Introduction to build process

By: Yehia M. Abu Eita

Outlines

- Introduction
- Why learning build process?
- Build process stages

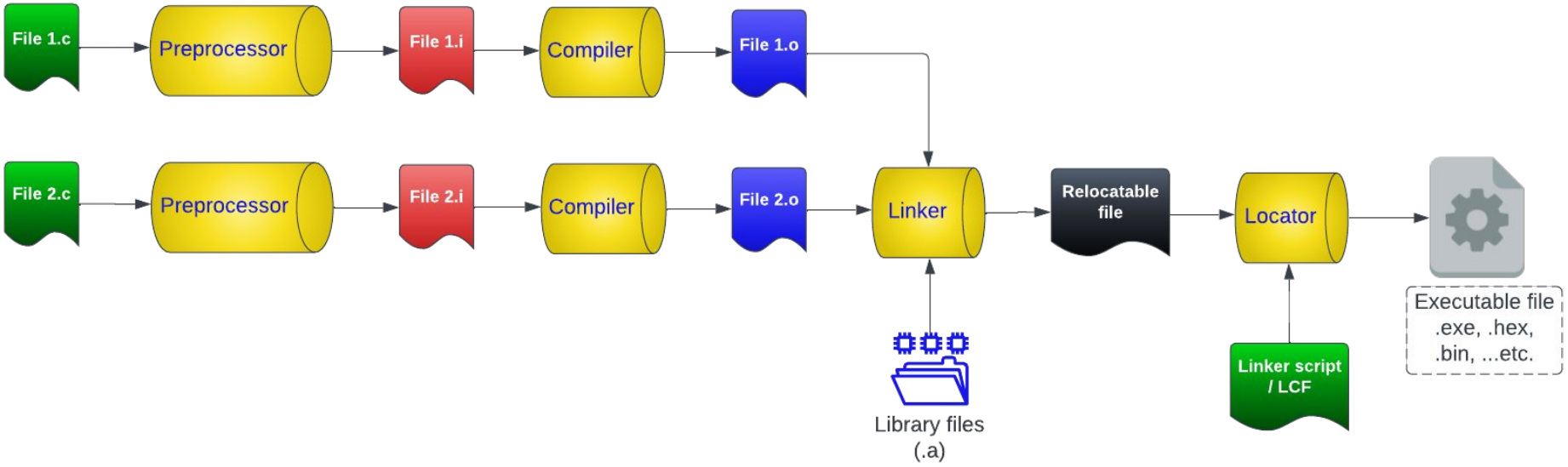
Introduction

- In order to make your **code executable**, you must provide your code to a **language** that is **suitable** to the **microprocessor**.
- Writing a **machine language** directly is **very hard**, so we use C language because it’s easier for us to understand.
- **Each processor** has its **own machine language**.
- The **C build process** can be defined as the **processes** done to **convert** your **code** to a **machine language** that is suitable for the selected targets.
- Build process make a level of **microprocessor abstraction**.

Why learning build process?

- **Learning build process will:**
  - Gives you better understanding how your code is converted to machine language.
  - Gives you better understanding about syntax errors that you are facing and how to fix them quickly.
  - Gives you better understanding about avoiding warnings that you are facing and how to eliminate them quickly.
  - Gives you better understanding about how to easily handle multi-file projects.
  - Gives you better understanding about how variables and functions are shared between files or are private in its files.
  - Gives you better understanding about how variables and functions are stored in the memory.

Build process stages



Summary

- Now you know what is build process
- Learning build process will make it easier for you to implement and debug your applications
- The build process has 4 stages, preprocessor, compiler, linker, and locator.