

Assignment-5

Tarun Shuman
Ap18110010459
CSE-H

1) Describe the various phase of compiler and trace the program segment.
 $a := a + b * c * 2$ for all phases.

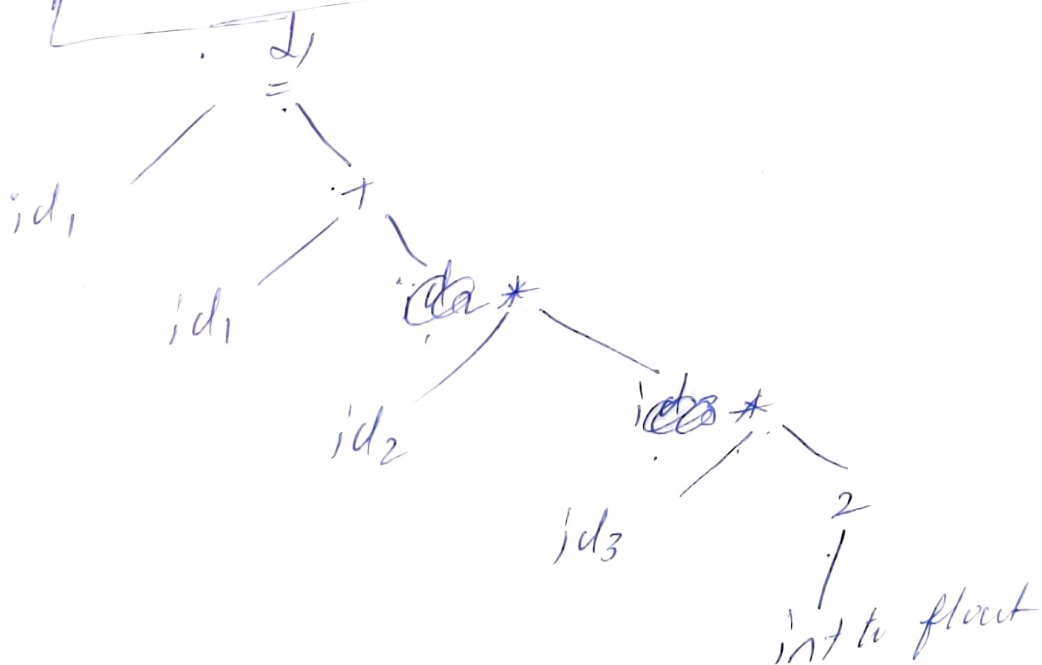
Soln-

$a := a + b * c * 2$

↓
Lexical analysis

↓
 $id_1 = id_1 + id_2 * id_3 * 2$

↓
Semantic analysis



↓
Intermediate code generation

↓
(P.T.O)

$$t_1 = \text{int to real}(2)$$

$$t_2 = \text{id}_3 * 2$$

$$t_3 = t_2 * \text{id}_2$$

$$t_4 = t_3 + \text{id}_1$$

$$\text{id}_1 = t_4$$

↓

[code optimization]

↓

$$t_1 = \text{id}_3 * 2.0$$

$$t_2 = \text{id}_2 * t_1$$

$$\text{id}_1 = \text{id}_1 + t_2$$

↓

[code generation]

↓

MOV id_3 , R1

MUL #2.0, R1

MOV id_2 , R1

MOV id_3 , R2, R1

MOV id_1 , R2

ADD R2, R1

MOV R1, id_1

Q2) Explain the role of linker, loader and preprocessor during execution program with neat diagram.

n) Linkers - Linker allow us to take a single ~~for~~ program from a several lines of relocatable machine code. These file may have been the result of several different compilation, and one or more may be library files of routine provided by a system.

Loader - The process of loading consists of taking relocatable machine code, altering the relocatable address and placing the altered instructions and data in memory at the proper location.

Preprocessor - It produces input to compiler.

They may perform various functions.

i) Macro processing: A preprocessor may allow user to define macros that are shorthand for longer constructs.

ii) File inclusion: A preprocessor provides the user with built in macro for construct like while statement or if statement.

iii) Rational preprocessor: - A preprocessor may include the ~~for~~ headers into the program text.

Source program

↓
Preprocessor

↓
modified

↓
Compiler

↓
Target assembly

↓
Assembler

↓
Relocatable M/C code

↓
Linker/Loader

↓
Absolute M/C code.

3) Distinguish b/w pass and phase?

Pass

→ pass is a physical scan over a source program. The portion of one (or) more phases are combined into a modules called.

Pass

→ Requires an intermediate file b/w two passes

→ splitting into more no. of passes reduces memory

→ single pass compiler is faster than pass

Phase

→ A phase is a logically cohesive that takes i/p in one form and produces o/p in other form

→ No need of an intermediate file

→ Reduces the complexity of the program

→ Reduction in no. of phases increases.