

# 한국어 정보처리에 관한 연구

박 동 순,<sup>o</sup> 맹 승 열  
한국표준연구소

## A Study on Korean Language Information Processing

Dong-Soon Park, Seung-Ryeol Maeng  
Korea Standards Research Institute

### 요 약 문

점차 언어정보 처리가 컴퓨터 응용에서 차지하는 비중이 커지고 컴퓨터 응용 확대를 위해서는 효과적인 언어처리가 필수적 과제로 부각되고 있다. 한국어는 영어문화권 언어와는 다른 독특한 특성을 가지고 있기 때문에 이를 처리하기 독창적인 기술개발이 요구된다. 본 논문에서는 언어정보 처리의 개념과 모델을 소개하고, 보다 근본적으로 한국어 정보를 처리하기 위해 연구되어야 할 분야를 살펴보았다. 또한 한국어 정보처리를 위해서 선행되어야 하고 가장 기본적인 한국어 정보처리인 한글처리 방식에 대하여 논하였다. 본 논문에 포함된 한글, 한자코드, 한글, 한자 처리방식, O.S 한글화 지침은 언어정보처리의 시스템 요소와 입출력에 관한 연구과제이다.

### I. 서 론

컴퓨터의 주기능이 산술연산에서 정보처리로 옮겨짐에 따라 언어처리가 컴퓨터 운용에서 큰비중을 차지하게 되었으며, 효과적인 언어처리는 컴퓨터 응용 확대를 위한 필수적인 과제가 되고 있다. 따라서 컴퓨터 언어정보처리에 관련된 기술적 문제의 해결이 시급하게 되었다. 특히 한글은 영어문화권의 언어와는 다른 독특한 특성을 가지고 있어 독창적인 기술이 정립되어야 한다.

60년대초 컴퓨터가 최초로 도입된 이래 한글의 입,출력 처리를 위한 다양한 방법들이 개발되어 사용되고 있으나 이들은 외국의 S/W나 H/W에 한글을 처리할 수 있도록 해당부분을 손질하는 방법들이었다. 이러한 방법은 근본적인 한글처리 방식이 되지 못하고 한글정보처리의 장애요인이 되고있다. 뿐만아니라 이러한 방법들조차도 통일이 되지 않아 혼란을 야기하고있다.

최근 한국어 정보처리에 대한 관심이 높아지면서 자연스런 한국어 정보처리를 위해서는 종합적이고 세계적인

연구가 있어야 한다는 의견이 제시되고 있다. 본 논문에서는 이러한 의견을 토대로 한국어 정보처리를 위한 연구과제를 살펴보고, 그동안 한국표준연구소에서 수행한 한글정보처리 연구결과를 소개하고자 한다.

### II. 한국어 정보처리 범위

#### 1. 언어정보처리 모델

동양문화권 언어에 대한 정보처리 모델은 중국의 Zhao pozhang에 의해 제시된 바 있다[5]. 이 모델에 따르면 언어 정보처리 과정은 5계층으로 나뉘고 정보처리 대상은 음성, 문자정보를 포함한다. 전통적인 키보드 등에 의한 입력과 CRT 디스플레이나 라인프린터에 의한 출력 이외에, 보다 자연스러운 입출력수단으로 음성에 의한 직접 입력, 문서에 의한 직접 입력 등을 들 수 있으며, 출력수단으로는 음성에 의한 출력, 국제화에 부응하는 자국어 입력에 의한 번역된 외국어 문자출력 등을 들 수

있다.

5계층 모델에 따르면 언어정보처리 과정은 외부코드 입력계층, 외부코드를 내부코드로 변환하는 계층, 내부코드처리 계층, 내부코드를 외부코드로 변환하는 계층, 외부코드 출력 계층으로 나누어 진다.

첫번째 계층인 외부코드 입력 계층은 가능한 입력방법, 즉 음성교환코드 입력, 문자교환코드 입력, 화자음성신호 입력, 문자인식 입력, 문자입력코드 입력들을 포함하고 있다.

두번째 계층은 외부코드를 내부코드로 변환하는 계층이다. 여러방법의 입력정보는 이 계층을 통해 해당 내부코드로 매핑된다.

세번째 계층은 내부코드 처리 계층이다. 내부코드는 음성코드, 문자코드, 폰트코드등이다.

