



C 语言编辑器

[打麻将吗小组]



团队信息				
姓名	班级	学号	邮箱	联系方式
石唯妍	08111705	1121072976	1021326870@qq. com	17801012116
任翔渝	08111706	1120173805	691540783@qq. com	13320390798
徐靖垚	08111706	1120172142	929616243@qq. com	13671092321
李传赫	08111706	1120172774	1085211797@qq. com	18801317991
杨思云	08111705	1120172981	Siyun_Y@163. com	18801119672

目录

第一部分：总体设计.....	1
1.1 模块外部设计.....	1
1.2 基本设计概念和处理流程.....	2
1.3 功能分配：.....	2
第二部分：接口设计.....	4
2.1 用户接口.....	4
2.2 外部接口.....	4
2.3 内部接口.....	4
第三部分：界面设计.....	5
3.1 界面设计图.....	5
3.1.1 启动界面.....	5
3.1.2 文件菜单.....	6
3.1.3 编辑菜单.....	7
3.1.4 调试菜单.....	8
3.1.5 帮助菜单.....	9
3.2 操作后界面.....	10
3.2.1 文件操作.....	10
3.2.2 其他操作.....	11
第四部分：功能设计.....	12
4.1 文件部分.....	12
4.1.1 新建功能实现.....	12
4.1.2 打开功能实现.....	12
4.1.3 保存功能实现.....	13
4.1.4 另存为功能实现.....	13
4.2 编辑部分.....	13
4.3 编译运行部分.....	13
4.4 帮助部分 4.4 帮助部分.....	16
第五部分：数据结构设计.....	17
5.1 逻辑结构设计.....	17
5.2 类图	18
5.2.2 控制台类 console	19
5.2.3 行号区域 LineNumberArea	19
5.2.4 高亮接口类 HighLighter	20
5.2.5 完善列表部件 CompleteListWidget 类.....	20
5.2.6 代码编辑类 CodeEditor.....	21
第六部分：运行设计.....	23
6.1 运行模块的组合.....	23
6.2 运行控制.....	23
6.3 运行时间.....	23
第七部分：出错处理设计.....	24

