

Hálózaton keresztüli MySQL-lekérdezés Python segítségével

Hogyan készítsünk kifinomult keresőfelületet Python CGI parancsfájlok alkalmazásával?

Mostanában ezernél is több Go-játékállás került a birtokomba (a Go ma is kedvelt táblás játék az ősi Keletről), amelyek profi játékosok vagy éppen lelkes műkedvelők lépéseit őrzik. Ezek Smart Game Format-ban (SGF) találhatóak – ami az egy-két játékoson alapuló táblás játékok állásainak tárolására tervezett szöveges formátum. Honlapomon keresztül ezeket természetesen más játékosok számára is elérhetővé tettem.

Az adatállományban való keresést megkönnyítendő először egy HTML-űrlapot használtam, amely egyetlen szöveges bemeneti mezőt tartalmazott. A felhasználó begépelhetett egy karaktersorozatot (például a játékos nevét), ami azután egy Python CGI parancsfájlokhoz került, és ez a jó öreg *grep*-et használta arra, hogy megtalálja a megfelelő fájlt (1. kép). Csakhogy meglehetősen nyers formája ez a keresésnek. Például az „egy adott játékos által játszott összes játékot” könnyen kikereshetjük ily módon, de az olyan összetettebb dolgokkal már nem tudunk mit kezdeni, mint „az összes játék, amelyet egy bizonyos játékos fekete kővel játszott” vagy „az összes, adott játékos által 1995-ben megnyert játék”.

Ahhoz, hogy jobb keresőmotort készíthessek, más megközelítést kellett választanom. Először is az adatbázisnak le kell írnia a játékfájlok gyűjteményét, majd többszörös bemeneti mezőt szükséges készíteni, hogy a felhasználónak egy időben több különféle adat szerinti keresésre is lehetősége nyíljon. Végezetül valami módot kell találnunk arra, hogy a böngésző és az adatbázis között kapcsolat épülhessen fel, valamint, hogy a keresés eredményét HTML-formátumban a felhasználóhoz tudjuk juttatni. A keresés teljes menetét az 1. kép szemlélteti.

Az eszközök

Adatbázisként a MySQL 3.22.32-változatát, CGI parancsfájl készítéshez pedig a Python 1.5.2 rendszert választottam. A Perl-en is gondolkodtam, de a Python stílusa inkább kedvemre való volt. Ezeknek a programoknak a telepítése már számos cikkben megjelent, így erről most külön nem szólnunk. Az olvasó további adatokat lelhet a telepítéssel kapcsolatban a cikk végén megadott címen. A MySQL és a Python közötti kapcsolattartást kiegészítő modul segítségével lehet megvalósítani, amire rövidesen kitérek.

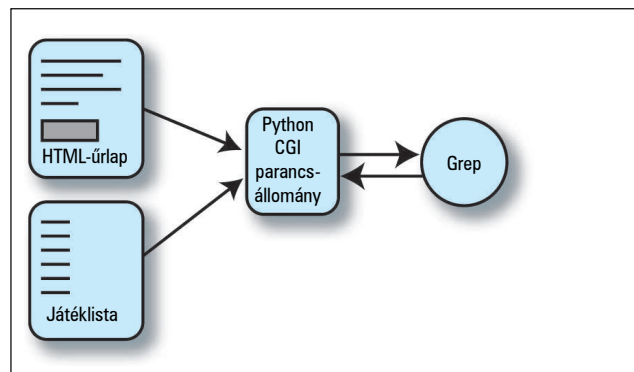
A MySQLmodule

A Python minden MySQL-lekérdezést a *Joerg Senekowitsch* által tervezett egyedi modulon keresztül végez. Természetesen jó néhány egyéb modul is található a hálózaton, de a MySQLmodule1.4 könnyen telepíthető és megérthető. Nálam igen jól bevált. Néhány rendszeren (mint például a FreeBSD) ezt a modult a Pythonnal egy időben fel lehet telepíteni. Az én Slackware 7.1 Linuxom alatt dinamikusan betölthető modulként kellett lefordítanom. Ez háromlépéses művelet: ki kell bontani a MySQLmodule tárolóállományát, le kell fordítani a megosztott modult, majd a készlet valahová a Python könyvtárszerkezetébe célszerű telepíteni.

