

## 함정용 전투체계 아키텍처 개선을 위한 실시간 운영체제 적용방안 연구

### A Study on Real-Time Operating Systems for Architectural Improvement of Naval Combat Systems

김 점 수*	장 혜 민*	주 정 현*	이 균 정*
Chum-Su Kim	Hye-Min Chang	Jung-Hyun Joo	Gyoon-Jung Lee

#### ABSTRACT

A combat system for navy's battleship is a system of systems who supports naval indigenous operations by integrating and inter-operating many different kind of weapon and non-weapon systems, which has characteristics of large-scale complex computing system. This paper considers a characteristics of naval combat system which has been developed by domestic technology and suggests a way to improve future naval combat system in terms of computing architecture by applying commercial real-time operating system technologies. This paper also provides an evaluation criteria for combat system adaptability of real-time operating systems.

Keywords : Naval Combat Systems(함정용 전투체계), Real-time Operating System(실시간 운영체제), Hard-Real Time (경성 실시간), Soft-Real Time(연성 실시간)

#### 1. 서론

함정 전투체계는 함정 고유의 작전 임무 수행을 지원하는 고도의 정밀성과 신뢰성이 요구되는 군사용 복합 무기체계로, 대함·대공·대잠·전자전 및 정보전을 포함하는 입체전을 수행하기 위해 센서체계, 무장체계 및 각종 통신체계등과 같이 다양한 개발기관으로부터 공급되는 이질적(운영 플랫폼 및 인터페이스 종류가 현저히 다른) 특성의 체계(부체계)들을 통합 운용

하는 네트워크 기반의 복합 무기체계이다<sup>[1]</sup>.

국내 함정 전투체계의 개발 및 운용 현황을 살펴보면, 90년대 말까지 대부분 기술도입생산이나 직도입을 통해 체계를 획득하여 운용해왔기 때문에, 국내 함정 전투체계는 운용체계의 제반 환경이 해외업체가 개발한 구조에 종속될 수밖에 없었다. 그러나, 2000년대 초부터 전투체계 획득방법이 국내 개발로 전환되면서 전투체계의 기본적인 구조를 우리 손으로 선택할 수 있는 가능성이 주어졌고 급속하게 발전하는 높은 성능의 상용 기술들을 전투체계에 적용하여 연구개발을 추진할 수 있게 되었다.

2000년대부터 개발이 시작된 국내의 함정전투체계는 과거 해외 선진업체로부터의 기술이전 기피, 과도한

† 2013년 2월 14일 접수~2013년 5월 17일 게재승인

\* 국방과학연구소(ADD)

책임저자 : 김점수(chskim@add.re.kr)

유지보수 비용 요구, 제반 기술 발전의 한계 등을 탈피하고자, 급격하게 발전하는 민수 기술과 국제 상용 표준 기술들을 적극적으로 도입할 수 있는 개방형 아키텍처를 기반으로 설계되고 있다. 합정용 전투체계를 위한 개방형 컴퓨팅 시스템 아키텍처는 과거 작전 임무 변경이나 확장의 한계, 새로운 무기체계 장비의 연동 제한, 성능 개선의 한계, 장기적 군수 지원/조달 문제, 기술 발전 수용 등의 한계를 지니는 임무지향적이고 비유연성의 특징을 지니는 폐쇄형 아키텍처와 구분되며, 정보/과학 기술의 발전을 적극적으로 수용함으로써, 임무 변화에 대한 적응과 정보처리능력의 향상, 장기적 기술 발전 능력 확보 등의 많은 기술적, 경제적 이점을 제공한다<sup>2~4)</sup>.

이는, 그동안 국내 기술로 개발 완료 또는 개발되고 있는 소형 및 중형의 합정용 전투체계가 민수 시장에서 급속하게 발전되는 상용 기술들이 신규 체계 개발 시 손쉽게 활용될 수 있도록 구조적인 기반을 갖추었음을 의미하며, 체계 성능 개선을 위해 새로운 민수용 기술을 적극적으로 조사/검토하여 체계 적용성을 고려할 수 있음을 의미한다.

본 논문에서는 개방형 아키텍처를 근간으로 그동안 국내 기술로 개발되고 있는 소형 및 중형 합정전투체계의 특성 및 제한점을 분석하고, 상용 운영체제 기술에 대한 분석을 통해 향후 국내 기술로 새롭게 건조될 중대형 수상함과 수중함을 위한 전투체계 개발 시 고려할 수 있는 아키텍처 측면에서의 개선 방안을 제시한다.

