

미션 크리티컬시스템에서 실시간 처리를 위한 프로세스 관리 방법

한관호*, 박근옥**, 임종태***

*(주) 휴먼테크

**한국원자력연구소

***공주대학교 컴퓨터공학과

e-mail:knu1445@lycos.co.kr

Realtime Process Management in A Mission Critical System

Goan-Ho Han*, Geun-Ok Park**, Jong-Tae Lim*

*Human Technology Company

**Korea Atomic Energy Research Institute

***Dept of Computer Engineering, Kongju National University

요 약

미션 크리티컬 영역의 실시간 소프트웨어의 프로세스들은 시스템 요구사항에 맞게 제한된 시간안에 수행되도록 프로세스를 설계되고 구현되어야 한다. 본 논문에서는 원자력발전소의 감시와 제어를 담당하는 MMIUNIX 소프트웨어 실시간 프로세스들을 관리 통제하는 방법을 제안한다. 제안한 방법이 원자력 발전소의 특수한 환경에 적합하게 사용할 수 있는지 알아보기 위해서 시간제약 조건 관점에서 성능평가를 수행한다.

1. 서론

최근 소규모의 생산 시설로써 전기와 식수를 만들 수 있는 원자력 발전 기술이 연구되고 있다. 원자력 발전소에서는 실제 유사 사고를 구현하여 훈련할 수가 없다. 이에 사용자(강사, 운전원, 감독원)들에게 한국원자력연구소는 이러한 현장과 똑같은 체험을 할 수 있게 실시간 시뮬레이터인 MMIUNIX(Man Machine Interface UNIX system)소프트웨어를 개발하고 있다.

MMIUNIX 소프트웨어는 원자력발전소 환경과 같이 현장 환경에서 발생할 수 있는 상황을 만들어 사용자가 실전과 같이 대처하는 훈련용 프로그램이다. 원자력 발전소의 MMIUNIX 소프트웨어는 시간 제약 조건이 중요한 미션 크리티컬(Mission-Critical) 시스템으로 주어진 제약상황에 맞게 설계되고 구현되어야 한다. MMIUNIX 소프트웨어는 시스템 요구사항에 맞게 여러개의 동시 프로세스들을 생성한다.

각 프로세스는 지정된 시간 내에 수행할 수 있어야 한다.

본 논문에서는 원자력발전소의 감시와 제어를 담당하는 MMIUNIX 소프트웨어에서 실시간 프로세스 관리방법을 제안한다. 본 논문 구성은 다음과 같다. 제 2장에서는 MMIUNIX 소프트웨어의 구조에 대해 설명하고, 3장에서는 제안한 실시간 프로세스 관리 방법에 설명한다. 4장에서는 원자력발전소의 시간 제약조건을 만족하는지 여부를 파악하기 위해 처리 시간 관점에서 성능평가를 실시하고, 5장에서 결말을 맺는다.

2. MMIUNIX 소프트웨어 구조

MMIUNIX 소프트웨어의 구조는 <그림 1>과 같다. 응용 소프트웨어는 신호모사기, 실행통제 소프트웨어, 그리고 인간기계연계 소프트웨어로 구성된다.

신호모사기는 원자력발전소의 노심(Reactor Core),

발전소의 일차계통과 이차계통에 대한 열수력학적 특성을 수학방정식으로 표현하여 모의 시험 데이터를 생성하는 일을 수행한다.

실행통제 소프트웨어와 인간기계연계 소프트웨어는 각각 GUI 부분과 논리처리 부분으로 나눌 수 있다. 실행통제 소프트웨어의 GUI 부분은 강사가 시뮬레이션 엔진에 대한 명령(run, freeze등)을 사용할 수 있는 그래픽 기반의 입력 인터페이스를 제공함과 동시에 입력 명령에 대한 시뮬레이션 엔진의 처리진행과 처리결과를 표시화면에 출력한다. 실행통제 소프트웨어의 논리처리 부분은 GUI 부분으로부터의 강사 입력 명령을 해독하여 시뮬레이션 엔진에 전달하고 시뮬레이션 엔진의 응답 및 처리결과를 저장 및 가공하여 GUI 부분 및 통신관리자에게 전달한다. 인간기계연계 소프트웨어의 GUI 부분은 발전소 운전상태 감시정보를 표시화면에 그래픽 기반으로 표시하고 운전원의 제어요구를 소프트웨어가 부르는 인터페이스로부터 입력받아 인간기계연계 소프트웨어의 논리처리 부분에 전달한다. 소프트웨어기는 전자회로 스위치를 그래픽 객체로 형상화한 것이다. 인간기계연계 소프트웨어의 논리처리 부분은 소프트웨어기로부터 명령을 받아 통신관리자를 통해 시뮬레이션 엔진에 전달하고, 통신관리자로부터 시뮬레이션 엔진의 데이터를 가공하여 GUI 부분에게 전달한다.