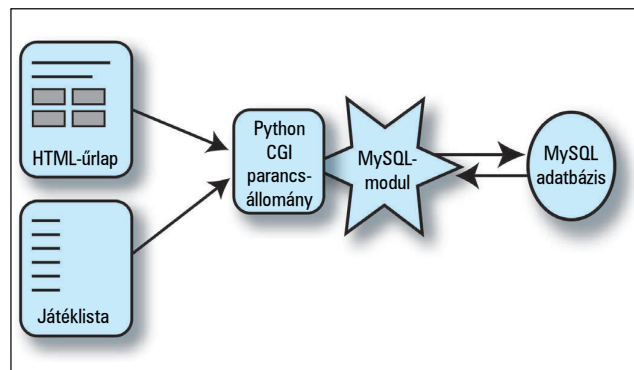
Az első lépés meglehetősen egyszerű. Rendszergazdaként a következőt gépelhetjük be:

```
myhost:~# tar xvzf MySQLmodule-1.4.tar.gz
```

Ezáltal néhány állományt tartalmazó új könyvtár jön létre MySQLmodule-1.4/ néven. Az itt található fájlok közül a legfontosabbak: a MySQLmodule.c (a lefordítandó modul forrásfájla), illetve a README (ami a telepítéssel kapcsolatos hasznos adatokat tartalmazza).



1. ábra A régi, kimunkálatlan keresési felület



2. ábra Kifinomultabb felület

A második lépésben már egy-két trükk is megbújik. Pontosan tudni kell például a MySQL és a Python könyvtárainak és fájllálmányainak a helyét. Az én rendszerem a MySQL 3.22.32 a /usr/lib/mysql könyvtárba helyezi a mysqlclient könyvtárat, a mysql.h include fájl pedig a /usr/include/mysql helyen lelhető fel. A Python-könyvtárak a /usr/lib/python1.5/config helyen találhatóak, míg az include fájllálmányok a /usr/include/python1.5-ben. A MySQLmodule fordítását végző parancs a következő:

```
myhost:~# gcc -shared -I/usr/include/mysql -I/
  ↳usr/include/python1.5 MySQLmodule.c -L/
  ↳/usr/lib/mysql -lmysqlclient -L/usr/lib/
  ↳python1.5/config -lpython1.5 -o MySQLmodule.so
```

Megszívlelendő tanács: a fenti parancsban megadott elemek sorrendje kötött, tehát nem szabad megváltoztatni! Hidd el nekem, ez drágán szerzett tapasztalat.

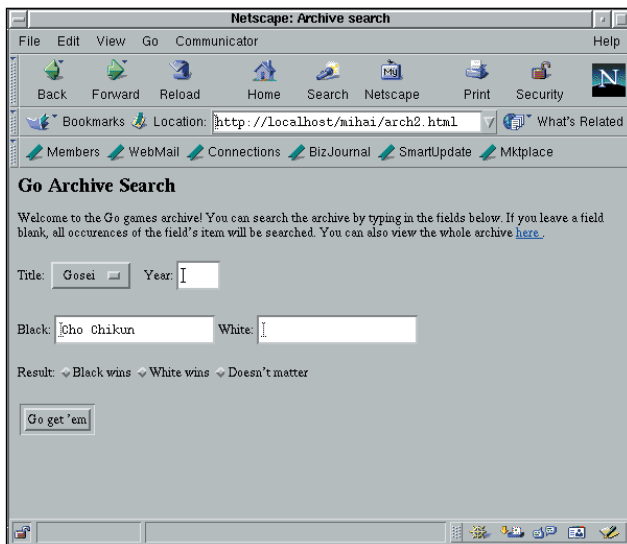
A harmadik lépés a MySQLmodule.so átmásolása egy olyan könyvtárba, ahol a Python futtatható állomány megtalálja. A Python 1.5 esetében ez a /usr/lib/python1.5/lib-dynload, ahol az egyéb megosztott objektumfájlok is találhatóak. A Python 2.0 esetén (ezt szintén kipróbáltam) a /usr/lib/python2.0/site-packages/ könyvtár használatát javaslom. Amint a modul telepítve lett, azonnal el is érhető a Pythonból. Célszerű rögvest ki is próbálni egy egyszerű import utasítással:

```
myhost:~$ python
Python 1.5.2 (#1, May 28 2000, 18:04:10)
Copyright 1991-1995 Stichting Mathematisch
Centrum,
Amsterdam
>> import MySQL
>>
```

Ha a Python nem „panaszkodott”, valószínűleg sikerült helyesen telepíteni a MySQLmodule-t és jól működik.

