EUC 틀안의 한글 처리 연구

김 용호, 윤 홍원, 김 경석
부산 대학교 전자계산학과

A Study of the Hangul Processing in the EUC framework

Yongho Kim, Hongwon Yun, Kyongsok Kim
Department of Computer Science, Pusan National University

요 약 문

본 논문은 국내 표준 한글 부호계와 국제 표준 한글 부호계를 유닉스의 EUC 틀 안에서 수용하는 방안을 제안한다. EUC (Extended Unix Code) 는 4 개의 코드 셑 (Code Set)를 지원하는데, 그 중 3 개의 코드 셑은 멀티바이트를 지원한다. EUC 에서는, ASCII 를 빼고는 모두 MSB (most significant bit) 가 1 이어야 한다.

이러한 제약 조건 아래서, KSC 5601 완성형 한글 부호계, 두 바이트 조합형 한글 부호계, 그리고 1992 년에 국제 표준으로 확정된 ISO 10646 의 새로운 조합형과 완성형 소리마디를 EUC 틀안에서 종합적으로 지원하는 방안을 검토해 본 결과, 짧게는 새로운 조합형과 KSC 5601 을 지원하되, 길게는 새로운 조합형으로 나가는 것이 바람직하다는 결론을 얻었다.

1. 서론

현재 쏘프트웨어의 차원에서 부호계의 문제는 사실상 심각한 문제를 안고 있다. 국내에서만 해도 많은 한글 부호계가 있었으며, 그 동안 많은 시행 착오와 많은 분들의 노력으로 지금은 완성형 한글과 두 바이트 조합형 한글이 복수 표준으로 정해졌다.

한편 유닉스의 EUC [CDC 93] 에는 여러 나라의 글자를 한꺼번에 지원할 수 있도록 여러 바이트를 쓰는 몇개의 코드 셑이 있다. 유닉스를 쓰는 많은 계산기에서는 이미 EUC 를 지원하고 있다. 본 논문은, 국내 표준 한글 부호계뿐만 아니라, 1992 년에 국제 표준으로 채택된 ISO 10646 [ISO 93a] 부호계도 충분히 고려하여 EUC 틀 안에서 바람직한 한글 처리 방안을 제시한다. 앞으로 한글 부호계에 대한 국제적 흐름을 잘 이해해서 대처해 나가는 것이 바른 방향이라고 본다.

2. 용어

여기서는 이 논문에서 쓸 용어를 정의하여, 용어의 혼란을 막고자 한다.

2.1 글자

글자는 영어의 'character' 또는 'letter' 에 해당하는 것으로, 한글에서 첫소리 글자 'ㄱ', 'ㄴ' 등, 가운뎃소리 글자 'ㅏ', 'ㅑ' 등, 그리고 끝소리 글자 'ㄱ', 'ㄴ' 등을 가리킨다. '가'와 같은 소리마디를 우리는 보통 "자" 라고 하여, 글자가 소리마디를 가리키는지 영어의 letter 를 가리키는 지에 대한 혼란을 없애기 위해서 글자를 이와 같이 정의한다.

2.2 소리마디

소리마디 (syllable) 는 한자말로 음절에 해당하며, 한글에서는 '가', '각' 등을 가리킨다.

한글에서 한 소리마디를 이루는 글자를 각각 첫소리 글자, 가운뎃소리 글자, 끝소리 글자로 부른다.

2.3 국제 표준 한글 부호계

국제 표준 한글 부호계는 1992 년에 확정되고, 1993 년에 문서로 배포된 ISO 10646 에 들어 있는 한글 부호계를 가리킨다. 국제 표준 한글 부호계에서는 한글이 완성형과 "새로운 조합형" 의 두가지 방식이 모두 들어가 있는데, 새로운 조합형은 이른 바 두 바이트 조합형 (또는 상용 조합형) 과는 다른 것이다. 한편, 유니코드 (Unicode) [Unic 91] 가 ISO 10646 BMP (Basic Multilingual Plane) (UCS-2 : Univesal Multiple-Octet Coded Character Set-2)와 하나로 통합됨으로써 이 두 부호계는 현재 거의 같아졌으며 앞으로 ISO 10646/Unicode 를 지원하는 프로그램이 개발될 것이다.

