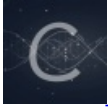


SetWindowLongPtr and GetWindowLongPtr 用法 实例

原创

ningto.com 于 2011-01-28 18:40:00 发布 8419 收藏 4
分类专栏: [windows](#) 文章标签: [null callback](#) [存储](#) [winapi struct application](#)
本文为博主原创文章, 转载请注明来源(泞途 - 泊客网 <https://ningto.com>)。
本文链接: <https://blog.csdn.net/tujiaw/article/details/6168193>
版权



[windows](#) 专栏收录该内容

46 篇文章 0 订阅
订阅专栏

SetWindowLongPtr与GetWindowLongPtr主要有两种用法:

- 第一种是: 改变指定窗口的属性;
- 第二种是: 设置一个值在额外存储空间的指定偏移位置。

下面所说的是第二种用法:

首先, 额外存储空间是指在设计一个窗口类时的cbWndExtra属性, 如:

```
wndclass.cbWndExtra = 2 *
```

用SetWindowLongPtr可以在这个空间的某个偏移位置设置一个值, 当然设置一个值用处不大, 通常在这里设置的是一个地址, 这个地址可以是某个结构体的地址, 这样存储的数据就多了, 如:

```
//GWL_EXTRAMORY 额外内存里
```

用GetWindowLongPtr获取上面所设置的值, 如:

```
hExtra = (HGLOBAL)GetWin
```

下面简单例子是展示它们的用法, 但是并不能体现出真正的用处。

真正的用处是: 当分配的内存存在堆上时, 我们可以通过GetWindowLongPtr随时访问这块内存, 为了防止造成不必要的错误, 访问时最好加锁。

```
#include <windows.h>
```

注意: 偏移位置加上指针的字节数不能超过额外存储空间

如: 32位系统额外存储空间为8, 偏移位置位4, 再加上指针是4个字节, 这样正好; 如果是64位系统, 指针是8个字节, 所以一定要注意。

15:13:11