

Debian GNU/Linux 2.2 Potato

Gyakorlati útmutató a Debian rendszer telepítéséhez és beállításához.

Első lépések

Fontos kiemelni, hogy a mellékelt két lemezhez tartozik egy harmadik is. Ez a lemez sokmindent tartalmaz, de szerencsére az alaprendszert fel tudjuk telepíteni az első kettő segítségével. A telepítés megkezdése előtt fontos, hogy ismerjük számítógépünk kiépítését, alkatrészeit. Ehhez a következő adatokra lesz szükségünk:

- a merevlemez(ek) és a CD-ROM(ok) típusa, mérete, csatolórendszere (IDE, SCSI),
- a memória (RAM) mérete,
- hálózati kártyánk adatai,
- a VGA-kártya típusa, lapkakészlete és memóriamérete,
- monitorunk felbontása és frissítési frekvenciája,
- az egér jellemzői (csatolófelület típusa: PS/2 vagy soros),
- a modem típusa (külső vagy belső, ha külső, akkor melyik kapura csatlakozik).

Fontos kérdés, hogy merevlemezünkön van-e már operációs rendszer, például Windows, DOS stb. Ha nincs, akkor egyszerűbb a helyzet, ha viszont van, és a lemezen nincs szabad hely új lemezrész (partíció) létrehozásához, akkor helyet kell készíteni. Szerencsés a helyzet, ha a lemezen FAT vagy FAT32 típusú lemezrész van, ezekhez a típusokhoz ugyanis találunk segédprogramot, mellyel a meglévő lemezrész hátulsó, üres részét leválaszthatjuk, és újat hozhatunk létre.

A helykészítéshez vagy egy külön erre kifejlesztett terméket (például a PartitionMagic), vagy az első lemezen lévő Fips programot tudjuk használni. Az új rész készítése három lépésből áll:

1. Hibaellenőrzés (ScanDisk),
2. Töredezettségmentesítés (Defrag), valamint
3. a fips.exe használata.

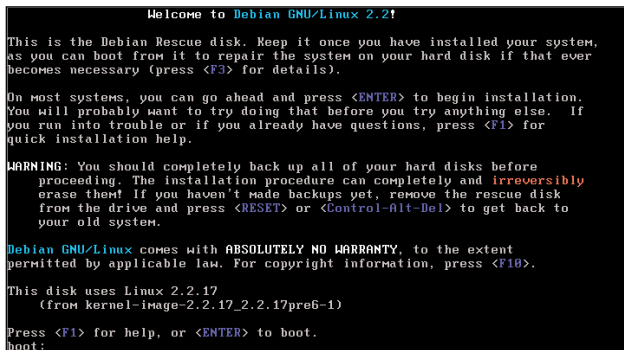
Mielőtt folytatjuk, külön felhívom mindenki figyelmét, hogy sokkal biztonságosabb minden adat elmentésével egy teljes lemezrész kiírni, és csak ezek után változtatni meg annak méretét! Akárhogy is döntünk, mindenképpen készítsünk biztonsági mentést a legfontosabb állományokról!

Lássunk hozzá. Az első két program Windows és DOS alatt is elérhető, a fips.exe-t pedig az első CD tools/fips.zip állománya tartalmazza. Ezt az ugyanitt lévő unz51x3.exe önkicsomagoló fájlban található unzip paranccsal lehet kicsomagolni.

Először a ScanDisket futassuk le az adott lemezrészén. Ez ellenőrzi a fájlrendszert, és kijavítja az esetleges hibákat, majd a töredezettségmentesítő (Defrag) programmal rendezzük a merevlemez tartalmát a meghajtó elejére. Ha ezzel elkészültünk és minden rendben lezajlott, akkor a keretben ismertetett módon a fips.exe programmal kettőszthatunk egy lemezrészre. Ezután jöhet a Debian Linux telepítése.

A telepítés indítása

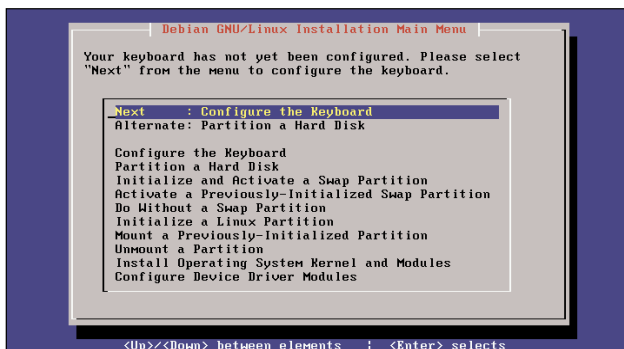
Az első lemez alkalmas a rendszerindításra. A legegyszerűbb, ha beállítjuk számítógépünket (a BIOS-ban), hogy CD-ről induljon a rendszer, majd az első lemez behelyezése után újraindítjuk a gépet. Amennyiben gépünk erre nem képes, akkor saját magunknak kell elkészíteni az indítólemezeket (az Install könyvtárban lévő rescue.bin és root.bin fájlokat a rawrite2.exe programmal írjuk ki egy-egy 1,44 MB-os hajlékonylemezre), majd a hajlékonylemezek segítségével indítsuk újra a rendszert.



1. kép A telepítés bejelentkező ablaka



2. kép Szerzői jogi információk



3. kép A beállítások megkezdése

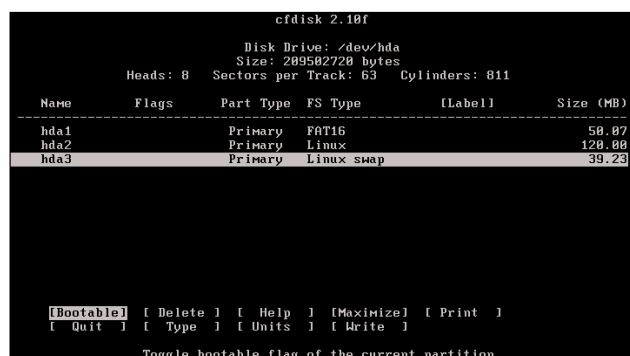
- A bejelentkező ablakban három választási lehetőségünk van (1. kép):
- ENTER A telepítés elindítása.
 - F1 A súgó megtekintése.
 - F10 A szerzői joggal kapcsolatos adatok.

Telepítés

Az első oldal a Debian GNU/Linux 2.2-vel kapcsolatos szerzői jogi leírást tartalmazza (2. kép), ezután jutunk el a telepítő menüjébe (3. kép), ahol elsősor a billentyűzet-kiosztást kell beállítanunk (4. kép). Amennyiben magyar kiosztást szeretnénk használni, akkor válasszuk a



4. kép Billentyűzetkiosztás beállítása



5. kép Cfdisk

„Hungary” pontot. Ezután a merevlemez részekre osztása következik. Habár a profi rendszerek (biztonsági és megbízhatósági szempontokból) gyakran öt-hat lemezzésre vannak szétvágtva, otthoni használathoz két lemezzész (egy adatterület, valamint egy csereterület) tökéletesen megfelel.

Lemezzések készítése: a cfdisk

Ezzel a programmal is nagyon vigyázzunk (lásd 5. kép)! Egy óvatlan mozdulattal lemezzéseket törölhetünk le. Ha a fips programmal készítettünk egy új, üres lemezzést, akkor azt most töröljük. A merevlemez szabad területén (FREE space) hozzunk létre két új lemezzést (New menüpont), a fájlrendszer számára egy Linux saját (83 – Linux native) típusút, a csereterületnek egy Linux swap (82) típusút. Ha kevesebb, mint 64 MB-unk van, akkor a csereterület legyen legalább kétszer akkora, mint a gépben lévő memória mérete, egyébként elég a memóriával megegyező méret. A Write menüponttal írhatjuk fel a merevlemezre az új felosztást. Itt még utoljára meggondolhatjuk magunkat, a megerősítés után azonban a módosítások életbe lépnek!

A fájlrendszer

Első lépésként állítsuk munkába a csereterületet. Egy ablakban kiválaszthatjuk az e célra használni kívánt lemezzést. A listában csak azok a részek szerepelnek, amelyeket a cfdisk-ben Linux swap típusúként adtunk meg. Ezután, ha szeretnénk a telepítő átvizsgálja a merevlemez felületét, így a későbbiekben rendszerünk nem használja a lemez hibás területeit. Ha a következő üzeneten is továbblépünk, a csererész már fel is éledt. Ezt követi a tényleges linuxos rész beállítása, mely a fentiekhez hasonló módon történik. A 2.2-es rendszermag (kernel) újabb típusú fájlrendszert használ, mint korábbi testvérei. Ha a merevlemez nem használjuk ennél korábbi változatszámú rendszermaggal, akkor az első kérdésre nemmel válaszoljunk (6. kép). Célszerű ezt a részt is leellenőrizni (nagy lemezzész esetén a folyamat

sokáig tarthat), majd a terület formázása következik. A feltett kérdések megválaszolása után fűzzük be (mount) a lemezzést a fájlrendszerbe (a Linuxokon az egyes lemezzések nem külön betűt kapnak, hanem az egyetlen „/” könyvtárból induló fájlrendszer egyes pontjainál lehet befűzni őket, így egy-egy alkönyvtárként jelenik meg minden lemezzész). Most már telepíthetjük a rendszer alapját és a rendszermag moduljait.

Modulok

Ebben a menüpontban beállíthatjuk a hálózati kártya, a hangkártya és az egyéb eszközök meghajtóit és betölthetjük azokat a memóriába. Például, ha PCI-os Ne2000 típusú vagy ezzel azonosan működő hálózati kártyánk van, akkor a net menüpont alatt a ne2k-pci modul fellesztve tudjuk az eszközünkhöz tartozó meghajtót betölteni a memóriába. A PCI-os csatlóú eszközeinket általában a rendszer automatikusan be tudja állítani.

Az alaprendszer telepítése

Most egymás után számos kérdés következik. A számítógépnév alapbeállítása debian, de természetesen tetszőleges nevet lehet adni a gépnek. Ezután ki kell választanunk, hogy milyen forrásból kívánjuk telepíteni a rendszert. Itt adjuk meg a /cdrom-ot (7. kép). Majd az

Útmutató a fips.exe használatához

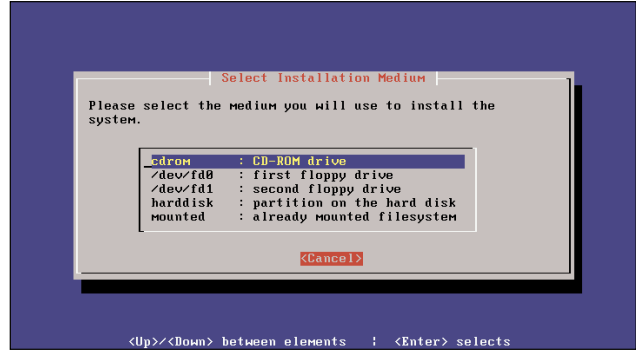
A fips program egy olyan segédeszköz, amellyel adatvesztés nélkül oszthatjuk két részre a merevlemezünkön már meglévő DOS lemezzéseket. (A fontos adatokról készítsünk mentést!) Hibaellenőrzés (ScanDisk) és töredezettségmentesítés (Defrag) után készítsünk egy DOS indítólemezt, ezt DOS alatt általában a sys c: a: vagy a format a: /s paranccsal hozhatjuk létre. Másoljuk rá a RESTORRB.EXE, FIPS.EXE és ERRORS.TXT fájlokat. Próbáljunk meg erről a lemezzéről indítani a számítógépet. Nagyon fontos, hogy semmilyen lemezzgyorsító programot (például Smartdrive) ne használjunk. DIR C:\ paranccsal próbáljuk ki, tudjuk-e olvasni a merevlemez meghajtónkat. Ezután a fips parancs kiadásával elindíthatjuk a programot. (Menet közben bármikor megszakíthatjuk futását a CTRL+C billentyűkombináció lenyomásával.) A fips végignézi a gépben lévő merevlemez(ek)e)t. Ha több is van megkérdezi, hogy melyik lemezen található a felosztandó lemezzész. A lemez gyökérterületének ellenőrzése után, ha több mint egy lemezzésünk van, ki kell választani, hogy melyiket szeretnénk elosztani. A program megvizsgálja a lemez végén lévő szabad hely méretét. A BAL/JOBB billentyűkkel tudjuk a részek méretét változtatni. Ha készen vagyunk, nyomjunk ENTER-t. Egy utolsó biztonsági ellenőrzés után a program kiírja az új felosztást. Most két lehetőség közül választhatunk: r (javítás) / c (folytatás). Ha a c-t választottuk, akkor az utolsó kérdésre y-t nyomva az összes változtatás érvénybe lép, és a program leáll. A számítógép újraindítása után futtassuk le a fips programot a fips.exe -t kapcsolóval, ha hiba nélkül eljutunk a lemezzész méretének állításáig, akkor a CTRL+C billentyűkombinációval lépünk ki a programból. A program lehetőséget ad biztonsági mentésre, amit az A:\ meghajtóban lévő lemezzre ROOTBOOT.00x fájl néven ment el (ahol az x 1–9-ig bármilyen szám lehet). Visszaállításra a RESTORRB.EXE programmal lehetséges.

A program fejlesztői: Arno Schaefer, Gordon Chaffee

© Kiskapu Kft. Minden jog fenntartva



6. kép Régi típusú ext2 fájlrendszer használata



7. kép A telepítés forrása

időzónát, a földrészt és a várost kell beállítanunk. Az időzónánál két lehetőségünk van, én a saját időzóna alkalmazását ajánlom.

Végző simítások újraindítás előtt

A Linuxnak saját indításkezelője van, ez a LILO. Ez egy csöpp kis program, mely bekerül a merevlemez (vagy valamelyik lemezrész) bevezető területére, és a rendszer indításakor fut le. A gép induláskor a elsődleges indító területen (Master Boot Record, MBR) lévő programot indítja el. Ha elsődlegesen a Linuxot akarjuk indíttatni, akkor a LILO-t az MBR-be kell írni, ha más indításvezérlőt használunk (például a Windows NT-hez járót), akkor az adatok tárolására készített lemezrész elejére telepítsük (ekkor az idegen indításvezérlőnek kell elindítania a rendszert).

A legutolsó lépés az indítólemez létrehozása. Ezt érdemes elkészíteni, hiszen ha a rendszer a merevlemezről nem indul, akkor ezzel a lemezzel tudjuk feléleszteni. Végül indítsuk újra a rendszert (a telepítőlemez végük ki még az újraindítás előtt).

További teendők

Újraindítás után hátra van még a rendszer beállítása, finomhangolása és a csomagok telepítése. Engedélyezhetjük vagy tilthatjuk az MD5 titkosítást (8. kép), ezután az árnyékjelszavak használatát (ezt mindenképpen javasolom), majd a rendszergazda (root) jelszavát kell megadni (9. kép). Következő lépésként hozzunk létre egy új felhasználót: adjuk meg felhasználónevét, valódi nevét, jelszavát.

Ezek után a hordozható gépek által használt PCMCIA-alrendszerrel kapcsolatos kérdéseken kell átrágnunk magunkat. Amennyiben nem laptop számítógépre telepítjük nyugodtan távolítsuk el, majd megkérdezi a rendszer, hogy akarunk-e Internetről letölthető csomagokat is telepíteni. Erre jelen esetben nincsen szükségünk.

Most helyezzük be az első lemezt, a rendszer beolvassa a lemez tartalmát, majd helyezzük be a másodikat is, és válaszoljunk igennel (mikor az olvasás végez, válaszoljunk nemmel).

Mit telepítsünk?

Két telepítési eljárás közül választhatunk:

1. Simple – egyszerű.
2. Advanced – haladóknak.

Ha először telepítünk Debian Linuxot, akkor kényelmesebb az egyszerű módot választani. Ekkor egy egyszerű menübe kerülünk, ahol csomagokat választhatunk ki. Egy-két érdekesebb csomagot kiemelek:

C++ Dev	C ++ fejlesztői környezet
C Dev	C fejlesztői környezet
Database Pg	PostgreSQL adatbázis
Dialup	telefonos Internetkapcsolathoz szükséges fájlok
Dialup isdn	ISDN Internetkapcsolathoz szükséges fájlok
DNS Server	tartománynév-kiszolgáló
Fortran	Fortran programozási környezet
Games	játékok
German	német nyelvű környezet
Gnome Apps	Gnome alkalmazások és segédprogramok
Gnome Desktop	Gnome ablakkezelő
Gnome Games	Gnome játékok
Gnome Net	Gnome internetes alkalmazások
Imap	Imap levélkiszolgáló
Laptop	segédeszközök a laptoutulajdonosok számára
Newbie Help	leírások kezdők számára
News Server	USENET hírkiszolgáló
Obj C	programozási környezet az Objectiv C nyelvhez
Pyton	Pyton programozási környezet
Tcltk	Tcltk programozási környezet
Tcltk Dev	Tcltk fejlesztői környezet
Tex	Tex szövegformázási környezet
X System	teljes X Windows grafikus felület
X Core	az X Windows legszükségesebb összetevői

Kedves Olvasó!

Lapunkat Olvasóink igényeihez kívánjuk igazítani. A magazin mellékletére is az Önök által javasolt és igényelt programokat szeretnénk feltenni. Először egy sokak által szidott, sokak által viszont ünnevelt Linux-változatot mellékelünk, a Debian GNU Linux 2.2 Official Release (Potato) rendszer első két lemezét.

Mivel a rendszer három lemezből áll, és e számunk mellé csak kettő fért, a harmadikat a következő számunk mellékleteként szerezhetik be, egy kiegészítő CD-vel együtt, melyen a megjelent javítások, frissítések, valamint fontos és hasznos, de a Debianban hivatalosan nem szereplő programok is helyet kapnak. Ilyen például a StarOffice 5.2 irodai csomag, a Netscape böngésző legfrissebb változata, valamint a KDE 2.0 ablakkezelő és irodai csomag. Ötleiteket, kéréseiket szívesen fogadjuk, és legjobb tudásunk szerint igyekszünk eleget tenni azoknak.

A mellékleteken a fontosabb vagy érdekesebb linuxos programokat meg szeretnénk jelentetni, így például nyomom követjük a Netscape testvérét, a Mozilla Projectet is.

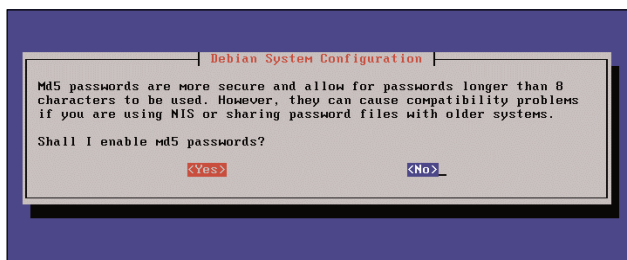
Ha grafikus felületet szeretnénk, akkor az X rendszert és a Gnome csomagokat érdemes telepíteni. A Dialup csomagra akkor van szükségünk, ha telefonon keresztül kívánunk kapcsolódni internetszolgáltatókhoz. A csomagok kiválasztása után a telepítés gyorsan lezajlik.

Az X Windows beállítása

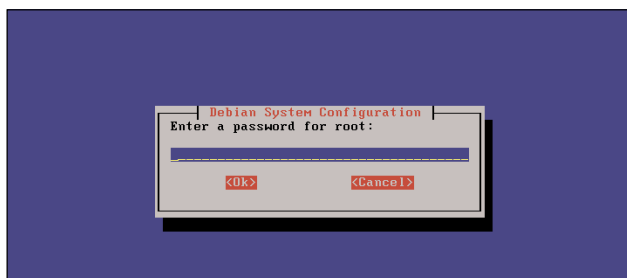
Az X Windows grafikus felület beállítása az anXious programmal történik. A program első lépésként megkísérli felismerni videokártyánkat. Ha sikerül neki, egyszerűen tovább tudunk lépni, ha nem, akkor nekünk kell kiválasztani a kártya típusát egy listából. Ezután dönthetünk, hogy a 75 vagy a 100 dpi-s betűkészletet használjuk, majd termináltípusokat választhatunk. Az X Windows grafikus rendszerhez számtalan ablakkezelő áll rendelkezésre, ezek közül bármelyiket választhatjuk, vagy akár többet is feltelepíthetünk, és a későbbiekben kitapasztalhatjuk, hogy számunkra melyik a legjobb és legkönnyebben alkalmazható.

Mikor továbblépünk, a program azt kérdezi, hogy az XDM elinduljon-e rendszerindításkor. Ha igennel válaszolunk, akkor a rendszer grafikus felülettel indul.

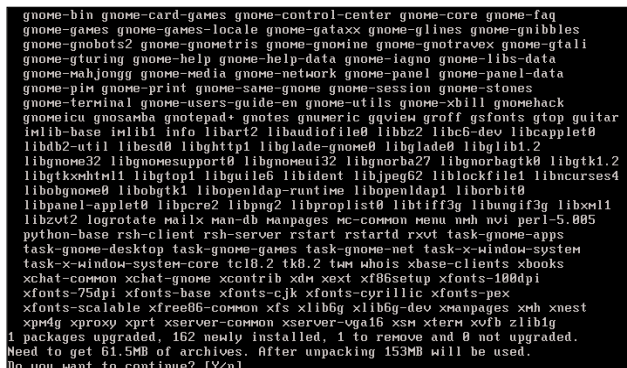
Az X rendszer alatt három gombot használunk, a harmadik gomb számos helyen a beillesztés parancsot hajtja végre, másutt egyéb hasznos szolgáltatást indít el. Ha egerünkön nincs középső gomb, azt kiválthatjuk a bal és a jobb gomb egyidejű megnyomásával, ha be-



8. kép MD5 jelszótítkosítás



9. kép A rendszergazda jelszavának megadása



10. kép Telepítendő programok listája

Behívás beállítása

Ha modemmel kívánunk csatlakozni internetszolgáltatókhoz, root -ként bejelentkezve indítsuk el a pppconfig programot. Hozunk létre új kapcsolatot, majd adjuk meg szolgáltatónk nevét, a protokoll típusát (többnyire a PAP megfelelő), a csatlakozáshoz szükséges jelszót, és végül adjuk meg, hogy milyen néven érhető el a rendszer számára a modem. Ez külső modem esetén többnyire a /dev/ttyS0 (COM1), belsők általában a /dev/ttyS2 (COM3) kapu találhatóak.

Az alapértelmezett átjárót (default gateway) nyugodtan hagyjuk jelenlegi értékén, és ne nyúljunk a IP-cím beállításához sem, csak akkor, ha szolgáltatónk állandó címet osztott ki számunkra. A modem sebességét és beállítóparancsát sem kell bántanunk, ha működik (gond esetén a modemhez járó leírásból kell kimazsolázzuk a megfelelő értékeket). Természetesen a felhívandó telefonszámot is meg kell adjuk.

Ezek után egy oldalon újra átfuthatunk a beállításokon, majd menthetjük a változásokat. No, ezzel is megvagyunk! Lépünk ki a programból, és kíséreljük meg a csatlakozást.

© Kiskapu Kft. Minden jog fenntartva

kapcsoljuk az Emulate 3rd button lehetőséggel. Az egér típusát is jó, ha ismerjük. Sőt, fontos adat, hogy melyik kapura csatlakozik. Ha ps/2, akkor /dev/psaux, ha pedig soros, akkor /dev/ttySx, ahol x értéke 0, 1, 2 vagy 3, a kapu sorszámától függően.

A monitor beállítása

A következő részben a monitor tulajdonságait kell megadnunk. Régi monitorokkal óvatosan bánjunk, ha túl nagy felbontást vagy frissítési gyakoriságot adunk meg, akkor bizony túlvezérelhetjük. Egy jobb 15"-os monitornál a 1280x1024, 60 Hz a gyakori beállítás. Ha nem vagyunk biztosak benne, inkább kisebb értékeket adunk meg. A grafikus felületen a CTRL+ALT+MÍNUSZ és PLUSZ (a számbillentyűzeten) parancsokkal tudunk menet közben felbontást változtatni. Ha több módot engedélyezünk, ezek között e billentyűkkel válthatunk.

A program létrehozza az XF86Config állományt, mely tartalmazza az összes szükséges beállítást.

Csomagok telepítése

Végere készen vagyunk! Most már csak egy ENTER-t kell üssünk, és a csomagok telepítése elkezdődik (10. kép). A telepítés után azonnal elkezdhetjük használni vadonatúj rendszerünket. Ha valaki internetes kapcsolatát is szeretné beállítani, indítsa el a pppconfig programot. Remélem, mindenki sikerrel járt, és már örömmel használja a Linuxot. Amennyiben a telepítés nem sikerült, ha tudok, szívesen segíték, illetve érdemes még körülnézni a linuxos oldalakon és levelező listákon is.

Kellemes linuxozást!

- ⇒ <http://www.debian.org>
- ⇒ <http://www.debian.hu>
- ⇒ <http://www.debianplanet.org>



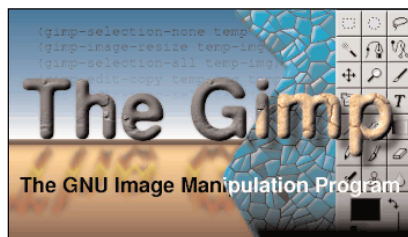
Csontos Gyula (Csontos.Gyula@linuxvilag.hu) a Linuxvilág hír- és CD-szerkesztője, valamint a www.linuxvilag.hu tartalomfelelőse. Szabadidejében szívesen mászik hegyet, kerékpározik és úszik.

gphoto

Ha digitális fényképezőgépinket Linux alatt is szeretnénk grafikus felületen használni, akkor a gphoto programot mindenképpen érdemes kipróbálni. Letölthetjük, módosíthatjuk és vissza is tölthetjük a kamera memóriájába képeinket. Nagyon sok digitális fényképezőgéphez találhatunk meghajtót az Agfától a Kodakon keresztül, a WWF kameráig. Ezek pontos listája megtalálható a www.gphoto.org/cameras.html oldalon. A gphoto jelenlegi változatszáma 0.4.3.
 ↻ www.gphoto.org
 ↻ www.gphoto.org/cameras.html

Gimp

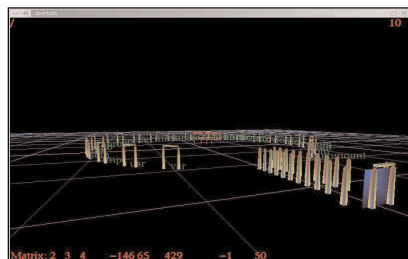
A linuxos világban a legnépszerűbb, ingyenesen hozzáférhető, nagy tudású képszerkesztőprogram a Gimp. Nagyon sok bővítésmény van hozzá, ezek megkönnyítik a munkánkat. Számos jól használható segítséget



kapunk internetes grafikáink megalkotásához, például előre elkészített gombok, feliratok háterek közül válogathatunk kedvünkre. Az új változat (az 1.2-es) már beépített dinamikus szövegkezelést is tartalmaz, sajnos azonban még csak fejlesztői állapotban van. Segítségével a grafikákba beépítendő szövegek kezelése egyszerűbbé válik.
 ↻ gimp.rulez.org
 ↻ www.gimp.org

Háromdimenziós „ablakkezelő”

Valójában nem egy egyszerű ablakkezelőről van szó, hanem egy alapszintű virtuális valóság (VR) kiterjesztésről. A fájlrendszert is háromdimenziós képként látjuk. A fejlesztők a későbbiekben teljes hálózatok 3D-s térképét is szeretnék láthatóvá tenni. Ebben a környezetben a felhasználók, ha van rá jogosultságuk, könnyedén elérhetnék a hálózat bármely gépét. Az alkotók szerint ez lenne az igazi CYBERSPACE.
 ↻ threesia.sourceforge.net



Digitális fényképezőgép CD-íróval

A Sony Mavica MVC-CD1000 fényképezőgép memóriakártya helyett 156 MB kapacitású CD-re menti a képeket. Ennek az eljárásnak köszönhetően 160 darab 1600x1200



képpontos képet tudunk tárolni. tízszeres optikai nagyítással rendelkeznek, ami megfelel a normál tükörreflexes fényképezőgépek 40–400 mm-es optikájának. A feldolgozás szempontjából ez természetesen számtalan előnnyel jár, nincs előhívási, tárolási költség. Rádásul számítógépünk CD-meghajtójába helyezve azonnal elérhetővé válnak a képek.
 ↻ http://64.14.40.97:80/explore_products/productindex.jsp

FreeDOS 32

A FreeDOS projekt célja rendszermagjának átültetése 32 bitre. Miért szükséges ez?



A fejlesztők szerint a DOS:
 • kicsi és egyszerű,
 • legmélyebb szintű közvetlen felügyeletet tesz lehetővé gépszinten,
 • nem többfelhasználós és nem többfeladatos, ez nagy előny, ha nincs ilyen tulajdonságokra igény,
 • csak egy alapszintű környezet az alkalmazásokhoz,
 • tökéletes a PC-ken futó beagyazott alkalmazásokhoz.

E tulajdonságainak köszönhetően még mind a mai napig több helyen használják. A DOS-t régi számítógépekhez tervezték, a processzor valós üzemmodját használja, ezért csak 1 MB memóriát képes kezelni. Az új 32 bites rendszermag kibővíti a memóriakezelést és a párhuzamos végrehajtás is lehetővé válik.
 ↻ <http://freedos-32.sourceforge.net>
 ↻ <http://www.freedos.org>

ICQ Linux alatt

Az ICQ megoldást jelent azok számára, akik szeretnék Linux alól is elérni az ICQ-t. Néhány szolgáltatása:

- File transfer – fájlcsere a felhasználók között,
 - URL – lehetőséget ad internetcímek cserélgetésére,
 - Invisible list – láthatatlanok maradhatunk a listában megadott felhasználók számára.
- ↻ www.licq.org

NetWhistle

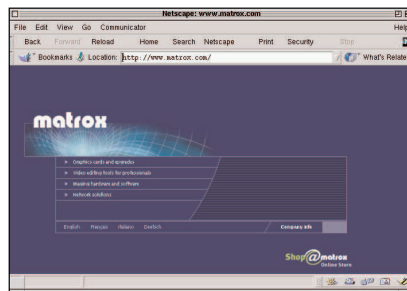
Egy érdekes és hasznos szolgáltatás kiszolgáltatót üzemeltetők számára. Folyamatosan figyelni a megadott számítógépet, ha elérhe-



tetlenné válik, akkor üzenetet küld a fenntartónak elektronikus levélben vagy mobil eszközre rövid szöveges üzenet formájában.

Grafikus kártya-meghajtók Linuxhoz

nVidia (tnt, tnt2, geforce stb.)
 ↻ <http://www.nvidia.com/Products/Drivers.nsf/Linux.html>
 Matrox
 ↻ <http://www.matrox.com/mga/support/drivers/latest/home.htm>



LME – A Linux hazai fellegrára

Ismerkedjünk meg a Linux-felhasználók Magyarországi Egyesületének rövid történetével és célkitűzéseivel.

A Linux-felhasználók Magyarországi Egyesülete (LME) céljával a GNU/Linux szabad operációs rendszer magyarországi megismertetését és elterjesztését tűzte ki, ezzel segítve a magyar gazdaság és a számítástechnikai kultúra fejlődését. Egyesületünk már megalakulásától kezdve minden szakmai rendezvényen igyekszik jelen lenni. Az LME üzemelteti a www.linux.hu-t, ahol minden érdeklődő elolvashatja a legfrissebb értesítéseket a rendszerről, illetve szándékaink szerint portálként is szolgál, hiszen számos linuxos cím is található az oldalon. Egyesületünknek több csoportja alakult, például Szolnokon és Veszprémbe, illetve létrehozták a KDE honi csapatát is. Ők elsősorban a KDE felülettel kapcsolatos munkákat kívánják összefogni: a felület alatt kifejlesztett hivatalos és „nem hivatalos” programok .po és sűgőfájljainak fordítását és a hazai fejlesztések támogatását végzik. Eddig is számos fordítás készült el, ezek egy része a KDE hivatalos változataiban már szerepel, és folyamatosan továbbítjuk az elkészült, illetve a szakmai bírálattal ellátott anyagokat. Fontosnak tartjuk, hogy a programok honosítását ne különállóan végezzük. Ennek megfelelően részt veszünk a magyar.linux.hu karbantartásában is. Ez egyben azt is jelenti, hogy – igény esetén – segítséget nyújtunk a más felületekre vonatkozó fordítási munkák összefogásában (ilyen például a gnome).

Tanítani...

A Fővárosi Pedagógiai Intézetben rendszeresen ingyenes előadásokat tartottunk a GNU/Linux alapjairól. Szó esett a GNU/Linux használatáról, a felépítéséről, valamint a hálózatokról és a biztonságról is. Jelenleg Szolnokon hallgathatnak hasonló témakörökben előadásokat az érdeklődők. Számos más helyszínen is szívesen látnák az egyesület oktatóit, de erőforrás-problémák miatt fel kellett hagyni az előadásokkal. Reményeink szerint januártól oktatóink ismét folytathatják tevékenységüket.

A GNU/Linux magyartásának megszervezésében és megvalósításában is tevékeny részt vállalt az egyesület, hogy ezáltal is megkönnyítse a GNU/Linux használatát (kézikönyvoldalak, programüzenetek, leírások fordítása).

A leírások honosításával párhuzamosan a magyar nyelvű oktatási anyag is hamarosan elkészül.

Az egyesület legfontosabb céljai közé tartozik az oktatások rendszeressé tétele, nem csak a jelenlegi helyszínen. Ezenkívül szeretnénk megalkotni egy teljesen magyarul beszélő GNU/Linux-rendszert, ehhez pedig minden segítséget szívesen veszünk.

LME naptár

- Az Info '99 kiállításon bemutattunk számos alkalmazást, valamint igyekeztünk megválaszolni az érdeklődők kérdéseit. Szakmai támogatóként, illetve előadókkal

is segítettük az ETB szervezésében tavaly májusban megtartott Ú Linux-konferenciát.

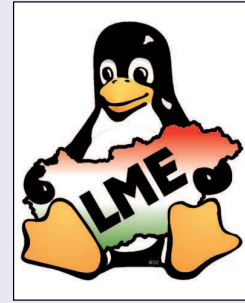
- 1999. szeptember 18-án megrendeztük az I. Linux Szakmai Konferenciát a MÁV BEIG épületében, nagy sikerrel, szakmailag elismert előadók részvételével, decemberben pedig eljött a várva várt Linux Mikulás. A résztvevők játékos formában vetélkedhettek: disztribúciós szaloncukorevészet és kötérlhúzás is szerepelt a programban.
- Jelen voltunk az Info 2000 kiállításon: képviseltük az egyesületet, az érdeklődők számára pedig leírásokat adtunk és szakmai tanácsadást nyújtottunk.
- Részt vettünk az ITB Konferencián, ahol a Nyílt forráskód: lehetőségek az EU és a Magyar Közigazgatás számára témakörében tartott előadásunkat elismerés fogadta.
- 2000. július 9-én az M1-es csatornán a Delta című műsorban elhangzott egy összeállítás a Linuxról, valamint egyesületünkről (ez a műsor letölthető a következő címről: http://www.linux.hu/inmedia/delta/delta_low.mpg).
- Ez év szeptemberében egyesületünk két előadást tartott a NetGeneration Konferencián, amely az elektronikus kereskedelemben résztvevő cégeket és az újságírókat célozta meg. Az egyik előadás az LME-ről és céljainkról, a másik pedig a Linux vállalati alkalmazásairól szólt.
- Október 7-én megtartottuk a II. Linux Szakmai Konferenciát, ahol a résztvevők két tagozatban hallgathatták meg a 12 előadást, továbbá a Morphy projekt is bemutatkozót tartott.
- Részt vettünk a Compfair2000 kiállításon, ahol naponta 4-5 előadást tartottunk a Linux-telepítés, a Postfix/Qmail, a Hacker-cracker és a Linux Clusters témakörökben. Ezenkívül az érdeklődők számára szakmai segítséget nyújtottunk.
- Idén ismét ellátogat hozzánk a Linux Mikulás, december 2-án a Fővárosi Művelődési Házba, ahová mindenkit szeretettel várunk. Megint lesz disztribúciós kötérlhúzás, tombola és vetélkedő. A belépés ingyenes.

Toborzó

Amennyiben Te is szeretnél belépni tagjaink sorába, vagy támogatni kívánod az egyesületet, nézd meg honlapunkat www.lme.hu, illetve írd az info@lme.linux.hu címre.

A magyar fejlesztők KDE alatti munkáiról kevés értesülésünk van, ezért várjuk azok jelentkezését, akik ezzel foglalkoznak. A levelezőlistánk címe: kde-csoport@qosip.ttt.bme.hu

Ha kedvet érzel ahhoz, hogy megosszad tudásodat a kidolgozott tárgykörök segítségével a tanulni vágyókkal, jelentkez nálunk az info@lme.linux.hu címen.



Kicsivel olcsóbban...

A *Linux Journal* 1999. augusztusi számában *Bruce Fryer* leírta egy látomását a Linux-alapú iMackel kapcsolatban. Ő valahogy így képzelte el: „Semmilyen mozgó alkatrészt (ventilátor, hajlékonylemez-meghajtó) ne használjunk, így még olcsóbb lesz. Mindent az alaplapra kell forrasztani, még a memóriát is, egy bővítőcsatlakozó kivételével. Egy nagyon egyszerű eszközzel van szó, ami megbízható, mint egy régi telefon. Lőjük be, mondjuk az 500 dolláros árkategóriát, hálózati csatlakozókkal, operációs rendszerrel, programokkal együtt. A pénzt főleg a memóriára és a kijelzőre fordítsuk, ami legyen mondjuk, aktív matrix, ha egyáltalán szükség van rá.” A cikk még folytatódik, de ennyiből is érezhető a lendület.

Egy ilyen masinára már nem is kell sokáig várnunk: a New Internet Computer (NIC) hasonló jellemzőkkel bír majd. A név ismerős lehet, hiszen itt *Larry Ellison*, az Oracle vezetője korábbi álmának újjászületéséről van szó. Míg az eredeti „NC”, vagyis Network Computer körülbelül egy okos terminálnak felel meg, a NIC egy linuxos munkaállomás. A 2.2-es Linux a CD-ről indítható. A processzor 266 MHz-es, a gépben 64 MB RAM van, 24-szeres sebességű CD-ROM, 2 USB csatlakozó, 10/100-as hálókártya, és billentyűzetet, egeret, valamint hangszórókat is kapunk hozzá. Az alkalmazásokért és az adattárolásért a hírek szerint egy távoli kiszolgáló felel majd, „bár internetes tárolásra is lehetőség van”. A gépen windowsos alkalmazásokat is futtathatunk, a Citrix MetaFrame ügyfél segítségével. Ára 199 dollár, 15"-os, 800×600-as felbontású képernyővel pedig 329,98 dollárba kerül majd.

A New Internet Computer Company elnöke *Gina Smith*, a régi szakíró, rádiós és tévés személyiség. Smith legutóbb a CNET News.com-jával működött együtt. A vállalat tervei között két olyan dolog is szerepel, mely komoly piacot teremthet az új gép számára. Az egyik, hogy Larry Ellison bejelentette: százmillió dollárt költenek az amerikai iskolák számítógépekkel való felszerelésére. A másik, hogy az Oracle nemrég 500 NIC gépet adományozott a chicagói általános iskoláknak, ezt újabb 500 követi a NACUBO (főiskolák és egyetemek gazdasági vezetőinek nemzeti szervezete) jóvoltából.

Szerző: *Doc Searls*

➔ <http://www.thinknic.com/>

Ők mondták

„Öt éven belül a kormányzat úgy kezeli, majd a nyílt forrású projektek indítását, mint most a lakásépítésekét.”

„Minden üzleti mintakép fontos eleme, hogy be kell perelni valakit. Ez az Amerikai Módszer.” (*Bill Weinberg, MontaVista*)

„Tanácsot olyankor kérünk, amikor tudjuk a megoldást, de azt gondoljuk, hogy bárcsak ne tudnánk.” (*Erica Long*)

„A szabadság alapfeltétele, hogy ne mások kénye-kedve szerint éljünk.” (*Arisztotelész*)

„Arra építenek, hogy majd a felhasználók készítenek alkalmazásokat a miniszámítógépekre, és nem a vállalat költségvetését terhelnének azzal, hogy hatalmas összegeket fordítanak programok kifejlesztésére és reklámozására. A digitális üzletkötők, az egymásnak árusító mérnökök a vásárlókkal való szoros és sokáig tartó kapcsolat kialakítását segítik elő... Meglepő viszont, hogy milyen kevés növekedést sikerült elérniük. Évekig a felhasználók által kifejlesztett érdekes alkalmazások vitték előre az egész ügyet.”

(*Tom Peters és Bob Waterman az In Search of Excellence* című, 1982-ben kiadott könyvükben írták a *Digital Equipment Corporation*ról.)

„Az idő érzékcsalódás. Az ebédidő még inkább.” (*Douglas Adams*)

„A betegségek gyógyíthatók, a sors nem az.” (*Kínai közmondás*)

„Jóslolni nehéz, különösen a jövővel kapcsolatban.” (*Niels Bohr*)

„Nem a legerősebb vagy a legokosabb fajok maradnak fenn, hanem azok, amelyek a leggyorsabban alkalmazkodnak a változásokhoz.” (*John McFee*)

„Ha ennek az írásnak a mondanivalóját egyetlen mondatba kellene sűrítanem, azt mondanám, hogy a Mount Everest csúcsa tengeri üledékből képződött mészkő.” (*John McFee – a geológiáról*)

„Én vagyok a szólásszabadság utolsó tökkelütöttje.” (*Howard Stern*)

„Felszállni nem muszáj. Landolni kötelező.” (*Graffiti egy kisebb repülőtéren*)

„Képzelje el, hogy a mindennap ugyanúgy bekötött cipője keddenként felrobban. Pontosan ugyanez történik a számítógépekkel is, de senkinek nem jut eszébe panaszkodni.” (*Jeff Raskin*)

„Határtalan butaságról tanuskodik, ha az ember csak egyféleképpen tudja egy szó helyesírását elképzelni.” (*Andrew Jackson*)

„Az ember nem hinni akar, hanem rájönni, hogy mi van a másik oldalon.” (*Bertrand Russel*)

Miért éppen Red Hatnek (Vörös Kalapnak) hívják?

„A Linux végül is a földből lett. És annyiban emlékeztet a kommunizmusra, hogy az emberek igen hamar, nagyon megszerették. Ja, és ingyen van.” (*Steve Ballmer*)

„A boldogságot az ember nem megtapasztalja, hanem emlékezik rá.” (*Oscar Levant*)

„Állj a bátrak közé. Úgyis ott van a legtöbb hely.” (*Anita Roddick*)

Beszélgetés Szergej Brinnel, a Google egyik alapítójával

Szergej arról mesél, hogyan vált a Linux a Web egyik legjobb keresőrendszerének alapjává.



A PC-k jelenleg nem csupán olcsóbbak, de pillanatok alatt beszerezhetők, hiszen hétköznapi árucikknek számítanak. Ez óriási előnyt jelent, így például mi is néhány hét alatt telepíteni tudtunk ezer számítógépet; ezt a többi típussal nehezen tehattük volna meg. (Szergej Brinn)

Jelenleg a Google az egyik legnépszerűbb internetes keresőrendszer, mely igen gyors és szinte mindig kizárólag a keresés témájához kapcsolódó címetek ad meg. A céget 1998-ban alapította Szergej Brin és Larry Page. Kettejük együttműködéséből egy új elveken alapuló keresőrendszer született, a PageRank. Azóta a Google komoly sikereket ért el – a Yahoo! is ezt a keresőmotort használja –, segítségével majd egymilliárd URL-hez férhetünk hozzá. Most következék a Szergej Brinnel, a Google egyik alapítójával és jelenlegi elnökével folytatott beszélgetés.

Miért éppen a Linux mellett döntöttek?

Larry Page-dzsel a stanfordi egyetem számítástechnikai doktori képzésében vettünk részt, itt fejlesztettük ki a Google-t. Munkánkhoz számos rendszer közül választhattunk. Eközben megismerkedtünk a HP, a Sun, az Alpha és a Linuxot futtató Intel gépekkel, s rengeteg tapasztalatot szereztünk az egyes rendszerekről. A Google indulásakor el kellett döntenünk, hogy melyiket akarjuk használni. A Linuxot választottuk, mivel ár-teljesítmény viszonylatban messze ezt találtuk a legjobbnak. A PC-k jelenleg nemcsak sokkal olcsóbbak, de pillanatok alatt be is szerezhetők, hétköznapi árucikknek számítanak. Ez pedig óriási előnyt jelent. Például néhány hét alatt tele-

píteni tudtunk 1000 számítógépet; ezt a többi típussal igen nehezen tehattük volna meg. Azt hiszem, ezt az előnyt még ma sem értékelik az érintettek.

A Linux tényleg jobb

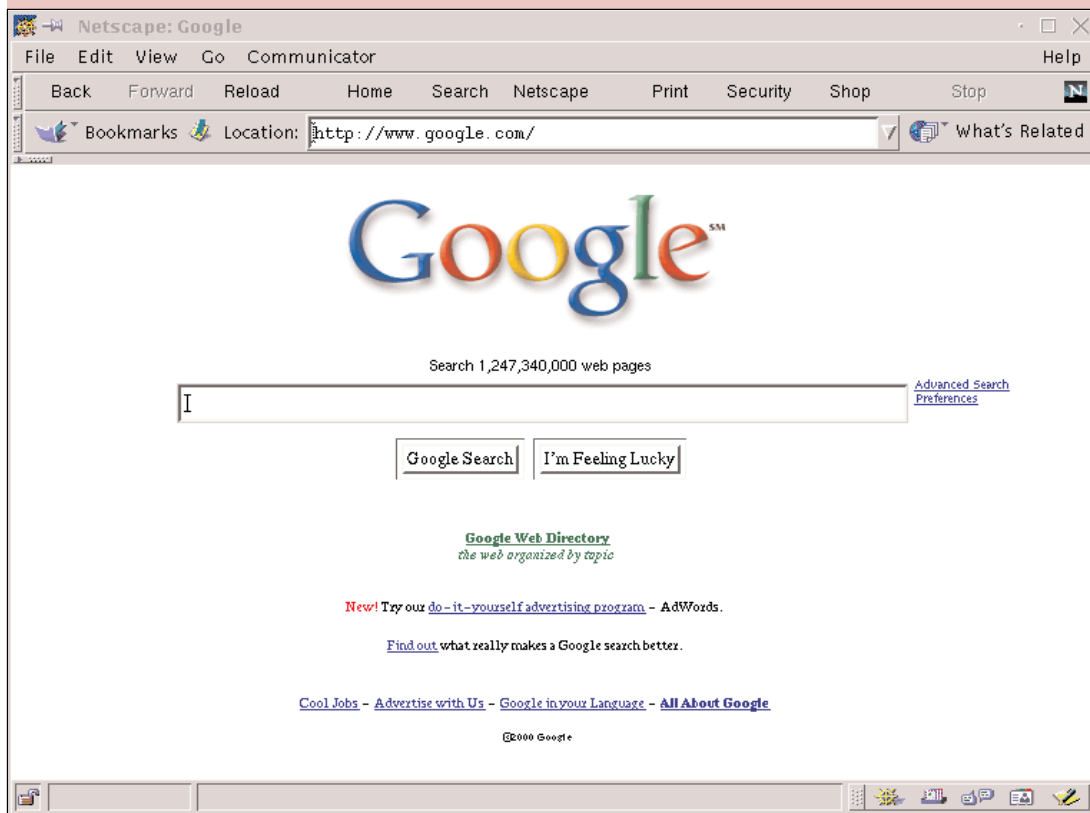
a többi rendszernél, vagy csak az ár számított?
Néhány szempontból a Linux a legjobb. Úgy éreztük, hogy céljaink megvalósításához e rendszer használatával kapjuk a legtöbb segítséget. Például a rendszermag fejlesztői személyesen adnak választ a kérdéseinkre, még hozzá igen gyorsan. Mióta a Google ennyire népszerű, azóta különösen segítőkészek: 15 perc alatt választ kapunk a kérdéseinkre, s azt hiszem, ez a gyorsaság verhetetlen. Ez ugyan nagyon fontos tényező volt, de be kell vallanunk, hogy leginkább a költségek szóltak bele a dologba.

A Red Hat-en keresztül jutottatok hozzá a szükséges információkhoz vagy a hírcsoportokból?

Általában tíz emberből álló csapattal dolgoztunk, akik igen sokat segítettek, emellett végigjártuk a hírcsoportokat és leveleztünk a szerzőkkel is. Ha egy feladatot nem tudtunk megoldani, akkor általában azonnal a szerzőkhöz fordultunk.

A Google-nál az asztali gépeken is Linuxot használtok?

Nem mindenki. Mérnökeink többsége Linuxot használ. Üzleti és hirdetési szakembereink inkább Windowszal





dolgoznak. Én Linuxot használok VMWare-rel, így bár-mikor átválthatok Windowsra. Sokan ezt két számítógéppel oldják meg, különösen azok a fejlesztők, akik a felhasználói felület kialakítását végzik, hiszen nekik Windows alatt is ellenőrizniük kell a rendszert. Számomra a VMWare használata egyszerűbb, mivel így csak egy gépen kell dolgoznom.

Mi hiányzik a Linuxból? Miben nem nyújt kellő segítséget?
Nagyon jól jönne egy 64 bites fájlrendszer, ezen ma is sokan dolgoznak, de a fejlesztés lassan halad. Úgy érzem, még mindig van tennivaló a megbízhatóság terén. Nem állíthatjuk, hogy a Linux az egyetlen rendszer, amely ezzel gonddal küzd, de a felhasználóknak megbízhatóságra van szükségük. Egy másik kérdés, hogy a 2 GB-os memóriahatár átlépésével a legkülönbözőbb nehézségek jelentkeznek. A hálózati veremk túlszűfálásakor szintén bajok adódtak: néha egyes számítógépekkel megszakadt a kapcsolat.

Jól gondolom, hogy rengeteg látogatótók van?

Igen, naponta körülbelül tízmillió keresési kérelem érkezik a Google.com-ra, s további hatmillió kérelmet kapunk az OEM-ügyfelektől. A világháló letapogatását nagyon gyorsan végezzük, így a rendszer folyamatos és igen nagy terhelésnek van kitéve.

Előfordult már teljes rendszerösszeomlás?

Nem, de egyes számítógépek néha lefagynak. A rendszerünket azonban úgy építettük fel, hogy a felhasználók kívülről nem veszik észre az ilyen nehézségeket.

Azt olvastam, hogy

saját hálózati telepítőeszközöket dolgoztatok ki...

Igen. Felhasználtunk mások által írt programelemeket is, de ezeket éppen mostanában írjuk újra. Jelenleg 5000 gépet használunk, és ezek telepítési feladatai óriási munkát igényelnek, ezért kialakítottunk egy saját hálózati telepítőrendszert, amellyel egyszerre 80 számítógépet tudunk kezelni. Ezenkívül egyedi ellenőrző és felügyelő programokat dolgoztunk ki; ezek tartják nyilván, hogy az egyes gépek éppen mivel foglalkoznak, milyen állapotban vannak stb. Ezt a felügyelőrendszert rengeteg munkával készítettük el.

A Google által használt 5000 gép csoportosítható a végzett feladat szempontjából? Tehát például 3000 gép végez kereséseket, 1000 foglalkozik OEM-mel, 500 a Web figyelésével.

Pontos számokat nem mondanék, a gépeknek körülbelül 80 százaléka elégíti ki a keresési kérelmeket; 10 százalékukat kutatási és fejlesztési célokra használunk, a maradékot pedig az anyagok előállítására, azaz a Web letapogatására és indexelésre.

Érdeemes volna ezeket az eszközöket nyílt forráskódként terjeszteni?

Jó kérdés. Nem túl sok olyan rendszerről tudok, amely méreteiben megközelíti a mi vállalkozásunkat, de biztosan akadnak olyan részletei, amelyeknek mások is hasznát vehetik. Jelenleg azonban egyetlen elem sem elég felhasználóbarát és kidolgozott. Bízom benne, hogy hamarosan elérjük azt a szintet, amikor már mások is közreműködhetnek a programok finomításában. Azt viszont biztosra veszem, hogy nem sokan fogják használni ezeket.

Mesélnél magadról, és arról, hogyan kerültél a Google-hoz?
Moszkvában születtem, hatévesen kerültem az Egyesült Államokba. Marylandben nőttem fel, majd a stanfordi számítástechnikai doktori program munkatársa lettem. 1993-ban itt kezdtem dolgozni, és elsősorban adatbányászattal foglalkoztam. Ez azt jelenti, hogy nagy mennyiségű adatot vizsgálunk át, s abban érdekes összefüggéseket, mintákat keresünk. Larry 1995-ben csatlakozott hozzánk, ő kezdte el letölteni az adatokat a világhálóról, mi pedig a hivatkozások szerkezetét vizsgáltuk. Azóta együtt dolgozunk.

Köszönjük, hogy időt szakítottál ránk, sok sikert kívánunk!

☞ <http://www.google.com>



Jason Schumaker (jason@ssc.com) már két éve dolgozik a Linux Journalnak. Ő a lap egyik segédszerkesztője, rendszeres szerzője, és amikor nem dolgozik, akkor a legkülönbözőbb sportokkal fásztja magát.

Linux-alapú óra

„Néhányan azt vallják, hogy a Linux kisebb rendszereken nem működik. Ez az állítás csak arra jó, hogy bebizonyítsuk az ellenkezőjét.” – *Takako Yamakura*, az IBM szóvivője ezekkel a szavakkal indokolta a karórákba szánt Linux megjelenését. Reutersnek kiadott közleményében pedig kiemelte: „a PC-kkel, a mobiltelefonokkal és más, vezeték nélküli készülékekkel együttműködő óra segítségével levelezhetünk, valamint rövid szöveges üzeneteket válthatunk.”

Hogy ne csak a kisebb gépekről essen szó, az IBM S/390-es nagygépeihez erőteljesen ajánlják a Linuxot. Az S/390 kiszolgálón máris több száz linuxos programot futtathatunk. A cég legutóbbi bejelentése szerint az év végére várható a Linux S/390-en futtatásához szükséges eszközök, és ekkorra jelenik meg a SuSE és a TurboLinux-változatok támogatása is.

