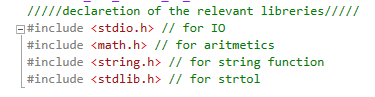
אסמבלר

האסמבלר נכתב בשפת סי ותפקידו להמיר את שפת האסמבלי SIMP לשפת מכונה בה כל שורה כוללת 5 תווים בבסיס 16.

הקלט (asm\_program) של האסמבלר הוא כותב המכיל שפת אסמבלי SIMP והפלט (memin) הינו קובץ טקסט המכיל את שפת המכונה התואמת.

להלן שלבי האסמבלר:

1. הצהרה על הסיפריות הרלוונטיות לקוד.



1. הגדרת משתנים גלובאליים נתונים.

A screenshot of a computer

Description automatically generated with medium confidence

1. הגדרת מבני נתונים של Label, LINE ו-Word.

Text

Description automatically generated

1. הצהרה על פונקציות בהן נשתמש.

Graphical user interface, text

Description automatically generated

1. פונקציית ה-main.

Graphical user interface, text, application

Description automatically generated

* 1. פתיחת שני קבצים ובדיקה כי הפעולה צלחה ללא תקלה. asm\_program לקריאה ו- memin לכתיבה.
  2. אתחול משתנים מרכזיים:
     1. מערך של תווים בשם "label\_array" (טיפוסים מסוג Label) בגודל 4096.
        1. פונקציית האתחול: initilaize\_label\_array(FILE\* asm\_program, Label\* label\_array)
        2. מעבר על קובץ הקלט asm\_program מההתחלה עד הסוף באמצעות לולאת while. בדיקת סוג השורה, במידה והשורה כוללת תווית, הכנסת התווית וכתובתה למערך. כתובת התווית נקבעת לפי משתנה שמתעדכן כל קריאה לפי סוג השורה. Rtype מגדיל ב-1, Itype מגדיל ב-2, תווית ופקודת ".word" לא מגדילים את המשתנה.
     2. טיפוס מסוג Word בשם "word".
     3. מערך מספרים בשם memory בגודל 4096 אשר יכיל את השורות שיודפסו לקובץ הפלט במיקום המתאים במערך.
        1. פונקציית האתחול: initilaize\_memory\_array(int\* memory)
        2. הכנסת המחרוזת "00000" לכלל 4096 ערכי המערך.
     4. מערך של שורות בשם "assembly\_lines" (טיפוסים מסוג LINE) בגודל 4096.
        1. פונקציית האתחול: translate\_to\_fields(FILE\* program, LINE\* assembly\_lines, Label\* label\_array, Word\* word, int max\_label\_array\_index, int\* memory)
        2. מעבר על קובץ הקלט "asm\_program" מהתחלה עד לסופו באמצעות לולאת while. ניקוי השורה מהערות, שמירת ערך של כל שדה בהתאמה בטיפוס LINE (opcode, rd, rs, rt, imm), המרת ערכי השדות למספר בבסיס 16 בעל 5 תווים ושמירתו במקום המתאים במערך memory. במידה והשורה כוללת פקודה מסוג Itype נכניס למקום המתאים במערך את ערך ה-imm. במידה והשורה כוללת פקודת ".word" נכניס את ערך ה-data למקום ה-address במערך memory.
  3. כתיבת המלל בקובץ הפלט "memin":
     1. פונקציית הכתיבה: write\_output\_to\_file(FILE\* output, int\* memory)
     2. מעבר על כלל ערכי המערך memory באמצעות לולאת for והדפסתם בקובץ הפלט "memin" שורה אחרי שורה בהתאמה למיקומם במערך.
  4. סגירת הקבצים:
     1. פונקציית הסגירה: close\_files(FILE\* asm\_program, FILE\* memin)
     2. סגירת שני הקבצים, הקלט asm\_program והקלט memin.