

פרוצדורות

פרוצדורות (שגרות) הם קטעים של תוכניות שניתן לקפוץ אליהן, לבצע אותן ולחזור פקודה אחת לאחר פקודת הקפיצה שקראה לפרוצדורה.
כדי לקפוץ לפרוצדורה (או לקרוא לה) משתמשים בפקודה CALL ולאחריה שם הפרוצדורה.
פרוצדורה מסתיימת בפקודה RET .

היתרונות בשימוש בפרוצדורות הם :

- * אם יש קטע של תוכנית שמבצעים אותו יותר מפעם אחת הוא ירשם פעם אחת ונוכל לקרוא לו כמה פעמים שנרצה.
- * הפרוצדורות מאפשרות ומקלות את כתיבתן של תוכניות מודולריות (מבנה) , בשיטת התכנות המבנית נהוג לפרק בעיה גדולה למספר תתי בעיות קטנות וכותבים לתת בעיה מודול (מבנה) מפורט.
- נוח לכתוב כל מודול כפרוצדורה ולשלב בתוכנית הראשית את כל המודולים באמצעות סדרת קריאה לפרוצדורות.

תוכנית ראשית :

```
START:  CALL INIT  
        CALL PROGRAM  
        CALL PRINT_DATA  
        JMP START
```

תכנון מודולרי מאפשר לאנשים רבים להשתתף בפרוייקט גדול שבו כל מתכנן אחראי לפיתוח מודולים אחרים.

ביצוע תהליך קריאה לשיגרה וחזרה ממנה

כדי לבצע בהצלחה את הקפיצה לשגרות וחזרה מהן משתמש ה-CPU בזיכרון מסוג מחסנית.
מחסנית :

המחסנית ממוקמת בזיכרון RAM הפנימי והיא זיכרון מסוג LIFO-LAST IN FIRST OUT נכנס אחרון יוצא ראשון. כלומר אנו יכולים תמיד לשלוף מהמחסנית את הנתון האחרון שהוכנס אליה ולאחריו את הנתון הלפני אחרון. האוגר שמצביע על המיקום של הנתון האחרון שהוכנס למחסנית נקרא SP-STACK POINTER מצביע המחסנית.
ה-CPU משתמש במחסנית בצורה אינטנסיבית, בתהליך הקפיצה לשיגרה וחזרה ממנה. כאשר ה-CPU קופץ לביצוע השיגרה הוא שומר לפני זה את האוגר IP במחסנית, נותן לאוגר IP את הכתובת של השיגרה, מבצע את השיגרה עד שהוא מגיע לפקודה RET שבסיומה הוא שולף מהמחסנית את ערכו של IP, וכך חוזר פקודה אחת לאחר הפקודה CALL שקראה לפרוצדורה.

תהליך בזמן CALL

SP=100H, IP=1003H
SP=FFH

SP=FEH

$SP \leftarrow SP - 1$
 $[SP] \leftarrow IPH$
 $SP \leftarrow SP - 1$
 $[SP] \leftarrow IPL$

לפני פקודת CALL

דחיפת IP למחסנית

SP	→	כתובת	נתון
		FDH	
		FEH	IPL
		FFH	IPH
		100H	

התהליך בזמן הפקודה RET

SP=FEH

SP=FFH
SP=100H

$IPL \leftarrow [SP]$
 $SP \leftarrow SP + 1$
 $IPH \leftarrow [SP]$
 $SP \leftarrow SP + 1$