

Elektronikuslevél-beépítés

Tartsuk ébren weboldalunk felhasználóinak az érdeklődését: figyelmeztessük őket elektronikus levélben a számukra érdekes hírekre és vitákra!

Több mint húsz éve használom a számítógépet az embertársaimmal való kapcsolattartásra. Kezdetben csak alkalmanként írogattam a helyi hirdetőtáblára, mindez mára személyes és szakmai életem elválaszthatatlan részévé vált. Ebben a hónapban – miközben folytatom a kicsomagolást új chicagói otthonomban, és küzdök a felmerülő hibákkal, amelyek megnehezítik, hogy folytassam a Bricolage-ról szóló cikksorozatamat – a jelenkor néhány, az elektronikus levelekkel kapcsolatos kérdésébe pillantok bele. Írásomban a web-, illetve a levelezésegysítésről, illetve a levelezés és az adatbázis egyesítéséről, sőt még az SMTP-szintű levélszemét elleni harcról is szó esik.

A web és az elektronikus levél egyesítése

Ma már valóban nehéz olyan nagy és korszerű weblapot találni, amelyen nincs megoldva felhasználói visszajelzések és jelentkezések kérdése. A kérdés csak az, hogy miként illeszthetünk be egy fórumot a weblapunkra? A válasz legtöbb esetben attól függ, hogy milyen oldalt használunk. Amennyiben nagyobb eszközkészletet alkalmazunk, például OpenACS-t, Zope-ot vagy PHPNuke-ot, legalább egy könnyen telepíthető csomagot fogunk találni a témában. Ilyenkor a felület egyrészt összhangban lesz a statikus oldalakkal és az egyéb alkalmazásokkal, valamint ezek az eszközkészletek oldalunk más részeivel azonos felhasználói és engedélyrendszerrel használják a fórumok elérésére. Más szavakkal, nem szükséges külön oldalgazdát és fórumgazdát kijelölnünk; bárki, aki rendszerfelügyeleti jogosultságokkal rendelkezik az oldalon, minden további beállítás nélkül képes lesz a fórumot és bármilyen más alkalmazást futtatni.

Másik megoldásként a kiszolgálónkon telepített kiszolgálóoldali technológiákat felhasználva felrakhatunk valamilyen önálló webfórumcsomagot. Ha például PHP-rendszerünk van, tegyük fel a *Phorum* csomagot. Ez ugyan nem fog teljes mértékben beépülni oldalunk egyéb részeibe, viszont igen hatékony és megbízható csomag, ami MySQL és PostgreSQL alatt egyaránt jól működik. Az alapul szolgáló technológiák szerint rengeteg lehetőség között választhatunk, a PHP és JSP nyelvektől kezdve egészen jó öreg Perlben írt CGI-programokig. Természetesen kereskedelmi terméket is választhatunk, megvásárolhatunk egy olyan terméket, mint például a *WebCrossing*. Néhány ügyfélnél használtam már a *WebCrossing*-ot, és meg kell mondanom, annak ellenére, hogy ez a csomag jóval több lehetőséget ajánl, mint a *Phorum*-hoz hasonló nyílt forrású csomagok, a különbségek a legtöbb esetben nem olyan nagyok, hogy megérje pénzt költeni rá, nem beszélve arról, hogy ehhez egy olyan csomagot kellene megvásárolni, amelyet nem lehet sem módosítani, sem fejleszteni.

Végül létrehozhatjuk a saját fórumcsomagunkat is, ahogy azt néhány évvel ezelőtt magam is megtettem e hasábon. A weblapú fórumok kialakítására képes programok megírása

nem is olyan nagyon bonyolult, de a fejlesztésre és hibakeresésre szánt időt és erőfeszítéseket sokkal érdemesebb egy létező csomag megtanulására fordítani. Akinek azonban van egy kis tapasztalata a web és az adatbázisok témában, az tudja, hogy egy webfórumcsomag létrehozása nemigen áll másból, mint hogy összerakol néhány táblát (felhasználók, üzenetek és a hírcsoportok), majd megadja a jogosultságot az embereknek, hogy bejegyzéseket (küldemények) helyezhessenek el az egyes hírcsoportokban. Ezek a rendszerek több mint elégségesek egy kis vagy közepes méretű weboldal számára; de ha időközben hatalmasra duzzadna is a rendszerünk, az elküldött üzenetek ezreivel és felhasználók százaival birkózva is valószínűtlen, hogy komoly hatása lenne ezekre a rendszerekre. Ugyanis valamennyi rendszer relációs adatbázisokat használ az adattárolásra, márpedig még a legkisebb és legegyszerűbb korszerű adatbázisrendszer is képes napi néhány ezer tranzakció kezelésére.

