

Blosxom

Modulok nélkül, SQL nélkül, gond nélkül. Készítsünk nagyteljesítményű blogot, akár a webkiszolgáló újraindítása nélkül!

A weblogok, más néven blogok népszerűsége sokat nőtt az elmúlt néhány évben. Az 1990-es évek elején és közepén mindössze néhány ember írt blogokat, mára a bloggolás mint jelenség, uralkodó irányzattá lett. Népszerűsége olyan nagymértékű, hogy a New York Times oldalain is jelent meg e témával foglalkozó írás, mely a saját blogokat készítő főiskolai hallgatókra összpontosított. Mostanában, hogy a demokrata vezetők a figyelem középpontjában állnak, minden jelölt rendelkezik legalább egy hivatalos webloggal. A hivatásosok és a vezető politikai tudósítók saját blogokat készítettek, amelyekben nyilvántartják a jelöltek naplóinak állításait és többnyire elemezik is őket. Múlt hónapban a Zope alapú COREBlogot vettük szemügyre, ami rendkívül megkönnyítheti saját weblogunk elkészítését. Természetesen a COREBlog használatához rendelkezniünk kell egy Zope-példánnyal, továbbá az sem árt, ha tudjuk telepíteni és módosítani is a termékeket. Nem mindenkinek adatik meg az ilyen luxus, ugyanakkor a Perl nyelven írt CGI programok futtatását szinte minden webszolgáltató engedélyezi. Ebben a hónapban a Perl nyelven írt és CGI programként futtatható Blosxom ([↗ http://www.blosxom.com](http://www.blosxom.com)) rendszerrel foglalkozunk. A Blosxomot *Rael Dornfest*, az O'Reilly egyik programozója készítette. Első ízben, mint bloggólásra aligha alkalmas eszközről írtam a Blosxomról, feltételezve, hogy kis mérete arányban áll képességeivel. A Blosxom ereje azonban nemcsak komoly képességekészletében rejlik, hanem fő vonzereje, hogy e képességek összeválogatását és keverését egyaránt lehetővé teszi.

Telepítés

Webkiszolgálók terén kicsit is jártasak számára a Blosxom telepítése gyerekjáték. Egyetlen, Perl nyelven írt CGI programot tartalmaz. Például nekem nem kellett mást tennem, mint a *blosxom.cgi* fájlt bemásolni a */usr/local/apache/cgi-bin* könyvtárba, és már működött is minden.

Természetesen minden programnak szüksége van némi állítgatásra, ez alól a Blosxom sem kivétel. Az összes beállítási feladatot a program elején található Perl változókon keresztül végezhetjük el. A megjegyzések az egyes változók célját viszonylag jól meghatározzák. Nálam a következő Blosxom változók átírására volt szükség:

- `$blog_title`: a weblog címe, ezt láthatják a felhasználók és az RSS hírügynökségek.

- `$blog_description`: a blog leírása, ami az RSS listában fog megjelenni.
- `$datadir`: a Blosxom weblog valamennyi bejegyzése tulajdonképpen nem más, mint a lemezen valahol elhelyezett szöveges állomány; a `$datadir` változó mondja meg, hol találjuk meg őket.

E három elem beállítása után weblogom máris működőképes volt.

Bejegyzések felvitele

A Blosxom kipróbálását egy *introduction.txt* nevű egyszerű szöveges állománnyal kezdtem a *\$datadir* könyvtárban.

Ez egy tesztbejegyzés.

```
<p>hello!</p>
```

Weblognak ugyan nem túl izgalmas bejegyzés, de annál érdekesebb, ahogy mindez meg fog jelenni a weblogon, kiegészítve egy dátum előtaggal, a szöveg utáni időbélyeggel, a maradandó hivatkozással, az első sor vastagon szedve, mintha főcím vagy cím lenne.

Más szavakkal, a Blosxom weblog rendszerébe új bejegyzéseket felvinni egyszerűen csak annyit jelent, hogy az adatkönyvtárban fájlokat hozunk létre. Valamennyi, a `$file_extension` változóban megadott kiterjesztésre végződő (azaz alapértelmezés szerint *.txt*) állomány weblogbejegyzésnek számít. Ebből következik, hogy például az Emacs ~ végződésű mentései nem számítanak bejegyzésnek.

Ha hozzám hasonló módon gyakran szoktuk menteni, amit írunk, esetleg furcsának találjuk majd, hogy weblogunk írás közben változik, és ezt az egész világ azonnal láthatja is.

