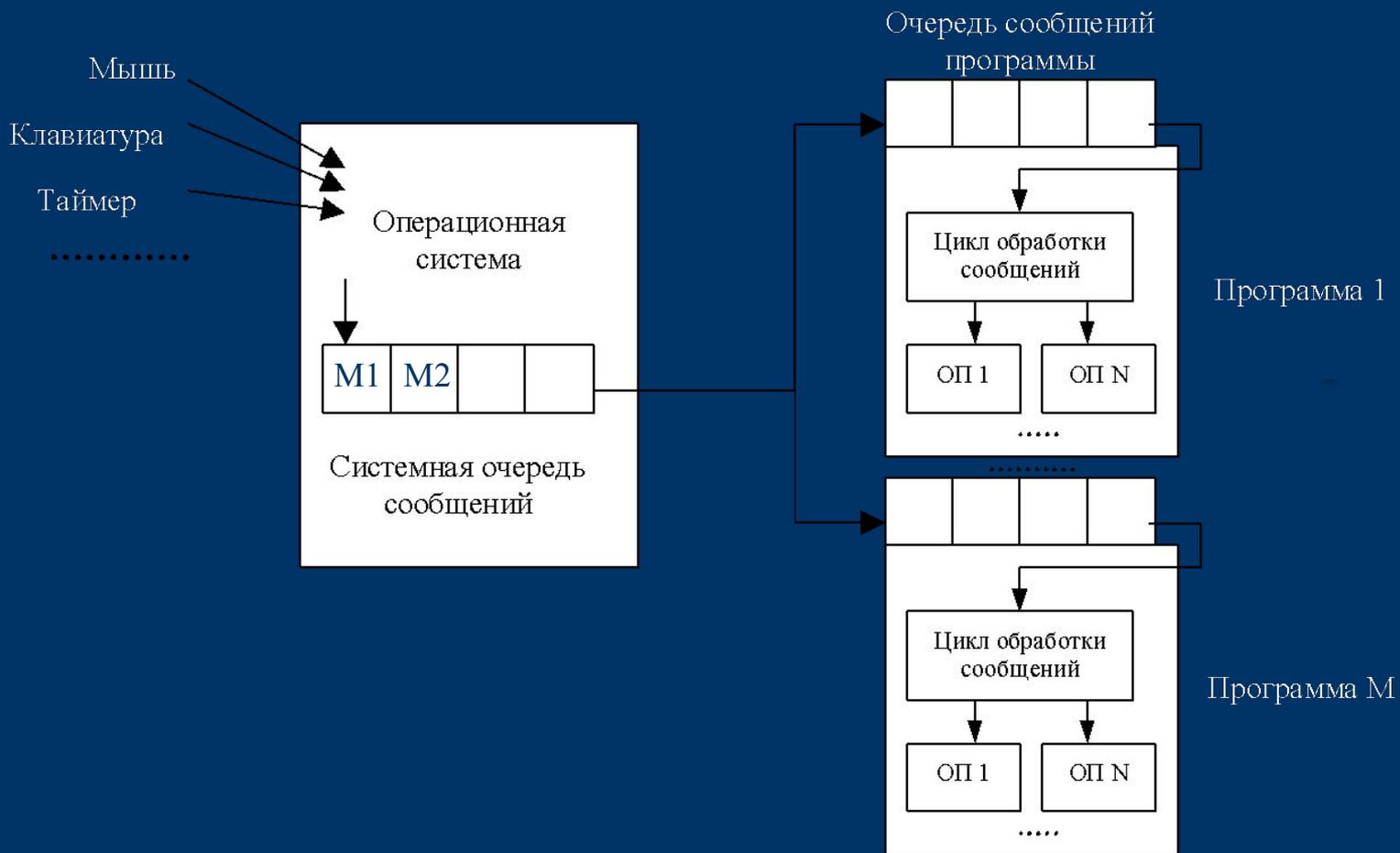


Лекция 6 Клавиатура



Порядок прохождения сообщений



Сообщение от клавиатуры



Сообщение от клавиатуры проходит две очереди прежде, чем попадет в вашу программу – системную очередь сообщений и очередь сообщений приложения. Из системной очереди Windows выбирает сообщения, предназначенные исключительно ей (например, что нажата перезагрузка машины <Ctrl+Alt+Del> или переключение между приложениями <Alt+Tab>). Таким образом, программа получает только адресованные ей сообщения от клавиатуры.

