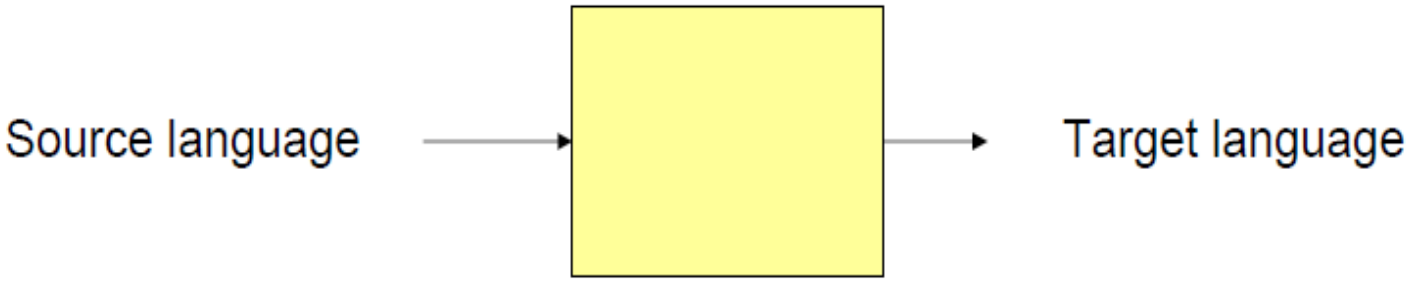
COMPILER BASICS

LECTURE 01

What is a Compiler?

A program that reads a program written in one language and translates it into another language.



Traditionally, compilers go from high-level languages to low-level languages.

What is a Compiler?

- Compilers are the bridges:
  - Tools to translate programs written in high-level languages to efficient executable code.

- Another important role of the compiler is to report any errors in the source program that it detects during the translation process.

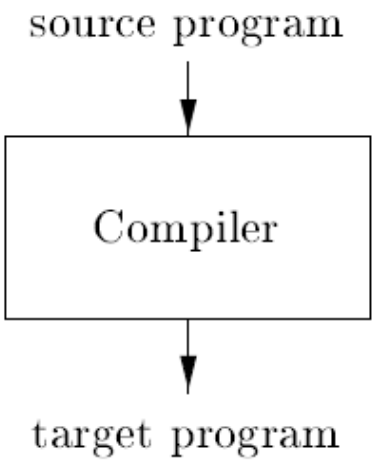


Figure 1.1: A compiler

Figure 1.2: Running the target program

What is an Interpreter?

- An interpreter requires no pre-computation it directly takes the source program and runs the input on it producing the desired output.

