

한글 코드 변환 인터페이스에 관한 연구

○
윤호상* 백두권** 황종선**

* 한국 국방 연구원 ** 고려대학교 전산학과

A Study on Hangeul Code Conversion Interface

Ho-Sang Yun* Doo-Kwon Baik** Chong-Sun Hwang**

* Korea Institute for Defence Anaysis

** Dept. of Computer Science, Korea University

< 요약 >

최근 컴퓨터의 발달로 인하여 많은 부분에서 컴퓨터가 사용되고 있다. 그러나 컴퓨터가 한글 사용을 전혀 고려하지 않고 개발되었기 때문에 한글 사용에 많은 문제점이 발생하였다. 본 논문에서는 컴퓨터에서 한글의 사용에 있어서의 문제점을 고찰해보고 이를 해결하기 위하여 한글 코드 변환 인터페이스를 연구하였다.

I. 서론

컴퓨터에서 한글을 사용하고자 하는 노력의 일환으로 2바이트를 사용한 한글 코드를 개발하여 사용하고 있다.[1,2] 그러나 컴퓨터를 개발할 때 한글 사용을 염두에 두지 않았기 때문에 한글 사용하는데 많은 문제점을 안고 있다.

가장 큰 문제점으로는 새로이 개발되는 한글 프로그램이나 패키지들은 개발과정에서 한글의 입력과 출력에 적합하도록 개발할 수 있으나 기존에 나와 있는 영문 프로그램이나 패키지들은 한글의 입출력에 많은 어려움을 가지고 있다. 이로 인하여 영문 소프트웨어를 한글화하는데 많은 노력을 들이고 있으며 새로운 소프트웨어가 나왔을 때 한글을 사용하기 위해서는 한글화하는 시간동안을 기다려야하는 불편함이 있었다.[3]

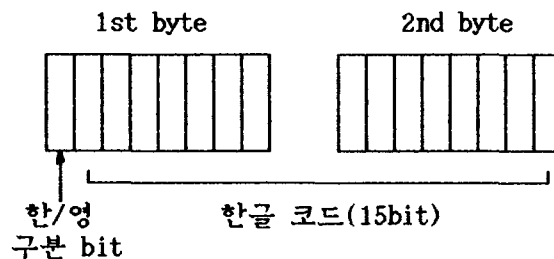
본 논문에서는 영문소프트웨어에서 한글을 사용하는데 있어서의 문제점 중에서 화면 출력에 있어서 영문 ASCII코드의 그래픽 코드부분과 한글 코드의 겹침으로 인한 그래픽 문자와 한글의 충돌을 방지하기 위한

화면 처리 코드 변환 인터페이스와 사용하는 내부코드와 상이한 한글 코드를 지원하는 프린터를 사용할 수 있도록하는 프린터 코드변환 인터페이스, 통신 포트를 사용하여 호스트 컴퓨터의 터미널로 사용하거나 기존의 네트워크에 연결하고자할 때 한글 코드의 차이를 해결해주는 통신상의 코드변환 인터페이스, 그리고 자료교환을 위한 자료의 저장이나 사용시 한글 코드의 차이를 해결해주는 디스크 I/O 코드 인터페이스에 관하여 연구하였다.

II. 기존의 한글 코드의 사용에 따르는 문제점

1. 한글코드의 구현 원리

한글코드는 그림 1과 같이 8bit코드 2개를 사용한 16bit를 사용하여 한글을 구현하였다.[3]



<그림 1> 한글 코드의 구현

한글 코드의 종류에 따라 한글 코드 부분(15bit)을 사용하는 방식이 다르다 조합형인 경우에는 15bit를 5bit씩 나누어 각각을 초성, 중성, 종성에 할당하여 한글을 나타내며, 완성형인 경우에는 15bit를 한글 각 글자에 대한 번호로 사용하여 한글을 나타낸다.[4]

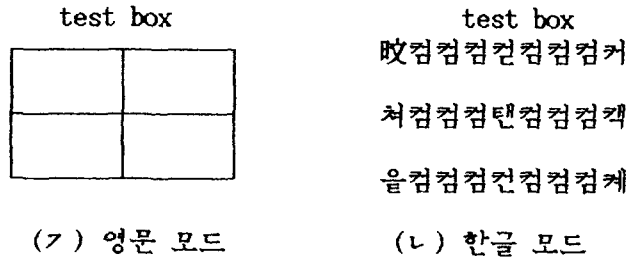
이와같은 한글 코드를 구현하는데 있어서의 사용상의 문제점과 기존의 개발된 한글코드의 통일된 사용이 이루어지지 않음으로 인한 문제점은 다음과 같다.

