

무인항공기 소프트웨어를 위한 리눅스 기반의

ARINC 653 프로세스 모델

한상현[○] 이상헌 진현욱

건국대학교 컴퓨터공학부

{whacker, mir1004, jin}@konkuk.ac.kr

ARINC 653 Process Model over Linux for Unmanned Aerial Vehicles

Sang-Hyun Han[○] Sang-Hun Lee Hyun-Wook Jin

Department of Computer Science and Engineering, Konkuk University

1. 서론

항공 시스템에 사용되는 소프트웨어는 높은 신뢰성과 생산성이 요구되어 이런 특수한 상황에 맞게끔 운영체제가 연구되고 있다. 특히 통합 모듈 항공전자(IMA, Integrated Modular Avionics)라는 개념이 등장하여 여러 연구가 진행되었고 ARINC 653[1]과 같은 항공 소프트웨어 표준에 적용 되었다. ARINC 653은 항공 전자기기를 위한 운영체제와 응용 프로그램 간의 인터페이스를 규정한다. 현재 ARINC 653은 VxWorks, LynxOS 등의 RTOS에서 구현되어 항공 전자기기 및 시스템을 대상으로 적용되고 있다. 리눅스는 이런 항공시스템에 사용될 가능성이 충분히 높다. 현재 리눅스는 높은 안전성을 인정받고 고성능 서버에도 많이 사용되고 있다. 또한 리눅스는 사용하기 위해 별다른 로열티나 비용이 필요하지 않다. 그리고 사용자가 시스템을 확장하거나 수정하기에 용이하다. 이러한 이유로 점차 리눅스를 항공기에 적용시키는 노력들이 생겨나고 있다. 대부분 무인 비행체(UAV, Unmanned Aerial Vehicles)의 제어 시스템이나 대형 항공기의 객실 시스템에 리눅스 시스템이 적용되고 있고 이와 같은 사실은 리눅스 시스템이 실제 항공기에 적용될 수 있음을 보여준다. 하지만 리눅스를 적용한 연구들 중에 ARINC 653을 지원한 연구는 아직까지 수행되지 않고 있다. 그렇지만 리눅스 시스템이 ARINC 653 표준을 지원할 수 있는 잠재력은 충분하다[2]. 본 논문에서는 리눅스 시스템에서 ARINC 653의 프로세스 모델을 지원하기 위한 방안을 연구한다.

2. 리눅스 기반의 ARINC 653 프로세스 모델

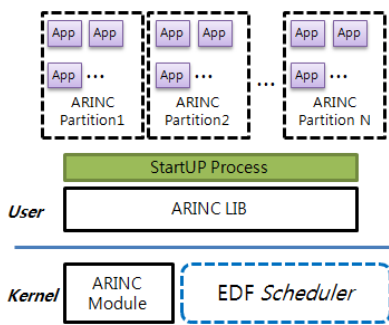


그림 1 리눅스기반 ARINC 653 지원

리눅스 환경에서 ARINC 653 표준을 지원하기 위하여 커널 영역에 ARINC 모듈, Earliest Deadline First(EDF) 스케줄러를 추가하였다. 그리고 사용자 영역에 ARINC 653 라이브러리와 StartUP 프로세스를 구성하였다. ARINC 커널 모듈은 시스템에서 수행될 파티션 정보와 프로세스들의 전체 정보를 가지고 있다. 그리고 파티션에 포함되어 있는 프로세스들의 주기와 실행시간 값을 EDF 알고리즘으로 계산하여 프로세스가 실행될 수 있는지 여부를 판단한다. EDF 스케줄러는 하나의 파티션에 속해 있는 프로세스의 스케줄링을 담당한다. 그리고 하나의 파티션 마다 소속되어 있는 프로세스들은 Red Black-Tree로 구성되어 있다. 사용자 영역의 ARINC 653 LIB는 ARINC 653 표준 API들을 프로세스들에게 제공한다. 그리고 제공된 API를 통해 프로세스들은 커널의 ARINC 모듈과 통신할 수 있다. 이를 통해 사용자 영역의 파티션 정보나 프로세스 정보를 커널 영역의 ARINC 모듈에 전달이 가능하게 된다. ARINC 653은 파티션 생성과 파티션 주기, 수행시간, 이름 등 많은 정보들을 시스템 개발자가 미리 설정 하도록 정의한다. 따라서 이런 정보들을 읽어 들여 실제로 파티션을 생성하고 스케줄러에서 사용할 수 있도록 전달해 주는 개체가 필요하다. 이를 위해 StartUP 프로세스는 파티션의 생성, 환경 설정파일 로딩 등 일련의 작업을 담당한다. 그림 1은 본 논문에서 제안하는 모델을 나타낸다.

