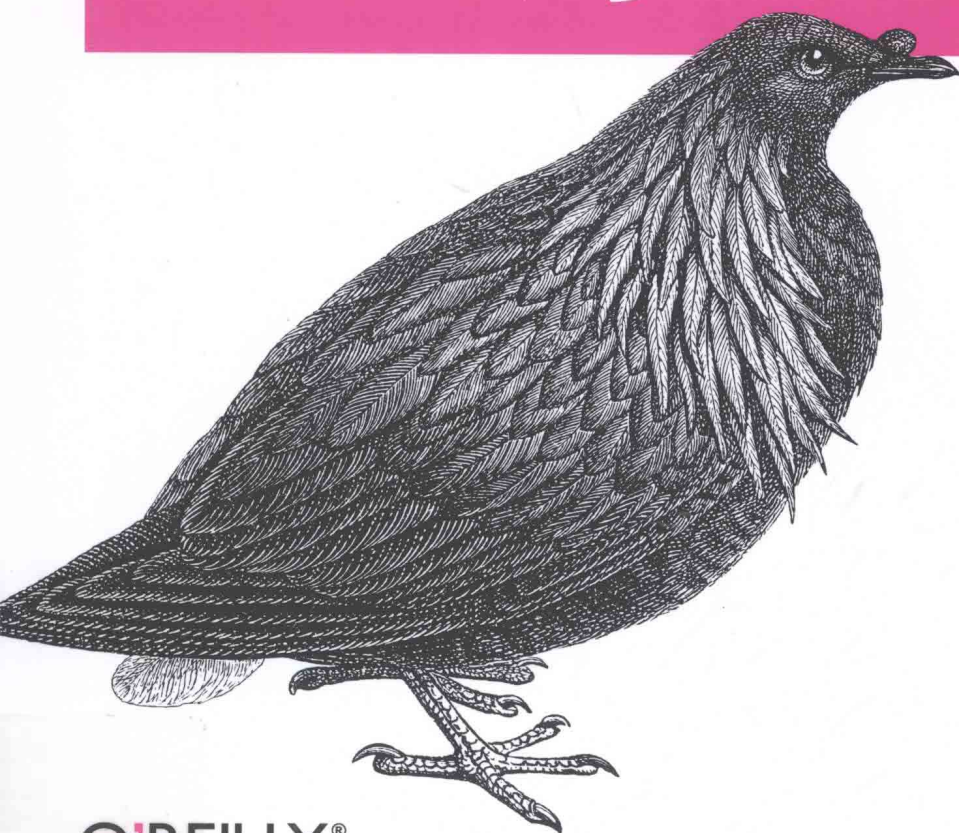


flex & bison



flex 与 bison

(中文版)



O'REILLY®

东南大学出版社

John Levine 著

陆军 译

flex与bison

John Levine 著

陆军 译

O'REILLY®

Beijing • Cambridge • Farnham • Köln • 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

flex 与 bison/ (美) 利文 (Levine, J.) 编, 陆军译. —南京: 东南大学出版社, 2011.3

书名原文: flex & bison

ISBN 978-7-5641-2605-6

I. ①f… II. ①利… ②陆… III. ①Linux 操作系统 ②UNIX 操作系统 IV. ①TP316.8

中国版本图书馆 CIP 数据核字 (2011) 第 003003 号

江苏省版权局著作权合同登记

图字: 10-2010-274 号

©2009 by O'Reilly Media, Inc.

Simplified Chinese Edition, jointly published by O'Reilly Media, Inc. and Southeast University Press, 2011. Authorized translation of the English edition, 2009 O'Reilly Media, Inc., the owner of all rights to publish and sell the same.

