



# 情報科学演習Ⅲ 第1期

## xv6のRustへの移植

4/20-5/13

理学部情報科学科4年



# xv6

- 教育用OS
- MITがUnix Version 6をx86に再実装した



# 内容

1. 動機
2. 移植の結果
3. 追加した機能



# 動機

- システムプログラミング実験からの興味
- Rustを学ぶきっかけにしたい
- AIの進歩



# 内容

1. 動機
2. 移植について
3. 追加した機能



# コードの変化

行数	kernel	user	mkfs
xv6-riscv	6215	5234	303
xv6-riscv-rust	7420	5289	404

- Kernelのファイルの比較
  - xv6-riscv | .c : 23 files / .h : 17 files / .S 4 files
  - xv6-riscv-rust | .rs : 29 files / .toml : 1 file



# 移植

- Rustcによるobjファイル -> ld -> qemu
- CのABIに統一
- カーネル全体のファイルを全てrust\_kernel.rsのクレートに統合



# 移植

- .S(ブート・トラップ・コンテキストスイッチ)を.rsファイルに記述
  - Rustコンパイラの恩恵(型検査, 最適化...)
- 多くの関数がunsafe



# 内容

1. 動機
2. 移植の結果
3. 追加した機能



# 追加した機能

- SpinMutex<T>, SleepMutex<T>
  - SpinLockをUnsafeCell<Spinlock>の形で持つstruct
  - ロックの取得 = SpinGuard<T>構造体の発行
  - Rustのライフタイムを使ってMutexを自動解放する
    - Dropと同時にreleaseする
- InodeGuardもinodeに対して同様の構造



# 追加した機能

- TxnGuard
  - ログの開始・終了をlifetimeで管理
    - Begin\_opで開始し、end\_opで終了する
  - End\_opを実行せずに抜けることを防止
    - カウンタが0にならないことを防止



## 追加した機能

- KallocPage
  - Drop時にkfreeができる
  - AtomicU16によるkref\_cnt
  - COWにこれを利用した