- 화면처리에 따르는 문제점
- 자료의 교환에 있어서의 문제점
- 주변장치 지원에 있어서의 문제점
- 정보통신에 있어서의 문제점

2. 화면 처리에 따르는 문제점

화면처리에 따르는 문제점은 한글 코드의 구현상의 문제점으로서 한글 코드가 한글/영문을 구분하기 위하여 MSB를 셋팅하기 때문에 영문 ASCII 코드의 그래픽 BOX문자와 겹친다. 이와같은 문제점은 한글 소프트웨어에서는 문제가 되지 않지만 영문 소프트웨어에서는 화면 처리를 위하여 여

러가지 다양한 BOX문자를 사용하기 때문에 한글을 사용하기 위하여 한글 카드를 사용한다든지 한글 구현 소프트웨어를 사용하게 되면 영문 소프트웨어에서는 화면이 알아보지 못할 정도로 깨지는 경우가 발생한다.(그림 2참조)



<그림 2> 영문 그래픽 문자와 한글의 충돌

이와같은 문제점은 사용자로 하여금 많은 불편을 초래하며 뛰어난 영문 소프트웨어에서 한글 사용을 어렵게 한다. 이는 기존 한글 코드의 구현 방식에 따른 문제이므로 이를 해결하기 위해서는 영문의 그래픽 box와 충돌을 일으키지 않는 한글 코드를 만들어 사용하는 수 밖에 없다. 그러나 현재의 상황에서 많은 한글 코드의 난립으로 한글 코드가 통일되지 못하는데 새로운 한글 코드를 개발한다는 것은 더욱 코드의 통일을 어렵게 한다. 본 논문에서는 이와같은 그래픽 box와 충돌하지 않는 한글 코드를 화면 처리에만 사용하고 내부 코드는 기존의 한글 코드를 사용하는 방법으로 이 문제를 해결하였다. 이를 해결하는 방법으로 화면 처리 코드 변환 인터페이스를 설계하였다.

3. 자료의 교환에 있어서의 문제점

자료교환에 있어서의 문제점은 한글 코드의 통일이 이루어 지지 않음으로 인한 문제인데 이는 서로 다른 한글 코드를 사용하는 컴퓨터 사이의 자료를 서로 교환하여 사용할 때 한글 코드의 차이로 인한 한글 코드 변환 작업이 필요하게 된다. 이와같은 문제점은 적은량의 자료를 교환하여 사용할 때는 별 문제가 되지 않을 지 모르나 많은 자료를 교환하고자 할 때는 자료들의 한글 코드 변환 작업에 많은 시간을 낭비하게 된다.[5]

본 논문에서는 이와같은 자료의 한글 코드 변환 작업을 손쉽게 하기 위하여 자료교환을 위하여 또는 자료 저장을 위하여 자료를 기억 매체에 저장할 때 자동적으로 코드변환 작업을 하고 저장된 자료나 교환한 자료를 사용하기 위하여 컴퓨터 내부로 읽어들일 때 자동적으로 코드 변환을 하는 디스크 I/O 코드변환 인터 페이스를 연구하였다. 그리고 저장된 자료를 관리하기 위한 방안으로 저장자료에 한글 코드에 대한 정보와 기타 유용한 정보를 저장할 수 있도록하는 방안을 연구하였다.

4. 주변기기(프린터)사용에 있어서의 문제점

주변기기(프린터)사용에 있어서의 문제점은 최근에 나오는 프린터들은 국내에서 사용되고 있는 한글 코드를 거의 모두 지원하고 있다. 그러나 몇몇 프린터들은 지원하기 못하는 한글 코드가 존재하기 때문에 사용자에게 많은 제약을 가하고 있다. 이와같은 문제점으로 인하여 사용자들은 자신의 워드 프로세서나 에디터로 자료를 작성한 후 이를 프린터가 지원하는 한글 코드로 변환하는 작업을 거치게 된다.

본 논문에서는 이와같은 문제점을 해결하기 위하여 프린터 출력 코드 변환 인터페이스에 대하여 연구 하였다.

