

A könnyed megoldások hónapja

Nagyon sokan foglalkozunk közvetve vagy közvetlenül a GNU/Linuxszal. Aki már hosszabb ideje használja, tudja, hogy – bár sokszor komoly szenvedés árán, de – rendkívül sok feladat megoldására használható. Ahogy a csomagok fejlődnek, ahogy az éles használat „kiköveteli” a finomításokat, egyre nehezebb feladatokat tudunk könnyedén megoldani. E havi számunkban sok-sok ilyen „könnyű megoldást” szeretnénk bemutatni.

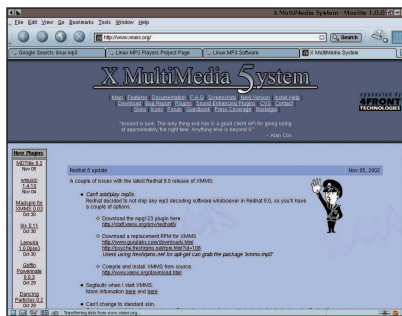
Itt van egyből a kapcsolat kérdése. Ma már szinte bárki könnyedén kapcsolódhat az Internethez vagy éppen a cég egy távoli telephelyéhez. Igaz, a kapcsolat nem mindig alkalmas videofonozásra, de legalább működik. Az egyik számomra legérdekesebb cikk például azt meséli el, hogy miként építették ki egy egyébként teljesen elavult rendszerből a világ végén remekül használható levelező hálózatot (Elektronikus levelezés rádióval Nyugat-Afrikában, 30–33. oldal).

Jellemző példa ez az emberi leleménységre. A minap a hazai helyzetről beszélgettem az IHM egyik munkatársával, és úgy éreztem, hogy pont ebbe az irányba szeretne a minisztérium is elmozdulni: nem az a cél, hogy egyes helyeken rendkívül jó gépek legyenek, hanem azok segítése és támogatása, akik használni is szeretnék őket. Érdekes programok indultak még korábban, amelyek lassan beérnek. Ezekről a kérdésekről azonban majd a januári számunkban írunk részletesebben.

Visszatérve a megoldásokra, ebben a hónapban Marcel is egy remek megoldást boncolgat. Nagyobb hálózatoknál értelmetlen minden egyes munkaállomást külön-külön telepíteni, karbantartani. Ehelyett a munkagépeken csupán egy grafikus felület futtassunk, a programokat pedig egy központi gépről kérjük. A felületet akár egy 486-oson is vígan elfuttathatjuk, de akár „lemeztelenített” ügyfeleket is használhatunk – elnézést a szöviccért.

Természetesen mindig nagy kérdés a nyelvek használata is. Idehaza is érezzük ezt, nem is kell messzire menni példáulért, elég csak az egymás mellett készülő OpenOffice.org-magyarításokra vagy a grafikus felületek egymástól

különböző fogalmaira gondolnunk. Ehhez hozzávehetjük a folyamatos ütközési felületnek tekinthető jövevény-szó-magyar szó vitákat is. Ismét csak Afrikába ugrunk át egy pillanatra, pontosabban Dél-Afrikába, ahol nem elég, hogy politikailag, de emberileg is nehezen kezelhető a helyzet, és nem csupán két nyelvről, hanem egyből tizenegy különböző nyelv egymás mellett éléséről van szó! Erről szól a 28. oldalon kezdődő Digitális szakadék áthidalása Dél-Afrikában című cikk.



De ne csak a nagyvilággal foglalkozunk! Idehaza is történt néhány érdekesség. Sőt! Hadd vegyem előre a 26. oldalon található beszámolót, ami-ben igyekeztem összefoglalni az Internet Hungary 2002 egy-két érdekes mozzanatát. Izgalmas volt látni, ahogy szakemberek, politikusok, gazdasági emberek és jogászok igyekeztek több fontos kérdésben megoldásra jutni. Azért írom, hogy izgalmas volt, mert számomra itthon, Magyarországon szokatlan jelenség, hogy több értelmes ember összegyűlik, és veszekedés nélkül sokáig képes vitatkozni egymással. Vannak megoldatlan kérdések, ezt senki sem vitatja. Ilyen például a Pedagóguspályázat, amelyről Szaló István kérdezte *Suhajda Attilát* (10. oldal). Szerencsére az idei GNU/Linux Szakmai Konferencia is jól zárult, a rendez-

vényről *Gibizer Tibor* ír a 12. oldalon. Sokan írták nekünk levélben, hogy most kezdik az ismerkedést a Linuxszal, és segítsünk abban, hogyan lehet ezt a rendszert a máshol megszokott kellemes tevékenységekre használni. E hónapban több ilyen érdekes téma kapott helyet: *Garzó András* a zenei lemezek és egyéb hangfájlok MP3-má alakításával foglalkozik, valamint a GNU/Linuxban található ALSA rendszerrel (44. oldal); *Fülöp Balázs* gyors segítséget nyújt a CD-lemezek írásához (46. oldal) *Varga Csaba* pedig elmagyarázza nekünk, miért is szerencsésebb Linux alól intézni számlaügyeinket, mint délutánonként sorban állni a bankokban.

A hálózati szakembereknek is tartogatunk érdekességeket! Körüljárjuk a hálózati biztonság témakörét, ismét elővesszük az IP Tablest (Biztonság a helyi hálózaton: IP Tables, 55. oldal), amivel sokszor fogunk még találkozni a lap hasábjain, hiszen rendkívül fontos eleme a hálózatba kötött linuxos rendszereknek. Foglalkozunk egy kicsit a jövővel, a már régóta létező, de még – valljuk be – sokáig csak érdekességként jelen lévő IPv6 megvalósításáról is írunk (IPv6-támogatás linuxos hazai csomópontokon, 62–66. oldal). Bevallom, már úgy hozzászóltam a négybájtos IP-címekhez, hogy megemelem a kalapom azok előtt, akik a jövőben a kisujjukból kirázzák a megfelelő címetek...

Szerencsére erre nem is lesz szükség, de erről részletesebben a 62. oldalon! És az egyik „komoly veszélyre” is kitérnünk, a hazai hálózatot többször is leültető nagytesó-hatásra: hogyan oldhatjuk meg a netes műsorszórásat kisebb hálózati terhelés mellett? Ehhez nyújt segítséget a csoportszórásokat (multicast) támogató linuxos útválasztó.

Úgy érzem, ez a lapszám igen sokszínűre sikerült, szép búcsú lesz a 2002-es évtől. Mindenkinnek örült ajándékvásárlást és családi békében eltöltött boldog karácsonyt kívánok!



Szy György
a Linuxvilág főszerkesztője,
a Kiskapu Kiadó vezetője.
Mindenki levélét örömmel
vár a következő levélcímen:
Szy.Gyorgy@linuxvilag.hu

Programvadászat

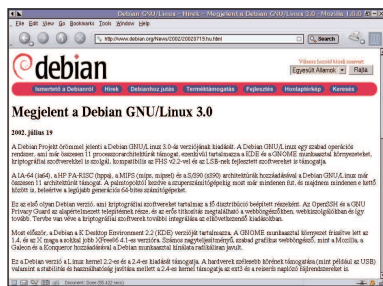


© Kiskapu Kft. Minden jog fenntartva

Ehavi mellékletünkön közkívánatra a Debian Woody első telepítőkorongját adjuk közre. Mivel ez teljesen kitölti a mellékletet, a magazinhoz tartozó anyagok sajnósan csak a következő, azaz a januári korongon lesznek elérhetők, illetve a <http://www.linuxvilag.hu> internetes oldalunkon lehet a fontos forráskódokat, illetve kiegészítéseket, valamint testvér-lapunk, a Linux Journal olvasói szavazásáról szóló cikket megtalálni.

Debian GNU/Linux 3.0 Woody

A fentebbi elnevezés biztosan sokaknak megdobogtatja a szívét, ahogy az enyémet is. 2002. július 19-én végre megszületett a nagy mű – egy újabb, frissebb, jobb rendszert kaptunk kézhez. A leg-



fontosabb, hogy végre nem azokat az ósdi csomagokat kell használnunk, amelyek a két évvel ezelőtt kiadott Potatokiadásban megkeserítették az ember életét. Gondolom, sok türelmetlen ember, mint amilyen jómagam is vagyok, már javában használta a Woody-kiadást a próbaváltozat (testing) óta, akár éles rendszereken is, ahol a biztonság esetleg nem a legfontosabb tényező, annál fontosabb azonban a csomagok frissessége. Nos, a Debian-kiadások mindig sokáig készültek, azonban ha megszülettek, egy nagyon kellemes, összefogott és átlátható rendszert kaptunk. Így volt ez a Woody esetében is, kiforrott, hatalmas programkészlettel rendelkező operációs rendszert kaptunk kézhez a hét CD-t magában foglaló telepítőkészlettel. Természetesen a rendszer telepítéséhez nem az összes korongra van szükségünk, az alapvető kiszolgáló- és felhasználói programok az első korongon kaptak helyet, így a rendszer fellesztése nem jelenthet gondot. Hasonlóképpen,

mint a Potatónál, itt is választhatunk a telepítés közben használandó rendszer-magváltozatok közül. Az alapértelmezett rendszermag még mindig a 2.2-es sorozat tagja, de a 2.4-es rendszermagot is választhatjuk, amivel már ext3- és ReiserFS-támogatást is kapunk – ezek a szolgáltatások egy mai rendszerben alapvetőek. A 2.4-es rendszermagot a bf24 „varázsszóval” hívhatjuk elő a telepítő bejelentkező ablakában.

Csomagok az első korongon

Mint fentebb is említettem, a legtöbb kiszolgálóprogram megtalálható ezen a korongon. Az alábbiakban ezekről egy korántsem teljes összegző felsorolást adok:

- Apache – a legnépszerűbb webkiszolgáló;
- Apache-ssl – a legnépszerűbb webkiszolgáló titkosított kapcsolatokat kezelő változata;
- Exim, Courier, Sendmail, Smail – levélkiszolgáló;
- PostgreSQL, MySQL – adatbázis-kiszolgáló;
- ssh – biztonságos távoli elérésre használt program („titkosított telnet”)
- ftpd – FTP-kiszolgáló;
- Bind, Bind9 – DNS-kiszolgáló;
- különféle IMAP- és POP3-levélkiszolgálók.

A felhasználók szemszögéből a KDE2, a Mozilla, az Abiword, az Aptitude, a wget, a Lynx és a Midnight Commander programok jelenthetik az alapot (természetesen ez nem teljes programlista, úgyhogy nyugodtan tessenek böngészgetni a CD-n).

Új korongok hozzáadása

Ha beszerezzük a többi CD-t, akkor ezeket is hozzá kell adni a telepítési forráshoz. Ki kell adjuk az `apt-cdrom` add parancsot, és ezt mindaddig ismételtessük, amíg az összes CD-t be nem olvastuk. Ha mind a hét korong szerepel a listánkban, akkor bizony a csomagok száma meghaladja a 8000-es határt, ami nagyon nagy választékot kínál ahhoz, hogy mindenki a saját szájíze szerint szabja teste rendszerét. Ilyen bőséges választék mellett még én is el szoktam tévedni a csomagok érdekében. Mivel az ember nem szívesen

jegyezze meg ennyi csomagnevet és azt, hogy melyik csomag mire is való, ezért ilyenkor jól jön egy ügyes kis parancs, aminek segítségével gyorsan kereshetünk a csomagok között. Adjuk ki a következő parancsot: `apt-cache search amit kereselek`. Az eredmény ugyan elég elrettentő lehet, de ne ijedjünk meg tőle. Ha például webmail programot keresek, elég kiadni az `apt-cache search webmail` parancsot, és máris egy rakás program közül választhatunk. Keresési feltételnek több szót is megadhatunk, ekkor változatos eredménnyel, több-kevesebb sikerrel egy szűkített listát kapunk a képernyőre. Akik a sokkal kényelmesebb grafikus felületre vágnak azoknak a KDE rendszer `kpackage` programját ajánlanám. Ebben a kereséseket és telepítéseket is könnyedén gyorsan, mindenféle „varázslást” mellőzve tehetünk meg.

Grafikus telepítés

A Woodyról bővebben a 2001. novemberi számban a 67. oldalon és a 2002. októberi számban a 44. oldalon olvashatnak. A Progeny cég fejlesztői kiadták a Woodyhoz a grafikus telepítőprogramot, amit a cég internetes oldaláról lehet letölteni (94,5 MB). Sajnos nem volt még alkalmam kipróbálni, úgyhogy bátorítani senkit nem tudok a használatára.

Ez ugyan nem sok, de az induláshoz éppen elég!



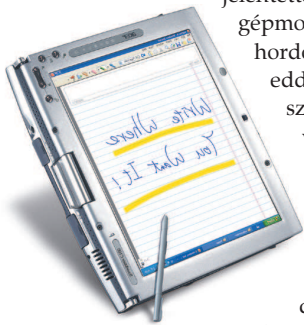
Csontos Gyula
(Csontos.Gyula@linuxvilag.hu) a Linuxvilág szakmai és CD-szerkesztője. Szabadidejében szívesen mászik hegyet és kerékpározik.

Kapcsolódó címek

- ➔ <http://www.debian.org>
- ➔ <http://www.debianhelp.org>
- ➔ <http://www.debian.hu>
- ➔ <http://www.debianuser.org>
- ➔ <http://www.progeny.com/news/pgi-woody102302.html>

Megjelentek az első táblagépek

A Microsoft Windows XP Tablet PC Edition operációs rendszerének megjelenésével világszerte számos gyártó jelentette be első tábla gépmo­delljeit. Az új hordozható gépek eddigi társaikkal szemben már valóban notesz­gépek nevezhetők, hiszen érintőképernyőjük oldalra is elforgatható, kifordítva



pedig a billentyűzetre fektethető, így valójában egyetlen színes érintőképernyőt tarthatunk a kezünkben. Ettől, illetve a különleges képességek használatát segítő gomboktól, kiegészítőktől eltekintve a gépek normál hordozható számítógépek, bár általánosságban elmondható, hogy mindenből a legújabb, legnagyobb teljesítményű került a belsejükbe.

☞ <http://www.tabletpc.com>

Bluetooth – 802.11b kártya

Az MSI a Cambridge Silicon Radio (CSR) lapkájára épülő, kombinált, Bluetooth és 802.11b kapcsolatok létesítésére egyaránt használható kiegészítő



kártyát jelentett be. A modul a cég 845PE Max2 jelzésű alaplapjaival lehet kapni, illetve külön is beszerezhető.

Az MSI egyes alaplapjait korábban ugyancsak a CSR egyik lapkáját felhasználva tette képessé Bluetooth-kapcsolatok létesítésére. Az új, kombinált lapkával elérhető áron válik képessé az egyre népszerűbb vezeték nélküli hálózatok, illetve vezeték nélküli egerek, billentyűzetek vagy nyomtatók használatára, a Bluetooth-kapcsolat segítségével pedig például mobiltelefonos mome­dmezésre nyílik mód.

☞ <http://www.msi.com>

☞ <http://www.csr.com>

10 Gb/s sebességű vezeték nélküli hálózat

A japán NTT kísérleti laborjában 10 Gb/s sebességű vezeték nélküli rádiós hálózati kapcsolatot sikerült létrehozniuk a kutatóknak, ami pontosan négyszeres gyorsulást jelent tavalyi, 2,5 Gb/s sebességű hálózatukhoz képest. Az összeköttetés a jelenleg használaton kívüli 120 GHz-es hullámhossza, pontosabban egy 17 GHz-nyi sávban épült fel, ASK modulációval. Az ASK moduláció meglehetősen egyszerű, később bonyolultabb eljárásra cserélve további sebességnövekedést lehet majd elérni.

Hasonló sebességeket értek el az NTT fejlesztői optikai átvitelrel is. Tesztrendszerük a cég bemutatóján 1,5 Gb/s sebességű, tömörítetlen HDTV-jelet továbbított, egyelőre tíz méteres távolságra. A két egység közötti távolságot a közeljövőben száz méterre növelik, így hamarosan valószínűleg a gyakorlati alkalmazásokban is hasonló sebességet lehet elérni.

☞ http://www.sctlg.ecl.ntt.co.jp/index_e.html

Lycoris Linux a Wal-Mart gépein

Az amerikai óriásáruház-lánc – avagy óriás áruházlánc – alsó kategóriás gépeinek kínálatát újabb linuxos géppel bővítette. A Wal-Mart volt az első olyan nagyobb kereskedőcég, amelyik operációs rendszer nélküli gépeket kezdett el árulni üzleteiben. A tengerentúlon – hazánkkal ellentétben – ez nem nagyon divat, ám az előre telepített rendszer



jellemzően Windows, ami nem sok esélyt ad a Linuxnak. Hasonlóan fontos lépés volt, amikor a gépekre Lindows, később pedig Mandrake Linux került. Az áruházlánc az ingyenes operációs rendszereknek és az egyéb olcsó alkatrészeknek köszönhetően már 199 dolláros gépet is kínál vásárlóinak (monitor nélkül). A gépekre Via C3 processzor, 128 MB memória, 10 GB-os merevlemez és 52xCD-ROM-meghajtó került. A Lycoris a Microsofthoz hasonlóan redmondi székhelyű, a céget egy volt Microsoft-tesztelő alapította. Terjesztésük, a Lycoris Desktop/LX arculata meg­tevesztésig hasonlít a Windows XP-re – állításuk szerint ők nem rémülnek meg attól, hogy átvegyék a jó dolgokat a Microsoft termékeiből.

☞ <http://www.lycoris.com>

Asztali DiVX-lejátszó

A KISS Technology – meglepetés! – nem valami vagány, Szilícium-völgybe letelepedett, a kockázati tőkebefektetők pénzéből villogó cég, hanem egészen közel, Dániában székel. Ehhez képest lenyűgöző, hogy elkészítették az első olyan asztali készüléket, amely DVD-k és CD-k mellett DiVX-fájlokat is le képes játszani. Természetesen tudása nem ennyiből áll, a DP-450-es készülék előlapján arasznyi hosszan sorakoznak a támogatott formátumok és adathordozók logói, a DVD+R és



DVD+RW lemezekon kívül gyakorlatilag mindenféle korongot elolvas. Formátum tekintetében – a gyári hanglemezek és DVD-k mellett – lényegében minden ISO-MPEG-4 szabvány szerinti kodeket támogat, így néhány korlátozással képes olvasni a DiVX-, Xvid- és Realmagic- RPM4-állományokat. Tudása nem tökéletes, hiszen nem rendelkezik beépített AC3-dekóderrel, és a változó bitebességű MP3-fájlokat sem szereti, de belső programja frissíthető, így az új formátumok támogatása is megoldható. Ne legyünk hát kukacoskodók, a készülék dögös, már csak ki kell várni, amíg hazánkba is megérkezik.

☞ <http://www.kiss-technology.com/>

ACM programozási verseny

Az ACM minden évben meghirdeti az ACM nemzetközi programozási versenyt, így történt ez idén is. Az október 26-án tartott válogató versenyen a háromfős csapatok 8 feladattal küzdöttek meg, változó sikerrel. Az első három



csapat Ausztria, Csehország, Horvátország, Lengyelország, Németország, Szlovákia és Szlovénia győztes csapataival mérkőzik meg a november 15–17-én a Varsói Egyetemen, a területi döntőben. Aki itt megszerzi a győzelmet, annak már csak a döntőben, jövő év tavaszán, a kaliforniai Beverly Hillsben kell helyt állnia.

☞ <http://www.cs.bme.hu/acm/>

☞ <http://icpc.baylor.edu/icpc/>

MySQL NetWare alatt

A Novell és a MySQL AB közös bejelentése szerint a Novell a MySQL adatbázis-kezelőnek a NetWare rendszerre hatékonyra tett változatát szállítja hálózati operációs rendszerének jelenlegi, 6.0-s és jövőbeni változataihoz. Az együttműködés keretében a NetWare-felhasználók a világ legnépszerűbb nyílt adatbázis-kezelő programjához jutnak hozzá, ám úgy fejlesztethetnek alá alkalmazásokat, hogy nem kell igazodniuk a nyílt forrású rendszerek szerződéseinek feltételeihez, tehát forráskódjaikat nem kell bárki számára elérhetővé tenniük. A MySQL for NetWare csomagnak része az Apache, a PHP és a Perl is, így a más rendszerekre írt alkalmazások módosítás nélkül átemelhetők NetWare-környezetbe. A csomag egyelőre kipróbálásra végleges változatban decembertől érhető el. <http://www.novell.hu>

DVD-íróval szerelt PowerBook gépek

Az Apple alaposan megújította PowerPC G4 processzoros Titanium PowerBook sorozatát, amikor beszívós – azaz tálcánélküli – DVD- és CD-író készülékkel látta el gépeit. Ha figyelembe vesszük, hogy a hordozható gépek az ATI Mobility Radeon 9000 3D vezérlőjét kapták, képesek vezeték nélküli vagy Gigabit ethernet hálózati kapcsolatok létesítésére, kijelzőjük akár 15,2" átmérőjű is lehet, vastagságuk pedig mindössze 2,5 cm, a 600 ezer forinttól induló árát kedvezőnek is nevezhetjük. A gépekbe kerülő, legfeljebb 1 GHz órajelű, 1 MB harmadszintű gyorsítótárral felszerelt processzorok a cég állítása szerint felveszik a versenyt a leggyorsabb Intel Pentium 4 processzorokkal is. Az IBM eközben új, személyi számítógépekbe szánt, 64 bites processzort jelentett be, amellyel az elemzők szerint hamarosan az Apple gépeiben találkozhatunk; bár ezt az almás számítógépgyártó nem erősítette meg. Az új PowerPC processzorok órajele 1,8 GHz-ről indul majd, és gyártásuk jövő év vége felé indulhat be – még meglátjuk, hogy az Apple kivárja-e ezt, vagy valóra válnak a pletykák, és Intel-alapokra vált át. Az IBM szerint az új PowerPC 970-es lapkák a Regatta nevű nagygépekben zakatoló Power4 processzorok egyszerűsített változatai lesznek. Az x86-64 processzorokhoz hasonlóan a 32- és 64-bites alkalmazások futtatására egyaránt képesek lesznek, és öröklik a Power4-elődök energiatakarékos jellemzőit is.

