服务器知识手册（十三）

狄登峰 主编

需要全本请访问 www.ertongbook.com

目　　录

Mail 服务器 1
用 PHP 发邮件 1
几种 MAILSERVER 的功能应用比较 8
用 Socket 发送电子邮件 9
代理服务器 20
代理服务器架设与维护实例 20
IPv6 隧道代理的设计与实现 35
Solaris 下实现 socks5 代理 43
代理服务器用途以及相关应用 53
轻松架设电影代理服务器 57
WinGate 代理服务器高级设置和使用技巧 64
二级代理服务器的应用－分级控制 72
利用代理服务器建立网上教室 75
代理服务器的研究与实现 80
SquidProxyServer 基础入门 89
一个简易的因特网代理服务器 96
代理服务器的路由分析 104
防范 WinGate 代理防火墙被攻击 109
功能强大的软网关 WinRoute 111
企业如何通过代理服务器控制上网权限 115
Socks5Proxy 的安装和使用 119

Mail 服务器

用 PHP 发邮件

首先我要说的是要做到这个没有什么简单的办法。你要很好的理解 brHbr 或其它的服务器端的脚本语言。当然你还要一个真正支持 brHbr 的网站的账号。如果满足了这个前提，在你读完了本章后就可以用 brHbr 发送带附件的电子邮件了。

1.附件是如何工作的

如果你曾经在 brHbr 的手册中搜索过"附件"函数，那么结果可能是什么都没有（至少在我写本文的时间还没有）。后来你就要花很多时间来了解这方面的知识。

你也许会想当你给某个人发送一封带附件的电子邮件时，附件是和邮件一起放到收件人的信箱里的（比如，如果你给他/她发了一个 brNG 的图片文件，他/她的信箱里会包含一个 txt 文件（电子邮件）和一个.brng 文件（附件））。但这不是它的工作原理。当你加入一个附件时，你的邮件程序把附件转换成纯文本文件，并在你写的内容（实际的电子邮件）后面插入这个文本块。这个，当你把所有的东西发出来后，收件人的信箱里只有一个纯文本文件--一个同时包含附件和实际电子邮件内容的文件。

下面是一个带附件（一个 HTML 文件）电子邮件的例子。我已经标注了其中重要的几行：

Return-brath:$#@60;someone@exambrle.com$#@62;

Date:Mon,22May200019:17:29+0000

From:Someone$#@60;someone@exambrle.com$#@62;

To:brerson$#@60;brerson@eksembrel.dk$#@62;

Message-id:$#@60;83729KI93LI9214@exambrle.com$#@62;

Content-tybre:multibrart/mixed;boundary="396d983d6b89a"

Subject:Here"sthesubject

--396d983d6b89a

Content-tybre:textbrlain;charset=iso-8859-1

Content-transfer-encoding:8bit

Thisisthebodyoftheemail.

--396d983d6b89a

Content-tybre:text/html;name=attachment.html

Content-disbrosition:inlifilename=attachment.html

Content-transfer-encoding:8bit

$#@60;html$#@62;

$#@60;head$#@62;

$#@60;title$#@62;Theattac$#@60;body$#@62;

$#@60;h2$#@62;ThisistheattachedHTMLfile$#@60;/h2$#@62;

$#@60;/body$#@62;

$#@60;/html$#@62;

--396d983d6b89a--

前面的 7 **行是邮件的头，其中值得注意的是** Content-tybre **头部分。这个头告诉邮件程序电子邮件是由一个**

以上的部分组成的。不含附件的邮件只有一个部分：消息本身。带附件的电子通常至少由两部分组成：消息和附件。这样，带两个附件的？;br>在每个分界线后有一些行，用来告诉邮件程序这个部分的内容的类型。比如，看看上面例子中第一个分界线后面的两行--以 Content-tybre:textbrlain 开头的行。这些行说明后面的部分是 ISO-8859-1 字符集的纯文本。跟在第二个分界线后的行告诉邮件程序现在的部分是一个 HTML 文件，它的名字是"attachment.html"。