7.1 出错输出信息.....	24
7.2 出错处理对策.....	24
7.3 系统恢复设计.....	24
第八部分：安全保密设计.....	25

第一部分：总体设计

1.1 模块外部设计

文件操作：用户可以新建，打开，保存文件。

文本编辑：实时显示用户输入，能够实现复制/粘贴，全选，撤销/重做，在输入关键字首字母后可以进行补全。

文本高亮：用不同颜色标记关键字，头文件，函数，注释。

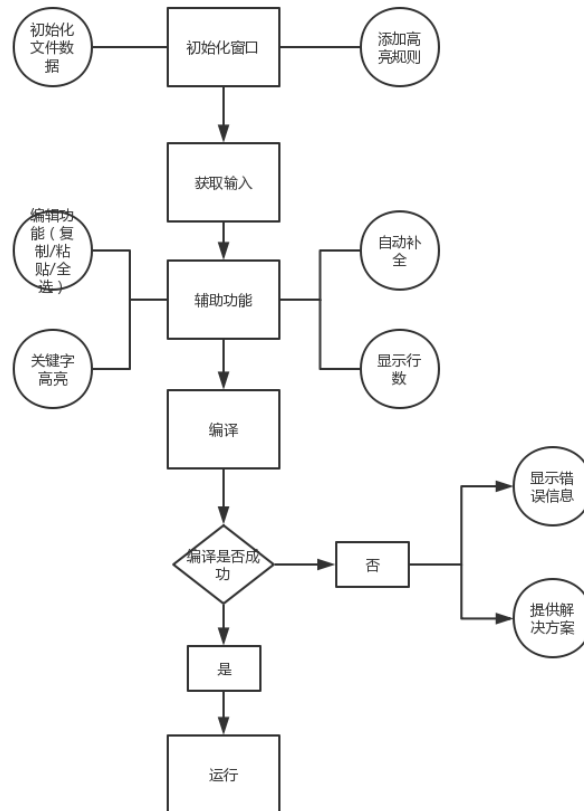
行数统计：当前光标所在行标红，并在左边显示行数。

编译：对当前代码进行编译，若编译成功返回“编译成功”，失败则返回错误信息。

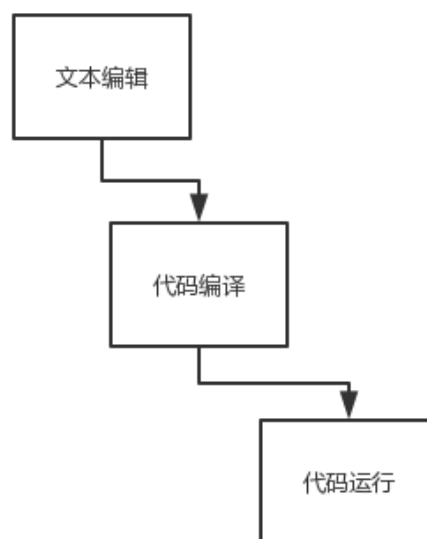
运行：对当前代码进行运行（需编译成功）。

1.2 基本设计概念和处理流程

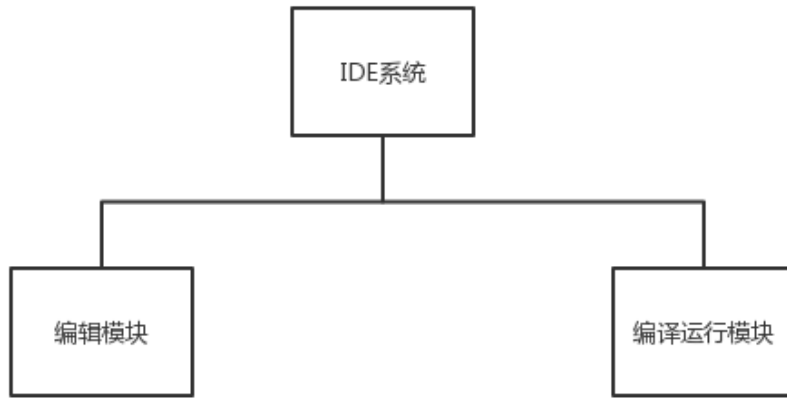
系统的总体处理数据流程如下图：



1.3 功能分配：



整体的功能模块图：



第二部分：接口设计

