

# פרויקט סיום בקורס "מעבדה בתכנות מערכות"

סמסטר 2024ב' – עידו קלמן ורואי פיימן – האוניברסיטה הפתוחה

## תוכן עניינים

2.....	הקדמה
3.....	תרשימי זרימה
4.....	אסמבלר עם 2 מעברים
5.....	מבני נתונים

## הקדמה

קובץ זה מכיל כמה תרשימי זרימה שהכנו על מנת להבהיר את אופן המימוש והתהליך שהאסמבלר שבנינו מעביר את הקבצים עד כדי הוצאת קבצי פלט מסודרים ומקודדים.

בנוסף צירפנו את אופן הפעולה הכללי של כל אחד משלושת המעברים :

- קדם מעבד (עליו אחראי הקובץ macros.c),
- מעבר ראשון (הקובץ first\_pass.c)
- מעבר שני (הקובץ second\_pass.c)

בצורה מתומצתת, בנוסף להערות והתייעוד הקיימים בתוך קבצי הפרויקט עצמו.

כמו כן מתוארים בהמשך (בקצרה) מבני הנתונים שבהם השתמשנו בפרויקט וכן מבנה הקבצים.

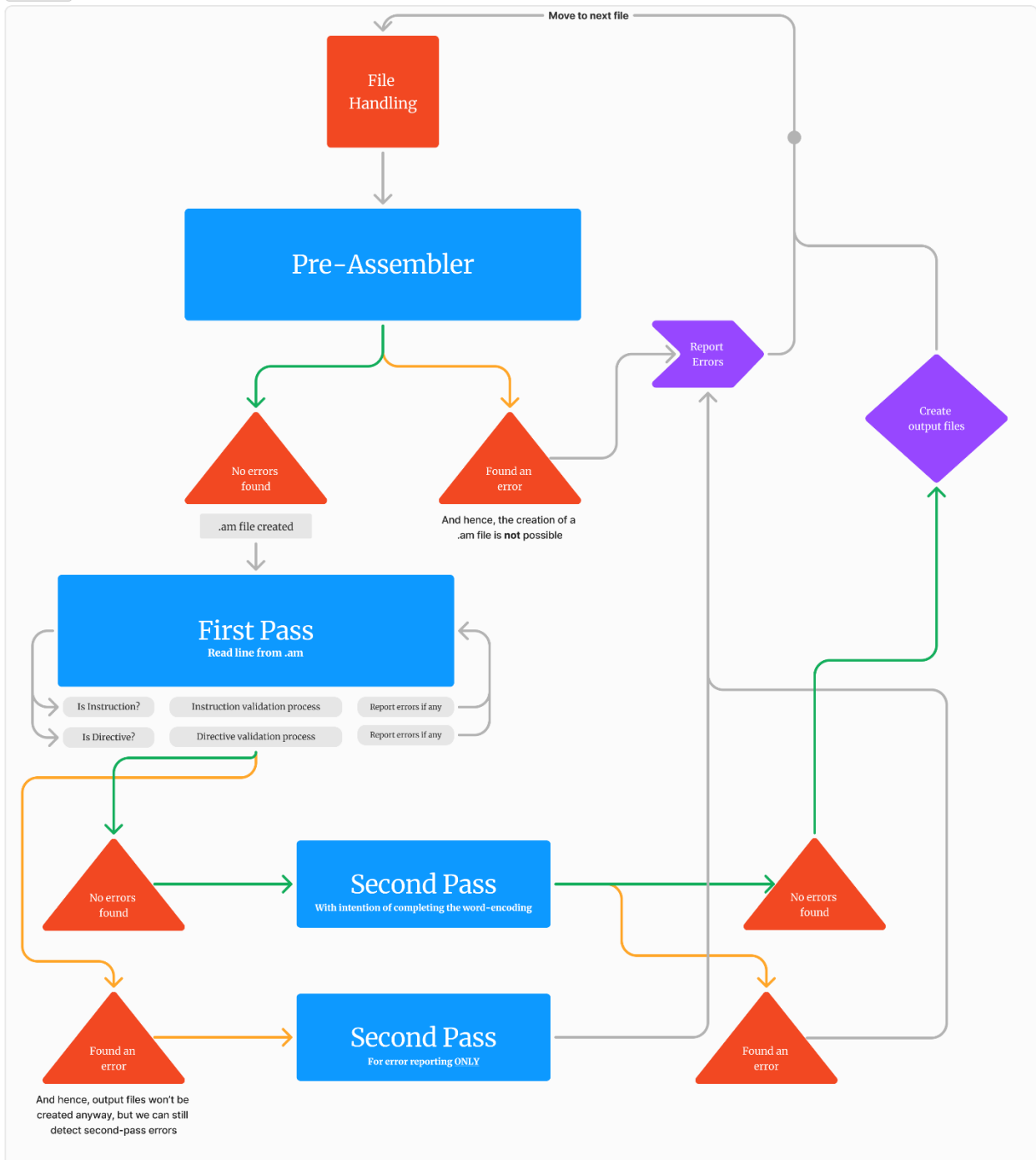
**בהצלחה ותודה!**

## תרשימים

להלן תרשים זרימה שמתאר באופן כללי את אופן הפעולה של האסמבלר על כל שלביו.

קישור לפתיחה במסך מלא ואינטראקטיבי

Section 01



## אסמבלר עם 2 מעברים

כפי שמוסבר בחוברת המטלות, האסמבלר מבצע פעולתו בתוך 2 מעברים על הקוד, ועוד מעבר מקדים שנקרא קדם-אסמבלר.

קדם-אסמבלר: אנו משתמשים בפונקציה מרכזית בקובץ `macros.c`, שמטפלת בהגדרה של מאקרואים בקובץ המקור, מוצאת בהגדרה שגיאה, מתעלמת משורות ריקות ומשורות הערה ועוד. גם קיימת בדיקה לכך ששמות התוויות בקובץ כולן אינן שמות של מאקרואים קיימים. **התהליך ייצור קובץ `am`. ובו קובץ המקור עם מאקרואים פרושים וללא הגדרות, אשר יעבור למעבר הראשון והשני אם לא התגלו שגיאות.** (אם התגלו שגיאות הקובץ יימחק מיידית, כפי שנאמר שניתן לעשות בפורום).

