



华南师范大学

本科学生实验（实践）报告

院 系：计 算 机 学 院

实验课程：编译原理

实验项目：C++单词拼装器

指导老师：黄煜廉

开课时间：2023 ~ 2024 年度第 1 学期

专 业：计算机科学与技术

班 级：计科 1 班

学 生：李达良

学 号：20203231004

华南师范大学教务处

华南师范大学实验报告

学生姓名 李达良 学 号 20203231004
专 业 计算机科学与技术 年级、班级 2021 级 1 班
课程名称 编译原理 实验项目 C++单词拼装器
实验时间 2023 年 9 月 23 日
实验指导老师 黄煜廉 实验评分 _____

一、实验题目

C++单词拼装器

二、实验内容

1. 把 C++源代码中的各类单词（记号）进行拼装分类。

C++语言包含了几种类型的单词（记号）：标识符，关键字，数（包括整数、浮点数），字符串、注释、特殊符号（分界符）和运算符等【详细的单词类别及拼装规则见另外的文件说明】。

2. 要求应用程序应为 Windows 界面。

3. 打开一个 C++源文件，列出所有可以拼装的单词（记号）。

4. 应该书写完善的软件设计文档。

三、实验目的

本实验旨在通过开发一个具有 Windows 界面的应用程序，实现对 C++源代码中各类单词（记号）进行拼装分类的功能，并列出所有可以拼装的单词。具体实验目的包括：

实现单词分类：通过解析 C++源代码，将其中的各类单词（记号）进行分类，包括标识符、关键字、数（包括整数、浮点数）、字符串、注释、特殊符号（分界符）和运算符等。这有助于深入理解 C++源代码的结构和语法。

开发 Windows 界面应用程序：要求设计并开发一个 Windows 界面应用程序，以提供用户友好的交互方式，使用户能够轻松加载 C++源文件并查看分类后的单词列表。这将提高实验的实用性和用户体验。

列出可拼装的单词：实现一个功能，可以打开 C++源文件，分析其中的文本内容，并列出所有可以拼装的单词（记号）。这将有助于学生更好地理解编译过程中的词法分析阶段。

编写完善的软件设计文档：在实验过程中，要求撰写详尽的软件设计文档，其中包括系统架构、算法设计、用户界面设计、数据结构说明以及关键代码片段的解释。这将有助于培养学生的软件工程和文档编写能力。

通过完成这些目标，本实验旨在帮助学生深入理解编译原理中词法分析的基本概念和技术，同时提高他们的编程和文档编写技能。同时，通过开发一个具有 Windows 界面的应用程序，提高了实验的实用性和吸引力。

