

## Lab. 1: DOS Interrupts (Int 21H)

### Objectives:

---

Introduce the student to:

- DOS functions 01H , 02H, 08H, and 09H, 40H, and 0AH.
- Using the above interrupts functions in programming examples.

### The Interrupt instruction

---

DOS includes several built in programs for various systems related operations. A user can also use these programs by using special software Interrupt instruction called **INT**. The INT instruction calls a DOS interrupt service routine (like a function) to perform a special task. For example, to read from the keyboard or disk or mouse, or to write a character to the screen we use special DOS functions.

General syntax for using interrupt service routine is shown below:

INT Interrupt Number

MS-DOS Operating system provides many common services through INT 21h. INT 21h MS-DOS services are procedures that provide input-output, file handling, and memory management. They are also called “MS-DOS function calls.”

The execution of an INT instruction causes an Interrupt Service Routine (ISR) associated with the *InterruptNumber* to be executed. Many of the ISRs have multiple sub-functions.

To specify which sub-function is to be executed under a particular InterruptNumber, the AH register is assigned a sub-function number before the execution of the INT instruction.  
Example:

```
MOV AH, 08H
INT 21H
```

causes sub-function number 08H of Interrupt number 21H to be executed. In addition, some sub-functions require other values to be passed to the ISR in particular registers.

Example:

Sub-function 09H of Interrupt 21H displays a \$-terminated string on the screen. The sub-function requires the offset of that string to be passed in the DX register:

```
MOV DX , OFFSET STRING
MOV AH , 09H
INT 21H
```

## DOS Function Calls (INT 21H)

DOS function calls preserve the contents of all the registers except the AX register and any other register or registers in which they explicitly return data.

### 1. Terminate program and return to DOS

Every time you want to terminate the program and return to DOS, you have to put the following codes:

```
mov AX , 4C00H
int 21h
```

### 2. Character input with echo

To take single input character thru a keyboard, you have to put the following codes:

The Codes	The Result
mov AH, 01h int 21h	The program is waiting for the input. Once a user presses a key, the ASCII Code of the input character is returned in the <b>AL</b> register and the input character is displayed as well.

NOTE: This service makes the program waits for the input. The user just needs to press the intended key WITHOUT pressing "enter" key.

### 3. Character input without echo

```
MOV AH , 08H
INT 21H
```

The code of the input character is returned in the AL register.

### 4. Character output

To display a character, you have to use the DOS function 02h.

The Initial requirement	The result
AH = 02h DL = <i>Character or ASCII Code</i>	The character stored in DL will be displayed.

#### **Example:**

The following code fragment will display a string 'Hey'.

```
mov DL, 'H'
mov AH, 2
int 21h
mov DL, 'e'
mov AH, 2
int 21h
mov AH, 2
mov DL, 'y'
int 21h
```

**Example:**

The following code fragment will move the cursor to the beginning of the next output line:

```

mov AH, 02H
mov DL, 0DH
int 21H
mov DL, 0AH
mov AH, 2
int 21H

```

**5. Displaying a string**

There are two ways to display a string.

**5.1. Display a string using (Service 09H)**

The DOS function 09h display a string that terminated by '\$'.

The initial requirement	The result
1.The string must be defined in DATA segment.	The terminating \$ is not displayed, even if it appears within the string. Thus this function cannot be used to display the \$ character on the screen.
2.The string must be terminated by '\$'.	
3.AH = 09h	
4.DX = Offset address of the beginning of the string	

**Example:**

```

DATA
. . .
STRING_NAME DB 'THE STRING TO BE DISPLAYED$'
. . .
CODE
MOV AH , 09H
MOV DX , OFFSET
STRING_NAME INT 21H

```

**5.2. Displaying a string using (SERVICE 40h)**

Another way to display a string:

The initial requirement	The result
AH = 40h	Display the string begins at the address specified in DX with length given in CX.
BX = 1	
CX = string length	The value in BX determines the device/file where the string will be displayed.
DX = offset address of the beginning of the string	

## 6. Reading a String (SERVICE 0Ah)

To read a string, we have to do two steps:

1. Define an array to store that string in DATA segment.
2. Invoke DOS function 0AH in CODE segment.

One way to define an array of string is as follow:

```
BUFFER_NAME          DB      Num1, Num2  DUP (?)
```

where:

- a) **Num1** = the maximum number of string characters to be read + **1**
- b) **Num2** = **Num1** + **1**
- c) The maximum value for **Num1** is FFh i.e., 255
- d) The last string character is the Carriage Return (0Dh).

The program will wait for the input. The user must press "Enter" key to end the inputting process. This is the reason for the point d above.

## Exercise:

Write a program that read your name and age and your grade (A,B,C,D,or F) from the keyboard and display them back on the screen?