Managing Projects with GNU Make

第三版
完全修订版

GNU Make

项目管理

O'REILLY®
東南大學出版社

Robert Mecklenburg 著
*O'Reilly Taiwan*公司 编译

GNU Make项目管理

第三版

Robert Mecklenburg 著

O'Reilly Taiwan 公司 编译

O'REILLY®

Beijing • Cambridge • Farnham • Köln • Paris • Sebastopol • Taipei • Tokyo

O'Reilly Media, Inc. 授权东南大学出版社出版

东南大学出版社

图书在版编目（CIP）数据

GNU Make 项目管理：第 3 版 /（美）梅克伦伯格（Mecklenburg, R.），著；O'Reilly Taiwan 公司编译 . －南京：东南大学出版社，2006.7

书名原文：Managing Projects with GNU Make,Third Edition

ISBN 7-5641-0352-3

I. G... II. ①梅 ... ② O... III. 操作系统（软件），GNU Make IV. TP316.7

中国版本图书馆 CIP 数据核字（2006）第 043509 号

江苏省版权局著作权合同登记

图字：10-2005-288 号

书　　名 / GNU Make 项目管理 第三版

书　　号 / ISBN 7-5641-0352-3

责任编辑 / 张烨

封面设计 / Edie Freedman，张健

出版发行 / 东南大学出版社（press.seu.edu.cn）

地　　址 / 南京四牌楼 2 号（邮编 210096）

印　　刷 / 扬中市印刷有限公司

开　　本 / 787 毫米 × 980 毫米　16 开本　18.75 印张　315 千字

版　　次 / 2006 年 7 月第 1 版　2006 年 7 月第 1 次印刷

印　　数 / 0001-3000 册

定　　价 / 37.00 元（册）

O'Reilly Media, Inc. 介绍

为了满足读者对网络和软件技术知识的迫切需求，世界著名计算机图书出版机构O'Reilly Media, Inc. 授权东南大学出版社，翻译出版一批该公司久负盛名的英文经典技术专著。

O'Reilly Media, Inc. 是世界上在 Unix、X、Internet 和其他开放系统图书领域具有领导地位的出版公司，同时也是联机出版的先锋。

从最畅销的《The Whole Internet User's Guide & Catalog》（被纽约公共图书馆评为二十世纪最重要的50本书之一）到GNN（最早的Internet门户和商业网站），再到WebSite（第一个桌面PC的Web服务器软件），O'Reilly Media, Inc.一直处于Internet发展的最前沿。

许多书店的反馈表明，O'Reilly Media, Inc. 是最稳定的计算机图书出版商 —— 每一本书都一版再版。与大多数计算机图书出版商相比，O'Reilly Media, Inc. 具有深厚的计算机专业背景，这使得O'Reilly Media, Inc. 形成了一个非常不同于其他出版商的出版方针。O'Reilly Media, Inc. 所有的编辑人员以前都是程序员，或者是顶尖级的技术专家。O'Reilly Media, Inc. 还有许多固定的作者群体 —— 他们本身是相关领域的技术专家、咨询专家，而现在编写著作，O'Reilly Media, Inc. 依靠他们及时地推出图书。因为O'Reilly Media, Inc. 紧密地与计算机业界联系着，所以O'Reilly Media, Inc. 知道市场上真正需要什么图书。

目录

第二部分 高级与特别的议题

序

make实用程序是一个令人满意的仆人，它总是随伺在侧、给人方便，对你而言就像是不可或缺的伙伴。make起初是一个未能充分发挥潜力的职员，你将一些临时的工作丢给它做，然后它渐渐掌控了整个企业（这是许多小说和电影中常见的情节）。

就在每个项目都被我改成以make为基础之际，我的老板——本书第一版的作者Steve Talbott注意到了我的狂热行为，邀请我为本书编写第二版。在我的人生历练中这的确是关键性的成长（也是相当大的冒险），我因此而进入O'Reilly的美妙世界，但是我们实在不知道第二版能在市场上存活多久。一版能存活13年吗？

