# 学习大纲

|  |
| --- |
|  |

## 1，windows api 有3个版本应该是xxxA版本，一个是xxxW版本，一个是xxxx版本

## 2.windows的数据类型也有3个版本如字符串，有ANSI版，UNICODE版本，和TCHAR版本，其中TCHAR会根据你是否使用UNICODE字符集而变化可以是ansi也可以是unicode，他们的初始化方法如下：

|  |
| --- |
|  |

## 3.WinMain函数也是有多个版本的，ansi使用WinMain，UNICODE使用wWinMain，TCHAR使用\_tWinMain

## 4.MessageBox函数的不同版本演示

|  |
| --- |
|  |

### 微软建议我们使用T版，也就是通用版本，不过现在需要逐渐项W版过渡，有一些不需要字符串作为参数的api只有一个版本，不分A版或者W版

# 参考：[三个版本的windows磁盘操作API的使用](Windows%20API学习%20三种版本%20字符转换%20遍历卷%20磁盘属性%20磁盘容量.docx)

# 演练

## 1.创建一个常规的空项目，叫做Lesson10-api-3-ver，然后添加一个main函数，我们尝试宽字符串很多普通ansi字符串的转换，我们有两个函数，一个是unicode-》multibyte，另一个是multibyte-》unicode，代码如下：

|  |
| --- |
| #include<Windows.h>  #include<iostream>  #include<Locale.h>  using namespace std;  DWORD dword;  void testMultiBytesToWideChar()  {  char c[] = "多字节字符串转换成宽字符串";  //先求出需要多大的空间  dword = MultiByteToWideChar(CP\_ACP,0,c,-1,NULL,0);  //根据这个值来创建一个宽字符串指针  wchar\_t \*wct = new wchar\_t[dword];  //再调用一次这个函数  MultiByteToWideChar(CP\_ACP,0,c,-1,wct,dword);  wprintf(L"result:%s\n",wct);  }  void testWideCharToMultiBytes()  {  wchar\_t wcs[] = L"宽字符串转换成多字节字符串";  //先求出需要多大的空间  dword = WideCharToMultiByte(CP\_OEMCP,0,wcs,-1,NULL,0,NULL,NULL);  ////根据这个值来创建一个符串指针  char \* str = new char[dword];  //再调用一次这个函数  WideCharToMultiByte(CP\_OEMCP,0,wcs,-1,str,dword,NULL,NULL);  printf("result: %s\n",str);  }  int main()  {  setlocale(LC\_ALL,""); //设置本地字符集为中文  //testMultiBytesToWideChar();  testWideCharToMultiBytes();  system("pause");  return 0;  } |

## 2.multibyte-》unicode的效果：

|  |
| --- |
|  |

## 3.unicode-》multibyte的效果：

|  |
| --- |
|  |

### 宽字符版本

|  |
| --- |
| #include<Windows.h>  #include<iostream>  #include<locale.h>  void TestWideCharToMuiltiBytes()  {  WCHAR szWC[] = L"好好学习,天天向上";  DWORD dwSize;  //获取长度  dwSize = WideCharToMultiByte(CP\_OEMCP, 0, szWC, -1, NULL, 0, NULL, NULL);  char\* str = new char[dwSize];  WideCharToMultiByte(CP\_OEMCP, 0, szWC, -1, str, dwSize, NULL, NULL);  printf("宽字符转化为多字节的结果:\n");  printf("%s\n",str);  delete[] str;  str = NULL;  }  void TestMultiBytesToWideChar()  {  char str[] = "恭喜发财,红包拿来";  //char\* cstr = "恭喜发财,红包拿来";//错误:不能够用const char\* 的值来初始化char\* 实体  //char\* cstr = str; //传递地址是可以的  DWORD dwSize;  //获取长度  dwSize = MultiByteToWideChar(CP\_ACP,0,str,-1,NULL,0);  WCHAR\* wStr = new WCHAR[dwSize];  //转换  MultiByteToWideChar(CP\_ACP, 0, str, -1, wStr, dwSize);  wprintf\_s(L"多字节转化为宽字符的结果:\n");  wprintf\_s(L"%s\n",wStr);  delete[] wStr;  wStr = NULL;  }  int main()  {  setlocale(LC\_ALL, "");  //TestWideCharToMuiltiBytes();//OK  TestMultiBytesToWideChar();  return 0;  } |

### 注意:不能够用const char\* 来初始化char \*,同理,不能用const wchar\_t\* 来初始化wchar\_t\*

|  |
| --- |
|  |

### 需要改为数组的形式

|  |
| --- |
|  |

### 但是,下面的写法是0k的

|  |
| --- |
|  |

### 说明:把字符数组的地址传递给char \*变量是可以的,但是把它他的值传递给char\* 变量却是不可以的

# 扩展：可以学习window磁盘API并且练习这三个版本