Amennyiben a háttérben szeretnénk dolgozni, mindaddig hagyjuk le a *.txt* kiterjesztést az állományainkról, amíg nem kívánjuk kiadni őket.

Saját gépemem a Blosxomot a fő */cgi-bin* könyvtárba telepítettem, így a Blosxom blogokat a *http://localhost/cgi-bin/blosxom.cgi* címen tudom elérni.

A Blosxom a bejegyzésekhez tartozó dátumot és időt a fájl létrehozási dátumából veszi. Minthogy az állományt február 11-én, délután négy órakor készítettem, a weblogbejegyzés is ehhez az időhöz tartozik. Ez egyben azt is jelenti, hogy a `touch` parancs segítségével megváltoztathatjuk a bejegyzéshez tartozó dátumot:

```
touch -t 200401011500 testing.txt
```

A fenti parancs a *testing.txt* állomány dátumát 2004. január 1-jén három órára változtatja. (Amennyiben a *testing.txt* állomány még nem létezne, létre is hozza.) Bár ez némiképp ellenkezik a weblogetikett szokásaival, nyilvánvalóan megoldható feladat.

Kicsit érdekesebb, hogy ugyanezzel a paranccsal a weblog-bejegyzés dátumát a jövőbe is állíthatjuk. Amennyiben a `$show_future_entries` változó értéke 1, az ilyen jövődőlbeli dátummal rendelkező bejegyzések mindig láthatóak. Az alapértelmezett beállítás esetében azonban ezek a bejegyzések csak akkor jelennek majd meg, amikor a valódi dátum megegyezik a hozzájuk rendelttel. Ez azt jelenti, hogy bejegyzéseinket egy adott dátumra és időpontra időzíthetjük.

Jellemzők (flavours)

Ha a Blossom csak ennyit tudna nyújtani, nem lennék túlságosan lenyűgözve. De ha kicsit közelebbről szemügyre vesszük, kiderül, hogy rendkívüli képességekkel bír. Ezeket a képességeket a megjelenítő sablonoknak köszönheti (amelyeket brit helyesírás szerinti „favour”-nak, jellemzőkészletnek nevez), valamint annak, hogy tetszőleges számú bővítmény (plugin) futtatására képes. A Blossom rendszert e két képessége rendkívül jól bővíthetővé teszi.

A Blossom rendszert két beépített jellemzővel kapjuk: az alapértelmezett HTML jellemzőkészlettel és az RSS hírügynökséghez szánt, választható RSS jellemzőkkel. Magunk is megtekinthetjük az RSS formátumot, ha a blog URL végére beszurjuk a `?flav=rss` kiegészítést. Ha tehát egyébként a `http://localhost/cgi-bin/blosxom.cgi` címen érjük el webnaplónkat, az RSS alakot a `http://localhost/cgi-bin/blosxom.cgi?flav=rss` kifejezéssel tekinthetjük meg. Egyébként a kért dokumentum végződésének megváltoztatásával is hozzáférhetünk a különféle jellemzőkészletekhez. Így például a `http://localhost/cgi-bin/blosxom.cgi/index.rss` címen az RSS alakot találjuk.

A Blossom weboldalán teljes jellemzőkészlet-tárat találunk. Az alapötlet nagyon egyszerű: az adatkönyvtárunkban, a weblogbejegyzéseink mellett készítünk egy HTML állományt, amelyet a megváltoztatandó Blossom-kimenet nevével azonos módon nevezünk el.

Az állomány utótagja a módosítandó jellemzőkészlet nevével azonos. A *header.html* tehát a weblog HTML jellemzőkészletének *header* elemét változtatja meg, míg a *date.blah* a *blah* jellemzőkészlet dátummegjelenítését befolyásolja. A felhasználók az URL-ben megadhatják a kívánt jellemzőkészlet név-érték-párosát (amint azt az imént is láthattuk), az alapértelmezett értéket pedig magában a *blosxom.cgi* fájl `$default_flavour` változójával állíthatják be. Minthogy a blogbejegyzések „txt” utótagra végződnek, txt nevű jellemzőkészletünk nem lehet.