☞ <http://www.s390.ibm.com/linux/>



dolgoznak. Én Linuxot használok VMWare-rel, így bár-mikor átválthatok Windowsra. Sokan ezt két számítógéppel oldják meg, különösen azok a fejlesztők, akik a felhasználói felület kialakítását végzik, hiszen nekik Windows alatt is ellenőrizniük kell a rendszert. Számomra a VMWare használata egyszerűbb, mivel így csak egy gépen kell dolgoznom.

Mi hiányzik a Linuxból? Miben nem nyújt kellő segítséget?
Nagyon jól jönne egy 64 bites fájlrendszer, ezen ma is sokan dolgoznak, de a fejlesztés lassan halad. Úgy érzem, még mindig van tennivaló a megbízhatóság terén. Nem állíthatjuk, hogy a Linux az egyetlen rendszer, amely ezzel gonddal küzd, de a felhasználóknak megbízhatóságra van szükségük. Egy másik kérdés, hogy a 2 GB-os memóriahatár átlépésével a legkülönbözőbb nehézségek jelentkeznek. A hálózati veremk túlszűfálásakor szintén bajok adódtak: néha egyes számítógépekkel megszakadt a kapcsolat.

Jól gondolom, hogy rengeteg látogatótok van?

Igen, naponta körülbelül tízmillió keresési kérelem érkezik a Google.com-ra, s további hatmillió kérelmet kapunk az OEM-ügyfelektől. A világháló letapogatását nagyon gyorsan végezzük, így a rendszer folyamatos és igen nagy terhelésnek van kitéve.

Előfordult már teljes rendszerösszeomlás?

Nem, de egyes számítógépek néha lefagynak. A rendszerünket azonban úgy építettük fel, hogy a felhasználók kívülről nem veszik észre az ilyen nehézségeket.

Azt olvastam, hogy

saját hálózati telepítőeszközöket dolgoztatok ki...

Igen. Felhasználtunk mások által írt programelemeket is, de ezeket éppen mostanában írjuk újra. Jelenleg 5000 gépet használunk, és ezek telepítési feladatai óriási munkát igényelnek, ezért kialakítottunk egy saját hálózati telepítőrendszert, amellyel egyszerre 80 számítógépet tudunk kezelni. Ezenkívül egyedi ellenőrző és felügyelő programokat dolgoztunk ki; ezek tartják nyilván, hogy az egyes gépek éppen mivel foglalkoznak, milyen állapotban vannak stb. Ezt a felügyelőrendszert rengeteg munkával készítettük el.

A Google által használt 5000 gép csoportosítható a végzett feladat szempontjából? Tehát például 3000 gép végez kereséseket, 1000 foglalkozik OEM-mel, 500 a Web figyelésével.

Pontos számokat nem mondanék, a gépeknek körülbelül 80 százaléka elégíti ki a keresési kérelmeket; 10 százalékukat kutatási és fejlesztési célokra használunk, a maradékot pedig az anyagok előállítására, azaz a Web letapogatására és indexelésre.

Érdeemes volna ezeket az eszközöket nyílt forráskódként terjeszteni?

Jó kérdés. Nem túl sok olyan rendszerről tudok, amely méreteiben megközelíti a mi vállalkozásunkat, de biztosan akadnak olyan részletei, amelyeknek mások is hasznát vehetik. Jelenleg azonban egyetlen elem sem elég felhasználóbarát és kidolgozott. Bízom benne, hogy hamarosan elérjük azt a szintet, amikor már mások is közreműködhetnek a programok finomításában. Azt viszont biztosra veszem, hogy nem sokan fogják használni ezeket.

Mesélnél magadról, és arról, hogyan kerültél a Google-hoz?
Moszkvában születtem, hatévesen kerültem az Egyesült Államokba. Marylandben nőttem fel, majd a stanfordi számítástechnikai doktori program munkatársa lettem. 1993-ban itt kezdtem dolgozni, és elsősorban adatbányászattal foglalkoztam. Ez azt jelenti, hogy nagy mennyiségű adatot vizsgálunk át, s abban érdekes összefüggéseket, mintákat keresünk. Larry 1995-ben csatlakozott hozzánk, ő kezdte el letölteni az adatokat a világhálóról, mi pedig a hivatkozások szerkezetét vizsgáltuk. Azóta együtt dolgozunk.

Köszönjük, hogy időt szakítottál ránk, sok sikert kívánunk!

☞ <http://www.google.com>



Jason Schumaker (jason@ssc.com) már két éve dolgozik a Linux Journalnak. Ő a lap egyik segédszerkesztője, rendszeres szerzője, és amikor nem dolgozik, akkor a legkülönbözőbb sportokkal fásztja magát.

Linux-alapú óra

„Néhányan azt vallják, hogy a Linux kisebb rendszereken nem működik. Ez az állítás csak arra jó, hogy bebizonyítsuk az ellenkezőjét.” – *Takako Yamakura*, az IBM szóvivője ezekkel a szavakkal indokolta a karórákba szánt Linux megjelenését. Reutersnek kiadott közleményében pedig kiemelte: „a PC-kkel, a mobiltelefonokkal és más, vezeték nélküli készülékekkel együttműködő óra segítségével levelezhetünk, valamint rövid szöveges üzeneteket válthatunk.”

Hogy ne csak a kisebb gépekről essen szó, az IBM S/390-es nagygépeihez erőteljesen ajánlják a Linuxot. Az S/390 kiszolgálón máris több száz linuxos programot futtathatunk. A cég legutóbbi bejelentése szerint az év végére várható a Linux S/390-en futtatásához szükséges eszközök, és ekkorra jelenik meg a SuSE és a TurboLinux-változatok támogatása is.

☞ <http://www.s390.ibm.com/linux/>

Beszélgetés Inder Singhel

Szerzőink a beágyazott Linux-rendszerekkel foglalkozó LynxWorks ügyvezető igazgatójával beszélgettek. A LynxWorks erősen bízik sikerében, lássuk, miért ilyen magabiztosak.



Dr. Singhel először New Yorkban találkoztam, amikor a LinuxWorldnél jártam. Rendkívül szívélyesen fogadott, és azonnal megtudtam tőle mindent, amire kíváncsi voltam. Szemmel láthatóan rajong a munkájáért. Reggeli közben beszélgettünk a beágyazott Linux jövőjéről, a BlueCat Linux 1.0 megjelenéséről, illetve a Linuxról általában.

Dr. Singh a LynxWorks ügyvezető igazgatója és elnöke, cége beágyazott alkalmazásokat készít OEM-partnerek számára. A beágyazott Linux-rendszerek jelenéről és jövőjéről faggattam.

Miért választották a Linuxot?

Milyen új fejlesztéseket, eszközöket hiányol belőle?

A LynxWorksnél (korábban Lynx Real-Time Systems volt a neve) mindig arra törekedtünk, hogy a valós idejű és a beágyazott rendszerek területén is nyilvános szabványok jelenjenek meg. Először létrehoztunk egy valós idejű operációs rendszert (RTOS, real-time operating system), a LynxOS-t, ez nagymértékben összekapcsolható volt a Unix rendszerrel (akárcsak most a Linux), és fontos szerepet játszottunk a POSIX szabványosításában. Elsőként valósítottuk meg a POSIX-megfelelő valós idejű kiterjesztéseket és szálakat, ezek ma már szinte minden Unix rendszerben megtalálhatók. A LynxOS volt az első olyan nem Unix rendszer, amely a POSIX követelményeit kielégítette.

Most úgy tűnik, hogy a Linux megvalósítja mindazt, amit a világ a Unix-tól várt: ez egy POSIX-megfelelő, nyilvános rendszer, mely sok géptípuson, számos szállítótól elérhető, ezenkívül rohamosan fejlődik a programellátottsága. Ebből következik, hogy a Linuxban hatalmas lehetőségek rejlenek számunkra, és természetes kiindulópontként szolgál a nyilvános rendszerek megvalósításához, mely fő célkitűzésünk. A LynxOS is POSIX-megfelelő és erősen valós idejű operációs rendszer, mely felbontható igen kis egységekre (a legkisebb kódméret 32 kB). A LynxOS számos más szempontból is sokkal közelebb áll a Linuxhoz, mint bármely más RTOS. Mivel a LynxOS POSIX-megfelelő, így gyakran forrásszinten is összeegyeztethető a linuxos rendszerekkel.

Ráadásul a LynxOS ugyanazt a fejlesztői környezetet használja, mint a Linux, hiszen mindkettőt a GNU eszközcsoport segítségével fejlesztették ki. A LynxWorks vásárlói rengeteg olyan kódot használtak fel a LynxOS-ben, amit a linuxos környezetben fejlesztettek ki (számos más Unix rendszerből is alkalmaztak részleteket, ilyen például a Solaris). Ez az újrafordítástól és újraépítéstől eltekintve igen kevés munkát igényelt. Ugyanilyen egyszerűen lehet

tett a Linux eszközmeghajtóit is átalakítani a LynxOS-hez. A rendszermag méretezhető, közbeavatkozásra kész „preemptive” (előjegyzéses) és valós idejű, kifejezetten a beágyazott valós idejű rendszerekhez fejlesztettük ki. viszont a LynxOS azon részeit, amelyek nem tartoznak a rendszermaghoz, nyílt forrásként fejlesztettük, és nagyon hasonlítanak a Linuxhoz. Ilyen például a GNU C/C++ fordító eszközcsoport, az X Windows, a TCP/IP hálózati csomag, számos BSDI segédprogram, valamint a Samba és az Apache. Mi itt a LynxWorksnél már alaposan megismerkedtünk és megbarátkoztunk a nyílt forráskódú fejlesztési mintával.

Így számunkra természetes lépés volt az első olyan RTOS-rendszer létrehozása, mely tartalmazza a Linux képességeit. A tavaly novemberben elindított Lynx Linux kezdeményezés (L2I, Lynx Linux Initiative) részeként bemutatottuk a BlueCat-rendszert. Ezt a Linux-változatot kifejezetten az olyan beágyazott rendszerekhez készítettük, amelyek a mi módszereinket és szerkezetünket használják. Ezeket a módszereket az elmúlt évtizedben fejlesztettük ki a beágyazott rendszerek felhasználóinak igényeit figyelembe véve. Emellett úgy döntöttünk, hogy továbbfejlesztjük a LynxOS-t is, és a LynxOS felhasználók számára elérhetővé tesszük a Linux számos előnyét.

A LynxOS későbbi változataiban teljes bináris hordozhatóságot teremtünk a Linuxszal. E lépésnek az a célja, hogy a Linux futtatható fájlljai LynxOS-rendszerben is használhatók legyenek, újrafordítás és egyéb átalakítások elvégzése nélkül (természetesen ennek feltétele, hogy mindkét gép hasonló gépcsaládba tartozzon).

Elmondaná, hogy milyen előnyei, illetve hátrányai vannak a Linux használatának?

A beágyazott rendszerek területén (melynek számos szabadalmaztatott operációs rendszer is részét képezi) a Linux elsőként ad lehetőséget arra, hogy egy nyilvános, sokak által fejlesztett rendszert hozzunk létre komoly program- és géptámogatással.

A nyílt forráskódú fejlesztési mintában az Internet játssza a döntő szerepet, itt a Linux minősége és sokoldalúsága kulcsfontosságú, és természetesen nem elhanyagolható az egyre nagyobb méreteket öltő népszerűsége sem.

A beágyazott rendszerek világában az ingyenes hozzáférés rendkívül vonzó a nagy beruházások és költségérzékeny alkalmazások számára. A forrás könnyen elérhető, emellett az óriási, ma is bővülő, Internet-alapú fejlesztői társadalom gyors és hatékony segítséget nyújt bármilyen új eszköz használatához, legyen az új processzortípus, vagy egy új adatkapu. A Linux ezáltal úttörő szerepet vállalhat az új módszerek és fejlesztések világában. Egyre több a Linuxhoz értő programozó és tanácsadó, ez szintén igen kedvező.

A Linux rendszermagot általános célokra fejlesztették ki, és a későbbi munkálatok a kiszolgálórendszerek igényeit szem előtt tartva zajlottak. A méretezhetőség nagy előnyt jelent ezen a téren a versenytársakkal (a Solarisszal és az NT-vel) szemben. Így a beágyazott Linux-rendszerek

A Linux nyilvános forrású és kereskedelmi alkalmazások számára is tökéletes kiindulópontként szolgál, így egyre nagyobb szerepet játszik a számítógépiparban, és különösképpen a beágyazott rendszerek világában.



fejlesztőinek (például a LynuxWorksnek) az a feladata, hogy a beágyazott rendszerek igényeire szabják a Linux lehetőségeit. Fontos területet jelentenek a több felületen egyidejűleg folytatott fejlesztések, a beágyazott rendszerek különleges eszközei és az itt előforduló géptípusok. A Linux beágyazott változatai számos felhasználási terület igényeit kielégíthetik. Vannak azonban olyan beágyazott rendszerek, amelyeknek jóval komolyabb valósidejű teljesítményre van szükségük, ezeknél továbbra is a szabadalmaztatott RTOS-felületek lesznek előnyben. E gondokat nem tudjuk kiküszöbölni egy olyan rendszerben, amit eredetileg más célokra fejlesztettek ki. A LynuxWorks elérhetővé teszi a Linux előnyeit az ilyen rendszerek számára is, hiszen az a szemlélet, hogy a Linux API-kat és ABI-kat a LynxOS segítségével kell megvalósítanunk. A nyilvános kód egyik legszebb lehetősége, hogy ugyanarra a nehézségre többféle fejlesztőtől eltérő megoldásokat találhatunk, így kiválaszthatjuk az igényeinknek legmegfelelőbb eszközt.

Milyen eszközöket,

fejlesztéseket látna szívesen a Linux-rendszerekben?

Egy gépfüggetlenséget létrehozó réteg (HAL) sokkal egyszerűbbé tenné a Linux áttemelését a különböző processzorrendszerek és felületek között. A beágyazott rendszerek esetében ez különösen fontos, de a kiszolgálói világ szempontjából is hasznos lenne, hiszen egyre többen döbbennek rá arra, hogy a Wintel-rendszerek csak a legalapvetőbb igényeiket elégíthetik ki. A LynxOS-ben az általunk kifejlesztett CSP/BSP (Chip Support Package/Board Support Package, lapkátámogató csomag, kártyátámogató csomag) rendszert használjuk, mely a processzorhoz kötődő, gépfüggő kódrészeket és a külső egységekkel foglalkozó elemeket elkülöníti az operációs rendszer egyéb területein található gépfüggetlen részeketől.

A cég neve mostanában változott Lynxről

LynuxWorksre. Miért történt ez, és honnan származik ez az érdekes írásmód?

A névváltozás utal a Linux elkötelezettségünkre. Az csupán véletlen, hogy korábbi nevünk (Lynx) egy kicsit hasonlít a Linux névre, és a Lynux írásmódban e kettő keveredik. A név jól mutatja, hogy továbbra sem szakadunk el a korábbi LynxOS megoldásoktól (hiszen az továbbra is alapvető és fontos része munkánknak), de emellett a Linux előnyeit is szeretnénk bemutatni a beágyazott rendszerek felhasználói számára.

Egyedi megközelítést valósítottunk meg azzal is, hogy vásárlóink számára kétféle rendszermagot adunk: a BlueCat egy nyilvános forrású Linux-változat, ezt a beágyazott rendszerek igényeihez igazítottuk, a LynxOS pedig egy Linux-megfelelő valósidejű operációs rendszer (RTOS), melynek háttérében az elmúlt évtized követelményei és feladatelemező alkalmazásai állnak. A felhasználók választhatnak a kétféle operációs rendszer közül. Ez teremti meg a vásárlóink igényeinek megfelelő

rugalmasságot, és lehetővé teszi, hogy a Linux előnyeit széles körben kiaknázzuk, például olyan alkalmazásokban, amelyeket korábban kizárólag beágyazott RTOS-rendszerek tudtak megvalósítani.

A beágyazott rendszerek szemszögéből a Linuxot kétféleképpen tekinthetjük:

- csodálatos, nyílt forrású operációs rendszer, mely számos beágyazott rendszerrel használható,
- egy nyilvános és szabványos felület programok futtatásához, a Linux API-k és ABI-k meghatározásának megfelelően.

Mindkét szempont igen fontos a Linux jelenlegi és jövőbeni sikerének szempontjából. Az előbbi még sok-sok változtatásra szorul a Linux mai állása szerint, utóbbi jelzi a fejlesztések jelentőségét. A Linux egyaránt tökéletes felületet ad a nyílt forrású és a kereskedelmi alkalmazások számára, így a Linux egyre fontosabbá válik a számítógépiparban, s ezen belül a beágyazott rendszerek birodalmában is.

A Linux fejlődésének üteme nem fog mérséklődni, és egyre nagyobb szerepe lesz, ha az alkalmazások nyilvános, szabványos felületeként széles körben elfogadottá válnak.

A unixos eredet és a POSIX-megfelelőség a Linux sikerének két legfőbb záloga, hozzátehetjük még harmadikként azt is, hogy forráskódja nyilvános.

A beágyazott rendszerek világában a Linux ezen jellegzetessége különösen fontos, hiszen itt egy erősen felabdalt piacról van szó, ahol semmilyen meghatározott szabvány vagy uralkodó géptípus, amihez igazodni lehetne nem létezik. A beágyazott rendszerek fele még ma is önálló, egyedi operációs rendszert használ, és a vezető kereskedelmi RTOS-rendszerek csak az új beágyazott alkalmazások kicsiny hányadát jelentik. Ennek következtében a beágyazott rendszerek területén nincsen olyan átfogó, ipari méretű programkészítés, mint amilyen a DOS és Windows környezetben kialakult.

A beágyazott alkalmazások fejlesztői rengetegszer megvalósítják ugyanazt, és szinte mindig mindent teljesen előlről kezdenek, ahelyett, hogy egy szabványos, alapeszköz-készletet bővítenének egyedi elemekkel, ahogy az más fejlesztési területeken szokás.

A Linux egy nyilvános szabvány, így elfoglalhatja a hiányzó közös felület helyét, s a beágyazott programkészítés alapja lehet. A beágyazott alkalmazások is egyre összetettebbé válnak, így a beágyazott és az egyéb rendszerek közötti határok kezdenek elmosódní. Ugyanazok a programelemek, amelyek jelenleg a linuxos rendszerekben elérhetők, a beágyazott alkalmazásokban is használhatóvá válnak.

Gratulálok a kinevezéséhez, mellyel a mostanában alakult Embedded Linux Consortium (ELC) vezetőjévé lépett elő. Milyen irányvonalat képvisel ön a testületen belül inkább gyakorlati szempontból közelíti meg a kérdéseket, és a szabványok megalkotásával foglalkozik, vagy az



üzleti/hirdetési oldalt erősíti, és az OEM-üzletfeleket biztatja arra, hogy beágyazott Linux-rendszereket használjanak?

Először is köszönöm. Ami a kérdést illeti, bízom benne, hogy a Linux fontos és hasznos szerepet fog játszani a beágyazott rendszerek világában. Valójában a Linux jobban használható a beágyazott rendszerek, mintsem a kiszolgálók megvalósításához, így az ELC nagy befolyásra tehet szert a jövőben.

Az ELC leginkább az üzleti élet és a kapcsolattartás területeinek kiépítésével foglalkozik, hiszen a beágyazott rendszerek világának bizonyítanunk kell a Linux előnyeit, de ez kétségtelenül a tagság igényei szerint történik majd. Nagyon fontos a beágyazott Linux világa számára, hogy ne ismétlődjön meg a Unix-háború. AZ ELC fontos szerepet játszhat abban, hogy a beágyazott Linuxot alkalmazók törekedjenek a szabványok (például a POSIX és a Linux Standard Base) betartására, bár jelenlegi terveink között nem szerepel, hogy az ELC közvetlenül részt vegyen a szabványok kidolgozásában.

Melyek azok a célok, amelyeket a beágyazott Linux-rendszerek fejlesztőinek a piaci verseny ellenére is együtt kell megvalósítaniuk?

A kíméletlen verseny ellenére el kell kerülnünk az újabb felaprózódást, és ez a legfontosabb cél, erről azonban az előző kérdésre adott válaszómban már beszéltem. A legfontosabb, hogy a Linuxot egy nyilvános, szabványos felülettel tudjuk alakítani az ilyen alkalmazások számára is, hiszen ezzel a beágyazott programkészítés egységeseen a Linux felé fordulhatna. A tagság egyetért abban, hogy a POSIX és az LSB csoportok szabványait kell felhasználnunk az egyetlen gyártó által kitalált új API-k és szabványok meghatározása helyett. Meg kell értetnünk mindenkiel, hogy az általánosan elfogadott Linux-szabvány közös érdekünk, még akkor is, ha egymással versenyzünk, és szeretnénk továbbra is használni egyéni megoldásainkat. Mindannyiunknak e célért kell dolgoznia, ha el akarjuk kerülni azt, ami a Unixszal történt az elmúlt évtizedben. Az is fontos a beágyazott Linux-rendszerek készítői számára, hogy szerepet vállaljanak a nyílt forrású fejlesztők közösségében, és tevékenyen közreműködjenek a nyilvános alapkódok létrehozásában, valamint pontosan megértsék annak szellemiségét. Én erősen bízom ennek megvalósulásában, annak ellenére, hogy a LynxWorks nemcsak nyilvános kódú hanem kereskedelmi termékeket is előállít.

Két kérdés a beágyazott Linux-fejlesztésekkel kapcsolatban. Mit ígér a beágyazott Linux világa az egyéb, kedvelt operációs rendszerekkel (Windows CE/ Windows Pocket PC) szemben? Melyek azok a területek, amelyeket a LynxWorks a beágyazott Linuxszal tervez meghódítani?

A PalmOS jelenlegi vezető szerepének és a Microsoft erős Windows CE- és PocketPC-fejlesztéseinek ellenére, a Linux képes felvállalni a közös alap szerepét. A PC-ken túli alkalmazások világa még ma is a változások korát éli. Hirtelen megjelentek az Internethez csatlakozó eszközök,

ezek az eljövendő egy-két évben a legfontosabb üzleti lehetőséget jelentik. A Linuxot sok gyártó támogatja, ezenkívül nyilvános, így a neve a beágyazott rendszerek világában is jól cseng. Több velünk együttműködő gyártó számos következő nemzedékes programot fejleszt a Linuxhoz. Az alacsony költségeknek, a függetlenségnek és a hatalmas fejlesztői társadalomnak köszönhetően a Linux a legmegfelelőbb választás.

A PC-független készülékeken túl szerepet vállalhatunk az autópárhuzamban, az eladóterminálok és a vezeték nélküli szolgáltatások területén, az Internet és az adatátvitel világában, valamint a hadiiparban. Meglepődve tapasztaltuk, hogy igen komoly az érdeklődés a Linux iránt, az általunk megcélzott területeken.

A Linux szabványos C könyvtára a GNU C könyvtár, a glibc, ami meglehetősen összetett. (A libc-2.1.3.so az én gépemem 900 kilobájtot foglal el, és egy statikusan fordított „Helló világ” program is több mint 200 kB.) Hogyan írhatnak a beágyazott rendszerek készítői kis programokat úgy, hogy közben megtartják a nagy fejlesztői és ellenőrző csapat előnyeit?

Ez az egyik olyan terület, ahol a beágyazott Linuxra fejlesztők megmutathatják, hogy mire képesek. Nemcsak a rendszermagot, hanem a fordítókat, könyvtárakat és a környezet egyéb részeit is át kell nézni. A könyvtárak esetében a cél az lehet, hogy csak a legszükségesebb kódrészeket építsük be a rendszerekbe, és minden felesleget vágjunk le.

Mindenki a felaprózódás elkerüléséről beszél, de már a megelőzésre alkalmazott elvek is szétessen látszanak. Mi az „egyetlen igaz út” a lehető legrosszabb elkerüléséhez? Egy operációs rendszer sikere a támogatott alkalmazások körétől függ. Tehát a felaprózódás elkerülésének legbiztosabb módja, ha sikerül egy erős, a teljes iparágra kiterjedő egyetértést kialakítani arról, hogy pontosan milyen egy „igazi linuxos alkalmazás”. Így a fejlesztők biztosak lehetnének abban, hogy programjuk minden linuxos rendszerben működni fog.

Az LSB (Linux Standard Base, Linux szabványos alap) tervezet elsődleges célja olyan szabványok kifejlesztése és elterjesztése volt, amelyek megteremtik a hordozhatóságot a különböző változatok között, és lehetővé teszik, hogy ugyanazok a programok különböző Linux-rendszerekben is futtathatók legyenek. Ugyanezzel a nehézséggel kell nekünk is megküzdenünk, s ehhez a fejlesztők és a felhasználók nagymértékű támogatására van szükségünk.

A beágyazott Linuxhoz rettentő sok az olcsó kártya, az ingyenes fejlesztői eszköz és az internetes adatszerzési lehetőség. Ha én kedvtelésből szeretnék létrehozni egy beágyazott Linux-rendszert (tegyük fel, szeretnék internetes autórádió-szolgáltatást megvalósítani, vagy a repülőmodellemhez óhajtok egy robotpilótát készíteni), akkor hogyan indulhatnék el?

Először is azt kell megvizsgálni, milyen követelményeket



támaszt az operációs rendszerrel szemben a létrehozandó rendszer. Ennek megfelelően kell megválasztani a gépet és a beágyazott Linux megfelelő változatát.

Egy kedvtelésből létrehozott alkalmazáshoz egy olcsó Pentium PC alaplap is elegendő. Egy kereskedelmi vállalkozás megvalósításához azonban érdemes megvizsgálni további felépítéseket is, például a SOC (system-on-chip, rendszer egy lapkán) megoldás sok kiegészítő lehetőséget teremt (adatátviteli vezérlők, megjelenítési vezérlők), és ezek könnyen csatlakoztathatók a kiinduló lapkához, így a beágyazott rendszer számos igényét kielégíthetjük.

Kereskedelmi változatok esetében kereshetünk olyan Linuxot gyártó céget, amely már ellenőrizte a beágyazott Linux-rendszerét az általunk használt processzor-típussal. Az egyszerű PC-ken ez az ellenőrzés nagyon alapos, de más alaplapokon vannak hiányosságok. Érdemes ellenőriznünk a rendszer hosszú távú megbízhatóságát is. Ehhez az ISO 9001 minőségi szabványt kell figyelembe vennünk. Ezenkívül érdemes megtudnunk, hogy a gyártó az elkövetkező tíz évben milyen változtatásokat tervez, vajon később is kapunk-e segítséget, jelenik-e meg frissítés a rendszerhez. A legtöbb beágyazott rendszer élete jóval hosszabb, mint egy egyszerű PC-é, tehát a megfelelő programfejlesztő megtalálása hosszú távú döntést jelent, és a teljes projekt sikerének egyik mérföldköve. Ezért válasszunk olyan processzor-családot, amelyhez legbiztosabb a támogatás a jövőben. Az Interneten is megkereshetjük a feladatunk megvalósításához szükséges eszközöket: linuxos meghajtókat, digitális hangolókat nagy számban találhatunk a legkülönbözőbb gépekhez. A Linux használatának az is nagy előnye, hogy a Világháló társadalma segítőkész, és a hasznos tanácsok mellett alkatrészekkel és programokkal kapcsolatos megoldás érhető el.

A TiVo a termékleírásában azt állítja, hogy bárki megkaphatja az általa módosított Linux rendszermagot egy bizonyos címről. Ez nagyszerű lehetőség a GPL követelményeinek kielégítésére és egy levelezési lista létrehozására az otthoni fejlesztők számára. Ezt az elvet fogják követni más cégek is, amelyek Linuxot használó eszközt készítenek?

Ez egy igen logikus megoldás a GPL követelményeinek kielégítésére az olyan eszközgyártók számára, amelyek Linux operációs rendszert használnak programjaik alapjául. Csak akkor van rá szükség, ha magát a Linux rendszermagot is megváltoztatják. Általában a fejlesztők csak egyszerűen felhasználják egy beágyazott Linux-rendszert (például a BlueCat Linuxot), és erre építik rá saját beágyazott alkalmazásukat, a mag módosítása nélkül. Ilyenkor a rendszer közzététele nem cél, hiszen a beágyazott operációs rendszer kifejlesztőjének feladata, hogy minden változtatáskor közzétegyék az új forráskódot. Ezt mi úgy érjük el, hogy termékeink eleve tartalmazzák a forráskódot, s az Interneten is minden elérhető.

A Linux rendszermag fejlesztői szeretnék olyan eszkö-

zöket használni, melyek tetszőlegesen beépíthetők és eltávolíthatók a rendszerből, például a PCMCIA kártyák vagy az USB-eszközök. Nem lesz túl bonyolult a beágyazott rendszerek magja az ilyen eszközök támogatása következtében? Mennyire kell a rendszermag fejlesztésének követnie a beágyazott Linux fejlesztőinek elvárásait?

Egyre több helyen használnak PCMCIA-s és USB-s eszközöket a beágyazott rendszerekben. A dinamikus beilleszthetőség támogatása az ilyen eszközöknél elengedhetetlen, főleg a beágyazott rendszereknél, ahol az újraindítás ritkán engedhető meg. A legfontosabb, hogy ugyanúgy, mint más kérdésekben, itt is a saját igényeink szerint formálhassuk a rendszert, tehát ne kelljen a kódot tárolnunk, ha nincs rá szükségünk.

Nem csak az USB- és a PCMCIA-eszközökről van szó. A dinamikus átállíthatóság minden beágyazott rendszerben fontos, akár eszközök felismeréséről van szó rendszerindításkor, akár az eszközök futás közbeni beillesztéséről és eltávolításáról. Ezért a rendszerfejlesztők gyakran kialakítanak egy belső, állandó magot, mely a rendszer alapját képezi, és a beállítási lehetőségeket megteremti. Különösen fontos ez a beágyazott eszközök esetében, ahol törekedni kell a legkisebb memóriahasználatra is. Így csak az éppen szükséges rendszermagszintű meghajtókat és alkalmazásokat kell beépíteni a rendszerbe az új eszköz észlelésekor. Bizonyos esetekben a beágyazott rendszerek fejlesztői a szükséges meghajtókat egy külön alkatrészen tárolják, melyet csak az eszköz telepítésekor használ a rendszer a meghajtó telepítéséhez. Az ilyen szintű dinamikus átállíthatóság a teljes rendszer hatékonyságát növelheti.

Terveztek további nyílt forrású programok közzétételét, melyet a beágyazott Linux fejlesztői felhasználhatnak? Mi a helyzet a multimédiával, a hang- és az írásfelismeréssel? Mostanában jelentettük be a BlueCat Linux LynuxWorks Messenger módszerének csatlakozását a nyílt forrású közösséghez, hogy új szabványokat határozhassunk meg a fejlett CompactPCI üzenetkezelőhöz az eddig kiadott és széles körben elérhető rendszerekben. Az üzenetkezelőt rendszermag-kiterjesztésként és a hozzá tartozó API-ként valósítottuk meg. A Messenger egy olyan módszer, melynek segítségével a CPU kártyák (akár alaplapok, akár nem) egyenrangú félként tarthatják a kapcsolatot a CompactPCI csatlakozón keresztül.

Szintén felajánlottunk a nyílt forrású közösségnek egy többszörös kiterjesztést a GDB hibakeresőhöz. Továbbra is szeretnénk értékes anyagokat adni a nyílt forráskódot használó fejlesztők számára.

Köszönjük, hogy időt szakított ránk!

*Don Marti (dmarti@linuxjournal.com)
a Linux Journal szakmai szerkesztője.*

*Jason Schumaker (jason@ssc.com)
a Linux Journal segédszerkesztője.*

Képzeld el, hogy akkor is kedvenc madridi csatornákat hallgathatjuk, ha éppen Seattle-ben van dolgunk. Ezt könnyen megtehetjük a PenguinRadio segítségével.

Interjú Andrew Laydennel, a PenguinRadio ügyvezetőjével

Az egyik legjobb internetes készülék a PenguinRadio. A céget Andrew Layden alapította, és szándéka szerint olyan internet-alapú szolgáltatást valósít meg, mellyel több ezer állomáshoz férhetünk hozzá a világ bármely tájáról.

Képzeld el, hogy akkor is kedvenc madridi csatornákat hallgathatjuk, ha éppen Seattle-ben van dolgunk. A PenguinRadio segítségével ezt könnyen megtehetjük. Bár a cég csak nemrégiben kezdte meg működését, s az igazi népszerűség még várat magára, a vállalkozás máris csodálatos eredményeket mondhat magáénak. A PenguinRadio kapcsolatba lépett a kábel nélküli adatátvitellel foglalkozó Ineva.com-mal, így nagy előnyre tett szert legközelebbi vetélytársával, a Kerbangóval szemben is. Az Ellipso (az Ineva egyik testvérvállalata) a jövőben a PenguinRadio rendelkezésére bocsátja műholdas megoldásait, így a csatornákat bárhol hallgathatjuk. Az Ellipsónak (ahogy ez a honlapjukról kiderül) három szabadalma is van „az elliptikus pályák és a telepítési jellemzők” szakte-

ilyen jellegű vállalkozásoknak. A világ legkülönbözőbb részein élő emberekkel leveleztünk és csevegtünk, így rengeteg tanácsot, illetve útbaigazítást kaptunk a rendszer kiépítéséhez.

Egy problémát azonban nem sikerült teljesen leküzdenünk: bizonyos körökben még mindig úgy vélekednek, hogy a Linux nem sokkal jobb az olcsó kis segédprogramoknál. Nekik megpróbáltuk elmagyarázni, hogy ez egyáltalán nem igaz, de hiába: azok, akiknek elegendő pénzüik és megfelelő programjaik vannak egy hasonló szolgáltatás megvalósításához, azok nem mindig értenek egyet a Linux használatával. Úgy gondolják, hogy csak az a rendszer vehető komolyan, amit pénzért árúsítanak, tehát nekünk is ehhez hasonló alapot kellene teremtenünk vásárlóink számára.

Előfordultak komoly alkatrésztűközések is?

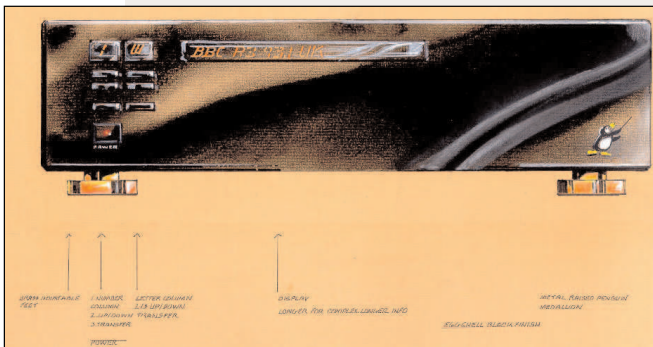
Kísérleteztünk néhány olcsó eszközzel, a tervezett 200 dolláros gyártási költség figyelembe vételével, ezek hangprocesszorát azonban igen szegényesen támogatja a Linux, így a használatukkal csak nehezen tudtunk elfogadható hangminőséget elérni.

Milyen operációs rendszert használnak a PenguinRadio munkatársai otthon és a munkahelyükön?

Többféle operációs rendszert használunk: a webes tervezést Macintosh gépeken végezzük, az irodai feladatokat Windows-rendszerek segítségével látjuk el, és a Linux teremti meg annak a feltételeit, hogy az egész hálózat összehangoltan és biztonságosan működjön. Vezető tervezőnk például külön webkamerával felszerelt linuxos gépet állított föl otthon, hogy munka közben is szemmel tarthassa kutyáit...

Mikorra lesz elérhető a rendszer a nagyközönség számára?

Célkitűzésünk az, hogy még ebben az évben elinduljon a szolgáltatás. Ha azonban azt szeretnénk, hogy a számítástechnikában járatlan felhasználók is boldogulhassanak vele, akkor még sok egyszerűsítést kell végrehajtani. Az emberek elvárják egy rádiótól, hogy ha bekapcsolják,



A PenguinRadio első változata

rületén. Egy ötletes megoldásnak köszönhetően a műholdak egymással is kapcsolatban állnak, így a Föld szinte bármely területén folyamatos szolgáltatást nyújtanak.

Andrew Layden-nel, a PenguinRadio ügyvezetőjével beszélgettünk a legutóbbi fejlesztésekről.

Miért éppen a Linux?

A PenguinRadio operációs rendszerét soha sem kérdőjeleztük meg. A Linux nemcsak a szakmai elvárásainkat elégítette ki – hiszen megbízható, olcsó és egyszerűen bővíthető –, de az üzleti gondolkodásmódunkhoz is nagyon jól illeszkedik. Úgy gondoljuk, hogy együttműködve másokkal olyan új rendszereket állíthatunk elő, amelyekkel egyre több ember számára tesszük elérhetővé az Internetet (és a Linuxot).

Megálta a helyét a Linux? Jelentkeztek problémák eddig?

A fejlesztés kezdetén néha komoly gondot okoztak az alkatrésztűközések, illetve a rádió bizonyos elemeihez nem találtunk meghajtóprogramokat, de a Linux-közösség bebizonyította, hogy képes öntetlenül segíteni az



A nagy vetélytárs, a Kerbangó



akkor az mindig működjön. A számítógépek azonban időnként lefagynak, bármikor történhet rendszerösszeomlás. Ezt egy rádió nem teheti meg. Olyan eszközt kell tehát készítenünk, amely az elvárásnak teljes mértékben megfelel, és ez nem egyszerű feladat.

Mennyit kell fizetnünk érte?

Bízunk benne, hogy készülékeinket 200–250 dollárért tudjuk előállítani. Ugyanis szerintünk ennyi pénzt költhet el egy házastárs a másik fél engedélye nélkül.

Milyen csatornákat hallgathatunk például az ausztráliai Cairns városból?

Adatbázisunkban jelenleg körülbelül 50 ausztráliai és óceániai rádióállomás szerepel, de naponta bővítjük a kínálatot. Úgy tűnik, hogy a szigetországokban (Új-Zélandon, Ausztráliában, Angliában) nagyon nagy az érdeklődés a szolgáltatás iránt, hiszen az ilyen helyeken a hagyományos rádióadásoknak komoly földrajzi akadályokat kell leküzdeniük.

Megtudhatnánk önről és a vállalatról néhány dolgot?

Legutóbb a washingtoni képviselőház kereskedelmi egyesületének jogi képviselője voltam. Ez az egyesület a távközlési hálózatok, az Internet, és az amerikai kereskedelem bírói szervezete. Itt ismertem meg számos olyan területet, amelyek végeredményben a PenguinRadio ötletének megszületéséhez vezettek.

Egy ideig próbálkoztam Interneten át sugárzott műsorokkal, de hamarosan beleütköztem abba a nehézségbe – amellyel már sokan találkoztak –, hogy a számítógépek a multimédiás anyagok lejátszása közben nem tudnak mással foglalkozni. Előbb-utóbb biztosan megakad a lejátszás. Sokáig bíztam benne, hogy ezt a gondot megoldhatjuk erősebb számítógépek beszerzésével, de a nehézségek a gépek fejlődésével sem szűntek meg. Ezért a másik irányba indultam el: tervezni kell egy kifejezetten médialejátszásra alkalmas készüléket. Rövid kutatás után rájöttem, hogy egy ilyen eszköz nagyon hasznos lenne saját háztartásomban is, függetlenül attól, hogy vevőkre talál-e a piacon. Tehát úgy egy éve kiléptem az egyesületből, felkerestem néhány befektetésben érdekelt személyt, és hamarosan megtaláltam az Internet Partnership Group nevű szervezetet.

Elérhető lesz-e a rendszer bármely része a nyílt forrás társadalma számára?

Még nem állapodtunk meg pontosan arról, hogy mely részek lesznek nyilvánosak, és melyek nem, de abban egyetértettünk, hogy bőkezűnek kell lennünk a Linux-közösséggel szemben, hiszen rengeteg segítséget kaptunk tőlük. Már ma is elérhető más honlapok számára az általunk összegyűjtött több ezer rádióállomás listája, és tervezzük, hogy más gyártók számára is nyilvánossá tesszük munkánk eredményét.

Milyen segítségnek örülnének a legjobban?

Szívesen fogadunk minden megjegyzést, tanácsot, bírálatot és magasztalást. Nagyra értékelünk mindent, amit valaki hasznosnak érez számunkra, vagy az olyan gondokat, amelyek az általunk felhasznált rendszerekben merültek fel.



Milyen Linux rendszermagot használnak?

Elsősorban a 2.2.x sorozattal kísérleteztünk. Mindenképpen az számít, hogy a rendszermag minél újabb legyen. A mérete nem lényeges, hiszen a memória ára lefelé megy.

Elérhető a Linux-változatok forráskódja?

Nem akarok kitérni a kérdés elől, de az internetes hangsgyűjtés olyan nagy lépésekkel fejlődik, ezért jelenleg nem érezzük úgy, hogy készen állnánk a forráskódok kiadására. Munkánk többnyire abból áll, hogy testre szabott alkalmazásokat készítünk, tehát az általunk használt változat nem sok különleges vagy szokatlan elemet tartalmaz.

Elérhető már a rádióknak valamilyen bemutatóváltozata, vagy még nem tart itt a fejlesztés?

Már készítettünk néhány mintadarabot, de a hivatalos bemutatóváltozat még nem jelent meg.

Kit tartanak a legkomolyabb vetélytársuknak?

A gép szempontjából a Kerbango jut elsőként eszembe, de a PenguinRadiónak ez csupán az egyik oldala. Az Ineva.com segítségével a mi internetes rádiónk bárhol használható, ahol a műholdjeleket lehet fogni. A PenguinRadio elméletének a vezeték nélküli távközlés is részét képezi, hiszen internetes adásainkat a szabadon elérhető phoneradio.com címen tesszük közzé.

Köszönjük, hogy időt szakítottál ránk!

☞ <http://www.penguinradio.com>



Jason Schumaker (jason@ssc.com) már két éve dolgozik a Linux Journalnak. Ő a lap egyik segédszerkesztője, rendszeres szerzője és amikor nem dolgozik, akkor a legkülönbözőbb sportokkal fásasztja magát.

Az új rádió

„Legyen személyre szabott. Ez minden vágyam a rádióval kapcsolatban.”

– Larry Josephson



A „Saturday Night Live” számomra egyik legkedvesebb pillanata az volt, amikor *Paul Shaffer Don Kirshnert*, a sisakhajú, örökbarnára sült lemezkiadó-elnököt parodizálta. 1978 áprilisában Shaffer (Kirshnert játszva) először mutatta be a Blues Brotherst: „Ők mostantól megszűntek eredeti bluescsapatként működni. A Blues Brothers ezennel üzleti termékékké vált.” Kirshner egész személye már akkor is időtlen volt. Addigra világossá vált, hogy az eredeti és az üzleti jelenségek közti különbség mindössze annyi, hogy a zeneipar a nyereségtől hajtva jól „megcsinálta” a maga csillagait. A fogyasztói zene és a műsorszórás ugyanazon gép két fogaskereke volt: a tömegek zenei étvágyának ingerlése és kielégítése, lehetőleg a legkevésbé változatos módon. A Saturday Night Live szintén e gépezet része volt. Mi más ment abban az időpontban a tévében? Vagy bármikor?

A rádió helyzete sem volt sokkal könnyebb ennél, hiszen még a legnagyobb városokban sem volt több néhány tucat – a városszéli autópálya-kijáráthoz érve hirtelen elhalkuló – állomásnál. Egy kézen össze lehetett számolni a zenei adókat. E néhány rádióállomás ugyanazt cselekedte: a közlés egysíkúvá formálásával máris jóval nagyobb hallgatóságról számolhatott be büszkén a hirdetőinek, s ennek anyagi vonzatait nyilván nem kell ecsetelnünk. Az eredmény az lett, hogy az igények ugyanannyira beszűkültek, mint a kínálat, s a legszűkebb ízlés kapta a legtöbb műsoridőt.

Szintén lekicsinylő az a hit, hogy a hallgatókat bármilyen szempont alapján „piaci kategóriákba” lehet sorolni. Az üzleti rádiózás gazdaságtudatói szerint nem a műsor a termék, hanem a hirdetőnek eladható hallgatóság. A rádióállomások és a hálózatok hirdetés idejét adtak el a hirdetőnek, és nem műsort a hallgatóknak. A zene és a többi program csak a csali szerepét töltötte be. A hallgatók igényei gyakorlatilag semmit sem jelentettek, hiszen ők csupán fogyasztók és nem vásárlók voltak.

Tehát hibáztathatjuk őket, ha egymás között ingyen adják tovább a zenét?

Minden bizonnyal. Legalábbis a zeneipar ezt teszi, amikor a Napstert támadja és fenyegetőzik a szerzői jogvédelem alatt álló zene „lopása”, tehát az MP3-ak internetes csereberéje ellen. *Eric Raymond* a művé-

szek és a lemezkiadók érdekeit képviseli, s álláspontja szerint csak nekik van joguk a zenét mint saját tulajdonukat terjeszteni. Nem veszi figyelembe azt, hogy milyen egyszerűen, feltűnés és bárki beleegyezése nélkül lehet zenét cserélni. A *Linux Journalban* megjelent vezércikkében Eric így vélekedik:

„A lényeg, hogy a szerző beleegyezése nélküli ingyenes továbbadás-sal megfosztjuk őt saját munkájának irányításától. Az igazi kérdés tehát az: támogatjuk-e a művészeket, vagy egyszerűen ellopjuk megmaradt önállóságuk utolsó morzsáit is?”

Tegyük fel, hogy a kérdés első felére igennel válaszoltunk. A következő kérdés másképpen feltéve így hangzik: lehetséges-e egy olyan piac létrehozása, ahol minden hozzáférhető és mindenki elveheti azt, ami neki jár?

Courtney Love válasza: „Keresem azokat az embereket, akik segítenek eljutnom a rajongókhoz, mert szerintem a rajongók az általam nyújtott szórakozástól függő értékű borraivalót hagynak az asztalon. Én nem félek nekik ingyen megmutatni néhány részletet. Azt hiszem, előbb-utóbb kialakul az a sokat emlegetett Világfalu, ahol egy művészt akár egymilliárd ember is elérhet, s ezek mindegyike adhat borraivalót.”

Tehát erőteljes egyenjogúsodásról van szó: minden művész elérheti minden rajongóját, és a rajongók az alkotót, illetve az őket a művészhez irányító embereket. Ehhez a tanácsot adó és a szakmai segítséget nyújtó emberekre van csupán szükség. A terjesztési „csőhálózatban” lebzselő s a rajongókat figyelmen kívül hagyó ingyenélők nem kellene. Ez volna a tökéletes terjesztői rendszer. Vajon létezik már ilyesmi, legalább kezdetleges formában?

Hogyan létezne! Úgy hívják: nyilvános műsorszórás. A szolgáltatást eladó intézmények üzletellenes természetén s a „támogatásról” szóló emelkedett elképzeléseiken túlesve már láthatjuk is a rendszert működés közben. Számos esetben jobban is működik, mint a hirdetési modell, azon egyszerű oknál fogva, hogy a fogyasztói és a vásárlói közönség egy és ugyanaz: ezt a kényelmet a hirdetésekkel támogatott műsorszórás soha nem érezheti. (Az előfizetőkre alapozó sajtó, mint ez az újság is, viszont igen, köszönjük!).

Lássunk egy hús-vér példát is! A San Francisco-i KJAZ néhány évvel ezelőtt még az Egyesült Államok legrégebben működő jazzrádiója címmel büszkélkedhetett, mikor tulajdonosa adósságba keveredett és meg kellett válnia a rádiótól (ami egyébként addig sem hozott sokat a konyhára, de ahhoz éppen eleget, hogy működni tudjon). Utolsó kísérletként a hallgatósághoz fordultak adományokért, és lass csodát: néhány hét alatt egymillió dollár gyűlt össze, ami több volt mint a rádió egész éves hirdetési bevétele. A fennmaradáshoz sajnos nem volt elég, így az adakozók visszakapták pénzüket, de e piaci modell működőképességét bizonyítja, hogy az egyik helyi nem kereskedelmi rádió, azt átvéve, nagyobb bevételre tett szert, mint mekkora a KJAZ-nek hirdetésekből valaha is sikerült.

Szívesen adok pénzt olyan műsorért, amit szeretek. Jelenleg négy rádió- és egy tévéállomásra fizetek elő. A befizetett összeget nem adománynak tekintem, sokkal inkább a fentebb említett „borraivalót”



Courtney Love

rendszer” megvalósulásának, bár nem bánom, hogy le lehet írni az adóból. Jó lenne, ha a modell köré komoly intézményhálózat épülne ki, és viszonylag kis pénzzel közvetlenül a művészeket támogathatnám.

Miért nem működhet ez minden tévéadó esetében így? Két ok a sok közül: 1. a szerkezet még nem fejlődött ki; 2. a műsorszolgáltató főmunkatársainak sehogyan sem fér a fejébe, hogy nemcsak a hirdetési pénzekből lehet nyereszkedni.

A leghatékonyabb reklám (ami az igényekre válaszol, vagy legalábbis igyekszik közel kerülni azokhoz) nem a márkát minősíti. Csak kétféle hirdetés célozza meg közvetlenül a közönség igényeit: az apróhirdetés és a sárga oldalak. Az ehhez hasonló újságokban látható hirdetések közelítenek hozzá, mivel ezek a szerkesztőség fizetéséhez



A Napsteren keresztül zajlik az internetes csere-bere

járulnak hozzá. Nélkülük a magazin kevesebbet érne. A tömegpiaci értékesítés azonban teljesen más elvek alapján működik.

A tökéletesen szabályos, kiegyensúlyozott hálózatos piacokon a tömegpiaci hirdetések túlságosan drágák és elszámolhatatlanok. Hiába próbálnánk pontosan kiszámíthatóvá tenni a dolgot, rövid időn belül valami teljesen más alakulna ki belőle.

Mi történne, ha a TiVo-felhasználók tényleg kapcsolatba kerülhetnének a tévék hirdetőivel? Vagy ha a némító gomb megnyomása közvetlenül a hirdetőkhöz juttathatná el a „nem ezt akarjuk” típusú üzenetet, és nem a műsorszórókhoz vagy az ügynökségekhez?

Vigyünk ezt tovább egy lépéssel! Mi lenne, ha mindenki elindíthatná saját rádió- vagy tévéállomását? Nos, pontosan ez történt a Napsterrel és a Gnutellával. A közvetlen MP3 csere lehetőségét kihasználók száma ugrásszerűen nő. Ebben mindössze azt kell látnunk, hogy a zenekedvelő emberek ismét felfedezik maguknak azt az utat, amit a rádiózás hőskorában az AM és FM adók követtek azzal, hogy hatalmas lemezyűjteményekből játszottak kiváló zenéket, ezek egyben kitűnő vásárlási ötleteket adtak a hallgatóknak. Ezt szüntették meg fokozatosan a kereskedelmi rádiók.

A zene továbbadása ugyanazt teszi a rádiózással, mint a PC megjelenése a számítástechnikával: közel viszi mindenkire, s ezzel együtt az egészet a közönség igénye, nem pedig a kínálat kezdi el irányítani. A néhány ezer, hirdetésekből élő, a közönség elvárásaival nem törődő rádió helyett itt van milliányi, pusztán az igények által működtetett állomás. A Radio Userland és a hozzá hasonló új szolgáltatások tisztában vannak azzal, hogy a rádióknak teljesen személyesnek kell lennie, s hogy az igény és a kínálat egyazon oldalon található.

A személyes rádiózás a zenék cseréjéről szól, nem pedig a hifiről. Courtney szerint „senki nem választana egy recsegős, Napsterről letöltött MP3-at az eredeti felvétel helyett”. Valójában a rádió mindig

is a gyengébb minőségnek felelt meg. Az amerikai country színtér zászlóshajója a Nashville-ből sugárzott WSM volt, melynek zúgó, torz hangú adása az ország délkeleti részén is fogható volt. Még az FM rádió sem éri el a CD-minőséget, hiszen a minta „összepréselése” miatt a magas hangokat, 15 kHz fölött levágja. Szóval hogyan is valószínűsíthetnénk meg azt a borraló-rendszert?

Először is meg kell szabadulnunk a Napster-egyezménytől. Ezt ma már sokan megteszik azok közül, akik a zene forgalmát nem a kiadók által irányított kereskedelmi tevékenységnek tekintik. Véleményem szerint meg lehet találni az arany középutat. *Don Marti* szokatlanul viccelni, hogy „az értesülés 6,95 dollárba akar kerülni”. Van benne igazság.

A viták ráébresztenek bennünket a továbbadott zene igazi értékére. Természetesen a viták alatt nem azt értem, hogy a Gonosz Kiadóknak pusztulniuk kell, meg hogy a Zene Szabad Akar Lenni, meg hasonlókat. Eric Raymondnak igaza van: ezek csak elterelik a figyelmet. Nyilvánvalóan lehetséges elegendő áru (a szabad áruk) értékét valahol a nulla fölött meghatározni.

Az árcédula csak százéves. A történelem folyamán a termékek értéke mindig is megbeszélés tárgyát képezte. Látogassunk csak el a harmadik világ bármelyik piacára, és a dolog nyilvánvalóvá válik. *Fr. Seán Oloaire*-t kiutasították Kenyából, mert tizenöt év ott letöltött szolgálat után élelmet akart bevinni a szegényeknek. Szerinte „az árcédula a falusi piacokon sem az eladó, sem a vevő számára nem jelent semmit”. Ha az eladó rendelkezne ezzel a hatalommal, s többé-kevésbé gyakorolná is, annak semmi értelme nem lenne. „Olyan lenne, mintha saját magunkkal beszélgetnénk” – állítja Oloaire.

Új földekre érkezünk, ahol az ipar elgondolásai a tulajdonról és annak terjesztéséről alkalmazhatatlannak és idejétmúltak bizonyulnak. A hagyományos piacok mintájára most a megegyezés keresésének kell átvennie a főszerepet.

- ➔ <http://radio.userland.com/>
- ➔ <http://www.napster.com/>
- ➔ <http://www.mp3.com/>

Hazai állomások a Neten

Kossuth rádió

- ➔ http://www.radio.hu/ram/kossuth_am.ram AM adás
- ➔ http://www.radio.hu/ram/kossuth_fm.ram FM adás

Petőfi rádió

- ➔ http://www.radio.hu/ram/petofi_fm.ram

Bartók rádió

- ➔ http://www.radio.hu/ram/bartok_fm.ram

Calypso rádió AM

- ➔ http://www.radio.hu/ram/calypso_am.ram

Pararádió

- ➔ www.pararadio.hu

Tilos Rádió

- ➔ www.tilos.hu



Doc Searls (doc@ssc.com)
a Linux Journal főszerkesztője és
a Cluetrain Manifesto társszerzője.

