

EAI(E-Mail Address Internationalization) bis 표준을 지원하는 SMTPUTF8 메일 서버의 설계 및 구현*

한동윤** · 김정석***

A Design and Implementation of SMTPUTF8 Mail Server to
Support EAI(E-Mail Address Internationalization) bis Standard*

Dongyun Han** · Kyongsok Kim***

■ Abstract ■

Recently EAI WG of IETF has published standards for multilingual support in the local part of the e-mail address. Between 2009 and 2010, EAI WG published RFC documents, called the EAI standard, extension to the existing mail-related RFC documents, and EAI has been implemented and tested. In the course of reviewing the results, this yielded a proposal to modify the EAI standard. As a result, EAI WG has published modified RFC documents, called the EAI bis standard. SMTPUTF8 mail server described in this paper has been implemented according to the EAI bis standard. In addition, it has the ability to support two different standards to support the multilingual domain part of the e-mail address, IDNA and EAI bis. Furthermore, SMTPUTF8 mail server can encode in punycode the local part of the e-mail address. Finally, interoperability testing shows that SMTPUTF8 mail server is well-implemented according to the EAI bis standard.

Keyword : E-mail, Address, Internationalization, Mail Server, SMTPUTF8

1. 서 론

인터넷이 보급된 이후 현재까지 가장 많이 사용되고 있는 서비스는 웹 서비스와 전자우편 서비스이다. 지금은 SNS(Social Network Service), 메신저 등 전자우편 서비스를 대체하여 타인에게 메시지를 전달할 수 있는 서비스들이 많이 이용되고 있지만 아직 많은 사람들이 전자우편 서비스를 이용하고 있다. 초창기의 전자우편 서비스는 영문으로 된 간단한 메시지를 주고받을 수 있었다. 지금은 메일 관련 기술들이 발전하여 파일 첨부, 다국어 지원과 같은 각종 기능들이 추가되었다. 전자우편 서비스는 사람들이 필요로 하는 각종 업무 처리에 활용되고 있으며 스마트폰을 통해 전자우편을 주고받을 수 있어 정보 전달의 수단으로 요긴하게 사용되고 있다.

이런 발전에도 불구하고 전자우편의 제목이나 내용 등은 여러 가지 언어를 사용할 수 있지만 전자우편주소에서는 로마자와 숫자, 그리고 일부 기호만 사용하고 있다. 만약 전자우편주소에 한글을 사용할 수 있다면 전자우편을 주고받는 사람들 사이의 식별이 원활해지고 자신을 쉽게 표현할 수 있으며 로마자에 익숙하지 않은 사람도 쉽게 전자우편주소를 만들 수 있다. 이런 상황은 비단 우리나라의 특수한 상황이 아니라 로마자를 쓰지 않는 국가들의 공통된 고민이라 할 수 있다. 특히 아직까지 인터넷 사용률이 낮거나 교육 수준이 낮은 국가들은 더욱 절실하게 느끼는 문제이다[1].

이 문제를 해결하기 위해 IETF(Internet Engineering Task Force)는 EAI WG(E-mail Address Internationalization Working Group)을 만들어 다양한 언어로 된 전자우편주소를 사용할 수 있도록 전자우편 관련 표준들을 확장하는 방안을 연구하여 새로운 RFC(Request For Comments) 문서들을 발표하였다[6].

EAI WG는 2009년에서 2010년에 걸쳐 다국어 전자우편주소를 지원하기 위해, 전자우편주소에 ASCII 문자만을 허용하던 ESMTP(RFC 5321)[9]

을 비롯한 전자우편 관련 표준들을 확장한 새로운 표준들을 발표하였다. 여기서 다국어란 한글과 한자, 로마자를 포함한 다양한 언어를 의미한다.

본 논문에서는 다국어 전자우편주소를 지원하지 않는 예전 표준들을 ESMTP 표준이라고 부르고, ESMTP 표준을 따르는 메일 서버들을 ESMTP 메일 서버라고 부른다.

다국어 전자우편주소를 지원하기 위해 ESMTP 표준을 확장한 표준들을 EAI 표준이라 부르고 이 표준을 따르는 메일 서버를 EAI 메일 서버라고 부른다.

우리나라를 비롯하여 일본, 중국, 대만, 캐나다의 Afiliat 등에서 EAI 표준에 따라 메일 서버를 구현하였고 상호 연동 시험을 통해 전자우편이 EAI 표준에 따라 정상적으로 송수신되는 것을 확인하였다.

2009년에서 2010년 사이에 발표한 EAI 표준은 experimental RFC였다. EAI 표준에 있던 다운그레이드는 EAI 메일 서버를 구현한 경험 및 검토 결과 표준에서 빠자는 제안이 나왔다. 그 결과 EAI WG는 2012년에 수정된 EAI 표준 네 가지를 발표하였고 관련된 다른 표준들도 2013년에 발표되었거나 발표될 예정이다. 이 표준들을 본 논문에서는 EAI bis 표준이라 부르고 이 표준을 따르는 메일 서버를 SMTPUTF8 메일 서버라고 부른다.

앞에서도 언급한 바와 같이 여러 곳에서 EAI 표준을 따르는 메일 서버를 구현했지만 아직까지 EAI bis 표준을 따르는 메일 서버, 즉 SMTPUTF8 메일 서버에 대한 상호 연동 시험을 해 보지는 않았다.

아직 ESMTP 표준을 따르는 메일 서버들이 대부분이기 때문에 다국어 전자우편주소를 사용하는 사람들이 거의 없다. 모든 메일 서버들을 갑자기 EAI bis 표준을 따르는 메일 서버로 바꾸는 것은 불가능하다. 따라서 다국어 전자우편주소를 대중화하기 위한 한시적인 방법으로 ESMTP 메일 서버는 그대로 두고 메일 인터페이스 수준에서 다국어 전자우편주소를 지원하는 방안을 강구할 필요가 있다.

본 논문의 기여는 다음과 같다. 첫째, EAI bis 표준을 따르는 메일 서버인 SMTPUTF8 메일 서버를 설계하고 구현하였다. 둘째, 전자우편주소의 도메인 파트(domain part)에 다국어를 지원하기 위한 방안은 EAI bis 표준에 따라 UTF-8을 이용하거나 IDNA 표준에 따라 유니코드를 이용해야 하는데 본 논문에서 구현한 메일 서버는 두 가지 방식을 모두 구현하였다. 마지막으로 다국어 전자우편주소를 대중화하기 위해 메일 인터페이스 부분에 유니코드를 이용한 방식을 추가하였다. 본 논문에서는 어떻게 SMTPUTF8 메일 서버를 설계하고 구현하였는지 설명하고 다른 메일 서버들과 메일을 주고받은 시험 결과를 보여줄 것이다.

