**פרוט של פסיקה 16**

**איליה זלדנר**

**Int 16H**

**Read char from buffer - wait if empty**

**Ah = 0**

**Return: ah = scan code , al = character**

**Check buffer - do not clear**

**Ah = 1**

**Return: flag zf = 0 = char in buffer ,ah = scan code ,**

**Al = character , flag zf = 1 = no char in buffer**

**Get shift status**

**Ah = 2**

**Al = shift status bits :**

**0 = right shift key depressed**

**1 = left shift key depressed**

**2 = ctrl depressed**

**3 = alt depressed**

**4 = scroll lock active**

**5 = num lock active**

**6 = caps lock active**

**7 = insert state active**

**Set delays (pcjr, at, ps/2)**

**Ah = 3**

**Al = 0: reset typematic (pcjr)**

**Al = 1: increase initial delay (pcjr)**

**Al = 2: increase continuing delay (pcjr)**

**Al = 3: increase both delays (pcjr)**

**Al = 4: turn off typematic (pcjr)**

**Al = 5: set typematic rate (at or ps/2)**

**Bh = 00 - 03 for delays of 250ms, 500ms, 750ms, or 1s**

**Bl = 00 - 1f for typematic rates of 30cps down to 2cps**

**Keyclick (pcjr only)**

**Ah = 4**

**Al = 0: click off**

**Al = 1: click on**

**Write to keyboard buffer (at or ps/2 with enhanced kbd)**

**Ah = 5**

**Ch = scan code**

**Cl = character**

**Get enhanced keystroke (at or ps/2 with enhanced kbd)**

**Ah = 10h**

**Return: ah = scan code**

**Al = character**

**Check enhanced keystroke (at or ps/2 with enhanced kbd)**

**Ah = 11h**

**Return: zf = 0 if keystroke available**

**Ah = scan code \ meaningless if zf = 1**

**al = character /**

**zf = 1 if kbd buffer empty**

**Get enhanced shift flags (at or ps/2 with enhanced kbd)**

**Ah = 12h**

**Return: al (same as for ah=02)**

**Bit 7: ins on**

**Bit 6: capslock on**

**Bit 5: numlock on**

**Bit 4: scrolllock on**

**Bit 3: either alt key down**

**Bit 2: either ctrl key down**

**Bit 1: left shift key down**

**Bit 0: right shift key down**

**Ah :**

**Bit 7: sysreq key down**

**Bit 6: capslock key down**

**Bit 5: numlock key down**

**Bit 4: scrolllock key down**

**Bit 3: right alt key down**

**Bit 2: right ctrl key down**

**Bit 1: left alt key down**

**Bit 0: right alt key down**