A DNS és a BIND biztonsága

Ha a DNS-kiszolgáló biztonságos, a hálózat is biztonságos és naprakész.

ASANS Institute nemrég kiadott, a tíz legfontosabb internetes biztonsági veszélyről szóló írásában első helyen foglalkozik a BIND sebezhetősége miatt előforduló hibákkal. A BIND nevű nyílt forrású programcsomag képezi a Világháló legtöbb DNS-kiszolgálójának lelkét. A SANS szerint a telepített BIND csomagok legalább fele a legismertebb támadási módszereknek sem képes ellenállni. A jó hír azonban az, hogy az itt ismertetésre kerülő, könnyen megérthető szabályok betartásával nagymértékben növelhetjük a BIND megbízhatóságát a linuxos (vagy más unixos) DNS-kiszolgálónkon. Mivel itt és most főleg a biztonsággal foglalkozunk, ajánljuk, hogy a BIND kezelésében kevésbé jártasak előbb olvassák el a csomag Interneten elérhető leírását, illetve az első két fejezetet az *Albitz-Liu* szerzőpáros *DNS and BIND* című híres könyvéből.

A BIND alapjai

Kezdjük a Domain Name Service (DNS) és a BIND működésének ismertetésével. Tegyük fel, hogy gépünk címe `myhost.someisp.com` (1. ábra) és a `http://www.wiremonkeys.org/` címen található honlapot szeretnénk megtekinteni. A DNS-kiszolgáló neve legyen `ns.someisp.com`. Mivel a `www.wiremonkeys.org` semmit nem mond az adattovábbításért felelős átjárók (gateway) számára, a felhasználó böngészőjének előbb meg kell tudnia a névhez tartozó IP-címet. Először a `myhost` az `ns` géphez fordul, hátha az tud valamit a címről. Mivel az `ns.someisp.com` nem felelős a `www.wiremonkeys.org` nyilvántartásáért, s az utóbbi időben nem érkezett hozzá a cím kiderítésére irányuló kérelem, ezért a felhasználó nevében kezdi meg a keresést. Az egyik kérelem kiszolgálása céljából intézett másik kérelmet rekurzióknak hívjuk.

Az `ns.someisp.com` egy főkiszolgálótól (root server) tudja meg a `www.wiremonkeys.org` címet nyilvántartó DNS-kiszolgáló nevét. (Minden DNS-kiszolgálón találunk egy szövegfájlt, amelyben az Internet tizenhárom főkiszolgálójának neve és címe található. Ez a lista az `ftp://ftp.rs.internic.net/domain` címen érhető el, neve „`named.root`”). Példánkban az `ns` az `E.ROOT-SERVERS.NET` főkiszolgálót kérdezi meg (címe `192.203.230.10`), mely azt válaszolja, hogy a `www.wiremonkeys.org` címet az `ns-wiremonkeys.wiremonkeys.org` kiszolgáló tartja nyilván, s ennek címe `55.100.55.100`.

Az `ns` ezután az `ns-wiremonkeys` géptől érdeklődik a `www.wiremonkeys.org` IP-címe iránt. Az `ns-wiremonkeys.org` az `55.100.55.200` választ küldi vissza, s ezt az adatot az `ns` a `myhost.someisp.com` címre, azaz saját gépünkre továbbítja. Végül a `myhost` kapcsolatba lép az `55.100.55.200` címmel HTTP-n keresztül, és így már elkezdődhet az adatcsere. Ez a névfeloldás legjellemzőbb példája, amit egyszerűen „lekérdezésnek” nevezünk. A lekérdezések számára az 53-as UDP kapu van fenntartva. Azonban nem minden DNS műveletben egy gazdát kérdezzünk le. Néha szükséges, hogy egész névtartomány- (vagy

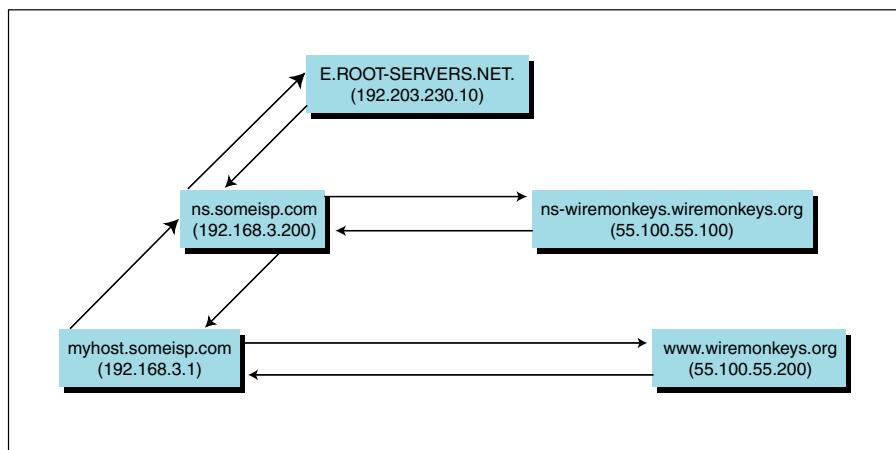
zóna-) adatbázisokat továbbítsunk. Ezt tartománytovábbításnak (zone transfer) hívjuk. Amikor a `dig` parancsot futtatjuk, vagy az `nslookup` program `ls` parancsát használjuk, akkor is ez a folyamat zajlik le. A tartománytovábbítás fő célja az, hogy az egyazon tartományért felelős névkiszolgálók összehangoltan működjenek. A tartománytovábbítás szintén az 53-as UDP kapun keresztül történik. Az utolsó általános DNS tulajdonság a gyorsítár (cache) alkalmazása. A névkiszolgálók a helyi tartomány minden fájlját gyorsítárba helyezik, s ugyanígy a legutóbbi újraindítás óta hozzájuk intézett lekérdezési kérelmeket is. Tulajdonképpen ennyi lenne az egész: minden forrásrekordnak (resource record, RRs) élettartam-beállítása is van, ezek az adatok értékek azt határozzák meg, hogy az adott RR meddig tartózkodhat a gyorsítárban, azaz mikor kell legközelebb frissíteni. Mindez természetesen csupán a töredéke annak, amit érdemes tudni a BIND-ről, hiszen a továbbítókat (forwarder) és a fordított lekérdezéseket nem is említettem. Remélhetőleg, ennyi is elég lesz a BIND biztonságával foglalkozó részek megértéséhez.

A DNS biztonsági elvei

A biztonságos DNS két alapvető szabálya a következő: mindig a választott DNS-programcsomag legfrissebb változatát használjuk, illetve soha ne engedélyezzük feleslegesen sok adat vagy szolgáltatás elérését idegenek számára. Röviden: legyünk naprakészek, gyanakvóak és zsigoriak!

A fő elvekből több eljárás következik, amelyeket alkalmaznunk kell. Az első, hogy korlátoznunk, esetleg tiltanunk kell a rekurziót. A korlátozás egyszerűen elvégezhető a beállításfájl értékeinek megváltoztatásával; megtiltani viszont csak abban az esetben tudjuk, ha ez nem akadályozza a kiszolgáló munkáját.

Ha egy gép külső DNS-kiszolgálóként működik, melynek egyetlen feladata a saját hálózatához tartozó gépek felé irányuló kérelmek teljesítése, akkor nyugodtan megtilthatjuk a rekurziót, hiszen a gépnek így nem kell külső gazdaneveket keresnie. Másrésztől, ha egy kiszolgáló DNS-névfeloldást ad a helyi hálózat gépei számára, akkor a helyi hálózathoz érkező rekurziós kérelmeket mindenképpen teljesítenie kell, a kívülről érkezőket azonban nyugodtan figyelmen kívül hagyhatja.



1. ábra A `myhost.someisp.com` lekérdezése

A DNS korlátozásának másik módja, hogy a szolgáltatásokat szétválasszuk egymástól (2. ábra).

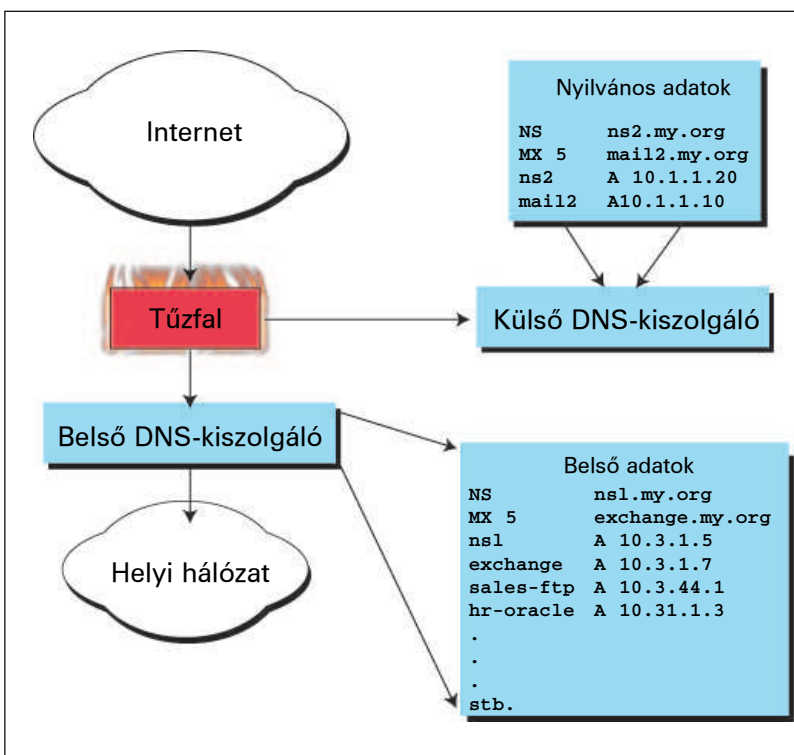
A „szétválasztott DNS” kifejezés azt jelenti, hogy minden egyes helyi résztartományhoz nyilvános és titkos adatbázist egyaránt fenntartunk. A nyilvános adatbázisban csak a legszükségesebb adatok találhatóak meg: a nyilvános névkiszolgálók nevét tartalmazó NS bejegyzések; a külső SMTP (e-mail) átjárókat meghatározó MX bejegyzések; a nyilvános webkiszolgálók és más gépek nevei, amelyek létezéséről a külvilág is tudomást szerezhet.

A titkos adatbázis a nyilvános adatbázis bővített változata is lehet, de akár teljesen különbözhet is attól. Például sok helyen a Microsoft Exchange kiszolgálót használják belső levelezésre, de a külső levelezést egy másik SMTP átjárórendszerrel bonyolítják le, ez általában a hálózat tűzfala vagy egy, a tűzfallal kapcsolatban álló, de a belső hálózatról leválasztott szabad sávban (Demilitarized Zone) lévő levelező-kiszolgáló. Az ilyen felépítés előnye egyértelmű: az SMTP átjáró sebezhetősége nem jelenti azt, hogy egy behatoló az egész belső levelezést elérheti. Más, ehhez hasonló módon szétválasztott szolgáltatások között az alábbiakat találjuk: a WWW (a kívülről elérhető adatokat, illetve a belső hálózat adatait választják el), az FTP és gyakorlatilag minden olyan TCP/IP-alapú szolgáltatás, amelynél az adatokat „ez nyilvános”, „ez meg nem” elvek alapján lehet csoportosítani. A DNS szétválasztása talán a legkényesebb téma, hiszen a legtöbb TCP/IP-szolgáltatás alapját a DNS képezi. A fentebb említett „gyanakvás” másik területe a tartományfájlok tartalma. Még a nyilvános adatbázis is feleslegesen sok adatot tartalmazhat. A gazdanevek túlságosan beszédesek lehetnek (ez megkönnyíti a behatolók dolgát), illetve túl sok vagy fontos adatot szolgáltathatnak. Néhány hálózat még az egyes rendszerek egyedi tulajdonságait is közzéteszi! Az ilyen jellegű adat a legtöbb esetben csupán a behatolók számára jelent valamit. A programok állandó frissítése és az ismert DNS-hibák kiszűrése legalább olyan fontos, mint annak eldöntése, hogy a DNS-kiszolgáló milyen kérélmeket teljesítsen. Ráadásul még egyszerűbb is, hiszen a BIND legújabb változatát ingyen letölthetjük az ftp.isc.org címről, és a BIND hiányosságairól szóló beszámolókat, vitákat több levelezési listán és hírcsoportban is figyelemmel kísérhetjük.

A DNS biztonságának harmadik alapvető elve nem csak a DNS-re vonatkozik: szánjunk időt a programcsomag biztonsági szolgáltatásainak átböngészésére és megfelelő beállítására! Szolgáltatónkra is figyeljünk oda: a DNS általános vezérléséért felelős Network Solutions és más hasonló cégektől gyakran levélben is kérhetünk értesítést a legújabb biztonsági frissítésekről.

A megfelelő BIND csomag kiválasztása és telepítése

E cikk megírása idején a legfrissebb változat a 8.2.2-es volt, az 5. hibajavítással (patch). A régebbi változatok egyik különösen alattomos hibája, hogy a behatoló a túlszordulás kihasználásával rendszergazdai jogokat szerezhet (erre a CERT #CA-99-14 számú jelentése is kitér), ezért különösen fontos, hogy a 8.2.2p5 változatot használjuk. Megjegyzendő, hogy a BIND v.8.1 1997 májusi megjelenése után, megbízhatóságának köszönhetően sokan továbbra is a v.4-et használták (s így az új beállításokat sem kellett megtanulniuk...). Az Internet Software Consortium (ISC) továbbra is támogatja ezt a változatot; a v.4.9.7 a 8.1 kiadása után egy évvel jelent meg. Az ISC sem javasolja azonban, hogy bárki ezt használja. Még egyszer ismétlem: ha BIND, akkor legalább 8.2.2p5.



2. ábra Szétválasztott DNS

© Kiskapu Kft. Minden jog fenntartva

A következő kérdés, hogy előfordított bináris csomagot (például RPM-et) használjunk-e, vagy magunk fordítsuk le a forráskódot? A legtöbb felhasználó számára tökéletesen megfelel a lefordított csomag, feltéve, hogy megbízható helyről szerezte be. Gyakorlatilag a BIND minden Unix-változat része, de ellenőrizzük, hogy nekünk a legújabb van-e meg.

A Red Hat Package Managerben az `rpm -q -v bind8` parancsot kell használnunk, ha a csomagot már telepítettük, vagy az `rpm -q -v -p ./<a_fájl_neve>` parancsot, ha van, de még nem telepítettük. Az rpm-es BIND csomagneve általában „bind8” vagy „bind”.

Ha kiderül, hogy régi változatunk van, akkor sincs nagy baj: a legtöbb csomagformátum frissítési szolgáltatást is tartalmaz. A legújabb változat letöltése után a csomagkezelővel frissíthetjük a régit. Az RPM esetében az `rpm -U ./<a_csomag_neve>` parancsot kell kiadnunk, ehhez természetesen még további kapcsolókat is fűzhetünk. Ha ez nem működne, akkor próbáljuk meg az `rpm -U --force ./<a_csomag_neve>` alakot.

Ha nem találunk megfelelő csomagot, akkor magunknak kell lefordítanunk a forrást. Ez sem nehéz, hiszen nincs beállítóprogram, és a BIND v.8x Makefile-jait sem kell módosítanunk. Csak kövessük a forrás INSTALL fájljában megadott útmutatást. A legtöbb esetben mindössze egy `make`, majd egy `make install` parancsra lesz szükségünk.

A named indítása: a lelakatolt cella

A BIND fő folyamatát, a namedet egyelőre még korai lenne elindítanunk. Arról azonban döntenünk kell, hogy mire akarjuk használni, hiszen ettől függ az összes beállítás. Itt az idő tehát, hogy szót ejtsünk néhány, a biztonságot növelő kezdeti beállításról.

Ahogy minden internetes szolgáltatást, a namedet is célszerű „lelakatolt cellában” futtatnunk, amelybe a behatoló szépen bezárja magát, ha a túlszordulás kihasználásával támad kedve kísérletezni. Ezt a cellát három kapcsolóval hozhatjuk létre: `-u <felhasználónév>`, `-g <csoportnév>` és `-t <könyvtárnév>`.

Az első hatására a named az adott felhasználói néven, a második hatására pedig a megadott csoportnéven fut. A harmadik a named által hivatkozott könyvtárak elérési útvonalainak gyökerét változtatja meg (chroot). Figyeljünk oda arra, hogy ez utóbbi beállítás még a named.conf beolvasása előtt érvénybe lép, ezért a

```
named -u named -g wheel -t /var/named
```

parancs kiadásakor a named.conf fájlt a rendszer nem a /etc könyvtárban, hanem a /var/named/etc-ben keresi. Tehát a named.conf alapértelmezés szerinti helye mindig a /etc, de ha a könyvtárak gyökerét /más/útvonal-ra helyeztük, akkor a named.conf-nak a /más/útvonal/etc-ben kell lennie.

Hogy miként növeli ez a biztonságot? A három kapcsoló használatával a named jogait, környezetét, sőt, még az elérhető fájlrendszereket is komolyan korlátozhatjuk. Ha egy behatoló netán átvenné a vezérlést a BIND fölött, nem kapna rendszergazdai jogokat (a BIND v.8 előtt ezt megtehetette, hiszen a BIND rendszergazdaként futott). Ehelyett egy közönséges felhasználó jogait kapja meg, ezenkívül (mivel a megadott gyökér fölött elhelyezkedő könyvtárak tartalma a named számára gyakorlatilag nem is létezik) szinte semmit sem láthat a fájlrendszerekből.

A named.conf

A fent ismertetett módszer már tökéletesen biztonságos, de ez még csak a kezdet! A BIND 8.x beállításfájlja (named.conf) rengeteg paraméterével a szolgáltatást egészen pontosan szabályozhatjuk. Vegyük az 1. listán látható példát. Az ehhez tartozó gép egy külső DNS-kiszolgáló. Mivel feladata szerint a helyi tartományról (coolfroods.org) szolgáltat adatokat a külvilág számára, a rekurziót nem engedélyezi. Valójában nincs is "." bejegyzése (ami egy listafájltra mutatna), tehát nem tud és nem is szerezhet adatot a helyi tartományon kívül található gépekről.

A helyi tartomány adatbázisait csak a megbízható másodlagos kiszolgálók egy csoportja számára adja ki, s majd minden előforduló eseményről naplóbejegyzés készül.

Tehát mi kerül a named.conf-ba, amittől nyugodtabban alhatnak a legfőbb rendszergazdák is?

Hasznos named.conf beállítások: acl{}

Az ACL (Access Control List) listák segítségével IP-címek egy csoportjához neveket rendelhetünk. Erre szükség is van, hiszen nyilvánvalóan tiltani szeretnénk a hozzáférést bizonyos IP-címekről.

Egy ACL-t elvileg a named.conf bármely részén meghatározhatunk, de mivel a fájlt a rendszer felülről lefelé olvassa be, ezért első használata előtt minden ACL-t meg kell határozni. Ebből következik, hogy az ACL-meghatározásokat a named.conf tetejére kell tennünk. A beállítás formája egyszerű:

```
acl név { IPcím1; IPcím2; ...};
```

Figyeljük meg, hogy minden IP-címlista teljes (x.x.x.x alakú) és hálózati (x.x.x.24, x.x./16 stb. alakú) címetek is tartalmazhat. A fájl beolvasásakor az ACL neve helyére az általa meghatározott címlista kerül.

1. lista Példa a named.conf-ra egy külső DNS-kiszolgálón.

```
acl trustedslaves { 192.168.20.202; 192.168.10.30};

options {
    directory "/";
    listen-on { 168.192.100.254; };
    recursion no; fetch-glue no;
    allow-transfer { trustedslaves; };
};

logging {
    channel seclog {
        file "var/log/sec.log" versions 5 size 1m;
        print-time yes; print-category yes;
    };
    category xfer-out { seclog; };
    category panic { seclog; };
    category security { seclog; };
    category insist { seclog; };
    category response-checks { seclog; };
};

zone "coolfroods.ORG" {
    type master;
    file "master/coolfroods.hosts";
};

zone "0.0.127.in-addr.arpa" {
    type master;
    file "master/0.0.27.rev";
};

zone "100.168.192.in-addr.arpa" {
    type master;
    file "master/48.42.208.rev";
};
```

Általános beállítások: options{}

Nézzük át az általános beállításokat. Néhány ezek közül a tartományokra vonatkozó szakaszokban is használható. Figyeljünk arra, hogy, ha egy beállítás az options{} részben, és a tartományok szakaszában is szerepel, akkor az utóbbi változat felülbírálja az előbit. Tehát a leíró részben megadott beállítások az általános beállítások kivételeit határozzák meg.

A 2. listán az options{} részben használható néhány hasznos értéket találunk.

Naplózás

Az általános beállítások mellett a naplózási szabályokat is meg kell határozni. Alapértelmezés szerint a named csupán néhány indítási üzenetet (a hibákat és a betöltött tartományokat) naplóz a syslogd démon segítségével, az pedig a /var/log/messages-be vagy máshová írja azokat. A biztonsággal kapcsolatos események naplózásához a logging{} szakaszt is be kell illesztenünk a named.conf fájlba. A logging{} két részből áll: egy vagy több channel{} (ezek a csatornák határozzák meg, hogy hová kerüljenek a naplóadatok), illetve egy vagy több category{} (itt a nyomon követett eseményekhez rendelhetünk csatornákat) szakaszt tartalmaz. A csatornák általában fájlokra vagy a helyi syslogd démonra mutatnak. A tárgy-

körök legtöbbször előre meghatározottak, tehát ezek közül választhatunk, és megadhatjuk, hogy az adott tárgykörre vonatkozó napló-adatok hová kerüljenek.

A csatornákat az alábbiak szerint határozhatjuk meg:

```
channel csatornanév {
    [ fájlnev | syslog syslogtípus | null ];
    print_time [ yes | no ];
    print_category [ yes | no ];
};
```

Ne felejtjük el, hogy a fájlnev alapértelmezés szerint a named munkakönyvtárába kerül, de teljes útvonalat is megadhatunk, mely (ha lehetséges) a módosított gyökérhez képest kerül értelmezésre. A tárgykörök meghatározása jóval egyszerűbb:

```
category kategória_neve {
    csatornalista ;};
```

Az IP-címlistákhoz hasonlóan a csatornalista elemeit is pontosvesszővel választjuk el egymástól, s az csak egy fentebb szereplő channel{} beállításban meghatározott csatornaneveket tartalmazhat. A BIND útmutatójában (BOG) a támogatott tárgykörök teljes listáját megtaláljuk, itt és most elég annyit megemlítenünk, hogy az xfer-out, security, load, os, insist, panic és maintenance minden biztonságmániás rendszergazdát érdekelhetnek.

Csak gyorstárazó névkiszolgálók

A csak gyorstárazó (caching-only) névkiszolgálók nem felelnek a tartományok nyilvántartásáért, így biztosításuk is jóval egyszerűbb, s a velük végzett munka során az alábbiak közül csupán néhány dologra lesz szükségünk.

A named.conf szakaszai közül utolsóként a zone{} kerül terítékre. Az options{} részhez hasonlóan a lentebb ismertetetteken kívül rengeteg további beállítás használható. A BOG-ból mindent megtudhatunk róluk.

A tartományonkénti biztonsági beállítások közül az alábbi három a legfontosabb:

```
allow-update { IP_vagy_ACL_lista ; };
allow-query { IP_vagy_ACL_lista ; };
allow-transfer { IP_vagy_ACL_lista ;};
```

Az allow-update beállításnál a tartományra vonatkozó dinamikus DNS-frissítéseket küldő gazdákat sorolhatjuk fel. Az allow-query határozza meg, hogy mely gépektől fogadhatjuk el a DNS-kérelmeket. Az allow-transfernél állíthatjuk be, hogy kik tölthetnek le egész fájlokat. Jegyezzük meg, hogy mindhárom paramétert használhatjuk bármelyik, vagy akár mindkét zone{} szakaszban, illetve az options{} szakaszban.

A fájlok biztonsága

DNS-szolgáltatásunk most már jó úton halad a tökéletes biztonság felé. De mi legyen az adatbázisokkal?

A jó hír az, hogy mivel jóval kevesebb lehetőségünk van, mint a named.conf esetében, ezért kevesebb feladatunk lesz. A rossz hír viszont az, hogy legalább egy RR idejétmúlt és használata veszélyes, így legjobb messze elkerülni. A 3. listán a boneheads.com fájlját láthatjuk.

Az első érdekes elem a Start-of-Authority (SOA) bejegyzés. A fenti példában a sorszám az ééééhhnn## formában van

megadva. Ez így kényelmesen használható és a biztonságot is növeli, továbbá csökkenti annak az esélyét, hogy véletlenül régi adatfájlt töltsünk be. A frissítési időközöt három órára állították, ami megfelelő középút a sávszélességgel való takarékoskodás és az üldözési mánia között. Minél gyakrabban frissítünk, annál kevesebb kárt okozhat egy esetleges „cache-poisoning” (gyorstárfertőzés) típusú támadás, hiszen a behatoló által szaporított rossz bejegyzések az adatok minden frissítésekor javításra kerülnek. A lejáratí idő két hét, ez az időtartam az, amíg a fájl érvényesnek tekinthető. Egy biztonságmániás rendszergazda kétféleképpen tekinthet erre az értékre. Egyrészt a hosszú időtartam megengedi, hogy ha az elsődleges kiszolgálót egy bizonyos időszakban sorozatosan érnék szolgáltatásmegtagadás (denial-of-service) típusú DOS támadások, a másodlagos kiszolgálók a gyorstárban elhelyezett adatokkal

2. lista {} Options {}

```
listen-on [port #] { a helyi IP-k listája ; };
# Meghatározza, hogy a DNS-lekérdezésekre és a
# kérelmekre mely csatorlókon keresztül válaszoljunk.
# Megadása nem kötelező. Ennek és minden más
# címlistának az elemeit pontosvesszővel kell
# elválasztanunk egymástól.

# allow-recursion { a rekurzió, mely
# IP-címek felé legyen
# engedélyezett ; };
# Rekurzív lekérdezéseket végez a meghatározott
# IP-listán,
# Ez a "none;" szót is tartalmazhatja.

allow-transfer { azon IP-k, amelyek számára
# engedélyezzük az adatok fogadását
# vagy "none" ; };

allow-update { IP- vagy ACL-lista ; };
# Ezen IP-k, ACL-ek és hálózatok felől
# engedélyezett a dinamikus DNS frissítés
# (vagy "none")

allow-query IP- vagy ACL-lista ; };
# Ezen helyek számára engedélyezett az
# egyszeres DNS-lekérdezés (vagy "none")

version " [a változatszám mellett megjelenő üzenet]";
# Ezt senkinek nem kell kiszolgáltatnunk.
# A legtöbben ide humoros üzenetet írnak.

recursion [yes | no];
# Az általános rekurziót kapcsolhatjuk ki-be.
# Ha kikapcsoljuk, a fetch-glue paramétert is
# "no"-ra kell állítanunk (lásd lejjebb).

fetch-glue [yes | no];
# A "glue" bejegyzések olyan RR-ek, amelyek más RR-ek
# értelmezéséhez szükségesek (például minden olyan
# névnek, melyre egy "CNAME" bejegyzés hivatkozik,
# lennie kell valahol egy "A" bejegyzésnek is.
# Alapesetben a glue bejegyzéseket a hagyományos
# lekérdezések során is átvihetjük, hacsak itt ki
# nem kapcsoljuk e lehetőséget.
```

továbbra is elérhetővé teszik a tartományt, persze a fő DNS-kiszolgáló kivételével. Azonban az adatok a támadás idején is változhatnak, s a régi adatok néha több bajt okoznak, mint ha nem is léteznének. Tehát az élettartamot elég rövidre kell állítanunk ahhoz, hogy az esetleges támadást követő helyreállítás gyorsan megtörténjen, viszont elég hosszúra ahhoz, hogy a sávszélességet ne pazaroljuk feleslegesen. (A TTL határozza meg, hogy az egyes tartományrészek RR-jei meddig maradhatnak más névkiszolgálók gyorsítótárában, lekérdezésük után.)

Másrészt a biztonság fontos tényezője még az is, hogy minél kevesebb felesleges adatot szolgáltatassunk ki. A lehető legkevesebb nevet („A record”) és másodnevét („CNAME record”) tartunk nyilván, csak azok a gépek legyenek jelen, amelyekre feltétlenül szükség van. Valójában a DNS-t szeretnénk szétválasztani, de ha ez valamilyen okból nem kivitelezhető, akkor minél „szűkszavúbb” adatfájlokat kell készítenünk.

Ez akkor következik be, ha egy kért bejegyzés olyan nevet tartalmaz, melynek IP címe („A” típusú bejegyzésben) nincs jelen a kiszolgálón. Más szóval, ha az X kiszolgáló tudja azt, hogy Y felelős a WUZZA.com tartományért, de nem ismeri Y IP-címét, akkor kezd csak igazán bonyolódni a dolog a rendszer a lehető legjobb úton halad afelé, hogy valaki támadással törjön be. Ezért, ha minden rekurziót ki szeretnénk szűrni, győződjünk meg arról, hogy egyik RR sem igényel többszintű feloldást (glue-fetching), s ezután állítsuk a „fetch-glue” értéket „no”-ra.

Az RP és TXT típusokat ne használjuk, vagy csak körültekintően. A HINFO típusok azonban soha (!) nem tartalmazhatnak fontos adatokat! Az RP (Responsible Person) alapértelmezés szerint a kiszolgálóért felelős személy levélcímét adja meg. Ide legjobb valami teljesen semmitmondó címet írni, például: information@wuzza.com, vagy hostmaster@wuzza.com. A TXT bejegyzés további szöveges adatot tartalmazhat a kapcsolattartó személyről (telefonszám stb.), de lehetőleg tényleg csak ezt írjuk ide. A legjobb, ha az egészet figyelmen kívül hagyjuk.

A HINFO a régi szép idők hagyatéka: egyesek itt annak idején a használt operációs rendszert, annak változatát, sőt, még a kiszolgálóhoz tartozó számítógépek tulajdonságait is megadták! Akkoriban az Internethez kapcsolódó hálózatok legnagyobb része egyetemi rendszer volt, a számítógépekre még csodálkozva tekintettek (kevesebb bajkeverő ügködött...), és semmi sem indokolta, hogy ezt az adatot eltávolítsák. A HINFO-nak manapság már nincs valódi használata, hacsak az nem, hogy hamis adatok megadásával félrevezethetjük a betörőket.

A 3. listára visszatérve: láthatjuk, hogy a három utolsó bejegyzés teljesen felesleges, a behatolók számára viszont valóságos aranybányát jelent. S bár úgy fogalmaztunk, hogy a SOA bejegyzésekkel minden rendben van, az utána következő NS bejegyzéssel együtt egy másik tartomány gazdájára mutat. Mivel az ilyesmit nem szeretjük, ezért az ns.otherdomain.com-hoz egy „A” bejegyzést is be kell illesztenünk.

A BIND további biztonsági kérdései: a TSIG

A dolog legnehezebb részén túl vagyunk, de még nem is érintettük a titkosítás vezérlését. A „Secure DNS” (biztonságos DNS) protokoll (az RFC 2535-ben leírt DNSSEC) ismertetésének akár külön cikket is szánhatnánk. A DNS e bővítésének segítségével titkosíthatjuk a tartományrészek között továbbított adatokat, beleértve a szükséges kulcsadatokat forgalmát is. Mivel a DNSSEC még nem terjedt el széles körben (a BIND v.8x még nem is támogatja teljesen), így most csak a TSIG-ek (Transaction Signature, csomagalírási) használatával foglalkozunk.

3. lista Példa egy adatfájlr.

```
@ IN SOA cootie.boneheads.com. hostmaster.boneheads.com. (
    200000215 ; sorszám
    10800 ; frissítés (3 óra)
    1800 ; újrapróbálkozás (30 perc)
    1209600 ; lejárat (2 hét)
    432000 ) ; RR TTL (12 óra)
IN NS ns.otherdomain.com.
IN NS cootie.boneheads.com.
IN MX 5 cootie.boneheads.com.
blorp IN A 10.13.13.4
cootie IN A 10.13.13.252
cootie IN HINFO MS Windows NT 3.51, SP1
@ IN RP john.smith.boneheads.com.
dumb.boneheads.com.
dumb IN TXT "John Smith, 612/231-0000"
```

Tegyük fel, hogy a tartomány fő- és másodlagos kiszolgálója között továbbított adatait szeretnénk titkosítani. Ehhez a következőket kell tennünk:

1. kulcsot kell készítenünk a tartományhoz;
2. minden egyes kiszolgáló named.conf-jában létrehozunk egy, a kulcsot tartalmazó key{} bejegyzést;
3. minden kiszolgáló named.conf-jába beillesztünk egy server{} bejegyzést, melyben a 2. pontban létrehozott kulcsra hivatkozó kiszolgáló neve található.

Az első lépést a BIND dnskeygen programjával végezhetjük el a leggyorsabban. Egy 512 bites, a fő- és a másodlagos kiszolgáló által egyaránt használható kulcsot a

```
dnskeygen -H 512 -h -n <kulcsnev>
parancsral hozhatunk létre. A kimenet a
K<kulcsnev>.+157+00000.key, illetve a
K<kulcsnev>.+157+00000.private nevű szövegfájlb kerül.
```

Itt a kulcs mindkét fájlban azonos lesz, s körülbelül így fog kinézni: "ff2342AGFASsdfs55BSopiue/-2342LKJDIjkVVVvfjweovzp2OIPOTXUEdss2jsdfAAalskj==".

A másik két lépéshez a named.conf fájlokat kell szerkesztenünk mindkét kiszolgálón (a kulcsnévnek mindkét gépen azonosnak kell lennie!):

```
key kulcsnev {
    algorithm hmac-md5;
    secret "<ide kell beilleszteni a kulcsot>";
}

server <a másik gép IP-címe> {
    transfer-format many-answers;
    # (a válaszokat kötegelve és nem egyenként
    # küldi)
    keys { kulcsnév; };
};
```

Figyeljünk meg, hogy a key{} parancsnak meg kell előznie a rá hivatkozó többi parancsot (például a server{}-t). A kulcs és a kiszolgáló megadásának legjobb helye az option{} és az adatok között van. Most már csak újra kell indítanunk a namedet (a kill -HUP vagy az ndc restart parancsral) mindkét kiszolgálón. Ezek után elmondhatjuk, hogy DNS-kiszolgálónk szinte tökéletes biztonságban van!

Kapcsolódó címek

A SANS Institute által kiadott tíz legnagyobb internetes biztonsági rész:

☞ <http://www.sans.org/topten.htm>

Információk a DNS biztonságáról

BIND és a DHCPD honlapja:

☞ <http://www.isc.org/>

Cricket Liu, a DNS Security Slides szerzőjének és a DNS and BIND társszerzőjének oldaláról letölthető PDF-állomány címe:

☞ <http://www.acmebw.com/papers/securing.pdf>

A comp.protocols.tcp-ip.domains hírcsoport leggyakoribb kérdései:

☞ <http://www.intac.com/~cdp/cptd-faq/>

„DNS Security Paper” (Craig Rowland):

☞ www.psionic.com/papers/dns/

Néhány érdekesebb RFC

☞ <http://www.rfc-editor.org/>

1035 A DNS-ről általában

1183 A forrásbejegyzések (RR) formája

2308 Negatív gyorstárazás

2136 Dinamikus frissítések

1996 DNS Notify (értesítések)

2535 DNS biztonsági bővítések

Néhány DNS/BIND biztonsági tanács

☞ <http://www.cert.org/>

CA-99-14: Multiple Vulnerabilities in BIND

CA-2000-03: Continuing Compromises of DNS Servers

CA-98-05: Multiple Vulnerabilities in BIND

CA-97.22: BIND

Összefoglalás

Az itt ismertetett irányvonalak és módszerek jó kiindulópontként szolgálnak DNS kiszolgáló(ik)nk biztonságos kialakításához. E módszerek tökéletesebb megértéséért javaslom, hogy olvassuk el a BIND felhasználói útmutatóját (a legtöbb bináris csomagban megtaláljuk, de a ☞ <http://www.isc.org/> címről is letölthető). Egy másik hasznos segédanyag Liu DNS Security című bemutatója (ez PDF-formátumban is hozzáférhető).

Ugyanilyen fontos, hogy minden BIND-üzemeltető iratkozzon fel legalább egy levelezési listára, mely a felfedezett biztonsági hibákkal foglalkozik, illetve jó tanácsokat ad mindenféle biztonsági kérdésben. Az én kedvencem a CERT, hiszen elég régóta működik ahhoz, hogy megbízhatnunk benne, viszont kis terjedelmének köszönhetően könnyen kezelhető. A CERT legfrissebb jelentéseit is olvassuk el, hiszen a gondok ismerete elengedhetetlen egy biztonságos rendszer kiépítéséhez.



Michael D. Bauer (mick@visi.com) az ENRGI nevű hálózati tanácsadó cég minneapolis-i képviselőjén dolgozik. 1995 óta a Linux elkötelezett híve, az OpenBSD-nek 1997 óta szerelmese. Hírhedt ama „perverziónjáról”, hogy imád tökéletes rendszereket felépíteni a leghaszálhatatlanabb gépekből is. Mick örömmel várja olvasóink kérdéseit.



© Kiskapu Kft. Minden jog fenntartva

Könnyű álmok (1. rész)

Avagy hogyan maradjunk nyugodtak a sok veszély ellenére.

Omne principium difficile... Bolygónk új fajjal gazdagodott: a homo szájberspésszel. Gigabájtokkal ébred és nem is talál ki hatvannégy bites világából egész álló nap. Nos igen, a számítástechnika napjaink csillaga. Nap mint nap hallani, hogyan teszi életünket egyszerűbbé, szebbé és hatékonyabbá, persze a rossz oldalai sem maradhatnak rejtve. Gonosz számítógépes bűnözők ülnek naphosszat mindentudó gépeik előtt, és azon dolgoznak, hogy százcsillió dollárt utaljanak át a szegény árvák bankszámláiról sajátjukra. Eddig a mese. Mi a valóság? Akad néhány fülsértően hozzáférő, aki túl gyakran és sokat beszél az informatikáról. Az ember azonnal megtudhatja tőlük, hogy mi a helyzet manapság, mi a „trendi” és mi nem. Van azonban egy apró gond: a félrevezetett felhasználók tartanak az Internet kalózeitól anélkül, hogy bármit is tudnának róluk és ismernék a védekezés lehetőségeit. A cégek felelős vezetői hihetetlen összegeket költenek méregdrága védelmi rendszerekre, amelyek biztonsági szintjét azonban senki sem tudja hitelt érdemlően bizonyítani. A biztonsági rendszerek felépítését és működését belső emberek ritkán látják át, mivel kevés cég engedheti meg magának hivatásos biztonsági szakember alkalmazását. Mások hajlamosak lebecsülni a nehézségeket, mivel ők hozzáfértek és mindenkinek be tudják bizonyítani, hogy miért nincsenek veszélyben. Ez a hamis biztonságérzet természetesen kihat az egész cégre, amikor pedig becsap a villám, mindenki csak néz és csodálkozik.

Érdemes tehát a biztonságról beszélni, és kicsit többet megtudni róla. A Linux-rendszerek mind az otthoni, mind a céges felhasználási területen rohamosan terjednek, így egyre többen fedezik fel az ingyenes rendszerek előnyeit. Kevesen tudják azonban, hogy kis odafigyeléssel és némi tapasztalattal kitűnő védelmi rendszer építhető belőlük. Szándékaink szerint írásunk egy olyan sorozat kezdete, melynek célja, hogy érthetővé tegye a rendszerek sérülékenységének okait, valamint elősegítse azok elhárítását. Nem cél azonban minden lehetséges rész feltárása, hiszen ez messze túllépi kereteinket. Fontos cél a figyelem felhívása a veszélyekre, de emellett mindenhol igyekezzünk megmutatni az adott bajok megakadályozásának módszereit is. A sorozat előrehaladtával egyre mélyebbre hatolunk a Linux-rendszer védelmének rejtelmeibe, és mire vég-zünk, egy alapos áttekintő képet kapunk a biztonságos rendszerről. Terveinkben szerepel, hogy mélyebben foglalkozunk adott rendszer hálózat felőli védelmével, egy hosszabb lélegzetű részben pedig a helyi biztonság növelésével. Ezek után következik a legérdekesebb témakör: a tűzfalak. Bemutatjuk a fajtáit, azok sajátosságait, milyen eszközök léteznek Linuxra a hálózati határvédelem kialakítására. Mire ide elértünk, addigra valószínűleg végleges lesz a 2.4-es rendszermag első példánya, így megismerkedhetünk annak lehetőségeivel. Végül bemutatjuk egy Linux-alapú védelmi rendszer tervezését és kivitelezését, mindezt ingyenes tűzfalalkalmazás segítségével. Így már érthető a főcím mondanivalója: véd magad és légy nyugodt.

Az első nyugodt álom: hálózati rendszerek biztonsága

Manapság mindenki a szupersztrádáról beszél, a nagytudásúak és a hangos elektronikus üzleti (e-business) reklámok azt azonban elfelejtik megemlíteni, hogy az új szabadság veszélyeket is rejt magában. Más összefüggésben ugyan sokat hallhatunk emlegetni a számítógépes bűnözésről, a kettő összefüggéseit azonban tudatosan elfe-

lejtik említeni. A sajtó döntő többségét olvasva nem is fedezzük fel az összefüggést. Balgaság lenne azt gondolni, hogy ha a kalózkodó a NASA számítógéprendszerét is feltörök, pont a mi otthoni banki átutalásokra használt kis programunkat hagyják békén. Az is kőszá tévhit, hogy gonoszok csak az Interneten vannak. Sőt! A számítógépes bűnelkövetők többsége a cég belső emberei közül kerül ki, legyen az multicég vagy egyetemi kollégium. A komolyabb támadásokat általában az anyagiak ösztönzik, de az indítékok között előfordul bosszú, erőfitogtatás, kalandvágy és ki tudja, mi még. Mivel nem tekinthetünk el a belső emberek által elkövetett ténykedésektől, így a hálózati támadásokat nem célszerű csak az Internetre értelmezni. A belső hálózat ugyanolyan támadási felület, azon belül is a lehető legnagyobb védelemről kell gondoskodnunk. A cikksorozat első néhány részének célkitűzése, hogy megvilágítsuk, miként mérsékkelhető annak az esélye, hogy valaki sikeres hálózati támadást intézzen az otthoni rendszer vagy a cég kiszolgálója ellen.

Mekkora az egészséges veszélyérzet?

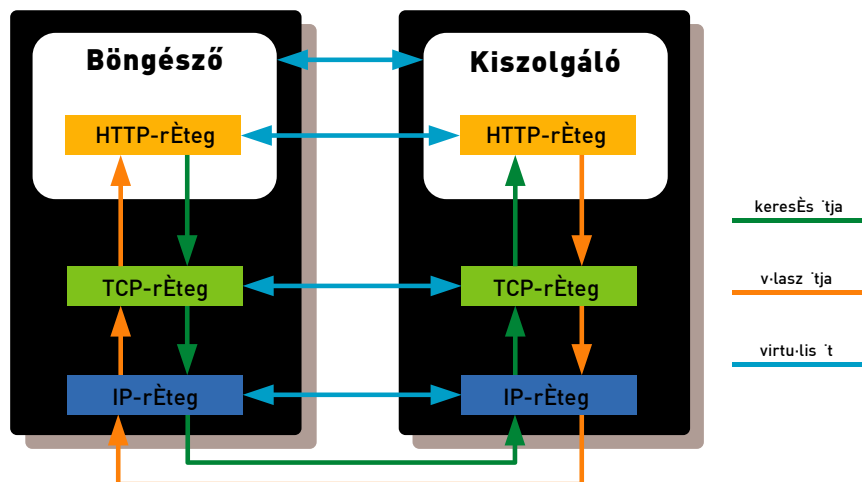
Általában elmondható, hogy a védelem ereje arányos a ráfordított energiával és az erőforrásokkal. Azt, hogy egy rendszert mennyire kell, vagy érdemes védeni, kizárólag a rendszer felhasználói tudják eldönteni. Gyakran hangzik el az a kijelentés: „Hozzánk nem érdemes betörni, itt nincs semmi értékes...” Ezzel kapcsolatban két dolgot vizsgáljunk meg közelebbről.

Érdemes védekezni?

Nincs olyan rendszer, amit ne lenne érdemes feltörni. Mindig védekezz!

Példaként említeném a következő elképzelt esetet: a cégnél van egy nyomtató, amit többen is használnak. Az elérhetőség céljából egy éve összeraktak egy régi elavult számítógépből, valamint egy linuxos nyomtatókiszolgálóból álló rendszert. Mivel a Linux-alapú kiszolgáló megbízható, sőt, a gép áramkimaradás esetén CD-ről indul, így hozzá sem kell nyúlni. Biztonsági intézkedéseket sem hozta, hiszen nincsen rajta semmi fontos, csak egy egyszerű nyomtatókiszolgáló. Egy nap a rendőrség kíséretében néhány Interpol-nyomozó érkezik, lefoglalják a nyomtatókiszolgálót és kihallgatják a cég összes alkalmazottját. Mi történt? Az értéktelen kiszolgáló biztonsági rendszerén valaki könnyedén keresztülhatolt, és ezen keresztül törte fel a Pentagon honlapját, majd ott malac képeket helyezett el.

A kalóz konkrét cél nélkül csak akkor tör be valahová, ha az egyszerű. Ugyanis ha egy rendszerben nincs olyan érték, ami miatt érdemes lenne feltörni, akkor a betörés leggyakoribb oka a rendszer védtelensége. Ha az erdő körül nincs kerítés, és a tóhoz az erdőn keresztül vezet egy rövidebb út, akkor mindenki arra fog menni, ha viszont van egy erdőkerülő két hatalmas kutyával, akkor inkább a hosszabb utat fogják választani. Így működik ez a számítógépes rendszerek esetében is. Ha nem fordítunk kellő figyelmet a biztonságra, akkor a rendszerbe könnyű lesz behatolni, és a kalózkodó átjárónak fogják használni rendszerünket, vagy használják erőforrásainkat. Mindenki érti, hogy a kalózkodó miként használják fel gépünket, amikor tovább akarnak lépni, nem egyértelmű viszont az, hogyan lehet az erőforrásokat kihasználni. Ahogy egyre több felhasználónak van elektronikus levélcíme, úgy az árucikkek Interneten zajló reklámkozásában is egyre



1. ábra A webkiszolgáló és az ügyfél közti kapcsolat

nagyobb üzleti lehetőségek rejlenek. A legegyszerűbb megoldás a célközönség által olvasott levelezőlisták irattáraiból a levélcímek kigyűjtése, azután már gyerekjáték egy reklámlevél (más szóval levélszemét, angolul spam) elküldése. Ez azonban nem etikus, ráadásul óriási hálózati forgalmat okoz, így nem célszerű közvetlenül a cég levelező-rendszerét erre felhasználni. Ha egy kalóz feltör egy rendszert, akkor ott olyan programot helyezhet el, amely a továbbítandó levelet egy előre megadott címlistára a feltört rendszer erőforrásait felhasználva juttatja el. Eközben a rendszer tulajdonosa hálózati keresztmetszetnek egy részét nem tudja kihasználni, nem beszélve az erkölcsi kárról, ha az eset kiderül (mert általában kiderül, csak már régen késő). Ha a behatolás nem könnyű, akkor inkább másik áldozatot keres a kalóz, ugyanis minden egyes biztonságosabb rendszerre harminc könnyű préda jut. Ha azonban a kalóz határozott céllal támad meg egy rendszert, akkor a komolyabb biztonsági rendszer is kevésnek bizonyulhat. Itt következik annak a mérlegelése, hogy rendszerünk mennyire értékes. Ha elég erős a védelmi rendszer – például jó minőségű a tűzfal –, akkor a támadóknak annyi energiát kellene befektetni annak kikerülésébe, hogy inkább feladják. Ennek ellenére be kell vallani, hogy van olyan támadó, aki ellen nem lehet védekezni. Ha például egy másik földrész hárombetűs hivatala akar behatolni...

Tévedni emberi?

Nem tudsz olyan kevés időt tölteni az Interneten, hogy ne találjanak meg. Mindig számíts támadásra!

Az alábbi példa velem esett meg: egy barátomnak segítettem Linuxot telepíteni. Beállítottuk a behívást, éppen azt magyaráztam, hogy mire jók a naplóállományok, és... Mít látok? Valaki megpróbálta feltörni a még csak négy perce az Interneten lógó névtelen rendszert. Hogyan találhattak meg? Egyszerű: nem ezt a gépet keresték, hanem egy tesztelőt, amit rosszul állítottak be az adott szolgáltató címtartományában. A naplóállományokból látszott ugyan, hogy a támadó program nem Linux-rendszert keresett, hiszen olyan porton (kapun) próbálkozott, amely egy másik rendszer tipikus hibáját rejtheti, de nem szabad megfeleledkezni arról, hogy a Linux terjed. Egyre gyakrabban keresnek majd rosszul beállított Linux-rendszert is.

Foglalkozunk két újabb tévhitel: a rövid idő és az ismeretlen cím biztonságnövelő hatásaival. Egyesek hajlamosak azt gondolni, hogy ha nem futtatnak nyilvános szolgáltatást – vagy senkinek sem szólnak róla –, és csak napi fél órát szörföznek, akkor nincsenek veszélyben. Ha helyesen állítjuk be a csomagszűrőt és figyeljük a napló-

állományokat, akkor gyorsan rájövünk, hogy ez bizony önámítás. Vannak rosszindulatú emberek által írt olyan programok, úgynevezett férgek (worm), melyeknek fő célja a terjedés (úgy, mint a vírusoknak), és egyszerű próbálkozással keresik leendő áldozataikat. Ezek ellen nem véd semmilyen névtelenség, különösen ha eleve olyan Internet-címtartományokkal dolgoznak, amelyeket a nagy szolgáltatók az oda behívó gépeknek osztanak ki. Azért dolgoznak gyakran éppen ezekkel a címekkel, mert a behívó rendszerek nagy része rosszul van beállítva, a biztonság a fent felsoroltak miatt e rendszerek gazdáinak nem túl fontos szempont. Ha sikerült bejuttatni a támadó programot, akkor már csak a rendszer újratelepítése adhat biztonságos megoldást. Mivel a legtöbb Linux-rendszer behívással kapcsolódik az Internetre,

fontosnak tartjuk újra felhívni a figyelmet: *a behívással kapcsolódó gépek biztonságának is figyelmet kell szentelni!*

Mennyibe kerül mindez?

Egy rendszer legkevesebb annyit ér, amennyibe a helyreállítása kerülne. Számolj!

Például az ügyvezető úgy gondolja, nincs szükség külön tűzfalra, mert a rendszeren tárolt legértékesebb információ a tavalyi könyvelés, annak a végeredményét pedig már úgymis nyilvánosságra hozták... Egyszer egy tréfás kedvű kölyök egy friss biztonsági rés működését bemutató program segítségével betör a kiszolgálóra. A támadó ráébredve arra, hogy mit követett el, gyökértől letörli az összes állományt a rendszerről, hogy eltüntesse saját nyomain. Mi történt? Mivel nem volt a könyvelést újra el kell készíteni. Ha ekkor jönne egy adóellenőr, az nagyon kellemetlen és drága mulatság lenne. Sokkal drágább, mint amennyire a rendszer kiegészítése lett volna. Rendszerünk, legyen az kiszolgálórendszer a hozzá tartozó adatokkal, vagy erőforrás-szolgáltató, mindenképpen okkal van a hálózatban. Azt lehet mondani, hogy másnak nem fontos, de ez nem jelenti azt, hogy nem kell óvni. Célszerű a következő szempontokat szem előtt tartani a védelmi rendszerre fordított eszközök meghatározásánál:

- fel kell mérni, hogy mennyit ér a rendszer nekünk, valamint azt is, hogy mennyit érhet a rendszer egy esetleges támadónak,
- figyelembe kell venni a jelenlegi átlagos védekezési erőfeszítéseket a hasonló jellegű rendszereken,
- fel kell mérni, kit tesznek felelőssé egy tőlünk kiinduló számítógépes bűncselekmény miatt,
- át kell gondolni, hogy mekkora erkölcsi kárt jelenthet, ha a gépet feltörik, és a kalóz saját céljaira használja fel.

Csak mindezek figyelembevételével szabad a rendszer védelméről dönteni. A rendszer védelme a vas és az operációs rendszer védelmével kezdődik, erre a későbbiek során részletesebben kitérünk. Ha a hálózat felől szeretnénk biztonságosabbá tenni rendszerünket, akkor meg kell ismerni a hálózatszintű és alkalmazásszintű támadásokat.

Hálózatszintű támadások

A számítógépes hálózatok hőskorában a biztonság nem volt tervezési szempont, így a kapcsolattartó protokollok általában könnyen támadhatók. A számítógépek közti információcsere jelenleg a leggyakrabban

TCP/IP protokollon keresztül folyik. Ez egy protokollcsoport, amely mindig tartalmazza az IP-t (Internet Protocol) és feladatától függően néhány egymásba ágyazott egyéb protokollt. Például amikor letöltök a kedvenc honlapomról egy képet, az alábbi rétegek működnek:

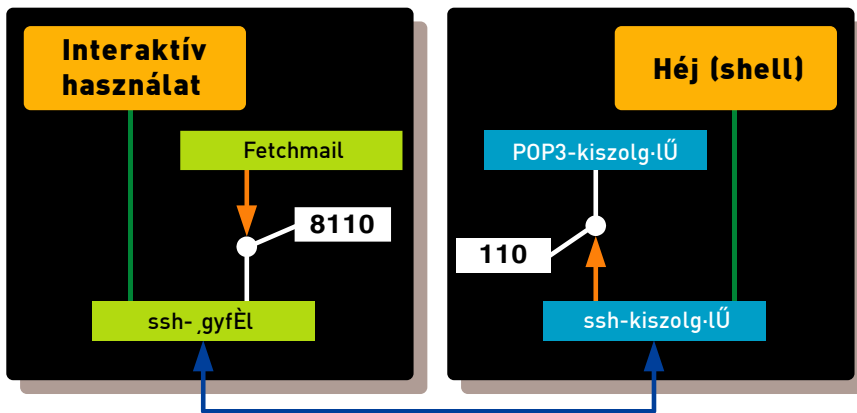
- A két gép között az IP gondoskodik az adatok továbbításáról. Az IP egy megbízhatatlan hálózatokra tervezett csomagkapcsolt protokoll, amelynél az egyetlen tervezési szempont az volt, hogy az adatok eljussanak A-ból B-be. Minden csomag tartalmazza a feladó és a címzett gép címét, ezek a címek azonban jelenleg nem hitelesítettek.
- A TCP létrehoz egy, a felsőbb protokollok számára folytonosnak és hibamentesnek tűnő kapcsolatot. A felettes protokollok adatait a TCP feldarabolja, sorszámozott csomagokba teszi, és elküldi. Így a túlóldali rendszer ellenőrizni tudja, hogy minden csomag pontosan egyszer érkezzon meg. Továbbá ő gondoskodik arról, hogy a csomagok a megfelelő sorrendben össze legyenek állítva. Lehetővé teszi még a célgép egy adott szolgáltatásának megcímzését is.
- A HTTP protokollt kifejezetten állománytovábbításra fejlesztették ki, az ő adatairészebe kerül be a tulajdonképpeni adat – jelen esetben a kép.

