按下一个键就会产生一条WM\_KEYDOWN 和 WM\_SYSKEYDOWN消息，释放一个键则会产生一条WM\_KEYUP 和 WM\_SYSKEYUP消息，同样也会被放到队列中。

击键消息的wParam参数含有一个被按过或被释放键的虚键码，

击键消息的lParam参数含有产生消息的击键的附加信息，这个信息包括重复计数、扫描码、扩展键标志、关联码、前一次键状态标志以及转换状态标志。

重复记数

通过检查重复计数以确定某条击键消息是否是多次击键．

扫描码

它是用来标识键被按过的一个设备相关值．

扩展键标志

扩展键包括右边的ALT和CTRL键，数字键盘左边的一组键：INS、DEL、HOME、END、PAGE UP、PAGEDOWN和箭头键；还有数字键盘上的NUMLOCK键；BREAK(CTRL+PAUSE)键；PRINT SCRN键；除号键(／)和ENTER键。如果是一个扩展键，扩展键标志就被置位。

关联码

关联码标明在产生击键消息时是否按下从ALT键，如果按下ALT为1．否则为0。

前一次镊状态标志

前一次键状态标志指示某键在产生击键消息之前是否被按下．如果被按下则为1，否则为0。对于WM\_KEYUP和WM\_SYSKEYUP消息则总是0。

转换状态标志 ’

转换状态标志指明是按键还是释放键产生的击键消息，对WM\_KEYDOWN和WM\_SYSKEYDOWN消息这个标志总是0，对WM KEYUP和WM\_SYSKEYUP消息则总是1。

**字符消息**

字符消息在击键消息之后，一般捕获字符消息而不是击键消息

因为TranslateMessage函数从WM\_KEYDOWN和WM\_SYSKEYDOWN消息产生了字符消息，所以字符消息是夹在按键消息之间传递给窗口消息处理程序的。

|  |
| --- |
| 表6-10 |

|  |  |
| --- | --- |
| 消息 | 按键或者代码 |
| WM\_KEYDOWN | 「A」的虚拟键码（0x41） |
| WM\_CHAR | 「a」的字符代码（0x61） |
| WM\_KEYUP | 「A」的虚拟键码（0x41） |

|  |
| --- |
| 表6-11 |

|  |  |
| --- | --- |
| 消息 | 按键或者代码 |
| WM\_KEYDOWN | 虚拟键码VK\_SHIFT （0x10） |
| WM\_KEYDOWN | 「A」的虚拟键码（0x41） |
| WM\_CHAR | 「A」的字符代码（0x41） |
| WM\_KEYUP | 「A」的虚拟键码（0x41） |
| WM\_KEYUP | 虚拟键码VK\_SHIFT（0x10） |

Shift键本身不产生字符消息。

|  |
| --- |
| 表6-12 |

|  |  |
| --- | --- |
| 消息 | 按键或者代码 |
| WM\_KEYDOWN | 「A」的虚拟键码（0x41） |
| WM\_CHAR | 「a」的字符代码（0x61） |
| WM\_KEYDOWN | 「A」的虚拟键码（0x41） |
| WM\_CHAR | 「a」的字符代码（0x61） |
| WM\_KEYDOWN | 「A」的虚拟键码（0x41） |
| WM\_CHAR | 「a」的字符代码（0x61） |
| WM\_KEYDOWN | 「A」的虚拟键码（0x41） |
| WM\_CHAR | 「a」的字符代码（0x61） |
| WM\_KEYUP | 「A」的虚拟键码（0x41） |

如果某些WM\_KEYDOWN消息的重复计数大于1，那么相应的WM\_CHAR消息将具有同样的重复计数。

|  |
| --- |
| 表6-13 |

|  |  |  |  |
| --- | --- | --- | --- |
| 按键 | 字符代码 | 产生方法 | ANSI C控制字符 |
| Backspace | 0x08 | Ctrl-H | \b |
| Tab | 0x09 | Ctrl-I | \t |
| Ctrl-Enter | 0x0A | Ctrl-J | \n |
| Enter | 0x0D | Ctrl-M | \r |
| Esc | 0x1B | Ctrl-[ |  |

**处理控制字符**

case WM\_CHAR:

switch (wParam)

{

case '\b': // backspace

break ;

case '\t': // tab

break ;

case '\n': // linefeed

break ;

case '\r': // carriage return

break ;

default: // character codes

break ;

}

return 0 ;