מבוא למערכות מחשוב – עבודה מספר 2.3

כללי – בעבודה זאת תממשו אסמבלר. העבודה קשה ומאתגרת, ורצוי להתחיל בה בהקדם האפשרי.

הפרויקט Assembler מכיל שלד של מימוש תהליך התרגום, כולל פונקציות רבות העוסקות בקריאה מקבצים ובעבודה מול מחרוזות. ניתן לבדוק את הפלט שלכם מול הפלט של האסמבלר שסופק בחבילת הכלים המצורפת. אתם יכולים לשנות את הפונקציות הפרטיות שניתנו לכם כרצונכם, אך לא את הפונקציות הפומביות. כזכור, לתהליך התרגום יש שלושה שלבים – תרגום מקרו, מעבר ראשון ליצירת טבלת הסמלים, ומעבר שני ליצירת שפת המכונה. אנו נממש את השלבים בסדר הפוך:

* 1. ממשו את השיטה TranslateAssemblyToMachineCode המתרגמת כל שורה בקוד למחרוזת המכילה בדיוק 16תווים מתוך {0,1} ומייצגת מספר בקוד בינארי, ללא התיחסות לסמלים. שלד השיטה כבר נתון לכם. את השיטות המתורגמות הכניסו לרשימה lAfterPass על ידי lAfterPass.Add(s). לאחר שסיימתם תוכלו להפעיל את האסמבלר על כל קבצי הקוד שאינם מכילים סמלים: Add, MaxL.
  2. הוסיפו מבנה נתונים (Dictionary) המייצג את טבלת הסמלים. ממשו את השיטה CreateSymbolTable היוצרת וממלאת את טבלת הסמלים – משתנים ותוויות. שנו את השיטה TranslateAssemblyToMachineCode כך שתתמוך בתרגום סמלים למספרים. לאחר שסיימתם, תוכלו להריץ את כל תוכנות האסמבלי שאינן מכילות מקרו.
  3. ממשו את השיטה ExpandMacro המקבלת שורת קוד אסמבלי בודדת העשויה להכיל מקרו ומחליפה אותה במספר שורות הממשות את המקרו. במידה ובשורה אין מקרו, יש לרשום לרשימת ההחזרה את השורה המקורית. עליכם לתמוך במקרו הבאים:
     1. עליכם לתמוך ב-increment/decrement, לדוגמא x++,y--,D++.
     2. direct addressing, לדוגמא –x=D , A=y, או x=y.
     3. direct+immediate addressing, לדוגמא x=5.
     4. קיצורים לקפיצות, לדוגמא – D;JGT:LOOP – מאפשר לרשום את יעד הקפיצה באותה שורה של פקודת הקפיצה.
     5. לאחר מימוש המקרו האלו תוכלו לתרגם את SquareMacro.asm.
  4. עליכם לוודא שהאסמבלר שלכם מזהה את הטעויות הבאות (לכל הפחות) וזורק exception מתאים:
     1. שימוש כפול ב-label המגדיר מספר שורה.
     2. טעויות סינטקסט – ערכים לא נכונים עבור שדות.
     3. labels לא חוקיים – למשל שאינם מתחילים באות.

בדקו את קוד שפת המכונה שיצרתם על גבי המכונה שלכם, או על גבי הסימולטור הנתון ב-tools.zip. וודאו שהתוכניות רצות כיאות ומוציאות את הפלט הנדרש.

לעבודה מצורפות תוכנות עזר בתיקיה tools.zip, הממשות, בין השאר, אסמבלר ומדמה (סימולטור) של התנהגות ה-CPU, שיאפשרו לכם לתרגם ולהריץ כל קוד אסמבלי שתרצו על מנת לבדוק האם המכונה והאסמבלר אותם בניתם מתפקדים כראוי. האסמבלר לא תומך במקרו שהגדרתם.

אתיקה – את התרגיל יש לעשות לבד (לא בזוגות). מותר להתיעץ עם סטודנטים אחרים, אך לא להעתיק קטעי קוד מסטודנטים או מהאינטרנט. כל שליחה או קבלה של קטע קוד ממקור כלשהו – סטודנט אחר או מהאינטרנט, במייל, בדיסק נייד, או דרך אתרי שיתוף קבצים אסורה בהחלט! אל תעתיקו!

שאלות ובעיות יש לפרסם בפורום. במידה ולא תקבלו תשובה, יש לפנות לגיא במייל [shanigu@bgu.ac.il](mailto:shanigu@bgu.ac.il)