



Οδηγίες για τη δημιουργία απλού κώδικα για τη γεννήτρια flex

Στόχος της χρήσης του εργαλείου flex είναι η δημιουργία λεκτικού αναλυτή. Στο παρόν έγγραφο θα εξετάσουμε δύο πολύ απλές περιπτώσεις / παραδείγματα. Αυτό θα δώσει τη δυνατότητα κατανόησης τόσο της χρήσης του εργαλείου όσο και ολόκληρης της διαδικασίας για τη δημιουργία του λεκτικού αναλυτή.

Βήματα για τη δημιουργία - μεταγλώττιση - εκτέλεση του λεκτικού αναλυτή

1. Δημιουργία του αρχείου με τον κατάλληλο κώδικα για το εργαλείο flex. Το αρχείο πρέπει οπωσδήποτε να έχει την επέκταση ".l". Πχ. file.l
2. Το αρχείο αυτό θα πρέπει να περάσει από το flex προκειμένου να μετατραπεί σε κώδικα γλώσσας C. Για να γίνει αυτό θα πρέπει να εκτελεστεί η εντολή:

flex -o file.c file.l

όπου file.l είναι το αρχείο εισόδου σε flex και file.c το αρχείο εξόδου σε γλώσσα C.

3. Ακολούθως θα πρέπει να γίνει η μεταγλώττιση του κώδικα C του αρχείου file.c προκειμένου να δημιουργηθεί το εκτελέσιμο πρόγραμμα του λεκτικού αναλυτή. Αυτό επιτυγχάνεται με την εντολή:

gcc file.c -o file

Όπου file.c είναι ο κώδικας σε γλώσσα C που έχει προκύψει από το flex και file είναι το εκτελέσιμο που θα παραχθεί.

4. Τέλος, για την εκτέλεση του λεκτικού μας αναλυτή αρκεί να δοθεί η ακολουθία χαρακτήρων ./ και το όνομα του εκτελέσιμου αρχείου. Για την παραπάνω περίπτωση αυτό θα είναι:

./file

Ολοκλήρωση Input όταν δίνονται χαρακτήρες στην οθόνη: ^D

Σφάλμα "undefined reference to yywrap"

Σε περίπτωση που εμφανιστεί το παραπάνω λάθος κατά την μεταγλώττιση του κώδικα C, το πρόβλημα μπορεί να αντιμετωπιστεί με έναν από τους παρακάτω τρόπους:

Η yywrap καλείται μετά την ανάγνωση ενός ολόκληρου αρχείου εισόδου. Επειδή εμείς έχουμε ένα και μόνο αρχείο εισόδου σε κάθε εκτέλεση, μας είναι άχρηστη. Για το λόγο αυτό:

- α) Συμπεριλαμβάνουμε τον διακόπτη -lfl στο gcc ώστε να συμπεριληφθεί η προκαθορισμένη yywrap που κι αυτή εξ' ορισμού δεν κάνει τίποτα.

Τρέξτε τον μεταγλωττιστή με παράμετρο -lfl που συμπεριλαμβάνει τη βιβλιοθήκη του flex. Πχ:

gcc file.c -lfl

ή

- β) Βάζουμε "%option noyywrap" στην αρχή κάθε αρχείου flex ώστε να αγνοείται η μη ύπαρξη της yywrap και να μη χρειάζεται ο διακόπτης -lfl.




ή

γ) Γράφουμε μια δική μας συνάρτηση `ywrap` μέσα στον `c` κώδικα που προκύπτει και η οποία πρακτικά δεν κάνει τίποτα απολύτως: `int ywrap() { }` ή `int yywrap() { return 1; }`

Και τα 3 παραπάνω βήματα έχουν πρακτικά το ίδιο αποτέλεσμα.





Παράδειγμα 1

Παράδειγμα κώδικα flex για τη δημιουργία λεκτικού αναλυτή που να υπολογίζει τον αριθμό των γραμμών και των αριθμό των χαρακτήρων ενός αρχείου εισόδου.

<pre>%{ #include <stdio.h> int num_lines = 0, num_chars = 0; }% %% \n { num_lines++; num_chars++; } . { num_chars++; } %% int main() { yylex(); printf("# of lines = %d\n", num_lines); printf("# of chars = %d\n", num_chars); }</pre>	 Κώδικας C
	 Περιοχή Κανόνων
	 Κώδικας C

Παράδειγμα 2

Παράδειγμα κώδικα flex για τη δημιουργία λεκτικού αναλυτή που αναγνωρίζει δυαδικούς αριθμούς και τους τυπώνει στην οθόνη με new-line ως διαχωριστικό.

<pre>%option noyywrap %{ #include <stdio.h> }% binary [01]+ %% {binary} { ECHO; printf("\n"); } . { } %% int main() { yylex(); }</pre>	 Κώδικας C
	 Περιοχή Ορισμών
	 Περιοχή Κανόνων
	 Κώδικας C

Σύνταξη της MAIN για άνοιγμα αρχείου εισόδου

1^η εκδοχή: //δέχεται είσοδο είτε από αρχείο είτε από standard input

```
int main(int argc, char **argv)
{
    int i;
    if(argc == 2)
        yyin=fopen(argv[1], "r");
    else
        yyin=stdin;
    yylex();
}
```

2η εκδοχή: //δέχεται είσοδο μόνο από αρχείο

```
main() {
    FILE *myfile = fopen("input.txt", "r");
    if (!myfile) {
        {
            printf("I can't open input.txt !");
            return -1;
        }
    }
    yyin = myfile;
    yylex();
}
```

3η εκδοχή: //δέχεται είσοδο από αρχείο και δίνει έξοδο σε αρχείο

Γίνεται έλεγχος των ορισμάτων στη γραμμή εντολών. Εάν τα ορίσματα είναι 3, το πρόγραμμα διαβάζει από το αρχείο που αναφέρεται στο 2^ο όρισμα και γράφει στο αρχείο που αναφέρεται στο τρίτο όρισμα. Αν τα ορίσματα είναι 2, γράφει στην οθόνη όπως στην εκδοχή 2. Υπενθυμίζεται ότι το πρώτο όρισμα είναι το όνομα του ίδιου του εκτελέσιμου αρχείου

```
int main(int argc, char **argv)
{
    if(argc == 3){
        if(!(yyin = fopen(argv[1], "r"))) {
            fprintf(stderr, "Cannot read file: %s\n", argv[1]);
            return 1;
        }
        if(!(yyout = fopen(argv[2], "w"))) {
            fprintf(stderr, "Cannot create file: %s\n", argv[2]);
            return 1;
        }
    }
    else if(argc == 2){
        if(!(yyin = fopen(argv[1], "r"))) {
            fprintf(stderr, "Cannot read file: %s\n", argv[1]);
            return 1;
        }
    }
    yylex();
}
```