

윈도우 운영체제의 역사



About..

유한대학교 컴퓨터소프트웨어공학과
김원일



목차



- 윈도우 운영체제
- 윈도우 이전 운영체제
- 윈도우 운영체제 역사
- Windows Desktop Application
- 기본 소스 분석
- 과제

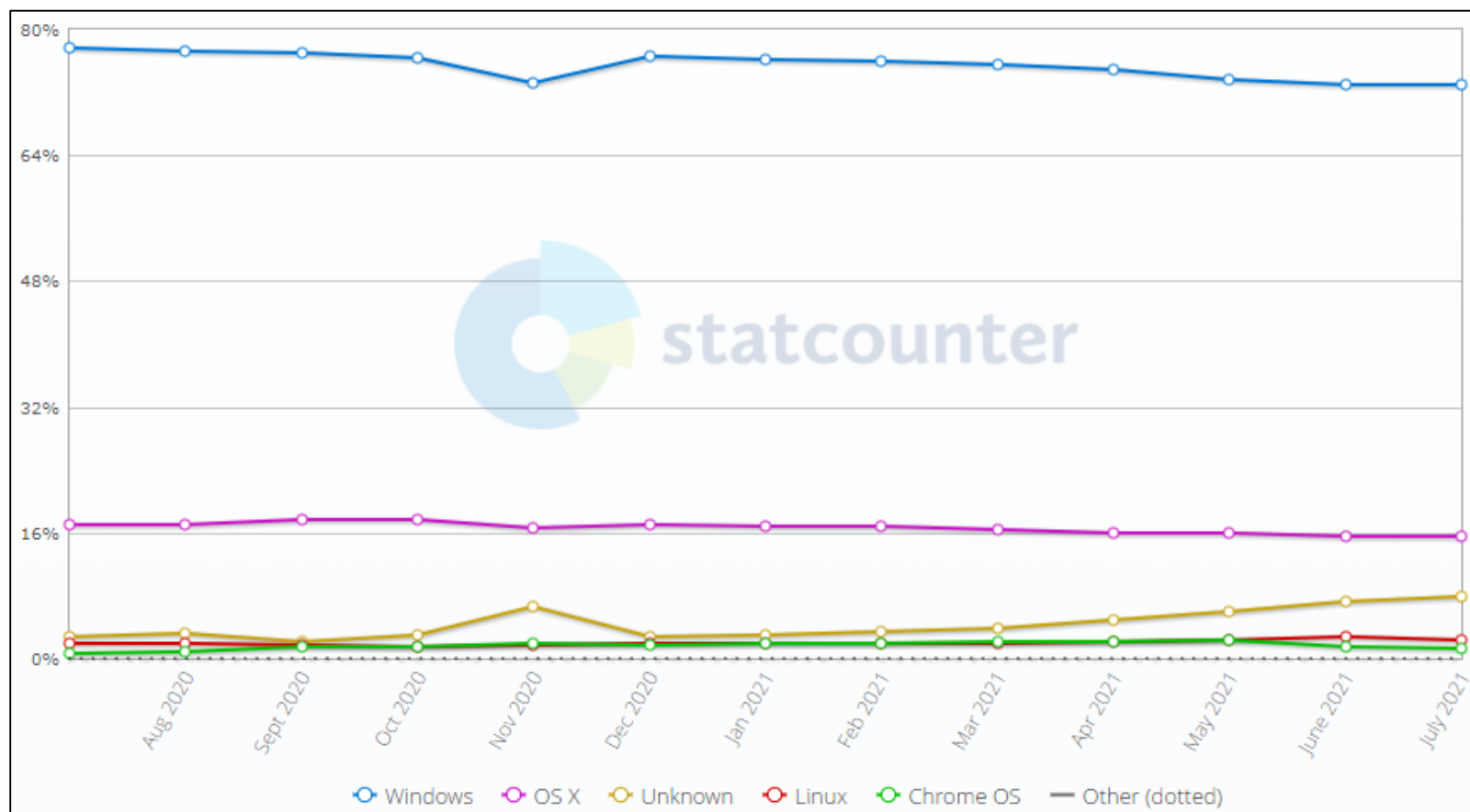


윈도우 운영체제 - 1



- Microsoft의 대표적인 PC 운영체제

- PC OS에서는 독보적인 지분을 차지
- 특수 용도를 제외하면 절대적으로 많이 사용되고 있는 운영체제

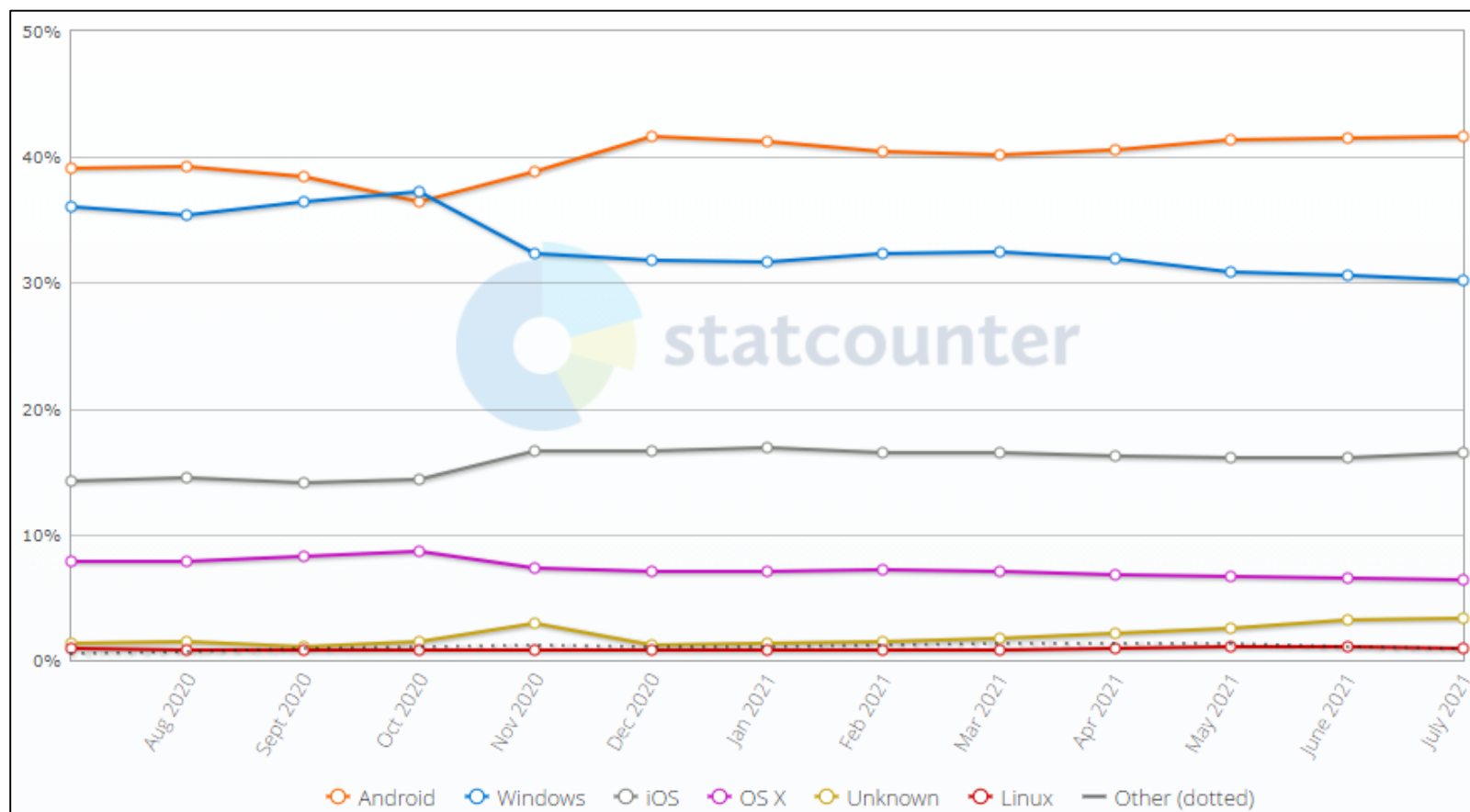




윈도우 운영체제 - 2



- 전체 OS에서도 매우 큰 지분을 가짐
 - 전체 OS에서는 안드로이드에 밀려 2위를 차지
 - PC가 없어도 핸드폰은 가진 사용자들이로 인해 모바일 OS가 우세함





윈도우 이전 운영체제 - 1



- UNIX
 - **POSIX 표준**을 통해 현대 운영체제의 원형이 된 운영체제
 - 벨 연구소에서 고급 언어인 **C언어**로 개발되어 현재까지 운용 중
- CP/M
 - MS-DOS 보다 먼저 개발된 인텔 8bit CPU용 운영체제
 - 마이크로 컴퓨터를 위한 **최초의 운영체제**

CP/M
PLUS

Copyright 1982,1983 Digital Research Inc.
Licensed to Tandy Corporation
Unbanked BIOS Copyright 1983 Tandy Corporation
Radio Shack Version 01.00.00 -- Serial Number 808F-000481

ENTER DATE (MM/DD/YY):

```
LS1: is LPT1:
A>stat
A: R/W, Space: 0k
B: R/W, Space: 56k

A>dir
A: CAT      COM : CONFIGIO  BAS : DOT      COM : BOOT      COM
A: MFT      COM : PATCH    COM : CPM60    COM : PIP      COM
A: STAT     COM : ASM      COM : AUTORUN  COM : LOAD      COM
A: COPY     COM : APDOS    COM : SUBMIT  COM : XSUB     COM
A: DUMP     COM : DUMP     COM : DOWNLOAD COM : MBASIC   COM
A: GBASIC   COM : ED      COM

A>dir b:
B: R      COM : READ      ME : CC      COM : CC2      COM
B: DEFF   CRL : DEFF2    CRL : CLINK  COM : CLIB    COM
B: BDSCIO H : HARDWARE  H : C      CCC

A>b:cc
BD Software C Compiler v1.50 (part I)
Usage:
cc <source_file> [-p] [-o] [-a <x>] [-d <x>] [-m <addr>] [-e <addr>] [-r <n>]
A>
```



윈도우 이전 운영체제 - 2



• 마이크로소프트의 최초 CUI 운영체제

- 빌 게이츠에 의해 개발된 **16bit CPU 기반 개인 PC 운영체제**
- UNIX와 같이 CUI(Command Line Interface)로 구성
- CUI는 명령어를 알아야만 사용할 수 있는 문제점
- "<https://github.com/microsoft/MS-DOS>"에 1.25와 2.0 버전 소스 공개
- IBM의 최대 실수

마이크로소프트 한글 MS-DOS 6.2 설치

마이크로소프트 한글 MS-DOS 6.2 설치 프로그램입니다.

