메시지를 보내는 스레드와 받는 스레드가 동일한 경우, 즉, 자신에게 메시지를 보내는 경우 SendMessage는 마치 함수 호출처럼 동작한다. 즉, WndProc을 호출한다.

하지만 메시지를 보내는 스레드와 받는 스레드가 다른 경우, 즉 다른 스레드로 메시지를 보낸다면 SendMessage는 좀 복잡하게 동작한다.

1. 메시지를 받는 스레드의 Send – message Queue에 메시지가 놓고, Wake Flags의 QS\_SENDMESSAGE flag를 Set한다.
2. 메시지를 받는 스레드가 다른 일을 하고 있었다면, 계속 그 일을 수행한다.(즉, 인터럽트 되지 않는다.)
3. 메시지를 받는 스레드가 메시지를 대기 하고 있는 상태가 된다면, Send-message Queue에 있는 모든 메시지가 차례로 처리되고 QS\_SENDMESSAGE Flag가 Reset된다.
4. 2,3의 작업동안 메시지를 보내는 스레드는 자신의 reply-message Queue에 Message가 놓인때 까지 대기한다.
5. 받는 메시지가 보낸 메시지를 다 처리한 후 return 하거나 ReplyMessage 를 호출하면 해당 값이 보낸 스레드의 Reply-Message Queue에 놓이게 된다.
6. Reply-Message Q에 값이 놓이면 해당 값은 SendMessage의 리턴값이 되고 메시지를 보낸 스레드를 다음 코드를 계속 실행하게 된다.
7. 비록 메시지를 보낸 스레드가 Reply-Message Q에 리턴값이 놓일 때 까지 idle 상태로 되어 있지만, 또 다른 스레드가 SendMessage()로 보낸 메시지는 즉시 처리할 수 있다