程和方法的思辨	138
需求管理的思辨	142

测试专栏

软件测试全解析	147
软件测试实践之测试需求与测试用例	151
软件测试实践之测试计划	154
性能测试用例	157

交互设计

界面困惑	160
什么是交互设计	163

开发之道

为更为实用的网页设计而奋斗	165
火星上的高尔夫球!	166
OO 杂谈	169

从个人软件到企业软件	172
------------------	-----

非程序员

用例：十年风雨	177
业务建模 vs. 系统建模	179
对用例工具的期望	182
打开需求之门——在中国实践用例技术的感悟	185
从基本用例到对象	188
需求问题根源需求启发技术	193

项目管理

项目进度的黑洞——已完成 90%	195
项目管理“九要”与“八不要”	195
管理技术人员的秘密	197
把握项目的脉搏	198
从技术专家到项目经理	202
开放源码——组织与协作	204
对一个驱动程序的重构过程	207
如何使用 google 解决问题	216
基于 IM 的团队协作	218

企业应用

workflow 技术应用案例——税务流程再造项目	222
企业信息系统的集成	224

凌波微步 II

代码地震	227
滚动的一天	231
并发系统的测试	234
打印机疑案	238
天下文章	242
契约式沟通	246

实战 OO

实战 OO：为问题域建模	250
实战 OO：用例建模	252
实战 OO：鲁棒分析	255
实战 OO：交互建模	258
实战 OO：过程总结	260
实战 OO：部署与实施	263
实战 OO：开启征程（结尾篇）	265

程序员手册

新名词解析	269
-------------	-----

加法魔术

Struts+Hibernate 谈 J2EE 的数据表示	271
Linux + Apache + Mysql + PHP 典型配置	271
使用 struts+spring+hibernate 组装 web 应用	273
对于 DIV + CSS 的开发方式，我们也要听听另外的声音	277

流行工具手册

Eclipse：遮得住太阳吗	279
使用 XmlSpy 开发 XML	281
JDeveloper10g：数据库里长出的咖啡豆	285

善用.NET 的 PropertyGrid 控件.....	287
InterSpy——一种集成的视窗消息跟踪与过滤装置	289
Xdoclet: “懒惰者”的掌中利器	291
JSPMaker 完全征服	293
加速实施 XP 的三剑客之持续集成与 AnthillPro	301
测试驱动与 JunitX	303
帮助文件制作工具的选择和使用	306
安装程序制作工具的选择和使用	309
Visual Basic 的九次蜕变	311
UML 工具的选择.....	313
评 Visual Studio 2005 Team System——团队协作开发的利器	314
工具介绍: Thinstall.....	316
每日构建利器: FinalBuilder.....	317
点评性能测试工具 OpenSTA.....	319
NUnit .NET 项目测试点评	321
IT 史记	
从代码编辑软件 SlickEdit 说起.....	323
Web 开发技术史话	325
数据库发展史.....	327
书写历史的甲骨文——ORACLE®公司传奇	331
Linux 创世纪.....	333
网站精粹与热帖索引	
2004 年度 Sourceforge 最佳开源项目.....	336
2004 年度 CSDN 论坛 TOP100	351
2004 年度 CSDN Blog TOP100.....	357
2004 年博客园 TOP50	361
2004 年嵌入式研究网 TOP30	364
2004 年软件中国网 TOP30	365
2004 年 Vchome 网站 TOP15.....	367

2004 年鉴

2004 软件业大事记	368
2004 年百本好书	
.NET 类.....	397
Java 类	397
C/C++类	399
Delphi 类.....	400
基础理论类	400
软件工程与方法学类.....	401
游戏开发	403
技术人文类	404
系列推荐类	405
2004 程序员薪资报告	406

3D 游戏之神：John Carmack

谈到游戏，从引爆电脑游戏核弹的《DOOM》（毁灭战士），到让无数玩家沉溺在死亡竞赛中进行光荣与梦想战斗的《QUAKE》（雷神之锤），再到如今万人联机激情对抗的《半条命》和《反恐精英》，都有着强大的游戏引擎在背后支撑着这些伟大的游戏，而这些引擎的背后，都镌刻着同一个人的名字：John Carmack。

如今，Carmack 的游戏引擎甚至已经成为衡量 PC 图形卡的公认标准，作为一个引擎编写者，他的一举一动能让硬件行业的大佬们胆战心惊。某些显卡生产商在研发新产品之前甚至会先同 Carmack 商量一下，以确保他们的硬件可以完美地支持 ID（Carmack 所在的公司）出品的游戏。

除了 3D 方面的成就外，John Carmack 还为游戏业带来了许多其他技术革新：网络代码中的客户端侦测、多重纹理和便于修改和可扩展的游戏代码，游戏内部的命令行指令……似乎所有该做的 Carmack 都已经做完了。Carmack 的出世无疑是游戏迷的幸事，但又不能不说是其他同期游戏引擎设计师的不幸。

是什么让 Carmack 在短短几年内，便神奇般达到图像编程禅宗般的境界，让那么多富于创新的游戏设计师塑造世界的理想变成现实的呢？他是生来功力深厚，还是背后有高人点拨？

事实上，在忍受了两个学期了无生趣的大学课程后，Carmack 即告退学。以世俗的眼光，一个高中毕业的毛头小伙能有何大作为？然而，就在 2001 年，刚而立之年的 Carmack 即步入美国互动科学艺术协会名人堂，成为第四位获此终生荣誉的游戏界名人，也是迄今为止最年轻的一位。

他是人？亦或是神？

研读完 Carmack 的过去，你会尊敬地发现，其专注精神举世无双。他所有的一切完完全全根植于自学和钻研，十年如一日地努力营造真三维的电脑空间。尽管他现在身价连城，拥有 20 多辆法拉利，但 Carmack 仍醉心于自己的工作，生活方式一如十年前的自己，像代码风格那样：简单、整洁、高效。

为了静心研究，他会如僧侣般与世隔绝，不出旅馆门一步。正如 Carmack 坦言：“我的人生就是以知识的积累和学习的不断深入来分为不同阶段。如果科班出身的图像程序员知道我在写 DOOM 时有多业余，他们肯定会大吃一惊，那时我甚至不能正确地剪切墙体多边形。我也为游戏的完成而自豪，但这途中所克服的各种困难却让我更铭记。”展望未来，Carmack 相信《毁灭战士 III》的图形技术将在未来 5 年内引领游戏引擎的潮流。

“教给你的同行一些东西，不会因此而失去什么。一辆法拉利小车只是物质享受，做人最重要的是真诚。”Carmack 不仅痛恨一切软件专利制度，而且还身体力行地去铸就自由文化的辉煌。其免费发布源代码使他获得玩家和社会的广泛尊敬。早在 1993 年 DOOM 游戏中，Carmack 就公布了关卡编辑器和其他辅助工具的源代码。Carmack 对游戏开发进度和想法永远是公开的，每天的工作思路和开发进度都写在一个后缀名为 *.plan 的文件里，任何人通过网络都可浏览。

