ПРАКТИЧНЕ ЗАНЯТТЯ 8
ВИКОРИСТАННЯ ПРИСТРОЇВ ВВОДУ/ВИВОДУ ІНФОРМАЦІЇ. КЛАВІАТУРА

8.1. Мета заняття:

ознайомитися з принципами роботи клавіатури на прикладі створення додатку, який за допомогою API-функцій створює повнофункціональне вікно на робочому столі.

8.2. Теоретичні положення

Клавіатура

Клавіатура: периферійний пристрій введення інформації.

Принцип роботи клавіатури:

Є три основних типи: мембранний, механічний і напівмеханічний.

Мембранна клавіатура – електронна клавіатура без окремих механічних рухомих частин, виконана у вигляді пласкої, зазвичай, гнучкої, поверхні з нанесеним на неї малюнком клавіш. Клавіатури цього типу відрізняються дуже низькою вартістю, винятковою компактністю (товщина складає долі міліметра), здатністю до згинання, високою надійністю і практично ідеальною захищеністю від бруду і вологи. Головним недоліком є майже повна відсутність тактильного зворотного зв'язку, що істотно ускладнює безпомилковий і сліпий набір.

У механічних клавіатурах для повернення клавіш використовуються металеві пружини.

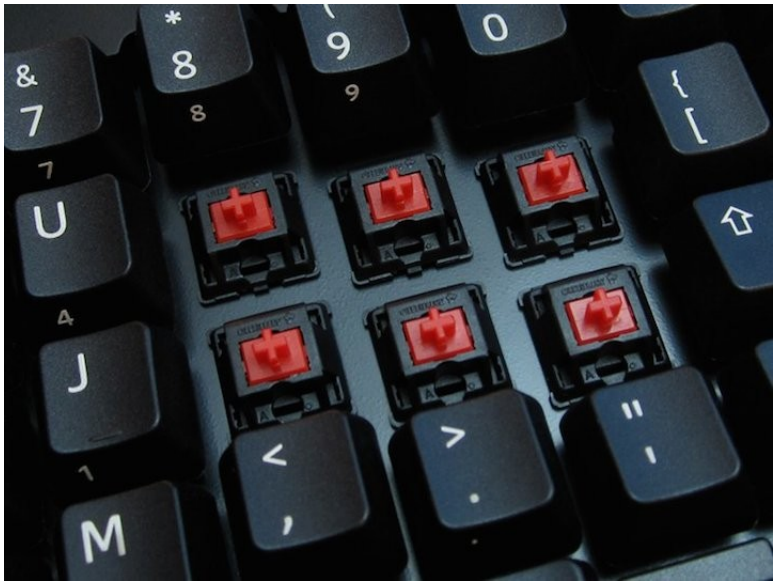

Рис. 1 Механічна клавіатура.

Напівмеханічні клавіатури це щось середнє між мембранними і механічними, де замість нижньої мембрани використовують друковану плату. Така конструкція вважається більш довговічною. Клавіша

повертається у вихідне положення також гумовим куполом. Іноді маленької пружинкою.

Лазерні клавіатури складаються з маленького проектора, який дозволяє вивести зображення клавіатури на будь-яку рівну поверхню. Передача даних бездротова. Можна налаштувати яскравість, звук друкування клавіш, чутливість. Правда це не гарантує стовідсоткове розпізнавання ваших рухів. Є і ще один мінус: клавіатуру не видно при яскравому освітленні.

У бездротових клавіатурах використовуються три основних види з'єднання: Bluetooth, інфрачервоне і радіочастотне.

Клавіатури з радіочастотним з'єднанням отримують живлення від акумулятора або через кабель USB, який використовується для підзарядки клавіатури. Клавіатури з інфрачервоним з'єднанням повинні знаходитися в радіусі дії пристрою, який приймає сигнал. Клавіатури з радіочастотним з'єднанням мають більший радіус дії, ніж клавіатури з інфрачервоним з'єднанням. У клавіатурах зі з'єднанням Bluetooth використовується технологія Bluetooth, яка забезпечує більший радіус дії, ніж у клавіатур з радіочастотним і інфрачервоним з'єднанням.

