**Computer Organization & Assembly Language**

**Lab 09**

**Topics:**

* Interrupts

**Interrupt**

* An interrupt is an event that causes the processor to suspend its present task and transfer control to a new program called the interrupt service routine (ISR).
* Three sources of interrupts
* Software interrupts
* Hardware interrupts
* Processor interrupts

**Common Software Interrupts:**

The Common Interrupts are

**GetStdHandle PROTO**

**WriteConsoleA PROTO, a1:DWORD, a2: PTR BYTE, a3: Dword, a4: ptr dword, a5: dword**

**ReadConsoleA PROTO, a1:DWORD, a2: PTR BYTE, a3: Dword, a4: ptr dword, a5: dword**

**invoke ReadConsoleA, eax, offset buffer, lengthof buffer, offset x, 0**

1. **Procedures (uses keyword):**

Procedures can also be used in a nested fashion. Any procedure can be called inside any other procedure. In this way the stack will save all the values of instruction pointers and are retrieved in the way of last in first out.

Keyword uses can be used if a register is to be used inside a procedure.

Example:

Array\_Sum proc uses si, cx

<instructions>

ret

Array\_Sum endp

1. **Macros:**

* Macros are just like procedures, but they exist only until your code is compiled, after compilation all macros are replaced with real instructions.
* Macro Definition
  + name MACRO [parameters,...]
  + <instructions>
  + ENDM

**Using Macros:**

* When you want to use a macro, you can just type its name. For example:
  + MyMacro
* Macro is expanded directly in program's code. So if you use the same macro 100 times, the compiler expands the macro 100 times, making the output executable file larger and larger, each time all instructions of a macro are inserted.

**Passing Arguments to Macro:**

* To pass parameters to macro, you can just type them after the macro name. For example:
  + MyMacro 1, 2, 3
* To mark the end of the macro ENDM directive is enough

Example:

* Unlike procedures, macros should be defined above the code that uses it.
* For Example
  + .code
  + mymacro macro p1,p2,p3
  + mov ax,p1
  + mov bx,p2
  + mov cx,p3
  + endm
  + main proc
  + mymacro 1,2,3
  + mov ah,4ch
  + int 21h
  + main endp
  + end

**Defining Macros in Separate file:**

* To define Macros in Separate file;
  + Open your assembler Source Directory
    - C:\masm615\include\
  + Create a File named “mymacros.inc”
  + Write your Macro in this file and save. Make sure your file have extension .inc
  + Include this file in your source program (\*.asm), by writing below line on top of your code
    - include mymacros.asm

Compile your Code.

**TASK**

1. Write Assembly program to Input Character from user and display it.
2. Write Assembly program to Input upper case letter from user and display lowercase.
3. Create an Array of 10 elements of size BYTE. Ask user to input index of an element and print that element and get the next element i-e (index + 1)
4. Write an assembly program that takes an input and tells whether the given number is even or odd
5. Write Assembly program to input letter from the user and display the previous character.
6. Take the marks from user and display the letter grade. (Use the following table)

|  |  |
| --- | --- |
| a < 5 | D |
| 5 ≤ a <7 | C |
| 7 ≤ a < 9 | B |
| =9 | A |

1. Write a program that displays the following output.
2. \*\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*

b) \*

\*\*

\*\*\*

\*\*\*\*

\*\*\*\*\*