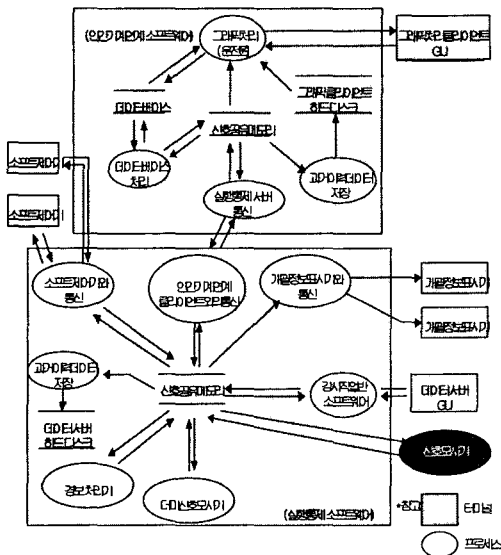


그림 1. MMIUNIX 소프트웨어의 프로세스 구조

3. 실시간 프로세스 관리방법

MMIUNIX 소프트웨어에서 독립적인 프로세스를 관리하기 위해서는 프로세스간의 우선순위, 프로세스간의 통신, 프로세스의 생성과 소멸이 필요하다.

프로세스간의 우선순위는 범용 OS상에서 제어할 수 없어 본 논문에서는 시간 간격을 조절하여 우선순위를 설정한다. 프로세스간의 통신은 시스템 내부에서 제일 빠른 공유메모리 방법을 사용한다. 외부 시스템일 경우에는 TCP/IP 기반인 Socket을 사용하여 처리한다. 프로세스간의 생성과 소멸은 MST(Master Scheduling Task)를 이용하여 관리한다.

실시간 프로세스들은 부모 자식 관계의 구조를 가지며, fork()명령을 사용하여 자식 프로세스를 생성한다. <그림 2>는 MMIUNIX 소프트웨어의 프로세스 생성 순서이다.

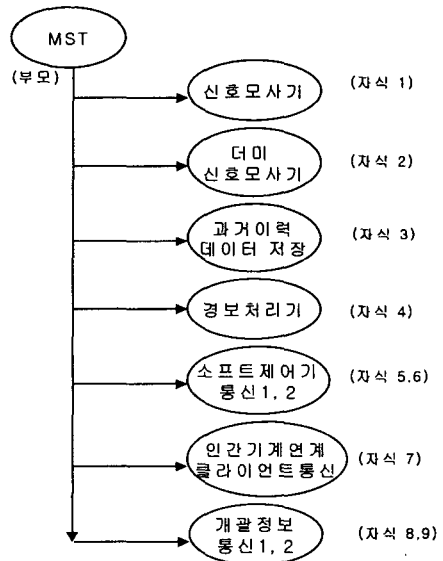


그림 2. MMIUNIX 소프트웨어의 프로세스 생성

3.1 신호모사기와의 연계 프로세스

실행 통제 소프트웨어와 신호모사기는 서로에게 할당된 자원(공유메모리)의 사용 권한에 대해 협조해야 하는 문제가 있다. 즉, 한 프로세스가 공유메모리 영역을 읽는다면, 다른 프로세스는 데이터를 처리하기 전에 읽기가 끝나기를 기다려야한다. 경보처리 프로세스의 지역메모리는 공유메모리 구조와

같으며, 경보처리 프로세스가 매 1초마다 공유메모리의 데이터를 읽어 지역메모리에 저장한다. 공유메모리를 중심으로 경보처리 프로세스와 신호모사기의 연계 관계는 <그림 3>와 같다. 프로세스간의 통신은 더미(dummy) 신호모사 프로세스, 과거 이력 데이터 저장 프로세스, 인간기계연계 클라이언트와의 통신 프로세스, 소프트웨어기와의 통신 프로세스, 개발정보표시기와의 통신 프로세스도 동일하다.