2. 관련 연구

2.1 다국어 전자우편주소 지원 표준

전자우편주소는 '@'을 기준으로 왼쪽 부분인 로컬 파트(local part)와 오른쪽 부분인 도메인 파트로 나누어진다. 이 중 오른쪽 부분인 도메인 파트에 다국어를 사용할 수 있도록 하는 방안으로 IDNA(International Domain Name in Applications)라는 표준이 만들어졌다[10]. 사실 이 표준은 전자우편 서비스만을 위해서 만들어진 표준이 아니라 영문 도메인 이름을 사용하는 웹, FTP를 비롯한 많은 서비스들을 위해 만들어진 것이다.

도메인 파트에 다국어를 사용하기 위한 방안으로 처음에 논의되었던 것은 도메인 파트에 UTF-8[16] 문자를 쓰는 방안으로 이것을 IDN이라 부른다. 이 경우 DNS(Domain Name System) 서버에 다국어 도메인을 그대로 저장하려면 전 세계의 모든 DNS 서버들을 유니코드를 인식하는 새로운 서버로 바꾸어야 하고 이는 현실적으로 거의 불가능하다. 결국 이 IDN 방안은 채택되지 못했고 대신 클라이언트, 서버, DNS 서버 등 네트워크 호스트 사이에서는 ASCII 문자열 형태의 도메인을 주고받지만 애플리케이션 수준에서 ASCII 도메인과 다국어 도메인

사이의 변환을 지원하는 IDNA 표준이 만들어졌다. 전자우편주소의 도메인 파트도 이 표준에 따라 유니코드를 이용해서 다국어를 지원할 수 있다[10].

IDNA의 도입으로 전자우편주소의 도메인 파트는 다국어를 지원할 수 있게 되었으나 로컬 파트는 여전히 로마자와 숫자, 일부 기호만 허용되었다. 전자우편주소의 로컬 파트에 다국어를 지원하기 위한 표준을 제정하기 위한 연구는 2006년부터 IETF의 EAI WG을 중심으로 본격적으로 시작되었다. 이 연구의 핵심 개념은 전자우편주소의 로컬 파트에 UTF-8 문자를 사용하겠다는 것이다. 이 활동의 결과, 2009년과 2010년에 걸쳐서 EAI 전체 프레임워크를 설명하는 RFC 4952를 비롯하여 EAI를 위한 SMTP 확장(RFC 5336), EAI를 위한 POP3 확장(RFC 5721), EAI를 위한 UTF8 헤더 정의(RFC 5335), EAI를 위한 DSN(Delivery status notification) 확장(RFC 5337)과 같은 EAI 표준 문서들이 만들어졌다[2, 4, 7, 11, 14].

EAI 표준은 ESMTP 메일 서버와의 호환을 위해 다운그레이드 기법을 제안하였다. 다운그레이드 기법이란 다국어 전자우편주소에 대응하는 ASCII 전자우편주소를 Alt-address 필드에 두고, EAI 메일 서버에서 ESMTP 메일 서버로 메일이 전달될 때 다국어 전자우편주소를 ASCII 전자우편주소로 대체하여 보내는 방법이다.

EAI 표준에 따라 한국, 일본, 중국, 캐나다 등에서 EAI 메일 서버들이 만들어졌고 상호 연동 시험이 진행되었다. 시험 및 검토 결과, EAI WG는 다운그레이드 기법을 빠자는 방향으로 논의를 진행하였다. 이 논의에 따라 2011년부터 다운그레이드 기법을 제거한 새로운 표준인 EAI bis 표준이 발표되었다.

RFC 6530(EAI 전체 프레임워크, RFC 4952를 수정), RFC 6531(EAI를 위한 SMTP 확장, RFC 5336을 수정), RFC 6532(EAI를 위한 UTF8 헤더 정의, RFC 5335를 수정), RFC 6533(RFC를 위한 DSN 확장, RFC 5337을 수정)와 같은 문서들이 다국어 전자우편주소를 지원하기 위한 EAI bis 표

준 문서로 만들어져 있다[5, 8, 13, 15].

2.2 ESMTP(Extended Simple Mail Transfer Protocol)

전자우편을 주고받기 위한 인터넷 표준 프로토콜인 SMTP는 RFC 821[12]에 정의되어 있으나 전자우편과 관련된 요구들이 많아지고 관련 기술들이 발전하면서 계속해서 확장되었다. 다국어 전자우편주소를 지원하지 않는 가장 최근의 ESMTP 표준 문서는 RFC 5321[9]이다. 이 RFC 두 개를 바탕으로 SMTP와 ESMTP를 대비하여 분석해 보면 다음과 같은 규칙들을 정의하고 있다.

- SMTP에서 규정하는 규칙들[12]
 1. SMTP는 단순 텍스트 메시지 포맷을 명령 기반으로 하는 프로토콜이다.
 2. SMTP는 클라이언트가 서버에 명령을 하면 서버는 클라이언트가 보낸 명령을 처리한 후 그 명령에 대한 응답을 보내는 동기식 통신 프로토콜이다.
 3. SMTP 메시지는 ASCII 라인피드 문자(LF) 앞에 나오는 ASCII 캐리지 리턴 문자(CR)로 행을 분리한다.
 4. 행 분리자는 플랫폼에 독립적이고 메시지가 처리되는 모든 운영 체제에 적용이 가능하다.
 5. 사용자 이름과 도메인은 최대 64문자까지 사용이 가능하다.
 6. 명령어 라인은 CRLF를 포함하여 512문자로 제한한다.
 7. 응답 라인 역시 명령어 라인과 같이 CRLF를 포함하여 512문자로 제한한다.
 8. 텍스트 라인은 CRLF를 포함하여 1000문자 미만으로 제한한다.
- ESMTP로 확장되면서 추가된 기능들[9]
 1. 8BITMIME(8bit data transmission) 지원 : 텍스트, 그림, 첨부 파일 등 다양한 형태의 메시

지들을 전달하기 위한 기술[3]

