

## מטלה 7 – ארכיטקטורת MIPS

הנחיות למטלה:

- הגשת המטלה בזוגות.
  - שם המטלה יהיה מספרי הזהות של שני המגישים, יחד עם \_ ביניהם.
  - המטלה תוגש כקובץ zip/rar.
  - אין הגשות באיחור. תיבת ההגשה נסגרת אוטומטית בתאריך שהוגדר לה, ולא נקבל הגשות במייל.
  - למרות האמור לעיל, נגדיר עוד פרק זמן כלשהו לקבלת הגשה באיחור בתיבה, אך לא נגיד לכם כמה זמן. אל תבנו על זה, אלא הקדימו הגשתכם.
  - לאחר ההגשה, בצעו download על הקובץ שהגשתם, כדי לוודא שאכן הגשתם את הקבצים הנכונים והכל תקין.
  - לכל שאלה במטלה יש לפתוח קובץ חדש, ששמו כשם השאלה.
  - שאלות 1-6 יקבלו את הקבצים 6.asm, ..., 1.asm, שאלות 7,8 יקבלו את הקבצים 7.txt, 8.txt.
  - בקבצים 7,8 יש לכתוב רק את התרגום הנדרש, ללא שום דרך, שאלות, קוד מקורי, תהליכי ביניים ושאר ירקות. מי שמעוניין יכול להגיש קבצים נוספים בשם 7\_explanation.txt, 8\_explanation.txt, אך סביר להניח שבבדיקת המטלה הם אפילו לא ייפתחו.
  - כזכור – העתקה במטלה תקנה 0 בציון **הקורס** לכל ארבעת הסטודנטים הרלוונטיים. עדיף שתגישו חלקית, או שלא תגישו בכלל, מאשר שתעתיקו במטלה.
  - למען הסר ספק – זוהי מטלת MIPS. מי שיגיש פתרון באסמבלי של HACK לא יקבל ניקוד כלל.
- מטלה זו מקנה עד 3 נקודות בonus, למי שעובר את המבחן.

1. הכניסו את מספרי הזהות שלכם לאוגרים לפי ההנחיות הבאות:
- שמונה הספרות השמאליות (ללא ספרת הביקורת ה LSB) של מספר הזהות המינימלי מבין שני השותפים ייכנסו לאוגר \$s2.
  - שמונה הספרות השמאליות של מספר הזהות המקסימלי מבין שני השותפים ייכנסו לאוגר \$s6.
  - ספרת הביקורת המינימלית תהיה הספרה השנייה מימין באוגר \$t7.
  - ספרת הביקורת המקסימלית תהיה הספרה השנייה משמאל באוגר \$t7.
  - (ההנחיה תקפה גם אם יש שיוויון בין שתי ספרות הביקורת).
  - אם שתי ספרות הביקורת הן אפס, יש להכניס לאוגר \$t7 את המספר 0x10203040.

יש להכניס את המספר, כאילו תעודת הזהות כתובה כבר בהקסה!  
לדוגמה: אם מספר הזהות (העשרוני) המינימלי הוא 123456789 אזי יתקבל  
 $s2 = 0x12345678$

2. יהיו ה-10-יות הסדורות הבאות:
- $$A = (1, 4, 16, \dots, 4^9), \quad B = (-1, 1, -1, \dots, 1), \quad C = (2^9, 2^8, \dots, 2^0)$$
- כתבו קטע קוד המחשב את הסכום הבא:

$$\sum_{i=0}^9 A_i \cdot B_i \cdot C_i$$

את התוצאה בסוף הקוד יש לשמור באוגר \$s5.  
הערה: אין להכניס את ה-10-יות הסדורות כמערכים בזיכרון, אלא יש להשתמש באוגר לכל 10-יה ולהתקדם בין הערכים בתוך הלולאה.

3. כתבו קטע קוד המכניס את הערכים 1,3,5,7, ..., 29, לזיכרון, החל מהכתובת 0x10011020 והלאה (בקפיות של ארבעה bytes). חובה להשתמש בלולאה.

4. כתבו קטע קוד המקבל base address הרשומה באוגר \$s1, ואת הפרמטר "גודל המערך" הרשום באוגר \$s2. הקוד צריך להעלות כל ערך בריבוע, והפלט הוא כמות הפעמים בה ההעלאה בריבוע חרגה מ-32 ביטים. את הכמות הזו יש לרשום באוגר \$t0.