5. 정보통신에 있어서의 문제점

현재 사용중인 한글 코드 중에서 KS로 표준화된 2바이트 완성형 코드는 국제 규격(ISO)의 부호확장법을 따르고 있으며 정보 통신에 적합한 코드라는 것이 장점이다. 그러나 이 2바이트 완성형 한글 코드의 단점으로는 한글의 모든 글자를 나타낼 수 없으며 한글의 각 음소 분리가 어렵고 SORTING에도 문제가 발생할 수 있다는 것이다. 이로 인하여 아직까지 많은 사람들이 2바이트 조합형을 주장하고 사용하고 있다. 그러나 2바이트 조합형의 1절에서 설명한바와 같이 2바이트를 1비트를 영/한 구분, 나머지 15비트를 초성, 중성, 종성으로 분리하여 사용하기 때문에 2번째 바이트가 정보 통신의 제어코드 영역(C0, C1)과 겹치는 경우가 발생하므로 정보 통신에 있어서 문제가 발생할 소지가 있기 때문에 통신에는 부적합한 한글 코드라고 볼 수 있다.[7]

본 논문에서는 이와같은 각 코드의 단점을 해결하는 방안으로 2바이트 완성형 코드를 정보 통신에 사용하고, 2바이트 조합형 코드나 완성형 코드를 컴퓨터 내부 코드로 사용할 수 있도록하는 방안을 제시하고 이를 위하여 통신 포트를 통한 정보 통신에 있어서 한글 코드를 자동적으로 변환해주는 통신 코드 변환 인터페이스에 관하여 연구 하였다.

III. 한글 코드 변환 인터페이스

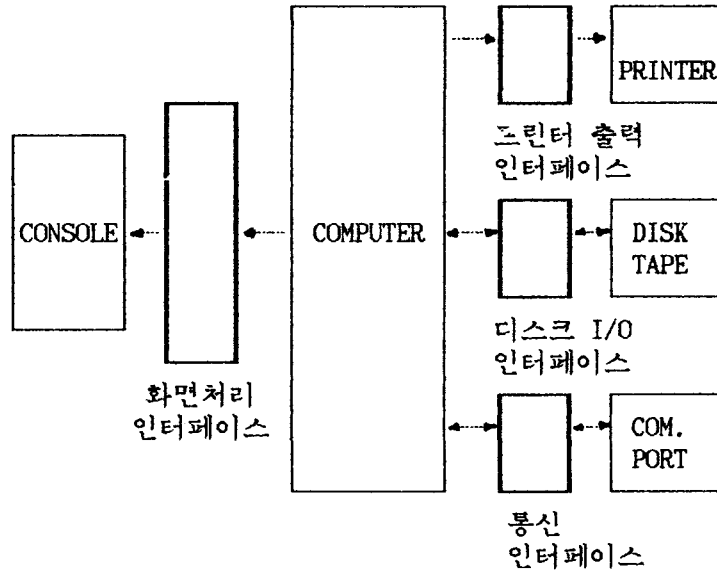
1. 한글 코드 변환 인터페이스의 구성

II절에서 제시한 컴퓨터에서 한글을 사용하는데 있어서의 문제점 해결 방안을 통합하면 크게 한글 코드 변환 인터페이스로 구성될 수 있다.

이와같은 한글 코드 변환 인터페이스들은 사용자에게 한글 사용에 따르는 여러가지 문제점과 한글 자료를 사용할 때 부과되는 코드 변환 작업 등 여러가지 불편한 점을 해결하기 위하여 응용 프로그램 수준이 아닌 OS수준의 프로그램을 사용하는데 이 프로그램은 OS에서 제공하는 function call을 수정하여 코드 변환 기능을 구현할 수 있다.

한글 코드변환 인터페이스는 앞에서 제시한 것과 같이 크게 4가지로 구성된다.

- 화면처리 코드변환 인터페이스
- 디스크 I/O코드변환 인터페이스
- 프린터 출력 코드변환 인터페이스
- 통신 코드변환 인터페이스



<그림 3> 한글 코드 변환 인터페이스의 구성

2. 화면 처리 코드변환 인터페이스

이 인터페이스는 앞에서 언급한 한글을 사용함으로써 야기되는 영문 소프트웨어의 그래픽 BOX문자와 한글의 충돌을 방지하기 위해서 컴퓨터 내부에서 CONSOLE로 출력되는 문자들을 검사하여 코드변환 하여 BOX문자와 충돌되지 않도록 하는 하는 코드 변환 인터페이스이다.

화면처리 인터페이스를 구현하는 방법은 크게 2가지로 나눌 수 있다.

