

## מעבדה 2

התקני קלט – פלט

# מנהלי התקן-Device drivers

- ▶ אלו תוכניות קלט / פלט שמשתמשות בפקודות מכונה מיוחדות
- וכתובות מיוחדות על מנת לתקשר דרך לוח האם עם התקנים חיצוניים .
- ▶ דיסקים , כרטיסי רשת , כרטיסי מסך , התקני USB וכו'..
- ▶ מי שמספק את התוכנות הללו הם בדרך כלל היצרנים של התוכנות הללו .
- ▶ התוכנות הללו חיוניות לתקשורת עם ההתקנים.
- ▶ לעיתים מנהל ההתקן ממומש כולו או חלקו כרוטינות טיפול בפסיקות חומרה/תוכנה

# מנהלי התקן-Device drivers

- ▶ מי אחראי שתוכנת ה Driver תותקן במחשב?
- ▶ לחלק מההתקנים אחראי יצרן המחשב (ASUS, HP..)
- ▶ לעיתים האחריות היא על מערכת הפעלה (כרטיסי רשת, מסך,
- ▶ הוספת הציוד ההיקפי נעשה על ידי המשתמש (מצלמת אינטרנט)
- ▶ כיום לעיתים קרובות הדרייבר מותקן אוטומטית כמעט תמיד עם החיבור למחשב המסופק על ידי היצרן המיועד ספציפית למערכת הפעלה מסויימת.

# שיטות קלט/פלט

▶ כיצד נראה הקוד של מנהל ההתקן מבפנים??

▶ תוכנות אלו מאוד תלויות בחומרה .

▶ רובן כתובות בשפת אסמבלי או בפקודות מיוחדות בשפה העילית.

▶ מבחינת חברת אינטל, קיימות שתי שיטות קלט/פלט

▶ :

▶ קלט/פלט דרך ports

▶ קלט/פלט ממופה זיכרון

▶

# קלט-פלט דרך PORTS

- ▶ PORT ים הם בעצם מרחב כתובות מיוחד ל-ק/פ
- ▶ מרחב הכתובות הוא בין 0-65535
- ▶ למה שמשמש PORT מסוים, נקבע ברמת החומרה.
- ▶ כל PORT הוא בית אחד. (8 סיביות)
- ▶ PORT הוא מעין חוט שקצה אחד הוא ה CPU וקצה השני הוא גורם חיצוני, בדרך כלל צ'פ על לוח האם.

# קלט-פלט דרך PORTS

- ▶ המתכנת צריך לקבל את מספרי ה-port הרלוונטיים, מה משמעותם ומה הפורמט של המידע שהם מספקים או מקבלים.
- ▶ הפקודות הבסיסיות ביותר של קלט/פלט דרך פורטים הם הפקודות IN, OUT.

# פקודות IN ו-OUT

- ▶ מבנה הפקודה IN :
- ▶ (למשל קריאת נתון מפורט מספר K לתוך AL)
- ▶ IN AL,k
- ▶ IN AX,k
- ▶ IN EAX,k
- ▶ IN AL,DX
- ▶ IN AX,DX
- ▶ IN EAX,DX
- ▶ k הוא מספר בין 0-255. אם מספר פורט מספרו גדול יותר ניגשים דרך DX.
- ▶ - IN, OUT עם אוגר EX, EAX משמעותו כתיבה ל port - ים עוקבים (כמו זיכרון)

# פקודות IN ו-OUT

▶ מבנה הפקודה OUT: (למשל כתיבת תוכן AL לפורט K)

- ▶ OUT k,AL
- ▶ OUT k,AX
- ▶ OUT k,EAX
- ▶ OUT DX,AL
- ▶ OUT DX,AX
- ▶ OUT DX,EAX



# פקודות IN ו-OUT

- ▶ כאשר האופרנד המידע הוא יותר מבית אחד (AX,EAX) הפעולה נעשית על port-ים עוקבים.
- ▶ לדוגמא, אם עושים
  - ▶ OUT 100h,AX
  - ▶ אזי תוכן AL ייכתב ל-port 100h ותוכן AH ייכתב ל-port 101h.
  - ▶ לדוגמא הפקודה
- ▶ IN AL,60h
- ▶ קלטה קוד של מקש נלחץ מהמקלדת.

# פקודות IN ו- OUT

▶ הפקודות:

- ▶ MOV AL,20h
- ▶ OUT 20h,AL

▶ משמשים לדיווח על סיום טיפול בפסיקות חומרה ללוח האם ( הודעה לבקר הפסיקות).