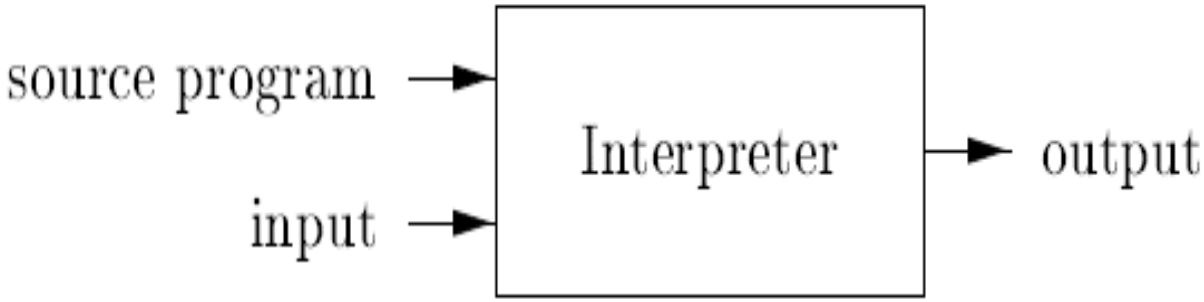
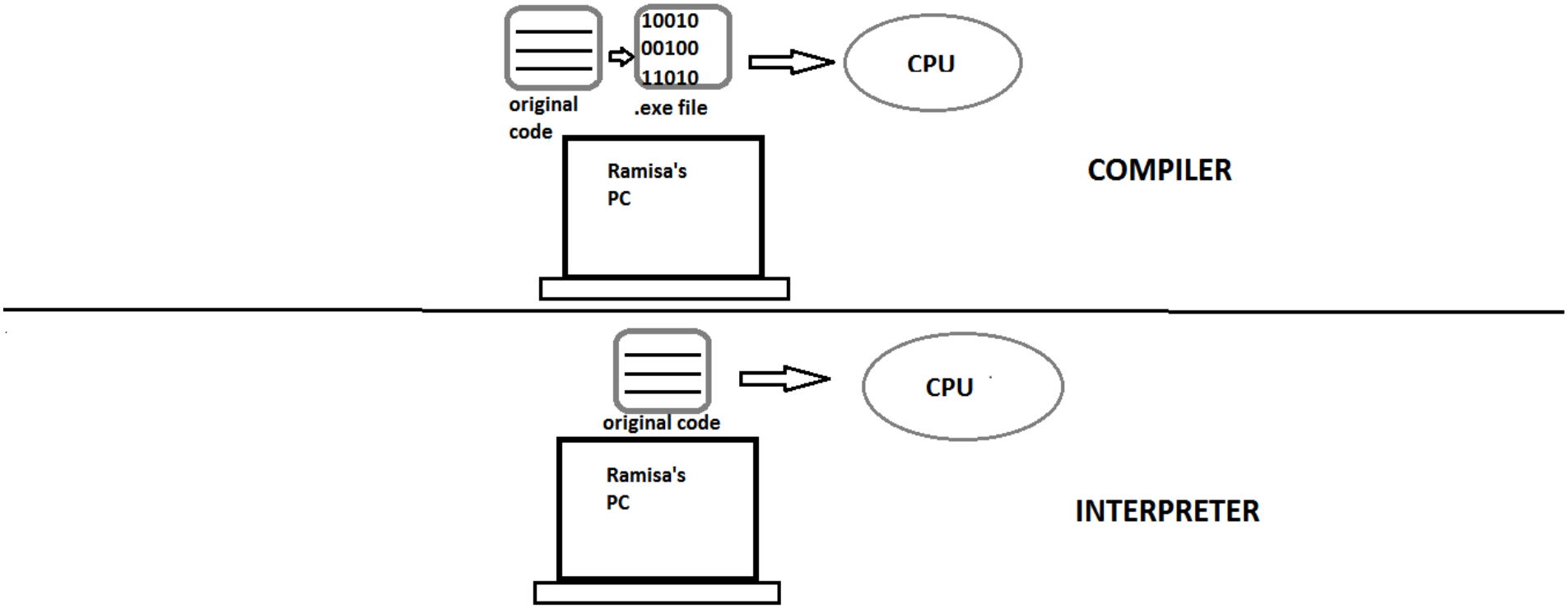


Figure 1.3: An interpreter

Compiler VS Interpreter



Compiler vs Interpreter

- Compilers and interpreters do similar jobs, but there are differences:

  To run a program you've written, it must first be translated into machine code so the computer can read it. This is what compilers and interpreters do.

  However, compilers convert the code all at once, save it, then run it; whereas interpreters translate the code one line at a time, as it is run.

- Programming language like Python, Ruby use interpreters.
- Programming language like C, C++ use compilers.
- Java?
- Error display?

Language Processing System

- Interpreters: discussed before
- Preprocessors: 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. The preprocessor may also expand macros into source language statements
- Assemblers: The assembly language is processed by a program called an assembler that produces relocatable machine code as its output.
- Linkers: 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
- Loaders: puts together all of the executable object files into memory for execution.

