

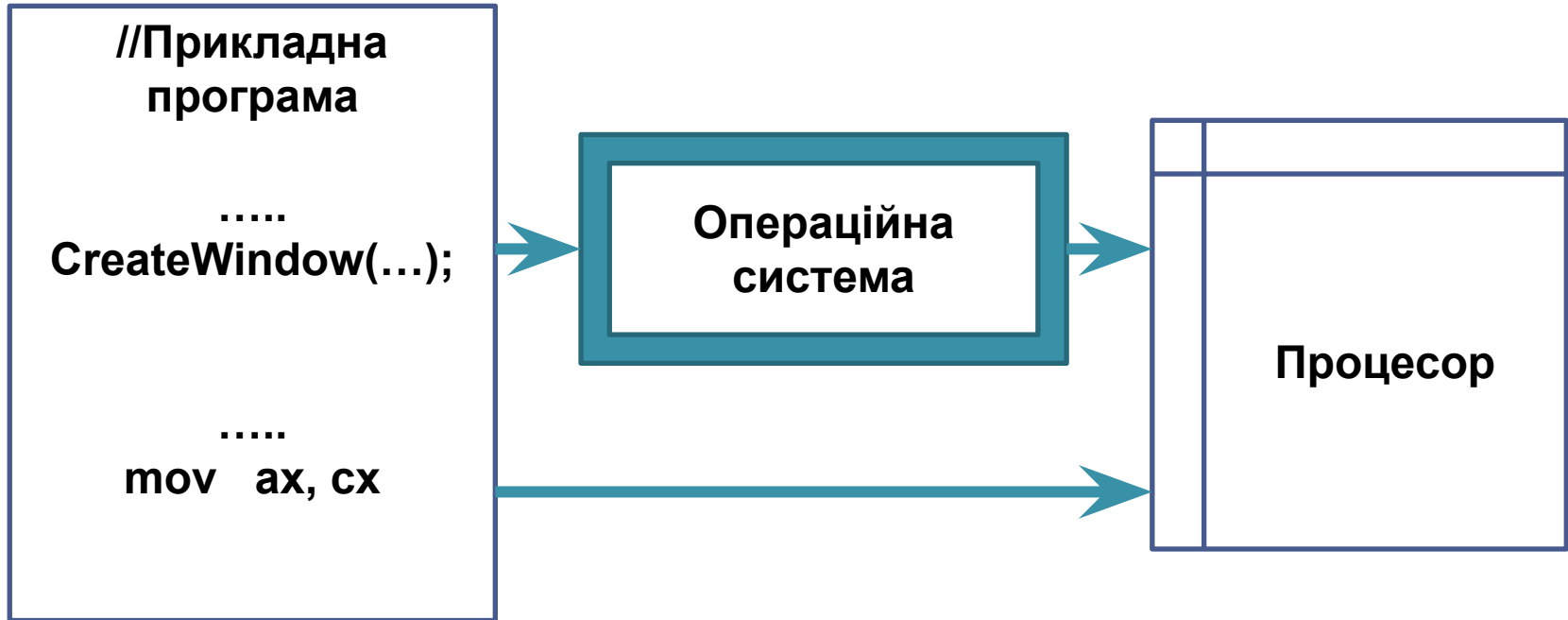


ПРОГРАМУВАННЯ ДЛЯ ОС WINDOWS.

План лекції

1. Особливості програмування під Windows
2. API
3. Структура Windows-додатку
4. Повідомлення та їх обробка

Прикладна програма



API

- **API** (Application Programming Interface) - бібліотека інтерфейсу додатків
- **Windows API (WinAPI)** – загальна назва набору базових функцій інтерфейсів програмування додатків операційних систем сімейств Microsoft Windows. Надає прямий спосіб взаємодії додатків з Windows. `#include <windows.h>`
- Реалізація:
 - Базові функції - kernel32.dll і advapi32.dll
 - Базові модулі GUI - kernel32.dll ш advapi32.dll
- Біблотеки :
 - Active Template Library (ATL)
 - Windows Template Library (WTL)
 - Microsoft Foundation Classes (MFC)
 - Net/WinForms/WPF
- Windows API представляє собою набір функцій, структур даних і числових констант, відповідних угодам мови C.

