



دانشگاه اصفهان

دانشکده مهندسی کامپیوتر

گروه فناوری اطلاعات

درس طراحی کامپایلر

عنوان پروژه:

طراحی و پیاده‌سازی کامپایلر زبان C

تهیه‌کنندگان:

امیرحسین عرب‌پور

سارا شاه‌طوسی

میلاذ محمدی

نام استاد درس:

دکتر زهرا زجاجی

نام دستیار آموزشی درس:

مهرداد قصابی

نیم‌سال اول تحصیلی ۱۴۰۰-۱۴۰۱

فهرست مطالب:

۲	چکیده:
۴	معرفی بخش‌های مختلف پروژه:
۴	بخش PreProcess:
۵	بخش Lexical:
۵	بخش Parsing:
۶	بخش‌های دیگر:
۷	چالش‌ها، یادگیری‌ها و پیشنهادات:
۷	چالش‌ها:
۷	یادگیری‌ها:
۸	پیشنهادهای:
۹	مراجع:

چکیده:

هدف اصلی این پروژه طراحی یک کامپایلر به زبان C است. که در هفت مرحله (که مراحل با اسم خان نام‌گذاری شده است) طرح شده است. به ترتیب شامل:

پیش‌پردازش، تحلیل لغوی، تحلیل نحوی، تحلیل معنایی، تولید کد میانی، بهینه‌سازی و تولید کد ماشین است.

تیم ما باتوجه به سر فصل درس داده شده و شرایط موجود، خان اول و دوم را به طور کامل و فاز سوم را در حد طراحی کلی، به انجام رسانده است.

منظور از تهیه این داکيومنت، در جهت مستندسازی فعالیت‌ها و توضیح در رابطه با ساختار کلی کد است.

معرفی بخش‌های مختلف پروژه:

در این بخش به معرفی فایل‌های مختلف کد پروژه و توابع پیاده‌سازی شده و استفاده شده هرکدام خواهیم پرداخت. هدف از این بخش تشریح کارکرد کلی هر بخش است و از بیان جزئیات پیاده‌سازی صرف نظر شده است. به جز موارد به‌خصوصی که عدم توضیح آن می‌تواند باعث به وجود آمدن ابهام یا سوال شود.

بخش PreProcess:

این بخش فعالیت‌های اصلی مربوط به پیش‌پردازش زبان C را انجام می‌دهد. منابع اصلی برای فهم کارکرد بخش پیش‌پردازش از سایت [Tutorials Points](#) و سایت [Microsoft](#) بوده است.

به طور کلی بخش پیش‌پردازش مربوط به معرفی‌هایی‌ست که به طور معمول در ابتدای فایل انجام می‌شود. مانند کتابخانه‌ها و فایل‌های استفاده شده در کد، و یا تعریف به خصوص اسم‌ها.

طراحی این بخش با میلاد محمدی و سارا شاه‌طوسی بوده است و پیاده‌سازی با میلاد محمدی.

جدول توضیح توابع خاص:

نام تابع	توضیح
re	کتابخانه Regular Expression های پایتون
file	توابع کار با فایل پایتون

جدول توضیح توابع پیاده‌سازی شده:

نام تابع	توضیح
name_files	برای دریافت path فایل‌ها مورد استفاده
include_add	برای اضافه کردن محتویات فایل‌های include
defines_to_dictionary	برای ساختن map مربوط به defines
replace_defines	استفاده از map ساخته شده برای جایگذاری defines
remove_comments	نمایش متنی مبنای ۱۶ به نمایش باینری
to_temp	نمایش باینری به نمایش مبنای ۱۶

بخش Lexical:

در بخش lexical هدف تبدیل کد ورودی به مجموعه‌ای از توکن‌ها است. که پیاده‌سازی این بخش با استفاده از مفاهیم regular expression ها و dfa ها بوده است.

جدول توضیح توابع خاص:

توابع کتابخانه‌ای خاصی در این قسمت استفاده نشده است.

