Roll No. 31164 Date: 25/05/21

**Problem Statement:** Design suitable data structures and implement pass-I of a two-pass macro-processor using OOP features in Java.

**Objective:**

* Understand the internals of language translators.
* Handle tools like LEX and YACC
* Understand the operating system internals and functionalities with implementation point of view.

**Software and hardware requirements:**

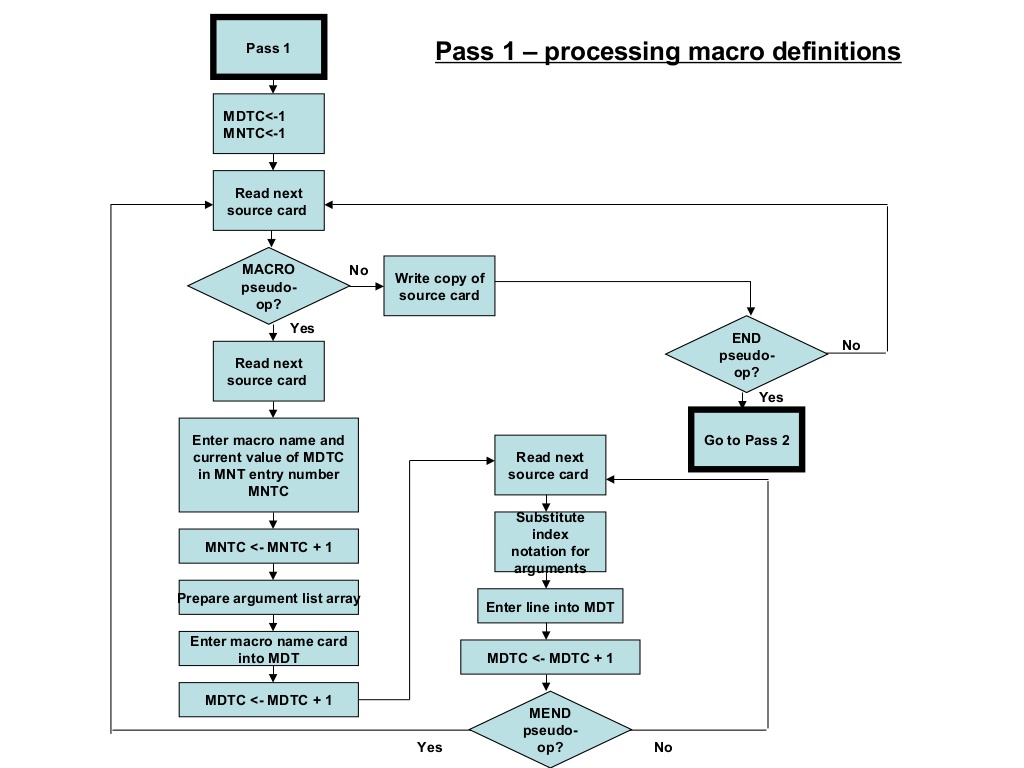
64-bit open source Linux (Fedora 20), Eclipse IDE, JAVA, I3 and I5 machines.

**Theory:**

Macro processing feature allows the programmer to write shorthand version of a program (modular programming). The macro processor replaces each macro invocation with the corresponding sequence of statements i.e., macro expansion.

* Tasks done by the macro processor are -
* Recognize macro definitions.
* Save the macro definition recognize macro calls.
* Expand macro calls.
* Tasks in pass I of a two pass macro processor
* Recognize macro definitions.
* Save the macro definition(Create MDT,MNT,ALA)Perform processing of assembler directives(e.g., BYTE, RESW directives can affect address assignment)
* Create intermediate code file.
* Steps to do /algorithm:
* Read .asm file.
* Create MNT and MDT.
* Create ALA.
* Create intermediate code file.

**State Diagram:**



**Conclusion:**

We successfully completed the implementation of macro statements using macro pass 1 and understood the OS internals and functionalities. We also handled tools like LEX and YACC.