

교육용 한글 파스칼 설계 및 구현

正會員 김 용 성* 正會員 홍 성 수** 正會員 심 재 홍***

Design and Implementation of Educational Hangul Pascal

Yong Sung KIM*, Seong Soo HONG**, Jae Hong SIM*** Regular Members

要 約 본 논문은 터보 파스칼 언어에 대응하는 한글 파스칼 언어를 설계하고 구현한 것이다. 한글 파스칼 명령어는 초·중·고 학생들이 쉽게 이해할 수 있는 용어로 선정하였으며, 도스(DOS) 명령과 오류 메시지도 한글화 하였으며, 한글 파스칼은 IBM PC XT / AT, MS DOS 상태에서 C 언어로 구현하였다.

ABSTRACT In this paper, we present a design and implementation of the Hangul pascal programming language corresponding to Turbo pascal. The instructions of Hangul-pascal were easily selected to be understood of elementary middle, high school students, and DOS commands and error messages were translated into Hangul. Hangul-pascal were implemented into C-language under IBM PC XT / AT, MS DOS.

1. 서 론

세계 각국은 부가가치가 많은 정보 산업을 육성하기 위해서 컴퓨터와 통신에 관련된 산업에 주력하고 있으며, 이를 뒷받침하기 위한 컴퓨터 문맹탈피(Computer literacy) 목적으로 초·중·고에 컴퓨터 교육을 활성화 하는 방안을 실행 또는 연구 중이다. 그러나 컴퓨터 교육에 대한 연구, 계획, 방법들은 아직 초창기 단계에 있는 것이 오늘의 현실이다. 컴퓨터 문맹탈피를 위한 가장 중추적이고 중요시 되는 분야는 프로그래밍 분야이다. 왜냐하면 컴퓨터 교육은 프로그래밍을 통해서 컴퓨터를 이해하고, 문제 해결력을 기르며, 논리적 사고와 창의력을 개발할 수 있기 때문이다. 또한 프로그래밍을 통해서 프로그래밍 언어의 개념과 프로그래밍 방법론 등의 기초 개념을 습득하기 때문이다.

우리나라는 이미 2백만대 이상의 크고 작은 컴퓨터가 보급되었고, 90년도부터 통신공사가 교육용 컴퓨터를 초·중·고에 무상으로 보급하여 95년도에는 각 급 학교에 컴퓨터 교육을 할 수 있는 환경을 제공하게 될 것이다. 이러한 정부의 컴퓨터 교육 활성화로 인하여 영어를 잘 모르는 국민 학생이나 일반 사회인들이 손쉽게 사용할 수 있는 한글 프로그래밍언어가 어느 때보다도 절실히 크게 요구되고 있는 실정이다.

그러나 컴퓨터 교육을 위해서 설계된 프로그래밍언어는 극소수에 불과하고, 대부분이 교육적 측면이나 언어학적 측면에서 부족한점이 많으며 국내의 환경에 적합하게 활용할 수 있는 언어는 거의 없다.⁽¹⁾

즉, 한글 프로그래밍 언어는 한국인이 보다 쉽게 컴퓨터와 대화할 수 있게 설계 되어야 하며, 한국인이 한글을 사용할 때 선택된 키워드가 친밀해야만 한다. 또한 한글 문장 구조의 post-fix 라는점과 기존의 파싱기법에 의해 컴퓨터로 처리하는데 어려운점 등과 같은 프로그래밍 언어의 일반적인 요건들을 고려해서 설계해야만

* 全北大學校 電子計算學科
Dept. of Computer Science, Chonbuk Nat'l Univ.
** 湖西大學校 컴퓨터工學科
Dept. of Computer Eng., Hoseo University
*** 光六大學校 電子計算學科
Dept. of Computer Science, Kwangwoon University
論文番號 : 91-95(接受1991. 8. 5)

한다. 그런데 이와 같은 요건을 모두 갖춘 한글 프로그래밍 언어를 설계하고 구현하는 것은 매우 어려운 일이다. 현재까지 한글 프로그래밍 언어 설계와 구현에 관한 연구는 많은 진척이 있었으나 한글이 구조적 특징을 잘 반영하지 못했을 뿐만 아니라 구조적 프로그래밍을 수행하기에 불편하였다. 이에 따라 최근에 한글 프로그래밍 언어에 대한 연구들이 활발하게 시도되었다. 그 중 중요한 내용은 한글 프로그래밍언어를 설계하는데 조사나 이미 변환을 사용하여 한글 언어구분을 직성하는데 많은 도움을 준 연구¹⁾, IMSAI용 베이직 언어를 한글로 대체하는 연구²⁾, Apple 기종에서 베이직 언어를 한글 명령화하여 한글모스 및 오유 매체지를 한글화 한 연구³⁾, 교육용 한글 프로그래밍언어 CELL의 설계에 관한 연구⁴⁾ 등이 있다.

