

Gondolatok kedvencünkről

Azaz mennyire és mire használható a mi kedvenc ingyenes operációs rendszerünk?

Nap mint nap beszélék emberekkel, akik azt kérlik: mire jó ez az ingyenes rendszer? Tavaly erre a kérdésre azt választottam, hogy rendkívül jó hálózati kiszolgálónak. Igen, a Linux már bizonyított a kiszolgálók piacán. Először a fájlkiszolgálás területén. A Samba több vizsgálaton is kenterbe verte vetélytársait. A másik fő terület, ahol egyértelműen kitűnt, a webkiszolgálás. Ma már nyugodtan mondhatjuk, hogy aki linuxos felületen kezd el webszolgáltatásokat fejleszteni (legyen az egyszerű HTML oldalak kiszolgálása, hálózati programok készítése, vagy akár adatbázis-rendszerek hálóra ültetése – lásd a 71–75. oldalon), nem a rendszerösszeomlásokkal és nem a megoldhatatlan feladatokkal fog bajlódni.

Ha valakit nem köteleznek más, egyedi levelezőrendszerek (mint például az Exchange) használatára, gondolom, már rég felfedezte a Linuxot. Ugyanígy vannak ezzel a hálózati szakemberek, akik egyre gyakrabban állítanak fel vállalati hálózatoknál Linuxokat útválasztónak, behíváskezelőnek vagy tűzfalnak. A Linux egyre több helyen bukkan fel, de figyelniük kell a biztonságra is. Az érdeklődőknek ajánlom az 50–53. és a 66–67. oldalon lévő cikkeket.

Kedvencünk már a nagygépek mellett is felütötte a fejét. Ma már akár kiszolgálótelepeket is készíthetünk segítségével, ha nem is tökéleteseket. Ezzel a témával kapcsolatban is találhatunk két érdekes cikket, egy elméleti síkon mozgót, a 61–63. oldalon, valamint egy terheléelosztásra alkalmas fájlrendszerrel (a PVFS-ről) szólót, az 56–60. oldalon.

Rendben, a Linux megmutatta, hogy a kiszolgálók piacán megállja a helyét. De a Linux nem csupán ennyi. Rendkívüli testreszabhatóságának köszönhetően felfedezték a beágyazott rendszerek fejlesztői (legutóbbi konferenciájukról olvashatunk a 12–14. oldalon), és olyan peremterületekre is betört, mint a rádió (lásd az előző számot), az

autóvezérlés, vagy a valósídejű rendszerek. Elég szép teljesítmény ettől a pár évvel ezelőtti csak „tanulmányi segédeszköz”-ként emlegetett rendszertől.

De nézzünk körül egy kicsit az irodákban és a lakásokban. Ezen a területen – valljuk be őszintén – nagyon sok fejlődnie volt a Linuxnak. Sokan elemezgették, hogy miért is nem tört be a Linux erre a piacra. Az elemzések szerint az okok a következők:

- az informatikában dolgozók nem ismerik a rendszert, és mint minden újtól, ettől is félnek;
- nem biztosítja az ismert felületet, a kényelmi szolgáltatásokat;
- nem támogatja eléggé az új alkatrészeket és megoldásokat;
- nincs megfelelő programkínálat (nem is beszélve a játékok hiányáról);
- bonyolult telepíteni, frissíteni és végül
- nincs megfelelő terméktámogatása.

Az év elején még körülbelül két évre becsültem azt az időt, amire a Linuxnak szüksége van a felzárkózáshoz. Véleményem szerint az elmúlt egy év alatt a munka oroszlánrésze elkészült. Nézzük végig a témaköröket.

Nem ismerik a rendszert

Szerencsére felismerte az oktatás (és a minisztérium is) a Linux fontosságát, és rengeteg helyen tanulják és oktatják. Habár még messze vagyunk attól, hogy a Linux általánosan ismert rendszer legyen, mégis, egyre több oktatóközpont veszi fel kínálatába tantárgyként és egyre több cég fogadja örömmel az önéletrajzokban.

Új alkatrészek támogatása

Ez a vád sem a linuxosok miatt alakult ki. Az igazat megvallva, amíg nem hitték el a nagy cégek, hogy a Linux mögött komoly vásárlóerő van, egyszerűen nem foglalkoztak vele. Szerencsére ma már nyilvánvalóvá vált, hogy a Linux nem csak afféle hóbort. A komoly alkatrészgyártók ma már a legtöbb esetben segítik a linuxos programozókat. Ha meg az USB-s alrendszerre gondolok, akkor csak a 2.4-es rendszermagban bízok.

A grafikus gyorsítás témakörében is megtették az első lépéseket, a 4-es X felülettel végre megtörtént a nagy áttérés, egyre-másra jelennek meg a meghajtók is.

A piacon lévő kártyákról egy összefoglaló cikket olvashatunk a 76–78. oldalon.



Nehezen használható felület

Mindig is voltak a régivágású unixosok, akik szerint a Linuxban található rengeteg grafikus felület mindent tud, amit kell. Valljuk be, ez azért nem takarta a valóságot.

Egy Windowshoz szokott felhasználó kitéphette a haját egy Ice vagy egy Next láttán. Nem mondom, hogy nem lehet megszokni őket, de az a felhasználó, aki hozzászokott a kényelmes felületekhez, nem akar „visszalépni”. Szerencsére két versenyző nagyon komolyra vette a figurát, a Gnome és a KDE is erősen fejlődésnek indult. Örülök, hogy mindkettőhöz tartozik magyar honlap is.

Programkinálat

Ez a terület csudamód fejlődésnek indult. Nem elég, hogy sok-sok szövegszerkesztő jelent meg (például AbiWord, KWord, hogy visszatérjek a két grafikus felülethez) a többi irodai programmal karöltve, de a Gimp mellett feltűntek más grafikus és animációkészítő programok, több zeneszerkesztésre és stúdiómunkára alkalmas program is elérhető és végre megjelentek a játékok Linux alá! Aki nem hiszi, olvasson bele januári számunkba!

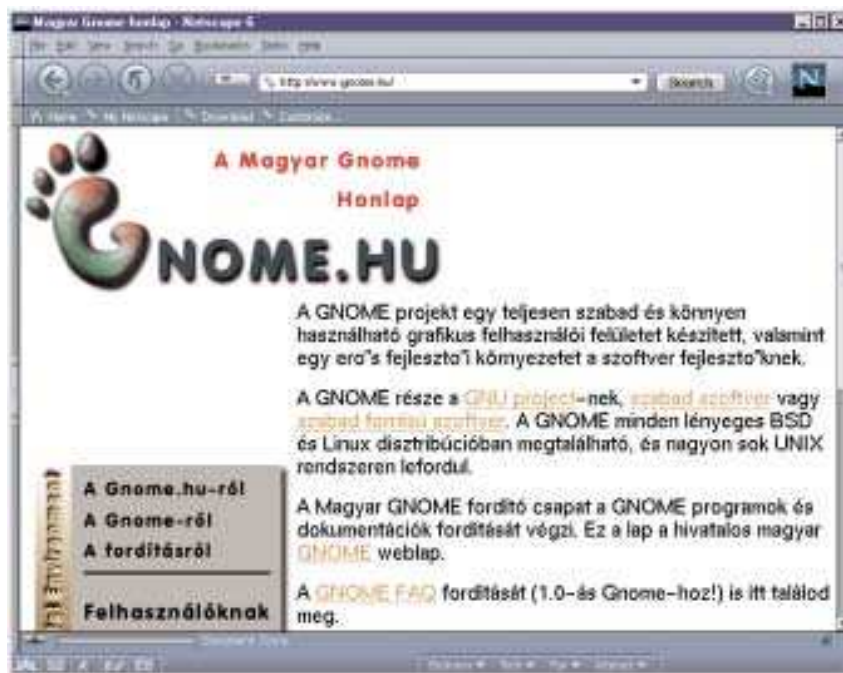
Bonyolult telepíteni

Mindkét fő csomagkezelő rendszerhez – Red Hat-vonal RPM-je és a Debian apt-je – tartoznak már könnyen használható és átlátható telepítőeszközök (a Debian csomagkezelőjéről írtunk a 39–41. oldalon). Mindkét csomagkezelővel könnyen és hatékonyan tudjuk telepíteni és karbantartani az operációs rendszert.

Támogatás

Valóban, a Linux legnagyobb erőssége – vagyis hogy nem egy cég tulajdona – ebben az esetben gyengeségnek tűnik. Szerencsére azonban a linuxosok legtöbbször szívesen ad tanácsot, segítséget, vannak levelezési listák és egyéb fórumok is. Nem mondom, hogy ez a megoldás is olyan kényelmes, mintha egy külső céget hívhatunk ki bármilyen hiba elhárítására, viszont általában annyiival olcsóbb, amennyibe az említett külső cég kerülne. Egy kis utánajárással vagy keresgéeléssel mindenki megtalálhatja kérdésére a választ.

Még egy témát kell boncolgatnunk. A nyelvet. Véleményem szerint, amíg nem jelenik meg egy jó nyelvhelyesség-ellenőrző rendszer Linux alá, addig bizony a vállalatvezetők inkább kiadnak egy vödör pénzt, mert így reménykedhetnek, hogy a fizetős program az egyébként kellemes látványt nyújtó titkárnyó csapnivaló helyesírását egy kicsit javítja. Vannak kezdeményezések, több csapat is elkezdte az „igazi” fejlesztését, de sajnos komoly eredmény még nem mutatkozik. Ha valaki tud ilyenről, kérem, azonnal szóljon! Addig is, amíg elkészül a várva várt ellenőrző, jobb, ha mi magunk kezdjük el a szavakat és a fogalmakat tisztázni. Előző számunkban írtam, hogy milyen fontosnak tartom a szakma jövője szempontjából egy tiszta magyar nyelvhasználat kialakítását, és örömmel vettem a támogató visszajelzéseket. Bár volt másféle tapasztalatom is... Egy Linux-változat egyik hazai terjesztője például, amikor beszélgettünk, kérte, hogy küldjem el neki az általunk használt fordítási listát. „Sok kérdéses szó van még a szakma nyelvében. Disztribúció, iteráció, konfiguráció, bootlemez, netkártya, funkció, ...” mondtam én, erre ő: „Igen, igen, ezeket használjuk mi is.” „Nálunk ez mind



tiltott szó”. Azóta sem került szóba ez a kérdés köztünk. Kíváncsi vagyok ezek után, hogy milyen lesz az a magyar nyelvű változathoz készülő kézikönyv, amit ők fordítanak. Ebben az esetben már nem metség, hogy „Itt mindenki a szabadidejét feláldozva dolgozik, örüljünk, ha dolgoznak”, ugyanis ezért komoly pénzeket fognak kiadni az emberek. És ha valami pénzbe került, a pénzünkért minőséget várunk. De mi is tehetünk a nyelvért, kérek mindenkit, akinek van egy kis ideje, lapozzon el az 63. oldalon lévő Szókinctárhoz, és ha van jó ötlete, vagy véleménye valamelyik kifejezéssel kapcsolatban, küldje azt be nekünk. Előre is köszönöm.

Merre tovább?

A Linux fejlődik, ugyanezek a lépcsőfokokon a legtöbb honosított rendszer is végigbuktácsolt, és a Linux javára legyen mondva, legtöbbjüknek sokkal hosszabb ideig tartott. Ha a Linux ilyen gyorsasággal fejlődik, akkor egy év múlva nem az a kérdés, hogy bejut-e a Linux az irodákba, hanem az, hogy melyik változat lesz a legszélesebb körben használt.

És nem csodálkoznék azon sem, ha hamarosan az amerikai választási hadjáratokhoz hasonló hangnemben jeleznének meg a hírdetések: „Ha az Iksz változatot választja, a gyermeke jövőjét alapozza meg...” Az emberiség imádja a háborúkat, ez alól nem kivétel a Linux-társadalom sem. A legutóbbi Linux Mikuláson például megrendezték a Disztribúciós (na tessék, megint leírtam...) háborút is, ami lényegében hangerőre ment. Nyilván nem ez a megoldás, hiszen nem véletlenül létezik ilyen sokféle változat, mindegyiknek megvan a maga erőssége, és megvan a maga gyengéje is. Itt az ideje, hogy összehangoljuk a munkát, és létrehozunk egy olyan alapot, amit mindegyik hazai Linux-változat elfogad!



Szy György a Linuxvilág főszerkesztője. A lappal kapcsolatos kérdéseket, kéréseket, ötleteket szeretettel várja az alábbi levélcímen:
Szy.Gyorgy@linuxvilag.hu

Programvadászat

Kedves Olvasóink, ígéretünkhöz híven a laphoz mellékeljük a Debian GNU/Linux 2.2 Potato harmadik lemezét, így válik teljessé a Debian-változat telepítő anyaga. A negyedik korongra pedig olyan anyagok kerültek fel, amelyekkel rendszerüket frissíthetik. Helyet kaptak rajta a Netscape különböző változatai, a Helix Gnome rendszer, valamint számos forrás, így mindenki saját igényének megfelelően telepítheti a programokat. A játékok szerelmeseinek pedig a Tuxraceret ajánljuk.

Debian GNU/Linux 2.2 III.

A lemezen található programok forráslistához adásával – ezt az apt-cdrom add -a paranccsal tudjuk megtenni – számos új csomag válik elérhetővé. Ilyen például az aolserver, ezt az AOL gépei futtatják webkiszolgálóként. Emellett megkapjuk az aolserver-postgres csomagot is, amellyel csatlakozhatunk a PostgreSQL-adatbázisunkhoz, illetve abból adatokat nyerhetünk ki. A diskless csomag segítségével egyszerűen alakíthatunk ki lemezegység nélküli munkaállomásokat a hálózati fájlrendszer (NFS) használatával. Az everybuddy programot azok fogják értékelni, akik több internetes csevegőszolgáltatást is igénybe vesznek, ugyanis ennek segítségével lehet használni az ICQ, a Yahoo chat és az AOL Instant Messenger rendszereket is.

Rendszermag

Lapzártánkkor a 2.4-es sorozatú rendszermag fejlesztése még tart, de a legutolsó változatot feltettük a lemezre. Természetesen az elérhető legfrissebb, megbízható mag, a 2.2.17-es is szerepel korongunkon.

Netscape 6

A Netscape 6-os böngésző Mozilla alapokon nyugszik. Legalább 64 MB memóriát és 133 Mhz-es processzort igényel, az ajánlott azonban egy izmosabb gép 128 MB memóriával. A grafikus telepítőfelületnek köszönhetően nagyon kényelmesen végrehajtható a telepítés. Könnyű beállítani, sőt, a már telepített Netscape beállításait is át tudja venni.

Beonex

Hasonló tudású és erőforrás-igényű a Beonex communicator, melynek alapja szintén a Mozilla. Egyelőre még csak a 0.6-os változatnál tart, ez jelzi, hogy nem a teljes, megbízható változatról van szó, de már ez is elég szépen működik. Mindenképpen ajánlom a vállalkozó kedvű felhasználóknak. A Mozillánál is bevezették a bőrök (skins) használatát, ezzel teljesen új felületet adhatunk kedvenc böngészőnknek. Mi is mellékelünk néhányat, ezek használhatók mind a Netscape 6-osban, mind a Beonex communicatorban. Ha böngészőnkből nyitjuk meg az adott fájlt, akkor a telepítés automatikusan lezajlik és jöhet a börcsere.

Netscape 4.76

A kisebb erőforrással bírónak továbbra is megmaradt a négyes sorozatú Netscape, ennek a legfrissebb változata is felkerült a korongunkra. Egyszerűen ki kell csomagolni, valamint rendszergazdaként futtatni a telepítő héjprogramcskát. A telepítést karakteres felületen végezhetjük.

Gnome Helix

Lemezünkön a legnagyobb terjedelmű anyag a Helix Gnome grafikus felület. Mind rpm, mind forráskód formájában megtalálható. Ennek a változatnak nagyobb felhasználói csomagok is összetevői:



- Gnumeric táblázatkezelő
- AbiWord szövegszerkesztő
- GIMP képszerkesztő
- Gphoto digitáliskamera-kezelő
- XMMS multimédia-lejátszó.

PostgreSQL 7.0.3

Erős, nagy tudású, megbízható adatbázis-kezelő. Használhatjuk grafikus és karakteres felhasználói felülettel is. Az előbbi könnyen áttekinthető, de sajnos nem minden szolgáltatás érhető el belőle. Kisebb

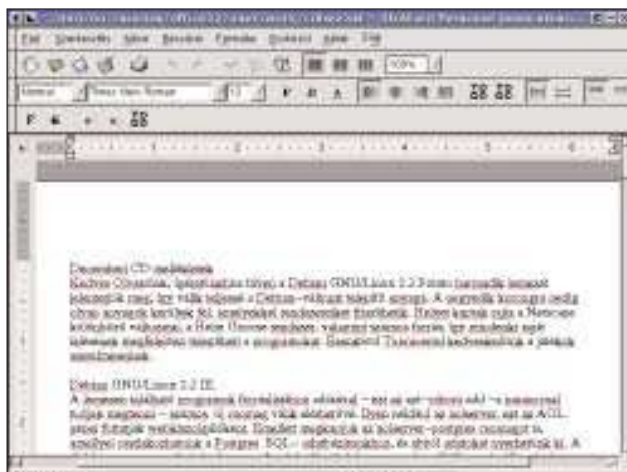


adatbázisoktól egészen a több gigabájtnyi adatot felolgozó rendszerekig használható. Ha sok adattal dolgozunk, akkor memóriánk ajánlott mérete 96 MB. Korongunkra felkerült forrásként és rpm formátumban is. Segítségével akár internetes boltot is készíthetünk, ehhez jó kiindulópont lehet a 71. oldalon található cikk. Könnyedén létrehozhatunk bármilyen, igényeinknek megfelelő, saját grafikus vagy karakteres felhasználói felületet, például

böngészőn keresztül adatbevitelt és adatbázis-felügyeletet is. A legtöbb ismert programnyelvhez – ilyen például a C/C++, Perl, PHP stb.– elkészítették a programozási könyvtárakat, melyek nagymértékben megkönnyítik munkánkat. A programmal érkező leírás hasznos segítség lehet terveink megvalósításához.

Iroda

Az AbiWord ígéretes választásnak tűnik a szabad forrású szövegszerkesztők közül. Számos rendszerhez elérhető. Ilyenek például: a Windows, a Linux/Intel, a Linux/PowerPC, a Linux/Alpha, a BeOS/Intel, a BeOS/PPC, a FreeBSD/Intel, a Solaris, a NetBSD, és az AIX.



Az AbiWord alapvető tulajdonságai:

- bekezdésigazítás,
- helyesírás-ellenőrzés,
- Word97 és RTF dokumentumok beolvasása,
- RTF, Text, HTML és LaTeX fájlformátumokba mentés,
- stíluskezelés,
- hasábok és képek kezelése stb.

Egyszerűen, gyorsan használható és kis erőforrásigényű program. Sajnos, a magyar nyelvű helyesírás-ellenőrzést a program még nem tartalmazza.



Tuxracer

Pingvinverseny a javából! Gyorsasági lesiklóverseny, ügyességi feladatokkal megtűzdelve. Versenyezhetünk ködben, hóban, szélben, sőt, még éjjel is. Ha a helyi versenyeket megnyerjük, akkor további pályákon is kipróbálhatjuk tudásunkat. Csak a leggyorsabb, legokosabb és legszerencsésebb nyerhet!

Rendszerkövetelmények

- Processzor: Pentium 200MMX,
- Memória: 64 MB,
- Merevlemez: 20 MB szabad tárhely,
- Grafikus kártya: 3D-s grafikus kártya OpenGL-támogatással,
- Hangkártya: 16 bites.

Ajánlott rendszerkövetelmények

- Memória: 128 MB,
- Merevlemez: 20 MB tárhely,
- Hangkártya: 16 bites,
- Grafikus kártya: TNT2/Voodoo3.

Az irányításhoz egér és billentyűzet szükséges, botkormány illetve játékpad is használható. Mivel a lemezre a program forráskódja került fel, ezért mindenkinek otthon kell a fordítást elvégeznie.

Rendszerünk karbantartása

A Debian Linux 2.2-höz rengeteg frissítés jelent meg, ezeket összegyűjtve közreadjuk, hogy olvasóink rendszere mindig a legkevesebb hibát tartalmazó változat legyen. E lemezen fellelhető csoma-



gokat szintén az apt-cdrom add -a paranccsal lehet a forráslistához adni. Ezután az apt-get upgrade paranccsal frissíthetjük fel rendszerünket. A telepítés folyamán minden olyan csomagot frissít, amit előzőleg már feltelepítettünk. Mindazoknak megfelelő megoldást jelenthet az apt, akiknek van

internetkapcsolata, hiszen hatékony eszköz lehet rendszerük automatikus frissítéséhez. Közelebbi megismeréséhez ajánlom a Debian csomagkezeléséről szóló cikkünket, mely a 39. oldalon található.

Maverik

Korongunkra rákerült az előző számban szerepelt Maverik 3D-s modellező környezet és a működéséhez szükséges MESA 3D könyvtárak forrása is. Aki kedvet érez a kipróbálásához, bátran fogjon hozzá. Az alábbi képek is igazolják a nagyfokú teljesítményt és sokoldalúságot.

Apache webkiszolgáló

Végül megemlítem az Apache webkiszolgáló legújabb, megbízható változatát, az 1.3.14-est. A legfőbb változások az előzőekhez képest: kijavították a mod_www_alias és mod_rewrite modulok biztonsági hibáit. A fejlesztők mindenkinek, de legfőképpen az 1.2-es sorozat használóinak ajánlják az új változatra frissítést. Az előző számunkban megjelent *Az Apache beállítása, trükkjei és hibakeresés* című cikkünk nagy segítséget nyújthat az Apache fordításához, beállításához és biztonságos üzemeltetéséhez.



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.

© Kiskapu Kft. Minden jog fenntartva

Debian-újdonások

Hivatalosan is bejelentették 2000. december elején a Debian GNU/ Linux 2.2rev2-es kiadását. Ez az eddig felderített hibák teljes körű javítását tartalmazza. Az apt használatával természetesen lehetőségünk nyílik rendszerünk frissítésére.



Két új internetes oldallal örvendeztették meg a Debian-kedvelők tábort.

➔ www.debianhelp.org-on, mutatja, sok-sok segítséget kaphatunk akár telepítési, akár beállítási gondok megoldásához.

A ➔ www.debianplanet.org-on pedig majdnem az összes Debiannal kapcsolatos hír és adat fellelhető. Magyar nyelven ezek sajnos nem érhetőek el.

KDE

KDE 2.0.1

Hat héttel a KDE 2.0 megjelenése után kiadták a 2.0.1-es változatot. A két legfontosabb változás a bővebb leírás és változások listáján néhány hibajavítás.

KDevelop 1.3

A KDevelop egy C/C++ grafikus fejlesztői felület a KDE környezethez. Ebben a változatban már fejleszthetünk KDE 2.0 alá is programokat. Letölthető a KDevelop honlapjáról futtatható formátumban, de természetesen forráskódban is.

- ➔ www.kde.hu
- ➔ www.kde.org
- ➔ www.kdevelop.org
- ➔ www.koffice.org
- ➔ www.konqueror.org
- ➔ kde.themes.org



OpenBSD

December elején megjelent az OpenBSD 2.8. Bővebb alkatrész-támogatást, további biztonsági kiegészítéseket, valamint beépített titkosítást tartalmaz.

Főbb jellemzői:

XFree86 3.3.6-current, gcc 2.95.3, Perl 5.6.0 hibajavításokkal, Apache 1.3.12 (+hibajavítások), Mod_ssl 2.6.2, OpenSSL 0.9.5a, DSO támogatással, ipf 3.3.18, groff 1.15, sendmail 8.10.1, lynx 2.8.2 HTTPS-támogatással, sudo 1.6.3p5, ncurses 5.2, legfrissebb KAME Ipv6, KTH Kerberos 1.0.2, OpenSSH 2.3.0.

VMware kiszolgáló

Eddig a VMware programmal gazdarendszerünk erőforrásait használva, újraindítás nélkül tudunk futtatni más operációs rendszereket, így windowsos programjainkat is tudtuk használni Linux alatt, és viszont. A program mostantól kiszolgálóként is elérhető, így egyszerre több gép hozzáférését is biztosítani tudja a rajta futó alkalmazásokhoz. Két változatban kapható. Az egyik a VMware GSX Server, ez csak Linuxon fut. Nyolc processzorig bővíthető a gazdagép, és processzoronként négy virtuális gépet tud futtatni. 400 MHz processzor és 128 MB RAM az alapkövetelmény.



A másik változat, az operációs rendszertől független VMware ESX Server, közvetlenül használja a gazdagépet. Jóval több követelménynek felel meg, mint a VMware GSX Server, ez természetesen erőforrásigényében is megmutatkozik. Itt már legalább 500 MHz-es processzor, 256 MB RAM és SCSI merevlemez szükséges.

- ➔ www.vmware.com

Opera ingyen

Az Opera böngésző igyekszik felzárkózni a „nagyok” mellé. Méretét és tudását egybevetve egyértelműen versenyképes, kezeli a HTML és az XML nyelveket, a stíluslapokat, a JavaScriptet, a hálózati és a biztonsági protokollokat (HTTP, SSL, TLS) stb. Mérete azonban messze a versenytársaké alatt marad. December eleje óta ingyenesen letölthető a cég honlapjáról.

Sajnos, a linuxos változat csak bétaállapotú, de már így is jól használható.

- ➔ www.opera.com



A rendszermag újdonásai

December elején megjelent a megbízható 2.2.18-as rendszermag.

A tizenkettedik próbaváltozat talán az utolsó lesz, mielőtt az év vége előtt megjelenik a 2.4-es sorozatú rendszermag. Számos újdonást tartalmaz, többek között: többprocesszoros működés támogatása 32 processzorig, 64 GB memória használata, 3D gyorsítókártyák és USB támogatása. A rendszermag lecserelésével elérhetővé válik a ReiserFS naplózó fájlrendszer is.

- ➔ www.kernel.org
- ➔ linuxgram.com/newsitem.phtml?sid=109&aid=11420

Gnome

Balsa 1.0

December elején megjelent a Balsa 1.0, a Gnome környezet levelezőprogramja, immár megbízható állapotú. Kezelése, beállítása egyszerű, képes egyszerre több postafiókot kezelni. Ez a szolgáltatás azok számára előnyös, akik több címmel rendelkeznek, viszont szeretnék egy helyen kezelni az összes postafiókjukat.

Gnome-DB

A Gnome-DB segítségével a Gnome alatt futó alkalmazásaink számára tehetjük elérhetővé adatbázisaink tartalmát. Csatlakozhatunk PostgreSQL, Oracle, MySQL stb. felületekhez.

- ➔ www.gnome.hu
- ➔ www.gnome.org
- ➔ www.gnome-db.org
- ➔ www.gnome-db.org/download.php
- ➔ www.lelixcode.com

Gnome Windowszo

A Linux forráskód újrafordítása Windows alatt nem éppen a Microsoft legkedvesebb álma. Közel négy millió sor a Gnome kezelőfelületből és alkalmazásokból, egy ember másfél heti munkája, és máris használhatjuk a Gnome felületet Microsoft Windows alatt is. Mindeközben alig száz sort kellett átírni. A nagy segítséget a U/WIN környezet nyújtotta. A program segítségével unixos forráskódot lehet könnyedén Windows alá vinni. Ennek köszönhetően elkészült a Gnome átültetése Windows alá.

- ➔ www.linuxpr.com
- ➔ www.gtline.com

Windows Linuxhoz

Linuxot használva sokszor előfordul az emberrel, hogy valamit szeretne megnézni Windows alatt, vagy egyszerűen csak szüksége van egy programra egy bizonyos feladat elvégzéséhez. Eddig az volt a jól bevált módszer, hogy a gépet újraindítva beléptünk a windowsos környezetbe, majd ismételt újraindítással térhettünk vissza a Linuxhoz. Ha újraindítás nélkül szeretnénk használni windowsos programjait linuxos környezetben, akkor a Win4Lin-t nekünk találták ki. Látszólagos gazdagépet nyújt, így Windowsunk biztonságosan futhat egy Linux-folyamatként. Futtathatunk például Adobe Photoshopot, MS Internet Explorerst stb.



A követelmények: Linux 2.2.x rendszer-maggal, CD-ROM-meghajtó, 32 MB RAM a Windows számára, 15 MB lemezhely a Win4Lin számára.

- ➔ www.lin4win.com

Góliát

A www.goliat.hu címen a kizárólag magyar internetes oldalak közötti keresésre szakosodott keresőprogramot és egy hivatkozásgyűjteményt – Témakörök néven – talál az ide látogató. A Góliát célja, hogy eddig ismeretlen távlatokat, teljesítményt és határfokot bocssáson a hálón kereső felhasználók rendelkezésére. Nemzetközi viszonylatban is egyedi, pontos, megbízható és Magyarországra értendően átfogó találatokat kap a Góliát használója.

A program teljes egészében magyar fejlesztés és közel egyéves munka eredménye, operációs rendszere Debian GNU/Linux. Tulajdonosai a fejlesztők, ennek köszönhetően folyamatos a fejlesztése és a karbantartása. A program használata teljesen ingyenes. Saját építőrobotjai követik nyomon az Internet bővülését, így a teljes adatbázis folyamatosan cserélődik és frissül.

A Góliát főbb tulajdonságai:

- gyors bejegyzés (általában 40-45 másodperc)
- gyors keresési folyamat (fél másodperc alatt),



- Magyarország legteljesebb, valamint legnagyobb folyamatosan bővülő URL listáját (kb. 4,8 millió URL) tartalmazza az adatbázis.
 - a Góliát robotjai igyekeznek kiszűrni a régóta nem élő hivatkozásokat, utalásokat.
 - a magyar ékezeteket nyelvhelyesen kezeli, azaz ha a keresőmezőben a GYŐR van, valóban a Győr szót keresi és találja meg.
- ➔ www.goliat.hu

Linuxos játékok

Az SDL (Simple DirectMedia Layer) több rendszertípuson is használható multimédia-könyvtár, ami gyors hozzáférést nyújt a grafikus kártyához és a hangkártyához. Használható MPEG visszajátszáshoz, emulátorokhoz és számos játékhoz. Ilyen például a Civilization: Call To Power, díjnyertes linuxos játék.

Az SDL elérhető Linux, Win32, BeOS, MacOS, Solaris, IRIX és FreeBSD alá. A könyvtárral számos játék használható, közöttük például a második lemezünkön szereplő Tuxracer is. A teljes listát a http://www.libsdl.org/games_db/games.php3 oldalon találjuk, ahol kiválaszthatjuk az operációs rendszerünknek megfelelő, valamint megtekinthetjük a játékok készütségi fokát is.

- ➔ www.libsdl.org



Alicebot

Aki ellátogat a www.alicebot.org internetes oldalra, kipróbálhatja, mit jelent egy értelmes géppel „társalogni”. Az ALICE név az Artificial Linguistic Internet Computer Entity-ből származik. Alkotói Java nyelvet használtak megteremtésére, ezzel is bizonyítva a Java erejét. Alicebot hatalmas tudásbázisát folyamatosan bővíti(k). Ingyenesen letölthetjük, felhasználhatjuk a GPL szabályozás alapján. Saját személyiséget adhatunk neki és a mi alicebotunk is a virtuális társadalom tevékeny, megbecsült tagjává válhat.



Munkaerő-piaci folyamatok

Linux vagy BSD?

El tudná valaki magyarázni, hogy miért használják Linuxot BSD helyett, azonkívül, hogy a Linuxnak jól hangzó neve van?

Nem. Ennyi az egész.

Ez a név nagyon király!

Roppant keményen dolgoztunk egy olyan név kitalálásán, amely tetszik az emberek többségének, és ez a munka minden kétséget kizáróan meghozta a gyümölcsét. A Linuxot használó emberek ezrei mondhatják: „OS/2? Ugyan már! Nekem Linuxom van. Milyen klassz név!”

A 386BSD készítői elkövették az a hibát, hogy számokat és érthetetlen rövidítéseket tettek a nevébe, és ezzel egy csomó embert elijesztettek, mert ez már túl bonyolult a számukra.

– Linus Torvalds válasza egy Linuxszal kapcsolatos kérdésre

Közismert tény, hogy idén lelassult a betöltetlen állások számának növekedése az informatika területén. A Szilíciumvölgyben az eddigi elbocsátásokat ötezer és 15 ezer fő közöttre becsülik. Az elbocsátások a megváltozott igények logikus következménye. Először a meghirdetett állások száma esett vissza, aztán létszámstop következett, majd jöttek az elbocsátások. Idén áprilistól – amikor a betöltetlen állások száma az addigi legmagasabbra rúgott –, öt hónap leforgása alatt mostanra eljutottunk az elbocsátások korszakába.

A helyzet azért nem egyértelműen gyászos. Az 1. ábra szerint a szabad programmérnöki álláshelyek száma növekszik a Szilíciumvölgyben és Amerikában egyaránt. Az üres álláshelyek számának csökkenése szintén lassuló irányzatot mutat.

Az ábrán szereplő értékeket az első havi helyzetet alapul véve számítottuk ki, vagyis a 2000. januárhoz tartozó érték: 1,00.

Technológiák

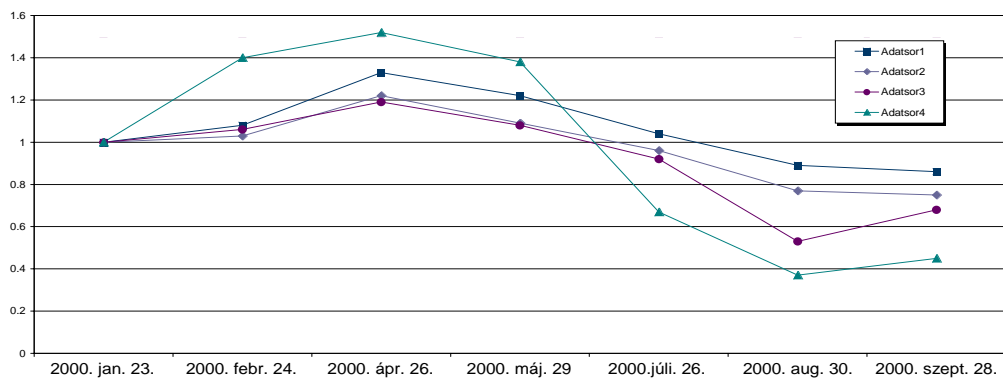
Amióta a technológiai folyamatokat nyomon követjük, először fordult elő, hogy egy, az Internethez nem kapcsolódó technológia iránti kereslet felülmúlta az internetes szaktudás iránti igényt. A 2. ábra szerint általános visszaesés tapasztalható az Internettel kapcsolatos szaktudás iránt. Ez egybevág azzal a megfigyeléssel, hogy az úgynevezett dot.com vállalatok növekedése lelassult. Noha ábránkon nem ezt mutattuk be, mégis érdemes megjegyezni, hogy az egyetlen említésre méltó növekedést mutató terület a linuxos jártasság iránti igény. A kereslet olyan erősen növekszik, hogy az elmúlt tizenhat hónapban a linuxos szaktudás iránti érdeklődés minden mást felülmúlt.

Reginald Charney jelenleg a C és C++ Felhasználók Szövetségének amerikai szekcióját vezeti.

További információkért keresse fel a honlapjukat:

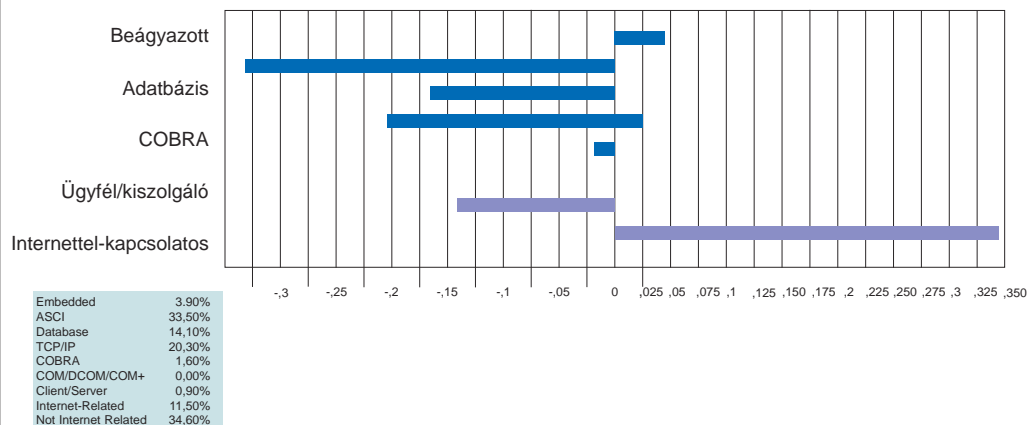
➔ <http://www.accu-usa.org/>.

Munkahelyek számának növekedése



00/ jan. 23.	00/ febr. 24.	00/ ápr. 26.	00/ máj. 29.	00/ júli. 26.	00/ aug. 30.	00/ szept. 28.
1	1,08	1,33	1,22	1,04	0,89	0,86
1	1,03	1,22	1,09	0,96	0,77	0,75
1	1,06	1,19	1,08	0,92	0,53	0,68
1	1,40	1,52	1,38	0,67	0,37	0,45

Technológiai irányzat



Linux Afrikában

Péntek délutánként *Githogori wa Nyangara-Murage*, a Xerox Kutatóközpont (Xerox Research Center) volt alkalmazottja, rendetlen irodájában ingyenes számítástechnikai előadásokat tart fiatal programozók számára...

A Linux egyre élénkebb érdeklődést kelt az afrikai szakemberek körében. Jó néhány Linux-felhasználót tartanak számon, még a földrész legtávolabbi csücskeiben is.

„A Szabad Szoftver alkalmazásának van értelme Afrikában. Lehetővé teszi, hogy Afrika a világ többi részével egyenrangú fél lehessen” – mondja Githogori. „Ha az óriáscégek termékei jelentenék az egyetlen lehetőséget, Afrika számára a számítástechnika elérhetetlen álom lenne, mert nincs a földrészen elég pénz ezekre a programokra.” Kivonat *Nick Wachira* Afrika, a linuxosok földrésze? című cikkéből.

➔ <http://www.wired.com/news/technology/0,1282,38749,00.html>

Ötletadó

Klónozni akarsz egy merevlemezt? Használ a tar parancsot. Kösdd be a klónozandó merevlemez a rendszeredbe (a gépet állítsd le a művelet alatt). Indítsd el a gépet. Rendszergazdaként lépj be a gyökérfájlyvtárba (cd /). Illeszd be az új merevlemez a /mnt könyvtár alá, majd futtasd le a következő parancsot:

```
$ tar cJf - . | ( umask \
0; cd /mnt; tar xvf - )
```

c = létrehozás (create)

l = csak a lokális fájlrendszert (ne lépje át a fájlrendszer határait).

f = fájl (a következő argumentum a tar fájl neve vagy „-“)

- = írj a szabványos kimenetre vagy olvasd a szabványos bemenetről

x = kicsomagolás (extract)

v = bőbeszédű üzemmód (verbose)

Az „umask 0” alkalmazásával elérhetjük, hogy az új fájlok jogosultságai megegyezzenek a régi fájlokéval.

Ők mondták

„A gondolkodás az emberiség egyik legősibb és leghasznosabb tevékenysége. A gondolat szüli az életet, az életkedvet, a közösséget. Az egyetértés nem számít. Ami viszont fontos, hogy gondolataink termékenyítő ereje teszi elevenné az életet, söpri ki életünkől a berozsdállt gondolatokat és intézményeket.”

(Michael Ventura)

„A zseni az, aki a fiatalok eredetiségét a felnőttek tapasztalatával képes ötvözni.” (Michael Polanyi)

„A dokumentáció olyan, mint a szex: ha jó, akkor nagyon, nagyon jó; de ha rossz, akkor is jobb a semminél.”

(Dick Brandon)

„Az idealizmussal nincs semmi baj, de amint megközelíti a valóságot, a költségek hirtelen emelkedni kezdenek.”

(William Buckley Jr.)

„A régi ócska operációs rendszereknek mégis van egy előnyük. legalább nem kell megszoknunk valami újat.”

(Neal Stephenson)

„Úgy tűnik, hogy az immár 120 éves vállalat nem lesz képes túlélni az elkövetkezendő 25 évet. Jogilag és pénzügyileg igen, de szerkezetileg és gazdaságilag nem.”

(Peter Drucker)

„Akár hiszik, akár nem, az Internetet még mindig nem istenítyük eléggé. Hiszen mi, az internetes üzletben dolgozók, a világ legújabb törvényes vagyonszerző hálózatának vagyunk tagjai.”

(L. John Dorr)

„Az örökké tartó növekedésben csak a ráksejt bízik.”

(Edward Abbey)

„Az egyszerű, tiszta szándékok és elvek az összetett és leleményes viselkedésnek nyitnak teret. A bonyolult szabályok és korlátozások pedig az egyszerű és ostoba magatartásnak.

(Dee Hock)

„Aki igazán ötletgazdag alkotó akar lenni, az ne ismétlje önmagát.”

(Rob Bresny)

„Sokszor teljesen más értelmet ad a szavaknak a köznyelv.

A legtöbb ember szótárában a látványterv csupán mázat jelent. Számomra azonban ez áll tőle a legtávolabb. A látvány és forma az emberi alkotás szíve-lelke.”

(Steve Jobs)

„A legnagyobb veszély nem az, hogy túl magasra tesszük a mércét, és leverjük, hanem hogy túl alacsonyra tesszük, és könnyedén átlépjük.”

(Michelangelo)

„A bizalom mindig vesztes a pénzzel szemben.”

(David Hodskins)

Aláíráskényszer:

„Kövezzetek meg. Állítom, hogy a leggyorsabban úgy oldódnak meg a gondjaim, ha feladok egy siránkozó üzenetet a hírcsoportokba: néhány pillanat alatt megkapom a megoldást és közben még a Neten is lóghatok...Huh...”

– Dave Phillips,
dlphilp@bright.net,
a nehézségek kiküszöböléséről a hírcsoportok segítségével

www.kiskapu.hu

Magyar és angol nyelvű számítástechnikai szakkönyvek boltja

Beágyazott rendszerek tanácskozás, 2000



Melyik és hány vállalat sorolta magát a beágyazott Linuxot használók közé egy évvel ezelőtt? A beágyazott piac jelentős átalakuláson megy keresztül a Linuxszal vagy anélkül.

Az ideai beágyazott rendszerek tanácskozáson a Linux a legtöbb pavilonban szinte kötelezően helyet kapott. Lássuk, mi történt a tavalyi tanácskozás óta.

Ha visszamennék az időben, a tavaly szeptemberben tartott Embedded Solutions Conference-ig, láthatnánk, hogy rövid egy évvel ezelőtt a beágyazott linuxos piac még nem is létezett. Kétségtelen, hogy egyre több fejlesztő, valamint maroknyi cég foglalkozni kezdett a beágyazott Linuxszal, de egyszerűen még nem jelent meg külön figyelmet érdemlő területként, létező piacként.

Melyik (és hány) vállalat sorolta magát a beágyazott Linuxot használók közé egy évvel ezelőtt? Vajon mennyi beágyazott Linuxról szóló sajtóközlemény jelent meg, illetve hány termékbemutató volt napirenden a tavalyi tanácskozáson? Nem sok. Egy évvel ezelőtt a beágyazott Linux viszonylag ritka jelenség volt, és az a lehetőség, hogy egyáltalán találkozhattunk vele, nem a hirdetési elképzelések vagy a reklámtervek, hanem az újítások iránt nyitott fejlesztők munkájának eredménye. Volt egy-két kivétel, néhány előfutár. Ahhoz, hogy bemutathassam mi ment végbe a beágyazott Linux világban egy évvel ezelőtt, a kedvenc keresési módszeremhez fordultam, és megvizsgáltam a tavalyi tanácskozás idején megjelent főcímeiket. A keresés felfedte a mai beágyazott linuxos piac alapjait. A beágyazott Linux támogató-sárol a következő cégek számoltak be (ábécérendben): Caldera, Cygnus, EMJ, FSM Labs, Lineo, MontaVista, PROSA és Zentropix. A beágyazott Linux előfutárainak első gép- és alkatrészgyártó partnerei: a Force, a JUMPtec, a Megatel, a Motorola Computer Group, a Synergy Microsystems és a Ziatech. A tavalyi rendezvényen három, a beágyazott Linuxszal kapcsolatos bejelentés már előre jelezte az Embedded Linux Consortium későbbi megalapítását:

- A Zentropix életre hívta a RealTimeLinux.org-ot, mellyel megpróbált némi pezsgést és egyetértést hozni a valós idejű linuxos megoldások területére. A Zentropixot később felvásárolta a Lineo.
 - A Cygnus nyilvánosságra hozta a EL/IX API-t – célja a beágyazott Linux több részre szakadásának megelőzése a beágyazott rendszerek piacán belül. A Cygnust a későbbiek során a Red Hat felvásárolta.
 - A Lineo bejelentette az Embedded Linux Advisory Board (EMLAB Beágyazott Linux Tanácsadó Testület –), létrejöttét. Ez egy gyártóktól is független szervezet a Linux beágyazott rendszerekben való használatának támogatására és segítésére.
- Lépünk vissza a jelenbe, és lássuk, mi a helyzet a beágyazott Linuxszal.

Ellenállhatlan erő a piacon

A Linux beágyazása, mely egy évvel ezelőtt elsősorban újító szándékú programfejlesztők tevékenysége volt, növekvő számú kereskedelmi cég tevékenységének közép-pontjába került. Az Embedded Linux Consortium, mely hét hónappal ezelőtt még csak nem is létezett, immár 75 vállalati taggal dicsekedhet. A beágyazott Linux területén a legnagyobb beruházásokat – összesen több százmillió dollár értékben – az ipar olyan nagyjai hajtották végre, mint a Motorola, az IBM vagy az Intel.

Fontos annak megértése, hogy a beágyazott piac jelentős átalakuláson megy keresztül, a Linuxszal vagy anélkül. A Linuxtól függetlenül a fejlesztők egymás sarkát taposva igyekeznek kielégíteni az intranet és internetkapcsolatra vonatkozó igényeket. Igyekeznek kihasználni az új, 32 bites, alacsony költségű, bőséges programkínálattal és flashmemóriával párosuló RISC-processzorok előnyeit. A magas fokon összevont, MIPS, ARM és PowerPC-alapú lapkagép-processzorok (system-on-chip; egyetlen lapkán teljes rendszert tartalmazó processzorok) révén könnyen és olcsón lehet teljes számítógéppnyi működést sűríteni akár a legegyszerűbb és legolcsóbb gépekbe is. Az újonnan megjelent módszerek soha nem látott mértékben javították a beágyazott eszközök képességeit, emellett megemelték az elvárások szintjét is.

Összességében a beágyazott Linux olyan játéktéren jelent meg, mely amúgy is felemelkedőben volt, fejlesztői keményen és gyorsan dolgoztak, a legújabb módszereket adva válaszul a legújabb kihívásokra. Mivel alacsony költségű, nyitott forrású, fejlett lehetőségek rejlenek benne, a Linuxot kiválóan sikerült elhelyezni a piacon ahhoz, hogy meglovalgassa a beágyazott rendszerek tengerének árhullámát. A forráskód nyílt elérhetősége, párosítva az együttes munka és a kapcsolattartás napjaink előre nem látott könnyűségével olyan készítő tényezőkké váltak, melyek lehetővé tették, hogy a fejlesztők gyorsan és hatékonyan alkalmazkodjanak egy gyorsan változó terület kihívásaihoz. A Linux tehát futótűzként kezdett terjedni a beágyazott piacon.

Mindenki Linuxban gondolkodik

A tavaly szeptemberi állapottal szemben, a Linux-támogatás gyakorlatilag mindenhol megjelent a tanácskozáson. Napjainkban szinten minden cégnek van saját linuxos terve – akár arról, hogyan lehetne kihasználni az előnyeit, akár arról, hogy hogyan lehet védekezni ellene. A nem linuxos szereplők, mint a Wind River, a Microsoft és a QNX jelzik, hogy életük – különböző mértékben ugyan, de – küzdelmes lesz abban a világban, ahol a beágyazott Linux egyre fontosabb tényezővé válik. Tavaly tavasszal például mind a Wind River, mind a QNX alapítótagként csatlakoztak az Embedded Linux Consortiumhoz.

Emellett *Jerry Fiddler*, a Wind River alapítója és elnöke mostanában beszédei alkalmával jelentős időt szán arra, hogy cégének a beágyazott Linuxhoz viszonyított helyzetét bemutassa. A legutóbbi ESC nyílt forráskódról szóló vitájában *John Fogelin*, a Wind River technológiai alelnöke így nyilatkozott: – „Úgy véljük, hogy az értékesítési eszkö-



zők, az ATM-ek, az ipari PC-k és az interneteszközök azok az alkalmazási területek, ahol a Linux helyettesítheti a DOS-t vagy a Windows NT-t. Támogatjuk a nyílt forráskódot, és mint választható operációs rendszert kínáljuk a Linuxot a Wind River vásárlóinak. Jelenleg is fejlesztés alatt állnak Linux-alapú megoldásaink.” A legutóbbi nyílt forráskód körüli vitában az ESC-n Fogelin megerősítette a Wind River nyílt forráskódra vonatkozó támogatását.

Ezekben a napokban a Microsoft Embedded & Appliance Platforms Group is megpróbálja bevezetni a nyitottság és rugalmasság egy új formáját. Bár közvetlen utalás nem történt a Linuxra, az olyan kifejezések, mint a „forráskód elérhetősége”, az „egyszerűsített szerződések”, a „rugalmas üzleti modell” és a „windowsos beágyazott fejlesztői közösség” is megjelentek a Microsoft szótárában.

Közvetlenül ugyan nem támogatja a Linuxot, a QNX Software Systems fontos kezdeményezést jelentett be júniusban. Ennek célja a POSIX-szabványnak megfelelő valós idejű operációs rendszer, a QNX „Linux-szerűként” való megjelenítése. A változások közt volt számos QNX modul forráskódjának nyíltá tétele – bár a QNX Neutrino rendszermagja továbbra is a cég titka és tulajdona marad –, illetve a QNX fejlesztői készlet ingyenes elérhetősége a fejlesztők és magánszemélyek számára.

A get.qnx.com nagyjából a tanácskozással egy időben indult.

A záráskor a QNX arról számolhatott be, hogy több mint száz ezer fejlesztő töltötte le az ingyenes QNX fejlesztői készletet. A leggyökeresebb változás azonban a Lynx Real-Time Systems részéről jelent meg, mely ősszel hozzálátott ahhoz a munkához, amellyel saját, a POSIX-szabványoknak szintén megfelelő valós idejű LynxOS operációs rendszere mellett a Linuxot is beépíti termékínálatába. Fél évvel később a cég megtette a következő lépését, mely még inkább tükrözte a két operációs rendszerre épülő elképzeléseit: LynxWorksre változtatta a nevét.

A siker mértéke

Nem meglepő, hogy az EE Times beágyazott piacra vonatkozó ideai elemzése hatalmas bővülést jelez a Linux fejlesztői felületként való használatában, ez 1400 százalékos növekedést jelent az elmúlt 12 hónap során.

Sajnos, a felmérés semmilyen hasznos adattal nem szolgál a Linux – vagy akár a Windows – beágyazott célfelületként való jelenlegi vagy tervezett használatára vonatkozóan, mivel a jelek szerint megengedhetetlen hibát követtek el a felmérés egyetlen erre vonatkozó kérdésében. Az egyetlen kérdés a célfelületet illetően úgy volt megfogalmazva, hogy a válaszok között csak valós idejű operá-

ciós rendszereket lehetett megadni. A felmérés kérdése így szólt: „A következő kereskedelmi operációs rendszerek melyikét (A) használta korábban, (B) használja jelenleg (C) tervezi használni a következő évben?” Lehetséges választásként pedig csak valós idejű rendszerek, mint a VxWorks, a QNX, a LynxOS, az OS9 stb. voltak megadva. Nem meglepő, hogy sem a Linux, sem a Windows nem jelentek meg értékelhető módon az eredményben.

A semmiből szűk hat hónap alatt

A Linux a beágyazott piacból kiharított részének egy másik beszédes mértéke az, hogy az Embedded Linux Consortium (ELC) pályája üstökösként ível felfelé, jelenleg 114 tagot számlál. 2000. márciusában 22 cég alapította, az ELC első általános taggyűlését ebben az évben tartotta az ESC-n, ahol a tagok találkozhattak az újonnan megválasztott vezetőséggel és egymással. A találkozón legalább 75 tag és más vendégek érdeklődésére számítottak.

Az ELC elnöke, *Inder Singh* megtartotta rövid beszédét, és szólt néhány szót a beágyazott linuxos piac jelenlegi állapotáról, illetve *John Cheuck*, az ELC elnökhelyettese vázolta az EMBLIX (japán beágyazott Linux konzorcium) céljait és küldetését, azután az ELC megkezdte a tényleges tervek és tevékenységek kiválasztását. A folyamatok elősegítésére hamarosan egy kérdőív kerül az ELC tagjaihoz, melyen kifejtethetik, szerintük mit kellene vállalnia az ELC-nek, illetve melyek azok a területek, melyektől véleményük szerint az ELC maradjon távol.

Jelenleg teljes egyetértés mutatkozik abban a kérdésben, hogy az ELC ne legyen szabványügyi szervezet, azonban annál nagyobb igény van arra, hogy a beágyazott Linux határozottan megjelenjen, mint márkanév. Ezzel kapcsolatosan az ELC elfogadott egy állandó logót – az elmaradhatatlan pingvinnel –, melynek népszerűsítését a weboldalakon és egyéb anyagokon a csoport minden tagjától határozottan elvárja.

Az ELC munkacsoportokat fog alakítani, ezek irányelveket és ajánlásokat dolgoznak ki, melyeket a különböző szabványügyi szervezeteknek lehet eljuttatni. A további ötletek még elbírálás alatt állnak.

Új felületek

Érezhető egy erőteljes törekvés, mely a beágyazott PC-s felépítéssel ellentétes irányba vezet. Ez nem túl meglepő jelenség, ha figyelembe vesszük, hogy a beágyazott PC – mindent összevetve – több mint 15 éves.

Az Intel termékbejelentései nagyjából egyenlően súlyozódtak a legújabb beágyazott Pentium processzorok és az új, a StrongARM utódaként fejlesztett XScale mikrorendszer között. További X86 processzorgyártók is megjelentek az ESC-n, így az STMicroelectronics és a ZF Linux Devices X86-alapú lapkagép-processzorokkal, illetve az AMD a legutóbbi, K sorozatú, beágyazott processzorral. Meglepetésként a National Semiconductor, a rendkívül népszerű Geode X86 lapkagép-processzorok gyártója azonban nem jelent meg a tanácskozáson.

Az ARM, StrongARM, MIPS, PowerPC és egyéb nem X86-felépítések megfelelően bemutatkoztak mind gyártóik pavilonjaiban, mind az eszközök, operációs rendszerek és alap-



1. kép Fantazein óra



2. kép Fantazein óra hátulról



lappgyártók oldalán. A Zilog az ELC egyik szervezeti alapító tagja és a jelek szerint újra életre kel. Így a nagyra becsült Z80 márkanév új jelentést nyer korszerűbb, kapcsolatköz-pontú processzorokkal. Ki tudja... talán a következő ESC-n a Zilog Z80-ra átültetett Linuxszal fog megjelenni! Az alaplapok továbbra is a hagyományos sínrendszereket tartalmazzák. Biztosat jóslolni egyelőre nem lehet, de egy erősödő irányzat mutatkozik, mely a PowerPC, az ARM és

Az sRTAI (vagy RTLinux) viszont könnyedén megteszi ezt. Nagyszerű bemutató arról, mit is jelent a szigorú és enyhe (vagy egyáltalán nem) valós idejű teljesítmény! Azzal együtt, hogy rengeteget beszélünk a nyílt forráskódú operációs rendszerekről és a velük kapcsolatos programokról, vajon mennyit lehet hallani nyílt forrású BIOS-okról? Eleget, ugyanis folyamatban van néhány projekt. Sajnos jelenleg még nincs valóban megfelelően támogatott erő-



a MIPS processzorokra épülő, sínrendszert nem tartalmazó alaplapok fejlesztését irányozza. Erre példák az ADS Bitsy (StrongARM), az Embedded Planet RPX (PowerPC), az Intrinsic CerfBoard (StrongARM) és még mások. Az Ampro, a beágyazott PC-s alaplapok egyik első és kiemelkedő gyártója is bejelentette, hogy szándékában áll kihozni új, EnCore PCI-alapú felületének MIPS változatát.

Mi a menő?

A fenti irányzatok és megfigyelések mellett további két Linuxszal kapcsolatos dolog ragadta meg a képzeletemet. A Lineo valós idejű óra bemutatója. Igen, igen, az órák általában valós idejűek, de ez teljesen más! Egy kicsit nehéz leírni, ezért elnézést kérek. A Fantazein olyan órát készített, amely egy balra-jobbra ingadozó pálcán található LED-sor segítségével jeleníti meg az időt (lásd az 1. és 2. képet). A LED-ek vezérlése annyira finom, hogy a pontos idő szinte a semmiben jelenik meg, ahogy a pálcá mozog. Rendben, feladom, akit érdekel, olvasson utána a cég honlapján ➔ <http://www.fantazein.com/how.html>.

Most, hogy megvan a kép, nekifutok még egyszer és megpróbálom elmagyarázni, hogy mit is alkotott a Lineo. Úgy alakították ki a LED-ek vezérlését, hogy azokat egy PC párhuzamos nyomtatókapuja vezérli. Egy amúgy mást nem végző Linux-rendszerrel az „órán” megjelenő szavak szilárdnak és olvashatónak tűnnek. Ezt követően a rendszer nekilát valamilyen fájlátvitelnek, a kijelző pedig teljesen olvashatatlanná lesz. Most a LED-ek vezérlését tegyük át RTAI-ra, egy szigorúan valós idejű Linux-rendszerre. Igen, kitaláltad, még fájlátvitel közben is sziklaszilárd a kijelző képe. Ebből látszik, hogy egy foltoztatott rendszermaggal felvértezett Linux sem képes a kijelző képét szilárdan tartani terhelés mellett.

fejlesztés a rendszerek felélesztését és indítását végző, átfogó módszerek megteremtésére. Minden jel arra utal, hogy ez az állapot megváltozik a Red Hat által fejlesztett, RedBoot névre hallgató, új programnak köszönhetően. A tervezet új, és még sok mindent meg kell oldani, de a RedBoot hamarosan elindulhat, mint saját fejlesztésű BIOS a számos beágyazott Linux-rendszer között. Ha akart volna beágyazott PC-t nem PC-s jellegű alkalmazásban használni, valószínűleg olyan nehézségekkel kellett megküzdenie, mint a szerzői jogi üzenetek hosszú listájának eltüntetése, testre szabott indítóképernyők létrehozása, az indítási folyamat felgyorsítása, az egyéni alkatrészek felélesztésének támogatása, vagy megbízható ellenőrző szolgáltatások készítése. A RedBoot lehet a megmentő.

Lássuk a jövőt!

Nos, az ESC kétségtelenül hatalmas rendezvény, és vitathatatlan, hogy voltak olyan figyelemre méltó irányzatok és érdekes bemutatók vagy termékek, melyeket nem láttam, és melyekről itt nem szóltam. Találkozunk a következő ESC-n? Lehet, hogy figyelembe véve a Linux terjedésének sebességét a rendezvény új nevet kap: ELSC – Embedded Linux Systems Conference.

Rick Lehrbaum (rick@linuxdevices.com)

hívta életre a ➔ <http://www.LinuxDevices.com/> honlapot, a „beágyazott linuxos honlapot”, mely nemrég a ZDNet Linux Resource Center része lett. Rick 1979 óta dolgozik a beágyazott rendszerek területén. Az Ampro Computers társalapítója, a PC/104 Consortium alapítója, illetve közreműködött az Embedded Linux Consortium indításában is.