2.1 用户接口

用户的输入界面

2.2 外部接口

g++编译器。。

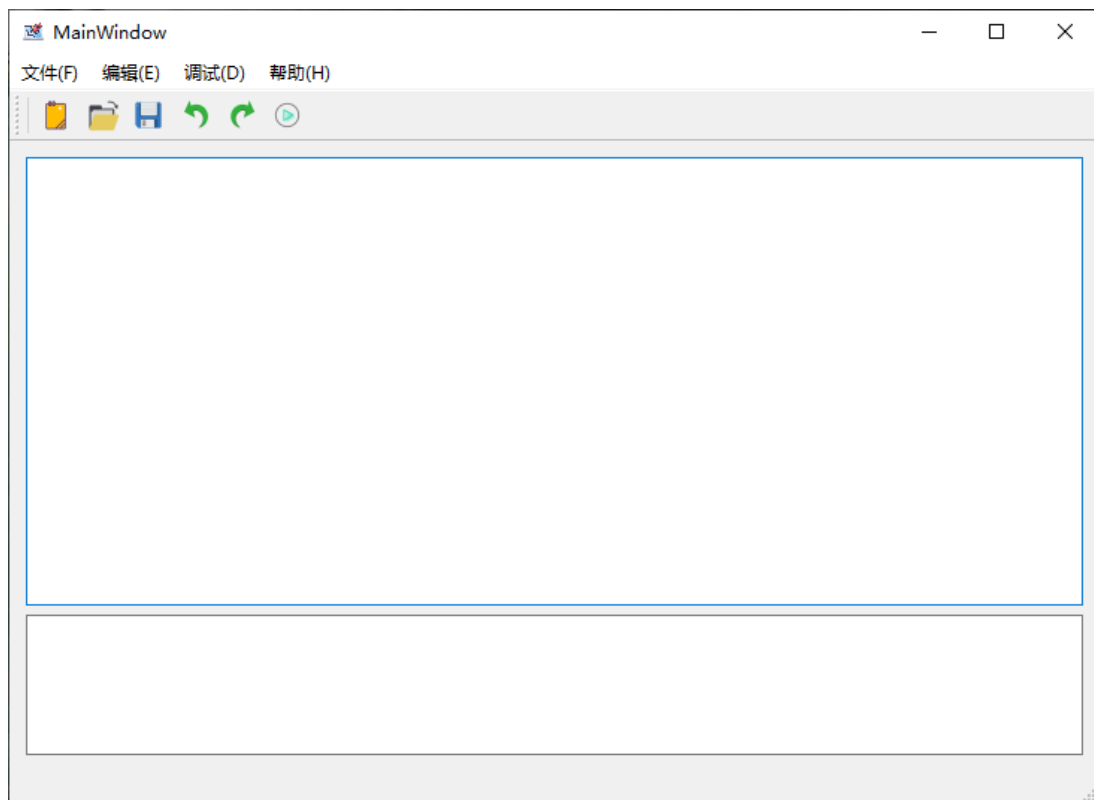
2.3 内部接口

函数接口（函数）

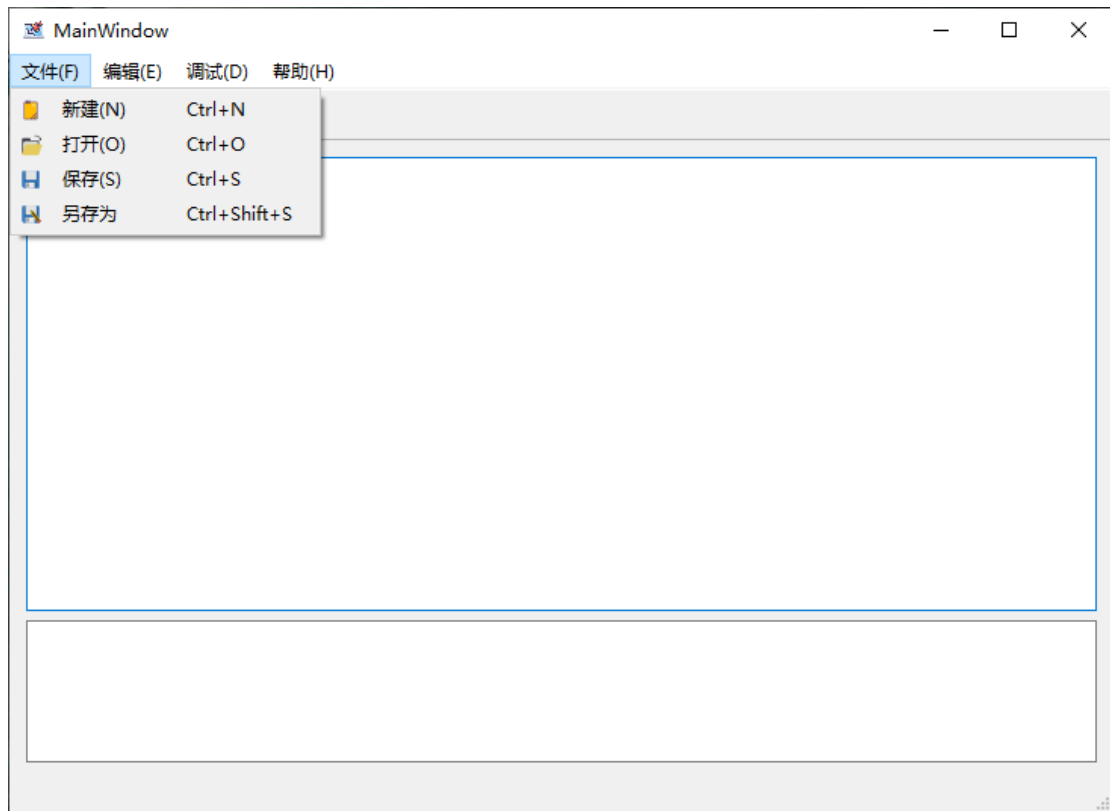
第三部分：界面设计

3.1 界面设计图

3.1.1 启动界面



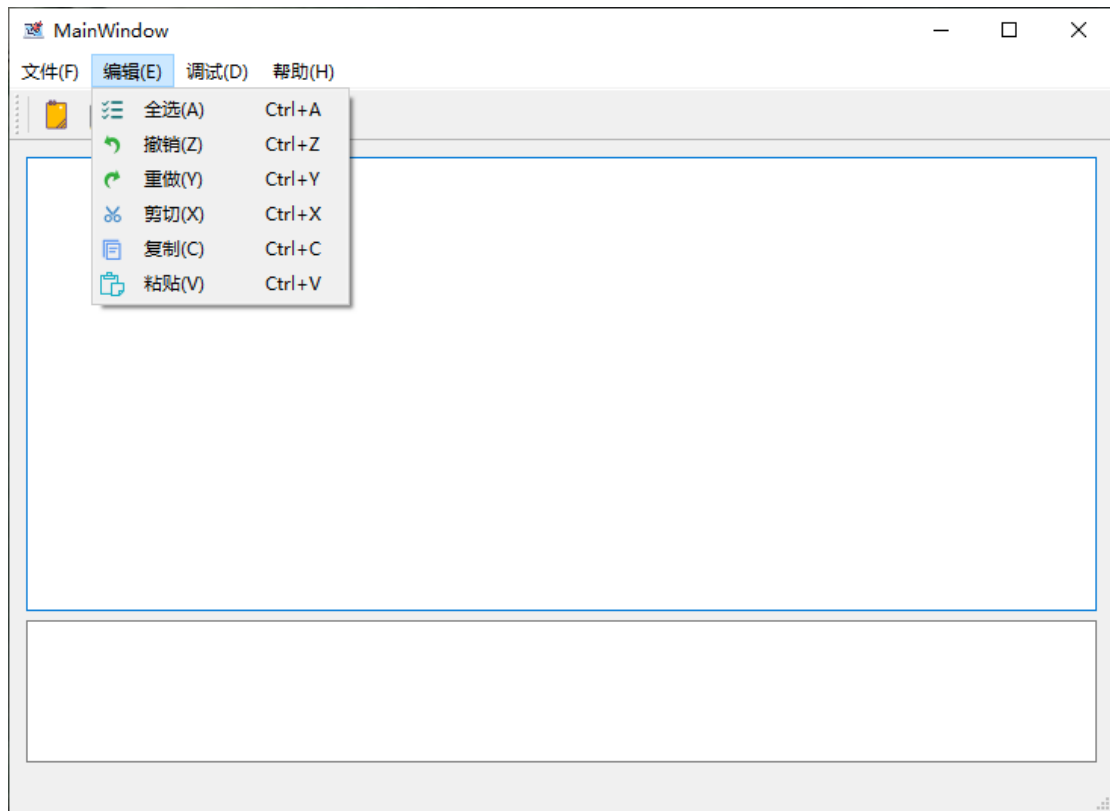
3.1.2 文件菜单



如图所示，文件菜单栏下拉显示功能有：新建、打开、保存、另存为，对应快捷键分别：

新建	Ctrl+N
打开	Ctrl+O
保存	Ctrl+S
另存为	Ctrl+Shift+S

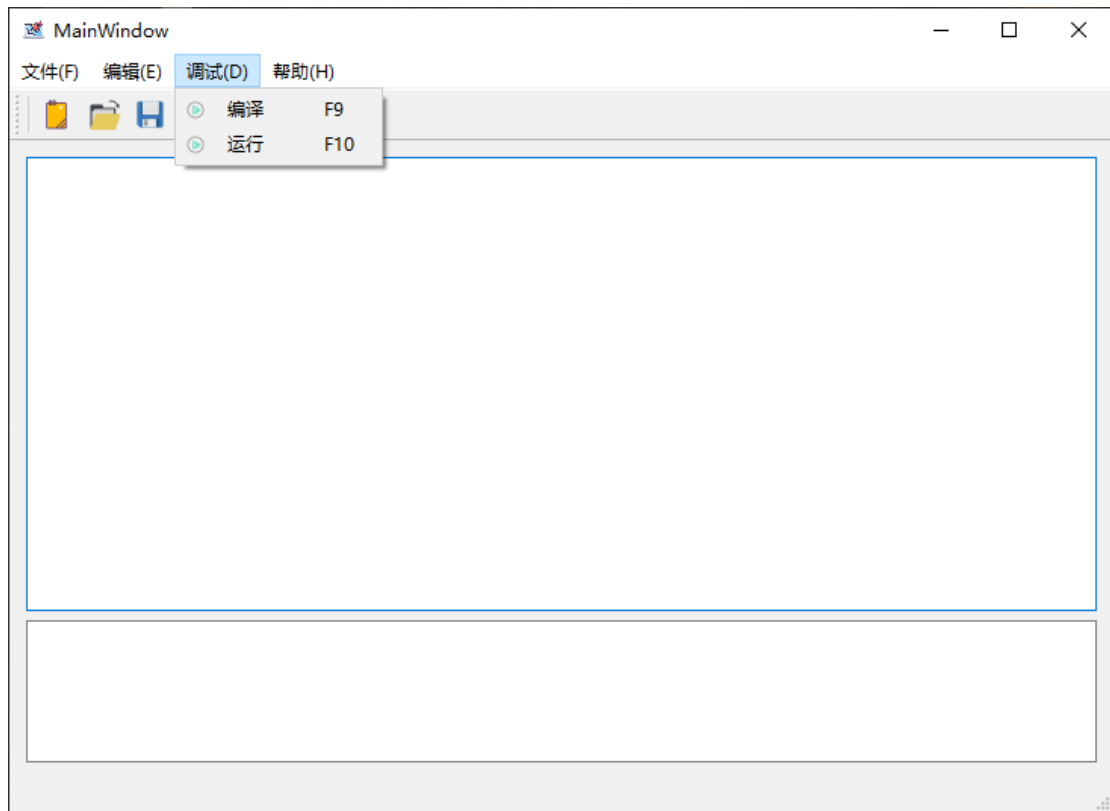
3.1.3 编辑菜单



如图所示，编辑菜单栏下拉显示功能有：全选、复制、粘贴、剪切、撤销、重做，对应快捷键分别为：

全选	Ctrl+A
复制	Ctrl+C
粘贴	Ctrl+V
剪切	Ctrl+X
撤销	Ctrl+Z
