COMPILER DESIGN

47	Date//
40	Page

UNIT - 1 INTRODUCTION TO COMPILING & LEXICAL ANALYSIS

\odot	Introduction to compiler -
	Combiles is a mogram which takes one language (rounce
	(human) as input and translates it into an equivalent another
	Compiles is a program which takes one language (source program) as input and translates it into an equivalent another language (target program)
	SOURCE PROLATE COMPILER TARGET PROLARM
	ERROR MESSAGES

Major Data structures in a compiler

Tokens - It represent banis program entities such as identifies,
Literals, reserved words, operators, delimiters etc.

Syntan Tree - It is generated by the parker. It is usually constructed as a standard pointer-band structure that is dynamically allocated as parring proceeds.

Syntal Table - It keeps information associated with all kind of identifies. Eg-constants, variables, functions, parameters, types, fields etc.

hiteral Table - It stores constant and things used in a program.

Orick insertion and lookup are exertial. Delition is not necessary.

Bootstapping and Porting

Bootstapping is a process in which simple language is used to translate more complicated program which in turn may handle for more complicated program. This complicated can juither handle even more complicated program and to on.

constraints. Hold the date of various stages

Temporary files - Used historically by old compiles due to memory

1- chagram A compiler written in language H (for hort

[5] T >> language) that translates language S (for house

[H] language) into language T (for target language).

Mycompanion

	Analysis of the source hu	gram contains three slips that is				
, °	Analysis of the source program contains three steps that is henrical analysis, bystan analysis and femantic analysis. Intermediate					
-	code is generated from the input touse program.					
7		s one involved that is intermediates				
	code generation, code generation	in and code ofteningation.				
	7	A CONTRACT OF THE STATE OF THE				
2	Vanous phans of a compiler -	0				
	Source Cock	Eg - a = b + c * 60				
	SCANNER	1 EMTAN. ALIMANIA				
·.	(LEXICAL ANALYZER)	LEXICAL ANAYYZER				
en i gent en		7				
	Synde Tokens	id1 = id2 + id3 * 60				
	PARSER	SYNTAX ANALYZER				
-	(SYNTAX ANALYZER)	= J				
LITERAL	S. t. Tage	ide of the				
TABLE	Syntan Tree	id ₂				
	SEMANTIC	J id3 60				
17 86.1 (8)	ANALYZER	SEMANTIC ANALYZER				
SYMBOL	Annotated Tree					
TABLE	THITERME DIATE	+				
	INTERMEDIATE CODE GENERATOR	id2				
	ODECUTE AND ADDRESS OF THE PARTY OF THE PART	ids int to float				
ERROR	9 ntermediate Code	Good Control (Discourse)				
HANDLER	THEFT CODE OPTIMIZER	ty: = inttofloat (60) ty!=id2+t2				
	CHARTER CHARTER	$t_1! = id_3 * t_1, * id_1! = t_3$				
		CODE OPTIMIZER				
	Target Code	t1:= id3 * 60.0 id1:=id2+t1				
	GENERATOR	CODE GENERATOR				
	GREENERATOR GREENERATOR	4				
	mycompanion Target Code	ADDF R ₂ , R ₁ MOVF Id ₂ , R ₂ MOVF R ₄ , id ₁ MOVF Id ₂ , R ₁				

4	
	1) henical Analysis (Scanning) -
	It is a phase of completion is which the complete
	source code is tranned and source program is broken up into
4	group of strings called token. A token is a regrance of characters
	having a collective meaning.
	(2) Syntan Analysis (Parsing) -
1	
-	growfied together to form a hierarchical thucture (called frame
1	tree or syntan tree.
1	3) fran Semantie Analysis -
$\frac{1}{1}$	
	It determines the meaning of the source string of like matching of parenthens, checking the proper of operation etc.
+	(4) Intermediate Code Generation
1	oche can be early converted to target corbe. It can represented as
	three-address code
	(5) Gode oftimization -
	It improves the intermediate code which reduces memory
-	consumption and have a faster enecuting code which improve the
	suntino of the target program.
-	(6) Code Generation -
	Tanget wide is generalled as assumbly unde or machine wide
	· ·
	Input Buffering -
	The lenical analyzes reanothe input string from left to night. One character at a time. It was two points train pointer (bptr) and forward pointer (fptr) to keep track of the posterior of the input reame byth
	One character at a time. It was two points train pointer (bptr) and
	forward points (fptr) to keep track of the parties of the input reams
-	in+ i, j; i=i+4; i= i+4;
	tetr Iritial Configuration
	Mycompanion (or comma, or operation)

