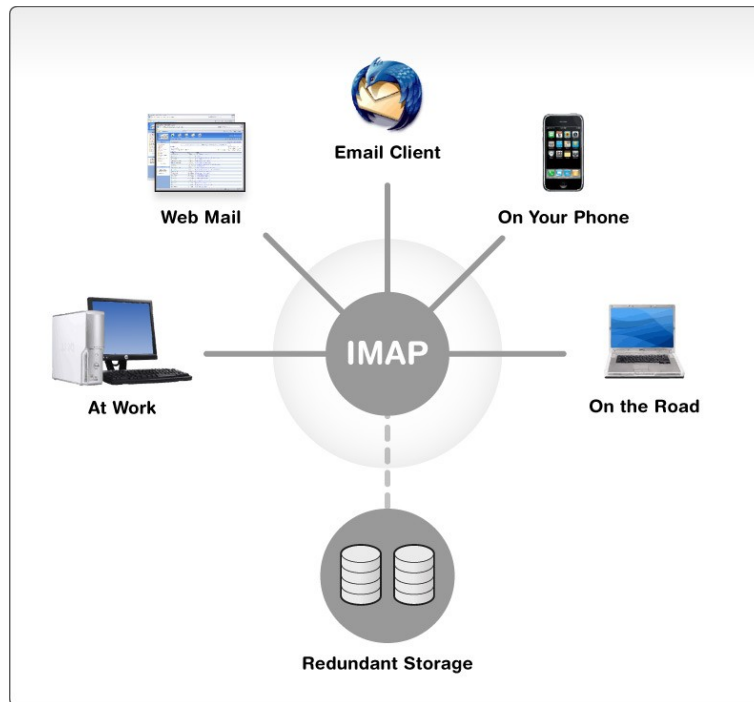


موضوع مقاله :

**پروتکل IMAP و نحوه‌ی استفاده از آن در سرویس‌دهندگی Courier
و گذری بر پروتکل SMTP در سرویس‌دهندگی Postfix
در سیستم عامل گنو/لینوکس**



ارائه دهنده :

بهرام بهرام‌بیگی

قابل دانلود از سایت :

دست نوشته‌های یک دانشجوی آی تی

<http://www.itnotes.info>

روزهای اولیه ایمیل :

پست الکترونیک یا Email یکی از راه‌های ارتباطی است که بیش از ۳ دهه است در زندگی روزمره‌ی انسان‌ها جای خودش را باز کرده است. ایمیل یکی از سرویس‌های غیرمتصل غیرقابل اعتماد در لایه‌ی کاربرد شبکه است. ایمیل تا سال ۱۹۹۰ فقط در سطح دانشگاه‌ها و مراکز علمی کاربرد داشت اما در همین سال (۱۹۹۰) وقتی ایمیل به صورت عمومی درآمد با چنان سرعتی رشد کرد که در طی یک دهه تعداد نامه‌های الکترونیکی از تعداد نامه‌های کاغذی بالاتر رفت. امروز Emailها به قدری از اطمینان و جذابیت رسیده‌اند که افراد عادی (و حتی در سطح بالای کشوری) برای نوشتن آن به جای نامه تردیدی به خود راه نمی‌دهند.

اولین سیستم ایمیلی که ساخته شد، تنها یک سیستم انتقال فایل (File Transfer) ساده بود. به طوری که در خط اول آدرس گیرنده‌ی نامه نوشته می‌شد. اما گذشت زمان ثابت کرد که چنین سیستمی مشکلاتی مانند مشکلات زیر را ایجاد می‌کند :

- ۱- فرستادن پیام به چند نفر مشکل بود. این اشکال بیشتر مدیران را آزار می‌داد. چون آن‌ها میل داشتند پیام‌های خود را به تمام افراد زیر دست خود بفرستند.
- ۲- پیام‌ها همچون ساختار داخلی نداشتند به همین دلیل پردازش کامپیوتری آن‌ها مشکل بود. برای مثال اگر پیامی از طرف یک شخص واسطه هدایت یا فرستاده می‌شد، استخراج قسمت هدایت شده مشکل بود.
- ۳- فرستنده‌ی نامه هرگز نمی‌توانست بداند که نامه‌اش به دست گیرنده رسیده است یا نه.
- ۴- اگر کسی می‌خواست که برای مدتی به مرخصی برود و در این مدت نامه‌های وارده به دست منشی‌اش برسند، کار ساده‌ای نبود.
- ۵- واسط کاربر (جایی که نامه را می‌نویسند) با قسمت ارسال نامه یکپارچه نبود. کاربر باید ابتدا نامه را می‌نوشت و برای ارسال آن باید برنامه‌ی ویرایشگر را ترک می‌کرد و به قسمت انتقال فایل می‌رفت.
- ۶- نامه‌ها فقط متن بود. ارسال تصویر، طرح، صدا و مانند آن‌ها ممکن نبود.

برای جلوگیری از این مشکلات نیاز بود تا نامه‌ها از فرمت خاصی پیروی کنند. که باعث شد در سال ۱۹۸۲ دو پروتکل RFC ۸۲۱ (برای انتقال) و فرمت RFC 822 (برای پیام) توسط آرپانت ارائه شود. سپس این پروتکل‌ها با تغییرات کوچکی به RFC 2821 و RFC 2822 به اینترنت راه یافتند. که در واقع پروتکل RFC 2821 همان پروتکل SMTP است. (Simple Mail Transfer Protocol) که در ادامه به توضیح آن خواهیم پرداخت.

توضیحی در مورد RFC :

درخواست برای توضیحات (Request for Comments) در واقع یک نامه‌ی غیر رسمی از طرف سازمان Internet Engineering Task Force یا IETF که مسئول ایجاد و توضیح استانداردهای اینترنت است. این سازمان پروتکل‌ها را از طریق RFCها استاندارد سازی کرده و آن را به طور کامل توضیح می‌دهد. همچنین نحوه‌ی استفاده از آن‌ها را به طور کامل تشریح می‌کند. این سازمان نیازی به عضویت رسمی ندارد که البته اسپانسرهای بزرگی همچون VeriSign را در پشت خود می‌بیند. هر استاندارد RFC به وسیله‌ی یک عدد یکتا پس از آن می‌آید که آن را به صورت یکتا مشخص می‌کند. این استانداردها می‌توانند پروتکل و یا فرمت خاصی باشند.

هر RFC می‌تواند دارای وضعیتی (Status) باشد که نحوه‌ی تکامل آن را مشخص می‌کند. این وضعیت‌ها می‌توانند به صورت زیر باشند.

Historic یا *Informational, Experimental, Best Current Practice (BCP), Standards Track* (برای توضیح هر کدام به صفحه‌ی ویکیپدیای *Request_for_Comments* در زبان انگلیسی مراجعه کنید) در این مقاله RFC 2822 یا SMTP و RFC 3501 یا IMAP بررسی خواهند شد.

ورود به دنیای عملی Email

هدف اصلی در این مقاله کار عملی با پروتکل‌های مطرح شده است. بنابراین ابتدا شباهت‌های ایمیل با دنیای واقعی را بررسی می‌کنیم :

سیستم ایمیل چیزی جز ایمیل در دنیای واقعی ما نیست. هنگامی که می‌خواهید یک نامه را بفرستید ابتدا آن را روی کاغذ می‌نویسید ، سپس آن را درون یک پاکت نامه می‌گذارید و آدرس گیرنده و سایر مشخصات را روی آن می‌نویسید و سپس آن را داخل صندوق پست می‌اندازید و یا اینکه آن را پست سفارشی می‌کنید (البته در دنیای مجازی خبری از این‌ها نیست ! همگی سفارشی هستند) سپس اداره‌ی پست با توجه به آدرس گیرنده باید تصمیم بگیرد که نامه را به کجا بفرستد. اگر نامه‌ی شما محلی باشد (یعنی در همان شهر محل زندگی) آن را مستقیماً تحویل مامور پست داده و به در خانه‌ی گیرنده می‌برد. در غیر اینصورت نامه باید تحویل یک اداره‌ی پست دیگر در یک شهر (یا کشور) دیگر شود. پس در این حالت اداره‌ی پست گیرنده وظیفه‌ی رساندن نامه به گیرنده را خواهد داشت.

حال بر می‌گردیم به دنیای مجازی. همه چیز شبیه سازی شده است. تنها چیزی که شما نیاز دارید آدرس گیرنده است. شما دارای یک صندوق پستی هستید که کلید آن فقط در دست شماست (و طبعاً نباید دست افراد ناشناس بدهید). شما در دنیای واقعی نیاز دارید که یک صندوق پستی بخرید ولی در دنیای مجازی سرویس‌دهنده‌های بسیار زیاد رایگان وجود دارند که به شما ایمیل مجانی می‌دهند (مانند Gmail ، Yahoo ، و یا حتی Hotmail) این سرویس‌دهنده‌ها از پروتکل‌های ایمیل به طور کامل پشتیبانی می‌کنند.

