### הטכניון – מכון טכנולוגי לישראל

**ארגון ותכנות המחשב**

תרגיל 2 - חלק יבש

המתרגל האחראי על התרגיל: איתי אילת.

שאלותיכם במייל בעניינים מנהלתיים בלבד, יופנו רק אליו.

**כתבו בתיבת subject: יבש 2 את"ם.**

שאלות בעל-פה ייענו על ידי כל מתרגל.

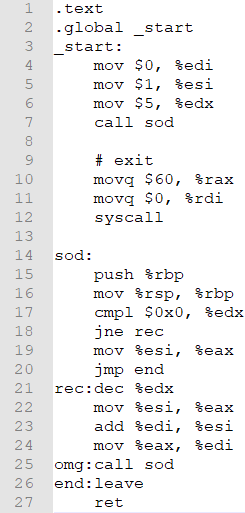
הוראות הגשה:

* 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 (35 נק') – שגרות:

נתונה התוכנית הבאה בשפת אסמבלי:



1. מה פלט הפונקציה (בהתאם לנתונים שמופיעים בתוכנית). (5 נקודות)

בחזרה מהפונקציה sod נקבל eax = 0x8 וזהו פלט הפונקציה (לא ידוע מה קורה ב32 ביט העליונים של rax)

1. האם הפונקציה עומדת בקונבנציה System v שנלמדה בתרגול? הקיפו בעיגול (או סמנו עם מרקר במחשב): (3 נקודות)

**כן / לא**

1. המירו את הפונקצייה לשפת c על ידי כך שתשלימו את המקומות החסרים בקוד (10 נקודות):

**הערה:** שלושת הפרמטרים צריכים להיות תואמים לשלושת הפרמטרים של פונקצית האסמבלי גם מבחינת תפקיד וגם מבחינת סדר. כלומר, num1 צריך להתאים בתפקידו לפרמטר הראשון שמועבר לפוקנציה בשפת אסמבלי גם מבחינת הקונבנצייה שלמדנו.

Int sod ( int num1, int num2, int n){

If (n == 0)

return num2;

return sod(num2, num1+num2, n-1);

}

1. **הערה: אין צורך לדעת קידוד פקודות בשביל לפתור סעיף זה. אבל יש צורך להבין שפקודה שמורה** **בזיכרון כרצף ביטים בדיוק כמו במקטע הנתונים ואחר כך המעבד מפרש את קידוד הפקודה לפקודה.**

נתונה טבלה שמייצגת את קידוד הפקודות שנמצאות בשורות 22-24

|  |  |  |  |
| --- | --- | --- | --- |
| שורה | 22 | 23 | 24 |
| קידוד | 0xf089 | 0xfe01 | 0xc789 |

החליטו להכניס בין השורה 19 לשורה 20 את הפקודות הבאות:

19.1 mov $omg-2, %rsi

19.2 mov %rsi, 8(%rbp)

נתון שלאחר השינוי הכתובת של שורה 25 (omg: call sod) היא: 0x4005eb והערך שבכתובות זו הוא: 0xe8.

1. הקיפו בעיגול את התשובה הנכונה: (5 נקודות)
2. התוכנית כלל לא תיבנה.
3. התוכנית תסיים את ריצתה אך פלט השגרה עלול להיות שונה מהפלט של התוכנית המקורית
4. התוכנית לא תסיים את ריצתה לעולם.
5. התוכנית תסיים את ריצתה עקב שגיאה במהלך זמן ריצה.
6. מה יהיה הערך של אוגר eax בסוף ריצת השגרה לאחר השינוי? שימו לב במידה ובסעיף הקודם רשמתם שהתוכנית לא מסתיימת עקב לולאה אינסופית רשמו מה יהיה הערך של האוגר אחרי כמה איטרציות בלולאה. במידה ועניתם שהתוכנית תקרוס בזמן ריצה רשמו מה היה הערך של eax ברגע הקריסה. במידה ולדעתכם לא ניתן לדעת את הערך יש לרשום זאת! (5 נקודות)

eax = 0x4005e9

1. (שימו לב השינוי בסעיף 4 לא תקף עבור סעיף זה) החליטו לשנות את הפקודה שבשורה 26 מleave לפקודה: add $24, %rsp.

עבור אילו ערכי edx (הערך שedx מקבל בתוכנית הראשית. שורה 5) הפונקציה תחזיר את אותם ערכים בדיוק כמו השגרה המקורית? הסבירו (7 נקודות)

כל ערך אי זוגי.

