Assignment 1 Responses

Operating Systems

Semester 1 5784

Submitting Students:

<Nehoray> <Cohen>

Date: <14.1 >

--------------------------------------------------------------------------

Section 1

(gdb) 1. File tee

(gdb) 2.b main

(gdb) 3.r

(gdb) 4.bt

(gdb) 5.step 9

(gdb) 6.q

(1)0X555555554040

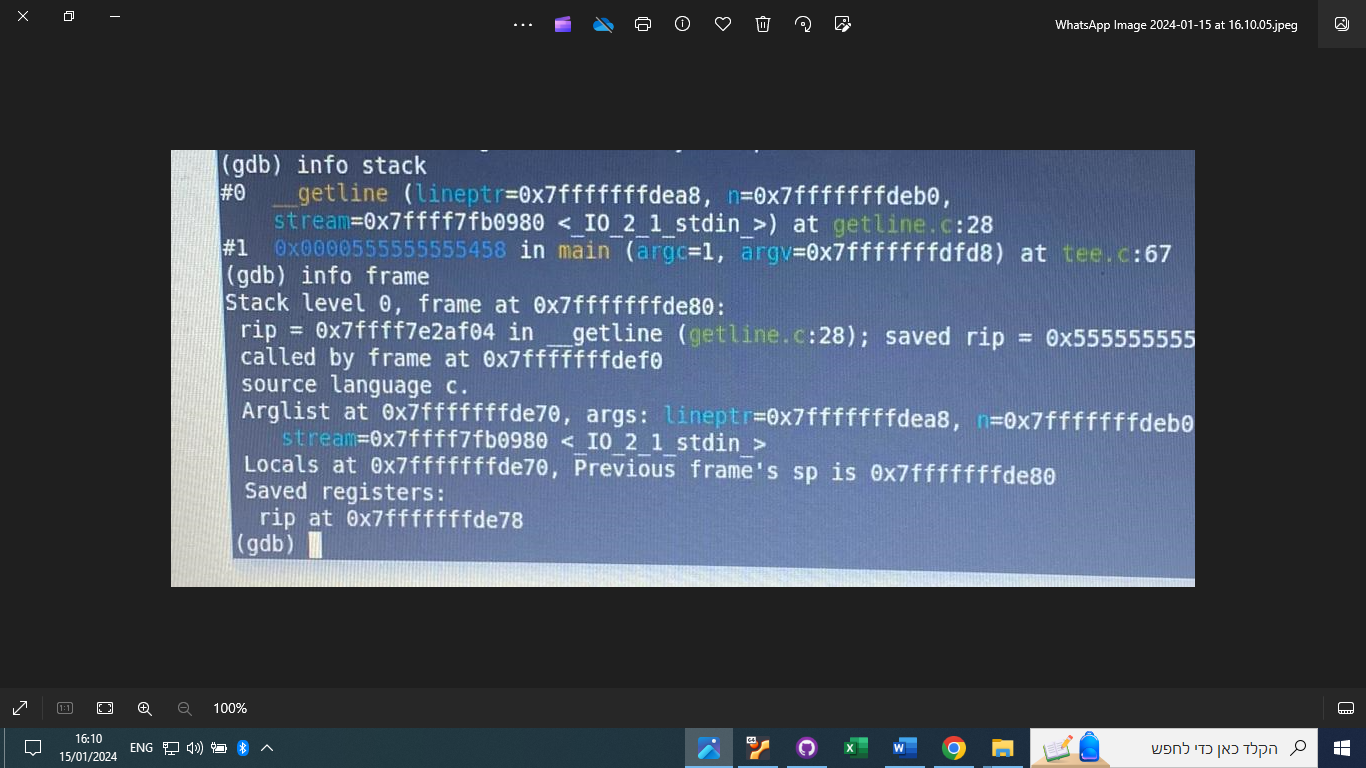
(2)0X400000006

(3)csu/libc-start.c:

(4)info stack מציג את מצב המחסנית

* את כתובת הזיכרון של הפונקציה
* את השורה בקוד שבה הפונקציה נעצרה
* את הארגומנטים שנמסרו לפונקציה
* את הערכים של המשתנים המקומיים של הפונקציה

5) info frame מספקת יותר מידע על מצב המחסנית העדכני   
שם, כתובת זיכרון ארגומנטים שהתקבלו כתובת החזרה



rax: מכיל את התוצאה של חישוב או את הערך המוחזרת של פונקציה.

rbx: מכיל מצביע לנתונים או לקוד.

rsi: מכיל מצביע למשתנים בקריאה לפונקציה.

rdi: מכיל מצביע לפרמטרים בקריאה לפונקציה.

rbp: מכיל מצביע למסגרת הפונקציה הנוכחית.

rsp: מכיל מצביע לראש הערימה.

rip: מכיל את המיקום של ההוראה הבאה לביצוע.

cs: מכיל את המצביע ל-segment code.

ss: מכיל את המצביע ל-segment stack.

ds: מכיל את המצביע ל-segment data.