## 2. 상용운영체제 적용을 통한 합정전투체계 아키텍처 개선방안

### 가. 합정용 전투체계 특성 및 제한 사항 분석

수상 및 수중함용 전투체계는 함정에 탑재되는 각종 무기체계/비무기체계들을 통합하여 운용자에게 작전 운용 기능을 제공하는 복합무기체계로, 일반적인 체계 구성은 다음 Fig. 1과 같다.

하드웨어 및 소프트웨어적 컴퓨팅 기술 측면에서 고려할 때, 합정전투체계는 다수의 무기체계들을 통합 운용하는 복합 시스템의 특성과, 다수의 표적들에 대한 즉시 처리 능력을 보유해야 하는 실시간 시스템의 특성, 다수의 무기체계 시스템과 내부 주요 구성장비들이 주요 처리 기능을 분산하여 협업하는 분산 시스

템의 특성 등을 보유한다. 또한, 합정용 전투체계는 체계 규모에 따라 차이가 있으나, 일반적으로 다음 Table 1과 같이 20~40여개 이상의 많은 수의 처리 장비들이 연결되어 운용되는 하드웨어와 소프트웨어가 결합된 대형 복합 컴퓨팅 시스템의 특성을 보유한다. 특히, 무기체계에 대한 통합성 요구 등과 같은 기능적/비기능적 요구사항이 증가하고 정보처리 기술 및 통신 기술들이 발전함에 따라 통합에 따른 위험도가 감소함에 따라 점점 더 많은 수의 컴퓨팅 장비들이 통합되어 운용되어 합정용 전투체계는 더욱더 복잡한 대형 시스템으로 발전하는 경향을 보이고 있다.

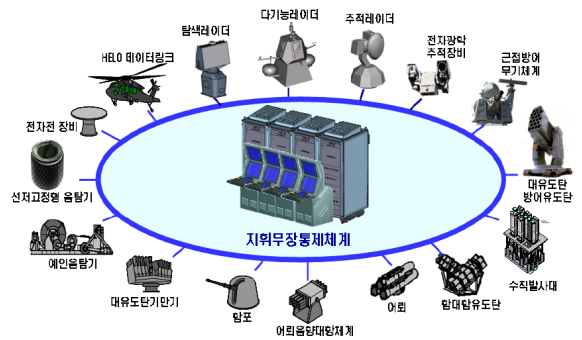


Fig. 1. General architecture of combat system

Table 1. System node size of combat system

	노드수	송신 개체수	수신 개체수	송수신 메시지수
소형체계	20여개	3,000여개	3,000여개	6,000여개
중형체계	60여개	6,000여개	8,000여개	14,000여개

일반적으로 합정용 전투체계는 센서체계, 무장체계, 지휘/무장통제체계, 통신체계 등 함정에 탑재되는 각종 부체계들을 통합하는 광의의 개념으로 정의할 수 있다. 지휘무장통제체계는 임무나 체원 등과 같은 개별 함정들의 요구 특성 상 서로 다른 공급업체들에서 독립 생산되는 다수의 무기체계들을 통합하고 상호 운용하여 개별 체계들로부터 종합된 운용 능력을 도출하는 체계이다. 본 논문에서는 협의의 개념으로 함정별로 서로 다른 탑재 무기체계들을 통합 운용해야 하는 지휘무장통제체계를 주요 개선 대상 체계로 고려한다.

Fig. 2는 지휘무장통제체계를 구성하는 주요 구성장비들을 보여준다.

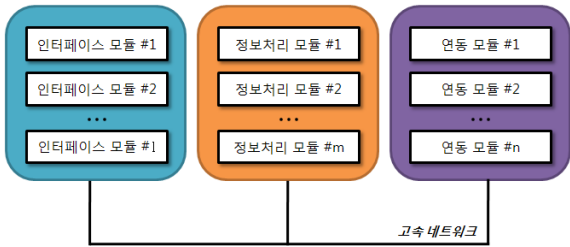


Fig. 2. Major equipments of command and fire control system

연동모듈은 센서체계나 무장체계 등의 부체계와 직접 연동하는 장비로 부체계와의 인터페이스 규약에 따라 정보처리모듈과 부체계간의 메시지 변환 및 교환을 담당한다. 정보처리모듈은 연동모듈을 통해 수집된 데이터를 처리하여 전술상황을 분석하고 무장의 교전을 통제하는 지휘무장통제체계의 핵심 장비로, 처리 기능에 따라 전술처리모듈, 훈련처리모듈, 지원처리모듈로 구분할 수 있다. 운용자인터페이스 모듈은 정보처리모듈로부터 자동 생성된 전술상황을 운용자에게 제시하고, 운용자의 지휘결심을 지원하는 장비이다. 지휘무장통제체계를 구성하는 각 장비들에 대한 운용 기능과 성능 요구사항 등을 고려하여 장비별 실시간 특성을 판단할 수 있으며, Table 2는 일반적인 장비별 주요 기능과 성능 요구사항 및 실시간 범주를 나타내고 있다. 이러한 장비에 대한 실시간 범주는 해당 장비 개발 시 하드웨어 및 소프트웨어적 관점에서 주요 고려사항으로 설정되고 해당 장비에 대한 실시간 요구사항 및 특성이 만족될 수 있도록 설계된다.

