



윈드리버 헬릭스 가상화 플랫폼

윈드리버 헬릭스 가상화 플랫폼은 다양한 산업군별로 안정성 인증을 마친 멀티코어 지원 플랫폼입니다. 성공적이고 안전 인증된 VxWorks® RTOS 기술의 검증된 실적을 바탕으로 멀티 OS와 다양한 중요도의 애플리케이션을 하나의 싱글보드에 통합하여 항공우주, 방위, 공업, 자동차 및 의료 시장에서 설계를 간소화, 보호 및 최신화합니다.

헬릭스 플랫폼은 트루 타입1 하이퍼바이저 (호스트 OS가 없이 운영하는 하이퍼바이저 형태)를 VxWorks나 Wind River Linux와 같은 사전 통합된 게스트 운영체제와 결합합니다. 항공, 자동차 및 공업 시장에서 사용되는 안전 인증 애플리케이션을 비롯한 중요한 인프라 개발 애플리케이션을 지원하도록 설계되었습니다. 또한 안전 인증 애플리케이션과 비인증 애플리케이션의 혼합이 필요한 시스템용으로도 디자인 할 수 있습니다.

헬릭스 플랫폼을 통해 고객은 현재의 요구사항에 대한 선택의 유연성뿐만 아니라 미래의 요구사항에 대한 적응 역량도 얻을 수 있습니다. 자세한 내용은 윈드리버 웹사이트(www.windriver.com/products/helix)를 참고하세요.

사용 사례

여러 시스템 간 애플리케이션 이식성 및 통합

디바이스 제조업체는 제품의 효율성, 생산성 및 품질 수준을 최적화해야 하기 때문에 기존의 워크로드와 애플리케이션을 새로운 하드웨어 플랫폼으로 이전해야 하는 경우가 많습니다. 또한 둘 이상의 개별 시스템을 하나의 가상화 하드웨어 플랫폼으로 통합해야 하는 경우도 많습니다. 동일한 컴퓨팅 플랫폼상에 있는 여러 시스템의 다양한 애플리케이션을 가상화로 통합하면 비용뿐만 아니라 공간/무게/전력 소비와 개발 업무량도 줄일 수 있습니다.

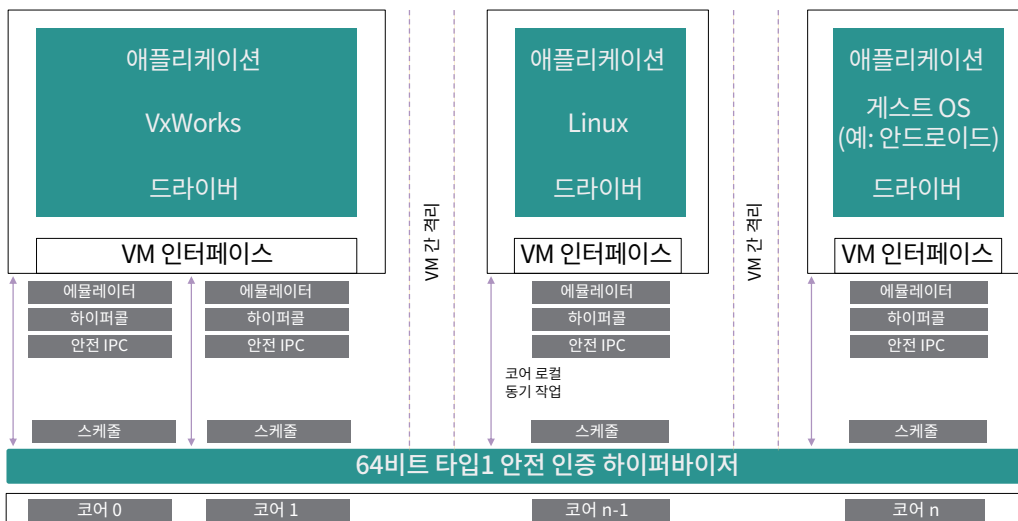


그림 1. 중요 시스템의 격리 및 보호

주요 결과:

- 레거시, 현재 및 미래 시스템을 연결
- 네이티브 성능에 근사한 수준으로 기존 애플리케이션 실행
- 공간, 무게, 전력 및 비용 절감
- 애플리케이션 소프트웨어 통합 및 재사용
- 새로운 하드웨어 플랫폼으로의 애플리케이션 이식 지원

다양한 중요도와 안전 수준의 시스템

안전 인증 시스템 제조업체는 또한 일부 시스템을 통합해야 합니다. 이를 위해 개발자는 다양한 중요도와 안전 인증 수준을 가진 게스트 운영체제 및 애플리케이션을 지원할 수 있는 안전 인증 가상화 솔루션이 필요합니다.

