

Microprocessor-Based Systems Labs

Lab Session 1: Addressing modes. Directives and operators.

The aim of this session lab is to prepare three different programs so as to obtain a better understanding of the different addressing modes and the memory directives set used in the 8086 processors.

Program labs1a.asm

Write a program that includes the following operations:

- Load 13H in AX
- Load BAH in BX
- Load 3412H in CX
- Load the content of CX in DX
- Load in AL the content of the memory address 65246H and in AH the content of the memory address 65247H
- Load in the memory address 40004H the content of CH
- Load in AX the content of the memory address pointed by DI
- Load in AX the content of the memory address pointed by BP + 8 bytes.

Program labs1b.asm

Write a program including the following data structures:

- Reserve memory for 1 byte length variable named COUNTER
- Reserve memory for 2 bytes length variable called GRAB, initialized with the value CAFEH.
- Reserve 100 bytes for a table called TABLE100
- Store in memory the text string ERROR1 previously initialized with "Incorrect data. Try again".

With the data structure previously defined, please include the following instructions:

- Copy the 6th character of the string ERROR1 in the position 53H of TABLE100
- Copy the content of the variable GRAB just after the position 22H of TABLE100
- Copy the most significative byte of GRAB into the variable COUNTER.

Program labs1c.asm

Assuming DS=0511H, BX=0211H & DI=1010H, determine the REAL memory address where the following instructions will access:

- a) MOV AL,DS:[1234H]
- b) MOV AX,[BX]
- c) MOV [DI],AL

Write a program to verify the results from the TD. The student is free to initialize the contents judged necessary. The final address expected shall be included in the source file as comments

DELIVERY: Date and contents.

Upload to Moodle a ZIP file containing **only the makefile and the source files** (.asm) of the exercises. Remember that only one member of the team can upload the file and the files shall contain the authors' name and the team number **in the header**.

Notice that the source files shall be correctly tabulated and commented. The lack of comments or poor quality ones will be qualified negatively.

The date limit to upload the files is March the 15th at 23:55h

The student may use the following program as a template:

```
;*****
; ASSEMBLY CODE STRUCTURE EXAMPLE. MBS 2018
;*****
; DATA SEGMENT DEFINITION
DATOS SEGMENT
    ;-- complete with the data requested
DATOS ENDS
;*****
; STACK SEGMENT DEFINITION
PILA SEGMENT STACK "STACK"
    DB 40H DUP (0) ; initialization example, 64 bytes set to 0
PILA ENDS
;*****
; EXTRA SEGMENT DEFINITION
EXTRA SEGMENT
    RESULT DW 0,0 ; initialization example. 2 WORDS (4 BYTES)
EXTRA ENDS
;*****
; CODE SEGMENT DEFINITION
CODE SEGMENT
ASSUME CS: CODE, DS: DATOS, ES: EXTRA, SS: PILA
; BEGINNING OF THE MAIN PROCEDURE
INICIO PROC
    ; INITIALIZE THE SEGMENT REGISTERS
    MOV AX, DATOS
    MOV DS, AX
    MOV AX, PILA
    MOV SS, AX
    MOV AX, EXTRA
    MOV ES, AX
    MOV SP, 64 ; LOAD THE STACK POINTER WITH THE HIGHEST VALUE
    ;
    ; PROGRAM START
    ; -- to be completed with the instructions requested
    ; PROGRAM END
    MOV AX, 4C00H
    INT 21H
INICIO ENDP
; END OF CODE SEGMENT
CODE ENDS
; END OF PROGRAM. OBS: INCLUDES THE ENTRY OR THE FIRST PROCEDURE (i.e. "INICIO")
END INICIO
```