

004. 2
76743

연구자료

09-04-012

한글 자동 인덱스 작성에 관한 연구

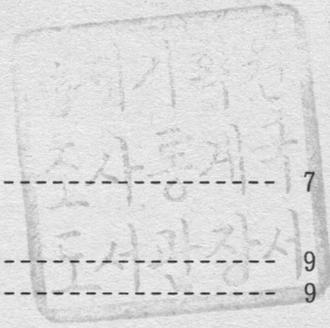
1990년12월

경제기획원 조사통계국
자료관리과

039321

목 차

1. 서론 -----	7
1.1 연구의 목적 -----	9
1.2 연구의 필요성 -----	9
1.3 이론적 배경 -----	9
1.3.1 색인이란? -----	10
1.3.2 반자동 색인 기법 -----	11
1.3.3 자동 색인 기법 -----	11
1.3.4 형태소 -----	11
1.3.4.1 어근 -----	12
1.3.4.2 접사 -----	12
1.3.4.3 조사 및 어미 -----	12
1.3.5 색인어 비교 방법 -----	13
1.3.5.1 어휘 사전 이용 방법 -----	13
1.3.5.2 언어학적인 분석 방법 -----	14
1.4 개발 방법 -----	15
1.5 기대 효과 -----	16
2. 업무 분석 -----	17
2.1 배경도 -----	19
2.2 LEVEL 0 DFD(시스템 개요도) -----	19
2.3 LEVEL 1 DFD(색인어 자동 출 시스템 개요도) -----	20
2.4 분석 명세서 -----	20
2.4.1 어절 분리 -----	20
2.4.2 불용어 삭제 -----	21
2.4.3 조사 삭제 -----	21
2.4.4 색인어 FILE 작성 -----	22
2.5 자료 사전 -----	22
2.6 시스템 FLOW -----	23



3. 업무 설계	25
3.1 프로그램 구조도	27
3.2 프로그램 명세서	29
3.2.1 MAIN PROGRAM	29
3.2.2 통계 D.B 색인 작성	29
3.2.3 도서자료 색인 작성	30
3.2.4 화면 입력 색인 작성	30
3.2.5 어절 분리	31
3.2.6 불용어 삭제	32
3.2.7 조사 추출	33
3.2.8 조사 삭제	34
3.2.9 색인 FILE 작성	34
3.2.10 VSAM FILE 생성 및 삭제	35
3.2.11 색인어 작성 확인	35
3.3 자료 사전	36
3.3.1 프로그램 변수 내용	36
3.3.2 불용어 TABLE	36
3.3.3 조사 TABLE	37
3.3.4 색인어 확인 FILE 작성	37
3.3.5 색인어 FILE	38
3.3.6 도서 FILE	38
3.3.7 통계 D.B 구조	39
3.3.7.1 D.B 구조도	39
3.3.7.2 통계 D.B SEGMENT 내용	40
4. 구현 및 평가	43
4.1 구현	45
4.1.1 구현 환경	45
4.1.2 구현 특성	45
4.1.3 구현 과정	45
4.1.4 프로그램 및 FILE	46
4.1.5 CICS SYSTEM TABLE	47
4.2 평가	48

5. 결론	53
6. 참고 사항	61
○ 한글 불용어 TABLE	64
○ 영문 불용어 TABLE	65
○ 조사 TABLE	66
○ IBM 한글 코드 특성	71
○ ON_LINE SYSTEM PROGRAM	73

일 리 두 기

이 보고서는 현재 중요하게 대두되고 있는 한글문헌자료의 색인화 필요에 따라, 한글 문법의 조사 생성 및 불용어의 원리를 컴퓨터에 적용하여 정확하고 신속한 색인어 추출 시스템을 제시 하므로써 각종 한글 자료의 색인어 작업에 효과적으로 이용토록 하였습니다.

이용중 문의 사항 및 의견이 있으시면 경제기획원 자료관리과 (이대형 처리관)에 연락 바랍니다.

전화 직) 720 - 2788
720 - 2789

교) 735 - 5371
(295)

1990년 12월

자료관리과장 최 든 철

1. 서 론

1. 서론

1.1 연구의 목적

통계 정보의 보급 및 이용 증대를 위한 색인어 검색 시스템 개발을 위하여, 검색에 필요한 색인어 작성시 수작업을 배제한 자동 생성 시스템을 개발하여 색인어 작성 시간 및 인원의 노력을 절감 시키고, 색인어 검색 시스템 개발의 환경을 제공코자 함.

1.2 연구의 필요성

당국에서는 일찌기 통계 자료의 보급에 힘써 각종 통계 자료 D.B를 구축하고 MENU 검색, 통계표 검색, CODE 검색등의 통계 D.B 검색 시스템을 개발, 운영 하고 있으나, 통계 자료의 이용 확산을 위하여는 통계 자료의 색인틀 이용한 색인어 검색 시스템의 개발이 요구되고 있음.

그러나 색인어 검색 시스템을 개발하기 위하여는 자료의 색인 작업이 필수적으로 요구되며, 이러한 색인 작업을 수작업으로 하기에는 많은 노력과 시간이 들어 이를 해결 하기 위하여 컴퓨터를 이용한 한글 자동 색인 시스템 개발이 필요함.

1.3 이론적 배경

컴퓨터의 발달과 함께 대용량 자료의 저장과 처리 시간이 짧아 짐에 따라, 각종 자료가 데이터 베이스화 되고, 이용자가 증대 되고있다. 특히 각종 기술 및 문헌 자료의 이용이 확산 되면서 이를 원활하게 이용 할수 있는 검색 시스템이 요구 되면서, 자료의 색인 작업이 중요한 문제로 등장하게 되었다. 초기에는 일일이 수작업으로 색인어를 작성 하였으나, 자료량이 방대해 짐에따라 수작업이 불가능 하여, 컴퓨터를 이용한 자동 색인 기법을 연구 하게 되었다.

자동 색인 기법에는 반자동 색인 기법(Semiautomatic Method), 통계적 기법(Frequency Criteria), 구문론적 기법(Syntex Method), 의미론적 기법(Semantically Oriented), 통계적 기법과 구문론적 기법의 혼합 기법 등이 연구 되고 있다.

그러나 이러한 방법은 대부분 영문을 대상으로한 연구이다. 이에 따라서 국내에서도 한글 색인에 대한 관심이 고조 대면서 어휘사전(LEXICON)을 구축하고, 이를 이용한 색인어 작성 방법이 제시되어 왔으나, 어휘 사전의 구축이 어렵고 하급 특성상 필요 어휘를 분리한다는 것이 힘들기 때문에 어휘 사전을 이용하지 않는 색인 방법이 연구 되어 왔다.

여기서는 불용어 TABLE을 이용하여 불용어를 삭제하고, 한글의 조사와 어미의 생성 규칙에서 조사의 구성 규칙을 분석하여, 조사를 분리하는 구문론적 색인어 작성 방법을 선택 하였다.

1.3.1 색인이란?

색인(INDEX)은 INDEX의 어원인 라틴어 'INDICARE', 즉 '지적하다', '가르치다'에서 나온 것으로 정보의 위치를 지시하여 필요 정보를 쉽게 찾기위한 도구이다.

우리 사전에서는 책의 내용이 되는 사항이나 자구를 일정한 순서로 배열하여, 쉽게 찾을수 있도록 꾸며놓은 목록 이라고 정의 하고 있다.

색인의 기능에는 지시 기능과 선별 기능이 있는데, 선별 기능을 효율적으로 이용 한것이 자동 색인 기법이다.

자동 색인의 기본 원리는 단어들을 주제어와 비주제어, 의미어와 무의미어로 구분하고 주제어와 의미어로 평가된 단어를 색인어로 선정하는 것이다.

일반적인 수작업 색인 작업은 먼저 문장의 구문(Syntex)과 의미를 파악한후 그 내용과 관계있는 지식과 이론을 집합하여 대표 단어를 추출한다. 추출된 단어의 불용어 유무를 판정한후 어휘집과 단어의 빈도수를 비교하고 최종적으로 동의어를 통일된 대표어로 대치하여 선정하는 작업이다.

1.3.2 반자동 색인 기법

색인 대상 입력 자료를 어떤 표준 자동 색인 방법으로 어휘를 발췌한후 색인 전문가가 인용 문헌이나 색인 어휘집을 이용하여 색인어를 선정하는 방법이다.

이러한 방법은 결국 색인 전문가가 최종 수작업으로 색인을 작성 하여야 하며 연구 결과 자동 색인 방법과 비교해서 더 나을것이 없다고 판정됐다.

1.3.3 자동 색인 기법

통계적 기법과 비 통계적 기법이 있으며, 통계적 기법은 단어의 빈도를 계산하여 기준치에 합당한 단어를 색인어로 선정하는 방법으로써, 단어의 발생 빈도를 계산하는 '단어 빈도론', 단어가 관련된 문헌의 빈도수를 계산하는 '문헌 빈도론' 그리고 단어의 빈도를 문헌의 빈도로 나눈 역 문헌 빈도론 등이 있으며, 최근에는 사전이나 색인어휘집을 컴퓨터내에 구축하여 색인의 효율성을 더욱 극대화 시키고 있다.

1.3.4 형태소

일정한 음성에 일정한 의미로 결합하여 된 언어 단위 가운데 의미를 가진 최소 단위를 형태소라 하며, 곧 개개의 의미를 갖는 최소의 요소라 할 수 있다.

형태소는 다음과 같은 관점에서 분류 된다.

1.3.4.1 어근

" 가축들이 화단을 짓밟아 놓았다. "

이 문의 통상적 성분은 " 가축들이 / 화단을 / 짓밟아 / 놓았다. " 와 같이 네개의 최소 자립 형태로 구성 되어 있으나, 여기서 /가축/, /화단/, /밟-/ 과 같은 형태소는 다른 형태소에 비해서 그뜻이 구체적이고 가장 중심이 되는 관념을 표시하는 형태소이다.

1.3.4.2 접사

" 가축-들-이 화단-을 짓-밟-아 놓-았-다" 에서 " -들, -이, -을, 짓-, -아, -았, -다" 등과 같은 형태소는 일정한 음과 일정한 의미 내용을 가지고 있는 점으로 보면 각각 하나의 언어 형식 이지만 이들 형태소는 단독으로 쓰일수 없고, 반드시 다른 언어, 형식에 의존해야 된다. 그중 /짓-/ 과 같은 접두사와 /-들/과 같은 접미사는 단어의 기본 개념을 어휘적으로 확대시키는 형태소이다.

1.3.4.3 조사 및 어미

" -이, -을, -아, -았, -다"와 같은 형태소는 어근에 붙어서 어근이 의미하는 어떠한 판단 형식을 결정하는 문법적 의미를 가지고 있는데, 이를 세분하여 보면, "-이, -을"과 같이 체언에 붙는 조사류 문법 요소와 "-아, -다"와 같이 용언에 붙는 어미형 문법요소로 분류되며, 자동 색인어를 비롯한 자연어의 처리를 위해서는 어절에서 접사 형태소와 조사 및 어미를 제거시킨 어근 형태소를, 통사 및 구문 분석을 위해서는 조사 및 어미 형태소를 이용하여야 한다.

한국어의 구조는 서구 언어와 달리 서술 자료에 의해 복잡하게 구성 되어 있어, 한 어간을 중심으로 파생되는 어휘가 무수히 많아 방대한 양의 체언과 용언을 모두 컴퓨터에 기억 시킬수 없으며, 어근으로 구성된 어휘

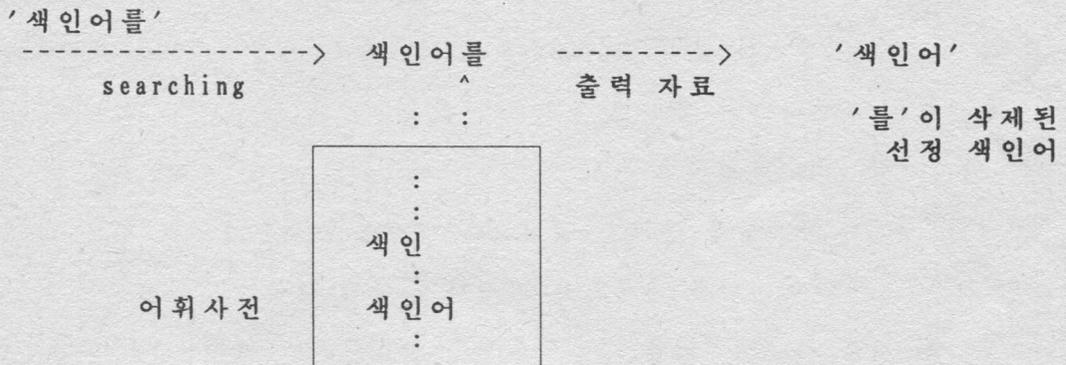
한국어의 구조는 서구 언어와 달리 서술 자료에 의해 복잡하게 구성 되어 있어, 한 어간을 중심으로 파생되는 어휘가 무수히 많아 방대한 양의 체언과 용언을 모두 컴퓨터에 기억 시킬수 없으며, 어근으로 구성된 어휘 사전의 구성에 어려움이 많으므로 색인어의 추출에서 제외되는 불용어에 대한 사전을 구성하고 체계적인 형태소 분석을 통해 이를 처리하여야 한다.

1.3.5 색인어 비교 방법

1.3.5.1 어휘 사전 이용 방법

1) 최장 일치법

분리된 어절을 어휘 사전과 비교하여 그중 가장 길게 일치되는 체언을 색인어로 선정.

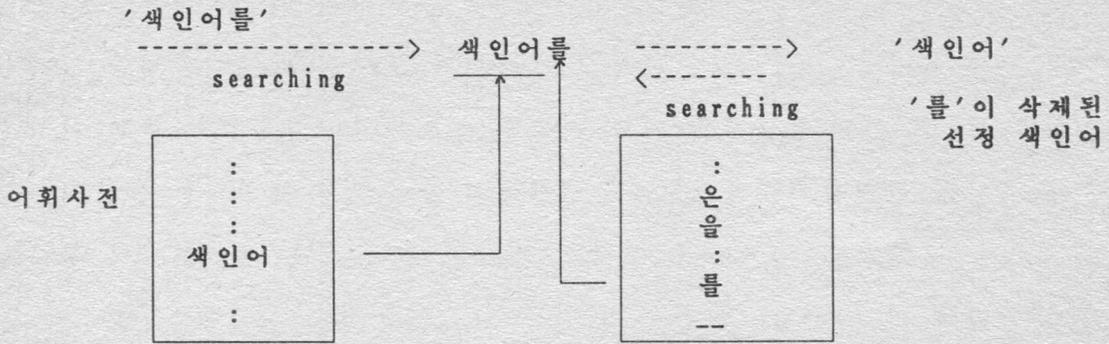


즉,

입력 자료 '색인어를'을 어휘사전에 수록된 어휘중 가장 길게 일치되는 '색인어'로 결정하는 방법.

2) Head_Tail법

분리된 어절을 어휘 사전과 비교하며, 어절뒤의 조사를 조사어미 사전을 이용 하여 비교하여 조사를 삭제하는 방법.



* 이러한 방법들은 어휘 사전의 작성을 전제로 하므로, 어휘 사전 작성에 많은 노력을 필요로 한다.

1.3.5.2 언어학적인 분석 방법

어휘 사전의 구축없이 언어학적인 분석을 통하여 불용어 LIST와 조사 형성 단계를 분석한 조사 LIST를 이용하여 불용어 및 조사를 비교하여 삭제하는 방법이다.

1) 어절 분리 단계

입력된 한글 자료를 공백을 중심으로 어절을 분리.

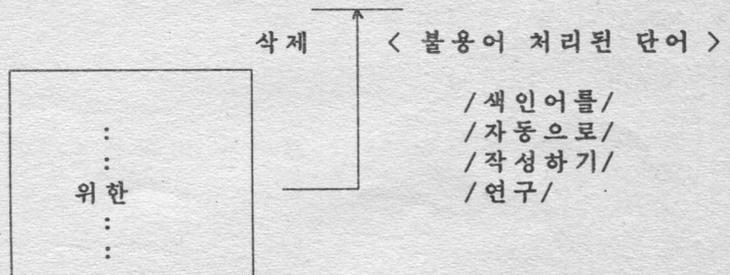
색인어를 자동으로 작성하기 위한 연구'

분리된 어절 => /색인어를/ /자동으로/ /작성하기/ /위한/ /연구/

2) 불용어 삭제 단계

불용어 LIST를 이용하여 분리된 어절중 불용어를 삭제.

/색인어를/ /자동으로/ /작성하기/ /위한/ /연구/



3) 조사 추출 단계

조사 형성 단계를 분석하여 작성한 조사 LIST를 이용하여 불용어 처리된 어절을 뒤에서 부터 조사 LIST와 대조하여 처리.

입력순서=> /색인어를/ ① /자동으로/ ② /작성하기/ ③ /연구/ ④ < 선정된 색인어 >

색인어를	:	/색인어/
<--	:	
자동으로	:	/자동/
<---	하기	
작성하기	:	/작성/
<---	으로	
연구	어를	/연구/
	:	

4) 불용어 처리

색인의 역할을 할수 없는 어휘에 대한 정보를 사전화 하여 초기 단계에서 이를 제외 시킴으로서 불필요한 시간을 제외 시킴.

예로써 한국 전자 통신 연구소 에서는 한글 206개, 영문 150개의 불용어를 선정하여 이용 하고 있다.

5) 조사 처리

자연어에서 색인어를 추출하기 위해서는 조사 및 어미의 처리가 필수적이다. 특히 한국어는 이들의 활용이 자유롭기 때문에 매우 까다로운 처리가 되어, 조사간의 연관 관계를 분석하여 조사의 활용 및 생성 규칙을 적용 이용하여야 함.

1.4 개발 방법

- ON_LINE 이용 형태로 개발.
- 언어학적인 분석 방법을 이용하여 개발.

