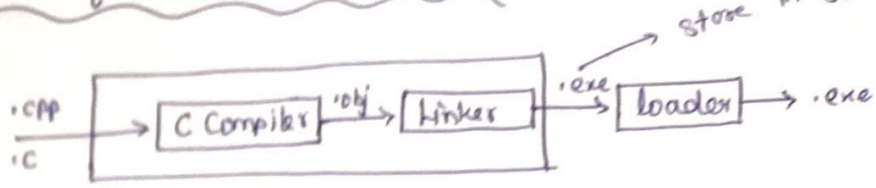
Compiler Design Lec-1

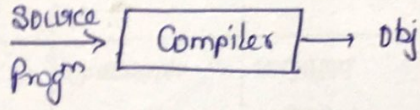
→ Process of C file Execution :-

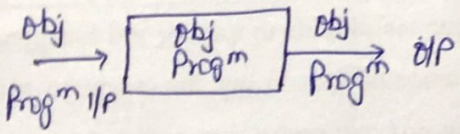


Translator is a progm that takes as I/P a progm written in one programming lang (source lang) and produces as O/P a progm in another lang (object or target lang).

→ Types of Translator :-

- Compiler → If the source progm is HLL such as FORTRAN, PL/1 or COBOL, obj lang is low level lang such as assembly lang or m/c lang then such a translator is called compiler.





- Interpreter
- Assemblers

→ Preprocessors

Why we need translators :-

With the m/c lang we must communicate directly but such a lang is terribly tedious.

- All the operands and operations must be specified in numeric code.
- Impossible to modify in a convenient manner.

Syllabus

- lexical Analysis
- syntax Analysis
- Intermediate code generation
- Syntax Directed translation
- Run time Env and Code Generation
- Code Optimization

Unit-1

Introduction to Compilers

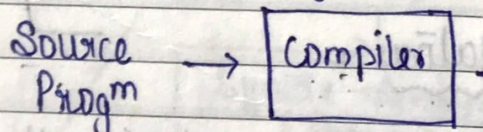
mnemonic names ← Assembly lang
M/C lang

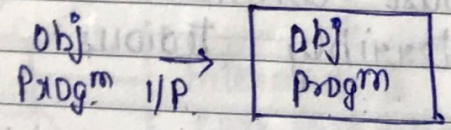
A Translator is a progm that takes as input a progm written in one programming lang (source lang) and produces as o/p a progm in another lang (object or target lang).

Types of Translator :-

① Compiler →
If the source progm is high level lang such as FORTRAN, PL/1 or COBOL, obj lang is low level lang such as assembly lang or m/c lang, then such a translator is called compiler.

The source progm must first be compiled, that is translated into obj progm. Then resulting progm is loaded into memory and executed.





Imp. Notes

② Interpreter →
Transform a programming lang into a simplified lang, called Intermediate code, which can be directly executed using a progm called an Interpreter.

Compiler	Translator Interpretor
✓ compilers are larger	✓ Interpreter are often smaller. ✓ Interpreter execute line by line ✓ execution time of Interpreter is slower.

Assemblers → If the source lang is assembly lang and the target lang is m/c lang, then the translator is called an assembler.

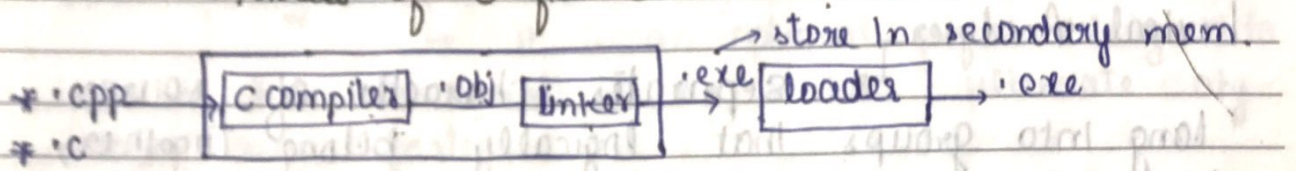
Preprocessor → The term preprocessor is sometimes used for translators that take progm in one HLL and convert into another HLL.

Why we need Translators:

With the m/c lang we must communicate directly but such a lang is terribly tedious.

Imp. Notes
✓ All operns and operands must be specified in numeric code.
✓ Impossible to modify in a convenient manner.

