

lehetőséget és játékeret hagyunk magunknak: mindössze a virtuális képernyőket váltogathatjuk az asztal alján található asztalváltó sávval.

A megnyitott ablakokat az előzőekben már szóba került PostScript-fájlok tartalmazta leírásokban található billentyűkombinációkkal is kezelni tudjuk. A `larswm` e téren számos lehetőséget nyújt.

A kilépés a `CTRL+ALT+BACKSPACE` billentyűkkel történik.

#### A Debian-Sidben található alapablakkezelők

<code>twm</code>	<code>phluid</code>	<code>vtvm</code>
<code>9wm</code>	<code>pwm</code>	<code>qwm</code>
<code>amiwm</code>	<code>ratpoison</code>	<code>uwm</code>
<code>aewm</code>	<code>gwm1</code>	<code>wm2</code>
<code>ion</code>	<code>ctwm</code>	<code>flwm</code>
<code>larswm</code>	<code>qvwm</code>	<code>failsafe</code>

#### `ctwm`

Szintén a `twm`-hez hasonló tudású ablakkezelő. Leírását úgy szintén még indítás előtt célszerű elolvasni. Ezenkívül ajánlom a leírás mellett található példafájlok `*.twmrc` valamelyikének saját könyvtárunkba tör-

tendő másolását `.twmrc` néven, nélküle ugyanis nagyon kevés eszköz fog a rendelkezésünkre állni. Én a `lynx.twmrc` fájlt próbáltam ki, amivel már kellemes eszközökre tettem szert. Az így kapott beállításokkal elérhetjük, hogy a bal egérgombbal kattintva alkalmazásmenü bukkan fel, a középső gomb használat esetén az ablakkezelő vezérlőmenü (kilépés stb.) jelenik meg, a jobb egérgombbal kattintva pedig az ablakok kezelését segítő menü (ablakbezárás, képernyőanimáció) áll a rendelkezésünkre. Szomorú tapasztalat, hogy ez volt az egyetlen ablakkezelő, amely esetenként nem érzékelte az egér használatát.

#### `vtvm`

Első ránézésre szinte teljesen azonos a `twm`-mel: mind a menü, mind a beállítások felépítése megegyezik. A beállításokat a `.vtvmrc` fájl tartalmazza a sajátkönyvtárunkban. A rendszer-szintű beállítások a `/etc/X11/rvtm` könyvtárban szerepelnek.

#### `gwm`

Alapvető képernyőkezelő program. A Debianban található változat nem tartalmaz beállítóeszközöket.

#### `uwm`

Saját menüvel is rendelkezik, amelyet mi magunk is szerkeszteni tudunk. A teljes rendszerre érvényes beállításokat a `/etc/X11/ude` tartalmazza (furcsa, hogy az `uwm` nevű ablakkezelő beállítófájljai az `ude` könyvtárba kerültek).

#### `wm2`

Alapszintű ablakkezelő, hiszen csak X-terminál indítására alkalmas, illetve ikonra változtatásukra és újbóli megjelenítésükre képes.

#### `flwm`

A Debian saját menüjét veszi át. Beállítási lehetőségei igen szegényesek. Egyetlen saját lehetőséget tartalmazott: több virtuális képernyőt tudunk vele kezelni.

#### `failsafe`

Nem is tudom igazán eldönteni, hogy a `failsafe` külön ablakkezelőnek minősíthető-e, hiszen az X feltelepítésekor önműködően feltevéődik a rendszerre. Önálló beállításokkal nem rendelkezik. Indítás után egy fejléc nélküli X-terminál nyílik meg, amelyből különböző alkalmazásokat indíthattunk el. Én annak idején a StarOffice 5.2 indításakor alkalmaztam,

hiszen ha teljes képernyőn használtam, akkor az apró eltéréseket leszámítva Windows-szerű felületet adott.

### Egy lépcsővel feljebb

Ebbe a csoportba azokat az ablakkezelőket soroltam, amelyek némiképpen egyediek, tehát valamilyen önálló ötletet tartalmaznak – és ennek köszönhetően az előbb felsoroltaknál bizonyos szempontból jobbak. Természetesen ezek sem közelítik meg a komolyabb ablakkezelők szintjét, de a megfelelő helyen alkalmazva őket igen hasznosak lehetnek a felhasználók számára.