All rights reserved including the rights of reproduction in whole or in part in any form

英文原版由 O'Reilly Media, Inc. 出版 2009。

简体中文版由东南大学出版社出版 2011。英文原版的翻译得到 O'Reilly Media, Inc. 的授权。此简体中文版的出版和销售得到出版权和销售权的所有者——O'Reilly Media, Inc. 的许可。

版权所有, 未得书面许可, 本书的任何部分和全部不得以任何形式复制。

flex 与 bison

出版发行: 东南大学出版社

地 址: 南京四牌楼 2 号 邮编: 210096

出版人: 江建中

网 址: <http://www.seupress.com>

电子邮件: press@seu.edu.cn

印 刷: 扬中市印刷有限公司

开 本: 787 毫米 × 980 毫米 16 开本

印 张: 17.5 印张

字 数: 343 千字

版 次: 2011 年 3 月第 1 版

印 次: 2011 年 3 月第 1 次印刷

书 号: ISBN 978-7-5641-2605-6

印 数: 1~4000 册

定 价: 49.00 元 (册)

本社图书若有印装质量问题, 请直接与读者服务部联系。电话 (传真): 025-83792328

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20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

前言	1
第1章 Flex和Bison简介	7
词法分析和语法分析	7
正则表达式和词法分析	8
文法与语法分析	16
二义性文法：并不多见	20
添加更多的规则	21
Flex和Bison与手写的词法分析器和语法分析器的对比	22
练习	23
第2章 使用Flex	25
正则表达式	25
Flex词法分析器中的文件I/O操作	29
读取多个文件	30
Flex词法分析器的I/O结构	32
起始状态和嵌套输入文件	34
符号表和重要语汇索引生成器	38
C语言交叉引用	44
练习	51

第3章 使用Bison	53
Bison语法分析器如何匹配输入	53
移进/归约分析	54
Bison语法分析器	57
基于抽象语法树的改进的计算器	58
移进/归约冲突和操作符优先级	63
一个高级计算器	66
使用高级计算器	84
练习	85
第4章 分析SQL	86
SQL概述	86
关系操作	88
使用SQL的三种方法	88
从SQL到逆波兰式 (RPN)	89
词法分析器	90
语法分析器	99
SQL分析器的Makefile	121
练习	122
第5章 Flex规范参考	123
Flex结构规范	123
BEGIN	124
C++词法分析器	125
上下文相关性	125
定义 (替换)	126
ECHO	127
输入管理	127
Flex库	130
交互模式和批处理模式的词法分析器	130
行号和yylineno	130
文字块	131

单一程序中的多重词法分析器	131
编译词法分析器的选项	133
Flex词法分析器的可移植性	133
可重入词法分析器	134
正则表达式语法	137
REJECT	139
从yylex()返回值	140
起始状态	140
unput()	141
yyinput() yyunput()	142
yylen	142
yyless()	142
yylex()和YY_DECL	143
yyomore()	143
yyrestart()	144
yy_scan_string和yy_scan_buffer	144
YY_USER_ACTION	144
yywrap()	144

第6章 Bison规范参考 145

Bison语法结构	145
二义性和冲突	148
Bison程序的问题	150
C++语法分析器	151
%code块	151
结束标记	151
错误记号和错误恢复	152
继承属性（\$0）	152
词法反馈	154
文字块	155
文字记号	155
位置	156

%parse-param	156
Bison语法分析器的可移植性	157
优先级和结合性声明	158
递归规则	160
规则	161
特殊字符	162
%start声明	164
符号值	164
记号	165
可变语法和多重语法	168
多重语法分析器	169
y.output文件	170
Bison库文件	171
YYABORT	172
YYACCEPT	172
YYBACKUP	173
yyclearin	173
yydebug和YYDEBUG	173
YYERROR	174
yyerror()	175
yyparse()	175
YYRECOVERING()	175

第7章 二义性和冲突..... 177

指针模型和冲突	177
冲突类型	179
语法分析器状态	181
name.output的内容	182
归约/归约冲突	183
移进/归约冲突	184
复习name.output中的冲突	187
常见的冲突例子	188

