

Une virée open source à Minneapolis : mes découvertes du Open Source Summit 2026

Open Source Summit 2026 • Minneapolis

 André Courchesne





THE LINUX FOUNDATION

OPEN SOURCE SUMMIT
NORTH AMERICA



Embedded Linux
Conference

Qui suis-je



Développeur logiciel senior
chez Intello by Telus Business



Fondateur et inventeur
des solutions de surveillance de la qualité de l'air VisiblAir



Inventeur et créateur —
je donne vie aux idées sous forme de produits concrets



Plus de 25 ans
à utiliser Linux et les technologies open source



VisiblAir



<https://visiblair.com>



LinkedIn



[https://www.linkedin.com/
in/netforces](https://www.linkedin.com/in/netforces)



OSS 2026 : Le Code Source Complet



12 Conférences d'ouverture (Keynotes)



Plus de 165 conférences majeures



12 parcours simultanés



Estimation de plus de 1200 participants sur site

Présentations auxquelles j'ai assisté — Jour 1

OPEN SOURCE SUMMIT 2026 • MINNEAPOLIS



Keynote: Welcome + Opening Remarks —
Jim Zemlin, CEO, The Linux Foundation



Keynote: UCP: The Evolution of an Open Standard for Agentic Commerce —
Anurag Sinha, Senior Staff Software Engineer & Manager, Google



Keynote: The First Decade of Open Quantum —
Sean Dague, Chief Services Architect, IBM Quantum



Keynote: From Open Source to Agentic Systems: Building the AI Native Era —
Brendan Burns, Technical Fellow



The Revolution Hiding in Plain Sight: CI/CD Platform Is About to Change Forever —
Dadisi Sanyika, Sol Duara, Inc.



Debug Everything: Building a Debuginfod Backbone for Embedded Linux at Scale —
Colin Pinnell McAllister & Joshua Pevehouse, Garmin



Lightning Talk: CI/CD Cybersecurity Guide — Open Source Tools to Improve DevOps Security —
Kate Scarcella, Independent



AI in CI/CD Without the Hype: Practical Patterns for Platform Engineers —
Jennifer Mulford, Okta



KernelScript: Unifying EBPf, Userspace, and Kernel Extensions in One Language —
Cong Wang, Multikernel Technologies



Image Composer Tool: Declarative Multi-Distro Linux Image Builds From Packages —
Mats Agerstam & Alpesh Rodage, Intel Corporation



From FreeRTOS To Zephyr: A Practical Migration Guide for Embedded Developers —
Jacob Beningo, Beningo Embedded Group

Présentations auxquelles j'ai assisté — Jour 2

OPEN SOURCE SUMMIT 2026 • MINNEAPOLIS



1. **Keynote Panel: From Repo to Rocketship: How Open Source Foundations Supercharge AI Ecosystems** — Deepyaman Datta, Open Source Maintainer; Jakub Kuderski, AMD; Libby Clark, AWS; Lin Sun, Solo.io; Stephen Chin, Neo4j



2. **Keynote: Strands - From Framework to Harness** — Belle Guttman, Senior Software Engineering Manager, AWS



3. **Keynote: Where AI Meets the Physical World: The Robot MCP Ecosystem as an Open Bridge Between AI and Robotics** — Rohit John Varghese, Director of Systems Engineering and Product, Contoro Robotics



4. **Corporate Vice President, Azure Cloud Native and Management Platform & Co-Founder, Kubernetes Open Source Project**



5. **State of Embedded Linux** — Walt Miner, The Linux Foundation



6. **Easy Bring-up Your RISC-V SBC Using Yocto Project - RISC-V Architecture Layer** — Khem Raj, Comcast



7. **Securing Software Workflows** — Tabatha DiDomenico, G-Research Open Source; Kádi McKean, ReversingLabs; Stacey Potter, OpenSSF; Katherine Druckman, JetBrains



8. **Architecting for Onboarding: Building a "Docs-as-Code" Pipeline for Open Source Sustainability** — Sai Sravan Cherukuri, Independent Contributor



9. **Secure Boot for Embedded Linux: Explained in Simple Words** — Roy Jamil, Ac6



10. **Optimize Linux Kernel To Fit Microcontrollers With 1 MB RAM** — Jim Huang & Chisheng Chen, National Cheng Kung University