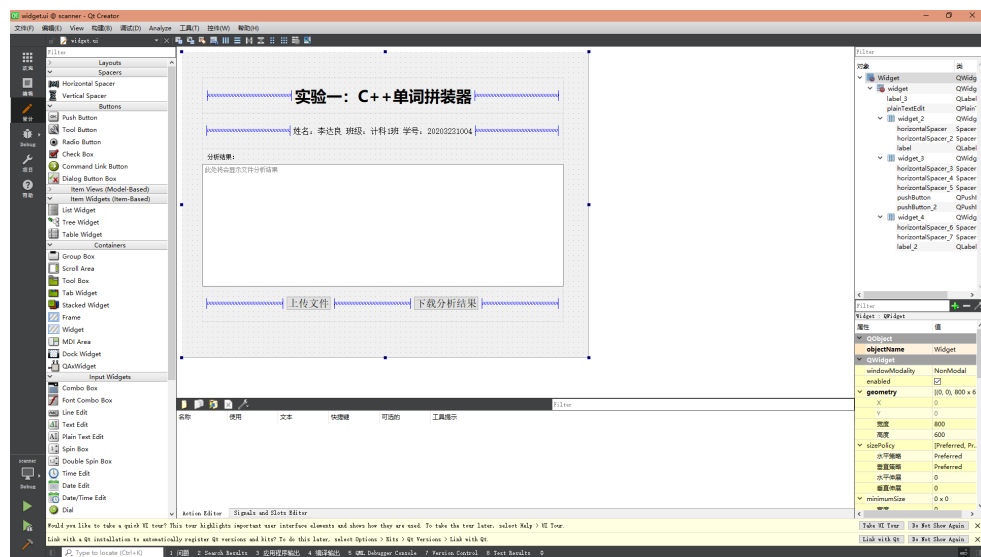
华南师范大学实验报告

学生姓名 李达良 学 号 20203231004
专 业 计算机科学与技术 年级、班级 2021 级 1 班
课程名称 编译原理 实验项目 C++单词拼装机
实验时间 2023 年 9 月 23 日
实验指导老师 黄煜廉 实验评分

四、实验文档

(1) 程序界面设计

通过 QT 实现 UI 的设计，如图所示，点击上传按钮，可以上传 C++ 文件或者 txt 文件，然后程序会对文件进行分析，得出结果显示在结果框上，并且可以点击下载分析结果来下载结果 txt 文件，操作简单便捷。

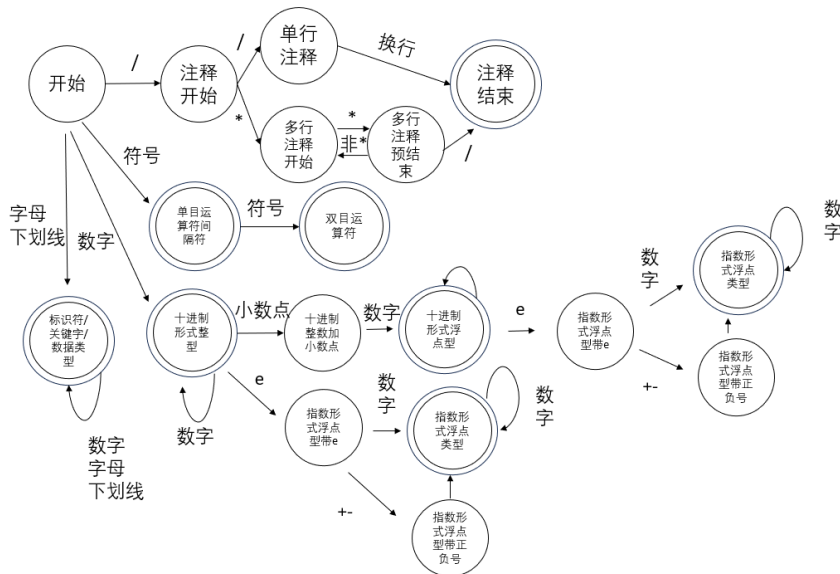


(2) 程序逻辑设计

对于这个实验，我们可以画出 DFA 图出来，有助于我们理清代码编写的顺序和逻辑，DFA 图如下：

华南师范大学实验报告

学生姓名 李达良 学 号 20203231004
专 业 计算机科学与技术 年级、班级 2021 级 1 班
课程名称 编译原理 实验项目 C++单词拼装机
实验时间 2023 年 9 月 23 日
实验指导老师 黄煜廉 实验评分



通过这个 DFA，我们可以有条理写出代码，使得程序完成大部分解析。

2.1 读入文件

其实 QT 读入文件非常简单，只需要一行就可以显示一个文件选取框，并拿到文件路径：

```
QString filePath = QFileDialog::getOpenFileName(this, tr("选择文件"), QDir::homePath(),  
tr("C++源文件 (*.cpp *.c);;文本文件 (*.txt);;所有文件 (*.*)");
```

这个代码限制了只能选择 cpp、c、txt 后缀的文件，并把文件路径传入 filePath 变量上，但是后面我们遇到一个问题，当这个路径是中文路径的时候，我们会读取文件失败，上网搜索了问题后，发现是字符集的问题，我们可以通过下面的代码进行解决：

```
if (!filePath.isEmpty())  
{  
    result.clear();  
    ifstream inputFile;  
    QTextCodec *code = QTextCodec::codecForName("GB2312");  
  
    string selectedFile =  
code->fromUnicode(filePath.toStdString().c_str()).data();  
    inputFile.open(selectedFile.c_str(), ios::in);  
  
    if (!inputFile) {  
        QMessageBox::critical(this, "错误信息", "导入错误！无法打开文件，请  
检查路径和文件是否被占用！");  
    }  
}
```

