concurrent REAL-TIME

Technical Education Services Course Specification

Course Number: RH3300

Course Title: RedHawk™ Linux® Real-Time Programming and NightStar™ 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. RedHawk Linux software interfaces include methods for controlling and scheduling processes, managing memory pools, communicating between processes, performing I/O, synchronizing processes, and optimizing process performance. Real-time application engineers need to understand what tools are available for these purposes and how to use them effectively on a Concurrent system. Concurrent's NightStar Tools 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 this level of knowledge.

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:

- Describe the special system functions contained in the RedHawk Linux operating system that support Real-Time applications.
- Explain how to optimize RedHawk Linux to provide Real-Time scheduling policies that enhance response from application processes.
- Explain methods to effectively manage processes running on a shared-bus, multiprocessor system.
- Create and use shared memory regions for inter-process communication between different parts of a Real-Time application.
- Describe the process synchronization tools available under RedHawk Linux and use them in a Real-Time application environment.
- Describe the POSIX interface capabilities available under RedHawk Linux and explain how to use them to support a Real-Time application.

- Write Real-Time programs using the rich set of features that RedHawk Linux provides.
- 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 Overview
   - A. Real-Time Applications
   - B. RedHawk Linux Real-Time Support
   - C. Kernel Tuning and Building
   - D. Process Access Privileges

II. Process Management
   - A. Basic System Architecture
   - B. Process Creation under RedHawk Linux
   - C. Process Priority Classes
   - D. Scheduling Administration
   - E. Real-Time Signal Processing using POSIX Calls

III. Memory Management
   - A. Physical Configuration

Laboratory Exercises:

Students are provided with the opportunity to perform hands-on exercises for topics presented and may consist of two basic types:

- Example source code that the student may use to run, modify, and explore various concepts to reinforce topics presented during lecture.
- 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.