**中国矿业大学计算机学院**

**2018 级本科生课程设计报告**

课程名称 系统软件开发实践

实验名称 Flex理论与练习实验1

报告时间 2021年3月3日

学生姓名 丁旭行

学 号 14184501

专 业 计算机科学与技术

任课教师 徐东红

**目 录**

[1 实验目的 1](#_Toc65769977)

[2 编译器开发工具Flex 1](#_Toc65769978)

[2.1 Flex简介 1](#_Toc65769979)

[2.2 Flex文件结构 1](#_Toc65769980)

[2.2.1 定义段 2](#_Toc65769981)

[2.2.2 规则段 2](#_Toc65769982)

[2.2.3 用户代码段 3](#_Toc65769983)

[3 Flex代码分析 3](#_Toc65769984)

[3.1 lex1.l源代码分析 3](#_Toc65769985)

[3.2 lex2.l源代码分析 3](#_Toc65769986)

[4 Windows环境下Flex的配置与使用 4](#_Toc65769987)

[4.1 下载安装Flex 4](#_Toc65769988)

[4.2 lex1.l源代码编写与执行 5](#_Toc65769989)

[4.2.1 lex1.l源代码编写与保存 5](#_Toc65769990)

[4.2.2 Visual Studio 2019命令行执行命令 5](#_Toc65769991)

[4.2.3 C语言代码编写与测试 7](#_Toc65769992)

[4.2.4 分析程序输出结果 7](#_Toc65769993)

[4.3 lex2.l源代码编写与执行 8](#_Toc65769994)

[4.3.1 lex2.l源代码编写测试 8](#_Toc65769995)

[4.3.2 分析程序输出结果 9](#_Toc65769996)

[5 Linux-Ubuntu环境下Flex的配置与使用 9](#_Toc65769997)

[5.1 Linux虚拟机的安装 9](#_Toc65769998)

[5.2 下载安装Flex 11](#_Toc65769999)

[5.3 lex1.l源代码编写与执行 12](#_Toc65770000)

[5.3.1 lex1.l源代码编写与保存 12](#_Toc65770001)

[5.3.2 C语言代码编写与保存 12](#_Toc65770002)

[5.3.3 lex代码运行测试 13](#_Toc65770003)

[5.3.4 分析程序输出结果 14](#_Toc65770004)

[5.4 lex2.l源代码编写与执行 14](#_Toc65770005)

[5.4.1 lex2.l源代码编写测试 14](#_Toc65770006)

[5.4.2 分析程序输出结果 16](#_Toc65770007)

[6 实验体会 16](#_Toc65770008)

# 1 实验目的

利用Flex 设计一个扫描器程序，用于计算一个文件中的字符数、单词数和行数。

# 2 编译器开发工具Flex

## 2.1 Flex简介

Flex是一个生成扫描器(scantier)的工具，生成的扫描器能够识别文本中的词 法模式(lexical pattern)。Flex接受文本格式的Flex文件(扩展名可以为.l、.flx、.lex 或者.flex)作为输入，生成一个C语言的源文件：lex.yy.c，其中定义了一个函数 yylex()，该函数就是扫描器。它根据Flex文件中定义的模式(pattern)对输入的文 本串进行分析，然后执行对应的动作(Action)，该模式和对应的动作叫做规则。 此外，lex.yy.c可以编译后执行，也可以被其他源文件中的函数调用。

## 2.2 Flex文件结构

Flex文件是词法规范定义文件，给出了单词的构成规则，以及在某个规则下应该执行的动作。它由定义段、规则段和用户代码段三个部分构成，中间用%%隔开。Flex的文件结构如图1所示。

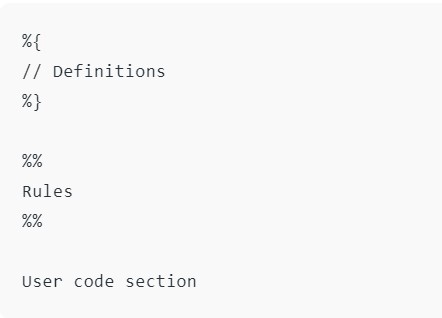


图1 Flex文件结构

### 2.2.1 定义段

定义段包括一些C语言代码、模式宏定义等。段中的定义不能有缩进，除模式宏定义外，文本要用%{和%}括起来。它们会原封不动地拷贝到输出文件lex.yy.c中。%{、%}也必须在单独的行，并且不能缩进。另外，不缩进的注释(用/\*和\*/括起来)也拷贝到lex.yy.c中。

简单名字的定义格式为：name definition，其中，name(名字)是一个以字母 或\_开始的单词，后面可以是零个或多个字母、数字、\_或-构成正则表达式(regular expression)。definition(定义)从name后第一个非空白字符开始，一直到行末。对name的definition之后可以通过{name}引用。例如：‘Digit [0-9]'和'ID [a-z][a-z0-9]\*',以上定义了两个 name为Digit和ID。Digit定义为一个正则表达式匹配一个字符，ID定义为一个正则表达式，第一个字符为a-z之间的字符，后面跟着零个或多个a-z之间的字符或0-9之间的数字。如果后面有个引用{Digit}'.'{Digit}\*，等价于([0-9])'.'([0-9])\*。

### 2.2.2 规则段

规则段包括一系列如下所示的规则：

⑴ Pattern(模式)：pattern使用一个正则表达式的扩展集表示。

⑵ 输入的文本如何匹配模式：当生成的扫描器运行时，它分析输入的文本，寻找匹配规则段中定义的pattern。如果有多个pattern匹配，选择匹配最长字符 串的pattern；如果有两个或多个相同长度的匹配，选择flex文件中最先列出的 pattern。输入文本中能够匹配某个pattern的字符串叫做一个token。一旦匹配确定，全局字符指针yytext会指向这个token，可以通过该指针引用它，这个token的长度也可通过全局整型变量yyleng读取。同时，该pattern对应的action(动作)会执行，然后，剩余的输入文本会继续被分析匹配。如果没有匹配的pattern，会执行缺省的规则：剩余输入文本中的下一个字符被认为已经匹配，然后拷贝到标准输出中。因此，如果没有定义规则，输入文本会原样不动拷贝到标准输出中。

⑶ Action(动作)：action跟在pattern的后面，可以是任意的C语言语句。一行中，pattern以第一个非转义的空白符作为结束标识，剩余部分就是action了。如果某个pattern后的action为空，匹配的token直接舍弃；如果action中包括{,则必有一个对应的}，中间全部都是pattern对应的action，即模式n {动作n}。

### 2.2.3 用户代码段

该段中的代码都被原样拷贝到lex.yy.c中。其中可以定义一些辅助函数或代码，供扫描器yylex()调用，或者调用扫描器(一般来说就是main()主函数了)。这一部分是可选的。如果没有的话，Flex文件中第二段的%%是可以省略的。