SuSE Linux Office Desktop áttérőknek

A SuSE januárban szeretné megjeleníteni kis- és középvállalkozások számára összeállított terjesztését SuSE Linux Office Desktop név alatt. A terjesztés a 8.1-es SuSE Linuxra alapul majd, ám része lesz a CodeWeavers CrossOver Office 1.2 is. A cél könnyen kitalálható: Linux alatt is futtathatóvá válik a Microsoft Office és Lotus Notes, így a felhasználók számára a Linuxra való áttérés kisebb gonddal jár. A terjesztés 129 dolláros áron lesz megvásárolható, ebben a 90 napos támogatás ára is benne foglaltatik. A CodeWeavers a Lindows készítőivel is szorosan együttműködött, a Lindows sikere mégis meglehetősen korlátozott volt. Mivel az Office csomagot ebben az esetben is meg kell vásárolni, kétséges, hogy a SuSE bukni vagy nyerni fog-e az ötleten. Az Office 80 ezer forinttól induló ára mellett egy 30 ezer forintos – azaz a SuSE terjesztésével nagyjából azonos áron kapható – Windows megvásárlása ugyanis viszonylag kis kiadást jelent, így látszólag nem sok értelme van egy Linux-Office párosításnak – hacsak nem az, hogy a felhasználók megteszik az első lépést az áttérés felé: ha már linuxoznak, akkor semmi nem akadályozza meg őket abban, hogy megismerkedjenek az OpenOffice vagy az Evolution képességeivel, esetleg meg is szeressék őket. <http://www.suse.com>

Kroupware a KDE fejlesztőitől

A KDE új tervezetet jelentett be, amelynek célja egy csoportmunka-alkalmazás (groupware) kifejlesztése. Az alkalmazást a KDE-hagyományoknak megfelelően – nem túl ötletesen, de legalább következetesen – Kroupware névre keresztelték. A KDE környezet már most is rendelkezik számos olyan programmal, amely az irodai munkavégzést segíti, így találunk benne levelezőprogramot, határidőnaplót, címjegyzéket, jegyzetkönyvet, Palm szinkronizáló programot, ám ezek nem állnak össze a Microsoft Outlookhoz hasonlóan egységes csomagba. A kérdés csak az, hogy a Ximian egy ideje már létező Evolution programja mellett érdemes-e felaprózni a fejlesztők idejét és energiáját egy újabb Outlook-klón készítésére? Kétségtelen, a választás lehetősége vonzó lehet a felhasználók számára, és az Evolution sem tökéletes, ám a versengés csak addig jó, amíg valóban versengés, és nem háborúskodás. <http://www.kroupware.org>



© Kiskapu Kft. Minden jog fenntartva

Palm Tungsten W

Új, saját termékvonalaán belül felső kategóriásnak számító modellt jelentett be a Palm. A Tungsten W az első olyan Palm kézigép, amely beépített GSM/GPRS adóvevővel rendelkezik, és a cégtől szokatlan módon beépített billentyűzettel is ellátták. Kijelzője a kor követelményeinek megfelelően színes. Ha megfelelő előfizetéssel rendelkezünk, a segítségével elintézhjük elektronikus levelezésünket, SMS-eket küldhetünk és fogadhatunk, mobiltelefonként használhatjuk, illetve letudhatjuk a megszokott személyes adatkezelő feladatokat.

A Palm ugyanakkor szövetséget kötött az USA, Kanada, Európa (Vodafone) és Ázsia egy-egy nagy mobilszolgáltatójával, a közös cél adat- és hangátviteli szolgáltatások biztosítása vezetékes, mobil- és internetes alapokon egyaránt. Az új Palm eleve számos kényelmi szolgáltatást nyújt, ezeknek a köre különféle kiegészítőkkal tovább bővíthető. Kezeli az SD, SDIO és MMC kártyákat, kapható hozzá kamera, Bluetooth és IEEE 802.11-es csatoló, GPS-vevő és MP3-lejátszó. Ára 549 dollár lesz, ha a jövő év elején az üzletkebe kerül.

➔ <http://www.palm.com/products/handhelds/tungsten-w/index.html>

Real-programok nyílt forrással

A RealNetworks médialejátszójának forrása mindenki számára elérhetővé válik – a cég ezzel a lépéssel szeretné ösztönözni a fejlesztőket ingyenes lejátszóprogramok készítésére, Windows, Mac OS X és Linux-rendszerek alá egyaránt. A cég további

termékeket is nyílttá kíván tenni Helix néven futó kezdeményezése keretein belül, a jövő évben egy nyílt forrású médiakiszolgálóra, illetve hang- és mozgóképfájlok tömörítésére használható megoldás is hozzáférhető lesz.

A Helix DNA-ügyfélprogram gyakorlatilag ugyanarra a motorra épül, mint a RealOne lejátszó. Az alkalmazás különféle eszközökön, például mobiltelefonokon, set-top-boxokon és számítógépeken egyaránt használható. A RealNetworks elsőként asztali operációs rendszereken szeretne ingyenes lejátszókat látni, majd némi módosítás árán Symbian vagy PalmSource rendszerekre is továbbterjeszkedni. A cél természet-

sen nem túl fennkölt: a RealNetworks így próbálja a saját médiaformátumát előnyhöz juttatni a Microsoft és az Apple által fejlesztett formátumokkal vívott harcban.

Éppen ez indokolja, hogy a fejlesztők számára a cég amúgy fizetős tömörítő és kibontó megoldásai is hozzáférhetők lesznek, ám ezek csak a különféle rendszerekre előre lefordított formában – forráskódként nem lesznek beszerezhetőek. A forráskód kétféle szerződéssel lesz elérhető. Aki nem kereskedelmi jellegű programot fejleszt – itt elsősorban az egyéni programozókra és az egyetemekre gondolnak –, az teljesen ingyenesen érheti el a kódot. Aki kereskedelmi célból készíti az alkalmazását, az egymillió eladott példány felett díjat köteles fizetni. A Helix-közösség máris kétezernél több taggal dicsekedhet.

➔ <http://www.helixcommunity.org>

➔ <http://www.realnetworks.com>

Állami informatikai pályázatok

Október közepén több pályázat is felkerült az Informatikai és Hírközlési Minisztérium honlapjára. A pályázatok túlnyomó részéből inkább az önkormányzatok, múzeumok, kulturális területen tevékenykedő nonprofit szervezetek csemezhetnek, ám néhány téma



mások számára is érdekes lehet.

Támogatják például az olyan tartalmak fejlesztését, amelyek az információs társadalom térnyerését segítik elő, akár kiadványok, akár műsorok formájában. Támogatást igényelhetnek színvonalas hálózati tartalmat már szolgáltató, ám a fejlesztésekhez segítséget igénylő szervezetek is.

Ugyancsak szép törekvés a fogyatékos emberek életminőségének javítását célzó program, amely szintén hálózati, közhasznú tartalmak, a fogyatékos emberek beilleszkedését elősegítő kiadványok készítését próbálja ösztönözni.

A Foglalkoztatáspolitikai és Munkaügyi Minisztérium és az IHM közösen igyekeznek fellendíteni a távmunka terjedését, ezzel javítva a hátrányos helyzetűek esélyeit a felzárkózásra. A kísérleti program 1000 munkahely létrehozását célozza, a szükséges eszközök beszerzésétől a bértámogatáson keresztül a képzésre, betanításra egyaránt igényelhető támogatás. ➔ <http://www.ihm.gov.hu>

Intel hálózati processzor beépített titkosítással

Az Intel új, IXP2850 jelzésű hálózati processzorával jelentősen növelhető a hálózatok biztonsága. A lapka nemcsak nagysebességű hálózati adatfeldolgozásra, csomagtovábbításra képes, de támogatja a virtuális



magánhálózatok, a nagy terhelésnek kitett webkiszolgálók és tárolóhálózatok üzemeltetését is. A terhelést több webkiszolgáló között képes elosztani, támogatja a 3DES és AES titkosítási algoritmusokat, valamint az SHA-1 kivonatólást – akár 10 Gb/s adatátviteli sebesség mellett. Segítségével az állapot- és egyéb adatok megoszthatók az alkalmazások között, felhasználásával a korábbiaknál kisebb helyet és kevesebb energiát igénylő készülékek tervezhetők.

A lapka – amely a termékcsalád korábbi tagjainak helyére további átalakítások nélkül is beépíthető – biztonsági szolgáltatásait az Intel fejlesztői eszközei is támogatják. Tudása bármennyire is vonzó, nagy tételben vásárolva is 180 ezer forint darabja, így egy ideig valószínűleg még nem lesz az amatőrök játékszere.

➔ <http://www.intel.com/design/network/products/npfamily/index.htm>

Jóéjt-párna

Az egyébként különféle ágyneműk, lepedők gyártásával foglalkozó távolkeleti Lofty a Matsushitával összefogva fénykibocsátó párnát tervezett. A cég állítása szerint az ember bioritmusa nagymértékben függ a fénytől, ezért a 20 darab fénykibocsátó diódát tartalmazó párna, amely tulajdonosának légzésétől függően fényesedik ki és sötétedik el, sokat segíthet a pihentető, nyugodt alvásban. A lámpácskák körülbelül húsz perc alatt alszanak ki teljesen, így használójuknak bőven marad ideje arra, hogy ellazulva hajtsa álomra a fejét.

A világító párnát, amelyet hamarosan az ágy szélére szerelhető világítóeszköz követ, jövőre tervezik piacra dobni.

Medgyesi Zoltán

(mz@rettesoft.hu) a BMGE 24 éves informatika szakos hallgatója. Szabadidejét legszívesebben a barátjával tölti. Szeret autózni és bográcsban főzni. A Linuxot hat éve ismeri, de még nem volt lelkijereje, hogy átterjen rá. A Linuxvilág magazin hírszerkesztője.

Cégvilág

**Ingyenesen letölthető
a Sun ONE Application Server 7**

A Sun Microsystems októbertől ingyen kínálja Sun ONE (Open Net Environment) Application Server 7 termékét. A vállalat a magas költség jelentette akadály elhárításával lehetőséget teremt arra, hogy a vásárlók a működés-kritikus rendszerek által igényelt rendkívüli teljesítménybeli és méretezhetőségi követelményeknek leginkább megfelelő megoldást válasszák.

A Sun ONE Application Server 7 a Java 2 Platform, Enterprise Edition (J2EE) szabványra épült, illetve tökéletesen megfelel ezeknek az előírásoknak. A Java-alkalmazásokat elég egyszer megírni, utána bármilyen környezetben futtathatók, így az alkalmazások gyorsan és mérsékelt költségekkel bocsáthatók a felhasználók rendelkezésére.

Az alkalmazáskiszolgáló három, a vállalati környezetekhez és igényekhez igazodó változatban érhető el: Platform Edition, Standard Edition és Enterprise Edition néven. A kiszolgálót számos olyan egyéb termék egészíti ki, amelyek az alkalmazások fejlesztését, kipróbálását és üzembe helyezését segítik, gyorsítják.

➔ http://www.sun.com/software/products/appsvr/appsvr_download.html

HP-támogatás és-pályázat iskoláknak

A HP Magyarország a számítástechnikai piac meghatározó szereplőjeként jelentős szerepet tölt be a hazai társadalmi és gazdasági életben, ugyanakkor kiemelten fontos területként kezeli az oktatást is. A cég szakközépiskolák számára meghirdetett pályázata már-már hagyománynak tekinthető: idén hetedik alkalommal írt ki pályázatot olyan alkalmazásokra, amelyeket Magyarországon még nem valósítottak meg, vagy amelyek a pályázat segítségével továbbfejleszhetők. A HP szakértőiből álló bíráló bizottság előnyben részesítette azokat a pályamunkákat, amelyek a mindennapi gyakorlatban is használható javaslatokkal szolgáltak, vagy egy működőképes üzleti alkalmazás megvalósíthatóságát vázolták fel.

Nagy érdeklődés követte a pályázatkiírást, csaknem 80 pályamű érkezett be. Az idei pályázat során is kiderült: a diákok nem ismertek lehetetlent, ötleteik nemzetközi mércével mérve is versenyképesek voltak. Az eredmények elbírálása után több nyertes iskola is kapott az értékes nyereményekből: a HP Magyarországtól 65 nagy értékű kiszolgálót, a Microsoft Magyarországtól pedig Windows 2000 Server operációs rendszereket.

➔ <http://www.hp.hu>

Brit nemzeti hálózat a mellrák szűrésére

Az Oxford Egyetem is csatlakozott az IBM és az Egyesült Királyság kormánya által megkezdett programhoz, amelynek célja olyan kifinomult számítástechnikai hálózat létrehozása, amelynek segítségével a mellrák korán felismerhető és diagnosztizálható, és amellyel az orvosok többet tudhatnak meg a betegség kezeléséről. A fejlett világban élete során minden nyolcadik nőben kialakul a

mellrák, és ez az arány egyre nő. A kórházak és az orvosok jobb diagnosztikai módszereket szeretnének kézhez kapni a mellrák felismeréséhez, mivel minél korábban ismerik fel a betegséget, annál jobb lehet az előrejelzés és a túlélés valószínűsége.

Az IBM és az Egyesült Királyság kormánya által fizetett közel hatmillió dolláros tervezet – melynek az Oxford Egyetem kutatói az „eDiamond” nevet adták – része az Egyesült Királyság kormánya által meghirdetett eScience kezdeményezésnek. Az eDiamond lesz az első olyan hálózat, amely teljes egészében kereskedelmi forgalomban kapható elemekből – többek között a Mirada Solutions által készített, kategóriájában vezető alkalmazásból –, valamint új és meglévő digitális mammográfékből áll. A hálózat segítségével hatalmas elektronikus fotóalbum jön létre mammográfok kezelésére, visszakeresésére, így könnyebb lesz a tárolásuk és visszakeresésük, valamint az orvosok is fejlettebb elemzőeszközöket használhatnak, amelyekkel a betegség pontosabban ismerhető fel. Lehetőség kínálkozik világméretű hálózat létrehozására is, amelyben Franciaország, Németország, Japán és további országok hasonló tervezetei fonódnának össze.

Bővült a Cisco hálózati tananyaga

A Cisco Systems által 1997-ben indított Cisco Hálózati Akadémia Program 2002-ben jelentős bővüléssel ment keresztül. A 149 országban futó program keretében mintegy tízezer akadémián több mint 300 000 diák számára elérhető Cisco hálózati tananyagok mellett Magyarországon most újabb webalapú oktatási anyagok lesznek hozzáférhetőek négy nemzetközi vállalat jóvoltából. A hazai CNA-közösség 1500 diákot számlál, és összesen 52 akadémián folyik az oktatás.

A HP Magyarország, a SUN Microsystems, az Adobe és a Panduit cégekkel kötött megállapodások értelmében jelenleg négy új, webalapú, a legújabb megoldásokat tartalmazó tananyag áll térítésmentesen az akadémiákon tanuló diákok rendelkezésére. A HP jóvoltából elérhető lesz az, IT alapfogalmi tananyag, amely általános informatikai ismereteket nyújt a közép- és felsőoktatási intézményekben tanuló diákoknak.

A SUN támogatásának köszönhetően öt akadémia oktatói végezték el a „train the trainer” tanfolyamokat, melyek során elfogadott Java és Unix rendszergazdai oktató képzést szereztek, lehetővé téve, hogy iskolákban megkezdődjön a rendszeres Java- és Unix-képzés. Ennek az öt oktatási intézménynek a számára a SUN öt munkaállomásból álló, a képzéshez szükséges technikai hátteret biztosító labort is felajánlott.

➔ <http://www.cisco.hu>

Medgyesi Zoltán (mz@rettesoft.hu)

a BMGE 24 éves informatika szakos hallgatója. Szabadidejét legszívesebben a barátnőjével tölti. Szeret autózni és bográcsban főzni. A Linuxot hat éve ismeri, de még nem volt lelkiereje, hogy áttérjen rá. A Linuxvilág magazin hírszerkesztője.

Pedagóguspályázat linuxos szemmel

A Széchenyi terv keretében több mint 13 000 pedagógus jutott számítógéphez. Az egészében sikeresnek tekinthető pályázatást eredetileg a Miniszterelnöki Hivatal Informatikai Kormánybiztossága végezte. Időközben azonban megváltozott az intézmény neve és szerkezeti felépítése.

Szaló István: A vállalkozásról és az engedélyekről kérdeztem Suhajda Attilát, az információs társadalom programok helyettes államtitkárát. Hogyan szólíthatjuk Önöket a továbbiakban, és mi változott az intézménynél?

Suhajda Attila: Mint az ismeretes, az új kormány önálló minisztériumot hozott létre az informatikával és a hírközléssel kapcsolatos feladatok megoldására. Így a májusban megalakult Informatikai és Hírközlés Minisztérium belül az Információs Társadalom Programok Helyettes Államtitkárságához kerültek a korábbi IKB-s pályázatok, és a jövőben ez a szervezet fogja kiírni az információs társadalom programjaihoz kapcsolódó új pályázatokat is. A most futó és a jövőben kiírásra kerülő pályázatok felelősei a helyettes államtitkárság szakértői, de a pályázatokkal kapcsolatos kérdések megválaszolására egy telefonos ügyfélszolgálati központot hoztunk létre, amelynek száma 06-40-200-454. Itt felkészült kezelők várják a felmerülő kérdéseket.

Sz. I.: A változások mennyiben érintik az SZT-IS-3 pályázatot? Például sok pályázó azt kérdezi, hogy mikor tud majd rákapcsolódni az Internetre. Milyen terveik vannak ezzel kapcsolatban?

S. A.: Ilyen nagyméretű csoport internet-hozzáféréssel történő kiszolgálására Magyarországon még nem volt példa. Így többek között meg kellett oldanunk azt a logisztikai és távközlési gondot is, hogy miképp lehet egységes szolgáltatást nyújtani az ország egész területén elszórtan élő, és távközlési szempontból egymástól jelentősen eltérő lehetőségekkel rendelkezők számára. Ez nem egyszerű feladat, a mindenki számára legjobb megoldást kellett megtalálnunk. A pályázók egy része (közel 12000 ember) már megkapta az internet-hozzáférés lehetőségét, a többiek pedig 2002. február végéig jutnak ehhez hozzá.

Sz. I.: Ez év januárjában a pályázókat még arról értesítették, hogy a katalógus alapján beszerzett gépekre mind a Microsoft, mind a Linux operációs rendszerek telepítésre kerülnek, s a felhasználónak a tervek szerint (az előírt határidőkön belüli rendelés, szállítás esetén) a gép átvételétől számítva legalább egy hónap áll majd a rendelkezésére, hogy eldöntse, melyik operációs rendszert kéri tartós használatra. A választása szerinti operációs rendszer megvásárlásának költségeit az IKB külön pályázat keretében átvállalja. A nem ingyenes operációs rendszert, illetve irodai csomagot választóknak külön pályázaton (SZT-IS-3/B) kellett volna indulni. Ma már tudjuk, hogy a pedagógusoknak soha nem kellett pályáznuk a Microsoft operációs rendszeréért, azt végül ingyen kapták meg. Mi történt időközben?

S. A.: Az előző kormány miniszteri döntése alapján a SuSE Magyarországgal és a Microsofttal kötött szerződés alapján a pályázaton nyertes pedagógusok ingyen kapják meg a SuSE Linux és a Windows operációs rendszereket, ezért az operációs rendszerre nem kellett külön pályázatot kiírni. Az Ön által ismert pályázati módszer megváltoztatásának oka nagyon egyszerű volt: az egytömögű beszerzés jelentősen csökkentette az árakat, és ezzel párhuzamosan elmaradtak a programok szétosztásánál óhatatlanul felmerülő egyéb költségek is.

Sz. I.: A minisztériumon belül mennyire ismertek a GNU/Linux rendszerek, illetve az alattuk futó irodai programok? Milyen tapasztalatai vannak a minisztériumnak a nyílt programokkal? Milyen terveik vannak az IKB utódjának a Linux operációs rendszerrel?

S. A.: Az IKB szakemberei először a Linux-felhasználók Magyarországi Egyesülete (LME) által szervezett előadásokon figyeltek fel a számukra is új megközelítést jelentő rendszerre, ennek alapján módot adtak az LME szakembereinek arra, hogy ismertessék elképzeléseiket. A tárgyalások végül is oda vezettek, hogy az LME tízmillió forintot kapott bizonyos Linux-elemek magyarítására. Ami a Linux használatát illeti: az IKB által korábban használt gépekről nincsenek teljes körű adataink, de az IHM-ben jelenleg csak Windows alapú rendszerek működnek, azonban tervezzük, hogy valamikor a Linux is megjelenik gépeinken. Az IHM-nek határozott szándéka a Linux elterjesztésének támogatása, de a tényleges együttműködési lehetőségeken még dolgozunk.

Sz. I.: A kiírt pályázat tervezésekor kikérték-e linuxos szakember tanácsait?

S. A.: Erre természetesen sor került, már csak azért is, mert az LME által tartott előadássorozat költségeihez az IKB is hozzájárult, így az IKB-ban voltak Linuxhoz értők is. Ebből következően a kapcsolattartó emberek számára a Linux lehetőségeiről az összes háttértájékoztató rendelkezésre állt.

Sz. I.: Szentiványi Gábor, a SuSE AG magyarországi irodájának igazgatója szerint a pályázat Linuxszal kapcsolatos eseményei meglehetősen meggondolatlanul zajlottak. Az IKB természetesnek vette, hogy a Linux ingyenes termék. De annak ellenére, hogy a Linuxnak nincs ára, az előállításnak és a terjesztésnek vannak költségei. Sz. G szerint az IKB ezeket a költségeket nem tervezte be, és az teljes egészében a szállítókra hárult, akik a végén ezt továbbhárították a Linux-terjesztőkre. Önök hogyan látják ezt?

