

**Lesson 4: Software for  
embedding in a System- Part 1  
ROM image, Programming Languages  
and Program models**

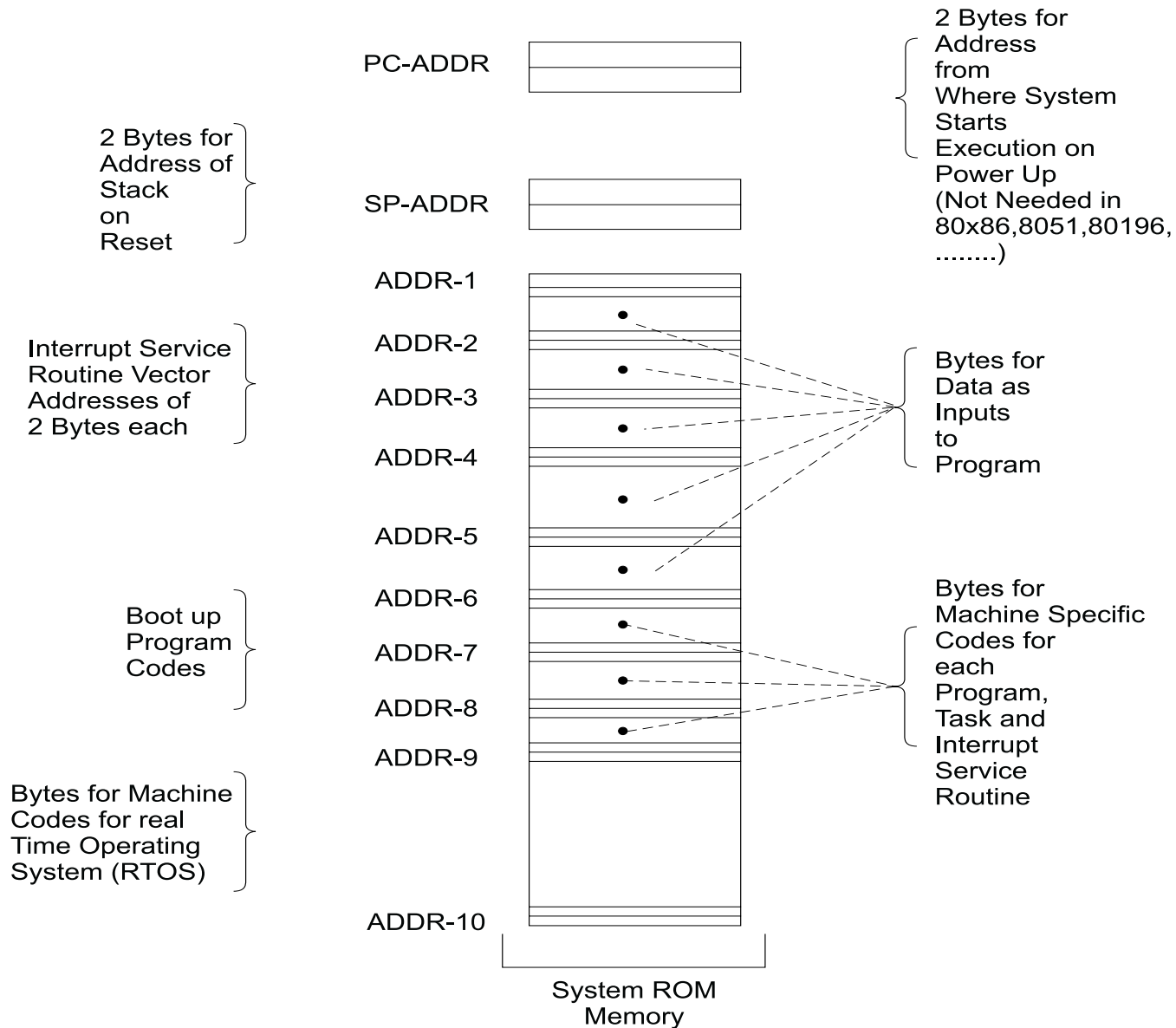
# Outline

- ROM image
- Programming Languages
- Program models

# 1. ROM Image

- Final stage software also called ROM image\*

\* (Just as an image is a unique sequence and arrangement of pixels, embedded software is also a unique placement and arrangement at each ROM address of bytes for instructions and data. )



# System ROM memory embedding the software, RTOS, data, and vector addresses

# Final machine software

- Bytes at each address defined for creating the ROM image.
- By changing this image, the same hardware platform work differently and can be used for entirely different applications or for new upgrades of the same system.

