



아주대학교  
소프트웨어학과

# MiniKVM: 고속 x86 하드웨어 호환

# 소프트웨어학과 설규원 지도교수 – 김상훈 교수

1. Multiplication Table ( $2 \times 1 \sim 9 \times 9$ )
  2. Counter (0-9)
  3. Echo (interactive)
  4. Fibonacci Sequence

---

**가상화 기술은  
복잡해야 할까요?**



# 전통적 하이퍼바이저의 복잡성

## QEMU

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

## Problem

교육 환경에서 이해하기 어려움  
임베디드 시스템에 과도한 리소스



# 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



# 아키텍쳐 개요

## Guest VMs

Real Mode (16bit) 또는  
Protected Mode (32-bit + 1K OS)

## Kernel Space

### Linux KVM 모듈

- 하드웨어 가상화 (VT-x)
- EPT 메모리 관리

↔ KVM ioctl()

## User Space(Host)

### Mini-KVM VMM (1,500줄)

- VM/vCPU 관리
- I/O 에뮬레이션
- 하이퍼콜 인터페이스



# 핵심 설계 혁신

## 1. Conditional IRQCHIP

Real Mode: IRQCHIP 없음 → 깔끔한 종료

Protected Mode: IRQCHIP 활성화 → 인터럽트 지원

## 2. Per-vCPU Memory Isolation

각 vCPU는 독립적인 물리 메모리 영역

Real Mode: 256KB/vCPU

Protected Mode: 4MB



## 3. Hypervisor Interface

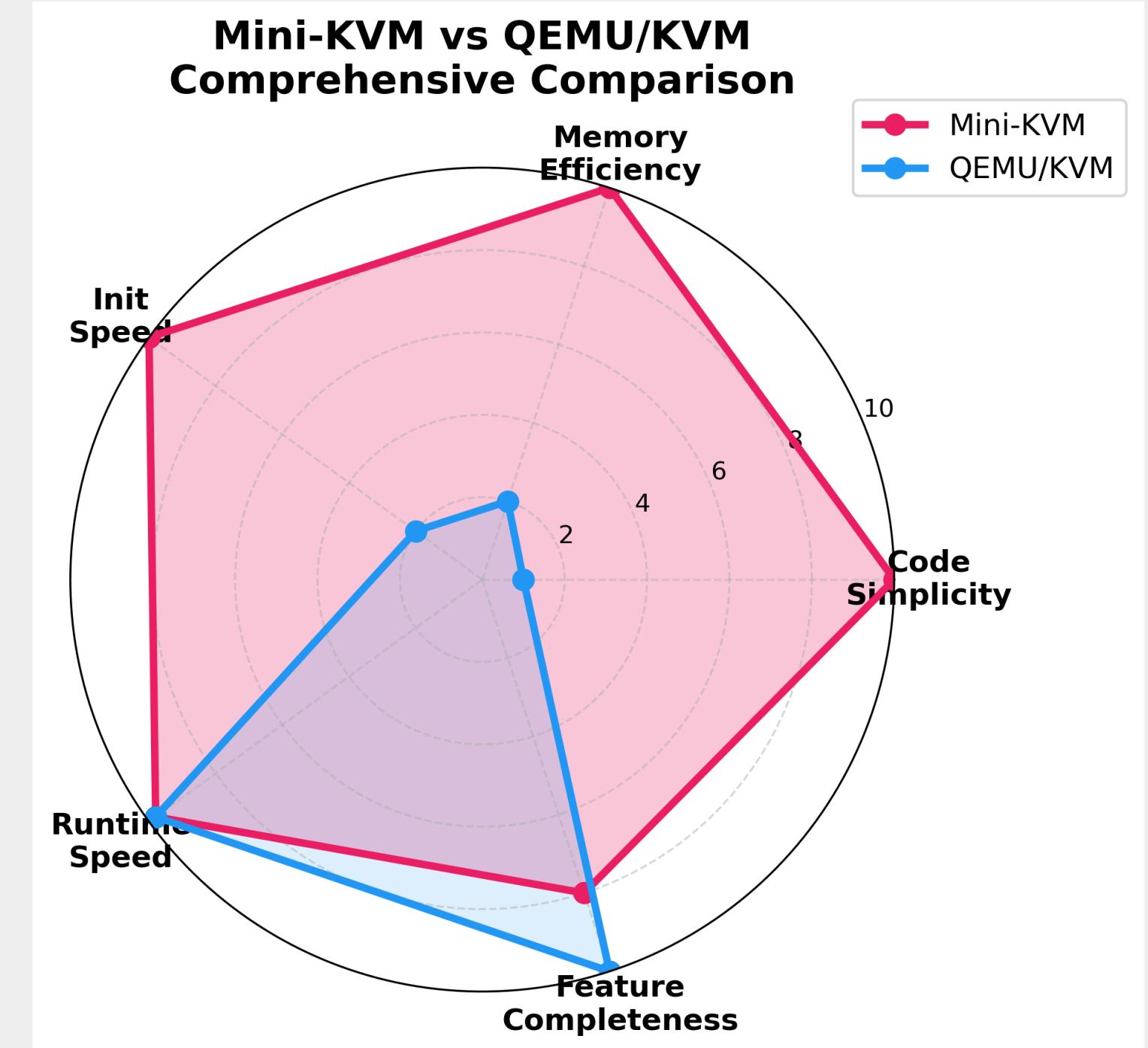
Port 0x500 을 통한 효율적인 통신

HC\_PUTCHAR, HC\_GETCHAR, HC\_EXIT 등

# 성능 비교

## 5가지 지표

1. 초기화 시간
2. 메모리 사용량
3. 코드 크기