2. ATRN(Authenticated Turn for On-Demand Mail Relay) : 하나 이상의 도메인을 매개 변수로 사용할 수 있게 하는 기술
3. SMTP-AUTH(Authenticated SMTP) : SMTP 서버에 대한 인증 기능
4. CHUNKING(Chunking) : SMTP 호스트가 데이터의 끝부분을 계속적으로 검사하지 않도록 메시지 전체 바이트 수가 포함된 인수와 함께 보내고 수신 서버에서는 메시지 바이트를 계산하여 받은 값과 같은 경우 모든 메시지 데이터를 받았다고 가정하는 기술
5. DSN : 전자우편 수신 확인 기능
6. ETRN(Extended Turn) : 다른 메일 서버에서 전자우편을 모두 보내도록 요청하는 기능
7. HELP : 도움말
8. PIPELINING : 명령 후 응답을 기다리지 않고 다음 명령을 전송하는 기능
9. SIZE : 메시지 크기 확인
10. STARTTLS(Transport layer security) : 네트워크 전송 계층에 보안 레이어를 추가했을 때 이를 지원하는 기능

3. SMTPUTF8 메일 서버 설계

제 3장에서는 EAI bis 표준을 따르는 메일 서버인 SMTPUTF8 메일 서버를 어떻게 설계하였는지 설명할 것이다. 제 3.1절에서 요구 사항을 분석하여 설명할 것이다. 제 3.2절에서 분석된 요구 사항을 바탕으로 설계한 내용을 설명할 것이다. 마지막으로 제 3.3절에서 다국어 전자우편주소를 대중화시키기 위한 한시적인 방법인 유니코드를 메일 인터페이스에 적용했을 때 SMTPUTF8 메일 서버가 어떻게 동작하는지 설명한다.

3.1 설계를 위한 요구 사항 분석

SMTPUTF8 메일 서버를 설계하기 위해 EAI

bis 표준 문서 중 하나인 RFC 6531[13]이 ESMTP 표준 문서 중 하나인 RFC 5321[9]과 어떤 점에서 다른지 확인해야 한다.

SMTPUTF8 메일 서버는 RFC 5321[9] 문서에서 Mailbox가 나올 수 있다고 규정한 모든 자리에서 UTF-8 문자를 받아들여야 한다. 예를 들어 Mail from이나 Rcpt to 명령에 대한 값인 <reverse-path>나 <forward-path>와 같은 부분에 Mailbox가 나올 수 있다고 규정되어 있는데 이 부분에서 UTF-8 문자를 쓸 수 있어야 한다.

여기서 Mailbox란 원래 받은 메일을 저장하는 공간을 의미하지만 RFC 5321[9]는 Mailbox의 이름이 “local-part@domain”으로 이루어진다고 정의한다. 즉, Mailbox 이름이 전자우편주소와 같은 형태로 되어 있기 때문에 Mailbox 이름이 들어올 수 있는 자리에 전자우편주소가 들어올 수 있다[9].

EAI bis 표준을 따르는 메일 클라이언트는 ESMTP 표준을 따르는 메일 클라이언트와 마찬가지로 메일 서버에게 “EHLO” 명령을 보낼 수 있다. 이 때 이에 대한 응답으로 SMTPUTF8 확장 키워드를 받으면 이는 메일 서버가 EAI bis 표준을 따르는 메일 서버라는 뜻이다. 이 응답을 받은 메일 클라이언트는 다음과 같이 할 수 있다.

- Mail from이나 Rcpt to와 같은 SMTP 명령에서 Mailbox 이름을 UTF-8 형태의 문자열로 서버에게 보낼 수 있다.

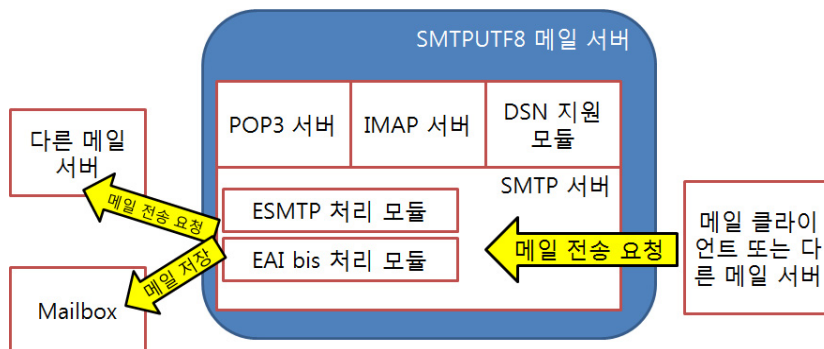
- RFC 6532[15]에서 규정된 UTF-8 형태의 헤더들을 보낼 수 있다(헤더에는 UTF-8 형태의 Mailbox 이름이 있을 수 있다).
- SMTP 명령이나 메시지 헤더 안에서 전자우편 주소의 도메인 부분은 IDNA 표준(RFC 5891[10])에 따라 유니코드로 보낼 수도 있고 EAI bis 표준(RFC 6531[13])에 따라 UTF-8로 보낼 수도 있다. 이 내용은 어떤 RFC 문서 하나에 명시적으로 나오지는 않는다. 본 논문에서는 EAI bis 표준 문서와 IDNA 표준 문서를 모두 분석하여 이런 결론을 내렸다.

만약 “EHLO”에 대한 응답에 250 SMTPUTF8 확장 키워드가 없다면 응답한 메일 서버가 ESMTP 표준을 따르는 메일 서버라는 의미이다. 이 때 메일 클라이언트는 Mailbox 이름으로 ASCII 문자만 사용할 수 있다.

3.2 SMTPUTF8 메일 서버 설계

SMTPUTF8 메일 서버는 ESMTP 메일 서버에 EAI bis 기능들을 추가하는 형식으로 설계한다. 이는 이미 존재하는 많은 메일 서버들을 쉽게 바꾸기 위한 것이다.

[그림 1]은 SMTPUTF8 메일 서버의 전체 구조를 표현한 것이다. 그림에서 보는 바와 같이 사용자들은 메일 클라이언트(또는 웹 사이트)에서 제



[그림 1] SMTPUTF8 메일 서버의 내부 구조

공하는 메일 인터페이스에 따라 메일을 만든 다음, 메일 서버로 전송을 요청한다. 또는 다른 메일 서버에서 SMTPUTF8 메일 서버에 메일 전송을 요청하는 경우도 있다.

