例子：在用w,a,s,d键控制一个矩形移动的同时，一个相同的矩形自动移动。

[复制代码](javascript:void(0);)

#include<windows.h>  
 LRESULT CALLBACK WinSunProc(HWND hwnd,UINT uMsg,WPARAM wParam,LPARAM lParam);//函数声明  
 VOID CALLBACK TimerProc(HWND hwnd,UINT uMsg,UINT\_PTR idEvent, DWORD dwTime);  
 int APIENTRY WinMain(HINSTANCE hInstance,  
 HINSTANCE hPrevInstance,  
 LPSTR lpCmdLine,  
 int nCmdShow)  
 {  
 WNDCLASS wndcls; //定义一个存储窗口信息WNDCLASS变量  
 wndcls.cbClsExtra=0; //默认为0  
 wndcls.cbWndExtra=0; //默认为0  
 wndcls.hbrBackground=(HBRUSH)GetStockObject(GRAY\_BRUSH); //背景画刷  
 wndcls.hCursor=LoadCursor(NULL,IDC\_ARROW); //光标  
 wndcls.hIcon=LoadIcon(NULL,IDI\_ERROR); //窗口图标  
 wndcls.hInstance=hInstance; //应用程序实例句柄由WinMain函数传进来   
 wndcls.lpfnWndProc=WinSunProc; //窗口消息处理函数  
 wndcls.lpszClassName="windowclass"; //窗口类名  
 wndcls.lpszMenuName=NULL; //窗口菜单名，没有菜单，为NULL  
 wndcls.style=CS\_HREDRAW | CS\_VREDRAW; //窗口类型，CS\_HREDRAW和CS\_VERDRAW 表明  
 //当窗口水平方向垂直方向的宽度变化时重绘整个窗口  
 RegisterClass(&wndcls); //把窗口信息提交给系统，注册窗口类  
 HWND hwnd; //用以存储CreateWindow函数所创建的窗口句柄  
 hwnd=CreateWindow("windowclass","first windows",   
 WS\_OVERLAPPEDWINDOW,0,0,600,400,NULL,NULL,hInstance,NULL); //创建窗口  
 ShowWindow(hwnd,SW\_SHOWNORMAL); //窗口创建完了，显示它  
 UpdateWindow(hwnd); //更新窗口，让窗口毫无延迟的显示  
 SetTimer(hwnd,1,200,(TIMERPROC)TimerProc); //设置定时器  
 MSG msg; //消息结构类型  
 while(GetMessage(&msg,NULL,0,0)) //获取消息  
 {  
 TranslateMessage(&msg); //此函数用于把键盘消息(WM\_KEYDOWN,WM\_KEYUP)转换成字符消息WM\_CHAR  
 DispatchMessage(&msg); //这个函数调用窗口过程处理函数，并把MSG里的信息处理后传给过程函数的四个参数  
 }  
 return 0;  
 }  
 VOID CALLBACK TimerProc(HWND hwnd,UINT uMsg,UINT\_PTR idEvent, DWORD dwTime) //定时器函数  
 {  
 static int x=0,y=0;  
 x+=15;  
 if(x>500)  
 {y+=15;x=0;}  
 HDC dc=GetDC(hwnd);  
 Rectangle(dc,x,y,x+30,y+30);  
 }  
 LRESULT CALLBACK WinSunProc(HWND hwnd,UINT uMsg,WPARAM wParam,LPARAM lParam)  
 { static int x=0,y=0;  
 switch(uMsg)  
 {  
 case WM\_CHAR:  
 {  
 if(wParam=='a') x-=10;  
 else if(wParam=='d') x+=10;  
 else if(wParam=='w') y-=10;  
 else if(wParam=='s') y+=10;  
 RECT rect;  
 GetClientRect(hwnd,&rect);  
 ::InvalidateRect(hwnd,&rect,TRUE); //发送WM\_PAINT消息，刷新窗口  
 }  
 break;  
 case WM\_PAINT:  
 HDC dc;   
 PAINTSTRUCT ps;   
 dc=BeginPaint(hwnd,&ps); //BeginPaint只能在响应WM\_PAINT,不能用GetDC获取设备上下文  
 Rectangle(dc,x,y,x+30,y+30);  
 break;  
 case WM\_CLOSE: //用户关闭了窗口  
 DestroyWindow(hwnd); //销毁窗口，并发送WM\_DESTROY消息  
 break;  
 case WM\_DESTROY: //如果窗口被销毁  
 PostQuitMessage(0); //让进程退出  
 break;  
 }  
 return DefWindowProc(hwnd,uMsg,wParam,lParam); //未处理的消息通过DefWindowProc函数交给系统处理  
 }