

Podstawowe typy danych

```
typedef short SHORT;
typedef unsigned short WORD;           // 16-bit
typedef unsigned long DWORD;          // 32-bit

#if defined( _WIN64 )
    typedef __int64 INT_PTR;
#else
    typedef int INT_PTR;
#endif

typedef DWORD COLORREF;
typedef DWORD *LPCOLORREF;

typedef PVOID HANDLE;
typedef HANDLE *LPHANDLE;

typedef HANDLE HWND;
typedef HANDLE HINSTANCE;

typedef LONG_PTR LPARAM;
typedef UINT_PTR WPARAM;
typedef LONG_PTR LRESULT;
```

Podstawowe typy dla znaków

```
typedef wchar_t WCHAR;  
#ifdef UNICODE  
    typedef WCHAR TCHAR;  
#else  
    typedef char TCHAR;  
#endif  
  
typedef CHAR *PCHAR;  
typedef CHAR *PSTR;  
typedef CHAR *LPSTR;  
typedef WCHAR *LPWSTR;  
#ifdef UNICODE  
    typedef LPWSTR LPTSTR;  
#else  
    typedef LPSTR LPTSTR;  
#endif  
  
typedef __nullterminated CONST CHAR *LPCSTR;  
typedef CONST WCHAR *LPCWSTR;  
#ifdef UNICODE  
    typedef LPCWSTR LPCTSTR;  
#else  
    typedef LPCSTR LPCTSTR;  
#endif
```

Notacja węgierska

bDone	boolean / byte
cTotal	counter - int, long, uint etc.
chFirstLetter	character
dwValue	double word
fDone	flag - boolean
nSize	integer
fpPrice	floating-point
dbValue	double
pPtr	pointer
rgValues	range - array
szName	zero-terminated string
pszText	pointer to zero-terminated string
lpszWindowText	long pointer to zero-terminated string
wValue	word
uwValue	unsigned word
lValue	long
hwndMain	handle of a window
m_Something	member of a class
m_rgfpValues	array of floating-point values

Makra

```
int GET_X_LPARAM(LPARAM lParam);  
int GET_Y_LPARAM(LPARAM lParam);  
  
BYTE HIBYTE(WORD wValue);  
BYTE LOBYTE(WORD wValue);  
WORD HIWORD(DWORD dwValue);  
WORD LOWORD(DWORD dwValue);  
  
DWORD MAKELONG(WORD wLow, WORD wHigh);  
WORD MAKEWORD(BYTE bLow, BYTE bHigh);  
  
WPARAM MAKEWPARAM(WORD wLow, WORD wHigh);  
LPARAM MAKELPARAM(WORD wLow, WORD wHigh);  
LRESULT MAKELRESULT(WORD wLow, WORD wHigh);
```

Struktury graficzne

```
typedef struct tagPOINT {
    LONG x;
    LONG y;
} POINT, *PPOINT;

typedef struct tagPOINTS {
    SHORT x;
    SHORT y;
} POINTS, *PPOINTS;

typedef struct _RECT {
    LONG left;
    LONG top;
    LONG right;
    LONG bottom;
} RECT, *PRECT;

typedef struct tagSIZE {
    LONG cx;
    LONG cy;
} SIZE, *PSIZE;
```

Obsługa myszy

■ Komunikaty obszaru klienta

- WM_MOUSEMOVE
- WM_LBUTTONDOWN, ...UP, ...DBLCLK (L, M, R)
- WM_MOUSEHOVER, WM_MOUSELEAVE, TrackMouseEvent()
- wParam powyższych komunikatów: MK_LBUTTON, MK_MBUTTON, MK_RBUTTON, MK_CONTROL, MK_SHIFT

■ Pozostałe komunikaty

- WM_NCHITTEST
- WM_NCMOUSEMOVE
- WM_NCLBUTTONDOWN, ...UP, ...DBLCLK
- WM_NCMOUSEHOVER, WM_NCMOUSELEAVE

■ Dwuklik - CS_DBLCLKS w klasie okna

- kolejność komunikatów: DOWN, UP, DBLCLK, UP

Cechy myszy

■ Kursor

- kolorowy, animowany, z punktem wskazującym (*hot spot*)
- LoadCursor(), LoadImage(), LoadCursorFromFile()
- GetCursorPos(), SetCursorPos()
- ShowCursor(), GetCursorInfo(), ClipCursor()
- SetClassLong(), CopyCursor(), DestroyCursor()
- WM_SETCURSOR

