

# 항공용 운영체제 기술 동향

## Trends in Avionics Operating System Technology

김병호 (B.H. Gim)	차세대OS연구실 연구원
손동환 (D.H. Son)	차세대OS연구실 선임연구원
신창민 (C.M. Shin)	차세대OS연구실 선임연구원
임동혁 (D.H. Lim)	차세대OS연구실 선임연구원
박사천 (S.C. Park)	차세대OS연구실 선임연구원
이화영 (H.Y. Lee)	차세대OS연구실 연구원
강동욱 (D.W. Kang)	차세대OS연구실 연구원
김태호 (T.H. Kim)	차세대OS연구실 실장

### 임베디드 소프트웨어 & 시스템반도체 기술 특집

- I. 서론
- II. 항공용 운영체제 기술
- III. 항공용 운영체제 현황
- IV. 결론

항공기에서 소프트웨어가 차지하는 비중은 점점 늘어나는 추세이다. 항공 전자 시스템의 기능에 대한 요구는 점점 커지고 있으며 시스템의 복잡성을 증대시키고 있다. 이러한 추가되는 기능들로 인해 코드의 양이 늘어나고 있으며 이에 대한 비용이 증가하고 있다. 또한 컴퓨터, 센서, 디스플레이 등과 같은 항공 전자 시스템의 기술 영역들은 빠르게 성장하여 부피와 무게 및 에너지 소모 등의 비용은 줄어든 반면 성능은 크게 향상되었다. 이런 변화에 맞추어 항공 전자 시스템의 구조도 변화하고 있으며 탑재되는 항공용 운영체제도 그에 맞게 변화하고 있다. 항공용 운영체제에 필요한 기술과 최신 운영체제 현황에 대하여 소개한다.

## I. 서론

항공기도 운영체제를 통해 구동되는 임베디드 시스템으로서 일종의 전자기기와 할 수 있다. 지난 한 해 전 세계 군사용 항공기 시장은 170억 달러에 육박하는 것으로 추정되며[1] 최근엔 무인항공기 시장도 급속하게 확대되고[2] 있는 가운데 항공기 개발 비용에서 소프트웨어가 차지하는 비중은 점점 늘어가는 추세이다.

항공 전자 시스템의 기능에 대한 요구는 점점 커지고 있으며 시스템의 복잡성을 증대시키고 있다. 이러한 추가되는 기능들로 인해 코드의 양이 늘어나고 있으며 이에 대한 비용이 증가하고 있다.

한편 컴퓨터, 센서, 디스플레이 등과 같은 항공 전자 시스템의 기술 영역들은 빠르게 성장하여 10년 전보다 크기와 무게 및 에너지 소모와 같은 비용이 줄어든 반면 성능은 크게 향상되었다.

이런 변화에 맞추어 항공 전자 시스템의 구조도 변화하고 있으며 탑재되는 항공용 운영체제도 그에 맞게 변화하고 있다. 항공용 운영체제에 필요한 기술과 최신 운영체제 현황에 대하여 소개하고자 한다.

## II. 항공용 운영체제 기술

### 1. 항공 전자 시스템 구조

항공 전자 시스템은 기술이 발전함에 따라 독립형(independent) 시스템으로부터 연합형(federated) 시스템 그리고 현재 통합형(integrated) 시스템에 이르기까지 변화를 거듭해 왔다.

처음에는 항공 전자 시스템이 개별 임무장비들이 각각 장착된 독립형 아날로그 시스템으로 구성되었으나 장착 장비들이 증가하면서 항공기 중량이 과도하게 되어 나타난 것이 직렬 디지털 버스 구조를 채택한 연합형 항공 전자 시스템이다.

연합형 구조는 F-16 전투기에 처음으로 적용되었으나 기능이 복잡해지고 컴포넌트의 수가 많아질수록 전체 시스템을 통합하고 분석하고 인증하는 데 많은 노력이 요구되고 더욱이 전체 시스템에서 하나의 기능의 향상을 위해 컴포넌트나 절차가 변경되거나 새로운 기능이 추가되는 경우 전체 시스템의 인증 과정을 처음부터 다시 받아야 하는 단점이 있다.

