### 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.

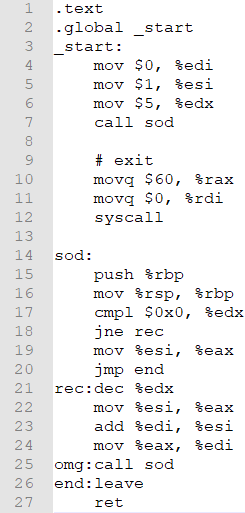
הוראות הגשה:

* 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)

8

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

**כן / לא אין שמירת רגיסטרים לפני הקריאה שגרה**

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)

5

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)

רק עבור ערכי edx אי-זוגיים הפונקציה תחזיר את אותם ערכים. הוספת 24 לrsp גורמת לפקודה ret להחזור לא לקריאה הקודמת, אלה אחת לפנייה (קופצים על כתובת החזרה של הקריאה הקודמת). ולכן על מנת להגיע לקריאה הראשונה ולא להגיע למקום שלא שייך לאף קריאה לשגרה חייב מספר זוגי של קריאות, שזה קורה רק עבור edx אי זוגי

# 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!"

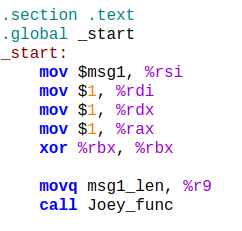
msg1\_len: .quad msg2 – msg1

msg2\_len: .quad msg1\_len – msg2

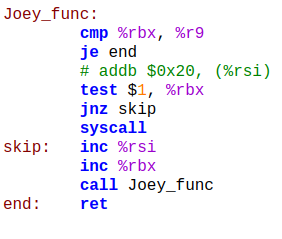
all\_msg\_len: .quad msg1\_len – msg1

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)

בסיום ריצת הקוד יודפסו כל התווים אשר נמצאים באינדקס א"ז.

כלומר:

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)

בסיום ריצת הקוד יודפסו הפעם האותיות באינדקס א"ז מ-2 המילים הנתונות כאשר:

* כל תו המייצג אות לטינית יעבור לlower-case
* התו רווח יוצג כ@
* התו ' יהפוך לG
* התו ? יהפוך ל\_ (למרות שלא יודפס כלל)

כלומר יודפס

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, מכיוון שRDI משמש כקלט לקריאת המערכת המציין את סוג השירות לכן דריסה שלו תגרור התנהגות לא צפויה.

מוניקה מציעה להחליף את השימוש בr9 בשימוש בr10.

כן, הרגיסטר פנוי לשימוש הן מבחינת קריאת המערכת והן מבחינת אי-תלות הקוד ברגיסטר זה.

צ'נדלר מציע להחליף את השימוש בr9 בשימוש בr11.

לא, הרגיסטר r11 נדרס לאחר קריאת המערכת (RFLAGS מושם אליו) ולכן נקבל התנהגות לא צפויה.

רוס מציע להחליף את השימוש בr9 בשימוש בrbp.

כן, מכיוון שגואי לא שמר על קונבנציות caller/callee אין חשיבות לתוכן rbp בשלב זה של ריצת הקוד לכן ניתן להשתמש בו.

פיבי מציעה להחליף את השימוש בr9 בשימוש בrcx.

לא, כמו במקרה של צ'נדלר הרגיסר rcx נדרס לאחר קריאת מערכת בrip מאותן סיבות תתקבל התנהגות לא צפויה של הקוד.

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:

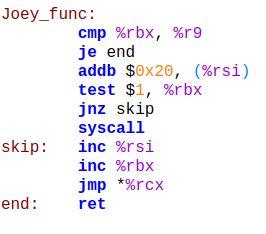
mov $Joey\_func, %rcx

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)

קריאת המערכת תדרוס את תוכן הרגיסטר rcx בrip ולכן לאחר קריאה אחת לwrite נקבל שבrcx יושם הכתובת של skip וכך בעצם ניכנס ללולאה אינסופית.

כלומר תודפס רק האות הראשונה, h.

# 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.

נבצע את הפקודות הזאות

pushfq

OR $128,(%rsp)

popfq

מה שעשינו כאן הוא להכניס את רגיסטר הדגלים כדי לשמור עליו, לבצע בפועלת או שמדליקה את הביט השביעי, ביט סימן ה0, בתוכן שעליו מצביע rsp שכרגע מבציע על שמורית הרגיסטר הדגלים שלנו. ואז נוציא את הרגיסטר מהמחסנית ונכניס אותו חזרה ךתוך רגיסטר הדגלים. בכך רק הדלקנו את ביט ה0.

נשים לב שרק שינו את רגיסטר הגדלים (בהוצאה ומבפועלת ה (OR, את rsp (כשהכנסנו והצענו מהמחסנית) ואת rip שהוא קורה את הפקודות.

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)

היא תשנה את הדגל IOPL ל 00

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)

ההיגיון מאחורי האיסור על השינוי הוא שאם יהיה לך את האישורים הגבוהים יותר אז היא תוכל לבצע פעולות שעלולות לפגוע בגרעין ובמהלך התקין של המחשב, בכך שתשנה מידע שאסור לה לגשת אליו (לגודמה מידע האחראי על אתחול מהחשב). אנו משתמשים בקריאות מערכת כי הם עטופות בקוד אשר דואג לא לבצע פעולות הרסניות.

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.