Compiler Design.

Hulf Past

High level language -> developed by humans by meadable by humans.

Machine level language -> Not readable by humans.

Compiler:

It is a soffware which converts high level large to Low Level larguage.

Source program

Compiler Forget program

Error

Types of Compiler:

Single pass:

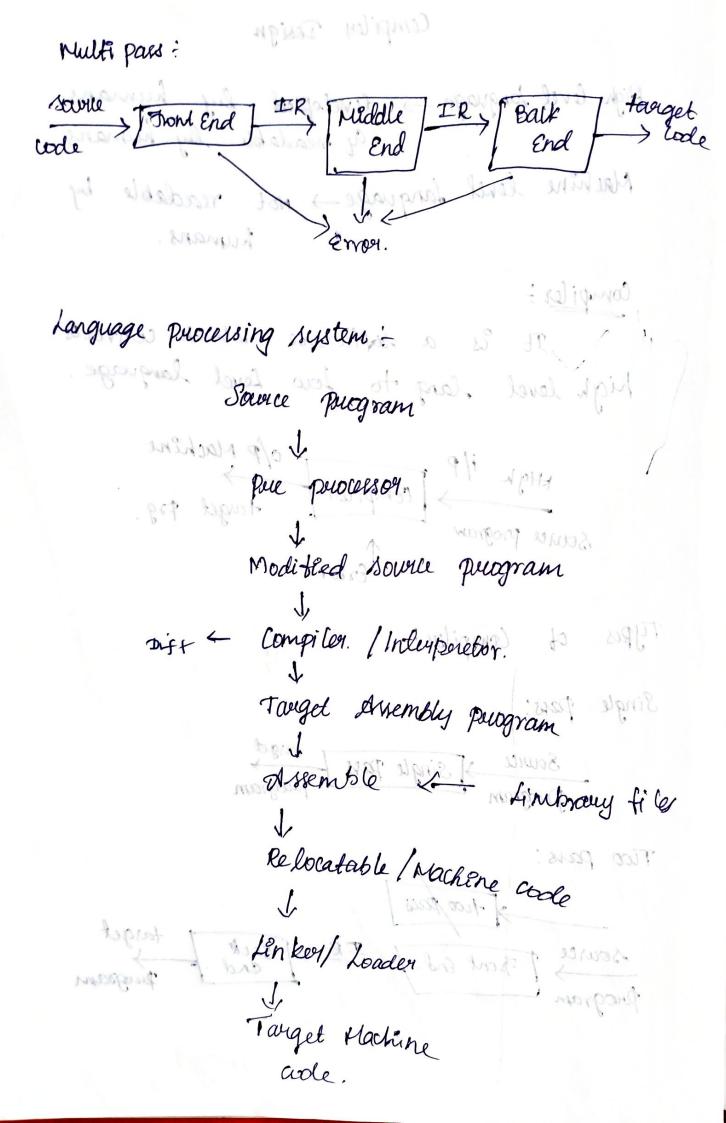
Source Single Pars target purgram

Two pars:

source Front End IR Back target program Pourgoam

Tayget Hadring

arde.



cousins of compiler: retal . count + mail + 10 -> Compiles haial analyer -→ Loaden Interq: prector sustained -> Assembled. . I griste de a maissex preparce uses - days - ladar Phases of compiler: 1. operator 1 operator Bowce Program Jexical Analyses - supplement rotys > Syntax Analyses error detection symbol Table cy Handling. Bemantic Analyses/ Management Intermediate generator I code optimization 3) Senantic -'code generalor 1) Wemidiate : Lang. dep (Language dependent) Machine Mac. dep 10 Synthesis part Code op Himi zates grandless (1) sumbly (sigh

sound by landing Total = count + male \$ 10

Lexical analyer:

grow delection

of Handbirg.

It stan source purgram and gives requence of strings.

Total = count + rate as 10 Jopenalor Loperator value.

Syntax Analysen: (pour tree)

egishpour somble -btal

Symbol rable

Mary Trens

Synthesia jart

MANAGEMENT

3) Semarlic - check for the rules!

