DBpia

論 文

교육용 한글 C 프로그래밍 언어 사전처리기의 설계및 구현

正會員 金 昌 熙* 正會員 이 상 락** 正會員 洪 性 秀*** 正會員 沈 在 洪*

Design and Implementation of Preprocessor for Educational Hangul C Programming Language

Chang Hee Kim,* Sang Rak Lee,** Sung Soo Hong,*** Jae Hong Sim* *Regular Members*

요 약

본 논문은 터보 C언어에 대응하는 한글 C언어를 설계, 구현하였다. 한글 C언어 명령어는 초·중·고등학생들이 이해하기 쉬운 용어로 선정하였으며 도스(DOS) 명령어와 오류 메세지도 한글화하였다.

ABSTRACT

In this paper, we present a design and implementation of Hangul C Programming language corresponding to Turbo C. The instructions of Hangul C were selected to be easily understood by elementary middle, high school students, and DOS commands and error message were translated into the Hangul.

I. 서 론

현재 세계는 컴퓨터의 기술발전으로 인해서 컴퓨터에 의한 교육이 중요시되고 있다. 이러한 컴퓨터 교육에서 가장 중요한 분야는 프로그래밍 언어 분야라고 하겠는데, 그 이유는 프로그래밍을 통해서 컴퓨터를 이해하고 문제의 해결력을 기르며, 논리적 사고와 창의력을 개발할 수 있기 때문이다. 그러나 프로그래밍 자체나 사용방법들이 외국어로 되어있어 외국어에 익숙하지 않은 초·중·고 학생들이나 컴퓨터를 처음 대하는 일반인들이 컴퓨터에 접근을 꺼리고 있는 실정이다. 따라서 한국인이 보다 쉽게 접할 수 있는 한글 프로그래밍 언어의 연구가 크게 필요하게 되었다.[11]

한글 프로그래밍 언어는 한국인이 보다 쉽게 대화할 수 있게 하는 도구이므로 프로그램밍의 일반적인 성질을 고려하여 설계해야한다. 그러나 프로그래밍 성질을 만족하는 한글 프로그래밍 언어의 설계를 하기 위해서는 우선 누구나 이해하기 쉬운 키워드를 선택해야 하고, 한글은 '조사'라는 품사를 갖고 있으며, 한글의 문장 구조가 post-fix라는 점을 감안하여야 한다. 현재까지 한글 프로그래밍 언어의 설계와 구현에 관한 연구가 있었으나 이런 한글의 구조적 특징을 잘 반영하지 못했을 뿐 아니라 구조적 프로그래밍을 수행하는데 쉽지 않았다.[2,12] 이미 논의된 연구에는 IMSAI용 베이직 언어를 한글로 대체하는 연구[8], 애플 기종에서 베이직 언어를 한글 명령화하여 한글

*光云大學校 電子計算學科
Dept. of Computer Science, Kwangwoon Univ.
**仁川大學校 電子計算學科
Dept. of Computer Science, Inchon Univ.
***湖西大學校 컴퓨터工學科
論文番號 : 93-26

www.dbpia.co.kr

도스및 오류 메세지를 한글화한 연구[11,14,15], 교육용 한글 프로그래밍 언어 CELL의 설계에 관한 연구[11], 자연어 처리를 위한 구문 구조에 관한 연구[1], 교육용 한글 파스칼이 설계에 관한 연구[9], 영어-한글 번역 시스템[4], 기계 번역 시스템에 필수적으로 필요한 변환 사전 기술에 관한 연구[5]등이 있다.

본 논문에서는 특이한 한글 구조에 의해 발생하는 문제점을 해결하면서 프로그래밍 언어의 일반적인 요건을 만족하는 한글 C언어 사전 처리기를 설계하고 구현함으로서 컴퓨터 마인드를 확산 할 수 있는 교육적 환경을 제공하는데 그 목적이 있다.

Ⅱ. 한글 C언어의 문법 표현

2-1. C 언어의 특성

1970년데에 개발된 C 언어는 다양한 특성을 지니는데, 첫째 수치계산, 데이터베이스 프로그램 작성, 컴퓨터 그래픽스 처리등의 용이함때문에 범용언어로서 폭넓게 사용할 수 있을 뿐만 아니라 시스템 프로그래밍 언어로서 사용이 가능하다. 둘째 다른 언어에 비해 적은 키워드를 제공하며, 다양한 제어구조와 자료형을 제공한다. 셋째 C언어는 많은 연산자를 제공하여 간결하게 표현하기 쉽고, 기계어적인 표현으로도 나타낼 수 있으므로 이해가 쉽다. 네째 다양한 라이브러리 함수를 제공하여 프로그램의 신뢰성을 높여준다. 또한 여러 기억 클래스(auto, extern, register, static)를 제공함으로서 모듈화 프로그래밍이 가능하다.[13]

2-2. 한글 C의 문법 표현

한글 C 프로그래밍의 언어는 기존의 C 컴파일러를 이용하여 C언어의 기본적인 프로그래밍의 특성을 그대로 유지하도록 하고, 한글의 구조적인 특성및 CFG(Context Free Grammer)로 표현 가능하게 하면서 읽기가 쉽고 사용하기 쉽게 설계하였다.

