



2019~2020第二学期

系统软件开发实践

计算机科学与技术学院

张博

Email: zbcumt@163.com



- 链接:

https://pan.baidu.com/s/14Rh_16ak04-R1JwV9FP5Bg


- 提取码:

- 3z8f

课程内容

利用Flex/Bison构造编译器

(实验文件夹里有课程PPT和参考资料)

- 1. Flex 理论与练习(8学时)
- **2. Bison理论与练习(8学时)** 
- 综合实验：
 - 3. 使用Flex和Bison开发了一个具有全部功能的桌面计算器 (16学时)
 - 4. 编译器后端实现及目标代码生成 (16学时)

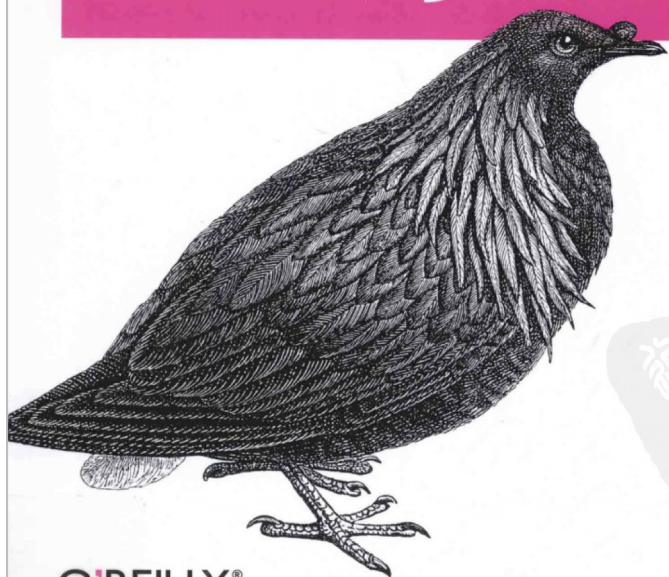
参考书

flex & bison



flex 与 bison

(中文版)



O'REILLY®

东南大学出版社

John Levine 著

陆军 译

参考书
PDG

Bison 理论与练习

Bison实验1（4学时）

•实验内容：

- 1. 阅读《Flex/Bison.pdf》第一章，第三章，掌握Bison基础知识。
- 2. 利用Bison设计一个简单的语法分析器，掌握移进/规约分析，掌握语法分析树，掌握抽象语法树。

•实验成果：

- 提交实验报告。

Bsion (Syntax Analyzer) 简介

- **词法分析 (Flex)** 的核心就是识别源代码。

它按照特定的正则表达式划分成一系列的
标记(Token)。

- 如、 alpha = beta + gamma ; 以 C 语言的划分方式可以分成 **alpha**、**=**、**beta**、**+**、**gamma**、**;** 这些标记。

- **语法分析 (Bison)** 的任务是确定这些标记是如何彼此关联的。

- 如、 **beta + gamma** 是一个表达式，而该表达式的值被赋给 **alpha** 。

YACC & Bison

- **YACC** (Yet Another Compiler Compiler) 是 Unix/Linux 上一个用来生成编译器的编译器 (编译器代码的生成器) 。
- **YACC** 生成的编译器是用 C 语言写成的语法解析器 (Parser) , 需要与词法解析器 **Lex** 一起使用。
- **Bsion** 是 **YACC** 的开源版本, 需要配合 Flex 使用, 主页 :

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

Windows环境下 安装Bison

- 百度网盘：
- 下载 [bison-2.4.1-setup.exe](#)
- 或
- <http://sourceforge.net/projects/gnuwin32/files/bison/2.4.1/>

GnuWin

GnuWin

Provides native Win32 open source ports and utilities

Brought to you by: [gnuwin32](#)

[Summary](#) | [Files](#) | [Reviews](#) | [Support](#) | [Wiki](#) | [Mailing Lists](#) | [Code](#) | [Tickets ▾](#) | [News](#) | [Discussion](#)

