

OpenACS-csomagok

Webalapú alkalmazások csomagjainak telepítése, megosztása és eltávolítása az APM segítségével.

A programfejlesztés egyik alapötlete szerint a nagyobb feladatokat kisebb modulokra kell bontani. A modularizálás egyszerűbbé teszi a rendszer egyedi igényekhez történő igazítását, hiszen megoldhatóvá válik a kívánt modulok beillesztése és a szükségtelenek eltávolítása. A modulok használatával a munka is könnyebben megosztható több programozó között. Egy gyors pillantás az Interneten jelenleg ingyenesen elérhető Linux-, Apache-, Perl- és Python-modulokra azonnal rávilágít a módszer jelentőségére.

Az OpenACS 4 (Open Architecture Community System) rendszert a múlt hónapban vizsgáltunk meg. Ez a hálózati közösségek létrehozására tervezett eszközkészlet legutóbbi változatai során megrázó változásokon ment keresztül. Talán mind között a legfontosabb eltérést a szolgáltatások modulokba osztása jelenti. Ezeket a modulokat az OpenACS-világban csomagoknak (packages) nevezik. Mivel minden csomag önálló, és mivel az összes csomagot bármely URL-hez csatlakoztathatjuk, az OpenACS 4 a rugalmas közösségi weblapok létrehozását minden eddiginél egyszerűbbé tette.

Ebben a hónapban elkezdjük az ismerkedést az OpenACS-csomagokkal, megnézzük, hogyan lehet őket telepíteni és használni (a cikkben feltételezzük, hogy korábban már feltelepítettük a PostgreSQL, AOLserver és OpenACS-mag programokat, ahogyan azt a Kovácsműhely előző két részében bemutattuk). Mivel a legtöbb OpenACS-hely nem próbál meg mindent nulláról felépíteni, hanem a beépített alkalmazásokkal érkező szolgáltatásokat használja, a csomagok telepítése olyasmiről, amit az alaprendszer feltelepítése után minden OpenACS-rendszergazdának végre kell tudnia hajtani.

Web/Adatbázis Csomagok

Figyeljünk meg a következő egyszerű, Perlben írt CGI-programot:

```
#!/usr/bin/perl

use strict;
use warnings;
use CGI;

my $query = new CGI;
print $query->header();
print $query->start_html(-title => "Testing");
print "<p>This is some text</p>\n";
print $query->end_html();
```

Helyezzük a fenti programot *test.pl* néven webkiszolgálónk CGI könyvtárába! A végrehajtás eredményét mindenki megtekintheti a <http://www.lerner.co.il/cgi-bin/test.pl> címen. Ha azt szeretném, hogy a programom több különböző néven is elérhető legyen, lemásolhatom; az újonnan kiválasztott nevet az URL is tükrözni fogja.

A dolgok némileg bonyolódnak, ha a kiszolgálóoldali alkalmazásom egyetlen program helyett több CGI-kódot is tartalmaz.

Amennyiben egy ilyen alkalmazásról szeretnék a rendszeremen másolatot készíteni, az összes programfájl le kell másolnom. A legtöbb esetben megkönnyíti a dolgomat, ha a fájlokat külön könyvtárba helyezem, majd ahányszor valahol máshol is le akarom futtatni, a könyvtárat a teljes tartalmával együtt lemásolom.

Az ilyen másolatok készítése azonban felvet néhány szinkronizációs problémát: ha kiküszöbölök valamilyen hibát a program egyik példányán, ugyanazt a változtatást a kód minden másolatában el kell végezni. A nehézségek egy részét ugyan a CVS használatával megoldhatom, de a helyzet úgy is megoldható, hogy csak egyetlen változatot tartok meg a fájlrendszeremen. Ezután a webkiszolgálót (legyen az Apache vagy AOLserver) úgy állítom be, hogy egy vagy több URL-en keresztül is elérhessem a programot (esetleg közvetett hivatkozást hozok létre, ez talán a legegyszerűbb – a fordító). Ezek után gondoljuk meg, mi történik, ha az alkalmazás relációs adatbázisokkal is dolgozik. Az alkalmazás telepítése már korántsem egyszerű fájlmásolásból vagy HTTP-kiszolgáló beállításból áll. Immár meg kell tudnunk oldani az alkalmazás-másolatok között fellépő esetleges ütközéseket és zavarokat, például azt, hogy a */foo/bboard* fórum adatbázisszinten ne keveredjen össze a */bar/bboard* másolattal. Amikor az alkalmazást eltávolítjuk a rendszerről, az általa használt adatbázis-táblákat is el kell tudnunk távolítani.