설치 프로그램은 한글 MS-DOS 6.2를 컴퓨터에 설치합니다.

- 한글 MS-DOS를 설치하려면 Enter키를 누르십시오.
- 설치 프로그램에 대해서 알아보려면 F1키를 누르십시오.
- 한글 MS-DOS를 설치하지 않고 종료하려면 F3키를 누르십시오.

주의: 백업을 하려면 한글 MS-DOS를 설치하기 전에 하십시오.
파일을 백업하려면 F3키를 눌러 설치를 종료한 다음
백업 프로그램을 사용하여 파일을 백업합니다.

설치 프로그램을 계속하려면 Enter키를 누르십시오.

Enter=계속 F1=도움말 F3=종료 F5=색상 제거 F7=플로피디스크로 설치

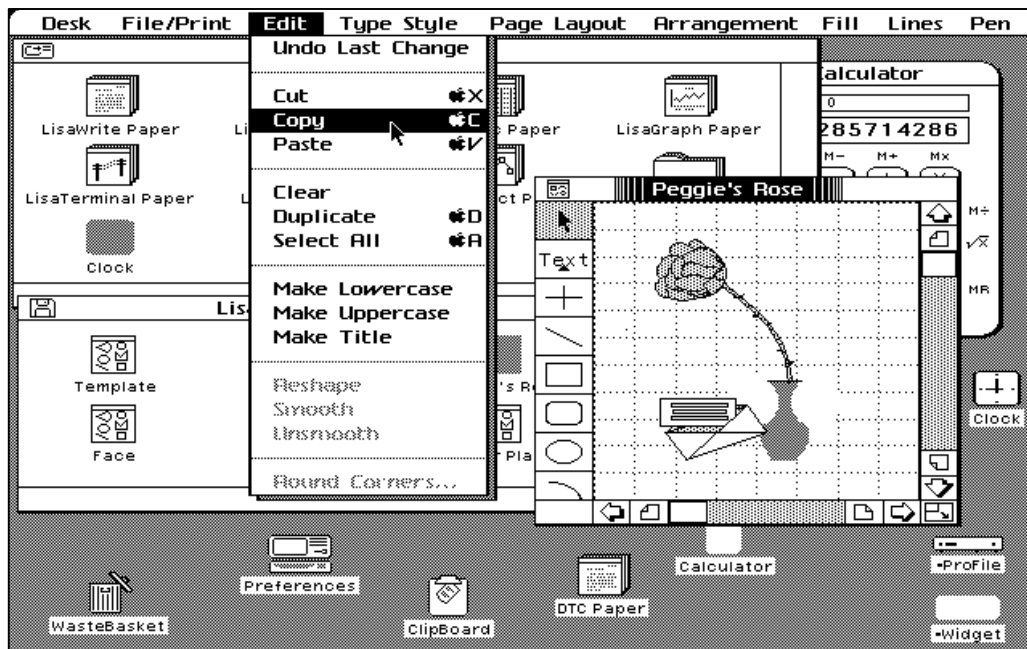
```
MBIOS 버전 1.20a.  
Copyright (C) Microsoft Corp 1993  
  
C:\>C:\DOS\SMARTDRV.EXE /X  
C:\>command  
  
Microsoft (R) 한글 MS-DOS(R) 버전 6.20  
(C)저작권자 (주)마이크로소프트 1981-1993  
  
C:\>dir  
  
드라이브 C의 이름은 MS-DOS_6  
일련 번호는 4707-0236  
디렉토리는 C:\>  
  
DOS             <DIR>             15-08-07      0:17  
COMMAND.COM     57,749 94-09-07      6:20  
MINA20          386      9,349 94-09-07      6:20  
CONFIG.SYS      169 15-08-07      0:19  
AUTOEXEC.BAT    78 15-08-07      0:19  
57개 파일      67,345 바이트  
2,130,444,288 바이트를 사용할 수 있습니다  
  
C:\>echo rtw4353_
```



윈도우 이전 운영체제 - 3

• GUI의 시작

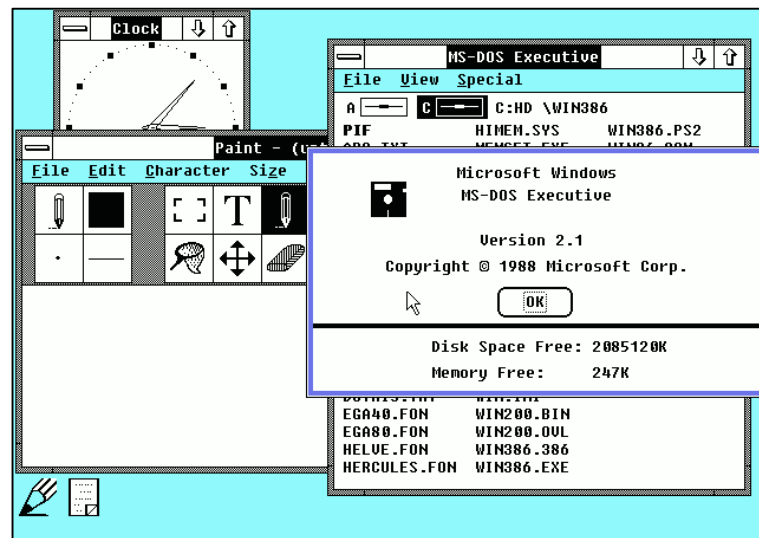
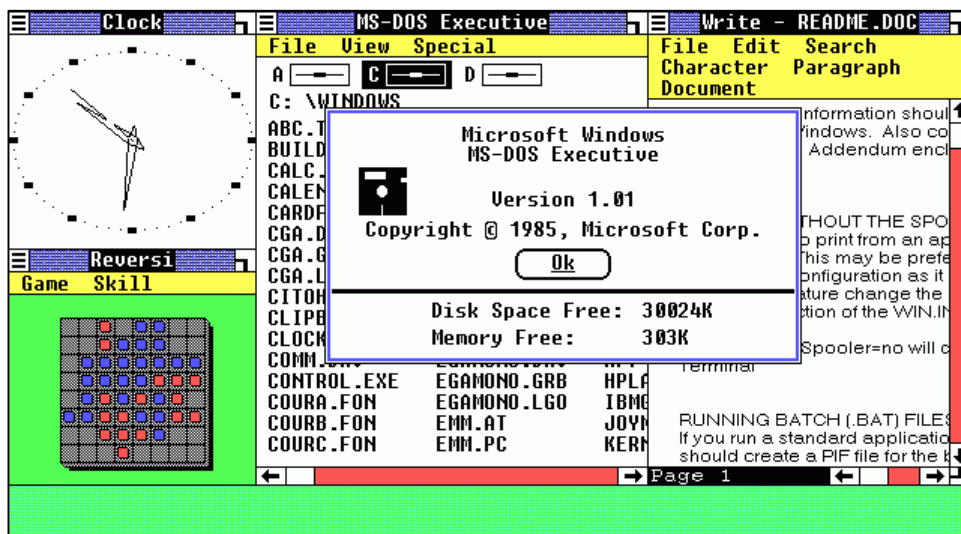
- Apple의 Lisa 운영체제가 GUI 환경의 시작을 알림
- 마우스를 이용한 최초의 개인 PC 운영체제
- 기존의 CUI를 대체하는 혁신적인 OS로 엄청난 가격을 자랑
- Microsoft가 GUI 운영체제를 가지고 싶게 한 비운의 OS





윈도우 운영체제 역사 - 1

- **우리도 한번 만들어 보자 Windows 1.0 ~ Windows 2.X**
 - 1.0 : 1985년 11월 20일에 출시된 MS 최초의 GUI 운영체제
 - 2.0 : 1987년 12월 9일에 출시, MS 워드와 엑셀이 구동되기 시작
 - 창 겹치기가 가능해짐
 - 이때까지만 해도 OS 실행도 되지 않는 경우가 많아 비난을 많이 받음

