

פרויקט אסמבלר – מעבדה בתכנות מערכות 2024.

על ידי: ישי אבקיב, שלמה ויס.

קובץ זה יעסוק בהסבר על קטע הקוד המבצע ניתוח תחבירי לתבנית ידוע שבחרנו של הטקסט המתקבל לאחר הקדם אסמבלר שהוא קובץ ללא מקרואים.

במהלך הניתוח התחבירי קטע קוד זה גם מזהה שגיאות תחביריות ובהתאמה מסווג את השורה לשגיאה, דוגמאות לשגיאות: תווית שאינה תקינה, שני פסיקים במקום אחד בין מספר למספר לדוגמה ב data ועוד.

הקבצים אשר קוד זה מכיל הם:

front.c - הקובץ הראשי שם יש פונקציות המפעילות את התהליכים, פונקציות פעולה על הרשימה המקושרת ועוד.

define.c, directive.c, instrucstion.c – קבצים לעיבוד הסוג בהתאם לקובץ והוראות השפה.

headeMethod.h – קובץ המכיל את כל החתימות של הפונקציות ואת הגדרת התבניות.

הפונקציה הראשית של קוד זה הנקראת כדי להשתמש בקטע קוד כולו היא הפונקציה:

```
struct Node *processAssemblyText(const char *filename)
```

אשר מקבלת את שם הקובץ, ומעבירה אותה לפונקציה

```
void processLine(FILE *file, struct Node **head)
```

שם מתבצע התינוח התחבירי על ידי קריאה לאות הראשונה של השורה, והפנייה לפונקציה void

```
categorizeWord(FILE *file, char *word, struct pattern *data, struct Node **head)
```

אשר בה מסווגים את סוג המשפט: define,directive,instrucstion>Error

הפונקציות הראשיות הממשות את הסיווג של כל סוג הם: defineFormat ,directiveFormat ,instructionFormat.

משם נשלחת בדיקת התקינות בהתאם להוראות שפת האסמבלר, אשר רוב הבדיקות לאותו סוג ימצאו באותו קובץ c בהתאמה לסוג המשפט.

בבדיקת התקינות מסווגים את המילים לתוויות מוגדרות מראש אשר הוגדרו ב headeMethod.h במידה ונמצאת השגיאה השמפט מסווג בתור משפט מסוג ERROR ומדפיסים הודעת שגיאה המכילה את סוג השגיאה, מספר השורה, שם הקובץ ועוד, וכן חוזרים לפונקציה categorizeWord ב front.c לבדיקה וסיווג של השורה הבאה.

```
int directiveFormat(FILE *file, char *word, struct pattern *data, struct Node **head)
```

אנחנו מטפלים בתווית היכולה להמיצאות במשפטים לפני הפקודה, וכן בודקים את תקינות התווית על ידי הפונציקה isValidLabel.

הסבר מפרוט לקוד ימצא בין שורות הקוד, וכן ברובו ב headeMethod.h שם לפני כל הגדרת חתימה של פונקציה מובא הסבר על הפונקציה ומשתניה.

תודה, ובהצלחה!: