

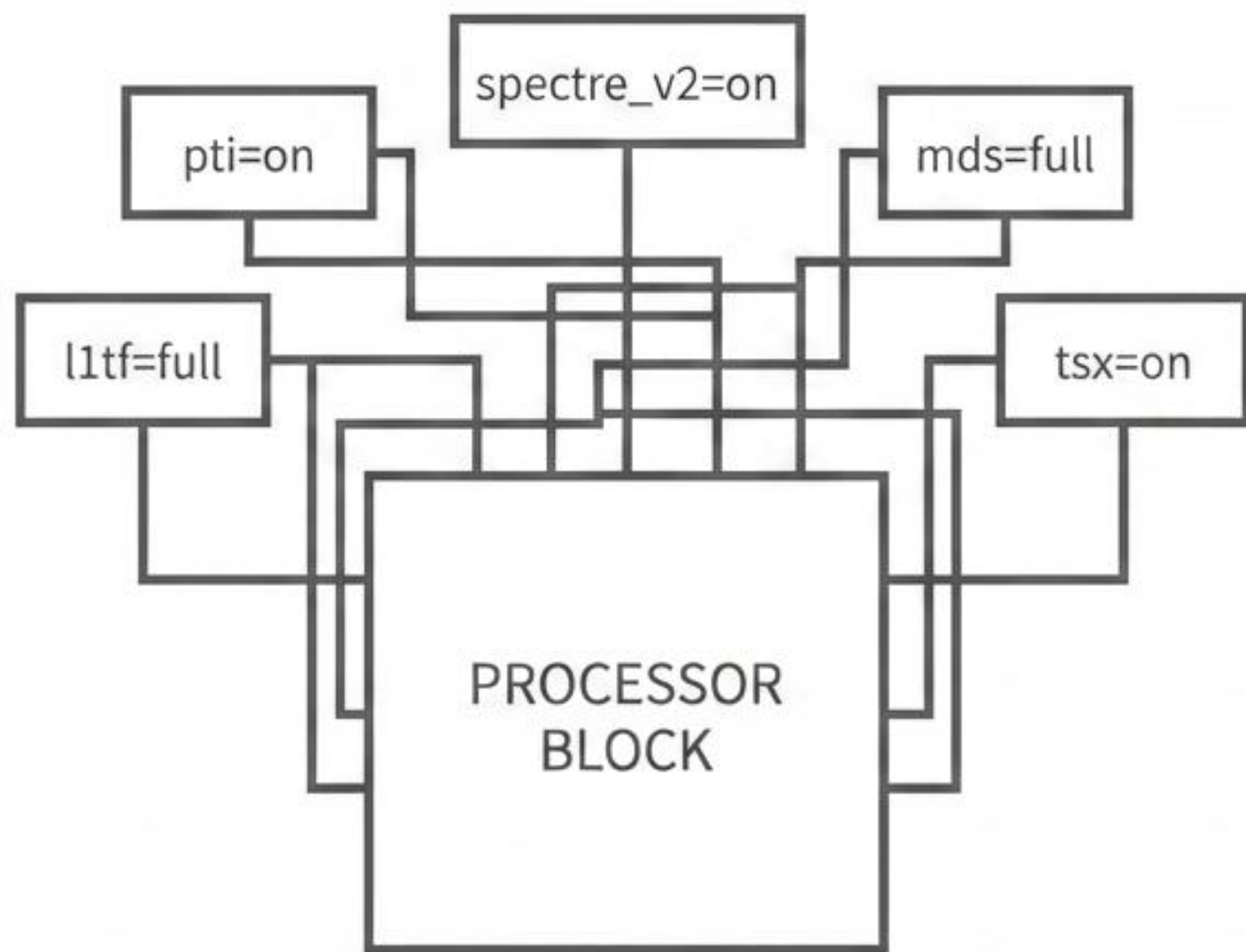
리눅스 커널 레포트 (v6.17 - v7.2)

박병철 @ SK하이닉스

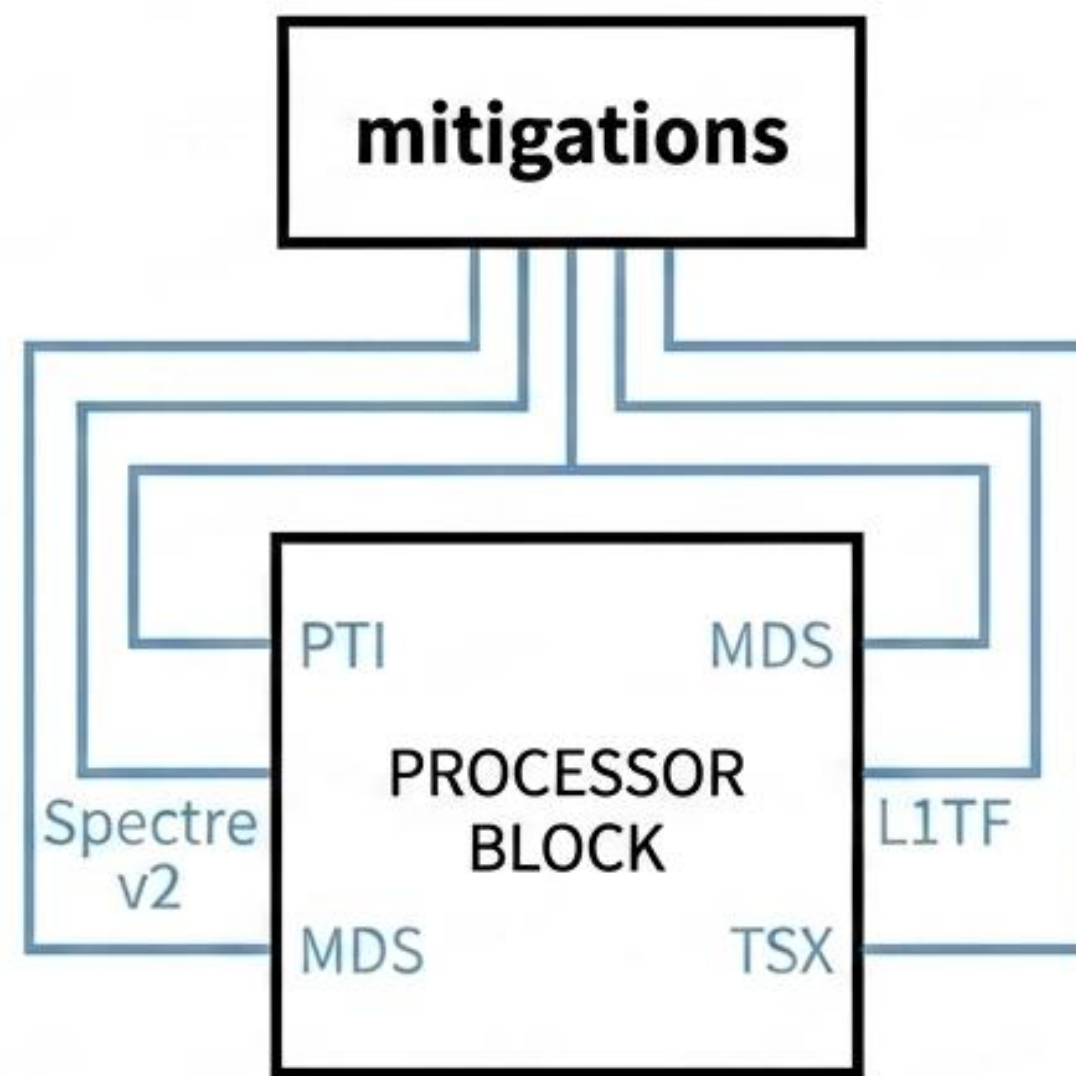
max.byungchul.park@sk.com

max.byungchul.park@gmail.com

[v6.17] CPU 보안 완화 옵션 제어 단순화

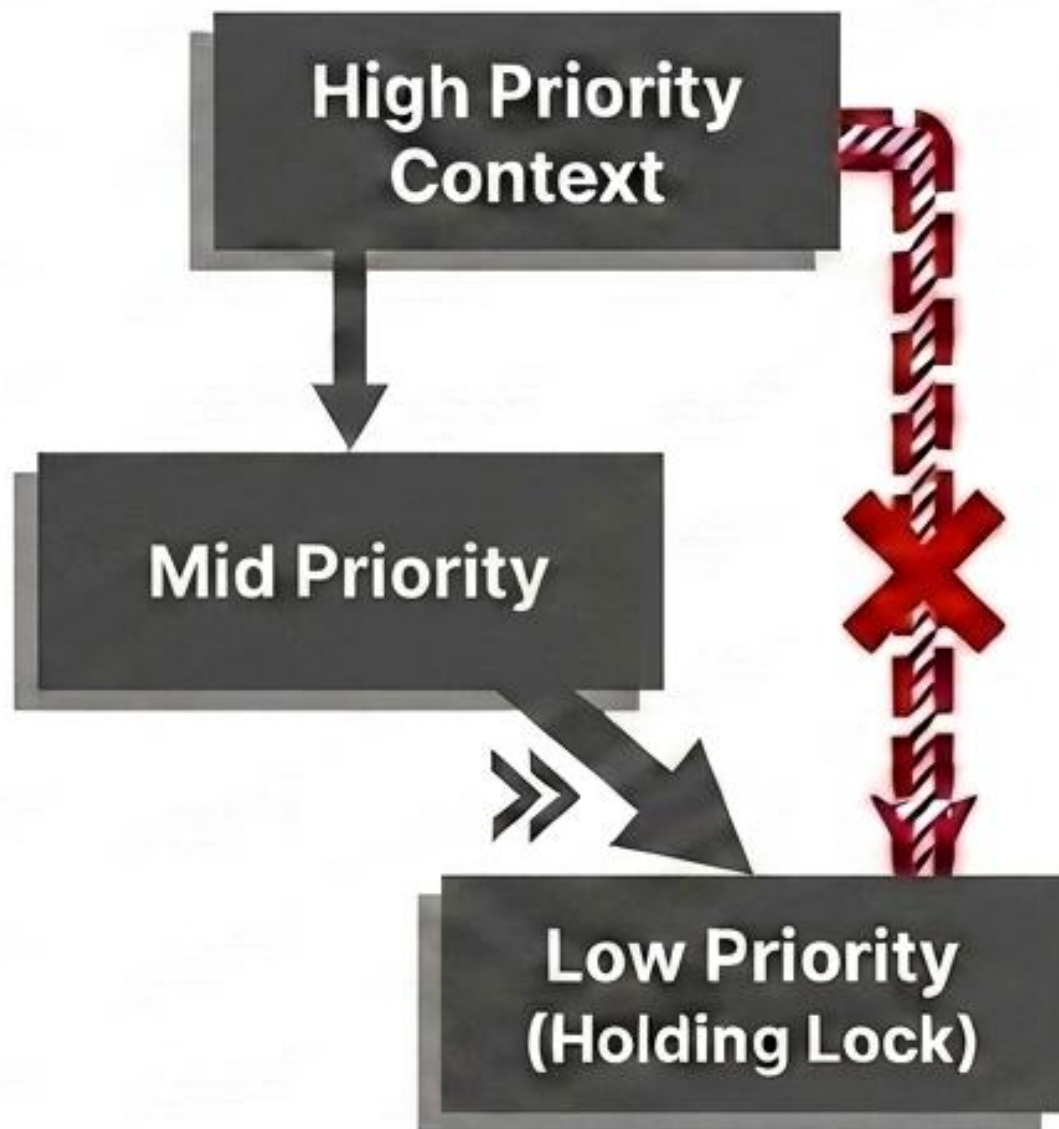


다수의 개별 커널 명령어를
부팅 파라미터에 각각 명시하여 제어

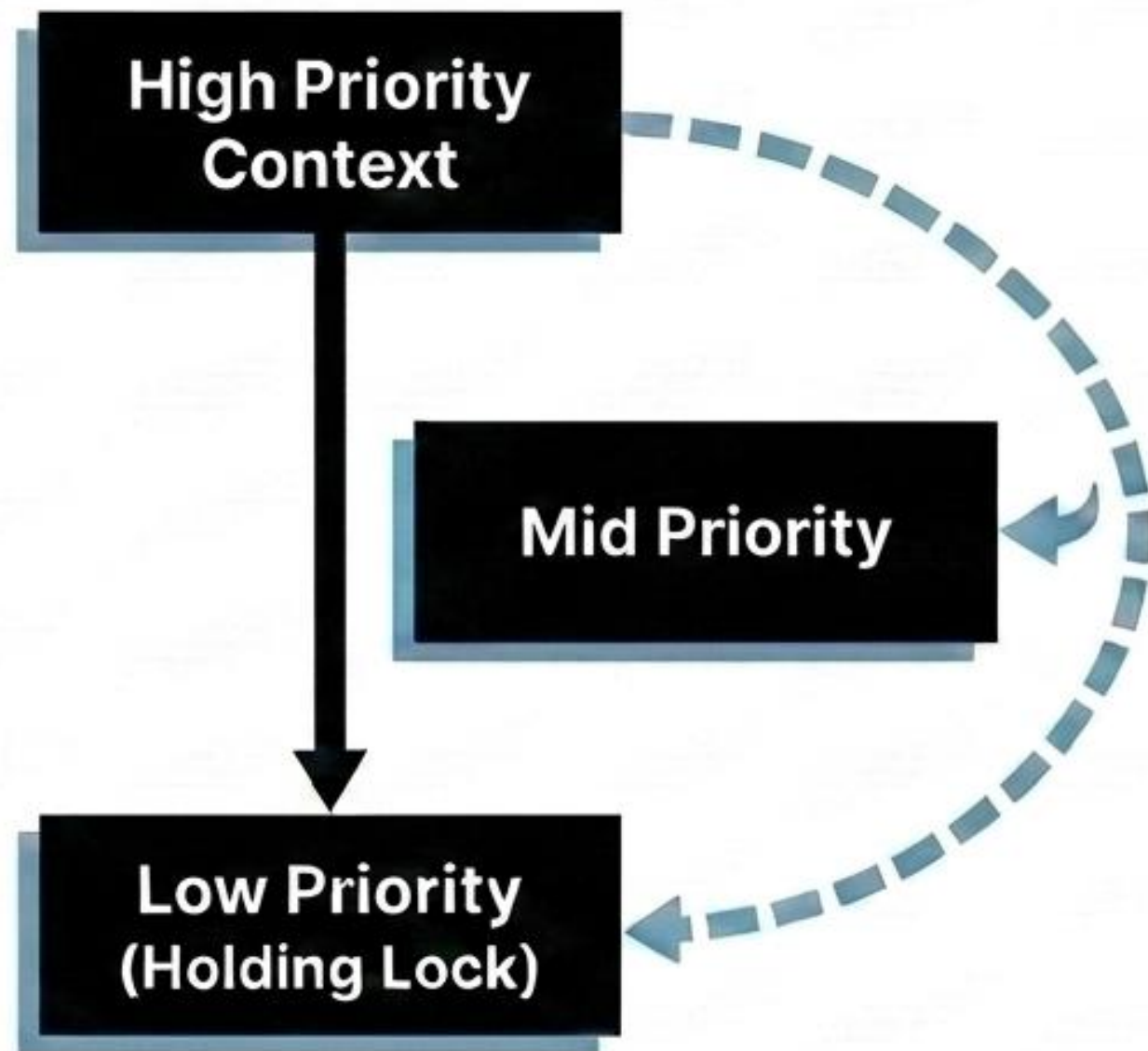