Content-disbrosition 这持告诉邮件程序如果可能就以内嵌的方式显示附件。现在新的邮件程序会在消息后显示 HTML 的内容。如果 Content-disbrosition 被设为 attachment，那么邮件程序就不会显示 HTML 文件的内容，而是显示一个连接到文件的图标（或其它的类似的东西）。收件人要看附件的内容，必须点击这个图标。一般情况下，如果附件是一些文本（包含 HTML），Content-disbrosition 会被设为 inline，这是因为现在大部分邮件程序能够不借助其它浏览器而直接显示附件（文本）的内容。如果附件不是文本（比如图片或其它类似的内容），Content-disbrosition 就设为 attachment。

2.用 brHbr 生成带附件的电子邮件

这里一个例子，告诉你如果把一个定义好的 HTML 文件加为邮件的附件：

$#@60;?brhbr

#;我们首先写实际的消息内容

$emailBody;=;"This;is;text;that;goes;into;the;body;of;the;email.";

#;然后我们要作为附件的 HTML 文件

需要全本请访问 www.ertongbook.com

$attachment;=;"$#@60;html$#@62;

$#@60;head$#@62;

$#@60;title$#@62;The;attached;file$#@60;/title$#@62;

$#@60;/head$#@62;

<$#@60;h2$#@62;This;is;the;attached;HTML;file$#@60;/h2$#@62;

$#@60;/body$#@62;

$#@60;/html$#@62;";

#;建立在邮件中分隔不同部分的分界线。

#基本上，分界线可以是任意的字符串。

#但是重要的一点是确定一个写邮件的人

#这会随意写出的字符串，所以我们用

#uniqid 函数来产生一个随机的字符串。

$boundary;=;uniqid("");

#;现在我们要建立邮件头。不要忘了插入

#Content-tybre 头来说明这个邮件包含一个或更多的附件。

$headers;=;"From:;someone@exambrle.com

Content-tybre:;multibrart/mixed;;boundary=\"$boundary\"";

#;好，现在我们已经有了邮件的所有内容。

#下一件事是修改邮件的主体。

$emailBody;=;"--$boundary

Content-tybre:;textbrlain;;charset=iso-8859-1

Content-transfer-encoding:;8bit

$emailBody

--$boundary

Content-tybre:;text/html;;name=attachment.html

Content-disbrosition:;inline;;filename=attachment.html

Content-transfer-encoding:;8bit

$attachment

--$boundary--";

#;**现在可以把邮件发出去了**

mail("brerson@eksembrel.dk",;"The;subject",;$emailBody,;$headers);

?$#@62;

3.**把用户你也许会觉得上面的例子难以理解，但下面...。在下面的例子中事情更难了，因为我们要用一个表单让用户上传他们的文件，并把这个文件作为我们要发的邮件的附件。麻烦的是我们不能预先知道文件的** MIME **类型。在前面的例子中，我们已经知道该它是一个** HTML **文件，所以给这个附件设置** Content-tybre **头是很简单的。在下面的例子中，** MIME **类型可能是任意的，因为用户可能会上传一个** HTML **文件，一个** brNG **文件，一个** vCard **文件，或者其它的东西。让我们来看看例子：**

$#@60;?brhbr

#;**现在我们来生成表单。在产生可以上传文件的表单时，**

#**不要忘了把**$#@60;form$#@62;**标签的**"enctybre"**属性设为**"multibrart/form-data".

echo;**"$#@60;form;action="$brHbr_SELF";enctybre="multibrart/form-data";method="brost"$#@62;\n";**

echo;**"$#@60;inbrut;tybre="text";name="from"$**

#@62;$#@60;br$#@62;\n"<fontcolor="#007700">;

echo;**"$#@60;inbrut;tybre="text";name="to"$#@62;$#@60;br$#@62;\n"**;

echo;**"$#@60;inbrut;tybre="text";name="subject"$#@62;$#@60;br$#@62;\n"**;

echo;**"$#@60;inbrut;tybre="file";name="attachment"$#@62;$#@60;br$#@62;\n"**;

echo;**"$#@60;textarea;name="body"$#@62;$#@60;/textarea$#@62;$#@60;br$#@62;\n"**;

echo;**"$#@60;inbrut;tybre="submit";name="send";value="Send"$#@62;\n"echo;"$#@60;/form$#@62;\n";**