IF/THEN/ELSE.....	190
你如何解决冲突?	192
IF/THEN/ELSE (移进/归约冲突)	192
嵌套循环 (移进/归约冲突)	194
表达式优先级 (移进/归约冲突)	195
总结	199
练习	199
第8章 错误报告和恢复	200
错误报告	200
错误恢复	208
Bison错误恢复.....	208
编译器错误恢复	211
练习	212
第9章 Flex和Bison进阶	213
纯词法分析器和纯语法分析器.....	213
GLR分析	234
C++语法分析器.....	239
练习	246
附录 SQL语法分析器文法和交叉引用	247
术语表	263

前言

flex和bison是为编译器和解释器的编程人员特别设计的工具，不过它们在其他应用领域也非常有用，因此吸引了许多非编译器编程人员的注意。任何应用程序，只要它在其输入中寻找特定的模式，或者它使用命令语言作为输入，都适合使用flex和bison。而且，flex和bison允许快速的应用程序原型开发，修改简单，维护方便。让我来激发一下你的想象力吧，下面列出了一些使用flex和bison，或者它们的前身lex和yacc，来开发的程序：

- 桌面计算器**bc**
- 工具**eqn**和**pic**，用于数学方程式和复杂图片的排版预处理器
- 用于特定应用设计的大量领域特定语言
- **PCC**，在许多Unix系统中被使用的可移植的C编译器
- flex自身
- SQL数据库语言翻译器

本书内容

第1章，*Flex和Bison简介*，概述了flex和bison如何和为何可以被用来创建编译器和解释器，并且演示了一些简单的应用程序，包括内置于flex和bison的计算器。也介绍了一些我们这本书会用到的术语。

第2章，*使用Flex*，描述了如何使用flex。我们在本章开发了一些flex应用程序来计算文件中的单词数，处理多个和嵌套的输入文件，以及统计C程序中的交叉引用信息。

第3章，*使用Bison*，实现了一个完整的例子，它使用flex和bison开发了一个具有全部功能的桌面计算器，能够支持变量、过程、循环和条件表达式。本章展示了抽象语法树（AST）的用法，它具有强大而易用的数据结构来表示解析过的输入。

第4章，分析SQL，为SQL关系型数据库的MySQL方言开发了一种语法分析器。该分析器能够检查SQL语句的语法，然后把它们翻译成适合于特定解释器的内部形式。本章展示了逆波兰表示法（Reverse Polish Notation，RPN）的用法，它是另外一种高效的组织形式，用来表达和解释分析结果。

第5章，*Flex*规范参考，和第六章，*Bison*规范参考，为flex和bison程序员提供了详细的特性和选项描述。这两章和后续的两章为flex和bison的初学者提供了他们在开发flex和bison的应用程序时所需的技术信息。

第7章，二义性和冲突，解释了bison的二义性和冲突，它们会导致bison无法为特定语法创建语法分析器的问题。本章接着描述了一些可以被用来定位和解决这类问题的方法。

第8章，错误报告和恢复，讨论了编译器或者解释器的设计人员如何来定位、识别和报告编译器输入中的错误的技术。

第9章，*Flex*和*Bison*进阶，包括了可重入词法分析器和语法分析器、通用的自左向右语法分析器（它能够处理普通bison语法分析器无法处理的语法）以及C++的接口。

附录部分提供了第4章所讨论的SQL语法分析器的完整语法和交叉引用信息。

术语表列出了语言和编译器理论中的技术术语。

你需要熟悉C语言，因为大部分例子是由C语言、flex或者bison写成的，还有少量例子是由C++、SQL和其他基于特定目的而开发的语言来编写的。

排版约定

本书应用了如下一些印刷排版上的约定：

斜体字 (*Italic*)

用于引入的新术语与概念。

等宽字 (Constant width)

用于程序列表与段落中提到的程序元素，如语句、类、宏、状态、规则、所有代码项以及文件与目录。

等宽黑体字 (Constant width bold)

用于命令或者其他的用户需要逐字键入的文本类型。

等宽斜体字 (Constant width italic)

用于需要使用用户提供的值或者由上下文确定的值替代的文本。

\$

命令行提示符。

[]

