

מבוא.

ביחידה זו נתמקד במבנה התכנית ונכיר את הנחיות הקומפיילר הדרושות על מנת לייצר תכנית שלמה. לאחר מכן נראה כיצד ניתן לכפות טיפוס (המרה לצורך פעולה) מסוג בית או מילה על משתנה בזיכרון. ונראה כיצד ניתן לקבל כתובת יחסית של משתנה שהוגדר בזיכרון.

מבנה תכנית בשפת סף.

כל תכנית כוללת שטחים שהם לקריאה בלבד ושטחים שמיועדים לקריאה / כתיבה. כזכור, שטחי התכנית בשפת סף מחולקים למקטעים ודרושים בדרך כלל לפחות שלושה שטחי זיכרון על מנת שהתכנית תוכל לפעול כהלכה.

שטח עבור אחסון משתנים : Data Segment .

שטח זה הוא החלק בתכנית המכיל את כל המשתנים הגלובליים שהוגדרו ע"י המשתמש. הוא בעל גודל קבוע מאחר וכל הנתונים המאוחסנים בתוכו הוגדרו לפני שהתכנית נטענה לזיכרון. המשתנים והנתונים שמוגדרים בשטח זה יהיו זמינים לתכנית בכל זמן הריצה, עבור כל הפונקציות שלה.

שטח עבור אחסון הפקודות : Code Segment .

שטח זה מכיל את הפקודות עצמן. שמבצעות מניפולציה על הנתונים. גם שטח זה קבוע בגודלו וידוע מראש. בדרך כלל השטח עצמו מיועד לקריאה בלבד, אך ניתן בשפת סף, להגדיר בו גם משתנים.

שטח עבור מחסנית : Stack Segment .

שטח המחסנית מיועד כדי לשמור בו ערכים באופן זמני, תוך כדי ריצה. ערכים זמניים כאלו עשויים להיות משתנים לוקאליים, וארגומנטים המועברים לפונקציות. בשפת סף נהוג להשתמש במחסנית כאשר אין לנו מקום לאחסון ערך באוגרים, ואין לנו צורך או רצון ליצור משתנה זמני בזיכרון לצורך זה.

לכל השטחים הנ"ל יש צורך לתת שם חוקי כלשהו. השם אינו מילה שמורה, אך נהוג לתת שמות בעלי משמעות.

בדרך כלל מקטע הנתונים קרוי פשוט Data או Dseg . מקטע הקוד - Code או Cseg , ומקטע המחסנית נקרא Sseg .

כעת נתבונן במבנה של תכנית שלמה : (שלד של תוכנית תקנית)

```
data segment
data
ends
sseg segment stack
db 100 dup (?)
sseg ends
```

```
code segment assume
cs:code,ds:data,ss:sseg
start: mov
ax,data      mov
ds,ax
```

גוף
התוכנית

```
exit:  mov
ah,4ch
      int 21h
code
ends end
start
```

ההנחיה **Segment** :

מילה שמורה זו מורה לקומפיילר להקצות שטח (מקטע) חדש. במקטע זה יש להתחיל את הקצאת הזיכרון החל מכתובת יחסית 0. כל ההקצאות של השטחים עבור הנתונים או הפקודות במקטע הזה יעשו ברצף ויסתיימו כאשר נגיע להנחיה **Ends**.

ההנחיה **Ends** :

מילה שמורה זו נועדה כדי לציין את המיקום של סוף הקצאת הסגמנט שנפתח.

ההנחיה **Stack** :

מילה שמורה זו נועדה כדי להורות לקומפיילר שהסגמנט הנ"ל הוא מחסנית ולכן מתוך הקצאה זו יש צורך לחשב את ערכו ההתחלתי של אוגר **SP** בזמן שהתכנית נטענת לזיכרון.

ההנחיה **Assume** :

הנחיה זו אומרת לקומפיילר לאיזה אוגר סגמנט צריך לשייך את הכתובת היחסית של המקטע המותאם לו.

שתי הפקודות הראשונות;

```
start: mov ax,data
mov ds,ax
```

פקודות אלו מבצעות אתחול של אוגר DS לערך כתובת שבה נמצא מקטע הנתונים.

המילה DATA מייצגת כאן את הכתובת שנבחרה ע"י DOS ככתובת התחלת מקטע הנתונים, בזמן הטעינה של התכנית. מערכת DOS מקצה שטח עבור נתונים אך תכנית ה loader שלה אינה מאתחלת את אוגר DS ולכן יש צורך לבצע זאת ידנית. עבור אוגר CS ואוגר SS תכנית הטעינה מבצעת אתחול אוטומטי ולכן אין צורך לעשות זאת בתכנית עצמה. מאחר ואין אפשרות לבצע מיעון מיידי על אוגר סגמנט אז אנו משתמשים באוגר עזר (AX) כדי לטעון את אוגר DS.

התווית Start מציינת את נקודת ההתחלה של הקוד עצמו שיש לקמפל. זו אינה מילה שמורה.

שתי הפקודות האחרונות;

```
exit:  mov ah,4ch  
       int 21h
```

פקודות אלו יוצרות קריאה למערכת DOS על מנת לסיים את התכנית ולהחזיר שליטה למערכת ההפעלה. הקריאה ל DOS נעשית באמצעות הפעלת שירות פסיקה ייחודי (INT 21H) עם הפרמטר המתאים באוגר AH. מערכת ההפעלה כאשר תבצע הפקודה INT 21h ע"י המעבד, הבקרה תעבור לשטחי הקוד של מערכת DOS ותופעל הפסיקה. מערכת DOS מצפה לקבל פרמטר באוגר AH המציין מהו השירות הדרוש. הערך 4CH שנמצא באוגר מציין שירות לסיום התכנית והחזרת השליטה למערכת DOS.

השורה End Start :

ההנחיה END מורה לקומפיילר שהמודול לתרגום נגמר. השם הצמוד אל ההנחיה זו מציין את השורה הראשונה להרצה. למעשה הקומפיילר יכול לפרש זאת כך : "פה מסתיימת התכנית שהתחילה בתווית Start".