- '한국 전자 통신 연구소' 선정, 한글 206개와 영문 110개의 불용어 사용.
- 43개 유형의 조사 TABLE 작성.
 - 조사의 마지막 음절 기준으로 그룹화.
 - 각 조사들의 상관관계를 규정하고 자모수에 따른 고유값 지정.
 - 가나다 순으로 정리.
 - 음운 법칙 특성 부여.
 - 받침없는 체언뒤 오는 조사 : X
 - 받침있는 체언뒤 오는 조사 : 0
 - 받침에 상관 없이 오는 조사 : 0X
- 통계 D.B 수록 자료명 적용
- 도서 정보 화일의 도서명 적용
- ON_LINE 입력 자료 색인화

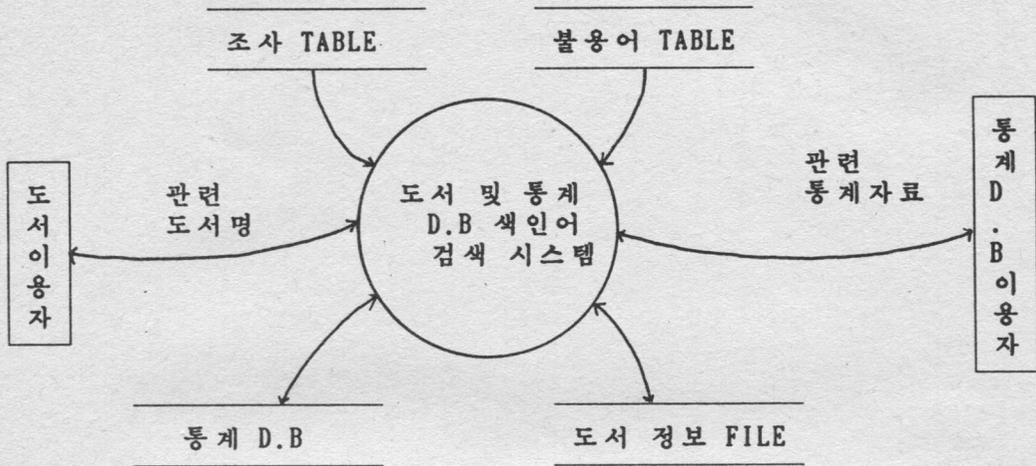
1.5 기대 효과

- 자동 색인화로 인력 및 시간 절감.
- 색인어 검색 시스템 개발 환경 제공.
- 색인 자동 작성의 이론적 정립.
- 전산 개발 환경의 상승.

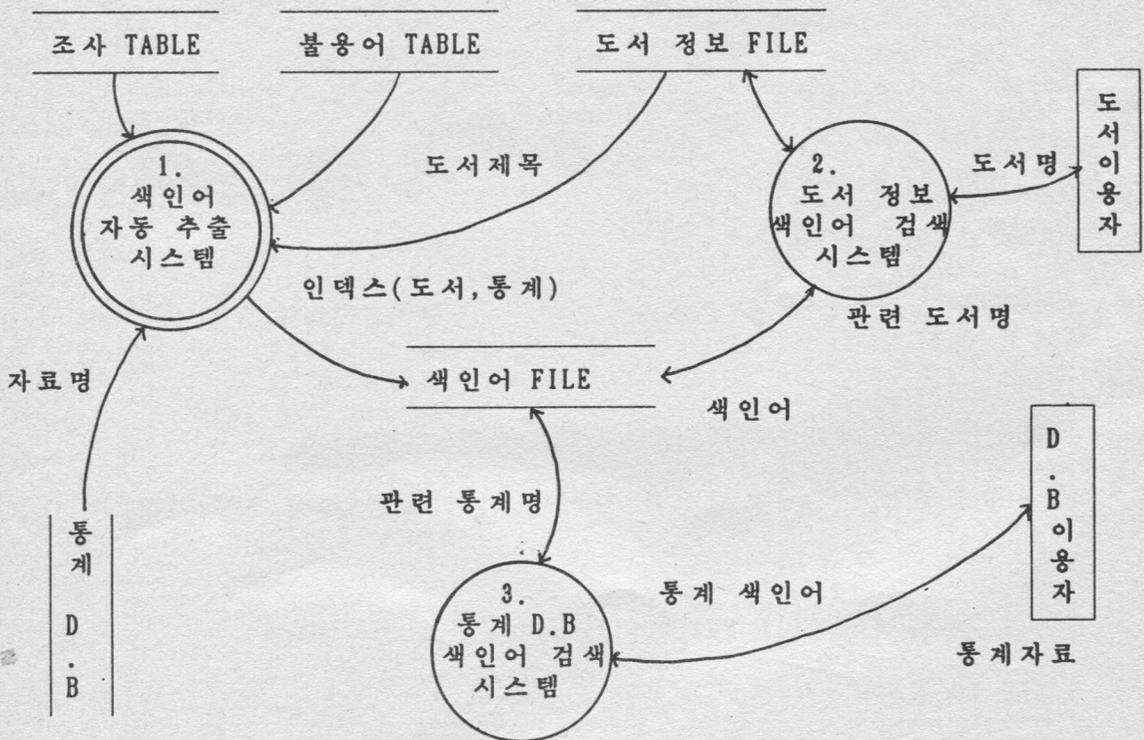
2. 업무 분석

2. 업무 분석

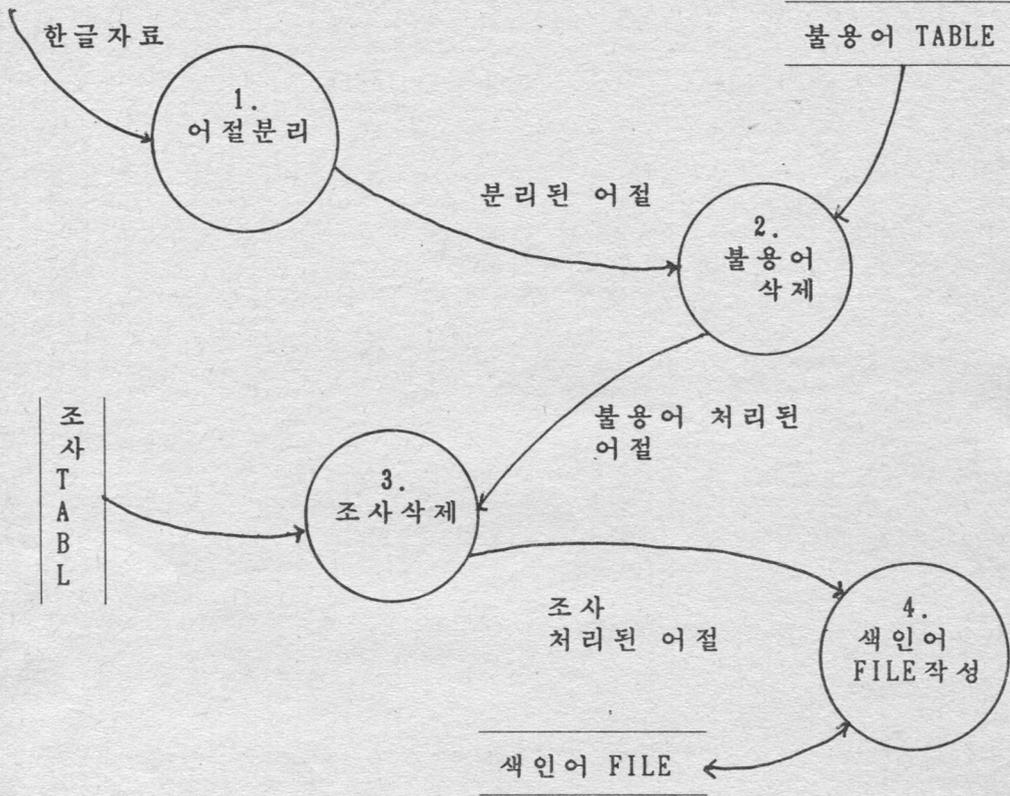
2.1 배경도(CONTEXT DIAGRAM)



2.2 LEVEL 0 DFD(시스템 개요도)



2.3 LEVEL 1 DFD(색인어 자동 추출 시스템 개요도)



2.4 분석 명세서

2.4.1 어절 분리

어절 분리
1. READ 한글 자료 INTO IN
2. INDEX (IN, ' ') * 단어와 단어 사이의 공백 검색 *
3. 공백전의 단어 추출
4. IF 추출=끝 THEN RETURN
5. GO TO 2.

2.4.2 불용어 삭제

불용어 삭제

1. INDEX = 1
2. IF 불용어(INDEX) BY 불용어.길이 = 현단어
THEN DO;
 현 단어 삭제
 RETURN
 END;
3. IF 불용어(INDEX) = 'END'
THEN RETURN
4. INDEX = INDEX + 1
5. GO TO 2.

2.4.3 조사 삭제

조사 삭제

1. 기본 조사어 검색 * SET 존재.INDEX *
 없으면 RETURN
2. 현.INDEX = 존재.INDEX
3. 다음.INDEX = 현.INDEX +1
4. IF 현.INDEX.길이 > 다음.INDEX.길이
 THEN DO:
 체언의 자모음 CHECK(b, 0, X, OX)
 DELETE 현 단어 조사 BY 존재.INDEX
 RETURN
 END;
5. 현.INDEX = 다음.INDEX
6. IF 현 단어 조사 = 현 조사어(현.INDEX)
 THEN 존재.INDEX = 현.INDEX
7. GO TO 3.

2.4.4 색인어 FILE 작성

색인어 FILE 작성

1. GET 색인.TABLE INTO IN
2. 색인.RECORD = IN
3. IF 색인.TABLE = 끝 THEN RETURN
4. WRITE 색인.FILE
5. GO TO 1.

2.5 자료 사전

한글자료 = {한글 문장} * 한글 문법에 의하여 작성된 문장으로서 색인어 검색 대상 자료 *

분리된 어절 = { 어절 } * 입력된 한글 자료에서 SPACE를 중심으로 추출한 어절 *

불용어 처리된 어절 = { 어절 } * 분리된 어절중 불용어를 삭제한 어절 *

조사 처리된 어절 = { 어절 } * 불용어 처리된 어절중 조사 유무를 검사하여 조사를 삭제한 어절 *

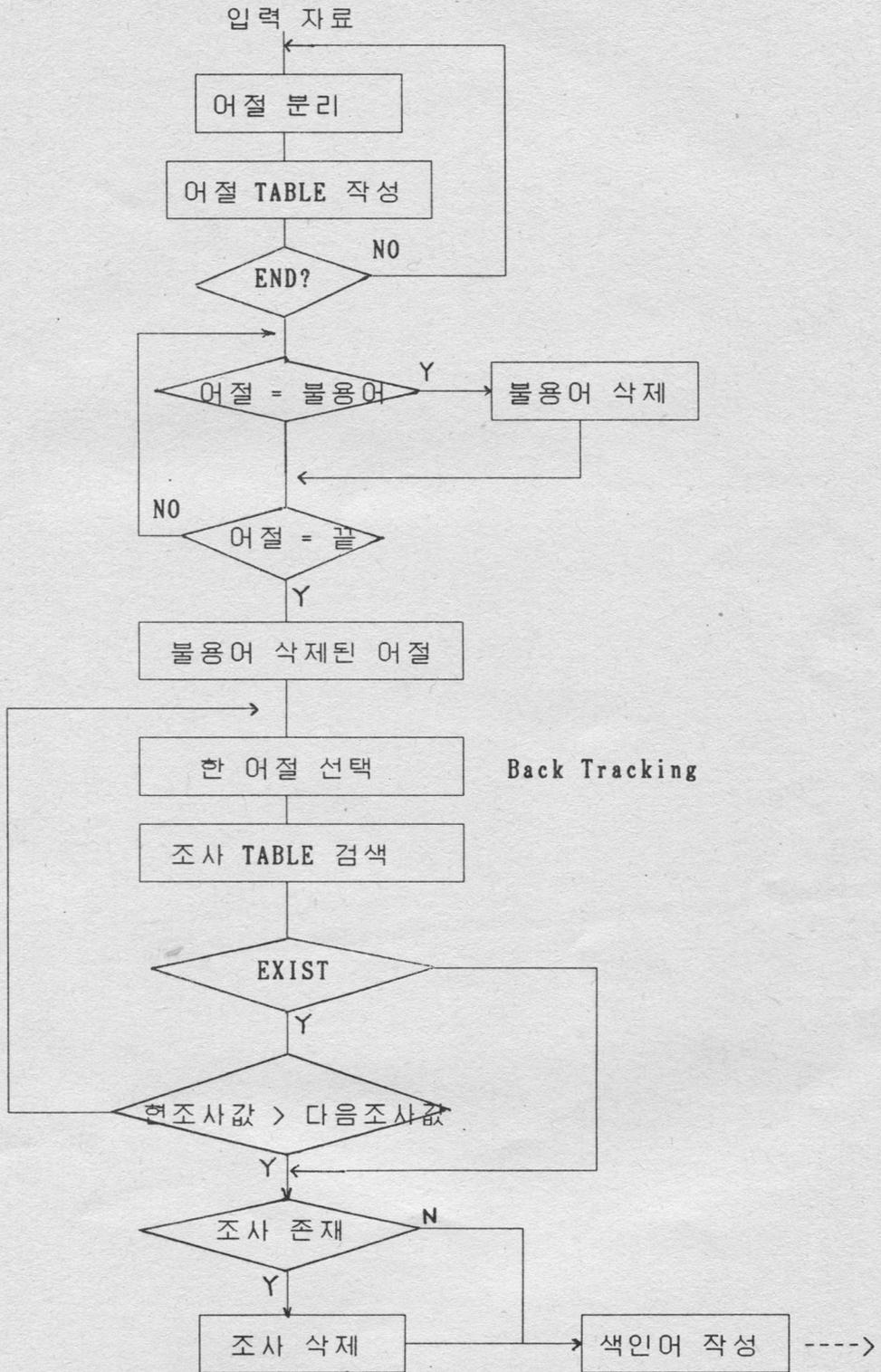
불용어 TABLE = { 불용어 + 특성 + 길이 }

조사 TABLE = { 조사 + 길이 + 특성 }

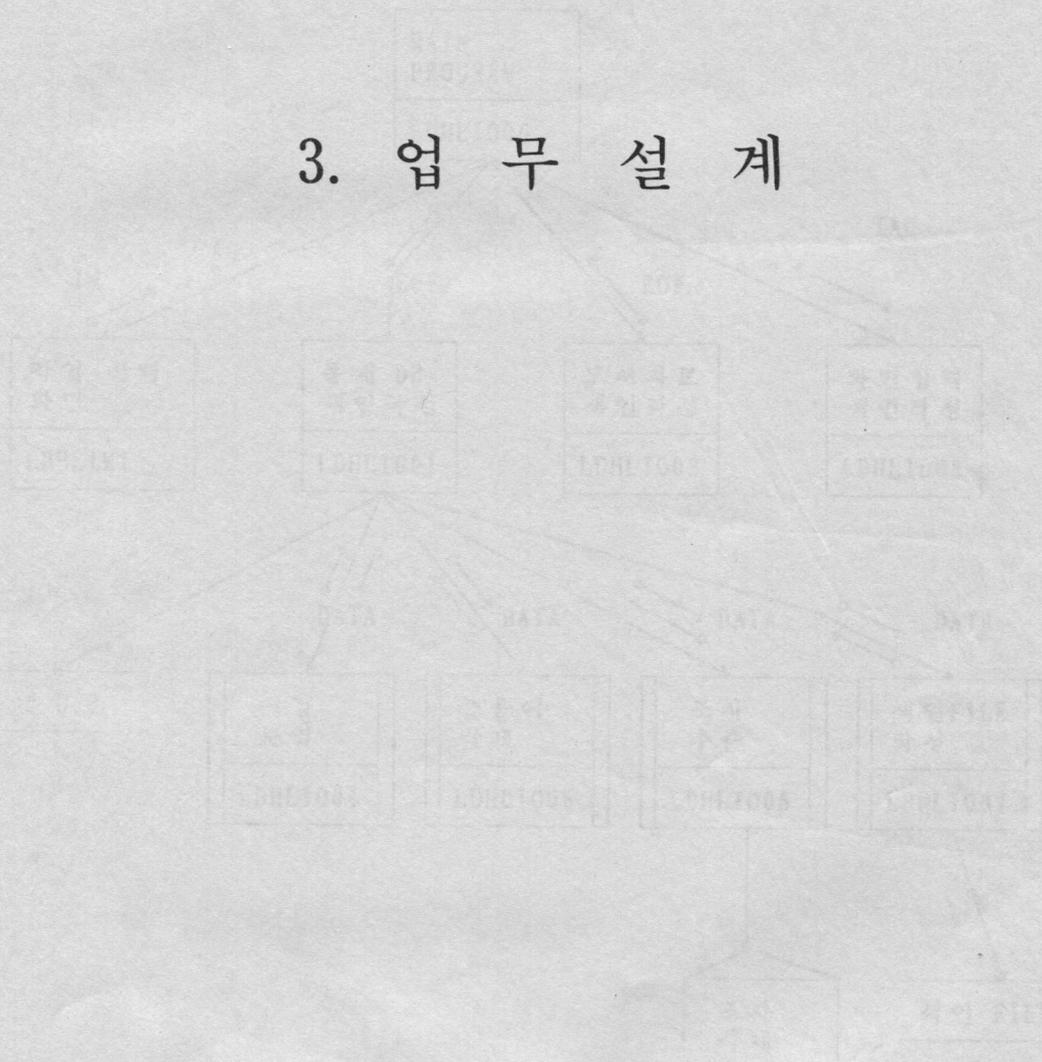
특성 = (0 | X | OX) * 0 : 받침 있는 체언 뒤에 오는 조사 *
* X : 받침 없는 체언 뒤에 오는 조사 *
* OX : 받침에 상관없이 오는 조사 *

