

# A PostgreSQL és a PHP

**V**arázsoljuk PostgreSQL-lel tárolt adatbázisainkat a Webre, és egyúttal ismerkedjünk meg napjaink egyik leggyorsabban fejlődő programozási nyelvével, a PHP-vel. A cikksorozat első részében telepítjük a PHP-t (3-as vagy 4-es változat) Debian alá, képessé tesszük PostgreSQL adatbázisok elérésére, és megismerkedünk néhány egyszerű PHP utasítással is.

## Miért Debian?

A változtatválasztásom oka egyszerűen az, hogy én ezt használom. A programok és a telepítés is Debianon lett kipróbálva, ennek ellenére működni kell a többi változat alatt is. Elképzelhető, hogy a beállítófájlok helye más (például RedHat esetén), de a PHP programoknak és a beállító fájlokban levő beállításoknak ugyanígy kell működniük. Debian esetében a csomagok kiterjesztése .deb és a csomagkezelő program a dpkg, míg például RedHat esetében .rpm a kiterjesztés és rpm a csomagkezelő neve. RPM csomagokat az RPM -i csomagnév utasítással lehet telepíteni. RedHat esetén a php3.ini-t a /etc/httpd könyvtárban, az Apache webkiszolgáló-beállító állományait pedig a /etc/httpd/conf könyvtárban találjuk.

## A PHP működésének ellenőrzése

Írásomban feltételezem, hogy az olvasó működő Apache webkiszolgálóval rendelkezik. Ennek telepítését ugyanis nem írom le. Legyen egy felhasználónk, akinek a weboldalait meg tudjuk nézni, és akinek a nevében bejelentkezve weboldalakat tudunk készíteni. A szükséges beállítások elvégzéséhez természetesen ismernünk kell a rendszer-

gazda jelszavát is. Helyezzük el a **phpinfo.shtml** fájlt (1. lista) a felhasználónk webes könyvtárában (ez általában a felhasználó saját könyvtárának public\_html nevű alkönyvtára), adjunk olvasási jogot mindenkinek (chmod o+r phpinfo.shtml), és próbáljuk elérni egy böngésző segítségével!

Ha látunk egy színes oldalt (1. kép), akkor minden rendben, működik a PHP, átugorhatjuk a következő részt. Ha bármi más történik, akkor vagy nincs telepítve PHP, vagy beállítási gondok vannak. Ellenőrizzük, telepítve van-e a php3 csomag! Debianon ezt a dpkg -l 'php3' paranccsal

nézhetjük meg. Ha a képernyőn a legelső oszlopban nem ii (2. lista) van, akkor telepítsük fel.

Használhatunk php4-et is, a cikk összes példája működik mindkét változattal, a két változat beállítása is szinte ugyanaz.

## A php3 telepítése

A php3 csomagot Debian alatt többféleképpen telepíthetjük. Használhatjuk a dselect programot: a főképernyőnél „/”-t nyomunk, majd beírjuk: „php”. A kívánt csomagoknál pluszt nyomunk, majd a csomagválasztó menüből kilépvé az „Install” pontot választjuk. Ha ismerjük a .deb fájl nevét és helyét, használhatjuk a dpkg parancsot is (3. lista):

2. lista A sor elején levő második betű (*n=Not installed*) mutatja, hogy nincs telepítve a php3. Ha ez *i(=Installed)*, akkor minden rendben

```
$ dpkg -l 'php3'
Desired=Unknown/Install/Remove/Purge
| Status=Not/Installed/Config-
files/Unpacked/Failed-config/Half-installed
| / Err?=(none)/Hold/Reinst-required/X=both-
problems (Status,Err: uppercase=bad)
| | Name Version Description
+++~
=====
pn php3 <none> (no description available)
```

