C Refresher Module Assignment-0.1

Name - Rahul Khatoliya

Roll No. – 2019265

Compilation Process step by step (Command Line Options) :

1. **Preprocessing :**

   Command line argument : **$ gcc -E main.c -o main.i**

   Understanding the Meaning :

   Here **gcc** helps to invoke the GNU C compiler , **-E** enables us to stop and save the output file at preprocessed step (with an **.i extension** , meaning it's a preprocessed file) , Moreover **main.c** is our C program to be preprocessed and **-o** helps to name the resultant output file .

   Description of the Output File :

   1. It gets rid of all the comments in the source file of our program .
   2. it includes the code of the *header file(s)*, which is a file with extension .h which contains C function declarations and macro definitions .
   3. Removes all #define (which ever not used in the program) .

4. it replaces all of the *macros* (fragments of code which have been given a name) by their values .

2. **Compiling :**

   Command line argument **: $ gcc -S main.i -o main.s**

   Understanding the Meaning :

   Here **gcc** helps to invoke the GNU C compiler , **-S** is used to stop and save the process at intermediate compilation step (with a **.s extension** , meaning it's an Assembly level instructions) , Moreover **main.c** is our C program to be and **-o** helps to name the resultant output file .

   Description of the Output File :

   1. File is purely based on Assembly language and contains Assembly commands .
   2. It will produce different Language designs according to different system architecture . for eg:- (x86/64) etc .
   3. Output file is the Resultant (Intermediate Representation) generated by compiler using the earlier preprocessed file .

3. **Assembly :**

   Command line argument : **$ gcc -c main.s -o main.o**

   Understanding the Meaning :

   Here **gcc** helps to invoke the GNU C compiler, **-c** is used to generate object file from the earlier formed assembly .s file and itself is a machine level instructions (with a **.o extension**) , Moreover **main.c** is our C program and **-o** helps to name the resultant output file . This is the next step of compiling, in this step the **main.s** (assembly level instructions) is used as the input file and converted to the **main.o** using the Assembler which is an object file (machine level instructions).

   Description of the Output File :

   1. The assembler takes the IR code and transforms it into object code, that is code in machine language (i.e. binary).
   2. Generated file is not human readable .
   3. It has .o extension .
   4. It is further linked , in the linking process .