MAX(Management and Administration of spaX)의 설계 및 구현

오 봉 진† · 하 영 국† · 김 도 형† · 김 채 규††

요 약

본 논문에서는 MAX(Management and Administration of spaX)의 설계와 구현에 대하여 기술한다. MAX는 주전산기 IV 과제로 개발된 SPAX(Scalable Parallel Architecture computer based on X-bar network)용 시스템 관리 도구이다. 시스템 관리자와 사용자는 GUI(Graphic User Interface) 형태의 사용자 인터페이스를 갖는 MAX를 이용하여 시스템 관리 작업을 쉽게 수행할 수 있다. MAX는 SPAX의 운영체제에서 제공하는 시스템 관리 명령어의 기능들을 대부분 포함하고 있으며, 각 기능들을 10개의 서브 도구로 구성하여 제공하고 있다.

Design and Implementation of MAX(Management and Administration of spaX)

Bong-Jin O† · Young-Gook Ha† · Do-Hyung Kim† · Chae-Kyu Kim††

ABSTRACT

This paper describes the design and implementation of MAX(Management and Administration of spaX). MAX is the system management tool for SPAX(Scalable Parallel Architecture computer based on X-bar network) developed by highly parallel computer IV project. System manager and users can perform system management job easily using MAX which has GUI(Graphical User Interface). MAX contains most of all functions of system management-related commands provided by operating system of SPAX and consists of 10 sub tools.

1. 서 론

1950년대 컴퓨터가 최초로 개발된 이후로 하드웨어 성능의 발전이 급속히 이루어져 초기의 대형 컴퓨터보다 뛰어난 성능을 갖는 컴퓨터가 개인용 컴퓨터의 위치에 자리잡고 있는 상황이 되었고, 은행이나 공공기관에서 사용하기 위한 목적으로 개발되는 초대형 컴퓨터들은 하드웨어 성능이나 지원되는 소프트웨어의 수준을 보면 가히 놀라울 정도이다. 컴퓨터 성능의 변화와 더불어 수행 환경의 변화도 주목할 필요가 있다[8].

초기의 컴퓨터는 독립된 하나의 개체로서 동작하였으나 점차적으로 네트워크라는 연결 매체를 이용하여 서로간의 정보 교환이 자유로워지게 됨으로써 이기종의 많은 컴퓨터로 이루어진 거대한 네트워크 환경의 한 일원으로써 동작하게 되었다. 이러한 컴퓨터 환경의 변화는 컴퓨터 분야에 시스템 관리라는 중요한 요소를 대두시켰다. 독립된 형태의 작업 환경이나 제한된 네트워크 환경에서 작업하던 시대까지는 하드웨어

† 정 회 원 : 한국전자통신연구원 컴퓨터·소프트웨어 기술연구소 연구원
†† 정 회 원 : 한국전자통신연구원 컴퓨터·소프트웨어 기술연구소 책임연구원
논문접수 : 1998년 8월 29일, 심사완료 : 1999년 2월 19일

자체의 구조도 단순하였으며, 단일 밴더(vendor)의 시스템을 이용하여 네트워크 환경을 구성하는 경우가 많아 시스템 관리가 용이하였고, 관리 기능의 항목도 하드웨어 자원을 효율적으로 관리하기 위한 제한된 분야에 국한되어 있었다. 하지만 최근의 분산 환경이나 웹(Web) 환경과 같은 대규모 네트워크 작업 환경에서는 다양한 종류의 컴퓨터가 연결되어 있고, 각 컴퓨터를 위한 운영체제 역시 모두 다르며 지원되는 응용소프트웨어의 수도 많고 종류도 다양하다. 따라서 시스템 관리자의 부담이 증대되고, 시스템과 여러 분야에 대한 고도의 전문 지식이 요구된다. 최근에는 시스템 관리를 한 명의 관리자가 총괄하여 수행하지 않고, 시스템 관리를 여러 분야로 나누어 각 분야별로 관리자를 두는 형태로 바뀌는 추세이다. 이러한 컴퓨터의 성능과 구조의 변화 그리고 작업 환경의 변화로 중요한 이슈로 떠오른 시스템 관리를 위한 연구가 활발히 진행되어 왔고, 시스템 관리와 관련된 많은 소프트웨어가 개발되었다[8].

시스템 관리 도구는 시스템 관리자와 운영자들이 효율적으로 시스템을 관리할 수 있게 하고, 일반 사용자들이 시스템을 좀 더 쉽고, 편리하게 사용할 수 있도록 한다. 개인용 컴퓨터로부터 대형 컴퓨터에 이르기까지 모든 종류의 컴퓨터에는 시스템 관리를 위한 기능이 필수적으로 제공되고 있으며, 특히 대형 컴퓨터 시장에서는 시스템 평가 요소로써 시스템 관리 도구의 기능을 중요하게 생각하고 있다[8].

시스템 관리 도구의 제공은 두 가지 형태로 이루어지고 있는데, 첫 번째 방법은 하드웨어 제조업체에서 운영체제 기능의 일부로써 제공하는 경우로, HP-UX, DEC UNIX, Solaris, UnixWare 등에서 그 예를 찾아볼 수 있다. 이런 종류의 관리 도구들은 운영체제를 구성하는 여러 기능 중 한 부분으로 제공되는 것이어서 관리 기능의 종류가 다양하지 못하며 기능도 다소 미약하다. 일반적으로 중소형 컴퓨터 환경에서 간단한 관리 기능을 지원하기 위해 많이 선택되는 형태이다[8].

두 번째의 방법은 특정 하드웨어나 운영체제와 독립적으로 제작되어 제공되는 경우로, DEC의 POLYCENTER, Novell의 Netware, CA-UNICENTER 등이 그 예이다. 시스템 관리의 목적으로 제 3의 업체(Third-party)에서 제공하는 형태로, 특정 시스템을 위한 것이 아니라 몇 가지 플랫폼에서 동작할 수 있도록 구현된다. 비교적 다양하고 고성능의 관리 기능을 제공하는 것부터 네트워크나 백업 혹은 성능 관리 등 특정 분야를 중점적으로 다루는 것까지 개발 회사에 따라 다양한 제품이 존재한다[8]. 이런 제품들은 비교적 여러 플랫폼을 고려하여 작성되므로 하나의 특정 시스템이 갖는 구조적 특성을 제대로 반영하지 못하는 단점이 있다.