# 3 Flex代码分析

## 3.1 lex1.l源代码分析

lex1.l源代码及分析如图所示：

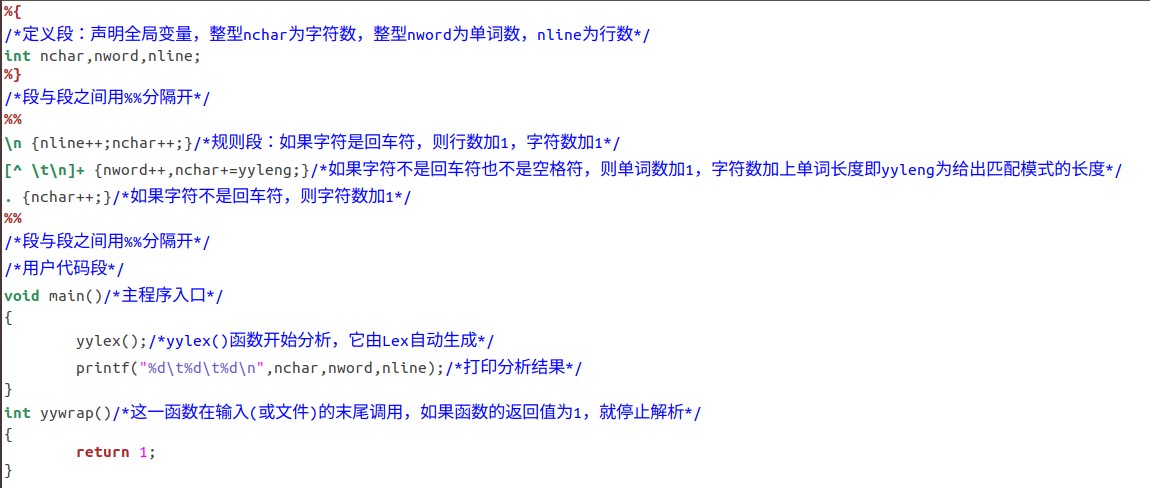


图2 lex1.l代码及分析

## 3.2 lex2.l源代码分析

Lex2.l源代码及分析如图所示：

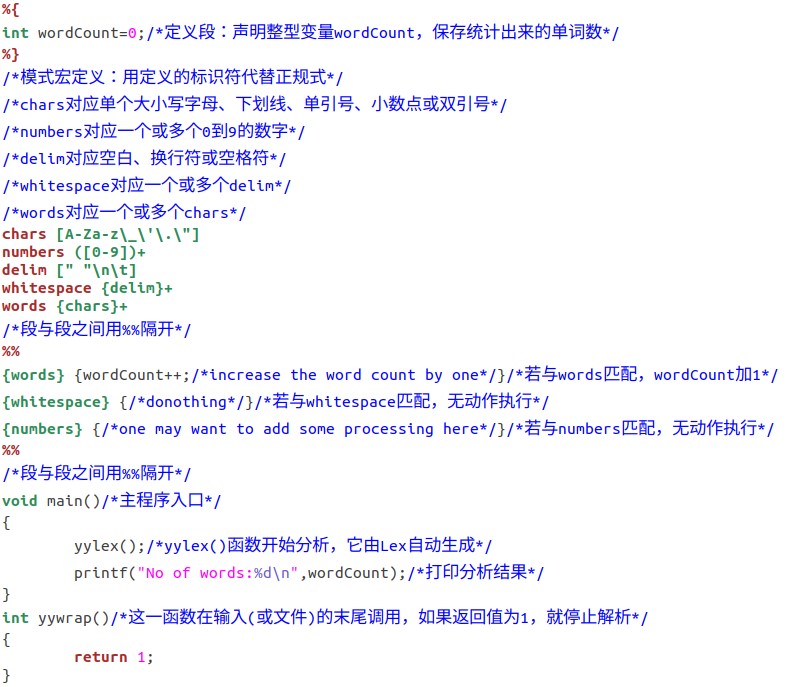


图3 lex2.l代码及分析

# 4 Windows环境下Flex的配置与使用

## 4.1 下载安装Flex

在Windows环境下打开Flex安装包(老师提供的)，按照指示步骤进行安装。

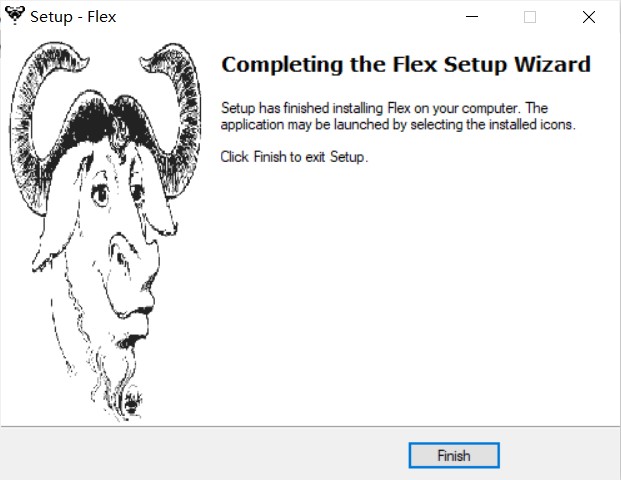
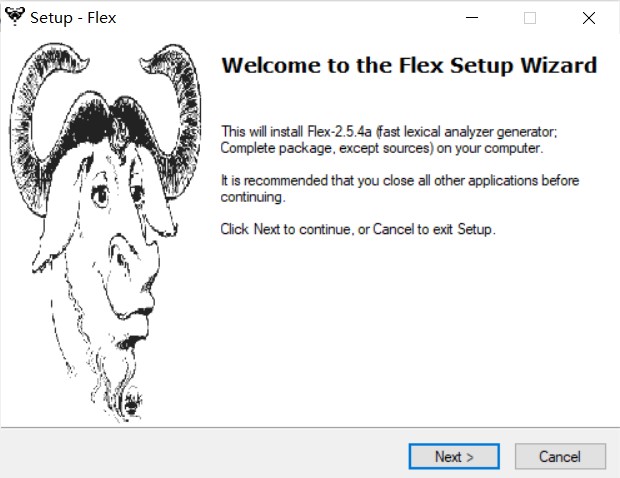


图4 Flex安装1 图5 Flex安装2

## 4.2 lex1.l源代码编写与执行

### 4.2.1 lex1.l源代码编写与保存

新建文本文档编写lex1.l源代码，编写完后，将其保存到Flex的安装目录下。

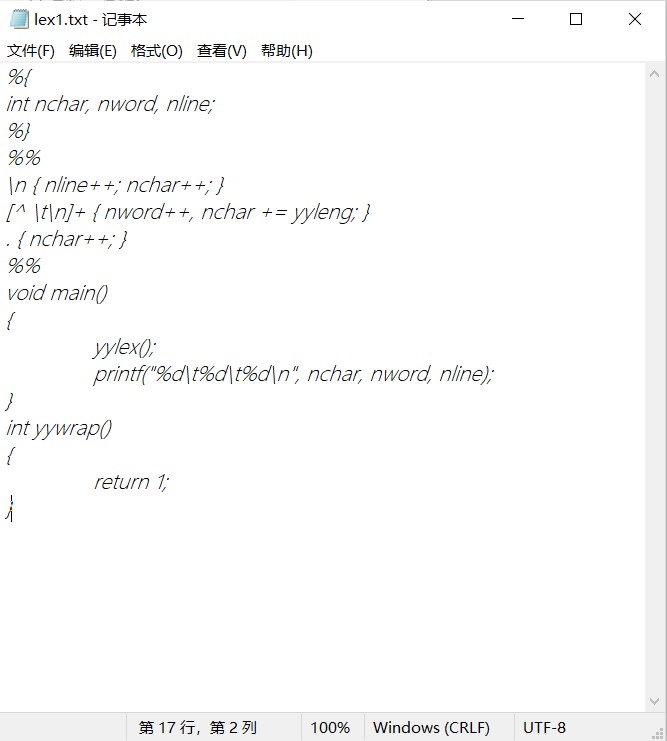


图6 lex1.l源代码

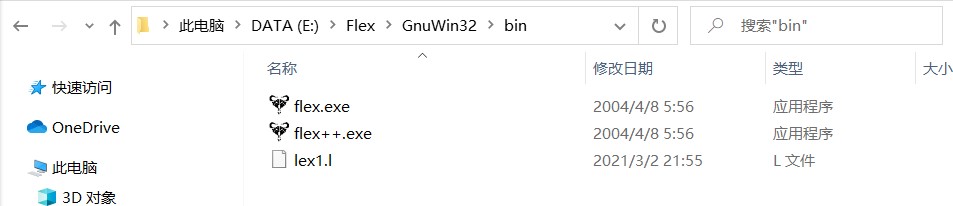


图7 lex1.l保存目录

### 4.2.2 Visual Studio 2019命令行执行命令

打开Developer Command Prompt for VS 2019命令窗口，进入Flex安装目录下，输入flex.exe -o “lex1.yy.c” lex1.l(调用flex.exe)生成lex1.yy.c文件。