A 2000. évi olvasói díjak

Kedves Olvasóink, amerikai társlapunkban minden évben szavazásra serkentik a szerkesztők a linuxosok egyre népe-sebb tábort. Ennek keretein belül megkérdezik az olvasók véleményét kedvelt Linux-változataikról, szolgáltatásokról, a Linuxot futtató gépekről és alkatrészekről, a szakirodalom-ról, a lap rovatairól, a legnépszerűbb linuxos honlapokról, valamint a böngészőkről. Sőt, az olyan egyszerű hétköznapi dolgok felől is érdeklődnek, mint hogy mi az olvasók ked-venc számítógépes bájitala. Természetesen kimaradhatatlan a legizgalmasabb tárgykör: melyik a legjobb, legnépszerűbb linuxos játék. Ennek az írásnak a közzétételével az a nem titkolt szándékunk, hogy mi magunk is kedvet csináljunk honi olvasótáborunknak hasonló szavazáshoz, hiszen a mi honlapunkon (☞ www.linuxvilag.hu) hamarosan üzembe állítjuk a szavazórendszert. Nézzük tehát, hogy milyen eredménnyel zárult a Linux Journal idei szavazása. A kö-vetkezőkben Heather Mead összefoglalóját olvashatjuk. Terítsük le a vörös szőnyeget, itt az idő, hogy bejelentsük a Linux Journal 2000. évi olvasói szavazásának (Readers' Choice Awards) eredményét. Mozgalmas évet tudhatnak maguk mögött a kedvenc operációs rendszerünket övező nagy felhajtás mindenki számára nyilvánvalóvá tette (mi biz-tosak voltunk ebben), hogy a Linux ma már minden terüle-ten az élmezőnyben jár. Sokan arra számítottak, hogy a Linuxszal kapcsolatos fejlesztési láz hamarosan csillapodik, ennek ellenére azt tapasztaltuk, hogy a „forradalom” egyre gyorsabban és erősebben érezteti hatását. Miközben kíván-csian várjuk az újabb és újabb alkalmazások, eszközök és szolgáltatások megjelenését, e díjjal kifejezhetjük elismeré-sünket a már megszületettek alkotói számára. A tárgykörök száma alapján (eddig ez volt a leghosszabb lista) pedig el-mondhatjuk, hogy már most is számos javaslatunk van, a vá-laszolók rengeteg egyéni jelöléséről nem is beszélve. Hogy még szórakoztatóbb legyen a dolog, minden tárgykörmél feltüntettem néhány tanulságos vagy humoros „jelölést” is. A 24 témában összesen több mint négyezer olvasónk adott le szavazatot, a kedvenc programnyelvtől a legked-vesebb játékig. Vajon milyen tanulságot szűrhetünk le az idén beérkezett szavazatokból? A szenvedélyes hangvé-telű, illetve a koffeinfüggő szavazók száma szinte megegye-zik egymással. Véletlen egybeesés? Nem hinném...

Kedvenc Linux-változat: Red Hat Linux

„Mindegyiket imádom.”

A Red Hat visszanyerte 1997-ben elért első helyét. A má-sodik helyezett SuSE a Red Hat szavazatainak kevesebb mint felét kapta. A Mandrake megkértszerezte tavalyi sza-vazatainak számát és 14 százalékkal áll a harmadik he-lyen. A tavalyi győztes, a Debian leszorult a dobogóról, a Slackware pedig a válaszadók 8,5 százalékának nyerte el tetszését. A legnépszerűbb egyéni jelölés a FreeBSD és a „saját magam által írt változatot használók” volt.

A legkedveltebb irodai programcsomag: StarOffice

„A StarOffice volt, de mostanában inkább a Koffice az.”

A szavazatok 63 százalékának elnyerésével magasan a StarOffice vezet a népszerűségi listán, a második helye-

zett WordPerfect mindössze 12 százalékot gyűjtött össze. Az irodai programcsomagok kérdése talán a legkénye-sebb téma a linuxosok körében, erre az egyéni jelölések-ből is következtethetünk. Jó néhányan röstellkedve kérnek elnézést azért, hogy a Microsoft Office-ra szavaztak, és legalább ugyanennyien írták, hogy „egyik sem, mindegy csak a baj van.” Az Emacs, a vi és változatai, illetve a GNOME Office Suite is számos szavazatot kapott.

A szavazók szívéhez legközelebb álló felhasználói környezet: KDE



„A parancssor.”

Bár sokan a KDE halálát jósolták a GNOME Foundation augusztusi bejelentései után, mégis már harmadik éve vezet a versenyt, legalábbis lapunk olvasóinak szavazatai alapján. A Gnome mindössze 400 szavazattal maradt le. A Window Maker és az Enlightenment közti különbség en-nek csupán a tizede volt, azonban egyikük sem ért el 9,7 százaléknál többet.

Kedvenc szövegszerkesztő: StarOffice

„A WP8, de nem az a windowsos vacak.”

A StarOffice ebben a tárgykörben is az élre tört, a máso-dik helyen végzett WordPerfect szavazatainak közel két-szeresével. A többi szavazat azonban sok, csekélyebb nép-szerűségű program között oszlott meg, ez is azt jelzi, hogy a szövegszerkesztők terén mindenki ragaszkodik saját, rég-óta bevált kedvencéhez.

Kedvenc karakteres szerkesztő: vi és változatai

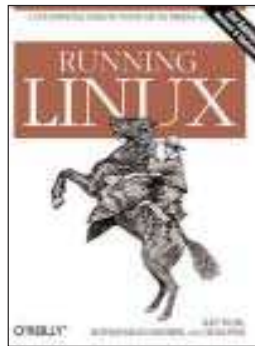
„Az Elvis – annyira jó, hogy külön tárgykört érdemelne!”

Itt aztán még erősebben érezhető a fentebb is említett ra-gaszkodás. A vi és sokféle változata 40 százalékot kapott, s ezzel első helyezést ért el. Az Emacs változatai (a GNU Emacs, az X Emacs és a LaTeX-Emacs összeállítás) is igen népszerűek. Néhányan javasolták, hogy a vi változatai (Elvis, VIM stb.) külön tárgykörbe kerüljenek. Talán jövőre.

Kedvenc linuxos könyv: Running Linux

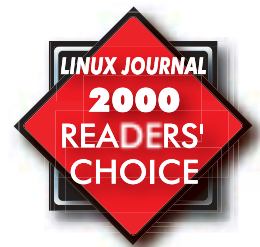
„Miféle könyvekről beszéltek?”

„Az egész O'Reilly sorozat – az életem sokkal nehezebb



volna nélkülük!”

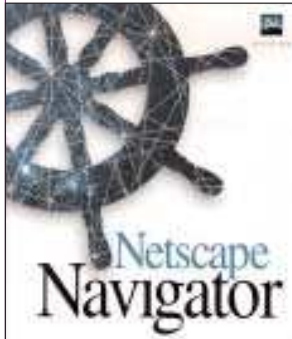
A *Matt Welsh* által írt *Running Linux* az első, az *Ellen Siever* által szerkesz-tett *Linux in a Nutshell* pe-dig a második helyezést érte el, azaz a tavalyi első kettő helyet cserélt egymással. A különbség azonban igen csekély, csupán 54 szavazat volt. A futottak még listáját (ez volt az összes tárgykör között a leghosszabb) olvasgatva nyilvánvaló, hogy számos linuxos kiadvány a szívünkhöz nőtt már. De nem mindenki olvas könyveket: sokan a súgóoldalakat (man) és az internetes leírásokat jelölték meg kedvenc olvasmányukként.





A legkedveltebb webböngésző: Netscape/Mozilla

„Nincs kedvencem; a böngésző inkább csak olyan szükséges dolog.”



„A Zen.”

Az egyéni jelölések alapján ez a tárgykör az amerikai polgárok novemberi választására emlékeztet: azt kell eldönteni, hogy melyik a kisebbik rossz... A Netscape/Mozilla hatalmas fölényrel (80%) nyert, a második helyezett Lynx 201 szavazatot

kapott. A Konquerornak 77 jelölés jutott, a többi szavazat pedig a „Mozilla, és nem a Netscape!”, a „mindegyiket utálom”, illetve az „Internet Explorer – bocs!” válaszok között oszlott meg.

A Linux Journal olvasóinak legtöbb szavazatát elnyert rovata: Kernel Korner

„Az a legjobb, hogy minden számban új rovat indul.”
„Minden túlzás nélkül állíthatom, hogy bármi, amit csináltak, nagyon tetszik.”

Hmmm... Ennyi dicsérettől egészen zavarba jöttünk... Tudjuk, hogy nem minden szám nyeri el maradéktalanul az olvasók tetszését (az egyik borító miatt például különösen sokan zúgolódtak), de úgy tűnik, hogy mindenki elégedett. A Kernel Korner (Szaktekintély) már negyedik éve bajnok, az At the Forge (Kovácsműhely) és a Best of Technical Support (A hónap szakmai tanácsai) pedig a második, illetve harmadik helyet érte el.

A leggyakrabban használt géptípus: Intel x86

„Az Amiga, RH 5-tel és NetBSD 4.x-szel.”

Bár a legtöbben azt állítják, hogy kényszerűségből használják, a szavazást az Intel x86 nyerte 60 százalékkal. Tavaly gyakran előfordult egyéni jelölésként, idén pedig bekerült a lista az AMD, és a szavazatok 20 százalékát meg is szerezte. A harmadik és a negyedik a sorban a PowerPC és az Alpha. A Transmeta idén először kapott egyéni jelölést, jövőre talán még többen választják.

A legtöbb szavazatot elnyert linuxos honlap: Slashdot.org



„Az eLinux.com.”

„Ez valami vicc? Természetesen az összes!”

Bár a Slashdot nem csak a Linuxszal foglalkozik, ennek ellenére úgy tűnik, ez a mi kedvenc helyünk. A három éve veretlen Slashdot a második helyezett Freshmeat.net szavazatainak kétszeresét kapta. A többi népszerű linuxos honlap közé tartozik a LinuxToday.com, a Linux Documentation Project és a Linux.org. Ebben a tárgykörben szintén sokféle egyéni jelölés érkezett, ezek között leggyakrabban a linuxfr.org szerepelt. Tu parles français, non?

Kedvenc üzenetváltó rendszer: Xchat

„Mi más, ha nem a Gnomeicu?”

Ebben az új tárgykörben a válaszadók közel fele adta le szavazatát. Az Xchat 20 százalékos eredménnyel első lett, és a Jabber is csak 15 szavazattal maradt le tőle. A BitchX összesen 14 szavazatával a harmadik helyre került. A Gnomeicu volt a leggyakoribb egyéni jelölés. Jövőre talán már mindenki megtalálja a kedvencét?

Legkedveltebb programozási nyelv: C/C++

„A sima C (a ++ nélküli).”

A régi harcos a C/C++ a szavazatok 40 százalékával büszkélkedhet. Sokan felhívták a figyelmünket arra, hogy a C és a C++ még csak véletlenül sem ugyanaz. A második és harmadik helyezett a Perl és a Java lett, a Python pedig tovább növelte rajongótáborát, idén már nyolc százalékot kapott.

A legnépszerűbb héj: Bash

„A kagylóhéj.”

Már megint a Bash nyert? És még csak nem is volt szoros a verseny? Bizony, bizony. A Bourne Again Shell három éve a legnépszerűbb, idén 78 százalékot tudhat magáénak. A messzi távolban felsejlik a tcsh 10 és a ksh 4,5 százalékos eredménye. Néhányan megjegyezték, hogy a bash-t csak parancsfájlok írására használják, a mindennapos feladatok elvégzéséhez inkább valami mást. A kagylóhéj, juhéj stb. típusú értelmességeket persze sokan nem bírták kihagyni... Ki mondta, hogy a számítógépeseknek nincs érzékük a (rossz) viccekhez?

A legkedveltebb linuxos játék: Quake 3

„Nincs időm játszani.”

„Az operációs rendszer telepítése.”

1998-ban a Quake, 1999-ben a Quake 2, idén meg a Quake 3 nyert... Az X-Bill 9, a Civilization: Call to Power 6 százalékot szerzett. Az első öt helyezett közé két ingyenes játék, a NetHack és a FreeCiv is bekerült. Az egyéni jelölések sokszínűségéből arra következtethetünk, hogy a linuxosoknak is egyre több okuk van arra, hogy ne lépjenek ki a házból.



Kedvenc zeneprogram: XMMS

„A jó kis zsebrádióm.”

„Nem lenne rossz, ha egy is működne közülük!”

Lássunk csodát, a helyzet egy év alatt is milyen sokat változhat. A tavalyi szavazáson az XMMS néhány egyéni jelölésben tűnt fel, az idén elért ötven százalékkal pedig a népszerűségi lista tetejére kúszott. Az összes tárgykör között itt volt a legszorosabb a verseny a második és a harmadik helyezett között: a Real Audio egyetlen szavazatnak köszönhetően tolta maga mögé az mpg123-at. Néhányan büszkén vallották, hogy köszönik szépen, de vígan elvannak a CD-lejátszójukkal.



A leggyakrabban választott adatbázis: MySQL

„Az, amit nem kell felügyelnem.”

„Az FBI amerikaellenes tevékenységet rögzítő adatbázisa.” Az adatbázis-felhasználók hamar lázba jönnek, ha a MySQL–PostgreSQL háború a téma. Olvasóink egyértelműen döntöttek: a MySQL közel kétszer annyi szavazattal büszkélkedhet, mint begépélhetetlen nevű rokona. Az Oracle 8i R2 11 százaléknyi szavazattal a harmadik helyre került, tehát a dobogós helyezések sorrendje nem változott a tavalyihoz képest. A többi adatbázis egyike sem érte el három százalékot.

Kedvenc hálózateszköz-gyártó testület:



Cyclades

„Izé... mi van?”

Nos, úgy tűnik, hogy nagyon sokan azt sem tudták, mit kérdezzünk. Ez legalábbis megmagyarázza, hogy a szavazók több mint

fele miért nem választott. Akik kitalálták, hogy mit akarunk, azok felerészben a Cycladesre szavaztak. A második és harmadik helyezett a Digi International és a Boca lett.

Kedvelt fejlesztőeszköz: Gnu Compiler Collection (GCC)

„A VIM élt, él és élni fog!”

Idén e tárgykör is azok közé tartozott, melyeknél az első és a második helyezett között óriási volt a különbség. A GCC 71 százaléka tízszerese volt a második helyezett Code Warrior szavazatainak. És valóban: mit is érnének a GCC nélkül? A KDevelop pusztán az egyéni jelöléseknek köszönheti harmadik helyét. Ez figyelemre méltó, szóval jövőre a listán a helye!

A legkedveltebb biztonsági mentést végző program: tar

„Biztonsági mentés?!? Minek az?”

Jó, jó. Bár sokan írták, hogy „egy igazi férfinak nincs szüksége biztonsági mentésre”, meg hasonló, azért úgy tűnik, hogy óvatosabb olvasóink szívesen használják a tart. Ez a megoldás elég gyakran szerepelt a tavalyi egyéni jelölések között, így idén bekerült a listába és nyert. A tar az előző év győztesét, a BRU-t 46 százalékkal előzte meg, ám a BRU még így is a második lett. Az Amanda és az Arkeia között szoros volt a verseny a harmadik helyért, melyet végül is az utóbbi nyert el, 11 szavazat különbséggel.

A leggyakrabban használt X-kiszolgáló: XFree86

„Csak XFree86-om van. Lehet, hogy a többi is ér valamit, nemtom’.”

Van egyáltalán valaki, aki otthon nem XFree86-ot használ? Nem valószínű. A válaszadók 93 százalékának véleménye alapján messze a legnépszerűbb a tárgykörben. A második helyen befutó Accelerated-X kemény 3,5 százalékot mondhat magáénak. A legtöbb egyéni jelölést pedig az Xpmc kapta.

A számítógépnél ülve legtöbbet fogyasztott életmentő ital: kávé

„Sajnos, egyelőre a dobozos kapitalizmus (Coke).”

„A cseresznyés Fun Dippel kevert Sprite.”

„Mountain Dew! Ez nem csak egy újabb üdítőital.”

A legdögösebb megjegyzéseket természetesen ehhez a tárgykörhöz kaptuk. Unalmas vagy nem, ennek ellenére szeretjük a kávé, főleg cukorral. A kávé a szavazók felének ki „számítógépes” itala. A többi alkoholfmentes ital szerepel a második helyen, bár sok válaszadó határozottan sértve érezte magát amiatt, hogy a Mountain Dew nem szerepelt külön tárgykörként. Meglepő, de a harmadik helyre a víz került.



Kedvenc fájlmegosztási szolgáltatás: Gnapster

„Mi a fene van?”

A Napster jogi botrányai mindenki figyelmét felkeltették a szolgáltatás iránt (ha már Courtney Love is képes összefüggően és érthetően beszélni, akkor tényleg nagy dologról lehet szó). Internetes szavazásunkban a Gnapster 45, a Gnutella 34 százalékot ért el.

Kedvenc reklámszűrő: Junkbuster

„Nem használok ilyesmit. Köszí a tippet.”

A tárgykörre leadott csekély számú szavazat alapján nyilvánvaló, hogy a legtöbben még mindig dühösen bámulják a szép lassan letűnő reklámcsíkokat. A szavazást a Junkbuster nyerte, negyven százalékot ráverve a második helyezett SquidGuardra, a legtöbb egyéni jelölést pedig az AdZapper kapta. Néhány külön programozó saját maga ír ügynököt a reklámok kiszűrésére. Valljuk be, ez azért nem semmi.

A legkedveltebb grafikai alkalmazás: GIMP

„A barátnőm szerint a GIMP, mert ő azt használja (és most itt bámul a vállam fölött).”

„A Photoshop... hüpp-hüpp... miért nem írja már át valaki Linuxra?”



Továbbra is a GIMP vezet toronymagasan, idén az összes szavazat 72 százalékát birtokolhatja; az xv 10, a CorelDRAW hét százalékkal kullog utána. Különösen a szerényebb rajztudású szavazók értékelik nagyra a GIMP rugalmasságát és egyszerű kezelését.

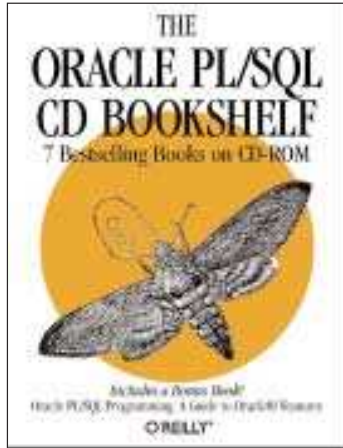
A Readers' Choice Awardsról és számos linuxos szolgáltatásról, illetve termékről sok érdekességet olvashatunk a <http://www.linuxresources.com/> címen.

Heather Mead amerikai irodalmat ment tanulni Seattle-be, de rájött, hogy a munka élvezetesebb, mint a tanulás. Hamarosan a Linux Journalhoz került szerkesztőként. Mostanában éppen egy tévéjátékot ír, de hát ki nem?

Új termékek

Oracle PL/SQL CD Bookshelf

Ez a CD-ROM a következő hét könyv teljes szövegét tartalmazza: Oracle PL/SQL Programming, Advanced



PL/SQL Programming, Oracle Web Applications, Oracle PL/SQL Language Pocket Reference, Oracle Built-in Pocket Reference és Oracle PL/SQL Programming: A Guide to Oracle 8i Features. Ez utóbbi könyv nyomtatott változata is megtalálható a csomagban, melyet készítői főként Oracle PL/SQL fejlesztők kezébe szánnak. A böngésző segítségével olvasható könyvek szövegében természetesen kereshetünk is, a tájékozódást pedig rengeteg kereszthivatkozás könnyíti meg. Adatok: O'Reilly & Associates, Inc., 101 Morris St., Sebastopol, CA 95472, telefon: 800-998-9938, fax: 707-829-0104, e-mail: order@oreilly.com, <http://www.oreilly.com/catalog/oraclecdbs/>

Sculptor

A Chilliware cég Sculptor programja az első linuxos kiadványszerkesztő. Kezelése jóval könnyebben elsajátítható, mint a Windowsra készült hasonló programok, a leírhatatlanul sok, a kiadványszerkesztők jelenlegi igényei szerint kialakított lehetőséggel pedig gyerekjáték a hirdetési táblák, falragaszok, szórólapok, névjegykártyák stb. készítése. Adatok: Chilliware, Inc., 3550 Wilshire Blvd., 18. Floor, Los Angeles, CA 90010, telefon: 213-365-8700, fax: 213-365-1150, e-mail: info@chilliware.net, <http://www.chilliware.net/>

HA Web Server Solution Slam v2.2.6

A Stabie-Soft Slam v2.2.6 nevű programjával nyomtatott áramköröket tervezhetünk. A legújabb változatban az áramkör böngészésére, szerkesztésére szolgáló programokban a DRC és LVS lehetőségeket is kihasználhatjuk. A Slamban a GDSII, Spice, Verilog és SDF felületek közül válogathatunk. A Tcl/Tk-alapú rendszer helyet ad a felhasználói bővítések számára, a beépített Tcl forráskódok pedig a más cégektől származó ellenőrzőprogramokkal való együttműködést is lehetővé teszik. Adatok: Stabie-Soft, 5828 Gentle Breeze Terr., Austin, TX 78731, telefon: 512-825-8914, e-mail: sales@stobie-soft.com, <http://www.stobie-soft.com/>

Atipa Firewall és Firewall Plus

A nagy teljesítményű Atipa Firewall Plus az ICSA által minősített termék, mely a Progressive Systems cég Phoenix Adaptive Firewall Technology (AFT) és Secure Management System (SMS) tulajdonságait ötvözi. Ezenkívül egyéb – mint amilyen a De-Militarized Zone (DMZ) – támadáselhárító képességet is magában foglal. Az új Atipa Firewall és Firewall Plus felületfüggetlen és frissíthető. Mindkét termék Intel x86-alapú processzort, 64 MB memóriát, 10,2 GB merevlemezt, webalapú kezelői felületet, távoli felügyeleti rendszert, DHCP-kiszolgálót, IP-maskarádézást tartalmaz, és egy 1U típusú keretbe illeszkedik. Adatok: Atipa Linux Solutions, 4700 Belleview, Suite 300, Kansas City, MO 64112, telefon: 800-360-4346, 816-595-3001, e-mail: sales@atipa.com, <http://www.atipa.com/>

Storm Linux 2000 Starter Edition, Deluxe Edition és Storm Firewall

A Starter Edition változatában a Linux telepítéséhez és beállításához szükséges eszközök kaptak helyet, ezeknek főleg a kezdő felhasználók örülhetnek: itt találjuk a Powerquest cég Partition Magic Linux Prep Tool nevű programját, a Netscape Navigator 4.73-at, és a Sun StarOffice 5.2-es változatát. A Starter Editionnal a cég azokat a fel-

használókat célozza meg, akik a Linux lehetőségeit más cégektől származó programok nélkül kívánják kihasználni. A Storm Linux 2000 Deluxe Edition segítségével megbízható Linux-rendszert építhetünk ki, s ebben a csomagban külső cégek programjait is megtaláljuk. A Deluxe Edition alapját a Stormix nagy teljesítményű, a Debian GNU/Linux 2.2 Potato-változatára épülő operációs rendszere képezi. A csomag öt CD-jén több mint négyezer Debian-csomag kapott helyet, és vásárlás után még hatvan napig telefonon, kilencven napig elektronikus úton kérhetünk segítséget angol nyelven az ügyfélszolgálattól. A Storm Firewall rugalmas, méretezhető biztonsági megoldás, mely elsősorban a kisebb vállalati, illetve otthoni felhasználók számára készült.



A program a Storm Linux 2000 operációs rendszere épül, de a Red Hat 6.x és a Debian GNU/Linux 2.2-es változatokkal is kifogástalanul működik. Adatok: Stormix Technologies, Inc., 555 West Hastings Street, Suite 2040, Vancouver BC, V6B 4N6, Canada, telefon: 604-688-9137, fax: 604-688-7317, <http://www.stormix.com/>

IRIS-Planning v4.2

A Bitbybit Information Systems jelentette időbeosztás- és helyfoglalás-tervező programjának 4.2-es változatát. A fejlesztések között a fő feladatokért felelős modul bővítése, valamint új topológiai kiépítésű és levélmodulok szerepelnek. A program a Weben keresztül történő helyfoglalásra is képes, az időtáblázatokat pedig HTML-formátumban is elkészíti. Adatok: Bitbybit Information Systems, Kluyverweg 2a, 2629 HT Delft, Hollandia, telefon: +31-15-2682569, fax: +31-15-2682530, e-mail: info@bitybit-is.com, <http://www.IRIS-Planning.com/>



LinuxMagic VPN Firewall

A LinuxMagic VPN Firewall felhasználói a legújabb titkosítási eljárások igénybevételével érhetik el irodai számítógépüket bárhol. A program beépített, 128 bites titkosítással, IPSEC-megfelelő VPN-szabványokkal dolgozik, és a LinuxMagic saját NO-WRITE eljárását alkalmazza.

Adatok: Wizard Internet Services Ltd., 13595 King George Hwy., Surrey, BC, V3T 2V1, Canada, telefon: 604-589-0037, fax: 604-584-0010, e-mail: sales@wizard.ca, <http://www.linuxmagic.com/>, <http://www.wizard.ca/>

ATI 2,5" IDE meghajtó

A BITMICRO Networks forgalomba hozta E-Disk ATI25 típusú, 2,5 hüvelykes IDE flashlemezét, mely 4,5 MB, burst-üzem módban pedig 5 MB adat olvasására vagy írására képes másodpercenként. Több modell is létezik, méretük 128 MB-tól 4 GB-ig terjed. Az ATI25 -40 és +85 C°-os hőmérsékleti értékek között működőképes, és 1000 G-s ütést, vagy 16,5 G-s rázkódást is elvisel. Az új ATI25 bármilyen hagyományos 2,5 hüvelykes merevlemez vagy flashlemez helyettesíthető.

Adatok: BITMICRO Networks, Inc., 45550 Northport Loop East, Fremont, CA 94538-6481, telefon: 510-623-2341, fax: 510-623-2342, e-mail: info@bitmicro.hu, <http://www.bitmicro.com/>

EtherLite for Linux

Megjelentek a Digi International cég EtherLite típusú soros hálózati elosztóinak linuxos meghajtói. Az EtherLite az RS-232, RS-422 és RS-485 csatlakozók hálózatba építését könnyítik meg azzal, hogy a helyi csatlakozók vezérlését és teljesítményét az Ethernet kapcsolatok rugalmasságával elegyítik. A 2, 8, 16 vagy 32 EtherLite kaput egyetlen TCP/IP folyamat irányítja; a kapuk Unix alatt helyi TTY-kként, Windows NT alatt pedig COM kapuként jelennek meg. Ezen helyi vezérlésű, gépszintű kapukat hatékonyabban irányíthatjuk, mint hagyományos társaikat. Az EtherLite Linux meghajtója 2.2-es, vagy magasabb válto-

zatszámú rendszermaggal működik, és a következő Linux-változatok is tartalmazzák: Red Hat 6.0, Caldera OpenLinux és SuSE Linux 6.3.

Adatok: Digi International, Inc., 11001 Bren Road East, Minnetonka, MN 55343, telefon: 800-344-4273, fax: 612-912-4952, e-mail: info@digicom.com, <http://www.digi.com/>

Apache számára

A SteelEye High Availability (HA) egy magas rendelkezésre állást biztosító webkiszolgáló-bővítés az Apache számára, mely a SteelEye cég LifeKeeper rendszerét is magában foglalja.

A csomaggal az Internetszolgáltatók és az IT szakemberek magas rendelkezésre állású, az e-üzleti rendszerekhez is kapcsolódó webkiszolgálókat építhetnek föl. A lenyűgöző jellemzők közül csak néhányat emelnénk ki: aktív/aktív beállítás, központosított, Java-alapú felhasználói felület, a beállításokat helyi és hálózati lemezekre is tárolhatjuk, az adatok összefüggőségét többirányú adatismétlő alrendszer adja, többszintű hibaellenőrző szolgáltatások, megosztott lemezkezelés és esetleges hiba esetén is folyamatos működés.

A HA Web Server Solution for Apache nevű csomagban találjuk a LifeKeeper 3.0 Linux változatát és az Apache Web Server Application Recovery Kit nevű hibakezelő rendszert. A cég egy évig éjjel-nappal elérhető angol nyelvű ügyfélszolgálattal segíti a vásárlókat számára. A csomag a Red Hat 6.1 és a Caldera eServer 2.3 változatokkal használható.

Adatok: SteelEye Technology Inc., <http://www.steeleye.com/>

Immunix Workgroup Server Appliance

A WireX bejelentette legújabb termékét, ennek az Immunix Workgroup Server Appliance (IWSA) a neve. Ez a könnyen kezelhető, olcsó, átfogó programcsomag a web-, a levelező-, a fájl- és a nyomtatókiszolgáló szolgáltatásait egyesíti, és az esetleges behatolóktól is megvédi a rendszert. Az IWSA csomaggal a rendszert Interneten keresztül távolról is felügyelhetjük, az interaktív, menükkel vezé-

relhető kezelői felületet pedig a gyakorlott és a kezdő rendszergazdák is hamar megszokják. A legtöbb mai PC-vel együttműködik.



Az IWSA alapját az Immunix OS képezi. Ez egy hagyományos Linux-változat, mely a WireX által kifejlesztett összetett kiszolgálóelemeket tartalmazza.

Adatok: WireX Communication, Inc., 920 SW 3rd Ave., Portland, OR 97204, telefon: 503-222-9660, fax: 503-241-5682, e-mail: info@wirex.com, <http://www.wirex.com/>

Modemes behívó ügyfél

A Perle Systems a 833 Access Server kiszolgálójához megjelentetett egy behívó ügyfélprogramot Linux alá. Az új ügyfél segítségével a linuxos gépek megoszthatják a modemot más Linux, vagy Windows 95/98, NT, valamint 2000 gépekkel. Ezt a program úgy éri el, hogy a Linux-alkalmazások adatfolyamát, mely alapesetben a COM kaphoz tartozik, a kiszolgáló modemgyűjtőjéhez (modem pool) irányítja át. Ezen programcskának köszönhetően a Perle 833 Access Server feleslegessé teszi azt, hogy a behívást igénylő Linux-munkaállomások mindegyikébe modemot helyezünk, és így a modemes behíváshoz kevesebb PBX kapura van szükség. Az új Dial-out Client for Linux elnevezésű alkalmazást a Perle 833 v5.64, a Perle 833IS v6.06 és v7.0, illetve a Perle 833AS v6.15 vagy ennél fejlettebb változatai támogatják. A programot ingyen letölthetjük a Perle honlapjáról.

Adatok: Perle Systems Inc., telefon: 800-337-3753, e-mail: info@perle.com, <http://www.perle.com/>



© Kiskapu Kft. Minden jog fenntartva

LME-hírek

Immár másodszor rendezte meg a Linux-felhasználók Magyarországi Egyesülete (LME) a Linux Szakmai Konferenciát, melynek idén a Mechanikai Szakközépiskola adott helyet. A tanácskozás főbb vonulatai az internetes biztonság és a webes fejlesztés voltak, ezzel is igazolva, hogy a Linux nemcsak jó irányba halad, hanem mutatja is az utat. A valamivel több mint kétszáz látogató elégedetten nyugtázhatta, hogy mind a színhely, mind az előadások színvonala megfelelt a követelményeknek és – ami a legfontosabb – az ő elvárásainak. Különösen kedvelt volt a nagyobb előadások színteréül szolgáló „görög színház”, miként az aulát a látogatók és a szervezők együttesen elnevezték. A második előadóterem inkább családiasabb légkört teremtett, hiszen az előadók szinte a közönség „öleiben ülve” tarthatták előadásait. Azok sem unatkoztak, akik éppen egyik előadáson sem tartózkodtak, ugyanis számos kiállító jött el bemutatni az általa használt vagy szállított linuxos megoldásokat, programokat és szakkönyveket. Ráadásul az LME által forgalmazott ajándéktárgyakra is szert tehetek. Mindezeket túl azonban a Konferencia leglényegesebb eseménye díjra az a bejelentés pályázhatott, melyet a Miniszterelnöki Hivatal képviselője tett: eszerint az LME tízmillió forintos pályázatot nyújthat be a Linux illetve nyílt forráskódú rendszerek elterjesztéséhez szükséges projektek megvalósításának előkészíté-

sére és kivitelezésére. A megnyitó és az ehhez kapcsolódó bejelentés után elkezdődtek a szakmai előadások is. A webes fejlesztést leginkább a PHP-s fejlesztők és az adatbázisok uralták, melyek kivétel nélkül nagy sikert arattak olyannyira, hogy a megjelent cégek képviselői valószínűleg megrohamozták az előadókat. Az adatbázisokhoz kapcsolódóan az egyik előadás egy egész könyvtári rendszer linuxos kiszolgálóra helyezéséről és az ahhoz kapcsolódó terminálokról szólt igen nagy sikert aratva. Ez a megoldás lényegesen olcsóbb és hatékonyabb, mint más felületen megvalósított társai. A biztonsággal foglalkozó előadások páratlanul nagy tömeget vonzottak. Ezenkívül a levelezési rendszerekről, biztonsági haditerv kidolgozásáról és egy magyar fejlesztésű tűzfalról is hangzottak el előadások. A tűzfal külön érdekessége, hogy GPL-szabályozású, tehát mindenki számára szabadon elérhető és ellenőrizhető. Nyugodtan kijelenthetjük, hogy az előadók a jelenlegi magyar informatika felső rétegét képviselték, ezzel is biztosítva a magas színvonalat és bizonyítva a Linux kiemelkedő alkalmazhatóságát. Zárásképpen köszönetet kell mondani azon támogatóknak, akik pénzzel, eszközzel vagy akár termékekkel segítettek a tanácskozás megrendezésében. Jövőre remélhetőleg még többen leszünk, és a jelenlegi látogatókat talán előadókként láthatjuk viszont.

LME Elnökség

A Linux újabb hódítása: Garden Grove

Ahogy mi gondoljuk

„Ejnye, kezdem azt hinni,
hogy a Linux tényleg
a legjobb dolog a szeletelt
kenyér óta!”

– Vance Petree, Virginia Power

Öt évvel ezelőtt a kaliforniai Garden Grove városának informatikai részlege jelentős költségvetési nehézségekkel küszködött. *Charles Kalil*, a város informatikai rendszereinek vezetője a kereskedelmi programokért való további pénzkiadás helyett, a Linux kipróbálása mellett döntött. Letöltötte az egyik ingyenes változatot, telepítette, és megetszett neki.

Idén a Linux már hat kiszolgálón és háromezernyolcvanhat munkaállomáson futott. A teljes hálózatot ez a hat gép szolgálja ki, a közművektől a tűzoltóságig, és nem is lehetnének elégedettebbek az eredménnyel.

„Az egész városban kiépített linuxos rendszerünk több mint egy éve folyamatosan, minden fennakadás nélkül működik” – mondta Kalil.

Tapasztalt NT-s rendszergazdaként úgy gondolja, az NT-vel is el lehet érni hasonló teljesítményt, de ez azt jelenti, hogy minden egyes NT-s gép csak egy feladatot tud ellátni.

„A Linux megbízhatóan tudja kezelni a fájlkiszolgáló és nyomtatási feladatokat, a levelezőrendszert és még további szolgáltatásokat egyetlen gépen” – mondta Kalil. A város eredetileg egy NT-alapú rendszert akart vásárolni, ez azonban több kiszolgáló és egyéb programok megvásárlását is jelentette volna. Ezzel szemben „a Linux ingyenesen letölthető volt, tartalmazta a webkiszolgálót, a levelezőkiszolgálót, a Samba fájl- és nyomtatómegosztást, valamint támogatta a hálózati fájlrendszert is.” – emlékezett vissza Kalil. „Az egyéb rendszerek, mint a SCO vagy az

NT ezeket nem tartalmazták, vagy az elemeket külön kellett volna megvenni”.

Garden Grove vezetőinek sem volt könnyű az első lépéseket megtenni, hiszen 1995-ben a Linuxot használni a támogatás hiánya és a kevés alkalmazás miatt sokkal kockázatosabb volt, mint ma. Ennek ellenére linuxos adatbázist állítottak fel, ami azóta is működik.

A költségek szempontjából a különbség lényeges volt: Kalil kicserélt egy 400 ezer dolláros Data General miniszámítógépet két Pentium 90-es kiszolgálóra, ez pedig mindössze ötezer dolláros kiadást jelentett a város számára.

Bizonyos alkalmazásokat azonban nem tudtak áttenni Linuxra. Kalil azt is hozzátette: „Továbbra is NT-t használunk a képszerkesztő programokhoz és az optikai tárunkhoz.” Így Garden Grove városa fenntart egy NT-t futtató GIS-kiszolgálót is, ugyanis ennek a GIS-rendszernek sincs linuxos változata.

Ha az áron kívül a megbízhatóságot és a rendelkezésre állást is figyelembe vesszük, akkor a város teljesen elhívott a Linux mellett. „Rájöttünk, hogy a Linux rendszermag hatékonyságának köszönhetően nincs szükségünk drága kiszolgálókra, továbbá nagyjából feleannyi memóriával elérhetjük ugyanazt az eredményt, mint az NT-vel” – nyilatkozta Kalil.

Drew Robb Los Angelesben él, szakmai kérdésekre szakosodott szabadúszó.

A hónap szakmai tanácsai

Valami igazán furcsa

A Mandrake 7.1 telepítések a rendszer lemezszeréseit úgy állítottam be, hogy a /dev/hdb4 a /usr, a /dev/hdb2 a / (a főkönyvtár), a /dev/hda1 pedig a /boot lett. A gondom az, hogy a Linux leállításakor vagy újraindításakor a fájlrendszerek leválasztása helyett a „/usr: device is busy” hibaüzenetet kapom. Ezután a rendszer lefagy, újraindítás után pedig nagyon sokáig tart a /dev/hdb4 ellenőrzése. Szeretném tudni, hogy mi okozza a hibát, illetve hogy miként háríthatnám el azt.

Charles Diaz-Alejandro, ogura@bellsouth.net

Az egész úgy hangzik, mintha a futásszintek kiiktatásakor a rendszerben még lennének futó folyamatok. Ha ezek valamelyike éppen az adott lemezrész valamelyik fájlját használja, akkor a mount parancs sikertelen lehet.

Előkerülhet a hiba, ha egyfelhasználós üzemmódban indítod a rendszert, majd megszámolod a folyamatokat és ellenőrzöd, hogy pontosan mely folyamatokról van szó. Az egyes futásszinten nagyon kevés folyamatnak kell lennie, hiszen a futásszintek közti váltáskor ezek többnyire leállnak – lehet, hogy az egyik folyamat nem válaszol, vagy nem lehet szabályosan leállítani. Egyfelhasználós üzemmódban fűzd be újra a /usr lemezszerést a mount -o remount,ro /usr paranccsal. Ha ez eredménytelen (továbbra is „busy” hibaüzeneteket kapsz), akkor kísérelj meg kilőni néhányat a nagyobb sorszámú folyamatok közül (az alacsony sorszámúakat nyilván mindenképpen meg kell tartani). Minden egyes kilövés után próbálkozz a fenti paranccsal egészen addig, amíg egyszer csak sikerül. Így kiderítheted, melyik folyamat okozta a lefagyást. Az is lehet, hogy a gondot az amd folyamat idézi elő. Jómagam már sokszor tapasztaltam, hogy az amd és az nfs okoz ilyeneket. Ha megvan a bűnös, akkor állítsd le a szolgáltatást, és dolgozz ki egy áthidaló megoldást.

Andy Bradford, andyb@calderasystems.com

A rendszer akkor tagadja meg a fájlrendszer leválasztását, ha annak egyik fájlja használatban van, illetve ha egy még futó folyamat az adott fájlrendszer valamelyik könyvtárában dolgozik. Ha az umount eredménytelen, akkor a legtöbb Linux-változat megkísérli csak olvashatóként befűzni a fájlrendszert, hogy a legközelebbi rendszerindításkor ne kelljen végigellenőrizni az adott lemezrészét. A csak olvashatóként történő újrafűzés csak akkor nem sikerül, ha írásra lett megnyitva egy fájl. Ha a mount parancs nem túl régi, cseréld le az umountot a umount -r parancsra (a sűgőoldal szerint „ha az újrafűzés sikertelen, próbáljuk meg csak olvashatóként befűzni a fájlrendszert”).

Nem tudom megítélni, hogy a leválasztás miért volt sikertelen, de a rendszerleállító héjprogramban az umount elé helyezett „fuser” parancs segíthet megtalálni a ludast: fuser -m /usr; ps aux; sleep 10

A „ps” a folyamatok megértésében segít. A további részletekért nézd át a fuser leírását (man3 fuser).

Alessandro Rubini, rubini@linux.it

A galibát valószínűleg a Mandrake egyik kisebb belső hibája okozza. Az ügy tisztázása céljából illeszd a következő sorokat a /etc/rc.d/init.d/halt fájl vége elé:

```
(fuser -vm /usr; ps auxww ) | more
read a
```

Mindezt a „# Now halt or reboot” rész elé helyezd. Ehhez általában a /usr/sbin/fuser-t a /sbin/fuser-be kell másolnunk, de mivel a /usr-t nem választjuk le, ezért onnan is el kellene érünk a parancsot.

A fuser azon folyamatokat listázza ki, melyek még mindig a /usr befűzési pontot próbálják meg elérni, a sorszámuk alapján pedig megtalálható a programot. Ha már megtaláltad a hibát, módosíthatod a leállási parancsfájlt úgy, hogy az kilője a beragadt folyamatokat (bár a leállási parancsfájlnak eleve minden feladatot le kellene állítania), vagy érdeklődj a MandrakeSoftnál, hogy ezek a folyamatok miért nem állnak le, és a lemezszerést miért nem lehet leválasztani.

Marc Merlin, marc_bst@valinux.com

Felfüggesztett üzemmód?

Képes-e a Linux felfüggesztett üzemmódba lépni, a Windows 98-hoz hasonlóan? Így a gépem üresjáratban szépen „aludni küldené magát”, és ha ismét használni szeretném, akkor feléledne.

Ronnie Bell, ronbell@cais.com

Ha a rendszermagba befodítottad az APM támogatást (Advanced Power Management, fejlett energiagazdálkodás, a Red Hat 6.2-ben ez az alapbeállítás), az AT-s ház Standby, vagy hasonló nevű kapcsolóját beállíthatod úgy, hogy annak rövid idejű nyomva tartása felfüggesztett állapotba hozza a rendszert. Parancssorból ezt az apm -s vagy apm -S parancsokkal végezheted el.

Marc Merlin, marc_bts@valinux.com

A rendszermag automatikusan leállítja a processzort, ha éppen nincsen feladata. Ilyenkor a processzor alacsony energiafelvételi állapotba kerül, új feladat érkezésekor pedig azonnal feléled. Mindebből a felhasználó semmit sem vesz észre. A rendszermagban az APM támogatását is bekapcsolhatod – feltéve, hogy laptopodban APM-megfelelő BIOS található.

A további energiatakarékoskodás céljából az X kiszolgálóval azt is közölheted, hogy ha a billentyűzetről vagy az egérről egy meghatározott ideig nem érkezik jel, akkor a DPMS (Display Power Management System, a képernyő energiagazdálkodási rendszere) segítségével kapcsolja ki a monitort. Én ezt parancsokkal oldottam meg:

```
xset +dpms
xset dpms 600 1800 3600
```

A sorok hatására a képernyő 600 másodperc után kikapcsol, 1800 másodperc múlva felfüggesztett állapotba kerül, 3600 másodperc elteltével pedig kikapcsolja magát. Mindhárom szintet érdemes saját munkamenetedhez igazítani, de arra azért figyelj, hogy ezek a



lehetőségek csak DPMS támogatással bíró monitorral és grafikus kártyával működnek – ezek hiányában semmi nem történik. Az xset parancs használata helyett az XF86Config fájl módosításával is bekapcsolhatod a DPMS-t: az XF86Config sűgőoldalain keress rá a „power_saver” szövegre; ott minden szükséges adatot megtalálsz a témával kapcsolatban.

Végül, a hdparm paranccsal a merevlemezt is leállíthatod. Ezt én nem javasolnám, mivel a leállítás és az ismételt felpörgetés körülbelül hatórai használatlaltal egyenértékű terhelésnek teszi ki a merevlemezt. Ha le akarod állítani, megteheted, a részleteket megtalálod a program leírásában (man hdparm). Én továbbra sem használom e lehetőséget – ugyanis igyekszem mindent megtenni a linuxos merevlemezem életéért...

Scott Maxwell, maxwell@ScottMaxwell.org

Használj a Red Hat-változatokban található apmd csomagot. Az „apmd” démont természetesen rendszerindításkor kell betöltenünk (a „setup” vagy a „linuxconf” segítségével illeszhetjük be a rendszerszolgáltatások közé).

Pierre Ficheux, pficheux@com1.fr

Különös rendszerleállítás

Lehetséges az, hogy az egér rendszerleállást okozzon? Ha X üzemmódban vagyok és az egeret használom (eddig a Microsoft Intellimouse és a Logitech First Mouse típusokat próbáltam ki), akkor a rendszer néha minden további nélkül lefagy, még az X-ből sem tudok kilépni. A két egér egyébként hibátlanul működik, mindkettőt kiprobáltam PS/2 és soros csatlakozókkal, valamint általános, illetve saját meghajtóikkal is. A lefagyás csak az egér használatkor következik be, de nem azonnal, hanem 5 és 45 perc közötti időtartam után. Az egész igen rendszertelen, a Red Hat szakemberei is csak a vállukat vonogatták. Én azonban makacsul keresem a megoldást, és mindennemű segítséget nagyon köszönök.

Mike, shirleymg@netscape.net

Azt figyelembe véve, hogy a PS/2-es egér a 12-es megszakítási vonalat használja, és hogy ezt egyetlen magára valamit is adó alaplapnak nem lenne szabad más eszköz-höz rendelnie, valószínű, hogy a baj nem az egér és valamely más eszköz ütközéséből fakad (hacsak nem egy ISA kártyával kell versenyt futnia a megszakításért).

Lehet, hogy az X kiszolgáló, vagy a grafikus kártya a ludas. Frissítsd az X-et, ha ez sem hozna megoldást, akkor a grafikus kártyát kellene lecserélned.

Marc Merlin, marc_bts@valinux.com

Ha a baj csak PS/2-es egérrel fordulna elő, akkor minden bizonnyal a rendszermag meghajtóinak változatszáma lenne a hibás. A PS/2-es egeret a billentyűzetvezérlő kezeli, ha tehát bibi van, akkor egyik eszköz sem használható. Ilyenkor a legújabb rendszermag telepítését javaslom.

Mivel azonban a hiba soros egérrel is előfordul, ezért nem hiszem, hogy az egérrel vagy a meghajtóval van a baj. A „Magic SysRq” szolgáltatással nyomtass ki néhány rend-

szerjellemzőt a lefagyáskor. A SysRq segítségével az X-et is kilőheted – ha ez nem sikerülne, akkor próbáld meg a szöveges üzemmódban előidézni a hibát, majd így kinyerni a szükséges adatokat.

Ha ez sem sikerül, akkor tényleg nagy a baj, és én inkább gép mintsem programhibára gondolok. Lehet, hogy a processzor túlmelegszik és leáll?

Alessandro Rubini, rubini@linux.it

Láthatatlan írás

Ha az X-ben a CTRL-ALT F1–F6 billentyűkombinációt használom, vagy kilépek a wm-ből, akkor egy szép fekete konzolt kapok. A parancsokat minden további nélkül elfogadja, de egyetlen karaktert sem jelenít meg, azaz teljesen vakon kell gépelnem. Újraindítás után minden rendben van, de ha elindítom az X-et, majd kilépek, akkor megint ugyanez történik. Már átírtam a /etc/inittab-ot úgy, hogy az ALT+FEL billentyűkre újrainduljon a rendszer, de ez egy kényszermegoldás és eléggé bosszantó. Tudja valaki, hogy mi okozhatja ezt az egészet, és hogyan javíthatnám ki? Nemrég fordítottam le a 4.01-es X-et és egészen addig jól működött, míg egy-két nappal később egy rendszer-mag-újrarendelés miatt újra kellett indítanom a gépet. Látam már ilyesmit, tehát tudom, hogy nem a 4.01-es X idézi elő a hibát. Valószínűleg én rontok el valamit, legálábbis azt hiszem. Nem SVGAtextmode-ot használok, és a karakterek sincsenek összeczagyválva, csak egyszerűen nem jelennek meg.

Steve Udell, hettar@home.com

Ha a szöveges üzemmód nem 80×25 karakterrel megy, akkor át kell állítanod arra. Néhány grafikus meghajtó a nem alapértelmezett szöveges üzemmódokat nem képes tökéletesen visszaállítani, és így csak a VesaFB marad (ez a pingvines logót tartalmazó grafikus szövegkonzol).

Marc Merlin, marc_bts@valinux.com

A VT100 utánzása

Ha egy linuxos PC-ről egy linuxos kiszolgálóra a telnettel jelentkezem be, majd a távoli gép egyik fájlját a vi segítségével szerkesztem, a képernyő hirtelen megtelik ANSI X3 helyőrzéző karakterekkel. Azt hiszem, hogy a PC-m nem utánozza rendesen a VT100-at. A TERM=vt100 parancsot már beillesztettem a .bash_profile fájlomba, az EXINIT='term=vt100' sort is használtam. Sőt, a vi-nak is megkíséreltem megmagyarázni (:term=vt100), sajnos nem sok sikerrel. Mit rontottam el és hol?

Dominic Wild, wildd@optusnet.com.au

Nekem úgy tűnik, hogy a TERM változó beállításával van a bibi. Mindent jól csináltál, de nem biztos, hogy vt100-at kell használnod; próbálkozz inkább az „ansi”-val. Azt se felejtse el, hogy a terminált a helyi és a távoli gépen egyaránt be kell állítanod. Tehát minden használt terminálnak egyformának kell lennie ahhoz, hogy tökéletesen működjenek. A helyi gép termináltípusának ellenőrzéséhez jelenítsd meg a változók tartalmát az „env” paranccsal.

Felipe E. Barousse Boué, fbarousse@piensa.com



A terminál beállításainak (\$TERM stb.) semmi köze magához a terminálhoz. A változók csak azt közlik a programokkal, hogy milyen típusú terminálon futnak, illetve hogy melyik megszakítóbillejtűt kell használnunk.

Ha tehát a \$TERM változót saját kezűleg vt100-ra állítod, akkor a programok egy olyan terminálnak küldenek vt100 parancsokat, amely nem is biztos, hogy vt100-as típusú. Gyanítom, hogy a terminálok típusa „linux”, „xterm”, „rxvt”, illetve az általad használt terminál nevének kell lennie. Mivel a \$TERM változót minden telnet kapcsolat használja, ezért átírásával nem mindig oldható meg a gond.

Alessandro Rubini, rubini@linux.it

A Num Lock kettős indításkor

A számítógépem indulásakor a Red Hat Linux 6.2 és a Windows 98 között választhatok. Ha a Windows 98-at indítom el, a Num Lock billentyű bekapcsolva marad, viszont Linux alatt kikapcsol. Van valami megoldás arra, hogy ne kelljen a Linux minden egyes indításakor a Num Lock billentyűre tapadnom?

Michael Kaneshige, kaneshige@uswest.net

Ha szöveges módban dolgozol, olvasd el a setleds leírását a man setleds paranccsal. Az egyik indítási parancsfájlból (mondjuk a /etc/rc.d/rc.local-ban) helyezd el az alábbi sorokat:

```
for i in 1 2 3 4 5 6 7 8 ; do
setleds -D +num < tty$i ; done
```

Ez a Linux indításakor bekapcsolja a Num Lock billentyűt. Ha grafikus felületet használsz, abban általában beállítható a Num Lock indításkor érvényes állapotát.

Felipe E. Barousse Boué, fbarousse@piensa.com

Írd a következő parancsokat a /etc/rc.d/rc.local indítási parancsfájlból:

```
INITTY=/dev/tty[1-8]
for tty in $INITTY; setleds -D +num \
< $tty done
```

Pierre Ficheux, pficheux@com1.fr

A Sendmail és a POP3 beállítása

Hogyan állíthatnám be a sendmailt úgy, hogy a belső hálózatomban POP3-at használhassak? A hálózatban egy Linux-kiszolgáló (satish.enet.com) található, melyen DNS és sendmail is van, a DNS-t megfelelően beállítottam. A linuxos gépek felhasználói levelek fogadására és küldésére is képesek, az Outlook Express programmal a Windows 9x gépek viszont nem.

Dasi Satish, sdasi@manraonline.com

A POP3 engedélyezéséhez törölnöd kell a megjegyzéseket a /etc/inetd.conf fájl pop3 bejegyzése elől, majd újra kell indítanod az inetd folyamatot az alábbi paranccsal:

```
/etc/rc.d/init.d/inet restart
```

Ezután az Outlook ügyfélben a satish.enet.com POP-kiszolgálót (beérkező levelek kiszolgálója) kell beállítani. Ehhez természetesen a Linux-kiszolgálón létre kell hozni a POP-

azonosítókat és jelszavakat. A linuxos gépen futó sendmail pedig az Outlook-ügyfelek számára nyújt SMTP (kimenő levélforgalom) szolgáltatást. Lehet, hogy a /etc/mail/access fájl ilyen sorokkal kell kiegészítened:

```
aaa.bbb.ccc.ddd RELAY
```

Itt aaa.bbb.ccc.ddd a hálózat gépeinek IP-címe. Így a PC-k úgy használhatják a Linux levelezési szolgáltatásait, hogy közben nem kell a „relaying denied” hibáüzenetektől tartanod. Töröld a /etc/mail/access.db fájlt, majd indítsd újra a sendmailt a

```
/etc/rc.d/init.d/sendmail restart
```

paranccsal, és az Outlookot futtató gépek máris fogadhatnak és küldhetnek leveleket a Linux-kiszolgálón keresztül.

Felipe E. Barousse Boué, fbarousse@piensa.com

A sendmail nem POP-kiszolgáló. A POP egy levélfogadási szolgáltatás, az állandó kapcsolattal nem rendelkező ügyfelek ennek segítségével tölthetik le a leveleket a kiszolgálóról. Az SMTP pedig egy levélküldési szolgáltatás, melyhez az szükséges, hogy a távoli rendszernek állandó kapcsolata legyen. Neked egy POP démonra van szükséged, az Interneten sok ilyen találhatsz, de biztosra veszem, hogy a Linux telepítésekor a rendszerre is felkerült egy. *Chad Robinson, crobison@rfgonline.com*

A laptop PCMCIA kártyája

Épp mostanában telepítem a Red Hat 6.1-et egy laptopra. Miként érhetném el azt, hogy a PCMCIA kártyát olvassa el és töltsse be, nem pedig az eth0 csatlakozót?

Anthony G., anthonyvns@relaypoint.net

A /etc/rc.d/rc3.d/* indítófájlokban ellenőrizd a PCMCIA és az eth0 indításának sorrendjét. Ha a fájlok nevében lévő számokat kisebbre vagy nagyobbra írod át (\$S10xxxxx, \$S20xxxxx stb.), akkor megváltoztathatod a sorrendet. Mivel számos fontos indítási szolgáltatás található itt, körültekintően írogasd át a neveket! A chkconfig -help parancsot is megnézheted; ezzel az egyik szolgáltatást ki-, a másikat bekapcsolhatod. Végül, a PCMCIA vagy az Ethernet kártya indításához és leállításához az alábbi parancsokat kell használni:

```
/etc/rc.d/init.d/pcmcia [stop|start|restart]
ifdown eth0
ifup eth0
```

Felipe E. Barousse Boué, fbarousse@piensa.com

A Linux Journal honlapján további segítséget találhattok számtalan gond megoldásához. A Sunsite tüköroldalait, a gyakran feltett kérdéseket és egyéb útmutatásokat a <http://www.linuxjournal.com/> honlapon olvashatjátok el. A rovatban közzétett válaszokat Linux-szakértők kis csapata készítette el. Szívesen fogadják további kérdéseiteket, a <http://www.linuxjournal.com/lj-issues/techsup.html> címen, ahol csak egy kérdőívet kell kitöltenetek angol nyelven, de levelet is írhattok a bts@ssc.com címre. A levél tárgyában szerepeljen a „BTS” kulcsszó.

Tegyük fel, hogy a hirdetési ipar csak egyetlen kísérlet az automatizálás alapjainak kiterjesztésére a társadalom minden területén.
– Marshall McLuhan

Mi, mint energiaforrások, könnyen megújíthatók és teljes egészében újrafeldolgozhatók vagyunk. Ahhoz, hogy irányíthassák ezt az újfajta energiaforrást, csupán birtokba kellett venniük az agyunkat.
– Morpheus a Matrixból

Az élet garázsában szerelők és vezetők vannak. Mi szerelőket keresünk. Lépj be a szervizbe!
– falfirka



A cső vége

Az a csalogató üzletág, amit kereskedelmi televízióknak nevezünk, készen áll a megmentésünkre...

Emlékszem a napra, amikor rájöttem, hogy a televízióknak kábító hatása van. 1985 tavaszán történt. Egyik este munkából hazaérve láttam, hogy tizenéves lányom és fiam meredt szemmel és tátott szájjal bámulják régi, 14"-os Toshiba televíziókat. A készülék elé léptem, a hátam mögé nyúltam és levettem a hangot. – Csak kíváncsiságból érdeklődöm: Mit néztek? – kérdeztem.

Mindkettő olyan arckifejezéssel ült ott, mint hal az akváriumban. Szinte láttam az üres gondolatléggömböket, amint kipukkadnak a fejük felett.

– Tessék? – kérdeztek vissza értetlenül.

– Tudjátok, mit néztek??

– Háát...

Nem tudták. Tényleg. Fogalmuk sem volt róla.

– Hmmm... Odébb mennél? – kérték kórusban. Tulajdonképpen odébb mentünk. Amerikán keresztül, egészen Palo Altóig, ahonnan – megváltásként – hiányzott a kábelrendszer. Hirtelen újra azt néztük, ami a tetőnennán keresztül bejött. Soha többé MTV-t. Soha többé HBO-t. Soha többé TNT-t, A&E-t vagy CNN-t.

Amikor a Toshiba tévénk elromlott, kicseréltem egy 21"-os RCA-ra, amit tíz dollárért vettem egy kiárúsításon. A dolog úgy kezdődött, hogy az esti szundikálást tökéletesen megakadályozó képernyővédő funkció jelent meg: a kép fokozatosan sötétedett, amíg erőteljesen meg nem ütögettük a készüléket. Mivel körülbelül öt percig tartott a sötétedési folyamat, így csak korlátozott ideig tudtunk nyugodtan pihenni.

Lassan teljesen magunkba fordultunk. Természetesen nagy ritkán filmeket is néztünk a videomagnón, esetleg az alkalmankénti PBS különkiadásokat. Viszont megszűnt a bamba semmittevés, a filmek végén többé nem medretünk órákon át a hirdetések családkeire. Hamarosan mindkét gyerek jobban tanult. A kezdeti évekhez képest pedig több száz ponttal javultak a lányom eredményei, és végül 3,9-del diplomázott a Berkeley-n. A fiam eredményei szintén hasonlóan alakultak. Azóta vártam arra, hogy hasonló szundikálásellenes dobozok hozzák el a megváltást a fogyasztói társadalom többi részének. Most megérkezett. Az a csalogató üzletág, amit kereskedelmi televízióknak nevezünk, készen áll a megmentésünkre, köszönhetően

a mi régi tíz dolláros ébrentartó készülékünk új típusú – és egyre célzatosabb – változatának. A neve TiVo vagy RePlayTV. Mindkettő olyan doboz, ami folyamatosan rögzít mindent, amire kíváncsi vagy, majd lehetőséget ad annak megválasztására, hogy mikor és hogyan akarod megnézni. Alapjában véve videomagnó továbbfejlesztései, de szalag helyett lemezre veszik fel az anyagot, és a használatuk is egyszerűbb. A műsort meg is állíthatod, amíg telefonálsz vagy kiszaladsz a fürdőszobába. A legfontosabb tulajdonságuk azonban az, hogy egy pillanat alatt átugorhatsz a reklámokhoz. Ez az, ami elsöpri a kereskedelmi tévéket.

A The New York Times Magazine egy nagyszerű cikkében (amit a <http://www.nytimes.com/library/magazine/home/20000813mag-boombox.html> címen lehet olvasni) Michael Lewis számol be arról, hogy a TiVo és RePlayTV-vásárlók 88 százaléka nem nézi meg a felvett reklámokat. Valószínűleg kihagynak egyebeket is, de ez nem baj. Lewis véleménye szerint csak az számít: – „Ha senki nem néz reklámokat, akkor nem léteznek többé kereskedelmi televíziók.”

Szólj hozzá!

Ha megnézzük a dolgok háttérét, észrevehetjük, hogy a TiVo operációs rendszere a Linux. Ugyanígy Linuxot találunk az új típusú beágyazott rendszeren is, mint amelyeket például a set-top gépeket gyártó cégek használnak. Ők azok a szállítók, akik hatalmas üzletet látnak a televíziókat működtető programokban, és amelyeket a televíziós ipar is követni fog, amikor tétlen fogyasztói végre tevékenyebbé válnak.