MAX는 첫 번째 형태의 관리 도구이며, 각 관리 기능을 대형 컴퓨터에 맞도록 강화시켜 구현하였다. 기존의 주전산기를 위한 운영체제에는 특별한 관리 도구가 제공되지 않았고, 시스템 관리 명령어 수준에서 시스템 관리자가 관리에 필요한 작업을 수행해야 했다. 따라서 시스템 관리자나 운용자가 시스템 관리에 대한 전문적인 지식을 갖고 있어야 했고, 관리 절차도 매우 많은 시간을 필요로 하였다. 따라서 한국전자통신연구원에서는 주전산기 IV 과제 수행의 일환으로 GUI 형태의 사용자 인터페이스를 갖는 편리하고 안정적인 시스템 관리 도구 개발의 필요성을 느끼고 MAX를 개발하게 되었다. MAX는 UnixWare 2.0에서 제공되는 시스템 관리 도구를 기반으로 기존의 기능을 SPAX의 다중 노드 구조에 맞도록 수정하고, Chorus 표준 관리 기능 등 새로운 관리 기능을 확장하여 SPAX를 위해 최적화되어 개발된 시스템 관리 도구이다.

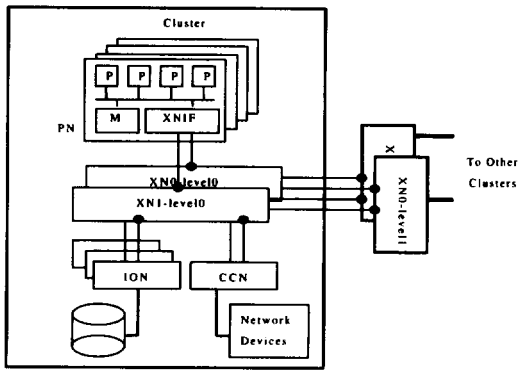
본 논문의 구성은 다음과 같다. 2장에서는 MAX가 구현되는 하드웨어, 소프트웨어 환경에 대해서 알아보고, 3장에서는 MAX의 구조와 구현에 대해 알아본다. 그리고 4장에서는 실제로 수행되는 예제를 살펴보고, 마지막으로 5장에서 결론 및 향후 과제에 대해서 기술한다.

2. 구현 환경

MAX가 최종적으로 동작될 플랫폼은 SPAX 시스템이다. SPAX는 클러스터 기반의 멀티미디어 컴퓨터로써 마이크로 커널 기반의 운영체제인 MISIX(MIcoro kernel based Single system Image uniX)에 의해 관리되어 진다. MAX는 GUI 인터페이스를 제공하기 위해 Motif1.2R과 X11R5의 라이브러리를 이용하였다.

2.1 SPAX 구조

SPAX의 구조는 (그림 1)과 같이 다중 처리기 노드들이 고속 내부 네트워크에 연결되어 클러스터를 이루는 2 계층 구조를 갖는다.



(그림 1) SPAX 구조

SPAX에는 최대 16개의 클러스터가 구성될 수 있으며, 각 클러스터에는 4개의 펜티엄 프로(P6) 칩과 1 GB의 지역 메모리를 갖는 노드가 8개까지 구성될 수 있다. 각 노드는 기본적으로 동일한 성능을 갖고 있으나 그 목적에 따라 계산 노드(PN), 입출력 노드(ION), 통신 접속 노드(CCN)로 구별되며, 하나의 클러스터에 4개의 계산 노드와 3개의 입출력 노드, 1개의 통신 접속 노드로 구성되는 것이 일반적이다. 각 클러스터들은 Xcent-Net으로 이중화 연결(클러스터 내의 노드들도 Xcent-Net으로 연결됨)되며, 이를 위해 각 노드에는 XNIF (Xcent-Net InterFace)가 설치되었다[9,13,15].

2.2 MISIX 구조

SPAX의 운영체제인 MISIX는 Chorus Kernel v3 r5를 기본 커널로 채택하고, 그 위에 UnixWare 2.0의 기능을 이식한 것이다. Chorus 커널은 SPAX 부팅시에 각 노드에 동작되고 메모리 관리, 인터럽트 처리, 프로세스와 쓰레드 관리와 같은 핵심적인 기능들만 처리한다. 운영체제가 갖추어야 할 다른 기능들은 서버 모듈로 나누어 구현되어 있으며, 각 노드별로 부팅될 때 노드의 역할에 따라 필요한 서버들이 메모리 상에 상주하여 동작하게 된다[10,15]. <표 1>에 계산 노드, 입출력 노드, 통신 접속 노드별로 상주되는 서버의 종류와 기능이 나타나 있다.

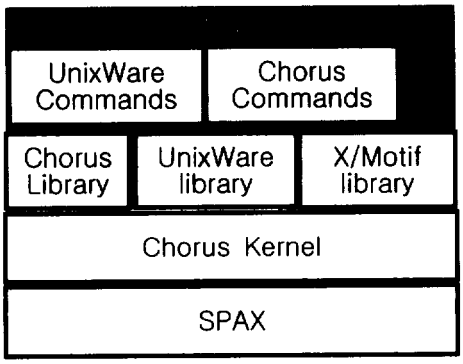
2.3 MAX의 수행 환경

MAX가 수행되는 환경은 (그림 2)와 같다. SPAX 시스템에 Chorus v3 r5 커널을 마이크로 커널로써 탑재하고, Chorus 표준 라이브러리, SPAX 시스템에 적합하게 변경된 UnixWare 시스템 라이브러리 그리고 X 라이브러리와 Motif 라이브러리를 올린다. 라이브러리 계층 위에 명령어들이 설치되는데 Chorus 표준으로 정의된 것들과 SPAX에 맞도록 수정된 UnixWare 명령어들이 올려지게 된다. 각종 서버들도 명령어 군에 속하며, MAX는 이 계층 위에 탑재되어 수행된다. MAX는 명령어나 라이브러리를 이용하여 관리 기능을 위한

<표 1> PN, ION, CCN을 위한 서버

Node Type	Server Name	Operation
PN	PM(Process manager)	프로세스 관리 및 시스템 호출
	UM(UID Manager)	사용자 자원 관리
	FM(File Manager)	파일 시스템 관리
	PNM(Path Name Manager)	파일 경로 이름 분석
	CM(Coherency Manager)	동시에 오픈된 파일의 일관성 유지
	DLD(DeadLock Detector)	파일 잠금(locking)시 교착상태 방지
	DPS(Device Port Server)	디바이스 이름에 포트 맵핑
	PIPE(Pipe manager)	유닉스의 파이프 관리
	STM(Streams Manager)	네트워크, tty 등의 스트림 디바이스 관리
	KM(Key Manager)	SVR4 IPC 기능 제공
ION	DCM(Disk Cache Manager)	디스크 캐쉬 관리
	FBM(Fast Back-up Manager)	고속 백업 관리
	BIOM(Block I/O Manager)	블록 입출력 장치 관리
	BCONM(Back-up Console Manager)	백업 콘솔 관리
CCN	CONM(Console Manager)	콘솔 장치 관리
	CFM(Configuration Manager)	시스템 형상 정보 관리
	BM(Boot Manager)	시스템 부트 관리
	Other manager	기타 노드에 연결된 통신 장치 관리

절차를 처리하거나, 관리에 필요한 정보를 담은 시스템 파일을 참조하고, 변경함으로써 관리자의 요구를 처리한다. 그리고 클라이언트는 MAX의 기능을 수행하기 위해 X 서버가 동작하는 시스템에서 작업을 해야 한다.



