

國語學 研究에 있어서의 퍼스컴의 活用 可能性

韓 榮 均

I. 머릿말

80년대에 들어서 이른바 애플이라고 알려진 8bit 컴퓨터가 보급되면서 적지않은 국어국문학 연구자들이 관심을 갖고 그 활용 가능성을 모색해 왔다. 그러나 당시 8bit 컴퓨터는 일반적으로 한글만 사용 가능하였고, 한자는 처리가 불가능했기 때문에 관심을 두는 정도에 그칠 뿐 직접 그것을 국어국문학 연구에 활용하게 되는 경우는 많지 않았다. 그러던 것이 16bit 개인용 컴퓨터가 보급되면서 한자의 입·출력이 가능해져서 점차 국어국문학 연구에 있어서의 PC(personal computer)의 활용 가능성이 높아지고 그에 따라 PC사용자가 점차 늘어나는 추세에 있다.

그러나 아직까지 대부분의 국어국문학 종사자들은 컴퓨터라고 하는 “기계”에 대한 거부감이나 두려움, 혹은 얼마만큼의 활용가치가 있느냐 하는 점에 대한 의구심 때문에 컴퓨터를 구입하여 사용하는 것을 꺼리고 있는 것이 사실이다. 경우에 따라서는 이미 PC를 구입하여 사용하고 있는 경우에도 그 기능을 충분히 활용하지 못하고 단순히 타자기와 같은 문서 작성기로만 사용하는 경우도 적지 않아, 컴퓨터의 효용성에 대한 의구심이 이는 것이 한편으로는 당연한 일이다. 그러나 본인의 경우 이미 개발되어 있는 software만 활용해서 수작업에 의한 자료의 정리·분류 등에 소요되는 시간을 대폭 줄일 수 있었고 앞으로 조금만 노력하면 더욱 체계적이고 효율적인 연구가 가능하다는 사실을 확인한 바, 이 글은 이러한 경험을 바탕으로 컴퓨터를 활용한 연구가 가능하리라고 생각하면서도 상기와 같

은 이유로 해서 망설이는 이들이나, 구입해 놓고서도 문서작성 이외의 목적에 사용하지 못하는 이들에게, PC란 무엇이며, 어떤 목적에 어떻게 사용할 수 있는 것인가, 또 국어국문학을 전공하는 사람이 PC를 사용할 때 봉착하게 되는 문제가 무엇인가 등을 필자의 경험을 바탕으로 설명함으로써 컴퓨터의 활용도를 높이고, 나아가 국어국문학의 연구에 조금이나마 도움이 될 수 있도록 하고자 하는 것이다.¹⁾

우선 컴퓨터에 대하여 전혀 사전 지식이 없는 경우를 위하여 II장에서 컴퓨터의 종류와 기계적 구성에 대하여 설명한 후, III장에서 PC의 활용에 필요한 software에 대하여 유형별로 간단히 소개하면서 국어학 연구에 활용할 만한 software에는 어떤 것이 있는가를 필자나름의 경험을 토대로 제시한다. IV장에서는 국어학 연구에서 PC를 사용할 경우 부딪는 한글 처리 문제와 고어자 및 음성기호의 입·출력 문제를 기존의 software를 이용하여 해결하는 방법을 설명한다. V장에서는 실제 필자가 이용하고 있는 방법 몇 가지를 소개함으로써 PC의 활용 가능성을 검토하기로 한다.²⁾

- 1) 필자의 경우 1988년 3월 초 컴퓨터를 구입하여 주변에 PC에 대해 조언을 구할 상대가 하나도 없는 상황에서, 알려져 있는 software와 관련서적을 손에 닿는 대로 모아서 사용해 보고 국어학 연구에 활용할 수 있는 software를 확인하는 데에만 약 2개월이 소요되었다. 물론 필자의 능력과 노력이 부족한 면도 있었겠지만, 간단하게라도 국어국문학 연구에서의 PC의 효용성이나 이용 가능한 software를 소개하는 글이 있었다면 어느정도 시간과 금전의 낭비를 줄일 수 있었지 않았을까하는 생각을 하게 되었고, 이 글도 그러한 생각에서 쓰게 되었다. 앞으로 PC를 사용할 예정이거나, PC를 구입하고도 그 기능을 충분히 살리지 못하는 분들에게 가능하다면 시간과 금전의 허비를 줄일 수 있는 길을 안내하면서, 아울러 지금까지 필자가 PC를 이용해 작업해 온 과정의 일단을 소개함으로써, PC 사용자 사이의 정보 교환의 실마리를 잡아보고자 하는 것이다.
- 2) 필자는 PC의 구조나 프로그래밍 언어에 대해서는 전혀 아는 바가 없다. 다만 이미 소개되어 있는 software들을 접하면서 그것들을 활용하는 방안을 강구했을 뿐이다. 따라서 PC의 구조에 밝거나, 프로그래밍 언어를 잘 알고 있는 컴퓨터 사용자에게는 이 글의 내용이 너무 진부한 것으로 여겨질 수도 있을 것이다. 그럼에도 불구하고 이런 글을 쓰게된 까닭은 주1)에서 고백한 바 필자와 같은 시행착오를 겪을지도 모르는 초보자들에게 조금이나마 도움을 주고자 하는 것임을 다시한번 강조해 두고자 한다.

II. PC의 종류와 기본 구성

(1) 퍼스널 컴퓨터의 종류

컴퓨터는 기본적으로 0과 1이라는 두 숫자를 사용하는 전기적 신호의 조합을 통해 명령을 수행하거나 내부의 작업을 수행한다. 이 0과 1이라는 두 숫자의 조합 하나의 단위를 가리켜 bit라 하고, bit가 8개 모여 하나의 또 다른 단위 byte를 이루는데, 이 byte가 컴퓨터가 수행하는 작업의 기초 단위가 된다.

퍼스널 컴퓨터는 기계의 중앙 연산 장치의 처리 단위에 따라 8bit형, 16bit형, 32bit형의 세가지 유형으로 나뉘는데, 앞에서 이야기한 애플형 컴퓨터는 8bit형의 대표적인 것이고, 요즘 일반화되고 있는 IBM-PC XT, AT기종은 16bit형의 대표적인 것이며 이름 뒤에 386이라는 숫자가 붙는 PC나 Apple사의 MacIntosch등은 32bit형이라고 이해하면 된다.

이들 사이의 차이는 중앙 연산 장치의 처리 단위의 차이에 있다. 간단히 설명하면 8bit형 컴퓨터는 한 번에 2의 8제곱 가지 즉 256가지의 명령을, 16bit형은 2의 16제곱 가지의 명령을 구별할 수 있는 것이다. 즉 16bit형은 그 중앙 연산 장치의 기능이 8bit의 256배에 달하는 것이다. 이러한 차이에서 오는 것이 한글·한자의 처리 능력이다. 일반적으로 컴퓨터에는 영문자의 사용이 기본적이며, 한글의 사용은 부수적이다. 따라서 한글을 입·출력하려면 컴퓨터로 하여금 그것이 영문자와 다르다는 것을 인식하도록 하여야 하고, 대개의 경우 한 byte의 첫 bit를 이에 할당한다(첫 bit가 0이면 영문자, 1이면 한글이라는 식으로). 따라서 8bit형의 경우 한 byte(명령 단위)에서 사용 가능한 명령의 조합이 2의 7승 즉 128개 밖에 되지 않아 초·중·종성의 조합으로 되어 있는 한글을 표현하기 어려운 반면, 16bit형의 경우 나머지 15bit를 각각 5개씩 나누어 한글 초·중·종성에 배당함으로써 각각 32개의 문자체를 구분할 수 있게 하고 그들의 조합으로 한글을 수월하게 사용할 수 있는 것이다. 따라서 국어국문학 연구에서는 16bit 이상의 기종을 사용하는 것이 필수적임을 쉽게 이해

할 수 있을 것이다.

16bit형도 그 기능에 따라 몇 가지로 나뉜다. 우선 한 사람만이 사용할 수 있는가 아니면 2사람 이상이 한대의 기계로 동시에 작업할 수 있는가에 따라서 XT형과 AT형으로 나뉘며, 또 XT형은 중앙 처리 장치의 연산 속도에 따라서 Turbo형과 비Turbo형으로 나누기도 한다. XT형은 당초에 개발될 때는 4.77Mhz의 연산 속도를 가진 것이었는데 짐차 그 속도가 빨라지면서 요즘 10Mhz의 속도를 가진 Turbo형이 일반적으로 보급되고 있고, 회사에 따라서는 16Mhz의 속도를 가지고 있다고 선전하기도 하는데, 이것이 사실이라면 CPU의 처리 속도에 관한 한 AT급에 해당한다고 할 것이다. 그러나 XT와 AT의 차이가 CPU의 처리 속도 상의 차이에만 있는 것은 아니다. 앞에서 이야기한 대로 single-user용인가 multi-user 용인가 하는 차이와 함께 보조 기억 장치(후술 참조)의 기억 용량의 제한도 적지 않은 것이다.

XT형은 기본적으로 20M byte의 용량을 가지는 보조 기억 장치를 부착할 수 있고 최대 40M byte까지 확장이 가능하다. 그러나 AT급은 30M byte의 Hard Disk내장이 기본적이며, 최소한 60M byte, 기종에 따라서는 최대한 160M byte까지 보조 기억 장치를 확장할 수 있다. 이러한 기억 용량의 차이는 컴퓨터를 이용한 작업의 초기에는 크게 문제시 되지 않지만 자료의 입력 양이 많아질 수록 기억 용량의 부족이 심각한 문제로 대두되기에 간과할 수 없는 문제인 것이다.

필자의 경우 40M의 Hard Disk를 내장한 AT급 PC를 이용해 중세어 문헌 자료를 입력하여 분석하는 작업을 진행 중인데, 불경 언해서 중에서 비교적 그 양이 작은 것에 속하는 「般若心經」을 모두 입력하고 그것을 색인화한 결과 어간·어미 통합형의 색인 자료만 무려 1,204,224 byte즉 1.2M byte에 달하고 있다. 「능엄경」의 경우 그 양이 언해 부분만도 「반야심경」의 10배가 넘는바, 같은 비율로 계산 하더라도 최소한 10M 이상의 기억 용량을 필요로 할 것이고, 자료의 양이 늘어나는 경우라면 현재 사용하고 있는 40M의 내장 disk도 넉넉치 않다고 할 것이며, 이들 자료를 바탕으로 한 Data Base의 구축을 위해서는 최소한 지금의 2배 정도의 용량이

필요하다고 생각되는 것이다.

따라서 PC의 구입시 반드시 16bit 이상이어야 함은 물론 보조 기억 장치의 기억 용량 확대 가능성을 고려하여야 하며, 최소한 20M의 내장 disk를 가지고 있는 16bit형이 바람직하다고 할 것이다. 앞으로는 이러한 전제 위에서 16bit PC를 중심으로 설명하도록 하겠다.

(2) 16bit PC의 기본 구성

16bit 컴퓨터의 기계적 구성(hard ware)은 크게 다음과 같은 세부분으로 나뉜다.

가. 본체(System Unit Box)

컴퓨터의 두뇌에 해당하는 중앙 연산 장치(Central Processing Unit)와 주 기억 장치(Main Memory), 주변 장치 제어기(Device Controller)가 붙어 있는 CPU board, Monitor의 화면에 글자나 그림을 표현하는 방식을 제어하는 video board, 한글·한자의 입출력을 제어하는 한글·한자 board의 세가지 board와 전원 공급 장치, 그리고 후술할 보조 기억 장치를 제어하는 Disk Controller가 들어 있는 금속제 cabinet를 가리킨다. 또한 16bit형에서는 하나 이상의 floppy disk driver가 기본적으로 장착되어 있으며, Hard Disk를 내장하는 경우 별도의 Hard Disk Controller가 장착된다(후술 참조).

나. 입력장치

일반적으로는 타자기의 자판과 유사한 모습을 가지는 Key board가 사용된다. 사용자가 컴퓨터에게 어떤 동작을 하도록 명령을 내릴 때 사용하는 것으로, key의 숫자에 따라 84key, 101key등이 있으나 어떤 것이든 크게 문제되지 않는다. 키보드 이외에 게임용 프로그램을 사용할 때 쓰는 Joystic, image를 나타낼 때 유용한 mouse등의 입력장치가 있는데, 이들을 이용하기 위해서는 별도의 port(연결장치)가 있어야 한다.

다. 출력장치

컴퓨터에 의해서 처리된 명령이나 자료를 육안으로 확인할 수 있게 보여주는 장치이다. 일반적으로 TV와 같은 외형을 가지고 있는 모니터(Monitor)가 사용되는데, 글씨나 그림이 녹색 혹은 주황색의 단색으로

표현되는 Monochrome Monitor와 여러 색을 사용할 수 있는 Color Monitor가 있다. 기본적으로 국어학 연구에는 단색 Monitor를 쓰면 충분한데, Color Monitor를 사용하고자 하는 경우 별도의 video board가 필요하다.

(3) 16bit PC의 보조장치

가. 기억장치

컴퓨터는 기본적으로 모든 작업을 CPU board상의 Main Memory부분에서 수행한다. 그러나 어떤 특정 작업을 하는데 필요한 program이나 작업을 통해 입력된 data를 기록하는 데에는 별도의 도구가 필요하다. CPU상의 RAM에는 그 작업을 필요한 program을 기억시켜 놓을 여분의 공간이 없기 때문이다.³⁾ 따라서 보조 기억 장치가 없으면 컴퓨터는 아무 일도 할 수 없기에, 이들은 보조 장치라기보다 필수장치에 속한다 할 것이다. 여기에는 floppy disk driver와 hard disk가 있다.

1) Floppy Disk Driver⁴⁾

8bit형 PC에서는 자료를 기록하는데 음성용 카세트 테이프와 유사한 테이프를 쓰기도 하였으나, 16bit PC에서는 음반과 유사한 floppy diskette를 사용하고 있는바, 이 diskette를 꽂을 수 있는 장치가 disk driver이다. 최소한 하나의 disk driver가 필수적이며, 원만한 작업을 위해서는 두 개가 필요하다. floppy diskette는 3.5인치형과 5.25인치형의 두 유형이 있는데 일반적으로 IBM-PC에는 5.25인치형이 쓰인다. 5.25인치형도 1D, 2D, 2DD, 2

3) CPU board상의 기억 장치에는 RAM과 ROM이 있다. 이때 사용자가 쓸 수 있는 영역은 RAM이고, ROM에는 컴퓨터를 기동시키는데 필요한 최소한의 program이 들어 있다. 일반적으로 Main Memory라고 하면 RAM의 용량을 가리키는데, 요즘 보급형 교육용 16bit 컴퓨터라고 해서 싼 값에 파는 PC들은 대개 이 RAM의 용량이 256byte에 불과하다. 일상적인 작업을 수행하기 위해서는 최소한 RAM의 용량이 512byte는 되어야 하고 경우에 따라서는 640 혹은 그 이상의 용량이 필요한 것을 명심할 일이다.

