

ספר פרוייקט - י"ג תוכנה ACE compiler



שם מרצה: מיכאל צ'רנובילסקי
שם תלמיד: ליחי סויסה
תעודת זהות: 214907651

תוכן עניינים

| | |
|---------|-------------------------|
| 3..... | מבוא |
| 4..... | תיאור הפרויקט |
| 5..... | הגדרת הבעיה האלגוריתמית |
| 6..... | הגדרת מטרות ומשימות |
| 7..... | שלבים בפרויקט |
| 8..... | תיאור מחלקות |
| 9..... | ארכיטקטורת פרוייקט |
| 10..... | מבני נתונים |
| 11..... | שפת תכנות |
| 11..... | סביבת העבודה |
| 12..... | אלגוריתמים מרכזיים |

מבוא

לפני שאסביר על תהליך בניית המהדר ארצה להציג בפני מי שקורא את ספר הפרויקט מונחים בסיסיים שעליהם אבנה במהלך כתיבת הספר וחובה על הקורא להבין ולדעת על מנת לא לפגוע ברצף הקריאה והבנת הפרויקט.

שפת תכנות - כאשר אנחנו מדברים על שפת תכנות אנחנו בעצם מדברים על אוסף הגדרות שכאשר נרכיב אותן ביחד ניצור תוכנית בשפת התכנות, מטרת שפת התכנות היא להוות מעין גשר בין האדם המתכנת לבין המכשיר הטכנולוגי שאותו הוא רוצה להפעיל, מבחינת המחשב לדוגמא, אין משמעות לשפת התכנות עצמה אלא למה שהיא הופכת אליה לאחר תהליך ההידור, אך מבחינת האדם היא קריטית ביותר שכן קשה מאוד לתכנת בשפת מכונה המורכבת מ2 ספרות שהן 0 ו1.

הידור - לשם כך נוצרו המהדרים (קומפיילרים) אשר כל מטרתם היא להמיר בין שפת תכנות שהומצאה על ידי אנשים, לשפת מכונה (כאמור 0 ו1) אשר מחשב יכול להבין ולהפעיל פקודות בהתאם

תהליך ההידור כולל 5 שלבים מרכזיים שאפרט בהרחבה בהמשך הספר, כעת נסתפק בשמם לקסר, פארסר, עץ סינטקס אבסטרקטי, ייצוג אמצעי, והפקת קוד הינם החלקים המרכיבים קומפיילרים מודרניים.

בפרויקט זה אבנה כל חלק בעצמי ואגדיר שפת תכנות חדשה שתיקרא ACE (ראשי תיבות רקורסיביים של Ace Compiling Environment)

בפרויקט זה ארצה להעמיק את הידע וההבנה שלי בתחום הLow level ולהבין מה קורה מאחורי הקלעים של דבר שנדמה לנו כל כך פשוט ומובן מאליו אך בעצם חבויים בו אתגרים רבים שגם עליהם אפרט בהמשך

קוד - הינו רצף של פקודות בשפת תכנות מסוימת אשר לאחר תהליך ההידור הופך לרצף תווים המורכבים מ0 ומ1, מעבד המחשב יודע להפוך תווים אלה לפקודות פיזיות בפועל בעזרת חשמל כאשר 0 מסמל "אין חשמל" ו1 מסמל "יש חשמל"

תיאור הפרויקט

המהדר שאני בונה מתוכנן להיות דומה כמעט בכל אספקט לשפת התכנות C מכיוון שאני מכיר אותה יחסית טוב ואוהב את הדרך האלגנטית והיפה שבה היא כתובה, השפה שאכתוב תהיה מעין גרסא מנוונת של שפת C (הרי בכל זאת יש בה המון פיצ'רים שלא אוכל להוסיף מפאת מגבלת הזמן) אך הכתיבה בה תהיה יותר נוחה ואינטואיטיבית למשתמש עם פירוט שגיאות יחסית מעמיק.

שפת התכנות שלי מיועדת למתכנתים מתחילים אשר רוצים ללמוד את עקרונות התכנות הבסיסיים ומהווה מעין גשר והכנה לקראת שפת C

מטרות לפרוייקט

- בניית כל חלקי המהדר בצורה יעילה יחסית
- למידה מעמיקה על תהליך ההידור
- יצירת שפה חדשה מאפס
- פיתוח יכולות למידה אישיות

מכיוון שבעבר כבר בניתי מהדר אני מודע לתהליך ולרמת הקושי, אך בניגוד לשפה הראשונה שכתבתי הפעם לא איעזר בטכנולוגיות חיצוניות (בעבר השתמשתי ב flex & bison) אלא אבנה הכל בעצמי ללא עזרה חיצונית (למעט ספריות של מבני נתונים ב CPP)

הגדרת הבעיה האלגוריתמית

בניית מהדר היא משימה מורכבת מבחינה אלגוריתמית. היא דורשת טיפול במספר רב של בעיות, כגון:

ניתוח תחבירי: ניתוח קוד המקור של התוכנית וזיהוי המבנה שלו, תוך יצירת עץ תחביר.
ניתוח סמנטי: הבנת משמעות קוד המקור, תוך זיהוי משתנים, פונקציות, טיפוסים וביטויים.
יצירת קוד ביניים: הפיכת קוד המקור לייצוג ביניים יעיל, כגון קוד P-code או LLVM.
מיטוב: שיפור יעילות הקוד הביניים, תוך שימוש בטכניקות כמו הסרת קוד מת, ניתוח זרימה נתונים ועוד.
יצירת קוד מכונה: הפיכת הקוד הביניים לקוד מכונה ספציפי למעבד.
כל אחת מהבעיות הללו היא בעיה אלגוריתמית קשה בפני עצמה. בנוסף, יש צורך לתאם בין כל השלבים הללו כדי ליצור מהדר יעיל וקורקטי.