A dolog szépsége az, hogy a kereskedelmi televízió mindig is a fogyasztóhoz szólt. Amit fogyasztói társadalomnak nevezünk, az egyben egyfajta termelői társadalom is. Teljes egészében annak a szállítói rendszernek a termelői oldala szabja meg működését, mely a szobánkban, a cső szélesebb felén végződik.

Különböző mértékben ugyan, de mindegyikünk része ennek a rendszernek. Mi még minduntalan szállítási időszakokban gondolkodunk és beszélünk az üzletről. Termékünk továbbra is valamilyen „tartalom”, amit „címzünk”, „csomagolunk” és „szállítunk” a „fogyasztónak” vagy „végfelhasz-

nálónak”. Továbbra is kézbesítő szolgáltatóról beszélünk. Továbbra is a csőben gondolkodunk. Az üzlet nem kézfogas vagy valami-féle társas viszony kiépítése, hanem egy cső. Az embereknek üzletelünk, és nem velük. Ennek bizonyítékaként nézzük, hogy milyen szállítási irányokat feltételezünk a kettes szám után a B2B és B2C rövidítésekben. Hatalmas különbség van aközött, ha az embereknek üzletelünk, vagy ha velük üzletelünk. Ha kettesekben gondolkodunk, olyan mélyre kerülünk a csőben, mint a lemezkiadók vagy a televíziós társaságok vezetői. Ne feledkezzünk meg a TiVO és a RePlayTV viszonylag felvilágosult készítőiről sem. Végző üzleti terveikben már bizonyára szerepel a végleteleg személyre szabott hirdetések továbbítása egyenesen a mi kis válogatós agyunkba. Ez az, amiért a felhasználók után kémkedő rendszert fejlesztettek, ami az FSF-hez teszi hasonlónvá a DoubleClicket. Ezek a dobozok minden választásodat nyomon követik, és – ahogy azt Lewis mondaná – elemi részletességgel rögzítik. A látszólagos ok az, hogy a dobozt jól nevelt szolgáltatóvá akarják tenni, mely tapasztalatai alapján meg tudja becsülni, hogy milyen hirdetések nyerik el tetszésedet. A nyilvánvaló gazdasági ok azonban az, hogy minél pontosabb és pontosabb célkeresztet rajzolhassanak a tarkódra. Amikor ezekről a dobozokról – és a társadalomra gyakorolt hatásukról – olvasol, vedd figyelembe az összes szempontot. Vedd észre, hogy szinte mindig a csőnek a termelői végre helyezi önmagát. A piacot még mindig valamilyen távoli dolognak tartja, valamilyen kényszerítő erőnek, népességi statisztikának, esetleg a szükség szinonimájának. Soha nem egy valóságos hely, ahol a közösség tagjai találkoznak, hogy üzleteket kössenek, kultúrát hozzanak létre, amit a piac az ipari forradalom győzelme előtt jelentett. Jó példáját láthatjuk ennek az öntudatlan szempontnak Lewis írásában: „Sok dolog meg fog változni, amikor a televízió képes lesz előre beállított üzeneteket suttogni a fogékonyabb fogyasztóknak, hát még akkor, ha millióknak közvetítenek sőtét, kegyetlen üzeneteket. Az üzenetek ára lesz az egyetlen, ami meg fog változni. Ha ők könnyebben elérhetőek lesznek, a célpontok összezsugorodnak, de mivel a célpontok összezsugorodnak, az eszköznek is össze kell zsugorodnia, ami megtalálja őket.” Mi vagyunk az „ők”, többes szám harmadik személyben. Nos, nézzük meg közelebbről az „öket”. Tényleg csak merev szemű, a gyártásra visszameledő egyének a cső fogyasztói végén, a hangszóró szélesebb felén? Ó, nem. Itt kint, ez az igazi piac. Ez a bazár. A hangszórós emberek ezt nem érthetik, mert soha nem volt gazdasági kapcsolatuk velem. A kereskedelmi televí-



ziós üzletág termelői részének a nézők soha nem voltak fogyasztók. Vásárlók voltak. Vagy Jerry Michalski találó szavaival élve: „balekok, akik csak azért élnek, hogy benyeljék a termékeket, és a kitegyék a lóvét”. Az igazi fogyasztók a hirdető cégek voltak. Itt kint a bazárban a kereskedelmi televízió éppen olyan elveszett, mint az az ember, aki az utcákon sétál Paul Simon: *You can call me Al* című dalában.

*Ez egy utca egy különös világban.
Talán ez egy harmadik világ.
Talán most van itt először.
Nem beszél a nyelvet.
Nincs pénze. Egy idegen ember.
Körülveszik a hangok...
Marha a piacon. Reménytelenség és árvaság...*

A piacok mindig a fogyasztók és a szakemberek közötti kapcsolatról szólnak. Azok a szakemberek, akik itt számítanak, ugyanazok, mint akik közszükségleti cikket szeretnének varázsolni a PC-ből, és minden termékbe PC-t szeretnének. Ezek a srácok egy olyan gépezet részei, amit nem valamilyen nagy gyártó szabályoz, hanem párbeszédéke százai. Ezek pedig új termékek kiötéséről vagy gondok megoldásairól szólnak, és egy újfajta háttérágazat kiépítésével – ezt a Linuxhoz vagy a Nethez hasonlóan senki sem birtokolja, és senkinek sem a szolgálja – lehetővé teszik mindkét tevékenységet. A világon mindkettőt könnyen és olcsón használatba lehet venni. Feltalálni, vagy csatlakozni nagyon könnyű, és nem csak azért, hogy a számítógépek linuxos dobozokká váljanak. A tervezők és programo-

zók apró csoportjai számára egyre egyszerűbbé válik, hogy gyakorlatilag bármibe intelligens vezérlést készíthessenek és ágyazzanak be. Azoknak az árucikkeknek a csoportja, amelyekhez a tömeggyártás szükséges, folyamatosan csökken. A nagy gyártók, akik számítanak, azok a szilíciumgyártók – a világ Inteljei, Motorolái, Samsungjai és Hitachijai. A beágyazott Linux csak azoknak a kis sorozatban, egyedi termékek készítő cégeknek csökkentheti a költségeit és a termékeik piacra kerülésének idejét, melyek a mindennapi árucikkek számító szilíciumot és a többi alkatrészt életre keltik. Néhány héttel ezelőtt Don Marti szerzőtársam és jómagam egy programozóval beszélgettünk, aki otthagya az egyik dobozos linuxot fejlesztő céget, és a Kerbangóhoz ment dolgozni, mely Linux-alapú rádiókat készít webes adásokhoz. Megunta a kiszolgálófürtökkel való munkát és a többi mindennapos tennivalót. „Az is jó dolog volt” – mondta –, „de ez itt igazán szuper.” Ez a fiú csak egyike volt az elsőnek. Vannak ezek, talán milliók, sokan közülünk a való világban, akik vágyódnak szuper, új dolgok feltalálására. Hála a beágyazott Linuxnak, ez egyre könnyebb és könnyebb lesz. Mint a fogyasztás gyógymódját, nagyon nehéz lesz legyőzni.



Doc Searls (doc@ssc.com)
A Linux Journal szerkesztője. A fentiek az ő egyéni véleményét, és nem a Linux Journal vagy az SSC álláspontját tükrözik.

A Linux és a hálózatok: egy újabb forradalom

A számítástechnika világában bekövetkezett változások nagymértékben megváltoztathatják azt, amit eddig a hálózatokról gondoltunk.

A „forradalom” kifejezéssel általában a heves és alapvető változásokat hozó eseményeket illetjük. Ha láttuk, hogy egy forradalom mit hozott, még nem biztos, hogy következtethetünk a hirtelen változás hatásaira. A forradalom új fejlődési irányokat szab meg, és átrendezi a hatalmi erőviszonyokat is – olyan változások ezek, melyek nyugtalanítóak és izgalmasak is egyszerre. Mondhatjuk, hogy a számítógépek világa a gépek és a programok oldaláról nézve egyaránt komoly változásokon megy keresztül. Mi történhet a hálózatok világával, ha ez a kétféle változás egymást felerősítve soha nem látott dolgokat szül? Különösen az érdekes, hogy a várható változások mennyiben érintik majd a felhasználók által többségében megvásárolt rendszereket. Ugyanis ma már nagyobb teljesítményű, újfajta internetes eszközöket (nem számítógépekre gondolok!) vásárolhatunk, mint ha ugyanazt a pénzt hagyományos ájtárókra és más hálózati felszerelésre költenénk. A Linux hálózatokra gyakorolt hatása várhatóan még a kiszolgálók területén végbevitt változásoknál is nagyobb horderejű lesz.

A hetvenes és nyolcvanas években a hálózatokat a gazdagépek uralták, a feladatok nagy részét csupán néhány nagyszámítógép végezte. A felhasználók a gépeket egyszerű szöveges terminálok segítségével kezelhették.

A kilencvenes évek elejére a hálózati rendszerek területén már csak néhány szabvány maradt, ezek azonban megerősödtek, általánosan elfogadottá váltak. Az asztali munkaállomásokban is a helyi hálózatok kerültek az érdeklődés középpontjába. Míg a hálózati szolgáltatásokat több kiszolgáló látta el, a hálózati kapcsolatok fenntartása a gazdagépek helyett egyre inkább a kifejezetten e célra készített hálózati eszközökre (útválasztókra, hálózati kapcsolókra és hozzáférés-kiszolgálókra) hárult. Ezek a hálózatkezelést megbízhatóbban és költséghatékonyabban valósították meg. Az útválasztó (vagy éppen ájtáró) által megteremtett hálózati kapcsolat és a kiszolgálók nyújtotta szolgáltatások határozottan különválnak egymástól – a legtöbben a mai napig így gondolkodnak a hálózatokról. Az olcsó alkatrészek és a nyílt forrású programok azonban megváltoztatni látszanak e helyzetet.

A késő nyolcvanas években bekövetkező PC-forradalom hatására csaknem minden íróasztalra számítógép került. Az éles minő-



ségi versenynek és a szabványosított gép- és programgyártásnak köszönhetően a PC-k egyre olcsóbbá és nagyobb teljesítményűvé váltak. Az Internet elterjedése tetőzte be a folyamatot, ennek következtében óriási mértékben megnövekedett a házi számítógépek iránti kereslet. A vállalatok sokkal több részegységet gyárthattak le, így az előállítási költségek és ezzel együtt az eladási árak is zuhanni kezdtek. A PC Magazine 1995. májusi száma így jellemzi az akkori helyzetet: „2000 dollárba (mintegy 300 000 forint) kerül egy Pentium/75 multimédiás számítógép, 8 MB memóriával, 750 MB merevlemezzel és 15”-os monitorral.” 1997. januárjában ugyanez a lap már a következőket írja: „Mit vehetünk tehát 1000 dollárért (nagyjából 300 000 forint)? Egy 120–133 MHz-es, teljes kiépítésű Pentium gépet.” Most, az új évtized elején, még ennél is olcsóbbak a PC-k. Százötvenezerért legalább egy 500 MHz-es, 128 MB memóriával épített gépet vehetünk, számos kiegészítővel. Ez az ár körülbelül annyi, mint amennyit egy útvonalválasztóért fizetnénk, mely egyébként jóval szerényebb teljesítményű. Mivel a kiszolgálók és a munkaállomások bel-

ső felépítése teljesen azonos, a gyártók a részegységek alacsony árát kihasználva hasonló teljesítményű, de megfizethetőbb árú kiszolgálókat építhetnek. Bár az ájtárók kiszolgálókkal történő helyettesítésének az alacsony ár volt az oka, ez ma már nem feltétlenül van így. A hagyományos alkatrészek olyannyira olcsóvá válnak, hogy a nem számítógép-típusú hálózati alkatrészek gyártói nem képesek lépést tartani ezzel. A folyamat hamar eléri azt a tőrészhatárt, ahol csak egy igazi újdonság megjelenése jelenthetné az áttörést. Ez az újdonság minden bizonnyal a Linux lesz. A rendszer mindenféle korlátozás nélkül szabadon terjeszthető, és a forráskódhoz is bárki hozzáférhet. Nem feltétlenül ingyenes, de tetszőlegesen testre szabhatjuk, és ezen változtatásokat másokkal is megoszthatva a Linux tovább és tovább fejlődhet. Az IDC augusztusi adatai szerint tavaly a Linux volt a második legnépszerűbb, illetve a leggyorsabban növekvő táború operációs rendszer a kiszolgálók piacán (24 százalék új telepítéssel a Windows 36 százalékos növekedést könyvelhetett el, de ezt a számot az összes Windows-változat együttesen adja). A Linux a legnépszerűbb rendszer az inter-

netes alkalmazások körében, és az üzleti felhasználók száma is rohamosan nő. A Linux hálózatkezelő képességei a kezdetektől fogva jóval több és megbízhatóbb szolgáltatást nyújtottak, mint más operációs rendszerek, s ez a helyzet azóta sem változott. A nyílt forráskódnak és a hatalmas, világméretű fejlesztői csapatnak köszönhetően a Linux minden másnál rugalmasabb, és fejlődése is sokkal gyorsabb vetélytársaiénál. A Linux operációs rendszer lehetőségei és megbízhatósága minden további nélkül alkalmassá teszi arra, hogy különleges hálózati és internetes feladatokat ellátó rendszer váljon belőle. Ehhez vegyük hozzá a csúcsmínőségű alkatrészeket: az eredmény egy igen nagy teljesítményű hálózati felület. Egyértelmű, hogy a Linux megváltoztatja a kiszolgálók piacát: a 24 százalék az majd egy negyed, tetszik vagy nem tetszik. Bár a munkaállomás/ügyfél oldalra gyakorolt hatásai még nem körvonalazódtak, az biztos, hogy a hálózati piacot fenekestül fogja felforgatni. A forradalom korai szakaszát éljük, és a módszerek egyesítése lesz az a fordulópont, mely az igazi áttörést előidézhetheti. Néhány, az eljárás módokban járatosabb felhasználó maga végzi az összevonást: hálózati kártyákból és PC-s alkatrészekből építi fel saját, Linux-alapú hálózati gépét. Erre azon internetszolgáltatók szolgálnak például, melyek a behívó szolgáltatást nem PPP hozzáférés-kiszolgálóval, hanem modembankokhoz kapcsolódó, rengeteg soros csatlakozóval ellátott linuxos kiszolgálókkal valósítják meg. Néhány, a módszerek egyesítésében élenjáró cég már meg is kezdte a következő nemzedékbeli hálózati eszközök gyártását. A Cobalt Qube például egy kis- és középvállalkozások számára készült internetes átjáró, melyet egy kártyával ellátva a WAN-kapcsolat is megoldható. Az eszköz minden internetes szolgáltatást magában foglal, beállítása nagyon egyszerű, és az ára körülbelül megegyezik az általa helyettesített és szerényebb képességű hagyományos átjáróval.

Ez az új hálózati eszköz azonban nem csak az átjáró helyettesítésének olcsóbb módja, hiszen bővíthetőbb és rugalmasabb is annál. Ahogyan a régi hálózati igényeknek a hagyományos átjárók feleltek meg, úgy a jövő elvárásait csak az új generációs hálózati eszközök elégíthetik ki.

A felhasználók olcsóbb, jobb és egyszerűbben használható új eszközöket vásárolhatnak, ezek helyettesíthetik a hozzáférés-kiszolgálókat és az átjárókat, új hálózati szolgáltatásokat valósíthatnak meg, s minden alkalmazáshoz testre szabhatók. Ebben az esetben tehát egy egészen más típusú termékről van szó.

E változásokban a Linux szerepe világosabban látszik, ha elolvassuk a

☞ <http://www.idc.com/> címen található legfrissebb adatokat. A kiszolgálók és ügyfe-



lek számára készült operációs rendszerek forgalma tavaly tizenhétmilliárd dollár volt. A Windows ebből közel nyolcmilliárdot, a Linux alig százmilliót mondhat magáénak. A Linux piaci részesedésének ismeretében a témában kevésbé járatosak számára ezen adatok megdöbbentők lehetnek. A felhasználó szempontjából a rendszerek értékét nem a használt operációs rendszer befolyásolja, tehát joggal várhatná el, hogy a bevételek a piaci részesedéssel egyenes arányban álljanak. Ha a Microsoft nyolcmilliárd dollár bezebelet, akkor hová vándorolt a Linux haszna? A válasz egyszerű. Mivel a Linux nyílt rendszer és a hozzá kapcsolódó üzletet teljesen átjárja a nyílt forráskód szellemisége, a „haszon” sokkal inkább a felhasználók megtakarításaiban jelentkezik. Mivel a Linux és a hozzá kapcsolódó programok ingyenesek, a felhasználó a megmaradt pénzt más eszközökre fordíthatja. Ez egyértelműen oda vezet, hogy a Linuxra épülő rendszerek mindenképpen nagyobb teljesítményűek lesznek.

Így már talán nyilvánvalóbb a Linux által előidézett változások jelentősége. Írások témája azonban a hálózatkezelés, nem pedig maga az operációs rendszer. A Data Communications legfrissebb adatai szerint a hálózati piac tavaly csupán az Egyesült Államokban hetvenmilliárd dollár bevételt hozott, az egész világon pedig százhuszmilliárd dollárt. Ez a pénz teljes egészében a hálózati eszközök gyártóinak zsebébe vándorol (Cisco, Nortel stb.). Amikor tehát a nyílt rendszerek váltják fel majd a hagyományos eszközöket, a pénz hirtelen felszabadul, így máshová kerülhet. A hálózati piacon ez az átmenet már elkezdődött mind a programok, mind a gépek terén.

Eddig azért használtak a legtöbben hagyományos hálózati eszközöket, mert azok jóval költségkímélőbb megoldást kínáltak a kiszolgáló-alapú megoldásokkal szemben. A Linux és a PC-k felszabadíthatják a piacot, és így a módszereket egyesítő cégek tökéletesebb rendszereket építhetnek ki, ezzel pedig versenyképesebbek lehetnek. A változást a jobb megoldások iránti igények viszik előre, a nyílt forráskód és a szabványos alkatrészek pedig lehetővé teszik ennek kiteljesedését. E változások fontos következményekkel járnak. Ma még a legtöbb felhasználónak a hagyományos hálózati eszközökkel kell beérnie. Ezek jól kezelik a kapcsolatokat, de nem bővíthetők új szolgáltatásokkal, és nem is alakíthatjuk át őket igényeink szerint. Tehát a felhasználó számára olcsóbb, ha a hálózati kapcsolatért, illetve a hálózati szolgáltatásokért felelős rendszereket elvlasztja egymástól. Akkor is ezt a megoldást választja, ha így a hálózat sokkal kezelhetlenebb, gondokkal terheltebb, ráadásul kevesebbet is tud, hiszen anyagilag jobban jár. A közeljövőben viszont a kapcsolatért felelős eszközök a nyílt forráskódú operációs rendszerek beépítésével jóval okosabbak lesznek. Ezenkívül az új gép- és programmegoldások egyesülése szintén felgyorsul. A felhasználók és a módszereket egyesítők pedig nem egy kapzsi vállalatól függenek majd: az irányítás a lánc legfontosabb tagjához, a felhasználóhoz kerül. Mint minden változás esetében, itt sem könnyű elsőre megérteni és felmérni az új típusú megközelítés jelentőségét és hatását.

Marcio Saito (marcio@cyclades.com)
a Cyclades Corporation szakmai igazgatója.

A tökéletes Linux munkaállomás

Megkérdeztünk néhány gépszakértőt arról, hogy milyen alkatrészekből építenék fel a tökéletes Linux munkaállomást. Az alábbiakban tehát egy csúcsteljesítményű 3D grafikával, gyors háttértárral, kristálytisztá hanggal és még ki tudja, mi mindennel bíró fenevad „receptje” következik.

Még 1996-ban jelent meg *Eric Raymond* nagy sikert arató cikke a *Linux Journal*-ban. Aki egy kicsit is figyel az újdonságokra, annak nyilvánvaló, hogy a pár éve javasolt Pentium 133/166-os processzor és a 2 gigabájtos merevlemez fölött kissé eljárt az idő. Így ismét kikértük e témában Eric és más szakértők véleményét.

A processzorok

1996-ban Eric azt ajánlotta, hogy „a csúcsmoделlektől egy-két osztállyal kisebb típusokat válasszuk”. Ez még ma is megszívlelendő tanács. A leggyorsabb processzorok és az eggyel lassúbbak közti árkülönbség által megtakarított összeget inkább költsük egy kis ráadás memóriára vagy más alkatrészekre, nem fogjuk megbánni. Bár Eric javaslata jó iránymutatás, ennek ellenére (mivel most a TÖKÉLETES linuxos munkaállomást állítjuk össze) mi a leggyorsabb processzorokat fogjuk választani. Igen, processzorokat. A többes szám jogos, hiszen ha az egy időben futó feladatok és folyamatok (MP3-készítés, GIMP stb.) külön-külön processzort használhatnak, az nagymértékben növeli a rendszer teljesítményét. Ha azonban szorít a költségvetés, de szeretnénk, hogy programjaink gyorsan induljanak, akkor az SMP nem sokat segít. Ilyen esetben inkább egyprocesszoros rendszert építsünk ki, mondjuk egy-két tízezres fordulat-számú SCSI merevlemezzel. Ez a megoldás jóval gyorsabb, mintha egy SMP-rendszert használnánk lassú merevlemezekkel.

Milyen processzort válasszunk? *Daryll Strauss* azon az Alpha-alapú linuxos gépekből álló telepen dolgozott, ahol a *Titanic* című film jeleneteinek leképezése folyt. „Imádom az Alphákat” – így Strauss. „Otthon egy kétprocesszoros 264-es rendszerem van, és teljesen elégedett vagyok vele. A cég nagyszerű munkát végez: nemsokára kihozzák gépeik következő nemzedékét, ezekben még gyorsabb processzorok, buszok és AGP grafikus kártya kap helyet. A processzorok egyetlen hibája, hogy túlságosan is egyediek. A kifejezetten az Alphára készült alkalmazásokat egyszerűen lehetetlen felülmúlni teljesítményben, ha azonban általánosabb célú rendszert kívánunk kiépíteni, akkor inkább Intel-megfelelő processzort válasszunk.”

A legtöbb fejlesztő Intel-típusú processzorokra írja, és azokon is ellenőrzi a programokat, így mi is a legnagyobb teljesítményű Intel-megfelelő processzorokra szavazunk. Az AMD processzorai még programozási szinten is csereszabatosak az Intel termékeivel, de sajnos egyetlen olyan alaplapt sem ismerünk, amely képes lenne két AMD fogadására.

Az alaplapp

Az alaplappínát olyan gyorsan változik, hogy egész honlapok megélnék az új típusok megjelenéséről szóló hírekből. Milyen elv szerint válasszunk? *Jason Collins*, a VA Linux Systems programmérnöke és a Cerberus fejlesztője állít össze egy otthoni linuxos gépet, s a következő jó tanácsal szolgál: „Mindig olvassuk el az alaplappok felhívásával foglalkozó géppörültek honlapjait. Ha egy alaplapp felhívott órajellel is megbízható, akkor ez záloga annak, hogy rendes órajelen mindig kifogástalanul fog működni.”

„Ne sajnáljuk a pénzt, ha van rá keret, egy jó kis Tyan mellett döntünk.”

– mondja *Sam Ockman*, a Penguin Computing igazgatója. A megbízható Tyan alaplappok neve nagyon jól cseng a linuxos gépek eladásával foglalkozó cégek körében. *Mike Higashi* linuxos tanácsadó, aki gyakran saját maga építi össze az igényelt gépeket, az Asus modelljeit ajánlja.

Daryll Strauss az Asus P2B-DS típusú alaplappját használja, ennek lelkét az Intel régi, de megbízható 440BX lapkakészlete alkotja. A BX alaplappok még mindig jó választásnak bizonyulnak az SMP számára, ha azonban erre nincs szükségünk, akkor ABit KA7-100 alaplappot és Athlon processzort ajánlunk.

Az alaplapp megvásárlása előtt győződjünk meg arról, hogy a kártya támogatja-e az ECC memóriát, illetve található-e elegendő RAM-illesztő rajta. Egy tökéletes Linux-munkaállomás memóriáigényét és a folya-



matos működést feltételezve elmondhatjuk, hogy aki nem ECC memóriát használ, az hamarosan szembenézhet a „biteldobás” nevű aranyos kis hibával. Létezik egy új „Linux ECC” meghajtóprogram, ez az egy bites ECC hibákat naplózza (különben a hibáknak nem lenne nyoma). Ha az alaplap támogatja ezt a szolgáltatást, akkor mindenképpen próbáljuk ki.

A grafikus kártya

Nyissunk meg egy nagy fehér ablakot a képernyőn és nézzünk el a monitor fölött a távolba. Ha a monitoron eközben enyhe villogást látunk, akkor a monitor vagy a grafikus kártya gyenge teljesítményű, vagy nincsen megfelelően beállítva. Ezt a villogást a képernyőt nézve általában nem érzékeljük, ennek ellenére szem- és fejfájást, illetve más egészségügyi gondokat okozhat. A legtöbb ember szeme 85 Hz fölött „nyugszik meg”, tehát a gép grafikus kártyájának legalább ezt az üzemmódot mindenképpen ismernie kell. Egy tökéletes linuxos munkaállomás ezt a képfrissítési módot nagy felbontásban, 24–32 bites színmélységgel is képes megjeleníteni. Ha a 3D nem érdekel bennünket, a szemünk egészségének megőrzéséért akkor is vegyünk egy jó minőségű kártyát.

A tökéletes linuxos munkaállomás viszont 3D gyorsításra is képes. Daryll Strauss szerint: „Most mindenki a Radeonra figyel, az ATI ezzel magasra tette a léceket a többi gyártó előtt. A kártya teljesítménye a GeForce2-ével azonos, és nemsokára a nyílt forrású meghajtók is megjelennek hozzá.” A linuxosoknak azonban erre még várniuk kell egy kicsit. Érdemes tehát rendszeresen olvasgatni kedvenc linuxos

honlapjainkat. Ami pedig a sebességet illeti, jelenleg az nVidia GeForce2 viszi el a pálmát, ez Daryll Strauss szerint: „nagyszerű kártya, a meghajtókat leszámítva”. Idén januárban a VA Linux Systems a Linux-közösség körében nagy felzúdulást keltve megjelentette az nVidia kártyákhoz készült gyors, de sajnos nem nyílt forráskódú 3D-s meghajtóit.

Nem a legjobb hasonlat, de a bináris meghajtók beillesztése egy alapvetően forráskódból fordított programokból építkező operációs rendszerbe olyan, mintha egy macskát evőpálcikával etetnénk. Tehát amíg az nVidia nem lép ez ügyben, addig jobban tesszük, ha a második vonalbeli kártyákkal foglalkozunk. Ilyen például a 3dfx cég Voodoo5-ös vagy a Matrox új G450-es modellje. Jason arra is felhívja a figyelmünket, hogy a Matrox kártyák egy gyors központi processzorról csodákra képesek: „Egy elég gyors processzorról az nVidia sebességét is elérhetjük.”

A hangkártya

A Creative Labs nemrég még ugyanabban a helyzetben volt, mint most az nVidia. Eleinte a csúcsmodellnek, a SoundBlaster Live!-nak nem volt nyílt forráskódú meghajtója. Miután ráeszméltek, hogy az egészen csak nyerhetnek, a Creative tavaly elindította új honlapját opensource.creative.com, erről letölthetjük a Live! nyílt forrású meghajtóját. A kártyához most már ALSA-támogatás is jár. A Creative egy új szabvány, az OpenAL fejlesztését is támogatja, ezt egyszerűen a „hangkártyák OpenGL-jének” nevezik. Ennek segítségével a fejlesztők rendszerfüggetlen térbeli hangalkalmazásokat készíthetnek.

© Kiskapu Kft. Minden jog fenntartva



A Live! hangja gyönyörű, és megvásárlásával a nemsokára szélesebb körben is elterjedő térbeli hangrendszerek világába is belépünk. A garanciakártyára feltétlenül írjuk rá, hogy „Linux”. Akinek „csak” megbízható és olcsó hangkártyára van szüksége, az is nyugodtan elfelejtheti az ősréginek számító ISA csatlakozós SoundBlastereket és az ISA Plug-and-Play beállításával eltöltött „kellemes” órákat. A Creative Ensoniq AudioPCI ES1371 egy olcsó, jó támogatással bíró PCI-os hangkártya (bár ez utóbbira a nevéből már következtethetünk).

SCSI vagy IDE?

A kiszolgálók általában SCSI meghajtókat használnak, egyszerűen azért, mert nagy átvitelt enged egyszerre több lemez esetén is, és segítségével nagyon egyszerű a további meghajtók csatlakoztatása. Az olcsóbb munkaállomásokban megfelel az IDE is, hiszen IDE vezérlőt minden alaplap tartalmaz, s a merevlemez is olcsóbbak. A „SCSI vagy IDE” vita leginkább a közép- és csúcsmínőségű munkaállomások építői között dúl.

A leggyorsabb merevlemez csak SCSI-változatban létezik. Tehát egy tökéletes munkaállomáshoz ez a megfelelő választás. A további meghajtók csatlakoztatásakor pedig értékelni fogjuk a SCSI ama tulajdonságát, hogy nem foglal le minden meghajtó egy-egy megszakítást az amúgy is szűkös készletből (az IDE viszont igen).

Az azonos gépi felépítésű SCSI és IDE meghajtók összehasonlítása során kiderül, hogy az IDE-változat gyorsabb adatátvitelre képes. Ez csupán azért van, mert az IDE csatlakozó egyszerűbb. De hallgassuk meg, mit mond erről Eric Raymond: „A jelenlegi igen gyors merevlemezeknél ez a különbség szinte elenyésző. A SCSI igazi előnye, hogy okosabb áramkörök találhatóak rajta, ennek köszönhető az is, hogy mindent összevetve a SCSI a gyorsabb.”

A tökéletes linuxos munkaállomásunkhoz a legújabb Ultra 160 SCSI meghajtókat választottuk. Két SCSI lemez esetén a viszonylag kis adatforgalom következtében nincs feltétlenül szükség az Ultra 160 szabványra, mi mégis emellett döntöttünk. Hogy miért? Először is, mi most egy tökéletes munkaállomásról beszélünk. A másik ok, hogy a SCSI kártyák közti árkülönbség elenyésző ahhoz az előnyhöz képest, amit egy jó kártyával szerzünk. Egy csúcsmínőségű kártya birtokában nyugodtan lehetünk afelől, hogy jó néhány újabb nemzedékbeli merevlemezrel is hosszú évekig megbízhatóan fog működni a rendszer.

Rick Moen egyszer azt tanácsolta nekem, hogy olyan alkatrészeket válasszak, amit a következő linuxos gépembe is nyugodt szívvel beépítenék. Amit ma a tökéletes linuxos munkállomáshoz telepítünk, lehet, hogy jövőre már a Freenet-kiszolgálónkba kerül. Az Adaptec 29160 jó példa az ilyen alkatrészekre: ezt még a 64 bites PCI csatlakozóra történő áttérés után is használhatjuk. A 29160 még annyira új, hogy a tavalyi rendszermagok nem biztos, hogy felismerik, tehát esetleg frissítésre is szükség lehet. Mindent megtudhatunk az általunk használt Linux-változat megfelelőségi listájáról.

A Symbios SCSI kártyák népszerűek a linuxos gép- és alkatrészkereskedők körében, ugyanis termékeik kitűnő teljesítményűek, valamint megfelelő támogatást kapnak.

A profi SCSI kártyára költött pénz felesleges kiadás, ha lassú merevlemezeket veszünk. Vásároljunk két vagy több tizedres fordulatszámú IBM, Quantum vagy Seagate merevlemez. Az arany szabály: mindenképpen olyan típusokat válasszunk, amelyeket a nagyobb linuxos gépgyártók is alkalmaznak gépeikben, kiszolgálóikban.

Ha inkább takarékoskodnánk, és az IDE mellett döntünk, akkor is feltétlenül ragaszkodjunk az említett három cég termékeihez. Andre Hedrick rendszermagzakértő, a linuxos IDE-meghajtóprogram fejlesztője idén egy felhasználó hozzá frott levele nyomán indított kutatás eredményeként közölte, hogy hibát fedezett fel a Western Digital merevlemezek hibaellenőrzésénél. A rendszeraggal foglalkozó levelezési listára elküldött hozzászólásában e szavakkal jellemezte a problé-



Bőséges kínálatból lehet választani az Asus honlapján

mát: „A WDC merevlemezek figyelmen kívül hagyják az UDMA CRC-ellenőrzését. Ez szerintem igen nagy marhaság”. A Western Digital honlapján nem sokkal ezután a következő sorok jelentek meg: „Ha ezen merevlemezekkel bármi gond van Linux alatt, azt minden bizonnyal a meghajtóprogram okozza és nem maga a készülék.” Én ennek ellenére Andrének hiszek és távol tartom magam a Western Digital termékeitől.

Eric azt tanácsolja, hogy saját fontos fájljainkat és a Linux-rendszert külön lemezrészeken (partíciókon) tároljuk, hiszen így egy esetleges rendszerösszeomláskor sokkal egyszerűbb a hibajavítás. „Ha két lemezrészrel dolgozunk, és a rendszer elszáll, akkor nyugodtan lehetünk afelől, hogy személyes fájljaink biztonságban vannak, és máris elkezdhetjük a rendszer újratelepítését. Ha a /home könyvtárat tároló lemezrész ment tönkre, akkor a rendszer és a hálózati kapcsolat talán még mindig működőképes, és különböző hibajavító programokkal nekiláthatunk a még megmenthető adatok visszahozásának, majd később (ha súlyos a hiba) a biztonsági mentésből egy új merevlemezre visszaállíthatjuk az egészet.”

A két merevlemez gyorsítja is a gépet, ha a /tmp és a /home, vagy a /home és a /var könyvtárakat egyszerre kell elérnünk (és ez gyakran előfordul). A Linux RAID-rendszerével az egyik lemez egész tartalmát tükrözhetjük a másik lemezre, a legnagyobb biztonság eléréseért. De még így se bizzuk a biztonsági mentést csupán az adott gép egy másik merevlemezére, hiszen a tápegység meghibásodása mindkét merevlemez tönkretelheti. A biztonsági mentést egy hálózati helyre vagy inkább szalagos egységre írjuk.

A hálózati kártya

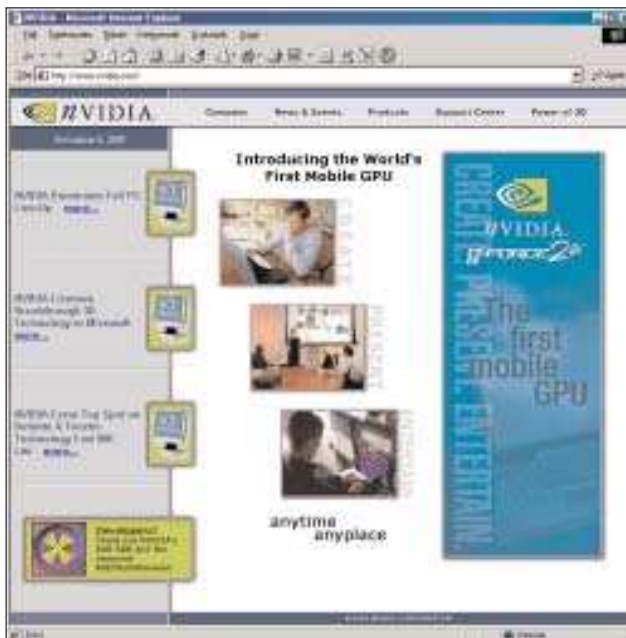
A hálózati kártyák piaca mostanában eléggé egyhangú: az Intel egyértelműen uralja a területet. Néhányan arra panaszkodnak, hogy a kártyák a meghajtóprogramokhoz képest túl gyorsan cserélődnek, s ez bizonyos típusú hálózati forgalom esetén kiszámíthatatlan galibákat okoz. Ha ma-gunk állítjuk össze a linuxos gépet, akkor készüljünk fel arra, hogy az Ethernet-meghajtó frissítésekor a rendszermagot is újra kell fordítanunk.

Egyéb apróságok: tápegység, ház, hűtőrendszerek

A legnagyobb melléfogás, ha egy gyors processzorral és meghajtókkal felszerelt gépet nem megfelelő hűtésű házba szerelünk, vagy kis teljesítményű tápegységre csatlakoztatjuk, ami ha egyszer felmondja a szolgálatot, mindent tönkretesz. A tápegység meghibásodásának egyik leggyakoribb módja, hogy a hálózati feszültség (220 V) áthalad a transzformátoron és az alkatrészekhez kerül. Ennek következményeit talán mindenki el tudja képzelni.

Lehetőleg úgy válasszunk házat, tápot és ventilátorokat, hogy azok a következő alaplapokkal és processzorokkal is használhatók legyenek. Sokan egy erős munkaállomás alaplapjából és processzorából építik fel webkiszolgálójukat. Tehát még egyszer: gondolkozzunk előre a vásárláskor, hiszen lehet, hogy a megvett alkatrészeket jövőre egy másik gépbe kívánjuk majd beépíteni.

Mike a PC Power and Cooling rendszerét használja, szerinte a második vonalban a Sparkle Power helyezkedik el. A PC Power and Cooling cég egyik legkülönlegesebb terméke a „Silencer” névre hallgató kis eszköz. Ha ezt beépítjük a gépbe, a ventilátorok, a merevlemezek stb. zajszintje jóval kisebb lesz.



Mikor nyit az nVidia?

Olyan házat válasszunk, amely jó sok helyet ad az alkatrészek és a szerelési műveletek elvégzése számára, emellett lehetőleg oldallappal és a ventilátorok számára több felfüggesztési ponttal bír. Mike terheléstűrő Linux-kiszolgálókat épít olyan cégek számára, amelyeknek nincs légkondicionált gépterme. Ehhez jól hűthető toronyházra van szükség. Mike a Supermicro házakat javasolja. Ezekben a típusokban a ventilátorok számára több felfüggesztési pont található, a hagyományos helyeken kívül a merevlemezek mellett is. A tökéletes linuxos munkaállomásba kerülő tízezres fordulatszámú merevlemezekhez elengedhetetlenül szükséges a folyamatos közvetlen légáram, különben szépen elégetik magukat.

Az AMK nevű kanadai vállalat gyártja az „Overlocker's Dream” nevű fényűző házakat, ezek tetején külön elszívólyukak találhatóak. Ezt persze csak akkor használhatjuk, ha soha nem felejtjük el, hogy a gép tetejére semmit sem helyezhetünk. Ez a módszer talán a legbiztonságosabb és leghatékonyabb az összes közül.

Jason azt is megjegyezte, hogy néha a túl sok ventilátor vagy lyuk jelent a gondot. Bizonyos elrendezések mellett egyes fontos alkatrészek

közébe csapdába kerül a forró levegő, vagy ami még rosszabb, túl sok kifújó ventilátort szerelünk be, és túl kevés szívót. A PC-jüket kedvtelésből barkácsolók néha további ventilátort vagy ventilátorokat szerelnek be a forró levegő kiszívásához, s közben elfelejtik azt, hogy ha a házban kevesebb a levegő, az legalább akkora galibát okoz, mint a meleg: a tápegység vagy a processzor ilyenkor is túlmelegedhet. A legtöbb ATX-es tápegység hátlapján egy fújó ventilátort találunk, ezenkívül néhány nyílást a processzor körül. Rendes körülmények között a ventilátor a processzor körül felforrósodott levegőt is elszívja, ezzel segítve a processzorhűtő munkáját, s közben a tápegységet is hűti. Ha azonban a ház túl sok kifújó ventilátort tartalmaz, akkor a nyomás vérszesen lecsökkenhet benne, ez pedig kiegyenlíti a tápegység ventilátorának munkáját. A tápegységben ilyenkor megáll a levegő s az egész villámgyorsan egy méregdrága kenyérpírtóvá alakul. A ventilátorok elhelyezésének első alapszabálya, hogy a fújó és szívó ventilátorok számának és teljesítményének egyensúlyban kell lennie. A második: soha ne felejtjük el, hogy a forró levegő felfelé száll, tehát a szívó ventilátorokat a gép aljára, a fújókat pedig a tetejére helyezzük. A gép dobozát mindig zárjuk be, ne üzemeltessük nyitott házzal pár percnél tovább (bár szereléskor néha szükséges). A nyitott ház ugyanis különböző külső sugárzásoknak teszi ki a gépet. „Megszólalt a mobiltelefonom, mire a gép teljesen megkattant,” emlékszik vissza szomorúan Jason.

A tökéletes linuxos munkaállomás bizony nem lesz túl halk masina. A merevlemezek, a processzorhűtők, a hátsó és az első ventilátorok... hamarosan a gép hangja egy porszívóéra kezd emlékeztetni. Jason Collins mostanában az AMK „DigitalDoc 2” nevű termékében reménykedik. Ez a vezérlőpult egy 5,25"-os meghajtó helyét foglalja el, és a ventilátorokat a beállított hőmérsékletnek megfelelően vezérli. Jason a ventilátorait a legújabb Delta modellekből válogatta, ezek másodpercenként több mint 1 m³ levegő kifújására képesek! A srác hangszigetelővel borította a ház belső oldalait, hogy a zajt és a rezgést csökkentse.

Természetesen szünetmentes áramforrásra is szükségünk lesz. Hogy miért kell ez egy munkaállomásba? Nem azért, hogy áramszünet esetén is folytathassuk a munkát, hiszen egy hosszabb áramkimaradás teljesen lemeríti a szünetmentes táp akkumulátorát. Sokkal inkább azért, hogy az áramszünet jelentkezésekor legyen időnk minden megnyitott adatot menteni, majd a munkát az áram visszatérése után folytathatjuk.

Az APC és a Tripp Lite szünetmentes áramforrásai a legmegbízhatóbbak. Ezek soros kapun keresztül is képesek kapcsolatot tartani a géppel. A Penguin Computernek választania kellett a két cég között. Amikor a Tripp Lite megkérdezte Ockmant arról, hogy miért nem az ő szünetmentes tápjakkal foglalkoznak, a Penguin Computer vezetője így válaszolt: „Majd ha nyílt forráskódúvá teszik a táp vezérlő programját.” Ez meg is történt. Ezzel szemben az APC még a használt protokoll leírását sem hozza nyilvánosságra.

Tehát amíg az APC fel nem ismeri, hogy a nyílt forráskódú terjesztés nem teszi tönkre a céget, írjuk fel őket is (az nVidiával együtt) arra a listára, amelyen azok a vállalatok szerepelnek, akiket szívesen támogatunk majd, ha egyszer észhez térnek.

- ➔ www.seagate.com
- ➔ www.smc.com/smc/pages_html/homef.html
- ➔ www.tyan.com



Don Marti

a Linux Journal szakmai szerkesztője, olvasóink a dmarti@linuxjournal.com címen érhetik el.

AMD Duron processzor

Ez a húszezer forintba kerülő processzor szövegszerkesztéshez, fotóretusáláshoz és programozáshoz tökéletesen megfelel. Az AMD Duron jó megoldást kínál a legtöbb közepes teljesítményű asztali linuxos gép számára.

A jövő asztali linuxos gépeinek két AMD Duronnak hívják. Néhány éve számos, szerényebb anyagi lehetőségekkel bíró linuxos vált az AMD K6 rajongójává – akkoriban az Intel még csak a drága Pentium II-es és a butácska, első sorozatú Celeron processzorokkal kísérlete meg felvenni a versenyt. Az AMD még mindig kifogástalan K6-2 és -3 típusokat dob piacra, ezek a legtöbb jelenlegi alkalmazásban bizonyítanak. A programok mérete azonban egyre nő, és így a felhasználók is nagyobb teljesítményt várnak el a processzoroktól. Éppen ezért a Duron átveheti a K6 helyét azon felhasználók gépeiben, akiknek nem csúcsteljesítményre van szükségük, de azt szeretnék, hogy rendszerük kiemelkedően gyors és megbízható legyen. A „túl gyors” processzor egyébként is csak pénzkidobás: ha a teljesítménytöbbletet úgysem használnánk ki, akkor inkább más alkatrészekre költünk a pénzünket.

Egy 600 MHz-es processzor jelenleg megközelítőleg nettó 18 ezer forintba kerül, tehát olcsóbb, mint egy azonos órajelen működő Celeron, viszont annál gyorsabb.

Akinek tehát feltétlenül szüksége van az „Intel Inside” matricára, az nyugodtan fizessen a Celeronért hat-nyolcezer forinttal többet (nem sok értelme van).

A Duron tulajdonképpen egy Athlon, melyből néhány dolgot kivettek, egy tulajdonsággal pedig bővítették. Kevesebb gyorsmemória található benne, ezenkívül a legmagasabb órajelű változatok nem kerültek piacra. A Duron azonban a „Socket A” csatlakozóval illeszkedik az alaplapra. Ez a megoldás megbízhatóbb működést eredményez, így a processzor kevésbé emelkedik ki az alaplapról, tehát alacsonyabb házakba is beépíthető.

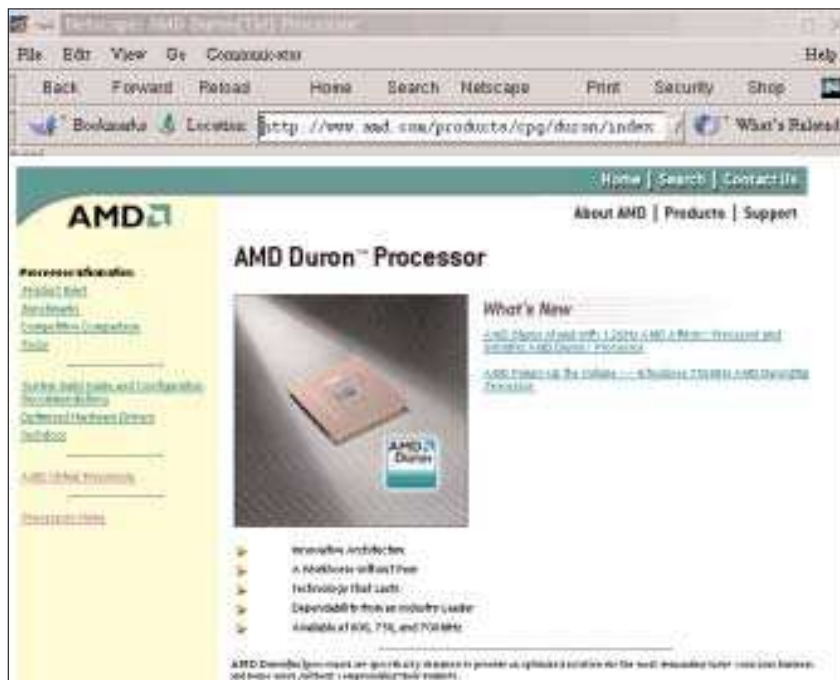
Mindez nagyon jól hangzik az alkatrészörültek számára, de a tavalyi programkínálatot figyelembe véve nem sokat jelent a programkereskedőknek. Azonban most, hogy a Sun a GNU GPL elvei szerint nyitotta tette a StarOffice forráskódját, hatalmas lehetőségek nyíltak meg azon vállalatok számára, amelyek a Microsoft Office termékek lecserélését is elfogadhatónak vélik, ha ezzel megtakaríthatnak valamennyi költséget.

A legtöbb cég egyelőre biztonságosan játszik és a szerződésekre meg a Windowsra költi



a pénzét azért, hogy ne kelljen a Microsoft Office rendszereket lecserélni, ami komoly fennakadást okozna szerintük a munkában. Ha a vállalat vezetőségi tagjai között konok Excel- vagy Access-felhasználók is ücsörögnek, akkor egyelőre tényleg nincs mit tenni. Az 1970-es években az Egyesült Államokban vagány dolognak számított, ha valaki a főnökét követve a külvárosba költözött. A számítástechnikában is hasonló jelenség figyelhető meg: a felettesünk valószínűleg egyszerűen azért ragaszkodik a Microsoft Office-hoz, mert a pénzeszsákon ülő nagyfőnök csak ehhez ért (úgyahogy). Ha azonban a vezetők semmilyen rendszer kezelésében nem járatosak, akkor valamivel jobb a helyzet. Ilyenkor ugyanis akármit adunk a kezükbe, mindenképpen panaszkodni fognak, mi meg kéjes vigyorral telepíthetjük a teljesen ingyenes GPL programokat a vállalat gépeire. Legalább megnyugodhatunk, hogy nem kell majd hallgatnunk az árak miatt, és még az internetes részleg költségvetését is karcsúbbá tehetjük, szóval a fentről jövő baráti vállveregetésekre is felkészülhetünk. Nem rossz jövőkép, ugye?

A StarOffice és a Duron a Linux előtt álló nagy változásokat képviselik. De a Linux-kereskedők még mindig főleg kiszolgálókat, kiszolgálókat és kiszolgálókat építenek, nem igaz? Nem hát. Itt van például az ASL nevű, 1995 óta tevékenykedő cég. Míg a legtöbb linuxos gépeket építő vállalat szinte kizárólag a kiszolgálók összeállításával és értékesítésével foglalkozik, addig az ASL elnöke, *Jeff Nguyen* szerint, cége eladásainak felét tették ki a munkaállomások. A Mandrake-kelet nemrég kötött megállapodás alapján az ő Linux-változatukat telepítik a szállított gépekre. Most épp egy Duron-alapú termékcsaláddal készülnek a megmérettetésre. Az ASL Duron-alapú munkaállomásának



megvásárlásakor érezhetjük, hogy egy igazán könnyed és profi megvalósítással van dolgunk. A csomagoláshoz tartozó kis fülecske az összes alkatrészt felsorolja, ez a rendszer újratelepítése során válhat nagy hasznunkra. Az egér, a billentyűzet, a vezetékek egy külön kis dobozban kaptak helyet, a gép beállítása pedig villámgyorsan elvégezhető. Sosem tudom megállni, hogy ne pillantsak egy frissen érkezett gép belsejébe, szóval három csavar eltávolítása után máris az ASL felépítését vizsgáltam. A ház teteje és mindkét oldala leszerelhető. Kizárólag csúcsmínőségű részeket láttam benne – IBM merevlemez, Matrox grafikus kártya, Intel hálózati kártya, Ensoniq hangkártya stb. Az alaplap lapkakészletet az a VIA gyártja, mely már a K6-rendszereket is megbízható alkatrészekkel látta el. Ha azon felhasználók közé tartozunk, akik – minden eshetőségre felkészülve – sosem szorítják meg a csavarokat, akkor megnyugodhatunk: a gépet egyetlen csavar is teljesen egyben tartja. De vajon milyen gyors a gép? Mivel úgy gondoltam, hogy az átlagos felhasználó számára nem a teljesítményelemző programok eredménye, hanem a mindennap használt alkalmazások sebessége a mérvadó, ezért a GIMP-et, a xaost és más processzorzábaló programokat futtattam a masinán. A szomszédos asztalon egy jóval drágább, 800 MHz-es Athlonra épült gépen is ugyanezeket próbáltam ki, az összevetés végett. Lássunk csodát: a gépek sebessége megkülönböztethetetlen volt! Aki RSA-kulcsok feltörésére, komoly leképezési munkákra használná a gépet, az először olvassa el az adott programmal elért sebességi adatokat, melyeket a levelezési listákon, honlapokon lelhetünk fel, az átlagos feladatokat végző felhasználó viszont nyugodt szívvel megvásárolhatja a Duron-rendszert. A még nem GPL-es StarOffice 5.1 különösen gyorsan és gördülékenyen válaszolt. Szívesen látnék egy ilyen gépet az irodámban...

Bár csupán egy kis Duron és egyetlen IDE merevlemez található benne, a bemutatópéldányba épített hűtőrendszer jóval komolyabb alkatrészek működését is képes biztosítani. A ház nagy, körülbelül akkora, mint a VA Linux Systems által szállított kétutas, Xeon-alapú rendszeré. Az első és hátsó ventilátorok pedig hangosak, akár egy webkiszolgálóban. Mindez egy IDE merevlemez és egy „Socket A” processzor hűtésére?

Nguyen szerint a helyzet nemsokára változni fog, ugyanis az ASL következő Duron-alapú sorozatánál ugyanezt az alaplapot egy Micro ATX házba szerelik majd. Ezt az ElanVital „Whisper Technology” nevű zajcsökkentő rendszerével is ellátják. Az előzetes bejelentés alapján a Whisper-rendszer a zajt 37-ről 30 dB-re képes csökkenteni. A Whisper egy



processzorhűtőből és -ventilátorból, egy hajlékony csőből (ez szállítja a külső levegőt a processzor ventilátorához), valamint egy hőszabályozós ventilátorral ellátott tápegység-ből áll. Jó, hogy akad olyan cég, amely az emberi tényezőt is figyelembe veszi, hiszen minden gépnek csendesnek kellene lennie, amivel emberek dolgoznak. Mondom, minden olyan gépnek csendesnek kell lennie, amivel emberek dolgoznak. A zajos ventilátorok kora szerencsére letűnőben van.

Az ASL előre telepített programjaival gondjaim voltak. Először is, a rendszermag nem volt felkészítve a DHCP támogatására. Ez elég meglepő egy asztali gép esetében. Aki azonban most rendel ASL gépet, annak már nem kell szembesülnie ezzel a nehézséggel, mert felhívtam rá a cég figyelmét. *John Kim*, az ASL mérnöke ígéretet tett, hogy az ezután szállított gépek DHCP támogatással is bírnak majd. A hiba kijavítása gyorsan megtörtént, hiszen a Linuxot bárki nagyon egyszerűen beállíthatja.

A második bibi: amikor a Xi Graphics cég Accelerated-X nevű X rendszerét használtam (ezt az ASL előre telepíti), akkor a StarOffice betűtípusai kicsit összezavarodtak. A gépre ekkor telepítettem a csomagban szintén megtalálható Mandrake-változatot, s XFree86 alatt már semmi galiba nem jelentkezett. Kim szerint, az ASL ellenőrzi a gondomat, sőt, a Xi Graphics céggel is egyeztet, de eddig még nem kaptam választ tőlük. Szerencsére, ez nem jelentős gond. A gépet

ugyanis X kiszolgáló nélkül is megrendelhetjük, s ha már úgyis kizárólag ingyenes programokat telepítünk, akkor miért pont az X lenne az egyetlen, amelynek az egyik fizetős változatát használjuk? Az Accelerated-X telepítéskor a csomagban található gyári számot is kéri, s már csak emiatt sem éri meg az egész. Telepítsük csak szépen az XFree86-ot, sokkal jobban járunk, a program ráadásul teljesen ingyenes és könnyen beállítható. Még ha nem tervezzük is a rendszer testreszabását, a fizetős programok akkor is hátrányt jelentenek, hiszen ilyenkor a szerződéseket is figyelemmel kell kísérnünk. Az apróbb szépséghibák ellenére a gépet szinte mindenkinek ajánlani tudom; főleg olyanoknak, akiknek egy egész irodára valót kell felügyelniük. A Duron egy jó processzor, a többi alkatrész is első osztályú, az ASL-nek pedig nem kevés üzleti és szakmai tapasztalata van a linuxos asztali rendszerek területén. Az ASL Duron gépeket saját igényeink vagy cégünk elvárásai szerinti ingyenes, előre telepített programkészlettel rendelhetjük meg, de ne felejtjük el lemondani az X kiszolgálót. Az Accelerated-X fizetős program – mit keres egy ilyen egy nyílt operációs rendszerben?



Don Marti
a Linux Journal szakmai szerkesztője, levelet a dmarti@linuxjournal.com címre írhatunk neki.

Egyetlen világ, egyetlen processzor?

Az Intel processzorok fokozódó uralomra törekvése kiszorítja a RISC-rendszereket a kiszolgálók piacáról.

A Linux egyik hatalmas előnye, hogy gyakorlatilag bármely fellelhető processzoron fut. A vetélytárs Unix-változatok túlnyomó része egyetlen CPU-típushoz kötődik. Csak hogy a kiszolgálók világában egyre kevesebb processzor közül választhatunk. Bár ez különösebben nem veszélyezteti a Linuxot, biztosan nem lesz hasznára sem. A változások mögött az Intel áll: a cég a kiszolgálók piacát szeretné a PC-s piac másává tenni, ahol gyártók százai egymáshoz nagymértékben hasonlító rendszereket árulnak – természetesen az Intel processzoraira építkezve. Egy-egy évben ugyan sokkal több PC-t adnak el, mint kiszolgálót, a két piacról származó jövedelem azonban hasonló nagyságú, mivel a kiszolgálók egyrészt jóval drágábbak, másrészt áruk nagyobb hasznot tartalmaz. Ha az Intel meg tudja szerezni szokásos részesedését ebből a haszonból is (azaz a túlnyomó részét), akkor ez ismét lényeges előrelépést jelenthet számára.

Az Intel nemrég kezdte meg nagyszabású tervének megvalósítását, melynek előzményei 1998-ra nyúlnak vissza, amikor két területre is súlyos csapatot mért azzal, hogy piacra dobta első Xeon processzorát – melyet kimondottan kiszolgálóhoz terveztek – és felfedte a felsőosztályba sorolható kiszolgálóhoz készülő Itanium processzorának terveit. A Sun kivételével minden jelentősebb kiszolgálógyártó cég hamarosan elfogadta a kétirányú tervet, és egyaránt kínált Intel-alapú rendszereket és saját készítésű RISC processzorral készült gépeket a vásárlóknak. A gond az, hogy a fogyasztók jelentős része az Intel-alapú rendszereket választotta, ezek ugyanis általános operációs rendszereket használnak (Windows 2000-t vagy Linuxot), és számos gyártó alkatrészeivel zökkenőmentesen együttműködnek. A RISC-rendszerek egyre drágábbak, és az egyes cégek fejlesztésében készülő Unix-rendszerekre épülnek. Az Intel az alacsonyabb árú (tízezer dollár alatt) termékek körébe tartozó kiszolgálók szinte teljes piacát uralja, és a drágább rendszerek piacának is több mint a felét megszerezte. A piaci eltolódás miatt a RISC-rendszerek eladói kevesebb bevételt tudtak megszerezni saját processzoraik támogatására. Emellett azonban a csúcsteljesítményű mikroprocesszorok tervezése a korábbiaknál is összetettebb feladat

lett, az új termékek fejlesztése pedig egyre nagyobb beruházást igényel. A két ellentétes folyamat következményeképpen a Compaq, a HP és az SGI annyira visszafogták RISC-rendszereik fejlesztését (Alpha, PA-RISC és MIPS szerkezetek), hogy azok teljesítménye



számos fontos alkalmazásban a Xeon processzorra épülő rendszerek teljesítményét nem érik el. A fokozatos hanyatlás a RISC-gyártók eladásait mélybe rántó örvényt is előidézte. Az eredmény az, hogy a HP és az SGI már bejelentették, hogy nem folytatják RISC termékvonaluk fejlesztését, és valószínűleg a Compaq is hasonló lépésre fogja elszánni magát. Az IBM továbbra is befektet PowerPC vonalába, de a cég kiszolgálóinak túlnyomó része Intel processzorokon alapul. A legutóbbi Microprocessor Forumon – a CPU-ipar legfontosabb rendezvényén – az IBM volt az egyetlen az említett négy cég közül, mely bemutatta jövőbeni RISC-processzorának terveit. A Sun azzal kerülte el a szakadék peremét, hogy kizárólag saját SPARC/Solaris felületére összpontosított, így a cég sosem volt jobb helyzetben, mint most. Jelenleg a tízezer dollár körüli áron kínált kiszolgálók piacának húsz százalékát magának mondhatja, ennek ellenére gondjai vannak az új processzorok fejlesztésével. A cég nemrég mutatta be UltraSparc-3 processzorát, de az eredeti időponthoz képest közel két évet késétt piacra dobásával, és máris 18–24 hónapos késétt jelentettek be az UltraSparc-4 és UltraSparc-5 típusok tervezetének tekintetében is. Az Intel azzal az elgondolással lehet egyeduralkodóvá a piacon, hogy ugyanazokat a processzorterveket használja mind a munkaál-

lomás, mind a kiszolgáló termékvonalaiban. Így nagyságrendekkel többet fordíthat processzorok tervezésére, mint bármely más cég. Ahogy a CPU-k tervezése egyre összetettebb feladattá válik, úgy ennek a beruházási szintnek a követése szükséges a többi cég számára is a versenyképesség megtartásához.