#### `amiwm`

A leírás alapján ez a képernyőkezelő felépítésében az Amiga ablakkezelőjére hasonlít. Ezt sajnos nem tudom eldönteni, mivel az Amiga felületét nem ismerem, de mindenképp érdekes felületet hoz létre, leginkább a MacOS-éra emlékeztetett (természetesen jóval egyszerűbb kivitelben). Az egér jobb gombjával a képernyő bal felső sarkában kattintva több menüt is kapunk, ezek közül a legelsőben található egy parancssor begépelését lehetővé tevő ablak. Ebből tudjuk elindítani az alkalmazásokat (ugyanítt lelhető fel az ablakkezelőből kiléptető `Exit...` menüpont is). Az így létrejövő ablakok teljes képernyőssé vagy ikonméretűvé tehetők, illetve az ablakok jobb felső sarkában lévő három ikonnal a háttérbe küldhetők. Az ablakok a bal felső sarokban szereplő ikonnal zárhatók be.

#### `ion`

Első látásra ez az ablakkezelő is nagyon egyszerűnek látszik. Amennyiben elolvassuk a súgót, a leírását, illetve betekintünk a bennük hivatkozott fájlokba (`/etc/ion/`), rájöhetünk, a látszat igenis csal: az `ion` ugyanis számos billentyűkombinációt tartalmaz. Az általa megjelenített képernyő igen sajátos, mert minden ablakot teljes képernyős ablakként jelenítünk meg, még a legutolsó beviteli ablakot is. Néhány fontosabb billentyű a könnyed használathoz:

- F1 a súgó-oldal kiválasztása és megtekintése;
- Mod\*+F1 a súgó-oldal kiválasztása és megtekintése;
- F2 az X-terminálemulátor indítása;
- F3 parancsfuttatás;
- Mod\*+F3 a lehetőség indítása;
- F4 SSH-kapcsolat létrehozása valamilyen kiszolgálóval;
- F5 fájl szerkesztése;
- F6 a fájlnezegető indítása;
- F9 munkaterület készítése, amennyiben már létezik, átlépés rá;
- Mod\*+F9 a munkaterület lekérdezése;
- F11 megerősítés után újra azt kérdezi tőlünk, hogy újraindítsa-e az `ion-t`;
- F12 megerősítés után újra azt kérdezi tőlünk, hogy kilépjünk-e az `ion-ból`.

\* (Módosító billentyű, ALT vagy CTRL változatfüggő.)

Nagyon zavaró, hogy a normál billentyűk is tartalmaznak ablakkezelő szolgáltatásokat, ami sok esetben ütközik az indított alkalmazással. Ez volt az egyik legegyszerűbb ablakkezelő.

**Folytatjuk.**



Tóth Béla (tothb1@freemail.hu)

Nős, két gyermek büszke atyja. Dolgozott földmérőként, majd térinformatikus szakmérnöki képesítést szerzett. Egyaránt otthonosan mozog a CAD és a térinformatikai programokban, valamint a DOS- és Windows-alkalmazásokban.

Legkedveltebb elfoglaltsága már két és fél éve a Linux.

## Váltás PostgreSQL adatbázisrendszerre

Néhány hasznos tanács arra vonatkozóan, hogy miként cserélhetjük le meglévő Microsoft Access adatbázisrendszerünket.

**E**gyre több vállalat kezd nyílt forrású üzleti rendszerekkel foglalkozni, sokan közülük teljes rendszert igyekeznek kiépíteni, amely a vásárlók számára hozzáférhető webes felülettel az alapként szolgáló adatbázisig terjed. Legtöbb esetben a Linux-PHP-Apache hármast, illetve egy nyílt forrású adatbázist (MySQL-t vagy PostgreSQL-t) foglal magában. A PostgreSQL népszerűsége főleg az utóbbi időben növekedett meg, hiszen mára a program elérte azt a minőséget, amelynél már megbízhatónak és jól használhatónak mondható. Több nagyvállalat biztosít állandóan elérhető támogatást, közöttük a RedHat is. Írásomban azt szeretném megmutatni, hogy mire számíthatunk, ha a Microsoft Accessről nyílt forrású adatbázisrendszerre (itt és most a BSD-típusú felhasználási szerződéssel rendelkező PostgreSQL-re) térünk át.

