INT-21H The General function Dispatcher

- 19PW13
Madhumitha.S

Most of the General functions are services offered by DOS are implemented through this interrupt. The functions available are well standardised and should be common to all MSDOS, PCDOS and DOS plus systems.

Well behaved programs, therefore should use these facilities in preference to any other methods available for the widest range of compatibility.

INT 21h in the 5.12's implementation of DOS plus 2.2 provides 77 official functions and return with no action. Within this range some class have subfunctions which further extend range of applications.

In all calls, on entry AH defines the function. Other parameters may also be required in other registers. In all cases, general programming technique is to set AH to the function pointer, setup the required register contents, then to issue the call by assembly code INT instruction

The General syntax for calling function is:

```
mov AH, function number
    ; input parameter
int 21H
    return value.
```

- **INT 4ch: Terminate process**

→ Ends the current program, returns an optional 8 bit return code to the calling process.

→ A return code of 0 usually indicates successful compilation

```
MOV AH, 4CH    ; terminate process
MOV AL, 0      ; return code
INT 21H
; same as ;
EXIT 0.
```

- **ASCII control characters:**

→ Many INT 21H functions act upon the following control characters.

* 08H - Backspace (moves 1 column left)
* 09H - Horizontal tab (skips forward n columns)
* 0AH - Lines feed (moves to next o/p line)
* 0CH - carriage return
* 1BH - Escape character.

SELECTED O/P FUNCTIONS

* 02H, 06H - write single character to standard O/P
* 05H - write single character to default printer
* 09H - write string (terminate with $) to standard O/P
* 40H - write block of data to a file or device

example: INT 21H function 05H : write character to default printer

```
// write the letter 'A'
    mov AH, 05H ; select printer o/p
    mov DL, 65  ; character to be printed
    INT 21H     ; call MS-DOS
```

SELECTED I/P FUNCTIONS:

* 07H, 08H - Read a single character from standard i/p
* 0AH - Read array of buffer characters from standard i/p
* 0BH - Get status of standard i/p buffer
* 3FH - Read from file/device.

example: INT 21H function 0AH : Read buffered array from standard i/p.

→ Requires a predefined structure to be setup that describes maximum i/p size and holds i/p characters

EXAMPLE :

```
count = 80
KEYBOARD STRUCT
    maxInput BYTE count ; max chars to i/p
    input count BYTE ?   ; actual i/p count
    buffer BYTE count DUP(?) ; holds i/p
chars
KEYBOARD ENDS
```

- **INT 21H function 0AH**

→ Executing the interrupt.

```
.data
  KybdData KEYBOARD< >
.code.
    MOV AH, 0AH
    MOV DX, OFFSET KybdData
    INT 21H
```

- **INT 21H function 3FH :**

Reads data from file/device. It reads block of bytes, can be interrupted by ctrl-Break (^c)

EXAMPLE – Read string from keyboard

- data

inputBuffer BYTE 127 dup(0)
bytesRead WORD ?

- code

```
MOV   AH,3FH
MOV   BX,0 ;      keyboard handle
MOV   CX,127 ;    max bytes to read
MOV   DX,OFFSET inputBuffer ; target location
INT   21H
MOV   bytesRead,AX ; save character count
```

- **DATE|TIME FUNCTIONS:**

* 2AH - Get system Date
* 2BH - Set system Date
* 2CH - get system time
* 2DH - set system Time.

EXAMPLE - displaying Date and Time

1. Get System Date :

```
//Returns year in CX, month in DH, day in DL, day of week in AL
MOV   AH,2AH
INT   21H
MOV   Year,CX
MOV   Month,DH
MOV   Day,DL
MOV   dayofWeek,AL
```

2. Set System time :

// sets system date

// AL = 0 if function was not successful in modifying the time.

```
MOV AH, 2DH
MOV CH, hours
MOV CL, minutes
MOV DH, seconds
INT 21H
CMP AL, 0
JNE failed.
```