Introduction

C is a popular programming language mostly used for writing system softwares. C is a compiled language. The C compiler compiles C code to produce machine instructions which can directly be run by CPU.

A typical way of compiling a C program is to run the compiler on C code. The compiler will generate assembly language code and run assembler to get CPU instruction code. The translation of C code into assembly language is dependent on the compiler. Different compilers may use different strategies. But the differences are not much and there is a common pattern that every compiler follow.

This article describes how a C code is translated into assembly language using **gcc compiler on i386 platform**. It describes the translation of almost all language constructs of C.

Pre req: C and a little bit of assembly language.

Assembly language syntax

Please note that the assembly language syntax used in this article is GNU assembler(GAS) syntax. The syntax is different from the MASM or NASM syntax.

Here is the major difference:

NASM or MASM:
instruction dest, src
GAS:
instruction src, dest

Readers should get familiar with the GNU assembler syntax before reading this article.

Who should read this

- One who is curious to know how C code is translated into assembly language.
- System programmer who writes code in C and assembly.
- One who wants to do reverse engineering of binary which is generated from some C code.

up (/cin/cin.html)
Program Structure › (/cin/programstructure.html)