A HTML-űrlap

A felhasználó több szempont szerint is kereshet az adatbázisban, mint például „a viadal neve”, „a fekete és a fehér játékosok neve”, „a játék kelte” (legalább az év), illetve „a játék győztese”. Mindezen adatok elérhetők az SGF fájlokban, a pillanatnyi játékmezőben. Írtam egy HTML-dokumentumot, amely az 1. listában (lásd a 15. CD Magazin/MySQL könyvtárban) látható. Szeretek kézzel írni HTML-kódot, ráadásul jelen esetben semmilyen bonyolult dologra nincs szükség. Természetesen a *myhost* szócskát a webkiszolgáló pillanatnyi nevével helyettesítjük. Az olvasó azt is észreveheti, hogy a HTML-űrlap – amint a *submit* (elküld) gombot leüti – a *search.py* nevű CGI parancsfájlt hívja meg. A dokumentum Netscape alatti megjelenése a 1. képen látható.



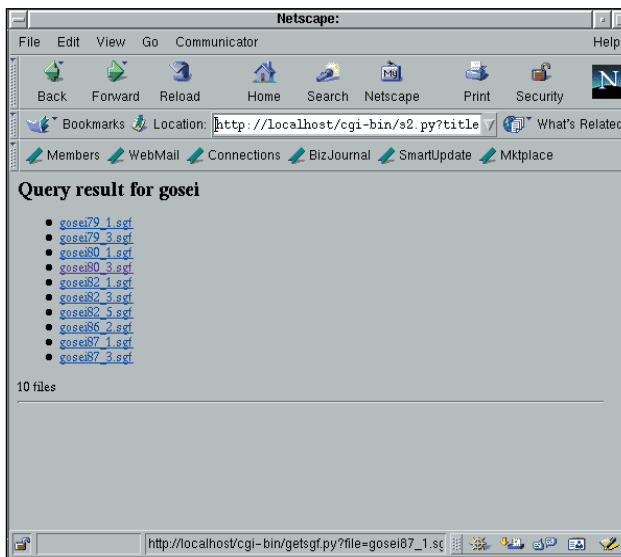
1. kép A dokumentum Netscape alatt

Az Igo Adatbázis

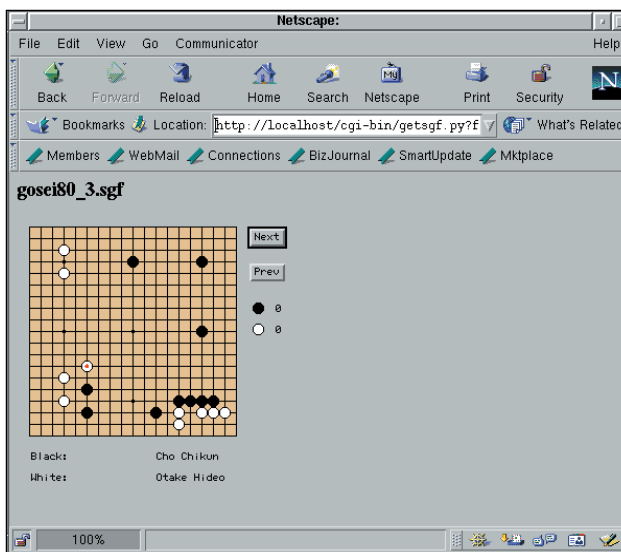
A HTML-űrlapon megjelenő összes elemet le kell írni az adatbázisban. Ezért készítettem egy új adatbázist, amely a táblákat tárolja. Ezt rendszergazdaként lehet végrehajtani, a következők begépelésével:

```
myhost:~# mysqladmin create igo
```