	henemes - Sequence of characters in the source program that are
	matched with the pattern of the token.
	matched with the pattern of the token. eg - int, i, hum, ans, choice;
	-loken bptr
	end of leneme. (int) i, j; i = i+1; j = j+1;
	end of leneme. (nt) i, j; i = i+1; j = j+1; fetr
-	Two methods are used -
1	→ One briffer schume -> in t i= i+1
	> Two heffer scheme -> [in + i = i + 1] buffer 1
	; j =] + 1; eot buffer 2
	If leneme is short we use on can use one hiffer wheme het.
	if leneme is long we have to use two hiffer scheme in which the
\ . :	of current heffer is reached the other heffer is filled.
	If the lingth of leneme is longer than the lingth of the hiffer them the input can not be scanned completely.
	eof is introduced at the end of with the hiffers is called
	sentinel which is used to identify the end of hiller
-	Sentinel which is used to identify the end of hiffer. Code for input hiffering -
2	if (tptr = = eof (buff1)) /* encounter end of tirst butter*/
. 37 ,	/* Refitt bufter2 */
	fptr++;
	8
	else if (fptr = = eof (butt2)) / = encounter end of second butter */
	?
	/* Refill buffer 1*/
- 2 -	fptr++;
÷	2
	clse if (fptr == eof (input))
	return; /* terminate scanning */
•	else
	fptr +4 /x still remaining input has to be scanned */
	<i>wy</i> companion
	<u> </u>

					<u> </u>			
2	Specification of tokens-							
	To specify tokens regular confromens are used When a fattern							
	is matched by nome regular enformans them taken can be recognized							
_	P & VILANISA	A CHANDE ENABLES	acu. ~		·			
	0	String is a	collection	of finite hum	nter of a	lfhabets .	or letters	•
	The st	rings one sy	nomymo	usly called as a	abrem			
	-	Collection	of shing	is called the	e langue	34		
	[5]	-> hength	of a thin	g E→ confit	thing	φ → confi	y whole	hings
	Operation	in en hangu	4. -	<u>d</u> . (7 -0	<u> </u>	· · · · · · · · · · · · · · · · · · ·	0
	Un	ion (11 U	L2) , C	motenation (L	1.12)	Kleen Clo	rune (L*	<u>, </u>
	Ion	MAC CUPYM						
==	Ala	nguage dev	wied by	regular enfrim	vone is d	and to be a	regular.	ret-
	1 171 / 1	CENTRAL ALLES ALLES AND A SECONDARY				and the second s		
	eg -	Regular	Enpremo	n for identifier	. ⇒ lett	in (letter	+ digit)*	
	*** Ur	repraction is	formation	n for identifier	refrient	d by rigu	la enfre	man'o
3	Recogni	tion of Toke	mo -		* * * * * * * * * * * * * * * * * * *			
	Token R	purentation >	Token Ty	he Token value	2			
4	l			token Token va			regarding	Tokens.
		_		reads the infu	•		- 0	
	symbo	table for t	Okens.					
	e g -	TOKEN	CODE	VALUE	TOKEN	(ODE	WLUE	
	0	if	1	_	1=	7	5	
		હા કલ્	2	, ,	(· ·	8	1	
		while	3	,)	8	2,	
		for	4	-	+	3	1	
		identitier	5	Ptr to symbol table	-	3	2	
		constant	6	Ptr to symbol fable	=	10		
		<	7	1	if	(a<10)		
	<= 7 ≥ i=i+2;							
		>	7	3	else			
	>= 7 4 i=i-2;							
	mycom	npanion			Æ			
	1.7			•	74			