Egyre gyakrabban fordul elő azonban, hogy az egyszerű weblapú fórum már nem bizonyul elegendőnek. Bár az emberek általában hajlandók végignézni a weblapú fórumokat, nemigen térnek vissza nap mint nap, hogy kövessék a vita fonalát. Hiszen míg az elektronikus levél aktív (push) médium, amelyben az adatokat elküldik nekünk, a fórumok passzív (pull), ahol az új üzenetek a lekérésünkre várnak.

Az aktív és passzív felosztás nem új találmány; akik még emlékeznek közülünk a web előtti internet korára, jól tudják, hogy ugyanezen felosztás alapján különböztettük meg a levelezőlistákat és az Usenet hírcsoportokat. A megoldás végül az lett, hogy néhány mail-Usenet átjárót kellett készíteni, amelyek közül sokat még ma is használnak. A GNU Emacs legfrissebb hibáiról egyaránt tájékozódhatunk, ha feliratkozunk a `gnu.emacs.bug` levelezőlistára, vagy ha elolvassuk a `gnu.emacs.bug` hírcsoportot a Useneten. A két dolog egyenértékű, s egy átjáróprogramon alapul, amely felesleges ismétlések nélkül viszi át az üzeneteket egyik rendszerről a másikra. Létezik ilyen rendszer a weblapú fórumokhoz is? A válasz egyértelmű igen. Végte is nem túl nehéz elérni egy web-, illetve adatbázisrendszeren, hogy új üzenetek érkezésekor bizonyos híryanagokat egy adott elektronikus levélcímlistára küldjön el. Kifinomult figyelmeztetőrendszer felépítése sem sokkal nehezebb, létezik például egy *Forums* OpenACS-csomag, amely lehetővé teszi, hogy a felhasználók az adott fórumokra vagy fórumon belüli hírcsoportokra feliratkozzanak, majd havi, napi, illetve azonnali frissítést kérjenek maguknak. Más szavakkal a Forum program képes elektronikuslevél-tömörítvényeket (digest) készíteni, pontosan úgy, ahogy azt a lista-programok teszik.

Levél küldése

Ha viszont azt szeretnénk, hogy a jelentkezők saját elektronikus levelezőprogramjukkal is üzeneteket tudjanak küldeni



a listára, a dolgok bonyolultabbá kezdenek válni, ugyanis figyelembe kell vennünk néhány tényezőt:

- **Biztonság és engedélyek** – bárki küldhet üzeneteket és válaszolhat a fórumra vagy csak a tagok tehetik meg? Ha a fórum elérése valamilyen szinten kötött, nyilván kell tartanunk az engedélyezett elektronikus levélcímeket. Minthogy közismerten nem túl nagy művészet összekalapálni egy hamis elektronikus levélfejlécet, nem igazán lehet biztonsággal megállapítani, hogy a küldemény valóban a feliratkozótól jött-e, esetleg éppen egy féreg vagy vírus üzenet a feliratkozó nevében.
- **MIME** – a legnépszerűbb elektronikus levelezőprogramok (különösen a Microsoft Outlook) alapértelmezés szerint egy vagy több csatolmánnyal küldik a leveleket. A Forum programnak elég intelligensnek kell lennie, hogy kezelni tudja az ilyen típusú leveleket, kiszedve a csatolmányt és a HTML-részeket.
- **Méret** – amennyiben a levélfórum-átjáró nem elég okos ahhoz, hogy kigyomlálja a különösen nagy küldeményeket, akkor egy hatalmas küldemény postázásával bárki szolgáltat-megtagadás- (Denial-of-Service) támadást intézhet a weboldalunk ellen. A programnak elég okosnak kell lennie ahhoz is, hogy az ilyen leveleket kiszűrje, és az oldal gazdájának lehetővé tegye a felhasználói levélméret korlátozását.
- **Témák** – a legtöbb fórumprogram az együvé tartozó küldeményeket a küldemény címe alapján együtt kezeli, vagy megjegyzi, hogy melyik levél melyik levélre volt válasz. Igaz, ez nem túl egyszerű, de a nyomon követés a web- és elektronikus levél egyesítése esetében is megoldható, bár mindkét médium más típusú rendszert alkalmaz az ilyesféle dolgok nyilvántartására.

