บทที่ 4

โครงสร้างข้อมูลของโปรแกรมมินิ-ไมโคร ซีดีเอส/ไอซิส

โครงสร้างข้อมูลของแฟ้มข้อมูลตารางการแปลงพยัญชนะเป็นตัวพิมพ์ใหญ่

แฟ้มข้อมูลตารางการแปลงพยัญชนะเป็นตัวพิมพ์ใหญ่ (ISISUC.TAB หรือ ISIS Uppercase Conversion Table) ทำหน้าที่จัดเก็บรหัสแอสกีในรูปเลขฐานสิบ (Decimal ASCII Code) จำนวน 256 รหัส แต่ละค่าเป็นเลขจำนวนเต็ม 3 หลัก คั่นด้วยช่องว่าง 1 ช่อง แบ่งออกเป็น 16 บรรทัด ๆ ละ 16 รหัส ดังภาพ 4.1

```
000 001 002 003 004 005 006 007 008 009 010 011 012 013 014 015
016 017 018 019 020 021 022 023 024 025 026 027 028 028 030 031
032 033 034 035 036 037 038 039 040 041 042 043 044 045 046 047
048 049 050 051 052 053 053 055 056 057 058 059 060 061 062 063
064 065 066 067 068 069 070 071 072 073 074 075 076 077 078 079
080 081 082 083 084 085 086 087 088 089 090 091 092 093 094 095
096 065 066 067 068 069 070 071 072 073 074 075 076 077 078 079
080 081 082 083 084 085 086 087 088 089 090 123 124 125 126 127
067 085 069 065 065 065 065 067 069 069 069 073 073 073 065 065
069 069 069 079 079 079 085 085 089 079 085 155 156 157 158 159
065 073 079 085 078 078 166 167 168 169 170 171 172 173 174 175
176 177 178 179 180 181 182 183 184 185 186 187 188 189 190 191
192 193 194 195 196 197 198 199 200 201 202 203 204 205 206 207
208 209 210 211 212 213 214 215 216 217 218 219 220 221 222 223
224 225 226 227 228 229 230 231 232 233 234 235 236 237 238 239
240 241 242 243 244 245 246 247 248 249 250 251 252 253 254 255
```

ภาพ 4.1 แสดงข้อมูลของแฟ้มข้อมูลตารางการแปลงพยัญชนะเป็นตัวใหญ่

จากภาพ 4.1 จะเห็นได้ว่า รหัสทั้ง 256 ค่า มีค่าอยู่ระหว่าง 000 - 255 เท่านั้น และมีบางรหัสมีค่าซ้ำกัน เช่น ในแถวที่ 7 ลำดับที่ 1 รหัสมีค่าเป็น 096 แต่ในลำดับที่ 2 กลับมีค่าเป็น 065 แทนที่จะเป็น 097 และในลำดับที่ 3 กลับมีค่าเป็น 066 แทนที่จะเป็น 098 เป็นต้น จะเห็นได้ว่า ผู้ติดตั้งระบบเป็นผู้กำหนดค่าให้กับลำดับที่ต่าง ๆ ของรหัสแอสกีทั้ง 256 รหัส เพื่อใช้ในการแปลงรหัสที่ผู้ใช้ระบบป้อนเข้าไปให้เป็นรหัสที่ต้องการ โดยเฉพาะอย่างยิ่ง การแปลงตัวพยัญชนะตัวพิมพ์เล็ก ให้กลายเป็นพยัญชนะตัวพิมพ์ใหญ่ รายละเอียดของการแปลงรหัสตามแฟ้มข้อมูลดังกล่าวเมื่อเทียบกับตารางรหัสแอสกีจะเป็นดังตาราง 4.1

ลำดับที่ของข้อมูลในแฟ้มข้อมูล	รหัสแอสกีของลำดับที่ลบหนึ่ง	สัญญลักษณ์	รหัสในแฟ้มข้อมูล	สัญญลักษณ์
1	00	NUL	000	NUL
2	01	SOH	001	SOH
3	02	STX	002	STX
4	03	ETX	003	ETX
5	04	EOT	004	EOT
6	05	ENQ	005	ENQ
7	06	ACK	006	ACK
8	07	BEL	007	BEL
9	08	BS	008	BS
10	09	HT	009	HT
11	10	LF	010	LF
12	11	VT	011	VT
13	12	FF	012	FF
14	13	CR	013	CR
15	14	SO	014	SO
16	15	SI	015	SI

ตาราง 4.1 เปรียบเทียบรหัสแอสกีกับรหัสในแฟ้มข้อมูล ISISUC.TAB

ลำดับที่ของข้อมูลในแฟ้มข้อมูล	รหัสแอสกีของลำดับที่ลบหนึ่ง	สัญญลักษณ์	รหัสในแฟ้มข้อมูล	สัญญลักษณ์
17	16	DLE	016	DLE
18	17	DC1	017	DC1
19	18	DC2	018	DC2
20	19	DC3	019	DC3
21	20	DC4	020	DC4
22	21	NAK	021	NAK
23	22	SYN	022	SYN
24	23	ETB	023	ETB
25	24	CAN	024	CAN
26	25	EM	025	EM
27	26	SUB	026	SUB
28	27	ESC	027	ESC
29	28	FS	028	FS
30	29	GS	028	FS
31	30	RS	030	RS
32	31	US	031	US
33	32	SP	032	SP
34	33	!	033	!
35	34	"	034	"
36	35	#	035	#
37	36	$	036	$
38	37	%	037	%
39	38	&	038	&
40	39	'	039	'
41	40	(	040	(

ตาราง 4.1 เปรียบเทียบรหัสแอสกีกับรหัสในแฟ้มข้อมูล ISISUC.TAB (ต่อ)

ลำดับที่ของข้อมูลในแฟ้มข้อมูล	รหัสแอสกีของลำดับที่ลบหนึ่ง	สัญญลักษณ์	รหัสในแฟ้มข้อมูล	สัญญลักษณ์
42	41	)	041	)
43	42	*	042	*
44	43	+	043	+
45	44	,	044	,
46	45	-	045	-
47	46	.	046	.
48	47	/	047	/
49	48	0	048	0
50	49	1	049	1
51	50	2	050	2
52	51	3	051	3
53	52	4	052	4
54	53	5	053	5
55	54	6	054	6
56	55	7	055	7
57	56	8	056	8
58	57	9	057	9
59	58	:	058	:
60	59	;	059	;
61	60	<	060	<
62	61	=	061	=
63	62	>	062	>
64	63	?	063	?
65	64	@	064	@
66	65	A	065	A

ตาราง 4.1 เปรียบเทียบรหัสแอสกีกับรหัสในแฟ้มข้อมูล ISISUC.TAB (ต่อ)

ลำดับที่ของข้อมูลในแฟ้มข้อมูล	รหัสแอสกีของลำดับที่ลบหนึ่ง	สัญญลักษณ์	รหัสในแฟ้มข้อมูล	สัญญลักษณ์
67	66	B	066	B
68	67	C	067	C
69	68	D	068	D
70	69	E	069	E
71	70	F	070	F
72	71	G	071	G
73	72	H	072	H
74	73	I	073	I
75	74	J	074	J
76	75	K	075	K
77	76	L	076	L
78	77	M	077	M
79	78	N	078	N
80	79	O	079	O
81	80	P	080	P
82	81	Q	081	Q
83	82	R	082	R
84	83	S	083	S
85	84	T	084	T
86	85	U	085	U
87	86	V	086	V
88	87	W	087	W
89	88	X	088	X
90	89	Y	089	Y
91	90	Z	090	Z

ตาราง 4.1 เปรียบเทียบรหัสแอสกีกับรหัสในแฟ้มข้อมูล ISISUC.TAB (ต่อ)

ลำดับที่ของข้อมูลในแฟ้มข้อมูล	รหัสแอสกีของลำดับที่ลบหนึ่ง	สัญญลักษณ์	รหัสในแฟ้มข้อมูล	สัญญลักษณ์
92	91	[	091	[
93	92	\	092	\
94	93	]	093	]
95	94	↑	094	↑
96	95	–	095	–
97	96	`	096	`
98	97	a	065	A
99	98	b	066	B
100	99	c	067	C
101	100	d	068	D
102	101	e	069	E
103	102	f	070	F
104	103	g	071	G
105	104	h	072	H
106	105	i	073	I
107	106	j	074	J
108	107	k	075	K
109	108	l	076	L
110	109	m	077	M
111	110	n	078	N
112	111	o	079	O
113	112	p	080	P
114	113	q	081	Q
115	114	r	082	R
116	115	s	083	S

ตาราง 4.1 เปรียบเทียบรหัสแอสกีกับรหัสในแฟ้มข้อมูล ISISUC.TAB (ต่อ)

ลำดับที่ของข้อมูลในแฟ้มข้อมูล	รหัสแอสกีของลำดับที่ลบหนึ่ง	สัญญลักษณ์	รหัสในแฟ้มข้อมูล	สัญญลักษณ์
117	116	t	084	T
118	117	u	085	U
119	118	v	086	V
120	119	w	087	W
121	120	x	088	X
122	121	y	089	Y
123	122	z	090	Z
124	123	{	123	{
125	124	|	124	|
126	125	}	125	}
127	126	~	126	~
128	127	DEL	127	DEL
129	128	ç	067	C
130	129	ü	085	U
131	130	é	069	E
132	131	â	065	A
133	132	ä	065	A
134	133	à	065	A
135	134	å	065	A
136	135	ç	067	C
137	136	ê	069	E
138	137	ë	069	E
139	138	è	069	E
140	139	ï	073	I
141	140	î	073	I

ตาราง 4.1 เปรียบเทียบรหัสแอสกีกับรหัสในแฟ้มข้อมูล ISISUC.TAB (ต่อ)

ลำดับที่ของข้อมูลในแฟ้มข้อมูล	รหัสแอสกีของลำดับที่ลบหนึ่ง	สัญญลักษณ์	รหัสในแฟ้มข้อมูล	สัญญลักษณ์
142	141	Ì	073	I
143	142	Ä	065	A
144	143	Å	065	A
145	144	É	069	E
146	145	æ	069	E
147	146	Æ	069	E
148	147	ô	079	O
149	148	ö	079	O
150	149	ò	079	O
151	150	û	085	U
152	151	ù	085	U
153	152	ÿ	089	Y
154	153	Ö	079	O
155	154	Ü	085	U
156	155	¢	155	¢
157	156	£	156	£
158	157	¥	157	¥
159	158	Pt	158	Pt
160	159	ƒ	159	ƒ
161	160	á	065	A
162	161	í	073	I
163	162	ó	079	O
164	163	ú	085	U
165	164	ñ	078	N
166	165	Ñ	078	N

ตาราง 4.1 เปรียบเทียบรหัสแอสกีกับรหัสในแฟ้มข้อมูล ISISUC.TAB (ต่อ)

ลำดับที่ของข้อมูลในแฟ้มข้อมูล	รหัสแอสกีของลำดับที่ลบหนึ่ง	สัญญลักษณ์	รหัสในแฟ้มข้อมูล	สัญญลักษณ์
167	166	a	166	a
168	167	o	167	o
169	168	¿	168	¿
170	169	⌐	169	⌐
171	170	¬	170	¬
172	171	1/2	171	1/2
173	172	1/4	172	1/4
174	173	¡	173	¡
175	174	<<	174	<<
176	175	>>	175	>>
177	176	▒	176	▒
178	177	▓	177	▓
179	178	▓	178	▓
180	179	│	179	│
181	180	┤	180	┤
182	181	╡	181	╡
183	182	╢	182	╢
184	183	╖	183	╖
185	184	╕	184	╕
186	185	╣	185	╣
187	186	║	186	║
188	187	╗	187	╗
189	188	╝	188	╝
190	189	╜	189	╜
191	190	╛	190	╛

ตาราง 4.1 เปรียบเทียบรหัสแอสกีกับรหัสในแฟ้มข้อมูล ISISUC.TAB (ต่อ)

ลำดับที่ของข้อมูลในแฟ้มข้อมูล	รหัสแอสกีของลำดับที่ลบหนึ่ง	สัญญลักษณ์	รหัสในแฟ้มข้อมูล	สัญญลักษณ์
192	191		191	
193	192		192	
194	193		193	
195	194		194	
196	195		195	
197	196		196	
198	197		197	
199	198		198	
200	199		199	
201	200		200	
202	201		201	
203	202		202	
204	203		203	
205	204		204	
206	205		205	
207	206		206	
208	207		207	
209	208		208	
210	209		209	
211	210		210	
212	211		211	
213	212		212	
214	213		213	
215	214		214	
216	215		215	

ตาราง 4.1 เปรียบเทียบรหัสแอสกีกับรหัสในแฟ้มข้อมูล ISISUC.TAB (ต่อ)

ลำดับที่ของข้อมูลในแฟ้มข้อมูล	รหัสแอสกีของลำดับที่ลบหนึ่ง	สัญญลักษณ์	รหัสในแฟ้มข้อมูล	สัญญลักษณ์
217	216	≠	216	≠
218	217	⌋	217	⌋
219	218	⌈	218	⌈
220	219	▪	219	▪
221	220	▬	220	▬
222	221	▌	221	▌
223	222	▐	222	▐
224	223	▄	223	▄
225	224	α	224	α
226	225	β	225	β
227	226	Γ	226	Γ
228	227	π	227	π
229	228	Σ	228	Σ
230	229	σ	229	σ
231	230	μ	230	μ
232	231	τ	231	τ
233	232	Φ	232	Φ
234	233	θ	233	θ
235	234	Ω	234	Ω
236	235	δ	235	δ
237	236	∞	236	∞
238	237	ϕ	237	ϕ
239	238	$\in$	238	$\in$
240	239	$\cap$	239	$\cap$
241	240	$\equiv$	240	$\equiv$

ตาราง 4.1 เปรียบเทียบรหัสแอสกีกับรหัสในแฟ้มข้อมูล ISISUC.TAB (ต่อ)

ลำดับที่ของข้อมูลในแฟ้มข้อมูล	รหัสแอสกีของลำดับที่ลบหนึ่ง	สัญญลักษณ์	รหัสในแฟ้มข้อมูล	สัญญลักษณ์
242	241	±	241	±
243	242	⩾	242	⩾
244	243	⩽	243	⩽
245	244	∫	244	∫
246	245	∫	245	∫
247	246	÷	246	÷
248	247	≈	247	≈
249	248	°	248	°
250	249	•	249	•
251	250	●	250	●
252	251	√	251	√
253	252	n	252	n
254	253	z	253	z
255	254	■	254	■
256	255		255	

ตาราง 4.1 เปรียบเทียบรหัสแอสกีกับรหัสในแฟ้มข้อมูล ISISUC.TAB (ต่อ)

หมายเหตุ สัญญลักษณ์คำย่อในตาราง หมายถึง

NUL - Null
SOH - Start of Heading
STX - Start of Text
ETX - End of Text
EOT - End of Transmission
ENQ - Enquiry
ACK - Acknowledge
DLE - Data Link Escape
DC - Device Control
NAK - Negative Acknowledge
SYN - Synchronous Idle
ETB - End of Transmission Bloc
CAN - Cancel
EM - End of Medium

BEL - Bell	SUB - Substitute
BS - Backspace	ESC - Escape
HT - Horizontal Tabulation	FS - File Separator
LF - Line Feed	GS - Group Separator
VT - Vertical Tabulation	RS - Record Separator
FF - Form Feed	US - Unit Separator
CR - Carriage Return	SP - Space (Blank)
SO - Shift Out	DEL - Delete
SI - Shift In	

ลักษณะการใช้งานแฟ้มข้อมูลตารางการแปลงพยัญชนะเป็นตัวพิมพ์ใหญ่

จากตาราง 4.1 รหัสที่ถูกแปลงค่ามีทั้งสิ้น 60 รหัส ได้แก่ รหัสลำดับที่ 30, 99 - 124, 130 - 156 และ 162 - 167 นั่นคือ พวกพยัญชนะตัวพิมพ์เล็ก (a, b, c, ... , z) จะถูกเปลี่ยนให้เป็นพยัญชนะตัวพิมพ์ใหญ่ (A, B, C, ... ,Z) นอกจากนี้แล้ว จะพบว่าพวกพยัญชนะที่มีสัญญลักษณ์เน้นเสียง (Accented Character) ต่าง ๆ ก็จะถูกเปลี่ยนให้กลายเป็นพยัญชนะตัวพิมพ์ใหญ่ธรรมดาด้วย เช่น รหัสลำดับที่ 130 ซึ่งค่าแอสกีควรจะเป็น 129 (อักขระ ü) ก็จะถูกเปลี่ยนเป็นรหัส 085 (อักขระ U) เป็นต้น เทคนิคการแปลงรหัสก็คือ ระบบจะอ่านข้อมูลทั้ง 256 ค่านี้เก็บไว้ในหน่วยความจำในลักษณะที่เรียงลำดับผลลัพธ์การแปลงรหัสตามรหัสแอสกี นั่นคือ ค่ารหัสแอสกีที่แปลงแล้วของรหัสแรก (000_{10}) ก็จะอยู่ที่หน่วยความจำไบท์แรกของตาราง ในกรณีนี้เป็นค่าเดิมคือ 000 แสดงว่ารหัสไม่ถูกเปลี่ยน ค่ารหัสแอสกีใหม่ของรหัสแอสกีลำดับที่เก้าสิบแปด (นั่นคือ อักขระ a ซึ่งมีค่าแอสกีเป็น 097_{10}) ก็จะอยู่ที่หน่วยความจำไบท์ที่เก้าสิบแปดเมื่อนับจากต้นตาราง ซึ่งจะมีค่าที่เก็บอยู่เป็น 065_{10} (นั่นคือ อักขระ A) เป็นต้น แต่ในหน่วยความจำนั้น ไม่ได้เก็บค่าแต่ละรหัสด้วยระบบฐานสิบเหมือนกับที่ปรากฏในแฟ้มข้อมูลตารางการแปลงพยัญชนะเป็นตัวพิมพ์ใหญ่ แต่จะเปลี่ยนรหัสซึ่งเป็นเลขฐานสิบจำนวนสามหลักนี้ให้กลายเป็นเลขฐานสิบหกจำนวน 1 ไบท์แทน ดังนั้นตารางนี้ ใช้เนื้อที่ในหน่วยความจำเพียง 256 ไบท์เท่านั้น

ในการใช้งานโปรแกรมมินิ-ไมโคร ซีดีเอส/ไอซิส ทั้ง 6 โปรแกรม คือ

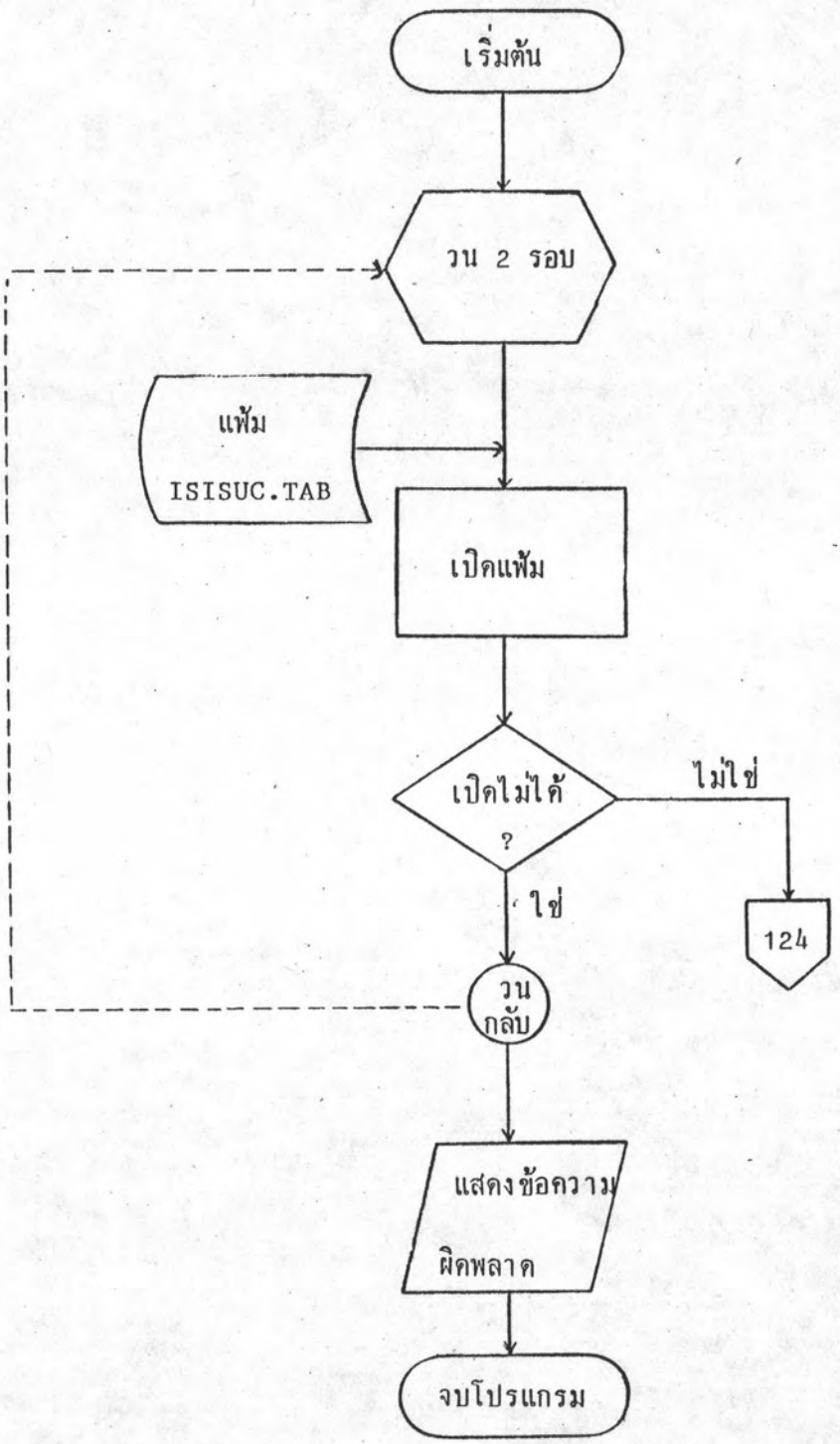
ISIS.EXE, ISISPRT.EXE, ISISDEF.EXE, ISISINV.EXE, ISISUTL.EXE และ ISISXCH.EXE นั้น เมื่อผู้ใช้เรียกให้โปรแกรมใดโปรแกรมหนึ่งทำงาน โปรแกรมจะทำการเปิดแฟ้มข้อมูลตารางการแปลงพยัญชนะเป็นตัวพิมพ์ใหญ่ เป็นแฟ้มข้อมูลแรกเสมอ อ่านรหัสทั้ง 256 รหัสเก็บในหน่วยความจำ แล้วปิดแฟ้มข้อมูล แฟ้มข้อมูลนี้จะถูกอ่านในช่วงแรกของการประมวลผลเพียงครั้งเดียวเท่านั้นของทุกโปรแกรม และผู้ใช้ไม่สามารถบรรณาธิกรแฟ้มข้อมูลนี้ได้โดยตรงจากรายการประมวลผลในโปรแกรม ถ้าต้องการบรรณาธิกรแฟ้มข้อมูลนี้ จะต้องใช้โปรแกรมบรรณาธิกรต่างหาก เช่น โปรแกรม EDLIN.COM ของระบบปฏิบัติการ หรือใช้โปรแกรมประมวลผลคำ (Word Processor) ทั่ว ๆ ไป เช่น โปรแกรม WORDSTAR, โปรแกรม TURBO PASCAL EDITOR, โปรแกรม PROFESSIONAL EDITOR ฯลฯ ซึ่งการเปลี่ยนแปลงแก้ไขทำได้ง่ายเนื่องจากแฟ้มข้อมูลดังกล่าวมีลักษณะเป็นแฟ้มข้อมูลแอสกี (ASCII File) ซึ่งข้อมูลเป็นตัวเลขจำนวนเต็ม สามหลัก คั่นด้วยช่องว่างอย่างน้อยหนึ่งช่อง แต่ละบรรทัดจะใส่รหัสกี่ค่าก็ได้ แต่ให้จบแต่ละบรรทัดด้วยการกดปุ่ม <ENTER>

รายละเอียดขั้นตอนการอ่านข้อมูลจากแฟ้มข้อมูลเก็บในหน่วยความจำ สามารถเขียนเป็นผังงานได้ดังในผังงานที่ 4.1

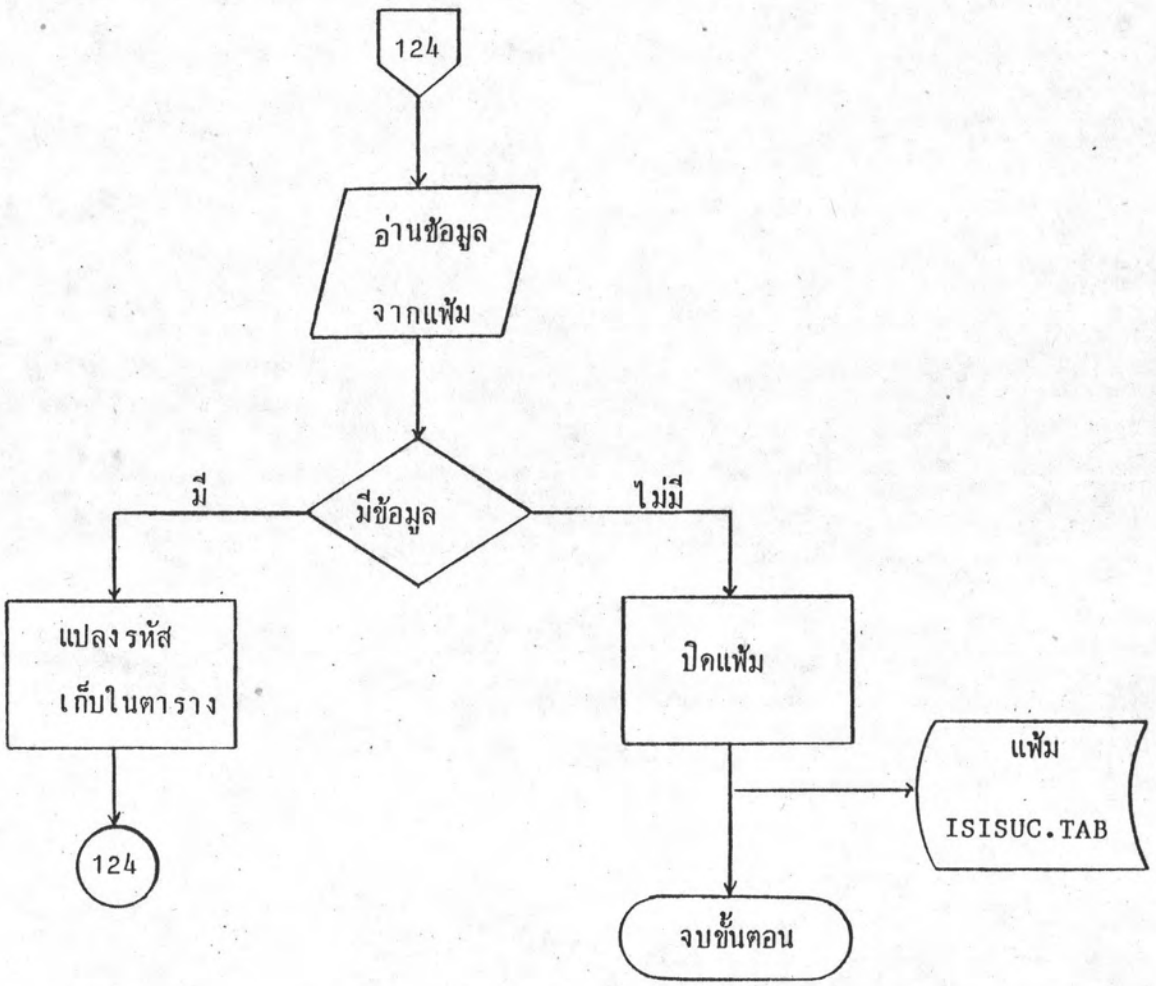
1. เมื่อเริ่มประมวลผล ระบบจะเปิดแฟ้มข้อมูล ISISUC.TAB ถ้าครั้งแรกเปิดไม่ได้ จะลองเปิดครั้งที่สอง ถ้าเปิดแฟ้มข้อมูลดังกล่าวไม่ได้ จะแสดงข้อความผิดพลาดบนจอภาพ แล้วจบการทำงาน ตัวอย่างเช่น

```
C>ISISPRT    <ENTER>
? Error:File not found error in file ISISUC.TAB
  Error Code 1032, Status 0002
PC = 0004: 2CB8; SS = 305F, FP = 0921, SP = C7E0
C>
```

ภาพ 4.2 จอภาพแสดงผลลัพธ์จากการประมวลผลโปรแกรมเมื่อไม่มีแฟ้มข้อมูล ISISUC.TAB



ผังงานที่ 4.1 ผังแสดงขั้นตอนการประมวลผลแฟ้มข้อมูลตารางการแปลงพยัญชนะเป็นตัวพิมพ์ใหญ่



