한국산학기술학회논문지
Vol. 11, No. 10 pp. 3885-3892, 2010

안전필수(Safety-Critical) 시스템의 실시간 운영체제에 대한 안전성 평가

강영두[1], 정길도[2*]
[1]한국원자력안전기술원, [2]전북대학교 전자정보공학부

Safety Evaluation on Real Time Operating Systems for Safety-Critical Systems

Youngdoo Kang[1] and Kil To Chong[2*]
[1]**Korea Institute of Nuclear Safety**
[2]**Division of Electronics and Information Engineering, Chonbuk National University**

요 약 원자력발전소의 발전소보호계통과 같은 안전필수 시스템은 예상 가능한 사고로부터 인간과 자연을 보호하기 위한 중요 기능을 수행하는 시스템으로써, 어떠한 조건 하에서도 고유의 안전기능을 안정적으로 수행할 수 있도록 설계되어야 한다. 원자력발전소의 안전필수 기능을 수행하는 계측제어시스템에 적용되는 최신의 컴퓨터에는 다양한 하부기기를 감시 및 제어하고, 응용 프로그램을 실행시키기 위한 실시간 운영체제가 탑재되어 있으며, 이러한 실시간 운영체제는 가장 엄격한 소프트웨어 품질이 요구된다. 또한, 예상 가능한 조건에서도 안전필수 시스템의 기능이 적절히 수행될 수 있도록 설계, 분석 및 평가되어야 한다. 그러나 지금까지 국내 원자력발전소 안전필수 시스템에는, 원자력 기준과 품질등급에 따라 개발된 제품이 아닌 상용제품의 실시간 운영체제를 정성적 측면에서 승인(Commercial Grade Item Dedication)하는 방식으로 적용되어 왔다. 이로 인해 실시간 운영체제가 안전필수 기능을 수행하는 데 적합한지를 평가하는 상세 방법론과 경험이 매우 부족한 것으로 파악되고 있다. 특히, 안전필수 시스템에 적용함을 목적으로 신규 개발되는 실시간 운영체제의 경우, 안전성을 평가하기 위한 적절한 방법을 도출하기에 어려움이 있는 것으로 파악되고 있다.
본 논문에서는 원전의 안전필수 기능을 수행하는 실시간 운영체제의 설계요구사항을 기반으로, 안전필수 실시간 운영체제에 대한 안전성 분석 및 평가 사례를 제시하고자 한다. 본 논문에서 제시한 상세 안전성 평가의 방법과 사례는 향후 타 산업분야에서의 안전필수 실시간 운영체제 개발 및 안전성 평가에 활용될 수 있을 것으로 기대된다.

Abstract Safety-Critical systems, such as Plant Protection Systems in nuclear power plant, plays a key role that the facilities can be operated without undue risk to the health and safety of public and environment, and those systems shall be designed, fabricated, installed, and tested to quality standards commensurate with the importance of the functions to be performed. Computer-based Instrumentation and Control Systems to perform the safety-critical function have Real Time Operating Systems to control and monitoring the sub-system and executing the application software. The safety-critical Real Time Operating Systems shall be designed, analyzed, tested and evaluated to have capability to maintain a high integrity and quality. However, local nuclear power plants have applied the real time operating systems on safety critical systems through Commercial Grade Item Dedication method, and this is the reason of lack of detailed methodology on assessing the safety of real time operating systems, expecially to the new developed one.
This paper presents the methodology and experiences of safety evaluation on safety-critical Real Time Operating Systems based upon design requirements. This paper may useful to develop and evaluate the safety-critical Real Time Operating Systems in other industry to ensure the safety of public and environment.

Key Words : Nuclear I&C Systems, Safety-Critical Systems, Real Time Operating Systems, Safety Evaluation

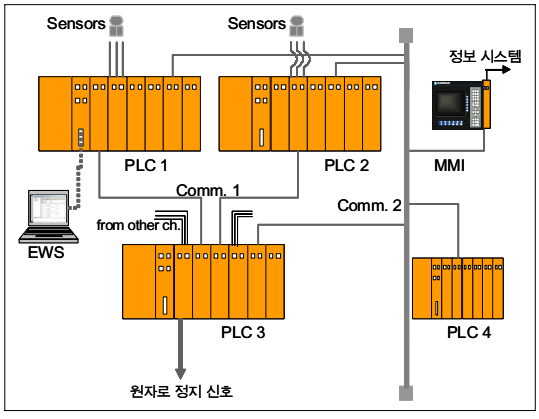
*교신저자 : 정길도(kitchong@chonbuk.ac.kr)
접수일 10년 08월 18일 수정일 (1차 10년 09월 04일, 2차 10년 09월 07일) 게재확정일 10년 10월 15일