*הסבר: ראשית כל, עבור ערך אי זוגי, יתצעו מס זוגי (גדול ב1) של קריאות רקורסיביות (כי זה כולל את 0 בו ייבדק התנאי).  
בכל קריאה רקורסיבית המחסנית גדלה ב16 בתים (8 על ידי call שדוחפת את כתובת החזרה ועוד 8 על ידי push %rbp בתחילת הפונקציה).  
באופן שגרתי בקוד זה, בכל חזרה מקריאה, leave מקטינה את המחסנית ב- 8 בתים (של %rbp) ולאחר מכן, ret מקטינה עוד 8 של כתובת החזרה ובסה"כ מקטינים את המחסנית ב16.  
כאשר מקטינים באופן ישיר את המחסנית ב24 ומבצעים ret, (בו המחסנית קטנה בעוד 8) למעשה מקטינים את המחסנית ב32 בתים וטוענים כתובת חזרה של המסגרת שני דורות מעלה ברקורסיה לעומת עליה אחת באופן רגיל. (32 = 2\*16)*

לכן עבור כמות זוגית של קריאות רקורסיביות (שנובעות מערך rdi התחלתי אי זוגי) נקרא אחרי מחצית מכמות החזרות הרגילה את ערך החזרה אל התוכנית הראשית.  
*ערך התוצאה הסופית (החישוב) קיים כבר בשלב העמוק ביותר ברקורסיה, כל צעדי החזרה הם אך ורק על מנת לשחזר את כתובת הקוד של התוכנית ממנה באנו ולא לצורך חישובים נוספים - לכן דילוג על מסגרות של קריאות רקורסיביות בדרך חזרה לא פוגעת בפעילות התקינה, כל עוד מגיעים לכתובת החזרה המקורית של הקוד הלא רקורסיבי.*

# שאלה 2 (38 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:

section .data

msg1: .ascii "HOW YOOOU DOOIN?"

msg2: .ascii "JOEY DOESN'T SHARE FOOD!"

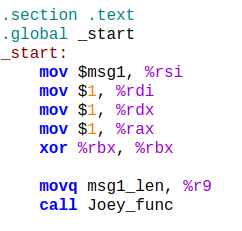
msg1\_len: .quad \_msg2-msg1\_\_

msg2\_len: .quad \_\_msg1\_len-msg2\_\_

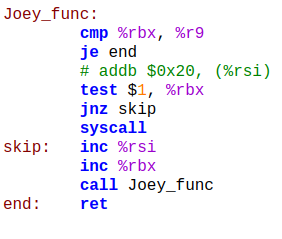
all\_msg\_len: .quad \_msg1\_len-msg1\_\_

עליכם להשלים את המקומות הריקים כך שמשתנה msg1\_len יהיה האורך של msg1, בmsg2\_len יהיה האורך של msg2 ובמשתנה all\_msg\_len יהיה שווה לסכום אורכי המחרוזות msg1 וmsg2. (3 נקודות)

1. כעת נתונה התוכנית הראשית שג'ואי כתב:



ובנוסף נתונה פוקנציה שהוא כתב:



מה יודפס בסיום ריצת הקוד? (שימו לב השורה השלישית בפונקציה נמצאת בהערה ולא רלוונטית לסעיף). (6 נקודות)

HWYOUDON

1. כעת מורידים את הסולמית שנמצאת בפונקציה (וכעת הפקודה חלק מהקוד) בנוסף מחליפים את השורה movq msg1\_len, %r9 בשורה: movq all\_msg\_len, %r9.

**הערה: שינויים אלו ילוו אותנו גם בסעיפים הבאים (בסעיפים 4 - 6 השינויים בסעיף 3 עדיין תקפים).**

מה יודפס כעת בסיום ריצת הקוד? (7 נקודות)

hwyoudonje@osG@hr@od

1. בזמן שג'ואי אכל בסלון סנדוויץ, חיית המחמד שלו (אפרוח) טיילה על המקלדת והוסיפה את הפקודה:

inc %r9. הפקודה נוספה שורה לפני הקריאה לפונקציה של ג'ואי בתוכנית הראשית.