Ezt a folyamatot az első ábra mutatja be. Ez a séma csak szemléltető jellegű, a rendszer a valóságban lényegesen összetettebb. A szemléltetésre néhány később bemutatott támadás átláthatóbbá tétele miatt van szükség. Az adatátvitel közbeni támadásoknak alapvetően két félék: passzív és az aktív.

nem szól bele a két gép közti adatátvitelbe, csak figyelj azt. Mivel az Interneten jelenleg használatos protokollokat viszonylag régen tervezték, általában nincs beépített titkosítási lehetőségük, így ezzel a támadási móddal még sokáig számolnunk kell. Néhány jelentősebb protokoll támogat valamilyen szintű titkosítást, de ez viszonylag ritka, és nem minden esetben kielégítő erősségű. Jó példa erre az elektronikus kereskedelem dinamikus terjedésével egyre gyakrabban használt HTTPS protokoll, amely a HTTP titkosítással kiterjesztett változata.

Ne figyelj ide!

Könnyen lehallgatható például a jelenleg kis- és közepes méretű hálózatoknál leggyakrabban használt Ethernet, ahol egyszerűbb esetben az azonos hálózaton lévő eszközök egymás közti forgalmát láthatják. Így, ha egy cég vagy iskola rendszerébe egy támadó képes bejutni, akkor a hálózaton áramló titkosítatlan adatokat megszeresheti. A hálózati forgalom irányítását ígérő eszközök (például hálózati kapcsolók – switchek) hamis biztonságérzetet adhatnak, hiszen gyakran korántsem olyan biztonságosak, mint amilyennek kiáltják őket. Az Internet felé irányuló forgalmat mind a szolgáltatató alkalmazottai, az oda behatoló kalózkodók lehallgatják. Ha valaki elég rámenős, a bérelt vonali forgalmat is lehallgathatja egy telefonpóznára szerelt eszközzel. Tehát a hálózati rendszerekről nem feltételezhetjük, hogy e támadás ellen hatékony védelmet tudnak adni. Az egyetlen biztos megoldás a hálózaton átáramló bizalmas adatok valamilyen módon történő titkosítása.



2. ábra Titkosított csatorna létrehozása ssh segítségével

Passzív támadásformák

Ha van rá mód, titkosítsd kapcsolataidat!

Példaként lássuk a következőket: Sára rendszergazda egy Kft.-nél. A személyzeti vezető nem bízik meg benne, ezért a levelezőrendszer üzemeltetői jogát elveszi tőle. Mivel tisztázni szeretné, hogy valóban megbízhatatlan-e a munkatárs, elektronikus levélben megkeresi Sára előző munkahelyét. A kérdésre megírták, hogy valóban gondok voltak vele. Sajnos lopott és megbízhatatlan. A személyzeti vezető elhatározza, hogy már másnap megváltik, de éjszaka a cég teljes bevételének lába kél, és Sárát sem látják soha többé. Mi történt? A személyzeti vezető nem volt tisztában azzal, hogy a levelek továbbítására szolgáló SMTP (Simple Mail Transfer Protocol) protokoll nem titkosított, így Sára tudomást szerezhetett a levelek tartalmáról. A két gép közti adatáramlás alapesetben semmilyen titkosítással nincs védve, így ha valaki azt le tudja hallgatni, akkor az átáramló adatok birtokába jut. Ezt a támadási módot hallgatózásnak hívják (sniffing), és klasszikus esete a hálózati szintű támadások egyik fajtájának, a passzív támadásnak. Ezeknél a támadó semmilyen úton

Titkok

Ha a protokoll nem támogatja a titkosítást, akkor kiegészítő eszközzel elérhető, hogy a két fél titkosított csatornán beszélgeszen. A leggyakrabban használt ilyen kiegészítők az úgynevezett SSL (Secure Socket Layer – titkosított csatornázó) programozási könyvtára épülő eszközök, ezek segítségével a titkosításra nem képes protokoll egy titkosított virtuális csatornán halad át. Például a levelek letöltését lehetővé tevő kiszolgáló, mely az egyszerű POP3 (Post Office Protocol) protokollt használja, kiegészíthető egy olyan réteggel, amivel a levelek letöltése titkosítva történik. Ezt a kiegészítést a levelezőprogramok jó része már támogatja, illetve a letöltés

alkalmas eszközzel egyszerűen megoldható. Ez Linux alatt például stunnel, valamint a fetchmail segítségével könnyen kivitelezhető.

Csatornázás

Jó megoldás egy rendszer távoli elérésére az ssh (secure shell) használata, amely titkosított csatornán keresztül viszi át az adatokat, és újabb csatornák létrehozását is lehetővé teszi. Ha a távoli gép POP3 portját (kapuját) szeretnénk saját gépünk helyi csatolóján látni, akkor a következő paranccsal kezdeményezhetjük a kapcsolat felépítését:

```
$ ssh -L 8110:localhost:110 -l atya -v gep.neve.hu
```

Ha a távoli ssh-kiszolgáló beállítása lehetővé teszi a port továbbítását (port forwarding), akkor a helyi gép 8110-es portjához kapcsolódáskor, az ssh ügyfél ezt a kapcsolatot továbbítja a távoli gép 110-es portjára (lásd a 2. ábrát). Azért kell a helyi gépen 8110-es portot használni, mert az egyszerű felhasználóknak nincs joguk a gép első 1024 portját megnyitni. Így egy nem titkosított protokollt,

ami a hálózaton való keresztülhaladáskor lehallgatható lett volna, egy titkosított csatornán vezetünk keresztül. Ez a titkosított csatorna kialakításának legegyszerűbb módja, és jelenleg már aligha találunk olyan rendszert, amit ne érhetnénk el ssh protokollon keresztül.

Egy másik lehetőség az átvitelre kerülő adatok titkosítása, erre jó példa a gpg (GNU Privacy Guard), amely a gpg adatállományok titkosítására vagy hitelesítésére szolgál, leggyakrabban bizalmas adatok levélben való továbbításakor használják a Sárával kapcsolatos gondot megoldotta volna. Megfelelően nagy méretűre választott titkosító kulcs segítségével a titkosítás olyan erős, hogy bizonyos országokban használata még ma is tiltott, ugyanis az államhatalom nem tudja megfejteni...

Végző megoldás

Ha több csatornát akarunk továbbítani, melyek különböző protokollokon beszélgetnek, akkor célszerű lehet egy úgynevezett VPN (Virtual Private Network), azaz egy virtuális magánhálózat kiépítése. A VPN olyan – célszerűen titkosított – csatornát hoz létre két független hálózati rendszer között, amely lehetővé teszi a két rendszer védett, bizalmas kapcsolattartását. Felépíthetünk tehát egy olyan rendszert, amely az Interneten keresztül kapcsolódik össze, mégsem kell tartani a lehallgatás veszélyétől, mert a végpontok közti adatátvitel titkosított. Virtuális magánhálózatot általában egy nagyobb cég telephelyei között szokták kiépíteni, hiszen így megtakarítható a telephelyeket összekötő bérelt vonal költsége.

Aktív hálózati támadások

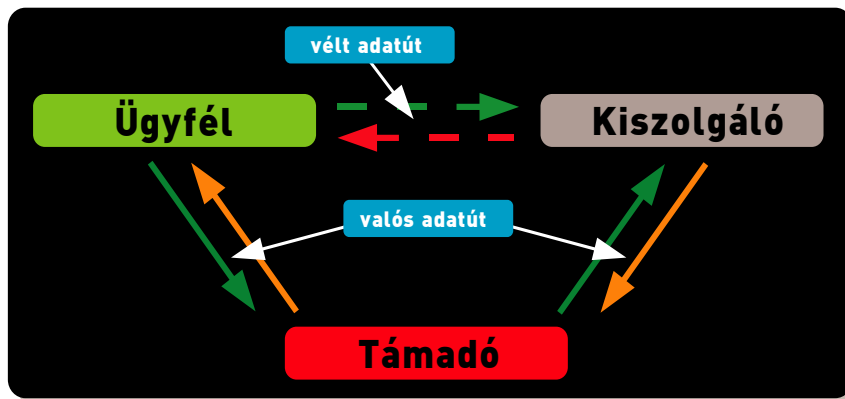
Akkor beszélünk aktív támadásról, ha a támadó valamilyen szinten beleszól a két fél közti adatáramlásba. Például a támadó a két gép között helyezkedik el, a forgalom rajta keresztül folyik és az átáramló adatfolyamot módosíthatja. Az aktív támadások sokfélesége, valamint a védekezés különleges volta miatt itt csak néhány formájáról ejthetünk szót.

A jó, a rossz és a jó

Bizalmas adatok továbbítására mindig használj titkosított csatornát! Tegyük fel, egy cégnek Linux-alapú könyvelőrendszere van. Az arra illetékesek telnet protokoll segítségével belépnek a kiszolgálóra, és ott végzik el a cég könyveléssel, illetve átutalással kapcsolatos tevékenységeit. A cégnél dolgozó *Terep János* úgy szeretne némi pluszjövedelemhez jutni, hogy amikor a főkönyvelő belép a könyvelési rendszerbe, akkor a főkönyvelő gépével elhiteti, hogy ő a könyvelést tartalmazó kiszolgáló, a kiszolgálóval pedig azt, hogy ő a főkönyvelő gépe. Így mindkét gép úgy fog hozzá fordulni, hogy azt hiszi, a másik rendszerrel beszél. Így Jánosnak lehetősége nyílik a könyvelő által végrehajtott utasítások közé csempészni néhány kisebb átutalást. Hogyan történhet ez meg? Az Ethernet hálózat mindamellett, hogy könnyen lehallgatható, könnyen meg is téveszthető.

A példában szereplő helyzetet a szaknyelv „IP spoofing”-ként (IP-megtévesztés) emlegeti, és a „man-in-the-middle” (középre belépő) támadások kategóriájába tartozik (lásd 3. ábra). A hiba oka egyértelműen az Ethernet hálózati protokolljának hiányossága. Ha a rendszerben meg tudnánk teremteni annak feltételeit, hogy a két gép közt áramló adatokhoz más ne férhessen hozzá, a közöttük lévő hálózati elemek biztonsága sérthetetlen legyen, akkor már csak kis valószínűséggel támadható a rendszer alacsony szinten. Újra figyelemzetez azonban mindenkit: a hálózati eszközök (jelelőszók, há-

lózati kapcsolók) nem biztonsági eszközök. Ha az eszköz alkalmas is a biztonsági feladatok ellátására, a beállításainak tökéleteseknek kell lenniük ahhoz, hogy biztonságosnak lehessen mondani, és még



3. ábra A gépek megtévesztésével a támadón keresztül zajlik az adatátvitel

így is kerülnek a felszínre olyan hibák a biztonsági levelezőlistákon, amelyek elgondolkodtatóak. Ha azonban az adatátvitelt titkosított csatornára bízunk, ahol a két rendszer mindenekelőtt azonosítja egymást (ilyen például az ssh), akkor egy középen elhelyezkedő támadónak csak akkor van esélye, ha már az első kapcsolatfelvétel előtt a két gép között van, és folyamatosan felügyeli a két gép közt zajló forgalmat. Ez igen komoly előkészítést igényel, és kikerülhető, ha a két gép egymás azonosítására szolgáló kulcsait első alkalommal nem a hálózaton át cseréli ki, hanem a két rendszer gazdája a kulcsokat független fizikai csatornán – például lemezen – juttatja el a egymáshoz, valamint használat előtt hitelesíti. Utóbbi megoldás biztonsági szakemberhez méltó jól fejlett paranoiáról árulkodik. :)

Záró szavak

Hálózati szintű támadásokkal foglalkozunk a következő alkalommal is, akkor azonban már a Linux-rendszer beállításaira is kitérünk. Megtárgyaljuk, hogy milyen beállításokkal és intézkedésekkel csökkenthető a sikeres hálózati támadások esélye. Szó lesz arról, hogy miként tanácsos beállítani egy Linux-rendszer csomagszűrőjét behívás esetén, valamint a cégek szolgáltatásokat is adó kiszolgálóin.



Mátó Péter (atya@andrews.hu), informatikus mérnök és tanár. Biztonsági rendszerek ellenőrzésével és telepítésével, valamint oktatással foglalkozik. 1995-ben találkozott először linuxos rendszerrel. Ha teheti, kirándul vagy olvas.



Borbély Zoltán (bozo@andrews.hu), okleveles mérnök-informatikus. Főként Linuxon futó számítógépes biztonsági rendszerek tervezésével és fejlesztésével foglalkozik. A 1.0.9-es rendszermag ideje óta linuxozik. Szabadidejét barátaival tölti.

A főszerkesztő ezúton kér elnézést a tisztelt olvasótól és a szerzőktől, ha úgy érzik, hogy a szerzők „technicus terminusainak” magyarázása csorbította a szöveg érthetőségét.

© Kiskapu Kft. Minden jog fenntartva

Linux a beágyazott ipari alkalmazásokban

Egy elemzést nem támogató ipari üzem adatainak kezelésére Linuxot futtató PC segítségével protokollátalakító átjárót helyeztek üzembe.

Egy olajfúró kút nagy turbókompresszorát egy tizenöt éves rendszer, DCS (Digital Control System) vezérli. A rendszer teljes körű felügyeletet tesz lehetővé, valamint egy nyomtató segítségével naplózza a lényeges eseményeket. Évek óta megfelelően működik, de a vállalat vezetése ezt a gépet is szeretné bekapcsolni a központi SCADA rendszerbe. A rendszer összegyűjti a különböző gépek adatait, és elérhetővé teremt az egész üzem egységes, átlátható irányítására.

A meglévő PLC-alapú DCS rendszer a felépítése miatt nehezen bővíthető, és sem a gép, sem a program szintjén nem teszi lehetővé adatátviteli módszerek beépítését, pedig ezek igencsak szükségesek lennének a feladat elvégzéséhez. A DCS rendszer újratervezése a nélkülözhetetlen változtatások figyelembevételével nagyon sokba kerülne.

A megoldás egy olyan, a DCS rendszerhez kapcsolt protokollátalakító átjáró, amely elfogja a nyomtatónak küldött adatfolyamot, a szöveg sorai közül kigyűjti az adatokat, és elérhetővé teszi őket a SCADA rendszer számára. A nagyrendszer irányába egyszerűségének köszönhetően, a Modbus RTU szabványprotokoll biztosít, ugyanis ezt már széles körben használják az üzemben.

A rendszer felépítése

Az ilyesféle alkalmazásoknál a legnagyobb kihívást a különleges környezet jelenti. Az üzem földrajzilag egymástól távol eső területeken helyezkedik el, valamint az összes, az iparban szokásos gond fennáll: a por, az elektromos hálózati zavarok, a nagymértékű hőingadozás és így tovább. Az ipari PC-k könnyen beszerezhetőek a piacon, csak van néhány olyan sajátosságuk, amivel egy átlagos PC esetében nem találkozunk.

- A processzoron nincs hűtőventilátor. A PC-k gyenge pontjait a mozgó alkatrészek képezik, és a lekicsinyített ventilátor különösen veszélyes e tekintetben. A főventilátorra mindenképpen szükség van, ezenkívül megfelelő szűrőket szintén fel kell szerelni. A processzorventilátor alkalmazását ki lehet kerülni úgy, hogy akkora órajelet határozzunk meg, amit a processzor túlemegegedés nélkül még elvisel.
- Nincs bennük merevlemez. Ezt egy megbízható Flash EPROM-lemez helyettesíti. Lehetőség van ugyan hajlékonylemez meghajtó beépítésére, de ez csak a programok feltöltésére alkalmas. Ebből következően a kihasználható lemezkapacitás igen korlátozott, általában nem több, mint 64 MB, sőt, esetünkben mindössze 8 MB.
- A szabványos RS-232 vezetékek nem felelnek meg a zajos környezetben történő soros adatátvitelhez. Erre a feladatra inkább az RS-485 szabvány a használatos.

- A PC általában billentyűzet és monitor nélkül üzemel, de mindkettőt csatlakoztathatjuk a géphez a karbantartási munkálatok elvégzése céljából.

Az előbbi megszorításokat és a beágyazott alkalmazásokra jellemző munka feltételeit figyelembe véve, meghatározhatjuk a programmal szemben támasztott követelményeket. Ezek a következők:

- Kevés háttétárral is működnie kell.
- Be kell érnie szerény processzorleljesítménnyel is.
- Emberi beavatkozás nélkül kell elindulnia.
- Minden gond nélkül újra kell indulnia áramkimaradás esetén, történjen ez programból vezérelt módon, vagy valamilyen hiba miatt.
- Fontos, hogy ne legyen szükséges különleges szakképzettség használatához.

Az világos, hogy a nyomtatandó adatfolyamot elemző programot a semmiből kellett felépíteni, a Modbus protokoll megvalósításánál pedig fel lehetett használni a meglévő könyvtárakat.

Az operációs rendszer kiválasztása

Miért nem valósídejű operációs rendszert használunk? Az adott feladat leírásából ki kell tűnnie, hogy itt nincsenek komoly valósídejű követelmények. A két bemeneti/kimeneti csatorna sebessége eleve felső korlátot szab az adatátvitel számára, és az előre nem látható, kisebb-nagyobb soros átviteli fennakadások sem szabad, hogy gondot jelentsenek. Ez teszi lehetővé, hogy hagyományos operációs rendszert használjunk a valósídejű rendszerek helyett. Ez jelentős költségmegtakarítást jelenthet.

Miért nem a DOS? A feladat aszinkron, azaz a rendszernek egyszerre kell fogadnia a DCS által küldött nyomtató-kimenetet, valamint válaszolnia a SCADA rendszer felől érkező kérésekre. Ez a két folyamat egyáltalán nincs összhangban egymással, tehát a DS nem jöhet szóba. Erőforrás szempontjából igen kedvező megoldás lenne, de a megvalósítás rendkívül sok alacsony szintű programozást igényelne.

Miért nem a Windows 95/98/NT? Az MS Windows (ezen belül főleg az NT) egyre nagyobb elismerést szerez a nagyvállalati felhasználás terén a nagyszámú és minden igényt kielégítő szavatossági és egyéb szolgáltatásoknak köszönhetően, valamint mert egy nagynevű kereskedő terjeszti. A gép adottságai következtében rendkívül nehéz, vagy talán lehetetlen feladat lenne a Windows bármely változatának a korlátozott tárkapacitáshoz és a processzorleljesítményhez igazítása. A Windows CE nem eléggé megbízható, tehát nem érdemes ezen a vonalon gondolkodni. A Windows többfeladatos megvalósítása is elég gyenge, és így az egymás mellett futó alkalmazások megírása is komoly kihívás.

www.kiskapu.hu

Magyar és angol nyelvű számítástechnikai szakkönyvek boltja

Miért a Linux? A cégvezetők általában félnek dönteni olyan kérdésekben, amelyek nem kifejezetten ipari természetűek. De nem haboznak a döntéssel, ha egy olcsó, nagy háttérrel bíró, neves forgalmazó termékéről van szó. A Linux még nem tart itt, így a kedvező döntés elősegítése céljából a szakmai adatoknak kell elég meggyőzőnek lenniük. Úgy hiszem, a Linux képességei sok szempontból meggyőzőek lehetnek, és számos érv szól mellette.

- A Linux-rendszer az adott igények szerint testreszabható, és ez a rendszermag beállításaira is vonatkozik.
- A rendszer testreszabásáról szóló leírás világos és könnyen hozzáférhető.
- Mivel a Linux RAM-lemezzel is képes futni, a Flash EPROM csak rendszerindításkor használatos. Ily módon a lemez akár írásvédett is lehet.
- A fejlesztés a célrendszeren is történhet.

Van azonban néhány hátránya is

- A nem szabványos eszközökhöz igen nehéz meghajtóprogramokat találni.
- A különböző gyártók termékeinek támogatása sokkal körülményesebb.

A részletek

A gép megvalósításához két, nem szabványos eszközre van szükségünk: a Flash EPROM-ra és az RS-485 soros csatlóóra. Ezek egyike sem okoz semmiféle gondot: a Flash EPROM használata ugyanúgy történik, mintha szabványos DOS-szal formázott lemez lenne, a soros csatlók pedig a COM kapukat helyettesítik, és a szabványos soros meghajtóprogram kezeli őket.

A programok felépítése

A protokollátalakító átjárót megvalósító program szerkezete igen egyszerű, ugyanis két folyamatból áll: egy adatfogadó és egy Modbus kiszolgáló folyamatból. Az adatokat fogadó a nyomtatótól a soros kapun át érkező sorokat megvizsgálja, kigyűjti belőlük az adatokat, és az értékeket egy megosztott memóriarészbe írja ki. A Modbus kiszolgáló pedig a SCADA rendszer felől érkező kéréseket fogadja a Modbus protokoll szabványának megfelelően, kiolvassa a kért adatokat a memóriarészből, és visszaküldi azokat. A hibakövetéshez két interaktív program áll rendelkezésre: az egyikkel az osztott memória tartalmát jeleníthetjük meg, a másikkal pedig írhatjuk azt. Természetesen mindkét program képes az átjáró folyamataival együtt futni. Az összes kód C nyelven íródott, az osztott memóriakezelés a System V szabványos, a folyamatok közti kapcsolatot támogató API segítségével valósul meg, ez biztosítja az osztott memóriarészek létrehozását, kezelését, és jelzőkkel támogatja az elérésük összehangolását. Mivel a feladat egyszerű, az összehangolás megoldható a teljes osztott terület zárolásával minden hozzáféréskor. Ezt az igen egyszerű megközelítést az teszi lehetővé, hogy a memóriaelérés gyors, az I/O-műveletek lassúak, ebből adódóan bármely folyamatnak viszonylag rövid ideig kell várakoznia. Az átjáró megfelelő működésének folyamatos ellenőrzése végett a memóriaterülethez egy számláló is tartozik, ennek értéke eggyel nő, ha az adatot fogadó folyamat végzett egy olvasási lépéssel. A számláló értékét a SCADA rendszeresen ellenőrzi, és hibát jelez, ha az egy bizonyos időtartamon belül nem növekedett.

Beállítások

Követve a Linux BootDisk-HOWTO irányelveit, egy kis Linux-rendszert építünk fel, a Red Hat 6.1 telepítőjéből kiindulva. Ez csak próbálgatás útján fog menni, ugyanis meg kell tudnunk, pontosan mely fájlok szükségesek az adott feladathoz. Igaz ugyan, hogy a Flash EPROM 8 MB tárolókapacitást ad, de az összes programot hajlékonylemezzel kell feltöltenünk a gépre, így célszerű a végső rend-

Kifejezések, meghatározások

PLC

A PLC programozható logikai vezérlőket széles körben használják az ipari alkalmazásoknál adatok kinyerésére, illetve vezérlőfolyamatok irányítására. A PLC-eket különleges nyelveken programozhatjuk, ezek segítségével adatokat nyerhetünk, kezelhetjük az állapotsori változásokat, riasztásokat válthatunk ki, valamint irányíthatjuk a különböző eszközöket.

Modbus RTU

A *Modbus RTU* egy nagyon egyszerű és megbízható protokoll, mely a programozható eszközök adatcserejét valósítja meg. Eredetileg a Modicon fejlesztette ki saját programozható logikai vezérlői számára, de később több gyártó is átvette, mivel a műszaki leírás bárki számára elérhető. Az eredeti kiépítésben a főegység küld parancsokat a soros kapura kötött több mellékegységnek, s így zajlik az adatcsere. A parancsok lényegében az adategységek írására, olvasására vonatkozó kérések. Két adattípust támogat a protokoll: a (16 bites) szót és a bitet. Minden parancs tartalmaz egy eszközazonosító számot, így csak a megcímzett eszköz válaszol a kérésre. Napjainkban bővítik ki a protokollt úgy, hogy az TCP/IP-kapcsolaton át is működjön.

SCADA

A *SCADA* (System Control and Data Acquisition) egy rendszerirányításra és adatelemzésre használt rendszer, mely a különféle forrásokból (egy üzem több alrendszeréből, vagy akár több üzemből) érkező adatok feldolgozását végző, a termelési folyamatokat összehangoló központi rendszernek felel meg. Egy termelőüzemben a SCADA-rendszert (mely a rendelkezésre álló nyersanyag mennyiségét, a raktári kapacitást, a termelési mutatókat, az egyes gépek állapotát, a megrendeléseket a tehergépkocsik útadatait, a hulladékkezelést stb. fogja össze) a gyártás irányítására, javítások elvégzésére, számításra, tehát a teljes termelési folyamat hatékonyabbá tételére használhatjuk.

szert a lehető legkisebbre tervezni. Az egyedien testreszabott rendszer tartalmazza a rendszermagot, valamint jó néhány hagyományos linuxos parancsot (elég bőkezűen jártunk el e tekintetben: inkább legyen kéznél minden, a karbantartáshoz esetleg szükségessé váló parancs) és az összes, kapcsolódó könyvtárat. A DOS formátumú lemezeket kezelő, betölthető modulok szintén a rendszer részét képezik. Ezek jól jöhetnek DOS formátumú hajlékonylemez vagy Flash EPROM használata során.

Mondanunk sem kell, hogy az átjárót kezelő négy program és néhány ASCII fájl szintén a rendszer része, ezekben a két futó folyamat kezdeti beállításait tároljuk. A Linux indulásához szükséges alábbi két fájl is megtalálható benne: a tömörített rendszermag (450 kB) és a szintén tömörített rendszerlenyomat (root image, 2500 kB). Ez csak egy kicsivel több, mint ami két hajlékonylemezre ráfér, így az egész csomag háromlemezes lett. A BootDisk-HOWTO sok ötletet ad ahhoz, miként csökkenthetjük a rendszerlenyomat méretét, de mi megelégedtünk az elért mérettel, és nem kívántunk erre több energiát fordítani.

Rendszerindulás

A kiválasztott Flash EPROM-lemezhez (M-Systems DiskOnChip terméke) létezik linuxos meghajtóprogram, és a rendszer el is indul róla. Ehhez a DiskOnChip meghajtóját a rendszermagba kell fordítani, de magát az eszközt is be kell állítani az arra szolgáló segédprogrammal, valamint a LILO egy különleges változata kell ahhoz, hogy a lemeztől elindulhasson a rendszer. Néhány próbálkozás után más megoldás mellett döntöttünk: a DiskOnChipet DOS-os indító-lemezként állítottuk be. Ezzel elkerültük mind a rendszermag újrafordítását, mind pedig a DiskOnChip ismételt beállítását (az eszközön máig az eredeti beállítások az érvényesek). Sőt, ez a megoldás sokkal megbízhatóbbnak tűnt a rendszer jövőbeli változatainak szempontjából is. A Linux-lenyomatok a DOS-os fájlrendszer alatt helyezkednek el, a Linuxot pedig a LOADLIN program indítja, ez körülbelül 160 kB-tal növeli meg a DOS alatt lévő fájlok méretét. A rendszer indulása tehát a következőképp zajlik:

1. Elindul a DOS.
2. Az AUTOEXEC.BAT-nak megfelelően elindul a LOADLIN, ez megkezdí a Linux betöltését. Ez a folyamat létrehoz egy linuxos fájlrendszert tartalmazó RAM-lemezt, majd kibontja a tömörített rendszerlenyomatot.
3. Elindulnak a protokollálatalkító folyamatok.

Ez utóbbiak a rendszer indításakor indulnak el, hiszen az inittab-ban bejegyeztük őket. Ez az automatikus újraindításukról is gondoskodik, ha netán valamelyik leállna.

A rendszer elindulása után, ha a képernyőt és a billentyűzetet is csatlakoztattuk, megjelenik a szokásos parancssor, és a rendszergazda bejelentkezhet. Ez lehetővé teszi a rendszer karbantartását, valamint

azt a fent említett két hibakereső program futtatását. Szükség szerint a Flash EPROM DOS-lemezként befűzhető a rendszerbe. (Erre akkor lehet szükség, ha a beállításfájlokat a rendszerlenyomat újraépítése nélkül szeretnénk megváltoztatni.)

Összefoglalás

Bemutattunk az ipari adatfeldolgozás területéről egy jellegzetes feladatot, melyre a Linux tökéletes megoldást kínál. 32 MB RAM és mindössze 8 MB háttértár segítségével már használható, és teljesen megbízható rendszert építhetünk fel. A rendszer indításakor a lemezt csak olvashatjuk, futásidőben pedig a fájlrendszer a RAM-lemezen helyezkedik el. Ez azt jelenti, hogy a rendszer váratlan leállása után sincs szükség lemezellenőrzésre, ami megakaszthatná az indítási folyamatot. Sőt, mivel az összes rendszerfájl csak olvasható lemezen van, nagyon valószínűtlen, hogy bárki is véletlenül módosíthatná őket. Teljes összeomlás esetén pedig a Linux képes újraindítani a folyamatokat.



Luca Fini (lfini@arcetri.astro.it)

húsz éve munkatársa az Osservatorio du Arcetinek, ahol programfejlesztőként dolgozik, illetve egy több mint száz gépet számláló hálózat rendszergazdája.

A gépeken Unix, Linux, Windows 95/98/NT, MacOS és további rendszerek is futnak.

Foglalkozik még irányítórendszerek tervezésével, valamint csillagászati eszközökkel, s alkalmanként független tanácsadóként dolgozik az iparban is.



Fejlett térbeli grafika: a GNU Maverik

Programozói környezet csúcsmínőségű, interaktív grafikus alkalmazásokhoz.

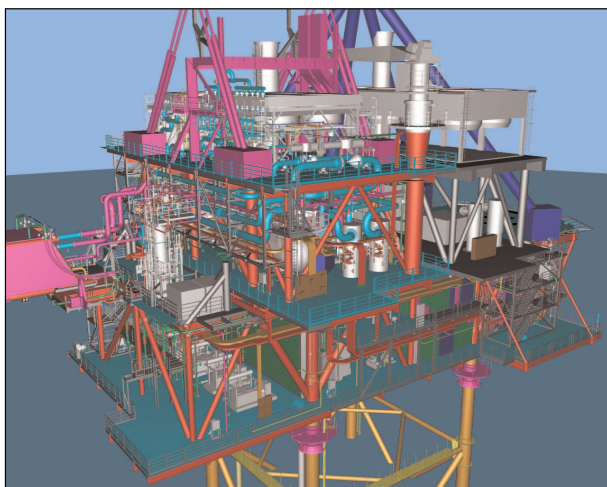
Egyszerűen modellezhetnek virtuális környezeteket (VK) a programozók az akadémiai kutatók által fejlesztett GNU Maverik segítségével. A fejlesztők elsődleges célja az volt, hogy a jelenlegi rendszerek hiányosságait kiküszöböljék.

A Maverik fő erőssége, hogy több alkalmazás összefogásával egységes keretet ad a fejlesztésnek, de számos beépített szolgáltatással igyekszik megkönnyíteni az alapokról induló felhasználók munkáját is. Lehetőségeit elsősorban a programozók használhatják ki, hiszen nem önálló programról van szó.

A Maverik a GNU része, és a GNU GPL alapján forráskódjával együtt szabadon terjeszthető. A csomagban gyakorlatok, példák és a teljes leírás is helyet kapott. 1999. februári megjelenése óta több ezren töltötték le, s a fejlesztőkhöz számos dicséző visszajelzés érkezett, üzleti és nem haszonelvű felhasználóktól egyaránt.

Cikkünkben a virtuális környezetet tervezők előtt álló akadályokról írunk, majd arról, hogy a Maverik miként könnyíti meg ezek leküzdését.

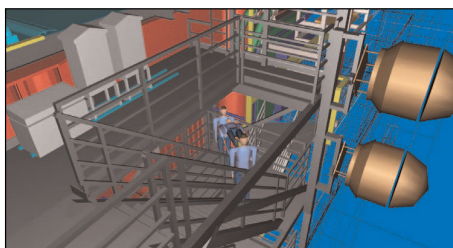
A térbeli grafikus tervezésbe nagyon könnyű beleszeretni. Mindenki szívesen körbejárja – ha csak virtuálisan is – jövőbeli házát. A mérnökök szívét is megdobogtatja a gondolat, hogy milyen egyszerű lenne a dolguk, ha a papíron látható épülettervet munkatársaiknak bemutatnák, esetleg véleményt is kérhetnének róla.



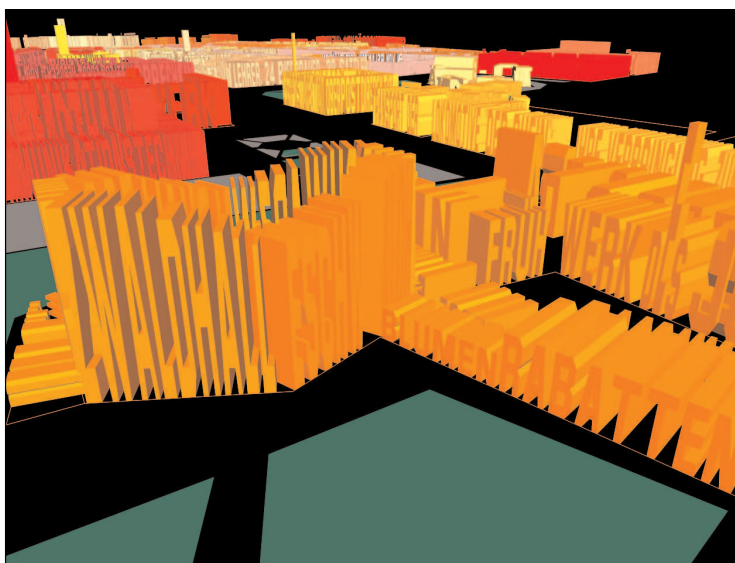
1. kép Tengeri rakodópart

Ezek csodálatos elképzelések, a valóság azonban jóval hétköznapibb. Mi is a gond? Egészen a közelmúltig úgy tűnt, az otthoni számítógépek és VR-eszközök teljesítménye a fő akadály. Az utóbbi néhány évben azonban a grafikus gyorsítókártyák zavarbaejtő sebességgel fejlődnek egyre több PC-tulajdonos gépében találjuk meg valamelyiket. A jelenlegi olcsó gyorsítókártyák teljesítménye a komoly munkáállomások teljesítményével vetekszik, és néha még 3D-s szemüveget is kapunk a kártyához.

Mára világossá vált: álmaink beteljesülését sokkal inkább a kevésbé hatékony programok, mintsem a számítógépek hátráltatják. A térbeli grafikai alkalmazások fejlesztése kemény munka. A lélegzetelállítóan élethű szörnyek, akikre



2. kép Drótvázás szerkezet



3. kép A Maverik segítségével készített Legible City II.

otthon vigyorogva lövöldöz mindenki, programozók és művészek több száz órai munkájának köszönhetik létüket. Egy-egy animációs film modellezése pedig még ennél is nagyságrendekkel több időt vesz igénybe. A Játékháború valósídejű modellezéséhez a jelenlegi legjobb PC-s grafikus kártyáknál tízezerszer nagyobb teljesítményre lenne szükség, főként az összetett grafikai elemek és a megvilágítás miatt. Ahogy azt a számítógépes játékok is bizonyítják, valós időben szintén lehet látványos jeleneteket készíteni. Ne felejtjük el azonban, hogy a legtöbb akciójátékban nem tehetünk meg akármit, például a falra akasztott képet sem vehetjük le, hiszen az igazából a fal része.

És ez még nem minden. Egy valóságű tengeri rakodópart vagy javítómedence leképezése könnyen elfogyaszthat fél gigabájtot is, és ezt az irtatlan mennyiségű adatot egy használható programnak másodpercenként legalább tízenötször fel kell dolgoznia (1. kép).

Vizsgáljuk meg közelebbről, hogy milyen műveleteket kell végeznünk egy ilyen összetett leképezéssel, ezáltal a Maverik előnyeit is értékelni tudjuk. A rakodópart esetében a megjelenítést a szükséges részletekre kell korlátoznunk. Ha egy sok szobából álló épületet belülről kell megjelenítenünk, a leképezés legnagyobb részét figyelmen kívül hagyhatjuk, hiszen egyszerre csak kis terület látszik. Hasonlóképpen járhatunk el egy városi utcakép leképezése során: a takarásban lévő épületekkel

nem kell törődnünk. Tehát a felhasználástól függően a leképezést sokféleképpen és gyorsan egyszerűsíthetjük. Ez a módszer azonban nem minden esetben alkalmazható. A rakodópart körül vitorlák, kötélzetek, csövek látszanak, tehát az egyszerűsítést más-képpen kell megoldanunk. Ha azonban a művelet sokáig

tart, kifuthatunk a rendelkezésre álló időből (ez általában a másodperc töredéke), tehát gyökeres változtatásra lesz szükség. A legegyszerűbb, ha a távolabbi részeket drótvázis szerzetként készítjük el. Ez nem mindig tetszetős, de ha így (2. kép) gyorsabban mozoghatunk a virtuális térben, akkor még mindig sokkal jobb választás, mint a kitarakás.

Egy másik ígéretes módszer, ha a drótvázis részeket állóképpel helyettesítjük, ezt a nézőponttól függően torzítva szép eredményt kaphatunk. Ez a megoldás jelenleg még kutatási szakaszban van. A virtuális térben különösen fontos, hogy a tárgyak ne „ugráljanak” a különféle megjelenítési módszerek közötti váltásokkor.

Ezen módszerek mindegyike jól használható külön-külön, de együttes alkalmazásuk hatékonysága nagymértékben függ az adott alkalmazástól. A képkockák tökéletes egymáshoz illesztése és a felhasználói beavatkozások figyelése nagy kihívást jelent. Jelenleg a feladat megoldására két módszer kínálkozik.

A program vázát megírhatjuk bármilyen korszerű programnyelven, a grafikai feladatokat pedig egy Mesa- vagy OpenGL-szerű grafikus könyvtárra bízuk. Így szinte bármit megtehetünk, a számítógép teljesítményét a legteljesebb mértékben kihasználhatjuk. Hátulütője a dolognak, hogy az egész túlságosan alacsony szintű – a munka ilyenkor leginkább arra hasonlít, mintha operációs rendszert kellene írunk gépi kódban. Emellett, mivel a módszer alkalmazásfüggő, szinte semmit nem vihetünk át az egyik fejlesztésből a másikba.

A másik megoldás, ha a virtuális világot valamilyen tervezőcsomaggal építjük fel, ez magas szintű szolgáltatásokat is lehetővé tesz, például egyszerre több felhasználó is barangolhat a térben. Természetesen ez a módszer sem tökéletes, hiszen előfordulhat, hogy a program lehetőségei meg sem közelítik az igényeinket. Egy általunk fejlesztett programnak nyilván saját szerkezete és algoritmus is van, ezeket az adott alkalmazásra kívánjuk kiélelni, tehát a számítógéppel éppen annyit szeretnénk elvégeztetni, amennyi szükséges, se többet, se kevesebbet. Visszatérve a példához: egy CAD-modellben a rakodópartot akár magas szintű kifejezésekkel is leírhatjuk: használhatunk létrákat, csöveket, szelepeket stb. Ha ezt az adathalmazt egy virtuális tértervező programban szeretnénk felhasználni, először a megfelelő formátumra kell alakítanunk azt. A közös nevező általában egy óriási poligonhalmaz, melynek beolvasása elkerülhetetlenül adat-és sebességvesztéssel jár.

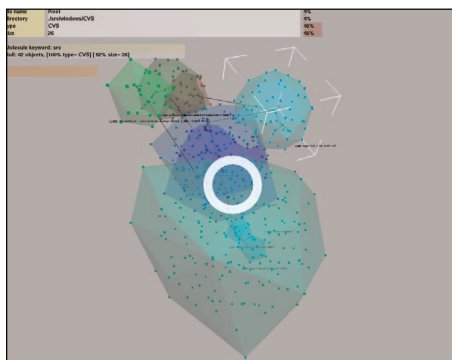
A második nehézség, hogy ha valaki a virtuális térben elmozdít egy tárgyat, ezt valahogyan a CAD-program számára is jeleznünk kell,



4. kép A játékprogramokban használatos leképezési módszerek



5. kép Megvilágítások használata



6. kép Különleges szerkezetek megjelenítése

melynek ezután módosítania kell az adatokat. Tehát a világot két változatban is tárolnunk kell: egyszer magában az alkalmazásban, másodsor pedig a térszerkesztőben.

Tervezőkor tehát hamar beleütközünk az alapvető kérdésbe: vagy túl sokat kell programoznunk, vagy a látvány minőségéből kell engednünk amiatt, hogy egyszerre két változatban is tárolnunk kell a teret.

Maverik: a középút

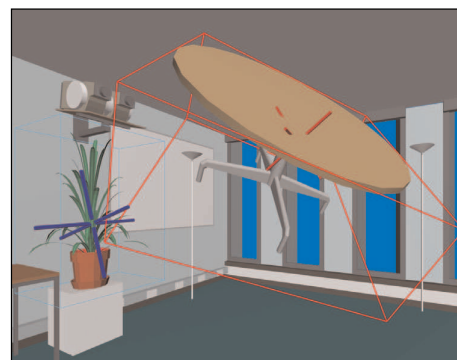
Vajon megtalálhatjuk-e az arany középút? Készíthető-e olyan keretalkalmazás, mellyel a virtuális terek készítését és kezelését megalkuvások nélkül, hatékonyan végezhetjük?

Mi bízunk ebben. Erőfeszítéseinket a Maverik léte igazolja, ugyanis ennek segítségével már néhány hónap múltán komoly, virtuális tereket kezelő alkalmazások készültek világszerte.

A Maverik egy C-alapú segédeszközgyűjtemény, amely egy 3D grafikai könyvtárat (Mesa, OpenGL) használ. Tulajdonképpen nem más, mint a könyvtárra épülő héj, mely megkönnyíti annak használatát. A Maverik egyik fő erőssége, hogy nincs saját belső adatszerkezete, hanem az X11 visszahívásaihoz hasonló módszerrel fér hozzá az alkalmazás adataihoz.

A program közli a Maverikkal, milyen tárgyakat kíván leképezni, majd

a Maverik térkezelő szerkezeteket (SMS, spatial management structure) hoz létre a különböző tárgy típusokhoz. Miközben a felhasználó a térben bolyong, a Maverik figyelemmel kíséri az SMS-eket, és a leképezéshez visszahívja az alkalmazást.



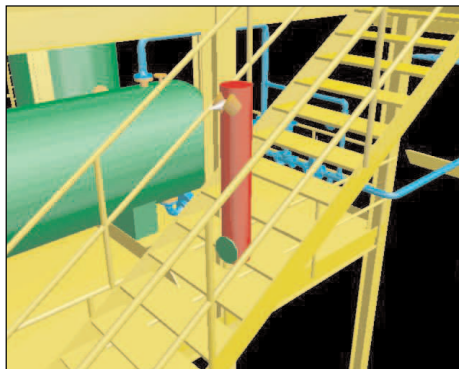
7. kép Hatékony vezérlés

Mit nyerünk ezzel?

Az adatokat az alkalmazás kezeli, tehát egyszerre csak egy adathalmazra kell figyelni. Ami még fontosabb: az alkalmazás minden erőforrását a leképezésre fordíthatja, nem kell a poligonrengeteg átfűlésével foglalkoznia. A térbeli szöveget egy hatékony adatformátumban néhány értékkel meghatározhatjuk – elég a szöveget, a betűtípust és a szöveg helyzetét tárolnunk. Ha azonban egy hagyományos térszerkesztőbe szeretnénk ezt bevinni, mindenképpen

a poligonok közvetítésére lenne szükségünk, így a rendszernek jóval több adatot kellene adott idő alatt feldolgoznia, ez pedig sebességcsökkenéssel, illetve minőségromlással jár.

A Maverik használatával ez a következőképpen néz ki. Az alkalmazásban bejegyezzük a „3D Szoveg” tárgytypust, majd ahogy a felhasználó a térben közlekedik, a Maverik az SMS-ek alapján észleli, hogy tárgy van a látómezőben. Kideríti, hogy szövegről van szó, s az alkalmazás megfelelő eljárásaival valós időben hozza létre a poligonokat.



8. kép Vezérlési eljárások

A program ezután elvégezheti a megfelelő minőségmeghatározást – így nagyon egyszerűen megvalósítható az, hogy gombnyomásra változzon a tárgyak részletezettsége. Ezáltal a Maverik tökéletes finomhangoló eszköz lehet a fejlesztők kezében.

A szövegpéldák Jeffrey Shaw (ZKM Germany) Legible City II. című

alkotásából származnak. Művében a fenti módszer alkalmazásával szavakból képezte le Amszterdam, Manhattan és Karlsruhe városát (3. kép). A 4. képen ugyanez a leképezés látható a játékprogramokban használatos módszerekkel.

Az előző példához visszatérve, a CAD-program saját adatszerkezete, melyben csövek, szelepek szerepelnek, hasonló módon kerül felhasználásra. Amikor az alkalmazás visszahívást kap egy torony leképezésére, eldöntheti, hogy az adott képkocka esetében mi lenne a legmegfelelőbb – le kell-e képezni a tornyot (mert az egy fontos tárgy a térben), avagy nem. Így az alkalmazás a feltételeknek megfelelően villámgyorsan módosíthatja a megjelenítést. Ugyanígy az összetett megvilágítási módszerekkel rendelkező (5. kép), illetve különleges szerkezeteket megjelenítő (6. kép) programok saját adatszerkezetet és módszereket használhatnak a valós idejű teljesítmény megőrzése céljából.

A Maverik segítségével készített műveket a program honlapján is megtalálhatjuk. Fontos megemlíteni, hogy a filmeket egy 450 MHz-es Pentium III PC-n, Voodoo2 kártyával mentették, így jól lemérhető, mire képes a Maverik egy elérhető gépen is.

Éz valami csalás?

A Maverik célja, hogy egymástól meglehetősen különböző alkalmazásokat egységes keretbe foglaljon, így megkönnyítve azokkal való munkát. Ez előnyös tulajdonság, de úgy tűnik, még mindig a programozóra hárul a legnehezebb feladat. Viszont e keretet nagyon egyszerűen bővíthetjük a megfelelő szolgáltatásokkal.

A csomagban számos kiegészítés is helyet kapott: geometriai alapelem, fájlbeolvasók, a vezérlést segítő szolgáltatások, térbeli

Kapcsolódó címek

A Maverik honlapján rengeteg animáció található, ezeket egy Linux-alapú PC-n, Voodoo2 grafikus kártyával készítették.
 ➔ <http://www.aig.cs.man.ac.uk/systems/Maverik>

A Maverik a Free Software Foundation honlapjáról is letölthető.
 ➔ <http://www.fsf.org/gnulist/production/maverik.html>

matematikai függvények, a 3D szemüveg és az egér kezelésére szolgáló programrészeket stb. (7. és 8. kép). Ezeket bárki tetszés szerint felhasználhatja, akár változtatás nélkül, de saját igényei szerint átalakítva is.

Miután megismerkedtünk a programcsomag szolgáltatásaival, három irányban indulhatunk el:

1. A készen kapott tárgyak és eljárások alkalmazása. Így használva a Maverik egy programozók számára készült térszerkesztővé válhat. Az oktató jellegű példákkal egy teljes rendszert építhetünk fel, ütközésérzékeléssel, valamint testreszabott vezérlési beállításokkal.
2. Saját tárgysztyálok meghatározása. A felhasználó által fejlesztett leképezési és más, ehhez kapcsolódó visszahívások használatával további sebességnövekedés érhető el. A csomag tartalmazza e témakör gyakorlófeladatait is, azonban jogi kérdések miatt a már említett rakodóparti példa nem kerülhetett be.
3. A Maverik rendszermag módosítása és bővítése. A leképezési és vezérlési visszahívások módosítása mellett szinte minden testre szabható: saját kitakarási, SMS és vezérlési módszerekkel is bővíthetjük a Maveriket. Ezenkívül az a legszebb, hogy ezek a fejlesztések nagyon egyszerűen, a már működő elemek veszélyeztetése nélkül beépíthetők a rendszerbe.

A Maverik igazi ellenőrzése, hogy milyen hatékonysággal használhatjuk fel saját bővítéseinket. Célunk nem a térszerkesztés és – kezelés programozásának leegyszerűsítése volt, hanem az, hogy a felhasználónak munka közben ne a rendszer nyűgjeivel kelljen foglalkoznia. A rendszer ezeddigi beváltotta a hozzá fűzött reményeket. A Maverik legutóbbi fejlesztései között egy erőtervezérlő-rendszer, valamint egy kiegészítő eljárás szerepel, mely a térben barangoló felhasználó előtt megjelenő tárgyakat ellenőrzi – hogy föl lehet-e rájuk mászni stb. Ezeket az elemeket egyelőre próbaváltozatban terjesztik, külön forráskódként.

Mire nem alkalmas a Maverik?

A Maverik nem tartalmaz mozgókép- és hangkezelő eljárásokat, ezeket magunknak kell elkészítenünk.

A Maverik egy egyfelhasználós kis rendszer (microkernel). Egyszerre több felhasználó jelenlétét, vagy egyszerre több tér használatát jelenleg nem támogatja. A hálózati alkalmazáshoz szükséges feltételeket (a terek összehangolása, frissítése) a felhasználónak kell megteremtenie. Ez természetesen óriási kihívást jelent; mi is dolgozunk rajta, fejlesztéseink eredménye az év második felében várható.

Mindenki véleményére, javaslatára számítunk, hiszen ezek adnak támpontot a további kutatásokhoz, és segítenek minket a programozók igényeinek felmérésében.

A visszajelzéseket a ➔ maverik@aig.cs.man.ac.uk címre várjuk.



Adrian West (ajw@cs.man.ac.uk) számítástechnikát oktat a manchesteri egyetemen. A virtuális tértervező rendszerek fejlesztésével foglalkozó Advanced Interfaces Research Group nevű társaság tagja.

Köszönetnyilvánítás

Az AIG köszöni az itt közzétett fejlesztéseket anyagilag támogató nagy-britanniai EPSRC csoport segítségét, illetve a kutatásban részt vevő szervezetek – a CadCentre Ltd., a Sharp Research és a Brown & Root támogatását.

3D grafika Linuxon

A játékfejlesztők legújabb fegyvertársai már Linux alatt is elérhetők.

Legszívesebben megváltozott a játékkörültek Linuxról alkotott véleménye az elmúlt évben nagymértékben megváltozott. Mára a Linux-alapú gépekre nem csak a hálózati játékok kiszolgálójaként tekintenek: egyre többen használják játékaikra is kedvenc operációs rendszerüket. A Linux beállítása már önmagában is nehéz feladat, a játékosok dolgát pedig a rengeteg rövidítés nehezíti meg. OpenGL, DRI, DGA: tudja valaki, mit jelentenek ezek? Írásunkban megkísérelünk egy kis fejtárgítást tartani e témakörben.

Mi is az OpenGL?

A körképet kezdjük az OpenGL-lel, ezzel biztosan találkozott már mindenki a játékokban vagy a csúcsmínőségű grafikai alkalmazásokban.

A Silicon Graphics (SGI) által fejlesztett, eredetileg IRIS GL néven futó OpenGL (Open Graphics Library) egy programkönyvtár, mely megkönnyíti a síkbeli és térbeli grafikai programozást. A csomag hamar a grafikai alkalmazás programozási felület (API-k) szabványává nőtte ki magát. A Linux Windows és Macintosh rendszereken egyaránt elérhető. Míg a programozók számára az OpenGL fontos munkaeszköz, addig a megrögzött játékosok leginkább az id Software Quake sorozatából ismerhetik. Az OpenGL az első rész óta a Quake leképezőmotorjának része. Itt fényesen bizonyította, hogy elképesztő játékokat is írhatunk segítségével. A szabvány egyik fő elve a nyitottság: az OpenGL minden operációs rendszertől és grafikus felületről független, innen ered a neve is (nyílt grafikus könyvtár).

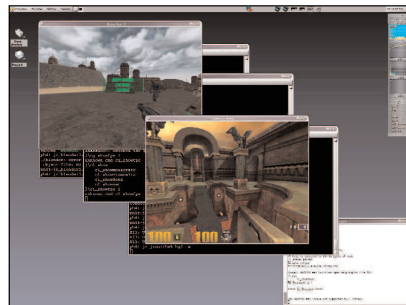
Ha az OpenGL-t bármilyen operációs rendszerben használni kívánjuk, valakinek meg kell írnia a szükséges eljárásokat, melyeket aztán az OpenGL program meghívhat. Ha azonban az OpenGL nevet szeretnénk ráakasztani a programunkra, a név használatának jogát is meg kell vásárolnunk az SGI-től vagy a Microsofttól. A megvásárolható csomaggal egy SI (Sample Implementation) nevű mintaalkalmazást kapunk, C nyelven írt forráskódjával együtt. Ennek alapján egy felkészült programozó elkészítheti saját felületén futó OpenGL könyvtárát. Mivel az OpenGL nyitott szabvány, az SI beszerzésére nincs is szükség. Hasonlóan gondolkodtak a Mesa fejlesztői is, ugyanis terméküket

teljesen az alapokról indulva, az OpenGL szolgáltatásait lemásolva készítették: megírásához egyetlen SGI-től kapott programsort sem használtak fel. Az OpenGL védjegy használatához természetesen ekkor is szükség van az SGI-vel kötött szerződésre, s az új változatot az OpenGL szabványra felügyelő szakmai testületnek, az ARB-nek is be kell mutatni. Az ARB (Architecture Research Board) eredetileg számítógépeket gyártó cégekből és az SGI-ből állt, mára azonban néhány olyan cég is tagja, amelyek játékokhoz szükséges részcégeket gyártanak. Érdekes megjegyeznünk, hogy az OpenGL-t az SGI fejlesztette ugyan ki, de a felügyelő bizottságban már csupán egy tag a többi között.

