

한글문 편집 System에 관한 연구 (A Study on the Editing System for Hanguel)

崔炳旭*, 李柱根*, 相磯秀夫**, 藤田廣一**

(Choi, Byung-Uk Lee, Joo-Kuen and Hideo Aiso, Hiroichi Fujita)

要 約

본 논문에서는 computer 제어에 의한 한글문 편집 system 의 개발에 관해 논한다.
한글의 24 기본자수로서 문자를 합성하고, 영수자를 혼용한 문장의 editing 이 간단히 되는 편집 system 을 minicomputer 에 실현시켜, 한글 입출력용 terminal 의 지능화의 가능성을 밝힌다.
나아가서, microprocessor 와 CRT-display 를 이용하여 값싸고 고기능을 발휘하는 system, HIT (Hanguel-Intelligent-Terminal) 의 구성예를 보인다.

Abstract

In this paper, the development of an editing system for Hanguel sentences is described.
Each Hanguel character is composed of 24 basic phonemes and 1 dummy code with satisfactorily good quality, and an editing is simply performed by using Hanguel and alphanumeric characters. The experiments demonstrate the possibility of developing an intelligent terminal for Hanguel characters. This experimental system is implemented on the minicomputer NOVA 1200.
Furthermore, HIT (Hanguel-Intelligent-Terminal) which is low cost but shows high level functions by the incorporation of a microprocessor and a CRT-display is given.

1. 序 論

근래, computer 에 의한 정보처리가 본격화 됨에 따라, 사용자의 층도 다종다양 해지고 방대한 data 를 신속 정확하게 처리하기 위하여 computer 의 고능화가 요구되고 있다. 따라서, 이의 사용효율과 처리효율을 높이기 위하여 다방면으로 연구가 진행되고 있

다. 첫째, software 면으로는 Program 이나 text 등 의 작성을 간단히 하기 위한 editor 등의 support-system 의 연구가 진행되고, 둘째, hardware 면으로는 LSI 를 사용한 단말장치에 상당한 처리능력을 부여함으로써 전체적으로 처리효율을 높이기 위한 intelligent-terminal 의 개발에 박차가 가해지고 있다.

한편, 이러한 연구의 일환으로 computer 의 한글 입출력에 관한 연구가 활발히 진행되고 있고 중요성도 강조되고 있다. 한글의 기계화는 본질적인 어려움을 내포하고 있으나, 수 종류의 system 이 상품화되어 있고 대부분의 system 이 microprocessor 에 의하여 구성되었으며, 또 특수용 초대형 display 도 가동되고 있다. 그러나, 한 두 system 을 제외하고는 대부분 입력을

* 正會員, 仁荷大學校 電子科

** 非會員, 慶應義塾大學 電氣科

(Dept. of Electronics Engineering In-ha Univ. and Keio Univ. Yokohama, Japan)

接受日字: 1978年 12月 29日

24 기본문자로 실현하지 못하고 입력부분에서 문제가 남아있다.

본 논문에서는 microprocessor를 이용한, 영수자 혼용의 한글문 편집 System의 개발을 위하여 다음과 같은 연구과제에 대하여 기술한다.

- (1) 한글의 문자합성
- (2) 인간대 기계간의 회화형식에 의한 한글 editor의 설계와 minicomputer 상의 실현
- (3) microprocessor와 CRT-display를 이용한 편집 system, HIT(Hanguel-Intelligent-Terminal)의 구성예

2. 한글의 문자합성

자음으로 시작하여 2~6자의 자소로 구성되는 한글은 Markov 과정에 속하고, 본질적으로는 자소만으로 단음절이 분리된다. 그러나 쌍자음으로 인하여 24 기본자소 만으로는 식별되지 않기 때문에, 종래에는 문자마다 단음절 space를 삽입하게 되어 입력속도의 저하를 가져왔다. 이 문제는 오랫동안 미해결의 것이었으나, 필자의 한 사람이 의하여 새로운 방법^[2]이 제안됨으로써 24 기본자소로서 단음절이 자동식별, 합성 가능하게 되었다.