한글의 문법 표현에는 사용하는 명사에 다양한 조사를 붙여 표현하는 문법규정언어(Meta-Language)라고 할 수 있다. 그러나 한글 C언어에서는 많은 조사를 사용하게 되면 문장의 의미와 적용이 애매하게 되므로 조사의 사용은 제산을 두었다.[3] 또한, 한글의 문법 구성은 명령어가 뒤에 오는 후위 형태라고 할 수 있다.[2] 그러나 이러한 형태를 적용하여 파싱할 경우 파싱이 쉽지 않기 때문에 가능한한 앞부분의 키워드를 주어, 파싱과정시의 문제점을 제거하도록 하였다.[3]

한글 문장의 기본적인 구조는 <주부>+<서술부>의 형태로 구성되는데, 한글 C언어의 문장을 컴퓨터에 명령하는 명령어 중심의 설계이므로 <주부>는 생략하기로 하며, <서술부>는 <관형어>+<목적어>+<부사어>+<동사>를 기본 구조로 한다. <목적어>부분은 프로그래밍에서는 <명칭>+<조사>의 형태를 갖는 구조로서 <목적어>부분에 키워드가 나타나지 않으므로 그대로 번역할 경우 파싱테이블의 크기가 증가하며 그 의미가 애매할 수 있으므로 키워드를 앞에 위치하게끔 변형하였다.[2, 9]

그 방법중에 하나로 컴퓨터가 행하는 행위를 일반적으로 (반복, for), (선택, switch), (만약, if), (참이면반복, while), (경우, case)등으로 바꾸어 준다. 또한 기존의 파싱 방법을 그대로 사용하기 위해서 목적어의 <조사> 부분을 C 명령어에 알맞게 대체함으로서 사용자가 사용하기에 자연스럽게 하였다. 따라서 한글 C 명령어는 키워드가 맨 앞에 오게 되고, 한국 사람들에게 자연스럽게 되어 목적하는 바를 이룰 수 있게 된다.

한편, 복합문(Compound Statement)은 문장의 앞부분이 <관형어>+<주어>의 형태로 한글 프로그래밍의 언어의 특이한 구조이나 한글 C 언어 복합문의 구성은 CFG의 특성인 Self-embedded한 특성을 이용하여 문장을 구성하였으며, 복합문마다 시작과 닫음 키워드로 {, }을 사용하여 문장의 범위를 알기 쉽게 하였다.

(예)

```
선택 (배열[중간], 키){
경우 '<' : 아래 = 중간 + 1 ; 중단 ;
경우 '>' : 위 = 중간 - 1 ; 중단 ;
경우 '=' : 찾는데 걸린 반복수 = count ; 돌아감 ; 중단 ;
}
```

Ⅲ. 한글 C 언어의 설계

프로그래밍 언어를 설계하려면 먼저 언어의 사용 목적을 정의하고, 이에 필요한 자료구조, 연산, 프로그래밍 구조등을 규정하며, 그 언어의 전체적 특성을 고려하여 설계해야만 한다. 한글 C 언어의 주요 초점은 먼저 사용자가 배우기 쉽고, 프로그래밍 언어의 일반적인 요건에 알맞게 한글의 구조가 잘 반영되어,

www.dbpia.co.kr

비교적 자연언어와 유사하여 이해하기 쉽게 하는데 초점이 있다. 또한 한글 C 언어의 설계 과정에서 생길 수 있는 문제점이나 기존 C의 확장성, 신뢰성을 그대로 이용하기 위해서 기존의 컴파일러를 사용할 수 있게 하였다.

3-1. 선언부

선언부는 데이터 형태를 규정하는 부분으로 변수의 범위, 정보, 연산 수행, 데이터의 자료구조등을 기술하는 부분이다. 한글 C 에서는 변수 선언과 그에 관련된 자료형을 선언할 수 있게 하였으며, C에서와 마찬가지로 전처리기의 선언도 가능하게 하였다. 이에 따른 한글 C 변수 선언부의 BNF형식 정의는 다음과 같다.

<변수선언부> ::=<공백>|변수{<변수선언> ;}
<변 수 선 언> ::=<명칭>{,<명칭>}:[=<초기 선언>]
<초 기 선 언> ::=<초기치>|<좌대괄호><초기치>{,<초기치>}<우대괄호>
<초 기 치> ::=<식>|<부호없는 변수><식>{,<식>}|<변수>=<식>
<변 수> ::=<일반변수>|<부분변수>
<일 반 변 수> ::=<변수명칭>|<첨자변수>|<레코드 변수>
<변 수 명 칭> ::=<명칭>
<첨 자 명 칭> ::=<배열명칭>(<식>{,<식>})
<배 열 명 칭> ::=<일반변수>
<레코드변수> ::=<레코드 명칭><장명칭>
<장 명 칭> ::=<명칭>
<부 분 변 수> ::=<오른쪽 끝>|<왼쪽 끝>
<오 른 쪽 끝> ::=<식>|<공란>
<왼 쪽 끝> ::=<식>|<공란>

(예)
화일포함 <stdio.h>
정의 최대값 100
정수 변수_일 ;
실수 변수_이 ;

3-2. 연산문

선언된 자료형태 따라 수식, 문장, 배정문이 프로그램내에서 사용되는데, 수식은 변수, 상수, 함수의 호출등으로 구성된다. 또한 일반적인 프로그래밍 언어와 같이 기본적인 데이터형에 따라 행할 수 있는 연산을 규정하였고, 다른 데이터형간의 연산은 제한을 두었으며, 필요에 따라서는 시스템에서 제공하는 함수를 제공받을 수 있도록 하였다.