PS/2 і USB – два різновиди дротового з’єднання.

Порт PS/2 вперше з'явився у 1987 році на комп'ютерах IBM PS/2 (до цього для підключення клавіатури використовувався DIN-5. Швидкість передачі даних – від 80 до 300 Кб/с і залежить від продуктивності підключеного пристрою та програмного драйвера.

Принцип роботи клавіатури.

Мембранна клавіатури зазвичай складається з трьох шарів. На двох з них нанесені провідні доріжки. Третій, ізолюючий шар є розділяє. У місцях, де розташовуються клавіші, він має вирізи, які дозволяють доріжках верхнього і нижнього шарів торкатися одна одну при натисканні. Товщина шарів клавіатури, зазвичай, не більше товщини паперу або картону.

Усередині корпусу клавіатури крім датчиків розташовані електронні плати дешифрування сигналу.

Обмін даними між клавіатурою і системною платою здійснюється 11-бітовими блоками (8 розрядів плюс службова інформація) по 2-провідного кабелю (сигнал і земля).

Принцип роботи клавіатури полягає у скануванні перемикачів клавіш. Замиканню і розмиканню будь-якого з перемикачів відповідає унікальний цифровий код (scan code).

Прийом і обробку сигналів від клавіатури виконує спеціальна мікросхема – контролер клавіатури.

Якщо розглянути сильно спрощену принципову схему клавіатури, можна помітити, що всі клавіші знаходяться в вузлах матриці:

Всі горизонтальні лінії матриці підключені через резистори до джерела живлення +5 В. Клавіатурний «комп'ютер» має два порти – вихідний і вхідний. Вхідний порт підключений до горизонтальних ліній матриці (X0-X4), а вихідний – до вертикальних (Y0-Y5).

Встановлюючи по черзі на кожній з вертикальних ліній рівень напруги, відповідний логічному 0, клавіатурний «комп'ютер» опитує стан горизонтальних ліній. Якщо жодну клавішу не було натиснуто, рівень напруги на всіх горизонтальних лініях відповідає логічній 1.

Якщо оператор натисне на будь-яку клавішу, то відповідна вертикальна і горизонтальна лінії виявляться замкнутими. Коли на цій вертикальної лінії процесор встановить значення логічного 0, то рівень напруги на горизонтальній лінії також буде відповідати логічному 0.

Як тільки на одній з горизонтальних ліній з'явиться рівень логічного 0, клавіатурний процесор фіксує натискання на клавішу. Він посилає в центральний комп'ютер запит на переривання і номер клавіші в матриці. Аналогічні дії виконуються і тоді, коли оператор відпускає натиснуту раніше клавішу.

Номер клавіші, що посилається клавіатурним процесором, однозначно пов'язаний з розпаюванням клавіатурній матриці і не залежить безпосередньо від позначень, нанесених на поверхню клавіш. Цей номер називається скан-кодом (Scan Code).

Слово scan ("сканування"), підкреслює той факт, що клавіатурний комп'ютер сканує клавіатуру для пошуку натиснутої клавіші.

При натисканні на яку або клавішу електроніка клавіатури генерує скан-код клавіші довжиною від 1 до 6 байт, який можна отримати читанням порту введення-виведення 0x60. Скан-код – унікальне число, яке однозначно визначає натиснуту клавішу, але не ASCII-код. Скан-коди бувають двох видів: при натисканні клавіші генерується так званий Make-код, а при її відпусканні Break-код. Відрізняються вони лише тим, що у Break-коді старший біт кожного байта встановлений в одиницю. Скан-коди поділяються на групи: звичайні, розширені, додаткові і код клавіші Pause. Звичайний скан-код складається з одного байта, розширений – з 2-х байтів, перший з яких – 0xE0. Довжина додаткового коду – від 2 до 4 байт. Він так само починається з 0xE0. Дуже сильно з колії вибивається Pause – це єдина клавіша, код якої складається з 6 байтів, причому Break-код у неї відсутній.

Скан-коди клавіатур є апаратно залежними (має значення виробник, розкладка (наприклад: AZERTY, QWERTY) тощо).

