西北工业大学
NORTHWESTERN POLYTECHNICAL UNIVERSITY

C程序设计

Programming in C

1011014

主讲：姜学锋，计算机学院

构建数据类型体系

1、用户自定义类型

2、类型名字体系

8.9 用户自定义类型

- 用户自定义类型主要是通过构造类型实现的。构造类型中的数组能够实现同一类型的多个实体，体现了自定义类型时量的需求；构造类型中的结构体能够实现不同类型的数据组合，体现了自定义类型时质的需求。而且C语言构造类型是可以递归声明的，即数组元素可以是结构体和数组，结构体成员也可以是结构体和数组，无限嵌套组合的结果使C语言具有表示任意复杂的数据类型的能力。

8.9 用户自定义类型

- 在开发应用程序和软件设计过程中，经常会自定义新的数据类型，甚至会建立起一整套适应开发需要的数据类型体系。

- 当程序中有大量自定义数据类型时，规范化的类型命名是建立类型体系的核心任务之一，typedef是实现这个核心任务的重要工具。

8.9 用户自定义类型

- 可以用typedef声明一个新类型名来代替已有类型名，其形式为：

```
typedef 已有类型名 新类型名;
```

- 其中已有类型名必须是已存在的数据类型的名称，新类型名是标识符序列，习惯上用大写标识；如果是多个新类型名，用逗号（，）作为间隔。最后以分号（；）结束。

8.9 用户自定义类型

- 例如：

```
typedef unsigned char BYTE;//按计算机汇编指令习惯规定的字节型
typedef unsigned short WORD;//按计算机汇编指令习惯规定的字类型
typedef unsigned long DWORD;//按计算机汇编指令习惯规定的双字类型
```

- BYTE即是unsigned char（基本数据类型）的新类型名。

```
unsigned char a,b,c; //定义无符号字符型
BYTE a,b,c; //定义字节型
```

8.9 用户自定义类型

- typedef是存储类别关键字，因此不能与auto、static、register、extern同时使用。例如：

```
typedef auto int INTEGER; //错误，不能在声明中有多个存储类别关键字
typedef static float REAL; //错误，不能在声明中有多个存储类别关键字
typedef extern int COUNT; //错误，不能在声明中有多个存储类别关键字
```

8.9 用户自定义类型

- 而且typedef允许递归声明，即将前一次typedef得到的新类型当作已有类型。例如：

```
typedef unsigned int UINT; //无符号整型
typedef UINT WPARAM; //32位消息参数类型
```

- 可以用typedef一次声明多个新类型名。例如：

```
typedef int UINT,BOOL; //无符号整型、逻辑型
```

8.9 用户自定义类型

- 使用typedef声明一个新类型名的方法是：
- ①先按变量定义的方法写出定义形式，如（char *s1,*s2），表示字符串；
- ②将变量名换成新类型名，如（char *LPSTR, *PSTR）；
- ③最后在前面加上typedef，如（typedef char *LPSTR, *PSTR），即得字符串类型。

8.9 用户自定义类型

- 下面针对不同数据类型列举一些typedef形式

```
①基本数据类型
    typedef 基本数据类型名 新类型名; //新类型名为基本数据类型
②数组类型
    typedef 元素类型 新类型名[常量]; //新类型名为一维数组类型
    typedef 元素类型 新类型名[常量1][常量2]; //新类型名为二维数组类型
    typedef char 新类型名[常量]; //新类型名为字符串类型
    typedef char 新类型名[常量1][常量2]; //新类型名为字符串数组类型
```

8.9 用户自定义类型

- 下面针对不同数据类型列举一些typedef形式

③指针类型

```
typedef 指向类型 *新类型名; //新类型名为指针类型
typedef 指向类型 *新类型名[常量]; //新类型名为指针数组类型
typedef 指向类型 (*新类型名)[常量]; //新类型名为数组指针类型
typedef char *新类型名; //新类型名为字符串类型
typedef char *新类型名[常量]; //新类型名为字符串数组类型
typedef 指向类型 **新类型名; //新类型名为指针的指针类型
```