산술문을 구성할 때 수학식에서 일반적으로 사용하는 연산자를 사용하였으며, 문장과 문장은 세미콜론으로 구별하였다.

C언어에서 제공하는 다양한 연산자들도 정의되며, 연산자들은 우선 순위와 결합성에 의해 정의된 것에 따라 실행한다.

(연산및 우선 순위)
(1) (),[] : 괄호, 함수호출, 배열 첨자
->, . : 포인터 표시, 구조형, 공용형
(2) -,++ /--,!,~,*,& : 부호변환, 증감연산자, 부정, 1의 보수, 포인터, 포인터주소
(3) *, /,% : 산술 연산
(4) +, - : 산술 연산
(5) ≪, ≫ : 좌우 쉬프트
(6) 관계 연산자
(7) &, |, ^ : 비트 단위 연산자
(8) &&, || : 논리연산자
(9) ? : : 조건 연산자
(10) 배정, 복합배정
(11) , ; 분리자

다음은 한글 C 문장의 구문 형식을 BNF형태로 표현한것이다.

<문 장> ::=<산술문>|<제어문>|<복귀문>|<일반문>|<입출력문>
<산 술 문> ::=<명칭>=<식>
< 식 > ::=<단순식>|<식><관계연산자><단순식>
<관계연산자> ::=<|>|<=|>=|==|!=|
<단 순 식> ::=<인자>|<부호><인자>|<단순식><가감연산자><인자>
<가감연산자> ::=+|-|
<인 자> ::=<요인>|<승제 연산자><요인>|<요인><이진연산자><요인>
<승제연산자> ::=*|/|^|&&|||?|
<요 인> ::=<원소>|<원소><요인>
<원 소> ::=<변수>|<부호없는 정수>|

www.dbpia.co.kr

(<식>)|<함수 수행문>
<함수수행문> ::=<함수명칭>(<실인자열>)
<함 수 명 칭> ::=<명칭>
<실 인 자 열> ::=<공란>|<실인자>{,<실인자>}
<실 인 자> ::=<식>

<예>
{
정수인자1=1, 인자2:
인자1+=10;
인자2=인자1 % 10;
++인자;
표준출력("인자1=%2d, 인자2=%2d\n", 인자1, 인자2);
}

3-3. 입출력문

C 언어는 파스칼 언어와는 입출력 처리를 위한 루틴이 라이브러리에 등록되어 있으므로서 사용자는 이들 루틴의 사용법과 기능을 이해하여 이들 라이브러리 함수를 자신의 프로그램에서 호출하고 링크하여 입출력 처리를 손쉽게 처리할 수 있게 되어 있다. 입출력 설계는 표준 입출력과 화일 입출력을 중심으로 하되 화일입출력의 경우 프로그램에서 화일을 열고(open) 각 레코드형에 대한 버퍼를 할당하고, 이것을 통해서 입출력이 행해진다.

다음은 입출력 선언부를 BNF 구문형식으로 표현 한 것이다.

<입출력선택선언>
<입 출 력 선 언 부> ::=<공란>|<양식선언><입출력 버퍼선언>
<공 란> ::=
<양 식 선 언> ::=<공란>|양식{<양식형>;}
<입출력레코드형> ::=<명칭>{,<명칭>};<입출력형>
|여백:<여백선언>
|<명칭>{,<명칭>{;레코드{<입출력 레코드형>;}
|<명칭>{,<입출력 선택 선언>;}
[{<입출력레코드형>;}]
<입출력선택선언> ::=<상수>{,<상수>}[{<입출력 레코드형>}]
<입 출 력 형> ::=<부호없는 정수><단순형>[=<초기선언>]
|<부호없는실수><단순형>[=<초기선언>]
<단 순 형> ::=실수|정수|논리수|음절|문자|<단순형명칭>

<예>
#화일포함<stdio.h>
화일*지시자, *화일열기();
지시자=화일열기(test1.c, "r");
. . . . . .
화일닫기(지시자);

3-4. 제어구조

제어문은 기본을 복합문으로 두어 CFG의 특징인 self-embeded한 구조와 일치시켜 구조적 프로그래밍이 쉽게 구성되게 했으며, 제어 흐름을 보다 쉽게 이해하기 위해 중괄호를 이용하여 불필요한 분기문을 줄이도록 하였다.

또한 해당 키워드를 문장 앞에 두게 하여 문장의 신뢰도를 증대시켰으며, 선택문의 경우 '아니면'이라는 키워드를 두었으며 '~이면 혹은 ~면'의 한글 의미는 같으나 번역상에 애매모호함을 피하기 위해 불필요한 조사는 제외하고 '~이면'으로 한정시켰다.

한글 C에서는 루프를 중간에서 끝낼 수 있게 중단(break)을 사용할 수 있으며, 계속(continue)을 사용하면 루프 중간에 있어도 다음 반복문을 시작할 수 있다. 다음은 한글 C 제어문을 BNF 형태로 표현한 것이다.