▶ הפקודות:

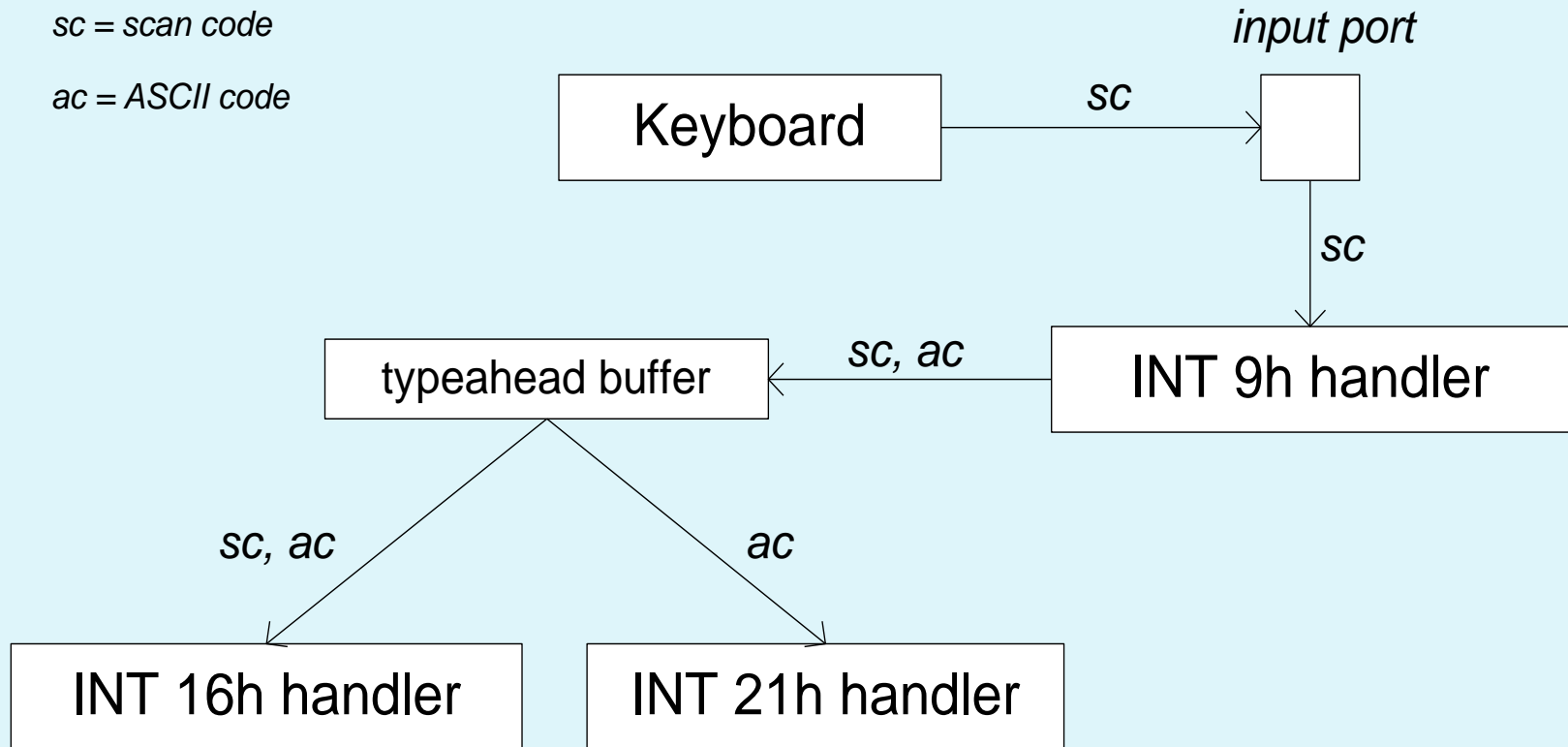
- ▶ MOV DX,3D4h
- ▶ OUT DX,AX

▶ מהווים כתיבה ל-port-ים 3D4h, 3D5h שמנחים את כרטיס המסך .

# בקר המקלדת

*sc = scan code*

*ac = ASCII code*



# בקר המקלדת למשל זה פורט 60H

- ▶ פורט 60H מחזיק את ה SCAN CODE שבקר ההתקנים קיבל מהמקלדת
- ▶ ערך זה הוא 7 ביט (בין 0 ל 127) ה MSB (הביט הגבוהה) מבדיל בין לחיצה לשחרור
- ▶ 0=לחיצה , 1=שחרור
- ▶ פורט 61H משמש להודיע לבקר ההתקנים שהתו הנוכחי נקרא.