색인어 FILE = { 색인어 + { 코드} }

2.6 시스템 FLOW



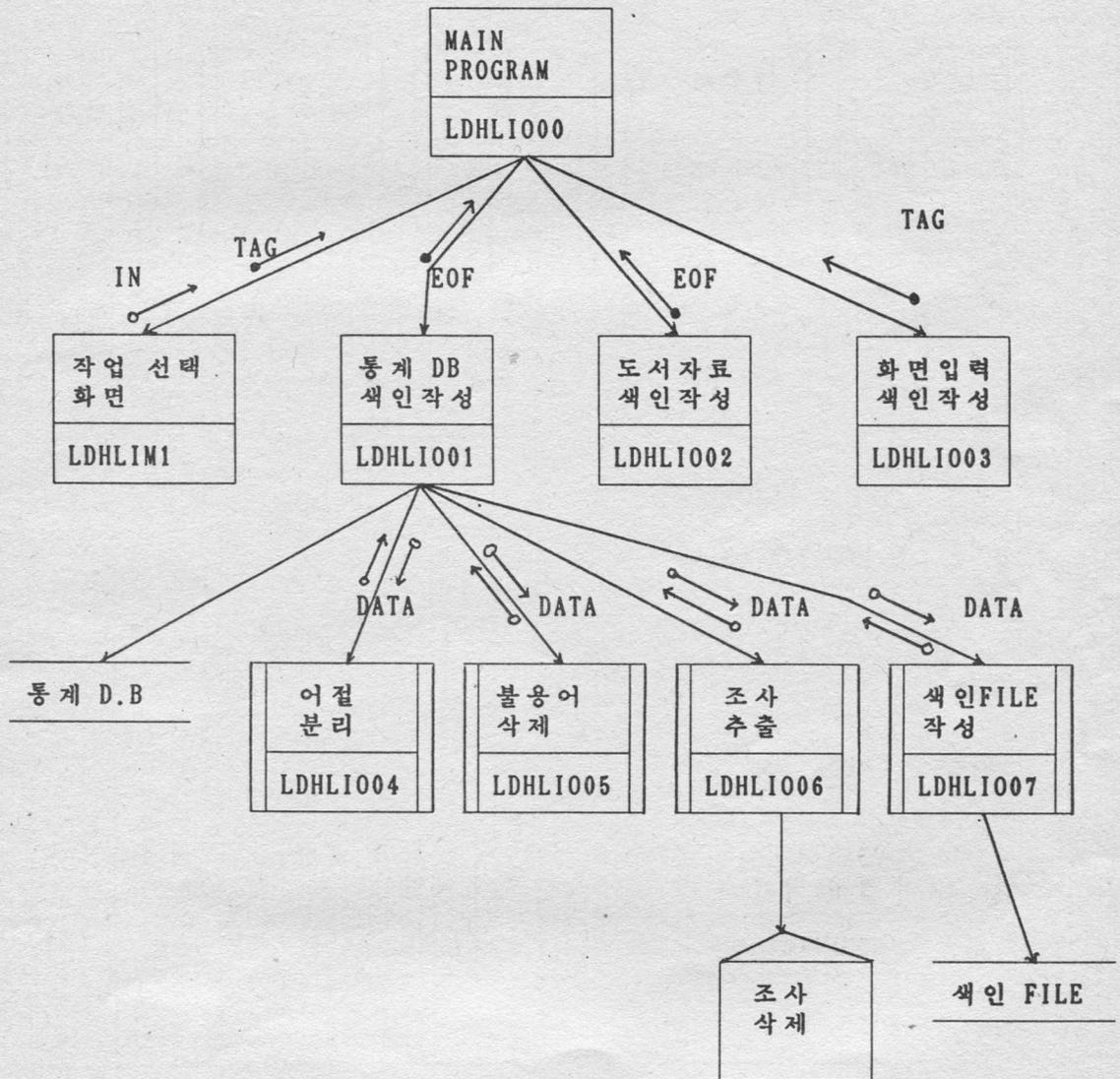
3. 업무 설계

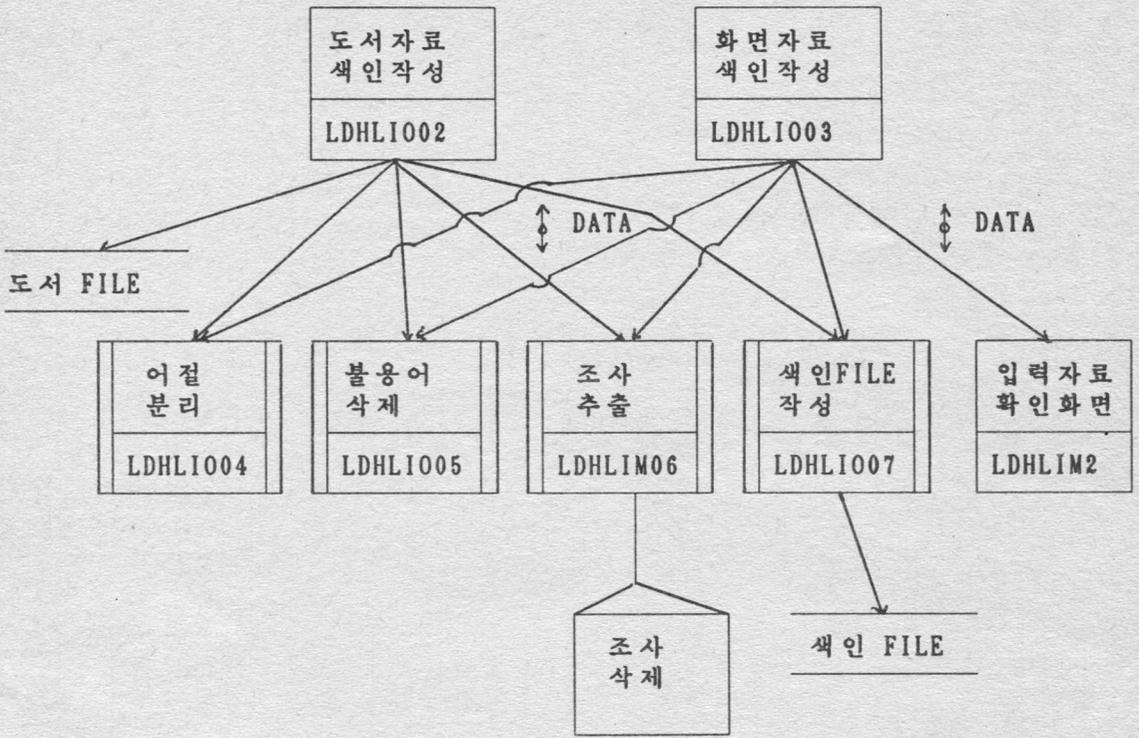


3. 업무 설계

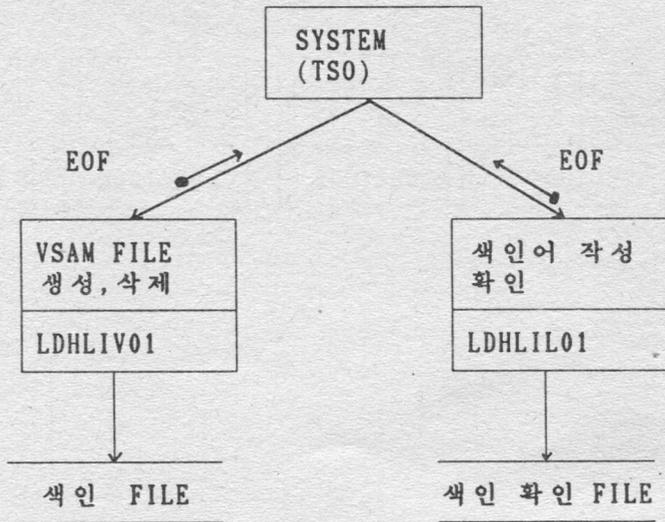
3.1 프로그램 구조도

0. ON_LINE PROGRAM CHART





0. BATCH PROGRAM CHART



3.2 프로그램 명세서

3.2.1 MAIN PROGRAM

프로그램명 : LDHLI000

프로그램 내용 : 인덱스 추출 시스템의 부 프로그램을 관리

PROCEDURE INDEX_MAIN_MODULE

DO WHILE(^ END);

SEND 작업 선택 화면
RECEIVE 작업 선택 화면 INTO INI,DBNAME;

IF INI = '1' THEN CALL LDHLI001(DBNAME); * 통계 DB 색인 작성
IF INI = '2' THEN CALL LDHLI002; * 도서자료 색인 작성
IF INI = '3' THEN CALL LDHLI003; * 화면입력 색인 작성

END INDEX_MAIN_MODULE;

3.2.2 통계 DB 색인 작성

프로그램명 : LDHLI001

프로그램 내용 : 통계 D.B의 자료를 읽어, 색인어를 작성

PROCEDURE STATIC_MODULE(DBNAME)

SCHEDULE D.B

READ D.B TO INPUT;

DO WHILE(^EOF);

CALL LDHLI004(INPUT,TOKEN_TAB,P,CNT); * 어절 분리 PROGRAM *
CALL LDHLI005(INPUT,TOKEN_TAB,P,CNT); * 불용어 삭제 PROGRAM *
CALL LDHLI006(INPUT,TOKEN_TAB,P,CNT); * 조사 분리 PROGRAM *
CALL LDHLI007(INPUT,TOKEN_TAB,P,CNT); * 인덱스 화일 작성 PROGRAM
READ D.B TO INPUT;

END SATATIC_MODULE;

3.2.3 도서 자료 색인 작성

프로그램명 : LDHLI002

프로그램 내용 : 도서 화일의 도서 제목 , 색인어를 작성

```
PROCEDURE BOOK_MODULE;  
  
  READ 도서_화일 INTO INPUT;  
  DO WHILE( ^EOF);  
    CALL LDHLI004(INPUT,TOKEN_TAB,P,CNT); * 어절 분리 PROGRAM *  
    CALL LDHLI005(INPUT,TOKEN_TAB,P,CNT); * 불용어 삭제 PROGRAM *  
    CALL LDHLI006(INPUT,TOKEN_TAB,P,CNT); * 조사 분리 PROGRAM *  
    CALL LDHLI007(INPUT,TOKEN_TAB,P,CNT); * 인덱스 화일 작성 PROGRAM  
    READ 도서_화일 INTO INPUT;  
  END;  
END BOOK_MODULE;
```

3.2.4 화면 입력 색인 작성

프로그램명 : LDHLI003

프로그램 내용 : 통계 D.B의 자료를 읽어, 색인어를 작성

```
PROCEDURE SCREEN_MODULE(DBNAME)  
  
  SEND 입력 화면;  
  RECEIVE 입력 화면 INTO INPUT;  
  DO WHILE( ^END);  
    CALL LDHLI004(INPUT,TOKEN_TAB,P,CNT); * 어절 분리 PROGRAM *  
    CALL LDHLI005(INPUT,TOKEN_TAB,P,CNT); * 불용어 삭제 PROGRAM *  
    CALL LDHLI006(INPUT,TOKEN_TAB,P,CNT); * 조사 분리 PROGRAM *  
    CALL LDHLI007(INPUT,TOKEN_TAB,P,CNT); * 인덱스 화일 작성 PROGRAM  
    M = 1;  
    DO I = 1 TO P;  
      J = INDEX(TOKEN_TAB(I),' ');  
      SUBSTR(OUTDATA,M,J-1) = SUBSTR(TOKEN_TAB(I),1,J-1);  
      M = M + (J-1) + 2;  
    END;  
    SEND 확인 화면 FROM OUTDATA;  
    SEND 입력 화면;  
    RECEIVE 입력 화면 INTO INPUT;  
  END;  
END SCREEN_MODULE;
```

3.2.5 어절 분리

프로그램명 : LDHLI004

프로그램 내용 : 입력자료를 한글 어절 TOKEN으로 분리

```
PROCEDURE TOKEN_CREATION_MODULE(INPUT, TOKEN_TAB, P, CNT);
```

```
ATT(9) = ( ) , . < > - _ /; * 삭제 문자 *
```

```
TOKEN_TAB(*) = ' ';
```

```
N = LENGTH(INPUT); * 입력 자료의 길이 계산 *
```

```
* 입력 자료중 삭제 문자를 검색하여 삭제하는 작업 *
```

```
DO I = 1 TO N;
```

```
DO J = 1 TO 9;
```

```
IF SUBSTR(INPUT, I, 1) = ATT(J);
```

```
THEN IF SUBSTR(INPUT, I-1, 1) = ' ';
```

```
THEN SUBSTR(INPUT, I-1, 2) = ' ';
```

```
END;
```

```
END;
```

```
L = 60;
```

```
I = 1;
```

```
P = 0;
```

```
* BLANK를 CHECK하여 입력 자료를 TOKEN(어절)으로 분리하여  
TOKEN_TAB에 저장 *
```

```
DO WHILE(INPUT(, I, 3) ^= ' ');
```

```
J = INDEX(SUBSTR(INPUT, I, L), ' ');
```

```
IF J = 1 THEN I = I + 1;
```

```
ELSE DO;
```

```
M = J - 1;
```

```
P = P + 1;
```

```
TOKEN_TAB(P) = SUBSTR(INPUT, I, M);
```

```
I = I + J;
```

```
L = L - J;
```

```
END;
```

```
END;
```

3.2.6 불용어 삭제

프로그램명 : UNUSED_DELETE_MODULE

프로그램 내용 : 불용어 유무를 확인하여 불용어 삭제

```
PROCEDURE UNUESD_DELETE_MODULE(INPUT, TOKEN_TAB, P, CNT);

DCL      UNUSED_TAB,      * 불용어 TABLE의 정의 *
         UNUSED_WORD,    * 불용어 *
         UNUSED_LEN;      * 해당 불용어의 길이 *

T = 0;

DO Q = 1 TO P;
  I = 1;

  * 분리된 어절이 불용어인지를 CHECK *

  DO WHILE(UNUSED_WORD(I) ^= '*****' &
           SUBSTR(UNUSED_WORD(I), 2, UNUSED_LEN) ^=
           SUBSTR(TOKEN_TAB(Q), 1, UNUSED_LEN(I)));

    I = I + 1;    * 불용어 TABLE의 위치 계산 *

  END;

  * 불용어인 경우 TOKEN_TAB에서 불용어를 삭제 *

  IF  UNUSED_WORD(I) = '*****'
    THEN DO:
      T = T + 1;
      TOKEN_TAB(Q) = INPUT;
      END;

END;

P=T;

END UNUSED_DELETE_MODULE;
```

3.2.7 조사 추출

프로그램명 : LDHLI006

프로그램 내용 : 조사삭제를 위하여 해당 조사를 검색

```
PROCEDURE SUFFIX_SEARCHING_MODULE(INPUT,TOKEN_TAB,P,CNT);

DCL SUFFIX_TAB,      * 조사 정의 *
    SUFFIX_LEN;     * 해당 조사의 길이 *

DO I = 1 TO P;
  S_INDEX = 1;
  L = INDEX(TOKEN_TAB(I),' ');
  Q = L - 2;

  * 어절을 뒤에서부터 조사 TABLE과 비교 기본 조사 위치 계산 *
  DO WHILE(SUFFIX_WORD(S_INDEX) ^= '*****' &
           SUBSTR(TOKEN_TAB(I),Q,2) ^= SUBSTR(SUFFIX_WORD(S_INDEX),
           (INDEX(SUFFIX_WORD(S_INDEX),' ') - 2),2));

    S_INDEX = S_INDEX + 1;      * 해당 조사 TABLE 계산 *
  END;

  P_INDEX = S_INDEX;

  * 기본 조사에서 파생된 조사가 있는지를 확인 *
  DO WHILE(SUFFIX_WORD(S_INDEX) ^= '*****' &
           SUFFIX_LEN(P_INDEX) =< SUFFIX_LEN(P_INDEX + 1));
    P_INDEX = P_INDEX + 1;
    IF SUBSTR(TOKEN_TAB(I),L - SUFFIX_LEN(P_INDEX),
              SUFFIX_LEN(P_INDEX) =
              SUBSTR(SUFFIX_WORD(P_INDEX),2,SUFFIX_LEN(P_INDEX))
    THEN S_INDEX = P_INDEX;
  END;

  * 입력 자료에 조사가 있으면 조사 삭제 프로그램으로 분기 *
  IF SUFFIX_WORD(S_INDEX) ^= '*****'
    THEN CALL SUFFIX_DELETE_MODULE
           ( SUBSTR(SUFFIX_WORD(S_INDEX),15,1),
             SUFFIX_LEN(S_INDEX),TOKEN_TAB(I),L);

END;
END SUFFIX_SEARCHING_MODULE;
```

3.2.8 조사 삭제

프로그램명 : SUFFIX_DELETE_MODULE

프로그램 내용 : 확인된 조사를 현어절에서 삭제

```
PROCEDURE SUFFIX_DELETE_MODULE(SUFFIX_ATT,SUFFIX_LEN,TOKEN,LENGTH);
```

```
  * 확인된 조사의 바로 앞글자의 종성을 종성없는 IBM 한글 코드  
  ('11100001'B) 와 OR 연산 *
```

```
  UNSPEC(COMP)=UNSPEC(SUBSTR(TOKEN,LENGTH - SUFFIX_LEN -1,1) |  
                    '11100001'B);
```

```
  SELECT(SUFFIX_ATT);
```

```
    * 앞글자에 종성이 없으면  
    WHEN('1')   IF UNSPEC(COMP) = '11100001'B  
                THEN SUBSTR(TOKEN,LENGTH - SUFFIX_LEN ) = ' ' ;  
                ELSE SUBSTR(TOKEN,LENGTH - SUFFIX_LEN+2) = ' ' ;  
    * 앞글자에 종성이 있으면  
    WHEN('2')   IF UNSPEC(COMP) ^= '11100001'B  
                THEN SUBSTR(TOKEN,LENGTH - SUFFIX_LEN ) = ' ' ;  
                ELSE SUBSTR(TOKEN,LENGTH - SUFFIX_LEN+2 ) = ' ' ;  
    WHEN('3')   SUBSTR(TOKEN,LENGTH - SUFFIX_LEN ) = ' ' ;  
    OTHERWISE;
```

```
  END;
```

```
END SUFFIX_DELETE_MODULE;
```