Présentations auxquelles j'ai assisté — Jour 3

OPEN SOURCE SUMMIT 2026 • MINNEAPOLIS



Keynote: Linus Torvalds,

Creator of Linux & Git, in Conversation with Dirk Hohndel, Founder, DH Consulting



OpenEmbedded / Yocto BoF —

Colin McAllister, Garmin & Chuck Wolber, The Boeing Company



Keynote: How Maintainers Can Build Their Way Through the AI Flood —

Madelyn Olson, Valkey Project Maintainer and Principal Engineer, AWS & Jacob Murphy, Valkey Project Maintainer and Software Engineer, Google



It Works on My Bench (And Nowhere Else): DevOps for Embedded Systems —

Colleen Lake, GitLab



Keynote: Zephyr: By Developers, For Developers —

Kate Stewart, VP Dependable Embedded Systems, The Linux Foundation



Using Embedded Linux for Autonomous Robot Control —

Chloe Zhu, The Admissions Authority



Keynote: Free to Use, Not Free to Run: Reinventing Package Registries —

Robin Bender Ginn, Executive Director, OpenJS Foundation



Building the Simplest Possible Linux System —

Rob Landley, Hobbyist

L'IA et la sécurité dans le monde de l'open source

Open Source Summit 2026 • Minneapolis

1



L'exploitation est désormais plus rapide que les correctifs

Délai moyen avant exploitation :

-7 jours

2018 : 63 jours → 2022 : 32 jours
→ 2024 : 5 jours → **2026 : -7 jours**

2



L'IA accélère à la fois les développeurs et les attaquants

Le cycle classique découverte → divulgation → correctif → déploiement a été conçu pour un adversaire plus lent.

3



La sécurité doit intervenir plus tôt

Les projets open source ont besoin d'analyses automatisées, de SBOM, de builds reproductibles, de paramètres sécurisés par défaut et de flux de travail plus rapides pour les mainteneurs.

4



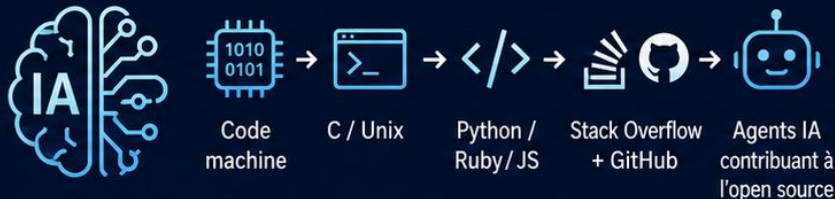
Les mainteneurs font désormais partie du périmètre de sécurité

Les registres de paquets, les dépendances et les pipelines CI/CD sont des infrastructures stratégiques.

L'IA et l'open source : accélération et pression

Open Source Summit 2026 • Minneapolis

1 L'IA est la prochaine couche d'abstraction



2 L'open source devient le socle



Le nouveau niveau d'abstraction ne remplace pas l'OSS — il en fait la bibliothèque standard sur laquelle l'IA s'appuie.

3 Le travail se redistribue



31%

hausse prévue des embauches techniques

8%

croissance des postes TI d'entrée



Montée en compétences
plutôt que recrutement externe

4 Les modèles ouverts rattrapent rapidement leur retard



La parité entre modèles fermés et ouverts est passée à 3–6 mois

~6×

moins chers

~90%

des performances de pointe

30%

d'utilisation mondiale

L'IA dans le CI/CD, sans sensationnalisme

Des approches concrètes pour les ingénieurs de plateforme — Jennifer Mulford, Okta

1



La vitesse ne remplace pas le jugement

Un résultat plus rapide ne remplace pas une validation humaine rigoureuse.

2



L'IA est un "yes-man" : surveillez-la de près

Il faut remettre ses suggestions en question et observer son comportement.

3



L'IA assiste, l'humain décide

La révision humaine doit rester une étape obligatoire.

4



L'injection de prompt devient une nouvelle menace CI/CD

Les issues, PR, MR et commentaires peuvent influencer des flux pilotés par l'IA.

5



Établir une reddition de comptes pour l'IA

Les décisions doivent être visibles, révisables et auditable.



Utiliser l'IA pour accélérer la livraison, pas pour remplacer le jugement.

Le processus : l'IA assiste, l'humain décide



Suggestion IA

L'IA analyse et propose des actions ou options.



