



Rust pour linux, où en est-on ?

Louis Chauvet

louis.chauvet@bootlin.com

Capitole du libre - Novembre 2023

© Copyright 2004-2023, Bootlin.

Licence Creative Commons BY-SA 3.0.

Dernière mise à jour : 27 novembre 2023

Mises à jour du document et sources :

<https://bootlin.com/pub/conferences/>

Les corrections, suggestions, contributions et traductions sont les bienvenues !





- ▶ Ingénieur Linux embarqué à Bootlin
 - **Expertise** en matière de Linux embarqué
 - **Développement**, conseil et formation
 - Forte orientation vers les logiciels libres
- ▶ Vit à **Toulouse**, France



Qu'est ce que *Rust for Linux* ?



Qu'est ce que le projet



Driver



Qu'est ce que le projet



Driver

APIs kernel



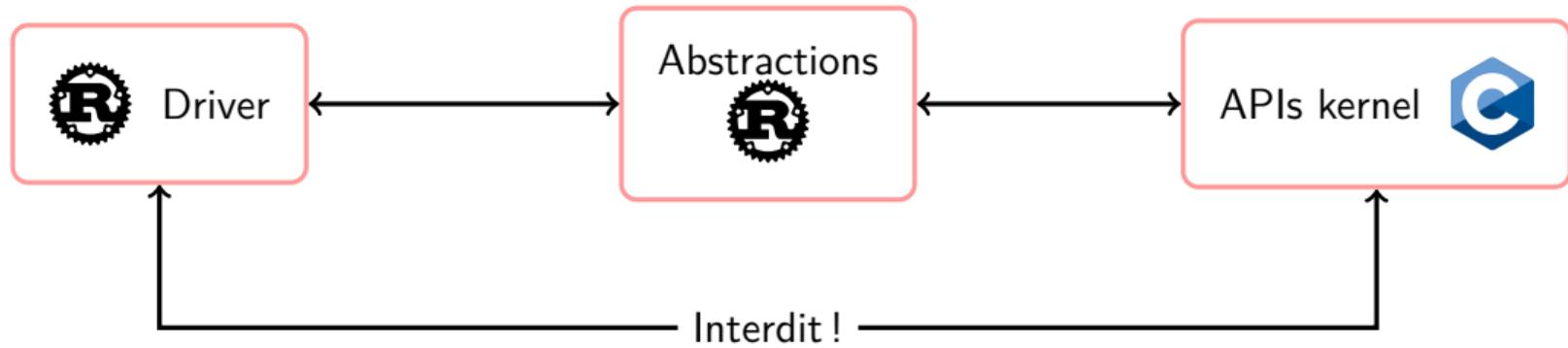


Qu'est ce que le projet





Qu'est ce que le projet





Petite chronologie



- 2019
- Linux Security Summit - Présentation d'un projet pour créer des modules kernel en Rust¹

1. <https://github.com/fishinabarrel/linux-kernel-module-rust>



Petite chronologie

- 2019 • Linux Security Summit - Présentation d'un projet pour créer des modules kernel en Rust ¹
- 2020 • Linux Plumbers Conference - Première conférence sur le sujet par les initiateurs du projet ²

-
1. <https://github.com/fishinabarrel/linux-kernel-module-rust>
 2. <https://lpc.events/event/7/contributions/804/>



Petite chronologie

- 2019 • Linux Security Summit - Présentation d'un projet pour créer des modules kernel en Rust¹
- 2020 • Linux Plumbers Conference - Première conférence sur le sujet par les initiateurs du projet²
- 2021 • Premiers financements d'entreprises
- Création du groupe Rust-for-Linux³
- Premier workshop Kangrejos⁴

-
1. <https://github.com/fishinabarrel/linux-kernel-module-rust>
 2. <https://lpc.events/event/7/contributions/804/>
 3. <https://lwn.net/ml/linux-kernel/20210414184604.23473-1-ojeda@kernel.org/>
 4. <https://kangrejos.com/>



Petite chronologie

- 2019 • Linux Security Summit - Présentation d'un projet pour créer des modules kernel en Rust¹
- 2020 • Linux Plumbers Conference - Première conférence sur le sujet par les initiateurs du projet²
- 2021 • Premiers financements d'entreprises
• Création du groupe Rust-for-Linux³
• Premier workshop Kangrejos⁴
- 2022 • Intégration dans le kernel mainline pour la 6.1
• Premiers drivers complexes : NVME, Binder IPC Android, GPU pour apple M1/M2...