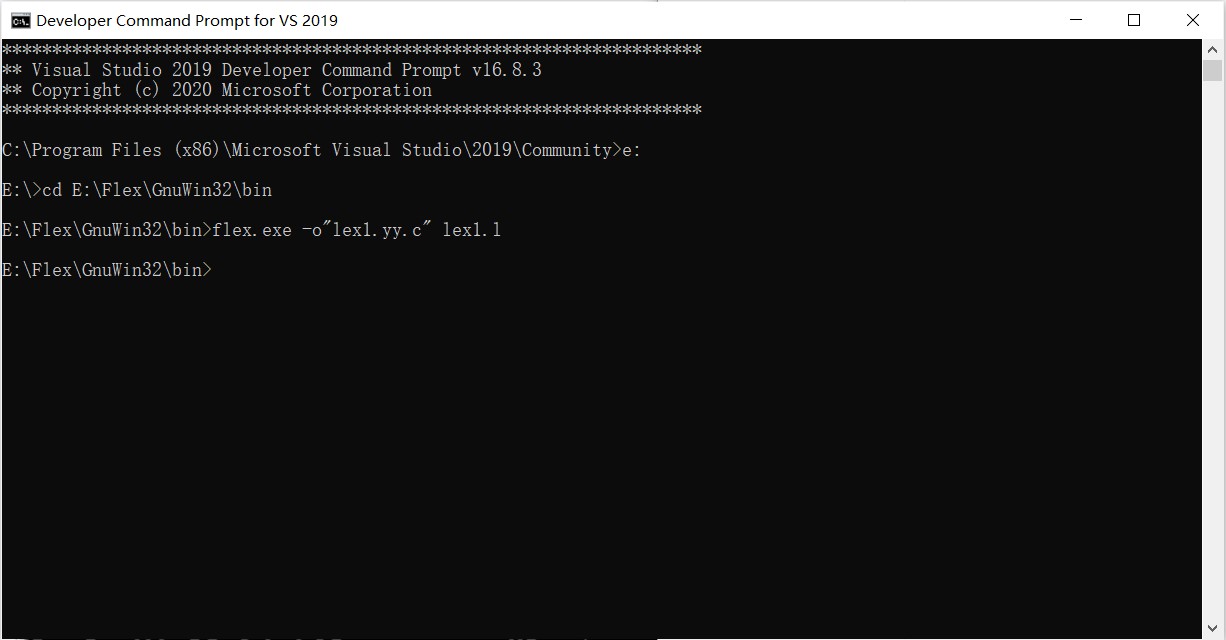


图8 生成lex1.yy.c输入的命令

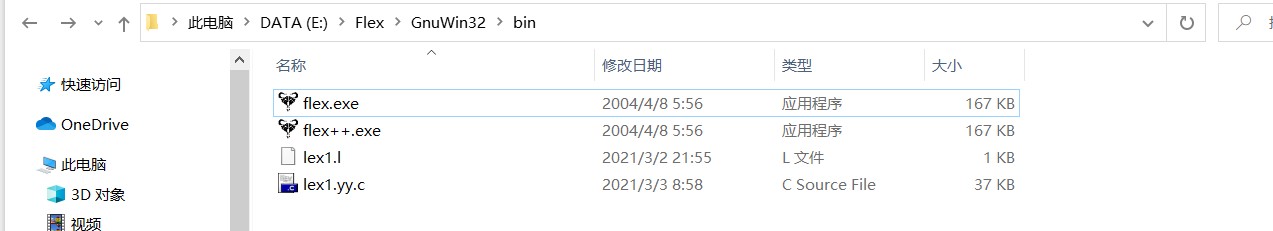


图9 生成lex1.yy.c文件所在目录

在命令行中输入cl lex1.yy.c来调用Visual Studio 2019编译器cl.exe，生成lex1.yy.exe文件。

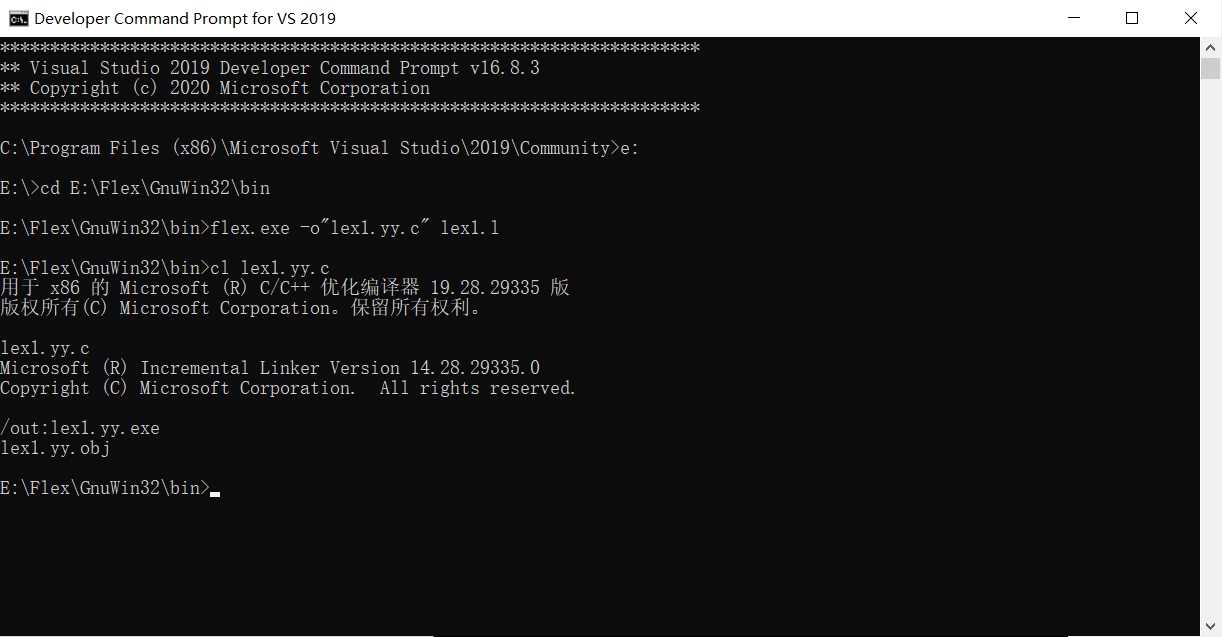


图10 生成lex1.yy.exe输入的命令

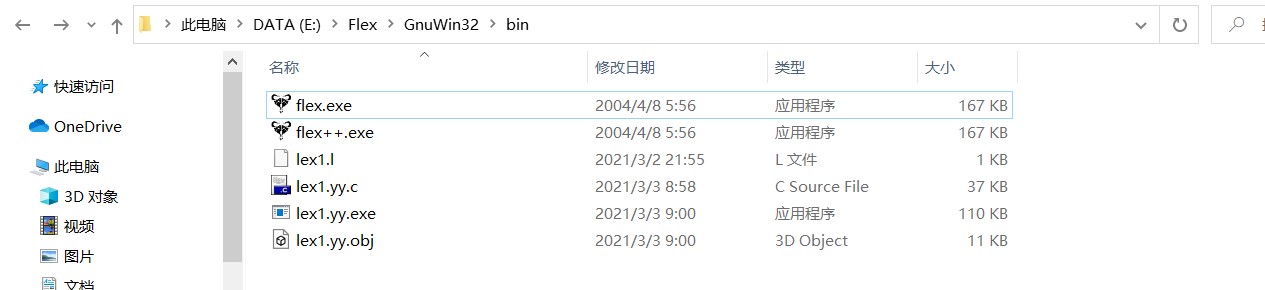


图11 生成lex1.yy.exe文件所在目录

### 4.2.3 C语言代码编写与测试

在Flex安装目录下新建cpp文件，编写测试代码1-1.cpp。在命令行中输入lex1.yy.exe < 1-1.cpp调用lex1.yy.exe对1-1.cpp文件进行词法分析并显示分析结果。

