Course Number: RH3200
Course Title: RedHawk™ Linux® Real-Time Programming
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. The 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. 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.

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.

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
   B. Resident Processes
   C. Shared Memory Support Techniques
   D. POSIX Message Queues
IV. File and Device I/O
   A. POSIX Clocks and Timers
   B. POSIX Synchronized I/O
   C. POSIX Asynchronous I/O
   D. Real-Time Device I/O
V. Process Synchronization
   A. POSIX Counting Semaphores
   B. Rescheduling Control Variables
   C. User-level Spin-Locks
   D. Client-Server System Calls
VI. Program Optimization
   A. Compiler Optimization Options
   B. Compiler Warnings
   C. Process Dispatch Latency
D. Shielded Processor Model
E. Increasing Determinism Considerations

VII. Thread Programming Overview
A. Concurrent Programming Considerations
B. Basic Thread Management
C. Thread Types and Scheduling
D. Thread Synchronization Techniques
E. Thread Program Development

VIII. Real-Time Services Overview
A. Overview of the Frequency-Based Scheduler
B. Overview of the Performance Monitor Utility
C. Data Recording Programming Interface

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.