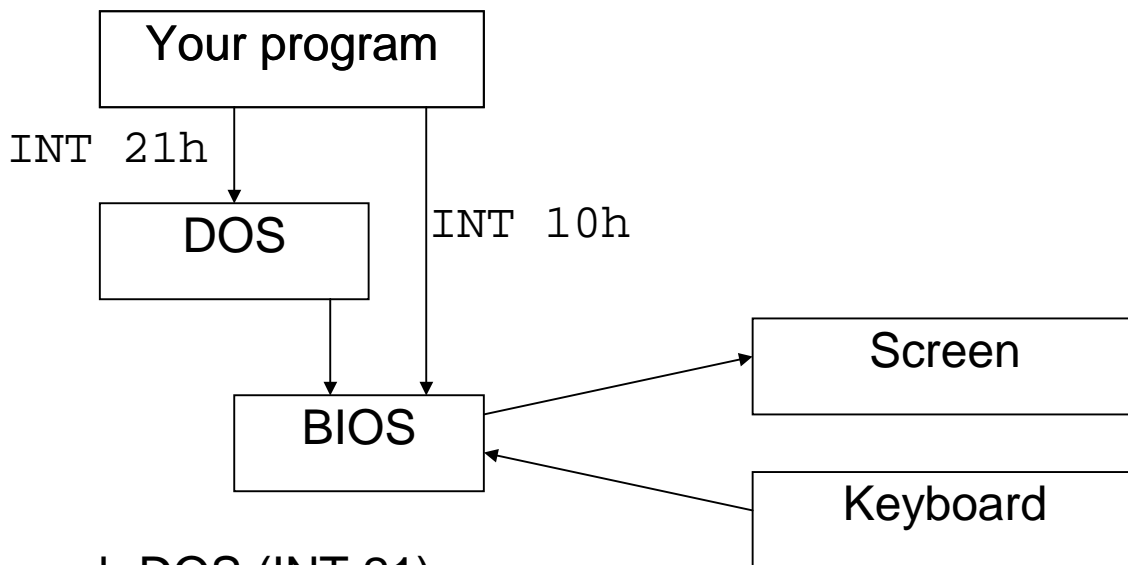


Lecture 11

Screen and Keyboard
Processing

Text: (4th edition) Chapter 9
(5th edition) Chapter 8

Two ways of controlling the screen and keyboard:



Through DOS (INT 21)

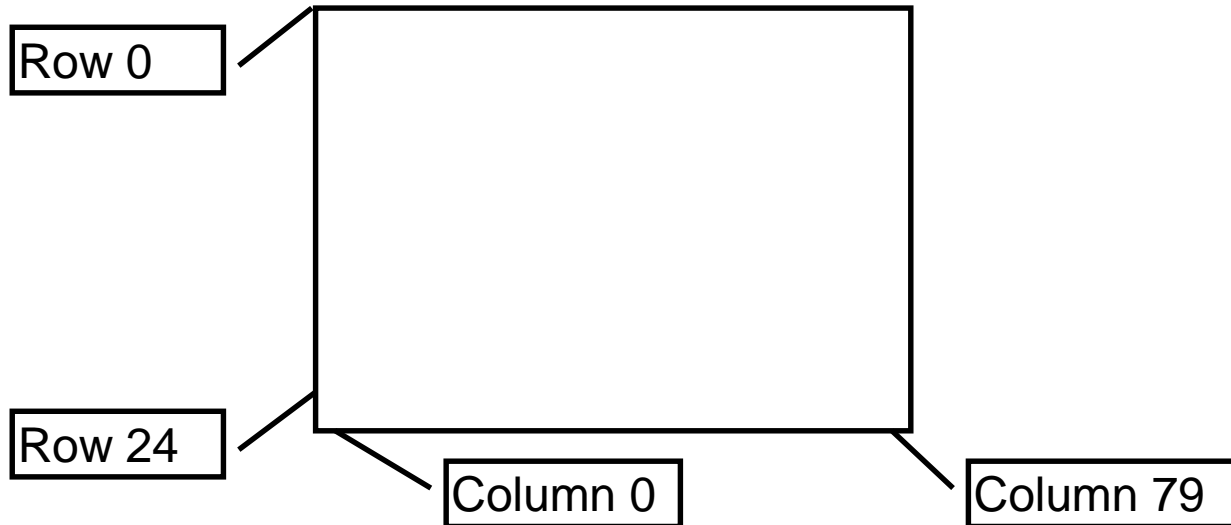
Through the BIOS (INT 10)

BIOS INT 10		DOS INT 21	
02h	Set cursor	02h	Screen display
06h	Scroll screen	09h	Screen display
		0Ah	Keyboard input
		3Fh	Keyboard input
		40h	Screen display

BIOS and DOS function values

The Screen

Grid of locations 25 rows by 80 columns.



Example: Moving the Cursor

```
MOV    AH,02h    ;code to set cursor
MOV    BH,00     ;page number 0
MOV    DH,05     ;Row 5
MOV    DL,12     ;Column 12
INT    10h       ;BIOS call
```

Example: Clearing the Screen

```
MOV    AX,0600H  ;06 scroll, 00 full
MOV    BH,71H    ;white(7) on blue(1)
MOV    CX,0000H  ;row 00: column 00
MOV    DX,184FH  ;row 24: column 79
INT    10h       ;BIOS call
```

Printing a STRING to the screen...

