



리눅스에서 한글 인터페이스

윤 지 현[†] 변 정 용^{††} 최 송 목^{†††}

◆ 목 차 ◆

- | | |
|---------------|----------------------------|
| 1. 서 론 | 4. Hanterm과 Netscape의 한글환경 |
| 2. 리눅스의 한글 환경 | 5. 한글 코드와 리눅스 |
| 3. 국제화와 지역화 | 6. 결 론 |

1. 서 론

현재 리눅스의 열풍은 국내뿐 아니라 전세계적이다. 이것은 무료로 구할 수 있다는 장점에다가 웹 서버로 그 성능을 충분히 입증 받았기 때문이다. 물론 웹 서버 뿐만이 아닌 삼바를 통한 파일 서버나 프린터 서버로도 충분히 성능이 검증되었고, 심지어 리눅스를 탑재한 PC로 라우터까지 대체하려는 프로젝트[16]가 진행중이다. 앞에서 살펴본 바와 같이 리눅스를 중소기업 서버로 운영하기에 큰 무리가 없다. 또한 Sun의 StarOffice [17]등 Microsoft의 MS Office 못지않은 훌륭한 소프트웨어가 있다. 이것은 리눅스가 충분히 서버용 OS일뿐만 아니라 클라이언트용 OS까지 대체할 수 있다는 것을 의미한다.

하지만 리눅스는 영어권의 나라에서 작성이 되어 한글 환경이 매우 미흡한 실정이다. 이것은 사용자 하여금 리눅스 사용에 대한 어려움을 느끼게 하는 가장 큰 요인이다. 따라서, 국내의 리눅스 저변이 널리 확산 되려면 한글 환경이 충분히 갖추어져야 할 것이다. 이것은 소프트웨어의 한글화뿐만 아니라 각종 출력 메시지, 프린터 등

에서도 이루어져야 한다.

본 논문에서는 현재 국내 리눅스 한글 환경을 되짚어보고 리눅스의 한글 인터페이스를 분석하며, 앞으로 나아가야 할 방향을 제시한다.

본 논문은 2장에서 리눅스의 한글 환경을 알아보고, 3장 국제화와 지역화, 4장 Hanterm과 Netscape, 5장 한글코드와 리눅스, 6장 결론 및 향후 연구 방향으로 구성되어 있다.

2. 리눅스의 한글 환경

2.1 리눅스에서 한글 사용 노력의 역사

리눅스에서 한글이 지원된 최초의 환경은 한텀(Hanterm)이다. 한텀은 리눅스의 기본 구성요소중의 하나인 엑스텀(xterm)의 소스를 가져와 한글의 입력이 가능하게 오토마타를 추가하고 한글의 출력을 위하여 한글 글꼴을 추가하여 제작된 것이다. 이 한텀 이후 리눅스에서 한글 환경을 구현하기 위한 노력은 계속되었는데, X-Window상이 아닌 콘솔(console)상에서 한글의 입출력을 위한 한(han)과 X-Window상에서 MS-Windows의 IME와 같은 역할을 하는 PXHan과 버루 등의 응용 어플리케이션이 개발되었다. PXHan과 버루와 같은 한글 입력기는 xterm과 같은 영어 권에서 개발된 리눅스 응용 프로그램에서도 한글의 입력이 가능

† 정회원 : 동국대학교

†† 정회원 : 동국대학교 경주캠퍼스 전자계산학과 부교수

††† 정회원 : 한국교육미디어 대표이사

하였다. 또한 리눅스에서 가장 많이 쓰이는 웹 브라우저인 리눅스에 기반 한 X-Window에서 작동하는 넷스케이프(이하 Netscape)에서도 한글 사용 노력은 계속되었는데 초기의 리눅스용 Netscape은 한글의 출력에 어려움이 많았고 한글의 입력은 한텀(Hanterm)에서 입력한 글자를 마우스를 이용해 드래그앤드롭(drag & drop) 방식을 이용하였다. 그러나 지금은 많은 개발자들의 노력으로 미려한 한글을 쉽게 출력하는 것 뿐만 아니라, 각종 메시지까지 한글화하기에 이르렀다. 하지만 리눅스의 가장 기본이 되는 라이브러리인 glibc.so가 기본적으로 멀티바이트(multi-byte)코드를 처리하지 못하여 여전히 문제[1]를 안고있다.