# Distinct ROM image in a distinct Embedded System

- Hardware elements between the distinct systems can be identical but it is the software that makes a system unique and distinct from the other.

# Compressed Codes and Data

- ROM image may alternatively be compressed software (for example, the zip format) and data (for example, the pictures in jpg or gif format) along with the software required for decompression algorithm

# Outline

- ROM image
- Programming Languages
- Program models



# 1. Machine Language Coding

- Programmer defines the addresses and the corresponding bytes or bits at each address.
- Used in configuring some specific physical device or subsystem like transceiver, the machine code- based coding is used

## 2. Assembly Language Coding

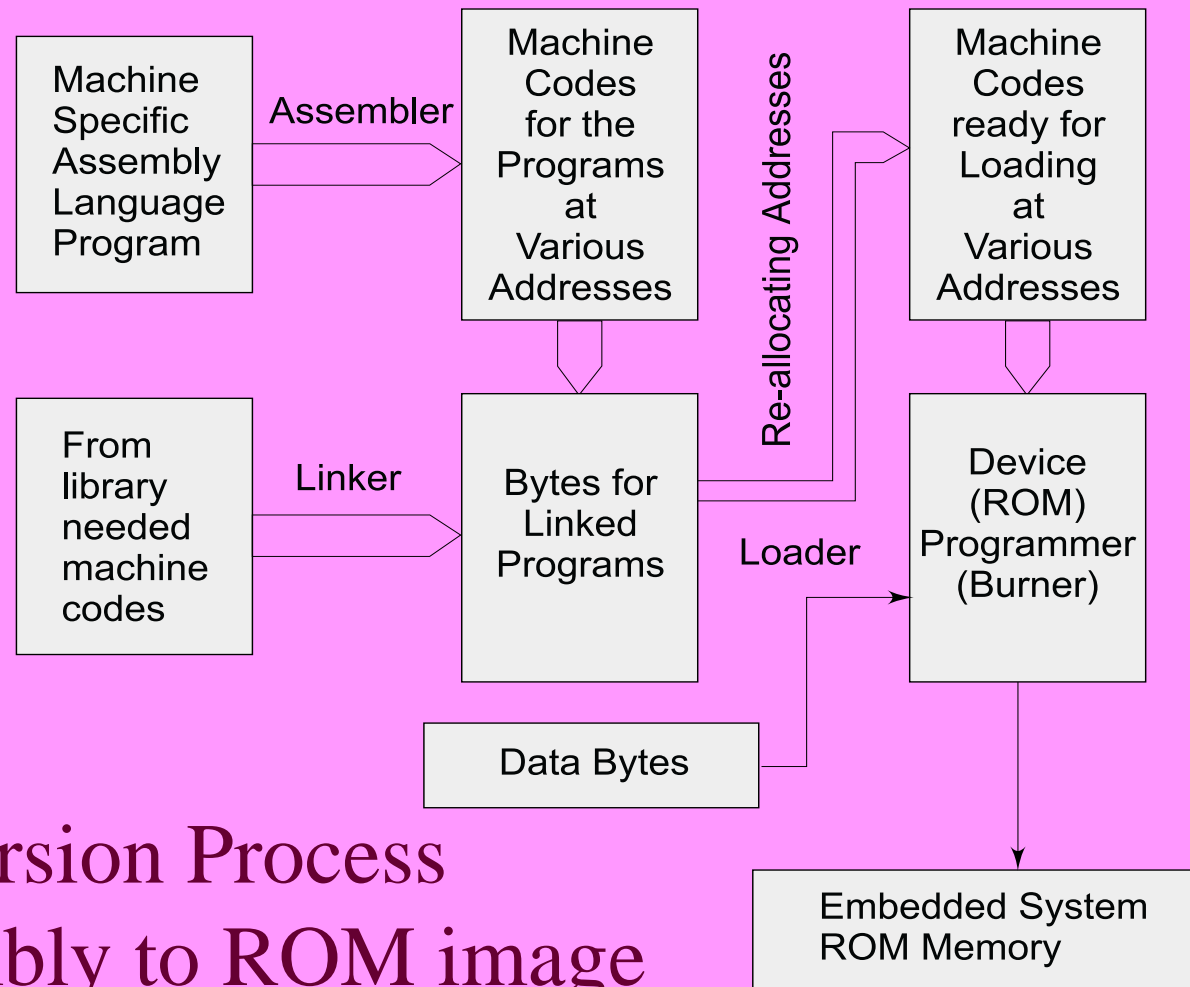
- Needed for Invoking Processor Specific Instructions
- Requires understanding of the processor and instruction set.
- A program or a small specific part coded in the assembly language using an Assembler (software used for developing codes in assembly).