1980년대 중반 이후 미국과 유럽에서는 항공 전자 연구개발 사업을 통해 연합형 시스템보다 향상된 시스템을 모색하기 시작하였고, 그 결과 표준 모듈을 표준 랙에 장착하여 표준 데이터 네트워크로 통신하는 이른바 통합 모듈형 항공 전자 시스템(integrated modular avionics)이 출현하게 되었다. 모듈형 시스템을 통해 일반적으로 중량, 전력, 공간, 비용 등을 줄일 수 있을 것으로 판단되나 이를 위해서는 연합형 시스템에 존재하는 서브시스템 간의 경계를 허물고 공통 여유 자원 풀을 활용하여 서브시스템 간에 공유함으로써 시스템의 가용도를 높이는 통합형 시스템으로 발전해야 한다. 통합형 시스템은 고장 허용 및 재구성을 통해 운용상의 융통성을 제공할 수 있는 부가적인 장점도 있으므로 군용 및 민간 항공기 공히 이 개념을 바탕으로 한 시스템으로 발전하고 있다.

통합형 시스템이 적용된 F-22 전투기의 항공 전자 구조는 전통적인 기능별 박스(line replaceable unit) 대신 이러한 기능들을 소프트웨어로 처리하는 공통 프로그램 가능 모듈로 되어있다. 이러한 시스템은 임무 효율의 증가는 물론 설계에 큰 융통성을 제공한다. 즉, 강건하고 고장 허용이 가능한 시스템 재구성 능력, 신뢰도, 지원성, 가용도, 중량, 확장성, 획득 및 유지 비용 측면에서 모두 우수한 특성을 가진다. F-22 항공 전자 구조는 공통 모듈화, 고집적 시스템으로 특징지어지며, 미군용 항공기 최초의 통합형 시스템이다. 두 대의 공통 통합 프로세서는 광섬유 데이터 버스로 상호 연결되고 공통의 범용 디지털 프로세서 모듈이 분산 병렬 처리 방

식으로 모든 항공 전자 기능을 수행하며, 공통의 운영체제가 모든 핵심 처리 모듈에 탑재되어 최대한의 융통성을 제공한다. 새로운 응용 분야나 신기술 적용을 위한 모듈 추가가 용이하다.

F-35(Joint Strike Fighter: JSF) 항공 전자 시스템은 F-22 시스템을 근간으로 하여 구성되었으며 통합 정도 향상, 시장구매(Commercial-Off-The-Shelf: COTS) 전자 구성품의 활용 및 개방 구조 채택 등을 특징으로 하고 있다. 미 국방성은 1994년 개방형 시스템을 위한 합동 task force를 구성하여 모든 무기체계의 전자 분야를 개방형 시스템으로 개발하도록 유도하고 있다. JSF는 1990년대에 시작된 Pave Pace 프로그램을 통해 비용 효율과 신뢰성이 높은 항공 전자 시스템을 제공하기 위해 개방형 접근 방식을 채택한 사상 최대의 군용 항공기 프로젝트로서 F-22의 항공 전자를 확장하여 통합 RF(Radio Frequency) 및 EO(Electro Optical) 센서 시스템을 채택하였다. 시스템은 6개의 주 카테고리, 즉 RF, EO, 핵심(core) 프로세싱, 조종실, 항공기 관리, 장착물 관리로 구성된다. 일반적으로 센서는 항공 전자 시스템의 비용, 중량, 체적 및 전력의 70%를 차지하므로 다기능 공유 안테나와 RF 신호 공통처리기 등을 사용하는 통합 센서 시스템을 채택함으로써 비용 및 체적은 1/4 수준으로 줄이면서 신뢰성은 10여 배 커질 것으로 예상된다.

## 2. 항공용 운영체제 주요 기술

IMA(Integrated Modular Avionics) 구조에서는 다양한 기능이 하나의 컴퓨터에서 실행되어야 하기 때문에 독립형 구조와 달리 운영체제를 필요로 한다. 이러한 운영체제는 criticality가 다른 응용들끼리 서로 영향이 없도록 시간적 공간적으로 파티셔닝을 필요로 한다. 또한 하나의 응용이 다른 시스템에도 사용되기 위한 호환성을 제공함으로써 개발 시간 및 비용을 단축할 수 있는데

이를 위해 항공용 응용 프로그램을 위한 표준 인터페이스가 항공 업계를 중심으로 논의되었고 ARINC 653이라는 표준이 만들어졌다.

