지구물리 제 4권 제 3호, p. 145-162 (2001년 9월)

효율적인 웹기반 전자 시스템의 설계 및 구현

안성옥 · 유성종 · 유현경
배재대학교 컴퓨터공학과

요 약

인터넷이 급속도로 발전하고 일반 텍스트 환경의 인터넷 서비스들의 WWW(World Wide Web)환경으로 바뀌어 가면서 쉬운 사용 환경으로 인한 인터넷 응용 서비스의 사용자들도 급속히 증가하고 있다. 따라서, 사용자들이 인터넷을 이용하면서 상대방에게 메시지를 보낼 수 있는 전자 메일 서비스를 이용하는 경우도 크게 증가하고 있다.

웹 기반 전자우편 시스템은 계정과 서비스를 제공하는 서버와 사용자와의 인터페이스 역할을 하는 클라이언트로 구성되며 웹 브라우저가 클라이언트의 역할을 담당한다. 즉, 일반 메일서버에 계정을 만들 수 없는 일반 사용자들이 웹 환경을 통하여 메일서비스를 제공받을 수 있도록 하는 것이다.

본 논문에서는 웹을 기반으로 하는 전자 우편 시스템을 설계하였으며, 인터넷 익스플로러(Internet Explorer)를 기반으로 하는 사용자 환경과 리눅스 시스템을 기반으로 기존 전자 우편 시스템의 문제점을 해결하고 저렴한 가격으로 큰 기대효과를 만족시켰다. 스팸으로 등록된 메일 처리, 다중 메일 처리 등을 통한 효율적인 사용자 편의성을 제공하고, 메일의 폭주와 사용자의 급증으로 인한 시스템 저하 현상을 극복하는 효율적인 메일 서비스 엔진을 구현해서 시스템의 안정성을 제공하는 전자 우편 시스템을 개발한다.

An, Syungog, Yoo, Sungjung, Yoo, The Design and Implementation of a Effective web-based electronic system

ABSTRACT: With the rapid advance of internet service and the corresponding migration of service environment from the text-based one to WWW (World Wide Web) environment, the number of internet users is growing rapidly due to its easy usage. Accordingly, usage of internet as services for sending electronic mails to the other party over the network is becoming increasingly prevalent. Web-based electronic mailing system is comprised of a server and a client. The former provides the users with e-mail accounts and services, while the latter serves as a user interface. In other words, it enables those public users who dos not own e-mail accounts on the existing mail server to have an access to the mailing service through the web.

In this paper, we designed a effective web-based electronic mailing system which is based on the internet explorer and linux operating system, which overcomes limitations of the existing e-mail systems and meets the need of a cost-efficient alternative. Our electronic mailing system also supports the convenience of users through appropriate handling of preregistered spam e-mails and multiple e-mails, and this facilitates the development of a stable e-mail system by being able to avoiding the low system performance due to the bursty characteristics of e-mail messages and the increasing number of users

1. 서 론

인터넷 기술이 발전함에 있어서 전자우편은 가장 활용성이 높은 인터넷 애플리케이션 중의 하나로서 TCP(Transmission Control Protocol)의 연결들 중 약 절반이 SMTP(Simple Mail Transfer Protocol)[1]를 위해 설정되어 있다.

그러나, 기존의 메일시스템은 메일 서비스를 받기 위해서 적절하게 세팅된 클라이언트 프로그램이 필요하다. 이와 같은 기존의 클라이언트/서버 메일 시스템들은 독자적인 클라이언트 소프트웨어의 사용을 요구하기 때문에 제한적이고 폐쇄적이며 번거로워 사용자들에게 불편함을 안겨 주었다. 그래서 이러한 문제점들을 해결하기 위해 메일 서버에 계정을 만들 수 없는 일반 사용자들에게 웹 환경을 통하여 메일 서비스를 제공받을 수 있도록 하는 웹 기반의 메일 서비스가 현재 제공되고 있다.

웹기반 메일 시스템은 어떤 클라이언트 소프트웨어보다도 보편적으로 이용되고 있는 웹 브라우져를 메일에 대한 클라이언트 소프트웨어로 이용함으로써 어디서나 사용이 가능하며, 메일 사용자들은 웹 브라우져 이외에 어떤 새로운 클라이언트 소프트웨어도 설치할 필요가 없게 된다.

본 논문에서는 이러한 웹기반 메일 시스템을 보다 효율적이고 다양한 기능을 제공하는 웹기반 메일 시스템을 SMTP를 이용해 설계하고 구현한다.

본 논문의 구성은 다음과 같다. 2절은 연구목적을 정의하고 2절은 웹기반 전자우편 시스템의 밑거름이 된 관련 연구들에 대해 살펴본다. 그리고 3절에서는 시스템의 구축 환경을 나타내며 4절에서는 제안된 웹기반 전자우편 시스템의 세부 구현 내용과 특징을 살펴본다. 5절에서는 구현 결과 및 주요 기능을 사진을 통해 설명하고 6절에서는 웹기반 전자우편 시스템의 향후 과제에 대해서 설명한다.

2. 관련 연구

2.1 POP3(Post Office Protocol 3)

POP은 RFC1460에 기술되어 있으며, 메일 서버에서 메일을 회수하는 프로토콜이다.

기존의 메일서버는 메일을 회수하기 위해서는 메일 서버에 로그인 한 후 메일 프로그램을 사용해서 메일을 확인하였으나, 현재 모든 Mail Agent(Netscape, Outlook Express, Eudora 등)은 이 POP 및 SMTP를 사용하여 메일 서버에 직접 로그인 하지 않고 메일을 확인할 수 있으며, 메일을 보낼 수 있다.[2]

2.2 IMAP4(Internet Messaging Access Protocol 4)

IMAP 클라이언트가 메일 서버에서 메일을 읽기 위한 인터넷 표준 프로토콜의 한 가지로서, TCP/IP의 상위 프로토콜이며 최신 버전인 IMAP4는 RFC 2060에 규정되어 있다. POP3보다 유연하고 뛰어나다고 할 수 있다.

서버측에 메일 박스를 둘 수 있다는 것과 메일 헤드만 읽을 수 있다는 것이 특징이다. 클라이언트는 서버의 메일 박스에서 메일을 삭제하지 않고 필요한 메일만 복사할 수도 있다.[2]

2.3 SMTP(Simple Mail Transfer Protocol)

SMTP는 두 MTA(Mail Transfer Agent) 사이에서 전자 우편을 전달하는 표준 규약이다. Internet의 Mail System은 RFC 821 “Simple

Mail Transfer Protocol"에 정의되어 있으며 또한 SMTP에서 사용되는 메시지의 형식을 RFC 822에 정의하고 있다. SMTP는 기본적으로 HELO, MAIL, RCPT, DATA, QUIT과 같은 5가지의 명령어를 사용하여 메시지를 전송한다. HELO는 클라이언트가 서버에게 자신의 호스트명과 도메인 메시지를 전송한다. HELO는 클라이언트가 서버에게 자신의 호스트명과 도메인(Domain)을 알려주는 명령이고, MAIL은 메시지를 보낸 사람의 전자 우편 주소를 알리는 명령이며, RCPT는 전자 우편 메시지를 받을 사람의 전자 우편 주소를 알려준다. DATA는 실제로 메시지를 전달할 때 사용하는 명령이며, QUIT는 클라이언트와 서버사이의 전자우편 메시지 전달을 종료하는 명령이다.[3]

2.4 MIME(Multipurpose Internet Message Extensions)

RFC 822의 표준을 확장하여 기존의 전자우편 메시지와 호환성을 유지하면서 멀티미디어 등의 다양한 메시지 형식 처리 및 다국어 처리를 위해 제안된 것으로, 기존의 전자 우편 시스템에서 SMTP에 따른 MTA가 전송할 수 있는 메시지가 7비트 ASCII 코드 형태이므로 이를 보완하고 X.400사용자들의 요구를 수용하여 X.400과 연동하여 수행하도록 되어 있다. MIME의 메시지 형식은 버전을 나타내는 MIME-Version, 데이터 종류를 나타내는 Content-Type, SMTP Content-Transfer-Encoding 필드가 있다.[4]

[그림 2-1]에서는 일반적인 MIME Format를 나타내고 있다.

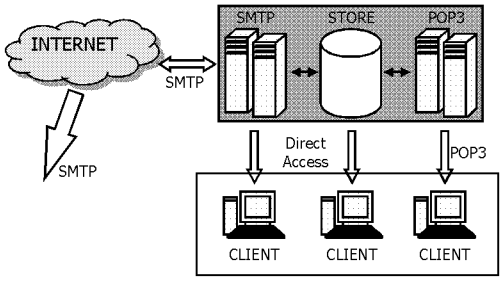
```
Return-Path: <4unme@paws21.com>
Delivered-To: 4unme@paws21.com
Received: (qmail 3992 invoked by uid 99); 14 Sep 2001
17:43:57 +0900
Message-ID: <20010914174357.3991.qmail@paws21.com>
Reply-To:    =?ks_c_5601-1987?B?wK+8usG+?=    <4unme@paws21.c
             om>
From: =?ks_c_5601-1987?B?wK+8usG+?= <4unme@paws21.com>
To: zzong@homenmedia.com,
             4unme@paws21.com,
             forunme007@dreamwiz.com
Subject:     =?ks_c_5601-1987?B?ssAgwN+wob7fuLggx9Kyq7WIILi2
             wfa4tyDF1726xq7A1LTPtNku?=
Date: Fri, 14 Sep 2001 17:43:57 +0900
MIME-Version: 1.0
Content-Type: multipart/alternative;
             boundary="----=_NextPart_000_529E_7445AA53"
X-Priority: 1
X-MSMail-Priority: High
X-Mailer: PAWS21 WebMail Ver1.42

This is a multi-part message in MIME format.

------=_NextPart_000_529E_7445AA53
Content-Type: text/plain;
             charset="ks_c_5601-1987"
Content-Transfer-Encoding: base64

uLbB9ri3IMXXvbrGriDA1LTPtNkuDQoNCsH4wqW3535+fgOKLSOtLSOt

------=_NextPart_000_529E_7445AA53
Content-Type: text/html;
             charset="ks_c_5601-1987"
Content-Transfer-Encoding: base64

PCFETONUWVBFIEhUTUwgUFVCTEIDICItLy9XMOMvLORURCBIVE1MIDQu
YXdzMjG/obyttMIgaHRtbMf8xcK3ziC6uMDMseYguve0z7TZLiANPGJy

------=_NextPart_000_529E_7445AA53--
```

[그림 2-1] 일반적인 MIME Format

2.5 기존의 전자우편 시스템

전자우편 클라이언트 프로그램으로 SMTP 서버에 편지를 보낸다. 전자우편 메시지 수신인의 도메인이 서버가 있는 도메인과 같으면 아스키 텍스트를 서버에 저장하고, 자기 도메인이 아닌 경우 해당 서버에 전달한다. 전자우편 메시지가 수신자의 SMTP서버에 도착하면 수신자에 할당된 사서함에 저장된다. 수신자는 아웃룩 익스프레스 같은 우편 클라이언트에서 POP3 프로토콜로 서버에 접속해서 자신에게 온 편지를 가져간다. 이 과정에서 포트25를 사용하는 SMTP로 편지를 서버에게 전달하고, 포트110을 사용하는 POP3프로토콜로 편지를 클라이언트까지 배달한다.

[그림 2-2]는 전자우편 배달 과정을 보여준다.

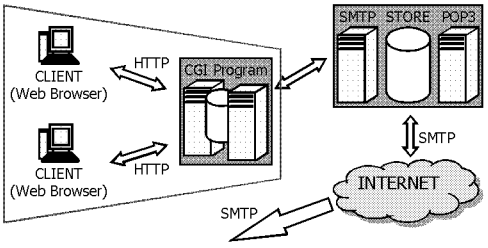


[그림 2-2] 기존의 전자우편시스템

일부 시스템에서는 SMTP 편지 저장 공간에 직접 접근 할 수 있다.

반면 웹 기반 전자우편 시스템은 계정과 서비스를 제공하는 서버와 사용자와의 인터페이스 역할을 담당한다. 브라우져를 통해서 전자우편 서비스를 제공하는 웹 사이트에 접속한 사용자는 ID와 암호를 가지고 자신의 정보를 관리하게 된다. 또한 사용자는 서버에 자신의 전자우편 주소를 가지게 되면 이를 이용하여 다른 사용자와 전자우편을 주고 받을 수 있다. 전자우편 클라이언트 역할을 하는 브라우져는 사용자로부터 데이터를 입력받으며, 이는 HTTP(HyperText Transmission Protocol)[6] 프로토콜을 이용하여 서버에 존재하는 CGI(Common Gateway Interface)프로그램에게 전달된다. CGI프로그램은 전달 받은 사용자 데이터를 수정하여 실제로 전달 가능한 전자우편의 형태로 만들며 SMTP프로토콜을 이용하여 수신자에게 전달된다.

[그림 2-3]은 웹 기반 전자우편 시스템의 동작을 보여준다.



[그림 2-3] 웹기반 전자우편 시스템

3. 시스템 환경

시스템의 환경은 기존의 서버/클라이언트 메일 시스템의 문제점을 해결하고 저렴한 가격으로 큰 기대효과를 위해 리눅스 시스템을 채택한다. 물론 일반적인 유닉스 시스템에 모두 유연하게 적용될 수 있다.

DataBase로는 mysql 서버를 사용하며 mail MTA로는 요즘 완벽한 보안성과 다중 사용자 인증 기능으로 각광 받고 있는 qmail을 채택한다.

사용자 환경과 관리자환경을 구축하기 위해서는 HTML[7]과 PHP script를 사용하여 구현하였으며 서비스엔진 서버측 모듈을 생성하기 위해 C++을 사용한다.

4. 웹 기반 메일 시스템 구현

제인된 웹기반 전자우편 시스템은 리눅스 기반에 PHP를 활용하여 구현된다.

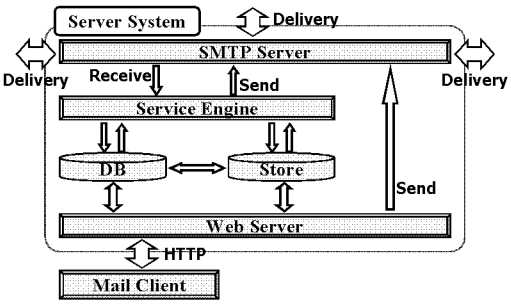
기본적으로 PHP내의 IMAP Function을 이용하여 기반 MTA와 연계하여 시스템이 구현된다.

시스템에 실제 계정을 없애고 가상 계정을 두어 실제 계정을 사용하면서 발생하는 접속 지연의 문제를 제거한다. 이 가상 계정의 구현은 DataBase와 MTA와의 결합하여 접속 권한을 DataBase의 쿼리문 하나로써 빠르게 처리할 수 있다. 또한 가상

계정을 사용함으로써 실제 시스템에 외부 접속 권한이 없기 때문에 보안상의 장점도 가진다.