3.2.9 색인 FILE 작성

프로그램명 : LDHLI007

프로그램 내용 : 색인어 를 색인어 FILE로 구성

```
PROCEDURE INDEX_WRITE_MODULE(INPUT,TOKEN_TAB,P ,CNT);
```

```
  OUT_DATA = INPUT;  
  OUT_TAB = TOKEN_TAB;  
  OUT_P    = P;  
  OUT_CNT = CNT;  
  KEY = OUT_CNT||SUBSTR(OUT_DATA,1,57);
```

```
  WRITE INDEX FROM(OUT_REC) BY KEY;
```

```
  CNT = CNT + 1;
```

```
END;
```

3.2.10 VSAM FILE 생성 및 삭제

프로그램명 : LDHLIV01

프로그램 내용 : 색인어 FILE 초기 생성

PROCEDURE STEP1:

IDCAMS:

DELETE INDEX VSAM FILE;

CREATE INDEX VSAM FILE;

END STEP1;

PROCEDURE STEP2:

OPEN INDEX;

OUT_REC = ' ';

KEY = ' ';

WRITE INDEX FROM(OUT_REC) KEYFROM(KEY);

CLOSE INDEX;

END;

3.2.11 색인어 작성 확인

프로그램명 : LDHLIL01

프로그램 내용 : 색인어 작성 결과 확인을 위하여 일반 파일로 전환

PROCEDURE PRINT:

OPEN INDEX,PRINT;

READ INDEX INTO(IN_REC);

DO WHILE(^EOF);

M = 1;

DO I = 1 TO IN_P;

J = INDEX(IN_TAB(I),' ');

SUBSTR(OUT_LIST,M,J-1) = SUBSTR(IN_TAB(I),1,J-1);

M = M + (J-1) + 2;

END;

OUT_CNT = IN_CNT;

OUT_TAB = IN_TAB;

WRITE PRINT FROM OUT_REC;

READ INDEX INTO(IN_REC);

END;

END PRINT;

3.3 자료 사전

3.3.1 프로그램 변수 내용

변 수 명	특 성	내 용
IN	CHAR(1)	'1' : 통계 D.B 색인 작성 '2' : 도서 자료 색인 작성 '3' : 화면 입력 색인 작성
DBNAME	CHAR(8)	통계 D.B NAME(IN이 '1'인 경우)
TAG	-	FUNCTION KEY CLEAR 값
EOF	-	통계 D.B 및 각 FILE의 EOF 값
통계 D.B	-	통계 D.B 구조 참고
DATA		
.INPUT	CHAR(74)	대상 입력 자료
.TOKEN_TAB	CHAR(20)	작성된 색인어 기억 10 개 까지
.P	PIC'999'	작성된 색인어 갯수
.CNT	PIC'999'	입력 RECORD 갯수
도서 RECORD	-	도서 FILE 참고

3.3.2 불용어 TABLE : UNUSED_TAB(200)

변 수 명	특 성	내 용
UNUSED_WORD	CHAR(10)	불용어 주의) 1번 BYTE : 한글 시작 코드 X'0E' 불용어 끝: 영문 시작 코드 X'0F'
UNUSED_LEN	BIN(15)	해당 불용어의 길이

3.3.3 조사 TABLE : SUFFIX_TAB(300)

변 수 명	특 성	내 용
SUFFIX_WORD	CHAR(20)	
	.1 COL	X'0E' 한글 시작 특성 코드
	.2 -11 "	조사 FIELD : 한글 끝에 X'0F':영문코드
	.13-14 "	조사길이
	.15 "	조사 특성 1: 받침없는 낱말뒤에 오는 조사 2: 받침있는 낱말뒤에 오는 조사 3: 두경우 모두 사용되는 조사 ' ': 처리 안함
SUFFIX_LEN	BIN(31,0)	해당 불용어의 길이

3.3.4 색인어 확인 FILE : PRINT

위 치	변 수 명	특 성	내 용
1 - 3	OUT_CNT	PIC'999'	RECORD 일련 번호
4	OUT_HS1	CHAR(1)	한글 시작 특성 FIELD(X'0E')
5 - 78	OUT_LIST	CHAR(74)	색인대상 자료
79	OUT_ES1	CHAR(1)	영문 시작 특성 FIELD(X'0F')
80 - 89	OUT_FI	CHAR(2)	사용 안함
90	OUT_HS2	CHAR(1)	한글 시작 특성 FIELD(X'0E')
91 -164	OUT_TAB	CHAR(74)	작성된 색인 DATA
165	OUT_ES2	CHAR(1)	한글 시작 특성 FIELD(X'0F')

3.3.5 색인어 FILE : INDEX

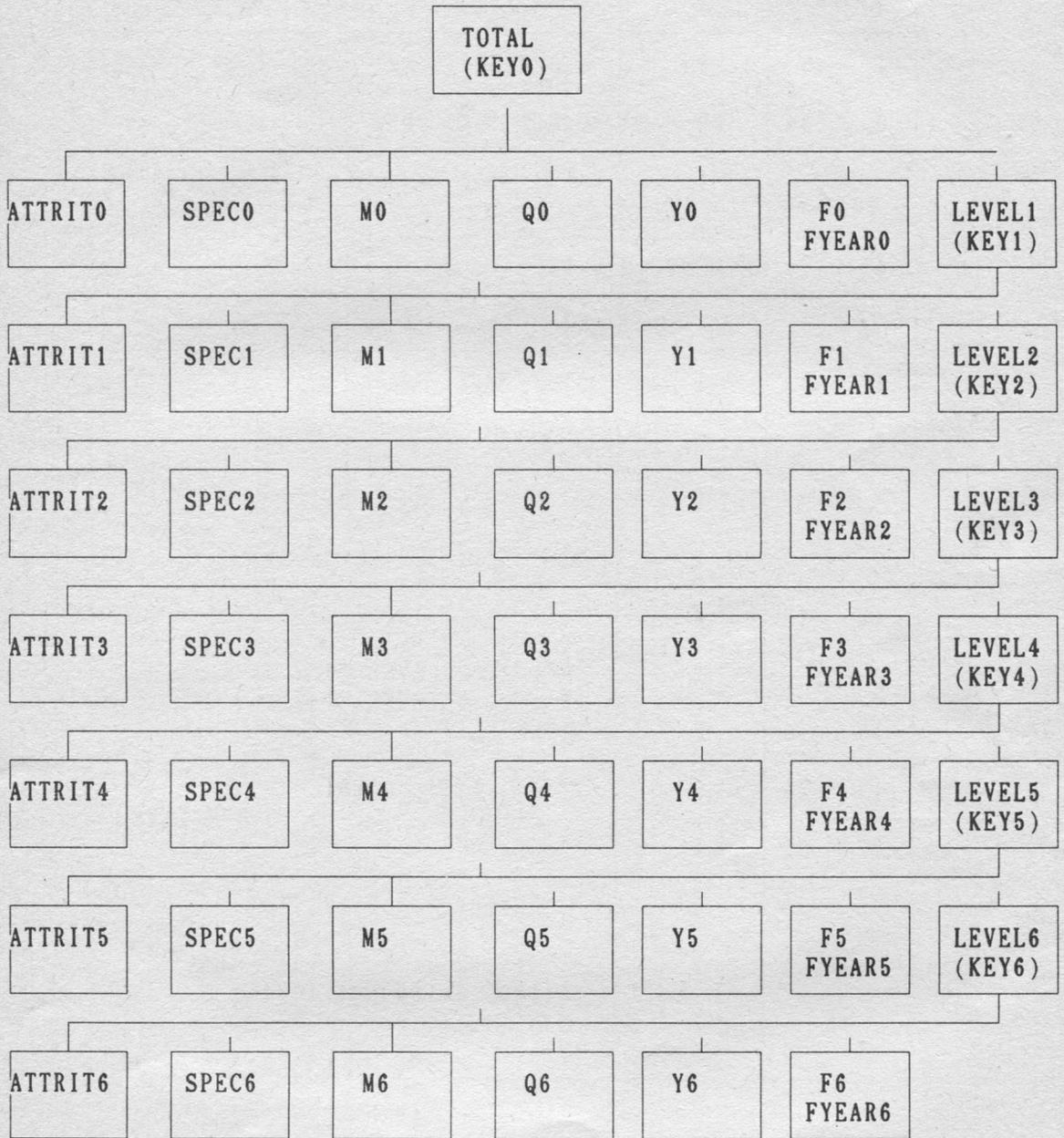
위 치	변 수 명	특 성	내 용
1 - 3	OUT_CNT	PIC'999'	RECORD 일련 번호 KEY CHAR(60)
4 - 74	OUT_DATA	CHAR(74)	작업 대상 입력 DATA
78 -277	OUT_TAB	CHAR(100)	색인화된 자료 : 10개 까지
278-280	OUT_P	PIC'999'	작성된 색인어수

3.3.6 도서 FILE : BOOK

위 치	변 수 명	특 성	내 용
1 - 8	IN_KEY	CHAR(8)	도서 분류 번호
9 - 38	IN_HAN	CHAR(30)	도서 한글명
39 - 68	IN_ENG	CHAR(30)	영문 도서명
69 - 80	FILLER	CHAR(12)	사용 않함

3.3.7 통계 D.B 구조

3.3.7.1 D.B 구조도



3.3.7.2 통계 D.B SEGMENT 내용

① TOTAL SEGMENT(35 byte)

칼 램	길 이	내 용
1 - 5	5	D.B NAME
6 - 34	29	D.B의 영문 명칭
35	1	주기 사항 유무 표시 (* 로 나타냄)

② ATTRITO SEGMENT(40 byte)

칼 램	길 이	내 용
1 - 6	6	자료 출처 코드
7 - 9	3	
10	1	지수 누계 평균 계산여부(M 표시)
11	1	보안 형태 표시(M:월, Q:분기)
12 - 13	2	보안 대상 자료의 최종 기간
14	1	'P': 하위 LEVEL 단위 사용 'W': 하위 LEVEL 가중치 사용
15 - 16	2	총 LEVEL의 갯수
17 - 21	5	자료의 타입 RAB01 분야 <--- ---> 차하위 LEVEL
22 - 32	11	단위
33 - 38	6	각 LEVEL(총 6개)의 크기
39	1	총 가중치의 크기, 10의 지수로 표시. (2:100, 3:1000, 4:10000)
40	1	소수 이하의 자리수

③ ATTRIT_i (i = 1,2,36) SEGMENT (15 byte)

칼 램	길 이	내 용
1 - 4	4	가중치
5 - 14	10	단위
15	1	소숫점 자리수

④ LEVEL_i (i = 1,2,3,4,5,6) SEGMENT (39 byte)

칼 램	길 이	내 용
1 - 9	9	D.B KEY
10 - 36	27	D.B의 영문 명칭
37	1	'&'일 경우 ATTRIT SEG의 소숫점 이용
38	1	'%'일 경우 ATTRIT SEG의 단위 이용
39	1	'*'일때 주기 사항 처리

⑤ SPEC_i (i = 1,2,3,4,5,6) SEGMENT (30 byte)

칼 램	길 이	내 용
1 - 30	30	D.B 자료 한글명

⑥ M_i (i = 1,2,3,4,5,6) SEGMENT (1458 byte)

칼 램	길 이	내 용
1 - 2	2	시작년
3 - 4	2	시작월
5 - 6	2	최종년
7 - 8	2	최종월
9 -1458	1450	FIXED BIN(31,0) 360개(30 년분)

⑦ Qi (i = 1,2,3,4,5,6) SEGMENT (490 byte)

칼 램	길 이	내 용
1 - 2	2	시작년
3 - 4	2	시작분기
5 - 6	2	최종년
7 - 8	2	최종분기
9 -1458	1450	FIXED BIN(31,0) 120개(30 년분)

⑧ Yi (i = 1,2,3,4,5,6) SEGMENT (138 byte)

칼 램	길 이	내 용
1 - 2	2	시작년
3 - 4	2	0
5 - 6	2	최종년
7 - 8	2	0
9 -138	1450	FIXED BIN(31,0) 30개(30 년분)

⑨ Fi (i = 1,2,3,4,5,6) SEGMENT (10 byte)

칼 램	길 이	내 용
1 - 2	2	자료 생산 년도
3 - 6	4	
7 - 10	4	FIXED BIN(31,0)

4. 구현 및 평가

4. 구현 및 평가

4.1 구현

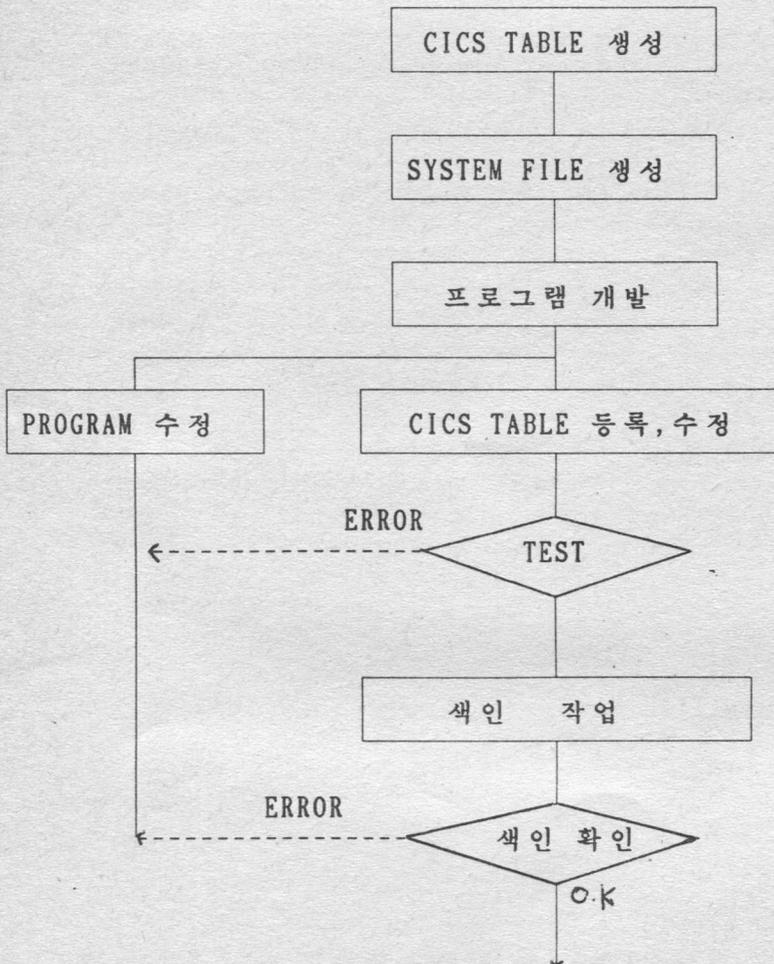
4.1.1 구현 환경

- 컴퓨터 기종 : IBM-4381/P03
- O/S : MVS/OS
- 사용 언어 : PL/1
- ON_LINE TOOL: CICS

4.1.2 구현 특성

- 작업의 효율적인 운영을 위하여 ON_LINE 시스템으로 개발
- 개발자가 직접 CICS TEST RESION을 사용하여 개발 즉, 관련된 CICS의 SYSTEM TABLE을 작성하고, 개발 진도에 따라 수시 변환

4.1.3 구현 과정



4.1.4 프로그램 및 FILE

프로그램, FILE 이름	위치 및 특성	내 용
LDHLI000	LDH.TEST.LIBRARY	색인 시스템 MAIN PROGRAM
LDHLI001	"	통계 D.B 색인 작업 MAIN
LDHLI002	"	도서 자료 " "
LDHLI003	"	화면 입력 자료 " "
LDHLI004	"	어절 분리 PROGRAM
LDHLI005	"	불용어 처리 PROGRAM
LDHLI006	"	조사 처리 PROGRAM
LDHLI007	"	색인어 FILE 작성 PROGRAM
LDHLIM1	"	작업 선택 MAIN MAP
LDHLIM2	"	화면 입력 및 출력 MAP
LDHLIV01	"	색인 FILE 초기 생성 PGM
LDHLIL01	"	색인어 확인 LISTING PGM
B40.LI.INDEX	DBD003, VSAM	색인 FILE
B40.LI.PRINT	DBD003, SAM	색인 확인 FILE
DBD.LB.HELP	DBD002, VSAM	도서 FILE

* 참고 사항

○ PROGRAM명 표준화 내역

1 - 3 : 작성자 INITIAL NAME

6 : 프로그램 특성

O : ON_LINE

M : MAP

V : VSAM FILE 생성

L : LISTING

B : BATCH

R : RUN STREAM 등

7 - 8 : PROGRAM 일련 번호

4 - 5 : 업무 코드

PM : 인사

LI : 인덱스 등

단 MAP은 한자리 즉 7 COL 사용

4.1.5 CICS ON_LINE SYSTEM PROGRAM (LDH.TEST.PROGRAM)

프로그램명	특 성	내 용
CICSRUN	RUN STREAM	CICS 실행 PROGRAM
C00LIB	CICS DATA SET	CICS에서 사용하는 시스템 LIBRARY 생성
C01SIT	SYSTEM INIT LIB	CICS SYSTEM 생성
C02FCT	FILE CONTROL TABLE	사용할 FILE 정의
C03PCT	PGM CONTROL TABLE	TRANSACTION NAME 등록
C04PPT	PGM PRODUCT TABLE	APPLICATION PGM 등록
C05DDIR	DBD PRODUCT TABLE	사용할 D.B의 DBD 등록
C06PDIR	PCB PRODUCT TABLE	D.B의 해당 PSB 등록
C07ACB	APPL CONTROL BLOCK	ON_LINE 수행 ROUTINE 설정
C08TCT	TERM CONTROL TABLE	사용 단말기 정의
C09SNT	SIGN ON TABLE	사용자 권한 정의
D01DBD	DBD GEN	DBD 생성
D02PSB	PSB GEN	PSB 생성

4.2 평가

시스템의 평가를 위하여 아래의 평가 자료를 입력하여 색인어 작성 효율을 평가 하고자 한다.