8.9 用户自定义类型

- 下面针对不同数据类型列举一些typedef形式

```
④函数类型
    typedef 函数类型 (新类型名)(形式参数列表);//新类型名为函数类型
    typedef 函数类型 (*新类型名)(形式参数列表);//新类型名为函数指针类型
⑤结构体、共用体、枚举类型
    typedef struct/union/enum 旧类型名 新类型名;
    //新类型名为结构体/共用体/枚举类型
    typedef struct/union 结构体类型名 {
        成员列表
    } 新类型名,*新类型名,新类型名[10];
    //分别为结构体，结构体指针，结构体数组类型
```

8.9 用户自定义类型

- 例如下面是Win32应用程序接口数据类型体系中的部分类型声明：

```
typedef long LONG; //32位有符号长整型
typedef LONG LPARAM, LRESULT;
//32位消息参数类型,有符号消息处理结果类型
typedef DWORD COLORREF; //颜色值类型
typedef unsigned __int64 ULONGLONG; //无符号64位整型类型
```

8.9 用户自定义类型

- 例如下面是Win32应用程序接口数据类型体系中的部分类型声明：

```
typedef const char *LPCSTR, *PCSTR; //常字符串类型
typedef void *HANDLE; //句柄类型（实为指针类型）
typedef struct tagPOINT { //已有类型是结构体类型
    LONG  x, y;
} POINT,*PPOINT,POINTS[10]; //POINT结构体，PPOINT结构体指针，
POINTS结构体数组
typedef int (CALLBACK *PROC)(); //回调函数指针类型
typedef LRESULT (CALLBACK* WNDPROC)(HWND,UINT,WPARAM,LPARA
M); //消息处理函数类型
```

8.9 用户自定义类型

- 使用typedef重新命名一个类型名的好处是：
- ①声明易于记忆的类型名或适应具体要求的类型名，如用COLORREF能让人见其名知其意，比起unsigned long更靠近应用；
- ②使变量或对象定义变得更直观，如LPSTR a比char *a更让人容易理解；

8.9 用户自定义类型

- ③为复杂的类型声明取一个简单的别名，得到原声明的简化版
- 如用WNDPROC掩饰了LRESULT (CALLBACK*)(HWND,UINT,WPARAM,LPARAM)的复杂性；
- ④声明与平台无关的类型，如GCC的无符号64位整型类型为unsigned long long，VC的无符号64位整型类型为unsigned __int64，都统一为ULONGLONG最方便跨平台。

- ⑤简化旧的C语言标准中的结构体类型名（struct 结构体名），这一点在最新版本的C/C++标准中因为已经简化为（结构体名）而过时。

8.9 用户自定义类型

- 需要注意typedef和#define两者的区别。例如：

```
typedef char *STRING1;
#define STRING2 char *
STRING1 s1, s2;
//s1和s2实际为char *类型
STRING2 s3, s4;
//s3实际为char *类型，s4实际为char类型，因为STRING2替换为char *
```

- 显然，就类型重命名来说，使用typedef要比#define要好。

8.9 用户自定义类型

- 通常，在建立应用程序数据类型体系时，习惯将所有typedef都写在一个头文件中，凡是需要用到新类型名的源程序只要用#include包含这个头文件即可。

8.9.1 构建数据类型体系

- 一般地，大规模程序都会构建独立的数据类型体系，用来规范、统一程序中的数据类型，好处是：
  - （1）程序中的数据类型是规范的，规模开发易于控制、实施；
  - （2）程序中的数据类型是统一的，程序版本易于维护、升级；
  - （3）程序中的数据类型是标准的，应用平台易于兼容、移植；

8.9.1 构建数据类型体系

- 1. SQL数据库引擎数据类型体系

8.9.1 构建数据类型体系

表14-1 SQL常用数据类型

ODBC数据类型	SQL语言数据类型	ODBC数据类型	SQL语言数据类型
SQL_CHAR	CHAR(n)	SQL_VARCHAR	VARCHAR(n)
SQL_LONGVARCHAR	LONG VARCHAR	SQL_DECIMAL	DECIMAL(p,s)
SQL_NUMERIC	NUMERIC(p,s)	SQL_SMALLINT	SMALLINT
SQL_INTEGER	INTEGER	SQL_REAL	REAL
SQL_FLOAT	FLOAT(p)	SQL_DOUBLE	DOUBLE PRECISION
SQL_BIT	BIT	SQL_TINYINT	TINYINT
SQL_BIGINT	BIGINT	SQL_BINARY	BINARY(n)
SQL_VARBINARY	VARBINARY(n)	SQL_LONGVARBINARY	LONG VARBINARY
SQL_TYPE_DATE	DATE	SQL_TYPE_TIME	TIME(p)
SQL_TYPE_TIMESTAMP	TIMESTAMP(p)		

