

Formation Linux Embarqué, Drivers et Temps Réel

Réf. : 5158

Objectifs

Ce stage permet aux développeurs, confrontés aux problèmes de **portage d'un noyau Linux** et des **applications temps réel sur plate-forme linux embarqué**, de pouvoir concevoir une **distribution optimisée et des drivers linux sur mesure**.

Ce stage de formation Linux embarqué aborde les concepts du portage d'un OS Linux sur cible embarquée par l'étude :

- des caractéristiques et architectures des systèmes Linux embarqué
- de la mise en œuvre d'une chaîne de développement croisé
- de la compilation d'un noyau et l'installation sur ROM/FLASH NAND et NOR
- de la préparation d'un BSP et d'un boot-loader Linux pour l'embarqué
- des packages et de la configuration de l'installation sur différents types d'architectures matérielles x86 et ARM9
- de la mise au point et la validation de modules et de pilotes de périphériques linux
- des API et des extensions temps réel sous Linux embarqué

Programme

1^{ère} journée

Distributions Linux Embarqué

- Définition des systèmes embarqués
- Projets existants: MontaVista, Bluecat linux, uCLinux, eldk, PeeWee Linux...
- Méthodes, outils et chaîne de développement croisé

Noyau Linux - Présentation

- Vue d'ensemble du système et rôle du noyau
- Historique, numérotation des versions
- Architectures matérielles supportées - Support processeurs et File system
- Spécificités des noyaux 2.4 et 2.6

Noyau Linux – Configuration et Compilation pour l'Embarqué

- Structure des sources et modules du noyau
- Fichiers utilisés par les outils de configuration
- Configuration, optimisation et Compilation d'un noyau
- Compilation croisée - Méthodes, outils et chaîne de développement croisés,

Méthodes et outils de validation

- Outils GNU de mise au point et portage d'une chaîne de débogueur à distance : serveur gdb
- Mise au point par port série, par réseau, par sonde JTAG (Abatron BDM BDI2000)
- IDE Eclipse pour développeur linux embarqué : Eclipse C/C++
- Debugueur Eclipse et pluggin Linuxscope JTAG Target Debugger pour sonde BDI 2000

Travaux pratiques

- Préparation d'une chaîne de développement croisé (compilateur et débogueur embarqué) pour cible ARM 9
- Configuration et compilation d'un noyau 2.6 « patché » pour cible ARM 9

2^{ème} journée

Services et configurations Linux embarqué

- Personnalisation du système : Script de démarrage et de configuration du système Linux embarqué
- Chargement des pilotes de périphériques
- Installation des services réseaux : Console série Inetd, Rsh, telnet, Nfs

Environnement utilisateur sous Linux embarqué

- Choix de bibliothèques : LibC : glibc, uClibc, NewLibC
- Shell et utilitaire en console pour l'embarqué : Busybox
- Console série, gestion de sessions utilisateurs
- Connection réseau : remote shell rsh, telnet, ...
- Portage et configuration d'environnements graphiques X: nanoX, XFree, QtE/QTopia, MicroWindows, ...

Travaux pratiques

- Outils de configuration uCLibC Buildroot pour cible ARM et validation du File System Linux complet sur émulateur QEMU

Processus Utilisateurs sous Linux embarqué

- Gestion de la mémoire virtuelle et application embarquée : page stack, overcommit memory ...
- Démons Unix, Socket et application réseau TCP/IP Client Serveur...

3^{ème} journée

Mise au point du Boot Loader et du kernel Linux sur système embarqué

- Préparation du boot loader U-BOOT, Setup de l'architecture et utilisation des commandes U-BOOT en console
- Paramétrage du chargeur de démarrage « bootloader »
- BSP et Init de Linux sur système embarqué : configuration du mapping E/S physique et mémoire RAM/FLASH de la cible

Travaux pratiques

- Utilisation de BSP Linux embarqué : application de patch au Kernel Linux officiel
- Mise en œuvre d'une configuration de boot loader U-BOOT sur plateforme ARM 9
- Débogage du boot loader et du noyau par sonde JTAG (Abatron BDM BDI2000) et GDB

Portage d'une image système Linux embarqué

- Technologies MTD : Flash Chip NOR et NAND, Disk Flash: CompactFlash, DiskOnChip,...
- Systèmes de fichiers, outils de génération d'image de file system Linux CRAMFS, JFFS2 et initramfs en mémoire Flash
- Portage d'un Shell et d'outils d'administration (BusyBox ...)

