Lab-06 A PL/0 p-code assembler

In this lab, we are going to implement an assembler that translate the PL/0 p-code to machine code.

1 Objectives

1. Construct a macro assembler iteratively.

2 Learning Outcomes

By completing this lab, learners should be able to

1. build a macro virtual machine iteratively.

3 An Assembler

An assembler is a simple translator that translate 1 assembly code into 1 machine code; hence, it is very straightforward to build. The p-code described in the previous lab is even easier as there is only one format for the assembly language. The format is shown below.

Function (F)	Level (L)	Value (V)

For example, a program to negate the number 15 may be written as follows:

Line	Assembly Code	Meaning
1	LIT 0 15	Store the value 15 onto the stack
2	OPR 0 2	Negate the value at the top of the stack
3	OPR 0 0	Stop the program

The corresponding machine codes for the above assembly codes are

000F E002 E000

4 Tasks

We will first build a naïve assembler that translate the naïve assembly code into machine code.

4.1 Phase I: Build a naïve assembler.

Write a program to read a program in p-code assembly language from a text file and produce the corresponding machine codes. Each instruction in naïve assembly language contains 3 part described above. All 3 part are separated by whitespaces. In addition, only one instruction per line is allows.

4.2 Phase II: Build a macro assembler

Enhance the naïve assembler build in phase I by allowing the assembler to translate both the naïve assembly code and the macro codes into machine code. The macro codes for this phase are defined below.

Macro assembly code	The corresponding naïve assembly code
ASSIGN V	LIT 0 V
ALLOC V	INT 0 V
JUMP V	JMP 0 V
JZ V	JPC 0 V
STOP	OPR 0 0
RET	OPR 0 1
NEG	OPR 0 2
ADD	OPR 0 3
SUB	OPR 0 4
MUL	OPR 0 5
DIV	OPR 0 6
ISODD	OPR 0 7
EQ	OPR 0 8
NEQ	OPR 0 9
LT	OPR 0 10
LE	OPR 0 11
GT	OPR 0 12
GE	OPR 0 13

Your program should accept programs with

1. Only naïve assembly codes
2. Only macro assembly codes
3. Mixed naïve and macro assembly codes

4.3 Phase II: Build a macro disassembler

Enhance the disassembler built in the previous lab to produce the macro assembly codes instead of the naïve codes.