3. lista A php3 telepítése. A LoadModule-lal kezdődő sort kell megjegyeznünk

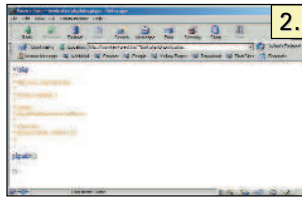
```
monkey:~# cd /cdrom/dists/slink/main/
binary-i386/web/
monkey:/cdrom/dists/slink/main/binary-i386/web#
dpkg -i php3_3.0.5-3.deb
Selecting previously deselected package php3.
(Reading database ... 28718 files and
directories currently installed.)
Unpacking php3 (from php3_3.0.5-3.deb) ...
Setting up php3 (3.0.5-3) ...
I see you have apache webserver installed and
so far you haven't
used the apache module version of php3 in your
apache. If you want to use it, you should
reconfigure the apache webserver and select to
load the php module. I can call the
apacheconfig script now for you to do it, or
you can insert the following line into
/etc/apache/httpd.conf manually:
LoadModule php3_module
/usr/lib/apache/1.3/libphp3.so
Do you want me to run the apacheconfig script
now [y/N] ?
Ok, not running apacheconfig. Please read the
docs in /usr/doc/php3
Press RETURN to continue
```

```
dpkg -i php3_3.0.5-3.deb
```

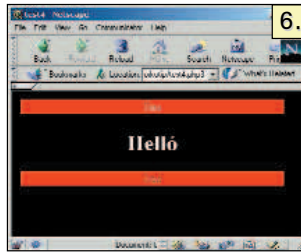
A kérdésre, hogy indítsa-e el a telepítő a webkiszolgáló beállítóprogramját, csak nyomjunk ENTER-t, és inkább állítsuk be mi kézzel! Ezután ki is írja, mit kell majd beszúrnunk. A PHP-hoz tartozó leírások külön csomagban találhatóak:

```
dpkg -i php3-doc_3.0.5-3.deb
```

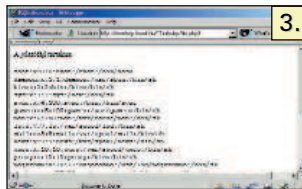
Ha a négyes változatot kívánjuk használni, természetesen a php4 nevű csomagokat kell telepíteni.



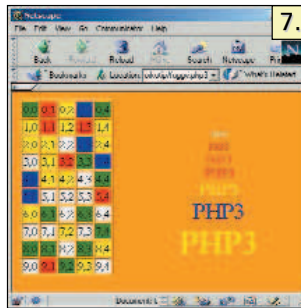
2.



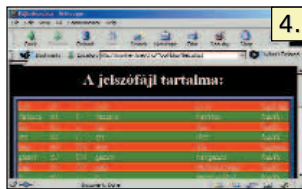
6.



3.



7.



4.



5.

Ha sikerült a telepítés, felkerül a gépünkre a *php3.ini* fájl, ami a php3 beállításait tartalmazza. Ezt Debianon a */etc/php3/apache* könyvtárban találhatjuk meg (vagy a *php.ini*

fájlt a */etc/php4/apache* könyvtárban). Egyelőre hagyjuk változatlanul, és módosítuk az Apache beállítóállományait!

## Beállítás

A */etc/apache/srm.conf* fájlban keressünk rá a php3 szóra! Alapértelmezésként megjegyzésben vannak a php3-ra vonatkozó részek, szedjük ki a # jeleket. Mivel a következő php program példák kiterjesztése php3, ezért módosítuk is az

```
AddType application/x-httpd-php3
```

sort, hozzáfűzve a php3 szót (4. lista

☞ [www.linuxvilag.hu/magazin/cikkekhez.html](http://www.linuxvilag.hu/magazin/cikkekhez.html)). Az

```
AddType application/x-httpd-php3-source .phps
```

sorból látható, hogy a phps kiterjesztésű fájlokat php3 forrásként kezeli, azaz meg tudja jeleníteni a tartalmukat (2. kép).