*Michael Calabrese*, a Bike Friday nevű kerékpárgyártó cég adatrendszerének felelőse nemrégiben a változás mellett döntött. A Bike Friday egy villámgyorsan növekedő, túra- és hegyi kerékpárok gyártásával foglalkozó cég, székhelyük az Oregon állambeli Eugene-ben található. A vállalatnál PostgreSQL adatbázisban rögzítik az eladásokkal, a gyártással és a vásárlói támogatással kapcsolatos összes adatot. Calabrese mostanában épp a cég e-üzleti rendszerét szándékozik ingyenes programokra lecserélni: Linuxra, Apache-ra és PostgreSQL-re. Egyelőre azonban úgy tervezi, hogy megtartja a munkaállomásokon használt Microsoft Access 97-et annak érdekében, hogy az adatbázis lecserélése miatt szükséges szünet a lehető legrövidebb ideig tartson. Calabrese szerint: „Ha a munkaállomásokon használt kezelőprogramokat nem akarjuk megtartani, egyszerű a dolgunk: csupán le kell futtatni az átalakítóprogramokat és elkezdhetjük megírni az új kezelőrendszert. Ha a kezelőrendszer az Access, amelyet továbbra is szeretnénk használni a PostgreSQL adatbázissal, akkor a fejlődés új irányait úgy jelöltük ki, hogy az induláshoz nincs szükség az egész rendszer átalakítására. A kezelőfelület befagyasztása után

nyugodtan számoljunk egy évet az átalakításra. Ha a változtatásokat fokozatosan vezetjük be, megmarad a választási lehetőség, hogy egy-egy új lehetőséget Accessben vagy PostgreSQL-ben valósítsunk meg”.

### A változtatás eszközei

Amennyiben a Microsoft Open DataBase Connectivity (ODBC) meghajtókat betöltjük a PostgreSQL-sablonadatbázisba, máris megtettük az első lépést az Access és a PostgreSQL összeházasításához. Az együttműködéshez az alapvető átalakító eszközök mellett (lásd a *További érdekességek* című részt) néha további ODBC kiszolgálóoldali szolgáltatásokra is szüksége van. Ezeket az *src/interfaces/odbc/odbc.sql* fájlban találhatjuk meg. A PostgreSQL felületfüggetlen 4-es típusú Java adatbázis-kapcsolati felület (JDBC) meghajtót is tartalmaz. Továbbá a C számára készült beágyazott felület (ECPG) is a PostgreSQL részét képezi. A telepítés végeztével Calabrese adatösszesítő eszközöket választott, például a pgAccess – ez Windows és Unix változatban, illetve az exSQL 3.1-es változatában egyaránt hozzáférhető.

Miután a meglévő adatbázisokról a rendelkezésre álló eszközökkel (vagy a *pg\_dumpall* segédprogrammal, vagy pedig a *pg\_dump* és a *pg\_dumpaccount.s* együttes használatával) biztonsági mentést készítettünk és a telepítőt (*Installer*) is lefuttattuk, az adatok átalakításának első lépéseként az Accessben használatos, azonban a PostgreSQL-ben nem megengedett fájlnevek levadászása következett. Az Access meglehetősen szabadelvű, hiszen a fájlnevekben számos olyan karakter használatát lehetővé teszi, amelyeket más adatbázisrendszerek (Oracle, Sybase, PostgreSQL stb.) nem ismernek fel. Így a Bike Friday adatbázisában szereplő nevek jó részét a PostgreSQL által is kezelhető formájúra kellett alakítani, például az *Order Detail* táblázatból *Order\_Detail*, a *Shipped?* mezőnevekből pedig *Shipped* vagy *ShippedYN* lett.



Az alapvető átalakítóeszközök minden meg nem engedett karaktert eltávolítanak. Ez súlyos gondokat okozhat, hiszen a mit sem sejtő munkaállomási kezelőprogramok (nem értve a helyzetet) minden további nélkül megszákíthatják az adatbázissal való kapcsolatot.

Calabrese azt javasolja, hogy ha a kezelőprogramokat megtartjuk, ne írjuk át az adatbázis neveit, vagy pedig az adatbázist és a kezelőfelületeket is párhuzamosan módosítsuk. Ő maga ezt úgy oldotta meg, hogy az adatbázisban és a kezelőprogramokban saját kezűleg, egyesével írta át a kérdéses karaktereket.

### 1. lista Meglehetősen pazarló lekérdezés

```
SELECT Orders.SalesRepID
Bikes_ColorsAvailable.Color,
OrderDetails.BuildABikeID, Orders.OrderDate,
BuildABike.ColorID,
Bikes_BasicFrameTypes.FrameName,
Reps.Rep, BuildABike.Frame, Orders.CustID,
Orders.OrderID, OrderDetails.PartID,
Orders.ShipDate,
BuildABike.BikeState
FROM (Reps RIGHT JOIN Orders ON
Reps.RepID = Orders.SalesRepID) INNER JOIN
(((Bikes_BasicFrameTypes INNER JOIN
(OrderDetails INNER JOIN BuildABike ON
OrderDetails.BuildABikeID =
BuildABike.BuildABikeID) ON
Bikes_BasicFrameTypes.FrameTypeID =
BuildABike.FrameTypeID) INNER JOIN
Bikes_ColorsAvailable ON BuildABike.ColorID =
Bikes_ColorsAvailable.ColorID)
INNER JOIN Contacts_CurrentFrame_BABID ON
BuildABike.BuildABikeID =
Contacts_CurrentFrame_BABID.BuildABikeID)
ON Orders.OrderID = OrderDetails.OrderID
WHERE (((OrderDetails.PartID)=6502))
ORDER BY Orders.OrderDate DESC;
```