3. 구현 및 성능측정

본 논문에서는 연구의 초기 단계로 하나의 파티션만을 지원하는 ARINC 653 프로세스 모델을 구현하였다. 그리고 파티션 안의 프로세스 스케줄링을 위한 EDF 스케줄러로 EVIDENCE[3]사의 SCHED_DEADLINE을 사용하였다. 구현 대상 시스템은 Ubuntu 리눅스(커널 버전 2.6.32.7)를 사용하였다. 구현된 소프트웨어는 다음과 같이 구성되어 있다.

본 논문은 지식 경제부의 대학 IT 연구센터 육성 지원 사업 중 무인 비행체를 이용한 실시간 방재 시스템연구(ITRC-C1090-1031-0003)과 차세대 임베디드 소프트웨어 공학의 세계적 선도 능력 확보 산업(WCU-2009-A419-0115)의 연구 결과로 수행되었음.

StartUP: 시스템 구동 시 정적으로 설정된 config 파일을 읽어서 해석한 정보를 커널 모듈에게 전달하는 역할을 한다. 그리고 ARINC 653 프로세스를 생성할 때 처음 진입점 역할을 한다.

ARINC_LIB: ARINC 653의 표준 API가 구현되어 있다. 각 API안에는 SCHED_DEADLINE에서 사용되는 시스템 콜이 호출되고 ARINC 653 표준에 맞는 프로세스 구조체와 스케줄러에서 사용될 프로세스 정보를 매핑한다. 그리고 커널 모듈에 요청하여 현재 생성될 프로세스가 EDF 스케줄러 알고리즘에 의해 정상적으로 스케줄링이 가능한지 문의하는 역할을 수행하는 과정이 포함되어 있다.

ARINC_MODULE: ARINC 모듈은 파티션과 프로세스의 정보를 저장하고 이것들이 시스템에 실행될 수 있는지를 검사하는 역할을 한다. 먼저 모듈이 실행되면 모든 파티션의 정보를 저장할 수 있는 리스트를 생성한다. ARINC 653에서 규정하고 있는 최대 파티션의 개수는 32개 이므로 모든 파티션에 대해 저장 공간을 미리 설정해도 큰 오버헤드가 되지 않는다. 또한 미리 저장 공간을 설정해 놓음으로 인해 config 파일에 설정된 파티션들이 시스템에 생성 가능한지 여부를 미리 알 수 있다.