2.4 새로운 조합형

새로운 조합형 방식은, 완성형처럼 소리마디 (Syllable: 예, 가, 각)를 하나치로 해서 부호화하지 않고, 글자 (보기, 첫소리 글자 ㄱ, ㄴ, ...) 를 하나치로 해서 부호화한 뒤, 한 소리마디는 이 글자를 조합해서 나타내는 방식이다 [KimK 88, KimK 90a, 90b, 92]. 글자를 조합해서 소리마디를 나타내는 방식이라고 해서 조합형이라고 하고, 두 바이트 조합형과는 다르기 때문에 "새로운"이라는 말을 붙였다]. 새로운 조합형은, 240 개의 글자로 요즘 한글뿐만 아니라 옛 겹글자, 방점등을 포함하여 옛한글도 완벽하게 지원한다. 옛한글이 우리의 글자인데도 불구하고, 이때까지는 주로 요즘 한글만 생각하다 보니, 5601 완성형이나 두바이트 조합형이 나왔다. 하지만, 한글 사전의 전산화, 옛 한글 문서의 전산화, 국어 교과서에 나오는 옛한글을 생각해 볼 때, 새로운 조합형에서 옛한글을 지원하는 것은 중요한 뜻을 지닌다. 또한, 이 때까지 국내에서 논란이 되어 왔던 완성형대 두 바이트 조합형 사이의 대결 문제를 제 3 의 안으로 한꺼번에 해결한 것으로 평가받고 있다.

3. Extended UNIX Code (EUC) 란?

유닉스에서 여러 나라의 글자를 지원하기 위해서 쓰는 EUC 는 4 개의 코드 셑을 지원한다. 기본 코드 셑 (Primary Code Set) 인 CS0 는 7 비트 아스키 (ASCII) 이고, 글자를 나타내는 바이트의 MSB (most significant bit) 는 늘 0 이다. 세가지 종류의 추가된 코드 셑 (Supplementary Code Set) 인 CS1, CS2 그리고 CS3 는 멀티 바이트를 지원하며, 글자를 나타내는 바이트의 MSB 는 늘 1 이다. CS2 의 글자임을 구분하기 위해서는 처음에 무조건 SS2 (0x8e: Single Shift Character) 를 붙이고, 그 다음에 다른 바이트가 와서 특정한 글자임을 지정한다. CS3 의 경우에는 CS2 의 SS2 대신에 SS3 (0x8f) 를 쓴다. 다음 표 1 은 코드 셑의 구성을 나타낸다. 표 2 는 EUC 의 특수 문자 (Special Character)의 구성을 나타낸 것이다. 이 구조는 ISO 2022 을 따른 것이다.

표 1. EUC 의 구성

(여기서 SS2 는 0x8e, SS3 는 0x8f 이다.)

Code Set	EUC Representation
0	0xxxxxxx
1	1xxxxxxx [1xxxxxxx [...]]
2	SS2 1xxxxxxx [1xxxxxxx [...]]
3	SS3 1xxxxxxx [1xxxxxxx [...]]

표 2. EUC 의 특수 글자 (Special Character) 의 구성

Control Character	EUC Representation
Space	00100000
Delete	01111111
Control Character Set 0 (C0)	000xxxxx
Control Character Set 1 (C1)	100xxxxx

추가된 코드 셑 (Sunpplementary Code Set)에서 아스키 코드 외의 언어를 표현하는 코드를 동시에 지원할 수 있다. 먼저 그 언어에 해당하는 로케일 (Locale)을 설정하고 난 뒤에 사용하고자 하는 코드 셑을 지정하고 몇 바이트를 이용할 것인가를 지정한다. 그러한 것을 지정하는 것은 cswidth 의 파라메터를 조정함으로써 얻어지는데 그 포맷은 다음과 같다.

cswidth x1[:y1], x2[:y2], x3[:y3]

명시된 각 파라메터를 살펴 보면,

x1: CS1 에서의 바이트 수.

x2: CS2 에서의 바이트 수, SS2 는 그 수에서 제외된다.

x3: CS3 에서의 바이트 수, SS3 는 그 수에서 제외된다.

y1, y2, y3: CS1, CS2 그리고 CS3 의 글자가 디스플레이되는 너비.

