




Formation temps-réel sous Linux avec *PREEMPT_RT*

Durée de la formation —

 2 jours – 16 h

Langue —

Transparents Anglais


Présentation Français
 Anglais


Formateur —

Un des ingénieurs suivants

- Maxime Chevallier

Contact —

 training@bootlin.com

 +33 4 84 25 80 96

Public visé

Entreprises et ingénieurs intéressés dans le développement et le benchmarking d'applications et de drivers temps-réel pour un système Linux embarqué.

Objectifs opérationnels

- Être capable de comprendre et de maîtriser les caractéristiques d'un système d'exploitation temps-réel
- Être capable de télécharger, compiler et utiliser le patch *PREEMPT_RT*
- Être capable d'identifier et de benchmarker la plateforme matérielle en terme de caractéristiques temps-réel
- Être capable de configurer le noyau Linux pour un comportement déterministe
- Être capable de développer, de tracer et de déboguer des applications user-space temps-réel.

Prérequis

- **Connaissance et pratique des commandes UNIX ou GNU/Linux** : les participants doivent être à l'aise avec l'utilisation de la ligne de commande Linux. Les participants manquant d'expérience sur ce sujet doivent se former par eux-mêmes, par exemple en utilisant nos supports de formation.
- **Expérience minimale en développement Linux embarqué** : les participants doivent avoir une compréhension minimale de l'architecture d'un système Linux embarqué : rôle du noyau Linux par rapport à l'espace utilisateur, développement d'applications espace utilisateur en C. Suivre la formation Linux embarqué de Bootlin permet de remplir ce pré-requis.
- **Niveau minimal requis en anglais : B1**, d'après le *Common European Framework of References for Languages*, pour nos sessions animées en anglais. Voir la grille CEFR pour une auto-évaluation.

Méthodes pédagogiques

- Présentations animées par le formateur : 50% de la durée de formation
- Travaux pratiques réalisés par les participants : 50% de la durée de formation
- Version électroniques de supports de présentation, des instructions et des données de travaux pratiques. Les supports sont librement disponibles [ici](#).

Modalités d'évaluation

Seuls les participants qui auront assisté à l'intégralité des journées de formation, et qui auront obtenu plus de 50% de réponses correctes à l'évaluation finale recevront une attestation individuelle de formation de la part de Bootlin.

Handicap

Les participants en situation de handicap qui ont des besoins spécifiques sont invités à nous contacter à l'adresse training@bootlin.com afin de discuter des adaptations nécessaires à la formation.



Équipement nécessaire

Pour les sessions en présentiel délivrées chez nos clients, notre client devra fournir :

- Projecteur vidéo
- Un ordinateur sur chaque bureau (pour une ou deux personnes), avec au moins 16 Go de RAM et Ubuntu Linux 24.04 installé dans une partition dédiée d'au moins 30 Go.
- Les distributions autres que Ubuntu Linux 24.04 ne sont pas supportées, et l'utilisation de Linux dans une machine virtuelle n'est également pas supportée.
- Connexion à Internet rapide et sans filtrage : au moins 50 Mbit/s de bande passante en téléchargement, et pas de filtrage des sites Web et protocoles.
- Les ordinateurs contenant des données importantes doivent être sauvegardés avant d'être utilisés dans nos sessions.

Pour les sessions en présentiel organisés dans les locaux de Bootlin, Bootlin fournit l'ensemble de l'équipement.