'mitigations' 단일 매개변수로
전체 보안 완화 옵션을 일괄 제어

[v6.17] 우선순위 역전 병목 해결을 위한 프로ksi 실행 초기 지원

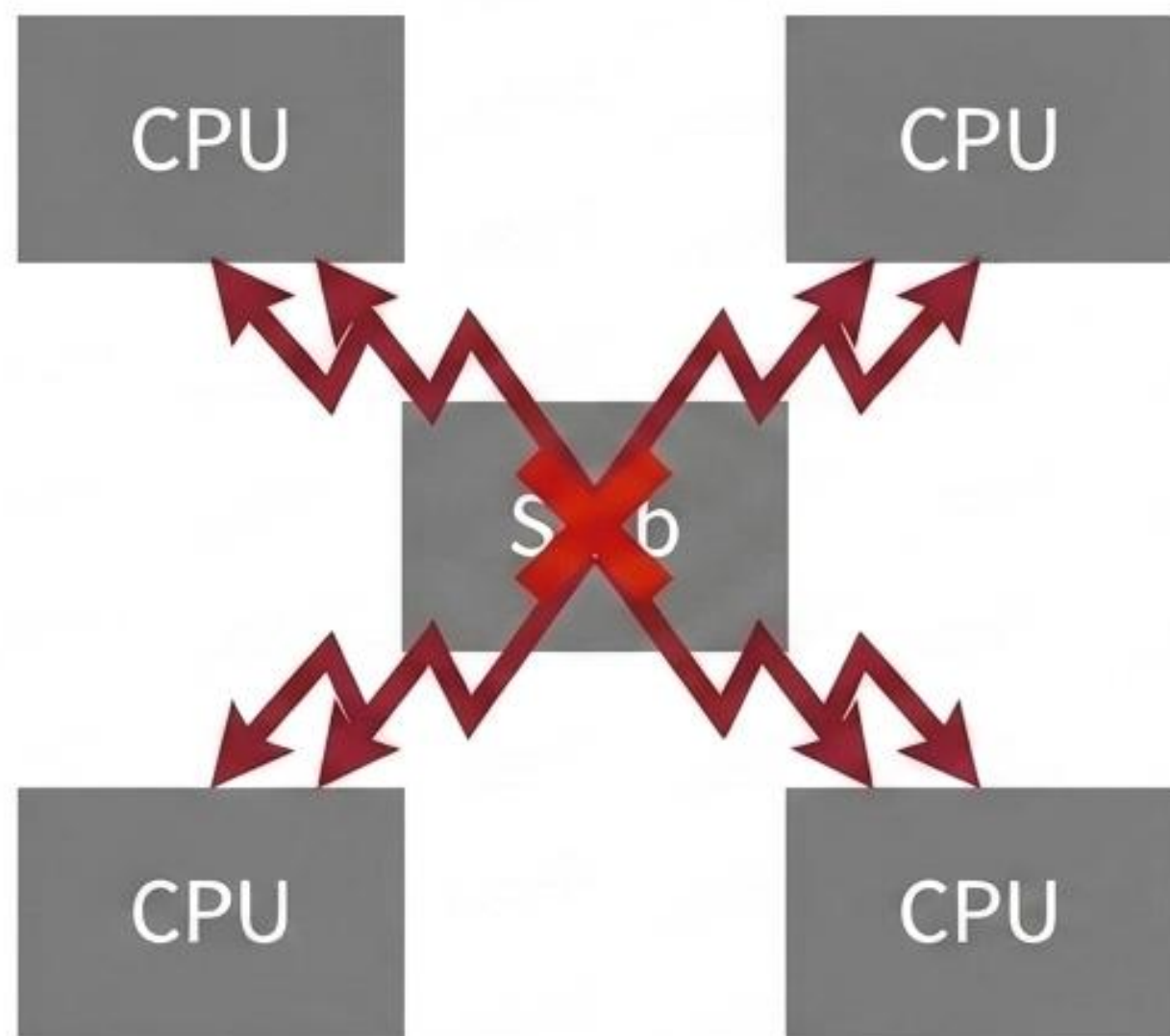


하위 실행이 락을 쥔 상태에서
중간 실행의 개입으로
고우선순위 실행이 무한 대기

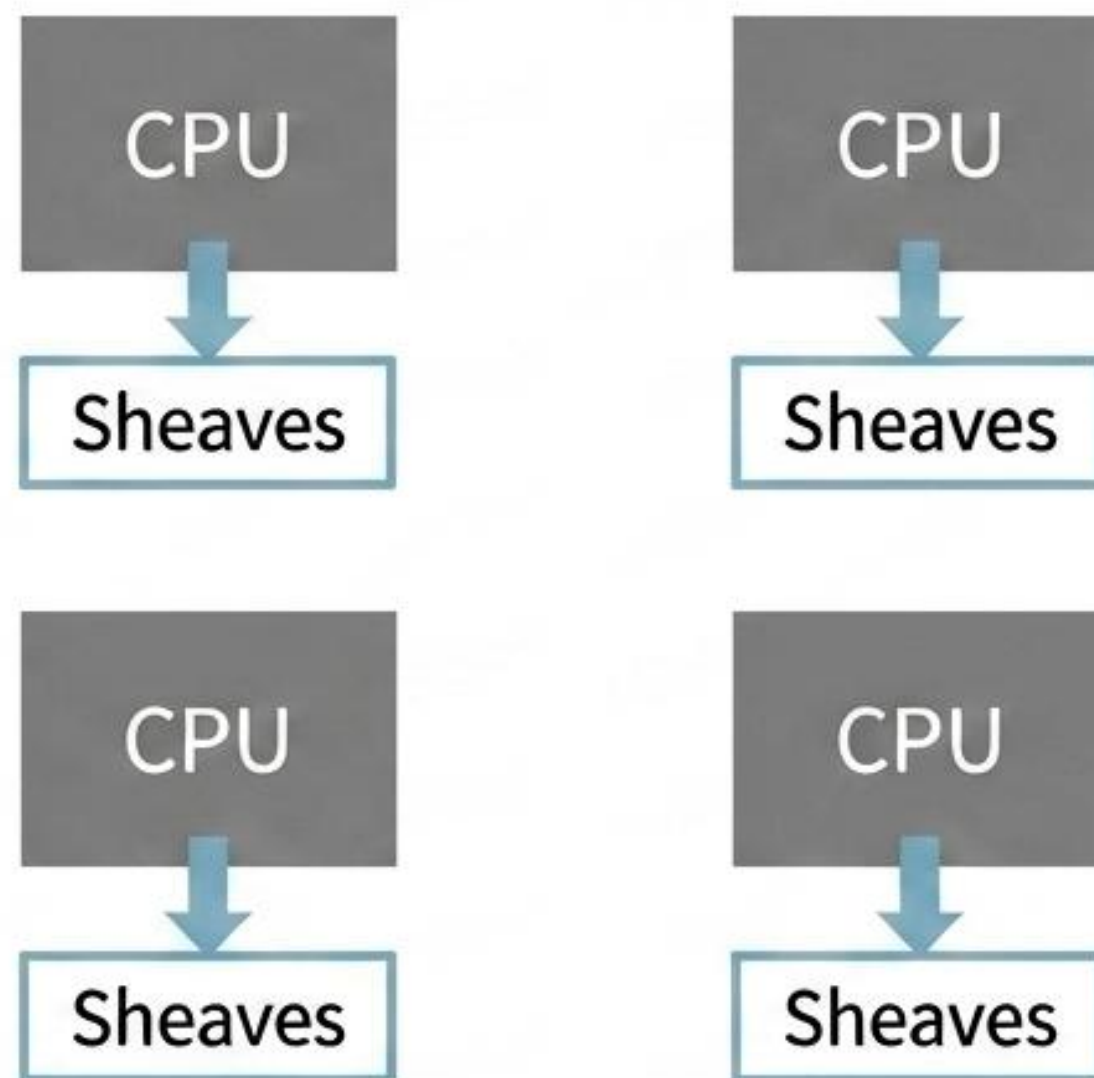


고우선순위 실행이 자신의 CPU
실행 기회를 락을 쥔 하위 실행에 고우선순위