Révision humaine

Un humain examine, questionne et décide.



Action prise

Le flux continue selon la décision humaine.



Piste d'audit

Chaque décision est tracée et consultable.



Nouvelle menace : injection de prompt



Issues

Descriptions et titres peuvent être manipulés.



PR / MR

Changements et métadonnées peuvent contenir des pièges.



Commentaires

Commentaires et mentions peuvent influencer le flux.



Pipeline piloté par l'IA

Les entrées peuvent orienter des actions assistées par l'IA.



Des entrées malicieuses ou manipulatrices peuvent influencer des actions assistées par l'IA.

Image Composer Tool

Création déclarative d'images Linux multi-distros à partir de paquets – Mats Agerstam & Alpesh Rodage, Intel Corporation

1



Builds déclaratifs

On construit des images Linux à partir de définitions de paquets plutôt qu'avec des scripts fragiles.

2



Prise en charge multi-distros

Plusieurs distributions Linux et plusieurs versions sont prises en charge.

3



Nombreux formats de sortie

Peut générer des images RAW, ISO, VHD, VHDX, QCOW2 et VMDK.

4



Builds reproductibles

L'épinglage des versions de paquets améliore la répétabilité, la traçabilité et l'auditabilité.

5



État du projet

L'installation non interactive est en cours; le redimensionnement automatique n'est pas encore offert.



À retenir



Épinglage des versions de paquets



Traçabilité orientée manifeste / SBOM



Souplesse multi-distros



Conçu pour simplifier la création d'images Linux sur mesure



Projet open source



github.com/open-edge-platform/image-composer-tool

Une approche prometteuse pour créer des images Linux répétables à partir de paquets, sur plusieurs distros.



Moins de scripts, plus de reproductibilité.

Secure Boot pour Linux embarqué

Expliqué simplement — Roy Jamil, Ac6

1



Exécuter seulement le logiciel approuvé

Le Secure Boot vérifie le code avant de l'exécuter.

2



La confiance commence dans le matériel

Boot ROM et eFuses ancrent la racine de confiance.

3



Chaque étape vérifie la suivante

ROM → bootloader → U-Boot → noyau Linux → rootfs.

4



La signature reste simple

Hash + signature privée; clé publique pour vérifier.

5



Connaître les limites

Ne corrige pas les bogues, ne protège pas tout l'espace utilisateur, et ne sauve pas une clé divulguée.

Chaîne de confiance



Boot ROM



SPL / TF-A



U-Boot



Noyau + DTB



Rootfs /
dm-verity

Erreurs fréquentes

- DTB non signé
- Console U-Boot ouverte
- Fuses non brûlés
- Clés sur un laptop de dev
- Tester seulement le bon scénario



Faire confiance au démarrage — mais sécuriser aussi tout ce qui suit.



Optimiser le noyau Linux pour tenir dans des microcontrôleurs avec 1 Mo de RAM

Jim Huang & Chisheng Chen, National Cheng Kung University

1



Linux sans MMU devient réaliste

Faire tourner Linux sur des microcontrôleurs n'est plus seulement expérimental.

2



Le XIP est l'élément clé

Le noyau et les applications s'exécutent directement depuis la flash pour réduire l'usage de la RAM.

3



La démonstration à 1 Mo a fonctionné

Sur STM32H750, Linux 7.x démarre avec busybox et environ 196 Kio de RAM libre après le démarrage.

4



Les bibliothèques partagées restent possibles

FDPIC permet l'édition de liens dynamique et des bibliothèques partagées de style Linux sur les systèmes sans MMU.

5



Un système plus riche est possible

Sur STM32F429, ils ont montré un prototype avec graphisme, écran tactile et interface de type téléphone.

Comment ils l'ont fait tenir



tinyconfig



XIP
depuis la flash



rootfs
ROMFS



liaison
dynamique
FDPIC



ajustements
noyau / DTB /
libc



Objectif : garder le code en flash et ne conserver en RAM que les données modifiables.

Résultats marquants

- STM32H750 : Linux v7.x sur un microcontrôleur avec 1 Mo de RAM
- Le nettoyage du device tree inutilisé a fait passer la RAM libre de 116 Kio à 196 Kio
- Le prototype STM32F429 démarre avec environ 6180 Kio de RAM libre
- Le XIP a permis d'économiser environ 332 Kio de RAM dans la configuration la plus riche



Linux sur de tout petits MCU, c'est réel : moins de gaspillage de RAM, plus de réutilisation et une meilleure maintenabilité.