Minden egyes jellemzőfájl apró HTML-részletet tartalmaz, mely az adott állományban megjelenítendő Perl-változónevekkel van megtűzdelve. Például a *story* jellemzőfájlok többek közt két értéket kapnak a `$title` és a `$body` változóknál. (A teljes listát a Blossom weboldalán találjuk.) Ezzel a módszerrel a törzsszöveg előtt álló fejléceket könnyedén nagyobb méretűre és jobbra igazítottakra lehet alakítani:

Az egotrip.pl

```
#!/usr/bin/perl

use strict;
use warnings;
use diagnostics;

package egotrip;

# 1-et adunk vissza, jelezve, hogy a bővítmény
# aktív
sub start
{
    return 1;
}

# Kiemeli a nevemet
sub story {
    my ($pkg, $path, $filename, $story_ref,
        $title_ref, $body_ref) = @_;

    $$body_ref =~ s|Reuven|<b>Reuven</b>|g;
    1;
}

1;
```

```
<p>
<h1 align="right">$title</h1>
<br />
$body
</p>
```

A fenti „flavour” blogunk tartalmát változtatás nélkül helyettesíti be a `$body` változó helyére. Ez módszer egész használható, feltéve, hogy a blog szerzője ismeri a HTML nyelvet és a bekezdéshatárokat hajlandó kézzel begépelni. Ha viszont azt szeretnénk, hogy az emberek a bekezdéshatárokat üres sorokkal jelölhessék, írásukon végig kell futtatniunk valamilyen programot. Szerencsére a Blossom rugalmas bővítménykezelője megkönnyíti az ilyen programok elkészítését.

Bővítmények

Valamennyi bővítmény egy-egy Perl program, amelyet a `require` függvény olvas be és hajt végre. Például a `require foo.pl` a *foo.pl* állományt nyitja meg, majd a benne lévő kódot végrehajtja. Általában azt szoktam javasolni, hogy kerüljük a `require` használatát, és inkább az `use` kulcsszót alkalmazzuk, amely számos parancsot képes végrehajtani, többek közt a `require` utasítást is. Csakhogy a `require` futásidőben értékelődik ki, míg a `use` a fordítás során hajtódik végre, jelen esetben tehát sokkal egyszerűbb ezzel dolgoznunk.

A Bloxom feltételezi, hogy minden állomány, amely a tetszőlegesen megadható `$plugin_dir` változó által megadott könyvtárban található, valóban bővítmény. A bővítményeket ábécérendben tölti be és hajtja végre, ha tehát valamely bővítményt hamarabb vagy később szeretnénk futtatni, előfordulhat, hogy át kell neveznünk.

A bővítmények nem többek egy egyszerű Perl programnál, amelyek egy vagy több alprogramot határoznak meg. Minden bővítménynek tartalmaznia kell egy *start* alprogramot, amely egyszerűen 1-et ad vissza. A Bloxom ennek segítségével tudja megállapítani, hogy a bővítmény élő-e, és készen áll-e a futásra. Számos egyéb bővítmény alprogram létezik, ezeket valamennyi bővítményben megadhatjuk, kezdve a bejegyzésektől (entries), amelyek bejegyzések listáját adják vissza, egészen a történetig (story), ennek segítségével pedig történeteink tartalmát módosíthatjuk. A feladatok ilyesfajta lebontásával a Bloxom hihetetlen mértékű kifinomult testreszabási lehetőséget kínál, miközben a magkód továbbra is tömör és hatékony marad.

Milyen képességeket nyújthatnak a bővítmények? Úgy tűnik, tényleg csupán néhány köztötség létezik. A weblogbejegyzések forrását, a módszert, ahogy e bejegyzéseket szűrjük, a bejegyzések megjelenítéséhez használt sablonokat, a bejegyzéseket, illetve akár a bejegyzések tartalmát egyaránt megváltoztathatjuk.

A Bloxom weboldalán rengeteg bővítményt találunk. Akadnak közöttük olyanok, amelyek más bővítményektől függenek, míg mások, például a naptár (calendar) csak akkor jelennek meg, ha a bővítményt támogató jellemzőkészletet használjuk. Más bővítmények azonnal kifejtik hatásukat, egyszerűen csak be kell dobnunk őket a bővítmény-könyvtárunkba.

Egyszerű példa az önmagában is használható bővítményre az atomfeed, amely az Atom hírügynökség-változatot támogatja. Az Atom az RSS vetélytársa, amit néhány nehézsúlyú bloggoló és programozó támogat, nem utolsósorban az RSS világában mostanában megjelent egymással versengő szabványok miatt. Amennyiben Atom-kimenetet is szeretnénk, nem kell mást tennünk, egyszerűen másoljuk az atomfeed bővítményt a bővítmény-könyvtárunkba. Saját Atom-változatunkat a `http://localhost/cgi-bin/bloxom.cgi?flav=atom` vagy a `http://localhost/cgi-bin/bloxom.cgi/index.atom` URL-re hivatkozva kaphatjuk meg.

