

## תרגיל 2 - חלק יבש

המתרגל האחראי על התרגיל: תומר כץ.

שאלותיכם במייל בעניינים מנהלתיים בלבד, יופנו רק אליו.

כתבו בתיבת **subject**: יבש 2 את"ם.

שאלות בעל-פה ייענו על ידי כל מתרגל.

### הוראות הגשה:

- לכל שאלה יש לרשום את התשובה במקום המיועד לכך.
- יש לענות על גבי טופס התרגיל ולהגיש אותו באתר הקורס כקובץ PDF.
- על כל יום איחור או חלק ממנו, שאינו בתיאום עם המתרגל האחראי על התרגיל, יורדו 5 נקודות.
- הגשות באיחור יש לשלוח למייל של אחראי התרגיל בצירוף פרטים מלאים של המגישים (שם+ת.ז).
- שאלות הנוגעות לתרגיל יש לשאול דרך הפיאצה בלבד.
- ההגשה בזוגות.

## שאלה 1 (45 נק') – שגרות:

ג'וני סטודנט אחראי כל בוקר בשש ו30 כותב קוד אסמבלי. לפניכם מקטע הנתונים שג'וני כתב:

1	.section .data	14	E: .int 8
2	A: .long 3	15	.quad 0
3	.quad B	16	.quad 0
4	.quad C	17	F: .int 9
5	B: .long 4	18	.quad G
6	.quad D	19	.quad H
7	.quad 0	20	G: .int 10
8	C: .int 5	21	.quad 0
9	.quad E	22	.quad 0
10	.quad F	23	H: .int 11
11	D: .int 7	24	.quad 0
12	.quad 0	25	.quad 0
13	.quad 0	26	

1. ציירו את הגרף המתקבל מפירוש מקטע הנתונים (מומלץ להסתכל בתרגול 3 תרגיל 1 ולהיזכר שם על אופן פירוש הזיכרון כרשימה מקושרת). בכל צומת בגרף ציינו את התווית המתאימה לו בלבד (אין צורך לציין ערכים נוספים) (3 נקודות)

ג'וני לא מפחד משגרה שוחקת ולכן כותב את השגרה func וקוד שמתשמש בה:

```
27 .section .text
28 .global _start
29 _start:
30     mov $8, %esi
31     mov $A, %rdi
32     call func
33     movq $60, %rax
34     movq $0, %rdi
35     syscall
36
37 func:
38     pushq %rbp
39     movq %rsp, %rbp
40     cmp (%rdi), %esi
41     jne continue
42     mov $1, %eax
43     jmp finish
44
45 continue:
46     cmpq $0, 4(%rdi)
47     je next
48     pushq %rdi
49     mov 4(%rdi), %rdi
50     call func
51     pop %rdi
52     cmp $1, %eax
53     je finish
54 next:
55     cmpq $0, 12(%rdi)
56     je fail
57     pushq %rdi
58     mov 12(%rdi), %rdi
59     call func
60     pop %rdi
61     cmp $1, %eax
62     je finish
63 fail:
64     mov $0, %eax
65 finish:
66     leave
67     ret
```

2. נתון שבתחילת התוכנית ערך של `rsp` הוא `x`. באשר `x` הוא מספר בקסדצימלי. מה הוא הערך המקסימלי ומה הערך המינימלי ש`rsp` יכול לאורך ריצת התוכנית? תנו נוסחא שהמספרים בה הם בבסיס הקסדצימלי (בטאו את התשובה בהאמצעות `x`). (5 נקודות)

---

---

3. רשמו מה יהיה פלט הפונקציה עבור קטע הקוד הנוכחי (7 נקודות)

---

---

4. המירו את הפונקצייה לשפת C על ידי כך שתשלימו את המקומות החסרים בקוד. העיזרו בהגדרת structn שנתונה לכם (10 נקודות):  
 structn הנתון:

```
typedef struct _Node {
    int data;
    struct _Node *left;
    struct _Node *right;
} Node;
```

הערה 1: שני הפרמטרים צריכים להיות תואמים לשני הפרמטרים של פונקציית האסמבלי גם מבחינת תפקיד וגם מבחינת סדר. כלומר, root צריך להתאים בתפקידו לפרמטר הראשון שמועבר לפונקציה בשפת אסמבלי גם מבחינת הקונבנציה שלמדנו.

