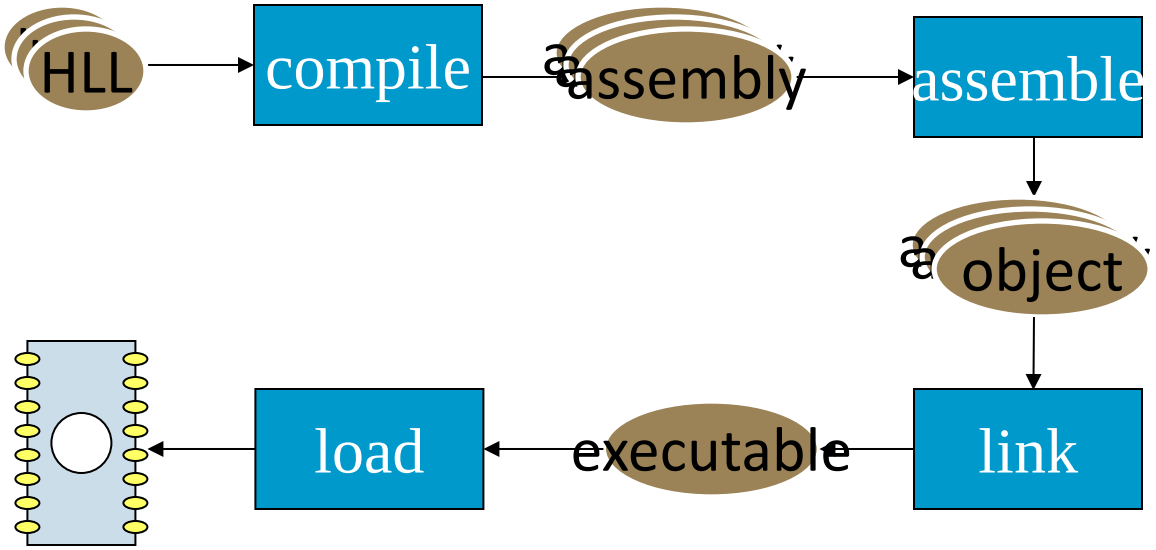
Assembly and linking

Last steps in compilation:



The embedded software is typically written in HLL and need to be executed on the embedded hardware platform. To accomplish this, the following intermediate steps are performed. Details of these are explained in later slides.

Multiple-module programs

- Programs may be composed from several files.

- Addresses become more specific during processing:

  - relative addresses are measured relative to the start of a module;
  - absolute addresses are measured relative to the start of the CPU address space.

To make it easy the programs are developed in multiple files. Some of these files may be executable code purchased as libraries.

Assemblers

Major tasks:

- generate binary for symbolic instructions;
- translate labels into addresses;
- handle pseudo-ops (data, etc.).

Generally one-to-one translation.

Assemblers are a refined version of machine language (binary executable) code.
Labels and pseudo operations are used to make the assembly code easier to work with.

Two-pass assembly

Pass 1:

- generate symbol table

Pass 2:

- generate binary instructions

Symbol table

Symbol table contains the addresses of the symbols relative to the start of the object module.

MIPS and ARM instructions take 4 bytes of storage for each instruction. Intel instructions are of variable length can take anywhere from 1 to more than 20 bytes land of memory storage.

Symbol table generation

- Use program location counter (PLC) to determine address of each location.
- Scan program, keeping count of PLC.
- Addresses are generated at assembly time, not execution time.

Relative address generation

- Some label values may not be known at assembly time.
- Labels within the module may be kept in relative form.
- Must keep track of external labels---can’t generate full binary for instructions that use external labels.

Pseudo-operations

Pseudo-ops do not generate instructions:

- ORG sets program location.
- EQU generates symbol table entry without advancing PLC.
- Data statements define data blocks.

Example

Code

label1	100
label2	108
label3	116

Symbol table