

UNIX 운영체제에서의 국제화와 한글화 방향에 관한 연구

안 선희, 김 해진, 남궁 한, 손 덕주
시스템 소프트웨어 연구실, 한국 전자 통신 연구소

A Study on the Internationalization, Localization and Direction of Koreanization on UNIX Operating System

SunHee Ahn, HeaJin Kim, Han Namgoong, DukJoo Son
System S/W Section, ETRI

요 약

본 논문에서는, 영어를 사용하지 않는 나라에서 그나라의 언어를 처리하고 문화적인 관습에 맞는 환경을 제공하는 자국어판 UNIX를 구현할때 표준화된 규격에 준하여 동일한 방식으로 구현함으로써 시스템간 호환성이 있고 동일 시스템상에서 여러언어를 처리할 수 있는 기능을 제공하는 UNIX 국제화(Internationalization)에 대해서 기술한다. 또한 국제화 표준규격을 토대로하여 한글 UNIX를 구현하는 방향을 기술하고 국제화 기능에 관한 표준화 활동 동향과 우리의 상황에 대하여 언급한다.

I. 서론

UNIX 운영체제는 1969년 Bell 연구소에서 내부연구용으로 개발되었지만 1978년 상용화된 version-7이 발표된 이후에 수많은 기술적인 발전과 변화를 거쳐 현재의 다양한 버전으로 다양한 컴퓨터 시스템 상에서 운영되고 있다. 그러나 수많은 UNIX 시스템 간에는 일반사용자 혹은 응용프로그램 수준에서도 상호 호환성이 없고 일반명령어, 시스템호출, 라이브러리의 사용 방법 및 기능까지도 일정치 않으며 개발자에 의한 시스템 통합은 더욱 불가능하다. UNIX 제품이 전 컴퓨터 시장에서 차지하는 비율이 1987년 약 18%인데 반해 1993년 경에는 33% 이상을 차지하게 될것으로 전망되므로 프로그램 호환성과 UNIX Portability 문제는 심각해질 것이다[1]. 따라서 표준화 확립은 다급한 과제라 할수 있다.

한편, UNIX의 응용범위와 시장이 넓어짐에 따라 UNIX 제품을 사용하는 각나라에서는 기존의 영어뿐만 아니라, 자국의 언어를 처리할 수 있고 자국의 관습 및 환경에 따라 서비스를 하는 UNIX의 환경기능을 새로이 요구하게 되었다.

이러한 요구에 부응하는 UNIX를 구현하는 작업을 UNIX Localization이라 한다. 얼마전까지만 해도 Localized 버전을 구현하는 일반적인 방법은 그 나라의 언어와 환경 정보를 사용해서 UNIX 소스의 부분을 수정하는 방식으로 이루어졌다. 따라서 개작된 버전은 한 나라의 요구와 환경만을 대상으로한 새로운 버전의 UNIX가 됨으로 인해 다른나라에서 그 나라의 요구를 위해 새로이 수정된 UNIX와는 호환성을 상실해 버리는 결과가 되었다. 즉 동일한 버전으로 각각 구현된 Localized UNIX는 새로운 UNIX버전으로 바뀌어 호환성을 상실하는것이다.

이러한 Localized UNIX의 예로는 우리나라의 KAE, 일본의 JAE, 독일의 GAE, 프랑스의 FAE 등이 있다.

수년전부터 정보산업계에 심각한 문제로 대두되어 활발히 연구중인 표준화란 어떠한 기능을 실현하기 위해 기본 모델을 개작하거나 동일한 기능을 수행하는 새로운 제품을 만들어낼 때 지켜야 할 국제 규격을 제정한다는 의미이다.

마찬가지로 UNIX에서도 이를 사용하는 각나라의 환경에 맞추어 각 나라 언어처리 기능과 Local conventions 문제를 처리할 때 규격화된 국제표준에 따라 구현되므로써 상호 호환성을 가지고 일정한 방식으로 개발될수 있도록 제공되는 것이 바로 UNIX Internationalization이다.

