

## 가상도메인을 이용한 웹메일 서비스 시스템의 설계 및 구현

류언무\*, 황현미\*\*, 이상철\*\*\*

### Design and Implementation of Web-Mail Service System using a Virtual Domain

Yeon-Moo Ryu\*, Hyun-Mi Hwang\*\*, Sang-Chul, Lee\*\*\*

#### 요 약

인터넷의 다양한 서비스의 등장으로 사용자의 수는 기하급수적으로 늘어나고 있으며, 이러한 경향으로 좀더 효율적으로 이용할 수 있는 시스템 및 서비스가 날로 증가하는 추세에 있고, 또한 계속적으로 발전하고 있다. 이러한 서비스 중에서 현재 가장 많이 이용되고 있는 것이 메일시스템이다. 본 논문에서는 메일 서버의 Virtual Domain 기능을 이용하여 메일사용자를 위한 계정관리가 필요 없도록 지원하고 사용자가 받을 메일을 미리 사용자에게 분류하는 기능을 지원하여 대량의 메일을 효율적으로 처리하도록 하였다.

#### Abstract

There has been increasing of using Internet as time goes by, and it means that the analysis technique of appetite for the users logging into the internet system has been absolutely advanced. Keeping pace with the intense expansion of information, the mailing system is the one to be very busy on Internet service at the present time. This paper introduces to support users that carry handling account management easier by using virtual domain control on mail server. It also finds out the merit to get more efficient way to control the bunch of mails that classify into mail-box to users in advance.

---

\* 마산대학 멀티인터넷학부 부교수  
\*\* 마산대학 멀티인터넷학부 겸임강사  
\*\*\* 경남 고성중앙고등학교 교사

## 1. 서론

현재의 전자우편 기본규약은 1982년에 표준으로 지정된 RFC822를 기반으로 자료교환이 이루어지고 있다[1].

RFC822에서는 단지 7비트(Bit) ASCII 영문자만 취급할 수 있었다[2]. 그러나 1992년 6월에 멀티미디어 메일(Mail) 전송의 필요성에 따라 RFC822를 수정 확장하여 새로운 규정이 만들어졌다.

이 규정을 MIME(Multipurpose Internet Mail Extensions)라고 한다[3]. 이 규정에 의해서 제한 없이 전자우편을 사용할 수 있게 되었으며, 7비트(Bit) ASCII가 아닌 일반 문자들도 사용할 수 있게 되었다. 또한 다양한 글꼴(Font)과, 이진파일(Binary Files)을 전송하게 되었을 뿐만 아니라, 이미지(Image), 목소리(Audio), 동화상(Video) 등 다양한 멀티미디어 메시지를 주고받을 수 있게 되었다[4]. 또한 96년 중반기부터 주목받기 시작한 IMAP4 프로토콜은 기본적인 역할은 POP3와 동일하지만 기능적으로 비교하면 IMAP4프로토콜이 뛰어나다[5]. IMAP4는 저속의 네트워크에 연결된 사용자도 사용할 수 있으며 편지함 관리 등 다양한 기능을 사용할 수 있다[6].

본 논문에서는 하나의 시스템을 이용하여 Virtual Domain을 지원하는 웹메일 시스템과 사용자의 계정 관리 없이 메일을 사용할 수 있도록 구성하였다. 따라서 관리자 측면에서 효율적으로 웹메일을 관리할 수 있으며, 사용자가 받을 메일을 각 사용자에게 미리 분류하는 시스템을 지원함으로써 시스템 지원을 효율적으로 이용할 수 있도록 웹메일 시스템을 설계 및 구현하였다.

## II. 시스템 설계 및 구현

### 1. 연구 배경

현재 사용되고 있는 웹메일 시스템은 단일 도메인을 사

용하게 되고, 사용자가 메일을 사용하기 위해서 계정의 추가와 각 계정의 관리가 필요하며, 각 운영체제에 따라서 시스템을 개발해야 하는 어려움을 가지고 있다[9].