4.1 제안된 웹기반 전자우편 시스템 구조

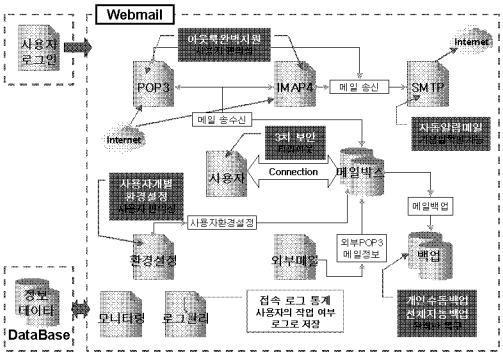
본 논문에서 제시한 웹 기반 전자우편 시스템의 구조는 기존의 전자우편 시스템의 구조를 바탕으로 하였다. [그림 4-1]은 전자우편 서버에 계정을 가진 사용자가 메시지를 작성하여 전달하는 구조를 보여주고 있다. 사용자는 클라이언트 역할을 하는 브라우저를 통해 메일을 송신하거나 수신한 전자우편을 확인할 수 있다.



[그림 4-1] 제안된 웹기반 전자우편 시스템 구조도

4.2 제안된 시스템의 자료흐름

[그림 4-2]는 앞의 [그림 4-1]의 구조도의 자료흐름을 표현한 자료흐름도 이다.

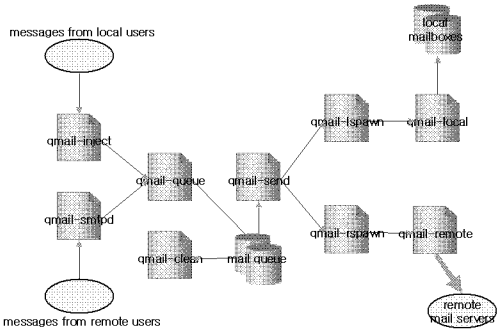


[그림 4-2] 제안된 시스템의 자료흐름도

4.3 qmail System의 분석

[그림 4-3]은 제안된 메일 시스템의 MTA인 qmail System의 구조도 이다. qmail은 완벽히 분산된 구조로 시스템 성능이 보다 많이 향상되는 장점이 있다.

qmail은 또한 로컬 사용자간의 메일 처리 루틴과 외부 사용자로부터의 메일 처리 루틴을 나누어 두 개의 프로세스가 별개의 동작을 한다.

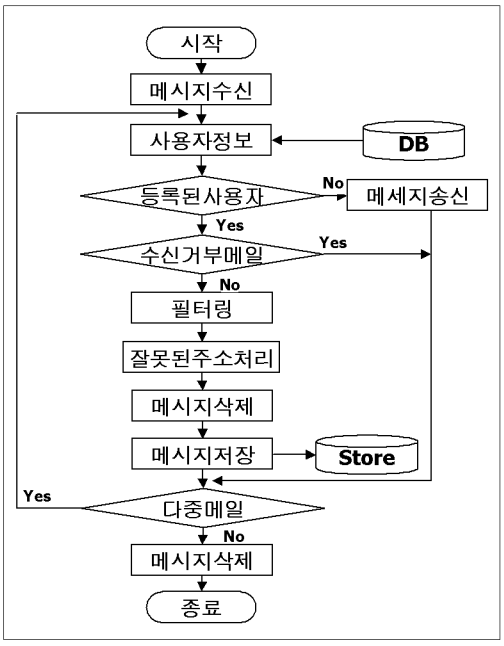


[그림 4-3] qmail 시스템 구조도

4.4 메일 시스템 프로그램 흐름도 및 설계

[그림 4.4]는 제안된 메일 시스템의 프로그램 흐름도 이다. 기본적으로 설계된 흐름도의 각 부분을 모듈화 하여 구현하도록 한다.

[그림 4.4]에 제시된 기능 이외에 주소록 관리 기능과 파일 관리 기능과의 연동으로 사용자 인터페이스를 최대한 편리하게 처리하며, 각각 루틴 또한 모듈화 하여 효율적인 루틴으로 구현한다.



[그림 4-4] 메일 시스템 프로그램 흐름도

① 시작
② 메시지 수신 처리
③ 사용자 정보 DataBase를 이용하여 확인
④ 등록된 사용자 여부 확인
⑤ 메시지 송신 처리
⑥ 스팸으로 등록된 메일 처리
⑦ 메일 자동 분류 기능 처리
⑧ 잘못된 메일 주소 반송 처리
⑨ 메시지 읽기, 삭제, 이동 등 기능 처리
⑩ 메시지 저장 기능 처리
⑪ 다중 메일 처리
⑫ 메시지 삭제 처리
⑬ 종료

위의 루틴에서 ⑨⑩⑪⑫ 의 루틴을 반복하면서 사용자의 메일을 관리하게 된다.

4.5 기반 DataBase 구현

```
인증 테이블
CREATE TABLE auth (
  mbox_host varchar(100) binary NOT NULL default '',
  id varchar(40) binary NOT NULL default '',
  name varchar(50) binary NOT NULL default '',
  crypt varchar(40) binary NOT NULL default '',
  passwd varchar(40) binary NOT NULL default '',
  uid int(10) unsigned NOT NULL default '65534',
  gid int(10) unsigned NOT NULL default '65534',
  shell varchar(100) binary NOT NULL default '',
  home varchar(100) binary NOT NULL default '',
  start_date date NOT NULL default '2000-01-01',
  expire_date date NOT NULL default '2005-12-31',
  last_conn date NOT NULL default '0000-00-00',
  active enum('Y','N') NOT NULL default 'Y',
  quota_size int(24) unsigned NOT NULL default '10240'
) TYPE=MyISAM;
사용자 테이블
CREATE TABLE user (
  mbox_host varchar(100) binary NOT NULL default '',
  id varchar(40) binary NOT NULL default '',
  sn1 varchar(6) binary NOT NULL default '',
  sn2 varchar(7) binary NOT NULL default '',
  sex enum('M','F') NOT NULL default 'M',
  birth date default '0000-00-00',
  calendar enum('+','-') NOT NULL default '+',
  homezip varchar(7) binary default '',
  homeaddr varchar(100) binary default '',
  hometel varchar(14) binary default '',
  hp varchar(14) binary default '',
  job varchar(50) binary default '',
  office varchar(50) binary default '',
  officezip varchar(7) binary default '',
  officeaddr varchar(100) binary default '',
  officepart varchar(30) binary default '',
  officeposition varchar(30) binary default '',
  officetel varchar(14) binary default ''
) TYPE=MyISAM;
파일 첨부 테이블
CREATE TABLE attach (
  mbox_host varchar(100) binary NOT NULL default '',
  id varchar(40) binary NOT NULL default '',
  home varchar(100) NOT NULL default '',
  file_name varchar(100) NOT NULL default '',
  file_size int(24) unsigned NOT NULL default '0',
  file_type varchar(100) NOT NULL default '',
  file_body longblob NOT NULL
) TYPE=MyISAM;
```

[소스 4-1] DataBase 테이블 소스

[소스 4-1]의 auth 테이블은 사용자의 MTA 인증 테이블로서 이 테이블로서 MTA와 DataBase가 연계된다. 사용자 정보테이블은 따로 두어 auth 테이블은 MTA와 연계될 때 성능 향상을 위해 구조를 최소화한다.

auth 테이블의 각 필드는 각각 메일 시스템에서의 기능들을 가지는데 그 기능들은 아

래의 표와 같다.

mbox_host	시스템의 도메인
id	아이디
crypt	MTA가 사용하는 패스워드
passwd	개발자가 사용하는 패스워드
uid	User Identify
gid	Group Identify
shell	계정의 시스템 쉘
home	계정의 메일 저장소
start_date	계정 생성 날짜
expire_date	계정 만기 날짜
last_conn	마지막 로그인 날짜
active	계정 사용 가능 여부
quota_size	계정 사용 가능 용량

[표 4-1] AUTH 테이블의 각 기능

4.6 기반 Class 구현

메일 읽기나 쓰기 편지함 생성 모든 부분에서 MailBox, MailHeader, MailBody 의 연계가 필요하다, 그러므로 본 시스템에서는 각각의 Class를 구현하여 모듈화 하였다.