华南师范大学实验报告

学生姓名 李达良 学 号 20203231004
专 业 计算机科学与技术 年级、班级 2021 级 1 班
课程名称 编译原理 实验项目 C++单词拼装器
实验时间 2023 年 9 月 23 日
实验指导老师 黄煜廉 实验评分 _____

```
        cerr << "Error opening file." << endl;
    }
    char c;
    string currentWord;

    while (inputFile.get(c))
    {
        // 后续代码……
    }
}
```

这样，我们就能把文件流传入到 `inputFile` 这个变量上了，然后我们直接对文件流的每一个字符进行解析，判断是字母、数字还是符号，然后进行分词 `token` 提取。

2.2 注释解析

通过 DFA 图，我们可以知道，判断是否是注释，只要我们判断到一个/就可以进行解析，但注意，如果不是注释的话，我们要把它当作是一个运算符进行处理：

```
// 首先识别注释
if (c == '/')
{
    currentWord += c;
    if (inputFile.peek() == '/')
    { // 单行注释
        string currentLine;
        getline(inputFile, currentLine);
        currentWord += currentLine;
        cout << currentWord << " 注释" << endl;
        result.append(QString::fromStdString(currentWord + " 注释\n"));

        currentWord.clear();
        continue; // 继续下一轮循环
    }
    else if (inputFile.peek() == '*')
    { // 多行注释
        inputFile.ignore(); // 忽略掉'*'
        currentWord += '*';
        bool commentClosed = false;
```

华南师范大学实验报告

学生姓名 李达良 学 号 20203231004
专 业 计算机科学与技术 年级、班级 2021 级 1 班
课程名称 编译原理 实验项目 C++单词拼装器
实验时间 2023 年 9 月 23 日
实验指导老师 黄煜廉 实验评分 _____

```
while (inputFile.get(c))
{
    currentWord += c;
    if (c == '*' && inputFile.peek() == '/') {
        inputFile.ignore(); // 忽略掉 '*' 和 '/'
        currentWord += '/';
        commentClosed = true;
        break;
    }
}
if (!commentClosed)
{
    cerr << "Error: Unclosed multi-line comment." << endl;
}
else
{
    cout << currentWord << " 注释" << endl;
    result.append(QString::fromStdString(currentWord + " 注
释\n"));
}
currentWord.clear();
continue; // 继续下一轮循环
}
else
{
    operatorToken(c, inputFile);
    currentWord.clear();
}
}
```

其中，这里面的 `operatorToken()` 函数，是运算符识别的函数，待会我们会细讲。

2.3 头文件解析

解析完注释后，我们可以发现我们还有头文件需要处理，但是这个 `#` 号，其实不只有 `#include`，还有 `#define` `#ifdef` `#endif` 等，所以本次实验我们只解析 `#include`，其实其他的头，我们只需要加 `if` 语句也可实现。

// 头文件

```
else if (currentWord.empty() && c == '#') {
```

华南师范大学实验报告

学生姓名 李达良 学 号 20203231004
专 业 计算机科学与技术 年级、班级 2021 级 1 班
课程名称 编译原理 实验项目 C++单词拼装机
实验时间 2023 年 9 月 23 日
实验指导老师 黄煜廉 实验评分

```
cout << "#" << "\t 特殊符号" << endl;
result.append(QString::fromStdString("# 特殊符号\n"));
currentWord.clear();
while (inputFile.get(c) && (c != '<' && c != '\\')) {
    currentWord += c;
}
cout << currentWord << "\t 关键字" << endl;
result.append(QString::fromStdString(currentWord + " 关键字\n"));

currentWord.clear();
currentWord += c;
while (inputFile.get(c) && (c != '>' && c != '\\')) {
    currentWord += c;
}
currentWord += c;
cout << currentWord << "\t 特殊符号" << endl;
result.append(QString::fromStdString(currentWord + " 特殊符号\n"));

inputFile.get(c);
currentWord.clear();
}
```

