DBpia

論文 94-19-8-20

클라이언트/서버 모델에 의한 메인프레임 부하 분산 사례연구

正會員 高 光 炳* 正會員 孔 勝 郁* 正會員 權 奇 睦* 正會員 康 昌 彦*

A Case Study of Mainframe Load Reduction Using The Client and Server Model

Kwang Byung Koh* Seung Wook Kong* Gi Mok Kweon*
Chang Eon Kang* *Regular Members*

본 논문은 1993년도 연세대학교 행정연구비 지원에 의하여 연구된 것임.

要 約

대학에서는 전산자원의 활용도를 높이기 위해 주전산기, 워크스테이션, 퍼스널컴퓨터등 다수의 전산자원을 LAN으로 연결하여 사용한다. 그러나 관리, 보안상의 이유로 대부분의 행정용 응용프로그램들은 주전산기에 집중되어 있어, 실제로 단기간에 전교학생이 수강신청을 하여야 하는 온라인 수강신청시스템과 같은 응용프로그램은 주전산기에 막대한 과부하를 초래하여 많은 문제점이 있다.

본 논문에서는 연구를 위하여 Y대학을 모델로하고 온라인 수강신청시스템을 분산처리 대상업무로 선정하여 모델 대학환경에 가장 적합한 분산처리 방법으로 IBM SNA의 LU6.2 링크를 통한 APPC를 제안하고, 온라인 수강등록 시스템을 클라이언트/서버 모델로 재설계하여, 주전산기는 화일의 입출력을 담당하는 화일서버로 워크스테이션은 처리를 담당하는 클라이언트로 구현한다. 적용실험을 통하여 분산처리 전보다 응답시간이 현저하게 향상되었음을 보인다.

ABSTRACT

In order to increase the utilization of the computing resources, universities connect a variety of computing resources such as mainframes, workstations, and personal computers via LAN. However, due to management and security reasons, most administrative applications are concentrated on mainframes which is the main cause of large work overload on mainframes for such applications as on-line course registration system where the entire student body must have access to the system during a short period of time.

* 연세대학교 전산원
Yonsei Information Systems Center
論文番號 : 94157
接受日字 : 1994年 6月 15日

www.dbpia.co.kr

In this study, using a university system as the model and choosing on-line course registration system as the targeted distributed computing. APPC through IBM SNA LU6. 2 link is proposed as the most appropriate means of distributed computing for the environment of the model university. In addition, the on-line course registration system is redesigned as client-server model where a mainframe serves as the file server responsible for file input and output and workstations becomes the client. Actual implementation and experiments have shown that the proposed distributed computing system yields a significant reduction in processing time.

I. 서 론

오늘날 퍼스날컴퓨터 및 워크스테이션이 마이크로프로세서의 급속한 발달로 인하여 성능은 날로 향상되고 있고 가격은 상대적으로 낮아짐에 따라, 기업은 물론 대학 및 연구소등 각종 기관에 확산되어 다양한 용도로 활용되고 있다. 따라서 주전산기에 의존하던 과거의 컴퓨팅 방식을 탈피, 다운사이징(downsizing)화하여 비용을 절감하는 동시에 다양한 서비스를 하고자 하는 경향이 뚜렷하다. 그러나 이와같은 환경하에서는 데이타베이스를 분산하여 관리해야 하는 어려움이 따르며, 정보보안의 문제가 대두되고, 정보관리가 용이하지 않아 자칫하면 다운사이징으로 인해 더 많은 유지보수 비용을 부담해야 하는 결과를 초래할 것이다. 그러므로 다운사이징은 계획단계에서 그 범위를 명확히 할 필요가 있으며, 무조건 주전산기를 배제할 것이 아니라 주전산기와 워크스테이션 각각이 가지는 특성을 최대한으로 살려 시스템을 구축하는 것이 바람직하다. 특별히 대형 데이타 베이스를 서어비스하는 조직에서는 주 전산기와 워크스테이션 간의 적절한 역할분담을 통하여 전산장비의 효율을 극대화하는 방안이 연구되어져야 한다.

그림 1은 연구를 위하여 모델로 선정한 Y대학의 네트워크 구성도이다. 주전산기는 IBM4381-R91 시스템으로 OS(Operating System)는 VM(Virtual Machine)을 사용하고 있으며, VM하에 학사, 재무, 인사 등 대학 행정업무를 위한 OS인 VSE/SP(Virtual Storage Extended System Package)와 도서정보 서비스를 위한 또하나의 VSE/SP를 설치하여 운영중이다. 교수 및 학생의 교육·연구지원은 주로 VAX 6000-420, 각종 워크스테이션, 다수의 PC들이 담당하고 있으나, 특정 통계 소프트웨어의 지원은 역시 IBM시스템이 담당하고 있다. 또한 IBM은 해외네트워크망인 BITNET(Because It's Time Network)의 전자우편 서버역할까지도 담당하고 있어 주전산기 의존도가 매우 높다.

현재 이 대학의 대표적인 네트워크 응용 소프트웨어로는 온라인 수강신청 시스템을 들 수 있다. 이는 네트워크상의 PC들이 TCP/IP 프로토콜을 이용하여 IBM시스템의 행정용 OS에 로그온하여 수강신청을 한다. 수강신청 시스템은 일시에 대단히 많은 수의 사용자를 서비스하여야 하므로, 수강신청 기간에는 IBM시스템의 부하가 극심하여 시스템 응답시간이 평소보다 약 10배 정도 늦다.

이와같은 문제의 해결방법으로는 응용프로그램을 다른 시스템으로 이식시키는 방법을 고려할 수 있으

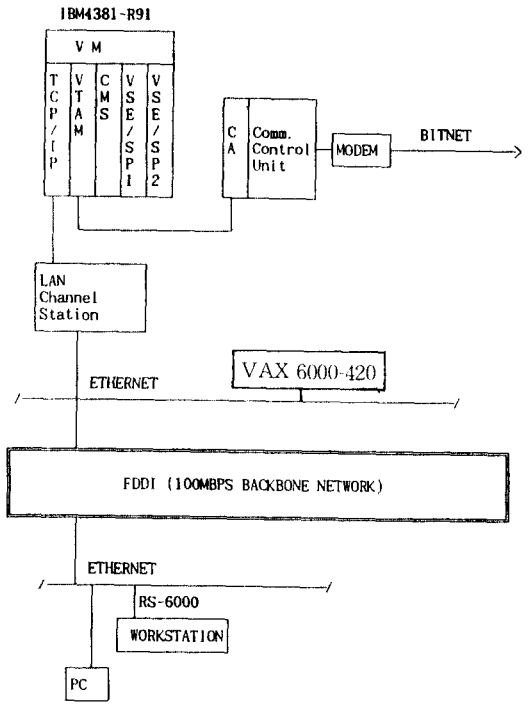


그림 1. Y대학의 네트워크 및 시스템 구성도

Fig 1. Network & system configuration of Y university.

www.dbpia.co.kr

나 수강신청 시스템이 여타의 학사행정 시스템에 연계되어 운영되므로 쉽게 타시스템으로의 이식을 선정할 수 없으며, 이 대학의 경우 마땅히 이식할 만한 시스템도 없다. 따라서 수강신청 시스템중 독립적으로 운영될 수 있는 부분을 네트워크상의 워크스테이션에서 구현하고 주전산기와 상호 협조체제를 유지하므로써, 주전산기의 부하를 경감시켜 전체적인 시스템의 성능을 향상시키는 방법을 모색하고 있다.