Az SGI nemrégiben adta ki az SI-t, ennek segítségével bármilyen számítógépre elkészíthető a grafikus könyvtár. E változat szinten teljesen megegyezik a grafikus kártyákat fejlesztő cégek számára átadott forráskóddal. Bár az SI állítólag nyílt forráskódú, ez nem teljesen igaz. Cikkünk megírásakor az SI szerződése még jó néhány olyan elemet tartalmazott, amelyek megakadályozták, hogy a csomag teljesen nyíltá váljon. Sokak számára az OpenGL egyet jelent a grafikai gyorsítással. Bár tényleg ragyogóan kihasználja a komolyabb kártyákat, az OpenGL-re írt programok futtatásához nincsen szükség gyorsító kártyára. Ugyanis ezek a programok a kártyák nélkül is ugyanazt a képet rajzolják ki, természetesen jóval lassabban. Ha az adott gépen elérhető a grafikai gyorsítás (tehát egy 3D kártya és a hozzá tartozó meghajtó), akkor az OpenGL alkalmazások gyorsan és simán futnak, hiszen a szükséges számításokat ekkor a kártya végzi el. Az OpenGL szolgáltatásait nem minden grafikus kártya támogatja százszázalékosan. Ha a grafikus könyvtár egy, a kártya által nem támogatott művelet észlel, akkor azt programból, a központi processzort terhelve végzi el. Például a Transform & Lighting (T&L) nevű szolgáltatást a legtöbb közismert gyorsító kártya sem ismeri, ilyenkor az OpenGL háromszögeléssel és különböző eljárásokkal oldja meg a helyzetet. Születése óta az OpenGL-en számos bővítést hajtottak végre, jelenleg az 1.2 változatnál tart. A Linux alatt a leggyakrabban használt OpenGL-változat a *Brian Paul* vezetésével kifejlesztett Mesa.

Mesa: a nyílt forráskódú OpenGL

A Mesa az OpenGL szerződés nélkül készített változata, ezt Linux, Windows és Macintosh operációs rendszerben egyaránt használhatjuk. 1995-ös indulása óta egy kutatócsoport fejlesztí, melyet Brian Paul vezet. A Mesa ingyenes és nyílt forráskódú, azaz bárki, akinek kedve van hozzá, átültetheti egy másik operációs rendszerre. A legfontosabb, hogy a Mesa használatával



A Heavy Gear II és a Quake III egyszerre fut

bármilyen OpenGL alkalmazást futtathatunk. A Mesával az eszközökben rejlő gyorsítás is kihasználható, például egy 3dfx kártya tulajdonosa letöltheti és telepítheti a Glide SDK-t a 3dfx honlapjáról. A Mesa újrafordításakor érzékeli a telepített Glide fejlécállományokat és könyvtárakat, a 3dfx kártya ezután használhatja az OpenGL gyorsítási lehetőségeit (a Glide 2.x segítségével). A Mesát is számtalanszor bővítették már, akárcsak az OpenGL-t, így a Mesa 2.x az OpenGL 1.1 szabványt, a Mesa 3.x pedig az OpenGL 1.2-t, ezenkívül a GLUT és a GLU szabványokat is támogatja. Van tehát az OpenGL, melynek egy nyílt változata a Mesa. Most nézzük meg, hogy mi kapcsolja össze az OpenGL-t az X Windows rendszerrel.

GLX: az OpenGL és az X Windows

Az OpenGL géptől, operációs rendszertől és grafikus felületről is független. Használatához tehát valamilyen módon egy grafikus felülethez kell kapcsolnunk. Ez a kapcsolat lehetővé teszi az olyan feladatok elvégzését, mint például egy ablak helyének megtalálása vagy a bemenő adatok feldolgozása. A Linux- és Unix-rendszerek esetében ezt a GLX nevű könyvtár teszi lehetővé (Windows alatt a WGL használatos).

Vagy inkább úgy fogalmaznánk, a GLX segítségével megoldható, hogy az OpenGL egy tetszőleges X képernyőt használjon. A Linuxhoz használatos GLX az SGI által 1999. februárjában kiadott forráskódon alapul. Néhányan talán már a Utah-GLX kifejezésbe is belebotlottak, így most essék szó erről kicsit bővebben.

Utah-GLX

A Utah-GLX célja, hogy az XFree86 3.3.x-et, az OpenGL-t és az újabb grafikus kártyákat (Matrox G400/G200, ATI Rage Pro/Rage 128) együtt használhassuk. A Utah-GLX meghajtója a gyorsítást legtöbbször közvetett leképezéssel éri el, azonban ez a módszer – ahogy a későbbiekben látni fogjuk – bizonyos esetekben csökkentheti a teljesítményt. A Utah-GLX-nek köszönhetően valósították meg a grafikus gyorsítást a PPC processzoros Linux gépeken, valamint a laptopokon egyaránt. Mindkét esetben az ATI Rage Pro grafikus kártyát használták.

Az XFree86 4.0 megjelenésével remélhetjük, hogy a Utah-GLX-et hamarosan közvetlen leképezéshez (DRI) is felhasználhatjuk. A közvetlen és közvetett leképezés közötti különbségek tisztázására most ismét következzenek egy kis magyarázat.

Közvetett és közvetlen leképezés

A két módszer között az adatfeldolgozás menetében van különbség. Minél kevesebbszer kell a grafikus adatokhoz nyúlni, annál gyorsabban jelenhet meg a kép.

A közvetett leképezésnél az adatok először az alkalmazásból az X kiszolgálóhoz kerülnek, mely ezután a kártyához továbbítja azokat. Ez természetesen idővesztéssel jár, hiszen az X kiszolgálóra küldés előtt az adatokat a megfelelő formátumra kell alakítani, majd az X kiszolgálónak is átalakítást kell végeznie, mielőtt a kép a monitorra kerülne. Síkbeli grafikák esetében ez a sebességsökkenés elhanyagolható, de a manapság gya-

kori 3D-s alkalmazások jelentősen lelassulnak a közvetett leképezés következtében. A közvetlen leképezés az alkalmazás számára lehetővé teszi a grafikus kártya közvetlen elérését, ennek köszönhetően az X kiszolgálónak ilyenkor sokkal kevesebb munkát kell végeznie. E módszert – mely a DRI (Direct Rendering Infrastructure) néven ismeretes – a Precision Insight fejlesztette ki, és az XFree86-ban is helyet kapott.

Most végre rátérhetünk a titokzatos DRI tárgyalására: megvizsgáljuk, milyen előnyökkel jár e módszer használata a Linux-felhasználók számára.

DRI: biztonság, megbízhatóság, gyorsaság

A DRI lehetővé teszi, hogy az alkalmazás biztonságosan és hatékonyan érje el a kártyát. Az Utah-GLX-szel ellentétben használatához nincs szükség magas szintű felhasználói jogokra: a munka nagy részét maga az alkalmazás, és nem az X kiszolgáló végzi.

A DRI három része: egy különleges X kiszolgáló egy, a közvetlen leképezést végrehajtó ügyfél és egy eszközmeghajtó, mely a rendszermagba épül be. A leképezőmotor további elemekből épül fel: valamely OpenGL-változatból (általában Mesa), a GLX-ből és Linux-közeli, de gépfüggetlen meghajtókönyvtárakból. A rendszermagba épülő meghajtó közvetlen memóriáhozáférést (DMA) enged a grafikus kártya számára, s úgy készítették el, hogy a továbbiakban a rendszermag módosítására csak nagyon különleges esetben legyen szükség. Egyszóval: a DRI a grafikus gyorsítás első működőképes megvalósítása. Használatához nincs szükség különleges felhasználói jogokra, biztonsága pedig jóval meghaladja az eddigi megoldásoknál tapasztalt szintet. Mindemellett a DRI az XFree86 része, így számos próbasorozatban ellenőrizték működését. Ezenkívül magában hordozza az XFree86-nál megszokott üzembiztonságot is. A DRI lehetővé teszi, hogy nagy erőforrás-igényű alkalmazásokat (magas szintű grafikai tervezőrendszerek, 3D-s játékok) futtassunk Linux alatt.

Egyéb finomságok

A DRI, az OpenGL, valamint a Mesa kifejezéseken kívül olvasóink bizonyára más titokzatos rövidítésekkel is találkoztak eddigi linuxos pályafutásuk során. Ne hagyjuk ki ezeket sem!

- *GLUT* – A GLUT alapját a *Mark Kilgard* által kifejlesztett aux könyvtár képezi, mely eredetileg az általa írt *OpenGL Programming for the X Window System* című könyv példáinak futtatásához készült.
- *DGA* – A DGA (Direct Graphics Architecture) nem gyorsítás, csupán az X kiszolgáló egyszerű bővítése, mely lehe-

tővé teszi a képernyőmemória közvetlen elérését az alkalmazások számára. Használatához magas szintű jogosultságokra van szükség. Az XFree86 4.0 a DGA 2.0-s változatát tartalmazza, ez már elérhetővé tesz némi gépszintű gyorsítást, de csak a 2D-s grafikai alkalmazások esetében. A DGA ezenkívül az egér és a billentyűzet jeleinek közvetlen olvasásában is az alkalmazás segítségére lehet. Alap esetben ugyanis ezek a jelek először az X kiszolgálóhoz érkezik, s csak innen kerülnek továbbításra a program felé.

- *VidMode* – A VidMode bővítés segítségével az alkalmazás bármikor lekérdezheti az X kiszolgáló által támogatott képernyőmódokat (frekvencia, felbontás, színmélység stb.), és ezek között váltani is tud. Általában olyan játékok használják, amelyek futásához egy adott képernyőmódra van szükség, az X kiszolgáló éppen használt módjától függetlenül.
- *Glade* – A Glade egy, a 3dfx által, saját Voodoo sorozatú kártyáikhoz kifejlesztett, teljes képernyős, gépszintű gyorsítást nyújtó API. A régebbi, gyorsítást használó játékok alkalmazták, ilyen például a Tomb Raider is. Az OpenGL általános sikerével a Glade jelentősége egyre csökken, ennek hatására a 3dfx nemrégiben nyíltta tette a forráskódot.

Akkor foglalkozzunk össze...

Az OpenGL tehát egy grafikus könyvtár, olyan eljárásgyűjtemény, mely a térbeli grafikák megjelenítését segíti elő. Linux alatt a Mesa teszi lehetővé az OpenGL alkalmazások futtatását, ezeket pedig a GLX hangolja össze az X kiszolgálóval. A még mindig XFree86 3.3.x-szel dolgozó felhasználók a Utah-GLX használatával helyettesíthetik az XFree86 4.0-t és a DRI-t. Az XFree86 4.0 azonban jóval nagyobb teljesítményt és megbízhatóságot jelent.

Reméljük, ezen összefoglaló cikk segített eloszlani a Linux 3D gyorsítással kapcsolatos sötét foltokat. Jó játékos és eredményes munkát mindenkinek!

Daryl Strauss 15 éve dolgozik Unix-rendszerekkel, 1995 óta pedig a Linuxszal. Ezen idő alatt különleges filmes hatások készítésével és nyílt forráskódú 3D-alkalmazásokkal foglalkozott. Jelenlegi munkaadója a VA Linux Systems.

Matt Matthews az NCSU PhD ösztöndíjas hallgatója. 1999 nyara óta tartozik a Linux-felhasználók népes táborába. Szívesen próbál ki új játékokat és grafikus kártyákat; az eredmények a <http://www.linuxgames.com> honlapon jelennek meg.

Kapcsolódó címek

XFree86:

➔ <http://www.xfree86.org/>

Mesa:

➔ <http://mesa3d.sourceforge.net/>

Precision Insight:

➔ <http://www.precisioninsight.com/>

DRI Development:

➔ <http://dri.sourceforge.net/>

Utah-GLX:

➔ <http://utah-glx.sourceforge.net/>

SGI Open Source:

➔ <http://oss.sgi.com/>

RAD a Linux alatt

Az Omnis Studio a Javához hasonló módon közelíti meg a programfejlesztést.

A linuxos programok fejlesztésére használt gcc-fordítóprogramot a teljes Unix-világ szabványként ismeri el. Gyors, érthető és szabványos C-re fordít (vagy C++-ra a g++ fordítóprogram esetében). Akik azonban már írtak nagy make-fájlokat, tudják, hogy ez önmagában nem állja meg a helyét teljes fejlesztői környezetként. A többprogramozós feladatoknál nagy segítséget jelentenek az olyan komoly eszközök, mint például a Cygnus cég Code Fusion és Source Navigator rendszerei, ezek egyre inkább megkönnyítik nagy projektek fejlesztését.

Vizont a legtöbb hivatásos programfejlesztő a RAD (Rapid Application Development) eszközökre támaszkodik az alkalmazások tervezése, modellezése és kódolása közben. Az ilyen alkalmazások nagy része az adatbázisok körül forog, hiszen sokukat az egyes szervezetek saját használatra vagy a különböző piaci rétegeket összefogó tanácsadók számára készítették. Ilyenek például a számlázási vagy adatbázis-elérési rendszerek, a fogyasztói tanácsadói rendszerek, illetve az olyan eszközök, amelyek már létező webes szolgáltatásokat olvasztanak össze, és így az alkalmazottak egy böngésző segítségével kereshetnek bizonyos adatokat. A belső fejlesztők más jellegű gondokkal küzdenek: a nagyon szűkös határidők, az illeszkedés a már létező rendszerhez, valamint az, hogy az adott eszköz fusson a szervezetben megtalálható többféle felületen is.

Egy komolyabb feladat elkészítése a következőképpen történhet:

1. Tervezés, majd a fejlesztés munkamenetének kidolgozása.
2. Modell (ha a dolgok lassan haladnak), vagy a bemutatópéldány (ha a dolgok gyorsan haladnak) elkészítése.
3. A program fejlesztése.
4. A program ellenőrzése először házon belül, majd a felhasználás helyén.
5. A program bevezetése és a szükséges javítások elvégzése.

Az ehhez hasonló tervek megvalósításához a Java megfelelőnek tűnik, hiszen rendszerek között átvihető nyelv, a Weben is működik, és úgy néz ki, hogy mindenki elfogadja. De ahogy több félbehagyott terv fejlesztője is elmesélhetné, a Java nem mindig jelent teljes körű megoldást. Az ilyen belső tervek megvalósítására gyakran használják még a Visual Basicet.

Ezekkel a belső tervezetekkel az a gond, hogy a bemutatópéldány elkészítése és a fejlesztés olyan sok időt és pénzt emészt fel, hogy a terv kivitelezhetetlenné vagy túl költségessé válhat. A fő nehézség, hogy miként lesz a tervből bemutatópéldány, majd ebből végleges program. Ami a többi gondot illeti, azt a megfelelő eszközök vagy a tehetség hiánya okozhatja, esetleg az, hogy több felületen működőre kell tervezni a programot (esetleg „webes” programot kell készíteni, ami szintén kihívás).

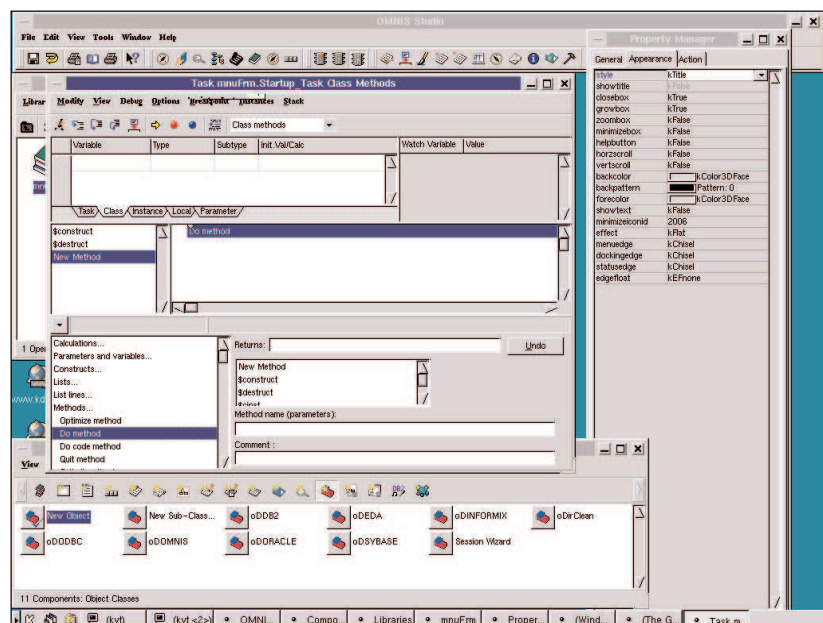
A keresett megoldást a RAD-eszközök nyújtják. Ezek ötvözik a grafikus kezelőfelület hatékonyságát a szükséges adatbázis-kezelő szolgáltatásokkal és egy olyan eszközzel, ami az alkalmazás kódolásában, hibajavításában és üzemeltetésében segít.

Az Omnis Studio

Míg néhány csúcsmínőségű eszköz Linuxon is elérhető már egy ideje – közülük jó pár tízezer, vagy annál is több dollárért – azok, akiknek RAD-eszközökre lenne szükségük, figyelembe sem veszik a Linuxot (a Borland ígéretes Kylix nevű eszköze sem jelent még meg).

A linuxos programfejlesztésben új játékos az Omnis Software cég által kiadott Omnis Studio. Annak ellenére, hogy ez a csapat és terméke már mintegy húsz éve a piacon van, csak a múlt év folyamán kezdtek linuxos termékeket forgalmazni. Elmondhatjuk, sikerült a magas színvonalú termékeket szerencsésen átültetni Linuxra. Az Omnis Studio a Javához hasonlóan gondolkodik, így hatékony grafikus környezetben, több felületen működő programot hozhatunk létre.

Az Omnis olyan futtatómotort használ, amellyel programokat fejleszteni és végrehajtani egyaránt lehet. Ha szerződésünk engedi a módosítást, akkor új programokat is létrehozhatunk, egyébként csak már létezőket futtathatunk. Meglepő módon a futtatómotor mindössze egy megabájtos (leszámítva az esetleg szükséges programozói könyvtárakat). Meglepő, hogy az alkalmazások ennek ellenére is elfogadható sebességgel futnak, főleg, ha figyelembe vesszük, hogy a rendszer futás közben értelmezi a programot. Az Omnis Studio programjainak futtatásához szükséges alkalmazások már léteznek Linux, Windows és Macintosh alatt is. Hogy ezt miért érdemes külön kiemelni? Képzelnék el, hogy fejlesztünk egy programot Linux alatt, majd ezt a programfájlt Windows alatt módosítjuk (mert például fejlesztői szerződésünk van a Studióra), majd a frissen módosított programot egy Macintoshon futtatjuk. Úgy kell elképzelni a programokat, mint egy parancsállományt, amit bármely gépen szerkeszteni lehet. Gyakorlott Omnis és egyéb eszközök (Java vagy Visual Basic) felhasználóitól hallottam, hogy a Visual Basic alatt eltöltött fejlesztési idő körülbelül 20 százalékára van szükség az Studióban megtalálható



Az Omnis Studio kezelőfelülete

ötleteknek köszönhetően. Ilyen segítség például a teljes projektben megtalálható változónevek automatikus kiegészítése gépelés közben, valamint az olyan egyfájlos munkák, amelyekben összegyűjtjük az összes fontos adatot.

Webes varázslók

Az Omnis úgy véli, hogy legnagyobb vetélytársa a Visual Basic. A jó hír az, hogy a Studióval a belső fejlesztési terveket Linux alatt windowsos munkaállomások, vagy költséges unixos RAD-eszközök nélkül fejleszthetjük, és tetszőleges felületen futtathatjuk, akár egy böngésző segítségével. A képeken a Studio kezelőfelületét láthatjuk. Nehéz csupán egy-két kép segítségével bemutatni a rendszer számtalan lehetőségét, a képeken azért látható néhány eszközkészlet, tulajdonság, megjegyzésmező és eljárászerkesztő. A gombok, a listák, a menük, a szövegek és egyéb szokványos eszközök a gyors elérhetőség céljából mind egy kupacban találhatóak. A notation nevű parancsnyelv segítségével lehet kódot készíteni az objektumokhoz, de ahogy én tapasztaltam, a Visual Basichez képest sokkal kevesebb kódolás szükséges. A termék teljesen objektumközpontú, objektumöröklődéssel és hatékony osztálykönyvtárakkal rendelkezik.

A Studio két területen nyújt kiemelten nagy segítséget: az adatbázis-hozzáférés és a webalapú fejlesztés területén. Ezen elemek segítségével könnyedén kezelhetünk adatbázisokat, jeleníthetünk meg, módosíthatunk vagy törölhetünk sorokat, mezőket. A rendszer Oracle, Sybase, Microsoft SQL, IBM DB2, vagy bármilyen ODBC-szabványú adatbázis-kiszolgálót támogat. A részletekkel nem kell törődnünk, mi csak megadjuk, hogy milyen formában jelenjenek meg a mezők, és lekódoljuk, hogy mi történjen azokkal az adatokkal, amelyekkel a végfelhasználó dolgozik, a többi az Omnis Studio végzi.

Mivel a fejlesztői felület maga a futtatómotor, az adatbázis-eszközt arra is használhatjuk, hogy az adatbázisokat valós időben kezeljük, akár fejlesztés közben. A sémákat (a mezők elrendezését) átmásolhatjuk az adattáblák között, közvetlenül a programba, vagy akár különböző adatbázis-kiszolgálókon levő adattáblákba is (akár különböző típusok esetén).

Az Omnis Studio olyan bővítményt is tartalmaz, melynek segítségével az általunk írt alkalmazások böngészőből is futtathatók. Alkalmazásunk „webesítéséhez” a programhoz csupán néhány elemet kell hozzáadni, viszont szükséges, hogy a webkiszolgálón telepítve legyen a futtatómotor, valamint minden ügyfél böngészőjébe be kell illeszteni a bővítményt. A lehetőség azonban, hogy a Linuxot alkalmazáskiszolgálóvá varázsoljuk, talán elegendő indokkal szolgál arra, hogy kísérletet tegyünk a Studióval.

Mintapéldány? Ugyan milyen mintapéldány?

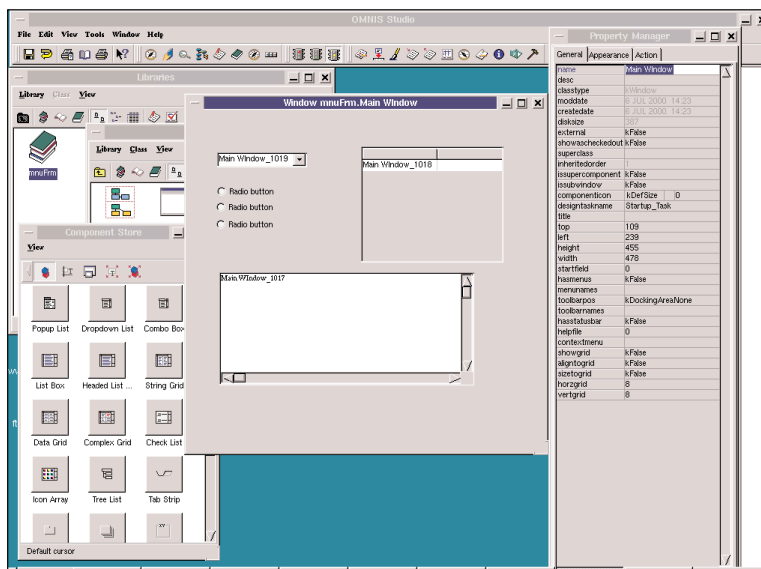
A Studio használatának legnagyobb előnye szerintem az, hogy a mintapéldány – amely az alkalmazásvarázsló, a szabványos eszközkészlet és néhány parancsfájl segítségével elkészíthető – már egy működő program, amit ki lehet próbálni. Ha a bemutatópéldányom még egy kis javításra szorul ahhoz, hogy megfeleljen a tervezetnek vagy az ügyfél kívánásainak, akkor azt azonnal megtehetem, és az új változatot akár a Weben keresztül is eljuttathatom a felhasználókhöz. Az automatikus frissítés lehetőségére nem minden fejlesztőnek van szüksége, de azt mindenki méltányolja, ha a minta-

példányból egy kis csiszolással működő program készíthető.

Az Omnis Studio teljes változata 149 dollárba kerül. Ebben nincs benne semmilyen nyomtatott kézikönyv, de a teljes leírás (több ezer oldal) megtalálható a CD-n. A nyomtatott anyag külön beszerezhető, ha szükség van rá. Az Omnis Studio weboldaláról számos dolgot tölthetünk le ingyenesen, beleértve további adatbázis-támogató modulokat, az Omnis Studio próbaváltozatát és a webes bővítményt (így saját böngészőnkkel is futtathatjuk a weboldalon elérhető, már kész alkalmazásokat). Az Omnis Studio 2.4 korlátozott változata megtalálható a Caldera OpenLinux 2.4 eDesktop termékben is.

A csapdák

Na jó, nagyon tetszik az Omnis Studio, de néhány figyelmeztetést is. Először is, sokáig tart a program használatának elsajátítása, talán a „könnyű megérteni, de bonyolult alkalmazni” leírás illik rá a legjobban – akár a Linuxra. A Studio saját parancsnyelvét, valamint rengeteg saját tervezésű varázslót és eszközt használ. Ezek mind hasonlítanak a Visual Basichez, de a Studiónak saját logikája van. Ha időt szánunk megtanulására, akkor lenyűgöző dolgokat tudunk



A RAD eszközöknél megszokott hatékony környezetet használhatjuk

készíteni, de ha csupán babrálunk vele, akkor még csak felszínesen sem fogjuk megismerni képességeit. Az Omnis komoly erőt fektet a termék használatának könnyebbé tételébe és rutinos fejlesztők óriási csapata érhető el a neten, akik szívesen segítenek. Emellett a CD-n található leírás is igen részletes és érthető, tele világos példákkal. A másik nagy csapda, hogy ez kereskedelmi program. A Fortune 500 cégek nagy unixos és windowsos alkalmazásainak örökségét viszi tovább. Bár az Omnis jelentősen csökkentette árait, amikor belépett a linuxos piacra, a termék elsősorban közép- és nagyvállalatok számára szánták. Ebből következően minden egyes ügyfél gépére meg kell vásárolni a futtatómotort. Ez csekély összeg, mindössze tízdolláros nagyságrendű, de ezért nem nevezhetjük nyílt forráskódnak az Omnis Studiót. Viszont ez még mindig az ár, amit szívesen kifizetünk, ha nevelésesen rövid idő alatt kell hatékony programot írni.

☞ <http://www.omnis-software.com/products/studio/index.html>

Nicholas Wells (nick.wells@lineo.com) a beágyazott Linuxokat forgalmazó Lineo Inc.-nél a szaknácásadó részleg ügyvezetője. Közel egy tucat könyve jelent meg a Linuxról, a KDE-ről és a Webről.

© Kiskapu Kft. Minden jog fenntartva

Jellemzők

Gyártó: Omnis Software, Inc.
Levél cím: us_sales@omnis.net
Ár: 149 dollár

TimeSys Linux/RT

A Linux/RT egy modul segítségével futtatja a valós idejű programokat.

A termék számos felhasználási területen sokféle igényt elégít ki, például az ipari automatizálás, a folyamatirányítás, a távközlés, valamint a webkiszolgálók terén. Sőt, még a légitársaságok is használják. A fejlesztők ígéretének megfelelően a Linux/RT a Linux megbízhatóságát megtartva képes valós idejű alkalmazások futtatására. Ahhoz, hogy ez a felépítés valóra válhasson, a TimeSys kibővítette saját Linux-változatát néhány rendszerhívással, valamint egy valós idejű erőforrás-rendszermagot (real-time resource kernel, a továbbiakban RK) használ. Az RK betölthető magmodulként (loadable kernel module, röviden LKM) illeszkedik a Linux rendszermaghoz. Ennek a felépítésnek az a hatalmas előnye, hogy az RK felhasználhatja az összes létező linuxos meghajtóprogramot. A már létező alkalmazások továbbra is a Linux rendszermaggal érintkeznek, a valós idejűek pedig a valós idejű magmodullal. Így az egész rendszer nem fog összeomlani attól, hogy egy valós idejű alkalmazás összeomlik vagy leáll. Ez azt is jelenti, hogy a korábbi (hagyományos) linuxos alkalmazások képesek együtt futni valós idejű társaikkal. Az RK modul – hasonlóan a többi a LKM-hez – bármikor betölthető a memóriába, és el is távolítható onnan.

A Linux/RT három kiadásban érhető el: Standard, Deluxe és Professional. A Professional tartalmazza a Linux/RT mellett a TimeTrace programot és – ahogy azt a weboldalukon is hirdetik – a TimeWiz próbaváltozatát. Az én változatomban nem volt TimeWiz, így erről nem szerezhettem gyakorlati tapasztalatokat. A TimeSys Linux/RT-je két valós idejű környezetet is tartalmaz. Az egyik a Carnegie Mellon Egyetem Linux-alapú erőforrás-rendszermagjára (Linux/RK) épül, és a Robust Embedded (RED), illetve egy Linux eseménynaplózót foglal magában. A másik környezet a korábbi rendszerekkel való együttműködést teszi lehetővé. Ez a Real Time Applications Interface (RTAI), vagyis a valós idejű alkalmazás illesztőfelülete. Azt azonban meg kell említenünk, hogy ezek közül egyszerre csak az egyik lehet betöltve. Az 1. kép az egyik Linux/RT-hez mellékelte valós idejű mintaprogramot ábrázolja.

A modell szerint az erőforrásmag képes a processzoridő, a hálózati sávszélesség, a háttértár-sávszélesség és memória bizonyos részének fenntartására. Ezeket a fenntartott erőforrásokat csoportokba foglalták, és így rendelik hozzá a folyamatokhoz. A Linux/RT-nek számtalan szolgáltatása van. Többek között 256 fontosságú szintű ütemezővel, processzoridő-foglalással, nagyfelbontású órákkal és időzítőkkel rendelkezik, valamint rendszeres valós idejű feladatokkal és osztott memóriával bír a valós idejű feladatokhoz. A mintaként kapott Linux/RT leírása hálózati és lemezes sávszélesség foglálásáról nem tett említést. A Linux/RT segédprogramjaival azonban erőforráskészleteket rendelhetünk bármely futó folyamathoz. Használhatjuk továbbá az úgynevezett szolgáltatásminőség-biztosítás (Quality of Service, QoS) lehetőséget is, és az előbb említett segédprogramokkal mind a hagyományos, mind a valós idejű linuxos alkalmazásokat különböző QoS szintekre sorolhatjuk.

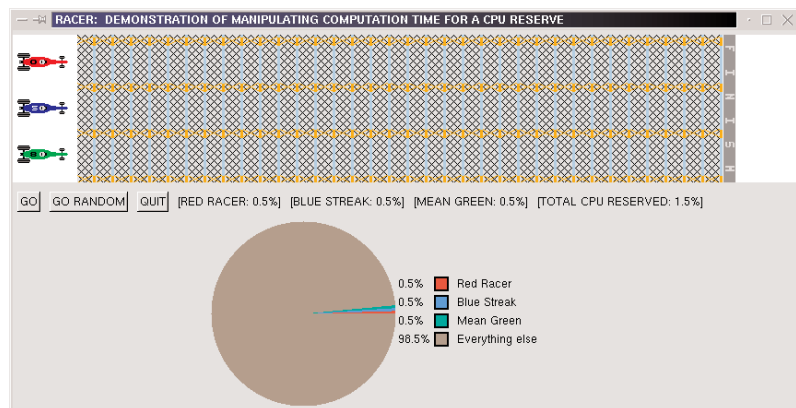
A Linux/RT TimeTrace programjával lehetőségünk nyílik arra, hogy a Linux/RT-vel felszerelt gépen futó

valós idejű programok és alkalmazások futási tulajdonságait mérjük és megjelenítsük. A 2. képen példát láthatunk a TimeTrace kezelőfelületére. A TimeTrace a saját felületén csücsül, és egy grafikus felületen keresztül enged hozzáférést a figyelemmel kísért számítógéphez. Megvizsgálhatjuk vele az ütemezést, a környezetváltásokat, a rendszerhívásokat, valamint a felhasználói üzeneteket. A TimeTrace segítségével megtudható a valós idejű feladatok leghosszabb és átlagos végrehajtási ideje. A TimeTrace nagyobbik változata képes több Linux/RT-t futtató gép megfigyelésére is.

A TimeWiz olyan modellező, elemző és szimulációs eszközt ad, melyek segítségével valós idejű programjaink viselkedését vizsgálhatjuk. A fejlesztők szerint segítségével elemezhetjük a „legrosszabb eset forgatókönyvét”, utánozhatunk átlagos időzítésű esetet, modellezhetünk átviteli teljesítményt, program-, illetve géptípusokat, ezenkívül programjelentésekkel is ellát bennünket. A TimeTrace-hez hasonlóan a TimeWiz is Wintel felületen fut.

Milyen nehéz telepíteni?

A Linux/RT alapja egy Debian-változat. Új telepítés esetén a Debiannál megszokott módszerrel kell eljárunk, ám a kézikönyv szerint nemcsak debianos változatokon használhatunk Linux/RT-t, hanem mondjuk a Red Hat-en, a SuSE-n és a Mandrake-en is. Én például Calderára telepítettem a nem debianos parancsfájllal. Ez szokásos és egyedi telepítést ajánlott fel. Az egyedi lehetőséget választva módunkban áll megadni a célkönyvtárat, és a Linux/RT-t a LILO-ba is bejegyeztethetjük. Én az alapértelmezett választottam, ezután a telepítés gond nélkül zajlott le. A Caldera 2.4 pillanatnyilag GRUB-ot használ rendszerbetöltőként, ezt azonban a Linux/RT nem ismerte fel, mert a LILO-t kereste. Hozzáadtam kézzel a GRUB-hoz a Linux/RT rendszermagot, majd miután újraindítottam a gépet azt tapasztaltam, hogy a TimeSys-féle mag mindent felismert a gépemben: a SCSI kártyát, a SCSI-s eszközöket és az Ethernet kártyámat. A TimeTrace for Linux/RT telepítése kétlépéses. Először magmodulokat kell telepíteni egy linuxos gépre, majd a TimeTrace-t egy Win NT/95-ösre. Ezután a windowsos gépre telepített két felhasználói



1. kép A Linux/RT egyik valós idejű mintaprogramja

felületen futó adatgyűjtő programot (a viewrk-t és a viewrtai-t) kell átmásolnunk a linux/RT-s gép(ek)re. A TimeTrace telepítése a hagyományos windowsos módszerrel történik, és zökkenőmentesen zajlik, ha sikerül beolvasatnunk a CD-t. Én úgy jártam, hogy az egyébként minden egyebet olvasó CD-meghajtóm nem szerette beolvasni ezt a lemezt, csak a DVD-m. A leírás nem tesz különösebb javaslatot az adatgyűjtő programok telepítési helyére vonatkozóan, így én például a /usr/local/bin könyvtárba raktam, ahová a többi RK segédprogram is került.

Milyen nehéz a használata?

A Linux/RT moduljai, az RTAI és a TimeTrace is jó pár parancs kiadásával vehető csak használatba. A Linux/RT esetében magmodulokat kell betöltenünk. A TimeTrace használata pedig feltételezi egy valósidejű alkalmazás és a megfelelő adatgyűjtő programok futását.

A Linux/RK segédprogram-csomag parancsaival a Linux/RT erőforrásokat a hagyományos programokhoz rendelhetjük. Természetesen ezeknek a programoknak semmiféle valósidejű programozási elemet nem kell tartalmazniuk. Tettem is egy próbát egy Netscape böngészővel. Csatoltam hozzá egy körülbelül öt százalékos processzoridőt lefoglaló erőforráskészletet, majd elindítottam egy nagy fájl átvitelét FTP-vel, valamint egy nagyobb PDF-dokumentum nyomtatását. A böngészőnek a szeme se rebbsz.

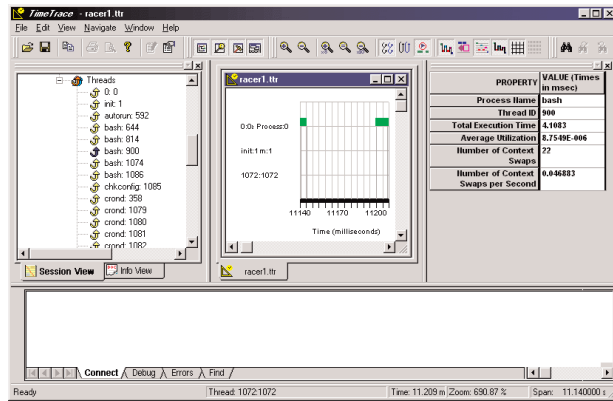
Támogatás és kézikönyvek

A termék próbaváltozatához három kézikönyv és egy füzet járt: TimeSys Linux/RT felhasználói kézikönyv, TimeTrace a TimeSys Linux/RT-hez felhasználói kézikönyv és a TimeSys Linux/RT programozói kézikönyv, a mellékelt füzet címe pedig „Valósidejű rendszerek tömör kézikönyve”, mely valósidejű szakkifejezéseket, fogalmakat és szerkezeteket tárgyal kivonatban formában. A TimeTrace és a Linux/RT felhasználói kalauz világosan bemutatja a két termék kapcsolatát. Minden leírás képekkel jól ellátott és nagyrészt jól megírt. A telepítési lehetőségek leírásai komoly tapasztalattal bíró telepítők számára íródtak. A Linux/RT és a TimeTrace felerészelt a valósidejű programokkal, illetve a tulajdonságaikkal foglalkozik általánosságban, és csak a könyvek másik fele szól magáról a termékről. A Linux/RT felhasználói kézikönyv és a Release Notes a grafikus erőforráskezelő taglalja, ennek ellenére az elindítása komoly kihívásnak bizonyult. Az én változatomban ez a program például le sem volt fordítva, valamint a fordítás menetének mikéntjére is csak nagy nehezen bukkantam rá a Release Notesban. Nem hiszem, hogy ez csak azért van így, mert a nem debianos telepítést választottam. Másfelől a Linux/RT felhasználói kézikönyv azt is leírja, hogy a measure-rk modult be kell tölteni a TimeTrace használatához. Ehhez képest sehoh sem találtam a lefordított measure-rk modul. Találtam viszont némi forráskódot, ami úgy sejtett, hogy ehhez a modulhoz tartozik, de ez egyáltalán nem volt nyilvánvaló. A TimeTrace könyve emlegetett még valami viewrk fájlt, hogy valaki azzal használta a TimeTrace-t. Így hát measure-rk modul hiányában a viewrk-val próbálkoztam. Ez legalább bevált.

Az „A” függelék tartalmazza a Linux/RT RK segédprogram-csomag parancsainak leírását. Ezek közül többnek a nevében kis–nagybetű elírás van, mások meg egyáltalán nem is léteznek. A /usr/local/bin könyvtárban található parancsok nagy részéről szintén nem sok derült ki a könyvből. Ott van például az órajel lekérdezése. A segédprogramlista szerint erre a clockfreq parancs szolgál. Na, ilyenrel egyáltalán nem találkoztam. Ezzel szemben az rkfreq parancs elárulta az órajelet is. Azután találtam egy charter nevű fájlt is. Erről semmiféle leírás

Jellemzők

Gyártó: TimeSys, Inc.
Levél cím: info@timesys.com
Ár: 199 dollár



2. kép A TimeTrace kezelőfelület

nem letem a könyvben, még csak man oldal sem tartozott hozzá. A PDF-formátumú felhasználói és programozói kézikönyvek a nyomtatott anyag pontos másának tűntek. Rákerestem hát bennük a fent említett fájlokra, de hiába. A 179 oldalas Programozói kézikönyv a valósidejű programok készítésével, végrehajtásával és hibakeresésével foglalkozik. Példaprogramok mutatják be a különféle függvényeket és rendszerhívásokat. Fejlesztéshez a GNU-Emacs, make, CVS, gcc és gdb ajánlható. A mellékelt mintaprogramok lefordításához azokat a librk.a könyvtárhoz kell csatolnunk. Három helyen is hivatkoztak erre a könyvtárra, de mindezek közül a Programozói kézikönyv volt a legbővebb. Készüljünk fel rá, hogy makefile-okban kell helyesbítünk a könyvtárra és a fejlécekre mutató útvonalakat. Hiányzó fájlok következtében néhány példaprogram fordítása kudarcba fulladt, de szerintem saját munkáinkkal ilyen gondunk nem lesz. Támogatáshoz a TimeSys weboldalán juthatunk. Természetesen levélben vagy faxon is tehetünk fel kérdéseket a TimeSysnek. A faxot nem próbáltam, így arról nem tudok nyilatkozni. A gyakran feltett kérdések gyűjteménye (FAQ) azonban nagyon üzleti kinézetű volt. Kereshető tudásbázis vagy elektronikus támogatás más formája nem volt számomra könnyen használható. Volt valami vitafórum-kezdmény, de a levelek csekély számából arra következtetek, hogy az egész nem túl hasznos. Amikor a racer és a rolling példaprogramokról kísérletet meg grafikonokat megjeleníteni a TimeTrace programban, apróbb nehézségekbe botlottam. Úgy tűnik, hogy a TimeSys éppen szervezeti bővítési gondokkal küszködik, így több napba telt mire válaszoltak. Miután azonban létrejött a levélkapcsolat, a TimeSys szakemberei segítettek a gondok leküzdésében. Mindent összevetve azt hiszem, a Linux/RT ígéretesnek tűnik, és képes megfelelően ellátni feladatát. A leírásra ráférne még egy kis javítás, hogy helyesen mutassa a fájlok helyét, illetve a parancsok használatának módját. A javításokat célszerű lenne még a Weben közzétenni, hogy ne kelljen az embernek mindig a teljes 12 MB-os tar fájlt letöltenie. Ha a támogatás minősége függ attól, hogy valaki megvette-e a terméket, akkor jó lenne, ha a vásárló ezt egyszerűen jelezhetné.

A próba során használt berendezések

A programot a következő számítógépeken vizsgáltam: 300 MHz-es Pentium II-es Wintel (Windowst futtató Intel processzoros) gépen, 64 MB memóriával és egy átlagos felbontású (1024x768-as) képernyővel. Az operációs rendszer Win98SE volt. A linuxos gép egy 333 MHz-es Celeront tartalmazott, 128 MB RAM-mal, azonos képernyővel, operációs rendszere pedig a Linux Caldera eDesktop2.4.

Daniel Lazenby (d.lazenby@att.net) 1983-ban találkozott először Unixszal, a Linuxot pedig 1994-ben fedezte fel, azóta lelkes híve.

Az Apache beállítása, trükkjei és hibakeresés

RPM, mod_perl, apachectl, telnet, mod_status, DSO, apxs, Apache::Status. Eléggé ijesztő? Lerner úr most mindent elmesél ezekről, és remélhetőleg megkönnyíti a rendszergazdák dolgát.

A legtöbb régi motoros tisztában van azzal, hogy a Linux, az Apache és más nyílt forráskódú programcsomagok megbízhatóbbak és beállításuk egyszerűbb, mint üzleti terjesztésű társaiké. Ez azonban egyáltalán nem azt jelenti, hogy az ingyenes programok hibamentesek és mindig minden úgy működik, ahogy beállítottuk. Az igazat megvallva, a nyílt forrású programok néha bosszantóan összetettek: nem mindig világos, hogy mely tulajdonságokat kell megváltoztatnunk, vagy egyáltalán hol kell kezdenünk a munkát.

E hónapban a rendszergazdák által az Apache beállításához szükséges eszközök egy részét tekintjük át. A cikk természetesen nem lehet teljes, hiszen megjósolhatatlan, hogy mi fog elromlani. Ha azonban a hiba okát sikerül feltárni, máris jó úton haladunk a megoldás, vagy legalábbis annak kidolgozása felé.

Az RPM-ek eltávolítása

Öt éve használom a Red Hat Linuxot. Akkoriban még a Red Hat egy kis connecticuti cég volt, és nem egy nagyvállalat, melynek nevét még az anyukám is ismeri. Azóta vagyok az RPM (Red Hat Package Manager) rajongója. Még emlékszem azokra az időkre, amikor az Internetről letöltött programok fordításához és telepítéséhez a makefile-okat kellett módosítani. Ám még mindig lenyűgöz az, hogy ma csak letöltöm a program bináris változatát, egyetlen paranccsal telepítem, és ugyanilyen egyszerűen eltávolíthatom a rendszerből. (Úgy hallottam, hogy a Debian csomagtelepítő rendszere ennél is barátságosabb, de ezt még nem volt alkalmam kipróbálni.)

Az RPM-et használó rendszergazdák általában egyszerűen lusták forráskódból telepíteni. Ez nemcsak időigényes művelet, hanem a program hónapokkal, évekkal későbbi eltávolítását is nehézkessé teszi. A hátrányok ellenére érdemes néhány programot forráskódból telepíteni; ezek közé tartozik a Red Hat csomag részét képező Apache is. Maga az Apache tulajdonképpen kis méretű program. A szolgáltatások többségét a fordítások beépülő modulok valósítják meg. Ha például azt szeretnénk, hogy az Apache automatikusan kijavítsa a félregepelt címekeket, a **mod_speling** (nem elírás!) modult kell a fordításkor felhasználnunk. Vagy ha biztos, hogy a kiszolgálón soha nem fut majd CGI program, akkor eltávolíthatjuk a **mod_cgi** modult. És így tovább – az Apache segítségével az igényeinknek tökéletesen megfelelő kiszolgálót alakíthatunk ki. Éppen ezért javaslom, hogy az Apache-t *mindenképpen* forráskódból telepítsük. A folyamat gyors, egy korszerű számítógépen nem tart tovább néhány percnél. A következő lépés a már létező RPM-ek eltávolítása azért, hogy elkerülhessük az RPM-adatbázis összezavarását. Ráadásul így mi is tudni fogjuk, hogy melyik fájl honnan származik.

Ha az Apache-t szeretnénk eltávolítani a Red Hat csomagból: az `rpm -e apache` parancsot kell használnunk. Ha pontosan szeretnénk tudni, hogy mi is zajlik a háttérben, a programot rábírhatjuk, hogy több üzenetet jelenítsen meg. Ehhez az `rpm -evv apache` parancsot adjuk ki. A legtöbb rendszer esetében azonban ez a parancs önmagában nem elég. Az RPM nemcsak a telepített fájlokról, hanem a csomagok egymás közötti kapcsolatáról is nyilvántartást vezet. Mivel a Red Hat-telepítés általában a **mod_perl**-hez és a **mod_php**-hoz is

tartalmaz RPM-eket, valószínűleg ezeket is törölnünk kell:

```
rpm -evv apache mod_perl mod_php
```

Ha egy csomag eltávolítása valamilyen kapcsolatot megszakít, az RPM azonnal hibaüzenetet ad és leáll, valamint azt is közli, melyik az a tevékenységünk, mely más csomagok használatát akadályozza meg. Ilyenkor el kell döntenünk, hogy a csomag (például mod_perl) csak egy másik csomaggal (például Apache) együtt használható-e, illetve hogy az eltávolítás nem okoz-e túl sok galibát.

A már létező Apache RPM-ek eltávolítása után elkezdhetjük a fordítást és a telepítést. (Természetesen a fordítás után is törölhetjük az RPM-eket; a lényeg, hogy a lefordított program telepítésekor a fájlok már ne legyenek a rendszerben.)

Töltsük le a legújabb változatot (a cikk írásakor ez az 1.3.12-es) egy helyi tükörlodról, vagy a <http://www.apache.org/> címről. A csomagot a

```
tar -zxvzf apache_1.3.12.tar.gz
```

paranccsal csomagolhatjuk ki. A z kapcsoló a csomagot először a gunzip használatával kibontja, a vv kapcsoló hatására pedig minden üzenet megjelenik a képernyőn. Mindkét kapcsoló csak a GNU tar programmal működik, mely a legtöbb Linux-változatban is megtalálható. A forráskódok kicsomagolása után az Apache-t a következő paranccsokkal fordíthatjuk le.

```
cd apache_1.3.12
# Belépünk az Apache könyvtárába.
./configure
# Az alapértelmezett beállítások betöltése.
make
# A forráskód lefordítása.
make install
# Az Apache-t a /usr/local/apache könyvtárba
# telepíti.
```

A fenti négy lépés az Apache-hoz kapcsolódó programokat a /usr/local/apache/bin, a naplófájlokat a /usr/local/apache/logs, a HTML fájlokat a /usr/local/apache/htdocs, a CGI programokat pedig a /usr/local/apache/cgi-bin könyvtárba helyezi. A „make install” parancsot rendszergazdaként kell kiadnunk.

Az Apache-t az **apachectl** paranccsal futtathatjuk, ennek alapértelmezés szerinti helye a /usr/local/apache/bin könyvtár. Az **apachectl** egy parancsfájl, mely az Apache indítását és leállítását könnyíti meg, illetve lehetővé teszi, hogy egyszerre csak egy Apache folyamat fusson a gépen. Az Apache indításához a

```
/usr/local/bin/apachectl start
```

parancsot kell kiadnunk. E sort az egyik rendszerindítási parancsállományba (például a /etc/rc.d/rc.local fájlba) illesztve az

Apache a Linux betöltésével egy időben indul el. Ez különösen fontos akkor, ha a már említett módszerrel eltávolítottuk az RPM-eket, hiszen az RPM-változat magától elhelyezi a megfelelő hívást az `/etc/rc.d/init.d` fájlba. Az `apachectl` segítségével le is állíthatjuk a kiszolgálót:

```
/usr/local/bin/apachectl stop
```

Az Apache indulásakor először a beállításfájlt olvassa be, ez a `/usr/local/apache/conf/httpd.conf`. Ez a fájl számos parancsot tartalmaz, melyeket egy vagy több érték követ. A kiszolgáló nevét például az alábbi sorral állíthatjuk be:

```
ServerName www.lerner.co.il
```

Minden modul saját parancsokat hoz létre, melyekkel az adott modul tulajdonságait szabályozhatjuk. Például a `mod_userdir` modul a `UserDir` parancsot teszi elérhetővé, mellyel a felhasználói könyvtárakon belül a honlap fájljai számára fenntartott könyvtárat határozhatjuk meg. De mi van akkor, ha a `mod_userdir` modult nem telepítettük? Ilyenkor az Apache nem tud mit kezdeni a `UserDir` paranccsal, s ezért hibüzenettel leáll. A megoldás az `apache configtest`, mely ellenőrzi, hogy a `httpd.conf` fájlban megadott parancsok és értékek értelmezhetőek-e. Ha minden utasítás ismert és az értékekkel sincs gond, akkor az „OK” üzenetet kapjuk.

Egy másik megoldás, hogy a bizonytalan kimenetelű parancsokat két `<IfModule>` tag közé zárjuk. Ez a rész csak akkor kerül végrehajtásra, ha a megadott modul be lett töltve. A `UserDir` parancs tehát mindig szerepelhet a fájlban, de csak ha szépen „becsomagoljuk”:

```
<IfModule mod_userdir.c>
    UserDir public_html
</IfModule>
```

Az `<IfModule>` rugalmassága különösen a DSO-ként lefordított modulok használatakor jön jól.

A Telnet használata

Az Apache-kiszolgáló lefordítása, telepítése, beállítása és elindítása után még mindig adódhatnak gondok. Az `apachectl configtest` paranccsal ellenőrizhetjük a beállításfájl parancsainak és értékeinek érvényességét, de ez még nem jelenti azt, hogy a parancsok pontosan azt teszik, amit mi szeretnénk.

A kiszolgáló működésének ellenőrzésére a telnet a legalkalmasabb. Ezt a programot minden Linux-felhasználó ismeri: segítségével egy másik gépre jelentkezhetünk be. A telnet azonban bármelyik TCP-kaput csatlakozhat a hívott számítógéphez, nem csak a 23-ason. Használhatjuk tehát a 25-ös (SMTP), a 110-es (POP) és akár a 80-as (HTTP) kapukat is. Ez a módszer tökéletesen alkalmazható nemcsak annak ellenőrzésére, hogy a webkiszolgáló működik-e, hanem alapvető ellenőrzésekre is.

A módszer használatához azt kell megértenünk, hogy minden TCP/IP szolgáltatáshoz egy-egy kapu tartozik, a helyi és a távoli gépen egyaránt. Ezért a két gép közötti telnet kapcsolathoz két IP-címre, a helyi gép egy tetszőleges kapujára és a távoli gép 23-as kapujára lesz szükségünk. Hasonlóképpen, egy levél továbbításához a helyi gép egy tetszőleges kapujáról a kiszolgáló 25-ös kapujára kell csatlakoznunk. A leggyakrabban használt szolgáltatások kapuszámát (beleértve azokat is, melyeket nem célszerű megváltoztatnunk: FTP, SMTP, TELNET) a `/etc/services` fájlban találjuk meg. Ha a telnet-tel kívánunk egy gép valamelyik kapujára csatlakozni, akkor egyszerűen adjuk meg a kapuszámot is (vagy a nevét, amennyiben az szerepel a `/etc/services` fájlban). Ha például a `www.lerner.co.il` gép 80-as

kapun elérhető HTTP-kiszolgálójára kívánunk csatlakozni, akkor a

```
telnet www.lerner.co.il 80
```

parancsot kell begépelnünk. Ha a kiszolgáló más kaput használ (ezeket az Apache Listen és Port parancsaival állíthatjuk be), akkor természetesen azt a számot kell megadnunk. Ha például a kiszolgáló a 8080-as kapun érhető el, akkor a

```
telnet www.lerner.co.il 8080
```

paranccsal kapcsolódhatunk.

Az Apache képes egyszerre több kapun is kérelmeket fogadni, ha azokat helyesen beállítjuk a `httpd.conf` fájlban.

Ha a megadott kaput semmilyen szolgáltatás nem használja, akkor a „connection refused” üzenetet kapjuk. Ilyenkor kukkantsunk bele az Apache naplófájljába (alapértelmezés szerint ez a `/usr/local/apache/logs/error_log`) a hiba felderítéséhez.

Ha a megadott kaput valóban egy HTTP-kiszolgáló használja, előbb az a karakter jelenik meg, mellyel visszatérhetünk a telnet parancsorához (általában CONTROL), majd létrejön a kapcsolat. Amit a bejelentkezés után látunk, az a hívott kiszolgáló típusától függ. Míg az SMTP, az FTP és a POP a kiszolgáló nevének kiírásával üdvözlő a felhasználót, a HTTP általában nem jelenít meg semmit. Feltételezi, hogy tudjuk a kiszolgáló nevét (ha már egyszer sikerült kapcsolódni hozzá...), és azt is, hogy a kapcsolat sikeres volt.

