

한글 프로그래밍 언어(한글 포트란)의 설계에 관한 연구

정순영 허용도^o 백두권 황종선
고려대학교 전산학과

요 약

한글 프로그래밍 언어는 새로운 개념에 의해 설계되어야 하는데 그에 따른 많은 문제점이 발생하게 된다. 따라서 한글 프로그래밍 언어를 설계할 때 문제가 되는 한글의 구조적 특징, 동사가 문장의 마지막 뒤에 나오는 Postfix 적 특징, 한글의 특이한 데이터 구조 및 조사라는 첨가어가 붙으므로 인한 렉시칼 문제점등을 구조적으로 분석하고 여기에 맞는 한글 프로그래밍 언어를 개발하여야 한다.

그러나 이러한 모든 조건에 맞는 한글 프로그래밍 언어를 개발하는 것은 쉬운 일이 아니며 상당한 시간을 요하는 일이므로 본 논문에서는 과학기술 용에 가장 많이 사용되는 FORTRAN 언어를 택하여 일차적으로 그 명령어를 한글화하였다.

1. 서론

80년대에 들어오면서 컴퓨터 가격이 현저하게 저렴화되고 개인용 컴퓨터가 극산화되면서 국민학생들까지도 컴퓨터를 사용하기에 이르렀다. 아울러 CRT 와 프린터에서 한글사용이 자유롭게 되면서 하드웨어면에서는 한글을 사용하는데 큰 불편을 느끼지 않게 되었다. 그러나 운영체제와 프로그래밍 언어를 비롯한 모든 시스템 소프트웨어들이 국내에서 개발되지 않았기 때문에 컴퓨터와의 모든 대화를 영어로 한다는 점이 일반 대중에게 컴퓨터의 사용을 어렵게 하였다.

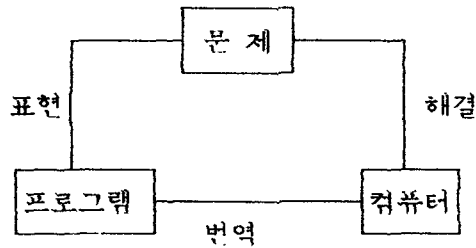
이에 따라 한글 프로그래밍 언어에 대한 연구들이 여러 방면으로 시도되기 시작하였다. 연구가 시도된 방면을 보면 우선 간결한 한글프로그래밍 언어를 구현하여 한글 프로그래밍 언어의 가능성을 조사하여 보려는 방향에서의 연구, 한글 프로그래밍 언어를 설계하는데 있어서 한글의 특성을 보유하면서도 Deterministic 한 파싱(Parsing) 이 가능한 한글 프로그래밍 언어 구문에 관한 연구, 그리고 개인용 컴퓨터의 기본 언어인 베이식 언어, 상업용 언어인 코볼 언어에 대응하는 한글베이식언어 및 한글코볼언어를 설계함으로써 한글 프로그래밍 언어를 실용화시켜보려는 연구등이 있었다.

그러나 이러한 한글 프로그래밍 언어들은 전문가들에게나 연구용으로서의 가치가 있을 뿐이며, 또한 이제까지의 컴퓨터 교육도 전문가 양성을 위한 교육이었으므로 현시점에서 대중화되고 보편적인 이용 수단의 컴퓨터 교육내지는 한글 프로그래밍 언어에 있어서는 상당히 낙후되었던 것이 사실이다. 그러므로 본 연구에서는 과학기술용으로 가장 많이 사용되는 FORTRAN 언어를 한글의 특성에 맞추어 한글화하였다.

2. 한글 프로그래밍 언어에 관하여

2.1 한글 프로그래밍 언어의 설계방법

일반적인 프로그래밍 언어를 Karl Buber의 언어 기관 모형에 적용시켜 보면 그림 1과 같다.



< 그림 1 > 프로그래밍 언어 모형

즉, 컴퓨터의 사용은 문제 해결에 있으며 해결을 위한 도구로써 프로그래밍 언어가 존재하고 이 도구에 의해 문제를 표현하는 과정이 프로그래밍이며 문제를 표현한 것이 프로그램이다. 그리고 이 표현된 프로그램을 컴퓨터가 알아볼 수 있는 언어(기계어)로 바꾸어 주는 작업이 번역이며 이 작업은 컴파일러(Compiler)라는 번역기에 의해 행해진다. 이 번역된 기계어를 컴퓨터가 수행하여 문제 해결의 목적을 달성한다. 프로그래밍 언어 설계에서는 그림 1의 세 과정이 효과적으로 이루어 질 수 있는 언어를 설계하는데 목표를 두어야 한다.