본 논문은 이러한 필요성에 의해, 클라이언트/서버 모델을 이용한 분산처리 시스템을 구축하여, 실제 업무에 적용해 봄으로써 그 효과를 확인하고자, 클라이언트/서버 구현방법을 조사하여 연구환경에 가장 적절한 방법을 제안하고, 수강등록시스템을 대상으로 제안한 방법을 적용하여 실제 업무에 적용하여 실험하고 결과를 분석하였다.

Ⅱ. 클라이언트/서버 모델과 프로토콜

그림 2는 일반적인 클라이언트/서버모델의 상태전이 그림이다. 개시상태(Initiation State)에서 클라이언트/서버 양측은 통신 경로를 설정하고 서비스에 필요한 자원을 확보하며, 정보교환에 필요한 규칙에 동의하고 프로그램을 가동한다. Begin Automic Action 상태에서 클라이언트/서버 양측은 트랜잭션들로부터 동시에 액세스되어져야 하는 자원의 locking에 동의한다. 이때 주로사용 되는 프로토콜이 two phase locking이다. 이 상태는 어플리케이션의 선택사항이어서 생략할 수도 있으며, 분산 데이타베이스 관리기에의해 자동처리될 수도 있다. 정보전달상태(Information Transfer State)에서 양측은 정보를 교환하게되므로 이 단계에서 정보교환 및 상태점검 명령등이 필요하게 된다. Synchronization & End Atomic 상태에서 commit processing, deadlock resolution 등 에러의 복구등을 수행하게되는데 two phase commit이 가장 널리 사용되는 프로토콜이다. 종료상태(Termination State)에서 할당된 자원을 되돌려주고 접속을 단절하고 프로그램을 종료한다. 이와같은 클라이언트/서버 모델은 여러 종류의 클라이언트/서버 시스템에 직접히 활용될 수 있다. 예를들어 비교적 단순한 시스템에 활용될 경우는 개시, 정보전달, 종료상태만으로 축한 것이지만 분산데이타베이스 관리기와 같이 복잡한 시스템에는 전 상태가 전부필요하게 된다.

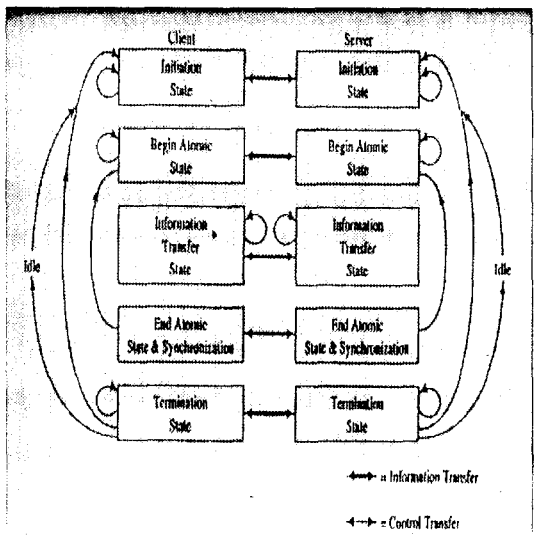


그림 2. 일반적인 클라이언트/서버 모델의 상태전이 다이어그램

Fig 2. The state transition diagram of a generic client server model

클라이언트/서버 시스템을 구현하기 위한 서비스로는 TCP/IP네트워크에서는 소켓, SNA네트워크는 APPC(Advanced Program to Program Communication), MAP(Manufacturing Automation Protocol)하에서는 MMS(Manufacturing Message Services)와 같이 네트워크 프로토콜에 의존적인 서비스들과 RPC(Remote Procedure calls)와 같이 네트워크에 독립적인 서비스로 나뉜다.

TCP/IP 소켓은 OSI모델의 4와 5번 Layer에 속하며, 클라이언트와 서버사이에 정보교환을 목적으로 하는 APIs(Application Programming Interfaces)로서, 비교적 간단하고 UNIX-TCP/IP 클라이언트/서버 시스템에 가장 많이 쓰이고 있다.[5] APPC/LU6.2는 IBM의 SNA네트워크하에서 상업용 응용프로그램간의 통신을 위한 프로토콜이며, MMS는 제조관련 응용소프트웨어에 쓰이는 프로토콜로서 현재 IBM PC, DEC, HP등 여러종류의 컴퓨터 시스템에서 사용가능하다. 소켓에 비해 복잡하고 정교한 APPC와 MMS는 표준화된 응용소프트웨어의 개발을 용이하게하지만, 반면 이해하기 어렵고, 커맨드마다 옵션이 많아 사용하기 어려운 단점을 지니고 있다. APPC와 MMS는 분산 응용소프트웨어 개발을 위한 정교한 프로토콜이라는 공통점이 있는 반면,

www.dbpia.co.kr

LU6.2는 MMS와는 달리 객체지향접근(object-oriented approach)을 할수 없으나, MMS에서는 지원하지 않는 정보보안(암호화, 여러단계비밀번호, 제한적 억세스등)과 two-phase commit protocol과 같은 데이타의 무결점을 유지하기 위한 기능등을 제공한다.