OpenACS alatt a gond megoldását az APM (ArsDigita Package Manager) használata jelenti. Az APM-et eredetileg a ma már nem működő ArsDigita nevű tanácsadó cég fejlesztette ki, amelyik az OpenACS elődjét is megalkotta. Az ACS kizárólag Oracle adatbázis-kiszolgáló alatt működött, az OpenACS azonban Oracle és PostgreSQL alatt egyaránt.

Az APM több, adatbázis-kezelőt használó, kiszolgálóoldali alkalmazásokra jellemző gondot is megold. Többek között változatkezelést végez, táblakészítő és -eltávolító parancsfájlokat tartalmaz és adatbázis-független. Az APM-et úgy tervezték, hogy az alkalmazás minden egyes példánya saját független beállításváltozókkal rendelkezessen, illetve egy vagy több URL-hez is hozzárendelhető legyen.

Fájlrendszerkiosztás

Az APM tulajdonképpen nem más, mint egy *.apm* kiterjesztésű *.tar.gz* fájl. Az állomány névformátuma általában: *csomagnév-0.5d.apm* formátumú, ahol a csomagnév a csomaghoz tartozó egyedi névvel helyettesítendő. Ez a példacsomag a 0.5-ös fejlesztői változatot tartalmazza. Ha a csomagot megnyitjuk a `tar -zxvf fájlnév` paranccsal, feltáru előtünk a szabványos fájl- és könyvtárszerkezet:

- *packagename.info*: a csomag tartalmát leíró XML-állomány. Ez a fájl – amit az OpenACS APM-alkalmazás általában önműködően készít el – mutatja meg az OpenACS-nek, hogy mely fájlok vannak a csomaghoz rendelve, illetve milyen beállítási értékeket érhet el a felhasználó. Azt is



megmutatja, hogy az alkalmazás singleton-e (azaz nyújt-e szolgáltatásokat a rendszer többi részének), vagy hagyományos alkalmazás (vagyis egy bizonyos URL alól futtatható).

- Az SQL könyvtár: itt találjuk a táblalétrehozó (és táblatörlő) parancsfájlokat. Eredetileg, amikor az ACS még csak az Oracle-t támogatta, ez a könyvtár mindössze két fájlt tartalmazott, *csomagnév-create.sql* és *csomagnév-drop.sql* néven. Az APM-telepítő a csomag telepítésekor a *create* (létrehozó) parancsfájlt, a csomag eltávolításakor pedig a *drop* (törlő) parancsfájlt futtatja (a *create* parancsfájl gyakran INSERT-eket is használ; ilyenkor a későbbi felhasználáshoz a táblákat szabványadatokkal tölti fel).
- A *tcl* könyvtár függvény meghatározásokat tartalmazó *Tcl*-fájlokat tárol. Ezek a függvények indításkor betöltődnek az AOLserverbe, ami sebességnövekedést jelent az OpenACS rendszer egyéb helyein használt *.tcl*- (vagy *.adp*-) lapokhoz képest.
- Jelenleg – mivel az OpenACS az Oracle mellett a PostgreSQL-t is támogatja – ez a könyvtárszerkezet módosult kissé. Az *sql* könyvtárban egy *oracle* és egy *postgresql* alkönyvtárat találunk, amelyek azonos szolgáltatású táblalétrehozó és -törlő parancsfájlokat tartalmaznak. Minden feltelepített OpenACS rendszer tárolja, hogy éppen milyen adatbázis-kezelőt támogat (ezt az adatot az AOLserver *nsd.tcl* beállításfájljának egyik változójában találjuk meg), és ennek megfelelően választja ki a legmegfelelőbb parancsfájlt.
- A *www* könyvtárban találjuk azokat a dolgokat, amelyek logikusan a webalkalmazás részét képezik. Ide helyezzük *.tcl*- és *.adp*-lapjainkat, illetve az összes grafikat, és az alkalmazáshoz tartozó egyéb kiegészítő állományokat is. Az OpenACS lekérdezőstovábbítója (query dispatcher) – ami lehetővé teszi, hogy a kiszolgálóoldali programok több adatbáziskiszolgálót is támogassanak – *.xml* kiterjesztésű XML-fájlokkal működik; ezek szintén a *www* könyvtárba kerülnek.
- Az OpenACS sablonrendszerének működési sajátosságai folytán nem ritka, hogy egyetlen honlap három fájlt is használ: egy *.tcl*-fájlt a változók beállításához, egy *.xml*-fájlt, ami az adatbázis táblasorainak SQL-lekérdezését teszi lehetővé, és egy *.adp*-fájlt, amely az adat HTML-átalakításáért felelős.

