הצעת פרויקט יג תוכנה

שם פרוייקט:

ACE(Ace Compiling Environment) compiler

שם תלמיד: ליחי סויסה

תעודת זהות: 214907651

שם מכללה: מכללת אורט הרמלין נתניה

סמל מוסד: 471029

רקע תיאורטי בתחום הפרויקט

מחשב
מחשב הוא כלי טכנולוגי הפועל על פי אותות חשמליים ויודע לפרש אותם על מנת לבצע חישובים מסובכים מקובל לפרש אותות חשמליים אלו כך: 1- יש חשמל 0- אין חשמל, על מנת להקל על בני אדם להורות למחשב מה לעשות נוצרו שפות תכנות

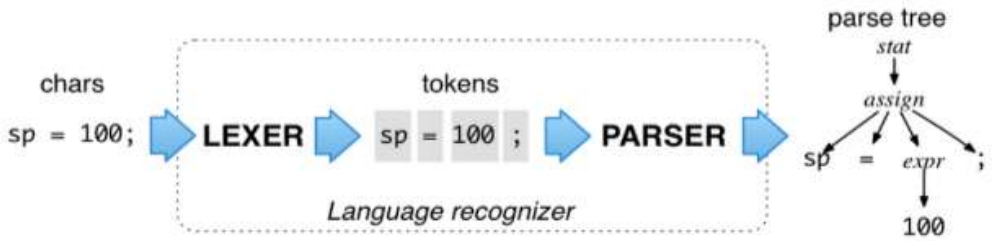
שפות תכנות
שפות תכנות הן שפות אשר מורות למחשב אילו פעולות לבצע באיזה סדר מתי ובאיזה אופן, שפות אלו נוצרו אך ורק בשביל בני אדם, שכן השפה היחידה שמבין המחשב היא כאמור 0- אין אות ו1- יש אות אך למען הנוחות נוצרו שפות תכנות

מהדר
על מנת להמיר בין שפת בני האדם(שפת התכנות) לשפת המכונה(0 ו1) נוצר המהדר או "קומפיילר" בלעז ותפקידו הוא לקחת קובץ טקסטואלי בשפת תכנות כלשהי ולהמיר אותו לפקודות שהמחשב יוכל להבין
על מנת לבצע זאת נהוג לחלק את פעולות הקומפיילר ל5 שלבים והם:

1. לקסר - ניתוח של השפה שמתקבלת בקובץ טקסט והגדרה של מטבעות קבועים מראש(טוקנים)
2. פרסר - חלוקת המילים שקיבלנו מקובץ הטקסט למטבעות וחלוקתם לקבוצות משותפות
3. עץ סינטקס אבסטרקטי - התפקיד של עץ בינארי אבסטרקטי הוא לחלק את הקוד המפורסר לתתי משפחות ומחלקות על פי הגדרות שניתנו מראש העץ הבינארי מזהה משפחות שונות של מילות מפתח ומחלק אותן לקבוצות קטנות
4. ייצוג אמצעי - מייעל ומסדר את הקוד וממיר את שפת הקוד לשפה אמצעית אשר נראית דומה לאסמבלי בדרך כלל וקל יותר להמיר אותה לשפת מכונה ולהריץ.
5. הפקת קוד - קוד הרצה אשר מכיל את הקוד המומר לשפת מכונה ויכול לרוץ על המחשב. קורא את ייצוג הביניים וממיר אותו לשפת מכונה ויוצר ממנו object file/executable file. הוא ממיר את ייצוג הביניים הקרוב לאסמבלי לפקודות אסמבלי ומשם הקוד מומר ישירות לשפת מכונה ומתווסף לקובץ ההרצה הסופי.

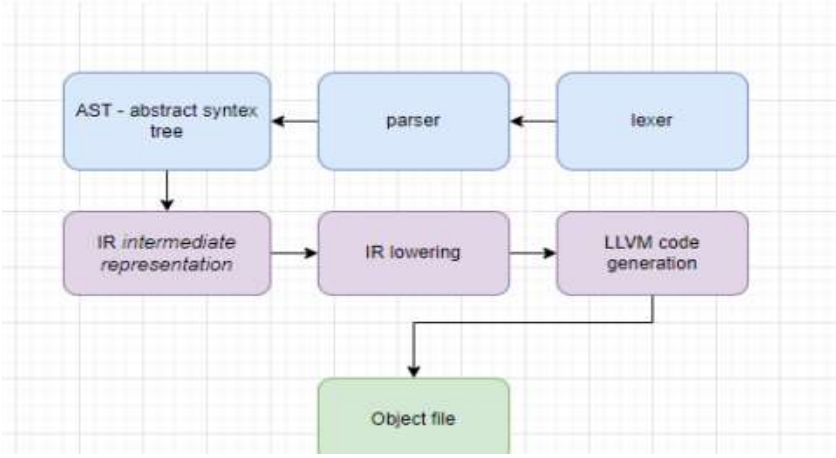
תיאור הפרויקט

בפרויקט זה ארצה לבנות מהדר משלי אשר ימיר שפה שאמציא בעצמי לשפת מכונה על מנת לבחון לעומק את תהליכי המחשב השונים בעת יצירת קוד, בנוסף ארצה להתעמק בנושאי אופטימיזציות של מהדרים שונים ולבחון את יכולותיי בפרויקט מסוג פיתוחי
השפה שאבנה תהיה מורכבת מפקודות אשר דומות לשפת c סטנדרטית ותעקוב אחרי חוקי שפה בסיסיים