1. 서론

국내 가동 중 또는 설계중인 원자력발전소의 계측제어 시스템은 원자력발전소 전체의 다양한 변수 또는 시스템의 연속적인 감시를 통해, 기 설정된 운전범위 이내로 이러한 변수와 시스템을 유지하는 기능을 수행한다. 특히, 원자력발전소의 예상되는 운전사건으로 인해 핵연료제한치가 기 설정된 값을 초과하지 않도록 시스템 또는 기기를 자동 제어하는 안전필수 기능을 수행한다. 이러한 안전필수 기능을 수행하는 대표적인 시스템인 발전소보호계통(Plant Protection Systems)은 원자력발전소의 예상되는 과도상태가 발생되거나 설계기준사고의 결과를 완화하기 위해 핵증기공급계통의 상태들을 감시하며, 만약 감시된 상태가 기 정해진 설정치에 도달하게 되면 정확하고 신속하게 원자로를 정지시키는 기능을 수행한다.

전형적인 발전소보호계통은 원자로의 핵 출력, 냉각재 온도, 가압기 압력, 증기발생기 수위 등 주요 변수를 연속적으로 감시하기 위한 감지기와, 감지된 다양한 변수를 설정치와 비교하고 필요시 제어신호를 발생시키기 위한 다수의 프로세서로 구성되어 있다.

그림 1은 국내 가동원전의 발전소보호계통의 단일 채널에 대한 단순 구성도이다.



[그림 1] 발전소보호계통 1개 채널 간략 구성도

발전소보호계통은 PLC(Programmable Logic Controller)를 통해 감시되는 변수의 입력값을 처리하고, 원자로 자동정지 등의 안전필수 기능을 수행한다. 이러한 발전소보호계통을 구성하는 PLC는 입출력 기기, 메모리, 통신모듈 등의 다양한 하부기기를 감시 및 제어하고, 안전필수 기능을 수행하는 응용 프로그램을 실행시키기 위한 실시간 운영체제(Real Time Operating Systems)를 탑재하고 있다. 안전필수 계측제어 시스템에 탑재되는 실시간 운영체제는 안전기능 수행의 필수적인 역할을 수행하며, 이러한 실시간 운영체제의 안전성은 전체 원자력발전소의 안전성 및 신뢰성에 영향을 미칠 수 있으므로, 안전기능을 수행하는 데 요구되는 다양한 기능 및 성능 특성을 만족하도록 설계되어야 한다. 또한, 안전기능을 적절히 수행할 수 있음을 보장하기 위해 가장 엄격한 품질보증체계에 따라 개발되고 운영되어야 한다. [1][2][3]

2. 안전필수 실시간 운영체제

2.1 실시간 운영체제의 개요

디지털 기술이 적용된 컴퓨터 시스템은 복잡하고 고성능인 하드웨어 자원의 효율적인 사용을 위해 운영체제 소프트웨어를 사용한다. 컴퓨터의 운영체제는 메모리의 관리, 통신, 스케쥴링, 응용프로그램과 하드웨어간의 인터페이스 등 컴퓨터의 모든 운영을 제어하는 기능을 수행한다.

안전필수 계측제어 시스템에 적용되는 실시간 운영체제는 최악의 상황에서도 모든 태스크가 데드라인 이내에 실행되어야 하는 경성 실시간(Hard Real Time) 특성을 보장하여야 한다. 실시간 운영체제가 원자력발전소의 안전필수 계측제어계통에 적용되기 위해서는 엄격한 성능 요건을 만족해야 하며, 까다로운 제약조건 하에서 다양한 기술적 현안사항들을 해결해야 한다. [6]

지금까지 국내 원자력발전소 안전필수 시스템에는, 원자력 기준과 품질등급에 따라 개발된 제품이 아닌 상용제품의 실시간 운영체제를 승인(Commercial Grade Item Dedication)하는 방식으로 적용되어 왔다. 상용제품의 인증 방식은 다양한 환경에서의 많은 운영 경험 또는 블랙박스 시험 등을 통해 원자력발전소의 안전필수 시스템에 적용할 수 있도록 하는 정성적 측면의 방법이다. 그러나 상용의 실시간 운영체제에 대한 인증 방식은 소스코드 단위모듈 시험 등을 수행할 수 없는 등 설계의 요구사항과 품질을 만족하는지를 명확히 파악하고 판단하는 데에는 제한적이다.