Számos megoldást láttam az ilyen és hasonló nehézségek kezelésére, de mind a mai napig nem találtam olyat, amelyik valamennyi kívánalmamat teljesítette volna.

A *Phorum* egy *phorummail* nevű fájlt ad nekünk, amit parancs-sorból kell meghívunk, feltételezhetően a *.forward* vagy *.qmail* állományból, vagy a *mail alias* meghatározásokat tartalmazó fájlból. A rendszergazdának készítenie kell egy álnevet (például *apartments-forum*), majd a *.forward* fájlt a *phorummailre* állítva átadja neki a kötelező *FORUM_ID* és az elhagyható *PATH_TO_FORUM* értékeket. Amint ezzel megvagyunk, bárki, aki levelet ír az *apartments-forum@sajat.webhely* címre, egyben a fórumunkra is írhat. Nyilvánvaló módon a *phorummail* helyes működéséhez az engedélyeket megfelelően kell beállítani.

A gond csak az, hogy a küldeményekkel kapcsolatban nincs különösebb biztonság; igaz ugyan, hogyha a fórum moderált, a levélben érkezett üzenetek addig elfogadatlannak lesznek bejelölve, amíg a moderátor meg nem nézi őket. A témakezelést többféle ügyes módszerrel is elintézhethetjük, úgy tűnik viszont, hogy nemigen megoldott a MIME- vagy a levélbomba-támadások kezelése. Más szóval, a *Phorum* levélfórum elvárásainknak megfelelően kezeli a leveleket, de ennél többet nemigen nyújt. Az *OpenACS* fórumprogramja már kifinomultabb, *qmail* alapú rendszert nyújt, amelyben minden kimenő figyelmeztető üzenet egyedi azonosítóval rendelkezik. Ez azt jelenti, hogy az elektronikus levélüzenetre küldött válasz a kérdéses témához kerül. Igazság szerint az, hogy az *OpenACS* az elektronikus levélcímeket használja bejelentkező névként, eléggé gyatra biztonságot ad, de továbbra is tény, hogy nagyon nehéz megállítani az üzenetek hamisítását.

Levelezőlisták és adatbázisok

Az általam ajánlható legjobb megoldás, ha mindent fejreállítva egy már létező elektronikus levéllista alapján hozunk létre magunknak webalapú fórumot. A *qmail* szerzője, *Dan Bernstein* által készített *ezmlm* például rendelkezik egy *ezmlm-idx* nevű bővítménykészlettel, amely többek közt lehetővé teszi, hogy a feliratkozók listáját MySQL- vagy PostgreSQL-adatbázisban tároljuk. A lista beállításakor a rendszergazda egyúttal néhány adatbázistáblát is létrehoz, majd az *ezmlm*-ben beállítja őket.

Így egy jó webfejlesztő létrehozhat egy elektronikus levéllistát, majd azt tükrözheti a weben. Bármi, ami a webről érkezik, úgy néz majd ki, mintha a pillanatnyilag bejelentkező felhasználótól érkezne, akinek feltételezhetőleg be kell jelentkeznie a webalapú fórumalkalmazásba. Bármilyen adat, amely a pillanatnyi felhasználótól érkezik, ugyanazon a szokásos ellenőrzésen megy keresztül, amit az *ezmlm* egyébként alkalmaz.

Egyelőre úgy tűnik, hogy a *mailman* levelezőlista, amely minden MTA-val képes együttműködni (például *qmail*, *Sendmail*, *Postfix* és *Exim*) és folyamatos, lenyűgöző fejlesztés alatt áll, semmiféle olyan megoldással nem rendelkezik, amellyel a felhasználóit relációs adatbázisba menthetné. Ugyanakkor viszonylag biztonságos, könnyen használható Berkeley DB fájlokban tárolja őket, így webalapú fórumcsomagunk innen kiolvashatja őket.