2.2 한글 플랫폼별 지원 방법 비교

I18N(Internationalization)과 L10N(Localization)은 X-Window상에서 작동하는 프로그램이 국제적으로 사용이 가능하게끔 해주는 표준 규약[1,3,10]이다. 즉, I18N과 L10N을 따른다는 것은 개발된 소프트웨어가 전세계 어느 나라에서도 실행이 가능하며 그 나라 고유의 문자의 입출력을 처리할 수 있으며 화폐기호나, 자릿수 구분자를 정확히 표시한다는 것을 의미한다. 이 I18N을 충실히 따르는 응용 프로그램으로는 Netscape[3]이 있다. 따라서 Netscape은 아스키 코드(ASCII Code)를 기반으로 작성이 되었지만 한국이나 일본에서도 잘 작동이 되며 한국어와 일본어를 훌륭히 표현한다.

Hanterm은 xterm을 기반으로 작성된 한글 입력 터미널(terminal)인데, 이것은 Netscape과 달리 내부에 한글 오토마타와 글꼴 처리기를 탑재하여 한글을 처리한다. 이러한 방법은 최근에는 많이 사용되지 않는 방법인데, xterm이 처음에 개발되었을 당시에는 I18N이나 L10N과 같은 규약이 정의되지 않았고, 현재는 I18N과 L10N을 따르는 소프트웨어를 개발하면 개발 비용이 많이 줄기 때문이다. 따라서 한국의 X-Window 프로그램 개발

자들의 노력으로 xterm에서 멀티바이트(multi-byte) 처리를 가능하게 하고, 한글 오토마타를 내장시켜 기본적인 한글 입출력 환경을 개발하게 되었다.

2.3 glibc.so의 버전별 multibyte지원 여부 파악

아래의 표는 C라이브러리 버전별 멀티 바이트 지원여부[1]를 나타낸 것이다. 표에서도 알 수 있듯이 cc의 libc5시스템과 gcc의 glibc2.0.x에서는 멀티바이트 문자열을 다룰 수 없어서 X-Window의 xlocale을 사용하여 한글 처리가 필요한 소프트웨어가 강제로 로케일 정보를 읽게 하였다. 하지만 glibc2.1.x에 되서는 다소 불안하지만 멀티 바이트 문자열의 처리가 지원되고, libwscmsmb라이브러리를 사용하면 xlocale을 사용할 필요 없이 멀티 바이트의 처리가 가능하다.

GNU의 glibc팀[18]에 따르면 glibc2.2.x에서는 완전한 멀티 바이트 처리를 할 것이라고 한다.

(표 1) glibc버전별 multibyte지원 상황

	Libc5시스템	glibc 2.0.x + xlocale	glibc 2.0.x libwscmsmb	glibc 2.1.x
대표적인 배포본	RedHat 4.2	알짜Linux 5.x	RedHat 5.x	RedHat 6.x
libc의 국제화	지원하지 않음	one byte 문자권 지원	불완전하지만 멀티 바이트 문자권 지원	
X의 국제화	xlocale 사용	xlocale 사용	xlocale 사용하지 않음	

2.4 대표적 한글 플랫폼인 Netscape와 Hanterm에서 한글 사용 시 문제점

Netscape에서 한글 사용 시 발생하는 문제는 거의 파악이 되었지만 Netscape가 사용하는 그래픽 라이브러리인 모티프(Motif)는 소스가 공개되지 않은 상용 라이브러리기 때문에 함부로 소스를 고칠 수 없다. 따라서 Netscape에서 정상적인 국제화 방법으로는 해결할 수 없다. Netscape은 리눅스에서 매우 사용 빈도가 높은 소프트웨어이

기 때문에 모터프 라이브러리의 버그가 나타나지 않게 패치(Patch)[1]를 한 후 사용하도록 한다.

Hanterm은 그 동안의 노력으로 충분한 안정성을 가지게 되었다. 그러나 Hanterm3.0.4beta7까지 개발되는 동안 개발자들간의 연계가 전혀 없이 패치만 이루어 졌기 때문[2]에 소스가 지지분하다. 또한 글자꼴의 부족으로 사용자 취향에 맞는 미려한 서체의 화면 표현이 어렵다.