이 논문에서는, 특정 시스템을 구현하는 기술적인 사항이 아니라 UNIX Internationalization의 개념과 이를 구현하기 위한 요구 사항들을 정리한 다음, Internationalization을 위한 국제 표준화 활동 및 여기에서 제공하는 인터페이스와 규격에 대해서 논하고 그 규격을 토대로한 한글 UNIX 구현 방향에 관해서 논하고자 한다.

II. 본론

1. UNIX Family 와 표준 UNIX, UNIX System V

UNIX Family의 수많은 버전도 크게는 세가지 그룹으로 나눌수 있다. 첫째, UNIX 시스템 V 계열로서 현재 미국 시장의 약 60%를 차지하고 있으며, 둘째는 4BSD 계열의 버전들로서 미국내 시장의 약 15%를 차지한다. 세번째 그룹은 System V+4BSD 기능의 그룹으로서 미국내 시장의 약 25%를 차지하고 있다. BSD 버전들은 풍부하고도 뛰어난 기술로 과학 및 엔지니어링 분야에서 여전히 선호되고, 제3그룹의 제품들도 시장에서 널리 사용되고 있지만 비즈니스 및 상업적으로 우세한 시스템 V가 UNIX 표준으로 채택되어 AT&T에서는 UNIX 시스템의 표준 모델을 정의하고 국제 공통 기능과 각국어 기능을 정의하여 제공하게 되었다[7].

각국어기능	유럽권	아시아권
국제공통	single-byte 지원기능	multiple-byte 지원기능
기본기능	International features	
	UNIX Kernal	

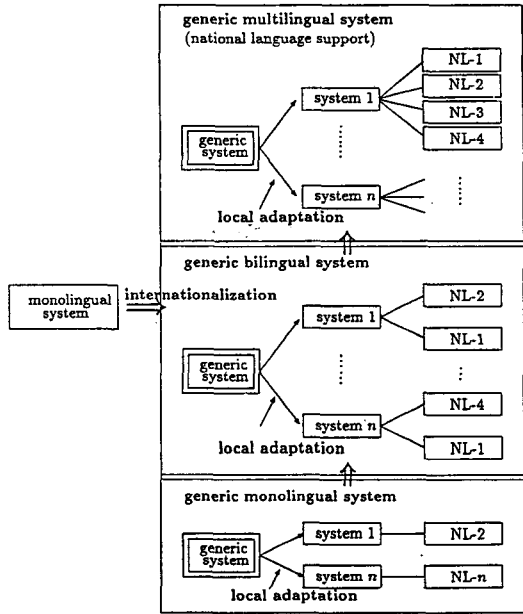
< 그림1 >. UNIX 시스템 구성 모델

2. UNIX Internationalization

UNIX에 영어와 동시에 자국의 언어도 처리할 수 있도록 컴퓨터가 이해할 수 있는 코드 문자 셋트를 확장하고 자국의 문화와 관습에 맞는 형태로 데이터를 처리할 수 있도록 재작성하는 하는것을 Localization이라 하며, Internationalization이란 위의 기능을 구현할 수 있도록 UNIX가 기반 정보 및 Interface를 제공하고 있어 어떤 특정언어 및 문화에도 의존적이지 않으면서 한 시스템이 여러 언어를 지원할 수

있는 universal한 기능을 가지고 있도록 구현되는 것을 말한다.

따라서 시스템 개발자들이 각 나라에서의 Localization을 위한 개발, 시험, 지원 등의 노력을 극소화하도록 시스템 차원의 광범위한 tool들을 제공하고 있다. 이러한 것들을 총체적으로 UNIX의 NLS (Native Language System) 인터페이스라고 한다.