2.2 국내 원전의 실시간 운영체제 적용현황

2.2.1 VRTX

VRTX는 현재 상용으로 사용되고 있는 대표적인 실시간 처리시스템 중 하나로, 스케줄링과 동기화 메커니즘 측면에서 기존의 타임 슬라이싱 운영체제의 커널과 유사하며, 기억장치의 관리 측면과 입출력 관리 측면에서 작고 빠른 우선순위 스케쥴링을 제공하는 커널로 구성되어

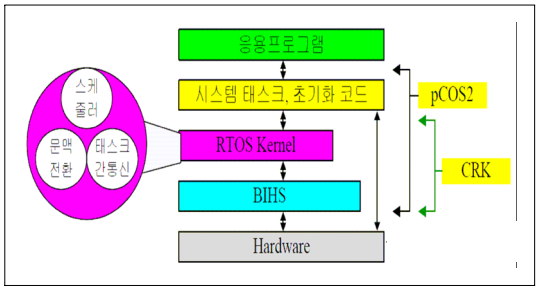
있다. VRTX는 울진 5,6호기의 디지털 발전소보호계통 및 디지털 공학적안전설비작동계통 등에 사용된PLC의 운영체제에 상용제품 인증 기법을 통해 적용되어 사용되고 있다. 이 PLC의 시스템 소프트웨어는 실시간 운영체제와 태스크 스케쥴러, 진단 기능, 통신 인터페이스 및 사용자 프로그램으로 구성되어 있다. [6]

2.2.2 QNX

QNX는 커널과 프로세스 관리기를 비롯한 여러 중요 모듈로 구성되어 있다. 커널은 프로세서간의 메시지 전송과 스케줄링을 담당하고 있으며, 네트워크 매니저, 파일 시스템 매니저, 그리고 시리얼 디바이스 매니저 등의 모듈 등으로 구성된다. QNX는 Unix 마이크로커널로서 멀티태스킹을 지원하며, 국내 신고리 1,2호기 노심보호연산기계통(CPCS, Core Protective Calculation Systems)의 보수시험반(MTP, Maintenance and Test Panel)에 탑재되어 있다. [6]

2.2.3 pCOS2

pCOS2는 국내 설계자에 의해 신규 개발되는 PLC에 탑재하기 위해 설계되고 있는 실시간 운영체제로서, 결정론적 실행을 위해 real time 커널을 사용한다.



[그림 2] 실시간 운영체제 pCOS2 구조

또한, pCOS2는 응용프로그램의 실행에 영향을 주지 않도록 진단 기능을 담당하는 태스크를 응용프로그램보다 낮은 우선순위를 갖도록 설계되는 특성을 가지고 있다. 그림 2는 pCOS2의 전반적인 구조를 보여준다.

2.3 안전필수 실시간 운영체제의 설계요구사항

일반적으로 실시간 운영체제는 결정론적 성능 특성, 효율적인 자원 관리 등을 포함하여 적용 시스템 또는 환경에 부합될 수 있는 요구사항이 충족되도록 설계되어야 한다. 안전필수 시스템에 적용되는 실시간 운영체제의 성능과 특성은 안전기능을 수행하는 시스템의 전체적인 성능에 큰 영향을 미친다. 따라서 안전필수 실시간 운영체제는 엄격한 품질보증체계에 따라 개발되어야 하며, 안전필수 기능과 성능 측면에서의 여러 요구사항에 충족되도록 설계되어야 한다.

본 절에서는 일반적인 실시간 운영체제의 설계요구사항에 기인하여, 안전필수 실시간 운영체제에서도 필수적으로 구현되고 분석되어야 하는 주요 설계요구사항을 제시하고 있다.

[표 1] 안전필수 실시간 운영체제의 설계요구사항

1	소프트웨어의 설계 품질
2	결정론적 성능
3	커널 기능 요건
4	태스크 설계 및 관리
5	메모리 및 자원 관리
6	오류 처리
7	통신 설계 및 성능

2.3.1 소프트웨어의 설계 품질

안전필수 실시간 운영체제 소프트웨어는 안전성 및 고신뢰성을 보장하기 위해, 가장 엄격한 소프트웨어 개발계획과 소프트웨어 생명주기에 따른 설계결과물이 산출되도록 개발되어야 한다. 또한, 개발 생명주기의 각 단계별 확인 및 검증(V&V, Verification and Validation)을 수행하여 이전 단계의 요구사항이 적절히 반영됨을 확인하는 절차를 수행하고, 소프트웨어의 구성요소가 안전필수 시스템에 영향을 미치는 위험요소를 분석하여 공통원인고장의 발생 가능성이 최소화될 수 있도록 개발되어야 한다. [4][5]