■ Przechwytywanie myszy poza oknem (*capture*)

- sposób na otrzymywanie komunikatów myszy poza obszarem okna
- SetCapture(), ReleaseCapture(), WM_CAPTURECHANGED

■ Ustawienia

- GetSystemMetrics(), SystemParametersInfo()

Dodatkowe możliwości myszy

- *Wheel* [od 95/NT 3.51]
 - obsługa pokrętła myszy
 - WM_MOUSEWHEEL
- *ClickLock* [Me/XP]
 - ignorowanie puszczenia przycisku
- *Sonar* [Me/XP]
 - rysowanie okręgów wokół kursora po naciśnięciu CTRL
- *Vanish* [Me/XP]
 - ukrycie kursora podczas pisania na klawiaturze
- *XBUTTONs* [2000+]
 - obsługa myszy z 5 przyciskami

Schemat działania klawiatury

1. Użytkownik naciska lub puszcza klawisz
2. Klawiatura wysyła do sterownika *scan code* klawisza
 - *scan code* - zależny od klawiatury identyfikator klawisza
3. Sterownik określa *virtual-key code* klawisza
 - *virtual-key code* - niezależna od klawiatury wartość określająca zastosowanie klawisza
4. Do systemowej kolejki komunikatów dodawany jest komunikat zawierający *scan code* i *virtual-key code*
5. System pobiera komunikat z kolejki i przesyła go do kolejki wątku, którego okno ma *focus*
6. Pobrany z kolejki komunikat trafia do procedury okna

Obsługa klawiatury

■ Komunikaty przycisków

- WM_KEYDOWN, WM_KEYUP
- WM_SYSKEYDOWN, WM_SYSKEYUP
- autopowtarzanie:
 - wiele WM_KEYDOWN, jeden WM_KEYUP
 - sprawdzać licznik powtórzeń w parametrze komunikatu

■ Komunikaty znaków

- generowane z komunikatów przycisków za pomocą TranslateMessage()
- WM_CHAR, WM_DEADCHAR, WM_SYSCHAR, WM_SYSDEADCHAR, WM_UNICHAR

■ Konwersje

- MapVirtualKey(), ToAscii(), ToUnicode(), VkKeyScan()

Cechy klawiatury

■ *Focus*

- *focus* może mieć tylko okno aktywne lub jego potomne
- SetFocus(), GetFocus()
- WM_SETFOCUS, WM_KILLFOCUS

■ *Caret*

- graficzny wskaźnik pozycji kursora klawiatury
- CreateCaret(), DestroyCaret()
- ShowCaret(), HideCaret()
- GetCaretBlinkTime(), SetCaretBlinkTime()
- GetCaretPos(), SetCaretPos()

■ Stan przycisków

- GetKeyState(), GetAsyncKeyState()
- GetKeyboardState(), SetKeyboardState()