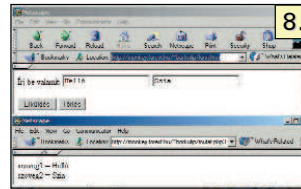
A */etc/apache/httpd.conf* fájlban keressük meg a LoadModullal kezdődő sorokat tartalmazó részt és írjuk a végére a libphp3.so modult betöltő sort (5. lista ☞ [www.linuxvilag.hu/magazin/cikkekhez.html](http://www.linuxvilag.hu/magazin/cikkekhez.html)):

```
LoadModule php3_module
/usr/lib/apache/1.3/libphp3.so
```

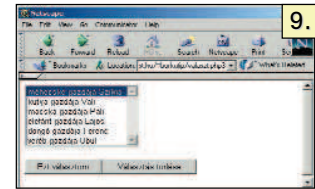
php4-hez az `AddType application/x-httpd-php` sorhoz fűzzük a `phtml php3 php4 php` szavakat, ugyanígy az `AddType application/x-httpd-php-source .phps` sor is kell. A `httpd.conf`-ba pedig a következő sort kell beírunk:

```
LoadModule php4_module
/usr/lib/apache/1.3/libphp4.so
```

Indítsuk újra a kiszolgálót a `/etc/init.d/apache restart` paranccsal és próbáljuk megnézni a `phpinfo.phtml` fájlt a böngé-



8.



9.

zőnkkel! Most már az 1. képhez hasonlót kell látnunk. Ha mégsem, ellenőrizzük a `php3` meglétét, az `srm.conf`-ot és a `httpd.conf`-ot!

## Ismerkedés a php3-mal

Helyezzük el a `public_html` könyvtárunkban a `phpinfo.php3` és a `phpinfo.phps` fájlokat! A `phpinfo.php3` betöltésekor ugyanazt kell kapnunk, mint a `phpinfo.phtml` megnézésekor (1. kép). Ha mégsem, akkor nem állítottuk be az `srm.conf`-ot. Ellenőrizzük az állományt a 4. lista alapján! A `phpinfo.phps` betöltésekor a böngészőben a forrásprogramot kell látnunk (2. kép).

A következő program a `file.php3` (6. lista) Tegyük ezt is a `public_html` könyvtárba és nézzük meg (3. kép).

A program megjeleníti a `/etc/passwd` fájl tartalmát a böngészőben (3. kép). Ez rengeteg adatot adhat egy betörő kezébe.

Gondolom, nem csak az én gyomrom rándul össze a `passwd` fájl tartalmát látva. A PHP igen erős eszköz a felhasználók kezében, jól gondoljuk meg, telepítjük-e éles rendszeren.

Vizsgáljuk meg a fájl tartalmát. Átlagos HTML-dokumentumot látunk, a

```
<? readfile('/etc/passwd'); ?>
```

sor kivételével, ami a `readfile()` php3 utasítást hajtja végre a kiszolgálóval.

A PHP kódot a HTML-oldalban helyezük el. Ezáltal a weboldalunk forrása áttekinthetőbb lesz, mint ha például Perl vagy C-alapú weboldal-előállító programot használnánk. Az utóbiainál ugyanis a programunk szerkezetét kell követnie a programba illesztett HTML-kódnak, míg a PHP-nál a PHP programkódot tesszük a HTML-oldal megfelelő helyére. Így nagyon könnyű módosítani a weboldal kinézetét. A PHP és a HTML-kód szétválasztására szolgál a `<? . . . ?>` tag. Tehát a PHP programunk részeit helyezzük el a `<? és a ?>` tagok között.

A PHP kódot a webkiszolgáló hajtja végre, ezért a forráskódunk nem jut el a böngészőhöz, csak a PHP program eredménye.

A `php3`-mal a további ismerkedéshez, csináltsuk az előző programot! A `file.php3` (7. lista ☞ [www.linuxvilag.hu/magazin/cikkekhez.html](http://www.linuxvilag.hu/magazin/cikkekhez.html)) szintén a `passwd` fájl tartalmát írja ki, de most soronként olvassuk a fájlt, így lehetőség nyílik az egyes sorok színezésére és táblázatba helyezésére (4. kép). Nézzük át a fontosabb részeket! (A program-példa sorai számozottak, de ez csak a magyarázat kedvéért szükséges. Az eredeti fájlból természetesen hiányoznak a sorszámok.)