○ 평가 입력 자료

지금 우리 경제가 안고 있는 어려운 문제의 원인에 대한 진단의 차이가 있고 시각의 차가 있을 수도 있습니다. 진단의 차와 시각의 차 때문에 또 처방의 차도 있을 수도 있습니다. 사실 어떻든 엄청난 문제들을 가지고 있는 것만은 사실입니다. '73년 중화학공업 선포 이후 계속 발전을 해왔습니다만 그때에는 저기술, 1차적인 기술로서도 생산성이 증대될 수 있는 그러한 성장의 시기였다고 볼 수가 있습니다. 이제는 모든 가격의 경쟁력, 노동력 등등에 있어서 불리한 입장에 있기 때문에 우리는 보다 발달된 기술에 의한 생산성 증대에 의한 경쟁을 통해서만 가능한 시기에 와 있습니다. 지금 한국은 일본의 '66년에 해당하며 2000년에 가야 '73년의 일본에 해당한다고 이야기를 하고 있고 일본을 현재 100으로 보면 한국은 20이 된다고 하고 있습니다. 이것은 제가 보전에 잘 봐준 것이라고 보는데, 현재의 한국을 100으로 봤을 적에 대만, 태국이 얼마나 하면 87, 85로 굉장히 높이 올라 왔습니다. 지금 우리가 고도 성장의 우등생이라고 해서 자만하는 동안에 말레이시아, 태국은 아주 바짝 쫓아오고 있습니다.

○ 평가 자료에 불용어 및 조사를 표시한 자료

지금 우리 경제(가) 안고 있는 어려운 문제의 원인에 대한 진단(의) 차(이)가 있고 시각(의) 차(가) 있을 수도 있습니다. 진단(의) 차(와) 시각(의) 차(때)문에 또 처방(의) 차(도) 있을 수도 있습니다. 사(실) 어떻든 엄청난 문제(들)을 가지고 있는 것만은 사실입니다. '73년 중화학공업 선포(이)후 계속 발전을(을) 해왔습니다만 그때에는 저기술, 1차적(인) 기술(로)서도 생산성이 증대될 수 있는 그러한 성장(의) 시기였(다)고 볼 수가 있습니다. 이제는 모든 가격(의) 경쟁력, 노동력 등등에 있어서 불리한 입장에(에) 있기 때문에 우리는 보다 발달(된) 기술(에)에 의한 생산성 증대(에)에 의한 경쟁(을) 통해서만 가능한 시기에 와 있습니다. 지금 한국(은) 일본(의) '66년(에) 해당하며 2000년(에) 가야 '73년(의) 일본(에) 해당(하)다고 이야기(를) 하고 있고 일본(을) 현재 100(으)로 보면 한국(은) 20(이) 된다고 하고 있습니다. 이것은 제가 보(전)에 잘 봐(주)는 것(이)라고 보(는)데, 현재(의) 한국(을) 100(으)로 봤(을) 적(에) 대만, 태국(이) 얼마나 하(면) 87, 85(로) 굉장히 높이 올라 왔(습)니다. 지금 우리(가) 고도 성장(의) 우등생(이)라고 해서 자만(하)는 동안(에) 말레이시아, 태국(은) 아주 바짝 쫓(아)오고 있습니다.

참고) 불용어 _____ , 조사 ○

○ 불용어와 조사를 삭제한 대상 색인어

우리/ 경제/문제/원인/
 진단/차이/시각/차/
 진단/차/시각/차/처방/차/
 엄청난/문제/
 사실/
 73년/중화학공업/선포이후/계속/발전/
 저기술/1차/기술/생산성/증대/
 성장/시기/
 가격/경쟁력/노동력/
 입장/우리/발달/기술/
 생산성/증대/경쟁/성장/
 가능/시기/
 한국/일본/66년/해당/2000년/
 73년/일본/해당/이야기/일본/
 현재/100/한국/20/
 잘/현재/한국/
 100/대만/태국/87/
 85/우리/고도/성장/
 우등생/자만/말레이시아/태국/

○ 자동 추출 1차 실행 결과

우리 경제 안고 어려운 문제 원인
 진단 차이 시각 차 수
 진단 차 시각 차 때문 처방 차 수
 어떠한 엄청난 문제 가지
 사실 입니
 '73년 중화학공업 선포이후 계속 발전 해왔습니다
 그때 저기술 1차적인 기술로서 생산성 증대 될
 수 그러한 성장 시기였다 볼 수
 이제 모든 가격 경쟁력 노동력 등등
 불리한 입장 때문 우리 발달된 기술
 생산성 증대 경쟁 통해서 성장
 가능한 시 와있습니
 한국 일본 '66년 해당 2000년 가
 '73년 일본 해당한다 이야기 일본
 현재 100 보 한국 20 된다
 제 보건데 잘 봐준 보는데 현재 한국
 100 봤을적대 태국 얼마 하 87
 85로 굉장히 높 올라 왔습니
 우등생 해 자만 말레이지 태국 아주
 바짝 쫓아오

○ 1차 실행 결과 색인어는 81.69%의 추출 성능을 나타냈으나 아직 만족할만한 성능은 아니다. 이것은 해당 불용어의 부족과 검출 대상 조사의 부족으로 인한 것인바 이를 해결하기 위하여 1차 실행 자료를 분석하여 불필요한 어절을 불용어 TABLE에 추가 하고, 미검출된 조사를 조사 TABLE에 추가하여 다시 실행하였다.

○ 1차 실행 자료에 불용어와 조사를 표시한 자료

우리 경제 안고 어려운 문제 원인
 진단 차이 시각 차 수 문제 원 인
 진단 차 시각 차 때 문 처 방 차 수
 어 떻 든 입 니
 사실 입 니
 ' 7 3 년 중 화 학 공 업 선 포 이 후 계 속 발 전 해 왔 습 니 다
 그 때 저 기 술 1 차 적 인 기 술 문 서 계 속 생 산 성 증 대 될
 수 그 리 한 성 장 시 기 었 다 불 용 력 수 등 등
 이 제 모 든 가 격 경 쟁 력 노 불 달 된 기 술
 불 리 한 입 장 때 문 경 쟁 우 리 노 발 달 된 기 술
 생 산 성 증 대 와 있 습 니 통 해 서 성 장
 가 능 하
 한 국 일 본 ' 6 6 년 해 당 2 0 0 0 년 가
 ' 7 3 년 일 본 해 당 한 다 이 야 기 일 본
 현 제 1 0 0 보 한 국 2 0 된 다
 제 보 건 데 잘 보 준 보 는 데 현 재 한 국
 1 0 0 봤 을 적 대 태 국 얼 마 하 8 7
 8 5 된 평 장 히 높 올 라 왔 습 니
 우 등 생 해 자 만 말 레 이 지 태 국 아 주
 바 짝 쫓 아 오

○ 추가 수록 불용어 및 조사

<불용어>

안고, 어려운, 입니*, 때문*, 가지고, 해왔*, 될, 그러*, 그렇*,
 등등*, 때*, 와있*, 보면, 된다고, 보건데, 봐*, 보아*, 보는*,
 봤*, 하면, 높*, 올*, 왔*, 해서, 아주, 바짝, 불리*, 지금, 대한

<조사>

적인, 로서, 었다, 된, 한다, 로

○ 2차(최종) 실행 검출 색인 결과 자료

우리 경제 문제 원인
 진단 차이 시각 차 수
 진단 차 시각 차 처방 차 수
 엄청난 문제
 사실
 '73년 중화학공업 선포 이후 계속 발전
 그때 저기술 1차 기술(루) 생산성 증대
 수 성장 시기 복 수
 이제 가격 경쟁력 노동력
 입장 우리 발달 기술 동력
 생산성 증대 경쟁 통해서 성장
 가능(한) 시
 한국 일본 '66년 해당 2000년 가
 '73년 일본 해당(하다) 이야기 일본
 현재 100 한국 20
 잘 현재 한국
 100 대○ 태국 얼마 87
 85 우리 고○ 성장
 우등생 자만 말레이지○ 태국

○ 색인이 검출 효율

평가 어절			색인 추출 결과					
최초	1차	2차	대상	1차	효율	2차	효율	비고
139	108	81	71	58	81.69	64	90.14	

○ 평가

1차 실행 결과 81.69%의 색인 작성 효율을 얻었으며, 1차 결과 자료를 분석하여 새로운 불용어와 조사를 추가하여 2차 실행 결과 90.14%의 상승된 결과를 얻었다. 이것으로 보면 조사와 불용어를 정확하게 분석하여 완전한 불용어 및 조사 TABLE을 구축 한다면 95% 이상의 정교한 색인어를 얻을 수 있으며 이는 수작업 작성 보다 높은 효율을 갖게 될 것이다.

5. 결 론

5. 결론

색인어 자동 추출을 위하여 많은 이론들이 제시되고 있으나 한글 색인 자동 추출 시스템의 개발은 거의 없는 형편인바, 이것은 한글의 특수한 문법적 성격으로 인하여 100%의 색인어 작성이 불가능한것에 기인한다.

한글의 가장 특수한 성격은 조사의 생성 원리에따라 무한한 어절이 파생 되며 한문과 한글의 혼합사용으로 인하여 확실한 뜻의 구분이 어려우며, 사전을 구축한다 하여도 완전한 색인의 검출은 어렵고, 사전 자체의 구축에 상당한 시간과 기억 장치의 소모를 요구 한다.

때문에 본 연구에서는 사건의 구축을 배제하고 한글의 문법적 체계를 분석하여 조사의 생성원리와 색인어의 대상이 안되는 불용어를 이용한 색인어의 자동 추출 시스템을 구현해본 결과 1차로 81.69%의 추출율을 나타냈으며, 이를 분석하여 새로운 조사와 불용어를 파악하고 이를 다시 해당 TABLE에 추가하여 2차 실행 하여 90.14%의 추출율을 얻는 결과를 얻을수 있었다. 그러나 현 시스템으로는 색인어 대상 단어가 조사와 같은 어미를 갖는경우 이의 완전한 추출은 어렵다. 즉 '말레이시아'에서 '아'가 조사로서 취급되어 삭제되므로 '말레이지'만 색인어로 추출되는 오류들을 나타냈다. 따라서 현 시스템에서는 담당자가 결과를 확인하는 절차가 요구 된다.

결론적으로 완전한 색인 자동 추출 위하여는 인간이 직접 색인 작업을 하는 과정을 같은 추출 시스템의 개발이 필수적이다. 이것은 인공 지능 기술이 없이는 불가능 한바 향후 인공 지능을 이용한 개발의 연구가 요구된다.

참 고 문 헌

참 고 문 헌

1. 김영희, "한국어 조사류어의 연구", 1986, 문법연구 제1집, P271-311
2. 김조천, "조사처리 기법을 이용한 한글 자동 색인 시스템 구현에 관한 연구", 석사학위 논문, 연세 대학교 산업 대학원, 1988
3. 박세영, 김권양, 오길록, "자연어 처리를 위한 한국어 SYNTAX구조에 관한 연구", 정보과학회 논문지, 제12권, 제2호, 1985, P134-151
4. 안현수, "한글 문헌의 자동 색인에 관한 실험적 연구", 석사학위 논문, 연세 대학교 산업 대학원, 1982
5. 우동진, "통계적 기법에 의한 한글 자동 색인의 연구", 한국 전자 통신 연구소 정보 관리실, 춘계 학술 발표회 논문, 1987
6. 이영주, "자동색인을 위한 한국어 형태소 분석 알고리즘에 관한 연구", 석사학위 논문, 1987
7. 이우범, "자동주제 색인의 기입 형식에 관한 연구", 국회 도서관, 1986
8. 이태영, "색인 시스템내의 자동화에 대한 고찰", 도서관, VOL.45, NO2(1986), P5-24
9. 정영미, 이태영, "자동 색인의 통계적 기법과 한국어 문헌의 실험", 도서관학 9집, 1982, P 99-118
10. Dillon, M. and Gray, A.S "FAIST: A Full automatic syntactically based indexing system," JASIS, Vol. 34, No. 2, 1983, p 99-108
11. G.Salton "A new Comparison between conventional indexing(MEDLARS) and automatic text processing(SMART)," JASIS, Vol.23, No1, 1972, P 83
12. G.Saiton, op. Cit., p 29
13. H.P. Luhn, "A Statistical approach to mechanized encoding and searching of literary information," IBM Journal of research and development, 1957, p 307-317
14. J.Braum and C.Schwind, "Automatic, sematica-based indexing of natural language texts for information retrieval system," Information processing and management, Vol.12, No.2, 1976, p147

15. Pao.M.L, "Automatic text analysis based on transition phenomena of word occurrence," JASIS, Vol.29, No.3,1978, p121-124
16. Rowbottom, M.E. and Willett, p."The effect of subject matter on the automatic indexing of full text," SASIS, Vol. 33, No.3, 1982, p130-137
17. Saiton, G. "A blueprint for automatic indexing," ACM SIGIR, Vol.16 No.2,1981, p22-38.
18. W.A. Gray and A.J. Harley, "Computer assisted indexing," Information storage and retrieval, Vol.7, No.4, 1971, p167-174
19. W.A Meulen and P.J.F.C. Janssen, "Automatic Versus manual indexing," Information processing and management, Vol.13, No.1, 1977. p13-21

6. 참 고 사 항

6. 참고 사항

- 한글 불용어 TABLE
- 영문 불용어 TABLE
- 조사 TABLE
- IBM 한글 코드 특성
- ON_LINE SYSTEM PROGRAM
 1. CICSRUN(CICS 실행 프로그램)
 2. COOLIB(LIBRARY 생성)
 3. C01SIT(CICS 시스템 생성)
 4. C02FCT(사용 FILE 등록)
 5. C03PCT(TRANSACTION NAME 등록)
 6. C04PPT(APPLICATION PROGRAM 등록)
 7. C05DDIR(D.B 등록)
 8. C06PDIR(PSB 등록)
 9. C07ACB(ON_LINE CONTROL 등록)
 10. C08TCT(사용 단말기 등록)
 11. C09SNT(사용자 권한 정의)
 12. D01DBD(D.B 구조 생성)
 13. D02PSB(D.B PSB 정의)

< 한글 불용어 TABLE >

불용어	특성	길이	불용어	특성	길이	불용어	특성	길이
따른		4	훨씬		4	서로		4
먼저		4	보기		4	서작		4
미리		4	본고		4	신속	히	4
만든		4	빌어		4	새로	운	6
물론		4	빈번하다		8	적절	히	6
명확히		6	부분		4	전체		4
최근		4	관한		4	지금		4
찾아내	는	8	거쳐	야	6	지난		4
취한		4	걸쳐		4	지니	고	4
어떤		4	값의		4	장래		4
아울러		6	각각		4	자세	히	6
임의		4	각종		4	주어	진	6
이미		4	간단	히	6	주도	일	6
이와		4	간의		4	줄줄	이	6
이어		4	간략	히	6	주	고	6
이의		4	찾추	어야	8	통	한	4
이후		4	가장		4	특	히	4
이더		4	가진		4	커	지	6
이르		4	갖고		4	크	게	4
인한		4	같이		4	다	루	4
이전		4	기할		4	살	퍼	4
의한		4	기타		4	미	치	4
연		4	그의		4	대	해	4
열	렸	6	그리	고	6	위	해	4
열	던	4	급속	히	6	위	하	4
여러		4	근래		4	위	하	4
동안		4	겪고		4	하	것	4
동시		4	경우		4	하	였	4
동기		4				하	기	4
동면		4	쓰이		4	다	기	4
당		4	쓰일		4	보	다	2
다양		4	높이	고	6	엄	*	2
다른		4	높이	기	6	때	*	2
두어		4	널리		4	또	*	2
대한		4	나아가	야	8	있	*	2
			나누	어	6	않	*	2
향후		4	나타	내	6	것	*	2
하나		4						
한편		4						

< 영문 불용어 TABLE >

불용어	값	불용어	값	불용어	값	불용어	값
SPACE	5	DATA	4	NEITHER	7	THERE	5
A	1	DID	3	NEVER	5	THERBY	6
ABLE	4	DO	2	NO	2	THESE	5
ABOUT	5	DOES	4	NONE	4	THEY	4
ABOVE	5	DOING	5	NOR	3	THIS	4
AFTER	5	DONE	4	NOT	3	THOSE	5
AGAIN	5	DURING	6	NOTHING	7	THOUGH	6
AGAINST	7	EACH	4	NOW	3	THROUGH	7
ALL	3	ETHER	5	OF	2	THUS	4
ALLMOST	7	EVEN	4	OFF	3	TO	2
ALONE	5	EVER	4	OFTEN	5	TOO	3
ALONG	5						
ALEADY	6	EVERY	5	ON	2	TOWORDS	7
ALSO	4	FEW	3	ONCE	4	UNDER	5
AMONG	5	FOR	3	ONLY	4	UNLESS	6
AN	2	FROM	4	OR	2	UNTIL	5
AND	3	GET	3	OTHER	5	UP	2
ANOTHER	7	GO	2	OUR	3	UPON	4
ANY	3	HAD	3	OURS	4	US	2
ARE	3	HOWEVER	7	OUT	3	VERY	4
AS	2	IF	2	OVER	4	WAS	3
AT	2	IN	2	PUT	3	WAY	3
AWAY	4	INSTEAD	7	RATHER	6	WE	2
BE	2	INTO	4	REALLY	6	WERE	4
BECAUSE	7	IS	2	SAME	4	WHAT	4
BECOME	6	IT	2	SHE	3	WHEN	4
BEEN	4	ITS	3	SHOULD	6	WHERE	5
BEFORE	6	ITSELF	6	SINCE	5	WHETHER	7
BEHIND	6	JUST	4	SO	2	WHICH	5
BEING	5	LET	3	SOME	4	WHILE	5
BETWEEN	7	LIKE	4	STILL	5	WHO	3
BEYOND	6	LIKELY	6	SUCH	4	WHY	3
BOTH	4	MANY	4	THAN	4	WILL	4
BUT	3	MIGHT	5	THAT	4	WITH	4
BY	2	MORE	4	THE	3	WITHIN	6
CAN	3	MOST	4	THEIR	5	WITHOUT	7
CANNOT	6	MUCH	4	THEM	4	YET	3
COULD	5	MUST	4	THEN	4	YOU	3
						YOUR	4