Most elkezdhetjük beírni a HTTP-parancsokat. A legegyszerűbb lekérdezés a GET /, mely után természetesen ENTER-t kell nyomnunk. Ezt a formát a kompatibilitás a legtöbb korszerű kiszolgáló támogatja, de már nem használatos. Nem sokkal az ENTER leütése után a honlap kezdőlapjának tartalmát láthatjuk. A formázatlan HTML-fájl olvasása először kissé nehézkes, de ne felejtjük el, hogy most csupán egy egyszerű hibakereső eljárásról beszélünk.

Összetettebb lekérdezéseket is intézhetünk a kiszolgálóhoz a HTTP/1.0 használatával. Ez a HTTP első olyan változata, mellynél a lekérdezésekben és válaszokban fejléc is található. A kérelmet és a választ egy vagy több fejlécsor vezeti be. A fejléc a nevét, egy kettőspontot és egy karakterláncot tartalmaz. A fejléc és a törzs között egy üres sor található.

A fenti egyszerű lekérdezést a GET / HTTP/1.0 alakban is megadhattuk volna, ezzel jelezve, hogy az ügyfél a HTTP 1.0-s változatát is megérti. Ezután kétszer kell ENTER-t nyomnunk – az első a sor végét jelöli, a másik pedig azt, hogy a parancssor és a HTTP-kérelmek között nem akarunk fejléccet elküldeni.

A telnet segítségével név-érték párokat is küldhetünk a kiszolgálónak, a két elem közé egyenlőségjelet, a párok elé pedig & jelet kell tennünk. Általában ezeket az értékeket egy GET kérelemmel adjuk át egy CGI vagy más, dinamikus tartalmat létrehozni képes programnak. Például:

```
GET /cgi-bin/foo.pl?nev1=ertek1&nev2=ertek2 HTTP/1.0
```

Az ENTER kétszeri leütése után a válasz fejlécét, majd a `foo.pl` nevű CGI program kimenetét láthatjuk.

A kérelemmel együtt fejléceket is továbbíthatunk. Az első GET sor után az ENTER-t egyszer lenyomva írunk be egy vagy több fejléccet, és ne felejtünk el mindegyik után egy sort kihagyni. Például:

```
GET /cgi-bin/foo.pl?nev1=ertek1&nev2=ertek2 HTTP/1.0
Accepts: text/html
Accept-language: text/html
```

Az utolsó fejlécsor után nyomjunk kétszer ENTER-t – egyszer a fejléc, másodsor pedig a kérelem befejezéséhez.

Hány kiszolgálót futtassunk?

Még a kevésbé látogatott weboldalak üzemeltetői is több Apache-folyamatot futtatnak egyszerre. A régebbi kiszolgálók általában addig várnak, amíg egy új kapcsolat szükségessé teszi az új folyamat indítását. Az Apache készítői nem javasolják ezt a módszert: az Apache már indulásakor több alfolyamatot is elindít.

Egy alfolyamat egyszerre csak egy HTTP-kapcsolatot kezel, ez azt jelenti, hogy a futó Apache-folyamatok számának a látogatók aktuális számával kell megegyeznie. A határértéket a `MaxClients` paranccsal állíthatjuk be. Ennek alapértéke 150. Ha a **MaxClientst** túl alacsonyra állítjuk, akkor az újonnan kapcsolódó látogatóknak várniuk kell, míg valamelyik alfolyamat felszabadul.

Az Apache a `httpd.conf` fájlban megadott elvek alapján állandóan változtatja a kiszolgálók számát, a bejövő kérelmeknek megfelelően. A **MinSpareServers** és a **MaxSpareServers** paranccsal szabályozhatjuk, hogy hány tartalék kiszolgáló fusson állandóan a háttérben, melyek feladata a friss kérelmek azonnali kiszolgálása. Ha a szabad tartalékok száma a `MinSpareServers`-ben megadott szintre csökken, az Apache több új kiszolgálót indít. Másrésztől, ha a szabad folyamatok száma eléri vagy meghaladja a `MaxSpareServers` paranccsal meghatározott szintet, az Apache azonnal leállítja a feleslegesen futó folyamatokat.

Ha a kiszolgáló elindul és válaszolni is képes, de a kapcsolódás túl sok időt vesz igénybe, akkor valószínűleg a fenti értékekkel lesz a gond. Ilyenkor növeljük a `MaxSpareServers` vagy a `MaxClients` értéket azért, hogy minél kevesebb felhasználónak kelljen várnia a kapcsolódásra.

Természetesen az új folyamatok indítása nagyon leterheli a számítógépet – a processzor kevesebb időt tud szánni egy-egy folyamatra, és a memória is vérszesen fogy. A `mod_perl` különösen sok memóriát fogyaszt, tehát lehetőleg ne indítsunk túl sok olyan Apache-folyamatot, mely a `mod_perl` modult is használja. A Linux `free` parancsával bármikor ellenőrizhetjük az elérhető fizikai és virtuális memóriát, a `top` paranccsal pedig az egyes folyamatok által elfogyasztott processzoridőt és egységadatokat jelenítheünk meg.

Mivel a webkiszolgálóknak a lehető leggyorsabban válaszolniuk kell a beérkező kérelmekre, és mivel a virtuális memória jóval lassabb a fizikai RAM-nál, ezért fordítsunk különös gondot a virtuális memória fogyasztására, korlátozzuk a használatát.

Ha a honlap és az adatbázis (például MySQL, vagy PostgreSQL) ugyanazon a gépen található, komolyabb gondok is előfordulhatnak. A dinamikusan létrehozott oldalak látogatottságának növekedésével természetesen egyre több Apache-folyamat fut a gépen.

De a látogatók kiszolgálása céljából az adatbázist kezelő kapcsolat számának is növekednie kell. Egy ponton a honlap saját népességének áldoztatává válik, hiszen az Apache és az adatbázis egyre véresebb küzdelmet folytat a rendszererőforrásokért. A nagy forgalmú, adatbázist is használó honlapok esetében érdemes tehát a két feladatot különválasztani: a webkiszolgáló külön gépéhez egy vagy több, az adatbázisok kezeléséért felelős gép csatlakozzon.

A `mod_status`

Az Apache állapotáról a `mod_status` modullal készíthetünk pillanatfelvételt. A `mod_status` a HTTP-kiszolgálók pillanatnyi állapotát írja le (új kapcsolatra vár, kérelmet olvas be, kezeli a kérelmet, a választ állítja össze stb.)

A `mod_status` alapértelmezés szerint az Apache része, tehát indításához mindössze a megfelelő parancsokat kell megadnunk, majd az alapértelmezett kérelemkezelő (handler) eljárást kell „`server-status`”-ra állítanunk. Ha ezután olyan URL érkezik, melynek kezelője „`server-status`”, az Apache állapotjelentést készít, a kérelem többi részét figyelmen kívül hagyva.

Ezért a `mod_status` általában csak egyetlen URL-lel használjuk. Pé-

dául létrehozunk egy „`/server-status`” nevű URL-t a kiszolgálón, ezt megtekintve a látogató a kiszolgáló állapotáról tájékozódhat. Érdemes mindig a teljes állapotjelentést megjeleníteni (`ExtendedStatus On`). Nézzünk egy egyszerű beállítást:

```
<Location /server-status>
    SetHandler server-status
</Location>
ExtendedStatus On
```

E négy sort illesztjük a `httpd.conf` fájlba, majd indítsuk újra az Apache-t (vagy küldjünk neki HUP jelet). Ezután a `/server-status` URL-t megtekintve az alábbihoz hasonló üzenetet kapunk:

```
Server Version: Apache/1.3.12 (UNIX) mod_perl/1.24
Server Built: Mar 29 2000 12:25:42
```

```
Current Time: Friday, 21-Jul-2000 16:02:51 IDT
Restart Time: Friday, 21-Jul-2000 16:02:48 IDT
Parent Server Generation: 2
Server Uptime: 3 seconds
Total accesses: 0 - Total traffic: 0 kB
CPU Usage: u0 s0 cu0 cs0
0 requests/sec - 0 B/second -
1 requests currently being processed, 4 idle servers
```

Az állapotadat a kiszolgáló indításának időpontját, a kapcsolatok számát és forgalmát tartalmazza. Azt is megjeleníti, hogy mennyi bájtnyi adatot szolgál ki éppen ez a folyamat, illetve hány folyamat várakozik feladat nélkül. A `mod_status` segítségével betekintést nyerhetünk az Apache-kiszolgáló működésébe, és azt is kideríthetjük, hogy megfelelően állítottuk-e be a `MaxSpareServers` értékét.

A `mod_status` ezután a következő alakban jelenít meg adatokat (igen, első ránézésre elég titokzatos):

```
W_____.....
.....
.....
.....
```

Minden „`W`” karakter egy feladat nélkül várakozó Apache-folyamatot jelöl. Az új kapcsolatra váró folyamatok jele „`_`”; a kérelmet beolvadó „`R`”, a választ küldőké „`W`”. A jelek természetesen állandóan változnak, hiszen az Apache-folyamatok állapota sem állandó. Ezt követően az egyes működő folyamatok állapotáról tájékozódhatunk. Láthatjuk, hogy mely kapcsolatok feldolgozása tart sokáig, melyek a legnépszerűbb kapcsolatok a hónapban, s még ezernyi nyit. Természetesen nem túl bölcs dolog az állapotadatokat az egész világ elé tárunk. Szerencsére az „`Order`”, „`Restrict`” és „`Deny`” parancsokkal a hozzáférést letilthatjuk, illetve elérhetővé tehetjük egy adott tartomány számára. Például:

```
<Location /server-status>
    SetHandler server-status
    Order deny,allow
    Deny from all
    Allow from .lerner.co.il
</Location>
```

A fenti beállítások hatására a `mod_status` kimenete csak a `lerner.co.il` tartomány gépei számára lesz elérhető. A más tartományokból érkező kérelmekre a kiszolgáló az „`Access forbidden`” (Hozzáférés megtagadva) üzenettel válaszol.

Tervezés a DSO-val

Ez idáig feltételeztük, hogy az Apache-t mindenki statikusan fordította, vagyis a modulok fordításakor bekerültek a programba. Ez az Apache fordításának hagyományos módja, és ez az alapértelmezés is, ha a `./configure` parancsot használjuk.

A fenti beállítás egyetlen hibája, hogy a rendszer így nem túl rugalmas. Mi történik például akkor, ha hónapokkal később kiderül: egy fontos modult elfelejtettünk a fordításakor megadni? Ilyenkor az egész csomagot újra kell fordítanunk, most már figyelmesebben meghatározva a programba kerülő modulokat. Ez a módszer első ránézésre nem tűnik túl hátrányosnak – elvégre nem kell mindennap új modulokat a programba fordítanunk.

Azonban a baj ennél sokkal mélyebben gyökerezik. Először is: miért foglalnánk le a nem használt modulok számára memóriát? Másodsor: miért kellene egy-egy új modul vagy modulváltozat megjelenésekor az egész csomagot újrafordítanunk?

A feladatot az Apache Dynamic Shared Objects (DSO) felhasználásával oldhatjuk meg. Így az Apache fordításához mindössze két modulra lesz szükségünk: az egyik a `mod_core` (ez az alapvető szolgáltatásokat tartalmazza), a másik pedig a `mod_so` (ezzel tölthetjük be a DSO-kat). A többi modult csak szükség esetén kell betöltenünk.

Mivel így a modulok a programon kívül helyezkednek el, ezért frissítésük is egyszerűbben, a program újrafordítása nélkül elvégezhető.

És éppen ez az, ami a folyamatos működést igénylő, forgalmas webkiszolgáló esetén rendkívül előnyös.

Ha az Apache-t DSO-k használatával fordítjuk, el kell döntetnünk, hogy mely modulokat kívánjuk a program részévé tenni, és melyeket DSO-ként használni. Javaslom, hogy mindent DSO-ként készítsünk el, kivéve az Apache által alapértelmezés szerint telepített modulokat. Ehhez a `configure` parancsfájl más módon kell meghívunk:

```
./configure --enable-shared=max
```

Így a `mod_so` is a program része lesz, az Apache pedig az összes alapértelmezés szerinti modullal lesz lefordítva. Miután a `make` parancsral lefordítottuk, a `make install` segítségével telepíthetjük az Apache-t, mely a szokásos módon működik ezután is. Az egyetlen észrevehető különbség, hogy a modulok csak akkor töltnének be, ha éppen szükség van rájuk.

Az `--enable-shared` változóval fordított Apache által létrehozott eredeti `httpd.conf` fájl némileg különbözik attól, mint amit a modulokkal egybefordított Apache készít. A legfontosabb különbség, hogy a parancsok `IfModule` tagok között foglalnak helyet, ez lehetővé teszi, hogy az Apache bizonyos modulok betöltése nélkül is elinduljon. Ezenkívül minden modult a `LoadModule` parancsral kell betöltenünk, majd az `AddModule` parancsral engedélyeznünk. Például:

```
LoadModule perl_module lobexec/libperl.so
AddModule mod_perl.c
```

A `LoadModule`-nak két értéke van: az egyik a modul neve, a másik az `.so` fájl. A névnek meg kell egyeznie azzal a névvel, mellyel a DSO modult fordítottuk, a fájlnevének pedig a `/usr/local/apache` könyvtár egyik alkönyvtárában lévő fájlra kell mutatnia. A fenti példában (és alapértelmezés szerint is) a DSO modulok a `/usr/local/apache/libexec` könyvtárban helyezkednek el.

A `httpd.conf` fájl legtöbb parancsával ellentétben a `LoadModule` és `AddModule` parancsok esetében elhelyezésük sorrendje is számít. A `LoadModule` parancsra meg kell előznie az `AddModule` parancsot. Mielőtt egy parancsot használnánk, az azt magában foglaló modult be kell töltenünk, és engedélyeznünk kell. Hogy még kellesebb legyen az életünk, néhány modul csak akkor működik, ha azt egy másik modul után töltsük be. Ha van olyan beállítóprog-

ram, mely a `LoadModule/AddModule` parancsokat automatikusan beilleszti, akkor használjuk azt, ugyanis segítségével rengeteg tökéletlen beállításból fakadó hibát küszöbölhetünk ki.

Az apxs

Miután az Apache-t DSO támogatással lefordítottuk, az új modulokat bármikor beilleszthetjük, de ezeket az Apache fordításánál használt beállításokkal kell lefordítanunk. Ezt a *Ralf S. Engelschall* által írt `apxs` (Apache Extension) nevű program automatikusan elvégzi. Az `apxs` segítségével a modulokból DSO-kat (`.so` fájlokat) készíthetünk, ezeket pedig beilleszthetjük az Apache-ba.

Sajnos, a programhoz nem sok leírás jár, tehát a program feladatát és működését magunktól elég nehéz megérteni.

Tegyük fel, hogy az Apache-t már lefordítottuk DSO-támogatással. Néhány héttel később észrevesszük, hogy a kiszolgáló egy csomó kapcsolatot „File not found” üzenettel tagad meg, mert a felhasználók nem képesek kiolvasni és begépelni az URL-ekben meghatározott különleges karaktereket. Az egyik megoldás, hogy az URL-eket teljesen átalakítjuk „normális” karakterekkel. Egyszerűbb azonban, ha a `mod_speling` modult telepítjük, s így a félregépelésből és a nagybetű-kisbetű különbségekből adódó hibákat nagymértékben kiszűrhetjük. A `mod_speling` DSO-ként történő fordításához a

```
/usr/local/apache/bin/apxs -c mod_speling.oc
```

parancsot kell kiadnunk. Az `apxs` a `gcc` meghívásával fordítja le a `mod_speling` modult. Eredményül nem futtatható fájlunk, hanem egy, az Apache által betölthető könyvtárfájl. Ha a fordítás sikeres volt, akkor a `mod_speling`-et a következő parancsral telepíthetjük:

```
/usr/local/apache/bin/apxs -i -n -a mod_speling.so
```

Az `apachectl configtest` parancs különösen jól jön új DSO modulok telepítésekor. A parancs lehetővé teszi, hogy a hozzáadott modul tényleg a helyén legyen és működjön, illetve az Apache megérti az `IfModule` tagokon kívül elhelyezkedő új parancsokat.

A mod_perl

A Perl nyelven írt új modulok írását és a meglévő beállítását lehetővé tevő `mod_perl` modult is használhatjuk DSO-ként. Ehhez, a `mod_speling`-hez hasonlóan, az `apxs`-re lesz szükségünk. A `mod_perl` azonban jóval összetettebb modul, mint a többi, és fordításához számos külső elemből származó adat szükséges. A `mod_perl` modult ezért az önálló Perl modulokhoz hasonlóan a `perl Makefile.PL` parancsral kell beállítanunk, melyet egy `make` és egy `make install` parancs követ.

Ha a meglévő Apache-változatba szeretnénk a `mod_perl` egy újabb változatát fordítani, adjuk ki az alábbi parancsot:

```
perl Makefile.PL \
USE_APXS=1 WITH_APXS=/usr/local/apache/bin/apxs
```

Ha a `mod_perl` nemcsak a PerlHandler, hanem a különböző Apache kezelők (handler) számára is elérhetővé kívánjuk tenni, akkor kapcsoljuk be az `EVERYTHING` kapcsolót:

```
perl Makefile.PL \
USE_APXS=1 WITH_APXS=/usr/local/apache/bin/apxs \
EVERYTHING=1
```

Miután a `Makefile`-t a fenti módon elkészítettük, a `mod_perl`-t a következő parancsokkal fordíthatjuk le és telepíthetjük a meglévő Apache-kiszolgálóba:

```
make
make test
make install
```

E módszerrel nemcsak a mod_perl egy új példányát telepíthetjük az Apache-ba, hanem frissíthetjük is a meglévő példányt. Annak ellenőrzéséhez, hogy a mod_perl bekerült-e az Apache-ba, jelentkezzünk be telnettel a kiszolgálóra a megfelelő (általában 80-as, vagy 8080-as) kapuzámmal, majd adjuk ki az alábbi parancsot:

```
HEAD / HTTP/1.0
```

Ez a kiszolgáló kezdőlapjának fejlécét küldi vissza, melyben többek között egy, a kiszolgáló típusát meghatározó Server fejléc sort is kell találnunk. A mod_perl ehhez az üzenethez saját jelzését is csatolja, tehát ha mindent jól csináltunk, akkor a következő szöveget kell látnunk a fejlécben:

```
Server: Apache/1.3.12 (UNIX) mod_perl/1.24
```

Mivel a mod_perl frissítései általában nem az Apache új változataival egy időben jelennek meg, ezért a fenti módszer különösen jól használható az új mod_perl telepítésére.

Az Apache::Status

A mod_status csak az Apache-folyamatok állapotáról tudósít, az egyes modulok pillanatnyi helyzetéről nem tudunk meg semmit. Mondhatnánk: miért is lenne ez fontos? Nem mindegy nekem, hogy épp mi történik a mod_mime, vagy épp a mod_speling modulban?

A mod_perl esetében azonban, ahol rengeteg összetett művelet folyik egyszerre, nem lenne rossz, ha állandó visszajelzést kaphatnánk ezekről. A Perl Apache::Status modulja, mely a mod_perl modulallal működik, pontosan ezt az leírást képes nyújtani.

Az Apache::Status indításához új szakaszt kell létrehozunk a httpd.conf fájlban. A mod_status-hoz hasonlóan most is új Location részt készítünk, mely kezelőt rendel egy virtuális URL-hez, a „/perl-status”-hoz:

```
PerlModule Apache::Status
<Location /perl-status>
    SetHandler perl-script
    PerlHandler Apache::Status
</Location>
```

A kiszolgáló újraindítása után (de küldhetünk neki HUP jelet is) a /perl-status URL lekérésekor egy menü tárul elénk, „Environment”, „Inheritance tree” és hasonló lehetőségekkel. A mod_perl más Perl moduljai, például a HTML::Mason, is képesek csatlakozni az Apache::Status-hoz, így azok tulajdonságaiba is betekintést nyerhetünk. A Mason esetében például egyszerű kezelőfelülettel tekinthetjük át a pillanatnyi helyzetet, illetve a lefordított és gyorstárba került összetevők listáját.

Egy rövid történet

A cikk írása előtti napokban éppen a fenti módszerekkel igyekeztem megoldást találni egy, a saját rendszeremben felmerült, a HTML::Mason telepítésével kapcsolatos nehézségre. Egy ismerősöm kiszolgálójának gondot okozott a növekvő forgalom kezelése: néhány óránként minden Mason-alapú kiszolgáló leállt, ezt egy-két órával később a nem Mason-alapú kiszolgálók leállása követte. A „leállítás” azért talán erős kifejezés: a böngésző elküldte a kérelmet a kiszolgálónak, de a kapcsolat úgy tíz perc után időtűllépés miatt megszakadt.

Tehát pontosan mi is történt ott, és hogyan javítottam meg?

Az első gondolatom az volt, hogy a kiszolgálón elfogyott a fizikai memória. A top és a free segítségével megvizsgáltam a rendszert, de semmi különöset nem tapasztaltam. Egyrészt ez megnyugtató érzés volt, viszont ez azt is jelentette, hogy a kiszolgálón egyszerűen elfogynak a szabad folyamatok, annak ellenére, hogy a kapcsolatok legnagyobb számát 150-re állítottam. Saját honlapom elég szép forgalmat bonyolít, de azért 150 egyszerre kapcsolódó felhasználó még nálam is ritkaságszámba megy. Valami más lehet a baj, gondoltam. Az világos volt, valami a Masonnal és talán a mod_perl-lel nincs rendben. Ekkor úgy döntöttem, hogy megpróbálom a mod_perl-t frissíteni a legújabb változatra (akkor ez az 1.24 volt), a fentebb ismertetett eljárások segítségével. Frissítettem tehát a HTML::Mason-t és a kapcsolódó modulokat, az apachectl paranccsal újraindítottam az Apache-t, és reménykedtem, hogy a galibát elfelejthetjük. Sajnos azonban a frissítés nem oldott meg semmit. A kiszolgáló néhány óra működés után továbbra is beszüntette a válaszadást a kérélmekre. Az Apache::Status-szal áttekintettem a mod_perl állapotsorát, de ott sem volt bibi.

Ekkor a mod_status modulal belenéztem az Apache állapotjelentésébe és észrevettem, hogy egyszer csak rengeteg Apache-folyamat leáll a válaszadás ("W" jel) közben. Tehát az állapotjelentés sorai lassan, de biztosan csupa "....." jelből (szabad folyamatok) "WWWWW" jelekké (válaszadás folyamatban) alakultak át. A Mason minden meghívása lefoglalt egy folyamatot, melyet soha többé nem engedett el! Ezek után nem csoda, hogy lassan leállt a kiszolgáló: ha a honlap Masonra alapuló részeit többen keresték volna fel, a leállás még hamarabb bekövetkezett volna.

Átnéztem a Mason beállításfájlját és rájöttem, hogy az Apache::Session modul (ez a mod_perl programjai számára a felhasználói tevékenység nyomon követését teszi lehetővé) nem tér vissza. Tehát a Mason-összetevő meghívásakor minden rendben zajlott, egészen addig, míg az eljárásnak vissza kellett volna térnie – ekkor ugyanis a futtató program a MySQL adatbázisra történő, örökkévalóságig tartó várakozásba kezdett. Az általam talált megoldás nem volt különösebben ügyes, de megállta a helyét: leállítottam az Apache::Session MySQL változatát (neve Apache::Session::MySQL) és helyette a hagyományos változatot indítottam el (neve Apache::Session::File). A kiszolgáló újraindítása után nagy örömeinkre minden a legnagyobb rendben működött.

Összegzés

Az Apache egy csodálatos és megbízható HTTP-kiszolgáló, ennek ellenére rendkívül összetett, használatához és megfelelő behangolásához elkél némi tapasztalat. Ha a mod_perl is bekapcsolódik a folyamatokba, a helyzet még bonyolultabbá válik. Szerencsére az Apache-t úgy is telepíthetjük, hogy a modulok telepítése és frissítése egyszerűen és fájdalommentesen végrehajtható legyen.

A számos modul és segédeszköz (például a mod_status, vagy az Apache::Status) lehetővé teszi, hogy a „motorháztetőt” működés közben hajtsuk fel, ezek segítségével a bajok felderítése és kijavítása egyszerűbbé és gyorsabbá válik. Ezzel pedig rengeteg időt nyerhetünk – senki nem görnyed órákon, napokon keresztül a képernyő előtt csak azért, hogy végre működésre bírja azt a fránya kiszolgálót.

Az ATF honlapja: <http://www.lerner.co.il/atf/>



Reuven M. Lerner (reuven@lerner.co.il) cége internetes tanácsadást vállal, székhelyük Modi'in-ben, Izraelben van. Éppen mostanában fejezi be (végre!) *Core Perl* című könyvét, mely a Prentice Hall kiadónál jelenik meg.

Tartalomkezelés

Kövessük nyomon honlapunk frissítését a Mason segítségével!

Társlapunk az utóbbi néhány hónapban a Masonnal, *Jonathan Swartz* webfejlesztő rendszerével foglalkozott, mely az Apache-t, a **mod-perl**t és a HTML/Perl mintákat kapcsolja össze.

Általában a Mason a webalkalmazásokhoz kötődik – elsősorban azokhoz, amelyek háttéradatbázist alkalmaznak, és dinamikus tartalmuk is van. A Masont én rendkívül jól használható eszköznek találtam a tanácsadói feladataimban, mivel lehetővé tette a weblapok gyors és könnyű előállítását.

A csomag azonban egyszerű kiszolgálóoldali webalkalmazásoknál többre is használható. Az egyik legérdekesebb Mason-alkalmazás a Mason Content Manager (Mason-CM), mely alapvető tartalomkezelési lehetőségeket teremt, a fájlok külső kiszolgálóra (production server) történő felvitelétől kezdve, a beépített helyesírás-ellenőrzőn át az RCS változatkezelésig, valamint a Mason összetevőinek szerkesztéséig.

Mivel a Mason-CM Masonben íródott (tehát szükséges hozzá az Apache, a mod_perl és a megfelelő Perl modulok), természetesen jól működik egyszerű, csak HTML-t és képeket tartalmazó statikus lapokkal is.

Mindenképpen ajánlom, hogy még azok a webmesterek is vessenek egy pillantást a Mason tartalomkezelőjére, akiket egyébként nem érdekel a Mason és a mod_perl, hiszen ennek az egyszerű és ingyenes csomagnak oly sok hasznos tulajdonsága van.

Mi az a tartalomkezelés?

Ahogy a webhelyek egyre bonyolultabbak lesznek, úgy válnak a hozzájuk tartozó szervezési munkák is egyre összetettebbé. Míg régebben általános volt, hogy egy profi webhelyet egyetlen ember kezelhetett, ma már nem ritka, hogy akár egy kisebb vagy közepes méretű helyet is legalább három ember tart karban: a tartalomfelelős vagy szerkesztő, a tervezőgrafikus és a programozó. Még az ilyen kis csapatoknál is előfordul, hogy több ember próbálja ugyanazt a fájlt módosítani egy időben. Ezt a gondot már évekkel ezelőtt megoldották az olyan változatkezelő rendszerek segítségével, mint amilyen az RCS vagy a CVS. Csakhogy ezeket a rendszereket programozók tervezték programozóknak, így gyakran elég elrettentők lehetnek egy tervezőgrafikus vagy szerkesztő számára.

A Web számos olyan új kihívást is jelenthet, amely különbözik a hagyományos programozás világában megszokott fejlesztésektől. Például a programokat általában megírják, lefordítják, ellenőrzik, és hibakeresést végeznek rajtuk, majd az egész kezdődik elölről, míg végül kiadják. A Web ellenben másként működik. Amint a HTML-fájl a kiszolgálóra kerül, azonnal hozzáférhető mindenki számára az Interneten. Ez jó, hiszen az esetleges hibákat azonnal felfedezhetjük, de azt is jelenti, hogy a helyek szabályos időközönként frissíthetik tartalmukat, hosszan tartó eljárás igénybevétele nélkül.

Egyúttal persze a módosítás eredménye – legyen az oka fejlesztés vagy akár hiba – azonnal elérhető mindenki számára, aki éppen a jó URL-t írja be a rossz időben. Emiatt a legtöbb közepes, illetve nagy webhelyet két webkiszolgálón futtatják.

A belső kiszolgálót (staging server) a fejlesztők, a szerkesztők és a grafikusok írhatják, szerkeszthetik, majd ellenőrizhetik változtatásaik hatását. Csak akkor kerülhetnek át az adatok a második kiszolgálóra, ha minden fájl rendben van. Ezt a helyet külső kiszolgálónak (production server) nevezik.

Természetesen ez a két kiszolgáló futtat ugyanazon a számítógépen is. A lényeg, hogy valahogyan elválasszuk őket egymástól, hogy a látogatók ne férhessenek hozzá a fejlesztés alatt álló tartalomhoz. Azt is lehetővé kell tennünk, hogy a belső és a külső kiszolgálónak azonos könyvtárszerkezete legyen.

A Mason-CM a Mason alkotóelemeiből épül fel, s viszonylag egyszerűvé teszi bárki számára egy tartalomkezelő rendszer beindítását a saját rendszerén. A felhasználói felület nem éppen csodaszép, és néhány kisebb nehézséggel találkoztam a telepítés közben. Mindazonáltal feladatát megfelelően ellátja, és lehetővé teszi, hogy egyszerre több felhasználó is dolgozzon ugyanazon a feladaton anélkül, hogy egymás „tyúkszemére lépnének”.

A Mason-CM telepítése

A Mason-CM telepítése előtt először szükség lesz egy működő Apache-változatra, **mod_perl** és Mason támogatással. Ezenkívül le kell töltenünk az alábbi Perl modulokat a CPAN-ról: MLDBM, Image::Size, URI::Escape és File::PathConvert. Győződjünk meg arról, hogy ezek a PerlModule jelzéssel kerültek be az Apache beállításfájljába (httpd.conf), és növeljük meg a különböző Apache al-folyamatok (child process) számára adott megosztott memória méretét. A Mason-CM gzippel tömörített tar fájlként tölthető le a termék honlapjáról (lásd a cikk végén a kapcsolódó címeket). A tárolófájl a Mason összetevőket tartalmazó könyvtárba érdemes kicsomagolni. A magam részéről a /usr/local/apache/mason/cm könyvtár választottam. Ha az állományt kicsomagoltuk, a /cm könyvtárnak egy README és egy INSTALL fájl kell tartalmaznia. Bár itt a telepítéshez szükséges összes lépést ismertetem, mégis jó ötletnek tűnik végigolvasni ezeket a fájlokat, csak a biztonság kedvéért. Mivel a Mason-CM-nek kell lehetővé tennie azt, hogy minden fájl csak egy felhasználó módosíthasson egyszerre, valamint azért hogy a tartalomkezelő rendszert csak az arra hivatott felhasználók érhesék el, a /cm könyvtár eléréséhez HTTP-hitelesítést kell beállítanunk (a Mason-CM csak jelszóval védett könyvtárból futtatható). Egy könyvtárat a legegyszerűbben úgy tehetünk jelszóvédté, hogy egy **.htaccess** fájlt helyezünk el benne. A .htaccess fájl felülbírálja az alapértelmezett Apache beállításokat az adott könyvtárra, illetve annak összes alkönyvtárára. Itt van például a saját tartalomkezelőmhöz tartozó .htaccess fájl:

```
AuthName "Content Management System"
AuthType Basic
AuthUserFile /usr/local/apache/conf/st-passwords
require valid-user
```

Az AuthName érték határozza meg a jelszó bekérésekor megjelenő szöveget. (Enélkül a felhasználó nem tudná, hogy éppen melyik rendszer kér felhasználói nevet és jelszót.) Az AuthUserFile egy, a felhasználói neveket és a kódolt jelszavakat tartalmazó fájlra mutat. A jelszófájl a honlap gyökérkönyvtárán kívül is elhelyezkedhet, így az böngészőn keresztül elérhetetlen. A jelszófájl elkészítéséhez vagy szerkesztéséhez használjuk a htpasswd programot. Ez alapértelmezés szerint a /usr/local/apache/bin-be kerül. A require valid-user hatására a felhasználói név, vagy az AuthUserFile-ban

található jelszó nélkül a felhasználó nem érheti el a könyvtárat.

Ha a `.htaccess` fájlban semmi hatása nem lenne, ellenőrizzük a `httpd.conf` fájlban lévő `AllowOverride` értéket. Ugyanis ez teszi lehetővé, hogy az Apache beállításait a `.htaccess` felülbírálhassa. Alapértelmezés szerint az Apache-kiszolgálók nem engedik meg a `.htaccess` fájlokban az `AuthConfig` típusú módosítók használatát. Ezt megváltoztathatjuk, ha az alábbi sorokat a `httpd.conf`-ba írjuk:

```
<Directory /usr/local/apache/mason/cm>
AllowOverride AuthConfig
</Directory>
```

Egy kisebb hibát találtam a telepített Mason-CM változatomban. A legtöbb összetevő `.html` kiterjesztés nélküli, ez nehézkessé vagy akár lehetetlenné is teheti az Apache számára a tartalom `text/html`-ként való azonosítását. Így, bár a Mason-CM összetevők HTML-formátumú kimenetet készítenek, a tartalmat a böngésző formátzatlan ASCII szövegnek ismerte fel. Tehát az Apache kis segítségére szorult. A `mod_perl content_type` eljárásával közvetlenül beállítottam a tartalom típusát. A következőket a `/cm` könyvtárba helyezve, a könyvtár összes dokumentumát automatikusan `text/html` típusúra állíthatjuk be:

```
<% $m->call_next %>

<%init>
$r->content_type("text/html");
</%init>
```

Az én `mason.pl` beállításfájlom figyelmen kívül hagy mindent, ami nem szövegfájl, így biztos lehetek benne, hogy a fentiek nem fognak véletlenségből JPEG és PNG képeket `text/html` típusúként felismerni.

Beállítás

A Mason-CM most már a helyére került, a működéshez azonban az Apache-nak néhány Perl modult be kell töltenie. A Mason beállításfájlba (amit én `mason.pl`-nek neveztem el, de a Mason leírása `handler.pl`-ként hivatkozik rá) szúrjuk be a következő Perl kódot:

```
@cx: {
    package HTML::Mason::Commands;
    use Fcntl;
    use MLDBM;
    use Image::Size;
    use URI::Escape;
    use File::PathConvert;
    use File::Copy;
    use File::Find;
    use IO::Handle;
    use IPC::Open2;
}
```

Miután az Apache és a Mason már tudja, hogy hol keresse a Mason-CM részeit, itt az ideje, hogy magát a Mason-CM-et is beállítsuk. Majdnem minden beállítást a `cmConfig` módosításával végezhetünk, mely a `/cm` könyvtárban található. A jelenlegi `cmConfig` fájl még a régebbi változatú Mason csatolóval készült; ez kissé zavaró lehet azok számára, akik a Mason 0.80-as változatát szokták meg. Például az indítórész neve `<%perl_init>`, nem pedig egyszerűen `<%init>`, és az egyik összetevő a másikat az `mc_comp` segítségével hívja, a `$m->comp` helyett. Az alkotórészek viszonylag könnyen felismerhetők és megérthetők bárki számára, aki használta már a Masont.

A `cmConfig` fájl elején beállítandó két legfontosabb változó a `$CM_HOME` és a `$CM_DATA`. Ezeket az eredeti `cmConfig` 25. sora határozza meg (ez a `<%perl_init>` rész elején található). Az első a Mason-CM telepítési könyvtárára, a második pedig az általa kezelt adatokat (zárolás, változatszám) tároló könyvtárra hivatkozik. Én így határoztam meg ezeket:

```
$CM_HOME = '/usr/local/apache/mason/cm';
$CM_DATA = '/usr/local/apache/cmdata';
```

A Mason-CM működéséhez mindkét könyvtárnak léteznie kell. Míg a `$CM_HOME` alapértelmezés szerint van beállítva (hiszen várhatóan a `cmContent` a `$CM_HOME` belsejében található), a `$CM_DATA` könyvtárat külön létre kell hoznunk.

Jegyezzük meg, hogy ez a könyvtár nem egyezik meg a Mason adatkönyvtárával – ez utóbbi általában a `/usr/local/apache/masondata`. A `$CM_HOME` és a `$CM_DATA` meghatározása után a `%cm_config` beállítására szolgáló rész következik. Itt változókat és azok értékeit találjuk. A legtöbb esetben az alapértelmezett beállítások megfelelőek, most csak azokat tárgyaljuk, amelyeket érdemes vagy szükséges módosítani.

Az admin kulcs a Mason-CM rendszergazdájának levélcímét tartalmazza. Ő felelős a törölt fájlok visszaállításáért, a zárolt fájlok feloldásáért és általában a tartalomkezelő vezérléséért. Alapértelmezés szerint ez `cm-admin-ra` van állítva, de bármit beállíthatunk.

Ágak meghatározása

A branches kulcshoz rendelt érték egy tömbhivatkozás, mely a változatkezelő rendszer *ágait* (branches) írja le. Az összes ágnak a `$CM_HOME` könyvtárban kell elhelyezkednie, így különbséget tehetünk az *alhelyek* (subsites) között. Például egy újság cikkei ágakra bonthatjuk, külön részt fenntartva a hírek, a sport és az üzleti témakörök számára. Minden ágat egy név azonosít, ezt pedig az ág tulajdonságai követik. Ha honlapunk egy ágat tartalmaz, melynek neve „Elso”, a branches kulcs valahogy így néz ki:

```
branches => [
    Elso => {
        path =>
        '/usr/local/apache/htdocs/staging/content',
        trg_from => 'staging',
        trg_to => 'production',
        components => 0
    }
]
```

A fenti ág a Mason-CM ágválasztójában (branch selector) is megjelenik, „Elso” néven és az `/usr/local/apache/htdocs/staging/content` alatti dokumentumokat irányítja. Győződjünk meg arról, hogy a könyvtár nem / karakterrel végződik, mert ebben az esetben a Mason-CM biztonsági hibával leáll.

A `trg_from` és a `trg_to` kulcsokat egyszerű behelyettesítésre használhatjuk; segítségével itt azt jelezzük, hogy a „staging” karakterláncot „production”-ra cseréljük le, és így másoljuk át a dokumentumokat a belső kiszolgálóról a külsőre. (A Mason-CM ezt a folyamatot „triggering”-nek nevezi.)

Így a tartalom kezdetben az `/usr/local/apache/htdocs/staging/content` könyvtárban van, majd az `/usr/local/apache/htdocs/production/content` könyvtárba kerül. Végül azt is jelezzük, hogy az ág HTML-kódot tartalmaz (és nem Mason-összetevőket): ehhez a `components` kulcsot kell 0-ra állítanunk.

Egy összetettebb webhelyen a következőképpen állíthatnánk be az ágakat:

```
branches => [
  News => {
    path => '/usr/local/ \
apache/htdocs/staging/news',
    trg_from => 'staging',
    trg_to => 'production',
    components => 1,
    hidden => 1
  },

  Business => {
    path => '/usr/local \
/apache/htdocs/staging/business',
    trg_from => 'staging',
    trg_to => 'production',
    components => 1,
    obj_dir => \
'/usr/local/apache/staging/obj',
    hidden => 1
  }
]
```

A fenti Mason-CM beállításnak két ága van, a News és a Business. Mivel a branches egy tömb, ezért az elemeket az eredeti sorrendjükben tárolja. Ez azt jelenti, hogy az ágkiválasztó a beírás sorrendjében jeleníti meg a fenti ágakat. Az ágak megjelenési sorrendjének módosításához cseréljük fel az ágak tömbhivatkozásait.

Ha az ágakat a fentiek alapján készítettük el, akkor az Apache beállításait átírhatjuk úgy, hogy minden /news-zal kezdődő címet /production/news-ra módosítson:

```
Alias /news
/usr/local/apache/htdocs/production/news
```

Így a belső (staging) kiszolgáló nem érhető el a böngészők segítségével. A webkiszolgálót azonban úgy is beállíthatjuk, hogy minden, a 8080-as (vagy bármilyen más) kapura érkező kérelem a belső kiszolgálóhoz kerüljön.

A hidden érték azt határozza meg, hogy alapértelmezés szerint az ág megjelenjen-e az ágválasztóban. Alap esetben minden ág megjelenik, és azok minden felhasználó számára elérhetők. Az ágak listáját bárki testre szabhatja a Mason-CM kezdőlap jobb felső sarkában található my.CM hivatkozás használatával, illetve bármelyik felhasználó hozzáadhat ágakat a menühöz és törölhet is abból. Az ágak elrejtésével az új felhasználók viszonylag tiszta képet kaphatnak a kezelőrendszer tartalmáról. Az Elso ággal ellentétben a News és a Business ág Mason összetevőket is tartalmaz. Ezek kimácsolása a külső kiszolgálóra abban különbözik a HTML-lapoknál megszokott módszertől, hogy a másolás előtt a Mason-CM megpróbálja lefordítani az összetevőket, és hibakezérést végez azokon. Így aztán egy sérült alkotórész nem okozhatja a külső kiszolgáló leállítását, legfeljebb a belsővel lehetnek bajok. Ha a lefordított Masonösszetevőket egy meghatározott könyvtárban kívánjuk tárolni, az **obj_dir** kulcsra lesz szükségünk.

A cmConfig fájl többféleképpen is átírható a Mason-CM-hez való együttműködéshez, de ha a \$CM_HOME-ot és a \$CM_DATA-t egyszer beállítottuk, máris használhatjuk a Mason-CM-et.

A kezdőlap

A Mason-CM felület főoldalának eléréséhez a böngészőbe a \$CM_HOME-ban megadott címet kell írunk. Nálam ez a `http://localhost/mason/cm/`.

Mivel ez a könyvtár jelszóval védett, ezért a felhasználói névre és a hozzá tartozó jelszóra is szükség lesz. A sikeres belépést követően

a Mason-CM kezdőlapot látjuk, amely körülbelül egy webalapú fájlböngésző. Segítségével az egyes ágak könyvtárai és alkönyvtárai között mozoghatunk, fájlokat nyithatunk meg olvasásra és írásra, illetve név vagy tartalom szerint kereshetünk a fájlok között.

A kezdőlapot mindig könnyen felismerjük: egy bűvész van rajta. Ezt a cmConfig-ban található juggler_src kulccsal tetszőleges képpel helyettesíthetjük. A képre kattintva mindig a Mason-CM kezdőoldalára jutunk vissza.

A kezdőlap jobb oldalán található az ágkiválasztó, mely a cmConfig-ban meghatározott ágakat sorolja fel. Az ágkiválasztó hivatkozásaira kattintva az ág külső kiszolgálóra másolását vezérelhetjük. Az éppen szerkesztett ág más háttérszínnel jelenik meg. Az aktuális könyvtár a képernyő közepén található, current directory fejléccel és alapértelmezés szerint kék háttérrel. Az aktuális könyvtár útvonalának minden összetevője az adott útvonalra mutató hivatkozás, így egyszerűen válthatunk könyvtárat az egérrel. Az alkönyvtárba lépéshez csak annak nevére kell kattintanunk. A képernyő közepén található szövegmező használatával új alkönyvtárt hozhatunk létre.

A current directory (jelenti könyvtár) vonal felett található a keresőrendszer. Nyilván nem én vagyok az egyetlen, aki eddig általában a **grep** vagy a **find** segítségével keresett fájlokat a webhelyen.

A Mason-CM mindkét programot magában foglalja, így még akár a Unixot nem ismerő felhasználóknak is lehetőségük nyílik az ágon belüli keresgélésre. A keresés a Perl szabványos kifejezéseit (regular expressions) is támogatja, tehát a fájlok között sokféle módon kereshetünk: név, de akár tartalom szerint is.

Azért legyünk óvatosak: nem mindegy, hogy mire keresünk rá.

A Mason-CM boldogan végigszalad fájlok ezreire egy-egy összetett szabványos kifejezés beírásakor, nem törődve azzal, hogy a művelet mennyi időt vesz igénybe.

A current directory vonal alatt az adott könyvtárban elérhető fájlok listája található. Minden fájlnál megjelenik a neve, az utolsó módosítás dátuma, az azt végrehajtó felhasználó, valamint a fájl jelenlegi állapota. Ez utóbbi lehet staging (ez azt jelenti, hogy csak a belső kiszolgálón létezik), prod (a fájl ugyanazon változata található mindkét kiszolgálón) és modified (a fájl mindkét kiszolgálón létezik, de megváltozott). A szövegmező és a create gomb használatával új fájl is létrehozhatunk. Ne tévesszük össze az alkönyvtár készítésére szolgáló gombot a fájlkészítő gombbal. Én módosítottam a gombfeliratokat (a dirTable és a fileTable elemeket), hogy a „Könyvtár létrehozása” és a „Fájl létrehozása” feliratok legyenek rajtuk.

Fájlok megtekintése és szerkesztése

A fájl tartalmának megtekintéséhez kattintsunk a nevére a táblázatban. A HTML-forráskód a böngésző felső részében jelenik meg. A lap alsó részén választhatunk a HTML leképezése, a 80. oszlop utáni sortörés (a nyers szöveg megjelenítése helyett), vagy a HTML-sorok számozása közül.

A fájlok szerkesztésére is van lehetőség a Mason-CM-ben, a program kezdetleges, de használható szerkesztőjének igénybevételével. Kattintsunk a fájl név melletti edit hivatkozásra, mire a fájl tartalmát mutató <textarea> elem jelenik meg. A <textarea> mezőbe szöveget gépelve módosíthatjuk a fájl tartalmát. A lap tetején található szövegmező használatával akár másolhatjuk, vagy át is nevezhetjük a fájlt. Ennél a szerkesztőnél egyszerűbbet nemigen készíthettek volna a fejlesztők: a kurzor mozgatását egy alig működő, Emacs-szerű billentyűkezeléssel oldották meg. A legfontosabb, hogy könnyen és gyorsan végezhetünk módosításokat.

A szerkesztőképernyőről számos lehetőség közül választhatunk:

- A save gomb frissíti a fájlt, és visszatér a szerkesztőképernyőhöz.
- A save and exit gomb menti a fájlt és visszatér a Mason-CM kezdőoldalára.

- A save and render gomb leképezi a HTML-fájlt, így a szerkesztett összetevő előképét is megtekinthetjük.
- A képernyő alján található redraw segítségével átméretezhetjük a szerkesztő ablakát.

A Mason-CM zárolja a szerkesztett fájlokat azért, hogy egyszerre csak egy felhasználó férhessen hozzá. A szerkesztett fájl a kezdőlap tetején található vörös téglalapban jelenik meg. Itt csak a megnyitott fájlokra mutató hivatkozások láthatók, ezekre kattintva azonnal szerkeszthetjük a fájlt, illetve az unlock lapon feloldhatjuk a zárolást. Ha egy másik felhasználó által éppen szerkesztett fájlt próbálunk megnyitni, a Mason-CM nem jeleníti meg a szerkesztőt. Ha a fájl zárolását feloldjuk (unlock), a többi felhasználó ismét hozzáférhet.

A fájlok másolása a külső kiszolgálóra

Ha úgy tűnik, hogy a fájl helyesen működik a belső kiszolgálón, akkor átmásolhatjuk a külsőre. Ennek végrehajtásához válasszunk ki egy vagy több fájlt a könyvtárból, a fájltáblákat bal oldalán található jelölőnégyzetekkel. Ezután kattintsunk a lap alján lévő trigger gombra, ekkor a fájlokat a külső kiszolgálóra másoltuk, s így azonnal a webhely friss (current) másolatává válnak.

A könyvtárban található összes fájlt egyszerre frissíthetjük a lap alján, közvetlenül a trigger gomb mellett található check all négyzet bejelölésével. Ez különösen akkor hasznos, ha új könyvtárat hoztunk létre, és egyszerre szeretnénk használni az összes elemet.

Ha egy fájl külső változatát módosítjuk, akkor a másolás fordítva is végbemegy („reverse-trigger”), azaz a belső kiszolgálóról a külsőre kerül. Ez meglehetősen veszélyes művelet, ezért a Mason-CM a művelet engedélyezése előtt rákérdez.

Helyesírás-ellenőrzés

Ha az alapvető Mason-CM szolgáltatások már működnek, valószínűleg néhány további képességet is szívesen kipróbálnánk. Talán a legérdekesebb lehetőség a helyesírás-ellenőrzés. Ez egy Mason összetevő, mely az ispell segítségével ellenőrzi a dokumentum helyesírását (sajnos csak angol nyelven). A Mason helyesírás-ellenőrző figyelmen kívül hagyja a HTML-tagokat, így nem kell például a href és társai szótárba illesztésével bajlódni. A helyesírás-ellenőrző indításához töröljük a megjegyzésjelet az ispell, main_dict és a supp_dict kulcsok sorából a %cm_config-ban:

```
ispell => '/usr/bin/ispell<\#180>',
main_dict => '/usr/lib/ispell/english.hash',
supp_dict => "$CM_DATA/suppDict",
```

Ha ezeket a kulcsokat egyszer meghatároztuk, a Mason-CM szerkesztőben megjelenik egy spell check jelölőnégyzet. Ennek bejelölése után a save, save and render vagy a save and exit használata során a szöveget ellenőrzi. A félregévelt szavakat egy kis JavaScript program segítségével javíthatjuk: választhatunk másik szót a szótárból, figyelmen kívül hagyhatjuk a hibát, vagy felvehetjük a szót a szótárba. A Mason-CM-rendszerben minden felhasználó ugyanazt a szótárat használja, tehát az egyik felhasználó által hozzáadott szavakat mindenki elérheti. Ez a módszer azonban veszélyt is hordoz magában, hiszen bármelyik felhasználóval megtörténhet, hogy véletlenül hibás szót vesz fel a szótárba. Legyünk óvatosak a javaslatokkal kapcsolatban!

Változatkezelés

A Mason-CM a változatkezeléshez szükséges RCS használatát is lehetővé teszi. Ehhez szükség van ahhoz, hogy a mason.pl elkészítse az Rcs modulok Image::Size, URI::Escape, és File::PathConvert részeit. Ezután a következő sorokat a cmConfig-ban meg kell

Kapcsolódó címek

A Mason honlapjáról tölthetjük le a Mason és a Mason-CM legutóbbi változatát. ➔ <http://www.masonhq.com/>.

Két hasznos cikk található a CVS és webes fejlesztés témakörében az alábbi címeken:

➔ <http://durak.org/cvswebsites/> és

➔ <http://www.daemonnews.org/199903/websites.html>

A Mason-CM futtatásához szükséges összes Perl modult letölthetjük a CPAN (Comprehensive Perl Archive Network) rendszeréről, mely tulajdonképpen FTP-helyek hálózata. Keressük meg a hozzánk legközelebb eső CPAN tükörkiszolgálót a ➔ <http://www.cpan.org/> címen, vagy használjuk a Perl minden változatában megtalálható CPAN.pm-et.

határozni, vagy ki kell törölni előlük a megjegyzésjelet:

```
rscs_bin => "/usr/bin",
rscs_files => "$CM_DATA/archive",
```

Az értékek beállítása után egy version label (változati címke) szövegmező jelenik meg a szerkesztőlap tetején. Ha mentéskor beírjuk a változati címkét, akkor az RCS automatikusan elindul, és a fájl régebbi változatát is megtartja.

A változatkezelés elindításakor a fájllistán a versions címke is megjelenik, ha rákattintunk, akkor a dokumentum történetét láthatjuk, a diff programot használhatjuk grafikus környezetben, és a régebbi változatokat is ellenőrizhetjük. A változatkezelés szinte nélkülözhetetlen a nagyobb webhelyeknél, mivel a hibák bármikor bekövetkezhetnek, és ilyen esetben kényelmesen visszaállíthatjuk a régebbi, de megbízható változatot.

A helyesírás-ellenőrzésen és az RCS-en kívül a Mason-CM jó néhány szolgáltatást tartalmaz még: a felhasználók feltölthetik a fájlokat HTTP-n és FTP-n keresztül, a rendszergazdák könyvtárként korlátozhatják a felhasználói hozzáférést. Mivel a Mason a Perl egy könnyen érthető változatában íródott, egyszerűen bővíthetjük képességeit (HTML-ellenőrzés másolás előtt stb.).

Összefoglalás

A Mason nagy segítségünkre lehet webhelyek kialakításakor, de a Mason-CM bizonyítja igazán az eszköz rugalmasságát. A Mason-CM megmutatja, hogy a Mason összetevőivel olyan eszközöket is készíthetünk, melyek nem közvetlenül a Weben keletkezett tartalomra vannak hatással. A sokféle Mason-CM eszköz változatossága meggyőzött arról, hogy szinte mindent megoldhatunk a csomag segítségével. Bár én valószínűleg ezután is a GNU Emacs szerkesztőt használom, több ügyfelem honlapján fogom alkalmazni a Mason-CM-et. Olyanokén is, akik Masont használnak a tartalom elkészítéséhez, illetve olyanokén is, akik egyszerűbb, kevésbé fejlett eszközökkel oldják meg a feladatot.



Reuven M. Lerner (reuven@lerner.co.il) egy internetes módszerekre szakosodott izraeli tanácsadó cég tulajdonosa. Éppen mostánában fejezi be (végre!) a *Core Perl* című könyvét, mely a Prentice-Hall kiadó gondozásában jelenik majd meg. Az ATF honlapja:

➔ <http://www.lerner.co.il/atf/>.

A Linux forradalmasítja a digitális hangrögzítést

Az otthoni hálózatok hamarosan korlátlan lehetőségeket hoznak a nappalinkba.

Az MP3 tényleg ugyanúgy a szemétdombra küldi a CD-ket, ahogyan azt tette a hangszalagokkal és a kazettákkal? Bizonyosan nem, ha az MP3-akat csak számítógépen, vagy egyórányi zenét rögzítő hordozható készüléken hallgathatjuk. Nyugalom, zeneimádók! Rengeteg új készüléket fejlesztenek manapság, és az MP3 hamarosan meghódít mindent, a hifitoronytól kezdve a sétálómagnóig. A Linux pedig kulcsszerepet játszhat e nemrég született, de máris hatalmas piacon.

Lemezmentesítsünk!