### 가. ARINC 653

모듈형 항공 전자 시스템은 기존의 연합형 시스템의 전자장비들을 하나의 컴퓨팅 장비로 통합하면서 장비의 사이즈와 무게, 그리고 전력소모를 줄이는 효과가 있지만, 하나의 응용에 오류가 생기면 시스템 전체가 영향을 받아 신뢰성에 문제가 생길 수 있다는 단점이 있다. 이러한 점을 해결하기 위해 응용들 간에 독립성을 보장할 수 있는 응용체제에 대한 요구가 생겼고 이를 위해 항공 전자 관련 학계와 업계가 모여 ARINC 653이라는 항공 전자기기를 위한 실시간 운영체제와 응용 프로그램 간의 표준 인터페이스를 규정하였다[3].

ARINC 653은 항공 전자기기의 운영체제와 응용 프로그램 간의 인터페이스(Application/Executive: APEX)를 제공하며 요구되는 서비스, 확장 서비스 및 적합 테스트 3개의 파트로 구성되어 있다. ARINC 653의 핵심 서비스는 시/공간 파티셔닝으로 하나의 시스템 내에서 응용 프로그램이 실행되는 공간과 시간을 분리시켜서 하나의 응용 프로그램의 오류가 다른 응용에 영향을 미치는 것을 방지한다. 이 외에도 프로세스 스케줄링, 헬스 모니터링, 파티션 간/파티션 내 통신 등의 기능을 정의하고 있다

### 나. ARINC 653 하이퍼바이저

ARINC 653 하이퍼바이저는 가상 머신 모니터인 XEN 하이퍼바이저의 파티셔닝 기능을 활용하여 ARINC 653 표준을 만족하도록 개발된 운영체제이다[4].

가상화 기술은 컴퓨터 자원을 공유하는 방식 중 하나로, 기업이나 금융 관련 솔루션 등 다양한 분야에서 대중화된 기술이다. 최근에는 고신뢰 실시간 운영체제 분야에서도 사용 사례가 나타나고 있다. 가상화 기술은

1960년대 메인 프레임 컴퓨터의 사용량을 최대한 끌어 내기 위해 시작되었으며, 시간이 흐른 후, 가상화 기술은 메인 프레임에서 마이크로프로세서 환경으로 이동하기 위한 수단으로도 사용되었다.

하드웨어에 대한 운영체제의 가상화된 접근은 가상 머신 모니터 혹은 하이퍼바이저를 통해 이루어진다. 이를 위해 하이퍼바이저는 세 가지 중요 특징을 가져야 하는데, 실제 머신과 동일한 환경을 제공해야 하며, 가상화로 인한 성능 저하가 크지 않아야 하고, 하드웨어 자원을 완전히 제어할 수 있어야 한다.

하이퍼바이저와 OS는 API나 트랩 혹은 에뮬레이션을 통해 서비스를 제공한다는 점에서 유사하다. 하지만 하이퍼바이저가 하드웨어에 대한 접근을 가상화하고 제어하는 기능만 수행한다는 점과 디바이스 드라이버, 파일 시스템, 네트워크 스택 등 나머지 서비스는 하이퍼바이저 밖에서 수행된다는 점이 다르다. 이런 이유로 작아진 하이퍼바이저는 규모가 큰 운영체제에 비해 정형 검증을 통한 신뢰성 확보에 용이하다.

하이퍼바이저의 설계는 주로 I/O보다는 CPU와 메모리에 초점이 맞추어져 있다. 가상화 기술은 모든 레벨의 컴퓨팅 기술에 사용되는데, 클라우드 컴퓨팅, 서버 가상화, 데스크톱 가상화 등이 그 예이다. 임베디드 시스템의 가상화는 최근에서야 시작되고 있는데, 그 이유는 임베디드 시스템이 주로 하나의 목적을 위해 동작하며 실시간 성능이 중요하기 때문이고 가상화 기술을 적용하기에는 하드웨어 성능이 상대적으로 떨어지기 때문이다. 하지만 항공우주 분야는 상황이 다르다. 높은 처리량이 요구되는 동시에 공간, 무게, 전력의 제한이 있는 분야라 가상화 기술 적합하다. 휴대폰 분야에서도 사용자 애플리케이션과 실시간 태스크를 동시에 실행하기 위해 가상화 기술의 적용이 고려되고 있다.