S. A.: A Linux-társadalom több esetben kapott támogatást az IKB-tól. Egyrészt az LME összességében további ötmillió forinthez jutott a linuxos programok további magyarításával és népszerűsítésével összefüggő kiadások fedezésére. Másrészt a magyar fejlesztésű UHU-Linuxot forgalmazó cég negyvenmillió forintos pályázati pénzt nyert el, amelyből egyrészt elkészítette az IKB szakembereivel közösen meghatározott tartalmú eredeti változatot, másrészt ez az összeg a terjesztés költségeire is fedezetet



jelentett. Így kimondható, hogy soha nem állt szándékunkban áthárítani ezeket a költségeket a terjesztőkre.

Sz. I.: *A pedagógusok hiányolták a Linuxszal kapcsolatos magyar nyelvű leírást, hiszen sokan először találkoztak ezzel az operációs rendszerrel. A Microsoft adott egy nem túl terjedelmes könyvecskét a CD és a nagyon díszes kiállítású felhasználási szerződés mellé. Az IKB arra számított, hogy a Linux-terjesztők ingyenes leírást is biztosítanak a telepítőanyag mellé, vagy nem is terveztek segítséget nyújtani ezen a téren?*

S. A.: A Linux-terjesztőknek természetesen nem feladatuk, hogy ingyenes leírást nyújtsanak a pályázaton nyertes pedagógusoknak, ezt soha nem gondolta így senki. Az LME, illetve az UHU-Linux által megkapott támogatás a felmerülő előállítási, dokumentációs költségek fedezésére is szolgált. E munka eredményeképpen ma már minden fontos Linux-elemhez van magyar nyelvű leírás. Természetesen nem lehet azt mondani, hogy ez teljes, de úgy véljük, hogy a programok világában megszokott módon a gyártóktól független szerzők művei pótolják ezeket a hiányosságokat. Természetesen feladat ettől függetlenül még akad jócskán. Így például a pályázat nyerteseinek kötött határidőn belül előre meghatározott ECDL-vizsgaszinteknek kell megfelelniük. Jelenleg azonban még csak nagyon kisszámú ECDL vizsga tehető le Linux-környezetben, ez pedig jelentősen hátráltathatja a Linux terjedését. Meglátásunk szerint a Linux-közösségnek nagy szerepe lehet a Linuxra támaszkodó ECDL-tananyagok készítésénél és kipróbálásánál.

Sz. I.: *A Microsoft-csomagban valóban majdnem minden jelen van, ami az általános Windows-oktatáshoz szükséges lehet. Ezzel szemben egyetlen egy SuSE Linux-lemezre igencsak kevés fér rá, hiszen a Linux sokkal több programot tartalmaz, mint a windowsos változat. Például többféle ablakkezelő közül választhatunk, számtalan programozási eszközt tartalmaz. Az eredeti SuSE Linux Professional terjesztés hét lemezt és háromkötetnyi leírást foglal magába. Vannak olyan tervek, hogy ezt a csomagot is eljuttatják a pedagógus felhasználókhoz?*

S. A.: Erre nincs szükség. Ugyanis a 7 CD-t tartalmazó csomag már a kiszolgálóbeállításához szükséges adatokat is magában foglalja, és ezeket a programokat csak az üzleti felhasználók, illetve a hivatásos fejlesztők tudják hasznosítani. A pályázati úton számítógépekhez jutott pedagógusok a gépet otthoni használatra kapták, ezért kapott az UHU-Linux támogatást arra, hogy egy kifejezetten a rendszer felhasználóinak igényeit kielégítő csomagot készítsen el. Másrészt pedig a kiszállított gépeken mindkét operációs rendszer (a Windows, illetve a végfelhasználók igényeinek megfelelően összeállított Linux-terjesztés) megtalálható, és a felhasználó a rendszerindításkor dönthet, hogy melyik operációs rendszerrel kíván dolgozni. Fontos hangsúlyozni, hogy ez nem egyszeri és megváltoztathatatlan döntés, mert az elindított rendszer nem törli le a másik programot, így mindkettő bármikor használható. Így egy korábbi válasza vissza-utalva, látni lehet, hogy a mostani helyzet nagyobb

választási lehetőséget nyújt a rendszerek felhasználóinak a korábbi tervhez képest, amely alapján csak egyszer lett volna alkalmuk a döntésre.

Sz. I.: *A fenti nehézségek ellenére a Linux felkerült a pedagógus PC-kre, és ezzel legalább a lehetősége megteremtődött annak, hogy a Linux bekerüljön a középiskolákba. Milyen további lépéseket tervez ez ügyben az IKB?*

S. A.: A Linux terjedése az Ön által említett folyamathoz képest éppen fordított irányú. Az IKB azért döntött a Linux terjesztésének támogatása mellett, mert a felmérések és statisztikák azt mutatták, hogy a középiskolai rendszerek egy részén (még akkor is, ha ez a rész ma nem túl jelentős) már megjelent a Linux. Így lehetőséget kívántunk nyújtani a tanároknak ahhoz, hogy otthoni gépükön is az iskolai rendszereken megszokott eszközöket használják.

Sz. I.: *Vannak, akik szerint a következő négy év a Microsoft egyeduralmát fogja jelenteni az IKB utódánál, és minden más operációs rendszer háttérbe fog szorulni a Linuxszal együtt. Valóban vannak ilyen törekvések az intézménynél?*

S. A.: Nincsenek ilyen törekvései az IHM-nek. A minisztériumnak az a feladata, hogy az információs társadalom kialakulását előmozdítsa. Ezzel az elvvel teljesen ellentétes lenne egy szabadon használható programcsomag háttérbe szorítása, különösen akkor, ha ez egy monopólium megtörését segíti. Van azonban még egy fontos szempont, a képzés kérdése. A pályázatok a pályázók elé informatikai képzettségük növelése tekintetében nagyon komoly követelményeket állítottak. Azonban egészen a közelmúltig az ECDL-tananyagok nem tartalmaztak Linux-oktató modulokat, így természetesen Linux-vizsgát sem lehetett tenni. Ez pedig azt jelentette, hogy a pályázók nem tudtak teljes mértékben a Linuxra hagyatkozni, mivel nem tudtak eleget tenni kötelezettségeiknek. Ez a helyzet már valamennyire megváltozott, de az látszik, hogy a Linux-oktatással még vannak nehézségek. Úgy véljük, hogy a Linux-társadalom jól felfogott érdekének kell lennie, hogy a képzés minél nagyobb tömegeket érjen el, és folyamatosan készüljenek tananyagok Linuxra. Természetesen a képzésnek piaci alapon kell lezajlania, de látszik, hogy tananyagfejlesztésre mindenképpen szükség van.

Sz. I.: *Tervezi-e Ön személy szerint, hogy a közeljövőben Linuxot telepít saját otthoni számítógépére?*

S. A.: Ott nem, de az IHM-ben a kiszolgálók szintjén mindenképpen, sok más ország kormányzatához hasonlóan.

Sz. I.: *Köszönöm a válaszokat.*



Szaló István

(ratiosoft@freemail.hu) tanár, immár több mint másfél évtizede foglalkozik programozással, de csak a Java és a Linux megismerése után tudta meg, hogy mi is az igazi programozás. Több írása megjelent már a hazai számítástechnikai lapokban.

IV. GNU/Linux Szakmai Konferencia

A Linux-felhasználók Magyarországi Egyesülete (LME) az Infoszférával közös szervezésben 2002. november 9-én Budapesten, a Best Western Grand Hotel Hungáriában rendezte meg szokásos éves szakmai tanácskozását.

Az LME négy évvel ezelőtt, 1998 őszén alakult meg, amit közel féléves jogi eljárás követett, emiatt a végleges

bejegyzés csak 1999 tavaszán történt meg. Az egyesület azóta minden évben megrendezi szakmai konferenciáját. Az eddigi évekre visszatekintve nyugodtan állíthatom, hogy az idei messze a legszínvonalasabb, igazán profi módon megrendezett esemény volt.

A 10 órás kezdésre minden kiállító elfoglalta a számára kijelölt kiállítói



helyet, a látogatók beléptetése is végetért, ennek köszönhetően Sári Gábor, az LME elnöke időben tudta köszönteni a jelenlévőket és egyben a tanácskozást megnyitni. A lényegében három különböző helyszínen tartott előadásokat könnyű volt megtalálni, hiszen az Infoszféra munkatársai szinte mindenütt ott voltak, segítőkészen, tájékozottan irányították a látogatókat.

Az előző konferenciákkal ellentétben most minden előadás az előre meghirdetett időpontban kezdődött, csúszás nem történt, ezért a 18 órás zárás is tartható lett volna, ha az események végére kialakult „házbulis” hangulatot a szervezők meg kívánták volna szakítani, de szerencsére nem tették.

Délelőtt a 363 bejegyzett látogató (a jelek szerint sokan nem tudtak eljönni) kapott egy ajándékcsomagot, ami többek között egy „szavazólapot” tartalmazott, melyen az általa legjobbnak tartott előadóra adhatott le szavazatot, valamint jelentkezhetett a zárás előtt lebonyolított tombolára.

Idén Kadlecsik József nyerte a legjobb előadó díját, ami a Kiskapu Kft. tízezer forintos könyvutalványja volt. Józsefet felkérték, hogy a tombolasorsoláson „húzó” emberként vegyen részt, amit természetesen örömmel elvállalt. A következő nyereménytárgyak kerültek kisorsolásra (zárójelben a díjat felajánló nevével):

- egy Soldier of Fortune for Linux eredeti játékprogram (Software Books);
- az „Informatikai rendszerek biztonsága” című szakkönyv (Infoszféra Kft.);
- egy FreeBSD-, egy OpenBSD- és egy NetBSD-telepítő CD-készlet (Magyar BSD Egyesület);
- öt GNU/Linux-póló (LME);
- fődíjként egy 1,5 m magas plüsspingvin (Pilátus-Comp Kft.).

Fontos megjegyezni, hogy a megjelent kiállítók jó hangulatúnak és kimondottan eredményesnek ítélték a rendezvényt. Ezt az is bizonyítja, hogy már másnap többen köszönőlevelet küldtek az egyesületnek, amelyben kérték, hogy a jövőre megrendezésre kerülő konferencia szervezésekor ismét számítsanak a részvételükre.

Az eddigi konferenciák hagyományait követve természetesen jelen volt az LME CD-író projektje, ahol a nap folyamán számos Linux-terjesztés gazdára talált.

A konferencián bemutatásra került előadások mennyisége, szakmai színvonalai is figyelemre méltó. Álljon itt egy kis ízelítő!

Csomagszűrő tűzfalak – a Netfilter

A Linux-mag korai változataitól kezdve létezett csomagszűrő szolgáltatás, ami több lépésben fejlődött, változott. A lépcsőket általában a felhasználói térben futó beállító-program nevéhez szokták kötni: ipfw, ipfwadmin, ipchains és a legfiatalabb a sorban az iptables. Az IP Tables rendszermagbéli „párja” a Netfilter.

A Netfilter a 2.3-as fejlesztői maggal jelent meg, a 2.4-es üzembiztos rendszermagsorozat ezt már tartalmazza. Léteznek visszafelé együttműködő modulok is, amelyekkel az IP Chains, sőt az ipfwadm formálisan tovább használható a Netfilter beállítására, de ez nem ajánlott, hiszen a megoldás erősen hasonlítana arra, mintha a jó öreg Trabi műsorfalát átszerelnék vadonatúj Mercedesünkbe, csak azért, mert megszoktuk...

Előadó: Kadlecsik József

Az embertől az államig: a nyílt forráskód kormányzati használata

A legfontosabb tényező a kormányzati rendszereknél nem(csak) a rendszer ára, hanem annak biztonsága. Szabad programok esetén lehetőség nyílik a programok ellenőrzésére, biztonsági felülvizsgálatára (auditálására). Ez egyre fontosabb, hiszen napvilágra került már olyan tény is, miszerint a legelterjedtebb operációs rendszerben olyan hátsó kapu volt (csak volt?) található, amelyhez egy kulccsal a Microsoft, egy kulccsal az amerikai nemzetbiztonsági hivatal, egy kulccsal pedig az ismeretlenek rendelkeztek.

Miért fontos a kód felülvizsgálhatósága? Egy képzett számítógépes betörő (cracker) nem kell, hogy rendelkezzen a rendszer forráskódjával, anélkül is be tud törni a rendszerbe. Elég csak azokra az időkre gondolni, amikor még nem terjedtek el a szabad programok és rengeteg betörés volt. A szakemberek azonban kijavíthatják a hibákat, ha rendelkeznek a forrással. A Világhálón pedig számos fejlesztő és cég tanulmányozhatja a forráskódot, míg a zárt forrású termékeknl erre esély sincsen.

A tapasztalatok alapján látszik, hogy a szabad programok jelentik az egyetlen lehetőséget a kormányzati munkában. Nem véletlenül választotta az Amerikai Nemzetbiztonsági Szolgálat (NSA) fejlesztési felületül az olyan szabad rendszereket, mint a Linux vagy a FreeBSD.

Előadó: Deim Ágoston



Magyar Ispell – Válasz a Helyes-e?-re

A Magyar Ispell nyílt forráskódú projekt eredményével a Linux hazai – asztali felületeken történő – elterjedésének egyik legnagyobb gátját, a magyar helyesírás-ellenőrző hiányát sikerült orvosolni. A „műtét” kimenete olyan szerencsés volt, hogy ma a Magyar Ispell jobbnak tekinthető, mint a Microsoft Office-szal piacvezetővé vált „Helyes-e?” programja (MorphoLogic).
Hogyan készül egy profi nyelvi program? Mi a siker titka? – erre kaphatott választ, aki meghallgatta az előadást.
Előadó: *Németh László*

Ördög és pokol – Gondolatok a BSD-kről egy Linux-konferencia ürügyén

Hogy mi az a BSD, és főleg mik azok a BSD-k, napokig lehetne tárgyalni. Megközelíthető jogi, nyelvészeti, valószínűleg néprajzi alapon is, mi maradunk a száraz szakmaiság útján.
A BSD-k gyakorlati szempontból olyan Unixnak tekinthető operációs rendszerek, amelyek többsége a Linuxhoz hasonló lehetőségekkel bír és a fejlesztésük is ahhoz hasonlóan alakult. Hasonlítanak a Linuxhoz például abban is, hogy többféle BSD létezik, ahogy Linuxból is több terjesztés érhető el. Hasonlít abban, hogy jó-rossz programozók tucatjai, százai és ezrei fejlesztik, akiknek döntő többsége nem ezeknek a programoknak a fejlesztéséből tartja fenn magát és a családját.
Milyen BSD-k vannak és mire jók? Mit kapunk egy BSD-rendszerrel? A fenti kérdésekre Zahemszky Gábor, a BSD egyesület elnöke adta meg a választ.
Előadó: *Zahemszky Gábor*

A PHP-GTK

A PHP egy ingyenes, könnyen tanulható parancsnyelv, a GTK pedig egy több nyelvhez illeszkedő, grafikus felület készítésére használható eszközkészlet.
Ennek a két nagyszerű eszköznek az összekapcsolására született meg a PHP-GTK felületfüggetlen kiterjesztés. Néhány fejlesztőnek eszébe jutott, hogy ötvözni kellene a fenti két, látszólag teljesen összeegyeztethetetlen programot. 2000 őszén nekiláttak, és 2001 márciusában elkészült az első változat. Az első megbízható kiadásra pedig 2002 januárjáig kellett várni.
A PHP-GTK olyan PHP-bővítvény, mely a GTK-elemeket mint objektumokat teszi elérhetővé. 95 osztály és több mint 970 függvényt ad a PHP sajtójához, így az egyik legnagyobb bővítvénye. Az előadáson a PHP-GTK példa-programokon keresztül került bemutatásra.
Előadó: *Körmendi Domokos*

Az Oracle-Linux kapcsolat

A Linux már elérte a vállalati üzleti felhasználhatóság színvonalát. Kevesebb cég, de nagyobb szakértelemmel, vállalati szinten is elvárható támogatással áll a piacon elterjedt Linux-változatok mögött. Az Oracle az elsők között ismerte fel, hogy a Linuxszal érdemes a nagyvállalati piacon és a saját termékkörében is foglalkozni. Az első szélesebb körben elérhető Oracle on Linux

(8.0.5) már közel négy éve megjelent. Megjelentek és teljes körű felhasználhatóságot kíván továbbá az Oracle Application Server széles körű elemhalmaza és maga az Oracle E-Business Suite is, ami a vállalat irányítási rendszerek teljes kínálatát biztosítja. A Real Application Cluster megjelenésével az Oracle szakított a kizárólagos elgondolással, hogy a komoly nagyvállalati rendszerek alá csak nagy, monolit típusú eszközöket szabad értékesíteni. A RAC program szerkezete lehetővé teszi, hogy az egyszerűbb, olcsóbb számítógépekből (jellemzően akár PC-kiszolgálókból) is nagy összeteljesítményű, hibatűrő kiszolgálógépet építeni telepek (cluster) használatára, és ezt jól ki is lehessen használni. A Linux pedig az egyik választható operációs rendszere ezeknek a PC alapú kiszolgáló-környezeteknek. E forradalmian új telepítési módszereinek a bemutatása volt az előadás leghangsúlyosabb része.
Előadó: *Markovics Péter*

Webkiszolgáló-védelem határvédelmi eszközökkel

Az előadás egy valós esetről nyert tapasztalatokat igyekezett feltárni, illetve az alkalmazott megoldásokat ismertetni. A feladat szerint egy portáljellegű oldalt kellett védelmi rendszerrel ellátni. A portál fejlesztése a végéhez közeledett, de tervezése során a biztonsági szempontokat nem vették figyelembe. Mivel a honlapok feltörése és tartalmuk megváltoztatása látványos, a betörők körében egy webkiszolgáló általában kedvelt célpont. A kockázatot jelen esetben az emelte, hogy az oldal várt látogatottsága magas, továbbá rendszeres sorsolásokon értékes ajándékok nyerhetők.
Előadó: *Scheidler Balázs*



Gibizer Tibor
(gibzo@linuxmania.hu)
újságíró, immár hét éve a Linux elkötelezett híve. Imádjá a kutyákat, a kerékpározást és az autós csavargást.



Körmendi András



Markovics Péter

Kiszolgáló és tűzfal nem csak baglyoknak

Alig pár hónapja jelent meg az UHU-Linux ügyfél-változata, és nemsokára kezünkbe vehetjük a kiszolgáló- és tűzfalváltásokat.

Az UHU-Linux fejlesztői csapata nem télenkedik, tagjai folyamatosan készítik az új RC2-es ügyfélváltatot, ami e cikk megjelenésekor feltehetően már elérhetővé válik a nagyközönség számára. A programozásban kicsit jártasak el tudják képzelni, milyen nagyszabású feladat egy önálló operációs rendszer elkészítése. A legtöbben azt hittük, hogy ezzel ki is merülnek az UHU erőforrásai, de tévedtünk. Több fejlesztő szinte egyszerre vetette fel, hogy egy hazai fejlesztésű kiszolgáló- és tűzfalváltatra is szükség lenne. Az UHU vezérkara nem télenkedett, ezért az ötlet megszületését azonnal tett követte. Beindult a mókuserék!

Jelenleg az UHU fogja össze a fejlesztéseket, azonban a munka jellegéből fakadóan több cég önállóan végzi a közösen megbeszélte és vállalt feladatokat. Lényegében mindenki abban segít a legtöbbet, amiben a legjobb, tehát igazi csapatmunka alakult ki. Tekintettel vannak egymásra, segítik egymás munkáját, hiszen közös a cél: legyen végre hazánknak is saját Linuxa, ami minden feladat ellátására alkalmas.

Hogyan lehet immár nem egy, hanem egyszerre három feladatra összpontosítani? A válasz egy régen elfelejtett fogalomban rejlik, ez az összefogás, ami sajnos egyre ritkábban lelhető fel a hazai fejlesztők között.

Miért van szükségünk kiszolgálóra és tűzfalra az ügyfélváltot mellett, amikor ezeket a feladatokat más Linux-terjesztésekkel is meg lehet valósítani? Azok a szakemberek, akik ezzel foglalkoznak, a felmerülő gondokat biztosan meg is tudják oldani, csak hogy gondolni kell a Linuxhoz kevésbé értékre is, akik csak azért nem térnek át, mert a tanulási görbe íve kezdetben közel sincs arányban az elért eredményével. Bár tudják, mi a cél, a megvalósításra számítható idő kevés. Nekik olyan megoldást kell nyújtani, ami magyar, jó leírással bír, és a kezelőfelülete minden elérhető beállítást tartalmaz. A különböző Linux-összeállítások előnye egyben hátrány is. A DVD méretűvé vált terjesztések túlzott programválasztéka riasztóan hat a felhasználókra. Mit használjanak, mire és miért pont azt? Vegyük példának a Debian GNU/Linuxot, amit az ilyen feladatok ellátásakor a leggyakrabban használnak. Több mint 8000 csomagból kell kiválasztani a számunkra megfelelőket, be kell őket állítani, ami az esetek többségében terminálon keresztül, parancsok bevitelével történik. Nem szabad elfelejteni azt sem, hogy az új összeállítások a Debian legendás megbízhatóságának eléréséhez sokszor a kívántnál is lassabban jelennek meg, bár a hibajavításokra az esetek többségében nem lehet panasz. Az angolul kevésbé értők számára szinte használhatatlan, ha mindennemű linuxos előképzettség nélkül kell egy kiszolgálót felépíteni vagy egy tűzfalat felhúzni. Így érthető, ha a felhasználói tábor növekedése ezeken a területeken messze elmarad az asztali felhasználásokhoz képest, annak elle-

nére is, hogyha valamire, hát a hálózati munkára a költséghatékonyság figyelmen kívül hagyása esetén is az egyik legjobb megoldást a Linux kínálja. Be kell látnunk, hogy a legtöbb cég nem bő programválasztékra vágyik. Nekik nem kell több, mint egy olyan kiszolgáló, ami a mindennapos irodai vagy egyéb számítástechnikai feladatokat összefogja és irányítja. Ennek felismerése, a mindennapok tapasztalatai és a visszajelzések vezették a készítőket arra, hogy egységes, könnyen karbantartható célterjesztést hozzanak létre.

