Πανεπιστήμιο Δυτικής Αττικής

Σχολή Μηχανικών

Τμήμα Μηχανικών Πληροφορικής και Υπολογιστών

ΕΡΓΑΣΤΗΡΙΟ ΜΕΤΑΓΛΩΤΙΣΤΕΣ ΕΡΓΑΣΙΑ Α2.

<u>ΗΜΕΡΟΜΗΝΙΑ ΠΑΡΑΔΟΣΗΣ:</u>

Τρίτη 7 Μαΐου 2024 - 11:59 μ.μ.

<u>ΟΜΑΔΑ ΕΡΓΑΣΤΗΡΙΟΥ:</u>

Τμ.Α2(Τετ 14:00-16:00)

<u>Υπευθύνος Ομάδας:</u>

ΙΟΡΔΑΝΑΚΗΣ ΜΙΧΑΛΗΣ

ΟΜΑΔΑ:

5) ΓΕΩΡΓΙΑ ΠΑΠΑΘΑΝΑΣΙΟΥ (ΠΑΔΑ-21390279)

ΠΑΝΑΓΙΩΤ ΣΠΥΡΟΠΑΛΗΣ (ΠΑΔΑ-19390218)

ΦΙΛΙΠΠΟΣ ΠΑΠΑΓΕΩΡΓΙΟΥ(ΠΑΔΑ-21390174)

ΣΑΝΤΑΣ ΑΝΤΩΝΙΟΣ (ΠΑΔΑ-21390199)

ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ. ……...2

ΕΙΣΑΓΩΓΗ. ……..3

ΕΙΣΟΔΟΣ ΚΑΙΕΞΟΔΟΣ.....4

ΠΑΡΑΤΗΡΗΣΕΙΣ/ΣΧΟΛΙΑΣΜΟΣ ΑΠΟΤΕΛΕΣΜΑΤΩΝ.....9

ΑΝΑΛΥΣΗ ΑΡΜΟΔΙΟΤΗΤΩΝ……..9

Εισαγωγή

Το μέρος Α3 της εργασίας ασχολείται με τη Λεκτική Ανάλυση, δηλαδή την αναγνώριση των λεκτικών μονάδων σε μια ροή συμβόλων. Η αναγνώριση των διάφορων tokens πραγματοποιείται έπειτα από αίτημα του Συντακτικού Αναλυτή, κατά τη διάρκεια της Μεταγλώττισης.

Κεντρικό θέμα της εργασίας αποτελεί η ενασχόληση με το εργαλείο flex. Είναι υπεύθυνο για τη δημιουργία λεκτικών αναλυτών. Περιλαμβάνει μια σειρά από κανόνες, οι οποίοι διαβάζουν τα σύμβολα που δίνονται ως είσοδος χαρακτήραχαρακτήρα και τα μετατρέπουν σε tokens. Το αρχείο που περιλαμβάνει τον κώδικα flex(.l) περνάει από έναν μεταγλωττιστή C. Έπειτα, «τρέχουμε» το αρχείο .c που παράγεται και παρατηρούμε τα αναγνωρισμένα tokens στο αρχείο εξόδου ή στην οθόνη του υπολογιστή μας.

Η δομή ενός προγράμματος flex είναι η εξής:

%{

Ορισμοί

}%

%%

Κανόνες

%%

Κώδικας χρήστη(Συνάρτηση main())

*Ιδιαίτερη αναφορά πρέπει να γίνει στη συνάρτηση **yylex,** η οποία είναι υπεύθυνη για την λεκτική ανάλυση. Όταν δηλαδή καλείται η **yylex** διαβάζεται το επόμενο σύμβολο(χαρακτήρας) του αρχείου εισόδου και εφαρμόζονται οι αντίστοιχοι κανόνες. Η παραπάνω διαδικασία συνεχίζεται έως την αναγνώριση του EOF(το τέλος του αρχείου), δηλαδή του τερματικού συμβόλου. Αναλυτικότερα, η ροή συμβόλων που χρησιμοποιήθηκε αποτελεί το αρχείο input.txt. Περιλαμβάνει ικανοποιητικό αριθμό έγκυρων αλλά και άκυρων λεκτικών μονάδων, με σκοπό να γίνεται κατανοητός ο τρόπος που χειρίζεται το πρόγραμμα τη ροή συμβόλων που του εκχωρούμε.*