본 논문은 컴퓨터 교육적 측면과 전산학적인 측면에서 유익한 구조적 프로그래밍 언어인 한글 파스칼 언어를 설계하고 구현하여 컴퓨터 마인드를 확산할 수 있는 교육적인 환경을 제공하는데 그 목적이 있다.

2. 한글 파스칼 언어의 문법 표현

2-1. 파스칼 언어의 배경과 특징

파스칼은 ALGOL의 특징을 계승하여 Niklaus wirth가 주로 교육용 프로그램을 위해 개발한 절차형(procedural)언어로 소형 컴퓨터에서 많이 활용되고 있다. 파스칼은 블럭 구조, 프로시쥬, 배열, 포인터, 리커시브 함수 및 간단한 입·출력 기능을 갖는 비교적 작은 언어로 파스칼 언어의 특징은 다음과 같다.

첫째, 새로운 데이터 형을 정의할 수 있는 기능과 한정된 스칼라 및 정수의 범위를 갖는 변수를 정의할 수 있고, 둘째, 내이니 형으로서 유한 스칼라(Scalar)의 멱집합(Power Set)이라는 점이다.

셋째, 모든 명칭(identifier)은 그 명칭을 사용하기 전에 선언 해야 한다. 즉, 파스칼은

ALGOL이나 PL/I와 같은 블럭 구조, PL/I과 같은 automatic storage인 heap이라는 일종의 based storage를 갖고 있어 포인터가 선언되면 그 포인터가 지칭하는 값의 형도 선언되어야만 하므로 묵시적 선언은 없다. 기본 데이터형으로 char, int, real, boolean, scalar 및 subrange형 등이 있으며, 구조적 데이터 형에는 배열, 레코드, 유니온, 집합(set), variant레코드 및 화일 등이 있다. 제어 구조는 배정문과 프로시쥬 call이 있고, 복합 제어 구조는 begin-end, for문, while 문, repeat문, if then else문 및 case문 등이 있으며, 입·출력은 builtin함수나 procedure로 이루어 지는 것이 특징이다.

2.2. 한글 파스칼 문법 표현

한글 파스칼 프로그래밍 언어는 프로그래밍 언어의 기본 요건을 만족하면서, 한글의 구조적 특징을 반영하여, 읽기 쉽고, 사용하기 쉬우며, 지언스리쉬야 한다는 점에 중점을 두었다.

또한 기존 컴파일러 기법을 사용하여 컴파일러 쉽게 작성 수 있도록 한다. 따라서 한글 파스칼 프로그래밍 언어는 한글의 언어적 특징을 잘 반영하면서 CFG(Context Free Grammer)로 표시할 수 있도록 설계·구현되도록 하였다.

한글 파스칼 언어의 문법표현은 조사를 명사등과 같이 문법 사용하는 문법 규정언어(Meta language)를 원칙으로 하나 문법규정언어는 애매 모호하기 쉽기 때문에 사용자가 쉽게 이해할 수 있도록 일부 조사에 제한을 두었다.⁵⁾

한글의 기본 문법 구성은 명령어가 뒤에 오는 post fix라고 할 수 있는데, 한글을 post fix 형태로 사용할 수 있으나⁶⁾ 이 경우 기존 파싱 방식을 사용한 경우⁶⁾ 파싱이 쉽지 않고 사용하기가 어려우므로 될 수 있는 한 앞 부분에 키워드를 주어 파싱과정에서 발생하는 문제점을 제거 하도록 하며 한글 파스칼에서 사용되는 일반적인 문장 구성에 관해서 기술하면 다음과 같다.

한글 문장의 기본 구조는 <주부>+<서술부> 형태로 구성된 수 있다. 그러나 한글 파스칼 언어의 문장은 컴퓨터에게 명령하는 명령이므로

주부는 거의 대부분이 생략되므로 이를 생략하며, 서술부는 <관형어>+<목적어>+<부사어>+<동사>를 기본 구조로 한다.⁽¹⁾ 서술부의 구성중 <목적어>는 <명칭>+<조사>의 구조로서 <목적어> 부분에 키워드가 나타나지 않으므로 기존의 파싱 방법을 사용할 경우 파싱 테이블 크기가 증가할 뿐만 아니라 구성이 어렵고 복잡해지기 때문에 한글 파스칼의 기본 문법 구조는 서술부의 구조를 변형해서 키워드가 맨앞에 나오도록 하는 방법을 채택했다.⁽¹⁾ 그 방법 중에 하나로 컴퓨터가 행하는 행위를 일반적으로 실행[하다]라는 동사로 바꾸어 준다. 또한 기존 파싱 방식을 그대로 사용하기 위해서 목적어의 <조사> 부분을 PASCAL 명령어에 국한하여 ':='에 (of),에서 (to)으로 (with)까지 (until)까지 실행 (do)등으로 대체 함으로서 사용자가 사용하기에 자연스럽게했다. 그러므로 이러한 문장 구조는 <부사>+<목적어>+<부사어>+<동사> 형태가 되어 키워드가 문두에 나오게 되므로 목적하는 바를 이룰 수 있다.