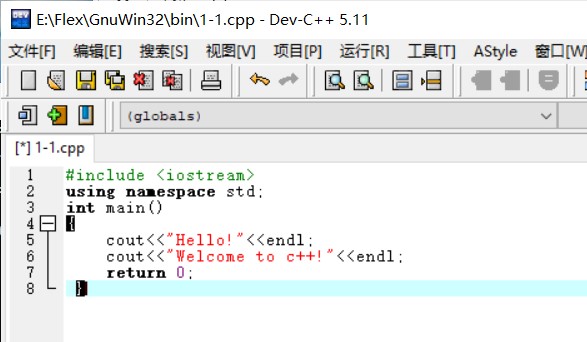


图12 C语言代码文件1-1.cpp

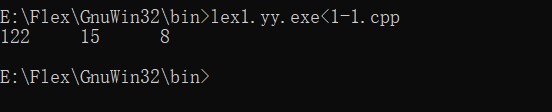


图13 调用lex1.yy.exe分析结果

### 4.2.4 分析程序输出结果

由分析结果知，1-1.cpp文件中共有122个字符、15个单词、8行。根据lex1.l源代码模式规则分析1-1.cpp文件，可知识别出的单词有#include、<iostream>、using、namespace、std;、int、mian()、{、cout<<”Hello!”<<endl;、cout<<”Welcome、to、c++!”<<endl;、return、0;和}共计15个，加上识别的空格符和换行符共122个字符，文件内容共8行，至此说明lex1.l词法分析成功。但是会发现识别的单词中包含了一些标点符号或者特殊字符而并非全是英文字母，而且有的单词由于符号的连接组成了一长串字符，这是因为lex1.l源代码中对单词的识别模式仅仅排除了换行符和空格符，而不是只识别英文字母。

## 4.3 lex2.l源代码编写与执行

### 4.3.1 lex2.l源代码编写测试

由于编写执行lex2.l的过程与lex1.l的过程一致，在此就不再赘述。其过程与分析结果截图如下所示：

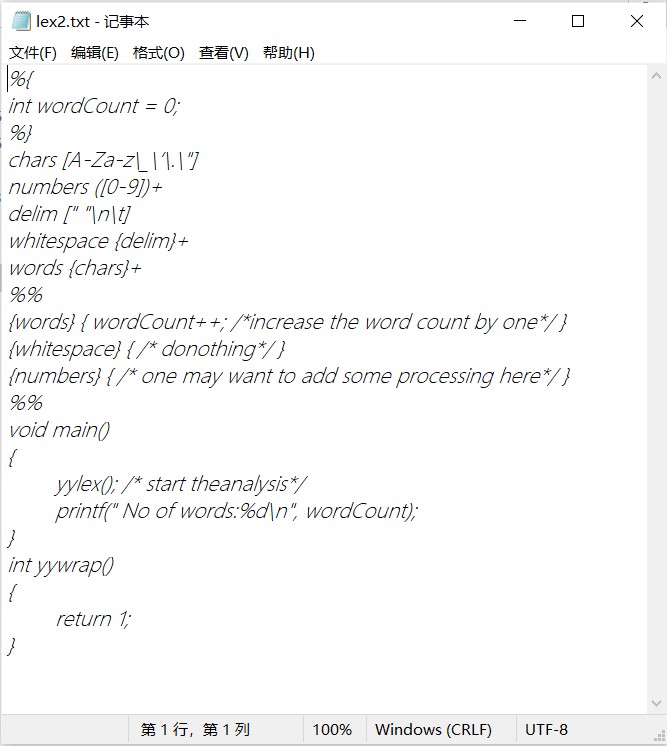


图14 lex2.l源代码

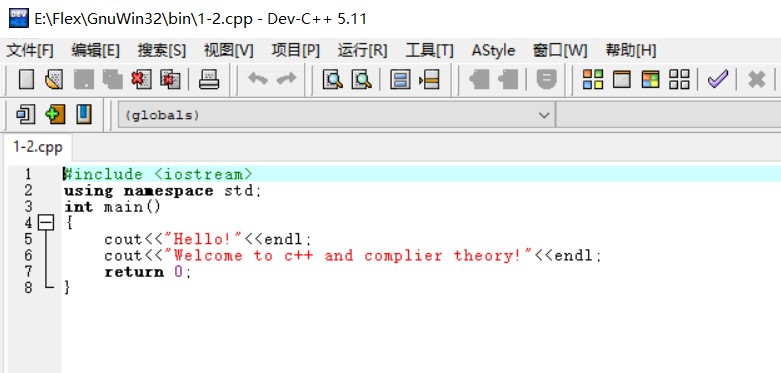


图15 C语言代码文件1-2.cpp

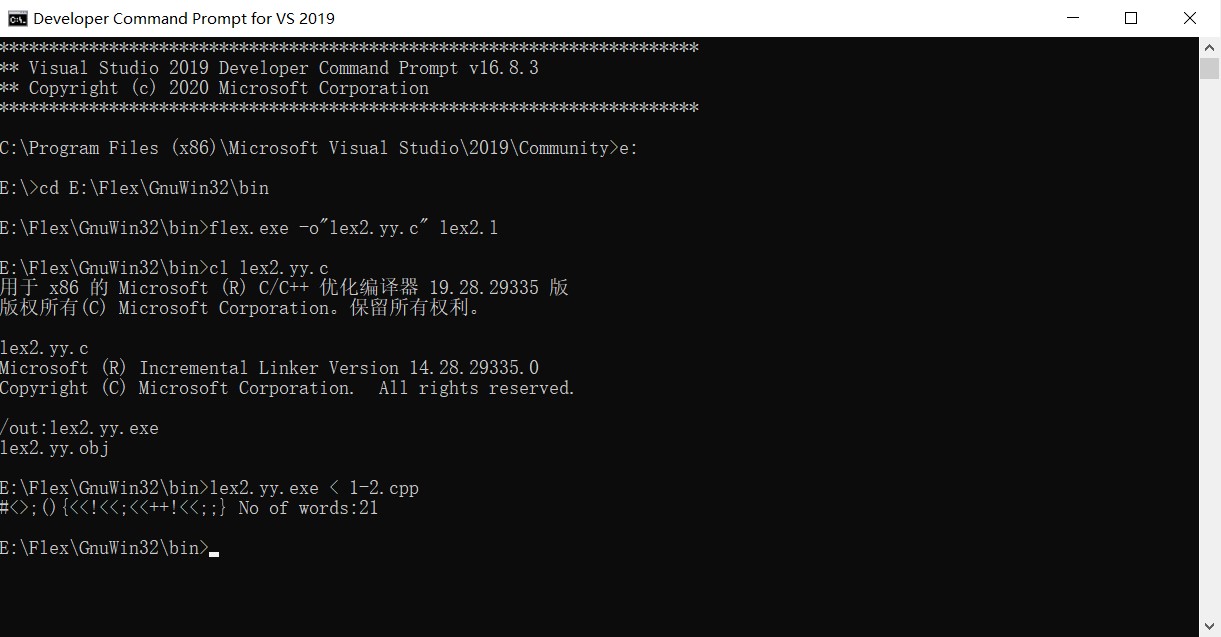


图16 调用lex2.yy.exe分析结果

### 4.3.2 分析程序输出结果

由分析结果知，1-2.cpp文件中共有21个单词。根据lex2.l源代码模式规则分析1-2.cpp文件，可知识别出的单词有include、iostream、using、namespace、std、int、main、cout、”Hello、”、endl、cout、”Welcome、to、c、and、complier、theory、”、endl和return共计21个单词，说明lex2.l词法分析成功。但会发现识别的单词出现了和lex1.l分析结果一样的问题，这是因为lex2.l源代码对于单词的识别虽然排除了一些特殊字符但仍旧包含单引号和双引号。