4) 플로피 디스크에 대해서는 대개의 MS-DOS설명서에 나와있고, 컴퓨터매거진 1988년 11월호, 소프트웨어드 1988년 10월호에도 자세한 설명이 있으므로 참조할 것.

HD의 4종류가 있는데, disk driver에 따라 구별해서 사용해야 하므로 주의를 요한다. 일반적으로 XT형에는 2D형이, AT급에는 2HD형이 사용된다.

ii) Hard Disk⁵⁾

내장 디스크(fixed disk)라고 하는바, 이름이 시사하는 바와 같이 컴퓨터 system box 내부에 고정시켜서 사용하도록 만든 데이터 보관 장치이다. 자료를 기록·저장하는데에 사용한다는 점에서는 floppy diskette와 같은 것이지만, floppy diskette가 가지고 있는 기억 용량상의 제약을 극복하기 위해 개발된 것으로 다양한 용량의 여러 기종이 시판되고 있다. 그런데, 같은 16bit PC라 하더라도 XT형인가 AT형인가에 따라 각각 사용가능한 것이 다르므로 구입시 주의를 요한다. 16bit형 컴퓨터 중 XT호환 기종은 주로 20M, AT호환 기종은 30M의 Hard Disk를 주로 사용한다.

iii) Hard Disk Driver

하드 디스크는 그 기억 용량이 커서 자료의 보관에는 편리하지만, 그 가격이 비싸고 기종에 따라 확대 가능한 기억 용량이 제약이 있으며, 또 컴퓨터에 내장된 형태가 대부분이어서 이동성이라는 측면에서는 floppy diskette보다 불편하다. 이러한 불편을 해소하기 위하여 개발된 것이 Hard Disk Driver이다. 이는 플로피디스켓과 하드디스크의 장점만을 모든 것이라고 할 수 있는데, 하드 디스크를 플로피 디스켓처럼 cartridge 형태로 만들고 별도의 Driver에 꽂아서 사용하도록 함으로써 자료의 이동을 용이하게 하는 한편, cartridge만 더 구입하면 얼마든지 기억 용량을 확대할 수 있도록 한 것이다. 다만 cartridge 하나의 용량이 10M로 되어 있어서, 자료 file의 크기에 제약이 있다는 단점을 가진다. 국내에서도 시판되고 있는데, cartridge 자체는 10M용량을 가지는 것이 개당 5만원 정

5) 하드 디스크의 사용법에 대해서는 컴퓨터매거진 1988년 8월호, 소프트웨어드 1988년 11월호를 참조. 또한 하드 디스크의 사용법만을 따로 설명한 매뉴얼도 있다. 예를 들어 Berliner, Don(1986), *Managing Your Hard Disk*, Que Corporation.

도이지만 Driver 자체의 가격이 XT급 PC와 맞먹을 정도여서 구입하여 사용하기에는 너무 부담이 크다.

나. 출력장치

처리된 자료를 화면이 아닌 종이에 인쇄하여 사용할 때 사용하는 것으로 보통 문서를 인쇄할 때는 printer가, 설계도면 등 도형을 주로 인쇄하는 경우 plotter라는 장치가 이용된다. printer는 인쇄 방식에 따라 크게 두 유형으로 나뉘는데, 하나는 dot matrix 방식이고, 다른 하나는 Laser Beam Printer(LBP)이다.⁶⁾

전자는 타자기처럼 먹끈을 printer head의 pin으로 때려서 인쇄하는 것으로, head의 pin의 숫자에 따라 9pin과 24pin의 두 유형으로 나뉜다. 그런데 9pin의 경우 글자체의 제약 등으로 인하여 고어가 또는 한자를 인쇄할 수 없으므로 기본적으로 24pin을 택하게 된다. LBP의 경우는 복사기처럼 인쇄 용지의 인쇄할 부분에 열을 가한 후 그 자리에 인쇄 잉크가 묻게 한 후 고열로 구워내는 방식을 사용하는데, dot matrix printer가 line 단위로 인쇄해 가는데 비하여, LBP는 page 단위로 인쇄해 감으로써 소음도 적고 속도도 빠르다. 다만 그 가격이 일반 24pin printer의 3~5배에 달하므로 개인이 구입해 사용하기는 어렵다는 단점이 있다.

Ⅲ. PC에 사용되는 SOFTWARE

PC를 자동차에 비유한다면, 앞에서 이야기한 hardware는 자동차의 기계적 제부속이고, software는 자동차를 움직이는 연료, 운전기사, 정비공, 길안내자, 세차원 등 실제 자동차를 움직이는데 도움을 주는 제반 요소라고 할 수 있다. 이들 software에도 상당히 많은 종류가 있는바, 실제 프로그래머가 아닌 일반 사용자(end-user)의 경우 software의 성격을 분명히 파악하고 선택적으로 구입하여 그 기능을 충분히 익혀서 자신이 사용하고자 하는 목적에 맞게 사용하는 것이 PC활용의 효율성을 높이는 첫

6) 프린터의 종류와 각각의 기능에 대해서는 소프트웨어 1988년 12월호 참조.

제 조건이다. 여기서는 이러한 software중 대표적인 것들에 대하여 유형별로 간단히 설명하기로 한다.

(1) OS(operating system)용 software

OS용 software는 컴퓨터가 사람의 명령을 받아들여 작업을 수행할 수 있는 상태로 만들어 주는 program을 뜻한다. 여기에는 크게 single user용과 multi-user용의 두 가지가 있다. single user용에는 8비트용 CP/M과 16비트용 MS-DOS가 대표적인 것이고 multi-user용으로는 UNIX와 XENIX의 두 가지가 대표적인 것인데, UNIX와 XENIX는 16bit-AT이상의 기종에서 두 사람 이상이 한 대의 컴퓨터를 동시에 사용할 수 있도록 해 준다. 혼자서 사용하는 경우가 대부분인 우리로서는 MS-DOS가 일반적인 OS용 software가 된다.

(2) WP(word processing)용 software

문서작성시에 사용할 수 있는 program을 말한다. 기본적으로 컴퓨터를 타자기와 같은 기능으로 사용할 수 있게 하는 것이지만, 수정, 첨가, 삭제 등의 기능과 함께 문서의 다양한 편집을 가능하게 하는 여러 가지 기능을 아울러 가지고 있어서 훨씬 능률적인 작업이 가능하다. 국내에서는 한글(영문, 한자 포함)문서 작성용과 영문전용의 두가지 널리 쓰인다.

가. 한글문서 작성용 word processor

한글문서 작성용 워드프로세서 역시 크게 두 유형으로 나뉜다. 하나는 컴퓨터 제작 업체에서 자기회사의 기종에서 사용할 수 있도록 개발하여 보급하는 program이고, 다른 하나는 software개발 전문 업체에서 만들어 보급하는 것으로 대부분 기종에 관계없이 사용할 수 있도록 되어있는 것이다.

전자의 가장 대표적인 예가 삼보 컴퓨터에서 개발 보급한 보석글 I, 보석글 II이다. 국내에서 한글용 컴퓨터를 최초로 개발 보급하면서 영문작성용 워드프로세서를 한글용으로 개조하여 함께 보급한 것이 보석글 I이고, 그것이 기능을 대폭 확장하고 사용자(특히 초보자)가 사전 지식이 별로

없어도 사용할 수 있도록 프로그램 안에 도움말과 MENU를 추가하여 보급한 것이 보석글 II이다. 이 보석글 II도 두 차례에 걸친 수정 보완을 거쳐, 요즘에는 2D디스켓 3장으로 된 것이 보급되고 있다. 문서작성용 프로그램 중 주석(footnote)기능을 가진 유일한 것으로 알고있다. 이외에, 바른글(현대), 명필(고려)등의 프로그램이 보급되어 있는데 컴퓨터를 만든 회사에 따라 서로 호환성이 없으므로 유의하여야 할 것이다. 특히 뒤에서 이야기할 고어자나 한자의 입·출력이나 음성기호의 database에의 입력자료와의 호환성을 고려한다면 hercules board상에서 hgc.exe와 nkp.com라는 프로그램을 사용할 수 있는 것을 선택하여야 한다.

소프트웨어 개발 전문업체에서 개발한 워드프로세서는 한글 2000 word(한컴퓨터 연구소), 한글 팔란티어 워드프로세서(한국 팔란티어 주식회사)가 있다. 문서작성 전용으로는 문제가 없으나, 뒤에 이야기할 한글 dbaseII와의 자료 호환시에 문제가 발생할 수 있고, footnote기능이 없는 것이 국어국문학 전공자들에게는 아쉬움으로 남는다.

나. 영문작성용 WP

WordStar가 가장 일반적으로 보급되어 있고, Word Perfect V.4.2 PWP, Sprint등이 사용되고 있다. 영문작성용 이라고는 하지만 hgkey.com으로 기동되는 세운한글카드, hskey.com으로 기동되는 호산한글카드를 별도로 본체에 장착하는 경우 한글만은 사용할 수 있다. 또 chp.com으로 기동되는 삼보 한글·한자카드나 omni카드를 내장하면 한자를 쓸 수 있다. 다만 한글을 사용하는 경우 영문작성시에 사용할 수 있는 기능 중 상당부분을 사용할 수 없게 된다.

(3) 자료 처리용 software

가. DATABASE형 software

이는 일반적으로 card 작업을 대신할 수 있는 program이라고 이해하면 크게 벗어나지 않는 것이다. 국내에서 가장 많이 사용되는 것은 한글 dbaseII와 dBASEIII, DBASEIII+가 있다. 기본적인 원리는 같은데, 그

기능상 많은 차이가 있다. 국어국문학의 연구에는 이들 소프트웨어가 가장 유용한 것으로 판단된다. 하나의 커다란 file안에 (file은 dbase의 기초 단위가 되며, dBASEII는 두개의 file을, dBASEIII와 dBASEIII+는 10개의 file을 한꺼번에 사용할 수 있다). 일련번호로 정리되어 있는 card가 들어 있어서(이때 card의 역할을 하는 것이 record인데, dBASEII는 한 file당 65,535개의 record를, dBASEIII는 10억개의 record를 둘 수 있다), 그 카드에 기록되는 각 항목의 내용을(그것을 가리켜 field라 하는데, dBASEII는 각 record당 32개의 field를 dbaseIII·III+는 공히 128개의 field를 가질 수 있다) 사용자의 필요에 따라 검색, 분류할 수 있도록 만든 program이다.⁷⁾

dBASE와 유사한 기능을 하는 것으로 pfs file이라는 program이 있는데, 이는 원래 apple computer용으로 개발된 것으로, IBM에서도 사용할 수 있으나 한자를 차용하는데 제약이 있어 국어국문학 연구에는 크게 도움이 되지 않는다.

나. 통계 처리용 package program

앞에서 이야기한 워드프로세싱용 program과 dBASE, spread sheet의 기능을 하나의 program 안에 묶은 것으로, Symphony가 대표적이다. Symphony를 한글용으로 개발한 것이 Harmony이고, 보석글 II도 유사한 기능을 가지고 있다. 그러나 아무래도 독립된 program보다는 그 기능이 떨어지므로 이제는 별로 사용하지 않는 경향이다.

(4) Utility Program

워드프로세서나 자료 처리용 소프트웨어 이외에 컴퓨터의 사용을 보다 편리하게 하기 위해 개발된 제program을 통틀어 utility program이라고

7) apple 형 8bit 컴퓨터에서 dbaseII를 이용한 어휘자료의 처리방법이 정인상 (1985)에 소개되어 있으며, 거기에는 dbaseII의 기능에 대해서 비교적 자세한 설명이 있고, 이 글의 V장 내용 대부분은 dbase를 이용한 연구의 활용 가능성을 검토하는데 할애될 것이다.

한다.

여기에는 크게 두가지 유형이 있는데, 하나는 이미 개발되어 사용되는 software의 기능을 강화하여 사용자의 편의를 돕는 것이고, 다른 하나는 기존의 software와는 무관한 별도의 software로서 컴퓨터의 기능 자체를 높여주는 program이다.

전자의 대표적인 예로 Turbo Pascal Toolbox류, dbase compiler류 등이 있으며, 후자의 예로는 floppy diskette이나 Hard Disk의 사용을 돕는 ptools, copyII PC, Norton Utility 등과 사용자가 PC를 보다 효율적으로 사용할 수 있도록 해주는 Side-Kick, Superkey, DR. HALO, Turbo-lightening Hard Disk의 관리를 도와 주는 Disk Manager, Mace+ 등이 있다. 국내에서 개발된 utility program으로는, 보석글의 기능을 강화하는 HIRS2, 만능보석글, 한글·한자 단어변환 프로그램 등이 있고, 별도의 한글 카드가 없이도 한글의 입·출력이 가능하게 하는 K-한글한자 시스템, 조합형 한글과 완성형 한글 사이의 code변환이 가능하게 하는 여우꼬리·문어발, 한자 card에 내장되어 있지 않는 글자를 사용자가 만들어서 쓸 수 있게 하는 한자-MASTER등이 있다.

(5) 기타

Basic, Quick Basic, Turbo Pascal, C, turbo C, Fortran등 프로그래머가 일련의 작업을 수행하기 위한 program을 작성하는데 사용할 수 있는 프로그래밍용 언어, Auto Cad, Versa Cad, Orcad 등 설계작업시 도안을 편하게 할 수 있도록 도와주는 CAD·CAM용 software, 교육용 program, Game용 program, PC사이의 통신을 위한 program등이 있으나, 국어학 연구에 직접 도움이 되는 것이 아니므로 줄이기로 한다.⁸⁾

8) Auto Cad의 경우 한글 글자형의 개발에 원용되고 있고, 교육용 software 중 hypertext 기법을 사용한 것은 dbase보다 더 효율적인 것으로 소개되고 있는데 아직 필자가 직접 접해 보지 못하여 자세한 내용을 소개할 수 없다. 다만 dbase의 경우 문자 밖에 저장·검색·분류하지 못하는데 비하여, hypertext는 도형·문자·그림을 함께 저장·분류할 수 있는 것으로 소개되고 있다. 이에 대해서는 Fiderio, Janet(1988), "A Grand Vision"(컴퓨터 매거진 1988년 11월호에"하이퍼 텍스트의 세계가 밀려오고 있다"는 제목으로 번역)에 간략한 설명이 있다.