8.9.1 构建数据类型体系

表14-2 ODBC的C语言类型

C类型标识	ODBC数据类型定义	C类型标识	ODBC数据类型定义
SQL_C_CHAR	SQLCHAR*	SQL_C_SSHORT	SQLSMALLINT
SQL_C_USHORT	SQLUSMALLINT	SQL_C_SLONG	SQLINTEGER
SQL_C_ULONG	SQLUINTEGER	SQL_C_FLOAT	SQLREAL
SQL_C_BIT	SQLCHAR	SQL_C_DOUBLE	SQLDOUBLE, SQLFLOAT
SQL_C_STINYINT	SQLSCHAR	SQL_C_TYPE_TIME	SQL_TIME_STRUCT
SQL_C_SBIGINT	SQLBIGINT	SQL_C_NUMERIC	SQL_NUMERIC_STRUCT
SQL_C_BINARY	SQLCHAR*	SQL_C_TYPE_DATE	SQL_DATE_STRUCT
SQL_C_UBIGINT	SQLUBIGINT	SQL_C_TYPE_TIMESTAMP	SQL_TIMESTAMP_STRUCT
SQL_C_UTINYINT	SQLCHAR		

8.9.1 构建数据类型体系

- 2．Windows数据类型体系Windows Data Types
- Win32应用程序接口数据类型

8.9.1 构建数据类型体系

表14-3 Windows常用名称前缀

前缀	数据类型	前缀	数据类型	前缀	数据类型
c/chr	char/TCHAR 字符型	n	short 短整型	i	int 整型
x,y	x和y坐标	cx,cy	x和y方向长度	b	BOOL 逻辑型
f	int，代表flag	w	WORD 字型	l	LONG 长整型
dw	DWORD 双字型	fn	function函数	s	string 字符串
sz	“\0”结尾字符串	h	句柄	p,lp	指针/长指针
by	BYTE 字节型	lpfn	指向函数指针	cb	字节数
lpsz	“\0”结尾字符串指针	g_	全局变量	c_	常量
m_	类数据成员	s_	静态变量	rc	矩形

8.9.1 构建数据类型体系

表14-4 Windows常用数据类型

Windows数据类型	C语言类型	含义
APIENTRY	#define APIENTRY WINAPI	系统函数调用约定
ATOM	typedef WORD ATOM;	原子类型
BOOL	typedef int BOOL;	逻辑型（可以是TRUE或者FALSE）
BOOLEAN	typedef BYTE BOOLEAN;	布尔型（可以是TRUE或者FALSE）
BYTE	typedef unsigned char BYTE;	字节型（8位）
CALLBACK	#define CALLBACK __stdcall	回调函数调用约定
CCHAR	typedef char CCHAR;	8位Windows ANSI字符
CHAR	typedef char CHAR;	8位Windows ANSI字符
COLORREF	typedef DWORD COLORREF;	32位RGB颜色数据值
CONST	#define CONST const	const限定

8.9.1 构建数据类型体系

续表14-4 Windows常用数据类型

Windows数据类型	C语言类型	含义
DWORD	typedef unsigned long DWORD;	32位无符号整型0~4294967295
DWORDLONG	typedef unsigned __int64 DWORDLONG;	64位无符号整型 0~18446744073709551615
DWORD_PTR	typedef ULONG_PTR DWORD_PTR;	无符号长整型指针
DWORD32	typedef unsigned int DWORD32;	32位无符号整型
DWORD64	typedef unsigned __int64 DWORD64;	64位无符号整型
FLOAT	typedef float FLOAT;	浮点型

8.9.1 构建数据类型体系

续表14-4 Windows常用数据类型

Windows数据类型	C语言类型	含义
HACCEL	typedef HANDLE HACCEL;	快捷键列表句柄
HALF_PTR	#ifdef _WIN64 typedef int HALF_PTR; #else typedef short HALF_PTR; #endif	指针大小的一半
HANDLE	typedef PVOID HANDLE;	通用对象句柄
HBITMAP	typedef HANDLE HBITMAP;	位图句柄
HBRUSH	typedef HANDLE HBRUSH;	画刷句柄

