

Développer, déboguer et analyser les performances avec l'Environnement de développement intégré QNX Momentics.

Pré-requis

Vous devez avoir une bonne connaissance générale de la programmation en langage C. Une connaissance de la programmation sous le système d'exploitation temps réel de QNX Neutrino (RTOS) est un atout.

Objectifs:

Apprendre à utiliser l'environnement intégré de l'interface de développement (IDE) de QNX Momentics® pour :

1. Développer vos applications
2. Déboguer des problèmes complexes.
3. Analyser les performances des systèmes

A la fin du cours, le participant aura acquis la pratique de l'IDE de QNX à partir d'un environnement croisé ou natif et réalisé de nombreux exemples de programmation d'applications pour le système QNX Neutrino.

Thèmes

Le cours est constitué des sections suivantes :

Introduction.

Présente quels seront les thèmes couverts.

Architecture du système d'exploitation temps réel QNX Neutrino.

Le cours commence par une vue d'ensemble de QNX Neutrino et met en avance les fonctionnalités du micronoyau de QNX Neutrino et de son gestionnaire de processus. Il traite de la protection mémoire et de son adressage, du modèle de processus et des « threads », du séquenceur et de sa gestion du temps. Les différents types de communication interprocessus, leurs synchronisations, la définition du gestionnaire de ressources, la localisation d'objets partagés et du système graphique Photon de QNX pour le multifenêtrage sont aussi abordés dans cette section

Environnement de Développement Intégré (IDE), fonctions de base.

Cette section donne une vue d'ensemble nécessaire à tout programmeur qui utilise l'environnement de développement intégré IDE QNX Momentics® pour développer des applications et systèmes QNX Neutrino, fournit le concept pour naviguer, configurer et personnaliser l'IDE selon les besoins et les préférences des utilisateurs. Elle met en œuvre aussi la gestion de projets, l'espace de travail et les modèles de connexions entre l'IDE et les cartes cibles avec les différentes façons de s'interfacer avec les cartes.

Gestion des projets C et C++

Le conteneur de codes de base qui est fourni avec l'IDE est un projet. Trois types de projets pour des codes C/C++ sont couverts dans cette section avec un projet ou le contenu et les structures de constructions sont personnalisables, un projet ou les structures sont prédéfinies et qui supporte toutes les plateformes ou famille de processeurs, et le projet des bibliothèques. Les fichiers d'entête, les dépendances du projet et une discussion pour importer du code externe et l'intégrer dans le projet sont aussi abordés.

Edition et compilation.

Cette section se focalise sur l'éditeur de codes de l'IDE, pour montrer toute la puissance de ses fonctionnalités et pour aller plus vite à l'essentiel, mais aussi elle montre comment construire du code, et comment l'IDE identifie les problèmes du code construit.

Exécution et Débogage.

Cette section, couvre comment lancer des programmes, en mode commande et aussi à partir du débogueur de l'IDE. Elle fournit une vue d'ensemble des fonctionnalités fondamentales telles que sauter dans le code, mettre des points d'arrêts et examiner les données.

Autres thèmes de Débogage.

Ce module élargit la notion de débogage en explorant les capacités supérieure de mise au point de l'IDE à celle d'une interface de débogage conventionnelle. Il commence en utilisant l'IDE pour déboguer des processus multithreads et des processus multiples. Il explore ensuite comment déboguer des problèmes de communication interprocessus (IPC), les appels réentrants, et comment assurer un débogage post-mortem, s'attacher à un processus qui est en cours d'exécution et déboguer des bibliothèques partagées.

Débogage de problèmes mémoire.

Une mémoire corrompue peut souvent être corrigés ou évités par des tests exhaustifs ou précis. Cette section explore les outils fournis par l'IDE pour trouver les problèmes de corruption mémoire, les fuites, la consommation de mémoire excessive.

Vue d'ensemble d'un système QNX embarqué.

Avant d'embarquer QNX véritablement sur une carte (cible), il est nécessaire de comprendre quels sont les composants qu'il faut embarquer et comment ils sont associés les uns aux autres. Ces informations sont aussi nécessaires au design du système embarqué. Pour expliquer cela, différentes méthodes de démarrage sont vues, avec leurs composants nécessaires et leurs relations.

Embarquer le RTOS QNX avec l'Environnement de Développement Intégré (IDE) QNX Momentics

Cette section détaille comment utiliser l'IDE pour créer une image de démarrage et ses composants nécessaires. Elle explique ensuite comment créer une image pour embarquer un système de fichiers qui contient les autres composants nécessaires après le démarrage. Souvent le programme d'initialisation (« IPL » - Initial Program Loader), les images ainsi que les autres composants doivent être convertis dans des formes intermédiaires avant d'être intégré sur la cible. Cette section traite comment réaliser la conversion. Elle explore aussi les fonctions de téléchargement sur une carte cible par une liaison série ou par TFTP.

Profiler une application.

Tous les détails sur comment utiliser l'IDE pour profiler un processus simple. En utilisant le profileur, les parties de votre code qui consomment le plus de temps CPU (processeur) sont identifiées aussi bien au niveau des fonctions qu'au niveau de la ligne de source. Le profileur permet de développer du code de manière plus efficace. L'utilisation au niveau exécution ainsi qu'au niveau post-mortem du profileur d'application est couverte dans cette section.

Couverture de code

Déterminer l'efficacité d'une suite de tests nécessite du travail. Cette section montre comment les outils de l'IDE peuvent déterminer combien de code a été réellement exécuté, pointant jusqu'à la ligne du code source par le test.

Profiler le Système.

Avec le profileur du système, le système complet peut être examiné pour comprendre ce qu'il se passe. Les détails sont disponibles au niveau noyau. La section aborde tout ce qui se passe au niveau noyau ainsi que les étapes nécessaires à la configuration. Elle montre comment récupérer les sessions d'activités ou « log » à travers l'IDE, à partir du mode ligne de commande, ou sous le contrôle d'un programme pour ensuite analyser le résultat des données et les afficher à travers différentes perspectives du profileur. La section se termine par comment insérer ses propres données dans des sessions d'activités.

Analyse des performances.

Cette section démontre comment le profilage d'une application et le profilage du système peut aider à comprendre pourquoi une application s'exécute lentement. Elle porte sur l'analyse de deux ou trois systèmes qui n'obtiennent pas les comportements attendus et utilise les outils d'analyse des performances pour déterminer l'origine et la cause de ces mauvais comportements.

Conclusion.

Donne un récapitulatif de ce qui a été appris et où trouver d'autres informations.