각 클래스의 내부에 각각의 Member Function을 두어 보다 효율적인 프로그래밍을 하였다.

[소스 4-2], [소스 4-3], [소스 4-4]는 각각 MailBox, MailHeader, MailBody 클래스의 알고리즘이다.

```
# -> MailBox Infomation Class
class MboxInfo
{
    var $perpage, $stream,$mbox, $msgid, $count;
    var $msgno, $rntno, $ureno, $tsize, $lsize, $begin;
    var $end, $sortend, $criteria, $reverse, $sort;
    var $mboxlist, $mboxbody;
    function MboxInfoInit()
    {
        $temp = imap_mailboxmsginfo($this->stream);
        $this->msgno = $temp->Nmsgs;
        $this->rntno = $temp->Recent;
        $this->ureno = $temp->Unread;
        $this->lsize = $temp->Size;
        $this->end                                    =
$this->msgno-(($this->count-1)*$th
is->perpage);
        if($this->end-$this->perpage < 0)
            $this->begin = 0;
        else
            $this->begin = $this->end-$this->perpage;
            $this->sortend                            =
($this->count-1)*$this->perp                  age;
        $this->sort       =       imap_sort($this->stream,
$this->crite             ria, $this->reverse);
    }
}
```

[소스 4-2] MboxInfo(MailBox) Class 알고리즘

```
# MIME Header Class
class MboxHead
{
    var $listdate, $viewdate, $subject, $to, $from, $cc;
    var $bcc, $recent, $unseen, $answered, $deleted;
    var $draft, $flagged, $header, $size;
    var $week = array("일","월","화","수","목","금","토");
    function MboxHeadInit($stream, $msgid)
    {
        $object = imap_header($stream, $msgid);
        DateTimeDecode($object->date);
        $this->listdate = date("Y/m/d", $object->date);
        $this->viewdate = date("Y/m/d(WEEK) Oah:i 발송"
        , $object->date);
        $this->viewdate                               =
ereg_replace("WEEK",$this->wee
k[intval(date("w",          $object->date))],$this->viewdate
);
        $this->viewdate = ereg_replace("am","전",$this->vi
        ewdate);
        $this->viewdate = ereg_replace("pm","후",$this->vi
        ewdate);
        $this->subject = MimeStringDecode($object->subj
        ect);
        $this->to                                     =
MimeStringDecode($object->toaddress          );
        $this->from = MimeStringDecode($object->fromad
        dress);
        $this->cc = MimeStringDecode($object->ccaddres
        s);
        $this->bcc                                    =
MimeStringDecode($object->bccaddr            ess);
        $this->recent = $object->Recent;
        $this->unseen = $object->Unseen;
        $this->answered = $object->Answered;
        $this->deleted = $object->Deleted;
        $this->draft = $object->Draft;
        $this->flagged = $object->Flagged;
        $this->size = $object->Size/1024;
        if($this->subject == "")
            $this->subject = "[제목없음]";
        $this->header = imap_fetchheader($stream, $msgi
        d);
    }
}
```

[소스 4-3] MboxHead(MIME Header) Class 알고리즘

```
# MIME Body Class
class MboxBody
{
    var $type, $subtype, $mime, $encode, $lines, $bytes;
    var $parts, $ifdisposition, $disposition,
$ifdparameters;
    var $dparameters, $ifparameters, $parameters;
    function MboxBodyInit($object)
    {
        switch($object->type)
        {
            case 0:
                $this->type = "text"; break;
            case 1:
                $this->type = "multipart"; break;
            case 2:
                $this->type = "message"; break;
            case 3:
                $this->type = "application"; break;
            case 4:
                $this->type = "audio"; break;
            case 5:
                $this->type = "image"; break;
            case 6:
                $this->type = "video"; break;
            case 7:
                $this->type = "other"; break;
            default:
                $this->type = "text"; break;
        }
        $this->subtype = strtolower($object->subtype);
        $this->mime = $this->type . "/" . $this->subtype;
        switch($object->encoding)
        {
            case 0:
                $this->encode = "7bit"; break;
            case 1:
                $this->encode = "8bit"; break;
            case 2:
                $this->encode = "binary"; break;
            case 3:
                $this->encode = "base64"; break;
            case 4:
                $this->encode = "quoted-printable";
break
            case 5:
                $this->encode = "other"; break;
            default:
                $this->encode = "base64"; break;
        }
        $this->lines = $object->lines;
        $this->bytes = $object->bytes;
        $this->parts = $object->parts;
        $this->ifdisposition = $object->ifdisposition;
        $this->disposition = strtolower($object->dispositio
        n);
        $this->ifdparameters = $object->ifdparameters;
        $this->dparameters = $object->dparameters;
        $this->ifparameters = $object->ifparameters;
        $this->parameters = $object->parameters;
    }
}
```

[소스 4-4] MboxBody(MIME Body) Class 알고리즘

4.7 다중 도메인 서비스 구현

일반적인 메일 시스템은 실제 계정을 사용하여 계정을 생성한다. 그리고 하나의 도메인으로 서비스를 하기 때문에 아이디 중복만 막는다면 메일 서비스를 할 수 있다. 하지만 제안된 웹기반 메일 시스템은 다중 도메인을 지원하면서 아이디 중복을 지원한다.

다중 도메인을 지원하기 위한 알고리즘은 아주 간단하다. 단순한 상식에서 벗어나면 아주 간단히 해결될 수 있는 문제이다.

[예] 아이디 중복의 예

```
sysop@paws21.com      계정 : sysop
sysop@nocom.co.kr     계정 : sysop
```

위와 같이 아이디가 중복되지만 제안된 웹메일 시스템은 각각의 실제 아이디는 sysop이지만 시스템 상의 아이디를 sysop@paws21.com, sysop@nocom.co.kr로 대체 함으로서 중복을 막게 된다. 사실상 시스템 내의 계정 이름은 뒤의 도메인까지 합하여 사용되지만 일반적인 시스템의 관점으로는 중복이 허용되는 것이다.

이러한 다중 도메인과 아이디 중복의 허용은 결과적으로 하나의 시스템이 여러 개의 별도의 시스템을 사용하는 것 같은 가상 시스템을 구현하게 되는 것이다.

4.8 메일 검색 기능 구현

특정 편지함이나, 전체 편지함에 있는 메일을 제목, 보낸사람, 크기등에 포함된 특정 단으로 검색을 하는 기능이다. 간단한 매칭 함수로서 구현 가능하다.

4.9 메일 읽기 구현

메일 읽기 기능은 IMAP에 로그인하여 메일의 헤더를 읽고 해당 정보를 얻은 다음 메일의 Body부분을 파싱하여 원하는 정보를 얻어지게 된다.