### 3. 항공용 운영체제 인증

인증은 개발과정에서 요구되는 각종 규격 및 절차를

만족했는지, 요구되는 안전성을 확보하였는지를 확인받는 과정이다. RTCA/DO-178B(이하 DO-178B)는 항공용 시스템과 장비 인증에 대한 소프트웨어 고려 사항이다. DO-178B는 1992년 12월 제정되었으며, 항공기 탑재 소프트웨어에 대한 FAA(Federal Aviation Administration) 인증 획득을 위해서 준수하여야 할 지침을 정의한다. 이 표준 문서는 RTCA와 EUROCAE의 합의하에 국제적 적용을 위하여 개발되었으며, 미국 캐나다, 유럽의 항공업계에서 주로 참여하였다. 현재 상용 항공기 소프트웨어를 위한 일반적인 표준 인증으로 자리잡았으며, 군용 항공기 인증에도 적용되고 있다[5].

#### 가. DO-178B

항공용 소프트웨어의 개발에는 많은 비용과 시간이 필요하기 때문에, 커피머신을 제어하는 소프트웨어와 자동 이착륙을 제어하는 소프트웨어를 똑같이 엄격하게 개발할 필요는 없다. 따라서 DO-178B에서는 결함(failure)의 조건에 따라 소프트웨어의 수준을 정의하고 소프트웨어의 개발이 해당 수준에 따라 관리되는지를 인증한다. 이 표준에서는 소프트웨어의 위험성 수준을 <표 1>과 같이 5개로 분류하는데 안전 평가 프로세스와 위험도 분석으로부터 수준을 결정한다. 소프트웨어의

<표 1> DO-178B 소프트웨어 수준

레벨	고장 영향성	소프트웨어의 위험성 수준
A	Catastrophic	추락의 위험을 갖는 고장 위험 수준
B	Hazardous	성능 또는 안전에 큰 부정적인 요소를 갖거나 물리적인 변형 또는 과도한 업무 부하로 인해 조종사가 비행을 유지할 수 없는 위험 수준
C	Major	심각한 수준이지만 위험한 상태는 아님(예를 들어 승객이 다치는 것보다 불편한 상태)
D	Minor	고장 상태가 보이지만 major 상태보다는 덜한 상태 수준(예를 들어 승객에게 불편을 초래하거나 비행 경로를 변경하는 상태)
E	No Effect	안전, 비행 조정 또는 조종사 과부하에 영향을 미치지 않은 고장 수준

고장 영향성을 판단하는 기준은 해당 소프트웨어가 탑재될 하드웨어의 신뢰성 수준을 따르게 된다. 또한 각 레벨에 따라 만족시켜야 할 요건의 수와 독립성 수준이 정해진다. 표준에 따르면, '독립성'이란 해당 개발 산출물의 개발자와 그 산출물을 검증하는 책임을 가진 사람이 명확히 분리되어 있어야 한다는 의미이다.

소프트웨어 수준(A에서 D까지)에 따른 목표를 만족시키기 위해서는 프로세스가 필요하다. DO-178B에서 프로세스는 크게 개발 프로세스와 통합 프로세스로 구성되지만, 실제 프로젝트에서는 구체적인 프로세스를 만족시킨다기보다는 DO-178B의 각각의 요건을 만족시키기 위한 활동들이 진행된다.

계획 프로세스의 산출물들은 프로젝트의 시작을 위한 일반적인 요구사항들이다. 요구사항/설계/구현 프로세스에서는 개발에 요구되는 일반적 산출물들이 있으며 이들 간의 추적성이 엄격하게 요구된다. 통합 프로세스는 검증, 형상관리, 품질보증, 인증지원 프로세스로 구성된다. 검증 프로세스에서는 모든 코드에 대한 검토 분석 결과와 시험 및 시험 결과로부터 요구사항으로의 추적성 요건 만족이 일반적으로 요구되며, 요구사항 기반의 소프트웨어 시험 도구 및 코드 커버리지 분석 도구 등이 사용된다. 형상관리 프로세스를 통해 결함보고 및 변경과 관련된 활동을 정의하고 통제한다. 형상관리 프로세스에서는 일반적으로 소스 코드 개발 환경, 시험 및 분석 도구 개발 환경, 소프트웨어 통합 도구, 모든 산출 문서, 그리고 소프트웨어 및 하드웨어 등에 대한 데이터를 보관하고 갱신하는 기능을 수행한다

