

# WinAPI

---

## 윈도우 프로그램 동작 과정

1. 윈도우 클래스 정의
  - 프로그램을 시작했을 때 윈도우의 여러가지 특성을 정의하는 윈도우 클래스라는 구조체를 생성한다.
  - 생성한 윈도우 클래스에 윈도우의 스타일이나 제목 등 특성들을 설정한다.
2. 윈도우 클래스 등록
  - 생성된 윈도우 클래스를 커널에 등록한다.
3. 윈도우 생성
  - 등록된 윈도우 클래스를 바탕으로 사용자에게 보여질 윈도우를 생성한다.
  - 생성된 윈도우의 핸들을 받는다.
4. 윈도우 출력
  - 윈도우 핸들을 사용하여 윈도우가 사용자에게 보여지도록 출력한다.
5. 메시지 루프
  - 생성된 윈도우가 발생한 이벤트는 커널에 의해 프로그램으로 전달된다.
  - 프로그램으로 전달된 메시지는 메시지 큐에 쌓인다.
  - 메시지 큐의 메시지를 하나씩 읽어들이며 메시지를 처리한다.

## 핸들이란?

리소스를 관리하는 운영체제가 리소스를 식별하기 위해 매기는 고유한 번호이다.

운영체제는 리소스를 안전하게 관리하기 위해 응용 프로그램 단위에선 직접적인 주소의 접근 대신 핸들이라는 식별자를 통해 리소스에 접근할 수 있게 한다.

## 메세지

사용자가 키보드 입력, 마우스 입력, 창 최소화 등 응용 프로그램과 상호작용하여 이벤트를 발생시키면 '메세지'라는 것이 발행된다.

그렇게 발행된 메세지는 커널에 의해 프로그램으로 전달되고 메세지를 받은 프로그램은 순서대로 메세지 큐에 저장한다.

GetMessage 혹은 PeekMessage를 사용하여 메세지 큐에 있는 메세지를 읽어와 처리할 수 있다.

## GetMessage

메세지 큐에서 메세지를 읽어오는 함수로 메세지 루프를 구현하는 데에 사용된다.

메세지 큐가 비어있는 경우엔 읽어올 메세지가 없어 대기 상태에 빠지는 데드 타임 현상이 발생한다.

때문에 이벤트를 발생시키지 않고선 다른 작업을 처리 할 수 없다는 특징이 있다.

또한 WM\_QUIT 메세지가 발행되지 않고선 GetMessage를 활용한 메세지 루프는 종료되지 않는다.

그렇기에 SetTimer 함수를 사용하여 일정 주기마다 WM\_TIMER라는 이벤트를 발생시킬 수 있다.

이를 통해 사용자의 입력과 상관 없이 작업을 처리할 수 있도록 구현할 수 있다.

하지만 SetTimer 주기는 10ms 미만으로 설정할 수 없다.

즉 초당 100회를 넘어선 WM\_TIMER의 메세지는 처리할 수 없다는 한계가 있다.

## PeekMessage

GetMessage와 SetTimer의 한계를 보완할 수 있는 함수로는 PeekMessage가 있다.

PeekMessage는 메세지 큐가 비어있다면 대기상태에 빠져버리던 GetMessage와 달리 메세지를 읽어왔다면 true, 메세지가 비어있다면 false를 반환하는 함수다.

이를 통해 이벤트가 발생하지 않아 메세지 루프가 멈추는 현상을 방지할 수 있다.

또한 초당 100회 까지만 사용자의 입력에 상관 없이 작업을 처리할 수 있었던 SetTimer와 달리 사용자 환경의 성능에 따라 응용 프로그램을 업데이트 할 수 있다.

하지만 PeekMessage를 사용한다면 WM\_QUIT 메세지에 대한 처리를 직접 해주어야 한다는 특징이 있다.