记事本

基础功能

- 支持文本文件的打开、保存、另存为功能
- 包含状态栏、能够展示文件大小

升级功能

1.标题展示文件名

2.支持输入回车

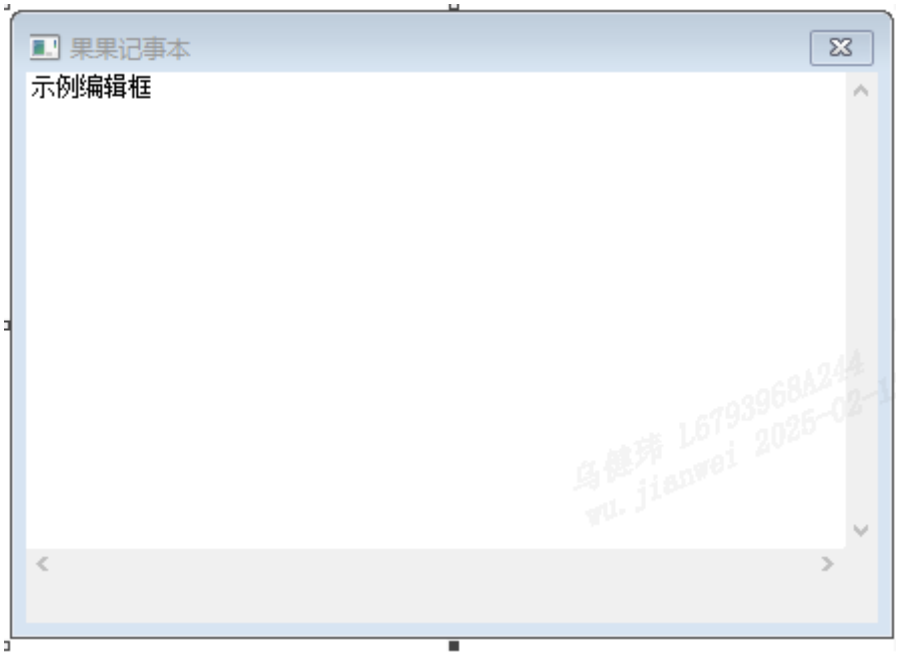
3.支持自动换行

4.支持自动伸缩（响应WM_SIZE信息）

5.支持快捷键，如保存：Ctrl+s

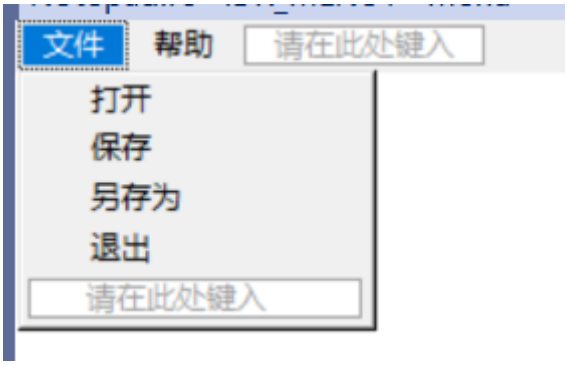
6.状态栏展示更多信息，如第几行、第几列、文件格式等等

首先把对话框配置好 中间一个edit框 下面一个静态文本框



然后就是把menu做好

在资源文件里把menu做好



打开

就把各种id改一下开始实现各个功能

```
case ID_MENU_OPEN:
        OpenFile(&npad);
        break;
```

第一个要实现的就是OpenFile(&npad);

一开始是传入的hwnd(窗口句柄) 但由于后面觉得窗口句柄内容不全 就做了个结构体 传入指针

```
struct NotePad {
    TCHAR FilePath[MAX_PATH];
    HWND hwndDlg;
    HWND hwndEdit;
    HWND hwndStatic;
};
```

然后来实现OpenFile(&npad);

直接用打开的这个函数就好

打开文件、另存为文件是常用功能，常用来选择本地文件或者保存为本地文件。

打开文件和另存为文件都用到了 `OPENFILENAME` 结构，它定义了对话框的外观和行为，具体参数说明见文档 OPENFILENAMEW (commdlg.h) - Win32 apps。