## A program szerkezete

Az 1–8 sorok normál HTML-fájl kezdősorai. A 9. sorban belépünk a `php3`-blokkba. Két változót hozunk létre értékadással. Minden változó neve \$ jellel kezdődik. Mindkét változó egy tömb. A `file()` utasítás a megadott fájl olvassa be, a tartalmát pedig tömbként adja vissza, ahol a tömb minden eleme a fájl egy-egy sorát tartalmazza. A 10. sorban létrehozuk a `$colors` tömböt, aminek két karakterlánc típusú eleme van. A 11. sorban kilépünk a `php3`-ból. A következő `php3` blokkot a 16–22-ik sorok mutatják. C-szerű számlálós ciklus a 16. sorban, a `count()` függvény egy tömb elemeinek számát adja meg. Mivel a tömbök első elemének sorszáma 0, így `$(count($lines) - 1)` következtében a tömb összes elemén végigmegyünk. A 17. sorban a `$color` változónak hol a

9. lista A php3.ini fájlba kerüljön bele az `extension=pgsql.so` sor

```

;;;;;;;;;;;;;;;;;;;;;;;;;
Dynamic Extensions ;
;;;;;;;;;;;;;;;;;;;;;;;;;
; if you wish to have an extension loaded
; automaticly, use the
; following syntax:
    extension=modulename.extension
; foe example, on windows,
; extension=mysql.dll
; or under UNIX,
; extension=mysql.so
; Note that it should be the name of the
; module only,
; no directory information
; needs to go here. Specify the location of
; the extension with the extension_dir
; directive above.
; extension=pgsql.so

```

<TD BGCOLOR=red>, hol pedig <TD BGCOLOR=green> értéket adunk, attól függően, hogy \$i páros-e vagy sem. (Az a % b művelet az a és b között végzett osztás maradékát adja.) A tömbök elemeit \$tömbnév[index] hivatkozással kapjuk meg, a . (pont) művelet pedig karakterláncokat fűz össze). A 18–20. sor egyetlen kiíró utasítást tartalmaz, a kinézet kedvéért három részre osztva. A php3 macskakörmök közé tett karakterláncokban elhelyezett változóneveket az értékükkel helyettesíti. A 19. sor szorol bővebb magyarázatra, ugyanis itt vágjuk szét a jelszófájl egy sorát és ragasztjuk össze úgy, hogy a mezők között a </TD><TD BGCOLOR=...> szöveg legyen. Az eplode(határolójel, szöveg) függvény szétdarabolja a szöveget – jelen esetben a „:” a határolójel –, míg az implode(határolójel, szövegtömb) egyesíti egy karakterláncá a szövegtömb elemeit úgy, hogy minden elem közé a határolójelet teszi. A 20. sorban levő \n hatására a print utasítás új sor karaktert ír ki. Most tekintsük át röviden, hogyan vehetjük fel a kapcsolatot egy PostgreSQL adatbázissal!

## A PHP3 PostgreSQL moduljának ellenőrzése

Próbáljuk ki a test.php3 programot, (8. lista megtalálható: [www.linuxvilag.hu/magazin/cikkekhez.html](http://www.linuxvilag.hu/magazin/cikkekhez.html)). Ez megpróbál kapcsolódni a `php_db` adatbázishoz a `php_user` nevű felhasználó nevében a `heureka` jelszóval. Ha a böngészőnk a *Fatal error: Call to unsupported or undefined function pg\_connect() in .../public\_html/test.php3 on line 2* hibaüzenetet adja, fel kell telepítenünk a `pgsql` modult a `php3`-hoz. Ez a `php3-pgsql_3.0.16-2.deb` fájl (`php4`-hez a `php4-pgsql_4.0b3-4.deb` fájl). A telepítéskor a `/usr/lib/php3/apache` könyvtárba bekerül a `pgsql.so` fájl. Ez az a modul, amit be kell állítanunk a `/etc/php3/apache` könyvtárban található `php3.ini` fájl segítségével. Szerkesszük ezt át! A `php3.ini` fájlban az `extension_dir`-rel kezdődő sor mutatja, hol kell lenniük a moduloknak. Ennek a `/usr/lib/php3/apache` értéket kell tartalmaznia. A *Dynamic Extensions* részbe írjuk be (9. lista): `extension=pgsql.so` (a „:”-vel kezdődő sorok megjegyzések). Ezután indítsuk újra az Apache-ot a `/etc/init.d/apache` restart utasítással! Értelemszerűen `php4`-hez a `/usr/lib/php4/apache` könyvtárban levő `pgsql.so` fájl és a `/etc/php4/apache` könyvtárban levő `php.ini` fájl

