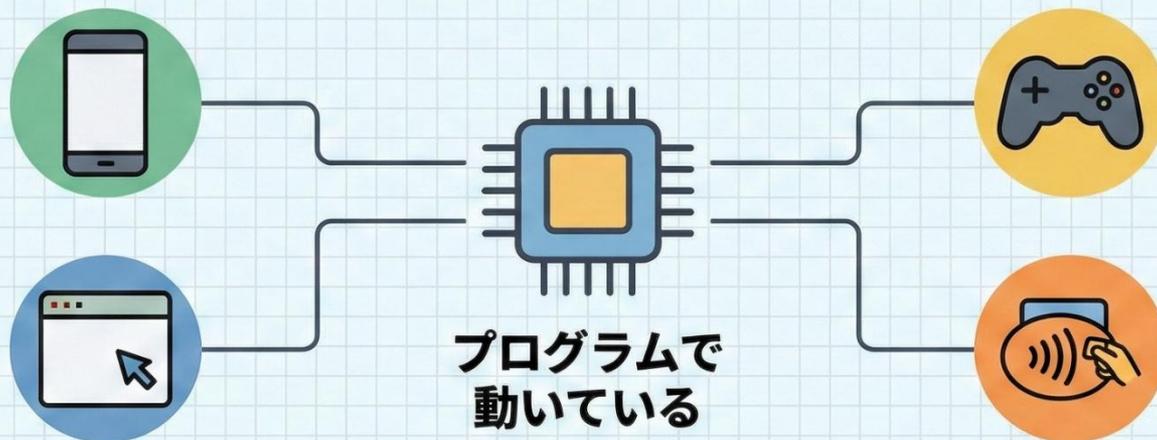


あなたが使うアプリを動かしているRust



プログラムとは、コンピュータへの指示を書いたもの

この指示を書くための言葉がプログラミング言語であり、Rustはその一つ

Rustの特徴



安全性



速度



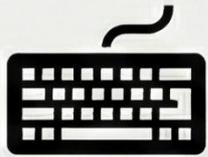
信頼性

社会での採用状況

大手企業での採用が進む
多くのアプリやシステムで活躍

Rust の特徴 (1)

従来のプログラミング



コード作成

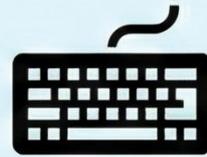


実行中に
誤りが発覚



修正

Rust



コード作成



実行前に
誤りを検出



安全に実行

Rustの特徴 (2)

プログラムの誤りが引き起こす問題

- 途中で止まる
- 予期しない動作
- 攻撃に利用される



Microsoftの発表 (2019年)