RPC는 세션계층 프로토콜로써 특별히 NFS을 위하여 개발되었지만 많은 네트워크 응용프로그램, 즉 클라이언트/서버 모델에서 널리 사용된다.[2, 5] RPC는 메세지 교환의 기본을 형성하며, 그 명칭이 가리키듯 고급 프로그램언어의 서브루틴 콜이나 프로시져(procedures)와 유사한 형태로 지원되기 때문에 분산처리 시스템의 설계 및 구현을 용이하게 한다. 이론적으로는 RPC가 LU6.2나 MMS와 같은 프로토콜과 서비스들의 상위에서 구현되어질 수 있지만 현재 사용가능한 대부분의 RPC는 소켓위에서 구현되었다. 그림 3은 PRC의 동작과정을 보여준다. 클라이언트와 서버 루틴은 두개의 독립된 프로세스이며, 대개는 두개의 독립된 시스템에 존재한다. RPC소프트웨어는 서버와 동일한 이름을 갖는 dummy procedure를 만들고 이를 클라이언트 프로세스에 상주시킨다. 이 dummy procedure를 stub라 부르며 콜링 파라메타를 취하여 이것들을 네트워크 전송 메세지의 형태로 구성한다. 서버쪽에서도 동일한 형태의 stub가 생성된다. RPC의 수행과정을 단계별로 보면 먼저 클라이언트가 client stub라 불리는 local procedure를 호출한다. 이 stub는 주로 RPC를 적절한 네트워크 메시지로 변형하고, 또한 서버의 실제 네트워크 주소를 결정해서 그것을 BIND한다. 그리고 네트워크 메세지는 네트워크 수송 메타니즘으로 보내지고, 클라이언트 시스템의 layer 1-4에 의해 통신매체를 통해 보내진다. 서버네트워크시스템은 서버stub에게 메세지 수신을 알리고, 서버 stub는 네트워크 메세지를 수신하여 그것을 local procedure call의 형태로 번역하여 서버에 실행을 요청하면, 서버는 call을 실행하여 결과를 서버stub에 보낸다. 서버stub는 실행결과를 하나 또는 여러개의 네트워크 메세지로 번역하여 네트워크시스템에 보낸다. 서버네트워크 시스템은 이를 클라이언트 네트워크 시스템으로 보내게된다. 그러면 클라이언트 네트워크 시스템은 client stub에 메세지를 보내고 call response로 번역하여 클라이언트에 보내게 된다.

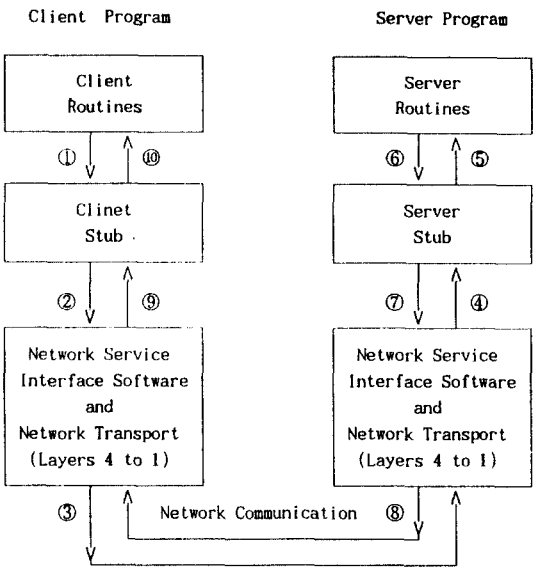


그림 3. RPC 동작과정
Fig 3. RPC facility

Ⅲ. 시스템 제안

RPC가 네트워크 프로토콜에 독립적이고 고수준의 커맨드를 제공하므로 네트워크상의 분산처리를 위해 훌륭한 도구임에 틀림이 없으나, 그림 1의 VM(Virtual Machine)하에 VSE(Virtual Storage Extended)를 사용하고 있는 연구환경에서는 적용할 수 없다. 왜냐하면 RPC는 유닉스이외에 설치 가능한 OS로는 MS-DOS, VMS, MVS등이고 더우기 LAN상에서 액세스하고자 하는 데이타베이스는 VSE하에 존재하며, VSE용 TCP/IP는 발표되지 않았기 때문이다. 이와 같은 상황에서 IBM시스템과 LAN상의 워크스테이션간에 TCP/IP를 통한 분산처리는 VM과 VSE의 통신 이외의 다른 방법은 없다. 그러나 이 방법은 UNIX에서 VM으로, VM에서 VSE로 패킷(PACKET)을 전달하여야 하므로 비경제적이다. 따라서 LAN상의 PC는 TCP/IP를 통하여 워크스테이션에 접근하고, IBM과 워크스테이션간에는 TCP/IP를 사용하지 않고 SDLC(Synchronous Data Link Control)로 그림 4와 같은 접속방법을 제안한다.

SDLC링크를 통한 통신은 IBM SNA(System Network Architecture)의 APPC를 이용하고 워크스테이션상에 APPC지원 도구로는 SNA서비스6000

www.dbpia.co.kr

시스템을 사용한다.

따라서 LAN상의 PC는 TCP/IP를 이용하여 RS-6000에 접속한 다음 클라이언트 프로세스를 가동하면, 이는 SNA SVCS/6000을 통하여, IBM 시스템의 데이타베이스 서버인 CICS(Customer Information Control System) 트랜잭션과 통신하게 된다.

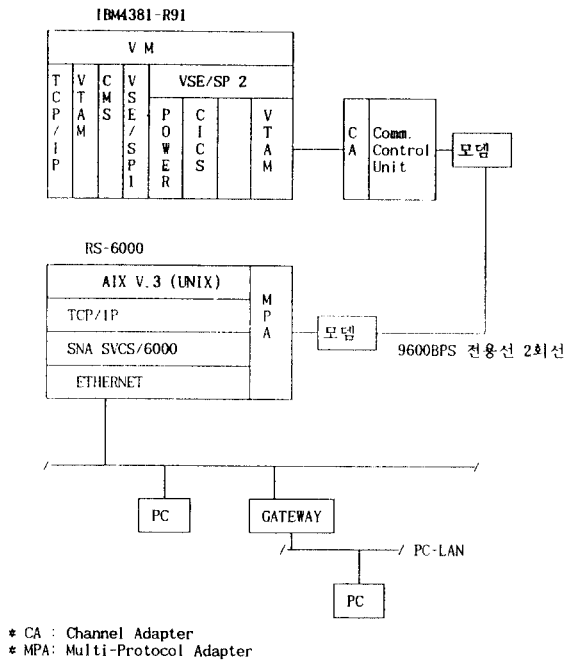


그림 4. IBM과 WS간의 SDLC 접속방법
Fig 4. The SDLC connection between IBM and WS

3.1 SNA

SNA의 구성은 그림 5와 같이 VTAM(Virtual Telecommunications Access Method), NCP(Network Control Program), SDLC로 구성된다. VTAM은 SSCP(System Services Control Point)라고도 불리며 메인프레임에 상주하면서 응용프로그램, 터미널, 워크스테이션들간의 end to end 세션을 관리하며, 메인프레임상의 응용프로그램에 인터페이스를 제공한다. NCP는 CCU에 상주하면서 네트워크 경로제어와 전송제어를 수행한다. SDLC는 SNA의 링크 프로토콜이며 OSI모델 2계층에 속하는 기능을 수행한다.