<제 어 문> ::=<조건문>|<선택문>|<순환문>|<참이면반복문>|<반복문>|<분기문>|<탈출문>
<조 건 문> ::=만약<식>[이면<문장열>]{<식>이면<문장열>}[아니면<문장열>]
<문 장 열> ::={<문장>}
<선 택 문> ::=<변수>{<선택열>}
<선 택 열> ::=<선택값>{,<선택값>}<문장열>
<선 택 값> ::=<숫자><문자>
<반 복 문> ::=반복(<반복식>;<반복식>;[반복식])
<참이면반복> ::=반복<반복식>

www.dbpia.co.kr

<반 복 식> ::= <번수><식><식>
<분 기 문> ::= <돌아감>
<순 환 문> ::= 순환<순환문장열>
<순환문장열> ::= {<순환문장>}
<순 환 문 장> ::= <문장> | <탈출문>
<탈 출 문> ::= [<식>]돌아감

<예>
참이면 반복(낮은<=최대)
{
count++;
중간=(낮은+최대) /2;
선택(크기비교(배열[중간,]키)){
경우 '<' : 아래 = 중간 + 1 ; 중단 ;
경우 '>' : 위 = 중간 - 1 ; 중단 ;
경우 '=' : 찾는데 걸린 반복수 = count ; 돌아감 중단 ;

3-5. 함수문, 순환문, 리스트

구조적 프로그래밍에서 중요한 함수는 소프트웨어 개발 관점에서 하향식 관점이 되는데 전체적인 문제에서 구체적이고 연관적인 문제들의 프로그램 모듈화를 기할 수 있는 블럭 구조를 갖으며, 각 단위별로 세분화된 프로그래밍이 가능하게 하여 프로그램의 작성과 오류 수정이 용이하게 설계하였다.

한글 C 프로그램에서는 함수문(function)밖에 없으며, 함수문 자체가 하나의 문으로 구성될 수 있으므로 반환된 값을 무시할 수 있다. 한글 C에서는 기본 타입(type)과 포인터만이 인수로 될 수 있어서 부작용을 줄일 수 있게 하였다. 또한 인수들은 call-by-value로 전해진다. 즉 그 값은 스택에 복사되며, 파라메타 값이 바뀌어도 처음에 호출한 함수값은 변하지 않는다, 따라서 호출시에 변화를 원할때는 그 파리메타에 해당하는 타입의 포인터를 선언해 주어야 한다. 또한 함수문이 호출되어 그 수행이 끝나기 전에 함수문을 호출 할 수 있는 순환(Recursion)기법을 사용할 수 있다.

다음은 한글 C 의 주 프로그램과 부 프로그램, 함수문을 BNF구문 형태로 표현한 것이다.

<주프로그램> ::= <부프로그램> | <함수문>
<부프로그램> ::= <부프로그램머리><프로그램 구성부>
<부프로그램머리> ::= <명칭>[(매개변수열)]
<함수문> ::= <함수문머리><프로그램구성부>
<함수문머리> ::= <명칭>([<함수매개변수열>]) | <함수매개변수>
<매개변수열> ::= <매개변수>{;<매개변수>} | <함수매개변수>
<매개변수> ::= 변수<함수매개변수>
<함수매개변수> ::= <변수선언>
<함수매개변수열> ::= <함수매개변수>{,<함수매개변수>}
<복귀문> ::= <함수복귀문> | <부프로그램복귀문>
<함수복귀문> ::= <식>돌아감
<부프로그램복귀문> ::= 돌아감

<예>
정수 최대공약수(갑,을)
    정수 갑,을 ;
    {
      돌아감 을? 최대공약수(을, 갑%을) : 갑 ;
    }

변화가 많은 문제를 다룰 때는 배열을 사용하기 보다는 동적자료 구조를 사용하는 것이 효율적으로 기억장소를 경영할 수 있다. 동적 자료구조에서는 배열처럼 고정된 자료와는 달리 필요할 때마다 각 노드에 대한 기억 장소를 할 수 있다. 한글 C 에서는 새로운 노드를 위한 기억장소 할당 기법을 사용할 수 있으며, 새로운 노드에 기억장소를 할당 시키기 위해서는 새로운 노드를 가리킬 수 있는 방법이 필요한데 한글 C 에서는 이를 위해 포인터 형이라는 특별한 형이 사용된다.

<예>
참이면반복(표준입력 "%d",갑)
  {
    포인터 = 시작 ;
    시작 = (구조노드*)메로리할당(구조노드) ;
          시작 -> 번호 = 갑 ;
          시작 -> 다음 = 포인터 ;

3-6. 배열과 구조체

배열은 동일한 자료형으로 취급되는 데이터들의 리스트이고, 배열의 각 원소는 하나의 변수로 처리할 수 있고, 일반적인 변수와 마찬가지로 선언할 수 있으며, 배열임을 나타내는 특별한 키워드를 갖지는 않는다.

www.dbpia.co.kr

<예>
구조체 무명씨{
정수나이 ;
문자 이름 ;
}개인신상명세[10] ;

한글 C에서는 배열 혹은 구조체로서 인수를 변수로 보낼수 없으나 프로그램 내에서 곳에 따라 데이타형을 다르게 정의할 수 있다.

<예>
공용체 (정수 갑 ; 실수 을) 병 ;

즉 위 프로그램에서 변수 병은 정수 또는 실수 포인터 값을 가질수 있다. 그밖에도 변수들이 외부변수 혹은 정적 변수도 사용 가능하게 하였다.

3-7. 오류처리

한글 C언어의 프로그래밍시에 오류처리는 한글 C 번역기 자체에서 표시하는 오류와 기존의 터보-C 2.0에서 나타내는 오류가 있다. 전자는 오류발생과 동시에 메세지를 화면에 출력하고 번역을 멈추거나, 경우에 따라서는 오류 메세지를 보낸체 번역을 계속하도록 설계했다. 후자는 터보 C에서 제공되는 오류메세지를 한글화 시켜서 출력 시켜준다.