4) Intermediate ade generator:

propriation (frequencing) Benting tible bust position asky

5) (ade op 4° mi zalos

cade generator (Assembly Gorgs

a: b+c * 60. Optimized (03) anost * 80: -Lexical snalyer 0.00 mobi en 18 1 - 101 - 151 id1 = ids + ids + 60. Syntax a nalifeer MOV MOVF ids, Re LUM MUJE 1160, Re MOVF ids, E) ADDE P1, R intermediate vode generalor. ts = int to float (60) t? = id8 x t8 23 = id2 + t2

optime Le t1 = id3 * float (60) ferteal strabyes fi= ids & 60.0 ide = ide + t1 id1 - ids + lds + bi I code generate Syntan andyron MOV Morfida, Re MUL MULF #60, Re ADD MOVF ide, RI ADDF RI, Re total = court + rate to 70. 00 abi Lexical rotys Analysa id1 = 102 + 10 3 x 70 Pd2 id3 70 9d2 (od) Jouls of this est Pd3 PN to feat (7.0) 121

In code generator. +1 = PN to float (70) t2 = Pd3 # +1 +3=1,9d2 + +2 tu-, id8 on 70.0. de 170, id3 NOVE +170, id3 id1 - ... code optivities id1 = id2 + +1 (m) 1- +in popf pr. 22 Code generator ind vis in company, reducerto MOVE id8, Ro. 10 Mil MOVE 122, ARI M refer of confiler ADDF RI, Re Soll and, polin in single pars. language prouving model. religiona de entros. frembler consiler books - dagments

(Lexical Analyer) -> Wht 4 DEA - NEA LOIDS AND Loroce.

DEA to NEA 2 - Loron Loroce. legular expression-) NFF -> Regular expression. -> Regular expression to NFA to DFA NEA. kleens theorem 8-NFA DFA - Minimization. identifies recogon ze pandrathut negular expression unit-I (PT) Larg processing redd. duembler, preprocesser, un ter, loader, diagram (integtes, compaire diff) locical analysis convertion (phases of conficer (A=B+C060) Types of Compiles Backend, functioned & dagreims, Language processing model Cowin of compiles -Avender, compiler boder -> déagranation representation

unit-III Saniatic analyser -> Grammer, CFG Convertion - left, right most derivation - parry tree from Given grammer or input string. -) TOP, down paring Bottom, down parrieng. -> converting data from one formed to emother. Grammer - und to derive some language. Non al reminal production reminal production Ly process of converting or transfer for relact appropriate data from one part or format to production rule another. Ly powers of converting or transfering another.

Back macking

Back macking

Back macking

Back macking

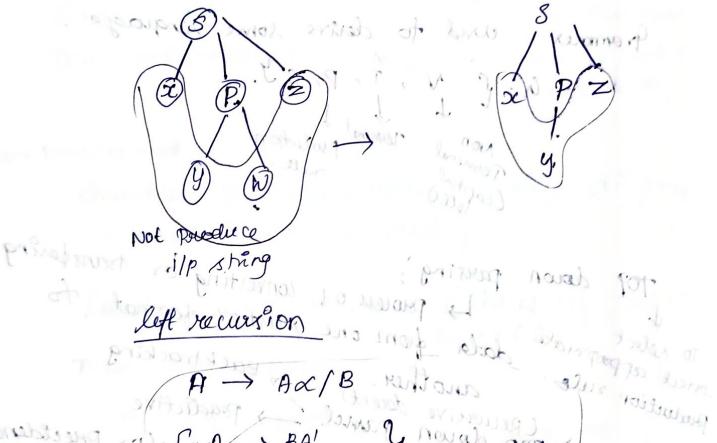
Powdictive

Prop cloud, panel

Powdictive > Bottom up, pourel. > operator Precidence SIR LRILALIN STATE Da (Dale (implementally proces) 11 BA E-A 3 - > Pal 6 3/00 = 9 1100 6 1A

May and wife TOP down pancel (Back tracking) from noot to children or lead I tavay tree from Given grundles or input soing.

 $S \longrightarrow \chi P_Z$ I/P: MZ $P \rightarrow (yw)ly$.



S. A. SAILE J. Pucof

(I)al SA ABC A. -> AalAdle soln; S-) ABC

S -> Aalb P > bp' Al - a Alls (implementing proof) A-> Ad 16 A -> bal A' -> 6 A' /E