(그림 2) MAX 수행 환경

3. MAX의 구조와 구현

3.1 MAX의 구조

MAX는 모두 10개의 서브 도구로 이루어졌다. 이는 MAX가 다양한 관리 기능을 지원하기 위해 MISIX에서 제공하는 시스템 관리 명령어를 이용하여 수행할 수 있는 관리 기능을 대부분 포함하므로 시스템 관리자나 일반 사용자가 원하는 관리 기능을 쉽게 찾아 수행할 수 있도록 각 기능을 특징별로 분류하여 서브 도구를 구성한 것이다. 각 서브 도구별로 구조와 기능은 다음과 같다.

3.1.1 시스템 구성 관리 도구

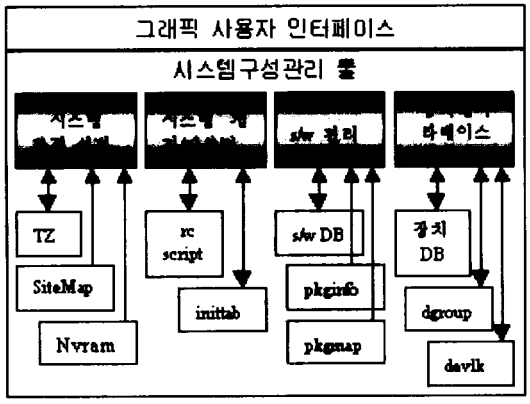
(그림 3)에 시스템 구성 관리 도구에 포함되는 기능의 종류와 필요한 시스템 정보의 종류가 나타나 있다. 시스템 구성 관리 도구는 SPAX 시스템이 일반 사용자에게 개방되기 이전에 시스템을 사용할 수 있도록 초기 작업 환경을 설정하거나, 시스템을 관리하는 동안에 발생하는 상황에 따라 하드웨어나 소프트웨어 자원을 관리할 필요가 있을 때 사용된다.

주요 기능을 간단히 기술하면 우선 시스템 환경 설정과 관련된 기능으로 날짜 및 타임존 설정, 운영체제 정보나 시스템의 호스트 명과 같은 시스템 정보 설정 기능, 클러스터 구성 정보 및 메모리와 주변 장치의 구성 정보를 상세하게 보여주는 것이다.

두 번째로 노드 단위로 부팅하거나 정지시키는 기능을 제공하는데, 부팅할 노드에서 동작하는 프로세스를 다른 노드로 이동시키는 기능도 포함한다.

세 번째로 소프트웨어 관리 기능에는 소프트웨어의 업그레이드나 새로운 소프트웨어 설치, 필요 없는 소프트웨어의 제거와 설치된 소프트웨어의 이상 유무를 검사하는 기능이 포함된다. MAX의 소프트웨어 관리 기능을 지원 받기 위해 각 소프트웨어는 SVR4의 소프트웨어 표준을 따르고, MAX나 MISIX에서 제공하는 설치 명령어를 통해 설치되어야 한다.

마지막으로 장치 데이터베이스 관리 기능에는 MISIX가 SPAX 시스템을 부팅하는 동안 시스템에 구성된 장치를 검색하여 생성한 장치 데이터베이스의 정보를 관리하는 기능으로 새로운 장치를 추가하거나 기존의 장치를 제거하는 경우, 시스템을 재부팅하지 않고 장치 데이터베이스의 내용을 변경할 수 있게 한다. 더불어 시스템 관리자의 관리 작업을 용이하게 하기 위한 장치 그룹과 장치 예약 기능을 제공한다. 장치 그룹은 비슷한 종류의 장치를 그룹으로 묶어 하나의 관리 기능을 장치 그룹에 속한 여러 장치에 한번에 명령을 내릴 수 있도록 하며, 장치 예약 기능은 특정 관리자가 독점적으로 관리할 수 있도록 권한을 부여한다.



(그림 3) 시스템 구성 관리 도구 구조

3.1.2 파일 시스템 관리 도구

파일 시스템 관리 도구는 파일 시스템 생성, 파일 시스템 검사, 파일 시스템 마운트(mount) 및 언마운트

(umount), 파일 관리 그리고 파일 시스템 유틸리티 기능으로 구성된다.

파일 시스템 생성 기능은 주어진 하드 디스크 파티션에 지정된 타입의 새로운 파일 시스템을 생성하며, 파일 시스템 마운트 및 언마운트 기능은 생성된 파일 시스템을 사용하기 위해 논리적 디렉토리 구조의 어느 한 곳에 파일 시스템을 연결(link)하거나 단절(unlink)시키는 기능을 처리한다. 파일 시스템 검사는 생성된 파일 시스템 중 마운트되지 않은 파일 시스템을 대상으로 파일 시스템 종류에 맞게 구조가 유지되고 있는지 검사한다.

파일 관리는 파일 시스템에 저장된 파일을 대상으로 복사, 삭제, 이동, 속성 변경 등의 기능을 수행한다. 각 기능은 파일 단위나 디렉토리 단위로 적용 가능하고, 와일드 카드(wild card)를 이용하여 필터링 기능을 사용할 수 있으며, 속성 변경의 경우는 하위 디렉토리를 탐색하여 재귀적으로 수행되도록 선택할 수 있다.

그 외의 파일 시스템 관리 기능들은 파일 시스템 유틸리티에서 처리되는데, 파일 시스템의 종류를 알아보는 기능, 파일 시스템의 레이블 정보 관리, 파일 시스템간 자료 복사, 파일 시스템의 사용량 및 사용 가능 용량 분석, 파일 검색 그리고 시스템 정지나 파일 시스템의 언마운트 시 이 파일 시스템을 사용하는 프로세스를 제거하여 잘못된 수행 결과를 초래하지 않도록 하는 기능이 포함된다.