בנוסף ארצה להוסיף אופטימיזציות ייחודיות ומעניינות לשפה אשר יהפכו את התהליך למהיר יותר וכזה אשר דורש פחות זיכרון וזמן מעבד.
לאחר כתיבת סינטקס השפה אקח קובץ קוד מוגמר בשפה ואתרגם אותו לשפת mips assmbly ואת זה אריץ בעזרת tasm שהוא אסמבלר אשר יתרגם את הקובץ שלי לשפת מכונה.
השפה שאבנה תהיה שפת תכנות פרוצדוראלית.
המהדר שאכתוב יהיה מהדר אשר ממיר שפת תכנות חדשה מומצאת לשפת מכונה והופך את הקוד שנכתב לקובץ הרצה, המהדר בנוי בשפת C++ ועושה תהליך של אופטימיזציית קוד.

מטרת הפרויקט היא להוות מתורגמן בין שפה מומצאת שאני קבעתי והגדתי מראש לשפת מכונה שמחשב יכול לקרוא ולהריץ
הפרויקט יתן למשתמש חוויה אינטואיטיבית ונוחה לכתוב קוד בצורה יעילה ברורה ואלגנטית.



תכולת השפה תהיה עד turing complete ואם ישאר לי זמן גם מעבר
פירוט:
השפה תכלול השמת משתנים, לולאות ותנאים, פונקציות, שמירת ערכים לזיכרון המחשב, ורקורסיה(היכולת לקרוא לפונקציה מתוך עצמה)
בנוסף ארצה להוסיף אופטימיזציות חישוב אשר יגרמו לקוד להיתרגם לקוד יעיל יותר.
הסינטקס של השפה יהיה דומה ברובו לסינטקס שפת c ויראה לפי הטבלה הבאה:

תיאור	מילה שמורה
מילה זו תשמש להגדיר ערכים מסוג מספר שלם	int
הגדרת תווים	char
הגדרת ערך אמת או שקר	bool
הגדרת מחרוזת	string
תנאי בסיסי	if
תנאי "אחרת"	else
התחלת לולאה חזרתית מוגדרת מראש	for
התחלת לולאה חזרתית לא מוגדרת	while
יציאה ממקטע הקוד הנוכחי	break
הכפלת סכום הבתים של ערך מוגדר לפני המילה	long
הגדרת טיפוס ריק	void
החזרת ערך	return
פעולת חיבור	+
פעולת חיסור	-
פעולת כפל	*
פעולת חילוק	/
פעולת שמירת שארית	%
השוואה ישירה	==
שלילת השוואה	=!
פעולת השוואת ערך קטן מ	<

פעולת השוואת ערך גדול מ	>
פעולת השוואת ערך גדול שווה	=>
פעולת השוואת ערך קטן שווה	=<
שער לוגי וגם (AND)	&&
שער לוגי או (OR)	||
שער לוגי שלילה (NOT)	!
פעולת השמה לתוך משתנה	=:
ביטוי טרינארי(תנאי מקוצר)	:?
תחילת בלוק קוד חדש	{
סיום בלוק קוד	}
גישה למשתנה סטטי	@
סיום שורת קוד	;

דוגמאות לקטעי קוד אפשריים בשפה:

```
int value := 6;
string bbq := "i love bbq";

if(value == 6)
{
    print(bbq);
}
else
{
    print("That was not awsome");
}

while(value > 1)
{
    value = value -1;
}
```

הגדרת הבעיה האלגוריתמית:

על מנת לסיים את הפרוייקט בהצלחה אצטרך להתמודד עם כמה בעיות אלגוריתמיות מאתגרות, קודם כל אצטרך להגדיר את סינטקס השפה עד סופו מראש על מנת להחליט איך תיראה השפה הכתובה ומה יהיו חוקיה, לאחר מכן אצטרך לנהל אלגוריתם אשר יבדוק מילות קוד שמורות ויחלק אותן לקטגוריות, בשלב זה גם יתבצע ניתוח שגיאות וכמה אופטימיזציות בסיסיות לקוד, לאחר מכן אעבור לשלב תרגום הקוד בו אצטרך להחליט לאילו קטעי קוד בשפת אסמבלי להמיר את הקוד שנכתב בשפה שלי על מנת לתרגם את הקוד בצורה היעילה והטובה ביותר.
לסיכום כל שלב ביצירת הקומפיילר מהווה בעיה אלגוריתמית בפני עצמו וכשאשלב את כל החלקים ביחד אצטרך לדאוג שהם כולם עובדים בסינכרון מושלם כדי ליצור שפה נוחה לשימוש ויעילה

