# פעולות ומחסנית

## הפסיקות בתכנית שלנו

|  |  |
| --- | --- |
| פקודה להדפסת מחרוזת  (שם משתנה המחרוזת msg) | mov dx, offset msg  mov ah, 9h  int 21h |
| פקודה לקליטת תו מהמקלדת | mov ah, 0h  int 16h |
| 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 |

## פרוצדורות

בתכנית שלנו יש הרבה פסיקות שחוזרות על עצמן, בקוד התכנית.

נוכל לקצר את התכנית ולסדר אותה בשימוש בפרוצדורות.

העלו את הקובץ שיצרנו בשיעור שעבר (עם הלולאה שבה cx משמש מונה הלולאה).

שמרו את הקובץ בשם חדש והמשיכו לעבוד בקובץ החדש.

## יצירת פרוצדורות

את הפרוצדורות כותבים בתחילת מקטע הקוד – CODESEG

ולפני התווית של ה – start:

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

על מנת שנוכל להשתמש בפקודות pusha ו popa בפרויקט שלנו הוסיפו בתחילת הקובץ p186

1. צרו פרוצדורה המדפיסה למסך מחרוזת. בתוך מקטע הקוד לפני תווית ה – start:

; print a string on the screen assuming the string offset is in dx

**pusha** - הפקודה המשמרת את הערכים של הרגיסטרים לפני הכניסה לפעולה

proc printString

pusha

mov ah, 9h

פקודות הפסיקה להדפסת המחרוזת למסך

int 21h ;interrupt that displays a string

popa

**popa** - הפקודה המחזירה את הערכים לרגיסטרים לפני הכניסה לפעולה

ret

**ret** – מחזירה אותנו למיקום הנכון בתכנית.

endp printString

**בתכנית הראשית שנו את הפקודות,**

במקום הפסיקה המדפיסה למסך את המחרוזת, השאירו את ההוראה המעבירה לרגיסטר dx, את מיקום תחילת המחרוזת

במקום שורות הקוד שנמצאות בפרוצדורה זמנו את הפעולה:

; print openning screen

mov dx, offset openScreen ;printing the opening screen string

call printString

שמרו, הריצו ובדוק שהקוד תקין.

1. שנו את ההוראות הדפסת המחרוזת השנייה למסך (המחרוזת עם שמכם)

אל תשכחו לפני הזימון לפעולת ההדפסה להעביר לרגיסטר dx את מיקום תחילת המחרוזת בסגמנט הנתונים.

1. השתמשו בפרוצדורה להדפסת המסגרת של המשחק
2. צרו פרוצדורה הממתינה למקש ושומרת אותו למשתנה.

ראשית ניצור משתנה שישמור את הערך של המקש שהוקש בסגמנט הנתונים:

chr db 0

נוסיף את הפרוצדורה למקטע הקוד:

; reads a character into chr

proc readChr

pusha

; waits for character

mov ah, 0h

int 16h

mov [chr], al ; save character to [chr]

popa

ret

endp readChr

בתכנית הראשית מחקו את הקריאות לפסיקה של המתנה למקש

וזמנו את הפעולה במקום.

; waits for character

call readChr

1. הוסיפו תנאי הבודק האם הקשנו על מקש q (quit) כדי לצאת מהמשחק.

cmp [chr], 'q'

je end\_game

1. כדי לצייר את השחקן, ראשית ממקמים את הסמן ולאחר מכן מציירים את השחקן במיקום הסמן.  
   צרו שתי פרוצדורות:

* מיקום הסמן - setCursorePosition
* ציור השחקן - drawCharacter

בתכנית הראשית זמנו את הפעולות במקום הפסיקות.

תזכורת תבנית יצירת הפעולה (פרוצדורה):

proc setCursorePosition

pusha

**…**

popa

ret

endp setCursorePosition

מחקו את הפסיקות של מיקום הסמן וציור השחקן בפעם הראשונה על לוח המשחק, וציירו את השחקן ע"י זימון הפעולות למיקום הסמן וציור השחקן.

שמרו, הריצו ובדוק שהקוד תקין.

1. בתוך הלולאה שנו את הפקודות ל"מחיקת השחקן" , הזזת הסמן וציור השחקן מחדש לפקודות המזמנות פעולות.

**הדרכה:**

שנו את צבע השחקן לשחור

השתמשו בפעולה drawCaracter במיקום בו נמצא הסמן כדי למחוק את השחקן.

שנו את הערכים של מיקום השחקן על ציר ה x וציר ה – y

שנו את הצבע של השחקן.

זמנו את הפעולה למיקום הסמן setCursorePosition

זמנו את הפעולה לציור מחדש של השחקן drawCaracter

שימו לב 🎔, מאחר ואנחנו בכל פרוצדורה "דוחפים" למחסנית את הערכים של הרגיסטרים ו"מושכים" אותם בסוף הפרוצדורה, אין צורך לשמור את הערך של רגיסטר cx - מונה הלולאה למשתנה.

**מהו אורך התכנית?**