



ΕΡΓΑΣΤΗΡΙΟ

ΜΕΤΑΓΛΩΤΤΙΣΤΕΣ

ΜΕΡΟΣ Α3



ΤΜΗΜΑ Β1

ΠΕΤΡΑΚΗ ΒΑΣΙΛΙΚΗ ΕΥΑΝΘΙΑ 19390193
ΠΑΠΑΧΡΙΣΤΟΔΟΥΛΟΥ ΑΙΚΑΤΕΡΙΝΗ 19390185
ΡΟΥΜΕΛΙΩΤΗΣ ΣΠΥΡΙΔΩΝ 19390205
ΜΑΡΓΙΩΛΗΣ ΧΡΗΣΤΟΣ 19390133

Περιεχόμενα

Στόχος εργασίας.....	2
Κώδικας Λεκτικού Αναλυτή	3
Header file.....	4
Αρχείο Input.....	5
Αρχείο Output.....	6
Ανάλυση Αρμοδιοτήτων	7
Βιβλιογραφία	8

Στόχος εργασίας

Στην παρακάτω εργασία, ως ομάδα, αναλύουμε ένα εισαγωγικό κομμάτι των μεταγλωττιστών, συγκεκριμένα των λεκτικών αναλυτών. Αυτό θα γίνει πάνω στην εικονική γλώσσα προγραμματισμού «Uni-CLIPS», χρησιμοποιώντας το λεκτικό αναλυτή FLEX.

Στο 1^ο κομμάτι υπάρχει ο κώδικας FLEX και το header αρχείο, σχολιασμένος, και στο 2^ο κομμάτι το αρχείο εισόδου μας και η έξοδος αντιγραμμένη σε ένα αρχείο εξόδου.

Γίνεται compile και τρέχει κανονικά, χωρίς κανένα πρόβλημα, τόσο σε 3 διαφορετικά μηχανήματα με Arch Linux, Fedora KDE και FreeBSD, όσο και σε Virtual Machine με Ubuntu.

Κώδικας Λεκτικού Αναλυτή

```

1. %option noyywrap
2.
3. %{
4. #include <err.h>      /* Για χρήση της err() */
5. #include <stdio.h>
6. #include <string.h>
7. #include <stdlib.h>
8.
9. #include "token.h"
10.
11. int lineno = 1;
12. %}
13.
14. /*
15.  * Με βάση το μέρος A2, υλοποιούμε τις κανονικές εκφράσεις και τις
16.  * αντιστοιχούμε στα κατάλληλα tokens.
17.  */
18. DELIM          [ \t]+
19. KEYWORD         deffacts|defrule|test|bind|read|printout
20. OPERATOR        =|\\+|-|\\*|\\/
21. INT             0|[+-]?[1-9]+[0-9]*
22. FLOAT           [+-]?[0-9]+((\\.[0-9]+)([eE][+-]?[0-9]*)?|([eE][+-]?[0-9]*)?)
23. STR             \"^[^\"\\\\]*(?:\\\\\\\\.[^\"\\\\]*)*\"
24. DEFIN           [A-Za-z]+[A-Za-z0-9_]*
25. VAR             \\?[A-Za-z0-9]+
26. COMMENT         ;.*
27. /*
28.  * Εάν συναντήσει token το οποίο δεν ξεκινάει με white space και αλλαγή
29.  * γραμμής, από 1 και παραπάνω φορές.
30.  */
31. UNKNOWN         [^ {DELIM}\\n]+
32.
33. /*
34.  * Όταν βρίσκει οποιοδήποτε token, επιστρέφει την αντίστοιχη κατηγορία στο
35.  * οποίο ανήκει
36.  */
37. %%
38. {DELIM}          { /* ignore whitespace */ }
39. {KEYWORD}        { return TOK_KEYWORD; }
40. {OPERATOR}       { return TOK_OPERATOR; }
41. {INT}            { return TOK_INT; }
42. {FLOAT}          { return TOK_FLOAT; }
43. {STR}            { return TOK_STR; }
44. {DEFIN}          { return TOK_DEFIN; }
45. {VAR}            { return TOK_VAR; }
46. {UNKNOWN}        { return TOK_UNKNOWN; }
47. {COMMENT}        { /* skip comments */ }
48. \\n              { lineno++; }
49. <<EOF>>         { printf("EOF\\n"); exit(0); }
50. %%
51.
52. static char *tokens[] = {
53.     "INT",          /* integers */
54.     "FLOAT",        /* floats */
55.     "STR",          /* strings */
56.     "DEFIN",        /* definitions */
57.     "VAR",          /* variables */
58.     "KEYWORD",      /* keywords */
59.     "OPERATOR",     /* operators */