АКТИВНОЕ ОКНО

The image illustrates an active window in a multi-window environment. It shows three overlapping Microsoft Internet Explorer windows:

- Top Window:** Displays a slide from a presentation. The title is "Введение Скелет программы Клавиатуры". The content includes a list of lecture topics such as "Лекция 1 Введение в сис...", "Лекция 2 Скелет оконной...", and "Лекция 3 Контекст устро...".
- Middle Window:** Displays the website for the "Кафедра ЭВМ" (Department of EEC). The page features a logo, a navigation menu on the left, a central photo of a man (Maxim Alexandrovich Volk), and links for "Методические указания", "Научные работы", and "Другие".
- Bottom Window:** Shows a blue dialog box titled "Цвет" (Color) with the text "СОБЫТИЕ!!!" (EVENT!!!).

The taskbar at the bottom shows the taskbar with icons for "Пуск" (Start), "Мой компьютер" (My Computer), and several open applications including "Microsoft PowerPoint", "Microsoft Word", and "Кафедра ЭВМ".

АКТИВНОЕ ОКНО



Активное окно – это окно, имеющее фокус ввода (input focus), либо имеющее дочернее окно, которое имеет фокус ввода.

WM_SETFOCUS - окно получает фокус ввода

WM_KILLFOCUS - окно теряет фокус ввода

Alt +

Классификация сообщений от клавиатуры



Типы аппаратных сообщений от клавиатуры

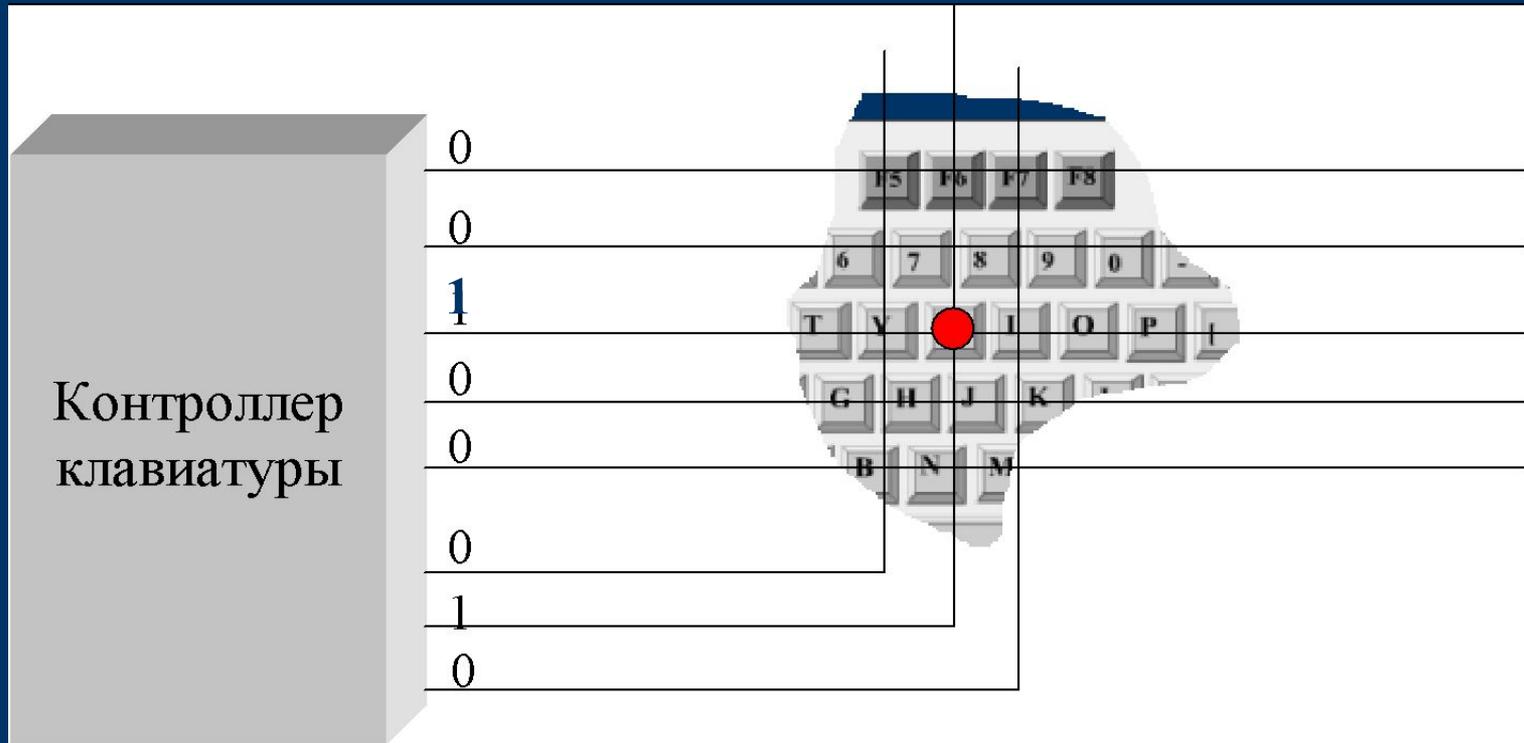


Типы сообщений	Клавиша нажата	Клавиша отпущена
Несистемные аппаратные сообщения	WM_KEYDOWN	WM_KEYUP
Системные аппаратные сообщения	WM_SYSKEYDOWN	WM_SYSKEYUP

Содержимое IParam



Скан - код клавиатуры



Содержимое IParam