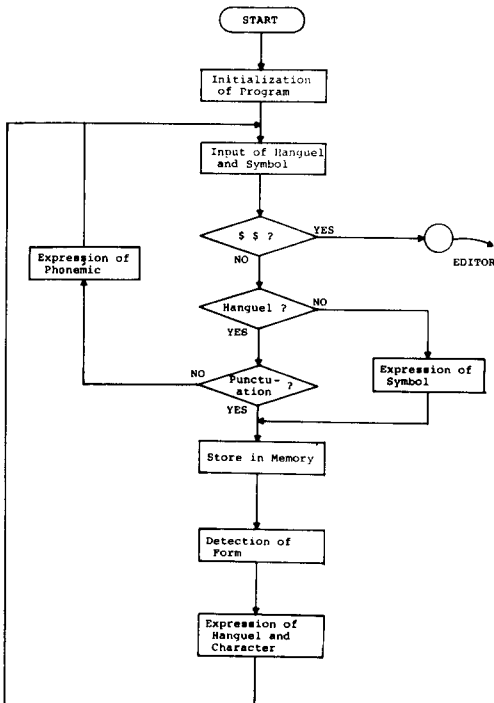


그림 1. 문자합성의 Flow-chart
Fig. 1. Flow-chart of composition of character.

이하, 한글의 문자합성을 그림 1의 flow-chart에 따라 간단히 기술한다.

(A) 한글자소는 서순(書順)대로 입력시키며, \$\$\$를 인식하면 editor-command를 실행한다.

(B) 입력된 한글자소로부터 단음절을 분리한다. 단음절의 자동식별법은 필자의 한 사람이 발표한^[2], 24 기본자소에 dummy-code를 도입한 25 기본입력에 의한, 방법을 채용했고 data(입력되는 자소 및 영수자 기호포함)의 상태천이는 그림 2와 같다.

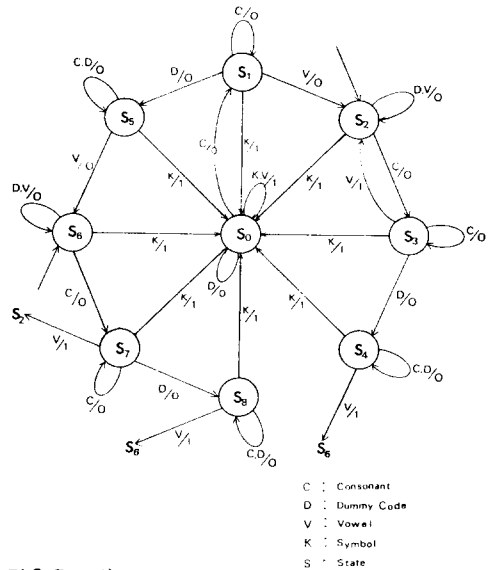


그림 2. Data의 상태천이도
Fig. 2. State diagram of data.

(C) 입력되는 자소는 단음절이 식별될 때까지 Display의 하단에 표시된다. 단음절이 식별되면, 입력된 자소는 문자로 조합되어져 상단에 표시됨과 동시에 다음 문자의 자소가 표시되기 시작한다. 그림 3에 그 예를 보인다. 이것은 사용자에게 입력에 대한

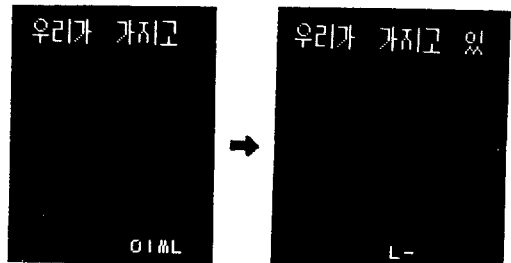


그림 3. 자소표시
Fig. 3. Expression of phoneme.

안도감을 주는 동시에 입력을 순조롭게 진행시킬 수 있으며, 인간공학적으로 중요한 문제이다.