Bâtir le système Linux le plus simple possible

Rob Landley, Hobbyiste

1



Toybox garde l'essentiel compact

Toybox regroupe de nombreux utilitaires Linux dans un seul petit exécutable multicall sous licence BSD.

2



L'objectif : un Linux minimal et auto-hébergé

L'idée est de construire le système Linux le plus simple capable de se reconstruire à partir du code source.

3



Peu de briques sont vraiment nécessaires

À la base, on retrouve : des utilitaires en ligne de commande, un compilateur, une bibliothèque C et un noyau.

4



Ce n'est pas qu'un exercice théorique

Toybox est utilisé dans AOSP / Android et contribue à une chaîne d'outils plus propre et auto-suffisante.

5



La simplicité est la grande leçon

En ramenant Linux à l'essentiel, la présentation montre ce qu'il faut vraiment pour démarrer, construire et comprendre un système.

Pourquoi c'était marquant

- Un excellent rappel qu'on peut comprendre Linux à partir des premiers principes
- Petit, rapide et raisonnablement conforme aux standards
- Très utile pour l'apprentissage, le bootstrap et l'expérimentation
- Une approche minimaliste peut améliorer la maintenabilité

Références open source



landley.net/toybox/



codeberg.org/landley/toybox

Toybox propose aussi une cible "make root" pour créer un chroot simple, puis, avec les sources du noyau, générer un petit système amorçable pour QEMU.



Parfois, la meilleure façon de comprendre Linux, c'est d'en construire la version la plus simple possible.

Linux embarqué et RTOS : grands thèmes du sommet

Open Source Summit North America 2026

1



Un sujet majeur à la conférence

Le Linux embarqué, les systèmes de build, les choix de RTOS et l'ingénierie de plateforme revenaient constamment dans les présentations.

2



Yocto était partout

Utilisé dans des organisations comme Boeing, Garmin et Honda pour bâtir des systèmes Linux embarqués.

3



Yocto dans les grands produits

Vega OS d'Amazon a illustré une orientation basée sur Yocto, en remplacement de Fire OS basé sur Android.

4



Zephyr vs FreeRTOS

Un thème récurrent : écosystème, portabilité, outillage et maintenabilité à long terme.

5



ROS + IA

ROS (Robot Operating System) est de plus en plus jumelé à l'IA pour des plateformes robotiques plus capables.

Linux au-delà des systèmes généraux



- **AGL** — Automotive Grade Linux pour les véhicules connectés et les plateformes automobiles embarquées



- **SGL** — Space Grade Linux pour l'aérospatial et les usages orientés espace



- Le Linux embarqué demeure une base naturelle dès que les systèmes dépassent le simple firmware



- La conférence a montré Linux à toutes les échelles, des petits appareils aux plateformes spécialisées

Grande idée à retenir



Le Linux embarqué n'est pas un créneau marginal : c'est une fondation commune pour l'automobile, la robotique, l'aérospatial et l'industriel.

Yocto s'est démarqué comme base de construction commune, pendant que les choix entre RTOS et Linux demeurent des décisions d'architecture centrales.



De Yocto aux robots, aux véhicules et aux systèmes spatiaux : le Linux embarqué était partout.

RISC-V et Linux : pourquoi c'est important

Open Source Summit North America 2026



1



Un grand thème de l'architecture

Le support de RISC-V dans le noyau Linux revenait souvent dans les présentations sur le Linux embarqué et l'ingénierie de plateforme.

2



RISC-V vs x86

RISC-V est une architecture de jeu d'instructions ouverte; x86 est une architecture propriétaire très mature dominée par Intel et AMD.

3



Pourquoi RISC-V attire l'attention

Sa conception modulaire permet de créer des SoC sur mesure tout en visant une ISA commune et la pile logicielle Linux.

4



L'élan côté Linux

Le support du noyau, des outils, du démarrage et des cartes continue de progresser, surtout pour les systèmes embarqués et edge.