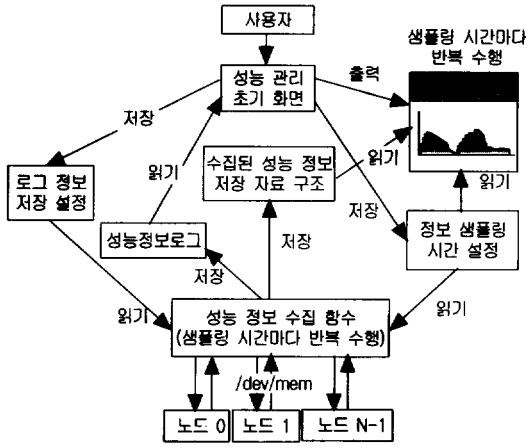
3.1.3 시스템 성능 관리 도구

시스템 성능 관리 도구는 SPAX의 구조적 특징을 반영하여 노드나 클러스터 단위의 성능 감시가 가능하고, 관리자가 쉽게 사용할 수 있는 인터페이스를 제공한다. 그리고, 관리자에게 다양한 감시 항목들을 제공하면서도, 이들을 적절히 구분하여 사용하기에 복잡하지 않도록 설계되었다.

(그림 4)는 성능 관리 도구의 기본 구조를 보여준다. 성능 관리 도구의 성능 정보 수집 함수는 성능 관리 도구를 수행하는 순간부터 종료될 때까지 설정된 데이터 샘플링 시간마다 반복 수행된다. 성능 데이터 수집 함수는 각 노드들로부터 정보를 수집하여 클러스터와 전체 시스템 정보를 계산하여 노드나 클러스터 혹은 전체 시스템의 정보를 선택적으로 보여줄 수 있도록 구성하였다.

정보 수집 부분은 현재 시스템 내에서 동작 중인 노드들의 정보를 이용하여, 각 노드의 '/dev/mem/'으로부터 필요한 정보를 수집, 가공하여 정의된 노드 자료구조에 저장한다. 저장된 노드별 정보와 클러스터 구성 정보를 이용하여 클러스터와 전체 시스템 정보를 계산하게 되는데, 클러스터 구성 정보는 어떤 노드가 어떤 클러스터에 속해있는지에 대한 정보를 제공한다. 자료구조에 저장된 정보를 갖고 사용자에 의해 선택되어진 감시 목록에 따라 성능 정보들이 그래프 형식의 출력 화면을 통해 관리자에게 보여진다.

성능 관리 도구에서 제공하는 관리 기능에는 CPU 사용율 감시, CPU 대기 queue 감시, 메모리 감시, 페이징 감시, 스와핑 감시, 디스크 감시 그리고 프린터 감시가 포함된다.



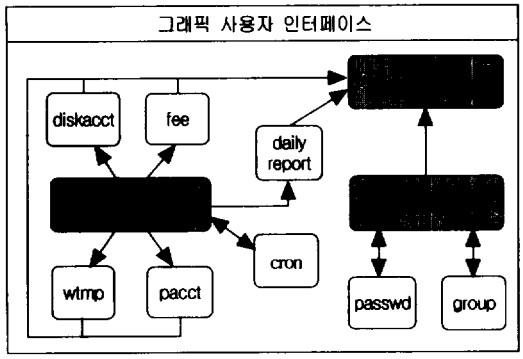
(그림 4) 성능 관리 도구의 기본 구조

3.1.4 계정 관리 도구

계정 관리 도구는 사용자의 계정에 대한 관리와 전반적인 시스템의 사용에 대한 정보의 수집, 이에 따른 사용료 부과 등을 그 목적으로 한다. (그림 5)에 계정 관리 도구의 구성과 관련 시스템 파일의 종류가 기술되어 있다. 계정 관리 도구는 사용자 계정 관리, 시스템 운용 정보 관리, 시스템 운용 정보 열람의 3가지 기능으로 구성된다.

사용자 계정 관리 기능은 시스템을 사용하는 모든 사용자에 대한 계정을 유지하고 관리하기 위하여 로그인 데이터베이스와 그룹 데이터베이스를 관리한다. 시스템 운용 정보 관리 기능은 시스템의 전반적인 사용에 대한 정보들을 수집하기 위한 초기 설정 및 회계

정보를 기초로 한 사용료 부과 등의 기능을 제공한다. 전체 시스템으로부터 수집되는 회계 정보는 각각 접속 회계, 프로세스 회계, 디스크 회계 및 사용료 데이터베이스에 저장이 되며, 이러한 자료들은 사용자에게 사용료를 부과하거나 시스템 성능 및 활용도에 관한 분석과 조정, 일별, 회기별 회계 보고서 작성을 통한 기간별 시스템 사용에 관한 정보 제공 등에 쓰일 수 있다. 시스템 운용 정보 열람 기능은 시스템 운용 정보 관리 기능을 통해 수집 및 생성되는 자료를 바탕으로 기간 및 다양한 항목별로 회계 내역을 검색할 수 있는 기능을 제공한다.



(그림 5) 계정 관리 도구의 구조

3.1.5 보안 관리 도구

보안 관리 도구는 DoD의 TCSEC에서 정의하고 있는 보안 등급 중 B1 등급 이상의 관리를 지원하기 위하여 접근 제어 관리 기능, 로그인 관리 기능, 패스워드 관리 기능, 암호화 관리 기능, 감사 관리 기능, 프리빌리지(Privilege) 관리 기능 및 보안 검사 기능으로 구성된다. (그림 6)에 보안 관리 도구의 구조가 나타나 있다.

로그인 관리 기능과 패스워드 관리 기능은 정당한 사용자만이 시스템에 접근을 하도록 허용하는 사용자 식별 및 인증 기능을 관리한다.

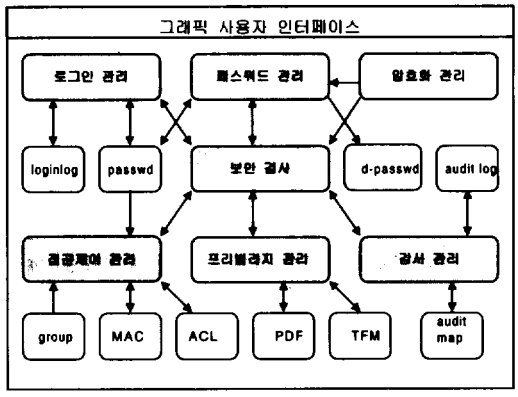
암호화 관리 기능은 DES(Data Encryption Standard) 및 IDEA(International Data Encryption Algorithm) 암호화 방법을 이용하여 시스템 제공 패스워드를 위한 암호화 키를 생성하거나, 파일을 암호화하는 기능을 제공한다.

감사 관리 기능은 시스템 보안에 영향을 줄 수 있는 프로세스나 사용자의 행동을 추적하고 감시할 수 있는 기능을 제공한다.

접근 제어 관리 기능은 DAC(Discretionary Access Control) 기능을 관리하기 위해 허가 비트 및 ACL (Access Control List) 데이터베이스를 이용하며, MAC (Mandatory Access Control) 기능을 위하여 보안 레이블(label)을 관리한다.