# בקר המקלדת

שימוש בפקודות IN ו-OUT בכדי לקרוא scancode של מקש שנלחץ דרך פורט 8255:

```
IN AL,60h    //Read scancode from port 8255 (scancode saved in AL)
```

```
/*Signal to device controller of successful input*/
```

```
IN AL, 61h
```

```
OR AL,80H    //(80h) = (10000000b)- turning MSB on
```

```
OUT 61h,AL
```

```
AND AL, 7Fh //(7Fh) = (01111111b) - turning MSB off
```

```
OUT 61h, AL
```

פקודת TEST מבצעת AND לוגי על שני האופרנדים מבלי לשנות אותם.  
כלומר, משפיעה אך ורק על אוגר הדגלים.

```
TEST AL,80h //Check if key was pressed (0) or released (1)
```

# קלט – פלט – ממופה זיכרון

- ▶ שיטת קלט/פלט שבה כתיבה /קריאה לכתובות זיכרון מסויימות משמעותה קלט/פלט.
- ▶ כתובות מסויימות של זיכרון מוקצות למטרות קלט /פלט
- ▶ הדבר נקבע ברמת החומרה.
- ▶ בקר הזיכרון הוא זה שמזהה שמדובר בכתובת מיוחדת מסוג זה.
- ▶ ברמת התוכנית, ברמת פקודות המכונה, הקלט/פלט בא לידי ביטוי כפקודות רגילות של כתיבה לזיכרון .

# קלט – פלט – ממופה זיכרון

- ▶ במחשבי X86 השימוש בקלט/פלט ממופה זיכרון הוא כנראה בעיקר לכתיבה לזיכרון של כרטיס מסך.
- ▶ הסיבה לכך שיעד(הזיכרון של כרטיס המסך) הוא זיכרון וזה נוח להתייחס אליו כך, וגם משום שמדובר בכמויות אדירות של מידע.
- ▶ כרטיס מסך עובד בשני מודים , מוד גרפי (פיקסלים) ומוד טקסטואלי.
- ▶ אנחנו נעבוד במוד טקסטואלי.

# קלט – פלט – ממופה זיכרון

## ▶ עבודה עם כרטיס המסך :

- ▶ כרטיס המסך מפרש את תוכן הזיכרון שלו לתצוגה על המסך.
- ▶ במוד טקסטואלי , המסך מחולק למשבצות, בכל משבצת מופיעה אות.
- ▶ התיאור בזיכרון מחולק ל שתי משבצות:
- ▶ בית שהוא קוד אסקי של האות
- ▶ בית מאפיינים המגדיר צבע אות וצבע הרקע

- ▶ הצבעים במסך הם שילוב של 3 צבעים בסיסיים R אדום, G ירוק , B כחול.