3. 국제화와 지역화

3.1 I18N과 L10N

유닉스와 리눅스가 처음 만들어 질 때는 영어권을 중심으로 하여 개발이 이루어져왔다. 따라서 이러한 운영체제 하에서 영어 이외의 문자를 표현하기 위해서 많은 노력이 있었다. 하지만 이러한 방법은 자국의 소프트웨어를 자국에서만 사용이 가능하도록 하는 결과를 낳았다. 이것은 똑 같은 기능을 하는 소프트웨어를 각 나라별로 개발할 때 드는 비용의 중복과 시간의 낭비라는 결과를 초래하게 되었다. 이러한 문제를 해결하기 위하여 로케일 시스템의 도입과 X-Window의 표준화된 다국어 입출력 방법, C언어 라이브러리에서의 I18N(국제화)가 마련되었다.

또한, 자국의 문자만을 입출력하는 방법뿐만 아니라 시간과 통화, 숫자 구분자 등의 각 나라 고유의 표기를 정확히 하기위한 노력도 이루어졌는데, 이것은 I18N을 위해 도입된 L10N을 따르는 로케일 시스템이 해결하여 준다.

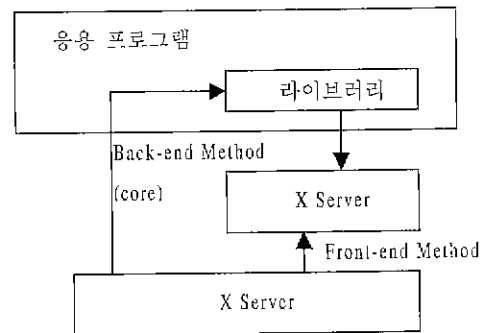
즉, I18N을 따른다는 것은 어떤 소프트웨어가 I18N에 맞추어 개발되면 전 세계 어떠한 나라에서도 국가 고유의 입출력에 문자가 없다는 것을 의미하고, L10N을 따른다는 것은 입출력되는 문자들이 그 나라 고유의 양식으로 변환된다는 것을 의미한다.

I18N을 따르는 대표적인 소프트웨어에는 Netscape

가 있는데, 유닉스나 리눅스 시스템에 자국의 글꼴과 언어 입력기만 있으면 Netscape에서 자국의 문자를 처리하기 위한 노력은 매우 적다. 여기서 완전한 처리가 불가능한 이유는 Netscape에서 사용하는 모터프에서 Multibyte문자 처리에 문제가 있기 때문이다.

3.2 XIM

X-Window의 국제화는 응용프로그램 개발자가 이식성있는 국제화된 프로그램을 만들고, 그것을 다른 고유의 언어와 지역적 관습, 문자열 부호화에 대한 요구사항에 적용시키는데 사용하는 공통의 API를 제공[10]한다. 지역화를 위한 기능 중의 하나로써 XIM(X Input Method)라고 하는 국제화된 텍스트 입력에 대한 기능적 인터페이스를 정의[10]하였다. hanIM이나 ami, KIMS등의 한글 입력서버는 모두 XIM을 따르는 응용 소프트웨어이다.



4. Hanterm과 Netscape의 한글환경

4.1 Hanterm에서의 한글 처리

한텀은 처음 개발단계에서는 자체 개발방식을 취하고 있었지만 한텀2.0부터는 xterm의 소스를 고쳐 한글의 입출력을 처리[2]하게 만들었다. 즉, 기존의 xterm이 영어 문자만을 처리하던 것을 한텀의 개발자가 한글 오토마타와 출력 처리부를 추가하여 만든 것이다.

한컴에서 입력이 가능한 한글 코드는 조합형, EUC-KR, UTF-8의 3가지 이다. 여기서 UTF-8은 유니코드를 사용하여 C언어와 같은 프로그래밍 언어로 작성된 텍스트파일을 처리할 때 백슬래쉬 (back-slash)나 null문자를 처리하기 위하여 만들어진 인코딩(Encoding) 방법이다. 이 3가지 한글 코드는 모두 옛 한글을 표현하기에는 역부족이다. 우선 조합형과 EUC-KR의 경우 그 태생적인 문제로 인하여 현대 맞춤법에 의거한 한글 처리에만 급급한 실정이며, 유니코드 역시 표현 가능한 모든 옛 한글을 표현한 것이 아니라 지금까지 문헌에서 발견된 문자들만을 코드화의 대상으로 삼았으므로 새로운 문자가 발견되면 코드셋에 추가하기가 어렵다.