Table 2. Real-time characteristics of major equipments

	인터페이스 모듈	정보처리모듈		연동 모듈
		전술/훈련	지원	
주요 기능	- 화면전시 - 명령수신 - 전자해도	- 전술편집 - 교전통제 - 무장통제	- 계획수립 - 기록재생 - DBMS	- 연동통제 - 메시지변환 - 원시표적관리
실시간 요구도 (msec)	수백 ~ 수천	수십 ~ 수백	수백 ~ 수천	수 ~ 수십
실시간 범주	Non-RT	Soft-RT	Non-RT	Hard-RT
국내체계 운영체제	Microsoft Windows 계열	Windriver VxWorks 계열	Microsoft Windows 계열	Windriver VxWorks 계열

Table 2의 운영체제는 최근 국내 개발된 함정용 전투체계의 주요 장비별 실시간 특성을 만족시키기 위해 채택된 상용 운영체제의 종류를 보여준다. 인터페이스 모듈은 레이더비디오, 광학영상, 전자해도 및 각종 표적 정보들을 쉽게 구현하고 운용자 친숙한 환경을 제공하기 위해, 마이크로소프트의 Windows 계열 운영체제를 적용하고 있다. 정보처리모듈의 지원처리모듈은 DBMS와 대용량 데이터 저장소의 구현 등을 위해 인터페이스 모듈과 동일한 운영체제를 적용하였다. 그 외 전술/훈련처리 모듈과 연동모듈은 실시간 특성을 고려하여 Windriver사의 VxWorks 계열 운영체제를 채택하고 있다.

전투체제는 고도의 실시간 능력을 요구하는 센서체계나 무장체계와는 달리, 이러한 부체계들을 통합하여 정보를 처리하고 운용자에게 지휘 결심을 지원하는 Soft-Real Time 범주의 장비들이 대부분 임에도 불구하고, 기존의 국내 개발 전투체제는 비실시간 영역의 장비를 제외한 대부분의 장비들은 실시간 특성에 대한 요구사항을 높이 평가하여 Hard-Real Time 범주의 운영체제를 채택하고 있다. Hard-Real Time 범주의 체계는 일반적으로 단일보드컴퓨터(SBC)와 같은 실시간 내장형 하드웨어 상에서 VxWorks, LynxOS, QNX 등과 같은 실시간 운영체제를 탑재하여 고정된 수의 소규모 응용 프로그램을 정적으로 할당하여 구현되며, 응용 프로그램과 운영체제간의 구분이 모호하고, 응용 프로그램들 간의 상호 연계성이 높아 새로운 기능의 추가나 복잡한 응용프로그램의 구동, 동적인 실행 등을 구현하는데 한계가 있다. 또한, 이러한 운영체제는 급속하게 발전하는 민간 상용 소프트웨어 기술들의 활용성이 제한된다.

나. 상용 운영체제 적용을 통한 아키텍처 개선 방안

앞서 언급한 바와 같이 현대의 전투체제는 작전 환경의 변화, 운용자의 요구조건 변경 등에 따라 새로운 기능적/비기능적 요구사항들이 증가하고 있고, 체계 규모 측면에 있어서도 Table 1에서 보는 바와 같이 중대형 전투체제로 갈수록 크게 증가하게 된다. 따라서, 전투체계를 구성하는 주요 장비들에 대한 실시간 요구사항을 만족하면서도, 발전하는 민간 기술들을 쉽게 수용할 수 있고, 중대규모의 복잡한 전술프로그램들을 효과적으로 처리할 수 있도록 전투체계 구성장비들을 위한 탑재 운영체제에 대한 제고가 필요한 시점이다.

폭넓은 사용자 및 개발자들에 의한 기술발전 속도가 상당히 빠르고 다양한 개발 방법론과 DBMS와 같은 다수의 어플리케이션들의 활용성, 각종 하드웨어 드라이버에 대한 지원 능력, 네트워크 통합 및 통신 능력, 다양한 하드웨어 플랫폼 상에서의 운용능력 등을 보유하고 있으며, POSIX 호환 운영체제로 미해군의 개방형 아키텍처 참조모델에서 권장되는 운영체제의 하나인 리눅스/실시간 리눅스 운영체제는 향후 전투체계 개발을 위한 상용 운영체제 환경으로서 좋은 대안이 될 수 있다.

