



ארכיטקטורה אסמבלי



הדליקו מצלמות
כבו מיקרופונים
השתיקו טלפונים
ארגנו מחברת וכלי כתיבה

שיעור 5

מחסנית המשך



ארכיטקטורה
אסמבלי



ארכיטקטורה
**מערכות
הפעלה**

מבוא לאסמבלי

אסמבלי – משתנים ופקודות

תנאים ולולאות

מחסנית ופונקציות

פרמטרים ופונקציות

פסיקות

סיכום אסמבלי

תרגיל בספות

מבוא למערכות הפעלה

כלי דיאגנוסטיקה ל-Windows

Processes and Threads

Memory

Linux shell

Shell המשך

מערכות קבצים

Bootstrapping

פרויקט סיכום מערכות הפעלה



בשיעור הקודם

למדנו על תנאים ולולאות באסמבלי

- למדנו על מבנה המחסנית
- וגם איך כותבים פונקציה
- תרגלנו את כתיבת הפונקציה שמחפשת מספר במערך.
- הקשר לנושא שנלמד היום ניהול המחסנית



שיעור 5

מחסנית המשך



סיכום

טיפול
בפרמטרים

סדר דחיפת
הפרמטרים למחסנית

שמירת
ערכי האוגרים

חזרה



מחסנית חזרה.

```
___ sum:
... גוף הפונקציה ;
Ret
ENDP ___
push 4
push ___
___ sum
```

- F
- E
- D
- C
- B
- A

1. השלימו את הקוד הבא

2. עבור כל תא במחסנית (F-A) רשמו את ערכו (מספר, שם או לא יודע) לאחר יצירת המסגרת

שיעור 5

מחסנית המשך



סיכום

טיפול
בפרמטרים

סדר דחיפת
הפרמטרים למחסנית

שמירת
ערכי האוגרים

מסגרת

שמירת ערכי האוורים

כאשר נכנסים לפונקציה יש צורך לוודא שהאוורים בהם אנו הולכים להשתמש לא מאבדים את הערך שלהם. בשביל זה, נשמור את הערך שלהם במחסנית ונשלוף מהמחסנית בסוף הפעולה.

האם אחראי לדאוג לשמירת האוורים מי שכותב את הפונקציה או מי שקורא לפונקציה?

מה דעתכם?
דברו

מוסכמות קריאה (Calling Conventions)

CODING IS AN ART



MODERN ART

האם אחראי לדאוג לשמירת האוגרים מי שכותב את הפונקציה או מי שקורא לפונקציה?

- בעקרון לא כ"כ חשוב מי יהיה אחראי.
מה שכן חשוב הוא שהן הכותב והן הקורא יגיעו להסכמה ביניהם מיהו האחראי (על מנת לא לבצע פעולה זרה פעמיים או לא לבצע אותה כלל).
- להסכמה זו קוראים מוסכמת קריאה (לפונקציה), או באנגלית – Calling Convention.

מוסכמות קריאה (Calling Conventions)

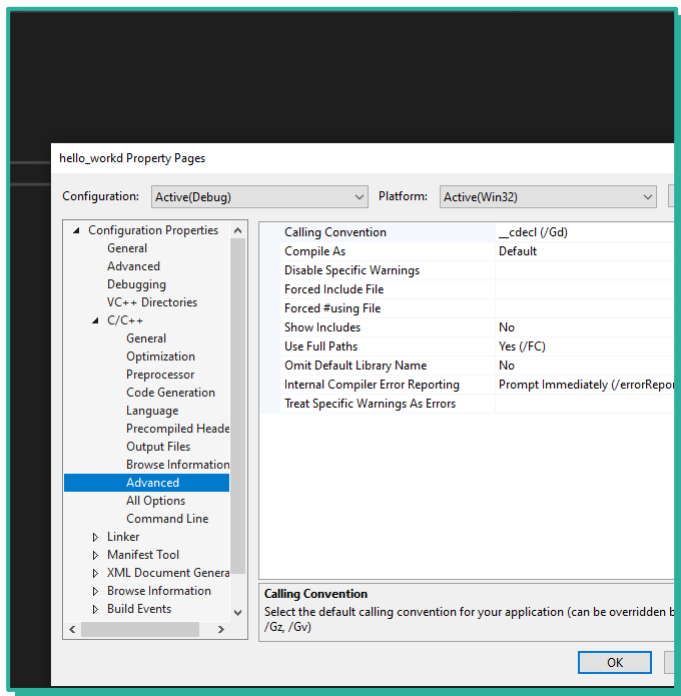
נביר שתי מוסכמות, שהן הכי נפוצות (תוכלו לקרוא עוד על שתי השיטות בסיכום השיעור):

- Cdecl
- STDCall
-

לפי שתי המוסכמות הנ"ל, שמירת ערכי האוגרים

AX, CX, DX היא באחריות הקורא לפונקציה. השאר באחריות כותב הפונקציה (הפונקציה הנקראת).

visual studio



- בהגדרות של כל פרויקט נבחר באיזה קונבנציה נשתמש, ניתן לראות כי ברירת המחדל היא cdecl עבור visual studio

שיעור 5

מחסנית המשך



סיכום

טיפול
בפרמטרים

סדר דחיפת
הפרמטרים למחסנית

שמירת
ערכי האגרים


מסגרת

סדר דחיפת הפרמטרים למחסנית

- בואו נתבונן בחתימה של הפונקציה הבאה:

```
void func (int first, int second);
```

- באיזה סדר נדחוף את הפרמטרים למחסנית?