Клавіатура може працювати в двох режимах: за опитуванням і за перериванням.

Проте програмні додатки потребують не скан-кодів, а ASCII-кодів (American Standard Code for Information Interchange).

Повідомлення.

Додатки Windows зобов'язані підтримувати для користувача введення з клавіатури, так як даний вид введення інформації є основним (спільно з мишею). Windows повністю підтримує для клавіатури ідеологію повідомлень, тобто будь-яка програма дізнається про натискання тієї чи іншої клавіші за допомогою повідомлень, які надсилаються віконній процедурі.

Повідомлення від клавіатури проходить дві черги перш, ніж потрапить до вашої програми – системну чергу повідомлень і чергу повідомлень додатка. З системної черги Windows вибирає повідомлення, призначені виключно їй (наприклад, перезавантаження <Ctrl+Alt+Del> або перемикання між додатками <Alt+Tab>). Таким чином, програма отримує тільки адресовані їй повідомлення від клавіатури.

Виникає питання: Windows – багатозадачне середовище і одночасно в системі можуть працювати декілька програм і, відповідно, відкриті кілька вікон, – яке ж із вікон буде отримувати повідомлення від клавіатури? Відповідь – вікно, яке має фокус введення ("input focus"). Концепція фокусу введення тісно пов'язана з концепцією активного вікна. Активне вікно – це вікно, яке має фокус введення, або має дочірнє вікно, яке має фокус введення. Тільки одне вікно в даний момент часу може бути активним. В цьому випадку, фокус введення належить або даному вікну, або одному з дочірніх вікон активного вікна (якщо вони існують).

Коли будь-яке вікно отримує від системи повідомлення WM_SETFOCUS, це означає, що вікно отримує фокус введення. Тепер всі повідомлення від клавіатури будуть надсилатися в це вікно. Вікно втрачає фокус введення, коли його віконна процедура отримує повідомлення WM_KILLFOCUS.

Програмі не потрібно реагувати на всі повідомлення від клавіатури, так як операційна система сама обробляє більшість клавіатурних повідомленнь (наприклад, всі ті, які починаються із префікса <Alt+>). Ці повідомлення будуть оброблені Windows і ваша програма отримає повідомлення, що є обробкою системного повідомлення (наприклад, повідомить вам, що вікно закривається, або вікно втрачає фокус введення).

Апаратні повідомлення.

Повідомлення, які додаток отримує від клавіатури, розрізняються на апаратні (keystrokes) і символьні (characters). Будь-яке натискання на клавіатурі генерує апаратне повідомлення. Натискання клавіші з будь-яким символом (буквою, цифрою, значком) має привести до того, що Windows, крім апаратного повідомлення, пошле в вашу програму символьне повідомлення.

Операційна система розрізняє у потоці апаратних повідомлень системні і несистемні повідомлення. Системні повідомлення зазвичай виробляються при натисканні клавіш у поєднанні з клавішею <Alt>. Ці повідомлення викликають опції меню програми або системного меню (<Alt+функціональна клавіша>, <Alt + Esc>), або використовуються для системних функцій, таких як зміна активного вікна (<Alt+Tab>). Зазвичай програма ігнорує системні повідомлення, проте, іноді виникає необхідність в їх обробці.

Типи повідомлень, які надходять у додаток від клавіатури, наведені у таблиці 8.1.

Таблиця 8.1 – Типи повідомлень.

Типи повідомлень	Клавіша натиснута	Клавіша відпущена
Несистемні апаратні повідомлення	WM_KEYDOWN	WM_KEYUP
Системні апаратні повідомлення	WM_SYSKEYDOWN	WM_SYSKEYUP

Зазвичай повідомлення про натискання і відпускання клавіші з'являються парами. Виняток становить той випадок, коли користувач не відпускає клавішу і включається автоповтор. В останньому випадку до програми направляється потік повідомлень про натискання клавіші.

Одночасно з одним з апаратних повідомлень надходять два параметри повідомлення: IParam і wParam. Змінна IParam складається з 6 полів (табл.8.2).

Таблиця 8.2 – Інтерпретація полів параметра IParam.