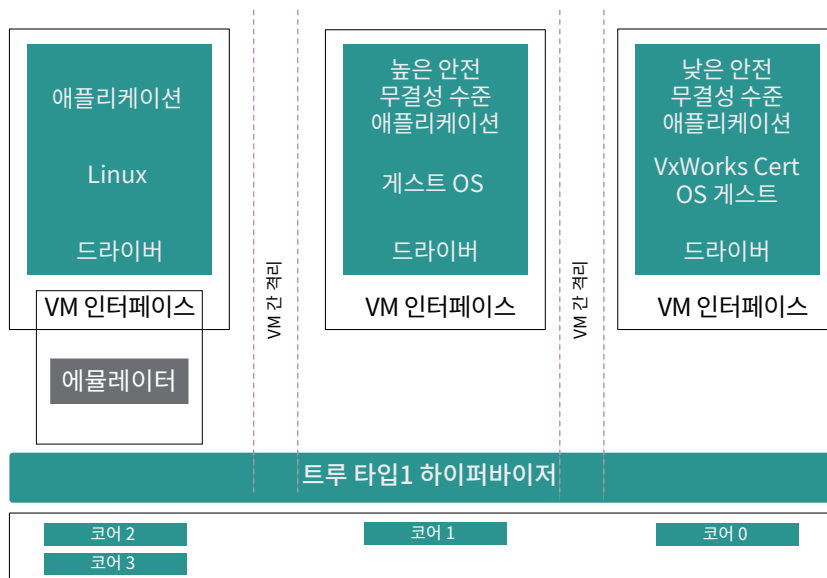


그림 2. 다양한 안전 수준을 가진 안전 인증 시스템

주요 결과:

- 다양한 안전 중요도 수준을 가진 애플리케이션을 통합
- 안전 인증 및 비인증 애플리케이션을 단일 하드웨어 플랫폼에 결합
- 공간, 무게, 전력 및 비용 절감

유연한 역할 기반 개발

대규모 프로젝트에서 시스템 구성을 모두 혼자서 담당하는 경우는 없습니다. 플랫폼과 게스트 운영체제는 책임자가 다르고 각자 담당하는 구성에 대해 책임집니다. 초대형 프로젝트의 경우, 이러한 책임자들이 서로 경쟁 관계에 있는 기업일 수도 있습니다.

IBLL(Independent build, link and load) 시스템을 지원하는 XML 구성 포맷을 사용하면 역할 기반 개발 원칙(예: DO-297 항공 표준에 정의된 원칙)에 따라 플랫폼 공급자, 서트 플랫폼 공급자, 애플리케이션 개발업체 및 시스템 통합업체 사이에 진정한 분리 및 독립적인 개발이 가능합니다. 이를 통해 각자의 목표에 따라 최대한 유연하게 개발 역할을 분산할 수 있으며 대규모 인증 소프트웨어 시스템 개발을 가속화할 수 있습니다.