네번째 계층은 내부코드를 외부코드로 변환하는 계층이다. 위에 열거된 내부코드는 이 계층에서 원하는 출력 형태로 변환된다.

다섯번째 계층은 외부코드 출력 계층이다. 이 계층은 음성과 문자의 출력 형태를 포함한다. 음성 교환코드와 문자 교환코드는 표준코드이다.

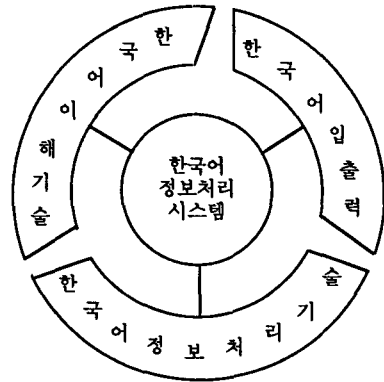
완전한 언어처리 시스템에서 입력 계층에 포함된 각각의 입력 형태는 출력 계층에 포함된 여러 출력 형태와 대응될 수 있다.

2. 한국어 정보처리를 위한 기술요소

한국어 정보처리는 정보화 사회에서의 모든 통신수단, 즉 인간과 인간, 인간과 기계, 기계와 기계간의 통신수단으로 한국어를 사용하고자 하는 의미를 갖는다. 따라서 한국어 정보처리의 범위를 인간의 언어생활에서 경험하는 모든 언어기능을 대상으로 한다.

고도화된 한국어 정보처리를 위해서는 다음과 같은 선결과제들이 있다. <그림1>은 한국어 정보처리를 위한 연구대상을 도시한 것이다. 이들의 범위를 명확히 구분할 수 있는것은 아니고, 한 분야의 연구를 위해서는 다른 분야의 연구결과를 필요로 하는 등 상호 깊은 연관을 맺고 있으나 편의상 구분해 보았다. 또한 <표1>은 연구 내용을 정리한 것이다.

첫째, 한국어 정보처리 시스템 기술은 컴퓨터 운영의 기본으로 한국어 정보처리를 위한 기본 연구과제이다. 여기에는 각종 한국어 정보의 코드계와 컴퓨터



<그림1> 한국어 정보처리를 위한 연구대상

운영체제를 포함, 여러 유틸리티의 한글지원등이다.

한국어 코드계는 정보표현의 기본단위로 언어정보처리 모델에서 살펴본 바와같이 기계내부로의 실질적 입력단위가 된다. 각종 코드의 설정과 코드체계의 표준화 국제표준화 동향에 대한 대처등 한국어 정보처리의 기본과제이다. 여기에는 문자코드를 포함 폰트 정보교환 코드, 음성코드등 언어정보의 기본 단위코드가 포함된다[6].

한글 운영체제는 한글을 자연스럽게 처리하기위한 기본과제이다. 지난 몇년간 외국에서 개발된 S/W에서 한글을 사용하기 위하여 임기응변적인 조치에 많은 노력을 낭비한 후 운영체제에서 한글이 지원되지겠다는 것을 배웠다.

다음의 연구과제는 각종 유틸리티, 프로그래밍 언어, DBMS등에서의 한글처리이다. 단순히 한글을 처리할 수 있는 것과 한국어를 기본으로 컴퓨터와 대화하는 것은 많은 차이가 있다. 한국어를 기본으로한 컴퓨터에서는 모든 명령어 및 프로그램 언어, 데이터베이스 질의어가 한글로 되어야 한다.

둘째로는 한국어 입출력이다. 자국어의 각종 정보를 처리하기 위해서는 언어의 입출력에 관련된 기술의 확보가 우선되어야 한다[1].

한글 입출력에 대한 연구는 지난 20여년간 이루어져 왔다. 이들은 주로 키보드에 의한 입력방식으로 가장 보편화된 방법이다. 그러나 최근 맨-머신 인터페이스 측면에서 새로운 입력방법들이 소개되고 있으므로 이들 문자입력 방법에 대한 연구가 요구된다.

한편, 첨단 입력방법으로 문자인식 및 음성인식의

연구도 소홀히 하지 말아야 한다. 음성인식 및 합성에 대한 연구의 필요성은 최종 출력결과가 음성 및 문자이고 보면 이에대한 연구도 중요한 과제이다.