A cégek jogos igénye, hogy a felhasználókat központilag lehessen felügyelni. Amennyiben egy vállalati rendszergazdának sok kiszolgálót és ügyfélgépet kell felügyelnie, nem sok ideje marad a különálló felhasználói adatbázisok kiszolgálóprogramok szerinti karbantartására. A többletteleherrel együtt a hibázás esélye is megnő, így kézenfekvő, hogy csak egy könnyen és jól felügyelhető rendszer elkészítését érdemes megvalósítani. A jelenlegi Linux-összeállításokon végigtekintve sajnos nemigen találunk olyat, ami könnyen telepíthető, a telepítés után rögtön rendelkezik a központi felhasználó-adatbázis elérhetőségével és az ehhez kapcsolódó vezérlőfelülettel. Olyat meg végképp nem lelünk, aminek a fejlesztése hazánkban történik, tehát a leghozzáértőbb segítség is helyben van.

Az UHU kezelőfelülete természetesen nemcsak a felhasználókezelést tartalmazza, hanem rajta keresztül a kiszolgálón futó alkalmazásokat is beállíthatjuk. Lényeges pont, hogy a felhasználók felügyeletével ne csak a felhasználók hozzáadását, csoportokba szervezését, jelszavaik módosítását lehessen megoldani, hanem a felhasználókat a szolgáltatásokhoz is hozzá lehessen rendelni. Ez utóbbit akár úgy is, hogy a felhasználó minden szolgáltatáshoz külön jelszót kapjon. Fontos azonban, hogy a felhasználó saját maga be tudja állítani jelszavát, személyi adatait vagy azt, hogy a levelei hova kerüljenek továbbításra (természetesen csak akkor, ha erre a rendszergazda engedélyt adott).

Nincs szükség nagy programválasztékra. Szükség van azonban megbízható működésre, a támogató cégtől pedig gyors és pontos terméktámogatásra. A fentiek végiggondolása után a célok egyértelműek:

- szabványos alapokon nyugvó központosított felhasználói adatbázis létrehozása,
- a központi adatbázist elérni képes alkalmazások használata,
- a kiszolgálóalkalmazások könnyű beállíthatósága,
- az elérhető legnagyobb biztonság megteremtése,
- a feladatokat maradéktalanul ellátó legkisebb csomag a könnyű karbantarthatóság és a felhasználói támogatás érdekében,
- különböző erőforrást feltételező beállítófájlok különböző géptípusokhoz,
- jól működő csomagkezelés,
- grafikus telepítőalkalmazás,
- kezelhető és könnyen bővíthető vezérlőeszköz,
- sok és pontos segítség a program mellé.

Kiszolgáló és tűzfal nem csak baglyoknak

Alig pár hónapja jelent meg az UHU-Linux ügyfél-változata, és nemsokára kezünkbe vehetjük a kiszolgáló- és tűzfalváltásokat.

Az UHU-Linux fejlesztői csapata nem télenkedik, tagjai folyamatosan készítik az új RC2-es ügyfélváltatot, ami e cikk megjelenésekor feltehetően már elérhetővé válik a nagyközönség számára. A programozásban kicsit jártasak el tudják képzelni, milyen nagyszabású feladat egy önálló operációs rendszer elkészítése. A legtöbben azt hittük, hogy ezzel ki is merülnek az UHU erőforrásai, de tévedtünk. Több fejlesztő szinte egyszerre vetette fel, hogy egy hazai fejlesztésű kiszolgáló- és tűzfalváltatra is szükség lenne. Az UHU vezérkara nem télenkedett, ezért az ötlet megszületését azonnal tett követte. Beindult a mókuserék!

Jelenleg az UHU fogja össze a fejlesztéseket, azonban a munka jellegéből fakadóan több cég önállóan végzi a közösen megbeszélte és vállalt feladatokat. Lényegében mindenki abban segít a legtöbbet, amiben a legjobb, tehát igazi csapatmunka alakult ki. Tekintettel vannak egymásra, segítik egymás munkáját, hiszen közös a cél: legyen végre hazánknak is saját Linuxa, ami minden feladat ellátására alkalmas.

Hogyan lehet immár nem egy, hanem egyszerre három feladatra összpontosítani? A válasz egy régen elfelejtett fogalomban rejlik, ez az összefogás, ami sajnos egyre ritkábban lelhető fel a hazai fejlesztők között.

Miért van szükségünk kiszolgálóra és tűzfalra az ügyfélváltot mellett, amikor ezeket a feladatokat más Linux-terjesztésekkel is meg lehet valósítani? Azok a szakemberek, akik ezzel foglalkoznak, a felmerülő gondokat biztosan meg is tudják oldani, csak hogy gondolni kell a Linuxhoz kevésbé értékre is, akik csak azért nem térnek át, mert a tanulási görbe íve kezdetben közel sincs arányban az elért eredményével. Bár tudják, mi a cél, a megvalósításra számítható idő kevés. Nekik olyan megoldást kell nyújtani, ami magyar, jó leírással bír, és a kezelőfelülete minden elérhető beállítást tartalmaz. A különböző Linux-összeállítások előnye egyben hátrány is. A DVD méretűvé vált terjesztések túlzott programválasztéka riasztóan hat a felhasználókra. Mit használjanak, mire és miért pont azt? Vegyük példának a Debian GNU/Linuxot, amit az ilyen feladatok ellátásakor a leggyakrabban használnak. Több mint 8000 csomagból kell kiválasztani a számunkra megfelelőket, be kell őket állítani, ami az esetek többségében terminálon keresztül, parancsok bevitelével történik. Nem szabad elfelejteni azt sem, hogy az új összeállítások a Debian legendás megbízhatóságának eléréséhez sokszor a kívántnál is lassabban jelennek meg, bár a hibajavításokra az esetek többségében nem lehet panasz. Az angolul kevésbé értők számára szinte használhatatlan, ha mindennemű linuxos előképzettség nélkül kell egy kiszolgálót felépíteni vagy egy tűzfalat felhúzni. Így érthető, ha a felhasználói tábor növekedése ezeken a területeken messze elmarad az asztali felhasználásokhoz képest, annak elle-

nére is, hogyha valamire, hát a hálózati munkára a költséghatékonyság figyelmen kívül hagyása esetén is az egyik legjobb megoldást a Linux kínálja. Be kell látnunk, hogy a legtöbb cég nem bő programválasztékra vágyik. Nekik nem kell több, mint egy olyan kiszolgáló, ami a mindennapos irodai vagy egyéb számítástechnikai feladatokat összefogja és irányítja. Ennek felismerése, a mindennapok tapasztalatai és a visszajelzések vezették a készítőket arra, hogy egységes, könnyen karbantartható célterjesztést hozzanak létre.

A cégek jogos igénye, hogy a felhasználókat központilag lehessen felügyelni. Amennyiben egy vállalati rendszergazdának sok kiszolgálót és ügyfélgépet kell felügyelnie, nem sok ideje marad a különálló felhasználói adatbázisok kiszolgálóprogramok szerinti karbantartására. A többletteleherrel együtt a hibázás esélye is megnő, így kézenfekvő, hogy csak egy könnyen és jól felügyelhető rendszer elkészítését érdemes megvalósítani. A jelenlegi Linux-összeállításokon végigtekintve sajnos nemigen találunk olyat, ami könnyen telepíthető, a telepítés után rögtön rendelkezik a központi felhasználó-adatbázis elérhetőségével és az ehhez kapcsolódó vezérlőfelülettel. Olyat meg végképp nem lelünk, aminek a fejlesztése hazánkban történik, tehát a leghozzáértőbb segítség is helyben van.

Az UHU kezelőfelülete természetesen nemcsak a felhasználókezelést tartalmazza, hanem rajta keresztül a kiszolgálón futó alkalmazásokat is beállíthatjuk. Lényeges pont, hogy a felhasználók felügyeletével ne csak a felhasználók hozzáadását, csoportokba szervezését, jelszavaik módosítását lehessen megoldani, hanem a felhasználókat a szolgáltatásokhoz is hozzá lehessen rendelni. Ez utóbbit akár úgy is, hogy a felhasználó minden szolgáltatáshoz külön jelszót kapjon. Fontos azonban, hogy a felhasználó saját maga be tudja állítani jelszavát, személyi adatait vagy azt, hogy a levelei hova kerüljenek továbbításra (természetesen csak akkor, ha erre a rendszergazda engedélyt adott).

Nincs szükség nagy programválasztéka. Szükség van azonban megbízható működésre, a támogató cégtől pedig gyors és pontos terméktámogatásra. A fentiek végiggondolása után a célok egyértelműek:

- szabványos alapokon nyugvó központosított felhasználói adatbázis létrehozása,
- a központi adatbázist elérni képes alkalmazások használata,
- a kiszolgálóalkalmazások könnyű beállíthatósága,
- az elérhető legnagyobb biztonság megteremtése,
- a feladatokat maradéktalanul ellátó legkisebb csomag a könnyű karbantartás és a felhasználói támogatás érdekében,
- különböző erőforrást feltételező beállítófájlok különböző géptípusokhoz,
- jól működő csomagkezelés,
- grafikus telepítőalkalmazás,
- kezelhető és könnyen bővíthető vezérlőeszköz,
- sok és pontos segítség a program mellé.



A tervezés kezdeti szakaszában felmerült, hogy az alaptól új összeállítás készüljön, azonban az UHU ügyfél-változat készítőivel egyetértésben sokkal hasznosabbnak látták, ha a minden változatban előforduló alkalmazásokat közös alapokra helyezik. Tehát mindhárom UHU-változatnál ugyanazokkal az alapsomagokkal találkozhatunk. Ez a későbbiekben jelentősen megkönnyíti a változtatásokat és az egységes terméktámogatás lehetőségét. A programok kiválasztásánál elsőrangú szempont volt, hogy teljesítse a saját típusán belüli követelményeket; és természetesen teljességgel figyelembe vették a „Linux Standard Base”, valamint a „Filesystem Hierarchy Standard” ajánlásait. Ennek alapján a következő kiszolgálóalkalmazások kerültek a rendszerbe: felhasználó-adatbázis és címtárszolgáltatások: OpenLDAP; levelezés: Postfix; a levelek letöltése: Courier-IMAP; webkiszolgáló: Apache; adatbázis-kezelő: PostgreSQL és MySQL; fájl-kiszolgáló: Samba; nyomtatókiszolgáló: CUPS; proxy: Squid; FTP-kiszolgáló: Proftpd; a levelezőlisták kezelése: Sympa; távolifelügyelet-vezérlés: OpenSSH és egy átalakított Webmin; névkiszolgáló: BIND 9-es változat; DHCP-kiszolgáló: ISC DHCPD; faxkiszolgáló: Hylefax. Mint látható, a vezérlőeszköznél jól bevált alkalmazásra támaszkodtak. A Webmin előnyei közé tartozik, hogy nem igényel külön webkiszolgálót, képes az SSL-kapcsolatok kezelésére, és ami a legfontosabb, modulszerűen épül fel, tehát új modulok készítése sem okoz nehézséget. A programozási felület jól dokumentált, átlátható, maga az alkalmazás Perlben íródott.

Megkerestem **Deim Ágostont (Agót)**, az LSC (Linux Support Center) Kft. ügyvezető igazgatóját, aki a fenti projektek irányítását végzi, és néhány, a munkájával kapcsolatos kérdést tettem fel neki.

Gibizer Tibor: Hol tart a fejlesztés, mikorra várhatók az első kipróbálható változatok?

Deim Ágoston: Nagyon jól haladunk, a fejlesztésben résztvevők munkabírása lenyűgöző. Bevallom, az egész egy kicsit álomszerűnek tűnt. Volt már a hazánkban több lelkes próbálkozás csodálatos tervekkel, de rendszerint – mire az érdemi munkára került volna a sor – a „világ megváltása” elmaradt. Tegyük hozzá, nagyon jó volt az időzítés. Több cég szinte egyszerre töprengett el azon, hogy szükség lenne egy hazai kiszolgáló- és tűzfalváltatra. Az UHU jó érzékkel látta meg a lehetőséget, és fogta össze, majd terelte egy mederbe a fejlesztőket. Ennek a még páratlan összefogásnak az eredménye, hogy ilyen hatékonyan tudunk dolgozni. A első kipróbálható változatok feltehetően e cikk megjelenésekor már elérhetőek lesznek az UHU-Linux honlapjáról (☞ <http://www.uhulinux.hu>).

Gibzo: Két különálló termékről beszélünk, tehát lesz egy UHU Szerver és egy UHU Tűzfal. Azt már tudjuk, hogy azonos alapokat használnak, így az együttműködéssel nem lehet gond. Ez azt is jelentheti, hogy nemsokára megjelennek azok a cégek, amelyeknél az UHU Tűzfal után egy UHU Szerver fog dolgozni, majd ezekhez UHU ügyfélváltatok kapcsolódnak?

D. Á.: Valóban külön kezelendő a kettő. Ez többek között azért fontos, mert önállóan is meg kell állniuk a helyüket. Ahol csak egy belső hálózat üzemeltetése a cél, ott felesleges a tűzfal, illetve biztonsági okokból lényegesen jobb, ha a tűzfal külön van. Mi még nem nagyon mertük ezt így megfogalmazni, de valóban reménykedünk benne, hogy lesznek olyan cégek, ahol minden feladatot az UHU-val látnak majd el.

Gibzo: Minden terjesztésnek érzékeny pontja a telepítés. Ez az UHU ügyfélváltatánál nagyon ügyesen lett megoldva. Miként fog zajlani a további két változat telepítése?

D. Á.: Hangulatát, egyszerűségét tekintve eszünkbe sem jutott elrugaszkodni az UHU ügyfélváltatától. A lemezek önműködően indulnak, ha a gép BIOS-ában ezt engedélyeztük. Majd a telepítésnél meglehetősen sok kérdést kapunk, hogy milyen szolgáltatásokra van szükségünk – természetesen mindez magyarul zajlik. A telepítő gyakorlatilag elég moduláris – lényegében ugyanaz a program, csak különböző modulokat hív meg és különböző feltételeknek felel meg. Amennyiben kiválasztásra kerül, hogy levelezésre szükség van, rögtön be kell állítani, hogy milyen tartománynevek vannak stb. A tűzfal telepítésénél be lehet állítani az összetevőket, erre a ZUI-t használjuk, új nevén az UZI-t (User Zorp Interface). Annyit még, hogy mindegyikben 2.4-es rendszermag lesz.

Gibzo: Milyen felületen lehet később kezelni?

D. Á.: A tűzfalnál egy Python newt alapú, tehát inkább karakteres felületet használunk. Ez jól finomhangolható rendszer, ezért nem árt, ha a felhasználó rendelkezik némi szakmai tudással. Ez tehát inkább a szakemberek számára készül, így mellőzi a díszes felületet. A kiszolgáló beállítása webes felületen történik, egészen pontosan egy webmin felület lett átalakítva. Egyedi menüket készítettünk hozzá, illetve saját könyvtárakat is írtunk hozzá, például az ldap libet. A kiszolgálónál lényegesnek találtuk, hogy közös adatbázisa legyen a felhasználóknak, tehát minden szolgáltatás egy adatbázisból tudjon hitelesíteni, ezért OpenLDAP alapú lesz. A levelezőkiszolgáló ldap-ból szedi a felhasználókat. Ezen lesz az Apache is, tehát akár az Internetre is kitehető, akkor értelemszerűen csak azt a szolgáltatást kell elindítani.

A felhasználók kezelésére igazán büszkék vagyunk, mert ez olyan dolgokra is figyel, ami a hasonló feladatokra készült programokban nem található meg, de ennek részleteit még szeretném titokban tartani.

Gibzo: Köszönöm a beszélgetést.

Természetesen az ügyfélváltozathoz hasonlóan ezek is GPL-esek lesznek, azaz mindenki számára szabadon hozzáférhetőek.



Gibizer Tibor

(gibzo@linuxmania.hu)

újságíró, immár hét éve a Linux elkötelezett híve. Imádja a kutyákat, a kerékpározást és az autós csavargást.

Sokoldalú asztali lejátszó

Valamikor két-három évvel ezelőtt ragadta meg a figyelmet egy azóta megszűnt számítógépbolt árlistáján egy – talán Yamakasi névre hallgató – asztali CD-, DVD- és MP3-lejátszó. A hasonló készülékek kezdetben sem magas ára azóta egyre barátságosabbá válik, és bár a márkanev sűrűn változik, a lényeg ugyanaz maradt: elérhető áron juthatunk olyan egységhez, ami illeszkedik a meglévő hifitoronyba, és számítógép nélkül is alkalmas az MP3-zenéket tartalmazó lemezek, a DVD-k, VideoCD-k és zenei lemezek lejátszására.

A nagy gyártók jó ideig ódkodtak attól, hogy hasonló termékekkel álljanak elő, hiszen az MP3-fájlok mindig a jogsértés szinonimájaként szerepeltek. A világ azonban nem áll meg, így az érdekeltek lassacskán elfogadják az új formátumot, és a nagy cégek is vagy a saját nevük alatt árulnak ilyen lejátszókat, vagy legalábbis a háttérrel, az alkatrészeket biztosítják ezekhez.

A Umax lejátszójával kapcsolatban is volt egy gyanúm: DVD-meghajtót és ilyesfajta elektronikát csak nem gyártanak minden utcacsarkon, valamilyen nevesebb gyártó kell adja a készülék belsejét. Kicsomagolás után azonnal szét is szedtem, hogy körülnézzem a belsejében.

Meglepetésemre odabent néhány áramköri lap, egy apró műanyag lábakon álló olvasóegység, egy tápegység és jó sok levegő fogadott. Az egész belső bőven elférne egy feleekkorás házban is – de hová lenne akkor a tekintély?

A gépezet áramellátásáról a már említett tápegység gondoskodik, ez külön lapon kapott helyet. A meghajtónak saját vezérlőelektronikája van, amelyet – meg mernék rá esküdni – szabványos IDE-kábel köt össze a dekódoló áramkörrel. Ennek lelke – a hatalmas felirat miatt nehéz lett volna nem észrevenni – egy Zoran lapka; ezzel a névvel különféle MJPEG- és MPEG-tömörítő-és -kibontókártyákon már éveken ezelőtt is találkozhattunk, a gyártó azóta is élvonalbelinek számít ezen a területen. Természetesen van előlapja, kijelzője és hátoldali kivezetései is a készüléknek, de ezek teljesen általános alkatrészek.

A nagy leleplezés viszont nem sikerült, egyik világmárka neve sem díszelgett a gépezet belsejében. Megnyugtató a tudat, hogy a mindennapos alkatrészeknek köszönhetően talán háziagos javítása sem lehet megoldhatatlan feladat. A készülék dobozában minden szükséges kellék meg-

találunk. Tartozik hozzá egy távirányító, amihez nem sajnáltak adni két darab elemet, illetve az a 392 forint sem hiányzik a költségvetésből, amit az euroscart, illetve egy RCA átkötő kábellel elbukik a gyártó. A kézikönyv sajnos csak angol, német és francia nyelven szól hozzánk, bár egy egyoldalas papírlapon a kezelőszervek nevét és a legfontosabb műszaki adatokat azért magyarul is megtaláljuk. A sötét színű lejátszó remekül illeszkedett ugyancsak sötét videomagnómhoz és tévéhez, de csomagolásának jelölései szerint feketében is létezik, ha valakinek ilyen az óhaja.

Üzembe helyezés után elsőként egy DVD-t etettem meg vele, gondolván: ha ezt lejátszsa, nagy baj már nem lehet. Amikor az előlap apró panelje lebillent, és így bukkant elő a szögletes tálcá, kicsit meglepődtem; számítógépes meghajtókban is találkoztam már ilyesfajta kivittel, és nekem nem tetszik. Külső megjelenését tekintve más rosszat a legnagyobb szigor mellett sem lehet elmondani a készülékről. A DVD behelyezése után azonnal megkezdődött a lejátszás; ha nem lennék kíváncsiak a bevezető szövegre vagy a rekláma (hozzá kell tennem, pofátlanságnak tartom, hogy a tisztességesen megvásárolt lemezen hirdetés legyen...), akkor a megfelelő gombot megnyomva azonnal a főmenüre ugorhatunk. Itt a megszokott DVD-s stílusban választhatjuk ki a kívánt nyelvet, feliratot, esetleg elérhetjük a különlegességeket, vagy indulhat maga a film.

Ha türelmetlenek vagyunk, azonnal a filmre ugorhatunk, például egy-két jelenetet túllépve. Ezt hátrafelé is megtehetjük, illetve 2, 4, 8 vagy 20-szoros sebességgel mindkét irányba kereshetünk, „tekerhetünk” a filmben. A részleteket 1/2, 1/4, 1/6 vagy 1/7-ed sebességű kockázással leshetjük meg, illetve a szünet gombot nyomkodva akár minden egyes képkockát kielemezhetünk. Jómagam a lejátszó és a tévé csatlakoztatása után gyakorlatilag azonnal tudtam filmet nézni a készülékkel. Mint – egyelőre hazánkban – a legtöbb embernek, nekem is 4:3 képarányú tévém van, ez azonban semmi gondot nem jelentett. Maga a kép egy keskenyebb csíkban jelent meg, a felirat pedig szépen, jól olvasható módon alá, a fekete részre került. A kép mintha egy lepkesóhajnyit sötét lett volna, és a készülékben sem lehetett sem színtelítettséget, sem kontrasztot vagy fényerőt állítani, de ezeket a jellemzőket a tévéken általában módosítani lehet, és annyira nem volt zavaró a jelenség, hogy fárasztottam volna magamat miatta. Aki jobban el akar merülni a lejátszó lelkivilágában, az belemászhat a menüjébe. Külön állítható be, hogy a tévén megjelenő feliratok, a film felirata, a film hangjai és a DVD-lemez menüje milyen nyelven jelenjen meg. Természetesen ezeket a beállításokat lejátszás közben vagy a lemez saját menüjéből felülbírálhatjuk, sok hasznukat meg nem nagyon vesszük, ugyanis a menüpontokban csak angol és német nyelvet választhatunk, magyart nem.