Screen Display with DOS function 09h, INT 21h

the DX register is loaded with the address of the first byte of the string.

```
Message      DB    'This is CS 141$'
;
;  move the cursor to the row/column
;  where the string should begin
;
      MOV     AH,02h      ;code to set cursor
      MOV     BH,00      ;page number 0
      MOV     DH,12      ;Row 12
      MOV     DL,33      ;Column 33
      INT     10h        ;BIOS call
;
;  now print the string to the screen
;
      MOV     AH,09H     ;code for display
      LEA    DX,Message ;addr of text
      INT     21h        ;DOS call
```

The end of the message is marked by the '\$' which is not printed.

```

page 60,132
TITLE      P09DOSAS (COM) Disp ASCII chars 00H-FFH
           .MODEL SMALL
           .CODE
           ORG      100H
BEGIN:     JMP      SHORT MAIN
CHAR       DB      00, '$'

;         Main procedure:
;         -----
MAIN       PROC     NEAR
           CALL     B10CLR      ;Clear screen
           CALL     C10SET     ;Set cursor
           CALL     D10DISP    ;Display characters
           MOV      AX,4C00H   ;Exit to DOS
           INT      21H
MAIN       ENDP
;         Clear screen:
;         -----
B10CLR     PROC     NEAR
           MOV      AX,0600H   ;Scroll full screen
           MOV      BH,07      ;Attrib: wh on blk
           MOV      CX,0000    ;Upper left loc
           MOV      DX,184FH   ;Lower right loc
           INT      10H
           RET
B10CLR     ENDP
;         Set cursor to 00,00:
;         -----
C10SET     PROC     NEAR
           MOV      AH,02H     ;Request set cursor
           MOV      BH,00      ;Page number 0
           MOV      DX,0000    ;Row 0, column 0
           INT      10H
           RET
C10SET     ENDP

```

```
;          Display ASCII characters:
;          -----
D10DISP   PROC
          MOV      CX,256      ;Do 256 iterations
          LEA     DX,CHAR      ;Init addr of char
D20:
          MOV     AH,09H      ;Display ASCII char
          INT     21H
          INC     CHAR        ;Incr for next char
          LOOP    D20         ;Do 256 times
          RET
D10DISP   ENDP
          END      BEGIN
```

To assemble, link and run this program:

```
C>tasm p09dosas
```

```
C>tlink /Tdc p09dosas
```

*the /Tdc option is necessary because this is a DOS
.COM program.*

```
C>p09dosas
```

Keyboard Input

DOS INT 21h, function 0Ah

Requires a *parameter list* - a list of bytes that provide the following information in exactly this order:

- Maximum number of input characters, including “ENTER” (DOS will accept fewer, but no more)
- Actual number of characters that were typed
- A sequence of bytes which are the characters that were entered.

Example:

Suppose we want the user to type a 5-digit zip code:

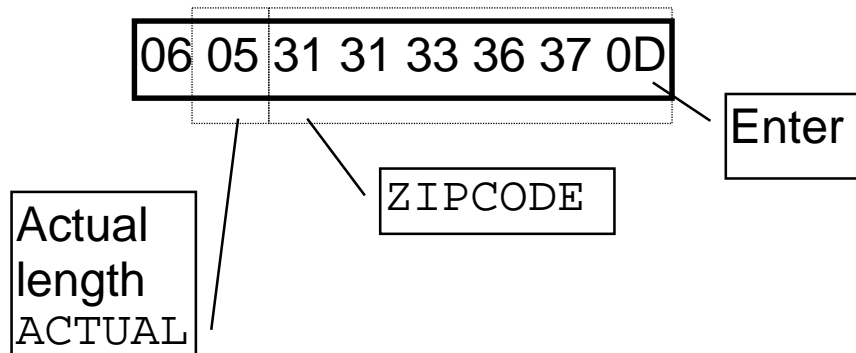
```
INLIST LABEL BYTE ;a place holder
MAXCHAR DB 6 ;zip code plus CR
ACTUAL DB ? ;get back # typed
ZIPCODE DB 6 DUP(?)
```

The code to do the input would be:

```
MOV AH,0Ah ;keyboard input
LEA DX,INLIST ;give p-list
INT 21H ;DOS call
```

Note that MAXCHAR is a value given to DOS by the programmer, and ACTUAL and ZIPCODE are filled in by DOS for the programmer to use once the input is completed.

If the user typed 11367, after the DOS call, INLIST would appear as



If you wanted to print the zip code to the screen:

```
MOV  ZIPCODE+5, '$'    ;replace CR with $
MOV  AH,09h           ;set up DOS call
LEA  DX,ZIPCODE       ;print zip code
INT  21H              ;DOS Call
```

Or better yet,

```
MOV  AH,02H          ;display sgl.char.
MOV  CX,7            ;set up looping
MOV  WORD PTR ZIPCODE+5,0D0Ah
LEA  DI,ZIPCODE
AGAIN:
MOV  DL,[DI]        ;get next char
INT  21H            ;print it
INC  DI              ;point to next char
LOOP AGAIN          ;keep going
```


Exercises - Lecture 11

Write appropriate DATA and CODE segments for a program which will

1. prompt the user for a three character major code.
2. Print one of three possible messages after clearing the screen:
if the code is print
026 Computer Science BA
027 Computer Science BS
other Not a computer science major