Ha a fórumokat levelezőlistaként használjuk, azért felmerülhet néhány gond: többek között a témacsoportok kérdése, amely – mint fentebb említettük – egészen másképp kezelhető az elektronikus levél és a web esetében. Vegyük ehhez hozzá azt a tény, hogy a legtöbb fórum a felhasználóinak lehetővé teszi, hogy kiemeléseket és csatolmányokat használjanak, sőt akár szerkeszthessék is a saját küldeményeiket – és máris nyilvánvalóvá válik, hogy ez a házasság a két médium között nem lesz éppen súrlódásmentes, mégha létre is tudjuk hozni. Mindazonáltal a szolgáltatások sok olyan ember számára hasznosnak bizonyulhatnak, akik hajlandóak lemondani a cifra dolgokról. Ehelyett ugyanis a fórumok egyre növekvő számú jelentkezőivel vannak elfoglalva, és a felhasználóknak megadják a lehetőséget, hogy eldöntsék, milyen formában szeretnének jelentkezni ezekre a fórumokra.

qpsmtpd

Ahogy nyilván önök közül is sokan, jómagam is nap mint nap szenvedek a kéretlen levelek áradatától. A nyílt forráskódú *SpamAssassin* képes kategorizálni a bejövő leveleket, így a levélszemét (spam) ellen folytatott küzdelmeimben kiváló szövetségesnek bizonyult. Amennyiben futtatunk már valaha levelezőlistát, kétségtelenül felfedeztük, hogy a levélfergek nemigen válogatnak: éppoly előszeretettel küldik magukat a listákra, mint az egyéneknek.

Bár a rendszerfelügyelet és az SMTP-kiszolgálók nem igazán illeszkednek szorosan cikkünk témájához, úgy éreztem, muszáj néhány dicsérő sort szentelnem *Ask Bjoern Hansen* nyílt forrású SMTP-kiszolgálójának, a *qpsmtpd*-nek. A *qpsmtpd* eredetileg *qmail* mellett működött, jelenleg már számos MTA-val képes együttműködni, ideértve a *Sendmail* és a *Postfix*et.

De miért használnánk *qpsmtpd*-t az alapértelmezett *qmail-smtpd* helyett? A Perl-guruk számára egyszerű az indok a váltásra, hiszen *qpsmtpd* Perlben készült. De ha ennél egy kicsit kevésbé vagyunk nyelvőrültek, akkor is számos jó dolgot találhatunk benne. Ugyanis a *qpsmtpd* az SMTP-k levél-

küldő eljárásait több állomásra osztja fel, mi pedig valamennyi ilyen állomáshoz saját szolgáltatásokat illeszthetünk be. A `qpsmtpd` saját lapjáról, a <http://www.developer.com>-ról (igen, két o betű áll egymás után) töltöttem le, majd a telepítési útmutatót követve húsz perc múlva már futott is. Ne feledjük, hogy a `qpsmtpd` teljes értékű SMTP-kiszolgáló, azaz nem fog futni, ha más SMTP-kiszolgáló ellenőrzi a 25-ös kaput. Amennyiben az SMTP-kiszolgáló rendszerindításkor való indulását és későbbi futását a `daemontools` program segítségével végezzük, érdemes ellenőrizni, nem maradt-e véletlenül egy régi SMTP-kiszolgálóra mutató hivatkozás a `/service` könyvtárban. Ilyen esetben ugyanis a legközelebbi újraindítás után két versengő SMTP-démonunk lesz. A `qpsmtpd` kulcselemei a *plugins* alkönyvtárban található bővítmények. A soronként egy bővítményt tartalmazó *config/plugins* állomány segítségével magunk is beilleszthetünk saját bővítményeket. Például a saját *config/plugins* állományom egyik részlete így néz ki:

```
# quit_fortune
check_earlytalker
count_unrecognized_commands 4
require_resolvable_fromhost
```

Más szóval, megjegyzésbe tettem a `quit_fortune` bővítményt, viszont aktiváltam a `check_earlytalker`, `count_unrecognized_commands` és `require_resolvable_fromhost` bővítményeket. A `count_unrecognized_commands` értéként egyetlen számot vár, amit itt meg is adtunk. Ha meg szeretnénk nézni ezeket a bővítményeket, esetleg sajátot szeretnénk elhelyezni, lépünk be *plugins* könyvtárba. Minden bővítmény tartalmaz egy regisztráló alprogramot, amellyel a `qpsmtpd` valamelyik állomásához csatlakozik, valamint egy másik alprogramot, amelyet azután az állomás meghívhat. Például a `require_resolvable_fromhost` bővítmény a következőképpen kezdődik:

```
use Net::DNS qw(mx);

sub register {
    my ($self, $qp) = @_;
    $self->register_hook("mail",
        ↪ "mail_handler");
}

Azaz a regisztráló alprogram értesíti a qpsmtpd-t, hogy amint az SMTP-ügyfél meghívja a mail parancsot, a mail_handler alprogramot is meg kell hívni. Ez az alprogram pedig a következőképpen működik:
```

```
sub mail_handler {
    my ($self, $transaction, $sender) = @_;

    $sender->format ne "<>"
        and $self->qp->config
            ("require_resolvable_fromhost")
        and !check_dns($sender->host)
        and return (DENYSOFT,
            ($sender->host
                ? "Could not resolve ". $sender->host
```

```
: "FQDN required in the envelope
    ↪ sender"));
    return DECLINED;
}
```