**נעיר:** לפי החוברת וההוראות בפורומים באתר הקורס, ניתן להניח כי תוויות שמוגדרת עם שם זהה לשם של מאקרו (אם כזו קיימת) תוגדר רק אחרי הגדרת המאקרו. כמו כן, ניתן להניח שקריאה לשם של מאקרו (כלומר, מקום בו הוא צריך להיפרש) תגיע רק אחרי שהמאקרו הוגדר. זאת כדי שנוכל לבצע את כל העבודה במעבר אחד ויחיד.

מעבר ראשון: במהלך המעבר הראשון, עוברים על שורות הקובץ `am`. ומבדילים בין שורת הוראה לבין שורת נתונים. מקודדים כל שורה בהתאם בתמונת הזיכרון ומדווחים על שגיאות רלוונטיות, באופן המפורט בתרשים הזרימה לעיל.

מעבר שני: בעת ההגעה למעבר השני, אנו מנסים לקודד מחדש את המילים שמורכבות מאופרנד או יותר שהם תוויות. כמו כן, במעבר זה מתייחסים לשורות הגדרה של `entry`. מאחר ויש לאכוף שההגדרה מתייחסת לתוויות שכבר קיימת בקובץ (ואת בניית טבלת התוויות מבצעים במעבר הראשון), ועוד בדיקות נוספות.

גם כאן נעיר כי במעבר השני אין אנו בודקים שגיאות שכבר בדקנו במעבר השני. כמו כן, כפי שמתואר בתרשים הזרימה, למעבר השני מועבר דגל (`flag`) שדלוק אם על המעבר השני לנסות לקודד מילים מחדש (אם המעבר הראשון עבר בהצלחה ללא שגיאות), או כבוי אם נמצאו שגיאות במעבר הראשון – ואז אין טעם לקודד את המילים (וזה גם יכול ליצור בעיות).