프리빌리지 관리 기능은 프리빌리지 및 롤(Role)의 관리를 위하여 PDF(Privilege Data File)와 TFM(Trusted Facility Management) 데이터베이스를 이용한다. TFM은 LPM(Least Privilege Module)을 통한 파일 기반의 권한 메카니즘을 이용하여 특정한 관리에 필요한 최소한의 권한들로 이루어지는 롤을 정의하게 해 주며, 관리자의 경험과 능력에 따라 정의된 롤을 부여할 수 있는 기능을 제공한다.

보안 검사 기능은 시스템 보안상의 허점을 검사하여 찾아 주고, 이를 보완하도록 도와주는 기능을 제공한다. 보안 검사 기능은 각 보안 관리 기능을 이용해 시스템 보안에 관련된 정보를 수집하여 이를 분석하고 관리자에게 결과를 알려 주며, 분석 결과에 따라 시스템 보안에 영향을 미칠 소지가 있는 부분에 대한 수정을 할 수 있도록 하여 이를 보완하게 해 준다.



(그림 6) 보안 관리 도구의 구조

3.1.6 단말기 관리 도구

단말기 관리 도구는 시스템에 연결되는 원격지(remote) 단말기나 지역(local) 단말기를 관리하는 것으로 포트 모니터 관리, 서비스 관리, 직렬 단말기 설정, 단말기 회선 설정 기능으로 구성된다.

모든 단말기는 포트 모니터에 의해 관리되며 포트 모니터 관리기능은 사용자의 단말기를 통한 시스템의 접근을 제어하는 것으로 사용자가 단말기를 통해 서비스를 이용할 수 있도록 시스템 부팅시 단말기를 초기화하고, 사용자의 접근을 감시하도록 포트 모니터를 설정한다.

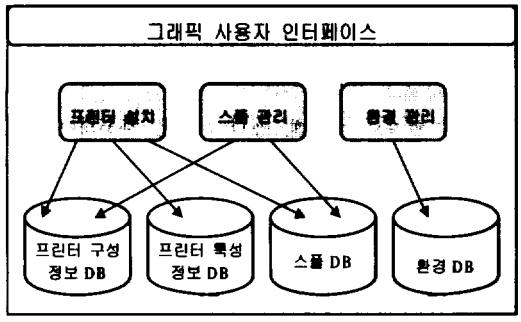
서비스 관리 기능은 단말기와 연관되어 제공되는 서비스를 사용하기 쉽게 설정하는 기능을 제공하는 것으로 특정 단말기에 새로운 서비스를 추가하거나 기존의 서비스를 동작하지 못하도록 설정할 수 있다.

단말기 설정 기능은 단말기를 시스템에 설치하거나 제거하고 단말기 파라미터를 변경하는 작업을 쉽게 할 수 있도록 해준다.

시스템이 부팅되어 초기화될 때 각 tty 포트를 위한 속도와 단말기 모드를 설정하기 위해 'ttydefs' 파일이 참조된다. 단말기 회선 설정 기능은 이 'ttydefs' 파일에 포함된 회선 정보를 관리한다.

3.1.7 프린터 관리 도구

프린터 관리 도구는 네트워크로 연결된 원격지 프린터를 관리하기 위한 도구로써 (그림 7)과 같이 프린터 설치, 스풀 관리, 환경 관리 기능으로 이루어져 있다.



(그림 7) 프린터 관리 도구 구조

프린터 설치 기능은 설치된 프린터를 제거하거나, 프린터의 특성을 변경하고자 할 경우, 프린터 구성 정보 데이터베이스와 특성 정보 데이터베이스를 참조, 변경하고 프린터 설치 및 제거 시에 자신의 스풀 데이터베이스를 생성, 제거한다.

스풀 관리 기능은 프린터 구성 정보 데이터베이스를 참조하여 프린터 큐를 제어하는 역할을 한다.

환경 관리 기능은 최적의 프린트 서비스를 제공하기 위해 프린팅 환경을 정의, 변경하는 것으로 프린팅 폼 관리, 필터 관리 그리고 폰트 관리를 수행한다.

3.1.8 백업 관리 도구

백업 관리 도구는 기본 백업, 복구 백업, 확장 백업 기능으로 구성된다. 기본 백업은 UnixWare 2.0에서 제공되는 기본 백업과 확장 백업의 기능들을 바탕으로 구현한 것으로 백업 시작, 백업 히스토리 관리, 백업 필요성의 통지, 백업 작업 요구의 응답, 자동 백업 설정, 백업 테이블 조작, 제외 목록 조작, 백업 상태 조사 기능을 처리한다.

복구 작업 기능은 백업된 자료를 다시 파일 시스템에 복구하는 기능들로 백업 자료 복구, 복구 작업 요구의 응답, 작업 요원 지정, 복구 작업 상태 열람 등이 지원된다.

확장 백업 기능은 기본 백업 외에 관리자의 편의나 자원 절약, 혹은 시스템의 이용률 증가의 측면에서 필요한 기능들로 다중 호스트 환경에서의 백업, 단일 사용자 모드에서의 자동 백업, 압축 모드 백업, 고속 백업 기능 등이 있다.

3.1.9 네트워크 관리 도구

네트워크 관리 도구는 SPAX 시스템을 TCP/IP 네트워크에 접속시키기 위해서 필요한 네트워크의 운영 환경을 설정한다. 그리고 TCP/IP 네트워크를 사용하다가 생길 수 있는 문제의 모니터링, 문제가 생겼을 때의 복구 지원 등을 포함하는 네트워크의 효율적인 이용을 위해 필요한 일들을 처리한다. 네트워크 관리 도구는 크게 기본 환경 설정, IP 주소 할당, 네트워크 상태 및 진단 기능을 포함한다.

기본 환경 설정 기능은 네트워크 관련 파일 편집, 경로 설정, 서브네트 설정 등을 처리하고, IP 주소 자동 할당 기능은 SPAX 시스템에서 IP 주소를 요청하는 클라이언트에게 자동으로 임의의 주소를 할당하여 클라이언트가 네트워크에 접근이 가능하도록 해주는 기능이다. 네트워크 상태 및 진단 기능은 네트워크 상태 모니터링 기능과 복구 지원 기능의 두 가지 세부 기능으로 구성되며, 시스템의 네트워크 설정 상태를 오류 없이 보고하는 것과 원격지로의 패킷 전송 경로 추적기능이 오류 없이 수행되는지를 검증한다.

