МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ "ЛЬВІВСЬКА ПОЛІТЕХНІКА"

ІНСТИТУТ ПІСЛЯДИПЛОМНОЇ ОСВІТИ



**ЛАБОРАТОРНА РОБОТА №2**

на тему:

**“Програмування USB пристроїв”**

*з дисципліни:* ***“Програмування процесорних систем”***

Виконав:

слухач групи ПЗС-31

Гринчук Т.А.

Перевірив:

доцент Сенів М. М.

« »\_\_\_\_\_\_\_\_\_\_\_\_\_ 2015 р.

∑ \_\_\_\_\_\_\_\_\_\_\_\_\_\_

Львів – 2015

**Тема.** Програмування USB пристроїв.

**Мета.** Навчитися створювати програми для роботи з USB пристроями на мові асемблера.

**Теоретичні відомості**

Стандарт USB розробили сім компаній: «Compaq», «Digital Equipment», IBM, Intel, «Microsoft», NEC і «Northern Telecom». З листопада 1994 до листопада 1995 року було анонсовано кілька версій протоколу (USB 0.7, 0.8, 0.9, 0.99, 1.0 Release Candidate). Влітку 1996 року на ринку з'явилися перші комп'ютери з портами USB.

Шина USB — послідовний інтерфейс передавання даних для середньо- та низько швидкісних периферійних пристроїв. Для високошвидкісних пристроїв кращою вважалася шина FireWire, хоча з випуском пристроїв на базі USB 3.0 це твердження стало сумнівним.

USB-кабель — це, по суті, дві звиті пари: однією з них передаються дані в кожному напрямку (диференціальне включення), а інша використовується для живлення периферійного пристрою (+5 В, 500 мА). Вбудовані лінії живлення дозволяють використовувати USB-пристрої, що не мають власного блоку живлення, чи заряджати акумулятори переносних пристроїв (фото- та відеокамер, плеєрів тощо), якщо ці пристрої споживають струм силою до 500 мА. Стандарт USB 3.0 допускає навантаження лінії живлення струмом до 900 мА.

З'єднання USB-кабелями формує інтерфейс між USB-пристроями та USB-хостом. Як хост використовується керований з операційної системи USB-контролер, до складу якого входить USB-концентратор, або ж хаб. Цей хаб є відправною точкою в створенні ланцюжка пристроїв, що відповідають вимогами топології «зірка». Він має спеціальну назву — кореневий концентратор. До роз'ємів його портів під'єднується інше USB-приладдя та зовнішні хаби. Загальні їх кількість не може перевищувати 127 пристроїв, увімкнених не більш ніж у п'ять каскадів, не рахуючи рівень кореневого хаба.

Конструкція USB-конекторів розрахована на «гаряче» приєднання та від'єднання пристроїв до хоста (від нього). Це забезпечено більшою довжиною контакту заземлення GND проти інших. Внаслідок цього потенціали корпусів вирівнюються ще до замикання сигнальних контактів, а це убезпечує електроніку приладу від пошкодження статичною електрикою.

**Виконання практичної роботи:**

Створимо програму, яка після запуску буде виконувати перевірку чи USB-накопичувач приєднаний до USB-порта та виводити його назву логічного розділу в системі. Для цього створимо програмний файл з розширенням .asm та запрограмуємо необхідний код:

; Програма перевіряє, чи USB-накопичувач під'єднано

; до USB порта та показує назву логічного розділу диску

.386 ;асемблювання інструкцій процесора 80386

.model flat, stdcall ;32-бітна модель пам'яті

;оголошення заголовних файлів, які підключаються, макросів

option casemap :none ;розрізняє регістр символів

include c:\masm32\include\windows.inc ;підключаємо зовнішні файли

include c:\masm32\include\kernel32.inc

include c:\masm32\include\user32.inc

include c:\masm32\include\Comctl32.inc

include c:\masm32\include\shell32.inc

includelib c:\masm32\lib\kernel32.lib ;підключаємо бібліотеки

includelib c:\masm32\lib\user32.lib

includelib c:\masm32\lib\Comctl32.lib

includelib c:\masm32\lib\shell32.lib

CTEXT macro Text

local szText

.data

szText byte Text, 0

.code

exitm <offset szText>

endm

DlgProc PROTO :HWND,:UINT,:WPARAM,:LPARAM ;показ діалогового вікна

GetVolumeName PROTO :DWORD ;назва розділу

DEV\_BROADCAST\_HDR STRUCT ;інформація про події пристрою

dbch\_size DWORD ?

dbch\_devicetype DWORD ?

dbch\_reserved DWORD ?

DEV\_BROADCAST\_HDR ENDS

DEV\_BROADCAST\_VOLUME STRUCT ;інфоррмація про логічний розділ

dbcv\_size DWORD ?

dbcv\_devicetype DWORD ?

dbcv\_reserved DWORD ?

dbcv\_unitmask DWORD ?

dbcv\_flags DWORD ?

