# 学习大纲

|  |
| --- |
|  |

## 1.什么是wchar\_t?

### 就是宽字符，他的api和char的api版本不一样

函数介绍：  
**wcslen**()   类似与char\*类型作参数的strlen()函数      用来获取wchar\_t\*变量的长度（不包含终结符）  
**wcscpy**()  类似与char\*类型作参数的strcpy()函数      用来进行wchar\_t\*变量之间的复制  
同样的还有  wcsncpy()  
**wcscmp**()  类似与char\*类型的strcmp()函数      用来对比两个wchar\_t\*变量的大小  
**wmemset**()  类似与char\*类型的memset()函数     用来初始化内存  
**wprintf**()  类似与char\*类型的printf()函数        用来输出wchar\_t\*字符串  
  
  
**简单记忆**：  
一般把char类型函数中的" str"替换为" wcs "即可  
wcs：wide char string

## 2宽字符处理函数函数与普通函数对照表

### 1)字符分类： 宽字符函数普通C函数描述

#### iswalnum（） isalnum（） 测试字符是否为数字或字母

#### iswalpha（） isalpha（） 测试字符是否是字母

#### iswcntrl（） iscntrl（） 测试字符是否是控制符

#### iswdigit（） isdigit（） 测试字符是否为数字

#### iswgraph（） isgraph（） 测试字符是否是可见字符

#### iswlower（） islower（） 测试字符是否是小写字符

#### iswprint（） isprint（） 测试字符是否是可打印字符

#### iswpunct（） ispunct（） 测试字符是否是标点符号

#### iswspace（） isspace（） 测试字符是否是空白符号

#### iswupper（） isupper（） 测试字符是否是大写字符

#### iswxdigit（） isxdigit（）测试字符是否是十六进制的数字

### 2)大小写转换：

#### 宽字符函数普通C函数描述

#### towlower（） tolower（） 把字符转换为小写

#### towupper（） toupper（） 把字符转换为大写

### 2.2)字符比较： 宽字符函数普通C函数描述

#### wcscoll（） strcoll（） 比较字符串

### 3)日期和时间转换： 宽字符函数描述

#### strftime（） 根据指定的字符串格式和locale设置格式化日期和时间

#### wcsftime（） 根据指定的字符串格式和locale设置格式化日期和时间， 并返回宽字符串

#### strptime（） 根据指定格式把字符串转换为时间值， 是strftime的反过程

### 4)打印和扫描字符串： 宽字符函数描述

#### fprintf（）/fwprintf（） 使用vararg参量的格式化输出

#### fscanf（）/fwscanf（） 格式化读入

#### printf（）/wprintf() 使用vararg参量的格式化输出到标准输出

#### scanf（）/ wscanf从标准输入的格式化读入

#### sprintf（）/swprintf（） 根据vararg参量表格式化成字符串

#### sscanf（）/ swscanf() 以字符串作格式化读入

#### vfprintf（）/vfwprintf（） 使用stdarg参量表格式化输出到文件

#### vprintf（）/ vwprintf() 使用stdarg参量表格式化输出到标准输出

#### vsprintf（）/vswprintf（） 格式化stdarg参量表并写到字符串

### 5)数字转换：

#### 宽字符函数普通C函数描述

#### wcstod（） strtod（）   把宽字符的初始部分转换为双精度浮点数

#### wcstol（） strtol（）   把宽字符的初始部分转换为长整数

#### wcstoul（） strtoul（） 把宽字符的初始部分转换为无符号长整数

### 6)多字节字符和宽字符转换及操作：

#### 宽字符函数描述

#### mblen（） 根据locale的设置确定字符的字节数

#### mbstowcs（） 把多字节字符串转换为宽字符串

#### mbtowc（）/btowc（）把多字节字符转换为宽字符

#### wcstombs（） 把宽字符串转换为多字节字符串

#### wctomb（）/wctob（） 把宽字符转换为多字节字符

### 7)输入和输出：

#### 宽字符函数普通C函数描述

#### fgetwc（） fgetc（） 从流中读入一个字符并转换为宽字符

#### fgetws（） fgets（） 从流中读入一个字符串并转换为宽字符串

#### fputwc（） fputc（） 把宽字符转换为多字节字符并且输出到标准输出

#### fputws（） fputs（） 把宽字符串转换为多字节字符并且输出到标准输出串

#### getwc（） getc（） 从标准输入中读取字符， 