شما هنگامی که می‌خواهید ایمیل ارسال کنید. کامپیوتر شما (و یا سرویس‌دهنده‌ی شما) نیاز به زبانی مشترک دارند تا با سرویس‌دهنده‌ی مقصد حرف بزنند. زبان مورد نیاز برای ارسال ایمیل همان SMTP است. همچنین هنگامی که می‌خواهید ایمیلی را دریافت کنید ، به زبان مشترکی نیاز دارید که این زبان می‌تواند POP و یا IMAP باشد.

پشت صحنه‌ی Email

سیستم ایمیل اینترنتی اصولاً ترکیبی از یک سیستم سرویس‌گیرنده-به-سرویس‌گیرنده و سرویس‌دهنده-به-سرویس‌دهنده به اضافه تعدادی عامل در پشت صحنه است. شما یک پیام ایمیل را در سرویس‌گیرنده‌ای مانند Mozilla Thunderbird و یا Microsoft Outlook می‌نویسید. پستی آن را به اداره‌ی پست مشخص شده می‌رساند که همان سرویس‌دهنده‌ی پست برگزیده‌ی شما است. سرویس‌دهنده ممکن است مجبور باشد با سرویس‌دهنده‌ی پست دیگری صحبت کند که این در صورتی است که گیرنده در شبکه‌ای که سرویس‌دهنده‌ی پستی مسئول آن است ، غیرقابل دسترسی باشد. در زمان دریافت پیام ، سرویس‌دهنده‌ی پستی در انتهای دریافت از سرویس‌دهنده‌ی مقصد می‌خواهد تا آن را در صندوق پستی گیرنده قرار دهد. این در واقع سرویس‌گیرنده‌ی ایمیل گیرنده است که پیام را از صندوق پستی خود در سرویس‌دهنده‌ی محلی خود می‌خواند.

برنامه‌ی سرویس‌گیرنده‌ی ایمیل شما Mail User Agent یا MUA نام دارد. و سرویس‌دهنده‌ی پستی Mail Transfer Agent یا MTA نام دارد. یک MUA نه تنها پیام‌های ایمیل را تحویل می‌دهد بلکه آن‌ها دریافت هم می‌کند.

همان‌طور که متذکر شدیم ، این برنامه‌ی MTA نیست که پیام‌های ایمیل را درون صندوق پستی گیرنده قرار می‌دهد بلکه برنامه‌ی کوچک دیگری به نام Mail Delivery Agent یا MDA مسئول انجام این کار است. پس همان‌طور که می‌بینیم عوامل زیادی در پشت صحنه برای ارسال و دریافت ایمیل وجود دارند.

فرمت پیام‌ها در استاندارد RFC 8222

یک پیام ایمیل چیزی به جز یک فایل متنی ساده نیست. هیچ اهمیتی ندارد که محتویات آن چیست. ایمیل یک سری از خطوط متنی است که جفت کاراکترهای CR/LF پایان یافته اند. نکته : یونیکس و سیستم‌های عامل مبتنی بر آن از کاراکتر \n برای پایان دادن به یک خط متن استفاده می‌کنند ولی در ویندوز از دو کاراکتر \n\r (یعنی برگشت به اول خط و رفتن به خط جدید) استفاده می‌کند. چون در این مقاله به بررسی IMAP در لینوکس می‌پردازیم لذا نیازی به \r نیست ولی برای اطمینان بیشتر توصیه می‌شود که از آن استفاده کنید.

اجازه بدهید یک نامه‌ی ساده را در فرمت ۸۲۲ ببینیم :

```
Return-Path: <test@about.com>
received: from about.com (IDENT:test@about.com[192.168.0.21])
  by mail.somewhere.com(8.9.3/8.9.3) with SMTP id WAA29446
  for bahram; Tue, 30 Dec 2009 1:17:09+0900
Date: Tue, 30 Dec 2009 10:17:09+0900
From: Bahram <test@about.com>
To: matin@trilexnet.com
Message-Id:<F894376DE411@about.com>
Subject: Just a Note
```

Hi Mat

What are you doing ?

همان‌طور که می‌بینید یک نامه شامل تعدادی فیلد و سپس مقادیر آن‌ها با یک کاراکتر دونقطه (:) از هم جدا شده‌اند(فیلدهای سرآیند) . که متن نامه با یک خط خالی (\n) از بقیه‌ی فیلدهای سرآیند جدا شده است. دو فیلد هدر اول که دو پیام ایمیل فوق ملاحظه می‌کنید عبارتند از : Return-Path و received که آن‌ها را Envelope Information می‌نامیم و مشخص کننده‌ی آدرسی است که در صورت برگشت داده شدن پیام باید مجدداً به آن برگردد. همچنین اطلاعاتی در مورد سرور ارسال کننده می‌دهد. حال به توضیح سرآیندها می‌پردازیم :

Return-Path

فیلد Return-Path توسط آخرین MTA ی دخیل در ارسال پیام نوشته شده است. و راه برگشت به فرستنده‌ی پیام را نشان می‌دهد. در صورتی که پیام به هر دلیلی برگشت داده شود ، به این فیلد هدر اشاره خواهد شد.

Received

فیلد سرآیند Received بوسیله‌ی هر سرویس‌دهنده‌ی SMTP به ابتدای همه‌ی پیام‌های وارد شونده اضافه می‌شود تا نشان دهد پیام از چند MTA عبور کرده است تا در نهایت به مقصد برسد (از آن به عنوان hops هم نام می‌برند.) اکثر سرویس‌دهنده‌های SMTP پیام‌هایی با بیش از ۲۵ فیلد Received را پس می‌زنند. Hop counting متدی برای جلوگیری از loop ها است و یک پیام در صورتی که از سرویس‌دهنده‌های SMTP بسیار زیادی عبور کند در نهایت پس

زده می‌شود و به آدرس مشخص شده در فیلد سرآیند Return-Path برگردانده می‌شود.

Message-Id

فیلد سرآیند Message-Id که توسط میزبان تولیدکننده‌ی پیام ایجاد شده است، یک شناسه‌ی پیام ایمیل را که باید در عرصه‌ی جهانی منحصر بفرد باشد را در خود نگه می‌دارد. اگرچه نام domain به کار رفته در فیلد باید از نوع میزبانی باشد که در آن ایجاد شده است، بیشتر میزبان‌ها برای خود طرح اختصاصی دارند که برای تولید Id ها در صورت فقدان نام domain یا دلایل امنیتی به کار می‌رود.

سرآیندهای اجباری

فیلدهای سرآیند برای هر پیام ایمیل لازم هستند. در حالی که بقیه اختیاری هستند. فیلدهای هدر زیر قابل حذف نیستند. نظیر Date, From یا یکی از Bcc یا To. اگر To وجود نداشته باشد، Bcc باید حاضر باشد و برعکس

Date

سرآیند Date یک TimeStamp است که نشان‌دهنده‌ی لحظه‌ای است که فرستنده دکمه‌ی Send برنامه‌ی MUA را برای صدور مجوز ارسال پیام فشار می‌دهد. که شامل موارد زیر است:

- زمان محلی MTA که کاربر بکار می‌برد
- ناحیه‌ی زمانی MTA
- زمان واقعی در Universal Time

برای مثال Timestamp زیر:

Tue, 30 Dec 2009 10:17:09+0900

موارد زیر را نشان می‌دهد:

- ساعت محلی MTA عبارتند از: ۱۰:۱۷:۰۹
- ناحیه‌ی زمانی آن +۰۹۰۰ است که ساعت ۹ در شرق UTC است. (Universal Time Coordinated)
- ساعت واقعی ۱۰:۱۷:۰۹ روز ۳۰ دسامبر سال ۲۰۰۹ است. که به صورت ساعت واقعی منهای ۹ بدست می‌آید.

From

آدرس ایمیل فرستنده در فیلد From قرار گرفته است. این فیلد ممکن است حاوی ۱ یا بیش از یک ایمیل باشد که با کاما (,) از هم جدا شده‌اند (اما معمولاً این فیلد تکی است) برای اینکه نام مستعاری را برای یک آدرس ایمیل مشخص کنید می‌توانید ابتدا نام مستعار را آورده و سپس آدرس ایمیل را در دو کاراکتر (<>) قرار دهید. برای مثال:

- bahram <bahram@test.com>
- ali ahmadi <ali@yahoo.com>
- "Reza Asadi" <reza@myowndomain.com>

که اغلب برنامه‌های MUA از این فرمت پشتیبانی می‌کنند و به‌جای آدرس ایمیل نام مستعار را نشان می‌دهند. در واقع برای ساختن کتابچه‌ی آدرس از این فرمت استفاده می‌شود.