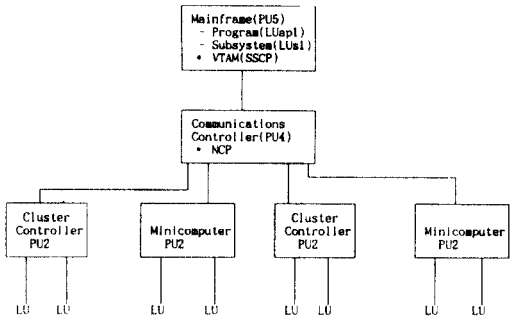


그림 5. SNA의 구성도
Fig 5. SNA components

• LOGICAL UNITS(LU)

SNA는 사용자들이 SNA네트워크를 통하여 상호 교신할 수 있는 접속점을 제공하는 LUs(logical units)를 정의한다. LU는 사용자가 볼 수 있는 소켓(socket)이나 포트(port)로 생각할 수 있으며, 물리적인 것이 아니라 논리적인 것이다. SNA는 몇가지 종류의 LU를 정의한다. 이들 각각은 사용자의 종류에 따라 각기 다른 전송능력과 서비스를 제공한다. LU는 여러가지 장치에 소프트웨어나 마이크로 코드의 형태로 구현되며 다음과 같은 LU 형태가 존재한다.

- LU0 : 자유롭게 데이터 스트림을 구성할 수 있는 응용프로그램용
- LU1 : 프린터나 커브드 등 디스플레이 장치가 아닌 입출력 장치 액세스
- LU2 : 표시장치 액세스
- LU3 : 프린터 액세스
- LU4 : LU1과 동일하게 단말 액세스(프로토콜에 다소 차이가 있음)
- LU6 : LU간의 통신(CICS<->CICS)
- LU6.1 : LU간의 통신(CICS<->IMS)
- LU6.2 : LU간의 범용적인 통신지원(APPC)
- LU7 : IBM 5250 데이터 스트림으로 디스플레이 장치 액세스

• PHYSICAL UNITS(PU)

SNA네트워크는 물리적으로 여러 종류의 장치와 이들을 연결하는 통신링크로 구성된다. 네트워크를 형성하는 장치들은 전산기, 여러가지 종류의 제어장치, 단말기등이 된다. SNA는 네트워크상의 실제장치들을 표현하기 위해 PUs(Physical units)라는 용어를 사용한다. PU는 특정 장치를 사용·관리하며,

www.dbpia.co.kr

이 장치와 관련된 통신접속장치를 관리하는데 필요한 서비스를 제공한다. SNA 노드는 네트워크상에 장치나 그것의 자원을 나타내기 위해 항상 하나의 PU를 갖는다. PU는 소유하고 있는 노드 타입과 같은 타입명칭이 부여된다. 따라서 PU는 PU타입 1, 2, 4, 5 중의 하나가 된다. 여기서 PU타입의 구조적 정의는 SNA가 발전하면서 강화되었다. 가장 포괄적인 기능을 구현한 PU타입 2는 현재는 PU 2.1로 발전되었다. 이는 APPC를 구현할때 LU6.2와 함께 사용되는 PU이다.

- PU1 : 디스플레이 단말이나 프린터
- PU2 : 단말 클러스터 제어장치
- PU2.1 : PU2의 강화형으로 대등통신(Peer to Peer)및 병렬세션의 지원
- PU4 : 통신제어 장치에 해당
- PU5 : 호스트에 해당

3.2 SNA 서비스

SNA 서비스는 IBM이 자사의 UNIX(AIX) 워크스테이션인 RS-6000에서 SNA를 지원하기 위해 개발한 소프트웨어로써 그림 6과 같이 구성된다. 시스템 지원 콘트롤러는 AIX 서브시스템들을 명령어나 서브루틴을 통하여 개시와 종료, 상태조회 등을 가능하게 한다. 응용프로그램은 라이브러리 서브루틴을 호출하여, 원격지 노드와의 접속을 개시하고 로칼 프로그램과 원격지 프로그램간의 데이타를 주고 받는다.

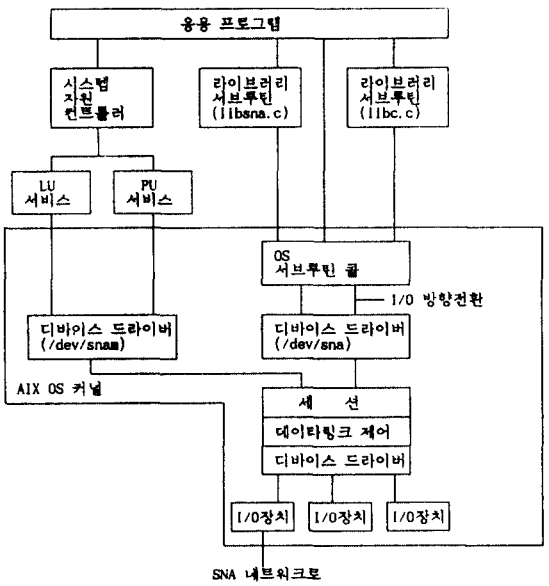


그림 6. AIX SNA서비스 시스템 구성도
Fig 6. AIX SNA Services/6000 system components

3.3 제안 시스템 환경설정

단말기처럼 주종관계가 아닌 두개의 프로그램이 동등한 입장에서 통신할 수 있는 이른바 APPC를 제공하는 LU6.2링크를 구현하기 위해서는 IBM호스트의 NCP, VTAM, CICS의 환경과 워크스테이션의 SNA서비스 환경설정이 필요하다.

NCP는 서브에리어 주소, PU와 LU들의 이름, 전송 속도, 최대 송수신 데이타의 길이 등 여러가지 정보를 갖도록 해야하며, VTAM에는 LU6.2 링크를 지원할 CICS를 등록해야 한다. 또한 CICS에는 이러한 LU들을 인식할 수 있도록 각종 제어 테이블을 수정하여야 한다. SNA서비스는 각종 프로토콜의 접속, PU와 LU이름과 특성, 가동될 트랜잭션 등 각종 제어정보를 해당 프로화일(profile)로 부터 얻고 있으므로, 이에 필요한 각종 정보를 작성해야 한다.