#### 나. DO-178C

제정된 지 20년이 지난 DO-178B로는 현재의 복잡한 시스템에 대한 안전성을 보장하기 어렵다는 문제 인식 하에 이를 극복하고자 2005년 봄 RTCA 특별위원회(SC-205)가 만들어졌다. 특별위원회의 목적은 항공 소

프트웨어의 안전 구현 증진, 시스템 개발 프로세스와 안전 프로세스 간의 명확하고 일관된 연계 방안 제공, 새로운 소프트웨어 동향 및 기술에 대한 접근, 그리고 기술발전 및 변화를 따라갈 수 있는 접근방법 구현을 찾는 것이었다. 여기서 다루었던 주요 이슈들은 산업계가 직면하고 있던 새로운 소프트웨어 기술들로서, 객체지향 기술, 외주 및 위탁 개발, 모델 기반 개발, 증가하는 소프트웨어 툴 사용, 그리고 RTOS(Real-time Operating System) 및 다른 COTS 구성품들의 사용 등이 이슈화되었다. 이러한 이슈들에 대한 연구를 통하여, 새롭게 제정된 DO-178C/ED-12C에는 기술에 독립적인 핵심 요건과 이에 대한 지침을 가능한 한 DO-178B의 요건과 가깝게 유지하면서, 특정 기술에 종속되는 부분에 대해서는 관련 지침을 요건과 함께 서플먼트의 형태로 제시했다. DO-178B에 새롭게 추가된 기술적 서플먼트는 다음과 같다.

- RTCA DO-330: Software tool qualification considerations
- RTCA DO-331: Model-based development and verification supplement to DO-178C and DO-278A
- RTCA DO-332: Object-oriented technology and related techniques supplement to DO-178C and DO-278A
- RTCA DO-333: Formal methods supplement to DO-178C and DO-278A

위에서 나열된 네 가지 기술적 서플먼트는 복잡해진 항공용 소프트웨어의 안전성을 보장하기 위해 적용되어야 할 최신의 소프트웨어 기술 요소들이다. 위의 네 가지 기술들은 이미 안전성이 중요한 소프트웨어 개발에 복합적으로 적용되고 있으며, FAA에 의한 항공용 소프트웨어 인증에 상당한 영향을 줄 것은 의심의 여지가 없다.

### III. 항공용 운영체제 현황

#### 1. VxWorks-653

임베디드 및 모바일 소프트웨어 선도기업인 윈드리버(Wind River Systems, Inc.)가 출시한 VxWorks-653은 IMA 시스템을 위한 ARINC 653 기반의 실시간 운영체제이다.

VxWorks-653은 프리스케일 e600 파워 아키텍처, 최신 커티스 라이트 컨트롤(Curtiss-Wright Controls)의 VPX6-185, 윈드리버 SBC8641D보드, GE 인텔리전트 플랫폼(GE Intelligent Platforms)의 V7768 보드를 포함하여 Intel 32비트 프로세서 아키텍처까지 지원한다. 또한 VxWorks-653 플랫폼은 전원 장애에 대비한 DO-178B 파일 시스템을 제공한다.

VxWorks-653은 보잉 787 드림라이너를 포함한 40가지 이상의 기체에서 180개 이상의 서브시스템에 전 세계 100여 고객들에 의해 사용되고 있다. VxWorks-653은 시간 및 공간의 두 영역의 엄격한 스케줄링을 제공하고 있으며, 사용자 애플리케이션을 각기 다른 DO-178B 인증 레벨을 갖는 파티션에 위치시킬 수 있다(그림 1) 참조. VxWorks-653은 또한 새로운 소프트웨어 모듈을 항공 전자 플랫폼에 추가 시 전체 환경의 DO-

178B 인증 재검사 없이도 장착할 수 있도록 하는 DO-178B 인증을 받은 시스템 통합용 개발 툴을 포함한다.