추가된 코드 셑 (Supplementary Code Set)을 이용하는 경우는 각 코드 셑의 부호값의 길이가 일정하지 않기 때문에, 그 코드로 저장된 파일을 읽거나 쓸 때 효율성이라는 면에서 심각한 문제가 있다. 따라서 C 프로그래밍 언어에서는 wide-character 라는 새로운 데이타 타입으로 멀티바이트 글자를 지원한다. 이 데이타 타입은 ANSI C 표준에서도 wchar_t 로 정의 되어 있다. EUC 로 나타낸 부호값의 길이가 일정하지 않은데 비해서, 이를 길이가 일정한 wchar_t 로 나타내어, 프로그램에서 문자열을 비교한다거나 쏘트를 하게 된다. wchar_t 를 지원하는 함수 (function) library 가 이미 유닉스 (BSD, SVR3, SVR4) 에서 많이 개발되어 있다.

4 EUC 틀안에서 국내 표준 한글 부호계를 수용하는 방안

국내 표준 한글 부호계로는 두 바이트 조합형 한글과 KSC 5601 완성형 2350 소리마디 가 있는데 다음에서 각각에 대해서 살펴 보자.

4.1 EUC 틀안에서 KSC 5601 완성형의 수용 방안

KSC 5601 은 MSB 가 늘 1 이기 때문에 CS1, CS2 또는 CS3 에서 각각 그대로 수용할 수 있다. 이 부호값은 바로 G1 에 속하기 때문에 다른 것은 신경쓸 필요가 없다. 보기를 들어, CS2 에 KSC 5601 을 쓰려면 x2 와 y2 를 2 로 설정하고 해당 부호값을 아무런 변환없이 그대로 쓴다. x2 를 2 로 설정하는 것은 5601 이 2 바이트 체계이기 때문이고, y2 를 2 로 하는 것은 인쇄할 때 한글 한 소리마디는 영어 두 글자에 해당하는 자리를 차지하기 때문이다.

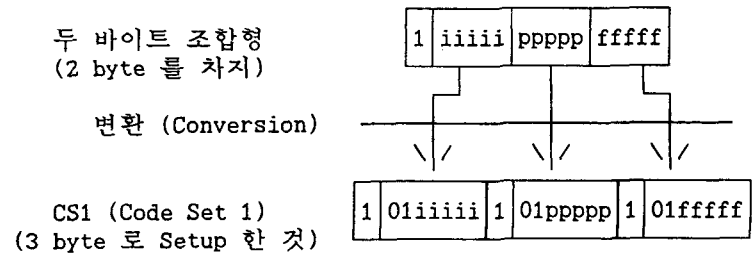
4.2 EUC 틀안에서 두 바이트 조합형 (또는 상용 조합형) 의 수용 방안

두 바이트 조합형에서는 MSB 를 쓰는데 비해서, EUC 의 CS1, CS2, CS3 에서는 MSB 가 늘 1 이어야 하므로, 두 바이트 조합형을 그대로 수용할 수는 없다. 다음에서 두바이트 조합형을 EUC 틀에서 지원하는 방안을 구체적으로 제시한다.

