

C LANGUAGE PROGRAM DEBUGGING UNDER UNIX®

General:

This course provides a discussion of a number of C Language program debugging techniques. It also proposes a methodology to prevent program bugs from occurring. The student is prepared to write error free code, as well as to detect and fix program bugs using various debugging tools available under the UNIX Operating System.

Objectives:

Upon successful completion of this course, the student will be able to:

- Identify the major types of program bugs along with their manifestation and chief causes.
- Describe several useful maxims and rules to assist in reducing program errors.
- Describe and use several debugging tools and utilities available under the UNIX Operating System.

Audience:

Experienced C Language Programmers.

Prerequisites:

Introduction to C Language Programming course or equivalent C Language Programming experience.

Duration:

Two (2) days including classroom lecture and lab sessions.

**C LANGUAGE DEBUGGING
UNDER UNIX
COURSE OUTLINE**

I. WHAT IS A BUG

- A. Types of Bugs
 - 1. Hard Bugs
 - 2. Soft Bugs
 - 3. Other Bugs
- B. Why Bugs Exist
- C. Manifestations of Bugs

II. DEFENDING AGAINST BUGS

- A. Golden Rules of Programming
- B. Mastery of Language and Programming
- C. Rules of Thumb

III. UNIX SYSTEM DEBUGGING TOOLS

- A. Debugging Methodology
- B. Using `lint` for Debugging
- C. Using `printf` for Debugging
 - 1. Advantages
 - 2. Disadvantages
 - 3. `DEBUG` Macro
 - 4. Activating Debugging
- D. Using `assert` for Debugging
- E. Using `ctrace` for Debugging
- F. Using `sdb` for Debugging
 - 1. Single Stepping
 - 2. Examining and Modifying Variables
 - 3. Break Pointing
 - 4. Flow Tracing
 - 5. Function Driving
 - 6. Absolute vs Symbolic Debugging
 - 7. Compiling Programs for use with `sdb`
 - 8. Basic `sdb` Concepts
 - 9. `sdb` Commands
- G. Common Sources of Bugs in C Programs