## Three steps when using assembly language

- 'Assembler',
- 'Linker' and
- 'Locator'

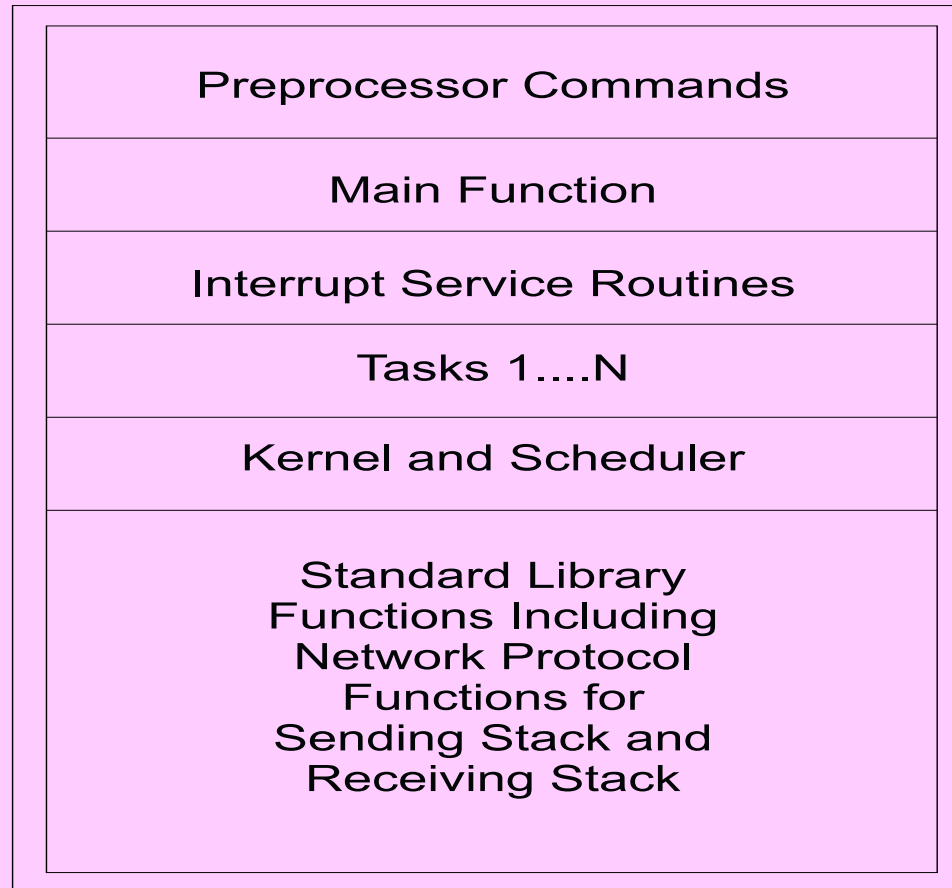
before finally burned at the ROM

# Assembly



## Conversion Process Assembly to ROM image

### 3. Programming language C or C++ or Visual C++ or Java .



- Application Software - Different Program Layers

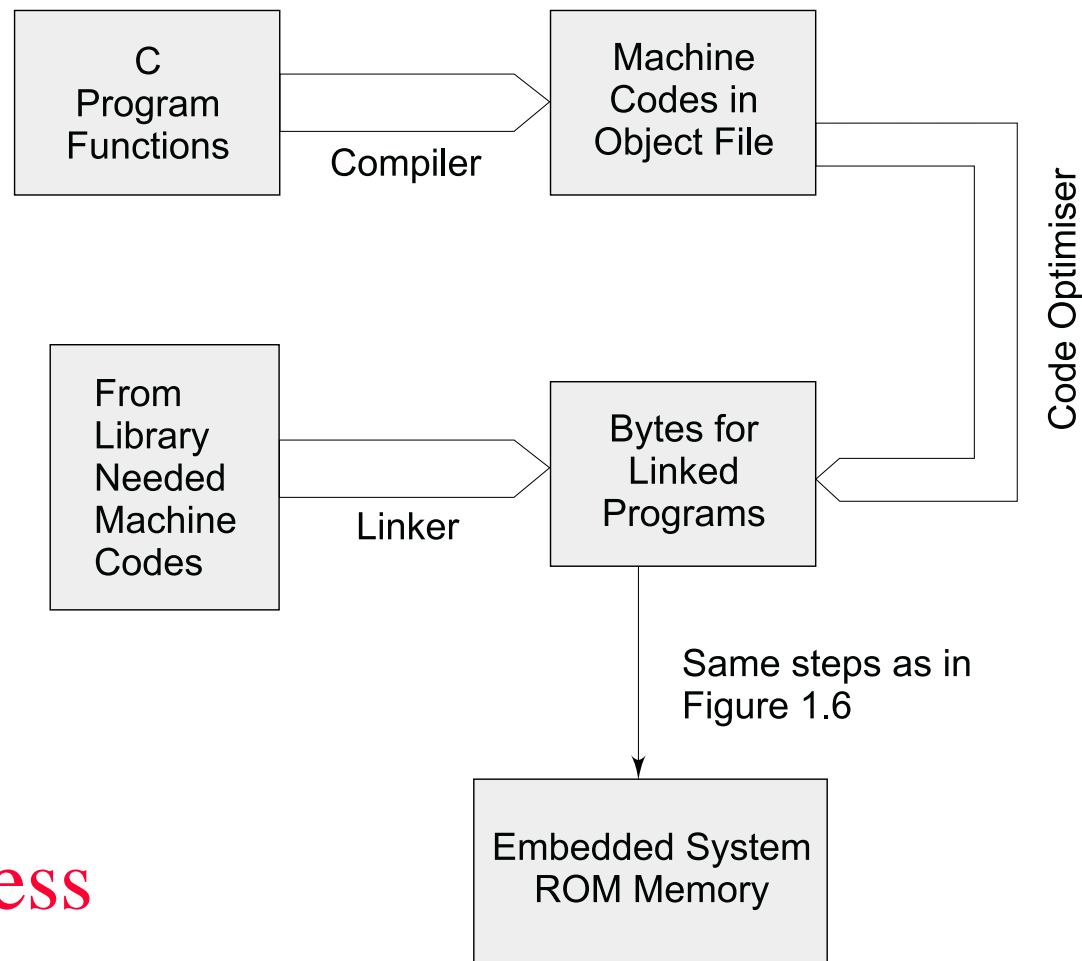
## **Program various layers–**

- **processor commands,**
- **main function,**
- **task functions and**
- **library functions,**
- **interrupt service routines**
- **and kernel (scheduler).**

# Compiler

- **Generates an object file. Using linker and locator, the file for ROM image is created for the targeted hardware. C++ and Java are other languages used for software coding.**

# Converting a C program into ROM image



Process



# Outline

- ROM image
- Programming Languages
- Program models

# Program Models

- Sequential Programming Model
- Object Oriented Programming Model
- Control and Data flow graphs or Synchronous Data Flow (SDF) Graph or Multi Thread Graph (MTG) Model
- Finite State Machine for data path
- Multithreaded Model
- Concurrent Processing of processes or thread or tasks

# Summary

We learnt:

- (i) embedded software saves as ROM image
- (ii) conversion steps from high level language codes to assembly and then to the ROM image
- (iii) program layers in the embedded software

## We learnt:

- (iv) high level language used for software development is C, C++, visual C++ or Java
- (v) Models, which are used for a embedded-software development process

# End of the Lesson –4