<예>

```
가 : =1에서 데이터수 까지 실행
<목적어>+<부사어>+<부사어>+<동사>
```

복합문(compound statement)는 문장의 앞부분이 <관형어>+<주어>의 형태로 한글 프로그래밍 언어의 특이한 구조이나 한글 파스칼 언어 복합문은 구성할 때 CFG의 특징인 Self-embedded한 특성을 이용하여 문장 구조를 만들었으며, 복합문마다 다음 키워드로 '끝'을 사용하여 문장의 범위를 알기 쉽게 했다.

<예>

```
선택 미지수가
0 : 피보 : =1 :
1 : 피보 : =1 :
    아니면 피보 : =피보(미지수-1)+(미지수-2) :
끝 :
```

3. 한글 파스칼 언어 설계

프로그래밍 언어를 설계하려면 먼저 언어 사용 목적을 정의하고, 이에 따라 필요한 자료구조, 연산, 프로그래밍 구조 등을 규정하며, 그 언어의 전체적 특성을 고려해서 설계 해야 한다.

한글 파스칼언어 설계의 주요 관점은 먼저 사용자가 배우기 쉽고 프로그래밍 언어의 일반적 요건에 알맞는 언어로 한글의 구조가 잘 반영되어, 비교적 자연언어와 유사하여 이해하기 쉽게 한다. 또한 한글 파스칼언어 설계 과정에서 발생할 수 있는 문제점을 쉽게 해결할 수 있도록 기존의 컴파일러를 사용하여 컴파일할 수 있도록 한다. 설계 내용은 다음과 같다.

3-1. 선언부

선언부는 데이터를 규정하는 부분으로 변수의 범위에 대한 정보, 연산의 수형, 데이터의 자료구조 선언 등을 기술하는 부분이다. 선언부의 구성은 먼저 라벨과 상수 선언을 할 수 있게 했으며, 그 다음에 문제의 데이터를 사용하여 문제 해결을 위한 변수 선언부 그리고 부프로그램이나 함수문 선언 순으로 되어 있고, 모든 변수는 변수 선언부에서 선언되어야만 사용할 수 있는 파스칼 언어처럼 strongly typed 형태를 취했다.⁽²⁾

<예>

```
상수 값=100 : // 상수 선언 //
형 테이블=배열 {0..100}의 경우 // 형 선언 //
변수 가, 나=테이블 : // 변수 선언 //
함수 비교하기(다, 라 : 정수) : 문자 : // 함수 선언 //
절차 이분법(변수 다 : 테이블 : 변수 가, 나 : 정수) // 부 프로그램 선언 //
```

3-2. 연산문

일반적으로 연산은 그 주어진 자료형에 따라 정해진다. 따라서 자료에 따른 연산을 직접 선언해 줄 수 있는 abstract data type 형태가 가장 바람직스럽다. 한글 파스칼의 연산은 이러한 abstract data type을 채택하여 기본적인 데이터

형에 따라 처리할 수 있게 했다. 산술문을 구성할 때 수학에서 일반적으로 사용하는 연산자를 사용했으며, 문장과 문장은 세미콜론으로 구별했다. 각 데이터형에 따른 연산순서는 아래와 같고, 각 연산자 사이에 우선 순위가 존재 한다.

- 1) 함수
- 2) 괄호
- 3) NOT
- 4) *, /, DIV, MOD, AND
- 5) +, -, OR
- 6) 관계 연산자 (=, >=, <=, >, <)

<예>

```
중간 := (아래+위) * 두개 곱하기
전대 바꿀이기 (기, 가 중간) 곱하기
고 : 아래 := 중간+1;
고 : 위 := 중간-1;
= : 시작 나 := 중간; 가라 99 곱하기
끝.
```

3-3. 제어문

제어문은 기본을 복합문으로 두어 CFG의 특징인 Self embedded한 구조와 일치시켜 구조적 프로그래밍이 쉽게 구성되게 했다. 다양한 제어흐름을 직접적으로 나타내기 위해서 여러가지 제어문을 구성했으며, 선택문인 '참이면(While)' 문을 제외한 모든 제어 문장들은 문장에 키워드

<예>

```
만약 [값] > 원나이면
시작
만[나] := 값[기];
가 := 가+1;
끝
아니면
시작
만[나] := 원[나];
나 := 나+1;
끝.
```

가 오게 했고, 그 키워드와 관련된 문미 키워드를 '끝'에 돕으로서 분기문의 사용을 가능한 줄일 수 있게 설계 했다.