<표1> 한국어 정보처리를 위한 연구내용

한국어 정보처리 시스템	- 한국어 정보코드계 - 한글 운영체제 (O.S) - 한글 유틸리티 - 각종 한글 System S/W
한국어 정보처리	- 한글 워드프로세서 - 한글 철자 및 띄어쓰기 검사 - 문서통신 - 한글 출판시스템 - 한글 로마자 변환
한국어 입출력	- 음성인식에 의한 입력 - 문자인식에 의한 입력 - 키보드에 의한 입력 - 한글 문자체에 대한 연구 (문자체 개발도구의 개발) - 새로운 한국어 입력방법
한국어 이해	- 한글 질의어 - 기계번역 - 한국어 이해를 위한 사전

셋째로 한국어 정보처리이다. 한국어 정보처리는 입력된 단위 정보를 사람이 원하는 형태로 가공하는 기술을 말한다. 여기에는 문서의 편집에서부터 통신까지 한국어로 표현되는 각종 정보의 처리 및 통신에 필요한 제반 기술을 포함한다.

마지막으로 한국어 정보처리를 위한 연구분야는 한국어 이해이다. 언어를 인간의 지적활동을 표현하는 수단으로 볼 때, 음성언어나 문자언어를 불문하고 언어자체가 지니는 정보를 완전하게 해석, 전달하는 기계적 수단의 확보는 대단히 중요하다. 자연어를 프로그램용 인공언어로 번역하는 것은 컴퓨터 언어를 배울 필요가 없으며, 입력된 자연어를 다른 자연어로 바꾸는 연구는 기계번역 시스템에 이용될 것이다.

지금까지 한국어 정보처리를 위한 연구과제들에 대해 살펴보았다. 이들은 상호의존 관계에 있다. 최근 정보화 사회라는 말이 자주 사용되는데 이를 위해서는 한국어 정보처리에 대한 연구가 선행되어야 할 것이다. 특히 현 기술수준에서 해결가능한 연구과제에 대한 체계적인 연구가 필요하다.

III. 한국어 정보처리 기본과제

1. 한글.한자코드 표준화

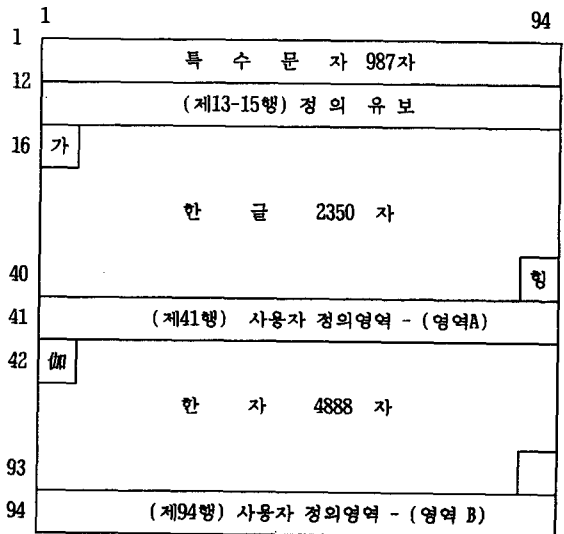
이 코드체계는 ISO-2022 복수바이트 개념을 사용한 2바이트 단위의 코드체계로서 국제 정보교환용 부호계의 제어영역인 C0, C1을 피하여 구성되어 있으며 코드의 비트구성은 <그림2>와 같다.

제 1 바 이 트								제 2 바 이 트							
b8	b7	b6	b5	b4	b3	b2	b1	b8	b7	b6	b5	b4	b3	b2	b1
M	x	x	x	x	x	x	x	M	x	x	x	x	x	x	x

default : M = 1

<그림2> 2바이트 완성형 코드 체계

<그림3>의 표준코드 체계내에는 사용빈도가 높은 한글 2,350자, 한자 4,888자와 숫자, 외국문자, 기호, 단위등과 특수문자 987자가 정의되어 있고, 이 규격에 포함되어 있지 않은 한글.한자 및 특수문자를 위한 사용자 정의영역을 포함하고 있다[2].



<그림3> KS C 5601 부호표

2. 한글.한자 입출력 방식

한글.한자코드 표준화에 이어 한글. 한자의 입출력 방법의 문제점과 개선방향을 제시하였다[4].