<예>
한글 오류메시지
"매개변수 리스트가 틀림"
"(경우)문에 ; 이 빠졌읍니다."
"{,}의 짝짓기가 잘못되었읍니다.(복합문)"
"선언 문법이 잘못되었습니다."
"0으로 나눔"
"잘못된 부동 소수점 연산."
"포인터에 포인터가 아닌 데이터 치환"
"실수 상수 오류가 발생했습니다."
"정의되지 않은 식별자를 사용했습니다."

3-8. 기타

일반적으로 한글 프로그래밍 언어는 명령어가 길어지기 쉽고, 한글 문자 입력의 번거러움이 발생되기 쉬우나 일반화되지 않은 약어나 다른 방법에 의한 한글 사용은 오히려 한글의 자연수러움을 저해할 수 있으므로 자연스럽게 번역하여 사용한다. 주요한 한글 C명령어와 DOS명령어는 표준 함수표는 표3-1과 표 3-2, 표3-3과 같다.

표 3-1 한글 C 명령어 일람표

한글 C	C	한글 C	C
중단	break	만약	if
경우	case	돌아감	return
계속	continue	구조체	struct
그외에	default	선택	switch
실행	do	형선언	typedef
아니면	else	공용체	union
집합	enum	형없는	void
반복	for	참이면반복	while

표 3-2 한글 C 명령어 일람표

한글 C	C	한글 C	C
아크코사인1()	acos()	소수나머지계산()	fmod()
아크싸인1()	asin()	자연로그()	log()
아크탄젠트()	atan()	상용로그()	log10()
코사인()	cos()	싸인()	sin()
지수계산()	exp()	제곱근()	sqrt()
가장가까운값()	floor()	탄젠트()	tan()

표 3-3 한글 DOS 명령 일람표

한 글	C
디렉토리작성	mkdir
시간날짜변환	unixtodos
시스템시간얻음	gettime
인터럽트	intr
복 사	copy
선행사	pred

Ⅳ. 한글 C 구현

한글 C는 기존의 C언어 명령어를 알기 쉽게 한글화하여 누구나 이해하기 쉽고 배우기 쉽게 프로그래밍할 수 있도록 설계하였다. 설계된 한글 명령어로 작성된 프로그램을 번역하는데에 직접 컴파일하거나 인터프리터 형태로 처리하기 보다는 기존의 터보-C의 컴파일러를 이용할 수 있도록 사전처리기(preprocessor)형태의 번역 방식을 채택하였다. 그

www.dbpia.co.kr

러므로 한글 C로 작성된 프로그램은 제시된 사전 번역기를 통해서 일반적인 IBM PC의 터보-C 2.0에서 컴파일될 수 있는 C언어의 프로그램으로 대응된다. 터보-C는 행단위로 전처리기와 컴파일이 수행되는 단일패스(single-pass)방식을 채택하고 있고, 중간화일(intermediate file)을 디스크에 따로 저장하기 보다는 메모리상에 직접 중간 데이터 구조를 구성함으로 번역 속도가 빠르다고 할 수 있다. 한글 C 사전 번역기의 구조는 그림 4-1.과 같다.

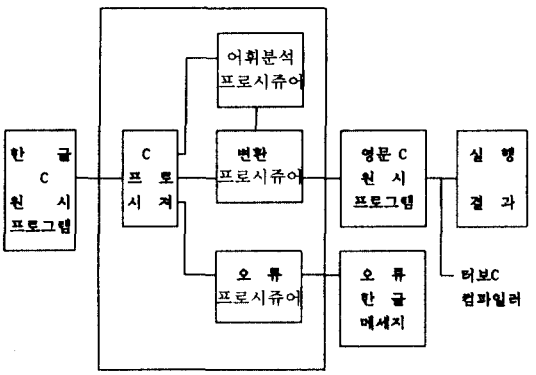


그림 4-1 한글 C 사전 번역기의 구조

4-1. 어휘분석 프로시쥬어(Lexical Analysis Procedure)

번역기의 설계시 첫번째 과정으로의 어휘분석은 원시 프로그램을 읽어 들여서 토큰의 문법적 단위로 구분하여 원시 프로그램을 분리하고 구문분석 단계의 입력이 되는 단계이다. 한글은 2바이트의 한글 완성형을 사용하였고, 어휘 분석 프로시쥬어 결과 분류되는 항목은 상수(constant), 명칭(identifier), 한글 지정어(reserved word), 연산자(operator), 기본자료형식(data type)등이다. 어휘 분석 프로시쥬어의 결과 각 단어들은 이름(name), 형(type), 줄(line)을 줄단위로 순서대로 생성하게 된다.

<예>

만약 (갑<10)이면 합=합+갑;

이름	만약	갑	10	이면	합	합	갑
형	id	id	정수상수	id	id	id	id
줄	5						

이때 <,=,+등과 같은 연산자는 C에서와 같은 의미로 사용되므로 그 자체를 출력시킨다.