2.4 基本函数准备

因为我们对程序进行解析，需要判断关键字、运算符、字母、数字甚至是是否是 0X 开头（判断十六进制），因此我们先准备一些基本的判断函数：

```
const string keywords[] = {

    "asm", "else", "new", "this", "auto", "enum", "operator", "throw", "bool", "explicit", "private", "true",

    "break", "export", "protected", "try", "case", "extern", "public", "typedef", "catch", "false", "register", "typeid",

    "char", "float", "reinterpret_cast", "typename", "class", "for", "return", "union", "const", "friend", "short", "unsigned",
```

华南师范大学实验报告

学生姓名 李达良 学 号 20203231004
专 业 计算机科学与技术 年级、班级 2021 级 1 班
课程名称 编译原理 实验项目 C++单词拼装机
实验时间 2023 年 9 月 23 日
实验指导老师 黄煜廉 实验评分

```
"const_cast", "goto", "signed", "using", "continue", "if", "sizeof", "virtual", "default",  
"inline", "static", "void",
```

```
"delete", "int", "static_cast", "volatile", "do", "long", "struct", "wchar_t", "double",  
"mutable", "switch", "while",
```

```
"dynamic_cast", "namespace", "template", "main", "cout", "cin", "using", "namespace", "  
std"  
};
```

```
const char operators[] = {  
    '+', '-', '*', '/', '=', '!', '>', '<', '|', '&', '^', '~', '?', '%'  
};
```

```
bool isDigit(char ch) {  
    if (ch >= '0' && ch <= '9') {  
        return true;  
    }  
    return false;  
}
```

```
bool isAlpha(char ch) {  
    if ((ch >= 'a' && ch <= 'z') || (ch >= 'A' && ch <= 'Z') || ch == '_') {  
        return true;  
    }  
    return false;  
}
```

```
bool isOperator(char ch) {  
  
    for (char i : operators) {  
        if (ch == i) {  
            return true;  
        }  
    }  
  
    return false;  
}
```


华南师范大学实验报告

学生姓名 李达良 学 号 20203231004
专 业 计算机科学与技术 年级、班级 2021 级 1 班
课程名称 编译原理 实验项目 C++单词拼装器
实验时间 2023 年 9 月 23 日
实验指导老师 黄煜廉 实验评分

}

```
bool isDelimiter(char ch) {  
    char delimiter[] = {  
        '(', ')', ',', ';', '{', '}', '#', '.', ':', '[', ']',  
    };  
  
    for (char i : delimiter) {  
        if (ch == i) {  
            return true;  
        }  
    }  
  
    return false;  
}
```

```
bool isKeyword(string token) {  
  
    for (string a : keywords) {  
        if (token.compare(a) == 0) {  
            return true;  
        }  
    }  
  
    return false;  
}
```

```
bool isOX(char ch)  
{  
    if ((ch >= 'a' && ch <= 'f') || (ch >= 'A' && ch <= 'F')) {  
        return true;  
    }  
    return false;  
}
```

2.5 数字解析

数字解析其实是比较麻烦的，主要麻烦在整数、浮点数的识别、科学计数法的识别、十

华南师范大学实验报告

学生姓名 李达良 学 号 20203231004
专 业 计算机科学与技术 年级、班级 2021 级 1 班
课程名称 编译原理 实验项目 C++单词拼装机
实验时间 2023 年 9 月 23 日
实验指导老师 黄煜廉 实验评分

六进制的识别, 但通过我们刚刚列出来的 DFA 解析图, 其实我们遇到数字就可以执行判断:

```
void numToken(char ch, ifstream& inputFile)
{
    char c;
    string numberToken;
    numberToken += ch;
    while (inputFile.get(c))
    {
        // 普通整型
        if (isDigit(c))
        {
            numberToken += c;
            continue;
        }
        // 浮点型+科学计数法
        else if (c == '.')
        {
            numberToken += c;
            while (inputFile.get(c))
            {
                if (isDigit(c))
                {
                    numberToken += c;
                    continue;
                }
                else if (c == 'e' || c == 'E')
                {
                    numberToken += c;
                    inputFile.get(c);
                    if (c == '+' || c == '-')
                    {
                        numberToken += c;
                    }
                }
                while (inputFile.get(c))
                {
                    if (isDigit(c))
                    {
                        numberToken += c;
                    }
                }
            }
        }
    }
}
```

