说明文档

1. 项目介绍

我们小组选择的是第二个推荐选题：简易文本编辑器。

我们的程序具备：新建文件、打开文件、文本输入与删除、保存、另存为、退出等的读写功能，还有撤销、恢复、复制、粘贴、全选等高级编辑功能以及字体设置（字体、删除线、下划线等）功能。

1. 开发环境

按照课程推荐，我们小组使用的IDE是Visual Studio2019和Visual Studio2022。

1. 实现原理

由于我们小组的选题同其他选题相比看起来似乎是一个相对简单的选题，所以我们决定在读写编辑功能的基础之上添加一些其他的附加功能（撤销、恢复、改变文本的颜色和字体、文本下划线和删除线等）。

主窗口由start主指令调用winMainProc定义实例参数，在winMainProcProc中处理消息，实现消息循环并按照接受的消息执行命令。quitProc退出窗口。

打开、新建和保存文件通过openProc和newProc、saveProc和saveasProc实现，procStream提供流输入输出支持。

checkModified在用户未保存文件的情况下发起新建或者打开或者退出的时候提醒用户当前文件要保存一下。

通过创建位图，显示行首高亮；editProc实现了右键编辑区域打开辅助菜单的功能。

撤销、恢复、全选等高级编辑操作，可在用户点击相应菜单项或者按下相应快捷键时，向编辑窗口发送Richedit控件提供的相关消息（EM\_UNDO、EM\_REDO、EM\_EXSETSEL）以实现，其中的EM\_EXSETSEL则需要指定所选字符范围为全文（0~-1）。其他编辑操作（复制、粘贴、剪切、删除）则是发送winuser消息（WM\_COPY、WM\_PASTE、WM\_CUT、WM\_CLEAR）以实现。

字体设置则在\_FONT中，用CHOOSEFONT结构体初始化字体设置对话框，根据用户的选择对CHARFORMAT结构体（定义为全局变量）进行定义，并向编辑窗口发送EM\_SETCHARFORMAT消息完成字体的自定义设置。

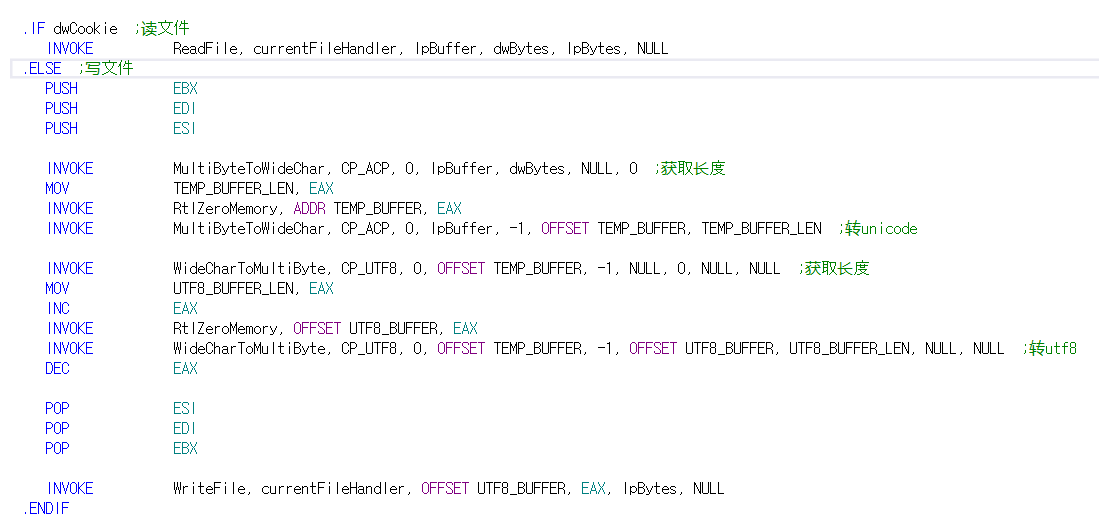
1. 创新点

原本的选题只是类似一个普通的记事本编辑器，不具备创新性，因此我们认为这个大作业选题的最大创新点就是为实现的那部分功能：改变文本的颜色和字体、文本高亮、文本下划线和删除线。虽然这些功能对于Word是再平常不过的，但是对于记事本确是极大的创新。

1. 难点

我们认为，最大的难点就是文件的读取，也是我们目前遇到的、无法克服的瓶颈：读取文件遇到中文时，只有通过这个编辑器生成的.txt文件中的内容可以正确显示出来，而其他的文件（.txt）不能正确显示出来。

分析其原因得知，我们的编辑器产生的.txt文件是用ANSI编码的，但是目前普遍使用的OS中的记事本.txt文件均使用UTF-8编码格式，导致OS中的记事本生成的UTF-8格式.txt文件无法正确读取。根据向助教咨询的结果以及网络搜索结果，我们进行了克服该难点的尝试，并且成功实现了我们的编辑器用UTF-8格式写文件（即，用UTF-8格式保存文件），但是最终未能完成读取UTF-8格式文件的功能。既然还是无法读取UTF-8格式的文件，我们的编辑器就算写一个UTF-8格式的文件也无法读回来。所以提交的程序中，我们的读写文件还是依然保持ANSI格式，删掉了实现的UTF-8写文件的部分。以下截图则是相关实现代码：



不过，我们的编辑器仍然可以支持ANSI、GB2312格式文件的完整读写！

1. 中期提交后的内容

我们希望做出一个不只是像记事本一样只具备文本编辑的文本编辑器，而是希望向Word靠近，做出一个有自己特色的文本编辑器，既具备文本编辑的基础功能，还具备供用户阅读文本的功能。

中期提交以后，我们添加的额外功能包括：行首高亮显示、字体设置、高级编辑功能（撤销、恢复、复制、粘贴、全选、剪切）。

1. 小组分工

中期前分工：

朴灿彬：搭建框架、文本读写

王展鹏：文本编辑、视频录制

岳 坤：窗口设计、书写文档

中期后分工：

朴灿彬：文本读写编码格式转换、行首高亮显示

王展鹏：字体设置功能

岳 坤：高级编辑功能（撤销、恢复、复制、粘贴、全选、剪切）