הערה 2: אורך הקו לא מלמד על אורך האיבר שצריך להשלים. מותר להשלים יותר ממילה אחת בכל קו אך לא יותר מפקודה אחת!

```
_____ func (Node* root, _____ x){

    If (root->data == _____)

        _____;

    if (root->left != null)

        if (_____)

            return _____;

    if (root->right != null)

        return _____;

    return 0;

}
```

הערה: בסעיפים הבאים יש כל מיני שינויים בקוד. כל שינוי מתקיים רק בסעיף בו מופיע. זאת אומרת הסעיפים לא תלויים אחד בשני.

5. מוני חבר של ג'וני הוא לא כמו ג'וני. הוא אוהב לעשות שינויים רבים בקוד. הוא מחליט לקחת את המקטע הנתונים של ג'וני ולשנות בכל struct את ה-int ב-quad. כלומר מקטע הנתונים ישתנה כך:

```
1 .data
2 A: .quad 3
3 .quad B
4 .quad C
5 B: .quad 4
6 .quad D
7 .quad 0
```

ובאופן דומה כל שאר האותיות יחליפו את הנתון הראשון ב-quad במקום ה-int. רשמו את השינויים שצריכים להיות בקוד על מנת שיעבוד בצורה תקינה עם מקטע הנתונים החדש (5 נקודות)

---

---

---

6. ג'וני מתחיל להתעייף מהשגרה ומחליט לקום ולשנות את מבנה הנתונים באופן הבא:

```
11 D: .int 7
12 .quad A
```

מה יהיה פלט התוכנית? יש לסמן תשובה מבין התשובות הבאות ולנמק בקצרה: (5 נקודות)

- התוכנית תסתיים ופלט הפונקציה יהיה 1
- התוכנית תסתיים ופלט הפונקציה יהיה b
- התוכנית תכנס ללולאה אנסופית
- התוכנית תקרוס במהלך ריצה
- התוכנית כלל לא תבנה

נימוק: \_\_\_\_\_

---

7. פתאום ג'וני כמו מוני! מחליט לבצע שינויים נוספים ולא שגרתיים בקוד מול כל שינוי שג'וני מציע עליכם לכתוב האם נכונות השגרה תיפגע (האם יש קלט עבורו השגרה לאחר השינוי שונה מהשגרה לפני השינוי). הסבירו בקצרה את תשובתכם! (10 נקודות)

- מחיקת הפקודת push וpop שבשורות 60 ו-57..
- מחיקת הפקודה pop בשורה 60
- מחיקת push וpop שבשורות 48 ו-51
- הוספת פקודה push %rdi אחרי continue בשורה 45
- הוספת הפקודה push %rdi אחרי continue בשורה 45, שינוי פוקדת pop שבשורה 51 לפקודה: mov(%rsp,%rdi) ומחיקת הפקודת push וpop שבשורות 60 ו-57.

נימוק: \_\_\_\_\_

---

## שאלה 2 (30 נק') – קריאות מערכת:

ג'ואי מרגיש מתוסכל מכך שחבריו חושבים שהוא פחות חכם מהם. לכן, הוא מחליט להרשים אותם בעזרת כתיבת קוד אסמבלי.