8.9.1 构建数据类型体系

续表14-4 Windows常用数据类型

Windows数据类型	C语言类型	含义
HCOLORSPACE	typedef HANDLE HCOLORSPACE;	颜色空间句柄
HCONV	typedef HANDLE HCONV;	DDE转换句柄
HCONVLIST	typedef HANDLE HCONVLIST;	DDE转换列表句柄
HCURSOR	typedef HICON HCURSOR;	光标句柄
HDC	typedef HANDLE HDC;	设备场境（DC）句柄
HDDEDATA	typedef HANDLE HDDEDATA;	DDE数据句柄
HDESK	typedef HANDLE HDESK;	桌面句柄
HDROP	typedef HANDLE HDROP;	内降结构句柄

8.9.1 构建数据类型体系

续表14-4 Windows常用数据类型

Windows数据类型	C语言类型	含义
HDWP	typedef HANDLE HDWP;	窗口位置结构句柄
HENHMETAFILE	typedef HANDLE HENHMETAFILE;	增强的metafile文件资源句柄
HFILE	typedef int HFILE;	用OpenFile打开的文件句柄
HFONT	typedef HANDLE HFONT;	字体句柄
HGDIOBJ	typedef HANDLE HGDIOBJ;	GDI对象句柄
HGLOBAL	typedef HANDLE HGLOBAL;	全局内存块句柄
HHOOK	typedef HANDLE HHOOK;	钩子句柄
HICON	typedef HANDLE HICON;	图标句柄

8.9.1 构建数据类型体系

续表14-4 Windows常用数据类型

Windows数据类型	C语言类型	含义
HINSTANCE	typedef HANDLE HINSTANCE;	应用实例句柄
HKEY	typedef HANDLE HKEY;	注册表句柄
HKL	typedef HANDLE HKL;	输入区域设置标识符
HLOCAL	typedef HANDLE HLOCAL;	局部内存块句柄
HMENU	typedef HANDLE HMENU;	菜单资源句柄
HMETAFILE	typedef HANDLE HMETAFILE;	metafile文件资源句柄
HMODULE	typedef HINSTANCE HMODULE;	进程模块句柄

8.9.1 构建数据类型体系

续表14-4 Windows常用数据类型

Windows数据类型	C语言类型	含义
HMONITOR	typedef HANDLE HMONITOR;	显示器句柄
HPALETTE	typedef HANDLE HPALETTE;	调色板句柄
HPEN	typedef HANDLE HPEN;	画笔句柄
HRESULT	typedef LONG HRESULT;	COM接口返回值（FAILED和SUCCEEDED）
HRGN	typedef HANDLE HRGN;	区域句柄
HRSRC	typedef HANDLE HRSRC;	资源句柄
HSZ	typedef HANDLE HSZ;	DDE字符串句柄
HWINSTA	typedef HANDLE WINSTA;	窗口站句柄

8.9.1 构建数据类型体系

续表14-4 Windows常用数据类型

Windows数据类型	C语言类型	含义
HWND	typedef HANDLE HWND;	窗口句柄
INT	typedef int INT;	32位有符号整型-2147483648~2147483647
INT_PTR	#if defined(_WIN64) typedef __int64 INT_PTR; #else typedef int INT_PTR; #endif	有符号整型指针
INT8	typedef signed char INT8;	8位有符号整型
INT16	typedef signed short INT16;	16位有符号整型

8.9.1 构建数据类型体系

续表14-4 Windows常用数据类型

Windows数据类型	C语言类型	含义
INT32	typedef signed int INT32;	32位有符号整型-2147483648~2147483647
INT64	typedef signed __int64 INT64;	64位有符号整型–9223372036854775808~9223372036854775807
LANGID	typedef WORD LANGID;	语言标识
LCID	typedef DWORD LCID;	本地标识
LCTYPE	typedef DWORD LCTYPE;	本地信息类型
LGRPID	typedef DWORD LGRPID;	语言组标识
LONG	typedef long LONG;	32位有符号整型-2147483648~2147483647

8.9.1 构建数据类型体系

续表14-4 Windows常用数据类型

Windows数据类型	C语言类型	含义
LONGLONG	#if !defined(_M_IX86) typedef __int64 LONGLONG; #else typedef double LONGLONG; #endif	64位有符号整型– 9223372036854775808~ 9223372036854775807
LONG_PTR	#if defined(_WIN64) typedef __int64 LONG_PTR; #else typedef long LONG_PTR; #endif	有符号长整型指针

