

QNX Course Curriculum

Real-Time Operating System using QNX

Course Category:	Professional Core (C)
Pre-Requisite Course(s):	Microprocessor/Microcontroller Co-Requisite Course(s): None
Progressive Course(s):	Advanced Embedded Systems
Duration	3 – Lecture 1 – Practical 0 – Tutorial

Course Objectives

By the end of this course, students will:

1. Understand the architecture and core concepts of the QNX RTOS.
2. Learn to develop and debug applications using the QNX Momentics IDE.
3. Gain knowledge of process and thread management, including synchronization techniques.
4. Explore inter-process communication (IPC) methods and their applications in QNX.
5. Understand hardware programming concepts, including interrupt handling and memory access.
6. Learn to build and configure QNX boot/OS images for embedded systems.

Course Outcome

At the end of the course, students will be able to:

1. Describe the QNX OS architecture and its microkernel-based design.
2. Develop and debug QNX-based applications using appropriate tools.
3. Apply process/thread management and synchronization techniques in QNX.
4. Implement inter-process communication methods for real-time systems.
5. Configure and build QNX boot/OS images for specific hardware platforms.
6. Develop strong knowledge on the POSIX standards that help in System Application Development

Course Curriculum

Unit I: Introduction to QNX OS Architecture (Contact Hours: 7)

- Overview of QNX OS architecture: microkernel, process manager, and standards.
- Protected address spaces, process/thread model, and scheduling.
- Introduction to inter-process communication (IPC) and synchronization.
- Resource managers and shared objects.

Unit II: Processes, Threads, and Synchronization (Contact Hours: 6)

- Process management: creation, termination, and memory protection.
- Thread management: creation, termination, and synchronization.
- Synchronization techniques: mutexes, semaphores, and condition variables.
- Hands-on exercises: process/thread creation and synchronization.

Unit III: Inter-Process Communication (IPC) (Contact Hours: 8)

- Overview of IPC methods in QNX: message passing, pulses, and shared memory.
- Comparing IPC methods: advantages and disadvantages.
- Practical implementation of IPC in QNX.
- Hands-on exercises: message passing and shared memory.

Unit IV: Hardware Programming and Timing (Contact Hours: 6)

- Hardware access methods: IO-mapped and memory-mapped IO.
- Interrupt handling and DMA-safe memory allocation.
- Timing architecture: periodic timing, one-shot timing, and timeouts.
- Hands-on exercises: interrupt handling and timing mechanisms.

Unit V: Building and Configuring QNX Boot/OS Images (Contact Hours: 8)

- Overview of QNX boot/OS image structure.
- Components of a boot image: startup code, kernel, drivers, and scripts.
- Building and loading boot images onto target hardware.
- Introduction to resource managers and their implementation.

Lab Experiments

Total Lab Contact Hours: 15

1. QNX configuration and application development using QNX Momentics IDE.
2. Process and thread creation, management, and synchronization.
3. Implementation of IPC methods: passing message and shared memory.
4. Interrupt handling and hardware access programming.
5. Building and deploying QNX boot/OS images.
6. Mini capstone project: Design and implement a QNX-based embedded system.

Learning Assessment

- **Continuous Learning Assessments (50%):** CLA-1 (10%), CLA-2 (5%), CLA-3 (5%), Mid-Semester Exam (15%), Practical (15%).
- **End Semester Exam (50%):** Theory (35%), Practical (15%).

Recommended Resources

1. QNX Neutrino RTOS User's Guide, QNX Software Systems.
2. Programming for Embedded Systems, Michael Barr, O'Reilly Media.
3. Hands-on RTOS with Microcontrollers, Brian Amos, Packt Publishing, 2020.
4. Operating System Concepts, Abraham Silberschatz, Peter B. Galvin, Greg Gagne, 9th Edition, Wiley, 2018.
5. Online Resource
 - QNX online training
 - QNX training material
 - QNX Documentation for SDP 8.0
 - QNX Everywhere page - <https://blackberry.qnx.com/en/products/qnx-everywhere>
 - QNX Gitlab page - <https://gitlab.com/qnx>
 - **QNX Developer Community:** Access to forums and communities where they can ask questions and interact with other QNX developers.
 - QNX on Reddit: <https://www.reddit.com/r/QNX/>
 - QNX on StackOverflow: <https://stackoverflow.com/questions/tagged/qnx>
 - QNX on YouTube: <https://www.youtube.com/qnxcam>

- Collection of open-source projects (incl Links) for QNX:
<https://gitlab.com/qnx/projects>
- Ported libraries that work on QNX: <https://github.com/qnx-ports>

PC and LAB Requirements

- I5 10th gen or Later PC /Laptop with USB Ports and Ethernet Support
- Raspberry Pi 4 Kit
- LCD monitor
- Wireless Keyboard and Mouse

This course provides a comprehensive understanding of QNX RTOS, equipping students with the skills to design and implement real-time systems for embedded applications.