Plateforme matérielle pour les travaux pratiques

Plateforme STM32MP1

Une de ces cartes de STMicroelectronics :
STM32MP157A-DK1, **STM32MP157D-DK1**, **STM32MP157C-DK2** ou **STM32MP157F-DK2**

- Processeur STM32MP157, double Cortex-A7, de STMicroelectronics
- Alimentée par USB
- 512 Mo DDR3L RAM
- Port Gigabit Ethernet port
- 4 ports hôte USB 2.0
- 1 port USB-C OTG
- 1 connecteur Micro SD
- Debugger ST-LINK/V2-1 sur la carte
- Connecteurs compatibles Arduino Uno v3
- Codec audio
- Divers : boutons, LEDs
- Écran LCD tactile (uniquement sur cartes DK2)



Jour 1 - Matin

Cours	Introduction au comportement temps-réel et au déterminisme	<ul style="list-style-type: none">▪ Définition d'un système d'exploitation temps-réel▪ Spécificités des systèmes multi-tâches▪ Principaux patterns de verrouillage et de gestion des priorités▪ Aperçu des systèmes temps-réel existants▪ Approches pour apporter un comportement temps-réel à Linux
Cours	Le patch <i>PREEMPT_RT</i>	<ul style="list-style-type: none">▪ Histoire et avenir du patch <i>PREEMPT_RT</i>▪ Améliorations temps-réel provenant de <i>PREEMPT_RT</i> dans le noyau Linux officiel▪ Fonctionnement interne de <i>PREEMPT_RT</i>▪ Gestion des interruptions : interruptions threadées, softirqs▪ Primitives de verrouillage : mutexes et spinlocks, spinlocks avec sommeil▪ Modèles de préemption
TP	Compiler un noyau Linux avec <i>PREEMPT_RT</i>	<ul style="list-style-type: none">▪ Télécharger le noyau Linux et appliquer le patch <i>PREEMPT_RT</i>▪ Configurer le noyau Linux▪ Démarrer le kernel sur une plateforme matérielle

Jour 1 - Après-midi

Cours	Configuration et limites du matériel pour le temps-réel	<ul style="list-style-type: none">▪ Interruptions et firmware▪ Interaction avec les fonctionnalités de gestion d'énergie : gestion dynamique de la fréquence du CPU et états de sommeil▪ DMA
Cours	Outils : Benchmarking, Stress et Analyse	<ul style="list-style-type: none">▪ Benchmarking avec <i>cyclictest</i>▪ Stress du système avec <i>stress-ng</i> et <i>hackbench</i>▪ L'infrastructure de <i>tracing</i> du noyau Linux▪ Analyse de la latence et de l'ordonnancement avec <i>ftrace</i>, <i>kernelshark</i> ou <i>LTTng</i>
TP	Outils : Benchmarking, Stress et Analyse	<ul style="list-style-type: none">▪ Utilisation des outils de benchmark et de stress▪ Techniques classiques de benchmarking▪ Benchmarking et configuration de la plateforme matérielle

Jour 2 - Matin

Cours	Infrastructures du noyau Linux et configuration	<ul style="list-style-type: none">▪ Bonnes pratiques pour le développement de drivers noyau Linux pour des systèmes temps-réel▪ Politiques d'ordonnancement et priorités : <i>SCHED_FIFO</i>, <i>SCHED_RR</i>, <i>SCHED_DEADLINE</i>▪ Affinité CPU et IRQ▪ Gestion mémoire▪ Isolation des CPUs avec <i>isolcpus</i>
Cours	Patterns de développement d'applications temps-réel	<ul style="list-style-type: none">▪ API POSIX pour les applications temps-réel▪ Gestion et configuration des threads▪ Gestion mémoire : allocation mémoire et verrouillage mémoire, gestion de la pile▪ Patterns de verrouillage : mutexes, héritage de priorité▪ Communication inter-processus (IPC)▪ Signalisation

Jour 2 - Après-midi

TP	Débugger une application de démonstration	<ul style="list-style-type: none">▪ Créer une application de démonstration déterministe▪ Utiliser l'infrastructure de <i>tracing</i> pour identifier la source de latence▪ Apprendre à utiliser l'API POSIX pour gérer les threads, le verouillage, la mémoire▪ Apprendre à utiliser l'affinité CPU et configurer la politique d'ordonnement
----	---	---