8.9.1 构建数据类型体系

续表14-4 Windows常用数据类型

Windows数据类型	C语言类型	含义
LONG32	typedef signed int LONG32;	32位有符号整型-2147483648~2147483647
LONG64	typedef __int64 LONG64;	64位有符号整型–9223372036854775808~9223372036854775807
LPARAM	typedef LONG_PTR LPARAM;	消息参数
LPBOOL	typedef BOOL far *LPBOOL;	BOOL指针
LPBYTE	typedef BYTE far *LPBYTE;	BYTE指针
LPCOLORREF	typedef DWORD *LPCOLORREF;	COLORREF指针

8.9.1 构建数据类型体系

续表14-4 Windows常用数据类型

Windows数据类型	C语言类型	含义
LPCSTR	typedef __nullterminated CONST CHAR *LPCSTR;	“\0”结尾字符串（8位ANSI字符）const指针
LPCTSTR	#ifdef UNICODE typedef LPCWSTR LPCTSTR; #else typedef LPCSTR LPCTSTR; #endif	带编码的字符串const指针
LPCVOID	typedef CONST void *LPCVOID;	任意类型const指针
LPCWSTR	typedef CONST WCHAR *LPCWSTR;	“\0”结尾字符串（16位Unicode字符）const指针

8.9.1 构建数据类型体系

续表14-4 Windows常用数据类型

Windows数据类型	C语言类型	含义
LPDWORD	typedef DWORD *LPDWORD;	DWORD指针
LPHANDLE	typedef HANDLE *LPHANDLE;	句柄指针
LPINT	typedef int *LPINT;	INT指针
LPLONG	typedef long *LPLONG;	LONG指针
LPSTR	typedef CHAR *LPSTR;	“\0”结尾字符串（8位ANSI字符）指针

8.9.1 构建数据类型体系

续表14-4 Windows常用数据类型

Windows数据类型	C语言类型	含义
LPTSTR	#ifdef UNICODE typedef LPWSTR LPTSTR; #else typedef LPSTR LPTSTR; #endif	带编码的字符串指针
LPVOID	typedef void *LPVOID;	任意类型指针
LPWORD	typedef WORD *LPWORD;	WORD指针
LPWSTR	typedef WCHAR *LPWSTR;	“\0”结尾字符串（16位Unicode字符）指针
LRESULT	typedef LONG_PTR LRESULT;	消息处理有符号返回值

8.9.1 构建数据类型体系

续表14-4 Windows常用数据类型

Windows数据类型	C语言类型	含义
PBOOL	typedef BOOL *PBOOL;	BOOL指针
PBOOLEAN	typedef BOOLEAN *PBOOLEAN;	BOOLEAN指针
PBYTE	typedef BYTE *PBYTE;	BYTE指针
PCHAR	typedef CHAR *PCHAR;	CHAR指针
PCSTR	typedef CONST CHAR *PCSTR;	“\0”结尾字符串（8位ANSI字符）指针

8.9.1 构建数据类型体系

续表14-4 Windows常用数据类型

Windows数据类型	C语言类型	含义
PCTSTR	#ifdef UNICODE typedef LPCWSTR PCTSTR; #else typedef LPCSTR PCTSTR; #endif	带编码的字符串指针
PCWSTR	typedef CONST WCHAR *PCWSTR;	“\0”结尾字符串（16位Unicode字符）指针
PDWORD	typedef DWORD *PDWORD;	DWORD指针
PDWORDLONG	typedef DWORDLONG *PDWORDLONG;	DWORDLONG指针

8.9.1 构建数据类型体系

续表14-4 Windows常用数据类型

Windows数据类型	C语言类型	含义
PDWORD_PTR	typedef DWORD_PTR *PDWORD_PTR;	DWORD_PTR指针
PDWORD32	typedef DWORD32 *PDWORD32;	DWORD32指针
PDWORD64	typedef DWORD64 *PDWORD64;	DWORD64指针
PFLOAT	typedef FLOAT *PFLOAT;	FLOAT指针

8.9.1 构建数据类型体系

续表14-4 Windows常用数据类型

Windows数据类型	C语言类型	含义
PHALF_PTR	#ifdef _WIN64 typedef HALF_PTR *PHALF_PTR; #else typedef HALF_PTR *PHALF_PTR; #endif	HALF_PTR指针
PHANDLE	typedef HANDLE *PHANDLE;	HANDLE指针
PHKEY	typedef HKEY *PHKEY;	HKEY指针
PINT	typedef int *PINT;	INT指针