Bővítmények készítése

Listánkon egy egotrip névre keresztelt egyszerű szűrőt találunk, amely a nevet annyszor szedi ki vastagbetűvel, ahányszor előfordul a weblogbejegyzésben. Figyeljük meg, hogy a bővítménynek saját csomagot kell megadnia; ezzel biztosítjuk, hogy valamennyi bővítmény alprogramjai külön névtérben maradjanak, valamint a Bloxom el tudja dönteni, hogy a bővítmény tartalmazza-e az adott modult. A tényleges munkát a *story* alprogramban végezzük, amit a Bloxom hat értékkel hív meg, a bejegyzésben felhasználható elemek számának megfelelően. Esetünkben kizárólag a bejegyzés törzsének módosítására összpontosítunk, amit az utolsó, `$body_ref` nevű változóban találunk. Mint a neve is mutatja, a változó skalár mutató, azaz tartalmát visszahivatkozással érhetjük el, a kettős `$$` (dollárjel) karakter hasz-

nálatával. Ezt figyelembe véve, nem lesz nagy meglepetés, hogy a nevem előfordulásait a következő sorral állíthatjuk vastag betűsre:

```
$$body_ref =~ s|Reuven|<b>Reuven</b>|g;
```

Természetesen ezt a lépést kicsit kifinomultabbra is tervezhetjük, például önműködő hiperhivatkozásokat illetve a különféle elemekhez:

```
$$body_ref =~ s|(Reuven Lerner)|
<a href="http://www.lerner.co.il/">$1</a>|g;
$$body_ref =~ s|(Linux Journal)|
<a href="http://www.linuxjournal.com/">$1</a>|g;
```

Valójában létezik is ilyen bővítmény; ez a közösség által vezetett Wikipedia hivatkozásait kezeli. Bármely szöveg, amelyet [[szögletes zárójel]] közé helyezünk önműködően hálózati referencia könyvre mutató hivatkozássá alakul. Figyeljük meg, hogy a jellemzőkészletek olyan HTML-sablonok, amelyekbe Perl-változókat helyettesíthetünk miközben a bővítmények Perl programok. E kettősség megértése a megjelenítés és a műveletek között némi időt vehet igénybe, de nem túl bonyolult feladat.

Mint azt a korábbi bekezdés-elvásztó példánk is mutatja, semmi szükség rá, hogy újra feltaláljuk a spanyolviaszt. Egyszerűen csak letöltjük a megfelelő bővítményt, nevezetesen a Bloxot, amely a bekezdéseket üres sorok alapján képes elválasztani blogbejegyzéseinkben. A bővítmény aztán kívánság szerinti HTML-elemekkel választja el a bekezdéseket. A Blox szerepel a Bloxom bővítménytárában is (lásd a hálózati források részt).

Mint említettem a Bloxom valamennyi bejegyzést és jellemzőkészletet egyazon könyvtárban tárol. Ez számomra kicsit zavaró és méretezhetőségi kérdéseket is felvet. Még ha a fájlrendszerem és Perl-csomagom gond nélkül kezel is ennyi állományt, biztos, hogy mindig valamennyin keresztül szeretnék gázolni? Amikor ez valóban kezd gondot jelenteni, az *entries* bővítmény siethet segítségünkre, amely külön könyvtárakból keresi vissza az állományokat, s a megfelelő hozzárendeléseket továbbítja a Bloxom-magnak.

Összegzés

A Bloxom hatékony weblogkészítő eszköz; sokkal többet nyújt, mint első pillantásra látszik. A Bloxom tartalmaz egy könnyen telepíthető, játszói módon beállítható, Perl nyelvű CGI programot, de igazi ereje mégis abban rejlik, hogy jellemzőkészletek (flavours, megjelenítő sablonok) és bővítményeljárások megfelelő kombinációjával a megjelenítés tetszőleges részét lecserélhetjük.

Linux Journal 2004. május, 121. szám



Reuven M. Lerner (reuven@lerner.co.il) Veterán web, illetve adatbázis programozási tanácsadó, jelenleg végzős tanárhallgató az evanstoni Northwestern University Egyetemen, Illinois államban.