실행 컨텍스트를 양도하여 신속한 락 해제

[v6.18] Slub Sheaves를 이용한 메모리 할당 최적화

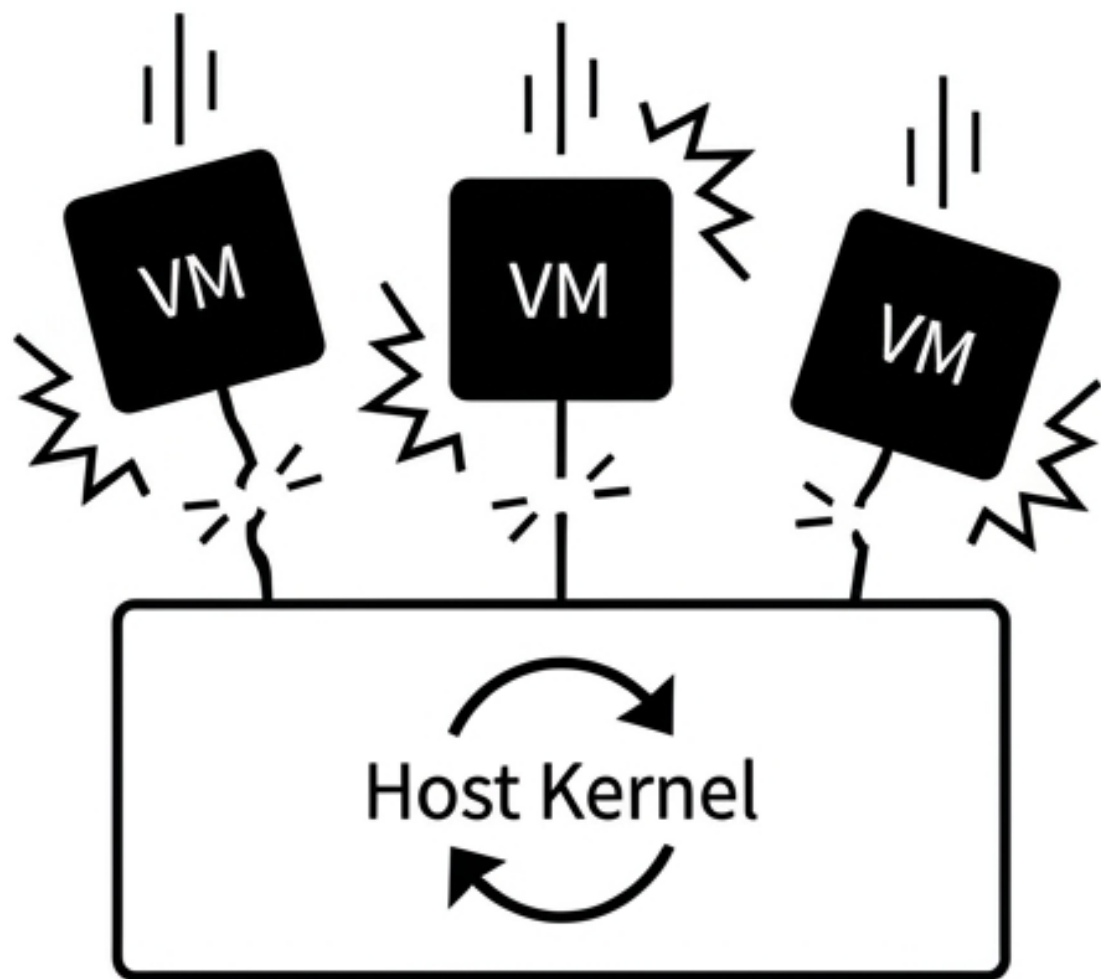


멀티코어 환경에서 여러 CPU가 공용 할당 구조(Slub)에 동시 접근하여 락 대기 지연 발생

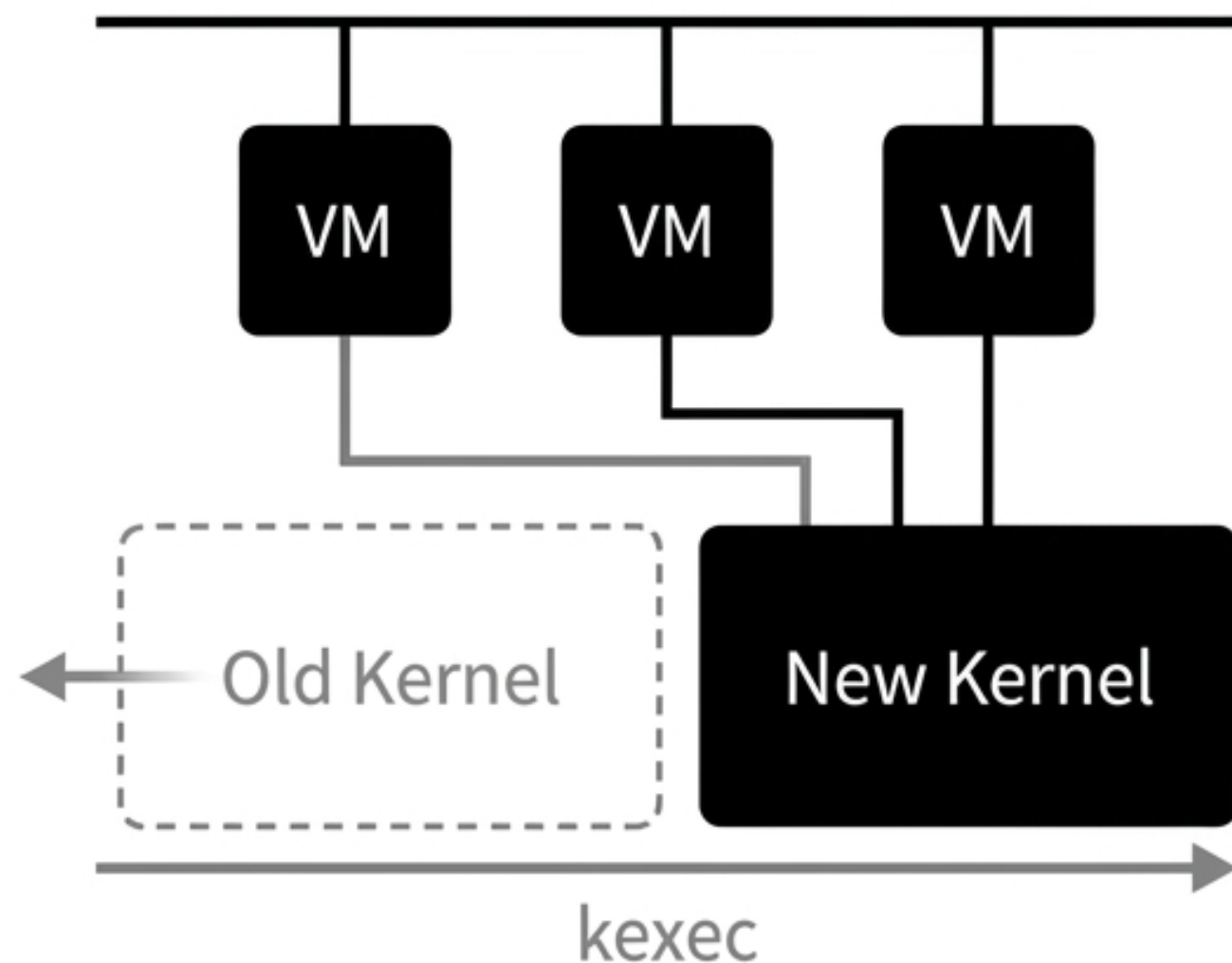


각 CPU 코어가 전용 독립 로컬 캐시인 '쉬브(Sheaves)'를 두어 락 없이 즉각 할당