IV. 국어학 연구에 있어서의 PC의 활용상의 문제점

국어국문학 연구에서 PC를 활용하는 데에는 크게 두 방향이 가능하다. 하나는 문서작성기로 사용하는 것이고, 다른 하나는 자료의 정리·분류·검색에 응용하는 방법이다. 문서작성기능에 대해서는 길게 설명하지 않겠다. 왜냐하면 기본적으로 컴퓨터를 타자기와 같이 사용하되, 문의 수정·편집 등을 좀 더 용이하게 하는 여러 기능이 추가된 것으로 이해하면 충분하고, 다만 각자가 사용하는 software의 기능과 그에 대한 사용자의 숙지도에 따라 활용의 정도가 달라질 뿐이기 때문이다. 이보다는 자료의 정리 등을 위해 행하는 수작업에 의한 card의 작성·분류·정리 등을 어떤 방법으로 대신 할 수 있는가가 관심의 대상이 될 것이므로 그를 중심으로 논의를 진행하고자 하는데, 그에 앞서 PC를 국어학 연구에 활용하려 할 때 부딪는 문제인 한글·한자·고어의 처리에 대해 설명하기도 한다.

국어학 연구자가 PC를 사용하려 할 때 제일 처음 부딪게 되는 문제는 바로 현재의 PC에서는 고어자(현대 맞춤법에서 사용되지 않는 글자)·음성기호 등 특수문자의 사용이 전혀 고려되지 않고 있다는 점이다. 만약 고어자나 한자의 입·출력이 자유롭지 않다면 컴퓨터를 이용한 자료 정리는 현대어 자료에 국한 될 것인바, 국어학 연구에서의 PC의 활용도는 절반 이하로 떨어질 것이며, 그렇다면 굳이 PC를 익히느라고 시간과 돈을 들일 필요가 있겠는가 하는 의문이 이는 것이 당연한 것이다. 이 장에서는 이 문제의 해결책에 대해 설명하고자 하는데, 우선 컴퓨터에 있어서의 한글·한자의 입·출력 방법의 기본적 원리를 간단히 설명한 후 원전자료의 처리에서 가장 큰 문제가 되는 고어 및 특수 문자를 쓸 수 있는 방법을 소개하고자 한다.

(1) IBM PC 호환 기종에서의 한글 사용

IBM PC 도입 초창기('84년)부터 관심의 대상이 되어온 한글 문제는 기본적으로 컴퓨터가 로마자 사용을 그 입·출력의 기준으로 하고 있었다

는 데에서 출발한다. 즉 로마자는 풀어쓰기를 사용하는데 비하여 한글은 모아쓰기를 사용하며⁹⁾, 한 글자를 구성하는 획 수에 있어서 한글과 로마자는 큰 차이를 가지고 있어서 점을 모아 글자를 표현하는 방식을 사용하는 컴퓨터에 있어서 로마자 위주의 방식을 그대로 사용하는 경우 한글을 표현하기란 불가능한 일이었기 때문이다. 따라서 한글 한 음절을 인식하도록 하기 위해서는 국어에서 나타나는 모든 가능한 음절을 먼저 하나의 표로 작성한 후 그 음절 하나하나에 번호를 붙여 컴퓨터가 인식하도록 하고, 키보드 상에서 초·중·종성의 순서로 입력되는대로 그 인식표에서 찾아서 작업하도록 하는 한글 코드화 작업이 선행되어야 했고, 다음으로는 입력된 한글을 모니터 상에 나타낼 수 있게 하기 위한 video board의 개조가 이루어져야 했다.

monitor 상에 한글을 구현하는 문제란 간단히 이야기해서 한 글자를 나타내는데 몇 개의 점을 사용하느냐 하는 것과 어떤 video board를 사용하느냐 하는 문제를 말한다. IBM의 MDA(Monochrome Display Adapter)의 경우 영문자 하나를 나타내는 데에 9개의 가로점과 14개의 세로점으로 이루어지는 문자들을 이용했는데, 현재 대부분의 경우 한글을 나타내는 데에는 16×16의 틀을 사용하고 있다. 앞에서 이야기한 바, 한글의 한 음절은 구성하는 획이 영문에 비해 복잡하기 때문에 9×14의 해상도로는 구별되지 않는 글자가 생기기 때문이다. 따라서 MDA를 그대로 사용하는 경우 한글을 표현하기 위해서는 글자가 커져야 하고(바꾸어 말하면 점의 수를 늘려야 하고), 결국 한 화면에 나타낼 수 있는 글자의 수가 줄어들게 된다(한 화면의 점의 숫자는 일정하므로). 또 MDA의 경우 단순히 문자만을 표현할 수 있을 뿐 graphic은 불가능하여 game을 즐길 수 없다. MDA와 함께 개발된 CGA(Color Graphics Adapter)의 경우 해상도라는 면에서 MDA보다 떨어져 그래픽에는 사용할 수 있어도 한글을 표현하기

9) 이러한 차이점에 착안하여 만들어진 것이 김정수 선생의 '기울여 풀어쓰기 한글'이다. 이에 대해서는 박현철(1987), "허클레스용 기울여 풀어쓰기 한글", 마이크로 소프트웨어 1987년 3월호 참조.

에는 더 적합치 않았다. 이러한 단점을 보완하기 위해서 IBM MDA와 호환성을 가지면서 monochrome에서 graphic을 할 수 있는 hercules graphic card(HGC)가 개발되었고, 국내에서도 이 HGC를 이용하여 한글을 구현하는 경우가 많아졌다. 삼보 한글 DOS나 보석글 program안에 들어 있는 HGC.EXE라는 program이 바로 이 허큘리스 카드를 이용해서 한글을 사용할 수 있게 해주는 program이다.

이 HGC.EXE는 허큘리스 보드의 해상도를 한글을 표현하기에 적합한 해상도(640×400)로 바꾸어 주는 역할을 한다. 해상도란 모니터의 가로줄과 세로줄에서 문자나 그림을 표현하기 위해서 사용되는 점의 수라고 이해하면 되는데, 앞에서 이야기한 대로 한글 한 음절을 나타내기 위해서는 최소한 16×16의 크기를 갖는 문자틀을 필요로 하는 바, 640×400이라는 해상도는 가로로 40자, 세로 25행의 한글을 나타낼 수 있는 조건인 것이다.

HGC.EXE에 의해 video board의 해상도를 640×400으로 바꾸고 난 후에는 한글코드에 내장하고 있는 프로그램을 사용해서 바로 한글을 입·출력할 수 있고, 그 코드 안에 한자까지 수록하면 한자까지도 사용할 수 있다. 이른바 software에 의한 한글 사용 방식인 것이다. 이 방식의 한글 실행 프로그램의 예가 바로 THP.COM과 NKP.COM이다. 양자는 모두 삼보 컴퓨터 사이에서 개발한 것인데, 기본적인 한글·한자 코드 방식은 2byte 조합형이며, 코드 자체도 같되 후자는 전자의 프로그램 상의 단점을 보완하고, 글자체를 개선한 것이라고 할 수 있다.¹⁰⁾

이 software 방식의 한글 사용은 프로그램을 만들기는 비교적 쉽지만 그 처리 속도가 더디고 영문전용 프로그램은 사용할 수 없다는 단점이 있다. 이러한 문제를 해결하기 위해서 컴퓨터 업계에서는 hardware 적으로 한글을 사용할 수 있는 방안을 모색하게 되었고, 그 결과 NKP.COM과

10) 그렇다고 해서 NKP.COM이 모든 면에서 THP.COM보다 낫다고 할 수는 없다. 한글 dbaseII의 경우 THP.COM의 사용을 전제로 해서 개발된 것이어서 NKP.COM사용시에는 cursor(화면의 입력 위치를 표시하는 반짝이)가 사라지고 보이지 않는 경우가 있는 것이다 (후술 참조).

한글 코드에 맞춰 내장 card에 의해 한글을 사용하게 한 CHP.COM이 개발 보급되었고, 1988년 8월에는 IBM에서 개발한 EGA(Enhanced Graphic Adapter)를 이용하여 KSSM한글 코드와 CHP.COM 한글코드를 겸용할 수 있는 새로운 한글카드가 소개되었다.

이외에도 대우·삼성·금성·현대등 대기업에서도 자기 나름대로의 한글 카드를 개발하여 hardware 방식의 한글 사용법을 택하고 있다. 그러나 이들은 모두 고어자의 사용은 염두에 두지 않았기 때문에 다음에 이야기할 고어자나 특수문자의 사용은 불가능하다.

이러한 한글 구현 방식상의 문제이외에도 한글·한자의 사용과 밀접한 관계를 갖는 것이 한글 코드의 문제이다. 한글 코드란 앞에서 설명한 대로 컴퓨터로 하여금 한글을 인식하게 하는 하나의 표를 말하는데, 처음 각 컴퓨터사마다 서로 다른 방법을 채택, 프로그램의 호환은 물론 어떤 레벨에서도 자료의 호환성이 없다. 한 회사의 PC를 이용해 작성한 문서를 다른 회사의 컴퓨터를 이용해 읽을 수 없다는 뜻이다.¹¹⁾ 여기에 앞에서 이야기한 대로 monitor상에 한글을 나타내는 방식도 달라, 타사의 기계사이의 호환성이란 아예 생각도 못하는 실정인 것이다.

이런 상황 하에서 어떤 기종을 선택할 것인가가 소기의 목적을 달성할 수 있을 것이냐 하는 점과 무관할 수 없다. 자칫하면 우리가 필요로 하는 software(한자 MASTER)를 사용할 수 없는 경우도 있기 때문이다. 결론부터 이야기하면, 필자와 같은 목적으로 컴퓨터를 사용하는 경우라면, HGC를 이용하면서, 한글은 THP.COM혹은 NKP.COM을 이용하는 기

11) 이러한 문제점을 보완하기 위하여 1987년 정부에서는 통일 한글 코드를 확정 발표하였다. 뒤 늦은 감은 있으나 그런대로 앞으로의 일을 위하여 다행한 일이라고 할 것이다. 그러나 현재에 있어서도 각 컴퓨터 회사마다 자신의 고유한 한글 코드를 계속 사용하고 있어서, 한글 코드 구성 방식만 하더라도 2바이트-7비트 완성형, 2바이트 완성형, 2바이트-8비트 조합형 등 여러가지가 사용되고 있고, 코드 방식은 같아도 코드 자체가 다른 경우가 있으며, 또 통일 코드를 사용하는 software가 많지 않아서 완전한 한글 코드의 통일은 아직 어렵다고 할 것이다. 정부에서 발표한 KSSM코드는 한글 2350자, 한자 4888자, 기타 특수문자와 사용자 정의 영역을 가지고 있는 2바이트 완성형이다.

중이어야 하며, 별도의 한자 card가 내장되어 있어야 한다.

(2) 한자 · 고어자 · 특수문자의 사용

앞에서 이야기한 기준을 선택하는 경우 사용할 수 있는 한자는 모두 3640자로 제한되는데, 이러한 제약은 국어학 전공자나 국문학 전공자에게나 공히 적지 않은 부담이 된다. 필요로 하는 한자의 수보다 너무 적은 것이다. 그러나 한문학 전공이 아닌 한, 한자의 부족은 사용자가 어떻게 자료를 정리하느냐에 따라서 크게 어느 정도 극복할 수 있으리라는 것이 필자의 생각이다. 물론 다소의 불편을 있겠지만 PC의 활용가능성을 배제할 정도는 아닌 것이다.¹²⁾

이보다 더 심각한 것은 고어자(현대국어 철자법에서 사용되지 않는 글자)와 음성기호 등의 특수문자를 어떻게 사용하느냐 하는 문제다. 이러한 문제는 국어학이나 국문학을 전공하는 이들에게 국한되는 것이라는 점이 더욱 더 문제의 해결을 어렵게 한다. PC를 사용하는 사람들이 공통적으로 느끼는 어려움이라면 PC에 대해 잘 아는 이가 해결해 줄 수도 있을 것이나, 이 경우는 국어국문학을 전공하면서 PC를 사용하는 사람들 자신만의 문제인 것이다. 최근에 들어 이 문제에 대하여 전산국어학회에서 관심을 갖기 시작하여 keyboard상에서 고어를 그대로 입력할 수 있는 방법을 모색하고 있으나, 상업성이 적은 관계로 기존 업계에서는 큰 관심을 가지지 않고 있다.¹³⁾ 따라서 Hardware를 이용하여 방법은 수년내에 개발되기 어려우리라는 것이 일반적인 생각이다. 결국 고어자나 PC에 내장되

12) 국학자료의 전산화에 기본적으로 몇 자 정도의 한자가 필요한가는 아직 구체적 통계가 없어서 확실하게 이야기 할 수 없다. 다만 조선왕조실록을 대상으로 한 샘플조사 보고가 있어서 참고가 되는데, 거기서는 대개 1만자 정도가 필요하다고 보고있다. 김기협(1983), "한문자료의 컴퓨터 처리 방법 연구", 「중국학지」 제 1집, 계명대학교 중국학 연구소.

13) Key board상에서 'ㅏ'와 'ㅑ'를 사용하는 글자를 직접 입력할 수 있는 새로운 한글인 nnkp.com이 최근 개발되었으나, 입력된 자료를 프린터로 출력할 수 있는 프린터의 한글 module이 함께 개발되지 않은 상태여서 현재로는 사용이 불가능한 것으로 들었다.

는 한자 module에 들어있지 않은 한자나 음성기호 등 특수문자의 사용은 software에 의존하는 수 밖에 없는 것이다.

