המשחק

# השעון

|  |
| --- |
| המטרה: מדידת זמן המשחק. נרצה לתת לשחקן שלנו לשחק רק דקה.  בתחילת המשחק נגדיר את זמן סיום המשחק לעוד 59 שניות. ובכל סיבוב של לולאת המשחק נבדוק אם הגיע זמן הסיום.  נכתוב תוכנית קטנה שמודדת 59 שניות ונשלב אותה במשחק |

## הפסיקות בתכנית שלנו

|  |  |
| --- | --- |
| פקודה להדפסת מחרוזת  (שם משתנה המחרוזת msg) | mov dx, offset msg  mov ah, 9h  int 21h |
| פקודה לקליטת תו מהמקלדת  לא מדפיסה את תו **ולא** ממשיכה בהוראות על לקבלת תו. | mov ah, 0h  int 16h  הפסיקה מחזירה בתוךal את קוד ה־ ASCII של התו שנמצא בראש הבאפר ובתוךah את ה־code scan שלו. בנוסף, הפסיקה "מנקה" את התו מהבאפר |
| פקודה לקליטת תו מהמקלדת  לא מדפיסה את תו, אם לא הוקש על מקש ממשיכה בפקודות התכנית. | mov ah, 1h  int 16h  הפסיקה מדליקה את ה zero flag– אם יש תו מוכן לקריאה (zf = 1) , ומכבה אם אין תו מוכן (zf = 0).  אם יש תו מוכן, al ו-ah יקבלו את ערכי ה־ASCII וה־code scan של התו. הפסיקה אינה "מנקה" את התו מהבאפר |
| כניסה לתצוגה גרפית  25\*40 | mov ax, 13h  int 10h |
| יציאה מתצוגה גרפית  80\*25 | mov ax, 2h  int 10h |
| מיקום הסמן על המסך | mov dh, [y\_cord] ; row  mov dl, [x\_cord] ; column  mov bh, 0 ; page number  mov ah, 2  int 10h |
| ציור תו על המסך במיקום הסמן | mov ah, 9  mov al, 2 ; al = character to display  mov bx, [color] ; bh = Background bl = Foreground  mov cx, 1 ; cx = number of times to write character  int 10h |
| השהייה  המתנה של פרק זמן קצר (חמישית השנייה) | mov cx, 03h ;High Word  mov dx, 4240h ;Low Word  mov al, 0  mov ah, 86h ;Wait  int 15h |
| קריאה של תו במיקום הסמן.  שמירת התו למשתנה  [screenChr] | mov bl, 0h ; Page=1  mov ah, 08h ; Read character function  int 10h ;return the character to al  mov [screenChr], al |
| קריאת השעון  הדקות יועתקו לתוך**cl**  מאיות השנייה יועתקו לתוך **dl**  השעות יועתקו לתוך **ch**  השניות יועתקו לתוך **dh** | mov ah, 2ch  int 21h |
| הדפסת תו היושב ב dl למסך | mov ah, 2h  int 21h |

|  |
| --- |
| IDEAL  MODEL small  STACK 100h  **p186**  DATASEG |

תרגול

פתחו את הקובץ base.asm ושימרו אותו בשם אחר. בראש הקובץ

הוסיפו p186

## קובץ clock

# הדפסת מספר דו סיפרתי

נרצה אפשרות לבדוק את מצב השעון. לכן, נכתוב תחילה פרוצדורה printNumber המדפיסה מספר בן 2 ספרות היושב במשתנה number ובעזרתה נוכל בהמשך להדפיס את מספר השניות בשעון

;print a number on the screen right corner

נגדיר 2 משתנים:

1. **number** - משתנה בגודל byte שיכיל את המספר הדו סיפרתי להדפסה
2. **ten** – משתנה שיכיל את המספר 10. בהמשך נחלק ב ten על מנת לקבל את
   * תוצאת החלוקה ב 10 – הספרה השמאלית
   * שארית החלוקה ב 10 – הספרה הימנית