Hasonlóan izgalmas a képpel kapcsolatos menüpont, ahol megadhatjuk, hogy PAL vagy NTSC tévékészülékkel



Umax DVD 6500

Jellemzők

- 96 kHz 24 bites D/A átalakító
- AC-3 5.1 dekóder
- 10 bites video D/A átalakító

Kimenetek

- koaxiális és optikai digitális hangkimenet
- S-Video aljzat
- RCA video- és hangkimenetek
- 1 db euroscart csatlakozó



rendelkezünk-e, a képarány 4:3 vagy 16:9 legyen, illetve CVBS vagy RGB kimeneti formátumot szeretnénk.

Gazdagabb a hangbeállítások tárháza, ahol az analóg és digitális kimenetre külön-külön adhatjuk meg az AC3, a DTS, az MP3 és egyéb hangok

jellemzőit. Analóg esetben két- és hatsatornás kimenetet kérhetünk, a digitális kimeneten pedig nyers (raw) és PCM formátumot választhatunk. Néhány korlátozással a táblázat elemeit szabadon állíthatjuk – sajnos sokcsatornás erősítő és hangfalkészlet nem áll rendelkezésemre, így itt nem sokat tudtam bűvészkedni.

Külön menüpontban kapott helyet a Prologic hangzás, illetve a mélyláda engedélyezése, valamint a gyerekzár kódjának és a tartalmi szintnek a megadása.

Meg kell említenem, hogy egy kicsit kényelmetlennek éreztem a menü és általában a készülék használatát abból a szempontból, hogy a műveletek végrehajtásának némi késéssel látott neki. Nem világos, hogy az elem gyengesége, netán a mechanika kényelmessége vagy a vezérlés lassúsága okozza-e ezt, ám sokszor az ütök-vágók felhasználó türelmetlenségével nyomkodtam a gombokat, nem értve, hogy miért nem történik már valami. Szerencse, hogy ha egy gombnyomás az adott helyzetben nem értelmezhető, a képernyőn egy áthúzott kör jelzi, hogy a készülék a parancsot nem fogadta.

Mivel VideoCD-eket nem sok helyen láttam hazánkban, csak az érdekesség kedvéért tettem be a Macskafogó gyári példányát. Mit mondjak, sikerült lejátszani, volt kép és hang, ám a DVD-k megjelenése óta a kis felbontású, gyengécske képminőségű VCD-k korszaka leáldozott. A zenei CD-k ugyan még nem futottak ki, ám lejátszót ezekhez egy marék lézerdiódából még egy ügyesebb tajvani óvodás is összeüt, nincs mit próbálgatni rajta.

Amitől az egész történet érdekes lesz, az az MP3-lejátszás. Hosszas keresgélés kellett ahhoz, hogy találjak egy változó sávszélességű (bitrate) tömörített MP3-állományt, de sikerült; ahogyan arról is meg tudtam bizonyosodni, hogy a készülék ezekkel is elboldogul. Meglepő módon az id3 elemekkel annál kevésbé – pontosabban semennyire, helyettük a fájlok nevét listázza a képernyőre. Miután behelyeztük a lemezt, megkezdte az első könyvtár első zeneszámának lejátszását, illetve megjeleníti a könyvtár tartalmát. A bal felső sarokban a könyvtár teljes neve megjelenik, ha nem hosszabb 7 karakternél; ellenkező esetben négy betű látható az elejéről, egy alulhúzás, majd két betű a végéről. Hát, végül is jobb, mint a semmi.

A zeneszámokban nem lehet előre-hátrapörgetni, egyenként viszont lépkedhetünk közöttük. Ha meg tudjuk jelezni, hogy például a 112-es szám a kedvencünk a lemezről, akkor közvetlenül is elérhetjük. Ehhez mindössze a +10 feliratú gombot kell 11-szer megnyomnunk, majd a 2-es gomb puha érintése után már hallhatjuk is a kívá-

lasztott művet. Ez a módszer a CD-lejátszókról ismert, és a húsz számnál többet nem tartalmazó zenei lemezek esetében jól használható, de mivel egy-egy MP3-lemezre 120–150 szám is felkerülhet, valami kényelmesebbet is kitalálhattak volna.

Érdekes, hogy az adatokat csak a tévé-képernyőn jeleníti meg a készülék, a saját kijelzőjén csak a zeneszám sorszámát láthatjuk, illetve azt, hogy pontosan hol tart a lejátszás. Televízió híján, bár nem megoldhatatlan, elég bajos az MP3-lejátszás.

Beteges MP3-gyűjtők arra is vetemedhetnek, hogy vegyesen írják meg lemezeiket, esetleg bonyolultabb könyvtárat alkossanak az egy album–egy könyvtár elrendezésnél. Vajon mi történik ekkor? Semmi.

A készülék szépen végiglépked a könyvtárakon, majd megjeleníti a tartalmukat. Igaz, nálam a második könyvtárszint állományait már egyetlen listába gyűjtötte, majd a 36 fájlból kihagyta azt a kettőt, ami adatfájl volt, illetve azt az egyet, amit egy felsőbb szinten már mutatott, így állt elő a 4-estől 36-os sorszámgig tartó lista. Azt nem egészen érttem, hogy miért a lista elejéről és nem a végéről maradtak el a sorszámok – olyan apróság ez, ami a használhatóságot ugyan nem befolyásolja, ám némiképp elnyagolt termékfejlesztésre utal.

Érthetetlen, hogy egyes emberek kiváló dolgokat alkotnak, de csak a műszaki tartalommal foglalkoznak, a megfelelő tálalással és felhasználói felülettel már nem. Vagy ami még rosszabb: csak felületet és tálalást tudnak készíteni, műszaki tartalmat nem. A Umax lejátszója az előbbi csoportba tartozik, kezelői felületének betegségeitől függetlenül kiváló kép- és hangminőséget nyújt.

Ha hasonló készülékre vágyunk, és nem ragaszkodunk a márkanévhez, akkor 10–20 ezer forintot takaríthatunk meg a vásárláskor. Érdeemes megismerkedni a drágább termékekkel is, majd eldönteni: nyújtanak-e az árukkal arányosan többet egy félig névtelen termékhez képest, és képesek vagyunk-e megalkudni néhány részlet tekintetében, a megtakarított pénzen pedig például kettővel több DVD-lemezt vásárolni.

A készüléket a Demand 2000 Kft.

(☞ <http://www.demand.hu>) bocsátotta rendelkezésemre.

A lejátszó ára 36 160 Ft + áfa.

Medgyesi Zoltán (mz@rettesoft.hu)

a BMGE 24 éves informatika szakos hallgatója.

Szabadidejét legszívesebben a barátnőjével tölti.

Szeret autózni és bográcsban főzni. A Linuxot hat

éve ismeri, de még nem volt lelkiereje, hogy

áttérjen rá. A Linuxvilág magazin hírszerkesztője.



Ami tetszett

- Kiváló hang- és képminőség
- Csendes működés
- Alapbeállításokkal is kifogástalanul működik



Ami nem tetszett

- Csúnyácska, ügyetlen kezelőfelület
- Lomhának tűnő működés



A hónap szakmai tanácsai

**Pusztulj, folyamat!**

A `ps` kimenetében mit jelent a D folyamatállapot? Miért nem engedi a rendszer kilőni a D állapotú folyamatokat? Mit lehet tenni, ha ezek a folyamatok rengeteg olyan erőforrást foglalnak le a rendszeren, amiket más folyamatok számára kellene felszabadítani?

Ankit Doshi, doshij@yahoo.com

A D állapot azt jelenti, hogy a folyamat megszakíthatatlanul alszik. Alszik, és vár egy esemény bekövetkezésére, de nem lehet megszakítani, még a `kill -9` paranccsal sem. Néha a D állapotú folya-

mat leválasztott vagy többé nem elérhető távoli állományrendszert próbál elérni. Ez esetben a befűzésnél a `mount` parancs `soft` lehetőségét használd.
Felipe Barousse Boué, fbarousse@piensa.com

Nem megy a nyomtató

Red Hat 7.2-t használok és a HP DeskJet 932C nyomtatómat szeretném telepíteni. Mindent megtettem, amit a könyv írt, az összes illesztőprogramot végigpróbáltam, de a nyomtató nem működik. Meg tudja valaki mondani, mi itt a trükk? Ez egy USB nyomtató.

Tim Fey, TFey@cfl.rr.com

Add ki a `modprobe printer` parancsot, és figyelj meg a `/var/log/messages` állományban, hogy a rendszer felismerte-e a nyomtatót. Ezután használható a `printtool` programot a nyomtató beállításához.
Christopher Wingert, cwingert@qualcomm.com

Próbáld ki egy másik USB-eszközt, hátha a nyomtatóval használt kapu nem megfelelően működik. Ha hibáüzeneteket kapsz, küldd el őket.

Don Marti, dmart@ssc.com

Nem írtad, hogy az USB-összeköttetés létrehozásával van-e gondod, vagy az USB-összeköttetés működik, csak nem sikerül adatokat küldeni a nyomtatónak. Ez utóbbi esetben azt javaslom, használd a legújabb CUPS illesztőprogramokat a <http://www.cups.org>-ról. Akár böngészőn keresztül is létrehozhatod a nyomtatóhoz szükséges beállítóállományt.

Felipe Barousse Boué, fbarousse@piensa.com

Nincs képernyő, de kellene az X

Red Hat 7.2-t használok egy Gateway PC-n, amelyben ATI RADEON 8500 grafikus kártya található. Már sok rendszerre telepítettem XFree86-ot, de ezzel elakadtam. A `/etc/X11/XF86Config-4` fájlban beállítottam a Radeon-meghajtót, és a napló szerint ez az ATI illesztőprogramjával és más egyébbel együtt ez rendben be is töltődik. Minden jónak tűnt egészen a végéig, amikor ez a hibaüzenet jelent meg:

```
(II) Primary Device is: PCI 01:00:0
(EE) No devices detected.
```

```
Fatal server error: no screens found
Chris Carlson, cwcarlson@cox.net
```

A beállítási hibák kizárásához próbáld az Xconfigurator által előállított `/etc/X11/XF86Config-4` fájlt használni.
Christopher Wingert, cwingert@qualcomm.com

Lehet, hogy frissítened kell az XFree86 4.2-es változatára (a Red Hat 7.2 eredetileg az XFree86 4.1-et tartalmazza), amely a te ATI Radeon kártyádat is támogatja (lásd <http://www.xfree86.org/4.2.0/Status6.html#6>).
Felipe Barousse Boué, fbarousse@piensa.com

RPM-lekérdezés eredményének formázása

Le akartam kérdezni a számítógémemre telepített összes RPM-csomagot, ezért az `rpm -qaR` parancsot használtam. Ez a parancs azonban rengeteg adatot írt ki, ami nem kimondottan felhasználóbarát. Elolvastam a Maximum RPM megfelelő fejezetét a `--queryformat` kapcsolóról. Ezután ezt a parancsot használtam:
`rpm -qaR "%{NAME} [%-8{REQUIRENAME} \n]"`
Az eredmény mégis pontosan ugyanaz lett, mint az előző lekérdezésé. Van valakinek ötlete, hogyan tehető a kimenet olvashatóvá, illetve használhatóvá?

Joe, joeslapnuts@yahoo.com

A `--qf` vagy a `--queryformat` kapcsolót közvetlenül a lekérdezés formátum-karakterláncra elé kell írni. Az `rpm --querytags` paranccsal az RPM-változatod által támogatott összes címkét kiírathatod.

Don Marti, dmarti@ssc.com

Ne fusson a program kétszer

Létezik a `libc`-ben olyan függvény, amellyel megállapítható, hogy fut-e már az alkalmazásom?

Pieter Coetsee, pieter@cosmosc.co.za

A legtöbb program ezt úgy oldja meg, hogy a folyamat-azonosítóját (PID) a `/var/run` könyvtárba helyezi el, egy fájlba (pl.: `/var/lrun/gdm.pid`).

Marc Merlin, marc_bts@merlins.org

A `/var/run` könyvtárba csak a rendszergazda írhat. Ha felhasználóként csak egy példányban szeretnél egy X-alkalmazást futtatni, a `gnome-moz-remote` valami ehhez hasonlót művel. Megkeresi, hogy fut-e a Mozilla egy példány, és ha igen, akkor az URL-t ebben a példányban nyitja meg, különben egy új Mozilla-folyamatot indít (lásd <http://cvs.gnome.org/lxr/source/libgnome-2/libgnome/gnome-moz-remote.c>).

Don Marti, dmarti@ssc.com

Egyéni csomagok készítése

Lehetséges-e saját IP-fejléceket létrehozni egy új statikus 32-bites mezővel, és elküldeni a célgépre?

Muguntha Kumarc, mugunth_kay@rediffmail.com

Természetesen. Vess egy pillantást a `sendip` programra: <http://www.earth.li/projectpurple/progs/sendip.html>.

Marc Merlin, marc_bts@merlins.org

A Linux Journal honlapján számtalan gond megoldásához találhattok további segítséget. A Sunsite tükörloldalait, a gyakori kérdéseket és az egyéb útmutatásokat a

www.linuxjournal.com honlapon olvashatjátok el.

A rovatban közzétett válaszokat Linux-szakértők kis csapata készítette el.

További kérdéseiteket szívesen fogadják (angol nyelven) a

www.linuxjournal.com/lj-issues/techsup.html

címen, ahol csak egy kérdőívet kell kitöltenetek, de a `bts@ssc.com` címre levelet is írhattok. A levél tárgyában szerepeljen a „BTS” kulcsszó.

Új termékek

CrossOver Office Server Edition

A CodeWeavers bejelentette a CrossOver Office Server Edition 1.2-es változatának megjelenését. A Server Edition lehetővé teszi Windows-



programok, többek közt az Office, az

Internet Explorer, a Lotus Notes és a Quicken futtatását osztott, vékony ügyfeles környezetben, Linux és Solaris operációs rendszerek alatt. A Server Edition egyidejűleg több száz felhasználót képes egyetlen kiszolgálón támogatni, így segítve a munkakörnyezet szabványosítását.

Adatok: CodeWeavers, Inc., 2550 University Avenue West, Suite 439S, St. Paul, Minnesota 55114, <http://www.codeweavers.com>

MontaVista Linux ingyenes előzetes

A MontaVista ingyenes előzetes csomagot adott ki, ami a beágyazott alkalmazások fejlesztői számára



lehetővé teszi a MontaVista Linux Professional Edition 2.1-es változatának kipróbálását. Az elő-

zetes ugyanazt a hálózatalapú fejlesztőkörnyezetet használja, mint a Professional Edition, és tartalmazza a DHCP, NFS és TFTP összetevőket. A teljes MontaVista Linux rendszermag – amely hat processzor-család tizenegy processzorán képes futni – része a csomagnak. A fejlesztők az előzetes csomag segítségével felfedezhetik az alkalmazások elkészítésének és hibakeresésének folyamatait, és kipróbálhatják a Pro Edition hálózati és valós idejű képességeit. A csomag a

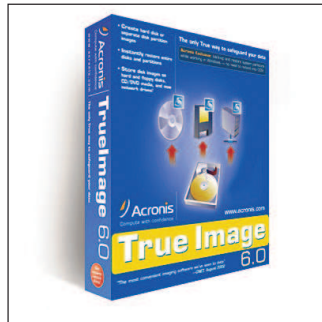
<http://www.mvista.com/previewkit/index.html> címről tölthető le.

Adatok: MontaVista Software, 1237 East Arques Avenue, Sunnyvale, California 94085, telefon: 408-328-9200, <http://www.mvista.com>

Acronis System Utilities

Az Acronis Software három segédprogram új változatának megjelenését jelentette be, amelyek vagy ön-

működően, vagy kézzel futtathatók, alkalmazkodva a különböző felhasználói szintekhez. A TrueImage 6.0 egy olyan lenyomatkészítő, amelynek segítségével adatvesztés után az összes adat és beállítás helyreállítható. A MigrateEasy 6.0 a merevlemezről merevlemezre történő költözést teszi lehetővé. Az OS Selector 8.0 segítségével egyszerre több operációs rendszert telepít-



hetünk egyetlen számítógépre, és a rendszerindításkor választhatunk közöttük; továbbá a program a merevlemez újrafelosztását is elvégzi, ha szükséges.

Adatok: Contact Acronis, 395 Oyster Point Boulevard, Suite 213, San Francisco, California 94080, telefon: 888-380-3736, <http://www.acronis.com>

Mobius Axon Wireless System

A Symbol Technologies Mobius Axon Wireless System terméke egy vezeték nélküli hálózati kapcsoló és hozzáférési pont, amely támogatja a 802.11b, 802.11a, 802.11g és a régi 802.11 frekvenciaugrós módszereket. Az 1U magas Mobius Axon Switch juttatja el az alkalmazásfüggő biztonsági elemekkel – (például titkosítás, hitelesítés és virtuális LAN (VLAN)) – védett vezeték nélküli forgalmat a Mobius Access Portokra. Az Access Portokat az asztalra, a falra vagy a mennyezetre lehet felszerelni. A háttérben működő Linux RTOS lehetővé teszi, hogy egy harmadik fél Bluetooth- és GRPS-hozzáférési pontokat adjon a rendszerhez – ezt követően az alkalmazások elérhetővé válnak.

Adatok: Contact Symbol Technologies, One Symbol Plaza, Holtsville,

New York 11742, telefon: 800-722-6234, <http://www.symbol.com>

3ware Serial ATA Controller

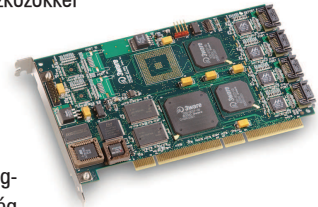
A 3ware egy olyan soros ATA RAID vezérlőkártya-családot jelentetett meg, ami nyílt forráskódú illesztőprogramokkal és kezelőeszközökkel rendelkezik Red Hat és SuSE alatt. Az Escalade 8500 sorozat több terabájt tárolóhelyet és pont-pont kapcsolattartást biztosít, valamint minden meghajtóhoz teljes sávszélességgel kapcsolódik. A kapcsolószerről kialakítás lehetővé teszi, hogy a vezérlőkártya az összes meghajtót egyszerre érje el. Az Escalade RAID-vezérlők négy, nyolc vagy tizenkét meghajtót támogatnak, a 12-es kialakításban akár két terabájt adat is tárolható.

Adatok: 3ware, Inc., 701 East Middlefield Road, Suite 300, Mountain View, California 94043, telefon: 877-883-9273, <http://www.3ware.com>

Qt 3.1

A Qt 3.1 a legújabb változata a több felületen elérhető C++ alkalmazásfejlesztő keretrendszernek, ami lehetővé teszi, hogy a fejlesztő a kódot egyetlen forrásfájlból kiindulva több felületre is lefordítsa. A 3.1-es változat újdonságai között van az ActiveQt, melynek segítségével ActiveX-vezérlőket hozhatunk létre a Qt használatával, és a Qt-alkalmazások ActiveX-kiszolgálók lehetnek. A Qt 3.1 lehetővé teszi a Qt alapú kód meglévő Motif-alkalmazásokba történő beillesztését is. További újdonságok: továbbfejlesztett tulajdonságszerkesztő a Qt Designer tervezőeszközben, ami egyszerre több elemet is képes dolgozni, továbbfejlesztett többszálúság és az osztálykönyvtárak növekvő szerepe.

Adatok: Trolltech Inc., 3350 Scott Boulevard, Building 55, Suite 2, Santa Clara, California 95054, telefon: 408-567-0212, <http://www.trolltech.com>



© Kiskapu Kft. Minden jog fenntartva

Linux Journal 2002. november

United Linux – a megoldás

Több hónapja izgatottan várjuk, mi lesz a United Linux csoportosulás vége. A négy Linux-forgalmazó (Connectiva, SCO Group, Suse és Turbo) megegyezett, hogy egy közös Linux-változatot hoz létre. A döntés mögött sokan a GNU/Linux eszmevilágától történő elszakadást, a rendszer végleges elüzettesítésére tett kísérletet láttak. A napokban a Comdex kiállításon jelentik be az új terméket, így napvilágra kerülnek a részletek. A helyzettel kapcsolatban **Greg Bogochwalski**-t, az SCO területi igazgatóját kérdeztem.



Kapcsolódó címek

Az SCO honlapja ➔ <http://www.sco.hu>

A United Linux csoport honlapja ➔ <http://www.unitedlinux.hu>

Szy György: Milyen fő okok bújnak meg a háttérben, miért is jött létre a négy vetélytárs szövetsége?

Greg Bogochwalski: Az ok leginkább a piac jelenlegi helyzete. Igazából a négy társaság nem annyira ellensége, inkább társa egymásnak. Többnyire más-más földrajzi területen dolgozunk, például a Turbo szinte csak a Távol-Keleten van jelen. A tervek szerint a négy cég nemhogy akadályozni, de egyenesen segíteni tudja majd egymást. A célok között szerepel például, hogy a világ bármely pontján legyen olyan cég, amelyik támogatja a United Linux bármely felhasználóját. Ezt a linuxos piac egyetlen szereplője sem képes felvállalni.

Sz. Gy.: A cél tehát a piac felosztása egymás között? Mennyi igaz abból, hogy a négy vállalat ezzel a lépéssel akarja a GNU/Linux világot is nyereségközpontúvá változtatni?

G. B.: Nem, semmilyen kizárólagosságot vagy ehhez

hasonlót nem rögzítettünk, számos országban egyszerre többen is jelen vannak. Ami nagyon zavaró a jelenlegi helyzetben mind a beszállítók, mind a felhasználók számára, hogy Linux és Linux között is óriási a különbség. A fejlesztők például kénytelenek többszörös kipróbálást végezni, vagy egy adott Linux-változatot kinevezni, ugyanis nem biztosított, hogy a termékük fut majd a felhasználó gépén. A második kérdés kicsit bonyolultabb. Nem az a célunk, hogy a nyílt fejlesztéseket akadályozzuk vagy megkössük. Sőt rengeteg ilyen csoportot támogattunk (például az RPM is eredetileg a Caldera támogatásával készült) és tervezzük is támogatni. Sokan azt mondják, hogy a hozzánk hasonló cégek „élősködnek” a nyílt közösségen, a tények viszont ellentmondanak ennek. Minél jobb és erősebb a nyílt közösség, annál nagyobb a felhasználói tábor, annál többen ismerik a rendszert és végső soron annál több cégnél jelentkezik az igény a szabad programokra alapuló megoldásokra. Ezek a cégek viszont igénylik, hogy a beszállító megbízható legyen, kipróbált módszereket adjon, megfelelő szakembereket biztosítson, szükség esetén elfogadható időn belül javítsa a hibákat és így tovább. Úgy vélem, a két terület – tehát az ingyen elérhető rendszereket használók, valamint a dobozos terméket és egyéb szolgáltatást vásárlók csoportja – kiegészíti, segíti egymást. A négy cég csoportja nem abból él, hogy mások munkáját eladja, hanem mindannyian komoly fejlesztéseket is végzünk.

