**PRACTICAL # 02**

**OBJECT:**

Introduction to interrupts, using interrupt to output a character.

**THEORY:**

**Interrupt** is a signal to microprocessor emitted by hardware or software indicating an event that needs immediate attention. An interrupt alerts the processor to a high-priority condition requiring the interruption of the current code the processor is executing (thread). The processor responds by suspending its current activities, saving its state and executing a function called **Interrupt Handler** (or interrupt service routine) specific to the event. Interrupt handlers are usually short functions that handle the interrupt. After the interrupt handler finishes, the processor resumes execution of the interrupted thread.

**Types of interrupts**

**1) Hardware Interrupt** is a signal sent to the processor from some device, either a part of the computer itself such as a disk controller or an external IO device. Pressing a key on keyboard or moving the mouse triggers hardware interrupts that cause the processor to read the keystroke or mouse position. Unlike the software interrupts, hardware interrupts are asynchronous. The act of initiating a hardware interrupt is referred to as an interrupt request (IRQ).

**2) Software Interrupt** is caused either by an exceptional condition in the processor itself, or a special interrupt instruction in the code which causes an interrupt when it is executed. The former is often called a ***trap***and is used for errors or events occurring during program execution that are exceptional enough that they cannot be handled within the program itself.

If there is an instruction to divide a number by zero, will cause a ***divide-by-zero exception***, causing the system to abandon the calculation and display an error message. Software interrupt instructions function similarly to a function call and are used for a variety of purposes, such as to request services from low-level system software such as device drivers. For example, computers often use software interrupt instructions to communicate with the disk controller to request data be read or written to the disk.

In our programs we will use DOS interrupts which are software interrupts. There are numerous DOS interrupts available for functions like getting keyboard or mouse inputs, displaying output on screen or printer, drawing graphics and much more. These Interrupts usually end with **iret** instruction short for interrupt return.

**Program:**

The first program is a basic character output program that displays a character on the screen. The **int 21H** is set of DOS interrupts containing a range of interrupts that cover the I/O functions. The interrupt number to use from this set of interrupts is set in **AH** register before the **int 21h** instruction. The interrupt number 2 is used to output a character to the standard output device, the screen.

.MODEL SMALL

.STACK 100H

.DATA

.CODE

MAIN PROC

MOV DL, ‘A’

MOV AH, 2

INT 21H

MOV AH, 4CH

INT 21H

MAIN ENDP

END MAIN

The DL register contains the value of the character to be output. The interrupt number is filled in register AH. Finally the interrupt is called to execute the operation.

**ACTIVITIES**

**Activity 1**

Write the above program in a text editor of your choice. Save the program with .asm extension in the programs in MASM folder. Now in DOS and navigate to the MASM directory. Build the program as ml myprogram.asm. if you don’t have ml exe file in your MASM folder, assemble and link individually using following commands.

**MASM.exe program.asm**

**LINK program.obj**

Note after issuing each command, it will ask for output file names, just continue pressing enter to generate default file names.

**REVIEW QUESTIONS**

1. What is interrupt?
2. What is the difference between hardware interrupt and software interrupt?
3. How does software interrupt work?
4. What is the last instruction typically found in interrupt handler function?

1. What is the purpose of the function **4CH** in **int** **21h**?