



Technical Education Services

Course Specification

Course Number: EW2400
Course Title: PowerMAX OS Real-Time Programming
Course Duration: 5 Days

Purpose:

The Night Hawk®, PowerHawk®, and PowerMAXION® Series of real-time, multiprocessing, super microcomputers provides “off-the-shelf” technology coupled with industry standard interfaces to satisfy real-time processing demands. The software interfaces include methods for controlling and scheduling of 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 PowerMAX OS™ operating system.

Course Objectives:

Upon successful completion of this course students are able to:

- Describe the special system functions contained in the PowerMAX OS™ operating system that support Real-Time applications.
- Explain how to optimize PowerMAX OS™ 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 PowerMAX OS™ and use them in a Real-Time application environment.
- Describe the POSIX interface capabilities available under PowerMAX OS™ and explain how to use them to support a Real-Time application.
- Write Real-Time programs using the rich set of features that PowerMAX OS™ provides.

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 Overview (4 Hours)
 - A. Real-Time Applications
 - B. PowerMAX OS™ Real-Time Support
 - C. Kernel Tuning and Building
 - D. Process Access Privileges
- II. Process Management (6 Hours)
 - A. Basic System Architecture
 - B. Process Creation under PowerMAX OS™
 - C. Process Priority Classes
 - D. Scheduling Administration
 - E. Real-Time Signal Processing using POSIX Calls
- III. Memory Management..... (6 Hours)
 - A. Physical Configuration
 - B. NUMA Memory Management
 - C. Resident Processes
 - D. Shared Memory Support Techniques
 - E. POSIX Message Queues
- IV. File and Device I/O..... (4 Hours)
 - A. POSIX Clocks and Timers
 - B. POSIX Synchronized I/O
 - C. POSIX Asynchronous I/O
 - D. Real-Time Device I/O
- V. Process Synchronization..... (6 Hours)
 - A. POSIX Counting Semaphores
 - B. Rescheduling Control Variables

- C. User-level Spin-Locks
- D. Client-Server System Calls
- E. User-level Interrupt Support
- F. Interrupt Level Control Calls
- VI. Program Optimization (4 Hours)
 - A. Compiler Optimization Options
 - B. The **analyze** Optimization Tool
 - C. The **ktrace** Optimization Tool
 - D. Process Dispatch Latency
 - E. Shielded Processor Model
 - F. Increasing Determinism Considerations
- VII. Thread Programming Overview (4 Hours)
 - 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 (6 Hours)
 - A. Overview of the Frequency-Based Scheduler
 - B. Overview of the Performance Monitor Utility
 - C. Data Recording Programming 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.