3-4. 부프로그램, 함수문, 순환문

파스칼의 프로지튜어블 구조는 부프로그램과 블록 구조가 결합된 형태로 자료 선언은 지역 변수값을 연속적으로 유지시켜 줌으로서 프로그래밍을 효율적으로 작성할 수 있다. 한글 파스칼도 각 단위별로 세분화된 프로그램을 할 수 있고 문제를 더 세분화할 수 있는 블록 구조를 택했다. 또, 부프로그램과 함수문은 프로그래밍 단위로 분리될 때 매개 변수에 의한 데이터를 교환하는 방법과(Call by reference, call by value) 국부 혹은 지역 변수를 사용할 수 있게 했다. 따라서 한글 파스칼 사용자는 부프로그램과 함수문을 사용하여 서로간의 통신을 자유롭게 할 수 있어 프로그래밍의 효율을 증대시킬 수 있다. 또한 함수문이나 부프로그램이 호출되어 그 수행이 끝나기 전에 자신을 다시 호출하거나, 호출되어 있는 다른 부프로그램 혹은 함수문을 호출할 수 있는 순환(recursion) 기법을 사용할 수 있다.

<예>

```
함수 리모(미지수 : 정수) : 정수;
시작
전대 미지수기
0 : 리모 := 1;
1 : 리모 := 1;
아니면 리모(미지수-1)+리모(미지수-2)
끝(*전대 곱*)
끝(*함수 곱*)
```

3-5. 포인터문

자료를 저장하고 정보를 얻어내기 위해서 배열을 많이 사용한다. 그러나 고정된 위치와 첨자로 구성된 배열은 변화하는 문제를 다루기는 어렵다. 또한 코기를 모르거나 가변인 자료를 다룰 때는 어느 정도 큰 배열을 선언해야 할지 모른

다. 이러한 경우 적합한 자료구조는 동적 자료 구조로 이 구조는 변화가 많은 문제를 다룰 때 유용하며, 효율적으로 기억장소를 경영할 수 있다.

동적 자료구조에서는 배열처럼 고정된 기억장소 할당과는 달리 필요할 때 마다 각 노드에 대한 기억장소를 할당 할 수 있어야 한다. 새로운 노드를 위한 기억장소 할당에는 새노드(new) 라는 문이 사용된다. 또한 기억장소가 할당된 새로운 노드에 자료를 저장하기 위해서 새로운 노드를 가리킬 수 있는 방법이 필요한데 이를 위해서 포인터형이라는 특별한 형이 사용된다. 따라서 포인터는 새로운 노드를 위한 기억장소의 주소를 갖는 변수이고, 할당되는 자료구조의 형은 포인터에 의해서 결정된다.

<예>

```

프로그램 쿼 ;
형  노드포인트=노드 ;
    노드      =레코드
                자료 : 정수 ;
                다음 : 노드포인트
                끝 :
변수 머리, 꼬리, 내용 : 노드포인트
    
```

3-6. 입 · 출력문

한글 파스칼에서 입출력을 행하려면 먼저 입출력 선언부에 각 파일의 데이터에 대한 레코드 타입을 규정하도록 설계했다. 그 다음에 프로그래밍 파일에서 파일을 열고(OPEN) 각 레코드형에 대한 버퍼가 잡혀지고, 이를 통해서 입출력이 수행되게 된다. 입출력문은 어느 파일

<예>

```

시작
출쓰기('-----') ;
출쓰기('한글 파스칼 테스트 프로그램') ;
출쓰기('-----') ;
끝 :
    
```

에서 어느 레코드로 입력 혹은 출력 하는가를 명시함으로써 수행하게 된다.

<예>

```

시작
할당(주소파일, 'Add.data') ;
파일생성(주소파일) ;
출입기 (이름) ;
출입기 (주소) ;
출입기 (전화번호) ;
출쓰기 (주소파일, 이름, 주소, 전화번호) ;
끝 :
다음(주소파일) ;
    
```

한글 파스칼 명령어와 DOS 명령어 및 표준함수 예는 표 3-1, 표 3-2 및 표 3-3과 같다.

표 3-1. 한글 파스칼 명령어 일람표

한글 파스칼	파스칼	한글 파스칼	파스칼
가 라	GOTO	까 지	UNTIL
까지실행	DO	끝	END
레 코드	RECORD	만 약	IF
반 복	REPEAT	문자열	STRING
배 열	ARRAY	시 작	BEGIN
절 차	PROCEDURE	집 합	SET

표 3-2. DOS 명령어 일람표

한글 파스칼	DOS 명령어	한글 파스칼	DOS 명령어
가용 메모리	MEMAVAIL	다음 파일	FINDNEXT
도스기상호출	MSDO	디렉토리구함	GETDIR
라 인 끝	EOL	문 자 연 결	CONCAT
변 수 크 기	SIZEOF	복 사	COPY
삽 입	INSERT	선 행 자	PRED

표 3-3. 한글 파스칼 표준함수 일람표

한글 파스칼	표준함수	한글 파스칼	표준함수
절대값(인수)	ABS(X)	바꿈(인수1,인수2)	SWAP(X,Y)
난 수(인수)	RND(X)	반 올 림(인수)	ROUND(X)
문자포(인수)	CHR(X)	바 림(인수)	TRUNC(X)
지 수(인수)	EXP(X)	선 행 자(인수)	PRED(X)
후속자(인수)	SUCC(X)	순 서 값(인수)	ORD(X)

3-7. 한글 오류 처리

한글 파스칼 언어 설계시 프로그래밍에서 오류가 발생했을 때 처리할 수 있는 방법은 한글 파스칼 번역기에서 제공되는 오류메세지와 터보 파스칼에서 생성되는 오류 메세지로 구분된다. 전자는 오류 발생과 동시에 메세지를 화면이나 출력 장치로 출력하고 번역을 멈추거나, 경우에 따라서는 오류 메세지를 내보낸 채 번역을 계속하도록 설계했다. 후자는 터보 파스칼에서 제공되는 오류메세지를 한글화 시키서 출력 시키준다.

