

דוקומנטציה לאסמבלר

האסמבלר בנוי מ-2 פונקציות עיקריות, `read_input`, `write_output`, שנקראות מפונקציית `main` בתורן.

האסמבלר תחילה קורא את קובץ תוכנית האסמבלי, שורה שורה, וממין את הפקודות, כתיבות הזיכרון (`.word`). והלייבלים למערכים גלובליים מהטיפוס המתאים. ולאחר מכן מסדר את הקלט כפי שהוא צריך להופיע בפלט, מתרגם את הלייבלים לערכם (`pcn`) וכותב לתוך קובץ המוצא.

טיפוסים עיקריים:

- **Status:** טיפוס מסוג `enum` המשמש אותנו לדיבוג. זהו טיפוס אותו מחזירות כל הפונקציות העיקריות כך שאם קיימת תקלה במהלך עבודת התוכנית, אנחנו יוצאים בצורה מסודרת והתוכנית מחזירה מספר תקלה שמזוהה עם משפט הסבר.
- **Opline:** מייצג את הפקודות. בעל שדות:
 - `opcode, rd, rs, rt, immediate` מסוג `int` המכילים את הערך המספרי של הפקודות והריגיסטרים בפקודה, על פי הטבלאות.
 - `label` מסוג `string` (מערך `char`) המכיל את שם הלייבל המופיע בשורת הפקודה, במידה ויש כזה.
- **Word:** מייצג את הכתיבות לזיכרון באמצעות `.word`. בעל השדות:
 - `address, data` המכילים את כתובת היעד לכתיבה והמידע שצריך לכתוב.
 - `pc` אשר מייצג את מספר הפקודה (`Opline`) שנכתבה אחרונה. משמש אותנו לצורך סידור הפלט בהתאם לסדר הכתיבה באסמבלי.
- **Label:** מייצג לייבלים. מכיל את שם הלייבל (`string`) ושדה `pc` שמייצג את מספר הפקודה (`Opline`) שנכתבה אחרונה. משמש אותנו לדעת מה ה-`pcn` בקובץ המוצא אליו הלייבל צריך להיות מתורגם.

פונקציות:

- **main:** מקבלת כקלט מהמשתמש את שם קובץ תוכנית האסמבלי (`input`) ואת שם קובץ המוצא (שפת מכונה). הפונקציה פותחת את הקבצים, תחילה את תוכנית האסמבלי, לקריאה, וקוראת לפונקציית `read_input`. לאחר מכן פותחת את קובץ המוצא, לכתיבה, וקוראת לפונקציית `write_output`. הפונקציה דואגת לסגור את הקבצים (גם במקרה תקלה).
- **read_input:** מקבלת כקלט את קובץ האסמבלי, עוברת עליו שורה שורה וממיינת את הקלט למערכים הגלובליים מהטיפוסים המתאימים. הפונקציה מתעלמת מההערות ומרווחים למיניהם. הפונקציה מעדכנת את מערך `Labels` משתמשת ב-2 פונקציות עזר, `update_word`, `update_opline`.
- **update_word:** מקבלת כקלט מצביע לשורה בה יש פקודת `.word`. ומעדכנת את הפקודה למערך `Words`. מתעלמת מרווחים למיניהם ותומכת במספרים דצימליים (חיוביים ושלייליים) ובמספרים הקסאדצימליים שמתחילים ב-`0x`.
- **update_opline:** מקבלת כקלט מצביע לשורה בה אין פקודת `.word`. ולאחר הלייבל (במידה ויש). במידה והשורה ריקה (בהתעלמות מרווחים למיניהם) הפונקציה מסיימת והתוכנית חוזרת ל-`read_input` שממשיכה לקריאת השורה הבאה. במידה והשורה לא ריקה, כלומר יש פקודה, הפונקציה מעדכנת את מערך ה-`Oplines` הגלובלי בהתאם למה שהוגדר בטיפוס המתאים.
- **write_output:** מקבלת כקלט את שם קובץ המוצא. עוברת על המערכים הגלובליים שעדכנו בפונקציות הקודמות ומסדרת אותם במערך עזר `output` לפי הסדר והפורמט בו הם צריכים

להופיע בקובץ הפלט. הפונקציה דואגת לרפד באפסים במידת הצורך (לפקודות word).
ומתרגמת את הלייבלים בפקודות לערכם (pc) ולבסוף כותבת את המערך המסוגר לקובץ
הפלט.

- **find:** פונקציית עזר המשמשת אותנו לתרגם את הרגיסטרים, הפקודות והלייבלים לערכם
המספרי המתאים. מקבלת כקלט את השם (string) וטיפוס enum המייצג את סוג הקלט
(רגיסטר, פקודה או לייבל)