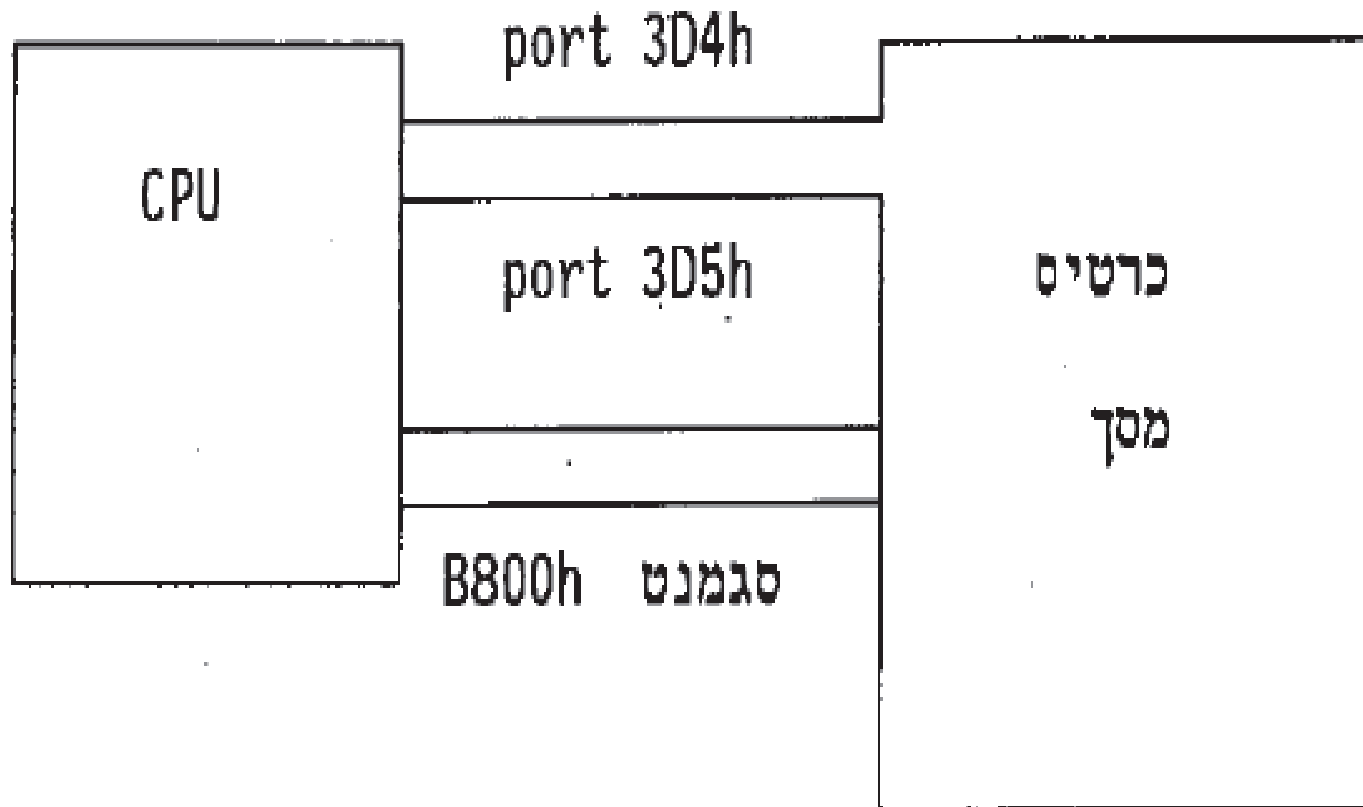
- ▶
- ▶ צבעי המסך מתקבלים משילוב של עוצמות של הצבעים הללו.
- ▶



# כרטיס מסך

- ▶ כרטיס המסך מחובר למעבד על ידי פורטים 3D4h ופורט 3D5h וכמו כן כרטיס המסך "נמצא" בסגמנט B800h .
- ▶ כלומר פניה לכתובת זיכרון בסגמנט זה משמעותה פניה לכרטיס המסך עצמו.
- ▶ כל משבצת במצב טקסט מיוצגת על ידי שני בתים, שהם בעצם כתובות בזיכרון בסגמנט B800h.
- ▶ הבית הראשון (הזוגי) מכיל את התו האסקי שיוצג על המסך באותה נקודה.
- ▶ הבית השני (האי-זוגי) מכיל את המאפיינים של התו (attributes) של התו (צבע, צבע רקע, , בהירות, הבהוב .

# כרטיס מסך



# כרטיס מסך

▶ בעזרת פורטים 3D4h ו- 3D5h אפשר לשלוט על גודל הסמן, מיקום הסמן, ועל הנקודה שתוצג בפינה השמאלית העליונה של המסך.

▶ פורט 3D4h הוא אוגר הפקודה-הוא קובע איך אפשר לפרש את הנתון שמועבר לפורט 3D5h.  
▶ בדרך כלל נעבוד כך :

▶ MOV DX, 3D4h

▶ OUT DX,AX

▶ כאשר AL מכיל את הערך לפורט 3D4h ו- AH את הערך לפורט 3D5h.

▶ לכל הגדרה כזאת יש להעביר ערך של 16 ביט (2 בתים)

▶ לכן יש צורך ב 2 כתובות לפורט. ( ע"י OUT )