Looking for the latest version? [Download sed-4.2.1-setup.exe \(2.0 MB\)](#)

[Home](#) / [bison](#) / 2.4.1

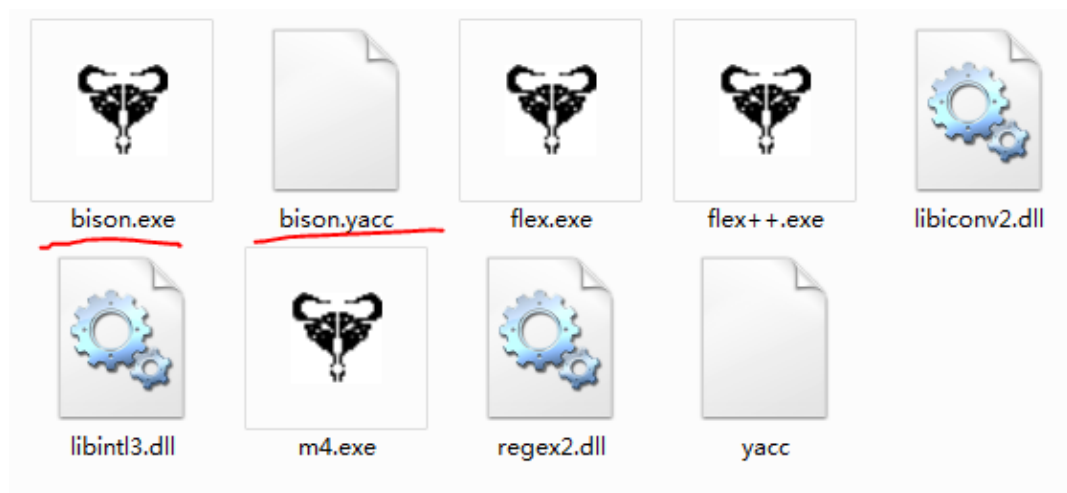


Name ▴	Modified ▴	Size ▴	Downloads / Week ▴	
↑ Parent folder				
bison-2.4.1-src-setup.exe	2009-05-04	2.3 MB	21 <input type="checkbox"/>	
bison-2.4.1-setup.exe	2009-05-04	3.8 MB	263 <input type="checkbox"/>	
bison-2.4.1-src.zip	2009-05-04	3.8 MB	44 <input type="checkbox"/>	
bison-2.4.1-bin.zip	2009-05-04	963.7 kB	68 <input type="checkbox"/>	

Windows环境下 安装Bison



与Flex安装在同一目录。



步骤1：

编写Bison源程序 (Name.y)

【目标】：利用 Bison+Flex 设计一个语法分析程序，对一段文本进行解析，并输出结果。

```
• %{
• typedef char* string;
• #define YYSTYPE string
• #include <stdio.h>
• %}

• %token NAME EQ AGE

• %%
• file : record
•       | record file
•       ;
• record : NAME EQ AGE {printf("%s is %s years old!!!\n", $1, $3);}
•       ;
• %%

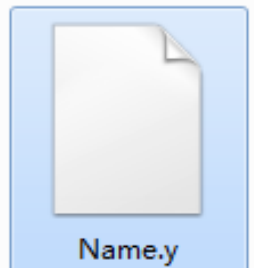
• int main()
• {
•     yyparse();
•     return 0;
• }
• int yyerror(char *msg)
• {
•     printf("Error encountered: %s \n", msg);
•     return 0;
• }
```