4.2 Netscape에서의 한글 처리

Netscape는 GUI기반 위젯으로 상용의 Motif1.2.4를 사용하고 있으며, 모티프 위젯은 X11의 국제화 방식에 따르는 국제화 기능을 제공하고 있으므로, FontList에 의한 다국어 출력과 XIM을 통한 입력기를 별다른 프로그래밍 없이 해당 언어를 지원할 수 있는 기능을 가지고 있다. 즉, XIM을 따르는 외부 한글 입력기(Automaton)에 의해 한글을 입력 받으면 한글 입력기가 한글을 조합하여 해당 코드로 변환시켜 준 뒤 Netscape로 보내는 구조를 취하고 있다.

현재 Netscape에서는 EUC-KR과 ISO 2022-KR, UTF-8의 세가지 한글 코드를 표현할 수 있다. 그러나 조합형 문서와 통합 완성형 등의 코드로 이루어진 문서는 읽을 수 없고[3], UTF-8의 경우 UTF-8을 내부적으로 재해석하여 ISO 8856-1은 ISO 8859-1 글꼴을 통하여 표시하고, KS C 5601 영역은 KS C 5601-1987.0글꼴을 통하여 표현하다. 이러한 경우 유니코드로 작성된 텍스트 문서를 웹 브라우저에서 읽을 수 있다는 장점이 있지만, KS C 5601에는 있으나 유니코드에는 없는 문자의

경우 글꼴을 표현하지 못한다는 단점[3]이 있다.

한컴의 경우와 마찬가지로 EUC-KR과 ISO 2022-KR, UTF-8은 생성 가능한 모든 옛 한글을 표현하기는 불가능하다. 이것은 훈민정음으로 작성된 수많은 고문헌을 데이터베이스화 하여야 하는 국어학자와 고문헌 정보가 필요한 일반 사용자들에게는 상당한 불편 사항이다. 이에 정음형 한글코드[5]를 사용하여 개발된 X-Window용 소프트웨어가 개발되었는데, 각각 Hanterm과 Netscape에 대응되는 Hunterm[4]과 Hmosaic[6]이다.

5. 한글 코드와 리눅스

5.1 한글의 표현 범위

현재 리눅스에서 가장 많이 쓰이는 한글 코드는 KS C 5601-1987(현행 KS X 1001)이다. 이미 수많은 논의를 통하여 완성형 한글 코드의 문제점은 익히 알려져 왔는데, 모든 현대 한글의 표현 불가능, 언어처리의 어려움, 옛 한글 표현의 불가능 등이 대표적이다. 특히 옛 한글 처리에 있어서 완성형 코드는 많이 쓰이는 옛 한글인 순정음 비읍이나 아래아 등을 전혀 표현하지 못하여 문제가 더욱 심각하다. 이는 단순히 옛 한글을 입출력하지 못한다는 의미 뿐만이 아니라 고문헌 데이터베이스나, 국어학자들을 위한 환경이 마련되지 못하여 리눅스의 입지가 불리해 질 것이다.

유니코드가 일반화될 시점이면 리눅스에서도 유니코드를 지원할 것이고, 리눅스에서도 옛 한글 지원이 다소 수월해질 것으로 보인다. 그러나 유니코드에서도 옛 한글의 지원이 완벽하게 이루어지는 것이 아니라서[7,8] 소스가 공개된 리눅스에서 근본적인 대책 마련이 요구된다. 즉, 앞으로의 리눅스 한글 환경에서는 유니코드상의 240자 자소형[12]이나 정음형 한글 코드[4,5,6,8,11]를 사용하여 옛 한글 처리 및 언어 처리의 요구까지 동시에 만족시켜야 한다.