ผังงานที่ 4.1 ผังแสดงขั้นตอนการประมวลผลแฟ้มข้อมูลตารางการแปลงพยัญชนะเป็นตัวพิมพ์ใหญ่ (ต่อ)

2. ถ้าเปิดแฟ้มข้อมูล ISISUC.TAB สำเร็จ ระบบจะเริ่มอ่านข้อมูลมาใส่หน่วยความจำ โดยอ่านมาครั้งละ 512 ไบท์ ดังตัวอย่างในภาพ 4.3 เป็นภาพแสดงหน่วยความจำหลังจากที่โปรแกรม ISISPRT.EXE อ่านข้อมูลจากแฟ้มข้อมูล ISISUC.TAB ครั้งแรก ค่าในรีจิสเตอร์แสดงว่า แฟ้มข้อมูล ISISUC.TAB ถูกกำหนดโดยระบบให้มีค่าแฮนเดิลหมายเลข 5 อ่านข้อมูลจำนวน 512 ไบท์ (200_{16} ไบท์) เก็บไว้ในหน่วยความจำหลักโดยเริ่มตั้งแต่หน่วยความจำ DS:DX คือ 305F:E0C2 จนถึง 305F:E2C1 โดยเก็บเรียงลำดับไปเรื่อย ๆ ในลักษณะเดียวกับที่เก็บอยู่ในแฟ้มข้อมูลจริง ๆ แต่เนื่องจากแฟ้มข้อมูล ISISUC.TAB มีขนาด 1041 ไบท์ ระบบจึงอ่านเข้ามาไม่หมด ในครั้งเดียว

```
*** TRACE ***
 STACK                  DATA1
   305F:C7BC  2CB8        305F:E0C2  30 30 30 20 30 30 31 20      000 001
   305F:C7BA  02E4        305F:E0CA  30 30 32 20 30 30 33 20      002 003
   305F:C7B8  C7C6        305F:E0D2  30 30 34 20 30 30 35 20      004 005
   305F:C7B6  0000        305F:E0DA  30 30 36 20 30 30 37 20      006 007
   305F:C7B4  0005        305F:E0E2  30 30 38 20 30 30 39 20      008 009
   305F:C7B2  1499        305F:E0EA  30 31 30 20 30 31 31 20      010 011
   305F:C7B0  E08E        305F:E0F2  30 31 32 20 30 31 33 20      012 013
   305F:C7AE  E08E        305F:E0FA  30 31 34 20 30 31 35 0D      014 015
   305F:C7AC  05CB        305F:E102  0A 30 31 36 20 30 31 37      016 017
   305F:C7AA  C7B8        305F:E10A  20 30 31 38 20 30 31 39       018 019
  305F:C7A8  E08E         305F:E112  20 30 32 30 20 30 32 31      020 021
 CPU                      305F:E11A  20 30 32 32 20 30 32 33       022 023
  AX=0200  BX=0005        305F:E122  20 30 32 34 20 30 32 35       024 025
  CX=0200  DX=E0C2        305F:E12A  20 30 32 36 20 30 32 37       026 027
  SP=C7A8  BP=C7AA        305F:E132  20 30 32 38 20 30 32 38       028 028
  SI=2ECA  DI=04D2       FILE
  CS=3050  DS=305F
  SS=305F  ES=305F       PROGRAM1
  IP=008B  T:1            3050:008B  72F1          JB      007E
  A:1 C:0 P:0 S:1        BREAKPOINTS
  Z:0 D:0 I:1 O:0         00:  2E89:1C9D 0001 STOP
```

ภาพ 4.3 แสดงหน่วยความจำที่เก็บข้อมูลจากการอ่านแฟ้มข้อมูล ISISUC.TAB ครั้งแรก

3. จากนั้น ระบบจะเริ่มอ่านข้อมูลที่เป็นรหัสในรูปเลขฐานสิบจำนวน 3 หลักทีละค่าไปแปลงเป็นรหัสแอสกีในรูปเลขฐานสิบหก 1 ไบท์เก็บไว้ในหน่วยความจำอีกที่หนึ่งซึ่งกำหนดไว้เป็นที่สำหรับมาค้นหาเพื่อแปลงรหัสในภายหลัง โดยเรียงลำดับไปเรื่อย ๆ ทีละไบท์ ดังตัวอย่างในตารางที่ 4.2

ข้อมูลในแฟ้มข้อมูล (3 ไบท์)	ข้อมูลที่แปลงเป็นรหัสแอสกีฐานสิบหก (1 ไบท์)
000	00
001	01
002	02
.	.
.	.
010	0A
011	0B
.	.
.	.
048	30
049	31
.	.
.	.
065	41
066	42
.	.
.	.
255	FF

ตาราง 4.2 แสดงการแปลงข้อมูลจากแฟ้มข้อมูล ISISUC.TAB ในรูปเลขฐานสิบหก

ภาพ 4.4 แสดงข้อมูลในตารางแปลงพยัญชนะเป็นตัวพิมพ์ใหญ่ในหน่วยความจำหลัก โดยเริ่มข้อมูลไบท์แรกที่ DS:DX คือ 305F:E30C เรียงลำดับไปเรื่อย ๆ รหัสละ 1 ไบท์ สำหรับการอ่านครั้งแรกนี้ ระบบจะจัดเก็บรหัสที่ใช้แปลงได้ถึงตำแหน่ง 305F:E389

```
STACK                    DATA1
  305F:C786  2CB8          305F:E30C  00 01 02 03 04 05 06 07
  305F:C784  02E4          305F:E314  08 09 0A 0B 0C 0D 0E 0F
  305F:C782  C790          305F:E31C  10 11 12 13 14 15 16 17
  305F:C780  0000          305F:E324  18 19 1A 1B 1C 1C 1E 1F
  305F:C77E  C784          305F:E32C  20 21 22 23 24 25 26 27     !"#$%&'
  305F:C77C  C731          305F:E334  28 29 2A 2B 2C 2D 2E 2F     ()*+,-./
  305F:C77A  E08E          305F:E33C  30 31 32 33 34 35 36 37     01234567
  305F:C778  E08E          305F:E344  38 39 3A 3B 3C 3D 3E 3F     89:;<=>?
  305F:C776  05CB          305F:E34C  40 41 42 43 44 45 46 47     @ABCDEFG
  305F:C774  C782          305F:E354  48 49 4A 4B 4C 4D 4E 4F     HIJKLMNO
 305F:C772  E08E          305F:E35C  50 51 52 53 54 55 56 57    PQRSTUVW
CPU                        305F:E364  58 59 5A 5B 5C 5D 5E 5F     XYZ[\]^_
 AX=0200  BX=0005          305F:E36C  60 41 42 43 44 45 46 47     `ABCDEFG
 CX=0200  DX=E0C2          305F:E374  48 49 4A 4B 4C 4D 4E 4F     HIJKLMNO
 SP=C772  BP=C774          305F:E37C  50 51 52 53 54 55 56 57     PQRSTUVW
 SI=2ECA  DI=04D2        FILE
 CS=3050  DS=305F
 SS=305F  ES=305F        PROGRAM1
 IP=004D  T:0             3050:004D  B43F            MOV     AH,3F
 A:1 C:0 P:0 S:1         BREAKPOINTS
 Z:0 D:0 I:1 O:0          00:   2E89:1C9D 0001 STOP
```

ภาพ 4.4 แสดงหน่วยความจำส่วนที่เป็นตารางการแปลงพยัญชนะเป็นตัวพิมพ์ใหญ่

4. เมื่อระบบอ่านและจัดเก็บข้อมูลในการอ่านครั้งแรกเสร็จแล้ว ระบบจะอ่านข้อมูลจากแฟ้มข้อมูล ISISUC.TAB อีกเป็นครั้งที่สอง จำนวน 512 ไบท์ โดยอ่านมาเก็บไว้ที่หน่วยความจำที่เดิมเดียวกับที่อ่านมาในครั้งแรก ซึ่งในตัวอย่างนี้คือ 305F:E0C2 จนถึง 305F:E2C1 ดังภาพ 4.5 แล้วแปลงให้อยู่ในรูปรหัสแอสกีในรูปเลขฐานสิบหก แล้วจัดเก็บไว้ในตารางแปลงพยัญชนะเป็นตัวพิมพ์ใหญ่ โดยเก็บข้อมูลไว้ที่หน่วยความจำที่อยู่ถัดจากที่เก็บไว้เดิม นั่นคือที่ DS:DX 305F:E38A วนทำเช่นนี้เรื่อย ๆ จนหมดข้อมูล ข้อมูลในตารางแปลงพยัญชนะเป็นตัวพิมพ์ใหญ่หลังจากการอ่านข้อมูลครั้งที่สองเข้ามาแล้วจะเป็นดังภาพ 4.6

5. ระบบจะอ่านข้อมูลจากแฟ้มข้อมูล ISISUC.TAB อีกเป็นครั้งที่สาม ได้ข้อมูลอีก 17 ไบท์ ดังในภาพ 4.7 แล้วแปลงให้อยู่ในรูปรหัสแอสกีในรูปเลขฐานสิบหก รหัสละ 1 ไบท์ได้อีก 4 รหัสสุดท้าย บันทึกต่อท้ายตารางเดิม คือ 305F:E408 เมื่อเสร็จแล้วจะมีข้อมูลในตารางแปลงพยัญชนะเป็นตัวพิมพ์ใหญ่ครบทั้ง 256 ไบท์ ตั้งแต่ 305F:E30C จนถึง 305F:E40B ดังภาพ 4.8

```
STACK                    DATA1
  305F:C788  2CB8          305F:E0C2  32 36 20 31 32 37 0D 0A      26 127
  305F:C784  02E4          305F:E0CA  30 36 37 20 30 38 35 20      067 085
  305F:C782  C790          305F:E0D2  30 36 39 20 30 36 35 20      069 065
  305F:C780  0000          305F:E0DA  30 36 35 20 30 36 35 20      065 065
  305F:C77E  C784          305F:E0E2  30 36 35 20 30 36 37 20      065 067
  305F:C77C  C731          305F:E0EA  30 36 39 20 30 36 39 20      069 069
  305F:C77A  E08E          305F:E0F2  30 36 39 20 30 37 33 20      069 073
  305F:C778  E08E          305F:E0FA  30 37 33 20 30 37 33 20      073 073
  305F:C776  05CB          305F:E102  30 36 35 20 30 36 35 0D      065 065
  305F:C774  C782          305F:E10A  0A 30 36 39 20 30 36 89      069 069
 305F:C772  E08E           305F:E112  20 30 36 39 20 30 37 39      069 079
CPU                        305F:E11A  20 30 37 39 20 30 37 39       079 079
 AX=0200  BX=0005          305F:E122  20 30 38 35 20 30 38 35       085 085
 CX=0200  DX=E0C2          305F:E12A  20 30 38 39 20 30 37 39       089 079
 SP=C772  BP=C774          305F:E132  20 30 38 35 20 31 35 35       085 155
 SI=2ECA  DI=04D2         FILE
 CS=3050  DS=305F
 SS=305F  ES=305F        PROGRAM1
 IP=008B  T:1             3050:008B  72F1            JB      007E
 A:1 C:0 P:0 S:1         BREAKPOINTS
 Z:0 D:0 I:1 O:0          00:  2E89:1C9D 0001 STOP
```

ภาพ 4.5 แสดงหน่วยความจำที่เก็บข้อมูลจากการอ่านแฟ้มข้อมูล ISISUC.TAB ครั้งที่สอง

```
STACK                    DATA1
  305F:C786  2CB8          305F:E38A  7E 7F 43 55 45 41 41 41      ~CUEAAA
  305F:C784  02E4          305F:E392  41 43 45 45 45 49 49 49      ACEEEIII
  305F:C782  C790          305F:E39A  41 41 45 45 45 4F 4F 4F      AAEEEOOO
  305F:C780  0000          305F:E3A2  55 55 59 4F 55 9B 9C 9D      UUYOU
  305F:C77E  C784          305F:E3AA  9E 9F 41 49 4F 55 4E 4E      AIOUNN
  305F:C77C  C732          305F:E3B2  A6 A7 A8 A9 AA AB AC AD
  305F:C77A  E08E          305F:E3BA  AE AF B0 B1 B2 B3 B4 B5
  305F:C778  E08E          305F:E3C2  B6 B7 B8 B9 BA BB BC BD
  305F:C776  05CB          305F:E3CA  BE BF C0 C1 C2 C3 C4 C5
  305F:C774  C782          305F:E3D2  C6 C7 C8 C9 CA CB CC CD
 305F:C772  E08E           305F:E3DA  CE CF D0 D1 D2 D3 D4 D5
CPU                        305F:E3E2  D6 D7 D8 D9 DA DB DC DD
 AX=0200  BX=0005          305F:E3EA  DE DF E0 E1 E2 E3 E4 E5
 CX=0200  DX=E0C2          305F:E3F2  E6 E7 E8 E9 EA EB EC ED
 SP=C772  BP=C774          305F:E3FA  EE EF F0 F1 F2 F3 F4 F5
 SI=2ECA  DI=04D2         FILE
 CS=3050  DS=305F
 SS=305F  ES=305F        PROGRAM1
 IP=004D  T:0             3050:004D  B43F            MOV     AH,3F
 A:1 C:0 P:0 S:1         BREAKPOINTS
 Z:0 D:0 I:1 O:0          00:  2E89:1C9D 0001 STOP
```

ภาพ 4.6 แสดงหน่วยความจำส่วนที่เป็นตารางการแปลงพยัญชนะเป็นตัวพิมพ์ใหญ่

```
STACK                          DATA1
  305F:C7CC  C7D6                305F:E0C2  35 32 20 32 35 33 20 32      52 253 2
  305F:C7CA  2CB8                305F:E0CA  35 34 20 32 35 35 0D 0A      54 255
  305F:C7C8  0558                305F:E0D2  1A 36 39 20 30 36 35 20      89 085
  305F:C7C6  C7D4                305F:E0DA  30 36 35 20 30 36 35 20      065 065
  305F:C7C4  E08E                305F:E0E2  30 36 35 20 30 36 37 20      065 067
  305F:C7C2  2ECA                305F:E0EA  30 36 39 20 30 36 39 20      069 069
  305F:C7C0  0385                305F:E0F2  30 36 39 20 30 37 33 20      069 073
  305F:C7BE  C7CC                305F:E0FA  30 37 33 20 30 37 33 20      073 073
  305F:C7BC  E08E                305F:E102  30 36 35 20 30 36 35 0D      065 065
  305F:C7BA  0000                305F:E10A  0A 30 36 39 20 30 36 39      069 069
 305F:C7B8  0011               305F:E112  20 30 36 39 20 30 37 39      069 079
CPU                              305F:E11A  20 30 37 39 20 30 37 39       079 079
 AX=0001  BX=0005                305F:E122  20 30 38 35 20 30 38 35       085 085
 CX=FFFF  DX=FFFF                305F:E12A  20 30 38 39 20 30 37 39       089 079
 SP=C7B8  BP=C7BE                305F:E132  20 30 38 35 20 31 35 35       085 155
 SI=2ECA  DI=005E              FILE
 CS=3050  DS=305F
 SS=305F  ES=305F             PROGRAM1
 IP=0060  T:0                  3050:0060  B442            MOV    AH,42
 A:0 C:0 P:1 S:1              BREAKPOINTS
 Z:0 D:0 I:1 C:0               00:   2E89:1C9D 0001 STOP
```

ภาพ 4.7 แสดงหน่วยความจำที่เก็บข้อมูลจากการอ่านแฟ้ม ISISUC.TAB ครั้งที่ 3

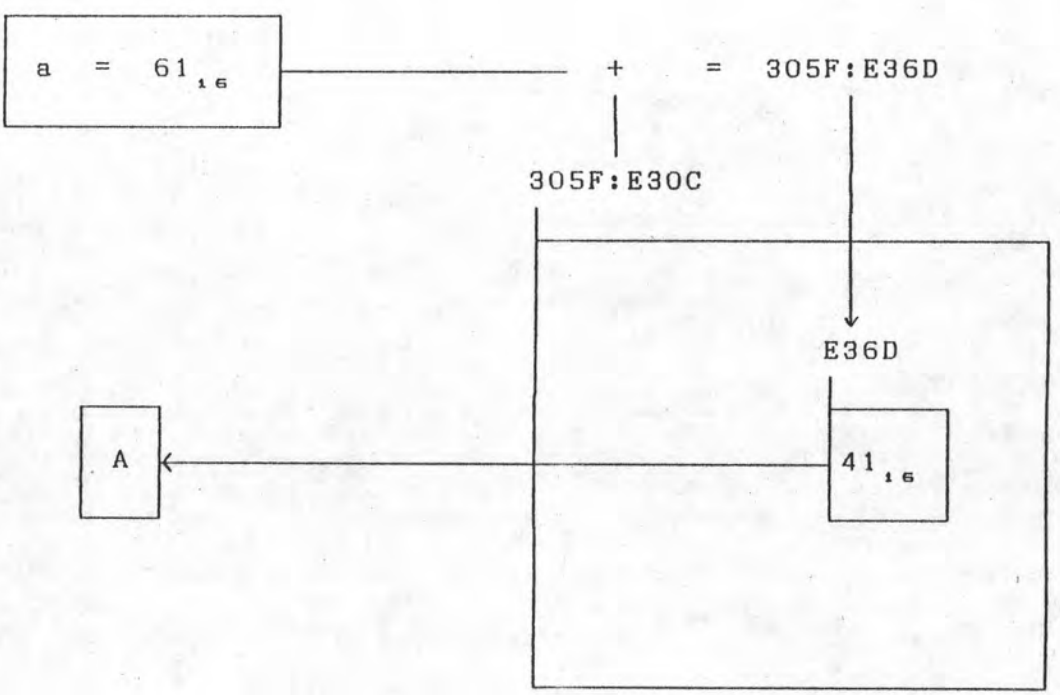
```
305F:E30C  00 01 02 03 04 05 06 07
305F:E314  08 09 0A 0B 0C 0D 0E 0F
305F:E31C  10 11 12 13 14 15 16 17
305F:E324  18 19 1A 1B 1C 1C 1E 1F
305F:E32C  20 21 22 23 24 25 26 27       !"#$%&'
305F:E334  28 29 2A 2B 2C 2D 2E 2F      ()*+,-./
305F:E33C  30 31 32 33 34 35 36 37      01234567
305F:E344  38 39 3A 3B 3C 3D 3E 3F      89:;<=>?
305F:E34C  40 41 42 43 44 45 46 47      @ABCDEFG
305F:E354  48 49 4A 4B 4C 4D 4E 4F      HIJKLMNO
305F:E35C  50 51 52 53 54 55 56 57     PQRSTUVW
305F:E364  58 59 5A 5B 5C 5D 5E 5F      XYZ[\]^_
305F:E36C  60 41 42 43 44 45 46 47      `ABCDEFG
305F:E374  48 49 4A 4B 4C 4D 4E 4F      HIJKLMNO
305F:E37C  50 51 52 53 54 55 56 57      PQRSTUVW

305F:E384  58 59 5A 7B 7C 7D 7E 7F      XYZ{|}~
305F:E38C  43 55 45 41 41 41 41 43      CUEAAAAC
305F:E394  45 45 45 49 49 49 41 41      EEEIIIAA
305F:E39C  45 45 45 4F 4F 4F 55 55      EEEOOOUU
305F:E3A4  59 4F 55 9B 9C 9D 9E 9F      YOU
305F:E3AC  41 49 4F 55 4E 4E A6 A7      AIOUNN
305F:E3B4  A8 A9 AA AB AC AD AE AF
305F:E3BC  B0 B1 B2 B3 B4 B5 B6 B7
305F:E3C4  B8 B9 BA BB BC BD BE BF
305F:E3CC  C0 C1 C2 C3 C4 C5 C6 C7
305F:E3D4  C8 C9 CA CB CC CD CE CF
305F:E3DC  D0 D1 D2 D3 D4 D5 D6 D7
305F:E3E4  D8 D9 DA DB DC DD DE DF
305F:E3EC  E0 E1 E2 E3 E4 E5 E6 E7

305F:E3F4  E8 E9 EA EB EC ED EE EF
305F:E3FC  F0 F1 F2 F3 F4 F5 F6 F7
305F:E404  F8 F9 FA FB FC FD FE FF
305F:E40C  00 00 00 00 00 00 00 00
305F:E414  00 00 00 00 00 00 00 00
305F:E41C  00 00 00 00 00 00 00 00
305F:E424  00 00 00 00 00 00 00 00
305F:E42C  00 00 00 00 00 00 00 00
305F:E434  00 00 00 00 00 00 00 00
```

ภาพ 4.8 แสดงหน่วยความจำส่วนที่เป็นตารางการแปลงพยัญชนะเป็นตัวพิมพ์ใหญ่ทั้งหมด

6. จากนั้น ระบบจะปิดแฟ้มข้อมูล ISISUC.TAB แล้วเริ่มประมวลผลต่อไป ซึ่งขณะนี้ ในหน่วยความจำมีตารางแปลงพยัญชนะเป็นตัวพิมพ์ใหญ่พร้อมที่จะทำการแปลงรหัสเรียบร้อยแล้ว โดยเป็นตารางขนาด 256 ไบท์ เวลาที่ระบบต้องการแสดงผลลัพธ์ของอักขระใด ๆ ในรูปที่จะแปลงค่า ก็จะมาค้นในตารางนี้ โดยมาที่ตำแหน่งที่ตรงกับลำดับที่ของอักขระนั้น ๆ แล้วเอาค่าแอสกีที่เก็บอยู่ ณ ตำแหน่งนั้นเป็นอักขระผลลัพธ์ ตัวอย่างเช่น ต้องการแปลงอักขระที่มีค่าแอสกีเป็น 61_{16} (อักขระ a) ระบบจะมาค้นในตารางนี้ โดยเอาค่า 61_{16} ใช้เป็นค่า offset และเนื่องจากตารางเริ่มต้นที่ตำแหน่ง 305F:E30C ดังนั้น เมื่อเอาค่ามาบวกกันก็จะได้ตำแหน่งของรหัสที่แปลงแล้วของ 61_{16}

$$305F:E30C_{16} + 0061_{16} = 305F:E36D_{16}$$

นั่นคือ อยู่ที่ตำแหน่ง 305F:E36D ซึ่งเก็บค่า 41_{16} นั่นคืออักขระ A นั่นเอง ระบบก็จะนำค่า 41_{16} หรือ อักขระ A นี้ไปเป็นผลลัพธ์ สามารถแสดงกลไกการแปลงพยัญชนะเป็นตัวพิมพ์ใหญ่ได้ดังภาพ 4.9



ภาพที่ 4.9 แสดงกลไกการแปลงพยัญชนะเป็นตัวพิมพ์ใหญ่

โครงสร้างข้อมูลของแฟ้มข้อมูลตารางอักขระพยัญชนะ

แฟ้มข้อมูลตารางอักขระพยัญชนะ (ISISAC.TAB หรือ ISIS Alphabetic Table Table) ทำหน้าที่จัดเก็บรหัสแอสกีในรูปเลขฐานสิบจำนวน 85 รหัส แต่ละค่าเป็นเลขจำนวนเต็ม 3 หลัก คั่นด้วยช่องว่าง 1 ช่อง แบ่งออกเป็น 5 บรรทัด ๆ ละ 16 รหัส โดยที่บรรทัดที่ 5 มี 5 รหัส ดังภาพ 4.10

```
065 066 067 068 069 070 071 072 073 074 075 076 077 078 079 080
081 082 083 084 085 086 087 088 089 090 097 098 099 100 101 102
103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118
119 120 121 122 128 129 130 131 132 133 134 135 136 137 138 139
140 141 142 143 144 145 146 147 148 149 150 151 152 153 154 160
161 162 163 164 165
```

ภาพ 4.10 แสดงข้อมูลของแฟ้มข้อมูลตารางอักขระพยัญชนะ

จากภาพ 4.10 จะเห็นได้ว่า รหัสทั้ง 85 ค่ามีค่าอยู่ระหว่าง 065 - 165 เท่านั้น ซึ่งเป็นรหัสที่ผู้ติดตั้งระบบจัดไว้ให้เป็นพวกพยัญชนะ เพื่อใช้เวลาสร้างดัชนีตลอดจนใช้ตรวจสอบข้อมูลในเขตข้อมูลที่เป็นพยัญชนะล้วน (Alphabetic field) รหัสที่ระบบถือว่าเป็นพยัญชนะทั้ง 85 รหัสเมื่อเปรียบเทียบกับตารางรหัสแอสกีจะเป็นดังตาราง 4.3

รหัสในแฟ้มข้อมูล ISISAC.TAB	สัญญลักษณ์ของรหัสแอสกี
065	A
066	B
067	C
068	D
069	E
070	F
071	G
072	H
073	I
074	J
075	K
076	L
077	M
078	N
079	O
080	P
081	Q
082	R
083	S
084	T
085	U
086	V
087	W
088	X
089	Y
090	Z

ตาราง 4.3 แสดงสัญญลักษณ์รหัสแอสกีของข้อมูลในแฟ้มข้อมูลตารางอักขระพยัญชนะ

รหัสในแฟ้มข้อมูล ISISAC.TAB	สัญญลักษณ์ของรหัสแอสกี
097	a
098	b
099	c
100	d
101	e
102	f
103	g
104	h
105	i
106	j
107	k
108	l
109	m
110	n
111	o
112	p
113	q
114	r
115	s
116	t
117	u
118	v
119	w
120	x
121	y
122	z

ตาราง 4.3 แสดงสัญญลักษณ์รหัสแอสกีของข้อมูลในแฟ้มข้อมูลตารางอักขระพยัญชนะ

รหัสในแฟ้มข้อมูล ISISAC.TAB	สัญญลักษณ์ของรหัสแอสกี
128	ç
129	ü
130	é
131	â
132	ä
133	à
134	å
135	ç
136	ê
137	ë
138	è
139	ï
140	î
141	ì
142	Ä
143	Å
144	É
145	æ
146	Æ
147	ô
148	ö
149	ò
150	û
151	ù
152	ÿ
153	Ö

ตาราง 4.3 แสดงสัญญลักษณ์รหัสแอสกีของข้อมูลในแฟ้มข้อมูลตารางอักขระพยัญชนะ (ต่อ)

รหัสในแฟ้มข้อมูล ISISAC.TAB	สัญญลักษณ์ของรหัสแอสกี
154	Ü
160	á
161	í
162	ó
163	ú
164	ñ
165	Ñ

ตาราง 4.3 แสดงสัญญลักษณ์รหัสแอสกีของข้อมูลในแฟ้มข้อมูลตารางอักขระพยัญชนะ (ต่อ)

จากตาราง 4.3 จะพบว่า อักขระที่ผู้ติดตั้งระบบกำหนดให้เป็นพยัญชนะมีทั้งสิ้น 85 อักขระ จากอักขระแอสกีทั้งสิ้น 256 รหัส ซึ่งนอกจากจะรวมถึงทั้งพวกพยัญชนะตัวพิมพ์เล็ก (a, b, c, ..., z) และพยัญชนะตัวพิมพ์ใหญ่ (A, B, C, ... ,Z) แล้ว ยังรวมพยัญชนะที่มีสัญญลักษณ์เน้นเสียงต่าง ๆ อีกด้วย ลักษณะการใช้งานตารางนี้ก็คือ ระบบจะเตรียมหน่วยความจำเอาไว้สำหรับตารางอักขระพยัญชนะภายในหน่วยความจำหลักโดยอยู่ติดกับตารางการแปลงอักขระเป็นตัวพิมพ์ใหญ่ ขนาด 256 ไบท์เท่ากัน แต่ตอนเริ่มต้น จะกำหนดให้ทุกไบท์ในตารางมีค่าเป็นศูนย์ แล้วอ่านข้อมูลจากแฟ้มข้อมูลตารางอักขระพยัญชนะ ซึ่งเป็นรหัสแอสกีของพวกพยัญชนะมาทีละค่า วิเคราะห์ว่าเป็นรหัสอะไร จากนั้นก็ไปกำหนดค่าในตารางอักขระพยัญชนะขนาด 1 ไบท์ ที่ตำแหน่งนั้น ๆ ให้มีค่าเป็นหนึ่ง เมื่อเสร็จหมดแล้ว จะได้ตารางอักขระพยัญชนะ ขนาด 256 ไบท์ ซึ่งมีค่าเป็น 0 หรือ 1 เท่านั้น ข้อมูลในตารางที่ตรงกับลำดับที่ของรหัสแอสกีใด ๆ ถ้าเป็นศูนย์แสดงว่า รหัสแอสกีนั้น ๆ ระบบถือว่าไม่ใช่พยัญชนะ ถ้าเป็นหนึ่ง แสดงว่า รหัสแอสกีนั้น ๆ ระบบถือว่าเป็นพยัญชนะ

ในการใช้งานโปรแกรมมินิ-ไมโคร ซีดีเอส/ไอซิส ทั้ง 6 โปรแกรมนั้น เมื่อผู้ใช้เรียกให้โปรแกรมใดโปรแกรมหนึ่งทำงาน โปรแกรมจะทำการเปิดแฟ้มข้อมูลตารางอักขระพยัญชนะเป็นแฟ้มข้อมูลลำดับที่สอง ถัดจากแฟ้มข้อมูลตารางการแปลงพยัญชนะเป็นตัวพิมพ์ใหญ่เสมอ โดยมีลักษณะการประมวลผลคล้ายกันทุกประการ คือ แฟ้มข้อ