## מבני נתונים וסידור קבצים

סידור הקבצים בפרויקט שלנו מורכב מכמה תיקיות :

1. תיקיית tests : בה אפשר למצוא שלל בדיקות (עם קבצי פלט, תיאור הבעיות שבהן אם ישנן וגם צילומי מסך כפי שנדרש בתיאור המטלה) עליהן הרצנו את האסמבלר. **נעיר**, כי עבור קובץ הבדיקה ה-3 אפילו מצורפת טבלת קידוד כפי שמופיעה בחוברת המטלות. השתמשנו בזה גם כדי לבדוק את עצמנו.

2. תיקיית structs : ההגדרות של מבני הנתונים והמימוש של הפעולות הרלוונטיות לכל מבנה נתונים :

- labels.c – פירוט נמצא גם בתוך הקובץ עצמו, אבל הרעיון הוא לממש רשימה מקושרת מיוחדת ששומרת יותר פרטים מאשר נתון אחד. אנחנו גם שומרים את סוג התווית כדי להשתמש בה במעבר השני ולדעת מתי צריך לקודד מחדש מילה. הקובץ כולל פונקציות לחיפוש לפי שם תווית וסוג, שם תווית בלבד, הכנסת תווית, יצירת המבנה ועוד.

- list.c – פירוט נמצא גם בתוך הקובץ עצמו, אבל הרעיון הוא לממש רשימה מקושרת רגילה, **אך** במקום לשמור רק סוג אחד של נתונים אנחנו שומרים משתנה מסוג void שמאפשר יותר חופש פעולה כדי להכניס משתנה מסוג שלם או מסוג מחרוזת. (בא לידי שימוש ברשימת השימושים ב- extern וברשימת המאקרואים שבונים במהלך הקדם-מעבד). בקובץ פונקציות ליצירת המבנה, הוספת משתנה מסוג שלם / מחרוזת בהתאמה, בדיקה אם הרשימה ריקה ועוד.

3. תיקיית syntax : בתיקייה זו נמצאים הקבצים שעוסקים בטיפול בשורת הוראה ושורת נתונים ופונקציות נלוות האחראיות על בדיקת **התחביר** של הקובץ הנוכחי. (בעיקר בשימוש במהלך המעבר הראשון)

4. תיקיית helpers : בתיקייה זו קיימים קבצים שונים כאשר כל אחד מטפל באספקט שונה של פונקציות שנדרשות במהלך אחד או יותר מהשלבים של האסמבלר.

- memory.c : פונקציות הקשורות בניהול זיכרון ופתיחת קבצים.

- logs.c : פונקציות הקשורות בהדפסות (הדפסת הצלחה / כישלון / אזהרה).

- strings.c : פונקציות שימושיות בעבודה עם מחרוזות (למשל, קפיצה על רווחים, העתקה לתוך מחרוזת עד לרווח הראשון, ועוד).

- symbols.c : פונקציות שימושיות שקשורות בעיקר בזיהוי של רגיסטר / פעולה / שיטת מיעון / סוג שורה (הוראה או נתונים), וגם פונקציות שרלוונטיות בהקשר של זיהוי מילה שמורה או ולידציה של פקודה מסויימת עם שיטות מיעון נתונות.

- misc.c : פונקציות נוספות הקשורות בולידציה וחילוץ נתונים מתוך שורה.

assembler.c – הקובץ הראשי של הפרוייקט עם פונקציית main ופונקציה זו בלבד.

file\_handler.c – קובץ האחראי על העברת קובץ יחיד בין כל 3 השלבים הנדרשים.

second\_pass.c / first\_pass.c – קובץ האחראי למימוש המעבר הראשון / השני (בהתאמה).

macros.c – קובץ האחראי למימוש המעבר המקדים (קדם-אסמבלר).

output.c – קובץ האחראי על יצירת קבצי פלט במידת הצורך.

१०८