并且转换为宽字符

#### getwchar（） getchar（） 从标准输入中读取字符， 并且转换为宽字符

#### None gets（） 使用fgetws（）

#### putwc（） putc（） 把宽字符转换成多字节字符并且写到标准输出

#### putwchar（） putchar（） 把宽字符转换成多字节字符并且写到标准输出

#### None puts（） 使用fputws（）

#### ungetwc（） ungetc（） 把一个宽字符放回到输入流中

### 8)字符串操作：

#### 宽字符函数普通C函数描述

#### wcscat（） strcat（） 把一个字符串接到另一个字符串的尾部

#### wcsncat（） strncat（） 类似于wcscat（）， 而且指定粘接字符串的粘接长度.

#### wcschr（） strchr（） 查找子字符串的第一个位置

#### wcsrchr（） strrchr（） 从尾部开始查找子字符串出现的第一个位置

#### wcspbrk（） strpbrk（） 从一字符字符串中查找另一字符串中任何一个字符第一次出现的位置

#### wcswcs（）/wcsstr（） strchr（） 在一字符串中查找另一字符串第一次出现的位置

#### wcscspn（） strcspn（） 返回不包含第二个字符串的的初始数目

#### wcsspn（） strspn（） 返回包含第二个字符串的初始数目

#### wcscpy（） strcpy（） 拷贝字符串

#### wcsncpy（） strncpy（） 类似于wcscpy（）， 同时指定拷贝的数目

#### wcscmp（） strcmp（） 比较两个宽字符串

#### wcsncmp（） strncmp（） 类似于wcscmp（）， 还要指定比较字符字符串的数目

#### wcslen（） strlen（） 获得宽字符串的数目

#### wcstok（） strtok（） 根据标示符把宽字符串分解成一系列字符串

#### wcswidth（） None 获得宽字符串的宽度

#### wcwidth（） None 获得宽字符的宽度

### 9)另外还有对应于memory操作

#### wmemcpy（）， wmemchr（）， wmemcmp（）， wmemmove（）， wmemset（）．

## 3.项目演练

## 1.新建一个控制台项目，取名：Lesson9-wchar\_t，然后新建一个源文件取名wchar\_t.cpp,代码如下

|  |
| --- |
| #include<stdio.h>  #include<stdlib.h>  #include <Locale.h>  int main()  {  /\*\*  在c语言中，char类型的变量只能用来保存英文字母，保存中文字是不行的  如果想用wprintf输出中文，需要包含Locale.h头文件，还需要调用这个函数setlocale(LC\_ALL, "chs");  \*/  // setlocale(LC\_ALL, "chs");  setlocale(LC\_ALL, ""); //这样写也行  wchar\_t\* zh = L"我是中国人";  wprintf(L"zh:%s\n",zh);  wchar\_t c = L'爱';  wprintf(L"%c\n",c);  //字符数组  wchar\_t wsz[] = L"伟大的中华人民共和国";  wprintf(L"wsz:%s\n",wsz);  system("pause");  return 0;  } |

### 别看只有短短几行代码，非常坑爹，想用wprintf输出中文，需要包含Locale.h头文件，还需要调用这个函数setlocale(LC\_ALL, "chs");也就是是c语言默认不能显示中文，显示中文必须设置为地点字符集

### 效果：

|  |
| --- |
|  |

## 2.然后我们来学习一下wcslen函数，需要包含string.h头文件

|  |
| --- |
| #include<stdio.h>  #include<stdlib.h>  #include <Locale.h>  #include<string.h>  int main()  {  /\*\*  在c语言中，char类型的变量只能用来保存英文字母，保存中文字是不行的  如果想用wprintf输出中文，需要包含Locale.h头文件，还需要调用这个函数setlocale(LC\_ALL, "chs");  \*/  // setlocale(LC\_ALL, "chs");  setlocale(LC\_ALL, ""); //这样写也行  wchar\_t\* zh = L"我是中国人";  wprintf(L"zh:%s\n",zh);  wchar\_t c = L'爱';  wprintf(L"%c\n",c);  //字符数组  wchar\_t wsz[] = L"伟大的中华人民共和国";  wprintf(L"wsz:%s\n",wsz); //length of wsz:10  //获取宽字符串的长度  wprintf(L"length of wsz:%d\n",wcslen(wsz));  system("pause");  return 0;  } |