Розряди	Призначення	Опис
15…0	Лічильник повторень	Дорівнює числу натиснень клавіш, яке відображено в повідомленні. У випадку, коли його значення відмінне від 1 (більше одного натискання), це означає, що програма не встигає обробити повідомлення у реальному часі, або система завантажена в даний момент будь-якої роботою. Програма може ігнорувати число натиснень (реагувати тільки на сам факт натискання), або обробляти всі натискань клавіші клавіатури.
23…16	Скан-код	Є кодом клавіатури, який генерується апаратурою, тобто є тим кодом, який безпосередньо приходить від клавіатури. Зазвичай ігнорується додатком.
24	Прапор розширеної клавіатури	Встановлюється в 1, якщо повідомлення прийшло від додаткових клавіш (клавіші управління курсором, Alt s Ctrl правої сторони клавіатури та ін.)
29	Код контексту	Код контексту встановлюється в 1, якщо натиснута клавіша <Alt>. За допомогою цього біта можна визначити системні повідомлення.
30	Прапор попереднього стану клавіші	Дорівнює 0, якщо в попередньому стані клавіша була відпущена, і 1, якщо в попередньому стані вона була натиснута.
31	Прапор нового стану клавіші	Дорівнює 0, якщо клавіші натиснута, і 1, якщо клавіша відпущена.

Другий параметр wParam містить віртуальний код клавіші (virtual key code), який ідентифікує натиснуту і відпущену клавішу. Віртуальні коди клавіш мають імена, визначені в заголовних файлах Windows (таб.8.3).

Таблиця 8.3. Приклади віртуальних кодів.

Натиснута клавіша	Ідентифікатор, визначений в windows.h	Десятковий код
Ctrl-Break	VK_CANCEL	3
Tab (табуляция)	VK_TAB	9
Shift	VK_SHIFT	13
Enter	VK_ENTER	16
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

Слід зазначити, що віртуальні коди клавіш відрізняються залежно від версії операційної системи Windows. Наприклад, десятковий код, який відповідає визначеній клавіші у версії типу Home Premium, у версії типу Professional буде відповідати іншій клавіші або, взагалі, не буде задіяний.

Для отримання шістнадцяткового значення будь-якого символу, набраного з клавіатури, відповідно до таблиці кодування Unicode, необхідно виділити цей символ і натиснути "Alt + x".

Символьні повідомлення.

Знання про віртуальний код і положеннях керуючих клавіш недостатньо для визначення натиснутого символу. Операційна система дозволяє підтримувати велику кількість національних клавіатур, алфавіт яких може значно відрізнятися один від одного. У зв'язку з цим, натискання однієї і тієї ж клавіші може означати натискання різних символів, які визначаються обраним в даний момент драйвером клавіатури. Наприклад, натискання клавіші з буквою "V", при обраному драйвері російської або української клавіатури, означатиме натискання символу "М".

Дане перетворення виконує функція TranslateMessage, яка перетворює апаратні повідомлення на символьні, використовуючи при цьому стан керуючих клавіш і драйвер клавіатури.

Функція TranslateMessage є своєрідним диспетчером повідомлень. Якщо чергове повідомлення, вбране функцією GetMessage з черги повідомлень, є символьним, то TranslateMessage ставить в чергу повідомлень до додатка ще одне повідомлення – символьне. При цьому, апаратне повідомлення продовжує свій шлях у циклі обробки повідомлень і потрапляє до віконної процедури вікна програми через функцію DispatchMessage.

Аналогічно апаратним повідомленнями, існує чотири символьних повідомлення (таб. 8.4).

Таблиця 8.4. Типи символьних повідомлень.

типи повідомлень	клавіша натиснута	клавіша відпущена
Несистемні символьні повідомлення	WM_CHAR	WM_DEADCHAR
Системні символьні повідомлення	WM_SYSCHAR	WM_DEADCHAR

Повідомлення WM_CHAR є наслідком повідомлень WM_KEYDOWN. У більшості додатків кращим є використання саме символьних повідомлень. Параметр lParam, у випадку повідомлення WM_CHAR, є таким самим, як і відповідний параметр у апаратному повідомленні, а wParam – визначає код символу ASCII.