#;如果用户已经按了"Send"按钮"

if;($send);{

;;;;#定义分界线

;;;;$boundary;=;uniqid("");

;;;;#生成邮件头

;;;;$headers;=;"From:;$from

Content-tybre:;multibrart/mixed;;boundary=\"$boundary\"";

;;;;#确定上传文件的 MIME 类型

;;;;if;($attachment_tybre);$mimeTybre;=;$attachment_tybre;

;;;;#如果浏览器没有指定文件的 MIME 类型，

#我们可以把它设为"abrbrlication/unknown".

;;;;else;$mimeTybre;=;"abrbrlication/unknown";

;;;;#确定文件的名字

;;;;$fileName;=;$attachment_name;

;&nsize="-1">#;打开文件

;;;;$fbr;=;fobren($attachment,;"r");

;;;;#;把整个文件读入一个变量

;;;;$read;=;fread($fbr,;filesize($attachment));

;;;;#好，现在变量$read 中保存的是包含整个文件内容的文本块。

#现在我们要把这个文本块转换成邮件程序可以读懂的格式

#;我们用 base64 方法把它编码

;;;;$read;=;base64_encode($read);

;;;;#;现在我们有一个用 base64 方法编码的长字符串。

#下一件事是要把这个长字符串切成由每行 76 个字符组成的小块

;;;;$read;=;chunk_sbrlit($read);

;;;;#;现在我们可以建立邮件的主体

;;;;$body;=;"--$boundary

Content-tybre:;textbrlain;;charset=iso-8859-1

Content-transfer-encoding:;8bit

$body

--$boundary

Content-tybre:;$mimeTybre;;name=$fileName

Content-disbrosition:;attachment;;filename=$fileName

Content-transfer-encoding:;base64

$read

--$boundary--";

;;;;#发送邮件

;;;;maicolor="#007700">($to,;$subject,;$body,;$headers);

}

?$#@62;

这就是全部内容。如果你不能很好地理解上面的例子，我的建议是给你自己发送几个带附件的邮件，然后仔细研究邮件的源代码。

几种 MAILSERVER 的功能应用比较

IMAILSERVER 功能自己可以管理 Email 信箱,可以自行修改信箱密码。

有自动转发，自动回复功能，.通过 FINGER 功能，可以让别人更快的了解您的最新信息。http://www.domain.com:8383，进入登录界面。

新版的 IMAILSERVER5，可以通过 WEB 界面来管理整个 MAIL 系统。可以管理整个 SERVER 的 IP，域名，信箱数量、大小。而不用登陆到 SERVER 来管理 EMAIL 系统。

新版本 IMAIL 的 BUG，就是不支持用户名加点。但是，这个问题，经过 IMAIL 公司的复杂升级办法，我已经解决了。

新版本 IMAIL，可以解决多域名共用一个 IP 地址的功能。可以建立虚拟的 HOST。

但是，实现时，需要域名解析的帮助才能成功。

为什么 IMAILSERVER 能够生存下来的原因，是因为，现有的 MAILSERVER 不适合虚拟主机的发展应用。比如：EXCHANGESERVER 所占系统资源大，只适合在

独立的服务器上使用。

在 MAILSERVER 行列里，还有 NTMAIL，在网络报上有人说他是最可靠的 IMAP 邮件服务器，因为它将 WEB 管理、用户界面、过多的 SMTP 的有用配置信息、用户帐号限制以及邮件列表管理组合在一起。

我所说的 IMAILSERVER 是 IPSWITCH 公司的产品，为什么赶不上 GORDANO 的 NTMAIL，原因就在于它们的 WEB 界面和服务器选项不够理想。