Carmack 已经不只满足于游戏业。他说：“游戏设计中的很多问题对我来说并不是什么真正的挑战，因为我经历的已经太多了。”已经“快达到图像编程顶点”的 Carmack，如今又重新拾起了儿时对火箭的爱好，他们的目标是一千万美元的“X 大奖”——把三名乘客送到外太空。对于专注的 Carmack 而言，这又有什么不可能呢？

“在信息时代，客观障碍已不复存在，”厌恶那种貌似先哲圣贤般地夸夸其谈的 Carmack 说：“所谓障碍都是主观上的。如果你想研发什么新的技术，你不需要几百万美元的资金，你只需要在冰箱里放满比萨和可乐，再有一台便宜的计算机，和与之献身的决心，你即可拥有任何你想拥有的编程深度！”

（Carmack 的传奇也正是传统“美国梦”的样板：他们的经历使其成为叛逆的偶像，他们用自己的热情点燃了一种新的艺术形式、一种文化现象。我们将有幸在电子工业出版社即将出版的《MASTERS OF DOOM》一书中中文版亲历他们所经历的激情岁月。）■

最热衷编程的 CEO：吉姆·古德奈特

奋战在一线编码的程序员，时常会觉得自己和公司高层间缺乏联系。管理层并不清楚如何关心程序员，除非他本身也是个热爱编程的程序员——比如 SAS 软件公司的 CEO 吉姆·古德奈特。

SAS 现已发展成为全球最大的私有软件公司（非上市公司）。《财富》前 100 名公司中有 98 个都使用它的软件。吉姆·古德奈特曾被《商业领袖》杂志评为十年一度的商业领袖。

古德奈特诞生于北卡罗来纳州的威尔明顿，在北卡罗来纳州立大学读二年级的时候，他选修了当时校内刚有的计算机课程。第一节课上他就被教室里那台 IBM 1620 计算机深深迷住了。“我需要更深入的了解它，”古德奈特回忆到：“开始一个月，我完全不得要领，然后我似乎突然就开窍了，一下子明白了一切！”那个学期结束后，他已经可以担任助教了。那个夏天，古德奈特找到了一份兼职：给学校的农业经济学系写程序。在他随后的大学生活中也一直在坚持写程序。

在攻读统计学博士学位期间，他和吉姆·巴尔一起编写了后来被称为 SAS（Statistical Analysis System 统计分析系统）的软件，古德奈特负责设计了软件的运行流程。SAS 软件最初的目的是帮助学校预测谷物收成。1968 年，他们将软件送给第一批顾客。1972 年，古德奈特看到高校缩减了对校内计算中心的投资，便去向 SAS 的用户们求助，恳请每人捐助 5000 美元以解决资金问题，并力图扩大 SAS 软件的目标市场。1976 年，在佛罗里达召开的第一次用户会议上，他们被到会的人数和用户显示的热情“吓了一跳”。古德奈特意识到 SAS 软件大有前途，于是这一年与合作者一起创建了 SAS 研究所。

SAS 是用于决策支持的大型集成信息系统，该软件系统最早的功能仅限于统计分析，至今，统计分析功能也仍是它的重要组成部分和核心功能。经过多年发展，SAS 已被全世界 120 多个国家和地区近三万家机构所采用，直接用户则超过三百万人，遍及金融、医药卫生、生产、运

输、通讯、政府和教育科研等领域。在英美等国,熟练使用 SAS 进行统计分析是许多公司和科研机构选材的条件之一。在数据处理和统计分析领域,SAS 系统被誉为国际上的标准软件系统,并在 1996~1997 年度被评选为建立数据库的首选产品。堪称统计软件界的巨无霸。

从校园搬出后,SAS 一开始就在赢利。最初的 4 名雇员就已经享有了每天的新鲜水果、M&M's 巧克力、丰盛的早餐、优厚的分红制度和宽松灵活的工作日程。

在公司具备了 5500 多名雇员和 115 间办公室的规模下,古德奈特仍然花费差不多一半的时间在领导开发团队和编程上。事实上,他还在负责维护他最初编写的流程中的若干部分。

古德奈特只花很少的时间在各种内部会议上。他说:“我天生不喜欢

开会。”而是更喜欢在办公室走来走去,与员工直接交谈。有时,为了自己感兴趣的项目,他甚至把办公室直接搬到工作小组那儿去。

“我热爱编程,”古德奈特说:“这好像整天在解一个谜,很有意思。为此,我组建了一个管理团队替我承担一部分管理责任。”

古德奈特最清楚如何创造一个能充分发挥技术人员想象力的环境。能有这样一位痴迷于技术,了解并深深关切程序员利益的领袖,SAS 公司的员工们非常幸福,“一般情况下,工作 5 年之后,你就想跳槽了。”已经在 SAS 工作了 16 年的研发部副总裁基思·柯林斯说,“但当我在这儿工作 5 年后,我发现我很喜欢这里。在这里,你肯定会做出成绩。”■

俄罗斯杀毒大王 Eugene Kaspersky

现在的病毒越来越厉害,给社会造成的损失也越来越大。但与此同时,这也给防病毒软件企业提供了发展的持续动力。除了赛门铁克、趋势科技等国际性厂商外,还有一个著名的厂商不能不提,这就是俄罗斯的防毒公司 Kaspersky Lab。这家公司的创始人和核心技术研究负责人就是 Eugene Kaspersky,他也是俄罗斯软件业的代表性人物。

1965 年,Eugene Kaspersky 出生在俄罗斯的诺沃罗斯基,他在大学时学习了密码学、通信和计算机科学。在 1989 年,他第一次接触到“Cascade”病毒,从那时起,他开始了研究计算机病毒的工作。从 1991 年到 1997 年,他在俄罗斯大型计算机公司“KAMI”的信息技术中心,带领一批助手研发出了 AVP 反病毒程序。

AVP 的反病毒引擎和病毒库,一直以其严谨的结构、彻底的查杀能力为业界称道。但是,由于整个杀毒引擎和库与主程序是完全分离的,软件的结构化非常好,所以里面的静态模块可以被拆出来复用。这样,数千个 obj 链接文件也可以被其他人在编译程序时编译进来,所以 AVP 核心模块和病毒库便被很多公司所“引用”,包括非授权的引用,这也一度成为很多公司快速进入杀毒软件领域的“快捷方式”。AVP 虽然是技术的巅峰之作,但由于长时间懒于开发市场,很长时间里用户并不多,导致部分开发人员流失。俄罗斯的另一家反病毒产品 Dr.Web,与 AVP 有很深的渊源,而 Symantec 产品的主力开发人员里面,也可见 Eugene 旧部的身影,还有人跳槽去了 McAfee,但这始终也改变不了 AVP 是全球顶

级反病毒引擎的事实。

1994 年,在部门负责人也是 Eugene Kaspersky 的妻子 Natalya Kaspersky 的带领下,公司逐步形成了职业化的管理层和开发小组,并开始发展俄罗斯及海外的销售网络。他们同时与芬兰的 F-Secure,德国的 G-Data,日本的 Vintage Solutions 等反病毒厂商合作签订了在这些产品中使用 Kaspersky AV 核心模块的协议。