< 조 사 TABLE (총괄) >

조 사	종 류	수 량
가, 걸, 고, 군, 계, 과		6
나, 냐, 는, 니		4
다, 도, 들, 데, 되		5
라, 란, 라, 러, 로, 룬, 료, 를		8
만, 먼, 며, 면		4
서, 손		2
아, 야, 여, 요, 은, 을, 이, 에, 의, 와		10
자, 즉, 지		3
씨		1
합 계		43

< 범 례 >

내 용	구 분
· 받침있는 체언뒤 조사 존재	0
· " " 없는 " "	X
· 받침 있거나 없는 체언뒤 조사 존재	0X

기본 조사 : 군, 계, 과

순번	조 사	길 이	비 고
1	군	2	X
2	이군	4	0
3	더군	4	X
4	로군	4	X
5	이더군	6	0
6	이로군	6	0
1	계	2	X
2	에게	4	X
3	이에게	6	0
4	들에게	6	X
5	이들에게	8	0
1	과	2	0
2	들과	4	0X

기본 조사 : 가, 걸, 고

순번	조 사	길 이	비고
1	가	2	X
2	이가	4	0
3	다가	4	X
4	는가	4	X
5	런가	4	X
6	인가	4	0
7	에다가	6	0X
8	이런가	6	0
1	걸	2	0
2	인걸	4	X
3	들인걸	6	0X

순번	조 사	길 이	비고
1	고	2	X
2	이고	4	0
3	라고	4	X
4	하고	4	X
5	인고	4	0X
6	되고	4	0
7	이로고	6	0
8	이라고	6	0
9	이하고	6	0
10	이되고	6	0
11	가되고	6	X
12	같으니라고	10	0X

기본 조사 : 나, 냐

순번	조 사	길 이	비고
1	나	2	X
2	이나	4	0
3	더구나	6	X
4	로구나	6	X
5	이로구나	8	0
6	이더구나	8	0

순번	조 사	길 이	비고
1	냐	2	X
2	이냐	4	0
3	더냐	4	X
4	이더냐	6	0

기본 조사 : 는, 니

순번	조 사	길 이	비고
1	는	2	X
2	라는	4	X
3	하는	4	0X
4	로는	4	X
5	에는	4	X
6	이는	4	0
7	이라는	6	0
8	으로는	6	0
9	런마는	6	X
10	하런마는	8	0

순번	조 사	길 이	비고
1	니	2	X
2	이니	4	0
3	하니	4	X
4	러니	4	X
5	더니	4	X
6	이더니	6	0
7	이러니	6	0

기본 조사 : 다, 도

순번	조 사	길 이	비고
1	다	2	X
2	이다	4	0
3	에다	4	X
4	로다	4	X
5	란다	4	X
6	한다	4	0
7	켰다	4	X
8	이로다	6	0
9	이란다	6	0

순번	조 사	길 이	비고
1	도	2	X
2	이도	4	0
3	로도	4	X
4	에도	4	X
5	라도	4	X
6	으로도	6	0
7	이라도	6	0
8	더라도	6	X
9	이더라도	8	0
10	하더라도	8	X
11	치더라도	8	X

기본 조사 : 들, 데, 되

순번	조 사	길 이	비고
1	들	2	OX
2	런들	4	X
3	이런들	6	0
1	데	2	
2	이데	4	OX
3	인데	4	OX
4	들인데	6	OX
1	되	2	
2	로되	4	X
3	이로되	6	0

기본 조사 : 라, 란, 라, 러

순번	조 사	길 이	비고
1	라	2	X
2	이라	4	0
3	더라	4	X
4	로라	4	X
5	이더라	6	0
6	이로라	6	0
1	란	2	X
2	이란	4	0
3	으란	4	0
4	들이란	6	OX
11	랴	2	X
1	이랴	4	0
2	들이랴	6	OX
3	러	2	
4	더러	4	0
5	이더러	6	X

기본 조사 : 로, 른, 료, 를

순번	조 사	길 이	비고
1	로	2	X
2	으로	4	0
3	에로	4	X
4	에게로	6	X
5	이에게로	8	0

순번	조 사	길 이	비고
1	룬	2	X
2	으룬	4	0
1	료	2	
2	구료	4	X
3	로구료	6	X
1	를	2	X
2	이를	4	0
3	예를	4	0

기본 조사 : 만, 언, 며, 면

순번	조 사	길 이	비고
1	만	2	X
2	이만	4	0
3	련만	4	X
4	라야만	6	X
5	이련만	6	0
6	이라야만	8	0
1	면	2	
2	구면	4	X
3	로구면	6	X
4	이러구면	8	0

순번	조 사	길 이	비고
1	며	2	X
2	이며	4	0
3	으며	4	0
4	하며	4	0
1	면	2	X
2	이면	4	0
3	라면	4	X
4	이러면	6	0
5	들이러면	8	0X

기본 조사 : 서, 손

순번	조 사	길 이	비고
1	서	2	X
2	로서	4	X
3	에서	4	0
4	으로서	6	0
5	에게서	6	0
6	이로서	6	0
1	손	2	
2	다손	4	0

기본 조사 : 아, 야, 예, 요, 은

순번	조 사	길 이	비고
1	아	2	X
1	야	2	X
2	이야	4	0
3	래야	4	X
4	이래야	6	0
5	이러야	6	0
1	여	2	X
2	이여	4	0
3	하여	4	0
1	요	2	X
2	리요	4	X
3	이리요	6	0X
1	은	2	0

기본 조사 : 을, 이, 예, 의, 와

순번	조 사	길 이	비고
1	을	2	0
2	들을	4	0X
1	이	2	0
2	들이	4	0X
1	예	2	X
2	이에	4	0
3	기에	4	X
4	들에	4	X
5	이기에	6	0

순번	조 사	길 이	비고
1	의	2	X
2	예의	4	0
3	이의	4	0
4	들의	4	0X
1	와	2	X
2	이와	4	0
3	러니와	6	X
4	이러니와	8	0

기본 조사 : 자, 즉, 지, 써

순번	조 사	길 이	비고
1	자	2	
2	이	4	OX
3	들이	6	OX
1	즉	2	
2	인	4	OX
3	한	4	OX

순번	조 사	길 이	비고
1	지	2	X
2	이	4	0
3	라	4	X
4	튼	4	X
5	인	4	OX
6	이	6	0
7	라	6	0
8	이	8	0
1	라	2	0
2	써	4	X
3	으	6	0

〈 IBM 한글 코드 특성 〉

2 - BYTE CODE의 범위

1st BYTE	2nd BYTE	내 용
X'40'	X'40'	BLANK
X'41'	X'41' - X'FE'	사용하지 않는 CODE
X'42'	X'41' - X'FE'	큰 크기 영문/숫자/특수문자
X'43' - X'83'	X'41' - X'FE'	사용하지 않는 CODE
X'84' - X'D3'	X'41' - X'FE'	한글
X'D4' - X'FE'	X'41' - X'FE'	사용하지 않는 CODE

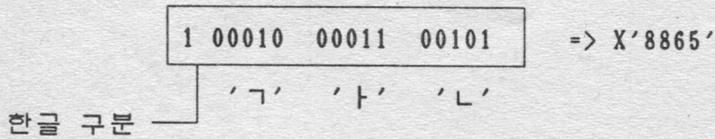
한글 자모의 BIT 구성

Bit 01234	초 성	중 성	종 성
00000	not used ***	restricted *	restricted *
00001	fill code **	restricted *	fill code **
00010	ㄱ	fill code **	ㄱ
00011	ㄴ	ㅏ	ㄴ
00100	ㄷ	ㅑ	ㄷ
00101	ㄹ	ㅓ	ㄹ
00110	ㅍ	ㅕ	ㅍ
00111	ㅎ	ㅗ	ㅎ
01000	ㅊ	restricted *	ㅊ
01001	ㅋ	restricted *	ㅊ
01010	ㆁ	ㅛ	ㄹ
01011	ㆂ	ㅜ	ㄹ
01100	ㆃ	ㅠ	ㄹ
01101	ㆄ	ㅡ	ㄹ
01110	ㆅ	ㅣ	ㄹ
01111	ㆆ	ㅣ	ㄹ
10000	ㅈ	restricted *	ㅈ
10001	ㅋ	restricted *	ㅊ
10010	ㅌ	ㅓ	ㅊ
10011	ㅍ	ㅕ	ㅊ
10100	ㅎ	ㅗ	ㅊ
10101	not used ***	ㅛ	ㅊ
10110	not used ***	ㅜ	ㅊ
10111	not used ***	ㅠ	ㅊ
11000	not used ***	restricted *	ㅊ
11001	not used ***	restricted *	ㅊ
11010	not used ***	ㅓ	ㅊ
11011	not used ***	ㅡ	ㅊ
11100	not used ***	ㅣ	ㅊ
11101	not used ***	ㅣ	ㅊ
11110	not used ***	not used ***	not used ***
11111	restricted *	not used ***	restricted *

한글 2-BYTE CODE의 특징

- 한글 2-BYTE중 첫번 1 BIT는 한글 구분을 위한 BIT로 한글일 경우 항상 ON.
- 초, 중, 종성은 각각 5 BITS로 구성
- 2-BYTE EBCDIC문자는 EBCDIC HEX값 앞에 X'42'를 붙여서 2-BYTE CODE값 부여.
- 모든 조합의 한글 표현 가능.
- 한글의 CODE값 자체가 '가나다'순으로 정해져 있어서 한글을 SORT할 때에는 CODE 값으로 자체 SORTING 가능
- SPACE는 X'4040'으로 표시
- 다른 문자의 추가 가능
- 한글 조합

예) 간 : ㄱ(초성) ㅏ(중성) ㄴ(종성)



- 2 - BYTE EBCDIC 문자 조합

예) A : X'C1'

A : X'42' + X'C1' => X'42C1'

< ON_LINE SYSTEM PROGRAM >

1. CICSRUN(CICS 실행 프로그램)

```
//B40CIRUN JOB CLASS=A,MSGCLASS=X,MSGLEVEL=(1,1)
//CICSRUN EXEC PGM=DFHSSIP,REGION=3000K,TIME=1439,
// PARM=('SIT=CT,TCT=CT')
//STEP1CAT DD DSN=CATALOG.IDBD001,DISP=SHR
//DFHRPL DD DSN=CICSVS.LOADLIB2,DISP=SHR
// DD DSN=B40.TEST.LOADLIB,DISP=SHR
// DD DSN=DBD.CICS.LOADLIB,DISP=SHR
// DD DSN=DBD.KOSIS.LOADLIB,DISP=SHR
// DD DSN=CICSVS.LOADLIB,DISP=SHR
// DD DSN=CICSVS.LOADLIB1,DISP=SHR
// DD DSN=OLD.PLILINK,DISP=SHR
// DD DSN=IMSVS.RESLIB,DISP=SHR
//DFHDMPIA DD DSN=&&MCICS1,DISP=(,DELETE,DELETE),
// UNIT=DISK,SPACE=(CYL,5)
//DFHRSD DD DSN=B40.TEST.RDO,DISP=SHR
//DFHTEMP DD DSN=B40.TEST.TEMP,DISP=SHR
//IMSACB DD DSN=DBD.KOSIS.ACBLIB,DISP=SHR
//DFSVSAMP DD DSN=DBD.IMSVS.SSSLIB(DFSVSAMP),DISP=SHR
//***** USER DATA SETS VSAM *****
//DBKOSIS DD DSN=DBD.DB.KOSIS,DISP=SHR
//INDEX DD DSN=B40.LI.INDEX,DISP=SHR
//BOOK DD DSN=DBD.LB.HELP,DISP=SHR
```

2. COOLIB(LIBRARY 생성)

```
//B40DMSIN JOB CLASS=E,MSGCLASS=X
//STEP1 EXEC PGM=IEFBR14
//SYSPRINT DD SYSOUT=*
//DD1 DD DSN=B40.TEST.DBDLIB,SPACE=(CYL,(1,1,10)),
// DCB=(RECFM=U,BLKSIZE=6144,DSORG=PO),UNIT=DISK,
// VOL=SER=DBD003,DISP=(NEW,KEEP)
//SYSIN DD DUMMY
/*
//STEP2 EXEC PGM=IEFBR14
//SYSPRINT DD SYSOUT=*
//DD1 DD DSN=B40.TEST.PSBLIB,SPACE=(CYL,(1,1,10)),
// DCB=(RECFM=U,BLKSIZE=6144,DSORG=PO),UNIT=DISK,
// VOL=SER=DBD003,DISP=(NEW,KEEP)
//SYSIN DD DUMMY
/*
//STEP3 EXEC PGM=IEFBR14
//SYSPRINT DD SYSOUT=*
//DD1 DD DSN=B40.TEST.ACBLIB,SPACE=(CYL,(1,1,10)),
// DCB=(RECFM=U,BLKSIZE=6144,DSORG=PO),UNIT=DISK,
// VOL=SER=DBD003,DISP=(NEW,KEEP)
//SYSIN DD DUMMY
/*
```

```

//STEP4 EXEC PGM=IEFBR14
//SYSPRINT DD SYSOUT=*
//DD1 DD DSN=B40.TEST.LOADLIB,SPACE=(CYL,(1,1,10)),
// DCB=(RECFM=U,BLKSIZE=6144,DSORG=PO),UNIT=DISK,
// VOL=SER=DBD003,DISP=(NEW,KEEP)
//SYSIN DD DUMMY
/*
//STEP5 EXEC PGM=IEFBR14
//SYSPRINT DD SYSOUT=*
//DD1 DD DSN=B40.TEST.LIBRARY,SPACE=(CYL,(1,1,10)),
// DCB=(RECFM=FB,BLKSIZE=8000,LRECL=80,DSORG=PO),UNIT=DISK,
// VOL=SER=DBD003,DISP=(NEW,KEEP)
//SYSIN DD DUMMY
/*
//

```

3. CO1SIT(CICS 시스템 생성)

```

B40.90.302,14:41:53,LDH.TEST.LIBRARY(CO1SIT)
//B40CISIT JOB CLASS=F,MSGCLASS=X,MSGLEVEL=(0,0)
//STEP1 EXEC CICAUPLK,REGION=4096K
//ASSEM.SYSUT1 DD *

```

```

PRINT NOGEN
SITCT DFHSIT TYPE=CSECT,
SUFFIX=CT, DFHSITCK X
ABDUMP=NO, ASRB DUMMP X
AKPFREQ=200, KEYPOINT VERY FREQUENTLY X
AMXT=10, MAX ACTIVE TASKS X
APPLID=KOSIS, APPLICATION ID X
AUTINST=(20,LDHL1000), AUTO INSTOL QUEUED DEVICE X
AUXTR=OFF, AUXILIARY TRACE X
BFP=NO, DDP BUILTIN FUNCTION PROGRAM X
BMS=(FULL,COLD,ALIGN), BASIC MAPPING SUPPORT X
DBUFSZ=4096, DYNAMIC BUFFER SIZE X
DBP=2$, DYNAMIC TRANS BACKUP X
DCT=(NO,COLD), LOCAL DESTINATION FOR ISC X
DDIR=CT, DLI DMB DIRECTORY X
DIP=NO, DDP DATA INTERCHANGE PROGRAM X
DLTHRED=5, CICS DL/I NO OF THREADS X
DLI=(YES,COLD), DLI SUPPORT X
DLLPA=YES, IMS/VB MODULES LINK PACK AREA X
DMBPL=100, NO OF 1024K BYTES BLOCKS X
DTB=AUX, DYNAMIC BUFFER SPILL TO X
DUMP=NO, CANCEL OR ABEND DUMP ABNOMAL TEXT X
DUMPDS=A, DUMP DATA SET NEED A AUTO DEFULX
ENQPL=16, MAX ENQ/DEQ POOL FO PI X
EXEC=YES, EXEC CICS COMMAND USE X
EXITS=NO, X

```

```

EXTSEC=NO,
FCT=(CT,COLD),
FLDSEP=' ',
FLDSTRT='$',
ICV=10000,
ICVR=20000,
ICVS=10000,
IOCP=0,
IRCSTRT=NO,
ISC=NO,
JCT=NO,
LPA=YES,
MSGLVL=0,
MCT=NO,
MXT=200,
NLT=NO,
OSCOR=200000,
OPNDLIM=100,
PCDUMP=NO,
PCT=(CT,COLD),
PDIR=CT,
PGCHAIN=X/,
PGCOPY=C/,
PGPURGE=T/,
PGRET=P/,
PGSIZE=4096,
PISCHD=YES,
PLI=YES,
PLISHRE=NO,
PLTPI=NO,
PPT=CT,
PRINT=PA1,
PSBPL=84,
SCS=200000,
SKRPF1='1',
SKRPF5='C',
SKRPF7='P',
SKRPF8='N',
SKRPF12='L',
SRT=CT,
START=COLD,
SVD=YES,
SYSIDNT=S000,
TCP=S$,
TCT=CT,
TRACE=(125,OFF),
TS=(COLD,10,8),
WRKAREA=512,
XPSB=NO,
XTRAN=NO,
XPCT=NO,
XPPT=NO,
XTP=NO,
XFCT=NO,
ZCP=(S$,HPO)
END