이 때 메일 전송을 요청받은 SMTPUTF8 메일 서버가 수신자의 메일 서버라면 ESMTP 표준 또는 EAI bis 표준에 따라 Mailbox에 메일을 저장한다.

만약 메일 전송을 요청받은 SMTPUTF8 메일 서버가 수신자의 메일 서버가 아니라면 SMTPUTF8 메일 서버는 메일 클라이언트가 되어 수신자의 메일 서버에 요청받은 메일을 전달한다.

이 때 송신자 또는 수신자 메일 주소의 로컬 파트가 다국어인 경우에는 [그림 1]의 EAI bis 처리 모듈에 의해 메일 주소가 UTF-8로 인코딩된 후, EAI bis 표준에 따라 수신자의 메일 서버로 메일을 전달한다. 만약 송신자 또는 수신자 메일 주소의 로컬 파트가 모두 다국어가 아니라면 [그림 1]의 ESMTP 처리 모듈에 의해 메일 주소가 ASCII로 인코딩된 후, ESMTP 표준에 따라 수신자의 메일 서버로 메일을 전달한다.

수신자의 메일 서버가 ESMTP 메일 서버라면 EAI bis 표준을 따를 수 없으므로 ESMTP 표준에 따른 메일 전송만 가능하다. 수신자의 메일 서버가 SMTPUTF8 메일 서버라면 EAI bis 표준에 따른 메일 전달과 ESMTP 표준에 따른 메일 전달이 둘 다 가능하다.

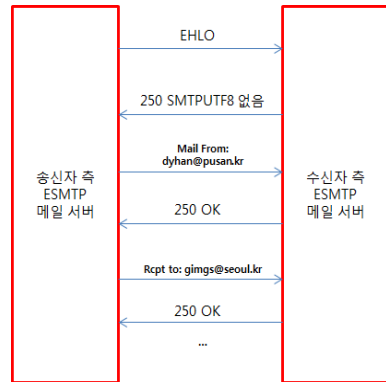
따라서 송신자의 메일 서버가 SMTPUTF8 메일 서버이고 수신자의 메일 서버가 ESMTP 메일 서버이며 EAI bis 표준에 따른 메일 전달이 필요한 경우에만 메일이 전송되지 않는다.

수신자의 메일 서버는 전송받은 메일을 수신자의 Mailbox에 저장하며 수신자는 이 메일을 웹 사이트나 POP3 클라이언트를 통해 메일을 확인한다. 수신자의 메일 서버가 SMTPUTF8 메일 서버라면 메일 클라이언트에서 UTF-8 인코딩된 전자우편주소의 로컬 파트를 디코딩하여 다국어로 보여주고 UTF-8 또는 유니코드로 인코딩된 도메인

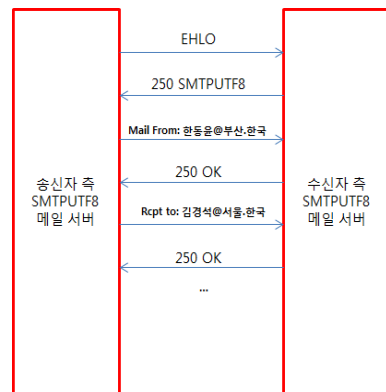
파트 역시 디코딩하여 다국어로 보여준다.

위에서 설명한 메일 전송 구조를 통해 다국어 전자우편주소를 지원하기 위해 필요한 사항들을 정리하면 다음과 같다.

1. 메일 인터페이스에서 다국어 전자우편주소를 지원해야 한다.
2. 작성된 메일을 메일 서버로 보낼 때 전자우편 주소의 로컬 파트는 UTF-8로 인코딩되어야 하고 도메인 파트는 UTF-8 또는 유니코드로 인코딩되어야 한다.



[그림 2] ESMTP 메일 서버 간의 메일 송수신 과정

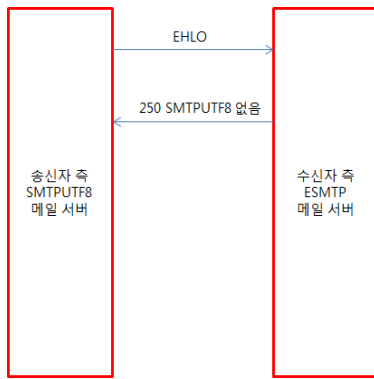


[그림 3] SMTPUTF8 메일 서버 간의 메일 송수신 과정

[그림 2]는 ESMTP 메일 서버들 간의 메일 송수신 과정을 나타낸 것이고 [그림 3]은 SMTPUTF8 메일 서버들 간의 메일 송수신 과정을 나타낸 것

이다.

[그림 2]와 [그림 3]에서 보듯이 SMTPUTF8 메일 서버들 사이의 메일 송수신 과정은 ESMTP 메일 서버들 사이에서의 메일 송수신 과정과 크게 다르지 않다. 다만 전자우편주소에 다국어어를 사용할 수 있게 바뀌고, “EHLO”에 대한 응답에 SMTPUTF8 확장 키워드가 추가될 뿐이다.



[그림 4] SMTPUTF8 메일 서버에서 ESMTP 메일 서버로 메일을 보내는 과정

[그림 4]는 SMTPUTF8 메일 서버에서 메일을 보내고 ESMTP 메일 서버에서 메일을 받을 때 일어나는 과정을 표현한 것이다. ESMTP 메일 서버는 EAI bis 표준에 따르면 다국어 전자우편주소를 처리할 수 없다. 따라서 송신자의 메일 서버가 수신자의 메일 서버에서 “EHLO”를 보냈을 때 응답으로 SMTPUTF8 확장 키워드를 받지 못하면 다국어 전자우편주소를 사용할 수 없다는 의미이므로 메일 송수신 과정을 종료한다.

송신자의 메일 서버가 ESMTP 메일 서버이고 수신자의 메일 서버가 SMTPUTF8 메일 서버인 경우에는 다국어 전자우편주소를 사용하지 않는다. 송신자의 메일 서버가 ESMTP 메일 서버이기 때문에 메일 인터페이스에서 다국어 전자우편주소를 지원하지 않고 ASCII 전자우편주소만 지원한다. 이 경우 수신자의 메일 서버는 SMTPUTF8 메일 서버이지만 ESMTP 처리 모듈을 통해 전송 받은 메일을 처리한다.