Az APM-ek ezeken felül számos egyéb fájlt is tartalmazhatnak, például adatbázis-továbbfejlesztő és összefésülő parancsfájlokat (azok számára, akik a csomag előző változatából szeretnének adatokat átmenteni), működési próbákat (a csomag hibátlan működésének az ellenőrzésére), karbantartói eszközöket (a *www/admin* alatt) és HTML formátumú csomagleírást (a *www/doc* alatt).

Csomagok betöltése és telepítése

Az APM-csomag használatba vételének első lépése a betöltés, ami általában annyit jelent, hogy ugyanabba a fájlrendszerbe másoljuk, ahol OpenACS rendszerünk is megtalálható. Amennyiben OpenACS rendszerünk a */web/atf/* könyvtárban helyezkedik el, minden csomagot a */web/atf/packages* könyvtárba kell helyezni (ezért aztán minden egyes csomagnak egyedi névre lesz szüksége; sok OpenACS-fejlesztő követi az Emacs-stílusú csomagelnevezési hagyományt, ahol a csomag nevét a fejlesztő vagy az ügyfél neve előzi meg – ez segít elkerülni az egyes csomagok közötti ütközést). A csomag teljes könyvtárszerkezetét másoljuk a */web/atf/packages*-be, és győződjünk meg róla, hogy a fájlok és könyvtárak olvashatók

(és írhatók) azon a felhasználói néven, amely alatt az AOLserver üzemel.

Az APM-csomagok telepítésének könnyebb és megbízhatóbb módja, ha a csomagkezelőben a *load packages* (csomagok betöltése) hivatkozásra kattintunk. Az OpenACS rá fog kérdezni az APM-hez rendelendő URL-re vagy a csomagokat tartalmazó könyvtárnévre. Az OpenACS ezt követően megkeresi az összes ott található *.apm*-fájlt, kicsomagolja és betölti őket a rendszerbe.



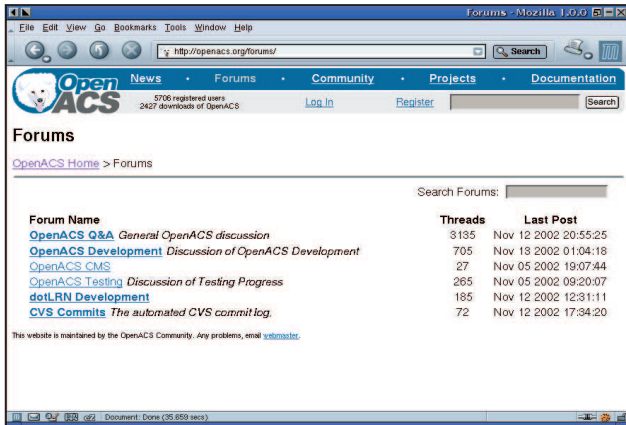
Miután a csomag betöltődött a fájlrendszerbe, fel kell telepítenünk a hozzá tartozó adatmodellt, és be kell jegyeznünk a rendszerbe. Ezt a weblapú csomagkezelő felhasználásával tehetjük meg, amit az OpenACS rendszereken általában a */acs-admin/apm* URL alatt találhatunk meg. A csomagkezelőhöz általában csak a hely karbantartója férhet hozzá.