DEV\_BROADCAST\_VOLUME ENDS

.const

IDD\_DIALOG equ 1000

IDC\_INFO equ 101

DBT\_DEVICEARRIVAL equ 8000h

DBT\_DEVICEREMOVECOMPLETE equ 8004h

DBT\_DEVTYP\_VOLUME equ 2h

.data

szDrivePath db "%c:\%s",0

.data?

hInstance dd ?

szBuf db 16 dup(?)

.code

\_start:

invoke GetModuleHandle,NULL

mov hInstance,eax

invoke InitCommonControls

invoke DialogBoxParam,hInstance,IDD\_DIALOG,NULL,addr DlgProc,NULL

invoke ExitProcess,0

DlgProc proc hWnd:HWND,uMsg:UINT,wParam:WPARAM,lParam:LPARAM

local dbt\_h : DEV\_BROADCAST\_HDR

local dbt\_v : DEV\_BROADCAST\_VOLUME

.if uMsg==WM\_INITDIALOG ;ініціалізація діалогу

.elseif uMsg == WM\_DEVICECHANGE

.if wParam == DBT\_DEVICEARRIVAL ;знайдено новий пристрій

mov ebx, lParam

assume ebx:ptr DEV\_BROADCAST\_HDR

.if [ebx].dbch\_devicetype == DBT\_DEVTYP\_VOLUME ;якщо пристрій - логічний диск

invoke SetDlgItemText,hWnd, IDC\_INFO,CTEXT ("USB-Drive found") ;виводимо повідомлення

mov edx,lParam

assume edx:ptr DEV\_BROADCAST\_VOLUME

invoke GetVolumeName,[edx].dbcv\_unitmask ;

push eax ; поміщаємо в стек

invoke wsprintf,offset szBuf,offset szDrivePath,eax,0

invoke GetDriveType,offset szBuf

.if eax == DRIVE\_REMOVABLE

invoke Sleep, 1500 ; очікуємо

pop eax ;витягти з стеку в eax

invoke SetDlgItemText,hWnd, IDC\_INFO,offset szBuf

.endif

assume edx:nothing

.endif

assume ebx:ptr nothing

.elseif wParam == DBT\_DEVICEREMOVECOMPLETE ;якщо пристрій відключено

invoke SetDlgItemText,hWnd, IDC\_INFO,CTEXT ("USB-Drive Removed") ;виводимо повідомлення

.endif

.elseif uMsg==WM\_COMMAND

.elseif uMsg==WM\_CLOSE

@Exit:

invoke EndDialog,hWnd,0 ; завершення діалогу

.else

mov eax,FALSE

ret

.endif

mov eax,TRUE

ret ;Повернення з процедури

DlgProc endp

GetVolumeName proc flag:DWORD ;визначити ім'я логічного диску

push esi

push ecx

mov ecx, 26

mov esi,flag

@@:

test esi,1h

jnz @f

shr esi,1h

dec ecx

test ecx,ecx

jnz @b

@@:

mov eax,5Bh

sub eax,ecx

pop ecx

pop esi

ret ;Повернення з процедури

GetVolumeName endp

end \_start ;кінець програми

**Виконання програми:**

Скопіюємо програмний файл у директорію компілятора Macro Assembler (MASM) - C:\masm32\bin. Запустимо консоль Windows та перейдемо в директорію.

Виконаємо асемблювання програми, тобто перетворення її в машинні коди, ввівши наступну команду:

*ml.exe /c /coff usb.asm*

Ключ */c* задає параметр для асемблювання програми без компонування;

параметр /coff – генерує об’єктний файл в COFF форматі.

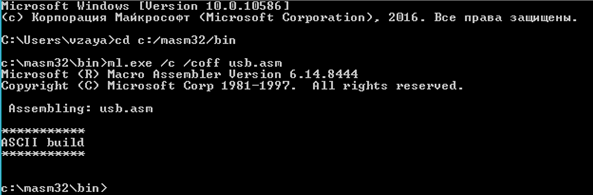


Рис. 1. Асемблювання програми.

Далі виконаємо компонування програми, зібравши у виконувану програму з розширенням .exe, яку можливо запустити у середовищі Windows (рис. 3).

*link.exe usb.obj /subsystem:windows*

Параметр */subsystem:windows –* визначає середовище, в якому повинна виконуватися програма.



Рис. 2. Компонування програми.

Запустимо наш скомпонований файл usb.exe (рис. 3).

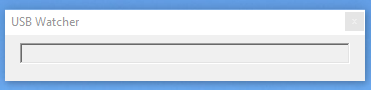


Рис. 4. Запущений діалог програми.

Під’єднаємо USB-накопичувач до USB-порту комп’ютера. Наша програма вказала у текстовому полі назву логічного розділу щойно під’єднаного USB-накопичувача (рис. 4).

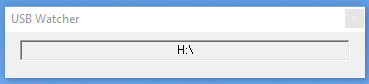


Рис. 4. Назва логічного розділу під’єднаного накопичувача.

**Висновок:**

Виконуючи дану лабораторну роботу, я навчився створювати програму для роботи з USB пристроями на мові асемблера. Програма дозволяє виконувати перевірку чи USB-накопичувач під’єднаний до USB-порта та виводити його назву логічного розділу. Після чого виконано асемблювання програми, тобто перетворення її в машинні коди та компонування програми.