5.2 언어처리의 불편함

컴퓨터 응용 기술이 발전함에 따라 한글 정보 처리 기술은 기존의 단순 문자 처리 중심에서 언어 처리 중심으로 발전하고 있다. 언어처리는 문자로 표현된 자연언어를 처리하며 그 처리 대상은 바로 문자 코드이다. 기존 연구[4,5,9]에 따르면 한글은 음소 및 음절 문자 특성이 있으며, 이 두 가지 특성은 문자 구조 정보와 기본적인 언어 정보의 표현을 지원한다고 한다. 그러나 리눅스에서 가장 대표적으로 사용하는 완성형 한글의 경우 코드화의 대상을 단순히 2350개의 음절만을 삼았으므로 언어 처리는 매우 어렵다. 따라서 리눅스를 이용하여 언어 처리 소프트웨어를 개발하려면 우선 내부적으로 사용하는 코드체계를 바꾸거나 언어 처리용의 함수를 작성하여야 하는 불편함이 뒤따른다. 그러므로 리눅스가 클라이언트 및 서버로써 국내에서 자리 매김을 확실히 하기 위하여서는 언어 정보 처리가 수월하여야 한다. 이는 워드프로세서에서 맞춤법 검사를 하여주고 검색엔진에서 단어를 확장하여 검색하여주는 세계적인 추세를 보면 쉽게 짐작이 가능하다. 또한, Microsoft의 SAPI(Speech API)와 JAVA의 SAPI(Speech API)가 현재 상당부분 실용화에 성공하고 있는 것을 볼 때 국내 리눅스 관련 학계 및 업계에서도 기초적인 언어처리 라이브러리를 구축하고 SAPI와 같은 더욱 발전된 언어처리 라이브러리 구축이 필요할 때이다.

5.3 유니코드와 리눅스

현재 유니코드가 점차 보급되면서 리눅스에서도 유니코드를 지원하는 응용 소프트웨어가 늘어나고 있다. 대표적으로 UTF-8인코딩 방식을 지원하는 한컴과 Netscape가 있다. 유니코드는 한글 표현을 위하여 완성형 한글음절 11172자와 옛한글 표현을 위한 한글 자모, 호환용 한글 자모의 3가지 종류의 한글이 들어있다. 그런데 이 3가지

방식의 한글 코드를 사용하여 옛 한글을 표현하려면 유니코드의 구현 수준3의 조합방식을 사용할 때만 가능[11]하다. 구현 수준3에서는 조합 방식을 허용하므로 자소형과 자모형은 물론 완성형을 포함한 혼합된 부호계로 음절 구성이 가능하다.

위에서 언급한 것처럼 구현 수준3에서 3가지 부호계를 혼합하여 음절자를 구성하면 한 음절에 대하여 8종류의 표현이 가능하며, 여기에 완성형과 조합 표현까지 합하면 무려 13종류의 표현이 나온다.

상업적으로 유니코드를 사용하여 옛 한글을 입력할 수 있는 환경은 Microsoft Word2000에서 처음 제공되었는데, 구현 수준3의 13가지 표현 방법 중 어떤 방법을 사용했는지 알 수 없다. 이 문제는 또다시 한글 코드 문제를 야기할 수 있는데, 모든 소스가 공개되고 개발자와 사용자의 의사소통이 활발한 리눅스에서 X-Window와 glibc.so라이브러리 차원에서 해결하기 위한 노력을 기울여야 할 것이다.

(표 2) 유니코드에서 '한'의 3가지 표현

한글 종류	바이트 수	코드열
11172자 완성형	2바이트	0xD55C
240자 자소형	6바이트	0x1122 1161 11AB
51자 자모형	6바이트	0x314E 314F 3134

6. 결 론

본 논문에서는 리눅스에서 한글 인터페이스에 관하여 알아보았는데 과거에 비하여 지금은 많은 사람들의 노력으로 리눅스에서 한글을 처리할 수 있는 환경이 마련되었다. 하지만 리눅스에서 가장 일반적으로 사용되는 한글코드는 KS C 5601-1987 코드인데, 본문에서 언급이 되었듯이 코드체계에 많은 문제가 있다. 따라서 언어 정보 처리 환경을

구축하거나 옛 한글의 지원 등을 위하여 사용하는 한글 코드를 정음형 코드나 유니코드내의 240자 자소형 코드로 바꾸어야 한다. 이것은 소스가 공개되어있어 누구나 개발자가 될 수 있는 리눅스에서 가능하다.

또한, 리눅스는 무료, 소스 공개, 고효율이라는 장점을 내세워 빠르게 보급되고 있는데, 아직 일반인들에게 까지 널리 확산되지는 못하고 있다. 이는 리눅스라는 새로운 운영체제를 다시 배워야 한다는 부담감도 있지만 Microsoft사의 Windows와는 달리 응용 프로그램이나 메시지가 대부분 영어로 작성되어있기 때문이다. 또한, 각종 응용 소프트웨어에서 한글을 입력하기 위해서는 라이브러리를 패치하고 한글 입력 서버를 설치해야 하는 등 일반인들이 리눅스를 사용하는 데는 많은 부담이 따른다. 이를 위해서는 리눅스 개발자와 배포판 제작사, 최종 사용자들이 긴밀한 유대관계를 가지고 한글의 입출력 및 사용환경 개선에 노력을 기울여야 할 것이다.