5. כתבו קוד רקורסיבי המחשב את סדרת פיבונאצ'י. פרמטר הקלט לפונקציה יהיה האוגר \$a1. הפונקציה תיקרא fib. (היזכרו כי  $fib(0) = 0, fib(1) = 1$ ).

6. המירו את הקוד הבא לאסמבלי של MIPS:

```
Public static Int StarngSum(i,addr){
int sum=0;
For (int j=0; j<=i; j+=3)
{
    Arr[j]*=3;
    sum+=Arr[j];
    j--;
}
multi(sum);
return sum;
}

public static int multi(s){
return (s*12);
}
```

יש להקפיד על שימוש נכון באוגרי הפונקציה, ועל שימוש תקין במחסנית. רשמו בהערה בתחילת הקוד את ההנחות שלכם (באיזה אוגר כתוב  $i$  ובאיזה כתוב  $addr$ ). למען הסר ספק:  $12s < 32 \text{ bits}$ .

7. המירו את הקוד הבא להקסה-דצימאלי. (למען עיקרון הסדר הטוב – הקוד חסר משמעות לחלוטין, ואין לי מושג מה הוא עושה. זה גם לא רלוונטי – התפקיד כרגע הוא להיות אסמבלר) נתונים נוספים:

```
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
0x00400030: sw $s0, 100($t3)
```

היעזרו בנתונים הבאים:

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

$func $t$ (add) = 32,$      $func $t$ (sub) = 34,$      $opcode(beq) = 4,$   
 $opcode(bne) = 5,$      $opcode(j) = 2,$      $opcode(addi) = 8,$   
 $opcode(sw) = 43,$      $opcode(lw) = 35,$      $func $t$ (slt) = 42,$   
 $opcode(addi) = 8,$      $opcode(jal) = 3,$      $func $t$ (sll) = 0,$   
 $func $t$ (and) = 36$

8. נתון הקוד הבא בפורמט הקסה-דצימאלי. תרגמו אותו לקוד *assembly*.

0x20101234  
0x00108400  
0x02058020  
0x0153482A  
0x11200003  
0x01378804  
0x0327A825  
0x08100003  
0x01A69022

נתונים נוספים:

$func $t$ (sll) = 0$  ובפקודה זו האוגר  $rs$  מאופס (הראשון הוא  $rd$  והשני הוא  $rt$ , ואילו  $rs$  לא קיים, כלומר הערך 0).

$func $t$ (sllv) = 4$  ובפקודה זו הסדר הינו  $rd, rt, rs$ .

$func $t$ (and) = 36,$      $func $t$ (or) = 37,$      $func $t$ (xor) = 38,$      $func $t$ (nor) = 39$

טסטים למטלה

0. הטסטים האלו הם בשבילכם. אל תגישו עם הטסטים!!!
1. תראו את מספרי הזהות שלכם במקום הרלוונטי בתוכנה.
2. זה חישוב ערך שאמור לצאת אחד לכולם. תשאלו חברים אם גם הם הגיעו לאותו ערך.
3. הריצו את הקוד וראו שהערכים אכן נכנסו למקומות הנכונים.
4. תייצרו מערך מתאים, עם כתובת התחלתית ואורך מאותחלים כראוי, ותבדקו שאכן קורה מה שאמור לקרות. רמז: תצטרכו לאתחל ערכים המכילים יותר מ-16 ביטים בשביל שזה יהיה רלוונטי.
5. כתבו *main* והריצו את הפונקציה הרקורסיבית. הרי אתם יודעים מה הערכים האמיתיים של פיבונאצ'י כי חפרנו לכם עליהם בכמה קורסים.
6. אתחלו מערך, עם כתובת במשתנה הנכון, ובדקו שהקוד תקין.
7. כתבו את הקוד ב-MARS ובדקו האם התרגום שלכם זהה לשלו.
8. כתבו את הקוד שיצא לכם לאחר התרגום ב-MARS, ובדקו האם התרגום שה-MARS מספק אכן זהה להקסה שאיתו התחלתם.

בהצלחה!