### 效果：

|  |
| --- |
|  |

## 3.我们来学习一下宽字符串比较函数wcscmp

|  |
| --- |
| #include<stdio.h>  #include<stdlib.h>  #include <Locale.h>  #include<string.h>  int main()  {  /\*\*  在c语言中，char类型的变量只能用来保存英文字母，保存中文字是不行的  如果想用wprintf输出中文，需要包含Locale.h头文件，还需要调用这个函数setlocale(LC\_ALL, "chs");  \*/  // setlocale(LC\_ALL, "chs");  setlocale(LC\_ALL, ""); //这样写也行  wchar\_t\* zh = L"我是中国人";  wprintf(L"zh:%s\n",zh);  wchar\_t c = L'爱';  wprintf(L"%c\n",c);  wprintf(L"length of zh:%d\n",wcslen(zh));  //字符数组  wchar\_t wsz[] = L"伟大的中华人民共和国";  wprintf(L"wsz:%s\n",wsz); //length of wsz:10  //获取宽字符串的长度  wprintf(L"length of wsz:%d\n",wcslen(wsz));  //宽字符串比较  int ret = wcscmp(zh,wsz);  if(0 == ret)  {  printf("zh is equal to wsz\n");  }  else if(ret<0)  {  printf("zh is shorter than wsz\n");  }  else  {  printf("zh is shorter than wsz\n");  }  system("pause");  return 0;  } |

### 效果：

|  |
| --- |
|  |

## 4.宽字符串拼接

|  |
| --- |
| #include<stdio.h>  #include<stdlib.h>  #include <Locale.h>  #include<string.h>  int main()  {  /\*\*  在c语言中，char类型的变量只能用来保存英文字母，保存中文字是不行的  如果想用wprintf输出中文，需要包含Locale.h头文件，还需要调用这个函数setlocale(LC\_ALL, "chs");  \*/  // 1.setlocale(LC\_ALL, "chs");  setlocale(LC\_ALL, ""); //这样写也行  wchar\_t\* zh = L"我是中国人";  wprintf(L"zh:%s\n",zh);  wchar\_t c = L'爱';  wprintf(L"%c\n",c);  wprintf(L"length of zh:%d\n",wcslen(zh));  //2.字符数组  wchar\_t wsz[] = L"伟大的中华人民共和国";  wprintf(L"wsz:%s\n",wsz); //length of wsz:10  //3.获取宽字符串的长度  wprintf(L"length of wsz:%d\n",wcslen(wsz));  //4.宽字符串比较  int ret = wcscmp(zh,wsz);  if(0 == ret)  {  printf("zh is equal to wsz\n");  }  else if(ret<0)  {  printf("zh is shorter than wsz\n");  }  else  {  printf("zh is shorter than wsz\n");  }  //5.宽字符串拼接  wchar\_t wcs1[40] = L"我爱你";  wchar\_t wcs2[40] = L"中国";  wcscat\_s(wcs1,wcs2);  wprintf(L"wcs1+wcs2=%s\n",wcs1);  system("pause");  return 0;  } |

### 效果：

|  |
| --- |
|  |

### 注意：不要把中文字符串赋值给普通的ansi字符串，这样子不好，你在获取字符串的长度时会得到错误的结果。应该吧它赋值给宽字符串

## 5.字符串复制wcscpy

|  |
| --- |
| #include<stdio.h>  #include<stdlib.h>  #include <Locale.h>  #include<string.h>  int main()  {  /\*\*  在c语言中，char类型的变量只能用来保存英文字母，保存中文字是不行的  如果想用wprintf输出中文，需要包含Locale.h头文件，还需要调用这个函数setlocale(LC\_ALL, "chs");  \*/  // 1.setlocale(LC\_ALL, "chs");  setlocale(LC\_ALL, ""); //这样写也行  wchar\_t\* zh = L"我是中国人";  wprintf(L"zh:%s\n",zh);  wchar\_t c = L'爱';  wprintf(L"%c\n",c);  wprintf(L"length of zh:%d\n",wcslen(zh));  //2.字符数组  wchar\_t wsz[] = L"伟大的中华人民共和国";  wprintf(L"wsz:%s\n",wsz); //length of wsz:10  //3.获取宽字符串的长度  wprintf(L"length of wsz:%d\n",wcslen(wsz));  //4.宽字符串比较  int ret = wcscmp(zh,wsz);  if(0 == ret)  {  printf("zh is equal to wsz\n");  }  else if(ret<0)  {  printf("zh is shorter than wsz\n");  }  else  {  printf("zh is shorter than wsz\n");  }  //5.宽字符串拼接  wchar\_t wcs1[40] = L"我爱你";  wchar\_t wcs2[40] = L"中国";  wcscat\_s(wcs1,wcs2);  wprintf(L"wcs1+wcs2=%s\n",wcs1);  //6.宽字符串复制  wchar\_t src[] =L"大家好，欢迎学习c语言";  wchar\_t dst[40] =L"";  wcscpy(dst,src);  wprintf(L"copy result=%s\n",dst);  system("pause");  return 0;  } |

### 效果：

|  |
| --- |
|  |

# 这一节就学习到这里当然，还有许多宽字符函数，需要慢慢学习和练习