NL: National Language

< 그림2 >. Scheme of Internationalization

2.1 Internationalization을 위한 기능

- 1) 문자와 코드셋 및 텍스트전송 문제 : 비영어권 문자를 처리할 수 있도록 ASCII 7-bit에 국한되어 처리 및 전송되는 코드문제는 ISO규격에 따라 8-bit와 멀티바이트 문자로 확장 가능토록함으로써 해결한다.
(* 참조 표준규정: ISO-646, ISO-2022, ISO-6937, ISO-8859 등)[2-5]

- 2) 메세지 문제 : 각나라의 언어로 메세지를 출력할 수 있도록 지원한다. 이것을 위해 SVID에서는 커널, 유틸리티, 응용 프로그램에서 출력되는 메세지와 텍스트를 별도로 저장하여 관리하고 이를 관리하는 유틸리티들을 구성하여 기존 메세지를 수정할수 있고 새로운 메세지 스트링을 구성할 수 있는 기능을 제공하는것으로 되어 있다.

X/Open (X/Open Portability Guide)에서도[6] 유사한 기능으로 프로그램 로직과는 분리된 메세지 카탈로그를

제공하며, 다른 나라의 언어로 메시지를 변환할수 있고, 프로그램 수행중에 메시지를 검색할 수 있는 기능을 제공하는 것으로 되어 있다.

3) Local Conventions : 나라의 문화적인 관습에 따라 달라지게되는 아래와 같은 local conventions 관련 사항들에 대해서 공통적인 사용형태(format)와 규칙을 정의하고, 인터페이스를 제공한다.

- 문자분류: 한 언어를 기본으로 하여 문자 클래스들을 정의할수 있는 기능을 제공해야 한다.

- 글자의 배열순서: 사용되고 있는 문자들의 collating 순서를 고려해야 한다

- 숫자표현 방식: 십진 delimiter(숫자그룹핑 기호와 소숫점이하를 표현하기 위한 기호)를 사용하여 숫자를 표현할 때 해석이 나라에 따라 다르다.

- 날짜와 시간을 표현하는 습관의 차이: 표현 순서와 delimiter 사용 관습이 다르다.

- 화폐 기호: 나라마다 화폐기호와 기호사용 위치가 다르므로 기호의 사용위치를 지정하여 사용할 수 있도록 규칙과 사용형태(format)를 제공해야 한다.

4) 유틸리티와 프로그래밍 언어 관련문제들 : 상수, 주석, identifier, keyword 등에 local 언어를 사용할 수 있도록 하는 기능이 필요하고 각 글자를 이루는 바이트수에 상관없이 문자를 나타낼 수 있는 데이터 타입이 필요하다.

5) 명령어들 : 기존 UNIX 명령어들 중에서 8번째 비트를 특별한 플래그 비트로 사용하던것, size 변환 때의 sign extension, 코드세트 사이즈에 기초한 array dimensioning, string truncation 등과 같은 코드세트 dependency 들을 제거해야 한다. 그리고 local conventions와 관련이 있는 명령어들은 convention에 부합시켜야한다.

2.2. UNIX Internationalization 기능규격

UNIX 표준화의 중심기관인 AT&T에서는 UNIX 표준규격으로 1985년 SVID(System V Interface Definition)를 처음 발표했고 1987년 추가버전을 발표했다[7]. 이것은 UNIX 시스템 V가 되기 위한 일련의 조건 및 제공되는 인터페이스들을 제시한 규격인데, 개발된 프로그램 및 시스템이 규격에 맞는지 아닌지 판별할 수 있도록 SVVS(System V Verification Suite)라고하는 검사용 tool을 제공하고 있다. SVID에서 internationalization에 관해 제시하는 사항은 다음과 같다.