保存到Bison
安装目录



bison.exe

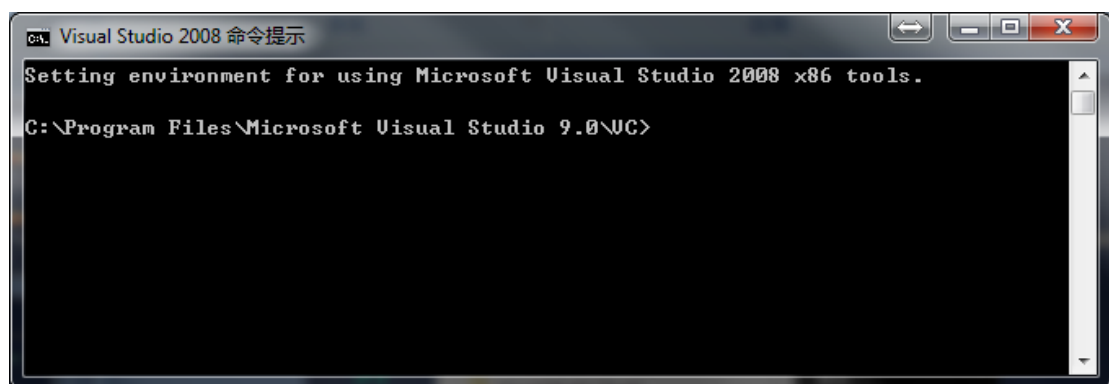
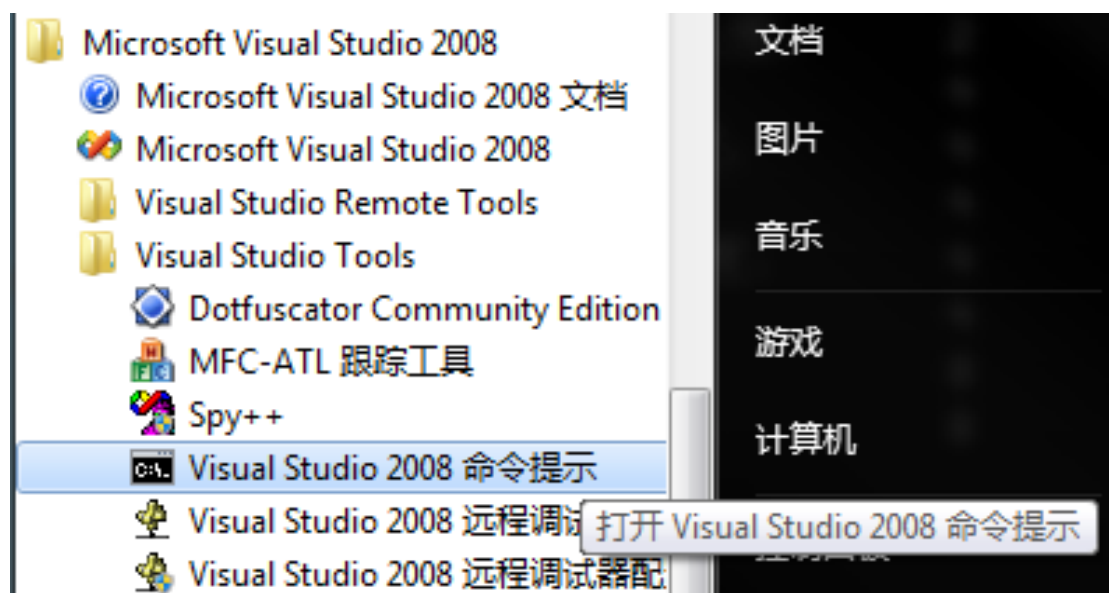