VxWorks-653은 윈드리버 workbench 내에서의 향상된 프로젝트 및 워크플로우를 지원하며, 이는 시간-공간 파티션을 나눈 VxWorks-653 프로젝트를 구상하거나 구축하고 있는 고객을 위한 것이다. 그리고 UDP/IPv4 네트워크 스택을 이용한 TCP, IGMPv1 및 멀티캐스트 지원을 포함함으로써 향상된 DO-178B 네트워크 스택을 지원한다. 또한 DO-178B TCP/IP 네트워크 스택을 포함한 VxWorks-653 런타임 구성요소에 대한 RTCA DO 178B와 EUROCAE ED-12B의 레벨A용의 제품화된 인증서 세트가 제공된다.

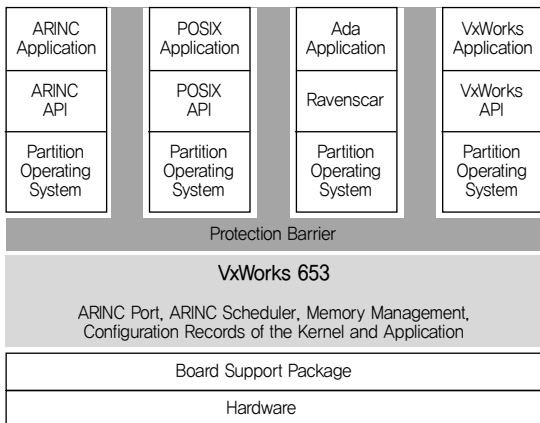
#### 2. LynxOS-178

RTCA DO-178B 표준은 항공 시스템용으로 안전성을 중시하는 소프트웨어의 엄격한 기술 요건이다. LynxWorks, Inc.가 1988년에 만든 RTOS인 LynxOS는 DO-178B 규격의 주요 구조적 고려 사항을 충족하고 있다.

LynxOS-178 패키지의 코어는 LynxOS이며 하드 리얼 타임, UNIX 스타일 운영체제로 디자인되었다. LynxOS와 LynxOS-178은 가장 엄격한 안정성이 요구되는 방산/항공 시스템에서 DO-178B 인증 채용에 기초가 되는 소프트웨어이다.

LynxOS-178의 주요 특성은 시간과 공간 모두에 있어 다중 하드 파티션을 지원한다는 것이다. 각 파티션은 완벽하게 분리되어 있어 파티션 사이의 오류가 생길 수 없다. 파티션은 오류를 분리시키고, 모든 리소스의 사용 가능성을 보장하기 위한 방법이다.

LynxOS-178의 하드 파티션은 가상머신으로 취급된다. 각 애플리케이션 프로세스는 자체 CPU 상에서 돌아가는 것처럼 자체 운영체제 환경 내에서 동작한다. 이는 시스템 애플리케이션 개발자에게 복잡한 시스템을 프로그래밍하는 부담을 덜어준다. 애플리케이션 개발자



(그림 1) VxWorks-653 구조[6]

가 시스템 파티셔닝보다 개발하는 애플리케이션에 집중할 수 있도록 파티션 관리는 VCT(Virtual-Machine Configuration Table)에 의해 관리된다.

LynxOS-178의 메모리 파티셔닝은 프로세서 MMU에 의해 가능하며, 어드레스의 브릭 월(brick wall) 파티셔닝과 각 가상 머신에서 사용 가능한 최대 메모리 양의 엄격한 할당 강화로 오류 억제 기능을 제공한다. LynxOS-178의 MMU 지원은 메모리 보호와 가상머신 사이의 하드 파티셔닝을 위한 하드웨어 맵 어드레스 보호라는 이점을 제공한다.

### 3. INTEGRITY-178B

Green Hills Software, Inc.에서 개발한 INTEGRITY-178B 운영체제는 NSA-managed NIAP lab에서 EAL6+ High Robustness 인증을 받았다. 이는 세계에서 가장 안전한 레벨의 운영체제를 의미하며, 다른 상용 운영체제는 EAL6+ High Robustness의 보안 수준을 달성하지 못했다. INTEGRITY-178B 운영체제는 FAA의 DO-178B Level A, EAL6+ High Robustness, FDA의 Class II와 Class III, ICE/EN의 Industrial Safety 61508 SIL3 등의 안전 및 보안 인증을 거쳤다.