우선 고어자의 입·출력에 사용할 수 있는 software에 대해 살펴 보기로 하자. 필자가 알기로는 현재 유통되고 있는것 중에서 이러한 목적에 쓸 수 있는 유일한 software가 ‘한자 MASTER’라는 프로그램이다. 원래 이것은 3640자라는 한자의 부족을 보완하기 위해서 한자 코드에 들어있지 않은 글자를 사용자가 만들어 쓸 수 있도록 개발된 것인데, 일찍부터 PC를 사용하는 국어학 연구자들 사이에서 고어자를 사용하는 도구로 쓰이고 있다. 이 ‘한자 MASTER’는 모두 816자를 만들어 쓸 수 있도록 되어 있는데, 그 원리는 비교적 간단하다. 이른바 Font Editor(글자꼴 개발 프로그램)을 내장하되, 24핀 프린터에서 사용하는 글자체(한 글자당 24×24의 해상도를 가진다)와 화면에 나타내는데 사용하는 글자체(16×16의 해상도를 가진다)를 각각 별도로 만들 수 있도록 하고, 만들어진 글자꼴들을 USERFONT부분에 담아 두었다가 software에 의한 한글을 사용할 때 이들 글자를 먼저 RAM에 올린 후(즉 컴퓨터의 CPU에 기억시킨 후), 한자를 사용할 때와 마찬가지로 코드자를 이용해 필요한 글자꼴을 불러다 쓰는 방법인 것이다. (한자 ‘國’의 경우 ‘국’을 코드로 해서 호출되는 바 이때 ‘국’을 ‘國’의 코드자라 한다). 물론 RAM에 기억시킨 부분은 전원을 끄면 지워지므로 매번 사용시마다 다시 기억시켜야 한다. 이러한 방식이 바로 RAM상주 프로그램인 것이다.

좀 궁색하긴 하지만 이 프로그램을 활용하면 그런대로 고어의 입·출력이 가능하기에 많은 사람들이 이 방식을 사용하는 것으로 안다. 다만 고어에서 나타날 수 있는 음절 수는 모두 2,000이 넘는데 비해 이 software의 사용자 정의 문자 수가 816자에 불과해 고어의 음절을 모두 나타낼 수가 없다.¹⁴⁾ 따라서 출현 빈도가 높은 것을 위주로 만들어 사용하고,

14) 전산국어학회에서 이 문제의 해결을 위해 프로그램의 개발 회사인 삼보컴퓨터사에 의뢰 사용자 정의 문자의 수를 1224자로 늘린 새로운 version을 만들었는데 프로그램상의 오류 수정과 보급 방법 상의 문제등으로 아직 시중에는 보급되지 않고 있다.

빈도수가 낮은 것은 별도의 표시를 해서 사용해야 하는 불편이 있다.

음성기호 등 특수 문자의 입출력도 최근에 개발된 '만능보석글'이라는 유틸리티 프로그램에 의해 어느 정도 해결된 것으로 보인다. 이 프로그램은 기본적으로 LQ 800, LQ 1500등 24핀 프린터가 가지고 있는 사용자가 정의 문자 사용법¹⁵⁾을 바탕으로 개발된 것으로, 현재 시판되고 있는 프로그램 디스켓 안에는 우리가 사용하는 보통글자 이외에 다음과 같은 6가지 문자체가 들어 있다.

보통문자	1 2 3 4 5 6 7 8 9 0 - = q w e r t y u i o p [] a s d f g h j k l ; ' z x c v b n m , . /	! @ # \$ % ^ & * () _ + ; Q W E R T Y U I O P { } A S D F G H J K L : " Z X C V B N M < > ?
수학문자	f d f φ J [l] φ ± + ≠ θ κ ε ρ τ [] c ω π f x α σ δ ϕ γ η φ κ λ ρ ε ς ξ x ∇ β υ μ λ ∙ -	→ ↑ ← ↓ ∩ ∪ ≡ ≠ - ∅ ∩ ≈ ∃ √ ∅ ≠ ∞ ∞ ∏ ∫ ∥ V Σ Δ Γ ∫ φ ∑ Δ ∅ ε [↑ 1 2 ∇ [≤ ≥]
이탤릭	1 2 3 4 5 6 7 8 9 0 - = q w e r t y u i o p [] a s d f g h j k l ; ' z x c v b n m , . /	! j l φ θ - @ " () - + ∞ Q W E R T Y U I O P φ φ A S D F G H J K L : " Z X C V B N M () ?
심플문자	∞ ∈ ∅ ∅ - / + ∇ ∅ / ∅ ∅ # < [∇ ∪ ↑ ∩ > ± √ ∪ ∅ - / f [< > < ∥ ∅ κ ≈ ‡ ∅ [∅] ≠ [∅] ∅ ∅ ∅	→ - ↓ ∅ « (\) ∅ ∅ ∅ ∅ ∅ ∅ ≡ ≥ ∅ ∅ ≤ ≈ ≥ ∅
기호문자	1 2 3 4 5 6 7 8 9 0 - = Q P φ - F R H Γ Δ ∅ } = o - J] ↓ / [{ } ∅ Z Q Γ N [K ∅ ∅ ∅ ∅	1 2 3 4 5 6 7 8 9 0 " +] Q ∅ ∅ ∅ ∅ ∅ ∅ ∅ ∅ ∅ ∅ ∅ [∅] () ∅ ∅ ∅ ∅ ∅ ∅ ∅ ∅ ∅ ∅ ∅ ∅ ∅ ∅ ∅ ∅ ∅ ∅ ∅
필기체	φ Q φ α α α α α α α α α α α α z x c v b n m ∅ ∅ ∅ ∅	/ ∅ ∅ ∅ ∅ ∅ ∅ ∅ ∅ ∅ ∅ ∅ Q W E R T Y U I O P φ φ A S D F G H J K L ∅ ∅ ∅ ∅ Z X C V B N M ∅ ∅ ∅ ∅
외국문자	∅ ∅ ∅ ∅ ∅ ∅ ∅ ∅ ∅ ∅ ∅ ∅ η υ π θ κ π λ φ ε ζ ∅ ∅ Δ ∅ ∅ ∅ ∅ ∅ ∅ ∅ ∅ ∅ ∅ ∅ ρ ∅ ∅ ∅ ∅ ∅ ∅ ∅ ∅ ∅ ∅ ∅	ü á ä å ä ≥ ä è è / ÿ τ ∅ ∈ N ∅ ∅ ∅ ∅ ∅ ∅ ∅ ∅ ∅ í ç ñ ñ ∅ ∅ ∅ ∅ ∅ ∅ ∅ ∅ Γ ∩ ú ∅ ∅ « í ∅ ∅ f

15) 이른바 download 방식이라는 것으로 프린터 구입시 딸려오는 사용 설명서에 그 방법이 자세히 나와 있다. '만능보석글'의 사용법과 원리에 대해서는 이해용(1988), "프린터의 다운로드 기능을 이용한 만능보석글", 마이크로소프트웨어 1988년 9월호 참조.

이 프로그램은 특수문자의 사용에 있어서는 한자 MASTER보다 나은 것으로 생각된다. 한자 'MASTER'의 경우 2byte 단위로 글자를 만들어 사용하는 데 비해 이것은 한 글자당 1byte를 사용함으로써 다른 로마자와 그 크기가 같고, 입력시에도 문자체 선택 Key만 누르면 직접 키보드 상에서 입력되므로 코드자에 의한 전환 방식보다 편리한 것이다. 다만 이것은 NKP.COM과 24핀 프린터의 사용을 전제로 하며, 또 보석굴 II에서만 사용하도록 되어 있어 자료정리용 software에서의 활용을 위해서는 별도의 수정이 필요할 것으로 생각된다.

V. 국어학 연구에서의 활용의 실제

국어학 연구에 있어서의 자료 정리는 크게 두 유형으로 나뉠 수 있다. 하나는 자신이 참고로 한 논저의 제목과 중요하다고 생각되는 내용의 요약 부분을 하나 하나 카드에 정리해 두었다가 필요할 때 찾아볼 수 있도록 하는 것이고, 다른 하나는 기존의 문헌이나 연구자 자신이 조사하여 정리한 원전 자료를 카드화 하였다가 필요할 때 찾아보는 일이다. 이러한 작업은 물론 수작업으로 할 수도 있는 일이지만 카드의 양이 사천장, 오천 장을 넘어가면, 그것을 정리하여 분류·보관하는 일만 해도 여간 큰 문제가 아니다. 이러한 경우 PC를 활용한다면 우선 카드의 양에 압도당할 이유도, 또 카드의 정리에 시간을 낭비할 필요도 없어지고, 각각의 카드에 어떤 내용이 들어 있는가를 손쉽게 신속하게 찾아볼 수 있으므로 훨씬 연구의 효율성이 높아지게 된다. 다행히도 PC에 사용할 수 있는 software중에는 이러한 목적에 사용할 수 있는 것들이 이미 개발되어 있어서, 사용자가 그 software의 기능만을 확실히 익힌다면 아주 효율적으로 연구에 이용할 수 있다.

여기서는 그러한 목적에 사용될 수 있는 software중 대표적인 것에 속하는 dbaseII와 dbaseIII·III+를 활용한 작업의 실제 예를 들기로 한다.

(1) 논저목록 정리 및 내용 요약 정리에 있어서의 dbase활용

우리가 어떤 글을 읽고 후에 그것을 참고할 필요가 있어서 카드에 정리한다고 할 때 제일 먼저 생각해야 할 것이 카드를 어떤 형태로 정리할 것인가 하는 점이다.

[그림 1] 논저 내용 요약용 카드

· 분야별 일련 번호 _____

논문 제목	필자	출전(출판사) 및년도	문제점(중심어)
페이지	내용 요약		
참고사항(관련 논문)			

[그림 1]은 필자가 컴퓨터를 사용하기 이전에 개인적으로 만들어서 사용하던 카드 양식인데, 사람에 따라 다소 차이는 있겠지만 논문을 읽고 정리해 둘 때에는 대개 이와 비슷한 양식을 사용할 것이다. 문제는 이러한 양식의 카드를 만들어둔다 하더라도 실제 필요할 때 필요한 카드를 손쉽게 찾기 어렵다는 데 있다. 즉 누군가가 훈민정음의 초성체계에 대해 쓴 글을 읽어었는데 그 글의 제목이 생각나지 않으면 필자명으로 찾아야 할 것이고, 이러이러한 내용의 글을 읽는 적이 있는데 필자나 논문의 제목이 생각나지 않으면 문제점(중심어)를 이용해 찾아야 할 것이다. 더구나 어떤 주제를 다룬 논문을 모두 확인해 볼 필요가 있을 때에는 카드 전체를 뒤져보아야 가능한 것인 바, 그에 소요되는 시간과 정력의 낭비도 간과할 수 없는 것이어서 결국 카드작업을 포기해 버리는 경우도 생기는 것이다. 뿐만 아니라 카드의 크기에 따라 안에 정리할 수 있는 내용의 양이 제약되므로 종이카드를 사용하는 방법은 여러가지 면에서 효율성이 적은 것이다.

그러나 이러한 어려움은 PC를 사용하는 경우 문제가 되지 않는다. dbaseII나 dbaseIII+라는 프로그램은 이런 경우를 위해서 만들어진 것이

라 해도 과언이 아니기 때문이다.¹⁶⁾ 우선 이 경우에 사용할 수 있는 dbase III+의 data file의 구조를 제시하고 그것을 중심으로 설명해 가기로 한다.

논저 내용 요약 database file의 구조(논문의 경우): file명 JOURNAL.DBF

FIELD	FIELD—NAME	TYPE	WIDTH	FIELD	FIELD—NAME	TYPE	WIDTH
1	NUMBER	C	6	9	VOLUME	C	4
2	TITLE	C	32	10	PAGE	C	10
3	SUBTITLE	C	32	11	WRITER	C	30
4	JOURNAL	C	32	12	FIELD	C	16
5	P—or—E	C	20	13	TOPIC	C	20
6	YEAR	C	4	14	KEY1	C	20
7	MONTH	C	2	15	KEY2	C	20
8	DATE	C	2	16	CONTENTS	M	10
TOTAL							261

우선 이해를 돕기 위해 위의 data file의 구조와 [그림 1]의 논저 내용 요약용 카드의 내용을 비교해 설명하기로 하자. III장의 소개에서 dbaseIII+에서 하나의 file은 10억 개까지의 record를 가질 수 있다고 했는데, 이때 record 하나 하나는 바로 카드 한 장의 역할을 한다. 결국 file이란 카드 보관함의 역할을 하는 것인데 dbaseIII·III+의 경우 그 크기가 10억장에 달하므로 거의 무제한이라고 할 것이다. 이 Journal.dbf의 각 record는 바로 위에서 제시한 16개의 field로 이루어진다(dbaseIII·

16) 앞에서 필자가 추천한 기종을 선택하는 경우 dbaseIII+에서는 한글 사용이 불가능하다. 바로 software적 처리에 의한 한글 사용의 단점의 단적인 예다. 그러나 전혀 해결방안이 없는 것은 아니다. 그 기종에 hskkey.com으로 기동되는 호산 한글 카드를 별도로 부착하면 dbaseIII+에서의 한글 사용이 가능해지는 것이다. 또, III장에서 소개한 'K-한글·한자'라는 프로그램을 이용하면 별도의 한글카드가 없어도 dbaseIII+의 사용이 가능하다.

III+는 한 record안에 128개까지의 field를 가질 수 있다). 즉 이 field는 바로 논저목록카드의 각 항목에 해당한다고 할 수 있다. 예를 들어 data file의 구조 부분에 있는 'NUMBER'는 카드의 '일련번호'에 해당하는 것이고, 'TITLE'과 'SUBTITLE'은 '논문제목'을 두가지로 나눈 것이다. 그 이하 각 field의 내용은 field name을 보면 알 수 있을 것이다. 다만 FIELD는 논문이 주제로 삼고 있는 국어학의 하위 분야를, TOPIC은 다루고 있는 주제를 각자의 분류 방식대로 기록하고, KEY1·KEY2는 그 요약 대상이 되는 부분의 내용을 대표할 수 있는 적당한 단어 혹은 어귀를 기록하는데 사용하며, CONTENTS는 직접 논문의 내용을 기록하거나, 그 요약을 정리하는데 사용한다는 사실만 부연하기로 한다.

FIELD-NAME 다음 column의 TYPE는 해당 field의 성격을 나타내는 것으로, C는 해당 field가 문자로 이루어지는 것임을, M은 Memo field임을 가리킨다(후술참조). TYPE다음의 WIDTH는 그 field의 길이를 가리키는데, TITLE의 32는 제목 부분에 들어갈 수 있는 글자의 수가 영문자로 32자, 한글로 16자 임을 뜻한다. 결국 이 data file의 전체 크기는 TOTAL에 나타난대로 261byte 즉 한글 130자를 사용할 수 있는 크기인데, 주의할 것은 field type가 M인 CONTENTS의 크기이다. 여기서는 10byte인 것으로 되어 있으나, 실제 Memo field의 크기는 4,000byte 즉 한글 2,000자를 쓸 수 있도록 되어 있어 내용을 요약하기에는 충분한 것이다.