包围程序语法描述中的可选项。（方括号本身无需输入）

获取Flex和Bison

经典的lex和yacc由贝尔实验室在20世纪70年代开发，flex和bison则是它们的现代版本。yacc由Stephen C. Johnson首先开发完成。lex由Mike Lesk和Eric Schmidt（现在他正领导着Google）设计，用来与bison协同工作。从第七版的Unix开始，lex和yacc就已经成为标准的Unix辅助工具。

自由软件基金会（Free Software Foundation）的GNU项目帮助发布bison，一种yacc的向前兼容版本。它最初由Robert Corbett和Richard Stallman编写。bison的帮助手册非常好用，特别是一些特性的参考。在BSD和Linux的发布版本中，你都可以找到bison，但是如果你希望使用最新的bison，你可以到它的主页下载：

<http://www.gnu.org/software/bison/>

BSD和GNU项目也发布flex（快速词法分析生成器），“它重写了lex来解决lex中存在的大量错误和缺陷。”flex最初由Jef Poskanzer编写，Vern Paxson和Van Jacobson相当多地改进了它。BSD和Linux的发布版本中包含flex的拷贝，但是如果你希望使用最新版本，它可以在SourceForge中找到：

<http://flex.sourceforge.net/>

本书的范例

本书的范例程序可以在线获得：

<ftp://ftp.iecc.com/pub/file/flexbison.zip>

它们可以通过任何浏览器和FTP客户端下载。zip压缩格式的文件可以通过在类Unix或者Linux系统中流行的自由软件unzip工具来解压，也可以在Windows XP或者更新的窗口系统中作为压缩文件夹来打开。

本书的范例程序在版本为2.5.35的flex和版本为2.4.1的bison上测试通过。

使用代码范例

希望本书能够帮助你完成工作。通常来说，你可以在你的程序和文档中使用本书的代码。除非你引用了显著数量的代码，否则你无需联系我们来获得许可。例如，编写一个使用了本书几段代码的程序就并不需要任何许可，但是销售或者发布带有O'Reilly书籍中的范例的光盘需要获得许可。引用本书和范例代码来回答问题也不需要许可，但把本书中显著数量的范例代码包含到你的产品文档中需要获得许可。

我们感谢，但并不要求，你给出出处。完整信息通常包括书名、作者、出版者和ISBN。例如：“书名、作者、copyright 2008 O'Reilly Media, Inc., 978-0-596-xxxx-x”。

如果你觉得你的范例代码使用方式超出常规的使用方式，或者并不在前面许可的范围内，请通过permissions@oreilly.com来联系我们。

如何联系我们

请将对本书的评价和存在的问题通过如下地址告知出版者：

美国：

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

中国：

北京市西城区西直门南大街2号成铭大厦C座807室（100035）
奥莱利技术咨询（北京）有限公司

O'Reilly的每一本书都有专属网站，你可以在那找到关于本书的相关信息，包括勘误表、示例代码以及其他的信息。本书的网站地址是：

<http://www.oreilly.com/catalog/9780596155988/>

对于本书的评论和技术性的问题，请发送电子邮件到：

bookquestions@oreilly.com

关于本书的更多信息、会议、资料中心和O'Reilly网络，请访问以下网站：

<http://www.oreilly.com/>

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

致谢

任何一本书都是众人的结晶，本书也不例外。我要感谢Tony Mason和Doug Brown，我的《lex & yacc》的共同作者，他们许可我摘录那本书的部分内容。许多人给我的草稿提供了宝贵意见，包括Lorenzo Bettini、Joel E. Denny、Danny Dubé、Ben Hanson、Jan Van Katwijk、Jeff Kenton、Timothy Knox、Chris Morley、Ken Rose和Juha Vihavainen。我要特别感谢Derek M. Jones，他在极短的时间里为每页提供了详尽的评审意见。Simon St. Laurent，为我长期受累的编辑，领导了本书的全部编辑和出版的过程而毫无怨言。