시스템상으로 보면 각 메일 파일명으로 받

은 날짜를 구분하며 메일의 상태, 즉 읽혀진 여부와 삭제 여부를 판단할 수 있다.

[소스 4-5]는 메일 읽기에 앞선 로그인과 메일 정렬 부분 알고리즘 이다.

```
# -> 메일 박스 초기화
$objMboxInfo = new MboxInfo;
if(!isset($NowMailBox))
{
    $objMboxInfo->mbox = imap_utf7_encode($MailBox[RE
    AL][1]);
    $NowMailBox = 1;
    $objMboxInfo->criteria =0;          //SORTDATE
    $objMboxInfo->reverse = 1;
}
else
{
    $objMboxInfo->mbox = $MailBox[REAL][$NowMailBox]
    if(!isset($NowMailCriteria))
    {
        $objMboxInfo->criteria = 0;    //SORTDATE
        $objMboxInfo->reverse = 1;
    }
    else
    {
        $objMboxInfo->criteria = $NowMailCriteria;
        $objMboxInfo->reverse = $NowMailReverse;
    }
}
$objMboxInfo->perpage = 15;
if(!isset($NowMailCount))
    $objMboxInfo->count = 1;else
    $objMboxInfo->count = $NowMailCount;
    $objMboxInfo->stream = imap_open("₩{$IMAP4[HOST
    ]:$IMAP4[PORT]}".$objMboxInfo->mbox,$SessionUser
   ."@".$SessionHost, $SessionPass);
```

[소스 4-5] 메일 읽기 로그인 알고리즘

[소스 4-6]은 메일 읽기 기능에서 MIME을 part별로 분석하는 알고리즘이다.

MIME은 part별로 구분되어 있기 때문에 MIME의 구조를 분석하기 위해서 재귀 함수를 이용한다. MIME의 구조를 분석하면서 메일의 본문 내용을 정리하여 반환하는 함수이다.

```
function PrintBodyParts($stream, $object, $parts)
{
    global $MailType,$NowMailBox;
    if(is_array($object->parts))
    {
        if($object->mime == "multipart/alternative")
        {
            $i = 1;
            $parts++;
        }
        else
        {
            $i = 0;
        }
        for($i; $i<count($object->parts); $i++)
        {
            $mbox_outer = new MboxBody;
            $mbox_outer->MboxBodyInit($object->parts[$i
            ]);
            if(is_array($mbox_outer->parts))
            {
                if($mbox_outer->mime == "multipart/altern
                ative")
                    $j = 1;
                else
                    $j = 0;
                for($j; $j<count($mbox_outer->parts); $j++
                )
                {
                    $mbox_inner = new MboxBody;
                    $mbox_inner->MboxBodyInit($mbox_ou
                    ter->parts[$j]);
                    $temp = $j+1;
                    $temp = $parts . "." . $temp;
                    $strMailBody .= PrintBodyParts($strea
                   m, $mbox_inner, $temp);
                }
            }
            else
            {
                $strMailBody .= PrintBodyParts($stream, $
                mbox_outer, $parts);
            }
            $parts++;
        }
    }
}
```

[소스 4-6] MIME Header Parts 분석 알고리즘(재귀)

[소스 4-7]은 일반적인 MIME Decode 알고리즘이다.

이 알고리즘은 Base64코드로 Encoding된 String을 일반 String으로 Decoding하여 반환하는 알고리즘이다.

```
function MimeStringDecode($string) {
    $string = eregi_replace( "=₩?([^?]+₩?[^?]+₩?[^?]+)
    ₩?=", "#@LM#@₩₩1#@RM#@", $string);
    $parts = split("#@LM#@", $string);
    for ($i=1; $i<count($parts); $i++)
    {
        $subparts = split("#@RM#@", $parts[$i], 2);
        $parts[$i] = MimeStringReplace($subparts[0]).$su
        bparts[1];
    }
    return implode("", $parts);
}
```

[소스 4-7] 일반적 MIME Decode 알고리즘

4.10 메일 쓰기 구현

메일 쓰기 기능에는 기본적으로 단순한 메일을 제작 발송 하는 기능을 제공하며 확장된 메일 발송을 지원한다.

확장된 메일 쓰기 기능으로 다중 파일 첨부 기능, HTML확장, 중요도, 보낸편지함에 저장, 메일 수신 확인 기능을 제공한다.

기본적으로 메일은 Base64 코드로 인코딩되어 보내지며 TEXT,HTML,TEXT/HTML방식의 메일 포맷을 지원한다.

메일 MIME Header 작성 알고리즘은 아래와 같다.

메일 쓰기는 MIME을 하나하나 작성하여 IMAP function으로 메일을 전송하게 된다.

① 메일쓰기 폼에서 사용자가 입력한 내용을 넘겨받는다.

② 수신자, 발신자, 제목, 내용등을 MIME Header에 작성한다. 즉, 아래의 알고리즘과 같다.

```
# -> MIME 표준 Header 작성
$StdMIMEHeader = "Reply-To: ".AddrNameEncode($SessionName)." <".$SessionUser."@".$SessionHost.">₩n";
$StdMIMEHeader .= "From: ".AddrNameEncode($SessionName)." <".$SessionUser."@".$SessionHost.">₩n";
$StdMIMEHeader .= "To: ".$MailTo."₩n";
if ($MailCc != "")
    $StdMIMEHeader .= "Cc: ".$MailCc."₩n";
if ($MailBcc != "")
    $StdMIMEHeader .= "Bcc: ".$MailBcc."₩n";
if ($MailSubject != "")
    $StdMIMEHeader .= "Subject: ".AddrNameEncode($MailSubject)."₩n";
else
    $StdMIMEHeader .= "Subject: ₩n";
setlocale("LC_TIME", "ko");
$StdMIMEHeader .= "Date: ".strftime("%a, %d %b %Y%H:%M:%S")." +0900₩n";
if ($MailOrganization != "")
    $StdMIMEHeader .= "Organization: ".AddrNameEncode($MailOrganization)."₩n";
$StdMIMEHeader .= "MIME-Version: 1.0₩n";
# -> MIME 비표준 Header 작성
if ($MailPriority != "")
{
    $NonStdMIMEHeader = "X-Priority: ".$MailPriority."₩n";
    if ($MailPriority == "1")
        $NonStdMIMEHeader .= "X-MSMail-Priority: High₩n";
    else if ($MailPriority == "5")
        $NonStdMIMEHeader .= "X-MSMail-Priority: Low₩n";
    else
        $NonStdMIMEHeader .= "X-MSMail-Priority: Normal₩n";
}
else
{
    $NonStdMIMEHeader = "X-MSMail-Priority: 3₩n".
    $NonStdMIMEHeader .= "X-MSMail-Priority: Normal₩n";
}
$NonStdMIMEHeader .= "X-Mailer: PAWS21 WebMail Ver1.42₩n₩n";
```

[소스 4-8] 메일쓰기 MIME Header 작성 알고리즘

③ IMAP function을 활용하여 메일을 전송한다.

```
# -> 메일 전송
$MIMETotal = $StdMIMEHeader.$MIMEBody;
$MailSocket = popen("/usr/lib/sendmail -t -f '".$SessionUser."@".$SessionHost."'","w");
fputs($MailSocket,$MIMETotal);
pclose($MailSocket);
// 보낸편지함에 저장
if($MailSave == "save")
{
    $objMboxInfo = new MboxInfo;
    $objMboxInfo->stream = imap_open("₩{$IMAP4[HOST]:$IMAP4[PORT]}".$objMboxInfo->mbox, $SessionUser."@".$SessionHost, $SessionPass);
    imap_append($objMboxInfo->stream, "₩{$IMAP4[HOST]:$IMAP4[PORT]}".$MailBox[REAL][2], &$MIMETotal);
}
```

[소스 4-9] 메일쓰기 메일 전송 알고리즘

④ 사용자가 보낸편지함에 저장을 선택하였으면 보낸편지함에 보관한다.

