



C Programming in the Guardian Environment

U4143S

Through a series of lectures and labs, you are taught how to use both the C compiler and the native mode C compiler used with HP NonStop™ servers, how to take advantage of various extensions, and how to interface with Guardian system procedures. In learning to use those compilers, you will become familiar with various aspects of the unique C run-time environment such as memory model issues, diagnostics, special debugging issues, and the retrieval of specific run-time and environment information. Prior C knowledge is essential.

C Programming in the Guardian

Price USD \$1,600

Links to local schedules, pricing and [US/Canada](#)
[Mexico/Latin America](#)
[Brazil](#)

HP course # U4143S

Category NonStop

Duration 2 days

Audience

- System analysts
- System programmers
- Programmers
- Programmer analysts

Prerequisites

- Concepts and Facilities for HP NonStop Systems (U4147S)
- ANSI/ISO C Programming experience
- Recommended: Experience with the Inspect or Native Inspect symbolic debugger
- Recommended: TAL programming experience

Course objectives

- Perform C Programming within the Guardian system procedures
- Understand the unique aspects of the C run-time environment such as memory model issues, diagnostics, special debugging issues, and the retrieval of specific run-time and environment information

Benefits to you

- Building TNC C and native C programs
- Native mode considerations
- The C run-time environment
- Compiler operation and diagnostics
- Debugging C programs with Inspect and Native Inspect
- Interfacing to Guardian system procedures
- Mixed language programming

Next steps

- Guardian API Programming U4152S

Course outline

Module 1 - Compiling and Linking TNS C programs

- Basic operation of the TNS C compiler
- Compiler syntax and the analysis of a compiler listing
- Binder commands needed to combine multiple object files to build a single executable program
- Lab Exercise: Compile a complete C program
- Lab Exercise: Separate compilation and binding

Module 2 - Native Mode Considerations

- Native mode
- Benefits and constraints of native mode and the native mode development tools
- Process attributes and organization

Module 3 - Native Mode Compilation

- Various native mode migration issues particular to C programming
- Using the TNS/E native compiler (CCOMP)
- Usage of the eld utility for linking separately compiled modules
- Lab Exercise: Use the CCOMP compiler to build runnable programs from separately compiled units

Module 4 - Run-time Environment

- NonStop server run-time environment for C programs
- Obtaining and processing environment and startup protocol messages
- Essentials of memory models available for NonStop C and show the four basic areas of heap, RTL, global and stack
- 32 bit pointers and access to system procedures that require 16 bits
- Large and wide memory models
- Lab Exercise: Gain hands-on experience in fetching process startup information

Module 5 - Tandem C Compiler Operation and Diagnostics

- Subset of the available compiler pragmas for Tandem C
- Compiler pragmas for the native C compiler
- Compile-time and run-time diagnostic facilities
- Demonstrating, with examples, various aspects of compiler operations

Module 6 - Debugging C Programs

- Basic framework of Inspect and Native Inspect commands which can be used for debugging NonStop server C programs
- Information regarding Inspect and Native Inspect commands specifically helpful for the NonStop server C environment
- Demonstrating a debugging session
- Lab Exercise: Use Inspect and Native Inspect on an C program

Module 7 - Interfacing to Guardian Procedures

- Specifics of interfacing to Guardian procedures from a C program
- Interface declarations for non-C functions
- Usage of the Guardian procedure calls reference manual and special/problem Guardian procedures
- Lab Exercise: Make calls to Guardian procedures needed for various terminal display options

Module 8 - Mixed-Language Programming

- Mixed language programming as it relates to NonStop C
- Interface declarations, common run-time environment (CRE), specific alignment rules, data storage requirements, and global data sharing
- Lab Exercise: Gain hands-on experience with interface between NonStop C and other languages such as TAL

Onsite-delivery equipment requirements

- One NonStop server with NonStop operating system, version H06 or later
- 6530 terminals or equivalent PC/workstation for instructor demonstrations
- 6530 terminal or equivalent PC/workstation per student
- C, NMC, nld, noft, TAL and pTAL software
- Printer for compilation listings

Learn more at

hpe.com/us/training/nonstop