MES Wadia College of Engineering Pune-01

Department of Computer Engineering

Name of Student:	**Class:**
Semester/Year:	**Roll No:**
Date of Performance:	**Date of Submission:**
Examined By:	**Experiment No: Part A-02**

PART: A

ASSIGNMENT NO: 02

AIM: Design suitable data structures and implement Pass-I and Pass-II of a two-pass macro-processor. The output of Pass-I (MNT, MDT and intermediate code file without any macro definitions) should be input for Pass-II

OBJECTIVES:

- To study the implementation of Macro of Two Pass Macro in Detail.
- To understand Handling of Macro Call and Macro Expansion.

PRE-REQUISITES:

1. Basic of Assembly Language programming

APPARATUS:

THEORY:

A. MACRO DEFINITION

A macro definition is enclosed between a macro header statement and a macro end statement. Macro definitions are typically located at the start of a program. A macro definition consists of.

- ***A macro prototype statement***
- ***One or more model statements***

- ***Macro preprocessor statements***

B. The macro prototype statement declares the name of a macro and the names and kinds of its parameters. It has the following syntax

<macro name> [< formal parameter spec > [,..]]
Where <macro name> appears in the mnemonic field of an assembly statement and
< formal parameter spec> is of the form
&<parameter name> [<parameter kind>]

C. M ACRO C ALL

- A macro is called by writing the macro name in the mnemonic field.
- Macro call has the following syntax.
- ***<macro name> [<actual parameter spec>[,..]]***
- Where an actual parameter resembles an operand specification in an assembly language statement.

D.DESIGN OF A MACRO PREPROCESSOR

- The macro preprocessor accepts an assembly program containing definitions and calls and translates it into an assembly program which does not contain any macro definitions and calls.

- The program form output by the macro preprocessor can be handed over to an assembler to obtain the target program.

Macro Processor Algorithm and Data Structures

The procedure DEFINE Called when the beginning of a macro definition is recognized,

- Makes the appropriate entries in DEFTAB and NAMTAB.

- The procedure EXPAND Called to set up the argument values in ARGTAB and expand a macro invocation statement.

- The procedure GETLINE Called at several points in the algorithm, gets the next line to be processed.

```
procedure DEFINE
    begin
        enter macro name into NAMTAB
        enter macro prototype into DEFTAB
        LEVEL  := 1
        while LEVEL > 0 do
            begin
                GETLINE
                if this is not a comment line then
                    begin
                        substitute positional notation for parameters
                        enter line into DEFTAB
                        if OPCODE = 'MACRO' then
                            LEVEL := LEVEL + 1
                        else if OPCODE = 'MEND' then
                            LEVEL  := LEVEL - 1
                    end {if not comment}
            end {while}
        store in NAMTAB pointers to beginning and end of definition
    end {DEFINE}
```

Pass 2 of Macro Processor – Processing for Calls and Expansion of Macro

1.Read the next source statement copied bypass 1.
2.Search into the MNT for a record and evaluate the operation code.
3.If the operation code has a macro name, go to Step 5.
4.Write the statement to the expanded source file.
5.If END pseudo-op found, pass the entire expanded code to the assembler for assembling and stop. Else go to Step 1.

6. Update the MDTP to the MDT index from the MNT entry.

7. Prepare the parameter (argument) list array.

8. Increment the MDTP by 1.

9. Read the statement from the current MDT and substitute actual parameters (arguments) from the macro call.

10. If the statement contains MEND pseudo-op, go to Step 1, else write the expanded source code and go to Step 8.

CONCLUSION:

QUESTIONS:

1. What is Macro Call and Macro Definition.
2. Explain Macro Definition table and Macro Name Table in detail.
3. Explain Advance Macro facilitates in detail.
4. Explain Handling of nested macro call
5. Explain formal , actual , positional and Keyword Parameters
6. Explain in detail what kind of Data Structure use for your pass II Code.