2.3.2 결정론적 성능

어떤 시스템이 타이밍 요건을 만족할 것인지를 예측하기 위해서는 그 시스템 내부 소프트웨어의 실행시간에 대한 예측 가능한 바운드를 설정하는 것이 필요하다. 따라서 안전필수 실시간 운영체제는 하드웨어, 소프트웨어 및 통신 연계 등을 고려하여 결정론적 실행시간 제한치를 만족할 수 있도록 설계되어야 하며, 최악의 실행시간 검증을 통해 마감시한을 만족함을 제시해야 한다. 실시간 운영체제의 타이밍 요건은 태스크 전환 지연시간, 인터럽트 지연 시간, 인터럽트 서비스 루틴 실행 시간, 태스크 실행시간 등의 수행시간이 예측 가능하도록 설계되어야 한다.

2.3.3 커널 기능 요건

커널은 태스크 스케쥴링이나 태스크간 통신 기능을 제공하는 실시간 운영체제의 핵심이다. 스케쥴러는 입력 스캔과 응용 프로그램이 각 사이클 이내에서 완료될 수 있음을 보장해야 한다. 또한, 정적 스케쥴링 알고리즘이 적용된다면 보다 높은 예측성을 확보할 수 있다.

우선순위 기반의 실시간 커널이 사용된다면, 시스템 태스크들의 우선순위 역전 현상에 유의해야 한다. 우선순위가 역전되는 현상을 최소화 시킬 수 있는 스케쥴링 알고리즘을 이용해야 하며, 시스템 태스크를 설계할 때 우선순위의 전도가 발생하지 않도록 유의해야 한다. 만약 우선순위 역전이 발생하더라도 역전시간은 제한치를 가져야 한다. [11]

2.3.4 태스크 설계 및 관리

안전필수 실시간 운영체제의 태스크는 시스템 소프트웨어가 데드라인을 만족함을 보장할 수 있도록 설계되고 관리되어야 한다. 태스크는 정적 우선순위나 정적 태스크 설계 기법을 통해 결정론적으로 설계되어야 하며, 태스크 스택은 최대 용량을 예측하고 충분한 여유도를 확보하도록 할당되어야 한다. 또한, 태스크의 최악의 실행 시간은 예측 가능해야 하며, 태스크들이 데드라인을 만족하는지 검증되어야 한다. [11]

2.3.5 메모리 및 자원 관리

안전필수 실시간 운영체제는 메모리와 자원관리 기능을 제공하도록 설계되어야 한다. 정적 메모리 할당 기법을 통해 메모리 및 자원에 대한 접근 시간은 제한치를 가질 수 있도록 설계되어야 하며, 메모리 관리 기법은 예측 가능성을 확보해야 한다.

2.3.6 오류 처리

안전필수 실시간 운영체제의 안전성 및 고신뢰성 확보를 위한 다양한 진단기능과 시스템이 정상 동작함을 감시할 수 있는 연속 감시기능이 구현되어야 한다. 안전필수 시스템이 보유해야 하는 체크섬, 워치독 타이머 등의 기능에 추가하여, 실시간 운영체제는 데드록, 스택 오버플로우, 데드라인 위배, 기아 상태 등을 검출하고 처리할 수 있는 진단 및 감시기능이 설계되어야 한다.

2.3.7 통신 설계 및 성능

통신 설계는 실시간 성능을 보장할 수 있는 상태기반(state-based)으로 설계되어야 하며, 통신 성능을 스케쥴링을 포함하여 결정론적으로 통신이 수행될 수 있도록 설계되어야 한다.

3. 안전성 분석 및 평가 사례

기존 원자력발전소의 안전필수 실시간 운영체제는 기개발된(pre-developed) 상용의 실시간 운영체제를 승인하는 방식으로 적용되었으며, 이러한 승인의 과정에서 실시간 운영체제가 안전필수 기능을 수행하는 데 적합한지에 대한 평가는 대부분 많은 운영 경험 또는 상위 시스템 수준에서의 블랙박스 시험 등을 통해 확인하는 것으로 수행되었다. 즉, 안전필수 실시간 운영체제에 요구되는 다양한 설계 특성에 대해 코드 수준의 분석 등을 수행한 연구나 평가의 사례가 매우 부족하다. 또한, 최근 국내에서는 안전필수 시스템에 적용함을 목표로 실시간 운영체제를 개발하고 있으나, 이러한 실시간 운영체제가 안전기능을 수행하는 데 충분한 품질과 성능을 보유하고 있음을 확인하고 평가할 수 있는 객관적인 방법론에 대한 연구가 부족한 실정이다.