下面的概述列出了自本书第二版发行以来make的发展：

- 本书第二版发行时GNU版的make已经是大多数程序员的选择，俨然成为业界的标准。
- GNU/Linux的兴起让GNU编译器工具链的使用更为广泛，其中包括GNU版的make。举例来说，Linux内核本身就相当依赖GNU版make所提供的扩展功能，正如本书第十一章所描述的那样。
- 以BSD的变体（Darwin）为核心（core）的Mac OS X继续向GNU工具链以及GNU版的make迈进。
- 越来越多的诀窍被发现，让你能够健全、无错误、可移植及灵活地使用make。对于大型项目上常见的问题，已经在社群中逐渐形成了标准的解决方案。已经到了将这些解决方案立言成书的时候了，这就是本书要做的事情。
- 尤其是出现了将make应用在C++和Java语言中的新需求，make开发出来时这些语言尚不存在。以下的例子可以说明make出现的年代：最初的make具有两项特

殊的功能，一个是支持FORTRAN的两个变体（这个功能还在！），一个是与SCCS的不怎么有用的集成。

- 即使有诸多限制，make仍旧是所有计算机开发项目中最重要的工具，恐怕当年对make的批评者或洞察者都没预料到这一点。这13年来，有意取代make的新工具如雨后春笋般不断推陈出新，它们都想超越make在设计上的限制，其中也不乏众多值得赞赏的巧思。不过，make的简单易用仍使它立于不败之地。

当我观察到这些趋势之后，十年前为本书编写新版的想法又涌上心头。不过我意识到自己的经验浅薄并不足以担负此重责大任。最后，我找到了Robert Mecklenburg，他的专业能力得到O'Reilly所有同仁的肯定。能将这本书交由他全权负责实在太好了，我则退居幕后成为本书的编辑，这让我的名字又可以出现在本书的版权页上。

Robert对他的博士学位保持低调，不过他思考的深度和精确度却在本书中展露无遗。或许更重要的是他把焦点放在实用性上。他会告诉你如何执行得更有效率，以及让你知道如何进行调试。

这是一个重大的时刻：O'Reilly最早期和最持久的一本书出新版了。坐下来，了解一下，一个保守的小工具何以有此能耐让几乎每个项目都要使用它。不要安于老旧而无法令人满意的*makefile*——开发你的潜力就在今朝。

— Andy Oram
Editor, O'Reilly Media
August 19, 2004

前言

迈向第三版

1979年首次遇到make的时候我还是Berkeley的大学生。当时我正为能够使用“最新的”设备感到兴奋不已：一台DEC PDP 11/70（具有128 kilobytes的RAM）、一台ADM 3a（具有屏幕的终端机）、Berkeley Unix以及另外20个同时上线的用户！记得有一次，大家在赶作业的时候从我键入账号名称到我看到命令提示符一共花了5分钟的登录时间。

毕业之后，当我再次使用Unix的时候已经是1984年了，这次我是美国太空总署Ames研究中心的程序员。我们买了第一部以微型计算机为基础的Unix系统，它的配置包含一个68000（不是68010或20）微处理器、1 megabyte的RAM以及Unix Version 7——只能有6个同时上线的用户。我所参与的最后一个项目就是使用C程序语言以及yacc/lex命令语言（当然还包括make）来实现出一个交互式卫星数据分析系统。

1988年，我返回学校并且参与构建“曲线几何建模”（spline-based geometric modeler）的项目。这个系统使用了大约120000行的C程序，涵盖了20个左右的可执行文件。这个系统的编译方法就是使用一个手工打造的工具genmakefile（它的功能类似于imake）将*makefile*模板展开成一般的*makefile*。这个工具会进行简易的文件引入、条件编译以及使用自定义的逻辑来管理源文件树和二进制文件树。当时普遍认为，make必须使用此类封装程序（wrapper）才算是完整的编译工具。直到几年前我发现GNU项目以及GNU make，我才了解到大概不再需要封装程序了。我重新建立了不使用模板或产生器的编译系统。这个编译系统被移植到了5种Unix版本中，并且包括独立的源文件树和二进制文件树、每夜自动编译，以及以编译系统填补短缺的二进制文件的方式来支持开发人员进行部分调出的动作。

下一个重要的make使用经验是在1996年。这是一个商用CAD系统，我的工作是将200万行的C++（以及40万行Lisp）程序从Unix移植到Windows NT，使用Microsoft C++编译器进行编译的工作。我就是在那个时候发现Cygwin项目的。这个编译系统还支持独立的源文件树和二进制文件树、多种Unix、几种图形功能、每夜自动编译和测试、以引用编译结果让开发人员进行部分调出的动作。