가. 한글 입력 방법

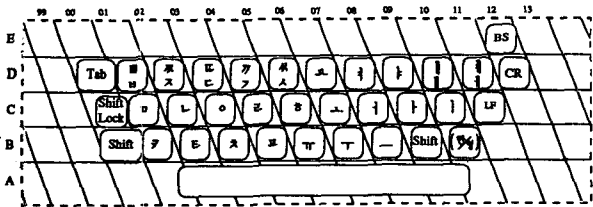
한글 풀어쓰기 키보드 입력방식에서 가장 중요한 것은

손으로 건반을 눌러 원하는 문자를 입력하게 되므로 인간공학적면이 신중히 검토되어야 한다. 건반의 갯수, 건반의 배열, 위치 그리고 터미널 제어를 위한 스위치등이 매우 중요한 요소가 된다. 또한 한글은 조합에 의한 음절구성이란 독특한 특성을 갖고 있어 음절조합에 사용되는 기본자소 (즉 건반에 배열되는 자모의 수)수와 조합방법(오토모타), 디스플레이상에 표현하는 방법등 복합적인 문제를 안고 있다.

정보처리용 건반배열(KS C 5715) 규격을 보면 기본적으로 26자만 수용하도록 되어 있고 5자의 쌍자음 (ㄲ, ㅋ, ㆁ, ㆅ, ㆆ) 및 2자의 복모음 ㅑ, ㅕ는 권장안으로 하고 있으나 이들도 입력 기본자소에 포함시켰다.

또한 한.영 선택 모드 지정키를 두어 한글 자모와 영문을 선택적으로 입력하여야 하며 (위치 B12), 한자 처리 시스템을 위하여 한자변환키를 두도록 한다.

<그림4>는 한글 입력 문자 집합과 한글입력을 위한 제어키 그리고 이들의 배열을 도시한 것이다.



<그림4> 한글 입력문자 집합과 배열

입력중인 한글표시에서 가장 문제가 되는것은 조합되지 않은 한글의 처리 (음절 디스플레이, 수정)와 입력한 한글의 수정시 표시방법이다. 입력중인 한글을 표현하기 위해서는 입력중 화면형태, 조합되는 자소 및 음절의 표현위치, 입력단위와 입력방식, 그리고 한글과 영문의 크기 비율등을 포함하여야 한다.

제어기능문자 대부분이 영문자와 같은 일차원적인 언어에 기준을 두고 기능이 정의되어 있어 한글 환경에서는 맞지 않는것이 있다. 따라서 이들에 대한 해결이 필요하다.

앞에서 설명한 Cursor의 상, 하, 좌, 우 이동이 기본적으로 음절단위 이동이 되어야 하며 추가, 삭제,

변경에 있어서도 대부분 음절 단위로 처리되어야 한다. 한글의 경우 이러한 기능을 구현하는데 불가피한 것이 Cursor addressing이다.

특수문자라 함은 주로 키보드에 정의되지 않은 문자를 말한다. 특수문자의 입력방법은 여러가지 방법이 사용되고 있다. 이들중 가장 사용빈도가 높은것은 메뉴에 의한 입력방법으로 조사되었다.

특수문자를 종류에 따라 분류하고 분류된 문자집합의 분류부호를 둔다. 사용자가 메뉴를 부르기 위하여는 분류부호를 입력한 후 한자변환 키를 누르면 해당되는 문자집합의 메뉴가 보인다. 여기서 숫자키를 이용하여 선택한다.

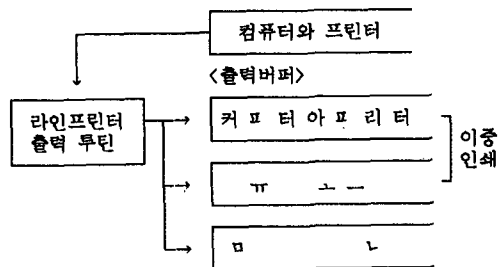
나. 한글 출력방식

본 절에서는 가장 보편적으로 사용되는 라인프린터, 점매트릭스 프린터의 자형 출력방식에 대하여 논한다.

1) 라인프린터의 한글 출력

영문 자료를 출력하는 경우 한줄 인쇄하는데 한개의 출력 버퍼만 있으면 가능하다. 그러나 한글은 앞서 언급된 바와같이 이차원적 구조를 갖기 때문에 인쇄방법이 영문과는 다르다. 일반적으로 한글은 상하, 좌우등 복합적인 위치를 갖는 자소로 이루어지고, 글자판은 한글을 한음절 단위로 만들기에는 그 수가 많으므로 자소들로 구성하고 최소 두줄에 중복 인쇄를 하여 프린트 해야한다.