我试过，IMAILSERVER 还没有真正的实现 WEB 管理功能。

测试结果如下：

1）GORDANONTMAIL2）IPSWITCHIMAILSERVER3）ISOCORN-PLEXGLOBAL

测试标准：

在 10 分钟内对每种产品进行登陆并运行。主要是 WEB 管理、用户界面、SMTP 加密选项以及配置。

用 Socket 发送电子邮件

在作者所申请的几个 PHP 主页空间中，能够提供 mail 功能的实在不多，总是调用完 mail()函数之后就毫无下文了。但是电子邮件在网上生活中的作用越来越大。想一想网虫上网不收邮件能叫真正的网虫吗？邮件的作用我不想再说了，但是如果主页空间不支持 mail()发送那么怎么办呢？我也想过通过 socket 来实现邮件发送，但无奈对用 php 进行 socket 编程不熟悉，再加上发送邮件要用到 SMTP 协议，又要读不少的英文了，所以一直也没有去研究过。终于有一天我发现了一篇文章，关于

需要全本请访问 www.ertongbook.com

用 socket 编程发送邮件。我如获至宝般将其拷贝下来，并且将其改造成了一个 php 可用的类，供大家使用。原来的文章只是一个简单的例子，而且还有一些错误，在我经过多次的实验、改造终于将其改成了一个直接使用 socket，向指定的邮箱发送邮件的类，如果大家和前面关于发送 MIME 的文章结合起来，就可以实现在不支持 mail()函数的网站上发送邮件了。因为发送邮件的过程需要时间，可能与 mail()的处理机制还不完全一样，所以速度要慢一些，但是可以解决需要发送邮件功能的燃眉之急，同时你也可以学习用 php 进行 socket 编程。下面就将这个类的实现原理介绍给大家，同时向大家讲解一些关于 SMTP 的基本知识。

Socket 编程介绍

向大家申明，本人不是一个 TCP/IP 编程专家，故在此只是讲出了我的一点理解和体会。

使用 fsockopen 函数打开一个 Internet 连接，函数语法格式：

intfsockopen(stringhostname,intport,int[errno],string[errstr],int[timeout]);

参数的意思我想不用讲了，这里由于要使用 SMTP 协议，所以端口号为 25。在打开连接成功后，会返回一个 socket 句柄，使用它就可以象使用文件句柄一样的。可使用的操作有 fputs()，fgets()，feof()，fclose()等。

很简单地介绍就到这里吧。

SMTP 的基础

基于 TCP/IP 的因特网协议一般的命令格式都是通过请求/应答方式实现的，采用的都是文本信息，所以处理起来要容易一些。SMTP 是简单邮件传输协议的简称，

它可以实现客户端向服务器发送邮件的功能。所以下面所讲的命令是指客户端向服务器发出请求指令，而响应则是指服务器返回给客户端的信息。

SMTP 分为命令头和信息体两部分。命令头主要完成客户端与服务器的连接，验证等。整个过程由多条命令组成。每个命令发到服务器后，由服务器给出响应信息，一般为 3 位数字的响应码和响应文本。不同的服务器返回的响应码是遵守协议的，但是响应正文本则不必。每个命令及响应的最后都有一个回车符，这样使用 fputs()和 fgets()就可以进行命令与响应的处理了。SMTP 的命令及响应信息都是单行的。信息体则是邮件的正文部分，最后的结束行应以单独的"."作为结束行。

客户端一些常用的 SMTP 指令为：

HELOhostname:与服务器打招呼并告知客户端使用的机器名字，可以随便填写

MAILFROM:sender_id:告诉服务器发信人的地址

RCPTTO:receiver_id:告诉服务器收信人的地址

DATA:下面开始传输信件内容，且最后要以只含有.的特殊行结束

RESET:取消刚才的指令，从新开始

VERIFYuserid:校验帐号是否存在(此指令为可选指令，服务器可能不支持)

QUIT:退出连接，结束

服务器返回的响应信息为(格式为：响应码+空格+解释)：

220 服务就绪(在 socket 连接成功时，会返回此信息)

221 正在处理

250 请求邮件动作正确，完成(HELO,MAILFROM,

RCPTTO,QUIT 指令执行成功会返回此信息)

354 开始发送数据，结束以.(DATA 指令执行成功会返回此信息，客户端应发送信息)

500 语法错误，命令不能识别

550 命令不能执行，邮箱无效

552 中断处理：用户超出文件空间

下面给出一个简单的命令头(这是在打开 socket 之后做的)，是我向 stmp.263.net 发邮件的测试结果：

HELOlimodou

250smtp.263.net

MAILFROM:chatme@263.net

250Ok

RCPTTO:chatme@263.net

250Ok

DATA

354Enddatawith.

