DOS FUNCTIONS AND INTERRUPTS  
(KEYBOARD AND VIDEO PROCESSING)

The Intel CPU recognizes two types of interrupts namely hardware interrupt when a peripheral devices needs attention from the CPU and software interrupt that is call to a subroutine located in the operating system. The common software interrupts used here are INT 10H for video services and INT 21H for DOS services.

**INT 21H:**
It is called the DOS function call for keyboard operations follow the function number. The service functions are listed below:

- **# 00H- It terminates the current program.**
  - Generally not used, function 4CH is used instead.
- **# 01H- Read a character with echo**
  - Wait for a character if buffer is empty
  - Character read is returned in AL in ASCII value
- **# 02H- Display single character**
  - Sends the characters in DL to display
  - MOV AH, 02H
  - MOV DL, ‘A’ ; move DL, 65
  - INT 21H
- **# 03H and 04H – Auxiliary input/output**
  - INT 14H is preferred.
- **# 05H – Printer service**
  - Sends the character in DL to printer
- **# 06H- Direct keyboard and display**
  - Displays the character in DL.
- **# 07H- waits for a character from standard input**
Microprocessors

lecture 6: Programming with 8086 Microprocessor

- does not echo

# 08H - keyboard input without echo
- Same as function 01H but not echoed.

# 09H - string display
- Displays string until ‘$’ is reached.
- DX should have the address of the string to be displayed.

# 0AH – Read string

# 0BH - Check keyboard status
- Returns FF in AL if input character is available in keyboard buffer.
- Returns 00 if not.

# 0CH - Clear keyboard buffer and invoke input functions such as 01, 06, 07, 08 or 0A.
- AL will contain the input function.

INT 21H Detailed for Useful Functions

# 01H
MOV, AH 01H; request keyboard input INT 21H
- Returns character in AL. IF AL= nonzero value, operation echoes on the screen. If AL= zero means that user has pressed an extended function key such as F1 OR home.

# 02H
MOV AH, 02H; request display character
MOV DL, CHAR; character to display
INT 21H

- Display character in D2 at current cursor position. The tab, carriage return and line feed characters act normally and the operation automatically advances the cursor.

# 09H
MOV Ah, 09H; request display
LEA DX, CUST_MSG; local address of prompt
INNT 21H
CUST_MSG DB “Hello world”, ‘$’
- Displays string in the data area, immediately followed by a dollar sign ($ or 24H), which uses to end the display.

# OAH
MOV AH, 0AH ; request keyboard input
LEA DX, PARA_LIST ; load address of parameter list
INT 21H

Parameter list for keyboard input area:
PARA_LIST LABEL BYTE; start of parameter list
MAX_LEN DB 20; max. no. of input character
ACT_LEN DB ? ; actual no of input characters
KB-DATA DB 20 DUP (''); characters entered from keyboard

- Label directive tells the assembler to align on a byte boundary and gives location the name PARA_LIST.
- PARA_LIST & MAX_LEN refer same memory location, MAX_LEN defines the maximum no of defined characters.
- ACT_LEN provides a space for the operation to insert the actual no of characters entered.
- KB_DATA reserves spaces (here 20) for the characters.

Example:

TITLE to display a string
.MODELSMALL
.STACK 64
.DATA
STRUDB ‘programming is fun’, ‘$’
.CODE
MAIN PROC FAR
MOV AX, @DATA
MOV DS, AX
MOV AH, 09H ;display string LEA
DX, STR
INT 21H
MOV AX, 4C00H
INT 21H
MAIN ENDP
END MAIN

INT 10H
It is called video display control. It controls the screen format, color, text style, making windows, scrolling etc. The control functions are:

# 00H – set video mode
MOV AH, 00H ; set mode
MOV AL, 03H ; standard color text
INT 10H ; call interrupt service

# 01H- set cursor size
MOV AH, 01H
MOV CH, 00H ; Start scan line
MOV CL, 14H ; End scan line
INT 10H ; (Default size 13:14)
# 02H – Set cursor position:
```
MOV AH, 02H
MOV BH, 00H ; page no
MOV DH, 12H ; row/y (12)
MOV DL, 30H ; column/x (30)
INT 10H
```

# 03H – return cursor status
```
MOV AH, 03H
MOV BH, 00H; INT 10H
Returns: CH- starting scan line, CL-end scan line, DH- row, DL-column
```

# 04H- light pen function

# 05H- select active page
```
MOV AH, 05H
MOV AL, page-no. ; page number
INT 10H
```

# 06H- scroll up screen
```
MOV AX, 060FH ; request scroll up one line (text)
MOV BH, 61H ; brown background, blue foreground
MOV CX, 0000H ; from 00:00 through
MOV DX, 184FH ; to 24:79 (full screen)
INT 10H
AL= number of rows (00 for full screen)
BH= Attribute or pixel value
CX= starting row: column
DX= ending row: column
```

