

Development Tools 1: Introduction

1. **Introduction**
2. Text Editors
3. IDE's
4. Compiling
5. Debugging and Tracing
6. Advanced Development

Development Tools for C on Linux

In these slides we will consider the following categories of tools to support the development of C programs on Linux systems:

- text editors
- IDE's
- compilers
- build tools
- debuggers
- execution tracing tools
- memory debugging tools

(We focus on development tools that are useful for this course, merely mentioning more advanced tools.)

Development Tools for C on Linux (contd.)

Text editors:

- C **source files** are often created using a **text editor**.
- Typically we want **syntax aware** editors.
- Text editors can be GUI or command line (or both).

IDE's:

- IDE's are not as popular for C as for C++ or Java.
- This is probably because the C language is much simpler.
- Several free IDE's are available, but students should also get used to working without an IDE!
- The IDE's often use GCC, etc. to do much of the real work.

Development Tools for C on Linux (contd.)

Compilers:

- A compiler turns source code into **object code** so it can be executed.
- The most popular C compiler is **GCC** (The GNU Compiler Collection).
- This is the only C compiler one should use in this class.

Build tools:

- The “**GNU toolchain**” includes a number of tools to assist in building software and in building cross-platform (UNIX) software.
- E.g., **Make**, which uses *dependency specifications* to determine what needs to be done to produce an executable, etc.

Development Tools for C on Linux (contd.)

Debuggers:

- The most popular debugger is **GDB** (The GNU Debugger).
- GDB is command line only, but **DDD** provides a GUI.

Execution tracing tools:

- There are tools besides “debuggers” that can help when debugging programs
- Some tools allow you to *trace library/system calls* being made by a running program.

Development Tools for C on Linux (contd.)

Memory debugging tools:

- Another class of debugging tools detect *memory management* bugs.
- Such tools may monitor memory operations during execution, or they may try to (statically) evaluate code looking for potential problems.