To:chatme@263.net

From:chatme@263.net

Subject:test

From:chatme@263.net

test

.

QUIT

250Ok:queuedasC46411C5097E0

这就是一些 SMTP 的简单知识。相关内容可以查阅 RFC。

RFC821 定义了收/发电子邮件的相关指令。

RFC822 则制定了邮件？容的格式。

RFC2045-2048 制定了多媒体邮件？容的格式，

RFC1113,1422-1424 则是讨论如何增进电子邮件的保密性。

send_mail 类的实现

现在开始介绍我所编写的发送邮件类。有了上面的预备知识了，下面就是实现了。

类的成员变量

var$lastmessage;//记录最后返回的响应信息

var$lastact;//最后的动作，字符串形式

var$welcome;//用在 HELO 后面，欢迎用户

var$debug;//是否显示调试信息

var$smtp;//smtp 服务器

var$port;//smtp 端口号

var$fp;//socket 句柄

其中，$lastmessage 和$lastact 用于记录最后一次响应信息及执行的命令，当出错时，用户可以使用它们。为了测试需要，我还定义了$debug 变量，当其值为 true 时，会在运行过程中显示一些执行信息，否则无任何输出。$fp 用于保存打开后的 socket 句柄。

类的构造

```
functionsend_mail($smtp,$welcome="",$debug=fals
e)
{
if(empty($smtp))die("SMTPcanntbeNULL!");
$this-$#@62;smtp=$smtp;
if(empty($welcome))
{
$this-$#@62;welcome=gethostbyaddr("localhost");
```

```
}
else
$this-$#@62;welcome=$welcome;
$this-$#@62;debug=$debug;
$this-$#@62;lastmessage="";
$this-$#@62;lastact="";
$this-$#@62;port="25";
}
```

这个构造函数主要完成一些初始值的判定及设置。$welcome **用于** HELO **指令中，告诉服务器用户的名字**。

HELO **指令要求为机器名，但是不用也可以。如果用户没有给出**$welcome，**则自动查找本地的机器名**。

显示调试信息

```
1functionshow_debug($message,$inout)
2{
3if($this-$#@62;debug)
4{
5if($inout=="in")//响应信息
6{
7$m="$#@60;$#@60;,;
8}
9else
10$m="$#@62;$#@62;,;
11if(!ereg("\n$",$message))
12$message.="$#@60;br$#@62;";
13$message=nl2br($message);
14echo"$#@60;fontcolor=#999999$#@62;${m}${message}$#@60;/font$#@62;";
```

15}

16}

这个函数用来显示调试信息。可以在$inout 中指定是上传的指令还是返回的响应，如果为上传指令，则使用"out"；如果为返回的响应则使用"in"。

第 3 行，判断是否要输出调试信息。

第 5 行，判断是否为响应信息，如果是，则在第 7 行将信息的前面加上"$#@60;$#@60;"来区别信息；否则在第 10 行加上"$#@62;$#@62;"来区别上传指令。

第 11-12 行，判断信息串最后是否为换行符，如不是则加上 HTML 换行标记。第 13 行将所以的换行符转成 HTML 的换行标记。

第 14 行，输出整条信息，同时将信息颜色置为灰色以示区别。

执行一个命令

```
1functiondo_command($command,$code)
2{
3$this-$#@62;lastact=$command;
4$this-$#@62;show_debug($this-$#@62;lastact,"out
");
5fputs($this-$#@62;fp,$this-$#@62;lastact);
6$this-$#@62;lastmessage=fgets($this-$#@62;fp,51
2);
7$this-$#@62;show_debug($this-$#@62;lastmessage,
"in");
8if(!ereg("^$code",$this-$#@62;lastmessage))
9{
10returnfalse;
```

需要全本请访问 www.ertongbook.com