



Technical Education Services Course Specification

Course Number: EW2800

Course Title: PowerMAX OS™ Real-Time Tools

Course Duration: 5 Days

Purpose:

The Concurrent Series of real-time, multiprocessing, supermicrocomputers provides “off-the-shelf” technology coupled with industry standard interfaces to satisfy real-time processing demands. These systems include unique utilities for controlling processes and analyzing faults, measuring process performance, capturing information about executing processes, and monitoring process interaction. The primary goal of this course is to provide the student with instruction and “hands-on” experience to achieve knowledge on using these tools.

Intended Audience:

This course is intended for software engineers who develop real-time applications on Concurrent systems using the PowerMAX OS™ operating system, Real-Time Services, and the NightStar® Toolset.

Course Objectives:

Upon successful completion of this course students are able to:

- Use the Real-Time Command Processor to create and manage an application running under the control of a Frequency Based Scheduler.
- List the special real-time tools available with the PowerMAX OS™ operating system, among which are those included in the NightStar® Toolset.
- Describe the purpose and features of the Frequency Based Scheduler and associated Performance Monitor in controlling and monitoring a real-time application.
- Explain what the capabilities of the NightSim® tool are and describe how this tool is used to setup and a Frequency Based Scheduler for controlling real-time processes.
- Perform data recording on real-time processes using the NightProbe® utility.
- Describe the capabilities of the NightTrace® profiling tool and explain how it is used to analyze process flow.
- Create customized NightTrace® displays showing application and system process flow.

- Use the NightView® debugger tool to monitor execution of associated real-time processes and resolve faults in a disfunctioning application.

Prerequisites:

- C Programming Language - Students need to be able to read C language source code and understand C language syntactical constructs.
- UNIX System Capability - Students need to understand and be able to use basic UNIX system commands.
- UNIX Programming Capability - Students should understand standard UNIX tools used to create programs or have comparable experience.
- A working knowledge of Ada or FORTRAN programming languages are also supported in this course.

Course Topic Outline:

- I. Real-Time Services..... (10 Hours)
 - A. Frequency-Based Scheduler Concepts
 - B. Performance Monitor Concepts
 - C. **rtcp** Services Interface
 - D. Library Call Interface
 - E. Real-Time Data Monitoring
- II. NightSim® Scheduler Tool..... (3 Hours)
 - A. NightSim® Tool Features
 - B. NightSim® System Requirements
 - C. NightSim® Process Control
 - D. **nsim** Command Syntax
 - E. NightSim® Main Window
 - F. Performance Monitor Utility
- III. NightProbe® Data Recording Tool..... (3 Hours)
 - A. NightProbe® Tool Concepts
 - B. GUI Interface Structure
 - C. Using the Data Monitor Window
 - D. Using the Target Process Window
 - E. Using the Target Attribute Window
 - F. Data Viewer Tools

- IV. NightTrace® Analysis Tool (16 Hours)
- A. Overview of the NightTrace® Tool
 - B. The **ntrace** Data Analysis Tool
 - C. Trace Point Library Calls and the Tracing Daemon **ntraceud**
 - D. Trace Display Components
 - E. Using Expressions
 - F. NightTrace® Built-in Tools
 - G. Kernel-level Tracing
- V. NightView® Debugger Tool..... (8 Hours)
- A. NightView® Concepts
 - B. NightView® Dialogues
 - C. Command Syntax
 - D. Process Control and Examination Commands
 - E. Source File Viewing Commands
 - F. Graphical User Interface
 - G. Dialogue Windows
 - H. Process Control Windows
 - I. On-line Help Interface

Laboratory Exercises:

Exercises are provided for all topics presented and consist of two basic types:

- Review exercises are fill-in type questions that require the student to review the material presented to respond. These questions reinforce the important points presented in each topic.
- Hands-on exercises provide the student with experience in using the commands, utilities, calls, and techniques from the material allowing the student to better understand what he or she has learned.