Sz. Gy.: A csoport tehát egy közös Linux-változatot fejlesztett ki, és mindannyian azonos rendszereket forgalmaznak majd?

G. B.: Itt nem arról van szó, hogy egyetlen termék jött létre. A United Linux maga csupán a rendszer alapja, a dobozos termékek első két lemeze lesz. Az első lemez tartalmazza az XML alapú telepítőt, illetve a 2.4.19-es maggal szerelt rendszert, a második lemezen pedig az összes GPL-es anyag kap helyet (a második lemez a neten is elérhető lesz). Ez az alaprendszer tartalmaz minden olyan fontos elemet, ami a terjesztés része kell legyen. Maga a United Linux is egy működő rendszer tehát, de csupán egy váz, amelyre mindegyik forgalmazó még felaggatja a saját megoldásait, eszközeit. Így például a levelezőrendszer vagy a tűzfalmegoldás nincs megkötve, mind a négy cég a saját elképzeléseit követve alakítja ki termékeit.

Sz. Gy.: Milyen távlati céljai vannak a United Linuxnak?

G. B.: Egyértelmű célunk, hogy olyan rendszert alakítsunk ki, amelyet mind a külső fejlesztők, mind a rendszergazdák elfogadnak. Gondolok itt olyan nehézségekre, mint például egy Oracle-rendszer telepítésére Linuxok alatt. Egy ilyen komoly rendszernél az ember elvárja, hogy az általa használt terjesztésre is hitelesítve legyen. Az együttműködés keretén belül a négy cég együtt dolgozhat a különböző típusú hitelesítések megszerzésén és illesztések kialakításán. Az előbb említett Oracle például a 9i után már minden termékét két rendszerre hitelesíti, a United Linuxra és a Red Hat Linuxra. Ugyanakkor



az IBM-től kaptunk támogatási nyilatkozatot. Tehát úgy néz ki, ezzel át tudjuk hidalni a szétmorzsolódó Linuxok jelenlegi legnagyobb nehézségét.

Sz. Gy.: Milyen céljai vannak az SCO Groupnak a United általános céljai mellett?

G. B.: Az SCO-n belül külön hangsúlyt kívánunk helyezni az irodai felhasználásra. Ehhez a két legfontosabb feltétel, hogy teljes megoldásokat kínáljunk, valamint hogy megfelelő felhasználói támogatás legyen. A Linux ma már megszokott a hálózathatárokon (tűzfalak, gyorstárak, útválasztók stb.), de célunk, hogy az irodai kiszolgálókon és az asztali gépeken is képes legyen a Linux a feladatait ellátni. A felhasználói támogatásban nagy előny számunkra, hogy a cég 1989 óta működik, remek szakembereket tudhat maga mellett, valamint kiterjedt felhasználótámogatási hálózata van világszerte. Terveink szerint hamarosan folyamatos, a nap huszonnégy órájában hívható támogatóközpontokat hozunk létre. A központok a különböző termékekhez különböző szintű szolgáltatásokat adnak majd.

Sz. Gy.: Milyen termékeket várhatunk, illetve ezekhez milyen támogatás vehető majd igénybe?

G. B.: Először nézzük végig a kiszolgálónak készült csomagokat. Mindegyik cég maga alakítja ki a csomagjait, illetve a csomaghoz járó szolgáltatásokat. Ami azonos, az alapcsomag, ez nálunk BASE néven fut (a teljes neve: SCO Linux 4.0 Base Edition, powered by United Linux 1.0). Ez a csomag egyébként mind a négy cégtől azonos áron kerül forgalomba. A fizetési rendszer nálunk „per server” működik, tehát nem a processzorok vagy a felhasználók száma után kell fizetni, hanem a rendszer futtató gépek száma után. A csomag tartalmaz egy év követést (egy webes bejegyzés után), de semmilyen támogatás nem jár hozzá.

A második fokozat a CLASSIC, amihez webes anyagok formájában telepítési és beállítási segítséget biztosítunk, illetve öt munkaórányi támogatást. A csomaghoz elektronikus formában személyes tanulást segítő anyagok, felkészítők és egyéb irodalom is jár. A CLASSIC szinttől felfelé ezeken kívül mindenki használhatja webhelyünk elektronikus szolgáltatásait.

Komoly támogatás jár a fentiekén túl a BUSINESS csomaghoz: vásárlóink akár telefonon is segítséget kérhetnek reggel tíz óra és este fél hét között, célunk pedig, hogy bármilyen gondra három órán belül megoldást tudjunk találni.

A legnagyobb testvér pedig az ENTERPRISE lesz, amihez folyamatos támogatást nyújtunk a nap huszonnégy órájában. Ezt külön telefonszámmal oldjuk meg. Természetesen nyitva hagyjuk a lehetőséget a külső cégekkel történő együttműködésre, amikor is a beszállító (például az Areco Systems Kft.) külön szerződés keretében adja a támogatást.

Az ügyfélgépekre szánt termékekkel kapcsolatban még nincs véglegesen kialakult kép, egyelőre annyi biztos, hogy jóval ötven dollár alatt tervezzük kihozni, a VMS-hoz (Volution Messaging Server) hasonló üzenettel.

Sz. Gy.: Ha jól tudom, a VMS egy MS Exchange kiszolgáló kiváltására készült rendszer. Az üzenet lényege tehát, hogy a MS termékei

mellett további választási lehetőséget kínálnak?

G. B.: Igen, sőt! A VMS amellett, hogy teljes körűen képes kiváltani egy MS Exchange kiszolgálót, minden Outlook-változattal együttműködik, de a legfontosabb üzenete az ára. Ezek a termékek az MS árainak a fele alatt kerülnek forgalomba.

Sz. Gy.: Visszatérve a támogatáshoz, ha jól tudom, nincs elég hazai szakemberük, hogy nagyobb mennyiségű hívást kezeljenek. Ezek a támogatóközpontok hol lesznek helyileg, illetve a kezelők beszélnek-e majd magyarul is? Emellett érdekes kérdés, hogy nyújtanak-e támogatást a linuxos GPL-es programokhoz is?

G. B.: Nem, egyelőre nem indítunk magyar nyelven beszélő támogatóközpontot, jelenleg az európai térséget kiszolgáló központ Angliában található, a huszonnégy órás szolgálatot pedig a világ három különböző pontjáról szolgáltatjuk. Az Enterprise vásárlóinak természetesen nem kell, hogy este egy ausztrál számot hívjanak, a hívás egy európai központba fut be, onnan továbbítjuk az éppen ügyeletes irodához. A támogatott programok köre pedig megegyezik a lemezekben lévő programokkal, tehát ha a lemezen megtalálható, segítséget nyújtunk hozzá. A dobozos termékben nem szereplő egyéb linuxos programokhoz (mivel rendkívül sok ilyen van) nem tudunk támogatást biztosítani.

Sz. Gy.: Az oktatással kapcsolatban két téma is felmerül, egyrészt az United Linux, tehát az alaprendszer megtanulását segítő anyagok, illetve az SCO által hozzáadott egyéb programok és rendszerek használatának az oktatása. Mennyire tervezik átvenni az United Linux csoport tagjai egymástól az oktatási anyagokat, illetve hogyan épülnek majd fel az oktatási anyagok? Ha valaki levizsgázik például a SuSE oktatóközpontjában, külön kell-e vizsgáznia a többi változatra is?

G. B.: Igen, nagy előnye a United Linuxnak, hogy az alapoktatást mindenhol egyértelműen az alaprendszerre készítjük, egy United Linux-tanfolyamot elvégző felhasználó vagy rendszergazda számára mindegy, hogy melyik termékkel kell később dolgoznia. Az SCO-nál harmincegy-néhány modul készült el, további néhány kifejezetten csak a United Linuxszal foglalkozó kiegészítés. Ha valaki egy elfogadott alapvizsgával rendelkezik, csak ebből a két-három kiegészítő modulból kell levizsgáznia. Terveink szerint egy oktatási központot is indítunk, ahol az alapszintet rendkívül szerény díjazás ellenében el lehet végezni.

Sz. Gy.: Köszönöm a válaszokat!



Szy György
a Linuxvilág főszerkesztője,
a Kiskapu Kiadó vezetője. Mindenki
levelét örömmel várja a következő
levélcímen: Szy.Gyorgy@linuxvilag.hu



Amikor a nagyok összegyűlnek...

Idén is, mint már több éve, megrendezésre került az Internet Hungary névre keresztelt kétnapos konferenciasorozat. A rendezvény elsődleges célja, hogy olyan cégek számára adjon lehetőséget a kitekintésre, amelyek tevékenysége szorosan kapcsolódik az Internet, illetve a korszerű távközlési eszközök világához.

Tihanyban, a Balaton partján, nyugodt környezetben vehettünk részt az előadásokon. De nem is a környezet a lényeg, hanem hogy a rendezvény szervezői szokásukhoz híven idén is izgalmas és fontos témaköröket gyűjtöttek össze, és valóban szakavatott és illetékes előadókat kértek fel a szereplésre.

Az ideai témaválasztást elsősorban az internethasználat hazai elterjedtségének lesújtóan alacsony mutatói határozták meg. Előadók és fórumok igyekeztek választ és megoldást keresni a hazai helyzet miatt felmerült kérdésekre, igényekre. Olyan igényekre, amelyek kiszolgálását az emberek legtöbbször az államtól várják, pedig gyakorta hallani azt a kijelentést, hogy az állam csak ne szabályozzon semmit, sőt a rendezvényen is sokszor hangoztatott üzenet volt a *politikamentes informatika*. Ennek ellenére, vagy éppen ennek érdekében, szinte minden előadáson jelen volt valamelyik minisztérium. A programtervet nézve először arra gondoltam, hogy az egész előadás kezd politizálásba fordulni. Szerencsére kellemesen csalódtam.

Kellemesen, ugyanis a hivatalos szervek képviselői nyitottak voltak, látszott, hogy ma már minden szinten látják és tudják, igény van a fejlődésre, a fejlesztésre, a gyors felzárkózásra. Ennek érdekében még forrásokat is hajlandók megmozgatni. Öröm az örömben, hogy ezt a lépést sajnos csak a jelenlegi, tényleg szomorú helyzetben teszik meg az illetékesek.

De mik is voltak a fő kérdések? Kovács Kálmán informatikai és hírközlési miniszter már megnyitó beszédében hangsúlyozott egy gyászos számadatot: jelenleg a *magyar háztartások mindössze hat százaléka* rendelkezik internetezésre alkalmas géppel. Megjegyzem, ez korántsem azt jelenti, hogy ennyien is interneteznek! Ez a szám azonban még a „keleti blokkban” is elkészerítő! Ha hozzávesszük, hogy hazánkban nemhogy gyorsult volna a széles sávú hálózatok fejlesztése, hanem lassult, érthető, miért aggódnak odafent is.

Természetesen a vita a maga szokásos, unalmas és értelmetlen módján kezdődött, a kereskedelmi társaságok képviselői szapulták az Axelerót, hogy miért nem szolgáltat ingyen, az államot, hogy miért nem ad mindenkinek „ajándékba” gépet, hogy miért nem csökkenti a szolgáltatók terheit, és különben is, miért nem szól rá felülről az Axeleróra, hogy a jogászok miért nem alakítanak ki rendes törvényeket és a többi. Igen, igen, szép volna, ha ilyen egyszerűen meg lehetne oldani a dolgokat... de könyörgök, a világ nem így működik! Később, ahogy egyre több érdekes előadást hallhattunk, tisztult a kép, sőt az egyik legizgalmasabb – és szerintem leghasznosabb – vitafórumon, amelyen a politikai szerencsétlenkedések megvitatása volt a fő cél (résztevők:

Baja Ferenc KITKH, Csepeli György IHM, Majsai Sándor Gábor OM, Rogán Antal OGY, Stumpf István Századvég), épeszű eszmecserenek lehettünk fültanúi. Kiragadnám az egyik érdekes kérdést, az elektronikus aláírás ügyét.

Elektronikus aláírás

Jelenleg Magyarországon, ha az ember bármilyen hivatalos ügyet el akar intézni, személyesen kell megjelennie az állami szervnél. Ez a törvény. Pedig mennyivel kényelmesebb volna, hogyha például ingatlantulajdon-átírást szeretnék intézni, egyszerűen felmegyek a földhivatal honlapjára, kattintgatok öt percig, majd pár nap múlva örömmel nyugtázom, hogy megérkezett a hivatalos értesítés. De hogyan azonosítsam magam a minisztérium, vagy a példánál maradva, a földhivatal honlapján? No, pont erre lenne jó a elektronikus aláírás. Természetesen ez a hivatalos szervek oldalán is rendkívül hasznosnak bizonyulhatna: gyorsabb és önműködő adatfeldolgozás, csak ellenőrizni kell; kevesebb hiba, kevesebb papírmunka, takarékosabb, papírmentes iroda stb. A kép, bevalom, egy kicsit utópisztikus. De lépünk tovább!

A törvény oldaláról még több ilyen kérdés merül fel, a hogyanok és a mikéntek egész sora, de érdemben nem tud vele senki sem foglalkozni. A vitafórumon nagyon gyorsan nyilvánvalóvá vált, hogy ez mindkét nagy „politikai erő” érdeke, majd a két fél „képviselésében” Rogán Antal és Baja Ferenc megegyeztek, hogy ha tudják, a szükséges tervezetet még az év vége előtt átvereksszik a gépezeten. Ugyanezt – gyanítom – ha levélben beszélték volna végig, eltartott volna pár hónapig. Természetesen ez nem azt jelenti, hogy januártól minden működni fog! Ez annyit tesz, hogy végre elkezdhetünk(-nek) dolgozni azon a rendszeren, amelyik alapja lesz a majdani munkának, ami megalapozza azt az együttműködést... és a többi. No igen, Rómát sem egy nap alatt építették fel.

Penetráció

Ez a szó nálam – a rendezvényen gyakran emlegetett kulcsszavak között – elnyerte a gusztagustalanság netovábbjának pálmáját. Mindenki a penetrációról beszélt, hogy az milyen kicsi. Ez alatt pedig a már említett internethasználat mutatóját értették. De azért mégiscsak vagányabban hangzik, ha azt mondom, hogy „a hazai penetráció sajnálatosan alacsony”. A hallgató számára úgy tűnik, hogy lám-lám, végre valaki tudja, mit beszél. Mint már említettem, az elején még féltem, hogy idén sem tapasztalunk mást, mint az állam bácsira való mutogatást és nyafogós asztalcsapkodást, hogy miért nincs ingyen internet mindenhol.

Szerencsére sok érdekességet is hallottunk, például azt, hogy a minisztérium is rájött, hogyha lenne egy rendes saját gerinchálója, döbbenetesen sokat megtakaríthatna, továbbá a hálózatfejlesztést is serkentené. Ezt a hírt néhányan szomorúan fogadták, mondván, hogy az állami megrendelések az IT-piac összmegrendelésének több mint egyharmadát teszik ki – most akkor a piac egyharmada csődbe megy? A felvetésre választ is kaptunk, méghozzá meglepően értelmeset: az államnak nem az



a célja, hogy a neki beszállító cégeknek minél húsozabb falatokat adjon, mintegy kitarva egy réteget, hanem hogy a piacot segítse a fejlődésben.

Gondolkodtam, hogyan is lehet ezt a hat százalékot gyorsan növelni... Mint tudjuk, sok helyen igénylik az Internetet. Ahogy én látom, három fő csoportra bontható a Világháló (vagy ha úgy tetszik: széles sávú kapcsolatot) nem használók tábora (természetesen rendkívül leegyszerűsítve). Egy: sokan egyszerűen nem akarják használni. Nem kell nekik és kész. Kettő: seregnien sejtik ugyan, hogy mi is az Internet, de nem értenek hozzá, és nem tudják hol mikor és kitől lehet elsajátítani a használatát. Három: a kutya sem hoz nekik elérhető áron Internetet a házba. Az első csoport kemény dió, és biztos, hogy nem rövid távú programokkal lehet megtörni a jeget. A második csoport az oktatás témaköréhez tartozik, és csak a harmadik csoport tagjait lehet azok közé sorolni, akik tombolnak, mert használnák, de nincs rá lehetőségük.

És ne gondoljuk, hogy ez a csoport csak a világ végén lévő csöpp települések fiatal lakóiból áll. Példáért nem kell messze mennem a szerkesztőségemig. Mi ugyanis a nyolcadik kerületben, a Népszínház utcában vagyunk. Itt akár modemes netkapcsolatra is lehetőségünk nyílik. Vagy ha nagyon komolyan összeszedi magát a szolgáltató, akár ISDN-es kapcsolatra is. Széles sávú kapcsolat viszont nincs – a Blaha Lujza tértől három sarokra. Jobban mondva van, ha fizethetek havi nyolcvan-százezer forintot egy telefonvonalal azonos kategóriás erősségű bérelt vonalért.

Mit is lehet valójában tenni? Mit tehet egy ház lakóközössége, ha oda éppen senki nem hoz kábelt, de a lakók – rúgja meg a ló – mégiscsak szívesen neteznének? Ilyenkor jön a jó öreg összefogás? De ki és hogyan építsen ki kapcsolatot? Vajon hova kapcsolódjon? Nos, pont ebben a két kérdésben segíthetne az állam, ugyanis Magyarországon jelenleg csak a kiváltságosok kiváltságosai juthatnak rendes gerincpáncsolathoz, illetve rendelkeznek olyan szakemberekkel, eszközökkel, amelyek elengedhetetlenek a minőségi szolgáltatás megindításához. Vagy nem csak a nagyok? Egy kis segítséggel nagyon sok olyan kisvállalkozás indulhatna be, ami – mondjuk megbízási alapon és nem szolgáltatásként – hálózatépítést tud vállalni társasházak számára. A gerincvonal kérdése már keményebb dió. Mert rendben van, hogy a ház rengeteget megtakaríthat egy belső hálózat kiépítésével, de honnan jöjjön a manna?

Ezért csillant fel a szemem, amikor a kormányzati informatika fejlesztéséről szóló előadáson hallottam a „stratégjáról”, meg az újonnan kiépítendő állami gerincről. Lehet, hogy egyszer (a próféta szóljon belőlem!) lesz rendes gerinc Magyarországon, amire elérhető áron rá lehet majd csatlakozni? Mert ha például a gerincet az állami fejlesztések kapcsán a hálózati kapcsolat után fizetett éves díj töredékéért lehet majd fenntartani, továbbfejleszteni, az államnak ez csak jót jelent. Az pedig egyenesen főnyeremény volna, ha egy ilyen gerinc

fenntartásával és a nagyközönség felé történő megnyitásával a nethasználat rettentő rövid idő alatt többszöröse erősödhetne! Van azonban még egy komoly akadály: a tudáshiány.

Jó pap holtig tanul!

A rendezvény második napján az oktatás volt a fő téma, ezt a napot *Magyar Bálint* nyitotta meg. Hadd idézzek a beszédéből egy-két gondolatot!

Kedvezmények vagy más jellegű ösztönzés szükséges a gépvásárláshoz. Terveiben megemlítette a „svéd modellt”, amelynek lényege, hogy a munkáltató adókedvezményt kap, ha a dolgozónak otthonra vásárol gépet, a közvetlen adókedvezményt (például a magánszemély szja-jából leírhat egy keretet, a tervek szerint 60 000 forintot gépvásárláskor), illetve a fejlett, multimédiás oktatóanyagok támogatását. Emellett bírálta a gondolatot, hogy az ECDL-vizsgát a diploma előfeltételévé tegyék, hasonlata szerint ez olyan, mintha azt mondanánk: aki nem tudja, hogy a fogkefe melyik végét kell a szájába tenni, az nem kap diplomát.

Hosszú távra az Oktatási Minisztérium a NAT felülvizsgálatát, a technikai keretrendszer megteremtését, net-alapú fejlesztéseket tűzött ki célul, valamint olyan oktatóanyagok kifejlesztését, amelyek „a tankönyvek méltó mellékletei tudnak lenni”. Természetesen rengeteg munka van még ezen a területen, akár a felnőttoktatást, akár a kisebb települések oktatási helyzetét tekintünk.

Más területeket is érintettek az előadások: külön ülésrész foglalkozott például a netes reklámokkal, sőt a nagytesó-hatást is elemezték. Nagyon érdekes volt hallani, hogy a szakemberek kimutatást készítettek, ami szerint a VV és a BB a széles sávú internethasználatot nagy mértékben előmozdította, ugyanis a felhasználók sokasága csak azért vásárolt széles sávú kapcsolatot, hogy ezt a két műsort élőben nézhesse. Erre mondják, hogy az egyik szemem sír, a másik nevet. Írásomban valóban csak gyors bepillantást tudtam nyújtani a két nap eseményeibe, és remélem, hogy az ott elhangzottak tényleg komolyabb párbeszédet indítottak el, és hamarosan mindannyian élvezhetjük ezek gyümölcsét. Külön érdeklődéssel várom a szervezők másik rendezvényét, a februárban megrendezésre kerülő IT Konferencia 2003-at!



Szy György
a Linuxvilág főszerkesztője,
a Kiskapu Kiadó vezetője. Mindenki
levelét örömmel várja a következő
levelcímen: Szy.Gyorgy@linuxvilag.hu

Kapcsolódó címek

Az Internet Hungary 2002 konferencia honlapja
➔ <http://www.mediahungaria.hu>

Az IT Hungary 2003 konferencia honlapja
➔ <http://www.ithungary2003.hu>



A jogi tanácsadás megfelelő kerete egy jogász-ügyfél kapcsolat, amely egy adott helyzet minden tényállását figyelembe veszi, és a helyileg érvényes jognak felel meg. Bár ezt a cikket egy jogász írta, a benne foglalt adatok nem helyettesíthetik az esetre szabott, bejegyzett jogásztól származó tanácsadást.