מה יהיה פלט התוכנית כעת? (5 נקודות)

hwyoudonje@osG@hr@od0

1. חברה טובה של ג'ואי פיבי אמרה לו ששימוש ברגיסטר r9 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)

רייצ'ל מציעה להחליף את השימוש בr9 בשימוש בrdi.   
לא! זה ישנה את ערוץ הפלט בקריאת המערכת (rdi מציין מה ערוץ הקלט ב sys\_write)

מוניקה מציעה להחליף את השימוש בr9 בשימוש בr10.  
 כן, אין בו שימוש בתוכנית ולא יישתנה ע"י הקריאת מערכת.

צ'נדלר מציע להחליף את השימוש בr9 בשימוש בr11.  
לא! קריאת מערכת שומרת את רגיסטר rflags ברגיסטרr11 (ערכו יידרס)

רוס מציע להחליף את השימוש בr9 בשימוש בrbp.  
כן, יעבוד במקרה זה כי אין שימוש בrbp אבל קוד גרוע, כי משתמש בrbp שלא לייעוד שלו.

פיבי מציעה להחליף את השימוש בr9 בשימוש בrcx.  
 לא! קריאת מערכת שומרת את רגיסטר rip ברגיסטר rcx (ערכו הקודם יידרס)

1. 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:

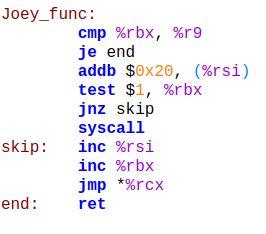
mov $Joey\_func, %rcx

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 נראית כך כעת:



כיצד שינוי זה משפיע על אופן ריצת הפונקציה. מה יודפס אם נריץ את הפונקציה? (7 נקודות)

התוכנית מדפיסה h, (קריאת המערכת הראשונה מתבצעת) אבל שומרת את רגיסטר rip ברגיסטר rcx (ערכו הקודם יידרס), לכן כשננסה לקפוץ לכתובת בrcx, למעשה נגיע לשורה שלאחר קריאת המערכת, ומשם נמשיך בלולאה אינסופית ולא יודפס דבר נוסף.

# שאלה 3 (27 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 דגל SF. שימו לב במידה והדגל כבר דלוק יש להשאירו דלוק. אין לשנות את שאר הביטים בריגסטר הדגלים. בנוסף, אין לשנות אף רגיסטר שהוא לא rflags, rip, rsp (גם לא באופן זמני). (8 נקודות)

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

pushfq העברה לזיכרון

or $128, (%rsp) #128 = 2^7 -> SF is the 7th bit כתיבה לזיכרון (אין הגנה)

popfq החזרה מהזיכרון (ששונה) למקום

באופן זה, לא ניגשנו באופן ישיר לrflags (פעולה שהייתה נחסמת) אבל בכל זאת הצלחנו לשנות ערך בו כי השתמשנו בהורדה לזיכרון והעלאה ממנו ובשלב הביניים כתבנו לזיכרון (דבר שמותר)

1. הולי התחמנית רוצה לאפשר לעצמה גישה ישירה אל התקני הקלט פלט ללא צורך בקריאות מערכת. איזה שינוי **באוגר הדגלים** יכול לעזור להולי במטרתה? (8 נקודות)

אם הולי תשנה בצורה דומה את IOPL, שמכיל את רמת ההראשה הנדרשת לביצוע פעולות I/O

0 זו רמת ההרשאה הגבוה ביותר ולכן תרצה לאפס את שני הביטים של IOPL.

1. הולי מחליטה לנסות את התעלול מסעיף א' רק שבמקום לשנות את SF היא רוצה לשנות את IOPL. להפתעתה, היא לא מצליחה לשנות את הביטים הללו. הסבירו מה ההיגיון בכך שהיא לא מצליחה לשנות את IOPL? התייחסולצורך בקריאות מערכת (5 נקודת)

לא ניתן לשנות את ערכם ברמת הרשאת משתמש, אילו היה אפשר, לא היה בהם צורך מלכתחילה.

הגיוני שלא ניתן לשנותם בהרשאת משתמש כי אם היה אפשר אז היה ניתן להעלות את רמת ההרשאה ולבצע פעולות של הרשאת גרעין ממרחב המשתמש.  
בעת ביצוע popfq ביטים אלו לא מתעדכנים כי רמת ההרשאה לא מתאימה (הפעולה נחסמת).

1. הערה: סעיף זה קשור לפסיקות מומלץ לענות על סעיף זה לאחר התרגול על פסיקות.

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. (6 נקודות)

1. כיבוי דגל IF באוגר הדגלים
2. הדלקת דגל IF באוגר הדגלים
3. שינוי CPL ל00
4. לא ניתן לחסום פסיקות תוכנה.