### 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.

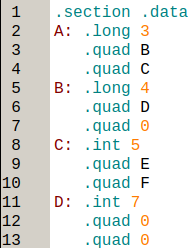
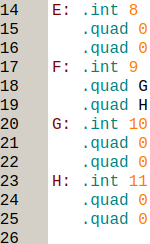
הוראות הגשה:

* 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 (45 נק') – שגרות:

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

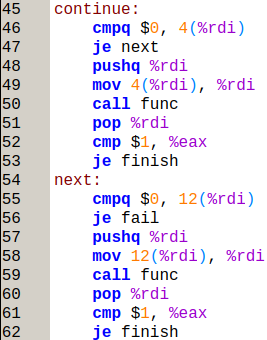
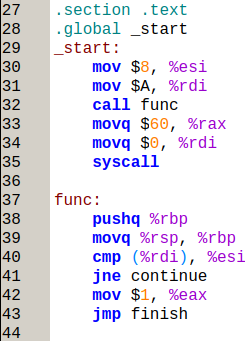
 

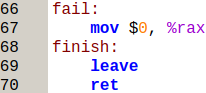
1. ציירו את הגרף המתקבל מפירוש מקטע הנתונים (מומלץ להסתכל בתרגול 3 תרגיל 1 ולהיזכר שם על אופן פירוש הזיכרון כרשימה מקושרת). בכל צומת בגרף ציינו את התווית המתאימה לו בלבד (אין צורך לציין ערכים נוספים) (3 נקודות)

A picture containing wall, rack

Description automatically generated

ג'וני לא מפחד משגרה שוחקת ולכן כותב את השגרה func וקוד שמתשמש בה (שימו לב סוף השגרה בעמוד הבא):





1. נתון שבתחילת התוכנית ערך של rsp הוא x.כאשר x הוא מספר בקסדצימלי. מה הוא הערך המקסימלי ומה הערך המינימלי שrsp יכיל לאורך ריצת התוכנית? תנו נוסחא שהמספרים בה הם בבסיס הקסדצימלי (בטאו את התשובה בהאמצעות x). (7 נקודות)

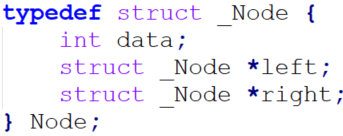
**תשובה:** מקסימלי – x, מינימלי – x-0x40.

1. רשמו מה יהיה פלט הפונקציה עבור קטע הקוד הנוכחי (5 נקודות)

**תשובה:** 1.

1. המירו את הפונקצייה לשפת c על ידי כך שתשלימו את המקומות החסרים בקוד. העיזרו בהגדרת הstruct שנתונה לכם (10 נקודות):

הstruct הנתון:



**הערה1:** שני הפרמטרים צריכים להיות תואמים לשני הפרמטרים של פונקצית האסמבלי גם מבחינת תפקיד וגם מבחינת סדר. כלומר, root צריך להתאים בתפקידו לפרמטר הראשון שמועבר לפוקנציה בשפת אסמבלי גם מבחינת הקונבנצייה שלמדנו.

**הערה2:** אורך הקו לא מלמד על אורך האיבר שצריך להשלים. **מותר להשלים יותר ממילה אחת בכל קו אך לא יותר מפקודה אחת!**

int func ( Node \*root, int x){

If (root->data == x )

return 1;

if (root->left != null)

if (func(root->left, x))

return 1;

if (root->right != null)

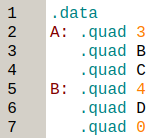
return func(root->right, x);

return 0;

}

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

1. 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 הint בquad. כלומר מקטע הנתונים ישתנה כך:



ובאופן דומה כל שאר האותיות יחליפו את הנתון הראשון במקום ב intבquad.

רשמו את השינויים שצריכים להיות בקוד על מנת שיעבוד בצורה תקינה עם מקטע הנתונים החדש (5 נקודות)

**תשובה:** בשורה 30 ו40, נשנה את esi ל- rsi,בשורה 46 ו-49 לשנות ל-8 במקום 4 את הקבוע.

בשורות 55 ו-58 לשנות את הקבוע מ-12 ל-16.

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



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

1. התוכנית תסתיים ופלט הפונקציה יהיה 1
2. התוכנית תסתיים ופלט הפונקציה יהיה b
3. התוכנית תכנס ללולאה אנסופית
4. התוכנית תקרוס במהלך ריצה
5. התוכנית כלל לא תבנה

**נימוק:** התוכנית במצב זה תגיע ל-D ולאחר מכן תמשיך ל-A שהוא כעת הבן השמאלי של D ומשם תחזור חזרה ל-"תחילת התוכנית" כאשר המחסנית ממשיכה לגדול ולכן תישאר בלולאה עד שהמחסנית תתמלא והתוכנית תקרוס.

