Planet Me blog aggregátor létrehozása

Bemutatjuk, hogyan hozhatunk létre saját blog aggregátort (hírolvasót) a „Planet” oldalakat (például a Planet Apache-ot) működtető kóddal.

A *Planet Projecttel* az online közösségek összegyűjthetik a közösség tagjainak blogjait egy központi weboldalon. A *Planet GNOME* és a *Planet Apache* közösségi blogok is a *Planet* kódon alapulnak például. A *Planet* kód ilyen online felhasználása nem kerül sokba, viszont az emberek szemmel tarthatják a közösséget. A cikkben bemutatjuk, hogyan hozható létre a *Planet* kóddal saját személyes blog hírolvasó gépünkön.

Telepítés

A *Planet* kódhoz a *Python 2.2*-re vagy későbbi változatára van szükség. A *Planet* telepítése úgy a legegyszerűbb, ha letöltünk a ➲ *planetplanet.org*-ról egy előző esti „pillanatfelvétel” (snapshot) tarballt és kicsomagoljuk saját könyvtárunkban. Én általában átnevezem a kitömörített könyvtárat, hogy szerepeljen a nevében a letöltés dátuma, és létrehozok rá egy hivatkozást, mint az aktuális *Planet Me* változatra.

Néhányszor saját könyvtáram elérési útvonalára hivatkozom a cikkben, ne felejtsük el azt saját adatainkkal helyettesíteni.

Az *1. Lista* két utolsó parancsa minta a hírcsatornák lehívására és kezdő *Planetünk* beállítására. Az utasítások változhatnak aszerint például, hogy kell-e proxy szervert használnunk az internet-hozzáféréshez. A parancsok futtatása után kell találnunk egy *Planet Me* nézetet böngészőnkben a *~/planet/me/index.html* címen. E lépések végrehajtása után *Planetünknek* az *1. ábrához* kell hasonlítania.

Nyilván be akarjuk állítani a megtekintendő hírcsatornákat. Ezt a *me-meta/config.ini* végén tehetjük meg. A beállításfájl szögletes zárójelek közé zárt szöveggel jelöli a szakaszokat. Egy-egy szakasz meghatározását *key=value (kulcs=érték)* párokként megadott opciók követik. Minden figyelendő bloghoz új szakaszt kell létrehoznunk, melynek neve az *RSS* csatorna *URL*-je. A *2. Listában* látható példa az alapértelmezett *config.ini* fájl egy részletét mutatja.

A fejlécben látszódik majd a blog neve, ahonnan a bejegyzések származnak, az arckép pedig a jobb oldalon jelenik meg, ha az alapértelmezett *HTML* sablonokat használjuk. Az arcszélesség és az arcmagasság alapértelmezés szerint szabadon beállítható. *Planet Me* oldalunk témaköreinek (*topic*) csinosítására felhasználható ikonokat számtalan oldalon találunk. A *3. Listában* például az egyik *Slashdot* szekció ikont használom (lásd az online forrásokat) a *Slashdot RSS* híreihez.

Ha a *Planet* beállításokat a cikk alapján csináljuk, a témaikonok helye a *~/planet/me/images* lesz. Saját *Slashdot* témaikon beállításom a *3. Listában* látható.

