

Amikor a Palm és a Linux beszélgetni kezd egymással...

Két egyetemi hallgató megoldotta a Palm és a linuxos számítógép összehangolását.

A Palm nagyszerű hordozható készülék: jegyzetelhetünk, találkozókát tervezhetünk vagy akár naplót is írhatunk vele. Csodálatos, hogy mindenhol velünk lehet útközben. Belső hálózatunk kiszolgálója szintén bámulatos masina. Vállalati terveinket, a megbeszélések napirendi pontjait, az érdekes feljegyzéseket, a címeket és a teljes üzleti adatbázist tárolja. Ez a kiszolgáló Linuxot, Apache-kiszolgálót és egy MySQL-adatbázist futtat, amelyeket egy, a célra tervezett alkalmazásmotor fog össze.

Ugye, csodálatos lenne, ha a két gépet össze tudnánk kapcsolni? Bárcsak sikerülne a Palmról elérni a Linux-kiszolgálón található adatbázist – máris jó úton haladnánk. A leírás szűkszavú, az Internet azonban hatalmas. *Kevin* és *Jeffrey* két egyetemista, akik kis erőráfordítással új megoldást fejlesztettek ki. Ennek alapján végezhetjük el a megfelelő változtatásokat a Palm-gépeken és a vállalati kiszolgálón, s e módosításokat a másik eszköz adatbázisaival is végrehajthatjuk.

A lejjebb látható kódokat egy RedHat 6.x Linuxot futtató Intel-alapú gépen és egy Palm OS 3.5-öt futtató, soros bölcsovel rendelkező Palm Vx-en próbáltuk ki, de más összeállítás sem okozhat gondot. Az alkalmazott könyvtárak a Palm első piaci megjelenése óta (akkoriban még a 3Com gyártotta a gépet) nem változtak. Feltételezem (bár nem próbáltam ki), hogy a program a Visor-gépeken is gond nélkül működik, mivel ugyanazt az operációs rendszert használják. Első lépésként az egyszerűbb résszel kezdjük: a Palmot kapcsoljuk össze a kiszolgálóval. Először is a Palm bölcsojét kell összekötnünk a kiszolgáló soros csatlakozójával. Ezután létrehozunk egy pilot nevű eszközt, ami nem más, mint a soros kapu (esetünkben a `/dev/ttyS0`) másod neve:

```
ln /dev/ttyS0 /dev/pilot
```

Most már egy C program és a *HotSync* gomb megnyomásával megnyithatjuk a kapcsolatot a Palmmal. Miután a kapcsolat létrejött, már csak a Palm adatbázisaiból kell a megfelelő adatokat kiolvasnunk.

A Palm és a számítógép közti kapcsolat és adatátvitel a *pi* könyvtárral egyszerűen megvalósítható. Ez a könyvtár a BSD-csatolófelületet utánozza: létrehoz egy foglalatot (socket), hozzákapcsolja az eszközhöz, figyel a bejövő kapcsolati kérelemre és elfogadja. A bejövő kapcsolatot a Palm és a bölcso kettőse váltja ki akkor, amikor a felhasználó megnyomja a HotSync gombot. Az 1. listán láthatjuk, hogyan kell létrehozni egy Palm-kapcsolatra várakozó demont.

Miután a kapcsolat megvalósult, miként érhetjük el a Palm adatbázisait? Ezek mindegyike névvel rendelkezik. Az adatbázisokat a nevükkel nyithatjuk meg, majd meg kell adnunk az elérni kívánt rekordot, sőt az egész adatbázist is átnézhetjük. Macintosh- vagy Windows-gépeken ezt csővezetékek alkalmazásával oldják meg. A Palm is biztosít csővezetékeket e felületeken az összes, a Palm OS csomagba tartozó szabványos adatbázis számára. A Palm adatbázis-kezelő lehetővé teszi, hogy az adatbázisnak csak a módosított rekordjait

tekintsük végig. Módosított – mióta is? Nos, a legutóbbi összehangolás óta, amikor a kiszolgáló utoljára érte el ezt az adatbázist. Ezt tehát programjainkban az összehangolás után kell megtennünk – a 2. listán látható nyitott kapcsolatnál kell futtatni.

Amennyiben a Palm-adatbázisból rekordokat olvasunk ki, az nem számít összehangolásnak. Ennél többre van szükség, például arra, hogy a Palmra írunk, törölünk belőle és a saját MySQL-adatbázisunkból olvasunk. Mivel a MySQL-adatbázishoz való kapcsolódás ismertetése túlmutat e cikk keretein, most nem szólunk az összehangolás további részleteiről.

Kevin Velghe nagyszerű leírást tett közzé a témáról, amelyre a http://www.duo.be/palm/mysql_palm.html címen bukkanhatunk rá.