# 07H-Scroll down screen
Same as 06H except for down scroll

# 08H (Read character and Attribute at cursor)
```
MOV AH, 08H
MOV BH, 00H ; page number 0(normal)
INT 10H
AL= character
BH= Attribute
```

# 09H -display character and attribute at cursor
```
MOV AH, 09H
MOV AL, 01H ; ASCII for happy face display
```
MOV BH, 00H  ; page number
MOV BL, 16H   ; Blue background, brown foreground
MOV CX, 60    ; No of repeated character
INT 10H

# 0AH-display character at cursor

MOV AH, 0AH
MOV AL, Char MOV BH, page _no MOV BL, value MOV CX, repetition INT 10H

# 0BH- Set color palette

✓ Sets the color palette in graphics mode
✓ Value in BH (00 or 01) determines purpose of BL
✓ BH= 00H, select background color, BL contains 00 to 0FH (16 colors)
✓ BH = 01H , select palette, Bl, contains palette MOV AH, 0BH

MOV AH, 0BH
MOV BH, 00H; background MOV BH, 01H ; select palette
MOV BL, 04H; red MOV BL, 00H ; black
INT 21H
INT 21H

#0CH- write pixel Dot

- Display a selected color
  AL=color of the pixel  CX= column
  BH=page number  DX= row

MOV AH, 0CH
MOV AL, 03
MOV BH,0
MOV CX, 200
MOV DX, 50
INT 10H
It sets pixel at column 200, row 50

#0DH- Read pixel dot

- Reads a dot to determine its color value which returns in AL
MOV AH, 0DH
MOV BH, 0 ; page no
MOV CX, 80 ; column
MOV DX, 110 ; row
INT 10H
#OEH- Display in teletype mode
- Use the monitor as a terminal for simple display
  MOV AH, 0EH
  MOV AL, char
  MOV BL, color; foreground
color INT 10H

#OF H- Get current video mode
Returns values from the BIOS video.
AL= current video mode MOV AH, 0FH
AH= no of screen columns INT 10H
BH = active video page

TITLE To Convert letters into lower case
.MODEL SMALL
.STACK 99H
.CODE
MAIN PROC
  MOV AX, @ DATA
  MOV DS, AX
  MOV SI, OFFSER STR
M:
  MOV DL, [SI]
  MOV CL, DL
  CMP DL, ‘$’
  JE N
  CMP DL, 60H
  JL L
  MOV DL, CL
  ADD DL, 20H
  MOV AH, 02H
  INT 21H
  INC SI
  JMP M

K:
  MOV DL, CL
  MOV AH, 02H
  INT 21H
  INC SI
  JMP M

L:
  MOV DL, CL
  ADD DL, 20H
  MOV AH, 02H
  INT 21H
  INC SI
  JMP M

N:
  MOV AX, 4C00H
  INT 21H
  MAIN ENDP
.DATA
STR DB ‘I am MR Rahul ‘, ‘$’
TITLE to reverse the string

.MODEL SMALL
.STACK 100H
.DATA
  STR1 DB “ My name is Rahul”, ‘$’
  STR2 db 50 dup (‘$’)
.CODE
MAIN PROC FAR
  MOV BL,00H
  MOV AX, @ DATA
  MOV DS, AX
  MOV SI, OFFSET STR1
  MOV DI, OFFSET STR2
L2:
  MOV DL, [SI]
  CMP DL, ‘$’
  JE L1
  INC SI
  INC BL
  JMP L2
L1:
  MOV CL, BL
  MOV CH, 00H
  DEC SI
L3:
  MOV AL, [SI]
  MOV [DI], AL
  DEC SI
  INC DI
  LOOP L3
  MOV AH,09H
  MOV DX, OFFSET STR2
  INT 21H
  MOV AX, 4C00H
  INT 21H
MAIN ENDP
END MAIN

TITLE to input characters until ‘q’ and display

.MODEL SMALL
.STACK 100H
.DATA
  STR db 50 DUP (‘$’)
.CODE
MAIN PROC FAR
MOV AX, @DATA
MOV DS, AX
MOV SI, OFFSET STR
L2: MOV AH, 01H
    INT 21H
    CMP AL, 'q'
    JE L1
    MOV [SI], AL
    INC SI
    JMP L2
L1:  MOV AH, 09H
    MOV DX, OFFSET STR
    INT 21H
    MOV AX, 4C00H
    INT 21H
MAIN ENDP
END MAIN