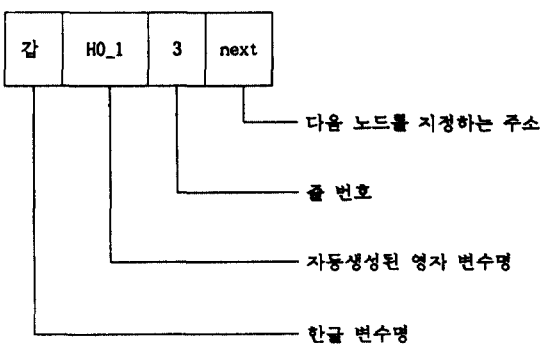
4-2. 변환 프로시쥬어(convertor procedure)

어휘분석 단계에서 생성된 단어들은 한글 예약어 테이블과 영문 예약어 테이블을 탐색하여 이들 테이블내에 존재하면 대응되는 C언어의 예약어로 출력하게되고, 존재하지 않으면 (사용자 언어일 경우)그에 대응하는 변수를 자동으로 생성한다. 사용자 정의 변수는 이전에 정의되었는지를 확인하며, 특히 C언어는 다양한 라이브러리 함수들을 제공함으로 일반 변수와 함수의 경우를 나누어서 변역될 수 있게 설계하였을 뿐만 아니라, 영문과 한글이 혼용되거나 기타 문자로 혼용되어도 처리할 수 있게 설계하였다.

C언어의 구조적 특징은 불럭안에 블럭을 가질 수 있으므로 한글 C에서도 이특징을 살리기 위해 테이블을 만들어서 처리한다. 따라서 한글 C언어도 이 특징을 살리기 위해 테이블을 만들어 처리한다. 이때 자료구조는 접속리스트(linked list)로서 표 4-1.과 같다.

테이블은 블럭 레벨({)에 따라 그 갯수 만큼 자동으로 생성되어 사용된 후 곧 소멸된다. 그리고 각 레벨에 따라 입구(entry)가 다르게 설계된다.

표 4-1 사용자 변수 테이블



4-3. 오류프로시쥬어(error procedure)

제시된 한글 번역기의 오류 처리는 번역과정에서의 문법적 오류와 기존 터보 C 2.0에서의 오류로 구분할 수 있다. 한글 번역기의 처리되는 오류에는 번역할 원시화일과 목적화일이 없거나, 번역의 사용법이 틀린 경우, 정의되지 않는 변수를 사용한다든지, 블럭의 구조({,})가 맞지 않고, 정의된 변수의 재정의등이 있으며, 주요한 오류는 오류 테이블을 구성하

www.dbpia.co.kr

여 발생한 오류의 이름과 라인수및 해당하는 오류 메시지를 출력하게 하였다.

4-4. 한글 C 알고리즘

3장에서 설계된 원리에 따라 구현된 한글 C 알고리즘은 알고리즘 4-1과 같고, 각 모듈별 내용은 표4-2와 같다.

4-1. 한글 C 알고리즘

입력 : 한글 C 프로그램

출력 : 영문 C 프로그램 과 실행 결과

단계 1) 한글 C 프로그램을 받아들여 확장자 HC를 붙이고, 번역된 영문 화일 이름 확장자 C를 보관한다.

단계 2) 입력된 문장을 어절(string)으로 받아들인다.

단계 3) 한 단어가 될 때까지 토큰을 받아들인다.

단계 4) 그 입력 문자열을 반복, 실행, 선택등과 같이 하나의 품사로 된 것인지를 검사한다.

단계 5) 만약 단계 4)에서 실패한다면 그 문자열은 동사와 다른 단어 혹은 명사와 조사가 유합된 것을 조사한다.
예) 실행하라, 갑에서, 을까지

단계 6) 단계 5)에서 명사+조사인 것을 분리해서 명사와 조사로 만들고 동사+다른 단어는 동사와 다른 단어로 분리 시킨다.

단계 7) 단어를 한글 예약어 테이블과 표준함수, 및 한글 DOS에서 검색하며 만약 존재하면 해당 영문으로 출력한다.

단계 8) 이때 단어가 블록구조({,})일 경우 그것의 순서에 해당하는 SCOPE 값을 결정한다.
(SCOPE=SCOPE+1, SCOPE=SCOPE-1)

단계 9) 만약 단어가 사용자 정의어일 경우 먼저 사용자 테이블에 그 단어를 추가시킨다.

9a) 사용자 테이블은 각 프로시져 단위와 블록구조 단위로 설계되어 있으며, scope에 따라 각각 입구가 틀리게 설계되었다.

9b) 사용자 정의 테이블의 자료구조는 접속리스트(linked-list)로 설계 되어 있으며, SCOPE 값에 따라 자동 변수들은 사용후 곧 소멸된다.

9c) 사용자 정의 테이블은 사용후 곧 삭제 되므로 사용자 테이블 사용자 변수를 검색하는 순서는 발생된 순서의 역으로 검색한다.

단계 10) 단계 1)--단계 9) 까지 모든 경우에 오류가 발생하면 오류 메시지를 발생시킨다. 이때 치명적인 오류(원시 화일이 없음, 삽입 화일이 없음) 등은 오류메시지 발생과 동시에 멈추고, 그렇지 않으면 단계 1)에서 단계 9)까지 반복해 수행한다.

단계 11) 만약 오류가 없으면 번역된 영문 C 을 실행시킨다.