אתגרים ספציפיים

בניית מהדר כפרויקט טומנת בחובה מספר אתגרים ספציפיים:

ניהול מורכבות: פרויקט מהדר מורכב בדרך כלל ממספר רב של קבצים ומודולים. ניהול המורכבות הזו הוא אתגר משמעותי.
בדיקות: בדיקת מהדר היא משימה קשה, כיוון שיש צורך לוודא שהוא עובד בצורה נכונה עבור מגוון רחב של קודי מקור.
תחזוקה: מהדרים מתעדכנים באופן קבוע כדי לתמוך בשפות תכנות חדשות, תכונות חדשות וטכנולוגיות חדשות. תחזוקה של מהדר לאורך זמן היא משימה אתגרית.
פתרונות קיימות מספר גישות לפתרון הבעיות האלגוריתמיות בבניית מהדר:

שימוש בטכניקות קיימות: ניתן להשתמש בטכניקות קיימות מתחום מדעי המחשב, כגון אלגוריתמי ניתוח תחבירי, אלגוריתמי ניתוח סמנטי וטכניקות אופטימיזציה.
שימוש בכלי תוכנה: קיימים מספר כלי תוכנה שיכולים לסייע בבניית מהדר, כגון גנרטורים של עצי תחביר ומפרשים (flex & bison).

הגדרת מטרות ומשימות

מטרת הפרויקט:

- בניית מהדר עבור שפת תכנות דמוית C, שתתאים למתכנתים מתחילים.
- למידה מעמיקה על תהליך ההידור.
- יצירת שפה חדשה מאפס.
- פיתוח יכולות למידה אישיות.

אפיון שפת התכנות:

- תחביר דומה ל-C, תוך פשטות ונוחות שימוש.
- פירוט שגיאות מעמיק.
- מיועדת ללימוד עקרונות התכנות הבסיסיים.
- מהווה גשר והכנה לקראת שפת C.

מגבלות הפרויקט:

- גרסה מנוונת של C, ללא כל הפיצ'רים המתקדמים.
- בניית כל חלקי המהדר בצורה יעילה יחסית.
- שימוש בספריות מבני נתונים ב-C++ בלבד, ללא טכנולוגיות חיצוניות (flex & bison).

אתגרים בפרויקט:

- מורכבות אלגוריתמית גבוהה.
- ניהול מורכבות קוד.
- בדיקות מקיפות.
- תחזוקה לאורך זמן.

שלבים בפרויקט

- **עיצוב שפת התכנות**

- הגדרת תחביר, סמנטיקה, טיפוסים נתונים ופונקציות מובנות.
- יצירת תיעוד מקיף עבור השפה.

- **ניתוח תחבירי:**

- פיתוח אלגוריתם ניתוח תחבירי (parser) עבור קוד המקור.
- יצירת עץ תחביר (syntax tree) לייצוג המבנה של הקוד.
- ניתוח סמנטי:
- ביצוע בדיקות סמנטיות על עץ התחביר.
- זיהוי משתנים, פונקציות, טיפוסים וביטויים.

- **יצירת קוד ביניים:**

- הפיכת עץ התחביר לייצוג ביניים יעיל (כגון P-code או LLVM).

- **אופטימיזציה:**

- שיפור יעילות הקוד הביניים באמצעות טכניקות מיטוב.

- **יצירת קוד מכונה:**

- הפיכת הקוד הביניים לקוד מכונה ספציפי למעבד.

- **בדיקות:**

- פיתוח מערך בדיקות מקיף עבור המהדר.
- בדיקת תקינות המהדר עבור מגוון רחב של קודי מקור.

- **תיעוד:**

- יצירת תיעוד מקיף עבור המהדר, כולל הוראות שימוש ומידע טכני.

- **יתרונות הפרויקט:**

- למידה מעמיקה של תהליך ההידור.
- פיתוח יכולות תכנות מתקדמות.
- יצירת כלי תוכנה שימושי עבור מתכנתים מתחילים.

סיכום: בניית מהדר היא פרויקט מורכב ומרתק, המצריך ידע מעמיק במדעי המחשב ובתכנות. הפרויקט הנוכחי מציע הזדמנות ייחודית ללמוד את תהליך ההידור על כל שלביו, תוך יצירת שפת תכנות חדשה ושימושית.

תיאור מחלקות

ארכיטקטורת פרוייקט

מבני נתונים

שפת תכנות

סביבת העבודה

אלגוריתמים מרכזיים