proc printNumber

pusha

|  |  |  |
| --- | --- | --- |
| ; sets cursor on the corner  mov dh, 19 ; row  mov dl, 35 ; column  mov bh, 0 ; page number  mov ah, 2  int 10h |  | ראשית נשים את הסמן בפינה הימנית התחתונה על מנת שהמספר יודפס שם |
|  |  |  |
| mov al, [number]  mov ah, 0  div [ten] |  | נעביר את number ל al ונאפס את ah.  כעת [number] = ax לכן כשנחלק ב 10 [ten] התוצאה (הספרה השמאלית) תהיה ב al והשארית (הספרה הימנית) ב ah |
|  |  |  |
| add ax, '00' |  | נהפוך את 2 הספרות לתווי ascii על יד הוספת ‘0’ לכל אחת מהן |
|  |  |  |
| mov dx, ax  mov ah, 2h  int 21h |  | נעביר את ax ל dx לכן al עבר ל dl ו ah עבר ל dh  נקרא לפסיקה שמדפיסה את התו ששמור ב dl |
|  |  |  |
| mov dl, dh  mov ah, 2h  int 21h |  | נעביר את dh ל dl ושוב  נקרא לפסיקה שמדפיסה את התו ששמור ב dl |

popa

ret

endp printNumber

**בדקו את הפרוצדורה.**

* היכנסו למוד גראפי
* הכניסו ל number ערך כלשהו
* זמנו את הפעולה הריצו ובדקו הדפסה על המסך.
* הוסיפו המתנה למקש כדי שנוכל לראות את התוצאה
* חזרו למצגת תצוגה רגיל

## התכנית

IDEAL

MODEL small

STACK 100h

**p186**

DATASEG

; --------------------------

number db 0

ten db 10

; --------------------------

CODESEG

proc printNumber

;print a number on the screen right corner

pusha

; sets cursor on the corner

mov dh, 19 ; row

mov dl, 35 ; column

mov bh, 0 ; page number

mov ah, 2

int 10h

; div [number] / 10 --> result -> al, remainder -> ah

mov al, [number]

mov ah, 0

div [ten]

; turn digits to ascii

add ax, '00'

; mov ax to dx and print dl (da = al)

mov dx, ax

mov ah, 2h ; print dl

int 21h

; mov dh to dx and print dl

mov dl, dh

mov ah, 2h ; print dl

int 21h

popa

ret

endp printNumber

start:

mov ax, @data

mov ds, ax

; --------------------------

; Graphic mode 320 \* 200

mov ax, 13h

int 10h

mov [number], 21

call printNumber

; Wait for char

mov ah, 0h

int 16h

; return to text mode 80\*25

mov ax, 2h

int 10h

; --------------------------

exit:

mov ax, 4c00h

int 21h

END start

# חישוב זמן הסיום

השניות והדקות בשעון מתאפסים כל 60 שניות או דקות. אם נרצה להמתין 10 שניות   
והשעה כרגע היא 9:15:56 ⇦כעבור 10 שניות השעה תהייה ⇦ 9:16:05

השעון של השניות התאפס והתחלנו לספור את השניות מחדש לאחר שהגענו ל – 60 שניות.

לכן כאשר נרצה לבדוק אם עבר פרק זמן מסוים עלינו להתחשב בשינוי של הדקות או השניות בהתאם לזמן הנתון.

כדי למדוד 10 שניות נבדוק האם מצב השניות הוא נמוך מ – 50.

* אם נמוך מ – 50 נקבע את הזמן הסיום ⇦ לזמן הנקרא + 10.
* אם ערך השניות גבוהה מ- 50 עלנו לקבוע את זמן הסיום ⇦ לזמן הנקרא – 50.

נכתוב פרוצדורה, **calculateEndTime** שתקרא את השעון בתחילת המשחק ותוסיף לשניות הנוכחיות שלו 59. כך יסתיים המשחק לאחר 59 שניות

נשים לב:

1שניות ⇨ 59שניות + 2שניות

5שניות ⇨ 59שניות + 6שניות

37שניות ⇨ 59שניות + 38שניות

ורק

59שניות ⇨ 59שניות + 0שניות

1. נגדיר משתנה בשם endTime בגודל byte, שבו נחזיק את זמן הסיום

endTime db 0

תזכורת:

קריאת השעון

mov ah, 2ch

int 21h

הדקות יועתקו לתוך**cl**

מאיות השנייה יועתקו לתוך **dh**

השעות יועתקו לתוך **ch**

השניות יועתקו לתוך **dh**

1. נשתמש בפסיקה הקוראת את השעון.
2. נעתיק את השניות מ dh ל endTime ונעדכן אותו.
3. אם הוא 0 נוסיף 59 אחרת נחסיר 1

proc calculateEndTime

pusha

mov ah, 2ch ; read clock

int 21h

mov [endTime], dh ;move Seconds

cmp [endTime], 0 ; if 0 we need to add 59 to get to one mint.

je **add59**

dec [endTime]

jmp endClaculate

**add59:**

add [endTime], 59

endClaculate:

popa

ret

endp calculateEndTime

זמנו את הפעולה ובדקו ב – td מהו הערך של המשתנה endTime בסוף הפעולה.