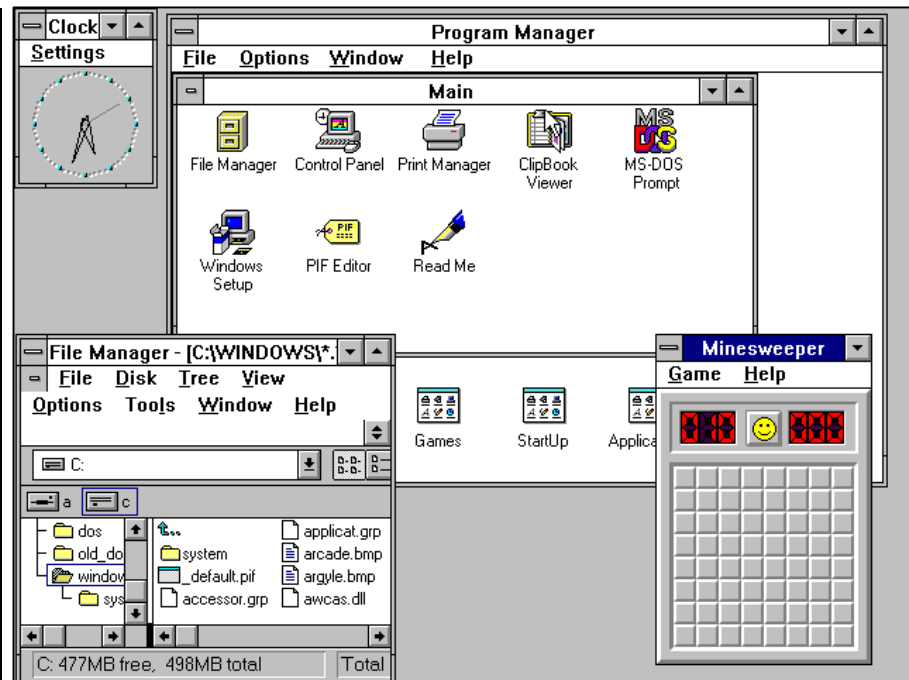
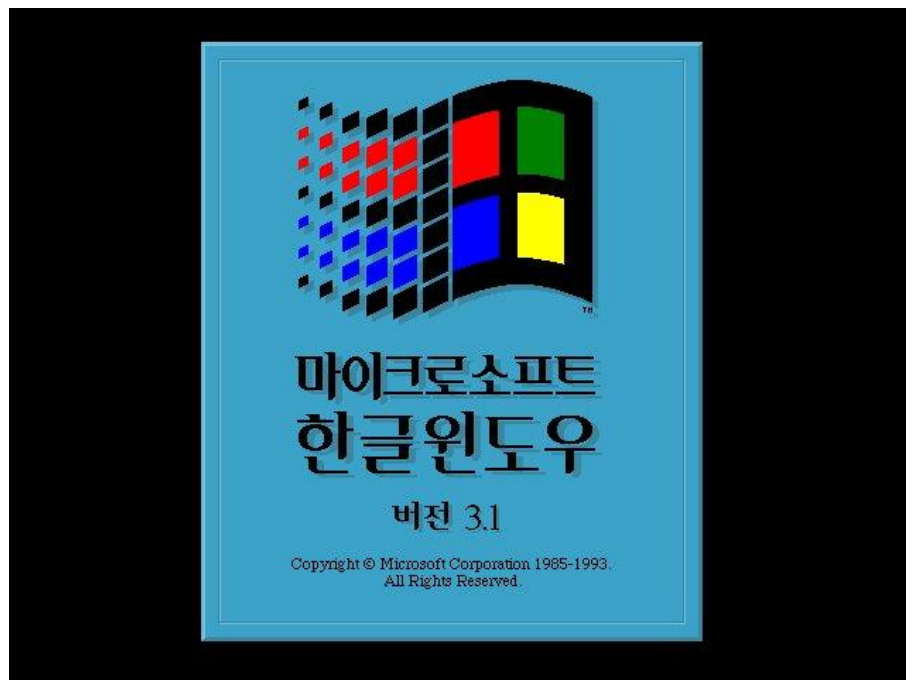




윈도우 운영체제 역사 - 2

• 이번엔 진짜야! Windows 3.0 ~ 3.X

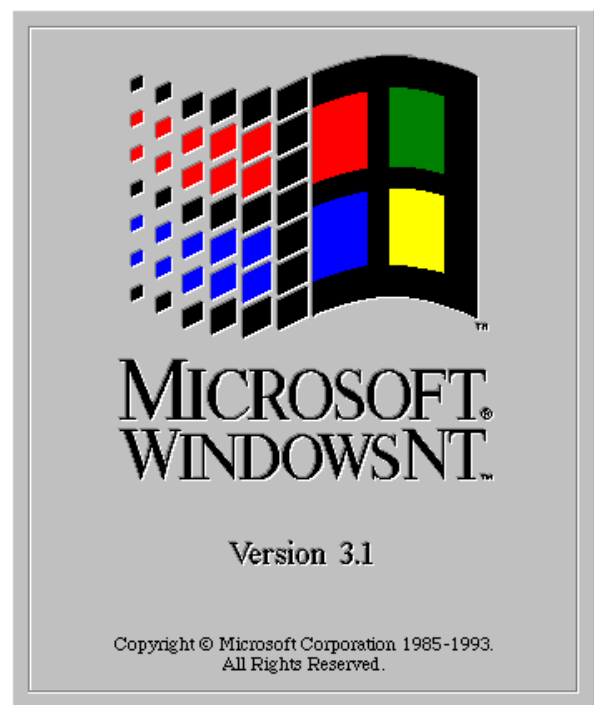
- 1990년 발매하여 윈도우 중 최초로 성공한 버전
- 1992년 4월 6일 발매한 3.1 버전부터 제대로 인정받기 시작
- 최초로 한글을 지원하기 시작한 버전
- Visual Basic, Visual C++, Win16 API 최초 도입





윈도우 운영체제 역사 - 3

- 기다려라 IBM! Windows NT(New Technology)의 시작
 - 1993년 7월 27일 3.1 버전이 처음으로 출시
 - Windows 3.1과 버전을 비슷하게 맞추기 위해 3.1부터 시작
 - IBM의 서버 운영체제에 대응하기 위해 개발 시작
 - 시작은 그야말로 미비



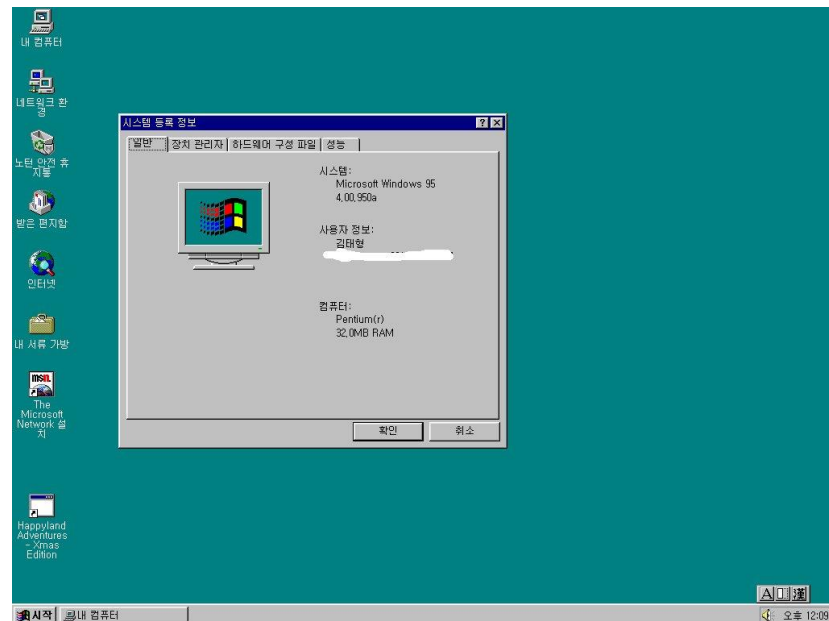


윈도우 운영체제 역사 - 4



• 전설의 운영체제 Windows 95

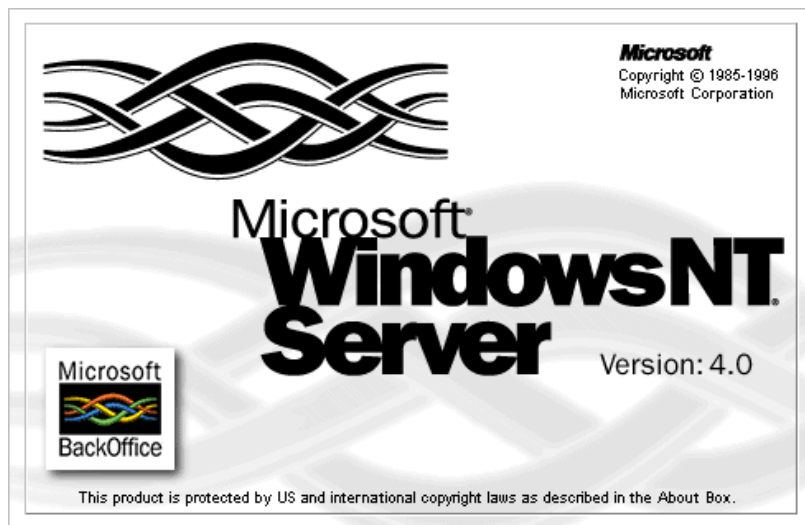
- 1995년 8월 24일 발매한 최초의 32비트 운영체제
- 최초로 CD로 발매된 운영체제
- 깔끔한 화면과 "시작" 버튼의 전설이 시작된 버전
 - 윈도우 운영체제의 상징과도 같은 역할을 수행
- MS가 개인 PC 운영체제 시장을 완전히 잠식하도록 만든 운영체제





윈도우 운영체제 역사 - 5

- IBM의 대한 반격의 서막 Windows NT 4.0
 - 1996년 8월 24일 출시로 저렴한 서버 구축에 많이 사용
 - 인터넷의 보급으로 인해 각종 SOHO(Small Office Home Office)에서 활용
 - 옵션 팩과 서비스 팩으로 웹 서버를 무료로 구축 가능
 - 각 팩은 MS 홈페이지에서 무료로 다운로드 가능
 - 운영체제 버전을 다수 개로 출시하게 된 최초의 운영체제

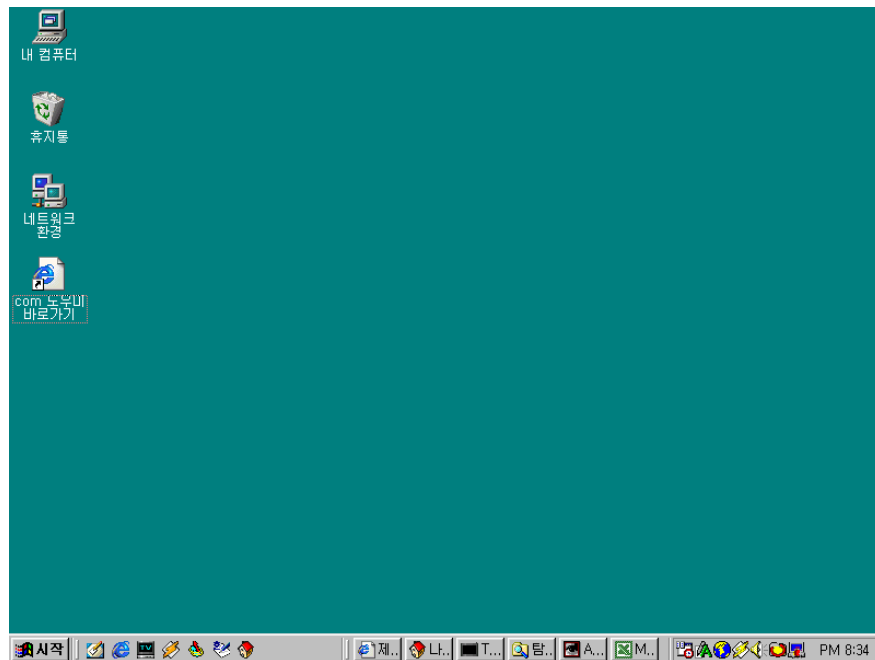




윈도우 운영체제 역사 – 6



- PC 운영체제 전쟁을 끝낸 Windows 98
 - 1998년 6월 25일 기본 인터넷 기능과 다양한 추가 기능을 추가하여 등장
 - 최초 시연 시에 블루스크린을 보이며 다운된 영상이 유행
 - PC 시장을 MS가 완전히 점유하도록 만든 운영체제
 - Windows 95 업데이트 기능들과 추가 기능들로 업무 환경까지 지원