To

فیلد To حاوی هویت گیرندگان اولیه‌ی پیام است. این فیلد تنها و تنها زمانی می‌تواند حذف شود که فیلد Bcc مشخص

شده باشد. چندین گیرنده توسط کاما از هم جدا می‌شوند. (اگر عادت کرده‌اید از سیمی‌کالون (;) استفاده کنید ، این عادت خود را ترک کنید !!)

Bcc

فیلد سرآیند Bcc تنها و تنها زمانی می‌تواند حذف شود که فیلد سرآیند To حاضر باشد. Bcc به معنای Blind Carbon Copy است که زمانی استفاده می‌شود که نخواهید گیرندگان نامه بدانند که نامه‌ی شما به چه کسان دیگری ارسال شده است. یعنی در فیلد Bcc تنها آدرس خودشان قرار داده می‌شود.

هدرهای اختیاری

فیلدهای سرآیند ذکر شده تا کنون ، در واقع حداقلی برای ارسال ایمیل هستند که در صورت نبود ، MTA شما پیغام error خواهد داد. بقیه‌ی سرآیندهایی که در زیر می‌آیند اختیاری هستند می‌توانند در سرآیند ذکر نشوند.

Reply-To

آدرسی را مشخص می‌کند که پاسخ‌ها باید به آن ارسال شوند. برای مثال اگر روی دکمه‌ی Reply سرویس‌گیرنده‌ی ایمیل خود کلیک می‌کنید ، این سرآیند به کار برده می‌شود. در صورت غیبت آن ، تمامی پاسخ‌ها با آدرس مشخص شده در فیلد From ارسال می‌شوند.

Cc

فیلد سرآیند Cc مخفف Carbon Copy دقیقاً مشابه Bcc است به استثناء اینکه در بلاک سرآیند ظاهر می‌گردد و تمامی گیرنده‌های درون سرآیند خواهند فهمید که چه کسان دیگری یک کپی از این پیام ایمیل را دریافت کرده‌اند.

Subject

همان‌طور که از نامش پیداست برای مشخص کردن یک عنوان برای نامه است. اگرچه این فیلد به عنوان یک فیلد اختیاری در سرآیند ذکر شده است اما بسته به برنامه‌ی سرویس‌گیرنده شما ممکن است آن را درخواست کند و یا شما را با پیغام خطا مواجه کند.

فیلدهای سرآیند سفارشی (Custom)

برنامه‌های کاربردی که با آن ایمیل‌ها را ارسال و یا دریافت می‌کنید (MUA) ممکن است سرآیندهای اختصاصی خودشان را به ایمیل اضافه کنند. فیلدهای سرآیند سفارشی باید با یک پیشوند «X-» همراه شوند تا آن‌ها از سرآیندهای استاندارد متمایز کنند. برای مثال

X-Mailer: Thunderbird 2.0 Build 12

این سرآیند تنها برای برنامه‌ی ارسال کننده قابل درک خواهد بود. که در صورتی که برنامه گیرنده هم از Thunderbird استفاده کند می‌تواند آن را بخواند و احیاناً برای سازگاری از آن استفاده کند (که مشخص کننده‌ی نسخه‌ی آن خواهد بود)

MIME - راهی برای جهانی سازی

با استفاده از RFC 8222 تنها می‌توانید نامه‌هایی بفرستید که در آن از کاراکترهای ASCII استفاده شده باشد. در حالی که می‌دانیم امروزه برای دیگر زبان‌ها (مانند فارسی) نیاز به کدینگ Utf داریم. همچنین پرواضح است که برای فرستادن

صدا و تصویر کدینگ ASCII جواب نمی‌دهد. (بلکه بایستی از کدینگ base64 استفاده کرد). بنابراین RFC 1341 یا همان MIME بوجود آمد که در واقع گسترش یافته‌ی RFC 822 است. در این فرمت به همان فرمت قبلی ۸۲۲ پنج سرآیند جدید اضافه می‌شود که عبارتند از :

MIME-Version , Content-Description, Content-Id, Content-Transfer-Encoding, Content-Type
 که سرآیند MIME-Version تعیین می‌کند پیام ما دارای کدینگی غیر از ASCII است و نبود آن باعث می‌شود MUA آن را به صورت متن عادی تلقی کند.

سرآیند Content-Description توضیحی در مورد محتوای پیام است. این سرآیند به گیرنده می‌گوید که آیا پیام ارزش خواندن را دارد یا خیر. برای مثال در سرآیند ذکر می‌شود که « این یک پیام از طرف رئیس است » و کارمند باید در اولویت اول آن را باز کند

سرآیند Content-Id مشابه سرآیند Message-Id است (یک شناسه‌ی یکتا در عرصه‌ی جهانی)

سرآیند Content-Transfer-Encoding که می‌تواند یکی از انواع ASCII یا base64 (برای فایل‌های باینری ، عکس و فیلم) و یا quoted-printable باشد.

سرآیند Content-Type مشخص کننده‌ی نوع داده‌ی ارسالی است. برای مثال اگر شما یک عکس با فرمت jpeg ارسال کنید این سرآیند بدین شکل خواهد بود.

Content-Type: image/jpeg

پس برنامه‌ی گیرنده می‌داند که این یک عکس است و باید آن را در برنامه‌ی دکودر (Decoder) عکس باز کند.

در صورتی که بخواهید متن را به همراه یک عکس و یا فیلم بفرستید بایستی از نوع multipart و نوع چندگانه (برای مثال alternative استفاده کنید. مانند :

Content-Type: multipart/alternative; boundary=qwertyuiopasdfghjkl

در جایی از نامه که می‌خواهید عکس و یا فیلم خود را بگذارید بایستی با این مرز شروع کنید :

--qwertyuiopasdfghjkl

مقادیر دیگری که می‌توانند به جای alternative قرار بگیرند ، parallel ، Mixed و digest هستند. که برای توضیح هر کدام به منابع مقاله مراجعه کنید.

پروتکل Telnet

این پروتکل در RFC 854 تعریف و تشریح شده است. این پروتکل کاربر را قادر می‌سازد که با استفاده از یک ترمینال ، از هر مکانی و با استفاده از یک خط ارتباطی همانند خط تلفن با یک سیستم راه دور ارتباط برقرار کرده و وارد سیستم شده ، از آن سرویس بگیرد. با این قابلیت ، تفاوتی بین یک کاربر که در کنار کامپیوتر مرکزی نشسته و یک کاربر راه دور نخواهد بود.

بنابراین در یک تعریف ساده ، برنامه‌ی telnet موظف است بر روی ماشین کاربر ، مشخصه‌های ترمینال حقیقی سرویس‌دهنده را شبیه‌سازی نماید. این ترمینال شبیه‌سازی شده را NVT می‌نامند.

از مهم‌ترین وظایف پروتکل telnet این است که باید خود را با ترمینال‌های متفاوت تطبیق دهد. به عنوان مثال فرض کنید که یک کاربر از کامپیوتری با کدهای ASCII استفاده می‌کند (یک سیستم PC) در حالی که تمایل دارد به سیستمی که استانداردهای آن کدهای EBCDIC است (یک کامپیوتر Mainframe) متصل شود.

سرویس‌دهنده‌ی telnet بر روی کامپیوتر میزبان نصب می‌شود و سرویس‌گیرنده‌ی telnet نیز بر روی کامپیوتر میهمان و در صورتی که کامپیوتر میزبان اجازه‌ی ورود را به میهمان بدهد ، می‌تواند از ترمینال NVT استفاده کند.

برای اتصال به یک کامپیوتر راه دور از طریق telnet در شل باید دستور زیر را وارد کنید (فرض می‌کنیم کامپیوتر راه دور ۱۹۲.۱۶۸.۰۳ است)

```
telnet 192.168.0.3 80
```

توجه کنید که telnet به صورت پیشفرض به پورت ۲۳ یعنی SMTP اتصال برقرار می‌کند در غیر اینصورت باید پورت سوکت موردنظر را بعد از نام کامپیوتر مقصد بنویسید.

در این مقاله از ترمینال telnet به عنوان MUA استفاده می‌کنیم تا کدهای واقعی پشت‌صحنه‌ی پروتکل‌های ایمیل را احساس کنیم.