Счетчик повторений. Равен числу нажатий клавиши, которое отражено в сообщении. В случае, когда его значение отлично от 1 (больше одного нажатия), это обычно означает, что программа не успевает обработать сообщения в реальном времени, либо система загружена в данный момент какой-либо работой. Ваша программа может как игнорировать число нажатий (реагировать только на сам факт нажатия), либо обрабатывать все нажатий клавиши клавиатуры.

Скан-код. Является кодом клавиатуры, генерируемым аппаратурой, то есть является тем кодом, который непосредственно приходит от клавиатуры. Обычно игнорируется приложением.

Флаг расширенной клавиатуры. Устанавливается в 1, если сообщение пришло от дополнительной клавиатуры (клавиши управления курсором, цифровая клавиатура и др.)

Код контекста. Код контекста устанавливается в 1, если нажата клавиша <Alt>. Часто, при помощи этого бита можно выделить системные сообщения.

Флаг предыдущего состояния клавиши. Равен 0, если в предыдущем состоянии клавиша была отпущена, и 1, если в предыдущем состоянии она была нажата.

Флаг состояния клавиши. Равен 0, если клавиши нажимается, и 1, если клавиша отпускается.

Второй параметр wParam содержит **виртуальный код клавиши** (virtual key code), идентифицирующий нажатую и отпущенную клавишу, чем реализуется аппаратная независимость программного представления клавиатуры.



Нажатая клавиша	Идентификатор, определенный в windows.h	Десятичный код
Ctrl-Break	VK_CANCEL	3
Tab (табуляция)	VK_TAB	9
Shift	VK_SHIFT	16
Enter	VK_ENTER	13
Ctrl	VK_CONTROL	17
Alt	VK_MENU	18
Esc	VK_ESCAPE	27
Пробел	VK_SPACE	32
Стрелка влево	VK_LEFT	37
Стрелка вправо	VK_RIGHT	38
Стрелка вниз	VK_DOWN	40
Стрелка вверх	VK_UP	39
Delete	VK_DELETE	46
End	VK_END	35
Home	VK_HOME	36
Page Up	VK_PRIOR	33
Page Down	VK_NEXT	34
F1	VK_F1	70

Получение времени нажатия клавиши и состояния управляющих клавиш.