본 논문에서는 원자력발전소 안전필수 계측제어 시스템의 핵심 운영체제로 사용기 위해 개발된 실시간 운영체제가 안전기능을 수행하는 데 적합한지 여부에 대한 안전성 평가 수행 사례를 제시하고자 한다. 제시하는 안전성 평가의 방법은 본 논문에서 정리한 7가지의 설계요구사항의 일부에 대한 적합성을 평가한 것으로써, 각 설계요구사항은 여러 시험 또는 분석을 통해 확인될 수 있다.

이러한 안전성 평가의 결과는 안전필수 기능을 수행하기 위해 기 설계된 내용을 수정토록 요구하고, 최종적으로 안전필수 기능을 수행하는 데 적합한지 여부를 판단하는 데 활용되었다.

실시간 운영체제의 안전성에 대한 분석 및 평가는 국내 원전의 안전필수 계측제어 시스템에 적용 가능하도록 개발된 pCOS를 대상으로 하였다.

3.1 평가 방법

안전필수 실시간 운영체제가 설계요구사항을 충족하는지에 대한 평가는, 소프트웨어의 품질에 대한 적합성과 일부 기능 및 성능 요건에 대한 평가를 통해 수행되었다. 특히, 안전필수 소프트웨어의 위험요소로 인한 시스템 전체의 안전성 저해 가능성을 확인하기 위해, 정상적인 상황에서의 오류 처리 기능뿐만 아니라 비정상적 상황이 발생한 경우를 가정하여, 시스템의 안전기능 수행에 어떠한 영향이 미치는지에 대한 분석이 수행되었다.

본 논문에서는 이러한 다수의 안전성 평가 내용 중 다음의 세 가지 평가 결과에 대해 제시하고자 한다.

① 소프트웨어 생명주기 산출물에 대한 평가
② 모의시험을 통한 설계요건의 확인
③ 비정상적 오류 상황에 대한 특성 분석

3.2 소프트웨어 생명주기 산출물에 대한 평가

안전필수 실시간 운영체제는 엄격한 품질보증 절차에서 정의된 생명주기 공정에 따라 개발되어야 한다. 이러한 요구사항에 충족되도록 안전필수 실시간 운영체제가 개발되었는지를 평가하기 위해, 평가대상 실시간 운영체제의 생명주기 산출물인 요구사항 명세서, 설계명세서 및 코드작성 절차에 따라 구현된 소스코드에 대한 추적성, 일치성 및 정확성 분석을 수행하였다. 이러한 분석은 요건추적 매트릭스(RTM, Requirement Traceability Matrix) 분석과 단위시험(Unit Testing) 결과 분석 등을 통해 수행하였다.

Waterfall 모델에 따라 개발된 실시간 운영체제 소프트웨어의 단위 모듈별 추적성 및 일치성에 대해 RTM을 통해 분석한 결과, 표 3의 예시와 같이 소프트웨어 생명주기 산출물에 포함되지 않은 일부 함수가 소스코드에 포함되어 있거나, 산출물과 다르게 작성된 소스코드가 구현되는 등의 일부 오류사례를 확인하였으며, 결과적으로 소프트웨어의 품질에 큰 영향이 있는 것으로 분석되었다.

[표 2] 실시간 운영체제 소프트웨어 RTM 예시

SRS	SDD	Source Code	Source Code location	CT 시험
SWR1 클럭 틱 인터럽트 처리 서비스 제공	6-4.1.2.5.1 Timer 인터럽트 핸들러 (OSTickISR, TimerISR)	OSTickISR, TimerISR()	vect.asm, main.c	K-TI-5-01 : TimerISR
SWR1.1 클럭 틱은 2ms 마다 주기적으로	5.1.2 CRK 클럭 틱은 2ms마다 주기적으로 발생하며, ~	OSTickISR(), init_timer()	vect.asm, timer.c	
SWR. 2 태스크 지연 및 타임아웃	5.1.2 CRK 커널의 시간을 클럭 틱의 정수배만큼 지연시는 OSTimeDly가 있다. 6-4.1.2.4.1 Task Delay 기능 (OSTimeDly) 6-1.1.2.15 OSTimeDly()	OSTimeDly()	ucos.c	K-TI-2-01 : OSTimerDly