Table 3의 1, 2단계는 리눅스 운영체제를 이용한 합정용 전투체계 지휘무장통제체계 주요 구성장비들에 대한 운영체제 적용 방안을 보여준다. 1, 2단계의 구분은 기 개발된 국내 전투체계 전술 어플리케이션들의 재사용성과 하드웨어 아키텍처에 대한 변경 소요를 고려한 것이다. 2단계는 장기적 관점에서 서버급 PC를 통한 클러스터 컴퓨터와 같은 발전하는 최신의 하드웨어 플랫폼 상에서 JAVA, DBMS와 같은 풍부한 민간 어플리케이션들을 활용할 수 있는 구조로, 이를 통해 다양한 기술적 대안들을 손쉽게 확보할 수 아키텍처이다.

최근 선진국에서도 수상/수중함용 전투체계를 위해, 기존의 Hard-RT 범주의 운영체제에서 탈피하여 실시간 특성을 만족하면서도 소프트웨어 개발 비용, 생산성, 장기적 유지보수성 등을 고려하여 Soft-RT 범주의 운영체제 또는 Non-RT 범주의 운영체제를 주요 장비들에 대한 기반으로 채택하는 경향을 보이고 있다.

Table 3. Change plan of operating system for combat system

	인터페이스 모듈	정보처리모듈		연동 모듈
		전술/훈련	지원	
실시간 요구도 (msec)	수백 ~ 수천	수십 ~ 수백	수백 ~ 수천	수 ~ 수십
실시간 범주	Non-RT	Soft-RT	Non-RT	Hard-RT
현재	Windows	VxWorks	Windows	VxWorks
1단계	Windows	RT Linux	RT Linux	RT Linux
2단계	Linux	Linux	Linux	RT Linux

### 3. 상용운영체제의 전투체계 적합성 평가시 고려요소 및 성능측정 방법

과거에는 군사용 무기체계 개발시 개발 기관에서 독자적으로 개발한 운영체제를 사용하는 경우가 있었으나, 최근에는 거의 대부분의 무기체계 개발에서 상용 운영체제를 활용하고 있다. 이러한 상용 운영체제들을 합정용 전투체계에 상용 운영체제를 적용하기 위해서는 요구조건이 충족되는지에 대한 충분한 고려가 필요하다.

상용 운영체제는 일반적으로 크게 3가지의 종류로 구분할 수 있다<sup>[7]</sup>.

- 실시간 운영체제(VxWorks, QNX, LynxOS 등)
- 일반 목적의 운영체제(Windows 계열 Linux 등)
- 하이브리드 실시간 일반 운영체제(Linux/RT 등)

이러한 상용 운영체제들을 합정용 전투체계에 적용하기 위해서는 기본적으로 실시간 요구사항 뿐 아니라, 점차 증가하는 무기체계/비무기체계와의 네트워크 중심의 통합 운용능력 등과 같은 비기능적 요구사항들을 만족시킬 수 있어야 한다. 이를 위해서는 기본적인 OS의 성능 이외에도 다양한 범위의 확장 성능이 고려되어야 한다. 본 논문에서는 고려요소를 OS 기본 능력, OS 확장 능력, 전투체계 지원 능력, 기타 고려사항의 4가지 범주로 나누어서 고려요소를 식별하였다. 각 범주에 대한 설명은 아래와 같으며, 범주별 식별된 고려요소는 Fig. 6과 같다.

- OS 기본 능력 : Task/Process Scheduling 능력을 포함한 운영체제 자체의 실시간 성능을 평가
- OS 확장 능력 : 다중 처리 코어 활용 능력과 네트워크 통신능력 등을 포함한 운영체제의 처리 능력을 평가
- 전투체계 지원 능력 : 전술어플리케이션 성능 요구사항 지원, 전투체계 데이터서비스 지원 능력 등을 포함한 전투체계 지원 능력을 평가
- 기타 고려사항 : 장기적 기술발전 및 기술 활용성, 풍부한 기술대안 지원성 등과 같은 기타 지원 능력을 평가