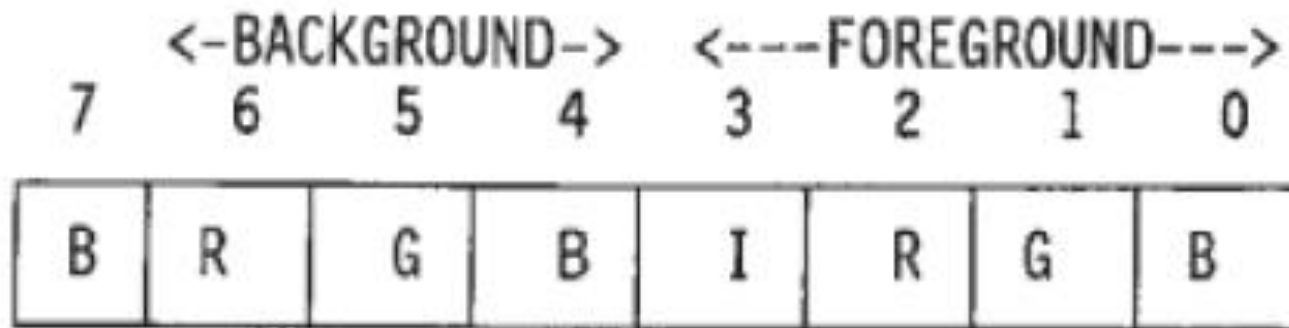
# כרטיס מסך

מוד המסך נקבע על ידי פסיקה 10h

## קביעת מוד מסך - דוגמא

MOV	AH,0	; Select function = 'Set mode'
MOV	AL,1	; 40 by 25 color image
INT	10h	; Adapter initialized. Page 0 displayed

# כרטיס מסך



- B - Blink ( 1 - Foreground character blinks )
- R - Red
- G - Green
- B - Blue
- I - Intensity ( 0 - Low, 1 - High)

# כרטיס מסך

קביעת גודל סמן:

0	גודל
15	סמן

תוכן 10 בפורט **3D4h**: פורט **3D5h** מפורש כגבול עליון לתצוגת הסמן (0 עד 14)

תוכן 11 בפורט **3D4h**: פורט **3D5h** מפורש כגבול תחתון לתצוגת הסמן (1 עד 15)

❖ (הסמן הרגיל הוא בעל גבול תחתון 15 וגבול עליון 14)

;Setting maximum cursor

MOV DX, 3D4h

MOV AX, 000Ah ; Border value = 0, Command = 10  
(upper border)

OUT DX, AX

MOV AX, 0F0Bh ; Border value = 15, Command = 11  
(lower border)

OUT DX, AX

;Setting minimum cursor

MOV DX, 3D4h

MOV AX, 0E0Ah ; Border value = 14, Command = 10  
(upper border)

OUT DX, AX

MOV AX, 0F0Bh ; Border value = 15, Command = 11  
(lower border)

OUT DX, AX

# כרטיס מסך

קביעת גודל\גבולות מסך:

תוכן 12 בפורט **3D4h**: פורט **3D5h** מפורש כבית עליון לנקודת ההתחלה של תצוגת המסך (היסט בבתים מתחילת סגמנט B800h).

תוכן 13 בפורט **3D4h**: פורט **3D5h** מפורש כבית תחתון לנקודת ההתחלה של תצוגת המסך (היסט בבתים מתחילת סגמנט B800h).

# כרטיס מסך

קביעת גודל\גבולות מסך – דוגמא

MOV DX,3D4h ; 3D4h - 3D5h: Display adapter ports

MOV AL,12 ; Display position high byte register

MOV AH,BH ; Move desired high byte value to AH

OUT DX,AX ; Output AL to port 3D4h, AH to port 3D5h

MOV DX,3D4h ; 3D4h - 3D5h: Display adapter ports

MOV AL,13 ; Display position low byte register

MOV AH,BL ; Move desired low byte value to AH

OUT DX,AX ; Output AL to port 3D4h, AH to port 3D5h



# כרטיס מסך

קביעת מיקום סמן:

תוכן 14 בפורט 3D4h: פורט 3D5h מפורש כבית עליון למיקום הסמן (היסט בבתים מתחילת סגמנט B800h).

תוכן 15 בפורט 3D4h: פורט 3D5h מפורש כבית תחתון למיקום הסמן (היסט בבתים מתחילת סגמנט B800h).

# כרטיס מסך

מיקום סמן - דוגמה

MOV DX,3D4H ; Point to 3D4H - 3D5H port pair

MOV AL,14 ; Address of cursor register pos high byte

MOV AH,BH ; Get desired value of cursor pos high byte

OUT DX,AX ; Port(3D4h) = 14, Port(3D5h) = Value of BH

MOV AL,15 ; Address of cursor register pos low byte

MOV AH,BL ; Get desired value of cursor pos low byte

OUT DX,AX ; Port(3D4h) = 15, Port(3D5h) = Value of BL