<예>

식별자가 중복되어 사용되었습니다.
 실수 실수 오류가 발생했습니다.
 너무 많은 화일을 오픈했습니다.
 '\$'은 잘못된 표기입니다.
 정의되지 않은 식별자를 사용했습니다.

3-8. 기타

일반적으로 한글 프로그래밍 언어에 있어서 명령어 길이가 길어지므로 한글 문자의 입력(key in)때 번거로움이 발생할 수 있으나, 일반화되지 않은 약어나 다른 방법에 의한 한글 사용은 한글의 자연스러움을 훼손하여 프로그래밍 언어에 심각한 문제를 야기할 수도 있으므로 그냥 자연스럽게 사용하는 것이 낫다. 또한 위 문제는 명령어의 초성이나 특수키 하나로써 선택할 수 있는 기능을 추가함으로써 문제를 해결할 수 있다.¹²⁾

4. 한글 파스칼 구현

한글 파스칼 언어는 개인용 컴퓨터의 파스칼 언어 골격을 동일하게 유지한 한글 프로그래밍 언어다. 즉 한글 파스칼 언어의 명령어는 파스칼 언어의 명령어와 일대 일 대응하도록 작성하였고, 영어를 전혀 모르는 사람도 배우기 쉽고 이해하기 쉬운 한글 명령어로 작성할 수 있다.

한글 파스칼을 구현함에 있어서 번역 기법이나, 인터프리터 기법을 고려할 수 있겠으나 기존의 파스칼 모든 장점을 그대로 이용하기 위해서 사전 번역기(preprocessor) 형태를 취했으며, 컴퓨터는 IBM PC XT / AT, MS-DOS, 컴파일러는 터보 파스칼로, 구현은 터보-C로 하였다. 본 논문에서 제시되는 사전번역기는 현재 사전처리기들에서 많이 사용되는 2단계(2 pass) 이상의 방식을 취하여 설계할 수 있겠으나 그럴 경우 처리 속도가 떨어질 우려가 있으므로 1단계(1 pass) 방식을 취했으며 구조를 도시하면 그림 4-1과 같다.

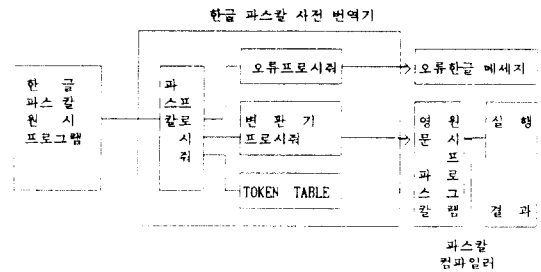


그림 4-1. 한글 파스칼 사전 번역기의 구조

4-1. 렉시칼 프로시듀어

렉시칼 프로시듀어는 번역기를 설계하는데 가장 중요하다. 왜냐하면 이 프로시듀어 토권이 잘못 나누어지면, 주요 명령 및 단어들에 대한 올바른 번역이 불가능하기 때문이다. 본 논문에서 제시되는 렉시칼 프로시듀어는 기존의 프로그래밍 언어에서 행해지는 방법과 같은 방법이나, 한글 처리도 할 수 있게 설계 되었다. 한글은 2 바이트 완성형을 채택했다. 렉시칼 프로시듀어 결과 분류되는 항목은 상수(constant), 명칭(identifier), 한글 지정어(reserved word), 연산자(operator) 등이다. 렉시칼 프로시듀어의 결과 각 단어들은 이름(name), 형(type), 줄(line)을 순서대로 생성하게 된다.

이때 <, :=, +는 파스칼에서 같은 의미로 사용되므로 바로 그 자체를 출력시키며, 만약, 값들은 순서대로 변환 프로시듀어를 사용하여

<예>

만약 갑(10이면 합 : =합+갑 :

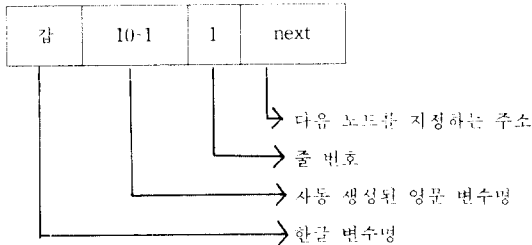
이름	만약	갑	10	이면	합	합	갑
형	id	id	정수상수	id	id	id	id
출	8						

해당 지정어나 변수명으로 바꾸어 준다.