<그림5>는 한글을 라인프린터로 출력하기 위한 출력용 버퍼의 구조를 보여주고 있다.



<그림5> 라인프린터의 버퍼구성

2) 점 매트릭스 프린터의 한글출력

점 매트릭스 프린터나 CRT의 경우 가장 큰 문제는

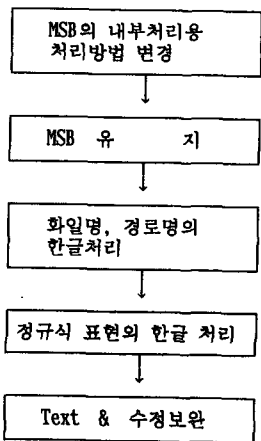
자료의 입,출력은 TTY 드라이브에 의해 이루어진다.

한글을 2바이트 완성형 코드로 표현한 경우 TTY 드라이브에 한글처리 기능을 제공하기 위해서 크게 두가지를 고려할 수 있다. 첫째는 표준처리 모드(Canonical Mode)시 미처리 대기행렬에 입력된 문자중 Erase 문자가 있는 경우 한글인 경우 2바이트가, 영문일 경우 1바이트가 지원되도록 해야한다. 둘째 Echo option이 사용되었을 때 한글 다음 Erase 문자가 입력되면 BS-BS-SP-SP-BS-BS로써 Erase 문자를 반향해야 한다. 그러기 위해서는 Canon() 루틴과 ttin() 루틴의 Echo 부분이 수정되어야 한다.

2) shell의 한글화

shell에서 한글의 처리는 명령어 인수(Argument)가 한글로 표현 가능해야 하고, 명령어에서 한글데이터를 처리할 수 있도록 하는것이다. 좀더 구체적으로 말하면, 한글 파일명 및 디렉토리명에 대한 처리 한글 문자열, 한글변수명, 경로리스트(path list) 또는 인수에 대한 한글처리가 가능해야 한다.

shell에서 한글처리가 가능토록하기 위해서 <그림8>과 같은 절차에 따라 구현하였다.



<그림8> shell의 한글구현 절차

shell에서는 내부 특수처리 용으로 MSB를 사용하고 있다. 싱글 인용부호(' ')와 이스케이프 문자 " "에 의해 인용되는 문자들이 특별한 의미를 가질경우 MSB를 '1'로 set시켜 일반문자화 하고 있다. 이 경우 한글과 중복을 일으킬수 있어 이에 대한 해결책이 요구된다.

한 방법으로 MSB 처리방법을 다음과 같이 변경할 수 있다.

- 인용부호 및 이스케이프 문자에 의해 인용되는 문자중 특수한 의미를 갖는 경우 더블 인용부호를 0x01, 명령어 대응부호 ( ' )는 0x02, 이스케이프 문자는 0x03, 문자형 부호 ( )는 0x04, 매개변수 대응부호 (\$)는 0x04로 대응시킨다. 이와같이 대응시키기 위해서는 Word() 루틴을 수정해야 한다. 수정내용은 참고문헌 [3]의 부록 A-8을 참고하기 바란다.
- 대응시킨 값을 원래의 상태로 환원해야 하므로 이를 위해 trim( ) 루틴에서 처리해 주어야 하며 수정내용은 참고문헌 [3]의 부록 A-9를 참조하기 바란다.

한편 2바이트 한글을 처리하기 위해서는 MSB를 0으로 Masking하는 모든 부분을 없애야 한다. MSB를 유지하기 위해서는 trim(), expend(), 그리고 Macro() 루틴을 수정해야 하는데 그 수정결과는 참고문헌[3]의 부록 A-10을 참고하기 바란다.

정규식 표현에서의 한글처리는 현재의 문자 또는 문자열의 확장용으로 사용되는 메타문자인 "?", "[ ]", "[-]"에 대하여 한글처리가 가능하도록 수정되어야 한다. 정규식 표현과 관련된 모듈은 expend(), 와 gmatch() 루틴이며, 한글처리를 위한 수정결과는 참고문헌 [3]의 부록 A-7과 같다.