2000年，我的工作是以Java编写实验室信息管理系统。这是我工作了那么多年之后首次遇到的完全不同的开发环境之一。参与项目的程序员大部分来自于Windows背景，而且Java似乎是他们所使用的第一个程序语言。这个编译环境几乎是由一个商用Java集成开发环境（Integrated Development Environment，简称IDE）所产生的项目文件构成的。尽管项目文件已经可以使用，但是它却很少被马上拿来使用，程序员们通常会坐在彼此的屏幕前处理许多编译问题。

当然，我开始使用make来编写编译系统，但是一个奇特的事情发生了：许多开发人员根本不愿意使用任何命令行工具。此外，许多人无法准确理解环境变量、命令行选项之类的概念，也不知道这些工具如何用来编译程序。IDE将这些问题都藏起来了。为解决这些问题，我所编写的编译系统变得更加复杂。我开始加入更好的错误信息、先决条件的检查、开发人员机器配置的管理以及对IDE的支持。

于是，GNU make使用手册被我读过不少于10次。当我在寻找更多资料的时候，我发现到了本书的第二版，它提供了许多有用的数据，不过缺少GNU make方面的细节。这并不令人感到惊讶，想想看它的出版时间。这是一本经得起时间考验的书，不过到了2003年是需要更新了。本书第三版的重点是GNU make。诚如Paul Smith（GNU make的维护者）所说："编写具可移植性的'*makefile*'是在自找麻烦，使用具可移植性的*make*吧！"

第三版有哪些新的内容

本书几乎所有内容都是新的。我将这些内容划分成三个部分：

第一部分 基本概念。适度说明GNU make的功能以及这些功能的使用方法。

第一章 如何编写简单的makefile。以简单但完整的范例来简介make。这一章将会说明make的基本概念，如工作目标以及必要条件，并解释*makefile*的语法。这应该可以让你具备编写*makefile*的能力。

第二章 规则。将会探讨规则的结构和语法。除了旧式的后缀规则（suffix rule），这一章还会非常详细地说明具体规则（explicit rule）和模式规则（pattern rule）。特殊的工作目标以及依存关系的产生也会在此处被讨论到。

第三章 变量与宏。将会说明简单变量与递归变量。这一章还会探讨当变量被展开时*makefile*是如何被解析的以及条件指令的处理。

第四章 函数。将会查看GNU make所支持的各种内置函数。此处也会以各种范例，包含一般的与深层次的概念来说明用户自定义函数。

第五章 命令。将会说明脚本的细节，内容涵盖脚本的解析与求值。此处也会探讨命令修饰符、命令结束状态的检查以及环境变量。我们还会探索命令行长度限制的问题，以及解决这些问题的若干方法。此刻你已经能够了解本书所要探讨的所有GNU make的功能。

第二部分 深入与特别的议题。包含了比较多的议题，如将make应用在大型项目上、可移植性以及调试等。

第六章 大型项目的管理。将会探讨以make编译大型项目时可能会遇到的许多问题。第一个议题是如何进行make的递归调用，以及如何使用单一非递归的*makefile*来实现前者所用到的许多*makefile*。此外，我们还会探讨大型系统的其他议题，像文件系统的配置、项目组件的管理以及自动化编译与测试。

第七章 具可移植性的makefile。将会探讨*makefile*在各种Unix操作系统与Windows系统间的可移植性。此处还会讨论Cygwin的Unix模拟环境，以及不具可移植性的文件系统功能与工具所引发的问题。

第八章 C与C++。将会举几个“如何分开源文件树和二进制文件树以及如何建立仅供读取的源文件树”的特例。此处会再次提到依存关系分析，不过这次将会强调与程序语言有关的解决方案。这一章与下一章将会探讨第六章所提到的许多延伸议题。

第九章 Java。将会说明如何把make应用在以Java为基础的开发环境中。此处还会提到管理CLASSPATH变量、编译大量文件、创建jar以及构造Enterprise JavaBeans的技术。

第十章 改进make的性能。首先会回顾若干make操作的性能特性，以作为如何编写具有效率的*makefile*的立论基础。此处还会探讨如何找出和解决瓶颈的技术，以及GNU make的并行编译功能。

第十一章 makefile实例。将会提供两个复杂的*makefile*实例。第一个实例是用来建立本书的*makefile*。这是个值得一看的例子，部分是由于这是对自动化的相当极端的应用，部分是由于它将make应用在非传统的领域。另一个实例摘录自Linux 2.6 kbuild系统。