پروتکل SMTP

هنگامی که یک MUA یک پیام ایمیل را به یک MTA می‌رساند ، آن‌ها با یک زبان مشترک به نام Simple Mail Transfer Protocol صحبت می‌کنند. همان‌طور که شاید از نام آن فهمیده باشید ، درک SMTP بسیار ساده است. SMTP یک پروتکل line-oriented است. هم سرویس‌گیرنده و هم سرویس‌دهنده فرامین و پاسخ‌ها را در قالب رشته‌های کاراکتری پایان یافته با یک جفت «CR/LF» یا carriage return /line feed انتقال می‌دهند. پروتکل‌های خطی به سادگی قابل یادگیری و انتقال هستند. برای پردازش آن هم کافی است حرف‌هایی که با فاصله از هم جدا شده‌اند را هم جدا و در یک صافی عبارات باقاعده (Regular expression) قرار دهید.

کار با SMTP در Telnet و نصب Postfix

حال بیایید وارد دنیای متنی SMTP با استفاده از telnet شویم. برای این کار باید یک سرویس‌دهنده‌ی SMTP داشته باشید. که ما می‌خواهیم آن را بر سرور گنو/لینوکس خود نصب کنیم. مشهورترین سرویس‌دهنده‌ی SMTP همان Postfix است. که در آدرس رسمی www.postfix.org می‌توانید آن را دانلود کنید و یا اسناد راهنمای آن را ببینید. این سرویس‌دهنده هم‌اکنون بر روی بسیاری از سیستم‌عامل‌ها پورت شده است. اما هدف ما در این مقاله سیستم‌عامل گنو/لینوکس است. اگر از توزیع دبیان استفاده می‌کنید کافی است که دستور زیر را بزنید :

```
sudo apt-get install postfix
```

و به راحتی این سرویس‌دهنده نصب می‌شود. در غیر این صورت در صفحه‌ی دانلود از سایت رسمی ، کد منبع آن را دانلود و از حالت فشرده خارج سازید(ابتدا با cd وارد پوشه‌ی مربوطه شوید و به جای ver نسخه‌ی مربوطه را وارد کنید) :

```
tar xzf postfix-ver.tar.gz
```

و سپس وارد پوشه‌ی extract شده ، شوید و آن را با دستورات زیر کامپایل کنید :

```
./configure && make && make install
```

به همین راحتی postfix به عنوان یک دایمن (Daemon) بر روی سرور شما نصب می‌شود و تقریباً آماده‌ی کار است. شما می‌توانید تنظیمات پیشفرض را برای سرور خود در آدرس زیر انجام دهید.

```
/etc/postfix/main.cf
```

که توضیحات مربوطه را در documentation سایت رسمی می‌توانید پیدا کنید.

در صورتی که postfix شما راه اندازی نشده باشد ، می‌توانید با دستور زیر آن را راه‌اندازی کنید (از ورود با اشتراک root اطمینان حاصل کنید)

```
/etc/init.d/postfix start
```

حال آماده‌ایم تا از سرویس‌دهنده‌ی SMTP خود استفاده کنیم. توجه کنید در جلسه‌های telnet زیر کامپیوترهای میزبان و مهمان یکی هستند.

ترمینال را در سیستم‌عامل گنو/لینوکس خود باز کنید و دستور زیر را تایپ کنید:

```
telnet localhost 25
```

پروتکل SMTP در پورت ۲۵ آماده‌ی خدمت است. در صورتی که postfix را به درستی نصب کرده باشید، پیغام زیر را مشاهده خواهید کرد (سیستم‌عامل تست شده Ubuntu است):

```
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
220 localhost ESMTP Postfix (Ubuntu)
```

حال آماده‌ایم تا درخواست خود را به postfix بفرستیم. برای شروع بایستی عرض ادب به سرویس‌دهنده بکنید با دستور زیر:

```
HELO localhost
```

و جواب می‌گیرید که:

```
250 localhost
```

کد ۲۵۰ نشان‌دهنده‌ی موفقیت آمیز بودن دستور وارد شده و عملیات می‌باشد. در صورتی که عددی غیر از ۲۵۰ بگیرد بایستی سعی کنید خطایی که در دستور رخ داده است برطرف کنید.

حال بایستی فرستنده‌ی نامه را مشخص کنید (بهتر است کاربری با این مشخصات در سیستم موجود باشد)

```
MAIL FROM: bahramwhh@localhost
```

و جواب آن، که با کد موفقیت آمیز ۲۵۰ مشخص شده است

```
250 2.1.0 Ok
```

حال نوبت گیرنده‌ی نامه است که با دستور زیر:

```
RCPT TO: root@localhost
```

و پاسخ آن:

```
250 2.1.5 Ok
```

حال نوبت نوشتن متن نامه است (متن نامه باید با نقطه تمام شود و نوشتن Subject همان‌طور که گفته شد اختیاری است):

```
DATA
354 End data with <CR><LF>.<CR><LF>
Hello Root
Please Check my account
it seems to be blocked
```

```
thanks
bahramwhh
.
```

توجه کنید که خط بعد از DATA مربوط به postfix است که توضیح داده‌است انتهای نامه باید با نقطه پایان یابد. خروجی به صورت زیر است:

```
250 2.0.0 Ok: queued as CCB1F40256
```

در صورتی پیغام با تگ ۲۵۰ گرفتید، موفق شده‌اید. حال شما یک نامه به root@localhost فرستاده‌اید که به محض خواندن صندوق پستی‌اش آن را خواهد دید. برای خروج هم کافی است که QUIT را بزنید.

```
QUIT
221 2.0.0 Bye
```

Connection closed by foreign host.

روش کار برای یک سرویس دهنده‌ی راه دور هم به همین صورت است فقط در هنگام اتصال به جای localhost کافی است سرویس دهنده‌ی راه دور خود را وارد کنید .

نکته امنیتی : اکثر سرویس دهنده‌های Postfix طوری تنظیم می‌شوند که امکان اتصال Remote وجود نداشته باشد. در غیر اینصورت افراد شرور به راحتی می‌توانند از طریق سرور شما ایمیل‌های جعلی بفرستند. برای این کار بایستی postfix خود را از طریق دستور shell به صورت postfix تنظیم کنید (محیط متنی و گرافیکی است)

پروتکل‌های POP و IMAP چرا دو پروتکل؟!

پروتکل SMTP از ابتدا تا کنون بسیار دقیق طراحی شده و مشکلی نداشته است. اما در همان روزهای ابداع SMTP ، پروتکلی لازم بود تا نامه‌ها را دریافت کند. این پروتکل POP نام گرفت که هم‌اکنون در نسخه‌ی ۳ خود (یعنی POP3) قرار دارد. پروتکل Post Office Protocol در RFC 1939 به طور کامل تشریح شده است. این پروتکل برای آن روزها مناسب بود. چرا که سیستم‌های دیجیتال موبایل (متحرک) به شکل امروزی متداول نشده بود و همچنین هر فرد بیش از یک کامپیوتر نداشت. اما امروزه افراد تمایل دارند علاوه بر آنکه در خانه ایمیل‌هایشان را چک می‌کنند در حال رفتن به سر کار (با موبایل یا PDA) و در کامپیوتر سر کار آن‌ها را مشاهده کنند. اما پروتکل POP برای دسترسی یک صندوق پستی در مکان‌های مختلف اصلاً مناسب نیست. چون دیگر سرویس دهنده قادر نیست تا بداند شما چه نامه‌هایی را خوانده‌اید و چه نامه‌هایی را نخوانده‌اید! شما مجبورید هر بار تمامی ایمیل‌ها را در رایانه‌ی خود بارگذاری کنید. (حتی بعضی از سرویس دهنده‌ها بعد از دریافت ایمیل‌ها ، آن‌ها را حذف می‌کنند!) بنابراین نیاز به پروتکلی احساس شد که بتواند از تگ گذاری پشتیبانی کند و مشکل دسترسی از چندجای مختلف را در یک صندوق پستی برطرف کند و همچنین ویژگی‌هایی دیگری مانند چند صندوق پستی برای یک نفر به سرویس دهنده‌ها اضافه کند. این نیاز باعث بوجود آمدن پروتکل Internet Message Access Protocol یا IMAP با استاندارد RFC 3501 شد. پروتکلی بسیار قدرتمند که از سال ۲۰۰۳ تا کنون تمامی افراد استفاده کننده از ایمیل در تمامی نقاط را راضی کرده است.

اما ذکر این نکته هم لازم است که پروتکل POP هنوز هم برای افرادی مناسب است. البته با اضافه شدن ویژگی‌هایی همچون انتخاب تاریخ و ساعت دریافت نامه‌ها (به طوری که کاربر مشخص می‌کند که نامه‌هایی را می‌خواهم که امروز رسیده باشند) بعضی از نواقص را رفع کرده است.