1997 年 6 月,私人控股的 Kaspersky Lab 公司正式成立,这家公司也可以说是 Kaspersky 的夫妻店。Eugene Kaspersky 是首席防毒技术研究员,主要负责反病毒技术的研究工作,他的妻子则担任 CEO。2000 年 11 月,基于各方面的原因,AVP 更名为 Kaspersky Anti-Virus。

Eugene Kaspersky 身上更多的是技术人员的特质,他不喜欢抛头露面,也不喜欢接受任何的采访,不过当谈到反病毒技术时,他总是能指出关键所在。Eugene Kaspersky 说:“俄罗斯杀毒软件之所以非常强大,是因为俄罗斯有国际顶尖的数理科技领域的特殊人才及优良的传统。”

现在,Eugene Kaspersky 开始对 Linux 上的防毒软件技术进行研究,并推出了这方面的防毒软件。Eugene Kaspersky 还是计算机反病毒研究员协会(CARO,该协会的成员都是国际顶级的反病毒专家)的成员。■

剑桥情缘

2003 年 2 月,剑桥失去了一位极为杰出的科学家 Roger Needham。在与病魔进行了长时间斗争之后,Roger 教授在一个星期五的夜晚与世长辞。作为剑桥大学著名教授和计算机科学家,Roger Needham 一生取得了无数的荣誉与成就,对现代计算机科学规则体系的形成与发展做出了卓越的贡献,而其晚年更是在剑桥大学建立了微软研究院,为这位伟大科学家与剑桥大学几十年的情缘画下一个完美的句号。

Roger 生于 1935 年,早在 20 世纪 50 年代,还在 Doncaster 语言学校读中学的他便获得剑桥奖学金进入剑桥修读数学专业,并于 1956 年获得数学与哲学双学位,接着又于 1957 年拿到了数字分析与自动计算专业学

位证书。此后在 1958 年 Roger 与和他一起修读博士的 Karen Sparck Jones 结婚,并在两年里完成了博士的研读与项目开发,于 1961 年进入剑桥数学实验室,也就是后来人们熟知的剑桥计算机实验室(CLC,Computer Laboratory of Cambridge)。

在加入剑桥数学实验室之后的 20 年里,Roger 参与了操作系统、时间共享系统、内存保护、局域网构架、分布式系统以及卫星制导等多方面前沿技术的研究,从 1977 至 1997 的每一年的 Roger 都会有 5 到 6 个星期在硅谷生活。Roger 在 PhD 期间参与开发了第一个核心系统 TITAN,此系统在世界上最早采用了缓存技术并且是第一个在美国之外

开发出来的多重访问操作系统。此后 Roger 还引入了函数进行文件口令加密，这一发明已经被所有现代计算机系统所采用，此外，TITAN 同时引入了全盘备份以及差异备份技术。

到 1980 年，Roger 正式成为剑桥计涯留下最为厚重的一笔，他成功地把微软研究院建立为如今世界一流的计算机研究机构，并把全世界无数杰出的人才吸引到剑桥。

从 Roger 教授进入还处于萌芽阶段的计算机行业到晚年建立起微软研究院并把它发展壮大，四十多年里所有有幸与他共事过的人们都对他的事迹敬佩不已。剑桥大学大臣 Sir Alec Broers 教授在谈起 Roger 时说道：“这些年来，Roger 为剑桥以至整个行业所作的贡献是无法计量的。从 50 年代在剑桥修读本科到 60 年代在数学实验室进行早期积累，从 20 世纪 70 年代到 80 年代所做的一系列突破性研究，从 1980 到 1995 负责计算机实验室工作并在 1996 到 1998 年间出任副使，Roger 向世人展示了一

种对学术执着的责任感，正是这种责任感让人们一直对他心存感激。他的过世对整个剑桥是一个巨大的损失，我们对他的妻子 Karen 献上最诚挚的慰候。”

作为微软研究机构高级副主席，Rick Rashid 带给了我们这样一段回忆：“当我一次听 Roger 的讲座，我便被他特别的演讲方式所深深吸引，也就是媒体后来所谓的‘精致而又富涵思想’。出席计算机科学讲座常常是一个痛苦的历程，你往往需要不断把从主讲人嘴里蹦出来的一个又一个生词连贯起来，进而理解其含义。但听 Roger 的报告则完全不同，我总是对他所说的每一个字词都感到意犹未尽，而又热切盼望着接下来的内容，回味起 1983 年那个夜晚总是如此美妙。”

2003 年 2 月 17 日，一本纪念 Roger Needham 的专著面世了，此专著由数十位世界顶级计算机科学家合力撰写而成，为 Roger 一生取得的成就以及他的不幸病逝寄上景仰与哀思。■

BEA 首席软件架构师 Adam Bosworth

“Adam 所设计和开发的世界级软件，比我能想到的其他任何人都要多。”对于 Adam Bosworth，同事 Joel Spolsky 一点都不吝惜自己的溢美之词。他俩曾共事于微软，当 Joel 正在设计所谓的应用程序可编程能力策略时（就是后来演变成 VBA 的玩意儿），Adam 正在设计 Microsoft Access。

对，正是 Access！虽然它只是一个 PC 版的数据库产品，在 DB2 那样的庞大物面前根本看不见它的影子——但是它的简洁和友好性，让数据库不再是专属于程序员的迷宫。冒昧揣测，Access 可能是世界上使用者最多的数据库！

在 Adam 的战绩表上，有 Borland 以前的 Reflex 关系数据库；还有 ODBC，这个让人既恨却又无法忘却的家伙；当然还有 IE 4.0 HTML 引擎。我至今仍能回忆起，用 Windows NT SP6 将 IE 升级到 4.0 后那一刻的欢跃；而在这个版本的 IE 中，我们应该记住这两个深刻的名字：Trident、DHTML。

Adam 一开始在 Borland 工作，直到 1989 年加入微软。他在那里呆了十一年，历任不同的高层管理职位，涉及产品有 Access、ODBC、IE 的 Trident 编辑器和文档对象模型、SOAP 和 XML。

对他最好的描述仍来自于曾经的同事 Joel，“他的思维快得令人难以置信，而且不断地被证实是正确的。与大多数这样的人不同的是，他还善于聆听。而且一旦他发现自己是对的（正如 1992 年那一个小时，当我令他确信宏语言是需要面向对象时），他会改变自己旧有的想法，而不会感情用事。作为程序经理，没有人能树立像 Adam 那么好的榜样”。

Adam 被公认为 XML 的先锋之一。他在微软时，曾任 WebData 部门的总经理，这个部门即致力于规划和促进微软的 XML 战略。后来，他和

Tod Nielson 以及另外 30 名微软前雇员从微软脱离出来，创建了 Crossgain 公司。该公司致力于 Web 服务，Tod Nielson 任 CEO，而 Adam 任 CTO。很明显，这已经威胁到微软在 XML 上的地位了。

