Overview of C/C++ Compilers

Deva Nayak

January 26, 2025

1 Introduction to C/C++ Compilers

A compiler translates human-readable C/C++ source code into machine-readable executable code. Compilers are essential for creating applications and are specific to platforms, operating systems, and sometimes hardware.

1.1 Key Features of a Compiler

- Converts source code into machine code.
- Ensures platform-specific compatibility.
- Provides debugging, optimization, and error diagnostics.

—

2 Compilers by Operating Systems

2.1 Linux

- **GCC (GNU Compiler Collection):**
  - Open-source and widely used for C and C++.
  - Rich debugging and optimization options.

  Example Command:

```
gcc -o program program.c
g++ -o program program.cpp
```

- **Clang (LLVM):**
  - Modern, fast, and modular compiler.
  - Compatible with GCC commands.

  Example Command:

```
clang -o program program.c
clang++ -o program program.cpp
```

2.2 Windows

- **MinGW (Minimalist GNU for Windows):**
  - GCC-based compiler for Windows.
  - Lightweight and straightforward.

  Example Command:

```
gcc -o program.exe program.c
```

- **Microsoft Visual C++ (MSVC):**
  - Part of Visual Studio.
  - Provides advanced debugging tools and Windows optimization.

  Example Command:

```
cl /EHsc program.cpp
```

- **Cygwin:**
  - Unix-like environment for Windows with GCC support.

2.3 macOS

- **Clang (Default):**
  - Comes with Xcode tools.
  - Optimized for macOS.

  Example Command:

```
clang -o program program.c
clang++ -o program program.cpp
```

- **GCC:**
  - Available via Homebrew.

  Example Command:

```
brew install gcc
gcc-<version> -o program program.c
```

—

3 Compilers for Embedded Targets

3.1 ARM (Embedded Systems)

- **ARM GCC (GNU Arm Embedded Toolchain):**
  - Free, open-source compiler for ARM Cortex-M boards.

  Example Command:

```
arm-none-eabi-gcc -o program.elf program.c
```

- **Keil MDK:**
  - Proprietary IDE and compiler for ARM devices.
- **IAR Embedded Workbench:**
  - Commercial compiler for embedded systems with advanced optimization.

3.2 AVR Microcontrollers

- **AVR-GCC:**
  - GCC-based compiler for AVR chips (e.g., Arduino).

  Example Command:

```
avr-gcc -o program program.c
```

3.3 ESP32/ESP8266

- **Xtensa GCC Toolchain:**
  - GCC-based toolchain for ESP32/ESP8266.
  - Works with ESP-IDF (Espressif IoT Development Framework).

3.4 RISC-V

- **RISC-V GCC Toolchain:**
  - GCC-based toolchain for RISC-V processors.

—

4 Summary of Popular Compilers

Compiler	**OS**	**Features**	**Usage**
GCC	Linux, macOS, MinGW	Open-source, cross-platform	General-purpose
Clang	macOS, Linux, Windows	Modern, modular, great diagnostics	Performance-critical projects
MSVC	Windows	Microsoft-supported, Windows optimization	Windows applications
ARM GCC	Cross-platform	Free toolchain for ARM microcontrollers	Embedded systems
Keil MDK	Windows	Proprietary, ARM Cortex-M optimization	Professional embedded projects
IAR Embedded Workbench	Cross-platform	Highly optimized	Embedded development
AVR-GCC	Cross-platform	Open-source for AVR microcontrollers	Arduino projects
Xtensa GCC	Cross-platform	Toolchain for ESP32/ESP8266	IoT development