8.9.1 构建数据类型体系

续表14-4 Windows常用数据类型

Windows数据类型	C语言类型	含义
PINT_PTR	typedef INT_PTR *PINT_PTR;	INT_PTR指针
PINT8	typedef INT8 *PINT8;	INT8指针
PINT16	typedef INT16 *PINT16;	INT16指针
PINT32	typedef INT32 *PINT32;	INT32指针
PINT64	typedef INT64 *PINT64;	INT64指针
PLCID	typedef PDWORD PLCID;	LCID指针
PLONG	typedef LONG *PLONG;	LONG指针
PLONGLONG	typedef LONGLONG *PLONGLONG;	LONGLONG指针

8.9.1 构建数据类型体系

续表14-4 Windows常用数据类型

Windows数据类型	C语言类型	含义
PLONG_PTR	typedef LONG_PTR *PLONG_PTR;	LONG_PTR指针
PLONG32	typedef LONG32 *PLONG32;	LONG32指针
PLONG64	typedef LONG64 *PLONG64;	LONG64指针
POINTER_32	#if defined(_WIN64) #define POINTER_32 __ptr32 #else #define POINTER_32 #endif	32位指针

8.9.1 构建数据类型体系

续表14-4 Windows常用数据类型

Windows数据类型	C语言类型	含义
POINTER_64	#if (_MSC_VER >= 1300) #define POINTER_64 __ptr64 #else #define POINTER_64 #endif	64位指针
POINTER_SIGNED	#define POINTER_SIGNED __sptr	signed指针
POINTER_UNSIGNED	#define POINTER_UNSIGNED __uptr	unsigned指针
PSHORT	typedef SHORT *PSHORT;	SHORT指针

8.9.1 构建数据类型体系

续表14-4 Windows常用数据类型

Windows数据类型	C语言类型	含义
PSIZE_T	typedef SIZE_T *PSIZE_T;	SIZE_T指针
PSSIZE_T	typedef SSIZE_T *PSSIZE_T;	SSIZE_T指针
PSTR	typedef CHAR *PSTR;	“\0”结尾字符串（8位ANSI字符）指针
PTBYTE	typedef TBYTE *PTBYTE;	TBYTE指针
PTCHAR	typedef TCHAR *PTCHAR;	TCHAR指针

8.9.1 构建数据类型体系

续表14-4 Windows常用数据类型

Windows数据类型	C语言类型	含义
PTSTR	#ifdef UNICODE typedef LPWSTR PTSTR; #else typedef LPSTR PTSTR; #endif	带编码的字符串指针
PUCHAR	typedef UCHAR *PUCHAR;	UCHAR指针

8.9.1 构建数据类型体系

续表14-4 Windows常用数据类型

Windows数据类型	C语言类型	含义
PUHALF_PTR	#ifdef _WIN64 typedef UHALF_PTR *PUHALF_PTR; #else typedef UHALF_PTR *PUHALF_PTR; #endif	UHALF_PTR指针
PUINT	typedef UINT *PUINT;	UINT指针
PUINT_PTR	typedef UINT_PTR *PUINT_PTR;	UINT_PTR指针
PUINT8	typedef UINT8 *PUINT8;	UINT8指针

8.9.1 构建数据类型体系

续表14-4 Windows常用数据类型

Windows数据类型	C语言类型	含义
PUINT16	typedef UINT16 *PUINT16;	UINT16指针
PUINT32	typedef UINT32 *PUINT32;	UINT32指针
PUINT64	typedef UINT64 *PUINT64;	UINT64指针
PULONG	typedef ULONG *PULONG;	ULONG指针
PULONGLONG	typedef ULONGLONG *PULONGLONG;	ULONGLONG指针
PULONG_PTR	typedef ULONG_PTR *PULONG_PTR;	ULONG_PTR指针
PULONG32	typedef ULONG32 *PULONG32;	ULONG32指针

8.9.1 构建数据类型体系

续表14-4 Windows常用数据类型

Windows数据类型	C语言类型	含义
PULONG64	typedef ULONG64 *PULONG64;	ULONG64指针
PUSHORT	typedef USHORT *PUSHORT;	USHORT指针
PVOID	typedef void *PVOID;	任意类型指针
PWCHAR	typedef WCHAR *PWCHAR;	WCHAR指针
PWORD	typedef WORD *PWORD;	WORD指针
PWSTR	typedef WCHAR *PWSTR;	“\0”结尾字符串（16位Unicode字符）指针
QWORD	typedef unsigned __int64 QWORD;	64位无符号整型