Як приклад розглянемо випадок, коли користувач програми натискає і відпускає клавішу "A". Якщо перемикач <CapsLock> не ввімкнений і не натиснута клавіша <Shift>, то віконна процедура отримає три наступних повідомлення:

WM_KEYDOWN Віртуальна клавіша "A"

WM_CHAR ASCII код "a"

WM_KEYUP Віртуальна клавіша "A"

Якщо ви натискаєте "A", утримуючи клавішу <Shift>, то віконна процедура отримає наступний ряд повідомлень:

WM_KEYDOWN Віртуальна клавіша VK_SHIFT

WM_KEYDOWN Віртуальна клавіша "A"

WM_CHAR ASCII код "A"

WM_KEYUP Віртуальна клавіша "A"

WM_KEYUP Віртуальна клавіша VK_SHIFT

Символьні повідомлення надходять до програми не тільки, коли користувач натискає клавіші з буквами. Наприклад, натискання табуляції призведе до приходу символьного повідомлення з кодом '\ t', повернення каретки - '\ r', забою - '\ b'.

Каретка.

Повертаючись до обговорення концепції "фокуса введення", згадаємо, що повідомлення клавіатури надходить до вікна, яке є активним і має "фокус

введення". Обробляючи повідомлення WM_SETFOCUS і WM_KILLFOCUS, програма може визначити, чи має вона фокус введення, чи ні.

Працюючи з реальними програмами, наприклад, з текстовими редакторами, необхідно визначити також те місце всередині вікна, куди необхідно здійснювати введення інформації. Коли ви набираєте текст, то, зазвичай, будь-якої символ (наприклад, прямокутник, або символ підкреслення) показує вам місце, де набираний символ з'явиться на екрані. Програма може сама стежити за малюванням, управлінням даними символом залежно від дій користувача. Проте, Windows надає зумовлений об'єкт, який реалізує перераховані функції, і носить ім'я "каретка" (caret).

Каретка є загальносистемних ресурсом (як і курсор, який пов'язаний з мишею), єдиним в операційній системі, і може використовуватися тільки вікном, яке має фокус введення. Тому, основним правилом використання каретки є наступне: віконна процедура викликає функцію CreateCaret при обробці повідомлення WM_SETFOCUS (отриманні фокусу вводу), і функцію DestroyCaret при обробці повідомлення WM_KILLFOCUS (втрати фокусу).

8.3. Контрольне завдання

Розглянути наведений код. Внести до нього зміни відповідно варіантам. Варіанти завдань наведені нижче.

```
.386
.model flat, stdcall
option casemap :none
include C:\masm32\include\windows.inc
include <\masm32\include\kernel32.inc>
include <\masm32\include\user32.inc>
includelib <\masm32\lib\kernel32.lib>
includelib <\masm32\lib\user32.lib>
include \masm32\include\masm32rt.inc
atoi PROTO C strptr:DWORD
.data
  msg1 db "1 chislo: ", 0
  msg2 db "2 chislo: ", 0
  ConsoleTitle db 'Lb4',0
formatStr db "Proizvedenie: %s*%s=%d", 0
.data?
  buffer1 dw 100 dup(?)
  buffer2 dw 100 dup(?)
  buffer3 dw 100 dup(?)
  buf db 100 dup (?)
  lens db ?
.code
start proc
  LOCAL hOutPut :DWORD
```

```
    LOCAL hInPut :DWORD
    LOCAL txtAtrib :DWORD
    call FreeConsole
    call AllocConsole
    invoke SetConsoleTitle, offset ConsoleTitle
    invoke GetStdHandle, STD_OUTPUT_HANDLE
    mov hOutPut, eax
    invoke GetStdHandle, STD_INPUT_HANDLE
    mov hInPut, eax
    mov EAX, 4h
    add EAX, 0h
mov txtAtrib, eax
    invoke SetConsoleTextAttribute, hOutPut, txtAtrib
    invoke SetConsoleCursorPosition, hOutPut, 655400
    push hOutPut
    lea eax, msg1
    push eax
    call StdOut
    push hInPut
    push 100
    lea eax, buffer1
    push eax
    call StdIn
    invoke SetConsoleCursorPosition, hOutPut, 655415
    push hOutPut
    lea eax, msg2
    push eax
    call StdOut
    push hInPut
    push 100
lea eax, buffer2
    push eax
    call StdIn
    ;invoke StdIn, offset buffer2, 100, hInPut
    invoke atoi, offset buffer1
    mov ebx,eax
    invoke atoi, offset buffer2
    imul eax,ebx
    invoke wsprintf, offset buffer3, offset formatStr, offset buffer1, offset buffer2,
eax
    invoke SetConsoleCursorPosition, hOutPut, 655430
    push hOutPut
    lea eax,buffer3
push eax
    call StdOut
```

```
WaitForCaps:
  invoke  GetAsyncKeyState, 12
  and     eax,    8000h
  jz      WaitForCaps
  invoke ExitProcess, NULL
start endp
end start
StdIn proc Buffer:DWORD,bLen:DWORD, InPut:DWORD
  LOCAL bRead  :DWORD
  invoke ReadFile,Input,Buffer,bLen,ADDR bRead,NULL
  mov eax, bRead
  ret

StdIn endp
StdOut proc Text:DWORD, OutPut:DWORD
   LOCAL bWritten :DWORD
   LOCAL sl       :DWORD
   invoke StrLen,Text
   mov sl, eax
   invoke WriteFile,OutPut,Text,sl,ADDR bWritten,NULL
   mov eax, bWritten
   ret
StdOut endp
```