4-2. convertor 프로시듀어

변환 프로시듀어는 렉시칼 프로시듀어 결과 한 단어가 생성되면 그 단어를 한글 예약어(표 3-1)과 표준함수테이블(표 3-2)에서 탐색하고 그 테이블에 존재하면 일대일 대응하는 파스칼 언어로 출력하고, 그렇지 않으면(사용자 언어일 경우) 그에 대응하는 변수를 자동으로 생성하게 된다. 파스칼의 구조적 특징은 블록(block) 안에 또 블록을 가질 수 있는 특징이 있다. 그러므로 한글 파스칼 언어도 이 특징을 살리기 위해 사용자 언어는 국부(global) 테이블과 지역(local) 테이블을 만들어 처리한다. 국부, 지역 테이블에는 변수, 상수, 레이블, 형, 함수, 프로시듀어들이 정의되어 있고, 자료구조는 linked-list로 <표 4-1>과 같이 구성되어 있다.

표 4-1. 국부, 지역 테이블의 자료구조



지역 테이블은 블록 레벨에 따라 그 갯수 만큼 자동으로 생성되어 사용된후 곧 소멸된다. 그리고 각 레벨에 따라 입구(entry)가 다르게 설계되었다. 다음과 같은 문장을 읽었을 때 국부와

지역 테이블은 표 4-2와 같다.

표 4-2. 국부와 지역 테이블

변수 갑, 을 : 정수 ;
함수 피보나츠(색인 : 정수) : 정수 ;

국부

갑	10-1	1		→	을	10 2	1
---	------	---	--	---	---	------	---

→	피보나츠	10-3	2	0
---	------	------	---	---

지역

색인	11 1	2	0
----	------	---	---

4-3. 오류 프로시듀어

한글 파스칼에서 발생할 수 있는 오류의 처리는 본 논문의 사전번역기에서 처리되는 오류 메세지와 터보파스칼에서 제공되는 오류메세지 2가지가 있다. 먼저 사전번역기에서 나타나는 오류 메세지는 다음과 같다.

- (1) 잘못된 토큰을 사용한 경우
- (2) 실수 표기를 잘못된 경우
- (3) 문자열의 길이가 정의된 한계를 넘는 경우
- (4) 삼입 파일을 오픈하지 않은 경우
- (5) 정의되지 않은 명칭을 사용한 경우
- (6) 원시 파일과 삼입 파일을 찾지 못하는 경우

(6)의 경우와 같은 치명적인 오류는 즉시 번역을 멈추고, 나머지 경우는 나머지 문장들을 끝까지 번역해 오류가 있는가를 조사한다. 터보 파스칼에서 생성되는 메세지는 해당오류에 대한 한글 오류 메세지로 바꾸어서 화면이나 출력 장치로 전송한다.

4-4. 한글 파스칼 알고리즘

3장에서 설계된 원리에 따른 개략적인 한글 파스칼 알고리즘은 다음과 같고, 실제 한글 파스칼 사전번역기의 각 모듈별 내용은 표 4-3과

같다.

입력 : 한글 파스칼 프로그램

출력 : 영문 파스칼 프로그램과 실행결과

1. 번역된 화일을 받아들여 확장자 H를 붙이고, 번역될 영문 화일 이름도 보좌한다.
2. 한 단어가 될때까지 토큰을 받아 들인다.
3. 단어가 명칭이 아니면 그 자제를 출력시키고 스택 2로 간다.
4. 단어를 한글 예약어 테이블과 표준 함수 테이블에서 검색하여 존재하면 해당 영문으로 출력된다.
 - 4a. 만약 단어가 부프로그램이나 함수물일 경우 그것의 순서에 해당하는 scope값을 설정한다. ($scope < scope+1$)
 - 4b. 만약 단어가 부프로그램이나 함수물이 아닐 경우 $scope < scope-1$ 로 해준다.
- 4a, 4b와 같이 해줌으로써 파스칼의 scope rule을 해결했으면, 한글 사용자 테이블인 지역 변수명은 scope 값에 의해서 결정된다.
5. 단어가 사용자 정의어일 경우 먼저 사용자 정의 테이블을 검색하고, 만약 존재하지 않으면, 사용자 정의 테이블에 추가시킨다.
 - 5a. 사용자 정의 테이블은 국부(global) 테이블과 지역(local) 테이블이 있고, 지역 변수들의 영문 이름은 scope의 영향을 받는다.
 - 5b. 사용자 정의 테이블은 linked list 자료구조로 설계되어 있으며, 각 부프로그램과 함수 프로그램에 해당하는 지역 변수 테이블의 입구(entity)를 다르고, 지역 변수 테이블은 사용 후 곧 소멸된다.
 - 5c. 지역 변수 테이블은 사용후 곧 삭제되므로 사용자 정의 테이블에서 사용자 언어를 검색하는 순서는 발생된 순서의 역으로 즉 지역 변수 마지막 테이블을 검색하고 최종적으로 국부 변수 테이블을 검색한다.