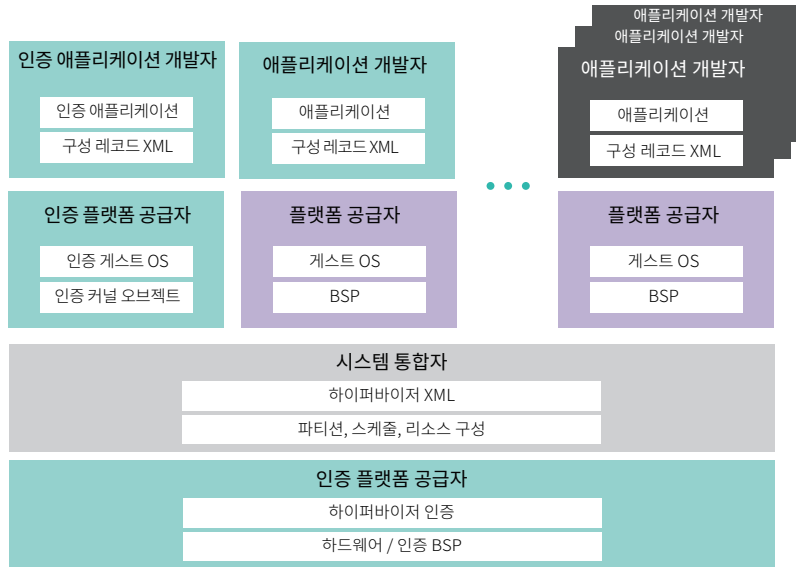


그림 3. 독립적인 역할 기반 개발

주요 결과:

- 애플리케이션 공급자가 독립적으로 소프트웨어 애플리케이션을 비동기식으로 개발, 테스트 및 제공
- 독립적인 공급자 빌드 프로세스로 여러 개발팀에 미치는 코드 변경 영향 감소
- VxWorks 또는 Linux 컨테이너 및 표준 클라우드 네이티브 툴을 통해 여러 복잡한 시스템에 걸쳐 애플리케이션 독립성 제공

주요 기능 및 이점

- 64비트 타입1 안전 인증 하이퍼바이저
 - 서비스 OS 또는 특별 권한 VM 불필요
 - 결정론적 게스트 OS에 대한 실시간 응답 속도
 - 다수의 코어 및 가상 머신(VM)에 대해서도 낮은 오버헤드 및 네이티브에 근사한 성능 지원
 - 코어 간 진정한 운영 독립성
 - VM에 로컬인 디바이스 에뮬레이터
 - VM 간 유연한통신
 - 스케줄러, 로컬 타이머, 로컬 에뮬레이터 분리
 - 디바이스, 통신 채널 및 메모리의 할당 및 관리를 지원하는 결정론적 구성
 - 직접 인터럽트 지원
 - 작은 설치 공간
 - 견고한 파티셔닝(모든 VM에 걸쳐 다양한 안전 수준 지원)
- 틱리스(tickless) 커널:
 - 타이머 인터럽트가 주기적으로 발생하지 않고 필요할 때만 제공됨
 - 실시간, 결정론적 성능 및 지연 시간
 - 하이퍼바이저의 낮은 오버헤드
 - 대기 중이 아니면 틱이 없으므로 시스템 틱에 의해 시간의 범위가 제한되지 않음
 - VM이 하나만 있는 경우에는 틱이 없음
 - 네이티브 성능에 근사

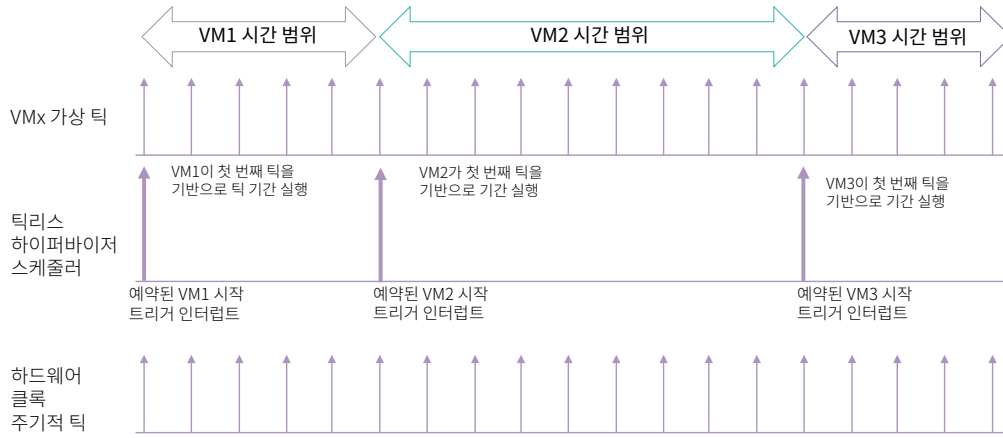


그림 4. 필요에 따라 제공되는 인터럽트를 통한 티리스 무제한 시간 범위

• 시간 파티션 스케줄러:

