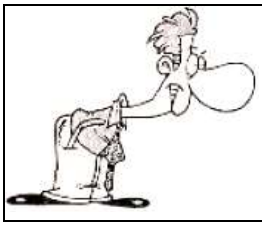
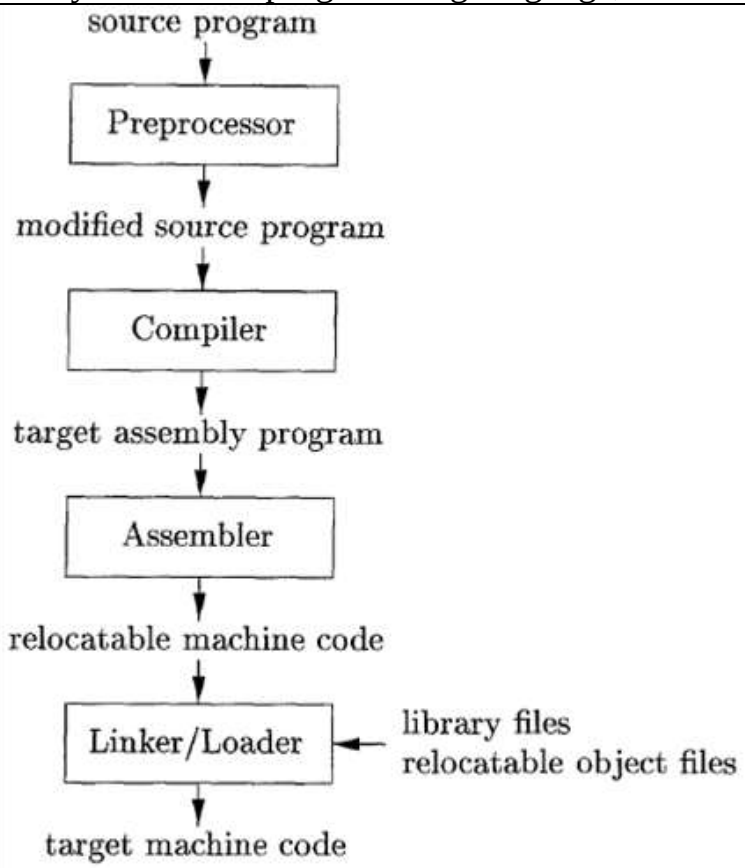
3.1 programming Language Processing



source language **:**
Likes traditional programming languages such as C# and Java.

target language **:**
May be another programming language, or machine language.



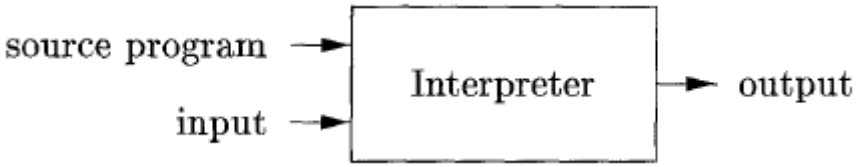
تعريف الـ
Preprocessor
و الـ
Assembler
و الـ
Linker/loader

A source program may be divided into **modules** stored in separate files. The task of **collecting** the source program is sometimes entrusted to a separate program, called a *preprocessor*.
Preprocessor may expand shorthands, called **macros**, into source language statements. Modified source program is fed to compiler.
The compiler may produce an assembly-language program as its output, because **assembly language is easier to produce** as output and is **easier to debug**. The assembly language is then processed by a program called an *assembler* that produces **relocatable** machine code as its output.
Large programs are often **compiled in pieces**, so the relocatable machine code may have to be linked together with other relocatable object files and **library files** into the code that actually runs on the machine. The *linker* resolves external memory addresses, where the code in one file may refer to a location in another file. The *loader* then puts together all of the executable object files **into memory** for execution.

تعريف الـ
interpreter

An **interpreter** is common kind of language processor.

Instead of producing a target program as a translation, an interpreter appears to directly **execute** the operations specified in the **source program** on **inputs** supplied by the user.
.. بينفذ على طووووول



صحصح للكلام المفيد

ميزة الكوموبايلر:
The machine-language **target program** produced by a compiler is usually much **faster** than an interpreter at mapping inputs to outputs .

ميزة الانتربرطر:
An interpreter, can give **better error diagnostics** than a compiler, because it executes the source program statement by statement.

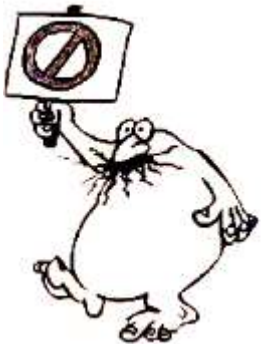
What is the difference between a **compiler** and an **interpreter**?

What are the advantages of
(a) a **compiler over an interpreter**
(b) an **interpreter over a compiler**?

What advantages are there to a language-processing system in which the **compiler produces assembly** language rather than machine language?