따라서 본 논문에서는 이러한 문제점을 개선하기 위하여 운영체제에 독립적으로 작동가능한 언어인 Python을 사용하여 가상도메인을 이용한 웹메일 서비스를 구현하고자 한다. Virtual Domain이란 기본도메인과는 전혀 다른 도메인으로 같은 서버에 연결되어있는 도메인을 말하는 것으로, <그림 1>은 Virtual Domain에 대한 부분이다.

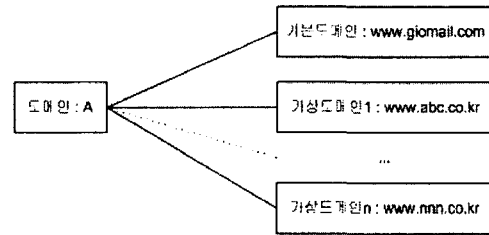


그림 1. Virtual Domain

### 2. 시스템 설계

본 논문에서는 MWMS(Multi-Domain Web Mail System) 시스템을 구현하기 위해 Python언어와 큐메일 시스템을 사용하였으며, 서로 다른 운영체제에서 메일 서비스를 제공한다[7,8].

MWMS는 한 시스템으로 여러 도메인의 메일을 받을 수 있도록 구성하며, 각 사용자들이 메일을 사용하기 위해서 계정의 추가 없이 사용자별로 웹메일을 사용할 수 있도록 하는 서비스를 제공한다.

<그림 2>는 전체 시스템 설계를 나타내며 다수의 클라이언트가 메일 요구 시, 웹메일 시스템은 메일서버에 접속하여 메일의 내용을 확인하고 클라이언트에게 메일 내용을 보여준다.

웹메일 시스템은 크게 두 부분으로 구성되었다. 우선 SMTP Module은 메일 전송 때 사용된다. 그리고 IMAP Module은 메일 확인 및 관리를 하기 위해서 사용된다.

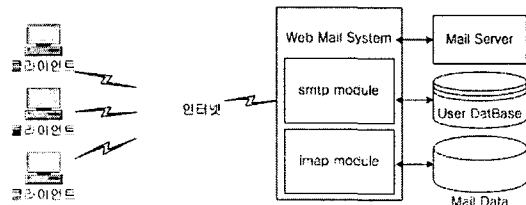


그림 2. 전체 시스템 설계

### 1.1 시스템 상세설계

본 시스템의 구성은 메일서버의 Virtual Domain기능을 이용하였다. 따라서 여러 도메인 및 여러 계정으로부터 도착한 메일을 한 개의 계정으로 도착하도록 하였으며, 메일의 header 정보를 분석하여 각 사용자 및 도메인별로 메일을 보여줄 수 있도록 구성된 시스템이다.

시스템의 전체 구성은 그림 3과 같이 메일 관리를 위한 Mail Module, 사용자 관리를 위한 UserManager Module, 새로운 메일 확인 및 각 사용자별로 메일 분류를 위한 MailCheck Module, DataBase를 관리하는 DB Module 등 4개의 모듈로 구성되어 있다.

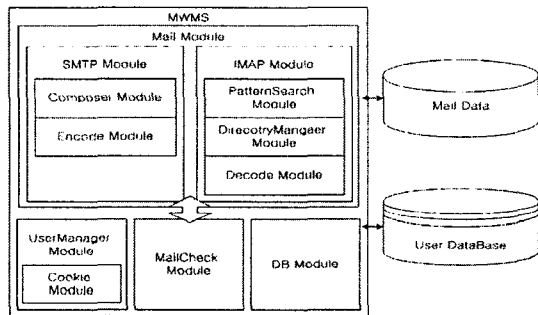


그림 3. 상세 시스템 설계

### 2.2 구성모듈의 고찰

#### (1) SMTP Module

SMTP Module은 Composer Module을 호출하여 메일의 Header 부분을 구성하며, Header 정보 중에서 MIME이 필요한 부분에 대하여 Encode Module을 호출하여 메일을 완성시킨다. 완성된 메일은 메일 서버를 통하여 보내고 싶은 사용자에게 전송하도록 구성되어 있다.

