

## מטלה 7 – ארכיטקטורה

למען הסר ספק, עברנו לחלק של MIPS. מי שיגיש קבצי asm של Hack לא יקבל ניקוד כלל. כווצו את כל קבצי האסמבלי שלכם לקובץ zip יחיד ששמו כתעודת הזהות. כמו כן, צרפו לאותו zip קובץ word/txt עבור שאלת התרגום.

ההגשה עד ה 10.06.2020 בשעה 23:59

כל סעיף הוא קובץ asm. חדש. הסעיפים יקראו 1.asm, 2.asm, ... 6.asm. סעיף 7 הינו קובץ word/txt בלבד.

שימו לב: בשאלות הקשורות לזיכרון, אין צורך לאתחל ערכים בזיכרון בשביל לכתוב את הקוד, ולכן אתחול ישמש לכם לבדיקה בלבד. את הקודים תגישו ללא אתחול. **ירדו נקודות על קודים שיש בהם אתחול ערכים.**

יש לכתוב הערות בקוד, במידה סבירה (לכל הפחות, כל שורה שלישית). קוד ללא הערות שלא יעבוד תקין לא יקבל ניקוד כלל.

(1) (שאלת חימום) הכניסו את תעודת הזהות שלכם לאוגר \$s6 (אל תמירו מעשרוני, תכניסו ישירות כהקסה!). את ספרת הביקורת שימו באוגר \$t2.

(2) בנו קובץ המבצע את החישוב הבא:

$$\sum_{i=1}^{i=10} A_i B_i C_i D_i$$

(סכום המכפלה המרובעת של  $A_i B_i C_i D_i$ , כאשר  $i$  הוא האינדקס במערכים)

כאשר

$$A=[1,3,5,...,19]$$

$$B=[1,3,6,10,15,21,28,36,45,55]$$

$$C=[1,4,16...2^{18}=262144]$$

$$D=[1,-1,1,-1,1,-1,1,-1,1,-1]$$

את התוצאה שימו באוגר \$s0. אם יש צורך ביותר מ 32 ביטים (נראה לי שלא, אבל תבדקו!) שימי את הביטים החזקים ב \$s1 ואת החלשים ב \$s0. יש להתעלם מ overflow של הסכימה.

(3) א. בנו קובץ הקורא את המערך A הרשום בזיכרון החל מכתובת 0x10020000 ועד כתובת 0x10020040 (לא כולל. כלומר, sixteen words), ומחשב את סכום כל ריבועי הערכים של המערך. הסכום יישמר באוגרים \$s1:\$s2 כך שהביטים הגבוהים של הסכום יהיו בתוך \$s1, והביטים הנמוכים של הסכום בתוך \$s2. יש להתעלם מ overflow של הסכימה.

(4) בנו קובץ הקורא את המערך A הרשום בזיכרון החל מכתובת 0x10020000 ועד כתובת 0x10020020 (לא כולל. כלומר, eight words), על כל מספר מחשב את החזקה השלישית שלו, וסופר את כמות התוצאות בהן היה צורך להשתמש באוגר hi. שימו לב- התוצאה הסופית היא רק "כמה חישובים השתמשו ב hi", ולא תוצאות החזקות עצמן. את התוצאה שימו באוגר \$s1.

<sup>1</sup> בואו נכיר במציאות: הראשון מהמחזור יבדוק, ויעדכן את כולם (כולל אותי). **אכן בדקו ואין צורך באוגר \$s1.**

(5) כתבו פונקציה רקורסיבית המממשת את הפסודו קוד הבא:

```
Int atzeret(a)
    If a==1 return a;
    Else return a*atzeret(a-1)
    (הניחו שאין שימוש ב hi, אלא כל התוצאה חסומה ב 32 ביטים)
    את התוצאה הסופית שימו באוגר $s0.
```

(6) תרגמו את הקוד הבא לבינארית, היעזרו בנתונים הבאים:

**שימו לב שצריך להגיש או בינארי או בהקסה!! מי שיגיש את שניהם מאבד ניקוד!**

```
Func(add)=32
Func(sub)=34
Func(and)=36
Func(sll)=0
Opcode(addi)=8
Opcode(bne)=5
Opcode(j)=2
Opcode(jal)=3
```

```
0x00400000: addi $s0, $0, 0x1D34
0x00400004: add $s1, $0, $s0
0x00400008: addi $s2, $0, 0x00A2
0x0040000C: bne $s0, $s1, target
0x00400010: sub $s1, $s1, $s2
0x00400014: j end
0x00400018: target: and $s1, $s1, $s2
0x0040001C: jal func1
0x00400020: j end
0x00400024: func1: sll $s1, $s1, 2
0x00400028: j end
0x0040002C: end: sll $s2, $s1, 1
```

טטים למטלה:

1. תראו את תעודת הזהות שלכם רשומה באוגרים.
2. חישוב
3. אתחלו ערכים אקראיים, וכיתבו קוד java שמבצע אותו דבר רק בלי הקריאה מהזיכרון, עם הערכים שהכנסתם. התוצאה אמורה להיות זהה (שימו לב להבדל בין הקסה לעשרוני)
4. אתחלו ערכים שאכן העלאתם בשלישית תיעזר באוגר high כדי לבדוק. זכרו למחוק את האתחול.
5. אתחלו ערך עבור המשתנה a. שימו לב שהקוד הסופי שאתם מגישים הינו הפונקציה בלבד, ואילו באתחול ובדיקה תצטרכו לכתוב גם main שייקרא לפונקציה הרקורסיבית.
6. בדקו ב MARS אם התרגום שלכם נכון.