



## Assignment No: Group B :- 2 (Pass 1 Macroprocessor)

1) Title: Design suitable datastructure and implement Pass-1 Macroprocessor of a two pass Macroprocessor for pseudo-machine. The output of Pass-1 (MNT, MDT, and intermediate code file without any macro definitions) should be input for Pass-2 Macroprocessor.

2) Software/Hardware Requirement:

\* Software requirement:

- 1) Java Development kit.
- 2) IDE
- 3) OR Notepad.

\* Hardware requirement

1) Computer System.

Processor : i5 4th Gen

Ram : 8GB.

2) I/O peripherals like keyboard & Mouse

3) Monitor : 720p/1080p FHD/IPS.

3) Learning Objective.

1) To understand what is a Macro and Macro processor.

2) Identify and understand two pass Macroprocessor.



#### 4) Learning Outcome.

- 1) One should be able to understand what is a Macro and Macro processor.
- 2) One should be able to implement pass 1 Macro processor.

#### 5) Concept Related to Theory:

A Macro instruction is the notational convenience for the programmer. For every occurrence of macro the whole macro body or macro block of statement gets explained expanded in the main source code. Thus Macro instructions make writing code more convenient.

#### Salient features of Macro Processor:

- i) It represents a group of commonly used statements in the source programming language.
- ii) Macro processor replaces each macro instructions with the corresponding group of source language statements. This is known as the expansion of macros.
- iii) Using Macro instructions programmer can leave the mechanical details to be handled by the macro processor.



iv) Macro processor involves definition, invocation and expansion

### \* Macro definition and Expansion:

Line	Label	Opcode
5	COPY	START
10	RDBUFF	MACRO
15		
.		
.		
90		
95		MEND

Here,

#### Line 10

RDBUFF (Read Buffer) in the Label part is the macro or name of macro definition of the macro. 2 INDEV 2 EBUFADR are the parameters present in the Operand part. Each parameter begin with character 2.

#### Line 15 to line 90

From Line 15 to Line 90 Macro Body is present. Macro directives are the statements that make up the body of the macro definition.

#### Line 95:

MEND is the assembler directive that means the end of the macro definition.



## Macro Invocation:

Line	Label	Opcode
180	first	STL
190	loop	RDBUFF
15		
255		END

Here,

Line 190,

RDBUFF is the macro invocation or macro call that gives the name of the macro instruction being invoked and FI, Buffer are the arguments to be used in expanding the macro. The statements that form the expansion of a macro are generated each time the macro is invoked.



## 6) \* Algorithm for Pass-1 Macroprocessor.

- 1) Initialization of counters for MDT & MNT.
- 2) Read Next Instructions (and divide it into its various field as Label, mnemonic (opcode arguments).
- 3) Check for macro definition start  
if opcode = MACRO goto step 5 else to step 4.
- 4) a) write copy of instruction to output of Pass-1  
b) Check whether opcode = END or not.  
c) if OPCODE = \* END goto step 2  
d) if OPCODE = END goto Pass-2 i.e End of this algorithm for Pass-1.
- 5) Start of macro definition is identified. Now Pass-1 will process content of macro definition after pseudo op MACRO to MEND\*1
- 6) Process other instructions in macro definition including MEND.

7) \* The data structures associated with Pass-1 Macroprocessor.

### 1) MDT (Macro Definition Table)

Index	MDT-Instruction

### 2) MNT (Macro Name Table)

Index	Macro-Name	MDT-Index



### 3) Argument List Array (ALA)

x Index      Macro-Name      MDT-Index.

Index	Dummy Argument

### 8) Conclusion:

Understood the working of Pass1 Macro-processor.

### 9) References:

- 1) Geeks for Geeks.
- 2) SCRIBD
- 3) Youtube / Ganesk kakde.