4.2.1 두 바이트 조합형을 EUC 의 CS1 에서 수용하는 방안

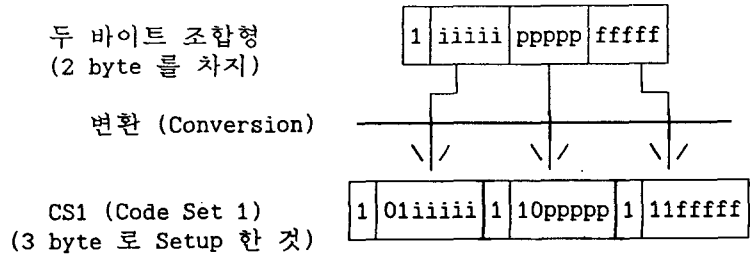
두 바이트 조합형을 CS1 에 수용하기 위해서는 x3 은 3 으로, y1 은 2 로 한다. 한편, 완성형은 CS2 나 CS3 에서 지원할 수 있다. 두 바이트 조합형에서 한글 소리마디가 비트 열로 나타내어 1iiiiippppfffff (16 비트) 라고 하자. MSB 는 늘 1 이며, 첫소리 글자 (syllable-initial) 는 iiiii 로, 가운뎃소리 글자 (syllable-peak) 는 ppppp 로, 끝소리 글자 (syllable- final) 는 fffff 로 각각 5 비트씩 차지한다.

가. 첫번째 방안



로 한다. MSB 다음의 두 비트는 아무런 뜻이 없으므로, 01 이 아닌, 10, 11 가운데 아무 것으로 해도 된다.

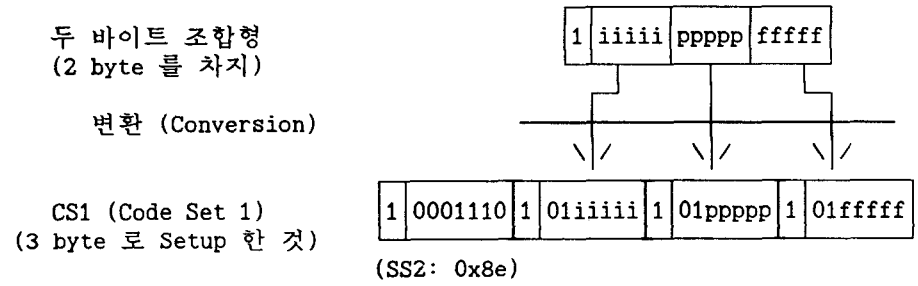
나. 두번째 방안



로 한다. MSB 다음의 두 비트로 첫소리, 가운뎃소리, 끝소리 글자를 구별하게 한다. 이렇게 해도 EUC 에서는 크게 달라지는 점은 없으며, 단지 부호계 자체로 볼 때는 읽기 (readability) 가 좋다는 장점이 있다.

4.2.2 두 바이트 조합형을 EUC 의 CS2 (또는 CS3) 에서 수용하는 방안

두 바이트 조합형을 CS2 에서 지원할 경우, x2 는 3 으로 하고, y2 는 2 로 둔다. 한편, 완성형은 CS1 에서 지원할 수 있다. 한 소리마디마다 첫소리 글자 앞에 한 바이트씩 SS2 (0x8E) (또는 SS3: 0x8F) 를 붙이면 CS2 (또는 CS3) 임을 알 수 있다. 한 소리마디를 나타내기 위해서는 모두 네 바이트가 있어야 한다. 다음은 CS2 의 경우다. CS3 이면 SS2 대신에 SS3 (0x8f) 로 대체하면 된다.



4.2.3 두 바이트 조합형과 ISO 10646 의 새로운 조합형

EUC 방식에서 두 바이트 조합형을 지원하느니, 차라리 새로운 조합형을 지원하는 것이 훨씬 낫다고 본다. 새로운 조합형은 두 바이트 조합형을 완전히 지원할 뿐만 아니라, 두 바이트 조합형이 지원하지 못하는 옛한글까지 완전하게 지원할 수 있기 때문이다. 자세한 것은 나중 뒤에서 따로 다루기로 한다.

5 EUC 틀안에서 국제 표준 한글 부호계를 지원하는 방안

국제 표준 한글 부호계에는 새로운 조합형을 위한 240 글자, KSC 5601 의 2,350 소리마디 완성형, KSC 5657 의 1,930 소리마디 완성형 그리고 어느 KSC 에도 안 들어가 있는 2,376 소리마디 완성형이 있다.

5.1 새로운 조합형

ISO 10646 의 새로운 조합형에서는 MSB 를 쓰는데 비해서, EUC 의 CS1, CS2, CS3 에서는 MSB 가 늘 1 이기 때문에 EUC 틀 안에서 새로운 조합형을 그대로 수용할 수는 없다. 또한 ISO 10646 의 새로운 조합형은 한 글자를 표현하기 위해서 2 바이트를 쓰기 때문에 한 소리마디는 네 바이트이거나 여섯 바이트인데 비해서, EUC 에서는 길이기 일정해야 한다는 문제점이 있다. 이제 구체적인 방법을 보자.