[표 3] 문서와 코드간 추적성 및 일치성 분석 예시

설계 문서			소스 코드		
함수명	Caller	Callee	함수명	Caller	Callee
watchdog_init	Startup		동일	main	
check_init_switch	Startup		동일	main	
OSInit	Startup	OSTaskCreate	동일	main	동일
OSStart	Startup		동일	main	
설계문서에 없는 함수			FlashBackup_SectErase	recv_cdata_block_down	OSTimeDly
설계문서에 없는 함수			makebit2word	io_update	
설계문서에 없는 함수			atoh	init_global_var	

이러한 분석 결과를 토대로, 소프트웨어 생명주기 산출물인 요구사항 명세 또는 설계 명세가 수정되거나, 소스코드의 불필요한 함수를 제거하는 등의 조치가 수행되어, 안전필수 시스템에 적용하기 위한 엄격한 소프트웨어 품질이 확보된 것으로 평가되었다.

3.3 모의시험을 통한 설계요건의 확인

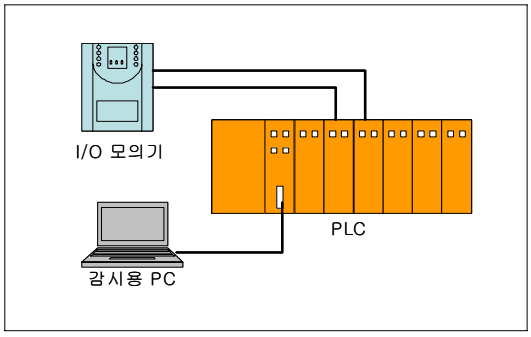
안전필수 실시간 운영체제가 설계 요구사항을 충분히 반영하여 설계되었고, 안전필수 기능을 수행하는 데 적합한지 여부는, 소프트웨어 생명주기 단계 동안 다양한 시험을 통해 확인되고 검토되어야 한다. 일반적으로 소프트웨어에 대한 시험은 단위시험, 통합시험(Integration Testing) 및 시스템 시험(System Testing)으로 구분할 수 있으며, 단위시험은 요건범위(Coverage of Requirement)와 코드 내부구조 범위(Coverage of Internal Structure)로 구분된 시험범위를 만족하도록 수행된다.

이러한 설계자 및 검증자에 의해 개발과정 동안 수행되는 소프트웨어 시험에 부가하여, 설계에서 고려된 최악의 조건에 대한 모의시험을 수행하는 것은 시스템의 안전성, 신뢰성 및 강인성을 확보하는 데 매우 유용한 방법이다.

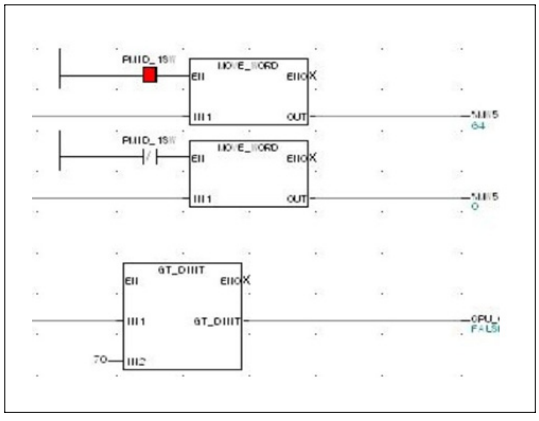
이와 같은 최악의 조건에 대한 시험을 위해, 프로세서 부하율 관련 설계요건에 대한 시험을 선정하였다. 일반적으로 원자력발전소 안전필수 시스템의 컴퓨터는 프로세서의 부하율을 60% 이내로 유지할 수 있도록 설계되고 있다. 만일 프로세서의 부하율이 60%를 초과하게 되면,

안전필수 시스템은 고장-안전 모드 또는 우선 고장모드로 진입하도록 설계된다. 평가 대상 실시간 운영체제는 프로세서의 부하율을 시스템 태스크 중 통계 태스크에서 처리하는 것으로 설계되어 있다.

평가 대상 실시간 운영체제가 최악의 상황에서도 프로세서의 부하율 감지하여 안전필수 기능을 적절히 수행하는지에 대한 평가를 수행하기 위해, 고의적으로 무한루프를 발생시키는 응용 프로그램을 PLC에 탑재하고, 엔지니어링 툴을 통해 프로세서의 부하율을 감시토록 시험 환경을 구축하였다. 기본적인 시험 환경과 프로세서 부하율 측정을 위한 시험용 응용프로그램은 그림 3 및 4와 같다.