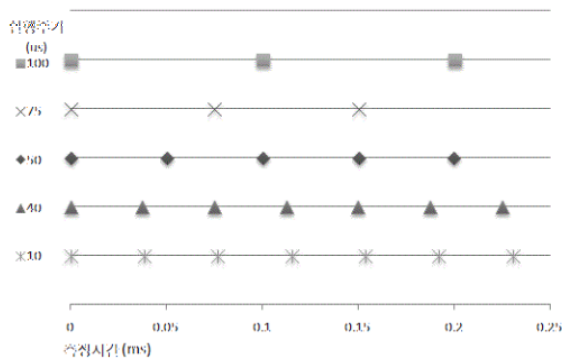


그림 2 최소 실행 가능 주기 측정 결과

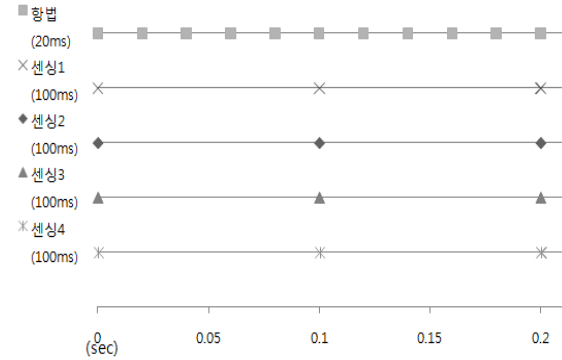


그림 3 실제 항공 소프트웨어 시나리오 적용 실험결과

이렇게 구현한 ARINC 653 프로세스 모델의 성능을 측정하기 위해 2가지 실험을 수행하였다. 우선 본 논문에서 구현한 리눅스 기반 ARINC 653 프로세스 모델이 지원할 수 있는 최소 실행 가능 주기를 측정하기 위하여 다음과 같은 실험방법을 사용하였다. 하나의 ARINC 653 프로세스를 생성하여 프로세스가 실행되는 주기를 변경해 가며 실험을 진행 하였다. 실험 횟수가 증가 할수록 프로세스가 실행되는 주기는 짧아진다. 그림 2는 최소 실행 가능 주기 측정 결과를 보여준다. 가로축은 실험을 진행한 시간을 나타내며 밀리초(ms)의 단위를 가지고 세로축은 프로세스가 희망하는 주기를 나타내며 마이크로초(us)의 단위를 가진다. 실험의 결과를 보면 40us까지 프로세스는 정확하게 정상적인 주기에 수행됨을 알 수 있다. 하지만 그 미만부터는 프로세스의 실행 주기가 정확하게 지켜지지 않는다. 따라서 본 논문에서 제시한 구현은 측정 환경에서 프로세스의 주기를 40us의 시간 해상도까지 지원해줄 수 있음을 알 수 있다. 이는 매우 높은 정확도를 가지고 있다고 판단할 수 있다.

두 번째로 항공기에 적용된 소프트웨어를 가지고 주기를 설정하여 실험을 진행하였다. 실험에서 참고한 프로세스의 주기 및 작업은 한국항공우주연구원에서 QNX 기반으로 개발된 무인헬기[4]에 탑재된 항공제어 소프트웨어를 참고하였다. 이 소프트웨어는 5개의 프로세스를 가지고 있다. 20ms의 주기를 가지고 실행되는 항법 프로세스와 100ms의 주기를 가지고 센서에서 시리얼포트를 통해 데이터를 읽어 들이는 센싱 프로세스 4개로 구성되어 있다. 실제 무인 헬기를 구성할 수 없기 때문에 본 실험에서는 GPS와 같은 간단한 센서를 네 개의 시리얼 포트에 연결하고 항공기를 제어하는 프로세스들을 ARINC 653 프로세스로 변경하여 실험을 진행하였다. 그림 3은 실험을 진행한 각 프로세스가 정확한 주기에 수행이 되었는지 실험한 결과이다. 그래프의 세로축은 수행 프로세스를 의미하며 가로축은 초(sec)단위이며 실험수행 시간을 나타낸다. 그래프를 보면 각 프로세스의 실행주기가 정확히 준수되는 것을 볼 수 있다. 그래프로 나타난 실험 결과는 본 논문에서 구현한 ARINC 653 프로세스 모델 위에 실제로 운용된 무인 헬기의 제어 시스템을 수행하더라도 무리 없이 동작하는 것을 보여준다. 이를 통해 본 논문에서 제시한 ARINC 653을 위한 프로세스 모델이 실제 항공기에 적용 가능함을 알 수 있다.

4. 결론 향후 계획

본 논문에서는 항공 소프트웨어 표준인 ARINC 653의 프로세스 모델을 리눅스 환경에서 효과적으로 지원할 수 있는 설계를 제안했다. 그리고 초기 단계로 실제 리눅스 환경에 구현하고 이것의 성능을 측정하였다. 측정결과 구현된 결과물은 충분히 무인 항공기에서 활용 가능함을 보였다. 향후 여러 개의 파티션을 지원하고 효율적인 파티션간 스케줄링 기법을 제공할 수 있는 방안을 연구할 계획이다. 아울러 실제 무인항공기에 적용할 계획이다.

참고 문헌

[1] Airlines Electronic Engineering Committee, "ARINC SPECIFICATION 653 1-2", December 2005.
 [2] 이상현, 진현욱, "무인 비행체를 위한 경량 실시간 프로토콜 기반의 ARINC653 지원," 한국정보처리학회 춘계학술발표대회, 2010년 4월.
 [3] <http://www.evidence.eu.com/content/view/313/390/>
 [4] 김성필, 이장호, 김봉주, 권형준, 김응태, 안이기 "위치 예측에 기반한 무인헬기 점항법 유도법칙 개발", 항공우주기술 5(2) 1-7.