

Sveučilište u Zagrebu

Fakultet elektrotehnike i računarstva

Josip PupiĆ

**Seminarski rad iz predmeta
Prevođenje programskih jezika**

Zadatak broj 205

Zagreb, siječanj 2009.

Seminarski rad iz predmeta Prevođenje programskih jezika

Student: Josip PupiĆ

Matićni broj studenta: 0036430601

Zadatak broj 205: pomoću programa *LEX* ostvariti leksičku analizu programskog koda za jezik *Euphoria*

Uvod

Leksička analiza je prvi korak u radu jezičnog procesora. Leksički analizator čita tekst izvornog programa znak po znak, grupira znakove u leksičke jedinice i određuje im tip. Također odbacuje komentare i ostale znakove koji nisu potrebni u daljnjem radu leksičkog analizatora. Leksički analizator provjerava da li leksička jedinica zadovoljava pravila jezika.

LEX (Lexical Analyzer Generator) je jedan od najpoznatijih programa koji olakšava izgradnju leksičkog analizatora.

LEX učitava datoteku sa definicijom leksičkog analizatora (*lex.l*), obrađuje je i stvara *lex.yy.c* datoteku. Tu datoteku je potrebno prevesti u jednom od C prevodioca i dobiva se izvršna datoteka leksičkog analizatora koja je spremna za korištenje.

%{ Deklaracije }%
Regularne definicije
%% Pravila prevođenja %%
Pomoćne funkcije

sl1: izgled lex.l datoteke

Regularne definicije mogu biti zadane i na početku programa.

Deklaracije unutar graničnika `%{ I }%` i pomoćne funkcije izravno se prepisuju u datoteku *lex.yy.c*

Ostvarenje

Ostvareni leksički analizator čita izvorni kod iz zadane datoteke i vrši leksičku analizu nad danim kodom. Analizator razlikuje tri vrste leksičke jedinice: KROS (ključna riječ, operator, specijalni znak), IDN (identifikatore) i KON (konstante).

Program ignorira praznine, tabulatore, oznaku novog reda i komentare.

Ulaz u program je datoteka koja se zadaje kao argument prilikom pokretanja programa. Ukoliko datoteka nije ispravno zadana, program javlja grešku.

```
if((yyin=fopen(argv[1],"r"))==NULL) {  
printf("Nemogu otvoriti ulaznu datoteku.\n");  
exit(1);};
```

Izlaz iz programa su četiri tekstualne datoteke koje predstavljaju izlaznu tablicu, KROS tablicu, tablicu identifikatora i tablicu konstanti.

U tablicu konstanti se zapisuje tip uniformnog znaka (KROS, KON ili IDN) i broj koji označava mjesto na kojem se uniformni znak nalazi u odgovarajućoj tablici.

U ostalim tablicama se zapisuje redni broj u tablici i leksička jedinka.

Na početku *euphoria.l* datoteke nalaze se regularne definicije za praznine, komentare, oznake tabulatora, novog reda, identifikatore i konstante.

```
nista ([ "\t\n"]+)  
real ([0-9]+."[0-9]*)([0-9]+."[0-9]*E("[+]"-")?[0-9]+)  
brojka [0-9]+  
idn [a-zA-Z][a-zA-Z0-9_]*  
string "\"[^\n]*\""  
komentar([^-]+[_a-zA-Z][_0-9a-zA-Z]*[\n])  
hex [#][0-9A-F]+
```

Nakon toga slijede deklaracije potrebne za leksičku analizu kao što su glavni program i potrebni brojači i prototipovi pomoćnih funkcija. .

```
int brKROS=1;  
int brKON=1;  
int brIDN=1;  
void dodaj(char *tip, int br);  
...  
  
int main(int argc, char *argv[])  
...
```

Poslije deklaracija dolaze pravila prevođenja.

Kod pravila prevođenja su navedene ključne riječi, specijalni znakovi i operatori jezika *Euphoria*.

```
"and"      |  
"as"       |  
"break"    |  
.....  
";"        |  
"!"        |  
":"        |  
"." {pretraziKROS(yytext);}
```

Ukoliko je ulazni niz znakova prepoznat kao ključna riješ, identifikator ili konstanta poziva se pomoćna funkcija *pretraziKROS*.

```
{nista}      {;}
{komentar}    {;}
{hex} |
{real} |
{brojka} |
{string} {dodaj("KON  ",brKON++);dodajKON(yytext);}
{idn} pretraziIDN(yytext);
```

U slučaju komentara, praznina i slično, program ne poduzima nikakve akcije, a u slučaju konstante ili identifikatora, poduzimaju se odgovarajuće akcije.