# 5 Linux-Ubuntu环境下Flex的配置与使用

## 5.1 Linux虚拟机的安装

下载VMware安装包，按照步骤逐步完成安装。

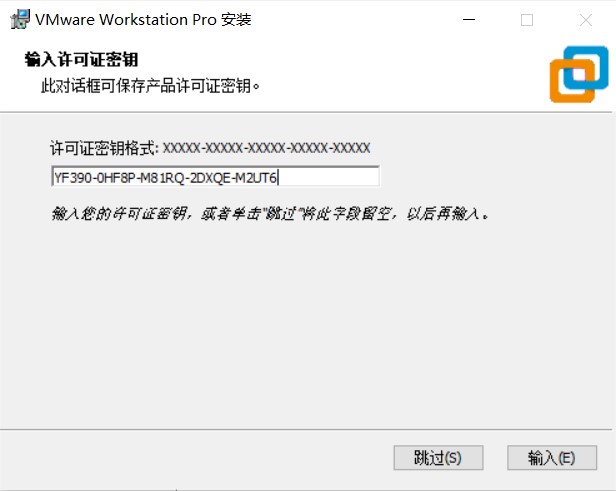
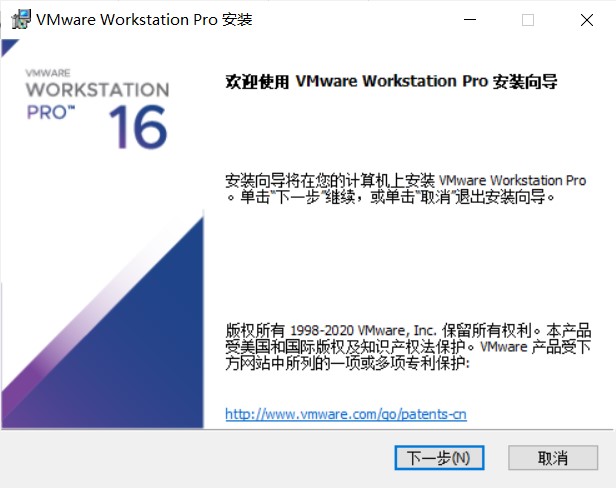