1. lista A Palm összehangolása

```
Main() {
    int sd;
    struct pi_sockaddr addr;

    sd = pi_socket(PI_AF_SLP,
PI_SOCKET_STREAM, PI_PF_PADP);
    addr.pi_family = PI_AF_SLP;
    strcpy(addr.pi_device, "/dev/pilot");
    pi_bind(sd, (struct sockaddr*)&addr,
sizeof(addr));

    sd = pi_listen(sd, 1);

    sd = pi_accept(sd, 0, 0);
    printf("Hurr! Lötrej tt a
        kapcsolat...");
    pi_close(sd);
}
```

2. lista A módosított rekordok átfésülése

```
{
    int db, len, I, attr;
    recordid_t id;
    unsigned char buffer[4096];
    ...l0trej n a kapcsolat...
    sd = pi_accept(sd, 0, 0);
    dlp_OpenDB(sd, 0, 0x40+0x80,
        ↪ "DateBookDB", &db);
    for (;;) {
        len = dlp_ReadNextModifiedRec
            ↪ (sd, db, buffer, &id,
            ↪ &I, 0, &attr, 0);
        if (len < 0) break;
        printf(buffer); printf("\n");
    }
    dlp_ResetSyncFlags(sd, db);
    dlp_CleanUpDatabase(sd, db);
    dlp_CloseDB(sd, db);

    pi_close(sd);
}
```

3. lista A pack függvény

```
{
    int app_size, len;
    recordid_t pal_id, new_id;
    unsigned char buffer[512];
    struct Appointment app;
    ...
    strcpy(app.description, "Tennival ");
    app.begin = ...
    ...
    size = pack_Appointment(&app, buffer,
        ↪ 512);
    palm_id = 0;
    len = dlp_WriteRecord(sd, db, 0,
        ↪ palm_id, 0, AppBuffer,
        ↪ Appointment_size, &new_id);
}
```



További érdekességek

Michael J. Hammel a Linux Journal 1998 júniusában megjelent „Linux and the PalmPilot” című cikke nem a legfrissebb, de ismerteti a pilot-xfer használatát, ami nagyon hasznos segédprogram. Elolvashatjuk a <http://www.linuxjournal.com/lj-issues/issue50/2711.html> címen.

A „PalmOS Desktop Howto” című HOGYAN-ja a <http://www.linuxdoc.org/HOWTO/PalmOS-HOWTO.html> címen található meg. Az <http://orbits.com>-ra mutató hivatkozások legtöbbje már nem működik, hiszen ez a vállalat minden bizonnyal megszüntette az említett HOGYAN-ok fejlesztését. Ha valakinek sikerül megtalálnia az elvesztett anyagokat, kérjük, küldje el őket a palm@duo.be címre, hogy a <http://www.duo.be/palm> címen mindenki számára elérhetővé tehesük.

A Palm-adatbázisban tárolt rekordok saját számokkal rendelkeznek. Amikor az eszközre rekordot írunk, ezt a számot kapjuk visszatérési értéként. Ezt a gépen vagy egy központi adatbázisban érdemes tárolnunk, így egy adott rekordot bármikor törölhetünk vagy frissíthetünk.

A `dlp_WriteRecord` egy Palm-rekordazonosítót fogad el. Amennyiben ez nulla, a Palm OS újat foglal le a számunkra; ha pedig létező azonosítót adunk át, akkor a megfelelő rekord kerül frissítésre. A legtöbb szabványos adatbázisrendszerben a rekordot a pack függvény csomagolja egy átmeneti tárolóba. Ez a folyamat a 3. listán látható.

A Palm azonosítása

Amennyiben egy kiszolgálóval (tulajdonképpen egy bölcsovel) több Palmot is használunk (mint ebben az esetben is), meg kell határoznunk, hogy éppen melyik Palm csatlakozik a bölcso-re. Amint a kapcsolat létrejött, hívjuk meg a `ReadUserInfo` függvényt:

```
{
    int db, len, I, attr;
    recordid_t id;
    struct PilotUser U;
    ...l0trej n a kapcsolat...
    sd = pi_accept(sd, 0, 0);
    dlp_ReadUserInfo(sd, &U);
    printf("Palm neve: %s", U.username);

    pi_close(sd);
}
```

Törölt rekordok

A Palm adatbázis-kezelő a rekordokat kiolvasás után nem törli – törlésre jelöli ki őket, de azt is megteheti, hogy a gépen vagy a kiszolgálón mentésre jelöli ki őket. Módosított rekord olvasásakor a fájltulajdonságokat ellenőrizni kell, amelyből megállapítható, hogy az adott rekordot törölni (vagy menteni) kell-e. Amint az adatbázis kitisztul, végérvényesen törlődik a Palmról:

```

{
    ...
    for (;;) {
        len = dlp_ReadNextModifiedRec(sd, db,
            ↪buffer, &id, &I, 0, &attr, 0);
        if (len < 0) break;
        if ((attr & dlpRecAttrDeleted) ||
            (attr & dlpRecAttrArchived))
            printf("T rlo$re kijel lve: %ld", id);
    }
}

```

Egy kicsi naplózás senkinek sem árt...