statute statut					
	Ou	lenical analyzer	will generate	. follow	ing token stream.
	Our lenical analyzer will generate following token stream. 1, (8,1), (5,100), (7,1), (6,105), (8,2), (5,107), 10, (5,107), (9,1), (6,110), 2, (5,107), (9,1), (6,110), 2, (5,107), (9,1), (6,110), 2, (5,107), (9,1), (6,110), 2, (5,107), (9,1), (6,110), 2, (5,107), (9,1), (6,110), 2, (5,107), (9,1), (6,110), 2, (5,107), (9,1), (6,110), 2, (5,107), (9,1), (6,110), 2, (5,107), (9,1), (6,110), 2, (6,110), (9,10), (9				
		(5,107) (9,2) (6,110		t de la companya de l	
	G	oneshouling symbolt	able for ider	tipis à	nd constants will be -
		LOCATION COUNTER	TYPE	VALUE	
,		100	IDENTIFIER	a	y sa war ing manifest the same
		105	CONSTANT	10	
:		107	IDENTIFIER	- 1000 P	
3 20 1 3	100000 pp	110	CONSTANT	2	
	~ , .			\$ 1.	
9	Bu	och schematic of le	mial analyze	<i>∨</i> −	
		0	t buffer	and the second	
		ĻE	XEME		
		Lexical Analyzer	1		v ,
3 . X		FINITE STATE	FINITE A	UTOMATA	A Gameson
age of the second secon	No.	MACHINE	SIMULA	TOR	
nay		¥.		Patter	ns
nguwasi Ayang da da Milian		,	PATTERN M	1 A TCHING	ALGORITHM Committee
pparter-section-sectio	relation and market relationships from		:		3
bride Anderson	a-ministratur v pillim ministratur propi		T	DKENS	
neducie de distribucion militari	Tz	144			
3	To	iannitión diagiam f	or Identifier	- R.E.	= letter (letter + digit)
			2 letter or digit	•	Section 19
	sta	So letter (51			A COLUMN
	T	rannition disciam !	Lor constant -		
	===		Control of the Contro		git (.) digit E(+/-) digit +)
	chard	Odiait		Qdigit	diait
	Siur	> (So) digit (S1)	S2) digit	(S3) E	>(Sy) +1- (S5) digit (S6)
	der stedenbliken			000000	

My companion

6	LEX (henical Analysis Generator) -
	It is a UNIX utility. In LEX tool, dengaing the rigular
	enpremons for corresponding tokens is a topical task
	LEX à une to unite specification file with . I (dot L) entennén
	b hEX converted 1 to c where is a a figure which is cotually
	a lenical analyzer program
	M LEX spicification, LEX lex. yy. c
	La file X. 8 compiler hunical analyzer program
	Porcy c
	lexigg.c C a out 1/16
1 1/2	a lenical analyzer program LEX spicification, LEX ien. yy. c file x. l compiler herical analyzer program Len: yy. c COMPILER Enecutable program COMPILER
	Input things a. Out Stream of tokens
	from source program tokago
	> (x.1) Specification file stores the regular Entremons of or the tokens
	> (x.1) Specification file stores the regular commons of or the tokens > lengy. Consists of tabular representations of the transmittin diagrams
\longrightarrow	LEX program connité of three parts -
-	6% 8
	DECLARATION SECTION
	\ °\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\
	1. % %
	RULE SECTION
	% %
	AUXILIARY PROCEDURE SECTION
	- Declaration oction > diclaration of variable content
	Rule fection - Combb of rigular enformin with arounder oction.
	Ry + Ry Eaction, 8
	Auniliany procedure rection -> Required procedures are definid
1	Mycompanion
11	

	Eg &-
	4 % &
	°/ ₃ }
	76 % 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
*	"Rama"
	"Seeta"
	"Geeta"
	"Neeta" = printf ("In Noun");
	"Sings"
	"dan ces" Rule 2
	"eats" printf ("In Verb").
	9, %
	main ()
	{
	yylen/); Routine, definis in yn len, yy c program.
	Scanner starts scanning the source program
	int yy wrop() - called when reanner encounters and of file.
. :	3
±1 4 ± 14 149	return 1;
	3
	Following commands one used to sun the len purgram n. I in UNIX.
. d	de en x.1
	& cc len, yy. c. co. , e de la server de la server.
	& sola out
	After entering this commands a blank space for entering input
	gets available. There we can give some valid input.
	Romaneats
	Noun
	ver b
	Then free either control + c or control + d to come out of the output

My companion

->	LEX actions -
	(1) BEGIN - It indecates the start state.
- 11	(2) ECHO - grenito the input as it is.
11	(3) gytent - When lever matches or recognizes the token from input
	token then the leneme is stored in a null terminated shing called gyters
	(4) yyin - Standard input file that stones input wince program
	(5) yyleng - stores the length of the input string.
	(6) gylen()
	(2) gywrep ()
	Derign of Lenical Analyzer Generator -
	Two of approaches -
	· Pattern makking using NFA
	· Uning DFA for linear analyzer
	Pattern matching using NFA
: : : :	
P 5	String Patterno Regular Regular Regular Analyzes String Patterno Enformano NFA DFA Generales
n l	Advantages
	(1) herical analysis and hinton analysis are who aled out which
	(1) himical analysis and hyptom analysis are separated out which reduces the budon on farming phase.
0	(2) Compiler efficiency get increased due to reparation
8	gruce of linical analyzer -
	(1) Ambiguity of words (fame word can have deffect meaning) (2) hookahead (identifying limits for kentence houndaries by - Dr. Hr.
1	
11	ı

mycompanion