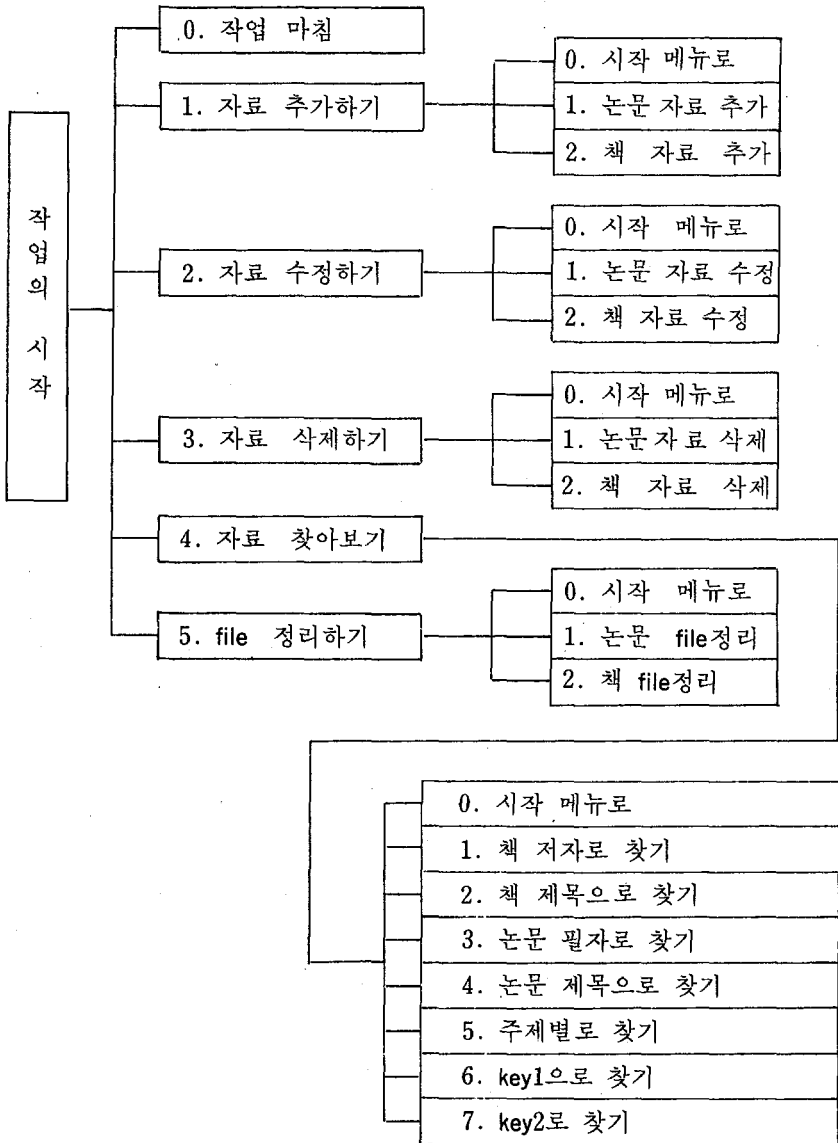
이상과 같은 대비를 통해서만도 dbase에 의한 논저 내용 요약 카드의 작성이 종이카드를 사용하는 것보다 훨씬 유리하다는 사실을 알 수 있다. 실제 종이 카드를 사용하는 경우 카드 내용 항목의 수가 10개를 넘어가면 사실상 그 활용 가능성이 없는데 비하여, dbaseII의 경우는 32개까지, dbaseIII·III+의 경우는 128개까지 가능하므로 비교가 되지 않고, 카드의 크기에 있어서도 dbaseIII나 III+를 이용하면 거의 원고지 20장(4,000)에 해당하는 분량을 기입할 수 있어(Text field 4,000 byte, Memo field 4,000 byte), 크기의 제약이 없다고 할 것이기 때문이다.

그러나 dbase 이용의 유리한 점은 그 항목의 다양성이나 카드의 크기에

있는 것이 아니다. 중요한 것은 dbase의 기능상 각 field(즉 카드의 각 항목)에 대한 내용 검색과 분류가 아주 빠른 시간에 이루어 진다는 것이다. 필자의 경험을 예로 들면, 약 4,500장의 종이 카드를 3가지 방법으로 재배열해서 그 내용을 확인하는데 꼬박 사흘 밤낮을 매달린 기억이 있는데, 같은 작업을 dbase를 이용한다면 그 결과를 print해 내기까지 약 3시간이면 충분한 바, 작업의 효율성이라는 측면에서는 도저히 비교되지 않는 것이다. 거기에다가 dbase의 명령어와 프로그래밍을 조금만 익히면 이러한 검색과 그 결과의 인쇄를 한꺼번에 처리할 수 있는 프로그램의 작성이 용이한 바, 작업의 편의를 극대화 할 수 있는 것이다. 그 한 예로 다음에 필자가 앞의 data file과 도서 목록의 data file을 이용하여 자료의 정리에 이용하고 있는 방법을 설명하기로 한다. 이는 dbaseIII 언어를 이용한 간단한 프로그램의 활용인데, 지면관계 상 프로그램을 전부 실지는 못하고 우선 기본적인 작업개념도를 통해 프로그램의 흐름을 개략적으로 보이고, 그 흐름 중 중요한 부분의 내용을 제시하여 기본원리의 이해를 돕도록 한다. 그 개략적인 작업 순서도는 [그림 2]와 같다.

프로그래밍이란 어떤 작업을 하고자 할 때 사용되는 여러 명령을 작업할 때마다 하나씩 입력하지 않고도 자동적으로 처리될 수 있도록 한 것을 의미하는 바, dbaseIII+언어를 이용한 프로그램 작성은 전문적 컴퓨터 언어를 전혀 알지 못하더라도 dbaseIII+를 이용한 작업에 사용되는 명령어만을 이용해 가능한 것으로 사용자 지향적 프로그램이라는 dbaseIII+의 강점을 잘 보여주는 것이다. dbase 언어를 이용한 프로그래밍의 기본은 자신이 data base file을 이용해 하고자 하는 작업의 순서를 정확히 정하는데 있다. [그림 2]에서 보듯이 작업은 앞에서 구조를 보였던 Journal.dbf와 도서목록 database file인 Book.dbf를 이용해서 자료를 추가할 것인지 아니면 이미 입력된 자료를 수정 혹은 삭제할 것인지 또는 입력된 자료의 내용을 검색해서 확인할 것인지 혹은 일부를 삭제한 입력내용이 있을때 완전히 file을 정리할 것인지 하는 작업의 내용을 선택하는데에서 출발한다. 그러한 작업으로 들어가게 하는 것이 부록의 [I. 시작

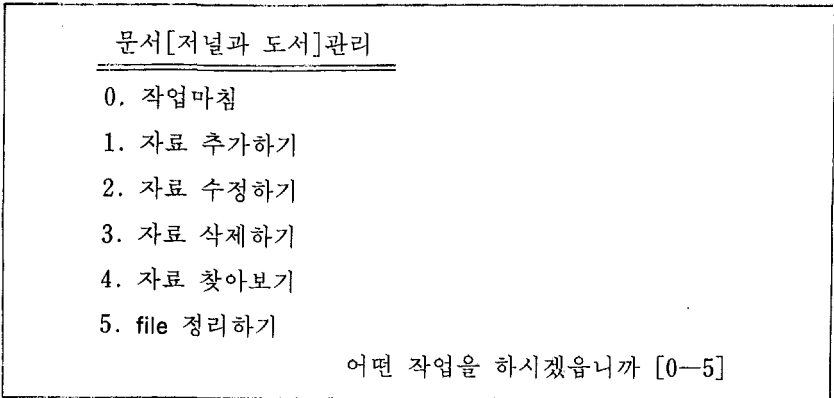
[그림 2] 논저목록 정리 및 내용요약 정리 검색 프로그램의 흐름



메뉴 프로그램]이다.¹⁷⁾

dbaseIII+를 기동시킨 후에 do RESTART라는 명령을 주면 [I. 시작 메뉴 프로그램]인 RFSTART.PRG가 기동되면서 바로 [그림 3]과 같은 화면 내용이 나와서원하는 작업중의 하나를 선택할 수 있게 하는 것이다.

[그림 3] 작업 시작 화면



이 [그림 3]의 메뉴에서 4를 선택하면 바로 부록의 [II. 자료 찾아보기 Menu Program]이 선택되어 실행되면서 자동적으로 [그림 4]의 화면이 나타난다. [I. 시작 메뉴 프로그램]의 Do Case 부분 중 Case selectnum=4가 수행되는 것이다. 이 [그림 4]의 화면에서 '0'을 누르면 [그림 3]의 화면이 다시 나오고, [그림 3]에서 '0'을 선택하면 dbase의 초기 상태로 되돌아간다.

[그림 4]의 메뉴에서 '4'를 선택하면 화면에 "찾으려는 논문 제목을 넣으세요 : " 라는 문구가 나와서 사용자가 ' : '의 뒷부분에 찾고자 하는 논문 제목의 일부만 넣어주면 그러한 단어를 갖고 있는 모든 논문들을 필자, 게재지, 호수, 페이지 등과 함께 화면으로 직접 프린트되어 나오도록 한 것이 바로, [III. 논문 제목으로 찾기 프로그램]이다. 결국 [일의 종류 선택]=>[자료 찾아보기]=>[논문 제목으로 찾기]=>[출력]이라는 일련의 과정을 자동적으로 진행하도록 한 것이 프로그램 I, II, III인 것이

17) 프로그램들은 이 글의 뒤에 부록으로 실는다.

[그림 4] 자료 찾아보기의 메뉴 화면

자료 찾아 보기

0. 시작 메뉴로
1. 책 저자로 찾기
2. 책 제목으로 찾기
3. 논문 필자로 찾기
4. 논문 제목으로 찾기
5. 주제별로 찾기
6. key1으로 찾기
7. key2으로 찾기

어떤 작업을 하시겠습니까 [0-7]

다. 필자가 사용하는 논저 목록 정리 및 내용 요약 정리용 프로그램은 부록에 예시한 프로그램 I, II, III 이외에 10여 개의 하위 프로그램과 앞에서 예시한 database file, 그리고 그에 따른 index file로 이루어져 있는데, 그 기본적 원리는 위에서 설명한 방식을 반복하여 일련의 작업이 이루어 지도록 한 것이다. 이 프로그램을 이용하면 자료의 양이 얼마가 되든 그 내용을 5분 이내에 검색하여 화면으로 확인할 수 있고, 별도로 인쇄하여 참조할 수도 있다. 처음 이 프로그램을 가동시켜 보고 나서, 4,500매의 카드를 검색·정리하는데 사흘 밤낮을 매달렸던 필자로서는 그저 경이로울 따름이었다.

혹자는 프로그램을 만들기까지 시간이 그만큼 많이 들지 않는가 하고 반문할 수도 있다. 그러나 여기의 프로그램은 이미 시중에 공개된 것을 참조하여 필자 나름대로 약간 수정한 것에 지나지 않는다.¹⁸⁾ 그렇게 하는데 소요된 시간은 약 일주일, 카드정리를 위해 보내는 시간에 비하면 결코 긴 시간이 아닌 것이다.

18) 이 프로그램을 작성하는데 참고한 서적은 다음 두 가지이다.

이기성·탁연상 공저(1988), [실무자를 위한 dbaseIII plus], 영진출판사.

(2) 원전 자료의 정리

현대어 자료가 아닌 중세어 혹은 근대어 그리고 漢籍 자료의 정리는 앞에서 이야기한 대로 dbaseII(한디비)를 사용해야 한다.¹⁹⁾ 그 이유는 IV장에서 밝힌 바, 고어의 입력에 사용할 '한자 MASTER'가 NKP.COM이나 THP.COM 등의 software 한글에서만 사용 가능하기 때문이다.

이 dbaseII는 앞에서 본 dbaseIII+보다 여러가지로 그 기능이 떨어진 다. 첫째로 메모필드가 따로 없고 문자 필드만 사용해야 한다는 것이 가장 큰 문제가 되고, 둘째로 한 레코드 당 사용 가능한 문자 수가 1,000 byte(한글 500자)에 불과하며, 한 필드의 길이가 254byte(한글 127자)로 제약된다는 점이 문헌자료 정리상의 큰 어려움이 된다. 세째로 index file의 수가 2개를 제약되어 있어 검색과 분류를 다양하게 하기 어렵다는 점이 data file 활용상 제약이 된다. 그러나 data file의 구조를 잘 조정하고 또 file name을 다양하게 하면 어휘자료건 문장 자료건 대개는 정리가 가능하다는 것이 필자의 생각이다.²⁰⁾

김우용·김영미 편저(1988), [고급 사용자를 위한 dbaseIII Plus], 영진출판사.

양자 모두 문헌관리에 응용할 수 있는 프로그램을 싣고 있는데, 적지 않은 오자가 있어 처음 그대로 입력했다가 프로그램이 제대로 실행되지 않아 상당히 애를 태웠다. 그러나 그 오자들을 찾아 내면서 dbase프로그램을 익혔으니 오히려 도움이 되었다고도 할 것이다.

19) 주16)에서 이야기한 것처럼 'K-한글·한자'를 사용하면 dbaseIII+의 사용이 가능하다. 그러나, 'K-한글·한자'의 경우 사용자 정의 문자 영역이 612자에 불과해서 고어자의 정리에는 아무래도 많이 부족하고, 한적 자료의 경우라면 한자가 4888자가 들어있어 어쩌면 더 적합할지도 모른다(NKP.COM의 경우 3640자 이므로). 'K-한글·한자'에 대해서는 컴퓨터 매거진 1988년 6월호 참조.

20) 물론 이 모든 경우에 대해 필자가 직접 작업에 임해 본 것이 아니어서 작업상 아무 문제가 없다고 자신있게 이야기할 수 있는 처지는 아니다. 필자의 경우 다음에 이야기할 문장 자료의 색인화 작업에 매달려 있어서, 다른 부분에 대해서는 그렇게 자신있게 이야기 할 수 없다. 다만 여기서는 그 가능성을 제시하는 것이다.

가. 어휘 자료의 data base를 이용한 정리

어휘 자료의 정리는 dbaseII를 이용하여도 충분하다. 우리말의 경우 10자를 넘어가는 단어는 없기 때문이다. 여기서는 간단히 천자문류나 훈몽자회 등의 한자 주석형 어휘 자료의 정리에 이용할 수 있는 data file의 구조를 제시하여 두기로 한다.

한자 주석 어휘 자료 data file의 구조

FIELD-NAME	WIDTH	FIELD-NAME	WIDTH
TITLE	30	HANJA	20
YEAR	20	HUN	20
AUTHOR	24	EUM	20
PAGE	4	REFER	200
TOTAL			338

TITLE : 책명. 원명을 그대로 사용할 수도 있으나, 혼동되지 않는 범위 안에서 약칭을 쓰는 것이 좋다.

