Seminar Abstract

Advanced C Programming

Course Overview:

This course teaches students advanced aspects of Standard C. The time is spent two thirds on lectures and one third on writing and debugging lab problems. To reinforce the theory, lab sessions are run immediately after the lectures to which they apply. The course is not hardware or operating system-specific.

Course Length: 3–5 days (depending on the coverage sought and the experience level of the attendees)

Goals:

Provided students meet the prerequisites, at the end of the course they should have a good understanding of the following:

- Be knowledgeable about important but subtle issues regarding operator precedence and order of evaluation.
- The subtle properties of the built-in types.
- Be comfortable with the type qualifiers: `const`, `volatile`, and `restrict`.
- Know how and when to use type synonyms.
- Be able to read and write arbitrarily complex declarations reliably.
- The more exotic aspects of the the Preprocessor.
- Know all aspects of data pointers.
- Know how and why to use pointers to void.
- Know how and why to use pointers to functions.
- Know how and why to use pointers to arrays.
- Know how to write an interrupt handler and how to register an exit handler.
- Have a basic understanding of C's support for internationalization
- Have a basic understanding of the new support for threading
- Be able to use the more advanced library functions that support things like variable-length argument lists, sorting, and searching.

Who Should Attend:

Programmers and technical managers who need formal training in the more advanced aspects of C. Also, experienced programmers not familiar with additions and changes made when C was standardized.

Advanced C Programming – Seminar Abstract

Prerequisites:

Successful completion of the 5-day "Programming in C" course, or its equivalent. Students are expected to be relatively fluent in the basic syntax and semantics of C, particularly with regard to data pointers, structures, and operators. Students interested in C from a portability perspective should have some knowledge of different hardware platforms and operating environments.

Materials:

Advanced Programming in C – This manuscript was written specifically for teaching. It serves as a useful reference once the course has been completed.

Detailed Topics:

The main topics covered are:

- Managing identifier namespace
- Mixing signed and unsigned types
- The Boolean type
- Operators, precedence, order of evaluation
- Sequence points and lvalues
- Designated initializers
- Compound literals
- Variable-Length Argument Lists
- Inline Functions
- Preprocessor operators and pragmas; a header design strategy
- All-things data pointers
- Pointer to `void`
- Pointer to a function
- Pointer to an array
- Input and Output (optional)
- Structure member alignment
- The comma operator
- Mastering declarations
- Lvalues
- Variable-length arrays
- Internationalization
- Signal handling
- Normal and abnormal program termination
- Non-local jumps
- Sorting and searching
- Introduction to threads