5.1.1 새로운 조합형을 CS1 에서 수용하는 방안

새로운 조합형을 CS1 에서 지원할 경우, 완성형과 두 바이트 조합형은 CS2 나 CS3 에서 지원하든지, 아니면 아예 지원할 수 있다. 새로운 조합형은 두 바이트조합형이 지원하는 표준 한글을 모두 지원할 뿐만 아니라 옛 한글까지도 표현할 수 있으므로 두 바이트 조합형을 지원하느니, 차라리 10646 새로운 조합형을 지원하는 것이 낫다.

새로운 조합형을 CS1 에서 지원할 경우, x1 은 3 으로 하고, y2 는 2 로 한다. 앞에서 말한 바와 같이 새로운 조합형은 한 글자에 해당하는 부호값의 크기가 2 바이트이다. 따라서 한글 한 소리마디를 첫소리 글자 (syllable-initial) 는 0xIIII 로, 가운뎃소리 글자 (syllable-peak) 는 0xPPPP 로, 끝소리 글자 (syllable-final) 는 0xFFFF 로 각각 나타낼 수 있다. 여기서 한글 소리마디는 '가' 와 같이 첫소리 글자와 가운뎃소리 글자만 오는 경우에는 0xIIIIPPPP (4 바이트) 로 나타낼 수 있고, '각' 과 같이 끝소리 글자까지 오는 경우에는 0xIIIIPPPPFFFF (6 바이트)로 나타낼 수 있다. 기본적으로 완전한 소리마디를 가정하면 다음과 같이 부호값을 변환해서 EUC 틀에 맞출 수 있다. 즉,

첫소리 글자는	0xXX, (= 0xIIII - 0x1100 + 0xa0)
가운뎃소리 글자는	0xYY, (= 0xPPPP - 0x1160 + 0xa0)
끝소리 글자는	0xZZ, (= 0xFFFF - 0x11a8 + 0xa0)

로 한다. 두 바이트로 나타내어져 있는 한글 글자를 한 바이트로 나타내기 위해서, 첫소리 글자의 경우 0x1100 (첫소리 글자 기역의 부호값) 을 빼고 나서, 0xa0 을 더해서, 첫소리 글자 기역의 값이 0xa0 가 되도록 한다. 다시 말해서, C0, G0, C1 을 피하고, G1 을 쓰도록 한다. 가운뎃소리 글자와 끝소리 글자의 경우도 이와 같이 하면 된다. 0xIIII 의 범위는 0x1100..115f, 0xPPPP 의 범위는 0x1160..11a2, 0xFFFF 의 범위는 0x11a8..11f9 이므로, 0xXX, YY, ZZ 의 범위는 모두 0xa0..ff 가 된다. 끝소리 글자가 없는 소리마디도 많은데, 그럴 때는 끝소리 글자에 해당하는 0xZZ 에 끝소리 글자가 없다는 것을 나타내는 부호값을 넣어 두면 된다. 다만 문제는, 현재 10646 에는 그런 목적으로 쓸 수 있는 한글 글자가 정의되어 있지 않아서, 앞으로 이에 대한 연구가 필요하다.

첫소리 글자, 가운뎃소리 글자, 끝소리 글자는 각각 91 개 (채움까지), 67 개 (채움까지), 82 개로, 그 각각이 94 보다 적으므로 위의 방법으로 EUC 틀 안에서 수용할 수 있다. 다만, 새로 옛 겹글자를 찾아내게 될 때는, 이 방법은 쓸 수 없게 되는데, EUC 에서 새로 찾아내는 옛 겹글자까지 수용하는 것은 너무 비능률적이고 거의 불가능하다고 본다.