Steps for Language Processing System

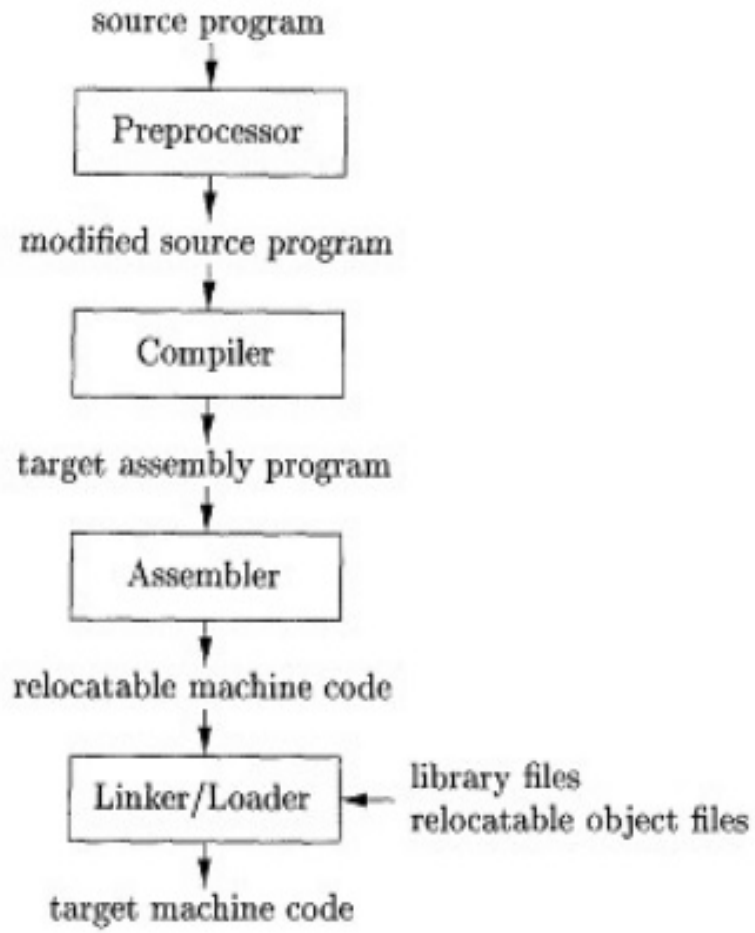


Figure 1.5: A language-processing system

Preprocessor

- **File Inclusion**

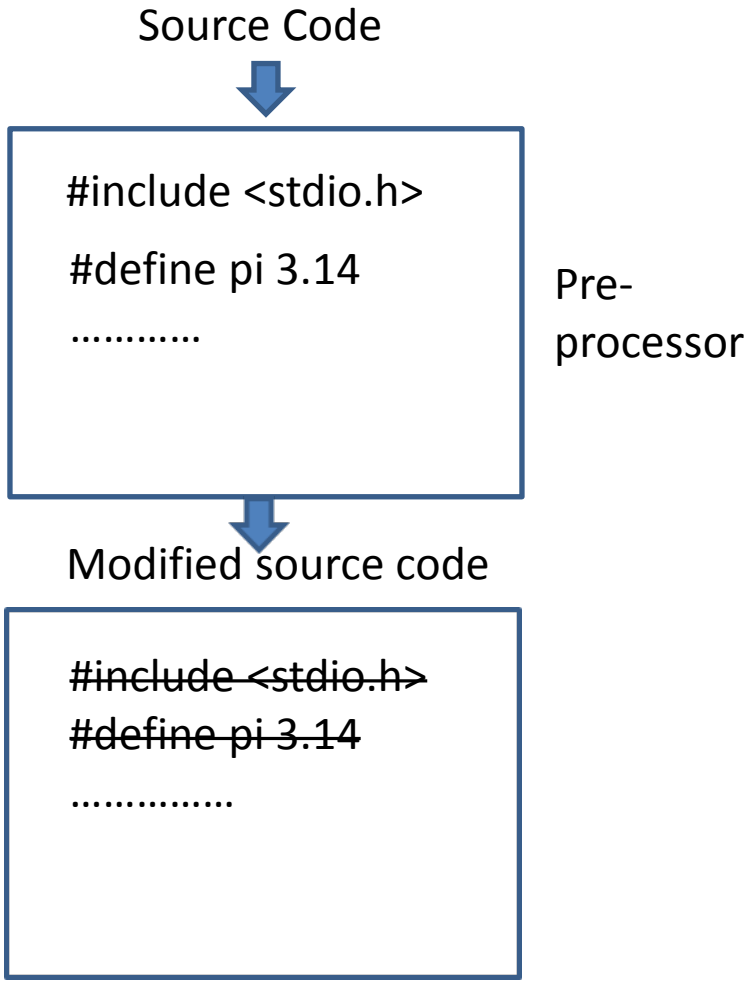
#include <stdio.h>

#include "myfile.h"