重做	Ctrl+Y

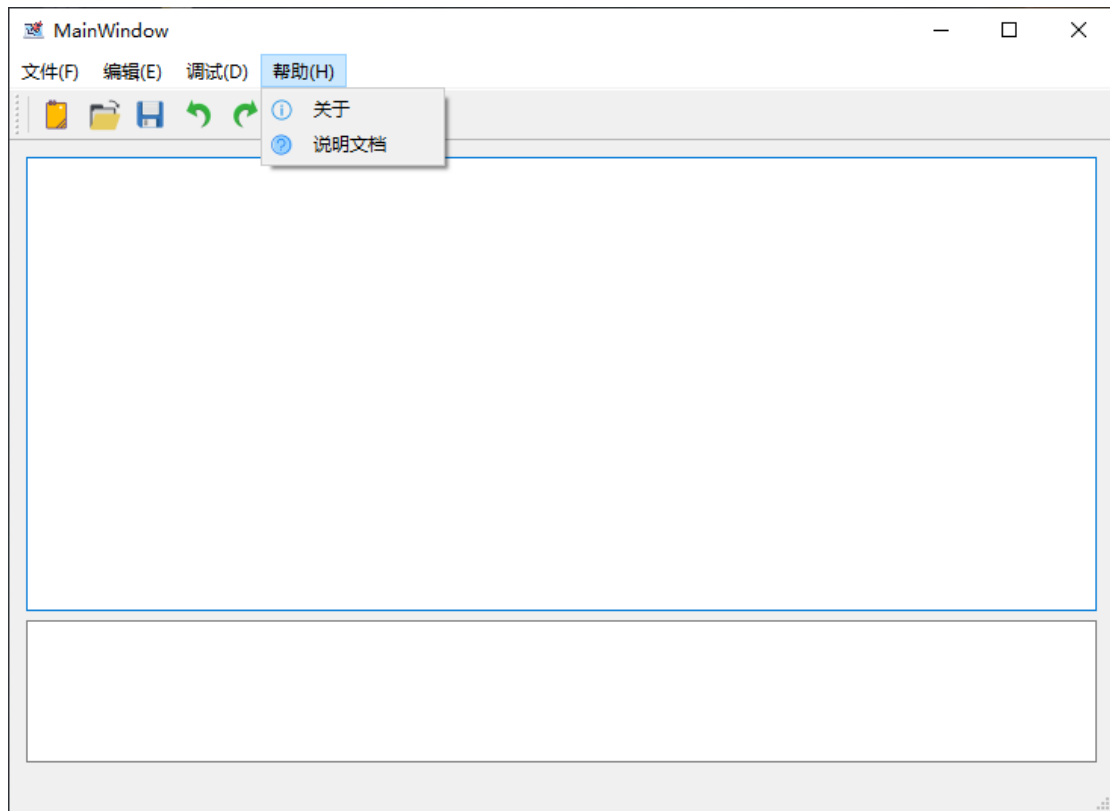
3.1.4 调试菜单



如图所示，调试菜单栏下拉显示功能有：编译、运行，对应快捷键分别为：

编译	F9
运行	F10

3.1.5 帮助菜单

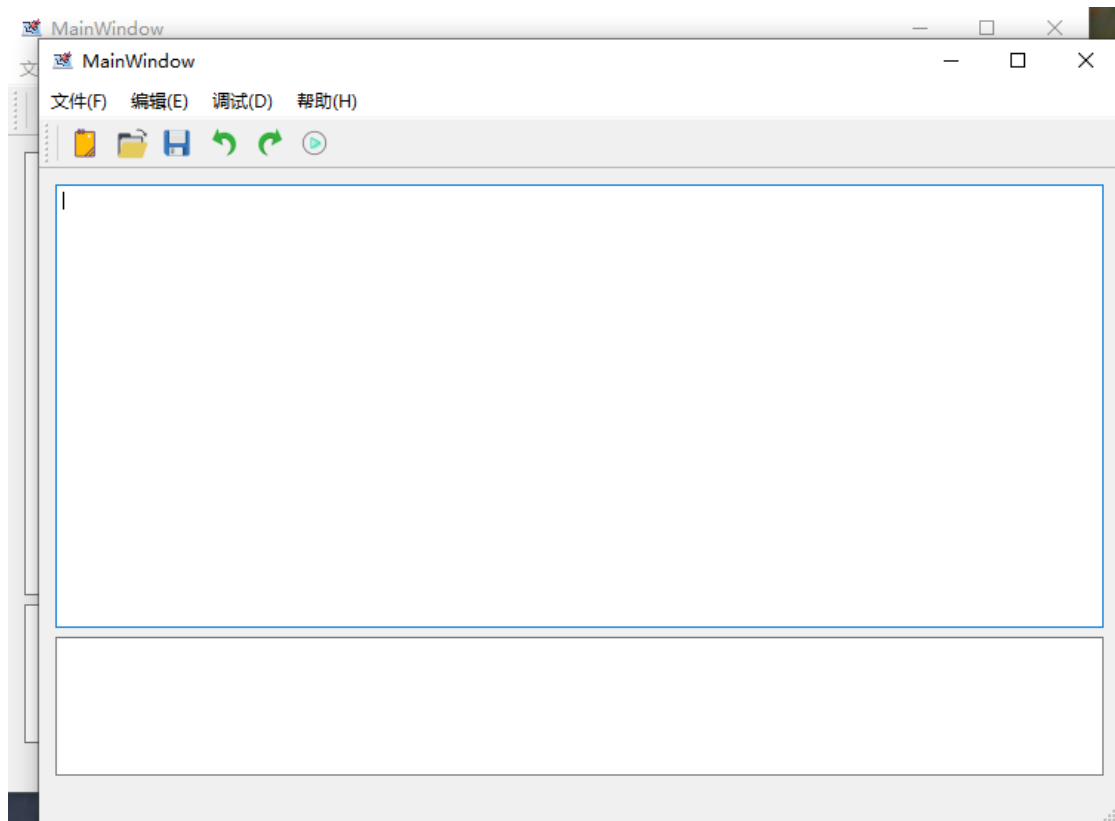


如图所示，帮助菜单栏下拉显示功能有：关于、说明文档

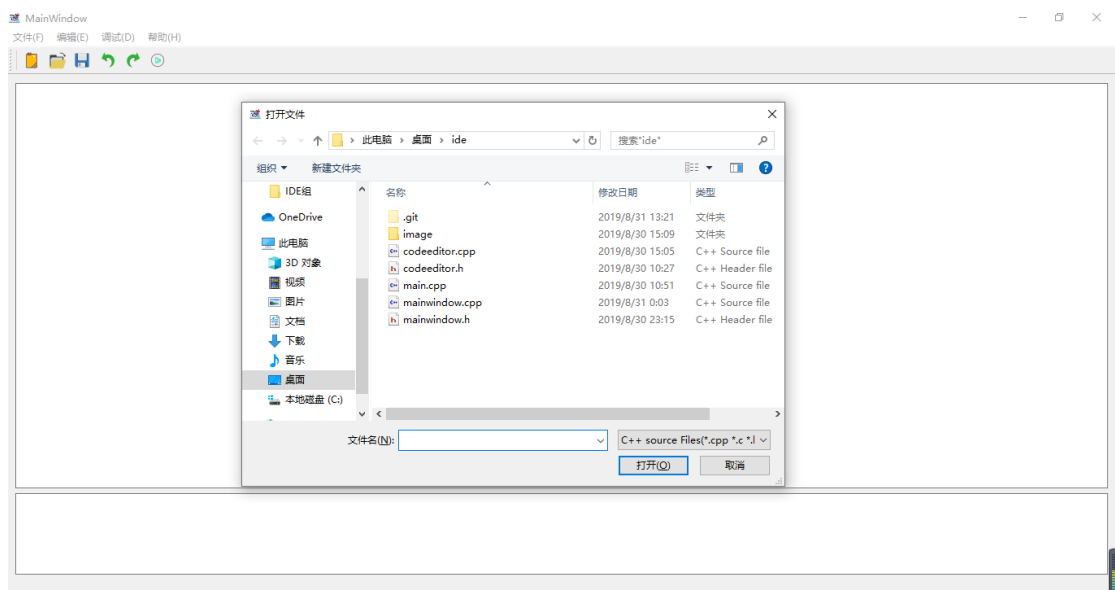
3.2 操作后界面

3.2.1 文件操作

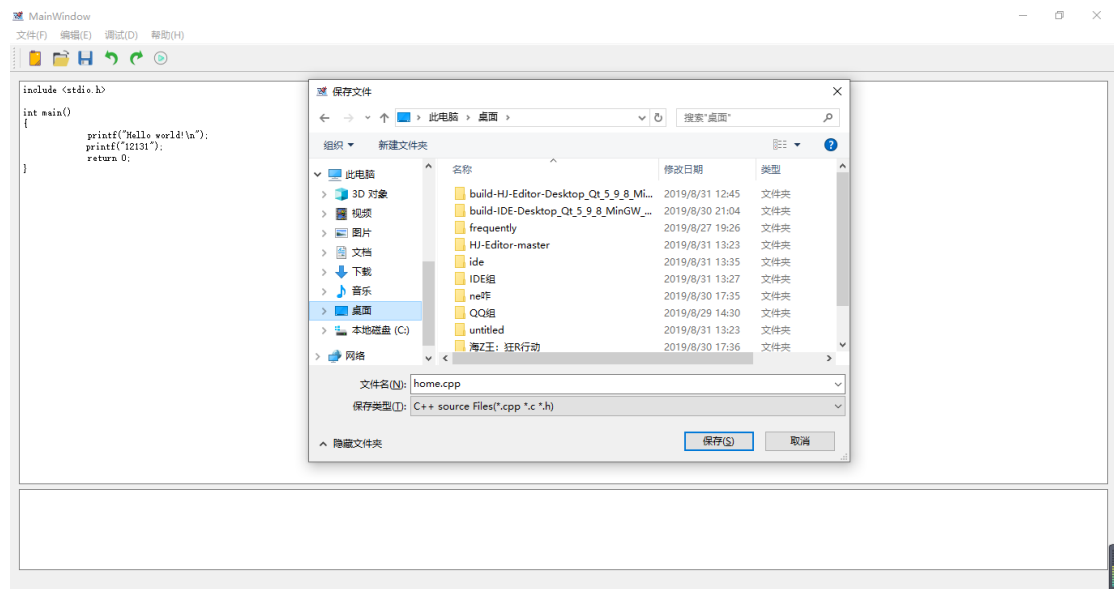
1. 新建：新建窗口



2. 打开文件



3. 保存文件



3.2.2 其他操作

1. 全选、复制、粘贴、剪切、撤销、重做等功能不做界面描述。
2. 编译运行显示输出结果、输入、报错信息均在控制台实现。
3. 帮助菜单下关于可查看开发人员信息。
4. 帮助菜单下说明文档可查看应用的具体操作手册。

第四部分：功能设计

4.1 文件部分

首先创建：

1. 私有变量 `Flag_isOpen`(bool 类型)初始值为 `false`，判断是否打开或创建了一个文件；
2. 私有变量 `Flag_isNew`(bool 类型)初始值为 `true`，判断文件是否为新建
3. 私有变量 `Last_FileName`(QString 类型)，记录最后一次保存的文件名；
4. 私有变量 `Last_FileContent`(QString 类型)，记录最后一次保存的文件内容；

4.1.1 新建功能实现

1. 新建主窗口对象
2. `QRect` 获取当前窗口位置信息
3. 新窗口 `x+10, y+10`
4. `Flag_isNew = true;`
5. `Flag_isOpen = true;`

4.1.2 打开功能实现

1. 首先获取文件名，判断文件名是否为空（是否直接关闭对话框）
2. 在判断文件是否可读
3. 如果没到文件末尾则一直读出到 `palintextedit` 中
4. `Flag_isOpen = true;`
5. `Flag_isNew = false;`