그런데 최근 반도체 기술의 발달로 컴퓨터 가격이 싸지고, 또한 컴퓨터가 대중화되어 가는 현실에서는 무엇보다도 문제를 효과적으로 표현하는 과정이 중요하다. 따라서 문제 표현 도구인 프로그래밍 언어를 설계할 때에는 제반 환경을 고려하여 문제를 직접적이고 친숙하게 표현할 수 있어야 하고 언어의 개념이 명확하여 언어의 질을 높일 수 있게끔 잘 고려하여 설계하여야 한다. 따라서 구조적 프로그래밍, 단계적 문제 기술을 위한 세분화등이 쉽도록 되어야 한다. 그리고 될 수 있는 한 언어를 간단하고 쉽게 설계하여 사용자가 배우기 쉽고 또 필요에 따라 언어를 쉽게 확장할 수도 있어야 한다 또한 여러 프로그래머가 공동으로 프로그램 할 때 문제를 쉽게 분리하여 프로그램할 수 있어야 하며 또한 프로그래머 상호간에 대화를 최소화하면서도 프로그래머는 효과적이고 쉽게 프로그램을 할 수 있어야 한다.

한글 프로그래밍 언어는 인공언어로서 자연언어인 한글과는 그 구조 자체가 상당히 차이가 있으나 한글 프로그래밍 언어 자체가 한글을 기반으로 하여 한글의 구조를 반영하면서 필요에 따라 그 구조를 변형시켜 설계되어야 하는 만큼 한글 프로그래밍 언어의 구조 분석 방법이 자연어에 가까워야 할 것이다.

2.2 한글 프로그래밍 언어의 문제점

컴퓨터의 이용이 일반화됨에 따라 컴퓨터의 사용자는 전문가보다는 일반 사용자가 주류를 이루게 되며 컴퓨터에 대한 교육이 대학이상의 과정에서 중, 고등학교 및 국민학교등으로 그 범위가 확대되어질 것이 분명하다. 따라서 이들이 컴퓨터를 쉽게 이해하고 사용하기 위해서는 많은 응용 프로그램이 개발되어야 할 것이며 또한 영어로 작성된 기존의 프로그래밍 언어보다는 우리에게 친숙하고 쉽게 터득할 수 있는 한글 프로그래밍 언어의 개발이 필요한 실정이다. 그러므로 전문가에게는 이미 습득한 지식의 한국적 표현, 즉 제약없는 한글 정보의 사용이 필요하고 비전문적 일반 대중에게는 한글화된 컴퓨터 환경에서 보다 친밀한 의사 소통이 가능한 여건

을 마련해 주어야 할 필요성이 대두되었다. 이런 필요성에 부합하여 최근 컴퓨터 시스템 및 컴퓨터 사용환경에서의 한글화는 많은 관심과 연구를 통해 발전되어 왔다. 그러나 한글 프로그래밍 언어의 개발에는 상당히 많은 문제점이 따른다.

첫째, 한글 프로그래밍 언어의 선결문제는 한글 입출력이다. 한글은 영어와는 달리 모아쓰기를 하므로 기존 영어의 입출력에 사용하는 입출력장치의 사용에는 많은 불편한 점이 있다. 따라서 한글 입출력에 알맞은 입출력장치의 개발이 선행되어야 한다.

둘째, 한글 입출력장치의 개발과 아울러 이를 보조하는 소프트웨어가 개발되어야 제기능을 발휘할 수가 있다.

셋째, 한글 프로그래밍 언어와 이를 위한 시스템 개발에는 막대한 비용이 소모되는 바 이로 인해 비롯되는 장점과 개발에 사용된 비용을 면밀하게 검토하여야 한다.

넷째, 한글 프로그래밍 언어는 과연 프로그래밍 하기에 적당한가? 한글은 반드시 초성, 중성, 종성으로 구성되므로 한음절이 3개의 기본 자모로 되어있다. 따라서 이는 현 프로그래밍 언어가 부호화하는 추세에 있는 것으로 볼 때 과연 한글도 부호화하여 사용하는 것이 편리한가? 한글은 문장의 의미가 조사 및 접미사에 의해 변화를 보이며 중요한 단어는 뒤에 나타나는 경향이 있는데 이것은 한글언어의 설계시 어떻게 반영될 것인가.