本节学习的功能函数，会使用就可以了，需要时再深入学习即可。

1. 打开文件

打开一个文件，其实目的就是获取一个本地文件的路径，我们编写最简单的版本，代码及说明如下：

```
void OpenFile(HWND hwndDlg)
{
    TCHAR path[MAX_PATH] = { 0 };      //定义数组用于保存文件路径

    OPENFILENAME ofn;                  //定义OPENFILENAME结构
    ZeroMemory(&ofn, sizeof(ofn));     //将结构体中的属性清0
    ofn.lStructSize = sizeof(ofn);     //结构体大小
    ofn.hwndOwner = hwndDlg;           //父窗口句柄
    ofn.lpstrFile = path;              //指向保存文件路径的数组
    ofn.nMaxFile = sizeof(path);       //保存文件路径的数组大小
    ofn.lpstrFilter = TEXT("头文件.h\0*.h\0All\0*.*\0");//过滤后缀名
    if (GetOpenFileName(&ofn)) {       //如果选择了文件，则弹出消息框
        MessageBox(NULL, path, TEXT("打开路径"), MB_OK);
        //获取文件路径后，一般会 open--->read 它
    }
}

```

Windwos 也提供了可以根据环境处理宽窄字符串的兼容版本库函数，一般用 `_t` 前缀修饰：

	ASCII 版	宽字符版	兼容版
字符类型	char	wchar_t	TCHAR
定义字符串	“”	L“”	TEXT()
字符指针	char * PSTR LPSTR	wchar_t * PWSTR LPWSTR	 PTSTR LPTSTR
只读字符指针	const char * PCSTR LPCSTR	const wchar_t * PCWSTR LPCWSTR	 PCTSTR LPCTSTR
求字符串长度	strlen	wcslen	_tcslen
字符串复制	strcpy strncpy	wcscpy wcsncpy	_tcscpy _tcsncpy
字符串连接	strcat strncat	wcscat wcsncat	_tcscat _tcsncat
字符串比较	strcmp	wcscmp	_tcscmp

这是通过【控件句柄】写入文本框的函数：

```
BOOL SetWindowText(HWND hWnd, LPCTSTR lpString);
```

	ASCII 版	宽字符版	兼容版
求字符串长度	strlen	wcslen	_tcslen
字符串复制	strcpy strncpy	wcscpy wcsncpy	_tcscpy _tcsncpy
字符串连接	strcat strncat	wcscat wcsncat	_tcscat _tcsncat
字符串比较	strcmp strncmp	wcscmp wcsncmp	_tcscmp _tcsncmp
字符串搜索	strstr strchr strrchr	wcsstr wcschr wcsrchr	_tcsstr _tcschr _tcsrchr
将字符串转换成数字	atoi atof	_wtoi _wtof	_ttoi _ttof
格式化字符串	sprintf snprintf	swprintf _snwprintf	_stprintf _sntprintf

```
void OpenFile(struct NotePad *np)
{
    TCHAR path[MAX_PATH] = { 0 };//地址 宽字符
    TCHAR Text[32] = { 0 };//内容
    long fileSize;//文件大小

    OPENFILENAME ofn;//ofn是结构体类型的
    ZeroMemory(&ofn, sizeof(ofn));
    ofn.lStructSize = sizeof(ofn);
    ofn.hwndOwner = np->hwndDlg;//由于传入的是个指针 把这个地方改一下就好
    ofn.lpstrFile = path;
    ofn.nMaxFile = sizeof(path);
    ofn.lpstrFilter = TEXT("txt文本文档\0*.txt\0All\0*.*\0");//文件过滤器 会把其他格式的过滤掉
    if (GetOpenFileName(&ofn)) {
        _tcscpy(np->FilePath, path);//把字符串复制一下 path本来就是宽字符 就不用变了 直接拷贝
        fileSize = LoadFileToEdit(np->FilePath, np->hwndEdit);//再写一个函数 专门处理把文件里的内容传到文本框里 并放回文件大小
        if (fileSize != -1) {
            _stprintf(Text, TEXT("文件大小：%ld 字节"), fileSize);//格式化处理
            SetWindowText(np->hwndStatic, Text);//把文本内容放到静态文本框中
        }
    }
}
```