használjuk. Próbáljuk újra a `test.php3`-at. Ha a hibaüzenet nem változott, ellenőrizzük a `php3.ini` tartalmát és indítsuk újra az Apache-ot. Ha viszont az alábbi üzenetet kapjuk:

```
Warning: Unable to connect to PostgreSQL server:
FATAL 1: Database php_db does not exist in
pg_database in /home/.../public_html/test.php3 on
line 2
```

akkor minden rendben, a kapcsolat felépült.

Ismerkedjünk kicsit a PostgreSQL adatbázis-kiszolgálóval. Kapunk hozzá egy konzolos segédprogramot, a `psql`-t. Ezzel hozzáférünk az adatbázisokhoz, táblákhoz és SQL-lekérdezéseket indíthatunk el. A programot a `psql` utasítással indíthatjuk el. A `\?` paranccsal segítséget kérhetünk. Minden SQL parancs végét pontosvessző zárja. A parancsok lehetnek többsorosak is, ekkor az utolsó sor végét zárjuk pontosvesszővel. A `psql` belső parancsai fordított perjellel kezdődnek és nem kell pontosvessző a lezárásukhoz. A `\l` kiírja az adatbázisokat, `\connect adatbázisnév` paranccsal kapcsolódhatunk a megadott adatbázishoz, `\d` kiírja az adatbázis tábláit.

Ha még nem hoztuk létre saját nevünkhöz PostgreSQL-felhasználót, akkor tegyük a következőt rendszergazdaként:

```
su - postgres
createuser
```

Ezután be kell gépelnünk a bejelentkezési nevünket. A user ID-re csak nyomjunk ENTER-t. Saját magunknak érdemes engedélyezni az adatbázisok létrehozását, és hogy esetleg mi legyünk a PostgreSQL rendszerfelügyelői (akik felhasználókat is létrehozhatnak) úgyhogy a következő két kérdésre válaszoljunk `y`-nal.

Következő lépésként hozzuk létre a `php_db` adatbázist. Ezt `psql`-ben a `create database php_db;` paranccsal tehetjük meg. Újra megpróbálva a `test.php3`-at, a *FATAL 1: SetUserId: user 'php\_user' is not in 'pg\_shadow'* üzenetet kell kapnunk, úgyhogy hozzunk létre a felhasználót is `psql`-ben:

```
create user php_user with password heureka
    norecreatedb norecreateuser;
```

utasítással. A `norecreatedb` hatására `php_user` nem tud adatbázisokat létrehozni, `norecreateuser` miatt pedig felhasználókat se. Ha a böngészőnk *Document contain no data* vagy hasonló üzenetet küld, helyben vagyunk. A `php3` képes elérni a PostgreSQL adatbázisokat (5. kép).

## Hogyan tovább?

A cikk folytatásában megismerkedünk a `php` PostgreSQL adatbázisokat kezelő függvényeivel, a PostgreSQL biztonsági beállításával. Közlünk egy webes alkalmazást és rövid útmutatót kapunk, hogyan mentjük át adatbázisainkat újabb változatú PostgreSQL alá.

Borkuti Péter (borkuti@freemail.hu)  
matematika-informatika szakos tanár, rendszergazda,  
informatikus, rendszerépítő és programozó.

### Kapcsolódó címek

PHP levelezőlista <http://weblabor.hu/wl-phplista/>  
PHP tükör <http://hu.php.net> címről a Documentation  
részről letölthetjük a leírás magyar fordítását.