Flex和Bison简介

flex和bison是用来生成程序的工具，它们所生成的程序能够处理结构化输入。最初flex和bison是用来生成编译器的，但是后来它们被证明在其他很多领域也非常有效。在第1章里，我们将以它们背后的一部分（不是太多）原理作为介绍的开始，然后我们会深入讲解一些具体的使用例子。

词法分析和语法分析

最早的编译器可以追溯到20世纪50年代，这些编译器使用很特别的技术来分析它们所编译的程序源代码的语法。在60年代，这个领域获得了很多学术层面的关注，在70年代初，语法分析成为一个众所周知的领域。

在这个领域中，一个关键的想法是把分析工作分成两个部分：词法分析（*lexical analysis*，或称*scanning*）和语法分析（*syntax analysis*，或称*parsing*）。

简单来说，词法分析把输入分割成一个个有意义的词块，称为记号（*token*），而语法分析则确定这些记号是如何彼此关联的。例如，看一下这个C代码的片断：

```
alpha = beta + gamma ;
```

词法分析器把这段代码分解为这样一些记号：alpha、等号、beta、加号、gamma和分号。接着语法分析器确定了beta+gamma是一个表达式，而这个表达式被赋给了alpha。

获取Flex和Bison

大多数Linux和BSD系统自带flex和bison作为系统的基础部分。如果你的系统没有包含它们，或者包含的是过时的版本，安装它们也很容易。

flex是一个SourceForge项目 (<http://flex.sourceforge.net/>)。2009年初的版本是2.5.35。通常从一个版本到另一个版本的变化是很小的，如果你的版本接近.35，你就没有必要更新版本，不过有些系统自带的版本是2.5.4或者2.5.4a，这种版本就显得更老了。

bison可以从<http://www.gnu.org/software/bison>获得。2009年初的版本是2.4.1。bison还处在相当活跃的开发中，所以值得我们获取一个最新的版本来看看有哪些新引入的特性。比如，版本2.4就增加了对Java语法分析的支持。BSD用户通常可以使用Ports Collection来安装当前版本的flex和bison。Linux用户可以找到最新的RPM。如果没有的话，flex和bison都使用标准的GNU编译过程，因此为了安装它们，我们可以从这个网站下载并解压最新的flex和bison的压缩包，运行./configure以及make来编译它们，然后我们需要切换到超级用户并使用make install进行安装。

flex和bison都依赖于GNU m4 宏处理器。Linux和BSD都应该有m4，如果没有或者当前的版本太老的话，最新的m4也可以从<http://www.gnu.org/software/m4>获得。

对于Windows用户来说，bison和flex都被包含在Cygwin Linux模拟环境 (<http://www.cygwin.com>) 中。你可以使用由Cygwin开发工具或者Windows本地开发工具所生成的C或者C++代码。

正则表达式和词法分析

词法分析通常所做的就是在输入中寻找字符的模式 (pattern)。例如，在C程序中，整型常量是一个或多个数字的字符串，变量名称是一个字母后面跟着零个或者多个字母和数字，而各种各样的操作符则是一个或者成对的字符。一种简洁明了的模式描述方式就是正则表达式 (*regular expression*，常简写为*regex*或*regexp*)。编辑器ed和vi以及搜索程序egrep就使用这种方式来描述其所需要查找的文本。flex程序主要由一系列带有指令的正则表达式组成，这些指令确定了正则表达式匹配后相应的动作 (*action*)。由flex生成的词法分析器可以读取输入，匹配输入与所有的正则表达式并且执行每次匹配后适当的关联动作。flex会把所有的正则表达式翻译成一种高效的内部格式，这使它几乎可以同时处理所有需要匹配的模式，因此它的速度可以成百倍地提高^(注1)。

注1： 内部格式使用确定性有穷自动机 (Deterministic Finite Automation, DFA)。值得庆幸的是，对于确定性有穷自动机，你唯一需要知道的就是它们非常快，而且它们的处理速度与模式的个数和复杂程度无关。