**STIK 1014 Computer System Organization**

Semester A172

**Extension Information for Project Description**

The project will be carried on in four stages:

1. Lexical Analysis
2. Syntax Analysis
3. Pass one
4. Pass two

**Lexical Analysis**

Lexical analysis divide the program which is separated by white spaces (tab, coma, space, etc) into different strings named as tokens. In short, lexical analysis will:

i. Create tokens

ii. Classify all the tokens into its appropriate types. For this project you will use the following types:

a. OPCODE

b. VARIABLE

c. LABEL

d. REGISTER

iii. For each line, count the lines of code - I byte for one address. I.e. instructions which use 2 bytes will spans two addresses and one byte - one address.

**Syntax Analysis**

This stage will check for the syntax (according to the language rules or format - I.e. instructions can have label (optional) followed by opcode and operand ( can either be registers or variable). if the language rules is violated than program will generate error message.

For this project, the syntax analysis part will be a bonus - optional part. So do it last if there is time.

**Pass One**

read pass the .code tokens. .code and .data is known as the assembler directives. They are not part of the instruction set or the assembly language. They are part of the directives used by the assembler program only. The .code signify the program part while .data signify the data part of the program. The line where assembler directives appear will not be counted as part of the addresses for the instruction of the program.

Firstly you will read the program part (started with .code). In this stage you will :

1. Read the token list from beginning, when ever you see a variable or a label, list them in the symbol table.
2. Calculate the line number for each instructions. Assigns address value for each variable and label seen and keep the information in the symbol table. During the pass through the .code part, you will only know the address for labels used.

Read through until you will see the data part (signify with .data).

1. Whenever you see a variable name defined, you will assign the address for the variable which you have listed in the symbol table based on the line number where the variable appeared in the data part.

**Pass two**

You will create the program translation file based on the instruction translation table specified in the first project assignment.

1. Read pass the .code part of the program again. Translate all the assembly code to machine code based on the translation table.
2. Translate all variables with the associated address values from the symbol table..
3. Translate all labels with the address value for the labels obtained from the symbol table listing.
4. Generate the object file containing the machine code translation of the program.
5. Generate the cross listing file containing the assembly code with the its translation.

**Important Due Dates for project:**

1. Lexical Analysis

Due date : 30th April 2018

Result expected:

List of tokens

Submit: program code and output

1. Pass one

Due date : 10th May 2018

Result expected:

Symbol Table with address of variables and labels

Submit : Program code and output

1. Pass two

Due date : 20th May 2018

Result expected:

Cross Listing file

Object module (machine language)

Submit : Program code and output

1. Report Due and Presentation

Due date : 25th May 2018

Submit: Report and present to lecturer in group.