为此，微软通过法律手段迫使他们遵守不竞争条款。这给了 Adam 一个喘息的机会，用两个月去沉思 Web 和端服务的未来。在此期间，Steve Gillmor 也和 Adam 讨论了 XML、SOAP、WSDL 以及 Crossgain 下一步的打算。

2001 年 7 月，Crossgain 为 BEA 所收购，Adam 也成为 BEA 的开发资深副总裁，负责为 BEA 的系统架构分部推进 engineering 效果。可以断定，BEA 与 IBM、微软联手推出 Web 服务，其中 Adam 的地位不可或缺。他自己也曾提到，在担任 BEA 的首席架构师和高级开发副总裁时，50% 的时间是花在企业级用户身上的。在经历了 Access、XML 后，Adam 正在面对第三个重大的挑战——Web 服务。

当接受 ComputerWorld 采访时，他谈到了技术产业的趋势，正如 Web 服务的演进。同时，他也指出了 Web 服务的两个不足之处：可靠的消息传递和异步性。对于前者，他谈到：“利用消息总线技术，我们逐渐做到了这一点，BEA 提供了一套可靠消息传递的可选方案。但是转到 B2B 领域时，你需要在 Web 服务这一层有相应的标准。因此，也是最近，我们和 IBM、微软一起推出了一个新的标准，用于实现 Web 服务中可靠的消息传递。”

Adam 对 Web 服务的自豪几乎是无法掩饰的，“PC 打开了第一扇门，HTML/HTTP 打开了第二扇门。毫无疑问，Web 服务将打开下一扇门”。让我们拭目以待，这位先导者在荒芜之地上，用他的勇气和毅力，开创下一个新的纪元。■

Ray Ozzie：网络协作的天才骑士

如果现在有人问“Groove 是什么”，也许不是所有人都能答得上来。但如果问题换做是“Lotus Notes 是什么？”，毫无疑问这位提问者肯定会被认为是外行。可是，有谁知道 Groove 和 Notes 的发明者是同一个人：

Ray Ozzie，一位专注于网络协作的天才。

七十年代，在伊利诺斯大学计算机系读书的 Ray Ozzie 第一次接触到 Plato，一种原始、功能非常有限的校园网络协作系统。这惊鸿的一瞥，

改变了 Ozzie 的一生。从那时起他就在不断地思考，如何“让计算机为人类互动服务”。

毕业后，Ozzie 加入了浩浩荡荡的程序员大军中。幸运的是，他能不断地与当时技术圈子里的巨头们共事：Johnathan Sachs 是推出 Lotus 1-2-3 的传奇人物；Dan Bricklin 是当时计算机界的牛人之一，在表格软件上曾取得创造性的突破；当然还有后来鼎鼎大名的 Bill Gates。Bricklin 把 Ozzie 介绍给当时还在 Comex 的 Bill Gates。他回忆说那时的 Ozzie 已经算得上是一个明星了。Gates 也曾说过：Ozzie 是地球上最好的五个程序员之一。

八十年代初，Sachs 力劝 Ozzie 加入 Lotus，两人合作相继开发了 Locus 1-2-3。此后，Lotus 全力支持 Ozzie 成立新公司 Iris 开发 Notes，这个面向网络的协作软件已在 Ozzie 头脑里酝酿多年。五年后，在计算机网络和 PC 图形界面都还不普遍的时代，Locus Notes 横空出世。它集通讯软件、电子表格软件和图形软件于一身，是第一套真正严格意义上的商务软件。它基于客户机/服务器结构实现，使得商务人员可以通过网络进行协作。Notes 的出现建立了 Lotus 群件技术领导者的地位。在中国，Notes 更是横扫千军，几乎成为所有的 OA 系统二次开发的平台。微软的 Exchange Server + Outlook 组合，在它的阴影下一直郁郁不得志。难怪达拉斯小牛队的老板 Mark Cuban 会说：“它是唯一让 Bill Gates 嫉妒的产品，它让微软的通信软件靠边站了至少 10 年”。这也是为什么 1995 年 IBM 愿意斥巨资 33 亿美元收购 Lotus，不就是为了得到 Ozzie 和他的 Notes 吗？

接下来的两年，Ray Ozzie 呆在新东家 IBM 里没什么大动作，不过新的想法在他头脑中源源不断地涌现。1997 年，Ozzie 离开 Lotus，与四个 Notes 时期的老战友共同成立了 Groove Networks，继续探寻网络协作的新方向。

如果说 Notes 是 C/S 潮流中网络协作的里程碑，那么 Groove 则是在

P2P 背景下的先锋。Groove 的灵感来自于 Ozzie 的偶然发现：他的孩子在网上聊天和玩游戏。这些遍布世界各地的玩家们建立他们自己的小组，制定规则，边玩边聊。这让 Ozzie 很震惊，“他们解决问题的方式比你做生意时用的更直接有效”，从中他看到了在线互动与应用共享的巨大潜力。想想七年前，大家才刚开始接触基于 C/S 的互联网服务。所以 Bricklin 评价说“Ray 在浪潮刚出现时就抓住了它，而我们现在才看见。”

Ray Ozzie 是一个低调而执著的人。在开发 Notes 的五年里他默默耕耘，快乐而富有成就。在做 Groove 时，他也如此。从 1997 年到 2000，在无数次的思考和讨论中，在用过无数张防水纸记录洗澡时解决的技术难题后，2000 年 10 月，Groove 1.0 版终于问世了。崭新的 Groove 带给人的不只是强大的跨群组网络协作功能，还有可扩展的开放式开发平台，以及对安全性的周密考虑。它从后台直接对协作过程加密，不会让用户感觉到加密的繁琐，却能于在线互动时保护他们的隐私。自由、交流、共享、安全；“让计算机为人类互动服务”，Ozzie 又前进了一步。

在几番融资后，Intel 和微软现在成为了 Groove 的主要投资者，这两个互联网时代的霸主对 P2P 领域也虎视眈眈。也许有人会问，难道 Groove 也会重蹈 Corel 的覆辙？那么多年对网络协作痴心不改、默默耕耘、赢得计算机界普遍尊敬和赞誉的 Ray Ozzie 会愿意吗？Ozzie 没有在他的 Blog 中说明。然而，早先《计算机经销商新闻》(CRN)采访他时，问到协作技术是否会被微软 Office 或是 Windows 操作系统吞没时，Ozzie 很是乐观。他说：“时间会证明一切。在这一刻，你可以说交互技术是操作系统的一部分；下一刻，你又可能说它是包含在生产率套件 (Productivity Suite) 之中的。……但我坚信必将出现一个中间的层面，很清楚这就是协作层。Groove 将在这一领域大显身手。”

对，时间会证明一切。我们也坚信，自由、协作的互联网精神绝不会轻易地退出历史的舞台。■

Niklas Zennstrom: P2P 会战中的维京战士



舒适 / 译

Niklas Zennstrom, 36 岁，瑞典人。乍一看，他是一个典型的斯堪的纳维亚人，腼腆沉静，精于思考；可就是这样一个人，在一系列的 P2P 的会战中生猛得像一个维京战士。

