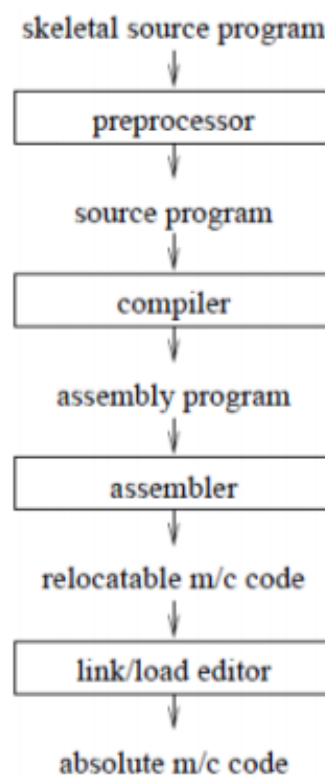


Lecturer (2)

1.2 The Context of a Compiler

The complete process of compilation is illustrated as:



1.2.1 Preprocessors

Preprocessing performs (usually simple) operations on the source file(s) prior to compilation. Typical preprocessing operations include:

- (a) Expanding macros (shorthand notations for longer constructs). For example, in C,

```
#define foo(x,y) (3*x+y*(2+x))
```

defines a macro foo, that when used in later in the program, is expanded by the preprocessor. For example, `a = foo(a,b)` becomes

```
a = (3*a+b*(2+a))
```

- (b) Inserting named files. For example, in C,

```
#include "header.h"
```

is replaced by the contents of the file header.h

1.2.2 Linkers

A linker combines object code (machine code that has not yet been linked) produced from compiling and assembling many source programs, as well as standard library functions and resources supplied by the operating system. This involves resolving references in each object file to external variables and procedures declared in other files.

1.2.3 Loaders

Compilers, assemblers and linkers usually produce code whose memory references are made relative to an undetermined starting location that can be anywhere in memory (relocatable machine code). A loader calculates appropriate absolute addresses for these memory locations and amends the code to use these addresses.