### 2. lista Ugyanaz a lekérdezés, de jóval gyorsabban

```
SELECT salesrepid, color, od.buildabikeid, o.orderdate,
bab.colorid, framename, reps.rep, bab.frame,
o.custid, o.orderid, partid, o.shipdate, bab.bikestate
FROM
reps, bikes_basicframetypes b_bft, orders as o,
orderdetails as od,
bikes_colorsavailable as b_co_a,
buildabike as bab
WHERE repid = salesrepid
AND od.buildabikeid = bab.buildabikeid
AND b_bft.frametypeid = bab.frametypeid
AND b_co_a.colorid = bab.colorid
AND o.orderid = od.orderid
AND bab.custid=[Forms]![Customers/Contacts]![ID]
AND entrydate = (
SELECT MAX(entrydate)
FROM buildabike as b2
WHERE bab.frame = b2.frame
);
```

Ez rendben is volt, hiszen a távolabbi tervek között a kezelőprogramok lecserélése is szerepelt. Ami a legfontosabb: a munka ezen szakaszában nagyon fontos, hogy folyamatosan ellenőrizzük, vajon minden működik-e. Ha a meg nem engedett karaktereket kiküszöböltük, az adatok máris készen állnak az átalakításra.

### Az adatok átalakítása

Ha azt tervezzük, hogy az Access megtartva kezelőfelületnek, első lépésként csak az adatokat ültetjük át PostgreSQL alá, a pgAdmin rendkívül hasznos eszköznek fogjuk találni. Calabrese az exSQL módosított változatát is felhasználta annak meghatározásához, hogy az Access és a PostgreSQL hogyan kezeli a táblázatok közötti kapcsolatokat. A [http://www.geocities.com/musica\\_6898/postgresaccess\\_home.html](http://www.geocities.com/musica_6898/postgresaccess_home.html) címen elérhető honlapján a nyilvánosság elé tárt változat egy héjprogramot futtat le, mely számos esetben módosítja a mezőtípus-átalakítás menetét (például az Access pénznemkezelését is). A Bike Friday Access kezelőfelülete a PostgreSQL számszerű decimális mezőit szövegmezőként értelmezte. Ennek érdekében, hogy az Access ezeket helyesen kezelje, Calabrese a mezőket *Float4*-típusúvá változtatta (a PostgreSQL így nevezi a négybájtos lebegőpontos számokat).

### A kezelőfelület kipróbálása

Száznál is több táblázatával a Bike Friday kezelőfelülete meglehetősen összetett. A felhasználó szemzőgéből nézve a Bike Friday több mint nyolcvan képernyőt használ a megrendelések beírásához, az alkatrésztáblázat megtekintéséhez, a gyártás ütemezéséhez és a raktárkészlet ellenőrzéséhez. Ezért Calabrese-nek biztosnak kellett lennie abban, hogy a rendszert több tucat felhasználó sem fagyaszthatja le. A kipróbálás jó néhány hétig tartott, közben az SQL-lekérdezéseket is szükség szerint módosítani kellett oly módon, hogy az Access-oldalról vagy (ha ez nem volt megoldható) az adatbázis oldalán újraírták őket, egészen addig, amíg elfogadható sebességet kaptak. Az 1. és 2. listában egy jellegzetes és egy gyorsított lekérdezést láthatunk.

A PostgreSQL-lekérdezések hatékonyabbá tétele általában a `Create index`, `vacuum`, `vacuum analyze`, `cluster` és `explain` parancsok használatával történik. Calabrese azonban figyelmeztet, hogy az Access alapértelmezett a lekérdezéseket nem közvetlenül továbbítja, hanem mindig az általa leghatékonyabbnak ítélt formára alakítja őket. Calabrese a program okoskodását közvetlen lekérdezés használatával kerülte meg, amely közölte az Access-szel, hogy hozzá ne nyúljon a lekérdezéshez, egyszerűen csak továbbítsa a kiszolgáló felé. A Bike Friday PostgreSQL-adatbázisának egyszerűsítése során Calabrese úgy ért el sebességnövekedést, hogy a lekérdezésekben az adatbázisból kisebb, pontosabban körülhatárolt adatokat vett ki. Százezer rendelési adat egyidejű lekérdezése