جدول توضیح توابع پیاده‌سازی شده:

[نیازمند تکمیل]

بخش Parsing:

برای بخش پارس با کمک از یادگیری منابع اینترنتی و جلسات تیمی که برگزار شد، به طراحی گرامرها مربوط به زبان C که تا حد ممکن ساده باشد اما فعالیت‌های اساسی را پوشش دهد پرداختیم.

فایل مربوط به گرامر به همراه مثال از گرامرها قرار داده شده است.

همچنین با کمک مفاهیمی که در رابطه با طراحی top down و طراحی bottom up بود، سعی کردیم با امتحان هر دو شیوه به یک طراحی کد مناسب برسیم که در این زمینه موفق نبودیم

جدول توضیح توابع خاص:

توابع کتابخانه‌ای خاصی در این قسمت استفاده نشده است.

جدول توضیح توابع پیاده‌سازی شده:

بخش‌های دیگر:

باتوجه به اینکه بخش‌های قبلی به شکل مناسب و کافی تکمیل نشده بود، نتوانستیم این بخش را طراحی و پیاده‌سازی کنیم. اما امیدواریم در آینده نزدیک پیاده‌سازی این بخش را نیز انجام دهیم.

چالش‌ها، یادگیری‌ها و پیشنهادات:

در این قسمت سعی کردیم نتایج و اندوخته خود از پروژه انجام شده را به طور خلاصه شرح دهیم.

چالش‌ها:

بخش الف:

دکیومنت ارائه شده برای پروژه می‌توانست کامل‌تر باشد. باتوجه به گستردگی زبان C، و باتوجه به مشخص نبودن انتظارات پروژه از دانشجو، ما دچار ابهام و شک زیادی در پروسه انجام و طراحی پروژه بودیم که سرعت ما را بسیار کاهش داد.

بخش ب:

عدم امکان کار تیمی به طور منظم و مدام باتوجه به شرایط کرونا، باعث شد که کار تیم بیش از پیش سخت‌تر شود. و باتوجه به اینکه مباحث نیاز داشت که تسلط همه اعضا گروه را داشته باشد، باعث وقفه در انجام پروژه بود.

بخش ج:

کلاس درس به طور تئوری به مباحث می‌پردازد و مثال‌های حل شده در سر کلاس به طور نمونه در بخش توضیح گرامرها، به گرامر ضرب و جمع محدود می‌شد. اما در پیاده‌سازی به حجم بسیار گسترده‌تری از گرامرها نیاز بود که پیچیدگی آن انجام آن را برای ما بسیار سخت می‌کرد.

بهتر بود در طول ترم چند جلسه آموزشی که به طور نمونه به کد نویسی و طراحی بخش‌های کوچکی از پروژه بود، اختصاص می‌یافت و از سمتی فرصت پرسش‌سوال را به دانشجویان می‌داد.

یادگیری‌ها:

بخش الف:

کارکرد زبان C که در گذشته به جزئیات یک زبان برنامه نویسی در این حد برخورد نکرده بودیم.

بخش ب:

پیچیدگی کارکرد یک کامپایلر که با توجه به سختی پیاده‌سازی یک نمونه کوچک آن می‌توان متوجه شد.

بخش ج:

کار تیمی در یک پروژه سنگین که نیاز دارد بخش‌ها به شکل بهینه‌ای بین اعضا پخش شود و به جز کلیات که همه در جریان باشند، افراد درگیر جزئیات بخش یکدیگر نشوند. تا کار با سرعت بهتری پیش برود و سعی شود خروجی کار یک فرد، ورودی و ادامه کار فرد دیگر باشد و نیاز به انجام متوالی کارهای برنامه نویسی نباشد و افراد به طور موازی بتوانند کار کنند. از این طریق زمان جلسات و وقتهای کار تیمی نیز کاهش می‌یابد.

پیشنهادهات:

مراجع:

1. StackOverFlow (همراه همیشگی برنامه نویسان)
2. <https://github.com/benjimr/AES-Key-Expansion/blob/master/AESKeyExpansion.py#L3>
3. <https://github.com/boppreh/aes/blob/master/aes.py>
4. <https://github.com/bozhu/AES-Python/blob/master/aes.py>

با تشکر از استاد درس و دستیار آموزشی درس