Annak ellenére, hogy az Intel uralja a processzorpiacot, maga is két különböző termékvonallal oszta fel: a Xeon és az Itanium processzorcsaládra. Ez utóbbi eddig üres buboréknak bizonyult, ám számos csúszást követően az eljövendő hónapok során az új processzor végre valóban megjelenhet a rendszerekben. Egyes források szerint azonban teljesítménye gyenge lesz, és a legtöbb gyártó megmarad a Xeon-alapú rendszereknél. Az Intel új Pentium IV-ese 2001 elején jelenik meg a Xeon vonalon, további lökést adva fejlődésének. Az Itaniummal kapcsolatos remények tehát annak következő nemzedékéhez fűződhetnek, mely valószínűleg csak 2002 során jelenik meg. Addig viszont az Itanium csupán szellemjártékos marad.

Az AMD is feljebb kapcsolt, szeretné bejutni Athlon processzorát a kiszolgálók piacára. Ha az AMD sikerrel jár, némi nyomás nehezedhet az Intelre, ez a Xeon árának csökkentésére és teljesítményének növelésére sarkallhatja, ahogy az idén már megtörtént, hiszen a cégnek válaszulnia kellett az Athlon megjelenésére is a PC-s piacon. Mivel mind az Athlon, mind a Xeon ugyanazt az x86 utasításkészletet használja, az AMD belépése megerősítheti a géptípust, és még a jelenleginél is sikeresebb teheti. A következő néhány évben lényegében a Xeon és az őt helyettesíteni képes processzorok adják majd a kiszolgálók túlnyomó részének alapját. Ez a folyamat a Microsoftnak kedvez – mely kizárólag erre a géptípusra összpontosít –, viszont a Linux értékesítési lehetőségeit csökkenti. A Linuxnak a szolgáltatások és teljesítmény terén szemtől szemben kell megküzdenie a Microsofttal, ha folytatni szeretné terjeszkedését a kiszolgálók piacán.



Linley Gwennap
(linleyg@linleygroup.com)
a The Linley Group – szakmai elemző cég alapítója és vezetője. A kaliforniai Mt. Viewban él.

Fejlesztés linuxos környezetben

Javaslatok és ötletek a Linux használatára „Linux-mentes” fejlesztőcsoportoknak.

Meglehetősen viharos változásokat hozott a Linux az utóbbi másfél évben a kiszolgálóoldali operációs rendszerek világában. Ennek tudható be, hogy manapság egyre többen látnak jövőt olyan rendszerekben is, melyek nem tartoznak azon néhány vállalatírási érdekeltségi körébe, melyek hagyományosan uralkodnak a nagyteljesítményű operációs rendszerek piacán. A Linux fejlődésének következő állomása a kilépés a háttérzolgáltatások világából – ahol a rendszer már bizonyított – az asztali rendszerek színpadára. Ez a lépés azonban nem is olyan egyszerű, hiszen a Linux tervezésekor alapvetően a kiszolgálók kívánalmait tartották szem előtt. Egészen a legutóbbi időig nem is történt komoly kísérlet arra, hogy olyan alkalmazásokat fejlesszenek ki a végfelhasználó számára, melyek versenybe szállhatnak a jelenlegi piacvezetőkkel. Jóllehet sokan úgy tartják, hogy a Linux még nem elég felkészült az asztali rendszerek megvalósítására, nincs már messze az idő, amikor a váltás bekövetkezhet. Írásomban megkísérlem felvázolni azokat a gondokat, melyekkel a Linux-felhasználóknak szembe kell nézniük, amikor a rendszer „megszelidítését” tervezik. Hogy ne csak a gondokról és kihívásokról halljunk, igyekszem bemutatni, miként lehet néhányukon felülkerekedni.

A téma esetében azért idősezerű, mivel jelenleg egy kis irodai rendszeren dolgozom, mely eszményi feltételeket teremtett arra, hogy kísérletet tegyek a Linux ilyen célú felhasználására. Nem kétlem, hogy ez a felállítás sokak számára nem megfelelő, azonban hogy ne bonyolódjak unalmas részletekbe, megkísérlem a megszokottabb feladatok megoldásának módjait bemutatni. A projekt valójában egy javas fejlesztés, a központi fejlesztőkörnyezet pedig egy Solaris-rendszer. A számomra kitűzött feladatok: a projekt megtervezése, a kód fejlesztése és ellenőrzése, a hibák felkutatása, valamint a napi ügykezelés – így a levelezés, a kutatási eredmények és a leírások olvasása és készítése – voltak.

Tervezőeszközök

A tervezés során jobbra az UML-t használtuk modellező nyelvként. Ez a nyelv sokféle jelölésmódot tartalmaz az objektumorientált programtervezés különböző feladattípusainak megjelenítésére. Míg a Windows környezet számos folyamatábra-készítő eszközt kínál, a Linux hasonló eszközei kevésbé ismertek, nehezebben beszerezhetőek, és talán nem olyan kiforrottak, mint Microsoft-világbeli társaik. Mivel a fejlesztőkörnyezet alapvetően Java volt, így megfelelő program keresésében nem kellett kizárólag a Linux világra korlátozódnom, hiszen a Java-programok természetüknél fogva rendszerfüggetlenek. Végeredményben, kalandozásaim során két olyan programot találtam, melyek megfeleltek igényeimnek.

Egyikük, az ArgoUML – melynek webhelyét a Kapcsolódó címek között is megtaláljuk – szerzői szerint „gondolkodó” tervezőeszköz. Ez voltaképpen annyit jelent, hogy terveinket áttekintve kísérletet tesz a hiányosságok és a következtetések felderítésére. A webhelyen elérhető programváltozat a fejlesztés egy köztes állapotában van – úgy látszik, hogy a szerzők nemrégiben átköltöztek egy másik webhelyre, és ott újult erővel kezdtek a fejlesztésbe.

A Windows alatt a Rational Rose UML eszköz érdemel kitüntetett figyelmet, ezért, ha képesek vagyunk a Rose által is használható projekt-fájlok írására és olvasására, szerencsésnek mondhatjuk magunkat.

A MagicDrawUML egy kereskedelmi UML tervezőeszköz, mely Javában íródott, így teljesen rendszerfüggetlen, és képes projektjeinket Rational Rose formátumban elmenteni, melyeket így megoszthatunk másokkal is. Mellékesen megjegyzem, akik a Java lassúságára panaszkodnak, itt egy szavuk sem lehet, hiszen a szerzők kitétek magukért e téren is. A felhasználói felület épp olyan gyorsan válaszol, mintha Windows- vagy Linux-rendszert használnánk. Az egyetlen hibája, hogy pénzbe kerül, bár az összeg töredéke a Rational Rose árának, így egy kereskedelmi fejlesztés esetében ekkora befektetés mindenképpen megéri.

Fejlesztőeszközök

E szakaszban megkísérlem felvázolni, milyen segédeszközök léteznek Linux alatt Java-projektek fejlesztéséhez.

A fordítókörnyezetet a make program szolgáltatta, mely általánosan elterjedt a Linux-rendszereken, emellett a felületek sokaságán elérhető, így jól szolgálhatja rendszerfüggetlen fejlesztési céljainkat. A legtöbb rendszerfüggetlen makró és feladatot szabványos fordításvezérlő fájlokban (Makefile) határozhatjuk meg, így a legkisebbre csökkenthetjük a különböző rendszerek használatával kapcsolatban fellépő akadályok mennyiségét.

Mindenekelőtt hozzuk létre az ARCH környezeti változót, értékéül pedig az uname parancs visszatérési értékét adjuk. Ehhez (ha bash héjat használunk) egyszerűen írjuk be a következő parancsot a .bash_profile fájlba:

```
export ARCH='uname '
```

Ezután helyezzük el rendszerfüggő Make eljárásainkat és meghatározásainkat egy fordításvezérlő fájlban, melynek adjuk a Makefile.\$ARCH nevet, ahol a \$ARCH az uname visszatérési értéke az adott rendszerben. Végezetül, a fő vezérlőfájlban helyezzük el a következő sort, ez lehetővé teszi, hogy a make parancs végrehajtásánál rendszerfüggő értékeink is érvényre juthassanak:

```
include Makefile.$(ARCH)
```

Ha mindent jól végeztünk, a helyes beállítások betöltése futásidőben megtörténik.

A fordítókörnyezet kialakításának következő lépése egy Linux alatt futó Java-fordító felkutatása. Jelenleg számos Java-fordító és -értelmező projekt ismert, azonban a munkánkhoz sikerrel használható Java fejlesztőkörnyezetek közül leginkább csak a Blackdown és az IBM termékei jöhetnek számításba. A helyzettől függően tehát ezek közül kellett választanunk. Mindkét típus lépést tart a Sun legfrissebb fejlesztéseivel. A Java használatának másik nagy előnye a fordítás után nyert bajtkód rendszerfüggetlensége. Ez azt jelenti, hogy ha további osztálykönyvtárakat, vagy jar fájlokat szeretnénk használni, nem kell a rendszerfüggő megvalósítások után keresgélni. Emellett számos Javaalkalmazás tartalmaz rendszerfüggő könyvtárakat, ezek használata így az adott rendszerre korlátozódik. A velük kapcsolatosan felmerülő nehézségek foka azonban nagymértékben függ alkalmazásuk gyakoriságától. Saját projektem például a JMS-t (Java Message Specification) használja,



melyet a Sun fejlesztett ki, a Java Message Queue (JMQ) pedig ennek megvalósításáért felel. A JMQ-nak jelenleg még nem létezik linuxos változata, következésképpen nem ellenőrizhettem vele a kódot a Linux alatt. Ehelyett újra kellett fordítanom a kódot, majd (mivel a könyvtárak többnyire jar formátumúak) utána át tudtam vinni a kész anyagot a gépemre, és segítségével összeépíteni a programot.

A kód fejlesztésében segítő alkalmazásokkal Dunát lehetne rekeszteni. Mindegyik szerkesztőnek és felületnek megvan a maga szószólója, így ez ügyben csak félve mer nyilatkozni az ember – mindazonáltal, erőt véve magamon, bemutatam a kedvencemet. Munkaköri feladataim során néhány éve rákényszerültem a vi használatának megtanulására. Miután ezzel végeztem, gyorsan áttértem a vim és a gvim alkalmazására, és soha többé nem használtam a vi-t. Ezután még sok más fejlesztőkörnyezetet kipróbáltam, köztük a JBuildert, sőt, a Microsoft Developer Studioját is, azonban amíg nem használhatom bennük a vi billentyűkombinációit, feléjük se nézek. Nos, való igaz, hogy ezek az összetett fejlesztőkörnyezetek a hibakeresés terén többet tudnak a jó öreg vi-nál, azonban egy-egy megfelelő helyen elhelyezett println() utasítás csodákra képes.

A vim és gvim képes színes szintaktikai kiemelésre, a behúzások automatikus létrehozására, és még számtalan más hasznos dologra. Sőt, megfelelő beállításokkal a programozás-fordítás-hibakeresés ciklus végrehajtására is használható. A vi család leghatékonyabb képessége pedig a makrók alkalmazása, melyek egyszerű ismétlődő feladataink elvégzésével sok időt takaríthatnak meg.

A következőkben még két projektkezelési feladatról kell szót ejtenünk, melyek általában teendőink végére maradnak, ezek a forráskódkezelés és a hibák nyomon követése. A legelterjedtebb forráskód-kezelési programcsomagok az SCCS, az RCS és a CVS. Az SCCS nem érhető el nyílt forrású alkalmazásként, így foglalkozunk a másik kettővel. Az RCS a kisebb projektek igényeinek felel meg, a CVS-sel való munkára pedig a közelmúltban nyílt lehetőségem. Ez utóbbi jobbra ugyanazt a fájltípust használja, mint az RCS, felhasználói felülete azonban kidolgozottabb, emellett pedig képes több fejlesztő és távoli ügyfél kiszolgálására is. Ez a lehetőség különösen jól jött saját projektben, CVS-tárolónk ugyanis egy Solaris gépen foglalt helyet. Hogy megmutassam, mi mindenre képes a Linux, egy CVS-ügyfelet telepítettem linuxos gépemre (ez jó eséllyel hamarosan szélesebb körben is elfogadottá válik). A CVSROOT környezeti változó beállításával elértem a távoli Solaris gép CVS tárolóját, így a forráskódkezelést helyben megoldhattam. A CVS az rlogin parancs segítségével képes távoli parancsok futtatására, ezért hát győződjünk meg az elérés biztonságáról. A CVSROOT környezeti változó értékét a következő alakban kell megadnunk a távoli gép használatához:

```
export CVSROOT=:ext:hostname:CVSRepository
```

A jól használható nyílt forrású hibákat nyomon követő programok igen ritkák – mind közül csak kettőt találtam kellően megbízhatónak, és még itt is kénytelen voltam kiejteni az egyiket, mert nem ment át követelményeink szigorú rostáján. Egyetlen alkalmazás maradt tehát: a Mozilla csoport terméke, mely csinos webalapú felülettel, valamint MySQL adatbázisháttérrel bír. Beállításai minden részletre kiterjednek, és néhány apróbb buktató leküzdése után sikeresen és megfelelő sebességgel futott.

Segédprogramok

Projektünk fejlesztésénél hasznát vehetjük néhány közismert linuxos segédprogramnak is. Itt nemcsak a kód megírása és a tesztelés nélkülözhetetlen segédeszközeire gondolok, hanem azokra az egyéb programokra, melyek életünket általánosságban megkönnyíthetik. Így például, a find, a cat, az awk és az egrep parancsokat sokszor alkalmazhatjuk a rendszerfelügyelet, vagy a forrásfájlok közti keresés során, nemegyszer rövid parancsfájlok hatékony részeként. Ha pedig nem telepítünk a Cygwin for Windows-hoz hasonló programot, a Windows-felületen

dolgozva igencsak hiányát érezzük majd hasonló segédprogramoknak. Létezik a segédprogramoknak egy különálló csoportja, melyek használata erősebben kötődik a kód írásának folyamatához. Ezek segítenek a programlista megfelelő formátumának kialakításában, igazodva valamely szabványhoz – például a Sun Microsystems Java kódolási szabványához. Az Indent segédprogram – mely megtalálható a legtöbb Linux-rendszeren – számos beállítási lehetőségével segít kódunk végső alakjának kialakításában. Jelen pillanatban projektünk megfelel a Sun Microsystems kódolási szabványának. Amikor ennek kialakításán fáradoztam, belebotlottam a Jindent segédprogramba, mely Javában készült, így rendszerfüggetlenül működik. Alapértelmezésben a Sun szabvány formátumát készíti el, de saját beállítási fájl készítésével magunk is meghatározhatjuk a formázás módját.

Rendszerbeállítások

Az adatmentés egyszerűbbé tétele végett fejlesztésünk eredményének nagyobb részét Unix-kiszolgálókon tartjuk, ügyfélgépeink többsége pedig Windows NT. A Unix gépeken NFS és SAMBA működik, így a meghajtók megosztásával helyileg hozzáférhetünk olyan erőforrásokhoz, melyek eléréséhez egyébként be kellene jelentkeznünk valamely távoli gépen. A linuxos gépeken befűzhetünk mind NFS-, mind SAMBA-megosztásokat, ha pedig ugyanazt a könyvtárszerkezetet alkalmazzuk, mint a távoli gépen, akkor kevesebb esetben lesz szükség gépfüggetlen fordításvezérlő fájlok készítésére. Így például, az egyik felhasznált jar könyvtár a távoli gépen a /opt/FSUNjmq/lib/jms.jar néven érhető el. Ha a /opt/FSUNjmq könyvtárnak helyi gépünkön ugyanazt a nevet feleltetjük meg, a vezérlőfájlok ugyanúgy használhatóak lesznek a helyi gépen és a távoli Unix-rendszeren. A távoli meghajtók megosztását a következőhöz hasonló paranccsal végezhetjük el (itt az NFS esetét mutatjuk be):

```
share /opt/FSUNjmq
```

A SAMBA esetében másoljuk be a /etc/smb.conf (más rendszeren /etc/samba/smb.conf) fájlba valamely megosztásra adott példát, és a kapott új sorokban szereplő könyvtárakat cseréljük ki az általunk kívántakra (mindeközben ne felejtsünk el hozzáférési engedélyt adni a megfelelő felhasználónak). Néhány esetben előfordulhat, hogy mégiscsak be kell jelentkeznünk a távoli gépre, hogy különböző folyamatokat lefuttassunk – például, ha a távoli gép egy különleges adottságát használja a program, vagy ha nagy mennyiségű adatot kell másolnunk, mozgatnunk, vagy elérnünk. Mindezek mellett, egy távoli gépen bejelentkezve a hálózati működés is hatékonyabbá válik. Itt segít az rlogin, mely egyszerűbb elérést biztosít, mint a tlnet. Használatánál előre beállíthatjuk a hozzáférési engedélyeket, így nem kell minden belépéskor beírunk az azonosítónkat és jelszavunkat. Ehhez mindössze meg kell adnunk a távoli felhasználói könyvtár .rhosts fájljában a helyi gép nevét. Ilyenkor azonban ügyeljünk arra, hogy az írási/olvasási engedélyeket csak a tulajdonos számára adtuk meg, és csoportos vagy általános elérést nem tettünk lehetővé – másként ugyanis automatikus azonosításunk nem fog működni.

A távoli gépekre történő bejelentkezés során az egyik igen zavaró gond az volt, hogy még egyszer meg kellett adnunk a hű működésének beállításait. Ez azonban könnyen kiküszöbölhető, csak egy közös beállításfájl (profile) kell készíteni, mely egyaránt elérhető a helyi és a távoli gépek számára. Ehhez készítsünk távoli felhasználói könyvtárunkban egy .commonProfile nevű fájlt, és fűzzük be ezt a felhasználói könyvtárat a helyi rendszeren. Így a /home/username könyvtár mellett egy /remote/username nevűt is kaptunk. Ezután már csak annyit kell tennünk, hogy mind a helyi, mind a távoli beállításfájlnak a .commonProfile fájlra hivatkozunk, így ha bármit meg szeretnénk változtatni a két környezetben, csak e fájl tartalmát kell módosítanunk.

A Unix-ok távoli elérésével kapcsolatban fontos megjegyeznünk, hogy felhasználói azonosítóinkat a helyi gépen nem választhatjuk meg akár-hogy – az azonosító, az uid és a gid meg kell, hogy egyezzen azokkal, melyek a távoli gépen hozzánk tartoznak. Ez ugyanis igencsak leegyszerűsíti a dolgunkat az írás, a befűzés és a bejelentkezés engedélyeivel kapcsolatban. A tapasztaltabb felhasználóknak ismerős lehet a NIS fogalma – ez egy olyan módszer, melyben több unixos gép felhasználói azonosítóit és jelszavait egyetlen közös tárolóban helyezik el. Némi kutakodással és megfelelő hálózati beállításokkal magunk is elérhetjük, hogy linuxos gépünk NIS-t használjon más rendszerek felhasználóinak kezelésére. Ezzel feleslegessé válik a felhasználói és csoportazonosítók többszörös használata az új felhasználók beiktatásánál.

Hasonlóképpen nagyszerű lehetőség az X Window ügyfelek futtatása a távoli gépről helyi rendszerünkön, hiszen így távoli elérésünkhöz grafikus felületet is kapunk. Mindazonáltal – jóllehet futtathatunk távoli alkalmazásokat X Window megjelenítőnk ablakaiban – egyes esetekben felmerülhetnek ellentétek a távoli alkalmazás elvárásai és a helyi X Window kiszolgáló lehetőségei között. Klasszikus példaként említhetjük azt az esetet, amikor az X kiszolgáló (a helyi gépen) 16 bites színeket használ, a távoli alkalmazás azonban csak 8 bites színek használatára képes. Ilyenkor nem kell feltétlenül leállítanunk X Window munkafolyamatunkat, és újraindítanunk 8 bites színekkel. Egyszerűbb megoldás lehet még egy X Window munkafolyamatot indítani 8 bites színekkel, vagy a távoli ablakkezelő megjelenítőjét alkalmazni a helyi gépen. Erre példa a következő parancs:

```
X :1 -query remotehost
```

A „:1” azt jelenti, hogy ezt a példányt második megjelenítőként szeretnénk futtatni. Ehhez szükségünk van arra is, hogy a távoli gép DISPLAY változóját szintén a második megjelenítő állítsuk:

```
export DISPLAY=yourhost:1
```

Jó tanácsok a mindennapokra

Zárszóként megemlítem, hogy a Windows mellett a Linux segítségével is elvégezhetőek mindazok a feladatok, melyek nap mint nap végigkísérik munkánkat.

A talán a legáltalánosabban előforduló feladat a Microsoft-fájlok írása és olvasása, legyen itt szó a Word, az Excel, vagy a PowerPoint fájljairól. A StarOffice nagyszerű programsomagot kínál mindezek kezelésére, legfrissebb változata pedig további Microsoft programtípusokat is szárnyai alá vett.

Az elektronikus levelezéshez ügyfelek gazdag választékából csemegezhethetünk. Természetesen a választék gazdagságát meghatározza az is, milyen levelezőkiszolgálót használunk. Ha POP vagy MAPI levelezésről van szó, semmi gondunk nem akadhat, hiszen az előbb említett StarOffice, valamint a Netscape is rendelkezik e protokollokat támogató beépített levelezőprogrammal.

A gondok azonban ott kezdődnek, amikor a levelezési szolgáltatónk a Microsoft Exchange-et használja, és nem teszi lehetővé az MAPI vagy a POP elérést. Jelenleg nem létezik MAPI-megfelelő ügyfél a Linuxhoz, így jelen ismereteim szerint csak olyan program jöhet szóba, mely a Windows alatt fut (ha bárkinek tudomása van más megoldásról, kérem, jelezze!). Mindazonáltal ilyenkor is létezik megoldás, erről a következőkben még szót ejtünk.

Sajnálatos módon a Linux jelenleg még nem rendelkezik azokkal a lehetőségekkel, melyekre napjaink irodáiban szükség van, ezért számos esetben a Microsoft termékeinek használatára szorulunk. A WINE lehetővé teszi, hogy Microsoft-alkalmazásokat futtassunk a Linux alatt, a VMWare pedig egy másik operációs rendszer példányát képes futtatni a Linuxon. Ezt mindenképpen érdemes kipróbálni, ha még nem tettük meg! Számomra a VMWare lehetőséget teremtett, hogy a Microsoft Outlook

Kapcsolódó címek

VMWare: <http://www.vmware.com>

WINE: <http://www.winehq.org/>

MagicDraw: <http://www.nomagic.com/>

ArgoUML: <http://argouml.tigris.org/>

Segítség a NIS beüzemeléséhez:

<http://www.suse.de/kukuk/nis-howto/HOWTO/NIS-HOWTO.html>

StarOffice:

<http://www.sun.com/products-n-solutions/software/prodapps/>

Linux alkalmazások kincsestárai:

<http://freshmeat.net/> és

<http://slashdot.org/>

Hasznos linuxos adatok:

<http://linux.com/>,

<http://www.altavista.com>,

<http://www.deja.com/usenet>,

<http://slashdot.org/>

Hasznos Java-adatok:

<http://java.sun.com/>

Bugzilla: <http://www.mozilla.org/bugs/source.html>

CVS: <http://www.cyclic.com/CVS/>

VIM: <http://www.vim.org/>



használatával elolvashassam az Exchange kiszolgálón tárolt leveleimet. Nos, ez az ágyúval verébre módszer jellegzetes esete, azonban mit tegyünk – valahogy csak el kell olvasni a leveleket!

A nyomtatás beállításai sok fejfájást okozhatnak a Linux-rendszer beüzemelése során, de ne aggódjunk, hiszen a nyomtatótámogatás és a meghajtók rohamléptekkel fejlődnek. Az alapvető szabályok szem előtt tartásával a linuxos hálózati nyomtatók beállítása nem okozhat különösebb nehézséget. Még akkor sem kell azonnal kétségbeesni, ha éppen nem találjuk a nyomtatóhoz használható meghajtót, hiszen a világhálón számos leírást találunk, melyek egy-egy nyomtatótípus működtetését írják le a megfelelő meghajtó hiányában. Így például, irodánkban működik egy Gestner PCL nyomtató – saját meghajtó hiányában a HP LaserJet III meghajtójával. Összefoglalva a leírtakat elmondhatjuk, hogy sok mindent megtehetünk a Linux-rendszerrel, még ha nem is kapunk közvetlen támogatást. Megpróbáltam érzékeltetni, milyen akadályokkal kerültem szembe munkám során, és hogyan oldottam meg őket. A Kapcsolódó címeknél néhány webhelyet találhatunk, melyek hasznos segítséget jelentettek, amikor arra került a sor, hogy valamilyen Windowsban szokásos feladatot Linux alatt oldjak meg. Mindazonáltal, fontos megjegyezni, hogy a Linux egyelőre még nem mindenható – néha vissza kell térnünk a nem linuxos alkalmazásokhoz.



Mark Stacey jelenleg a dublini székhelyű ICL vállalatnál dolgozik, érdeklődésének középpontjában a Linux és a Java áll. Munkaidején kívül szívesen látogat el messzi országokba. Kérdésekkel a mark.stacey@e-merge.ie címen kereshetjük fel.

A MOD formátum (1. rész)

Írásunkban a No Starch Press kiadónál megjelent Linux Music & Sound című könyv egyik fejezetét bővítettük és gyarapítottuk.

Sokféle modulformátum létezik (.mod, .it, .s3m/stm, .xm, .med, .669). Az eredeti .mod formátumot a világ első olcsó multimédiás számítógépén, az Amigán futó ProTracker használta, ez volt a legelső „tracker” (modulszerkesztő) programok egyike. A modulok kiterjesztéséből általában következtethetünk arra, hogy a fájl melyik zeneszerkesztőből származik: az .it az Impulse Tracker, az .s3m/stm a ScreamTracker, a .med az OctaMED formátuma, és még sorolhatnánk. A formátumok közötti különbségek a rendelkezésre álló sávok, hangszerek (hangminták) számában, valamint a hangminták minőségében (8 vagy 16 bit) jelentkeznek. A Linux-felhasználók szerencséjére a legnépszerűbb formátumokat (.mod, .xm és .it) a fellelhető modulszerkesztők és -lejátszók mindegyike támogatja. Megjegyeznénk, hogy a szerkesztők többsége csupán két-három formátumban képes olvasni és menteni, a lejátszókkal azonban jóval több formátumot használhatunk. A legtöbb Linux-változatban megtalálható MikMod lejátszó például legalább tizenöt modulformátumot ismer, a kiváló XMMS nevű lejátszóprogram MODPlug bővítménye pedig húsznál is több modul típust képes beolvasni.

Mi az a modulszerkesztő?

Egy modulszerkesztő és néhány hangminta segítségével bárki ropant gyorsan készíthet zenét. A szerkesztő felépítése nagymértékben hasonlít a MIDI sávrögzítők panelekből álló szerkezetére. A panelek legfontosabb tulajdonsága a hossz: ez a benne található lépések számát jelenti. Minden lépésnél meghatározhatjuk a hangminta hangmagasságát, hangszerszámát, hangerejét és rengeteg más hatást (például a vibratót, szűrőket és a hang elhelyezkedését a sztereó térben). A paneleket az általunk megadott sorrendben lejátszva egy teljes dalt kapunk, amit a különböző modulformátumok valamelyikében menthetünk. Az első modulszerkesztők Amigára íródtak. E gépeken a hangkezelésért felelős alkatrész az alapra kapott helyet, és 8 bites, négy-csatornás (két bal és két jobb csatorna), mintavételezett hangok lejátszására volt képes. Az olcsó PC-s hangkártyák megjelenésével az MS-DOS rendszerek váltak a modulszerkesztők eszményi környezetévé, manapság pedig már Windows, Macintosh és természetesen Linux rendszerekhez is elérhetők ilyen programok. E szerkesztők különösen alkalmasak ritmusközpontú zenék (pop/rock, techno, dance stb.) készítésére, de mivel bármilyen hangmintát felhasználhatunk, a programokkal tetszőleges előadásmódban kísérletezhetünk. Látogassunk el például a MOD Archive, a MODPlug Central vagy a United Trackers honlapokra, ahol saját szemünkkel – és fülünkkel – is meggyőződhetünk a modulszerkesztőkkel készíthető zenék hihetetlen sokféleségéről.

A MIDI és a MOD

A modulszerkesztő a MIDI sorozatok szerkesztésére szolgáló programokra emlékeztet. A panel alapértelmezés szerint 64 lépésből áll, ennek játékeje külön beállítható, de a dalban bármikor módosítható, az ütemtől függően. Ezt BPM-ben adhatjuk meg (beats per minute, azaz ütemek száma percenként). Az egyes lépéseknél bármit tehetünk a hanggal: indíthatjuk egy tetszőleges hangminta lejátszását, értékek nélkül vagy azokkal. Ha egy néhány lépéssel fentebb elindí-

tott hang még mindig szól, a hang lejátszásának időtartama alatt a következő lépéseknél tetszőlegesen módosíthatjuk annak bármely tulajdonságát (hangerő, hangmagasság, kitartás, különböző hatások). Természetesen egy hangminta bármit tartalmazhat, egy koppanástól kezdve a teljes dobritmusokon át Wagner egész életművéig. A MIDI fájlok jóval rövidebbek a moduloknál, de ez csupán abból adódik, hogy az előbbiek nem tárolják magukat a hangmintákat, hanem a hangkártyák többé-kevésbé szabványos hangszerszerkezetét vagy hangkeltési képességeit (szintézis) használják ki. A modulok a panelek adatai mellett a hangmintákat is tartalmazzák, tehát ezért nagyobbak, mint a MIDI fájlok.

A General MIDI (GM) szabvány a hangkártyákban elhelyezett hangminták kiosztását egységesíti. A gyártók azonban igen változó minőségű hangszerszerkezetekkel látják el termékeiket: egyes kártyákban profi, másokban szinte használhatatlan a beépített szabványos hangszergyűjtemény. Mivel a modul a hangmintákat is tartalmazza, ezért teljesen mindegy, hogy milyen típusú kártyán játszunk le azokat: az eredmény ugyanaz lesz.

Lehet, hogy most sokan azt gondolják: minek van egyáltalán szüksége a MIDI-re? Nos, erre jó néhány indok említhető: a MIDI-szerkesztők általában jóval fejlettebb képességekkel bírnak, képesek külső eszközöket (MIDI-billentyűzet, szintetizátor, külső programok stb.) kezelni. Emellett a fájl méretből is származik előny, hiszen a csekély méretű MIDI fájlok még a kis sávzélességű vonalakon is gyorsan továbbíthatók. A MIDI Manufacturers Association nagy gondot fordít a szabvány egységesítésére, és állandóan bővíti a lehetőségeket. Ezzel szemben a modul szintér jóval kevésbé átlátható. Számos modulszerkesztő teljesen egyedi formátumot használ, így manapság már legalább húszféle modulformátum létezik. A modulok és egyes szerkesztők képességeit fejlesztik ugyan, de összefogó szervezet híján követhetetlen a fejlődés. Ha azonban kizárólag mintavételezett hangokból kívánjuk felépíteni zenénket, csak az általunk megálmodott hangszerekkel szeretnénk hallgatni a dalokat és beérjük az inkább „középsúlyúnak” nevezhető formátummal, akkor egy jó modulszerkesztő az, amire szükségünk van.

Kapcsolódó címek

No Starch Press

➔ <http://www.nostarch.com/>

Általában:

MODPlug Central – kiváló modulforrás:

➔ <http://www.modplug.com/>

United Trackers – minden a modulokról:

➔ <http://www.united-trackers.org/>

Programok:

MikMod – modullejátszó:

➔ <http://mikmod.darkorb.net/>

SoundTracker – a legjobb linuxos modulszerkesztő:

➔ <http://www.soundtracker.org/>

Debian-csomagkezelés: felhasználói leírás (1. rész)

A csomagkezelő rendszerek terén az összes Linux-változat közül a Debiané az egyik leghatékonyabb és legrugalmasabb megoldás. Ha egyszer rászánod magad és elkezded használni, egyre könnyebb dolgod lesz, ezt megígérhetem.

A Debian alapvető csomagkezelő eszköze a dpkg. A dpkg jelenleg a dpkg-deb-re épül, erre később még visszatérek. Több, a dpkg eszköze épülő program is létezik, így a dselect, apt-get, console-apt. Ez a cikk a hogyanról kíván szólni, azon utasításokról és parancsokról, amelyek segítségével a dpkg működtethető, illetve amelyekkel a megfelelő adatok megszerelhetők egy Debian vagy Debian-alapú rendszerről. Nem szándékozom kitérni mindenre, de ahhoz elegendőre, hogy a Debian-alapú rendszer csomagkezelő képességeit könnyedén kezelhessük. Minden Debian csomag egy tárállomány, amely „deb” kiterjesztéssel végződik. Ebben a cikkben a csomagokra „deb”-ként fogok hivatkozni. A debek általában a következőképpen vannak elnevezve:

```
zsh_3.1.6.pws21-1.deb
```

zsh a csomag neve, a _3.1.6 a csomag változatszáma és a pws21-1 a kiterjesztése.

Ebben a cikkben a „ezaz.deb” nevet fogom használni, amikor a debet kell behelyettesíteni, és „ezaz”-t amikor a csomag nevét kell behelyettesíteni.

```
zsh_3.1.6.pws21-1.deb
```

- zsh a csomag neve
- 3.1.6pws21 a program változatszáma
- 1 Debian csomagváltozat
- .deb a fájl kiterjesztése

dpkg: minden Debianok alapja

A legtöbb ember nemigen fog dpkg-t használni a mindennapos csomagkezelés során, de ez egy elképesztően hatékony és hasznos eszköz. A dpkg alapvető képessége a csomag telepítése, amit a dpkg -i ezaz.deb parancs segítségével végezhetünk el. Ez a parancs telepíti a csomagot, miközben elmenti a csomag már telepített változatát, amennyiben létezik ilyen. A dpkg -i -R /tmp/ezaz parancs a megadott könyvtárban lévő összes debet telepíti. Mindannyian tapasztalhattuk már, hogy néha egy-egy csomag nem telepítődik helyesen, vagy visszautasítja a beállítást. Akár a felhasználó is megszakíthatta a beállítási folyamatot. A dpkg tartalmaz néhány hasznos eszközt a beállítás megkönnyítésére. Ilyen esetekben a dpkg --configure <csomagnév> befejezi a megadott csomag beállítását, a dpkg --configure --pending pedig minden függőben lévő csomagot beállít. Ugyanilyen könnyű törölni is egy csomagot. A dpkg -r <csomagnév> vagy a dpkg --remove <csomagnév> eltávolítja a megadott csomagot, de meghagyja a beállítófájlokat. Ha az összes, csomaggal kapcsolatos fájlt le szeretnénk törölni, beleértve a beállítófájlokat is, akkor használjuk a dpkg -purge <csomagnév> parancsot.

Adatgyűjtés

A Debian csomagkezelő rendszerében több lehetőség is adódik annak lekérdezésére, hogy hány csomag van éppen telepítve, és az egyes csomagok milyen fájlokat tartalmaznak. A csomagok kiválogatásának egyik módja a minták használata, amit a dpkg -l <minta> parancs

valósít meg. Ha szükséges, helyettesítő karakter is használható a mintában. A dpkg -l parancs önmagában az összes, a rendszeren éppen telepített csomagot listázza.

A Debian csomagkezelő azt is lehetővé teszi, hogy megnézzük, milyen fájlokat telepített egy bizonyos csomag. Ezt a dpkg -L <csomagnév> parancs segítségével érhetjük el. Ennek kiderítésére, vajon melyik csomaghoz tartozik egy bizonyos fájl, használhatjuk a dpkg -S <fájlnév> parancsot. Ezeket a kereséseket mintákkal is végre lehet hajtani, beleértve a helyettesítő karaktereket is.

Ha egy bizonyos debről szóló adatra van szükségünk, hasznos lehet a következő pár parancs:

- dpkg -I ezaz.deb: részletes listát ír ki a debről,
- dpkg -c ezaz.deb: kilistázza a deb tartalmát, hasonlóan a tar -c kapcsolójához.
- dpkg -x ezaz.deb <könyvtár>: a debet a megadott könyvtárba csomagolja ki,
- dpkg -X ezaz.deb <dir>: kilistázza a fájlokat kicsomagolás közben, mint a tar -v kapcsolója.

Kényszerítsük!

Mindaz a rengeteg hasznos képesség, ami a dpkg-ban rejlik, nemcsak a hatékony eszközhasználatot teszi lehetővé, hanem azt is, hogy elbizonytalanodjanak a felhasználók. A dpkg leírásában ez áll: „Figyelem: ezek többnyire szakembereknek szánt lehetőségek. Hatásuk teljes megértése nélküli használatuk a teljes rendszert tönkretelheti.”

Ezeket a lehetőségeket itt most nem mutatom be, mivel igazából soha nem is lesz szükségünk rájuk. Ha valaha mégis rá szeretnénk venni a dpkg-t a függőségek vagy az ütközések figyelmen kívül hagyására, más csomagok által telepített fájlok felülírására, vagy bármi más olyan dologra, aminek megakadályozására a Debian csomagkezelőt megalkották, nézzünk bele a program leírásába.

APT

APT az „Advanced Package Tool” azaz a fejlett csomagkezelő eszköz rövidítése. Dicsőséges feladata a Debian-felhasználó életének egyszerűbbé tétele. Többnyire én is az APT-t használom. Hogy miért erőltetem a dpkg-t? Azért, mert fontos megérteni, mi is rejtezik az APT mögött. Az APT rendszere három fő részből áll: a beállítási fájlból, a sources.list listából, és két programból, melyek neve apt-get és apt-cache. Van még néhány kisebb része is, mint például az apt-cdrom.

sources.list

Ha egyszer megtanultad a használatát, az /etc/apt/sources.list lesz az egyik legfontosabb fájl a rendszereden. A sources.list legnagyobb-részt olyan ftp és http címeket tartalmaz, ahonnan az APT adatokat kérhet le. Ez kitűnő lehetőség arra, hogy a forrásaidat egyetlen helyre csoportosítsd. Egy jellemző bejegyzés valami ilyesmi lehet:

```
deb http://http.us.debian.org/debian unstable and
main contrib non-free\
```

A bejegyzés első része lehet „deb”, ha olyan sorról van szó, ami végrehajtható állományok helyét határozza meg, vagy „deb-src”, ami azt mondja meg, honnan szedjük a Debian forráskód-csomagokat. A következő részt URI-nak nevezik, ami nagyon hasonlít az URL-re. Ez a Debian könyvtárszerkezet gyökere. Az URI után következnek az a változat, amit általában „stable”, „frozen” vagy „unstable” néven adnak meg, de akár nevesített változat, mint például a Hamm, Slink vagy a Potato is használható. A változat ezenkívül megadható pontos elérési útként is, ilyenkor „/”-jellel kell végződnie, és nem lehet az alkotóelemeket meghatározni. Az alkotóelemek általában „main”, „contrib”, „non-free”, illetve „non-us/main”, „non-us/contrib” vagy „non-us/non-free” névre hallgatnak. Ez minden, amit tudnunk kell, hogy kihasználhassuk a sources.list előnyeit. A CD-ROM bejegyzéseket az apt-cdrom készíti (erről bővebben szöveg a későbbiekben), így ezeket nem kell kézzel beírni. Szükség lehet azonban a source.list fájl szerkesztésére is, ha meg akarjuk változtatni azt a helyet, amit az APT használ (hogy melyik tükörről töltsék az adatokat), esetleg egy új URI-t szeretnénk beilleszteni valamilyen „nem hivatalos” debekhez, vagy akár olyan helyekhez, amelyek újabb debeket tartanak fenn, mint például a KDE és a HelixCode. Ha átszerkesztjük ezt a fájlt, mindig le kell futtatni egy `apt-get update` parancsot. Hogy megtudjuk, melyik Debian tükörodalt célszerű használni, nézzünk el a <http://www.debian.org/mirror/> honlapra.

apt-cdrom

Az egyik lehetőség arra, hogy egy Debian CD-ROM-ot adjunk az APT adatbázisához, a megfelelően könnyen használható apt-cdrom. Az apt-cdrom add parancs automatikusan befűzi a CD-meghajtót (feltételezve, hogy a /dev/cdrom egy helyes hivatkozás a meghajtóra), megvizsgálja annak tartalmát, és létrehoz egy megfelelő bejegyzést a /etc/apt/sources.list fájlban. Mint azt a Linuxnál már megszokhattuk, nem ez az egyedüli út! Ha a Debian CD-ROM be van fűzve (a befűzési pontnak meg kell jelennie az /etc/fstab-ban), akkor a feladat megoldására használhatunk kapcsolókat is. A `-d <befűzési_pont>` kapcsolót a befűzési pont meghatározására használhatjuk, az `-f` kapcsoló segítségével pedig rávehetjük az apt-cdrom-ot, hogy ne ellenőrizze az egyes csomagokat, míg a `-a` kapcsolót használhatjuk a lemez összes könyvtárának módszeres felderítésére.

apt-get

A Debian legnagyobb büszkesége az apt-get, egy hihetetlenül értelmes (na jó, a legtöbb esetben értelmes), könnyen használható csomagkezelő eszköz, amely automatikusan kezeli a csomagok közötti függőségeket és ellentmondásokat. Az első tennivaló (és ezt mindig tegyük is meg, ahányszor csak módosítjuk az /etc/apt/sources.list állományt), hogy lefuttatunk egy `run apt-get update` parancsot, amely letölti az apt-get és apt-cache által használt csomagadatokat a megadott helyekről.

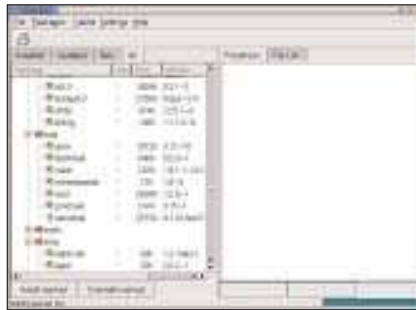
Nos, immár elkezdhetjük a telepítést. Az APT valószínűleg legtöbbször használt lehetősége a telepítés. Egy csomag telepítéséhez (és függőségeinek automatikus kielégítéséhez), használjuk az `apt-get install <csomagnév>` parancsot. Ha más csomagokat is telepíteni kell, az APT szólni fog mielőtt letölti őket. Amikor egy debet a feltételei teljesülése nélkül telepítünk, vagy valamilyen gond jelentkezik a telepítés vagy törlés



1. kép A csomagok állapotának vizsgálata



2. kép Gnome-apt: csomagkezelő



3. kép Kpackage

során, gyakran nem is tudunk tovább dolgozni a dpkg-val a gond megoldása nélkül. Az egyik lehetséges módszer az ilyen nehézségek elhárítására az `apt-get -f install` parancs, amely igyekszik mindent rendbe rakni, miközben folyamatosan kérdéseket tesz fel.

Az APT képes forrásokat is letölteni az `apt-get source <csomagnév>` parancsral, ha van legalább egy deb-src sor a /etc/apt/sources.list állományban. A csomagok eltávolításához használjuk az `apt-get remove <csomagnév>` parancsot. Ha le szeretnénk a törölni a tárolófájlokat, két lehetőségünk van: az összes tárolófájl törléséhez használjuk az `apt-get clean` parancsot, vagy, ha csak a régi tárolófájlokat szeretnénk törölni, akkor inkább az `apt-get autoclean` parancsot adjuk ki. Amikor eljön az ideje, hogy az összes csomagot a legújabb változatra frissítsük, egyetlen könnyű parancs – az `apt-get upgrade` – megoldja a gondodat. Az `apt-get dist-upgrade` parancs a teljes Linux-változat újratelepítése céljából készült, és automatikusan újraszervezi az összes megváltozott nevű csomaghoz tartozó függőségeket.

Van még pár parancssori lehetőség, ami hasznos lehet az apt-get használata során: a „-d” csak letölti a tárolófájlokat, de nem telepíti vagy csomagolja ki őket; a „-s” szimulációt jelent, a program valójában semmit nem tesz ilyenkor; az „-y” minden kérdésre automatikusan igennel válaszol; és végül a „-b” megkísérli letöltés után lefordítani a forráscsomagot.

Továbbá, az apt-get képes szabványos kifejezőmintákat értelmezni telepítéskor. Ilyenek például az `apt-get install '.name.'` vagy az `apt-get remove 'mozil.'` Az APT-gel adott ponton megjelenő elemeket is képes azonosítani a ka-

raktorsorozatban. Például, ha a „pilot” szó csak a szó elején fordulhat elő, mint a „pilot-manager” de úgy nem, mint a „gnome-pilot”, akkor használhatjuk a „^” jelet (ami tulajdonképpen a „sor kezdete” karakter) a kifejezésben, valahogy így: `apt-get install '^pilot.'`

Frissítés

Mostanáig még egyetlen más Linux-változatot sem találtam, ahol a frissebb változatra történő átállás olyan egyszerű lenne, mint a Debiannál. Amint azt korábban említettem, ha új változatra szeretnénk áttérni, egyszerűen változtassuk meg az URI megfelelő részét az /etc/apt/sources.list fájlban, aztán adjunk ki egy `apt-get dist-upgrade` parancsot. Gyakran hasznos az `apt-get dist-upgrade` többszöri futtatása, hogy minden simán legyen telepítve és beállítva. Ezt a módszert csak akkor használhatjuk, ha új változatra térünk át, stable-ről frozenre, stable-ről unstable-re vagy frozenről unstable-re. Visszafelé ez nem működik.

apt-cache

Az apt-cache kitűnő eszköz, ha a csomagokról szóló adat kereséséről van szó, még abban az esetben is, ha a csomag még nincs is telepítve! Az `apt-cache show <csomag>` különböző adatokat jelenít meg a csomagról, beleértve a függőségeket, a teljes nevet, azt, hogy mit tesz lehetővé, a hosszú és rövid leírást, illetve a legfontosabbat: a kicsomagolt méretet. Az `apt-cache depends <csomag>` parancs listát ad arról, hogy mely más csomagok telepítésére van szükség a <csomag> telepítéséhez és helyes működéséhez. A teljes csomaglista kinyomtatásához használjuk az `apt-cache pkgnames` parancsot.

Az apt-cache általam leghasznosabbnak talált szolgáltatása talán az apt-cache search <karaktorsorozat>. Ez a szolgáltatás végigkeresi az összes csomagnevet és leírást az általunk megadott <karaktorsorozat>-ért. Nyilvánvaló, hogy ez igen nagy időmegtakarítást jelenthet. A rendszerkarbantartás további segítésére van még néhány kiaknázható lehetőség az apt-cache-ben. Például az -i csak a fontos függőségeket listázza ki, az -f teljes rekordokat ír ki (ahogyan a „show”) miután végignézte és – csak névre korlátozottan kereste a csomagneveket.

Fülek

Bár a legtöbb esetben a dpkg-t használom csomagkezelésre, az apt-cache eszközt a keresésre és az apt-get programot a telepítésre, néha valami kényelmesebbet szeretnék. Ez az, ahol a dselect, a console-apt és a gnome-apt belép a képbe.

dselect

A dselect a Debian kezelőfelületek nagypapája. Mint a telepítő folyamat első része, ez az első dolog, amivel az új Debian-felhasználók szembekeverülnek, s amitől aztán halálra is rémülnek. És bizony a dselect pont olyan rémisztő, mint milyen nehéz felfogású. Lássunk hát egy rövid bemutatót a dselectről. Először is, melegen ajánlom, hogy a dselect használatakor minden képernyőre kerülő segítséget olvassunk el (ez, sajnos angolul van). Habár nem túl felhasználóbarát, de mindent tartalmaz, amit csak tudni kell. Amikor első ízben indítjuk el a dselectet, be kell lépni az „[A]ccess” (elérés) részbe, hogy kiválasszuk a megfelelő elérési módot. Azaz, válasszuk az „nfs”-t, ha van a közelben egy Debian-tükör egy nfs-megosztáson, és válasszuk az „apt”-ot, ha az Interneten keresztül (http/ftp) telepítünk, vagy ha már beállítottuk a sources.list fájlt a helyi címekre. Az egyéb módszerek: CD, Multi-CD (ha a változat egy helyett több CD-ROM-ot is tartalmaz), Floppy és Mounted (befűzött).

Ha az elérési módot már kiválasztottuk, a dselectnek meg kell mutatni, milyen csomagokhoz férhet hozzá; ehhez ki kell választani az „[U]pdate” (frissítés) menüpontot, és várni egy kicsit.

A „[S]elect” (kiválasztás) az a rész, ahol a tulajdonképpeni csomagkezelés történik. Először is egy sűgőlapot látunk. Olvassuk el! A szóközzel léphetünk ki a sűgőből, illetve a PONT billentyűvel nézhetjük meg a billentyűparancsokat. A csomaglistában a kereséssel vagy a kurzorvezérlő gombokkal mozoghatunk. Lássunk néhány hasznos dselect gyorsbillentyű parancsot:

```
/ keresés
\ az utolsó keresés ismétlése
? sűgő
d lefelé görgeti a csomagadatot
u felfelé görgeti ugyanazt
```

Miután kijelöltünk egy csomagot, a következőket tehetjük:

```
+ telepítés vagy frissítés
- törlés
= jelenlegi állapotban hagy
```

Kapcsolódó címek

Magyar tüköroldalak:

- ➔ ftp.kfki.hu/pub/linux/debian/
- ➔ ftp.hu.debian.org/debian/
- ➔ debian.inf.elte.hu/debian/
- ➔ linuxvilag.hu/debian

A gnome-apt honlapja:

- ➔ <http://www.debian.org/~hp/gnome-apt.html/>

Kpackage:

- ➔ <http://www.general.uwa.edu.au/u/toivo/kpackage/>



Ha készen vagyunk, a kiválasztómenüt a következő módon hagyhatjuk el:

```
ENTER elfogadás, kilépés és függőség-ellenőrzés,
Q kilépés, elfogadás és függőségek figyelmen kívül hagyása,
X vagy Esc változtatások visszavonása.
```

Lehet, hogy az Enter leütése után egy csomaglistával találjuk szemben magunkat, ezek a csomagok vagy függőségi viszonyban vagy elmentésben vannak azokkal a csomagokkal, amit kiválasztottál. Nézd végig ezeket a csomagokat; a hiba leírása a képernyő alsó felében jelenik meg. A gondok megoldása után üssünk ENTER-t. Miután a változtatásokat elvégeztük a „[S]elect” részben, válasszuk az „[I]nstall” (telepítés) vagy a „[R]emove” (eltávolítás) menüpontokat, hogy a rendszerben a változtatások életbe lépjenek. A „[C]onfig” (beállítás) menüpont csak akkor szükséges, ha valamelyik csomagbeállítás sikertelen volt.

gnome-apt

A gnome-apt a Debian GUI (grafikus felhasználói felületű) csomagkezelője, amely a Gnome alá íródott, rendkívül leleményes és igen csak hasznos. A Gnome-apt kitűnően megszerkesztett felhasználói felületen keresztül jeleníti meg a csomagméreket, a függőségeket és majdnem minden egyéb adatot.

A csomagok rendezett faszervezetben jelennek meg (lásd a 2. képen), valamint a gnome-apt meg tudja jeleníteni őket ábécésorrendbe, csoport, állapot vagy fontossági sorrend szerint rendezve is. Ezenkívül egyszerű, de hatékony keresési szolgáltatással is rendelkezik. A Gnome-apt része a Debian 2.2-nek (Potato) és a jelenlegi unstable változatnak (Woody).

Kpackage

A KPackage a KDE csomagkezelő felülete az RPM-hez és a debek-hez. A listás és a fastruktúras megjelenítés keverékét használja (lásd a 3. képet), és feladatában hasonló a gnome-apt-hez. A KPackage egyik kedvező tulajdonsága, hogy minden függőség hiperhivatkozás a szóban forgó csomagra. A Kpackage az összes, már telepített csomagban található fájl is listázza, és ellenőrzi, hogy valóban léteznek-e.

console-apt

A console-apt egy új felület az APT-hez. Jelenleg csak a Debian unstable változatában található meg.

Mivel azonban még fejlesztés alatt áll, nem is szeretnék túl sok szót vesztegetni rá. A Console-apt-nak van néhány hasznos tulajdonsága, érthetőbb és a csomagok keresésének, rendezésének és szűrésének képességével is rendelkezik.

Végszó

Ezeket a lehetőségeket és képességeket használva, már magunk is könnyen kezelhetjük és karbantarthatjuk Debiansomagjainkat. Sokkal bonyolultabban hangzik, mint amilyen valójában!

Megpróbáltam annyi módszert leírni, amennyit csak lehetővé tesz a Debian, de néhányat ki kellett hagynom. Sajnos, nemigen sikerült eddig az APT-n kívül más hasonlót találnom. Őszintén szólva, még egyetlen RPM felületet sem láttam, amit egy lapon tudnék említeni az APT-vel.



David Blackman hallgatói rendszergazda a New York-i Stuyvesant főiskolán. Mostanában a Helix Code Gnome-ba szerelmes. Élvezi a C pointer-aritmetikáját és a Perl ördögeit.

Debian 2.2 Potato: egy számítógépbetyár emlékére

Egy olyan szervezetről, mely amellett, hogy önkéntesekre támaszkodik a saját Linux-változata kidolgozásában, és folyamatosan azon munkálkodik, hogy jobbá tegye azt, sok mindent elárul az, hogy elismeri mindazok közreműködését is, akik a gyarapításán dolgoznak.

A Debian a 2.2 hivatalos változatát Joel „Espy” Klecker emlékének szentelte, így fejezve ki elismerését a közreműködéséért. Joel idén júliusban halt meg, személyében a Debian az egyik legünnepelebb programfejlesztőjét veszítette el. Az írott ajánlás megtekinthető a <http://www.debian.org/doc/dedication-2.2.txt> címen.

Maga a 2.2-es változat is ugyanilyen mérvű nagylelkűséget mutat, a kezdő felhasználókkal szemben. A Debian hírneve távolabb állt a kezdő felhasználóktól, mint a legtöbb változaté. A telepítéstől a használaton át a finomhangolásig: a Debian 2.2 egy olyan változat, mely a kevésbé türelmesek számára is szórakoztató.

Előkészítés és telepítés

A gép jellemzői:

- processzor: 500 MHz K6/2,
- memória: 64 MB RAM,
- merevlemez: 9,2 G Quantum Fireball,
- 3-Com ADSL modem,
- hangkártya: SoundBlaster Live!,
- videokártya: Diamond Viper 770 Video,
- hálózati kártya: DLink 530-TX NIC,
- rendszermag: 2.4.0test8.

A Debian telepítéséhez szükséges indító- (boot), rendszer- (root) és meghajtólemezekhez minden leírás és útmutatás elérhető a

<http://www.debian.org/releases/2.2/i386/install> oldalon. Itt jó tanácsokat is kapunk az indítólemezekhez.

A legfontosabb talán, hogy jó minőségű lemezeket használjunk.

A Debian hálózati telepítése is egyszerűbb és gyorsabb vált. Nem mindenkinek van erős internetkapcsolata, de ha a szerencsések közé tartozunk, fél óra alatt végig lehet zavarni egy telepítést. Valószínűleg ez a Debian legjelentősebb változása: a könnyű letöltés és az FTP-n keresztül történő telepítés.



<http://www.debian.org>



<http://mf.linux.rulez.org/mailman/listinfo/linux-kezdok>

A felhasználó választhat az egyszerű és a részletes telepítés között: előbbi a néhány leggyakrabban használt csomag gyors telepítését végzi, és különféle segédprogramokat tartalmaz (pl.: C++ Fejlesztőkörnyezet, Gnome játékok), utóbbi pedig lehetővé teszi a tapasztaltabb Debian-felhasználók számára, hogy egyesével válasszák ki a telepíteni kívánt csomagokat. A dselect végzi a függőségi ellenőrzést, és egyszerűsíti további csomagok visszaellenőrzését és beállítását.

Beállítás

Általában a dselect gondoskodik a legtöbb csomag beállításáról, lehetővé téve a felhasználó számára mind a .deb csomagok beállításainak megőrzését, mind egy működő rendszer beállítását. FTP-letöltés esetén, a megfelelő hálózati beállítás a telepítés megkezdése előtt történik. Néhány „nem ingyenes” csomag telepítéskor szükség lehet további programok letöltésére – lásd a Real Playert, vagy az IBM JDK-ját –, és a beállítás csak azután történhet meg. Kicsit aprólékos munka, de megéri a fáradságot. Továbbá akad még néhány olyan csomag, melyek terjesztése csak forráscsomagként engedélyezett. Az ilyen csomagokat nekünk kell lefordítanunk, ehhez segítséget kapunk *readme* állományokban (pl. Pine). A lefordított anyagokat .deb csomagokká alakíthatjuk, amelyeket már tudja telepíteni a dpkg.

Mint a legtöbb programozó, én is kicsit lusta vagyok, és nem szeretek olyan programokat írni, amit mások már elkészítettek. A Fetchmailhez csatolt fájlok között például van néhány, melyek segítségével könnyen áthidalhatjuk a telepítéssel és üzembe helyezésével járó legtöbb gondot. Ezek a parancsfájlok elérhetők a `</usr/doc/fetchmail/contrib>` könyvtárban és hasznosak a kezdő rendszergazdák, vagy a levélküldő szolgáltatások használatában járatlanok számára.



⇒ <http://www.telnet.hu/hamster/linux/>

Érvek, ellenérvek

A Debian híres megbízhatósága volt valószínűleg a fő érv a Storm, a Libranet vagy a Corel számára. Mindhárom cég jelentős bevételre tett szert saját változata kiadásával, melyek elsősorban egy könnyen használható, csillogó-villogó telepítővel lettek kiegészítve. A CD-ről telepítő felhasználók tipikus hibába futhatnak bele. Ez a hiba a rosszul összeállított lemez. Mivel bárki összeállíthat saját lemezt, előfordul, hogy hibásan összeállított lemezeket kapunk. Szerencsére a hivatalos kiadásokban rendkívül ritka az ilyen hiba. A hálózati telepítés nagymértékben javult, mind a letöltési sebesség, mind a biztonság tekintetében.

Az XFree86 4.x sorozatát nem tartalmazza ez a változat, de a programfejlesztők azon dolgoznak, hogy a következő kiadásban már benne legyen, vagy legalábbis engem így tájékoztattak. Jelentőségteljes fejlődés lenne, mivel a jelenlegi 3.3.6x változat legjobb esetben is a kiszámíthatatlan jelzővel illethető. Az Xserverek nem működnek (de legalábbis nem megbízhatóan). Még mindig vannak gondok a GLX-szel bizonyos kártyáknál. Mindenesetre a történethez tartozik, hogy ez nem a Debianra jellemző hiba. (Aki most tervezi az új videokártya beszerzését, lapozzon a 76. oldalra, ahol összefoglaló cikket talál a témáról.)

A Debian 2.2 széles csomagkínálata (matematikai és tudományos programok, játékok, szerkesztők, grafikus felületek stb.) vetekszik a SuSE kínálatával, de sok veszélyt is tartalmaz. A választási lehetőségek, a bonyolult függőségi kapcsolatok és a sok csomagütközés mind-mind óvatosságra int. Némelyik csomagot ideje lett volna frissíteni, például a rengeteg programkönyvtár, melyek leg többje csupán a visszírnyú együttműködés biztosítása miatt került fel a

A Debian 2.2 Potato jellemzői:

Gyártó: Debian/GNU Linux
E-mail: leader@debian.org
URL: <http://www.debian.org>

A rendszer dobozos terméként is beszerezhető, a Linuxvilág magazin első és második számának CD-mellettén megtalálható, illetve ingyen letölthető (például az ftp.linuxvilag.hu-ról).

lemezre. A Debian nem sok kész megoldást kínál cégek számára, de számos eszköz és játék van a programfejlesztők, a grafikusok, az egyetemisták és a szórakozni vágyók számára.

A Debian nem titkolja, hogy célrétege nem tartalmazza a teljesen kezdő linuxosokat. Ennek ellenére a levelezési listákon mindig kapunk segítséget.

Frissíthető?

A Debian egyetlen gyengéje egy könnyen használható telepítő hiánya volt. Az új telepítő, bár nem grafikus felületen fut, de lényegesen átláthatóbb, hatékonyabb lett. Röviden összefoglalva, tapasztalt Linux-felhasználók számára egyszerű a telepítés, az üzemeltetés és a frissítés is.



⇒ <http://debian.dedasz.hu/lama/>

Várható irányok

A cikk megjelenésekor már két hivatalos frissítéscsomag is megjelent. Ne feledjük, a Linux nagy erőssége, hogy a felmerülő hibákat gyorsan kijavítják. Volt egy hiba például az egyik rendszerindító lemezben is. Egyébként tervezik, hogy a telepítéshez használható hajlékonylemezeket eltávolítják a hivatalos anyagból.