(1) 기존의 한글 코드를 그대로 사용하면서 BOX문자를 한글 모드에서의 BOX문자로 변환하는 방법

이 방법은 2바이트 완성형 한글 코드를 사용하는 경우에 가능하다. 2바이트 완성형 한글 코드는 한글 모드에서 단선 BOX문자를 제공한다. 그러므로 화면에 출력되는 문자중에서 영문 그래픽 BOX문자를 한글 코드의 BOX문자로 변환하는 방법이다.

이 방법의 단점은 완성형 한글 코드가 단선 BOX문자([] ' ' | -)만을 제공하기 때문에 영문 모드의 겹선 BOX문자들도 모두 단선 BOX문자로 나타내야하는 단점이 있다. 그러나 이방법은 코드 변환 문자의 수가 작기 때문에 처리 속도가 빠른 장점을 가지고 있다.

(2) 그래픽 BOX문자와 한글과 충돌을 일으키지 않는 한글 코드를 사용하여 화면처리를 하며 출력되는 한글 코드를 화면처리 한글 코드로 변환하는 방법

이 방법은 모든 한글 코드에 대하여 적용가능하며 화면 처리용 한글 코드는 일명 "도깨비 II" 한글 코드를 사용하는데 이 한글 코드는 조합형 한글 코드로서 한글 코드 영역과 영문의 그래픽 BOX문자의 충돌을 제거한 한글 코드로서 영문 소프트웨어에서 완벽한 그래픽 BOX문자를 사용할 수 있는 한글 코드이다.[8]

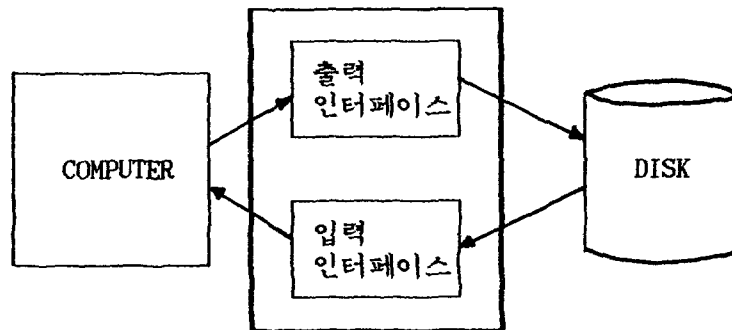
구현 방법은 화면 출력되는 모든 문자를 "도깨비 II" 한글코드로 변환하여 출력하는 방법이다. 이 방법의 장점은 모든 한글 코드에 대하여 적용할 수 있다는 것이다. 그러나 출력되는 모든 코드를 변환해야하므로 화면처리 속도면에서는 다소 떨어지는 단점이 있다.

3. 디스크 I/O 코드변환 인터페이스

디스크 I/O인터페이스는 자료교환을 위하여 자료를 기억매체에 저장하거나 저장된 자료를 사용할 때 사용자로 하여금 코드 변환 작업을 하는 번거로움과 불편을 해결하는 방법으로 디스크의 화일을 읽거나 화일을 저장할 때 정보교환을 위한 한글 코드 변환 작업을 한다.[6]

이와같은 디스크 I/O인터페이스가 자동적으로 한글 코드 변환해 주기 위해서는 자료를 저장할 때 사용 한글 코드에 대한 정보를 같이 저장함으로써 코드 변환 인터페이스가 저장된 자료에 대한 한글 코드를 인식하고 이를 이용하여 저장된 자료의 코드변환을 수행한다.

본 논문에서 구성한 디스크 I/O인터페이스는 그림 4과 같다.