6. 스택 1~스택 6까지 모든 경우 오류가 발생하면 오류 메시지를 발생시킨다. 이때 치명적 오류(원시 화일이 없음, 삽입화일이 없음, 메모리 부족 등)는 오류 메시지를 발생과 동시에 번역을 멈추고 그렇지 않으면 스택 1~6까지 반복해 수행한다.
7. 오류가 없으면 번역된 영문 파스칼을 실행시킨다.

4-5. 한글 파스칼 프로그래밍 예제

본 논문에서 제안된 사전 번역기는 한글 파스칼의 변수, 상수, 함수, 예약어, 도스 호출, 인터럽트, 그래픽에 이르기까지 영자뿐만 아니라 순수한 한글로 프로그램이 가능하게 구현했다. 또한 오류 발생시 해당되는 오류 메시지도 한글로 표시했다. 이에 따른 예제는 다음과 같다.

```

프로그램 선택정리 :
사용유니트 정의 :
상수 자료 : 배열(1,10)의 정수=(5,6,2,4,8,9,12, 2,3,1) :
자료수=10 :
변수 값,음,장 : 정수 :
중역자료 : 텍스트 :
원자 재부르기 :
지역
  중쓰기( ' ' : 15, ' ' : ' ' : ' ' : ' ' ) :
  중쓰기( ' ' : 15, 한글 파스칼 선택 정리 ) :
  중쓰기( ' ' : 15, ' ' : ' ' : ' ' : ' ' ) :
  중쓰기 : 중쓰기 :
  음 :
  선택 값미루기(변수 원소1, 원소2 : 정수) :
  변수 입지장소 : 정수 :
  지역
    입지장소 : =원소1 :
    원소1 ' ' : =원소2 :
    원소2 ' ' : =입지장소 :
  음 :
  시작 ( * 주 프로그램 * )
  화면지움 :
  선택 (출력자료, 'bin.dat') :
  파일열림(출력자료) :
  해부쓰기 : 쓰기( ' ' : 10) :
  반복 값 : =1에서 자료수까지 실행
    쓰기(자료, 값, ' ' : 2) : 중쓰기 :
  반복 값 : =1에서 자료수 -1까지 실행
  지역
    음 : =값 :
    반복 정 : =음 +1에서 자료수까지 실행
    만약 자료, 음, 자료, 정이면 음 : =정 :
    값미루기(자료, 값, 자료, 음) :
  음 :
  
```



```
출쓰기 : 쓰기(' ' : 10);
반복 갓 : =1에서 자료수 까지 실행
          쓰기(출력자료, 자료[갓],");
반복 갓 : =1에서 자료수 까지 실행
```

한글 파스칼로 작성된 선택 정렬 프로그램이
영문 파스칼로 번역된 프로그램은 다음과 같다.

```
program H0_0;
uses crt;
const H0_1 : array[1..10] of integer=(5,6,2,4,8,9,12,2,3,1);

H0_2:=10;
var H0_3, H0_4,H0_5 : integer;
H0_6 : text;

procedure H0_7;
begin
writeln(' ' : 15, '=====' );
writeln(' ' : 15, '한글 파스칼 선택 정렬');
writeln(' ' : 15, '=====' );
writeln : writeln;
end;

procedure H0_8(var H1_0,H1_1 : integer);
var H1_2 : integer;
begin
H1_2 := H1_0;
H1_0 := H1_1;
H1_1 := H1_2;
end;

begin(* 주 프로그램 *)
clrscr;
assign(H0_6,'bin.dat');
rewrite(H0_6);
H0_7 : write(' ' : 10);
for H0_3 := 1 to H0_2 do
write(H0_1[H0_3], ' ' : 2) : writeln;

for H0_3 : 1 to H0_2-1 do
begin
H0_4 : H0_3;
for H0_5 := H0_4+1 to H0_2 do
if H0_1[H0_4]>H0_1[H0_5] then H0_4 := H0_5;
H0_8(H0_1[H0_3],H0_1[H0_4]);
end;
writeln; write(' ' : 10);
for H0_3 : 1 to H0_2 do
write(H0_6,H0_1[H0_3], ' ');
for H0_3 : 1 to H0_2 do
write(H0_1[H0_3], ' ');
writeln;
close(H0_6);
end.
```

4-6. 한글 파스칼의 분석

일반적으로 한글의 자연어 처리할 수 있는

프로그래밍 언어를 설계하는 문헌은 많이 있으나⁽³⁾ 실제로 컴파일러(compiler)를 설계하고 구현하는 것은 매우 어렵다. 지금까지 한글 프로그래밍 언어의 구현은 대체로 기존의 특정 언어를 한글로 대체 시키는데 초점이 맞추어져 있으며, 구조적 프로그래밍 언어나 모듈화된 한글 프로그래밍 언어를 행하기에 불편이 많았다. 또한 현재 많이 사용하고 있는 사전번역기는 두단계(2-pass) 이상 되어 있어서 실행시간이 많이 지연 되었다. 본 논문에서 제안되는 한글 파스칼 사전번역기는 파스칼의 구조적 특징을 그대로 유지하여 순한글 혹은 한글과 영자 혼합된 형태로 프로그래밍이 가능할 뿐만 아니라 한단계(1-pass)로 구성되어 있어 처리 속도가 빠르다.