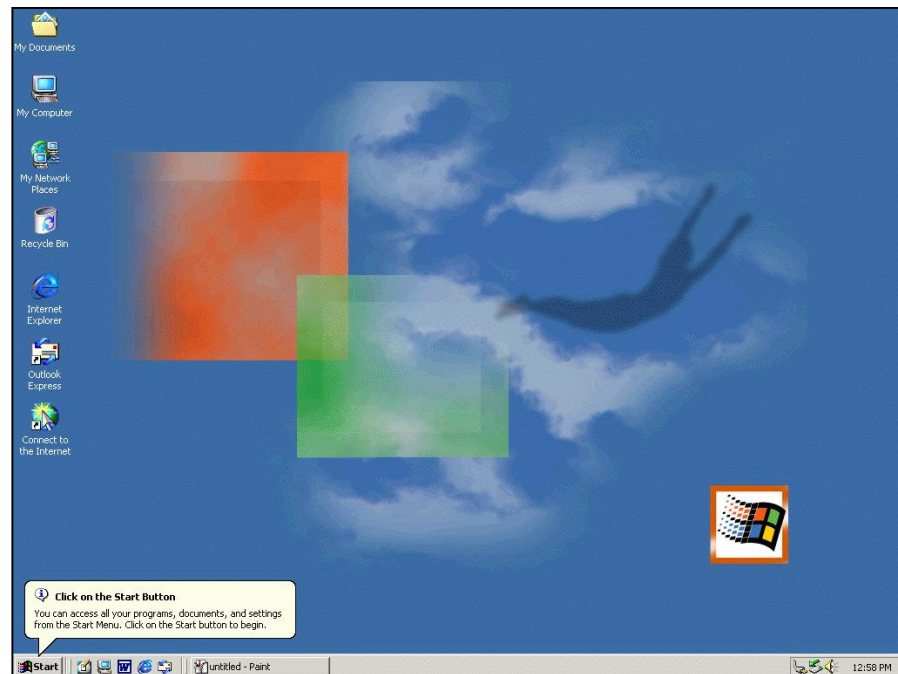
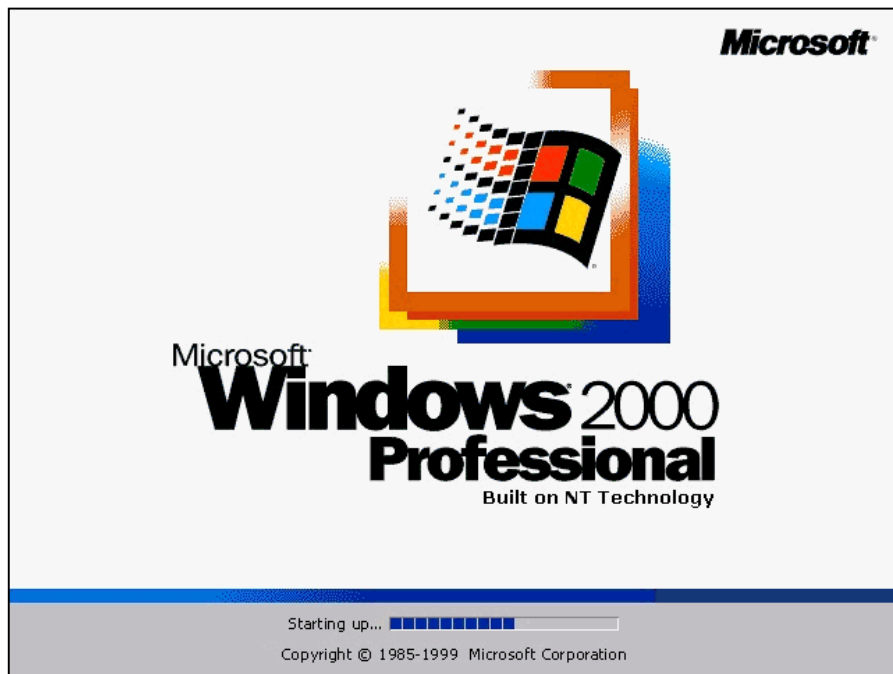


윈도우 운영체제 역사 - 7



• 윈도우 안정화의 시작 Windows 2000

- 2000년 2월 17일에 발표한 NT 계열 운영체제
 - 이 버전부터 NT 라는 이름을 사용하지 않게 됨
- 개인 사용자 용과 서버 군까지 같은 이름으로 발매
- 매우 안정적으로 블루스크린 현상을 개선
- NT 계열의 고질적인 드라이버 문제를 개선하여 사용자 버전과 동일하게 동작



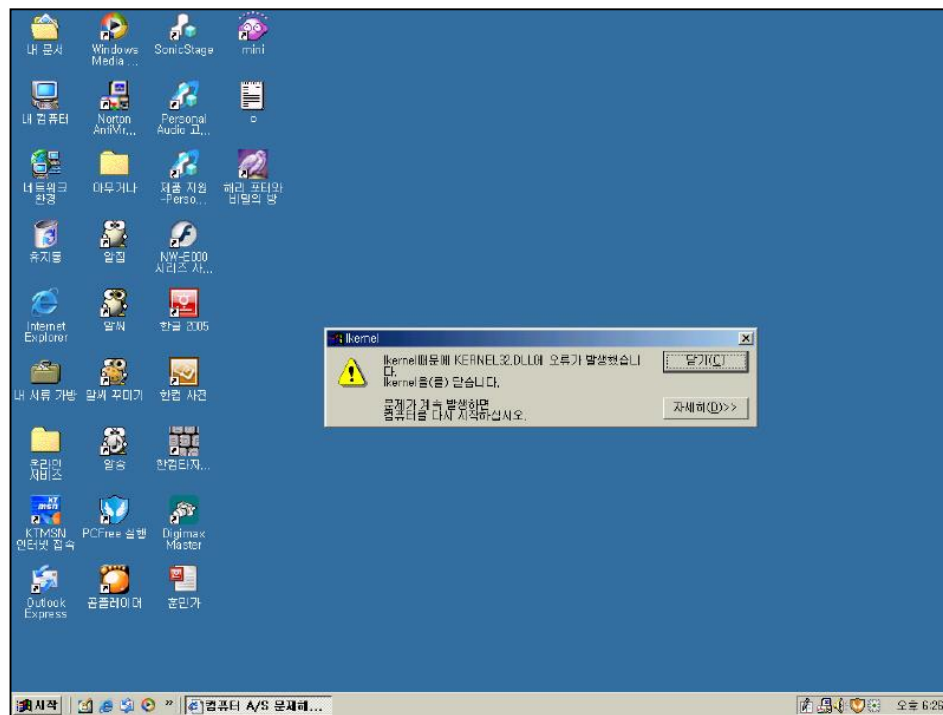
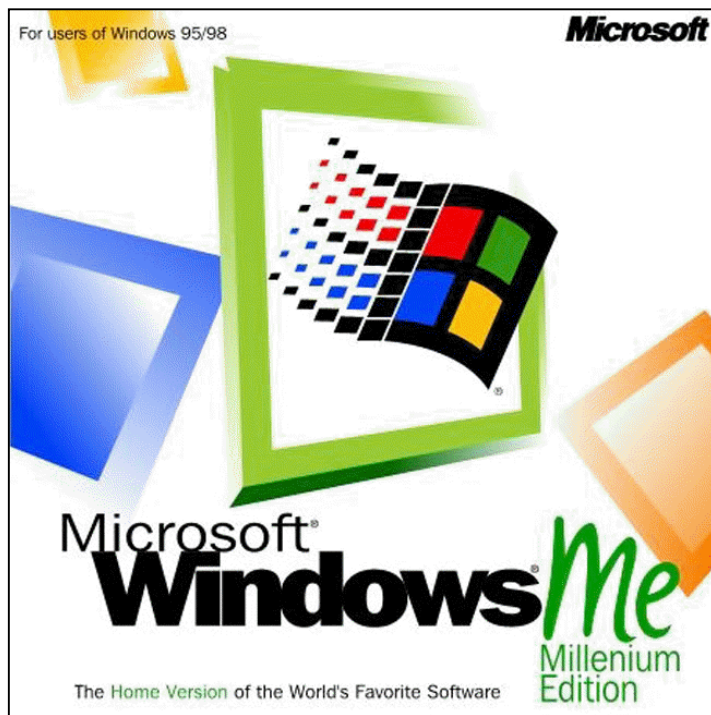


윈도우 운영체제 역사 - 8



• 전대 미문! 최악의 OS Windows ME

- 2000년 9월 14일 발표
- MS에서도 실패작임을 인정한 최악의 OS
 - 95/98 시리즈의 모든 버그를 한번에!
- 더 이상 자세한 설명은 생략한다.