セキュリティ欠陥の約70%がメモリ管理ミス



Rustの利点

- 誤り検出

コンピュータのメモリ管理

メモリとは？



コンピュータの
一時的な「作業机」。
プログラム動作中、
データをここに置く。

課題：「片付け」の管理



いつ、誰が片付ける？



忘れる → ゴミが溜まる



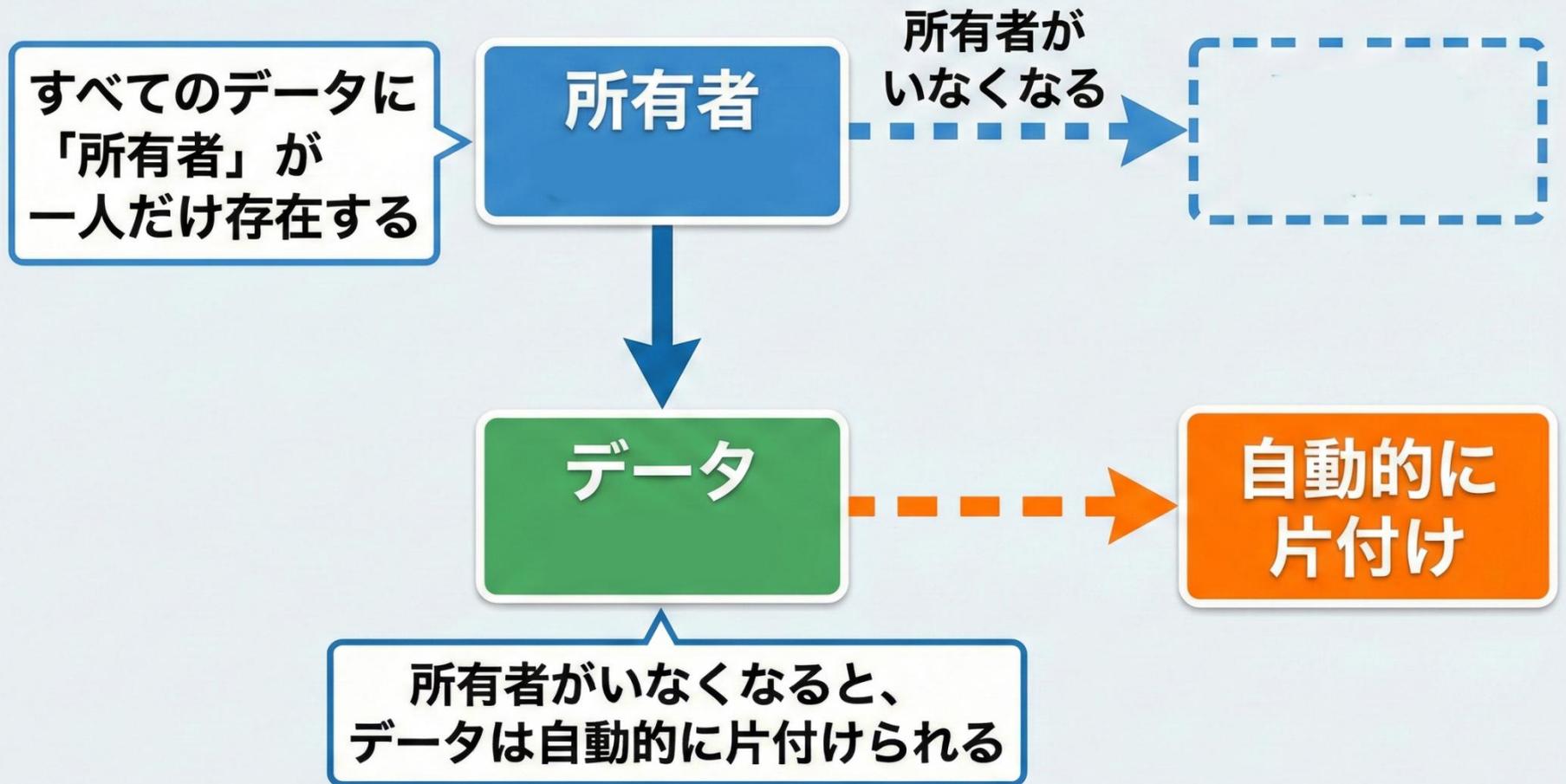
使用中を片付ける
→ 動作がおかしくなる



Rustの解決策：「所有権」

「誰がデータを持つか」を明確にするルール。
この仕組みで、誤りを事前に防ぐ。

Rustの所有権の仕組み



この規則により、「誰が片付けるか」が常に明確になる

Rust のプログラム例

1. メモリに「こんにちは」を記録 (s1 が所有者)

```
let s1 = String::from("こんにちは");
```

~~s1~~

メモリ

こんにちは

2. s2 = s1 で、所有権がs2へ移動 (s1は無効)

```
let s2 = s1;
```

s2

3. 移動後にs1を使おうとすると、Rustが誤りとして検出



結果: メモリ管理の誤り防止。

Rustの普及状況 (1)

Discordでの採用事例



通話機能付きコミュニケーションサービス

Go言語

書き換え (2020年)



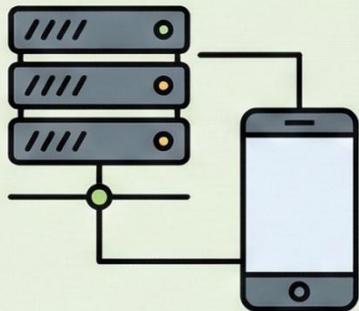
Rust

結果

- ✔ 性能が向上
- ✔ 遅延の問題が解消

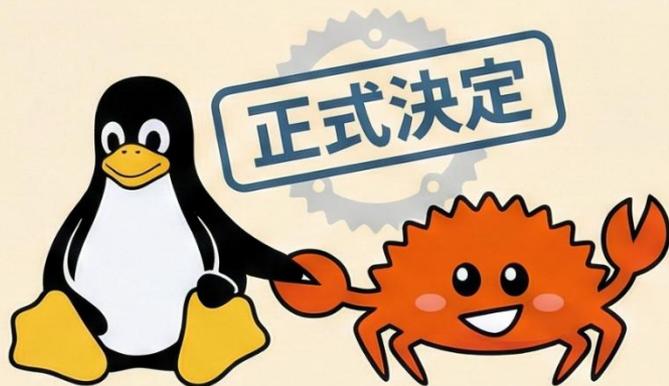
Rustの普及状況 (2)

Linuxとは



サーバーやスマホの
土台となるシステム

Rustの採用 (2025年12月)



2025年12月：
Linux開発者がRustを
正式言語に決定

全体まとめ

安全性の比較



従来：実行時に失敗



Rust：実行前に発見

所有権メカニズム

所有者



データ



所有者が不在 ⇒ 自動で片付け

採用実績



実績：Linux、Discord