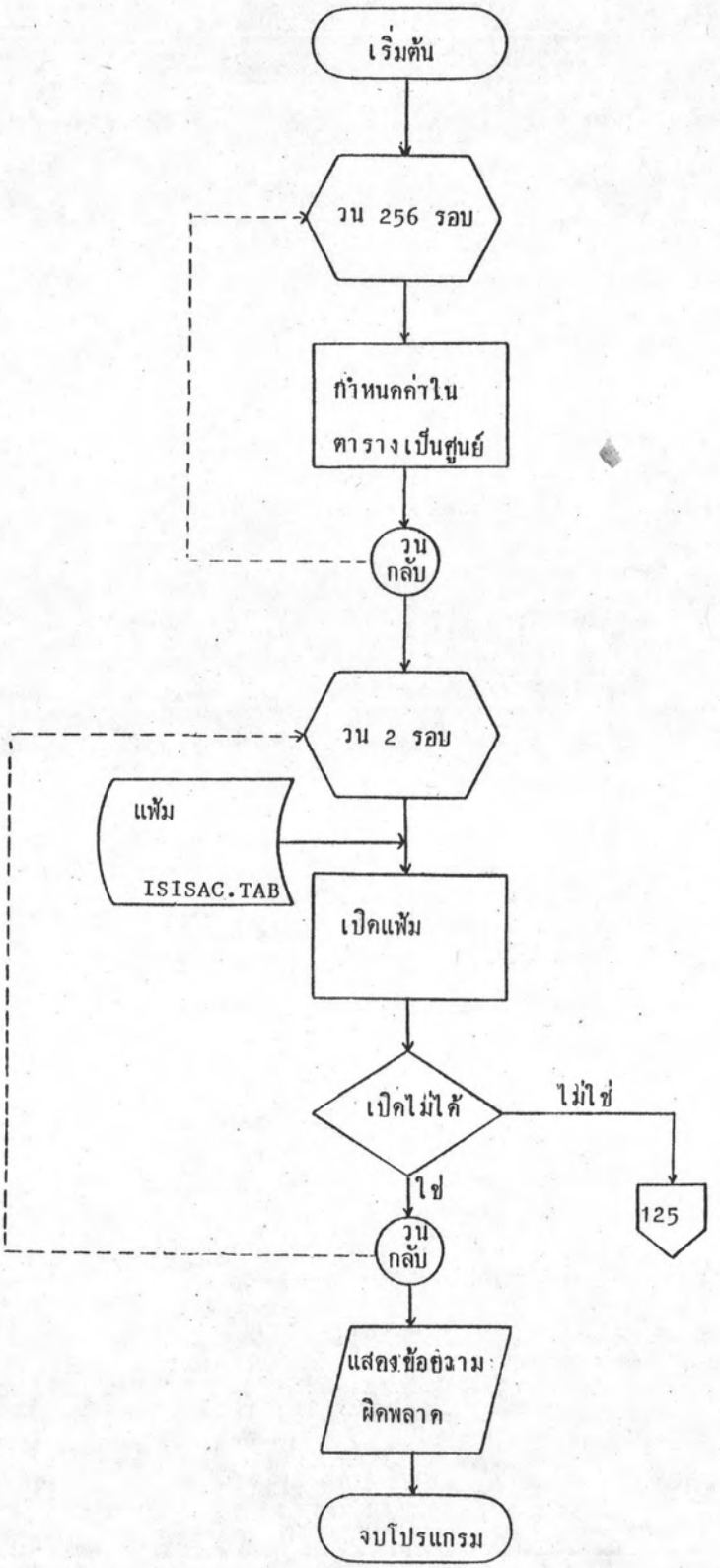
มูลนี้จะถูกอ่านในช่วงแรกของการประมวลผลเพียงครั้งเดียวเท่านั้น และผู้ใช้ไม่สามารถบรรณาธิกรแฟ้มข้อมูลนี้ได้โดยตรงจากรายการประมวลผลในโปรแกรม ถ้าต้องการบรรณาธิกรแฟ้มข้อมูลนี้ จะต้องใช้โปรแกรมบรรณาธิกรต่างหาก ซึ่งการเปลี่ยนแปลงแก้ไขทำได้ง่ายเนื่องจากแฟ้มข้อมูลดังกล่าวมีลักษณะเป็นแฟ้มข้อมูลแอสกี ซึ่งข้อมูลเป็นตัวเลขจำนวนเต็ม สามหลัก คั่นด้วยช่องว่างอย่างน้อยหนึ่งช่อง แต่ละบรรทัดจะใส่รหัสกี่ค่าก็ได้ แต่ให้จบแต่ละบรรทัดด้วยการกดปุ่ม <ENTER>

รายละเอียดขั้นตอนการอ่านข้อมูลจากแฟ้มข้อมูลเก็บในหน่วยความจำ สามารถเขียนเป็นผังงานได้ดังในผังงานที่ 4.2

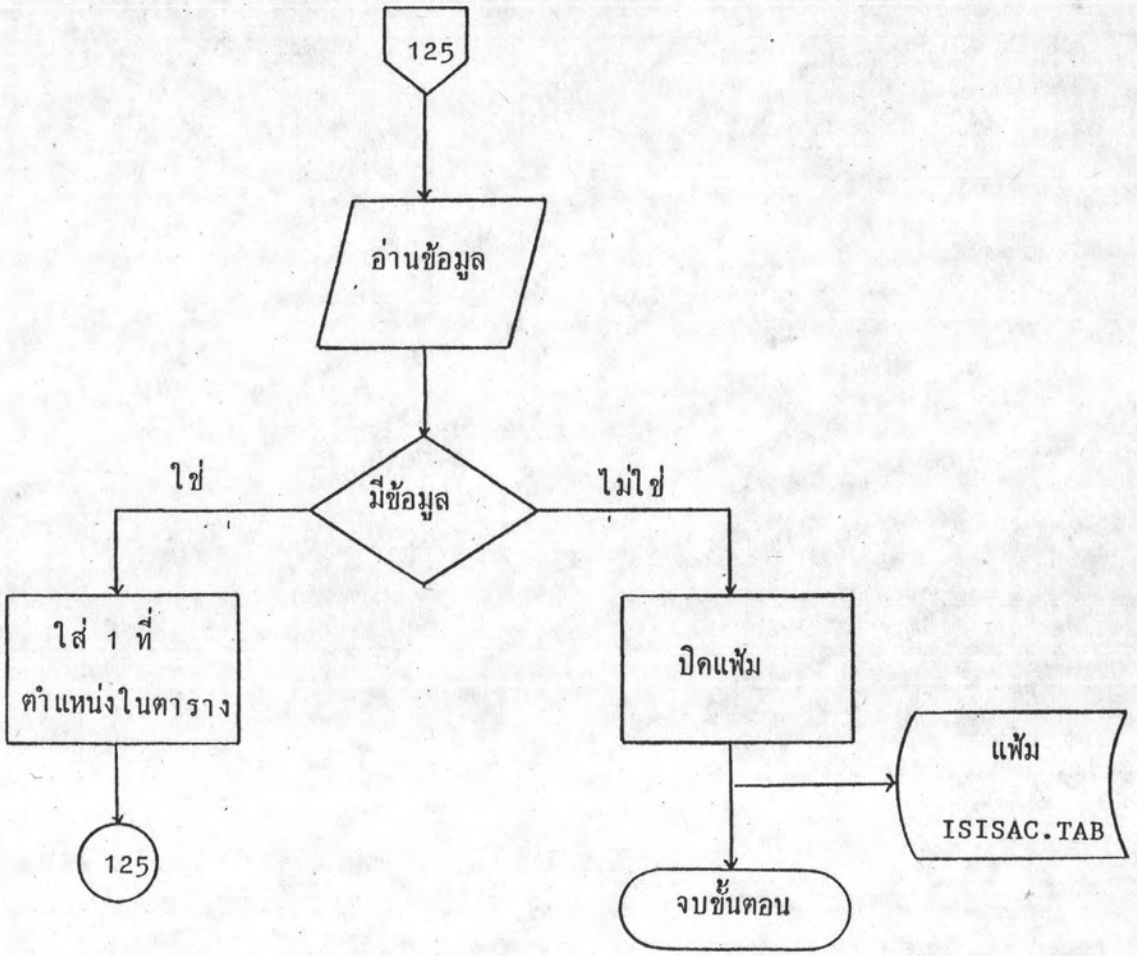
1. เมื่อเริ่มประมวลผล ภายหลังจากที่ระบบจัดเตรียมตารางการแปลงพยัญชนะเป็นตัวพิมพ์ใหญ่เสร็จ ก็จะเริ่มเตรียมตารางอักขระพยัญชนะในหน่วยความจำ โดยใช้เนื้อที่ต่อจากตารางการแปลงพยัญชนะเป็นตัวพิมพ์ใหญ่ จากตัวอย่างเดิม ตารางการแปลงพยัญชนะเป็นตัวพิมพ์ใหญ่ใช้เนื้อที่สิ้นสุดที่ 305F:E40B ระบบก็จะจองเนื้อที่ต่อจากนั้นอีก 256 ไบท์ นั่นคือ ตั้งแต่ 305F:E40C จนถึง 305F:E50C และกำหนดให้ทุกไบท์มีค่าเป็นศูนย์ (30_{16}) ดังภาพ 4.11

```
305F:E40C  30 30 30 30 30 30 30 30      00000000
305F:E414  30 30 30 30 30 30 30 30      00000000
305F:E41C  30 30 30 30 30 30 30 30      00000000
305F:E424  30 30 30 30 30 30 30 30      00000000
305F:E42C  30 30 30 30 30 30 30 30      00000000
305F:E434  30 30 30 30 30 30 30 30      00000000
305F:E43C  30 30 30 30 30 30 30 30      00000000
305F:E444  30 30 30 30 30 30 30 30      00000000
305F:E44C  30 30 30 30 30 30 30 30      00000000
305F:E454  30 30 30 30 30 30 30 30      00000000
305F:E45C  30 30 30 30 30 30 30 30      00000000
305F:E464  30 30 30 30 30 30 30 30      00000000
305F:E46C  30 30 30 30 30 30 30 30      00000000
305F:E474  30 30 30 30 30 30 30 30      00000000
305F:E47C  30 30 30 30 30 30 30 30      00000000
305F:E484  30 30 30 30 30 30 30 30      00000000
305F:E48C  30 30 30 30 30 30 30 30      00000000
305F:E494  30 30 30 30 30 30 30 30      00000000
305F:E49C  30 30 30 30 30 30 30 30      00000000
305F:E4A4  30 30 30 30 30 30 30 30      00000000
305F:E4AC  30 30 30 30 30 30 30 30      00000000
305F:E4B4  30 30 30 30 30 30 30 30      00000000
305F:E4BC  30 30 30 30 30 30 30 30      00000000
305F:E4C4  30 30 30 30 30 30 30 30      00000000
305F:E4CC  30 30 30 30 30 30 30 30      00000000
305F:E4D4  30 30 30 30 30 30 30 30      00000000
305F:E4DC  30 30 30 30 30 30 30 30      00000000
305F:E4E4  30 30 30 30 30 30 30 30      00000000
```

ภาพ 4.11 แสดงหน่วยความจำที่ระบบจัดเตรียมไว้สำหรับเป็นตารางอักขระพยัญชนะ



ผังงานที่ 4.2 ผังแสดงขั้นตอนการประมวลผลแฟ้มข้อมูลตารางอักขระพยัญชนะ



ผังงานที่ 4.2 ผังแสดงขั้นตอนการประมวลผลแฟ้มข้อมูลตารางอักขระพยัญชนะ (ต่อ)

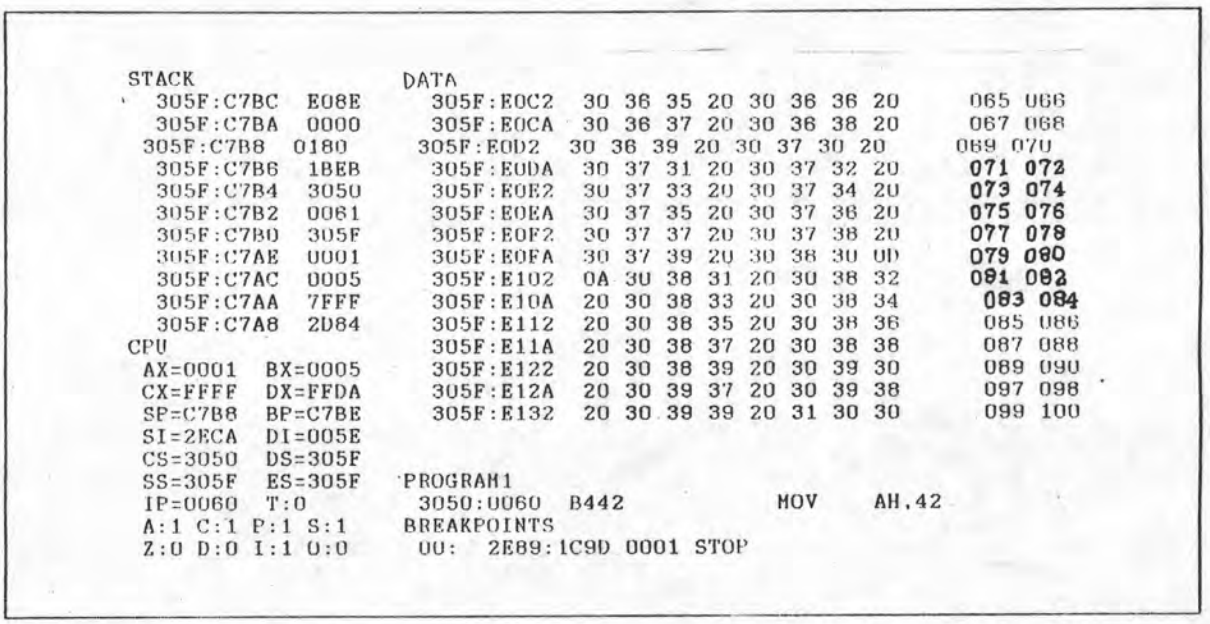
2. จากนั้น ระบบจะเปิดแฟ้มข้อมูล ISISAC.TAB ถ้าครั้งแรกเปิดไม่ได้ จะลองเปิดครั้งที่สอง ถ้าเปิดแฟ้มข้อมูลดังกล่าวไม่ได้ จะแสดงข้อความผิดพลาดบนจอภาพ แล้วจบการทำงาน ดังตัวอย่างในภาพ 4.12

```
C>ISISPRT     <ENTER>
? Error:File not found error in file ISISAC.TAB
  Error Code 1032, Status 0002
PC = 0004: 2CB8; SS = 305F, FP = 0921, SP = C7E0

C>
```

ภาพ 4.12 จอภาพแสดงผลลัพธ์การประมวลผลเมื่อไม่มีแฟ้มข้อมูล ISISAC.TAB

3. ถ้าเปิดแฟ้มข้อมูล ISISAC.TAB สำเร็จ ระบบจะเริ่มอ่านข้อมูลมาใส่หน่วยความจำ โดยอ่านมาครั้งละ 512 ไบท์ ดังตัวอย่างในภาพ 4.13 เป็นภาพแสดงหน่วยความจำหลังจากที่โปรแกรม ISISPRT.EXE อ่านข้อมูลจากแฟ้มข้อมูล ISISAC.TAB ครั้งแรก ค่าในรีจิสเตอร์แสดงว่า แฟ้มข้อมูล ISISAC.TAB ถูกกำหนดโดยระบบให้มีค่าแฮนเดิลหมายเลข 5 อ่านข้อมูลจำนวน 512 ไบท์ (200_{16} ไบท์) เก็บไว้ในหน่วยความจำหลักโดยเริ่มตั้งแต่ DS:DX คือ 305F:E0C2 จนถึง 305F:E2C1 โดยเก็บเรียงลำดับไปเรื่อย ๆ ในลักษณะเดียวกับที่เก็บอยู่ในแฟ้มข้อมูลจริง ๆ

4. จากนั้น ระบบจะเริ่มอ่านข้อมูลที่เป็นรหัสในรูปเลขฐานสิบจำนวน 3 หลักทีละค่าไปแปลงเป็นรหัสแอสกีในรูปเลขฐานสิบหก 1 ไบท์ ใช้เป็นค่า offset บวกกับตำแหน่งเริ่มต้นของตาราง ตัวอย่างนี้คือ DS:DX = 305F:E40C ได้ตำแหน่งที่ตรงกับลำดับที่ของรหัสแอสกีนั้น ๆ ในตาราง แล้วนำค่า 1 ไปเก็บไว้ในหน่วยความจำตำแหน่งนั้น



ภาพ 4.13 แสดงหน่วยความจำที่เก็บข้อมูลจากการอ่านแฟ้มข้อมูล ISISAC.TAB ครั้งแรก

5. ระบบจะวนอ่านข้อมูล แปลงเป็นตำแหน่ง แล้วใส่ 1 ที่ตำแหน่งนั้น ๆ ไปเรื่อย ๆ จนหมดข้อมูลในแฟ้มข้อมูลตารางอักขระพยัญชนะ ในที่นี้คือ 85 รหัส

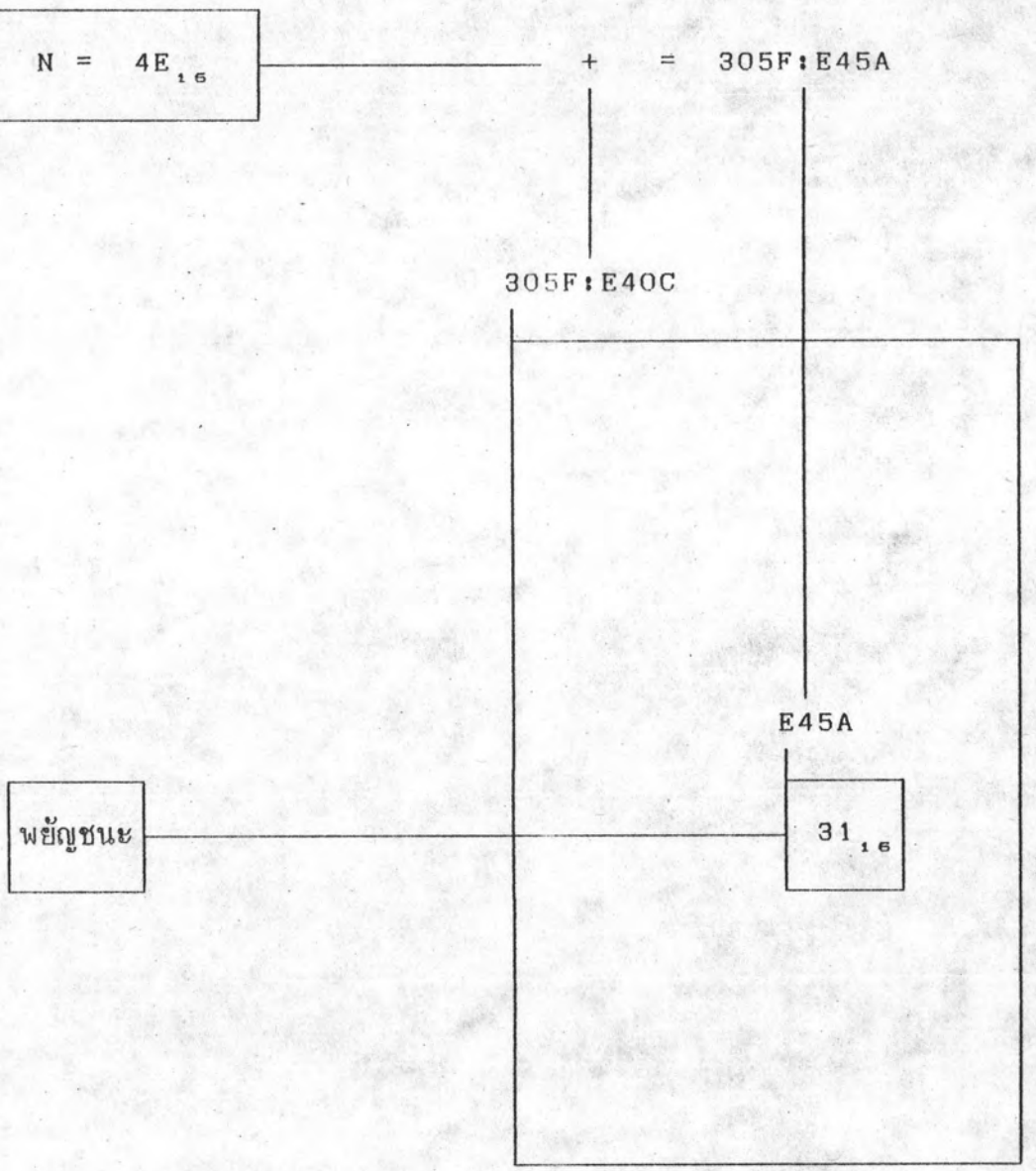
6. จากนั้น ระบบจะปิดแฟ้มข้อมูล ISISAC.TAB แล้วเริ่มประมวลผลต่อไป ซึ่งขณะนี้ ในหน่วยความจำที่อยู่ติดกันกับตารางการแปลงพยัญชนะเป็นตัวพิมพ์ใหญ่จะมีตารางอักขระพยัญชนะพร้อมที่จะให้ระบบทำการตรวจสอบอักขระใด ๆ ว่าเป็นพยัญชนะหรือไม่แล้ว โดยดูจากค่าในตาราง ซึ่งตำแหน่งที่ระบบถือว่าอักขระนั้น ๆ เป็นพยัญชนะก็คือ ตำแหน่งที่เก็บค่า 1 (31_{16}) และตำแหน่งที่ไม่ใช่พยัญชนะจะเก็บค่า 0 (30_{16}) ดังตัวอย่างในภาพ 4.14 เวลาระบบจะตรวจสอบ ก็จะมาค้นในตารางนี้ โดยมาที่ตำแหน่งที่ตรงกับลำดับที่ของอักขระนั้น ๆ แล้วเอาค่าที่เก็บอยู่ ณ ตำแหน่งนั้น (1 ไบท์) ไปเปรียบเทียบ ถ้าเป็น 0 (30_{16}) แสดงว่าไม่ใช่พยัญชนะ ถ้าเป็น 1 (31_{16}) แสดงว่าเป็นพยัญชนะ ตัวอย่างเช่น ต้องการตรวจสอบว่า อักขระ N ซึ่งมีค่าแอสกีเป็น 78_{10} หรือ $4E_{16}$ เป็นพยัญชนะหรือไม่ ระบบจะไปค้นในตารางอักขระพยัญชนะ โดยเอาค่า $4E_{16}$ ใช้เป็นค่า offset และเนื่องจากตารางเริ่มต้นที่ตำแหน่ง 305F:E40C ดังนั้น เมื่อเอาค่าทั้งสองมาบวกกัน ก็จะได้ตำแหน่งที่ตรงกับลำดับที่ของรหัสระบุพยัญชนะ

```
305F:E40C  30 30 30 30 30 30 30 30     00000000
305F:E414  30 30 30 30 30 30 30 30     00000000
305F:E41C  30 30 30 30 30 30 30 30     00000000
305F:E424  30 30 30 30 30 30 30 30     00000000
305F:E42C  30 30 30 30 30 30 30 30     00000000
305F:E434  30 30 30 30 30 30 30 30     00000000
305F:E43C  30 30 30 30 30 30 30 30     00000000
305F:E444  30 30 30 30 30 30 30 30     00000000
305F:E44C  30 31 31 31 31 31 31 31     01111111
305F:E454  31 31 31 31 31 31 31 31     11111111
305F:E45C  31 31 31 31 31 31 31 31     11111111
305F:E464  31 31 31 30 30 30 30 30     11100000
305F:E46C  30 31 31 31 31 31 31 31     01111111
305F:E474  31 31 31 31 31 31 31 31     11111111
305F:E47C  31 31 31 31 31 31 31 31     11111111

305F:E484  31 31 31 30 30 30 30 30     11100000
305F:E48C  31 31 31 31 31 31 31 31     11111111
305F:E494  31 31 31 31 31 31 31 31     11111111
305F:E49C  31 31 31 31 31 31 31 31     11111111
305F:E4A4  31 31 31 30 30 30 30 30     11100000
305F:E4AC  31 31 31 31 31 31 30 30     11111100
305F:E4B4  30 30 30 30 30 30 30 30     00000000
305F:E4BC  30 30 30 30 30 30 30 30     00000000
305F:E4C4  30 30 30 30 30 30 30 30     00000000
305F:E4CC  30 30 30 30 30 30 30 30     00000000
305F:E4D4  30 30 30 30 30 30 30 30     00000000
305F:E4DC  30 30 30 30 30 30 30 30     00000000
305F:E4E4  30 30 30 30 30 30 30 30     00000000
305F:E4EC  30 30 30 30 30 30 30 30     00000000

305F:E4F4  30 30 30 30 30 30 30 30     00000000
305F:E4FC  30 30 30 30 30 30 30 30     00000000
305F:E504  30 30 30 30 30 30 30 30     00000000
305F:E50C  42 57 02 00 10 47 44 42     BWGDB
305F:E514  44 53 33 32 4B 2E 50 41     DS32K.PA
305F:E51C  53 20 20 20 20 00 00 00     S
305F:E524  42 57 02 00 0C 47 44 42     BWGDB
305F:E52C  4D 45 4D 2E 50 41 53 20     MEM.PAS
305F:E534  20 00 00 00 00 00 42 57      BW
305F:E53C  02 00 0C 47 44 42 55 54     GDBUT
305F:E544  4C 45 2E 50 41 53 20 00     LE.PAS
305F:E54C  00 00 42 57 02 00 0C 47     BWG
305F:E554  44 42 45 44 49 54 2E 50     DBEDIT.P
305F:E55C  41 53 20 00 00 00 00 00     AS
```

ภาพ 4.14 แสดงหน่วยความจำส่วนที่เป็นตารางอักขระพยัญชนะทั้งหมด

$$305F{:}E40C_{16} + 4E_{16} = 305F{:}E45A_{16}$$

นั่นคือ อยู่ที่ตำแหน่ง 305F:E45A ซึ่งเก็บค่า 1 (31_{16}) แสดงว่าเป็นพยัญชนะนั่นเอง สามารถแสดงกลไกการตรวจสอบอักขระว่าเป็นพยัญชนะหรือไม่ได้ดังภาพ 4.15



ภาพที่ 4.15 แสดงกลไกการตรวจสอบอักขระว่าเป็นพยัญชนะหรือไม่

โครงสร้างข้อมูลของแฟ้มข้อมูลตารางกำหนดเขตข้อมูล

แฟ้มข้อมูลตารางกำหนดเขตข้อมูล (Field Definition Table หรือ FDT) เป็นแฟ้มข้อมูลระบบที่เก็บโครงสร้างฐานข้อมูล กำหนดเขตข้อมูล ความยาวเขตข้อมูล และลักษณะข้อมูลที่เก็บ ตลอดจนชื่อแฟ้มข้อมูลแผ่นงานบันทึกข้อมูล, ชื่อแฟ้มข้อมูลการแสดงรูปแบบผลลัพธ์ทางจอภาพ และชื่อแฟ้มข้อมูลตารางเขตข้อมูลที่เลือกของแต่ละฐานข้อมูลที่ผู้ใช้สร้างขึ้น องค์ประกอบของแฟ้มข้อมูลประกอบด้วย 2 ส่วนใหญ่ ๆ ได้แก่ ส่วนระบุชื่อแฟ้มที่เกี่ยวข้อง และส่วนระบุโครงสร้างฐานข้อมูล

1. ส่วนระบุชื่อแฟ้มที่เกี่ยวข้อง

ส่วนระบุชื่อแฟ้มที่เกี่ยวข้องประกอบด้วยระเบียนข้อมูลความยาวแปรผัน แบ่งได้เป็น 3 พวก ได้แก่ เขตข้อมูลระบุชื่อแผ่นงานบันทึกข้อมูล, เขตข้อมูลระบุชื่อรูปแบบการแสดงผลลัพธ์บนจอภาพ และเขตข้อมูลระบุชื่อตารางเขตข้อมูลที่เลือก

1.1 เขตข้อมูลระบุชื่อแผ่นงานบันทึกข้อมูล เป็นเขตข้อมูลความยาวแปรผัน ประกอบด้วยเขตข้อมูลย่อยที่มีความยาวสูงสุดไม่เกิน 62 ไบท์ ขึ้นต้นด้วย "W:" แล้วตามด้วยชื่อแผ่นงานบันทึกข้อมูลได้ไม่เกิน 10 ชื่อ แต่ละชื่อความยาวคงที่ไม่เกิน 6 อักขระ ถ้าไม่ถึงระบบจะใส่ช่องว่างให้ ถ้าระบุชื่อยาวเกิน 6 อักขระ ระบบจะตัดส่วนที่เกินทางขวาทิ้ง ถ้ามีแผ่นงานมากกว่า 10 แฟ้ม ระบบจะเก็บในเขตข้อมูลย่อยถัดไปโดยปิดท้ายเขตข้อมูลเดิมด้วย <ENTER> (OD_{16} OA_{16}) คือขึ้นบรรทัดใหม่แล้วเริ่มต้นเขตข้อมูลย่อยแผ่นงานบันทึกข้อมูลถัดไปเรื่อย ๆ เขตละไม่เกิน 10 ชื่อ โดยไม่จำกัดจำนวนเขตข้อมูลย่อย แต่จะขึ้นต้นเขตข้อมูลย่อยใหม่ด้วย "W:" เหมือนเดิม ดังตัวอย่าง

```
W:TEST  TEST01TEST02TEST03TEST04TEST05TEST06TEST07TEST08TEST09
W:TEST10
```