هدف اصلی ما در این مقاله بررسی پروتکل بسیار قدرتمند IMAP است که شرکت‌های بزرگی همچون گوگل ، یاهو و مایکروسافت و دیگران از آن برای سرویس‌های ایمیل خود استفاده می‌کنند است. اما نگاهی اجمالی هم به یک جلسه‌ی POP3 نمونه خواهیم انداخت.

اما قبل از این موارد بایستی نحوه‌ی نصب سرویس دهنده‌ی IMAP و POP را در گنو/لینوکس شرح دهیم. ما برای این مقاله ، سرویس دهنده‌ی Courier را انتخاب کرده‌ایم . (<http://www.courier-mta.org>)

نصب سرویس دهنده‌ی Courier در گنو/لینوکس

در صورتی که از توزیع دبیان (www.debian.org) و یا توزیع‌های مبتنی بر آن استفاده می‌کنید ، کافی است دو خط را در shell وارد کنید :

```
sudo apt-get install courier-imap
sudo apt-get install courier-pop
```

اما در صورتی که بسته باینری برای توزیع خود پیدا نکردید کافی است که سورس آن را از اینجا (<http://www.courier-mta.org/download.php>) بگیرید و طبق دستورالعمل کامپایل postfix در بخش‌های قبلی، آن را کامپایل کنید. توجه کنید که بسته‌ی Courier-IMAP را هم دانلود و کامپایل کنید.

اما سرویس‌دهنده‌ی Courier برخلاف postfix نیاز به تنظیمات برای هر کاربر دارد. بعد از نصب، دستورات زیر را در شل وارد کنید:

```
maildirmake /etc/skel/Maildir
maildirmake /etc/skel/Maildir/.Drafts
maildirmake /etc/skel/Maildir/.Sent
maildirmake /etc/skel/Maildir/.Trash
maildirmake /etc/skel/Maildir/.Templates
```

بعد از آن دستورات زیر را وارد کنید (به جای myuser نام کاربری خود را وارد کنید)

```
cp -r /etc/skel/Maildir /home/myuser/
chown -R myuser:usergroup /home/myuser/Maildir
chmod -R 700 /home/myuser/Maildir
```

حال تنظیمات به پایان رسیده است. در صورتی که دایمن (Daemon) سرویس‌دهنده‌ی courier اجرا نشده است، آن را با دستورات زیر اجرا کنید (قبل از آن از ورود با root اطمینان حاصل کنید و یا sudo را قبل از آن قرار دهید):

```
/etc/init.d/courier-imap start
/etc/init.d/courier-pop start
```

هم‌اکنون سرویس‌دهنده‌ی شما به درستی نصب شده است. فایل پیکربندی در صورتی که می‌خواهید نام کامپیوتر خود در شبکه را تغییر دهید، به صورت زیر است:

در صورتی که تغییراتی ایجاد کرده‌اید بایستی سرویس‌دهنده‌ی Courier را دوباره راه‌اندازی کنید. (به جای start، در دستورات بالا restart را قرار دهید)

برای IMAP:

```
/etc/courier/imapd
```

برای POP:

```
/etc/courier/pop3d
```

هم‌اکنون آماده‌ایم تا یک جلسه‌ی POP را انجام دهیم. پروتکل POP به صورت پیشفرض در پورت ۱۱۰ آماده‌ی کار است. برای همین دلیل بایستی به این پورت (در این مثال در itnotes.info) از telnet استفاده کنیم. یک جلسه‌ی POP نمونه را در زیر می‌بینید:

```
telnet itnotes.info 110
Trying 74.55.179.243...
Connected to itnotes.info.
Escape character is '^]'.
+OK Hello there.
user myuser
+OK Password required.
pass mypasswordhere
+OK logged in.
LIST
+OK POP3 clients that break here, they violate STD53.
1 1399
2 2788
```

12

3 3283

.
RETR 2
+OK 2788 octets follow.
Return-path: <noreply@aryawebads.net>
Envelope-to: info@itnotes.info
Delivery-date: Sat, 26 Sep 2009 07:17:17 -0500
Received: from [82.146.52.199] (helo=srv44.minivds.com)
by persia.irandns.com with esmtps (TLSv1:AES256-SHA:256)
(Exim 4.69)
(envelope-from <noreply@aryawebads.net>)
id 1MrWDF-0004Km-Mf
for info@itnotes.info; Sat, 26 Sep 2009 07:17:17 -0500
Received: from srv44.minivds.com (localhost.localdomain [127.0.0.1])
by srv44.minivds.com (8.13.8/8.13.8) with ESMTTP id n8QCHHfi014077
for <info@itnotes.info>; Sat, 26 Sep 2009 05:17:18 -0700
Received: (from apache@localhost)
by srv44.minivds.com (8.13.8/8.13.8/Submit) id n8QCHHoP014075;
Sat, 26 Sep 2009 05:17:17 -0700
Date: Sat, 26 Sep 2009 05:17:17 -0700
Message-Id: <200909261217.n8QCHHoP014075@srv44.minivds.com>
X-Authentication-Warning: srv44.minivds.com: apache set sender to
noreply@aryawebads.net using -f
To:
Subject: AryaWeb Services
MIME-Version: 1.0
From: noreply@aryawebads.net
Reply-To: noreply@aryawebads.net
X-Sender: noreply@aryawebads.net
X-Mailer: noreply@aryawebads.com
Precedence: bulk
List-Unsubscribe: unsubscribe@aryawebads.com
Content-Type: text/html;charset=utf-8
Content-Transfer-Encoding: quoted-printable
X-MIME-Autoconverted: from 8bit to quoted-printable by srv44.minivds.com id
n8QCHHfi014077

```
<html><head><title></title> <meta http-equiv=3D'Content-Type' content=3D'=
text/html; charset=3Dutf-8'><meta http-equiv=3D'Content-Language' content=
=3D'fa'></head><body><table border=3D'0' cellpadding=3D'0' cellspacing=3D=
'0' width=3D'100%'><tr><td align=3D'center'>=EF=BB=BF<a href=3D"http://ww
w.aryaweb.com" target=3D"_blank">
<img border=3D"0" src=3D"http://www.aryawebads.com/ads/aryaweb008/service=
s.jpg" width=3D"600" height=3D"768" alt=3D"aryaweb.com" title=3D"=D8=AE=D8=
=AF=D9=85=D8=A7=D8=AA =D8=B4=D8=B1=DA=A9=D8=AA =D8=A2=D8=B1=D9=8A=D8=A7=D9=
=88=D8=A8">
</a><img src=3D'http://www.aryawebads.com/global/counter.php' style=3D'vi
sibility:hidden' width=3D'1' height=3D'1' /></td></tr><tr><td align=3D'ce
nter'><table border=3D'0' cellpadding=3D'0' cellspacing=3D'0' width=3D'60=
0'><tr><td><p dir=3D'ltr' align=3D'center'><font face=3D'Courier New' sty
le=3D'font-size: 9pt' color=3D'#808080'>This email has been sent to you b
y AryaWebAds Online Advertising Solutions<br>If you don't want to rec
eive similar emails, you can <a href=3D'mailto:unsubscribe@aryawebads.com=
' style=3D'text-decoration: none'><font color=3D'#00B360'>unsubscribe</fo
nt></a> at any time.<br>Interested in sending similar emails?<a href=3D'h
ttp://www.aryawebads.com/global/index.php/m_order' style=3D'text-decorati
on: none'><font color=3D'#00B360'>more info</font></a>.</font></td></tr><=
```

```
/table></td></tr></table></body></html>
```

```
.
DELE 2
+OK Deleted.
QUIT
+OK Bye-bye.
Connection closed by foreign host.
```

همان طور که می بینید در ابتدا به پورت ۱۱۰ از سایت itnotes.info (سایت خودم!) ارتباطی برقرار شده است و سرور با خط +OK Hello there به معنای ارتباط موفقیت آمیز است. اما در سرویس دهنده ی POP چون شما مستقیماً می توانید به نامه های افراد دسترسی داشته باشید بایستی نام کاربری و رمز عبور خود را وارد کنید. برای همین با دستور user و سپس نام کاربری در صورتی که نام کاربری معتبر باشد، بایستی پیغامی شبیه (+OK Password required) بگیرید. پس از آن بایستی رمز عبور را با فرمان pass و سپس رمز عبور خود وارد کنید. در صورتی که رمز درست باشد بایستی پیغامی مانند (+OK logged in) بگیرید.

پس از ورود به حساب کاربری می توانید که دستورهای مختلف را بر روی console وارد کنید. برای نمایش لیست ایمیل های موجود، فرمان LIST را وارد می کنیم. که با کاراکتر نقطه (.) پایان می یابند. برای نمایش محتویات یک ایمیل کافی است شماره ی آن را که در دستور LIST پیدا کرده ایم بعد از دستور RETR بیاوریم. که در این مثال یک نامه ی واقعی با تمامی سرآیندها آورده شده است. پایان دستور RETR با نقطه (.) مشخص می شود. با دستور DELE و شماره ایمیل، می توان آن را حذف کرد. و در نهایت برای بستن جلسه از دستور QUIT می توان استفاده کرد.