Ενώ, με βάση τις κανονικές τους εκφράσεις, αλλά και τη γραμματική της γλώσσας Uni-C τα tokens(λεκτικές μονάδες) είναι τα εξής: Αναγνωριστικά, Λέξεις κλειδιά, Λεκτικά κυριολεκτικά, Δεκαδικοί ακέραιοι, Αριθμοί κινούμενης υποδιαστολής, Σχόλια, και

Unknown_Tokens, τα οποία αποτελούν τα σύμβολα που δεν περιγράφονται σε καμία από τις παραπάνω κατηγορίες.

Σημείωση: *Τα σχόλια αναγνωρίζονται και έπειτα αγνοούνται. Επιπροσθέτως, αγνοούνται και οι white_spaces χαρακτήρες. Όταν εντοπιστεί χαρακτήρας αλλαγής γραμμής, ο μετρητής γραμμών(int line) αυξάνεται κατά ένα.*

Επίσης*, στο αρχείο* ***token.h*** *γίνεται η δήλωση του συνόλου των token, πριν από τη συγγραφή του κώδικα: αρχείο simple-flex-code.l Τα αποτελέσματα της Λεκτικής Ανάλυσης βρίσκονται στο αρχείο output.txt .*

Οι εντολές για την δημιουργία ενός εκτελέσιμου προγράμματος το οποίο δέχεται ως είσοδο το input.txt και παράγει ένα output.txt ως έξοδο βρίσκονται στο αρχείο Makefile και εκτελούνται με την εντολή make.

Εντολές στο Makefile:

```
flex -o simple- flex -code.c simple-flex-code.l

gcc -o simple- flex -code simple-flex-code.c

./simple-flex-code
```

input.txt output.txt Είσοδος και έξοδος Αρχεία: input.txt & output.txt Δημιουργήσαμε το αρχείο input.txt το οποίο περιλαμβάνει ένα σύνολο έγκυρων και μη περιπτώσεων για την δοκιμή κάθε κατηγορίας token της γλώσσας uni-C.

Input.txt

// Basic integer values

100

-100

0

1234567890

// Hexadecimal and octal numbers

0x1a2b3c

0XABCDEF

0755

0123

// Floating-point numbers

0.123

123.456

-0.987

-.1234

1234.

12e34

12.34e+56

-12.34E-56

1E10

// Keyword

break do if sizeof case double int struct

func else long switch const float return void continue for short while

// Operators

*= += -= *= /= == != > < >= <= && || ! % & | ^*

// Identifiers

identifier

_id123

abc_def

_123abc

// Strings with escapes and quotes

"Hello, world!"

"Escape sequence: \n \t \r \\ \""

""

// Comments

// This is a single line comment

*/**

This is a multi-line comment.

It spans multiple lines.

**/*

Άπο το παρακάτω input αρχείο και τον κώδικα παραγέται το output.txt

Output.txt

Line=2, token=INTEGER, value="100"

Line=3, token=INTEGER, value="-100"

Line=4, token=INTEGER, value="0"

Line=5, token=INTEGER, value="1234567890"

Line=8, token=INTEGER, value="0x1a2b3c"

Line=9, token=INTEGER, value="0XABCDEF"

Line=10, token=INTEGER, value="0755"

Line=11, token=INTEGER, value="0123"

Line=14, token=FLOAT, value="0.123"

Line=15, token=FLOAT, value="123.456"

Line=16, token=FLOAT, value="-0.987"

Line=17, token=FLOAT, value="-.1234"

Line=18, token=INTEGER, value="1234"

Line=19, token=FLOAT, value="12e34"

Line=20, token=FLOAT, value="12.34e+56"