본 논문을 위한 환경은 IBM 호스트의 VTAM 3.2, CICS 1.7, NCP 5.2.1, SSP 3.4.1과 RS-6000시스템의 AIX 버전 3.2하에서 구현하였다.

Ⅳ. 시스템 설계

온라인 수강신청시스템은 단순히 네트워크를 통한 수강신청만을 받는 시스템이 아니라, 과목신청시 수강정원 초과, 이수과목 중복, 학점제한, 수강학과 제한, 선수과목 이수여부 등 각종 점검을 위해서 사전에 학생신상 및 이수학기별 성적관리를 비롯하여 일관성 있는 교과목 코드관리, 강의실 관리등 학사관리 전반에 걸쳐 전산화가 선행되어져야 한다. 그러므로 온라인 수강신청 시스템은 단독 시스템으로 운영될 수 없고 학사관리시스템의 보조시스템으로 운영된다.

온라인 수강등록 시스템은 학생들의 원활한 수강신청과 학생 1인당 단말기 점유시간의 단축을 위하여 다음과 같은 서비스가 가능하여야 한다.

1) 정원초과, 학과제한등의 이유로 인해 해당과목의 신청이 불가능할 시 타분반 조회 서비스
2) 과목 입력시점에서의 시간표 서비스
3) 대학별 또는 학과별 수강신청 안내 서비스
4) 학사 일정 조회 서비스

www.dbpia.co.kr

5) 비밀번호 변경

이 외에도 현주소가 변경된 학생의 경우, 별도로 학적과를 통하지 않고 온라인 수강신청시스템을 이용하여 학생이 직접 수정토록 함으로써 가정 통신문이나 등록고지서 발송 등의 업무를 신속히 처리할 수 있도록 하여야 한다. 또한 수업관리에 대한 학생들의 불만이나 제도 개선요구에 능동적으로 대처하기 위해 학생이 건의 사항을 입력할 수 있도록 한다. 온라인 수강등록 시스템은 그림 7과 같은 구조도를 갖는다.

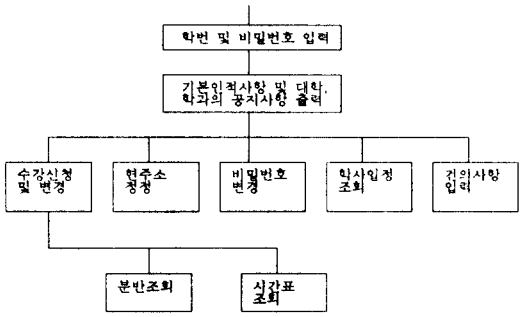


그림 7. 온라인 수강신청 시스템 구조도

Fig 7. Block diagram for the on-line course registration system

학생의 기본인적사항, 교과목, 성적자료와 같은 수강신청 시스템 운영에 필요한 각종 데이타들은 호스트 시스템에 유지된다. 워크스테이션의 성능을 고려하여 워크스테이션에서의 화일 입출력은 하지 않으며, 각종 코드변환 작업은 워크스테이션쪽에서만 수행한다. 처리흐름도는 그림 8과 같다.

클라이언트 프로세스는 화면상의 입출력, 입력된 데이타의 에러 점검기능, 한글 변환 작업, 시간표 작성등과 같은 기능을 수행하게 되며, 서버 프로세스는 클라이언트로부터 전송된 데이타에 따라 해당 화일을 액세스하여 클라이언트가 요구하는 형태로 처리하여 전송하는 역할을 수행한다.

송수신되는 데이타는 그림 9와 같이 가변장 레코드로 하며, 처음 2바이트는 화일 액세스를 요청하는 클라이언트(워크스테이션)쪽에서 보낼 때는 수행요청 기능코드가 되고, 서버가 되는 호스트 트랜잭션쪽에서 보낼 때는 수행한 기능의 에러코드가 되도록 설계한다. 그 다음 바이트 부터의 데이타는 수행요청 코드에 따라 송수신 데이타 형식이 결정된다. 송수신 데이타의 총 길이는 전송효율을 고려하여 1,024 바이트이내로 한다.

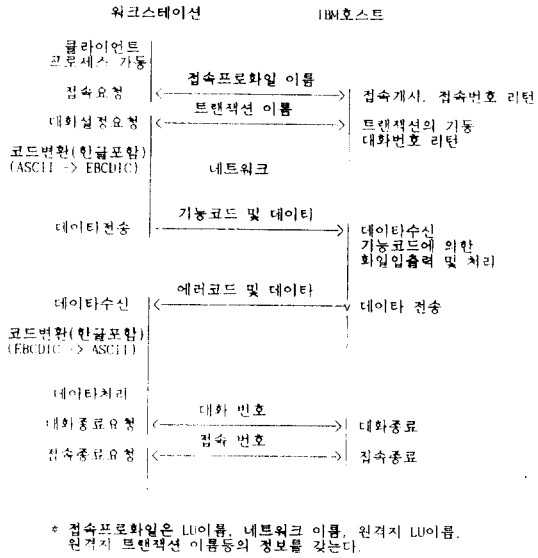


그림 8. 수강신청 시스템 처리 흐름도

Fig 8. Flowchart of the course registration system

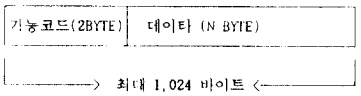


그림 9. 송수신 데이타 구조

Fig 9. A structure of send/receive data

클라이언트 프로그램은 통신할 호스트의 LU이름, 네트워크이름 등의 정보를 갖고 있는 접속 프로화일 이름만을 알면 족하고 그 이하의 네트워크상에 복잡한 통신과정은 프로그램에는 투명(transparent)하다. 따라서 프로그램은 SNA서어비스가 제공하는 몇 가지 라이브러리 함수를 사용하여 구현하면 된다. 그러나 통신상에 일어나는 모든 유형의 오류에 대한 처리 및 한글을 포함한 코드변환 등은 개발자의 몫이 된다. 통신 및 응용프로그램상의 각종 오류처리를 위하여 표1과 같이 오류코드표를 작성하였다.