Ha már volt dolgunk `mod_perl` alatti webfejlesztéssel, akár ismerősnek is tűnhet. A `mail_handler` egyrészt visszaadhat `DECLINED` értéket, ami azt jelenti, hogy minden simán ment, és a levél keresztlétehet. Küldhet `DENYSOFT` értéket is, ami a felhasználónak azt üzeni, hogy inkább később próbálkozzon. Ezt azért tesszük, mert nem szeretnénk azonnal elutasítani a leveleket csak azért, mert a DNS-kiszolgáló laállt – kizárólag a kéretlen levélküldőket szeretnénk megbüntetni, illetve azokat, akiknek nem kellene közvetlenül elküldeniük a levelet. Küldhetünk továbbá `DENY` üzenetet, amely véglegesen elutasítja az üzenetet. Annak ellenére, hogy nemigen kapunk hozzá túlságosan jó leírást, a `qpsmtpd` letöltését követően néhány órával már meg tudtam írni az első működőképes bővítményemet, és biztos vagyok benne, hogy a többi felhasználó is hasonló tapasztalatokat fog szerezni. Minthogy a `qpsmtpd` Perlben íródott, természetesen gyorsan és könnyen elérhetjük a Perl-programok által hozzáférhető összes lehetőséget, valamint a fejlesztést leegyszerűsítő a CPAN-modulokat.

Több szinthez is csatlakozhatunk bővítményekkel, többek közt a `helo`, `ehlo`, `connect`, sőt akár a `rcpt` szinthez is, amelyek mindegyikén különféle tesztek véggezhetünk el. Még *SpamAssassin*-bővítmény is létezik `qpsmtpd`-hez, amely a híres levélszemélszűrő programot hívja meg, mielőtt a levél megérkezne a levelesládánkba.

Körülbelül egy hónapja használok `qpsmtpd`-t, és a levelesládám vándorló kéretlen levelek száma örvendetesen megcsappant, mégha néhányat át is enged a *SpamAssassin*. Ha a saját gépünkön dolgozunk, jó szívvel ajánlom a `qpsmtpd`-t. Kiváló példa arra nézve, hogyan írjunk tetszőleges bővítményeket kezelő programot, ráadásul csak azt a levelet kapjuk meg, amelyet tényleg meg kell kapnunk.

Összegzés

A levelezés létfontosságú része az internetnek, mint azt kétségtelenül mindenki tudja, aki ezeket a sorokat olvassa. Az internet azonban folyamatosan terjeszkedik, így a levelezésre több irányból is nyomás nehezedik. Ebben a hónapban a webalapú fórumokkal ismerkedhettünk meg, illetve megnéztük, miképpen lehet csatlakoztatni őket a levelezőlistákhoz. Röviden bepillantottunk a `qpsmtpd`, egy választható, különlegesen bővíthetőre és átláthatóra tervezett SMTP-kiszolgáló világába. Remélem, a következő hónapban visszatérhetünk a *Bricolage* és a többi nyílt forrású tartalomkezelő rendszer birodalmába, nagy hangsúlyt fektetve arra, hogyan illeszthetünk be már létező weboldalainkba valamilyen CMS-t.

A Kapcsolódó címek megtalálhatóak az 55. CD Magazin/Elektronikuslevel könyvtárban.

Linux Journal 2003. december, 116. szám



Reuven M. Lerner ([↪ http://www.lerner.co.il/atf](http://www.lerner.co.il/atf))
Nyílt forrású programokra, valamint web- és adatbázis-alkalmazásokra szakosodott tanácsadó. Könyve, a *Core Perl*, 2002 januárjában jelent meg a Prentice Hall gondozásában. Reuven feleségével és lányaival nemrég költözött Chicagóba.