⑤ 보낸편지함에 저장된 메일은 수신 확인 기능으로 확인한다.

4.11 외부 POP3 관리 기능 구현

다른 시스템의 메일을 POP3로 전송하여 가져오는 기능을 제공한다 사용자가 자신이 소유하고 있는 POP3메일 계정을 관리 기능에 설정하고 전송 받을 수 있는 기능이다. 이러한 외부 POP3기능은 여러개의 메일 계정을 하나의 시스템에서 관리할 수 있도록 편의를 제공하게 된다.

① 외부 POP3 계정으로 로그인한다.

② POP3를 이용하여 사용자의 디렉토리로 메일 전송

4.12 부재중 관리 기능 구현

사용자가 부재중 설정과 부재중 메모를 남기면, 메일 시스템은 부재중인 사용자에게 메일이 도착했을 경우 도착한 메일은 시스템에 저장하고 발신인에게 사용자의 계정으로 설정해둔 메모를 전달하는 기능이다.

예를 들어 사용자가 장시간 메일을 확인하지 못할 경우 부재중 설정을 해두면 중요한 사항일 경우 발신인에게도 도움을 주는 기능이다.

[소스 4-10]는 부재중 메일 전송 알고리즘이다.

```
sub send_reply() {
    if (-f "$message_file")
    {
        open(MSG, "$message_file");
        undef $/;
        $vacation_msg = <MSG>;
        close MSG;
    }
    else
    {
        exit(0) if ($no_msg_no_reply);
    }
    $vacation_msg =~ s/₩$SUBJECT/$subject/g;
    open(MAILPROG, "| $mailprog");
    print MAILPROG << "EOF" To: $sender Precedence: ju
    nk EOF"
    print MAILPROG $vacation_msg;
    close(MAILPROG);
}
```

[소스 4-10] 부재중 메일 전송 알고리즘

4.13 메일필터링 기능 구현

메일이 도착하면 메일 시스템이 자동으로 지정한 작업을 수행하게 하는 기능이다.

각각의 규칙에 따라 도착하는 메일을 분류 정리 할수 있는 기능으로 SPAM성 메일에 대응할 수 있도록 사용이 가능하다.

① 메일 시스템 로그인

② 필터링 여부 및 동작 확인

③ 각 동작별로 필터링 수행

4.14 스팸 관리 기능 구현

스팸으로 등록된 주소로부터 메일이 도착하면 메일 시스템이 수신 즉시 삭제해 버리는 기능이다.

메일 필터링 기능과 유사하지만 스팸으로 등록한 메일은 시스템에 저장되지 않고 바로 삭제되는 차이점이 있다. 즉 스팸으로 등록된 메일은 사용자가 확인할 수 없다.

① 메일 시스템 로그인

② 스팸 목록과 전송된 메일의 발송자 패턴 조사

③ 관련 스팸 메일 삭제

앞서도 이야기 했지만 스팸 관리와 필터링 관리는 약간의 차이점이 있다 메일의 발송자에 따라서 필터링을 한다는 것은 개개의 편지함으로 자동 분류하여 관리 한다는 기능이고 스팸 관리는 발송자이 따라서 메일을 계정 디렉토리에 저장하지 않고 바로 삭제하는 기능이다.

따라서 스팸 관리에 등록된 발송자의 메일을 스팸 목록에서 제거하지 않는 이상 사용자는 받아볼수 없게 된다.

4.15 서명 첨부 기능 구현

사용자가 메일을 보낼 때, 메일의 하부에 특정한 글이나 이미지를 삽입시킬 수 있도록 미리 정의할 수 있다. 이러한 기능을 이용하면 메일을 보낸 사람이 수신자가 알고 있는 사람인지를 확인하는데 도움을 주며 실제의 서명과 같은 효과를 낼 수 있다. 보안메일을 보낼 때의 방법중의 하나로 사용되기도 한다.

아주 간단한 알고리즘이지만 기대효과가 아주 큰 기능이다.

① 메일 발송시에 MIME 부분 밑에 저장

된 서명내용 전체를 첨가한다.

4.16 알림 메일 기능 구현

사용자가 설정해둔 특정한 날짜에 자동으로 시스템이 사용자에게 메일을 전송하여 기념일이나 주의사항들을 메일로서 알려주는 기능이다.

이러한 기능을 활용함으로서 기억하기 힘든 일정을 관리하여 미리미리 대처할수 있다.

① 사용자가 특정 날짜에 받을 메일을 예약 설정해 둔다.

② Crond 데몬이 매 분 체크하여 해당 시간에 발송해야할 메일 데이터 베이스를 체크한다.

③ 해당 시간에 발송할 메일이 존재하면 원본 그대로를 MIME으로 변환한다.

④ 변환된 메일을 발송한다.

4.17 편지함 관리 기능 구현

사용자가 필요에 의해 새로운 편지함을 생성, 수정, 삭제하여 자신의 메일을 관리할 수 있는 기능으로 본 시스템에서는 이 기능을 지원함으로서 기존의 메일 시스템과 같이 편지함으로 분류하여 자신의 메일을 관리할 수 있도록 지원한다.

[소스 4-11]은 편지함 관리의 편지함 목록 출력 부분 알고리즘이다.

```
$sql = "SELECT quota_size FROM auth WHERE mbox_host
='".$SessionHost."' AND id='".$SessionUser."'";
$rs = mysql_query($sql,$dbc);
$row = mysql_fetch_row($rs);
$intQuotaSize = $row[0]*1024;
for ($i=1; $i<=$MailBoxRows; $i++)
{
    imap_reopen($objMboxInfo->stream, imap_utf7_encode(
    "₩{$IMAP4[HOST]:$IMAP4[PORT]}".$MailBox[REAL][$i
    ]));
    $check = imap_mailboxmsginfo($objMboxInfo->stream)
    if($check->Size > 1024000)
    {
        $intSize = $check->Size/1024000;
        $intSize = number_format($intSize, 2) . "MB";
    }
    else
    {
        $intSize = $check->Size/1024;
        $intSize = number_format($intSize, 2) . "KB";
    }
    $intTotalUnseen += $check->Recent+$check->Unread;
    $intTotalAll += $check->Nmsgs;
    $intTotalSize += $check->Size;
    // 기본 편지함 목록 설정
    $strMailBoxList .= "<tr>";
    $strMailBoxList .= "    <td          class=₩"subject₩"><a
href=₩".          ./mailbox/MailList.html?NowMailBox=".$i."₩"
onMouseO                                        ver=₩"javascript:
StatusBarMsg('".$MailBox[VIEW][$i]."'          );return true;₩"
onMouseOut=₩"javascript:  StatusBarMs          g('');return
true;₩">".$MailBox[VIEW][$i]."</a></td>";
    $strMailBoxList .= "    <td class=₩"number₩"
align=₩"right ₩">".($check->Recent+$check->Unread)."</td>";
    $strMailBoxList .= "    <td class=₩"number₩"
align=₩"right ₩">".$check->Nmsgs."</td>";
    $strMailBoxList .= "    <td class=₩"number₩"
align=₩"right
    ₩">".$intSize."</td>";
    $strMailBoxList .= "    <tdalign=₩"center₩">";
}
if($intTotalSize > 1024000)
{
    $strTotalSize = $intTotalSize/1024000;
    $strTotalSize = number_format($strTotalSize, 2)."MB";
}
else
{
    $strTotalSize = $intTotalSize/1024;
    $strTotalSize = number_format($strTotalSize, 2)."KB";
}
```

[소스 4-11] 편지함 관리 알고리즘

4.18 주소록 관리 기능 구현

사용자는 주소록에 많은양의 전자우편 주소를 관리하고 메일 전송시에 정리된 주소록에서 받을 사람을 선택하여 메일을 전송할수 있고 전체 이름을 입력하지 않고 이름만을 입력하여 전체 메일 주소로 바뀌어 전송될수 있는 기능을 제공한다.