Kazaa 战役

Kazaa 跟 Napster 一样，是一种基于 P2P 技术的音乐文件共享软件。2000 年秋季，Niklas Zennstrom 和 Janus Friis 在荷兰阿姆斯特丹的一个小办公室里进行了 Kazaa 的开发工作，并在网上试运行。起初，他们预料可能只会有几千人对他们的网站感兴趣，但随着 Napster 在 2001 年 2 月受到重创之后，大量的 Napster 用户纷纷涌向 Kazaa 网站。据 CNET 数据显示，迄今为止全球已有 2 亿人次下载过 Kazaa。

Kazaa 一路高歌猛进，无数网友利用它寻找、交换、下载各类免费的 MP3；而另一面，全球唱片业的正版 CD 销量持续下滑，并且首次出现了可刻录 CD 盘销量超过已刻录盘销量的情况。搞倒了 Napster 的唱片公司又坐不住了，调转矛头指向 Kazaa 等三家网站，并把 Niklas Zennstrom 和 Janus Friis 二人附加为个人被告。

对此，Zennstrom 不屑一顾。他坚信，比起现在网络上流行的集中式服务器和流媒体来说，P2P 在宽带的助威下是当今最廉价、最快捷的音

乐销售技术。这也比传统制造 CD 并通过店铺售卖的方式便宜了很多。唱片公司不应当那么短视，拒绝接受这项新技术带来的好处和潜在的商机。

幸运的是，在这次唱片工业与 P2P 技术的大战中，荷兰最高法院作出了对 Kazaa 有利的裁决：Kazaa 不需为该 P2P 网络上的侵权行为负责，并拒绝绝对 Kazaa 加以控制。Niklas Zennstrom 与 Janus Friis 兴奋日表示“这场胜利奠定了文件共享软件/网站在欧盟，甚至全球的合法地位。”

Skype 战役

如果说唱片公司短视，看不到技术革命带给传统商业的巨大好处的话，那么技术人员的短视，可能就是看不到新技术的潜在商机：在各类传统商业中应用、扩展。Zennstrom，有人认为他内心深处与 Bill Gates 有着惊人的相似，他清楚的知道文件共享只是 P2P 技术的一种应用，P2P 能够解决好多问题。所以，在 Kazaa 之后，Zennstrom 与 Janus Friis 再度联手成立了 SkyperP2P 电话公司，为用户提供高质量的互联网电话服务。

其实，这项技术并不新鲜，1996 年时 ICQ 就能为在线用户提供及时语音通信。随后，MSN Messenger、Net2Phone 等类似软件层出不穷。Skype 在这些软件的基础上优化了某些性能，如加强用户通话的保密性，通话

无延迟，通话质量甚至比固话还好，不进行任何配置就能兼容所有防火墙、路由、NAT（网络地址转换）……最重要的是，Zennstrom 表示这一高质量的 Skype 语音服务将永远免费。

当然，天下没有白吃的午餐。免费的东西总是有限制：只有电脑对电脑的基础语音服务才免费，如果想拥有电脑对固话、移动电脑对固话或移动电话等服务，用户就得付费。只不过，相对于传统的电话收费，Skype 的收费低廉多了。难怪 Skype 推出至今已经拥有 550 万用户，遍及 170 个国家和地区，成为世界上最大的电脑对电脑语音通信网络。连美国 FCC 主席 Michael K. Powell 是用户之一。

可以说，Zennstrom 在利用 P2P 技术获得商业成功的同时，也实践了部分互联网的精神：技术革命为人类提供更质优价廉的服务。

Joltid 战役

Zennstrom 还尝试用另一种途径将 P2P 技术商业化。他在 Kazza 之后

还成立了 Joltid 公司，创建了世界上最大的安全对等内容分发网络。Joltid 的软件之一 PeerEnabler，能让用户将自己的计算机变成类似以前软件公司在各地的镜像服务器，因此，如果软件公司使用 PeerEnabler 网络来发布他们的更新版本的话，用户就不用从开发者那边的服务器下载，而是去那些最近的已经下载过该文档的用户那里获得。按照 Joltid 的说法，这可以减少内容发布者用于发布的费用，包括在带宽和服务器上的花费。另一方面，这也提高了网络传送的效率，避免网络堵塞。而 Joltid 可以通过授权方式从内容发布者那里收取费用。

这种想法还不错，但是不是 P2P 商业化的一种途径呢？还要拭目以待。

虽然商业利益无处不在，虽然这些永远是自由、互助之网络精神的阻碍，但是总是有一些人在不断尝试，发明新技术，探寻各方意志的平衡点，所以，无论如何他们都应当获得喝彩。■

Bill Joy：技术天才+公共知识分子



方茜 / 文

其貌不扬的瘦高个，凌乱的亚麻色头发，这就是被《财富》杂志誉为“网络时代的爱迪生”的 Bill Joy。他曾是 Sun 的创始人之一，并在 Sun 担任首席科学家长达二十一年。他是一位令人崇敬的软件天才，同时也是一个标准的技术狂人——在记载英雄的历史中，我们注意到，天赋和近乎疯狂的偏执这两者总是结伴出现，并在英雄的身上达到一种交融的境界。

然而，软件英雄史诗令人遗憾地忽略了 Bill Joy 的另一面——作为一名以追求公民社会正义和道德为己任的公共知识分子。他也曾多次尝试，希望通过自己的独立思考，并以个人言论的方式来影响社会，推动社会进步和解决公共问题。

话说当年，AT&T 在反垄断法的判决下，不得已放弃了从 Unix 开发和支持中获利的念头，从而专心致志地开始卖 Unix 的源代码许可证。当时，Bill Joy 所在的 Berkeley 计算机科学系就在贝尔实验室所发布源代码的基础上，开始了他们的 Unix 研究——可能当时所有人都没有意识到，这一行为对后来的世界产生了多大的影响。在 Berkeley 持续不断的研究中，Unix 具备了如此多的现代操作系统特征：基于分页的虚存系统、统一的文件系统、强大而完善的脚本、网络……

最早基于分页的虚存系统就是由 Bill Joy 加入到 Unix 内核中的，这使得 Unix 得以打败 DEC 的 VMS 操作系统而成功获得 DARPA 的支持。国防部雄厚的资金成为 Bill Joy 和他的伙伴们强有力的支柱，让他们能够源源不断地挥发灵感，继而完成了 csh、vi、TCP/IP 等等。csh（C Shell）以其强大的功能获得了广大程序包括各种不同版本的 Unix 和 Linux；至于 TCP/IP 就更不用说了，您能在遥远的中国了解到 Bill Joy 的英雄历程，正是基于 TCP/IP 的互联网所赐。1984 年，Bill 又发布了 NFS 网络文件系统，其后则是在此基础上的 PC-NFS。

在成立 Sun 微系统公司后，Bill Joy 又担当设计了 Sparc 微处理器最关键的一部分电路。每年 Sun 公司靠 Sparc 服务器和工作站的生意能赚到上百亿美元。而 Java 虽然是由 Sun 的 James Gosling 所写，却也是因为他的全力支持，从而走到台前，从一种编程语言演变成今天流行的开发平台。