1. Lista A Planet telepítése

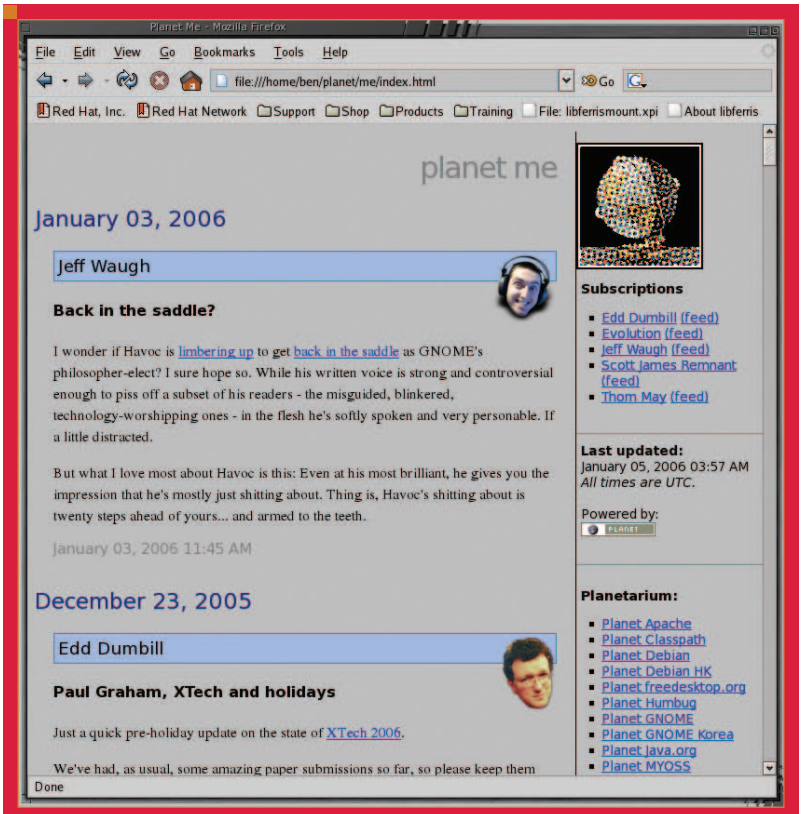
```
$ cd ~
$ tar xjvf planet-
➥nightly.tar.bz2
$ planetdated=planet-$(date
➥+'%d%b%y')
$ mv planet-nightly
➥$planetdated;
$ ln -s $planetdated planet
$ cd planet
$ cp -av fancy-examples
➥me-meta
$ cd me-meta
$ cp ../examples/*.xml* .
$ edit config.ini
name = Planet Me
link = file://home/ben/planet/
➥me/index.html
owner_name = John Doe
owner_email = root@localhost
# lejjebb a fájlban
# a sablonfájloknak egy sorban
# kell lenniük
template_files = me-meta/
➥index.html.tmpl
me-meta/rss20.xml.tmpl
➥me-meta/rss10.xml.tmpl
me-meta/opml.xml.tmpl
➥me-meta/foafroll.xml.tmpl
# lejjebb a fájlban módosítjuk
# fancy-examples/
# index.html.tmpl
[me-meta/index.html.tmpl]
items_per_page = 30
$ cd ..
$ mkdir cache
$ ln -s output me
# proxy nélkül
$ python planet.py
➥me-meta/config.ini
# szabványos squid proxyval
# a "dairiserver" gazdagépen
$ http_proxy=http://
➥dairiserver:3128/ \
  python planet.py
  ➥me-meta/config.ini
```

2. Lista Példa hírolvasó meghatározására

```
[http://www.gnome.org/~jdub/
➥blog/?flav=rss]
name = Jeff Waugh
face = jdub.png
facewidth = 70
faceheight = 74
```

3. Lista
Így szerzünk képeket a Slashdotról

```
$ cd ~/planet/me/images/
$ wget  \
http://images.slashdot.org/
➥topics/topicslashback.gif
# az ImageMagick convert
# parancsával alakítsuk át
$ convert topicslashback.gif
➥slashdot.png
```



1. ábra A Planet telepítése

4. Lista A Slashdot ikon haszálatának meghatározása

```
$ edit ~/planet/me-meta/
➥config.ini
[http://rss.slashdot.org/
➥Slashdot/slashdot]
name = Slashdot
face = slashdot.png
$ cd ~/planet
$ python planet.py
➥me-meta/config.ini
```

5. Lista Cron feladat a blogok összegyűjtéséhez

```
$ mkdir -p ~/mycron
$ cd ~/mycron
$ vi upd-planet.sh
#!/bin/sh
cd ~/planet;
http_proxy=http://dairiserver:
➥3128/ \
  python planet.py
  ➥me-meta/config.ini
$ chmod +x upd-planet.sh
$ echo \
'00 04  *  *  * /home/ben/
➥mycron/upd-planet.sh' \
 >|upd-planet.cron
# csak ha a cront a ~/mycron
# könyvtáron kívülről
# használjuk
$ crontab -l >|oldcrontab.cron
$ cat *.cron >|newtab
$ crontab newtab
$ rm -f oldcrontab.cron
```

A *4. Lista* a ***config.ini***-hez csatolandó új szakasz, amely beépíti a ***Slashdot*** ikont saját ***Planet Me***-nkbe.

Dinamikus tartalom

A blogok összegyűjtéséhez és a begyűjtendő blogok listájának egyszerű módosításához most futtatnunk kell a ***Planet*** kódot.
Az ütemezett begyűjtéshez a ***cron***-t használhatjuk. Az *5. Listából* megtudjuk, hogyan frissíthetjük a ***Planet Me***-t napi (éjjeli) rendszerességgel.