华南师范大学实验报告

学生姓名 李达良 学 号 20203231004
专 业 计算机科学与技术 年级、班级 2021 级 1 班
课程名称 编译原理 实验项目 C++单词拼装器
实验时间 2023 年 9 月 23 日
实验指导老师 黄煜廉 实验评分

```
        continue;
    }
    break;
}
break;
}
}
// 整型的科学计数法
else if (c == 'e' || c == 'E')
{
    numberToken += c;
    inputFile.get(c);
    if (c == '+' || c == '-')
    {
        numberToken += c;
    }
    while (inputFile.get(c))
    {
        if (isDigit(c))
        {
            numberToken += c;
            continue;
        }
        break;
    }
}
// 十六进制
else if (c == 'x' || c == 'X')
{
    numberToken += c;
    while (inputFile.get(c))
    {
        if (isDigit(c) || isOX(c))
        {
            numberToken += c;
            continue;
        }
    }
}
```

华南师范大学实验报告

学生姓名 李达良 学 号 20203231004
专 业 计算机科学与技术 年级、班级 2021 级 1 班
课程名称 编译原理 实验项目 C++单词拼装器
实验时间 2023 年 9 月 23 日
实验指导老师 黄煜廉 实验评分

```
        break;
    }
}
break;
}
inputFile.unget(); // 回退一格
cout << numberToken << " 数字" << endl;
result.append(QString::fromStdString(numberToken + " 数字\n"));
}
```

2.6 字母解析

对于扫描到字母，只要后面仍然是字母或者数字，我们都一并添加到 token 里面，然后我们进行查表，判断是关键字还是标识符：

```
void alphaToken(char ch, ifstream& inputFile) {
    string alpha_token_value;
    alpha_token_value += ch;
    char c = inputFile.peek();

    //后面字符是字母或数字
    while (isAlpha(c) || isDigit(c))
    {
        alpha_token_value += c;
        inputFile.ignore();
        c = inputFile.peek();
    }

    //查表, 若不是保留字则是标识符
    if (isKeyword(alpha_token_value)) {
        cout << alpha_token_value << " 关键字" << endl;
        result.append(QString::fromStdString(alpha_token_value + " 关键字\n"));
    }
    else {
        cout << alpha_token_value << " 标识符" << endl;
        result.append(QString::fromStdString(alpha_token_value + " 标识符\n"));
    }
}
```

华南师范大学实验报告

学生姓名 李达良 学 号 20203231004
专 业 计算机科学与技术 年级、班级 2021 级 1 班
课程名称 编译原理 实验项目 C++单词拼装机
实验时间 2023 年 9 月 23 日
实验指导老师 黄煜廉 实验评分 _____

2.7 运算符解析

对于运算符，我们采用特殊判断的方法，因为其实运算符不是特别多，扫描到运算符的时候，我们要判断他后一个字符甚至后两个字符是不是也是运算符，如果是的话要拼起来，比如>, >>, >>=, 这些都是用 if-else 来进行判断，方便快捷：

```
void operatorToken(char ch, ifstream& inputFile) {  
    string operatorToken;  
    operatorToken += ch;
```

```
    char nextChar = inputFile.peek();
```

```
    if (nextChar == '=') // 各种 X=  
    {
```

```
        operatorToken += "=";  
        inputFile.ignore();  
    }
```

```
    else if (ch == '>' && nextChar == '>')  
    {
```

```
        operatorToken = ">>";  
        inputFile.ignore();  
        if (inputFile.peek() == '=') {  
            operatorToken = ">>=";  
            inputFile.ignore();  
        }  
    }
```

```
    else if (ch == '<' && nextChar == '<')  
    {
```

```
        operatorToken = "<<";  
        inputFile.ignore();  
        if (inputFile.peek() == '=') {  
            operatorToken = "<<=";  
            inputFile.ignore();  
        }  
    }
```

```
    else if (ch == '&' && nextChar == '&')  
    {
```

```
        operatorToken = "&&";
```