Dodatkowe możliwości klawiatury

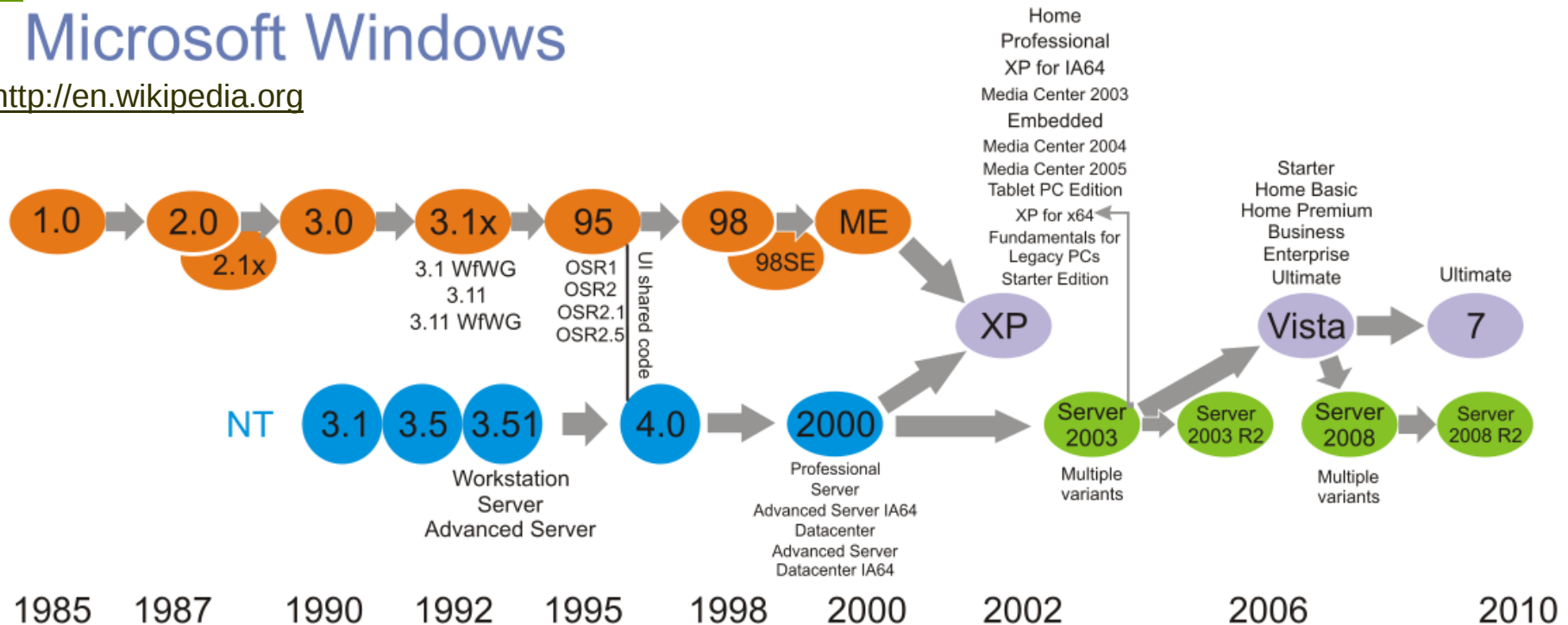
- *Hot-key*
 - kombinacja klawiszy generująca WM_HOTKEY
 - rejestracja: RegisterHotKey()
 - wyrejestrowanie: UnregisterHotKey()
 - WM_SETHOTKEY – ustawienie skrótu klawiszowego aktywującego okno z komunikatem WM_SYSCOMMAND
- Klawisze specjalne [Me/2000]
 - WM_APPCOMMAND
- Ustawienia językowe
 - LoadKeyboardLayout(), UnloadKeyboardLayout()
 - ActivateKeyboardLayout()
- Symulacja klawiatury i myszy
 - SendInput() [98/NT4SP3]
 - keybd_event(), mouse_event() [95/NT3.51]

Historia systemu Windows

- *Xerox, graficzny interfejs, lata 70-te*
- *MS-DOS – 1981*
- *Macintosh – styczeń 1984*
- Windows 1.0 – 20.11.1985
- Windows 2.0 – 9.12.1987
- *OS/2 1.1 – październik 1988*
- Windows 3.0 – 22.05.1990
- Windows 3.1 – kwiecień 1992
- Windows 95 – 24.08.1995
- Windows 98 – 25.06.1998
- Windows Me – 19.06.2000
- Windows NT 3.1 – 27.07.1993
- Windows NT 4.0 – 29.07.1996
- Windows 2000 – 17.02.2000
- Windows XP – 25.10.2001
- Windows Server 2003 – 24.04.2003
- Windows Vista – 30.01.2007
- Windows Home Server – 16.06.2007
- Windows Server 2008 – 27.02.2008
- Windows 7 – wersja beta od stycznia 2009