다섯째, 한글만을 명령어로 사용하는 경우 기존의 프로그램이나 외국의 프로그램을 활용하는데 계약을 받으므로 한글 및 영어 명령어를 동시에 사용하는 것이 바람직하다. 이 작업은 전문적인 프로그래머에게는 좋은 디버깅의 도구가 된다. 즉, 한글이나 영문으로 서로 바뀌지 않는 부분은 잘못 입력된 것이므로 키편치상의 문법오류를 한꺼번에 쉽게 찾아낼 수가 있을 것이다.

여섯째, 한글 프로그래밍 언어 처리시스템 완성후 이의 판매시장이 얼마나 넓을 것인가? 시스템의 개발 및 생산은 막대한 비용이 소모되므로 이들 소모된 비용을 전부 환수할 수 있을 것인가? 또한 한글의 수출은 가능할 것인가?

이상과 같은 문제점이 있지만 그래도 한글은 우리 고유의 말이며 문자이다. 따라서 한글 언어의 개발은 민족의 자주성, 컴퓨터의 보급, 컴퓨터 산업의 발달, 컴퓨터의 보급이 생활에 주는 이득, 행정전산화로 인한 행정적 간소화 등의 효과를 생각하여 반드시 개발되어야 한다.

2.3 기존 프로그래밍 언어에서의 한글 입출력

1970년대까지는 고급 프로그래밍 언어에서의 한글 자료처리가 불가능하였으나 국내에서 컴퓨터가 상업용 업무에 사용되면서 보고서에 한글 자료의 출력이 크게 요구되었다. 이에 따라 1970년대 초 고급 언어에서의 한글 자료의 입출력 방법이 개발되었다. 이 방법은 라인프린터 드럼의 소문자 위치에 한글 폰트를 넣은 기법으로 입력은 한글 표시의 특수 코드 다음에 나오는 영문자를 한글 자모에 대응시켜 풀어쓰기 형식으로 입력하였고 출력은 모아쓰기 부프로그램을 호출하여 n 바이트형 코드 즉, 초성, 중성, 종성 세 레코드로 구성되는 라인프린터용 한글 이미지로 변환하여 인쇄하는 방법이었다.

한편, 도트 매트릭스 프린터가 사용되면서 개인용 컴퓨터에서도 한글처리 방법이 발전하였다. 다시말해 도트 매트릭스 프린터가 16*16 또는 24*24 점으로 문자를 표현하기 때문에 한글 한 음절에 대응되는 이미지를 기억시켜 인쇄할 수 있게 되었다. 따라서 입력장치인 CRT에서도 한글 풀어쓰기 입력코드를 받아 자체에서 2바이트 완성형 또는 2바이트 조합형으로 코드를 변환시켜 주어 처리가 되기 때문에 고급 언어에서의 한글 자료처리가 자연스럽게 해결된다.

2.3.1 코볼언어에서의 한글 입출력

현재 국내에서 코볼언어의 사용은 대부분 중형이상의 컴퓨터 시스템에서 이루어지기 때문에 입력은 기존의 영문 키보드를 출력은 라인프린터를 사용한다. 코볼언어에서 한글 입력은 한글 풀어쓰기 형태와 대응되는 영문코드로 입력되므로 초성, 중성 및 종성이 각각 한 바이트씩 할당되어 입력된다. 그러므로 입력 기억장소의 필드는 한글 풀어쓰기의 갯수만큼 할당되어야 한다. 또한 한글 문자의 출력을 위해서는 작업영역절에서 모아쓰기 표현이 가능한 기억장소를 배정한 후 한글 모아쓰기 부프로그램을 호출하여 모아쓰기 형태로 저장한다.

일반적으로 코볼에서는 라인프린터를 사용하므로 한글 출력 명령문은 초성과 중성, 종성을 각각 순서에 따라 세번을 인쇄하는 일련의 명령문으로 이루어진다. 특히 중성의 인쇄는 초성이 인쇄된 후 그 인쇄된 초성과 같은 줄에 인쇄할 수 있도록 하는 특수문자를 사용하여 중복 인쇄를 하여야 한다.