[v6.19] Live Update Orchestrator 기반 물리 호스트 무중단 커널 교체

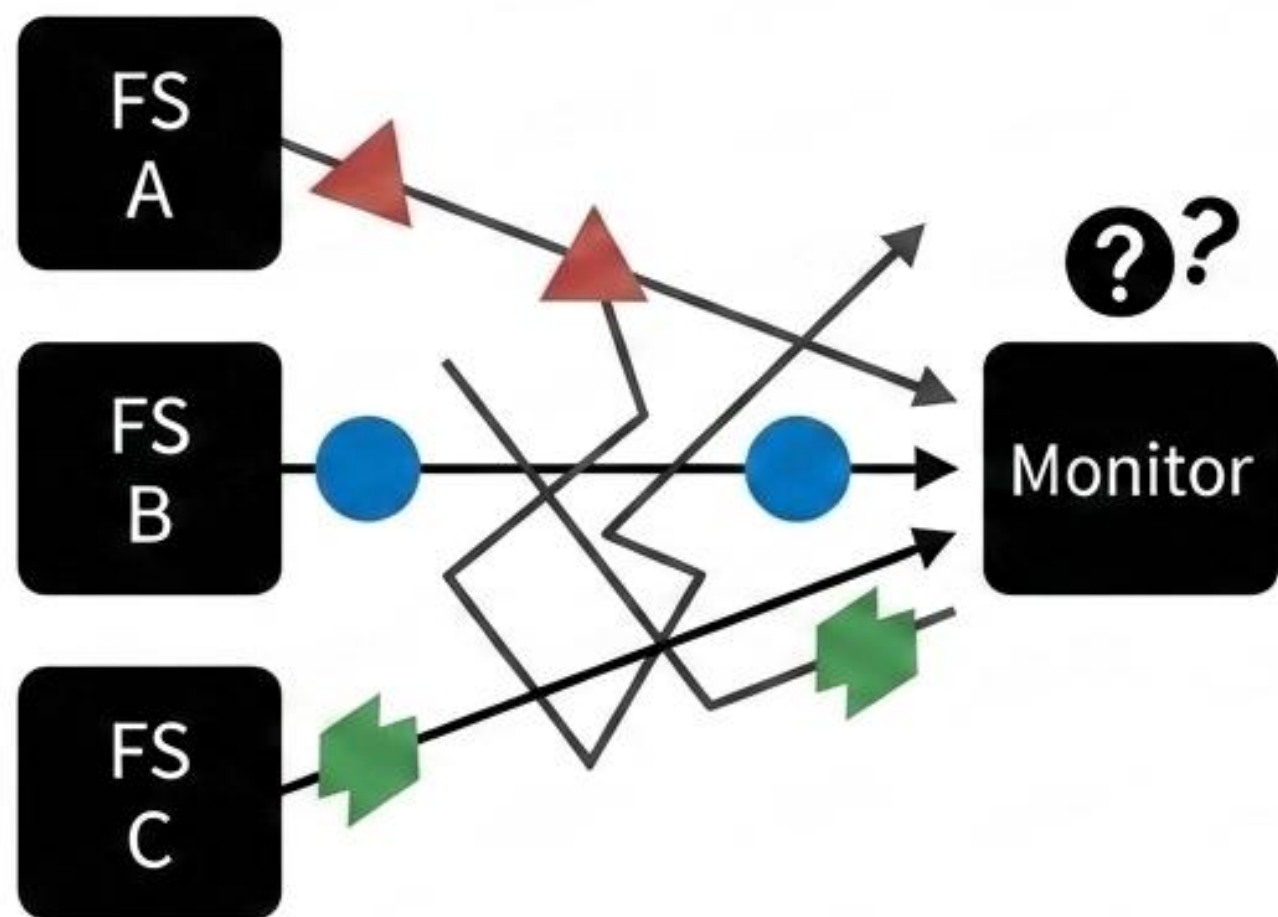


커널 업데이트를 위한 물리 서버 재부팅 시
구동 중인 가상 머신 및 서비스 전체 중단

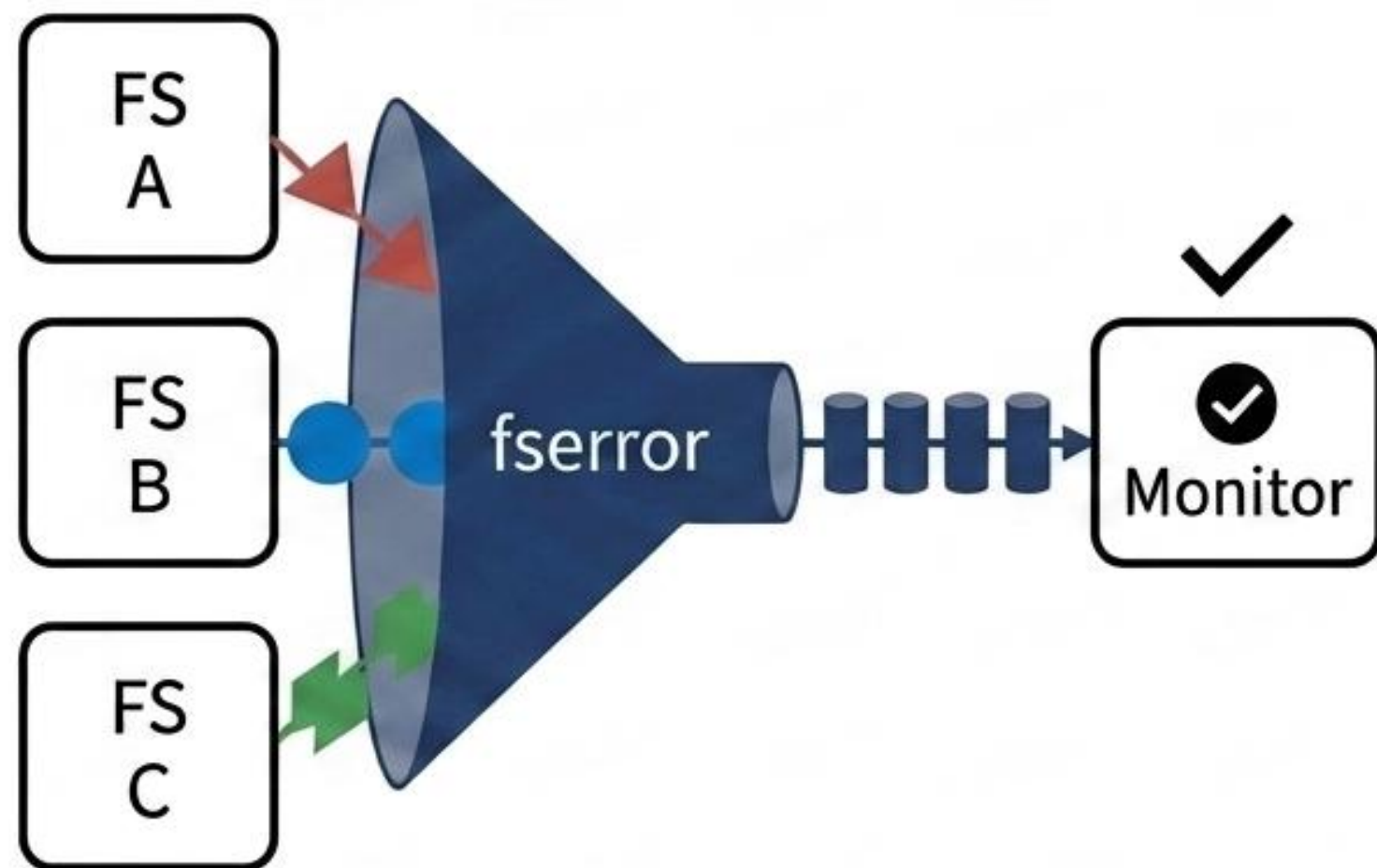


가상 머신의 실행 상태와 장치 연결을 유지한 채
kexec로 새 커널만 초고속 교체하여 무중단 운영

[v7.0] 파일 시스템 에러 표준 보고 인프라

