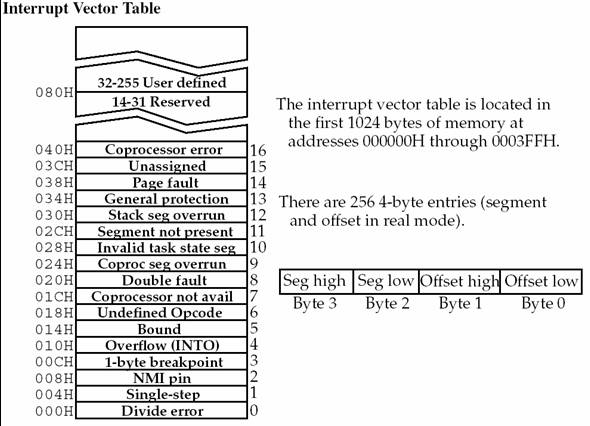
**Lab 13: Interrupt Handling, Macros and Structures**

**Objective:**

* Interrupt Handling
  + Interrupt Vector Table
  + Exceptions, Traps and Interrupts
  + Divide by Zero Exception
  + Overflow Exception
* Macros

**Interrupt Vector Table:**



**Replacing Existing Handler with user defined Handler:**

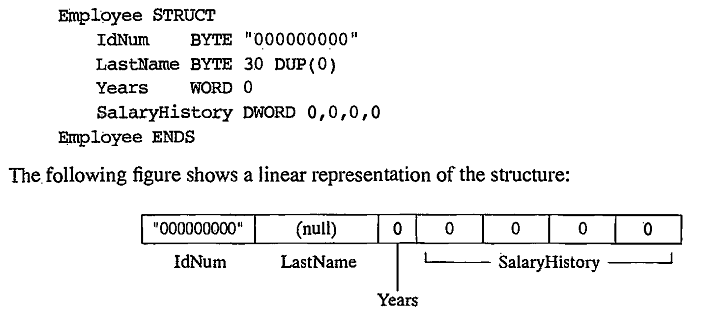
* int 21h/25h
  + ds:dx contains address of new function
  + ‘al’ register should contains Interrupt Number for which Handler/Function is to be changed.
* Example
  + mov dx, seg myfunc
  + mov ds, dx
  + mov dx, offset myfunc
  + mov al, 00h; Int 0h is triggered for Divde Zero Exception
  + mov ah, 25h
  + int 21h

**Structures**:

* Defining
  + Defined using ***struc*** and ***ends*** directives
  + Inside the structure, you define fields using the same syntax as for ordinary variables.
* Basic Syntax
  + name STRUCT
  + *data declaration*
  + nameENDS

**Example:**

* Defining a Structure Employee information with different fields as below



**Declaring Structure Variables:**

* Under Data Segment
* .data
  + worker Employee <>

**Field Referencing:**

* To Access Structure Variable Fields, Under Code Segment
* .code
  + mov dx, worker.years
  + mov worker.salaryhistory,20000

**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:\masm611\include\
    - C:\emu8086\inc\
  + 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.