(D) 다음절이 분리되면 다음절 표시 code를 붙여서 memory에 기억시킨다. 이것은 data의 편집시와 전송시에 필요 불가결한 조작이다.

(E) 분리된 다음절의 자소에 해당되는 문자 form^[1]을 검출한다. 기본입력을 25로 함으로써 form을 간단히 했다.

(F) 문자의 합성에 있어서는 일반적으로 가장 많이 쓰이는 dot 방식을 채용했다. dot 표시방법에는 기본 pattern을 설정하여 축소 확대하여 표시하는 방법과, 필요한 pattern을 다수 준비하여 적당한 것을 선택하여 표시하는, 두가지 방법이 있고 이들은 각각 장단점을 갖고 있다. 본 system에서는 memory의 용량은 약간 커지나 문자의 질과 처리속도를 중요시하여 후자의 방법을 채용하고, 24 기본자소를 147종으로 pattern화 했다. 또한 문자의 크기는 균형이 잡히고 한글로서 표시 가능한 최소 pattern인 9*13dot로 정했다.

이상과 같이 한글을 합성, 표시하고 출력예를 부록 1에 보인다. 이 program은 minicomputer NOVA 1200에 implement했다. 사용언어는 assembler이고 용량은 약 1.2K step이다. 한글의 자소 pattern에 소비되는 memory의 량은 약 2.5K byte이다. 그러나, 반도체기술의 발달로 값싼 memory가 입수가능하게 된 현재 주어진 dot 내에서 문자의 변형이 가능한 점등, 융통성의 면에서 보더라도 적당한 방법이라 생각된다. CRT-display는 실험용 자작의 graphic-display로 256 * 128 dot로써 표시 가능한 문자수는 23 * 7자이다.

3. 한글 Editor

일반적으로 program이나 text 등의 작성 및 수정에 있어서 그 조작을 쉽게 하기 위하여 computer에 의하여 data의 수정 및 편집 등을 행하기 위한 Program을 editor라 부른다. 이러한 editor를 한글문에 도입함으로써 한글 data의 효율적인 처리가 가능하다. 여기에 본 system에 도입한 한글 editor에 대해 소개한다.

(A) 요구되는 기능

(i) 각 명령의 실행은 인간대 기계화의 대화형식에 의해서 수행되어야 할 것.

(ii) 조작의 간단과 적은 수의 명령으로 최대의 효과를 얻기 위하여 macro 명령이 필요

(iii) window로서 graphic-display가 요구되기 때문에 display 제어용으로 강력한 명령이 필요

(iv) 그림, 표, 식 등의 배치 및 편집결과 monitoring이 한 눈에 알기 쉬워야 할 것.

(v) 한글의 다음절내, 자소의 처리기능이 필요

(B) 설계방침

상기와 같이 요구되는 기능을 고려하여 다음과 같이 설계했다.

(i) free-form text-editor^[3] 방식을 도입

(ii) text의 search는 linear-search^[4] 방식을 채용

(iii) 명령은 영문자 1~2자로서 구성

(iv) 명령의 처리에 있어서는 pointer의 이동방식과 table 참조방식을 병용

(v) 오작조에 대한 back-up으로 error-message의 출력과 실행중지의 기능첨부

(C) 구성

editor는 system 전체를 제어하고 입력되는 명령을 해석하는 main-routine과 각 명령을 실행하는 복수의 subroutine으로 구성된다.

(i) 그림 4에 NOVA 1200 상의 program의 영역과 memory의 구분은 보인다. 점선부분은 memory를

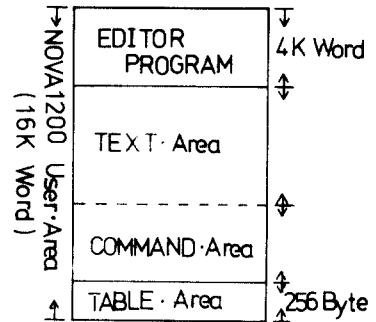


그림 4. 메모리의 구성

Fig. 4. Structure of memory.