华南师范大学实验报告

学生姓名 李达良 学 号 20203231004
专 业 计算机科学与技术 年级、班级 2021 级 1 班
课程名称 编译原理 实验项目 C++单词拼装器
实验时间 2023 年 9 月 23 日
实验指导老师 黄煜廉 实验评分 _____

```
        inputFile.ignore();
    }
    else if (ch == '|' && nextChar == '|')
    {
        operatorToken = "||";
        inputFile.ignore();
    }
    else if (ch == '-' && nextChar == '>')
    {
        operatorToken = "->";
        inputFile.ignore();
    }
    else if (ch == '+' && nextChar == '+')
    {
        operatorToken = "++";
        inputFile.ignore();
    }
    else if (ch == '-' && nextChar == '-')
    {
        operatorToken = "--";
        inputFile.ignore();
    }

    cout << operatorToken << " 运算符" << endl;
    result.append(QString::fromStdString(operatorToken + " 运算符\n"));
}
```

2.7 字符串和单字符解析

对于扫描到双引号和单引号，我们要进行一个特殊处理：

字符串：

```
currentWord += c;
        while (inputFile.get(c) && (c != '\"')) {
            currentWord += c;
        }
        currentWord += c;
        cout << currentWord << " 字符串" << endl;
```

华南师范大学实验报告

学生姓名 李达良 学 号 20203231004
专 业 计算机科学与技术 年级、班级 2021 级 1 班
课程名称 编译原理 实验项目 C++单词拼装机
实验时间 2023 年 9 月 23 日
实验指导老师 黄煜廉 实验评分 _____

```
result.append(QString::fromStdString(currentWord + " 字符串\n"));

currentWord.clear();

单个字符:
currentWord += c;

while (inputFile.get(c) && (c != '\n')) {
    currentWord += c;
}
currentWord += c;
cout << currentWord << " 单字符" << endl;
result.append(QString::fromStdString(currentWord + " 单字符\n"));

currentWord.clear();
```

(3) 程序输出设计

我们将所有的结果，存入 result 字符，并在全局进行定义：

```
QString result;
```

然后，将 result 放入文本框内：

```
ui->plainTextEdit->setPlainText(result);
```

点击下载按钮逻辑：

```
void Widget::on_pushButton_2_clicked()
{
    // 保存结果到文本文件
    QString saveFilePath = QFileDialog::getSaveFileName(this, tr("保存结果文件"), QDir::homePath(), tr("文本文件 (*.txt)"));
    if (!saveFilePath.isEmpty() && !result.isEmpty()) {
        QFile outputFile(saveFilePath);
        if (outputFile.open(QIODevice::WriteOnly | QIODevice::Text)) {
            QTextStream stream(&outputFile);
            stream << result;
            outputFile.close();
            QMessageBox::about(this, "提示", "导出成功!");
        }
    }
}
```

华南师范大学实验报告

学生姓名 李达良 学 号 20203231004
专 业 计算机科学与技术 年级、班级 2021 级 1 班
课程名称 编译原理 实验项目 C++单词拼装器
实验时间 2023 年 9 月 23 日
实验指导老师 黄煜廉 实验评分

```
else if(result.isEmpty())  
{  
    QMessageBox::warning(this, tr("提示"), tr("结果为空, 请重试!"));  
}  
}
```

五、小结

本实验成功实现了一个用于词法分析的 Windows 界面应用程序。我们能够识别 C++ 源代码中的各种单词（标识符、关键字、数、字符串、注释、特殊符号、运算符），并将其分类。实验中，我们特别处理了单行注释和多行注释的情况，确保了完整的注释处理。此外，我们通过 Windows 界面提供了用户友好的交互方式，使用户能够轻松加载源文件并查看分类后的单词列表。

通过这个实验，我们能够更深入地理解编译原理中的词法分析概念，提高了编程和文档编写技能。总之，本实验成功地达到了教育和实践的双重目标，帮助学生更好地理解编译原理和构建实用的工具。