A fejlesztők sem unatkoznak, már elérhető a fejlesztői (nem megbízható) változat is, kifejezetten azok számára, akik szeretnének bekapcsolódni a munkába (⇒ [ftp://ftp.debian.org/debian/dists/woody](http://ftp.debian.org/debian/dists/woody)). Az új IPv6 szabvány elterjedésével együtt a Debian fejlesztői is sok energiát áldoznak arra, hogy a rendszer megfeleljen az új követelményeknek.

Összefoglalás

A Debian fejlesztői hihetetlen mértékű munkát fejtettek ki a 2.1-es változat óta, az új változat az én tetszésemet teljesen elnyerte.

A Debian 2.2 hamarosan nagyon sokak számára válik komoly választási lehetőséggé. Olyanokra gondolok, akik nem teljesen analfabéták a Linuxhoz, és szeretnének tanulni egy keveset, hogy sokat hozzanak ki a rendszerükből.

Joel Klecker büszke lenne rá.

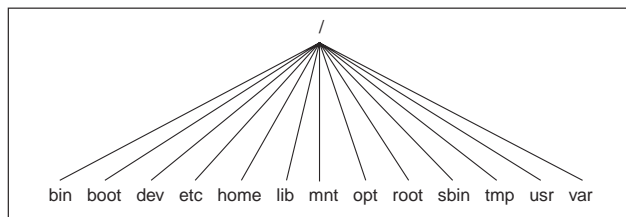
Stephanie Black (alphafemale@radiantnet) új szereplő az IT területén. A Coastal Den Computing nevű linuxos tanácsadó céget irányítja. Pályafutásának nyolcvan százalékát a Linuxszal töltötte.

Hová telepítsük saját fejlesztésű Linux-programjainkat?

Részletes ismertetés az alkalmazások megfelelő helyéről.

Néhány programfejlesztő és független kereskedő tanácstalan abban a kérdésben, hogy hová kell a saját fejlesztésű programokat telepíteni. Néhányan a /usr/bin vagy a /usr/local/bin, mások a /opt könyvtár mellett döntöttek. A választást általában a szerző unixos háttere befolyásolja, ez többnyire a System V, a Berkeley BSD vagy a GNU/Linux lehet.

A fájlrendszerek felépítésének szabványa (Filesystem Hierarchy Standard 1, FHS) 2.1-es változata éppen ezen különbségek megelőzésére született, és részletes iránymutatást ad a rendszerszolgáltatások, beállítások és programok telepítési helyéről a Unix vagy Unix-alapú rendszerekben. Az FHS részletesen ismerteti a főbb könyvtárak tartalmát és feladatát is (1. ábra).



1. ábra Az FHS főkönyvtárszerkezete

Dióhéjban: az operációs rendszer működéséhez szükséges programok helye a /sbin, a /bin vagy a /usr könyvtár. A rendszergazda forráskódból is telepíthet programokat a /usr/local/bin könyvtárba.

A csak bináris változatban elérhető, a rendszerhez nem szervesen tartozó programok gyökere a /opt/csomagnév könyvtár; azaz a bináris fájlok a /opt/csomagnév/bin-be, az esetleges sűgőfájlok pedig a /opt/csomagnév/man könyvtárba kerülnek.

A /opt könyvtárat több nemzetközi szabvány (az AT&T által 1990-ben előterjesztett System V Application Binary Interface, az Intel Binary Compatibility Standard, a Common Operationg System Environment és a Linux Standard Base) és a unixos közösség is elfogadta a kiegészítő programcsomagok helyéül.

A /opt terület számára a rendszergazdának külön lemeزرészt (partíciót) kell létrehoznia, a felhasználók PATH környezeti változójában pedig szerepelniük kell a /opt/csomagnév/bin és /opt/bin útvonalaknak.

A felhasználói héj általában a /opt/csomagnév/bin könyvtárban találja meg a bináris fájlokat, a rendszergazda azonban a /opt/bin könyvtárban elhelyezett hivatkozásokkal vagy parancsállományokkal megkönnyítheti a programok indítását.

A /opt futtatható fájljainak a gépre vonatkozó beállításait a /etc/opt/csomagnév könyvtárakban kell tárolnunk. Ez a /opt csomagok beállításfájljainak helye, hiszen a Unix-alapú rendszerekben minden beállításfájl a /etc könyvtárban található.

A rendszer futása közben változó fájlok helye a /var/opt/csomagnév. A /var tartalma általában az adott gépre jellemző, és a könyvtár mindig külön lemeزرészen található, a fő fájlrendszer véletlen teleírásának megelőzése céljából.

Ezen szabályoktól indokolt természetesen esetben eltérhetünk: például

ha egy programnak új eszköze van szüksége, akkor azt a /dev könyvtárban kell elhelyezni.

Az FHS kiegészítő programcsomagokra vonatkozó ajánlásaink ismeretében készítsünk el és telepítsünk egy képzeletbeli, Whizbang nevű programcsomagot. A Linux Standard Base szabvány értelmében az RPM Package Managert kell használnunk, és a csomagot a /opt könyvtárba telepítésre kell előkészítenünk. Ez a whizbang-1.2-3.spec beállításfájl 18–20. soraiból is kiderül (1. lista). A 8. sor mutatja be, hogy miként tehetjük a csomagot áthelyezhetővé, majd. A nem szabványos könyvtárba telepítést azonban nem ajánljuk.

A Whizbang RPM csomagját az alábbiak szerint építhetjük fel.

A whizbang-1.2-3.spec fájl alapján az RPM elkészíti a whizbang-1.2-3.src.rpm nevű forráscsomagot és a whizbang-1.2-3.i386.rpm nevű bináris csomagot.

Edward Bailey Maximum RPM című könyve, illetve az RPM honlapja <http://www.rpm.org/> kitűnő példákkal szolgál az RPM használatában elmélyedni kívánó felhasználók számára. A részletekkel egyelőre ne foglalkozunk, csak tartsuk szem előtt a javasolt telepítési útvonalakat. A whizbang-1.2-3.i386 elkészítéséhez tegyük a következőket:

```
# rpm -ba /usr/src/redhat/SPECS/whizbang-1.2-
# 3.spec
Processing files: whizbang
Finding provides..
Finding requires..
Prereqs: /bin/sh
Wrote: /usr/src/redhat/SRPMs/whizbang-1.2-\
3.src.rpm
Wrote: /usr/src/redhat/RPMS/i386/whizbang-1.2-\
3.i386.rpm
```

Az épp most elkészített whizbang-1.2-3.i386.rpm csomagot az alábbiak szerint telepíthetjük:

```
# rpm -i /usr/src/redhat/RPMS/i386/whizbang-1.2-
# 3.i386.rpm
```

Miután a whizbang programot az /opt/whiz/bin könyvtárba telepítettük, kísérreljük meg parancssorból futtatni. Megtalálta a héj? Szerepelt a /opt/whiz/bin a PATH környezeti változónkban? Mi a teendő, ha az egésznet még kényelmesebbé kívánjuk tenni azzal, hogy a /opt/bin/whizbang hivatkozással a /opt/whiz/bin/whizbang programra mutattunk? Ezt az RPM telepítés után végezhetjük el, a következő módon:

```
%post
P=$RPM_INSTALL_PREFIX
mkdir $P/bin > /dev/null 2>&1
ln -fs $P/whiz/bin/whizbang $P/bin/whizbang
mkdir $P/man/man1 > /dev/null 2>&1
```

1. lista Minta RPM beállítás fájl

```

1. #
2. # RPM Package Manager (RPM) File for "Whizbang"
   # RPM csomagkezelő (RPM) fájl a Whizbang
   # programhoz
3. #
4. Summary: Some neat Whiz Bang program
5. Name: whizbang
6. Version: 1.2
7. Release: 3
8. Prefix: /opt
9. Copyright: Commercial
10. Group: Application/Productivity
11. URL: http://www.ibm.com/linux/
12. Vendor: International Business Machines
13. Packager: George Kraft IV gk4@us.ibm.com

14. %description
15. This is a demonstration of a relocatable RPM
16. package for a fictional productivity application.

17. %files
18. /opt/whiz/bin/whizbang
19. /opt/whiz/man/whizbang.1
20. /etc/opt/whiz/whiz.conf

21. # Post-install stuff would go here.
   # Utólagos telepítési és beállítási adatok
22. # EOF (VAF=vége a fájlnek)

```

```

ln -fs $P/whiz/man/whizbang.1
$P/man/man1/whizbang.1
# EOF

```

Ezt az „áthelyezhető”, telepítés utáni parancssorozatot az 1. lista 20. sora után helyezhetjük el, de a hivatkozások eltávolítására szolgáló %postun eljárást mindenki készítse el magának (gyakorlásként is jó lesz). Néha előfordul, hogy egy csomagot nem az eredetileg tervezett helyre, hanem máshová kell telepítenünk. Ekkor távolítsuk el a whizbang-1.2-3 RPM csomagot és telepítsük egy másik helyre:

```

# rpm -e whizbang-1.2-3
# rpm -i --prefix /usr/local /usr/src/redhat
# /RPMs/i386/whizbang-1.2-3.i386.rpm

```

Összegzésképpen: a csak bináris formában hozzáférhető, nem a programcsomaghoz tartozó programokat telepítsük a /opt/csomagnév/bin könyvtárba. Áttekintettük az áthelyezhető RPM csomagok készítésének és felépítésének módját, majd a módszer rugalmasságát is bizonyítottuk azzal, hogy az eredetileg a /opt könyvtárba telepítésre tervezett csomagot más helyre telepítettük. Az FHS szabvány követése az első lépés afelé, hogy alkalmazásink jobban illeszkedjenek az LSB3 előírásaihoz.



George Kraft IV, más néven „GK4”, az IBM Linux Technology Center nevű részlegénél a Linux Standards Base kidolgozásával foglalkozik. George 1982 óta dolgozik BSD operációs rendszeren. A GNU/Linux rendszert 1993 óta használja.

Védelem a hirdetésekkel szemben

A DoubleClick.net egy kifejezetten szalaghirdetésekre szakosodott webhely. Ha tudatjuk névkiszolgálónkkal, hogy ő a birtokosa a DoubleClick.net tartománynak, akkor arra a kérdésre, hogy „Hol található a doubleclick.net?” magabiztosan a következő választ adja: „Birtokomban van minden adat, ami csak elérhető a doubleclick.net tartományról, és állíthatom, hogy nem létezik ilyen hely!”. Ha pedig a böngészők nem képesek megtalálni a tartományt, az sem képes felkutatni a látogatókat. Mivel a felhasználók rendszerint névkiszolgálókat használnak, így ez a módszer nem csak a leggyorsabb hirdetésselenes módszer, de jelentős szerepe lehet a felhasználók védelmében is. Jelentkezünk be rendszergazdaként, majd tegyük a következőket:

1. Keressük meg a named.conf fájlt (jobbára a /etc, vagy a /etc/bind könyvtárban lelhetjük meg). Ha nem sikerül ráakadnunk, használjuk a következő parancsot:

```
find / -name named.conf
```

2. Nyissuk meg a fájlt kedvenc szerkesztőnkben. Keressük meg a „localhost” területet, mely valahogy így fest:

```
find / -name named.conf
```

3. Nem számít, ha a „file” kezdetű sorban található fájlnev eltér a szerkesztőben láthatótól.

```
zone "localhost" {
    type master;
    file "/etc/bind/db.local";
};
```

Készítsünk másolatot a „localhost” területről valahol máshol a fájl területén, majd ebben a másolatban a „localhost” kifejezést írjuk át „doubleclick.net”-re. Az eredmény valahogy így fest majd:

```
zone "doubleclick.net" {
    type master;
    file "/etc/bind/db.local";
};
```

Mentsük el a fájlt, és lépünk ki a szerkesztőből. Ha valamit összekevertünk volna, lépünk ki mentés nélkül, és végezzük el újra a 3. lépést.

4. Indítsuk újra a névkiszolgálót. Erre két lehetőségünk van. Ha létezik a /etc/init.d/bind fájl, adjuk ki a következő parancsot:

```
/etc/init.d/bind restart,
```

ha nem, akkor keressük meg a következő paranccsal a named folyamatazonosítóját:

```
ps ax | grep named
```

Tegyük fel, hogy a következő eredményhez jutunk:

```
7907 ?          S          0:03 /usr/sbin/named
```

Ekkor így járhatunk el:

```
kill -HUP 7907
```

Természetesen a parancsban azt a folyamatazonosítót kell használnunk, amit magunk kapunk. Készen is vagyunk. Úrítsuk ki böngészőnk gyorsítárát és örvendezzünk!

Don Marti

A Linux testreszabása az alapoktól

Ha nem találunk kedvünkre való Linux-változatot, készítsük el sajátunkat!

Sokféle Linux-változattal találkozhattunk már az eddigiekben, és talán még több lesz. Némelyik apró, mint a DLX a maga egyetlen hajlékonylemeznyi méretével, némelyek pedig kimondott óriások, mint a Red Hat 6.2, mely öt CD-t is megtölt. Úgy tűnik, minden egyre bonyolultabbá válik, ahogy a rendszerek mérete nő. Egyre kevésbé érthető, hogyan képes a rendszer innen-onnan származó kódok darabjaiból összeállni, és egyre kisebb reményt látunk arra, hogy képesek legyünk saját rendszerünk összeállítására és testreszabására bizonyos célok szem előtt tartásával.

A dolog szerencsére nem ilyen kilátástalan, hiszen az alaprendszer szemszögéből nézve minden változat hasonlóan áll össze. Az egyetlen különbség, hogy a nagyok több programcsomaggal és kiegészítővel szélesebb közönség megnyerésére törekednek, míg a kicsik kevesebb eszközzel szűkebb felhasználói kört céloznak meg.

A széles körű képességekkel bíró Linux-változatok többnyire szükségtelenül nagyok az egyedi alkalmazások igényeihez. Vegyük például a beagyazott rendszereket, ezek nem támasztanak túl nagy igényt egy-egy feladat megvalósításán túl a Linux-rendszerrel szemben. Mindemellett a lehetőségek és az igények olyannyira különbözőek lehetnek, hogy a gyártók képtelenek olyan változatot készíteni, mely kellően átfogó, egyszersmind a legtöbb felhasználó elvárásainak megfelelő. Az alkalmazások a hatékony működéshez rendszerint teste szabott alaprendszert kívánnak. Ez egyszerűen megoldható, csak fizetnünk kell érte valamely változat gyártójának. Van azonban más megoldás is – végezzük el a testreszabást magunk! Ez a „csináld magad” módszer nem csupán nagyszerű multság, de sok esetben létfontosságú megoldást nyújt, így egyre többen értékelik a Linux megbecsülendő lehetőségeiként. Ha itt tapasztalatokat nyerünk, nemcsak a rendszermag testreszabását tanulhatjuk meg, hanem rendszerünk többi összetevőjét is képesek leszünk beállítani az igényeinknek leginkább megfelelő teljesítmény eléréséhez.

Írásomban szeretném megmutatni, hogy saját Linux-alaprendszerünk elkészítése nem lehetetlen feladat. Megosztom e téren szerzett tapasztalataimat. Alaprendszeréről van szó, mely tehát kicsiny, átlátható és működőképes. Megkíséreljük egyszerűvé tenni mindazt, ami bonyolultnak látszik. Lépésenként haladunk a felépítésben, a testreszabással pedig elérjük, hogy a rendszer akár egy hajlékonylemezen is elférjen. Ha mindennek a végére jutottunk, a kapott összeállítás jó kiindulási pont lehet az egyes alkalmazások igényeinek megfelelő alaprendszer kiépítésében.

A Linux-rendszer

A rendszerek általában több, egymással kapcsolatban álló részből tevődnek össze. A Linux esetében ezek két fő területre csoportosíthatók: a rendszermag, valamint a többi összetevő, amelyek nélkül a rendszermag semmire sem volna jó. A rendszermagon kívül minden programelem fájlrendszerben kap helyet. Így a gyakorlat szempontjából a Linux-rendszert felfoghatjuk úgy, mint a rendszermag és a fájlrendszer egységét – és ez alól egyik Linux-változat sem kivétel. Így például egy teljes egészében telepített Linux-változat egyszerűen egy rendszermagból és egy hatalmas fájlrendszerből áll. A Linux telepítőlemezeivel a teljes rendszert telepíthetjük, míg egy vészrendszer az esetleges hibák bekövetkeztekor lehet életmentő segítség. Mindkettő hasonló módon épül fel, vagyis rend-

Rendszermag

Fájlrendszer

1. ábra A Linux-rendszer

szermagot és egy kezdeti fájlrendszert tartalmaznak – ez utóbbi azonban kicsiny, így csak néhány feladat elvégzéséhez szükséges eszközöket tartalmaz (lásd az 1. ábrát). A Linux rendszermag alkalmas a fájlrendszer felkutatására és üzembe helyezésére, legyen az hagyományosan egy lemezrészre telepítve vagy egy tömörített lemezlenyomatba összepréselve.

Mi az alaprendszer?

Tegyük fel, hogy van egy alkalmazásunk, melyet egy osztott könyvtárakat használó rendszerrel szeretnénk futtatni. Tételezzük fel továbbá, hogy ez az alkalmazás nem igényel semmiféle különleges lehetőséget, mellyel a rendszer jelenleg nem rendelkezik. Egy olyan alaprendszert szeretnénk létrehozni, amelyen futtathatjuk ezt az alkalmazást, emellett működését némiképpen szabályozhatjuk is.

A 2. ábra egy jellegzetes alaprendszert ábrázol – meg kell azonban jegyeznünk, hogy a kép nem teljes, hiszen egy program lehet statikusan fordított is, melynek így nincs szüksége az osztott könyvtárakra, továbbá lehet a.out típusú is, ami viszont azt jelenti, hogy nincs szüksége a dinamikus betöltőre.

Ha az alkalmazás „önálló”, vagyis minden futás közben szükséges segédeszközzel statikusan lett fordítva, akkor nincs szüksége az osztott könyvtárakra, így közvetlenül a rendszermag föléthathat – az alaprendszer tehát ilyen esetben magát a rendszermagot jelenti. Mindazonáltal szinte minden rendszerben szükség van bizonyos segédeszközök támogatására, legyen szó akár a fájlműveletekről, akár a rendszerfigyelésről (gondolhatunk itt a mount, illetve a ps parancsokra). Alaprendszerünk felépítésébe tehát beleértjük a rendszermagot, a dinamikus betöltőt, valamint néhány elengedhetetlenül szükséges könyvtárat és segédeszközt.

Célunk egy olyan alaprendszer kiépítése, mely elfér egy hajlékonylemezen, és tartalmazza a rendszermagot, valamint egy tömörített kezdeti fájlrendszert (lásd a 3. ábrát). Ezt a tömörített fájlrendszert a rendszermag kicsomagolja, majd elhelyezi a memóriában. Az ilyen és hasonló alaprendszerek készítése gyakran nem nehéz feladat, azonban sokak számára túlságosan hosszadalmas. Nos, ezen fogunk változtatni a következőkben.

A fent vázolt alaprendszer testreszabása nagymértékben függ attól az alkalmazástól, amelynek rendszerünket alá szeretnénk rendelni. Válasszuk ki a rendszermag megfelelő beállításait, egy dinamikus betöltőt, az alkalmazás és a segédprogramok működéséhez szükséges könyvtárakat, valamint olyan alapvető segédeszközöket, melyek a rendszer szabályozásához és fenntartásához szükségesek. Ezután fordítsuk le mindezeket egy megfelelő környezetben, csomagoljuk be az eredményt, majd tegyük az egészet indíthatóvá. Ha valami még hiányzik, kezdjük újra az egészet, belevéve az előzőekben kimaradt összetevőt is.

Szerény célkitűzéseink

A kíváncsi olvasó megkérdezheti, miért tesszük mindezt, és vajon mire jó ez az egész? Nos, másokhoz hasonlóan mi is szeretnénk összetett alkalmazásokat futtatni, mondjuk többszörös többfeladatos programokat egy jellemzően PC-szerű gépen, amelyben azonban nincsen merevlemez, vagy esetleg nem kapcsolódik hozzá képernyő. Szükségünk van tehát egy rendszermagra, és néhány egyéb elemre, amelyekkel elkezdhetünk kísérletezni. Az így kapott rendszertől három fontos dolgot követelünk meg: legyen megbízható, fenntartható és testre szabható. Sokunk számára egy saját rendszermag létrehozása e célra némiképp túlméretezett feladatnak tűnhet. Itt lép be a képbe a Linux, és a nyílt forrás társadalma – együtt megkímélnék a terhes feladattól.

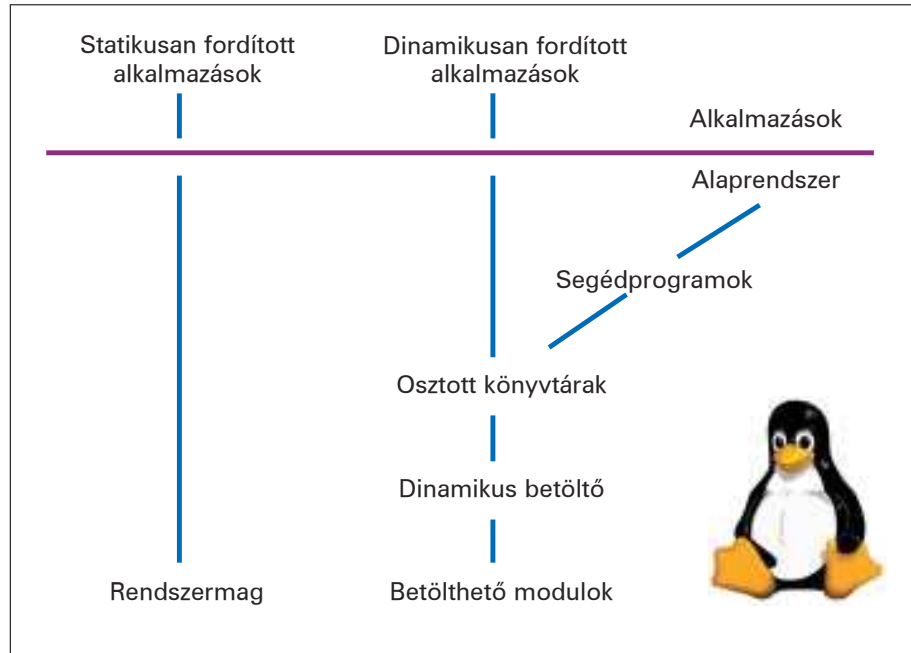
Alapanyagaink tehát készen állnak, ráadásul ingyenesen beszerezhetők. Feladatunk csak annyi, hogy kiválasszuk a megfelelő építőköveket és felépítsük belőlük saját rendszerünket.

Mielőtt azonban mindennek nekikezdenénk, néhány alapvető kérdésre választ kell kapnunk: Hogyan fordítsuk le a rendszermagot? Vagy egy osztott könyvtárat? Miként készíthetjük el a kezdeti fájlrendszert? Hogyan helyezzük el a rendszermagot és a tömörített fájlrendszert egy hajlékonylemezen vagy EPROM-ban? Miként futtathatjuk az osztott könyvtárakat használó alkalmazásokat? Hogyan végezzük a hibakeresést? Számos ilyen és hasonló kérdést tehetünk még fel. A válaszok pedig, ha nem is egy helyen, de fellelhetők. Ami most következik, az saját tapasztalataimon alapul, és a sorok között, ha rejtve is, választ találunk sok fenti kérdésre.

A kivitelezés lépései

Ha a testreszabás tervei előttünk állnak, megkezdhetjük a kivitelezés lépéseit. Kezdjük az általános tennivalókkal:

1. A fejlesztőkörnyezet üzembe helyezése.
Fejlesztőkörnyezetként telepítsünk egy teljes Linux-változatot, mint például a Red Hat 6.0. Győződjünk meg arról, hogy a gcc támogatja az adott rendszert. A dolgok egyszerűsítése végett tegyük fel, hogy célgépünk (melyen majd a testre szabott rendszer működik), valamint gazdagépünk (melyen a fejlesztést végezzük) ugyanolyan típusú processzorral működik, vagyis esetünkben Intel x86-tal, egyébként elő kell készítenünk egy keresztfordítót.
2. A rendszermag beállítása.
Szerezzük be a legfrissebb megbízható (stable) rendszermag forrását (a cikk írása idején ez a 2.2.17-as változat). A beállításokról és a fordításról a forrásfájlok megfelelően tájékoztatnak, így az ott leírtakat itt nem ismételjük meg. Azt azonban meg kell jegyeznünk, hogy szükség lesz a kezdeti memórialemez, valamint a betölthető modulok támogatására és még számos beállításra. Ha alaprendszerünkben valamilyen egyedi eszközt – például egy hálózati kártyát – kívánunk használni, modulként támogathatjuk. Ezeket a modulokat azután szabadon telepíthetjük és használhatjuk az alaprendszerben.
3. Az alapkönyvtárak előkészítése.
Szerezzük be a glibc legfrissebb változatát (jelenleg ez a glibc2 2.1.3). Itt szinte mindent megtalálhatunk, amire szükségünk lesz – a dinamikus betöltőt, a szabványos C könyvtárat, a matematikai könyvtárat és máségeket. Mindenre természetesen nem lesz szüksé-



2. ábra Az alaprendszer összetevői

© Kiskapu Kft. Minden jog fenntartva

günk, azonban jobb egyszerre előkészíteni az egészet, és azután kiválasztani a megfelelőket. A glibc források között található leírások elegendő útmutatást adnak a fordításhoz. Mivel azonban a könyvtárakat a célgépen fogjuk használni, a fordítást a 2. lépésnek megfelelő rendszermag-fejlécfájlokkal kell elvégeznünk, ehhez a with-headers kapcsolót kell használnunk. Mindemellett a könyvtár fejlécfájlokat is telepítenünk kell, hogy a célgép számára fordítandó többi programhoz is elérhetők legyenek.

4. A keresztfordítás beállítása.

A rendszermag és a glibc fejlécfájloinak telepítése után fel kell készítenünk a gcc fordítóprogramot a használatukra. Erről részletesen a Glibc2-HOWTO fájlban olvashatunk. Röviden összefoglalva, teendők lényegében az, hogy a -b kapcsolóval megadjuk a gcc-nek, merre találja a meghatározásokat. Esetünkben – mivel a célgép és a gazdagép alapjában ugyanaz – elegendő a gazdagép adatait használnunk, amiket a gcc -v paranccsal deríthetünk ki. Saját géptől esetén ezt a választ kaptam a parancsra:

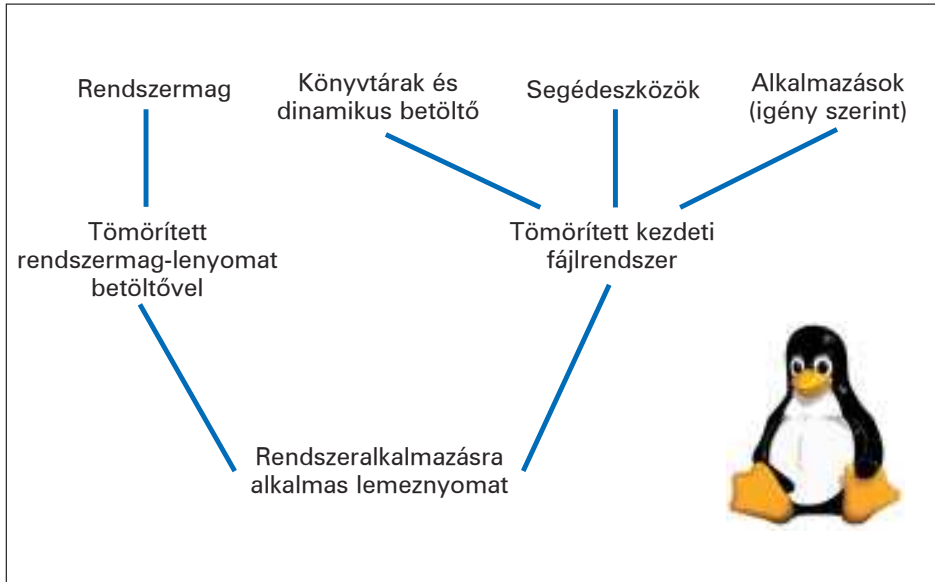
```
Reading specs from
/usr/lib/gcc-lib/i386-redhat-linux/egcs-2.91.66/
specs
gcc version egcs-2.91.66 19990314/Linux
(egcs-1.1.2 release)
```

Fordítsuk le a rendszer többi elemét is a -b kapcsolóval, valahogy így:

```
gcc -b i386-redhat-linux
```

5. A kívánt könyvtárak és segédeszközök kiválasztása és előkészítése.

Egy segédprogram – például a mount – vagy valamely glibc-ben nem szereplő könyvtár, mondjuk a termcap lefordításához mindenképp tudatnunk kell a fordítóval, merre találja a fejlécfájlokat (include) és a szükséges könyvtárakat. Először is, a --nostdinc kapcsoló használatával közöljük a fordítóval, hogy ne keresse a fejlécfájlokat az alapértelmezett útvonalak mentén. Ezután a -b \$MACHINE kapcsolóval tudassuk vele azt is, hogy a fordítás a célgép számára készül. Ha ezzel megvagyunk, adjuk meg a rendszermag és a szabványos könyvtár fej-



3. ábra A rendszer elemeinek elrendezése

lécfájlnak helyét a -I kapcsolóval. Végezetül a -L és -I segítségével határozzuk meg, mely könyvtárakat használja a betöltő, és ezeket hol találhatja meg.

6. Az alkalmazások elkészítése.

A könyvtárakhoz és segédprogramokhoz hasonlóan az alkalmazások keresztfordítására is sort kell kerítenünk, ha a cégépen szeretnénk használni őket. A rendszer szempontjából semmiféle különleges intézkedésre nincs szükség, feltéve természetesen, hogy az alkalmazás működéséhez szükséges egyéb elemeket telepítettük.

7. Becsomagolás.

Ha az elemekkel elkészültünk, már csak megfelelő elrendezésükről kell gondoskodnunk, hogy rendszerindításkor minden a helyére kerüljön. A következő szakaszban kiderül, hogyan.

8. További kiegészítések.

Az alaprendszert kiindulási pontként használhatjuk kézzel fogható céljaink megvalósításában. Ezek eléréséhez rendszerünket további kiegészítésekkel láthatjuk el, de most ennyi elég is, hiszen a témának külön szakaszt szenteltünk.

Az alaprendszer kialakítása

Még nem találkoztam olyan leírással, mely összefoglalná, hova helyezzük a leíratokat, a futtatható fájlakat, a lefordított kódokat, és a parancsfájlokat, vagyis hogyan csomagoljuk össze a rendszert. Ez nem csoda, hiszen ahány rendszer, annyiféle csomagolási módszer bizonyulhat megfelelőnek. Mindeközben természetesen az adott elemek elkészítése nem különbözik. A legegyszerűbb és leggyakrabban alkalmazott csomagolási módszer a hajlékonylemezek használata.

Ennek általános lépései a következőkben foglalhatók össze:

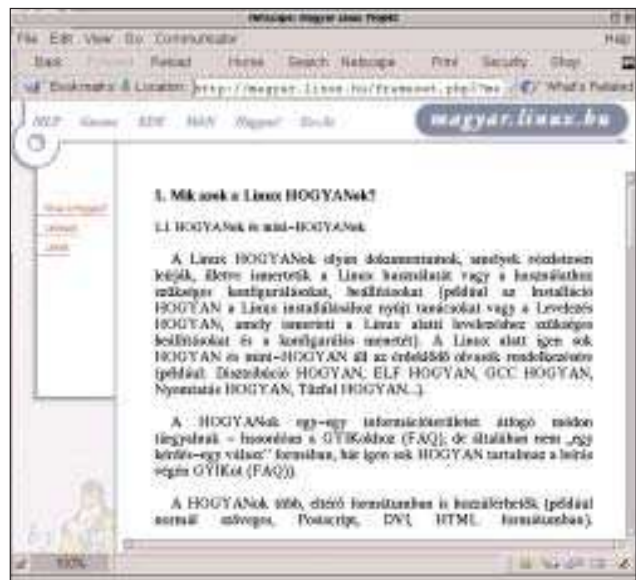
1. Készítjük el a rendszerhez szükséges alkotóelemeket, így a rendszermag lenyomatát, a könyvtárakat, a parancsfájlokat, a beállítókat és minden más szükséges összetevőt.
2. Hozzuk létre az alaprendszer kezdeti fájlrendszerének könyvtár-szerkezetét.
3. Helyezzük el az összetevőket a fájlrendszerben, és hozzuk létre a létfontosságú elemeket, köztük az eszközöket képviselő leírókat (device nodes) is.
4. Tömörítjük a fájlrendszert.
5. Adjuk meg a rendszermag számára, hol található meg a kezdeti fájlrendszer lenyomatát, ehhez állítsuk be a megfelelő jelzőket a rendszermag lenyomatában.

6. Írjuk fel a rendszermagot és a tömörített főlenyomatot egy hajlékonylemeze, és tegyük azt indíthatóvá. Az eljárás részletesebb bemutatására írtunk néhány fordításvezérlő fájlt, melyek gyakorlatilag a fenti lépések megvalósításához tartalmazznak útmutatásokat, és kis alaprendszert hoznak létre ingyenesen elérhető programcsomagok segítségével. Egy egyszerű – szintén ingyen letölthető – alkalmazást futtatunk, mely a netperf severt kapta, feladata pedig a TCP/IP veremek teljesítményének ellenőrzése. A Netper honlapján olvashatunk a rendkívül hasznos programról (➔ <http://www.netperf.org>). Az alkalmazások indítása az alaprendszer beállításaitól függően sokféleképpen történhet. Az esetek többségében a Linux rendszermag egy indítási parancsfájlt vagy egy

futtatható fájlt (init, illetve linuxrc) indít el a kezdeti fájlrendszerben a rendszerindításkor. Az init program teendői közé tartozik általában az alapfájlrendszer újrafűzése, az olvasás/írás engedélyek beállítása, más fájlrendszerek – így a proc – befüzése, valamint a rendszer más részeinek – köztük a bejelentkezési felület – üzembe helyezése, vagy közvetlenül az alkalmazás indítása. A SysVinit közismerten jól kezeli ezt a feladatot.

Alaprendszerünk esetében a bemutatáshoz nincs szükségünk bonyolult rendszerindítási folyamatra, csak egy egyszerű héjprogramot alkalmazunk, melyet bárki kedve szerint módosíthat vagy kiegészíthet:

```
mount -n -o remount,rw /
mount /proc -t proc
echo MyCompanyName, Version X.Y. Built Z,
➔ August 2000
exec /bin/sh
```



A hazai HOGYAN-ok központi oldala

Feladatként próbáljuk az echo sor tartalmát áthelyezni alkalmazásunkba, melyet a héj helyett rögtön el is indíthatunk. Az adatok kiíratását C++-ban a következőképpen oldhatjuk meg:

```
cout << COMPANY << VERSION_NO << BUILD_NO
➤ << __DATE__ << __TIME__;
```

Esetünkben a kiíratás után visszkapjuk a parancssort:

```
pipe-elix> MyCompanyName, Version X.Y,
➤ Build Z, August 2000
pipe-elix>
```

További kiegészítések

Ha elkészültünk az alaprendszerrel, elgondolkodhatunk azon, hogy miként tegyük gazdagabbá néhány érdekes alkalmazással. Mivel alaprendszerrel van szó, itt is lépésről lépésre haladva kell kiépítenünk a maink programbirodalmát. A következőkben felsorolt lehetőségeket mindenesetre érdemes megfontolnunk:

- egy rendszerfelépítő program: a SysVInit nagyszerű választás, azonban egyszerű alkalmazások számára kissé túlméretezett,
- egy biztonsági lehetőség: bejelentkezések támogatása,
- egy szerkesztőprogram: vi vagy emacs.
- további hálózati lehetőségek: telnet vagy ftp démonok,
- grafikus felület: X,
- nem felejtő tároló: flashmemória és merevlemez,
- további betölthető modulok,
- csomagkezelés az rpm-mel.

Valójában nincs szükség minden csomag újbóli lefordítására, ugyanis könnyen található az adott processzorhoz már lefordított változatot. Esetünkben például, ahol a cél- és a gazdagép azonos típusú, egyszerűen felhasználhatjuk a gazdagépen található lefordított fájlokat. Vegyük például a top segédprogramot – csak átmásoljuk a futtatható fájlt az alaprendszerbe, és más teendőnk nincs is. A helyzet nem minden esetben ilyen egyszerű, hiszen a helyes működéshez fel kell kutatnunk a futtatható fájl függőségeit, vagyis azokat a könyvtárakat és beállításfájlokat, amelyekre szüksége van a működéshez, de ezeket a program futtatásáig nem igazán tudjuk azonosítani. Azért itt is akad segítség, az ldd és az strace segítségével felderíthetjük a függőségeket. Nekem például csak át kellett másolnom az Emacs futtatható fájlját (emacs-nox), néhány osztott könyvtárat és beállítási fájlt az alaprendszerbe, és láss csodát – a szövegszerkesztő működött. Ha nem szeretnénk mágneslemezzel indítani a rendszert, egy kis pluszmunkával a következő eszközökről is megtehetjük:

- EPROM-ról,
- hálózaton keresztül,
- különböző eszközökről, így flashmemóriából, egy lemezzel, CD-ről vagy ziplemezekről, vagyis a hajlékonylemez helyett szinte bármiről.
- másik operációs rendszerből.

Kapcsolódó címek

Linux HOWTO-k:

A rendszermag, a ramdisk, a gcc, a glibc, az amdisk és más HOWTO fájlok letölthetők a következő címről:

- <http://metalab.unc.edu/pub/Linux/docs/HOWTO>
- <http://vbzo.li/linux/Magyarul-HOGYAN.html>
- <http://magyar.linux.hu>

A Netperf honlapja:

- <http://www.netperf.org/>

Ha elég időt és fáradságot szánunk rá, rendszerünket ily módon ugyanolyan gazdaggá tehetjük, mint amilyenek befutott társai.

Hibakeresés

A Linux használatának egyik nagy előnye, hogy számos segédeszköz és leírás áll rendelkezésre rendszerünk testreszabásához és gondjaink megoldásához. Beleláthatunk minden program kódjába, és a rendszer semmit sem titkol el előlünk. Emellett számos hasznos forrásmű látott napvilágot mind nyomtatásban, mind pedig a világhálón. Nincs még egy olyan rendszer, mely e tekintetben versenybe szállhatna a Linuxszal – még a FreeBSD sem.

Ha egy nehézséggel állunk szemben, általában több megoldási módot is kitalálhatunk. Természetesen mindig a legjobbat szeretnénk választani, azonban hogy melyik bizonyul annak, azt legtöbbször csak a kipróbálás után tudjuk eldönteni. Lapozzuk fel a megfelelő Linux HOWTO-kat és leírásokat, vagy kérdezzük meg linuxos ismerőseinket. Feladhatjuk kérdésünket egy linuxos hírcsoportban is, ahol valaki talán kisegít egy gyors válasszal.

Ha pedig mindenképpen laposabb szeretnénk tenni pénztárcánkat, fordulhatunk olyan Linuxszal foglalkozó cégekhez is, melyek ilyen kérdések megválaszolására hívatottak. A gdb nagyszerű hibakeresési eszköz az alaprendszer alkalmazásaihoz. Ha nem tudjuk a teljes gdb programcsomagot futtatni célrendszerünkön, távolról is üzemeltethetjük kisebb részeit, akár egy gdb csont, akár egy gdb kiszolgáló futtatásával. Végezetül, a hibakeresést nagymértékben segítheti a syslogd démon is.

A hibák felderítésére számos jól bevált módszer létezik: mindegy, melyiket használjuk – a cél a megfelelő kiút megtalálása. Általában egy sikeres példából tanulhatunk a legbiztonságosabban. Ha belekukkantunk a Red Hat vészrendszerébe, érdekes dolgokat fedezhetünk fel. Ehhez csak a következő parancsokat kell alkalmaznunk:

```
cat rescue.img | gzip -d > rescue_root.img
mkdir rescue_root
mount -o loop rescue_root.img rescue_root
```

A rescue.img az images könyvtárban található tömörített lemezle nyomat. Tartalmát a következőképpen tekinthetjük meg:

```
ls rescue_root
```

Az eredmény a következő:

```
bin dev etc lib lost+found mnt proc sbin tmp usr
```

Összefoglalás

Ez az írás csupán bevezetőként szolgál az alaprendszer testreszabásához. A valódi esetekben a folyamat ennél jóval összetettebb lehet. Különösen, ha a kód szintjén is szükség van módosításokra, például egyedi eszközök támogatásánál. Mindenesetre azt láthatuk, hogy a feladat végrehajtása semmiképpen sem ördögösség, és éppen ennek bemutatása volt a célom. Saját testre szabott alaprendszerünkkel pedig immár biztosabban haladhatunk célkitűzéseink megvalósítása felé.



He Zhu

(hezhu@yahoo.com)

a rendszerprogramok és a hálózatok terén otthonos. Jelenleg New Jersey-ben dolgozik, a Bell Labs kötelékében.

Könnyű álmok (2. rész)

Nézzük végig az alapvető biztonsági kérdéseket a hálózattervezés, az operációs rendszer telepítése és a programfejlesztések háza táján.



Mielőtt rátérnénk a hálózati biztonsági veszélyforrások ismertetésére, meg kell értetnünk ezek kiváltóit. Az okok feltárása nélkül nehezen érthető meg az, hogy mi ellen kell megvédenünk rendszerünket. A leggyakrabban olyan – a programfejlesztők által elkövetett – tervezési vagy kivitelezési hibák okozzák a rendszerek támadhatóságát, amelyek a korszerű tervezési módszerek alkalmazásával és némi ráfordítással elkerülhetők. Miért kell mégis félnünk a betörésektől? Mert a fejlesztő cégek a költségek csökkentéséért nem alkalmazzák ezen módszereket. A hálózati rendszerek tervezői és kivitelezői pedig hajlamosak a tapasztalatukra, vélt szaktudásukra hagyatkozni. Ha az ember figyel a biztonsággal foglalkozó levelezési listákat (*lásd a táblázatban*), akkor nap mint nap tapasztalhatja, hogy a nagy és drága rendszerek tervezői is fittyet hányanak ezen elméletek betartására, és olyan alapvető biztonsági hibákat hagynak rendszerükben, amit a hasonló célú rendszerekből már évekkkel ezelőtt kifogtak. Ha a rendszerek fejlesztői csak arra vennék a fáradságot, hogy a korábbi fejlesztési tapasztalatokat megismerjék, a biztonsági hibák nem lennének olyanok, mint a visszajáró kísértek. Jó példa az a hiba, amit a webkiszolgálók fejlesztői szinte kivétel nélkül meghagynak a rendszerükben: a szakma csak úgy hívja, a pont-pont hiba (dot-dot bug). Ennek az a lényege, hogy a webkiszolgáló nem ellenőrzi az elérési útban található „.” karaktereket, vagy azok kódolt változatát, így nemcsak a beállítófájlból engedélyezett könyvtárak tartalmához lehet hozzáférni, hanem bármihez. Ez a jellegzetes „állatorvosi ló” esete. Havonta van olyan HTTP protokollt használó eszköz, melyben felfedezik ezt a hibát. Olyan webkiszolgáló is létezik, ahol egyszerűen kijavították, aztán visszakerült.

A nyílt forráskód előnye

Amikor egy hiba napvilágra kerül, a rendszer fejlesztőjének joga van eldönteni, hogy mikor javítja ki. Ezzel korábban igen komoly nehézségek akadtak, mivel a felderített hibákat a fejlesztők lassan vagy egyáltalán nem javították ki. A biztonsággal foglalkozó levelezési listák nagy előnye, hogy sikerült kikényszeríteni a nagy programgyártó cégekből is a gyors visszahatást, hiszen ha itt megjelenik egy hiba, akkor azt azonnal ki kell javítani. Nagyságrendekkel jobb a helyzet a nyílt forráskódú rendszereknél, ugyanis mivel a kód nyílt, a fejlesztők nem helyezhetnek el semmilyen meglepetést (olyan programrészt, amely például adatokat továbbít a fejlesztőcégnak). A nyílt forráskód következtében egyszerűbb egy átfogó kódvizsgálat alá vetni, ezeket a rendszereket (léteznek ilyen kezdeményezések). A nyitott forráskódú rendszerek sem mentesek a hibáktól, azonban a forráskód nyíltságának köszönhetően a hibák javítása itt lényegesen gyorsabb. Ha egy levelezési listán egy rendszer hibájáról olvashatunk, a hiba felfedője gyakran elküldi a javítást is. Nézzük most az általánosan jelentkező hibákat.

Rendszertervezési hibák

Ma már elfogadott tény, hogy komolyabb rendszert megfelelő minőségben nem lehet létrehozni átfogó tervezés híján. A rendszer elfogadható minőségéhez pedig ugyanúgy hozzátartozik a biztonsági szempontok figyelembevétele, mint a kis erőforrásigény vagy a hibátlan adattárolás. A tervezési munkát megkönnyítendő a számítástechni-

kában is létrejöttek programtervezési és fejlesztési módszertanok. Ezek leírják, hogy a programtervezőknek és fejlesztőknek milyen lépéseket kell megtenniük ahhoz, hogy a végtermék a várt minőségű legyen. Akkor járunk a legközelebb az igazsághoz, ha úgy képzeljük el ezeket a módszertanokat, mint az informatika minőségbiztosítási szabványrendszerét – ugyan itt nincs elfogadott szabvány. Ha egy rendszert valamelyik módszertan szerint terveztek és fejlesztettek, akkor biztosak lehetünk benne, hogy abban nincs több hiba, mint amennyit az adott módszertan megenged. Több ilyen módszertan létezik, minden rendszer jellemzője azonban, hogy a szolgáltatásbiztonságra összpontosít (azon belül is elsősorban adatközpontú), és általában nem szentel elegendő figyelmet a biztonságnak.

Fejlesztési hibák

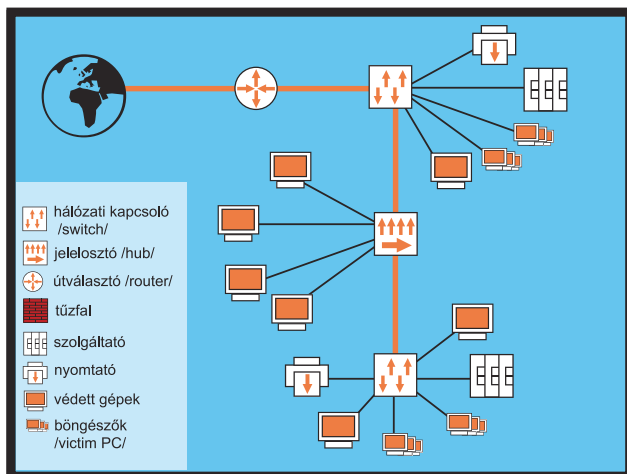
Amennyiben a rendszer terve megfelelő, a labda a rendszerfejlesztőké. Ha ők valamilyen hibát ejtenek, akkor a rendszer a tervezők lelkiismeretes munkája ellenére is sebezhető lesz. A legtöbb programozási hiba sokszor ismétlődik, így a fejlesztők fel tudnak – tudnának – készülni ellene. Ilyen elkövetett programozási hibák minden programozási nyelvben megtalálhatók. Vannak ugyan olyan leírások is, melyek a hiba elkerülésének útját írják le [1., 2.], ezek elolvasására azonban nem minden fejlesztéssel foglalkozó cég kötelezi a fejlesztőit. Ha tanulmányoznák ezeket az anyagokat, akkor a hibák jó része kiküszöbölhető lenne. A forráskód felülvizsgálatakor ezek a hibák megfoghatók, azonban ennek lényegesen komolyabb erőforrásigénye van, így célszerűbb már a fejlesztés idején nem hibázni.

Common Criteria

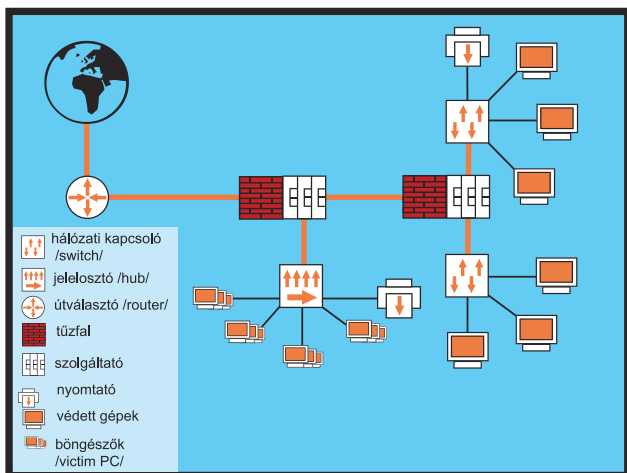
Ha egy program fejlesztésénél a biztonság fontos szempont, akkor célszerű a CC (Common Criteria – Általános elvárások) [3.] időszertű változatát figyelembe venni a tervezésnél és fejlesztésnél. A CC egy nemzetközileg elfogadott fogalom- és követelményrendszer, amelyet korábban az egyes országok saját követelményrendszereiként használt belső szabványokból (többek között a TCSEC-ből) fejlesztettek tovább. Jelenleg még nem minden országban elfogadott, de a jelentősebb informatikabiztonsági fejlesztők már figyelembe veszik, esetenként a fejlesztés teljes egészében a CC iránymutatásai szerint folyik. A felhasználói számára biztosítja, hogy az egyes biztonságtechnikai fogalmak minden tervező és fejlesztő számára ugyanazt jelentsék, és a termékek biztonsága nemzetközileg is mérhető legyen. Ha egy rendszer fejlesztői el akarják ismertetni terméküket mint CC alapján bevizsgált eszközt, akkor rákényszerülnek, hogy a rendszerre leleselkedő veszélyeket felderítsék és elhárítsák.

Követelmények

A CC hét garanciaszintet határoz meg, ezek egyre erősebb intézkedésekkel kényszerítik a fejlesztőket arra, hogy az eredeti elképzelések szerint, hibátlanul működő rendszert hozzanak létre. Tehát a garanciaszint *nem befolyásolja* a biztonsági szintet, csupán biztosítja, hogy a program csak a tervezői által elképzelt szolgáltatásokat nyújtja. Az alacsonyabb szintek követelményei viszonylag egyszerűen teljesíthetők (például kötelező jó minőségű felhasználói leírást szállítani a rendszerrel), a harmadik és negyedik szint már erősebb követelmé-



1. ábra Rosszul tervezett hálózat vázlata



2. ábra Jól tervezett hálózat vázlata

nyeket támaszt (többek között átfogó ellenőrzést kell végezni), míg a hetedik szinten olyan követelményeknek kell megfelelni, melyek kielégítéséhez igen jó szakemberekre és komoly energiabefektetésre van szükség. Követelmény például az, hogy minden algoritmus helyességét matematikai módszerekkel bizonyítani kell(!). Belátható, hogy a magasabb garanciaszintek lényegesen magasabb szintű biztonságot adnak ugyan (jó tervezés mellett), de sokkal nagyobb az erőforrásigényük. Nagyobb mennyiségű és általában jobb minőségű munkaerőre van szükség – így nő a költség is. Mivel azonban a költség növekszik, és ma még szinte minden cég elsődleges célja a lehető legnagyobb nyereség elérése, így kevés a CC szerint fejlesztett rendszer. Ha egy rendszer minősítetten megfelel a CC valamely szintjének, akkor lényegesen nyugodtabbak lehetnek a rendszer felhasználói: ennél a fejlesztésnél a biztonsági szempontokra is gondosan ügyeltek. Különösen kellemetlen hibák az operációs rendszerek tervezési hibái (ideértve a különböző Linux-változatok hibáit is).

Az operációs rendszer

Minden számítógép működésének alapja az operációs rendszer, ennek szempontunkból legfontosabb jellemzői: többfelhasználós, több feladat párhuzamos végrehajtására képes, és számítógépes hálózatot tud kezelni. Amennyiben ezek a tulajdonságok jelen vannak – ez ma már szinte minden operációs rendszerre igaz –, akkor a rendszer veszélyeztetett. Ha tervezési vagy telepítési hibák miatt biztonsági hiányosságok adó-

1. ábra

Néhány a felmerülő gondok közül: a rendszerben középen elhelyezkedő jelelosztóhoz csatlakoztatott ügyfelek lehallgathatják a két hálózatrész forgalmát, sőt, akár meg is támadhatják valamelyik rendszert, miközben a másik rendszernek adják ki magukat (spoofing). A két hálózati kapcsoló között kifeszített virtuális hálózatok (VLAN) is támadhatók.

2. ábra

A rendszer logikai részei fizikailag is jól elkülönítettek. A két szolgáltató tűzfalakra jellemző feladatokat is ellát, így a rendszerek elérhetőége finoman szabályozható. A világhálóval érintkező gépen a kívülről is elérhető szolgáltatások száma minimalizálható. A megvalósításához elég egyetlen nyilvános IP-cím.

nak, akkor a rendszerbe betolakodók hatolhatnak be. Ha a rendszer nem megfelelően felügyelt, vagy erre nincs lehetőség, akkor a kalóz azt tesz az ott található adatokkal és erőforrásokkal, amit akar. Ha a behatolók elég felkészültek, az üzemeltetők nem értenek a dolgukhoz a szükséges szinten vagy maga a rendszer nem ad lehetőséget a teljes felügyeletre, a rendszer jogos felhasználói a behatolást észre sem veszik. Erre az esetre fejlesztik a behatolásérzékelő rendszereket (IDS – Intrusion Detection System [4.]), mézes bödönöket (honey pot) és más eszközöket, de ez egy másik mese lesz. Ha a rendszerben felfednek egy rést, akkor általában csak az operációs rendszer vagy az adott részek frissítése segíthet. Bizonyos rendszereken ezt Szervizcsomagként, máshol „patchmatrix”-ként emlegetik, a Linux-rendszerben – erős modularitása következtében – elegendő a megfelelő csomagok frissítéséről gondoskodni.

Az operációs rendszer hibái

Az operációs rendszerben öt szinten fordulhat elő hiba: a rendszermagban, függvénykönyvtáraiban (lib vagy dll), az alaprendszer futtatható segédprogramjaiban, a rendszerdémonokban vagy a rendszer alapbeállításáiban. Szerencsére elmondható, hogy a Linux rendszermag stabil változatában nagyon ritkán találunk komoly biztonsági rést. A legutóbbi hibát 2000. június 7-én hozták nyilvánosságra. A hiba a 2.2.15-ös rendszermagban volt és a 2.2.16pre6 számúban azonnal javították, illetve elérhető volt egy olyan rendszermodul, amely segítségével a 2.2.15-ös rendszereken sem volt a hiba kihasználható. A lehetőség megragadásával egy rendszer helyi felhasználója bizonyos körülmények között rendszergazdai jogokra tehetett szert. A rendszer függvénykönyvtáiról is viszonylag ritkán derül ki hiányosság (sajnos, az utóbbi időben volt rá néhány példa), a rendszer alapvető segédprogramjai ritkán tartalmaznak biztonsági réseket, a Linux-változatokat pedig nagy gondalal tervezik, így a telepítésnél ritka az olyan biztonsági rés, amely az operációs rendszer alapbeállításainak hibájából adódik. Más rendszerek gyakran igen komoly biztonsági réseket tartalmaznak, melyek folyamatosan kerülnek felszínre, így a rendszerek felhasználói soha nem alhatnak nyugodtan. A helyzetet súlyosbítja, hogy az említett rendszerek forráskódja általában nem érhető el, így nem lehetséges egy átfogó forráskódvizsgálattal a jelentősebb hibák kiszűrése. A zárt forráskód miatt a hibák javítása is nehezekebb. Sajnos, még mindig kerülnek használatba olyan operációs rendszerek, melyeken adott felhasználóknak – biztonsági vagy bemutató célzattal – előre ismert jelszavuk van, ezzel azonban megkönnyítik a kalózok dolgát.

Telepítés – a karácsonyfa modell

Biztonsági szempontból kényes pont a telepítés. Ha az operációs rendszer telepítőkészlete lehetővé teszi a könnyű telepítést – rengeteg szolgáltatással, akkor gyakorlatlan felhasználók kevés tapasztalattal is nekiállnak és telepítenek. Így születnek azok a rendszerek, amit Lilo barátunk így jellemezett: – „Karácsonykor odamegyünk és letesszük

alá az ajándékokat”. Egyszóval karácsonyfa. Ezzel a kettősséggel küzdenek a Linux változatai: ha a rendszert népszerűbbé akarják tenni, akkor a felhasználók dolgát a telepítésnél meg kell könnyíteniük, ha biztonságra törekszenek, akkor a rendszer telepítésekor a használhatóságot a lehető legbiztonságosabban kell elérni. Régi mondás: „Bármely rendszer biztonsága fokozható a teljes használhatatlanságig.” Ez aztán ellentmondás a javából. Valamit segítenek azok a telepítőkészletek, melyek kézen fogva vezetik a felhasználót a szükségesnek vélt szolgáltatások telepítéséig, azonban a tapasztalatlan felhasználó hajlamos olyan eszközöket is feltelepíteni, melyre nincs szükség. Azt gondolják: – „Jól hangzik a neve, majd kipróbálok, mire jó!”. Ezeknek a programoknak a nagy részét soha nem indítják el.

Csomagok és kezelők

Az egyedi telepítés kiválasztásával talán elkerülhető lenne a jól ismert „Felhasználói munkaállomás” típusú telepítés, ez azonban hosszadalmas. A típusos telepítés viszont magával hozza azokat a típushibákat is, amelyeket a terjesztés figyelmetlen összeállítói esetleg benne hagytak a telepítőkészletben. A csomagkezelőből adódó nézőpontbeli különbség a két legnépszerűbb Linux-változat között: a Debian rendszere (deb kiterjesztésű csomagok) az adott alrendszer telepítésekor megkísérli annak beállítását és elindítását is. Ennek a megoldásnak sajnálatos mellékhatása, hogy egy csomag telepítésekor olyan szolgáltatások is elindulhatnak, amelyeket a rendszer gazdája egyelőre nem akart elindítani. Ha egyszer sikeresen leállítottunk egy ilyen szolgáltatást és arról is gondoskodtunk, hogy a rendszer indulásakor se induljon újra automatikusan, akkor nincs is több gond vele – a következő frissítésig. Akkor ugyanis újraindulnak a szolgáltatások. Ez a dpkg csomagkezelő program jelen változatában csak komoly kommandózással küszöbölhető ki, mivel az kizárólag a csomagban tárolt utasításokat hajlandó elfogadni.

A Red Hat által fejlesztett csomagkezelő rendszer (rpm kiterjesztésű csomagok) a felhasználók szempontjából kissé kényelmetlenebb. A csomagok telepítésekor nem kényszeríti ki azok beállítását, és a rendszergazdára bízta annak eldöntését, hogy az adott szolgáltatást el akarja-e indítani. Biztonsági szempontból mindenképpen az utóbbi eljárás a hatékonyabb. A Debian csomagokban lévő beállítási állományok alapértékei általában biztonságosabbnak mondhatók, és ki tudna lemondani az apt-get parancs kényelméről. Ezek után mindenki döntse el, mit választ. Biztonsági szempontból mindkettő megfelelően használható, ha valaki tapasztalt, akkor mindkét rendszert biztonságossá tudja tenni. Ha valaki önállóan kezd egy Linux-rendszer telepítéséhez, és szempont a biztonság (ha valamilyen módon hálózathoz fog csatlakozni) akkor javasoljuk csak azokat az alrendszereket telepíteni, amelyek valóban szükségesek. Mindenki döntse el, hogy pontosan mire szeretné használni a rendszerét, és ennek függvényében válogasson a rendelkezésre álló csomagok közül. Kezdetben célszerű csak a szükséges és ismert csomagokat telepíteni, később szükség

esetén bővíteni lehet az elérhető rendszerek számát. A csomagkezelők függőségkezelő rendszere segít, mivel telepíti azokat a csomagokat, amelyekre a kiválasztott alrendszer működéséhez szükség van.