Könnyedén felvehetünk és eltávolíthatunk blogokat, ha egy blog-meghatározás fájl listát használunk magának a beállításfájlnak kézzel történő módosítása helyett. Erre a célra használható a *6. Listában* szereplő ***generate-config*** (beállítás-készítő) szkript, amely a blognevet és az ***URL***-eket egyszerű fájlokba helyezi a blog alkönyvtárban. Beállításfájlokat parancssorban vagy valamely fájlkezelőben is felvehetünk és eltávolíthatunk, de használhatunk akár egy ***Firefox*** kiterjesztést is, amellyel menüből adhatunk új RSS csatornákat a ***Planet Me***-hez. Az archívumok kezelése (ezt később bemutatjuk) is egyszerűsödik, ha a bloginformációkat ***config.ini***-n kívül tároljuk.

A kinézet módosítása

Két fájl szabályozza a ***Planet*** kinézetét: a ***me-meta/index.html.tmpl*** és a ***me/planet.css***. Az előbbi az oldaltartalom sablonja, utóbbi pedig a ***CSS*** stíluslap.

6. Lista Az összegyűjtendő blogokat meghatározó fájlok létrehozása

```
$ cd ~/planet/me-meta
$ mv config.ini
➥config.ini.template
$ edit config.ini.template
# töröljünk minden blog URL
# részt a fájl végéről
# keressünk a http: -ra
$ mkdir blogs
$ echo http://
➥rss.slashdot.org/Slashdot/
➥slashdot \
  >blogs/slashdot.blog
$ ./generate-config
```

7. Lista Fájlok használata beállítások létrehozására

```
#!/bin/sh
cp -av config.ini.template
➥config.ini
for if in blogs/*.blog
do
        base=$(basename
        ➥$if .blog);
        content=$(cat $if);
        echo "" >>
        ➥config.ini
        echo "[$content]"
        ➥>> config.ini
        echo "name = $base"
        ➥>> config.ini
        echo "face =
➥$base.png" >> config.ini
done
```

Alapértelemezésként az arc, a bejegyzés, a dátum és az oldalsáv meghatároz stílusokat, melyek a stíluslappal testre szabhatók. Saját betűtípusokat is használhatunk, mindössze a ***font-family CSS***-címkét kell módosítanunk. Az ***index.html.tmpl*** sablon további címkéket tartalmaz, melyeket a ***Planet*** kód a végleges ***index.html*** file létrehozásához használ. A legfontosabb címkék a `TMPL_LOOP`, a `TMPL_IF` és a `TMPL_VAR`. A hírcsatornákat a `<TMPL_LOOP Items>` ***HTML***-szerű címkével és a megfelelő zárócímkével