앞으로 리눅스에서 바르고 정확한 한글의 표현이 가능하게 된다면 더욱 많은 사용자가 리눅스를 사용할 것이고 올바른 언어 정보 처리 환경을 구축하는데 많은 도움이 될 것이다.

참고문헌

[1] netscape과 hanIM,
<http://www.nizico.co.kr/software/hanIM/hanIM-ns.htm>, 미지리서치

[2] Hanterm HOMEPAGE, <http://elf.kaist.ac.kr/hanterm/>

[3] 최준호, Netscape와 한국어 처리
<http://www.kr.freebsd.org/cjh/freetime/osshangul/pse199908>,

[4] 금대식·변정용·함경수, 훈민정음이 살아있는 Hunterm의 구현, 한국정보과학회 봄 학술

발표논문집, 22권, 1호, pp. 937-940, 1995.[4]

[5] 변정용, 훈민정음 창제 원리의 공학화에 기반한 한글 부호계의 발전 방향, 한국정보과학회지, 12권, 2호, 1994.

[6] 강진곤·변정용, “언어정보처리에 기반한 한글 코드의 효율성 평가”, '96 봄 학술발표논문집, 제23권 제1호, 한국정보과학회, p953-956, 1996

[7] 변정용, KS C 5700-1995의 구현상의 문제점과 개선 방향, 한국 정보 처리학회 춘계 학술발표논문집 4권, 1호, pp. 387-392, 1997

[8] 변정용, “한글부호계 제정에서 반복되는 시행착오의 고리를 끊으려면”, 한글공학 특집, 정보처리학회지, 제5권 5호, p19-28, 1998

[9] MS의 확장완성형한글 코드 문제에 관한 컴의 입장-(1),
<http://www.hnc.co.kr/press/paper/hpaper09.htm>, (주)한글과 컴퓨터, 1995

[10] 그래픽 사용자 인터페이스 한글 처리 표준: XIM 프로토콜, <http://nca.nca.or.kr/data/standard/kisstd/k069s95.htm>, 정보통신부, 1995

[11] 변정용, “국제문자부호계에 포함된 한글 부호계의 개선 방향”, '98 가을 학술발표논문집, 제25권 2호, 한국정보과학회, p192-194, 1998

[12] 한국표준협회, 국제문자부호계 KS C 5700-1995, 1995

[13] 변정용, 한글정보처리를 위한 입력 라이브러리 연구 : 1,2,3차년도 시스템공학연구소, 1995, 1996, 1997

[14] <http://www.unicode.org/unicode/report/tr8.html>, The Unicode Standard Version 2.1, Technical Report#8, 1998

[15] Seamonkey Input Method Specification, <http://www.mizilla.org/projects/intl/input-method-spec.html>, The Mozilla Organization, 1998

[16] <http://www.linuxrouter.org>, Linux Router Project,

1999

[17] StarOffice, <http://www.sun.com/staroffice>, SUN, 1999

[18] GNU C Library-GNU Project-Free Software Foundation, <http://www.gnu.org/software/libc/libc.html>, GNU, 1999



변정웅

- 1980년 동국대학교 전자계산학과 (학사)
- 1983년 동국대학교 전자계산학과 (석사)
- 1994년 홍익대학교 전자계산학과 (박사)

1982년-1987년 한국전자통신연구소 연구원
 1995년-1996년 영국 요크대학 전산학과 객원교수
 1988년-현재 동국대학교 경주캠퍼스전자계산학과 부교수



최송목

- 1975년 경주고등학교 졸업
- 1983년 동국대학교 경상대학 전산학과 졸업
- 1982년-1987년 국민은행 근무
- 1987년-1995년 교보증권 근무
- 1989년 금융전산망 증권 소위원회 위원

1995년 한국교육미디어 상임 감사
 1996년 한국교육미디어 상임 감사
 1999년-현재 한국교육미디어 대표이사
 1982년 정보처리기사 1급
 1990년 기술지도사
 관심분야 : 한글교육과 정보처리