- 엄격한 스케줄에 따라 파티션 또는 VM 예약
- 코어당 여러 VM을 지원하는 구성가능한 스케줄 지원
- VM이 지정된 코어에서 지정된 시간 범위 내에서만 실행되도록 지원
- 결정론적 방식 (예측된 시간내에 항상 동작하는 방식)으로 각 게스트에게 특정한 CPU 할당 시간 보장
- 단일 시스템에 대해 여러 스케줄을 정의하고 런타임에 동적으로 전환할 수 있도록 지원

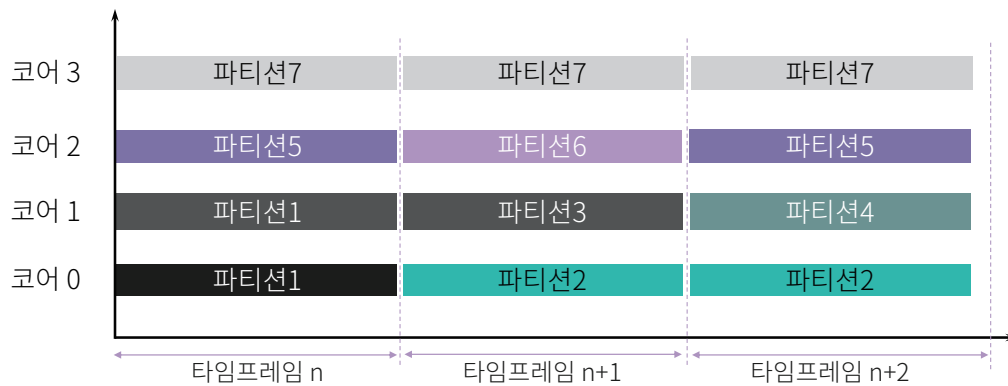


그림 5. 서로 다른 코어에서 동시에 실행되는 서로 다른 파티션

• 디바이스에 대한 직접 인터럽트 및 액세스

- 직접 인터럽트는 물리적 디바이스에서 게스트 운영체제로 직접 이동하므로 네이티브에 근사한 응답 속도를 제공합니다.
- 하이퍼바이저는 디바이스와 주고받는 데이터 경로에 관여하지 않습니다.
- 게스트가 직접 인터럽트 모드로 실행 중인 경우 VM 종료는 필요하지 않으므로 인터럽트 지연 시간이 개선됩니다.
- 메모리, PCI 속성 및 인터럽트는 게스트 운영체제에 직접 매핑될 수 있습니다.

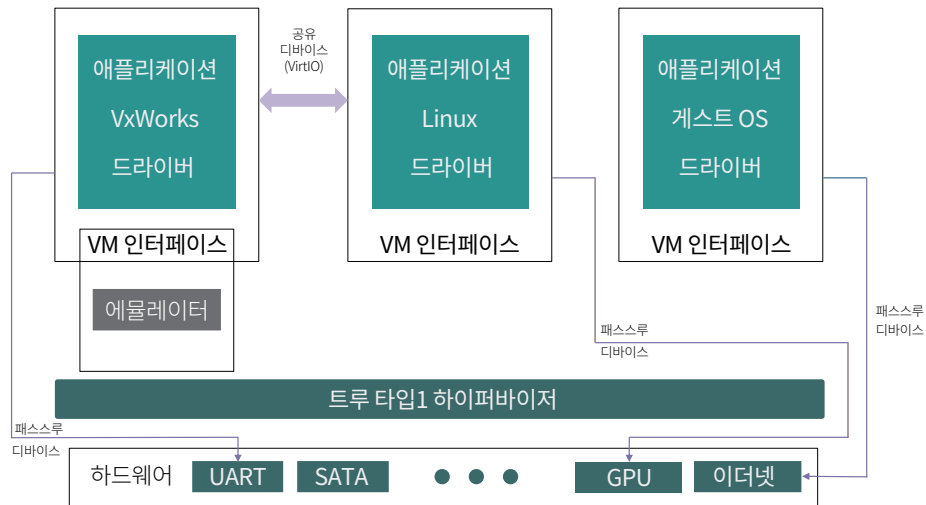


그림 6. 디바이스에 대한 직접 인터럽트 및 액세스

• 확장성:

- 코어 및 게스트 수 무제한
- 코어 수가 증가해도 성능이 저하되지 않음
- 다양한 중요도의 게스트 운영체제 지원
- 수정하지 않은 게스트 지원
- 멀티 코어 파티션/VM 지원
- VxWorks, VxWorks Cert Edition 및 Wind River Linux에서 즉시 사용 가능한 게스트 OS 지원
- 안전 멀티 OS 통신(안전 IPC)

• 품질 및 성능 중심 틀:

- VxWorks RTOS를 사용하여 여러 VM의 시스템 모드 디버깅을 지원하는 플랫폼 디버거로 전체 VM에 대한 동기화된 뷰 지원
- 윈드리버 시스템 뷰어를 통해 VxWorks 및 하이퍼바이저를 사용하는 전체 VM에서 발생하는 OS 이벤트에 대한 뷰를 제공하여 개발자가 복잡한 목표 활동을 시각화하고 문제를 해결하도록 지원
- 부트 프로파일러를 통해 초기화 중에 하이퍼바이저와 VM 게스트 OS에서 소요되는 시간에 대한 일관된 뷰를 제공하여 집중적인 부팅 최적화를 지원
- 이벤트 인젝션을 지원하고 사용자 정의 및 구성 가능한 핸들러와 해당 이벤트의 로깅을 제공하는 프레임워크인 구성 가능한 상태 모니터링(ARINC 653 호환)

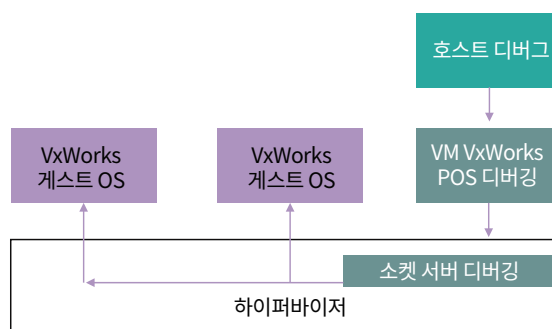


그림 7. 여러 VM의 시스템 모드 디버깅을 지원하는 플랫폼 디버거

주요 특징

- **게스트 OS 지원**
 - VxWorks (헬릭스 플랫폼에 포함)
 - VxWorks Cert Edition (헬릭스 플랫폼에 안전 인증 게스트 포함)
 - Wind River Linux (헬릭스 플랫폼에 Linux 바이너리 포함)
 - 베어메탈 게스트
 - 타사 게스트
 - 헬릭스 플랫폼으로 가상화되는 게스트 운영체제는 컨테이너를 지원함(VxWorks 및 Wind River Linux 포함)
- **소스 코드 100% 제공**
 - 긴 제품 수명 주기
- **프로세서 지원**
 - 64비트 프로세서 지원, 32비트 및 64비트 게스트 지원
 - ARMv8 및 Intel x86-64 프로세서
 - Intel, Xilinx, NXP, Renesas, Aptiv와 같은 제조사의 SoC 지원 포함
- **디바이스 에뮬레이션**
 - 디바이스 트리
 - VirtIO
 - 공유 메모리
 - 공유 메모리를 통한 가상 네트워크 인터페이스(VNIC)
 - 사용자 지정
- **안전 인증**
 - 다양한 중요도 지원
 - DO-178C DAL A, IEC 61508 SIL 3 및 ISO 26262 ASIL-D 인증
 - 330개 고객사를 대상으로 120기 이상의 민간 및 군용 항공기에 750개 이상의 안전 프로그램 배포
- **보안**
 - DO-356A 지원
 - 게스트 격리 및 분리
 - 리소스 액세스 제어
 - 보안 정책 구성을 통한 침입 방지
 - 게스트 OS 보안 기능 지원
 - MITRE CVE 모니터링 및 해결
- **윈드리버 워크벤치 개발 환경**
 - 프로젝트 빌드 및 구성
 - 시스템 통합 지원
 - 빌드, 구성 및 패키징 독립성(IBLL)
 - 독립적인 페이로드를 위한 여러 프로젝트, 지원되는 하이퍼바이저 프로젝트 유형:
 - 워크벤치 디버깅
 - 헬릭스 플랫폼 시스템 디버그(인증 무관)
 - 헬릭스 플랫폼 시스템 뷰어
 - 하이퍼바이저(비인증)
 - VxWorks 게스트 OS(비인증)
 - VxWorks 다이렉트 디버그(비인증)
 - VxWorks 다이렉트 시스템 뷰어(비인증)
- **윈드리버 스튜디오 내 빌드 지원**