A hálózati felépítés hibái

Ha a számítógépes hálózat tervét nem ellenőrzi biztonsági területen jártas szakember, akkor bizony előfordulhat az alábbi elképzelt (sajnos nem ritka) eset: egy cég éppen új irodába költözik. A vezető a költségek csökkentése miatt olyan céget bíz meg a hálózat tervezésével és kivitelezésével, amelynek hihető referenciái ugyan nincsenek, de saját állítása szerint jó a szakmában és – a cégnek elsősorban ez számít – olcsón dolgozik. A cég hálózatát a lehető legegyszerűbb felépítésre tervezik. (1. ábra) A tervek szerint a könyveléssel foglalkozó részleget nem lehetséges leválasztani a cég programfejlesztő részlegetől. Ha a cég valamelyik fejlesztőmérnöke rosszindulatú, hozzáférhet a fizetési adatokhoz, akár módosíthatja is azokat. Ha a hálózat helyesen lenne tervezve (2. ábra), akkor a két hálózat fizikailag leválasztható lenne egymásról, és a pénzügyi részleg egy tűzfalal védhető lenne. Ha egy hálózaton van minden gép, akkor megvalósítható a cikksorozat előző részében ismertetett lehallgatás. Ennek elkerülésére szokták – szintén költségmegtakarítási okból – a hálózatot szétválasztani virtuális hálózatokra (VLAN). Ez azt jelenti, hogy a hálózatban lévő aktív eszközökkel (leggyakrabban hálózati kapcsolókkal) a forgalmat úgy irányítják, hogy az egyes virtuális hálózatokban lévő gépek csak a velük azonos hálózatban lévő gépeket láthassák. Van azonban egy kis baj: amennyiben a kapcsoló vezérlőprogramjában biztonsági rés van (sajnos, többször is előfordult már), akkor az egyes virtuális hálózaton lévő gépek a hálózati kapcsoló megtévesztésével láthatják más VLAN-on elhelyezkedő gépek forgalmát, rossz esetben módosíthatják is azt. Miután a rosszindulatú hálózatot újra egyesítette, a lehallgatás már kivitelezhető, vagy megvalósítható egy középre belépéses („man-in-the-middle”) támadás is. Jelenleg általánosan elfogadott álláspont: a hálózati kapcsolók (switch) nem biztonsági eszközök. Tervezősikkor általában nem a biztonság az elsődleges szempont.

Hibás rendszerek használata

Biztonsági hibákat lépten-nyomon találnak szinte minden rendszerben. Unatkozó varázslók (hacker, geek) találják meg, vagy kalózkod, ez csak szerencse kérdése. Ha varázslók, akkor a hibát először a fejlesztőkkel közlik, általában a javítással együtt, és várják a hiba és a megoldás közzétételét. Ezek után – jó esetben – a fejlesztők értesítik a nyilvánosságot, hogy a rendszerben olyan biztonsági gond merült fel, amely veszélyezteti a használók biztonságát. Amennyiben a fejlesztők nem válaszolnak, akkor a hiba felfedezői elküldik annak részletes leírását, a bemutatására alkalmas programot és a javítóködöt valamelyik biztonsági levelezőlistára. Amikor a hiba napvilágot lát, akkor adódhat egy kis bökkenő. Ha ugyanis egy felhasználó nem olvassa az adott listát, akkor rendszere átmenetileg védtelen a behatók ellen. Az ilyen felfedezett, de az adott rendszeren még nem kijavított hibák a betörések leggyakoribb okai. Ha a hibát nem is egyszerű kihasználni, a szemléltető program felhasználásával, vagy annak kis módosításával betörőeszköz nyerhető. Ha a betörő nem rendelkezik elegendő szakértelemmel, akkor a hiba tudatában is tehetetlen lenne, a jóindulatú bemutatóprogram segítségével azonban... Az ilyen kisebb szakértelemmel, mások által fejlesztett eszközök segítségével behatók kalózkodnak hívja az angol szaknyelv „script kiddie”-nek. Ennek az általánosan elterjedt gyakorlatnak a megállítására egyre gyakrabban csak a szemléltető program vázát adja közre a hiba felderítője, így a hozzáférők látják a gondot, a fent említett hozzá nem értők viszont segítség nélkül nem tudnak mit kezdeni a példa-

Biztonsággal foglalkozó levelezőlisták

LEVELEZŐLISTA CÍME	HOL LEHET FELIRATKOZNI
bugtraq@securityfocus.com	http://www.securityfocus.com
vuln-dev@lists.securityfocus.com	http://www.securityfocus.com
secprog@securityfocus.com	http://www.securityfocus.com
sf-news@securityfocus.com	http://www.securityfocus.com
firewall-wizards@nfr.com	http://www.nfr.com/forum/firewall-wizards.html
debian-security@debian.org	http://www.debian.org/MailingLists/subscribe
debian-security-announce@lists.debian.org	http://www.debian.org/MailingLists/subscribe
linux-security@redhat.com	https://listman.redhat.com/mailman/listinfo/linux-security
suse-security@suse.com	http://www.suse.com/us/support/maillinglists/index.html
security-l@sunserv.kfki.hu	http://sunserv.kfki.hu/mailman/listinfo/security-l

programmal. A hibák ilyen szemléltetése arra mindenképpen jó, hogy szükség esetén rákényszerítse a rendszerek fejlesztőit a hiba mielőbbi javítására. Ha a hibára kalóznak akadnak rá, akkor a helyzet lényegesen súlyosabb. Míg az adott résen való behatolást nem sikerül valakinek észlelnie, addig zavartalanul járhatnak ki-be azokon a rendszereken, melyek a hibás alrendszert tartalmazzák. Jó példa az akkor tájt egyik legismertebb biztonsággal foglalkozó nemzetközi weblap www.rootshell.com feltörése, ahol a betörés módjára csak hónapokkal később sikerült rájönni. (Az incidens rövid ismertetése a [11.] *hivatkozásnál* található.) A gondot súlyosbítja, hogy a kalóznak kiterjedt nemzetközi betörőprogram-cserehálózata van. Abban az esetben, ha a rendszer forráskódja nyílt, akkor lehetőség nyílik olyan megelőző intézkedések végrehajtására, melyek a hibák felkutatására irányulnak. Jelenleg is folynak ilyen irányú megelőző erőfeszítések az LSAP (Linux Security Audit Project) [5.], az OpenBSD Security [6.] és számos más kezdeményezés keretében.

Beállítási hibák

Ha a rendszert sikerült megfelelően megtervezni, kifejleszteni és a telepítés is jó, akkor már csak a beállításoknál véthetünk hibát. Nagyon gyakori hiba a következő: egy közepes cég eljut arra a szintre, hogy már feleslegesen sok az ISDN vonal használatából adódó havi költségük. Úgy döntenek, hogy bérelt vonali kapcsolatot építtetnek egy Linux-alapú tűzfalal, amin keresztül leveleznek és böngésznek. A megbízott cég olcsón dolgozik és látszik rajtuk, hogy nagyon értenek hozzá, mert olyan szavakat használnak beszéd közben, mint „lokálintérfész” meg „ipécsénszrül” és így tovább. Megteszik. Elmennek. Minden ment, mint a karikacsapás. Egy nap csörög a telefon, és egy kedves üzlet-társ közli, hogy nem kapja meg a levelet cégünk vezetőjétől, amit pedig régen elküldött, és az ő rendszergazdájuk azt üzenete, hogy állítsák be rendesen a levelezőkiszolgálót, mert felkerültek valami fekete-listára... Elég homályos ügy. Mi történt? Hiszen eddig működött... A hiba oka az volt, hogy a jó minőségű levelezőrendszert nem elég hozzáértéssel állították be, és a rendszer elfogadott olyan levelet is továbbításra (spam), amely nem neki szólt. Ezt hívják nyitott átjárónak (angolul: open relay), és lehetővé teszi levélszemét továbbítását anélkül, hogy a küldőnek saját rendszerét le kellene terhelnie. Eljuttatja a levél tartalmát és a címlistát a rosszul beállított rendszerre, az pedig szorgalmasan kiszórja a megadott címekre. Ha ilyen rendszer-től kapunk levelet, akkor az jó eséllyel érdektelen hirdetési vagy propagandaanyag, így több olyan szervezet is létrejött, mely célja ezen rendszerek működésének akadályozása. Az alapötlet zseniális: az ellenőrző rendszer képes arra, hogy megállapítsa egy szolgáltatóról a nyitottság tényét, és ha egy rendszert nyitottnak talál, akkor azt feljegyzi. Ha valaki használni kívánja ezt a szolgáltatást, akkor időnként letölti az éppen időszzerű listát, és a listán szereplő kiszolgálóktól nem fogad el levelet. Mikor a hálózaton valaki észleli egy rendszerről, hogy nyitva áll az idegen levelek előtt, akkor jobb esetben szól a rendszer postamesterének. Ha azonban azonnal az ilyen rendszereket nyilvántartó ORBS (Open Relay Behaviour-modification System) [7.] vagy MAPS (Mail Abuse Prevention System LLC) vagy RBL (Realtime Blackhole List) [8] rendszereket értesíti, akkor az ezeket használó rendszerek nem fogadják el tőle levelet mindaddig, míg a beállítás ki nincs javítva. Az eset tanulsága, hogy egy jól tervezett és megvalósított szolgáltató-rendszert is be lehet állítani úgy, hogy a rendszer biztonsága durván sérül. Sok példát lehetne még hozni, de nincs értelme, hisz mindenki látja, mi a hiba. Egy rendszer telepítése előtt annak leírását nagyon alaposan át kell tanulmányozni, különben a beállításokban komoly hibát véthetünk. A rendszerek általában tartalmaznak valamilyen beállítórendszert is, ebben azonban az esetek nagy többségében nem érdemes megbízni. Érdemes minden beállítást leellenőrizni, hiszen ezeknek az automatáknak a tervezői nem gondolhattak minden

lehetőségre és általában a rendszer későbbi biztonsági bővítései sem állíthatók be velük. Egy ilyen automata beállító rendszer biztonsági szempontból csak a rendszer működőképességét mutatja be, a finomhangolást kézzel kell elvégezni.

Összegzés

A rendszer biztonsága tehát a tervező–fejlesztő–felhasználó szentháromság jó összmunkáján alapszik. Ha valamelyik láncszem nem elég erős, akkor a rendszer sebezhető lesz. Ha szolgáltató vagy védelmi rendszert kell választanunk, akkor célszerű meggyőződni arról, hogy egy *független* szakértőkből álló csapat mit mond az adott rendszerről. A fejlesztők gyakran elfoglaltak a saját termékükkel szemben. Ha a rendszert egy országosan, esetleg nemzetközileg elismert csoport biztonságosnak ítéli, akkor jó esély van rá, hogy az alapvető biztonsági követelményeknek megfelel. *De nem tökéletes*. Hiteles szakember nem állíthatja, hogy az általa fejlesztett rendszerben nincs hiba. Csak akkor lehet egy rendszer biztonsága tökéletes, ha az nem csinál semmit. Ennek az elvnek kitűnő bemutatása a tökéletes tűzfal leírása, melyet a szakma egyik legnagyobb embere, Marcus J. Ranum készített [10.]. Ami meglepő: a tökéletes tűzfal nem drága és a leírás alapján akár egy avatatlan is képes a telepítésére. Mindössze egyetlen gond van vele: szerszámboltban árulják és a magyar neve csípőfogó. Folytatjuk.

Hivatkozások:

- [1.] Secure UNIX Programming FAQ:
➔ <http://www.whitefang.com/sup/>
- [2.] WWW Security FAQ:
➔ <http://www.w3.org/Security/Faq/www-security-faq.html>
- [3.] Common Criteria: ➔ <http://www.commoncriteria.org/>
- [4.] Linux Intrusion Detection System: ➔ <http://www.lids.org/>
- [5.] Linux Security Audit Project: ➔ <http://lsap.org>
- [6.] OpenBSD security audit: ➔ <http://www.OpenBSD.org>
- [7.] ORBS (Open Relay Behaviour-modification System):
➔ <http://www.orbs.org/>
- [8.] MAPS (Mail Abuse Prevention System LLC) RBL (Realtime Blackhole List): ➔ <http://mail-abuse.org/rbl/>
- [9.] Marcus J. Ranum: <http://web.ranum.com/>
- [10.] Marcus J. Ranum: The ULTIMATELY Secure Firewall:
➔ <http://web.ranum.com/pubs/a1fwall/index.htm>
- [11.] ➔ <http://www.linuxvilag.hu/cikkek/2000dec/konnyu2/>



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” magyarítása csorbította a szöveg érthetőségét.

Rendszergazda a pokolból

Manapság rengeteget hallani a munkahelyi erőszakról. Most a rendszerfelügyelethez kapcsolódó feszültségekről lesz szó. A borítón látható kép ellenére a Linuxvilág nem támogatja az erőszakot.

Vajon miért is van mindig valamiféle válsághelyzetben a rendszergazda mint személy, és a rendszerfelügyelet mint foglalkozás? Miért van az, hogy a rendszerfelügyeleti munkakör mindenkinek örökös harcot és beteges tervezésű új rendszerek üzembe pofozását jelenti? Hol van az elismerés és a jutalom a rendszergazda szakértelmének, amivel működésre bírja az eszközöket? Ne nevels!

A válasz az utolsó kérdésre: „Sehol. Természetesen. Üdv neked a kiegészítés elkerüléséért folyó versenyben. Ne kísérled megváltoztatni a rendszert; kaszálj nagyot és szállj ki.”

A gazdasági szakértők mindenhol azt tanácsolják a menedzsereknek, ne próbáljanak kutyát tartani és ugatni egyszerre. Ez a cikk azonban abban a hitben íródott, hogy a vezetőség egyáltalán nem törődik azzal, hogy jó hirdetési hadjáratot bíró, de idióta termékeket zúdít a szegény, hadviselő rendszergazda nyakába. Pedig nem így kellene lennie. *Norman Solomon* ezt írja: „Az egyik legjobb útja annak, hogy elvegyük az emberek kedvét a lázadástól, ha hamis lázadások lehetőségét ajánljuk fel nekik.” És a rendszergazdáknak semmijük nincs a hamis lázadás lehetőségén kívül. A csapóján keresztül, ami pont olyan közel van, mint a legközelebbi SSH-ügyfél, a rendszergazda-kultúra kivirágzott a vezetés elnyomása alatt. Virágai azok a hírcsoportok, levelezőlisták és honlapok, amelyek mind ugyanazt a tételt hirdetik: minden rendszergazda felsőbbrendű. Ez nem meglepő egy ilyen sokféle háttérrel és képességekkel bíró embercsoporttól. A rendszergazdák mindent összevetve kitűnő írók és alkotók. Akkor mégis miért ekkora kín a rendszergazdaság? *Simon Travaglia* Pokoli Rendszergazdája a rendszergazdák népi hőse, történetei mindenhol ott vannak. Ha még nem olvastad volna a P. R. történeteit, ő egy öntörvényű ember, aki élvezetét leli mindazok kínzásában, akik kérdezni merészelnek tőle (és azokéban



is, akik nem, csak a biztonság kedvéért). Csakhogy az emberekkel való rosszindulatú szórakozás még nem teszi jobbbá a rendszert. P. R. szórakoztató, de P. R. útja csak több gondhoz, kevesebb segítséghez és még több elveszett felhasználóhoz vezet, akikkel dűlőre kell jutni. Amit P. R. képvisel, lehet, hogy hatalomnak tűnik, de az igazi hatalom a hasznos dolgok létrehozásának képességében rejlik. És ezt nem érheted el pusztán azáltal, hogy végrehajtod a „yengeelműjűek” követeléseit, és bosszút állsz, ahol csak tudsz. Nem az információs minisztériumnak dolgozol. A valós világban a rendszergazda inkább odébb sétál előlük. És a Linuxot ismerő emberek számára meglévő munkaerőpiacon elérhető a következő nehézfelfogásútól, és az őt követőtől is. Képzeld el egy céget, ami a te módszered szerint dolgozik, mert olyan rendszert építettél, ami a te módszered szerint működik, vagy egyszerűen csak működik, pont. Mert lehetséges. A Pokoli Rendszergazda csak pusztítani képes, de egy rendszergazda alkotni is.

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

Kapcsolódó címek

- A Pokoli Rendszergazda Naplója
- ➔ <http://ludens.elte.hu/~tiz/pokoli1.html>
- The Bastard Operator from Hell Complete
- ➔ <http://members.iinet.net.au/~bofh/>
- The Trouble with Dilbert by Norman Solomon
- ➔ <http://www.freespeech.org/normansolomon/dilbert/book/>



Rendszerfelügyelet

Egy csomó rendszerfelügyelettel kapcsolatos cikket gyűjtöttünk össze ebben a számban. Ha valaki komolyan gondolja, hogy nagy rendszereket akar karban tartani, itt az ideje, hogy leüljön egy Linux elé, és nekiálljon megismerkedni a segédeszközökkel!

A nagy unixos kiszolgáló helyettesítése egy szekrényre való olcsó linuxos géppel csábítóan hangzik. A váltás egyfelől ígéri a költségek csökkenését, másfelől a nagyobb megbízhatóságot és teljesítményt a kiszolgálóoldali alkalmazások számára. Ha egy magas rendelkezésre állású Linux-fürt összeállítását tervezzük, nem árt, ha megvizsgáljuk, hogy a választott fürtözési eljárás hogyan válaszol a Tim Burke cikkében részletezett négyféle hibalehetőségre.

Ha már a fürtözésnél tartunk, azoknak, akik a legjobb teljesítményt szeretnék kihozni egy linuxos fürtből, ajánlom Ibrahim Haddad cikkét a Párhuzamos Virtuális Fájlrendszeréről (PVFS). Ne a könyvelési részlegen vezessük be! A PVFS sebességéről szól ugyan, de valószínűleg nem szolgál azzal a hibátűrési szinttel, amit elvárnának tőle. Az egyik fontos dolog, amit meg kell tanulnia minden rendszergazdának: az a biztonsági mentések elkészítése. Michael O'Brien: Linuxon alapuló automatizált mentési rendszer c. cikkéből nem csak azt tudhatjuk meg, hogy hogyan menthetjük le az adatokat a linuxos rendszerekről, de azt is, hogy miként menthetünk linuxos gépen futó parancsfájl segítségével a Windowst használó PC-kről.

J. R. „Bob” Dobbs elmeséli nekünk: „A túl sok még mindig jobb, mint a kevés”. Ez nagyon találó a parancsfájlrásra. Marcel Gagné bevezet minket az Expect rejtelmibe, amelynek mottója: „Gyógyít-suk meg azokat a legyőzhetetlen interaktivitási rohamokat”. Ha sze-



retünk mindenre parancsfájl írni, akadályokba ütközhetünk, amikor egy program azt várná, hogy menürendszerben böngészve, jelszavakat beírva végezzük munkánkat. Nem kell

rögtön feladni és a feladatot kézzel megoldani! Egy Expect-parancsfájl segítségével érvényesíthetjük akaratunkat, hisz annak kell lennie, amit a Rendszergazda akar (szigorúan nagybetűvel), nem az, amit a programok!

Első pillantásra úgy gondolhatjuk, hogy a kapupasztázók és társaik betörésre valók és nem hálózatzfelügyeletre. Olvassuk el Lawrence Teo cikkét és próbáljunk ki időnként egy-egy ilyen programot saját hálózatunkon. Találhatunk feltört gépeket, amelyek részt vehetnek osztott DoS támadásokban, rosszul beállított rendszereket, vagy egyszerűen tapasztalatlan felhasználókat, akik webkiszolgáló programot telepítenek a gépükre, és közben bizalmas adataikat is megosztják. Ha valakinek jó ötlete van a rendszerfelügyelettel kapcsolatban, írjon nekünk! Lehet, hogy felkérjük egy cikk megírására, és amikor megjelenik, valaki azt fogja mondani: „Köszönöm!”.

Don Marti
Szakmai szerkesztő

© Kiskapu Kft. Minden jog fenntartva

PVFS: párhuzamos virtuális fájlrendszer linuxos fürtökhöz

Megismerkedhetünk a PVFS összetevőivel és szakszókincsével, miközben a szerző bevezet a telepítés és a beállítás rejtelmibe.

Szerző: Ibrahim F. Haddad, lásd az 56. oldalt.

Fürtellenőrzési lista

Nem minden fürtözési eljárás egyenértékű. Tanuljunk meg különbséget tenni a győztes és a vesztes rendszerek között, aszerint, hogy miként viselkednek a tervezett karbantartások, a rendszerösszeomlások, kapcsolattartási hibák vagy rendszerleállások során.

Szerző: Tim Burke, lásd a 61. oldalt.

Linuxon alapuló automatizált mentési rendszer

Védjük a pótolhatatlan adatokat automatizált mentéssel. Tanuljunk meg egy egyszerű hálózatalapú automatikus mentési rendszer beállítását. Olcsó, hatékony és megbízható.

Szerző: Michael O'Brien, lásd a 64. oldalt.

Magyarázat a kapupasztázásról és az általános pingről

A hálózat ellenőrzésére használt eszközök működésének megértése kulcsfontosságú ahhoz, hogy fényt derítsünk a segítségükkel kivitelezett támadásokra. Fedezzük fel néhányat a legjobb kapupasztázó és általános pinget felderítő eszközökből, és ismerkedjünk meg működésükkel.

Szerző Lawrence Teo, lásd a 66. oldalon.

Linux Rendszerfelügyelet: Felhasználói Kézikönyvből

Emlékezzünk rá, a kemény munka egyszer majd csak megtérül, a lustaság viszont már most meghozza gyümölcsét. Fedezzük fel, mi vár odakinn a lusta (azaz hatékonyan tevékenykedő) rendszergazdára és tanuljunk meg, hogyan hajthatjuk igába a rendszert Expect parancsfájlok segítségével.

Szerző: Marcel Gagné, lásd a 68. oldalt.

PVFS: Párhuzamos virtuális fájlrendszer linuxos telepek számára

Írásunkban a Parallel Virtual File System nevű rendszert ismertetjük, majd bemutatjuk, hogy egy vállalat miként telepítette és próbálta ki.

A hálózati fájlrendszerek használata a Unix-szerű rendszerek (például a Linux) esetében általános módszer a tárolóhely megosztására. Ezt a módszert először a Sun alkalmazta a Network File System (NFS) fejlesztése során. Ez fájlmegosztást tesz lehetővé a hálózaton. Az NFS egy kiszolgálóalapú rendszer, segítségével a távoli fájlokat a helyi fájllokhoz hasonlóan olvashatjuk, tárolhatjuk és frissíthetjük. Az NFS hamar általános szabvánnyá nőtte ki magát unixos körökben. Protokollja a Remote Procedure Call nevű adatközlési módszert használja. Az NFS esetében egy felhasználó vagy a rendszergazda egy egész fájlrendszert, illetve annak egy részét is befűzheti. A befűzött rész fájljainál is alkalmazhatjuk a szokásos hozzáférési jogokat (olvasás, írás, végrehajtás).

A rendszer népszerűségének növekedésével egyre több hálózati fájlrendszer látott napvilágot. Ezek megbízhatóságuk, biztonságuk, méretezhetőségük és sebességük tekintetében is lekörözték elődjüket. Az Ericsson Research Canada rendszerkutató részlegének munkatársaként az volt a feladat, hogy értékeljem a Linux alatt elérhető hálózati fájlrendszereket, és találgaj meg köztük azt, amelyik képes lehet saját linuxos telepeinket kiszolgálni. Jelenleg Linuxszal és a teleprendszerrel (cluster systems) kísérletezünk, és olyan linuxos telep kifejlesztésén fáradozunk, mely minden eddiginél nagyobb fokú méretezhetőséget és megbízhatóságot képes nyújtani. Egy ilyen rendszer kiépítésében elsődleges fontosságú kérdés, hogy melyik hálózati fájlrendszert kívánjuk alkalmazni. A kipróbált rendszerek között a Coda, az Intermezzo, a Global File System (GFS), a MOSIX File System (MFS) és a Parallel Virtual File System (PVFS) szerepelt. Minden lehetőség számbavétele után mi a PVFS rendszert választottuk linuxos próbatelepiünkhöz. A MOSIX csomagban (lásd a Kapcsolódó címek részben) található MOSIX fájlrendszert is felhasználtuk, főként azért, mert ez a teleprendszer kialakításához szükséges szolgáltatásokkal bővíti a Linux rendszermagot.

Írásunkban megosztjuk olvasóinkkal a rendszer első tapasztalatainkat a PVFS rendszerről. Elsőként a PVFS működési elveit ismertetjük, hogy mindenki megérthesse a PVFS szóhasználatot és a rendszer összetevőinek szerepét. Ezt követően a montreali Ericsson Systems Research laboratóriumában felállított hét Linuxból álló telep felépítéséről és beállításáról szólunk, végül pedig a PVFS erősségeit és hiányosságait mutatjuk be, ezzel is megkönnyítve a még döntés előtt állók helyzetét.

A PVFS áttekintése

A linuxos teleprendszer sokat fejlődtek az utóbbi pár évben. Az első gépek sebessége rendkívüli mértékben megnőtt, és a párhuzamos programkínálat is fejlettebbé vált. Az I/O támogatás fejlődése ennek ellenére, ahogy szokott, továbbra is a számítógépek és a programok mögött kullog. Pedig ez a terület nagymértékben befolyásolja a külső erőforrásokra támaszkodó műveletek teljesítményét.

A PVFS készítőit két cél vezérelte. Először is, mindenképpen kellett egy olyan felület, mely a linuxos telepek párhuzamos fájlrendszereivel kapcsolatos kutatások alapjául szolgálhatott. A második, hogy a teleprendszer területén egyre nagyobb igény mutatkozik a nagy teljesítményű párhuzamos fájlrendszerekre.

A PVFS jellemzői:

- Egy megosztott fájl több gép is megnyithat olvasásra és írásra, és e műveletek nagy sávszélességen zajlanak.
- Több API támogatása. Ezek között a PVFS API, a Unix/POSIX I/O API és az MPI-IO (a ROMIO-n keresztül) is szerepel.
- A hagyományos unixos eszközök (ls, cp, rm stb.) használhatók a PVFS fájllokkal.
- A Unix I/O API alá írt alkalmazások újrafordítás nélkül működnek a PVFS-sel is.
- Megbízhatóság és méretezhetőség.
- Egyszerű telepítés és használat.

A PVFS tagjai

A telep egy gépe (azaz egy tagja) több szerepet is játszhat a PVFS-rendszerben: számításokat, I/O műveleteket végezhet, vagy egyéb kezelési feladatokat láthat el. Ez utóbbi feladatra általában egy gép használatos, míg gépek egy-egy csoportja végzi a másik kettőt. Az is megvalósítható, hogy minden gép egyaránt végezzen számítási és I/O műveleteket. A PVFS démonokból és programkönyvtárból áll, az ezekben található függvények segítségével érhetjük el a fájlrendszert. Kétféle démon létezik, vezérlő és I/O démon. Legtöbbször egy egyszeres vezérlődémon fut a vezérlési feladatokat ellátó gépen, és több I/O démon az I/O gépeken. A könyvtárhívásokat a számítási, vagy ügyfélgépeken futó alkalmazások veszik igénybe, így tartanak kapcsolatot a vezérlődémonnal és az I/O démonokkal. A PVFS felépítését az 1. ábrán láthatjuk.

Vezérlő- és I/O démonok

A vezérlődémonoknak (vagy egyszerűbben: a vezérlőknek) két feladatuk van: a fájlengedélyek ellenőrzése és a PVFS fájl metaadatok karbantartása. Ez a két feladattípus a metaadatok tartalmazó fájlkor körül forog. Egy vagy több fájlrendszerhez kapcsolódó összes ilyen típusú feladatot képes egyetlen vezérlődémon ellátni. A vezérlő fájlrendszer könyvtárszerkezetének karbantartását is elvégzi. A számítási feladatokat ellátó gépeken futó programok a könyvtár tartalom listázása, fájl megnyitása, törlése és egyéb műveletek során a vezérlővel tartanak kapcsolatot. Másrésztől, az I/O démonokra csupán a PVFS fájl adatok eléréséhez és a saját, valamint az alkalmazások adatainak összehangolásához van szükség. Az alkalmazások és az I/O kiszolgálók között közvetlen kapcsolat van, így azok közvetlenül cserélhetnek adatokat olvasási és írási műveletek közben.

Az ügyfélelemek elérése

Az ügyfélelemek számára több lehetőség van egy PVFS-fájlrendszer elérésére. Először is, használhatjuk a megosztott vagy statikus programkönyvtárakat. Ehhez azonban a `pvfs_open` és más hasonló eljárásokat használó programokat kell írni. A másik megoldás a PVFS rendszermodul használata, mely teljes körű hozzáférést enged a Linux VFS-rendszerén keresztül. Ez a betölthető modul lehetővé teszi, hogy a PVFS-t a többi fájlrendszerhez hasonlóan fűzhessük be. A harmadik módszer, hogy a PVFS-sel kapott könyvtárakban lévő C burkolókat használjuk. Ezek a megnyitási, bezárási és más eljárások



www.clemson.edu/

meghívását még azelőtt „elcsípi”, mielőtt azok a rendszermagot elérnék. Ez a módszer nagyobb teljesítményt nyújt, de hátránya, hogy kevesebb rendszerrel képes hibamentesen együttműködni, ráadásul a buszok csak a glibc bizonyos támogatott változatai mellett működnek. Végül az MPI-IO csatolót is használhatjuk, amit az MPI-2 szabvány a párhuzamos alkalmazások üzenet továbbításához használ. A PVFS MPI-IO csatolója tulajdonképpen a ROMIO nevű MPI-IO megvalósítást (lásd a Kapcsolódó címek részt), és az MPI alkalmazások számára lehetővé teszi az MPI-IO lehetőségeinek kihasználását a PVFS elérésekor. Arról is gondoskodik, hogy az MPI kód a ROMIO által támogatott más párhuzamos fájlrendszerekkel együtt tudjon működni.

Telepítési környezet

Az Ericsson montreali laborjában felállított próbakörnyezet hét, lemez nélküli Pentium gépből állt, ezek mindegyikében 256 MB memória volt. Ezek a gépek először a gyártótól kapott eszközökkel elkészített mini rendszermagot töltik be, flashlemezről. Ezt követően IP-címet kapnak és egy DHCP- és TFTP-kiszolgálóként működő linuxos gépről letöltenek egy RAM-lemezt. Ugyanez a gép a hét másik számára NFS-kiszolgálóként is működik, lemezerületet biztosítva számukra. Amikor a PVFS kipróbálása mellett döntöttünk, néhány PC-re volt szükségünk, hogy többségük I/O gépként, egy pedig vezérlőgépként működjön. A PC1 a vezérlő, a PC2, PC3 és PC4 pedig (35 GB tárhellyel) I/O gép lett. A rendszer tehát az alábbi gépekből állt:

- hét, lemez nélküli ügyfélgép,
- egy vezérlő,
- három I/O gép.

A telepítés lépései

Bár a PVFS fejlesztői RPM-eket adnak bármilyen típusú géphez, mi a forráskód újrafordítása mellett döntöttünk. Ezt azért tettük, hogy minél jobban testre tudjuk szabni a rendszert a lemez nélküli ügyfelek számára. A dolog egyszerűen zajlott a PVFS tarball csomag segítségével. A vezérlő és I/O gépekhez az RPM csomagokat, a Red Hat 6.2 változatot és a 2.2.14-5.0 rendszermagot használtuk. A lemez nélküli gépek ugyanezen rendszermag egyszerűsített változatát futtatták.

A vezérlő beállítása

A PVFS felállításának első lépése a PVFS vezérlőt tartalmazó RPM csomag letöltése és telepítése volt. A PVFS alapértelmezés szerint a /usr/pvfs könyvtárba kerül. Az önműködő telepítési folyamat befejeztével létre kell hoznunk a beállításfájlokat. A PVFS működéséhez ebből kettőre van szükség: a „pvfsdir” írja le a PVFS könyvtárát, az „iodtab” pedig az I/O démonok helyét határozza meg. Ezeket az mkiodtab parancsfájl rendszergazdaként futtatásával állíthatjuk elő:

```
[root@pc1 /root]# /usr/pvfs/bin/mkiodtab
```

Az *listán* a PVFS iodtab beállításait láthatjuk. A .pvfsdir fájl a gyökérkönyvtárban jön létre.

Az iodtab telepítése

```
Enter the root directory: /pvfs
Enter the user id of directory: root
Enter the group id of directory: root
Enter the mode of the root directory: 777
Enter the hostname that will run the manager:
pc1
Searching for host...success
Enter the port number on the host for manager:
(Port number 3000 is the default) 3000
Enter the I/O nodes: (can use form node1,
node2, ...
)or nodename{#-#,#,#})
pc2,pc3,pc4
Searching for hosts...success I/O nodes:
pc2,pc3,pc4
Enter the port number for the iods:
(Port number 7000 is the default) 7000
Done!
[root@pc1 /root]#
```

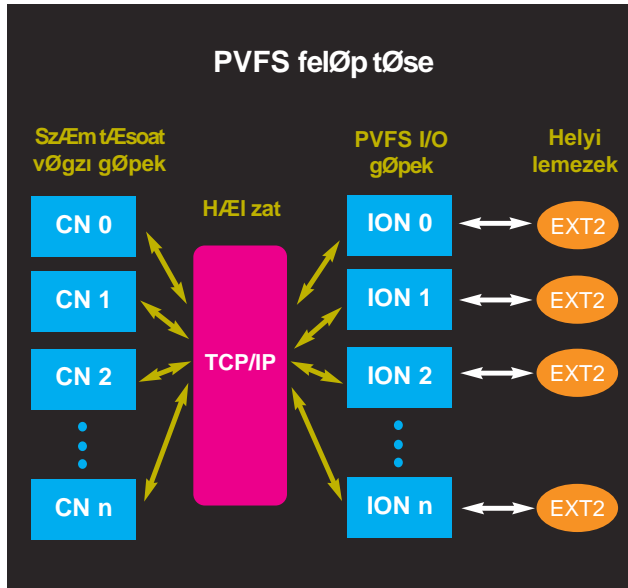
Amikor az mkiodtab-ot a vezérlőn (PC1) futtattuk, az közölte, hogy nem találja az I/O gépeket. Kiderült, hogy elfelejtettük ezeket beállítani a /etc/hosts fájlban. A fájl frissítése és az mkiodtab ismételt elindítása után minden rendben lezajlott. Az mkiodtab a /pvfs könyvtárban létrehozott egy iodtab nevű fájlt, melyben az I/O gépek voltak felsorolva:

```
-----/pvfs/.iodtab-----
pc2:7000
pc3:7000
pc4:7000
-----
```

Az I/O démonhoz az ügyfelek a 7000-es kapun csatlakoznak. A mkiodtab futtatása után így indítottuk el a PVFS vezérlőként szolgáló PC1-et:

```
% /usr/pvfs/bin/mgr
% /usr/pvfs/bin/enablemgr
```

Az enablemgr futtatása a vezérlőn lehetővé teszi, hogy a gép újraindításakor a démon magától elinduljon, tehát nincs szükség kézi indításra. Az enablemgr parancsot egyszer kell indítani a megfelelő hivatkozások létrehozásához.



1. ábra

Az I/O gépek beállítása

Az I/O gépek telepítése ugyanilyen egyszerű. Először az RPM csomagokat telepítettük, majd minden I/O démonot így indítottunk el:

```
% /usr/pvfs/bin/iod
% /usr/pvfs/bin/enableiod
```

Az enableiod futtatása az I/O gépeken biztosítja, hogy a gépek újraindítása során a démonok elindulnak, tehát nincs szükség kézi indításra. Az enableiod parancsot egyszer kell indítani a megfelelő hivatkozások létrehozásához.

Az I/O démonok beállításfájlja a /etc/iod.conf, ebből tudják meg, hogy az adatokat hol kell tárolniuk. Az RPM magától létrehozza ezt a fájlt, és az I/O démonok adatait a /pvfs_data könyvtárba tereli:

```
% mkdir /pvfs_data
```

A lemez nélküli gépek beállítása a számítási feladatok elvégzésére

Az ügyfélgépek telepítése már kicsivel összetettebb volt, hiszen (mint fentebb említettük) a memórialemezeken a lehető legkevesebb helyet akartunk lefoglalni. Az ügyfelek telepítéséhez az alábbi fájlokat mindenképp használnunk kellett:

```
—A számításokat végző gépekre telepített fájlok—
/etc/pvfstab
/usr/local/pvfs/pvfsd
/usr/local/pvfs/pvfs.o
/usr/local/pvfs/mount.pvfs
/usr/local/pvfs/libpvfs.so.1.4
```

Az ügyfelek a /etc/pvfstab fájlt a vezérlő és a PVFS fájlok helyének meghatározására használják, formátuma nagyon hasonlít a /etc/fstab fájlhoz. Mi az alábbiakat helyeztük el benne:

```
—————/etc/pvfstab—————
pc1:/pvfs /pvfs pvfs port=3000 0 0
```

Az előbbi sor az alábbiakat határozza meg:

- A vezérlő neve: PC1.
- A vezérlő által a metaadatok tárolására használt könyvtár: /pvfs.
- A PVFS fájlrendszer befűzési pontja az ügyfeleken: /pvfs.
- A vezérlő kapuszáma: 3000.

A PVFS démon neve /usr/pvfs/bin/pvfsd, és a rendszermag modullal együttműködve biztosítja a fájlrendszerrel való kapcsolatot. A démon ugyanazokat a PVFS könyvtárhívásokat használja, mint bármely más alkalmazás, de a magmodul által érhető formára alakítja azokat, így a hívások csak a PVFS-hez fordított alkalmazások számára érhetőek el. Hasonló módszert találunk a Coda esetében is (lásd Kapcsolódó címek). Ott a felhasználói szintű démon a Coda rendszermagködjével együttműködve éri el a fájlrendszert.

A /usr/pvfs/bin/mount.pvfs a PVFS-hez kapott különleges befűző parancs, mellyel az ügyfelek a PVFS fájlrendszert egy helyi pontra fűzhetik be. E gépekhez egy kis héjprogramcskát hoztunk létre (/etc/rc.d/rc.pvfs), mely a gépek indításakor fut le, és azokat számítási műveleteket végző gépekként állítja be. Az rc.pvfs tartalma a következő:

```
—————/etc/rc.d/rc.pvfs—————
#!/bin/sh
/bin/mknod /dev/pvfsd c 60 0
/sbin/inssmod /usr/pvfs/bin/pvfs.o
/usr/pvfs/bin/pvfsd
/usr/pvfs/bin/mount.pvfs pc1:/pvfs /mnt/pvfs
```

A parancsfájl a /dev-ben egy, a pvfsd által használt eszközelemet hoz létre. Betölti a PVFS modult, elindítja a PVFS démonot, és a PVFS fájlrendszert befűzi a /mnt/pvfs helyre.

Ahogy már korábban is említettük, bármelyik I/O gép vagy vezérlőgép szolgálhat számítási műveleteket végző gépként is. Ezért a PVFS ügyfelet tartalmazó RPM-et telepítettük minden I/O gépre, mivel ezeken a gépeken van elég szabad terület. A /etc/pvfstab és a /etc/rc.d/rc.pvfs fájlokat ugyanúgy állítottuk be, mint a lemez nélküli gépeknél. Most tehát a lemez nélküli ügyfelek és az I/O gépek ugyanúgy érhetik el a fájlrendszert.

A telepítés kipróbálása

A fenti lépések elvégzése után a PVFS fájlrendszer fájljait minden gépről elérhettük. A gépekre telepített memórialemez az Apache webkiszolgáló és a mozgókép folyamok kiszolgálójaként működő Real Server egy részét is tartalmazta. A ZDNet.com WebBench nevű rendszerével forgalmat irányítottunk a gépekre, az Apache és a Real Server beállításfájljait pedig úgy módosítottuk, hogy gyökérdokumentumaikat a PVFS fájlrendszerben keressék. Ilyen körülmények között minden gép saját IP-címmel bíró önálló webkiszolgálóként és multimédiás kérélmeket teljesítő Real kiszolgálóként működhetett. A PVFS fájlrendszerben tehát nagyméretű MP3, rm és ehhez hasonló fájlokat helyezhettünk el.

Együttműködés más fájlrendszerekkel

Mivel bizonyos alkalmazások (sajátos elérési módszereik következtében) jobban teljesítenek egy adott fájlrendszerben, fontos, hogy a PVFS zavartalanul használható legyen más fájlrendszerekkel együtt. A PVFS rendszer kifogástalanul működött az ugyanazon környezetben található JFS, NFS, SFS, sőt, MOSIX fájlrendszerekkel is. Ez a helyes kis felállítás nagyméretű kérélmeket (például MP3 letöltés) is hibátlanul képes volt kiszolgálni. A MOSIX fájlrendszert a MOSIX-szal használtuk, és így a folyamatokat mindig a legmegfelelőbb gépre tudtuk irányítani.

A PVFS általában az ext2 fájlrendszere épül. A Linux elkövetkező

fájlrendszerei azonban még fejlettebbek lesznek: a gépek és programok hibáitól azzal igyekeznek megvédeni a felhasználót, hogy folyamatosan naplózzák a fájlmodosításokat, lehetőség szerint egy másik merevlemezre. Ha az elsődleges fájlrendszer megsérül, a másolat segítségével tetszőleges mélységben visszaléphetünk a félbemaradt feladatok sorában (ezt leginkább a több programban megtalálható Visszavonás szolgáltatáshoz tudnánk hasonlítani).

A következő lépés, hogy a PVFS teljesítményét az ext3 és GFS fájlrendszereken is kipróbáljuk. Ezt a kísérletet az új telepen végezzük majd el (lásd alább).

A telepítés méretezése

A PVFS-hez hasonló fájlrendszerek kiválasztásánál nem elhanyagolható szempont az sem, hogy milyen jól méretezhető rendszert kapunk. Elsőre felmerül, hogy mekkora gondot okoz az egyetlen példányban futó vezérlő, hiszen ez erős terhelés mellett gyorsan szűk keresztmetszetté válhat. Azonban a vezérlő nem végez semmilyen olvasási vagy írási műveletet, ezeket közvetlenül az ügyfelek és az I/O gépek in-

tézik. A vezérlő csak akkor van nagymértékben leterhelve, ha gyors egymásutánban sok fájlt hozunk létre, nyitunk meg vagy zárunk be. Mivel a méretezhetőséget nem tudtuk megfelelően kipróbálni, nekünk felépítünk egy új PVFS telepet, 16 PIII 500 MHz-es gépből, ezek mindegyikében 512 MB memória lesz. Nyolc géphez 18 gigabájtos SCSI merevlemez csatlakozik (RAID 1 és RAID 5 vegyesen). A tervezett telepítésben egy vezérlő, hét I/O gép és 14 ügyfél lesz (az I/O gépek ügyfélként is működnek majd). Ez a telep lehetővé teszi számunkra, hogy megvizsgáljuk a PVFS és a saját alkalmazásaink együttműködését. Emellett a PVFS teljesítményét összehasonlíthatjuk a többi fájlrendszer (NFS stb.) teljesítményével is. Más PVFS rendszereken végzett próbák bizonyítják, hogy a PVFS akár 64 elemből álló telepeken is megállja a helyét. (Lásd a PVFS honlapján található PVFS: A Parallel File System for Linux Clusters című cikket.)

A PVFS előnyei

A PVFS telepítése és beállítása egyszerű, a csomagban kapott telepítési útmutató pedig megkönnyíti a rendszergazdák dolgát. A rendszer jókora teljesítményt nyújt a nagy adatforgalom-igényű párhuzamos vagy terjesztett alkalmazások számára. A létező alkalmazásokkal együttműködik, tehát azokat nem kell a PVFS miatt módosítani. A PVFS fejlesztése levelezési listákon keresztül kényelmesen figyelemmel kísérhető.

A PVFS sebezhetősége

A PVFS jelenleg nem tartalmaz sem többszörözött adattárolási lehetőséget (redundanciát), sem pedig hibakezelést. Az ügyfélgépek számának növekedésével vezérlés szintjén is előfordulhatnak szűk metsetek. A PVFS alatt bizonyos korlátozásokat is el kell viselnünk. Ezek főleg a TCP/IP protokoll miatt jelentkeznek (például egy időben korlátozott számú csatlakozópontot – socket – használhatunk, vagy ott van a TCP/IP-nél megszokott forgalomtöbblet). A PVFS-nek elég egyszerű biztonsági szolgáltatásai vannak, ezeket a védett telephálózatokhoz tervezték. Jelenleg a Linux két gigabájtos fájlméret-korlátozását sem képes túllépni a PVFS.

A PVFS előnyeit legjobban a következő alkalmazások használhatják ki:

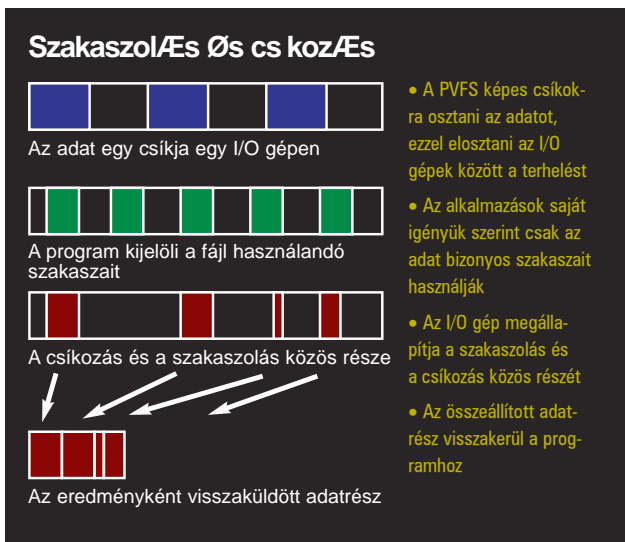
- Nagy adatforgalmat bonyolító alkalmazások (tudományos számítások, multimédiás adatok valós idejű feldolgozása).
- A párhuzamos alkalmazások – hiszen az adatokat egy időben elérő ügyfelek számának növekedésével a sávszélességnek növekednie kell.

Az alábbi alkalmazásokat viszont kifejezetten hátráltatja a PVFS:

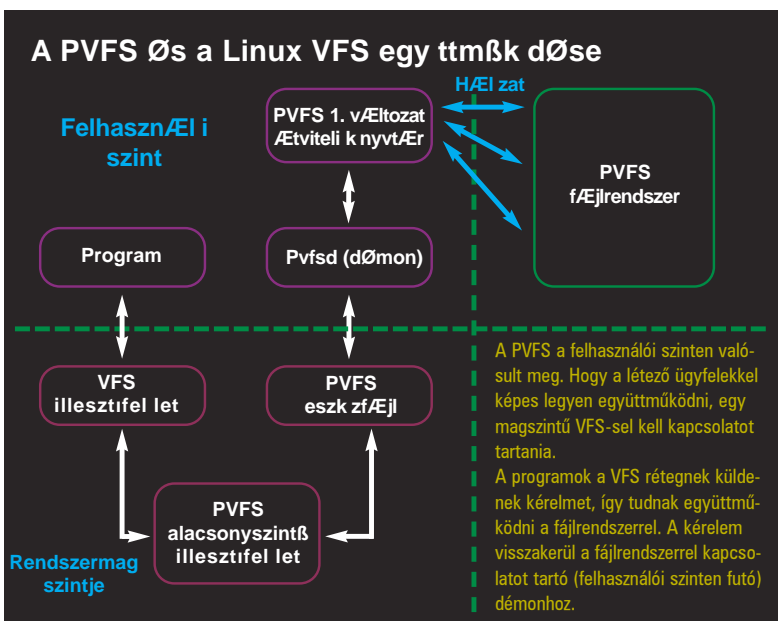
- Sok kis méretű, rövid követési időközzel érkező kérelmek, például statikus HTML oldalak (az ilyen kérelmek hatalmas többletforgalmat okoznak).
- Hosszú idejű tárolást, vagy többszörözött kiszolgálókat, gépfürtözést igénylő alkalmazások – a PVFS önmagában nem képes a többszörözött (redundáns) adattárolás kezelésére.

PVFS programok készítése

Ahogy azt már korábban is megjegyeztük, a már létező alkalmazásaink a PVFS-t a magmodulon keresztül, vagy pedig a programkönyvtár-burkoló segítségével érhetik el. Ehhez a felhasználó szempontjából semmiféle módosításra nincs szükség. Ha azonban a párhuzamos alkalmazásokból a lehető legtöbb szeretnénk kihozni, akkor az alkalmazásokban valamivel összetettebb felületet kell használnunk. Erre mindjárt két módszer is kínálkozik. Használhatunk PVFS könyvtárhívásokat; ennek segítségével fejlettebb lehetőségeket is kiaknázhathunk (például az I/O



2. ábra



3. ábra

gépek számának beállítása stb.). Lehetőségünk van a fájlokat szakaszokra osztani, valamint szétszórni több gépre (csíkokra vágni), így az olvasási műveletek csak a fájl egy-egy részére, illetve csak egy-egy I/O gépre vonatkoznak (lásd a 2. és 3. ábrát). Erről a PVFS használati útmutatójában találunk leírást.

Az MPI-IO a PVFS programok írásának legcélszerűbb módja. Ez a PVFS-t további lehetőségekkel, többek között csoportos fájlműveletekkel és kétlépéses adatovábbítással bővíti. A illesztőfelületről az MPI-2 szabvány leírásában olvashatunk.

Biztonsági kérdések

Ahogy már korábban is említettük, a PVFS jelenleg nem nyújt semmiféle biztonsági szolgáltatást. Ez abból következik, hogy a rendszert nem nyilvános telephálózatokhoz tervezték, ahol a kapcsolódó ügyfelekben teljességgel megbízhatunk. Az ügyfélkapcsolatokra semmilyen korlátozás nem vonatkozik, és a felhasználók azonosításához kulcsokat vagy titkosítást sem használhatunk. Az ügyfélgépek által szolgáltatott UID adatokban a rendszer teljesen megbízik, és az NFS-hez hasonlóan ezek segítségével állítja be a jogosultságokat és a fájlok tulajdonosait.

A PVFS jövője

A PVFS minden bizonnyal rengeteg újítással gazdagodik a jövőben. A jelenlegi PVFS-változat komoly fejlesztéseken megy keresztül, ezek célja, hogy a csomag méretezhetőbb legyen. A több TCP/IP csatlakozópont támogatása és a 64 bites fájl mérettárolás kérdése is hamar megoldódik. Ezzel a PVFS számára megnyílik az út a több száz vagy ezer ezer gépből álló teleprendszer felé.

A PVFS ezzel egy időben teljes átszerkesztésen megy át: a készítőik tanultak a korábbi változatok hibáiból. Az új változatra egy darabig még várni kell, de a fejlesztés már gőzerővel folyik.

A következő változatokban támogatott szolgáltatások:

- Érzékelés beosztás, mely a PVFS számára lehetővé teszi, hogy a rendszer állapota és az alkalmazások terheltsége alapján hozzon döntéseket.
- Több hálózati rendszer modulszintű támogatása. Így a fájlrendszer nem függ többé a TCP/IP-től, hanem a jövőben megjelenő fejlettebb üzenettovábbítási protollokat is használhatja.
- Több tárolási módszer modulszintű támogatása. Ezek segítségével az I/O démonok a helyi adatokat sokféleképpen elérhetik (például nyers vagy aszinkron I/O).
- Több vezérlőgép támogatása.
- A munkaadatok és a metaadatok többszörözött (redundáns) tárolása a rendszerösszeomlások esetére.
- Fejlesztések a Unix-megfelelőség területén.
- Fejlettebb lehetőségek az adatterjesztés és -megjelenítés területén.

Jó néhány osztott fájlrendszer kipróbálása után a PVFS-t választottuk nagy adatforgalmat bonyolító alkalmazásainkhoz. A PVFS jelenleg nem támogat semmilyen biztonsági szolgáltatást, de a kutatás és fejlesztés folyik, mi pedig reménykedünk. Hiszünk abban, hogy ha a PVFS képes lesz kezelni a másolatkészítést, helyet kapnak benne biztonsági szolgáltatások, akkor a linuxos telepek legtekélyesebb fájl-



A PVFS honlapja

rendszerévé válhat. A dolgok jelenlegi állása szerint a rendszer olyan telepeken állja meg a helyét, ahol a hatékonyság és a teljesítmény fontosabb, mint a magas rendelkezésre állás.

Kellemes élmény volt a PVFS-sel való ismerkedés. Aki olyan osztott fájlrendszert keres, amely képes hatalmas adatforgalom lebonyolítására, annak én jó szívvel ajánlom a PVFS-t. Ingyenesen hozzáférhető a GPL szabályozás alapján, és a Clemson Egyetem honlapjáról tölthető le (lásd Kapcsolódó címek).

Köszönetnyilvánítás

A szerző megköszöni a PVFS fejlesztői, név szerint *Robert Ross* (Mathematics and Computer Science Division, Argonne National Lab) *Philip Carns* és *Walt Ligon* (Parallel Architecture Research Lab, Clemson University) segítségét. A NASA Goddard Space Flight Center és az Argonne National Lab részlegei szintén számos hasznos adatot adtak. Köszönjük az Ericsson Research Canada rendszerkutató részlegének a rendelkezésre bocsátott gépparkot, valamint hozzájárulását a cikk közzléséhez.

Philip Carns (pcarns@hubcap.clemson.edu) a Clemson Egyetem Parallel Architecture Research laboratóriumának végzős hallgatója.

Robert Moss (ross@mcs.anl.gov) az Argonne Nemzeti Laboratórium matematikai és számítástudományi részlegében dolgozik. Doktori fokozatát decemberben szerezte meg.



Ibrahim F. Haddad (ibrahim.haddad@lmc.ericsson.se) az Ericsson Research Canada rendszerkutató részlegének munkatársa, és hamarosan megszerzi doktori fokozatát a montreali Concordia Egyetemen.

Kapcsolódó címek

MOSIX: ➔ <http://www.mosix.org/>

ROMIO: ➔ <http://www-unix.mcs.anl.gov/romio/>

Coda: ➔ <http://www.coda.cs.cmu.edu/>

Parallel Virtual File System:
➔ <http://www.parl.clemson.edu/pvfs/>

Üzembiztos kiszolgálótelepek

Az utóbbi időben egyre több állandó rendelkezésre állást ígérő szolgáltatás jelenik meg a piacon. Célszerű utánanézni, hogy igényeinknek melyik megoldás felel meg a legjobban.



© Kiskapu Kft. Minden jog fenntartva

A jelenlegi éles versenyhelyzetben különösen gyakran halljuk „az idő pénz” kifejezést. Az üzleti adatok hálózati tárolása és folyamatos elérhetősége képezi a vállalati kiszolgálók lényegét. Legyen szó háttérben működő adatbázisokról, levelek és felhasználók nyilvántartásáról vagy hálózati fájlrendszerekről (NFS), az adattárolási rendszer üzemzavara gyászos következményekkel járhat. A leginkább költségkímélő megoldásnak egy hibatűrő géptelep (más szóval kiszolgálófűrt) felállítása tűnik. A fogalom valójában több kiszolgáló összekapcsolását jelenti, melyek bármelyike képes a többi adatbázis- vagy alkalmazás-kiszolgáló feladatait azonnal átvenni. Ha a telep egyik tagja meghibásodna, a többi gép átveszi a leállt kiszolgáló által működtetett szolgáltatásokat. Ez az átvétel szerencsés esetben úgy történik, hogy a felhasználó észre sem veszi.

Egy jellemző megoldás például, hogy több számítógép csatlakozik egy megosztott adattárolóhoz, ami általában SCSI vagy FibreChannel adatsínre kapcsolt lemezeket jelent. A vállalati felhasználásra készült telepeket eredetileg kizárólag a jól ismert cégek, például a Digital, a HP és az IBM gyártottak. A Linux-alapú, olcsó gépeken is működő rendszerek csak nemrégiben váltak elérhetővé.

A világhálón körülnézve bárki találhat többféle, Linuxra épülő teleprendszert is, ezek legtöbbször igen vonzóknak tűnnek – legalábbis papíron. Az ígéretekben általában az szerepel, hogy a rendszer szempillanatnyi idő alatt elvégzi a váltást, legyen szó akárhány elemből álló telepről és tetszőleges számú szolgáltatásról. Könnyen előfordulhat, hogy nem jó megoldásszállítót választunk. Az igazság az, hogy nem minden magas rendelkezésre állást biztosító rendszer növeli adataink megbízhatóságát és elérhetőségét. Éppen ellenkezőleg – egy nem megfelelő választással értékes fájlrendszeinket és adatbázisainkat tulajdonképpen kiszolgáltatjuk a rossz szándékú behatolóknak. Néhány gyártó igyekszik elhallgatni ezt, mások esetében pedig csak hosszas kutatómunka után lelhetjük fel e tényeket a szerződésben. Jómagam több mint hét éve dolgozom a Unix/Linux-alapú, állandó rendelkezésre állást biztosító rendszerek területén, és jó néhány termék tündöklésének, majd csúfos bukásának voltam tanúja. Felháborítónak tartom, hogy egyes rendszerek hirdeteiseiben nagy mellénnyel hivatkoznak olyan szolgáltatásokra, melyeket nyilvánvalóan képtelenek nyújtani. Itt a felhasználó értékes adatai kerülnek veszélybe, ráadásul a hozzá nem értő cégek botrányai rossz fényt vetnek az egész szakmára. Hosszú évek tapasztalatát összegzi az a négy pontból álló lista, mely segítségével felmérhetjük, hogy az adott termék képes-e megfelelni az általunk támasztott követelményeknek. Valójában ezek a szempontok nemcsak a Unix/Linux felületre igazak, hanem bármilyen más operációs rendszer és géptípus esetében is alkalmazhatók. Tehát mielőtt egy fillért is kiadnánk egy magát tökéletesnek hirdető szolgáltatásért, győződjünk meg arról, hogy használata valóban megvédi rendszerünket és adatainkat az alábbi négy helyzetben:

1. Tervezett karbantartás és leállítás.
2. Rendszerösszeomlás.
3. Kapcsolattartási zavar.
4. Rendszerlefégyás.

Mind a négy helyzetet nemcsak részletesen tárgyaljuk, és a jellemző

hibákról is szót ejtünk, előtte azonban engedjék meg némi magyarázat arról, hogy mit jelent az adatok épsége. Ennek egyik alapköve, hogy az adatok pontosak és frissek legyenek. Ez eddig elég egyszerűen hangzik, nemde? Egy számítógéptelepen az adatok épsége bír a legnagyobb fontossággal, sorrendben még a folyamatos elérhetőséget is megelőzi. A példák tanulmányozásával bizonyára mindenki érteni fogja, hogy miről beszélek. Az 1. ábra egy kételemű telepet mutat be, az A és B jelű elemeket egy megosztott SCSI sín köti össze az 1. lemezzel (az egyszerűség kedvéért használtam két elemet, a példa természetesen tetszőleges elemszám esetén is alkalmazható).

A legtöbb operációs rendszerben a lemezen lévő adatokat fájlrendszeren keresztül érthetjük el. Általában a fájlrendszer befűzi a tárolólemezt, majd létrehozza a felhasználói kapcsolatokat. A nagyobb teljesítmény eléréseért a fájlrendszerek saját adataik friss másolatait a memóriában tárolják. Ebből következik, hogy adataink legfrissebb változata (amit a példában az A elem szolgál ki), az A gép memóriájában tárolt, illetve a lemezen lévő adatokból áll össze.

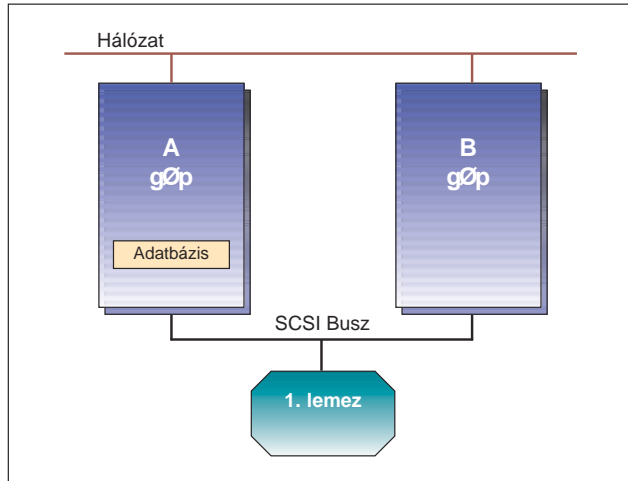
Nézzük most meg a B nevű gépet. Ha B ugyanezt a fájlrendszert próbálná meg befűzni és használni, a fájlrendszer valódi adatait az A memóriájában tárolt, a B memóriájában lévő, valamint a lemezen elhelyezkedő adatok együttese képezi. Ahhoz, hogy ez a rendszer hibamentesen működjön, olyan fájlrendszerre van szükség, mely képes az egyszerre több helyen tárolt adatok kezelésére. Az olyan rendszereket, ahol több gép is befűzheti ugyanazt a fájlrendszert, telepíjendőknek hívjuk. Csupán néhány Unix-változat tartalmaz ilyen jellegű támogatást, linuxos környezetben pedig nem tudok ilyenről (bár valami már készülődik – látogassunk csak el a GFS projekt <http://www.gfs.lcse.umn.edu/> honlapjára).