8. Lista Feltétel vizsgálata a TMPL_IF címkével a megjelenítéshez

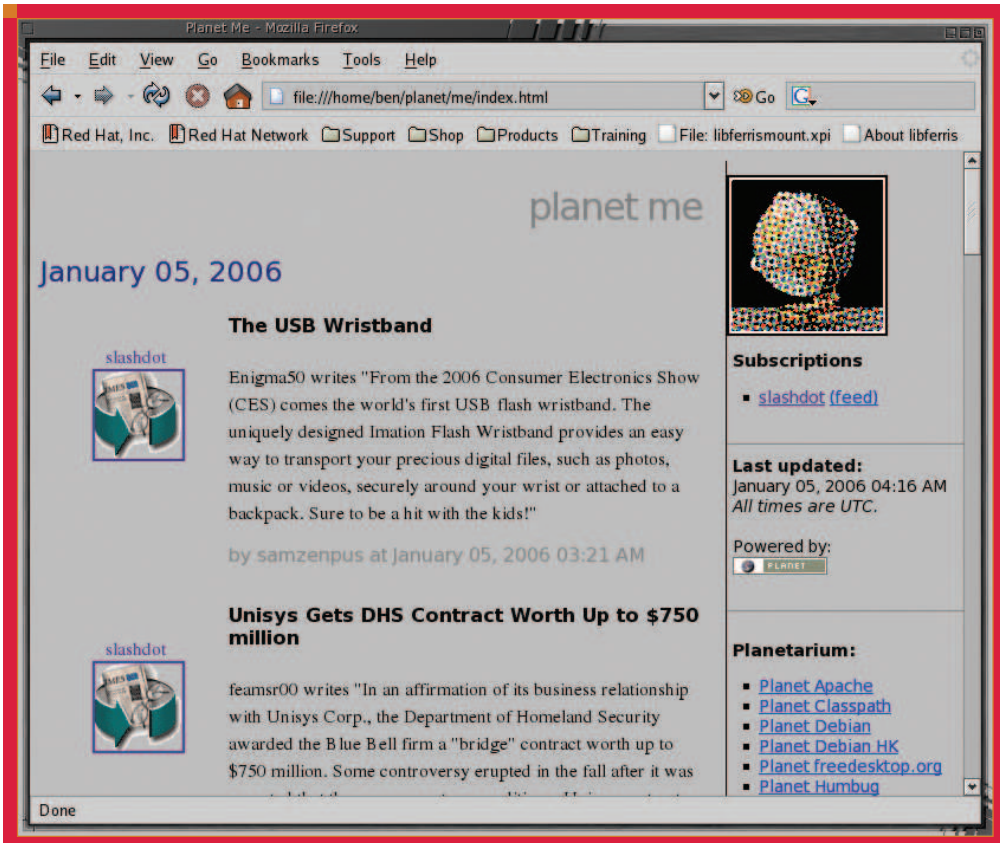
```
<TMPL_IF title>
<a href="<TMPL_VAR link
➥ESCAPE="HTML">">
<TMPL_VAR title>
</a>
</TMPL_IF>
```

helyezzük a kimeneti oldalra. Ezen címkék közötti ***HTML*** elemek megjelennek minden megjelenítendő bejegyzéssel. Ezek határozzák meg, mi és hogyan jelenjen meg az egyes bejegyzésekkel.
Ezzel a változók jelölik a hírcsatornák tartalmának helyét. A ***Planet*** kód a `<TMPL_VAR title>` címkét például az aktuális hírelem címével helyettesíti. A `TMPL_VAR`-nak nincs zárócímkéje. A `TMPL_IF` címke bizonyos feltételek és adatok meglétét ellenőrzi. Például néha a híreknek nincs címük.
A *8. Listában* szereplő kód kiírja a címadatokat, ha vannak, és nem ír ki semmit, ha nincsenek. A `TMPL_VAR` címke `escape` attribútuma miatt a ***Planet*** ellenőrzi, hogy a hivatkozásváltozó értéke megengedett ***HTML*** attribútum formátumú-e.
A ***me-meta/index.html.tmpl***-t és a ***CSS*** fájlokat kell szerkesztenünk a csatornaikonnak a hírelemek bal szélére való mozgatáshoz. Alapértelmezésként az ***index.html.tmpl*** csak akkor jeleníti meg a csatornaikont, ha az aktuális hírelem más csatornáról származik, mint az azt megelőző.
Az ***index.html.tmpl*** arcképet megjelenítő része körül töröltem a `<TMPL_IF new_channel>` címkéket, ahogy azt a *9. Listában* látható részlet is mutatja. A csatorna kép ***CSS*** osztályát `news-item-icon`-ként, a hír fő részét `news-item`-ként, az aktuális csatornához tartozó képét pedig `embedded-face`-ként határoztam meg.
A *10. Listában* található új stíluslap kóddal beállítjuk a csatorna képet a hírelem bal oldalán. ***Planet Me***nknek most a *2. ábrához* kell hasonlítania. Ha a jobb oldalra szeretnénk tenni az ikonokat, változtassuk

9. Lista Új csatorna elemek szakasz a ~/planet/me-meta/index.html.tmpl-hez

```
<TMPL_LOOP Items>
<TMPL_IF new_date>
<h2><TMPL_VAR new_date>
➥</h2>
</TMPL_IF>
<div class="news-item-
➥icon">
 <a href="<TMPL_VAR
 ➥channel_link ESCAPE=
 ➥"HTML">"
    title="<TMPL_VAR
    ➥channel_title
ESCAPE="HTML">">
 <TMPL_VAR channel_name>
 <br/>
 <img class="embedded-face"
 ➥alt=""
src="images/<TMPL_VAR
➥channel_face ESCAPE=
➥"HTML">" >
 </a>
</div>
<div class="news-item">
 <TMPL_IF title>
 <h4><a href="<TMPL_VAR link
 ➥ESCAPE="HTML">">
        <TMPL_VAR title>
        ➥</a></h4>
 </TMPL_IF>
 <div class="entry">
 <p>
 <TMPL_VAR content>
 </p>
 <p class="date">
 <a href="<TMPL_VAR link
 ➥ESCAPE="HTML">">
<TMPL_IF creator>by <TMPL_VAR
➥creator> at </TMPL_IF>
<TMPL_VAR date></a>
 </p>
 </div>
</div>
</TMPL_LOOP>
```