1. <https://github.com/fishinabarrel/linux-kernel-module-rust>

2. <https://lpc.events/event/7/contributions/804/>

3. <https://lwn.net/ml/linux-kernel/20210414184604.23473-1-ojeda@kernel.org/>

4. <https://kangrejos.com/>



[rustc_codegen_gcc](#) – Antoni Boucher

Compiler du code Rust à l'aide de GCC au lieu de LLVM

Actuellement : Compile et boot un kernel mainline avec QEMU

https://github.com/rust-lang/rustc_codegen_gcc

[GCC Rust \(gccrs\)](#) – Arthur Cohen, Philip Herron

Un compilateur rust intégré à gcc

Actuellement en cours d'upstream dans GCC, première release visée 14.1

<https://github.com/Rust-GCC/gccrs>

[Coccinelle for Rust](#) – Julia Lawall, Tathagata Roy

Un outil de vérification principalement utilisé dans le kernel.

Vient d'être publié.

<https://gitlab.inria.fr/coccinelle/coccinelleforrust>



Pourquoi changer ?

5. <https://ldpreload.com/p/kernel-modules-in-rust-lssna2019.pdf>



Pourquoi changer ?

Quels sont les problèmes du C ?

5. <https://ldpreload.com/p/kernel-modules-in-rust-lssna2019.pdf>



Quels sont les problèmes du C ?

La gestion de la mémoire

5. <https://ldpreload.com/p/kernel-modules-in-rust-lssna2019.pdf>



Quels sont les problèmes du C ?

La gestion de la mémoire

- ▶ Gestion de la libération



Quels sont les problèmes du C ?

La gestion de la mémoire

- ▶ Gestion de la libération
- ▶ Buffers et pointeurs mal manipulés



Quels sont les problèmes du C ?

La gestion de la mémoire

- ▶ Gestion de la libération
- ▶ Buffers et pointeurs mal manipulés
- ▶ Mémoire non initialisée



Quels sont les problèmes du C ?

La gestion de la mémoire

- ▶ Gestion de la libération
- ▶ Buffers et pointeurs mal manipulés
- ▶ Mémoire non initialisée
- ▶ Race condition (accès à une donnée partagé)



Quels sont les problèmes du C ?

La gestion de la mémoire

- ▶ Gestion de la libération
- ▶ Buffers et pointeurs mal manipulés
- ▶ Mémoire non initialisée
- ▶ Race condition (accès à une donnée partagé)
- ▶ ...



Quels sont les problèmes du C ?

La gestion de la mémoire

- ▶ Gestion de la libération
- ▶ Buffers et pointeurs mal manipulés
- ▶ Mémoire non initialisée
- ▶ Race condition (accès à une donnée partagé)
- ▶ ...

Plus de **la moitié** des vulnérabilités découvertes sont des problèmes mémoire !⁵

- ▶ 49% pour Chrome, 72% pour Firefox, 81% d'après Google Project Zero
- ▶ 88% pour macOS, 70% pour Microsoft, 65% pour Ubuntu et Android, ...

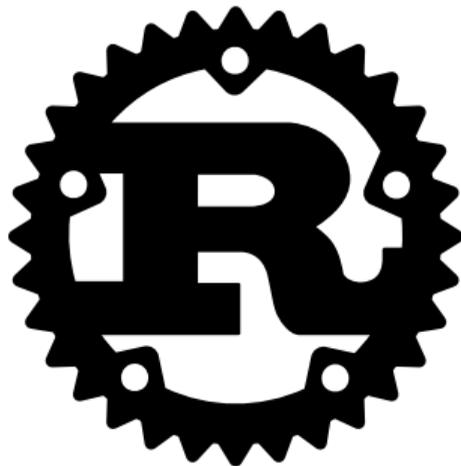
5. <https://ldpreload.com/p/kernel-modules-in-rust-lssna2019.pdf>



Pourquoi Rust ?