INTEGRITY-178B의 분리형 커널은 프로세스 중 할당된 메모리 영역이 아닌 영역에 대한 사용을 방지함으로써 여러 또는 유해코드로부터 손상을 예방한다. 뿐만 아니라, INTEGRITY 파티션들은 데이터가 상주하는 파티션에 대해 의도하지 않은 외부의 데이터 접근을 차단한다. 다중의 보호된 가상 주소 공간들을 지원한다. 이들 각각은 다중 응용 태스크들을 포함하고 있으며, INTEGRITY-178B 커널은 kernel-mode task들과 함께 고유의 어드레스 스페이스 내에서 그 자체로 보호된다.

INTEGRITY-178B는 여타 시스템 또는 프로세스 이벤트와 상관없이 주어진 어드레스 스페이스를 지원하여 특정 비율의 CPU 시간 지원을 보장할 수 있는 EPS

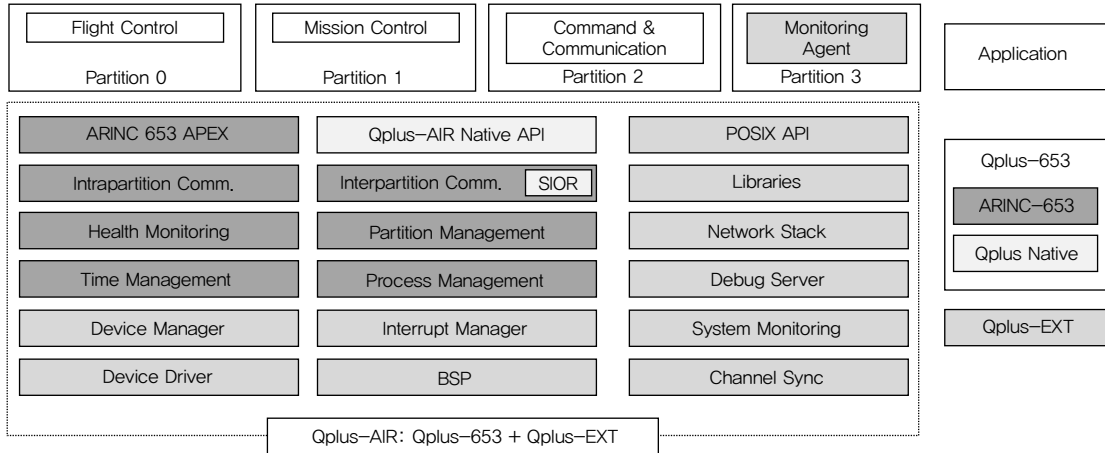
(Multi-level Optimized Enhanced Partition Scheduler)를 반영하고 ARINC 653-1을 선택사항으로 제공하고 있다. 각 파티션의 경우, EPS는 해당 어드레스 스페이스 내의 태스크에 항상 가용한 CPU time window를 지정한다. EPS는 또한 다른 파티션 내에 있는 태스크에 악영향을 미칠 수 있는 버그, 악성 코드, 바이러스 및 해커의 공격을 차단하도록 CPU time window의 경계를 정한다.

### 4. Qplus-AIR/Esto-AIR

Qplus-AIR와 Esto-AIR는 ARINC 653을 지원하는 실시간 운영체제 및 개발 도구이다. ‘무인항공기용 표준 SW 플랫폼 및 Test-Bed 개발’ 과제는 지식경제부 WBS(World Best Software) 과제 중 하나로서, ARINC 653을 지원하는 운영체제 상에서 비행 제어 SW, 임무 제어 SW 등을 통합하여 표준 SW 플랫폼을 개발하는 과제이고, 본 과제에서 Qplus-AIR라는 ARINC 653 지원 운영체제가 개발되었다. 또한, 이를 지원하는 지원 도구로 Esto-AIR도 개발되었다.