- **Defining Macros**

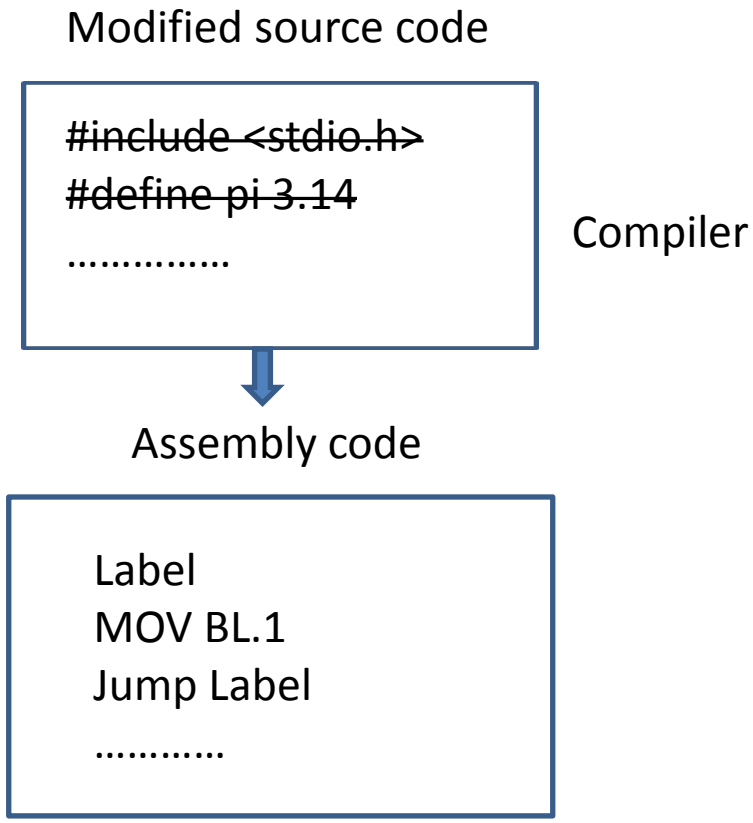
#define pi 3.14

#define square (x) ((x)(x))*



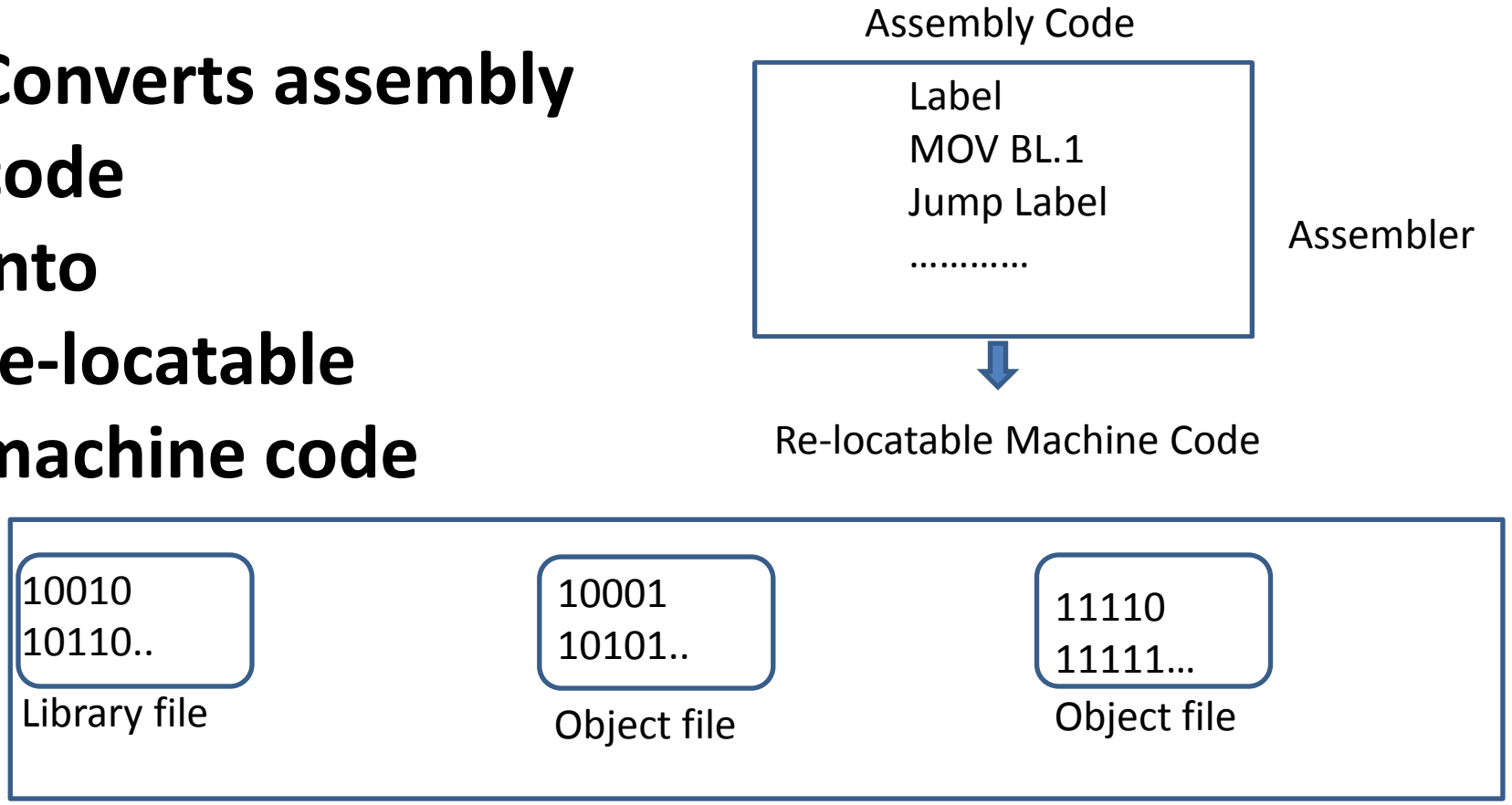
Compiler

- **Converts modified source code into assembly language**