3.3 다국어 전자우편주소 대중화를 위한 SMTPUTF8 메일 서버 확장

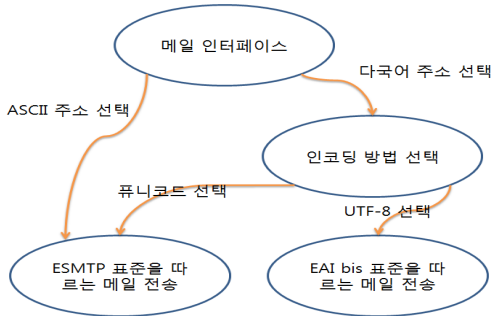
제 3.2절에서 설계한 SMTPUTF8 메일 서버는 EAI bis 표준에 따라 설계한 것이다. 하지만 현존하는 대부분의 메일 서버들이 ESMTP 표준을 따르고 있고 이 서버들을 모두 EAI bis 표준을 따르도록 바꾸는 것은 현실적으로 어렵다. 따라서 서서히 EAI bis 표준을 따르는 메일 서버들을 늘려가야 한다. 이를 위해 일단 다국어 전자우편주소를 대중화시키는 것이 중요하다. 따라서 IDNA[10]에서 도메인을 유니코드로 바꾼 것처럼 전자우편주소의 로컬 파트를 유니코드로 바꾸면 ESMTP 메일 서버는 전혀 바꾸지 않고 그대로 둔 채 메일 인터페이스만 수정하여 다국어 전자우편주소를 사용할 수 있다. 다만 이것은 EAI bis 표준을 따르는 메일 서버들이 대중화되기 전에 한시적으로 사용할 수 있는 방법이다. 이것을 정리하면 다음과 같다.

- EAI bis 표준에 따른 다국어 전자우편주소
: UTF-8@UTF-8 또는 UTF-8@punycode
- 한시적으로 사용할 다국어 전자우편주소
: punycode@punycode

본 논문에서 구현한 SMTPUTF8 메일 서버는 메일 인터페이스에서 어떤 인코딩 방법을 선택하느냐에 따라 다국어 전자우편주소의 로컬 파트를 UTF-8 인코딩으로 지원할 수도 있고 유니코드 인코딩으로 지원할 수도 있다.

[그림 5]는 메일 인터페이스에서 어떤 언어로 된 주소를 선택하고 어떤 인코딩 방법을 선택하느냐에 따라 달라지는 내부 흐름을 나타낸 것이다. 메일 인터페이스에서 ASCII 로컬 파트를 사용하는 경우에는 ESMTP 표준에 따라 메일을 전송한다. 다국어 로컬 파트를 사용하더라도 유니코드 인코딩 방법을 선택하면 ASCII 형태로 인코딩되기 때문에 ESMTP 표준에 따라 메일을 전송한다. 다국어

로컬 파트를 선택하고 UTF-8 모드를 선택하면 EAI bis 표준에 따라 메일을 전송한다.



[그림 5] 메일 인터페이스에서 주소와 인코딩 방법 선택에 따른 내부 흐름

4. SMTPUTF8 메일 서버 구현 및 시험

본 논문에서 구현한 SMTPUTF8 메일 서버의 구현 환경은 다음과 같다.

- 설치 OS : CentOS
- 개발 언어 : Python
- 메일 인터페이스 : 웹 사이트

이 메일 서버 프로그램은 Sourceforge.net을 이용하여 오픈 소스로 개발하였다. 따라서 이 프로그램 소스를 다운로드하기 위해서는 <http://sourceforge.net/projects/pnu-eai/>에 접속하면 된다.

메일 송수신 시험을 위하여 본 논문에서 구현한 SMTPUTF8 메일 서버를 두 대의 컴퓨터에 설치하였고 각각의 도메인 이름은 eai0.pusan.ac.kr과 eai1.pusan.ac.kr이다. 이 서버에서 한글 계정을 만들고 사용하려면 웹을 통해 접속하면 된다.

<표 1>은 SMTPUTF8 메일 서버인 EAI0, EAI1 서버 간의 메일 송수신 시험과 ESMTP 메일 서버인 다음 메일과의 메일 송수신 시험 결과를 나타낸 것이다. <표 1>에서 송신자의 로컬 파트와 수신자의 로컬 파트의 의미는 다음과 같다.

<표 1> 메일 송수신 시험 결과

송신자의 로컬 파트 \ 수신자의 로컬 파트	다음 (ASCII)	EAI1 (ASCII)	EAI1 (다국어 유니코드)	EAI1 (다국어 UTF-8)
다음 (ASCII)	○	○	○	×
EAI0 (ASCII)	○	○	○	○
EAI0 (다국어 유니코드)	○	○	○	○
EAI0 (다국어 UTF8)	×	○	○	○

- 송신자의 로컬 파트가 다음(ASCII) : 다음 사이트에서 ASCII 로컬 파트(ID)로 로그인하여 메일을 보냈다는 의미(다음 메일 서버는 ESMTP 메일 서버이므로 ASCII 로컬 파트만 제공)
- 송신자의 로컬 파트가 EAI0(ASCII) : EAI0 사이트에서 ASCII 로컬 파트(ID)로 로그인하여 메일을 보냈다는 의미
- 송신자의 로컬 파트가 EAI0(다국어 유니코드) : EAI0 사이트에서 다국어 로컬 파트(ID)로 로그인하여 메일을 보냈는데 송신자 전자우편주소의 로컬 파트에 대해 유니코드 인코딩을 선택했다는 의미
- 송신자의 로컬 파트가 EAI0(다국어 UTF8) : EAI0 사이트에서 다국어 로컬 파트(ID)로 로그인하여 메일을 보냈는데 송신자 전자우편주소의 로컬 파트 주소에 대해 UTF-8 인코딩을 선택했다는 의미
- 수신자의 로컬 파트가 다음(ASCII) : 수신자의 메일 서버가 다음 메일 서버이며 따라서 수신자 전자우편주소의 로컬 파트는 ASCII 문자들이라는 의미
- 수신자의 로컬 파트가 EAI1(ASCII) : 수신자의 메일 서버는 EAI1이고 수신자 전자우편주소의 로컬 파트는 ASCII 문자들이라는 의미
- 수신자의 로컬 파트가 EAI1(다국어 유니코드) : 수신자의 메일 서버가 EAI1이며 수신자 전자우편주소의 로컬 파트는 다국어이고 이를 유니

코드 형태로 변환하여 적거나(다음에서 보낼 때) 한글로 적고 유니코드 인코딩을 선택(EAI0에서 보낼 때)했다는 의미

- 수신자의 로컬 파트가 EAI1(다국어 UTF-8) : 수신자의 메일 서버가 EAI1이며 수신자 전자우편주소의 로컬 파트는 다국어이고 이를 그대로 적거나(다음에서 보낼 때) 그대로 적고 UTF-8 인코딩을 선택(EAI0에서 보낼 때)했다는 의미

실제 시험에서는 한글 로컬 파트를 써서 다국어 UTF 또는 다국어 유니코드 로컬 파트를 시험하였다. ASCII 로컬 파트는 로마자와 숫자, 그리고 몇 가지 기호로 이루어진 로컬 파트를 의미한다. 시험에 사용된 메일 주소는 다음과 같다.