2.3.2 포트란에서의 한글 입출력

포트란 언어에서의 한글 이용은 문자자료를 선언하는 character 선언문을 이용한다. 한글 문자를 처리할 수 있도록 하기 위하여 한글 모아쓰기를 위한 부프로그램을 제공한다. 포트란에서의 한글 자료 처리 역시 코볼과 비슷하지만 호출하는 형식과 부프로그램 이름이 컴퓨터가 다른 경우는 전혀 다르다. 또한 동일 기종일 때에도 대부분의 경우 코볼언어에서 주어진 방법과 상이하여 포트란 한글 부프로그램의 호출형식을 별도로 숙지해야만 한다. IBM-PC의 호환기종의 경우 한글 사용은 2 바이트 조합형을 사용하여 한글을 기존 자료처럼 그대로 사용할 수 있다.

2.3.3 PASCAL 언어에서의 한글 입출력

PASCAL 언어 역시 한글 문자의 입출력을 위해서는 모든 기종에서 한글 변환 라이브러리 프로그램이 필요하겠지만, 현재의 컴퓨터 기종에서는 한글 자료를 처리할 수 있도록 하는 PASCAL용 한글 처리 부프로그램을 제공하는 것이 거의 없는 실정이다. 그 이유는 일반적으로 대형 컴퓨터 기종이나 라인프린터를 위주로 하여 작업을 수행하는 업체에서는 PASCAL 언어가 거의 쓰이지 않는 실정이며, 현재 사용하고 있는 곳은 각 대학에서나 교육용으로 사용되고 있고 16-BIT 퍼스널 컴퓨터나 UNIX 환경하에서 사용되고 있기 때문에 PASCAL 컴파일러를 공급하는 공급자 입장에서 사용자 측에서 PASCAL 용 한글 처리 부프로그램을 요구하고 있지 않으므로 대부분의 기종에서 제외되고 있는 상황에 있다. 그러나, IBM-PC 호환 기종에서는 한글 2바이트 조합형을 사용할 수 있어 CHARACTER 형 PACKED ARRAY를 선언하여 사용하면 FORTRAN에서와 마찬가지로 한글을 사용할 수 있다.

2.3.4 PL/I에서의 한글 입출력

PL/I 에서의 한글 자료 입출력 역시 포트란, 코볼 언어에서와 마찬가지로 사용되고 있으며 그 구체적인 사용방법은 기종마다 다른 방법으로 한글 자료 처리 부프로그램을 호출하고 있다. PL/I 언어는 포트란 언어와 코볼 언어의 특성을 모두 가지고 있어 한글 모아쓰기의 부프로그램 형식은 이들 언어와 유사한 형식을 갖추고 있으며 시스템 내의 부프로그램 루틴은 서로 다르게 사용될 수 있다. 또한 PL/I 언어의 국내 이용이 이들 언어와 같이 널리 사용되고 있지 않기 때문에 컴퓨터 기종에 따라서는 해당 한글 처리 부프로그램이 제공되어 있지 않은 것도 있다.

2.3.5 C언어에서의 한글 입출력

한글 입출력을 제공하기 위한 부프로그램이 제대로 제공되어 있지 않은 실정이며 그 이유 역시 파스칼과 거의 유사하며 특히 이 C 언어는 고급 프로그래밍 기술을 요구하는 분야, 즉 시스템 프로그래밍이나 소프트웨어 개발의 도구로 사용되고 있어 한글 처리 및 입출력 요구가 크지 않은 편이다. 그러나 이러한 이유가 있더라도 궁극적으로 국내에서 사용되고 있는 모든 프로그래밍 언어에서는 한글 자료처리가 가능하여야 된다. 한편 IBM-PC 호환기종에서는 C 언어 역시 파스칼 언어처럼 한글 입출력이 2바이트 문자형으로 쉽게 처리된다.

2.3.6 ADA 언어에서의 한글 입출력

현재 ADA 언어는 컴파일러가 널리 보급되지 않아 아직 한글 처리 루틴이 만들어져 있지 않은 실정이지만 곧 ADA 언어가 보편화될 것에 대비하여 한글처리에 관한 기준이 요구된다. 그렇지 않으면 코볼이나 포트란 언어와 같은 개념으로 한글 처리 부프로그램들이 제공될 것이다. 더욱이 이 ADA 언어의 컴파일러를 국내에서 구현한다면 한글 자료 처리에 관한 기준이 더욱 요구된다.