주소록에서는 그룹별로 메일 주소를 관리하며 검색 기능을 지원하여 보다 효율적인 메일 주소록 관리를 가능하게 한다.

4.19 파일 관리 기능 구현

메일에 첨부된 파일을 사용자의 파일함에 저

장하거나 사용자가 파일을 업로드하여 중요 파일을 관리하는 기능으로서 파일 관리 기능에 저장되어 있는 파일을 바로 첨부하여 메일로 전송할 수 있으면 첨부되어 수신된 메일의 파일을 사용자의 파일함에 저장하여 관리할 수 있다.

이러한 파일 관리 기능은 자신의 컴퓨터와 같은 환경을 제공하는게 아주 탁월하다.

5. 구현 결과 및 주요 기능 설명

5.1 메일 수신 기능

메일 수신 기능은 메일 함을 오픈하고 메일 목록의 메일을 선택하여 내용을 보는 일반적인 클라이언트의 인터페이스를 그대로 따라서 구현하였다.

각각의 메일의 MIME을 완벽히 분석하여 메일 읽기에 도움을 주었으며 일반적인 MIME 규칙을 따르지 않아서 본 시스템에서 분석하기 어려운 메일은 메일의 헤더,원본을 볼 수 있도록 구현하였으며, 메일 원본 자체를 다운로드 받을 수 있도록 구현하였다. 메일 원본을 다운로드 받아서 사용자의 컴퓨터에서 실행하면 아웃룩 익스프레스는 대부분의 MIME을 처리 할 수 있는 장점을 이용하는 것이다.

또한 읽은 메일과 읽지 않은 메일 삭제된 메일등을 형상화된 아이콘으로 처리 하여 구분되도록 구현하였다.

메일 번호, 보낸 사람, 제목, 보낸 날짜, 메일 크기 등으로 메일을 정렬하는 기능을 구현하여 메일의 관리를 편리하게 하였다.

또한 메일 전체 선택과 해제를 구현하여 보다 빠른 메일 관리를 돕고 있다.

[그림 5-1]는 메일 수신 기능의 기본 편지함의 스크린 샷으로 편지들의 목록을 출력해 주는 기능이 구현되어 있다.

[그림 5-1] 메일 수신 기능

[그림 5-2]는 메일 수신 기능의 메일 읽기의 스크린 샷으로 메일의 MIME을 분석하여 그 내용을 출력해 주고 답장, 전체 답장, 전달, 삭제, 원본 보기, 메일 다운로드, 메일 이동, 첨부파일 저장 또는 개인 파일함으로 이동 등의 기능이 구현되어 있다.

[그림 5-2] 메일 읽기 기능

5.2 메일 송신 기능

메일 송신 기능 크게 두 가지로 분류되는데 기본적인 메일 송신 기능과 파일을 첨부하거나 서명을 첨부하는 확장된 메일 송신 기능이 있다.

본 시스템에서는 파일 첨부를 위해서

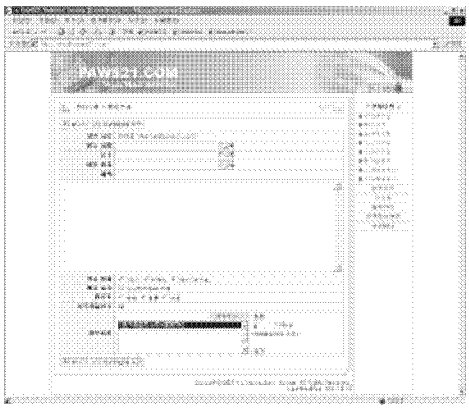
DataBase에 먼저 파일을 입력한 후에 파일 전송시에 MIME으로 생성하여 전송하는 방식을 사용하고 있다. 3M이상의 파일을 첨부 하기 위해서 시스템의 패킷 허용 용량을 3M로 확장하였다.

파일 첨부 루틴은 list 박스를 사용하여 첨부되는 파일을 실시간으로 확인 가능하도록 구현하였다.

TEXT, HTML, TEXT/HTML 의 세가지의 기본 메일 포맷을 지원하며 서명 파일을 첨부할 수 있도록 구현했으며 메일 전송시에 보낸편지함에 메일을 저장하여 메일의 수신 확인이 가능하도록 구현하였다.

메일 수신확인 알고리즘은 메일에 간단한 이미지 프로그램을 첨부하여 메일을 열어보는 동시에 시스템이 이를 감지하여 수신 여부를 체크하게 되어 있다.

[그림 5-3]은 메일 송신 기능의 메일 쓰기의 스크린 샷으로 위의 모든 기능이 구현되어 있다.



[그림 5-3] 메일 송신 기능

5.3 메일 검색 기능

메일 검색 기능은 메일 제목, 보낸 사람, 크기 등으로 검색 가능하도록 구현하였다.

메일 검색시에 검색어와 검색 대상을 입력하고 실행하면 대상 메일의 내용에서 패턴을 조사하여 해당 검색된 메일의 목록을 출력하게 되어 있다.

메일 검색 기능은 기본적으로 메일 수신 기능의 일부로 구현되어 있다.

[그림 5-4]는 메일 검색의 스크린 샷이다.

[그림 5-4] 메일 검색 기능

5.4 편지함 관리 기능

편지함 관리 기능은 본 시스템의 부가 기능인데 아웃룩 익스프레스등 기존의 메일 시스템의 클라이언트 프로그램들은 사용자의 컴퓨터에 편지함을 생성하고 메일을 저장하거나 관리 하는 기능을 가지고 있다.

본 시스템은 사용자의 메일 디렉토리에 편지함을 생성, 수정, 삭제 할 수 있는 기능을 두어 메일의 효과적인 관리를 돕는다.

[그림 5-5] 편지함 관리 기능

5.5 부재중 관리 기능

부재중 관리 기능은 사용자가 미리 설정해 두고 부재중일 때 수신된 메일에 대하여 메일을 저장하고 사용자의 계정으로 메일을 부재중을 메일로 알리는 기능이다.

[그림 5-6] 부재중 관리 기능

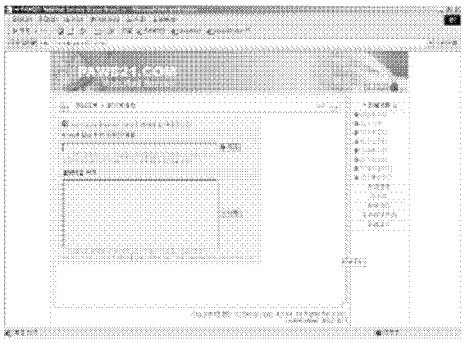
5.6 메일필터링 기능

메일 필터링 기능은 메일 수신시 자동으로 설정한 규칙에 맞추어 메일을 분류하는 기능이다.