파일명, 경로명의 한글처리를 위해서는 파일명과 경로명을 체크하는 macro인 letter와 alphanumeric을 수정해야 한다. 이 두 macro는 헤드 파일인 "ctype.h" 파일에 포함되어 있으며 다음과 같이 수정되어야 한다.

```

# define letter (c) (((c) & QUOTE) == 0&& ctype2(c)
& (T_IDC) || ((c) & QUOTE))

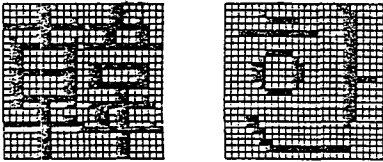
# define alphanum(c) (((c) & QUOTE) == 0 &&
ctype 2 (c) & (T_IDCH)) ||
((c) & QUOTE))
    
```

지금까지 우리는 UNIX 한글화를 위해 TTY 드라이브와 shell의 한글화 방법에 대해 살펴보았다. 그밖에도 UNIX 한글화를 위해서는 정규식에 한글이 포함될 때 처리방법,

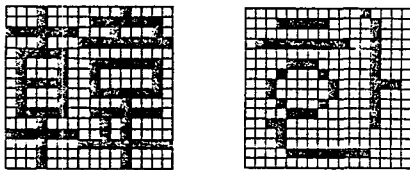
한글자형의 크기와 문자 생성기(ROM)에 자형을 어떻게 저장할 것인가 하는 문제이다. 본 연구결과에 따르면 점 매트릭스에 대한 견해는 다음이 지배적이었으며 이를 점 매트릭스 프린터의 권장안으로 한다.

- 1) 한글, 한자 자형은 프린터에서는 24x24, CRT에서는 16x16으로 하고, 자형은 완성형, 즉 음절단위로 갖는다.
- 2) 점 매트릭스 프린터와, 디스플레이 장치의 한글, 한자 코드는 KS C 5601에 따른다.
- 3) 프린터와 디스플레이 장치의 한글자형은 명조, 고딕체를 선택할 수 있도록 한다.

다음 <그림6>은 프린터와 디스플레이 장치에서 사용하고자 하는 한글자형의 크기를 예시한 것이다.



(a) 프린터의 한글, 한자 자형



(b) 디스플레이 장치의 한글, 한자 자형

<그림6> 프린터와 디스플레이 장치를 위한 한글, 한자 점 매트릭스자형

다. 한글 모아쓰기 오토마타

입력자소의 선택, 입력상의 표현방법, 출력자형 선택, 한글부호체계등 구현상의 차이로 인하여 모아쓰기 방법이 달라진다. 따라서 지금까지 논의된 내용을 근거로 한글 모아쓰기 오토마타를 구성하였다. 이 오토마타는 완성형 부호에 조합형 자형을 사용할 때와 완성형 부호에 완성형 자형을 사용할 때로 구분하여 구성되었다.

라. 한자의 변환방식

한자 변환방식이 다양하고 실제로 이들 방법을 구현할

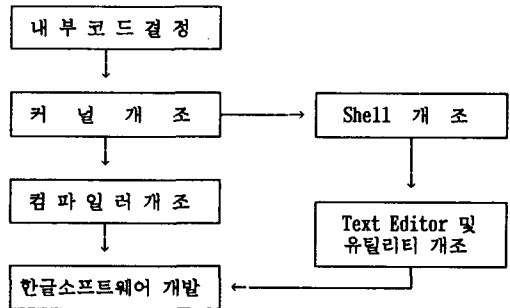
때 약간의 차이가 있어 사용자가 실제 사용할 때는 혼선을 가져오고 있다. 따라서 한자 변환방법의 통일이 필요하다. 본 연구에서는 한글의 한자변환 구동, 배열된 한자에서의 Cursor이동 방법, 한자의 선택 방법, 한자선택 취소, 한글의 한자변환등에 대하여 살펴 보았다.

3. O.S 한글화 지침서 개발

운영체제(OS) 수준에서의 한글처리는 시스템 전반적인 한글화를 의미하는것으로 운영체제의 커널(Kernel)중 TTY 드라이브(Terminal Driver)의 키인터럽트 부분에 한글처리 기능을 부여하고 화일장치로의 한글 입,출력, 한글 화일명, 한글 디렉토리명, 커맨드 인터프리터의 한글 데이터 처리, 한글 패터처리, 한글 메시지 기능을 부여하는 것이다. 본 연구에서는 표준 OS로 채택되고 있는 UNIX OS를 선정, UNIX의 한글화에 필요한 지침을 개발하였다[3].