서버와 송수신되는 데이타는 기능별로 분류하여 표2와 같이 작성하였다. 프로그램의 가동은 접속 프

www.dbpia.co.kr

표 1. 각종 오류 코드표
Table 1. Error code table

코드	오류 내용
00	정상
01	접속 실패
02	트랜잭션 가동실패
03	전송 실패
04	전송 자료 없음
05	수신 실패
06	수신 자료 없음
07	트랜잭션 종료 실패
08	접속 종료 실패
09	CICS 오류
10	학번 오류
11	비밀번호 오류
12	화일 공간 없음
13	구분 오류
14	종별 오류
15	분반 오류
16	학정번호 오류
17	해당 교과목 수강 불가
18	해당 분반 수강불가
19	이미 이수한 과목
20	수강 과목 초과
21	학기당 이수학점 초과
22	수강정원 초과
23	수강신청과목 없음
24	당학기 개설하지 않음
25	학기당 이수학점 미달

로화일 이름을 공급받음으로써 개시되고, 처음 로고 화면에서 학생은 학번과 비밀번호를 입력한다. 그러면 프로그램은 접속 프로화일내의 정보를 이용하여 호스트와 세션을 맺고, 표2에서 보인바와 같이 학생 신상자료를 요청하기 위한 기능코드 '01'과 학번및 비밀번호를 전송한다. CICS 트랜잭션은 데이타를 수신하여 코드 '01'에 따른 데이타를 인식하고 학번과 비밀번호를 축출하여 해당 화일을 액세스한다. 트랜잭션은 이를 미리 클라리언트 프로세서와 약속한 형태의 레코드로 작성한 다음 전송하게 된다. 그러면 클라이언트는 이를 수신하여 처리를 계속하게 된다. 클라이언트에서 송신하는 모든 데이타는 IBM호스트 내부 코드인 EBCDIC 코드로 변환하여 송신되고 서버로부터 수신된 모든 데이타는 다시 ASCII 코드로 변환된다.

표 2. 기능코드별 송수신 데이타
Table 2. Send/Receive data corresponding to each operational code

기능코드	송신데이타	수신데이타
01	학번, 비밀번호	학생 기본인적사항 대학별 공지사항
02	수강신청학기 수강신청 화일이름	수강신청 내역
03	추가 및 삭제코드, 이수구분 학정번호, 분반번호	교과목명, 교수명 요일별 시간
04	학정번호	신청가능여부, 교수명 요일별 시간
05	새로운 비밀번호	
06	전화번호, 우편번호 현주소	
07	학번	건의사항
08	건의사항	
09	조회 시작 KEY와 방향	학사일정

• 접속 프로화일 이름의 동적 할당

수강등록 시스템은 불특정 다수에 의해 가동되는 어플리케이션이므로 접속 프로화일 이름 조차도 사용자에게 입력하게 할 수 없다. 따라서 프로그램이 가동될때 마다 동적으로 접속 프로화일 이름을 활당하기 위해서 다수의 프로세스가 동시에 사용할 수 있는 공유 메모리(shared memory)에 프로화일 이름 테이블을 관리하는 서버를 별도로 작성하였다. 공유 메모리의 구조는 다음과 같다.

```
struct dataStream{
  unsigned int   clientPid;
  char           luTable[MAXLU];
};
```

클라이언트 프로세스는 공유메모리의 상호배제를 위하여 먼저 clientPid를 검사한다. clienPid가 '0'이면 pid(process id)를 clientPid에 쓰고 luTable를 검사하여 비어있는 ('0')곳이 있으면 할당('1')하고 clientPid를 다시 '0'으로 만든 다음 테이블의 첨자를 리턴한다. 리턴된 테이블의 첨자는 "RSDLU"와 조합하여 접속 프로화일 이름이 된다. 만약 clientPid가 '0'보다 크면 프로세스는 대기하게 된다. 프로세스가 종료하기 전에는 반드시 위와 같은 방법으로 공유 메모리를

www.dbpia.co.kr

액세스하여 해당 데이블을 '0'으로 써서 다른 프로세스가 가동될 때 사용가능하도록 한다.

• SNA서비스 함수의 구현

호스트 시스템과의 대화 개시, 트랜잭션의 가동, 데이타의 송수신등 SNA 서비스함수를 이용하여 구현되는 APPC 부분만을 위한 함수 sna( )를 별도로 작성하여 필요시 호출하여 사용할 수 있도록 하였다. sna( ) 수행과정은 그림 10과 같다.

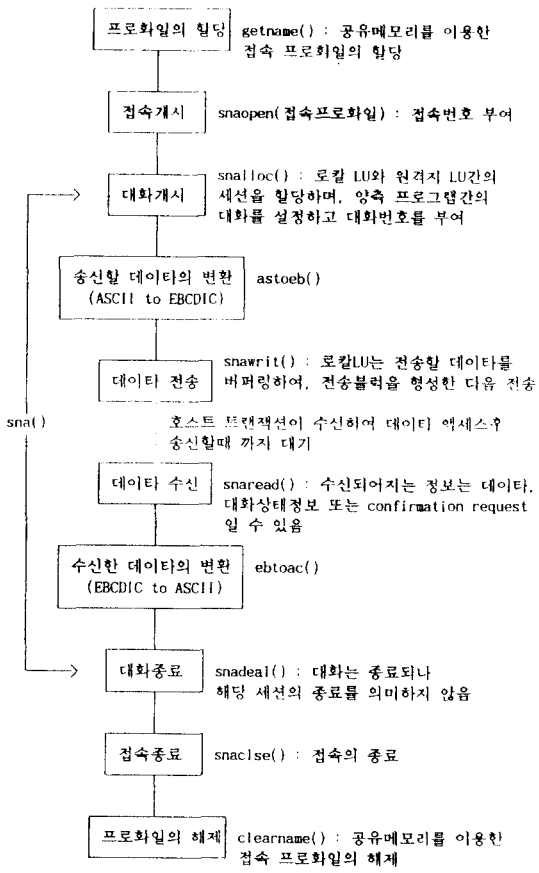


그림 10. sna( )의 수행과정과 과정별 SNA서비스 함수
Fig 10. Flow of sna( ) and SNA service functions

• 수강신청 및 변경 프로그램

온라인 수강신청 시스템은 그림 7에서와 같이 크게 5가지의 주 프로그램과 2가지의 서브 프로그램으로 구성된다. 이중 가장 핵심인 수강신청 및 변경 프로그램의 수행과정을 설명하면 다음과 같다.

1) 그림 11과 같은 수강신청 및 변경화면을 출력한다.

2) 기능코드 '02'와 수강신청 화일 이름 및 수강신청 학기를 전송하고 서버로부터 수강신청 내역을 수신한 다음 추가, 삭제가 용이하도록 링크드 리스트로 구성한다.

3) 링크드 리스트의 내용을 화면에 출력한다.

4) 화면으로부터 수강신청 및 삭제를 위한 구분코드, 이수구분, 교과목코드, 분반번호를 입력 받는다.

5) 기능키 F1이 감지되면 4)에서 입력받은 교과목코드의 분반조회 함수를 호출하여 수행후 4)로 간다.