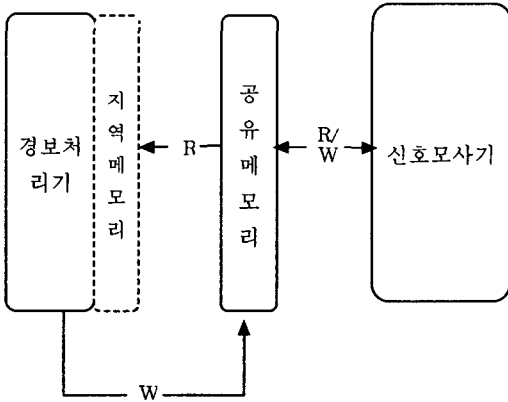


그림 3. 경보처리기와 신호모사기의 연계처리 (R:Read, W:Write)

3.2 데이터 통신 처리 프로세스

데이터 통신 처리 프로세스는 <그림 4>의 실행통제 소프트웨어와 인간기계연계 소프트웨어의 통신 연계 구조와 같다.

실행통제 소프트웨어와 인간기계연계 소프트웨어의 통신은 TCP/IP 프로토콜을 이용하여 매 0.5초 주기로 통신한다. 실행통제 소프트웨어의 통신모듈은 인간기계연계 소프트웨어 통신모듈과 교신하여 매 0.5초 주기로 신호모사기에 의해 발생하는 신호의 현재값을 인간기계연계 소프트웨어로 송신한다.

인간기계연계 소프트웨어의 통신모듈은 매 0.5초 주기로 실행통제 소프트웨어로부터 신호의 현재값을 수신하여 공유메모리 영역에 저장한다. 또한 인간기계연계 소프트웨어의 통신모듈은 공유메모리 영역에서 운전원의 정보인지 행위에 따른 신호값을 읽어 실행통제 소프트웨어의 통신모듈에 전달한다.

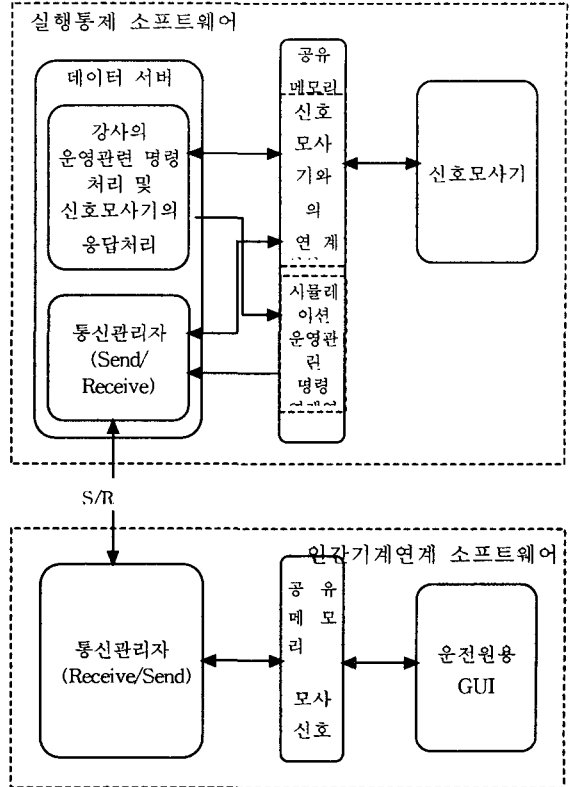


그림 4. 실행통제 소프트웨어와 인간기계연계 소프트웨어의 통신 연계 구조(S:Send, R:Receive)

3.3 MST 처리 프로세스

MST는 각각 생성된 프로세스들을 항상 체크하여 건전성 유무를 보여주고, 좀비 상태를 존재하는지 판단하여 좀비 프로세스를 대한 오류를 출력한다. MST는 일정한 시간이 지나도 더 이상 수행되지 않는 프로세스에 대해 소멸시킨다. 또한 MST는 사용자의 시뮬레이션 운영관련 데이터와 신호모사코드의 응답을 사용하여 시뮬레이션의 전반적인 실행 상태를 제어할 수 있다. MST는 사전에 정의한 공유메모리 영역의 특정 변수에 시뮬레이션 운영 및 프로세스 번호가 저장된 곳을 판독함으로써 시뮬레이션 모델의 실행을 제어한다.

시뮬레이션 운영관련 데이터 및 프로세스 번호를 공유메모리에 기록하고 응답을 받기까지 MST는 응답 대기시간을 갖는다. 응답 대기시간 이후, 신호 모사코드의 응답이 정의된 값 이외의 값이나, 사용자의 요청에 대한 올바른 응답이 아닐 경우 MST는

강사작업반 사용자에게 오류가 발생했음을 인식할 수 있도록 처리한다.