A zene túl régóta „röghöz kötött”: egy-egy dal meghallgatásához a jó öreg 120 milliméteres CD-t kell magunkkal „cipelnünk”. Addig nincs is különösebb baj, amíg csak egy korongról van szó, de az egész gyűjteményünk szállítása már kényelmetlen. Ráadásul a CD-írók elterjedéséig nem is készíthettünk egyéni válogatásokat, pedig a digitális zene csupán egyesek és nullák sorozatából áll. Az MP3-as formátum, a gyors modemek, valamint a nagyobb merevlemezek megjelenésével egyszerűbb egyik készülékről a másikra átvinni a dalokat. Ez a lehetőség azonban természetessé tette a (törvényes vagy törvénytelen) másolatok készítését. Mára már ezt a feladatot a PC végzi, hiszen ez az egyetlen olyan készülék a háztartásban, melynek hálózati kapcsolata és nagy háttértára van. Persze a PC sem tekinthető tökéletes

megoldásnak. A rendszer elindulása sok időt vesz igénybe, a felhasználói felület kezelését meg kell tanulni, a gép gyakran lefagy, ráadásul még a ventilátor is zúg. Az összes szobába egy PC-t rakni kicsit költséges multság lenne, ráadásul a gépek legtöbbször nincs is jó minőségű hangkimenet.

A megoldás egy házi zenehálózat létrehozása lehet. Először is szükségünk van egy zenei kiszolgálóra. Ez egy önálló készülék, mely bekapcsolás után azonnal használható. Merevlemezzel, Internet-kapcsolattal, CD-meghajtóval, tv-kimenettel, s a hang számára csúcsmínőségű ki- és bemenettel. A felhasználók letölthetik a zenéket az Internetről, de saját fájlokat is készíthetnek egy CD-ről vagy bármilyen külső hangforrásról. A tárolt zenéket természetesen bármikor meghallgathatjuk, de internetes rádióadásokat is élvezhetünk.

A rendszer természetesen Linuxon futna, nincs szükség Windows-megfelelőségre, a Linuxban megvan mindaz, ami alkalmasabb teszi – így a hálózatkezelés és az elegendő teljesítmény – az egyidejű letöltés, a beködolás és a zenehallgatás ellátására. Most adjunk ügyfeleket a hálózathoz. Ezek kéréseket küldenek a kiszolgálóhoz, és megszólaltatják az onnan vagy az Internetről érkező hangfolyamot. Így nincs szükség merevlemezre vagy CD-re, de a Linux e készülékek esetében is jó választás olcsóságának és hálózati támogatásának köszönhetően. Ahogyan jelenleg is több rádió- és tévékészülék van egy lakásban, úgy jövőbeli otthonainkat szintén számos, a zenei kiszolgálóhoz csatlakozó ügyféllel szerelhetjük fel.

Az új hálózat kialakításához többféle technológiát használhatunk, például a telefonvonalat (HPNA), az elektromos hálózatot (HomePlug), de vezeték nélküli adattovábbítással is próbálkozhatunk (802.11) Ezek bármelyikének segítségével megvalósítható a több készülékből álló otthoni zenei hálózat. A zenehallgatásnak ennél az új módjánál nincs szükség adathordozóra. A zene bitek sorozataként kerül minden olyan készülékbe, amely kapcsolódik a hálózathoz. Elméletileg még vidéki nyaralónkban is hallgathatunk otthoni zenét, feltéve, hogy ott is van internethozzáférés. A dalokat a sokféle hordozható készülék valamelyikére letölthetjük, így hálózati csatlakozás híján is hallgathatjuk azokat.

Az új készülékek

Az első cég, mely e jövőképet elfogadta és megvalósította, az S3 volt. A Rio sorozatú hordozható MP3-lejátszó az ő tulajdonukban van. Nemrégiben dobták piacra a Rio Receiver nevű készüléket, amely ügyfélként tud kapcsolódni a PC-hez, mint kiszolgálóhoz, a meglévő telefonkábelezés segítségével (a telefon használatának zavarása nélkül).

Ára 250 dollár körül mozog, emellett a PC-ben telepíteni kell egy 49 dolláros, telefonvonalat használó hálózati kártyát. Az S3 egy zenei kiszolgálót is fejleszt, de még semmit sem hoztak nyilvánosságra róla. Az új készüléket mind az S3, mind a Dell Computer árusítja.

Más kisebb gyártók, így

a Lansonics, a Lydstrom és a Zapmedia is hirdetik digitális zenei rendszereiket, változatos szintű szolgáltatásokkal és hálózati képességekkel. Azonban az S3 az egyetlen olyan cég, amely mind kiszolgálókat, mind olcsó ügyféleszközöket kíván gyártani. A Rio volt az első hordozható MP3-lejátszó, s piaci előnyétől máig sem fosztotta meg senki. Az S3 pedig már az otthoni hifirendszerek felé tapogatózik, ugyanis csak minden harmadik amerikai használja az Internetet, zenét azonban szinte mindenki hallgat. A digitális zene (az MP3 és más formátumok közvetítésével) lehetne az az alkalmazás, amely elősegíthetné az otthoni hálózatok kiépülését. Mint a legegyszerűbb nem-PC eszköz hálózathoz csatlakoztatásához, a Linuxnak számos előnye van ezen az új piacon is.

- ☞ <http://www.riohome.com/>
- ☞ <http://www.mp3-tech.org/>
- ☞ <http://www.mp3.com/>
- ☞ <http://bladeenc.mp3.no/>



Linley Gwennap (linleyg@linleygroup.com) a Linley Group elemző cég alapítója és vezető elemzője a kaliforniai Mt. View-ban.

Linux a tévéállványon

Az Indrema forradalmian új, csúcsteljesítményű konzolja beágyazott Linuxszal és nyílt forrású programfejlesztéssel készül a piac meghódítására.

Az Indrema honlapja szerint „a Linux útja a tévé felé vezet”. Tovább olvasgatva megtudtam, hogy az Indrema rendszer egy „forradalmian új termék, ami a forradalmi operációs rendszerre, a Linuxra épül”. Arra gondoltam, ismét egy kikapcsolódást szolgáló, egyszerűen használható készülékről van szó a „Csak kapcsoljuk be és dőlünk hátra!” szemlélet jegyében.

„Ez nem egy asztali Linux-rendszer; ez egy tévés Linux, a játékosok számára, a tökéletes szórakozásért. Kicsomagolva és a tévéhez kapcsolva azonnal játszhatunk kedvenc 3D-s játékkal, nagy sebességgel böngészhetünk az Interneten, tévét is nézhetünk vagy akár MP3-at hallgathatunk.”

„Ez nem egy átlagos szórakoztatóelektronikai cég...” – gondoltam. Az Indrema a jelek szerint végleg elkötelezte magát a Linux mellett, s ezt komolyan is gondolja. A fejekben természetesen azonnal kérdések fogalmazódtak meg: kik ezek a srácok? Mire készülnek? Hogyan viszonyulnak a Linux szellemiségét meghatározó nyílt forráskódhoz?

Önmagam és az olvasók megnyugtatása végett gyorsan felhívtam az Indrema ügyvezető igazgatóját, *John Gildredet* és időpontot egyeztettem egy kis internetes csevegéshez, amelyre egy hét múlva sort is kerítettünk. Beszélgettünk a tervekről, a termékletekről és természetesen a nyílt forrás elméletéről is.

Mi az az Indrema?

„Tulajdonképpen egy éjszakába nyúló Quake-partin vetődött fel az ötlet, hogy mi lenne, ha létrehoznánk egy olyan nyílt forrású játékkonfiguratort, amelynek segítségével egy *John Carmack*-típusú srác (ő a Quake szülőatyja), vagy a következő játékos megalkodója jóval egyszerűbben hozzáférhetne a játék- és konzolpiachoz?” – tekintett vissza Gildred a másfél évvel azelőtti kezdetekre.

Gildred és az Indrema többi alapítója megfigyelték, hogy a PC-s játékok fejlesztése sokkal gyorsabban halad, mint konzolos társaiké. Szerintük ennek az az oka, hogy egy magányos fejlesztőnek igen nehéz betömnie a konzolpiacra a technikai és anyagi korlátok miatt. Ezért úgy döntöttek, hogy



Az IES doboza egy teljes Linuxot rejt magában

új konzolt készítenek. A tervezést teljesen az alapokról kezdték, valamint különös gondot fordítottak arra, hogy a környezet a szerényebb lehetőségekkel bíró fejlesztőket se zárja ki a versenyből. Abban is egyetértettek, hogy az új rendszer alapját a nyílt forráskód, a nyílt API-k és a Linux fogja képezni.

A tervet számos otthoni fejlesztő azonnal támogatta, s az egészben az a legszebb, hogy a szükséges technológiai háttér már régóta elérhető volt mindenki számára.

Milyen lesz a készülék?

„Az Indrema Entertainment System (IES) doboza kívülről egy csúcscsintű videolejátszóra emlékeztet, senki sem mondaná meg, hogy Linux rejlik benne. Ha bekapcsoljuk, úgy működik, mint egy hagyományos szórakoztató elektronikai készülék. Nézhetünk tévét, ahogy máskor, de lejátszhatjuk a kedvenc MP3-as zenéinket, vagy böngészhetünk az Interneten is.” – mondta Gildred. A készülék széles sávú hozzáférést kínál a beépített 10/100 megabites hálózati illesztőn, vagy telefonos kapcsolaton keresztül. Található mellette egy játékegyeztető és egy előre telepített játék, amellyel bekapcsolás után azonnal játszhatunk. További játékokat a beépített DVD-meghajtó segítségével, illetve az online vásárlást követően az

Internetről letöltve telepíthetünk.

„Akkor ez most egy több szolgáltatást biztosító set-top box?” – kérdeztem naivan.

„A set-top box kifejezés használatától világgá szaladunk!” – hangzott az ingerült válasz. Elmagyarázta, hogy nem szeretné, ha ebbe a kategóriába sorolnák be terméket, hiszen az Indrema elsősorban egy játékgép. „A játékok nagyon gyorsan futnak rajta, valamint nyílt fejlesztői környezetet valósít meg.”

Persze az Indrema nem csak ennyire képes. A legjobb lenne talán multimédiás Linux számítógépnek nevezni. „Sok olyan alkalmazás fog megjelenni rá, amelyek semmiképpen nem férnek bele a játék kategóriába, hiszen a hardvernek sokkal komolyabb hangbéli és grafikai képességei vannak.” Ezek alatt a HDTV-t (nagy felbontású, tökéletes minőségű televíziókép), a mozgóképek letöltését és lejátszását, valamint a céggel együttműködő tévétársaságok műsorainak vételét (az Interneten keresztül) kell érteni. Ja, és természetesen ott van még a Personal TV nevű szolgáltatás, ez azt jelenti, hogy tetszés szerint tölthetünk le és nézhetünk meg tévéműsorokat. Nem minden IES-képesség lesz elérhető a legolcsóbb játékkonzolban, például a Personal TV-t csak felárért építik be csúcskategóriás modellbe.

Mi rejtőzik a dobozban?

Kukkantsunk most bele az Indrema karcsú dobozába. A számítógép lelke egy 600 MHz-es x86-kompatibilis processzor, valamint egy dedikált és átalakított grafikus csövezeték. A grafikus alrendszer MP3 és AC3 kódoló és visszaalakító elektronikát, digitális-analóg átalakítót, valamint az nVidia grafikus processzorát (GPU) tartalmazza.

A memória 64 MB és nem bővíthető, a memlemez azonban legalább 8, de legfeljebb 50 GB lehet. Természetesen megtaláljuk a 10/100-as Ethernet csatlót is a készüléken. A játékvezérlők és egyéb külső eszközök bekötését USB csatlókkal képzelték el a tervezők (az alapmodellen 4 ilyen találunk). A tévét a hagyományos kompozit, az S-videó vagy a HDTV-kimeneten csatlakoztathatjuk a készülékre. Kompozit és S-videó bemenet szintén található a gépen. Ami a hangot illeti, van sztereó analóg ki- és bemenete, valamint digitális kimenete is. Hogy milyen processzort fog használni az Indrema, az IES-ben egyelőre még eldöntetlen kérdés. Gildred szerint egy még nem kibocsátott következő generációs Intel vagy AMD processzort fognak használni: „valami újat, nagyon gyorsat, kifejezetten a céljainknak megfelelőt”.

A rendszer elektronikájában csak a grafikus modult lehet majd cserélni, ugyanis az nVidia GPU és a videomemória egy kártyára van építve, ez a készülék hátuljában található foglalatban kapott helyet, s ide kerülhet később az új kártya.

A négy USB csatló jelent a hardveres bővítés legegyszerűbb módját. Mivel az IES egy teljes értékű, multimédiás, Internetre csatlakoztatható Linux számítógép, nem meglepő, hogy a kiegészítők széles választéka használható hozzá, nem csupán a játékvezérlők. A komolyabb modellek hátlapján további USB csatlók is találhatóak.

Mitől olyan különleges ez a készülék?

Természetesen a programoknak köszönhetően. Három szoftverretről beszélhetünk:

- a nyílt forrású Linux operációs rendszerről (DV Linux);
- az IES hardverét kiszolgáló saját alkalmazásokról (ezek csak binárisan elérhetőek, forráskódként nem);
- a különböző felhasználói alkalmazásokról (játékok, böngészők stb.).

Az Indrema honlapja:

➔ <http://www.indrema.com>

Set-top az Interneten:

➔ <http://www.set-top.com>

➔ <http://ruel.net/top/box.list.htm>



Az IDN honlapja

Az Indrema honlapjáról a készülékhez tartozó fejlesztői környezetet (SDK) is letölthetjük: ezek az OpenGL, az OpenAL, az OpenStream és az Extrema. Az első három nyílt forráskódú, az Extrema pedig az Indrema saját alkalmazásait foglalja magában, ez csak bináris formában érhető el.

Miért épp nyílt forráskód? Miért éppen a Linux?

Mindamellett, hogy az Indrema bizonyos programelemek tulajdonjogát főntartja, elkötelezte magát a nyílt forráskódú fejlesztés mellett. A cég számos nyílt forrású projektben vesz részt, ilyenek például a Linux rendszermag, valamint az OpenAL és a Mesa 3D fejlesztése (ez utóbbi az OpenGL nyílt forrású változata). Emellett a vállalat egy új, nyílt forráskódú videofolyam-rendszert is fejleszt, ennek OpenStream a neve.

Az Indrema egyik legfontosabb nagyszabású terve a tévéhez csatlakoztatható Linux konzolok operációs rendszere, a DV Linux létrehozása.

„Azt szeretnénk, hogy a DV Linux szabvánnyá váljon, s a felhasználók is felismerjék, ez valóban nyílt forrású. Így szabvánnyá tehetünk egy játékgepfelületet, megkönnyítve ezzel az összes játékfejleszt-

tő dolgát. Lépéseket teszünk azért, hogy a piac további fontos szereplői is csatlakozzanak hozzánk. A DV Linux számunkra a legfontosabb eszköz az IES felület és saját fejlesztői környezetünk teljesen nyílttá tételéhez.

A sorozatgyártás

A korábbi bejelentések szerint már idén karácsonyra a polcokra került volna a készülék, jelenlegi tervek szerint a jövő év tavaszán jelenik meg. Az alapszintű IES fogyasztói ára – a pletykák szerint – 299 dollár lesz. A fejlesztőknek természetesen nem kell addig várniuk: az SDK-t már az letölthetik az Indrema honlapjáról.



Rick Lehrbaum

(rick@linuxdevices.com) a beágyazott Linux világával foglalkozó LinuxDevices.com portál

létrehozója. Ez nemrég vált a ZDNet Linux Resource Center részévé. Rick 1979 óta dolgozik beágyazott rendszerekkel. Az Ampro Computers társalapítója, illetve alapítója a PC/104-nek, és az Embedded Linux Consortium indításában is segédkezett.

© Kiskapu Kft. Minden jog fenntartva

RTcmix a Linuxon (1. rész)

A valós idejű hangképzéssel foglalkozó sorozat első részében az RTcmix történetével és alapjaival ismerkedhetünk meg.

Valós idejű hangképzést és -feldolgozást tesz lehetővé az RTcmix nevű programcsomag. Tulajdonképpen ez nem más, mint a Cmix nyelv valós idejű változata. A program futását egy MINC (MINC Is Not C) nevű parancsnnyelvvvel vezérelhetjük. Ez már akkor is nyílt forrású volt, amikor e kifejezés csak kevesek számára jelentett valamit.

Története

Az elmúlt húsz év alatt több vállalkozó kedvű programozó is hozzátett valamicskét az RTcmixhez. Írásomban megkísérlem minden érintett nevét felsorolni, ám ha valakit mégis kihagytam volna, a csomagban található AUTHORS fájl tartalmazza mindet. A Cmix a MIX-ből fejlődött ki, ezt húszsávú keverőprogramból írta *Paul Lansky* a princetoni egyetemen. A VMS-alapú IBM nagygépekre tervezett eredeti programot 1978-ban Fortran nyelven írták. Paul a hangképzési képességekkel bővített programot 1983–84-ben átírta BSD 2.9-es Unixra. 1985-ben a fejlesztés egy Ultrixot futtató DEC MicroVAX-ra költözött át. A Cmixet a továbbiakban NeXT, illetve más Unix rendszereken lehetett elérni, és a legtöbb felhasználója az egyetemi körökből került ki. Már akkor is főleg zeneszerkesztésre és zenei kutatásokra használták. 1995-ben *Doug Scott* és *Paul Lansky* az első valós idejű változatot átírta SGI-re. Én 1993-ban, a Columbia egyetem hallgatójaként találkoztam először a program linuxos változatával, amely még Soxot használt a fájlkezelés megkönnyítésére. Azok voltak csak a szép idők... 1995 végén *Brad Garton*, az egyetem számítógépes zenetanszékének (CMC) vezetője és jómagam is készítettünk egy valós idejű változatot, ütemezővel együtt. Az eredeti RTcmix IRIX alatt futott. Átültetése Linuxra annyit jelentett, hogy az IRIX hangkezelő rutinjait OSS (Open Sound System) eljárásokra cseréltük le. Az első IRIX-változatot az 1997-es International Computer Music Conference keretein belül mutatták be. Röviddel ezután a CMC másik vezetője, *Luke DuBois* a TCP csatlóadatok írását megkönnyítő eljárásokat épített bele. *Doug Scott* a hangszerek dinamikus, futásidőben történő betöltését már 1997-ben megvalósította. *John Gibson* (*Bill Schottstaedt* sndlib-jének felhasználásával) különböző hangfájl-formátumok olvasásának és írásának lehetőségével bővítette a programot. *Johnnal* nemrégiben

kidolgoztunk két újítást: a hangadatok belső „buszon” történő továbbítását, illetve a többcsatornás hang támogatását. Munkánk eredményét a SEAMUS (ez egy számítógépes zenével foglalkozó társaság) idei gyűlésén mutattuk be. További terveink között szerepel a hang-

1. lista WAVETABLE_REVERBIT_1.sco

```
/* WAVETABLE_REVERBIT_1.sco */
/* Megmutatja, hogyan illeszthetjük be a visszhanghatást bármilyen
   kottába. */
rtsetparams(44100, 2)
load("WAVETABLE")
load("REVERBIT")
/* Ha fájlba akarunk írni, töröljük a következő két sorból a
   megjegyzésjeleket. */
/* set_option("CLOBBER_ON") */
/* rtoutput("wavetable_reverbit1.aiff") */
/* A WAVETABLE kimenete az aux 0-1 buszra érkezik, ezt olvassa be a
   REVERBIT, mely a feldolgozott adatokat a hangkártyára küldi. */
bus_config("WAVETABLE", "aux 0-1 out")
bus_config("REVERBIT", "aux 0-1 in", "out 0-1")
/* A WAVETABLE hurok és a REVERBIT hívás hosszát vezérli. */
totdur = 6
reset(2000) /* enélkül a WAVETABLE rövid hangjai kattognának */
/*-----*/
setline(0,0, 1,1, 5,0)
makegen(2, 10, 10000, 1,.5,.3,.1)
dur = 1
amp = 10000
freq = 9.00
pctleft = 1
srand(3284)
dur = .08
for (st = 0; st < totdur; st = st + .3)
    WAVETABLE(st, dur, amp, freq, random())
freq = 7.10
for (st = 0; st < totdur; st = st + .45)
    WAVETABLE(st, dur, amp, freq, random())
/*-----*/
amp = 1
revtime = 1.5
revpct = .2
rtchandel = .15
cf = 200
setline(0,1, 1,1) /* felülbírálja a WAVETABLE burkológörbét */
/* az inskip értéke 0 KELL legyen, ha az aux buszról olvasunk */
REVERBIT(st=0, insk=0, totdur, amp, revtime, revpct, rtchandel, cf)
```

képzés paramétereinek valós idejű változtatása, különböző felhasználói felületek létrehozása, valamint egy megbízhatóbb csatlótámogatás megvalósítása.

A CVS (Concurrent Versions System) kiépítésével számos új együtt-

működés és fejlesztés jött létre. Az RTcmix magját a GPL (GNU Public License) alatt jelentettük meg, az egyes összetevők, például a hangszerek és hatások között azonban más feltételekkel terjeszthető elemeket is találhatunk.

Az RTcmix beszerzése és telepítése

Az RTcmix valósidejű képességeinek kihasználásához az OSS linux-os meghajtót kell telepítenünk, valamint beállítanunk (lásd még a További információk című részben). Az ALSA (Advance Linux Sound Architecture) támogatást nem próbáltuk ki ugyan, de szerepel terveink között. Ha az RTcmixet valósidejű hangfeldolgozásra szeretnénk használni, akkor a hangkártyánknak teljesen duplexnek kell lennie (ez azt jelenti, hogy a kártya egyszerre képes felvenni és lejátszani is). A cikk megszületésekor a legújabb változat száma még a 3.0.2 volt. Az egyes összetevőket külön csomagokban találjuk. Az RTcmix-3.0.2.tar.gz a magot és az ütemezőt, az insts.jg-3.0.2.tar.gz a John Gibson által írt hangszereket, az insts.mch-3.0.2.tar.gz a többcsatornás hangszereket, az insts.std-3.0.2.tar.gz pedig a hagyományos hangszereket tartalmazza.

Térjünk ki most röviden a program szóhasználatára: az RTcmix hangszer lehet hanghatás is (például egy DELAY), de hangképző is (például egy FM szintetizátor).

A hangszercsomagokat a maggal azonos könyvtárba csomagoljuk ki. Ha tehát a fő kódot a /usr/local/src-be helyezzük, akkor végeredményül a /usr/local/src/RTcmix-3.0.0-t kapjuk. Az insts.std.tgz pedig a /usr/local/src/RTcmix-3.0.0/ insts.std-3.0.0-ba fog kerülni. A hangszerek és az RTcmix-változatok következetességéért minden csomagnévben elhelyeztük annak változatszámát is. A fordítást egyszerűbbé tehetjük, ha a hosszú nevek helyett hivatkozásokat használunk, például:

```
ln -s insts.std-3.0.0. insts.std
```

Miután mindent kicsomagoltunk és a megfelelő hivatkozásokat is létrehoztuk, kezdődhet a fordítás. A beállításokat saját magunknak kell elvégeznünk, az RTcmix gyökérkönyvtárban lévő makefile.conf fájl módosításával. Csupán néhány sort kell megváltoztatnunk:

```
# Az RTcmix gyökérkönyvtára:
TOPDIR = /usr/local/src
```

```
# Az RTcmix könyvtárának neve:
CMIXDIR = $(TOPDIR)/RTcmix-3.0.0
```

```
# A hangszerekre mutató hivatkozásokat
# tartalmazó könyvtár:
LIBDESTDIR = $(CMIXDIR)/shlib
```

```
# Itt adhatunk a programhoz új hangszereket:
INST_DIRS = insts.base #insts.std insts.jg
# insts.mch
```

A két könyvtárnév, amit szinte biztos, hogy meg kell változtatnunk: a TOPDIR és az INST_DIRS, attól függően, hogy milyen hangszercsomagjaink vannak. A megfelelő változtatások elvégzése után az RTcmix könyvtárban adjuk ki a make, majd a make install parancsot. Ez mindent lefordít és egy CMIX nevű futtatható fájlt helyez el a TOPDIR/bin

könyvtárban, az osztott könyvtárakat pedig a TOPDIR/shlib-be rakja. Győződjünk meg arról, hogy a bin könyvtár a PATH része.

Zeneszerkesztés

A MINC parancsnyelvvvel egyszerűen és hatékonyan vezérelhetjük az RTcmixet. A MINC-parancsfájlokat a program kottafájként kezeli. Ha a C nyelvet alapjaiban ismerjük, a MINC elsajátítása sem lesz túl nehéz. A két legfőbb különbség, hogy a sorok végére nem kell pontosvesszőt tennünk, és függvények sincsenek. A MINC és az RTcmix

2. lista RT_REVERBIT.sco

```
/*RT_REVERBIT.sco */
/* Valósidejű visszhang a mikrofonbemenethez. */
set_option("full_duplex_on")
rtsetparams(44100, 2, 512)
load("REVERBIT")
rtinput("AUDIO", "MIC")
bus_config("REVERBIT", "in 0", "out 0-1")
dur = 60
amp = .9
revtime = 0.4
revpct = .8
rtchandel = .02
cf = 1000
setline(0,0, 1,1, 9,1, 10,0)
REVERBIT(st=0, insk=0, dur, amp, revtime, revpct, rtchandel, cf)
```

segítségével történő zeneszerkesztés többéves főiskolai tananyag, tehát kizárt, hogy egy cikk keretein belül elmagyarázzam. Ehelyett nézzünk végig néhányat a lehetőségeit bemutató kottafájlok közül. A kottafájl futtatásához küldjük el azt a CMIX-nek egy átirányítás segítségével:

```
CMIX < kottafájl
```

Ekkor a fájl az értelmezőbe kerül, ez az ütemező felé továbbítja. Valósidejű felhasználás esetén az RTcmix értelmezője egy TCP-csatolót figyel, s a bejövő adatokat értelmezi. Ebben a változatban az értelmező és az ütemező a linuxthreads könyvtárat használja. A kottafájlokat John Gibson készítette, és az RTcmix 3.0 csomagban is megtaláljuk ezeket. Az 1. lista egy viszonylag egyszerű, két RTcmix hangszert használó kottafájlt mutat be. Természetesen rengeteg ilyen példafájllal találkozhatunk a csomagban. A WAVETABLE egy hullámtábla-kereső, a REVERBIT pedig visszhangosítja a hangfolyamot. Ebben a példában a WAVETABLE kimenete a REVERBIT-hez kerül, majd innen a hangkártyához, illetve egy fájlba. Nézzük végig ezt a kottafájlt:

```
rsetparams(44100, 2)
```

E parancsnak minden RTcmix kottafájlból szerepelnie kell. Hatására a hangkártya sztereó, 44 100 Hz-es üzemmódba vált. Az elhagyható harmadik értékkel a gyorstár méretét állíthatjuk be, ezzel a késést szabályozhatjuk. Az alapértelmezett érték 8192 hangminta. Ezután a használni kívánt „hangszereket” kell megadnunk:

```
load("WAVETABLE")
load("REVERBIT")
```

Ahogy már említettük, az RTcmix támogatja a könyvtárak dinamikus betöltését. Ezek a parancsok futásidőben töltik be a hangszereket, így

azok kevesebb memóriát foglalnak el, mint ha mindegyiket egyszerre betöltenénk. A módszer segítségével több hangszert alkalmazó kottafájlokat is készíthetünk.

Az alábbi sorok megjegyzésben láthatók, tehát nem működnek. A jelek kitörlésével a hangot a kártyára vagy hangfájlba küldhetjük. A `set_option` parancs ki- és bekapcsolja az RTcmix bizonyos tulajdonságait. Például a `set_option("AUDIO_OFF")` hatására a hangkártyát nem használja a program; ez olyan esetben hasznos, ha a kottafájl túlságosan leterhelné a gépet. Ha a hangot az egyik helyről a másikra (például a hangkártyára vagy egy fájlba) szeretnénk irányítani, a

```
bus_config("WAVETABLE", "aux 0-1 out")
bus_config("REVERBIT", "aux 0-1 in", "out 0-1")
```

parancsokat használhatjuk.

Már említettem, hogy az RTcmix egyik legújabb szolgáltatása, hogy segítségével megváltoztathatjuk a hangfolyam útját. Hogy e bővítés összhangban maradjon a már jó ideje használt keverési eljárásokkal, bevezettük a „buszokat” (bus). Ezeknek a belső útvonalaknak „aux” a neve; olvasáskor bemenetként („in”), íráskor pedig kimenetként („out”) működnek. A buszok meghatározása a következőképpen történhet: `bus_type "szám" [in vagy out]`. Tehát az "out 0" egy kimeneti útvonal lesz (például egy hangszóró), az "in 0" pedig egy vonal- vagy mikrofonbemenet. Ezeket ne tévesz- szük össze az "aux 0 in" formával, ez ugyanis egy hangszer által olvasott belső busz meghatározása.

Tehát a fenti parancsok először a WAVETABLE kimenetét terelik a 0. és 1. számú auxra. Ezután a REVERBIT `bus_config`-ját úgy állítják be, hogy a fenti két kimenetet olvassa be, s a kimeneteket egy sztereó kártyára küldi (out 0-1). Egy hangszerhez több buszbeállítást is használhatunk, mindegyik a legutóbbi beállítást veszi alapul.

A fájl következő sora a `totdur` globális változót állítja be. A változó a fájl beolvasásakor kapják meg értéküket. Foglalt változók nincsenek, tehát bármilyen nevet használhatunk.

A hangképzés belső értékei (például az amplitúdógörbe) frissítéséhez a `reset` parancsot használjuk. A `reset (2000)` a másodpercenkénti 44 100 hangmintát 2000-szer frissíti, tehát átlagosan minden 20. minta frissül.

A következő két sor már titokzatosabb:

```
setline(0,0, 1,1, 5,0)
makegen(2, 10, 10000, 1,.5,.3,.1)
```

Az RTcmix, hasonlóan a hosszú évek alatt kifejlesztett programokhoz, a régebbi rendszerekkel való együttműködés megőrzéséért rengeteg régi szolgáltatást is tartalmaz. A generátorokat a `makegen` paranccsal határozhatjuk meg. Ezekkel a hangszer számos tulajdonságát megváltoztathatjuk: az amplitúdógörbét, a hullámformát, a vibratót stb.

A `setline` parancs is ide kapcsolódik: segítségével itt az amplitúdógörbét adjuk meg. Az idő-érték párokkal a 0. időpontban 0-t, az 1. időpontban 1-et, az 5.-ben pedig 0-t állítunk be. Az idő itt viszonylagos, tehát a görbe időtartama a hang („note”, lásd lejjebb) hosszától függ. A WAVETABLE hangszernek szüksége van valamilyen hullámformára a hang létrehozásához. Ezt az adatot a 2-es tárolóba (slot) helyezi, míg az 1-es tárolót a `setline` által létrehozott amplitúdógörbe tárolására tartja fenn. Ennek a `setline` parancsnak a helyére egy `makegen` parancsot is írhatnánk (gen24). A `gen10` a megfelelő tárolóba (most a 2-esbe) egy szinuszhullámot tölt be. A hullám tábla mérete 10 000 hangminta. A többi paraméter a harmóniasor következő elemeinek amplitúdóját határozza meg. Ennél bonyolultabb már nem lesz a dolog, már ami ezt a cikket illeti...

Néhány további globális változó meghatározása után alaphelyzetbe

Kapcsolódó címek

Az RTcmix linuxos változatának címe:

➔ <ftp://presto.music.virginia.edu/pub/Rtcmix>

A Virginia Center for Computer Music, az RTcmix fejlesztésének központja:

➔ <http://www.virginia.edu/~music/VaCenterCompMusic.html>

A virginiai egyetemen működő McIntire Department of Music tanszéken zeneszerzést tanulhatunk. A legújabb PhD programjuk kiváló lehetőséget jelent a számítógépes zenében komolyabban elmélyülni kívánók számára:

➔ <http://www.virginia.edu/~music>

A Princeton Sound Kitchen honlapján a MINC parancsokról és a Cmix felépítéséről olvashatunk:

➔ <http://www.music.princeton.edu/winham/cmix.html>

A Columbia Egyetem Computer Music Center részlegének honlapján rengeteg érdekes leírást és kottafájlt találhatunk:

➔ <http://www.music.columbia.edu/cmix/>

John Gibson honlapja:

➔ <http://www.people.virginia.edu/~jgg9c/musi540/index.html>

Példafájlokat és az RTcmixszel kapcsolatos adatokat találhatunk a következő címen is:

➔ <http://presto.music.virginia.edu/RTcmix/linuxjournal/>

A cincinnati egyetem honlapjáról különböző RTcmix kezelői felületeket tölthetünk le:

➔ <http://meowing.ccm.uc.edu/>

OSS:

➔ <http://www.opensound.com/>

Az RME Digi96 sorozatú hangkártyák:

➔ <http://www.rme-audio.com/>

Az ICMC és a SEAMUS honlapjai:

➔ <http://www.computermusic.org>

➔ <http://seamus.lsu.edu/>

állítjuk a véletlenszám-generátort a `srand(3284)` paranccsal.

A zenét a következő sorok hozzák létre:

```
for (st = 0; st < totdur; st = st + .45)
WAVETABLE(st, dur, amp, freq, random())
```

Ez egy egyszerű hurok, amely addig szól, amíg az `st` nagyobb, mint a korábban meghatározott `totdur`. A WAVETABLE paramétereire (csakúgy mint az RTcmix más szolgáltatásaira és hangszereire) p-mezőkkel hivatkozunk. Ezek a WAVETABLE esetében a következők:

- p0 = indulási idő
- p1 = hossz
- p2 = digitális amplitúdó (itt: 0 és 32 767 között vehet fel értéket)
- p3 = frekvencia (Hz)
- p4 = pan

Ha p4=0, akkor a teljes kimenet a fentebb található `bus_config` által meghatározott 0-ás csatornára kerül; ha p4=1, akkor pedig az 1-esre. 0,5 megadáásával a kimenet egyenlő mértékben oszlik el a csatornák között. Figyeljünk arra, hogy a `random()` egy 0 és 1 közötti

lebegőpontos értékkel tér vissza.

A kotta utolsó néhány sora az amplitúdógörbét a `setline(0,1,1,1)` paranccsal alaphelyzetbe állítja, mert a REVERBIT-hez egy kissé különböző görbére lesz szükség. Ezt a REVERBIT (`st=0, insk=0, totdur, amp, revtime, revpct, trchannel, cf`) segítségével hívhatjuk meg.

A REVERBIT p-mezői a következők:

- p0 = indulási idő
- p1 = az indulástól a bemenet olvasásának megkezdéséig eltelt idő (A `bus_config`-ok esetében 0-t használjunk, fájlból olvasáshoz pedig valamilyen más értéket.)
- p2 = hossz
- p3 = amplitúdószorzó (általában 0 és 1 közötti szám)
- p4 = a visszhang hossza
- p5 = a visszhang százalékos aránya
- p6 = a jobb oldali csatorna késleltetése (delay)
- p7 = az aluláteresztő szűrő levágási frekvenciája (Hz)
- p8 = DC blokkoló szűrő hozzárendelése (logikai érték, elhagyható).

Ennyi volna. A kotta futtatásához írjuk be az alábbi parancsot:

```
CMIX < WAVETABLE_REVERBIT_1.sco
```

A 2. lista zeneileg egyszerűbb példa, mégis egy rendkívül hasznos tulajdonságot mutat be. Segítségével a bejövő hangfolyamot valós időben dolgozhatjuk fel. A kétirányú adatforgalomra a kártyát a

```
set_option("full_duplex_on")
```

paranccsal készíthetjük fel, ezt mindenképpen az `rtsetparams` előtt kell lefuttatnunk, hiszen az néhány, az `rtsetparams` által használt jelzőt is beállít. Az előző példához hasonlóan 44 kHz-es sztereo üzemmódba lépünk, de a belső gyorstár méretét 512 hangmintára csökkentjük, a valósidejű adatfeldolgozás gyorsítása céljából.

Az `RTcmix` az `rtinput("AUDIO", "MIC")` parancs hatására a hangbemenetekről olvas. A `MIC` paraméter csak az IRIX-változatban szükséges. Hangfájl beolvasásához egyszerűen az `rtinput("fájlnev.aiff")` parancsot kell használnunk.

Összegzés

Az `RTcmix` egy rendkívül sokoldalú és könnyen használható, nyílt forráskódú programcsomag. A jelenlegi szolgáltatások számos programozó munkáját dicsérik, s a fejlődés még ma is tart. A Linux tökéletes platformként szolgál a fejlesztéséhez és a használatához egyaránt. A csúcsmínőségű hangkártyák linuxos támogatásának megjelenésével az `RTcmix` képességei az operációs rendszerrel együtt bővülnek. Itt a rengeteg szolgáltatásnak csupán töredékét mutathattam be, a sorozat további részeiben azonban a hangszerek készítésével és valósidejű vezérlésükkel is foglalkozunk. Izgalmas, hogy egy számítógép, egy operációs rendszer és egy programcsomag milyen hasznos segítőtársa lehet egy zenésznek.

David Topper (topper@virginia.edu)

a virginiai egyetem zenei központjának munkatársa.

A Linux azóta kedvenc operációs rendszere, amióta letöltötte a 40 lemeznyi Slackware-t, az 1.0.9-es rendszerrel együtt. Makacsul bízik abban, hogy a Linuxhoz hasonló nyílt forrású rendszerekkel a számítógép olyan fontos eszközzé válhat korunk emberének, mint a távcső az ókori csillagászoknak. Honlapja az alábbi címen érhető el:

☞ <http://www.people.virginia.edu/~djt7p>



Kicsi a bors, de erős

Apró programok sokszor többórás munkától kímélhetnek meg.

Egyszer, régebben nyomon követtem a Linux-rendszer-mag levezetési lista egyik vitáját, ahol arról vitatkoztak, hogy miért nem alkalmas a Linux otthoni, általános munkavégzésre, játékra (ott ugyanis senki sem hitt ebben). Nem vettem részt a vitában, mert úgy gondoltam, hogy az egész téma nem illik az amúgy is túlerőltetett rendszer-maglistára, ennek ellenére nem értek egyet a levelezőkkel. A vita alapját az képezte, hogy a Linuxból még sok minden hiányzik, például az egyszerű ablakkezelés, bizonyos alkalmazások stb. Nos, éppen nemrég telepítettem Linuxot egy iskolai gépterem összes gépére (33 munkaállomásra volt szó), és úgy tűnik, hogy van igény a munka folytatására. Azt tapasztaltam, hogy egy megfelelően telepített és beállított Linux a diákok és a tanárok számára egyaránt könnyebben használható, mint bármilyen eddig látott rendszer. A titok az egyszerűségben rejlik, hiszen minden alkalmazásnak megtalálhatjuk a linuxos megfelelőjét (talán a Quicken az egyetlen kivétel). Bár a Calderát telepítettem (de bármelyik Linux-változat ugyanolyan jó), ennek ellenére nem a KDE-t, a Caldera alapértelmezett grafikus felületét használtam. Csupán annyi ügyeskedésre volt szükség, hogy a kezdő felhasználók számára ikonokat kapcsoljunk néhány egyszerűbb parancsállományhoz. A diákok még nem találják a játékokat, de szerintem erre sem kell sokáig várni... Mindenki tökéletesen elégedett, és szó esett a többi terem gépeinek átalakításáról is. Tehát csak ne becsljük le a Linuxot mint otthoni rendszert! Úgy vettem észre, ha a kezdő felhasználó egy jól behangolt Linux elé ül le, sokkal elégedettebb vele, mint az összes többi eddig használt rendszerrel. És ők valószínűleg egy Microsoft operációs rendszert sem lennének képesek telepíteni. Egyszerűen azzal dolgoznak, ami a gépen van, én pedig egy másik, jobban használható megoldást kínáltam számukra. Nem is rossz egy ilyen „partizán” operációs rendszertől...

Checkout

Egyszerű rendszer, mely a munkahelyi jelenléti íveket helyettesítheti. A nézetekkel részlegenként ellenőrizhetjük, hogy épp ki dolgozik, ki tartózkodik „házon kívül”, illetve utóbbi esetben a távolmaradás okát is megtudhatjuk (ha az illető volt szíves bepötyögni ezt). Akinek nem tetszik az egyszerű kezelőfelület, annak csupán a HTML-kódba kell itt-ott belepiszkálnia. A program böngészőt, PHP/MySQL támogatással rendelkező webkiszolgálót igényel és a MySQL-t.

☞ <http://www.draenor.org/checkout/>

sgalaAlive

Bizonyos folyamatokat állandóan futtatni szeretnénk? Nem túl vidám dolog, ha az FTP- vagy webkiszolgáló egyszer csak leáll. Ez a Perl parancsfájl a szükséges szolgáltatások megfelelő működését igyekszik felügyelni, és levélben értesít bennünket, ha valamelyiket újra kell indítani. A program telepítése gyerekjáték. Használatához a Perl és a cron szükséges.

☞ <http://www.sgala.com/>

Készítsünk saját indítólemezt!

Nyilván mindenki látott már olyan indítólemezt, mely összeomlás esetén használatos. Még az is lehet, hogy ott lapul egy a polcunkon. Vajon nem lenne jobb egy olyan, amely saját kedvenc segédprogramjainkat tartalmazza, és nem mások ízlése és kívánságai alapján lett összeállítva? Nosza, készítsünk magunknak egyet! A BYLD (Build Your Own Linux Disk) segíthet. Ha egy lemezen nem férne el minden, akár egyéni lemez-mérettel is dolgozhatunk. Az alapértelmezés már eleve 1680 kB, de nézzünk csak be a /dev könyvtárba, ahol még ennél nagyobbakat (1760,

sőt, 1840 kB!) is találunk. A programnak a bash héjra, néhány alapvető rendszerhívásra és a rendszer-mag forráskódjára van szüksége.

☞ <http://byld.sourceforge.net/>

Pronto Mail

A CSCMail készítői úgy döntöttek, hogy a fejlesztést C nyelven folytatják, ezzel egy időben a Perl-hívó felhasználók egy csoportja pedig a Perl felé haladt tovább. A Pronto egyik előnye az egyszerű telepítés. Ha a CSV kivételével mindent (MySQL, PostgreSQL, mSQL stb.) használunk, akkor egy üres adatbázist kell készítenünk a felhasználó nevével és a rá vonatkozó jogosultságokkal. A Pronto letöltés után automatikusan telepíti magát, a kezelőfelület tetszetős, a program megbíz-

hatóan működik. A Perlre és az alábbi Perl-modulokra van szükségünk: Gtk::XmHTML, Date::Manip, DBI, Text::CSV_XS, SQL::Statement, DBD::CSV, MIME::Base64, HTML::Parser, IO::Wrap, MIME::Parser, Mail::Header, MIME::Types, URI::URL, IO::Socket, Lingua::Spell.

☞ <http://www.muhi.net/pronto/>

passwdgen

Jómagam számos jelszókészítő programot használtam már, eddig a makepasswd volt a kedvencem. A passwdgen azonban rendelkezik néhány új szolgáltatással: a jelszavak lehetnek csak nagy-, illetve kisbetűsek, állhatnak számokból vagy nyomtatható karakterekből stb. A tulajdonságokat keverhetjük is egymással, így egész összetett jelszavakat is készíthetünk. Olyan jelszavakat is gyárthatunk, melyek beírásához elég csak a jobb vagy bal kezünket használnunk (ez azonban csak a QWERTY kiosztású billentyűzeteken működik). A program igényel glIBC-t.

☞ <http://members.home.com/denisl/passwdgen>

Simple Network TOP

Sok gépből álló hálózatot kell felügyelnünk? Csupán arra vagyunk kíváncsiak, hogy mely gépek működnek és melyek nem, s az egyes szolgáltatások állapota nem érdekes számunkra? Ez a segédprogram pontosan ezt a feladatot látja el, beállítása pedig egyszerű és gyorsan elvégezhető. Használat során a program egy konzolon fut, letiltott parancsbillentyűkkel. A kimenetet HTML-formátumban is megjeleníthetjük. Az sntop az fping segítségével ellenőrzi a kapcsolatokat, így a hálózatot szinte egyáltalán nem terheli. Az sntopnak a glIBC-re van szüksége.

☞ <http://sntop.sourceforge.net/>

folytatás a 80. oldalon



Figyelemreméltó teljesítmény, hogy három éve verhetetlen a bajnokságokban, de a hirdetésben nem ilyen típusú tapasztalatra gondoltunk.

Régi idők ízei

Régi kedvenceinket is viszontláthatjuk a Linux jóvoltából.

François, érted már a Linux konyhaművészetét? Ezen a lemezen egy teljes Linux-csomag található, segítségével egy 386-osból is tűzfalat vagy átjárót készíthetünk. Mais oui, erről a *Linux Journal* honlapján olvastam, a rendszergazdáknak szóló rovatban. Óh, François, néha-néha elmerengve visszavágyom a kisméretű, gyors alkalmazások idejébe, abba az időbe, amikor még a programok néhány száz bájt memóriával, pár ezreléknyi lefoglalt lemezterülettel csodákra voltak képesek. A Linuxszal főzőcskézve ez az időutazás is könnyedén megvalósítható. Hiszen régen sem minden a hatékonyságról szólt, nem?

```

20 PRINT "ENTER THE NUMBER OF YEARS"
30 INPUT NUMYEARS
40 PRINT "ENTER THE INTEREST RATE"
50 INPUT THERATE
55 TOTALAMT = AMOUNT * (1 + (THERATE / 1
60)) ^ NUMYEARS
65 PRINT "TOTAL IS ":TOTALAMOUNT
70 PRINT TOTALAMONT$
100 DUCKS = 5
110 DUCKSQUARED = DUCKS ^ 2
120 PRINT DUCKSQUARED
130 PRINT DUCKSQUARED * 5
140 PRINT (1 + (DUCKS / 100))
READY.
RUN
ENTER AN AMOUNT
? 100
ENTER THE NUMBER OF YEARS
? 12
ENTER THE INTEREST RATE
? 5
?SYNTAX ERROR IN 55
READY.
  
```

100%, 50 fps

1. kép Marcel Basic-tudása egy kissé megkopott

Oui, François. Qu'est-ce que tu dit? Ah, mes amis! Bocsássanak meg nekem. Az álmodozás közben észre sem vettem, hogy megérkeztek. Jöjjenek csak be. François mindjárt hozza a vacsorát, én pedig a borról gondoskodom. Hm... mit is szolgáljak föl?

Néha szívesen használnánk újra azt a rendszert, amely bevezetett minket a számítógépek világába? Commodore 64, PET, Atari, Amiga, TRS-80 vagy akár az IBM-1130... Ha megdobbán a szívük valamelyik név hallatán, nem kell

mindjárt lélegzőgyógyászhoz szaladni: a nosztalgia fontos szerepet játszik érzelmeink között. A jó borok-

hoz hasonlóan ezekben a gépekben is volt valami különleges, valami megismételhetetlen, amire oly jól esik visszavágnyi.

A lap következő számának nagy részét alkatrészekkel foglalkozó cikkekkel töltjük meg, így ízelítőként „kotyvaszunk” mi is egy kis különlegességet. Ugyan a Linuxszal főzőcskézünk, az asztalra azonban olyan fogás kerül, ami nem kifejezetten a Linuxról szól...

S ha már a bor szóba került... François, hozd fel, légy szíves a '89-es Pomerolt a pincéből, a vendégeink már bizonyára szomjasak.

Emulátorokkal bizonyosan sokan találkoztak már a linuxosok közül is: a régi DOS alkalmazásokat a DOSEMU, a windowsos programokat pedig a WINE segítségével futtathatjuk – bár tudom, hogy a WINE

nem emulátor („WINE” = WINE Is Not an Emulator). Ha pedig tökéletes emulációt szeretnénk, akkor virtuális gépet kell létrehozunk a Linuxon belül. Erre a VMWare (<http://www.vmware.com/>) is képes, segítségével bármely Windows-változatot futtathatjuk a Linuxon belül. Az emulátorok ismét ráirányíthatják a figyelmet a régen elfeledett rendszerekre.

Azt azonban soha ne feledjük, hogy az itt leírt termékek nagy része a szerzői jogvédelem alá esik. Néhány emulátor futtatásához ugyanis az operációs rendszert vagy a ROM-ot tartalmazó fájlokra is szükségünk van. Ezek beszerzésének vagy elkészítésének ismertetése messze meghaladná e cikk kereteit, és azt hiszem, közben François-nak félóránként borért kellene szaladnia, hogy tökéletes legyen a összpontosítás :-). Az emulátorokkal foglalkozó honlapokon azonban rengeteg ilyen fájl található. Emellett arról se feledkezzünk meg, mindig is léteztek szabadon használható (public domain) osztályba tartozó programok, így senkinek sem kell szégyenkeznie amiatt, hogy lopott vagy feltört programmal játszik. Az általam leghasznosabbnak ítélt címetek összegyűjtöttem, és a cikk végén találom.

Az emulátorok beszerzése nagyon egyszerű: látogassunk el a honlapra, majd töltsük le a megfelelőt. A csomagok lefordítását és telepítését most hadd ne magyarázzam el, nyilván mindenki cselekedett már ilyet:

```

tar -xzvf emulator-1.01.tar.gz
cd emulator-1.01
./configure
make
make install
  
```

Először az egyik kedvencemről szeretnék beszélni. Egyszer nagyon-nagyon régen az Önök kedvenc séfje egy Commodore 64 büszke tulajdonosa volt, hogy az akkorra már a szekrényben pihenő PET-ről, meg a VIC20-ról ne is beszéljünk. Akinek e korszakból még maradt pár doboz lemeze a polcok mélyén, valamint a Mermaid Madnessszel eltöltött órákat visszasírja, annak nyugodt szívvel ajánlom a VICE (Versatile Commodore Emulator) nevű programot, amellyel máris négy gépet (C64, C128, VIC20, PET) emulálhatunk, ráadásként még hangjuk is van! A VICE Team a programot a GPL feltételei szerint terjeszti. Az összes Commodore-modell közül nekem legjobban a C64-hez húz

Tim Mannról csak annyit, hogy eredetileg TRS-80 rendszerprogramozó volt, és nála ez a helyzet az eltelt évek alatt sem változott semmit!

a szívem. Mindjárt nekiláttam egy pofonegyszerű Basic programcska megírásának. Ahogy az *1. képen* látszik, Basic-tudásom alaposan megkopott az eltelt évek alatt. A Commodore emulátor indításához az alábbi három parancs közül választhatunk:

```

x64      (C64-es üzemmód)
xp64     (PET üzemmód)
x128     (C128-as üzemmód)
  
```

Et oui, még több is van a tarsolyunkban. A C64 egy igazi számítógép volt, billentyűzettel, ezzel pedig programokat írhattunk. A hetvenes

évek végén, a '80-as évek elején megjelent gépek közül azonban nem sokból lehetett ezt elmondani. A korszak egyik legsikeresebb videójátéka az Atari 2600 Video Computer System volt – akkoriban minden srácnak volt ilyen, csak nekem nem, bár a haveroknál mindig órákig játszottam vele. Azok számára, akiknek álmatlan éjszakákat okoz e játékok hiánya, ott van a *Stella*, ennek fejlesztését *Bradford W. Mott* indította el, és mára egész komoly kis csapat szerveződött köré. A *Stella* ingyenesen hozzáférhető, de nem a GPL szerződési feltételei szerint. Ehhez a rendszerhez is rengeteg ingyenes játékot találhatunk az Interneten. A *Stella* játékok futtatása egyszerű, mindössze az

```
xstella _a_játék_útvonala
```

parancsra van szükségünk.

Az Atari 2600 csak egyike volt a cég által piacra dobott termékeknek. *Petr Stehlik* első számítógépe egy Atari 800XL volt, ezt azóta is az első szerelemhez hasonlítható érzéssel imádja. Jelenleg ő az Atari 800 emulátor fejlesztője (a program alapjait *David Firth* rakta le), és társaival együtt további változatok megjelentetésén munkálkodik.

A michigani egyetem irattárában rengeteg 8-bites Atari programot lelhetünk. Egy-egy adag letöltésével jó néhány óra felhőtlen szórakozást szerezhetünk magunknak. Az akciójátékok között megbúvó *chicken* nevű programmal talán még arra a régi kérdésre is megtalálhatjuk a választ, hogy mi volt előbb, a tyúk vagy a tojás.



2. kép Ó, azok a bájos 8-bites grafikák...

Ezeket a játékokat nagyrészt még a régi ARC formátumban tömörítették, tehát az *unstuff*-ra (Alladin Expander) lesz szükségünk a kicsomagolásukhoz.

Egy másik kedvenc régi gépem (ezen tanultam assemblerben programozni!) a TRS-80 Model-1, Level II. Ezen a gépen „szoktam rá” a BASIC-re is (a Fortrannal pedig egy IBM-1130-ason találkoztam először). A Radio Shack TRS-80 elvakult híveinek számára kötelező meglátogatni *Tim Mann* oldalait, ahol különböző TRS-80 emulátorokról és más, a géppel foglalkozó honlapokról olvashatnak. Itt tengernyi érdekességet találhatunk e majdnem elfeledett rendszer legapróbb részleteiről is. *Tim Mann*-ról csak annyit, hogy eredetileg TRS-80 rendszerprogramozó volt, és nála ez a helyzet az eltelt évek alatt sem változott semmit!

A gépemulátorok a programfejlesztők számára is nagy segítséget nyújthatnak. Az *Xcopilot* szintén ilyen program, a *PalmPilot*hoz. A rendszerben található *getrom* programocskával segítségével elkészíthetjük a kis gép ROM-jának egyfájlos változatát, és az *Xcopilot*tal már is programozhatjuk a Linuxon belül működő virtuális *PalmPilot*ot. E módszer előnye, hogy magát a *PalmPilot*ot

Kapcsolódó címek

Alladdin Expander (unstuff):

☞ http://www.alladdin-sys.com/expander/expander_linux.html

Atari 8 Bit World (egy Atari 800 emulátor):

☞ <http://joy.sophics.cz/a800.htm>

Commodore 64 emulátor:

☞ <http://mars.wiwi.uni-halle.de/ec64/>

Ktamaga:

☞ <http://ktamaga.sourceforge.net/>

Stella (egy Atari 2600 emulátor):

☞ <http://www4.ncsu.edu:8030/~bwmott/2600>

Tim Mann TRS-80 oldala:

☞ <http://www.research.compaq.com/SRC/personal/mann/trs80.html>

A michigani egyetem irattára:

☞ <http://www.umich.edu/~archive/>

A WINE „főhadiszállása”:

☞ <http://www.winehq.com/>

amúgy elég könnyen „hazavághatjuk” egy-két kísérletező jellegű programmal (tapasztalatból beszélek...), így viszont nyugodtan kísérletezhetünk vele, a Linuxnak nem ártunk semmit. Az alkalmazásokat a grafikus felületen is futtathatjuk, ha menthetlenül beleszerettünk a *Pilot* kezelői felületébe.