4.1.3 保存功能实现

1. 首先确定：
2. 新建文件后 `Flag_isNew=true` `Flag_isOpen=true`
3. 打开文件后 `Flag_isNew=false` `Flag_isOpen=true`
4. 如果是新文件：
 - a) 判断文件是否为空
 - b) 获取保存文件名，防止不命名直接关闭
 - c) 获取保存时的文件内容

4.1.4 另存为功能实现

在上述保存基础上判断文档是否保存过是否为打开的，如果是直接覆盖源文件保存即可。

4.2 编辑部分

编辑部分复制粘贴撤销等功能 `QTextEdit` 已封装好。

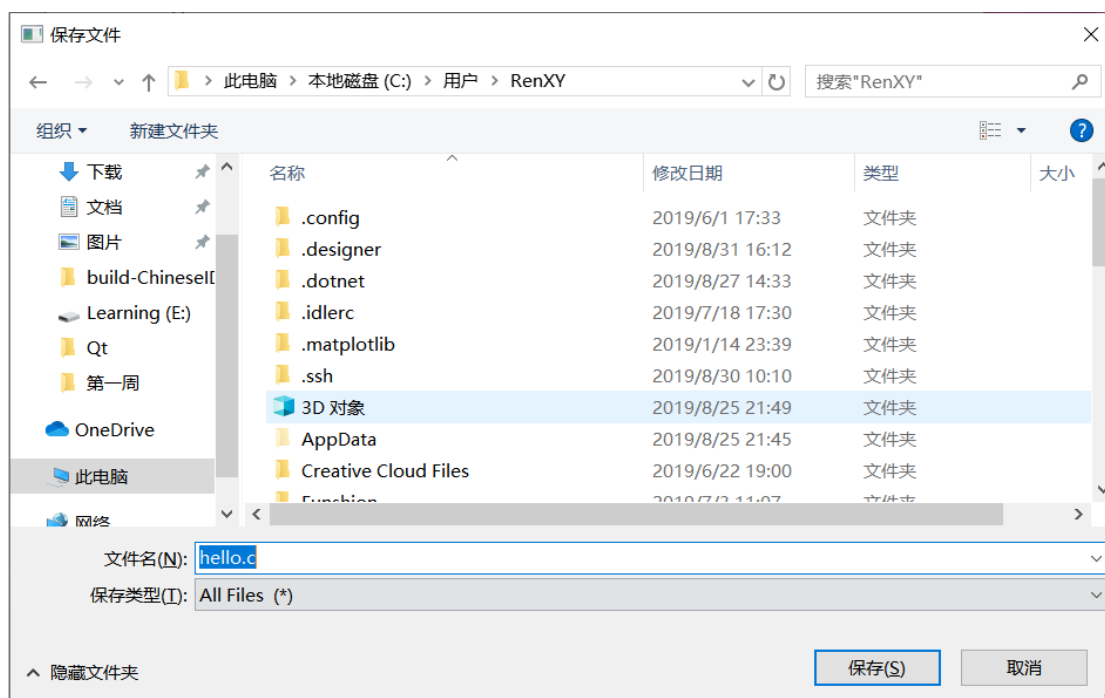
4.3 编译运行部分

以如下代码为例：



```
1 #include <stdio.h>
2 #include <stdlib.h>
3 int main()
4 {
5     printf("hello world\n");
6     system("pause");
7     return 0;
8 }
```

1. 如果是第一次编译，先保存文件

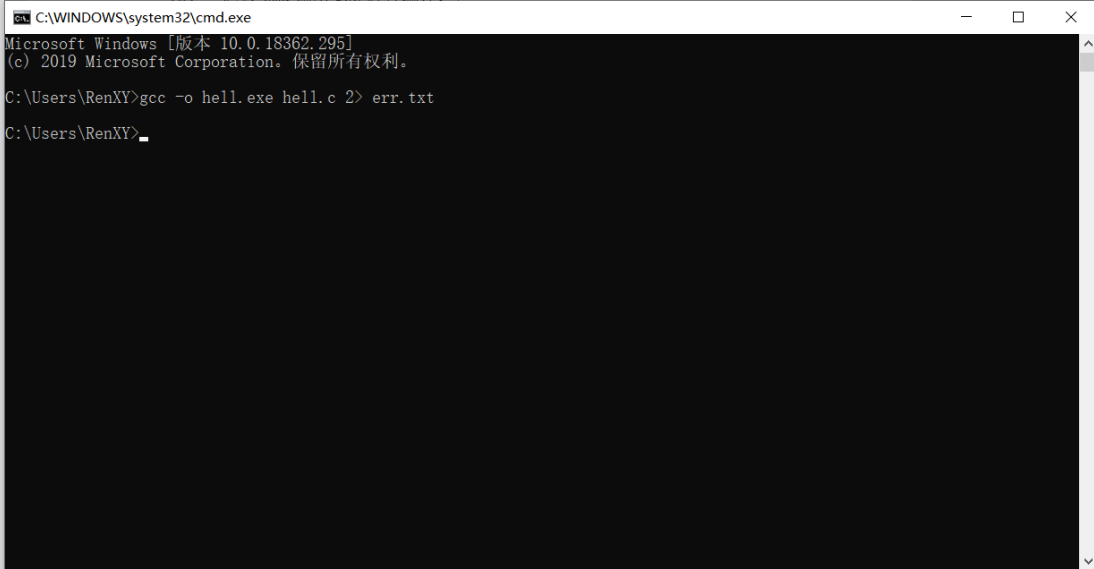


2. 编译过程:

- (1) 通过 system() 函数调用 cmd
- (2) 通过 cmd 调用 gcc 进行编译:

执行 gcc 命令 “gcc -o *.exe *.c 2> err.txt”，该命令可将 *.c 文件编译并转换为 *.exe 文件，并将错误信息输出至 *.c 文件所在目录的 err.txt