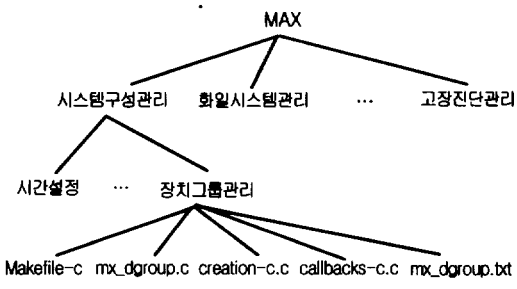
3.1.10 고장 진단 관리 도구

SPAX는 하드웨어 및 커널에 대한 진단 기능과 고

장 감내 기능을 제공하고 있다. 고장 진단 관리 도구는 진단 커널에 의해 로그 큐에 저장된 사건 정보들을 수집하고 이를 분석하여 화면에 보여줌으로써 관리자가 시스템의 고장 유무를 알 수 있도록 한다. 출력되는 시스템 메시지는 키워드에 의해 메시지를 검색하거나 특정 시스템 메시지가 발생하는지의 여부를 감시할 수 있도록 하여 특정 사건의 발생 유무를 알 수 있게 한다.

3.2 MAX의 구현

MAX의 각 기능은 ANSI-C로 작성이 되었고, GUI 인터페이스를 구성하기 위해 Builder Xcessary 3.5 버전을 사용하였다. 전체 프로그램 소스의 구조는 (그림 8)과 같이 구성된다. MAX 디렉토리 밑에 각 서브 도구별로 디렉토리가 존재하고 서브 도구별 디렉토리 밑에 각 기능의 디렉토리가 존재하며, 그 밑에 각 기능을 구현하기 위해 필요한 5개의 파일인 'mx_*기능*-c.c', 'creation-c.c', 'callbacks-c.c', 'Makefile-c' 그리고 'mx_*기능*.txt'들이 위치한다.



(그림 8) MAX 프로그램 구조

'mx_*기능*.c'는 각 기능의 주 프로그램으로 'main.c()' 함수를 포함하며, 작업을 위해 초기 화면을 생성하도록 X 서버에 요청하고, X 서버가 사용자의 입력에 따라 발생되는 이벤트를 받아서 처리하도록 X 이벤트 루프를 발생시킨다.

creation-c.c에는 각 기능이 제공하는 사용자 인터페이스를 생성하기 위한 함수들이 포함되어 있다. 즉, 작업을 위한 초기 화면과 사용자의 입력에 따라 발생되는 사건을 처리하는 과정에 생성되는 화면들을 위한 함수들로 구성된다. 이런 함수들은 개발 도구인 Builder Xcessory v 3.5의 화면 제작 기능을 이용하여 1차로 화면을 구성하고, 코드 생성기를 통해 기계적으로 생성된 코드를 개발자가 수정 및 보완하여 완성하였다.

callbacks-c.c에는 사용자의 입력에 따라 발생되는 이벤트를 처리하는 함수들이 포함되는데, 실제 시스템 관리 기능에 해당하는 핵심적인 작업이 여기에 포함된 함수들에 의해서 수행된다. 시스템 관리 기능 처리는 세 가지 방법을 이용하여 이루어지는데, 첫 번째로 MISIX에서 제공하는 관리 명령어를 이용하여 처리하는 방법, 두 번째로 시스템 라이브러리 함수를 이용하는 방법 그리고 마지막으로 관리 기능과 관련된 시스템 파일이나 커널 메모리를 직접 접근하는 방법으로 나누어진다. 시스템 관리 명령어를 이용하는 방법은 MISIX가 제공하는 관리명령어를 인자와 함께 'system()'이나 'pipe()' 시스템 함수를 이용해 하나의 프로세스로 발생시켜 작업을 처리하는 방법이다. 그 예를 보면 다음과 같다.

```
system ("pkgadd -iv MAX < response-file > result-file");

if ('result-file' has error)
        display window for error message
else
        display window for success message
```

'system()' 함수는 관리 명령어가 수행 과정에서 사용할 응답 파일과 관리 명령어의 수행 결과 생성되는 오류 메시지나 결과 메시지를 포함하는 결과 파일을 추가적인 인자로 갖는데, 응답 파일은 사용되는 관리 명령어가 대화식 기능을 갖는 경우 사용자의 입력을 필요로 할 때가 존재한다. 이런 경우 자동으로 적절한 응답을 할 수 있도록 미리 응답 파일을 만들어 놓고, 관리 명령어가 그 내용을 입력으로 받게 하는 것이다. 결과 파일은 관리 명령어가 정상적으로 종료되었는지, 오류를 발생하고 종료를 했는지를 알아내기 위해 사용되는 파일로써 관리 명령어의 수행 과정에서 출력하는 모든 메시지를 포함한다. 이러한 개발 방법은 수행할 관리 기능을 위한 절차가 관리 명령어에서 처리되므로 간단하게 필요한 기능을 구현할 수 있어 개발자의 부담이 적고, 빠른 시간에 개발을 끝낼 수 있다는 장점이 있으나 사용되는 관리 명령어가 수행되는 중간에 어떠한 간섭도 할 수 없을 뿐만 아니라 대화식 작업의 경우, 모든 경우에 대하여 적절한 입력을 넣어주기가 불가능하여 하나의 경우만을 설정하고 그 이외의 모든 경우에 대해서는 오류를 발생하도록 구현해야 하는 단

점이 있었다.

두 번째 방법인 시스템 라이브러리를 이용한 방법은 다음의 예와 같이 관리 기능의 수행을 위한 절차를 시스템 라이브러리 함수들을 이용하여 직접 작성하는 것이다.

```
add_new_devicecb(widget w, XtPoint cb,
XtPointer client) {
    devlist  = getdev (DDB, DEV_ALL);
    infolist  = new device's attribute values;
    devname= new device's alias;

    if (devlist has device with the same name
    as devname) {
        display warning message;
        if (user order to replace old device with
        new device)
            if (remdevrec (devname, FALSE) ==
            NULL) {
                display error window;
                return;
                }
        else return;
        }
    if (adddevrec (devname, infolist, FALSE)
    == NULL)
        display error window;
    else
        display success window;

return;
}
```

관리 명령어를 이용한 방법에 비해 개발자가 해야 할 작업이 양이 많아지고 개발 기간이 길어진다는 단점은 있으나, 관리 명령어를 이용한 방법과 동일한 수준의 기능을 처리할 수 있으며, 에러의 처리나 사용자와의 대화식 수행과 같이 세밀한 기능 조작이 가능하다는 장점이 있다. MAX에서는 이러한 장점을 중요시 여기고, MISIX에서 필요한 함수를 제공하는 한 이 방법을 이용하여 개발하는 것을 기본으로 하였다.