5.1.2 새로운 조합형을 CS2 (또는 CS3) 에서 수용하는 방안

새로운 조합형을 CS2 (또는 CS3) 에서 지원할 경우, 완성형은 CS1 에서 지원할 수 있다. 그리고 한 소리마디마다 첫소리 글자 앞에 한 바이트씩 SS2 (0x8E) (또는 SS3: 0x8F) 를 붙여야 한다. 나머지 cswidth 를 설정하는 것과 부호 값을 변환하는 것은 CS1 에 적용한 방법과 같다.

5.2 EUC 틀안에서 10646 의 완성형 한글 부호계를 수용하는 방안

10646 의 완성형에서는 MSB 를 쓰는데 비해서, EUC 의 CS1, CS2, CS3 에서는 MSB 가 늘 1 이기 때문에 EUC 틀 안에서 10646 의 완성형을 그대로 수용할 수는 없다. 10646 에 들어 있는 세 덩어리의 완성형 소리마디에 대해 각각 살펴보자.

첫째, 5601 의 경우는 굳이 EUC 방식으로 지원하고자 하면, 10646 의 부호값을 쓰는 것 보다는 차라리 5601 의 부호값을 그대로 쓰는 것이 낫다고 본다. 10646 의 완성형 부호값을 쓰면, C0, G0, C1 의 값이 있기 때문에 그 부호값들을 G1 이 되도록 고치는 것이 아주 복잡하게 되므로 차라리 MSB 를 1 로 하고 5601 의 부호값을 그대로 쓰는 것이 바람직하다.

둘째, 5657 의 경우도 위의 5601 의 논리가 그대로 적용된다.

세째, 5601 이나 5657 에 속하지 않으면서 10646 에 들어간 2,376 소리마디를

EUC 방식으로 지원하는 것은 바람직하지 않다고 본다. 굳이 이를 CS2 나 CS3 에서 지원할 경우, 새로운 조합형 (두 바이트 조합형까지 포함해서), 5601, 5657 가운데 하나는 포기해야 하는데, 이 2,376 소리마디는 전혀 중요하지 않기 때문에, 아예 지원하지 않는 것이 좋다고 본다. 이 2,376 소리마디는 국내 표준으로 수용하지도 말고, 길게 볼 때는, 10646 에서 아예 빼버리는 것이 가장 바람직하다고 본다.

6. EUC 틀 안의 10646 과 KSC 한글 부호계의 종합적 지원 방안

EUC 틀 안에서 여러 한글 부호계를 지원하는 방안은 위에서 본 바와 같다. 대체로 그 부호계 그대로는 수용되지 않고, 한 번의 변환을 거쳐서 수용할 수 있다. 그런데, EUC 에서 지원하는 코드 셀은 ASCII 를 빼면 최고 3 개이므로, 여러 한글 부호계 가운데서 어느 부호계를 어느 코드 셀에 수용하는 것이 바람직한지에 대해서 살펴 보자

6.1 EUC 틀 안에서 각 한글 부호계 지원 방안에 대한 종합적 분석

첫째 새로운 조합형을 EUC 방식으로 지원하면 사실상 두 바이트 조합형까지 지원하는 결과가 되므로, 두 바이트 조합형은 따로 지원할 필요없이 새로운 조합형을 지원하는 것이 바람직하다고 본다.

둘째, KSC 5601 은 과거와의 호환성 때문에 당분간 지원하는 것이 좋다고 본다.

세째, 굳이 원한다면, KSC 5657 도 지원할 수가 있겠지만, 많은 사람이 쓸 것 같지는 않으며, 차라리 새로운 조합형으로 지원하는 것이 바람직하다고 본다.

네째, 5601 이나 5657 에 속하지 않으면서 10646 에 들어간 2,376 소리마디는 완성형으로 지원하는 것이 별 뜻이 없으므로, 차라리 새로운 조합형으로 지원하는 것이 바람직하다고 본다.

6.2 EUC 틀안에서 10646 및 KSC 한글 부호계의 종합적 지원 방안

EUC 틀 안에서 현실적으로 가능성이 있는 바람직한 한글 부호계 종합 지원 방안은 다음의 세 가지로 요약될 수 있다고 본다.

1) KSC 5601 완성형만 지원.