1) 메시지의 서비스 및 취급

2) 나라의 문화적인 관습에 따른 사항

- 사용하는 문자들의 배열 순서

- 문자 분류 및 그룹핑

- 날짜, 시간등의 입출력 형태

- 숫자의 표현

- 화폐기호 및 표현

3) 각 바이트의 8번째 비트 clean-up

4) 코드 세트와 문자의 지원

- 코드세트에 대한 template

- 코드 세트 지시

- 비표준 코드 세트 사용

- 사용 글자의 지원

SVID 모델에 따라 국제화기능을 가진 UNIX버전으로 1987년 UNIX 시스템 V Rel.3.1이 발표되었다. 이것은 AT&T 3B2 시스템상에서 구현된 것으로 internationalization 기능으로 다음 문제를 해결하였다.

1) 8비트 코드세트를 지원하는 문제

2) 나라에 따라 시간, 날짜를 선택하여 사용할 수 있는 기능

3) 문자분류와 conversion 규칙을 선택할 수 있는 기능

또 1990년초 시스템 V, Rel.4.0이 발표될 예정인데 이것은 AT&T와 BSD계열 제품만을 생산해오던 Sun-마이크로 회사가 공동 개발한 것으로 UNIX 업계간의 동맹 및 제휴를 통한 표준화 작업의 구체적인 산출물이라 볼 수 있다 이 버전은 국제화 기능을 위해 다음과 같은 기능을 보장하여 제공하는 것으로 되어있다.

1) 문자의 collating 순서

2) 날짜, 시간 표현을 위한 conventions

3) 숫자 표현

4) 라이브러리와 유틸리티에서의 추가 지원

2.3 UNIX Internationalization을 위한 표준화 활동

UNIX 국제화 기능에 관한 표준화 활동으로는 다음과 같은 것들이 있다.

1) X/Open Working Group for Internationalization : 유럽 컴퓨터 공급업체에 의해 '84년 형성된 컨소시엄으로 SVID에 근거한 사용자, S/W 업자, 컴퓨터시스템 공급업자들간의 CAE라 불리는 공통응용환경(Common Application Environment)을 제안하였다. 이것은 시스템간 완전한 호환성을 제공하기 위한 것으로 X/Open 회원들은 CAE 규약에 따르는 제품을 개발하기로 하고 지침서인 Portability Guide를[6] 발표하였다. Internationalization을 위해서는 Working Group을 구성하여 문자와 코드세트 확장, 지정 및 전송하는 문제, 환경 변수, ANSI 표준 C와

호환성있는 기능함수들, X/Open에서 추가한 기능함수들, 국제화 기능을 가진 정규표현 처리 기능함수, 자국어로 메세지 및 에러를 처리할 수 있도록 하는 기능함수들을 정의하고, 국제화 기능을 가진 라이브러리 함수, 유틸리티, NLS 공통 헤더파일, C-프로그램의 수행중 자국어 환경을 지원할 수 있도록 하는 기능 및 인터페이스들을 제공하여 자국어 버전을 쉽게 구현할 수 있도록 하고 있다.

2) IEEE POSIX: SVID에 대한 대안으로 UNIX 사용자그룹 /usr/group이 만들었으나 기초는 SVID를 따라 약 70%가 동일하다. 그러나 현재로는 SVID가 언급하고 있는 문제의 반경도를 커버하고 있다고 한다. 이 그룹은 POSIX를 IEEE에 상정하였고 IEEE는 POSIX를 그들의 표준운영체제로 채택하였다. IEEE POSIX는 12개의 Working 그룹을 만들어 3개월마다 그룹회의를 하면서 연구중인데 그 분야는 다음과 같다.

- (1) P1003.0 : POSIX Guide
- (2) P1003.1 : System Service Interface
- (3) P1003.2 : Shell and Tool
- (4) P1003.3 : Testing Method
- (5) P1003.4 : Real-Time Extension
- (6) P1003.5 : Ada Binding
- (7) P1003.6 : Security
- (8) P1003.7 : System Administration
- (9) P1003.8 : Networking
- (10) P1003.9 : Fortran Binding
- (11) P1003.10 : Supercomputing
- (12) P1003.11 : Transaction Processing

이중 국제화 기능과 관련된 것으로는 P1003.1에서 운영체제 인터페이스와 동작, C언어 internationalization 기능, C언어 범위밖의 time-zone같은 것들을 정의하며, P1003.2에서는 collation과 정규 표현에서의 국제 기능 등 shell과 명령어에서의 처리를 위해 /usr/group subcommittee와 긴밀히 협조하면서 연구중에 있다.