윈도우 운영체제 역사 - 9

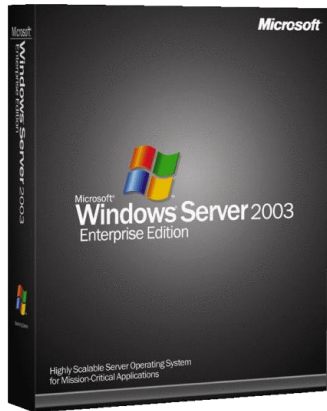


- MS 운영체제를 팀 킬 한 독재 OS Windows XP
 - 2001년 10월 25일 출시. CD만으로 출시한 마지막 버전
 - 이후 DVD로 발매되거나 다운로드 형식으로 판매
 - 2014년 4월 8일 지원 종료될 때까지 이후 출시한 자사 OS를 모두 팀 킬
 - 현재도 다수의 ATM기와 같은 장치에서 사용 중



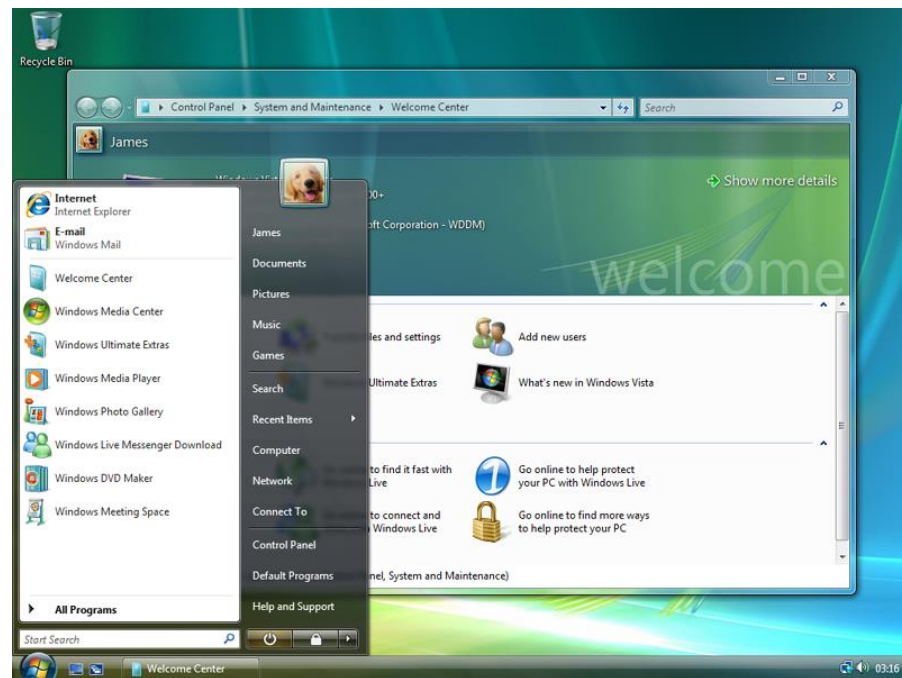


- 어서 와 윈도우 서버는 처음이지? Windows Server 2003 ~





- MS 향후 100년을 내다보다 Windows Vista
 - 2007년 1월 30일에 개인용 버전 발매
 - CD, DVD로 동시 발매한 최초이자 최후 버전
 - 기존 윈도우의 호환성 포기를 시도한 최초의 OS
 - 관련 영상 : <https://www.youtube.com/watch?v=l60HHWWo9z4>

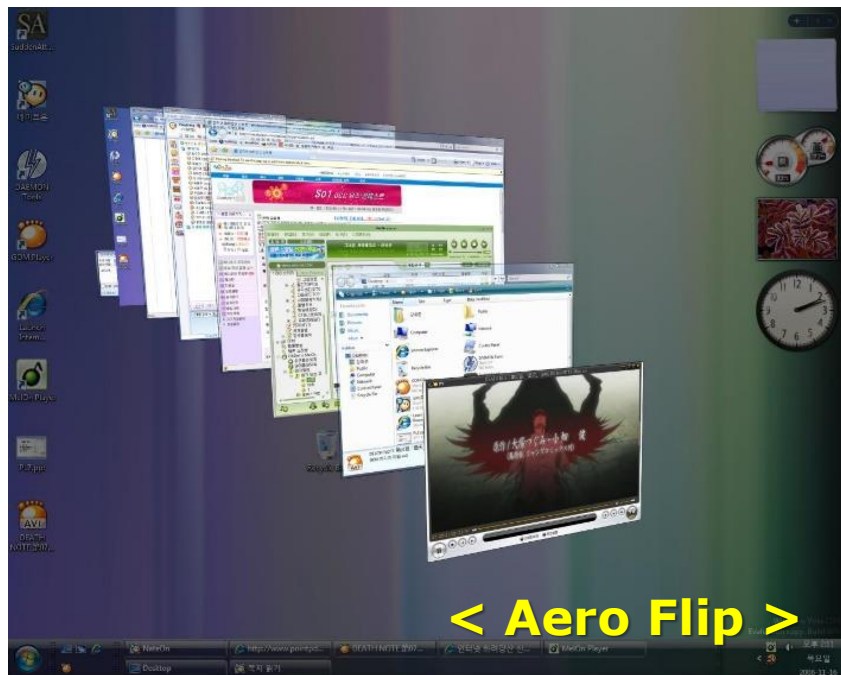




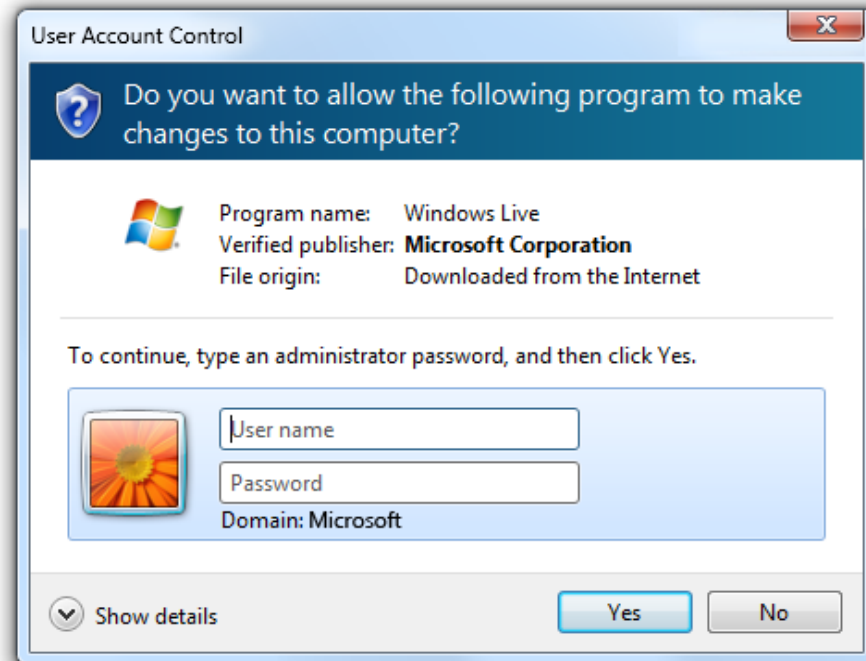
윈도우 운영체제 역사 - 12



- XP에게 팀 킬 당한 최초의 OS Vista
 - 나열하기 어려울 정도로 새로운 기능과 편의성이 추가
 - 보안 뿐만 아니라 개발자 관련 기능들도 대거 추가
 - UAC(User Access Control) 추가로 불편함 증가 대신 보안 향상
 - ActiveX 미 지원과 호환되지 않는 프로그램들로 사용자들에게 외면 당함



< Aero Flip >



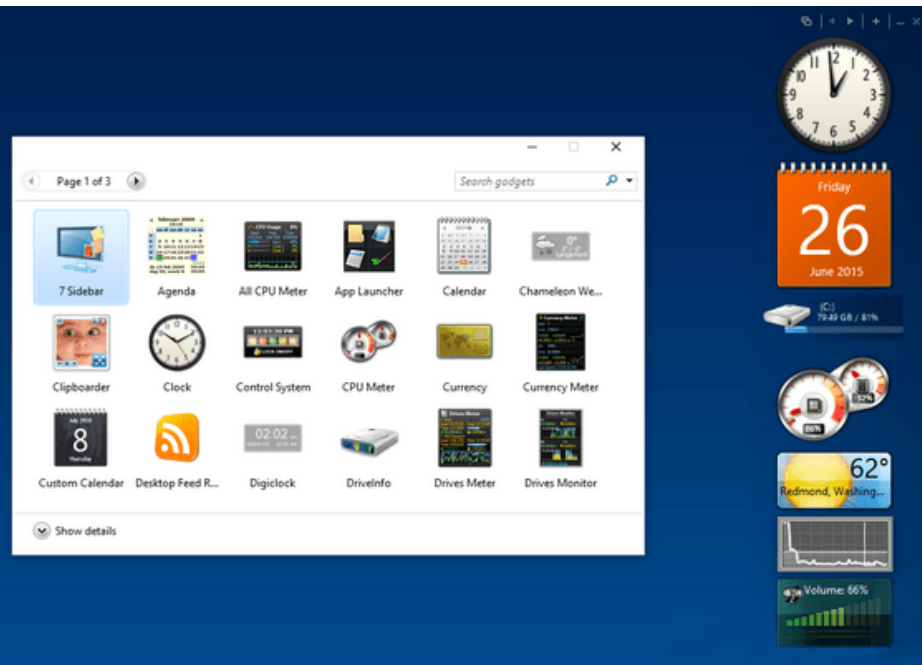
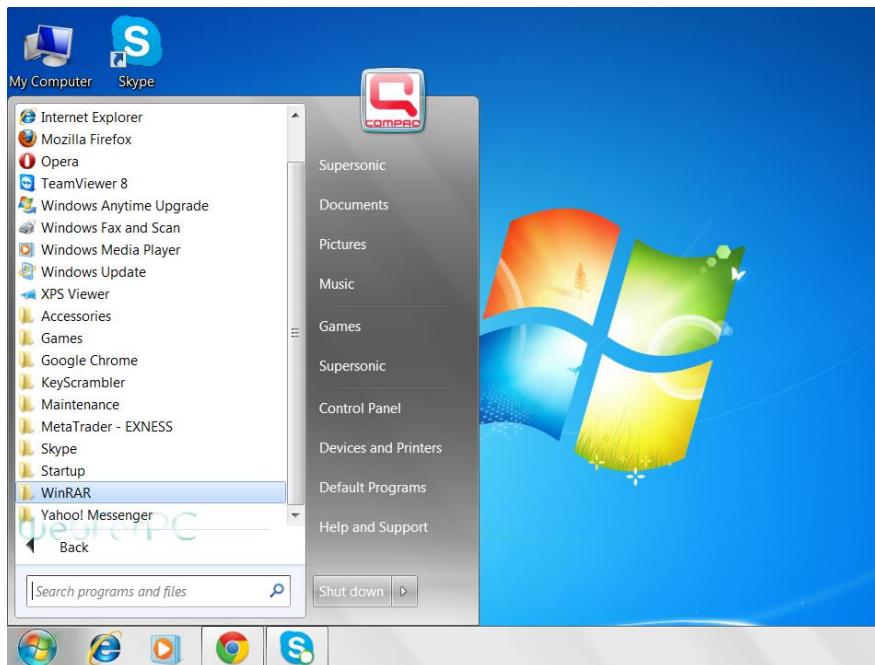
All things are difficult, before they are easy.