ตัวอย่างข้างต้นแสดงว่ามีแผ่นงานบันทึกข้อมูลอยู่ 11 กลุ่ม อยู่ในฐานข้อมูล ชื่อ TEST (แผ่นงานบันทึกข้อมูลลำดับแรกเป็นแผ่นงานที่ถูกสร้างขึ้นเมื่อเริ่มกำหนดฐานข้อมูล จะมีชื่อเดียวกันกับฐานข้อมูล และจะต้องมีอยู่ ไม่สามารถลบทิ้งได้) ทุกชื่อเวลาถูกเรียกใช้งาน ระบบจะไปค้นหาแฟ้มตามชื่อที่ระบุแต่จะเพิ่มอักขระ A, B, C,... ไว้ข้างหน้าชื่อ และมีส่วนขยายชื่อแฟ้มเป็น .FMT เช่น ถ้าเรียกใช้

แผ่นงานชื่อ TEST ระบบจะไปเปิดแฟ้ม ATEST.FMT, BTEST.FMT,... เป็นต้น เมื่อ A, B,... เป็นลำดับที่ก่อน-หลังของแผ่นงานบันทึกข้อมูล

1.2 เขตข้อมูลระบุชื่อแฟ้มรูปแบบแสดงผลลัพธ์ เป็นเขตข้อมูลความยาวแปรผัน ประกอบด้วยเขตข้อมูลย่อยที่มีความยาวสูงสุดไม่เกิน 62 ไบท์ ขึ้นต้นด้วย "F:" แล้วตามด้วยชื่อแฟ้มรูปแบบการแสดงผลลัพธ์ได้ไม่เกิน 10 ชื่อ แต่ละชื่อความยาวคงที่ไม่เกิน 6 อักขระ ถ้าไม่ถึงระบบจะใส่ช่องว่างให้ ถ้าระบุชื่อยาวเกิน 6 อักขระ ระบบจะตัดส่วนที่เกินทางขวาทิ้ง ถ้ามีชื่อแฟ้มมากกว่า 10 แฟ้ม ระบบจะเก็บในเขตข้อมูลย่อยถัดไปโดยปิดท้ายเขตข้อมูลเดิมด้วย <ENTER> (OD_{16} OA_{16}) คือขึ้นบรรทัดใหม่แล้วเริ่มต้นเขตข้อมูลย่อยชื่อแฟ้มรูปแบบการแสดงผลลัพธ์ถัดไปเรื่อย ๆ จำนวนเขตละไม่เกิน 10 ชื่อ โดยไม่จำกัดจำนวนเขตข้อมูลย่อย แต่จะขึ้นต้นเขตข้อมูลย่อยใหม่ด้วย "F:" เหมือนเดิม ดังตัวอย่าง

```
F:BASE  BASE01BASE02BASE03BASE04BASE05BASE06BASE07BASE08BASE09
F:BASE10BASE11
```

ตัวอย่างข้างต้นแสดงว่ามีแฟ้มรูปแบบการแสดงผลลัพธ์อยู่ 12 แฟ้ม อยู่ในฐานข้อมูล ชื่อ BASE (แฟ้มรูปแบบการแสดงผลลัพธ์ลำดับแรกเป็นแฟ้มที่ถูกสร้างขึ้นเมื่อเริ่มกำหนดฐานข้อมูล จะมีชื่อเดียวกันกับฐานข้อมูล และจะต้องมีอยู่ ไม่สามารถลบทิ้งได้)

1.3 เขตข้อมูลระบุชื่อแฟ้มตารางเขตข้อมูลที่เลือก เป็นเขตข้อมูลความยาวแปรผัน ประกอบด้วยเขตข้อมูลย่อยที่มีความยาวสูงสุดไม่เกิน 62 ไบท์ ขึ้นต้นด้วย "S:" แล้วตามด้วยชื่อแฟ้มตารางเขตข้อมูลที่เลือกได้ไม่เกิน 10 ชื่อ แต่ละชื่อความยาวคงที่ไม่เกิน 6 อักขระ ถ้าไม่ถึงระบบจะใส่ช่องว่างให้ ถ้าระบุชื่อยาวเกิน 6 อักขระ ระบบจะตัดส่วนที่เกินทางขวาทิ้ง ถ้ามีชื่อแฟ้มมากกว่า 10 แฟ้ม ระบบจะเก็บในเขตข้อมูลย่อยถัดไปโดยปิดท้ายเขตข้อมูลเดิมด้วย <ENTER> (OD_{16} OA_{16}) คือขึ้นบรรทัดใหม่แล้วเริ่มต้นเขตข้อมูลย่อยชื่อแฟ้มตารางเขตข้อมูลที่เลือกถัดไปเรื่อย ๆ เขตละไม่เกิน 10 ชื่อ โดยไม่จำกัดจำนวนเขตข้อมูลย่อย แต่จะขึ้นต้นเขตข้อมูลย่อยใหม่ด้วย "S:" เหมือนเดิม ดังตัวอย่าง

```
S:STOCK1STOCK2STOCK3STOCK4STOCK5STOCK6
```

ตัวอย่างข้างต้นแสดงว่ามีแฟ้มรูปแบบการแสดงผลลัพธ์อยู่ 6 แฟ้ม อยู่ในฐานข้อมูล ชื่อ STOCK1 (แฟ้มรูปแบบการแสดงผลลัพธ์ลำดับแรกเป็นแฟ้มที่ถูกสร้างขึ้นเมื่อเริ่มกำหนดฐานข้อมูล จะมีชื่อเดียวกันกับฐานข้อมูล และจะต้องมีอยู่ ไม่สามารถลบทิ้งได้)

2. ส่วนระบุโครงสร้างฐานข้อมูล

ส่วนระบุโครงสร้างฐานข้อมูลประกอบด้วยระเบียนข้อมูลที่มีความยาวคงที่ โดยแต่ละระเบียนจะแทนโครงสร้างข้อมูล 1 เขตของฐานข้อมูลที่กำหนด อยู่ต่อเนื่องกับส่วนระบุชื่อแฟ้มที่เกี่ยวข้อง โดยมีบรรทัด "***" คั่นระหว่าง 2 ส่วนนี้ แต่ละระเบียนประกอบด้วยเขตข้อมูลความยาวคงที่ จำนวน 6 เขต ดังนี้

2.1 เขตหมายเลขประจำเขตข้อมูล ขนาด 3 อักขระ ประกอบด้วยตัวเลข 3 หลัก มีค่าตั้งแต่ 1 - 999 ใช้อ้างถึงเขตข้อมูล หรือเขตข้อมูลย่อยของฐานข้อมูลที่สร้าง

2.2 เขตชื่อเขตข้อมูล (Field Name) ประกอบด้วยตัวอักษร ความยาวไม่เกิน 30 ตัว ใช้แสดงความหมายหรือประเภทข้อมูลที่เก็บในเขตข้อมูลนี้ ถ้าความยาวไม่ถึง 30 อักขระ ระบบจะเติมช่องว่างให้จนครบ แต่ถ้าเกิน ระบบจะตัดข้อความทางขวาทิ้ง

2.3 เขตความยาวสูงสุดของข้อมูล (Maximum Field Length) ขนาด 4 อักขระ ใช้กำหนดว่าในเขตข้อมูลเก็บข้อมูลได้มากที่สุดกี่ตัวอักษร โดยระบบยอมให้กำหนดความยาวสูงสุดของข้อมูลแต่ละเขตได้ไม่เกิน 1650

2.4 เขตประเภทข้อมูล (Field Type) ขนาด 1 อักขระ เป็นตัวอักษรแสดงข้อมูลที่จะเก็บว่าเป็นประเภทใด ใน 4 ประเภท ได้แก่

X เมื่อข้อมูลเป็นตัวอักษร ตัวเลข หรืออักขระพิเศษ
A เมื่อข้อมูลเป็นพยัญชนะ
N เมื่อข้อมูลเป็นตัวเลข 0 - 9
P เมื่อข้อมูลอยู่ในรูปแบบที่กำหนดให้

2.5 เขตข้อมูลดัชนีแสดงการซ้ำ ขนาด 1 อักขระ เป็นตัวกำหนดว่าเขตข้อมูลนั้น ๆ สามารถเก็บหน่วยข้อมูลได้หลาย ๆ ชุดหรือไม่ ถ้าเป็นช่องว่างแสดงว่าเขตข้อมูลนั้นไม่มีการซ้ำ แต่ถ้าเป็นตัวอักษร 'R' แสดงว่าซ้ำได้ จำนวนครั้งที่ซ้ำได้ไม่จำกัด แต่ในการป้อนข้อมูล ผู้ใช้จะต้องใส่เครื่องหมายเปอร์เซนต์ '%' เพื่อแสดงการซ้ำของข้อมูลแต่ละครั้ง นอกจากนี้ ประเภทของเขตข้อมูลที่ซ้ำได้จะต้องเป็นชนิด 'X'

2.6 เขตสัญญลักษณ์แสดงเขตข้อมูลย่อยหรือรูปแบบการบันทึกข้อมูล (Subfield Delimitor/Pattern) ขนาด 20 อักขระ เป็นตัวอักษรแสดงเขตข้อมูลย่อย หรือเป็นเขตที่ระบุรูปแบบการบันทึกข้อมูล ถ้าเขตข้อมูลใดมีการแบ่งออกเป็นเขตข้อมูลย่อย จะต้องกำหนดตัวอักษรกำกับทุกเขตข้อมูลย่อย และมีเครื่องหมาย '^' นำหน้าตัวอักษรกำกับเขตข้อมูลย่อย แล้วจึงตามด้วยข้อมูล เช่น ^AMr. ^BRobert ^CMartin แสดงว่ามี 3 เขตข้อมูลย่อย (A B และ C เป็นตัวอักษรกำกับเขตข้อมูลย่อย) และประเภทข้อมูลที่จะแบ่งออกให้มีเขตข้อมูลย่อยนั้น จะต้องเป็นชนิด 'X' หรือถ้าเขตข้อมูลใดกำหนดประเภทข้อมูลเป็น P ก็จะต้องมาระบุรูปแบบการบันทึกข้อมูลในเขตนี้

ทุกระเบียนซึ่งแทนการกำหนดโครงสร้าง 1 เขตของฐานข้อมูล จะมีโครงสร้างข้างต้นเหมือน ๆ กัน ระเบียนละบรรทัด ตัวอย่างรายละเอียดโครงสร้างฐานข้อมูลแสดงในภาคผนวก

โครงสร้างข้อมูลของแฟ้มข้อมูลแผ่นงานบันทึกข้อมูล

แฟ้มข้อมูลแผ่นงานบันทึกข้อมูล (pxxxxx.FMT) เป็นแฟ้มข้อมูลระบบที่เก็บรายละเอียดเกี่ยวกับจอภาพที่ใช้ในการบันทึกข้อมูล ว่าต้องการให้แสดงข้อความกำกับเขตข้อมูลแต่ละเขตอย่างไร ที่ตำแหน่งใด ชนิดข้อมูลใด มีข้อความช่วยเหลือสำหรับแต่ละเขตว่าอย่างไร และมีค่ากำหนดล่วงหน้าเท่าใด โครงสร้างแฟ้มข้อมูลประกอบด้วยเขตข้อมูลความยาวแปรผัน โดยแบ่งออกเป็นบรรทัด ในแต่ละบรรทัดประกอบด้วยรายละเอียดดังตาราง 4.4

บรรทัด	เขตข้อมูล	รายละเอียด	ชนิดข้อมูล	หมายเหตุ
1	1	จำนวนเขตข้อมูลทั้งสิ้นในจอภาพ	ตัวเลข	
1	2	รหัสระบุแผ่นงานสุดท้าย	ตัวเลข	ถ้าเป็น 0 มีต่อ ถ้าเป็น 1 จบ
1	3	จำนวนไบท์ของแผ่นงาน	ตัวเลข	
2	1	หมายเลขแถวที่จะแสดงข้อความกำกับเขตข้อมูลแรก	ตัวเลข	
2	2	หมายเลขแถวที่จะแสดงข้อความกำกับเขตข้อมูลที่ 2	ตัวเลข	
2	3	หมายเลขแถวที่จะแสดงข้อความกำกับเขตข้อมูลที่ 3	ตัวเลข	
2	.	หมายเลขแถวที่จะแสดงข้อความกำกับเขตข้อมูลที่ .	ตัวเลข	
2	.	หมายเลขแถวที่จะแสดงข้อความกำกับเขตข้อมูลที่ .	ตัวเลข	
2	.	หมายเลขแถวที่จะแสดงข้อความกำกับเขตข้อมูลที่ .	ตัวเลข	
2	n	หมายเลขแถวที่จะแสดงข้อความกำกับเขตข้อมูลที่ n	ตัวเลข	
3	1	หมายเลขสดมภ์ที่จะแสดงข้อความกำกับเขตข้อมูลแรก	ตัวเลข	
3	2	หมายเลขสดมภ์ที่จะแสดงข้อความกำกับเขตข้อมูลที่ 2	ตัวเลข	
3	3	หมายเลขสดมภ์ที่จะแสดงข้อความกำกับเขตข้อมูลที่ 3	ตัวเลข	
3	.	หมายเลขสดมภ์ที่จะแสดงข้อความกำกับเขตข้อมูลที่ .	ตัวเลข	
3	.	หมายเลขสดมภ์ที่จะแสดงข้อความกำกับเขตข้อมูลที่ .	ตัวเลข	
3	.	หมายเลขสดมภ์ที่จะแสดงข้อความกำกับเขตข้อมูลที่ .	ตัวเลข	
3	n	หมายเลขสดมภ์ที่จะแสดงข้อความกำกับเขตข้อมูลที่ n	ตัวเลข	
4	1	หมายเลขแถวที่จะแสดงเขตข้อมูลแรก	ตัวเลข	
4	2	หมายเลขแถวที่จะแสดงเขตข้อมูลที่ 2	ตัวเลข	
4	3	หมายเลขแถวที่จะแสดงเขตข้อมูลที่ 3	ตัวเลข	
4	.	หมายเลขแถวที่จะแสดงเขตข้อมูลที่ .	ตัวเลข	
4	.	หมายเลขแถวที่จะแสดงเขตข้อมูลที่ .	ตัวเลข	
4	.	หมายเลขแถวที่จะแสดงเขตข้อมูลที่ .	ตัวเลข	
4	n	หมายเลขแถวที่จะแสดงเขตข้อมูลที่ n	ตัวเลข	
5	1	หมายเลขสดมภ์ที่จะแสดงเขตข้อมูลแรก	ตัวเลข	
5	2	หมายเลขสดมภ์ที่จะแสดงเขตข้อมูลที่ 2	ตัวเลข	

ตาราง 4.4 รายละเอียดโครงสร้างแฟ้มข้อมูลตารางบันทึกข้อมูล

บรรทัด	เขตข้อมูล	รายละเอียด	ชนิดข้อมูล	หมายเหตุ
5	3	หมายเลขสดมภ์ที่จะแสดงเขตข้อมูลที่ 3	ตัวเลข	
5	.	หมายเลขสดมภ์ที่จะแสดงเขตข้อมูลที่ .	ตัวเลข	
5	.	หมายเลขสดมภ์ที่จะแสดงเขตข้อมูลที่ .	ตัวเลข	
5	.	หมายเลขสดมภ์ที่จะแสดงเขตข้อมูลที่ .	ตัวเลข	
5	n	หมายเลขสดมภ์ที่จะแสดงเขตข้อมูลที่ n	ตัวเลข	
6	1	ชนิดของการแสดงที่เขตข้อมูลแรก	ตัวเลข	
6	2	ชนิดของการแสดงที่เขตข้อมูลที่ 2	ตัวเลข	
6	3	ชนิดของการแสดงที่เขตข้อมูลที่ 3	ตัวเลข	
6	.	ชนิดของการแสดงที่เขตข้อมูลที่ .	ตัวเลข	
6	.	ชนิดของการแสดงที่เขตข้อมูลที่ .	ตัวเลข	
6	.	ชนิดของการแสดงที่เขตข้อมูลที่ .	ตัวเลข	
6	n	ชนิดของการแสดงที่เขตข้อมูลที่ n	ตัวเลข	
7	1	ประเภทข้อมูลของเขตข้อมูลที่ 1	ตัวเลข	
7	2	ประเภทข้อมูลของเขตข้อมูลที่ 2	ตัวเลข	
7	3	ประเภทข้อมูลของเขตข้อมูลที่ 3	ตัวเลข	
7	.	ประเภทข้อมูลของเขตข้อมูลที่ .	ตัวเลข	
7	.	ประเภทข้อมูลของเขตข้อมูลที่ .	ตัวเลข	
7	.	ประเภทข้อมูลของเขตข้อมูลที่ .	ตัวเลข	
7	n	ประเภทข้อมูลของเขตข้อมูลที่ n	ตัวเลข	
8	1	ความยาวข้อความกำกับเขตข้อมูลแรก	ตัวเลข	
8	2	ความยาวข้อความกำกับเขตข้อมูลที่ 2	ตัวเลข	
8	3	ความยาวข้อความกำกับเขตข้อมูลที่ 3	ตัวเลข	
8	.	ความยาวข้อความกำกับเขตข้อมูลที่ .	ตัวเลข	
8	.	ความยาวข้อความกำกับเขตข้อมูลที่ .	ตัวเลข	
8	.	ความยาวข้อความกำกับเขตข้อมูลที่ .	ตัวเลข	
8	n	ความยาวข้อความกำกับเขตข้อมูลที่ n	ตัวเลข	
9	1	ความยาวเขตข้อมูลแรก	ตัวเลข	

ตาราง 4.4 รายละเอียดโครงสร้างแฟ้มข้อมูลตาราง บันทึกข้อมูล (ต่อ)

บรรทัด	เขตข้อมูล	รายละเอียด	ชนิดข้อมูล	หมายเหตุ
9	2	ความยาวเขตข้อมูลที่ 2	ตัวเลข	
9	3	ความยาวเขตข้อมูลที่ 3	ตัวเลข	
9	.	ความยาวเขตข้อมูลที่ .	ตัวเลข	
9	.	ความยาวเขตข้อมูลที่ .	ตัวเลข	
9	.	ความยาวเขตข้อมูลที่ .	ตัวเลข	
9	n	ความยาวเขตข้อมูลที่ n	ตัวเลข	
10	1	ความยาวข้อความช่วยเหลือเขตข้อมูลแรก	ตัวเลข	
10	2	ความยาวข้อความช่วยเหลือเขตข้อมูลที่ 2	ตัวเลข	
10	3	ความยาวข้อความช่วยเหลือเขตข้อมูลที่ 3	ตัวเลข	
10	.	ความยาวข้อความช่วยเหลือเขตข้อมูลที่ .	ตัวเลข	
10	.	ความยาวข้อความช่วยเหลือเขตข้อมูลที่ .	ตัวเลข	
10	.	ความยาวข้อความช่วยเหลือเขตข้อมูลที่ .	ตัวเลข	
10	n	ความยาวข้อความช่วยเหลือเขตข้อมูลที่ n	ตัวเลข	
11	1	ความยาวค่ากำหนดล่วงหน้าเขตข้อมูลแรก	ตัวเลข	
11	2	ความยาวค่ากำหนดล่วงหน้าเขตข้อมูลที่ 2	ตัวเลข	
11	3	ความยาวค่ากำหนดล่วงหน้าเขตข้อมูลที่ 3	ตัวเลข	
11	.	ความยาวค่ากำหนดล่วงหน้าเขตข้อมูลที่ .	ตัวเลข	
11	.	ความยาวค่ากำหนดล่วงหน้าเขตข้อมูลที่ .	ตัวเลข	
11	.	ความยาวค่ากำหนดล่วงหน้าเขตข้อมูลที่ .	ตัวเลข	
11	n	ความยาวค่ากำหนดล่วงหน้าเขตข้อมูลที่ n	ตัวเลข	
12	1	หมายเลขเขตข้อมูลที่จะแสดงลำดับแรก	ตัวเลข	
12	2	หมายเลขเขตข้อมูลที่จะแสดงลำดับที่ 2	ตัวเลข	
12	3	หมายเลขเขตข้อมูลที่จะแสดงลำดับที่ 3	ตัวเลข	
12	.	หมายเลขเขตข้อมูลที่จะแสดงลำดับที่ .	ตัวเลข	
12	.	หมายเลขเขตข้อมูลที่จะแสดงลำดับที่ .	ตัวเลข	
12	.	หมายเลขเขตข้อมูลที่จะแสดงลำดับที่ .	ตัวเลข	
12	n	หมายเลขเขตข้อมูลที่จะแสดงลำดับที่ n	ตัวเลข	

ตาราง 4.4 รายละเอียดโครงสร้างแฟ้มข้อมูลตาราง บันทึกข้อมูล (ต่อ)

บรรทัด	เขตข้อมูล	รายละเอียด	ชนิดข้อมูล	หมายเหตุ
13	1	ข้อความกำกับเขตข้อมูลแรก	อักขระ	
13	2	ข้อความช่วยเหลือเขตข้อมูลแรก	อักขระ	
13	3	ค่ากำหนดล่วงหน้าเขตข้อมูลแรก	อักขระ	
13	4	ข้อความกำกับเขตข้อมูลที่ 2	อักขระ	
13	5	ข้อความช่วยเหลือเขตข้อมูลที่ 2	อักขระ	
13	6	ค่ากำหนดล่วงหน้าเขตข้อมูลที่ 2	อักขระ	
13	7	ข้อความกำกับเขตข้อมูลที่ 3	อักขระ	
13	8	ข้อความช่วยเหลือเขตข้อมูลที่ 3	อักขระ	
13	9	ค่ากำหนดล่วงหน้าเขตข้อมูลที่ 3	อักขระ	
13	10	ข้อความกำกับเขตข้อมูลที่ .	อักขระ	
13	11	ข้อความช่วยเหลือเขตข้อมูลที่ .	อักขระ	
13	12	ค่ากำหนดล่วงหน้าเขตข้อมูลที่ .	อักขระ	
13	13	ข้อความกำกับเขตข้อมูลที่ n	อักขระ	
13	14	ข้อความช่วยเหลือเขตข้อมูลที่ n	อักขระ	
13	15	ค่ากำหนดล่วงหน้าเขตข้อมูลที่ n	อักขระ	

ตาราง 4.4 รายละเอียดโครงสร้างแฟ้มข้อมูลตารางบันทึกข้อมูล (ต่อ)

ทุกแฟ้มแผ่นงานบันทึกข้อมูลจะมีรายละเอียดข้างต้น เป็นระเบียนที่มีความยาวแปรผัน โดยเขตข้อมูลใดไม่มีข้อมูล ก็จะมีความยาวเขตข้อมูลนั้น ๆ เป็นศูนย์และไม่มีข้อมูลในเขตรายละเอียดข้อมูลเป็นการประหยัดเนื้อที่ นอกจากนี้ เนื่องจากแผ่นงานบันทึกข้อมูลอาจจะแบ่งออกเป็นหลายจอภาพ ซึ่งระบบจัดเก็บแยกไว้ แฟ้มละจอภาพ โดยขึ้นต้นชื่อด้วยอักขระ A, B, C,... ไปเรื่อย ๆ ซึ่งระบบใช้เขตข้อมูลที่ 2 ในบรรทัดที่ 1 เป็นตัวทดสอบ ถ้าเป็น 1 แสดงว่าเป็นแผ่นงานสุดท้าย ถ้าเป็น 0 แสดงว่าไม่ใช่แผ่นงานสุดท้าย ยังมีแผ่นงานต่อไปอีก ตัวอย่างรายละเอียดโครงสร้างแผ่นงานบันทึกข้อมูลแสดงในภาคผนวก

โครงสร้างข้อมูลของแฟ้มข้อมูลตารางเขตข้อมูลที่เลือก

แฟ้มข้อมูลตารางเขตข้อมูลที่เลือก (Field Select Table หรือ FST) เป็นแฟ้มข้อมูลระบบที่ใช้เก็บรายละเอียดวิธีการสร้างคำหลัก (Keyword) เพื่อใช้ในการค้นหาข้อสนเทศ หรือเพื่อกำหนดวิธีการเรียงลำดับข้อมูลที่ค้นหาได้ก่อนจะพิมพ์ข้อมูลนั้นออกไป ระเบียนของแฟ้มข้อมูลแต่ละระเบียนประกอบด้วยเขตข้อมูล 3 เขต คั่นด้วยช่องว่าง และจบระเบียนด้วย <ENTER> (OD_{16} OA_{16}) แต่ละระเบียนจะแทนรูปแบบการสร้างดัชนี 1 เขต ซึ่งมีโครงสร้างข้อมูลดังนี้

1. เขตข้อมูลตัวชี้เขตข้อมูล (Field Identifier) ความยาวสูงสุด 3 อักขระ เป็นหมายเลขเขตข้อมูลที่ต้องการนำมาสร้างเป็นคำหลัก เพื่อการค้นหา มีค่าได้ตั้งแต่ 1 - 999

2. เทคนิคการสร้างดัชนี (Indexing Technique) ขนาด 1 อักขระ เป็นตัวเลขจำนวนเต็ม มีค่าตั้งแต่ 0 - 4 ใช้อ้างอิงวิธีการสร้างคำหลักจากเขตข้อมูล ได้แก่

0 ถ้าใช้ข้อมูลทั้งเขตข้อมูลที่ระบุมาสร้างเป็นคำหลัก

1 ใช้เขตข้อมูลย่อยมาสร้างเป็นคำหลัก (ผู้ใช้ต้องกำหนด MPU ใน Format เสมอ)

2 ใช้ข้อความที่อยู่ระหว่างเครื่องหมาย / มาสร้างเป็นคำหลัก

3 ใช้ข้อความที่อยู่ระหว่างเครื่องหมาย < และ > มาสร้างเป็นคำหลัก

4 ใช้คำทุกคำในเขตข้อมูลมาสร้างเป็นคำหลัก

3. รูปแบบ (Format) เป็นวิธีการดึงข้อมูลจากฐานข้อมูล คำสั่งในรูปแบบนี้จะอยู่ในลักษณะที่เป็นภาษาในการจัดรูปแบบการพิมพ์

ตัวอย่างแฟ้มข้อมูลตารางเลือกเขตข้อมูลดังแสดงในภาคผนวก

โครงสร้างข้อมูลของแฟ้มข้อมูลรูปแบบการแสดงผลลัพธ์บนจอภาพ

แฟ้มข้อมูลรูปแบบการแสดงผลลัพธ์บนจอภาพ (Display Format หรือ PFT) เป็นแฟ้มข้อมูลระบบที่ใช้เก็บรายละเอียดรูปแบบการแสดงข้อมูลที่ได้จากการค้นหา หรือได้จากการดึงข้อมูลจากฐานข้อมูลโดยตรง มีลักษณะโครงสร้างเป็นกลุ่มของอักขระที่เป็นภาษาในการจัดรูปแบบการพิมพ์ เก็บในรูปแฟ้มแอสกีธรรมดา ถูกเรียกใช้โดยโปรแกรม ISIS.EXE โดยระบุเครื่องหมาย @ นำหน้าชื่อแฟ้มข้อมูลรูปแบบการแสดงผลลัพธ์นี้ เมื่อเลือกทางเลือก F ระบบจะไปเปิดแฟ้มนี้ อ่านข้อมูลเก็บใส่หน่วยความจำ แล้วแสดงข้อมูลตามรูปแบบที่ระบุในแฟ้ม ตัวอย่างแฟ้มข้อมูลดังแสดงในภาคผนวก

โครงสร้างข้อมูลของแฟ้มข้อมูลหลัก

แฟ้มข้อมูลหลัก (Master File หรือ MST) เป็นแฟ้มข้อมูลที่เก็บระเบียนข้อมูลทั้งหมดของฐานข้อมูลแต่ละฐานข้อมูลที่ผู้ใช้ระบบบันทึก โครงสร้างของแฟ้มข้อมูลหลักประกอบด้วยเขตข้อมูลความยาวแปรผัน โดยถูกออกแบบให้เขตข้อมูลหนึ่ง ๆ สามารถแบ่งออกเป็นเขตข้อมูลย่อยได้ โดยมีเครื่องหมายแสดงเขตข้อมูลย่อยนำหน้าหน่วยข้อมูล นอกจากนั้น ยังมีสามารถมีเขตข้อมูลซ้ำกันได้ แต่ละเขตข้อมูลถูกอ้างอิงโดยใช้เลขต่อท้าย (Tag) ซึ่งเป็นเลขจำนวนเต็ม มีค่าตั้งแต่ 1 - 999 ใช้เป็นเสมือนชื่อเขตข้อมูลสำหรับอ้างอิง