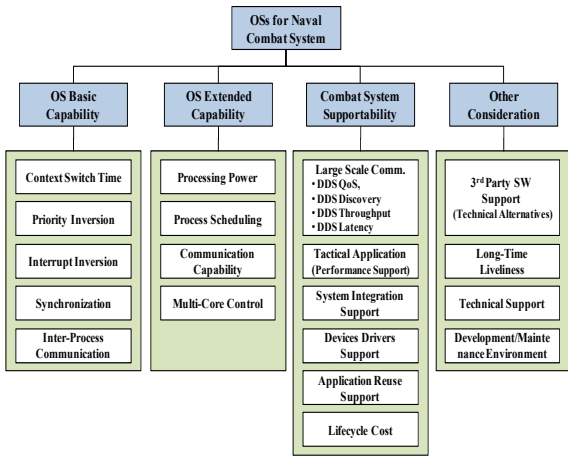


Fig. 3. Consideration of operating system for combat system

가. OS 기본성능시험

- Context Switching : VxWorks와 RT Linux는 멀티태스킹 기능을 지원한다. 이 기능은 다수의 비 동기화된 task를 가진 실시간 응용에게 매우 중요한 요소이다. 하나의 task가 중지되고 또 다른 task가 재개하여 동작하는 context switching 시간은 실시간 운영체제의 성능을 판단할 수 있는 중요한 항목이다.
- Processing Power : 응용 프로그램에서 특정 연산을 반복수행 하는 시간을 측정하여 두 실시간 운영체제의 성능을 비교한다.
- Memory Operation : 주어진 크기만큼의 메모리에 특정 값을 할당하거나 메모리 복사를 반복수행하여 그 시간을 측정함으로써 성능을 비교한다.
- OS Resource 사용량 : 응용 프로그램에서 필요한 컴포넌트만을 포함한 커널 이미지의 크기와 이미지 로딩 후 메모리 크기 사용량을 측정한다.

나. OS 확장성능시험

- 네트워크 성능 : 네트워크 성능 측정이 가능한 TTCP 프로그램을 이용하여 데이터 손실 없이 최대 수신 가능한 상태에서 성능을 측정한다.
- 미들웨어 성능 시험 : 전투체계에 주로 사용되고 있는 미들웨어 성능을 측정한다.
- 파일 시스템 검증 시험 : 파일 읽기 및 쓰기 시험을 반복 수행하여 OS의 파일 시스템을 검증한다.
- task/process 생성 시험 : 다수의 태스크를 생성하는데 걸리는 시간을 측정한다.

다. 응용 SW 분산통제 기능 구현 가능성 검토

기본적으로 분산 시스템 구조를 가지는 전투체계는 다수의 노드에 응용 SW가 실행되고 이를 적절히 통제하여 일부 노드나 응용 SW에 문제가 발생하더라도 다른 노드에서도 같은 기능을 원활히 수행할 수 있어야 한다. 이러한 기능이 구현 가능한지 운영체제별로 검토한다.

4. 상용운영체제의 전투체계 적합성 실험평가

본 논문에서는 앞에서 식별된 전투체계 적합성 평가 항목에 대하여 실험평가를 수행하였다.

가. 시험환경 구축

1) 시험환경 구축

- 하드웨어 : Ge Fanuc사의 V7768(CPU : Intel Core 2 Duo Processor, 2GB Memory)
- 운영체제 : VxWorks 6.6, Windriver Linux 3.0.2

2) 시험대상

- Vxworks/task
- Vxworks/RealTimeProcess
- Linux/process

나. 성능시험 결과

1) 기본성능시험 결과

Context switching은 VxWorks/task에서 가장 나은 성능을 보인다. 이는 VxWorks/task에서는 실시간 task만 존재하며 하나의 메모리 공간에서 모든 task가 공존하여 task를 실행하기 위한 환경을 설정하는데 드는 비용이 적기 때문이다. 이에 비해 리눅스상에서는 실시간 프로세스와 비실시간 프로세스가 공존하며, VxWorks의 RealTime Process는 커널외부에서 사용자 모드로 수행되므로 낮은 성능을 보인다.

하나의 task 또는 프로세스에서 플로팅 연산을 반복수행하는 Processing Power 시험결과는 모든 경우 유사한 성능을 보였다. 메모리 연산에서는 VxWorks/RealTime Process에서 성능이 현저하게 낮게 나오며, 이는 Posix 호환성을 제공하기 위한 구조를 채택하였기 때문이다. OS 리소스 사용량은 두 OS 모두 필요한 컴포넌트만을 포함시켜서 모듈화가 가능하여 매우 작은 크기로 최소화할 수 있다.