Line=21, token=FLOAT, value="-12.34E-56"

Line=22, token=FLOAT, value="1E10"

Line=25, token=KEYWORD, value="break"

Line=25, token=KEYWORD, value="do"

Line=25, token=KEYWORD, value="if"

Line=25, token=KEYWORD, value="sizeof"

Line=25, token=KEYWORD, value="case"

Line=25, token=KEYWORD, value="double"

Line=25, token=KEYWORD, value="int"

Line=25, token=KEYWORD, value="struct"

Line=26, token=KEYWORD, value="func"

Line=26, token=KEYWORD, value="else"

Line=26, token=KEYWORD, value="long"

Line=26, token=KEYWORD, value="switch"

Line=26, token=KEYWORD, value="const"

Line=26, token=KEYWORD, value="float"

Line=26, token=KEYWORD, value="return"

Line=26, token=KEYWORD, value="void"

Line=26, token=KEYWORD, value="continue"

Line=26, token=KEYWORD, value="for"

Line=26, token=KEYWORD, value="short"

Line=26, token=KEYWORD, value="while"

Line=29, token=OPERATOR, value="="

Line=29, token=OPERATOR, value="+="

Line=29, token=OPERATOR, value="-="

Line=29, token=OPERATOR, value="*="

Line=29, token=OPERATOR, value="/="

Line=29, token=OPERATOR, value="=="

Line=29, token=OPERATOR, value="!="

Line=29, token=OPERATOR, value=">"

Line=29, token=OPERATOR, value="<"

Line=29, token=OPERATOR, value=">="

Line=29, token=OPERATOR, value="<="

Line=29, token=OPERATOR, value="&&"

Line=29, token=OPERATOR, value="||"

Line=29, token=OPERATOR, value="!"

Line=29, token=OPERATOR, value="%"

Line=29, token=OPERATOR, value="&"

Line=32, token=IDENTIFIER, value="identifier"

Line=33, token=IDENTIFIER, value="_id123"

Line=34, token=IDENTIFIER, value="abc_def"

Line=35, token=IDENTIFIER, value="_123abc"

Line=38, token=STRING, value=""Hello, world!""

Line=39, token=IDENTIFIER, value="Escape"

Line=39, token=IDENTIFIER, value="sequence"

Line=39, token=STRING, value="""

Line=40, token=STRING, value="""

Γενικές παρατηρήσεις/Σχολιασμός αποτελεσμάτων :

- Σύμφωνα με την δήλωση των tokens , το σύμβολα @,#,$.. θα αναγνωρίζεται πάντα ως UNKNOWN TOKENS
- Η μεταγλώττιση των αρχείων με κώδικα flex και των ενδιάμεσων με κώδικα c, που παράχθηκαν, έγινε με επιτυχία και χωρίς την ανάγκη αντιμετώπισης τυχών προβλημάτων.
- Οι καθώς και οι δοκιμές για το το κωδικά από το input file ήταν επιτυχές και δεν είχαμε κάποιο προβλήμα.
- Ο κωδικάς δημιουργήθηκε και γινανέ δοκιμές στο seed Ubuntu 16.04 , δίοτι είχαμε προβλήμα εγκατάστασης στα windows

Αρμοδιότητες Μελών:

Για την εκπόνηση της συγκεκριμένης εργασίας υπήρξε δια ζώσης και εξ' αποστάσεως συνεργασία μεταξύ των μελών της ομάδας, που αναγράφονται παρακάτω. Αναλυτικότερα, τα εξής μέλη ασχολήθηκαν με το σύνολο των τμημάτων της εργασίας:

Για το Α3 της εργασίας δουλεψάμε ολοί μαζί πανώ στα ίδια κομμάτια της εργασίας δήλαδη συγραφή κωδικά flex , αρχείον είσοδου και συνταξη τεκμιριωσής , καθώς και την τελική παροδοσή της εργασίας έτσι ώστε όλη η ομάδα να έχει πλήρη γνώση του flex και της λεκτικής αναλύσης.