또 OSF, UNIX International, NIST, X/Open에서 POSIX를 표준 운영체제로 채택하고 연구중에 있다.

3) ANSI X3J11 : 여기서는 C 언어의 international 기능을 위한 여러가지 proposal을 내놓고있다. 그중 ANSI-Programming Language C, 1986에서는 ANCI C Localization Library를 제안하여 프로그램 코드를 수정하지 않고 재컴파일하지 않고도 영어 환경처럼 각 나라 언어 환경을 이용할수 있는 localization 환경을 제공하였다.

- setlocale 기능함수를 이용하여 특정한 환경을 선택할수 있도록 하는 기능
- strftime 기능함수를 이용하여 자국 date/time정의 기능

- 표준 C 라이브러리, libc : 복수바이트 문자 라이브러리, libw.
- 비 ASCII 문자의 collation 순서를 정의하는것 등, 새로운 함수를 추가하는 일.

4) /usr/group internationalization subcommittee : UNIX 사용자 단체인 /usr/group에서는 표준화관련 technical committee 산하에 internationalization subcommittee를 두어 활동중에 있다. 이 모임은 현재 진행중인 POSIX 안이 7bit을 기준으로 제안되었기 때문에 아시아나 유럽의 언어를 모두 수용하기에 적합하지 않은고로 이에 대한 해결 방안을 모색하고 제안을 하기위한 모임이다. ANSI X3J11의 C 표준화 위원회와 밀접한 관련을 가지고 있으며 X/Open의 Portability Guide에서 담고 있는 internationalization을 많이 고려하고 있다.

5) 그외에도 표준화 단체인 OSF나 UNIX International Inc.에서도 internationalization을 위한 활동중에 있으며 범위는 위에서 기술한것과 유사하다.

3. Localization -- 한글 UNIX

UNIX가 자국어 처리기능을 위한 기반 환경을 제공하고 있는 위에서 Localized 버전을 구현하는것은 호환성과 Portability 유지 뿐 아니라, 시스템 개발시에도 많은 노력의 절감된다.

현재 전자통신 연구소에서는 행정 전산망용 주전산기를 개발하고 있는데 이 시스템은 응용 분야의 성격상 운영체제에서의 한글 지원이 필수적이다. 따라서 주전산기 운영체제에서의 한글처리 기능을 개발하기 위한 작업으로 현재는 기본 참조 모델인 UNIX 시스템 V Rcl.3.1상에서 한글화를 구현하는 단계이다. 이 버전은 SVID를 토대로한 국제화 기능을 제공하고 있으므로 주어진 규격에 따른 Localized 버전의 구현이 가능하다[8-9]. 한글 UNIX를 구현하기 위한 요구사항은 UNIX 국제화 기능(Internationalization)에서 이미 고려되어 있으며 이 기능 위에서 한국 환경으로의 적응(Adaptation)은 다음과 같이 구현되었다.

- 1) 문자의 지원: SVID에서 비영어권 문자를 지원할수 있도록 UNIX 확장문자(EUC) 사용 형태(format)를 규정하였는데 이것은 기본 문자셋으로 표준 ASCII 셋트를 지정하고 3개의 보조 셋트를 두어 필요에 따라 문자코드셋트를 확장하여 UNIX가 사용할수 있는 문자를 확장한다는 개념이다.