پروتکل IMAP

دلایل پیدایش پروتکل IMAP را در مطالب قبلی (از جمله ناتوانی در دستیابی به ایمیل در چندمکان و داشتن چندین اشتراک) برشمردیم. حال می خواهیم به بحث عملی این پروتکل پرداخته و وارد بحث اصلی شویم. IMAP در مقایسه با POP پروتکلی به مراتب پیچیده تر است. بوسیله ی POP با تعدادی فرمان می توان به نتیجه رسید. اما در IMAP داستان کاملاً متفاوت است. تفاوت اصلی مابین IMAP و دیگر پروتکل های خطی دیگر این است که هر فرمان صادر شده توسط یک سرویس گیرنده باید بوسیله ی رشته ای از کاراکترهای الفبایی عددی به نام tags همراه شود. با تگ ها، یک سرویس گیرنده ی IMAP می تواند دریابد که کدام پاسخ سرویس دهنده توسط چه فرمانی برگشت داده می شود. سرویس گیرنده ی IMAP باید تگ های منحصر بفردی را در حین یک جلسه تولید کند. سرویس دهنده نیز به نوبه ی خود تگی که سرویس گیرنده به پاسخ آن ارسال می دارد را الصاق می کند. سرویس گیرنده نیز ممکن است پاسخ ها را بدون تگ بدهد. پاسخ های بدون تگ بوسیله ی یک ستاره (*) ابتدای خط ظاهر می شوند.

IMAP به شما اجازه می دهد که سلسله مراتبی از چندین صندوق پستی را به همان طریقی که دایرکتوری فایل ها را دستکاری می کنید، ایجاد نمایید. قالب های سلسله مراتبی صندوق پستی، بسته به پیاده سازی سرویس دهنده فرق می کنند، اما ساده ترین فرم یک قالب، سیستم فایل پستی است. در این قالب هر صندوق پستی می تواند به یک دایرکتوری یا فایل نگاشت شود. INBOX همیشه یک دایرکتوری است.

یک جلسه IMAP نمونه

برای شروع کار بایستی به پورت ۱۴۳ از یک هاست telnet کنید.

```
telnet itnotes.info 143
Trying 74.55.179.243...
Connected to itnotes.info.
```

Escape character is '^'.

```
* OK [CAPABILITY IMAP4rev1 UIDPLUS CHILDREN NAMESPACE THREAD=ORDEREDSUBJECT
THREAD=REFERENCES SORT QUOTA IDLE ACL ACL2=UNION STARTTLS] Courier-IMAP
ready. Copyright 1998-2008 Double Precision, Inc. See COPYING for
distribution information.
```

در صورتی که با پیغام بالا مواجه شدید ، یعنی سرویس‌دهنده‌ی IMAP شما به درستی تنظیم شده است و آماده‌ی ورود کاربر است. همان‌طور که توضیح داده شد ، دستورات را بایستی با یک تگ آغاز کنید. در این مثال از تگ WCK به همراه یک عدد یکتا استفاده می‌کنیم تا کلاً تگ‌های استفاده شده ، یکتا باشند.

حال بعد از تگ بایستی نام کاربری و رمز عبور را وارد کرده تا لاگین کنیم :

```
WCK01 login itnotes mypassword
WCK01 OK LOGIN Ok.
```

حال آماده‌ایم تا دستورات IMAP را وارد کنیم.

دستور LIST

با استفاده از این دستور می‌توانید لیست دایرکتوری‌های موجود در سرویس‌دهنده مربوط به اشتراک خود را ببینید. این دستور دو آرگومان می‌پذیرد که اولی نام دایرکتوری و دومی زیردایرکتوری آن است. برای مثال

```
WCK02 LIST "" "*"
* LIST (\HasNoChildren) "." "INBOX.Trash"
* LIST (\Marked \HasChildren) "." "INBOX"
* LIST (\HasNoChildren) "." "INBOX.Sent"
* LIST (\HasNoChildren) "." "INBOX.Drafts"
WCK02 OK LIST completed
```

ملاحظه می‌شود که در این اشتراک ، یک پوشه‌ی INBOX و سه زیر پوشه‌ی Sent ، Trash و Drafts وجود دارد که با نقطه بعد از INBOX مشخص شده‌اند. توجه کنید که خطوطی که با ستاره (*) شروع شده‌اند یعنی Untagged هستند و جواب اصلی آنی است که با تگ وارد شده جواب داده شده است (یعنی WCK02). در هر سطر مشاهده می‌کنید که بعد از List در داخل پرانتزها پرچم‌های (Flag) هر پوشه مشخص شده است. برای مثال در INBOX مشخص شده است که \HasChildren به معنای اینکه این پوشه شامل زیرپوشه است. که در ادامه کنترل بیشتری بر این پرچم‌ها خواهیم داشت.

دستور STATUS

این دستور بدون داخل شدن در یک پوشه می‌تواند اطلاعاتی در مورد آن به ما بدهد. اطلاعاتی از قبیل تعداد کلی نامه‌ها ، تعداد نامه‌های اخیر ، تعداد نامه‌های خوانده شده. که هر سه نوع آن را در سه مثال زیر می‌بینید:

```
WCK03 STATUS inbox(messages)
* STATUS "inbox" (MESSAGES 2)
WCK03 OK STATUS Completed.
WCK04 STATUS inbox(recent)
* STATUS "inbox" (RECENT 2)
WCK04 OK STATUS Completed.
WCK05 STATUS inbox(unseen)
* STATUS "inbox" (UNSEEN 2)
WCK05 OK STATUS Completed.
```

دستورهای SELECT و EXAMINE

در حالت کلی برای کار کردن با یک پوشه و گرفتن اطلاعات کامل از آن نیاز دارید تا آن را انتخاب کنید. هر دو دستور SELECT و EXAMINE این کار را برایتان انجام می‌دهند، اما دستور EXAMINE پوشه را در حالت Read-Only باز می‌کند و دیگر امکان تغییرات و یا حذف فایل‌ها و فلگ‌ها وجود ندارد.

مثالی از انتخاب inbox با select

```
WCK06 EXAMINE INBOX
* FLAGS (\Draft \Answered \Flagged \Deleted \Seen \Recent)
* OK [PERMANENTFLAGS ()] No permanent flags permitted
* 2 EXISTS
* 2 RECENT
* OK [UIDVALIDITY 1262294407] Ok
* OK [MYRIGHTS "acdilrsw"] ACL
WCK06 OK [READ-ONLY] Ok
```

مثالی از انتخاب inbox با examine

```
WCK07 SELECT INBOX
* FLAGS (\Draft \Answered \Flagged \Deleted \Seen \Recent)
* OK [PERMANENTFLAGS (\* \Draft \Answered \Flagged \Deleted \Seen)] Limited
* 2 EXISTS
* 2 RECENT
* OK [UIDVALIDITY 1262294448] Ok
* OK [MYRIGHTS "acdilrsw"] ACL
WCK07 OK [READ-WRITE] Ok
```

همان‌طور که ملاحظه می‌کنید دستور SELECT این کار را به صورت READ-WRITE و دستور EXAMINE به صورت READ-ONLY انجام داد. خطوط اول و دوم پرچم‌های موجود و قابل اعمال را نشان می‌دهند. در خط سوم نشان از وجود ۲ نامه و در خط چهارم وجود ۲ نامه‌ی خوانده نشده دارد. خط پنجم شناسه‌ی این پوشه را مشخص کرده است و خط ششم حقوق‌های این پوشه را نشان داده است (حقوق در ادامه بررسی خواهند شد)

ایجاد، ویرایش و یا حذف پوشه‌ها

شما می‌توانید زیر پوشه‌هایی برای INBOX ایجاد، تغییر نام و یا حذف کنید. (البته توجه کنید که امکان تغییر نام پوشه‌های پیشفرض وجود ندارد). برای ایجاد پوشه‌ای جدید به راحتی از دستور CREATE و نام پوشه استفاده می‌کنیم. توجه کنید که تمامی پوشه‌ها بایستی زیر پوشه‌ی INBOX باشند.

```
WCK08 CREATE INBOX.bahram
WCK08 OK "INBOX.bahram" created.
```

برای تغییر نام کافی است از دستور RENAME استفاده کنید که دو آرگومان آن نام اولیه و نام جدید خواهد بود.

```
WCK09 RENAME INBOX.bahram INBOX.bahramwhh
WCK09 OK Folder renamed.
```