```

```

EXTERNAL SECURITY RACF SIGN ON X
FILE CONTROL TABLE X
4 X'40' DDP DEFAULT FLDSEP X
FIELD START 3790 UNFORMAT INPT X
INTERVAL CONTROL VALUE X
RUNAWAY TASK TIME X
DELAY BEFORE STALL PURGE X
DEFAULT X
DLI IRC FOR DLI BATCH X
ISC LOAD MODULE X
JOURNALLING X
USE MNGMT MODULES IN LPA X
PRINT START-UP MSGS X
MONITERING CONTROL TABLE X
SPECIFY NO TERMNLS+1 X
NUCLEUS TABLE DEFAULT X
LEAVE PLENTY FOR OS 160K X
ASRA DUMP X
TRANSACTION ID X
DLI PSB DIRECTORY X
PAGE CHAINING COMMAND X
PAGE COPY COMMAND X
PAGE PURGE COMMAND X
PAGE RETRIEVAL COMMAND X
SET FOR MVS X
PROGRAM ISOLATION X
PLI TO BE INITIALIZED X
NO PL/I SHARED LIBRARY X
PROGRAM LIST TABLE X
CICS PROGRAMS X
PRINT REQUEST KEY X
DLI MAX PSB POOL SPACE X
MIN. SIZE OF 4 PAGES X
FIRST LEVEL OF PAGE X
CURRENT LEVEL OF PAGE X
PREVIOUS LEVEL OF PAGE X
NEXT LEVEL OF PAGE X
LAST LEVEL OF PAGE X
USER ERROR RECOVERY X
COLD START UNLESS OVERRIDE X
TAKE STORAGE VIOLATION DUMPS X
INTER COMMUNICATION SYSID X
TEMINAL CONTROL PROGRAM X
REMOTE & LOCAL 3277,3286 X
MAIN STORAGE TRACE TABLE X
BUFFER AND STRING TEMP STORAGE X
COMMON WORK AREAS X
PSB RACF CHECK X
EXTERNAL TRANID X
TRANID RACF CHECK X
PROGRAM RACF CHECK X
TRANS ROUTING TRANFORMAER PGM X
FILE RACF CHECK X
HIGH PERFORMANCE X

```

```

/*
//LNKEDT.SYSLMOD DD DSN=B40.TEST.LOADLIB,DISP=SHR
//

```

4. C02FCT(사용 FILE 등록)

```

B40,90.302,14:43:10,LDH.TEST.LIBRARY(C02FCT)
//B40CIFCT JOB CLASS=A,MSGCLASS=X,MSGLEVEL=(1,1)
//ASM EXEC CICAUPLK,OUTC=X,REGION=3072K
//ASSEM.SYSUT1 DD *
      PRINT NOGEN
FCTCT  DFHFCT TYPE=INITIAL,
      SUFFIX=CT
      DFHFCT TYPE=SHRCTL,
      KEYLEN=65,
      RSCLMT=30
INDEX  DFHFCT TYPE=DATASET,
      DATASET=INDEX,
      DSNNAME=B40.LI.INDEX,
      ACCMETH=VSAM,
      DISP=SHR,
      SERVREQ=(READ,BROWSE,ADD,DELETE,UPDATE),
      RECFORM=(FIXED,BLOCKED),
      FILSTAT=(ENABLED,CLOSED),
      RSL=PUBLIC,
      STRNO=1,
      LSRPOOL=1
BOOK   DFHFCT TYPE=DATASET,
      DATASET=BOOK,
      DSNNAME=DBD.LB.HELP,
      ACCMETH=VSAM,
      DISP=SHR,
      SERVREQ=(READ,BROWSE),
      RECFORM=(FIXED,BLOCKED),
      FILSTAT=(ENABLED,CLOSED),
      RSL=PUBLIC,
      STRNO=1,
      LSRPOOL=1
      DFHFCT TYPE=FINAL
      END
/*
//LNKEDT.SYSLMOD DD DSN=B40.TEST.LOADLIB,DISP=SHR
//

```

5. C03PCT(TRANSACTION NAME 등록)

```

B40,90.302,14:49:16,LDH.TEST.LIBRARY(C03PCT)
//B40CIPCT JOB CLASS=A,MSGCLASS=X,MSGLEVEL=(0,0)
//ASM EXEC CICAUPLK,OUTC=X,REGION=4096K
//ASSEM.SYSUT1 DD *
      PRINT NOGEN
PCTCT  DFHPCT TYPE=INITIAL,
      DTB=YES,
      EXTSEC=NO,
      SUFFIX=CT

```


6. C04PPT(APPLICATION PROGRAM 등록)

B40.90.302,14:53:16,LDH.TEST.LIBRARY(C04PPT)
 //B40CIPPT JOB CLASS=A,MSGCLASS=X,MSGLEVEL=(0,0)
 //ASM EXEC CICAUPLK,OUTC=X,REGION=4096K
 //ASSEM.SYSUT1 DD *

PRINT NOGEN
 PPTCT DFHPPT TYPE=INITIAL,
 SUFFIX=CT

X

**

INDX DFHPPT TYPE=ENTRY,PGMLANG=PL/I,RSL=PUBLIC,PROGRAM=LDHLI000
 DFHPPT TYPE=ENTRY,PGMLANG=PL/I,RSL=PUBLIC,PROGRAM=LDHLI001
 DFHPPT TYPE=ENTRY,PGMLANG=PL/I,RSL=PUBLIC,PROGRAM=LDHLI002
 DFHPPT TYPE=ENTRY,PGMLANG=PL/I,RSL=PUBLIC,PROGRAM=LDHLI003
 DFHPPT TYPE=ENTRY,PGMLANG=PL/I,RSL=PUBLIC,PROGRAM=LDHLI004
 DFHPPT TYPE=ENTRY,PGMLANG=PL/I,RSL=PUBLIC,PROGRAM=LDHLI005
 DFHPPT TYPE=ENTRY,PGMLANG=PL/I,RSL=PUBLIC,PROGRAM=LDHLI006
 DFHPPT TYPE=ENTRY,PGMLANG=PL/I,RSL=PUBLIC,PROGRAM=LDHLI007
 DFHPPT TYPE=ENTRY,PGMLANG=PL/I,RSL=PUBLIC,PROGRAM=LDHLIM1
 DFHPPT TYPE=ENTRY,PGMLANG=PL/I,RSL=PUBLIC,PROGRAM=LDHLIM2
 DFHPPT TYPE=ENTRY,PGMLANG=PL/I,RSL=PUBLIC,PROGRAM=LDHLIM3
 DFHPPT TYPE=ENTRY,PGMLANG=PL/I,RSL=PUBLIC,PROGRAM=LDHLIM4

*

*SYSTEM	GROUP	
AKP	DFHPPT TYPE=GROUP, FN=AKP	ACTIVITY KEYPOINT PROGRAM
AUTOSTAT	DFHPPT TYPE=GROUP, FN=AUTOSTAT	AUTO STATS SUMMARY PGM
BACKOUT	DFHPPT TYPE=GROUP, FN=BACKOUT	DYNAMIC BACKOUT
BMS	DFHPPT TYPE=GROUP, FN=BMS	BASIC MAPPING SUPPORT
EDF	DFHPPT TYPE=GROUP, FN=EDF	EXECUTION DIAGNOSTIC FACIL
FE	DFHPPT TYPE=GROUP, FN=FE	FE TERMINAL TEST PGM
INTERP	DFHPPT TYPE=GROUP, FN=INTERPRETER	COMMAND INTERPRETER
HARDCOPY	DFHPPT TYPE=GROUP, FN=HARDCOPY	3270 PRINTER-BTAM & VTAM
INQSET	DFHPPT TYPE=GROUP, FN=INQUIRESET	PROGRAMMABLE INQUIRE/SET
MVS	DFHPPT TYPE=GROUP, FN=MVS	VSAM SUBTASK MONI-MVS ONLY
OPENCLSE	DFHPPT TYPE=GROUP, FN=OPENCLSE	DYNAMIC OPEN/CLOSE PGM
OPERATRS	DFHPPT TYPE=GROUP, FN=OPERATORS	ENH FNS-MAST,SUPV&TERM OPS
STANDARD	DFHPPT TYPE=GROUP, FN=STANDARD	APPL PGMS REQD IN ALL CICS
PLI	DFHPPT TYPE=GROUP, FN=PL/I	PL/I SUPPORT IN CICS/VS
MASTTERM	DFHPPT TYPE=GROUP, FN=MASTTERM	MASTER TERMINAL FUNCTION
SIGNON	DFHPPT TYPE=GROUP, FN=SIGNON	SIGN ON & OFF PGMS & TABLES
TIME	DFHPPT TYPE=GROUP, FN=TIME	TIME ADJUSTMENT PGM
VTAM	DFHPPT TYPE=GROUP, FN=VTAM	VTAM ABNORM & ERROR...PGMS
VTAMPRT	DFHPPT TYPE=GROUP, FN=VTAMPRT	VTAM TERM CNTL PRINT KEY FN
	DFHPPT TYPE=FINAL	
	END	

/*

//LNKEDT.SYSLMOD DD DSN=B40.TEST.LOADLIB,DISP=SHR
 //

7. C05DDIR(D.B 등록)

```

B40,90.302,14:54:54,LDH.TEST.LIBRARY(C05DDIR)
//B40CIDDR JOB CLASS=F,MSGCLASS=X,MSGLEVEL=(1,0)
//STEP1 EXEC CICAUPLK,REGION=4096K
//ASSEM.SYSUT1 DD *
                PRINT NOGEN
                DFHDLDBD TYPE=INITIAL,DLI=1.3.0,SUFFIX=CT
                DFHDLDBD TYPE=ENTRY,DBD=KOSISDBD,ACCESS=RO
                DFHDLDBD TYPE=ENTRY,DBD=BOSTEST,ACCESS=RO
                DFHDLDBD TYPE=FINAL
                END DFSIDMDO
/*
//ASSEM.SYSLIB DD DSN=SYS1.MACLIB,DISP=SHR
//              DD DSN=CICSVS.MACLIB,DISP=SHR
//              DD DSN=IMSVS.GENLIB,DISP=SHR
//              DD DSN=IMSVS.GENLIBA,DISP=SHR
//              DD DSN=IMSVS.GENLIBB,DISP=SHR
//ASM.SYSLIB DD DSN=CICSVS.MACLIB,DISP=SHR
//            DD DSN=SYS1.AMODGEN,DISP=SHR
//            DD DSN=SYS1.MACLIB,DISP=SHR
//            DD DSN=IMSVS.GENLIB,DISP=SHR
//            DD DSN=IMSVS.GENLIBA,DISP=SHR
//            DD DSN=IMSVS.GENLIBB,DISP=SHR
//LNKEDT.SYSLMOD DD DSN=B40.TEST.LOADLIB,DISP=SHR
//

```

8. C06PDIR(PSB 등록)

```

B40,90.302,14:56:04,LDH.TEST.LIBRARY(C06PDIR)
//B40CIPDR JOB CLASS=F,MSGCLASS=X
//STEP1 EXEC CICAUPLK,REGION=4096K
//ASSEM.SYSUT1 DD *
                PRINT NOGEN
                DFHDLPSB TYPE=INITIAL,DLI=1.3.0,SUFFIX=CT
                DFHDLPSB TYPE=ENTRY,PSB=KOSIPSBL
                DFHDLPSB TYPE=ENTRY,PSB=KOSIPSBG
                DFHDLPSB TYPE=ENTRY,PSB=KOSIPSBA
                DFHDLPSB TYPE=ENTRY,PSB=BOSTSTG
                DFHDLPSB TYPE=FINAL
                END DFSIDIRO
/*
//ASSEM.SYSLIB DD DSN=SYS1.MACLIB,DISP=SHR
//              DD DSN=CICSVS.MACLIB,DISP=SHR
//              DD DSN=IMSVS.GENLIB,DISP=SHR
//              DD DSN=IMSVS.GENLIBA,DISP=SHR
//              DD DSN=IMSVS.GENLIBB,DISP=SHR
//ASM.SYSLIB DD DSN=CICSVS.MACLIB,DISP=SHR
//            DD DSN=SYS1.AMODGEN,DISP=SHR
//            DD DSN=SYS1.MACLIB,DISP=SHR
//            DD DSN=IMSVS.GENLIB,DISP=SHR
//            DD DSN=IMSVS.GENLIBA,DISP=SHR
//            DD DSN=IMSVS.GENLIBB,DISP=SHR
//LNKEDT.SYSLMOD DD DSN=B40.TEST.LOADLIB,DISP=SHR
//

```

9. C07ACB(ON_LINE CONTROL 등록)

```

B40,90.321,09:42:52,LDH.TEST.LIBRARY(C07ACB)
//B40CIACB JOB CLASS=F,MSGCLASS=X
//RUN EXEC PGM=DFSRRCOO,PARM='UPB,POSTCOMP',REGION=512K
//SYSPRINT DD SYSOUT=*
//STEPLIB DD DSN=IMSVS.RESLIB,DISP=SHR
//DFSRESLB DD DSN=IMSVS.RESLIB,DISP=SHR
//IMS DD DSN=B40.TEST.PSBLIB,DISP=SHR
// DD DSN=B40.TEST.DBDLIB,DISP=SHR
//IMSACB DD DSN=B40.TEST.ACBLIB,DISP=SHR
//SYSUT3 DD UNIT=SYSDA,SPACE=(80,(100,100))
//SYSUT4 DD UNIT=SYSDA,SPACE=(256,(100,100)),DCB=KEYLEN=8
//COMPCTL DD DSN=IMSVS.PROCLIB(DFSACBCP),DISP=SHR
//SYSIN DD *
BUILD PSB=ALL
/*

```

10. C08TCT(사용자 단말기 등록)

```

B40,90.321,09:43:56,LDH.TEST.LIBRARY(C08TCT)
//B40CITCT JOB CLASS=A,MSGCLASS=X,MSGLEVEL=(0,0)
//ASM EXEC CICAUPLK,OUTC=X,REGION=3072K
//ASSEM.SYSUT1 DD *
PRINT NOGEN
TCTCK DFHTCT TYPE=INITIAL,ACCMETH=VTAM, *
SUFFIX=CT
IT60 DFHTCT TYPE=TERMINAL,TRMIDNT=IT60,TRMTYPE=LUTYPE2, *
ACCMETH=VTAM,BRACKET=YES,CHNASSY=YES,DEFSCRN=(24,80), *
FEATURE=(PS,OUTLINE,SOSI,EXTDS),GMMSG=YES, *
RELREQ=(YES,YES),RUSIZE=256,TCTUAL=64,PGESIZE=(24,80), *
TIOAL=(512,4096),TRMMODL=2,TRMSTAT=TRANSCEIVE, *
NETNAME=IT60,BUFFER=1024,CONNECT=AUTO
*
IT28 DFHTCT TYPE=TERMINAL,TRMIDNT=IT28,TRMTYPE=LUTYPE2, *
ACCMETH=VTAM,BRACKET=YES,CHNASSY=YES,DEFSCRN=(24,80), *
FEATURE=(PS,OUTLINE,SOSI,EXTDS),GMMSG=YES, *
RELREQ=(YES,YES),RUSIZE=256,TCTUAL=64,PGESIZE=(24,80), *
TIOAL=(512,4096),TRMMODL=2,TRMSTAT=TRANSCEIVE, *
NETNAME=IT28,BUFFER=1024,CONNECT=AUTO
*
DFHTCT TYPE=FINAL
END
/*
//LNKEDT.SYSLMOD DD DSN=B40.TEST.LOADLIB,DISP=SHR
//

```

11. C09SNT(사용자 권한 정의)

```
B40,90.321,09:45:47,LDH.TEST.LIBRARY(C09SNT)
//B40CISNT JOB CLASS=A,MSGCLASS=X,MSGLEVEL=(0,0)
//STEP1 EXEC CICAUPLK,REGION=4096K
//ASSEM.SYSUT1 DD *
PRINT NOGEN
SIGNON DFHSNT TYPE=INITIAL,EXTSEC=YES
        DFHSNT TYPE=ENTRY,
        OPIDENT=B15,
        OPNAME='B15',
        OPPRTY=125,
        USERID=B15,
        TIMEOUT=1,
        RSLKEY=(1,2,3,4,5,6,7,8,9,
        10,11,12,13,14,15,16,17,18,19,
        20,21,22,23,24),
        SCTYKEY=(1,2,3,4,5,6,7,8,9,
        10,11,12,13,14,15,16,17,18,19,
        20,21,22,23,24,25,26,27,28,29,
        30,31,32,33,34,35,36,37,38,39,
        40,41,42,43,44,45,46,47,48,49,
        50,51,52,53,54,55,56,57,58,59,
        60,61,62,63,64)
        DFHSNT TYPE=(ENTRY,DEFAULT),
        RSLKEY=(1),
        SCTYKEY=(1)
DFHSNT TYPE=FINAL
END
/*
//LNKEDT.SYSLMOD DD DSN=B40.TEST.LOADLIB,DISP=SHR
//
```

*
*
*
*
*
X
X
X
X
X
X
X
X
X
X
X
*

12. D01DBD(D.B 구조 생성)

B40,90.321,09:49:50,LDH.TEST.LIBRARY(D01DBD)