Program koristi sedam dodatnih funkcija koje služe za spremanje leksičkih jedinki u odgovarajuće tablice.

Funkcija *dodaj* služi za spremanje tipa uniformnog znaka i odgovarajuće kazaljke u izlaznu datoteku.

```
void dodaj(char *tip, int br)
{
FILE *dat;
dat=fopen("izlaz.txt","a");
fputs(tip,dat);
fprintf (dat, "%d\n", br);
fclose(dat);
}
```

Funkcije *dodajKROS*, *dodajIDN* i *dodajKON* su skoro identične i koriste se za spremanje leksičke jedinke u pripadnu tablicu.

```
void dodajIDN(char *s)
{
FILE *dat;
char *pom="\n";
dat=fopen("tablicaIDN.txt","a");
fprintf (dat, "%d  ", brIDN-1);
fputs(s,dat);
fputs(pom,dat);
fclose(dat);
}
```

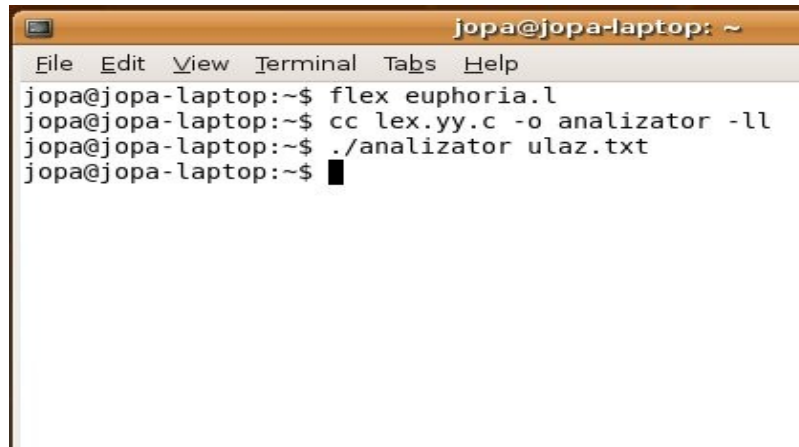
Funkcije *pretraziKROS* i *pretraziIDN* su također identične. One najprije provjeravaju postoji li prepoznata leksička jedinka u pripadnoj tablici. Ukoliko postoji, poziva se funkcija *dodaj* koja sprema uniformni znak i kazaljku na leksičku jedinku u odgovarajućoj tablici, Ako leksička jedinka nije pronađena u tablici, pozivaju se funkcije *dodaj* i *dodaj???* koje spremaju uniformni znak u izlaznu tablicu i dodaju leksičku jedinku u odgovarajuću tablicu.

```

void pretraziIDN(char *s){
    int x;
    char pom[100];
    int flag=0;
    FILE *dat;
    dat=fopen("tablicaIDN.txt","r");
    if(dat!=NULL)
    {
        while (fscanf(dat,"%d %s\n",&x,&pom)!=EOF){
            if(strcmp(s,pom)==0){
                flag=1;
                break;}}
        fclose(dat);}
    if (flag==1) {dodaj("IDN  ",x); }
    else {dodaj("IDN  ",brIDN++);dodajIDN(yytext);}
}

```

Leksički analizator rađen je pod Linux okruženjem. Stvaranje izvršne datoteke leksičkog analizatora i pokretanje prikazano je na sljedećoj slici.



```

jopa@jopa-laptop: ~
File Edit View Terminal Tabs Help
jopa@jopa-laptop:~$ flex euphoria.l
jopa@jopa-laptop:~$ cc lex.yy.c -o analizator -ll
jopa@jopa-laptop:~$ ./analizator ulaz.txt
jopa@jopa-laptop:~$ █

```

Naredbom *flex euphoria.l* dobivamo datoteku *lex.yy.c*.

Sljedećom naredbom *cc lex.yy.c -o analizator -ll* dobivamo izvršnu datoteku *analizator* koju pokrećemo naredbom *./analizator ulaz.txt* nakon čega program stvara izlazne datoteke.

Zaključak

Program *LEX* vrlo učinkovito pojednostavljuje izgradnju leksičkog analizatora. Uz ovaj program, izgradnja leksičkog analizatora se svodi na pisanje regularnih definicija i pravila prevođenja. Međutim, i on ima svoje nedostatke. Najbitniji od tih nedostataka su neodgovarajuće korisničko sučelje, neučinkovitost generiranog program i nedostatne poruke o pogreškama. Mislim da je potrebno napomenuti kako ovaj program ne radi dobro pod *Windows* okruženjem i prilikom korištenja *Lexa* preporučam neku od *Linux* distribucija.