보다 효율적으로 사용하기 위하여 일련의 명령이 실행될 때마다 memory의 영역을 갱신하도록 하여, text-area와 command-area는 명령이 실행됨에 따라 상대적으로 변한다.

(ii) Command (명령)의 구성

명령은 editor에 대하여 희망하는 동작을 지정하는 것으로 영문자 1~2자로 구성된다. 명령을 영문자로 구성함으로써 string과의 구별을 명확히 하고, 처리를 간단하게 했다. 명령의 형식은 다음과 같다.

(N) Command (string) \$

N ; argument

\$; escape-code의 echo-back

()의 내용은 경우에 따라 생략가

복수개의 Command의 지정은 다음과 같다.

command₁ \$(N₂) command₂ (string₂) \$...

...command_{n-1} (string_{n-1}) \$command_n \$\$\$

여기서 \$가 하나 입력되면 명령의 terminate 로서 해석되어 지고 계속하여 둘이 입력되면 명령의 종료로 해석 되어져 지정된 명령의 실행을 개시한다. 부록 2에 사용예를 보인다.

(D) 명령

기능적으로 보아 text-area 내의 data 를 직접 수정 및 편집하는 부류와, 2차기억 또는 File의 입출력을 행하는 부류로 나뉘어 진다. 이들 명령에 대해서 간단히 기술한다.

(i) 입출력에 관한 명령

INput ; 현재의 memory의 내용을 지우고 1 page 분의 Data를 입력시킨다.

Append ; 1 page 분의 data를 현재의 text의 끝에 입력시킨다.

Output ; memory의 내용을 출력 file에 출력시킨다.

Type ; memory의 내용을 display에 표시한다.

*Long-Range ; 한문자를 1 dot에 대응시켜 text를 format대로 display에 표시한다.

(ii) Pointer의 이동에 관한 명령

Begin ; text의 선두로 pointer를 이동시킨다.

End ; text의 끝으로 pointer를 이동시킨다.

*Set-Pointer ; 임의의 page, line, column을 지정하여 pointer를 이동시킨다.

Move ; 현재의 위치에서 지정한 만큼 pointer를 이동시킨다.

Move-Line

Move-Character

*Move-Element (한글자소의 경우)

Search ; 현재의 위치부터 지정한 string을 찾아 그 string의 뒤로 pointer를 이동시킨다.

(iii) 수정에 관한 명령

Insert ; string을 입력시킨다.

Kill ; 지정한 만큼 string를 소거시킨다.

Kill-Line

Kill-Character

*Kill-Element (한글자소의 경우)

- ; 명령의 밑줄 그은 부분은 사용자 약자를 의미한다.

* ; 한글 editor에서 특히 중요시 되는 명령

Change ; 현재의 위치부터 string₁을 찾아 string₂로 변경시킨다.

*Read ; 현재의 위치부터 string₁과 string₂를 찾아 string₁의 선두와 string₂의 끝을 기억한다.

*Write ; 현재의 위치부터 string을 찾아 read로 지정한 문장을 string앞에 삽입시킨다.

(iv) 편집에 관한 명령

BLank ; 지정한 만큼의 line을 뺀다.

*Diagram ; 그림을 삽입할 부분의 공백을 지정한다.

*Format ; text의 인쇄형식을 지정한다.

*InDent ; 각 line의 선두에 지정한 만큼 공백으로 한다.

*UnDent ; indent에서 만든 공백을 소거한다.

(v) Macro명령

동일한 명령을 반복하여 실행시킬 경우에 정의하여 사용한다.

정의 ; XMcommand₁\$command₂\$... ...command_n\$\$\$

실행 ; NX\$\$\$ (N ; 실행회수)

소거 ; XD\$\$\$

(E) 특수 Key

(i) 전술한 \$-key (escape-code)

(ii) Rubout-key ; 입력중의 한글자소만을 삭제한다.

(iii) \ -key ; 입력중의 영수자 또는 한글문자를 삭제한다.