A telepíjendők hiányában mi történik akkor, ha több elem egyszerre kísérel meg elérni ugyanazt a fájlrendszert? A következő esetek képzelhetők el:

- **Pontatlan adatok.** Tegyük fel, hogy a Las Vegas-i utunk jól sikerült, és száz dollárt szeretnénk a számlánkra tenni. Mondjuk, hogy a befizetést az A gép kezelte, melynek eredményeképpen a korábbi 25 dolláros egyenlegünk 125 dollárra emelkedett, ezt az A (mint legfrissebb adatot) mindjárt el is helyezi memóriájában lévő gyorstárbán. Ekkor, mondjuk hazarepülünk, és kiderül, hogy autónkat csak ötven dollár ellenében vehetjük át a parkolóházból. Ezt a műveletet most B végzi, mely a lemezhez fordulva még mindig 25 dollárt lát, és a „Nincsen elég pénz a számlán” üzenetet küldi. Mindez azért fordulhatott elő, mert a valódi egyenleget (125 dollár) az A gép a saját memóriájában tárolja. A telepek felépítésénél tehát föl kell tennünk a kérdést: milyen kárt okozhat a vállalat számára, ha téves adat adódik továbbításra?
- **Rendszerleállás.** A felhasználói adatok (például a számlaegyenleg) mellett a fájlrendszerek saját értékeiket is a lemezen tárolják, amelyek a felhasználói adatok elrendezésének módját határozzák meg (ezeket legjobban egy tartalomjegyzékhez lehetne hasonlítani). A teljesítmény megtartása érdekében ezek az értékek szintén bekerülnek a memóriában lévő gyorstárba. A fájlrendszerek hamar összezavarodnak, ha ezek az értékek megsérülnek, és egy-egy ilyen hiba általában teljesen

kiborítja őket (mi ezt csak rendszerösszeomlásnak hívjuk). Egy valódi telepíjlrendszer hiányában, ha egy adatot két gép próbál meg elérni egyszerre, az kavarodásokhoz vezet a tartalomjegyzékben, és ebből származnak a „legtakarosabb” rendszerösszeomlások.

Ha egy fájlrendszer valamelyik adata vagy tartalomjegyzéke megsérül, az adatsérülést is okozhat, azt pedig csak a legutolsó biztonsági mentésről történő visszaállítással háríthatjuk el. (Mindenkori gyakran és rendszeresen végez biztonsági mentést, ugye? A 64. oldalon olvashatunk egy praktikus megoldást is.) A baj csak az, hogy a műveletek (és az ezekkel



1. ábra Kétegéses telep egy megosztott SCSI lemezzel

járó adatváltozások) sokkal gyakrabban végrehajtódnak, mint a biztonsági mentések, és így az adatsérülés helyrehozása gyakran napokba is beletelik, „pedig a hirdetésben másodperceket emlegettek...”

A fenti elv, mely szerint az elemeknek összehangoltan kell elérniük a fájlrendszereket, az adatbázisokra is érvényes. A legtöbb adatbázis-megoldás nem teszi lehetővé az egyes elemek számára, hogy ugyanazt az erőforrást egyidejűleg érjék el. Figyelemre méltó kivételként az Oracle Parallel Servert (a linuxos változat már készül) és az Informix Extended Parallel Servert említenék meg.

Az eddig vázoltak lényege, hogy a választott teleprendszernek lehetővé kell tennie, hogy egy fájlrendszert vagy adatbázist egyidejűleg csak egyetlen gép módosíthasson – ez így elég egyszerűen hangzik, de olyan teleprendszer található, amely ezt az idő száz százalékában képes megvalósítani, már sokkal nehezebb lesz. Most lépünk tovább és vizsgáljuk meg, hogy mindezek mennyiben érintik a fent említett négy helyzetet.

Tervezett karbantartás

A magas rendelkezésre állású telepek egyik legszebb (és legjobban mellőzött) tulajdonsága, hogy bármelyik gépet eltávolíthatjuk anélkül, hogy ez zavarná a gépen futó szolgáltatásokat (hiszen a többiek átveszik annak feladatait). Így például egyszerűen telepíthetjük valamelyik program legfrissebb változatát, vagy memóriával bővíthetjük a gépet, és eközben a telep egyetlen szolgáltatása sem szünetel, egy tízedmásodpercig sem. Majdnem minden teleprendszer képes így kezelni a tervezett karbantartással járó leállásokat.

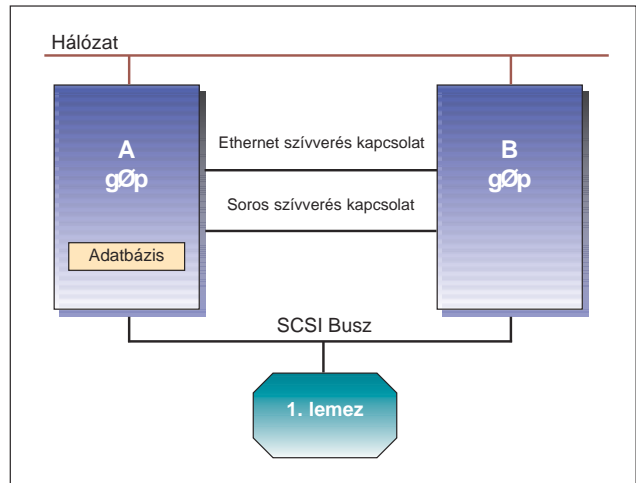
Rendszerösszeomlás

Ha valaki azt hiszi, hogy vannak fagyásbiztos operációs rendszerek, az csak szóljon nekem, majd én mutatok neki egy-két érdekes dolgot. Nézzünk szembe a tényekkel: a rendszerösszeomlás az élet része. A dolgunk „mindössze” az, hogy gyakoriságukat a lehető legkisebbre csökkentsük. Ha a telep egy gépe leáll, a többi erről azonnal tudomást szerez és elosztja a kiesett gépen futó szolgáltatásokat egymás között. Rendszerösszeomlás esetén elméletileg minden teleprendszer képes

a meghibásodott gépek feladatait szétosztani. És ez így nagyon jól van – tehát ezek alapján bármelyik rendszert választhatnánk? Na, azért ennyire ne szaladjunk előre. Az alábbi szempontok alapján szépen kiválaszthatjuk a valóban értékes megoldásokat a használhatatlanoktól.

Kapcsolattartási zavar

Általában a magas rendelkezésre állású telepek gépei folyamatosan kapnak egymásról adatokat (cluster interconnects). Régebben a nagy telepgyártók saját alkatrészekkel oldották meg e kapcsolatok kezelését. Ez a módszer megbízható, de természetesen nagyon drága és az adott céghez köt bennünket. Olcsóbb megoldást találunk számos más rendszerben: ezeknél a hagyományos hálózaton (Ethernet vagy PPP) keresztül történik az adatcsere. Az ilyen megoldásoknál az egyes gépek rendszeresen váltanak üzenetet, és a kapott válaszok alapján döntenek arról, hogy a kérdéses gép működik-e vagy sem. Az adatforgalom ezen típusát „szívverésnek” (heartbeat) nevezzük.



2. ábra Kétegéses telep Ethernet és soros kapcsolattal

A szívverés-alapú telepek leggyakoribb gondja a kapcsolattartási zavar. Ez akkor következik be, amikor a telep tagjai működnek, de nem képesek kapcsolatba lépni egymással. Nézzük például a 2. ábrán látható Etherneten és soros kábellel csatlakozó kételemű telepet.

Tegyük fel, hogy üzembe helyeztük a telepet, majd a hűtőgégre Las Vegasba utaztunk. Ezalatt otthon a szokásos nagytakarítás közben valaki a seprűvel kirántja a hálózati vezetékét. Ilyenkor a két gépnek el kell döntenie, hogy a magas rendelkezésre állás szem előtt tartása mellett mit tegyen. A gépek nem képesek elérni egymást, ezért önállóan kell meghozniuk ezt a döntést. Az alábbiakban felvázolunk néhány elvet. Ezeket számos megvásárolható teleprendszer alkalmazza (sajnos):

- **A lehető legrosszabb eset feltételezése.** Az A gép tudja, hogy ő szolgálja ki az adatbázist, de nem tud B állapotáról, tehát A folytatja az adatbázis elérését. Eközben B nem képes kapcsolatba lépni A-val, így feltételezi, hogy A kiesett. B ekkor szintén megpróbálja kiszolgálni az adatbázist, ami a már említett kettős hozzáféréshez és rendszerösszeomláshoz vezet (bármilyen hihetetlen, de sok rendszer működik így!).
- **A lehető legjobb eset feltételezése.** Egy, az egész telepre kiható áramkimaradás után A és B egyszerre indul el. Egyik elem sem képes meghatározni a másik állapotát, és a biztonság kedvéért egyikük sem kezdi el az adatbázis kiszolgálását. Mindkettőt azt feltételezi ugyanis, hogy a másik működik (csupán kapcsolattartási zavar lépett fel), és a kettős elérés megelőzésére inkább kiszáll a játékból. A végeredmény: mindkét gép üresjáratban várakozik, a szolgáltatások pedig nem működnek. Ez sem tűnik kellemes helyzetnek, de az átmeneti leállás még mindig sokkal jobb, mint ha adataink sérülnének egy rendszerösszeomlás miatt.

Más hibák is tűnhetnek kapcsolattartási zavarnak, például:

- egy hálózati kártya tönkremegy,
- a gépeket összekötő útválasztó elromlik,
- egy Ethernet-vezeték meghibásodik.

E kapcsolattartási zavarok elkerüléséhez alkalmazzunk többféle kapcsolatot a gépek között. Beállíthatjuk a rendszert, hogy az elemek több kapcsolaton, vagy Ethernet- és soros kapcsolaton egyszerre beszélgethessenek egymással. Ugyanígy az is megvalósítható, hogy a hálózati kapcsolatok külön jelelosztókon és útválasztókon keresztül haladjanak, vagy alkalmazzunk kizárólag soros kapcsolatokat.

A legtöbb teleprendszer lehetővé teszi, hogy többszörös kapcsolatokat alakítsunk ki, ezzel is csökkentve a kapcsolattartási zavarok lehetőségét. Ha a választott rendszer nem képes erre, akkor jobban tesszük, ha azonnal egy másikra állunk át.

Rendszerlefagyás

Ez a legveszélyesebb hiba, ami egy teleprendszerrel megtörténhet. Mindenki látott már olyat, hogy egy gép valamilyen különös okból egyszerűen felfüggesztette működését, és olyankor csak a Reset gomb vagy a ki-, majd újra bekapcsolás segített. Szerencsére a dolog ritkán fordul elő. Legalább ennyire furcsa az is, amikor a lefagyott gép egyszer csak ismét működni kezd. Talán már olyan esettel is akadt dolgunk, amikor egy gép megállt, majd kis idő elteltével ismét válaszolt a kérelmekre, tehát működött tovább. Ez bármelyik operációs rendszerrel megtörténhet. A dolog annyiban érinti a teleprendszereket, hogy tisztában kell lennünk azzal, miként viselkedik a telep egy esetleges lefagyáskor, majd ismételt elindulásakor. Nézzünk most egy példát, mellyel talán sikerül rávilágítanunk erre a nagyon fontos tényezőre. Tegyük fel, hogy az A gép nem válaszol. Mivel előre gondoltunk a kapcsolattartási zavarokból származó hibákra, ezért a telep gépei két Ethernet- és egy soros kábellel csatlakoznak egymáshoz. Nos, lefagyás esetén teljesen mindegy, hogy egy vagy ötven kapcsolat van a gépek között – a „szívverést” ezek egyike sem képes továbbítani, hiszen a gond magával a géppel van. B észleli, hogy A a három csatorna egyikén sem válaszol az üzenetekre, és feltételezi, hogy A leállt. Ekkor B megpróbálja átvenni A feladatait: befűzi a fájlrendszereket és kiszolgálja az adatbázist. De ekkor A ismét működni kezd és folytatja az adatbázissal félbehagyott munkát. Ugye innen már ismerős a helyzet: az erőforrást a két gép egyszerre próbálja meg elérni, és ennek eredményeképpen ismét egy csinos kis rendszerösszeomlás következik be. Ez utóbbi helyzetben bizonyíthatja a telep, hogy valóban képes teljes mértékben ellátni feladatát. Az adatok épségének megvédéséért a szolgáltatás(ok) átvétele előtt minden elemnek meg kell győződnie arról, hogy a leállt elem nem módosíthatja a fájlrendszert vagy az adatbázist. Ezt a módszert I/O korlátozásnak vagy I/O elrekesztésnek (I/O Fencing, I/O Barrier) hívjuk. Néhány gyártó e kellemetlen gondot egyszerűen azzal intézi el, hogy nem törődik vele, mondván, úgysem túl gyakori helyzetről van szó. Szerencsére ez így is van, de ne felejtjük el, hogy az adatok biztonságának megőrzéséért a lehető legvalószínűtlenebb helyzetben is azonnal cselekednie kell a rendszernek.

Összefoglalás

Aki minden körülmények között az adatok épségét tartja a legfontosabb szempontnak, az a teleprendszer felépítése előtt győződjön meg arról, hogy a választott megoldás mind a négy fent említett tárgykörre képes megfelelő választ adni. Soha ne felejtjük el, a hálózat legfontosabb feladata, hogy az adatok minden pillanatban elérhetőek és helyesek legyenek. Ha az adatok épségének megőrzésén takarékoskodunk, az hosszabb leállási időket és elvesztett ügyfeleket jelenthet – ezek bármelyike akár beláthatatlan következményekkel is járhat.

Tim Burke a Mission Critical Linux, Inc. telepmérnöke.

➔ <http://www.burke@missioncriticallinux.com/>

Szókincstár

Fontosnak tartjuk, hogy kialakuljon végre egy egységes szóhasználat az informatika, és ezen belül a Linux világában. Ennek érdekében igyekszünk minden hónapban összegyűjteni néhány kérdéses szót. Ha bárkinek ötlete, véleménye, kérdése van bármelyik szóval, vagy egyáltalán a témával kapcsolatban, kérjük, írja meg nekünk a szokincstar@linuxvilag.hu címre.

Szótárba

alias	másodnév
gateway	átjáró
hard link	közvetlen (direkt) hivatkozás
hot-swappable	menet/használat közben cserélhető
hub	(hálózati) jelelosztó, elosztó
implementation	megvalósítás
integrity	épség (adaté, hivatkozásé)
interface	illesztőfelület, illesztő, csatolófelület, kezelőfelület, felület
kernel	rendszermag
link	hivatkozás
option	lehetőség
package manager	csomagkezelő
packet filtering	csomagszűrés
platform	felület, géptípus
port	kapu
process	folymat
router	útválasztó
script, shell script	parancsállomány, héjprogram
service	szolgáltatás
session	munkafolyamat
shell	héj
switch	hálózati kapcsolat
symbolic (soft) link	közvetett (átteteles) hivatkozás
thread	szál

Kérdéses szavak

Az itt következő szavakra találtunk ugyan magyar megfelelőt, de vagy mi magunk sem vagyunk vele teljesen megelégedve, vagy nem tudjuk, hogy minden szempontból, minden szövegkörnyezetben megfelelő-e.

concurrent	együttfutó, egyidejű, versenyhelyzetben lévő
console	konzol
cracker	kalóz, betörő
exploit	akna (a biztonság témakörében)
firewall	tűzfal
hacker	betyár (A hálózatokhoz és a gépekhez értő jóindulatú szakember, aki képes felderíteni egy rendszer biztonsági réseit.)
named pipe	nevesített cső/csővezeték
redundancy	többszörös adattárolás, adattöbbszörözés

Ötletet várunk

Ezek a szavak kemény diók. Legtöbbjüket használjuk jövevényszóként, de reméljük, hogy találunk helyett magyar szót. Ha bárkinek van jó ötlete, szeretettel várjuk!

IP multicast, unicast

record	bejegyzés, sor (néha)
recursive	
transaction	
socket	illesztőpont, csatlakozó, foglalát, aljzat
compile, link, build	fordít, összeállít, elkészít

Linuxon alapuló automatizált mentési rendszer

Hogyan gondoskodjunk a biztonsági mentésekről egyszerűen és ötletesen?

Hajlamosak vagyunk azt hinni, hogy a számítógépen tárolt adatok biztonságban vannak. Ez a hiedelem lehet az alapja annak, hogy bár fontos adatokat dolgozunk fel vagy tárolunk a számítógépen, az adatokat mégsem mentjük rendszeresen. A rendszergazdák természetesen tisztában vannak a megbízható mentések fontosságával, de elképzelhető, hogy azok az emberek, akikkel együtt kell dolgozniuk, nem fordítanak erre elegendő figyelmet, és az ő adataik elvesztésekor is a rendszergazdákat veszik elő.

Egy tudományos kutatócsoportban dolgozom. Munkahelyünk korszerű, gyakorlatilag minden adatfeldolgozást Windows 95-öt futtató számítógépekkel végzünk. Tömören összefoglalva az egész tevékenységünk nem más, mint a számítógépekben tárolt adatok feldolgozása. Az adatvesztés több százezer forintba kerül, különösen, ha az adatokat előállító munkatársak fizetését is figyelembe vesszük. A pótolhatatlan adatok elvesztésének megakadályozására kidolgoztam egy automatizált hálózati mentési rendszert. A költsége elenyésző volt, ugyanis egy paragonon heverő 486/66 számítógépet és egy 3 GB tárhelyű merevlemez használtam fel hozzá, így összesen alig több mint harmincezer forintba került az egész. Több különféle rendszert kipróbáltam az elmúlt két évben, Windows 95-tel kezdtem, végül is egy gyors és hatékony Linux-rendszerrel kötöttem ki. Ezt a rendszert könnyű kiépíteni, olcsó, hatékony és megbízható. Feltéve, hogy rendelkezünk egy hálózatra kapcsolt Linuxot futtató számítógéppel, az itt leírtak segítségével rövid idő alatt kiépíthető az automatizált mentési rendszer.

Szükséges eszközök

Az automatizált mentési rendszerhez szükséges összes eszköz megtalálható a legtöbb Linux-változatban. Az egyik a Samba, egy kiváló nyílt forráskódú csomag, amely lehetővé teszi, hogy Unix-alapú rendszerek Windows-alapú rendszerekkel a TCP/IP protokollon keresztül fájlmegosztást végezzenek. A linuxos változat tartalmazza az smbmount segédprogramot, amely kihasználva a rendszer mag egyedi álló smb fájlrendszer támogatását, lehetővé teszi, hogy bármely windowsos gépen elhelyezkedő könyvtárat a Linux fájlrendszerébe fűzzünk, és úgy használjuk az ott lévő fájlokat, mintha a linuxos gép merevlemezén lennének. Így a mentést végző program (frissítési módomban) még azelőtt meg tudja állapítani a windowsos gépen lévő fájlról, hogy kell-e menteni, mielőtt átmozgatná a hálózaton keresztül. Ez jelentősen csökkenti a hálózati forgalmat, a processzor és a merevlemez terhelését.

Rengeteg biztonsági másolatot készítő program található Linux alatt, például a tar, a bzip2, vagy az egyszerű cp parancs. Választásom mégsem ezekre, hanem a nyílt forráskódú Info-ZIP eszközeire esett. Ezek kicsik, gyorsak, a legtöbb Linux-változat tartalmazza, más felületeken is elérhetők, és kiválóan együttműködnek a Windowszal is. Rádásul az Info-ZIP programok tömörítési képességeinek köszönhetően a tároló-fájl mérete jelentősen csökken.

Előzetes lépések

A windowsos gépeken létre kell hozni azokat a hálózati megosztásokat (merevlemez vagy egy alkönyvtár), amelyeket menteni szeretnénk. Ha a fájlmegosztás nincs engedélyezve, be kell állítanunk a Vezérlőpult Hálózatok részében. Ezután a Windows Intézőben kattintsunk jobb egérgombbal arra a meghajtóra vagy könyvtárra, ame-

1. lista A DATA könyvtár mentése

```
#!/bin/sh
# Mentés a megadott SHARE megosztásról
SHARE="//higgins/d_drive";
DATADIR="DATA";
USERNAME="notneeded"
PASSWD="mjdomo";
BACKUPFILE="higgins_data.zip";
BACKUPDRIVE="/dev/hdc1";
BACKUPMP="/mnt/backups/";
SMBMP="/mnt/smb/";
export PASSWD;
echo 'go';
smbmount $SMBMP;
smbmount $$SHARE $$SMBMP -N -n $USERNAME;
if (ls $$SMBMP$DATADIR) then
  echo "$SHARE mentése folyamatban";
  mount -t vfat $BACKUPDRIVE $BACKUPMP;
  cd $SMBMP;
  zip -r -u -v $BACKUPMP$BACKUPFILE
  $DATADIR;
  cd /;
  umount $BACKUPDRIVE;
fi;
smbmount $SMBMP;
```

lyet meg szeretnénk osztani, és válasszuk a Megosztás pontot. Javasolom, hogy „csak olvasható” megosztásokat hozzunk létre, hogy a betörők ne tehessenek kárt a fájlokban akkor sem, ha valahogy megszerették a jelszavakat. Jegyezzük meg jól a megosztások neveit. Jó ötlet beírni a windowsos gépek netbios nevét, DNS-nevét és IP-címét a Linux /etc/hosts fájljába (ahogy az a /etc/hosts megjegyzéseiben le van írva), esetleg a névkiszolgálóba, különösen akkor, ha a számítógépek több különböző alhálózatban vannak.

Ha ez megvan, elő kell készíteni a linuxos gépet az adatok elérésére és tárolására. Hozunk létre egy könyvtárat, ahová a windowsos megosztásokat fogjuk befűzni: `mkdir /mnt/smb`. Ezután el kell döntenünk, hogy hová tesszük a lementett adatokat.

Én egy különálló 1 gigabájtos vfat (Windows) lemezterületen tartom a lementett fájlokat, ez nincs a rendszerbe illesztve, csak a mentés idején. Így a fájlok a lehető legjobban védettek az áramkimaradásból eredő károkkal szemben, a meghajtó kivehető a linuxos számítógépből és áttehető egy windowsos gépbe az adatok helyreállításának céljából. Ennek megfelelően egy /mnt/backups nevű befűzési pontot is létrehoztam.

Parancsállományok

A parancsállomány egy szövegfájl, amely olyan parancsokat tartalmaz, amelyeket az ember általában a parancssorban ad ki. Bonyolult feladatokat ismétlődő jelleggel elvégeztethetünk velük. A parancsállomány létrehozható bármelyik szerkesztőben. A fájl elkészülte után `chmod u+x` segítségével tehetjük futtathatóvá.

2. lista Példák windowsos gépekhez

```
# ----- VÁLTOZÓK a Rick nevű géphez -----
SHARE="//rick/rick_c";
DATADIR="Data";
USERNAME="notneeded"
PASSWD="icepick";
BACKUPFILE="rick_data.zip";
BACKUPDRIVE="/dev/hdc1";
BACKUPMP="/mnt/backups/";
SMBMP="/mnt/smb/";
# -----
# ----- VÁLTOZÓK a TC nevű géphez -----
SHARE="//tc/data";
DATADIR="*";
USERNAME="obrien"
PASSWD="flyme";
BACKUPFILE="tc_data.zip";
BACKUPDRIVE="/dev/hdc1";
BACKUPMP="/mnt/backups/";
SMBMP="/mnt/smb/";
# -----
```

Az 1. lista egy parancsállományt mutat be, amely a Higgins nevű gép d_drive megosztásáról a DATA könyvtárat menti le. A parancsállomány neve root/backup/higgins.

Az első, megjegyzésnek tűnő sor tulajdonképpen arra utasítja a számítógépet, hogy a basht használja a parancsállomány futtatásához. Ezután következnek a mentéskor használt változók. Ha a parancsállomány elején található változókban tároljuk az adott gépre érvényes adatokat, akkor a fájl elejét átírva könnyen hozhatunk létre újabb parancsállományokat a különböző gépekhez. A 2. lista egy másik változókészletet mutat be egy Windows 98-at futtató gépre (a gép neve Rick és a C: meghajtó van megosztva) és egy Windows NT-t futtató gépre (a gép neve TC és a „data” könyvtár van megosztva). Ne feledjük, hogy Windows NT esetén meg kell adni a felhasználói nevet és a hozzá tartozó jelszót! A többi sor végzi el a tulajdonképeni munkát. Az <<export paswd>> parancs a jelszót egy környezeti változóba rakja, amelyet az smbmount program automatikusan beolvas. Az smbmount programot azért hívjuk meg, hogy leválasszuk az esetlegesen befűzött SMB megosztásokat. (Ha semmi nincs ott, akkor az smbmount egy értelmetlen hibáüzenetet ad, és a parancsállomány tovább fut.) Az smbmount megkísérel befűzni a távoli megosztást. A -N kapcsoló arra utasítja, hogy ne kérje el a jelszót, hanem a <<paswd>> környezeti változóból vegye. A -n kapcsolóval adható át a felhasználói név.

Egy if szerkezet megvizsgálja, hogy a megadott mentendő fájlok léteznek-e, mielőtt a mentés megkezdődne, hiszen elképzelhető, hogy a hálózat nem működik, vagy a távoli számítógép ki van kapcsolva. Ilyenkor a parancsállomány a befűzési pont újrafelszabadítása után leáll.

Ha a linuxos gép el tudja érni a távoli fájlokat, a mentést a zip parancs végzi. A -r kapcsoló a rekurzívítást biztosítja, azaz a zip az adatokat tartalmazó könyvtár minden alkönyvtárát is végignézi. A -u kapcsoló állítja be a frissítési üzemmódot, amely csak akkor adja hozzá a fájlt a tárolófájlhoz, ha az még nincs benne, vagy újabb, mint ami a tárolófájlban található. A -v kapcsoló bőbeszédű üzemmódra készíti a zip programot, minden feldolgozott fájl nevét kiírja – ez hasznos segítség a hibakeresés során.

Miután az összes számítógéphez elkészítettük a mentést végző parancsállományt, készíthetünk egy főparancsállományt (nevezzük masternek), amely egymás után végrehajtja az összes mentést végző parancsállományt. Az én főparancsállományom a 3. listán olvashatóhoz hasonló.

3. lista Főparancsállomány

```
#!/bin/sh
# A mentést végző parancsállományok meghívása
/root/backup/higgins
/root/backup/rick
/root/backup/tc
```

A rendszer beindítása

A parancsállományok megírása után tegyünk egy, a főparancsállományra mutató közvetett hivatkozást (symbolic link) a /etc/cron.d alkönyvtárainak egyikébe, hogy a számítógép automatikusan elvégezze a mentést. Én az alábbi parancsral értem el, hogy hetente lefusson:

```
ln -s /root/backup/master /etc/cron.d/weekly/master
```

Szükség esetén napi mentést is végezhetünk, hiszen a tárolóprogramok frissítési üzemmódja nagymértékben csökkenti az erőforrásigényt. A mentést végző parancsállomány első futtatása azonban nagy sávszélességet és sok processzoridőt igényel. A leghelyesebb, ha ezt a feladatot először kézzel indítjuk, vagy az at parancs segítségével éjjel.

Figyelmeztetések

1. Minden jelszót tartalmazó parancsállományt rejtünk el mindenki elől. Ehhez használjuk a `chmod go-r` parancsot.
2. A nagyon fontos adatok védelmére külön biztonsági intézkedéseket kell tennünk, elkerülendő, hogy ipari kémek betörjenek gépünkre, és ellopják az egy helyre összegyűjtött adatokat.
3. Az smbmount program kissé különbözik az egyes Linux-változatokban. Ezért, ha az itt megadott parancsállományok nem működnek, nézzük végig az smbmount parancs leírását (man smbmount).
4. A Windows-felhasználókat meg kell tanítani, hogy adataikat egy központi helyen tárolják, például az „adatok” könyvtárban, ahelyett hogy mindenfelé szétszórják a fájljaikat. Sok ember lusta áthelyezni az adatait egy területre, annak ellenére, hogy ez csak öt másodpercet venne igénybe. Elképzelhető, hogy nekünk kell átraknunk a fájlokat, amíg ők is megtanulják, hogy egy központi könyvtárban tartsák az adataikat. Jegyezzük meg, hogy ezek a lusta emberek jelentik a legnagyobb veszélyt egy szervezetben az adatbiztonság tekintetében, hiszen saját adataik mentésével sem törődnek.
5. A merevlemez nagyon célszerű tárolóeszköz a pótolhatatlan adatok számára. Az én rendszeremen a tárolófájlok kevesebb mint 400 megabájtot foglalnak, ami 1,5 GB adatnak felel meg. Megfontolandó esetleg egy nagy tárhelyű eltávolítható eszközre író meghajtó beszerzése. Így időszakonként kimenthetjük a tárolófájlokat a merevlemezről az eltávolítható eszközre, és hazavihetjük, ezzel védekezve a gép fizikai megsemmisülése vagy ellopása ellen.

Összefoglalás

Ha több, hálózatba kapcsolt számítógépről kell lementeni a pótolhatatlan adatokat, a Linux-alapú hálózati mentési rendszer olcsó, megbízható, könnyen telepíthető és könnyen bővíthető megoldást jelent a feladatra. Ha egy lemez meghibásodik, a befektetett egyórás munka sok pénz és időt takaríthat meg csoportunknak vagy vállalatunknak azáltal, hogy az adatok nem vesznek el. Nálam például a Pentium 150-es gép több év alatt felhalmozott fontos adatot tárol három épületben és két külön alhálózaton lévő nyolc különböző számítógépről. Alig egy perc alatt tudok egy új gépet hozzáadni a rendszerhez, hála az állományok elején lévő változóknak.

Michael O'Brien (mobrien@unm.edu) az Új-mexikói egyetem végzős hallgatója, optikát tanul. A számítógép egyszerűre a kedvtelése és segítőtársa a munkában. Szabadidejében egy kis számítógépszobát felügyel.

Kapupásztázás és általános ping

Ismerkedjünk meg a betörők által használt két leggyakoribb hálózatvizsgáló eljárással.

A nagyobb hálózatok rendszergazdái általában azt állítják, hogy az ő hálózatukon már volt betörési kísérlet. Mivel a betörő-programokból egyre több van, ráadásul mind népszerűbbek, már nem csak a nagyobb hálózatokat fenyegeti veszély. A hálózatvizsgáló programok önmagukban nem veszélyesek ugyan, mégis figyelniük kell rájuk, hiszen ezek használata egyértelműen betörési kísérletet jelez. Ahogy a mondás is tartja, jobb félni, mint megjedni.

Kapupásztázás

A hálózat vizsgálatainak legnépszerűbb módja a kapupásztázás (port scan). Ez egy olyan eljárás, aminek segítségével a betolakodó felderítheti a kiszemelt számítógépen futó szolgáltatásokat. Ezen adatok birtokában a támadó megkísérelheti felhasználni a futó szolgáltatások egyikének-másikának ismert hiányosságait. Például ha a betolakodó azt látja, hogy a 143-as kapu (az IMAP kapuja) nyitva van, akkor kideríti, hogy az IMAP melyik változata fut a számítógépen. Ha ez a változat sebezhető, akkor képes rendszergazdai jogosultságot szerezni egy akna segítségével (ezek olyan programok, amelyekkel ismert biztonsági hibákat, réseket lehet kihasználni). A kapupásztázás tulajdonképpen nagyon egyszerű dolog. Csupán kapcsolódni kell a kiszemelt gép kapuira, és megfigyelni, hogy melyik válaszol és melyik nem. Egy egyszerűbb kapuvizsgáló programot egy jó programozó Javában vagy Perlben 15 perc alatt elkészít. Azonban ezt a fajta kapuvizsgálót a kiszemelt számítógép operációs rendszere is könnyen észreveszi. Az 1. listán lévő naplófájlban (a /var/log/messages egy részlete) egy kapuvizsgáló program nyomai láthatók. Látható, hogy a kapcsolódások különböző szolgáltatásokhoz kevesebb, mint három másodperc alatt történtek. Mivel ez könnyen felismerhető, a behatolók manapság már nem is használják a kapuvizsgálókat ezt a fajtáját. A kapupásztázás másik, rejtettebb változata a „félnyitott” SYN-pásztázás. Itt is a kapuhoz kapcsolódik a program, de még a kapcsolat teljes felépítése előtt megszakítja a kapcsolatot (innen a „félnyitott” elnevezés). Mivel a teljes kapcsolat soha nem jön létre, az operációs rendszer nem is naplózza ezeket kísérleteket. A működési elv biztosan jobban érthető, ha egy kicsit elmélyedünk a TCP/IP protokoll belső működésében. A rendes TCP/IP-kapcsolatban, a két eszköz összekapcsolódásához egy háromlépéses üdvözlésre (three-way handshake) van szükség. A félnyitott SYN-pásztázásnál a program csak arra kíváncsi, hogy működik-e az adott kapu, ezért a harmadik lépés előtt megszakítja a kapcsolatot.

Ismerkedjünk meg napjaink legnépszerűbb és legerősebb hálózatvizsgáló programjával, melynek neve Nmap (Network Mapper). Az Nmap képes a kapupásztázás mindkét említett típusára, sőt, képes más típusú vizsgálatra is, de erre majd később visszatérünk. A 2. listán láthatjuk, hogy milyen jelentést készít. Felmerül a kérdés, hogy ha a kapupásztázást rejtve végzik, akkor vajon hogyan lehet felderíteni. A jó hír az, hogy

a legtöbb kapupásztázást felderíthetjük az erre a célra készült eszközökkel. A Solar Designer kifejlesztette a scanlogd nevű programot, mely démonként a háttérből figyeli a hálózati csatoló kapuit. Ha a scanlogd kapupásztázást érzékel, ezt jelzi a rendszernaplóban. A 3. listán látható, hogy miként jelez a scanlogd egy felderített kapupásztázást. Természetesen más olyan eszközök, illetve programok is léteznek, amelyekkel a pásztázás ugyanilyen eredményesen felderíthető. Most nem ezekkel foglalkozunk, de az érdeklődők a fejezet végén találhatják. Kapcsolódó címek között tallózva számos hasznos címet találunk. Ajánlom még a tcplogd programot, mely egy remekül beállítható, TCP-kapupásztázást felderítő alkalmazás: meghatározható a naplózni kívánt csomagok típusa az elárasztás (flooding) megelőzésére, valamint a megbízható helyek és kiszolgálók is beállíthatók.

Általános ping

Az általános pingek (ping sweeps) a hálózatvizsgálók másik típusát képviselik. A behatoló ICMP ECHO csomagokat küld a hálózat számítógépeinek (általában meghatározott IP-címtartományban), és figyeli, hogy melyik válaszol. (A ping eredetileg hangutánzó szó.) Így állapítja meg, hogy melyik gép működik, és melyik nem. Ez kicsit olyan, mint ha hajnali háromkor kopogtatnánk a szomszédok ajtaján, és azt figyelniük, hogy ki alszik, és ki nem (ezzel a módszerrel azért csak csínján bánjunk!). Ha a betolakodó megtudja, hogy melyik számítógép üzemel, arra fog összpontosítani, és munkához lát. Az fping egy olyan program, amely kitűnően használható általános pingeléshez. A program a kapott IP-címeknek küldözget ping-csomagokat. A hagyományos pinggel szemben az fping csak egy csomagot küld az első címre, majd azonnal a következő címet veszi, így forogtatja körbe a kapott listát. A 4. listán egy olyan egyszerű Perl parancsfájl láthatunk, amely C osztályú IP-címeket hoz létre (a példánkban 192.168.0.1-től 192.168.0.20-ig). Az 5. listán megfigyelhetjük, hogy a Perl parancsfájl miként működik együtt az fpinggel a meghatározott IP-címtartományban üzemelő számítógépek felderítésében. A -a kapcsolóval csak a válaszoló gépeket jeleníthetjük meg (enélkül az fping az elérhetetlen gépeket is megjeleníti). Mind a kapupásztázást, mind az általános pinget fel lehet deríteni egyedi eszközök használatával. Az ippl egy olyan IP-protokollnaplózó alkalmazás, amely rögzíti a TCP-, az UDP- és az ICMP-csomagokat. Hasonlóan a scanlogdhez, a háttérben csücsül, és onnan figyeli a csomagokat. A 6. listán látható, hogy az ippl miként jelzi az elkapott pingsomagokat. Óvatosnak kell lenniük az ippl használatával,

1. lista A kapupásztázás nyomai a naplóban

```
Jul 18 02:42:25 target sshd[2370]: log: Connection from 192.168.0.1 port 2107
Jul 18 02:42:25 target sshd[2370]: fatal: Did not receive ident string.
Jul 18 02:42:25 target wu.ftpd[2369]: connect from root@attacker
Jul 18 02:42:25 target in.telnetd[2368]: connect from root@attacker
Jul 18 02:42:26 target imapd[2366]: connect from root@attacker
Jul 18 02:42:26 target in.pop3d[2367]: connect from root@attacker
Jul 18 02:42:26 target ftpd[2369]: FTP session closed
Jul 18 02:42:26 target telnetd[2368]: tloop: read: Broken pipe
Jul 18 02:42:28 target in.fingerd[2365]: connect from root@attacker
```

2. lista Az Nmap eredménye

```

root@attacker# nmap -sS -O target.example.com

Starting nmap V. 2.53 by fyodor@insecure.org/
(http://www.insecure.org/nmap/)
Interesting ports on target.example.com
(192.168.0.2):
(The 1507 ports scanned but not shown below
are
in state: closed)
Port      State Service
21/tcp    open  ftp
22/tcp    open  ssh
23/tcp    open  telnet
25/tcp    open  smtp
37/tcp    open  time
79/tcp    open  finger
110/tcp   open  pop-3
111/tcp   open  sunrpc
113/tcp   open  auth
143/tcp   open  imap2
515/tcp   open  printer
901/tcp   open  samba-swat
2049/tcp  open  nfs
6000/tcp  open  X11
7100/tcp  open  font-service

TCP Sequence Prediction:
  Class=random positive increments
  Difficulty=2135704 (Good luck!)
Remote operating system guess:
Linux 2.1.122 - 2.2.14

Nmap run completed - 1 IP address
(1 host up) scanned in 3 seconds

```

3. lista A scanlogd segítségével felderített kapupásztázás

```

Jul 18 02:56:22 target scanlogd: 192.168.0.1
to 192.168.0.2 ports 38681, 18, 1127, 1486, 966,
1493, 682, 401, 211, ..., f??pauxy, TOS 00 \
@02:56:21

```

mert ha egy nagy forgalmú hálózaton használjuk, az ippl naplófájl (amit általában a /var/log/ippl/ elérési úton találhatunk meg) nagyon gyorsan megtelhet.

Van néhány más lehetőség is az ippl-ben, sajnos ezekbe még nem tudtam belemélyedni. Valami azonban felkeltette az érdeklődésemet, és ez nem más, mint a pingd. Ez olyan felhasználó oldali démon, amely kiszolgálószinten kezeli az ICMP-forgalmat. Nagyon hasznos a pingd-ben, hogy együttműködik a TCP-burkolókkal, így a hozzáférést szabályozó állományban (/etc/hosts.allow és /etc/hosts.deny) jól beállítható, hogy ki pingelheti a gépet és ki nem.

Egyéb hálózatvizsgáló eljárások

A kapupásztázás és az általános ping csak kettő a hálózatvizsgáló eljárások közül. A jelenlegi módszerek száma jelentős, amellet folyamatosan fejlesztés alatt állnak, ez pedig azt jelenti, hogy a rendszergazdák egyre érdekesebb kísérletekkel találkozhatnak a jövőben. A rengeteg egyéb típusú kísérlet közül egy párat ismer még az Nmap, ilyen pél-

4. lista Egy egyszerű Perl parancsfájl

```

#!/usr/bin/perl
$networkid = "192.168.0";
$begin = 1;
$end = 20;
for ($hostid = $begin; $hostid <= $end;
$hostid++) {
    print "$networkid.$hostid\n";
}

```

5. lista Az fpinggel futtatott általános ping

```

root@attacker# perl gen.pl | fping -a
192.168.0.17
192.168.0.15
192.168.0.12
192.168.0.10
192.168.0.1

```

6. lista Az ippl segítségével felderített pingkérelem

```

Jul 19 04:19:37 ICMP message type echo \
request from
attacker.example.com [192.168.0.1] \
(192.168.0.1->192.168.0.2)

```

dál a csalétek pásztázás (Decoy Scan). Ennél a pásztázásnál a gép különböző IP-címekről kap kérelmet, így nem tudja megállapítani, hogy melyik gépről is indult ki a támadás. Ennek fő oka a rendszergazda megzavarása. A csalétek pásztázás (Decoy Scan) mellett, az Nmap képes megállapítani a kiszemelt számítógép operációs rendszerének típusát és annak változatát is. Ez a „TCP/IP-verem ujjlenyomat” (TCP/IP stack fingerprinting) nevű eljárás segítségével lehetséges. Ahogy azt a 2. listán is láthattuk, az Nmap felismerte, hogy a célszámítógép a Linux 2.1.122–2.2.14 közötti változatát futtatta (a valóságban a 2.2.12-es változat futott a gépen). Írásom elkészítésekor az Nmap 2.53-as 465 különböző operációs rendszert (különböző változatokkal), útválasztót és egyéb eszközt képes felismerni és azonosítani.

Állandó figyelem

Remélem, hogy ez a kis írás hasznos lesz a két leggyakoribb hálózatvizsgáló eljárás működésének és felderítési módjainak megértéséhez. A megfelelő biztonságot természetesen folyamatosan szeretnénk fenntartani. A hálózatvizsgáló programokból egyre több lesz, újabb és újabb biztonsági lyukakat fedeznek fel, és gyakorlatilag naponta szerepelnek a hírekben, úgyhogy érdemes naprakész adatok birtokában lenni. Ezért ajánlatos feliratkozni egy biztonsági témájú levelezőlistára (például BUGTRAQ), vagy a hasonló témájú hírcsoportokat és weblapokat sűrűn látogatni.

Nmap: ➔ <http://www.insecure.org/nmap>

fping: ➔ <http://www.stanford.edu/~schemers/docs/fping/>



Lawrence Teo (lawrenceteo@usa.net) jelenleg a behatolási módszerek felderítéséről folytat kutatásokat a tudományos fokozat megszerzéséhez Ausztráliában, a Monash egyetemen. Amikor nem unixos gépekkel vacakol, egy jó japán éttermet keresve kóborol Melbourne utcáin.

Linux rendszerfelügyelet: felhasználói kézikönyv

Kivonat a mi francia szakácsunk hamarosan megjelenő könyvéből.

Jövőre jelenteti meg új könyvem az Addison Wesley Longman kiadó, címe Linux rendszerfelügyelet: felhasználói kézikönyv. Felkértek, hogy adjak egy kis ízelítőt a hamarosan megjelenő könyvből.

Először hadd mutassam be az alaphelyzetet. Sötét, viharos éjszaka van (ezt mindig is le akartam írni), a magányos rendszergazda már régen szerette volna befejezni a munkáját és hazamenni.

Ez a részlet a 15. fejezetből való, amit én egyszerűen csak „alkotó lustaság”-ra kereszteltem. Néhány évvel ezelőtt kisebb-nagyobb társaságban szívesen emlegettem, hogy én úgy gondolom, hogy az alkotó lustaság egy csodálatos dolog. Mindig is csodálatra méltónak találtam a lusta embereket, mert mindig megtalálják a feladat megoldásának legegyszerűbb és legelegánsabb módját. Idézném a tudományos-fantasztikus irodalom nagyját, *Robert A. Heinleint*, aki egyszer a következő gondolatot vetette papírra: „A lusta ember a dolgokat mindig a legegyszerűbben intézi el.”

Ebben a fejezetben az automatizálás eszközeit fogom bemutatni. Már elmúlt este 11 óra. A pizza már a múlté, a kávéházban a kávé is elfogyott. Fények. Kamera. Csapó, tessék...

Az interaktív folyamatok automatizálása: expect

Először is úgy tűnik, igencsak fogytán vagy a szerencsédnek, ha az emberi beavatkozást szeretnéd automatizálni. Ezek azok a dolgok, amihez egy emberre is szükség van: választani egy menüből, beírni a jelszót vagy dönteni a megjelenő adatok alapján. Az interaktív alkalmazásoknak szükségük van a felhasználó beavatkozására, nem igaz? A lusta rendszergazda szerint nem mindig, köszönhető ez egy expect nevű kis programnak.

Bár én már régebben is hallottam az expectről, csak néhány éve fedeztem fel, mennyire hasznos. A munkatársammal egy olyan webalapú rendszert fejlesztettünk, ahol szükség volt egy egyszerű frissítésre a központi számítógép adatbázisából, ahol nem volt engedélyezett a parancsfájlok használata. A végrehajtáshoz szükséges adatok, az SQL utasítás szerint, csak a felhasználói felületen vihetők be. Ez az SQL utasítás fogja létrehozni a webes felülethez szükséges adatfájlokat. Az egész folyamat mozgatórugója a felhasználó, aki a terminál előtt ülve különböző helyekre különféle adatokat ír be. Később az expect lehetővé tette a weboldal további igények szerinti fejlesztését.

Vajon tényleg szükségünk van erre? Emlékszünk még a fejezet elején található elmélkedésre a lustaságról? Nos, tegyük fel, késő éjszakai dolgozol, és indulás előtt az utolsó teendő, hogy bejelentkezz egy távoli weboldalra és megbizonyosodj arról, hogy az alkalmazás elvégezte a munkát (hajnali háromra mindig kész is lesz), majd letöltöd az alkalmazás által létrehozott fájlt a helyi oldalra.

Este tíz órakor már inkább a hazamenésen jár az eszed, mint a fájl elkészülésének varázslatos pillanatán. Csak el kellene indítanod a programot, amely elindítja az ncftp-t a letöltéshez, de nem tudod a fájl nevét, mivel az minden futtatáskor megváltozik. A nevet megtalálod, ha bejelentkezel a rendszerbe és megvizsgálod a befejezési naplót (completion log).

Az egész aprócska folyamatot csupán a szemléltetés kedvéért mutattam be, mert ez is jól példázza, hogy egy egyszerű kis parancsállomány milyen nagy segítséget jelenthet.

Az expect parancsfájl alapformája a következő:

```
#!/usr/local/bin/expect
# Megjegyzések (név, mit csinál stb.,
# opcionális)
spawn valamilyen_parancs
set response énválaszom
expect "Valami válasz . . . ."
send $response\r
close
```

Nézzük, mi is történt: a spawn kulcsszó megadja az expectnek, hogy indítsa el a programot. Ez lehet egy parancsállomány (spawn /bin/bash), vagy a folyamat bármilyen parancsa. A set kulcsszóval beállítottam egy választ az előre meghatározottak közül. A nyelv érdekessége, hogy az expect kulcsszó itt pontosan azt teszi, amit jelent. Megvizsgálja a spawn-ban meghatározott parancs kimenetét, és megkeresi az egyező szöveget. Azután a send parancssal válaszol a keresett szövegre az első változóval, \$response. Most nézzünk egy valódi példát.

Egy Apache webkiszolgálót futtattam a rendszeremen OpenSSL-kiegészítéssel, a biztonságos ügyletek céljából. Az Apache, az OpenSSL kiegészítéssel, az indulás során egy biztonsági jelszót (pass phrase) kér, mivel a szerveren az egyedi kulcsfájlok kódolva vannak. Addig minden rendben van, amíg én ott vagyok és beírom a jelszót, de mi történik akkor, ha a kiszolgáló éppen akkor áll le, amikor én nem vagyok ott? Ez hónapokig nem fordul elő, csak akkor számíthatok rá, ha éppen szabadságra megyek. Egy örült pillanatomban egy SCSI kártyát építettem be az új szalagos meghajtómhoz. Csakhogy, elfelejtettem (megtörtént) újraindítani a kiszolgálót, ami az OpenSSL-t is futtatta. Most mi lesz? A feledékenység kiküszöbölésére írtam az alábbi egyszerű expect parancsfájlt:

```
#!/usr/bin/expect
# Routine: startapachessl
# Purpose: A web szerver elindítása az OpenSSL \
# kiegészítéssel
#
log_file -a /tmp/expectlog
#log_user 0
spawn /bin/bash

sleep .2
send "usr/local/apache/bin/apachectl startssl\r"
expect "Enter pass phrase*"
sleep .2
send "azéntitkosválaszom\r"
sleep .2
close
```

Ha a rendszer újraindul, akár ott vagyok, akár nem, ez a parancsfájl újraindítja az Apache kiszolgálót is az OpenSSL kiegészítéssel. A programot vizsgálva, néhány érdekes dologra lehetünk figyelmesek. Például a log_file értéke új. Meg kell határozni egy log fájlt a parancsállomány végrehajtásához. Azt, hogy a fájlba írjunk-e, vagy sem, a log_user értékében határozhatjuk meg. Ha ez „1”, a naplózás meg fog történni. Én a parancsfájl ellenőrzése során szívesen hasz-

nalantam a log_user értéket; mindenki el tudja dönteni, hogy az állandó naplózásra szüksége van-e vagy sem. Megjegyzem, én a bash héjból indítottam a kiszolgáló újraindítására szolgáló parancsfájlt, ami azután alvó állapotba került. Minden esetben, a parancsállományban egyötöd másodpercet vártam a folytatással. Végül, a záró lépés jelzi az elindított folyamatnak, hogy a továbbiakban nincs rá szükség. Ezzel az expect parancs kilépett, és visszatért az őt elindító folyamathoz. A továbbiakban nincs szükség arra, hogy más nyelveken programozzuk ezeket a funkciókat, mert az expect használatával sokkal egyszerűbb. Amint azt az előbbiekből is láthattuk, nem minden eszköz felel meg minden feladathoz. Az interaktív alkalmazások gyors és aljas indokból elkövetett automatizálásához, az expect parancsnál jobbat nem igen találunk. Az expect teljes megismeréséhez egy egész könyvre szükség lenne (nekem például van egy). Hogy is tudnám jobban megismertetni a használatát, mint példákkal bemutatni a nyelv minden egyes aspektusát. Mielőtt elszaladnál kísérletezni, nézzük meg az alábbi példákat.



1. kép Űrlap jelszótároltathoz

Mindannyian tudjuk, a jelszótároltás alapvető dolog, ahogy egy jó jelszó kiválasztása is. Ez nem olyan nagy dolog, ha a felhasználó éppen bejelentkezett; de egy kicsit bonyolultabb, ha egyáltalán nincs felhasználói fiókja. Értem ezalatt azokat az e-mail felhasználókat, akik POP3 (vagy webes) eléréssel használják a kiszolgálót, de nincs parancssori hozzáféréstük. Számos irodában, ahol e-mail kiszolgálóként és Internet-átjáróként Linux-alapú rendszert használnak, pontosan ezzel a gonddal küzdenek. Hogyan engedélyezzük a felhasználóknak jelszavaik megváltoztatását, ha nincs engedélyük a belépésre? Mindenek előtt a jelszó-változtatási eljárás az alábbiak szerint történik:

```
[root@myhost] # passwd
New UNIX password:
```

Kicsit bonyolultabb a helyzet, ha a felhasználó nem rendszergazda, és mégis meg szeretné változtatni a jelszavát. Ehhez először meg kell adnia a régi, így a párbeszéd még bonyolultabb. Most mi lesz? Létre kell hozni egy webalapú űrlapot, ahová a felhasználó minden szükséges adatot beírhat (lásd 1. kép). Az űrlap mögött dolgozó Perl parancsfájl kiemeli a változókat és beilleszti az expect parancsállományt, ami már elvégzi a további feladatokat. Ha szeretnéd alaposan megvizsgálni vagy használni ezt az ügyes kis webes alkalmazást, nyugodtan letöltheted a weboldalamról. Addig is nézzük meg ezt a részt az alkalmazás oldaláról.

```
#!/usr/bin/expect
# Routine: psdcmd
# Purpose: A felhasználói jelszó megváltoztatása \
          az expect használatával
log_user 1
set uservar [lindex $argv 0]
```

```
set currpassword [lindex $argv 1]
set newpassword [lindex $argv 2]
set renewpassword [lindex $argv 3]
#
# log_file -a /tmp/expectlog
# send_user "Spawning passwd command with \
            uservar.\n"
spawn su -l -c "passwd" $uservar
expect "Password:"
sleep .1
send "$currpassword\r"
sleep .1
#
expect {
    "(current) UNIX password:" {send \
                                "$currpassword\r"}
    "su: incorrect password" {exit 0}
}
sleep .1
expect {
    "su: incorrect password" {exit 0}
    "New UNIX password:" {send \
                           "$newpassword\r"}
}
sleep .1
expect {
    "BAD PASSWORD:" {exit 0}
    "Retype new UNIX password:"
    & {send "$renewpassword\r"}
}
sleep .1
expect {
    "su: incorrect password" {exit 0}
    "New UNIX password:" {exit 0}
}
#End of password change routine
```

A set varname [lindex \$argv num] felépítés az expectnek megfelelő tartományt mutatja. Megjegyzem, a „spawn” paraméter hívja a su parancsot a felhasználó jelszavának megváltoztatásához. Alaphelyzetben a webkiszolgálón található CGI parancsfájlok hajtják végre az átlag felhasználók (mint „nobody”, „www”) jelszavainak megváltoztatását, de a kiemelt felhasználók jelszavait nekünk kell megváltoztatni. Mellesleg, az Apache-kiszolgáló alapértelmezett felhasználóját meg kell tartani nem rendszergazda jogosultságúnak, különben bormalmas eseményeknek leszünk szem- és fültanúi.

Ebben a parancsfájlban egy másik új elemet is találhatunk, a send_user értéket. Ez tulajdonképpen egy állapotnyomtatási sor. Szándékosan benne hagytam a parancsfájlban, mert szerettem volna, ha tisztán látszik a hibakeresés (debugging) működése az expect parancsfájlban. Minden programozó beilleszt egy hibakeresési sort, hogy láthassa mi, miért és hogyan történik. Ez ebben az esetben is így van. A send_user parancsot használhatjuk a parancsállomány külvilággal társalgására, beleértve a végrehajtás folyamatát is. Mivel a Perl parancsfájl segítségével látom az expect kimenetét, jól láthatom az alábbi üzeneteket is. Együttal a Perl meghívja az alábbi parancsot:

```
$return_code =3D './psdcmd "$username"
"$currpassword" "$newpassword" "$renewpassword" ';
```

Amint látszik, az expect parancsfájl hívásra kerül a felhasználói névvel, a jelenlegi jelszóval, az új jelszóval és az új jelszó ismétlésével. Egyszerűen beírhatjuk az új jelszót kétszer, de az ellenőrzési eljárást

a parancssori megoldáshoz hasonlóan kell megoldani. Azonkívül feltétlenül jó ötlet a felhasználói jelszóváltoztatás megerősítésének kikényszerítése.

Az interaktív automatika automatizálása

Most, hogy túl vagyunk az expect parancsállomány bevezetőjén, hihetetlenül könnyen tudunk különböző eljárásokat automatizálni. A parancsállomány létrehozására nem lehetne egy programot készíteni? Az expect telepítésekor felkerül a gépre egy autoexpect nevű apró kis program. Egyszerűen szólva, az autoexpect figyeli az interaktív folyamatban való ténykedésünket, és elkészíti számunkra az expect parancsállományt. A parancsot az alábbi alakban kell használni:

```
autoexpect -f script_outputfile command_string
```

Tegyük fel, hogy be szeretnénk jelentkezni egy tűzfal mögött lévő távoli gépre, kétszeres bejelentkezéssel. Miután bejelentkeztünk a tűzfalra, bejelentkezünk a belső hálózatba is (telnet, ssh stb.), majd elindítjuk a szokásos menüt. Ezt az egész bejelentkezési és programindítási eljárást szeretnénk automatizálni. A parancssorba a következőket kell begépelni:

```
autoexpect -f superlogin.script telnet \  
firewall.mycompany.com
```

Adatok

Szerző: Marcel Gagné

Cím: Linux rendszerfelügyelet: Felhasználói kézikönyv

A kiadás tervezett ideje: 2001

Azonosítója: ISBN 0-201-71934-7



Amikor végeztünk a bejelentkezéssel, ki kell lépni a menüből, és ki kell jelentkezni. Az autoexpect szépen rögzítette az egész folyamatot. Mielőtt az új parancsállományt futtatnánk, akad némi szerkesztési munkánk, mivel az autoexpect egy kicsit bőbeszédűbb, mint azt mi szeretnénk. Továbbá el kell távolítani a kilépést a menüből és a rendszerből, de az alap-parancsállomány teljes testi épségében elérhető. Már csak indítható állományt kell elkészíteni a parancsállományból, és már végeztünk is.

Még valamit azonban meg kell tennünk, az új expect parancsállomány végére be kell illeszteni a következőt: interact.

Ez a parancs utasítja az expectet, hogy adja vissza a vezérlést a rendszernek, miután elvégezte a feladatát. Enélkül elfoglalná az egész gépet, és csupán be- és kijelentkezhetnénk, de azt nagyon gyorsan. Véleményem szerint, az expect használatáról nem lehet teljes képet adni, de remélem, hogy e rövid bevezetővel kellemes étvágyat keltetem, és megfelelő képzelőerő segítségével, az alkotó lustaság tökéletességig való fejlesztéséhez lökést adtam. Ezután más elfoglaltságot kereshetünk magunknak.

Mi ez az izé a varázsköpenyről a képernyődön?



Marcel Gagné (mggagne@salmar.com)

Mississaugában, Ontarióban él. A Salmar Consulting Inc. (rendszerintegrációs és hálózati tanácsadó vállalat) elnöke. Továbbá pilóta, tudományos fantasztikus regények szerzője, és a TransVersions tudományos fantasztikus,

fantasy- és horrorválogatás szerkesztője. A Linux és Unix-változatok nagy kedvelője, és ezt nyilvánosan is hangoztatja. Honlapján rengeteg más érdekes dolog is található

➔ <http://www.salmar.com/marcel/>.



PHP4 és PostgreSQL: komoly webes alkalmazások készítése nyílt forráskódú programeszközökkel

Tim Perdue végigvezet bennünket egy egyszerű webes alkalmazás készítésének lépésein, hogy bemutassa a PHP és a PostgreSQL képességeit.

Nem is olyan rég egy komoly webalkalmazás könnyű felépítése egyet jelentett kemény pénzüsszegek kipengetésével egy nagyobb rendszerért (mint amilyen például a Cold Fusion), valamilyen üzleti adatbázis-kezelőért (például a Sybase) és egy Sun kiszolgálóért. Szerencsére, ezeknek a napoknak immár vége van. Az Apache felemelkedése és a jelenlegi ingyenes adatbázis-kezelők kifejlődése végre valós (esetleg még jobb) választást kínálnak az üzleti alkalmazások mellett.

A nyílt forráskód legjobb hajtásai: egy Perl-szerű parancsnyelv, a PHP és egy hatékony objektumközpontú adatbázis-kezelő, a PostgreSQL. Ha összekapcsoljuk a PHP-t és a PostgreSQL-t, szinte bármit felépíthetünk az egyszerű vendégkönyvtől kezdve a hatalmas webalapú bejelentkezési rendszerekig. A PHP biztosítja az agyat, míg a Postgres adja az erőt.

Be szeretnék mutatni egy nagyon egyszerű PHP vásárlási kártya-és leltárrendszert, ami kihasználja a Postgres tranzakciós képességeit. A honlapomon (PHPBuilder.com) letöltheted a forráskódot, és egyéb példaprogramokat is találsz.

Az első dolog, amivel foglalkozni szeretnék, az alkalmazás szerkezete. Minden webalkalmazásban, amit csak PHP-ben fejleszték, első lépésként mindig létrehozok egy átfogó könyvtárat, amit aztán a webhely minden lapjáról becsatolhatok. Ezt a könyvtárat common.php-nak neveztem el, és az include nevű könyvtárban található meg. A programkönyvtár segít elvégezni az olyan megszo-kott feladatokat, mint például az adatbázishoz való csatlakozást, a felhasználó kilétének megállapítását, a lap fejlécének és láblécének beállítását stb. Azáltal, hogy ezeket az eljárásokat egy helyen tartjuk, az alkalmazásunk sokkal átláthatóbb és könnyebben alakítható lesz. Például ahelyett, hogy mindenhol közvetlenül a kódba helyeznénk el az adatbázis-kapcsolat létrehozását, csak egyszer írjuk meg, magában a programkönyvtárban.

Nos, a könyvtárunk első változata máris használható. Kapcsolatot teremt az adatbázissal, sőt, néhány egyszerű HTML-absztrakciót is ad számunkra. Ahogy a honlap HTML része egyre bonyolódik, a legtöbb részt beszúrhatjuk a fejlécbe/láblécbe (site_header/site_footer) ezáltal egyszerre végezve el a változtatást a teljes alkalmazásban.