2) KSC 5601 완성형과 새로운 조합형 (두 바이트 조합형은 저절로 포함됨) 을 지원. 국내의 현실과 국제적인 흐름을 고려해서 EUC 틀 안에서 가장 바람직한 한글 부호계의 구체적인 수용 방안은 다음과 같다고 본다.

가. CS0 에서는 7-비트 아스키를 지원한다.

나. CS1 에서 새로운 조합형을 지원한다 (두 바이트 조합형은 저절로 지원된다.).

다. CS2 에서는 KSC 5601 한글을 지원한다.

3) 새로운 조합형만 (두 바이트 조합형은 저절로 포함됨) 지원.

이렇게 함으로써 사실상 가장 우선적으로 필요로 하는 한글들은 모두 지원하는 것이 된다.

6.3 EUC 는 ISO 2022 을 완전히 따르지는 못하고 있다.

현재 SS2 나 SS3 뒤에 나오는 모든 바이트의 MSB 는 EUC 에서 반드시 1 이어야 하는데, 이것은 모든 바이트의 MSB 값이 같아야 한다는 ISO 2022 규정과는 조금 다르다. 따라서 EUC 가 2022 를 따른다고 하고 있지만, 엄밀히 말하면 'almost compliant or questionably compliant' 라고 말할 수 있다. [ISO 93b] 이런 문제점을 풀기 위해서 현재 ISO/IEC JTC1/SC2/WG3 에서 2022 규정을 고치는 것을 검토하고 있는데, 거의 반대가 없을 것으로 예상되므로, 앞으로 1-2 년 안에 2022 가 고쳐질 것으로 본다. 2022 가 고쳐지면 EUC 는 2022 를 완전히 따르게 되는데, 우리도 이에 대한 검토와 연구를 해야 하겠다.

7. 결론

본 논문에서는 국내 표준 한글 부호계와 국제 표준 한글 부호계를 유닉스의 EUC 틀안에서 수용하는 방안을 살펴 보았다.

KSC 5601 완성형 한글 부호계, 두 바이트 조합형 한글 부호계, 그리고 1992 년에 국제 표준으로 확정된 ISO 10646 의 새로운 조합형과 완성형 소리마디를 EUC 틀안에서 종합적으로 지원하는 방안을 검토해 본 결과, 짧게는 새로운 조합형과 KSC 5601 을 CS1 과 CS2 에서 각각 지원하되, 길게는 새로운 조합형으로 나가는 것이 바람직하다는 결론을 얻었다.

8. 참고 문헌

[CDC 93] EP/IX Internationalization Guide, Control Data. 1993.

[ISO 93a]] ISO/IEC 10646-1:1993(E). International Standard. 1st edition. Information technology - Universal Multiple-Octet Coded Character Set (UCS) - Part 1: Architecture and Basic Multilingual Plane, May 1, 1993. ISO (the International Organization for Standardization).

[ISO 93b] ISO/IEC JTC1/SC2/WG3 N195, Proposed Enhancement to ISO 2022 to support Extended Unix Code (EUC) Encodings.

[KimK 88] A New Proposal for a Standard Hangul (or Korean Script) Code: How to Accommodate both Databases and Word Processing, Kyongsok Kim, et al. Report No. UIUCDCS-R-88-1447, Department of Computer Science, University of Illinois at Urbana-Champaign, Urbana, IL., U.S.A., August 1988, 54 pages.

[KimK 90a] "A New Proposal for a Standard Hangul (or Korean Script) Code", Kyongsok Kim. Computer Standards & Interfaces, Vol. 9, No. 3, pp. 187-202, 1990.

[KimK 90b] "Databases supporting Hangul (or Korean Script)", Kyongsok Kim. 1990 IEEE International Conference on Systems, Man and Cybernetics. pp. 485-490, Nov. 4-7, 1990. Universal City, CA., U.S.A.

[KimK 92] "A Common Approach to Designing the Hangul Code and Keyboard", Kyongsok Kim. Computer Standards & Interfaces. will be published soon.

[Unic 91] The Unicode Standard, Worldwide Character Encoding, Version 1.0, Volume 1. 1991. Addison-Wesley.