안전 인증

윈드리버는 지난 40년 이상 100기 이상의 민간 및 군용 항공기에 600개 이상의 프로그램을 배포하고 자동차, 항공, 공업 분야에서 360개 이상의 고객사가 안전 인증 제품에 윈드리버 안전 플랫폼을 사용하는 등 광범위한 안전 인증 포트폴리오를 구축해 왔습니다. 윈드리버의 운영체제 플랫폼은 ISO 26262 자동차 기능 안전 표준, DO-178C 항공 전자 안전 표준, IEC 61508 산업 기능 안전 표준 인증을 모두 획득했습니다.

아키텍처와 인증을 통한 총소유비용 절감:

- DO-178C(DAL A) 항공 시스템 소프트웨어 표준, IEC 61508(SIL 3) 산업 기능 안전 표준 및 ISO 26262 자동차 안전 표준의 엄격한 요구사항에 따라 안전이 중요한 애플리케이션의 간소화된 인증을 지원하도록 설계되었습니다.

- 모듈형의 개방형 아키텍처는 강력한 파티셔닝 기능을 채택하여 파티션/VM 중 하나가 변경될 때 전체 플랫폼의 재테스트 및 재인증 필요를 크게 줄여줍니다.
- 전체 시스템에 대한 일반적인 재테스트 요구사항 없이 새로운 파티션을 모든 디바이스에 손쉽게 추가할 수 있어 총소유비용이 감소합니다.

윈드리버 전문가 서비스

CMMI Level 3 인증을 받은 윈드리버 전문가 서비스 조직은 오랜 시스템 설계 및 개발 경험을 바탕으로 고객의 설계 및 프로그램 팀과 협업합니다. 전문가 서비스는 시스템 요구사항을 해석하고 플랫폼 옵션을 설계하여 비즈니스, 기술 및 프로그램 목표를 위한 권장사항을 제공합니다. 윈드리버 전문가 서비스 팀은 또한 인증 BSP(board support package), 미들웨어 및 애플리케이션 소프트웨어를 비롯한 추가적인 소프트웨어 구성품에 대한 safety evidence를 제공할 수 있는 인증지원 서비스도 운영합니다.

자세한 내용은 윈드리버 웹사이트(www.windriver.com/services)를 참고하세요.

윈드리버 교육 서비스

윈드리버는 언제 어디서나 이용할 수 있는 온라인 구독 기반 e-러닝을 비롯하여 전문 강사가 진행하는 온디맨드 및 멘토링 교육 서비스를 제공합니다. 자세한 내용은 윈드리버 웹사이트(www.windriver.com/education)를 참고하세요.

윈드리버 고객 지원

헬릭스 플랫폼은 수상 경력에 빛나는 윈드리버의 글로벌 지원 조직이 지원합니다. 윈드리버는 여러 시간대에 걸쳐 제공되는 실시간 지원, 다양한 자가 지원 옵션을 제공하는 온라인 윈드리버 지원 네트워크, 옵션으로 선택할 수 있는 프리미엄 서비스를 통해 고객이 최대한 신속하게 문제를 해결하도록 지원합니다.

자세한 내용은 윈드리버 웹사이트(www.windriver.com/services/customer-support)를 참고하세요.

WINDRVR