5. 결 론

컴퓨터 교육중에서 가장 중요한 분야에 하나는 프로그래밍 분야이다. 그러나 컴퓨터 교육에 대한 연구, 계획, 방법들은 초기단계를 벗어나지 못하였고, 컴퓨터 교육을 위해서 설계되고 구현된 언어는 극소수에 불과하다. 지금까지 한글 프로그래밍 언어에 관한 많은 문헌들은 연구용으로 국한되었으며 실용화된 경우가 극히 드물고, 구조적 프로그램을 하기가 불편했다. 본 논문에서 제안한 한글 파스칼 언어는 실제로 파스칼의 구조적 특징을 그대로 유지하며, 순한글, 혹은 한글과 영자, 한자와 혼합된 형태로 프로그래밍이 가능하게 했다. 또한 도스(DOS) 명령과 오류 메시지도 한글화 했으므로 컴퓨터를 처음 대하는 사람이나, 초·중·고생들이 자연스럽고, 쉽게 한글 프로그램을 작성하고 이해할 수 있을 것이다.

향후 연구과제는 의미론을 부과하여 실제 파싱하여 실행 가능한 연구를 계속하면 한글 파스칼 언어가 구현될 것이다.

이 논문은 1990년도 문교부 지원 한국학술진흥재단의 지방대 육성 학술연구조성비에 의해서 연구되었음

참 고 문 헌

1. 김영택, 김종상, 이석호, 조유근, 권혁철, "한글 프로그래밍 언어 설계에 관한 연구", 한국 정보 과학회지 Vol 11, No.2. pp81-101. May,1984
2. 김영택, "프로그래밍 언어론", 홍릉 출판사, 1982
3. 김영택, "김파일러 구성론", 탐 출판사, 1987
4. 김민수, "국어 문법론", 일조각, 1982
5. 나일균, "구조화된 프로그래밍을 위한 CYBER 모드라 5의 사전 번역기 설계" 한국 정보과학회 논문지 Vol 11, No2, pp81-89. Feb,1985
6. 배명진, 안수길, "한글 명령 컴퓨터 설계", 마이크로 소프트웨어 1984
7. 원유현, "한글 정보 처리를 위한 프로그래밍 언어의 설계

- 와 구현", 고려대학교 박사 학위 논문
8. 원유현, 이태욱, 김명려, "컴퓨터 교육용 프로그래밍 언어 CELL의 설계에 관한 연구", 한국 정보 과학회 논문지 Vol 16, No4. pp350-361, July,1989
9. 원유현, "한글 프로그래밍 언어", 한국 정보 과학회지 2권 1호 pp19-26, 1984
10. 황종선, "한글 베이직의 표준화에 관한 연구", 공업진흥청 pp15-16, 1987
11. 황종선, 원유현, 과호형, "한글 베이직 언어의 설계와 구현", 한국 정보 과학회지 Vol 12, No1, Feb, pp52-59, 1985
12. A.N.Haberman, "Critical comments on programming language PASCAL", Acta Informatica, Vol 3, No1 . pp47-57, 1973



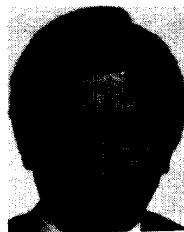
김 용 성 (Yong Sung KIM) 正會員
 1978年: 高麗大學校 數學科 卒業
 1986年 8月: 光云大學校 大學院 電子計算
 學科 博士課程修了
 1978年 1月~1980年 9月: 京畿고흥 자치
 조정실 전산팀 課長
 1986年~現在: 全北大學校 電算統計學
 科

專任講師을 거쳐 現在 助教授로 在職중

關心分野: 크래프 알고리즘, 소프트웨어공학, 교육공학

홍 성 수 (Seong Soo HONG) 正會員
 光云大學校 電算學科 卒業
 光雲 컴퓨터室 勤務
 1983年: 光云大學校 電算學科 碩士學位
 取得
 1990年: 光云大學校 電算學科 博士學位
 取得
 現在: 호서大學校 컴퓨터工學科 副教授
 在職

關心分野: 알고리즘과 시스템 프로그래밍, 병렬처리



심 재 홍 (Jae Hong SIM) 正會員
 1967年: 서울大學校 數學科 卒業
 1960年: 高麗大學校 大學院(理學碩士)
 1981年~1988年: 慶熙大學校 大學院(理
 學博士)
 1984年~1986年: 情報科學會 副會長
 歷任

現在: 光云大學校 教授로 在職중

關心分野: 크래프 알고리즘, 크래프의
 유지세션.