Name.y

步骤2：

使用Bison『编译』源程序（Name.y）

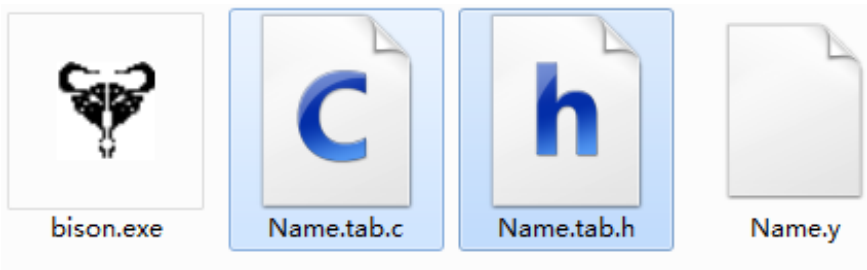
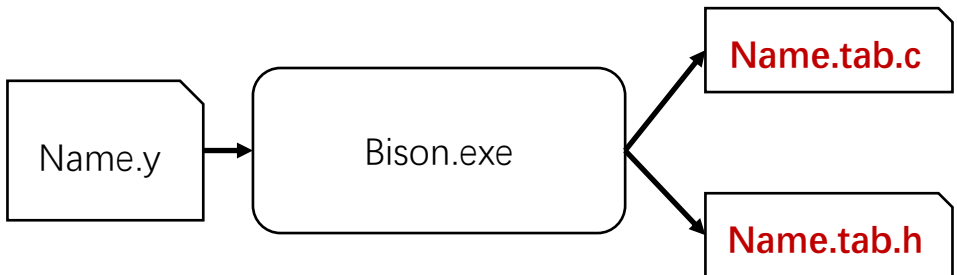
打开Visual Studio 2008 命令行



步骤2：

使用Bison『编译』源程序（Name.y）

- 在Visual Studio 2008 命令行界面，输入以下命令：
- 1、进入Bison安装目录
- > **cd C:\GnuWin32\bin**
- 2、调用bison.exe
- > **bison.exe -d Name.y**



步骤2：

使用Bison『编译』源程序（Name.y）

- 在Visual Studio 2008 命令行界面，输入以下命令：
- 1、进入Bison安装目录
- > **cd C:\GnuWin32\bin**
- 2、调用bison.exe
- > **bison.exe -d Name.y**

Name.tab.h

```
/* Tokens. */
#ifndef YYTOKENTYPE
# define YYTOKENTYPE
    /* Put the tokens into the symbol table, so that GDB and other debuggers
       know about them. */
    enum yytokentype {
        NAME = 258,
        EQ = 259,
        AGE = 260
    };
#endif
```

→ %token **NAME EQ AGE**

```
#if ! defined YYSTYPE && ! defined YYSTYPE_IS_DECLARED
typedef int YYSTYPE;
# define YYSTYPE_IS_TRIVIAL 1
# define YYSTYPE YYSYMBOL_YYSTYPE /* obsolescent; will be withdrawn */
# define YYSTYPE_IS_DECLARED 1
#endif
```

```
extern YYSTYPE yylval;
```

→ %
typedef char* string;
#define **YYSTYPE** string
%}

步骤3：

编写Flex源程序 (Name.l)

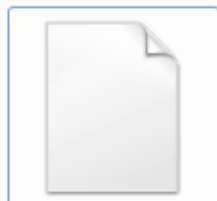
```
%{  
#include "Name.tab.h"  
%}  
  
char      [A-Za-z]  
num       [0-9]  
eq        [=]  
name      {char}+  
age       {num}+  
  
%%  
  
{name}    { yylval = strdup(yytext); return NAME; }  
{eq}      { return EQ; }  
{age}     { yylval = strdup(yytext); return AGE; }  
  
%%  
  
int yywrap()  
{return 1;}
```



flex.exe



flex++.exe



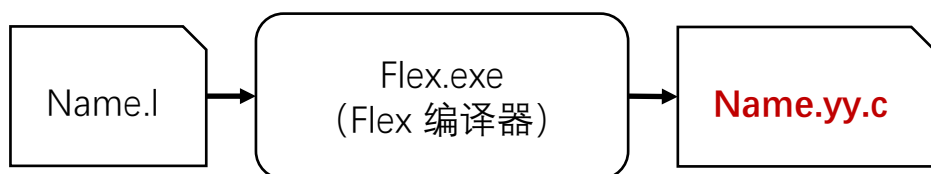
lex1.l

保存到flex安装目录

步骤4：

使用Flex 『编译』 源程序 (Name.l)