<그림 4> 디스크 I/O 코드변환 인터페이스

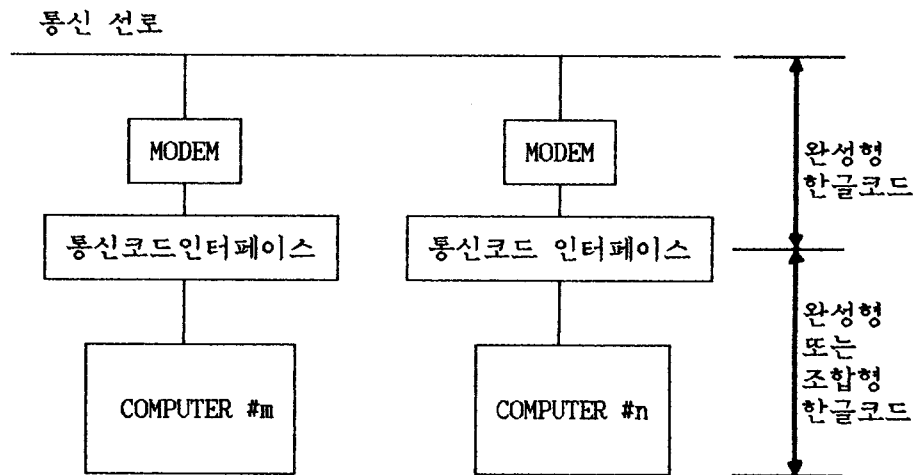
4. 프린터 출력 코드 변환 인터페이스

프린터 출력 코드변환 인터페이스는 컴퓨터 내부에서 사용하는 한글 코드와 프린터에서 제공하는 한글 코드가 다를 때 이를 해결하기 위하여 출력되는 자료를 중간에서 자동적으로 코드 변환시키는 기능을 가진다.

5. 통신상의 코드 변환 인터페이스

통신상의 코드변환 인터페이스 역시 통신 포트를 통하여 다른 컴퓨터와 통신할 때 컴퓨터 내부에서 사용하는 코드와 독립적으로 통신에 적합한 한글 코드를 통하여 통신을 함으로써 통신에 부적합한 한글 코드를 사용하여 통신함으로써 야기되는 문제점을 해결할 수 있다.[7]

본 논문에서는 통신상의 한글 코드는 2바이트 완성형 한글 코드를 사용하며 내부 코드로서는 완성형 또는 조합형을 사용할 수 있도록 하는 통신상의 코드 변환 인터페이스를 구성하였다.(그림 5참조)



<그림 5> 통신 코드 변환 인터페이스

IV. 한글 코드 변환 인터페이스의 사용

III절에서 제시한 한글 코드 변환 인터페이스는 기존의 한글 코드를 사용할 때 발생하는 여러 문제점을 해결하기 위하여 연구 되었다. 컴퓨터를 사용하는 사용자들이 가지고 있는 컴퓨터 시스템이 각기 다르기 때문에 한글 사용의 문제점 역시 각기 다르다. 그러므로 위에서 제시한 한글 코드 변환 인터페이스 중에서 사용자의 요구에 따라 원하는 인터페이스만 사용함으로써 한글 사용의 문제점을 해결할 수 있다.

V. 결론

본 논문에서는 기존의 한글 코드의 사용에 따르는 문제점인 화면처리에 있어서의 영문의 그래픽 BOX문자의 한글과의 충돌, 자료 교환에 따르는 한글 코드의 불일치로 인한 코드 변환 작업의 불편함, 사용한글 코드를 지원하지 못하는 주변기기(프린터)의 사용에 따르는 문제점, 기존의 한글 코드의 사용에 따르는 통신상의 문제점을 해결하기 위하여 통합된 한글 코드 변환 인터페이스를 제시하였다.

그리고 이러한 코드 변환 인터페이스를 응용 프로그램 수준이 아닌 OS 수준의 인터페이스로 구현함으로써 사용자에게 최대한의 편의성을 제공한다.

VI. 참고 문헌

1. 과학 기술처, 한글 정보 처리 표준화 연구, 1986.7
2. 과학 기술처, 한글 정보 처리 표준화 연구(3차년도), 1986.7
3. 오길복, 박세영 "한글 정보 처리를 위한 기초 연구" 정보과학회지 2권 4호 pp 7-19, 1984.12
4. 황대길 "한글 자료처리용 부호계" 정보과학회지 2권 4호 pp 38-41, 1984.12
5. 윤호상, 손진곤, 황종선, "한글문서화일의 효과적인 관리에 관한 연구", 한국정보과학회 논문지 Vol 16, No.2, pp 705-708, 1989.10
6. 윤호상, 백두권, 황종선, "한글 문서화일 관리 시스템(HT-FMS)의 설계 및 구현", 한국정보과학회 논문지 Vol 17, No.1, pp 329-332, 1990.4
7. 한국 전자 통신 연구소, 한글 코드 표준화, 1987.12
8. 마이크로 소프트웨어, "현행 한글 코드의 문제점과 해결 방안", (주) 정보시대, 1989.8
9. Peterson, Silberscharts, "Operating system concept", Addison-Wesley Publishing Company, 1985.
10. 김지운, 장성익 "IBM-PC 정복", 달리만듬 기획, 1990.1
11. 한성국, "시스템 프로그래머를 위한 IBM-PC XT/AT, PS/2 기술 사전", 집문당, 1988.12