1. לפניכם מקטע הנתונים שג'ואי כתב מבלי ערכי הנתונים עצמם:

```
.section .data
msg1: .ascii ???????
msg2: .ascii ???????
msg1_len: .quad ____
msg2_len: .quad ____
all_msg_len: .quad ____
```

ג'ואי לא יודע עדיין אילו מחרוזות הוא יכתוב. עליכם להשלים את המקומות הריקים שקשורים לאורכי המחרוזות כך שמשתנה `msg1_len` יהיה האורך של `msg1`, `msg2_len` יהיה האורך של `msg2` ובמשתנה `all_msg_len` יהיה שווה לסכום אורכי המחרוזות `msg1` ו-`msg2`. שימו לב עליכם לעשות זאת בצורה כזו שהאורכים יהיו נכונים בעת ריצת התוכנית ללא קשר לאיזה מחרוזות ג'ואי ישים ב-`msg1` וב-`msg2`. (3 נקודות)

2. בעת נתון מקטע הנתונים שכולל את המחרוזות:

```
.section .data
msg1: .ascii "HOW YOOOU DOOIN?"
msg2: .ascii "JOEY DOESN'T SHARE FOOD!"
msg1_len: .quad ____
msg2_len: .quad ____
all_msg_len: .quad ____
```

לפניכם נתונה התוכנית שג'ואי כתב:

```
.section .text
.global _start
_start:
    mov $msg1, %rsi
    mov $1, %rdi
    mov $1, %rdx
    mov $1, %rax
    xor %rbx, %rbx

    movq msg1_len, %r9
    call Joey_func
```

ומוצגת כאן גם הפונקציה שכתב:

```
Joey_func:
    cmp %rbx, %r9
    je end
    # addb $0x20, (%rsi)
    test $1, %rbx
    jnz skip
    syscall
skip:    inc %rsi
        inc %rbx
        call Joey_func
end:    ret
```

מה יודפס בסיום ריצת הקוד? (שימו לב השורה השלישית בפונקציה נמצאת בהערה ולא רלוונטית לסעיף). (5 נקודות)

3. כעת מורידים את הסולמית שנמצאת בפונקציה (וכעת הפקודה חלק מהקוד) בנוסף מחליפים את השורה  
9movq msg1\_len, %r בשורה: 9movq all\_msg\_len, %r.  
הערה: שינויים אלו ילוו אותנו גם בסעיפים הבאים (בסעיפים ד - ו השינויים בסעיף ג עדיין תקפים).  
מה יודפס כעת בסיום ריצת הקוד? (5 נקודות)

---

---

4. בזמן שג'ואי אכל בסלון סנדוויץ, חיית המחמד שלו (אפרוח) טיילה על המקלדת והוסיפה את הפקודה:  
9inc %r. הפקודה נוספה שורה לפני הקריאה לפונקציה של ג'ואי בתוכנית הראשית.  
מה יהיה פלט התוכנית כעת? (2 נקודות)

---

5. חברה טובה של ג'ואי פיבי אמרה לו ששימוש ברגיסטר 9r מביא מזל רע. ג'ואי נלחץ נורא והחליט שיש לבצע  
שינוי בקוד מבלי לשנות את תוצאות הפעולה של הפונקציה (כלומר הפלט צריך להיות זהה). כיוון ולא ידע  
איך לשנות את הקוד הוא החליט לבקש את עזרת חבריו.  
בסעיף הזה יופיעו העצות של כל החברים. עליכם לרשום ליד כל עצה האם היא לדעתכם תעזור לג'ואי. נמקו  
בקצרה (!) (10 נקודות)

○ צ'נדלר מציע להחליף את השימוש ב9r בשימוש בrcx.

---

---

○ מוניקה מציעה להחליף את השימוש ב9r בשימוש ב11r.

---

---

○ פיבי מציעה להחליף את השימוש ב9r בשימוש בrdi.

---

---

○ רייצ'ל מציעה להחליף את השימוש ב9r בשימוש ב12r.

---

---

○ רוס מציע להחליף את השימוש ב9r בשימוש בrbp.