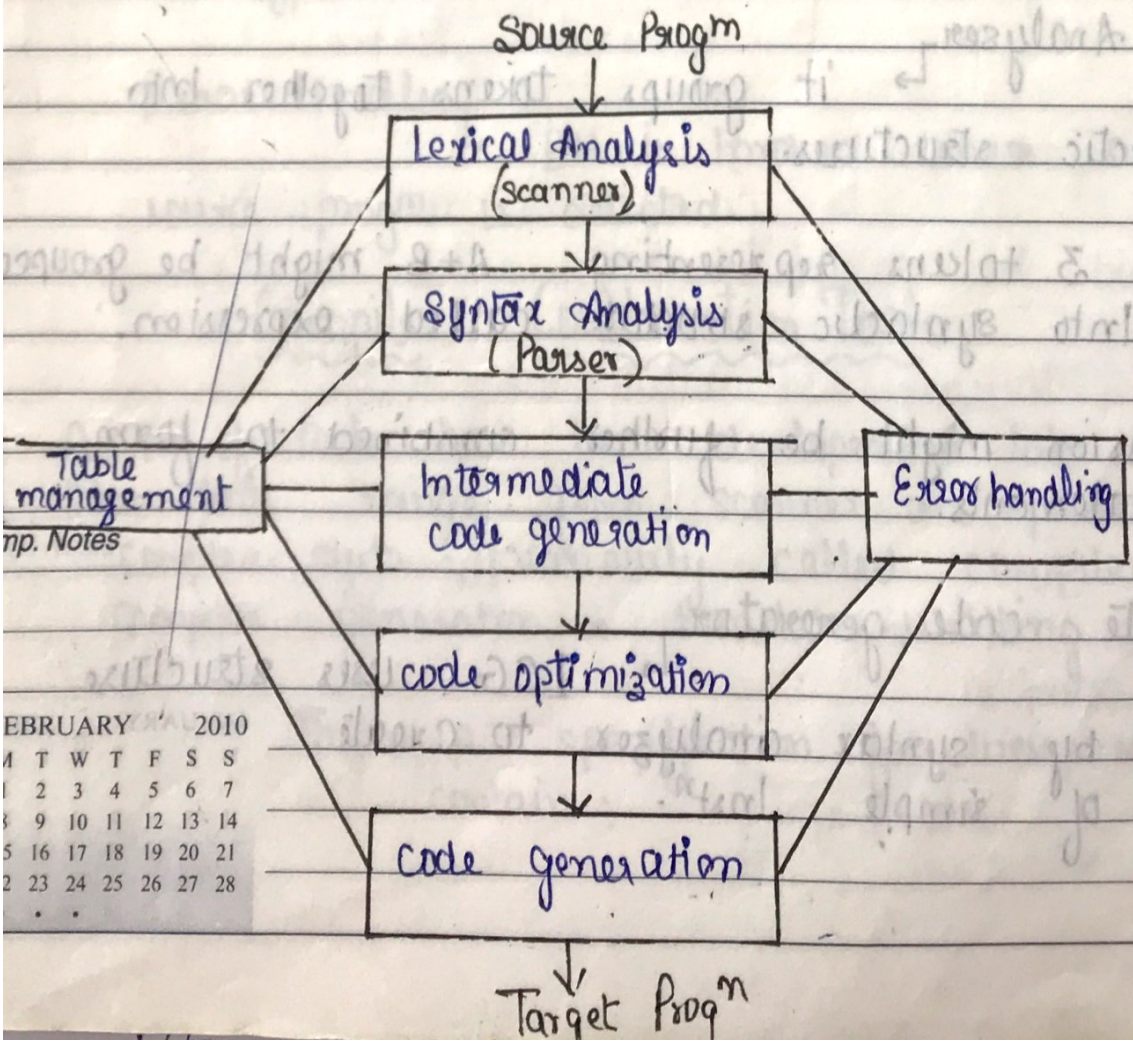
Process of C file execution :



Structure Of a Compiler

Compiler consist of 7 phases.

✓ Phase
↳ A phase is a stage or a step that takes as one form of source code and convert it into another form of code.



Lexical Analyser → separates character of source lang into groups that logically belong together, these groups are called tokens.

The usual tokens are :-
- keywords DO, IF
- identifiers X, NUM
- Operator symbol <=, +.
- Punctuation : comas (,), () Paranthesis.

O/P of Lexical Analyser is a stream of tokens which is pass to the next phase.

Syntax Analyser → it groups tokens together into syntactic structures.

for eg → 3 tokens representing A+B might be grouped into synlactic structure called expression.

Expression might be further combined to form statements.

Imp. Notes

Intermediate code generator → ICG uses structure produced by synlax analyzer to create stream of simple Instn.

Code optimization → it is an optional phase designed to improve the intermediate code so that ultimate obj prog^m runs faster.

Code Generation → produces obj. code by deciding on the memory loc for data, selecting code to access each data and selecting reg in which each computation is to be done.

Table management → it keeps the track of names used by prog^m and record essential i/f about each such of its type.

Error Handler → EH is invoked when a error in source prog^m is detected.

Compiler Construction Tools :-

A no. of tools have been developed to help construct compilers. These tools range from scanner and parser generator to complex sys^m, variously called compiler-compilers, compiler-generator or translator-writing sys^m.

Imp. Notes

The i/p specification for these sys^m may contain :-

- a description of lexical and syntactic structure of source lang.
- a description of what OIP is to be generated for each source lang construct.
- a description of target m/c.

A no of useful compiler-compilers exist, but the chief problem is that there is a trade off b/w how much work the compiler-compiler can do automatically for its user and how flexible the sys^m can be.

Principal aids provided by compiler-compilers are:-

- Scanner Generator → generate tokens.
- Parser Generator → parsing is done can be a great aid in organisation of entire compiler.

  Adv → Increase reliability.

SUNDAY 17

3. facilities for code generation → The generation of Intermediate, assembly or obj code is provided by compiler-compiler.

Imp. Notes

Why Compilers?

Every computer system has a m/c lang and prog^{m} in m/c lang can be executed by the h/w of the computer sys^{m}.

→ Unless a prog^{m} is written in the m/c lang of the comp. sys^{m}, it can't be executed by the h/w of the sys^{m}.

→ All the s/w running on all the computers is written in some prog^{ming} lang, before a prog^{m} can be run, it first must be translated into a form in which it can be executed by a computer.

→ The s/w sys^{m} that do this translat^{n} are called compilers.

Compiler Design

two majors parts of compiler

Analysis
Intermediate represent^{n} is created

- lexical Analyser
- syntax (gr. to be followed)
- Semantic (holding some meaning)

Synthesis
Equivalent target prog^{m} is created

- Intermediate code generator
- Code generator
- Code optimizer

Source progm → [Compiler] → Target Progm

↓ error

Imp. role of compiler is to report any error in the source progm that it detects using the translat^{n} process.