Escuela Politécnica Superior

UAM

Lab Session 2: The instruction Set.

Types of basic instructions:

- Data transfer
- **Arithmetic operations**
  - **ADD, SUB, INC, MUL, IMUL, IDIV..**
- **Logic operations**
  - **AND, OR, NOT, TEST…**
- Control transfer
- **Interrupts**
  - **INT, RET**
- Activation of flags

Arithmetic operations

MUL: Multiply the operand by **AX** (8-bit operand) or the pair **DX**:**AX** (16-bit operand).
MUL operand

IMUL: Multiply with sign.
IMUL operand

DIV: Divide **AX** (8-bit operand) or **DX**:**AX** (16-bit operand) by the unsigned operand.
Quotient in **AL** and remainder in **AH** (8-bit operand).
Quotient in **AX** and remainder in **DX** (16-bit operand).
DIV operand

IDIV: Divide with sign.
IDIV operand

imul Signed multiply

Syntax:
imul op8
imul op16

op8: 8-bit register or memory
imul Bl
imul TABLA[9] (DB elements based)

op16: 16-bit register or memory
imul BX
imul TABLA[8] (DW elements based)

Action:
If operand is op8, signed AX = AL * op8
If operand is op16, signed DX::AX = AX * op16

Flags Affected: OF, SF=?, ZF=?, AF=?, PF=?, CF

Interrupts

- Interrupts are calls to system routines (usually BIOS and OS services).
- These routines are “resident” in memory.
- The memory positions where the routines start are stored in a memory table.
- This table is located at the beginning of memory in DOS: from address 0 to 3FFh.
- Every 4 bytes in this table constitute an interrupt vector
- (offset and segment where the associated interrupt service routine starts)

Print a single character on screen

INT 21H AH = 2H
Description: The function of this routine is to visualize a character.
Use: Input: AH = 2H
DL contains ASCII character to print.
Output: None

Affected registers: None

Example:

```
mov ah, 2 ; Function Number = 2
mov dl, 'A' ; Character to print in ASCII
int 21h ; Executes OS routine
```

Int 21h AH=9: Print ASCII string in screen

Description: Print in the screen a ASCII string that finishes with character ASCII=$.
Input Parameters: AH = 9.
DX = Offset in memory from the base-address of the data segment where the first character of the string is stored in memory.

Output Parameters: None.
Modified Registers: None.
Example:

```
.DATA
TextStr DB "Hello world",13,10,'$' ; string ending with characters CR,LF y'$'
. CODE
.....
; If DS is the segment where the string to print is located:
mov dx, offset TextStr ;    DX : offset to first position of the string text to print
mov ah, 9 ; Function number = 9 (print string)
int 21h ; Software interruption to the operating system
```

Directives for Segment definition and prodecures

- PUBLIC: Indicate to linker that a label (variable or procedure) declared in the file may be referenced from other files (public).
- EXTRN: Indicate to linker that a label is declared in another file(extern)

```
; ============================================================
;                                   fopen
; Input:
;      DS:DX = Address of ASCIIZ filename
; Output:
;      CF = Set on error
;      AX = File handle (CF=0) / Error code (CF=1)
; ============================================================
```

```
extrn fopen:FAR
name db “file.WAV",0
……
mov dx, OFFSET name
call fopen
jc open_error
Mov bx, ax   ; bx = File handle
```

ASCIIZ : ASCII String finished by null character (Hex 0).