[그림 3] 시험환경 구성



[그림 4] 부하율 측정시험용 PLC 프로그램 예시

이러한 임의 환경에 대한 시험을 수행한 결과, 부하율 60% 미만에서는 정상 상태의 범위를 보였으며, 부하율이 60%를 초과할 경우 경보가 발생되고 스캔타임 오류가 발생됨을 확인할 수 있었다. 그러나 PLC의 프로세서 부하율이 99% 이상인 경우에도 부하율에 대한 감지 기능을 수행할 수 있을 것으로 예상되었으나, 정상적으로 산출되지 않은 것으로 확인되었다. 즉, 프로세서의 부하율이 최악의 상황에 도달하였을 경우, 안전필수 시스템이 부하율 초과에 따른 고장-안전모드 또는 우선고장모드 진입을 수행하지 못할 가능성이 존재하는 것으로 분석되었다. 이러한 원인은 응용 프로그램의 과부하로 실시간 운영체제의 시스템 태스크 중 우선순위가 낮은 통계 태스크가 수행되지 못하였기 때문인 것으로 분석되었다.

이러한 문제를 해결하기 위해 통계 태스크의 우선순위를 사용자 태스크보다 높도록 변경할 수도 있으나, 이는 안전필수 시스템의 결정론적 성능 특성을 만족하지 못하는 결과를 초래할 수 있다. 따라서 프로세서의 부하율을 측정하는 기능을 우선순위가 높은 Shell 태스크가 계산하여 감지할 수 있도록 설계를 변경토록 조치하였다.

3.4 비정상적 오류조건에 대비한 설계 평가

안전필수 실시간 운영체제는 응용 프로그램의 비정상적 동작을 감지하고 제어하는 기능이 제공되어야 하며, 주변기기의 비정상적 상황에 대해서도 적절히 대처하도록 설계되어야 한다. 그러나 비정상적 상황을 모두 가정하여 시험 및 분석하는 것은 매우 어렵고 많은 노력이 요구된다.

안전필수 실시간 운영체제가 비정상적인 오류조건이 발생할 경우의 대처에 대한 평가를 위해, 국제 표준인 IEC 60880에서 제시하고 있는 설계 항목에 근거하여 총 6가지 분류의 17개 오류항목이 선정되었으며, 시험 또는 분석을 통해 오류조건에 대한 대처능력을 분석 및 평가하였다.

17가지의 오류항목에 대한 분석은 소스코드에 대한 검토 또는 결함 주입(메모리 일부 영역의 임의 변경 등) 방법을 통해 수행되었다. 예를 들면, 특정 코드 영역(0x00023～0xa00055)을 '0'으로 세팅하였을 경우, 이러한 변경을 검출하는지에 대해 엔지니어링 툴을 통해 확인하였다.

표 4는 선정된 17개 항목의 오류조건과 그에 대한 시험 및 분석 결과의 다음 세 가지 일부 시험 시나리오에 대한 예시를 보여주고 있다.

① 무한루프를 유발하는 입력은 없는가?
② 코드나 데이터를 포함하는 메모리 부분에 의도하지 않은 변경을 검출할 수 있는가?
③ 소프트웨어의 이상 행위를 검출할 수 있는가?

[표 4] 비정상적 오류조건에 대한 분석 결과 예시

항목	함수	시험절차	입력값	결과
①	OSTimeTick	대상 함수 진입지점에 BP 설정, 입력값 변경후 결과 확인	ptcb->OSTCBNext=ptcb	무한루프 발생
②	임의지점	코드영역 변경 후 검출여부 검사	0xa00023 (os text영역) ~0xa00055 에 0 세팅	변경 미검출
③	임의지점	stack overflow 유발 후 동작 검토	재귀 함수 호출	OS 정지

응용 프로그램의 무한 루프, 데드록 등 제어 흐름에 비정상적 조건이 발생하는 오류조건에 대한 실시간 운영체제의 대처 능력을 평가한 결과, 일부의 시험 사례에서 메모리 의 코드영역에 대한 변경을 검출하지 못하거나 운영체제가 정지되는 결과가 나타나는 것으로 분석되었다. 또한, 태스크의 스택 오버플로우 여부를 감지할 수 있는 기능이 구현되어 있지 않은 것으로 확인되었다.