Végül azok számára, akiknek tényleg rengeteg szabadidejük van, ajánlok egy mindennél érdekesebb emulátort. Emlékszünk még azokra a kis nyavalyás, kulcstartó méretű pityegő izékre, a tamagocsikra? A nagy és a kisunokák nagy-nagy öröme megérkezett a *Ktamaga* nevű emulátor, amit a KDE grafikus felület alatt használhatunk.

Ha olyan emulátort keresünk a Linuxhoz, amelyről itt nem esett szó, csak látogassunk el a ☞ Freshmeat.net honlapra, és keressünk rá az „emulator” szóra, biztosan jó sokra rábukkanhatunk.

Nos, itt az idő, hogy bezárjuk Marcel séf borospincéjét. Azért indulás előtt még teletöltöm a poharakat.

Nocsak, François eltűnt. Biztosan félrevonult a kedvenc csirkés játékkal, non? Remélem, hogy ez a kis körutazás mindenki számára bebizonyította, hogy a Linuxszal visszarepülhetünk a régen elfeledett első játékaink, valamint számítógépeink idejébe. Viszlát a legközelebbi találkozásig a *Chez Marcel*ben, a kedvenc asztalunknál! Nézzenek be máskor is Marcel séfhez!

Á votre santé! Bon appétit!



Marcel Gagné (mggagne@salmar.com) az ontariói Mississaugában él.

A rendszerépítéssel és hálózati tanácsadással foglalkozó *Salmar Consulting Inc.* elnöke. Gyakorló pilóta, sci-fi és fantasy-író; a *TransVersions* című sci-fi, fantasy- és horrommagazin szerkesztője (nemsokára egy válogatásuk is megjelenik).

Imádja a Linuxot és a Unix összes változatát, jelenleg a *Linux Systems Administration: A User's Guide* című könyvén dolgozik, mely az *Addison Wesley Longman* kiadónál jelenik majd meg. *Marcel* honlapján (☞ <http://www.salmar.com/>) rengeteg érdekességre bukkanhatunk.

Biztonsági mentést végző programok összehasonlítása

A kisvállalkozások és az otthoni felhasználók körében az egyik legelhanyagoltabb rendszerfelügyeleti teendő a biztonsági másolatok készítése. Pedig erre igencsak szükség lenne. Ha egy, kizárólag a merevlemezek megbízhatóságától függő rendszer felmondja a szolgálatot, akkor foghatjuk a fejünket. Nem is beszélve az ezzel járó esetleges jogi problémákról – elég csak a határidős munkákat említenünk... Éppen ezért a biztonsági másolatok költségeit (programok, szalagok stb.) a biztosításokkal egyenrangúan kell kezelni. Csak egy egyszerű példa: otthoni irodámban öt számítógépen (két Linux-, két Windows 95-alapú és egy többrendszeres gép) két felhasználó dolgozik. A hálózat hagyományos 10-base T. két szalagos egység (egy Conner/Seagate 4 GB Travan és egy HP DDS 3 DAT-meghajtó), valamint az alábbiakban elemzett programok közül kettő gondoskodik az adatok biztonságáról. A PerfectBACKUP+, a BRU és az Arkeia a Weben keresztül is megrendelhető. Az Arkeiát a Red Hat, a Mandrake és a SuSE csomagokban is megtaláljuk. A Quick Restore azonban csak CD-n érhető el.

PerfectBACKUP+

Ez az egyetlen ingyenes program a négy közül, így nem is vártam túl sokat tőle (1. kép). Ennek ellenére meglepődve tapasztaltam, hogy a szalagváltókat is támogatja – erre igazán nem számítottam egy ilyen csekély szol-



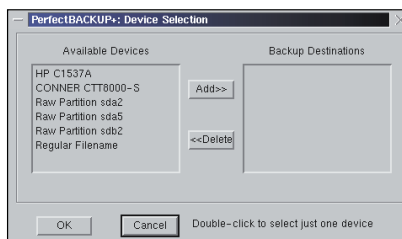
1. kép A PerfectBACKUP+ főablaka

gáltatású programban. Az egyetlen bosszantó dolog, hogy indításakor látható kis üzenettel figyelmeztet a bejegyzetés lehetőségére. Megnyugtató érzés volt, hogy a program mindkét SCSI szalagos meghajtómat felismerte, így más illesztőkkel nem is kísérleteztem. A két meghajtó nem a megfelelő eszközfájlokat kapta a /dev-ben, tehát ezeket nekem kellett beállítanom. Szóval a telepítés

néhány pillanat alatt elvégezhető, ami nem mondható el a többi, szalagos egységet is kezelő programról.

A PerfectBACKUP+ egyik előnyös tulajdonsága, hogy az eszközválasztó menü a meghajtókat a készülék által jelzett név alapján azonosítja, így nem kell az eszközneveket megjegyeznünk.

A PerfectBACKUP+ egy jól használható meghajtópróbát is tartalmaz, ezt szívesen

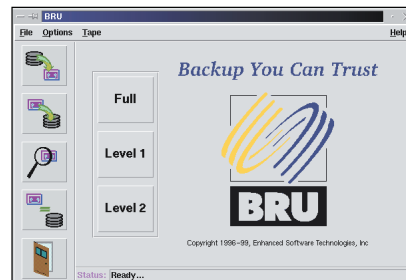


2. kép A célmeghajtó kiválasztása a Perfect BACKUP+-ban

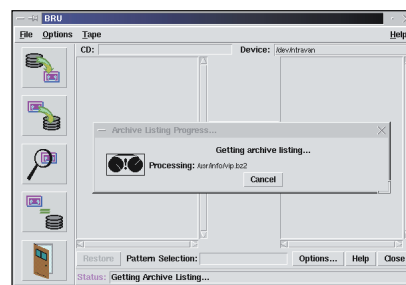
látnám más programokban is. Nagyobb hiányossága viszont az, hogy az adatmennyiséget nem lehet kézzel beállítani, azaz a nagyobb adatsomagok fogadását nem próbálhatjuk ki. A program által alapértelmezett adatmennyiség olyannyira kicsi, hogy a próba eredménye jóformán értékelhetetlen. A másolt csomag (a PB szóhasználatában „package”) beállítása egyszerű: a Backup menüben egyesével állíthatunk be minden jellemzőt, majd a Save As Package menüponttal menthetjük a csomagot, mely ezt követően bármikor visszaállítható és futtatható. Könnyen megtudtam adni egy kisebb csomagot, majd ugyanilyen könnyű volt a mentéseket elkészíteni.

A visszaállítással azonban néha gondok vannak. A következő titokzatos üzenetet kaptam: „Ez a mentés a főkönyvtárból készült, és Ön most a /test könyvtárat adta meg célként. Ha a másolat RELATÍV volt, akkor nem érdemes továbblépnie.” Mihez képest relatív? A leírást átböngészve semiféle adatot nem találtam a „relatív” jelentésével kapcsolatban. Ezután a program felkínálta a munkakönyvtár megváltoztatásának lehetőségét, és amennyiben ezzel nem éltem, hibáüzenettel befagyott. Ellenkező esetben, az előzőleg kiválasztott munkakönyvtár begépelése után a visszaállítás tökéletesen működött.

Az ilyen hibák jellemzően a nem megfelelő



3. kép Az XBRU főablaka



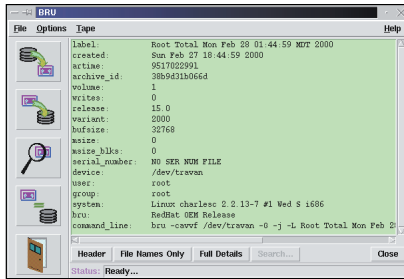
4. kép Egy tároló listájának lekérdezése

kipróbálásnak „köszönhetően” maradnak a programban. A grafikus felületen ilyen apróságot elviselek, ha el tudom kerülni. De egy ilyen hiba esetén magától értetődő a kérdés: vajon milyen kellemetlen meglepetéseket tartogat még számunkra a program? A PerfectBACKUP+ a kilépés után leállt folyamatokat hagy a rendszerben, ez szintén nem túl kellemes tulajdonsága.

Az ellenőrzés egyszerűen megy. A szalag tartalmát az eredeti fájlokkal is összehasonlíthatjuk, de CRC-ellenőrzést is végezhetünk. A meghajtó ismételt befűzésével (mount) kipróbálhatjuk a fejbeállítást. Ezt érdemes rendszeresen megtennünk, hiszen gyakori hibaforrást jelent a gyengébb minőségű szalagok és meghajtók használata.

Azt tapasztaltam, hogy az ellenőrzésre a másolási művelet közben nincs lehetőség, pedig ezt már a '80-as évek végén megjelenő PC-s programokban is megszokhattuk (elég csupán a Colorado Memory Systems termékeit említenünk).

A programokhoz kétféle leírás is elérhető. Az egyik HTML-formátumú, és a Netscape segítségével olvashatjuk el. A példaképernyők között olyat is találunk, amelyen egy egértámogatással is bíró xterm ablak látható; ehhez képest a program teljesen a grafikus



5. kép Egy szalag főbb adatai az XBRU alatt

felületet használja. A másik súgócsomag a menüből érhető el; ebben viszont nem tudunk keresni, amellét állandó szélességű karakterekkel jelenik meg, ez nagy felbontás esetén olvashatatlaná teszi a szöveget. Mindkét csomag más-más adatokat közöl, illetve egyik sem említi a parancssori kapcsolókat és értékeket, ez bizony nagy kár. Szomorúan kell megállapítanom, hogy a PerfectBACKUP+ befejezetlen; használatához meg kell várnunk az újabb, céltudatosabban ellenőrzött és jobb súgót tartalmazó változatokat. Mivel a kilépés után befagyott folyamatokat hagy a rendszerben, én nem bíznék meg benne. A programnak rengeteg hasznos szolgáltatása van, többek között képes a meghajtók név szerinti azonosítására. Remélem, hogy a Merlin kijavítja a hibákat és hamarosan megjelentet egy működőképesebb változatot.

BRU

A BRU a régi tar (Tape ARchive, szalagos tároló) programot helyettesítheti. A két alkalmazás olyannyira hasonlít, hogy még a megszokott parancssori kapcsolókat is használhatjuk, tehát az átváltás senkinek sem okozhat különösebb gondot. A program grafikus felülete, az XBRU csupán körítés: a felhasználó által megadott értékeket közvetlenül átadja a parancsoknak.

A BRU egyetlen hiányossága, hogy a grafikus felülettel nem készíthetünk parancsot, amit azután parancsfájlba menthetnénk. Pedig ez igencsak megkönnyítené a parancsok begépelésében kevésbé járatosak munkáját. Segítséget csak a cég honlapjáról kaphatunk, ez nagyszerű, hisz mindig a legújabb leírásokat érhetjük el. Ilyenkor felmerül az emberben a kérdés: a hivatkozások a leírások egy új változat megjelenése után is használhatóak lesznek a nálunk lévő programhoz? És fiúk, nem mindenki T1-es kapcsolaton át éri el a Világhálót!

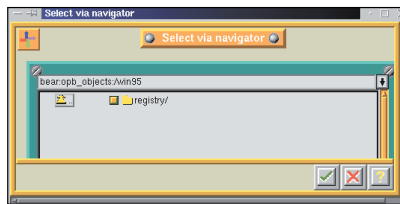
A telepítés mindössze néhány pillanatot vesz igénybe. Az EST Inc. honlapján található bemutatóváltozatot telepítettem, ez ugyanis letölthető egyetlen tar fájlként. A megfelelő helyre való másolás és a kicsomagolás után a ./install parancsot kell begépelnünk. A telepítőfájl a szalagos

egységekről kérdezősködik: meg kell adnunk a meghajtó (tömörítetlen) méretét, és a visszacsévélésre alkalmas, illetve az arra alkalmatlan eszközök nevét. Az eszközökhez ezután nem kellett hozzányúlnom.

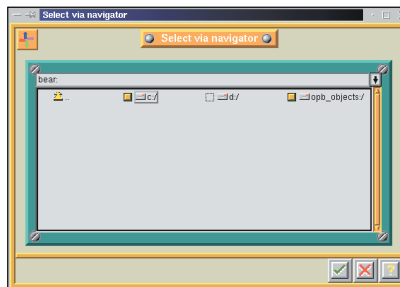
A honlapról számos meghajtó beállításfájlt leszedhetjük. Egyszerre több XBRU-t is futtathatunk, így egyidejűleg több szalagos egységet is kezelhetünk. Ez egy ügyes húzás, de meg sem közelíti az Arkeia rugalmasságát.

A másolat ellenőrzése hasznos dolog, s ezt az EST is így gondolja: minden másolásnál felkínálja ennek lehetőségét. Az automatikus ellenőrzést egyszerűen beépíthetjük parancsfájljainkba is.

A fizetős változatot nem láttam, de azzal akár a tömörített fájlok adatbázisát is elké-



6. kép Egy Windows 95-ös rendszerleíró adatbázis kiválasztása



7. kép Egy Windows 95-ös géphez tartozó gyökérkönyvtár megadása



8. kép Biztonsági mentés az Arkeiában

szíthetjük, így egy fájl keresésekor nem kell az összes szalagot átvizsgáljunk. A BRU nem támogatja a szalagváltós egységeket. Mivel én nem használok ilyesmit, nem nyilatkozhatom megbízhatóságukkal

kapcsolatban sem. Inkább úgy szervezem meg a biztonsági mentéseket, hogy ne kelljen cserélhető szalagot használnom. Sajnos a merevlemez-gyártók mérethasználata kevédi megfertőzni a szalagos egységek készítőit is, szóval a jövőben minden bizonnyal ezek terjednek majd el.

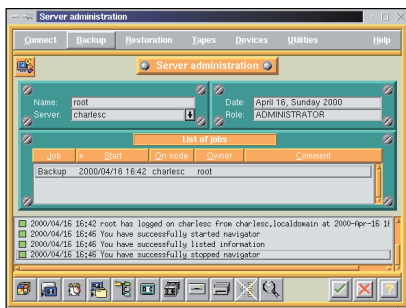
Arkeia

A sorban a következő a Knox Software cég Arkeia programja, mely három részből áll. A *kiszolgálói* (a forrásgépen fut) program végzi a másolást, a megfelelő fájlengedélyek beállításával együtt. A *kiszolgáló* menti és visszaállítja a fájlokat, az *ügyfél* kezeli a szalagos egységeket, a *felhasználói felülettel* pedig az egész folyamatot irányíthatjuk. Ez utóbbinak parancssoros és grafikus változata is létezik.

E felállás egyik előnye, hogy az Arkeia windowsos kiszolgálója a rendszerleíró adatbázist közvetlenül metaadatként is mentheti, nemcsak egyetlen fájlként. Azonban a program levelezési listájának néhány hozzászólója szerint ezzel óvatosan kell bánni. Az Arkeia használata előtt tehát ellenőrizniük a rendszerleíró adatbázis mentését és visszaállítását valami egyszerűbb gépen.

Elvileg a részeknek nem kell ugyanazon a gépen lenniük. Otthon a szalagügyfelet és a felhasználói felületet egyazon számítógépen futtatom. Ezen a gépen, valamint a hálózat másik négy gépen kiszolgálók is működnek. Az elrendezés lehetővé teszi a folyamatos adatáramlást, amit a program „flow” néven emleget. A legtöbb biztonsági mentést végző program egyszerűen végigjárja az menteni kívánt könyvtárakat; erre talán a Windows NT biztonsági mentése a legjobb példa. Az Arkeiával a kiszolgálók egyszerre küldhetnek adatokat az ügyfélnek, így egy időben több biztonsági mentést is készíthetünk. Saját rendszeremben mind az öt gépet egy adatfolyam (flow) képviseli. Az eredmény: rendkívül gyors biztonsági mentés. A HP DDS 3 DAT meghajtóm akár 57 MB-ot is képes fogadni percenként; ez pedig 3 GB-ot jelent kevesebb, mint egy óra alatt! Ez az adatáramlás lehetővé teszi a meghajtó folyamatos működését, ami a szalag és a szalagos egység élettartamára is kedvező hatással van. Bár jómagam nem próbáltam ki, de elvileg egyszerre több gépről több gépre is menthetünk, ilyenkor az Arkeia a folyamatos működés megteremtéséért irányítja az adatok áramlását.

Az Arkeia a biztonsági mentéseket egy adatbázisban naplózza. Ez nagyszerű, hiszen így gyorsan áttekinthetjük az akár több száz szalagból álló mentésünket. Azonban itt jelentkezik a program első komolyabb hiányossága is: maga az adatbázis nem kerül föl a szalagra. Ha tehát a naplót tároló



9. kép A kiszolgáló felügyelete

számítógép unja a banánt és fölmondja a szolgálatot, az adatbázis visszaállításához minden egyes szalagot át kell böngészniük. A nehézséget a legegyszerűbben úgy hidalhatjuk át, ha az adatbázist a BRU segítségével utólag lementjük.

Az Arkeiáról szóló előző cikkemben (*Linux Journal*, 1999. áprilisi szám,

www.linuxjournal.com/issue60/3166.html) tökéletes vásárlói támogatásról számoltam be. Bár mostanában nem szorultam rá e szolgáltatásra, a levelezési listán egyesek morognak miatta. Többek között azt is említették, hogy a Knoxról szóló írásom óta nem nagyon akarózik válaszolni a segítséget kérő levelekre. Lehet, hogy odabenn azt gondolják: egy levelezési lista helyettesítheti a kifogástalan vásárlói támogatást. Hát szerintem nem.

Az Arkeia telepítése komoly feladat. Először is mindent be kell állítanunk, a meghajtóktól a szalagokon át a mentési csomagokig. A biztonsági másolat előkészítése a mentési csomag és az tárolandó fájlok kiválasztásából áll. Ezután bármikor elindíthatjuk a műveletet, akár saját kezűleg, akár megadott időpontokban, automatikusan. A kezdeti nehézségekért bőven kárpótol a program hihetetlen rugalmassága. Igaz-e varázslóval könnyebben boldogulnánk, mint az Arkeia által használt keszkesza menükkel. A leírás részletesen ismerteti az első teendőket, egészen a próbamentés elkészítéséig.

A négy program közül az Arkeiát használhatjuk a legtöbb operációs rendszerben. Ha a Linuxon és a Windows NT-n kívül más rendszert is használunk, érdemes átböngészni a Knox honlapját; itt megtalálhatjuk a támogatott operációs rendszerek listáját. Az Arkeia legnagyobb hiányossága az, hogy a másolatokat nem tudjuk ellenőrizni. Sem adat-, sem pedig CRC-ellenőrzésre nincs lehetőség. A legtöbb, amit tehetünk, hogy a másolat egy részét visszaállítjuk egy átmeneti helyre, majd azt az eredeti adatok megfelelő részével összehasonlítjuk. Persze így csak egy töredék biztonsága felől lehetünk nyugodtak. Az ismertett programok közül egyedül az Arkeiából hiányzik ez a szolgáltatás.

A termék ára a hálózatba kötött gépek számától függ. Pontos ajánlatért látogassunk el a program honlapjára. Innen egyébként egy bemutatóváltozatot is letölthetünk, mely egyetlen szalagos egység és meghatározott számú ügyfél használatát teszi lehetővé.

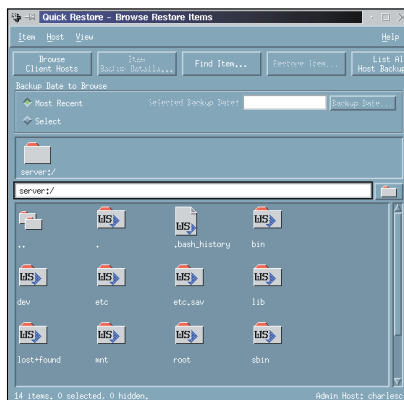
Quick Restore

A négy közül az egyetlen, valóban üzleti felhasználásra szánt program, a Workstation Solutions cég terméke a Quick Restore. A legtöbb Linux-felhasználó megszokta, hogy az Internetről letöltött csomagokat azonnal használja. A Workstation Solutionsnál erről kissé (!) máshogyan gondolkodnak. A kipróbálható változat csak CD-n hozzáférhető, ennek megrendelése során pedig a vállalat ügyintézőjével kell időpontot egyeztetni. A telepítést egy szakember felügyeli, aki segít tájékozódni a beállítások rengetegében. A megrendelés hosszas oda-vissza levelezéssel kezdődik. Én néhány levél után dühömben megmakacsoltam magam, és nem válaszoltam, így nem kevés időbe telt, míg megkapharinttam a hön áhított lemezt és létrejöhett a találkozó. Abban mindenestre biztos vagyok, hogy ez az eljárás garantáltan elriasztja az egyszerű érdeklődőket. Maga a telepítés ehhez képest leányálom. A legtöbb telepítőprogramhoz vagy nem kapunk segítséget; ha viszont igen, akkor az igazán kényes nehézségekre legtöbbször nem találunk választ. A Quick Restore felhasználók által igénybe vehető telefonos ügyfélszolgálatának előnyeit nyilvánvalóan nem szükséges eszetelnünk: a vonal másik végén egy kedves hang igyekszik segíteni minden bajunkon.

A főképernyő (11. kép) egy hat gombból álló eszköztár. A gombok mindegyike külön ablakot nyit, akár a legtöbb X prog-



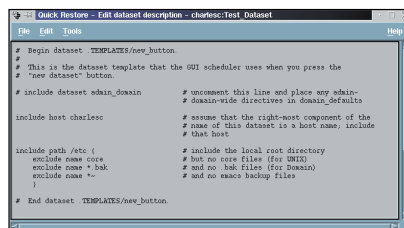
10. kép A Quick Restore főablaka



11. kép A visszaállítandó fájlok kiválasztása

ram. A dolog hátránya, hogy könnyen megtelhet a képernyő; viszont az az előnye, hogy az ablakok között – a legtöbb windowsos alkalmazásétól eltérően – bármikor válthatunk.

A visszaállításhoz a grafikus felületre van szükségünk (12. kép). Az Arkeiához hasonlóan, a másolt fájlokról készített adatbázis faszerűen rendezett, s így az eredeti könyvtárszerkezetet mutatja. Katintunk a visszaállítandó fájlokra, majd válasszuk ki a kötetet, végül a szalagot.



12. kép A mentésre kerülő Windows 95 rendszerleíró adatbázis az Arkeiában

Az Arkeiával ellentétben (6. kép) a másolatokat parancsállományok segítségével határozhatjuk meg. A Quick Restore-nak beépített szerkesztője is van, ez azonban sajnos csak egy egyszerű CUA-szerkesztő (legjobban az MS-DOS edit.com programjához tudnám hasonlítani). A jó hír viszont az, hogy az állományok hagyományos szövegfájl formátumúak, tehát módosításukhoz bármilyen szerkesztőt használhatunk. A leírásban nem találtam utalást arra, miként lehetne külső szerkesztőprogramot (vi, emacs stb.) használni a program sajátja helyett.

A programnyelv világos és részletes leírást kapunk hozzá; az emacs használatával az adott feladathoz tökéletesen megfelelő parancsfájlokat tudtam készíteni. A beépített szerkesztő ellenőrizi is a szöveget, ezt a szolgáltatást érdemes minden esetben igénybe vennünk.

Az adatsomag meghatározása után hozzá kell rendelnünk egy biztonsági mentést. Ezeket naponta, hetente, havonta vagy évenként ismételtethetjük. Azonnali mentésre nincs lehetőség, inkább állítsunk be várakozási időnek egy percet, és dőlünk hátra. A Quick Restore telepítése során a szakember segített a művelet lebonyolításában, és próbaképpen egy kisebb csomagot mentettem. A programot nagyon hamar meg lehet szokni, de talán mégsem olyan könnyen, mint az Arkeiát. Hiába, a nagyobb rugalmasságért többet kell fizetnünk. A Quick Restore azonnali használatához egyszerűen helyettesítsük a rendszer tar parancsát a qtarral a már létező parancsfájlaikban. A Quick Restore számos parancsori eszköz tartalmaz. Az egyik a fent említett qtar, ez hibátlanul kezeli a tar formátumú

A programok adatai

PerfectBACKUP+

Gyártó: Merlin Software Technologies, Inc.
 Levél: info@merlinsofttech.com
 Honlap: http://www.merlinsofttech.com/
 Ár: 69–89 dollár

BRU

Gyártó: Enhanced Technologies Software, Inc.
 Levél: info@estinc.com
 Honlap: http://www.estinc.com/
 Ár: 245 dollár

Arkeia

Gyártó: Knox Software Corp.
 Levél: sales@arkeia.com/
 Ár: ingyenes

Quick Restore

Gyártó: Workstation Solutions
 Levél: info@worksta.com
 Honlap: http://www.worksta.com/
 Ár: érdeklődjön

fájlokat és annak parancssori kapcsolóit, is támogatja. Ezeket héjprogramokba (shellscript) foglalhatjuk, szerkesztésüket a csomagban található példák is segítik. A grafikus felület valójában a parancssori eszközök kezelését könnyíti meg. Itt is kényelmesebb lenne, ha a felületből közvetlenül menthetnénk parancsállományokat. Egy másik segédeszközzel rengeteg adatot tudhatunk meg a szalagos egységről, még akár az esedékes tisztítás javasolt időpontját is (amennyiben ezt a meghajtó támogatja). A Quick Restore kifejezetten a nagyvállalkozások, kiszolgálótelepek és ehhez hasonló méretű rendszerek számára készült: egy mezei Travan meghajtót fel sem ismer. A DDS 3 szalagos egységem még éppen befért a támogatott eszközök közé, de ennél gyengébb teljesítményével akár ne is próbálkozzunk. A DDS 3 és a Travan közti fő különbség egyébként az, hogy előbbi a szalag bármely pontját képes közvetlenül elérni, míg a Travan csak soros hozzáférést nyújt az adatokhoz. Mind a honlapon, mind pedig a csomagban található leírás tökéletes. Tartalmuk nem azonos, ezért telepítés előtt mindenképp olvassuk el a nyomtatott leírást. Én nem így tettem, s bár túlestem a nehezén, beismerem, hogy jóval egyszerűbben ment volna az egész, ha időt szakítok nyomtatott oldalak átböngészésére, amelyek egyébként egészen jó kiindulópontot jelentenek a biztonsági másolatok készítésében járatlan felhasználók számára is.

A Quick Restore, az Arkeiához hasonlóan minden olyan gépen a háttérben fut, ahonnan az adatokat mentjük. A programnak jelenleg Linux-, többféle Unix- és Windows NT-változata létezik. Windows 95/98 gépekről is készíthetünk mentést, de csak SMB-fájlszerveren keresztül. Ez azzal a hátránnyal jár, hogy a rendszerleíró adatbázist csak hagyományos fájlként tárolhatjuk. Ezt egyébként mindenképpen érdemes elvégeznünk, hiszen a Windowsnál nem fordítottak erre kellő figyelmet, s legtöbbször csak az efféle óvatosság menthet meg minket. Az említett négy program közül csak a Quick Restore használja az NDMP-t (Network Data Management Protocol) a kiszolgáló és az ügyfél közti adatforgalomban. Megálmoldói szerint ez az egyre elterjedtebb internetes szabvány a különböző gyártók termékei

közti adatcserét segíti elő, ami nagy előny a többrendszeres környezetek esetén. A Quick Restore nem olcsó. A Workstation Solutions 1750 dollárt kért tőlem egy kiszolgálóért, s ehhez jön még 1595 dollár az egyéves támogatásért. A Linux és a FreeBSD ügyfeleket ingyen adják. Az NT ügyfelek 2250 dollárba kerülnek, további 495 dollárt kell fizetnünk az éves támogatásért. Az Arkeiával ellentétben itt nincs lehetőség otthoni használatra vásárolni. A Quick Restore-ért elkért rettentő összeg tökéletes terméktámogatást előlegez, s bár én a telepítés óta nem vettem fel a kapcsolatot a gyártókkal, megbízhatónak tűnnek.

Szalagformátumok

A PerfectBACKUP+ a cpio, a Quick Restore a tar formátumot használja. Mindkettőt viszonylag régen fejlesztették, így a mai formátumok esetében kérdéses lehet az adatbiztonság. A BRU ezzel szemben saját formátumot használ, ennek előnye (legalábbis a cég szerint) a tökéletes adatbiztonság, hátránya viszont, hogy a visszaállítás csak a programmal végezhető el. Megfigyeléseim szerint az Arkeia is belső formátumban tárolja az adatokat. A csomag parancssori segédeszközt is tartalmaz, ezzel visszaállíthatjuk a szalag tartalomjegyzékét, s ennek alapján végezhetjük a visszaállítást.

Összegzés

Csodálatos dolog a piacgazdaság! Egy feladatra számtalan megoldás kínálkozik, más-más minőségben és árban. Az egyetlen terület, mely aggodalmaskodást válthat ki a felhasználókból, a Windows NT biztonsági mentése. A Windows 2000 több fájljellemző bittel bír, mint a Windows NT, ezenkívül a fájlrendszerek adatszerkezete is különbözik a két rendszerben. A kérdés az, hogy a Windows NT-t is támogató programok felismerik-e, ha éppen Windows 2000-rel van dolguk? Az ismertetett két NT-s program leírásban semmiféle tájékoztatást nem találtam ebben a témakörben, így a kérdést nem vizsgálhattam meg közelebbről. A Workstation Solutions illetékes szerint már dolgoznak a megfelelő kódrészleteken, s hasonló munka minden bizonnyal a Knox berkeiben is folyik. Ha nem a saját pénzünket költjük, illetve ha van pénzünk a programra, de a beállítások elvégzésére nincsen elég emberünk s még Windows 95/98-alapú gépekkel sem kell bajlódnunk, akkor nyugodt szívvel választhatjuk a Workstation Solutions termékét. A Quick Restore-nak van a legjobb leírása az itt leírt négy program közül.

Amennyiben Windows 95/98 is található a hálózatban, és ezen gépek adatait is menteni szeretnénk, továbbá nem bánjuk, ha egy kissé zilált felhasználói felülettel kell megbirkóznunk, szerezzük be az Arkeiát. Az Arkeia a sok biztonsági másolat készítését igénylő rendszerekben is jól megállja a helyét. Kezelése gyorsan megtanulható, és hamar elkészíthetjük a számunkra megfelelő parancsállományokat. A program segítségével a legtöbbet hozhatjuk ki a szalagos egységekből, akár folyamatos terhelés mellett is. Egy területen azonban behozhatatlan a Quick Restore előnye az Arkeiával szemben, ha több adatbázisról kell biztonsági másolatot készítenünk, ezt nyilván a lehető legrövidebb idő alatt szeretnénk elvégezni, hiszen így megtekinthető az adatbázis csaknem százszázalékos rendelkezésre állása. Az adatbázis kivitele (exportálása), majd a másolat mentése helyett egyszerűbb, ha az adatbázist egy időre lezár-

További érdekességek

- A Linuxot támogató szalagos egységek (az EST nyilvános szolgáltatása):
- ➔ <http://www.linuxtapecert.org/>
- Linux Mall:
- ➔ <http://www.linuxmall.com/>
- HP's One-Button Disaster Recovery (OBDR):
- ➔ http://www.hp.com/tape/papers/obdr_wp.html
- Network Data Management Protocol (NDMP):
- ➔ <http://www.ndmp.org/>

juk, elkészítjük a mentést, újra üzembe helyezzük, majd ugyanezt a műveletsort a többivel is elvégezzük. Az Arkeia használata során csak a mentés előtt és után futtathatunk parancsállományt. Az állásidő minél rövidebbre „faragásáért” minden adatbázishoz külön biztonsági mentést kell beállítanunk. A Quick Restore ezzel szemben a mentés bármely pillanatában engedélyezi a parancsfájlok futtatását, így az egész számítógép tartalmát menthetjük, s emellett az adatbázisok csak a lehető legszükségesebb időtartamra maradnak zárva. A BRU grafikus felülete egyáltalán nem teszi lehetővé a parancsfájlok futtatását, azonban a parancssori eszközökkel ez a hiányosság teljesen kiküszöbölhető.

A PerfectBACKUP+ úgy tűnik, semmilyen lehetőséget nem nyújt a szkriptek futtatására. A program profi változata állítólag az Arkeia és a Quick Restore vetélytársa lehet, de e cikk írásakor sajnos még nem volt elérhető. A BRU otthoni változata tökéletes eszköz Linux-alapú gépeink és más, az NFS vagy a Samba segítségével hozzá kapcsolt gépek biztonsági mentéséhez. A kis- és középvállalkozások számára áthidaló megoldást ajánlok: a hálózatot az Arkeiával, a szalagos kiszolgáltót pedig a BRU-val mentjük. A tökéletes védelemhez ezeken kívül az EST Crash Recovery Utility vagy QuickStart Data Rescue programjára is szükség lehet. Mivel egy HP OBDR (One-Button Disaster

Recovery) szalagos egységem van, ezért nekem a CRU tűnik igazán vonzónak. A Windows-alapú gépeket is magában foglaló otthoni hálózatokhoz az Arkeia próbaváltozata a legmegfelelőbb, de ne felejtjük el a program adatbázisát a BRU-PE vagy a tar segítségével menteni.

Charles Curley (ccurley@trib.com)
Wyoming államban él. Lovagol, emellett marhákat, macskákat és elektronokat tenyészt. Ezek közül persze csak az utóbbi fizet jól, így egy redmondi kis fejlesztő cég számára készít leírásokat.

© Kiskapu Kft. Minden jog fenntartva

Összefoglalás

Program	Arkeia	BRU	PerfectBACKUP+	Quick Restore
Gyártó	Knox Software	Enhanced Software Technologies	Merlin Software	Workstation Solutions
URL	http://www.knox-software.com/	http://www.estinc.com/	http://www.perfectbackup.com/	http://www.worksta.com/
Ingyenesen letölthető?	Igen	Igen	Igen	30 napos demo CD
Ár	Ingyenes vagy változó; érdeklődjön	245 dollár (a Linux Mallon)	69 dollár (dobozzal 89 dollár)	Érdeklődjön
Leírás	PDF	A honlapon	A honlapon	A honlapon + 3 könyv
Szerződés	Üzleti	Üzleti	Nincs adat	Üzleti
Csomag	RPM, tgz	tar	tar	CD-ROM
Lefagyás utáni visszaállítás?	Nincs	CRU (a HP OBDR-jével), Quickstart	Linux alatt nincs	Nincs
Ellenőrzés	Nincs	Van	Van	Van
Hálózati mentés	Van	Van (A Personal Editionban nincs)	NFS	Van
Nyers lemezrész	Nincs	Van (A Personal Editionban nincs)	Van	Nincs
Visszaírás fájlba	Van	Van	Van	Van
Windows	Saját démon	Samba	Samba	NT, W95/98 a Sambán keresztül
Grafikus felület	Van	Van	Van	Van
Konzol	Korlátozott	Van	Hiányos leírás	Van
Levelezési lista	Van	Ismeretlen	Ismeretlen	Ismeretlen
Ütemezés	Van	Van	Van	Van
Szalagváltók támogatása	Van	Csak parancssorból	Van	Van

Megjegyzések a táblázattal kapcsolatban:

Ár: az Arkeia ingyenesen letölthető változatát csak meghatározott ideig és feltételekkel használhatjuk.

Szerződés: A PerfectBACKUP+ szerződési típusát nem tudtam megállapítani, de a forráskód minden bizonnyal nem jár a csomaghoz.

Lefagyás utáni visszaállítás: ez egy egyszerű kényelmi szolgáltatás.

Ellenőrzés: a mentés befejezése után végrehajtható ellenőrzés.

Windows: lehet-e windowsos gépről biztonsági mentést készíteni, a fájlengedélyekkel együtt? Ne felejtjük el, hogy a Samba nem minden esetben kezeli megfelelően a Win32 fájlleíró biteit, éppen ezért nem a legjobb megoldás. A Sambán keresztül végzett biztonsági mentéseknél a regisztrációs adatbázissal is gondok lehetnek.

Levelezési lista: létezik-e a cég által támogatott levelezési lista a program felhasználói számára?

A VoIP és a beágyazott Linux

Az Aplio az internetes telefonhívásokhoz kínál új lehetőséget, Microsoft Windows operációs rendszereken. A készülék beágyazott Linuxot tartalmaz.

A telekommunikáció hatalmas piac, mely óriási lépésekkel fejlődve egyre összetettebbé válik, és egyre nagyobb igény mutatkozik az intelligens készülékek iránt. A telekommunikációs piac aránylag új ága az IP-n keresztül továbbított hang, a VoIP (Voice over IP).

Az általános recept szerint egy Windows-alapú PC, hangkártya és megfelelő program segítségével ingyenes távolsági hívást bonyolíthatunk le az Interneten keresztül.

A dolognak persze hátrányai is vannak, hiszen ilyenkor egy PC-re is szükségünk van a hívások fogadására és küldésére. Az Aplio – pontosabban az Aplio/Pro önálló készülék – használata során mindkét oldalon egy-egy készülék kell, számítógépre azonban nincs szükségünk. Bár az Aplio már eddig is ellátta termékeivel ezt a piacot, az Aplio/Pro megjelenése most azt mutatja, hogy a cég a továbbiakban a nyílt forráskódú fejlesztésekre összpontosít.

Henri Tebeka, az Aplio technológiai osztályának vezetője szerint: „A Linux a jelenlegi technika számára ideális operációs rendszer. Internetkapcsolatra épül, jogdíjak nélkül beszerezhető, emellett nyílt forráskódú elemekből felépülő programszerkezete megteremti fejlesztéseink folyamatosságát, és erősíti helyzetünket az internetes telefonkészülékek piacán. Ha a VoIP-technológia legújabb eredményét, az Apliót a Linux operációs rendszerrel kapcsoljuk össze, akkor olyan fejlett modellt kapunk, amely képes a folyamatos, nem digitális hang, illetve szövegek, álló- és mozgóképek közvetítésére.” Az Aplio/Pro beépített processzort, modemet, flash memóriát tartalmaz, valamint teljesen duplex (felvételre és lejátszásra egyszerre képes) hangrendszere van, ez hardverszinten valósítja meg a hangátvitel tömörítését.

Jerome Calvo, az Aplio cég ügyvezetője és egyben elnöke kételkedik abban, hogy a felhasználók számítógép igénybevételével szeretnének telefonálni: – „Szerintünk az internetes telefonáláshoz nincs szükség számítógépre. Mi önálló eszközt fejlesztünk.” Az internetes hangátvitel ma már jóval piacképesebb, mint egy-két évvel ezelőtt. Néhány cég még mindig a PC és a megfelelő program párosításában látja a jövőt, de ez természetesen meg sem közelítheti a hardveres megoldások minőségét.

A Linux-alapú Aplio/Pro beüzemelése egyszerű: a készülék egyik csatlakozóját – az üzenetrögzítőhöz hasonlóan – a telefontal, a másikat egy Ethernet kapuval kell összekötni.

Az Aplio/Pro beállításához az Internet-szolgáltatói adatok beírásán kívül nem sok mindent kell tennünk. A készülék alján található tízjegyű azonosító a saját készülékünk legfontosabb adata. Ennek ismeretében máris hívhatunk egy másik Apliót, vagy egy internetes telefonálást támogató számítógépet, tehát akár egy Microsoft NetMeeting programmal felszerelt Windows PC-t is.

Híváskor emeljük fel a kagylót és nyomjuk meg az „Aplio” gombot, üssük be a hívni kívánt Aplio készülék azonosítókódját, ezután várjuk meg a partner jelentkezését. Ilyen egyszerű az egész! Az Aplio tűzfal vagy NAT készülék mögötti használatához csupán a megfelelő kapukat kell beállítanunk, ahogyan azt az ICQ-ban vagy más valósídejű átviteli protokollt használó programban tennénk.

Mi az a VoIP? Mi az az Aplio?

A VoIP (Voice over Internet Protocol) az internetprotokollon keresztül történő hangátvitelt jelenti, illetve bármilyen szolgáltatást, melynek segítségével az Interneten keresztül telefonálhatunk. Ha belső hálózatról beszélünk, akkor általában VoIP-nak nevezzük a módszert. Ha a Világhálóról van szó, akkor leginkább az internetes telefónia kifejezést használjuk – bár a két kifejezés a legtöbb szövegvagykörnyezetben felcserélhető egymással. Míg az internetes VoIP hangminősége a hagyományos telefonvonal minőségével vetekszik, az internetes telefónia hangminősége szélsőségesen változó – minden előnye ellenére leginkább ez nehezítette meg a módszer világméretű elterjedését.

Az internetes telefónia kiépítésében a VoIP-szállítók, az Internet-szolgáltatók és a webes cégek járnak az élen. A Linuxszal ellentétben (mely bár jó néhányszor bizonyított, például a webes alkalmazások területén) a legtöbb vállalat egyelőre csak kísérletezik VoIP-vel. Hogy miért? Először is nagyon nehéz olyan megoldásokat találni, amelyek tökéletesen kezelik az egyes alkalmazási helyeken előforduló igényeket, adatszerkezeteket. Ez a legtöbb érdekelt cég kedvét elvette attól, hogy üzletét kizárólag a VoIP-re építse. Bár a távolsági telefonhívások kiváltásával igen komoly összegeket takaríthatnak meg, ennek ellenére nem az óriás vállalatok lesznek a VoIP első használói. Valójában a VoIP lehetőségeit kihasználó szervezetek ugyanazok, melyek a kezdetektől fogva Linuxot használnak: a kisebb üzletek és cégek.

Az Aplio az 1998-as Comdex kiállításon robbant be az internetes telefónia területére az Aplio/Phone 2.0 megjelenésével. Az ott bemutatott változatban már számos fontos szolgáltatást megvalósítottak (az OpenApliótól a PacketPlusig), ennek köszönhetően az Aplio/Phone messze kiemelkedett az akkori mezőnyből. Az OpenAplio a H.323 szabvány segítségével lehetővé teszi a PC-k és az Aplio/Phone közötti telefonbeszélgetést. A PacketPlus az internetes telefónia két legnagyobb kihívását, a hang késését és a forgalmi akadályokat kísérli meg legyőzni. Ezt úgy éri el, hogy állandóan figyeli a Világhálót, a csomagméretet folyamatosan változtatja, s így a hagyományos telefonbeszélgetések minőségét képes visszaadni.

Az Aplio/Phone 2.0 az Aplio Configuration Assistant nevű programot is tartalmazta, mellyel a készülék távolról is beállítható. Az *Internet Telephony* magazin néhány hónappal később Az „Év terméke” díjjal tüntette ki az eszközt és alkotóit. Az Aplio/Phone már korábban belopta magát a lap szerkesztőinek szívébe, hiszen ugyanabban az évben már elnyert egy másik díjat is az újságtól. Az Aplio/Phone egyszerű telepítésének és kiténő ár–teljesítmény viszonyának értékelése után a bírák így fogalmaztak: „Az Aplio/Phone az utóbbi évek legfontosabb felfedezése és leghasznosabb terméke a hang- és adattovábbítás területén.”



Az Aplio fejlesztése

Hogyan illeszkedik a képbe a Linux? Az Aplio az internetes telefonkészülékek fénykorában kezdte világszerte terjeszteni fejlesztéseit. Első terméke, az Aplio/Phone hagyományos (PSTN) telefonvonalon működik, és az Integrated Systems (ma ez a Wind River Systems csoport tagja) által szabadalmazott valósídejű PSOS operációs rendszert használja.

Az Aplio/Pro ezzel szemben önálló, Linux-alapú internetes telefonkészülék. Akárcsak az Aplio/Phone, a Pro is elnyerte „Az év terméke” díjat mind az *Internet Telephony*, mind a *Communications Solutions* angol nyelvű magazinoktól. Az *Internet Telephony* szerint „az Aplio/Pro a legjobb VoIP készülék”. Az Innomedia, az InterStar és a Komodo néhány versenyképes termékének elismerése mellett is megállapíthatjuk, hogy „az Aplio/Pro a legjobb választás, mind az üzembe helyezés kényelmét, mind pedig használhatóságát tekintve”. Az Aplio/Pro operációs rendszere egy beágyazott és módosított *uClinux/ARM*, ezt a *uClinux*-ból, a memóriaegység nélküli mikroszámítógépekre készített Linux-változattól fejlesztették ki.

A *uClinux/ARM* rendszert először az Aplio TRIO chipje használta. A *uClinux*-ot a 2.0.38-as rendszermagból alakították ki, az *ARM* és a *uClinux* hibajavítások beillesztésével.

Vadim Lebedev, az Aplio fejlesztőmérnöke és a linuxos változatot készítő csapat vezetője szerint, a legnagyobb nehézséget a rendszermag átalakítása okozta: – „Sokkal nehezebb volt azt megvalósítanunk, hogy a GCC/ARM fordító a felhasználói alkalmazásokhoz helyzetfüggetlen kódot készítsen.”

Lebedev az Aplio/Pro VoIP fejlesztése során kezdett Linuxszal foglalkozni. A *uClinux* melletti döntést három ténnyel indokolja. Először is a forráskód nyílt, így sokkal könnyebben módosítható. Másodsorban, a kód kényelmesen tesztelhető Linux munkaállomásokon, majd egyszerűen újrafodítva könnyen a készülékhez igazítható. Végül, de nem utolsósorban a Linux-közösségtől kapott segítség komoly hátszínnyel teremtett a fejlesztés számára. „Elmondhatjuk, hogy sokkal egyszerűbb egy Linuxra épülő beágyazott szoftvert kifejleszteni, mint egy PSOS-re épülőt. Hiszem, hogy a beágyazott rendszerek világában fényes jövő vár a Linuxra.” – tette hozzá Lebedev.

A mostanában kifejlesztett Aplio/TRIO chip belső Ethernet kapu segítségével hozza létre a kapcsolatot az Internettel, s ezzel egyidőben valósídejű hangtömörítést és visszhang-csillapítást végez. Az Aplio/TRIO egy 20 MHz-en futó ARM7TDMI mikrovezérlő rendszermagra, valamint két 40 MHz-es DSP társprocesszorra épül. Az Aplio/Phone-hoz hasonló környezetnél az egyik DSP társprocesszor felelős a hangátvitelért, beleértve a hangtömörítést és a visszhang megszüntetését, míg a másik végzi a hagyományos feladatokat: a tárcsázást, a hívó fél azonosítását stb.

Az Aplio és a VoIP piaca

Az *aplio.com* közvéleménykutatása alapján a jövőben a VoIP telefonok piacának növekedésre számíthatunk. A Probe Research kutatóintézet becslése szerint 2005-re az Interneten keresztül folytatott telefonbeszélgetések ideje elérheti, vagy akár meg is haladhatja a 250 billió percet! Aplio készülékekkel 1999. júniusáig világszerte ötmillió percet beszéltek.

Calvo szerint a világvárosok lakói adják az Aplio/Phone és Aplio/Pro felhasználók 70 százalékát. Ezek az emberek nagyvárosban dolgoznak, azonban családjuk máshová köti őket. A VoIP nélkül nekik sokkal többbe kerülne egy távoli ismerőssel, rokonnal folytatott beszélgetés. Calvo számításai alapján az Aplio ügyfeleinek fennmaradó 30 százaléka főként olyan nemzetközi üzletekben érdekelt vállalkozókból áll, akik szintén meg szeretnék kerülni az üzleti életben oly gyakori távolsági hívások magas díjait. Azt is megkérdeztük a tervezőtől, hogy miben látja az Aplio előnyét más VoIP készülékekkel szemben. Válaszában kiemelte, hogy „az Aplio/Phone és az Aplio/Pro PC-független termékek, mindkettőnek kiváló minőségű a hangátvitel, használatuk pedig egy tízéves gyerek számára sem okozna gondot. Alapelvünk szerint több cég számos lehetőséget teremthet. A Linux széles körben elterjedt, nyílt forrású operációs rendszer, segítségével gyorsan elkészíthetjük a prototípusokat, ezeket aztán tovább fejleszthetjük, ellenőrizhetjük, és az ebből leszűrt tapasztalataink alapján újabb termékeket tervezhetünk.”

© Kiskapu Kft. Minden jog fenntartva



Nesze neked szita!

Vannak emberek, akik a végtelenben is folyamatosan keresik a legnagyobbat.

Hogyha a számítógépes tudomány jelenlegi állását illetően szeretnénk borongani, és most itt nem a Windows-rendszerek futtatására-összeomlására gondolok, akkor érdemes átfutni *Peter Neumann RISKS* jelentéseit és a SANS Institute által rendszeresen megjelentetett „sebezhető pontok” listáját. Úgy tűnik, hogy egyetlen operációs rendszer, rendszermag, fordító, könyvtár, szintaktikai elemző és parancs sem mentes ezektől, legyen az bármennyire is „érett”. Most nem ezermilliárdos könyvelési hibákról vagy eltűnt úrhajókról beszélünk, hanem teljesen mindennapos, mindenki számára nyilvánvaló, el nem kapott kivételekről, amelyeket, a véget nem érő figyelmeztetések ellenére, a C fondorlatos tömbhasználatának köszönhetünk.

Kapcsolódó címek

1. Az említett könyvhöz kapcsolódó webhely:
☞ <http://www.realworldlinuxsecurity.com/>
2. Kapcsolódó cikk: „GIMPS Finds Anoter Prime”
Dale Buske–Sandra Keith, Math Horizons, 2000. április.
Honlap: ☞ <http://www.mersenne.org/prime.htm>

Egy nemrégiben megjelent SANS-jelentés az Access, a Word 2000 és az Internet Explorer 5.0+ alkalmazásokban rejtőző veszélyekre hívja fel a figyelmet. A SANS díjakat ajánl fel azok számára, akik „gyors” megoldásokkal szolgálnak, ezek a hibák azonban meglehetősen „sajátságosak”. A Microsoft, nézzünk szembe a ténnyel, több mint egy profi programozókból álló csapat, akiket a legfrissebb fejlesztési tanfolyamokon képeztek ki. A Microsoft kiadója (a Microsoft Press) sorra ontja magából azokat a könyveket, amelyek arról szólnak, hogy miként készíthetünk bizonyíthatóan nagy teherbírású alkalmazásokat. Azonban úgy tűnik, hogy a „nyilvánvaló” hibák (a memóriaszivárgások, a túlcsoordulások) kiszűrésére szolgáló nyelvi, illetve fejlesztőeszközök mit sem érnek; a MS programozóinak ez az elit csoportja még így is tud hibázni. A hatalmas kihívás abban rejlik, hogy mielőtt nekilátnánk a hibák kiirtásának, meg

kell tudnunk határozni azokat. Még Stephen Hawking igen tágran értelmezhető *időhisztográfia*ja sem teszi lehetővé számunkra, hogy simára csiszolt kódunk minden egyes if–then–else ágát végignézzük. Egyáltalán mit jelent az „emberi” nyelvre lefordítva, hogy kódunk megfelel az elő- és utófeltételeknek? (Ha a Hawking-féle időmegfordítás az Ősrobbanáshoz vezet vissza bennünket, akkor a GOTO-k helyén COMEFROM-okat látunk majd?) A nyílt kódú megközelítés sikere azon alapul, hogy olyan elfogulatlan programozók is hozzáférhetnek a kódhoz, akikben tényleg a cél elérése dolgozik, szemben azokkal a belső „szolgákkal”, akik úgy gondolják, hogy részvényük nem ér annyit, hogy komolyabb erőfeszítéseket tegyenek.

Míndezen ellenére, a Linux rendszermagjában nemrégiben felfedezett biztonsági hibák azt jelzik, hogy még a többször, gondosan átnézett, egyszerű kódrészletek is okozhatnak meglepetéseket. A dolgnak természetesen megvannak a jó oldalai is: (i) a hibákat gyorsan és nyíltan beismerik, kijavítják; (ii) a bűnbakok listája végtelen – fel a kezekkel, bárki is volt az a száznegyvennégyezer ember közül. *Peter Salus* feljegyezi majd a vallomásukat. A témához kapcsolódik *Bob Toxen* hamarosan megjelent könyve, a *Real World Linux Security* (lásd a kapcsolódó címeket).

GIMPS

Ugorjunk át a széles körben elterjedt, nyílt számítástudomány egy másik alkalmazási területére. A GIMPS (Great Internet Mersenne Prime Search – Nagy Internetes Mersenne-prím keresés) programban több mint 8000 egyéni felhasználó vesz részt, akik kivétel nélkül arra kötelezték el magukat, hogy minél nagyobb prímszámokat találjanak. Euklidesz bebizonyította annak idején, hogy a keresésnek sosem lesz vége, mivel ha P a legnagyobb prímszám, akkor P!+1 vagy egy P-nél nagyobb prím, vagy pedig rendelkezik P-nél nagyobb prímszótóval. A dolog tehát bebizonyított!



Azt mondják, hogy ha nem látjuk ennek a bizonyításnak a szépségét, akkor sosem lesz belőlünk matematikus.

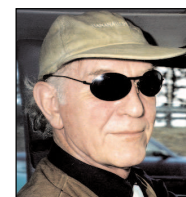
Ha viszont mégsem hiszünk a bizonyításnak, és szeretnénk a saját szemünkkel is meggyőződni annak igazáról, akkor ezt könnyen megtehetjük: Erasztoténész (egy másik halott görög) szítja egy olyan eljárást ad, amelyk az összes prímszámot felsorolja növekvő sorrendben. Saját Kelly-szítám az összetett és a prímszámokat is képes felsorolni egyetlen for-ciklus segítségével.

A kérdés az, hogy vannak-e prímek P és P!+1 között – és 1999. júniusában a GIMPS csapatban tevékenykedő Nayan Hajratwala új (átmeneti) világrekordot állított fel: 2^{6,972,593} – 1

Megkímélem az Olvasót attól a tíz kilométer hosszúságú számsortól, amely ennek a számnak a printf()-fel történő kiírásához szükséges.

A jövő hónapban: hogyan találhatunk idegen világkultúrát a SETI együttműködési programjával. A kis zöld matematikusok talán már ismerik a következő legnagyobb prímet! Addig is látogassanak el a kutatás honlapjára

(☞ <http://www.setiathome.ssl.berkeley.edu/>) és csatlakozzanak a Linux Journal csoporthoz!



Stan Kelly-Bootle (skb@atdial.net) az ötvenes években épített EDSAC I (Anglia, Cambridge Egyetem) óta van jelen a számítástechnikában. Rengeteg cikket és számos könyvet is megjelentetett, többek között a Computer Contradictionary (MIT Press) és a UNIX Complete (Sybex) címűeket. Rendszeresen ír a ☞ <http://sarcheck.com/> és a ☞ <http://www.unixreview.com/> oldalaira.