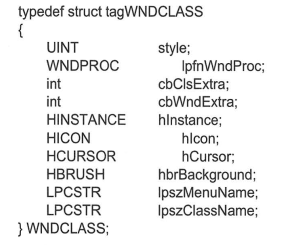
WinAPI 정복

1. 윈도우
   1. 특징 및 장점
      1. 그래픽 기반의 운영체제이다.
      2. 멀티 태스킹이 가능하다.
      3. 메시지 구동 시스템이다.
      4. 장치에 독립적이다.
      5. 일관성이 있다.
      6. 리소스가 분리되어있다.
   2. 개발툴
      1. API : 윈도우, DC, 메시지, 컨트롤, 리소스 등등의 핵심들로 구성되어있다.
      2. 여러가지 개발 방법
         1. SDK : SDK란 윈도우를 만든 MS에서 마이크로소프트사에서 제공하는 윈도우용 프로그램 개발 킷이다. 운영 체제가 제공하는 API를 사용하여 프로그램을 작성하며 가장 원론적이고 가장 먼저 발표된 개발 방법이다.
         2. 클래스 라이브러리 : 복잡하고 양이 많은 API 함수들을 좀더 쉽고 빠르게 사용하기 위해 클래스로 잘 포장하여 객체 위주로 프로그램을 작성하는 방법이다. MFC와 OWL가 있지만 MFC를 주로 사용한다.
         3. 비주얼 툴 : 손으로 일일이 코드를 작성하지 않고 마우스로 마치 그림을 그리듯 화면을 디자인 하며 꼭 필요한 곳에만 간단한 코드를 작성하는 개발툴이다.
   3. 먼저 알아야 할 것들
      * 1. 핸들이란 : 구체적인 어떤 대상에 붙여진 번호이며 문법적으로는 32 또는 64비트의 정수값이다. 윈도우에는 윈도우 핸들(hWnd)을 붙여 번호로 관리하며 DC에 대해서도 핸들을 사용한다.
        2. 핸들을 사용하는 이유 : 대상끼리의 구분을 위해서는 문자열보다 정수를 사용하는 것이 속도가 훨씬 더 빠르고 간편하기 때문이다.
        3. 핸들의 특징
           1. 운영체제가 발급하며 사용자는 쓰기만 한다. 사용자가 직접 핸들을 만드는 경우는 없다.
           2. 같은 종류의 핸들끼리는 절대로 중복된 값을 가지지 않는다. 다른 종류의 핸들끼리는 중복된 값을 가질 수도 있다.
           3. 핸들은 정수형이므로 값을 가지겠지만 그 실제 값이 무엇인지는 몰라도 상관없다.
2. 첫 번째 예제
   1. WinMain
      1. 헤더 파일 : windows.h를 include한다 윈도우에서는 하나의 헤더 파일에 모든 API 함수들의 원형과 사용하는 상수가 모두 정의되어 있어서 windows.h만 포함하면 된다. windows.h 헤더 파일은 기본적인 데이터 타입, 함수 원형, 매크로 상수 등을 정의하며 그 외 윈도우 프로그래밍에 필요한 보조 헤더 파일을 포함한다.
      2. 시작점 : 콘솔 프로그램과의 또다른 차이점은 프로그램의 시작점인 엔트리 포인트가 main이 아니라 WinMain이라는 점이다. 모든 윈도우 프로그램은 WinMain에서 실행을 시작한다. 콘솔의 main 함수는 인수 사용 여부에 따라 여러 원형이 있지만 WinMain의 원형은 위와 같이 고정되어 있다. WinMain은 4개의 인수를 취하는데 각 인수의 의미는 다음과 같다.
         1. hInstance : 프로그램의 인스턴스 핸들이다.
         2. hPrevInstance : 이전에 실행된 현재 프로그램의 인스턴스 핸들이다. 신경 X
         3. lpszCmdParam : 명령행으로 입력된 프로그램 인수이다. 콘솔 프로그램의 args인수에 해당하며, 보통 실행 직후에 열 파일의 경로가 전달된다.
         4. nCmdShow : 프로그램이 실행될 형태이며 최소화, 보통 모양 등이 전달된다.

hInstance 외에는 잘 사용되지 않는다. 인스턴스라는 말은 실행중인 프로그램 하나를 칭하는 용어이다. 윈도우는 여러 개의 프로그램이 동시에 실행되는 멀티태스킹 시스템이며 하나의 프로그램이 여러 번 실행될 수도 있다. 많은 API 함수들이 hInstance값을 인수로 요구하는데 그래서 이 예제에서는 WinMain의 인수로 전달된 hInstance값을 전역변수 g\_hInst에 대입하여 사용한다. hInstance 인수는 지역 변수이기 때문에 WinMain 밖에서는 사용할 수 없기 떄문이다.

* + 1. 메시지 처리 함수 : 메시지 처리 함수인 WndProc은 사용자와 시스템이 보내오는 메시지를 처리하는 일을 한다. 콘솔에서는 main 함수만으로도 프로그램을 작성할 수 있지만 윈도우에서는 WinMain과 WndProc 함수가 모두 있어야 한다. WinMain은 메인 윈도우를 만들고 화면에 윈도우를 표시하기만 할 뿐이며 대부분의 일은 WndProc에서 이루어진다. WinMain은 형태가 거의 고정된 반면 WndProc는 프로그램에 따라 형태가 천차만별이다.
  1. 윈도우 클래스
     1. 윈도우를 만들려면 윈도우 클래스를 먼저 등록한 후 CreateWindow 함수를 호출해야 한다. 윈도우 클래스는 만들어질 윈도우의 여러 가지 특성을 정의하는 구조체이며 모든 윈도우는 윈도우 클래스의 정보를 기반으로 만들어진다. 해당 구조체는 다음과 같다. 순서대로 윈도우 스타일, 메시지 처리 함수, 예약영역, 인스턴스, 아이콘, 커서, 배경 색상, 메뉴 속성, 클래스 이름이다. 생성된 윈도우 클래스는 RegisterClass를 통해 등록된다.
  2. 윈도우 생성
     1. 윈도우 클래스를 등록한 다음 이것을 기본으로 CreateWindow 함수를 사용하여 실제 윈도우를 생성한다.