Pass - 2 Assembler.

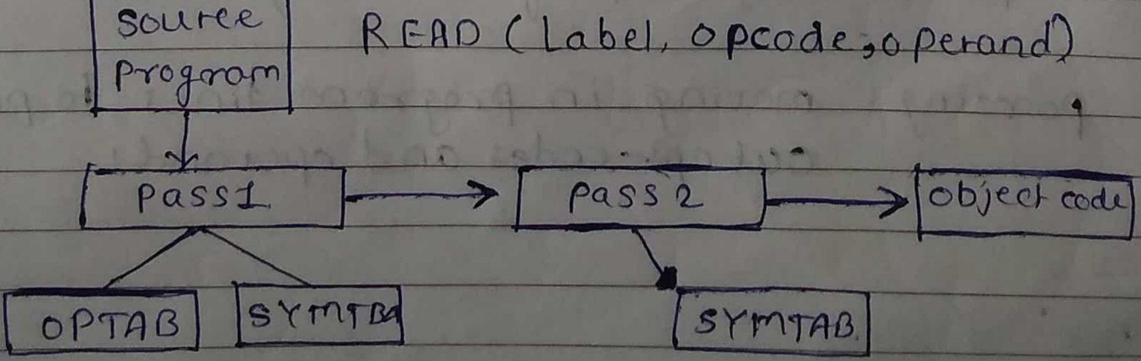
Page No.
Date

Aim : To design data structure for Pass-2 Assembler.

Problem statement - Implemented Pass-II of two pass assembler for peasdo-machine in Java. using object oriented features. The output of. assignment-I should be input for this assignment

Theory :
Two-pass Assembler.
The two-pass assembler performs two passes over Source program. In first pass, it reads the entier sources program, looking only for label. definitions. All the labels are collected, assigned. address, and placed in the symbol table in this. pass, no instructions as asssembler and at end of. Symbol table should contain all the In second pass the instruction are again read and are assembled using symbol tables.

A simple Two Pass Assembler Implementations.



Mnemonics and opcode mappings are referenced from here.	Label and address enter here	label and message are referenced from here.

Differences between one Pass and Two Pass. Assemiblers.

- A one pass assembler passes over source file exactly onces, in the same pass collecting the labels, resolving futures references and doing the actual assembly. The differences part is to resolue future label references And assembly code in one pass. The one pass assembler prepares an internmediate files, which is used as input by the two pass assembler.
- A Two pass assembler does twopasses over sources file. In the first pass all it does is looks for label definitions and introduces them in the symbol table.

A two-pass Assembler perform two sequential scan over source code:

Pass 1: Symbol and literals are defined.
pass 2: object program is generated.

parsing: moving in program lines to pull out op-codes and operands.

Data structures -

- Location Counter (LC) : points to next location where the code will beplaced.
- op-code translation table: contains symbolic instructions, their lengths and their-op codes
- symbol table (ST) : contain labels and their values.
- String Storage buffer (SSB): Contain ASCII character for the strings.
- forward references table (FRT): contains pointer to the string in SSB and offset where its value will be inserted in the object code.

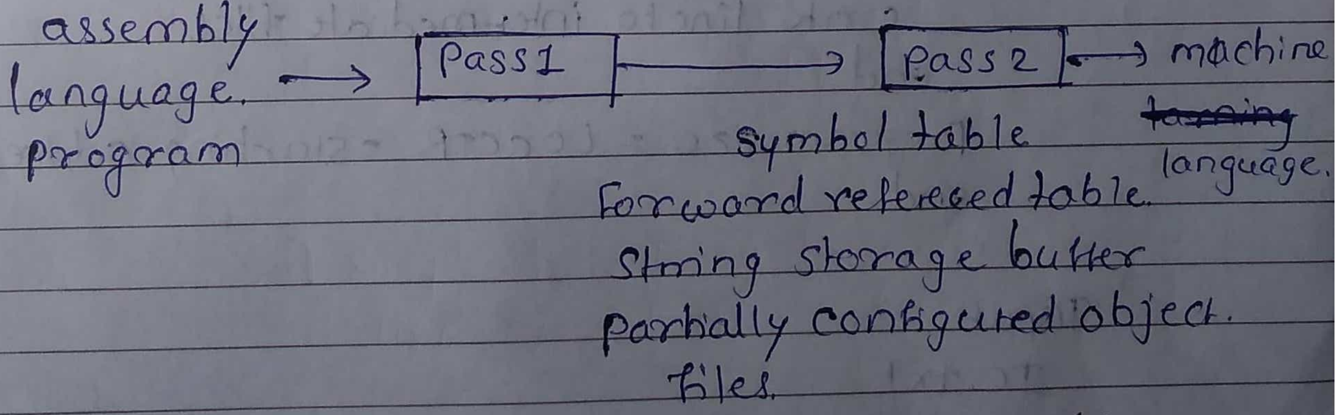

assembly language program
Pass 1
Pass 2
machine ~~lanaing~~ language.
symbol table
Forward refereced table.
String storage buffer
partially configured object files.

Fig. A simple two pass Assembler.

Algorithms.

Page No.	
Date	

```
begin;
    if starting address is given.
        LOCCTR = starting address;
    else
        LOCCTR = 0;
    while OPCODE != END do.          ;; or EOF.
        begin.
        read a line from the code
        If there is a label.
            If this label is in SYMTAB, then error
        else insert (label, LOCCTR) into SYMTAB.
        Search OPTAB for the opcode.
    If found.
        LOCCTR += N     ;; N is length of this instructions
    else if this is an assembly directives.
        update LOCCTR as directed
        else error
        write line to intermediate files.
      end
    program size = LOCCTR - starting address;
    end
```

Input

IC.txt

AD	01	C	200	
IS	04	1	L	1
IS	05	1	S	1
IS	04	2	L	2

MEERA
PAGE NO.:
DATE: / /

IS	04	3	S	3
AD	05			
IS	01	3	L	3
IS	00			
DL	02	C	1	
DL	02	C	1	
AD	02			

LITTAB.txt.

='4'	204
='6'	210
='1'	205

SYMTAB.txt

A	208
LOOP	203
B	209

POOLTAB.txt.

1
3.

Expected output:

```
200    04    1    204
201    05    1    208
202    04    2    210
203    04    3    209
204    00    0    204
```

205	00	0	006.
206	01	3	205
207	00	0	000.
208			
209			
210	00	0	001

Conclusion →

Thus wehave generated machine code for the source program.