תהליכים עיקריים בפרוייקט:

- לקסר
  - הקלט הוא הקוד אשר אנו רוצים לקמפל לשפת מכונה, בדר"כ, ייקרא תחילה כקובץ ואז נקרא כל מילה בנפרד (התעלמות מרווחים).
  - הפלט הוא Token, מספר או תו אשר מזהה מה הסוג של המילה שנקראה(שם/מזהה, מילת מפתח, ערך, פעולה בינארית וכו').

- הוא קורא את המילה/מספר הנוכחי/ת, מזהה איזה סוג הם ומחזיר את ה-token התואם תוך שמירה של המילה/תו לשימוש מאוחר יותר ב-oarser.
- ע"י קריאת התו הראשוני של המילה/מספר. לאחר מכן הוא בודק לאיזה קריטריון בסיסי היא תואמת (מזההים, ערכים, פעולות בינריות וכו'). לאחר מכן הוא קורא את הערך בהתאם ומזהה איזה token שייך לו(למשל למזההים, האם זה מזהה כללי, או מילת מפתח ספציפית וכו').

- פרסר
  - הקלט הוא קוד מחולק לTOKENS.
  - הפלט הוא הבסיס לעץ בינארי אבסטרקטי. (יוסבר בהמשך)
  - מחלק TOKENS לקבוצות ומשפחות על מנת להפוך אותם לבסיס עבור השלב הבא שהוא עץ בינארי אבסטרקטי.
  - הפרסר מזהה מילות מפתח אותן הוא ממיין לקטגוריות שונות שנקבעות על פי סוג מילת המפתח והקשרה.

- עץ בינארי אבסטרקטי
  - הקלט הוא קוד ששעבר תהליך פרסור על ידי הפרסר
  - הפלט הוא קוד שמחולק לתתי משפחות ומוכן לעבור לשלב התרגום לקוד המיוצג על ידי שפת אמצע
  - התפקיד של עץ בינארי אבסטרקטי הוא לחלק את הקוד המפורסר לתתי משפחות ומחלקות על פי הגדרות שניתנו מראש
  - העץ הבינארי מזהה משפחות שונות של מילות מפתח ומחלק אותן לקבוצות קטנות

- intermediate representation
  - הקלט הוא קוד ממויין ומסודר לקטגוריות בעץ בינארי מופשט.
  - הפלט הוא קוד בשפה אמצעית שדומה לאסמבלי אשר מיועל ומוכן להפוך לקוד בשפת מכונה.
  - מייעל ומסדר את הקוד וממיר את שפת הקוד לשפה אמצעית אשר נראית דומה לאסמבלי בדרך כלל וקל יותר להמיר אותה לשפת מכונה ולהריץ.

- קמפול לשפת מכונה
  - ייצוג ביניים של הקוד שקרוב יותר לאסמבלי אשר מיועל ומוכן להפוך לשפת מכונה
  - קוד הרצה אשר מכיל את הקוד המומר לשפת מכונה ויכול לרוץ על המחשב.
  - קורא את ייצוג הביניים וממיר אותו לשפת מכונה ויוצר ממנו object file/executable file.
  - הוא ממיר את ייצוג הביניים הקרוב לאסמבלי לפקודות אסמבלי אשר מיוצגות בייצוג הביניים ומשם הקוד מומר ישירות לשפת מכונה ומתווסף לקובץ ההרצה הסופי.

שפות תכנות

שפת התכנות c++.

סביבת עבודה

סביבת העבודה שלי תהיה
Visual studio code

לוחות זמנים

1(3.11.20) – הצעת ותיאור פרוייקט
2(17.11.20)– נוסח סופי להצעת פרוייקט
3(1.12.20)– הגשת הצעה למשרד החינוך
4(26.1.21)– סיום לקסר ופרסר
5(אפריל 21) – הגשת דו" ח ביניים + סיום הפקת קוד
6(אפריל 21)– סיום סופי של הקומפיילר
7(אפריל -21)-בחינת מתכונת כולל תיק פרויקט
8(אפריל 21)- סיום תיקון שגיאות
9(אפריל 28) – הגשת תיק פרויקט
10(מאי 21)– הגנה על עבודת הגמר – פנימי
11(מאי 21)– הגנה על עבודת הגמר – חיצוני

חתימת סטודנט:_____ חתימת רכז המגמה:_____

חתימת משרד החינוך:

_____