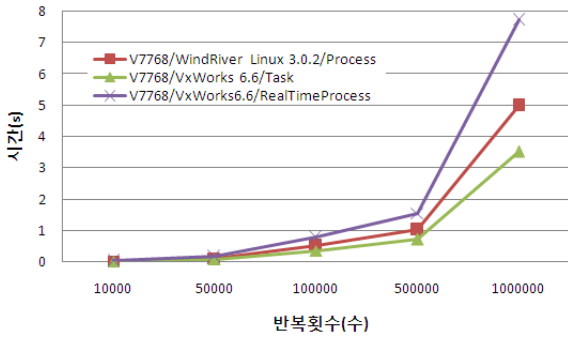


Fig. 4. Test result of context switching performance

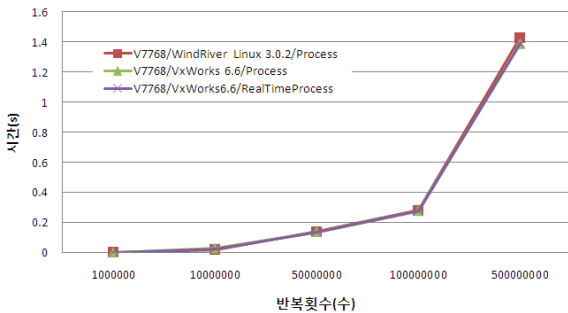


Fig. 5. Test result of processing power performance

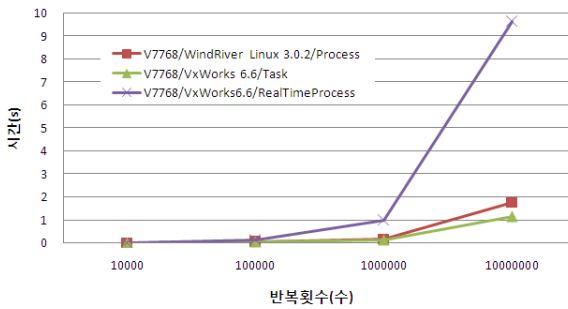


Fig. 6. Test result of memory set performance

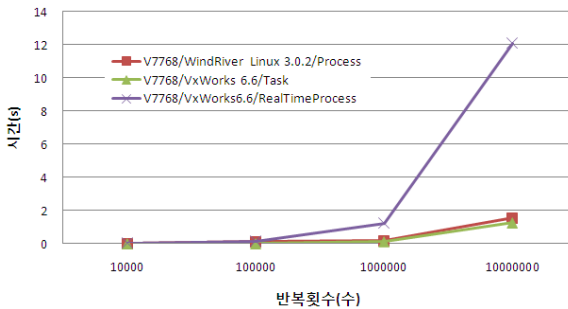


Fig. 7. Test result of memory copy performance

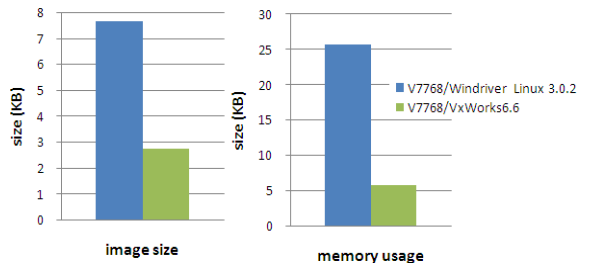


Fig. 8. Comparison of resource usage

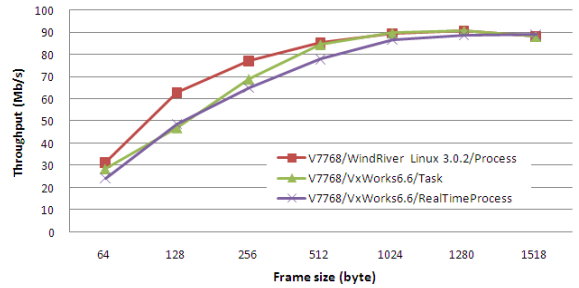


Fig. 9. Test result of network performance

## 2) 확장성능시험 결과

네트워크 및 미들웨어 성능시험 결과 운영체제별로 모두 유사한 성능을 보인다. 이는 네트워크 관련 성능은 운영체제 차이보다 랜 카드, 랜 케이블 및 네트워크 드라이버에 의해 결정되기 때문이다. 파일 읽기 시간은 운영체제별로 유사하며, 파일 쓰기 시간은 리눅스가 VxWorks에 비해 약 1.9배 이상이 걸린다. 이는 파일에 쓰기 동작과정에서 메모리 상에 있는 데이터를 실제 파일에 쓰는 주기와 방식의 차이로 인한 것이다. task 및 프로세스의 생성 시간을 살펴보면 VxWorks RealtimeProcess가 다른 시험 대상 항목에 비해 현저히 낮은 성능을 보인다. 이는 RealtimeProcess가 커널 외부

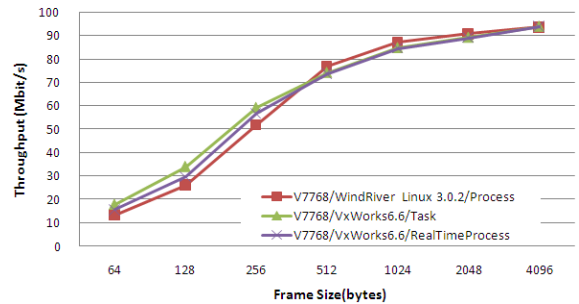


Fig. 10. Test result of middleware performance(unicast)