원자력발전소의 안전필수 시스템은 이러한 비정상적 상황을 대비하기 위해 별도의 Watchdog Timer를 통해 단일 PLC의 상태를 감시하고 있거나, 다중의 채널개념을 통해 오류 상황에 대비하도록 구현하고 있다. 그렇다 하더라도 안전필수 시스템의 핵심 실시간 운영체제는 구현 가능한 수준에서 충분히 오류상황에 대처할 수 있는 기능을 보유하는 것이 타당한 것으로 검토되었으며, 이에 따라 시스템 태스크 및 사용자 태스크의 스택 오버플로우와 메모리 오류가 발생 시 메모리 오류에 대한 감지기능을 구현하도록 조치하였다.

4. 결론

원자력발전소의 발전소보호계통 등 안전필수 시스템에 적용되는 실시간 운영체제는 엄격한 품질보증체계에 따라, 안전필수 기능을 수행하는 데 필요한 다양한 요구사항에 충족되도록 개발되어야 한다. 그러나 안전필수 기능을 수행하기 위한 다양한 요구사항을 충분히 반영하고 있음을 확인하는 안전성 평가에 관한 연구 또는 실제 적용 사례는 매우 부족한 실정이다.

본 논문에서는 소프트웨어 설계 품질 등의 설계요구사항을 기반으로, 원전 안전필수 시스템에 적용하기 위해 신규 개발된 실시간 운영체제의 안전성에 대한 분석 및 평가 사례를 제시하였다. 이러한 분석 및 평가의 방법은 향후 신규 개발되는 원자력발전소의 안전필수 실시간 운영체제에 대한 적합성을 판단하는 데 활용될 수 있을 것으로 기대한다.

또한, 본 논문에서 제시된 실시간 운영체제의 안전성 평가 방법 및 사례는 안전필수 기능을 수행하는 산업분야의 실시간 운영체제에 대한 설계 및 분석에 활용될 수 있을 것으로 기대되며, 향후 본 논문에서의 방법 및 사례를 토대로, 원자력발전소 등 안전필수 소프트웨어의 개발에 활용 가능한 기술기준 또는 지침의 개발에 활용될 수 있을 것으로 기대된다.

참고문헌

[1] IEEE Standard 7-4.3.2-2003, "Criteria for Digital Computers in Safety Systems of Nuclear Power Generating Stations"

[2] IEEE Standard 1012-2006, "Software Verification and Validation"

[3] IEC 60880, "Nuclear Power Plants - Instrumentation and control systems important to safety - Software aspects for computer-based systems performing category A functions"

[4] KEPIC EME 3700, “소프트웨어 검토”, 전력산업기술기준, 대한전기협회, 2007.

[5] KEPIC EME 3400, “소프트웨어 프로젝트 생명주기(SPLC) 공정 개발”, 전력산업기술기준, 대한전기협회, 2007.

[6] KINS/RR-483, 원전 실시간운영체제에 대한 평가기술 개발, 한국원자력안전기술원, 2007.2.

[7] 김장렬외, “원전 상용기기(Commercial Grade Item) 승인 및 평가 방법론, 한국원자력학회, 추계학술발표대회, 1997.

[8] 구철회, “실시간 소프트웨어 개발기술 동향”, 항공우주산업기술동향 2권1호, pp 86-93, 2004.

[9] 이영준외, “설계명세서를 이용한 안전등급 PLC 운영체제 컴포넌트 시험방법”, 한국컴퓨터종합학술대회 논문집, Vol. 33, No. 1(C), 2006.

[10] 연제명외, “실시간 운영체제를 적용한 제어시스템의 모델기반 설계 및 검증”, 한국자동차공학회 논문집 제16권 제2호, 2008.

[11] Hyung Tae Kim, et al, "A Study on Requirements of Real-time Operating System for Safety Evaluation in Nuclear Power Plants", American Nuclear Society, NPIC/HMIT, 2009.

강 영 두(Youngdoo Kang) **[정회원]**

- 1998년 2월 : 전북대학교 제어계측공학과 학사
- 2000년 2월 : 전북대학교 제어계측공학과 석사
- 2000년 9월 ~ 현재 : 한국원자력안전기술원 선임연구원

<관심분야>
원전 계측제어 시스템, 안전성 평가, 제어시스템 사이버 보안

정 길 도(Kil To Chong) **[정회원]**

- 1984년 5월 : Oregon State University 기계공학 학사
- 1986년 12월 : Georgia Institute of Technology 기계공학 석사
- 1993년 5월 : Texas A&M University 기계공학 박사
- 1995년 3월 ~ 현재 : 전북대학교 전자정보공학부 교수

<관심분야>
Robotics, Marine Navigation, Control Systems