A csomagkezelő oldala a rendszerbe töltött összes csomagot bemutatja, illetve jelzi, melyek vannak telepítve, melyiket helyettesítette egy frissebb változat, vagy melyek várnak még telepítésre. A lap tetején fellelhető menü segítségével a rendszer csomagjainak különféle alhalmazait is leválogathatjuk, például kérhetjük, hogy csak azokat mutassa, amelyek fejlesztéséért és kezeléséért személyesen felelünk.

A csomagkezelő a legfontosabb eszköz, amellyel a csomagok létrehozását, módosítását, frissítését és telepítését végezzük:

- Telepíthetjük a csomaghoz tartozó adatmodellt, és bejegyezhetjük a csomagot az OpenACS-be. Az OpenACS következő indításakor a csomag *tcl* könyvtárban található fájl az AOLserver memóriájába töltődnek. Csomagunk ezt követően az OpenACS honlaptérképén keresztül csatolható URL-hez, ahogy azt hamarosan láthatjuk.
- A csomagkezelővel új OpenACS-csomagokat is készíthetünk. Az új OpenACS-csomag előállításának első lépése, hogy a csomagkezelővel egy új könyvtárat és *.info* fájlt hozunk létre.
- Megvizsgálhatunk bármilyen, korábban a rendszerbe töltött fájlt, és lekérhetjük a csomaghoz rendelt kapcsolók, fájlok vagy bármilyen egyéb adat listáját.
- A csomagot módosíthatjuk is. Megváltoztathatjuk a kapcsolóit, fájljait, illetve a csomaghoz tartozó egyéb adatokat.

A csomag telepítéséhez kattintsunk az *install packages* hivatkozásra a lap alján. A csomagkezelő előbb új csomagokat keresve végignézi a csomagkönyvtárat, majd kiválaszthatjuk a telepí-



teni kívánt csomagot (ha első ízben telepítjük az OpenACS-t, egyetlen csomag sincs még telepítve, ezért itt igen hosszú listát találunk). Minden csomag egy vagy több másik csomagtól függhet. Ha olyan csomagot próbálunk feltelepíteni, amelynek az előkövetelményei még nincsenek fenn, a csomagkezelő a folytatás előtt megerősítést kér.

A telepítöben megadhatjuk, hogy csak a csomag adatmodelljét szeretnénk-e feltenni, vagy a csomag használatát is engedélyezzük a webhelyen. A magam részéről én minden felrakott csomagot engedélyezek, de biztos vagyok benne, hogy van olyan helyzet, amikor ezt nem szeretnénk megtenni. Miután ellenőriztük a megfelelő dobozokat, kattintsunk a *Next* gombra, ami az adatmodellt fogja telepíteni. Ezek után az AOLserver-t újra kell indítanunk, hiszen sok modul függ az AOLserver indításakor betöltődő *tbl* könyvtáraktól. A csomagok addig nem működnek, amíg a kiszolgálót újra nem indítjuk.

Csomag befűzése

Az AOLserver újraindítását követően ismét lépünk a csomagkezelőbe, és nézzük meg az engedélyezett csomagok listáját. Az összes betöltött csomagnak látszania kell a listában. Ezen a helyen a csomagokat tetszés szerint módosíthatjuk, újabb elemeket tölthetünk be, illetve a betöltött csomagot tényleges honlappá alakíthatjuk. Ezt a befűzésnek (mounting) nevezett folyamatot az OpenACS helytérképén végezhetjük el. Az OpenACS karbantartó rendszerének egyik legzavaróbb tulajdonsága, hogy a helytérkép nem a teljes helyhez tartozó felügleleti lapon található, hanem a fő webhely felületére került. Más szavakkal a csomagokat a */acs-admin/apm* alatt tudjuk kezelni, ellenben a térképet a */admin/site-map* alatt érhetjük el. Ennek is megvan az oka, de inkább megtéveszti az embereket, semmint segít rajtuk.

A hely térképe mutatja meg az OpenACS-nek, hogy milyen URL-t kell az alkalmazáshoz csatolnia, például ha a *bboard* OpenACS-csomagot a */forum* URL vagy a */bboard* URL alól szeretnénk elérni. Néhány esetben ténylegesen jól jöhet, ha mindkét helyen elérhetjük. A térképen mindezt egérgattintásokkal oldhatjuk meg.