#### (2) Composer Module

Composer Module은 SMTP Module에 의해서 호출되며 From, To, CC, Bcc, In-Reply-To, Subject, X-Mailer, MIME-Version, Organization, Date 등으로 구성된 Header 정보를 구성하고 첨부파일은 Encode Module을 호출하여 메일을 완성시킨다.

#### (3) Encode Module

Encode Module은 Composer Module에 의해서 호출되며, 메일의 Header 정보와 첨부파일에 사용되어 MIME Type을 구성한다.

#### (4) IMAP Module

IMAP Module은 사용자가 받은 메일을 관리하고 내용을 보여준다. 각각 받은 메일은 newmail, receive, send, delete의 디렉토리로 메일을 관리하게 된다.

여기서 newmail 디렉토리는 사용자가 새로 도착한 메일을 받을 때 보관한다. 그리고 receive 디렉토리는 사용자가 새로 도착한 메일을 확인시 이동하게 되어 받은 메일로 보관한다. send는 사용자가 보낸 메일을 보관하며, delete는 사용자가 삭제한 메일을 보관하는 부분들로 구성되어 있다. 각각 받은 메일은 Header를 분석하여 메일을 받는 사람, 받은 시간, 메일 크기, 제목 등을 Decode Module을 호출로 Header 정보와 Body 내용을 보여준다. 그리고 첨부파일이 존재하면 Decode Module을 호출하여 사용자에게 MIME Type을 보여주고 다운로드 받을 수 있도록 구성되어 보여준다.

#### (5) Directory Module

사용자가 받은 파일을 디렉토리 단위로 관리할 수 있도록 구성되며, "사용자아이디.메일 Box"의 형태를 이룬다. 메일 Box는 기본적으로 각 사용자별로 newmail, receive, send, delete의 디렉토리를 만들어 관리하게 된다.

#### (6) PatternSearch Module

PatternSearch Module은 DB Module로부터 사용자가 시스템에서 사용하고 있는 이메일 주소를 넘겨받아 받은 메일의 Header 정보를 분석하여, 이메일 주소와 일치하는 사용자들에게 메일을 분류할 수 있도록 메일의 Header 정보를 분석하는 Module이다.

#### (7) Decode Module

Decode Module은 사용자가 메일의 내용을 확인할 때 호출된다. 이는 메일의 Header 정보와 첨부파일에 사용되어 MIME를 보여주기 위한 Module이다.

#### (8) MailCheck Module

MailCheck Module은 PatternSearch Module을 호출하여 메일서버가 받은 메일을 확인하는 모듈로 구성되어 있다. DB Module의 사용자 정보 중에서 이메일 부분을 PatternSearch Module로 넘겨주고, 각 메일의 Header 정보 중에서 Received, Delivered-To, To 등의 내용을 메일서버로부터 가져와서 일치하는 이메일을 사용자 별로 분류하여 각 사용자의 newmail 디렉토리로 메일을 이동하도록 구성하였다.

MailCheck Module이 필요한 이유는 각 사용자가 로그인할 때마다 새로운 메일 여부를 체크하는 것은 시스템 부

하에 영향을 미치게 되며, 많은 메일을 확인해야 될 때 시간적으로 반응시간이 너무 느리기 때문에 시스템 차원에서 메일을 미리 확인하기 위한 Module이다.

(9) UserManager Module

UserManager Module은 DB Module로부터 사용자 정보를 전송받아 각각 필요한 Module에게 사용자 정보를 전송한다. 그리고 Cookie Module로부터 관리된 부분을 확인하여 불법적인 접속을 할 수 없도록 구성되어 있다.

(10) DB Module

DB Module은 DataBase의 연결과 사용자 정보를 관리하며, UserManager Module에서 사용자 정보를 요구하면 DB Module이 사용자 정보를 전송한다.

따라서 전체적인 데이터흐름은 MailCheck Module에 의해 각각의 필요한 모듈들이 호출되어 사용자별로 메일이 분류되는데, 이때 Directory Module에 의해 사용자 별로 관리된 디렉토리로 메일은 이동하게 된다. 분류된 메일은 사용자 로그인 후 확인하게 된다. 메일작성시는 각 엔코딩에 필요한 Module을 호출하여 작성된 메일은 SMTP Module을 호출하여 메일을 전송하게 된다.

