**Compilation process for c program**

Generally, the C program building process involves four stages and utilizes different tools such as preprocessor, compiler, assembler & linker.

Consider the C program for Bubble sort. Below are the stages that happen in order regardless of the operating system & architecture.

# Preprocessor

In this stage, lines starting with a # character are interpreted by the *preprocessor* as *preprocessor commands*. These commands form a simple macro language with its own syntax and semantics. This language is used to reduce repetition in source code by providing functionality to inline files, define macros and to conditionally omit code.

Before interpreting commands, the preprocessor does some initial processing. This includes joining continued lines (lines ending with a \) and stripping comments.

Capturing the preprocesses output in bubble\_sort.i

$gcc -E bubble\_sort.c >> bubble\_sort.i

Observations:

* The code from the header files gets included.
* The inline functions are provided with functionality.
* The comments are removed.

# Compilation

In this stage, the preprocessed code is translated to *assembly instructions* specific to the target processor architecture. These form an intermediate human readable language.

It takes the output of the preprocessor, and the

source code, and generates assembler source code.

On compilation, the bubble\_sort.s file gets created.

$gcc -S bubble\_sort.c

Observations:

* This will create a file named bubble\_sort.s, containing the generated assembly instructions.
* This is intermediate human readable form.

# Assembly

During the assembly stage, an assembler is used to translate the assembly instructions to machine code, or *object code*. The output consists of actual instructions to be run by the target processor.

On assembly, the bubble\_sort.o file gets created.

$gcc -c bubble\_sort.c

Observations:

* Running the above command will create a file named bubble\_sort.o, containing the object code of the program.
* The contents of this file are in a binary format.

# Linking

The object code generated in the assembly stage is composed of machine instructions that the processor understands but some pieces of the program are out of order or missing. To produce an executable program, the existing pieces have to be rearranged and the missing ones filled in. This process is called linking.

Command to generate the file:

$gcc bubble\_sort.c

Observations:

* The result of this stage is the final executable program.
* When run without options, cc will name this file a.out.
* Linker will also add pieces containing the instructions for library functions used by the program.

When we run the a.out file, the C program gets executed.

Command: $./a.out