文件中

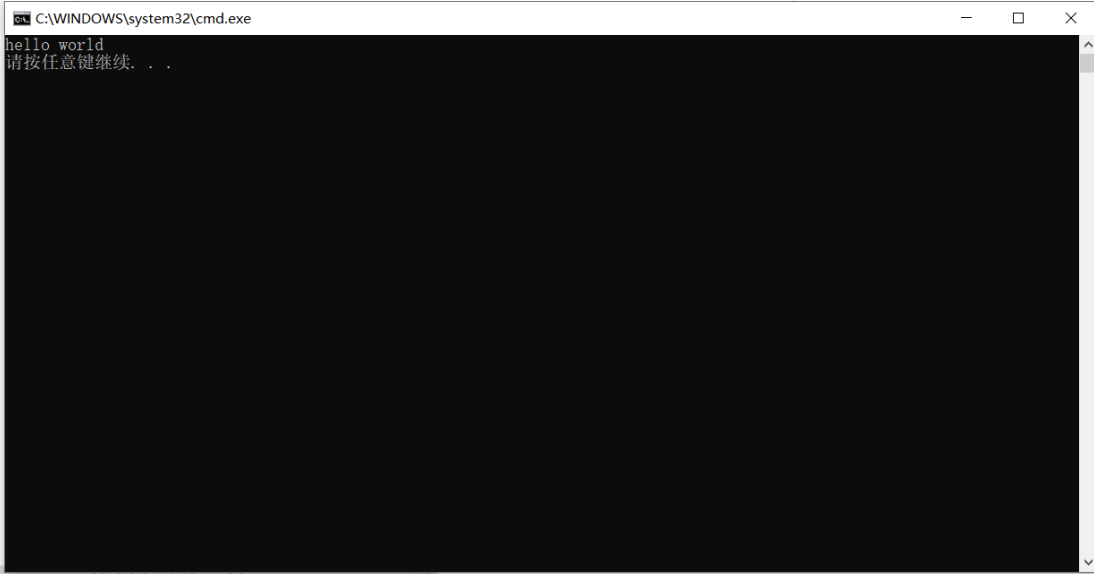


```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows [版本 10.0.18362.295]
(c) 2019 Microsoft Corporation. 保留所有权利。

C:\Users\RenXY>gcc -o hell.exe hell.c 2> err.txt
C:\Users\RenXY>
```

3. 运行过程:

- (1) 打开*.c 文件所在目录的 err.txt 文件
- (2) 若 err.txt 文件内容为空, 说明编译成功, 直接打开*.exe 文件展示输出结果即可



```
C:\WINDOWS\system32\cmd.exe
hello world
请按任意键继续. . .
```

- (3) 若 err.txt 文件内容不为空, 说明编译失败, 将错误信息输出至编辑器控制台显示即可

```
err.txt
1 hello.c: In function 'main':
2 hello.c:6:17: error: expected ';' before 'return'
3     system("pause")
4         |           ^
5         |           ;
6     return 0;
7     ~~~~~
8
```

4.4 帮助部分 4.4 帮助部分

参考: <https://blog.csdn.net/u014213012/article/details/54318137>

第五部分：数据结构设计

5.1 逻辑结构设计

文本文件以 .cpp 的格式在电脑中存储 以 QTextStream 类的方式在程序中存储

相关功能说明（如何实现）：

光标移动：用 `key_event()` 接收键盘输入，判断是否是功能按键后输入到文档中，光标自动跟进。 `int pos=this->textCursor().position()-1;`

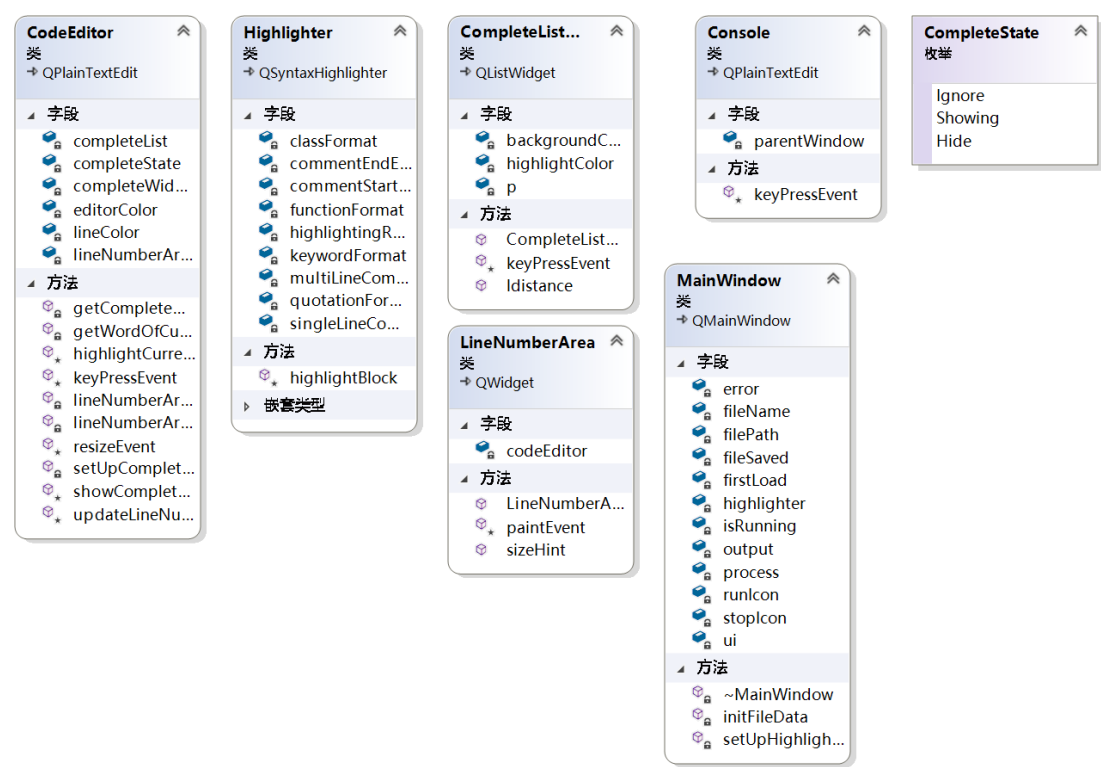
关键字高亮显示：事先录入需要高亮显示的关键字，每打出一个单词都会匹配，有匹配到的则高亮显