--------------------------------------------------------------------------

Section 2

(mp) 1.בקובץ recur.c נצטרך להוסיף & שידפיס את הכתובות ולא את הערכים

(a) &localChar

(b) &localInt

(c) &localFloat  
&localArr

(mp) 2.בקובץ map.c

נדרש לצרף אליו את קובץ הrecur.c כדי שירוץ

(a)גם נצטרך להוסיף & שידפיס כתובות בשורות 26,29,30,31

(b)

(c)

(d)

2.1

-זהו אזור שמכיל נתונים קבועים שאינם ניתנים לשינוי, כמו מחרוזות או מספרים. האזור הזה מוגן מפני כתיבה, והוא משמש לשמירת נתונים שאינם צריכים להשתנות במהלך הריצה. a .rodata

אזור שמכיל את הקוד הבינארי של התוכנית, כלומר את ההוראות שמבצע המעבד. המקום גם מוגן מפני כתיבה, והוא משמש להרצת התוכנית .text (b)

data (c) מכיל נתונים מוגדרים מראש שניתנים לשינוי, כמו משתנים גלובליים או סטטיים. האזור הזה אינו מוגן מפני כתיבה, והוא משמש לאחסון נתונים שצריכים להשתנות

(d)

(e)

(mp) 6.

(mp) 7.

(mp) 8.

(mp) 9.

2.2

(a) g text data

(b) O data com

(c) F text

(d)

\*UND\*

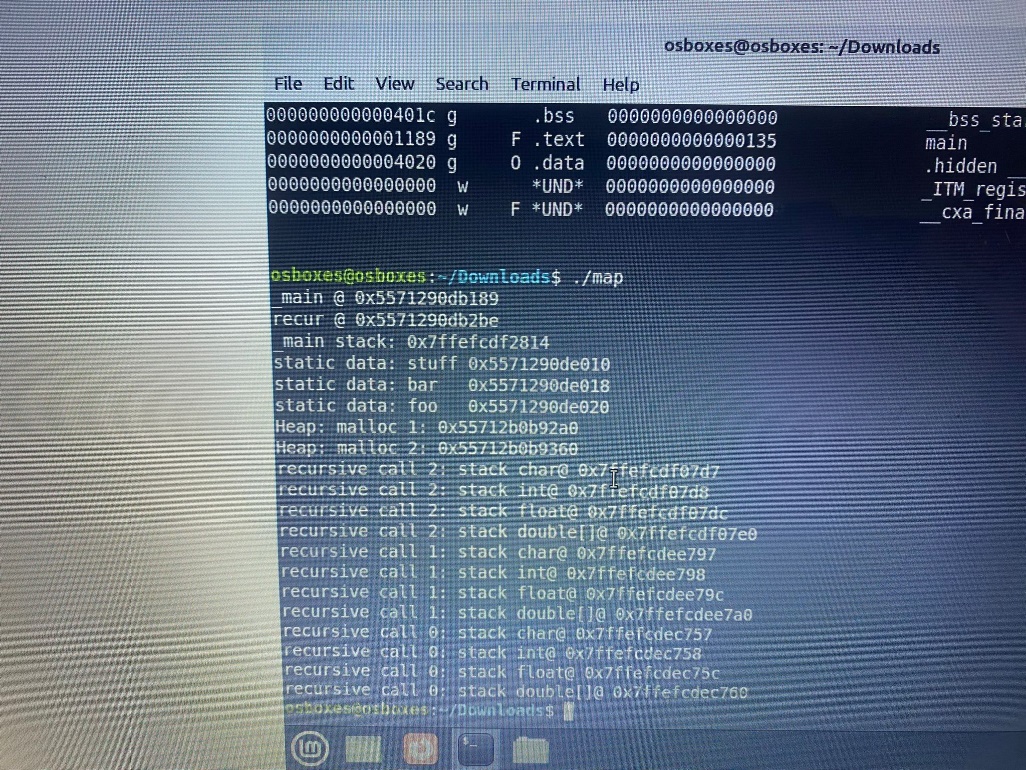
2.3 עכשיו יש הרבה שינויים   
F init text und\*

הg data text bss

ה O bss data got fini array

Und\* F

2.3ב  
הסמלים printf@@GLIBC 2.2.5 ו-malloc@@GLIBC 2.2.5 נשארים \*UND\* בטבלת הסמלים של קובץ ההפעלה משום שהם שייכים לספריית זמן הריצה של C (libc), והם לא נמצאים בקובץ ההפעלה עצמו

2.4. 

2.5 **) main ערך1189 מוגדר בtext כתובת 0135**

**(ב) recur ערך 14b כתובת 12be מוגדר בtext**

**(ג) stuffערך 0008 כתובת 4010 מוגדר בdata**

**(ד) foo (ערך 0004 כתובת 4020 מוגדר ב bss**

**ה) bar**ערך 0004 כתובת 4018 מוגדר בDATA

2.6

2.9 הם קרובים אחד לשני הראשון בכתובת 1080 השני בכתובת 1090 לכן נגיד שהם צמודים

--------------------------------------------------------------------------

Section 3

(objd) 1.אריכטקטטורה i386:x86-64   
פורמט elf64- x86-64

(objd) 2.

section.plt  
section.text

Section.init  
symbol table

Section.plt.got

Section.plt.sec

Section.fini

(objd) 3 כתובת main 0000000000001249  
section.text . נמצאת תחת

(objd) 4.כן בפונקציית ה main רואין פעולות על מחסנית כמו push %rbp   
ורואים את המחסנית מתחת לתווית section

--------------------------------------------------------------------------

Section 5

(tl) 1. עשיתי את זה בוירטואל בוקס אולי טעיתי בקוד stack

(tl) 2. ציפיתי שיהיו נתונים פחוות טובים הופתעתי לטובה אם במידה ונגיע למגבלה בתהליך המקסימלי המחשב יקרוס

3

(tl) 3.  