ahol az igo az új adatbázis neve. Ez természetesen még nem elég,



2. kép A search.py eredményei



3. kép gose:80_3.sgf.applet

mivel így egyedül a rendszergazdának lesz hozzáférése az új adatbázishoz. Ahhoz, hogy minden felhasználónak megadhassuk a SELECT jogosultságot (és csak azt) az új adatbázishoz, a rendszergazdának a következőket kell begépelnie:

```
myhost:~# mysql mysql
mysql> insert into db values (
    '%', 'igo', '', 'Y', 'N', 'N', 'N',
    'N', 'N', 'N', 'N', 'N', 'N');
```

A másik megoldás az lehet, ha GRANT SELECT ON 'igo.*' TO '%' parancsát használjuk, ebben az esetben a flush parancsra sincs szükség – *MonK*, a fordító.

Ez a MySQL belső adatbázisában megváltoztatja a táblaadatbázist, és új bejegyzést készít az igo adatbázis számára. Ezután a MySQL démont újra kell indítani vagy gépeljük be a flush privileges; parancsot, hogy a MySQL démon a jogosultságváltozásokat figyelembe vegye.

(Javaslatom esetén semmi ilyesmit nem kellene tenni, mindez szépen, önműködően megtörténne – MonK.)
Így már minden felhasználó elérheti az igo adatbázis adatait, de csak a rendszergazda képes megváltoztatni azokat.

A MySQL-táblázatok

A táblák készítése (minden viadalhoz egy-egy) meglehetősen könnyű volt. Például a Gosei viadalhoz tartozó *gosei* nevű tábla a következő parancsokkal készült:

```
myhost:~# mysql igo
mysql> create table gosei (
black varchar(30),
white varchar(30),
dt date,
rez varchar(10),
fname varchar(30)
);
Query OK, 0 rows affected, (0.00 sec)
mysql>
```

A táblának öt oszlopa van: a fekete játékos neve, a fehér játékos neve, a játék dátuma, az eredmény és végül a megfelelő SGF fájl neve. Az adatok betöltése a táblába azonban már más kérdés. Az egyik módszer akár ez is lehetne:

```
mysql> insert into gosei values (
Cho Chikun', 'Kato Masao', '1987-07-03',
'B+3.5', 'gosei87_1.sgf ');
```

de inkább megtanulok egy új programozási nyelvet, mintsem ezt ezer-szer begépeljem. Szerencsére akad más módja is annak, hogy az adatokat a MySQL táblába betöltsük – mégpedig szövegfájlokon keresztül. A fájlban minden sor egy táblasornak felel meg a táblában, a mezőket pedig szóközzel választják el, ahogy ez az első táblában látható. Tételezzük fel, hogy ezt a fájlt is goseinek nevezzük. Ahhoz, hogy ezeket az adatokat a MySQL táblába tölthessük, a következőt szükséges begépelni:

```
mysql> load data infile "gosei" into table gosei;
```

Az olvasó joggal kérdezhetné: „Rendben, de a hatalmas szöveges állományok elkészítése talán nem éppolyan nagy munka?” A válasz: nem, mert ez feladat meglehetősen könnyen elvégezhető egy másik egyszerű Python-parancsfájl segítségével (ezt most nem fogom bemutatni, mivel nem tartozik közvetlenül a tárgyhoz). Amint az igo adatbázis elkészült és a táblákat feltöltöttük adatokkal, már csak egyetlen dolog maradt hátra: megírni azt a Python CGI parancsfájlt, amely a felhasználóhoz juttatja el HTML-formátumban az űrlapot, lekérdezi az adatbázist és listázza a megtalált játékfájlokat.