הפקודות להרצה ב – TD

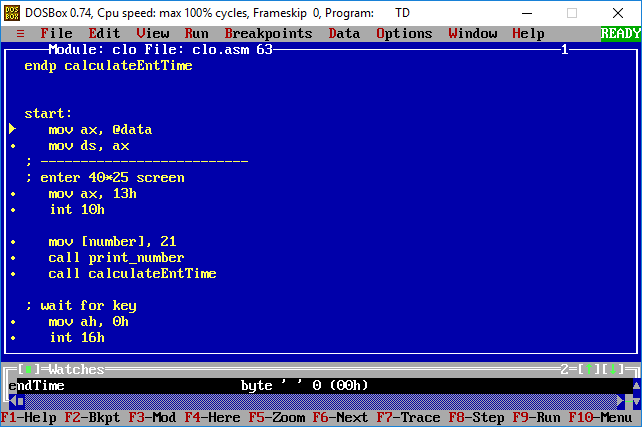
tasm /zi file\_name

tlink /v flie\_name

td tile\_name

ב – Turbo Debuger

לפני ההרצה בקשו לצפות במשתנה endTime



הריצו את התכנית בעזרת F8

שימו לב שלאחר ביצוע הפרוצדורה של calculateEndTime - המשתנה קיבל ערך חדש. (alt + x ⇦ יציאה מה – td).

# בדיקה: האם הגיעה זמן בסיום?

מחקו את הזימון לפרוצדורות calculateEndTime ו - printNumber בתכנית הראשית, ואת הפסיקה להמתנה למקש.

נכתוב פרוצדורה, **checkEndTime** שבודקת האם הגיעה זמן הסיום.

1. נגדיר, )כמו במשחק( משתנה בגודל byte בשם lost ונאתחל אותו ל 0

lost db 0

1. נקרא את השעון באמצעות הפסיקה
2. נשווה את dh ל endTime
3. במידה והם שווים נעביר ל lost 1
4. בכל מקרה נעביר את dh למשתנה number ונקרא ל prinNumber כדי להדפיס את ערך השניות בשעון על המסך

proc checkEndTime

pusha

mov ah, 2ch ; read clock

int 21h

cmp dh, [endTime] ; dl = second

jne **endCheckEndTime**

mov [lost], 1

**endCheckEndTime:**

mov [number], dh ; move second to print on screen

call printNumber

popa

ret

endp checkEndTime

# התוכנית

נשנה את התכנית הראשית שתחכה 59 שניות ותצא.   
בכל שניה יודפס ערך השניות של השעון לפינה הימנית התחתונה במסך

* לאחר הכניסה למוד הגראפי
* חשבו את זמן הסיום – קראו ל calculateEndTime
* כתבו את הלולאה הבאה:
* בדקו האם הגיע זמן הסיום – קראו ל checkEndTime
* בדקו את lost. אם ערכו 1, קפצו ל exit אחרת קפצו לתחילת הלולאה

start:

mov ax, @data

mov ds, ax

; --------------------------

; Graphic mode 320 \* 200

mov ax, 13h

int 10h

call calculateEndTime

gameLoop:

call checkEndTime

cmp [lost], 1

**je text\_mode**

jmp gameLoop

**text\_mode:**

; return to text mode 80\*25

mov ax, 2h

int 10h

; --------------------------

exit:

mov ax, 4c00h

int 21h

END start

# המשחק: יציאה אחרי דקה

## קובץ bg13

שלבו את התוכנית שכתבתם במשחק.

* העתיקו את המשתנים (ללא משתנה lost)
* העתיקו את הפרוצדורות
* לפני לולאת המשחק זמנו את **calculateEndTime**
* הכניסו ללולאה את checkEndTime.

## קוד התכנית הראשית

**call calculateEndTime**

mainGameLoop:

; delete chraracter => draw blank - chraracter at cursor position

mov [color], 0

call drawCharacter

**call checkEndTime**

; set cursor location and direction

call calculateCords

call setCursorePosition

הריצו ובדקו את התנית.