6) 기능키 F2가 감지되면 시간표조회 함수를 호출하여 수행후 4)로 간다.

7) 기능키 F3가 감지되면 처리를 종료한다.

8) 수강신청 기간을 점검하여, 수강신청 기간이 아니면 입력된 데이타를 무시하고 4)로 간다.

9) 구분코드와 이수구분코드를 점검하고, 링크드 리스트로부터 기 수강신청한 과목과의 중복 신청여부를 체크하여, 오류가 발생하면 오류내역을 화면에 출력하고 4)로 간다.

10) 기능코드 '03'과 입력된 데이타를 호스트로 전송하고, 서버로부터 오류코드, 교과목명, 담당교수명, 시간표등의 데이타를 수신한다. 만약 오류가 발생했으면, 해당 오류내용을 화면에 출력하고 4)로 간다.

11) 추가 및 삭제코드에 따라 링크드 리스트를 정리하고 3)으로 간다.

프로그램 개발시 화면 형식이나 기능키 설정, 과목 입력방법 등은 사용자의 혼동을 피하기 위해 현재 모델 대학에서 운영되고 있는 온라인 수강 신청시스템의 예를 그대로 따랐다. 특별히 기능키를 자유롭게 설정하기 위해서 input_string( )함수를 별도로 작성하였다. 이 함수는 화면의 특정 위치로부터 원하는 양만큼의 데이타를 입력받고, 입력과 동시에 데이타 구분(영숫자, 숫자 등)에 따라 입력데이타를 체크할 수 있도록 하였으며, 기능키의 변경 등 키맵(key map)이 수정될시 변경이 용이하도록 작성하였다.

• 서버용 CICS 프로그램

데이타 서버인 CICS 트랜잭션 프로그램은 아주 많은 수의 사용자를 지원해야 하므로 CICS의 메모리 용량을 고려하여, 필요할 때마다 클라이언트로부터 불러서 쓰는 형태를 취하도록 한다. 트랜잭션은 단지 데이타를 수신한 후 해당 화일을 액세스하여 처리하여 전송하면 되고, 워크스테이션쪽의 프로그램을 호출

www.dbpia.co.kr

```
*** 수 강 신 청 및 변 경 ***

학 번 : 1234567    대 학 : 이과대학        학 과 : 전자계산과
성 명 : 홍길동  (남) 3 학년               1993/02/24 복학 허가

과목종별  학정번호   학점  교 과 목 명                    교수명
교 선     U1212-01   1     베드민턴                       강교수
전 필     C1206-01   3     소프트웨어 실습                윤교수
.
.
.
.
.
.
.
.
.
.

구분 :   과목종별 :   학정번호 :   -          신청학점 계 : 4
MESSAGE :

F1 = 과목HELP   F2 = 시간표   F3 = 종료   구분 : A = 신청  D = 삭제
```

그림 11. 수강신청 및 변경 화면 형식
Fig 11. Screen format for course registration

할 필요가 없으므로 일반적인 온라인 트랜잭션 프로그램과 동일하다.

수강신청 및 변경프로그램의 수행시, 기능코드에 '03'과 구분코드에 과목추가를 알리는 'A'를 수신하였을 때의 작업수행 내용을 소개하면 다음과 같다.

1) 해당 학생의 수강신청 화일을 읽는다.

2) 당하기 개설과목에 대한 모든 정보(정원, 담당교수, 수강제한 학과, 선수과목 등)를 갖고 있는 화일에서 추가하고자 하는 과목의 레코드를 읽는다.

3) 과목의 수강제한 학과 여부를 체크한다.

4) 학기별 성적화일을 읽어 기 이수한 과목인지를 체크한다.

5) 수강과목의 초과 여부를 체크한다.

6) 전학기 학사 경고등의 이유로 당학기 신청학점의 제한 여부를 체크한다.

7) 수강정원의 초과 여부를 체크한다.

8) 1-7)수행시 오류가 발견되면 해당 오류코드를 전송하고, 그렇지 않으면 수강신청 화일에 추가 과목을 쓰고, 개설과목 화일의 수강전원에 1을 더하여 수정한다.

9) 추가된 과목의 학점, 담당교수, 강의시간 등의 자료를 클라이언트로 전송하고 종료한다.

V. 적용실험 및 결과분석

실험은 개발시스템을 이용한 접속과 TN3270을 통한 접속을 병행사용한 경우와 전체 단말을 TN3270을 통하여 메인프레임에 접속하여 사용한 경우의 평균 응답시간을 비교하였으며, 표3과 같은 결과를 얻었다. 개발시스템 사용자는 워크스테이션의 용량을 고려하여 50대로 제한하였다. 실험결과 단말수 150대에서부터 응답시간의 차이를 보여 단말수가 많아질수록 그차가 커짐을 볼 수 있다. 이는 워크스테이션의 용량을 증가시키거나, 또다른 워크스테이션을 활용하여 클라이언트 수를 늘인다면 더욱 좋은 결과를 얻을 수 있음을 의미한다.

표 3. 단말수별 평균응답시간 비교표(단위 : 초)
Table 3. Comparison table of average response time

전체 단말수	TN3270만 이용시 응답시간	개발시스템과 병행시 응답시간
100	4	4
150	6	5
200	9	7
250	13	10.5

그러나 클라이언트수를 대량으로 늘릴 경우, 현재 9600BPS 전송 속도로는 CCU(Communication Control Unit)의 병목현상을 초래할 것이므로, CCU에 토큰링 어댑터를 장착하여 10M BPS이상의 전송속도를 유지하는 것이 필요하며, 데이타 버퍼를 위해 CCU의 메모리를 증설해야 할 것이다.

네트워크상의 이러한 유형의 데이타 액세스는 호스트 시스템의 부하를 네트워크 상의 워크스테이션으로 분산함으로 인해 호스트 시스템의 수명(Life Cycle)연장의 효과를 가져오게 되며, 호스트 시스템으로의 직접 로그온을 하지 않기 때문에 정보 보안을 유지할 수 있으며, 네트워크 상의 시스템을 다양하게 활용하여 전산장비 증가를 억제, 이에 따른 비용을 절감할 수 있다. 또한 워크스테이션이나 퍼스날 컴퓨터의 특성을 살려 새로운 유형의 사용자 인터페이스의 제공을 가능하게 한다.

www.dbpia.co.kr

Ⅵ. 결 론

본 논문은 LNA상에 다양한 전산자원을 보유하고 있는 대학 전산환경에서, 보안이나 관리상의 이유로 응용프로그램들을 전적으로 대형 주전산기에 의존하게 함으로써, 빈번한 주전산기의 교체로 인한 비용의 증가, 한정된 서비스의 제공 등 발생할 수 있는 여러 가지 문제점을 해소하기 위하여, 네트워크상의 다양한 전산자원을 효과적으로 활용하는 분산처리시스템을 구현하고자 하였다.