표 4-2 한글 DOS 명령 일람표

main()	한글 C 사전 번역기의 메인 프로그램
scan()	subcan으로 부터 한 단어를 받아 메인으로 복귀하면, #include문, 주석문을 처리한다.
subcan()	입력자료 화일로 부터 한 문자를 받아들여 한 단어를 생성한다.
sel()	입력문자가 한글, 영문, 기타 문자인가를 판별한다.
strcat2()	문자와 스트링을 결합한다.
strcpy2()	스트링에 문자를 복사한다.
retract()	화일 포인터를 1 뒤로 한다.
searcH()	한글 예약 테이블을 참조하여 scan()에서 얻어진 단어를 검색한다.
searchA()	영문 예약 테이블을 참조하여 scan()에서 얻어진 단어를 검색한다.
searchUsr()	사용자 테이블을 참조하여 scan()에서 얻은 사용자 변수를 검색하며, 그 결과 사용자 테이블에 변수가 존재하면 그 변수를 반환하고, 그렇지 않으면 한글 변수를 영문 변수로 자동 생성한다.
store()	변수 이름이 한글이면 영문으로 변환하여 저장한다.
fstore()	함수 이름이 한글이면 영문으로 변환하여 저장한다.
PushError()	해당되는 오류 메세지를 저장한다.
PopError()	큐에서 해당 메세지를 화면에 출력시킨다.
gettoc()	화일에서 한 문자를 읽어 들인다.
Make-Filename()	해당 화일 이름을 자동으로 생성한다.
FileChange()	현재 처리 화일 포인터를 바꾸어 준다.

www.dbpia.co.kr

본 논문에서 제안된 한글 C 사전 번역기는 한글 C의 변수, 상수, 예약어, 라이브러리, 도스 명령어등 영자, 영한글 혼용 뿐만 아니라 순수한 한글 프로그래밍이 가능하게 구현하였다. 또한 프로그래밍시 발생할 수 있는 오류 메세지도 한글로 표시했다. 이에 따른 예제는 다음과 같다.

4-5. 한글 C 프로그래밍 예제

```
#화일포함<stdio.h>
#정의 자료수 10
메인()
{
   정수 자료[10]={5,6,2,4,8,9,12,-2,3,1};
   정수 갑, 을, 정;
   화면지움();
   제목쓰기();
   반복(갑=0;갑<자료수-1;++갑){
    을=갑;
     반복(정=을+1;정<자료수;++정)
      만약(자료[을]<자료[정]) 을=정;
     값바꾸기(&자료[갑],&자료[을]);
   }
   반복(갑=0;갑<자료수;++갑){
       표준출력("%d",자료[갑]);
       }
        표준출력("\n");
}
제목쓰기()
{
    표준출력("한글 C 선택정렬 예제 \n");
    표준출력("\n\n");
}
값바꾸기(정수*갑, 정수*을)
{
  정수 임수;
  임시=*갑;
  *갑=*을;
  *을=임시;}
```

이상의 한글 C언어 프로그램을 번역한 C프로그램은 다음과 같다.

```
#include <stdio.h>
#define HC_0 10
main()
{
    int HC_1[10]={5,6,2,4,8,9,12,-2,3,1};
    int HC_2,HC_3,HC_4;
    clrscr();
    FUNC_0();
    for(HC_2=0;HC_2<HC_0-1;++HC_2){
      HC_3=HC_2;
        for(HC_4=HC_3+1;HC_4<HC_0;++HC_4;
          if(HC_1[HC_3]<HC_1[HC_4])HC_3=HC_4;
        FUNC_1(&HC_1[HC_2],&HC_1[HC_3];

    for(HC_2=HC_2<HC_0;++HC_2({
      printf("%d",HC_1[HC_2]);
      }
        printf("\n");
    }
    FUNC_0()
    {
      printf("한글 C 선택정렬 예제 \n");
      printf("\n\n");
      }
    FUNC_1(int *HC_2,int *HC_3)
    {
      int HC_5;
      HC_5=*HC_2;
      *HC_2=*HC_3;
      *HC_3=HC_5;
    }
```

4-6. 한글 C 분석

한글 자연어를 처리하는 프로그래밍 언어를 번역하는 컴파일러를 설계하는 것은 매우 어려운 작업이다. 지금까지 한글 프로그래밍 언어의 구현은 대체로 기존의 특정 언어를 한글로 대체시키는데 촛점이 맞추어져 있으며, 구조적 프로그래밍 언어나 모듈화된 한글 프로그래밍 언어를 행하기에 불편이 많았다.

본 논문에서 제안된 한글 C언어는 C언어의 구조적 특징을 그대로 유지하면서 순한글 혹은 한글과 영문이 혼합된 형태로 프로그래밍이 가능할 뿐만 아니라 단일 패스 컴파일 채택하고 있어 처리속도가 빠르다. 표 4-3은 본 논문에서 제안한 한글 C 프로그래밍과 다른 비슷한 연구들의 비교표 이다.

www.dbpia.co.kr

표 4-3 한글 C 프로그래밍과 다른 한글 언어와의 비교표

제 목	내 용
한글 베이직 언어의 설계와 구현	이 논문은 베이직 언어에 대응하는 한글 베이직을 설계하고 구현한것으로 구조적 프로그램이나 모듈화가 어렵다.
구조화된 프로그램을 위한 CYBER 포트란 V의 사전 번역기 설계	이 논문은 포트란 프로그래밍의 약점이 비 구조화라는 사실에 착안하여 구성된 논문이나 완전하게 구현이 안된 논문이다.
컴퓨터의 교육용 프로그래밍언어 CELL의 설계에 관한 연구	이 논문은 교육용 프로그래밍으로 구조화와 모듈화 프로그래밍이 가능하게 설계되어있고, 한국인에게 쉽게 대화할 수 있게 설계되어 있으나 아직 구현이 안된 논문이다.
교육용 한글 C 사전처리기의 설계 및 구현	본 논문은 터보 C 에 대응하는 한글 C 언어를 설계하고 구현한 언어로 초·중·고 학생들이 이해하기 쉬운 순 한글 용어로 설계되어있고 구조적 프로그래밍도 가능한 언어이다.

