Intro

What is compiler

> ℹ Info
>
> Computer program which translate a program in a source language into a equivalent program in a target language.
> source language : Python, C, Java, C++, Assembly
> target language : mostly Machine/Intermediate code

Overview of compiling

> ℹ Info
>
> step 1 (Keyboard):
> Got a stream of characters
> step 2 (Scanner) :
> Separate stream of characters to stream of tokens
> step 3 (Parser) : (Context-free Grammar)
> Checking grammar produce parse/ab-syntax tree
> step 4 (Semantic Analyzer) :
> Type Checking + Produce Annotated tree
> step 5 (Intermediate code Genarator) optional :
> Generate Intermediate Code
> step 6 (Code optimazation) optional :
> Reduce InterCode size/make it more effective

step 7 (Code genarator) :
Generate Target Code
step 8 (Code optimazation) optional :
Reduce TargetCode size/make it more effective

Interested Knowledge

Info

scanner-intermediate code generator can be used for every compilers.
code optimizer/generator required for each target language.

One of the popular intermediate code is
Three-address code = x = y op z

Code-optimization/improvement
Could be done after->
1.Sementic Analysing : performed on parse tree
2.Inter code gen : performed on intermediate code
3.Target code gen : performed on target code

Error Handling

Info

Error found during compilation = static error
Error found during execution = dynamic error

Data Structure

Symbol Table

ℹ **Info**

Mostly use hash table for efficiency
Store Information associated with identifiers
Identifier are names of variables like constant, function, data type, etc.

Information associated with different type of identifier can be different->
function : name, address, return type, parameter
variable : name, address, array size

Will be accessed in every phases of compilers
Scanner, Parser and Sementic analyzer->
Put names of identifiers into symbol table.
Sementic analyzer also store more info ex : data-type
Both Intermediate/Target Code Generator/Optimizer->
Use infomation to generate appropriate Code.

Literal Table

ⓘ Info

Store constants and strings in program for reusing.
Can be combined with symbol table.

Parse Tree

ⓘ Info

Dynamically-allocated, Pointer-Based structure
Node store pointer to information which stored in other data structure e.g. symbol table