Pourquoi Rust ?





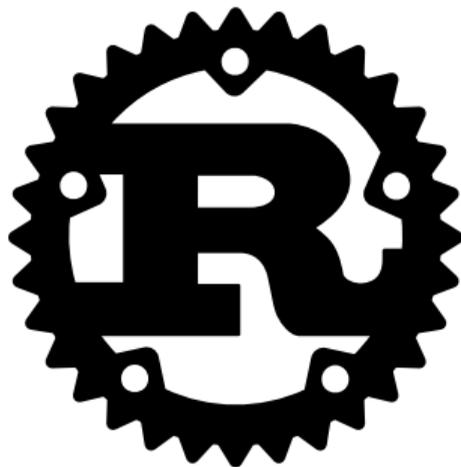
Pourquoi Rust ?



- ▶ Langage "sûr", gestion de la mémoire stricte



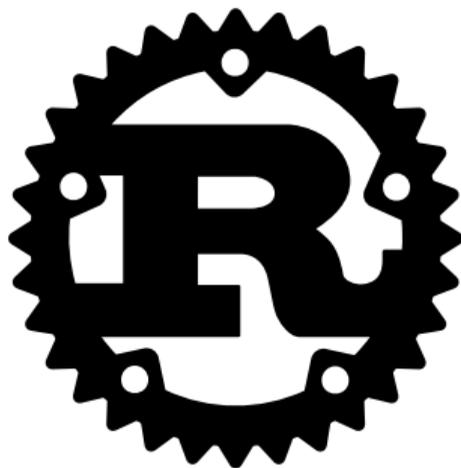
Pourquoi Rust ?



- ▶ Langage "sûr", gestion de la mémoire stricte
- ▶ Convention d'appel compatible avec le C



Pourquoi Rust ?



- ▶ Langage "sûr", gestion de la mémoire stricte
- ▶ Convention d'appel compatible avec le C
- ▶ Pas de garbage collector



Aujourd'hui



Objectifs du projet



Objectifs du projet

- ▶ Expérimentation grandeur nature
 - Pas prêt pour la production
 - Pas d'objectif clair à atteindre pour l'instant



Objectifs du projet

- ▶ Expérimentation grandeur nature
 - Pas prêt pour la production
 - Pas d'objectif clair à atteindre pour l'instant
- ▶ Écriture de nouveaux drivers
 - Extension des interfaces disponibles avec chaque nouveau driver



Objectifs du projet

- ▶ Expérimentation grandeur nature
 - Pas prêt pour la production
 - Pas d'objectif clair à atteindre pour l'instant
- ▶ Écriture de nouveaux drivers
 - Extension des interfaces disponibles avec chaque nouveau driver
- ▶ Ne pas remplacer l'existant
 - Ne pas remplacer core
 - Ne pas remplacer des drivers existants fonctionnant correctement



Comment l'utiliser



Comment l'utiliser

- ▶ Utilisation de LLVM pour la compilation du kernel : `make LLVM=1`



Comment l'utiliser

- ▶ Utilisation de LLVM pour la compilation du kernel : `make LLVM=1`
- ▶ Choix de la version Rust : `rustup override set $(scripts/min-tool-version.sh rustc)`
 - Nécessaire pour l'instant à cause de l'utilisation de certaines features non stables de Rust.