meg úgy a stíluslapot, hogy a `news-item-icon float` címkéje `right` legyen, a hírelem `margin-left`-je pedig `0px`.
A csatorna meghatározás fájlban használt `face=akármi.png` sor nem feltétlenül szükséges. Bármilyen más változót megadhatunk az egyes



2. ábra Saját Planet Me oldalam testreszabott sablonnal és CSS fájllal

10. Lista A ~/planet/me/planet.css-hez adandó új stílusok

```
div.news-item-icon {
      float: left;
      position: relative;
      left: 4px;
      margin-top: 25px;
      padding: 0 20px 30px 0;
      width: 120px;
      text-align: center;
}
div.news-item-icon a {
      text-decoration: none;
}
div.news-item {
         margin-left: 140px;
}
```

csatornákhoz, s azok az *index.html.tmpl*-ben elérhetők lesznek. A *11. Listában* például a `foo` változó szerepel, melyet talán egy csatorna `foo=bar`-jaként adunk meg a csatorna leírása után a *config.ini* fájlban.

11. Lista A TMPL_IF-fet így is használhatjuk

```
<TMPL_IF channel_foo>
Have foo:<TMPL_VAR
➥channel_foo ESCAPE="HTML">
</TMPL_IF>
```

A *Planet Me* testreszabását más *Planet* weboldalak tanulmányozásával is elsajátíthatjuk: *HTML* és *CSS* fájljaikból megtudhatjuk, hogy ők hogyan módosították a kinézetet.

Archívumok tárolása és megtekintése

A *Planet* képes sok forrásból hírcsatornákat összegyűjteni és azokat visszamenőlegesen megjeleníteni egy oldalon. *Planet Me*-vel való helyi használat esetén egy hírcsatorna egy korábbi időintervallumban megjelent hírei is megtekinthetők.
A *Planet Me* létrehoz egy érvényes *RSS RDF* hírcsatornát, melyet felhasz-

12. Lista Hírcsatornák ismételt archiválása

```
$ cd ~
$ unzip Jena-2.3.zip
$ edit ~/.bashrc
# a classpath változó
# beállítása
JenaSetup() {
    for if in ~/Jena-2.3/
    ➥lib/*.jar; do
        export CLASSPATH=
        ➥$CLASSPATH:$if;
    done
}
$ . ~/.bashrc
$ JenaSetup
# hírcsatorna archiválása
# három ismételhető lépés
$ cd ~/planet/me
$ mv -f archive.xml
➥rss10-archive.xml
$ java jena.rdfcat rss10*xml
➥>archive.xml
```

13. Lista
Keresés Planet Me oldalunkon Jenával

```
$ cat rss-by-date.sparql
PREFIX dc:
➥<http://purl.org/dc/
➥elements/1.1/>
PREFIX xsd:
➥<http://www.w3.org/2001/
➥XMLSchema#>
DESCRIBE ?channel ?bnode
➥?a WHERE
{
 ?channel ?items ?bnode .
 ?bnode ?hasitem ?a .
 ?a dc:date ?date .
 FILTER ( xsd:dateTime(?date)
    >= xsd:dateTime
    ➥("2006-01-03T00:00:00")
 && xsd:dateTime(?date)
    <= xsd:dateTime
    ➥("2006-01-05T00:00:
    ➥00") )
}
$ cd ~/planet/me
$ java jena.sparql -data
➥archive.xml \
 -query rss-by-date.sparql
 ➥-results RDF/XML \
 >my-query-result.rss
```