2.3.7 베이식 언어에서의 한글 처리

현재 베이식 언어에서의 한글 자료처리는 8비트 컴퓨터인 경우는 CALL 3327 방식의 한글 소프트웨어를 창작하여 그래픽 모드 상태에서 처리하는 방법과 IBM 퍼스널 컴퓨터 호환 기종에서 처럼 2바이트 한글 코드를 사용하는 방법, 대형 컴퓨터에서의 한글 처리 부프로그램을 호출하여 사용하는 방법등의 여러가지 형식이 사용되고 있다. 베이식 언어에서의 한글 자료 처리는 변수에 저장되는 문자에 대한 길이의 제한을 프로그래머가 정의하는 것이 아니라 시스템에서 이미 결정해 놓은 상태로 이용되고 있으며 자료의 입출력도 특별히 그 형식을 지정하는 선언이나 명령이 사용되지 않는다는 이유에서 앞의 다른 프로그래밍 언어보다는 비교적 한글자료의 사용이 편리하다고 볼 수 있다. 그렇지만 대형 컴퓨터에서 한글 자료 사용은 역시 다른 프로그래밍 언어에서와 마찬가지로 한글 처리를 위해서는 특별한 선언과 형식을 가지고 있어 다른 프로그래밍 언어보다도 더욱 혼란의 여지가 있다고 할 수 있다.

2.4 한글 프로그래밍 언어의 구현단계

한글 프로그래밍 언어를 처리하기 위한 가장 기본적인 요소는 한글의 입출력이다. 한글의 입출력이 자유롭다고 가정하고 한글 프로그래밍 언어의 구현단계를 살펴보면 다음과 같다.

- 제 1단계 : 기존의 프로그래밍 언어와 한글을 일대일 대응시켜 기존의 인터프리터 (Interpreter), 컴파일러(Compiler), 및 소프트웨어를 사용하는 단계
- 제 2단계 : 한글 프로그래밍 언어의 정의는 영어 언어와 일대일 대응이나 수행 (Execution) 에 있어서의 내부조직을 한글에 맞게 개편하는 단계
- 제 3단계 : 한글 프로그래밍 언어의 정의는 같으나 영어로의 전환이 직접 기계어를 발생시키는 컴파일러(Compiler)를 설계하는 단계
- 제 4단계 : 한글의 문법에 맞는 기계어를 갖는 처리기의 개발단계
- 제 5단계 : 한글의 문법에 맞는 프로그래밍 언어 자체가 기계어인 하드웨어를 개발하는 단계

3. 한글 포트란 언어의 설계

3.1 FORTRAN 식 고급언어

FORTRAN 은 최초로 개발된 고급언어로 과학계산을 위해서 만들어진 언어이며 약간의 결점에 도 불구하고 현재까지 많이 사용되고 있는 실정이다. FORTRAN 언어의 특징은 언어자체의 구조가 간단하여 예약어가 극히 제한된 수에 지나지 않아 배우기 쉽고 사용하기 쉽다는 점이다.

FORTRAN 식 한글 고급 언어는 다음과 같이 구성되었다. 언어 자체의 특성은 변화시키지 않은 채 영어로 된 예약어와 영어로 쓰게 되어 있는 사용자 정의 변수들을 한글로 바꾸어 입력시켜 이것을 전처리기를 통해서 영문 포트란 언어의 명령어로 바꾸는 형태이다. 이때 각 변수들의 내부표현 특히 스트링 변수에 관한 변환은 규정을 하지 않는다.

3.2 한글포트란 언어의 어휘 및 코딩

한글 포트란 문장은 크게 할당문, 제어문, 입출력문 및 선언문들로 나눌 수 있으며 이들 문장에 사용되는 연산자를 제외한 일반적인 예약어들은 그림 2 와 같다.

또한 영문 포트란에서 사용하는 특수기호(보기 : +, -, *, /, ...)들은 그대로 사용하며 그 밖의 모든 문장형식도 영문 포트란의 어법을 그대로 따른다. 단, 사용자가 정의하는 변수들의 첫자는 모두 한글 자소로 시작하며 영문 포트란에서는 첫자가 I, J, K, L, M, N 등의 문자로 시작하면 정수형 변수이고 그 밖의 다른 문자로 시작되면 실수형 변수로 간주하지만 한글 포트란에서는 특별히 선언되지 않는 한 모든 변수는 실수형으로 간주한다.