API

- **Типи даних**

- **BYTE** - 1-байтне ціле без знаку.
- **COLORREF** - використовується для роботи з кольором (4 байта).
- **DWORD** - 4-х байтне ціле або адреса.
- **LONG** - 4-х байтне ціле знакове.
- **UINT** 4-х байтне ціле без знака.
- **NULL** – нульовий покажчик
- **BOOL** – аналог bool

API

- **TCHAR** – символний тип — аналог char і wchar_t.
- **LPCSTR** – 4-х байтний покажчик на константний рядок символів (long pointer constant string); закінчується '\0'
- **LPCTSTR** – покажчик на константний рядок символів без UNICODE (long pointer constant TCHAR string)
- **LPCWSTR** – покажчик на константний UNICODE рядок (long pointer constant wide character string).
- **LPSTR** – покажчик на рядок символів (long pointer string); закінчується '\0'
- **LPTSTR** – покажчик на рядок символів без UNICODE (long pointer TCHAR string)
- **LPWSTR** – покажчик на UNICODE рядок (long pointer wide character string)

API

- **HANDLE** - 4-х байтне ціле , яке використовується в якості дескриптора (ідентифікатор) об'єкта.
- **HWND** - дескриптор вікна.
- **HBITMAP** - дескриптор растрового зображення (handle bitmap)
- **HBRUSH** - дескриптор кисті
- **HCURSOR** - дескриптор курсору
- **HDC** - дескриптор контексту пристрою
- **HFONT** - дескриптор шрифту
- **HICONS** - дескриптор криптограми
- **HINSTANCE** - дескриптор екземпляра додатка
- **HMENU** - дескриптор меню
- **HPEN** - дескриптор пера

Головна функція Windows-додатку

```
int WINAPI WinMain(  
    HINSTANCE hInstance,  
    HINSTANCE hPrevInstance,  
    LPSTR lpCmdLine,  
    int nCmdShow)
```

Дескриптор даного
екземпляру додатку

Дескриптор
додатку, який
запустив це вікно

Покажчик на
параметри
командного рядка

Режим запуску
(напр., “звернуте
вікно”)

```
{
```

```
}
```


Функція WinMain

- Віконна процедура.
- Цикл обробки повідомлень.

I Визначити загальні властивості вікон-об'єктів

```
typedef struct tagWNDCLASSEX
{
    UINT cbSize; //розмір структури в байтах
    UINT style; //стиль класу вікна
    WNDPROC lpfnWndProc; //покажчик на функцію вікна
    int cbClsExtra; // додаткова пам'ять в байтах
                        //для класу вікна
    int cbWndExtra; //додаткова пам'ят в байтах
                        //для кожного вікна цього класу
    HINSTANCE hInstance; //дескриптор примірника додатку
    HICON hIcon;
    HCURSOR hCursor;
    HBRUSH hbrBackground; //для зафарбовування фону вікна
    LPCTSTR lpszMenuName; //ім'я меню вікна
    LPCTSTR lpszClassName; //ім'я класу вікна
    HICON hIconSm; //дескриптор малої піктограми
} WNDCLASSEX;
```

I Визначити загальні властивості вікон-об'єктів

- Більшість значень - за замовчуванням (==0)
 - Функція `memset(&wc, 0, sizeof(wc))// WNDCLASS wc;`
- Стиль вікна - побітове об'єднання констант. Наприклад,
 - **CS_DBLCLKS** програма може реагувати на подвійне клацання мишкою
 - **CS_VREDRAW / CS_HREDRAW** вікно перерисовується при кожній зміні розміру по вертикалі / горизонталі
 - **CS_NOCLOSE** заборона закриття вікна користувачем
- `wc.hIcon=LoadIcon(hInst,MAKEINTRESOURCE(IDI_APP_ICON));`

2 Зареєструвати клас

Використовується функція

```
RegisterClassEx (&w);
```

Повертає унікальний ідентифікатор
зареєстрованого класу вікна

3 Створення вікна

- На базі зареєстрованого класу
- Функція

HWND CreateWindowEX (

LPCTSTR lpClassName, // назва класу вікна **LPCTSTR**

lpWindowName, // заголовок вікна

DWORD dwExStyle, **DWORD dwStyle,** // стиль вікна

int x, **int y,** // позиція вікна

int nWidth, **int nHeight,** // розміри вікна

HWND hWndParent, // покажчик на

// батьківське вікно

HMENU hMenu, // покажчик на меню

HINSTANCE hInstance, // покажчик на

// додаток, який створює вікно

LPVOID lpParam // додаткові параметри

);