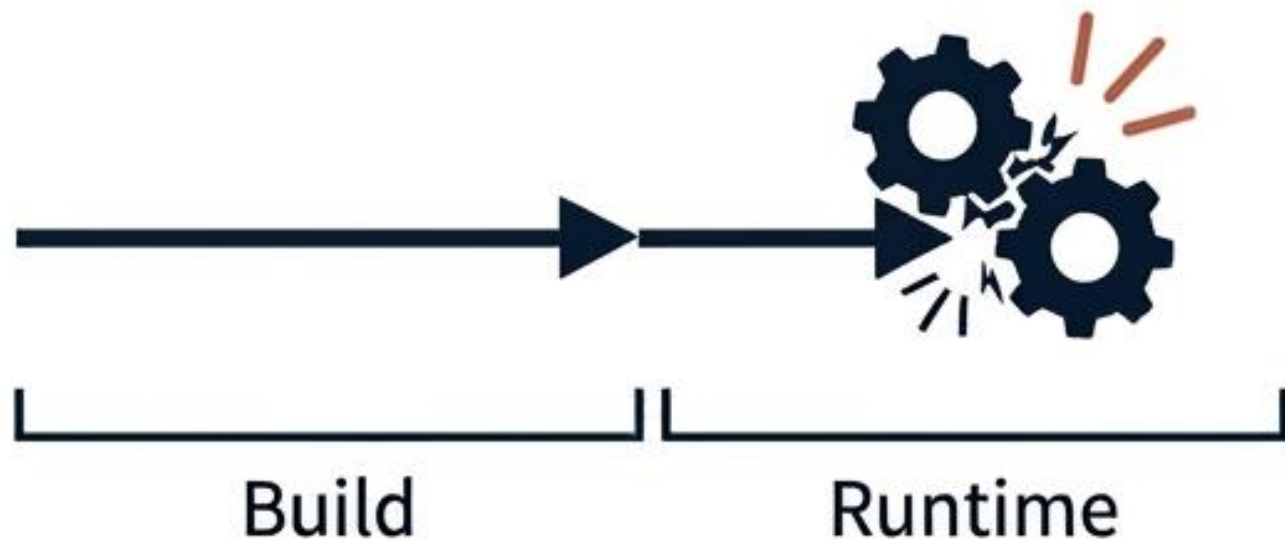


파일 시스템마다 오류 보고 포맷이 제각각이어서
일관된 중앙 감시 및 조기 감지 어려움

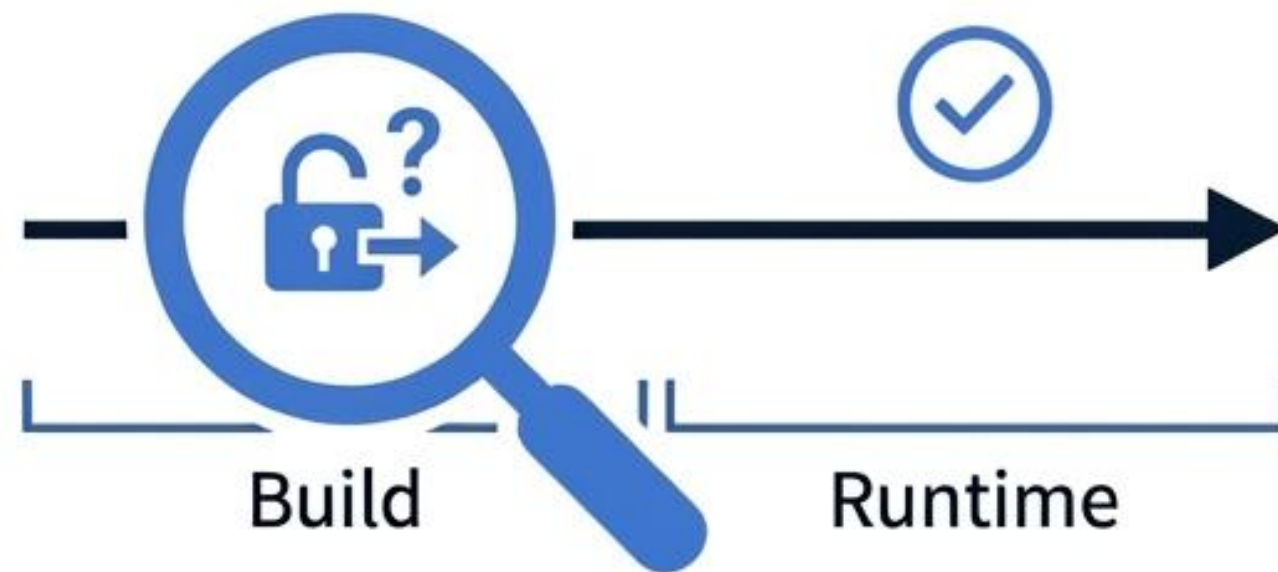


공통 에러 인프라와 통합 알림을 연계하여
표준화된 공통 양식 메시지로 원격 모니터링 가능

[v7.0] 컴파일 시점의 락 정적 분석 검증

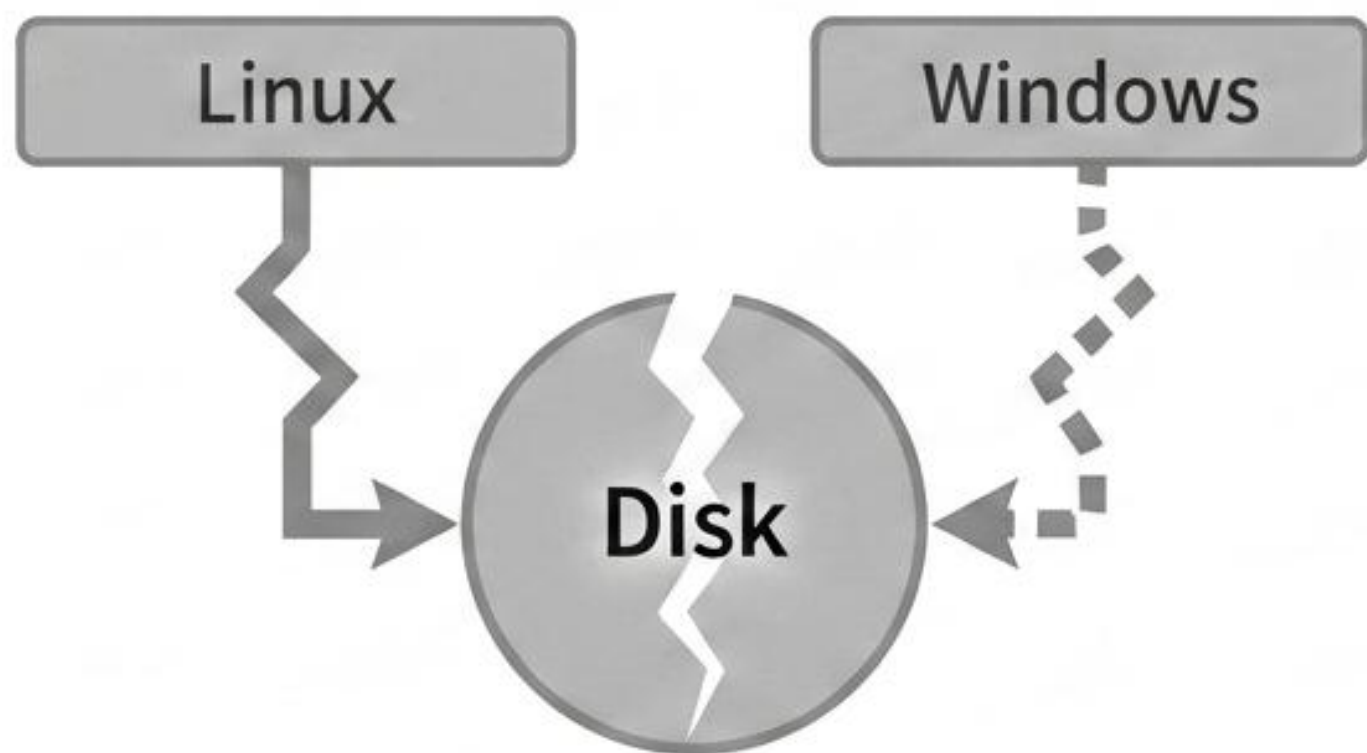


락 해제 누락이나 이중 락 획득 오류 코드가 런타임에 데드락을 유발하기 전까지 감지 불가