A könyvtár egy egyszerű absztrakciós réteget is nyújt a Postgres-lekérdezésekhez. Ahhoz, hogy csökkenteni tudjam a megírandó

kód mennyiségét, a lehető legegyszerűbb módon készítettem el. Végül pedig meghívja a PHP4 beépített munkafolyamat kezelő-kódját. A session_start meghívása arra kéri a PHP4-et, hogy töltsen újra a bejegyzett változókat, és így azok elérhetőek lesznek a teljes alkalmazás alatt. Honlapunk minden lapja alapvetően valahogy így fog kinézni:

```
?php

// common programkönyvtár használatba vétele
require ($DOCUMENT_ROOT.'/include/common.php');

echo site_header('Példa honlap');

/*
    a lap belső logikája
*/

echo site_footer();

?
```

Általában minden alkalmazás tervezésekor érdemes annyira elválasztani a gerincet alkotó belső logikát az éppen időszerű megjelenítéstől (jelen esetben a HTML-től) amennyire csak lehetséges. Én általában a belső logika részeit funkcióhívások belsejébe rejtem,

amelyeket aztán a webhely bármely lapjáról csatolni és hívni lehet. Ez azért előnyös, mert esetenként előfordulhat, hogy más csatolófelületeket is szeretnél az alkalmazáshoz illeszteni – talán épp egy különlegesen könnyű súlyú felületet a vezeték nélküli alkalmazások számára. Ha a logika elemei a HTML-megjelenítés részeként szerepelnek a kódban, meg kellene ismételni a teljes logikát minden egyes felülethez. Ha viszont függvénykönyvtárakba választjuk szét, minden felület ugyanazt a logikát használhatja. A PHP függvénykezelésével csak egy gond van: nincs általános kivételkezelő eljárás. Ha egy belső hiba történik egy függvényben, honnan fogja tudni ezt a hívó kód, hogy figyelmeztethesse a felhasználót? Más nyelvekben, mint például a Javában berakhat-



A webkiszolgáló és az ügyfél közti kapcsolat

nád a kivételt a függvény belsejébe munkafolyamatként (method). Amikor meghívod a Java munkafolyamatot, átírányíthatnád egy try/catch utasítással, s így a gond meg is oldódna. PHP-ban általában úgy oldom meg ezt a nehézséget, hogy minden függvény visszatérési értéke igaz vagy hamis, és mindig beállítom a \$feedback (visszajelzés) nevű globális változót. Az eredmény így lekérdezhető, és a \$feedback tartalma szükség esetén a képernyőre írható. Van egy PEAR <http://pear.php.net/> néven futó ilyen irányú kezdeményezés, amely megpróbálja egységesíteni többek közt a hibakezelést és az adatbázis-elérést, de jelenleg még nem igazán terjedt el. Íme egy példa, hogyan hívhatunk függvényeket az általam megadott igaz/hamis módszerrel:

```
<?php
$eredmeny=fuggvenynev();

if (!$eredmeny) {
    // hiba történt, jelenítsd meg
    echo $feedback;
} else {
    // folytatd, ha sikeres volt
}

?>
```

Nos, lássuk akkor a bevásárlókosarunkat. Szükségünk lesz néhány alapvető adatszerkezetre, amiben elraktározhatjuk a kosár adatait: kezdésként csupán egy készletadatbázisra, ami felsorolja a nevet, az alkatrészszaámot, az árat és a készlet mennyiségét. Ezenkívül nélkülözhetetlen a vásárlóink és az általuk vásárolt áruk követése. Ez már elég összetett feladat lesz mindnyájunknak.

Ennyi elég ahhoz, hogy egy csökevényes virtuális bevásárlókosarat készítsünk. Az adatbázis séma egységesítése céljából külön táblát készítettem, ez felsorolja a vásárló kosarának tartalmát. Így, hogy a vásárló többször tehet árut a kosarába, emellett egyszerű módon csatlakozhatunk kosarának tartalmát a készletadatbázishoz.

Most ideje meggondolni, milyen műveleteket szeretnénk elvégezni a virtuális boltunk függvényeivel. A legalapvetőbb feladatok: új kosár kérése, az áru kosárba helyezése, valamint a kijelentkezés. Egy valódi hálózati boltban természetesen sokkal több dologra lenne szükség. Ilyen például az áruk közötti válogatás lehetősége, a mennyiség megváltoztatása és a többi megszokott tevékenység. Ezeket a feladatokat azonban önökre bízom.

Egy egyszerű feladattal kezdem, ezzel egy új vásárlót lehet létrehozni. A következők tartoznak ide: az általunk készített vásárló (costumer) szekvencia következő értékének lekérése, ezen érték beszúrása a costumer táblába, majd pedig az érték felvétele a PHP4 beépített folyamatkezelő kódjába.

Ez kicsit több kódot vett igénybe, mint szerettem volna, viszont bemutatja, miként kell helyesen elindítani és befejezni egy tranzakciót Postgresben, és hogyan kell ellenőrizni a lekérdezések hibamentességét. Ugyanazt a rutint fogom hibellenőrzésre használni a teljes kód során, mint amit mindenkinek ajánlok.

Mindig tervezzük meg, miként fogjuk kezelni a helyzetet, ha a lekérdezés sikertelen. Teljesen leállítjuk a parancsfájlt, újra próbáljuk a lekérdezést, vagy csak egyszerűen továbblépünk, mintha mi sem történt volna? Óvatosan mérlegeljük az összes lehetőség hatását. Például, ha nem tudunk lekérni a következő vásárló azonosítóját, nem igazán kellene továbblépni az új vásárlórekord készítésére. Ha a vásárlórekord létrehozása sikertelen, később nem tudjuk frissíteni a cím adatait, és nem tudunk árut rakni a kosarába.

Csak logikusan, rendben?

Most lássuk az áru kosárba helyezésének munkafolyamatát. Ez szintén elég egyszerű eset. Mielőtt beraknánk egy árut a kocsiba, előbb ellenőrizni kell, létezik-e egyáltalán ilyen azonosító az adatbázisban. Ezt gyakorlati megfontolásokból tesszük, mivel az áruazonosító a böngészőtől érkezik, amibe így könnyen belepiskálhattak. Ha egyszer tudjuk, hogy az azonosító létezik, megnézhetjük, hogy már a kocsiban van-e. Ha már van, akkor megnöveljük a mennyiségét új sor beszúrása helyett. Ha viszont nincs még a kocsiban, akkor beszúrjuk a bevásárlókocsiba az alapértelmezett egy mennyiséggel. Most már tudunk új vásárlókat létrehozni, és árut rakni a kocsijukba. Már csak arra van szükségünk, hogy a vásárlók ki tudjanak jelentkezni a boltból, és emellett megfelelően csökkentjük a készleteket. Ez a legbonyolultabb része az egész boltnak, és most jó hasznát vesszük a Postgres tranzakcióinak, valamint fejlett zárolási rendszerének. Kezdetnek a Postgres SELECT...FOR UPDATE szintaxisát fogjuk használni, ami tulajdonképpen lezárja a kiválasztott sorokat, így

1. lista Példa a programkönyvtár kódjára

```
common.php:

<?php

//Csatlakozás a PHP adatbázishoz.
$conn=pg_pconnect("user=tim dbname=db_example");

//ellenőrizzük, sikeres volt-e a csatlakozás
if (!$conn) {
    //csatlakozási-hiba lap
    //hiba esetén is folytathatod a munkát
    //az adatbázis nélkül. Tőled függ.
    echo pg_errormessage($conn);
    exit;
}

//Állítsunk be valami általános fejléctet.

function site_header ($title) {
    return '<HTML>
<HEAD>
<TITLE>'.$title.'</TITLE>
</HEAD>
<BODY>';
}

//Egy közös HTML-rész a lapok végére

function site_footer () {

    return '</BODY></HTML>';
}

//Egy egyszerű rövidítés
//minden postgres lekérdezéshez.

function query($sql) {
    global $conn;
    return pg_exec($conn,$sql);
}

// PHP4 utasítása, hogy automatikusan minden lapon
// beállítsa/visszatöltse a megfelelő
// munkafolyamat-állapotot.
session_start();

?>
```

2. lista Bevásárlókosár-adatszerkezetek

Cart.sql:

```
# Egy szekvencia segítségével hozzuk létre
# a felhasználói azonosítókat.
# Nagyobb lépésközt állítottam be, hogy
# csökkentsem az esélyét annak, hogy a fel-
# használók egymás azonosítóját kitalálják.
#
#
create sequence seq_customer_id
    increment 26 start 1;

create table customers (
    customer_id int not null default 0 primary key,
    name text,
    address text,
    credit_card text,
    total_order MONEY DEFAULT '$0.00'
);

create table cart_items (
    cart_item serial,
    customer_id int,
    part_number int,
    quantity int
);

create index idx_cart_customer
    on cart_items(customer_id);

create table item_inventory (
    part_number serial,
    name text,
    price float,
    inventory int
);
```

frissítheted őket, és elküldheted a változásokat a tranzakcióval. Ennek a szintaxisnak a tranzakción belüli használatával biztos lehetsz afelől, hogy az adatok konzisztensek maradnak. Némely adatbázis-kezelőben, például a MySQL-ben, könnyen zárolhatsz megadott adatsorokat, hogy megakadályozd a többi folyamatot abban, hogy csökkentsék a készleteket, amikor magad is ugyanezt szeretnéd tenni.

Ez a lekérdezés alválasztásokat (subselect) is használ, ez szintén egy általános lehetőség a hatékonyabb adatbázis-kezelők közt. Az alválasztások lehetővé teszik, hogy két lekérdezést foglalj egybe, hogy megkönnyítsd az élelet.

Miután zároltuk a sort, egy lekérdezést kell végrehajtanunk, hogy csökkentsük a készletszámot a kocsiban lévő minden egyes áru esetén. Az egyszerűség kedvéért, nem figyelmeztetünk a készletek kimerülésekor, de sikeresen állítjuk negatív értékre az áruszámlálót (item_count), ha a készlet mennyisége nulla alá csökken. Esetleg felállíthat egy vezérlőlapot, amin megnézheted a negatív darabszámú árukat, és rendelhetsz utánpótlást.

Végül, szeretnénk kibővíteni a vásárlótáblát a látogató hitelkártyaszámával, szállítási adataival, valamint a teljes eladási mennyiséggel, majd pedig megsemmisítjük a PHP4 vásárló munkafolyamatot. Íme egy viszonylag összetett tranzakció, amelynek minden lépését megfelelően kell végrehajtani. Ha ez nem történik meg, mindent vissza kell állítani az eredeti helyzetbe.

Ha nem részesültél a példában szereplő tranzakciók élvezetében, és a lekérdezés sikertelen volt valamelyik készletadat módosítása

3. lista Új vásárló bejegyzése

<?php

```
function cart_new() {
    global $conn, $customer_id, $feedback;

    //Nyitunk egy tranzakciót.
    query("BEGIN WORK");

    //A következő azonosító lekérdezése.
    $res=query(
        "SELECT nextval('seq_customer_id')");

    //A hibák ellenőrzése.
    if (!$res || pg_numrows($res)<1) {
        $feedback .= pg_errormessage($conn);
        $feedback .= ' Hiba - Nem kaptuk meg a
        ↳következő azonosítót';
        query("ROLLBACK");
        return false;
    } else {
        //A visszatérési érték elmentése egy
        //helyi változóban.
        $customer_id=pg_result($res,0,0);

        //Az azonosító bejegyzése a PHP4-gyel.
        session_register('customer_id');

        //Az új vevőhöz tartozó sor beszúrása.
        $res=query("INSERT INTO customers
        ↳(customer_id) VALUES
        ↳('$customer_id')");

        //Hibák ellenőrzése.
        if (!$res || pg_cmdtuples($res)<1) {
            $feedback .=
                pg_errormessage($conn);
            $feedback .= ' Hiba - nem tudtam
            ↳bejegyezni az új vásárlót';
            query("ROLLBACK");
            return false;
        } else {
            //A tranzakció érvényesítése.
            query("COMMIT");
            return true;
        }
    }
}

?>
```

közben, akkor bizony sok gondod lesz. Szét kell választanod a készletet, megváltozott és változatlan részekre. Ha a látogató újra megkísérli elérni a lapot, honnan fogod tudni, melyik készletadatot kell ismét csökkenteni? A válasz: sehonnan, bizony egy pontatlan készletet kapsz eredményül.

Ennek a cikknek nem célja, hogy teljes körű vásárlókocsi-leírást adjon – egy egész könyvet írhatnék erről, ha lenne elég időm. A cél tulajdonképpen annyi, hogy bemutassuk a háttér munkákat, és szemléltessük azt a tervezési és végrehajtási módszertant, amit minden webfejlesztőnek tanácsolok. További adatokért keresd a [PHPBuilder.com](http://phpbuilder.com) lapot.

Az a néhány rész, amit kihagytam, főleg a kocsi tartalmának megtekintésével, illetve az árukészlet böngészésével foglalkozik. Ezek a részek, az itt található kódokkal együtt, zipfájlból elérhetők a <http://www.phpbuilder.com/columns/linuxjournal200009.php3L> honlap vitaasztalán (discussion/comment board).

4. lista Új elem kosárba helyezése

```

<?php
function cart_add_item($item_id,$quantity=1) {
    global $customer_id, $feedback, $conn;

    //Egy tranzakciót kell nyitni, hogy csak egy
    //lekérdezés módosíthassa az adatbázist.

    //Lekérdezzük a termékek közül a
    //következőt.
    $res=query("SELECT * FROM item_inventory
        ➤WHERE part_number='$item_id'");

    //A hibák ellenőrzése.
    if (!$res || pg_numrows($res)<1) {
        $feedback .= pg_ErrorMessage($conn);
        $feedback .= ' Error-item not found ';
        return false;
    } else {
        //Megnézzük, hogy az elem már a kosárban
        //van-e. Ha igen, megnöveljük a mennyiségét.

        //Tranzakciót kezdünk, hogy zároljuk a
        //sorokat, ha megtaláljuk azokat.
        query("BEGIN WORK");

        $res=query("SELECT * FROM cart_items ".
            "WHERE part_number='$item_id' AND
            ➤customer_id='$customer_id' FOR
            ➤UPDATE");

        //A hibák ellenőrzése.
        if (!$res || pg_numrows($res)<1) {
            //Rakjuk a kosárba!

            $res=query("INSERT INTO cart_items ".
                "(customer_id,part_number,quantity)".
                "VALUES
                ➤($customer_id,$item_id,$quantity)");

            //Volt a beszúrásnál hiba?
            if (!$res || pg_cmdtuples($res)
                ➤< 1) {
                $feedback .=
                pg_ErrorMessage($conn);
                $feedback .= ' Hiba - A kosárba
                ➤helyezés nem sikerült';

                //Semmi sem változott, mégis jobb,
                //ha érvénytelenítjük az egész
                //tranzakciót.
                query("ROLLBACK");
                return false;
            } else {
                query("COMMIT");
                return true;
            }
        } else {
            //Az elem már jelen volt.

            //Mennyiség.
            $res=query("UPDATE cart_items
            ➤SET quantity = quantity +
            ➤$quantity WHERE part_number=
            ➤'$item_id'AND customer_id=
            ➤'$customer_id'");

            if (!$res || pg_cmdtuples($res)
                ➤< 1) {
                $feedback .=
                ➤pg_ErrorMessage($conn);
                $feedback .= ' Hiba - nem tudom
                ➤megnövelni a mennyiséget';

                //Most sem történt semmi.
                query("ROLLBACK");
                return false;
            } else {
                //A változtatott mennyiség
                //érvényesítése.
                query("COMMIT");
                return true;
            }
        }
    }
}
?>

```

5. lista Ellenőrzés és a lista csökkentése

```

<?php
function cart_checkout($credit_card,$address,$name) {
    global $conn, $customer_id, $feedback;

    //Tranzakciót kezdünk.
    query("BEGIN WORK");

    /*
    A vásárló kosara alapján zároljuk a megfelelő
    sorokat.

    Ezt egy egyszerű subselecttel oldjuk meg.
    */
    $sql="SELECT * FROM item_inventory ".
        "WHERE part_number ".
        "IN (SELECT part_number FROM cart_items ".
        "WHERE customer_id='$customer_id') ";

    "FOR UPDATE";
    $res=query($sql);

    //A hibák ellenőrzése.
    if (!$res || pg_numrows($res)<1) {
        //Nincs megfelelő adat, vagy adatbázishiba.
        $feedback .= pg_ErrorMessage($conn);
        $feedback .= ' Hiba - nem zároltunk elemeket ';

        //A tranzakció megszakítása.
        query("END WORK");
        return false;
    } else {
        /*
        A megfelelő sorokat zároltuk.

        Az adatok kiolvasása a kosárból.
        */

```


Grafika: húzz egy kártyát... de melyiket?

A grafikus kártyák egyre olcsóbbak lesznek, és az egyre bővülő választékot szemlélve az ember azt sem tudja, melyik mellett döntsön.

Atavalyi téli fordulópontot jelentett a háromdimenziós linuxos alkalmazások számára. A két legnagyobb ász, az Epic Unreal Tournamentje és az id Software Quake III Arena című játéka csaknem egyszerre jött ki Windowsra és Linuxra. A linuxos grafikus gyorsítóprogramok hiánya miatt azonban még egy évvel ezelőtt is csak sopánkodtunk, hogy bizony a lemaradás komoly e téren a Windowszal szemben. Mára szerencsére a legtöbb, grafikus kártyákat fejlesztő cég egymással versengve jelenteti meg legújabb termékeinek linuxos meghajtóját. A Linuxot használó játékosok is elértek a sosem várt paradicsomba: több tucat 3D-s kártya közül választhatnak.

A játékosok

Kezdetben a grafikus gyorsításra Linuxon egyetlen megoldás létezett: a 3dfx. *Daryll Strauss* munkájának köszönhetően a Voodoo sorozatba tartozó kártyák a Glide-on és az OpenGL-en (és a Mesán) keresztül kiemelkedően jó linuxos támogatással bírtak. A 3dfx nemrégiben kötött szerződést a Precision Insight csoporttal, hogy elkészítik a cég Direct Rendering Infrastructure (DRI) eljárását kihasználó XFree86 4.0 meghajtókat a tökéletes 3D gyorsítás céljából.

A G200 és G400 kártyák szerkezetének nyilvánossá tételével immár a Matrox is a támogatók sorát bővíti. Kártyáiknak nem csupán 2D-s adottságait, de a Utah-GLX és az OpenGL segítségével most már 3D-s gyorsítási lehetőségeit is kihasználhatja minden linuxos játékfüggő. Az új XFree86 4.0 megjelenésével egy időben, a Precision Insighttal karöltve, a G400-ashoz is elkészülhet a DRI-vel támogatott 3D-s gyorsítás. Az ATI, az egyik legnagyobb grafikus kártyákat gyártó cég újabban egyre több finomsággal kecsegteti linuxos híveit. A régebbi Rage Pro kártyák Utah-GLX meghajtói után az XFree86 4.0-ban már a Rage 128 és a Rage 128 Pro kártyákat is beizzíthatjuk. Az ATI új grafikus processzorára, a Radeonra épülő, mostanában megjelenő kártyákhoz is teljes körű Linux-támogatást ígér.

Végül, de nem utolsósorban az nVidia, az utóbbi hónapok egyik legnagyobb hírverését keltve, minden termékéhez megjelentette az XFree86 4.0-s meghajtókat, így most már az nVidia név is határozottan jól cseng minden Linux-felhasználó számára, legyen szó akár a TNT-ről, akár a legújabb GeForce2-ről. Az általuk megalkotott OpenGL-támogatás minőségének és sebességének köszönhetően jelenleg a Linux-világ piacvezető gyártói.

A felsorolt cégek termékei közül így legalább egy tucat gyorsító-kártyát azonnal használhatunk Linuxszal is. Az alábbiakban mindegyiket kivesézzük, szót ejtünk erősségeikről, és természetesen nem rejtjük véka alá hiányosságaikat sem.

A kártyák

Nézzük most végig a Linux-támogatással is bíró legnépszerűbb grafikus kártyákat. Ezeket gyártóik szerint csoportosítottunk; így könnyebb összehasonlítani az egyes cégek termékeit.

3dfx

A 3dfx sorozata a manapság már kicsit gyengécskének tűnő Voodoo1-től a legújabb Voodoo4 és Voodoo5 kártyáig terjed. A Voodoo1 és 2 csak 3D-s gyorsításra képes, tehát szükség van egy „hagyományos” grafikus kártyára is a gépben, a legújabb modellek

viszont már 2D-s és 3D-s üzemmódban egyaránt használhatók. Minden 3dfx kártya támogatja a cég által fejlesztett Glide nevű grafikus illesztőt. A Glide számos linuxos játékban bizonyított. Ilyenek például a Myth II és az Unreal Tournament. Ezekben a játékokban a megjelenítés gyorsításához a 3dfx általában a legjobb (ha nem az egyetlen) választás. Sajnos, a legtöbb 3dfx kártyából hiányoznak bizonyos – a megszállott játékosok számára elengedhetetlenül szükséges – lehetőségek, például a teljes 32 bites színmélység támogatása vagy a stencil gyorsítár. Az újabb, Voodoo4 és 5 kártyákban már ezeket is megtaláljuk a rengeteg egyéb újdonság mellett. Ami az OpenGL-t illeti, a 3dfx meghajtók a Mesára épülnek, mely az OpenGL nyílt forráskódú változata. Ez azt jelenti, hogy a 3dfx kártyák OpenGL alatti teljesítménye leginkább a Mesa fejlesztésének függvénye.

Voodoo Graphics (Voodoo1)

ELŐNYEI: elsősorban az alacsony ár és a Glide támogatása. **HÁTRÁNYAI:** A Voodoo Graphics csak 3D-s gyorsítást végez, tehát a munka többi részét a videokártyánknak kell elvégeznie. Emellett (kevés memóriája miatt) csak 640x480-as felbontásra képes.

IRÁNYÁR: 9000 Ft (már csak használtan vehetjük meg).

Voodoo2 és Voodoo2 SLI

ELŐNYE: ez a kártya még mindig az egyik legolcsóbb megoldásnak számít, és igazi erejét párban mutatja meg. Két kártyával ugyanis kihasználhatjuk a Scan Line Interleave (SLI) lehetőséget; ilyenkor mindegyik kártya csak a kép egyik felének felépítésével foglalkozik, ezzel is növelve a teljesítményt és a sebességet. **HÁTRÁNYAI:** kisebb testvérehez hasonlóan a Voodoo2 is „csak” egy 3D-s kártya, tehát egy másik grafikus kártyára is szükségünk van. Ezek szerint az SLI felállítás 3 PCI csatlakozót foglal el, ami azért kicsit sok. A viszonylag magas árat és az elavult megvalósítást tekintve a Voodoo2 már egyáltalán nem tűnik vonzónak.

IRÁNYÁR: 17 000 Ft.

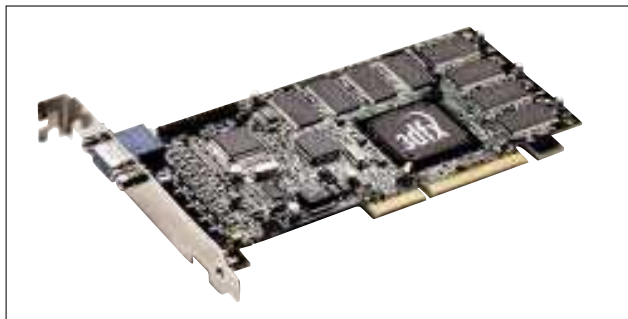
Banshee

ELŐNYEI: a Banshee egy 2D és egy 3D kártya egyben, tehát előrelépést jelent a Voodoo1 és 2-höz képest. Kedvező árának és a Glide-támogatásnak köszönhetően megfelelő választás a szerényebb igényű felhasználók számára. **HÁTRÁNYAI:** bár nagyobb órajelen működik, mint a Voodoo2, csupán egyetlen mintázattfeldolgozóval rendelkezik. Néhány játékban tehát gyorsabb lehet kisebb testvéreénél, de a sok mintázatot használó alkalmazásokban (azaz szinte az összes mai játékban) szemmel látható a sebességsökkenés.

IRÁNYÁR: 15 000 Ft.

Voodoo3-2000, 3000, 3500TV

ELŐNYEI: A Voodoo3 jóval gyorsabb és több szolgáltatást is kínál, mint a Voodoo2 SLI, 2D-s képességei pedig a Banshee színvonalán mozognak. A Voodoo3-as a sorozat legnépesebb családja, ezek közül szinte mindenki megtalálhatja a neki tetsző kártyát. A 3500TV-t tévészere is használhatjuk, akár Linux alatt is. **HÁTRÁNYAI:** a legjobb teljesítmény eléréséhez mindenképpen a 3500TV változatot kell



Voodoo3-3000

megvásárolnunk, ennek viszont nem sok értelme van a hasonló árú, de többet nyújtó 4-es és 5-ös változat mellett.

Ár: a sorozat darabjainak ára 15 000 és 35 000 forint között mozog, attól függően, hogy melyik modellt választjuk.

Voodoo4 és Voodoo5

Előnyei: ezekben a Voodoo3 minden szolgáltatását megtaláljuk, reneget újítással megtűzdelve (például: 32 bites színmélység, az óriás mintázatok támogatása, stencil gyorsítár). E kártyák a VSA-100 processzor köré épülő, jól méretezhető rendszerek, melyeket PCI- és AGP-változatban egyaránt megvásárolhatunk. A 3dfx most épp az FSAA, azaz a teljes képernyős simítás (anti-aliasing) gépi támogatásának megvalósításán fáradozik. Ez a lehetőség (melyet természetesen csak az új Voodoo4/5 kártyákba építenek bele) soha nem látott mértékben növeli majd a játékok látványosságát, és egyáltalán nem terheli sem a gép, sem pedig a kártya processzorát. HÁTRÁNYAI: az FSAA természetesen



Voodoo5-5500

csak az átlagosnál több memóriával valósítható meg: a Voodoo5-5500-at például 64 megabájttal szállítják (és többek között ezért is kerül nyolcvanezer forintba). A legjobb V5 kártya, a Voodoo5-6000, sajnos még drága, így a legtöbb játékosnak erről le kell mondania.

Ár: a legegyszerűbb Voodoo4-es 36 000, a család legerősebb tagja, a Voodoo5-6000 180 000 forintba (nem elírás!) kerül.

Matrox

A céget kristálytisztá képminőségű és gyors 2D kártyáiról ismerjük, azonban legújabb fejlesztéseikből, például a G400Max-ból már a 3D-s gyorsítást sem felejtették ki. A kártyák szerkezetének nyilvánosságra hozatala, valamint a Utah-GLX és a Precision Insight kemény munkája meghozta gyümölcsét: mára e kártyák 3D-s képességeit kihasználhatjuk Linux alatt is. A 3dfx és az ATI mintájára a Matrox is a Mesa-féle OpenGL-t használja illesztőként.

G200

ELŐNYEI: a G200-as egy olcsó kártya, gyönyörű 2D-s képpel és megfelelő teljesítményű 3D-s gyorsítással. Alacsony árfekvéséből nem következtethetnénk rá, de támogatja a 32 bites színmélységet 3D-ben is! HÁTRÁNYAI: sok jó tulajdonsága ellenére a kártya eléggé lassú, így ne reménykedjünk abban, hogy a Quake III-at bármilyen elfogadható sebességgel játszhatjuk.

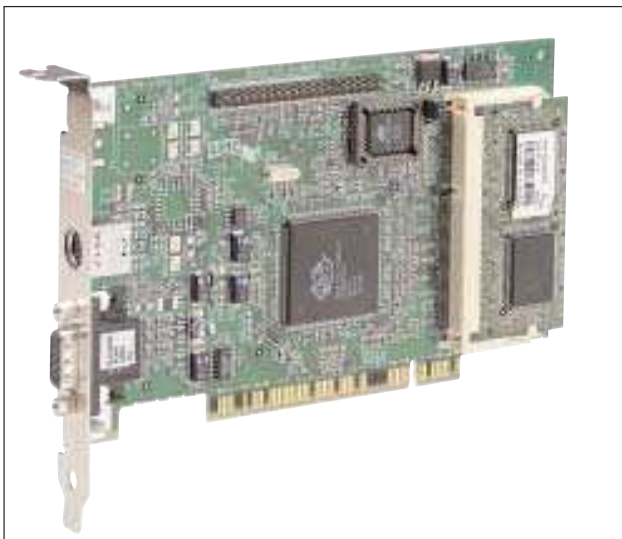
IRÁNYÁR: 15 000 Ft.

G400

ELŐNYEI: A G400Max a Utah-GLX meghajtókkal szemtelenül gyors 3D-ben is, a kétfejes változattal pedig egyszerre két képernyőt is vezérelhetünk. A G400 szerencsére több változatban is létezik, így mindenki kiválaszthatja a neki megfelelőt. A G400 minden korszerű 3D-s szolgáltatást támogat, így a 32 bites színmélységet és a nagy mintázatokat is, ráadásul mindezt igen jó sebességgel. Emellé társul a Matrox híresen tökéletes 2D-s képessége, ezért a G400 kártyák igen jó választásnak tűnnek. HÁTRÁNYAI: a G400Max alatti kártyák 3D-ben lassabbak, így aki a tökéletességre vágyik, annak bizony elég szép összeget kell otthagynia az üzletben. Ezenkívül az sem lenne rossz, ha a Matrox végre Linuxra is megjelentetné az érdességi térképek (bump mapping) támogatását, mint ahogy azt a windowsos meghajtók esetében már megtette. Ár: az alap G400 egyfejes (single head) 30 000 forint, a G400Max kétfejes (dual head) 65 000 forint körüli áron vásárolható meg.

ATI

A Matroxhoz és a 3dfx-hez hasonlóan az ATI is több fejlesztőcsoporttal együttműködött kártyái linuxos meghajtóinak kifejlesztése során. A cég a 2D-s gyorsítás linuxos megvalósításán is dolgozik, ezt például a DVD-meghajtókkal használhatjuk ki. A Radeon processzorra épülő, egy-két hónapja megjelent, és máris a 3D-s világ új királyaként számon tartott kártyájához is készülnek a linuxos meghajtók. A Rage Pro- és a Rage 128-alapú kártyák kiváló linuxos támogatása biztató jövőképet mutat.



ATI Rage Pro

Rage Pro

ELŐNYEI: ez a kártya már öreg motorosnak számít, így igen olcsón hozzájuthatunk. Utah-GLX támogatással általában gyorsabb, mint Windows alatt! HÁTRÁNYAI: a Rage Pro teljesítménye jócskán elmarad a mai követelményektől. A Quake III Arena csak az összes különleges hatás kikapcsolása után játszható rajta. E kártya képviseli a linuxos 3D gyorsítás legalsó osztályát.

IRÁNYÁR: 10 000 Ft.

© Kiskapu Kft. Minden jog fenntartva

Kapcsolódó címek

3dfx Interactive, Inc.:

- ➔ <http://linux.3dfx.com/>
- ➔ <http://www.3dfx.com/>
- ➔ <http://www.3dfxgamers.com/>

ATI Technologies Inc.: ➔ <http://www.ati.com/>

Matrox Electronic Systems Ltd.: ➔ <http://www.matrox.com/>

nVidia Corp.: ➔ <http://www.nvidia.com/>

OpenGL: ➔ <http://www.opengl.org/>

Mesa: ➔ <http://mesa3d.sourceforge.net/>

Precision Insight: ➔ <http://www.precisioninsight.com/>

DRI: ➔ <http://dri.sourceforge.net/>

Rage 128, Rage 128 Pro

ELŐNYEI: a Rage 128 sorozatot az XFree86 4.0 is támogatja, valamint a DRI kihasználásával árához képest hihetetlenül jó teljesítményre képes. HÁTRÁNYAI: a meghajtókat lassan fejlesztik, így jelenleg a többi gyártó modelljei között gyorsabb és hatékonyabb illesztőprogramokkal szállított kártyákat is találhatunk. ÁR: a Rage 128-as 20 000 Ft, a Rage 128 Pro 40 000 Ft.

nVidia

A cég vetélytársaihoz képest később, idén tavasszal kapcsolódott be a linuxos fejlesztésekbe, ekkor jelentek meg ugyanis az első XFree86 4.0 meghajtók. A windowsos illesztőkben használt módszerekre alapozva a létező leggyorsabb linuxos OpenGL gyorsítást valósították meg. Az nVidia minden kártyájához ad linuxos támogatást, a legolcsóbb TNT-től a rendkívüli GeForce2-ig. E kártyák minden új szolgáltatást elérhetővé tesznek, tehát a 32 bites színmélységet és a stencil gyorsítást is. A GeForce és a GeForce2 újdonsága a grafikus processzor, ez a számítógépes műveletek (koordináta átalakítások, megvilágítási hatások stb.) elvégzésének terhéet veszi a gép processzorának vállairól. Ez hatalmas sebességnövekedéshez vezet, főleg egy kisebb teljesítményű gépen. Az nVidia minden, a Windowsban kihasználható szolgáltatást Linux alatt is támogat, például az AGP adatküldést, a mintázat tömörítést és a síkképernyős kijelzők vezérlését is. A Windows alatt nemrégiben megvalósított teljes képernyős simítást hamarosan Linux alatt is élvezhetjük. Az nVidia másik előnye a 3dfx-szel, a Matroxszal és az ATI-val szemben, hogy a cég csak grafikus processzorokat készít, a kártyákat más vállalatok szerelik össze, ez pedig sokkal egészségesebb piaci verseny kialakulását teszi lehetővé.

TNT

ELŐNYEI: a TNT kifogásalan OpenGL-támogatással és 32 bites színmélységgel bír. HÁTRÁNYAI: az árából máris következtethetünk arra, hogy egy régi és viszonylag lassú kártyáról van szó, mely a jelenlegi kihívásoknak (pl.: a kevés memória miatt) nem tud teljes mértékben megfelelni. IRÁNYÁR: 10 000 forint (csak használtan vásárolhatjuk meg).

TNT2 és TNT2 Ultra

ELŐNYEI: a családba sokféle képességű és árú kártya tartozik, ezek közül még a játékrajongók is biztosan tudnak választani. A TNT2 Ultra még mindig nagyon gyors kártyának számít, 32 MB memóriájának és kifogástalan OpenGL-támogatásának köszönhetően valószínűleg még egy jó darabig semmi más nem tudja elvenni tőle a „legnépszerűbb középosztályú 3D-s kártya” címet. HÁTRÁNYAI: a csúcsmoddell TNT2 Ultra teljesítményéhez képest viszonylag drága, így

vélhetőleg számos vékonyabb pénztárcájú játékos kedvéért elveszi a vásárlástól. Emellett néhány tulajdonsága csak a szemmel láthatóan lassúbb 32 bites üzemmódban használható ki.

IRÁNYÁR: egy kisebb TNT2 12 000, a TNT2 Ultra pedig 40 000 Ft.

GeForce SDR és DDR

ELŐNYEI: az nVidia grafikus processzor egy régebbi gépben is komoly teljesítménynövekedést valósít meg. A mintázattömörítés, a 32 bites színmélység és még számos tulajdonság kiváló OpenGL gyorsítótár teszi a kártyát. HÁTRÁNYAI: a GeForce sávszélessége még DDR memóriára használatával is korlátozott a csúcsteljesítményű gépeken. A grafikus processzort jelenleg csupán néhány játék használja ki.

IRÁNYÁR: a GeForce 32 MB SDR 40 000, a GeForce 32 MB DDR 70 000 forintba kerül.

GeForce 2 GTS

ELŐNYEI: ebben a kategóriában jelenleg ez a létező leggyorsabb grafikus kártya, és a grafikus processzornak köszönhetően éveket is a csúcson maradhat. A linuxos játékkörülteknek ezt kell választaniuk! Mivel több gyártó is készít ilyen kártyákat, az árak valószínűleg egyre lejjebb süllyednek. Ha az nVidia végre Linuxra is kihozza a teljes képernyős simítást támogatását, ez lesz az egyetlen kártya, amellyel az újdonságokat ki is tudjuk használni. Elmondhatjuk, hogy aki GeForce 2 GTS-t vásárol, annak jó ideig nem kell foglalkoznia új grafikus kártya kiválasztásával. HÁTRÁNYAI: a legnagyobb teljesítményért tényleg hűzős árát kell fizetnünk. Egy GeForce 2 GTS árából akár új alaplapot is vehetnénk a legújabb processzorral együtt... Sajnos, itt is előfordulnak néha a fentebb említett sávszélességgondok, főleg gyors gépekben és nagy felbontású üzemmódban. A grafikus processzor lehetőségeit jelenleg jóformán egy játék sem használja ki.

ÁR: 90 000 forint körül.

Melyiket válasszam?

A nem is olyan korlátozott választék áttekintése után következzen néhány kérdés, amelyeket megválaszolva talán könnyebben megtalálhatjuk a céljainknak leginkább megfelelő típust.

- Fontos-e a gyors és megbízható OpenGL-támogatás? A Quake III Arena-típusú játékokhoz keresünk-e kártyát? Ha igen, akkor az nVidia kártyái közül válasszunk, mert a kártyáik alapadottsága és OpenGL-támogatása egyaránt kiváló.
- Az Unreal Tournamentet vagy a többi, Glide-alapú játékot kedveljük? Az UT Linux alatt jelenleg Glide használatával fut a leggyorsabban. Egy jó 3dfx kártya kell nekünk, ez legalább egy Voodoo3-at jelent.
- Kétmonitoros rendszert szeretnénk üzemeltetni, de nem akarjuk, hogy egy másik kártya még egy bővítőhelyet elfoglaljon? Akkor bizony a Matrox G400 Dual Head a nyerő, ez a Mesa kihasználásával igazán gyors 3D-s teljesítményt nyújt. A másik lehetőség, hogy két kártyát veszünk, azonban a Matrox sokkal ötletesebb megoldást jelent.
- Kevés pénzből szeretnénk venni egy jó 2D/3D-s kártyát?

Az ATI Rage 128 kártyáknak (az nVidia TNT2-höz hasonlóan) nagyon jó az ár/teljesítmény viszonya.

A megfelelő döntést leginkább a személyes tapasztalat segítheti elő. Mivel nem próbálhatunk ki minden kártyát ingyen a gépünkben, kérdezzük meg ismerőseinket mi a véleményük a saját kártyájukról. Ha lehetséges, próbáljuk ki a rendszert, és ezután már könnyebben kialakíthatjuk saját véleményünket is e témakörben.



Matt Matthews PhD ösztöndíjasként tanul. Tavaly óta dolgozik Linuxszal. Imád játszani, és a legújabb kártyákat kipróbálni. Amikor ideje éppen engedi, véleményeit honlapján ➔ <http://www.linuxgames.com/> is közzéteszi.

Linux Music & Sound

A legtöbb jógaelmélet a mantra szót használja az élet tényezőjének jelölésére. Egy jóga gurunak a mantra maga a hang, mely a világmindenség sarkai közt visszhangzik, a mindenség pedig maga Isten. A legtöbb általam ismert Linux guru is így vélekedik a Linux szerepéről az operációs rendszerek között. E két világ ütközése egyesek számára végzetes, mint a merevlemezre kiömlő reggeli kávé. Sokaknak azonban e terület olyannyira izgalmas, hogy még az elkövetőre sem haragszanak: legalább nem kell a munkával foglalkozni, és jöhet a tiszta szórakozás. Remélem, hogy a Linux Music & Sound elolvasása után – egy kis segítséggel – még a kezdők is merészen belevágunk majd a zenélésbe és a Linux alaposabb megismerésébe.

Hogy a lényegről is szó essék: *Dave Philips* kalandos utazásra hív bennünket a Linux és a hangkezelő programok őserdein keresztül. Indulás előtt a Linux történetével ismerkedhetünk meg, majd a szerző a legfontosabb alkalmazásokat, illetve az azok minél jobb kihasználásához szükséges gépi környezet elemeit mutatja be. Szó esik még az X Window System legfrissebb változatához járó hangalkalmazásokról is, majd elkezdődik a túra komolyabb része: a MIXViews, Snd, Kwave, Broadcast 2000 és Ceres hangfájlszerkesztők leírása. Az ezt követő fejezetekben a Mod zenei formátumot támogató programok, a MIDI, az MP3, a merevlemez hangrögzítés, a keverés, a hangképzés, a kottaprogramok, a hálózati zenei alkalmazások, a DJ digitális eszközei és a dobgépek birodalmába is ellátogatunk, és még az olyan különlegességek sem maradnak ki, mint például a más operációs rendszerek zenei programjainak felhasználása vagy a linuxos játékok.

A könyv igazi „fenevadja” a hangképzésről szóló 9. fejezet. Aki soha nem használt Linuxot és számítógépes zenével sem foglalkozott, annak e fejezet épp annyira érthetetlen lesz, mint az, hogy miért szeretik annyian a Backstreet Boyst vagy Britney Spearst. Azonban a Linux „törzsi nyelven” értők számára nyilvánvalóvá válik, hogy a Csound és kiegészítői miként segíthetik a hangfájlok létrehozását és szerkesztését, valamint a MIDI kezelést. A fejezetben a programcsomagok beállításáról, magáról a nyelvről és a külső eszközök lehetőségeiről is szó esik. A digitális stúdiófelvétel a könyv felfedezetlen területe. A terjedelmi korlátok miatt lehetetlen a hétköznapi zenészt megtanítani arra, hogy amihez egy Linux nélküli stúdióban hozzászokott, azt miként kama- toztathatná a Linux-alkalmazások használata során. Phillips e könyvével többféle olvasótábor (linuxos zenészek, linuxos programozók, nem Linuxszal dolgozó zenészek) is célba vesz, de érthető okokból nem szerepelhet az összes alkalmazás leírása a könyvben.

Néhány fejezetben a szerző már-már zavarbaejtő alapossággal magyaráz: „Győződjünk meg arról, hogy minden csatlakozó a helyén van-e, hiszen a rossz érintkezés kihagyásokat, zajt kelthet.” Később pedig feltételezi, hogy egy-egy fejezetből mindenki azonnal megértett mindent, amit a hangképzésről és a MIDI-ről tudni kell, és néhány oldallal arrébb már arról értekezik, hogy miként használhatjuk fel tudásukat egy digitális stúdióban. Azért remélem, hogy aki használt már



számítógépet, az tisztában van a dugók szakszerű kezelésével. Is- merek viszont jó néhány olyan embert, aki több CD-t is rögzített már stúdióban, de a hangképzésről vagy a MIDI-ről azt sem tudja, hogy mi fán terem. Nem ajánlom e könyvet azoknak, akik soha nem hallottak még a digitális hangrög- zítésről, illetve a linuxos és a nem nyílt forrású zenei alkalmazások közti különbségekről. A könyv és a mellékelt programok megértéséhez elkél némi linuxos (netán programozói) gyakorlat, illetve az sem árt, ha járatosak vagyunk a digitális hangrögzítésben és utómunkában. Az oroszán ketrecébe csak tapasztalt idomárok lépjenek...

Az igazán nagyszerű az a könyvben, hogy röviden bemutat minden alkalmazást. Ezeket (mivel nyílt forráskódú programokról van szó) a mellékelt CD-n megtalálhatjuk, de ingyenesen letölthetjük az Inter- netről is. A szövegben rengeteg honlap címe szerepel. Aki megérti a programok képességeit, az bizonyára nagyra értékeli majd, hogy a szerző a teljes kínálat bemutatását választotta ahelyett, hogy saját véleményét erőltetné az olvasóra.

Jómagam inkább vagyok digitális zenész, mint Linux-felhasználó, így meg tudom érteni azokat, akik nem a Linuxot használják e célra. Más programok, mint például a Pro Tools, a Logic vagy a SonicFoundry cég Acid és Sound Forge programjai használatához nincs szükség külön- sebb előképzettségre. Minden egy helyen található, és a munka során nem kell állandóan több alkalmazás között váltogatnunk. Ezen progra- mok 30 és 400 dollár közötti áron vásárolhatók meg. Természetesen ez már közel sem hangzik olyan jól, mint az, hogy „nyílt forráskódú, in- gyenes”. Ez azonban főleg a programozókat bosszanthatja. Nem érzem úgy, hogy a fizetős programok hátrálatnák a stúdiómunkát, és egy Linuxban nem járatos zenész is inkább a pénzét költi, ha ezzel megsz- hatja egy új operációs rendszerrel való ismerkedés „örömeit”.

Bár a Linuxon megjelent zeneprogramok leg többjének változatszáma még az 1.0-t sem éri el, ellenére legalább olyan minőségű és sebes- ségű munkára képesek, mint jóval drágább társaik. Ha a programo- zók lelkesedése sem hagy alább, illetve végre a zenészek is fölfede- zik maguknak a Linuxot, akkor biztos vagyok benne, hogy egyre több stúdióban állnak át linuxos programokra, hiszen ez a változás más szakterületeken már végbement.

Aki a linuxos digitális hangrögzítés és szerkesztés részletes leírását keresi, az akár le se vegye a polcra e könyvet. A Linux Music & Sound leginkább egy idegenvezetéshez hasonlít. Ha még nem dön- töttük el, hogy milyen programokat kívánunk használni, illetve hogy ezeket hol találhatjuk meg, akkor mindenképpen olvassuk el a köny- vet. Segít abban, hogy a Linux-programok e viszonylag kevésé ismert területét is fölfedezhessük.

(A könyv magyar nyelvű változatának megjelenésére már csak egy röpké hónapot kell várnunk. A szerk.)



Deric Mendes évekig tanult zeneelméletet, gitáron játszik, az utóbbi két évben pedig otthonában tart zeneórákat. Idejét az SSC-nél munkával, emellett egy saját CD felvételével és különböző zenei stílusok és hangrögzítési eljárások megismerésével tölti.

A könyv adatai

Szerző: Dave Philips
 Kiadó: No Starch Press
 E-mail: info@nostarch.com
 Honlap: <http://www.nostarch.com/>
 ISBN: 1-886411-34-4



© Kiskapu Kft. Minden jog fenntartva

Fókuszban a programok



Nemrégiben kezdtem el használni (és néhány ügyfelem számára felszerelni) az egyik legújabb és legolcsóbb hálózati kapcsolatot, a vezeték nélküli adattovábbítást. A legtöbb ilyen jellegű rendszer csak kis távolságon belül használható, de az egyik vállalat hirdetése felkeltette a figyelmemet. Azt állították, hogy az ő vezeték nélküli rendszerük akár ötven kilométeres távolságról is képes a jelet továbbítani és fogadni ➔ <http://www.ydi.com/> Az erősítőjüket még nem szereltem föl, de nekem 25 kilométerről enélkül is vígan megy az adatátvitel, néhány kisebb irányított antenna segítségével.

A dolognak persze két hátránya is van: először is, a két pont között nem lehetnek házak, fák, tehát el kell látnunk az egyikétől a másikig. Ami a kisebb baj, hogy a készülék által használt 2,4 GHz-es frekvencia szabad sáv, azaz rengeteg zavaró jelre kell felkészülnünk. Lekopogom, nekem eddig még semmi ilyen jellegű gondom nem volt. Az eszköz segítségével egyszerűen és olcsón összeköthető az iroda és azoknak az alkalmazottaknak a lakása, ahonnan légvonalban ellátni a főépületig. Az adattovábbítás hihetetlenül gyors, nálam 2 Mbit másodpercenként, szép időben. Ha esik az eső, akkor valamivel lassabb, de ez az esőcseppek nagyságától és a két pont közti távolságtól is nagymértékben függ. A rendszer ára állomásonként 600 dollár (kb. 180 ezer forint), azt mondhatjuk, hogy egy megfizethető és jól használható szolgáltatással van dolgunk. Ehhez természetesen számoljunk hozzá a fejlődőmást, ennek többirányú antennával kell bírnia, tehát a fenti árat már egy kapcsolat esetén is meg kell szoroznunk kettővel. Mi tagadás, nagyon élvezem ezt az újdonságot. Egyetlen vágyam, hogy az iroda ne egy 256 Kbit-es vonalon, hanem mondjuk T1-en csatlakozzon az Internethez... Nincs valakinek egy kis felesleges sávszélessége?

E havi témánk (na, nem mintha lenne olyan általában) az Őrület (főleg az enyém), melyet úgy ejtünk: WML. Olvassunk csak tovább, és meglátjuk, miért.

WML

Üdvözljük a WML (Web Meta Language) világában! A WML-fájlokkal egyszerűen elkészíthetünk egy egész HTML-alapú honlapot. A Engelsehall cég honlapján példákat és oktató jellegű bemutatókat is találunk, ezekkel igen gyorsan felépíthetjük HTML oldalainkat a WML segítségével. Némi programozói gyakorlat azért elkél, és itt főként a fejlécfájlokra gondolok. Ha valaki sok webes grafikai munkát végez, mindenképpen látogasson el ide. A rendszer használatához a Perlre és a következő Perl modulokra van szükségünk: Getopt::Long, Bit::Vector, File::PathConvert, Image::Size, IO::File, Term::ReadKey. Emellett az alábbi könyvtárfájlok is kellenek: libncurses, libperl, libndbm, libgdbm, libdb, libltdl, libm, libcreposix (posix), libcrypt, libpng és glibc.

➔ <http://www.engelsehall.com/sw/wml/>

Wml-tools

Ez egy gédeszköz-gyűjtemény, mely a WML-WAP programozást igyekszik megkönnyíteni. A mobiltelefonok számára kifejlesztett nyelv (Wireless Markup Language, WML) és protokoll (Wireless Application Protocol, WAP) segítségével a telefonon megjeleníthető weboldalakat készíthetünk. A WML-WAP szóhasználatában a kártya (card) egy weboldalt, a pakli (deck) pedig a weboldal gyűjteményét, azaz a honlapot jelöli. Ezen eszközök segítségével azonnal elkezdhetjük fejleszteni WML oldalainkat. A csomagban néhány példa is helyet kapott, használatához a libxml, libz és glibc könyvtárakra van szükség.

➔ <http://pwot.co.uk/wml/>

wApua

Ha úgy döntöttünk, hogy a fenti eszközcsoomag segítségével máris nekilátunk WML-WAP oldalaink elkészítésének, jó, ha tudjuk, hogy létezik egy WML böngésző is, ezzel azonnal kipróbálhatjuk a fájlokat. A WML, a WAP és az XML megjelenése óta nagyon felgyorsultak az események, és az ember örül, ha lépést tud tartani a fejlődéssel. Azt hiszem, nekem is el kellene mélyednem e témakörben, mert sohasem lehet tudni, hogy mikor következnek be a WML, a WML-WAP vagy épp az XML nagy áttörései. Addig is, ezzel az alkalmazással könnyen ellenőrizhetjük WAP-WML oldalainkat. Használatához a Perlre, valamint az LWP::UserAgent, az URI és a HTML::TokeParser nevű Perl modulokra van szükségünk.

➔ <http://fsinfo.cs.uni-sb.de/~abe/wApua/index.html>

Házi feladatok nyilvántartása

E kis webalapú program segítségével a diákok ellenőrizhetik, hogy milyen feladatokkal adóskodnak még, illetve hogy lesz-e idejük elkészülni a feladattal a határidőig. Bár a program gimnáziumok számára készült, ennek ellenére bármely oktatási intézményben használhatják, ahol található egy MySQL- és PHP-támogatással bíró webkiszolgáló. A felhasználói felülettel minden tanár könnyedén elboldogul, a diákok az igényeik alapján készült felületet pedig még ennél is egyszerűbben kezelhetik. A programot természetesen cégek is használhatják, például a határidős munkák, jelentések nyilvántartásához. Úgy vettem észre, hogy ha van egy program, mely igényeinknek majdnem megfelel, akkor csak egy kis képzelőerő és néhány apró módosítás kell ahhoz, hogy a lehetőségeket teljes mértékben kihasználhassuk. A programhoz egy PHP4- és MySQL-támogatású webkiszolgálóra, egy MySQL kiszolgálóra, valamint egy böngészőre van szükségünk.

➔ <http://www.beaconschool.org/~clehmann/projects/>

taglog

Itt van egy nagyon egyszerűen használható (a használati utasításból minden kiderül), elvégzett feladataink naplózására szolgáló programcska. A bejegyzéseket automatikusan időpontozza, a kezdő és befejezési időpontokat természetesen mi magunk is megváltoztathatjuk. Ha egy kezdési időpontot átállítunk, a program az előző befejezési időpontot is igazítja. Nagyszerűen használhatják olyan cégek, ahol a megrendelők az elvégzett munkaóra alapján fizetnek. Így minden ügyfél számára külön projektet hozhatunk létre. A programhoz Tcl/Tk-ra van szükségünk.

➔ <http://www.paladin.demon.co.uk/tag-types/taglog/>

gcdb

Ez a program egy böngésző segítségével kíséri figyelemmel vásárlóinkat, számláinkat és kifizetéseinket. Az egész nem túl összetett, nem is tud sokat, de a havidíjas szolgáltatók számára lehet, hogy pont ennyi is elég. Önálló számlákat is készíthetünk, de a program írói nem erre a feladatra szánták terméküket. Jelenleg angol és német nyelvű változata létezik, de a fordításhoz mindössze az egyik fájlban található szövegeket kell átírnunk (és a fejlesztők meg is hálálják segítségünket). A program használatához PHP4- és MySQL-támogatással rendelkező web-, illetve egy MySQL kiszolgálóra és egy böngészőre van szükség.

➔ <http://www.deepfreeze.org/gcdb/>

David A. Bandel

Középpontban a levelezés



A levelezési listák rémülettel töltik el a listafelügyelők szívét mindenhol, azonban használatuk nélkülözhetetlen. Szórákkozottatók, mulatságosak, bosszantók, időrablók és további melléknevek egész sorával illethetnénk (különösen akkor, amikor senki sem tud segítséget nyújtani egy bizonyos kérdéssel kapcsolatban a listáról). A legjobb dolog, amit megállapítottam arról a néhányról amelyre feliratkoztam, hogy oktató jellegűek. Nincs sok időm rájuk és néha siránkozom amikor többnap postát kell kitörölnöm anélkül, hogy akár a címüket is elolvassam (hálás vagyok a szűrőkért, mert nem szeretem elszalasztani az olvasói leveleket). Amennyiben nem olyan listán vagy, mely ellenszenvet érez a kereskedők vagy azok iránt, akik kíméletlenül tűzbe hozzák a kezdőket, valószínűleg állandó adatáradatban részesülsz. Tehát ne írjuk le olyan könnyedén a levelezési listákat. Évekkel ezelőtt, egy levelezési listán való jelenlétemnek köszönhetően felkérést kaptam egy könyv megírására, továbbá számos szerződéshez jutottam. Óvatosan válaszd ki a levelezési listát, írd közérthetően és érezd jól magad. Ne felejtse el, hogy a többiek csak a leveleiden keresztül ismernek.

phpPhotoAlbum

Ha van egy köteg fényképed (jpeg, gif, png, psp és más formátumok) amit szeretnél közzétenni a weben, a phpPhotoAlbum a segítségére lesz. Az egyetlen hiányossága az, hogy nem láthatod a fényképed előnézeti képét, mielőtt megnéznéd az egészet. Valószínűleg a 100 K-nál kisebb képek esetén nem olyan nagy gond ez, de az igazán nagyméretűeknél az előnézeti kép hasznosabb lenne. Ha már állítottál be valaha más PHP-alkalmazást, akkor a phpPhotoAlbumot könnyű lesz elindítani. Szükséges: internetkiszolgáló PHP-támogatással, webböngésző és képek.
[➔ http://www.phpphotoalbum.com/home/](http://www.phpphotoalbum.com/home/)

pemail

Vajon hány ember használta valaha is a Unix-rendszeren belül a „mail” parancsot? Én nagyon kedvelem ezt az utasítást, bár akad néhány hiányossága is. Nem utolsósorban az, hogy a levelezési listát csak a lokál hostról nézi meg. Ez rendben is van akkor, ha hálózati fájlrendszert (NFS) használsz azért, hogy növelj a levelezési szervered teljesítményét, de ha a beérkező leveleidet egy internetes szerverről olvasod, akkor a unixos „mail” parancsnak nincs akkora jelentősége. A Perl Mail (pemail) folyamatosan beolvassa a leveleidet egy távoli szerverről és megmutatja azt is, hogy hány leveled érkezett, valamint egy lista visszaküldi az üzenetek számát és méretét (ez igen hasznos, főként akkor, ha nem akarsz letölteni nagyméretű üzeneteket). A lista azonban nem ír vissza. Amikor a programfejlesztő a pemailt egy tárgysorral és még talán egy küldősorral is bővíti, akkor ez egy fárasztó segédprogram lesz (legalábbis a parancssort kedvelő szakemberek számára). Szükséges: Perl és Perl modulok Mail::POP3Client, MIME::Lite, Term::ReadKey.

➔ <http://www.the-den.org/pemail/>

nettop

A nettop használata kötelező minden hálózatzfelügyelő számára. Majdnem mindegyik rendszeren futtatható, még a RAM vagy CPU kihívásos rendszereken is, és jó képet kaphatsz a hálózati forgalomról. Az oszlopok a csomagok méretének százalékát, valamint a csomagok teljes számát, az átbocsátóképesség százalékát, a teljes átbocsátóképességet, az átlag csomagméretet és a csomagtypust mutatják. A sorok

a forgalomtípus alapján színekódoltak. Az ipv4 és az arp például bíbor, az ipv4, mely udp, tcp és icmp-re bomlik, zöld. A protokollokat, úgy mint a http-t, az ssh-t és a POP3-at stb. a kék szín jelzi. A képernyő tetején láthatod a jelenlegi rendszeridőt és dátumot, valamint azt, hogy a nettop mióta fut. Véleményem szerint csak egyetlen hiányossága van ennek a programnak, az adatokat rendszeres időközönként egy adatállományba kellene tömörítenie.

Szükséges: libslang, libpthread, libdl, libm, glibc.

➔ <http://srp.portico.org/scripts>

phpweather

Szeretnéd figyelemmel kísérni az időjárás? A phpweather leolvassa a NOAA által közzétett METAR/TAF adatokat és megjeleníti azokat a böngészőn. Választhatasz egy rövid kijelzés (feldolgozatlan METAR adatok) vagy a létrehozott, „emberi” fogyasztásra alkalmas oldal (inkább, mint a meteorológiai halandzsa) között. Az egyetlen hibaforrást a nem-szabványtípusú METAR adatok jelentik. Ha amerikai vagy európai állomások adataihoz hozzáférsz, akkor nem lehet gond, de néhány ország nem megfelelő formátumú adatokat küld vissza. Szerintem, a Nemzeti Oceanográfiai és Atmoszferikus Felügyelet (NOAA) úgy véli, hogy a helytelenül feldolgozott adatok még mindig jobbakk, mintha egyáltalán semmilyen adatot nem érhetünk el, de ez kétségtelenül fárasztóssá teszi a dolgok elemzését.

Szükséges: PHP és MySQL támogatású internetkiszolgáló.

➔ <http://www.gimpster.com/php>

➔ www.met.hu

scanSSH

Nagy pártolója vagyok a titkosításnak. Kódolom az Interneten küldött adataim legtöbbjét, sőt, a saját gépemén lévő anyagokat is. (Bárkitől származik is ez a mondás: „A Linux használata azt jelenti, hogy soha többé nem kell kitörölnöd a szerelmes leveleidet”, az illető tudta, hogy miről beszél.) Mindenesetre fontos, hogy az általam fenntartott gépek mindegyikén a legfrissebb OpenSSH-programok fussanak. Ehhez nyújt segítséget a scanSSH, mellyel gyorsan megállapíthatom, hogy kiszolgálóim friss anyagot tartalmaznak-e vagy valamelyiken elfelejtettem frissíteni valamit. A program nemcsak a biztonságos héjat ellenőrzi, hanem segít a gépen futó egyéb szolgáltatások számbavételében is.

Szükséges: glibc.

➔ <http://www.monkey.org/~provos/scanssh/>

SING

A SING (Send ICMP Nasty Garbage – Küldj gonosz ICMP-szemetet) segítségével sokkal többet tehetsz, mint pinggel. Pingelhetsz (küldhetsz ICMP-visszhangkérelmet) ha akarsz, de ennél sokkal többet is művelhetsz. Néhány netszűrő-beállítás ellenőrzésére kezdem el használni. A kísérő man oldal rendkívül teljes körű és igen jól elmagyarázza a SING használatát. A SING továbbá támogatja nagy adatcsomagok használatát (ne feledkezz meg arról, hogy nem ez a javasolt módja a sebezhető rendszereid megtalálásának), valamint a forrásátírást és a forrásátírást (source routing, source rewriting). Valójában alig létezik valami, amit ne tudnál átírni ezzel az eszközzel az ICMP fejlécben.

Szükséges: libnsl, libresolv, glibc és af_packet.

➔ <http://www.sourceforge.net/projects/sing/>

David A. Bandel