3. 시스템 구현

3.1 시스템구성

본 논문에서 제안한 시스템은 메일서버의 Virtual Domain 기능을 이용하여 사용할 도메인을 등록하고, 가상으로 사용할 사용자 E-Mail도 함께 등록하여, 등록된 사용자 E-Mail로 전송된 메일을 하나의 계정으로 받을 수 있도록 시스템을 구성한다. 그리고 사용자가 웹메일 시스템을 사용할 수 있도록 하기 위하여 DataBase에 등록하여 사용하도록 하였다. 제안된 시스템의 설치 및 운용에 대한 구현 환경은 표 1과 같다.

표 1. 구현 환경

분류	웹메일 시스템
사용 운영체제	Linux Redhat 7.2
탑재 프로세서	펜티엄 II 400
RAM 크기	512M
개발언어	Python 1.5.2
사용 데이터베이스	MySQL 3.23
사용 서버/사용 브라우저	Apache 1.3.22/ Explorer 6.0
사용 메일서버	QMail 1.0.3

3.2 구현알고리즘

<그림 4>는 각 사용자들에게 메일을 분류하도록 하는 PatternSearch 알고리즘을 나타내고 있다. 찾고 싶은 이메일 주소를 넘겨주면 PatternSearch Module은 메일의 특정부분을 가져와서 해당 메일 주소와 일치 여부를 확인하게 된다.

```

array PatternSearch(pattern)
{
    i = 0;
    find = false;
    list() = null;
    total = 메일 전체수;
    while(i++ <= total) {
        headerinfo = i 번째 메일 Header 정보가져오기;
        addrlist() = headerinfo 중에서 'Received' 부분의 주소만 가져오기
        j = 0;
        while(addrlist(j++)) {
            if (search(pattern, addrlist(j))) {
                list()에 i 추가;
                find = true;
                break;
            }
        }
        if (!find) {
            if (search(pattern, headerinfo의 'To' 부분내용))
                list()에 i 추가;
            else if (search(pattern, headerinfo의 'Delivered-To' 부분내용))
                list()에 i 추가;
        }
    }
    return list()
}
    
```

그림 4. PatternSearch 알고리즘

3.3 구현 시스템의 실행 결과

제안된 시스템은 인터넷 사용이 가능한 환경과 웹브라우저만 설치되어 있으면 장소에 관계없이 사용이 가능하다.

<그림 5>는 Directory Module에 의해서 각 사용자별로 디렉토리가 만들어진 결과를 나타내고 있다.

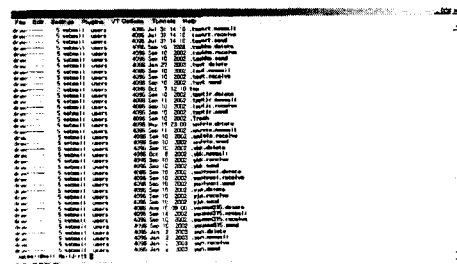


그림 5. 사용자별 메일 Directory 관리

newmail 디렉토리는 새로 받은 메일, receive 디렉토리는 읽은 메일, send 디렉토리는 보낸 메일, delete 디렉토리는 삭제된 메일들이 위치하게 된다.

〈그림 6〉은 웹메일 시스템에서 사용자 확인을 위한 Login 화면이다. 사용자 아이디와 비밀번호를 입력하면 웹 메일을 사용할 수 있고, 메일을 관리할 디렉토리가 없을 경우 디렉토리를 생성하게 된다.

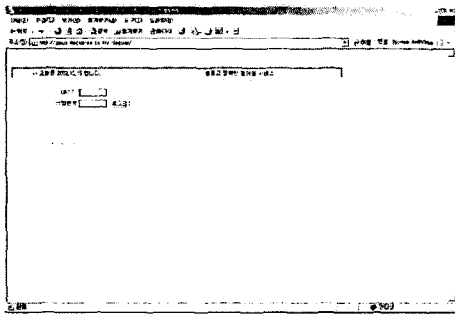


그림 6. Login 화면

〈그림 7〉은 사용자가 받은 메일을 확인하는 화면으로써, 윗 부분은 사용자가 받은 메일의 Header 부분의 정보를 보여준다. 그리고 아랫 부분은 사용자가 보고 싶은 메일의 Header 부분과 Body 부분을 함께 보여주는 화면으로 구성되었다.