第十二章 makefile的调试。我们将会钻研修复*makefile*的魔法。这一章将会介绍“如何发现make背地里在做什么以及如何减轻开发期痛苦”的技术。

第三部分 附录。包含了补充资料。

附录一 运行make。提供了GNU make命令行选项的参考指导。

附录二 越过make的极限。将会探索GNU make两个不太可能被用到的功能：管理数据结构以及进行算数运算。

排版约定

斜体字（*Italic*）

用来表示新项目、网址、电子邮件地址、文件名、文件扩展名、路径名称以及目录。

等宽字（Constant Width）

用来表示源代码命令、命令行选项、文件的内容或是命令的输出。

等宽黑体字（**Constant Width Bold**）

用来表示应该由用户逐字键入的命令或其他文字。

等宽斜体字（*Constant Width Italic*）

其所标示的文字应该被替换成用户所提供的值。

范例程序代码的使用办法

这本书可以协助你把工作做好。一般而言，你可以在自己的应用程序和说明文件中使用本书的程序代码。除非你要重制重要的程序代码，否则不必取得我们的许可。例如，你使用本书的程序代码片段写了一个应用程序，并不需要取得我们的许可；但是，把O'Reilly书籍的程序范例制作成光盘贩卖或散布，就需要取得授权。引用本书的文字和范例程序代码来回答问题，不需要取得许可；但把本书大量的程序范例整合到你的产品的说明文件中，则需要取得授权。

虽然不是必要，但若能注明来源我们会很感谢。注明来源通常包括书名、作者、出版商以及ISBN。例如：Managing Projects with GNU Make, Third Edition, by Robert Mecklenburg. Copyright 2005 O'Reilly Media, Inc., 0-596-00610-1。

如果你对书中程序范例的使用情况有别于上述情况，不用客气，尽管和我们联络：*permissions@oreilly.com*。

建议与评论

本书的内容都经过测试，尽管我们做了最大的努力，但错误和疏忽仍然是在所难免的。如果你发现有什么错误，或者是对将来的版本有什么建议，请通过下面的地址告诉我们：

美国：

O'Reilly Media, Inc.
1005 Gravenstein Highway North
Sebastopol, CA 95472

中国：

100080 北京市海淀区知春路 49 号希格玛公寓 B 座 809 室
奥莱理软件（北京）有限公司

询问技术问题或对本书的评论，请发电子邮件到：

info@mail.oreilly.com.cn

与本书有关的在线信息（包括勘误、范例程序、相关链接）：

原文书

http://www.oreilly.com/catalog/make3/index.html

中文书

http://www.oreilly.com.cn/book.php?bn=7-5641-0352-3

最后，您可以在 WWW 上找到我们：

http://www.oreilly.com
http://www.oreilly.com.cn

致谢

我要感谢 Richard Stallman 所编织的梦想以及对美梦终能成真的信心。当然，没有 Paul Smith，GNU `make` 不会有今日的表现，谢谢你。

我也要感谢我的编辑 ——Andy Oram，对我始终如一的支持和热诚。

应该感谢 Cimarron Software 为我提供一个环境，让我得以开始此计划。还应该感谢 Realm Systems 为我提供一个环境，让我得以完成此计划。尤其要感谢 Doug Adamson、Cathy Anderson 和 Peter Bookman 等人。

谢谢本书的评阅者们，Simon Gerraty、John Macdonald 和 Paul Smith，提供了许多见解深刻的意见，并且修正了许多令我难为情的错误。

应该感谢的人还有为本书付出贡献的：Steve Bayer、Richard Bogart、Beth Cobb、Julie Daily、David Johnson、Andrew Morton、Richard Pimentel、Brian Stevens 以及 Linus Torvalds。

还要感谢为我在暴风之海提供安全避风港的如下集体：Christine Delaney、Tony Di Sera、John Major 和 Daniel Reading。

最后，感谢我的妻子 Maggie Kasten 以及我的两个孩子 William 与 James，在最后这 16 个月期间对我的支持、鼓励以及爱。

第一部分

基本概念

在第一个部分里，我们会将重点放在make的功能上：它们能够做什么以及如何正确地使用它们。我们首先会做个简介，并且告诉你如何创建你的第一个*makefile*。这部分的内容涵盖了make的规则、变量、函数以及脚本。

看完第一部分之后，你将会获得相当完整的GNU make操作知识，并且掌握许多高级的用法。