A Palm összehangolása után hasznos gyakorlat, hogyha a Palm naplófájljában megjegyzéseket is hagyunk róla. Akármint beleírhatunk, de az időpont és a dátum mindenképpen belekerül. Írjuk tehát az alábbi kódot programjaink végére:

```

{
    ...
    dlp_ResetSyncFlags(sd, db);
    dlp_CleanUpDatabase(sd, db);
    dlp_CloseDB(sd, db);

    dlp_AddSyncLogEntry(sd, "A Pilotr l
        ↪beolvastuk a m dos t$ sokat.\n");

    pi_close(sd);
}

```

Mi kell az induláshoz?

Ha a Pilotot linuxos géppel szeretnénk használni, szerezzük be a *pilot-link* csomagot. A kezelőfelületek számos rendszerhez elérhetők (Next, BSD, Solaris, OS/2, Linux stb.). A segítségükkel Python, Java, Perl, Tcl, C/C++ nyelvű programokat írhatunk. A szükséges fájl a <http://ryeham.ee.ryerson.ca/pub/PalmOS> címen elérhető FTP-kiszolgálón található meg, a neve: *pilot-link.0.9.3.tar.gz*, linuxos gépen minden gond nélkül lefordítható. A csomag tényleg több, mint csak egy csatolókönyvtár a használatát bemutató egyszerű példaprogramokkal. Ezek az egyszerű eszközök nagyon hasznosak: a segítségükkel teljes biztonsági mentést készíthetünk a Pilotról (és vissza is állíthatjuk), adatokat és adatbázisokat másolhatunk róla és vihetünk föl rá stb. A könyvtár C/C++, Perl, Python, Tcl és még néhány más nyelvből hívható meg. Aki rendelkezik némi programozási készséggel, az a csomagba tartozó példaprogramok és cikkünk listái segítségével könnyűszerrel összeállíthatja a számára szükséges eszközöket.

Az új programokat, a leírás kiegészítéseit, a megjegyzéseket, a HOGYAN-okat az olvasók a palm@duo.be címre küldhetik, mi pedig a <http://www.duo.be/palm> című honlapon mindenki számára elérhetővé tesszük őket.

Telepítsük a csomagot az alábbi paranccsal:

```
tar -xzvf pilot-link.0.9.3.tar.gz
```

Ez létrehozza a *pilot-link.0.9.3* könyvtárat, benne a forráskóddal. Lépünk is bele.

Adjuk ki a `./configure` parancsot, ami átnézi a rendszert a fordításhoz szükséges kiegészítő fájlokat keresve. A `configure` a programot alapértelmezés szerint a `/usr/local` könyvtárba telepíti. Ha ez nem megfelelő számunkra, a `./configure --prefix=K NYVT`R` paranccsal állítsuk be a telepítés helyét.

Adjuk ki a `make` parancsot, ami lefordítja a csomagot. Ekkor a fájlok még nem kerülnek a helyükre, így a végleges telepítés előtt alkalmunk nyílik kipróbálni a programot. Ha egy régebbi változatra egy újabbat telepítünk, ellenőrizzük, hogy minden működik-e. Általában természetesen semmiféle gond nem szokott adódni.

Rendszergazdaként adjuk ki a `make install` parancsot, mely a fájlokat a megfelelő könyvtárakba helyezi. Ha rendszergazdaként nem tudunk belépni, akkor olyan helyre telepítsük a programot, amelyhez rendelkezünk írási jogosultsággal. Ne felejtjük el a futtatható fájlokat tartalmazó könyvtárakat beírni a rendszer alapértelmezett útvonalába (`PATH` változó). Nézzünk meg a csomag mellé kapott hasznos példaprogramokat is, amelyek leírását a *További érdekességek* részben említett helyen olvashatjuk el.



Johan Coppieters

(palm@duo.be) a Duo nv nevű cég vezetője. A belgiumi Brugesben székelő vállalat Belgium legnagyobb vállalatai számára készít weboldalakat, belső hálózatokat és internetes alkalmazásokat.

Kevin Velghe

írta a C nyelvű Palm-Linux összehangoló programot, s egy három hónapos iskolai gyakorlat cseretanfolyam közben a kutatás egy részét is ő végezte. A Duo nv-nél találhatjuk meg.