然后写一下LoadFileToEdit

```
// 加载文件到编辑框
long LoadFileToEdit(TCHAR* FilePath, HWND hwndEdit)
{
    char file_name[MAX_PATH];//文件的名字 窄字符
    WideCharToMultiByte(CP_ACP, 0, FilePath, -1, file_name, MAX_PATH, NULL,
NULL);//把宽字符传进来的文件名字转为窄字符
    FILE* file = fopen(file_name, "rb");//用流打开文件
    if (file == NULL)
    {
        MessageBox(hwndEdit, TEXT("打开文件失败"), TEXT("提示"),
MB_ICONEXCLAMATION);
        return -1;
    }

    long fileSize = get_file_size(file);//写一个函数来得到文件大小 并返回文件大小作为这
个函数的返回值
    if (fileSize == -1){
        MessageBox(hwndEdit, TEXT("获取文件大小失败"), TEXT("提示"),
MB_ICONEXCLAMATION);
        // 关闭文件
        fclose(file);
        return fileSize;
    }

    // 读取文件内容到缓冲区
    size_t bytesRead = fread(buffer, 1, fileSize, file);//将文件内容给到buffer

    MultiByteToWideChar(CP_ACP, 0, buffer, -1, wBuffer, bytesRead);//由于文件内容是
窄字符 要给到edit需要宽字符就需要转换
    // 将文件内容加载到编辑框
    SetWindowText(hwndEdit, wBuffer);//把宽字符的文件内容给到edit文本框

    // 关闭文件
    fclose(file);

    return fileSize;//返回值为文件大小 之后可以给静态文本框使用
}
```

然后实现一下

获取文件位置指针偏移量 `ftell` 函数

原型

```
long ftell(FILE *stream);
```

功能

用于获取当前文件位置指针的偏移量

参数

`stream`：指向 `FILE` 结构的指针。

返回值

当前文件位置指针的偏移量，从 `0` 开始。

移动文件位置指针 `fseek` 函数

原型

```
int fseek(FILE *stream, long offset, int whence);
```

功能

将文件位置指针设置到文件的任意位置。

参数

`stream`：指向 `FILE` 结构的指针。

`offset`：偏移量，可以为正数或负数，表示相对于 `whence` 参数的偏移，单位字节。

`whence`：定位的参考点，支持3个参数：

- `SEEK_SET`（文件起始处）
- `SEEK_CUR`（当前文件位置）
- `SEEK_END`（文件末尾）

返回值

- 成功：0
- 失败：-1

```c
long get_file_size(FILE *file)
{
    long size = -1;  // 默认值，表示获取文件大小失败

    if (file != NULL) {
        // 获取当前文件位置
        long currentPosition = ftell(file);

        if (currentPosition != -1) {
            // 移动文件位置指针到文件末尾
            if (fseek(file, 0, SEEK_END) == 0) {
                // 获取文件末尾的位置，即文件大小
                size = ftell(file);
            }

            // 恢复文件位置指针到原来的位置
            fseek(file, currentPosition, SEEK_SET);//SEEK_SET 文件其实位置
```

```
        }
    }

    return size;
}
```

第一个打开文件的功能就实现了

然后来实现保存的功能

```
    case ID_MENU_SAVE:
        SaveFile(&npad);
        break;
```

设计了一个从edit文本框保存到文件的函数

```
void SaveFile(struct NotePad *np)
{
    SaveEditToFile(np->hwndEdit, np->FilePath);
}
```

把文本框里的内容 传到buufer上 再传到文件里

```
void SaveEditToFile(HWND hwndEdit, TCHAR* FilePath)
{
    // 获取编辑框中的文本内容
    GetWindowText(hwndEdit, wBuffer, sizeof(wBuffer));//把编辑框内容给到wBuffer
wBuffer是宽字符 文本框里的也是宽字符
    int used_buffer_size = WideCharToMultiByte(CP_ACP, 0, wBuffer, -1, buffer,
sizeof(buffer), NULL, NULL);//把wBuffer转为窄字节的buffer 并且把文件大小返回给
used_buffer_size

    char file_name[MAX_PATH];//窄字符的文件名
    WideCharToMultiByte(CP_ACP, 0, FilePath, -1, file_name, MAX_PATH, NULL,
NULL);//把宽字节的FilePath转为file_name
    FILE* file = fopen(file_name, "wb");//可写 打开流
    if (file == NULL)//打开失败
    {
        MessageBox(hwndEdit, TEXT("打开文件失败"), TEXT("提示"),
MB_ICONEXCLAMATION);
        return ;
    }

    //写入字符串的长度，不包含结尾的 null 字符
    fwrite(buffer, sizeof(char), used_buffer_size - 1, file);//将buffer写入文件中

    // 关闭文件
    fclose(file);
}
```