حال در صورتی که دوباره دستور LIST را برای همه (*) بزنییم تا خروجی کار را ببینیم:

```
WCK10 LIST "" "*"
* LIST (\HasNoChildren) "." "INBOX.Trash"
* LIST (\Marked \HasChildren) "." "INBOX"
* LIST (\HasNoChildren) "." "INBOX.Sent"
* LIST (\HasNoChildren) "." "INBOX.Drafts"
* LIST (\HasNoChildren) "." "INBOX.bahramwhh"
WCK10 OK LIST completed
```

همان‌طور که ملاحظه می‌کنید زیر پوشه‌ی bahramwhh به INBOX اضافه شده است. برای حذف آن کافی است که از

دستور DELETE استفاده کنید.

```
WCK11 DELETE INBOX.bahramwhh
WCK11 OK Folder deleted.
```

نکته : می‌توانید زیر پوشه برای زیرپوشه نیز بسازید. بدین صورت که ابتدا یکی از زیر پوشه‌ها را SELECT کرده و سپس با دستور CREATE یک زیر پوشه بسازید. (و یا اینکه به صورت مستقیم به دستور CREATE ارسال کنید. مثلاً (INBOX.Drafts.bahram

بازیابی نامه‌ها با FETCH

حال نوبت آن رسیده‌است که با استفاده از دستور FETCH نامه‌های خود را بازیابی کنید و بخوانید. همان‌طور که توضیح دادیم، ایمیل‌ها دارای فرمت RFC 822 هستند. بنابراین در این استاندارد ما می‌توانیم فیلدهای سرآیند و متن را جداگانه بازیابی کنیم. پس از دستور FETCH بایستی شماره‌ی نامه را وارد کنید و پس از آن محدوده‌ای از پیام که می‌خواهید نمایش داده شود. برای مثال rfc822.header باعث می‌شود که سرآیندهای نامه نمایش داده شود. و یا rfc822.text باعث می‌شود متن نامه نمایش داده شود.

مثالی از هر دو نمونه را در زیر می‌بینید :

بازیابی سرآیندهای نامه‌ی شماره ۱ :

```
WCK12 FETCH 1 rfc822.header
* 1 FETCH (RFC822.HEADER {530})
Return-path: <>
Envelope-to: itnotes@persia.irandns.com
Delivery-date: Wed, 02 Sep 2009 01:35:46 -0500
Received: from mailnull by persia.irandns.com with local (Exim 4.69)
        id 1MijRa-000318-Ig
        for itnotes@persia.irandns.com; Wed, 02 Sep 2009 01:35:46 -0500
Auto-Submitted: auto-replied
From: Mail Delivery System <Mailer-Daemon@persia.irandns.com>
To: itnotes@persia.irandns.com
Subject: Mail failure - no recipient addresses
Message-Id: <E1MijRa-000318-Ig@persia.irandns.com>
Date: Wed, 02 Sep 2009 01:35:46 -0500
```

```
)
WCK12 OK FETCH completed.
```

بازیابی متن نامه‌ی شماره ۱ :

```
WCK13 FETCH 1 rfc822.text
* 1 FETCH (RFC822.TEXT {869})
A message that you sent contained no recipient addresses, and therefore no
delivery could be attempted.
```

----- This is a copy of your message, including all the headers. -----

```
To:
Subject: [ITLPoll] Your ITLPoll Account Password Changed
From: "itlpoll@sourceforge.net"
From: "itlpoll@sourceforge.net"@persia.irandns.com
Message-Id: <E1MijRa-000317-ID@persia.irandns.com>
Date: Wed, 02 Sep 2009 01:35:46 -0500
```

Hi Dear User,
The password of ITLPoll Script has been reseted now.

```
*****
You can login with this account :
UserName : admin
Password : itl952
*****
```

Login Address : <http://itlpoll.itnotes.info/itlpoll3/admin/index.php>

```
Thanks
ITLPoll Version 3
Programing : Bahrambeigy (bahramwhh@gmail.com)
)
WCK13 OK FETCH completed.
```

شما می توانید بعد از دستور هم با آکولاد حداکثر تعداد کاراکترهای خروجی را مشخص کنید.
به صورت سفارشی هم می توانید تعیین چه سرآیندهایی را می خواهید بازیابی کنید :

```
WCK14 fetch 1 (body[header.fields (from to subject date)])
* 1 FETCH (BODY[HEADER.FIELDS ("from" "to" "subject" "date")] {184}
From: Mail Delivery System <Mailer-Daemon@persia.irandns.com>
To: itnotes@persia.irandns.com
Subject: Mail failure - no recipient addresses
Date: Wed, 02 Sep 2009 01:35:46 -0500
```

```
)
* 1 FETCH (FLAGS (\Seen))
WCK14 OK FETCH completed.
```

یعنی با استفاده از header.fields و مشخص کردن سرآیندهایی که می خواهید و قرار دادن آن به عنوان اندیس آرایه ی body می توانید به صورت سفارشی هر سرآیندی را که خواستید ، بازیابی کنید.
توجه کنید که در انتهای FETCH به ایمیل پرچم خوانده شده (\Seen) نسبت داده می شود.

بازیابی پرچم نامه ها با دستور FETCH

با استفاده از دستور FETCH می توانید پرچم نامه ها را قبل از خواندن آن ها بازیابی کنید. قبل از آن توجه کنید که در پروتکل IMAP از دونقطه (:) برای بیان محدوده یا بازه استفاده می شود. برای مثال در صورتی که بخواهیم بازه ی ۱ تا ۳ را مشخص کنیم به صورت ۱:۳ نمایش می دهیم.

حال با استفاده از دستور FETCH می خواهیم پرچم ایمیل های ۱ تا ۲ را بازیابی کنیم :

```
WCK15 FETCH 1:2 FLAGS
* 1 FETCH (FLAGS (\Seen))
* 2 FETCH (FLAGS ())
WCK15 OK FETCH completed.
```

ملاحظه می شود که نامه ی اول خوانده شده و نامه ی دوم هنوز باز نشده است (یعنی دستور FETCH بر روی آن اعمال نشده است)

نکته: برای اعمال بر روی نامه های به طور مثال ۱ تا آخر بازه را به صورت *۱: مشخص می کنیم.

اعمال پرچم‌ها با استفاده از دستور STORE

با استفاده از دستور STORE سه نوع اعمال پرچم می‌توانید انجام دهید :

- اضافه کردن پرچم به پرچم‌های موجود :

STORE message +flags [flag list]

- حذف پرچم از پرچم‌های موجود :

STORE message -flags [flag list]

- پاک کردن پرچم‌های قبلی و درج پرچم‌های جدید :

STORE message flags [flag list]

هم‌اکنون هرکدام از این رویه‌ها را انجام می‌دهیم.

اعمال پرچم bahramwhh به نامه‌های اول تا دوم :

```
WCK16 STORE 1:2 +FLAGS bahramwhh
* FLAGS (bahramwhh \Draft \Answered \Flagged \Deleted \Seen \Recent)
* OK [PERMANENTFLAGS (bahramwhh \* \Draft \Answered \Flagged \Deleted
\Seen)] Limited
* 1 FETCH (FLAGS (\Seen bahramwhh))
* 2 FETCH (FLAGS (bahramwhh))
WCK16 OK STORE completed.
```

خروجی چیزی شبیه دستور SELECT است با این تفاوت که پرچم‌های جدید اعمال شده نیز نمایش داده می‌شوند.

حال این پرچم را از دو ایمیل اول و دوم می‌گیریم:

```
WCK17 STORE 1:2 -FLAGS bahramwhh
* 1 FETCH (FLAGS (\Seen))
* 2 FETCH (FLAGS ())
* FLAGS (\Draft \Answered \Flagged \Deleted \Seen \Recent)
* OK [PERMANENTFLAGS (\* \Draft \Answered \Flagged \Deleted \Seen)] Limited
WCK17 OK STORE completed.
```

اکنون به نامه‌ی اول پرچم Star می‌دهیم بدین صورت که پرچم قبلی آن (یعنی خوانده‌شده) حذف گردد :

```
WCK18 STORE 1 FLAGS Star
* FLAGS (Star \Draft \Answered \Flagged \Deleted \Seen \Recent)
* OK [PERMANENTFLAGS (Star \* \Draft \Answered \Flagged \Deleted \Seen)]
Limited
* 1 FETCH (FLAGS (Star))
WCK18 OK STORE completed.
```

چون در این فرمان تنها پرچم ایمیل اول را تغییر دادیم دیگر نیازی به نمایش پرچم دیگر ایمیل‌ها نیست.