마지막 방법인 시스템 파일을 이용한 방법은 수행할 기능이 시스템 파일을 조작하여 간단히 처리할 수 있는 경우나 각 노드의 구성 정보를 얻는 것과 같이 시스템 라이브러리나 관리 명령어에서 제공해주지 않는 기능을 위해 선택한 방법이다. 그러나 시스템 파일을 접근하는 것은 커널과의 동기화가 이루어지지 않음으로써 공유되는 시스템 파일의 무결성(Consistency)을 침해할 가능성이 있기 때문에 동기화가 필요 없는 경우에 한해서만 사용하였다.

Makefile-c는 각 기능을 컴파일하기 위한 절차를 갖고 있는 make 파일이며, mx_기능.txt는 사용자에게 각 기능의 사용법을 알려주기 위한 파일이다.

이렇게 구현된 각 기능은 MAX를 위해 제공되는 간단한 데스크탑에서 각 관리 기능을 나타내는 아이콘(icon)을 더블 클릭하여 수행하거나, 명령어 라인 상에서 직접 해당 명령어를 넣음으로써 수행 할 수 있다. MAX는 기본적으로 '/usr/MAX'나 환경 변수 'MAXDIR'가 정의하는 디렉토리 밑에 위치하게 되며, 실행 파일들은 'bin' 밑에 그리고 도움말 파일들은 'help' 디렉토리 밑에 위치하게 된다.

4. MAX 수행 예제

SPAX 시스템에는 윈도우 데스크탑 매니저가 제공되지 않기 때문에 MAX를 수행시키기 위한 간단한 데스크탑을 구현하였으며, (그림 9)에 데스크탑을 통해 나타나는 MAX의 초기화면이 나타나 있다. 초기 화면에는 10개의 하위 도구를 대표하는 아이콘들이 나타나고, 관리자나 사용자는 원하는 기능을 포함하는 관리 도구의 아이콘을 선택함으로써 하위 관리 도구를 위한 화면으로 이동할 수 있다.

(그림 9) MAX의 초기 화면

하위 관리 도구 화면의 구성은 MAX의 초기 화면과 같으며, 시스템 관리자는 각 관리 기능을 위해 표현된 아이콘들을 선택하여 원하는 작업을 수행할 수 있다.

MAX를 위한 데스크탑의 메뉴에는 'FILE', 'VIEW',

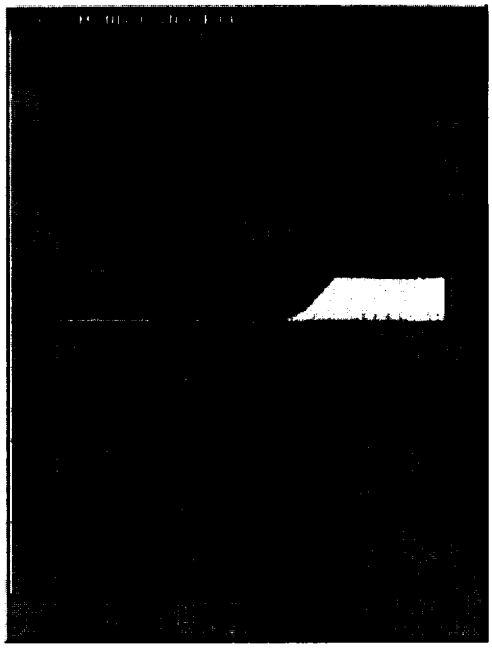
'OPTIONS', 'HELP' 등이 등록되어 있는데, 현재 선택된 디렉토리나 실행 파일을 더블 클릭 하지 않고 바로 수행하는 기능, 화면에 보여지는 파일들의 종류를 조정하는 기능, 더블 클릭 대신 싱글 클릭을 통해 선택할 수 있도록 하는 기능과 데스크탑의 사용법을 가르쳐주는 도움말 기능의 역할을 한다. MAX에서 제공되는 데스크탑은 기본적으로 MAX의 각 기능을 수행하기 위해 구현하였으나 일반 프로그램도 등록하여 수행할 수 있다.

(그림 10)은 시스템 구성 관리 도구 중 장치 데이터베이스 관리 기능의 수행 화면이다. SPAX 시스템에서는 노드나 디스크가 온라인 상태에서 추가되거나 제거가 가능하므로, 그런 상황이 발생했을 때 하드웨어 구성과 관련된 정보를 변경 시켜줘야 한다. 장치 데이터베이스는 부팅시 검색된 장치 정보가 수록되어 있으므로 온라인 상태에서 변경된 하드웨어 장치의 정보를 반영할 수 있는 기능이 제공되어야 한다.

(그림 10) 장치 데이터베이스 관리 화면

장치 데이터베이스 관리 기능의 초기 화면에는 현재 등록된 장치의 목록이 리스트로 보여지며, 선택된 장치를 제거하거나 속성 정보를 열람, 편집할 수 있다. 선택된 장치의 속성 편집은 새로운 작업 윈도우를 통해 수행되며, 새로운 속성을 추가하거나 기존의 속성을 제거, 갱신 할 수 있다. 새로운 장치를 추가하는 경우는 기존의 장치 중 비슷한 속성을 갖는 장치의 속성 정보를 이용하여 쉽게 등록할 수 있도록 하였다. 예를 들면, 새로운 디스크가 설치된 경우 기존의 디스크 장치인 'disk1'을 선택하고 '->' 버튼을 이용하여 속성 정보를 복사한다. 복사된 속성 정보 중 공통된 정보를 제외한 정보의 값을 수정하여 '<-' 버튼을 통해 장치 데이터베이스에 등록하도록 하는 것이다.

(그림 11)은 성능 관리 도구의 CPU 감시 기능 수행을 보여주는 화면이다. 성능 관리 도구에서는 감시할 목록을 초기 화면에서 선택하며, 한번에 하나의 대상 혹은 여러 대상을 선택하여 감시할 수 있다. 각각의 대상을 위해 독립된 화면이 생성되고 결과가 보여진다. 설정된 간격에 따라 반복적으로 샘플링된 결과는 각 노드별로 혹은 클러스터 및 시스템 전체에 대해 선택적으로 화면에서 볼 수 있으며, 결과는 시간의 흐름과 사용률을 항목으로 하는 그래프를 통해 보여짐으로 직관적으로 알 수 있다.



(그림 11) CPU 감시 화면

(그림 12)는 보안 관리 도구의 ACL 관리 기능의 수행 화면을 보여준다. DAC의 일부로서 파일의 ACL 목록을 추가, 변경, 삭제 및 열람할 수 있게 해준다. 허가 비트가 소유자, 그룹, 그 외의 사용자 단위로만 접근에 대한 제한을 할 수 있는데 비해 ACL은 개개의 특정 사용자에 대해서도 파일 및 디렉토리의 접근을 제한할 수가 있어서 허가 비트보다 강력한 보안 기능을 제공한다.