4. 실행 속도
5. 기능 완성도



## 성능 수치 요약

	Mini-KVM	QEMU	비율
초기화 시간	<5ms	~50ms	10배 개선
실행 속도	Native-like	TCG	50~100배 향상
메모리	1.5MB	50MB	1/30로 감소
코드 크기	1,5000 LOC	100,000+ LOC	1/70로 축소

# Conditional IRQCHIP

```
int init_kvm(bool need_irqchip) {
    kvm_fd = open("/dev/kvm", O_RDWR);
    vm_fd = ioctl(kvm_fd, KVM_CREATE_VM, 0);

    // Protected Mode에서만 IRQCHIP 생성
    if (need_irqchip) {
        ioctl(vm_fd, KVM_CREATE_IRQCHIP, 0);
    }

    return vm_fd;
}
```

핵심: 모드에 따라 불필요한 오버헤드 제거



# 응용 분야

## 교육



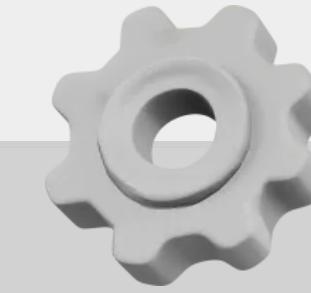
가상화 개념 학습용 플랫폼  
읽기 쉬운 코드로 하이퍼바이저 구현 이해

## 연구



하이퍼바이저 실험 플랫폼  
새로운 가상화 기법 프로토타이핑

## 임베디드 시스템



리소스 제약 환경에서 VM 실행  
빠른 초기화로 실시간 대응

## 빠른 테스트



VM 즉시 프로비저닝  
개발 환경 빠른 구축

# 기술 스택

## 지원 모드

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



# 가치 제안

## 단순하지만 완전함



복잡성을 제거했지만 모든 핵심 기능 제공

## 오픈 소스



MIT 라이선스 - 누구나 학습, 수정, 배포 가능

## 실전 성능



교육용이지만 프로덕션 수준의 성능

## 명확한 코드



1,500줄로 가상화의 본질 이해

---

# 감사합니다



Q&A: [seolcu0112@proton.me](mailto:seolcu0112@proton.me)

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

---