

Lab - 2

Input and Output

Introduction:

In this experiment you will be introduced to the basic Input and Output (I/O) operations using assembly language. You will use the DOS interrupt (INT 21H) function calls to access the keyboard and video display. More details will also be given on the structure of an assembly language program.

The following major points are discussed:

- Variable declaration using: DB, DW, DD
- Constant declaration using: EQU
- OFFSET operator
- INT 21H with the functions 1, 2, 8 and 9.

Objectives:

- 1-Demonstrate keyboard access using the DOS INT 21H function calls 01, 02 and 08.
- 2-Demonstrate string display using the DOS INT 21H function call 09.
- 3-Show the difference between keyboard read functions, with echo and without echo.

I/O DOS Function Calls:

Table 2.1 summarizes the main I/O functions. These functions are mainly used to read a character or a string from the keyboard, which could be an input data to a program, and display characters or strings, which could be results, or an output, of a program:

Function	Input in	Output in	Effect
01H	AH	AL	Read a character with echo on the screen.
02H, 06H	AH, Character in DL	No output	Display a character on the screen. Note: Interrupted by Ctrl + Break
08H	AH	AL	Read character without echo.
09H	AH	No output	Display a string terminated by a '\$' sign
0AH	AH	Offset in DX	Read a string of characters from the keyboard

Table 2. 1: Simple I/O DOS function calls

DOS Display Functions:

These are DOS functions 02 and 06 for a single character display, and 09 for a string display.

DOS Functions 02 and 06:

Both functions are identical, except that function 02 can be interrupted by a control break (Ctrl-Break), while function 06 cannot. To display a single character ASCII character at the current cursor position use the following sequence of instructions:

```
MOV AH, 06H           ; (Or: MOV AH, 02H)
MOV DL, Character Code
INT 21H
```

The Character Code may be the **ASCII** code of the character taken from the ASCII table or the character itself written between quotes.

The following displays number 2 using its ASCII code:

```
MOV AH, 06H
MOV DL, 32H
INT 21H
```

This code also displays 2:

```
MOV AH, 06H
MOV DL, '2'
INT 21H
```

DOS Functions 09:

This function is used to display a string of characters ended with a '\$' sign. The following code displays the string MESSAGE defined as:

```
MESSAGE DB 'This is the Message to be displayed', '$'

.CODE
MOV DX, OFFSET MESSAGE
MOV AH, 09H
INT 21H

Or

MESSAGE DB 'This is the Message to be displayed', '$'
.CODE
LEA DX, MESSAGE
MOV AH, 09H
INT 21H
```

DOS Input functions:

These include reading a single character, with or without echo, functions 01 and 08, and reading a whole string.

Function 01H and 08H INT 21H:

To read single character and have it echoed (displayed) on the screen, use the following code:

```
MOV AH, 01H
INT 21H
;AL contains now the ASCII code of the character read from the
;keyboard.
```

If the character is to be read without echo, such as reading a password, use the following code:

```
MOV AH, 08H
INT 21H
AL contains now the ASCII code of the character read
```

Reading a String:

Reading a string is accomplished by Function 0AH INT 21H. DOS function 0AH will accept a string of text entered at the keyboard and copy that string into a memory buffer. DOS 0AH is invoked with DS:DX pointing to an input buffer, whose size should be at least three bytes longer than the largest input string anticipated.

Before invoking DOS function 0AH, you must set the first byte of the buffer with the number of character spaces in the buffer. After returning from DOS function 0AH, the second byte of the buffer will contain a value giving the number of characters actually read from the keyboard.



Figure 2. 1: Keyboard buffer structure

Function 0AH	Read from Keyboard
Entry	0AH ; DX = address of keyboard input buffer First byte of buffer contains the size of the buffer (up to 255)
Exit	Second byte of buffer contains the number of characters read. Reading operation continues until buffer full, or a carriage return (CR = 0DH) is typed.

Table 2. 2: Functions 0AH of DOS interrupt.

Example:

Below is an example on the use of function 0AH, when the user enters the word “hello”.

Input:

XX XX XX XX XX XX XX XX XX									
0	X	X	X	X	X	X	X	X	X
8	X	X	X	X	X	X	X	X	X
MOV AH, 0AH									
INT 21H ;Read from keyboard the word “hello”									

Output:

0	0	6	6	6	6	0	X	X	X
8	5	8	5	C	F	D	X	X	X

Lab Work:

- 1- Assemble and Link program 1.
- 2- Type the program's name at the prompt to run the program.
- 3- What does the program do? Notice how the program handles the three different characters.
- 4- Assemble, Link and Run program 2
- 5- Replace the line: MOV DX, OFFSET MESSAGE
by: LEA DX, MESSAGE
Then repeat step 4, what do you notice?
- 6- Check with CodeView the effects of the instruction LEA and the OFFSET operator?
- 7- Assemble, Link and Run program 3
- 8- After running the program, notice here the effect of the characters 0DH and 0AH at the end of the line containing: MESSAGE. What is your conclusion?
- 9- Notice also the effects of the function calls 01H, 08H.
- 10- Write down all your conclusions.

Lab Assignment:

Write an assembly language program that prompts you to enter a password of 3 characters in length. The password should not be echoed to the screen. The program then displays your name and ID number on the screen.

Submit your work at the end of the lab.

This program displays the characters A B C, using INT 21H function 02.

```
.MODEL SMALL
.DATA
X EQU 'B'
Y DB 43H

.STACK 200

.CODE

MOV AX, @DATA
MOV DS, AX

MOV AH, 02      ; LOAD FUNCTION 02

MOV DL, 'A'      ; LOAD CHARACTER TO BE DISPLAYED
INT 21H          ; CALL INTERRUPT 21H

MOV DL, X        ; LOAD CHARACTER TO BE DISPLAYED
INT 21H          ; CALL INTERRUPT 21H

MOV DL, Y        ; LOAD CHARACTER TO BE DISPLAYED
INT 21H          ; CALL INTERRUPT 21H

MOV AX, 4C00H    ; Exit to DOS
INT 21H
END
```

PROG#2

This program displays a string terminated by a \$ sign using INT 21H function 09H.

```
.MODEL SMALL

.DATA
    MESSAGE DB 'This is the message to be displayed', '$'

.STACK 200

.CODE
    MOV AX, @DATA
    MOV DS, AX
    MOV DX, OFFSET MESSAGE
    MOV AH, 09H
    INT 21H
    MOV AX, 4C00H ; ; Exit to DOS
    INT 21H
END
```

PROG#3

Character input with echo INT 21H, function call 01H
; Character input without echo INT 21H, function call 08H

.MODEL SMALL

.DATA

MESSAGE DB 'Enter a character: ','\$'

MESSAGE2 DB 'The character you typed is: ',0DH, 0AH,'\$'

.STACK 200

.CODE

MOV AX,@DATA

MOV DS,AX

LEA DX, MESSAGE

MOV AH,09H

INT 21H ; Display message

MOV AH,02 ; Function 02H, display character

MOV DL,AL ; Load character to be displayed

INT 21H ;

LEA DX, MESSAGE

MOV AH,09H

INT 21H

MOV AH, 08H ; Function read character without echo.

INT 21H ; Character read is returned in AL register. No echo on the display.

MOV BL,AL ;Save character read in BL register

LEA DX, MESSAGE2

MOV AH,09H ;Display MESSAGE2

INT 21H

MOV AH,02 ; Function 02H, display character

MOV DL,BL ; Load character to be displayed from BL

INT 21H

MOV AH,4CH ; Exit to DOS

INT 21H

END