1. 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). הסיברו **בקצרה** את תשובתכם!: (10 נקודות)
2. מחיקת הפקודת push וpop שבשורות 60 ו 57.

**תשובה:** נכונות השגרה לא תיפגע בשום מקרה ללא תלות בקלט. הסבר: בקריאה הרקורסיבית אמנם ערך הרגיסטר rdi לא יתעדכן לחוליה בה הוא אמור להיות לפי קריאת הרקורסיה, אבל לא יהיה בו שימוש בהמשך ריצת התוכנית או עד שהשימוש הבא בו יהיה לאחר חזרה מקריאה רקורסיבית מבן שמאלי של צומת מסוימת ולאחריה כן יתעדכן ערך הרגיסטר לערך המצופה.

1. מחיקת הפקודה pop בשורה 60

**תשובה:** במקרה זה נכונות השגרה לא תיפגע. באופן דומה לכך שבסעיף א' לא נפגעה נכונות השגרה (אין שימוש ב-rdi אחרי החזרה) לא תפגע השגרה גם כאן. במחסנית יישארו ערכי ה-rdi בהם עברנו (הצמתים בהם עברה הרקורסיה) אך בעת ביצוע ה-leave נחזיר את rbp ו-rip לערכים המתאימים להם והתוכית תעבוד.

1. מחיקת push ו-pop שבשורות 48 51

**תשובה:** נכונות השגרה תיפגע, בקריאה לבן השמאלי אנחנו "מאבדים" את הדרך לחזור לאבא ובפרט גם לתת העץ הימני של האב, מכיוון שאנחנו מסיירים על העץ בpreorder , ואנחנו תמיד מתחילים בהליכה לבן השמאלי, אז כשנגיע לעלה השמאלי ביותר בעץ, "איבדנו" את כל הצמתים בעץ שאינם במסלול שחיפשנו או בתת העץ הימני של עלה זה (במידה וקיים) ולכן ישנם מקרים בהם לא נמצא את הערך למרות שהוא קיים בעץ.

1. הוספת פקודה push %rdi אחרי continue בשורה 45

**תשובה:** נכונות השגרה לא תיפגע, תקרא שגיאה דומה לזו המתוארת בסעיף ב' אך הקוד יעבוד כמצופה.

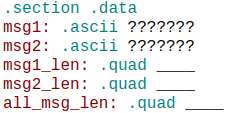
1. הוספת הפוקדה push %rdi אחרי continue בשורה 45, שינוי פוקדת הpop שבשורה 51 לפקודה: mov (%rsp), %rdi ומחיקת הפקודות push וpop בשורות 60 57.

**תשובה:** נכונות השגרה לא תיפגע, תקרא שגיאה דומה לזו המתוארת בסעיף ב', אך הפעם ישמרו שני עותקים של rdi במחסנית אחד אחרי השני ואף אחד מהם לא יצא מהמחסנית אבל בעת הפקודה mov (%rsp), %rdi נכניס ל-rdi את הערך המצופה בעת המשך הקריאה הרקורסיבית ולכן התוכנית תעבוד כרגיל.

# 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:



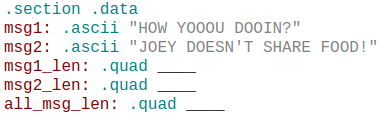
ג'ואי לא יודע עדיין אילו מחרוזות הוא יכתוב. עליכם להשלים את המקומות הריקים שקשורים לאורכי המחרוזות כך שמשתנה msg1\_len יהיה האורך של msg1, בmsg2\_len יהיה האורך של msg2 ובמשתנה all\_msg\_len יהיה שווה לסכום אורכי המחרוזות msg1 וmsg2. שימו לב עליכם לעשות זאת בצורה כזו שהאורכים יהיו נכונים בעת ריצת התוכנית ללא קשר לאיזה מחרוזות ג'ואי ישים בmsg1 ובmsg2. (3 נקודות)

**תשובה:** msg1\_len = msg2-msg1

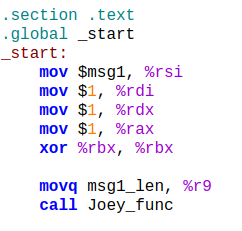
msg2\_len = msg1\_len-msg2

, all\_msg\_len = msg1\_len – msg1

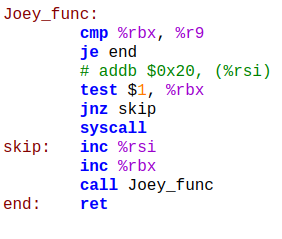
1. כעת נתון מקטע הנתונים שכולל את המחרוזות:



לפניכם נתונה התוכנית שג'ואי כתב:



ומוצגת כאן גם הפונקציה שכתב:



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

**תשובה:** הפונקציה תדפיס את האותיות במקומות האי-זוגיים של msg1 – ”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 ד - ו השינויים בסעיף ג עדיין תקפים).**

מה יודפס כעת בסיום ריצת הקוד? (5 נקודות)

**תשובה:** כעת בסיום ריצת הקוד יודפסו האותיות במקומות האי-זוגיים של שתי המחרוזות ברצף, אבל באותיות קטנות במקום גדולות ועבור תווים שאינם אותיות נקבל את התו שנמצא 32 תווים אחריו בטבלה של ערכי ascii.