Ⅵ. 결 론

일반적으로 한글 자연어를 처리하거나 한글 프로그래밍 언어를 설계하는 문헌은 많이 있으나, 실제로 컴파일러를 설계하고 구현하는 것은 매우 어렵다. 지금까지의 한글 프로그래밍 언어의 구현은 대체로 기존의 특정 언어를 한글로 대체시키는데 촛점이 맞추어져 있으며 구조적 프로그래밍이나 모듈화된 한글 프로그래밍 언어를 행하기에 불편이 많았다. 또한 현재 사용되고 있는 사전 번역기는 2단계 이상되어 실행시간이 많이 지연되었고 한글 오류 처리가 불분명하였다.

본 논문에서 제안된 한글 C 언어는 실제로 C 언어의 구조적 특성을 유지하면서 순한글 혹은 한글과 영문의 혼합된 형태도 프로그래밍 가능하게 하였다. 또한 도스명령과 오류 메세지도 한글화 했고, 컴퓨터를 이용하여 보다 쉽게 한글프로그래밍을 이해하고 작성할 수 있게 하였으며 단일 패스 방식을 채택함으로 처리 속도가 빠르다.

앞으로의 연구는 이러한 한글프로그래밍을 기본으로 하여 보다 구체적인 한글 자연어 처리나 지식 베이스를 이용한 한글 추론 시스템을 구현하는 것도 바람직하다고 하겠다.

참 고 문 헌

1. 강승식, 김영택, "한국어 형태 분석기에서 선어미말 어미의 분석 모형" 한국 정보 과학 회지, Vol 18. No.5. Sep, 1991.
2. 김영택, 김종상, 이석호, 조유근, 권혁철, "한글 프로그래밍 언어 설계에 관한 연구", 한국 정보 과학회지 Vol 11. No.2.pp81-101., May, 1984.
3. 김영택, "프로그래밍 언어론", 홍릉출판사, 1982.
4. 김영택, 이호석, "영어-한글 기계번역 시스템의 설계와 구현", 한국정보과학회, Vol 12. No.1. Feb, pp.45-51,1985.
5. 김영택, 심광섭, "변환 사전 기술 언어", 한국정보과학회, Vol 18.No.5, Sep, pp.1-11,1992.
6. 나일균, "구조화된 프로그래밍을 위한 CYBER 포트란 V의 사전 변역기 설계", 한국 정보과학회 논문지 Vol 11, No2.pp.81-89, Feb, 1985.
7. 박세영, 김권양, 오길록, "자연어 처리를 위한 한국어 Syntax 구조에 관한 연구", 한국 정보 과학회지, Vol 12. No.2.May, 1985.
8. 배명진, 안수길, "한글 명령 컴퓨터 설계", 마이크로 소프트웨어, 1984.
9. 심재홍, 홍성수, 김용성, "교육용 한글 파스칼 설계및 구현", 한국 통신 학회, Vol.16.No.10,Oct, pp.1009-1081 1991.
10. 원유헌, "한글 정보 처리를 위한 프로그램 언어의 설계와 구현", 고려대학교 박사 학위 논문, 1985.
11. 원유헌, 이태욱, 김명렬, "컴퓨터 교육용 프로그래밍 언어 CELL 의 설계에 관한 연구", 한국 정보 과학회 논문지 Vol 16.No.4. pp350-361, July 1989.
12. 원유헌, "한글 프로그래밍 언어", 한국 정보 과학회지 Vol 2. No.2, pp19-26, 1984.
13. 이재호, 이상문, 석상기, "C 박사", 정일출판사, 1991
14. 황종선, "한글 베이직의 표준화에 관한 연구", 공업진흥청 pp15-16, 1984.
15. 황종선, 원유헌, 곽호형, "한글 베이직 언어의 설계와 구현", 한국 정보 과학회지 Vol 12. No1. Feb. pp52-59, 1985.

www.dbpia.co.kr

본 논문은 1991년도 문교부 지원 학술진흥재단의 학술연구조성비에 의해서 연구되었음.

金 昌 熙(Chang Hee Kim)정회원
1988년 3월 : 광운대학교 전자계산학과 졸업
1990년 2월 : 광운대학교 대학원 전자계산학과 석사학위 취득
1991년 9월 ~ 현재 : 광운대학교 대학원 전자계산학과 박사과정 재학중
※관심분야 : 컴퓨터 그래픽스, 자연어 처리

이 상 락(Sang-Rak Lee) 정회원
1971년 : 서울대학교 사범대학 물리과 졸업
1983년 : 光云大學校 大學院 電算學科 卒業(工學碩士)
1988년 : 光云大學校 大學院 博士課程 修了
1988년 ~ 現在 : 仁川大學校 電子計算學科 助敎授로 근무중
※관심분야 : 알고리즘, 그래픽스, 자연어 처리

洪 性 秀(Sung Soo Hong) 正會員
光云大學校 電算學科 卒業
쌍용 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 副敎授 在職
※關心分野 : 알고리즘, 자연어 처리, 병렬처리

沈 在 洪(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 敎授로 在職중
※關心分野 : 그래프알고리즘, 그래픽스, 수치해설

www.dbpia.co.kr