נכון

```
push 5 ; second  
push 3 ; first  
call func
```

או

```
push 3 ; first  
push 5 ; second  
call func
```



נכון

מה דעתכם?
דברו

סדר דחיפת הפרמטרים למחסנית

- **שוב, הסדר לא כל כך חשוב.**
מה שחשוב הוא שהן הכותב הפונקציה והן הקורא (לפונקציה) יגיעו להסכמה ביניהם.
- לפי Cdecl ו- STDCALL , סדר דחיפת פרמטרים למחסנית הוא בסדר הפוך לסדר הקריאה להם בשפת C. לדוגמא:

```
.ASM
push b    ; second
push a    ; first
call func
```

```
.C
func(a, b);
```

מנוחת אלופים!

זמן להפסקה קצרה



שיעור 5

מחסנית המשך



סיכום

טיפול
בפרמטרים

סדר דחיפת
הפרמטרים למחסנית

שמירת
ערכי האגרים

חזרה

טיפול נכון בפרמטרים

היזכרו בתרגיל 3 (עבודת בית 4), שבו ביקשנו לממש פונקציית main הקוראת לפונקציית max:-imin

```
void max(int *maxNumber, int num0, int num1);  
void min(int *minNumber, int num0, int num1);
```

יכול להיות ששכחנו משהו?

מה דעתכם?
דברו



מחסנית המשך:

- מקטע המחסנית המוקצה לכל תכנית מוגבל
- אזור זה בזיכרון מנוהל ע"י המתכנת ולכן עלינו לדאוג להשתמש בו נכון
- בהמשך נדבר על מה הסכנות של **חריגה ממסגרת המחסנית (stack overflow)**
- כרגע נעסוק בלהבין איך לדאוג לכך שבתכנית שכתבנו לא יקרה מצב בעייתי שכזה



יצירה וקיפול של מסגרת

הפעל

```
push 4  
push 2  
call func  
push 6  
push 8  
call func  
...
```

- נסתכל על המחסנית ביחד ל BP ומה יקרה כתוצאה מהקוד הבא

	00F6
func ip	00F8
8	00FA
func ip	00FC
2	00FE
4	0100

SP

תרגיל כיתה

לאחר הרבה קריאות באלה, בסוף המחסנית תתמלא
מה עושים? **ננקה אותה!**
איך? **תרגיל כיתה!**

פתחו את תרגיל 3 (עבודת בית 4) ותקנו אותו כך שלאחר חזרה
מהפונקציה המחסנית תישאר ללא הפרמטרים



איך אתם עשיתם את זה!?

כמו שוודאי שמתם לב, יש יותר מדרך אחת לפתור את הבעיה, וכולן טובות!

- השאלה באיזו קונבנציה (מוסכמה) עובדים
- אפשר לדאוג לנקות את המחסנית מהפרמטרים עוד בתוך הפונקציה הנקראת (לפי קונבנציית **STDCall**)
- אפשר לדאוג לזה אחרי הקריאה בפונקציה הקוראת (לפי קונבנציית **Cdecl**). דוגמא:
`add sp, number`
- תזכורת: תוכלו לקרוא עוד על שתי השיטות בסיכום השיעור

ניקוי פרמטרים – דוגמא בשיטת Cdecl

main

```
PUSH AX
PUSH BX
PUSH CX
CALL special
ADD sp,6
MOV AH, 0
INT 16h
RET
```

special

```
PROC special
PUSH BP
MOV BP, SP
; עשה משהו מיוחד
MOV SP, BP
POP BP
RET
ENDP special
```

מה דעתכם?
דברו

ניקוי פרמטרים – דוגמא בשיטת STDCALL

main

```
PUSH AX  
PUSH BX  
PUSH CX  
CALL special
```

```
MOV AH, 0  
INT 16h  
RET
```

special

```
PROC special  
PUSH BP  
MOV BP, SP  
; עשה משהו מיוחד  
MOV SP, BP  
POP BP  
RETN 6  
ENDP special
```

מה דעתכם?
דברו



ביצוע Hands-On
*נמצא בתיקיית החניכים



סיכום שיעור



תרגיל הבית



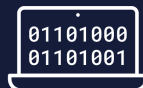
מעבר נוסף
על המצגת

איך כדאי לחזור
על החומר?

שאלות? תהיות?

תכתבו עכשיו שאלות בצ'ט

ארכיטקטורה
אסמבלי



מגזר ימים
תכנית הסייבר הלאומית

המרכז לחינוך סייבר
CYBER EDUCATION CENTER



שיעור 5 סיימנו!!!

סיכום

טיפול
בפרמטרים

סדר דחיפת
הפרמטרים למחסנית

שמירת
ערכי האוגרים

חזרה