//B40DMDBD JOB CLASS=A,MSGCLASS=X
// EXEC DBDGEN,MBR=KOSISDBD

```

//*****
//***** PROGRAMER NUMBER = B40 *****
//***** DBD NAME = KOSISDBD *****
//***** RANDOMIZING MODULE = DFSHDC30 *****
//***** ACCESS METHOD = HDAM *****
//***** FILE STRUCTURE = VSAM/ESDS *****
//***** ROOT ADDRESSBLE AREA = 600 *****
//***** LABEL = 8 *****
//***** CREATED 26, DEC. 1989 *****
//*****

```

```

//C.SYSIN DD *
DBD NAME=KOSISDBD,ACCESS=HDAM,RMNAME=(DFSHDC30,1,600,4077)
DATASET DD1=DBKOSIS,DEVICE=3380,SCAN=4
SEGM NAME=TOTAL,BYTES=35
FIELD NAME=(KEY0,SEQ,U),BYTES=5,START=1,TYPE=C
FIELD NAME=TITLE0,BYTES=30,START=6,TYPE=C
SEGM NAME=ATTRITO,BYTES=40,PARENT=((TOTAL))
SEGM NAME=SPECO,BYTES=30,PARENT=((TOTAL))
SEGM NAME=MO,BYTES=(1458),PARENT=((TOTAL))
SEGM NAME=QO,BYTES=(490),PARENT=((TOTAL))
SEGM NAME=YO,BYTES=(138),PARENT=((TOTAL))
SEGM NAME=FO,BYTES=10,PARENT=((TOTAL))
FIELD NAME=(FYEAR0,SEQ,U),BYTES=6,START=1,TYPE=C
SEGM NAME=LEVEL1,BYTES=39,PARENT=((TOTAL))
FIELD NAME=(KEY1,SEQ,U),BYTES=9,START=1,TYPE=C
SEGM NAME=ATTRIT1,BYTES=15,PARENT=((LEVEL1))
SEGM NAME=SPEC1,BYTES=30,PARENT=((LEVEL1))
SEGM NAME=M1,BYTES=(1458),PARENT=((LEVEL1))
SEGM NAME=Q1,BYTES=(490),PARENT=((LEVEL1))
SEGM NAME=Y1,BYTES=(138),PARENT=((LEVEL1))
SEGM NAME=F1,BYTES=10,PARENT=((LEVEL1))
FIELD NAME=(FYEAR1,SEQ,U),BYTES=6,START=1,TYPE=C
SEGM NAME=LEVEL2,BYTES=39,PARENT=((LEVEL1))
FIELD NAME=(KEY2,SEQ,U),BYTES=9,START=1,TYPE=C
SEGM NAME=ATTRIT2,BYTES=15,PARENT=((LEVEL2))
SEGM NAME=SPEC2,BYTES=30,PARENT=((LEVEL2))
SEGM NAME=M2,BYTES=(1458),PARENT=((LEVEL2))
SEGM NAME=Q2,BYTES=(490),PARENT=((LEVEL2))
SEGM NAME=Y2,BYTES=(138),PARENT=((LEVEL2))
SEGM NAME=F2,BYTES=10,PARENT=((LEVEL2))
FIELD NAME=(FYEAR2,SEQ,U),BYTES=6,START=1,TYPE=C
SEGM NAME=LEVEL3,BYTES=39,PARENT=((LEVEL2))
FIELD NAME=(KEY3,SEQ,U),BYTES=9,START=1,TYPE=C

```

```

SEGM      NAME=ATTRIT3, BYTES=15, PARENT=((LEVEL3))
SEGM      NAME=SPEC3, BYTES=30, PARENT=((LEVEL3))
SEGM      NAME=M3, BYTES=(1458), PARENT=((LEVEL3))
SEGM      NAME=Q3, BYTES=(490), PARENT=((LEVEL3))
SEGM      NAME=Y3, BYTES=(138), PARENT=((LEVEL3))
SEGM      NAME=F3, BYTES=10, PARENT=((LEVEL3))
FIELD     NAME=(FYEAR3, SEQ, U), BYTES=6, START=1, TYPE=C
SEGM      NAME=LEVEL4, BYTES=39, PARENT=((LEVEL3))
FIELD     NAME=(KEY4, SEQ, U), BYTES=9, START=1, TYPE=C
SEGM      NAME=ATTRIT4, BYTES=15, PARENT=((LEVEL4))
SEGM      NAME=SPEC4, BYTES=30, PARENT=((LEVEL4))
SEGM      NAME=M4, BYTES=(1458), PARENT=((LEVEL4))
SEGM      NAME=Q4, BYTES=(490), PARENT=((LEVEL4))
SEGM      NAME=Y4, BYTES=(138), PARENT=((LEVEL4))
SEGM      NAME=F4, BYTES=10, PARENT=((LEVEL4))
FIELD     NAME=(FYEAR4, SEQ, U), BYTES=6, START=1, TYPE=C
SEGM      NAME=LEVEL5, BYTES=39, PARENT=((LEVEL4))
FIELD     NAME=(KEY5, SEQ, U), BYTES=9, START=1, TYPE=C
SEGM      NAME=ATTRIT5, BYTES=15, PARENT=((LEVEL5))
SEGM      NAME=SPEC5, BYTES=30, PARENT=((LEVEL5))
SEGM      NAME=M5, BYTES=(1458), PARENT=((LEVEL5))
SEGM      NAME=Q5, BYTES=(490), PARENT=((LEVEL5))
SEGM      NAME=Y5, BYTES=(138), PARENT=((LEVEL5))
SEGM      NAME=F5, BYTES=10, PARENT=((LEVEL5))
FIELD     NAME=(FYEAR5, SEQ, U), BYTES=6, START=1, TYPE=C
SEGM      NAME=LEVEL6, BYTES=39, PARENT=((LEVEL5))
FIELD     NAME=(KEY6, SEQ, U), BYTES=9, START=1, TYPE=C
SEGM      NAME=ATTRIT6, BYTES=15, PARENT=((LEVEL6))
SEGM      NAME=SPEC6, BYTES=30, PARENT=((LEVEL6))
SEGM      NAME=M6, BYTES=(1458), PARENT=((LEVEL6))
SEGM      NAME=Q6, BYTES=(490), PARENT=((LEVEL6))
SEGM      NAME=Y6, BYTES=(138), PARENT=((LEVEL6))
SEGM      NAME=F6, BYTES=10, PARENT=((LEVEL6))
FIELD     NAME=(FYEAR6, SEQ, U), BYTES=6, START=1, TYPE=C
DBDGEN
FINISH
END

```

END

```

/*
//L.SYSLMOD DD DSN=B40.TEST.DBDLIB(KOSISDBD),DISP=SHR
//

```

13. D02PSB(D.B PSB 정의)

B40,90.321,09:50:21,LDH.TEST.LIBRARY(D02PSB)

//B40DMPSB JOB CLASS=B,MSGCLASS=X

//STEP1 EXEC PSBGEN,MBR=KOSIPSBL,SOUT='*'

//*****

//* PSBNAME = KOSIPSBL *****

//* CREATED 30 DEC. 1988 *****

//*****

//C.SYSIN DD *

PCB	TYPE=DB,KEYLEN=65,DBDNAME=KOSISDBD,PROCOPT=L
SENSEG	NAME=TOTAL,PARENT=0
SENSEG	NAME=ATTRITO,PARENT=TOTAL
SENSEG	NAME=SPECO,PARENT=TOTAL
SENSEG	NAME=MO,PARENT=TOTAL
SENSEG	NAME=QO,PARENT=TOTAL
SENSEG	NAME=YO,PARENT=TOTAL
SENSEG	NAME=FO,PARENT=TOTAL
SENSEG	NAME=LEVEL1,PARENT=TOTAL
SENSEG	NAME=ATTRIT1,PARENT=LEVEL1
SENSEG	NAME=SPEC1,PARENT=LEVEL1
SENSEG	NAME=M1,PARENT=LEVEL1
SENSEG	NAME=Q1,PARENT=LEVEL1
SENSEG	NAME=Y1,PARENT=LEVEL1
SENSEG	NAME=F1,PARENT=LEVEL1
SENSEG	NAME=LEVEL2,PARENT=LEVEL1
SENSEG	NAME=ATTRIT2,PARENT=LEVEL2
SENSEG	NAME=SPEC2,PARENT=LEVEL2
SENSEG	NAME=M2,PARENT=LEVEL2
SENSEG	NAME=Q2,PARENT=LEVEL2
SENSEG	NAME=Y2,PARENT=LEVEL2
SENSEG	NAME=F2,PARENT=LEVEL2
SENSEG	NAME=LEVEL3,PARENT=LEVEL2
SENSEG	NAME=ATTRIT3,PARENT=LEVEL3
SENSEG	NAME=SPEC3,PARENT=LEVEL3
SENSEG	NAME=M3,PARENT=LEVEL3
SENSEG	NAME=Q3,PARENT=LEVEL3
SENSEG	NAME=Y3,PARENT=LEVEL3
SENSEG	NAME=F3,PARENT=LEVEL3
SENSEG	NAME=LEVEL4,PARENT=LEVEL3
SENSEG	NAME=ATTRIT4,PARENT=LEVEL4
SENSEG	NAME=SPEC4,PARENT=LEVEL4
SENSEG	NAME=M4,PARENT=LEVEL4
SENSEG	NAME=Q4,PARENT=LEVEL4
SENSEG	NAME=Y4,PARENT=LEVEL4
SENSEG	NAME=F4,PARENT=LEVEL4
SENSEG	NAME=LEVEL5,PARENT=LEVEL4
SENSEG	NAME=ATTRIT5,PARENT=LEVEL5
SENSEG	NAME=SPEC5,PARENT=LEVEL5

```

SENSEG NAME=M5 , PARENT=LEVEL5
SENSEG NAME=Q5 , PARENT=LEVEL5
SENSEG NAME=Y5 , PARENT=LEVEL5
SENSEG NAME=F5 , PARENT=LEVEL5
SENSEG NAME=LEVEL6 , PARENT=LEVEL5
SENSEG NAME=ATTRIT6 , PARENT=LEVEL6
SENSEG NAME=SPEC6 , PARENT=LEVEL6
SENSEG NAME=M6 , PARENT=LEVEL6
SENSEG NAME=Q6 , PARENT=LEVEL6
SENSEG NAME=Y6 , PARENT=LEVEL6
SENSEG NAME=F6 , PARENT=LEVEL6
PSBGEN LANG=PL/I , PSBNAME=KOS IPSBL
END

```

```
/*
```

```
//L.SYSLMOD DD DSN=B40.TEST.PSBLIB(&MBR) , DISP=SHR
```

```
//STEP2 EXEC PSBGEN , MBR=KOS IPSBG , SOUT='*'

```

```
//*****

```

```
//** PSBNAME = KOS IPSBG *****

```

```
//*****

```

```
//C.SYSIN DD
```

```
PCB * TYPE=DB , KEYLEN=65 , DBDNAME=KOS ISDBD , PROCOPT=GO
```

```

SENSEG NAME=TOTAL , PARENT=0
SENSEG NAME=ATTRITO , PARENT=TOTAL
SENSEG NAME=SPECO , PARENT=TOTAL
SENSEG NAME=M0 , PARENT=TOTAL
SENSEG NAME=Q0 , PARENT=TOTAL
SENSEG NAME=Y0 , PARENT=TOTAL
SENSEG NAME=F0 , PARENT=TOTAL
SENSEG NAME=LEVEL1 , PARENT=TOTAL
SENSEG NAME=ATTRIT1 , PARENT=LEVEL1
SENSEG NAME=SPEC1 , PARENT=LEVEL1
SENSEG NAME=M1 , PARENT=LEVEL1
SENSEG NAME=Q1 , PARENT=LEVEL1
SENSEG NAME=Y1 , PARENT=LEVEL1
SENSEG NAME=F1 , PARENT=LEVEL1
SENSEG NAME=LEVEL2 , PARENT=LEVEL1
SENSEG NAME=ATTRIT2 , PARENT=LEVEL2
SENSEG NAME=SPEC2 , PARENT=LEVEL2
SENSEG NAME=M2 , PARENT=LEVEL2
SENSEG NAME=Q2 , PARENT=LEVEL2
SENSEG NAME=Y2 , PARENT=LEVEL2
SENSEG NAME=F2 , PARENT=LEVEL2
SENSEG NAME=LEVEL3 , PARENT=LEVEL2
SENSEG NAME=ATTRIT3 , PARENT=LEVEL3
SENSEG NAME=SPEC3 , PARENT=LEVEL3
SENSEG NAME=M3 , PARENT=LEVEL3
SENSEG NAME=Q3 , PARENT=LEVEL3
SENSEG NAME=Y3 , PARENT=LEVEL3
SENSEG NAME=F3 , PARENT=LEVEL3
SENSEG NAME=LEVEL4 , PARENT=LEVEL3

```

```

SENSEG      NAME=ATTRIT4,PARENT=LEVEL4
SENSEG      NAME=SPEC4,PARENT=LEVEL4
SENSEG      NAME=M4,PARENT=LEVEL4
SENSEG      NAME=Q4,PARENT=LEVEL4
SENSEG      NAME=Y4,PARENT=LEVEL4
SENSEG      NAME=F4,PARENT=LEVEL4
SENSEG      NAME=LEVEL5,PARENT=LEVEL4
SENSEG      NAME=ATTRIT5,PARENT=LEVEL5
SENSEG      NAME=SPEC5,PARENT=LEVEL5
SENSEG      NAME=M5,PARENT=LEVEL5
SENSEG      NAME=Q5,PARENT=LEVEL5
SENSEG      NAME=Y5,PARENT=LEVEL5
SENSEG      NAME=F5,PARENT=LEVEL5
SENSEG      NAME=LEVEL6,PARENT=LEVEL5
SENSEG      NAME=ATTRIT6,PARENT=LEVEL6
SENSEG      NAME=SPEC6,PARENT=LEVEL6
SENSEG      NAME=M6,PARENT=LEVEL6
SENSEG      NAME=Q6,PARENT=LEVEL6
SENSEG      NAME=Y6,PARENT=LEVEL6
SENSEG      NAME=F6,PARENT=LEVEL6
PSBGEN      LANG=PL/I,PSBNAME=KOSIPSBG
END

```

```
/*
```

```
//L.SYSLMOD DD DSN=B40.TEST.PSBLIB(&MBR),DISP=SHR
```

```
//STEP3 EXEC PSBGEN,MBR=KOSIPSBA,SOUT='*'

```

```
//*****

```

```
//** PSBNAME = KOSIPSBA *****

```

```
//*****

```

```
//C.SYSIN DD *
```

```

PCB TYPE=DB,KEYLEN=65,DBDNAME=KOSISDBD,PROCOPT=A
SENSEG NAME=TOTAL,PARENT=0
SENSEG NAME=ATTRITO,PARENT=TOTAL
SENSEG NAME=SPECO,PARENT=TOTAL
SENSEG NAME=MO,PARENT=TOTAL
SENSEG NAME=QO,PARENT=TOTAL
SENSEG NAME=YO,PARENT=TOTAL
SENSEG NAME=FO,PARENT=TOTAL
SENSEG NAME=LEVEL1,PARENT=TOTAL
SENSEG NAME=ATTRIT1,PARENT=LEVEL1
SENSEG NAME=SPEC1,PARENT=LEVEL1
SENSEG NAME=M1,PARENT=LEVEL1
SENSEG NAME=Q1,PARENT=LEVEL1
SENSEG NAME=Y1,PARENT=LEVEL1
SENSEG NAME=F1,PARENT=LEVEL1
SENSEG NAME=LEVEL2,PARENT=LEVEL1
SENSEG NAME=ATTRIT2,PARENT=LEVEL2
SENSEG NAME=SPEC2,PARENT=LEVEL2
SENSEG NAME=M2,PARENT=LEVEL2
SENSEG NAME=Q2,PARENT=LEVEL2
SENSEG NAME=Y2,PARENT=LEVEL2
SENSEG NAME=F2,PARENT=LEVEL2

```

```

SENSEG      NAME=LEVEL3,PARENT=LEVEL2
SENSEG      NAME=ATTRIT3,PARENT=LEVEL3
SENSEG      NAME=SPEC3,PARENT=LEVEL3
SENSEG      NAME=M3,PARENT=LEVEL3
SENSEG      NAME=Q3,PARENT=LEVEL3
SENSEG      NAME=Y3,PARENT=LEVEL3
SENSEG      NAME=F3,PARENT=LEVEL3
SENSEG      NAME=LEVEL4,PARENT=LEVEL3
SENSEG      NAME=ATTRIT4,PARENT=LEVEL4
SENSEG      NAME=SPEC4,PARENT=LEVEL4
SENSEG      NAME=M4,PARENT=LEVEL4
SENSEG      NAME=Q4,PARENT=LEVEL4
SENSEG      NAME=Y4,PARENT=LEVEL4
SENSEG      NAME=F4,PARENT=LEVEL4
SENSEG      NAME=LEVEL5,PARENT=LEVEL4
SENSEG      NAME=ATTRIT5,PARENT=LEVEL5
SENSEG      NAME=SPEC5,PARENT=LEVEL5
SENSEG      NAME=M5,PARENT=LEVEL5
SENSEG      NAME=Q5,PARENT=LEVEL5
SENSEG      NAME=Y5,PARENT=LEVEL5
SENSEG      NAME=F5,PARENT=LEVEL5
SENSEG      NAME=LEVEL6,PARENT=LEVEL5
SENSEG      NAME=ATTRIT6,PARENT=LEVEL6
SENSEG      NAME=SPEC6,PARENT=LEVEL6
SENSEG      NAME=M6,PARENT=LEVEL6
SENSEG      NAME=Q6,PARENT=LEVEL6
SENSEG      NAME=Y6,PARENT=LEVEL6
SENSEG      NAME=F6,PARENT=LEVEL6
PSBGEN      LANG=PL/I,PSBNAME=KOSIPSBA
END

/*
//L.SYSLMOD      DD  DSN=B40.TEST.PSBLIB(&MBR),DISP=SHR
//

```