图17 VMware安装 图18 输入许可证密钥

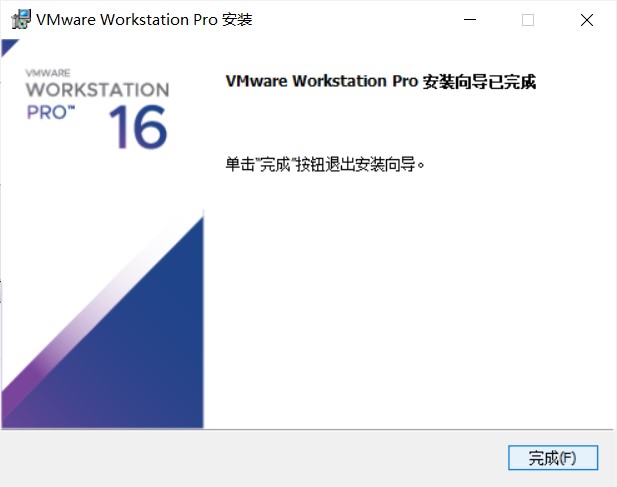


图19 VMware安装完成

打开VMware创建新的Linux虚拟机。



图20 创建Linux虚拟机

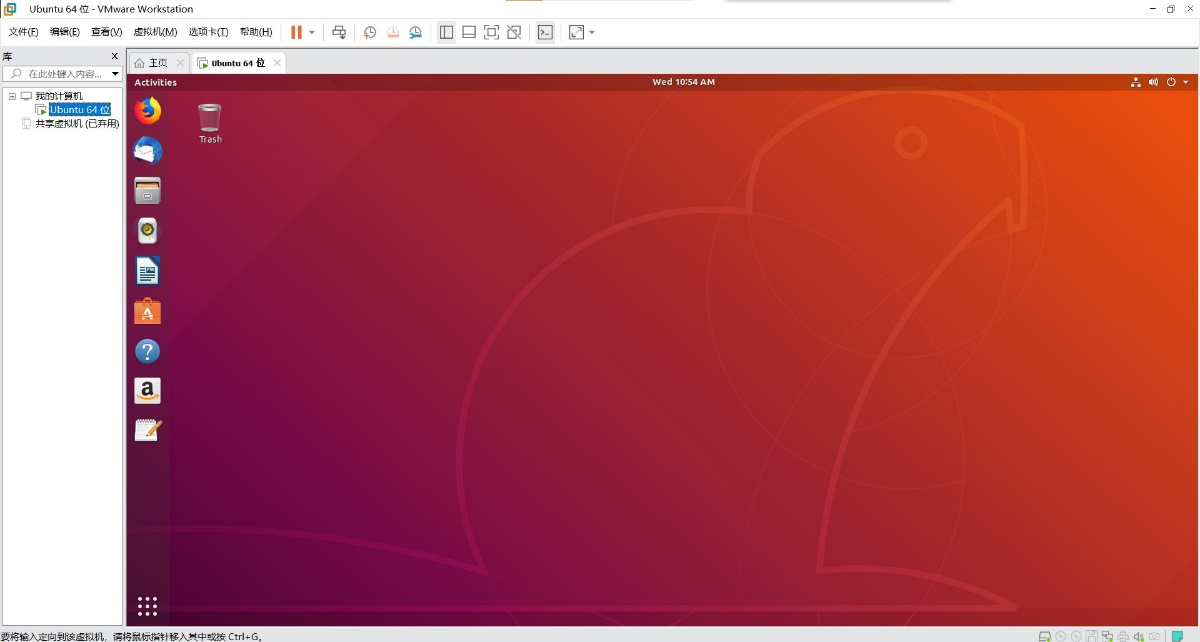


图21 Linux虚拟机创建完成

## 5.2 下载安装Flex

在Linux环境下安装Flex，在命令窗口输入sudo apt-get install flex，进行Flex的安装。

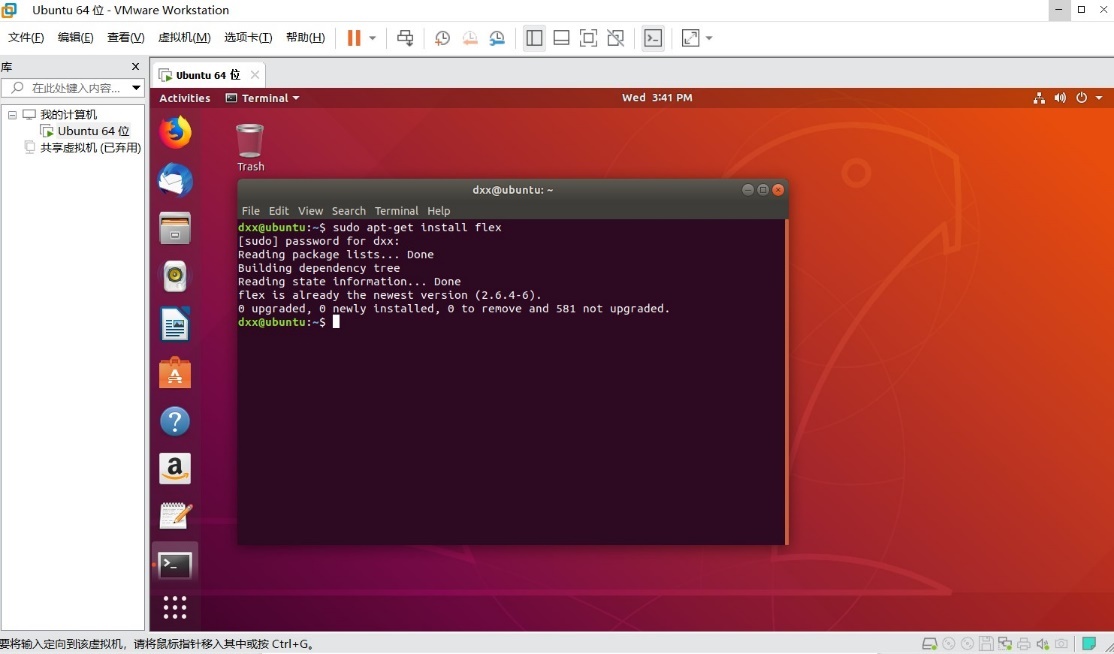


图22 Linux下安装Flex

## 5.3 lex1.l源代码编写与执行

### 5.3.1 lex1.l源代码编写与保存

打开文本编辑器，输入lex1.l源代码并保存到桌面。