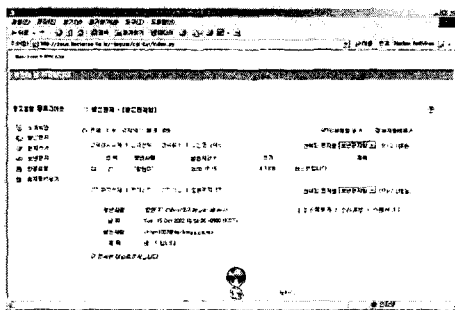


그림 6. 받은 메일 확인 화면

### III. 결론

본 논문에서는 현재 사용되고 있는 웹메일 시스템의 단일 도메인의 문제점을 개선하기 위하여, 메일서버의 기능 중에서 Virtual Domain 과 Virtual User 기능을 이용하여 계정관리가 아닌 DataBase 관리로 이루어지도록 함으로써, 보다 더 효율적인 시스템 관리가 이루어 질 수 있도록 설계하고 구현하였다.

본 시스템은 웹호스팅이나 메일 서비스를 제공하는 업체에 더욱 더 효율적인 메일서비스를 제공할 수 있을 것으로 사려된다. 그러나 본 시스템을 사용하기 위해서는 다수의 작업을 shell 상태에서 해야 한다는 어려움이 있으므로, 모든 작업을 웹에서 설정할 수 있도록 하는 지원이 필요하다.

향후 좀더 완벽한 웹메일 시스템의 구현과 XML을 지원 하는 시스템에 대한 추가적인 연구가 필요하다.

### 참고문헌

- [1] Jonathan B. Postel, "Simple Mail Transfer Protocol," RFC821, 1982.
- [2] David h. crocker, "Standard for the format of ARPA Internet Text Messages," RFC822, 1982.
- [3] Moore.K., "MIME(Multipurpose Internet Mail Extension): Part Two: Message Header Extension for Non-ASCII Text," RFC1522, Univ. of Tennessee, 1993.
- [4] Korean Character Encoding for Internet Messages, RFC1557, 1993,  
<http://www.faqs.org/rfcs/rfc1557.html>
- [5] M. Crispin, "IMAP4(INTERNET MESSAGE ACCESS PROTOCOL-VERSION 4rev1", RFC2060, Univ. of Washington, 1996

- [6] IMAP Information Center, Univ. of Washington, 2002, <http://www.cac.washington.edu/imap/>
- [7] Introductions to Python programming for non-programmers, <http://www.python.org>
- [8] Mod\_python - Apache/Python Integration, 2002, <http://www.modpython.org/>
- [9] 한은용, 배대룡, "인터넷 웹 메일 시스템의 설계 및 구현", 한국정보처리학회 추계 학술발표 논문지 제7권 제2호, pp399-402, 2000

### 저자 소개



#### 류 언 무

1983년 광운대학교 전자계산학과  
졸업(이학사)  
1987년 숭실대학교 산업대학원  
전자계산학과 졸업(공학석사)  
1998년 명지대학교 대학원  
전자계산학과 졸업(공학박사)  
1989년 ~ 현재 마산대학  
멀티인터넷학부 부교



#### 황 현 미

1996년 경남대학교 전자계산학과  
졸업(공학사)  
2003년 경남대학교 교육대학원  
전자계산학과 졸업(교육석사)  
2003년 ~ 현재 마산대학  
멀티인터넷학부 겸임강사



#### 이 상 철

1989년 경상대학교 사범대학  
수학교육과 졸업  
2000년 경남대학교 교육대학원  
전자계산학과 졸업(교육석사)  
2002년 ~ 경남고성중앙고등  
학교근무