A Python CGI parancsfájl

Ez a search.py nevű program a 2. listán látható (elérhető az <ftp://ftp.ssc.com/pub/lj/listings/issue85> címen). Két nagyszűrő modult használ, amelyeket a harmadik, illetve a negyedik sorban importáltak. A CGI modul csaknem olyan, mint egy varázslat: a HTML-űrlap által továbbított adatokat a Python könyvtárként kapja meg. A programozónak nem is kell olyan részletekkel foglalkoznia, hogy vajon milyen eljárás (GET vagy POST) alkalmazásával érkezzék az adat a CGI parancsfájlhoz. Ugye, milyen csodálatos a Python? A MySQL-module-t szintén könnyű használni: a kívánt adatbázissal négy egyszerű utasítással létrehozza a kapcsolatot, elküldi a megfelelő lekérdezést, megkapja az eredményt, és sorlistákban (rowlist, olyan Python-

lista, amelynek minden eleme egyetlen sort tartalmaz) tárolja azokat. A parancsfájl szerkezete a következő: a HTML-adat lekérdezése, az adatbázis-lekérdező karaktersorozat elkészítése az űrlap adatainak megfelelően, a tábla lekérdezése, majd a megfelelő sorok kiírása. Mivel minden Go-viadal külön táblában helyezkedik el, a harmadik lépést egy ciklusban ismételtjük, ameddig csak szükséges. A Python-kód és a csatolt megjegyzések önmagukért beszélnek, ezért csak a következő sorokra térek ki bővebben:

```
print '<li><a href="http://myhost' + \
'/cgi-bin/getsgf.py?file=' + e[0] + '>'
print e[0] + '</a>'
```

Itt az *e[0]* az SGF fájl neve. A fájlnev egyszerű kiírása helyett (ez csupán korlátozott segítséget jelentene a felhasználók számára), a `print` utasítás egy HTML-hivatkozást (anchor) készít, amely a fájlnevet a `getsgf.py` CGI parancsfájlhoz továbbítja. Ez utóbbi parancsfájl (az egyszerűség kedvéért most nem mutatjuk be) kikeresi az SGF fájl jelenlegi elérési útját, és elindít egy Java appletet, hogy a fájl tartalmát szép grafikus felületen jelenítse meg. Természetesen mind a `search.py`, mind pedig a `getsgf.py` fájlokat futtatható állománnyá kell tenni, és át kell helyezni a `cgi-bin` könyvtárba. Végül az utolsó elkerülendő csapda: minden SGF játékfájlt valahová a `DocumentRoot` könyvtárba helyezünk (annak megfelelően, ahogy azt a `httpd` beállítófájljában meghatároztuk), máskülönben a CGI parancsfájl nem találja meg őket. A `search.py` futtatásának eredményét a 2. képen figyelhetjük meg, ahogyan azt a felhasználó látni fogja. Bármely fájlra kattintva az 3. képen látható applet indul el, amely a játék egyes lépéseit a megfelelő sorrendben mutatja be.

Összegzés

A Python csodálatos nyelv: a `MySQLmodule` segítségével egyszerűvé válik az olyan apró programok írása, amelyek a MySQL-adatbázisból nyernek adatot. A Python CGI parancsfájlok írására is kiválóan alkalmazható. Egy adatbázis webes elérhetőségének megteremtése mindössze pár soros kód megalkotását jelenti. Az itt bemutatott alkalmazás meglehetősen korlátozott: a felhasználó mindössze öt változó szerint képes választani, és ezt is csak kötött, a HTML-kódba előre beégetett módon teheti meg. Ugyanakkor az is elképzelhető, hogy a felhasználó egy textarea bemeneti mezőbe saját adatbázis-lekérdezését írja be, majd a lekérdezés eredményét közvetlenül megszemlélheti. A lehetőségeknek tulajdonképpen csak a programozói képzelet szab határt.



Mihai Bisca

A Romanian Go bajnokság előző nyertese (AKA 5dan). Megszámlálhatatlan órát tölt Slackware Linuxán játékkal, és arról álmodozik, hogy otthonról fog dolgozni, az Interneten keresztül. Másik – nappali – életében szemorvosként tevékenykedik.

Kapcsolódó címek

Go ➔ <http://www.cwi.nl/people/jansteen/go/index.html>
MySQL ➔ <http://www.mysql.com/>
MySQLmodule ➔ <http://mysql.he.net/Downloads/Contrib/>
Python ➔ <http://www.python.org/>
SGF ➔ <http://www.red-bean.com/sgf/>
A szerző honlapja ➔ <http://igo.profnis.ro/>