가. 한글구현 방향

운영체제에서 한글의 구현은 같이 커널뿐만아니라 코맨드 인터프리터나 유틸리티를 포함하여야 하며 그 구현 절차는 <그림7>과 같이 설정하였다.



<그림7> 한글 구현 절차

나. UNIX 시스템 한글화 내용 및 구현방법

본 연구는 이미 표준코드로 채택된 KS C 5601 2바이트 완성형을 O.S 한글화를 위한 코드로 선택하였다. 운영체제 한글화 대상은 kernel의 TTY 드라이브, 각종 명령어의 정규식내 한글표현, Shell에서의 한글처리, 시스템 호출의 한글처리, Text 에디터의 한글처리이다.

1) TTY 드라이브의 한글화

UNIX 시스템의 각 디바이스들은 디바이스 드라이버라는 커널 모듈에 의해 제어를 받으며 실질적인

문서편집기의 한글 구현등이 필요하다. 자세한 것은 참고문헌 [3]을 참고하기 바란다.

#### IV. 결 론

본 논문은 '86년부터 한국표준연구소에서 수행한 한글정보처리 표준화연구의 결과를 요약한 것이다. 한국표준연구소는 한글정보처리 문제를 보다 체계적으로 접근해 보고자 노력하였으며, 그 결과로 한글.한자 코드에서 부터 한글.한자 입출력 방식 그리고 O.S 한글화에 대한 문제를 깊이있게 다루게 되었다.

본 연구의 내용은 새로운것을 찾기보다는 기존의 방식들을 분석. 검토하여 보다 효율적인 방법을 도출하는데 중점을 두었다. 본 연구의 결과로 각 분야의 의견을 모아 결정한 한글.한자 코드체계는 이미 KS로 제정되어 폭넓게 사용되고 있으며, 한글.한자 처리방식과 O.S 한글화 지침은 소프트웨어 및 하드웨어 설계자의 좋은 참고자료로 평가되고 있다. 그러나 한글.한자 표준 코드와 같이 좋은 결과를 얻었으면서도 확장을 위한 후속 연구가 뒷바침되지 않아 그 효과를 살리지 못하는 경우가 있다. 앞으로 이들에 대한 후속연구가 뒷바침되어 지난 연구결과와 효용성을 증대시킬 수 있길 희망한다.

끝으로, 본 연구는 언어정보처리 모델과 한국어 정보처리 대상에 비추어 볼 때 그 일부에 불과하며 아직도 한국어를 근본적으로 다루지 못했다. 따라서 한국어 정보처리에 대한 근본적인 연구의 필요성이 있다. 앞으로의 한국어 정보처리 연구의 방향은 한글을 시스템에 적용하기 위한 연구가 아니라 한국어를 자연스럽게 처리하기 위한 시스템 개발로 방향전환이 필요하다.

그동안 한글정보처리에 대한 우리의 태도를 보면 단편적이고 국부적인 시각에서 출발하였다. 시행착오를 거듭하면서 경험한 바에 따르면 한국어 정보처리의 완전한 해결을 위해서는 종합적인 연구가 필요하다는 사실이다. 특히 궁극적인 한국어 정보처리의 대상이 한글 뿐만아니라 일상생활에서 사용되는 모든 정보의 단위이고 보면 이 문제를 보다 신중히 고려할 필요가 있다. 이는 컴퓨터 전문가만의 노력으로는 불가능하고 관련된 각 분야의 도움이 필요하며, 국가적 차원에서 추진되어야 할 것이다.

#### 참 고 문 헌

- [1]. 오영환, "한글 입출력 자동화의 연구동향", KSRI 측정표준, 제11권 4호, 1988
- [2]. 박동순외, "한글.한자 코드 표준화에 관한 연구", 한국표준연구소, 1987
- [3]. 박동순외, "OS 한글화 지침서 개발에 관한 연구", 한국표준연구소, 1987
- [4]. 박동순외, "한글정보처리 표준화 연구", 한국표준연구소, 1988
- [5]. Zhao pozhang, "one of the Essential Design of East Asian Language Information Processing", (will be published)
- [6]. ISO, "NWI-Multimedia and Hypermedia Information Coded Representation", 1989. 6