Amikor a csomagot első ízben szeretnénk valamilyen URL-hez rendelni, a / elérési út jobb oldalán található *New subfolder* befűzést kell használnunk. Meg kell adnunk, hogy melyik URL alá szeretnénk az új alkalmazást elhelyezni. Ahhoz például, hogy a *bboard* csomag a */forum* alól legyen elérhető, a forum szót kell megadnunk (a bevezető perjel nélkül).

Amennyiben ennyivel megelégszünk, az új alkönyvtár semmivel sem lesz több egy mappánál, amibe újabb mappákat,

illetve állandó dokumentumokat helyezhetünk. Ezért a mappához tartozó *New application* befűzésre kell kattintanunk, ahol a feltelepített alkalmazáscsomagok közül a megfelelőt kiválasztva azt emberi fogyasztásra alkalmas névvel láthatjuk el. Ez a név fog megjelenni a fejlécekben és a címsorokban. Előfordulhat ugyanis, hogy a csomagot a */bboard* URL mögé szeretnénk helyezni, de névként inkább az érthetőbb *Vitaforum* nevet adjuk neki. A címet nem lehet egykönnyen megváltoztatni, ezért válasszunk megfontoltan!

A *new application* befűzés egy új csomagpéldányt hoz létre, majd azt csatlakoztatja az általunk választott URL-hez. Ha még egy nevet szeretnénk az alkalmazáshoz rendelni, készítsünk egy újabb alkönyvtárat, és használjuk a *mount* hivatkozást. Eltartott egy darabig, mire rájöttem, hogy a *mount* létező alkalmazáspéldányok csatlakoztatására is képes, a *new application* pedig egy teljesen új alkalmazáspéldányt hoz létre. Így már értelmet nyer az alkalmazásleoldás (unmounting) lehetősége is (az elérési út jobb oldalán található *unmount* hivatkozással érhetjük el), ami a kiválasztott alkalmazást nem törli, csak elérhetetlenné teszi. Az adott alkalmazáspéldány végleges törléséhez a térképen az *unmounted application* hivatkozásra kell kattintanunk, majd a *delete link*-re kell – közvetlenül az *unmounted application* után – rábóknunk.

Az alkalmazások minden egyes példánya saját jogosultságrendszerrel és kapcsolókkal bír. A kapcsolók használatát különösen hasznos résznek találtam az OpenACS-ben, hiszen így annak ellenére, hogy az egyszer létrehozott alkalmazást több helyen használom, minden egyes példányhoz saját beállításokat rendelhetek. A térkép megfelelő hivatkozásait követve megtekinthetjük, illetve megváltoztathatjuk a csomagpéldányhoz rendelt kapcsolókat.

Összegzés

A *.apm*-fájl formájában terjesztett és CVS-ben az APM-alkalmazással karbantartott OpenACS-csomagok web- és adatbázis-alapú alkalmazások készítését és terjesztését teszik lehetővé. Importálás után az APM-csomag több példányban is felhasználható, egyedenként külön jogosultságokkal és eltérő kapcsolókkal. Mint azt a következő hónapban látni fogjuk, az APM-et saját web-, illetve adatbázis-alkalmazás csomagjaink készítésénél is felhasználhatjuk, ezáltal könnyedén terjeszthetjük társalkotóink és a közösség más tagjai között.

Linux Journal 2002. november, 103. szám



Reuven M. Lerner

(reuven@lerner.co.il) egy kisebb webes és internetes módszerekkel foglalkozó tanácsadó cég tulajdonosa és vezetője. Az ATF honlapon érhető el (☞ <http://www.lerner.co.il/atf/>).

Kapcsolódó címek

Az OpenACS-ről bővebben a

☞ <http://www.openacs.org>-on olvashatunk. A weblap leghasznosabb része valószínűleg a

☞ <http://www.openacs.org/bboard> címen található fórum, ahol vezető OpenACS-fejlesztők beszélgetnek egymással, illetve vitatják meg az eszközkészlet jövőjét.