# שאלה 1 – קידוד פקודות

לאחר מתקפת הסייבר הכבדה על הטכניון הבינו מומחי המחשבים בבניין טאוב שנוצרה בעיה.

1. בגלל המתקפה האחרונה, כל האסמבלרים בפקולטה הפסיקו לתרגם פקודות לשפה מכונה. עזרו לסגל קומפילציה לתקן את הנזק ע"י תרגום הפקודות הבאות בצורה תקינה מאסמלי (AT&T syntax) לשפת מכונה.

הערה: יש למלא את הערכים בhexadecimal.

<start>:

400000: **0x334DE4**  xor %r12, %r12

400003: **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**  shr $2, %r8

400007: **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**  sub $5, %ecx

40000A: **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**  lea 11(%rip) , %r8

400011: **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**  jmp \*0x1234(%rip)

1. מה יהיה ערכו של רגיסטר r8 בעת ההגעת הקוד לכתובת 0x400011? : \_\_\_\_\_\_\_\_\_\_\_\_\_\_
2. סגל קומפי הצליח לתקן את כל האסמבלרים בעזרתכם אבל כעת התגלתה בעיה אחרת. המעבדים בפקולטה לא מצליחים לעשות decode לפקודות. תרגמו את הרצף הבינארי הבא מפקודות מכונה לפקודות אסמבלי.

55 48 89 e5 48 83 ec 08

הרצף הנ"ל נתון בהקסא, משמאל לימין (הבית הראשון ברצף הוא 0x55). את רצף הפקודות שמקודד עליכים לכתוב בשורות הבאות:

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

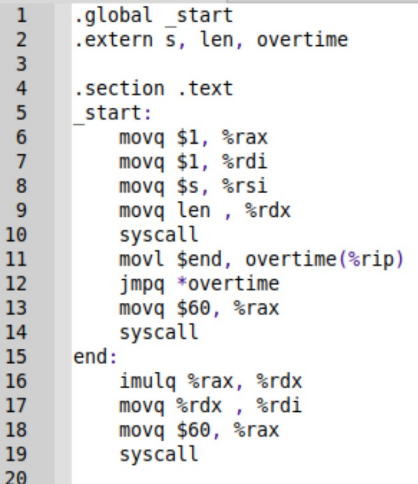
**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

הערות: כל פקודה חייבת להופיע בשורה נפרדת. ניתן להשאיר שורות ריקות.

# שאלה 2 – קבצי ELF וקישור סטטי

A screen shot of a computer program

Description automatically generated with low confidenceלרגל המונדיאליטו חברכם גיא החילט לכתוב תוכנית באסמבלי המתפרשת על שני קבצים.

להלן תוכן הקבצים: 

U20worldCup1.asm

U20worldCup2.asm

גיא התלהב מהקוד שכתב והריץ בטרמינל את הפקודות הבאות:

as U20worldCup1.asm -o U20worldCup1.o

as U20worldCup2.asm -o U20worldCup2.o

ld U20worldCup1.o U20worldCup2.o -o U20worldCup.out

./U20worldCup.out

גיא טס לצפות במשחקים בארגנטינה ושם הוא דיבר עם אוהדים מכל העולם. התברר לגיא שאף אחד מהם לא יודע איך טבלאות הסמלים של שני הקבצים יראו.

1. עזרו לאוהדי העולם ומלאו את טבלאות הסמלים של U20worldCup1.o ושל U20worldCup2.o.

הערות:

1. ניתן להשאיר שורות ריקות
2. בעמודה Nxt עליכם לכתוב את שם ה section או UND (ולא מספר).

U20worldCup1.o symbol table:

|  |  |  |
| --- | --- | --- |
| (section) Nxt | Bind(נראות) | name |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

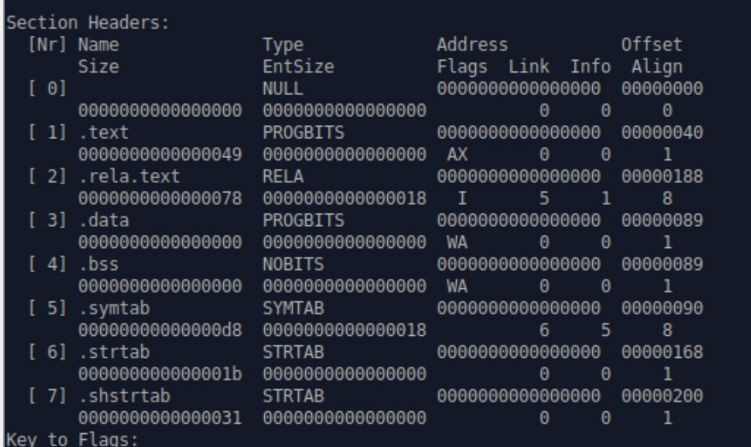
**השאלה ממשיכה בעמוד הבא**

U20worldCup2.o symbol table:

|  |  |  |
| --- | --- | --- |
| (section) Nxt | Bind(נראות) | name |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

גיא החליט להתחפש כדי שאף אחד לא יזהה אותו ולכן גם חבריו של גיא לא מזהים אותו. בשביל לדעת באמת מי זה גיא אותם חברים הראו לו את טבלת הsection header של הקובץ U20worldCup1.o שנוצרה ע"י הרצת הפקודה : readelf -S U20worldCup1.o. ואת התוכן של הקובץ U20worldCup1.o ע"י הפקודה hexdump. להלן התוצאות:

