

# **Mini-KVM**

**1,500줄로 구현한 완전한 하이퍼바이저**

**Linux KVM 기반 교육용 가상머신 모니터**

**가상화 기술은  
복잡해야 할까요?**

# 전통적 하이퍼바이저의 복잡성

## QEMU

- 코드 크기: 100,000+ LOC
- 초기화 시간: ~50ms
- 메모리 사용량: ~50MB
- 학습 곡선: 매우 가파름

## 문제점

- 교육 환경에서 이해하기 어려움
- 임베디드 시스템에 과도한 리소스
- 빠른 프로토타이핑에 부적합

# Mini-KVM의 접근

## 핵심 원칙

- 최소화된 복잡성 - 꼭 필요한 기능만
- 하드웨어 가상화 활용 - Linux KVM API
- 명확한 코드 - 읽기 쉬운 1,500줄

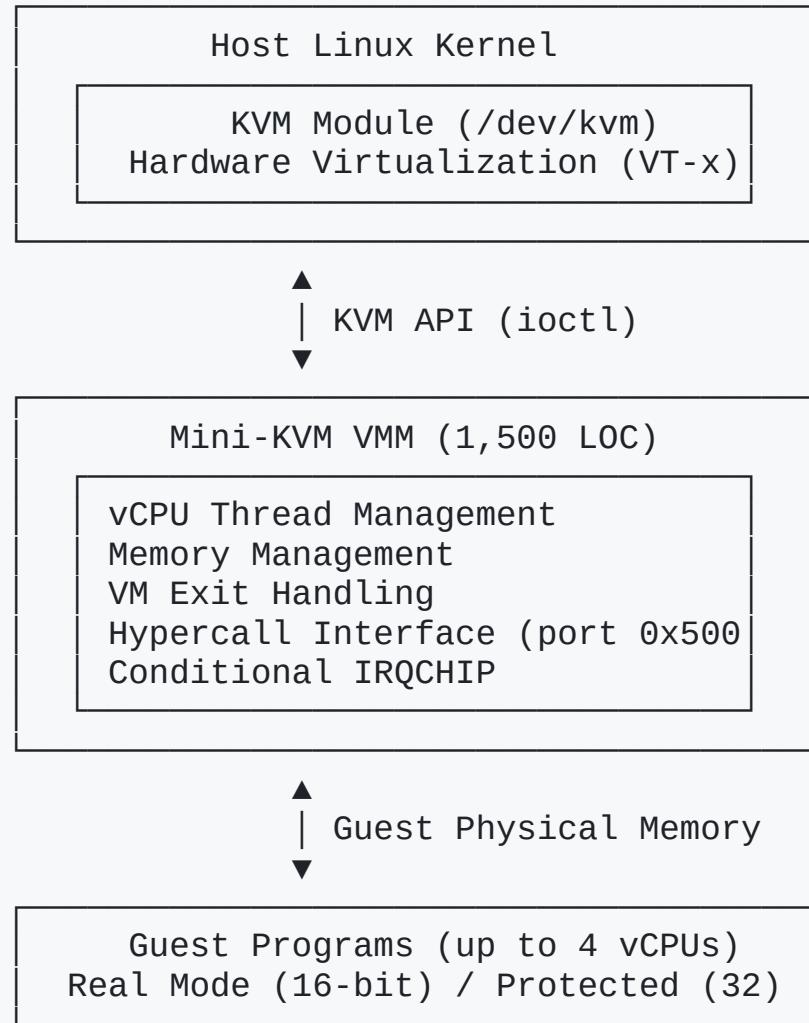
## 결과

10배 빠른 초기화

50-100배 빠른 실행

30배 적은 메모리

# 아키텍처 개요



# Real Mode 데모

## 멀티 vCPU 병렬 실행

- 곱셈표 계산 + 카운팅 동시 실행
- 출력이 문자 단위로 인터리빙 됨
- 진정한 병렬 처리의 증거
- 4 vCPU에서 90%+ 확장성

# Protected Mode - 1K OS 데모

## 1K OS 메뉴 화면

- 9개 사용자 프로그램 선택 가능
- GDT/IDT 설정으로 안정적 인터럽트 처리
- 4MB PSE 페이지 활성화

## 실행 중인 사용자 프로그램

- 곱셈표, 피보나치, 에코 프로그램
- 사용자 모드와 커널 모드 분리
- 타이머/키보드 인터럽트 정상 동작



# 응용 분야

## 교육

- 가상화 개념 학습용 플랫폼
- 읽기 쉬운 코드로 하이퍼바이저 구현 이해

## 임베디드 시스템

- 리소스 제약 환경에서 VM 실행
- 빠른 초기화로 실시간 대응

## 연구

- 하이퍼바이저 실험 플랫폼
- 새로운 가상화 기법 프로토타이핑

## 빠른 테스트

# 기술 스택

## 지원 모드

- **Real Mode** (16-bit): 단순 게스트, 멀티 vCPU 병렬성
- **Protected Mode** (32-bit): GDT/IDT, 4MB PSE 페이징, 인터럽트

## 1K OS 포팅

- RISC-V → x86 변환
- 9개 사용자 프로그램
- 커널/사용자 모드 분리

## 개발 환경

- Linux KVM API
- GCC, GNU AS/LD

## 프로젝트 통계

항목	수치
개발 기간	13주
총 커밋	310+
코드 라인	2,900 (VMM 1,500 + Guests 1,400)
Guest 프로그램	6개 (Real Mode) + 9개 (Protected Mode)
라이선스	MIT License
테스트 커버리지	단위 테스트 + 통합 테스트

# 가치 제안

## 단순하지만 완전함

복잡성을 제거했지만 모든 핵심 기능 제공

## 실전 성능

교육용이지만 프로덕션 수준의 성능

## 오픈소스

MIT 라이선스 - 누구나 학습, 수정, 배포 가능

## 명확한 코드

1,500줄로 가상화의 본질 이해

# **Mini-KVM**

**가상화의 본질을 1,500줄로 증명합니다**

**지금 바로 GitHub에서 확인하세요**

**[github.com/seolcu/mini-kvm](https://github.com/seolcu/mini-kvm)**

# Thank You

**Questions?**

GitHub: [github.com/seolcu/mini-kvm](https://github.com/seolcu/mini-kvm)

License: MIT