

Programme de Formation

QNX ESSENTIALS:

Programmation temps réel avec QNX Neutrino et prise en main des outils "QNX Momentics IDE", introduction à la programmation matérielle, traitement multi-cœurs et partitionnement adaptatif

OBJECTIFS

L'objectif de ce cours est d'inculquer aux développeurs les concepts de la programmation d'application temps réelles sous QNX Neutrino et d'apprendre à utiliser l'environnement de développement intégré QNX Momentics pour développer des applications, déboguer des problèmes complexes et analyser et optimiser les performances des systèmes.

PREREQUIS

Connaissance des commandes de base POSIX ainsi que la connaissance du langage C sont souhaitables.

PROGRAMME

Architecture du système d'exploitation temps réel QNX Neutrino :

Revue des fonctionnalités du micronoyau QNX Neutrino et de son gestionnaire de processus, détail de l'adressage et de la protection mémoire, notion de processus et de « threads », du séquenceur et de sa gestion du temps. Communication interprocessus et synchronisations, gestionnaire de ressources et notions d'objets partagés.

Prise en main de l'Environnement de Développement Intégré (IDE) :

Introduction à « QNX Momentics » pour QNX Neutrino, navigation, configuration et personnalisation de l'IDE. Notions de gestion de projet, d'espace de travail, connexions aux cartes cibles. Exercice d'édition de code, de construction d'application et de compilation. Exécution et débogue sur une cible, point d'arrêt et examen des données.

Synchronisation de processus et de « Threads » :

Définition d'un thread, ses priorités, ses algorithmes de séquençement, lancement et arrêt d'un thread, synchronisation et accès aux ressources, fonctions POSIX.

Communication inter-processus (IPC).

Introduction aux méthodes de communication et de synchronisation entre processus, échange d'informations et contrôle : utilisation des messages synchrones. et asynchrone, des impulsions, et de la mémoire partagée.

Minuteur, horloge et temps échus.

Notion de déterminisme temporel, introduction à la gestion des horloges et du temps (horloge périodique, tic d'horloge, marquage de temps, temps échus et performance), influence de l'architecture matérielle.

Introduction à la programmation du matériel.

Méthodes d'accès au matériel, adressage des E/S et de leurs mémoires, gestion des DMA et des interruptions.

Introduction aux gestionnaires de ressources

Notions de gestionnaire de ressources : résolution de noms de chemin, format des messages IPC et structure générale d'un gestionnaire de ressource. Exercices pratiques.

Profiler une application.

Utilisation de l'IDE pour profiler un processus simple, analyse de la consommation de temps CPU, identification des fonctions chronophages et optimisation du code, utilisation du profileur en mode post-mortem.

Couverture de code

Déterminer l'efficacité d'une suite de tests nécessite du travail utilisation des outils de l'IDE pour déterminer le code source réellement exécuté par le test.

Débogue de problèmes mémoire.

Introduction aux problèmes de corruption mémoire, de fuites et de consommation de mémoire excessive. Prise en main des outils d'analyse fournis par l'IDE.

Profiler le Système.

Introduction au profileur du système, configuration de l'outil, création et analyse des « logs » à travers l'IDE, affichage via le profileur, insertion de traces/données dans un log.

Construction d'une image QNX Neutrino, amorcer un démarrage.

Configuration et amorçage, du micro-noyau, puis du réseau, du gestionnaire de fichiers, des pilotes, des utilitaires et de l'applicatif. Génération d'une image du RTOS avec démarrage automatique. Introduction aux différentes méthodes de boot des systèmes embarqués.

Traitement Multiprocesseurs Symétrique (SMP) et Bound (BMP)

Utilisation de l'IDE pour profiler un processus simple, analyse de la consommation de temps CPU, identification des fonctions chronophages et optimisation du code, utilisation du profileur en mode post-mortem. Notions et mesure des migrations de thread d'un cœur à l'autre, notion d'affinité Thread-Cœur (bound microprocessing). Notion de programmation multi-cœur.

Partitionnement Adaptatif.

Utilisation du partitionnement adaptatif sous QNX neutrino : notions et mise en œuvre, Caractéristiques du « thread scheduler », protection des threads critiques. Considération d'architecture, précaution d'utilisation avec les systèmes multi-cœurs. Utilisation des outils Momentics (avec le noyau instrumenté) pour l'optimisation des budgets CPU.

REFERENCE: TRAIN_CC_EI