MAX를 구성하는 다른 기능들도 여기에서 보여진 기능들과 비슷한 인터페이스를 갖고 있다. 기본적으로

초기 화면에 전체적인 정보와 선택된 옵션을 개략적으로 보여주고, 추가적인 상세한 옵션 설정과 같이 수행에 필요한 환경 설정이나 정보 열람 등은 새로운 작업 화면을 생성하여 처리하는 계층적 화면 구조를 갖는다. 그러나 하나의 기능을 수행하기 위해 너무 많은 작업 화면을 생성하면 사용자에게 혼란을 줄 수 있으므로 대략 2~4개 이내의 작업 화면을 생성하도록 구현하였다. 또한 성능 관리 도구나 계정 관리 도구의 경우는 일정한 기간동안 샘플링된 데이터를 통계화 하여 보여주는 기능을 포함하는데, 사용자가 데이터의 통계 결과를 쉽게 파악할 수 있도록 그래프 형태의 화면을 제공하는 방법도 사용하였다.

(그림 12) ACL 관리 화면

5. 결 론

본 논문에서는 주전산기 IV 과제로 개발된 SPAX 시스템용 관리 도구인 MAX의 설계와 구현에 대해 기술하였다. MAX는 GUI 기반의 사용자 인터페이스를 갖는 관리 도구로써 시스템 관리자와 사용자가 손쉽게 관리 기능을 처리할 수 있도록 도와주며, UnixWare 2.0의 시스템 관리 도구의 기능을 기본으로 하여 SPAX 시스템에 맞도록 기능을 확장, 추가하였으며 Chorus 관리 명령어와 라이브러리의 기능을 일부 포함한다.

모든 기능들은 특징에 따라 10개의 서브 도구로 분류하였고, MAX 수행을 위해 제공되는 간단한 데스크탑 매니저를 통해 아이콘(Icon)을 선택하거나 명령어 라인 상에서 직접 명령어를 넣음으로써 수행할 수 있다.

앞으로의 보완점이라면 좀더 세련되고 직관적인 GUI의 제공으로 사용상의 편리함과 안정성을 더욱 확보하는 것이며, CDE(Common Desktop Manager)와 같은 표준에 맞도록 데스크탑 환경을 구성하여 사용자로 하여금 익숙한 형태의 도구로써 접근해 가는 것이다.

참 고 문 헌

[1] W. Richard Stevens, "Advanced Programming in the Unix Environment," Addison Wesley, 1992.

[2] "POLYCENTER Solution Guide," DEC., 1993. 1.

[3] "Using CA-UNICENTER," Computer Associates inc, March, 1993.

[4] "Solaris 2.x System Administration," Sun Microsystem Computer Corporation, 1993.

[5] "UNICOS System Administration TR-USA," Cray Research Inc, 1994.

[6] Douglas A. Young, "The X Window System programming and Applications with XT," Prentice Hall, 1994.

[7] "Unix System V Release 4.2 Multiprocessor System Administration," Novell, May., 1994.

[8] "고속 병렬 컴퓨터(주전산기 IV)의 시스템 관리 소프트웨어 개발에 관한 연구(I)," 한국전자통신연구소, 1995. 1.

[9] "고속 병렬 컴퓨터(SPAX) 하드웨어 서브시스템 설계서," 한국전자통신연구소, 1995. 4.

[10] "운영체제 서브시스템 설계서 version 1.0", 한국전자통신연구소, 1995. 6. 17.

[11] "Unixware 2.0 System Administrators Guide Manual Page," Novell co, 1995.

[12] "병렬 운영체제 MISIX API(Ver. 1.0)", 한국전자통신연구소, 1996. 5. 8.

[13] 정성인, "고속병렬컴퓨터의 메시지 전송 구조 및 성능 평가," 한국정보과학회 추계 학술 발표회, 1996. 10.

[14] "고속 병렬 컴퓨터(주전산기 IV)의 시스템 관리 소프트웨어 개발에 관한 연구(IV)", 한국전자통신연구원, Jan., 1998.

[15] 정성인, 이재경, 김해진, 임기욱, "고속병렬컴퓨터의 크로스바 네트웍 라우터 에뮬레이터 설계 및 구현," 한국정보처리학회 논문지, pp.488-502, 1998. 2.

오 봉 진

e-mail : bjoh@etri.re.kr

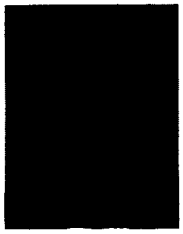
1993년 부산대학교 전자계산학과 졸업(이학사)

1995년 부산대학교 대학원 전자계산학과 졸업(이학석사)

1995년~1997년 시스템공학연구소 연구원

1998년~현재 한국전자통신 연구원 컴퓨터·소프트웨어 기술연구소 연구원

관심분야 : 병렬 파일시스템, 자바 embedded S/W, Home Network 등



하 영 국

e-mail : ygha@econos.etri.re.kr

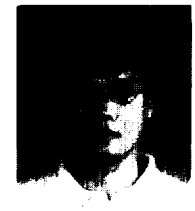
1993년 건국대학교 전자계산학과 졸업(공학사)

1995년 건국대학교 전자계산학과 대학원 졸업(공학석사)

1995년~1997년 시스템공학연구소 연구원

1998년~현재 한국전자통신 연구원 컴퓨터·소프트웨어 기술연구소 연구원

관심분야 : 시스템보안, 컴퓨터네트워크, 실시간 OS 등



김 도 형

e-mail : dhkim@etri.re.kr

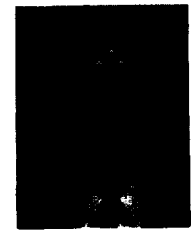
1993년 경북대학교 컴퓨터공학과 졸업(공학사)

1995년 포항공과대학 전자계산학과 졸업(이학석사)

1995년~1997년 시스템공학연구소 연구원

1998년~현재 한국전자통신 연구원 컴퓨터·소프트웨어 기술연구소 연구원

관심분야 : 결함포용, 병렬처리, 성능평가, 실시간 커널 등



김 채 규

e-mail : kyu@etri.re.kr

1978년 고려대학교 수학과 졸업(이학사)

1993년 호주 Univ. of Technology Sydney 전산학(석사)

1997년 호주 Univ. of Wollongong 전산학(박사)

1977년~1990년 KIST 전산개발센터 연구원, 시스템공학연구소 책임연구원

1998년~현재 한국전자통신연구원 컴퓨터·소프트웨어기술연구소 책임연구원

관심분야 : 데이터베이스, 병렬처리 등