Windows 7

• 동작 그만! 밀장 빼 기냐? Windows 7

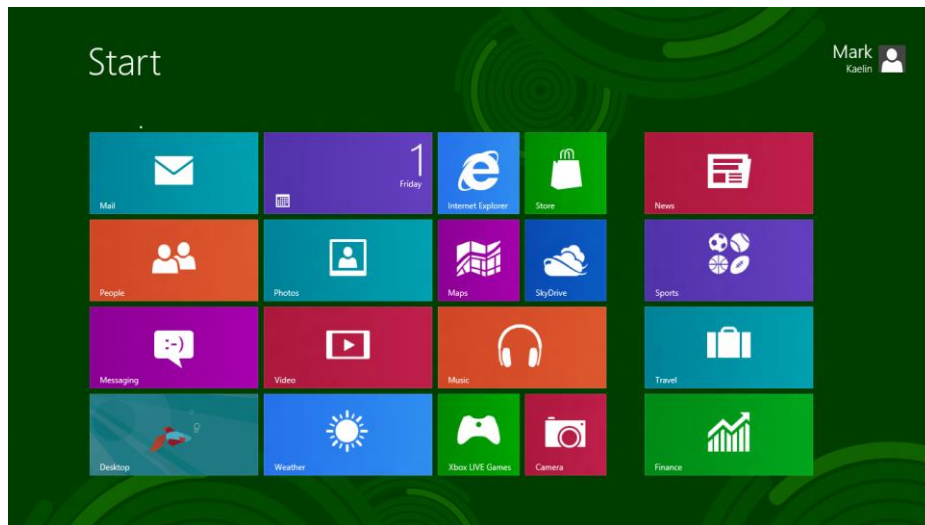
- 2009년 10월 22일 발매.
- Windows XP의 정식 후속 작으로 인식
- 자격증명 관리자, 생체 인식 등과 같이 보안 기능을 추가
- Vista의 부진을 극복하고, XP와 어깨를 나란히 할 수 있는 운영체제



All things are difficult, before they are easy.



- 시작 버튼? 그거 먹는 건가요? Windows 8
– 2012년 10월 26일 발매



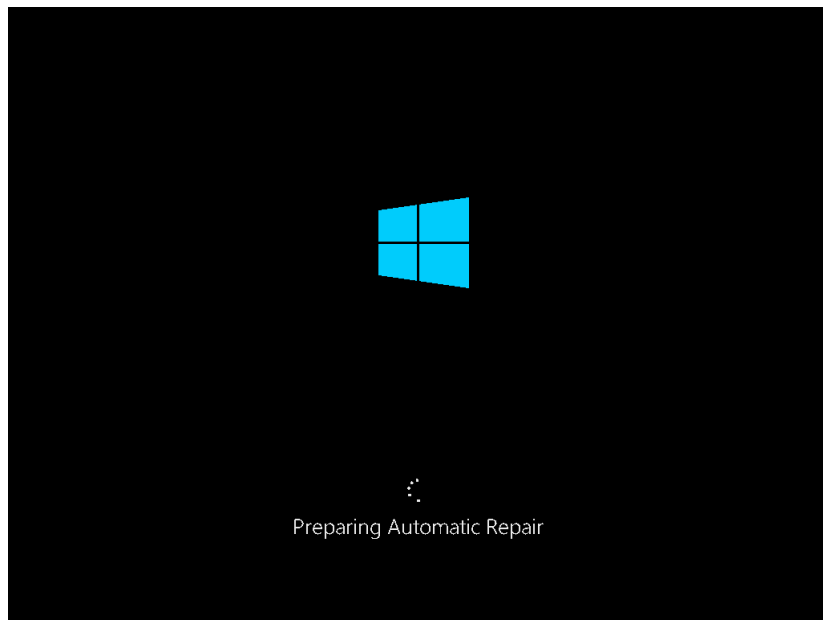
All things are difficult, before they are easy.



Windows 10



- 이제 더 이상의 윈도우 다음 버전은 없습니다. Windows 10
– 2015년 7월 29일 발매

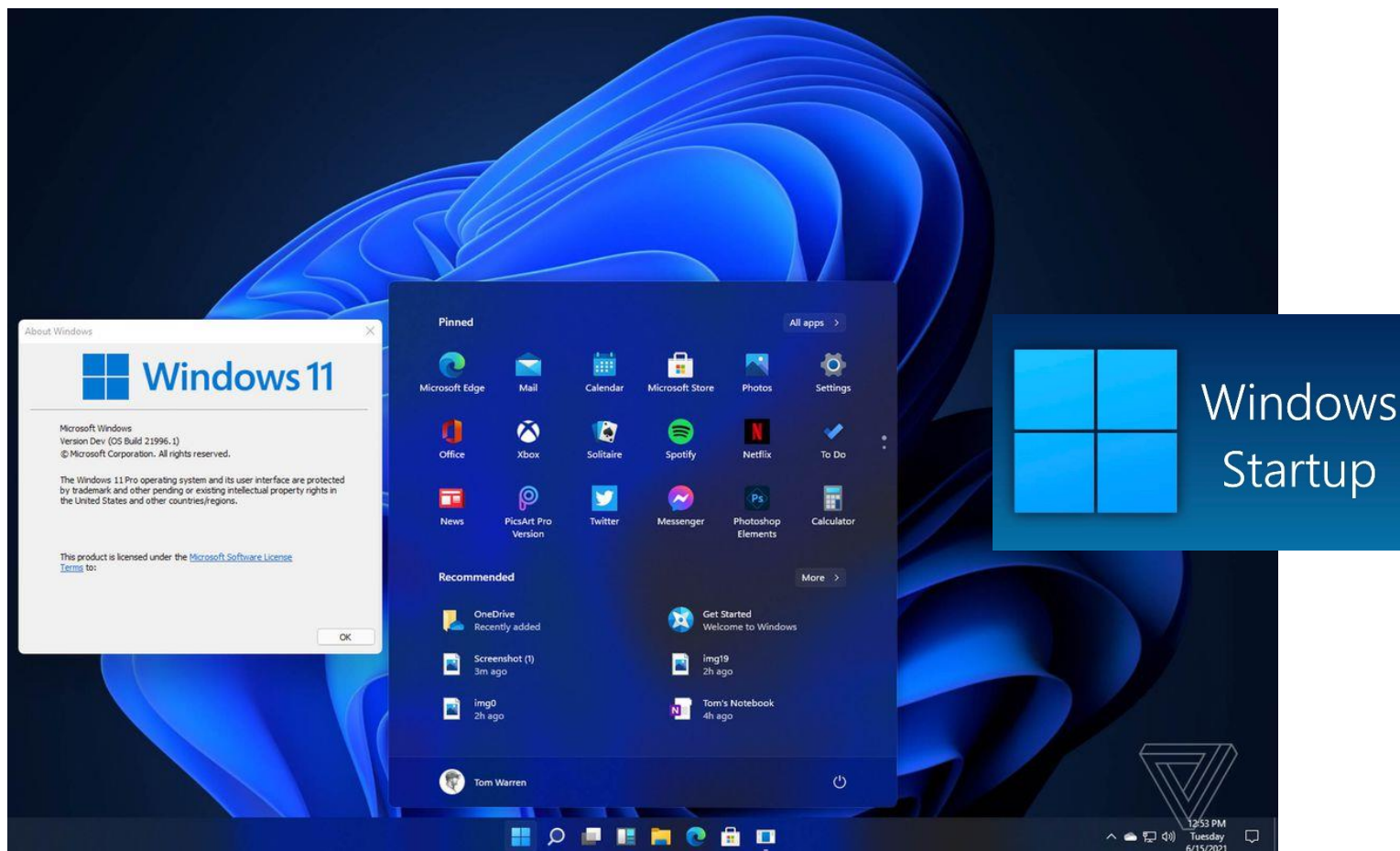


All things are difficult, before they are easy.



Windows 11

- 약속 따윈 장식인데, 사용자들은 그걸 몰라요 Windows 11
 - 2021년 현재 개발 진행 중



All things are difficult, before they are easy.



All things are difficult, before they are easy.



All things are difficult, before they are easy.



All things are difficult, before they are easy.