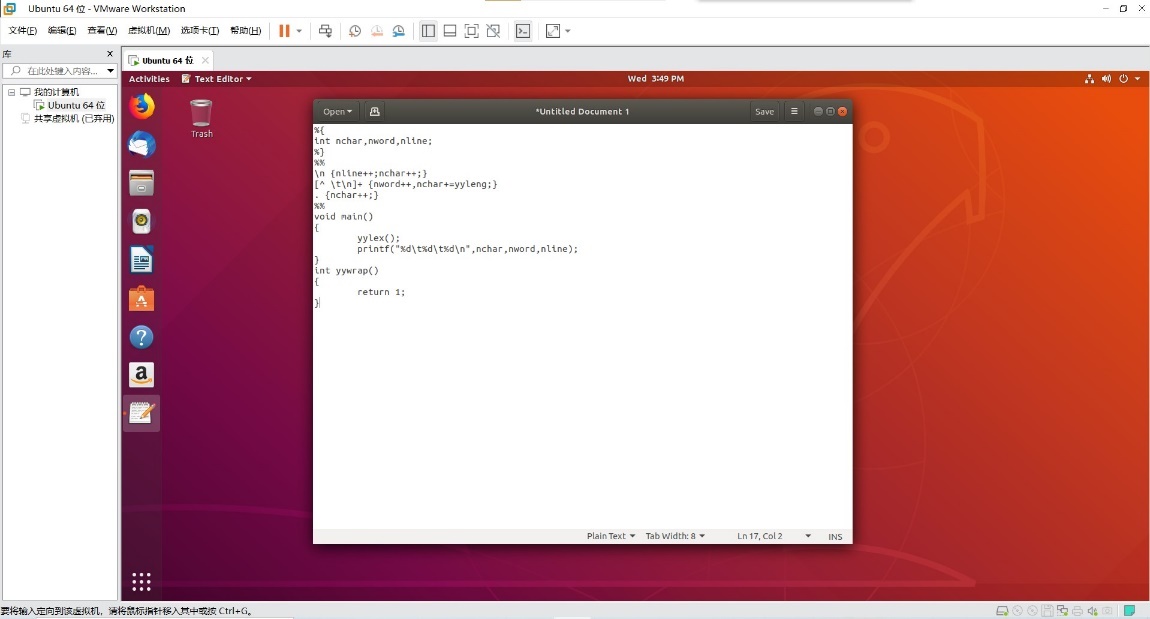


图23 编写lex1.l文件

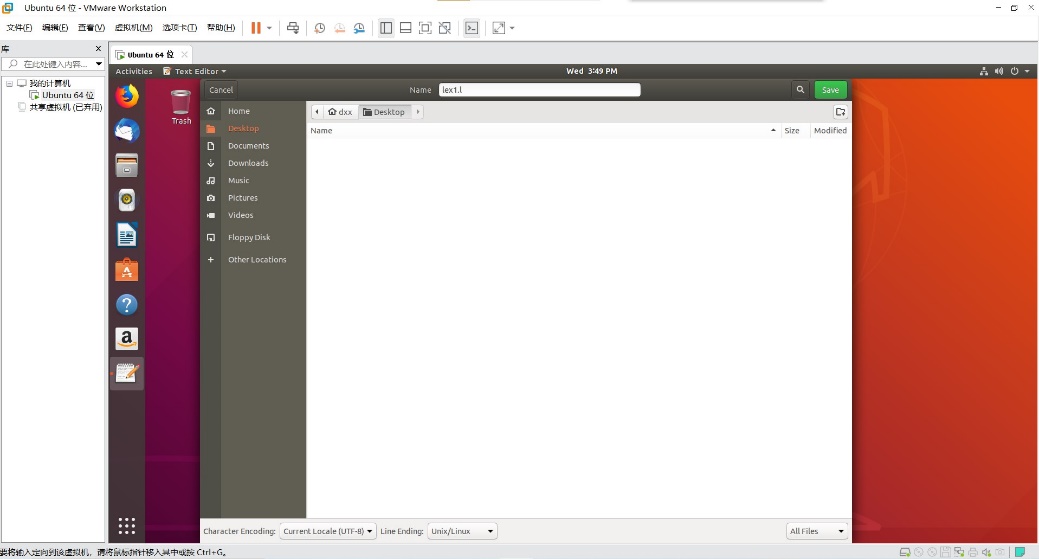


图24 保存lex1.l文件至桌面

### 5.3.2 C语言代码编写与保存

打开文本编辑器，输入C语言代码，将其命名1-1.txt保存至桌面。

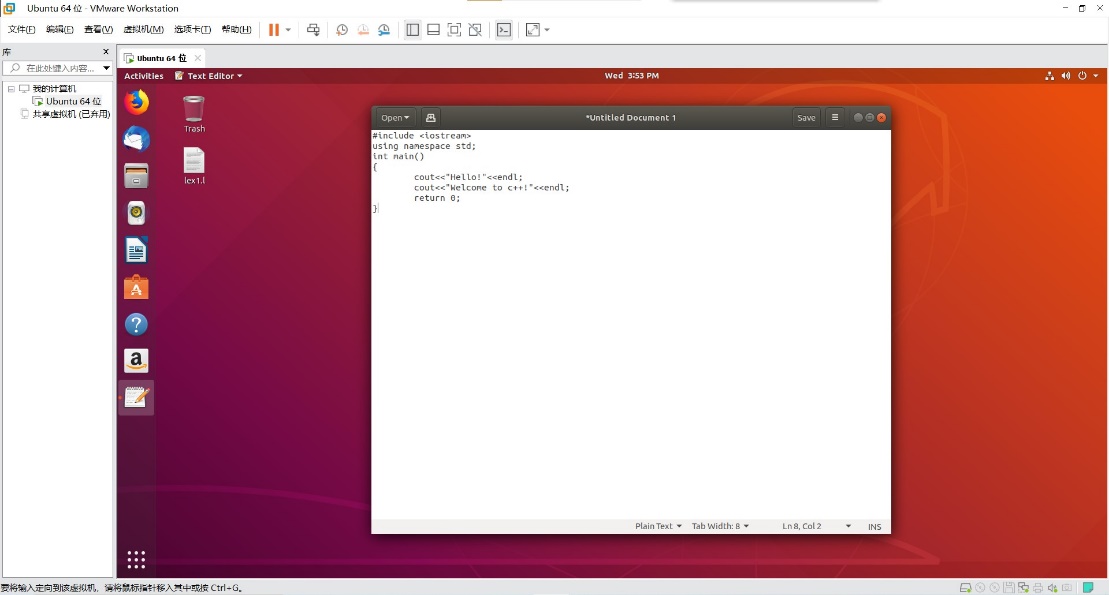


图25 编写C语言代码

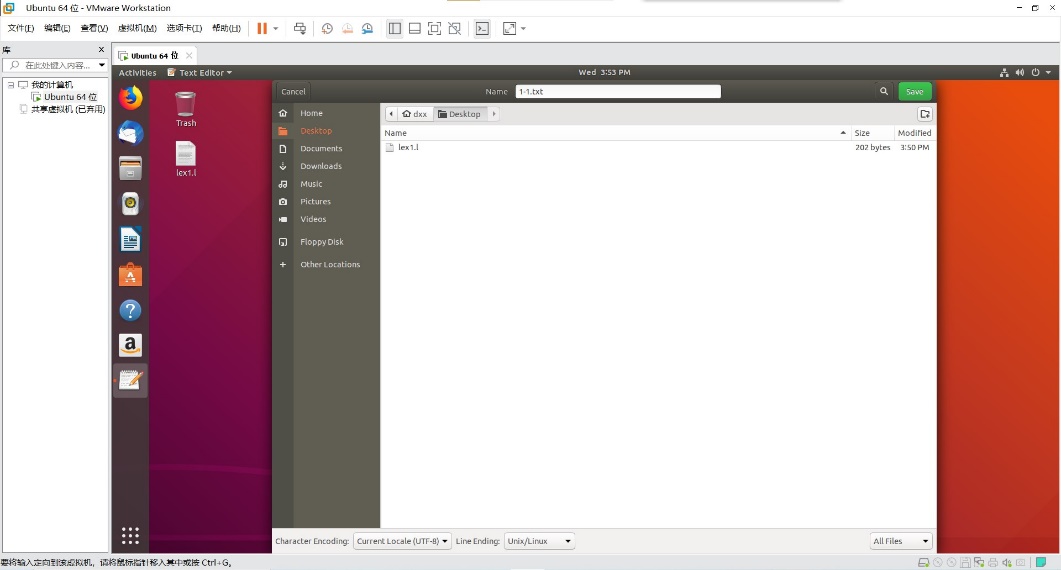


图26 保存1-1.txt文件至桌面

### 5.3.3 lex代码运行测试

打开命令窗口，进入桌面目录，依次输入以下命令：flex lex1.l、cc -o parser lex.yy.c、./parser < 1-1.txt，C语言代码将被解析并输出结果。