8.9.1 构建数据类型体系

续表14-4 Windows常用数据类型

Windows数据类型	C语言类型	含义
SC_HANDLE	typedef HANDLE SC_HANDLE;	服务控制管理器数据库句柄
SC_LOCK	typedef LPVOID SC_LOCK;	服务控制管理器数据库锁
SERVICE_STATUS_HANDLE	typedef HANDLE SERVICE_STATUS_HANDLE;	服务状态值句柄
SHORT	typedef short SHORT;	16位整型–32768~32767
SIZE_T	typedef ULONG_PTR SIZE_T;	指针最大字节数
SSIZE_T	typedef LONG_PTR SSIZE_T;	有符号SIZE_T

8.9.1 构建数据类型体系

续表14-4 Windows常用数据类型

Windows数据类型	C语言类型	含义
TBYTE	#ifdef UNICODE typedef WCHAR TBYTE; #else typedef unsigned char TBYTE; #endif	带编码的BYTE

8.9.1 构建数据类型体系

续表14-4 Windows常用数据类型

Windows数据类型	C语言类型	含义
TCHAR	#ifdef UNICODE typedef WCHAR TCHAR; #else typedef char TCHAR; #endif	带编码的CHAR
UCHAR	typedef unsigned char UCHAR;	无符号CHAR

8.9.1 构建数据类型体系

续表14-4 Windows常用数据类型

Windows数据类型	C语言类型	含义
UHALF_PTR	#ifdef _WIN64 typedef unsigned int UHALF_PTR; #else typedef unsigned short UHALF_PTR; #endif	无符号HALF_PTR
UINT	typedef unsigned int UINT;	无符号INT型

8.9.1 构建数据类型体系

续表14-4 Windows常用数据类型

Windows数据类型	C语言类型	含义
UINT_PTR	#if defined(_WIN64) typedef unsigned __int64 UINT_PTR; #else typedef unsigned int UINT_PTR; #endif	无符号INT_PTR型
UINT8	typedef unsigned char UINT8;	无符号INT8型
UINT16	typedef unsigned short UINT16;	无符号INT16型
UINT32	typedef unsigned int UINT32;	无符号INT32型0~4294967295

8.9.1 构建数据类型体系

续表14-4 Windows常用数据类型

Windows数据类型	C语言类型	含义
UINT64	typedef usigned __int 64 UINT64;	无符号INT64型 0~18446744073709551615
ULONG	typedef unsigned long ULONG;	无符号LONG型0~4294967295
ULONGLONG	#if !defined(_M_IX86) typedef unsigned __int64 ULONGLONG; #else typedef double ULONGLONG; #endif	64位无符号整型 0~18446744073709551615

8.9.1 构建数据类型体系

续表14-4 Windows常用数据类型

Windows数据类型	C语言类型	含义
ULONG_PTR	#if defined(_WIN64) typedef unsigned __int64 ULONG_PTR; #else typedef unsigned long ULONG_PTR; #endif	LONG_PTR
ULONG32	typedef unsigned int ULONG32;	无符号LONG32型0~4294967295
ULONG64	typedef unsigned __int64 ULONG64;	无符号LONG64型 0~18446744073709551615

8.9.1 构建数据类型体系

续表14-4 Windows常用数据类型

Windows数据类型	C语言类型	含义
UNICODE_STRING	typedef struct _UNICODE_STRING { USHORT Length; USHORT MaximumLength; PWSTR Buffer; } UNICODE_STRING;	Unicode字符串
USHORT	typedef unsigned short USHORT;	无符号SHORT型0~65535
USN	typedef LONGLONG USN;	USN
VOID	#define VOID void	任意类型（空类型）

8.9.1 构建数据类型体系

续表14-4 Windows常用数据类型

Windows数据类型	C语言类型	含义
WCHAR	typedef wchar_t WCHAR;	16位Unicode字符
WINAPI	#define WINAPI __stdcall	系统函数调用约定
WORD	typedef unsigned short WORD;	16位无符号整型0~65535
WPARAM	typedef UINT_PTR WPARAM;	消息参数


程序设计