

گزارش کار پروژه تحلیلگر لغوی

پیمان روحی مقدم - ۴۰۰۱۲۳۴۱۰۵۴۱۲۵

طراحی کامپایلر - استاد فرهادی - ساعت ۱۳:۳۰ الی ۱۶:۱۵

هدف

تحلیلگر لغوی، اولی بخش یک کامپایلر است که رشته هایی از کد منبع را تبدیل به رشته هایی از جنس توکن می کند. هدف از این کار این است که دستورات و بخش های مختلف کد که از جنس زبان برنامه نویسی کد منبع هستند را از یکدیگر تفکیک کنیم تا در محله بعدی، تحلیلگر نحوی بتواند به راحتی این بخش ها را تشخیص دهد و از درستی نحو کد مطلع شود. همچنین در این مرحله برای اولین بار در روند کامپایل باید جدول نمادها ساخته شود. در آخر، از مهمترین کارهای این نوع تحلیلگر تشخیص خطاهای لغوی و فراهم کردن پیغام های خطایی متناسب با نوع خطای لغوی است.

شرح وظیفه

جنبه های فنی:

- ساخت قواعد و توکن هایی برای کلمات کلیدی، علائم نشانه گذاری، عملگرها، شناسه ها، ثابت ها و توضیحات
- تشخیص توکن مناسب از روی تحلیل لغات رشته ورودی
- اختصاص توکن های مناسب به لغات و ساخت جدول نمادها
- تشخیص خطاهای لغوی
- خروجی دادن رشته ای از توکن ها بر حسب رشته کاراکتر ورودی

شیوه کار

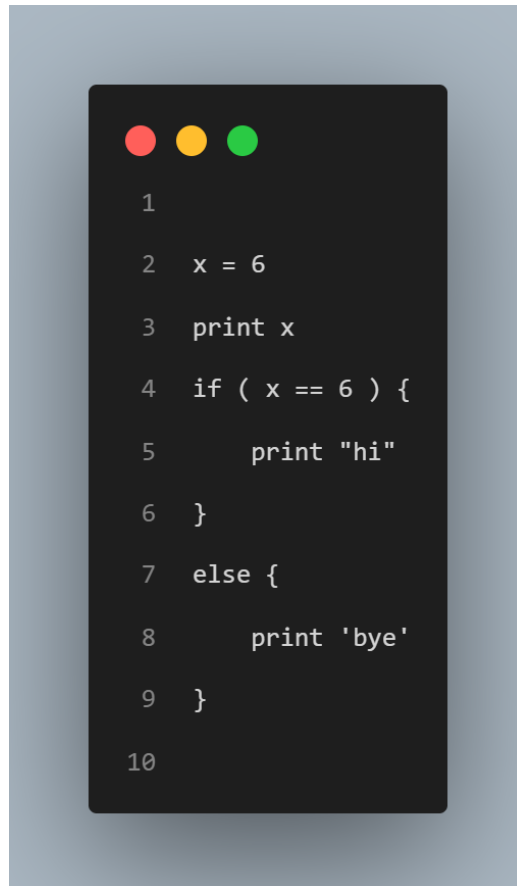
خروجی را لیستی از لیست ها در نظر می گیریم که هر لیست دارای توکن های هر خط است. لغات ورودی باید بر حسب فاصله از یکدیگر جدا باشند تا قابل شناسایی باشند. سپس با استفاده از یک دیکشنری توکن مورد نظر برای هر لغت را پیدا خواهیم کرد. اگر لغاتی بودند که توکن مشخصی برای آنها تعریف نشده بود، آن لغت به عنوان یک شناسه تشخیص داده می شود و در تابعی جداگانه به آن رسیدگی می شود تو در جدول نمادها قرار گیرد و توکن مناسبی دریافت کند

نتایج:

پیشرفت پروژه تاکنون به این گونه است:

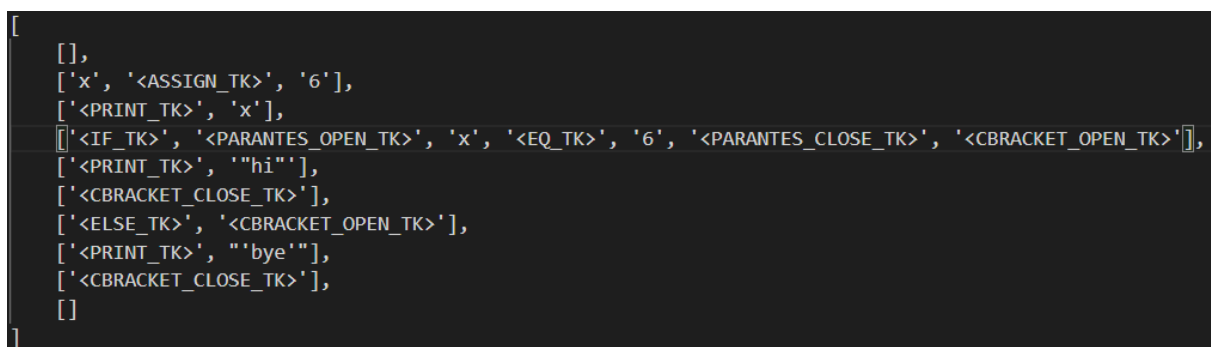
جز توکن شناسه ها، بقیه توکن ها به درستی تولید می شوند.

بخش تولید توکن شناسه ها و تولید و کار با جدول نمادها هنوز پیاده سازی نشده. فعلا به جای این توکن ها صرفا خود آنها در رشته توکن ها قرار می گیرند.



```
1
2  x = 6
3  print x
4  if ( x == 6 ) {
5      print "hi"
6  }
7  else {
8      print 'bye'
9  }
10
```

نمونه ورودی



```
[
  [],
  ['x', '<ASSIGN_TK>', '6'],
  ['<PRINT_TK>', 'x'],
  ['<IF_TK>', '<PARANTES_OPEN_TK>', 'x', '<EQ_TK>', '6', '<PARANTES_CLOSE_TK>', '<CBRACKET_OPEN_TK>'],
  ['<PRINT_TK>', '"hi"'],
  ['<CBRACKET_CLOSE_TK>'],
  ['<ELSE_TK>', '<CBRACKET_OPEN_TK>'],
  ['<PRINT_TK>', "'bye'"],
  ['<CBRACKET_CLOSE_TK>'],
  []
]
```

نمونه خروجی

قدم های بعدی:

- پیاده سازی تابع ساخت توکن شناسه ها و جدول نماد
- بررسی نحوه پیاده سازی تحلیلگر وقتی لغات ورودی با فاصله جدا نشده باشند