(iv) / -key ; 한글과 영문자의 mode변환

이상과 같이 한글 editor를 설계하고, NOVA 1200 상에 implement했다. program 용량은 문자합성 Routine을 포함하여 약 3.5K step이다. 부록 2에 그 사용예를 보인다.

4. System의 구성

이상과 같은 한글문 편집 system을 micro-processor와 CRT-display 등으로 구성한 system, HIT의 구성예를 그림 5에 보인다.

(A) CPU ; interrupt의 처리가 강력하고 DMA 전송 속도가 빠르고 처리에 필요한 memory가 비교적 적게 소요되는 M-6800을 채택

(B) Memory ;

(i) 주기억은 전 program의 크기와 editing 영역을 고려하여 16K byte의 RAM과 data용(한글 pattern + 영수자와 기호 pattern)으로 4K byte의 ROM이 필요

(ii) 보조기억은 값싸고 사용에 편리한 floppy -

disk를 사용한다. 영문에 비하여 정보량이 약 3 배(1)인 한글문의 경우, 한장의 disk (0.3M byte)에 논문 2부(10-15 page) 정도의 Text가 적당가능

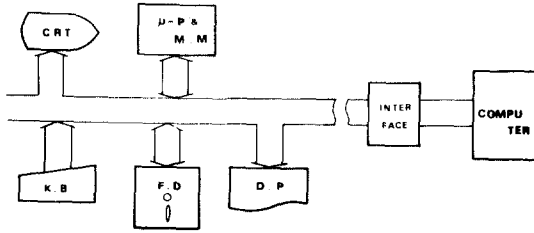


그림 5. HIT의 구성도

Fig. 5. Block diagram of HIT.

(C) Keyboard; 본 system에 사용한 keyboard는 현재 사용되고 있는 표준 TTY의 key를 25입력으로 개량한 것이다.

(D) Display; 한글의 특수성에 비추어서 graphic display가 이상적이다. dot 수는 많을수록 유리하나 가격면을 고려할 필요가 있다.

(E) Dot-printer; 깨끗한 한글을 출력하기 위하여 dot-printer가 요구되고, 이러한 printer가 요구되고, 이러한 printer는 고가인 때문에 bus 상에 설치하여 다목적적으로 사용함이 적절하다.

(F) Interface; off-line뿐만 아니라 interface를 부가함으로써 on-line으로 대형 computer의 terminal로서 직접 사용 가능하다.

(G) data의 전송에 있어서는 bus 방식을 도입함으로써 전송로의 융통성을 살린다.

이상과 같이 구성한 HIT는 다음과 같은, editing-system의 필요조건을 충분히 만족시킨다.

- ㄱ) 원고의 저자 자신이 간단히 조작가능
- ㄴ) 장치는 소형으로 값싸게 이용가능
- ㄷ) 고기능을 발휘
- ㄹ) 출력은 보다 큰 system의 입력으로 사용가능

5. 結 論

이상, 한글 입출력용 terminal에 있어서 미해결의 문제점을 해결하고 기능화의 가능성을 보여, micro-processor를 이용하여 값싸고 고기능인 한글문 편집 system의 개발에 대하여 논했다.

근래, 우리들의 일상생활에는 논문이나 문서의 작성 및 정리, list 검색 등의 일이 증가되어 가고 있다. 이와 같이 개인적인 일이나 소규모의 사무처리에 대형 computer를 이용한다는 것은 가격면에서나 사용면에서도 부적당하다고 할 수 있다. 그러나 최근 기술의 발달로 인하여 소형으로 값싸게 자유자재로 사용 가

