



公告

昵称: 岁月静好--lyr  
园龄: 1年3个月  
粉丝: 0  
关注: 2  
[+加关注](#)

< 2016年12月 >						
日	一	二	三	四	五	六
27	28	29	30	1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31
1	2	3	4	5	6	7

搜索

找找看

谷歌搜索

常用链接

[我的随笔](#)  
[我的评论](#)  
[我的参与](#)  
[最新评论](#)  
[我的标签](#)

我的标签

[c++基础\(1\)](#)  
[docker\(1\)](#)  
[Java中常见乱码问题\(1\)](#)  
[map遍历方法\(1\)](#)  
[map排序\(1\)](#)  
[操作系统\(1\)](#)  
[存储管理\(1\)](#)  
[数据库\(1\)](#)  
[完整性\(1\)](#)  
[文件上传中表单数据提交\(1\)](#)  
[更多](#)

随笔分类

[c\(3\)](#)  
[java\(5\)](#)  
[安装过程\(3\)](#)  
[操作系统\(1\)](#)  
[数据库\(2\)](#)

随笔档案

[2016年11月 \(5\)](#)  
[2016年8月 \(3\)](#)  
[2016年4月 \(1\)](#)  
[2016年3月 \(1\)](#)  
[2015年8月 \(5\)](#)

阅读排行榜

1. [Java 使用itext生成pdf以及下载\(25\)](#)  
2. [jspsmartupload 文件上传时input数据和文件上传同时提交\(25\)](#)  
3. [Java中常见的几个乱码问题以及解决方法\(21\)](#)  
4. [HasnMap的一种遍历方式: Map.Entry 和 Map.entrySet\(\)\(15\)](#)  
5. [HashMap的两种排序方式\(11\)](#)

## flex 和bison的安装和使用

1.在ubutu上安装 yacc的命令:

```
sudo apt-get install flex bison
```

flex:词法分析器

flex是一个词法分析器。用来将一个.l文件生成一个.c程序文件。即生成一个词法分析器。然后读取输入, 和正则表达式匹配, 再执行相应的动作, 实现了程序的功能。

我们可以发现flex实现在程序外部就可以接受输入的功能。

bison:语法分析器

根据文法把一系列的记号转换成一个语法分析树。

把两者联合使用可以实现对sql语言的解析。

由于调用这些命令的时候会使用很多命令行语句, 即编译过程复杂的时候, 我们可以把这些命令行语句写在一个makefile中。

2.首先介绍使用flex的一个例子:

```
/* 正如 Unix的wc程序 */
%{
int chars = 0;
int words = 0;
int lines = 0;
}%
%%
[a-zA-Z]+ { words++; chars += strlen(yytext); }
\n      { chars++; lines++; }
.        { chars++; }
%%

main(int argc, char **argv)
{
    yylex();
    printf("%8d%8d%8d\n", lines, words, chars);
}
```

```
$ flex fb1-1.1
$ cc lex.yy.c -lfl
$ ./a.out
The boy stood on the burning deck
shelling peanuts by the peck
^D
2 12 63
$
```

第一步: 首先我们写一个.l文件, 文件中程序就是作图。该程序的功能就是计数器, 单词的个数, 字符个数, 以及行数。

程序中: 分为三个部分。第一部分是%{%}中的部分, 用来声明变量。

第二部分是%% %%中的部分。是指定的匹配规则。形式为正则表达式 加上执行动作。其中, 正则表达式用来匹配用户输入的内容, 一旦匹配成功, 则执行对应的动作。

第三部分是c语言程序部分。调用flex提供的词法分析例程yylex()函数。执行, 得到一个c程序。

第二步: 在ubutu命令行终端中写入右图中命令。

该命令的意思是: 通过flex将一个.l文件生成一个lex.yy.c的程序。中间那两行文字是程序的输入, 2,12,63是程序执行的结果。./a.out是将程序输出的结果保存在这个文件中。

总结:

## 3.bison的使用

例1-5: 简单的计算器**bl-5.y**

```

/* 计算器的最简版本 */
%{
#include <stdio.h>
%}

/* declare tokens */
%token NUMBER
%token ADD SUB MUL DIV ABS
%token EOL

%%

calclist: /* 空规则 */
| calclist exp EOL { printf("= %d\n", $2); }
;

exp: factor default $$ = $1
| exp ADD factor { $$ = $1 + $3; }
| exp SUB factor { $$ = $1 - $3; }
;

factor: term default $$ = $1
| factor MUL term { $$ = $1 * $3; }
| factor DIV term { $$ = $1 / $3; }
;

term: NUMBER default $$ = $1
| ABS term { $$ = $2 >= 0? $2 : - $2; }
;
%%
main(int argc, char **argv)

```

```

{
    yyparse();
}
yyerror(char *s)
{
    fprintf(stderr, "error: %s\n", s);
}

```

**bison**程序包含了（不是巧合）与**flex**程序相同的三部分结构：声明部分、规则部分和C代码部分。声明部分包含了会被原样拷贝到目标分析程序开头的C代码，同样也通过**%{和%}**来声明。随后是**%token**记号声明，以便于告诉**bison**在语法分析程序中记号的名称。通常来说，记号总是使用大写字母，虽然**bison**本身并不这么要求。任何没有声明为记号的语法符号必须出现在至少一条规则的左边。（如果一个语法符号既不是记号也没有出现在任何规则的左边，它会像C程序里没有被引用的变量。它并不会破坏任何事情，但是它可能意味着程序员犯了一个错误。）

第二部分包含了通过简单的BNF定义的规则。**bison**使用单一的冒号而不是**::=**，同时由于行间隔并不那么明显，分号被用来表示规则的结束。同样，像**flex**那样，C的动作代码在每条规则之后用花括号括起。

.y文件中的规则定义部分中。包含两部分内容：

第一部分：终结符的声明。在sql语言中就是SELECT FROM WHERE 等这些。

第二部分：定义了语法。即"ADD" 就是表示一个加法。

<http://www.cnblogs.com/me115/archive/2010/10/27/1862180.html> 这篇文章中对文法的讲解比较朴素易懂。

## 4.flex和bison联合编译的过程。

分类: [安装过程](#)

好文要顶

关注我

收藏该文





岁月静好--lyr  
关注 - 2  
粉丝 - 0

+加关注

« 上一篇: [c++ map](#)  
» 下一篇: [bison 编译sql的基本知识](#)

posted @ 2016-11-24 22:03 岁月静好--lyr 阅读(7) 评论(0) 编辑 收藏

[刷新评论](#) [刷新页面](#) [返回顶部](#)

- 注册用户登录后才能发表评论，请 [登录](#) 或 [注册](#)，[访问](#)网站首页。
- 【推荐】50万行VC++源码: 大型组态工控、电力仿真CAD与GIS源码库
- 【推荐】融云发布 App 社交化白皮书 IM 提升活跃超 8 倍
- 【邀请】网易云渠道合作伙伴招商大会，邀您共创未来



- 最新IT新闻:
- 印度全球最便宜智能机告吹 曾拿中国山寨机冒充
  - 外媒：三星迟早会甩掉PC包袱 联想收购并无价值
  - Netflix终于允许用户下载视频离线观看了
  - 网秦第三季度应占净亏损900万美元 同比亏损扩大
  - 旧版安卓出现一款新流氓软件：会主动到谷歌Play购买消费
- » 更多新闻...



- 最新知识库文章:
- 高质量的工程代码为什么难写
  - 循序渐进地代码重构
  - 技术的正宗与野路子
  - 陈皓：什么是工程师文化？
  - 没那么难，谈CSS的设计模式
- » 更多知识库文章...