Még mindig ellenezük az UCITA-t!

Ködösség, alkalmazhatatlanság és elégtelen változtatások – jobb volna előlről kezdeni!

Nemrég a Red Hat egyik jogásza arra kért, hogy csatlakozzam ahhoz az Államok Törvényeinek Egységesítése Megbízottainak Nemzeti Gyűlése (National Conference of Commissioners on Uniform State Laws, NCCUSL) számára készített beadványához, amelyben a testület arra kéri, hogy változtassák meg az UCITA, a Számítógépes információk ügyleteit szabályozó egységes törvény (Uniform Computer Information Transactions Act) elfogadásáról szóló 1999-es döntésüket.

Korábban már kifejtettem a véleményemet az UCITA-ról (Linuxvilág, 2002. július). A rovat olvasói emlékezhetnek rá, hogy az UCITA olyan mintának szánt törvény, amelyet azzal a szándékkal hoztak létre, hogy minden állam elfogadja, ily módon egységesítve a programok felhasználói engedélyének jogi szabályozását. Az UCITA alapértelmezés szerint olyan szabályokat tartalmaz, amelyek életbe lépnek, ha egy terjesztési engedélyből lényeges feltételek kimaradnak. Az UCITA további célja, hogy meghatározza, mely terjesztési feltételek állnak szemben a közérdekkel, amelyek nem léphetnek érvénybe, még akkor sem, ha az engedély tartalmazza őket.

A Nyílt Forrás Kezdeményezés (Open Source Initiative) nevében írtam az NCCUSL-nek, hogy utasítsa el az UCITA-t. Azért tettem így, mert az UCITA egyelőre megolthatlanul hagy számos, a nyílt kódú programok terjesztői és felhasználói számára lényeges kérdést. Noha az UCITA közelmúltban elkészített függelékei foglalkozni kezdtek sajátos szempontjainkkal, a törvényjavaslat továbbra is elhibázott, hiányos, félrevezető, valamint elfogult a jogdíjas programok terjesztői javára.

Az UCITA megfogalmazói számos függeléket javasoltak sajátos kérdéseink kezelésére, de még mindig csődöt mondtak az ügyünkben. Íme a javaslatok és az okok, amiért azok számunkra elfogadhatatlanok.

Egy közelmúltbeli kiegészítés szerint „az a szerzői jogi nyilatkozat, amely csupán engedélyt ad a program használatára és nem képezi szerződés részét, nem tartozik az UCITA hatálya alá.” Ez vagy közhely (a szövetségi szerzői jogi törvény mindenképpen elsőbbséget élvez az állami szintű szerződésjoggal szemben), vagy nem alkalmazható számos olyan nyílt forrású terjesztési engedélyre, amelyeket szerződésnek szántak. Úgy látom, ez a kiegészítés a GNU General Public License (GPL) felhasználói számára próbál megoldást nyújtani. A GPL szerzője arra szólít fel, hogy az engedélyt kizárólag szerzői jogi nyilatkozatnak tekintsük. Mi a helyzet azokkal a terjesztési engedélyekkel, amelyek megfelelnek a nyíltforráskód-meghatározásnak (Open Source Definition

☞ <http://www.opensource.org/docs/definition.php>), és amelyeket szerzőik szerződésnek szántak? Ez a kitétel nem segít rajtuk; egyszerűen nem alkalmazható rájuk. Egy másik függelék a terjesztési engedély kibocsátóját felmenti a hallgatóságos jótállási kötelezettségek alól, „ha a program ingyenes (a program másolatainak átadása, illetve a használat vagy terjesztés ellenőrzése

mögött nem áll nyereség vagy a gazdasági haszon szerzésének szándéka)”. Ebben a függelékben a „free” szó „ingyenes”-t jelent, nem pedig a program szabad (free) használatához, másolásához, módosításához és terjesztéséhez való jogot, illetve a program kódjához való hozzáférést, ami ezeket a jogokat értelmezhetővé teszi.

A szabadság e második értelemben a Szabad Szoftver Alapítvány (Free Software Foundation

☞ <http://www.fsf.org>) és a Nyílt Forrás Kezdeményezés

(☞ <http://www.opensource.org>) meghatározó elve, ám az UCITA-ban nem játszik szerepet. Az UCITA javasolt megfogalmazása kódos és félrevezető, amennyiben olyan kifejezésekre épül, mint a „nyereségszerzés szándéka” és „alapvetően gazdasági hasznosításra irányuló”. Ez a jogdíjas programok terjesztői számára – akik elrejtik a programok kódját, és korlátozzák azok szabad másolásának, módosításának és terjesztésének a jogát – lehetővé teszi, hogy részesüljenek a jótállás alóli kivételek előnyeiben, annak ellenére, hogy ezzel ügyfeleiket tevőlegesen akadályozzák abban, hogy a programot „minőségileg megfelelővé”, illetve „meghatározott célra alkalmassá” tegyék.

Egy harmadik függelék szerint a más programokkal való együttműködés céljából végzett kódvisszafejtést egy terjesztési engedély nem tilthatja meg. Ez egy fontos lépés – noha inkább csak tyúklépés – a Digitális Évezred Szerzői Jogi Törvény (Digital Millennium Copyright Act) által oly súlyosan megkurtított méltányos használati (fair use) jogok biztosítása felé. Sajnos a szövetségi törvények elsőbbsége miatt ennek a kitételnek a hatása valószínűleg csekély. Továbbá ez az elképzelés nem egyenértékű azzal, mintha az NCCUSL határozottan kinyilvánítaná, hogy minden olyan terjesztési engedély, amely a programok méltányos használatának jogát korlátozza, tisztességtelen és ellentétes a közérdekkel. Attól félek, hogy a jelenlegi gyenge és korlátozott, kódvisszafejtéssel kapcsolatos UCITA-függelék sokakat abba a hitbe ringathat, hogy a korábbi jogaik álltak helyre. Meggyőződésem, hogy az alapoktól kell újragondolni az UCITA-t, figyelembe véve azt az újonnan kialakult környezetet, amelyben nyílt forrású programok versenyeznek a tőkeerős cégek által terjesztett jogdíjas, zárt kódú programokkal. Az UCITA nem sok segítséget nyújt a bíróságoknak a nyílt forrású programok terjesztési engedélyeinek betartatásában, vagy a kongresszusnak abban, hogy a köz érdekében visszaállítsa a méltányos használat jogát. Ennek hiányában a nyílt forrás közösségének nincsen szüksége az UCITA-ra.

Linux Journal 2002. november, 103. szám



Lawrence Rosen

(☞ <http://www.rosenlav.com>) magán-gyakorlatot folytató jogász. A Nyílt Forrás Kezdeményezés ügyvezető igazgatója és jogtanácsosa (☞ <http://www.opensource.org>).

Ők mondták

Az embereknek általában a BIND és a Sendmail jut az eszükbe, amelyek tényleg nagyon fontosak, de van egy sokkal gyakorlatiasabb megközelítés, amely szerint a szabad és a nyílt forráskód segítette az Internet terjedését. Ez az a döntés, ami a forrást felfedő böngészők megalkotását is eredményezte. A forrás állandóan elérhető. Az emberek nem úgy tanulták meg a HTML-t, hogy első lépésben megvették Tim (O'Reilly) könyvét. Ehelyett lenyúlták egymás weblapját, átalakították a saját igényeik szerint, majd azután vették meg Tim könyvét, hogy utánanézzenek, legközelebb hogyan csinálhatnák jobban.

(Lawrence Lessig)

Alapjában véve mindenkinek, aki a kereskedelemben dolgozik, megvan a saját napirendje, és ez nagyon jól van így, mert az embert arra sarkallja, hogy a gyakran ütköző napirendek ellenére olyan rendszert próbáljanak elérni, ami tényleg mindenkinek megfelel.

(Linus Torvalds)

Ha tudod, hogyan működik a forráskód, sokkal több esélyed van a nehézségek elhárítására. Jobban össze tudod kapcsolni a programot az operációs rendszerrel. Kevesebbet kell költeni a karbantartásra, azaz a költsé-

gek is kisebbek lesznek. Ugyancsak sokkal kevesebbe kerülne házon belül fejleszteni a programot, hogy pontosan úgy működjön, ahogyan akarsz.

(Egy kiskereskedés üzletvezetője a High Streetről)

Az (amerikai) Nemzetbiztonsági Hivatal továbbra is feladatának tartja az operációs rendszerek védelmére irányuló kutatást általában véve, különösen a továbbfejlesztett biztonsági rendszerű Linux-prototípusra épülő kutatások folytatását. A nyílt forrás kutatóival fenntartott kapcsolataink nagyon eredményesnek bizonyultak, és reméljük, a jövőben tovább bővíthetjük az ilyen jellegű együttműködéseket.

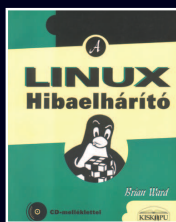
(Grant M. Wagner, az Egyesült Államok Nemzetbiztonsági Hivatalában a biztonságos rendszereket kutató részleg (Secure Systems Research Office) műszaki igazgatója)

Szilárd véleményem, hogy a VCR olyan az amerikai filmgyártók és az amerikai közönség számára, mint a bosztoni fojtogató az otthonában egyedül tartózkodó nőnek.

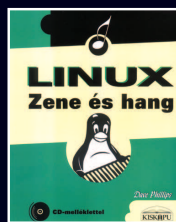
(Jack Valenti mondta a Kongresszus előtt, 1983-ban)

Linux Journal 2002. november, 103. szám

Kapu a Linux világába



Ár: 3220 Ft
281 oldal
felhasználói szint:
kezdő, haladó
melléklet: CD



Ár: 4900 Ft
397 oldal
felhasználói szint:
kezdő, haladó
melléklet: CD



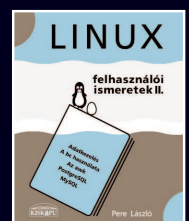
Ár: 2660 Ft
256 oldal
felhasználói szint:
kezdő-haladó



Ár: 6440 Ft
672 oldal
felhasználói szint:
kezdő-profi



Ár: 2660 Ft
256 oldal
felhasználói szint:
kezdő



Ár: 2660 Ft
256 oldal
felhasználói szint:
kezdő

www.kiskapu.hu



Linux-index

1. Ennyiszer gyorsabban fut az új RealNetworks Helix Universal, mint a Microsoft médiakiszolgálójánál Linuxon: 4
2. A Linux-felhasználói csoportok száma: 500
3. Ennyi országban alakult Linux-felhasználói csoport: 80
4. Ennyi millió dollár bevétel származott a Linux-eladásokból 2001-ben: 80
5. Várhatóan ennyi millió dollár bevétel származik a Linux-eladásokból 2006-ban: 280
6. Ennyi milliárd dollár bevétel származott a Microsoft Windows-eladásokból 2001-ben: 10
7. A Linux éves növekedési mértéke a Tower Group szerint: 30%
8. A Linux éves növekedési mértéke az IDC szerint: 37%
9. A vállalati informatikai költségvetés ennyi százalékát fordítják Linuxra: 9
10. A kiskereskedelmi üzletvezetők ennyi százaléka használ nyílt forráskódú programokat: 32
11. Az IBM ennyi telekommunikációs üzletfele használ Linuxot: 50
12. Ennyi IBM ISV kiszolgálón található Linuxon működő alkalmazás: 3800
13. Ennyi Linuxszal foglalkozó szakember található az IBM-nél: 5000
14. Új-Zélandon ennyi Compaq kiszolgálót váltottak ki egyetlen IBM Z Series nagygéppel: 150
15. Ennyi Linux-példány található a new yorki Solomon Smith Barney cég Z Series kiszolgálóján: 62
16. Ennyi Linux-példány található a Koreai Légitársaság IBM Z Series kiszolgálóján: 10
17. A Koreai Légitársaság ennyi munkatársa használja a nagygépen azóta állandósult menetrendkezelő rendszert: 4000
18. A Sharp állítólag ennyi ezer Zaurus PDA-t vitt ki a LinuxWorld Expóra 2002 augusztusában arra számítva, hogy mind el fog kelni: 3

Források

- 1.: RealNetworks
- 2–3.: Linux Counter
- 4–6.: International Data Corp., a CNET-ben is megjelent
- 7–9.: International Data Corp., melyre az IBM hivatkozott
- 10.: Tower Group, az Open Forum Europe idézetében
- 11–17.: IBM
- 18.: Hírforrás a LinuxWorld Expo

Linux Journal 2002. november, 103. szám



Cím	Szerző	Ár	Kiadás dátuma	ISBN
Linux Hibaelhárító + CD	Brian Ward	3220 Ft	2001. június	963 9301 24 8
Linux Zene és hang + CD	Dave Phillips	4900 Ft	2002. március	963 9301 31 0
Unix/Linux héjprogramozás	Dr. Büki András	2660 Ft	2002. szeptember	963 9301 10 8
Linux rendszerfelügyelet	Marcel Gagné	6440 Ft	2002. szeptember	963 9301 40 x
Linux felhasználói ismeretek I.	Pere László	2660 Ft	2002. augusztus	963 9301 37 x
Linux felhasználói ismeretek II.	Pere László	2660 Ft	2002. november	963 9301 47 7
Számlázási cím		Postázási cím		
Megrendelő neve:	Megrendelő neve:			
Cég neve:	Cég neve:			
Cím:	Cím:			
Telefon:	Telefon:			
Fax:	Fax:			
E-mail:	E-mail:			
A megrendelőlapot erre a címre kérjük elküldeni (postán vagy faxon):		Kiskapu Kft. 1081 Budapest, Népszínház utca 29.	Telefon: 303-9119	Fax: 303-1619

A megrendelőlapon feltüntetett kiadványokból az általam megadott tételeket postai utánvétellel megrendelem.

.....
cégszerű aláírás

A digitális szakadék áthidalása Dél-Afrikában

A Nyílt Forráskód Közössége a nyelvi nehézségek megoldását tűzte ki célul a soknyelvű Dél-Afrikában.

Dél-Afrikában az emberek tizenegy hivatalos nyelvet használnak. Ugyanakkor a gazdagok és szegények közötti szakadék is itt a legnagyobb a világon, s ez a megosztottság a műszaki területeken még szembe-tűnőbb. Ez az informatika világában létező szándékon kívül is kizárta a tömegeket a fejlődésből, és így számos dél-afrikai lakos számára elérhetelenné vált az informatika véráramába való bekapcsolódás. A faji elkülönítésen alapuló politikai rendszer (apartheid) társadalmi-gazdasági körülményeit és kiegyensúlyozatlan oktatási alapelveit a nyelvi korlátokkal együtt a kirekesztés külön-külön tényezőiként ismerték fel.

Dr. Neville Alexander politikai aktivista – akit *Nelson Mandela* egykori elnökségének idején a Robben-szigeten börtönöztek be és jelenleg az „Alternatív oktatás Dél-Afrikában” felmérés téma-vezetője – mindig is kulcsfontosságú szereplő volt a nyelvi vitában. Alexander az egynyelvű társadalomtól való elmozdulás szükségességét hangsúlyozza.

A „csak angol” vagy akár a „főképp angol” társadalmi berendezkedés a legtöbb embert, és illetéknéppen az ország egészét a középszerűség állapotára ítéli, minthogy az emberek képtelenek természetes megnyilvánulásokra, magabiztos és alkotó gondolkodásmódra, ha nem használhatják anyanyelvüket.

A virtuális világbeli bolyongás az első alkalommal utazók számára önmagában is elég ijesztő, tehát anélkül is, hogy a sajátjuktól eltérő nyelven kelljen megtenniük. A számítógépes programok olyan dél-afrikai nyelvekre történő átültetése, amilyen például a xhosa, tavalyig úgyszólván fehér holló volt. A legtöbb számítógépes program kizárólag angolul férhető hozzá, és Afrika második legelterjedtebb nyelve, az afrikaans is csak gyengén támogatott. A többi kilenc hivatalos nyelv semmilyen programban nem található meg – érdekes megemlíteni a zulut is, Dél-Afrika leggyakrabban beszélt nyelvét, amelyet közvetlenül a xhosa követ.

A fenti nehézségek megoldása érdekében létrehozták a [translate.org.za](http://www.translate.org.za) web-

helyet (☞ <http://www.translate.org.za>), avégett, hogy a különféle számítógépes programokat lefordítsák a Dél-Afrika-szerte használt hivatalos nyelvekre. Ez az első vállalkozása a Zuza Program Alapítványnak (Zuza Software Foundation), amely egy olyan nem haszonelvű szervezet, ami a programok fejlesztését és a nyílt forráskódú programok használatát segíti elő Afrikában. A Zuza Program Alapítvány – neve az egyik helyi nyelven ingyen adott ajándékot jelent – programhonosítási, üzletfejlesztési és oktatási javaslatokon dolgozik, nyílt forráskódú programok felhasználásával. Ugyanakkor a Zuza más vállalkozásért is felelős: a Linuxlabért

(☞ <http://www.linuxlab.org.za>), amely a hátrányos helyzetű iskolák számára biztosít felújított számítógépeket és ingyenes támogatást.

A fordítási feladat a KDE honosításával kezdődött, a figyelem azonban másra terelődött, mert a KDE nem vihető át az egyik futtatási felületről a másikra. A KDE 3.0-ban már választható a xhosa, a zulu és a venda nyelv. Mind a fordítással foglalkozó, mind a műszaki csoport most a Mozillára és az OpenOffice-ra összpontosít, mert Windows-, Linux- és Mac-környezetben egyaránt használhatók, és ennek köszönhetően a felhasználók számára megkönnyítik a programok telepítését, valamint egyre több embernek mutatják be a nyílt forráskód közösséget. A Zuza lelkiismeret-furdalás nélkül használja ezeket a programtermékeket, hogy népszerűsítse az operációs rendszereket.

A fordítás a gettexttel, a GNU honosítási rendszerével és eszközeinek segítségével zajlott – a gettext előnye abban áll, hogy szétválasztja a programozást és a fordítást. A Zuza programban használt fordítások kiegészíthetők vagy éppen javíthatók, mivelhogy nincsenek belefördítve az alkalmazásba. A fordítócsapat a tényleges fordítás elvégzéséhez a KBabel programot használja. A grafikus felhasználói felület (GUI) rendkívül fontos, mivel a fordítók nem rendelkeznek elmélyült műszaki ismeretekkel, s a grafikus környezet könnyebben

használható felületet biztosít munkájukhoz. A Mozilla és az OpenOffice programok a lefordítandó szövegeket saját állománytípusokban tárolják, amelyek nagymértékben eltérnek a gettext PO-állományaitól.

Dwayne Bailey, a www.translate.org.za weblap alapítója és igazgatója úgy vélekedik, hogy egy eszköz jobb, mint három: „Már sok energiát fektettünk abba, hogy megtanuljunk a KBabelt, ami képes dolgozni a gettext PO-állományokkal. Nehéz elvárni a fordítócsapatunktól, hogy elsajátítsa egy új eszköz használatát, vagy kézzel végezze az egyes állományok szerkesztését, ráadásul ez nem is igazán eredményes. Nem akarjuk, hogy olyan környezetben dolgozzanak, amelyben három eszköz is szükséges, inkább azt szeretnénk, ha a munkánk gyümölcsét mindhárom feladatban zavartalanul lehetne használni. A fenti gondok megoldásaként a csapat olyan átalakító programokat fejlesztett ki, amelyek a Mozilla DTD állományait PO állományokká alakítják, valamint olyan héjprogramokat, amelyek az OpenOffice állományokat PO állományokká alakítják és fordítva. Ezek a segédprogramok megóvják a felhasználókat a különböző állományformátumokkal való bajlódástól, és lehetővé teszik, hogy a már elkészült munkákat a csapatok könnyedén újra és újra felhasználják. Az egyes állományformátumokra történő átalakítás pedig a szakmailag képzett emberek kezében van, nem pedig a fordítókéban.

A Linuxban a KDE honosítása már előrehaladott állapotban van; de amint a választható nyelvek köre újjal bővül, a hiányosságokra is fény derül. Ennek kedvező hozadéka is van: ez pedig számos új szolgáltatás a honosítási keret számára. Az olyan programoknál, mint a Mozilla vagy az OpenOffice, egyelőre honosítási éretlenséget tapasztaltunk, például helytelenül kezelik a megnevezett dolgok többes számát, amit például a KDE már helyesen kezel. A nyílt forráskód előnye éppen abban rejlik, hogy a hiányosságokat gyorsan orvosolják.” Egy újabb terület, amely segíti a feladat

előrehaladását és közreműködik a honosításban, egy webalapú fordítóeszköz létrehozása, amely az egymástól távol dolgozó fordítók számára lehetővé teszi, hogy könnyedén dolgozzanak közösen a feladaton, ráadásul anélkül, hogy ehhez valamiféle programot kellene telepíteniük, de legalábbis Linuxot kellene használniuk. Bailey egyértelműen megfogalmazza,

nyílt forráskód közössége már egy éve igen. Bailey úgy látja, hogy sok nyelv elvesztette a jelentőségét a fajüldöző politika története során, ez pedig a nemzeti és nyelvi büszkeség hiányához vezetett. Engem a Linux-felhasználók német és francia környezetben végzett munkája ébresztett rá arra, hogy ez a dél-afrikai nyelveknél is működne. Remélem, hogy az az emberek számára

A fordítócsapat elsőként kiválasztotta a Dél-Afrikában használatos tizenegy hivatalos nyelvet. Amint ezek a feladatok érett szakaszba lépnek, a csapat kitekint a határokon kívülre, az afrikai földrész más területeire is. „A nyílt forráskód utat nyit az afrikaiak számára, hogy önkiszolgálóvá váljanak, ne kelljen a fejlett világra várniuk, elkezdhessek és el is végezhessek saját teendőiket. Rajtunk kívül valószínűleg senki sem fog programokat honosítani szuahéli nyelvre” – vélekedett Bailey.