FORTRAN	한글	FORTRAN	한글
IF	만일	THEN	이면
GO To	가기	CONTINUE	계속
DO	반복	FORMAT	서식
STOP	멈춤	EXTERNAL	외부
FUNCTION	함수	SAVE	저장
SUBROUTINE	부루틴	CHARACTER	문자형
DIMENSION	배열	OPEN	열기
COMMON	공유	CLOSE	닫기
REAL	실수	INQUIRE	문의
RETURN	돌아가기	CALL	호출
END	끝	PAUSE	중단
READ	읽기	I	정
WRITE	쓰기	F	실

< 그림 2 > 한글 예약어와의 비교

3.3 한글 포트란 예제 프로그램

(1) 반복문

(한글 프로그램)	=====>	(영문 프로그램)
합 = 0		SUM = 0)
반복 10 가 = 1 , 10		DO 10 I = 1 . 10
합 = 합 + 가		SUM= SUM + I
10 계속		10 CONTINUE

(2) 입출력문

(한글 프로그램)	=====>	(영문 프로그램)
읽기(6,10) 이름, 점수		READ(6,100) NAME, ISCORE
10 서식(문6,정2)		10 FORMAT(A6, I2)
쓰기(1,20) 이름, 점수		WRITE(1,20) NAME, ISCORE
20 서식(문6,2칸,정2)		20 FORMAT(A6, 2x, I2)

(3) 제어문

(한글 프로그램)	=====>	(영문 프로그램)
만일(점수>100) 가기20		IF(SCORE.GT.100) GOTO20
쓰기(1,10)		WRITE(1,10)
10 서식("데이터오류")		10 FORMAT("DATA ERROR")
20		20

4. 결론

한글 고급언어 처리를 위하여 가장 필수적인 것은 한글의 입출력 방식이다. 현존 시스템에서 한글의 입력방식과 출력방식은 높은 수준에 와 있지는 않다. 따라서 한글 고급 언어를 위하여서는 먼저 한글의 입출력이 개선되어야 한다. 또한 한글 문법은 영어와는 달리 조사와 접미사가 문장에서 중요한 역할 즉, 영어의 전치사에 해당하는 일을 하며 동사가 뒤에 나오는 것이 특징이다. 따라서 본 논문에서 예를 보인 것 처럼 영어 고급 언어를 한글로 번역하여 사용하는 것은 약간의 무리가 따르나 사용가능함을 보여주었다.

따라서 한글 포트란 언어의 궁극적인 목적은 우리나라 실정에 맞는 좋은 프로그래밍 언어를 개발해서 프로그래머 들이 프로그램 작성에 효율을 갖게 하려는데 있다. 이러한 국내 실정에 맞는 좋은 프로그래밍 언어를 설계해서 구현하는 작업은 상당히 어려운 작업이므로 본 논문에서는 일차적으로 영문 포트란의 명령어들을 한글화 하였다. 앞으로 본 논문의 2.4절에서 설명한 프로그래밍의 구현단계에 맞는 좋은 한글 프로그래밍 언어가 만들어진다면 기존의 프로그래밍 언어보다 더 효율적이고 실용성있는 한글 프로그래밍 언어가 될 것이다.

<참고문헌>

1. 과학기술처, "한국정보처리 표준화 연구", 1986.7
2. 과학기술처, "한국정보처리 표준화 연구(3차년도)", 1988.7
3. 과학기술처, "OS 한글화 지침서 개발에 관한 연구", 1987.7
4. 김영택외 4인, "한글 프로그래밍 언어 설계에 관한 연구", 정보과학회 논문지 VOL. 11, NO2, PP81-101, 1984
5. 이진태, "한글 프로그래밍 언어의 설계 및 구현", 한국과학 기술원, 석사논문, 1983
6. 원유현, "한베이식 언어의 설계 및 명령어 처리에 관한 연구", 홍대 논총 제 15집, PP259-267, 1983
7. 한영국, 한글 프로그래밍 언어의 설계 및 구현에 관한 연구", 홍익대학교 석사논문, 1983
8. 오길득, 박세영, "한글 정보처리를 위한 기초연구", 정보과학회지 2권 4호pp7-19, 1984.12.
9. 허용도, "한글 프로그래밍 언어의 설계에 관한 연구", 고려대학교 석사논문, 1988
10. 원유현, 프로그래밍 언어 개념, 정익사, 1983
11. Pratt, "Programming languages(Design and Implementation)", 1984
12. Phn zorrella" Language Translators", 1982
13. Ellis Horowitz, "Fundamentals of Programming Languages", 1984