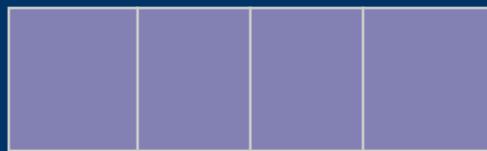
LONG GetMessageTime(void); // время возникновения
обрабатываемого сообщения

SHORT GetKeyState(VK_SHIFT); // состояние клавиш в момент
образования сообщения

SHORT GetAsyncKeyState(VK_SHIFT); // состояние клавиш в
момент настоящий момент

Возникновение символического сообщения

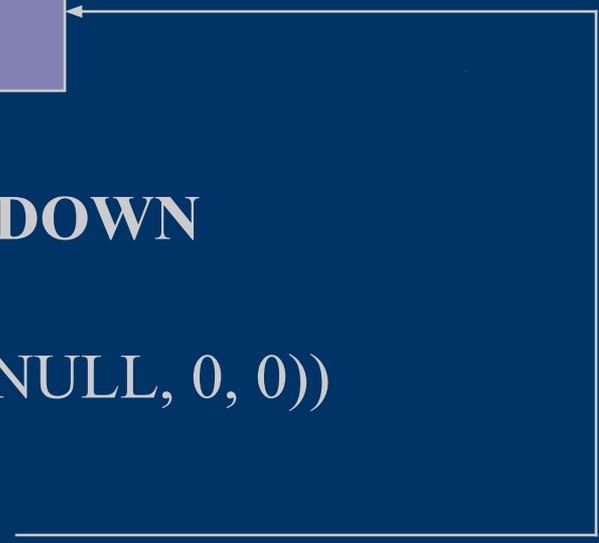
Очередь сообщений



WM_KEYDOWN

WM_CHAR

```
while (GetMessage(&msg, NULL, 0, 0))  
{  
    TranslateMessage(&msg);  
    DispatchMessage(&msg);  
}
```



Типы символьных сообщений



Типы сообщений	Клавиша нажата	Клавиша отпущена
Несистемные символьные сообщения	WM_CHAR	WM_DEADCHAR
Системные символьные сообщения	WM_SYSCHAR	WM_DEADCHAR

lParam - аналогично аппаратным сообщениям

wParam - содержит символьный код клавиши в системе ASCII

Пример 1

В качестве примера рассмотрим случай, когда пользователь программы нажимает и отпускает клавишу "А". Если переключатель <CapsLock> не включен и не нажата клавиша <Shift>, то оконная процедура получит три следующих сообщения:

Сообщение	Клавиша или код
WM_KEYDOWN	Виртуальная клавиша "А"
WM_CHAR	ASCII код "а"
WM_KEYUP	Виртуальная клавиша "А"

Пример 2

Если вы нажимаете "А" при нажатой клавише <Shift>, то оконная процедура получит следующий ряд сообщений:

Сообщение	Клавиша или код
WM_KEYDOWN	Виртуальная клавиша VK_SHIFT
WM_KEYDOWN	Виртуальная клавиша "А"
WM_CHAR	ASCII код "А"
WM_KEYUP	Виртуальная клавиша "А"
WM_KEYUP	Виртуальная клавиша VK_SHIFT

СОБЫТИЕ!!!



CreateCaret(hWnd, pImage, cxChar, cyChar); // создание каретки
SetCaretPos(cxChar, cyChar); // установить позицию каретки в позицию cxChar, cyChar
GetCaretPos(); // получить положение каретки
GetCaretBlinkTime(); // Получение частоты мигания
SetCaretBlinkTime(); // Установка частоты мигания
ShowCaret(hWnd); // показать каретку
HideCaret(hWnd); // спрятать каретку
DestroyCaret(); // удалить каретку