Comment l'utiliser

- ▶ Utilisation de LLVM pour la compilation du kernel : `make LLVM=1`
- ▶ Choix de la version Rust : `rustup override set $(scripts/min-tool-version.sh rustc)`
 - Nécessaire pour l'instant à cause de l'utilisation de certaines features non stables de Rust.
- ▶ Installation de certains modules nécessaires :
 - `rustup component add rust-src`
 - `cargo install --locked --version $(scripts/min-tool-version.sh bindgen) bindgen-cli`



Comment l'utiliser

- ▶ Utilisation de LLVM pour la compilation du kernel : `make LLVM=1`
- ▶ Choix de la version Rust : `rustup override set $(scripts/min-tool-version.sh rustc)`
 - Nécessaire pour l'instant à cause de l'utilisation de certaines features non stables de Rust.
- ▶ Installation de certains modules nécessaires :
 - `rustup component add rust-src`
 - `cargo install --locked --version $(scripts/min-tool-version.sh bindgen) bindgen-cli`
- ▶ Activation de rust dans le kernel : `CONFIG_RUST`
 - Supporté uniquement par certaines plateformes (au moment de la rédaction uniquement x86-64 en mainline, des patchs arrivent pour AArch64 !)



Comment l'utiliser

- ▶ Utilisation de LLVM pour la compilation du kernel : `make LLVM=1`
- ▶ Choix de la version Rust : `rustup override set $(scripts/min-tool-version.sh rustc)`
 - Nécessaire pour l'instant à cause de l'utilisation de certaines features non stables de Rust.
- ▶ Installation de certains modules nécessaires :
 - `rustup component add rust-src`
 - `cargo install --locked --version $(scripts/min-tool-version.sh bindgen) bindgen-cli`
- ▶ Activation de rust dans le kernel : `CONFIG_RUST`
 - Supporté uniquement par certaines plateformes (au moment de la rédaction uniquement x86-64 en mainline, des patchs arrivent pour AArch64 !)

Plus d'infos : <https://docs.kernel.org/rust/quick-start.html>



Petit aperçu



Exemple C

```
#include <linux/module.h>

static int __init example_init(void)
{
    pr_info("C minimal sample\n");
    return 0;
}

static void __exit example_exit(void)
{
}

module_init(example_init);
module_exit(example_exit);
MODULE_LICENSE("GPL");
MODULE_AUTHOR("Kernel Contributors");
```



Exemple C

```
#include <linux/module.h>

static int __init example_init(void)
{
    pr_info("C minimal sample\n");
    return 0;
}

static void __exit example_exit(void)
{
}

module_init(example_init);
module_exit(example_exit);
MODULE_LICENSE("GPL");
MODULE_AUTHOR("Kernel Contributors");
```

Exemple Rust

```
use kernel::prelude::*;

module! {
    type: RustMinimal,
    name: "rust_minimal",
    author: "Rust for Linux Contributors",
    description: "Rust minimal sample",
    license: "GPL",
}

struct RustMinimal {}

impl kernel::Module for RustMinimal {
    fn init(_module: &'static ThisModule)
        → Result<Self> {
        pr_info!("Rust minimal sample\n");
        Ok(RustMinimal {})
    }
}
```



Conclusion



Conclusion

- ▶ Projet bien avancé, toujours en développement
- ▶ Permettrait une écriture plus simple des drivers
- ▶ Permet d'éviter de nombreuses erreurs si les bindings sont bien écrits
- ▶ Pour l'instant
 - Très limité en mainline
 - Beaucoup de bindings et drivers en cours de développement :
Binder, NVMe, DRM (Apple GPU), VFS (tarfs, PuzzleFS), PHY, mitigations, ...



Sources principales :

- ▶ Information générales : <https://rust-for-linux.com/>
- ▶ Documentation du projet :
<https://www.kernel.org/doc/html/next/rust/index.htm>
- ▶ Liste de diffusion : <https://lore.kernel.org/rust-for-linux/>
- ▶ NetDev 0x17 - Rust for Linux Networking Tutorial

Questions ? Suggestions ? Commentaires ?

Louis Chauvet
louis.chauvet@bootlin.com

Bootlin recrute !

Ingénieur – Stagiaire

Drivers et support matériel linux/u-boot

Sécurité des BSP linux

Amélioration d'elixir.bootlin.com

jobs@bootlin.com

Slides sous CC-BY-SA 3.0

<https://bootlin.com/pub/conferences/>