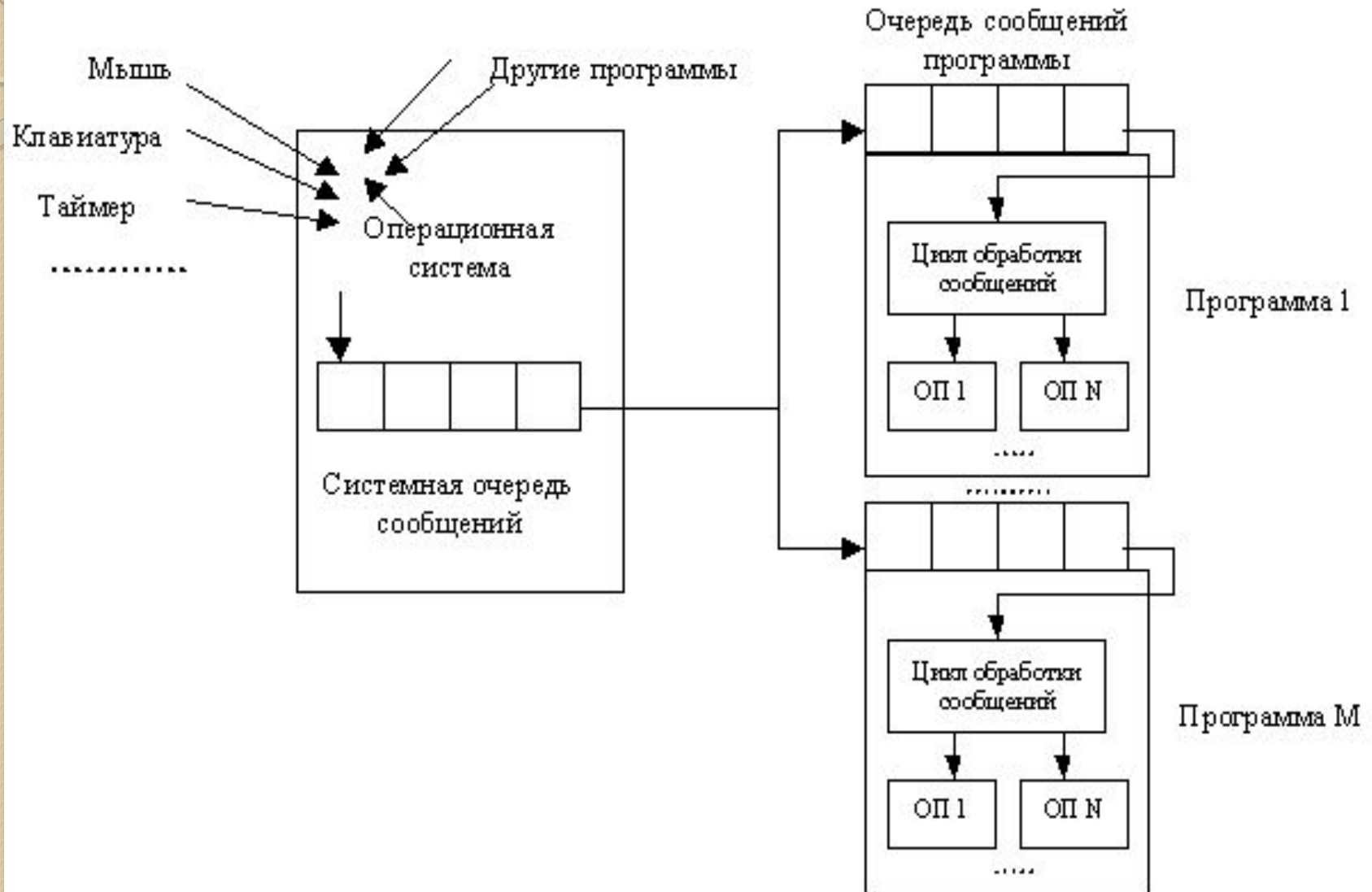
4 Відображення вікна

- `BOOL ShowWindow(
 HWND hWnd, // дескриптор вікна
 int nCmdShow // вид вікна
 // (із парметрів WinMain)
);`
- `BOOL UpdateWindow (
 HWND hWnd // окно
);`

4 Обробка повідомлень

- Повідомлення є реакцією системи на події, що в ній відбуваються , і єдиним засобом зв'язку вікна (і всього додатку) з операційної системою
- Від “миші”
 - **WM_LBUTTONDOWN/WM_RBUTTONDOWN**
 - `afx_msg void OnLButtonDown(UINT nFlags, CPoint point);`
 - **WM_LBUTTONDOWNDBLCLK/WM_RBUTTONDOWNDBLCLK**
 - **WM_MOUSEMOVE**
- Від діалога
 - **WM_INITDIALOG**

4 Обробка повідомлень



Цикл обробки повідомлень

```
while(GetMessage(&msg,NULL,0,0))
{
    TranslateMessage(&msg);
    /*перетворює апартні повідомлення від
    клавіатури в симовли*/
    DispatchMessage(&msg);
    /* викликає віконну функцію того вікна,
    якому призначено повідомлення*/
}
```