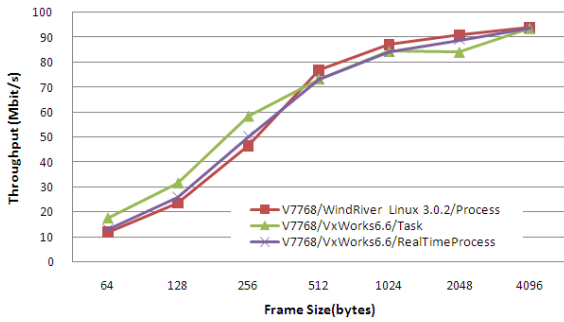


Fig. 11. Test result of middleware performance(Multicast)

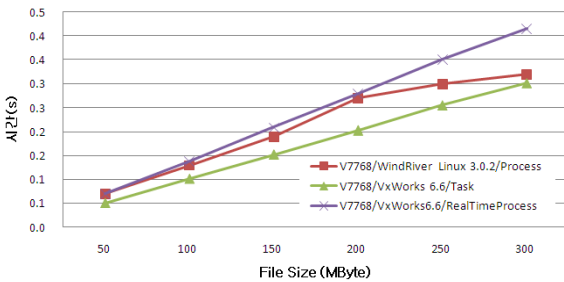


Fig. 12. Test result of file system performance(read)

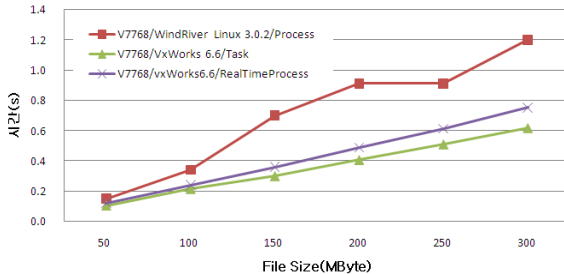


Fig. 13. Test result of file system performance(write)

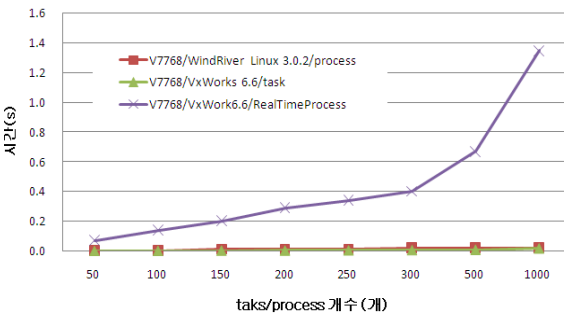


Fig. 14. Test result of task/process management performance

에서 유저 모드로 동작하기 때문이며, 따라서, 다수의 프로그램이 동시에 동작해야하는 경우 RealtimeProcess와 task를 병행하여 사용해야 성능상의 문제점을 줄일 수 있을 것이다.

3) 응용 SW 분산통제 기능 구현 가능성 검토 결과

Table 4는 운영체제 별로 task와 process의 특성을 비교하여 응용 SW 분산통제 기능의 구현 가능성을 검토한 결과이다. VxWorks의 RealTimeProcess는 성능이 현저히 낮기 때문에 이를 단독으로 사용하여 응용 SW 분산통제 기능을 구현하기는 어렵다. VxWorks의 task는 같은 메모리를 공유하므로 하나의 task에서 소프트웨어 문제가 같은 컴퓨터 내의 다른 소프트웨어에 영향을 주며 전체 시스템의 동작을 멈추게 하는 경우도 있다. 반면에 리눅스의 프로세스는 프로세스 단위별로 독립된 메모리를 사용하여 하나의 프로세스에서 소프트웨어 문제가 같은 컴퓨터 내의 다른 소프트웨어에 영향을 미치지 않고 문제가 되는 프로세스만 비정상 종료 또는 오동작을 한다. 따라서, 소프트웨어 단위의 세밀한 분산 통제 및 장애 복구 기능을 구현하려면 리눅스의 프로세스 기반으로 사용해야 한다.