- 在Visual Studio 2008 命令行界面，输入以下命令：
- 1、进入flex安装目录
- > **cd C:\GnuWin32\bin**
- 2、调用flex.exe
- > **flex.exe -o"Name.yy.c" Name.l**



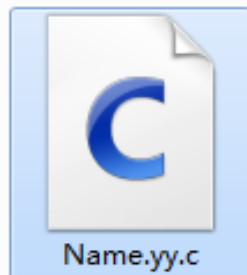
flex.exe



flex++.exe



Name.l



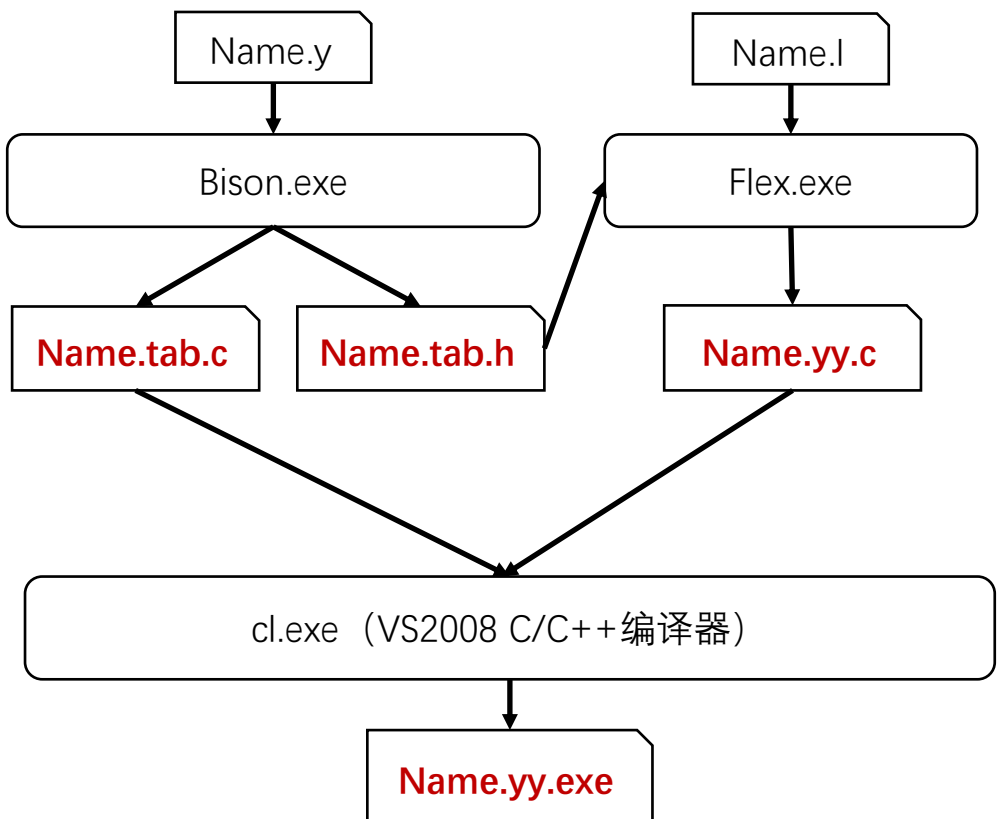
Name.yy.c

步骤5：

使用VS2008『编译』

Name.yy.c、Name.tab.c

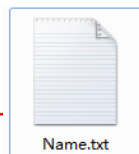
- 在Visual Studio 2008 命令行界面，输入以下命令：
- 1、进入flex安装目录
- > **cd C:\GnuWin32\bin**
- 2、调用vs2008的编译器 cl.exe
- > **cl Name.yy.c Name.tab.c**