在 Java 之后，Bill Joy 还主持了 Jini——一种连接分布式计算机系统的技术的开发。任何联网的小装置（数码相机、电视机、打印机等）都可以由包含有 Java 写成的简单程序实现自己的功能，并且供其他设备使用；还有 Jxta，这是一套开放的 P2P 协议，允许任何互联网上相连的设备（如手机与 PDA，个人电脑与服务器）交流和协作。

在短短的二十年内，Bill Joy 创造出了那么多令人心动的软件，不得不令人折服。可提到哪一个对程序员的影响最大呢？人们众说纷纭。也许小小的 Vi 编辑器的影响是最持久的。正如 Reg 网络杂志做的调查，大多数程序员都评论说：“没有 NFS、Java 和其他的技术还能活；但是没有 Vi，简直没法活了！”

Bill Joy 能在 IT 圈外广为人知，不仅因为他是个技术天才，也不仅因为他是个人人羡慕的亿万富翁，主要还是因为 2000 年他在《连线》杂志上发表的一篇文章《未来还需要我们吗》。他宣称生物技术和纳米技术的轻率进展也许会给人类带来灭顶之灾，也许未来不再需要人类。

Joy 的这篇跨越其专业的文章引来了很大争议。有人认为，Joy 只是个程序员，并没有资格谈论生物技术的复杂性。就算是 IT 界的同仁尼葛洛庞帝在接受采访时也说：“Bill Joy 当年写这篇文章时，正处在中年危机之中。我正好知道这一点。因此其文反映了他那段时间的失意。”

难道只有所谓的专业人士才有资格评说技术灾难吗？难道公众就缺乏基本的分析与判断吗？所以，Bill Joy 面对这些非难并不放在心上，他认为提出这个命题的目的不是危言耸听，而是要让公众都关心技术危害的问题，思考我们应该做些什么来避免不想见到的未来。这就是一个知识分子的公共良知。

此外，他还陆续发表了《为数字革命而设计》等多篇涉及技术的文化影响的文章，引起了广泛的注目。

2003 年 9 月，Bill Joy 离开 Sun。当时 Sun 公司的股票也应声下跌了 3.2%。后来，Bill Joy 在接受《连线》杂志的采访时高兴地说“嗨，我辞职了！”。可是这并不意味着他的职业生涯就此结束。除了每天在家中面壁沉思外，他还在考虑着未来的技术。说不定哪一天这个网络的爱迪生又会带给我们新的惊喜。■

操作系统“天神” David Cutler

 方茜 / 文

David Cutler, VMS 和 Windows NT 的首席设计师, 1988 年去微软前号称硅谷最牛的内核开发人员, 在操作系统领域摸爬滚打几十年, 其间的经历就像一部标准的外省青年奋斗记。

与许多计算机界的前辈牛人们一样, David Cutler 并不是计算机科班出生, 他在大学拿的是数学学士, 主攻物理, 满怀热情地想成为一位建造事物的工程师。所以, 毕业后他进入杜邦公司从事材料测试。一次偶然的机会, David 被指派负责在 DEC 的计算机上运行模拟程序, 还为多台单机实时系统编写中央控制程序, 调度各种任务、监控系统运作。这个经历不仅丰富了 David 的软件知识, 还让他做出了一个重大的决定: 去一家真正从事计算机业务的公司, 开发操作系统。

1971 年, David Cutler 离开杜邦公司来到 DEC。他的第一项任务就是为 DEC 的 PDP-11 微处理器开发操作系统——RSX-11M。PDP-11 是为工业控制和制造控制而设计的 16 位微处理器。David 结合总体概念和设计原则, 利用汇编语言在非常有限的内存空间内实现了多项系统功能, 如: 树型文件系统、交换应用程序、实时调度和一整套开发工具等。据 David 回忆, 当时连他的橡皮图章上都刻着开发这个操作系统的目标——“容量就是一切!” 后来, 这些概念和原则也体现在了 NT 上。

70 年代后期, DEC 公司在 PDP-11 的基础上开发出 32 位的 VAX 处理器。与之相应, 也要开发基于 VAX 的操作系统 VMS, 要能兼容 RSX-11M, 可以在不同大小的机器上运行。David Cutler 成为这个项目的负责人, 设计 VMS 的架构。1977 年, VMS 1.0 问世。David 唯一的遗憾是, 为了迎合商业进度, VMS 也是用汇编语言写的, 尽管当时完全可以用高级语言。所以, 技术上正确的事并不见得是商业上的最佳选择。随后, David 继续研制 VMS 的后续版本, 不过他有些不耐烦了。1981 年, David 威胁要离开 DEC。为了挽留它的明星开发者, DEC 给了 David 大约 200 位软硬件工程师。David 把他的小组搬到西雅图, 并建立了一个开发中心。这个精英小组的目标是设计一个新的 CPU 体系结构和操作系统, 可以把 DEC 带到九十年代。DEC 把这个小组的硬件项目称为 Prism, 操作系统为 Mica。

很不幸, Prism 项目于 1988 年被 DEC 撤销, 很多项目成员也被解雇。

因此 David Cutler 萌生了去意。此时, 为了未来能够与 Unix 抗争、开发新的操作系统, Bill Gates 见缝插针, 竭力劝说 David 加入微软。David 去了, 还带去了许多与他一同开发 VMS 和 Mica 的程序员。进入微软, David 领导一个工程小组, 负责设计一种能提供文件服务、打印服务和应用服务的对称多处理。操作系统, 起名为 Windows New Technology(NT)。这就是 Bill Gates 想用来对抗 Unix 的新型武器。

经过近 4 年的开发工作, 在 1993 年 6 月发布的第一版 Windows NT 3.1, 已经具备了现代操作系统的雏形——抢先式多任务、虚拟内存、对称多处理器、图形界面、C2 安全级、坚固而稳定的内核、内置网络支持、完全的 32 位代码等。而 1994 年推出的 Windows NT 3.51 和 1996 年推出的 Windows NT 4.0, 在性能上有了更进一步提高; NT4.0 甚至提供了当时最先进的 Windows 95 风格界面。David Cutler 在自己的天梯上继续攀升, Bill Gates 也在销售数字面前笑得合不拢嘴。

毫无疑问, NT 操作系统有一个优秀的内核, David Cutler 成功地引入了硬件抽象层、内核对象这些天才的思想。虽然我们没能得见它的源代码, 但在钻研 NT DDK 的过程中、在埋头可能就是由 David 亲笔撰写的文档中时, 总能有那些闪光点, 让我们可以在不同的时空与大师对话。

如今, Built On NT Technology 的 Windows 2000 和 XP 的成绩有目共睹, 而针对 64 位处理器的 XP 也即将推出。回顾开发操作系统的历程, David 情不自禁地感叹道: “我也不知道, 自己竟是那么的幸运, 能够在有生之年开发好几个操作系统, 而对于任何一个人来说, 哪怕只开发一个都是非常难得的机会。”