코드세트	EUC 표현	한국	일본	유럽
세트0	0xxxxxx	ASCII	ASCII	ASCII
세트1	1xxxxxx 1xxxxxx 1xxxxxx 1xxxxxx 1xxxxxx 1xxxxxx	한글,한자	간지	유럽글자
세트2	SS2 1xxxxxx SS2 1xxxxxx 1xxxxxx SS2 1xxxxxx 1xxxxxx 1xxxxxx		half-sized katakana	
세트3	SS3 1xxxxxx SS3 1xxxxxx 1xxxxxx SS3 1xxxxxx 1xxxxxx 1xxxxxx		Gajji 문자세트	

<그림3>. UNIX 확장코드의 사용형태(format)와 사용 예

위 <그림3>에서 처럼 한글 표준코드 체계인 KSC5601, 2-바이트 완성형 코드세트는 EUC 포맷에서 보조 세트1을 사용하고 있다[10].

한글의 지정 및 지원은 /lib/chrclass 디렉토리에 참조화일을 만들고 shell 환경 변수 CHRCLASS를 이용하여 환경 설정을 하면 된다. 그러나 완성형 코드 테이블에 없는 문자에 대한 처리와 한글이 2바이트로 구성됨에 따라 지움문자에 대한 처리는 공통적인 인터페이스로 해결될수 없으므로 별도의 처리가 필요하다[11].

완성형 테이블에 없는 글자가 입력될 경우는 접속되는 단말기에 따라 완성형 터미널인 경우는 터미널의 기능에 따르고 N-바이트형 터미널인 경우는 터미널 입력 모드에 따라 Cooked 모드와 Raw 모드로 나누어 처리를 하도록 구현되었다. 지움문자의 처리는 문자 단위의 처리를 기본으로 하고 바이트 단위의 처리를 사용자가 선택할 수 있는 기능을 제공하였다

2) Local conventions 문제 해결: shell 기능을 이용하여 환경변수 LANGUAGE를 한국 환경으로(korean)으로 지정하면 자국어 메시지 출력이 가능하며, 날짜, 시간정보의 출력도 shell이 한국식 규격화일인 /lib/cftime/korean을 참조로 하여 한국식으로 출력한다. 여기서의 shell은 시스템 V 표준 shell인 Bourne-shell이다. 위와 비슷한 서어비스 기능은 국제표준안이 보완되어 갈수록 다양하고 풍부해 질것이다.

3) 비영어권 패턴처리 문제[12] : 패턴처리의 문제는 국제 공통적인 인터페이스로만 해결될수 없는 부분으로 UNIX 국제화 기능에서 제공하지 않고 있다. 종래의 UNIX에 비해 시스템 V, Rel.3.1에서는 대부분의 정규표현 관련 명령어에서 패턴처리 부분만을 공통화일로 독립시킴으로서 나라에 따른 패턴처리 방식이 달라질 경우 공통 화일만 수정하면 해결될 수 있도록 개선했다.

한글 UNIX에서도 정규표현 처리 관련 명령어의 한글 지원은 독립된 정규표현 처리 기능 함수를 한글화하고, 각 명령어와의 인터페이스를 맞추어 구현하였다. shell과 명령어에서의 한글 패턴 처리는 바이트 단위가 아니고 문자 단위의 처리를 한다.

4) 명령어들의 한글 지원[13] : UNIX 시스템 V 표준 명령어에서 한글 지원이 필요한 명령어 갯수는 매우 많은데 다음과 같이 크게 분류를 하여 필요한 수정을 하였다.

- (1) 문서 편집을 위한 명령어
- (2) C-프로그래밍 언어 관련 명령어,
- (3) 패턴 처리 관련 명령어
- (4) 입출력 관련 명령어
- (5) 시스템 정보 처리 관련 명령어
- (6) help 기능을 제공하는 명령어
- (7) 한글 자료 처리 기능 명령어

여기서 패턴관련 명령어와 시스템 정보처리 관련 명령어에 대해서는 앞에서 기술한것 처럼 공통 함수화하여 수정이 이루어졌으며, 나머지는 부분적인 수정을 통해 이루어졌다.

