**Lab 11: BIOS Level Programming**

**OBJECTIVE:** Learn about Keyboard and Mouse control at BIOS level.

**Objective:**

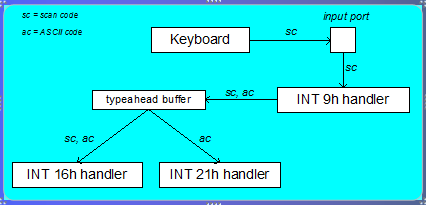
* BIOS interrupts
* Keyboard Input with INT 16h
* Scan code and ASCII code

**PC-BIOS:**

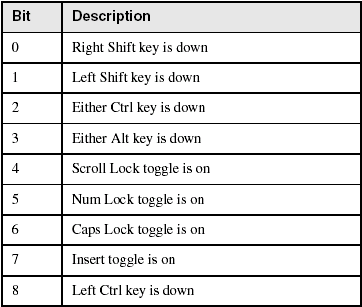
* The BIOS (Basic Input-Output System) provides low-level hardware drivers for the operating system.
  + accessible to 16-bit applications
  + written in assembly language, of course
  + source code published by IBM in early 1980's
* Advantages over MS-DOS:
  + permits graphics and color programming
  + faster I/O speeds
  + read mouse, serial port, parallel port
  + low-level disk access

**Keyboard Input with INT 16h:**

* How the Keyboard Works
* INT 16h Functions
* Keystroke sends a scan code to the keyboard serial input port
* Interrupt triggered: INT 9h service routine executes
* Scan code and ASCII code inserted into keyboard typeahead buffer



**Keyboard Flags:**



**INT 16h Functions:**

* Provide low-level access to the keyboard, more so than MS-DOS.
* Input-output cannot be redirected at the command prompt.
* Function number is always in the AH register
* Important functions:
  + set typematic rate
  + push key into buffer
  + wait for key
  + check keyboard buffer
  + get keyboard flags

**Function 10h: Wait for Key:**

If a key is waiting in the buffer, the function returns it immediately. If no key is waiting, the program pauses (blocks), waiting for user input.

* .data
* scanCode BYTE ?
* ASCIICode BYTE ?
* .code
* mov ah,10h
* int 16h
* mov scanCode,ah
* mov ASCIICode,al

**Activities:**

* Write a key logger using BIOS level interrupts, which will keep the record of all the keys pressed on keyboard.
* Note: Use Scan and ASCII codes for the activity.

**Lab 11: Mouse Handling**

**OBJECTIVE:** Mouse handling in Assembly.

**Objectives:**

* MS-DOS functions for reading the mouse
* INT 33h functions
* Mouse Tracking Program Example

**Reset Mouse and Get Status:**

* INT 33h, AX = 0
* Example:
  + mov ax,0
  + int 33h
  + cmp ax,0
  + je MouseNotAvailable
  + mov numberOfButtons,bx

**Show/Hide Mouse:**

* INT 33h, AX = 1 (show), AX = 2 (hide)
* Example:
  + mov ax,1 ; show
  + int 33h
  + mov ax,2 ; hide
  + int 33h

**Get Mouse Position & Status:**

* INT 33h, AX = 4
* Example:
  + mov ax,4
  + mov cx,200 ; X-position
  + mov dx,100 ; Y-position
  + int 33h

**Get Button Press Information:**

* INT 33h, AX = 5
* Example:
  + mov ax,5
  + mov bx,0 ; button ID
  + int 33h
  + test ax,1 ; left button down?
  + jz skip ; no - skip
  + mov X\_coord,cx ; yes: save coordinates
  + mov Y\_coord,dx

**Other Mouse Functions:**

* AX = 6: Get Button Release Information
* AX = 7: Set Horizontal Limits
* AX = 8: Set Vertical Limits

**Activity:**

* Get the mouse button press information, whether the mouse click was left, right or middle.