Microsoft Windows

<http://en.wikipedia.org>



Microsoft
Windows 95

Microsoft
Windows 98

Microsoft
Windows NT



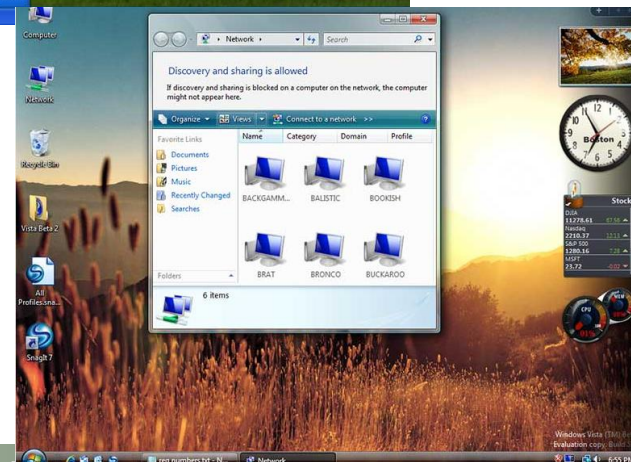
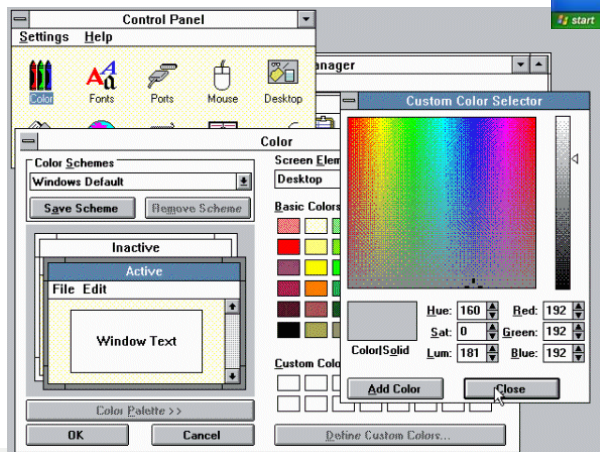
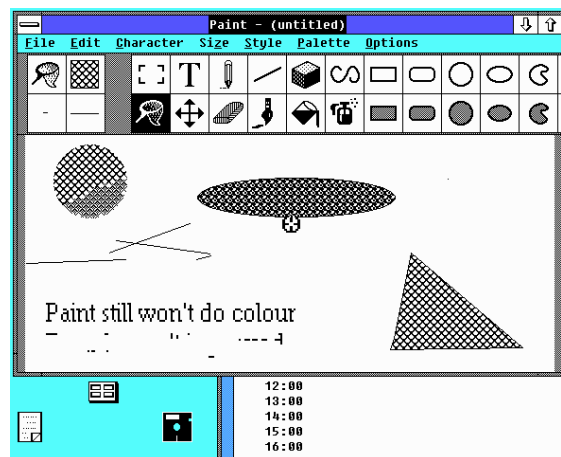
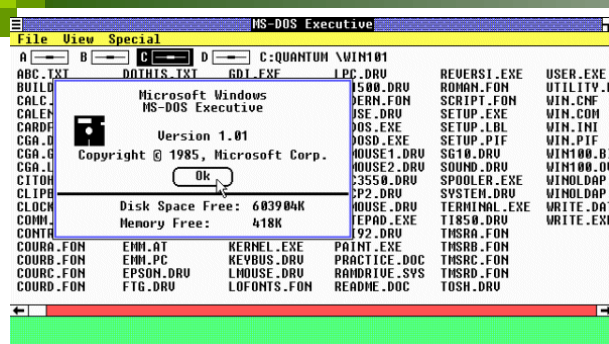
Microsoft
Windows 2000

Windows Server 2008



Windows Vista

Windows 7



Sprawdzanie wersji Windows

■ GetVersionEx()

	major	minor	platformId	productType
95	4	0	VER_PLATFORM_WIN32_WINDOWS	
98	4	10	VER_PLATFORM_WIN32_WINDOWS	
Me	4	90	VER_PLATFORM_WIN32_WINDOWS	
NT 3.51	3	51	VER_PLATFORM_WIN32NT	
NT 4.0	4	0	VER_PLATFORM_WIN32NT	
2000	5	0	VER_PLATFORM_WIN32NT	
XP	5	1	VER_PLATFORM_WIN32NT	
2003	5	2	VER_PLATFORM_WIN32NT	
Vista	6	0	VER_PLATFORM_WIN32NT	VER_NT_WORKSTATION
2008	6	0	VER_PLATFORM_WIN32NT	VER_NT_SERVER
7	6	1	VER_PLATFORM_WIN32NT	VER_NT_WORKSTATION
2008 R2	6	1	VER_PLATFORM_WIN32NT	VER_NT_SERVER

■ VerifyVersionInfo()

Statystyki użycia systemów operacyjnych

■ Systemy operacyjne użytkowników Internetu:

Polska		Świat	
Windows XP	84,1%	Windows XP	71,4%
Windows Vista	13,6%	Windows Vista	15,6%
Windows 2000	0,6%	Mac	5,3%
Windows 98	0,6%	Linux	3,8%
Linux	0,4%	Windows 2000	1,7%
MacOS X	0,3%	Windows 2003	1,7%
Windows 2003	0,1%	Windows 98	0,1%
Windows ME	0,1%		

<http://www.ranking.pl> – styczeń 2009

<http://www.w3schools.com> – grudzień 2008