```

```

60.  "UNKNOWN",          /* unknown token */
61. };
62.
63. int
64. main(int argc, char *argv[])
65. {
66.     int tok;
67.
68.     if (argc < 2) {
69.         fprintf(stderr, "usage: %s input [output]\n", *argv);
70.         return (-1);
71.     }
72.     /* Χρησιμοποιούμε την συνάρτηση err() για εμφάνιση σφαλμάτων */
73.     /* Ανοίγει το αρχείο εισόδου, εάν υπάρχει */
74.     if ((yyin = fopen(argv[1], "r")) == NULL)
75.         err(1, "fopen(%s)", argv[1]);
76.     /* Ανοίγει το αρχείο εξόδου, εάν υπάρχει */
77.     if (argc == 3 && (yyout = fopen(argv[2], "w")) == NULL)
78.         err(1, "fopen(%s)", argv[2]);
79.
80.     while ((tok = yylex()) >= 0) {
81.         /* Για κάθε token που βρίσκεται, εμφανίζονται οι πληροφορίες του */
82.         fprintf(yyout, "line: %5d\ttoken=%s\tval='%s'\n",
83.             lineno, tokens[tok-1], yytext);
84.     }
85.
86.     return (0);
87. }
88.

```

Header file

```

1.  #define TOK_INT      1
2.  #define TOK_FLOAT    2
3.  #define TOK_STR      3
4.  #define TOK_DEFIN    4
5.  #define TOK_VAR      5
6.  #define TOK_KEYWORD  6
7.  #define TOK_OPERATOR  7
8.  #define TOK_UNKNOWN  8

```

Αρχείο Input

```
1. +1234
2. 50
3. -115
4. 3.14
5. -10.0
6. +0.0001
7. 3.14e-10
8. 0e0
9. static-facts
10. MoveUp
11. CUBES
12. sum-1
13. table
14. pacman
15. A-21-b
16. ?x
17. ?X
18. ?3
19. ?ad
20. ?X1b23
21. ?32AbC
22. ?ABcd1234de
23. ""
24. "Test"
25. "Hello world"
26. "Mark said, \"Boo!\""
27. ; this is a comment
28. ignore                whitespace
29. #unknown ?2 ? ?hello ?world
30. deffacts defrule test
31. 2 + 2
32.
```

Αρχείο Output

```

1. line: 1 token=INT val='+1234'
2. line: 2 token=INT val='50'
3. line: 3 token=INT val='-115'
4. line: 4 token=FLOAT val='3.14'
5. line: 5 token=FLOAT val='-10.0'
6. line: 6 token=FLOAT val='+0.0001'
7. line: 7 token=FLOAT val='3.14e-10'
8. line: 8 token=FLOAT val='0e0'
9. line: 9 token=DEFIN val='static-facts'
10. line: 10 token=DEFIN val='MoveUp'
11. line: 11 token=DEFIN val='CUBES'
12. line: 12 token=DEFIN val='sum-1'
13. line: 13 token=DEFIN val='table'
14. line: 14 token=DEFIN val='pacman'
15. line: 15 token=DEFIN val='A-21-b'
16. line: 16 token=VAR val='?x'
17. line: 17 token=VAR val='?X'
18. line: 18 token=VAR val='?3'
19. line: 19 token=VAR val='?ad'
20. line: 20 token=VAR val='?X1b23'
21. line: 21 token=VAR val='?32AbC'
22. line: 22 token=VAR val='?ABcd1234de'
23. line: 23 token=STR val='""'
24. line: 24 token=STR val='"Test"'
25. line: 25 token=STR val='"Hello world"'
26. line: 26 token=STR val='"Mark said, \\"Boo!\\\""'
27. line: 27 token=DEFIN val='ignore'
28. line: 28 token=DEFIN val='whitespace'
29. line: 29 token=UNKNOWN val='#unknown'
30. line: 29 token=VAR val='?2'
31. line: 29 token=UNKNOWN val='?'
32. line: 29 token=VAR val='?hello'
33. line: 29 token=VAR val='?world'
34. line: 30 token=KEYWORD val='deffacts'
35. line: 30 token=KEYWORD val='defrule'
36. line: 30 token=KEYWORD val='test'
37. line: 31 token=INT val='2'
38. line: 31 token=OPERATOR val='+'
39. line: 31 token=INT val='2'
40.

```

Παρατηρούμε στην γραμμή 29, πως μας εμφανίζει το κομμάτι στο οποίο υπάρχει error, και ύστερα συνεχίζει κανονικά να ελέγχει εάν τα επόμενα tokens είναι σωστά ή όχι.

Ανάλυση Αρμοδιοτήτων

- Πετράκη Βασιλική Ευανθία

Συγγραφή παραδοτέου, debugging κώδικα

- Παπαχριστοδούλου Αικατερίνη

Debugging κώδικα

- Ρουμελιώτης Σπυρίδων

Debugging κώδικα

- Μαργιώλης Χρήστος

Συγγραφή κώδικα

Η συγγραφή της εργασίας πραγματοποιούταν σε πραγματικό χρόνο, μέσω ομαδικής κλήσης, οπότε όλα τα μέλη είναι ενημερωμένα για όλα τα κομμάτια της.

Βιβλιογραφία

- <https://stackoverflow.com/>
- <https://eclass.uniwa.gr/courses/CS118/>
- «Μεταγλωττιστές» - Alfred V. Aho, Monica S. Lam, Ravi Sethi, Jeffrey D. Ullman
- “Regular Expressions Cheat Sheet” by DaveChild