YEAR : 간행년도. 미상인 경우 추정년대를 쓸 수 있도록 10자를 부여한다.

AUTHOR : 편찬자, 간행기관.

PAGE : 어휘 자료의 출전 페이지.

HANJA : 대상 한자.

EUM : 한자음. 둘 이상인 경우 해당 훈을 기록할 수 있도록 10자를 부여한다.

HUN : 한자의 훈. 둘 이상인 경우 해당 음을 기록해 둔다.

REFER : 기타 필요하다고 생각되는 사항을 기록하는데 사용.

이상의 data base file을 이용하면 동일 한자에 대한 음·훈을 시대별, 문헌별로 검색할 수 있으며, 동일한 훈을 가지는 한자 또는 동일한 음을 가지는 한자 등 어휘론적인 문제를 모두 검색할 수 있을 것이다. 다만 dbaseII의 경우 index가 두 field에만 허용되므로 자료의 입력이 끝나면

HANJA field를 대상으로 sorting한 후 EUM과 HUN field를 대상으로 index file을 만들어 사용해야 한다는 불편이 따른다(음과 훈이 두 개 이상인 경우는 각기 별개의 record에 기록한다). 이 경우에도 앞에서처럼 프로그래밍을 해 둔다면 더욱 효율적인 작업이 가능할 것이다.

나. 문장 자료의 database를 이용한 정리(KWOC 작성을 중심으로)

원전자료 중 문장 자료를 정리한 것은 그 목적에 따라 두 유형의 작업이 가능한 것으로 보인다. 하나는 원전 자료의 전체 내용을 대상으로 하는 분석 정리이고, 다른 하나는 원전 자료의 내용 중 일부를 후에 참조할 필요가 있다고 생각되어 정리해 두는 경우이다. 우선 후자의 경우를 먼저 설명해 두기로 하자.

이 경우는 앞에서 예시한 두 경우를 조합한 방법을 사용하면 간단할 것이다. 즉 논저목록 및 내용요약 data file의 구조를 원용하되, Contents field의 경우 dbaseII에는 메모 필드가 없으므로 문자 필드를 이용하고, 기록해야 할 내용이 많은 경우를 대비해서 Contents 필드를 몇 개 만들어 이어서 사용하는 것이다. 다음에 data base file의 구조를 예시하여 그 가능성을 보이기로 한다.²¹⁾

국어학 자료에 있어서 원전 자료 전체를 대상으로 하는 분석·정리 작업이란 기본적으로 원전 자료를 대상으로 하는 색인화 작업을 뜻한다. 언해 자료라면 어간형·어미·조사 그리고 어간·어미의 통합형 등 각 유형의 목록과 출전을 쉽게 확인할 수 있도록 하는 것이고, 차차 자료라면 구결, 이두, 향가 등에서 차차된 한자의 목록과 출현 환경(출전), 용법 등을 찾아볼 수 있도록 하여야 할 것이며, 나아가서 번역원전 혹은 해독 결

21) 지면관계 상 자세한 설명은 생략한다. 그러나 대개의 이용 방법은 앞에서 설명한 것과 같으므로 충분히 이해할 수 있을 줄로 믿는다. 다만 이 dbase file은 기본적으로 TOPIC필드와 KEY필드를 중심으로 운용한다는 점, 즉 TOPIC과 KEY를 대상으로 index file을 만들고, 그를 이용해서 일련의 작업이 이루어지도록 하는 것임만 지적해 둔다. 따라서 필자나 책의 제목 등에 의한 검색이 필요하다면 똑같은 file을 다른 이름으로 copy하여 사용하도록 하여야 할 것이다.

원전 내용 참조용 data file의 구조

FIELD—NAME	WIDTH	FIELD—NAME	WIDTH
TITLE	30	KEY	30
AUTHOR	20	CONTENTS1	200
PAGE	4	CONTENTS2	200
TOPIC	20	CONTENTS3	200
TOTAL			704

과와의 대비·참조가 가능하도록 하여야 할 것이다. 이러한 작업의 중요성은 누구나가 인정하는 것이지만, 그 방대한 양의 문헌자료를 수작업을 통해서 색인을 만들기란 엄청난 시간과 인력이 투입되어야 하는 것이어서 부분적인 색인화는 가능할지라도 전체를 대상으로 하는 색인 작성이란 업무를 내기 어려운 일이었다. 더구나, 어떤 어미나 조사의 기능과 의미를 정확히 파악하기 위해서는 그들이 사용되는 문맥까지도 고려하지 않을 수 없는 것이므로, 단순히 어휘나 조사·어미 하나하나를 카드화하는 작업은 수고에 비해서 얻어지는 것이 적은 비생산적인 일이고, 따라서 생각은 있으면서도 선뜻 나서기 어려운 것이 사실이다.

필자가 dbaseII의 기능에 대해 어느 정도 알게 되었을 때 착안한 것이 dbaseII가 가지고 있는 sort나 index기능을 잘 활용하면 문장 자료의 색인화가 가능하겠다는 것이었고, 나아가서 개개의 어휘들의 출현 환경까지도 함께 확인 할 수 있는 KWIC(KEYWORD IN CONTEXT) 혹은 KWOC(KEYWORD OUT OF CONTEXT)의 작성이 가능하리라는 것이었다.²²⁾

22) 이러한 고민은 일본인인 福井 玲이 1985년 여름 8bit PC를 이용해서 월인 천강지곡 상의 KWIC색인을 작성한 결과물을 보고부터인데, 사실상 필자가 PC에 관심을 갖게 된 것도 거기에 자극받은 바 크다. 福井의 방식은 Basic Language를 이용해서 한글 자소 하나에 영문자 하나를 대응시켜 입력하고, 그것을 다시 음절 단위로 복원하는 방식을 사용한 것이었는데, 한글을 그대로 입력할 때의 두 배이상의 시간이 소요될 뿐 아니라, KWIC program을 알지 못하는 필자로서는 이용할 수 없는 것이어서, 뒤에 이야기할 dbaseII를 활용하는 것보다 그 활용도에 있어서 낫다고 하기 어려운 것이 아닌가 한다. 福井의 KWIC 작성 방식에 대한 것은 福井玲(1985), 月印千江之曲 上 KWIC 索引, 私家版참조.

이러한 생각은 간경도감본 불경언해서들을 보면서 구체화 되었다고 할 수 있는데, 그 자료들은 형식상 일정한 체제를 가지고 있어서, 하나의 틀만 잘 만들면 간경도감본 불경언해서 전체를 대상으로한 DATABASE의 구축이 가능하리라는 확신을 갖게 되었다. 그리하여 1988년 6월경부터 간경도감본 불경언해서 중에서도 비교적 그 분량이 적은 [반야심경]을 data file화하는 작업에 착수했는데 data base file의 구조 즉 한 record에 있어서의 field의 숫자, field의 크기, 한 field안에 들어갈 내용 등을 결정하기 까지 여러차례의 시행 착오를 겪었고, 그것을 바탕으로 KWIC 작성용 BASE file을 만들고 결과물을 얻기까지 또 몇 차례의 시행착오를 겪었다. 그러나 결국은 그런대로 쓸만한 KWOC을 작성해 낼 수 있었다.²³⁾

이제 여기서 [반야심경]의 KWOC작성시에 사용한 database file (SIM DBF)의 구조를 제시하고 그것을 KWOC작성용 BASE file로 만든 과정, 그리고 KWOC의 작성과 인쇄에 이르기까지의 과정을 나누어 설명하기로 한다.²⁴⁾

i) KWOC 작성용 dbase file의 구조

간경도감본 불경언해서는 그 양식이 한 면당 9행에 행당 17자로 되어있는데, 언해 부분의 경우 한 행에 두 줄이 들어가 행당 글자 수는 34자가 된다. 필자는 기본적으로 한권의 책을 하나의 file로, 한 행을 한 record로, 그리고 하나하나의 어휘(어간·어미 통합형)를 하나의 field에 할당하였다. 그 결과 [반야심경] 언해 부분 database file의 구조는 다음과 같이 설정된다.

23) 입력된 [반야심경] 언해 부분의 일부와 KWOC중 어간 어미 통합형의 일부를 이 글의 뒷 부분에 부록으로 실는데, 아직 필자가 생각하는 [반야심경] KWOC의 전체가 완성된 것도 아니고, 또 지면이 충분치 않은 까닭에 전체는 실지 못하고 일부의 모습만 참고삼아 보이도록 하겠다.

24) 이 이하의 내용은 컴퓨터에 대해서 전혀 알지 못하는 경우 이해하기 어려울지도 모른다. 그러나 실상 보석글II와 dbaseII를 조금만 익히면 쉽게 이해할 수 있는 것이므로 그리 어려운 방법이 아니다. 여기서는 필자의 작업내용을 설명하기 위한 것이므로 그대로 논의를 진행하기로 한다.

[도표 3] 반야심경언해본 dbase file(sim, dbf)의 구조

FIELD—NAME	WIDTH	FIELD—NAME	WIDTH
RECORD	5	W5	24
TITLE	7	W6	24
PAGE	6	W7	24
W1	24	W9	24
W2	24	W10	24
W3	24	W11	24
W4	24	NULL	7
TOTAL			295

여기의 field명 중 TITLE, PAGE, LINE은 각각 해당 문헌의 약칭, 면수 그리고 그 record가 나타는 행 수를, W1에서 W11까지는 그 행에 나타나는 하나하나의 어휘를 입력한다. 처음에는 어휘의 숫자를 가늠하지 못하여 W16까지 설정했는데, 실제로 작업에 임해본 결과, 한 행 (즉 언해부분 두 줄)에 들어가 있는 어휘 수의 최대치는 13개였다. 그러나 그러한 경우는 극히 드물어서 후에 KWOCBASE file을 만드는 경우 공연해빈 field를 대상으로 하는 경우가 많아지고, 또 file의 크기만 커져서 11개로 설정하는 것이 적당하겠다는 결론을 얻었다.

여기서 중요한 것은 RECORD와 NULL이라는 field이다. RECORD이라는 field는 각 레코드의 일련번호를 입력하는데, 후에 KWOCBASE file을 형성할 때에 이 RECORD field의 번호가 중요한 역할을 하게 된다. NULL이라는 field에는 “_____”를 입력해 두는데, 후에 KWOC에서 Keyword의 출현 위치를 나타내는데 사용하게 된다(부록 참조).

ii) KWOCBASE file의 형성

일단 (1)에서의 DBF에 전체 자료의 입력이 완료되면, 그것을 바탕으로 하여 KWOC을 작성하기 위한 몇 가지 작업이 필요하다. 다소 번거롭기는 하지만 그리 많은 시간을 요하는 것은 아니고, dbaseII의 명령어 몇 가지와 보석글 II를 이용하면 간단히 처리된다.

우선 `sim.dbf`를 대상으로 몇 가지 조작이 필요하다. 우선 W7에서 W11까지의 어휘 field와 RECORD field만을 `dbaseII`의 `append` 명령을 이용하여 따로 떼어내어 `presim1.dbf`를 만든 다음, 그것을 다시 `dbaseII`의 `copy`명령을 이용해서 `presim1.txt sdf`로 만든다. 즉 `dbaseII`상태에서 만들어진 자료를 보석글 II에서 사용할 수 있도록 하는 것이다. 그리고 나서 `dbase` 상태에서 빠져나와 보석글을 기동 시킨 후 `get presim1.txt sdf`라는 명령으로 그 자료를 편집할 준비를 갖추고 `edit` 상태에서 '자료옮기기'라는 명령을 이용해서 RECORD field의 내용(번호)를 한행씩 위로 올린다. 즉 원래의 DBF에서 '00001'이던 것을 '00002'로, '00002'이던 것을 '00003'으로 바꾸는 것이다. 만약 data file의 크기가 커서 보석글 II에서의 editing 작업이 불가능하다면, `dbaseII`의 `browse` 명령을 사용해서 하나씩 고쳐가도록 한다. 이렇게 수정된 `presim1.txt sdf`를 [도표 4]와 같은 구조를 가지는 `presim2.dbf`에 `append` 명령을 이용하여 입력한다. 동일한 방법으로 `sim.dbf`의 W1에서 W5까지의 어휘 field와 RECORD field를 따로 떼어내서 `aftsim2.dbf`를 만든다. 다만 `aftsim2.dbf`의 생성시에는 RECORD field의 내용을 역으로 한 번호씩 작게한다. 즉 '00002'이던 것을 '00001'로, '00003'이던 것을 '00002'로 바꾸는 것이다. `aftsim2.dbf`는 field-name을 [도표 5]와 같이 한다. 이렇게 만들어진 `presim2.dbf`와 `aftsim2.dbf`를 `dbase`의 `join` 명령을 이용하여 결합하여 [도표 6]과 같은 구조를 가지는 `consim.dbf`를 만들고 이 `consim.dbf`와 원래의 `sim.fbf`를 다시 `join` 명령을 이용하여 결합해서 [도표 7]과 같은 구조를 가지는 `kwocbase.dbf`를 만드는 것이다.²⁵⁾

25) 이 `join` 명령은 두개의 data base file을 결합할 때 사용하는 것으로 두개의 data base file안에 같은 이름의 field가 있고 그 field안의 data가 같은 것이 있을 때에 그것을 기준으로 두 data base file을 결합한다. 처음 `sim.dbf`를 만들 때 RECORD라는 field를 두고, 보석글에 들어가서 그 RECORD라는 field의 내용을 바꾼것은 바로 이 `join`명령을 사용해서 원래의 DBF에서의 W1에서 W5까지의 어휘 항목의 문맥 파악에 필요한 어휘들과 W7에서 W11까지의 어휘 항목의 문맥 파악에 필요한 어휘들을 일일이 입력하지 않고 한번 입력된 자료를 위치를 바꾸어 그대로 사용하기 위한 것이다.