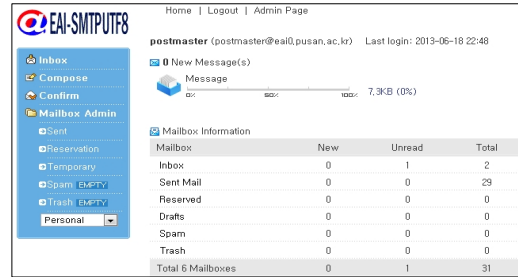
- 다음(ASCII) : zeno_requiem@hanmail.net
- EAI0(ASCII) : postmaster@eai0.pusan.ac.kr
- EAI1(ASCII) : postmaster@eai1.pusan.ac.kr
- EAI0(다국어 유니코드) : 관리자@eai0.pusan.ac.kr, 유니코드로 표현하면 xn-zb0b93v2yk@eai0.pusan.ac.kr
- EAI1(다국어 유니코드) : 관리자@eai1.pusan.ac.kr, 유니코드로 표현하면 xn-zb0b93v2yk@eai1.pusan.ac.kr
- EAI0(다국어 UTF8) : 관리자@eai0.pusan.ac.kr
- EAI1(다국어 UTF8) : 관리자@eai1.pusan.ac.kr



[그림 6] eai0.pusan.ac.kr 로그인 전

[그림 6]과 [그림 7]은 SMTPUTF8 메일 서버인 EAI0에서 메일을 보내고 받은 메일을 확인하기 위

해 로그인하는 과정을 보여준다. 이 그림에서는 한글 ID인 “관리자”로 로그인하는 과정만 보여주지만 영문 ID인 “postmaster”로 로그인할 수도 있다.



[그림 7] eai0.pusan.ac.kr 로그인 후

표에 나오는 16가지 경우를 다음과 같이 4가지 묶음으로 정리할 수 있다.

1. 아래 9가지 경우는 송신자 전자우편주소의 로컬 파트와 수신자 전자우편주소의 로컬 파트가 모두 ASCII 문자열인 경우이다. 9가지 경우 모두 메일은 잘 전송된다.
 - 1-1 다음(ASCII) → 다음(ASCII)
 - 1-2 다음(ASCII) → EAI1(ASCII)
 - 1-3 다음(ASCII) → EAI1(다국어 유니코드)
 - 1-4 EAI0(ASCII) → 다음(ASCII)
 - 1-5 EAI0(ASCII) → EAI1(ASCII)
 - 1-6 EAI0(ASCII) → EAI1(다국어 유니코드)
 - 1-7 EAI0(다국어 유니코드) → 다음(ASCII)
 - 1-8 EAI0(다국어 유니코드) → EAI1(ASCII)
 - 1-9 EAI0(다국어 유니코드) → EAI1(다국어 유니코드)

[그림 8]은 시험 1-3을 위해 다음에 로그인하여 EAI1로 메일을 전송하는 화면이다. 송신자의 메일 주소는 ASCII(영문) 주소이고 수신자의 메일 주소는 유니코드로 인코딩된 다국어(한글) 주소이다. [그림 9]에서 보듯이 메일은 잘 전송되지만 다음의 메일 인터페이스에서 유니코드 인코딩을 지원하지 않기 때문에 [그림 8]에서 보는 바와 같이 수신자

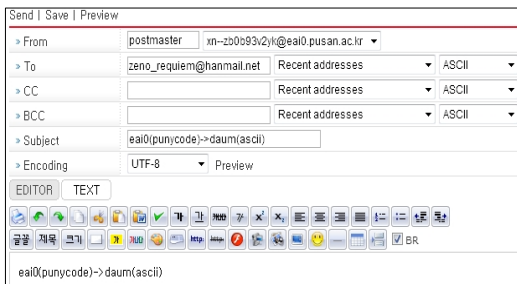
의 메일 주소를 적을 때 유니코드 형태로 적어야 한다.



[그림 8] 1-3 다음(ASCII) → EAI1(다국어 유니코드) : 메일 보내기



[그림 9] 1-3 다음(ASCII) → EAI1(다국어 유니코드) : 메일 확인



[그림 10] 1-7 EAI0(다국어 유니코드) → 다음(ASCII) : 메일 보내기

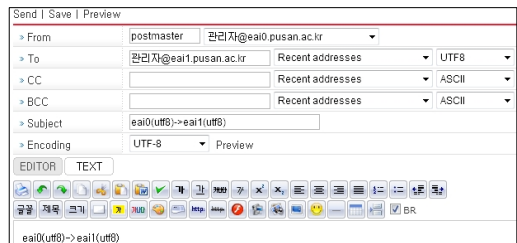


[그림 11] 1-7 EAI0(다국어 유니코드) → 다음(ASCII) : 메일 확인

[그림 10]은 시험 1-7을 위해 EAI0에 로그인하여 다음으로 메일을 전송하는 화면이다. 송신자의 메일 주소는 유니코드로 인코딩된 다국어(한글) 주소이고 수신자의 메일 주소는 ASCII(영문) 주소이다. [그림 11]에서 보듯이 메일은 잘 전송되지만 다음의 메일 인터페이스에서 유니코드를 다국어로 디코딩하지 않기 때문에 송신자의 메일 주소는 유니코드 형태로 보인다.

2. 아래 5가지 경우는 송신자 전자우편주소의 로컬 파트 혹은 수신자 전자우편주소의 로컬 파트 가운데 적어도 한 쪽에 UTF-8이 사용되었고 양쪽 메일 서버가 SMTPUTF8 메일 서버인 경우이다. 다섯 가지 경우 모두 메일은 잘 전송된다.