括号匹配：每键入一个(或“时，在文档中相当于键入()和””（即用()和””替换(和”）

关键字联想：事先录入可能联想到的关键字，每打出一个单词都会匹配，有匹配到的则会在下拉框中显示

文档编辑基本功能（复制、剪切、粘贴、撤销等）：使用 QT 中 QTextEdit 类自带的文档编辑功能

5.2 类图



5.2.1 主界面类 MainWindow

属性	类型	描述		
runIcon	QIcon	运行图标		
stopicon	QIcon	停止图标		
process	QProsess	用于启动其他程序		
fileName	QString	文件名		
filePath	QString	文件路径		
fileSaved	bool	文件是否存储		
isRunning	bool	代码是否运行		
firstLoad	bool	判断是否第一次加载用于更改顶端文件名显示		
output	QString	显示输出		
error	QString	显示报错		
方法 (private)	类型	传入参数	返回值	描述
setUpHighlighter()	void			代码高亮
initFileData()	void			初始化文件数据 (名称+位置)
方法 (public)	类型	传入参数	返回值	描述
changeSaveState()	void			更改项目顶端显示名称
newfile()	void			新建文件
savefile()	void			保存文件
openfile()	void			打开文件

undo()	void			撤销
redo()	void			重做
run()	void			运行
runFinished()	void	int code		运行完成
updateOutput()	void			更新输出
updateError()	void			更新错误
about()	void			项目信息
inputData	void	QString data		输入代码并限制代码运行时编辑代码
方法(protected)	类型	传入参数	返回值	描述
resizeEvent()	void	QResizeEvent* event		调整窗口大小时自适应
closeEvent()	void	QCloseEvent* event		关闭 IDE 时判断文件是否保存

5.2.2 控制台类 console

成员 (private)	类型		描述
parentWindow	MainWindow*		控制台输入
方法(protected)	参数	返回值	描述
keyPressEvent()	QKeyEvent	void	获取键盘输入
方法(private)	参数	返回值	描述
resetCursorPosition()	无	void	控制光标

5.2.3 行号区域 LineNumberArea

方法(public)	参数	返回值	描述
------------	----	-----	----

sizeHint()	无	Qsize	返回计数栏宽度, 调用 lineNumberAreaWidth()
方法 (protected)	参数	返回值	描述
paintEvent()	QPaintEvent	void	打印行号, 调用 lineNumberAreaPaintEvent()

5.2.4 高亮接口类 Highlighter

属性 (private)	类型	描述
highlightingRules	QVector<HighlightingRule>	数组存放关键字
commentStartExpression	QRegularExpression	多行注释的开始字段
commentEndExpression	QRegularExpression	多行注释的结束字段
keywordFormat	QTextCharFormat	关键字格式
classFormat	QTextCharFormat	类字符格式
singleLineCommentFormat	QTextCharFormat	单行注释格式
multiLineCommentFormat	QTextCharFormat	多行注释格式
quotationFormat	QTextCharFormat	头文件包含格式
functionFormat	QTextCharFormat	函数字符格式

方法 (public)	类型	传入参数	返回值	描述
Highlighter	构造函数	QTextDocument *parent		添加关键字对应字体格式规则到数组
方法 (protected)	类型	传入参数	返回值	描述
highlightBlock	void	QString &text		匹配多行注释 (/**/)

5.2.5 完善列表部件 CompleteListWidget 类

返回类型		参数	描述
	CompleteListWidget 类		
	CompleteListWidget(QWidget *parent=0);		构造函数, 设置背景 颜色、高亮
static int	ldistance(const string, const string);	两个 字符	计算并返回两个单 词之间的相似度 (完

		串	全相同为 0，越不相似值越大) 用于判断是否当前单词有联想单词
void	keyPressEvent(QKeyEvent *) override;	事件指针	接收键盘输入，并区分 alt 与 numlock 按键

5.2.6 代码编辑类 CodeEditor

	CodeEditor 类		
	CodeEditor(QWidget *parent = 0);		构造函数, 设置背景颜色, 初始化补全列表
void	lineNumberAreaPaintEvent(QPaintEvent *);	事件指针	打印左边行计数栏中的行数
int	lineNumberAreaWidth();		计算左边行计数栏占的宽度
void	resizeEvent(QResizeEvent *) override;	事件指针	调整左边行计数栏的宽度
void	keyPressEvent(QKeyEvent *) override;	事件指针	自动补全()""; 配置 alt、numlock、回车、空格键; 配置换行后的缩进; 配置其它所有按键
void	highlightCurrentLine();		高亮当前行
void	updateLineNumberArea(const QRect &, int);	滚动的长方形; 滚动行数	文本框滚动时同时滚动行数

void	showCompleteWidget();		关键字联想功能
QString	getWordOfCursor();		得到当前光标位置的字符串
int	getCompleteWidgetX();		返回联想关键字栏的行坐标 (下拉栏在屏幕中的位置)

第六部分：运行设计

6.1 运行模块的组合

具体软件的运行模块组合为程序多窗口的运行环境，各个模块在软件运行过程中能较好的交换信息，处理数据。

6.2 运行控制

软件运行时有较友好的界面，基本能够实现用户的数据处理要求。

6.3 运行时间

系统的运行时间基本可以达到用户所提出的要求。

第七部分：出错处理设计

7.1 出错输出信息

在用户使用错误的数据或访问没有权限的数据后，系统给出提示：“对不起，你非法使用数据，没有权限！”而且用户的密码管理可以允许用户修改自己的密码，不允许用户的匿名登录。

7.2 出错处理对策

由于数据在数据库中已经有备份，故在系统出错后可以依靠数据库的恢复功能，并且依靠日志文件使系统再启动，就算系统崩溃用户数据也不会丢失或遭到破坏。但有可能占用更多的数据存储空间，权衡措施由用户来决定。

7.3 系统恢复设计

如果当前系统出现一些异常，而影响了用户的正常使用，因为数据库有备份，程序代码也有备份，软件开发人员应尽快发现问题的原因，及时改正过来，以保证系统的正常运行。

第八部分：安全保密设计

系统的系统用户管理保证了只有授权的用户才能进入系统进行数据操作，而且对一些重要数据，系统设置为只有更高权限的人员方可读取或是操作。系统安全保密性较高。