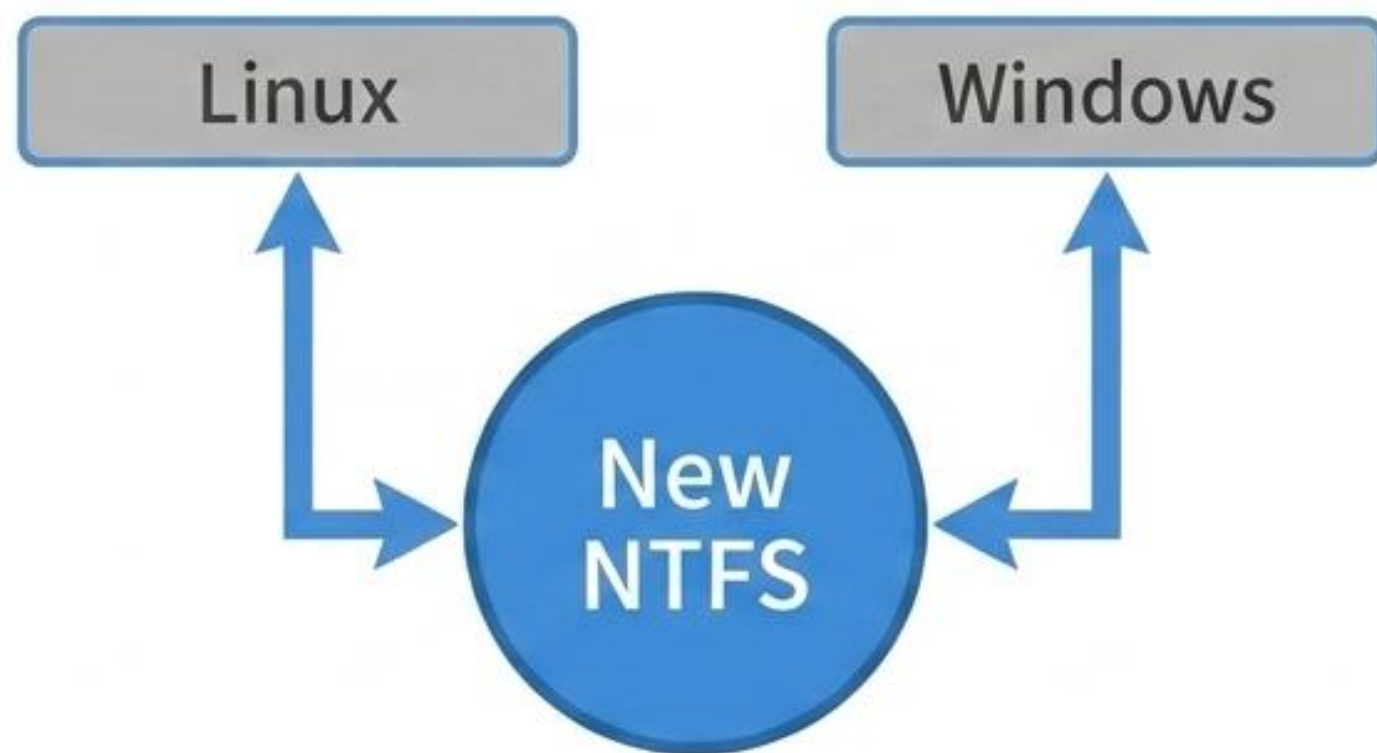


Clang 정적 분석과 연계하여 빌드 단계에서 동기화 소유 규칙을 정밀하게 사전 검증

[v7.1] NTFS 파일 시스템 전면 리메이크

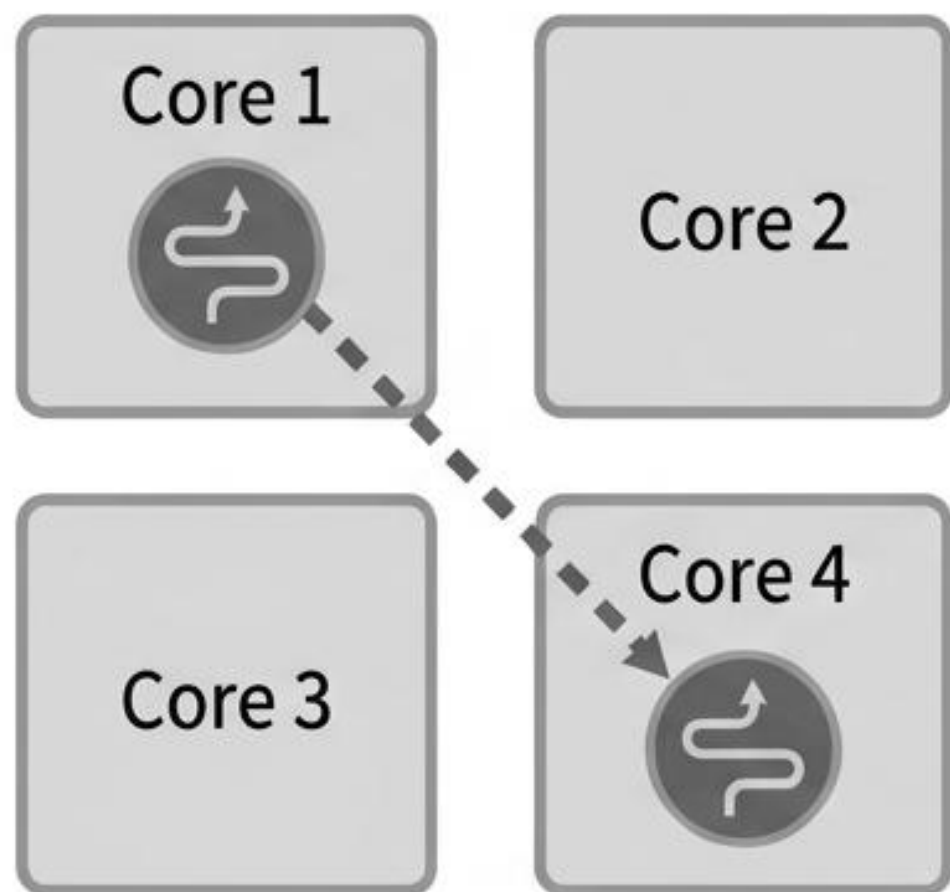


기존 레거시 NTFS 드라이버의
호환성 한계로 대용량 데이터 공유 시
정합성 유지 실패 및 시스템 불안정 유발