1. Колір фону – білий, символів – чорний, позиція 10, 10, подія – натиснута клавіша Shift.
2. Колір фону – чорний , символів – білий , позиція 20, 10, подія – натиснуто лівий ALT
3. Колір фону – білий , символів – зелений , позиція 30, 10, подія – натиснута ліва кнопка миші.
4. Колір фону – чорний, символів – жовтий, позиція 40, 10, подія – натиснуто правий ALT.
5. Колір фону – жовтий, символів – червоний, позиція 10, 5, подія – натиснута права кнопка миші.
6. Колір фону – червоний, символів – зелений, позиція 20, 5, подія – натиснута клавіша CapsLock.
7. Колір фону – зелений, символів – чорний, позиція 30, 5, подія – натиснута клавіша Enter.
8. Колір фону – чорний, символів – білий, позиція 40, 5, подія – натиснута клавіша Shift.
9. Колір фону – синій, символів – чорний, позиція 10, 15, подія – натиснута клавіша Delete.
10. Колір фону – білий, символів – чорний, позиція 20, 15, подія – натиснута клавіша стрілка вгору.

11. Колір фону – зелений, символів – синій, позиція 30, 15, подія – натиснута клавіша F2.

12. Колір фону – білий, символів – синій, позиція 40, 15, подія – натиснута клавіша NumLock.

13.Колір фону – синій, символів – червоний, позиція 10, 20, подія – натиснута ліва кнопка миші.

14.Колір фону – чорний, символів – зелений, позиція 20, 20, подія – натиснута клавіша стрілка вниз.

15. Колір фону – білий, символів – чорний, позиція 30, 20, подія – натиснута клавіша стрілка вправо.

16. Колір фону – червоний, символів – чорний, позиція 40, 20, подія – натиснута клавіша стрілка вліво.

17. Колір фону – зелений, символів – жовтий, позиція 10, 5, подія – натиснута клавіша F1.

18. Колір фону – чорний, символів – білий, позиція 20, 5, подія – натиснута клавіша F3.

19. Колір фону – синій, символів – білий, позиція 30, 5, подія – натиснута клавіша F4.

20. Колір фону – білий, символів – чорний, позиція 40, 5, подія – натиснута клавіша F5.

21. Колір фону – синій, символів – чорний, позиція 10, 15, подія – натиснута клавіша F10.

22. Колір фону – зелений, символів – чорний, позиція 30, 5, подія – натиснута клавіша F9.

23. Колір фону – жовтий, символів – червоний, позиція 10, 5, подія – натиснута клавіша F8.

24. Колір фону – білий , символів – зелений , позиція 30, 10, подія – натиснута клавіша Ctrl.