نحوه‌ی حذف نامه‌ها

در IMAP دستور مستقیمی برای حذف نامه‌ها وجود ندارد! برای این کار شما بایستی ابتدا پرچم \Deleted را به ایمیلی که می‌خواهید حذف کنید ، بدهید و سپس دستور CLOSE و یا EXPUNGE را بزنید.

برای مثال در دستور زیر نامه‌ی ۱ را حذف کرده‌ایم :

```
WCK19 STORE 1 FLAGS \Deleted
* 1 FETCH (FLAGS (\Deleted))
* FLAGS (\Draft \Answered \Flagged \Deleted \Seen \Recent)
* OK [PERMANENTFLAGS (\* \Draft \Answered \Flagged \Deleted \Seen)] Limited
WCK19 OK STORE completed.
```

```
WCK20 CLOSE
WCK20 OK mailbox closed.
```

اکنون در صورتی که SELECT INBOX را بزنید ، می بینید که ایمیل شما حذف شده است (۱ ایمیل وجود دارد) :

```
WCK21 SELECT INBOX
* FLAGS (\Draft \Answered \Flagged \Deleted \Seen \Recent)
* OK [PERMANENTFLAGS (\* \Draft \Answered \Flagged \Deleted \Seen)] Limited
* 1 EXISTS
* 0 RECENT
* OK [UIDVALIDITY 1262294448] Ok
* OK [MYRIGHTS "acdilrsw"] ACL
WCK21 OK [READ-WRITE] Ok
```

نکته : تفاوت دو دستور CLOSE و EXPUNGE در این است که CLOSE بعد از حذف نامه‌هایی که پرچم \Deleted دارند ، پوشه‌ی INBOX را از حالت SELECT خارج می کند ولی EXPUNGE بدون بستن جعبه‌ی پستی شما این کار را می کند.

دستور COPY

با استفاده از این دستور می توانید ایمیل‌ها را به یک زیرپوشه‌ی خاص کپی کنید. مانند مثال زیر که نامه‌ی اول را به زیر پوشه‌ی Draft کپی می کند :

```
WCK22 COPY 1 INBOX.Drafts
WCK22 OK [COPYUID 1262298071 2 1] COPY completed.
```

دستور IDLE

با این دستور می توانید پوشه‌ی INBOX خود را در حالت مانیتورینگ قرار دهید تا در صورتی که ایمیلی رسید و یا تغییری در پوشه‌ی شما ایجاد شد ، باخبر شوید. برای پایان این حالت کافی است که DONE را بزنید.

```
WCK23 IDLE
+ entering idle mode
DONE
WCK23 OK IDLE completed
```

دستورهای ثبت نام در یک پوشه

سه دستور LSUB و SUBSCRIBE و UNSUBSCRIBE بیشتر برای برنامه‌های کاربردی کلاینت‌ها (MUA) کاربرد دارند. که با استفاده از دستور SUBSCRIBE و نام یک پوشه ، ثبت نام کرده و با استفاده از دستور LSUB می توانید آن‌ها را مشاهده کنید. که دستور LSUB بسیار شبیه دستور LIST است. به دلیل عدم کاربرد آن‌ها در کلاینت telnet فقط با ذکر مثالی از آن‌ها گذر می کنیم :

```
WCK24 SUBSCRIBE INBOX.Sent
WCK24 OK Folder subscribed.
WCK25 LSUB "" "*"
* LSUB (\HasNoChildren) "." "INBOX.Sent"
* LSUB (\Noselect \HasChildren) "." "INBOX"
WCK25 OK LSUB completed
WCK26 UNSUBSCRIBE INBOX.Sent
WCK26 OK Folder unsubscribed.
```

دستور خروج از جلسه

با استفاده از دستور LOGOUT می‌توانید جلسه‌ی IMAP را بسته و به ارتباط پایان دهید :

```
WCK27 LOGOUT
* BYE Courier-IMAP server shutting down
WCK27 OK LOGOUT completed
Connection closed by foreign host.
```

مشاهده‌ی حقوق دسترسی من

برای مشاهده‌ی حقوق دسترسی‌تان از دستور GETACL می‌توانید استفاده کنید. با زدن این دستور و بعد از آن پوشه‌ی مورد نظر ، خروجی زیر را خواهید دید :

```
WCK28 GETACL INBOX
* ACL "INBOX" "owner" "acdilrsw" "administrators" "acdilrsw"
WCK28 OK GETACL completed.
```

همان‌طور که می‌بینید حقوق برای دو نفر : دارنده (owner) و مدیر (administrator) مشخص شده است.

هر دسترسی با یک حرف مشخص می‌شود که آن‌ها در زیر می‌بینید :

l یا lookup flag : جعبه‌ی پستی برای دستورات LIST و LSUB قابل مشاهده است.

r یا read flag : امکان انتخاب (SELECT) پوشه‌ها ، کپی ، خواندن و جستجوی نامه‌ها وجود دارد.

s یا seen flag : امکان دریافت اینکه نامه‌ها خوانده شده یا نشده اند.

w یا write flag : امکان اعمال پرچم‌های حذف و خوانده‌شده بر روی ایمیل‌ها

i یا insert flag : امکان اجرای دستورات COPY و APPEND بر روی ایمیل‌ها

p یا post flag : امکان ارسال ایمیل به آدرس‌های ثبت شده

c یا create flag : امکان ایجاد زیر پوشه‌ها بر روی هر سلسله‌مراتب تعریف شده

d یا delete flag : امکان اعمال پرچم Deleted و سپس اجرای EXPUNGE یا CLOSE

a یا administer flag : امکان اجرای ACLSET برای کاربر

در این مقاله سعی شد اکثر دستوراتی که می‌توان از آن‌ها در IMAP استفاده کرد را به همراه مثال‌های مبسوط بررسی کردیم. اما واقعیت این است که بررسی تمامی دستورات IMAP نیاز به کتابی حجیم خواهد داشت و از بررسی اجمالی ما در این مقاله خارج است. اما به هر حال برای علاقه‌مندان لیست کامل دستورات IMAP به همراه شماره‌ی RFC آن‌ها جهت رجوع ، آورده شده است. امید است مفید واقع گردد.

لیست کامل دستورات IMAP به همراه شماره RFC:

Command	References
APPEND	RFC 3501 , RFC 3502 , RFC 4466 , RFC 4469
AUTHENTICATE	RFC 3501
CAPABILITY	RFC 3501
CHECK	RFC 3501
CLOSE	RFC 3501

COMPARATOR	RFC 5255
COMPRESS	RFC 4978
CONVERSIONS	RFC 5259
COPY	RFC 3501
CREATE	RFC 3501 , RFC 4466
DELETE	RFC 3501
DELETEACL	RFC 4314
EXAMINE	RFC 3501 , RFC 4466
EXPUNGE	RFC 3501
FETCH	RFC 3501 , RFC 4466
GETACL	RFC 4314
GETQUOTA	RFC 2087
GETQUOTAROOT	RFC 2087
LIST	RFC 3501
LISTRIGHTS	RFC 4314
LOGIN	RFC 3501
LOGOUT	RFC 3501
LSUB	RFC 3501
MYRIGHTS	RFC 4314
NOOP	RFC 3501
RENAME	RFC 3501 , RFC 4466
SEARCH	RFC 3501 , RFC 5032
SELECT	RFC 3501 , RFC 4466
SETACL	RFC 4314
SETQUOTA	RFC 2087
SORT	RFC 5256
STARTTLS	RFC 3501
STATUS	RFC 3501
STORE	RFC 3501 , RFC 4466
SUBSCRIBE	RFC 3501
THREAD	RFC 5256
UID	RFC 3501 , RFC 4315 , RFC 4466
UNSELECT	RFC 3691
UNSUBSCRIBE	RFC 3501
X<atom>	RFC 3501

Response	Reference
BAD	RFC 3501
BYE	RFC 3501
NO	RFC 3501
OK	RFC 3501
PREAUTH	RFC 3501

منابع :

۱. شبکه‌های کامپیوتری - اندرو اس. تنن‌بام - ویراست چهارم - ترجمه : دکتر حسین پدرام و دیگران چاپ چهاردهم - ناشر : نص
۲. مرجع کامل PHP4 - لوئیس آرگریک - ترجمه : گروه مهندسی پژوهشی ساحر - چاپ سوم - ناشر : انتشارات روزنه
۳. مرجع کامل دستورات IMAP در آدرس <http://www.networksorcery.com/enp/protocol/imap.htm>
۴. ارتباط با IMAP از طریق telnet در آدرس وبلاگ <http://adityo.blog.binusian.org/?p=335>
۵. <http://www.isi.edu/in-notes/rfc3501.txt>
۶. <http://www.postfix.org/documentation.html>
۷. ویکی‌پدیای انگلیسی و ویکی سایت لینوکس ابونتو (<http://help.ubuntu.com>)