另存为

然后再来实现另存为

```
    case ID_MENU_SAVE_AS:
        SaveAsFile(&npad);
        break;
```

首先 直接用这个 然后在最后做点修改

2. 另存为文件

另存为文件的目的其实也是获取一个文件的路径，与打开文件的区别是，它获取的是一个**还不存在的文件的路径。**

```
1   void SaveAsFile(HWND hwndDlg)
2   {
3       TCHAR path[MAX_PATH] = { 0 };     //定义数组用于保存文件路径
4
5       OPENFILENAME ofn;                 //定义OPENFILENAME结构
6       ZeroMemory(&ofn, sizeof(ofn));    //将结构体中的属性清0
7       ofn.lStructSize = sizeof(ofn);    //结构体大小
8       ofn.hwndOwner = hwndDlg;          //父窗口句柄
9       ofn.lpstrFile = path;             //指向保存文件路径的数组
10      ofn.nMaxFile = sizeof(path);      //保存文件路径的数组大小
11      ofn.lpstrFilter = TEXT("头文件.h\0*.h\0All\0*.*\0");//过滤后缀名
12
13      if (GetSaveFileName(&ofn)) {      //如果填写了文件名，则弹出消息框
14          MessageBox(NULL, path, TEXT("另存为的路径"), MB_OK);
15          //获取文件路径后，一般会 open--->write 它
16      }
17  }
```

	ASCII 版	宽字符版	兼容版
字符类型	char	wchar_t	TCHAR
定义字符串	“”	L“”	TEXT()
字符指针	char * PSTR LPSTR	wchar_t * PWSTR LPWSTR	 PTSTR LPTSTR
只读字符指针	const char * PCSTR LPCSTR	const wchar_t * PCWSTR LPCWSTR	 PCTSTR LPCTSTR
求字符串长度	strlen	wcslen	_tcslen
字符串复制	strcpy strncpy	wcscpy wcsncpy	_tcscpy _tcsncpy
字符串连接	strcat strncat	wcscat wcsncat	_tcscat _tcsncat

```
void SaveAsFile(struct NotePad *np)
{
    TCHAR path[MAX_PATH] = { 0 };

    OPENFILENAME ofn;
    ZeroMemory(&ofn, sizeof(ofn));
```

```
    ofn.lStructSize = sizeof(ofn);
    ofn.hwndOwner = np->hwndDlg;
    ofn.lpstrFile = path;
    ofn.nMaxFile = sizeof(path);
    ofn.lpstrFilter = TEXT("txt文本文档\0*.txt\0");//文件过滤器

    if (GetSaveFileName(&ofn))
    {
        _tcscat(path, TEXT(".txt"));//自动加上文件格式
        SaveEditToFile(np->hwndEdit, path);//调用之前写的把文本框内容保存到文件里
    }
}
```

退出

然后实现退出的功能 这个最简单

```
    case ID_MENU_QUIT:
        EndDialog(hwndDlg, 0);
        break;
```

关于

然后实现关于

```
    case ID_MENU_ABOUT:
        ShowVersion(hwndDlg);
        break;
```

在ShowVersion函数里实现一下版本信息的更新

```
void ShowVersion(HWND hwndDlg)
{
    MessageBox(hwndDlg, TEXT(" \
作品更新历史:\n \
<-------------------------------------------------------->\n \
V1.0\n \
首个版本，实现了文本文件的打开、保存、另存为功能，能够展示文件大小 \
"), TEXT("版本号V1.0"), MB_OK);
}
```