nálhatunk ***Planetünk*** archiválására. Az ***RDF*** fájlok tartalma hármas csoportokat képez. A hármas csoportok tagjaira mint alanyra, állítmányra és tárgyra hivatkozunk. Egy hármas jelentheti azt, hogy a hírelemnek van egy megjelenési dátuma, például: `item57 has-date 3-Jan-2006`. Egy ***RSS*** hírcsatorna meghatároz egy hírcsatornát, hozzárendel híreket, minden hírhez pedig olyan tulajdonságokat, mint a címe, közreadásának időpontja és szöveges tartalma. A `has-date`-et és hasonlókat hosszú ***URI***-k írják le, hogy ne lehessen két hármasnak azonos karakterlánc értéke.
Könnyen, de egyszersmind hatékonyan archiválhatjuk ***Planetünk RSS***-ét a ***Jena Projecttel***. Ha van telepítve Java virtuális gépünk, akkor a ***Jena*** telepítéséhez csak egy tarballt kell letöltenünk, kitömörítenünk és a ***classpath***-hoz (osztályútvonal) adnunk.
A *12. Listában* a telepítés lépései és a hírcsatornák archiválásának

14. Lista
Jena lekérés eredményekkel módosíthatjuk Planet Me hírolvasónkat

```
$ cd ~/planet/me-meta
$ cp -av config.ini.template
➥config.ini
$ echo \
"[file:///home/ben/planet/
➥me/my-query-result.rss]"
➥ \
  >>config.ini
$ echo "name = archive"
➥>>config.ini
$ cd ~/planet
$ rm -f cache/file.home*
$ python planet.py
➥me-meta/config.ini
```

ismétlődő folyamata látható. Saját hírcsatornánk archívumát ***Jenával*** egy adatbázisba helyezhetjük, ha hosszú idő alatt tetemes mennyiségű csatornát halmoznánk fel.
Jenával igen hatékony lekérdezéseket hajthatunk végre az archívumunkban, és könnyen újraépíthetjük a ***Planetünket***.
A *13. Listában* egy egyszerű, időintervallum alapú hírcsatorna-lekérés látható. A lekérés ***SPARQL*** lekérdezőnyelven íródott, ez használatos ***RDF*** tárolókból való lekérésekhez.
A lekérés hivatkozik a csatornára, a hír- és a dátumelemre, mielőtt egy szűrőt alkalmazna arra, hogy a hírelem dátuma alapján melyik hírelemet kell visszaadnia.
Most már könnyedén megváltoztathatjuk ***Planet Me*** oldalunkat, hogy bemenetként csak a saját lekéréseinkből vett eredményeket használja.
Lásd a *14. Listát*: a blog ***URL***-jeit és metaadatait a fenti leírás alapján külön fájlokba helyeztük.
A fenti lekéréssel a csatornaikon ugyanaz marad, mert egyetlen hírcsatornát kérdezünk le: a sajátunkat. A `regex()` egy másik kifejezés, amelyet a `FILTER` szakaszban használhatunk. A *15. Lista* kódja az összes hírelemet szűri, és csak azokat mutatja meg, amelyek megfelelnek a kis- és nagybetűket meg nem különböztető szabályos kifejezésnek.

15. Lista
Szabályos kifejezéseket is használhatunk szűrőként

```
PREFIX dc:
➥<http://purl.org/dc/
➥elements/1.1/>
PREFIX xsd:
➥<http://www.w3.org/2001/
➥XMLSchema#>
PREFIX rss:
➥<http://purl.org/rss/1.0/>
PREFIX content:
➥<http://purl.org/rss/1.0/
➥modules/content/>
DESCRIBE ?channel ?bnode ?a
➥WHERE
{
 ?channel ?items ?bnode .
 ?bnode ?hasitem ?a .
 ?a content:encoded ?
 ➥content .
 FILTER ( regex(?content,
 ➥".*product.*", "i") )
}
```

Összegzés

A ***Planet Me*** online közösségi blog aggregátornak készült (tehát eredetileg nagy látogatottsághoz tervezték), de a ***Planet*** kód alapján igen hatékony blog hírolvasót hozhatunk létre személyes használatra is. Némi munkát igényel ugyan, de megéri, hiszen az eredmény: egy effektív, személyes blog hírolvasó, ahol a közösséget mi magunk alakíthatjuk, valamint archiválhatunk és egy hathatós lekérdezőnyelv segítségével kereshetünk is a régi hírek között.

Linux Journal 2006., 144. szám

Ben Martin

Virtuális fájlrendszerek (libferris) létrehozásával és a bennük való adatbányászattal tölti szinte minden idejét. Most e fájlrendszerek kiterjesztésén dolgozik, hogy az Emacs és Firefox fájlrendszerekként csatolhatók legyenek.

KAPCSOLÓDÓ CÍMEK

➲ www.linuxjournal.com/article/8830