เขตข้อมูลแต่ละระเบียนของแฟ้มข้อมูลหลักจะมีเขตข้อมูลพิเศษอันหนึ่ง เป็นตัวเลขกำกับระเบียน ซึ่งจะมีค่าไม่ซ้ำกับระเบียนอื่น ๆ มีไว้อ้างอิงถึงระเบียนนั้น ๆ เรียกว่า เลขแฟ้มข้อมูลหลัก (Master File Number หรือ MFN) เป็นเลขจำนวนเต็ม มีค่าตั้งแต่ 1 ถึง 32767 โดยระบบจะเป็นผู้กำหนดเลขแฟ้มข้อมูลหลักให้เองเมื่อมีการบันทึกข้อมูลเข้าไปเก็บในฐานข้อมูล สำหรับเขตข้อมูลอื่น ๆ ที่ประกอบขึ้นเป็นระเบียนนั้น ก็เป็นเขตข้อมูลที่มีความยาวแปรผันได้

โครงสร้างข้อมูลของแฟ้มข้อมูลหลักสามารถแบ่งออกได้เป็น 3 ส่วนใหญ่ ๆ ได้แก่ ส่วนหัวของแฟ้มข้อมูลหลัก, ส่วนหัวของระเบียน และส่วนระเบียน

1. ส่วนหัวของแฟ้มข้อมูลหลัก

ที่ต้นแฟ้มข้อมูลหลักของทุกฐานข้อมูล ระบบจะจัดเตรียมเนื้อที่ไว้จำนวน 18 ไบท์ เพื่อเก็บข้อมูลเกี่ยวกับแฟ้ม ดังในตาราง 4.5

ไบท์ที่	ข้อมูลในรูปเลขฐานสิบหก	ความหมาย
1 - 2	0000	หมายเลขบล็อค
3 - 4	0000 จนถึง FF7F	หมายเลขแฟ้มข้อมูลหลักลำดับถัดไป
5 - 6	ตั้งแต่ 0100 ไปเรื่อย ๆ	ขนาดของแฟ้มหน่วยเป็นเซกเตอร์
7 - 8	ตั้งแต่ 1300 ไปเรื่อย ๆ	ตำแหน่งท้ายแฟ้มที่จะบันทึกข้อมูลถัดไป
9 - 10	0000	
11 - 12	ตั้งแต่ 0000 ไปเรื่อย ๆ	จำนวนระเบียนที่อ่านจากการสำเนา
13 - 14	0000	
15 - 16	ตั้งแต่ 0000 ไปเรื่อย ๆ	จำนวนบล็อคจากการสำเนา
17 - 18	0000	

ตาราง 4.5 แสดงรายละเอียดส่วนหัวของแฟ้มข้อมูลหลัก

ตัวอย่างรายละเอียดของโครงสร้างส่วนหัวของแฟ้มข้อมูลหลัก ดังแสดงในภาคผนวก

2. ส่วนหัวของระเบียน

ถัดจากส่วนหัวของแฟ้มข้อมูลหลัก จะประกอบด้วยส่วนหัวของระเบียน ซึ่งแบ่งออกเป็น 7 เขตข้อมูล ความยาวคงที่ในทุกระเบียน แต่ละเขตมีขนาด 2 ไบท์ ดังมีรายละเอียดดังในตาราง 4.6

ไบท์ที่	ข้อมูลในรูปเลขฐานสิบหก	ความหมาย
1 - 2	0000 จนถึง FF7F	หมายเลขแฟ้มข้อมูลหลักของระเบียน
3 - 4	ตั้งแต่ 0100	ความยาวระเบียนทั้งสิ้น
5 - 6	ตั้งแต่ 0000	เซกเตอร์ของตำแหน่งระเบียนเดิม
7 - 8	ตั้งแต่ 0000	ตำแหน่งระเบียนเดิมภายในเซกเตอร์
9 - 10	ตั้งแต่ 0000	ตำแหน่งฐานของข้อมูลจากต้นระเบียน
11 - 12	ตั้งแต่ 0100	จำนวนเขตของระเบียนนั้น ๆ
13 - 14	0000 หรือ 0100	รหัสระบุว่าระเบียนถูกลบหรือไม่ ถ้าเป็น 0100 แสดงว่าถูกลบ

ตาราง 4.6 แสดงรายละเอียดส่วนหัวของระเบียน

3. <u>ส่วนระเบียน</u>

ถัดจากส่วนหัวของระเบียน จะตามด้วยส่วนระเบียน ซึ่งเป็นกลุ่มของเขตข้อมูลซึ่งมีความยาวแปรผันตามจำนวนเขตที่บันทึกในระเบียนนั้น ๆ นอกจากนี้ ข้อมูลในแต่ละเขตก็ยังมีความยาวแปรผันได้อีกด้วย ส่วนระเบียนประกอบด้วยเขต 2 กลุ่มใหญ่ ๆ ได้แก่ กลุ่มระบุเขต และกลุ่มข้อมูล

3.1 กลุ่มระบุเขต เป็นกลุ่มของเขต 3 เขตที่มีความยาวคงที่ คือ เขตละ 2 ไบท์ รวมเป็น 6 ไบท์ต่อกลุ่ม แต่ละเขตมีรายละเอียดดังในตาราง 4.7

ไบท์ที่	ข้อมูลในรูปเลขฐานสิบหก	ความหมาย
1 - 2	0100 จนถึง E703	หมายเลขต่อท้ายของเขต
3 - 4	ตั้งแต่ 0000	ค่าออฟเซทจากตำแหน่งฐานของข้อมูลจากต้นระเบียน
5 - 6	ตั้งแต่ 0100	ความยาวของเขตข้อมูล

ตาราง 4.7 แสดงรายละเอียดกลุ่มระบุเขต

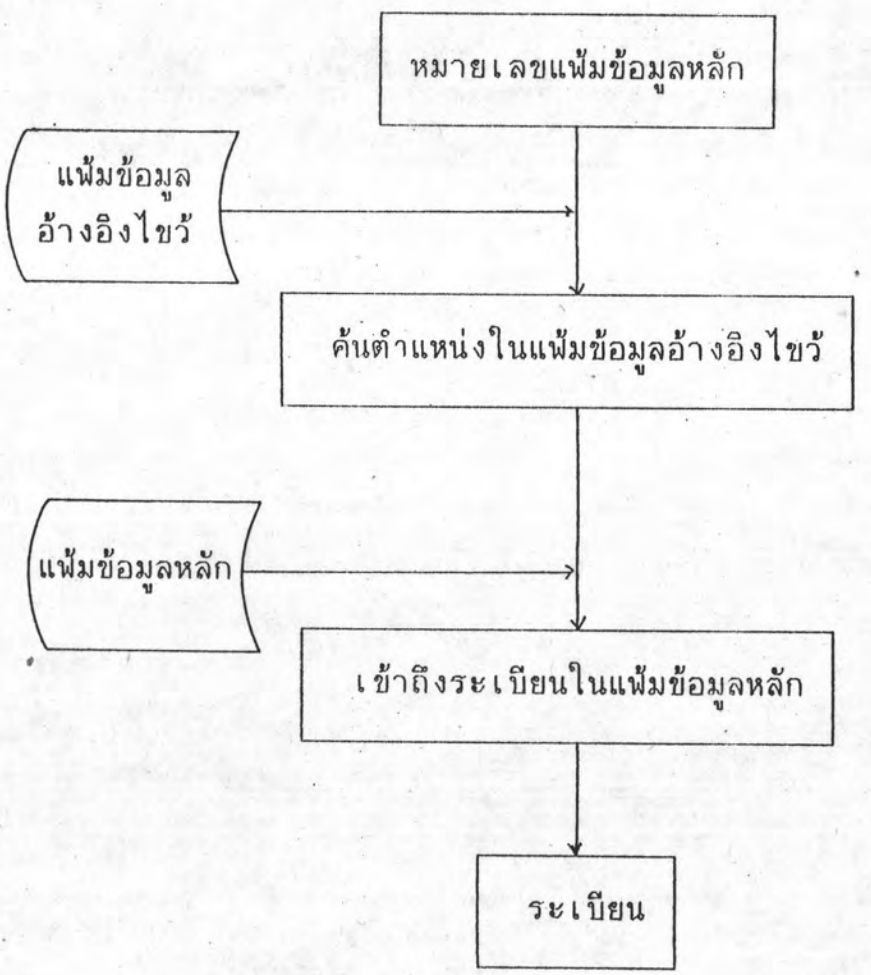
ถ้าระเบียนประกอบด้วย 4 เขต ก็จะมีกลุ่มระบุเขต 4 กลุ่ม ขนาดกลุ่มละ 6 ไบท์อยู่เรียงต่อกัน รวมเป็น 24 ไบท์ เป็นต้น ตัวอย่างรายละเอียดของกลุ่มระบุเขตดังแสดงในภาคผนวก

3.2 กลุ่มข้อมูล อยู่ต่อจากกลุ่มระบุเขต เป็นชุดของอักขระต่อเนื่องกันไปเรื่อย ๆ จนจบระเบียน ระบบสามารถแยกข้อมูลแต่ละเขตได้โดยทำการวิเคราะห์จากหัวระเบียนและกลุ่มระบุเขต ดังนั้น กลุ่มข้อมูลนี้จึงเป็นเขตที่มีความยาวแปรผัน และข้อมูลจะถูกบันทึกในแฟ้มข้อมูลหลักในลักษณะเรียงลำดับต่อท้ายแฟ้มไปเรื่อย ๆ โดยลักษณะการจัดเก็บเขตข้อมูลก็ไม่ได้เรียงลำดับจากหมายเลขต่อท้ายน้อยไปมาก แต่ขึ้นอยู่กับลำดับก่อนหลังของการบันทึกข้อมูล และไม่จำกัดจำนวนเขต คือ ระเบียนใดไม่มีการบันทึกเขตใด ๆ ระบบก็จะไม่บันทึกกลุ่มระบุเขตของเขตนั้น ๆ ไว้ในแฟ้ม จึงเป็นการประหยัดเนื้อที่ในหน่วยความจำ

เนื่องจากโครงสร้างแฟ้มข้อมูลหลักเป็นแบบเรียงลำดับ ระบบจึงจัดสร้างแฟ้มข้อมูลอ้างอิงไขว้เพื่อเป็นแฟ้มดัชนีในการเข้าถึงระเบียนข้อมูลในแฟ้มข้อมูลหลักได้อย่างรวดเร็ว

โครงสร้างข้อมูลของแฟ้มข้อมูลอ้างอิงไขว้

แฟ้มข้อมูลอ้างอิงไขว้ (Cross Reference File หรือ XRF) เป็นแฟ้มข้อมูลที่เก็บตำแหน่งที่อยู่ของระเบียนต่าง ๆ ในแฟ้มข้อมูลหลัก เพื่อให้การเข้าถึงระเบียนข้อมูลโดยอาศัยหมายเลขแฟ้มข้อมูลหลักทำได้รวดเร็ว นั่นคือ เป็นเสมือนดัชนีในการค้นระเบียนข้อมูลโดยใช้หมายเลขแฟ้มข้อมูลหลักเป็นคีย์นั่นเอง เวลาต้องการจะเข้าถึงระเบียนใด ก็จะต้องระบุหมายเลขแฟ้มข้อมูลหลัก จากนั้น ระบบจะมาค้นหาในแฟ้มข้อมูลอ้างอิงไขว้ เพื่อหาตำแหน่งที่อยู่จริงของระเบียนนั้น ๆ ในแฟ้มข้อมูลหลัก แล้วจึงไปอ่านระเบียนจากแฟ้มข้อมูลหลัก ตามตำแหน่งที่ค้นได้ นำมาใช้งาน ตัวอย่างกลไกการเข้าถึงระเบียนโดยใช้แฟ้มข้อมูลอ้างอิงไขว้เป็นดังภาพ 4.16



ภาพ 4.16 แสดงกลไกการเข้าถึงระเบียนโดยใช้แฟ้มข้อมูลอ้างอิงไขว้

โครงสร้างของแฟ้มข้อมูลอ้างอิงไขว้ ประกอบด้วยระเบียนความยาวคงที่ แต่ละระเบียนประกอบด้วย 2 เขตข้อมูล ๆ ละ 2 ไบท์ รวมเป็นระเบียนละ 4 ไบท์ แต่ละระเบียนถูกกำหนดตำแหน่งตายตัวสำหรับเก็บตำแหน่งระเบียนที่อ้างถึงในแฟ้มข้อมูลหลักในลักษณะเรียงลำดับตามหมายเลขแฟ้มข้อมูลหลักจากน้อยไปมาก ลักษณะการประมวลผลแฟ้มข้อมูลอ้างอิงไขว้เป็นลักษณะการเข้าถึงโดยตรง เพื่อให้เข้าถึงข้อมูลได้รวดเร็ว มีรายละเอียดของแต่ละระเบียนดังตาราง 4.8

ไบท์ที่	ข้อมูลในรูปเลขฐานสิบหก	ความหมาย
1 - 2	0100	ส่วนหัวของแฟ้มข้อมูลอ้างอิงไขว้ ชี้ไปยังเซกเมนต์แรกของแฟ้มข้อมูลหลัก
3 - 4	0100	ส่วนหัวของแฟ้มข้อมูลอ้างอิงไขว้ ชี้ไปยังตำแหน่งแรกของเซกเมนต์
5 - 6	ตั้งแต่ 0000	เซกเตอร์ของระเบียน MFN # 1
7 - 8	ตั้งแต่ 0000	ตำแหน่งของระเบียน MFN # 1
9 - 10	ตั้งแต่ 0000	เซกเตอร์ของระเบียน MFN # 2
11 - 12	ตั้งแต่ 0000	ตำแหน่งของระเบียน MFN # 2
13 - 14	ตั้งแต่ 0000	เซกเตอร์ของระเบียน MFN # 3
15 - 16	ตั้งแต่ 0000	ตำแหน่งของระเบียน MFN # 3
17 - 18	ตั้งแต่ 0000	เซกเตอร์ของระเบียน MFN # 4
19 - 20	ตั้งแต่ 0000	ตำแหน่งของระเบียน MFN # 4
21 - 22	ตั้งแต่ 0000	เซกเตอร์ของระเบียน MFN # .
23 - 24	ตั้งแต่ 0000	ตำแหน่งของระเบียน MFN # .
25 - 26	ตั้งแต่ 0000	เซกเตอร์ของระเบียน MFN # .
27 - 28	ตั้งแต่ 0000	ตำแหน่งของระเบียน MFN # .
29 - 30	ตั้งแต่ 0000	เซกเตอร์ของระเบียน MFN # .
31 - 32	ตั้งแต่ 0000	ตำแหน่งของระเบียน MFN # .

ตาราง 4.8 แสดงรายละเอียดระเบียนของแฟ้มข้อมูลอ้างอิงไขว้

ในกรณีที่มีการลบระเบียนข้อมูล ระบบจะยังไม่ลบระเบียนข้อมูลออกจากแฟ้มจริง แต่จะกำหนดให้ไบท์ที่ 13 - 14 ในส่วนหัวระเบียนของแฟ้มข้อมูลหลักให้มีค่าเป็น 0100 และกำหนดให้เขตข้อมูลเซกเตอร์ของระเบียนนั้น ๆ ในแฟ้มข้อมูลอ้างอิงไขว้มีค่าเป็น FFFF ซึ่งผู้ใช้สามารถเรียกข้อมูลที่ลบกลับคืนมาได้ ซึ่งระบบจะเปลี่ยนค่าไบท์ที่ 13 - 14 ให้เป็น 0000 และกำหนดให้เขตข้อมูลเซกเตอร์ของระเบียนนั้น ๆ เก็บค่าเซกเมนต์ของระเบียนตามเดิม

ในกรณีที่มีการปรับปรุงแก้ไขระเบียน ระบบจะทำการบันทึกระเบียนที่ผ่านการปรับปรุงแก้ไขไว้ที่ท้ายแฟ้มข้อมูลหลัก แล้วมาปรับปรุงระเบียนในแฟ้มข้อมูลอ้างอิงไขว้ให้ชี้ไปยังตำแหน่งใหม่ที่มีระเบียนที่แก้ไขแล้วบันทึกอยู่แทน ซึ่งจะทำให้ขนาดของแฟ้มข้อมูลหลักใหญ่ขึ้นเรื่อย ๆ แต่ขนาดของแฟ้มข้อมูลอ้างอิงไขว้จะเท่าเดิมถ้าไม่มีการเพิ่มข้อมูลเข้าไป ซึ่งผู้ใช้ระบบสามารถลดขนาดของแฟ้มข้อมูลหลักลงได้โดยทำการจัดแฟ้มข้อมูลหลักใหม่ โดยเลือกทางเลือก C ในโปรแกรม ISISXCH.EXE ซึ่งระบบจะจัดทำสำเนาแฟ้มข้อมูลหลัก โดยอ่านเฉพาะข้อมูลที่ปรับปรุงล่าสุดและข้อมูลที่ไม่ได้ถูกลบอ่านกลับมาใส่แฟ้มข้อมูลหลักอีกครั้งหนึ่ง ตัวอย่างรายละเอียดของแฟ้มข้อมูลอ้างอิงไขว้ดังแสดงในภาคผนวก

โครงสร้างข้อมูลของแฟ้มข้อมูลหกกลับ

แฟ้มข้อมูลหกกลับ (Inverted File) ของโปรแกรมมินิ-ไมโคร ซีดีเอส/ไอซิส เป็นกลุ่มของแฟ้มข้อมูลระบบจำนวน 6 แฟ้ม ที่ทำหน้าที่จัดเก็บค่าข้อมูลที่ใช้เป็นคำหลักที่เข้าถึงระเบียนข้อมูลที่ต้องการในฐานข้อมูล ด้วยวิธีการอื่น ๆ นอกเหนือจากการเข้าถึงโดยทราบหมายเลขแฟ้มข้อมูลหลักโดยตรง นั่นคือ ผู้ใช้สามารถเข้าถึงระเบียนได้โดยอาจจะระบุ ค่าในเขตข้อมูล, เขตข้อมูลย่อย หรือหน่วยข้อมูลใด ๆ ตลอดจนสามารถเลือกคำศัพท์, วลี หรือข้อความใด ๆ ในระเบียนมาสร้างเป็นคำหลักเพื่อการค้นหา โดยกำหนดค่าที่ต้องการในตารางเขตข้อมูลที่เลือก ซึ่งระบบจะอ่านข้อมูลจากแฟ้มข้อมูลหลัก เลือกคำตามกฎเกณฑ์ที่กำหนด จัดสร้างแฟ้มข้อมูลหกกลับเพื่อให้ผู้ใช้สามารถค้นหาข้อสนเทศต่าง ๆ ในฐานข้อมูลได้อย่างสะดวกและรวดเร็ว ลักษณะโครงสร้างภายนอกของระเบียนในแฟ้มข้อมูลหกกลับประกอบด้วย 2 ส่วน คือ คำหลัก และ โพสติ้ง

1. <u>คำหลัก</u> เป็นคำหรือข้อความที่ระบบนำมาจากระเบียนในแฟ้มข้อมูลหลักตามกฎเกณฑ์ที่ระบุในตารางเขตข้อมูลที่เลือก เป็นคีย์ในการค้นหาข้อมูลจากแฟ้มข้อมูลหลักนั่นเอง มีความยาวสูงสุดไม่เกิน 30 อักขระ

2. <u>โพสติ้ง</u> เป็นรายละเอียดของระเบียนต่าง ๆ ในฐานข้อมูลที่มีคำหลักอยู่ ประกอบด้วย 4 เขตข้อมูลย่อย ได้แก่

2.1 หมายเลขแฟ้มข้อมูลหลักของระเบียนที่พบคำหลัก

2.2 เลขต่อท้าย (Tag) ระบุเขตที่พบคำหลัก

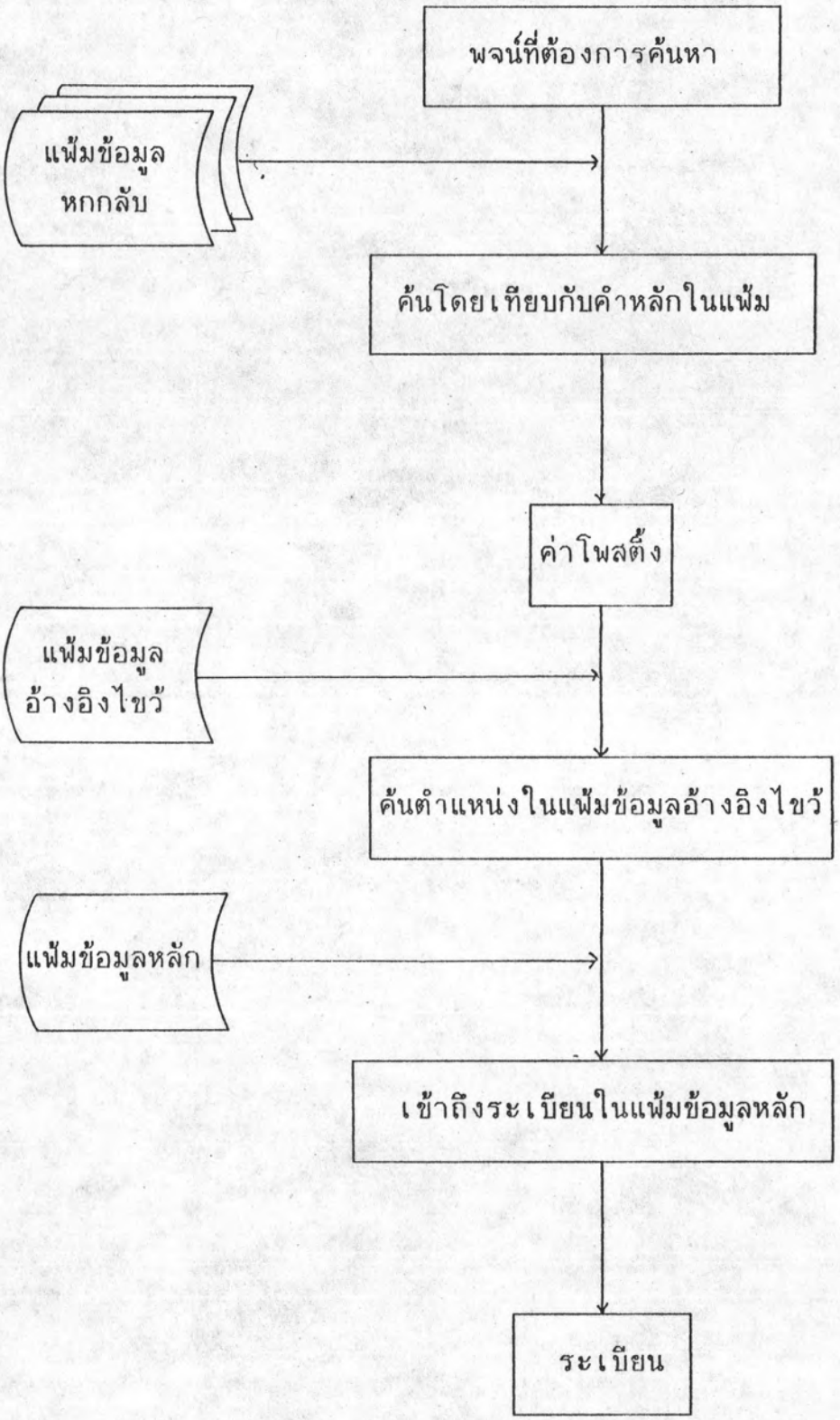
2.3 หมายเลขระบุเขตข้อมูลย่อย ระบุว่าพบคำหลักในเขตข้อมูลย่อยที่เท่าใด

2.4 หมายเลขลำดับพจน์ (Term Sequence Number) ระบุว่าพบคำหลักเป็นพจน์ที่เท่าใดจากต้นเขตข้อมูลที่ระบุ

เมื่อผู้ใช้ต้องการค้นพจน์ใด ระบบจะนำพจน์ที่ต้องการค้นมาค้นในแฟ้มข้อมูลหกกลับโดยเทียบกับคำหลัก เมื่อพบแล้วก็จะนำค่าโพสติ้งไปค้นตำแหน่งของระเบียนที่พบจากแฟ้มข้อมูลอ้างอิงไขว้ แล้วจึงเข้าถึงระเบียนจากแฟ้มข้อมูลหลักนำข้อมูลไปใช้งานต่อไป ดังตัวอย่างในภาพที่ 4.17

การจัดสร้างแฟ้มข้อมูลหกกลับของโปรแกรมมินิ–ไมโครซีดีเอส/ไอซิส แตกต่างจากการจัดสร้างแฟ้มข้อมูลหกกลับทั่ว ๆ ไป ในลักษณะที่ในฐานข้อมูลหนึ่ง ๆ จะมีแฟ้มข้อมูลหกกลับกลุ่มเดียว ซึ่งจะจัดเก็บคำหลักที่ถูกกำหนดด้วยวิธีใด ๆ จากเขตข้อมูลใด ๆ ของแฟ้มข้อมูลหลักตามที่กำหนดไว้ในตารางเขตข้อมูลที่เลือก โดยระบบจะเก็บคำหลักต่าง ๆ ที่ตรงตามเงื่อนไขไว้ในแฟ้มข้อมูล เรียงลำดับจากค่าแอสกีน้อยไปมาก โดยใช้โครงสร้างบี*ทรี จำนวน 2 ทรี โดย ทรีแรกใช้สำหรับจัดเก็บคำหลักหรือพจน์ที่มีความยาวไม่เกิน 10 อักขระ และทรีที่สองใช้สำหรับจัดเก็บคำหลักที่มีความยาวเกิน 10 อักขระ

โครงสร้างภายในของแฟ้มข้อมูลหกกลับ แบ่งออกเป็น 6 แฟ้ม ได้แก่ แฟ้มข้อมูลควบคุมพจนานุกรมพจน์ที่ใช้ค้นหา, แฟ้มข้อมูลบี*ทรีโหนด สำหรับพจน์ที่ความยาวไม่เกิน 10 อักขระ, แฟ้มข้อมูลบี*ทรีลีฟ สำหรับพจน์ที่ความยาวไม่เกิน 10 อักขระ, แฟ้มข้อมูลบี*ทรีโหนด สำหรับพจน์ที่ความยาวเกิน 10 อักขระ, แฟ้มข้อมูลบี*ทรีลีฟ สำหรับพจน์ที่ความยาวเกิน 10 อักขระ และแฟ้มข้อมูลโพสติ้งของแฟ้มข้อมูลหกกลับ



ภาพ 4.17 แสดงกลไกการเข้าถึงระเบียนโดยใช้แฟ้มข้อมูลหกกลับ

1. แฟ้มข้อมูลควบคุมพจนานุกรมพจน์ที่ใช้ค้นหา

แฟ้มข้อมูลควบคุมพจนานุกรมพจน์ที่ใช้ค้นหา (B*Tree (Search term dictionary) Control File หรือแฟ้ม xxxxxx.CNT) เป็นแฟ้มข้อมูลที่จัดเก็บรายละเอียดเกี่ยวกับบี*ทรี 2 ทรี ที่เป็นที่เก็บพจน์ที่ใช้ค้นหา เป็นแฟ้มที่มีโครงสร้างแบบเรียงลำดับ ชนิดระเบียนมีความยาวคงที่ ประกอบด้วย 2 ระเบียน ขนาดระเบียนละ 20 ไบท์ ระเบียนแรกระบุรายละเอียดของทรีสำหรับจัดเก็บคำหลักหรือพจน์ที่มีความยาวไม่เกิน 10 อักขระ ส่วนระเบียนที่สองระบุรายละเอียดของทรีสำหรับจัดเก็บพจน์ที่มีความยาวเกิน 10 อักขระ โครงสร้างข้อมูลของแต่ละระเบียนมีรายละเอียดดังในตาราง 4.9

ไบท์ที่	ข้อมูลในรูปเลขฐานสิบหก	ความหมาย
1 - 2	0100 หรือ 0200	ระบุว่าเป็นทรีที่หนึ่ง หรือทรีที่สอง
3 - 4	0500	
5 - 6	0500	
7 - 8	0A00	
9 - 10	0400	
11 - 12	0000 หรือ FFFF	ระบุว่ามีข้อมูลในทรีหรือไม่
13 - 14	ตั้งแต่ 0000	ระบุจำนวนข้อมูลในทรี
15 - 16	ตั้งแต่ 0000	ระบุจำนวนข้อมูลในทรี
17 - 18	ตั้งแต่ 0000	ระบุจำนวนข้อมูลในทรี
19 - 20	ตั้งแต่ 0000	ระบุจำนวนข้อมูลในทรี

ตาราง 4.9 แสดงรายละเอียดระเบียนของแฟ้มข้อมูลควบคุมพจนานุกรมพจน์ที่ใช้ค้นหา

ในการใช้งานแฟ้มข้อมูลหกกลับ ระบบจะมาตรวจสอบข้อมูลในแฟ้มนี้ก่อนเสมอว่ามีพจน์อยู่ในทรีที่หนึ่งหรือสองหรือไม่ โดยดูจากไบท์ที่ 11 - 12 ถ้าเป็น FFFF แสดงว่าไม่มีข้อมูลในทรีนั้น ๆ ตัวอย่างรายละเอียดโครงสร้างของแฟ้ม ดังแสดงในภาคผนวก