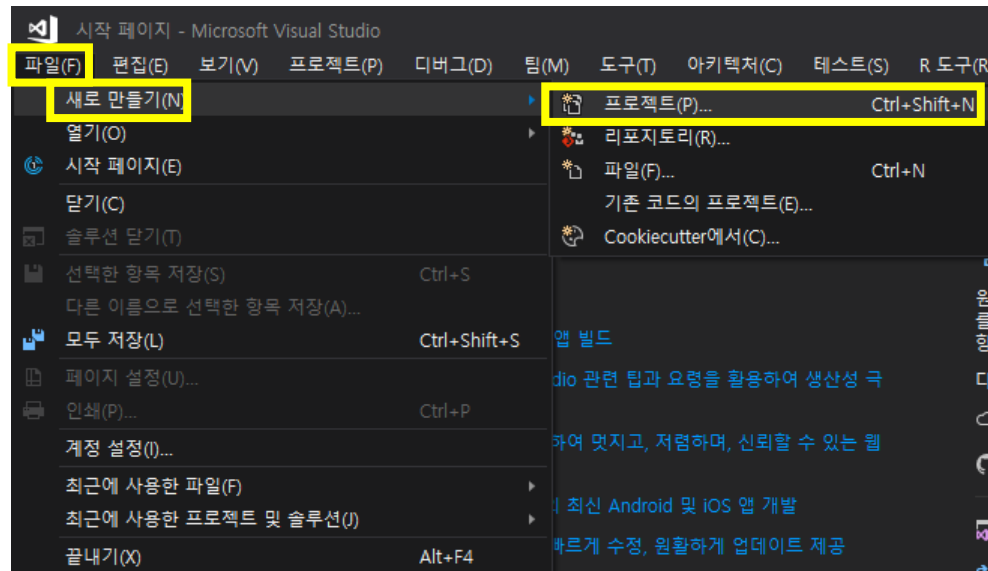


윈도우 어플리케이션 생성 - 1



- Visual Studio 2017 - 1

- Visual Studio 2017 실행 → 파일 → 새로 만들기 → 프로젝트



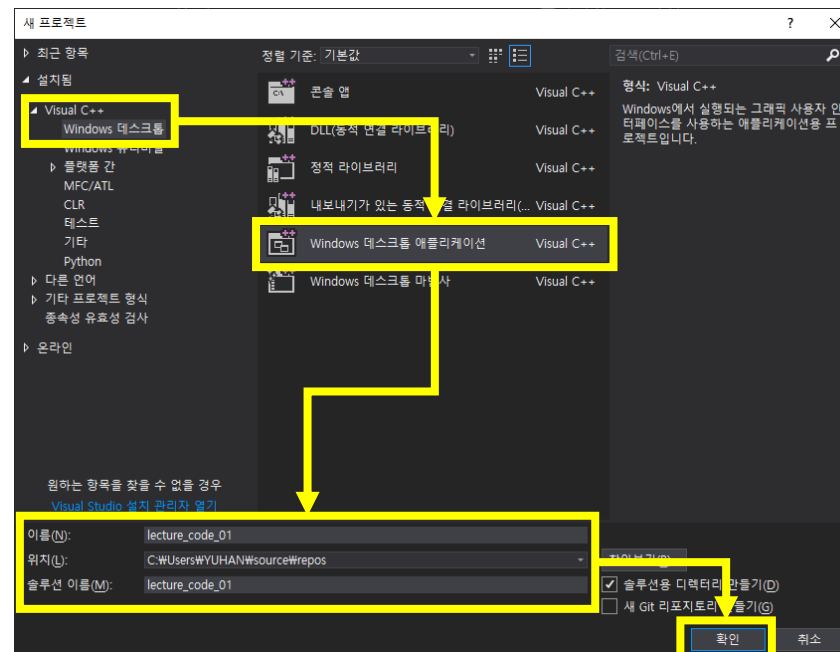


윈도우 어플리케이션 생성 - 2



• Visual Studio 2017 - 2

- Visual C++ → Windows 데스크톱 → Windows 데스크톱 어플리케이션
- 프로젝트 이름과 위치를 설정한 다음 "확인" 클릭하여 프로젝트 생성





윈도우 어플리케이션 생성 - 3



- Visual Studio 2017 - 3
 - 생성된 Windows 데스크톱 애플리케이션 기본 코드

```
lecture_code_01.cpp
lecture_code_01 (전역 범위) wWinMain(HINSTANCE hInstance, HINSTANCE hPrevInstan

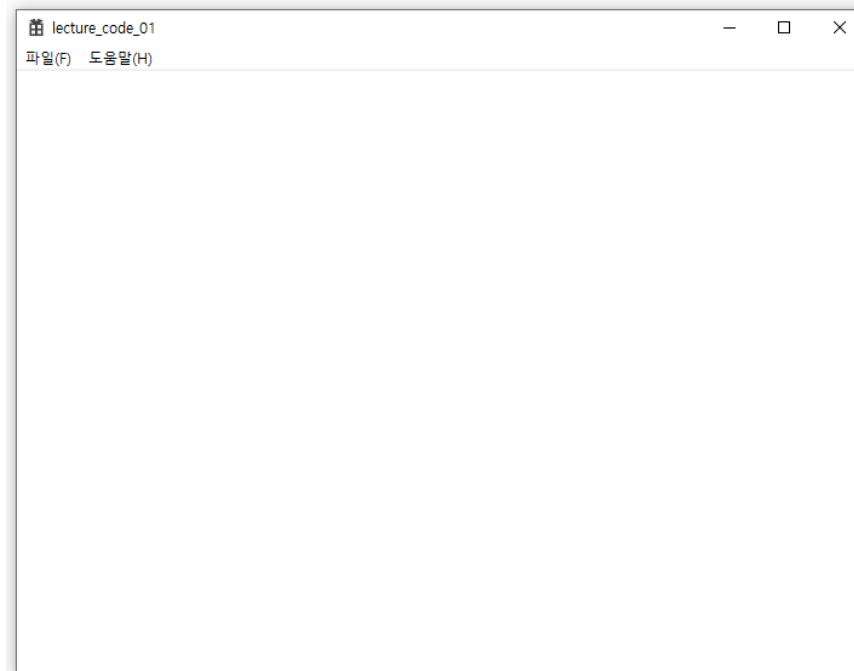
1 // lecture_code_01.cpp : 애플리케이션에 대한 진입점을 정의합니다.
2 //
3
4 #include "framework.h"
5 #include "lecture_code_01.h"
6
7 #define MAX_LOADSTRING 100
8
9 // 전역 변수:
10 HINSTANCE hInst; // 현재 인스턴스입니다.
11 WCHAR szTitle[MAX_LOADSTRING]; // 제목 표시줄 텍스트입니다.
12 WCHAR szWindowClass[MAX_LOADSTRING]; // 기본 창 클래스 이름입니다.
13
14 // 이 코드 모듈에 포함된 함수의 선언을 전달합니다:
15 ATOM MyRegisterClass(HINSTANCE hInstance);
16 BOOL InitInstance(HINSTANCE, int);
17 LRESULT CALLBACK WndProc(HWND, UINT, WPARAM, LPARAM);
18 INT_PTR CALLBACK About(HWND, UINT, WPARAM, LPARAM);
19
20 int APIENTRY wWinMain(_In_ HINSTANCE hInstance,
21                     _In_opt_ HINSTANCE hPrevInstance,
22                     _In_ LPWSTR lpCmdLine,
23                     _In_ int nCmdShow)
24 {
```



윈도우 어플리케이션 생성 - 4



- Visual Studio 2017 - 4
 - "ctrl + F5" 키 조합으로 실행

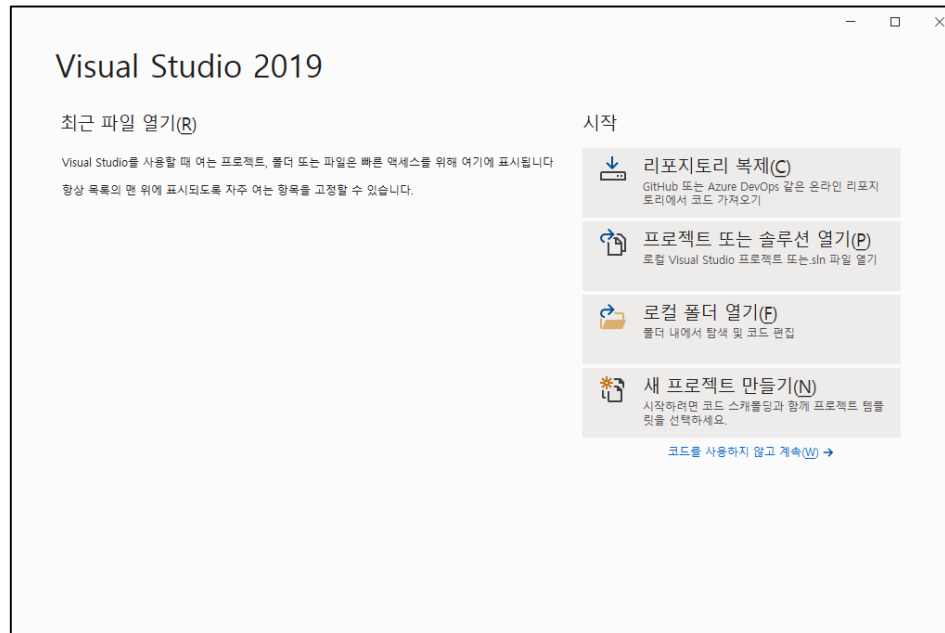




윈도우 어플리케이션 생성 - 4



- Visual Studio 2019 – 1
 - Visual Studio 2017 실행 → 새 프로젝트 만들기





윈도우 어플리케이션 생성 - 5

• Visual Studio 2019 - 2

- 상단 리스트 박스에서 언어와 플랫폼을 C++, Windows로 변경
- "Windows 데스크톱 어플리케이션" 선택하고 다음





윈도우 어플리케이션 생성 – 6



- Visual Studio 2019 – 3
 - 새 프로젝트, 솔루션의 이름과 경로를 설정하고 “만들기”

The screenshot shows the 'New Project' dialog box in Visual Studio 2019. The title bar says '새 프로젝트 구성' (New Project Configuration). Below the title, it says 'Windows 데스크톱 애플리케이션' (Windows Desktop Application) with 'C++', 'Windows', and '데스크톱' (Desktop) as sub-options. The '프로젝트 이름(I)' (Project Name) field contains 'code'. The '위치(L)' (Location) field shows the path 'C:\Users\WYUHAN\source\repos' with a browse button. The '솔루션 이름(M)' (Solution Name) field also contains 'code'. At the bottom, there is a checkbox labeled '솔루션 및 프로젝트를 같은 디렉터리에 배치(D)' (Place solution and project in the same directory), which is currently unchecked. At the bottom right, there are two buttons: '뒤로(B)' (Back) and '만들기(C)' (Create).

All things are difficult, before they are easy.