Table 4. Comparison of task/process features

	VxWorks 6.x		WRLinux
	Task	RealTimeProcess	Process
생성방법	taskSpawn API 사용	rtpSpawn API 사용	fork API 사용
생성시간 (1,000개)	0.017초	1.34초	0.02초
종료방법	taskDelete API 사용	rtpDelete API 사용	KILL 사용
종료시간 (1,000개)	0.01초	측정불가	0.01초
Memory Operation 시간	1.26초	12.084초	1.53초
Context Switching 시간	7.01초	15.446초	9.98초
종료 시 감지방범	task hook library를 이용하여 감지	종료 시 발생한 시그널을 통한 감지	종료 시 발생한 시그널을 통한 감지

## 5. 결론 및 향후 연구

분산 실시간 내장형 시스템의 특징을 지는 군사용 무기체계는 최근 상용 기술의 급속한 발전에 따라 민간 정보 기술(Information Technology)을 최대한 수용하고, 과거 개별 무기체계 임무 중심의 소규모 폐쇄형 시스템에서 다양한 종류의 품질 특성(QoS : Quality of Service)을 보이는 개별 무기체계들을 네트워크를 통해 통합 운용하는 대규모의 개방형 시스템 형태로 변화하는 네트워크 중심의 패러다임으로 전환되고 있다. 함정용 전투체계 역시, 과거 센서와 무장을 연결하여 사격통제를 수행하는 임무 중심의 시스템에서 다수의 센서, 무장과 같은 무기체계와 각종 통신장비 및 지원장비와 같은 비무기체계들을 통합운용하는 네트워크 기반의 체계로 발전하고 있다.

본 논문에서는 네트워크 기반의 전장 환경으로의 무기체계 패러다임이 전환되는 상황에 맞추어, 그동안 실시간 성능 위주로 개발되어온 함정용 전투체계가 향후 급속하게 성장하는 민수용 기술을 보다 적극적으로 수용할 수 있는 구조로 변환될 수 있도록 운영체제 측면에서의 개선을 제안하고, 상용 운영체제를 함정용 전투체계에 적용하기에 앞서 짚어 보아야 할 기술적/비기술적 성능에 대한 고려 요소들을 제시하였다.

본 논문에서는 Windriver사의 실시간 운영체제인 VxWorks와 RT Linux의 기본 및 확장 성능시험을 통하여 성능을 측정하였다. VxWorks task는 모든 시험에서 뛰어난 성능을 보였으나 소프트웨어 단위의 분산 통제 및 장애 복구기능을 구현할 시에는 제약사항이 있는 것으로 파악되었다. VxWorks RealTimeProcess는 프로세스 단위로 동작되어 task가 가지는 단점을 극복하였으나 프로세스 관리 측면의 성능상에 한계가 존재함을 확인하였다. 리눅스의 프로세스는 context switching, 파일 쓰기에 있어서 VxWorks task에 비해서 다소 성능이 떨어지나, 전반적으로 우수한 성능을 보였으며 전투체계와 같이 안정성과 실시간성이 요구되는 대형 복합 시스템에 적용 가능함을 확인하였다. 또한, 리눅스는 독립된 메모리를 가지는 프로세스 기반이어서 소프트웨어 단위의 분산 통제 및 장애 복구 기능을 구현하기에 용이한 장점을 가지고 있다.

향후 연구로는 각 평가 요소들에 대한 세부적인 평가 방법론 및 평가 환경을 개발하고, 이를 통해 상용 운영체제들의 능력을 평가하는 것과, 기존의 VxWorks 운영체제 기반에서 동작하는 전투체계 단위 모듈들을 리눅스 운영체제 기반으로 이식하여, 리눅스 운영체제 환경에서의 전투체계 전술 응용 SW들의 실시간 성능을 분석하는 것이다.

## References

- [1] 정용환, “대형 실시간 분산시스템을 위한 하이브리드 경로탐색기법에 관한 연구”, 제1권 제5호, 군사과학기술학회 2008.
- [2] Capt. Thomas J. Strei, “Overview of Naval Open Systems-The Plans, The Approach, The Promise”, PEO IWS.
- [3] NSWCDD, “Open Architecture(OA) Computing Environment Design Guidance”, 2004.
- [4] OMG, Data Distribution Service for Real-Time Systems Version 1.2, 2007.
- [5] Douglas C. Schmidt, “Operating System Performance in Support of Real-time Middleware”, 7th IEEE International Workshop on Object-Oriented Real-Time Dependable Systems(WORDS '02), Jan. 7~9, 2002.
- [6] Benjamin Ip., “Performance Analysis of VxWorks and RTLinux”, 2001.
- [7] WindRiver Systems Inc, “Wind River Workbench User's Guide, 3.0(VxWorks Version)”, 2007.
- [8] WindRiver Systems Inc, “VxWorks Application Programmer's Guide, 6.6”, 2007.
- [9] WindRiver Systems Inc, “Wind River Linux GETTING STARTED 3.0”, 2009.
- [10] WindRiver Systems Inc, “Wind River Linux USER'S GUIDE 3.0”, 2009.
- [11] 고순주 외, “함정전투체계의 해외 기술동향 및 국내 발전추세에 대한 고찰”, 한국방위산업학회 제16권 제2호, 2009년 12월.