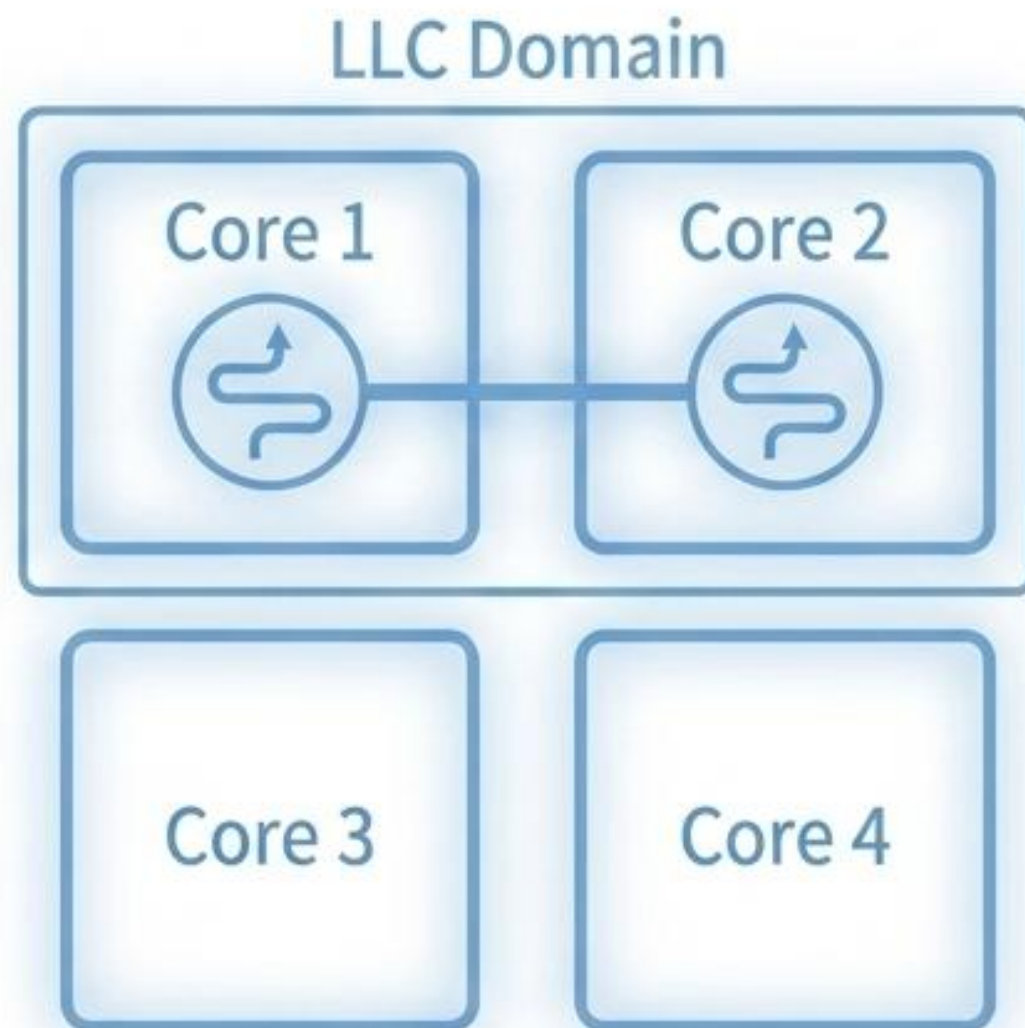


커널 수준에서 새롭게 재구성된
현대적 NTFS 구현을 도입하여 이종 OS 간
안정적인 양방향 읽기/쓰기 환경 제공

[v7.2] 공유 데이터 구조 중심의 캐시 인식형 CPU 스케줄링



긴밀하게 데이터를 공유하는 스레드들이 서로 먼 코어로 분산 배치되어 코어 간 캐시 동기화 지연 가중



스케줄러가 LLC 하드웨어 경계를 파악하여 **친화적 스레드들**을 밀접한 **도메인 내 코어로 묶어 모뎀 배치**

Question ?