[도표 4] PRESIM2.DBF의 구조

FIELD-NAME	WIDTH	FIELD-NAME	WIDTH
RECORD	5	PRE3	24
RRE5	5	PRE3	24
RRE4	24	PRE2	24
TOTAL			125

[도표 5] AFTSIM2.DBF의 구조

FIELD-NAME	WIDTH	FIELD-NAME	WIDTH
RECORD	5	AFT3	24
AFT1	24	AFT4	24
AEF2	24	AFT5	24
TOTAL			125

[도표 6] CONSIM.DBF의 구조

FIELD-NAME	WIDTH	FIELD-NAME	WIDTH
RECORD	5	AFT1	24
PRE5	24	AFT2	24
PRE4	24	AFT3	24
PRE3	24	AFT4	24
PRE2	24	AFT5	24
PRE1	24		
TOTAL			245

일단 [도표 7]과 같은 구조를 가지는 kwocbase.dbf의 생성이 끝나면 KWOC 작성을 위한 작업의 절반은 끝난 셈이다. kwocbase.dbf에서 PRE5에서 PRE1까지의 다섯 필드는 처음 자료를 입력한 sim.dbf의 W7에서 W11에 해당하는 자료들로 바로 앞 RECORD의 마지막 다섯 단어가 그 다음 RECORD의 첫 단어 앞에 나타나게 된다. 그렇게 함으로써 각 RECORD의 첫 단어의 출현 환경을 확인할 수 있게 하는 것이다. 또한 AFT1에서 AFT5까지의 다섯 필드는 sim.dbf의 W1에서 W5까지의 자료들이며, 바로 다음 RECORD의 처음 다섯 단어가 앞 RECORD의 마지막 필드의 다음에 나타나게 된다.²⁶⁾

iii) KWOC 색인의 작성

(2)에서 만들어진 kwocbase.dbf를 이용하여 KWOC 색인을 만드는 데에도 몇 가지 사전 절차가 필요하다. 그것은 각 어휘항목 (W1~W11 field)을 keyword의 위치에 가져다 놓는 작업과, 각각의 어휘항목을 하나의 file 안에 묶음으로써 keyword에 대한 index의 작성이 가능하도록 하는 작업이다.

각 어휘 항목을 keyword의 위치에 가져다 놓기 위해서는 [도표 8]과 같은 구조를 가지는 database file이 있어야 한다.

[도표 8]의 exw1.dbf는 W1 즉 각 record의 첫째 어휘를 keyword의 위치를 가져다 놓는 data base file이다. 구조를 보면 알겠지만 해당 어휘의 앞에 나오는 단어 다섯과 뒤에 나오는 단어 다섯을 함께 볼 수 있도록 하고, keyword의 출현 위치를 NULL이라는 field를 이용해서 나타내도록 하였다. 처음 sim.dbf를 만들면서 NULL이라는 field를 설정한 까닭이

26) 물론 이 경우 CONSIM.DBF를 생성할 때 다섯 단어 이상 11단어까지 끌어다 사용할 수도 있다. 그러나 11단어 모두를 끌어다 사용하는 경우 최종적으로 생성하는 KWOCBASE.DBF의 한 record의 크기가 817byte가 되고, 그를 바탕으로 만들어 낸 KWOC file의 RECORD의 크기 역시 거의 600byte에 달하게 되어 132 column printer를 사용한다 해도 최소한 2,3행을 차지하게 되므로 그리 좋은 방법이 아니다. 여기서 제시한 다섯 단어의 추출은 각 어휘의 환경이 한 행에 모두 나타나게 하기 위한 것이다.

[도표 7] KWOCBASE.DBF의 구조

FIELD-NAME	WIDTH	FIELD-NAME	WIDTH
RECORD	5	W5	24
TITLE	7	W6	24
PAGE	6	W7	24
LINE	5	W8	24
PRE5	24	W9	24
PRE4	24	W10	24
PRE3	24	W11	24
PRE2	24	AFT1	24
PRE1	24	AFT2	24
W1	24	AFT3	24
W2	24	AFT4	24
W3	24	AFT5	24
W4	24	NULL	7
TOTAL			534

[도표 8] KEYWORD의 추출을 위한 dbase file(EXW1.DBF)의 구조

FIELD-NAME	WIDTH	FIELD-NAME	WIDTH
TITLE	7	PRE1	24
PAGE	6	NULL	7
LINE	5	W2	24
W1	24	W3	24
PRE5	24	W4	24
PRE4	24	W5	24
PRE3	24	W6	24
PRE2	24		
TOTAL			289

이와 같은 방식으로 exw2.dbf부터 exw11.dbf까지 모두 11개의 여기에 있다.

database file에 append명령을 이용하여 kwocbase.dbf의 자료를 끌어다 넣은 후 이들을 하나로 합하고, 합해진 file의 keyword field를 대상으로 index file을 만들면 작업이 끝나는 것이다. 이러한 작업은 다만 시간이 소요될 뿐 크게 복잡한 것이 아니어서, 일련의 명령을 하나의 프로그램으로 작성해 놓고 그것을 이용하면 일일이 명령을 주지 않아도 되므로 아주 간단히 처리된다. [부록 2]의 make-exw.prg와 make-sdf.prg가 바로 그러한 과정을 수행하는 것이다.²⁷⁾

[도표 9] SIM-KWOC.DBF의 구조

FIELD-NAME	WIDTH	FIELD-NAME	WIDTH
TITLE	7	CON5	24
PAGE	6	POINT	7
LINE	5	CON6	24
KEYWORD	24	CON7	24
CON1	24	CON8	24
CON2	24	CON9	24
CON3	24	CON10	24
CON4	24		
TOTAL			289

27) 그런데 이 11개의 databasefile을 합하기 전에 각각의 dbase file에서 W2, W3등을 대상으로 sorting을 한 후에 그 field가 비어 있는 record는 삭제해 버리는 작업을 먼저 해 두는 것이 좋다. W1의 경우에만 비어있는 record가 있을 수 없으나 W2이하의 경우 비어 있을 수도 있고, 실상 W11의 경우 전체의 10분의 1 정도만이 자료가 들어 있을 뿐 나머지는 비어있는 record인 것이다. 한 record 즉 원전의 한 행 안에 11개의 어휘가 들어 있는 경우가 드물기 때문이다. MAKE-EXE.PRG와 MAKE-SDF.PRG를 하나로 묶지 않은 것은 바로 그러한 작업을 두 프로그램 사이에 할 수 있도록 하기 위해서이다.

그런데 make-sdf.prg를 실행하기 위해서는 다음과 같은 구조를 가지는 sim-kwoc.dbf가 있어야 한다. 이 dbase file이 바로 [반야심경]이라는 문헌자료의 KWOC색인이 되는 것이다.

make-sdf.prg의 실행으로 sim-kwoc.dbf에 자료 입력이 끝나면, keyword field를 대상으로 sim-key.ndx라는 index file을 만들고, sim-kwoc.dbf와 sim-key.ndx를 같이 사용하면 자신이 확인하고 싶은 어형의 확인은 아주 짧은 시간 안에 이루어진다. 그런데 일반적으로 어떤 문헌자료의 색인은 항상 참조할 수 있는 상태여야 하므로 일단 KWOC 색인의 작성이 끝나면 그것을 인쇄해서 참조하는 것이 편리하다. [부록 2]의 kwocprin.prg는 필자 나름대로 이 sim-kwoc.dbf의 내용을 참조하기 편히하게 인쇄하도록 만든 프로그램이다.

이상 조금 장황해 졌지만 필자가 16bit PC와 LQ1500프린터 그리고 dbaseII를 이용해서 [반야심경]의 kwoc색인을 만든 과정을 설명하였다.²⁸⁾ 여러므로 궁리한 결과이기도 한 해도, 아직 보완되어야 할 점이 많을 것이다. 그런 의미에서 부록에 필자가 작업에 사용한 일련의 프로그램을 실는다. 필자보다 dbase에 대해서 혹은 컴퓨터 프로그래밍에 대해서 더욱 많은 것을 알고 있는 분들이 필자의 방법보다 더 좋은 방법을 개발해내서 PC를 활용한 국어학 연구가 더욱 활성화되기 바란다.

VI. 마무리

필자가 처음 이 글을 쓰게된 동기는 머릿말에서 밝혔듯이 앞으로 PC를 사용할 예정이거나, PC를 구입하고도 그 기능을 충분히 살리지 못하는 분들에게 가능하다면 시간과 금전의 허비를 줄일 수 있는 길을 안내하면서, 아울러 지금까지 필자가 PC를 이용해 작업해 온 과정의 일단을 소개

28) 필자가 사용한 방법에 의해 KWOC색인을 작성하기 위해서는 반드시 20M 이상의 하드 디스크가 내장된 시스템을 사용하여야 한다. KWOCBASE, DBF나 SIM-KWOC, DBF의 경우 그 양이 1M가 넘기 때문이다. 필자가 II장에서 기억용량이 중요하다고 강조한 이유의 하나가 여기에 있다.

함으로써, PC사용자 사이의 정보 교환의 실마리를 잡아보고자 하는 것이었다. 또한 지금까지 많은 이들이 PC를 이용한 국어학 연구의 가능성을 모색해 왔고, 실제로 여러가지 작업을 한 것으로 아는데 어찌된 까닭인지 국어학 연구에 PC를 활용한 구체적 방법을 보인 글들은 찾아보기 어렵다는 사실에 대한 반성의 의미도 내포되어 있다고 할 것이다.

그러나 이 글이 그러한 목적에 충분한 내용이라고는 생각하지 않는다. 다만 V장의 내용에서 일부분이나마 살펴보았듯이 국어학 연구에서의 PC의 활용은 기대 이상의 좋은 결과를 얻을 수도 있는 것이라는 사실의 일부만이라도 확인 하는데 도움이 되었다면 다행이라고 하겠다.

이제 국어학 연구에서의 PC활용의 활성화를 위해 몇 가지 필자의 생각을 제안하면서 이 글을 마무리 짓고자 한다.

첫째, 현재 PC를 사용하고 있는 국어학 연구자 사이의 정보 교환의 활성화이다. 필자의 경험에 비추어, 어떤 문제에 봉착했을 때 혼자서 그 문제를 해결하는데 드는 시간이 열이라면, 이미 그러한 문제를 경험한 사람으로 부터 해결 방법을 배울 수 있다면 그에 들어가는 시간은 하나도 되지 않는다. 따라서 다양한 정보의 교환을 통해서 보다 효율적인 활용이 가능하리라는 것은 자명한 사실이다.

이를 위해서는 국어학 연구자로서 PC를 사용하고 있는 이들 사이의 BBS(bulletin board system)의 구축이 바람직하지만, 사설 통신망을 구축하는 것이 어렵다면 현재 DACOM에서 개설하고 있는 '한글사서함'에 각자가 가입하는 것도 괜찮다. 이렇게 함으로써 서로 직접 만나지 않더라도 컴퓨터를 이용해서 여러가지 정보를 교환할 수 있을 것이다.

둘째, 공통적인 관심사를 갖고 있는 이들끼리 서로 공동작업을 하고 그 결과를 교환하는 일이다. 한 예를들어 현재 필자가 관심을 두고 있는 문제는 중세국어 문헌자료의 database 구축인데, 워낙 자료의 양이 많다 보니 입력에 소요되는 시간이 여간 많지 않다. 이러한 어려움을 극복하기 위해서는 여러사람이 각각 몇 개의 문헌을 나누어 입력하고, 그 자료를 서로 교환함으로써 입력에 소요되는 시간을 대폭 줄일 수 있도록 하면 그만큼 시간이 절약될 것이다. 실상 국어사의 연구에 바쳐지는 시간이란 문

현과의 씨름이 그 대부분을 차지하는 것인 바, 일단 문헌자료에 대한 database가 구축되면 그러한 시간은 거의 소요되지 않을 것이고, 그만큼 국어사 연구의 발전이 약속되는 일인 것이다.

이러한 공동 작업과 자료 교환이 가능하려면, 무엇보다도 자료의 호환성이 보장되어야 한다. 이때의 자료 호환성이란 단순히 기종간의 그것을 뜻하는 것이 아니라, 자료의 정리에 사용하는 software나 정리 방법도 동일해야 한다는 것을 의미한다. 그렇게 할때에야 서로 교환한 자료를 각자 새로운 조정을 거치지 않고도 그대로 사용할 수 있을 것이기 때문이다. 필자가 KWOC색인의 작성 방법을 설명하면서 번거로움을 무릅쓰고 각 dbase file의 구조를 나열한 것도 이 점을 고려한 것이다. 만에 하나, 이 글을 읽고 필자와 같은 방법으로 문헌의 색인을 작성하고자 하는 이가 있다면 가능한한 필자와 같은 구조로 정리하여 서로 자료를 교환해 그대로 쓸 수 있도록 할 것을 염두에 두었던 것이다.

세째, 국어학 연구에 활용할 수 있는 software의 공동 개발이다. 한 예로 이 글에서 필자가 소개한 KWOC색인의 작성은, dbaseIII+를 이용한 경우 훨씬 더 간단히 이루어질 수 있는 것이고, 또한 그 운용도 효율적인 것이 된다. '반야심경 KWOC 색인'의 경우는 record의 수가 4,000을 넘지 않아 간단히 처리되었지만, [능엄경]을 대상으로 동일한 작업을 한다면 record의 수가 60,000이 넘을 것이고, [법화경]의 경우라면 100,000 정도일 것으로 예상되는 바, dbaseII를 사용하는 경우 한 file에 허용되는 record의 수는 65,535개여서 하나의 file에 여러가지 문헌을 함께 묶어 사용할 수 없음은 물론, 동일한 문헌도 한 file안에 넣을 수 없는 경우가 생

29) 실제 이러한 프로그램이 개발되어 있다. III장에서 소개한 'K-한글·한자'라는 software가 그것이다. 이 프로그램은 dbaseIII+, Lotus, Word star 등 영문 software에서 한글과 한자를 마음대로 사용할 수 있게한 utility program인데, 프린터에 별도의 한글 module을 장착하지 않아도 그대로 인쇄할 수 있고, 9핀 프린터에서도 24핀 프린터와 같은 인쇄 결과를 얻을 수 있는 등 여러가지 장점이 있다. 다만, 사용자 정의 문자의 수가 612자에 불과해 현재의 상태로는 고어의 입력에는 많이 부족한 상태이다. 그러나 프로그램 개발업체에 문의한 결과 수요만 어느 정도에 이를 경우 그 영역을 늘릴 수 있다는 응답을 들었다.

기는 것이다. 그러나 이러한 어려움은 dbaseIII+를 사용하면 전혀 문제가 되지 않는다. dbaseIII+에서는 한 file당 10억개의 record를 쓸 수 있어서 거의 file 크기에 따른 제약이 없기 때문이다. 그러나 앞에서 밝힌 바와 같이, dbaseIII+에서는 ‘한자 MASTER’를 쓸 수 없기에 어쩔 수 없이 dbaseII를 사용하는 것이다. 이 경우 ‘한자 MASTER’와 같은 기능을 하면서 dbaseIII+에서 사용 가능한 프로그램만 개발된다면 문제가 간단히 해결된다.²⁹⁾ 그러나 이러한 프로그램의 개발은 전문적인 지식을 요하는 것이어서 필자와 같은 PC이용자 개인이 시도할 수 있는 것은 아니다. 결국 전문가에게 의뢰하여야 하는데, 그때 소요되는 경비 또한 개인에게는 너무 큰 금액이다. 이때 같은 software를 필요로 하는 사람이 100명만 되면 개발을 의뢰하기도 쉽고, 또 개인의 부담도 작아지는 바, 그만큼 컴퓨터의 활용도가 높아질 수 있을 것이다.