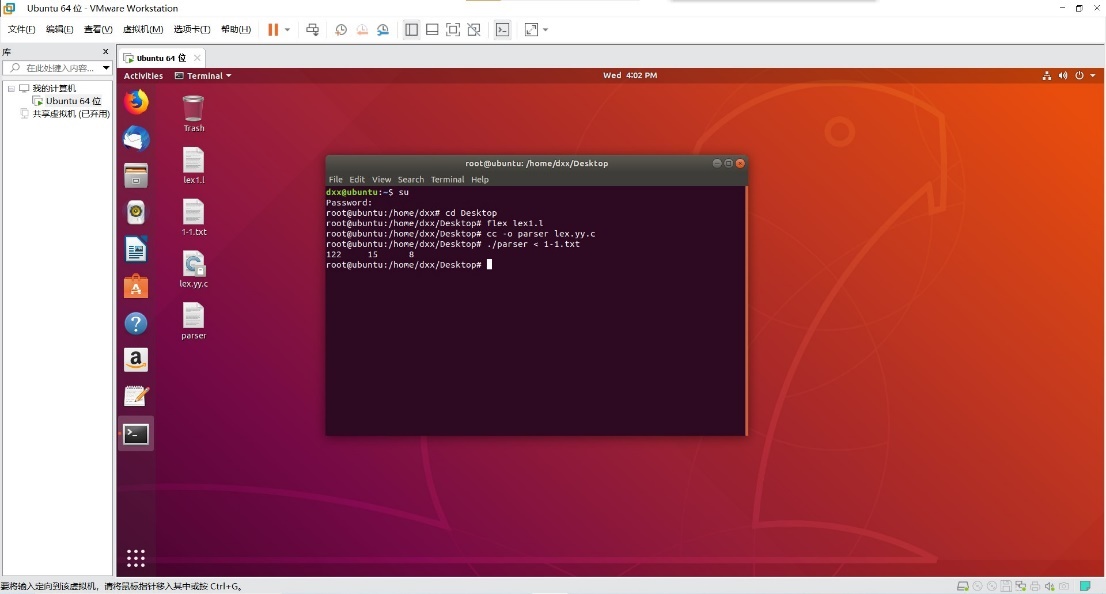


图27 命令输入与结果输出

### 5.3.4 分析程序输出结果

结果分析同Windows环境下的lex1.l执行结果分析一致，在此不再赘述。

## 5.4 lex2.l源代码编写与执行

### 5.4.1 lex2.l源代码编写测试

由于编写执行lex2.l的过程与lex1.l的过程一致，在此就不再赘述。其过程与分析结果截图如下所示：

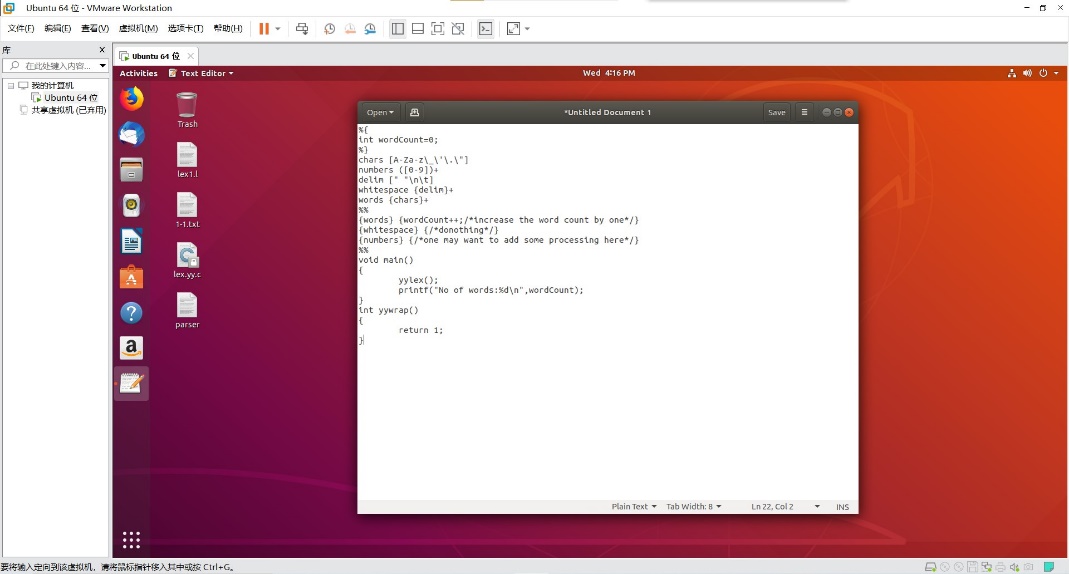


图28 编写lex2.l文件

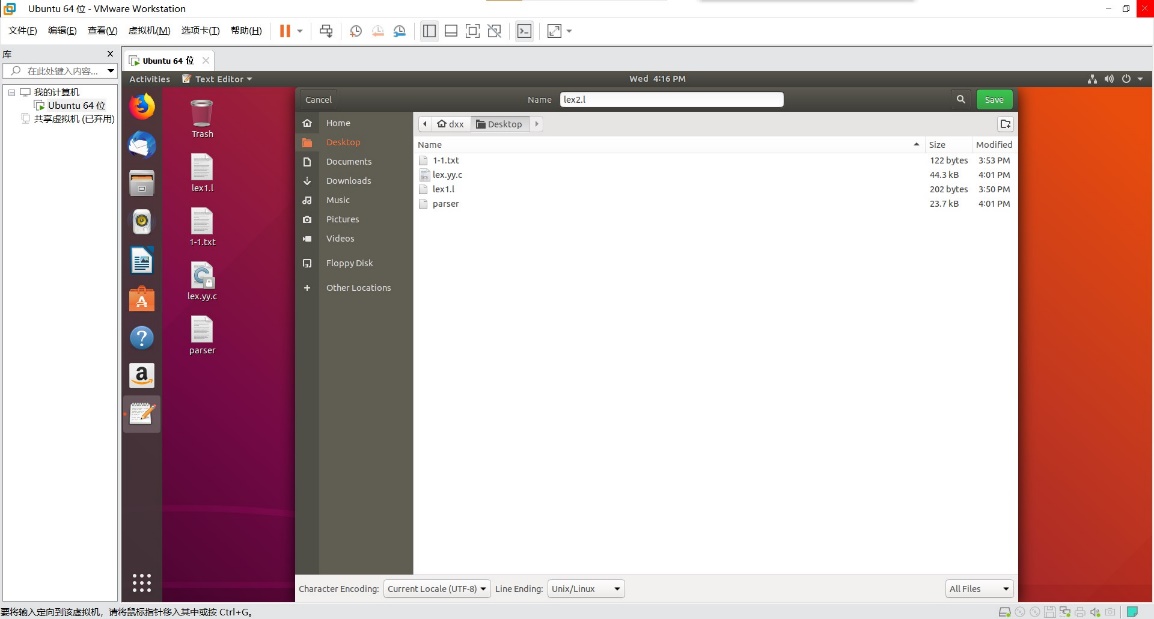


图29 保存lex2.l文件至桌面

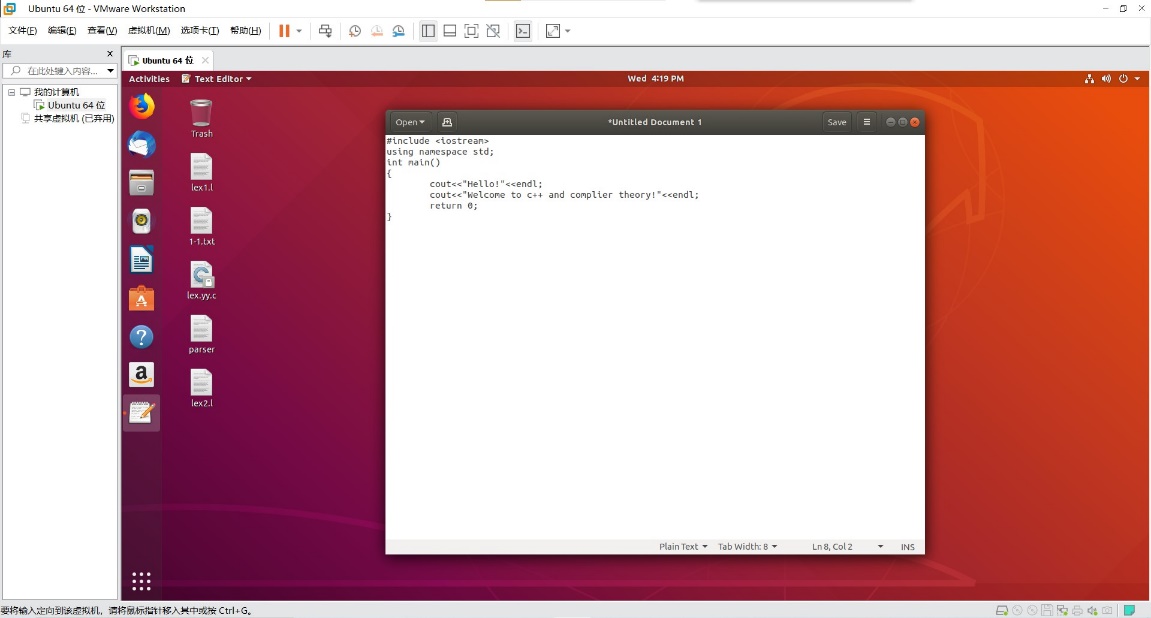


图30 编写C语言代码

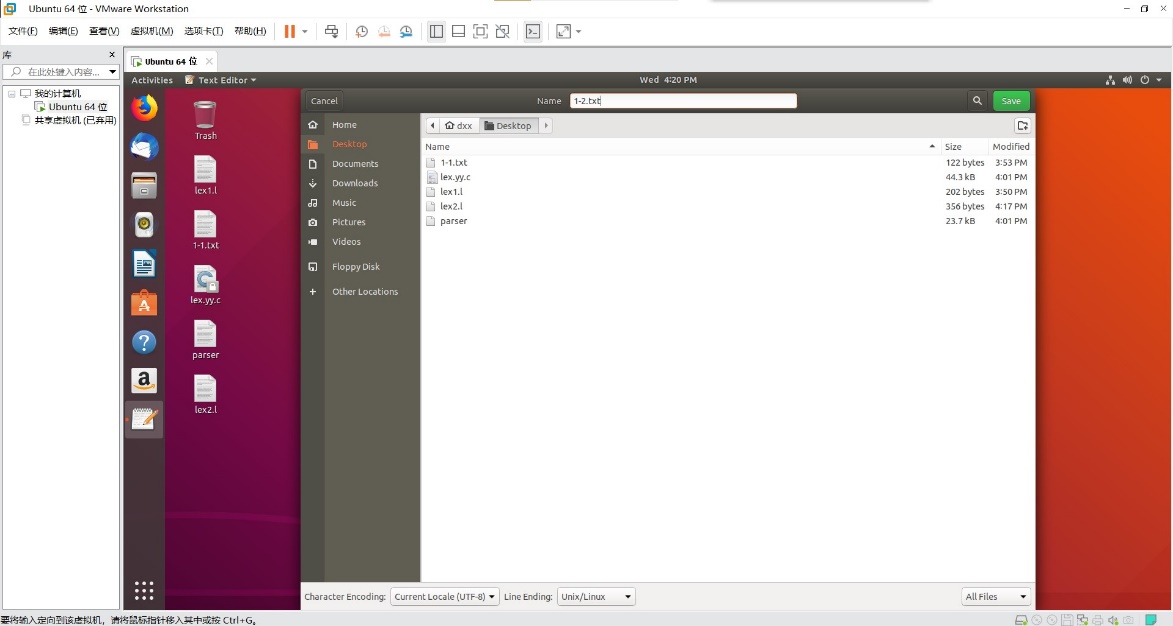


图31 保存1-2.txt文件至桌面

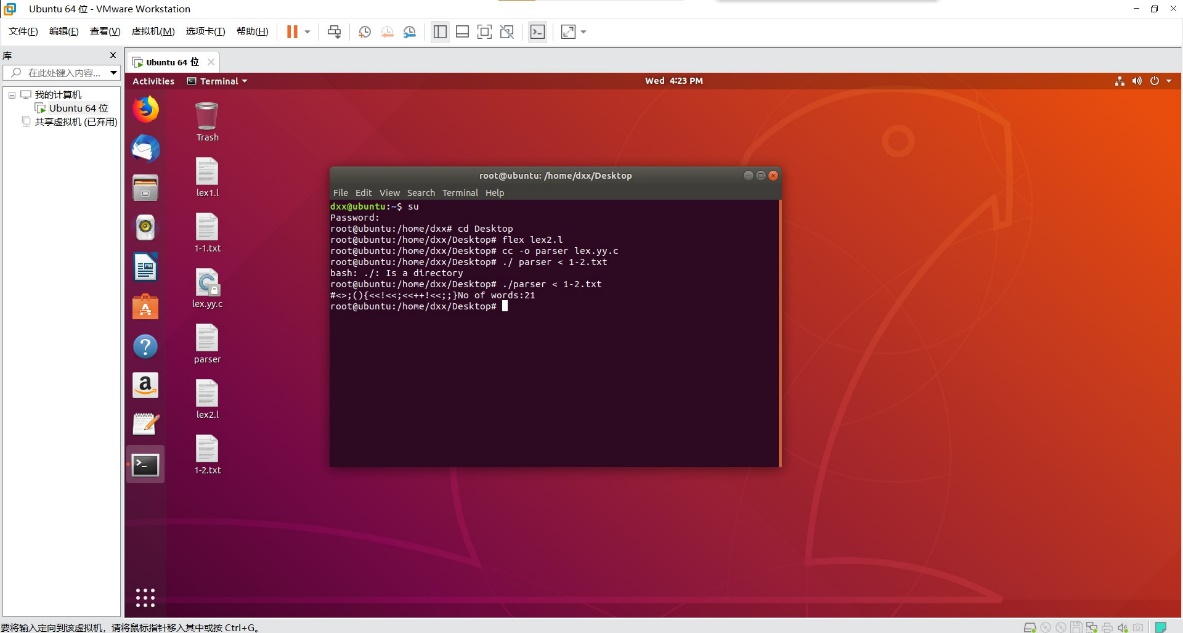


图32 命令输入与结果输出

### 5.4.2 分析程序输出结果

结果分析同Windows环境下的lex2.l执行结果分析一致，在此不再赘述。

# 6 实验体会

通过本次实验，我熟练掌握了Lex的基本语法，对Flex也有了基本的了解。进而学习了利用编译器开发工具Flex分别在Windows和 Linux-Ubuntu环境下对文件进行词法分析的方法，为今后的综合实验打下了基础。当然，在本次实验中也遇到了一些问题，比如：VMware15平台与Win10系统不兼容导致Linux虚拟机无法正常开机，最后只能更换VMware16平台进行虚拟机安装；命令操作不熟悉，Windows下cd命令无法从C盘直接进入E盘，Linux下su命令不能切换root，最后通过查询相关资料得知：Windows下可通过“盘符:”进入对应目录，进而进入Flex安装目录，Linux下需要先输入sudo passwd root命令过一次，之后就可以使用su命令再输入密码切换到root。