- 2-1 EAI0(다국어 UTF-8) → EAI1(ASCII)
- 2-2 EAI0(다국어 UTF-8) → EAI1(다국어 유니코드)
- 2-3 EAI0(다국어 UTF-8) → EAI1(다국어 UTF8)
- 2-4 EAI0(ASCII) → EAI1(다국어 UTF8)
- 2-5 EAI0(다국어 유니코드) → EAI1(다국어 UTF8)



[그림 12] 2-3 EAI0(다국어 UTF8) → EAI1(다국어 UTF8) : 메일 보내기

[그림 12]는 EAI0에 로그인하여 EAI1으로 메일을 전송하는 화면이다. 이 때 송신자와 수신자의 메일 주소는 모두 다국어 로컬 파트를 가지고 있으며 둘 다 UTF8 인코딩을 선택하였다. [그림 13]은 이 메일을 확인하는 화면으로 EAI bis 표준에

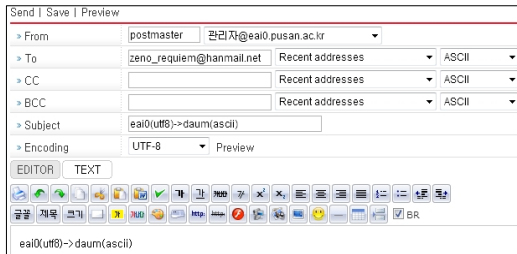
따라 정상적으로 메일을 수신하였음을 보여준다.

Subject	eai0(utf8) -> eai1(utf8)
From	"postmaster" <관리자@eai0.pusan.ac.kr >
To	관리자@eai1.pusan.ac.kr
CC	
Date	2013-06-18 13:37
eai0(utf8) -> eai1(utf8)	
Reply Forward Reply All	

[그림 13] 2-3 EAI0(다국어 UTF8) → EAI1(다국어 UTF8) : 메일 확인

3. 아래 1가지 경우는 주소에 UTF8이 사용되고 SMTPUTF8 메일 서버에서 ESMTP 메일 서버로 메일을 전송하는 경우이므로 메일 전송이 되지 않는다.

3-1 EAI0(다국어 UTF-8) → 다음(ASCII)

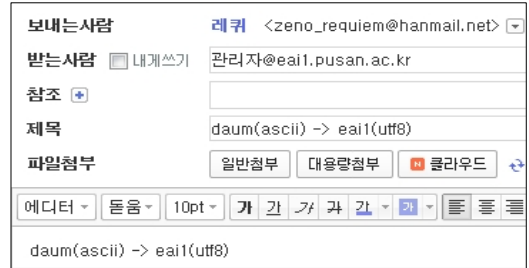


[그림 14] 3-1 EAI0(다국어 UTF8) → 다음(ASCII) : 메일 보내기

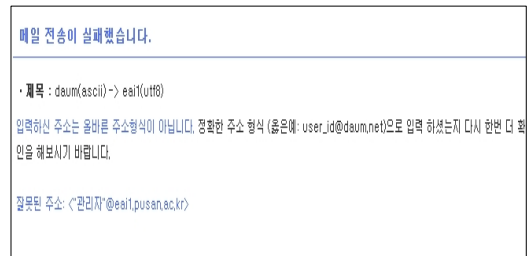
[그림 14]는 EAI0에서 한글 ID로 로그인하여 다음으로 메일을 보내는 화면이다. 송신자의 메일 주소는 다국어(한글) 주소이며 이를 UTF8로 인코딩하도록 선택하였다. 이 때 EAI0에서는 메일이 전송되었다고 나타나지만 다음에서 확인해 보면 메일은 수신되지 않는다.

4. 아래 1가지 경우는 다음의 메일 인터페이스에서 UTF-8 인코딩을 지원하지 않기 때문에 메일 전송이 되지 않는다.

4-1 다음(ASCII) → EAI1(다국어 UTF-8)



[그림 15] 4-1 다음(ASCII) → EAI1(다국어 UTF8) : 메일 보내기



[그림 16] 4-1 다음(ASCII) → EAI1(다국어 UTF8) : 메일 전송 결과

[그림 15]는 다음에서 수신자의 메일 주소를 다국어(한글)로 입력하여 메일을 보내는 화면이다. [그림 16]에서 보듯이 메일이 전송되지 않는다.

1-1에서 1-9의 시험은 송신자 메일 주소와 수신자 메일 주소의 로컬 파트가 모두 ASCII 문자열인 경우다. EAI bis 표준에 따르면 이런 경우에 SMTPUTF8 메일 서버는 ESMTP 메일 전송 규칙에 따라 메일을 송수신해야 하므로 SMTPUTF8 메일 서버 간에도 메일 송수신이 잘 되어야 하고 ESMTP 메일 서버인 다음 메일 서버와도 메일 송수신이 잘 되어야 한다. 시험 결과, 9가지 경우 모두 메일 송수신이 잘 되었다.

2-1에서 2-5의 시험은 송신자 메일 주소와 수신자 메일 주소의 로컬 파트 중 적어도 하나는 UTF8 문자열인 경우다. EAI bis 표준에 따르면 이런 경우에 SMTPUTF8 메일 서버는 EAI bis 메일 전송 규칙에 따라 메일을 송수신해야 하므로 SMTPUTF8 메일 서버 간에는 이 경우에도 메일 송수

신이 잘 되어야 한다. 시험 결과, 5가지 경우 모두 메일 송수신이 잘 되었다.

3-1과 4-1의 시험은 송신자 메일 주소와 수신자 메일 주소의 로컬 파트 중 적어도 하나는 UTF8 문자열이고 송신자의 메일 서버와 수신자의 메일 서버 중 적어도 하나는 ESMTP 메일 서버인 경우이다. EAI bis 표준에 따르면 이런 경우에 메일을 송수신할 수 없다.

따라서 1-1에서 4-1를 종합한 시험 결과는 본 논문에서 구현한 SMTPUTF8 메일 서버가 EAI bis 표준을 잘 따르고 있음을 보여준다.

단 기존의 메일 서버들이 대부분 ESMTP 메일 서버이므로 SMTPUTF8 메일 서버와 ESMTP 메일 서버의 연동 문제를 고려할 필요가 있다. 3-1과 4-1의 결과에서 알 수 있듯이 메일 주소의 로컬 파트가 다국어일 때 UTF8 인코딩을 하게 되면 SMTPUTF8 메일 서버와 ESMTP 메일 서버는 서로 메일을 주고받을 수 없다.