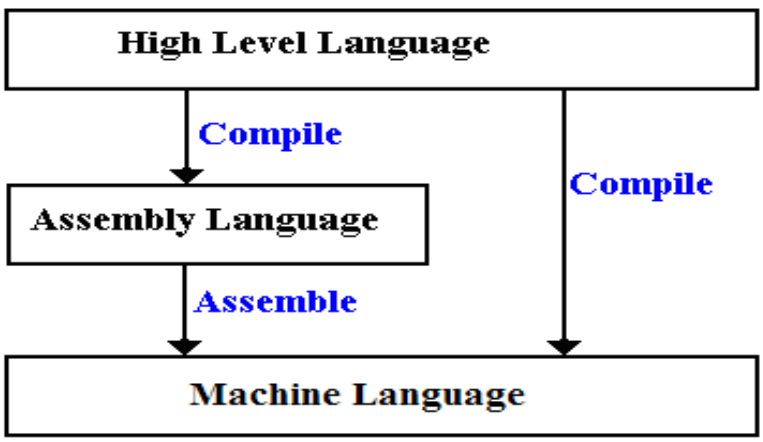
A compiler that translates a high-level language into another high-level language is called a *source-to-source* translator. What advantages are there to **using C as a target language** ?

Describe some of the **tasks** that an **assembler** needs to perform.

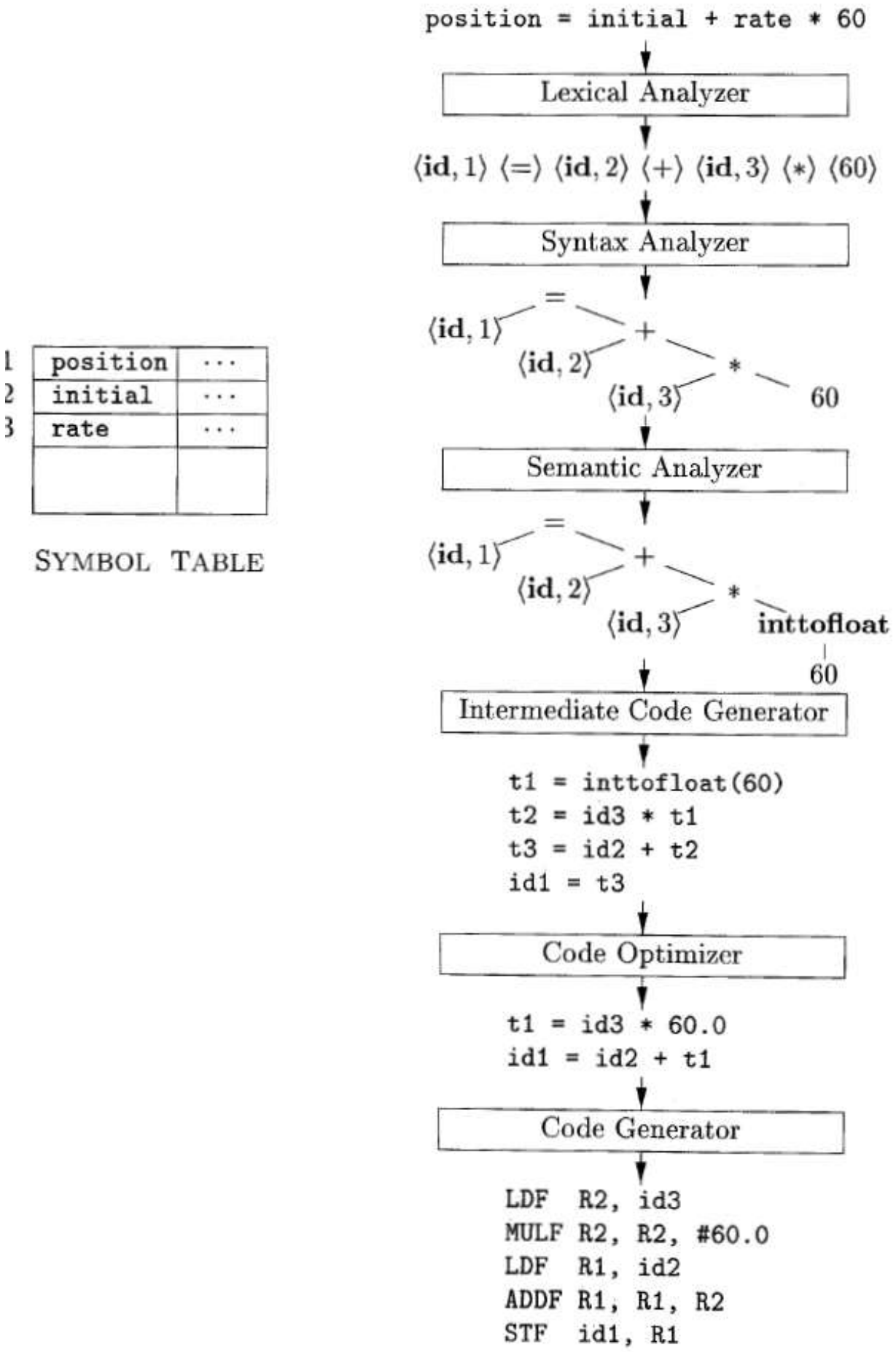


مع المفتش كورومبو
أسئلة على اللي فات

To summarize



Phases of Compiler



References

Alfred V. Aho, Monica S. Lam, Ravi Sethi, and Jeffrey D. Ullman, Compilers: Principles, Techniques, and Tools