5



Où on en est

RISC-V progresse rapidement, mais son écosystème demeure moins mature que celui de x86 dans l'informatique grand public.

x86 vs RISC-V



x86

- ✓ Écosystème poste de travail et serveur très mature
- ✓ Excellentes performances et compatibilité
- ✓ Beaucoup d'héritage et de complexité
- ✓ ISA contrôlée par un petit nombre d'acteurs



RISC-V

- ✓ ISA ouverte et plus facile à personnaliser
- ✓ Très intéressant pour l'embarqué, l'éducation et la recherche
- ✓ Écosystème Linux en croissance
- ✓ Support matériel et logiciel moins mature

Avantages et inconvénients



Avantages

- Standard ouvert, sans verrouillage de licence ISA
- Extensions personnalisées et grande flexibilité
- Bien adapté aux appareils embarqués et spécialisés
- Favorise la diversité de l'écosystème



Inconvénients

- Moins de systèmes et de cartes grand public
- Écosystème outils et micrologiciel encore en maturation
- Compatibilité logicielle plus limitée dans certains cas
- Le haut de gamme est moins mûr que x86



RISC-V ne remplacera pas x86 du jour au lendemain — mais pour Linux, c'est une **alternative ouverte de plus en plus sérieuse pour l'embarqué, l'edge et les plateformes de demain.**



OPEN SOURCE SUMMIT
NORTH AMERICA



EMBEDDED LINUX
CONFERENCE

Linus Torvalds : Linux, IA et sécurité

Discussion d'ouverture avec Dirk Hohndel • Open Source Summit North America 2026



-  **Les divulgations de sécurité évoluent**
Le noyau Linux documente maintenant plus clairement le processus de divulgation des bogues de sécurité, avec des consignes mises à jour pour les signalements.
-  **Les signalements assistés par l'IA changent les règles**
Devant l'avalanche de doublons, si l'IA a été utilisée pour identifier un bogue, celui-ci doit être considéré comme une information publique plutôt que comme une divulgation privée.
-  **Sashiko annonce la revue de code agentique**
Torvalds a mis en avant Sashiko, un système agentique de revue de code pour le noyau Linux, comme exemple de la façon dont l'IA pourrait aider les flux de travail d'ingénierie.
-  **L'IA locale sera la prochaine grande vague**
Un message fort était que l'IA utile tournera de plus en plus localement, au plus près des développeurs et des utilisateurs.
-  **L'IA change la programmation, pas les fondements**
Les pratiques de programmation vont évoluer, mais les fondamentaux de l'ingénierie restent essentiels. Rappel utile : c'est toujours le compilateur qui produit le code exécutable final.

Idées marquantes

- 600 000 projets GitHub sont considérés comme critiques pour l'entreprise.
- La vitesse et l'automatisation ne remplacent pas le jugement d'ingénierie.
- L'IA peut aider les développeurs, mais elle ne change pas les fondamentaux de la conception système.
- Le volume de rapports entrants rend la rigueur des processus encore plus importante.

Références / exemples

- [Processus de divulgation de sécurité Linux : docs.kernel.org/process/security-bugs.html](https://docs.kernel.org/process/security-bugs.html)
- [Sashiko : revue de code agentique pour le noyau Linux](#)
- [Projet parallèle GuitarPedal : github.com/torvalds/GuitarPedal](https://github.com/torvalds/GuitarPedal)

Le projet GuitarPedal apportait une touche plus personnelle et rappelait que l'expérimentation reste importante.



L'IA change notre façon de programmer — mais pas l'importance d'une bonne ingénierie, d'un processus rigoureux et du jugement humain.



OPEN SOURCE SUMMIT
NORTH AMERICA



EMBEDDED LINUX
CONFERENCE



Enregistrements, diapositives et photos

Open Source Summit 2026 • Minneapolis



Enregistrements

Regarder les sessions enregistrées.



YouTube

youtube.com/@LinuxfoundationOrg/videos



Diapositives

Parcourir le matériel de présentation.



Sched

osselcna2026.sched.com/overview/audience/Yes?iframe=no



Photos

Voir l'album photo officiel de la conférence.



Flickr

flickr.com/photos/linuxfoundation/albums/72177720333089343/

Scannez les codes QR pour accéder aux ressources officielles.

Open Source Summit North America 2027

 Prochaine destination : **Vancouver**

 17–19 mai 2027

 Vancouver



<https://events.linuxfoundation.org/open-source-summit-north-america/>

En savoir plus

— Scannez le code QR —



Questions ?

Merci !

Open Source Summit 2026 • Minneapolis