이상의 필자의 제안은 전혀 새로운 것이 아니다. 실제 전산국어학회에는 위의 제안 내용들을 어느 정도 실현해가고 있는 것으로 안다. 다만 여기서는 다시한번 보다 많은 사람이 이러한 문제에 관심을 가질 것을 기대하면서 다시한번 그 중요성을 강조하고자 한 것이다.

〈참 고 문 헌〉

- 김경태(1988), [초보자를 위한 dBASEIII PLUS], 서울:영진출판사.
- _____ (1988), [프로그래머를 위한 dBASEIII PLUS], 서울:영진출판사.
- 김기철(1983), "한문자료의 컴퓨터 처리 방법 연구", [중국학지] 제1집, 계명대학교
중국학연구소
- 김우영·김영미 편저(1988), [고급사용자를 위한 dBASEIII Plus], 서울:영진출판사.
- 박현철(1987), "허클레스용 기울여 풀어쓰기 한글", [마이크로 소프트웨어]1987. 3.
- 삼보컴퓨터(1988), [보석글 II 사용설명서], 서울:삼보컴퓨터사.
- _____, [Trigem MS-DOS 사용설명서], 서울:삼보컴퓨터사.
- _____, [LQ-1500 OPERATING MANUAL], 서울:삼보컴퓨터사.
- 이기성·탁연상 공저(1988), [실무자를 위한 dBASEIII PLUS프로그램 모음], 서울:
영진출판사.
- 이해용(1988), "프린터의 다운로드 기능을 이용한 만능보석글", [마이크로 소프트웨
어]1988. 9.
- 정인상(1985), "컴퓨터를 이용한 방언자료의 처리 방법", [방언] 8집, 한국정신문화
연구원 어문연구실.

[부 록 1]

I. 시작 메뉴 프로그램

```

SET PATH TO A :
SET HEADING OFF
SET SAFETY OFF
* PROGRAM.. : RFSTART.PRG
CLEAR ALL
SET TALK OFF
SET BELL OFF
SET DELIMITER OFF
STORE SPACE(80) TO CLEARLINE
DO WHILE. T.
    CLEAR
    @ 0,0 SAY "HAN YOUNG—GYUN 1988"
    @ 4,5 SAY "문서[저널과 도서]관리"
    @ 5,2 SAY "===== "
    @ 9,10 SAY "0. 끝"
    @ 10,10 SAY "1. 자료 추가하기"
    @ 11,10 SAY "2. 자료 수정하기"
    @ 12,10 SAY "3. 자료 삭제하기"
    @ 13,10 SAY "4. 자료 찾아보기"
    @ 14,10 SAY "5. 삭제 자료 빼고 새 화일 만들기"
STORE 6 TO selectnum
do while selectnum < 0. or. selectnum >5
    STORE " " TO select
    @ 20, 15 SAY "어느 것 [0—5]"
    @ 20, 35 GET select PICTURE "#"
    READ
STORE VAL(select) TO selectnum
ENDDO
DO CASE
    CASE selectnum =0
        CLEAR ALL
        SET DELIMITER ON
        SET BELL ON
        USE
        CANCEL
    CASE selectnum =1
        DO RF—ADD

```

```

        USE
CASE selectnum=2
        DO RF—EDIT
        USE
CASE selectnum=3
        DO RE—DEL
        USE
CASE selectnum=4
        *PRINT PUT
        CLOSE PROCEDURE
        CLOSE DATABASE
        CLOSE INDEX
CASE selectnum=5
        DO RF—PACK

ENDCASE
ENDDO T
CLEAR ALL
*EOF : RESTART.PRG
    
```

II. 자료 찾기 Menu 프로그램

```

*PROGRAM. : RE—LO—PRE
SET HEADING OFF
SET SAFETY OFF
SET TALK OFF
SET BELL OFF
SET DELIMITER OFF
STORE SPACE(80) TO clearline
DO WHILE, T,
        CLEAR
@ 4,30 SAY "자료 찾아보기"
@ 5,30 SAY "===== "
@ 9,15 SAY "0. 시작 메뉴로"
@10,15 SAY "1. 책 제목으로 찾기"
@11,15 SAY "2. 저널 저자로 찾기"
@12,15 SAY "3. 저널 제목으로 찾기"
@13,15 SAY "4. 저널 제목으로 찾기"
@14,15 SAY "5. 주제별로 찾기"
@15,15 SAY "6. Key 1으로 찾기"
@16,15 SAY "7. Key 2로 찾기"
STORE 8 TO selectnum
*
DO WHILE selectnum < 0, or, selectnum >7
    
```

```
STORE " " TO select
@20, 32 say "어느것 [0-7] : "
@20, 45 GHT select PICTURE "9"
READ
STORE VAL(select) TO selectnum
ENDDO
*
DO CASE
    CASE selectnum=0
        CLEAR ALL
        SET DELIMITER OFF
        SET BELL OFF
        SET TALE OFF
        USE
        RETURN
    CASE selectnum=1
        DO RF--NAMB
    CASE selectnum=2
        DO RFBLIST
    CASE selectnum=3
        DO RF--NAMJ
    CASE selectnum=4
        DO RFJLIST
    CASE selectnum=5
        DO RFTOPIC
    CASE selectnum=6
        DO RFKEY2
    CASE selectnum=7
        DO RFKEY2
ENDCASE
ENDDO T
CLEAR ALL
*EOF : RF--LOCA, PRG
```

III. 논문 제목 이용해서 찾기 프로그램

```
*program.. : RFJLIST, PRG
SET HEADING OFF
SET SAFETY OFF
CLEAR
SET TALK OFF
SET DELIMITER OFF
SET BELL ON
```

```

SET MARGIN TO 1
STORE 1 TO pagenum, datanum
STORE "REFERENCES" TO pagehdg
STORE " " TO TEST
ACCEOT "찾으려는 논문 제목을 넣으세요 : "TO TEST
    STORE TRIM(TEST) TO TEST
    USE JOURNAL INDEX FOURNALI
    STORE " "TO PRTEST
    CLEAR
@ 2,30 SAY pagehdg
@ 3, 0 SAY "제목===== "
@ 3,11 SAY TEST
@ 4, 0 SAY "===== "
@ 4,40 SAY "===== "
    LOCATE FOR UPPER(TEST) $ UPPER(TITLE)
IF EOF()
    ? CHR(7) + "해당 자료가 없습니다"
ELSE
    STORE " " TO select
    @6,0 SAY "화면으로 보일까요, 인쇄할까요? [S/P]" ;
        GET select PICTURE "/"
    READ
    IF SELECT="P"
        6,0 SAY "프린터의 종류는 무엇입니까? [1=EPSON, 0=QUNIX]" ; GET
        PRTEST PICTURE "#"
    READ
ENDIF
DO CASE
    CASE select="S"
        CLEAR
        STORE 22 TO pagelen
    CASE select="P"
        CLEAR
        STORE 56 TO pagelen
        SET CONSOLE OFF
        SET PRINT ON
        IF PRTEST="1"
            ? CHR(27) + "h"
            TAB=CHR(27) + "T"
        ELSE
            ? CHR(27) + "h" + CHR(1)
            ? CHR(27) + "g" + CHR(1)
        ENDIF
    ENDIF

```

```

        CLEAR
    ENDCASE
ENDIF
IF SELECT='P'. AND. PRTEST='1'
    EPSON=.T.
ELSE
    EPSON=.F.
ENDIF
DO WHILE .NOT. EOF( )
    IF datanum=1
        ? "PAGE NO.", STR(pagenum, 3)
        ? SPACE(30)+pagehdg
        ? "Content=", TEST
        ? "===== "
        ? ? "===== "
    ENDIF
    POS=70+AT(' ', SUBSTR(TITLE, 71, 80))
    IF EPSON
        TRIM(AUTHOR)+TAB+'068'+YEAR
    ELSE
        ? AUTHOR, " ", TEAR
    ENDIF
    ? TRIM(SUBSTR(TITLE, 1, POS))
    ? TRIM(SUBSTR(TITLE, POS, 151-POS))
    IF EPSON
        ? TRIM(JOURNAL)+TAB+'062'+VOLUME+TAB+'068'+PAGE
    ELSE
        ? JOURNAL+" "+VOLUME+" "+PAGE
    ENDIF
    ? "-----"
    ? + "-----" +
    datanum=datanum+1
    IF datanum > 1. AND. SELECT="P"
        EJECT
        pagenum=pagenum+1
        datanum=1
    ENDIF
    CONTINUE
ENDDO
IF datanum > 1. AND. SESECT="P"
    EJECT
ENDIF
RELEASE ALL

```



```

SET PRINT OFF
SET CONSOLE ON
STORE " " TO AD
?
?
?
?

```

@ 22, 23 SAT “다 끝났으면 Enter를 누르세요” ;
GET AD

READ

RETURN

*EOF : RFFLIST

[부 록 2]

I . MAKE—EXW.PRG

[이 프로그램은 dBASE로 입력한 중세어 문헌자료에서 KWOCSORT에 의해 각 문헌별 KWOC(KEYWORD OUT OF CONTEXT)를 작성하기 위해 각 단어를 Keyword의 위치에 옮겨 놓고 그를 대상으로 SORT를 실행하여 KWOC에서의 불필요한 공백을 삭제하는 과정을 거칠 수 있는 자료를 만드는 과정이다].

```

USE EXW1
APPE FROM KWOCBASE
SORT ON W1 TO EXW1—S
USE
USE EXW2
APPE FROM KWOCBASE
SORT ON W2 TO EXW2—S
USE EXW3
APPE FROM KWOCBASE
SORT ON W3 TO EXW3—S
USE EXW4
APPE FROM KWOCBASE
SORT ON W4 TO EXW4—S
USE EXW5
APPE FROM KWOCBASE
SORT ON W5 TO EXW5—S
USE EXW6
APPE FROM KWOCBASE
SORT ON W6 TO EXW6—S
USE EXW7
APPE FROM KWOCBASE

```

SORT ON W7 TO EXW7—S
USE EXW8
APPE FROM KWOCBASE
SORT ON W8 TO EXW8—S
USE EXW9
APPE FROM KWOCBASE
SORT ON W9 TO EXW9—S
USE EXW10
APPE FROM KWOCBASE
SORT ON W10 TO EXW10—S
USE EXW11
APPE FROM KWOCBASE
SORT ON W11 TO EXW11—S
USE EXW1—S
COPY TO EXW1—S. TXT SDF
USE EXW2—S
COPY TO EXW2—S. TXT SDF
USE EXW3—S
COPY TO EXW3—S. TXT SDF
USE EXW4—S
COPY TO EXW4—S. TXT SDF
USE EXW5—S
COPY TO EXW5—S. TXT SDF
USE EXW6—S
COPY TO EXW6—S. TXT SDF
USE EXW7—S
COPY TO EXW7—S. TXT SDF
USE EXW8—S
COPY TO EXW8—S. TXT SDF
USE EXW9—S
COPY TO EXW9—S. TXT SDF
USE EXW10—S
COPY TO EXW10—S. TXT SDF
USE EXW11—S
COPY TO EXW11—S. TXT SDF
USE SIM—KWOC
APPE FROM EXW1—S. TXT SDF
APPE FROM EXW2—S. TXT SDF
APPE FROM EXW3—S. TXT SDF
APPE FROM EXW4—S. TXT SDF
APPE FROM EXW5—S. TXT SDF
APPE FROM EXW6—S. TXT SDF
APPE FROM EXW7—S. TXT SDF

```
APPE FROM EXW8—S, TXT SDF
APPE FROM EXW9—S, TXT SDF
APPE FROM EXW10—S, TXT SDF
APPE FROM EXW11—S, TXT SDF
USE
*EOF
```

III. KWOCPRIN, PRG

[이 프로그램은 만들어진 KWOC.DBF를 일정한 format로 인쇄하도록 하는 프로그램이다.]

```
SET HEADING OFF
SET TALK OFF
ERASE
SET BELL ON
SET MARGIN TO 1
STORE 1 TO pagenum, datanum
STORE 57 TO pagelen
STORE 136 TO pagewid
ERASE
USE SIM—KWOC INDEX SIM—KEY
STORE 0 TO PAGECTR
STORE 90 TO STORE “<般若波羅蜜多心經 KWOC 索引>” To pagehdg LINECTR
SET PRINT ON “<般若波羅蜜多心經 KWOC 索引>” To pagehdg LINECTR
SET CONSOLE OFF
? CHR(27)+CHR(104)
? CHR(27)+“G”
DO WHILE .NOT. EOF( )
    IF datanum=1
        ? ”
        ?
        ? “PAGE NO.”, STR(pagenum, 3)
        ? “-----”
        ? ? “-----”
        ? “page line keyword”
        ? “-----”
        ? ? “-----”
        ?
    ENDIF
    STORE PAGE TO tpa
    STORE LINE TO tli
    STORE KEYWORD TO tw1
    STORE CON1 TO tw2
```

```

STORE CON2 TO tw3
STORE CON3 TO tw4
STORE CON4 TO tw5
STORE CON5 TO tw6
STORE POINT TO tw7
STORE CON6 TO tw8
STORE CON7 TO tw9
STORE CON8 TO tw10
STORE CON9 TO tw11
STORE CON10 TO tw12
STORE CON11 TO tw13
? "&tpa &tli"
? ? TRIM("&tw1")
? ? TRIM("&tw2")
? ? TRIM("&tw3")
? ? TRIM("&tw4")
? ? TRIM("&tw5")
? ? TRIM("&tw6")
? ? TRIM("&tw7")
? ? TRIM("&tw8")
? ? TRIM("&tw9")
? ? TRIM("&tw10")
? ? TRIM("&tw11")
? ? TRIM("&tw12")
? ? TRIM("&tw13")
    STORE 7 to linectr
    STORE datanum+1 TO datanum
    IF datanum > 50
    EJECT
    STORE pagenum+1 TO pagenum
    STORE 1 TO datanum
ENDIF
CONTINUE
ENDDO
IF datanum > 1
    EJECT
ENDIF
RELEASE ALL
SET PRINT OFF
SET CONSOLE ON
* EOF

```