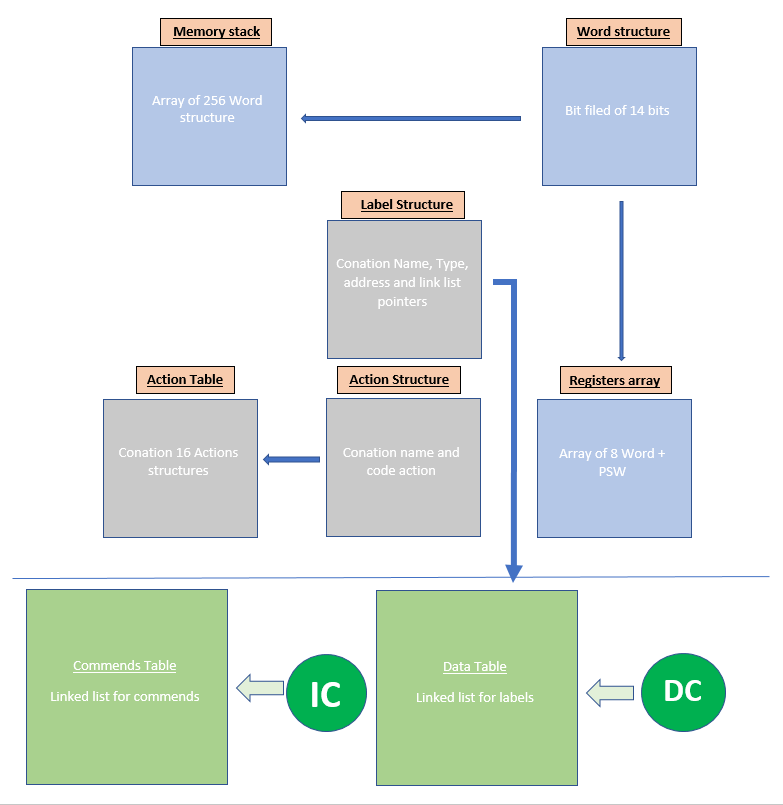
# ארכיטקטורה ומימוש פרויקט גמר – מעבדה בתכנות מערכות

## מבנה התכנית:

### הגדרת מבני הנתונים:



## קבצים:

### Data.c

* יכיל את המימוש של מבני הנתונים המפורטים למעלה עם כל הפונקציות המטפלות בהם.
* צריך לשכתב את כל הפונקציות מחדש.

### Parsing.c

* בדיקה האם השורה ריקה או הערה.

#### IfLabel

בודק האם יש תווית בשורה.

* בודק האם מתחיל בתו אלפא – בית.
* מתחילה בעמודה הראשונה.
* יש רק אותיות או ספרות בשם התווית
* נגמרת בנקודתיים
* אורך של עד 31 תווים.
* מחזיר את שם התווית ללא מערך.

#### ifDirective

בודק האם מדובר בסוג הנחייה מסוג string או data

* בודק האם קיימת תווית
  + במקרה וכן מחפש נקודה מיד לאחר הנקודתיים והרווחים.
  + במקרה ולא לחפש ישר את הנקודה בתחילת השורה.
* לאחר מציאת הנקודה לחפש האם מדובר בסוג הנחייה מסוג string או data
* מחזיר את סוג ההנחיה במחרוזת.

#### ifGlobalDirective

בודק חוקיות עבור משפטי הנחיה extern ו- entry

* בודק האם קיים תווית.
  + במידה וכן, מוציא אזהרה לקובץ השגיאות.
* לאחר מציאת נקודה לוודא האם extern או entry
* מחזירה את סוג ההנחיה בצורה הבאה:
  + עמודה ראשונה – עמודה 0:
    - 0 עבור entry
    - 1 עבור extern
  + את שם התווית בעמודות 1-31

#### VerifyDataCommand

בדוק את החוקיות של הערכים בהנחיה מסוג data.

* להשתמש ב- IfDirective כדי לוודא שזה data.
* בדיקה שמדובר במספרים שלמים.
* פסיקים בין המספרים.
* מחזיר מצביע למערך של int עם כל הערכים לפי הסדר.

#### VerifyStrCommand

* בודקת את החוקיות של הערכים בהנחיה מסוג .string
* מחזירה מצביע למחרוזת אם המחרוזת שהוזנה תקינה או NULL בכל מקרה אחר.

#### תהליך ניתוח שורת הפקודה - ifCommand

* מקבלת שורה ודגל סטטוס.
* בדיקה האם קיימת תווית.
* בדיקת שם הפעולה לפי הטבלה.
* בדיקה האם מדובר בשיטת מיעון 2 (לולאה וכדומה).
  + הכנסה של כל הפרמטרים והאופרנדים לתוך הפונקציה verifyOperand
* במקרה וישנו רק אופרנד יחיד / אין אופרנדים בדיקה האם קיים פסיק.
* במקרה ואין אופרנדים לבדוק שאין כלל תווים לאחר שם הפקודה.
* אם ישנם אופרנדים:
  + להכניס לפונקציה verifyOperand ביחד עם שם הפעולה ודגל האם מדובר באופרנד יעד או מקור (נדרש לממש - אליחי)
    - בדיקה האם מיעון ישיר, מיידי או אוגר
    - בדיקה האם סוג המיעון מתאים לפעולה.
* מחזירה ערך 1 לפקודה תקינה. 0 לפקודה שאינה תקינה ומעדכנת את RS בהתאם לשגיאה.

#### getFirstOperand

* מקבלת שורה ודגל סטטוס
* מחזירה את האופרנד הראשון (כל המחרוזת בין שם הפעולה לבין הפסיק).
* מחזירה NULL בכישלון ומעדכנת את RS

#### getSecondOperand

* מקבלת שורה ודגל סטטוס.
* מחזירה את האופרנד השני (כל המחרוזת אחרי הפסיק)
* מחזירה NULL בכישלון ומעדכנת את RS

#### getFirstParam (רלוונטי רק עבור שיטת מיעון 2).

* מקבלת שורה ודגל סטטוס.
* מחזירה את הפרמטר הראשון (בין הסוגריים הראשונות לפסיק)
* מחזירה NULL בכישלון ומעדכנת את RS.

#### getSecondParam

* מקבלת שורה ודגל סטטוס.
* מחזירה את הפרמטר השני (בין הפסיק עד הסוגריים השניות)
* מחזירה NULL בכישלון ומעדכנת את RS.

#### verifyOperand (אליחי – מימוש בע"מ)

* מקבלת actionID וכן מחרוזת עם שם האופרנד, דגל האם מדובר ביעד או במקור (0 – מקור 1 – יעד 2 -פרמטר של שיטת מיעון 2) ודגל סטטוס.
* בדיקה האם מיעון ישיר, מיידי או אוגר
* בדיקה האם סוג המיעון מתאים לפעולה.
* מחזירה תקין (1) או לא תקין (0) ומעדכנת בהתאם את RS

### FirstPass.c

### SecondPass.c

### OutCreation.c