이를 위해 모델 대학의 환경을 파악하여, 온라인 수강신청 시스템을 분산처리 대상업무로 선정하였으며, 네트워크 환경하에서의 분산처리 방안을 고찰하여 모델 대학에 가장 적합한 분산처리 방안을 제안하였다. 제안한 분산처리 방안은 온라인 수강신청 시스템을 화일 입출력 부분과 각종 처리 부분으로 나누어, 주전산기인 IBM시스템이 화일 서버 역할을 하고, 워크스테이션으로 하여금 사용자 인터페이스를 비롯한 각종 처리를 담당하게 하는 이른바 클라이언트/서버 모델을 이용한 분산처리 시스템을 구현하는 것이다.

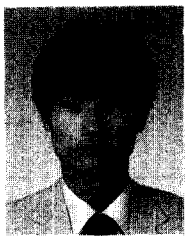
시스템의 구현을 위하여, 주전산기인 IBM과 워크스테이션간의 물리적 연결은 SDLC링크를 이용한 전용회선을 사용하여 접속하였으며, SNA의 LU6.2 (APPC)링크를 통하여 주전산기의 CICS 응용프로그램과 워크스테이션의 응용프로그램간의 연결처리를 구현하여, 실제 업무에 적용하여 실험하였다. 실험결과 주전산기에 부하를 경감시켜 전체적인 성능이 현저하게 향상되었음을 확인하였다. 향후 SDLC 접속을 토큰링 어댑터로 대체하여 10M BPS상의 전송속도를 유지하고, 워크스테이션의 수를 증가시키면 더욱 좋은 결과를 기대할 수 있을 것이다.

이러한 형태의 분산처리 시스템은 작고, 값싸고, 강력한 기능을 보유한 워크스테이션의 보급으로 인하여 네트워크를 가진 대다수의 기업이나 기관들이 고려하여 있으며, 급속도로 확산될 전망이다. 그러나 분산시스템은 장애 발생시 유지보수 작업이 어렵고, 이기종간의 인터페이스를 위한 지원도구들이 부족하며, 시스템 개발방법의 변경에 따른 프로그래머 재교육의 필요성 등의 단점도 있다. 따라서 분별없는 분산처리 시스템의 구현은 유지보수를 위한 인력 및 비용의 낭비를 가져올 수 있으므로 신중한 분석 단계를 거쳐서 실시되야 한다고 본다.

참 고 문 헌

1. 정진욱, 변옥환, IBM 네트워크 기술과 응용, Ohm사, 1989.
2. Michael Santifaller, *TCP/IP AND NFS*, Addison-Wesley, 1991.
3. James Martin and Joe Leben, *Data Communication Technolgy*, Prentice-Hall International, Inc. 1988.
4. W. Richard Stevens, *UNIX Network Programming*, Prentice-Hall International, Inc. 1991.
5. Amjad Umar, *Distributed Computing*, Prentice-Hall International, Inc. 1993.
6. Andrew S. Tanenbaum, *Computer Networks*, Prentice Hall International, Inc. 1988.
7. Shatz, S. M., and J. P. Wang, *Introduction to Distributed Software Engineering*, IEEE Computer, Oct. 1987.
8. Nehmer, J., and Mattern, F., *Framework for the Organization of Cooperative Services in Distributed Client Server System*, Computer Communication, Vol. 15, No. 4 May 1992, pp.261-269.
9. Ozsu, M., and Valdurez, P., *Distributed Database System: Where are We Now?*, IEEE Computer, Aut. 1991, pp. 68-78.
10. Wilbur, S., and Bacarisse, B., *Buiding Distributed Systems with Remote Procedure Calls* Software Engineering Journal, Sept. 1987, pp. 148-159.
11. Griswold, C., *LU6.2: A View from the Database*, Database Programming and Design, May 1988, pp34-39.
12. Gray, J., et al., *Advanced Program to Program Communications in SNA*, IBM Systems Journal, Vol.22, No.4, Special Issue on SNA, 1983, pp. 298-318.
13. Benjamin, J. H., *Interconnecting SNA Networks*, IBM Systems Journal, Vol. 22, No.4, Special Issue on SNA, 1983, pp.344-366.
14. Gantz, J., *Cooperative Processing and The Enterprise Network*, Networking Management, Jan. 1991, pp.25-40.

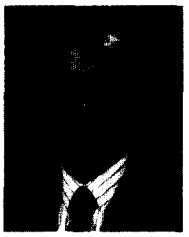
www.dbpia.co.kr



高 光 炳(Kwang Byung Koh) 正會員
1957年 10月 25日生
1983年 2月 : 光云大學校 電子計算學科(理學士)
1994年 2月 : 延世大學校 産業大學院 電子計算專攻(工學碩士)
1983年 12月 ~ 現在 : 延世大學校 電算院 主任

孔 勝 郁(Seung Wook Kong) 正會員
1953年 1月 1日生
1975年 2月 : 慶熙大學校 電子工學科(工學士)
1994年 2月 : 延世大學校 行政大學院 一般行政專攻(行政學碩士)
1981年 3月 ~ 現在 : 延世大學校 電算院 主任

權 奇 睦(Gi Mok Kweon) 正會員
1953年 5月 20日生
1978年 2月 : 國際大學 經濟學科(經濟學士)
1982年 2月 : 延世大學校 産業大學院 電子計算專攻(工學碩士)
1979年 8月 ~ 現在 : 延世大學校 電算院 課長



康 昌 彦(Chang Eon Kang) 正會員
1938年 8月 26日生
1960年 : 延世大學校 電氣工學科(工學士)
1965年 : 延世大學校 大學院 電氣工學科(工學碩士)
1969年 : 美國 미시간주립大學校 大學院 電氣工學科(工學碩士)
1973年 : 美國 미시간주립大學校 大學院 電氣工學科(工學博士)
1967年 ~ 1973年 : 美國 미시간주립大學校 工業研究所 先任研究員
1973年 ~ 1981年 : 美國 노던일리노이大學校 電氣工學科 助教授, 副教授
1982年 ~ 現在 : 延世大學校 電子工學科 教授
1987年 ~ 1988年 : 本 學會 副會長
1989年 ~ 1990年 : 本 學會 會長

www.dbpia.co.kr