



מבנה מחשבים – מיני-פרויקט 1

תאריך הגשה: יום חמישי ה-29 בפברואר בשעה 23:55

שאלות מעשיות: כתיבת והרצת קוד אסמבלי ל-MIPS

עבור שתי השאלות הבאות צורף לתרגיל שלד קוד אסמבלי ל-MIPS בו חסר מימוש לחלק מהפונקציות.

בשני המקרים יש להוסיף את תוכן הפונקציות לקבצים הקיימים ולצרפם בעת הגשת התרגיל.

שימו לב - במיני-פרויקט 1 מותר להשתמש רק בפקודות שראינו בשיעור או בתרגול.

על מנת לוודא שהקוד עובד, מומלץ להשתמש בסימולטור MARS או ב-SPIM אותם תוכלו למצוא ב:

• <http://courses.missouristate.edu/KenVollmar/mars/index.htm>

• <http://spimsimulator.sourceforge.net>

ישנן אופציות נוספות (שהן אינן מומלצות) כגון:

• <https://brunonova.github.io/drmips/>

• <http://rivoire.cs.sonoma.edu/cs351/wemips>

אך הן אינן מומלצות מאחר והן סובלות מבאגים במימושים של חלק מהפקודות (למשל JAL).

תוכלו למצוא הדרכה לשימוש בסימולטור MARS וכן הדגמה להתקנה של הסימולטורים MARS ו-SPIM בסרטון המצורף להוראות התרגיל.

שאלה 1

לתרגיל הבית צורף קובץ בשם `fibonacci.asm`. קובץ זה מהווה שלד להרצת תוכנית המקבלת מספר כקלט מהמשתמש ומדפיסה את האיבר המתאים לו בסדרת פיבונאצ'י. לדוגמה, עבור קלט של 20, התוכנה מדפיסה את האיבר ה-20 בסדרת פיבונאצ'י (6765).

לצורך חישוב המספרים הקוד קורא לפונקציה `fib`, אך מימושה חסר.

ממשו את הפונקציה החסרה, תוך הקפדה על עקרונות ה-Calling Conventions.

תזכורת: את סדרת פיבונאצ'י ניתן לחשב בעזרת קוד C הבא:

```
int fibonacci(int n) {
    if (n == 0) {
        return 0;
    }
    if (n == 1) {
        return 1;
    }
    return fibonacci(n-1) + fibonacci(n-2);
}
```



שאלה 2

לתרגיל הבית צורף קובץ בשם `palindrome.asm`. קובץ זה מהווה שלד להרצת תוכנית המקבלת מהמשתמש מחרוזת כקלט ובודקת האם המחרוזת היא פלינדרום (כלומר, האם קריאתה מימין לשמאל או משמאל לימין זהה). לצורך השאלה, גם מחרוזת באורך 0 הינה פלינדרום.

לצורך חישוב המספרים, הקוד קורא למספר פונקציות אותן יש לממש תוך הקפדה על עקרונות ה-`Calling Conventions`.

א. ממשו את הפונקציה `get_length` המקבלת ברגיסטר `%a0` את המיקום בזיכרון של תחילת המחרוזת ומחזירה את אורכה בבתים. לצורך כך הניחו כי לכל קלט שמתקבל מהמשתמש נוסף באופן אוטומטי תו סיום השורה: `'\n'`, אך האורך המוחזר אינו צריך לכלול אותו (כלומר, אורכה של המחרוזת `'abc'` יינתן על פי 3 תווים ולא 4). השורה הראשונה במימוש הנתון של הפונקציה מציבה את תו זה ברגיסטר `%t0`.
שימו לב כי המחרוזת בשאלה נתונה כמערך (בו כל תו מהווה מילה, כלומר 4 בתים).

ב. ממשו את הפונקציה `is_pali_loop` המקבלת ברגיסטר `%a0` את אורך המחרוזת בבתים, וב-`%a1` את המיקום בזיכרון של תחילת המחרוזת. הפונקציה תחזיר 1 במידה והמחרוזת היא פלינדרום ו-0 אחרת. על הפונקציה להיות ממומשת על ידי לולאות, וללא קריאות רקורסיביות.

ג. ממשו את הפונקציה `is_pali_recursive` המקבלת קלט זהה ומחזירה פלט זה לסעיף (ב). על הפונקציה להיות ממומשת על ידי רקורסיה, ללא שימוש בלולאות.

בהצלחה!