

BSP UBoot et Linux embarqué, construire son propre système

Cours Pratique de 5 jours - 35h

Réf : BLE - Prix 2024 : 3 450CHF HT

Cette formation vous permettra de comprendre les étapes nécessaires pour porter Linux sur une nouvelle carte afin d'y adapter votre code métier. Vous apprendrez à configurer et installer le noyau, les bibliothèques système et les utilitaires dans un système Linux embarqué totalement personnalisé.

OBJECTIFS PÉDAGOGIQUES

À l'issue de la formation l'apprenant sera en mesure de :

Comprendre les étapes nécessaires pour la mise en œuvre d'un BSP Linux embarqué sur une nouvelle carte

Connaître les constituants d'un système Linux et les spécificités liées à l'environnement embarqué

Savoir porter et développer efficacement du code applicatif métier sur un système Linux embarqué

TRAVAUX PRATIQUES

Les TP seront réalisés avec des cartes Raspberry Pi.

LE PROGRAMME

dernière mise à jour : 01/2018

1) Linux embarqué

- Licences, implications pour l'embarqué. Principe des drivers.
- Composant d'un système Linux. Boot d'un système embarqué.
- Outils de développement. Chaîne de compilation croisée

Travaux pratiques : Compilation, installation et test de Qemu.

2) Build Systems

- Présentation de Buildroot.
- Construction d'une chaîne de cross-compilation (Crosstool-NG).
- Construction d'un système embarqué avec Buildroot.
- Présentation du Yocto Project.

Travaux pratiques : Création d'une chaîne de cross-compilation. Production d'une image pour Raspberry Pi.

3) Noyau Linux

- Fonctionnalités du noyau. Modèle de développement.
- Configuration et compilation du noyau.
- Application de patches.
- Transfert de l'image du noyau. Paramètres de démarrage du kernel.
- Types de système de fichiers. Formatage et préparation d'une arborescence.

Travaux pratiques : Compilation et test d'un noyau (Buildroot, émulateur Arm, Raspberry Pi). Application d'un patch.

4) Bootloader et Root Filesystem

- Les bootloaders : Lilo et Grub. U-Boot, barebox...
- Installation de U-Boot.

PARTICIPANTS

Architecte ou ingénieur devant porter Linux sur un nouveau matériel ou étendre un système embarqué.

PRÉREQUIS

Bonnes connaissances du langage C et du système Linux.

COMPÉTENCES DU FORMATEUR

Les experts qui animent la formation sont des spécialistes des matières abordées. Ils ont été validés par nos équipes pédagogiques tant sur le plan des connaissances métiers que sur celui de la pédagogie, et ce pour chaque cours qu'ils enseignent. Ils ont au minimum cinq à dix années d'expérience dans leur domaine et occupent ou ont occupé des postes à responsabilité en entreprise.

MODALITÉS D'ÉVALUATION

Le formateur évalue la progression pédagogique du participant tout au long de la formation au moyen de QCM, mises en situation, travaux pratiques...

Le participant complète également un test de positionnement en amont et en aval pour valider les compétences acquises.

MOYENS PÉDAGOGIQUES ET TECHNIQUES

- Les moyens pédagogiques et les méthodes d'enseignement utilisés sont principalement : aides audiovisuelles, documentation et support de cours, exercices pratiques d'application et corrigés des exercices pour les stages pratiques, études de cas ou présentation de cas réels pour les séminaires de formation.
- À l'issue de chaque stage ou séminaire, ORSYS fournit aux participants un questionnaire d'évaluation du cours qui est ensuite analysé par nos équipes pédagogiques.
- Une feuille d'émargement par demi-journée de présence est fournie en fin de formation ainsi qu'une attestation de fin de formation si le stagiaire a bien assisté à la totalité de la session.

MODALITÉS ET DÉLAIS D'ACCÈS

L'inscription doit être finalisée 24 heures avant le début de la formation.

ACCESSIBILITÉ AUX PERSONNES HANDICAPÉES

Vous avez un besoin spécifique d'accessibilité ? Contactez Mme FOSSE, référente handicap, à l'adresse suivante psh-accueil@orsys.fr pour étudier au mieux votre demande et sa faisabilité.

- Boot par TFTP et par NFS.
- Formats pour disques et mémoire flash. Systèmes de fichiers spéciaux.
- Organisation des partitions. Gestion de versions.

Travaux pratiques : Test de Grub. Compilation et test de U-boot sur Raspberry.

5) Espace utilisateur

- Busybox : présentation, configuration.
- Configuration du réseau et des serveurs . Authentification des utilisateurs et connexion distante.

Travaux pratiques : Compilation et installation de Busybox.

6) Code métier

- Outils de développements : compilateurs, interpréteurs, IDE.
- Débogueur, profileur, couverture de code.
- Bibliothèques statiques et dynamiques.
- Gestion de mémoire sous Linux.

Travaux pratiques : Utilisation d'Eclipse, de gdbserver, de Gprof et de Gcov.

7) Drivers et modules spécifiques

- Principe des drivers : intégration de modules externes.
- Principes de la programmation noyau.
- Aperçu de la programmation de drivers en mode caractère.

Travaux pratiques : Compilation et chargement de module externe.

8) Multiprocesseurs et temps réel

- SMP, multicœurs, hyperthreading. Gestion des tâches et des interruptions.
- Temps réel : principes. Temps-réel souple.
- Amélioration avec le patch PREEMPT_RT.
- Temps-réel strict : Présentation de Xenomai.

Travaux pratiques : Configuration des emplacements des tâches et des interruptions.

LES DATES

CLASSE À DISTANCE

2024 : 15 juil., 04 nov.