

Ecriture de pilotes pour le système d'exploitation temps réel (RTOS) QNX

Pré-requis

Le participant doit avoir une bonne connaissance de générale de la programmation en langage C et une connaissance de la programmation sous le système d'exploitation temps réel (RTOS) de QNX Neutrino.

Objectifs

Ce cours est destiné pour le développement de pilotes pour le système d'exploitation temps réel QNX Neutrinos. A la fin du cours, le participant aura écrit un échantillonnage de pilotes.

Ce cours permet de :

1. Familiariser le développeur avec les méthodes pour développer différents types de pilote pour le système d'exploitation temps réel QNX Neutrinos.
2. Se focaliser sur des problèmes réels et leurs techniques pour les régler.
3. Fournir des exercices pratiques réalisés par le participant pour mettre en œuvre les concepts introduits dans le cours.

Thèmes

Le cours est constitué des sections suivantes :

Introduction

Le cours commence par une vue d'ensemble de QNX Neutrino et des pilotes pour QNX, met en avant les fonctionnalités clés, les services et l'architecture. Cette introduction servira ensuite de fil conducteur pour le reste du cours, pour expliquer comment développer un pilote pour QNX Neutrino.

Edition et compilation.

Cette section est une introduction rapide sur éditer, compiler, exécuter et déboguer une application à partir de l'environnement de développement intégré (IDE) QNX Momentics, elle sera ensuite nécessaire pour la réalisation des exercices pratiques des différentes sections.

Programmation de base de QNX Neutrino

Ce module donne une revue des concepts de bases de la programmation pour QNX Neutrino, tels que ceux appliqués aux threads, mutex, passage de messages, impulsions, et temps. Ce module permet de rappeler les bases au participant qui a une certaine familiarité avec ces concepts mais qui manque de pratique.

Thème avancé pour les gestionnaires de ressources.

Cette section couvre les différentes méthodes pour sortir des gestionnaires d'interruptions, gérer les accès et modifier les temps, laisser les clients bloqués, utiliser des gestionnaires de ressources multithread, implémenter des messages combinés, exécuter des déblocages, recevoir des impulsions, et réaliser des select() et ionotify. Les concepts des gestionnaires de ressources sont renforcés à travers la réalisation d'exercices pratiques.

Entrées/Sorties

Cette section couvre en détail tous les aspects d'un pilote, expliquant comment accéder aux E/S, les déclarer dans la mémoire, comment configurer le DMA de la mémoire, et aussi comment utiliser l'API du PCI.

Interruptions

Si l'on compare QNX Neutrino RTOS à d'autres systèmes d'exploitations conventionnelles, il est plus facile de développer et déboguer un gestionnaire d'interruption pour QNX Neutrino. Cette section présente différentes approches sur les gestionnaires d'interruptions et discute sur la façon de lier les gestionnaires d'interruptions aux gestionnaires de ressources. Un exercice pratique permet d'appliquer les concepts et de les appliquer à des problèmes réels.

Exercices d'E/S

Une collection d'exercices intéressants et pertinents pour faciliter l'apprentissage de l'écriture d'un pilote. Les exercices utilisés dépendent de la disponibilité du matériel. Le participant peut choisir parmi les exercices suivants :

Exercice sur les pilotes d'écran.

Ecriture d'un gestionnaire d'E/S qui réalise les fonctions de lecture read(s), d'écriture write(s) et de contrôle devctl(s) pour changer la configuration de mode VGA.

Exercice sur le pilote de clavier

Ecriture d'un pilote simple pour la gestion d'une entrée clavier sous X86

Gestionnaires de ressources pour systèmes de fichiers.

Une discussion sur comment écrire un gestionnaire de ressources pour un système de fichiers. Quelles sont les procédés pour présenter les données sous forme de fichiers ou de gestionnaire de fichiers. Ce module comporte un exercice pratique sur l'écriture d'un gestionnaire de ressource qui présente un fichier de type tar pour un système de fichiers.

Ecriture sur un pilote de caractères.

Ce module regarde comment utiliser la librairie « io-char » de QNX Neutrino pour l'écriture d'un pilote pour des périphériques caractères tels que des composants de lignes séries qui supportent le mode RS-232.

Exercices pratiques

Chaque section du cours permet de mettre en pratique l'apprentissage du participant et lui permettra d'appliquer chaque technique de programmation aux situations réelles qu'il pourra rencontrer dans ses applications.