이 문제의 근본적인 해결책은 기존의 ESMTP 메일 서버들을 모두 SMTPUTF8 메일 서버로 바꾸는 것이다. 하지만 이 방법은 많은 시일이 요구되기 때문에 다국어 메일 주소를 사용하기 위한 한시적인 해결책으로 본 논문에서는 유니코드를 활용하는 방안을 제안하였다. 이 방법은 1-3, 1-7의 결과에서 나타나듯이 지금 바로 사용할 수 있다. 하지만 ESMTP 메일 서버의 메일 인터페이스에서 메일을 쓸 때는 다국어 주소의 유니코드를 직접 입력해야 하고, 메일을 볼 때는 다국어로 나타나지 않기 때문에 사용자 입장에서는 매우 불편하다. 따라서 유니코드를 활용하는 방안을 대중화시키기 위해서는 ESMTP 메일 서버의 메일 인터페이스에서 유니코드 인코딩과 디코딩을 지원해야 한다.

참고로 본 논문에서 개발한 SMTPUTF8 메일 서버는 EAI bis 표준에 따라 제대로 동작하는지 확인하기 위한 프로토타입 개발이므로 메일 서버 간의 전송(처리) 속도 또는 메일 서버의 H/W 사용량 등은 본 논문의 연구 범위에 포함시키지 않

았다. 다만 구현 경험으로 볼 때 ESMTP 메일 서버에 견주어 SMTPUTF8 메일 서버의 처리 속도가 크게 떨어지거나 H/W 사용량이 크게 늘어나지는 않을 것이라고 확신한다.

5. 결론 및 향후 과제

본 논문에서는 EAI bis 표준을 따르는 메일 서버인 SMTPUTF8 메일 서버를 설계하고 구현하였다. 그리고 전자우편주소의 도메인 파트(domain part)에 다국어를 지원하기 위한 방안은 EAI bis 표준에 따라 UTF-8을 이용하거나 IDNA 표준에 따라 유니코드를 이용해야 하는데 본 논문에서 구현한 메일 서버에서는 두 가지 방식을 모두 구현하였다. 마지막으로 다국어 전자우편주소를 대중화하기 위해 메일 인터페이스 부분에 유니코드를 이용한 방식을 추가하였다.

또 제 4장에서 구현한 메일 서버가 EAI bis 표준을 따르고 다국어 전자우편주소를 대중화시키기 위한 유니코드를 이용한 방식도 잘 동작한다는 것을 확인하였다.

향후 연구 과제로는 먼저 메일 서버를 구성하는 다른 부분들, 즉 POP3 서버나 IMAP 서버, DSN 서버 등도 EAI bis 표준에 따라 구현하는 것을 생각해 볼 수 있다. EAI bis 표준들은 이미 발표되었거나 발표되기 직전이기에 때문에 이 구현은 빠른 시일 안에 가능할 것이다.

두 번째로 이 서버들을 이용하여 사람들이 다국어 전자우편주소에 친숙해질 수 있도록 각종 메일 애플리케이션을 개발하는 작업도 필요하다. 특히 스마트폰 앱을 통해 다국어 전자우편주소를 사용하여 메일을 주고받게 되면 다국어 전자우편주소의 대중화에 큰 보탬이 될 것이다.

참 고 문 헌

- [1] 김경석, 「다국어 전자메일주소 국제표준화 활동 및 연구」, 한국인터넷진흥원, 서울, 2008.

- [2] Abel, Y., "RFC 5335 : Internationalized E-mail Headers", IETF, USA, 2008.
- [3] Freed, N. and N. Borenstein, "RFC 2047 : Multipurpose Internet Mail Extensions (MIME) Part One : Format of Internet Message Bodies", IETF, USA, 1996.
- [4] Gellens, R. and C. Newman, "RFC 5721 : POP3 Support for UTF-8", IETF, USA, 2010.
- [5] Hansen, T., C. Newman, and A. Melnikov, "RFC 6533 : Internationalized Delivery Status and Disposition Notifications", IETF, USA, 2012.
- [6] IETF EAI WG, "<http://tools.ietf.org/wg/eai/>."
- [7] Klensin, J. and Y. Ko, "RFC 4952 : Overview and Framework for Internationalized Email", IETF, USA, 2007.
- [8] Klensin, J. and Y. Ko, "RFC 6530 : Overview and Framework for Internationalized Email", IETF, USA, 2012.
- [9] Klensin, J., "RFC 5321 : Simple Mail Transfer Protocol", IETF, USA, 2008.
- [10] Klensin, J., "RFC 5891 : Internationalized Domain Names in Applications(IDNA) : Protocol", IETF, USA, 2010.
- [11] Newman, C. and A. Melnikov, "RFC 5337 : Internationalized Delivery Status and Disposition Notifications", IETF, USA, 2008.
- [12] Postel, J. B., "RFC 821 : Simple Mail Transfer Protocol", ISI, USA, 1982.
- [13] Yang, A., S. Steele, and N. Freed, "RFC 6532 : Internationalized Email Headers", IETF, USA, 2012.
- [14] Yao, J. and W. Mao, "RFC 5336 : SMTP Extension for Internationalized Email Addresses", IETF, USA, 2008.
- [15] Yao, J. and W. Mao, "RFC 6531 : SMTP Extension for Internationalized Email Addresses", IETF, USA, 2012.
- [16] Yergeau, F., "RFC 3629 : UTF-8, A translation format of ISO 10646", IETF, USA, 2008.

◆ 저 자 소 개 ◆

**한 동 윤 (dyhan1@pusan.ac.kr)**

부산대학교 정보컴퓨터공학부를 졸업하고 동대학 대학원 컴퓨터공학과 석사과정 후, 현재는 동대학원에서 박사과정 중이다. 관심분야는 메일 시스템, 플래시 메모리, 데이터베이스이다.

**김 경 석 (gimgs@pnu.kr)**

일리노이 주립대(어바나-샴페인) 전자계산학 박사과정 취득 후 미국 노스다코타 주립대학교 전자계산학과 조교수로 재직하였으며, 현재는 부산대학교 정보컴퓨터공학부 교수로 재직 중이다. 관심분야는 데이터베이스, 한글/한말 정보처리, 인터넷 컴퓨팅 등이다.