



Technical Education Services Course Specification

Course Number: RH3400

Course Title: RedHawk™ Linux® Real-Time Tools

Course Duration: 5 Days

Purpose:

The iHawk™ Series is Concurrent Computer Corporation's high-performance PCI-based computer platform for real-time data acquisition, simulation, and industrial systems applications. These systems include unique utilities for controlling processes, 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 RedHawk Linux operating system, Real-Time services, and the NightStar™ Development Tool Package.

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 RedHawk Linux operating system, among which are those included in the NightStar Tools.
- 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 malfunctioning application.
- Use NightTune™ to monitor system performance and load balance applications on an iHawk computer system.

Prerequisites:

- C Programming Language – Students need to be able to read C language source code and understand C language syntactical constructs.
- Linux System Capability – Students need to understand and be able to use basic Linux system commands.
- Linux Programming Capability – Students should understand standard Linux tools used to create programs or have comparable experience.

Course Topic Outline:

I. Real-Time Services

- A. Frequency-Based Scheduler Concepts
- B. Performance Monitor Concepts
- C. Real-Time Command Processor Services Interface
- D. Library Call Interface
- E. Real-Time Data Monitoring

II. NightSim Scheduler Tool

- A. NightSim Tool Features
- B. NightSim System Requirements
- C. NightSim Process Control
- D. NightSim Command Syntax
- E. NightSim Main Window
- F. Performance Monitor Utility

III. NightProbe Data Recording Tool

- 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

- A. Overview of the NightTrace Tool
- B. The NightTrace Data Analysis Tool
- C. Trace Point Library Calls and the Tracing Daemon
- D. Trace Display Components
- E. Using Expressions
- F. NightTrace Built-in Tools

G. Kernel-level Tracing

V. NightView Debugger Tool

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

VI. NightTune

A. NightTune Concepts

B. Monitoring/Tuning Processes

C. Tuning Applications

D. Controlling CPUs

E. Monitoring System Activity

Laboratory Exercises:

Students are provided with the opportunity to perform hands-on exercises for topics presented and may 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.