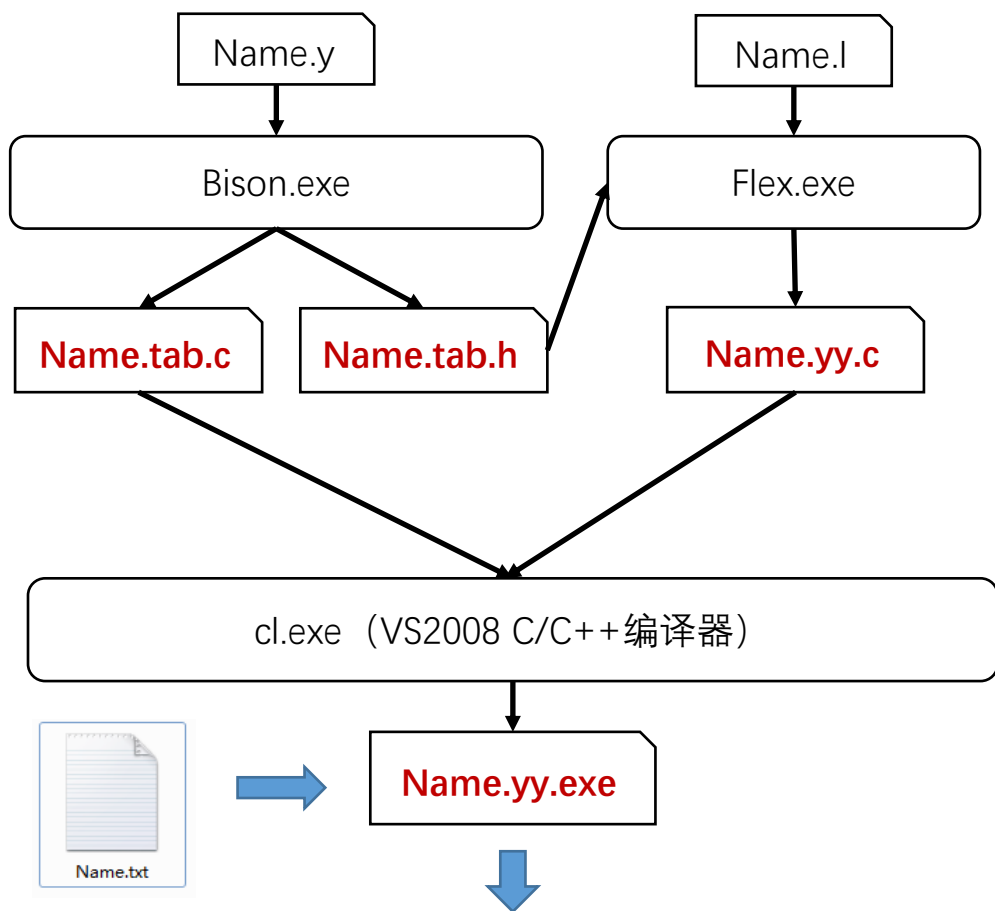
步骤6：

调用 Name.yy.exe 分析 Name.txt

- 在Visual Studio 2008 命令行界面，输入以下命令：
- 1、进入flex安装目录
- > **cd C:\GnuWin32\bin**
- 2、执行以下命令：
- > **Name.yy.exe < Name.txt**



Tom=40
JERRY=30



```
C:\GnuWin32\bin>Name.yy.exe < Name.txt
Tom is 40 years old!!!

JERRY is 30 years old!!!
```

提交成果

- 1、上传Bsion代码——Name.y、Name.l；
- 2、提交Windows和Linux环境下的**Name**结果截图到[雨课堂](#)。
- 3、实验报告
 - 3.1 Windows, Linux两种环境下的配置和使用过程；
 - 3.2 分析Bison源代码（ Name.y ）、分析Flex源代码（ Name.l ）
 - 包括编程步骤、Bsion程序组成、**Bsion的语法规则、语法分析树**;
 - 3.3 结合输出结果，说明FLEX+Bison的协同工作过程。
- 说明：
 - 实验报告参考百度网盘样例《【实验三】陈柏翰 利用 FlexBison 构造编译器.pdf 》
 - 下载《系统软件课程设计报告_模板.docx 》