



Formation temps-réel sous Linux avec *PREEMPT_RT*

Séminaire en ligne, 3 sessions de 4 heures

Dernière mise à jour : 01 July 2024

Titre	Formation temps-réel sous Linux avec <i>PREEMPT_RT</i>
Objectifs opérationnels	<ul style="list-style-type: none">• 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 <i>PREEMPT_RT</i>• 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.
Durée	Trois demi-journées - 12 h (4 h par demi-journée)
Méthodes pédagogiques	<ul style="list-style-type: none">• Présentations animées par le formateur, par visioconférence. Les participants peuvent poser des questions à tout instant.• Démonstrations pratiques réalisées par le formateur, basés sur les travaux pratiques de la formation, par vidéo-conférence. Les participants peuvent poser des questions à tout instant. Optionnellement, les participants qui ont accès aux accessoires matériels de la formation peuvent reproduire par eux-même les travaux pratiques.• Messagerie instantanée pour questions entre les sessions (réponse sous 24h, hors week-end et jours fériés)• Version électronique des supports de présentation, des instructions et des données de travaux pratiques. Les supports sont librement disponibles sur https://bootlin.com/doc/training/preempt-rt.
Formateur	Maxime Chevallier https://bootlin.com/company/staff/maxime-chevallier/
Langue	Présentations : Français Supports : Anglais



Public visé	Entreprises et ingénieurs intéressés dans le développement et le benchmarking d'applications et de drivers temps-réel pour un système Linux embarqué.
Pré-requis	<ul style="list-style-type: none">• 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 disponible à l'adresse bootlin.com/blog/command-line/.• 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 <i>Linux embarqué</i> de Bootlin, disponible sur bootlin.com/training/embedded-linux/, permet de remplir ce pré-requis.• Niveau minimal requis en anglais : B1, d'après le <i>Common European Framework of References for Languages</i>, pour nos sessions animées en anglais. Voir bootlin.com/pub/training/cefr-grid.pdf pour une auto-évaluation.
Équipement nécessaire	<ul style="list-style-type: none">• Ordinateur avec le système d'exploitation de votre choix, équipé du navigateur Google Chrome ou Chromium pour la conférence vidéo.• Une webcam et un micro (de préférence un casque avec micro)• Une connexion à Internet à haut débit
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.



Plateforme matérielle pour les travaux pratiques

Carte STMicroelectronics STM32MP157D Discovery Kit 1

- Processeur STM32MP157D (dual Cortex-A7) de STMicroelectronics
- Alimentation par USB
- 512 MB DDR3L RAM
- Ethernet Gigabit
- 4 ports USB 2.0 hôte
- 1 port USB-C OTG
- 1 slot Micro SD
- Debugger ST-LINK/V2-1
- Connecteurs compatibles Arduino
- Codec audio, boutons, LEDs



1^{ère} demi-journée

Cours - Introduction au comportement temps-réel et au déterminisme

- 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 *PREEMPT_RT*

- Histoire et avenir du patch *PREEMPT_RT*
- Améliorations temps-réel provenant de *PREEMPT_RT* dans le noyau Linux officiel
- Fonctionnement interne de *PREEMPT_RT*
- Gestion des interruptions : interruptions threadées, softirqs
- Primitives de verouillage : mutexes et spinlocks, spinlocks avec sommeil
- Modèles de préemption

Démo - Compiler un noyau Linux avec *PREEMPT_RT*

- Télécharger le noyau Linux et appliquer le patch *PREEMPT_RT*
- Configurer le noyau Linux
- Démarrer le kernel sur une plateforme matérielle

Cours - Configuration et limites du matériel pour le temps-réel

- Interruptions et firmware
- Interaction avec les fonctionnalités de gestion d'énergie : gestion dynamique de la fréquence du CPU et états de sommeil
- DMA

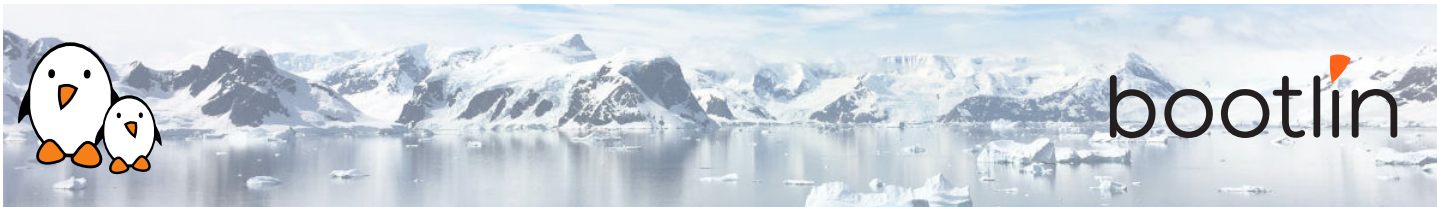
2^{ème} demi-journée

Cours - Outils : Benchmarking, Stress et Analyse

- Benchmarking avec *cyclicttest*
- Stress du système avec *stress-ng* et *hackbench*
- L'infrastructure de *tracing* du noyau Linux
- Analyse de la latence et de l'ordonnement avec *ftrace*, *kernelshark* ou *LTTng*

Démo - Outils : Benchmarking, Stress et Analyse

- Utilisation des outils de benchmark et de stress
- Techniques classiques de benchmarking
- Benchmarking et configuration de la plateforme matérielle



Cours - Infrastructures du noyau Linux et configuration

- Bonnes pratiques pour le développement de drivers noyau Linux pour des systèmes temps-réel
- Politiques d'ordonnancement et priorités : *SCHED_FIFO*, *SCHED_RR*, *SCHED_DEADLINE*
- Affinité CPU et IRQ
- Gestion mémoire
- Isolation des CPUs avec *isolcpus*

3^{ème} demi-journée

Cours - Patterns de développement d'applications temps-réel

- API POSIX pour les applications temps-réel
- Gestion et configuration des threads
- Gestion mémoire : allocation mémoire et verouillage mémoire, gestion de la pile
- Patterns de verrouillage : mutexes, héritage de priorité
- Communication inter-processus (IPC)
- Signalisation

Démo - Débugger une application de démonstration

- Créer une application de démonstration déterministe
- Utiliser l'infrastructure de *tracing* 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'ordonnancement

Questions / réponses

- Questions / réponses avec les participants autour du noyau Linux
- Des présentations supplémentaires s'il reste du temps, selon les sujets qui intéressent le plus les participants.