כלומר: 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? (2 נקודות)

**תשובה:** כעת הפלט יהיה 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)

צ'נדלר מציע להחליף את השימוש בr9 בשימוש בrcx.

**תשובה:** לא (מנומק מטה)

מוניקה מציעה להחליף את השימוש בr9 בשימוש בr11.

**תשובה:** לא , צ'נדלר ומוניקה לא זכרו שבעת קריאת מערכת rcx ו-r11 נדרסים לטובת רגיסטרים rflags וrip- ולכן יש לשמור אותם ולשחזרם לפני ובחזרה מקריאת המערכת syscall.

פיבי מציעה להחליף את השימוש בr9 בשימוש בrdi.

**תשובה:** לא, מכיוון שבקריאת המערכת יש שימוש בrdi בשביל לדעת לאן לכתוב את התווים.

רייצ'ל מציעה להחליף את השימוש בr9 בשימוש בr12.

**תשובה:** כן, למרות שרייצ'ל לא זוכרת קונבנציות קריאה לפונק' , זה יעבוד כי אין שימוש בr12 בתוכנית, מלבד למטרה זו.

רוס מציע להחליף את השימוש בr9 בשימוש בrbp.

**תשובה:** רוס בחור חכם ואנחנו סומכים עליו, בכל מקרה בגלל שאין שימוש בrbp בתוכנית ניתן להשתמש בו והתוכנית תרוץ כמצופה.

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:

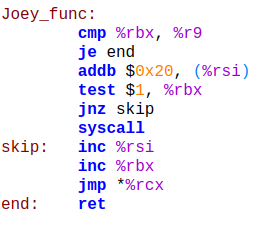
mov $Joey\_func, %rcx

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 נראת כך כעת:



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

**תשובה:** לאחר השינוי יודפס התו h.

לאחר הקריאה לsyscall בפעם הראשונה, rcx יקבל את rip , שמצביע על ה-skip , ולכן מעכשיו rcx הולך להצביע על skip וכל פעם אנחנו פשוט נקפוץ לskip במקום לקפוץ לJoey\_func . ולכן אנחנו נכנס ללולאה אינסופית של skip.

# 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 OF וCF. 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.

**תשובה:** נעשה pushfq, נבצע or $2049,(%rsp) מה שישנה בדיוק את הדגלים OF, CF – במידה ואינם דלוקים , ואז נעשה popfq. ובכך נמנע מהשינוי הפוטנציאלי שהפעולה or תבצע לאוגר הדגלים במידה ולא נעשה pushfq ו-popfq.

1. 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**

**תשובה:** הולי צריכה לשנות את הדגל IOPL ובכך תאפשר לעצמה גישה ישירה להתקני הקלט והפלט.

1. הולי מחליטה לנסות את התעלול מסעיף א' רק שבמקום לשנות את CF וOF היא רוצה לשנות את IOPL. להפתעתה, היא לא מצליחה לשנות את הביטים הללו. הסבירו מה ההגיון בכך שהיא לא מצליחה לשנות את IOPL? התייחסולצורך בקריאות מערכת (4 נקודת) ולכן

**תשובה:** ההיגיון בכך הוא שיש סיבה לכך שצריך להשתמש בקריאות מערכת בשביל לגשת להתקני הקלט והפלט – מערכת ההפעלה לא רוצה לאפשר גישה ישירה למשתמשים להתקנים אלה ולכן כל פקודה שתנסה לשנות את דגלים אלה כאשר CPL הוא 3 תהיה בלתי חוקית.

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

1. 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. (5 נקודות)
2. כיבוי דגל IF באוגר הדגלים
3. הדלקת דגל IF באוגר הדגלים
4. שינוי CPL ל00
5. לא ניתן לחסום פסיקות תוכנה.
6. 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? (5 נקודות)
7. כיבוי דגל IF באוגר הדגלים
8. הדלקת דגל IF באוגר הדגלים
9. עליו לחבר את הפסיקות לכניסת NMI ואז לכבות את דגל IF
10. לא ניתן לחסום פסיקות חומרה ולכן לא יצליח.