2. แฟ้มข้อมูลบี*ทรีโหนดสำหรับพจน์ที่ความยาวไม่เกิน 10 อักขระ

แฟ้มข้อมูลบี*ทรีโหนดสำหรับพจน์ที่ความยาวไม่เกิน 10 อักขระ (xxxxxx.NO1) เป็นแฟ้มข้อมูลที่เก็บพจน์ที่เป็นโหนดกิ่ง (Branch Node) และเก็บตัวชี้ไปยังโหนดใบ (Leaf Node) ที่เก็บตำแหน่งของโพสติ้งของพจน์ที่ความยาวไม่เกิน 10 อักขระ โดยระบบจะใช้แฟ้มนี้ร่วมกับแฟ้มข้อมูลบี*ทรีลีฟสำหรับพจน์ที่ความยาวไม่เกิน 10 อักขระ ซึ่งเป็นที่เก็บคำหลักและตำแหน่งของโพสติ้ง เวลาระบบจะค้นหาข้อมูล แทนที่จะเข้าไปค้นตั้งแต่ต้นทรีในแฟ้มบี*ทรีลีฟ ระบบก็จะมาเริ่มค้นที่แฟ้มนี้ก่อนเพื่อดูว่าข้อมูลที่จะค้นตกอยู่ในช่วงใดของทรีโดยเทียบกับพจน์ที่เป็นโหนดกิ่ง ดูว่าชี้ไปยังโหนดใบกลุ่มใด จากนั้นระบบจึงไปค้นในกลุ่มของโหนดใบที่ตัวชี้ชี้มา เพื่อเข้าถึงระเบียนซึ่งเก็บตำแหน่งของโพสติ้งอีกทีหนึ่ง แล้วจึงไปเข้าถึงโพสติ้งในแฟ้มข้อมูลโพสติ้งของแฟ้มข้อมูลหกกลับ (xxxxxx.IFP)

โครงสร้างข้อมูลของแฟ้มข้อมูลบี*ทรีโหนดสำหรับพจน์ที่มีความยาวไม่เกิน 10 อักขระ เป็นแฟ้มข้อมูลแบบเรียงลำดับชนิดระเบียนความยาวคงที่ ประกอบด้วยกลุ่มของระเบียนที่มีโครงสร้างเหมือนกัน คือประกอบด้วยกลุ่มของพจน์ที่เป็นโหนดกิ่ง กลุ่มละ ไม่เกิน 10 พจน์ โดยระบบจองเนื้อที่ไว้ตายตัว 10 ระเบียน เรียงลำดับตามค่าแอสกีจากน้อยไปมากในแต่ละกลุ่ม ซึ่งมีหมายเลขกำกับ แต่ละพจน์จะมีตัวชี้ในลักษณะลิงค์ลิสต์ ชี้ไปยังบล็อคและออฟเซทของแฟ้มข้อมูลบี*ทรีลีฟที่เก็บกลุ่มคำหลักที่มีค่าแอสกีมากกว่าหรือเท่ากับพจน์นั้นแต่ไม่เกินพจน์ถัดไป ดังนั้น เวลาระบบค้นหาคำหลักก็จะนำพจน์มาเปรียบเทียบกับระเบียนแรกของแฟ้มนี้ ถ้าค่าแอสกีของพจน์ที่จะค้นน้อยกว่าค่าแอสกีของพจน์ที่เป็นโหนดกิ่งที่สอง แสดงว่าข้อมูลอยู่ในโหนดกิ่งแรก ระบบจะใช้ค่าตัวชี้ของโหนดกิ่งแรกเพื่อไปค้นในแฟ้มที่เก็บโหนดใบต่อไป แต่ถ้าค่าที่จะค้นเท่ากับโหนดกิ่งที่สอง แสดงว่าข้อมูลอยู่ในโหนดกิ่งที่สอง แต่ถ้าค่าที่จะค้นมากกว่าโหนดกิ่งที่สอง ระบบก็จะเทียบกับระเบียนที่เป็นโหนดกิ่งถัดไปเรื่อย ๆ จนกว่าค่าแอสกีของพจน์ที่จะค้นเท่ากับ แสดงว่าค่าที่ต้องการอยู่ในโหนดใบของพจน์นั้น ๆ หรือจนกว่าค่าแอสกีของพจน์ที่จะค้นน้อยกว่าค่าในโหนดกิ่ง แสดงว่าพจน์ที่จะค้นอยู่ในโหนดก่อนหน้า ระบบก็จะไปยังแฟ้มข้อมูลบี*ทรีลีฟ ตามค่าตัวชี้ แล้วค้นหาต่อไป

โครงสร้างข้อมูลของแต่ละกลุ่มระเบียนประกอบด้วย 23 เขตข้อมูล ความยาวคงที่ ประกอบด้วยตั้งแต่ 1 กลุ่มระเบียนขึ้นไปอยู่ต่อเนื่องกันแบบเรียงลำดับ มีรายละเอียดของแต่ละกลุ่มระเบียนดังในตาราง 4.10

ไบท์ที่	ข้อมูลในรูปเลขฐานสิบหก	ความหมาย
1 - 2	ตั้งแต่ 0100	ระบุหมายเลขกลุ่มของพจน์
3 - 4	0100 - 0A00	ระบุจำนวนพจน์ในกลุ่ม
5 - 6	0100	
7 - 16	20 หรือ พจน์	เก็บพจน์ที่เป็นโหนดกิ่งลำดับ 1
17 - 18	0000 หรือ FFFF ลงมา	เก็บค่าตัวชี้ของพจน์ลำดับ 1
19 - 28	20 หรือ พจน์	เก็บพจน์ที่เป็นโหนดกิ่งลำดับ 2
29 - 30	0000 หรือ FFFF ลงมา	เก็บค่าตัวชี้ของพจน์ลำดับ 2
31 - 40	20 หรือ พจน์	เก็บพจน์ที่เป็นโหนดกิ่งลำดับ 3
41 - 42	0000 หรือ FFFF ลงมา	เก็บค่าตัวชี้ของพจน์ลำดับ 3
43 - 52	20 หรือ พจน์	เก็บพจน์ที่เป็นโหนดกิ่งลำดับ 4
53 - 54	0000 หรือ FFFF ลงมา	เก็บค่าตัวชี้ของพจน์ลำดับ 4
55 - 64	20 หรือ พจน์	เก็บพจน์ที่เป็นโหนดกิ่งลำดับ 5
65 - 66	0000 หรือ FFFF ลงมา	เก็บค่าตัวชี้ของพจน์ลำดับ 5
67 - 76	20 หรือ พจน์	เก็บพจน์ที่เป็นโหนดกิ่งลำดับ 6
77 - 78	0000 หรือ FFFF ลงมา	เก็บค่าตัวชี้ของพจน์ลำดับ 6
79 - 88	20 หรือ พจน์	เก็บพจน์ที่เป็นโหนดกิ่งลำดับ 7
89 - 90	0000 หรือ FFFF ลงมา	เก็บค่าตัวชี้ของพจน์ลำดับ 7
91 -100	20 หรือ พจน์	เก็บพจน์ที่เป็นโหนดกิ่งลำดับ 8
101 -102	0000 หรือ FFFF ลงมา	เก็บค่าตัวชี้ของพจน์ลำดับ 8
103- 112	20 หรือ พจน์	เก็บพจน์ที่เป็นโหนดกิ่งลำดับ 9
113- 114	0000 หรือ FFFF ลงมา	เก็บค่าตัวชี้ของพจน์ลำดับ 9
115- 124	20 หรือ พจน์	เก็บพจน์ที่เป็นโหนดกิ่งลำดับ 10
125- 126	0000 หรือ FFFF ลงมา	เก็บค่าตัวชี้ของพจน์ลำดับ 10

ตาราง 4.10 แสดงรายละเอียดแต่ละกลุ่มระเบียนของแฟ้มข้อมูลบี*ทรีโหนดสำหรับพจน์ที่ความยาวไม่เกิน 10 อักขระ

ตัวอย่างรายละเอียดของแฟ้มข้อมูลบี*ทรีโหนดสำหรับพจน์ที่ความยาวไม่เกิน 10 อักขระ ดังแสดงในภาคผนวก

3. แฟ้มข้อมูลบี*ทรีลีฟสำหรับพจน์ที่ความยาวไม่เกิน 10 อักขระ

แฟ้มข้อมูลบี*ทรีลีฟสำหรับพจน์ที่ความยาวไม่เกิน 10 อักขระ (xxxxxx.L01) เป็นแฟ้มข้อมูลที่เก็บคำหลักและตำแหน่งโพสติ้งของคำหลัก ซึ่งเมื่อระบบค้นตำแหน่งตัวชี้จากแฟ้มข้อมูลบี*ทรีโหนดได้แล้ว ระบบจะมาทำการค้นในแฟ้มนี้จนพบหรือไม่พบคำหลักที่ระบุ โครงสร้างข้อมูลของแฟ้มนี้เป็นแฟ้มข้อมูลเรียงลำดับชนิดระเบียนความยาวคงที่ ประกอบด้วยกลุ่มของระเบียนที่มีโครงสร้างเหมือนกัน คือ ประกอบด้วยกลุ่มของพจน์ กลุ่มละ ไม่เกิน 10 พจน์ โดยระบบจองเนื้อที่ไว้ตายตัว 10 ระเบียน เรียงลำดับตามค่าแอสกีจากน้อยไปมากในแต่ละกลุ่ม ซึ่งมีหมายเลขกำกับ แต่ละพจน์จะมีเขตระบุตำแหน่งของโพสติ้งในแฟ้มข้อมูลโพสติ้งของแฟ้มข้อมูลหกกลับ ชี้ไปยังบล็อคและออฟเซทของแฟ้มข้อมูลโพสติ้งดังกล่าว โครงสร้างข้อมูลของแต่ละกลุ่มระเบียนประกอบด้วย 34 เขตข้อมูล ความยาวคงที่ ประกอบด้วยตั้งแต่ 1 กลุ่มระเบียนขึ้นไปอยู่ต่อเนื่องกันแบบเรียงลำดับ มีรายละเอียดของแต่ละกลุ่มระเบียนดังในตาราง 4.11

ไบท์ที่	ข้อมูลในรูปเลขฐานสิบหก	ความหมาย
1 - 2	ตั้งแต่ 0100	ระบุหมายเลขกลุ่มของพจน์
3 - 4	0100 - 0A00	ระบุจำนวนพจน์ในกลุ่ม
5 - 6	0100	
7 - 8	ตั้งแต่ 0000	เก็บหมายเลขกลุ่มของพจน์ถัดไป
9 - 18	20 หรือ พจน์	เก็บพจน์ที่เป็นโหนดใบลำดับ 1
19 - 20	ตั้งแต่ 0100	บล็อคของค่าโพสติ้งของโหนด 1
21 - 22	ตั้งแต่ 0000	ออฟเซทของค่าโพสติ้งของโหนด 1
23 - 32	20 หรือ พจน์	เก็บพจน์ที่เป็นโหนดใบลำดับ 2
33 - 34	ตั้งแต่ 0100	บล็อคของค่าโพสติ้งของโหนด 2
35 - 36	ตั้งแต่ 0000	ออฟเซทของค่าโพสติ้งของโหนด 2
37 - 46	20 หรือ พจน์	เก็บพจน์ที่เป็นโหนดใบลำดับ 3

ตาราง 4.11 แสดงรายละเอียดแต่ละกลุ่มระเบียนของแฟ้มข้อมูลบี*ทรีลีฟสำหรับพจน์ที่ความยาวไม่เกิน 10 อักขระ

ไบท์ที่	ข้อมูลในรูปเลขฐานสิบหก	ความหมาย
47 - 48	ตั้งแต่ 0100	บล็อคของค่าโพสติ้งของโหนด 3
49 - 50	ตั้งแต่ 0000	ออฟเซทของค่าโพสติ้งของโหนด 3
51 - 60	20 หรือ พจน์	เก็บพจน์ที่เป็นโหนดใบลำดับ 4
61 - 62	ตั้งแต่ 0100	บล็อคของค่าโพสติ้งของโหนด 4
63 - 64	ตั้งแต่ 0000	ออฟเซทของค่าโพสติ้งของโหนด 4
65 - 74	20 หรือ พจน์	เก็บพจน์ที่เป็นโหนดใบลำดับ 5
75 - 76	ตั้งแต่ 0100	บล็อคของค่าโพสติ้งของโหนด 5
77 - 78	ตั้งแต่ 0000	ออฟเซทของค่าโพสติ้งของโหนด 5
79 - 88	20 หรือ พจน์	เก็บพจน์ที่เป็นโหนดใบลำดับ 6
89 - 90	ตั้งแต่ 0100	บล็อคของค่าโพสติ้งของโหนด 6
91 - 92	ตั้งแต่ 0000	ออฟเซทของค่าโพสติ้งของโหนด 6
93 -102	20 หรือ พจน์	เก็บพจน์ที่เป็นโหนดใบลำดับ 7
103 -104	ตั้งแต่ 0100	บล็อคของค่าโพสติ้งของโหนด 7
105 -106	ตั้งแต่ 0000	ออฟเซทของค่าโพสติ้งของโหนด 7
107 -116	20 หรือ พจน์	เก็บพจน์ที่เป็นโหนดใบลำดับ 8
117 -118	ตั้งแต่ 0100	บล็อคของค่าโพสติ้งของโหนด 8
119 -120	ตั้งแต่ 0000	ออฟเซทของค่าโพสติ้งของโหนด 8
121 -130	20 หรือ พจน์	เก็บพจน์ที่เป็นโหนดใบลำดับ 9
131 -132	ตั้งแต่ 0100	บล็อคของค่าโพสติ้งของโหนด 9
133 -134	ตั้งแต่ 0000	ออฟเซทของค่าโพสติ้งของโหนด 9
135 -144	20 หรือ พจน์	เก็บพจน์ที่เป็นโหนดใบลำดับ 10
145 -146	ตั้งแต่ 0100	บล็อคของค่าโพสติ้งของโหนด 10
147 -148	ตั้งแต่ 0000	ออฟเซทของค่าโพสติ้งของโหนด 10

ตาราง 4.11 แสดงรายละเอียดแต่ละกลุ่มระเบียนของแฟ้มข้อมูลบี*ทรีลีฟสำหรับพจน์ที่ความยาวไม่เกิน 10 อักขระ (ต่อ)

ข้อมูลไบท์ที่ 7 - 8 เป็นลิงค์ลิสต์ที่เชื่อมโยงระหว่างกลุ่มระเบียนต่าง ๆ ทำให้ระบบสามารถเข้าถึงข้อมูลแบบเรียงลำดับจากน้อยไปมากได้ และเนื่องจากโครงสร้างเป็นแบบความยาวระเบียนคงที่ ระเบียนที่ไม่ใช้ระบบจะไม่ทำการลบค่าข้อมูลในเขตออก แต่จะใส่ค่า 0000 ไว้ในเขตข้อมูลที่เก็บค่าบล็อคและออฟเซทของค่าโพสติ้งของโหนดนั้น ๆ แทน นอกจากนี้ระบบยังระบุจำนวนระเบียนภายในกลุ่มไว้ในไบท์ที่ 3 - 4 ของทุกกลุ่มระเบียนอีกด้วย ซึ่งมีค่าได้ ตั้งแต่ 1 ถึง 10 ค่าสูงสุดเวลาที่มีการเพิ่มข้อมูลเก็บเป็นค่าที่ 11 ระบบก็จะทำการปรับทรีใหม่ โดยย้ายข้อมูลในกลุ่มระเบียนที่เต็มแล้วไปไว้ในโหนดพี่น้อง แล้วปรับค่าโหนดกิ่งใหม่ ตามโครงสร้างของบี*ทรี แบบลำดับเป็น 10 จะพบว่า ในการเก็บคำหลักนั้น ระบบจะจัดเก็บแบบความยาวคงที่ คำละ 10 ไบท์ คำใดความยาวไม่ถึง ระบบจะเติมช่องว่างให้ทางขวามือจนครบ 10 ไบท์ เพื่อความสะดวกในการประมวลผล โดยทำเป็นแบบความยาวระเบียนคงที่ ดังนั้น ระบบจึงไม่สามารถใส่ค่าโพสติ้งไว้ในแฟ้มนี้ได้ เพราะคำหลักบางคำอาจจะมีค่าโพสติ้งหลายค่าทำให้ความยาวแปรผัน ผู้พัฒนาโปรแกรมจึงออกแบบให้แฟ้มนี้มีความยาวคงที่ แล้วแทนที่จะใส่ค่าโพสติ้งไว้ในแฟ้ม ก็ใส่เป็นตำแหน่งของค่าโพสติ้ง ซึ่งได้แยกเก็บค่าโพสติ้งไว้ต่างหากในแฟ้มข้อมูลค่าโพสติ้งของแฟ้มข้อมูลหกกลับแทน วิธีนี้ทำให้ทรีมีขนาดไม่ใหญ่จนเกินไป และไม่จำเป็นต้องเก็บคำหลักที่ซ้ำซ้อนกันให้เปลืองเนื้อที่ ส่วนคำหลักที่มีความยาวเกิน 10 อักขระ ระบบก็แยกไปจัดเก็บไว้ในแฟ้มข้อมูลบี*ทรีโหนดสำหรับพจน์ที่ความยาวเกิน 10 อักขระแทน

4. แฟ้มข้อมูลบี*ทรีโหนดสำหรับพจน์ที่ความยาวเกิน 10 อักขระ

แฟ้มข้อมูลบี*ทรีโหนดสำหรับพจน์ที่ความยาวเกิน 10 อักขระ (xxxxxx.NO2) เป็นแฟ้มข้อมูลที่เก็บพจน์ที่เป็นโหนดกิ่ง (Branch Node) และเก็บตัวชี้ไปยังโหนดใบ (Leaf Node) ที่เก็บตำแหน่งของโพสติ้งของพจน์ที่ความยาวเกิน 10 อักขระ โดยระบบจะใช้แฟ้มนี้ร่วมกับแฟ้มข้อมูลบี*ทรีลีฟสำหรับพจน์ที่ความยาวเกิน 10 อักขระ ซึ่งเป็นที่เก็บคำหลักและตำแหน่งของโพสติ้ง เวลาระบบจะค้นหาข้อมูลแทนที่จะเข้าไปค้นตั้งแต่ต้นทรีในแฟ้มบี*ทรีลีฟ ระบบก็จะมาเริ่มค้นที่แฟ้มนี้ก่อนเพื่อดูว่าข้อมูลที่จะค้นตกอยู่ในช่วงใดของทรีโดยเทียบกับพจน์ที่เป็นโหนดกิ่ง ดูว่าชี้ไปยังโหนดใบกลุ่มใด จากนั้นระบบจึงไปค้นในกลุ่มของโหนดใบที่ตัวชี้ชี้มา เพื่อเข้าถึงระเบียนซึ่งเก็บตำแหน่งของโพสติ้งอีกทีหนึ่ง แล้วจึงไปเข้าถึงโพสติ้งในแฟ้มข้อมูลโพสติ้งของแฟ้มข้อมูลหกกลับ (xxxxxx.IFP)

โครงสร้างข้อมูลของแฟ้มข้อมูลบี*ทรีโหนดสำหรับพจน์ที่มีความยาวเกิน 10 อักขระ เป็นแฟ้มข้อมูลแบบเรียงลำดับชนิดระเบียนความยาวคงที่ ประกอบด้วยกลุ่มของระเบียนที่มีโครงสร้างเหมือนกัน คือประกอบด้วยกลุ่มของพจน์ที่เป็นโหนดกิ่ง กลุ่มละ ไม่เกิน 10 พจน์ โดยระบบจองเนื้อที่ไว้ตายตัว 10 ระเบียน เรียงลำดับตามค่าแอสกีจากน้อยไปมากในแต่ละกลุ่ม ซึ่งมีหมายเลขกำกับ แต่ละพจน์จะมีตัวชี้ในลักษณะลิงค์ลิสต์ ชี้ไปยังบล็อคและออฟเซทของแฟ้มข้อมูลบี*ทรีลีฟที่เก็บกลุ่มคำหลักที่มีค่าแอสกีมากกว่าหรือเท่ากับพจน์นั้นแต่ไม่เกินพจน์ถัดไป ดังนั้น เวลาระบบค้นหาคำหลัก ก็จะนำพจน์มาเปรียบเทียบกับระเบียนแรกของแฟ้มนี้ ถ้าค่าแอสกีของพจน์ที่จะค้นน้อยกว่าค่าแอสกีของพจน์ที่เป็นโหนดกิ่งที่สอง แสดงว่าข้อมูลอยู่ในโหนดกิ่งแรก ระบบจะใช้ค่าตัวชี้ของโหนดกิ่งแรกเพื่อไปค้นในแฟ้มที่เก็บโหนดใบต่อไป แต่ถ้าค่าที่จะค้นเท่ากับโหนดกิ่งที่สอง แสดงว่าข้อมูลอยู่ในโหนดกิ่งที่สอง แต่ถ้าค่าที่จะค้นมากกว่าโหนดกิ่งที่สอง ระบบก็จะเทียบกับระเบียนที่เป็นโหนดกิ่งถัดไปเรื่อย ๆ จนกว่าค่าแอสกีของพจน์ที่จะค้นเท่ากับ แสดงว่าค่าที่ต้องการอยู่ในโหนดใบของพจน์นั้น ๆ หรือจนกว่าค่าแอสกีของพจน์ที่จะค้นน้อยกว่าค่าในโหนดกิ่ง แสดงว่าพจน์ที่จะค้นอยู่ในโหนดก่อนหน้า ระบบก็จะไปยังแฟ้มข้อมูลบี*ทรีลีฟ ตามค่าตัวชี้ แล้วค้นหาต่อไป

โครงสร้างข้อมูลของแต่ละกลุ่มระเบียนประกอบด้วย 23 เขตข้อมูลความยาวคงที่ ประกอบด้วยตั้งแต่ 1 กลุ่มระเบียนขึ้นไปอยู่ต่อเนื่องกันแบบเรียงลำดับเหมือนกับแฟ้มข้อมูลบี*ทรีโหนดสำหรับพจน์ที่มีความยาวไม่เกิน 10 อักขระ ทุกประการ เพียงแต่เขตข้อมูลที่เก็บพจน์ที่เป็นโหนดกิ่งนั้น ระบบจองเนื้อที่ไว้ 30 ไบท์ แทนที่จะเป็น 10 ไบท์ นั่นคือ สามารถจัดเก็บพจน์ที่มีความยาวตั้งแต่ 10 ถึง 30 อักขระนั่นเอง รายละเอียดของแต่ละกลุ่มระเบียนดังแสดงในตาราง 4.12

ไบท์ที่	ข้อมูลในรูปเลขฐานสิบหก	ความหมาย
1 - 2	ตั้งแต่ 0100	ระบุหมายเลขกลุ่มของพจน์
3 - 4	0100 - 0A00	ระบุจำนวนพจน์ในกลุ่ม
5 - 6	0200	

ตาราง 4.12 แสดงรายละเอียดแต่ละกลุ่มระเบียนของแฟ้มข้อมูลบี*ทรีโหนดสำหรับพจน์ที่ความยาวเกิน 10 อักขระ

ไบท์ที่	ข้อมูลในรูปเลขฐานสิบหก	ความหมาย
7 - 36	20 หรือ พจน์	เก็บพจน์ที่เป็นโหนดกิ่งลำดับ 1
37 - 38	0000 หรือ FFFF ลงมา	เก็บค่าตัวชี้ของพจน์ลำดับ 1
39 - 68	20 หรือ พจน์	เก็บพจน์ที่เป็นโหนดกิ่งลำดับ 2
69 - 70	0000 หรือ FFFF ลงมา	เก็บค่าตัวชี้ของพจน์ลำดับ 2
71 -100	20 หรือ พจน์	เก็บพจน์ที่เป็นโหนดกิ่งลำดับ 3
101 -102	0000 หรือ FFFF ลงมา	เก็บค่าตัวชี้ของพจน์ลำดับ 3
103 -132	20 หรือ พจน์	เก็บพจน์ที่เป็นโหนดกิ่งลำดับ 4
133 -134	0000 หรือ FFFF ลงมา	เก็บค่าตัวชี้ของพจน์ลำดับ 4
135 -164	20 หรือ พจน์	เก็บพจน์ที่เป็นโหนดกิ่งลำดับ 5
165 -166	0000 หรือ FFFF ลงมา	เก็บค่าตัวชี้ของพจน์ลำดับ 5
167 -196	20 หรือ พจน์	เก็บพจน์ที่เป็นโหนดกิ่งลำดับ 6
197 -198	0000 หรือ FFFF ลงมา	เก็บค่าตัวชี้ของพจน์ลำดับ 6
199 -228	20 หรือ พจน์	เก็บพจน์ที่เป็นโหนดกิ่งลำดับ 7
229 -230	0000 หรือ FFFF ลงมา	เก็บค่าตัวชี้ของพจน์ลำดับ 7
231 -260	20 หรือ พจน์	เก็บพจน์ที่เป็นโหนดกิ่งลำดับ 8
261 -262	0000 หรือ FFFF ลงมา	เก็บค่าตัวชี้ของพจน์ลำดับ 8
263- 292	20 หรือ พจน์	เก็บพจน์ที่เป็นโหนดกิ่งลำดับ 9
293- 294	0000 หรือ FFFF ลงมา	เก็บค่าตัวชี้ของพจน์ลำดับ 9
295- 324	20 หรือ พจน์	เก็บพจน์ที่เป็นโหนดกิ่งลำดับ 10
325- 326	0000 หรือ FFFF ลงมา	เก็บค่าตัวชี้ของพจน์ลำดับ 10

ตาราง 4.12 แสดงรายละเอียดแต่ละกลุ่มระเบียนของแฟ้มข้อมูลบี*ทรีโหนดสำหรับพจน์ที่ความยาวเกิน 10 อักขระ (ต่อ)

ตัวอย่างรายละเอียดของแฟ้มข้อมูลบี*ทรีโหนดสำหรับพจน์ที่ความยาวเกิน 10 อักขระ ดังแสดงในภาคผนวก

5. แฟ้มข้อมูลบี*ทรีลีฟสำหรับพจน์ที่ความยาวเกิน 10 อักขระ

แฟ้มข้อมูลบี*ทรีลีฟสำหรับพจน์ที่ความยาวเกิน 10 อักขระ (xxxxxx.L02) เป็นแฟ้มข้อมูลที่เก็บคำหลักและตำแหน่งโพสติ้งของคำหลักที่ความยาวเกิน 10 อักขระ ซึ่งเมื่อระบบค้นตำแหน่งตัวชี้จากแฟ้มข้อมูลบี*ทรีโหนดได้แล้ว ระบบจะมาทำการค้นในแฟ้มนี้จนพบหรือไม่พบคำหลักที่ระบุ โครงสร้างข้อมูลของแฟ้มนี้เป็นแฟ้มข้อมูลเรียงลำดับ ชนิดระเบียนความยาวคงที่ ประกอบด้วยกลุ่มของระเบียนที่มีโครงสร้างเหมือนกัน คือประกอบด้วยกลุ่มของพจน์ กลุ่มละ ไม่เกิน 10 พจน์ โดยระบบจองเนื้อที่ไว้ตายตัว 10 ระเบียน เรียงลำดับตามค่าแอสกีของคำหลัก จากน้อยไปมากในแต่ละกลุ่มซึ่งมีหมายเลขกำกับ แต่ละพจน์จะมีเขตระบุตำแหน่งของโพสติ้งในแฟ้มข้อมูลโพสติ้งของแฟ้มข้อมูลหกกลับ ชี้ไปยังบล็อคและออฟเซทของแฟ้มข้อมูลโพสติ้งดังกล่าว โครงสร้างข้อมูลของแต่ละกลุ่มระเบียนประกอบด้วย 34 เขตข้อมูล ความยาวคงที่ประกอบด้วยตั้งแต่ 1 กลุ่มระเบียนขึ้นไปอยู่ต่อเนื่องกันแบบเรียงลำดับ มีรายละเอียดเหมือนกับแฟ้มข้อมูลบี*ทรีลีฟสำหรับพจน์ที่มีความยาวไม่เกิน 10 อักขระ ทุกประการ เพียงแต่เขตข้อมูลที่เก็บพจน์ที่เป็นโหนดใบนั้น ระบบจองเนื้อที่ไว้ 30 ไบท์ แทนที่จะเป็น 10 ไบท์ นั่นคือ สามารถจัดเก็บพจน์ที่มีความยาวตั้งแต่ 10 ถึง 30 อักขระนั่นเอง โครงสร้างข้อมูลของแต่ละกลุ่มระเบียนดังแสดงในตาราง 4.13

ไบท์ที่	ข้อมูลในรูปเลขฐานสิบหก	ความหมาย
1 - 2	ตั้งแต่ 0100	ระบุหมายเลขกลุ่มของพจน์
3 - 4	0100 - 0A00	ระบุจำนวนพจน์ในกลุ่ม
5 - 6	0200	
7 - 8	ตั้งแต่ 0000	เก็บหมายเลขกลุ่มของพจน์ถัดไป
9 - 38	20 หรือ พจน์	เก็บพจน์ที่เป็นโหนดใบลำดับ 1
39 - 40	ตั้งแต่ 0100	บล็อคของค่าโพสติ้งของโหนด 1
41 - 42	ตั้งแต่ 0000	ออฟเซทของค่าโพสติ้งของโหนด 1
43 - 72	20 หรือ พจน์	เก็บพจน์ที่เป็นโหนดใบลำดับ 2