Readelf -S U20worldCup1.o:



1. אותם חברים רצו שגיא יסמן

**Hexdump U20worldCup1.o:**

0000000 457f 464c 0102 0001 0000 0000 0000 0000

0000010 0001 003e 0001 0000 0000 0000 0000 0000

0000020 0000 0000 0000 0000 0238 0000 0000 0000

0000030 0000 0000 0040 0000 0000 0040 0008 0007

0000040 c748 01c0 0000 4800 c7c7 0001 0000 c748

0000050 00c6 0000 4800 148b 0025 0000 0f00 c705

0000060 0005 0000 0000 0000 ff00 2524 0000 0000

0000070 c748 3cc0 0000 0f00 4805 af0f 48d0 d789

0000080 c748 3cc0 0000 0f00 0005 0000 0000 0000

0000090 0000 0000 0000 0000 0000 0000 0000 0000

00000a0 0000 0000 0000 0000 0000 0000 0003 0001

00000b0 0000 0000 0000 0000 0000 0000 0000 0000

00000c0 0000 0000 0003 0003 0000 0000 0000 0000

00000d0 0000 0000 0000 0000 0000 0000 0003 0004

00000e0 0000 0000 0000 0000 0000 0000 0000 0000

בHexdump את מקטע הtext

בשביל להוכיח שהוא הגיא האמיתי.

עזרו לגיא וסמנו את מקטע הtext

בhexdump הבא:

**השאלה ממשיכה בעמוד הבא**

לצורך הסעיף הבא נתון פלט הobjdump של U20worldCup1.o:

0000000000000000 <\_start>:

0: 48 c7 c0 01 00 00 00 mov $0x1,%rax

7: 48 c7 c7 01 00 00 00 mov $0x1,%rdi

e: 48 c7 c6 00 00 00 00 mov $0x0,%rsi

15: 48 8b 14 25 00 00 00 mov 0x0,%rdx

1c: 00

1d: 0f 05 syscall

1f: c7 05 00 00 00 00 00 movl $0x0,0x0(%rip) # 29 <\_start+0x29>

26: 00 00 00

29: ff 24 25 00 00 00 00 jmpq \*0x0

30: 48 c7 c0 3c 00 00 00 mov $0x3c,%rax

37: 0f 05 syscall

0000000000000039 <end>:

39: 48 0f af d0 imul %rax,%rdx

3d: 48 89 d7 mov %rdx,%rdi

40: 48 c7 c0 3c 00 00 00 mov $0x3c,%rax

47: 0f 05 syscall

1. מלאו את הטבלה הבאה של הrelocation של הtext section:

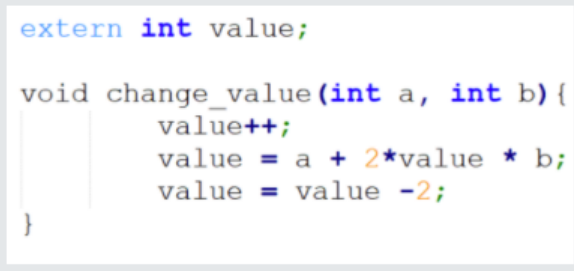
|  |  |  |  |
| --- | --- | --- | --- |
| addend | Symbol name | type | offset |
|  |  |  | 0x11 |
|  |  | קבוע |  |
|  |  |  | 0x21 |
|  | .text |  |  |
|  |  |  | 0x2c |

**הערה**: ב"Type "ניתן להשלים רק "יחסי" או "קבוע" ואין צורך להשתמש בשמות המלאים.

1. האם בניית התוכנית תצליח? (יווצר קובץ הרצה תקין?) הקיפו את התשובה הנכונה. כן / לא
2. בהמשך לסעיף הקודם, אם עניתם לא הסבירו מדוע. אם כן רשמו מה יהיה פלט התוכנית ומה ערך היציאה שלה.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

# שאלה 3 – קישור דינמי

1. לפניכם קוד של ספריה דינאמית שקומפלה:

כמה תיקונים יצטרך לעשות הקשר הדינאמי עבור הסמל value? הסבירו את איפה יתבצעו התיקונים.

1. נתון לכם PLT של תוכנה מסוימת.

Calendar

Description automatically generated with medium confidence

נתמקד בפקודה בכתובת 0x1030.

1. מה סוג הקפיצה שבו משתמשים?
2. מהו סוג האופרנד (אם מדובר בכתובת, ציינו שיטת מיעון)?
3. האם ידוע לאיזה כתובת נקפוץ באת ביצוע הפקודה? אם כן מהי הכתובת ואם לא מדוע לא ניתן לדעת ומה כן ניתן לדעת על אותה כתובת.

**השאלה ממשיכה בעמוד הבא**

1. הסבירו מה תכיל הכתובת 0x4018 בתחילת ריצת התוכנית. התייחסו למקרה שבו התוכנית קומפלה עם lazy binding ולמקרה שבו היא לא.
2. הסבירו מתי נרצה לקמפל עם lazy binding ומתי לא נרצה.