是的, David Cutler 做到了, 在操作系统领域中纵横了几十年, 缔造了许多传奇和神话。然而, 又有谁会去看他几十年的专注、寂寞、付出与艰辛呢? 可能每个人在开始自己的职业生涯时都会设定一个目标。然而只有那么一些人会抓住目标紧紧不放、全心投入, 最后这些人成了我们眼中的成功者、技术天才。也许, 这就是成就天才与普通人的不同之处。■

Kevin Lynch, 穿梭在图形界面 与应用程序中的探索者

 方茜 / 文

提到 Kevin Lynch, 你最先想到的是什么? Bingo, 是 Dreamweaver! 任何一个网页制作者, 不论是菜鸟还是开发高手, 都离不开它。其实, Dreamweaver 只是 Kevin Lynch 刚加入 Macromedia 公司时的力作, 在此之前和之后, 他还有许多软件问世。回顾二十多年的开发历程, Kevin 说他做的都围绕着一个主题: 协助人类交流和扩展用户界面设计的界限。

1984 年夏季, Kevin 刚高中毕业, 进入伊犁诺大学学习计算机图形学。那时候第一部 Macintosh (苹果机) 正闪亮登场。Kevin 路过橱窗, 看到电视里的 Macintosh, 感到非常震惊, 鼠标、彩色屏幕、形象的功能图标, 这一切都前所未有的, “就像来自未来的机器”。从那时起 Kevin 就开始为苹果公司开发应用软件。他开发的第一款商用软件是名为

“Legacy”的图形冒险游戏。他和同伴们只凭借复印来的 Macintosh 内核 (Inside Macintosh) 文档就完成了一项极富挑战性的开发，而当时应用软件开发才开始兴起，Kevin 也只是个刚获得公民权的毛孩子。接着，Kevin 又陆续开发出 3D 绘图软件以及桌上型出版应用程序，这些都引入了现在常用的用户界面元素。这期间 Kevin 认识了 Marc Canter，即后来 Macromedia 的创始人。

Kevin Lynch 大学毕业后搬到加州，进入由 Steve Kirsch 创办的 Frame 技术公司，设计用户界面，并开发出第一代 Macintosh 版的 FrameMaker。FrameMaker 是一种排版软件，但是由于没有所见即所得的预览效果，需要像 WPS 一样使用代码来控制段落和格式，渐渐遭到了冷落，最后被 Adobe 公司收购。不过，在这段工作经历中，Kevin 进一步形成了用户界面的思想。

1990 年，三名苹果公司工程师成立了 General Magic 公司，主要研发便携式通信终端的操作系统“Magic Cap”。最初产品在界面设计上并不令人满意，直到 1992 年借助 Kevin Lynch 的灵感才最终定型。尽管去年 Magic 公司不幸破产，Kevin 还是非常怀念那时的工作，整个公司不只是一个团队，更像一个家。

1996 年，在 Marc Canter 的召唤下 Kevin Lynch 加入到了 Macromedia，

定义并主导 Dreamweaver 初步开发。Dreamweaver1.0 一推出，市场反映很热烈，成为 Macromedia 的核心产品，占领了 80% 的专业网页开发市场，处于绝对领导地位。它像素级的图形设计功能远远超越了其它软件，第一次使“所见即所得”成为大众的现实。而与 FALSH 同根同源的帧操作特性，的确为动态网页吸引了大量用户。

试想如果没有 Kevin Lynch 在图形界面和用户可用性上的丰富经验，没有融入开放结构的一贯理念，Dreamweaver 不可能兼容 ASP、JSP、PHP 和 ColdFusion 等各类开发工具和资源；不可能既满足初学者对简单易用的要求，又满足高级开发者对强大功能、跨平台操作的需求；也就不可能在竞争激烈的市场上独领风骚。

现在 Kevin 为 Macromedia 的新产品 Central 忙碌着。这项技术能够为 Flash 的应用创造一种新的网络环境，这有可能是 Macromedia 公司对网络世界的巨大贡献之一。它就好比一个容器，可以根据用户的需要，以在线或离线的方式储存、显示或应用各种资源和资料。Kevin 说：“它将以开发者从未想到的方式集合各种应用程序。”

Kevin Lynch 最崇拜本杰明·弗兰克林。他说“他不仅是一位发明家，还是一位伟大的思想家。我真的很崇拜拥有这两种能力的人。”这也许能解释为什么 Kevin 要把扩展用户界面设计和协助人类交流联系到一起。■

盘点 Rob Pike



方茜 / 文

Rob Pike 何许人也？

作家？不错，Rob 和贝尔实验室的 Kernighan 合作出版了《Unix 编程环境》(The Unix Programming Environment) 和《程序设计实践》(The Practice of Programming)。这两本书都有中文译本，为初学者研习和提高编程技术提供了很好的方法和理念。

天文学家？这是 Rob 的副业。牛烘烘的他曾在 70 年代得到加拿大皇家天文协会多伦多中心的资助，对光污染进行研究，写有相关论文若干篇。更厉害的是，他设计的伽马望远镜差一点被用在航天飞机上（如果他没有开玩笑的话）。

运动员？网上盛传他是 1980 年奥运会射箭银牌得主。呵呵，别中他招儿。Rob 喜欢开玩笑，谁都知道他是在美国工作过数十年的加拿大公民，而这两个国家都没有参加那一年的奥运会。

Rob Pike 就是这样一个人，有着多面的人生。不过，对程序员来说，研发操作系统才是他最经典的人生。

1980 年，Rob 开始在贝尔实验室的 Unix 小组中工作。他写出了第一个 Unix 下的位图视窗系统，著名的 Blit 终端。这个 Blit 为 Unix 系统提供了一套异步多窗口环境，从而突破传统终端的限制，提高了 Unix 进行多重程序设计的能力。

此后，Rob 设计并编写了 Plan 9、Inferno 等新分布式操作系统。这里，最值得一提的是 Plan 9，它是 Rob 号称锐意革新的操作系统，尽管它并没有引起太多人的注意。操作系统在初期有很多个发展方向：多重处理与并行运算、计算机网络化、以线程闻名的“子程序”概念以及后来诞生的“微内核”概念等。Plan 9 承袭了“微内核”概念，让内核提供操作系统的核心功能；同时，Plan 9 在融入 Unix 系统特点的基础上，重新设计元素，包括：用户端的终端、存储固定文件的服务器、更快的 CPU 计算服务器以及提供用户验证和网关特性。目前，Plan 9 的文件系统可以支持所有的系统服务进程，也就是说任何用户可以使用的资源在

文件系统中都可以找到惟一的命名。它通过网络层的协议 9P 来访问所有的资源，支持异构的网络。

Plan 9 是 Rob 的得意之作，可是反响不大。想想这也不奇怪。当今操作系统开发周期越来越长，代价越来越大，而受使用习惯的影响，普通用户越来越集中到少数几个系统上，因此快速成长长期后出现的操作系统难免被冷落，而那些对操作系统的创新变革之举也只能被老系统的测试和修补而掩盖。这很无奈，不过历史就是如此，总是技术创新与市场选择互相博弈的结果。