ตาราง 4.13 แสดงรายละเอียดแต่ละกลุ่มระเบียนของแฟ้มข้อมูลบี*ทรีลีฟสำหรับพจน์ที่ความยาวเกิน 10 อักขระ

ไบท์ที่	ข้อมูลในรูปเลขฐานสิบหก	ความหมาย
73 - 74	ตั้งแต่ 0100	บล็อคของค่าโพสติ้งของโหนด 2
75 - 76	ตั้งแต่ 0000	ออฟเซทของค่าโพสติ้งของโหนด 2
77 -106	20 หรือ พจน์	เก็บพจน์ที่เป็นโหนดใบลำดับ 3
107 -108	ตั้งแต่ 0100	บล็อคของค่าโพสติ้งของโหนด 3
109 -110	ตั้งแต่ 0000	ออฟเซทของค่าโพสติ้งของโหนด 3
111 -140	20 หรือ พจน์	เก็บพจน์ที่เป็นโหนดใบลำดับ 4
141 -142	ตั้งแต่ 0100	บล็อคของค่าโพสติ้งของโหนด 4
143 -144	ตั้งแต่ 0000	ออฟเซทของค่าโพสติ้งของโหนด 4
145 -174	20 หรือ พจน์	เก็บพจน์ที่เป็นโหนดใบลำดับ 5
175 -176	ตั้งแต่ 0100	บล็อคของค่าโพสติ้งของโหนด 5
177 -178	ตั้งแต่ 0000	ออฟเซทของค่าโพสติ้งของโหนด 5
179 -208	20 หรือ พจน์	เก็บพจน์ที่เป็นโหนดใบลำดับ 6
209 -210	ตั้งแต่ 0100	บล็อคของค่าโพสติ้งของโหนด 6
211 -212	ตั้งแต่ 0000	ออฟเซทของค่าโพสติ้งของโหนด 6
213 -242	20 หรือ พจน์	เก็บพจน์ที่เป็นโหนดใบลำดับ 7
243 -244	ตั้งแต่ 0100	บล็อคของค่าโพสติ้งของโหนด 7
245 -246	ตั้งแต่ 0000	ออฟเซทของค่าโพสติ้งของโหนด 7
247 -276	20 หรือ พจน์	เก็บพจน์ที่เป็นโหนดใบลำดับ 8
277 -278	ตั้งแต่ 0100	บล็อคของค่าโพสติ้งของโหนด 8
279 -280	ตั้งแต่ 0000	ออฟเซทของค่าโพสติ้งของโหนด 8
281 -310	20 หรือ พจน์	เก็บพจน์ที่เป็นโหนดใบลำดับ 9
311 -312	ตั้งแต่ 0100	บล็อคของค่าโพสติ้งของโหนด 9
313 -314	ตั้งแต่ 0000	ออฟเซทของค่าโพสติ้งของโหนด 9
315 -344	20 หรือ พจน์	เก็บพจน์ที่เป็นโหนดใบลำดับ 10
345 -346	ตั้งแต่ 0100	บล็อคของค่าโพสติ้งของโหนด 10
347 -348	ตั้งแต่ 0000	ออฟเซทของค่าโพสติ้งของโหนด 10

ตาราง 4.13 แสดงรายละเอียดแต่ละกลุ่มระเบียนของแฟ้มข้อมูลบี*ทรีลีฟสำหรับพจน์ที่ความยาวเกิน 10 อักขระ (ต่อ)

6. แฟ้มข้อมูลโพสติ้งของแฟ้มข้อมูลหกกลับ

แฟ้มข้อมูลโพสติ้งของแฟ้มข้อมูลหกกลับ (Inverted file postings หรือแฟ้ม xxxxxx.IFP) เป็นแฟ้มข้อมูลระบบที่จัดเก็บค่าโพสติ้งของคำหลักแต่ละคำไว้สำหรับไปค้นหาในแฟ้มข้อมูลหลักโดยจัดเก็บค่าโพสติ้งของคำหลักเรียงตามลำดับก่อน-หลังของการบันทึกข้อมูลเข้าของผู้ใช้ โครงสร้างแฟ้มข้อมูลเป็นแฟ้มข้อมูลลักษณะเรียงลำดับชนิดความยาวระเบียนแปรผัน และประยุกต์ใช้ลิงค์ลิสต์ในการเชื่อมโยงข้อมูลถึงกันทั้งภายในแฟ้ม และระหว่างแฟ้มข้อมูลนี้กับแฟ้มข้อมูลบี*ทรีลีฟ นั่นคือ ระบบจะได้บล็อคและออฟเซทของค่าโพสติ้งจากแฟ้มข้อมูลบี*ทรีลีฟ แล้วจะมาค้นในแฟ้มข้อมูลโพสติ้ง ซึ่งจะเก็บรายละเอียดของค่าโพสติ้ง ได้แก่ หมายเลขแฟ้มข้อมูลหลักของระเบียนที่พบคำหลัก, เลขต่อท้ายระบุเขตที่พบคำหลัก, หมายเลขระบุเขตข้อมูลย่อย และหมายเลขลำดับพจน์ ซึ่งเนื่องจากในคำหลักหนึ่ง ๆ อาจจะปรากฏอยู่หลายระเบียน ดังนั้น ระบบจึงกำหนดเขตที่จัดเก็บค่าโพสติ้งนี้ในลักษณะความยาวแปรผัน และในกรณีที่มีการปรับปรุงแฟ้มข้อมูลหกกลับในภายหลังแล้วมีคำหลักซ้ำเดิมอีก ถ้าที่จัดเก็บเดิมไม่พอ ระบบก็จะจัดเก็บไว้ท้ายแฟ้มข้อมูลโพสติ้ง แล้วโยงลิงค์จากระเบียนที่เก็บคำหลักเดิมชี้มายังคำหลักใหม่ เพื่อไม่ต้องขยับข้อมูลทั้งแฟ้มให้เป็นการเสียเวลา

โครงสร้างข้อมูลของแฟ้มข้อมูลโพสติ้งสามารถแบ่งออกได้เป็น 3 ส่วนใหญ่ ๆ ได้แก่ ส่วนหัวของแฟ้มข้อมูลโพสติ้ง, ส่วนหัวของระเบียน และส่วนระเบียน

1. ส่วนหัวของแฟ้มข้อมูลโพสติ้ง

ในการจัดเก็บข้อมูลโพสติ้ง ระบบจะจัดเตรียมแฟ้มข้อมูล โดยแบ่งเก็บเป็นบล็อค ๆ ละ 512 ไบท์ ทุกบล็อคจะใช้ 2 ไบท์แรกสุดเป็นเขตระบุบล็อค มีค่าตั้งแต่ 0100_{16} ไปเรื่อย ๆ ถัดจากนั้นก็จะตามด้วยข้อมูล สำหรับบล็อกแรกสุดนั้น ระบบจะใช้ไบท์ที่ 3 - 6 จำนวน 4 ไบท์ เอาไว้เก็บหมายเลขบล็อค และค่าออฟเซทของระเบียนใหม่ที่จะบันทึก นั่นคือระบุว่าแฟ้มข้อมูลจบอยู่ที่ไหน เพื่อให้นำระเบียนใหม่มาต่อท้ายนั่นเอง ถัดจากนั้นก็จะตามด้วยส่วนหัวของระเบียนและระเบียนต่อไป ลักษณะโครงสร้างส่วนหัวของแฟ้มข้อมูลโพสติ้งดังแสดงในตาราง 4.14

ไบท์ที่	ข้อมูลในรูปเลขฐานสิบหก	ความหมาย
1 - 2	ตั้งแต่ 0100 ไปเรื่อย ๆ	ระบุหมายเลขบล็อคนั้น ๆ
3 - 4	ตั้งแต่ 0100 ไปเรื่อย ๆ	หมายเลขบล็อคสุดท้ายของแฟ้ม
5 - 6	ตั้งแต่ 0100 ไปเรื่อย ๆ	ออฟเซทที่จะบันทึกระเบียนใหม่

ตาราง 4.14 แสดงรายละเอียดส่วนหัวของแฟ้มข้อมูลโพสติ้งของแฟ้มข้อมูลหกกลับ

ตัวอย่างรายละเอียดของโครงสร้างส่วนหัวของแฟ้มข้อมูลโพสติ้ง ดังแสดงในภาคผนวก

2. <u>ส่วนหัวของระเบียน</u>

ถัดจากส่วนหัวของแฟ้มข้อมูลโพสติ้ง จะประกอบด้วยส่วนหัวของระเบียน ซึ่งแบ่งออกเป็น 5 เขตข้อมูล ความยาวคงที่ในทุกระเบียน แต่ละเขตมีขนาด 2 ไบท์ ดังมีรายละเอียดดังในตาราง 4.15

ไบท์ที่	ข้อมูลในรูปเลขฐานสิบหก	ความหมาย
1 - 2	ตั้งแต่ 0000	บล็อคของโพสติ้งของคำหลักเดียวกัน
3 - 4	ตั้งแต่ 0000	ออฟเซทของโพสติ้งของคำหลักเดียวกัน
5 - 6	ตั้งแต่ 0000	จำนวนโพสติ้งของคำหลักเดียวกัน
7 - 8	ตั้งแต่ 0000	จำนวนระเบียนใช้งานในกลุ่มระเบียน
9 - 10	ตั้งแต่ 0100	จำนวนระเบียนที่จองไว้ในกลุ่มระเบียน

ตาราง 4.15 แสดงรายละเอียดส่วนหัวของระเบียน

จำนวนระเบียนที่จองไว้ในกลุ่มระเบียนจะมีค่าเริ่มต้นเป็น 1 เมื่อมีการปรับปรุงแฟ้มข้อมูลหกกลับในเที่ยวใดที่มีคำหลักซ้ำกันมากกว่า 1 คำ ระบบจะจองจำนวนระเบียนเพิ่มในลักษณะอนุกรมแบบ 1, 2, 4, 8, 16, 32, ... เอาไว้เพื่อบันทึกค่าโพสติ้งโดยไม่ต้องมีส่วนหัวของระเบียนใหม่ เป็นการประหยัดเนื้อที่ได้ถึง 10 ไบท์ต่อระเบียน แต่ถ้ามีการมาปรับปรุงแฟ้มภายหลังอีก ถ้าเนื้อที่ที่จองไว้เต็มแล้ว ระบบจะบันทึกข้อมูลไว้ท้ายแฟ้ม และใช้ลิงค์ลิสต์โยงจากระเบียนเดิมมายังระเบียนใหม่ โดยระบบจะดูจากข้อมูลในไบท์ 1-2 และ 3-4 ว่าเมื่ออ่านข้อมูลระเบียนนี้จบแล้ว มีต่ออีกหรือไม่ ถ้าไม่มี ข้อมูลจะเป็น 0000 0000 ถ้ามีจะเป็นค่าบล็อคและออฟเซท ระบบก็จะไปเข้าถึงระเบียนนั้น ๆ ต่อไป จะเห็นได้ว่า ด้วยเทคนิคนี้ เป็นการลดเวลาในการปรับปรุงแฟ้ม เพราะไม่ต้องจัดให้ระเบียนอยู่ติดกันจริง ๆ แต่ใช้ลิงค์ลิสต์เชื่อมโยง แต่เมื่อเวลาผ่านไปมาก ๆ จะทำให้การค้นใช้เวลามาก ระบบได้จัดทางเลือก F ในโปรแกรม ISISINV.EXE เพื่อสร้างแฟ้มข้อมูลหกกลับใหม่ ในลักษณะที่ระบบจะจัดเรียงลำดับคำหลักเสียก่อนที่จะมาบันทึกในแฟ้มข้อมูลโพสติ้ง ทำให้คำหลักค่าเดียวกันอยู่ในระเบียนเดียวกัน ลดลิงค์ลิสต์ที่ไม่จำเป็น ทำให้ลดเวลาในการค้นหาข้อมูลได้มาก

3. <u>ส่วนระเบียน</u>

ถัดจากส่วนหัวของระเบียน จะตามด้วยส่วนระเบียน ซึ่งเป็นกลุ่มของเขตข้อมูลซึ่งความยาวคงที่แต่จำนวนแปรผันได้ ส่วนระเบียนประกอบด้วยเขต 4 เขต ความยาวเขตละ 2 ไบท์ ได้แก่ หมายเลขแฟ้มข้อมูลหลักของระเบียนที่พบคำหลัก, เลขต่อท้ายระบุเขตที่พบคำหลัก, หมายเลขระบุเขตข้อมูลย่อย และหมายเลขลำดับพจน์ รายละเอียดส่วนระเบียนของแฟ้มข้อมูลโพสติ้งตังแสดงในตาราง 4.16

ไบท์ที่	ข้อมูลในรูปเลขฐานสิบหก	ความหมาย
1 - 2	0100 จนถึง FF7F	หมายเลขแฟ้มข้อมูลหลักของระเบียน
3 - 4	0000 จนถึง 6300	เลขต่อท้ายระบุเขตที่พบคำหลัก
5 - 6	ตั้งแต่ 0100	หมายเลขระบุเขตข้อมูลย่อย
7 - 8	ตั้งแต่ 0100	หมายเลขลำดับพจน์

ตาราง 4.16 แสดงรายละเอียดส่วนระเบียนของแฟ้มข้อมูลโพสติ้ง

ข้อมูลในไบท์ที่ 1 และ 2 เก็บหมายเลขแฟ้มข้อมูลหลักของระเบียนที่มีคำหลักปรากฏอยู่ ส่วนข้อมูลในไบท์ที่ 3 และ 4 เก็บเลขต่อท้ายระบุเขตที่พบคำหลัก ซึ่งมีค่าได้ตั้งแต่ 0 - 99_{10} ข้อมูลในไบท์ที่ 5 และ 6 เก็บหมายเลขระบุเขตข้อมูลย่อย ในกรณีที่มีเขตข้อมูลย่อย ถ้าไม่มี ระบบจะกำหนดให้เป็น 1 เสมอ ส่วนข้อมูลในไบท์ที่ 7 และ 8 เก็บหมายเลขลำดับพจน์ในเขตนั้น ๆ ว่าระบบพบคำหลักเป็นพจน์ที่เท่าใดจากต้นเขต เช่นถ้าพบเป็นคำแรกก็จะกำหนดค่าเป็น 1 เป็นต้น ข้อมูลทุกเขตจะถูกกำหนดค่าเป็น $FFFF_{16}$ ในกรณีที่จองเนื้อที่ไว้แล้วไม่ได้ใช้ หรือมีการลบระเบียนที่มีคำหลักนี้ทิ้ง ระบบก็จะมาปรับปรุงแฟ้มข้อมูลโพสติ้งในเขตนี้

ผู้ใช้สามารถพิมพ์รายการค่าโพสติ้งของฐานข้อมูลใด ๆ ออกมาทางเครื่องพิมพ์ได้โดยเลือกทางเลือก D ในโปรแกรม ISISINV.EXE ซึ่งระบบจะแสดงคำหลักพร้อมค่าโพสติ้งออกมาเก็บไว้ในแฟ้ม IFLIST.LST โดยมีรูปแบบดังตัวอย่างในภาพ 4.18

```
Inverted file Listing                                Data base: TEST
    1     1  ANT
                        1/ 38   5/6/1/1
    2     2  ARE        1/ 56   6/6/1/2
                                 7/4/1/4
```

ภาพ 4.18 แสดงตัวอย่างรายละเอียดในแฟ้มข้อมูลหกกลับ

จากภาพที่ 18 แสดงว่า คำหลัก ANT อยู่เป็นลำดับแรก พบ 1 แห่ง เป็นคำที่หนึ่งในเขตข้อมูลที่มีเลขต่อท้ายเป็นหก และเป็นระเบียนหมายเลขแฟ้มข้อมูลหลักที่ห้า ค่าโพสติ้งเก็บอยู่ในแฟ้มข้อมูลโพสติ้งบล็อคที่ 1 และค่าออฟเซทเป็น 38_{10} นั่นคือ อยู่ถัดจากต้นแฟ้มข้อมูลโพสติ้งไป 76 ไบท์นั่นเอง (ค่าออฟเซทของแฟ้มนี้คิดหน่วยละ 2 ไบท์) ส่วนคำหลัก ARE พบ 2 แห่ง คือที่หมายเลขแฟ้มข้อมูลหลัก 6 และ 7 ซึ่งค่าโพสติ้งเก็บอยู่ในระเบียนเดียวกันของแฟ้มข้อมูลโพสติ้ง คือ บล็อคที่ 1 ออฟเซทเป็น 56_{10} นั่นคือกลุ่มระเบียนที่เก็บค่าโพสติ้งของคำหลักนี้อยู่ถัดจากต้นแฟ้มข้อมูลโพสติ้งไป 112 ไบท์ทั้งสองค่า ตัวอย่างของแฟ้มข้อมูลโพสติ้งของแฟ้มข้อมูลหกกลับดังแสดงในภาคผนวก

โครงสร้างข้อมูลของแฟ้มข้อมูลเก็บรักษา

แฟ้มข้อมูลเก็บรักษา (Save File created during retrieval หรือ xxxxxx.SAV) เป็นแฟ้มข้อมูลระบบที่เก็บรายละเอียดค่าโพสติ้งของการค้นหาข้อมูล โดยการใช้ทางเลือก S ของโปรแกรม ISIS.EXE เพื่อนำไปออกรายงานด้วยโปรแกรม ISISPRT.EXE ในภายหลัง โครงสร้างแฟ้มข้อมูลเป็นแบบระเบียนความยาวแปรผัน ประกอบด้วย รายละเอียดของค่าโพสติ้งของระเบียนที่ค้นหาได้ ได้แก่ หมายเลขแฟ้มข้อมูลหลักของระเบียนที่พบคำหลัก, เลขต่อท้ายระบุเขตที่พบคำหลัก, หมายเลขระบุเขตข้อมูลย่อย และหมายเลขลำดับพจน์ ดังรายละเอียดในตาราง 4.17

ไบท์ที่	ข้อมูลในรูปเลขฐานสิบหก	ความหมาย
1 - 2	ตั้งแต่ 0100 ไปเรื่อย ๆ	ระบุหมายเลขบล็อคนั้น ๆ
3 - 4	0101	
5 - 6	0000	
7 - 8	ตั้งแต่ 0100 ไปเรื่อย ๆ	จำนวนโพสติ้งภายในบล็อคนี้
9 - 10	ตั้งแต่ 0000 ไปเรื่อย ๆ	จำนวนระเบียนที่ใช้งานในบล็อค
11 - 12	ตั้งแต่ 0100 ไปเรื่อย ๆ	จำนวนระเบียนที่จองไว้ในบล็อค
13 - 14	0100 จนถึง FF7F	หมายเลขแฟ้มข้อมูลหลักระเบียนที่ 1
15 - 16	0000 จนถึง 6300	เลขต่อท้ายระบุเขตที่พบคำหลัก
17 - 18	ตั้งแต่ 0100	หมายเลขระบุเขตข้อมูลย่อย
19 - 20	ตั้งแต่ 0100	หมายเลขลำดับพจน์
21 - 22	0100 จนถึง FF7F	หมายเลขแฟ้มข้อมูลหลักระเบียนที่ 2
23 - 24	0000 จนถึง 6300	เลขต่อท้ายระบุเขตที่พบคำหลัก
25 - 26	ตั้งแต่ 0100	หมายเลขระบุเขตข้อมูลย่อย
27 - 28	ตั้งแต่ 0100	หมายเลขลำดับพจน์
.	.	.
.	.	.

ตาราง 4.17 แสดงรายละเอียดของแฟ้มข้อมูลเก็บรักษา

โครงสร้างข้อมูลของแฟ้มข้อมูลเก็บรายละเอียดระเบียนที่ค้นหาพบในช่วงการดึงข้อมูล

แฟ้มข้อมูลเก็บรายละเอียดระเบียนที่ค้นหาพบในช่วงการดึงข้อมูล (Temporary storage for hit lists created during retrieval หรือแฟ้ม TSF.TMP) เป็นแฟ้มข้อมูลระบบชั่วคราวที่ถูกสร้างขึ้นใหม่โดยโปรแกรม ISIS.EXE ทุกครั้งที่เรียกโปรแกรมนี้ทำงาน และเมื่อจบโปรแกรม ระบบก็จะลบแฟ้มข้อมูลนี้ทิ้งไปโดยอัตโนมัติ ทำหน้าที่จัดเก็บรายละเอียดระเบียนที่ค้นหาพบโดยการใช้ทางเลือก S ของโปรแกรม ซึ่งเมื่อผู้ใช้ต้องการเก็บระเบียนเหล่านี้ไว้พิมพ์ในภายหลังก็จะเลือกทางเลือก P ซึ่งระบบจะอ่านข้อมูลจากแฟ้มนี้ไปจัดสร้างแฟ้มข้อมูลเก็บรักษาได้ รายละเอียดโครงสร้างระเบียนประกอบด้วยกลุ่มระเบียนความยาวคงที่ กลุ่มละ 508 ไบท์ แต่ละกลุ่มก็เก็บข้อมูลแต่ละหน่วยของสูตรการค้นหาที่เลือกโดยทางเลือก S ของโปรแกรม ISIS.EXE เรียงลำดับกันไปเรื่อย ๆ ลักษณะโครงสร้างแฟ้มข้อมูลดังมีรายละเอียดดังในตาราง 4.18

ไบท์ที่	ข้อมูลในรูปเลขฐานสิบหก	ความหมาย
1 - 2	ตั้งแต่ 0100 ไปเรื่อย ๆ	ระบุหมายเลขบล็อคนั้น ๆ
3 - 4	0101	
5 - 6	0000	
7 - 8	ตั้งแต่ 0100 ไปเรื่อย ๆ	จำนวนโพสติ้งภายในบล็อคนี้
9 - 10	ตั้งแต่ 0000 ไปเรื่อย ๆ	จำนวนระเบียนที่ใช้งานในบล็อค
11 - 12	ตั้งแต่ 0100 ไปเรื่อย ๆ	จำนวนระเบียนที่จองไว้ในบล็อค
13 - 14	0100 จนถึง FF7F	หมายเลขแฟ้มข้อมูลหลักระเบียนที่ 1
15 - 16	0000 จนถึง 6300	เลขต่อท้ายระบุเขตที่พบคำหลัก
17 - 18	ตั้งแต่ 0100	หมายเลขระบุเขตข้อมูลย่อย
19 - 20	ตั้งแต่ 0100	หมายเลขลำดับพจน์
21 - 22	0100 จนถึง FF7F	หมายเลขแฟ้มข้อมูลหลักระเบียนที่ 2
23 - 24	0000 จนถึง 6300	เลขต่อท้ายระบุเขตที่พบคำหลัก
25 - 26	ตั้งแต่ 0100	หมายเลขระบุเขตข้อมูลย่อย
27 - 28	ตั้งแต่ 0100	หมายเลขลำดับพจน์
.	.	.

ตาราง 4.18 แสดงรายละเอียดของแฟ้มข้อมูล TSF.TMP

โครงสร้างข้อมูลของแฟ้มข้อมูลเก็บรายละเอียดการค้นหาข้อมูล

แฟ้มข้อมูลเก็บรายละเอียดการค้นหาข้อมูล (Temporary storage for query elements หรือ แฟ้ม TSQ.TMP) เป็นแฟ้มข้อมูลระบบชั่วคราวที่ถูกสร้างขึ้นใหม่โดยโปรแกรม ISIS.EXE ทุกครั้งที่เรียกโปรแกรมนี้ทำงาน และเมื่อจบโปรแกรม ระบบก็จะลบแฟ้มข้อมูลนี้ทิ้งไปโดยอัตโนมัติ ทำหน้าที่จัดเก็บรายละเอียดการค้นหาข้อมูลในการประมวลผลครั้งหนึ่ง ๆ ด้วยทางเลือก S ซึ่งในการประมวลผลครั้งเดียวกันนี้ ผู้ใช้สามารถเรียกรายละเอียดการค้นหาเดิมมาประมวลผลอีก หรือมาปรับปรุงแก้ไขได้ รายละเอียดโครงสร้างระเบียนประกอบด้วยกลุ่มระเบียนความยาวคงที่กลุ่มละ 270 ไบท์ แต่ละกลุ่มก็เก็บชื่อฐานข้อมูล, สูตรการค้นหา และจำนวนโพสติ้งที่ค้นพบของสูตรนั้น ๆ ลักษณะโครงสร้างแต่ละระเบียนมีรายละเอียดดังในตาราง 4.19

ไบท์ที่	ข้อมูลในรูปเลขฐานสิบหก	ความหมาย
1 - 10	ตั้งแต่ 20_{16}	ชื่อฐานข้อมูลที่เรียกค้นหา
11 - 12	ตั้งแต่ 0000	จำนวนโพสติ้ง
13 - 14	ตั้งแต่ 0100	ความยาวของสูตรที่ใช้ค้นหา
15 -270	ตั้งแต่ 00_{16}	สูตรที่ใช้ค้นหา

ตาราง 4.19 แสดงรายละเอียดของแฟ้มข้อมูล TSQ.TMP

สูตรที่ใช้ค้นหา ระบบจองเนื้อที่ไว้ 256 ไบท์คงที่ ไม่ว่าจะบันทึกสูตรจำนวนเท่าใดก็ตาม แต่ระบบก็มีเขตระบุความยาวสูตรที่ใช้ค้นหาในไบท์ที่ 13 - 14 ด้วยเพื่อให้ประมวลผลได้รวดเร็วขึ้น โดยไม่ต้องลบค่าในพื้นที่ที่ไม่ใช้งาน ส่วนชื่อฐานข้อมูลนั้นระบบจองเนื้อที่ไว้ 10 ไบท์ตายตัว โดยจะใส่ช่องว่างทางขวามือให้ถ้าชื่อยาวไม่ถึง 10 ไบท์ ซึ่งชื่อที่เก็บไว้นี้ มีไว้แสดงเมื่อผู้ใช้เรียกดูสูตรที่เคยบันทึกไว้ในการประมวลผลครั้งนั้นเท่านั้น ถ้าผู้ใช้เปลี่ยนไปเปิดฐานข้อมูลอื่นอยู่และต้องการประมวลผลสูตรเดิมโดยใช้ฐานข้อมูลเดิม ผู้ใช้จำเป็นที่จะต้องเปลี่ยนฐานข้อมูลโดยเลือกทางเลือก C เสียก่อนให้เป็นฐานข้อมูลเดิม แล้วจึงเรียกประมวลผลด้วยทางเลือก G

โครงสร้างข้อมูลของแฟ้มข้อมูลสำรองของแฟ้มข้อมูลหลัก

แฟ้มข้อมูลสำรองของแฟ้มข้อมูลหลัก (Master File Backup หรือ แฟ้ม xxxxxx.BKP) เป็นแฟ้มข้อมูลระบบที่เก็บสำเนาของระเบียนข้อมูลจากแฟ้มข้อมูลหลัก โดยเก็บเฉพาะระเบียนที่ใช้งานจริงเท่านั้น ระเบียนที่ถูกลบ หรือระเบียนเดิมที่ถูกแก้ไขและค้างอยู่ในแฟ้มข้อมูลหลักจะไม่ถูกย้ายมา โครงสร้างแฟ้มข้อมูลมีความคล้ายคลึงกับแฟ้มข้อมูลหลักทุกประการ แบ่งออกเป็น 3 ส่วนใหญ่ ๆ ได้แก่ ส่วนหัวของแฟ้มข้อมูลหลัก, ส่วนหัวของระเบียน และส่วนระเบียน

1. ส่วนหัวของแฟ้มข้อมูลสำรองของแฟ้มข้อมูลหลัก

ที่ต้นแฟ้มข้อมูลสำรองของแฟ้มข้อมูลหลักของทุกฐานข้อมูล ระบบจะจัดเตรียมเนื้อที่ไว้จำนวน 18 ไบท์ เพื่อเก็บข้อมูลเกี่ยวกับแฟ้ม ดังในตาราง 4.20

ไบท์ที่	ข้อมูลในรูปเลขฐานสิบหก	ความหมาย
1 - 2	0000	หมายเลขแฟ้มข้อมูลหลักลำดับถัดไป
3 - 4	0000 จนถึง FF7F	
5 - 6	0000	
7 - 8	0000	
9 - 10	0000	
11 - 12	ตั้งแต่ 0000	จำนวนระเบียนที่อ่านจากแฟ้มข้อมูลหลัก
13 - 14	0000	
15 - 16	ตั้งแต่ 0000	จำนวนบล็อคข้อมูลของแฟ้มข้อมูลหลัก
17 - 18	0000	

ตาราง 4.20 แสดงรายละเอียดส่วนหัวของแฟ้มข้อมูลสำรองของแฟ้มข้อมูลหลัก

2. ส่วนหัวของระเบียน

ถัดจากส่วนหัวของแฟ้มข้อมูลสำรองของแฟ้มข้อมูลหลัก จะประกอบด้วยส่วนหัวของระเบียนซึ่งแบ่งออกเป็น 7 เขตข้อมูล ความยาวคงที่ในทุกระเบียน แต่ละเขตมีขนาด 2 ไบท์ดังมีรายละเอียดดังในตาราง 4.21

