1. UNICODE字符，什么是UNICODE字符，为什么要有UNICODE字符？？
2. 字符编码：

**UTF-8：**一个字节一个字符，有些字符是2个字节，有的字符是3个字节，还有的字符是4个字节。

**UTF-16：**大部分字符都是2个字节。

**UTF-32：**所有字符都是4个字节。

1. 字符的数据类型：

UNICODE类型的字符都是宽字符，在C中的数据类型是：wchar\_t。在VC2010中，这是一个基本数据类型！！！

一个宏，可以兼容UNICODE和ASCII字符，这个宏就是TEXT，数据类型用TCHAR！！！

1. 在windows中，有UNICODE类型的函数和ASCII类型的函数，例如CreateWindowEx函数。

在WinUser.h中，有如下定义：

#ifdef UNICODE

#define CreateWindowEx CreateWindowExW

#else

#define CreateWindowEx CreateWindowExA

#endif // !UNICODE

在windows98的时候，

CreateWindowW()

{

UNICODE\_TO\_ASCII()

return CreateWindowA()

}

在windows2000以后呢，正好相反：

CreateWindowA()

{

ASCII\_TO\_UNICODE ()

return CreateWindowW()

}

也就是，在我们的程序中，如果定义了UNICODE，那么我们使用的是宽字符，否则，使用的是ASCII类型的函数，所有我们统一使用CreateWindowEx函数，让系统区分是否使用宽字符或者ASCII字符类型的函数。这样程序的兼容性更好。

创建自己的动态链接库的时候，可以使用这种方法！！！

1. 一些C运行库函数，没有自己的ASCII版和UNICODE版，如何区分。

例如strlen：

|  |  |  |  |
| --- | --- | --- | --- |
| **TCHAR.H routine** | **\_UNICODE & \_MBCS not defined** | **\_MBCS defined** | **\_UNICODE defined** |
| **\_tcslen** | **strlen** | **strlen** | **wcslen** |
| **\_tcsclen** | **strlen** | **\_mbslen** | **wcslen** |
| **\_tcsclen\_l** | **strlen\_l** | **\_mbslen\_l** | **wcslen\_l** |

通过MSDN，或者VC2010的帮助，可以获得对应的TCHAR.h版函数，我们尽量使用TCHAR.h中的函数，这样我们就可以不考虑我们当前程序是否是UNICODE或者ASCII版。

1. 我们编程的时候，尽量使用安全字符串，例如strcpy就是一个非安全函数，当你再程序中使用这个函数的时候，你就会发现，编译器会出现警告，同时给出建议，请遵守。

编译器会提示我们使用strcpy\_s函数，此时，我们可以查找这个函数，并找到这个函数的TCHAR.h版。

|  |  |  |  |
| --- | --- | --- | --- |
| **TCHAR.H routine** | **\_UNICODE & \_MBCS not defined** | **\_MBCS defined** | **\_UNICODE defined** |
| **\_tcscpy\_s** | **strcpy\_s** | **\_mbscpy\_s** | **wcscpy\_s** |

1. 几个有用的StringSafe.h函数：

**HRESULT StringCchCat(**

**\_\_inout LPTSTR pszDest,**

**\_\_in size\_t cchDest,**

**\_\_in LPCTSTR pszSrc**

**);**

pszDest：目标字符串；

cchDest：目标字符串内存大小；

pszSrc： 源字符串。

这是一个字符串拷贝函数，在strsafe.h中。

**HRESULT StringCchPrintf(**

**\_\_out LPTSTR pszDest,**

**\_\_in size\_t cchDest,//目标内存大小**

**\_\_in LPCTSTR pszFormat,**

**\_\_in ...**

**);**

这个函数和printf函数用法类似，只是将字符写入了pszDest内存块中，而不是屏幕上！

这两个函数如果成功的话，返回：S\_OK，

STRSAFE\_E\_INVALID\_PARAMETER：传入了无效参数。

STRSAFE\_E\_INSUFFICIENF\_BUFFER：目标字符串内存的大小不够。

1. 两个字符串比较函数：

**int CompareString(**

**\_\_in LCID Locale,**

**\_\_in DWORD dwCmpFlags,**

**\_\_in LPCTSTR lpString1,**

**\_\_in int cchCount1,**

**\_\_in LPCTSTR lpString2,**

**\_\_in int cchCount2**

**);**

**int CompareStringOrdinal(**

**\_\_in LPCWSTR lpString1,**

**\_\_in int cchCount1,**

**\_\_in LPCWSTR lpString2,**

**\_\_in int cchCount2,**

**\_\_in BOOL bIgnoreCase**

**);**

第二个和语言无关，速度更快！！！建议使用！！！

1. 款字符和ASCII字符之间的转换函数！！

**int MultiByteToWideChar(**

**\_\_in UINT CodePage,**

**\_\_in DWORD dwFlags,**

**\_\_in LPCSTR lpMultiByteStr,**

**\_\_in int cbMultiByte,**

**\_\_out LPWSTR lpWideCharStr,**

**\_\_in int cchWideChar**

**);**

**int WideCharToMultiByte(**

**\_\_in UINT CodePage,**

**\_\_in DWORD dwFlags,**

**\_\_in LPCWSTR lpWideCharStr,**

**\_\_in int cchWideChar,**

**\_\_out LPSTR lpMultiByteStr,**

**\_\_in int cbMultiByte,**

**\_\_in LPCSTR lpDefaultChar,**

**\_\_out LPBOOL lpUsedDefaultChar**

**);**

**BOOL IsTextUnicode(**

**\_\_in const VOID \*lpv,**

**\_\_in int iSize,**

**\_\_inout LPINT lpiResult**

**);**

**内核对象：**

内核对象的一般特性，而不是那个内核对象的特性！！！！

首先，内核对象是一个比较抽象的概念！！

1. 什么是内核对象？

令牌对象token

时间对象Event

文件对象File

文件映射对象Map\_File

线程对象Thread

时钟对象Timer

线程池对象ThreadPool

I/o完成端口对象，Completion port

工作对象Job

邮槽对象mailslot

互斥对象Mutex

管道对象pipe

进程对象process

信号灯对象semaphore

1. 每个内核对象，其实就是一块内存块，这个内存块位于操作系统的内科地址空间