Travaux pratiques

- Mise en œuvre d'une configuration de boot loader U-BOOT sur plateforme ARM 9
- Installation d'un système bootable via réseau et montage nfs avec U-BOOT sur cible TI OMAP ARM 926

Introduction au développement Kernel Mode sous Linux

- Introduction à la programmation en mode noyau
- Architecture d'un module linux simple
- Gestion de paramètres de modules, communication avec les systèmes de fichiers sysfs et le procfs

4^{ème} journée

Introduction au développement de pilotes

- Pilotes de périphériques sous Linux et système de fichiers devfs
- Programmation de pilotes de périphériques simples : Structure de la File Operation
- API du noyau Linux et gestion mémoire en Kernel Mode
- Programmation de pilotes de périphériques Linux, Signaux et Timer en kernel mode
- Driver bloquant, gestion d'interruption sous Linux
- Cas particuliers des périphériques USB, SPI et SDIO sur système embarqué
- Intégration de codes sources personnels dans le système de configuration et de compilation du kernel Linux
- Création de patches pour noyau Linux

Travaux pratiques

- Construction et compilation de pilotes de périphériques pour carte E/S PC104 Linux embarqué
- Débogage de modules et pilotes de périphériques linux par sonde JTAG (Abatron BDM BDI2000) et GDB

5^{ème} journée

Introduction aux applications embarquées temps réel en mode utilisateur

- API POSIX temps réel souple: ordonnancement des processus et signaux UNIX et POSIX sous Linux
- Programmation multi-thread et extension API PThread , NPTL, ...
- Résolution des Timers Linux et choix du Tick system pour l'embarqué

Extensions temps réel embarquées en mode noyau sous Linux

- Temps réel dur: Patches low-latency, préemptifs, temps réel RTAI/RTLinux
- Introduction à la programmation temps réel : Module RTAI, Tâche et ordonnanceur temps réel
- Mécanisme de communication et synchronisation entre tâches temps réel RTAI
- Timer temps réel RTAI et gestion d'IT
- LTT, outils d'évaluation et de mise au point temps réel sous linux : Linux Trace Tools Kit

Interface temps réel dur et en mode Utilisateur

- FIFO temps réel RTAI entre module RTAI et processus Linux
- LXRT : mécanisme de communication et synchronisation entre tâches temps réel RTAI et Thread Linux
- Le framework Xenomai : API temps réel embarqué en mode noyau et utilisateur
- Xenomai et RTDM : modèle de driver temps réel

Travaux pratiques

- Programmation d'une tâche temps réel de traitement périodique RTAI à 10Khz
- Mise en œuvre de LTT et analyse de traces temps réel RTAI

Participants

Cette formation est particulièrement **adaptée aux techniciens et ingénieurs**, confrontés aux problèmes de portage de solutions **linux sur systèmes embarqués, drivers et temps réel**.

Matériel à disposition des participants

Les travaux pratiques de la formation Linux Embarqué, Drivers et Temps Réel sont réalisés sur :

- station de travail Redhat ou Fedora Core
- cibles ARM9 TI OMAP, et AXIS ETRAX 100
- cibles PC104 de type Acrosser Geode 300Mhz et Seco M570

- sonde JTAG Abatron BDI2000

Supports de la formation Linux Embarqué, Drivers et Temps Réel :

- Classeur de cours
- CD-ROM sources des exercices

Modalités

Sessions interentreprises :

- Tarif : **1 920 euros HT**
- Durée : 5 jours
- Formateur : Chef de projet Bureau d'Etudes CénoSYS

Sessions spéciales: *Calendrier et tarifs, nous consulter*

- Organisation de session en interentreprises sur demande
- Possibilité de session en intra-entreprise (adaptée sur mesure)

© **2007 CénoSYS** - 10 rue Xavier Bichat - 72000 Le Mans - France
Tel: +33(0) 243 511 797 - Fax: +33(0) 243 511 798 - contact@cenosys.com