[그림 5-7] 메일 필터링 기능

5.7 스팸 관리 기능

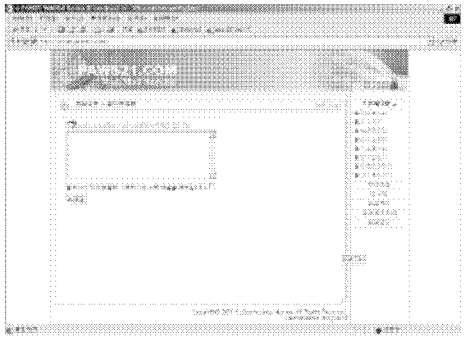
스팸 관리 기능은 메일 수신시에 스팸메일록 등록된 리스트와 메일 송신자를 비교하여 메일 디렉토리에 저장하기 않고 스팸으로 등록된 송신자가 보낸 메일은 바로 삭제하는 기능이다.



[그림 5-8] 스팸 관리 기능

5.8 서명 첨부 기능

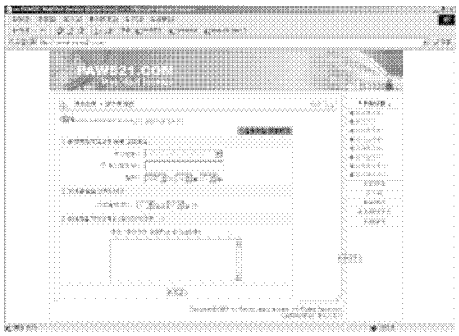
서명 파일 기능은 서명 파일을 설정하는 기능으로 여기서 설정한 서명 파일이 메일 발송 기능에서 서명 파일 첨부를 선택했을 때 첨부되어 보내지게 된다.



[그림 5-9] 서명 파일 기능

5.9 알림 메일 기능

알림 메일 기능은 미리 설정해둔 메일을 특정 날짜에 메일로 발송되는 기능이다.



[그림 5-10] 알림 메일 기능

5.10 외부 POP3 관리 기능

외부 POP3 관리 기능은 사용자가 POP3를 지원하는 시스템의 메일을 본 시스템으로 가져오는 기능을 한다. 본 시스템의 이 기능을 활용하면서 얻어지는 효과는 보통 여러 개의 메일 주소를 가지고 있는 사용자가 여러 곳의 메일을 관리할 필요 없이 본 시스템으로 모든 메일을 유도하여 본 시스템만의 관리로 모든 시스템으로 오는 메일을 관리 할 수 있도록 편의를 돕는 기능이다.

[그림 5-11] 외부 POP3 관리 기능

5.11 주소록 관리 기능

주소록 관리 기능은 본 시스템에서 주소록에 메일을 추가 하고 메일을 관리함으로써 보다 효율적으로 메일 시스템을 사용할 수 있도록 도와준다.

본 시스템에서의 메일 목록은 메일 쓰기에서 주소록으로의 연결을 통해서 따로 기억할 필요 없이 다량의 메일 주소를 관리 할 수 있도록 도와준다.

본 시스템의 메일 쓰기에서는 주소록에 저장되어 있는 메일 주소라면 메일을 수신할 대상의 Email주소를 입력 하지 않고 이름만을 입력함으로써 바로 메일을 전송할 수 있는 메일 주소 변환 기능을 제공한다.

[그림 5-12] 주소록 관리 기능

5.12 파일 관리 기능

본 시스템에서는 또 파일 관리 기능을 지원하는데 파일 관리 시스템은 사용자가 파일을 업로드하여 파일을 관리 할 수 있도록 도우며, 메일로 첨부된 파일을 개인의 파일 디렉토리로 저장하여 관리할 수 있으면 반대로 파일 디렉토리에 저장된 파일을 바로 첨부하여 메일로 대상에게 전송할 수 있다.

이러한 파일관리 기능은 따로 파일을 업로드하고 관리하는 기능을 본 시스템에 통합시키면서 보다 효율적인 메일을 관리할 수 있도록 편의를 돕는 기능이다.

[그림 5-13] 파일 관리 기능

6. 결론 및 향후 연구 과제

본 논문에서는 기존의 메일 시스템에서는 한계가 있는 기능들을 웹 기반으로 구현하였다. 서비스 엔진이라는 개념을 도입하여 사용자가 직접 SMTP서버로부터 메일을 가지고 오는 것이 아니라, 본 시스템에서 이를 자동으로 수행하게 된다. 시스템은 사용자의 전반적인 기능을 수행하게 되며, 직접 사용자의 메일을 사용자의 디렉토리로 저장하기 때문에 사용자가 급증 시 시스템에 과부하를 주던 기존의 메일 시스템의 문제를 파격적으로 제거하였다. 이러한 시스템의 변화는 적은 사양의 시스템에 보다 많은 사용자를 수용할 수 있는 부가가치가 아주 높은 시스템으로 발전하는데 크게 이바지하였다.

통계에 의하면 사용자들의 대부분은 비슷한 시간, 즉 아침 또는 저녁으로 메일함을 열어 보게 된다. 그러한 통계에 비춰볼 때 본 시스템은 자정과 정오에 쌓인 메일을 각 사용자의 디렉토리로 이동하며 각각의 사용자가 자신의 받은 편지함을 열어볼 때 사용자의 디렉토리로 메일을 이동한다. 이렇게 메일의 양을 분산하여 특정 시간대의 부하를 미연에 방지하게 된다.

본 메일 시스템은 일정한 시간을 간격으로 SMTP의 메일 디렉토리에서 각각 사용자의 메일 디렉토리로 메일을 받아온 후, 부재중처리, 잘못된 사용자의 처리, 메일필터링, 스팸관리 등과 같은 기능들을 수행하고, 수행한 결과를 보내거나, 사용자 디렉토리에 저장하게 된다.

본 메일 시스템은 이식성이 좋은 PHP Script와 C/C++로 구현하여 어떠한 시스템이라도 동작할 수 있다.

운영 서버로는 Unix 계열과 Window계열을 동시에 지원하여 보다 유연한 시스템을 보장한다.

메일 서버 구축을 위해 별도의 소프트웨어의 설치없이 아파치 웹서버와 PHP, MySQL을 설치 하여 본 시스템을 인스톨 하여 구축할 수 있다.

서비스 엔진은 메일 서버의 사용자 급증으로 인한 시스템 성능 저하 문제를 해결함과 동시에 사용자에게 다양하고 편리한 기능들을 제공한다.

향후 메일 도착시에 사용자의 이동전화로 메시지를 보내주는 SMS 연동기능, 메일에 첨부된 파일의 바이러스 검사 기능, 메일 작성시 한글 맞춤법 검사 기능들을 추가로 연구하여 할 계획이다.

8. 참고 문헌

[1] W. Richard Stevens, "TCP/IP Illustrated, Vol 1," Addison-Wesley Publishing Company, Inc, 1994.

[2] RFC 1460.

[3] RFC 2060.

[4] RFC 821.

[5] RFC 1521, 1522.

[6] T.Berners-Lee, R.Fielding and H.Neilsen, "Hyper text Transfer Protocol -- HTTP/1.0" RFC1945, 1996.

[7] Alex Homer and Chris Ullman, "INSTANT IE4 Dynamic HTML," Wrox Press Ltd, 1997

[8] "New And Emerging E-mail Applications," IEEE communications Magazinc, November 1999.

[9] 정진호, “SMTP를 이용한 하이퍼 메일 시스템의 설계 및 구현”, 한국과학기술원 석사학위논문, 1993.