Qplus-AIR는 항공 시스템과 같은 고신뢰 임베디드 시스템을 위한 실시간 운영체제로써 ARINC 653 규격을 준수하며 APEX 인터페이스 및 POSIX 인터페이스를 제공한다. 또한 (그림 2)와 같이 파티션을 생성하고 관리하는 파티션 관리, 파티션 간의 통신, 파티션 내의 프로세스들의 스케줄링, 프로세스 간 통신 및 응용 및 시스템의 에러를 처리하기 위한 헬스 모니터링 등의 ARINC 653 기본 기능을 제공한다. 이 외에도 응용 간의 원활한 통신을 위한 shared IO region, 다채널 간 시간 동기화를 위한 channel synchronization 기능을 지원한다.

항공 분야에서는 고성능 프로세서를 사용해서 다수의 애플리케이션을 운영하는 IMA 방식이 선호되고 있다. Qplus-AIR는 플랫폼 수준에서 애플리케이션들의 안전



(그림 2) Qplus-AIR 구조

한 운영을 위해 ARINC 653을 적용한 항공용 운영체제이다. Esto-AIR는 Qplus-AIR 기반의 애플리케이션을 쉽게 개발하도록 돕는 도구이다. Esto-AIR는 이클립스 CDT 기반의 개발 지원 도구이며 두 종류의 사용자 그룹을 갖는다.

먼저 Qplus-AIR를 이용한 개발자 관점에서 컴파일/빌드 및 디버깅 기능들이 지원되며, 또한 통합자 관점에서 시간과 공간의 파티셔닝을 보장받으며 편리하게 커널 및 파티션을 설정하고 해당 설정코드를 자동 생성하는 기능이 지원된다.

설정 코드 생성을 위해서 이클립스 모델링 프레임워크를 사용해서 설정 XML 파일에 대한 중간 모델을 만들고 이로부터 코드를 생성하도록 구현하였다. 특히 원격 디버깅 및 모니터링 장비와의 연동을 위해서 디버깅/모니터링 파티션을 생성할 수 있게 했고, 사용자 편의성을 높이기 위해서 테이블 형태의 직관적인 설정이 가능하도록 했다.

#### IV. 결론

임베디드 시스템이 여러 분야에 적용되고 소프트웨어의 비중이 날로 높아지며 그 중 한 분야인 항공기에서도

소프트웨어가 매우 중요해지고 있다. 특히 항공기 IMA 구조에 따라 운영체제의 중요성이 높아졌다. 아직은 많은 기술 표준과 인증 기준도 외국산이며 업계의 대부분의 점유도 외산 운영체제가 차지하고 있는 것이 사실이다. 국산 항공기 개발에 박차를 가하고 있는 요즘 운영체제를 비롯한 소프트웨어의 국산화가 시급하다 할 수 있겠다.

#### 용어해설

**하이퍼바이저** 호스트 컴퓨터에서 다수의 운영체제를 동시에 실행하기 위한 논리적 플랫폼으로 물리적인 장치들을 가상화하는 역할을 함.

**파티션** 응용이 실행되는 단위로 내부에서 여러 개의 프로세스를 동작시킨다. 다른 응용으로 오류를 전파시키지 않기 위해 파티션들은 시간적, 공간적으로 분리되어 운용됨.

#### 약어 정리

APEX	Application/Executive
API	Application Programming Interface
COTS	Commercial-Off-The-Shelf
EO	Electro Optical
EPS	Multi-level Optimized Enhanced Partition Scheduler
FAA	Federal Aviation Administration
IMA	Integrated Modular Avionics



JSF	Joint Strike Fighter
OS	Operating System
RF	Radio Frequency
RTOS	Real-time Operating System
VCT	Virtual-Machine Configuration Table
WBS	World Best Software

## 참고문헌

[1] DefenceTalk, “The Military Aircraft Avionics Market

2012-2022,” Feb. 14th, 2012.

[2] 매일경제, “한국형 무인항공기…KAI의 도전,” 2012. 7. 1.

[3] ARINC. <http://www.arinc.com/>

[4] S.H. VanderLeest, “ARINC 653 HYPERVISOR,” *IEEE/AIAA 29th Digit. Avionics Syst. Conf. (DASC)*, 2010, pp. 5.E.2-1-5.E.2-20.

[5] RTCA. <http://www.rtca.org/>

[6] Wind River. <http://www.windriver.com/>