그러나 한글 지원용 C-라이브러리와 Curses-라이브러리를[14] 별도로 구성하고, 이를 사용하여 구현함으로써 명령어, shell, 유틸리티에서의 한글 처리를 공통화하여 지원하도록 하였다.

5) 기타 : troff, nroff와 같은 text formatter에 대한 한글처리 기능을 제공하였고, 한글 완성형 코드와 N-바이트 코드 체계간의 변환 프로그램을 구성하여 기존에 많이 쓰이는 N-바이트체계 입출력 장치를 통한 한글 입출력을 가능하도록 하였다.

그리고 현재 대상 shell인 Bourne-shell 이외에 C-shell, Korn-shell에 대한 한글화를 고려중이다.

III. 결 론

UNIX 국제화의 표준규격과 그물 토대로 한 한글 UNIX의 구현 방향에 대하여 살펴 보았다. 이미 국내에는 한글을 지원하는 UNIX 제품이 기업체에서 개발되어 상용화 되었지만 이들은 국제화 기능을 가진 버전 이전의 것이다. 따라서 구현시의 어려움뿐 아니라 표준화, 규격화, 호환성 등이 제외되었다는 문제점이 있다. 이러한 문제점들을 해결하면서 행정 전산망용 주전산기의 운영체제를 위한 UNIX 시스템 V, Rel.3.1상의 한글화 작업이 ETRI에서 표준 규격에 준하여 이루어지고 있다. 앞으로 UNIX 시스템 V, Rel.4.0에서 제공되는 기능이 추가되고 여러 표준화 활동

그룹에서의 연구가 진행될수록 Internationalization을 위해 제공되는 표준규격 및 기능은 풍부해질 것이다.

또한 ETRI는 UNIX 표준화 활동을 위해 현재는 IEEE POSIX, X/Open, UNIX International Inc., OSF에 가입하여 정보를 수집하는 단계이나, 향후 표준 규격을 수용하는 제품이 UNIX 시장에 중대한 영향을 미칠것에 대비하여 여러 표준 규격에 대한 기술적인 분석을 하고 한국의 환경과 상황을 수렴하여 국제표준 기구에 반영할 수 있는 능동적인 단계로 발전할것이다.

IV. 참고 문헌

- [1] "THE VOICE OF THE INDUSTRY STANDARD UNIX SYSTEM V," UNIX International, 1989
- [2] ISO, "Information processing - ISO 7-bit coded character set for information interchange," ISO, 2nd Ed., 1983
- [3] ISO, "Information processing - ISO 7-bit and 8-bit coded character sets - Code extension techniques," ISO, 3rd Ed., 1986
- [4] ISO, "Information processing - Coded character sets for text communication," ISO, 1st Ed., 1983
- [5] ISO, "Information processing - 8-bit single-byte coded graphic character sets," ISO, 1987
- [6] X/Open, "X/Open Portability Guide," X/Open company, Ltd., 1988
- [7] AT&T, "System V Interface Definition," Issue2, AT&T, 1986
- [8] 김 해진, 안 선희, 유 우종, 유 인석, "한글 UNIX 기초 연구," TD88-1740-02, ETRI, 1988
- [9] 김 해진의 6인, "운영체제에서의 한글 처리 기능 설계 규격서 Rev.2.0," TD89-6140-192, ETRI, 1989
- [10] 한국 표준 연구소, "한글 정보 처리 표준화에 관한 연구," 한국 표준 연구소, 1986
- [11] 유 우종, 김 주만, "장치 관리 볼륨 설계서," TD89-6140-006, ETRI, 1989
- [12] 안 선희, "UNIX 정규 표현 관련 명령어의 한글 지원을 위한 설계," TD89-6140-204, ETRI, 1989
- [13] 김 해진의 3인, "명령어 한글 처리 기능 하부 볼륨 설계서," TD89-6140-292, ETRI, 1989
- [14] 김 진미의 3인, "Curses Library 하부 볼륨 설계서," TD89-6140-207, ETRI, 1989