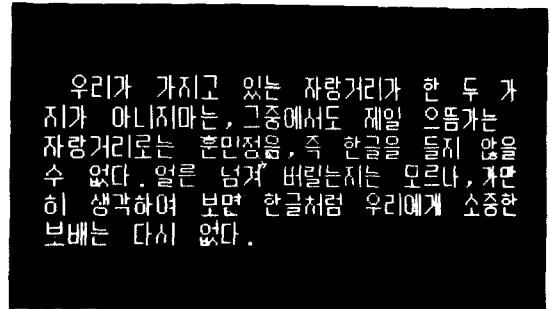
능한 editing-system이 가능하게 되었고, 이들은 사무용 기기로서 office-automation이나 library-automation 등에도 용도가 확대되어 가고 있다. 이러한 환경에서 본 연구는 한글 editing-system에 있어서 경제적인 면에서 필요조건만 고려했다. 앞으로 이 분야의 활발한 연구가 기대 되어진다. 금후의 과제로서는 실제 사용함에 있어서 요구 되어지는 기능을 보충하고, query-system 등을 도입함으로써 data-base의 access가 가능한, intelligent-terminal의 기능확장에 대한 문제라고 생각된다.

끝으로, 본 연구를 진행하는데 있어서 여러가지로 협력해 준 본 연구실 석사과정 재학중의 이길구씨와 연구실 동료에게 감사드립니다.

參 考 文 獻

1. J.K.Lee; "Korean Character Display by Variable Combination Method." Keio Engineering Reports. Vol. 26, No. 10, 1973.
2. 李柱根, 南宮在贊; "한글정보처리에서 다음절의 자동식별." 電子工學會誌 Vol. 13, No. 15, Dec. 1976.
3. Andries Van Dam, David E Rice; "On-Line Text Editing." Computing Survey. Vol. 3, No. 3, Sept. 1971.
4. D.E. Knuth; "The Art of Computer Programming. Vol. 3, Sorting and Searching." Addison-Wesley publishing company, p 389- p 550. 1975.
5. G.E. Coulouis, I Durham; "The design and implementation of an integractive Document Editor." software-practic and experience. Vol. 6, 1976.
6. Roy Blacksher; "Control an intelligent teletype-writer." Elctronic Design 2 Jan. 18. 1977.

附 錄 1



1-a. 출력예(문자)

```

ㄱ ㄴ ㄷ ㄹ ㅁ ㅂ ㅅ ㅇ ㅈ ㅊ ㅋ
ㄲ ㅅ ㅎ
ㅏ ㅑ ㅓ ㅕ ㅗ ㅛ ㅜ ㅠ ㅡ ㅣ
1234567890
ABCDEFGHIJKLMNOPQRSTUVWXYZ
XYZ
!"#$%&'(<) : - * = ; + , . <> ?
    
```

1-b. 입력예(기호)

```

*2ML$1T$$
    피고 이 땅에 래어났다.
*C이 $저 $0ML$1T$$
    피고 저 땅에 래어났다.
*C저 $이 $0ML$1T$$
    피고 이 땅에 래어났다.
*
    
```

2-d. Change 의 예

附 錄 2

```

#1 (국민 교육 현장)
1. 우리는 민족 중흥의 역사적 사명을
    피고 이 땅에 래어났다.
3. 안으로 자주 독립의 자세를 확립하고,
#
#
    
```

2-a. Insert 의 예

```

*B$$
*SID$B$4T$$
    (국민 교육 현장)
1. 우리는 민족 중흥의 역사적
    사명을
    피고 이 땅에 래어났다.
*
    
```

2-e. In Dent 의 예

```

#1T$$
3. 안으로 자주 독립의 자세를 확립하고,
*B$$
*3ML$$
*12. 조상의 빛난 얼을 오늘에 되살려,
##
*
    
```

2-b. Type, Begin, Move Line 의 예

```

LONG RANGE
*LONG RANGE의 예.
    
```

2-f. Long Range 의 예

```

1. 우리는 민족 중흥의 역사적 사명을
    피고 이 땅에 래어났다.
2. 조상의 빛난 얼을 오늘에 되살려,
3. 안으로 자주 독립의 자세를 확립하고,
*52. 조상의 $0ML$1T$$
2. 조상의 빛난 얼을 오늘에 되살려,
#
    
```

2-c. Search 의 예