Assembler

- **Converts assembly code into re-locatable machine code**

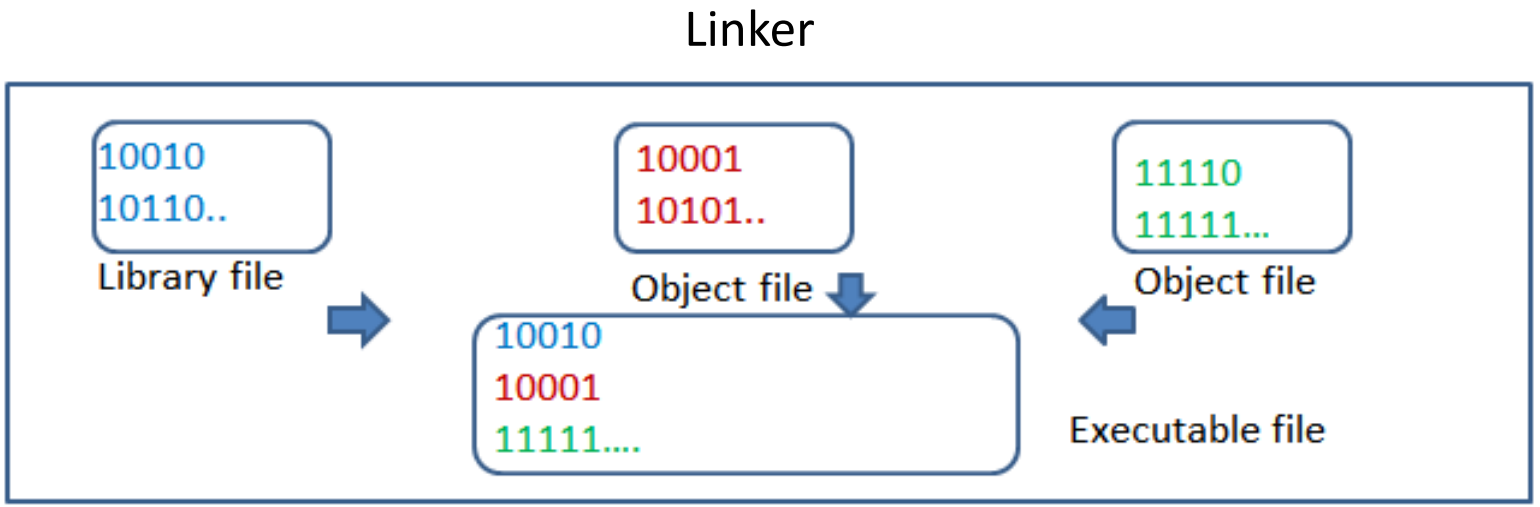


Linker

- **Linker:**

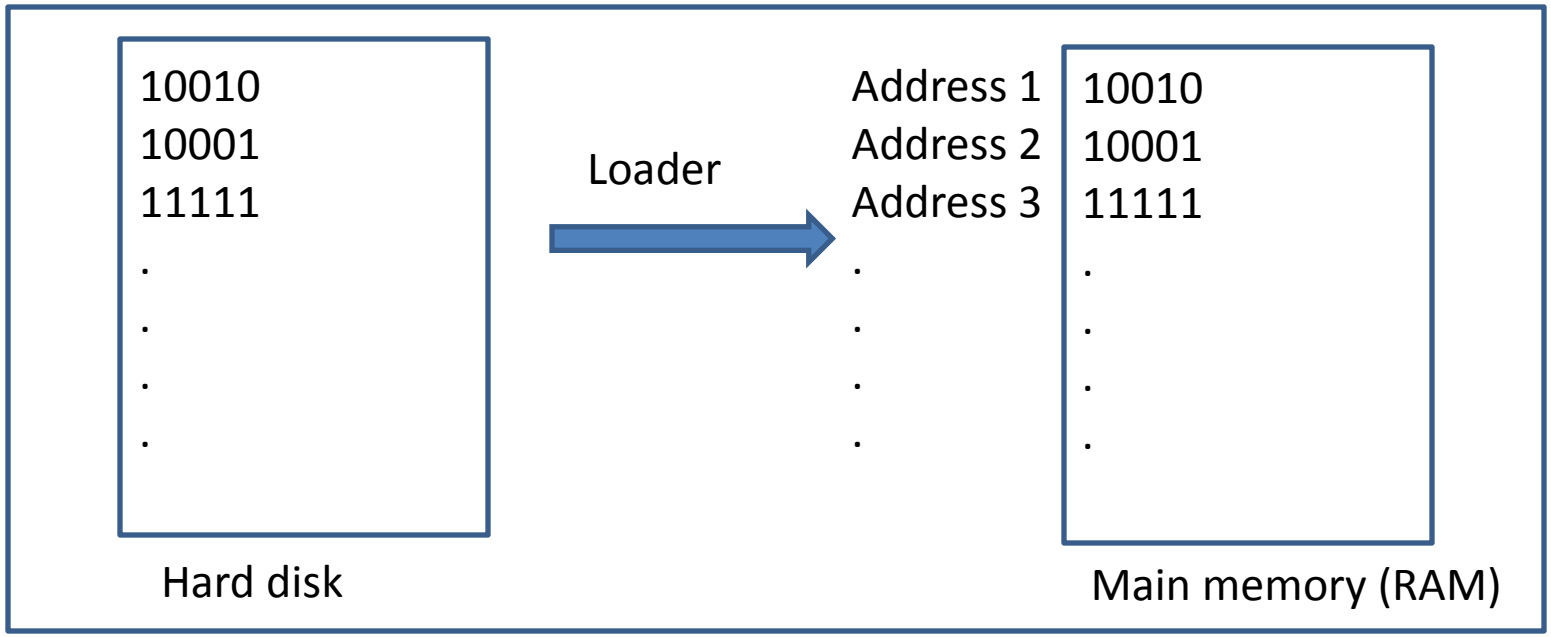
Links relocatable object files

Into executable file



Loader

- Loads executable machine code into main memory for execution



Summary

- Compiler
- Compiler vs interpreter
- Steps of a language processing system

Thank You

Questions?