在贝尔实验室期间，Rob 也同 Ken Thompson (Unix 的发明者之一) 合作开发了 UTF-8，即：8 位 Unicode 转换格式。这是一种无损耗、长度可改变的 Unicode 字符编码，主要用在 Unix、Linux 等类似的 POSIX 系统上。别以为这是个平常物，它提供了一种简便而向后兼容的方法，使得那种完全围绕 ASCII 设计的操作系统，比如 Unix、Linux，也可以使用 Unicode，让基于各种各样 ASCII 扩展的文件交换变得不再困难。

坊间曾流传是 IBM 设计的 UTF-8，然后由 Plan 9 来实现。然而，据 Pike 回忆，事实上是 Ken 和他设计的。1992 年，他们尝试使用取自 ISO 10646 的原始 UTF 让 Plan 9 支持 16 位字符。在邻近封装到系统上运行时，IBM 的 FSS/UTF 研发人员给 Rob 打来电话，说有个 X/Open 委员会的会议。这激发了 Rob 的灵感，他和 Ken 决定将这个 UTF-8 的雏形变为一种标准，并借 X / Open 委员会推广出去。于是，他俩只用了一个晚上就写完了所有的代码，并在一天后完全实现了在 Plan 9 上成功运行。

2002 年，Google 大肆招揽各种操作系统人才，Pike 也被成功挖角。在 Google 里，Rob 继续着他的操作系统之旅，目光主要集中到分布式计算系统的容错能力和负载平衡能力。但愿 Rob 这一跳槽能带来更多更棒的系统。■

中国程序员，昂首走过 2003

中国程序员已经超过 60 万，我们是一个越来越庞大、越来越强大的群体。我们已经有了自己的信念，自己的文化，也已经到了应该整体上审视自己的时候。回首过去的 2003 年中，我们昂起头。

中国的软件行业诞生较晚。起初一批个人英雄者挑起了中国软件业的大梁，造就了中国软件业的雏形，也成为了程序员心中的偶像。数年之后，当中国的软件市场形成了一定规模，程序员数目呈几何倍数增长，各种各样的新技术、新机会如雨后春笋般涌现。进入 21 世纪，软件产业已经在整个 IT 中占到了更大的份额，而基于各种类型的技术、工具、厂商以及进入此热门行业的人员更是不断增加。

时间进入 2003 年，这是软件技术和软件开发工具强势竞争的一年，形成了以 .NET 和 Java 为首的两大技术体系。众多软件厂商围绕这两个主流技术，使出自己的浑身解数，或是单打独斗，或是合纵连横，将整个软件工具市场推向了一个新的高度。众多更高集成度的开发环境、生命周期的套件以及更加成熟的各种测试、管理工具将软件开发的生产效率也提高到了一个更高的层次。纵观整个软件界，群雄争霸的战国时代已然来临……

2003 年，也是一个开发者受到空前重视的一年。众多厂商或是召开开发者大会；或是 MVP 评选，都只有一个目的，牢牢抓住开发者。让他们熟悉自己的开发体系，这将是市场份额争夺中最有效的手段。

2003 年，我们也不应忘记开发源代码所带给我们的惊喜，Linux、Eclipse、JBoss 等等，是它们让我们有了更多的选择，让我们有了更多的掌握。也正是这些不守传统、勇于尝试的开源软件从另外一个方向推动着软件技术的稳步向前。

在 2003 岁末，让我们回首审视这已然成为历史的 2003 年，看看技术的发展走势，了解当前的业界英雄……横卷在手，感悟技术的变迁、公司的兴衰，再展望 2004 年的技术趋势，规划自己的 IT 人生发展之路！

群雄争霸的时代

——2003 年软件技术大盘点

2003 年，是软件技术和软件开发工具强势竞争的一年，形成了以 .NET 和 Java 为首的两大技术体系。众多软件厂商围绕这两个主流技术，使出自己的浑身解数，或是单打独斗，或是合纵连横，将整个软件工具市场推向了一个新的高度。众多更高集成度的开发环境、生命周期的套件以及更加成熟的各种测试、管理工具将软件开发的生产效率也提高到了一个更高的层次。纵观整个软件界，一幅群雄争霸的时代已然来临……

2003 年，也是一个开发者受到空前重视的一年。众多厂商或是召开开发者大会；或是 MVP 评选，都只有一个目的，牢牢抓住开发者。让他熟悉自己的开发体系，这将是市场份额争夺中最有效的手段。

2003 年，我们也不应忘记开发源代码所带给我们的惊喜，Linux、Eclipse、JBoss 等等，是它们让我们有了更多的选择，让我们有了更多的掌握。也正是这些不守传统、勇于尝试的开源软件从另外一个方向推动着软件技术的稳步向前。

围绕此目的，本专题共分为五部分：Java 大时代、.NET 的过去与未来、波澜不惊的 C++、稳步求进的开源世界、数据库的嬗变。

Java 大时代

透明 / 文

这是最好的时代，这是最坏的时代

——狄更斯，《双城记》

在 Java 的历史上，2003 年注定会是一个值得纪念的年份——哪怕除了“Tiger”之外没有其他任何事。以百兽之王作为代号的 J2SE 1.5 是 Java 进入第二版之后改变最大的一个版本，泛型、新的循环语句、自动装箱/拆箱、静态导入等新特性修改了 Java 的语法，这对于开发者、厂商、Sun 公司和 Java 本身都将是一个巨大的挑战。围绕着 J2SE 1.5，Sun 提出了“让 Java 更简单”的口号，全世界的 Java 人都在翘首期盼这个故事的下一章节。到目前为止，J2SE 1.5 只推出了两个 Early Access 版本（实际上是预编译器），计划在 2004 年夏季正式发布。联想到 Websphere 直到最

近才发布支持 J2SE 1.4 的版本，也许你不应该期望 Tiger 能够很快进入你的项目——还得继续忍受没有泛型容器的痛苦。

整整一年里，Rickard Oberg 不断地向世人证明他在 J2EE 社群的地位——我们还记得去年底他揭穿 .NETPetstore 老底的那一幕。在 Oberg 看来，“要不要 AOP”早已不是一个问题，“如何 AOP”才是。于是我们看到，Oberg 的好友 Jon Tirsén 发布了基于动态代理的 AOP 框架 Nanning，Oberg 和 Tirsén 一起加入了 PicoContainer 项目，Oberg 在 weblog 上写“Why IoC”，Oberg 说自己在项目里获得了 AOP 的新突破……和从前一样，J2EE 开发者们也开始小心翼翼地触摸 Oberg 的爱物：eXo 是基于 AOP 的 portal，Workflow 是基于 AOP 的工作流，Spring 实现了 AOP-Alliance 制订的 AOP 接口，BEA 和 IBM 也想在应用服务器里加上 AOP 的特性。可是，Rickard，你何必要在圣诞节前那么开心地说“Bye bye JBoss”呢——只为 JBoss 终于把你的照片从网站上拿掉？多少有些孩子气的言行，会让多少人扼腕叹息。

如果你在一家做 J2EE 的企业供职，我愿意和你打赌：你们公司的产品中肯定有一套 portal。在忍受了 JetSpeed 的缓慢和 Websphere Portal 的