윈도우 어플리케이션 생성 - 7



- Visual Studio 2019 - 4



- C++에 비해 새롭게 추가된 자료형은 없음
 - 대부분 기존의 기본 자료형을 #define 으로 선언하여 사용
 - int, short, float, void 등의 기존 자료형을 그대로 사용
 - VC++에서는 모두 대문자로 자료형을 선언하여 사용
 - 처음 보는 자료형은 오른쪽 버튼 클릭 후, "정의로 이동"으로 확인 가능



```
typedef unsigned long    DWORD;  
typedef int              BOOL;  
typedef unsigned char    BYTE;  
typedef unsigned short   WORD;  
typedef float            FLOAT;  
typedef FLOAT            *PFLOAT;  
typedef BOOL near        *PBOOL;  
typedef BOOL far         *LPBOOL;  
typedef BYTE near        *PBYTE;  
typedef BYTE far         *LPBYTE;  
typedef int near         *PINT;  
typedef int far          *LPINT;  
typedef WORD near        *PWORD;
```

< minwindef.h >



• C++에서 추가된 자료형도 사용

- 마찬가지로 별도 자료형으로 대문자 선언하여 사용
- Ex. WCHAR → wchar_t

```
// 전역 변수:  
HINSTANCE hInst;  
WCHAR szTitle[MAX_LOADSTRING];  
WCHAR szWindowClass[MAX_LOADSTRING];  
  
// 이  
ATOM  
BOOL  
LRESULT  
INT_F
```

빠른 작업 및 리팩터링... Ctrl+.
이름 바꾸기(R)... Ctrl+R, Ctrl+R
포함 파일의 그래프 생성(I)
정의 피킹 Alt+F12
정의로 이동(G) F12
선언으로 이동(A) Ctrl+F12
모든 참조 찾기(A) Shift+F12

```
//  
// UNICODE (Wide Character) types  
//  
#ifndef _MAC  
typedef wchar_t WCHAR; // wc, 16-bit UNICODE character  
#else  
// some Macintosh compilers don't define wchar_t in a conven  
typedef unsigned short WCHAR; // wc, 16-bit UNICODE cha  
#endif  
  
typedef WCHAR *PWCHAR, *LPWCH, *PWCH;  
typedef CONST WCHAR *LPCWCH, *PCWCH;  
  
typedef _Null_terminated_ WCHAR *NWPSTR, *LPWSTR, *PWSTR;  
typedef _Null_terminated_ PWSTR *PZPWSTR;  
typedef _Null_terminated_ CONST PWSTR *PCZPWSTR;  
typedef _Null_terminated_ WCHAR UNALIGNED *LPUWSTR, *PUWSTR;  
typedef _Null_terminated_ CONST WCHAR *LPCWSTR, *PCWSTR;  
typedef _Null_terminated_ PCWSTR *PZPCWSTR;  
typedef _Null_terminated_ CONST PCWSTR *PCZPCWSTR;  
typedef _Null_terminated_ CONST WCHAR UNALIGNED *LPCUWSTR, *
```



- Windows 사용을 위한 매크로가 추가

```
LRESULT CALLBACK WndProc(HWND, UINT, WPARAM, LPARAM);  
INT_PTR CALLBACK About(HWND, UINT, WPARAM, LPARAM);
```

```
int APIENTRY wWinMain(_In_ HINSTANCE hInstance,  
                     _In_opt_ HINSTANCE hPrevInstance,  
                     _In_ LPWSTR lpCmdLine,  
                     _In_ int nCmdShow)  
{  
    UNREFERENCED_PARAMETER(hPrevInstance);  
    UNREFERENCED_PARAMETER(lpCmdLine);
```

```
    LRESULT CALLBACK WndProc(HWND hWnd, UINT message,  
    {  
        switch (message)  
        {  
            case WM_COMMAND:  
            {  
                int wmlId = LOWORD(wParam);  
                // 메뉴 선택을 구문 분석합니다:  
                switch (wmlId)  
                {
```



• 특수 자료형 - 1

- HINSTANCE, HWND와 같이 대문자 H로 시작하는 자료형
- 내부적으로는 모두 HANDLE(void*) 자료형
 - 내부에서는 내용을 알 수 있으나, 외부에서는 포인터 값만 저장하기 위해 사용

```
//  
// Handle to an Object  
//  
...  
#ifdef STRICT  
    typedef void *HANDLE;  
#if 0 && (_MSC_VER > 1000)  
    #define DECLARE_HANDLE(name) struct name##__ {int unused;}; typedef struct name##__ *name  
#else  
    #define DECLARE_HANDLE(name) struct name##__ {int unused;}; typedef struct name##__ *name  
#endif  
#else  
    typedef PVOID HANDLE;  
    #define DECLARE_HANDLE(name) typedef HANDLE name  
#endif  
typedef HANDLE *PHANDLE;
```



• 특수 자료형 - 2

- HWND, HINSTANCE 등의 자료형 선언부
 - 결국 HANDLE(void*) 자료형을 선언함을 확인할 수 있음
- 모두 대문자 H로 시작하는 것을 확인할 수 있음

```
DECLARE_HANDLE(HKEY);  
typedef HKEY *PHKEY;  
DECLARE_HANDLE(HMETAFILE);  
DECLARE_HANDLE(HINSTANCE);  
typedef HINSTANCE HMODULE;  
DECLARE_HANDLE(HRGN);  
DECLARE_HANDLE(HRSRC);  
DECLARE_HANDLE(HSPRITE);  
DECLARE_HANDLE(HLSURF);  
DECLARE_HANDLE(HSTR);  
DECLARE_HANDLE(HTASK);  
DECLARE_HANDLE(HWINSTA);  
DECLARE_HANDLE(HKL);
```

< minwindef.h >

```
DECLARE_HANDLE(HWND);  
DECLARE_HANDLE(HHOOK);
```

< windef.h >



- wWinMain() 함수

```
int APIENTRY wWinMain(_In_ HINSTANCE hInstance,
                      _In_opt_ HINSTANCE hPrevInstance,
                      _In_ LPWSTR lpCmdLine,
                      _In_ int nCmdShow)
{
    UNREFERENCED_PARAMETER(hPrevInstance);
    UNREFERENCED_PARAMETER(lpCmdLine);

    // TODO: 여기에 코드를 입력합니다.

    // 전역 문자열을 초기화합니다.
    LoadStringW(hInstance, IDS_APP_TITLE, szTitle, MAX_LOADSTRING);
    LoadStringW(hInstance, IDC_LECTURECODE01, szWindowClass, MAX_LOADSTRING);
    MyRegisterClass(hInstance);

    // 애플리케이션 초기화를 수행합니다:
    if (!InitInstance(hInstance, nCmdShow))
    {
        return FALSE;
    }
}
```

All things are difficult, before they are easy.



기본 소스 분석 - 1



```
HACCEL hAccelTable = LoadAccelerators(hInstance, MAKEINTRESOURCE(IDC_LECTURECODE01));

MSG msg;

// 기본 메시지 루프입니다:
while (GetMessage(&msg, nullptr, 0, 0))
{
    if (!TranslateAccelerator(msg.hwnd, hAccelTable, &msg))
    {
        TranslateMessage(&msg);
        DispatchMessage(&msg);
    }
}

return (int) msg.wParam;
}
```



기본 소스 분석 - 1



```
ATOM MyRegisterClass(HINSTANCE hInstance)
{
    WNDCLASSEXW wcex;

    wcex.cbSize = sizeof(WNDCLASSEX);

    wcex.style          = CS_HREDRAW | CS_VREDRAW;
    wcex.lpfnWndProc    = WndProc;
    wcex.cbClsExtra     = 0;
    wcex.cbWndExtra     = 0;
    wcex.hInstance      = hInstance;
    wcex.hIcon          = LoadIcon(hInstance, MAKEINTRESOURCE(IDI_LECTURECODE01));
    wcex.hCursor        = LoadCursor(NULL, IDC_ARROW);
    wcex.hbrBackground  = (HBRUSH)(COLOR_WINDOW+1);
    wcex.lpszMenuName   = MAKEINTRESOURCEW(IDC_LECTURECODE01);
    wcex.lpszClassName  = szWindowClass;
    wcex.hIconSm        = LoadIcon(wcex.hInstance, MAKEINTRESOURCE(IDI_SMALL));

    return RegisterClassExW(&wcex);
}
```



기본 소스 분석 - 1



```
BOOL InitInstance(HINSTANCE hInstance, int nCmdShow)
{
    hInst = hInstance; // 인스턴스 핸들을 전역 변수에 저장합니다.

    HWND hWnd = CreateWindowW(szWindowClass, szTitle, WS_OVERLAPPEDWINDOW,
        CW_USEDEFAULT, 0, CW_USEDEFAULT, 0, nullptr, nullptr, hInstance, nullptr);

    if (!hWnd)
    {
        return FALSE;
    }

    ShowWindow(hWnd, nCmdShow);
    UpdateWindow(hWnd);

    return TRUE;
}
```

All things are difficult, before they are easy.



기본 소스 분석 - 1



```
HRESULT CALLBACK WndProc(HWND hWnd, UINT message, WPARAM wParam, LPARAM lParam)
{
    switch (message)
    {
        case WM_COMMAND:
        {
            int wmlId = LOWORD(wParam);
            // 메뉴 선택을 구문 분석합니다:
            switch (wmlId)
            {
                case IDM_ABOUT:
                    DialogBox(hInst, MAKEINTRESOURCE(IDD_ABOUTBOX), hWnd, About);
                    break;
                case IDM_EXIT:
                    DestroyWindow(hWnd);
                    break;
                default:
                    return DefWindowProc(hWnd, message, wParam, lParam);
            }
        }
        break;
    }
}
```

All things are difficult, before they are easy.



기본 소스 분석 - 1



```
case WM_PAINT:
{
    PAINTSTRUCT ps;
    HDC hdc = BeginPaint(hWnd, &ps);
    // TODO: 여기에 hdc를 사용하는 그리기 코드를 추가합니다...
    EndPaint(hWnd, &ps);
}
break;
case WM_DESTROY:
    PostQuitMessage(0);
    break;
default:
    return DefWindowProc(hWnd, message, wParam, lParam);
}
return 0;
}
```

All things are difficult, before they are easy.



기본 소스 분석 - 1



```
INT_PTR CALLBACK About(HWND hDlg, UINT message, WPARAM wParam, LPARAM lParam)
{
    UNREFERENCED_PARAMETER(lParam);
    switch (message)
    {
        case WM_INITDIALOG:
            return (INT_PTR)TRUE;

        case WM_COMMAND:
            if (LOWORD(wParam) == IDOK || LOWORD(wParam) == IDCANCEL)
            {
                EndDialog(hDlg, LOWORD(wParam));
                return (INT_PTR)TRUE;
            }
            break;
    }
    return (INT_PTR)FALSE;
}
```

All things are difficult, before they are easy.



기본 소스 분석 - 1



```
LRESULT CALLBACK WndProc(HWND hWnd, UINT message, WPARAM wParam, LPARAM lParam)
{
    switch (message)
    {
        case WM_COMMAND:
        {
            // 처리
        }
        break;

        case WM_PAINT:
        {
            // TODO: 여기에 hdc를 사용하는 그리기 코드를 추가합니다...
        }
        break;

        case WM_DESTROY:
            break;

        default:
            return DefWindowProc(hWnd, message, wParam, lParam);
    }
    return 0;
}
```


All things are difficult, before they are easy.



기본 소스 분석 - 1



All things are difficult, before they are easy.



All things are difficult, before they are easy.



All things are difficult, before they are easy.



All things are difficult, before they are easy.



All things are difficult, before they are easy.



All things are difficult, before they are easy.



All things are difficult, before they are easy.