ไบท์ที่	ข้อมูลในรูปเลขฐานสิบหก	ความหมาย
1 - 2	0000 จนถึง FF7F	หมายเลขแฟ้มข้อมูลหลักของระเบียน
3 - 4	ตั้งแต่ 0100	ความยาวระเบียนทั้งสิ้น
5 - 6	ตั้งแต่ 0000	เซกเตอร์ของตำแหน่งระเบียนเดิม
7 - 8	ตั้งแต่ 0000	ตำแหน่งระเบียนเดิมภายในเซกเตอร์
9 - 10	ตั้งแต่ 0000	ตำแหน่งฐานของข้อมูลจากต้นระเบียน
11 - 12	ตั้งแต่ 0000	จำนวนเขตของระเบียนนั้น ๆ
13 - 14	0000 หรือ 0100	รหัสระบุว่าระเบียนถูกลบหรือไม่ ถ้าเป็น 0100 แสดงว่าถูกลบ

ตาราง 4.21 แสดงรายละเอียดส่วนหัวของระเบียน

3. ส่วนระเบียน

ถัดจากส่วนหัวของระเบียน จะตามด้วยส่วนระเบียน ซึ่งเป็นกลุ่มของเขตข้อมูลซึ่งมีความยาวแปรผันตามจำนวนเขตที่บันทึกในระเบียนนั้น ๆ นอกจากนี้ ข้อมูลในแต่ละเขตก็ยังมีความยาวแปรผันได้อีกด้วย ส่วนระเบียนประกอบด้วยเขต 2 กลุ่มใหญ่ ๆ ได้แก่ กลุ่มระบุเขต และกลุ่มข้อมูล

3.1 กลุ่มระบุเขต เป็นกลุ่มของเขต 3 เขตที่มีความยาวคงที่ คือ เขตละ 2 ไบท์ รวมเป็น 6 ไบท์ต่อกลุ่ม แต่ละเขตมีรายละเอียดดังในตาราง 4.22

ไบท์ที่	ข้อมูลในรูปเลขฐานสิบหก	ความหมาย
1 - 2	0100 จนถึง 6300	หมายเลขต่อท้ายของเขต
3 - 4	ตั้งแต่ 0000	ค่าออฟเซทจากตำแหน่งฐานของข้อมูลจากต้นระเบียน
5 - 6	ตั้งแต่ 0100	ความยาวของเขตข้อมูล

ตาราง 4.22 แสดงรายละเอียดกลุ่มระบุเขต

ถ้าระเบียนประกอบด้วย 4 เขต ก็จะมีกลุ่มระบุเขต 4 กลุ่ม ขนาดกลุ่มละ 6 ไบท์อยู่เรียงต่อกัน รวมเป็น 24 ไบท์ เป็นต้น ตัวอย่างรายละเอียดของกลุ่มระบุเขตดังแสดงในภาคผนวก

3.2 กลุ่มข้อมูล อยู่ต่อจากกลุ่มระบุเขต เป็นชุดของอักขระต่อเนื่องกันไปเรื่อย ๆ จนจบระเบียน ระบบสามารถแยกข้อมูลแต่ละเขตได้โดยทำการวิเคราะห์จากหัวระเบียนและกลุ่มระบุเขต ดังนั้น กลุ่มข้อมูลนี้จึงเป็นเขตที่มีความยาวแปรผัน และข้อมูลจะถูกบันทึกในแฟ้มข้อมูลหลักในลักษณะเรียงลำดับต่อท้ายแฟ้มไปเรื่อย ๆ

โครงสร้างข้อมูลของแฟ้มข้อมูลเชื่อมโยงที่ยังไม่ได้เรียงลำดับ

แฟ้มข้อมูลเชื่อมโยงที่ยังไม่ได้เรียงลำดับ (Unsorted Link File หรือแฟ้ม xxxxxx.LNU) เป็นแฟ้มข้อมูลผลลัพธ์ที่ได้จากการสร้างแฟ้มข้อมูลหกกลับในช่วงแรก คือ ระบบจะอ่านข้อมูลจากแฟ้มข้อมูลหลัก เพื่อเลือกคำหลักตามเงื่อนไขในตารางเขตข้อมูลที่เลือก จากนั้นจึงไปเรียงลำดับจากน้อยไปมากสร้างเป็นแฟ้มข้อมูลเชื่อมโยงที่เรียงลำดับแล้วอีกครั้งหนึ่ง แล้วจึงนำไปสร้างทรีสำหรับการค้นหา โครงสร้างแฟ้มข้อมูลนี้ประกอบด้วยระเบียนความยาวคงที่ 5 เขตข้อมูล ระเบียนละบรรทัด ดังมีรายละเอียดดังในตาราง 4.23

เขตที่	จำนวนไบท์ของข้อมูล	ข้อมูล
1	30	คำหลักที่อ่านจากแฟ้มข้อมูลหลัก
2	4	หมายเลขแฟ้มข้อมูลหลักของระเบียน
3	2	เลขต่อท้ายระบุเขตที่พบคำหลัก
4	2	หมายเลขระบุเขตข้อมูลย่อย
5	2	หมายเลขลำดับพจน์

ตาราง 4.23 แสดงรายละเอียดระเบียนของแฟ้มข้อมูลเชื่อมโยงที่ยังไม่ได้เรียงลำดับ

โครงสร้างข้อมูลของแฟ้มข้อมูลเชื่อมโยงที่เรียงลำดับแล้ว

แฟ้มข้อมูลเชื่อมโยงที่เรียงลำดับแล้ว (Sorted Link File หรือแฟ้ม xxxxxx.LNK) เป็นแฟ้มข้อมูลผลลัพธ์ที่ได้จากการเรียงลำดับแฟ้มข้อมูลเชื่อมโยงที่ยังไม่ได้เรียงลำดับ หรือได้จากการทำสำเนาแฟ้มข้อมูลหกกลับ ซึ่งระบบจะอ่านข้อมูลจากทรีมาสร้างเป็นแฟ้มนี้ขึ้น มีลักษณะโครงสร้างแฟ้มเหมือนกันกับแฟ้มข้อมูลเชื่อมโยงที่ยังไม่ได้เรียงลำดับทุกประการ เพียงแต่ข้อมูลจะถูกเรียงลำดับจากค่าแอสกีน้อยไปมาก

โครงสร้างข้อมูลของแฟ้มข้อมูลเอนีย์

แฟ้มข้อมูลเอนีย์ (ANY File) เป็นแฟ้มข้อมูลที่ผู้ใช้สร้างขึ้นเพื่อใช้ประโยชน์ในการค้นหาข้อมูลที่จัดเป็นกลุ่มเดียวกันไว้ล่วงหน้า เวลาค้นก็เพียงใส่คำว่า ANY แล้วตามด้วยพจน์ที่ต้องการ ระบบก็จะไปเปิดแฟ้มนี้แล้วค้นหาข้อมูลที่อยู่ในกลุ่มเดียวกันนี้มาแสดงให้ มีโครงสร้างเป็นแฟ้มข้อมูลแอสกีธรรมดา บรรทัดละระเบียน ในแต่ละระเบียนแบ่งออกเป็น 2 เขต คือ เขตเทอมที่เป็นชื่อที่กำหนดความหมาย และเขตความหมาย โดยเขตเทอมที่เป็นชื่อที่กำหนดความหมาย ให้บันทึกที่คอลัมน์ 1 มีคำว่า ANY นำหน้า แล้วตามด้วยเทอม โดยให้พิมพ์ชิดซ้าย ส่วยเขตความหมายให้พิมพ์ที่คอลัมน์ 31 และพิมพ์ชิดซ้ายเช่นกัน ตัวอย่างข้อมูลเป็นดังภาพ 4.19

```
ANY SCANDINAVIA          SCANDINAVIA
ANY SCANDINAVIA          DENMARK
ANY SCANDINAVIA          FINLAND
ANY SCANDINAVIA          GREENLAND
ANY SCANDINAVIA          ICELAND
ANY SCANDINAVIA          NORWAY
ANY SCANDINAVIA          SWEDEN
ANY HOBBY                SWIMMING
ANY HOBBY                PLAYING GOLF
ANY HOBBY                FISHING
ANY HOBBY                READING
```

ภาพ 4.19 แสดงตัวอย่างข้อมูลในแฟ้มข้อมูลเอนีย์

จากตัวอย่างในภาพ จะเห็นว่ามีการกำหนดเทอม SCANDINAVIA ซึ่งระบบจะค้นว่าระเบียนใดมีชื่อประเทศใดประเทศหนึ่งใน 7 ประเทศที่ระบุก็จะถือว่าเป็นระเบียนที่ต้องการ ซึ่งเป็นการประหยัดเวลาในการบันทึกสูตรการค้นหาข้อมูล ไม่ต้องบันทึกชื่อแต่ละประเทศแล้วใช้คำเชื่อม OR ก็สามารถค้นหาได้เช่นกัน

โครงสร้างข้อมูลของแฟ้มข้อมูลคำหยุด

แฟ้มข้อมูลคำหยุด (Stopword File หรือแฟ้ม xxxxxx.STW) เป็นแฟ้มข้อมูลที่ใช้ร่วมกับตารางเขตข้อมูลที่เลือกในการสร้างคำหลักสำหรับแฟ้มข้อมูลหกกลับ ซึ่งผู้ใช้สามารถบันทึกคำที่ไม่ต้องการนำไปสร้างเป็นคำหลัก เก็บไว้ในแฟ้มข้อมูลคำหยุดนี้ได้ เวลาระบบสร้างแฟ้มข้อมูลหกกลับ ก็จะมาตรวจสอบว่า คำใดตรงกับคำในแฟ้มนี้ก็จะไม่นำไปสร้างเป็นคำหลัก มีโครงสร้างเป็นแฟ้มข้อมูลแอสกีธรรมดา บรรทัดละคำ โดยต้องเริ่มต้นที่คอลัมน์ที่หนึ่ง คำหยุดต้องเรียงลำดับตามค่าแอสกีจากน้อยไปมาก และต้องเป็นตัวพิมพ์ใหญ่ทุกอักขระ ตัวอย่างข้อมูลเป็นดังภาพ 4.20

```
A
AN
AND
AS
BY
FOR
FROM
IN
INTO
ITS
OF
ON
THE
TO
UPON
WITH
```

ภาพ 4.20 แสดงตัวอย่างข้อมูลในแฟ้มข้อมูลคำหยุด

โครงสร้างข้อมูลของแฟ้มข้อมูลข่าวสารระบบ

แฟ้มข้อมูลข่าวสารระบบ (System Message File หรือแฟ้ม xMSG.MST) เป็นแฟ้มข้อมูลที่เก็บข้อความบอกบทต่าง ๆ ที่แสดงบนจอภาพขณะที่โปรแกรมกำลังทำงานต่าง ๆ และแสดงทางเครื่องพิมพ์เวลาออกรายงาน เป็นฐานข้อมูลหลักของโปรแกรมมินิ–ไมโคร ซีดีเอส/ไอซิส แบบพิเศษ คือเป็นแฟ้มข้อมูลชนิดความยาวระเบียนแปรผัน แต่ระบบจองเนื้อที่แต่ละระเบียนไว้คงที่ ระเบียนละ 512 ไบท์ จำนวนทั้งสิ้น 44 ระเบียน ในแต่ละระเบียนประกอบด้วย 10 เขตข้อมูล ซึ่งในแต่ละเขตข้อมูลก็คือ ข้อความบอกบทต่าง ๆ และข้อความต่าง ๆ นั่นเอง ด้วยวิธีนี้ทำให้ผู้ใช้สามารถเปลี่ยนข้อมูลเหล่านี้ให้เป็นภาษาต่าง ๆ ได้โดยไม่ต้องไปปรับแก้โปรแกรม เพราะระบบได้จัดเตรียมแฟ้มข่าวสารระบบภาษาต่าง ๆ รวมทั้งมีวิธีสร้างภาษาอื่น ๆ ได้อีกด้วย เนื่องจากแฟ้มนี้จะถูกเข้าถึงอยู่เกือบตลอดเวลา การเข้าถึงข้อมูลจึงไม่ได้ใช้แฟ้มข้อมูลอ้างอิงไขว้หรือแฟ้มข้อมูลหกกลับแบบฐานข้อมูลทั่ว ๆ ไป แต่ระบบจะใช้วิธีเข้าถึงโดยตรง โดยกำหนดขนาดระเบียนไว้ตายตัว ระเบียนละ 512 ไบท์ และทุกระเบียนมี 10 เขต ซึ่งระบบก็จะเข้าถึงข้อมูลได้โดยตรงเมื่อทราบว่าต้องการข้อความบอกบทที่อยู่ในระเบียนใด เขตใด ระบบก็จะคำนวณในลักษณะสัมพัทธ์กับต้นแฟ้มและต้นระเบียน จึงทำให้การเข้าถึงข้อมูลทำได้รวดเร็วกว่าฐานข้อมูลธรรมดา ส่วนโครงสร้างระเบียนภายในนั้น มีลักษณะเหมือนแฟ้มข้อมูลหลักของโปรแกรมมินิ–ไมโครซีดีเอส/ไอซิสทุกประการ เพียงแต่ทุกระเบียนมี 10 เขตข้อมูลเหมือน ๆ กัน ซึ่งเขตใดไม่ใช้ ระบบก็จะปล่อยให้ว่างไว้

เวลาเรียกใช้โปรแกรม ระบบจะเปิดแฟ้มข้อมูลนี้เสมอ อ่านและแสดงข้อความอยู่ตลอดเวลา ซึ่งนอกจากแฟ้มนี้จะใช้เก็บข้อความบอกบทแล้ว ระบบยังได้จัดเก็บค่าพารามิเตอร์ที่ใช้งานไว้ 3 ระเบียน ได้แก่

1. ในหมายเลขแฟ้มข้อมูลหลักที่ 28 เลขต่อท้าย 3 อักขระแรกในเขตนี้ใช้ระบุการแสดงหัวตารางของตารางเลือกเขตข้อมูล และตารางเขตข้อมูลที่เลือก ซึ่งมีค่าที่กำหนดไว้เดิมเป็น "=="

2. ในหมายเลขแฟ้มข้อมูลหลักที่ 28 เลขต่อท้าย 4 เป็นการกำหนดอักขระที่ใช้เป็นตัวแยกคำเมื่อสั่งพิมพ์โดยใช้โปรแกรม ISISPRT.EXE มีค่ากำหนดไว้เดิมเป็น ", "

3. ในหมายเลขแฟ้มข้อมูลหลักที่ 36 เลขต่อท้าย 1 เก็บรหัสการแสดงข้อความบนจอภาพ ตามที่ผู้ใช้สามารถแก้ไขเปลี่ยนแปลงได้โดยโปรแกรม ISISUTL.EXE

โครงสร้างข้อมูลของแฟ้มข้อมูลรายการประมวลผลระบบและแผ่นงานระบบ

แฟ้มข้อมูลรายการประมวลผลระบบและแผ่นงานระบบ หรือ แฟ้ม pctnnn.FMT (p เป็นอักษรระบุหน้า เรียงลำดับจาก A, B, C,..., c เป็นอักษรระบุภาษาที่ใช้, t เป็นอักษรระบุชนิดของแฟ้มข้อมูล (ถ้าเป็น X หมายถึงรายการประมวลผลระบบ และถ้าเป็น Y หมายถึงแผ่นงานระบบ) และ nnn เป็นชื่อที่กำหนด) เป็นแฟ้มข้อมูลที่จัดเก็บรายละเอียดของการแสดงและรับข้อมูลของรายการประมวลผลระบบและแผ่นงานระบบที่ถูกเรียกใช้โดยโปรแกรมต่าง ๆ ทั้งหกโปรแกรม ถูกออกแบบขึ้นเพื่อให้ผู้ใช้มีความคล่องตัวในการปรับปรุงแก้ไข หรือสร้างรายการประมวลผลระบบหรือแผ่นงานระบบขึ้นใช้โดยไม่ต้องแก้โปรแกรม และช่วยให้ผู้ใช้ระบบสามารถปรับแก้ให้แสดงรายการประมวลผลระบบตลอดจนให้ข้อความในแผ่นงานระบบเป็นภาษาที่ต้องการได้โดยง่าย

ลักษณะโครงสร้างข้อมูลของแฟ้มข้อมูลรายการประมวลผลระบบและแผ่นงานระบบนี้มีโครงสร้างระเบียนและเขตเหมือนกับแฟ้มข้อมูลแผ่นงานบันทึกข้อมูลในตาราง 4.4 ทุกประการ เพียงแต่ได้มีการระบุค่าต่าง ๆ ไว้ล่วงหน้าตรงตามที่โปรแกรมจะมาเรียกใช้ ถ้าต้องการปรับแก้ผู้ใช้ระบบจึงต้องใช้โปรแกรมพิเศษในการปรับแก้ นั่นคือโปรแกรม ISISUTL.EXE ซึ่งจะมีรายการให้เลือกว่าต้องการปรับแก้แฟ้มข้อมูลรายการประมวลผลระบบ หรือแผ่นงานระบบ ดังขั้นตอนในบทที่ 3 ผู้ใช้ไม่สามารถปรับแก้แฟ้มดังกล่าวโดยใช้โปรแกรม ISISDEF.EXE ซึ่งถูกออกแบบไว้สำหรับสร้างหรือบรรณาธิกรแผ่นงานบันทึกข้อมูลธรรมดาได้ และในทำนองเดียวกัน ผู้ใช้ก็ไม่สามารถใช้โปรแกรม ISISUTL.EXE ในการสร้างหรือบรรณาธิกรแผ่นงานบันทึกข้อมูลธรรมดาเช่นกัน ต้องใช้โปรแกรม ISISDEF.EXE ในการดำเนินงานดังกล่าว

เนื่องจากโปรแกรมได้มีการกำหนดตายตัวเกี่ยวกับลำดับก่อน-หลัง ตลอดจนค่าพารามิเตอร์ต่าง ๆ ดังนั้น ชื่อแฟ้มข้อมูลจะต้องตั้งให้ตรงตามกฎเกณฑ์ข้างต้น เพื่อที่โปรแกรมจะได้สามารถนำไปประมวลผลได้อย่างถูกต้อง ตัวอย่างเช่น แผ่นงานระบบที่ใช้สำหรับบันทึกข้อกำหนดการพิมพ์มาตรฐานของโปรแกรม ISISPRT.EXE ที่เป็นภาษาอังกฤษ ถูกเก็บอยู่ในแฟ้มชื่อ AEYPRT.FMT ถ้าผู้ใช้ต้องการสร้างแผ่นงานระบบดังกล่าว แต่ให้แสดงข้อความกำกับเขตข้อมูล ตลอดจนค่าที่กำหนดล่วงหน้าให้เป็นภาษาไทย ผู้ใช้ระบบก็จะต้องทำสำเนาแฟ้มเดิม เก็บในแฟ้มใหม่ซึ่งต้องตั้งชื่อเป็น ATYPRT.EXE เมื่อ T เป็นรหัสระบุภาษาไทย แล้วเข้าไปบรรณาธิกรให้ข้อความกำกับเขตข้อมูลและค่าที่กำหนดล่วงหน้าเป็นภาษาไทยตามที่ต้องการ

โครงสร้างข้อมูลของแฟ้มข้อมูลเรียงลำดับตามมาตรฐานของ ISO 2079

แฟ้มข้อมูลเรียงลำดับตามมาตรฐานของ ISO 2079 เป็นแฟ้มข้อมูลที่ใช้ประโยชน์ในการถ่ายข้อมูลระหว่างฐานข้อมูลของโปรแกรมมินิ-ไมโคร ซีดีเอส/ไอซิส ทั้งระหว่างฐานข้อมูลในเครื่องไมโครคอมพิวเตอร์ และในเครื่องคอมพิวเตอร์ขนาดใหญ่ โดยผู้ใช้ระบบสามารถรับข้อมูลจากฐานข้อมูลอื่นได้ถ้าข้อมูลจากฐานข้อมูลนั้น ๆ ถูกแปลงให้อยู่ในรูปแบบมาตรฐานนี้เสียก่อน จากนั้น ผู้ใช้ระบบสามารถใช้ทางเลือก I ในโปรแกรม ISISXCH.EXE ในการรับข้อมูลดังกล่าว แปลงให้อยู่ในรูปแบบของแฟ้มข้อมูลหลักปกติเก็บไว้ในฐานข้อมูลได้ หรือในทางกลับกัน ผู้ใช้ระบบสามารถใช้ทางเลือก E เพื่อที่จะแปลงข้อมูลในแฟ้มข้อมูลหลักในฐานข้อมูลให้อยู่ในรูปแบบมาตรฐานดังกล่าว แล้วนำไปใช้งานอื่น ๆ ต่อไป เช่น ถ่ายข้อมูลไปยังเครื่องคอมพิวเตอร์ขนาดใหญ่ หรือถ่ายข้อมูลไปยังฐานข้อมูลอื่น ๆ ในเครื่องไมโครคอมพิวเตอร์ได้สะดวก

โครงสร้างข้อมูลของแฟ้มข้อมูลเรียงลำดับในมาตรฐาน ISO 2079 เป็นแฟ้มข้อมูลแอสกีที่ประกอบด้วยระเบียนความยาวแปรผัน ที่สามารถแบ่งออกได้เป็น 3 ส่วนใหญ่ ๆ ได้แก่ ส่วนหัวของระเบียน, ส่วนระบุข้อกำหนดเขต และส่วนข้อมูล

1. ส่วนหัวของระเบียน

ส่วนหัวของระเบียน ทำหน้าที่เก็บข้อมูลเฉพาะของระเบียนนั้น ๆ ประกอบด้วย เขตข้อมูล 11 เขต ความยาวคงที่ทุกระเบียน คือ 24 ไบท์ มีโครงสร้างดังในตาราง 4.24

เขตที่	ความยาวข้อมูล (ไบท์)	ข้อมูล
1	5	ตัวเลขฐานสิบระบุขนาดทั้งสิ้นของระเบียนนี้ รวมทั้งส่วนหัวของระเบียนและส่วนระบุข้อกำหนดเขตด้วย
2	1	ตัวเลขระบุสถานภาพระเบียน จะเป็น 0 ถ้าเป็นแฟ้มข้อมูลผลลัพธ์

ตาราง 4.24 แสดงรายละเอียดส่วนหัวของระเบียนแฟ้มข้อมูลมาตรฐาน ISO 2079

เขตที่	ความยาวข้อมูล (ไบท์)	ข้อมูล
3	4	ตัวเลขรหัสการติดตั้ง โปรแกรม ISISXCH .EXE กำหนดเป็น 0000 ถ้าเป็นผลลัพธ์
4	1	ความยาวของอินดิเคเตอร์ที่ต้นเขตข้อมูล
5	1	ความยาวของสัญญลักษณ์แบ่งเขตข้อมูลย่อย
6	5	ตำแหน่งฐานของระเบียนข้อมูล (บอกว่าส่วนข้อมูลเริ่มที่ใด)
7	3	สำหรับผู้ใช้บางระบบ ปกติเป็น 000 ในแฟ้มข้อมูลผลลัพธ์
8	1	ตัวเลขระบุจำนวนหลักของเขต "ความยาวเขตข้อมูล" ในส่วนระบุข้อกำหนดเขต
9	1	ตัวเลขระบุจำนวนหลักของเขต "ความยาวตำแหน่งเริ่มต้นเขตข้อมูล" ในส่วนระบุข้อกำหนดเขต
10	2	เก็บไว้ใช้ในอนาคต

ตาราง 4.24 แสดงรายละเอียดส่วนหัวของระเบียนแฟ้มข้อมูลมาตรฐาน ISO 2079 (ต่อ)

2. <u>ส่วนระบุข้อกำหนดเขต</u>

ส่วนระบุข้อกำหนดเขต ทำหน้าที่เก็บข้อมูลเฉพาะของแต่ละเขตในระเบียน ประกอบด้วยกลุ่มของเขตข้อมูล กลุ่มละ 3 เขต ได้แก่ เลขต่อท้าย, ความยาวของเขต และตำแหน่งออฟเซทของข้อมูลในเขต กลุ่มของเขตข้อมูลนี้มีความยาวคงที่ทุกกลุ่มภายในระเบียน จำนวนกลุ่มไม่คงที่ในแต่ละระเบียน ขึ้นกับว่าระเบียนนั้นประกอบด้วยกี่เขต มีโครงสร้างแต่ละกลุ่มดังในตาราง 4.25

เขตที่	ความยาวข้อมูล (ไบท์)	ข้อมูล
1	3	เลขต่อท้าย หรือรหัสระบุเขตข้อมูล
2	จากเขตที่ 8 ในส่วนหัวของระเบียน	ความยาวของข้อมูลในเขตข้อมูล
3	จากเขตที่ 9 ในส่วนหัวของระเบียน	ค่าออฟเซทของตำแหน่งของข้อมูลในเขต

ตาราง 4.25 แสดงรายละเอียดส่วนระบุข้อกำหนดเขตของแฟ้มข้อมูลมาตรฐาน ISO 2079

3. ส่วนข้อมูล

ส่วนข้อมูล เป็นส่วนที่จัดเก็บข้อมูลในแต่ละเขตอยู่เรียงลำดับต่อกันไปเรื่อย ๆ โดยเป็นชุดของอักขระเก็บแบบเรียงลำดับ มีเครื่องหมายแบ่งเขตข้อมูลกำกับระหว่างแต่ละเขต และมีเครื่องหมายแบ่งระเบียนอยู่ท้ายระเบียนด้วย เป็นเขตข้อมูลที่มีความยาวแปรผัน ระบบสามารถมาเข้าถึงเขตข้อมูลได้โดยเริ่มมาที่เขตแรกก่อนโดยวิเคราะห์จากส่วนหัวระเบียนและส่วนระบุข้อกำหนดเขต จากนั้นจะเข้าถึงเขตถัด ๆ ไปโดยดูจากเครื่องหมายแบ่งเขตข้อมูล และทราบว่าจบระเบียนเมื่อพบเครื่องหมายแบ่งระเบียน ซึ่งปรกติ โปรแกรม ISISXCH.EXE จะกำหนดเครื่องหมายแบ่งเขตข้อมูลและระเบียนให้เป็น "#"

ตัวอย่างระเบียนในมาตรฐาน ISO 2079 ดังแสดงในภาพ 4.21

```
00042000000000037000450000600040000#fir##
00043000000000037000450000600050000#kdsj##
```

ภาพ 4.21 แสดงตัวอย่างระเบียนในมาตรฐาน ISO 2079

จากภาพ 4.21 ประกอบด้วย 2 ระเบียน ระเบียนแรกมีความยาวทั้งสิ้น 42 ไบท์ ตำแหน่งฐานของระเบียนคือไบท์ที่ 37 ความยาวเขตข้อมูลคือ 4 หลัก ความยาวตำแหน่งเริ่มต้นเขตข้อมูลคือ 5 หลัก เขตข้อมูลแรกมีเลขต่อท้ายเป็น 6 ข้อมูลยาว 4 ไบท์ และเริ่มข้อมูลที่ออฟเซท 0 ข้อมูลได้แก่ "fir" นำหน้าและปิดท้ายด้วยเครื่องหมายแบ่งเขตข้อมูล "#" เมื่อจบระเบียนก็ปิดท้ายด้วยเครื่องหมายแบ่งระเบียน "#" เช่นกัน แล้วขึ้นระเบียนใหม่ในบรรทัดถัดไป นั่นคือปิดท้ายระเบียนด้วยรหัส <ENTER> ($OD_{16}OA_{16}$)

ระเบียนที่สองมีความยาวทั้งสิ้น 43 ไบท์ ตำแหน่งฐานของระเบียนคือไบท์ที่ 37 ความยาวเขตข้อมูลคือ 4 หลัก ความยาวตำแหน่งเริ่มต้นเขตข้อมูลคือ 5 หลัก เขตข้อมูลแรกมีเลขต่อท้ายเป็น 6 ข้อมูลยาว 5 ไบท์ และเริ่มข้อมูลที่ออฟเซท 0 ข้อมูลได้แก่ "kdsj" นำหน้าและปิดท้ายด้วยเครื่องหมายแบ่งเขตข้อมูล "#" เมื่อจบระเบียนก็ปิดท้ายด้วยเครื่องหมายแบ่งระเบียน "#" เช่นกัน

โดยการใช้โปรแกรม ISISXCH.EXE ผู้ใช้ระบบสามารถประมวลผลโปรแกรมในลักษณะกลุ่มได้อย่างสะดวก โดยถ่ายข้อมูลจากฐานข้อมูลหนึ่ง ๆ ไปยังฐานข้อมูลอื่น ๆ ได้โดยใช้แฟ้มข้อมูลเรียงลำดับตามมาตรฐาน ISO 2079 นี้เป็นตัวกลาง และนอกจากนี้ ผู้ใช้ระบบยังสามารถนำข้อมูลจากการประมวลผลโดยโปรแกรมอื่น ๆ ที่ถูกกำหนดให้มีรูปแบบมาตรฐานดังกล่าวมาใช้ในฐานข้อมูลของโปรแกรมมินิ–ไมโครซีดีเอส/ไอซิสได้อีกด้วย