---

---

6. חבריו של ג'ואי מסבירים לו שהשימוש שלו ברקורסיה מיותר ובזבזני והוא יכול את אותו קוד בדיוק לכתוב

בלולאות. ג'ואי מחליט לבצע את השינויים הבאים:

בתוכנית הראשית בשורה שלפני ביצוע הפקודה call ג'ואי מוסיף את הפקודה:

```
mov $Joey_func, %rcx
```

ובתוך הפונקציה ג'ואי מוחק את השורה בה יש שימוש בפקודה call והחליף אותה בפקודה:

```
jmp *%rcx
```

שימו לב שהתווית end נמצאת אחרי פקודה זו.

לצורך הבהרה הפונקציה נראת כך כעת:

```
Joey_func:
    cmp %rbx, %r9
    je end
    addb $0x20, (%rsi)
    test $1, %rbx
    jnz skip
    syscall
skip:    inc %rsi
        inc %rbx
        jmp *%rcx
end:    ret
```

כיצד שינוי זה ישפיע על אופן ריצת הפונקציה. מה יודפס אם נריץ את הפונקציה? (5 נקודות)

---

---



## שאלה 3 (25 נק') – רמות הרשאה ואוגר הדגלים:

1. הפקודה `pushfq` דוחפת את הערך של אוגר הדגלים למחסנית. והפקודה `popfq` מוציאה את אוגר הדגלים מהמחסנית. הסבירו כיצד באמצעות שילוב של שתי פקודות אלו ניתן להדליק את הדגלים `CF` ו `OF`. שימו לב במידה ואחד הדגלים כבר דלוק יש להשאירו דלוק כלומר, בסיום התהליך על שני הדגלים להיות דולקים. אין לשנות את שאר הביטים בריגסטר הדגלים. בנוסף, אין לשנות אף רגיסטר שהוא לא `rflags`, `rip`, `rsp` (גם לא באופן זמני). (7 נקודות)

הערה: במידה ובדקתם את עצמכם באמצעות דיבגר וראיתם שנדלק גם דגל `TF` זה בסדר תלמדו בהמשך מדוע הוא נדלק תוך כדי דיבוג.

---

---

---

---

2. הולי התחמנית רוצה לאפשר לעצמה גישה ישירה אל התקני הקלט פלט ללא צורך בקריאות מערכת. איזה שינוי באוגר הדגלים יכול לעזור להולי במטרתה? (4 נקודות)

הערה: לא צריך לציין פקודה ספציפית, רק להגיד מה צריך לעשות ברמה התיאורטית

---

---

---

---

3. הולי מחליטה לנסות את התעלול מסעיף א' רק שבמקום לשנות את `CF` ו `OF` היא רוצה לשנות את `IOPL`. להפתעתה, היא לא מצליחה לשנות את הביטים הללו. הסבירו מה ההגיון בכך שהיא לא מצליחה לשנות את `IOPL`? התייחסו לצורך בקריאות מערכת (4 נקודות)

---

---

---

---

הערה: הסעיפים הבאים קשורים לפסיקות מומלץ לענות עליהם לאחר התרגול על פסיקות.

4. וולי החבר המבולבל של הולי מתלבט כיצד ניתן לחסום פסיקות תוכנה לכן הוא שואל את הולי. אילו מבין התשובות הבאות על הולי לענות לו? יש לסמן את האפשרות הנכונה וגם לנמק בקצרה (5 נקודות)

- כיבוי דגל IF באוגר הדגלים
- הדלקת דגל IF באוגר הדגלים
- שינוי CPL ל00
- לא ניתן לחסום פסיקות תוכנה.

נימוק:

---

5. כעת נתון שוולי הצליח להגיע למצב שבו CPL שווה ל0. וולי מעוניין לחסום פסיקות חומרה שאינן מועברות דרך כניסת NMI. כיצד הוא יכול לעשות זאת? יש לסמן את האפשרות הנכונה וגם לנמק בקצרה (5 נקודות)

- כיבוי דגל IF באוגר הדגלים
- הדלקת דגל IF באוגר הדגלים
- עליו לחבר את הפסיקות לכניסת NMI ואז לכבות את דגל IF
- לא ניתן לחסום פסיקות חומרה ולכן לא יצליח.

נימוק:

---