4. 성능평가

제안한 프로세스 관리 방법이 미션 크리티컬 영역의 시간제약조건을 만족하는지 여부를 파악하기 위해서 다음과 같은 시험 환경에서 성능평가를 수행하였다.

- 하드웨어 장비 : HP L2000서버
- 개발언어 : C 언어, ILOG Views 4.0

상기 시험 환경에서 MMIUNIX 소프트웨어를 구동시켜서 시스템 수행시간 대별로 각 프로세스의 처리 완료 시간을 측정하였다. <표 1>은 각각의 프로세스에 대한 성능 측정 결과이다.

표 1 프로세스 성능 측정 결과 (단위 초)

프로세스		수행시간					제한 시간
		30 분	1 시간	24 시간	96 시간	168 시간	
신호모사기와의 연계 기능		0.34	0.35	0.33	0.35	0.36	0.5
더미(dummy) 신호모사 기능		0.32	0.33	0.36	0.35	0.34	0.5
데이터통신기능	인간기계연계 클라이언트	0.36	0.37	0.41	0.35	0.37	0.5
	개괄정보1	0.38	0.39	0.40	0.36	0.39	0.5
	개괄정보2	0.38	0.39	0.40	0.36	0.39	0.5
	소프트제어기1	0.37	0.37	0.39	0.39	0.37	0.5
	소프트제어기2	0.37	0.37	0.39	0.39	0.37	0.5
과거 이력 데이터 저장 기능		0.78	0.79	0.80	0.77	0.79	1
경보 처리기 기능		0.56	0.58	0.59	0.57	0.58	1
MST 처리 기능		0.70	0.77	0.79	0.77	0.80	1

<표 1>에서 제한 시간의 의미는 각 프로세스가 처리 완료되어야 할 시간제약 조건을 의미한다. 예를 들어 신호모사기와의 연계기능 프로세스는 매 0.5초마다 수행을 시작하여 다음 수행시작까지 기능 처리가 완료되어야 한다. <표 1>에서 시스템의 안정된 상태에 있을 때, 즉 수행시간이 168시간 일때에 각 프로세스의 처리 시간이 제한 시간 내에 있음을 알 수 있다. 따라서 미션 크리티컬 시스템인 MMIUNIX 소프트웨어에서 제안된 프로세스 관리방법이 실시간 처리의 중요한 요소인 시간 제약 조건을 만족한다.

5. 결론

MMIUNIX 소프트웨어는 시스템 요구사항에 맞게 여러개의 동시 프로세스들을 생성한다. 각 프로세스는 지정된 시간 내에 수행할 수 있어야 한다.

본 논문은 실시간 프로세스의 생성과 소멸방법, 실시간 프로세스간의 통신방법, 실시간 프로세스간의 MST 처리방법을 제시했다. 본 논문에서는 감시와 제어를 담당하는 MMIUNIX 소프트웨어를 시간 제약 조건 관점에서 성능평가를 통해 확인함으로써 수행능력을 알 수 있었다. 또한, MMIUNIX 소프트웨어는 원자력 발전소의 특수한 환경에 적합하도록 시간제약 조건 관점에서 건전성 유무를 확인했다.

참고문헌

- [1] 박재창, "Compact Nuclear Simulator 성능향상 기술개발", 한국원자력학회, 추계학술발표논문집, 1999.
- [2] 심봉식, "Development of a full scope Human Machine Simulator for Human Factors Experiments", 한국원자력학회, 춘계학술대회논문집, 1997.
- [3] 박근옥, "설계해석코드 기반의 원자력발전소 훈련용 시뮬레이터 개발전략", 춘계공동학술회의 논문집, 2000.
- [4] 박근옥, "공유메모리 변수 기반의 CNS 응용 소프트웨어 구조", 한국정보과학회, 추계학술대회 논문집, 2001.
- [5] 병용원, UNIX 시스템, 정익사, 1998.
- [6] 송영재, 소프트웨어 엔지니어링, 3개정판, 홍릉과학출판사, 2000.
- [7] Roger S. Pressman, Software Engineering A Practitiners' Approach, 3rd Ed., McGraw Hill.
- [8] IEEE std 1008-1987, Software Unit Testing.
- [9] IEEE std 829-1998, Software Test Documentation.
- [10] IEEE std 1233-1998, Developing System Requirements Spec.