A munka jelenlegi szakaszában a translate.org.za webhely fiatal diplomásokat foglalkoztat fordítóként. Ezenkívül olyan hallgatók is dolgoznak nekik, akik a fokvárosi egyetem (University of Cape Town) tandíjartzóásaikat a gazdasági hivatalokon keresztül az itt végzett munkával egyenlítik ki.

Magától értetődő, hogy a nyelvet körülvevő egyéb nehézségek – például az egyenlőtlenség és a szegénység – általánosak és az élet minden területén éreztetik hatásukat, de a Zuza célja az, hogy a támogató szerepében lépjen fel. „A programok lefordítása nem váltoítja el az összes korlátot a számítógéphez való hozzáférés elől – állítja Bailey –, de egyet legalább segít kiiktatni. Ez az erőfeszítés az alacsony költségű számítógépekkel, a nyílt forráskódú programokkal és az olcsó internet-hozzáféréssel együtt nagy hatást fog gyakorolni a dél-afrikaiakra, különösen a hátrányos helyzetben élőkre.”

Az első lépés a programok lefordítása, de az emberek gépek elé ültetése nélkül lehetetlen a kívánt hatást elérni. Ez az a pont, ahol a Zuza másik tevékenységi körét folytató Linuxlabbal való kapcsolat fontossá válik. A két vállalkozásnak kart karba öltve kell együttműködni. A Linuxlab a hátrányos helyzetű iskolák számára alacsony költségű vékonyügyfél-megoldásokat keres, amelyek Dél-Afrikában gyakran olyan embereket szolgálnak, akiknek nem az angol az anyanyelvük.

A Linuxlab küldetése az, hogy elősegítse az iskolákban a számítógéphez és az Internethez való hozzáférést, hogy lehetővé tegye az oktatók számára, hogy tartalmat hozzanak létre és fejlesszenek, a tanulókat pedig csúcsmínőségű, csúcstechnikát képviselő informatikai berendezésekhez juttassa.

Dr. Evan Summers, a vállalkozás igazgatója úgy véli: „A tudományos hagyományok körében a tanulás és tudásmegosztás kultúráját a GNU Szabad Szoftver Alapítvány (GNU Free Software Founda-



miért fordít olyan sok időt a fordítási vállalkozásra: „A szabad programok közössége az együttműködésnek és megosztásnak olyan szellemiségét közvetíti, hogyha az ember egy ideig egy ilyen közösségben dolgozik és felismeri, mennyi mindent kapott, gyakran úgy érzi, hogy valamit neki is vissza kell adnia – legyen az csupán egy jól átgondolt hibajelentés. Valahol a hírnév ízébe való belekóstolás vágya is. A hibajelentések azonban egy idő után kezdtek unalmassá válni, és a hibaelhárítás lett az a terület, ahol úgy éreztem, valamivel magam is segíthetek, különösebb programozói tapasztalat nélkül. De nem csupán a feladat műszaki oldaláról van szó – a nyelv Dél-Afrikában meglehetősen fontos kérdés. Neville Alexander azt állítja, hogy „Dél-Afrika poszt-apartheid társadalmában a nyelvvel kapcsolatos alapelv és gyakorlat a fajüldözés-ellenes eszközök együttesében olyan fontos alkotóelem, amelytől függünk ahhoz, hogy az ország valóban és láthatóan átalakíthassuk.” Semmilyen kereskedelmi programgyártó cég nem foglalkozott még megfelelően a nyelvi kérdéssel Dél-Afrikában, de a

feltároló lehetőség, hogy a saját nyelvükön használhatják a számítógépet, büszkeséget ébreszt bennük anyanyelvük iránt. Ugyanakkor, ha az ember valamit az anyanyelvén sajátíthat el, természetesen egyúttal könnyebbé is jelent. A dél-afrikai Linux-támogatási és fejlesztési vállalkozás, az Obsidian Systems felvállalta ezt a feladatot, megosztva helyiségeiket, erőforrásaikat és szakértelmüket, hogy a fordítási munka beinduljon és működjön. A Shuttleworth Alapítvány – amely annak köszönhetően kapott nagyobb nyilvánosságot, hogy egyik alapító tagja „az első afrikai-ként” éppen mostanában járt a világűrben – biztosította annak a feladatnak a pénzügyi háttérét, amely a számítógépekhez való hozzáférés végtelen történetének épp csak a felszínét érintette. Mark Shuttleworth Thawte nevű programtermékét – melyet nyílt forráskódú programon fejlesztett – 575 millió dollárért adta el a VeriSignnak. Alapítványának egyik kulcsfontosságú célkitűzése, hogy Dél-Afrikában nyílt forráskódú programokat fejlesszen, és ez egyre nagyobb teret kezd nyerni az informatika szakterületein, a kormányzati és otthoni felhasználásban egyaránt.

tion) képes a legjobban közvetíteni, amely a tanulókat a programközösség résztvevőiként ismeri el, támogatja az IT iránti zavartalan lelkesedést, és előmozdítja az együttműködési kultúra fejlődését informatikai társadalmunkban”.

Az első Linuxlabet a fokvárosi Alexander Sinton Iskolába telepítették, amely egy a Shuttleworth Alapítvány által támogatott kiszolgálógépből és lemeznélküli X-terminálként működő felújított 486-os gépekből állt. Az egy gépre eső költség így száz dollár alatt marad, a hálózati kiépítést és a kiszolgálógépet is beleértve. A mexikói Red Escolar, amely tulajdonképpen egy a fent bemutatott tervhez hasonló elgondolás volt, a program kivitelezésének hiányossága miatt hiúsult meg. *Dr. Edgar Villanueva Nunez* egy széles körben megjelentetett, a Microsoft perui székhelyű általános igazgatójához címzett írásában a következőt állítja: „... a mexikói vállalkozás mögött működő hajtóerők a felhasználási szerződésből eredő költségeket használták fő érvként a vállalkozásunkban szereplő egyéb okok helyett, amelyek ennél sokkal fontosabbak. A fogalmi tévedés és a közoktatási államtitkár támogatásának hiánya miatt az a feltetelezés kapott lábra, hogy az iskolákba történő ingyenes rendszerek telepítéséhez az iskolák programokra szánt költségvetésének csökkentése mellett elegendő, ha helyettük GNU Linuxot tartalmazó CD-ROM-ot küldenek nekik. Ez természetesen nem vezetett sikerre, és nem is lehetett másképp, hiszen az iskolai géptermekek működésképtelenné válnak, ha jogdíjas programokat használnak, de a telepítésre és karbantartásra már nem futja a költségvetésből.” Ebben a kísérletben, amikor a számos embert avatnak be a számítástechnika tudományába, létfontosságú, hogy a nagyobb értéket a készgátvitelbe

fektessék be. Ezt tartva szem előtt Dr. Summers rendszeresen ingyenes Linux-műhelyeket rendez oktatók számára, valamint kifejlesztette a Precist, ami egy új, a Javához közel álló, a linux alatti programozás iskolai tanítására alkalmas programfejlesztési környezet. Az iskolák az oktatók és a tanulók részvételével bonyolítják le a projektet, miközben a Linuxlab segít, hogy a „géptermag” megkezdje működését, a készségek átadását és a folyamatos támogatás biztosítását az iskola számára. A Linuxlab mindezt az kéri cserébe, hogy ezek az iskolák fogadjanak oktatókat a szomszéd iskolákból, akik csatlakoznak majd a Linux-műhelyek munkájához, továbbá segítsék az újabb iskolák géptermeinek beüzemelését, amint maguk már e-megfelelők (e-ready) lettek. És így folytatódik egyre tovább ez a folyamat.

A célkitűzés az, hogy minden tanuló számára biztosított legyen a csúcsműködésű informatikai tanulási környezet, és ez nem is lehetetlen feladat. Summers érvei a következőképpen hangzanak:

„Érdekes megfigyelni az európai és amerikai cégek fejlődési irányvonalát, amelyek olyan személyi számítógépektől igyekeznek nagy mennyiségben megszabadulni, amelyek nem tehetők alkalmassá Windows XP fogadására, ugyanakkor szeméttelre dobásuk ólomtartalmuk miatt ellentmond az Egyesült Államok környezetvédelmi törvényének.

Az előrejelzések szerint az elavulttá váló gépek száma a következő tíz évben egyedül az Egyesült Államokban százmilliós nagyságrendű lesz. Ez újabb lendületet adhat a számítógép-felújító iparnak Dél-Afrikában, ami kedvező álláslehetőségeket teremt, és növeli az olcsó, gyenge működési jellemzőkkel bíró

gépekhez történő hozzáférést, amelyek azonban még mindig alkalmasak iskolai lemeznélküli X-termináloknak.” A Linuxlab segít feltárni az iskoláknak az olyan kútfőket, amelyekből lemeznélküli X-terminállal alakítható gépek nyerhetők. Az Alexander Sinton Iskolában a kiszolgálógépen Linux Terminal Server működik KDE ablakkezelővel, a translate.org.za webhelytől kapott honosításokkal beépítve. A fokvárosi Linux-közösség önkéntesei telepítik a géptermekeket. A géptermekek összeállításához az iskola önkénteseket toboroz a közösség sorából és a környező iskolákból, és ide tartozik a hálózati kábelezés és bútorzat összeszerelése is, amellyel a „géptermag” költségeit ezer dollár alatt tartják. Ez a hátrányos helyzetben levő iskolák nevelőit képessé teszi arra, hogy valóra váltsák álmukat: saját tanulóiknak is biztosítani tudják a csúcsműködésű tanulási környezetet. A műszaki háttérrel jól ellátott és az azt nélkülözők közti szakadéknak nem szabad tovább szélesednie. Csupán azoknak a gondolkodásmódját kell átszabni, akik azt hiszik, hogy a műszaki fejlődés csak a kiváltságos keveseké. A nyílt forráskód alapelv – úgy gondolják – megváltoztatja ezt a gondolatrendszert. A translate.org.za és a linuxlab.org.za két olyan vállalkozás, amely arra törekszik, hogy a műszaki fejlődés eredményeit minden dél-afrikai számára elérhetővé tegye.

Linux Journal 2002. november, 103. szám

Linda Martindale

a dél-afrikai Fokvárosban dolgozik. Idejét a helyi társadalom feszítő kérdéseinek vizsgálatával tölti. Első könyve „Celebrate Hope” címmel mostanában jelent meg.



Elektronikus levelezés rádióval Nyugat-Afrikában

Távoli hálózat rövidhullámú rádióon 2400 baudos rádiós modemekkel, Dan Bernstein qmailje és a Linux.

Guinea forró, zöld területeinek a mélyén, Nyugat-Afrika oldalsó kiszögelésén, az egymástól messze található Dabola, Kissidougou és Nzérékoré városokban szétszórtan dolgozó nemzetközi segélymunkások most már élvezhetik a rendszeres internetes levelezést, mégpedig a saját asztalukról. Se közel, se távol nem látunk egyetlen telefondrótot vagy műholdvedőt sem. A leveleket több száz kilométeres távolságra – őserdővel borított hegyeken, és pálmafákkal sűrűn benőtt szavannákon át – rövidhullámú rádió segítségével mozgatjuk. Ezt a vállalkozást rádiós levélnek kereszteltük, és a következőkben elmesélem a történetét.

A Guineai Köztársaság kesudióhoz hasonlító ország a Csendes-óceán partján, Nyugat-Afrikában, az egyenlítőtől tíz fokra északi irányban. Gyönyörű és természeti erőforrásokban gazdag ország, mely nagyjából akkora, mint Oregon állam. Az afrikai államok között Guinea a béke és nyugalom félszkének számít, és általában nem von magára sok figyelmet. Az elmúlt néhány évben Guinea csöndben hősi szerepet vállalt magára a világ eseményeinek színterén, ugyanis biztos és barátságos menedéket nyújtott félmillió olyan embernek, akiknek kegyetlen háborúk és polgári forradalmak miatt kellett elhagyniuk lakóhelyüket a szomszédos Sierra Leonéból és Libériából.

A Nemzetközi Segélybizottság (IRC) egyik legfontosabb működési területe Guinea, ahol szolgáltatásokat és támogatást nyújtanak az ország különböző részein levő táborokban élő mintegy kétszázezer menekültnek. Én akkor kerültem kapcsolatba az IRC-vel, amikor a feleségem 2001 nyarán elfogadta az országos igazgatói megbízatást ebben a programban. Hamarosan már úton is voltunk, hogy megvizsgáljuk a táborokat, és hosszú utunk során meglátogattuk a program három helyi irodáját is az ország belsejében.

Ha az ember elhagyja Conakryt, a fővárost, rögtön tapasztalja, hogy Guineában szegényes a háttérágazat, különösen ami az áramellátást és a távközlési rendszereket illeti – hogy a széles sávú internetkapcsolatról ne is beszéljünk. Az IRC helyi irodáinak ezért maguknak kell gondoskodniuk a háttérrel: dízelgenerátorral állítják elő az áramot, és rövidhullámú rádiókészülékkel tartják a kapcsolatot a többi irodával és a mozgó egységekkel, melyek akár több száz kilométerre is lehetnek.

Mivel ilyenfajta elszigeteltségre és a kapcsolódási lehetőségek teljes hiányára számítottam, igencsak megdöbbentem, amikor láttam, hogy egyik rádiósunk a felszerelését arra használja, hogy asztali gépéről elküldjön egy bináris állományt egy másik helyi irodába – s mindezt vezeték nélkül! Rádiós készülékének a tetején a személyi számítógép soros kapujára csatlakoztatva feküdt egy kopottas fekete doboz, amelynek címkéjén „9002 HF Data Modem” állt. A rádiós egy nem szabványos, MS-DOS alapú programot használt az állományátvitelre, ám ez nekem rögtön szöveget ütött a fejembe. Ha ez az eszköz bináris adatokat mozgat rádióhullámokon keresztül, miért is ne telepíthetnénk rá Linuxot és PPP alapú hálózati kapcsolatokat is?

Mivel az IRC-nek a legtöbb eszköz már eleve a birtokában volt, és mivel Linuxot és más szabadon elérhető programokat akar-

tunk használni, az egész rendszer kiépítésének költsége elhanyagolható volt. Kifejlesztettem a rendszer terveit és jellemzőit, és a rádiós levélre keresztelt rendszer immár 2002 januárja óta folyamatosan működik.

Rövidhullámon szeljük át a távolságot

Ha az utóbbi időben tértél át vezeték nélküli kapcsolatra, akkor több mint valószínű, hogy mikrohullámú kapcsolattal dolgoztál, a 802.11b szerinti magashullámhosszakkal. Ha így van, akkor tudod, hogy egy tiszta napon talán mintegy tíz mérföldes, szabad szemmel látható távolságra létesíthetsz kapcsolatot. A rövidhullámú rádió teljesen más. Hosszabb hullámái visszautükröződnek az ionoszféráról, így követik a Föld görbületét, s így a rövidhullámú jelek többszáz mérföldes távolságra is eljutnak. Conakryból Nzérékoréig (itt van az IRC Guinea legtávolabbi helyi irodája) rövidhullámon a jelek minden baj nélkül eljutnak több mint 600 kilométeres távolságra is.

A rövidhullám nagy előnye tehát az, hogy képes nagy távolságokat átszelni, és az útjába kerülő akadályokat könnyedén átgörgeti. De van egy rossz hír is: bár a távolságot tekintve a rövidhullám a nyertes a vezeték nélküliség versenyében, az adatátviteli képességet tekintve szomorú képet ad. Ha a 802.11b-t szélessávú kapcsolatnak tekintjük, a rövidhullámra inkább csaknem nullás sávként gondoljunk. Az általunk használt rádiómodemek névleges teljesítménye soványka 2400 baud! És ez még nem minden! A kétirányú rádió jellemzően olyan kapcsolattartási módszer, ami csak félduplex. Más szóval vagy te adsz (a gombot megnyomva beszélsz), vagy fogadod az adást, a kettő egyszerre nem megy. Ez, illetve a modembe beépített nagyméretű hibaellenőrzés azt eredményezi, hogy a valóságban tapasztalt kapcsolat inkább 300 baudhoz áll közel. Emlékszik még valaki a 300 baudos kapcsolatokra? Ha nem a radiokarbon módszerrel mérik a türelmedet, akkor a távoli bejelentkezéssel működő kapcsolatokról szóló álmaid bizonyára apró darabokra törnek és megsemmisülnek.

A hagyományos, tárolást és továbbítást magukban foglaló alkalmazási területek esetén viszont, mint amilyen például a szöveges elektronikus levelezés, beválik a rövidhullámú rádiózás kínálta megoldás. Gondosan oda kell figyelni a beállításokra, és amit csak lehet, egyszerűsíteniük kell. A rövidhullámú rádiózás esetében minden egyes csomag értékes.

A rádiós hálózat felépítése

Figyelembe véve a rövidhullámú kapcsolat eme lehetőségeit és korlátait, a rádiós modemeket a 2. ábra szerinti elrendezésben használtuk fel a tervezés során. Conakryben van egy állandó internetes átjárónk a Coyah nevű gépen, és a Congo nevű gépen levő rádiós modem szolgál betárcsázó elosztóként mindhárom helyi iroda számára. Bizonyos időközönként PPP-kapcsolatot hozunk létre a helyi iroda és Conakry között, és a gondosan kiválasztott ügyfél-, illetve kiszolgáló protokollokat használjuk TCP/IP-felületen. Bár e helyett UUCP segítségével is megvalósíthattuk volna a helyi levelezést, TCP/IP-re alapuló



1. ábra A Guineai Köztársaság a térképen

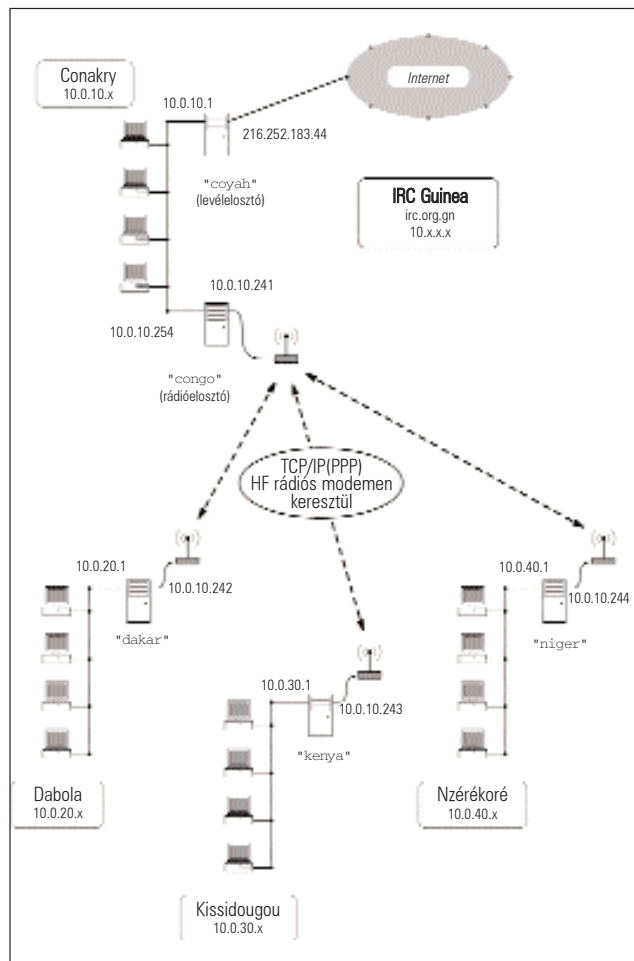
rendszerünket könnyebb beállítani és hozzáilleszteni a már meglevő hálózathoz és levelezési rendszerhez. Ezt a rendszert egyszerűbb átalakítani a gyorsabb távközlési kapcsolatokhoz, például a felszíni vonalakkhoz és a műholdakhoz, ha ezek a területen egyszer majd elérhetőek lesznek. Egyelőre viszont nincs lehetőségünk arra a kényelmes megoldásra, hogy a rádiókat teljes időben adatkapcsolatokra használjuk. A rádiós felszerelés fő célja tulajdonképpen továbbra is a szóbeli kapcsolattartás. A megvalósításnak és az eljárásoknak ezt a felhasználást nem szabad korlátozni, sőt támogatniuk kell. Mivel a rádiós adatkapcsolatok mindkét oldalon gátolják a hanghívásokat, olyan irányelveket és beállításokat használunk, amelyek biztosítják, hogy a kapcsolatok alkalmanként negyedóránál rövidebb ideig tartsanak, így a rádiók szabadok maradjanak a hanghívásokra. A helyi rádióknak meghatározott előírásaik vannak arra, hogy időközönként „tárcsázzanak be” Conakrybe, és napközben rendszeres időközönként cserélik ki elektronikusan leveleiket. A rádiók módosítják az időtervet, és szükség esetén meg is szakítják a kapcsolatot, ha sürgősen hangkapcsolatra van szükségük. A fenti okokból a behívásokat a rádiók hatáskörébe adtuk, nem pedig cron feladatokkal vagy a diald-vel intézzük őket. Itt használt felszerelésünket a Codan, egy ausztrál gyártó állította elő. Bár léteznek más gyártók is, mint például a Motorola, Kenwood és Yaesu, úgy tűnik, a világnak ebben a részében a nemzetközi segítségnyújtó szervezetek a Codan márkát választják. Nagy, fehér, hivatalos felirattal ellátott terepjáróik mindegyikének első lökhárítóján feltűnő helyre tekintélyes méretű Codan szalagantennát csavaroztak. Ezeknek a vastag, fekete antennáknak jelképes tekintélyük a legjellemzőbb vonásuk, ami például a katonai ellenőrzőpontokon való áthaladásnál tapasztalható. Mi a 9002-es típusú modemet használjuk. Ezek a modemek fel vannak szerelve a Hayes-szerű alap AT-parancskészlettel, ezért bármilyen távközlési alkalmazással könnyű beállítani, működtetni és javítani őket.

E modem és a Sportster modemcsalád között van azonban néhány jelentős különbség. Először is a Codan készüléket valójában DTE-ként (adatterminál készülék) készítették, nem pedig DCE-ként (adatkapcsolatra szánt készülék). Ahhoz, hogy soros kapura kössük, szükségünk van egy DB-9 nullmodem kábelre, amelyet *David Lawler*-nek a szöveges terminál leírásában felvázolt módján kell bekötni (<http://www.tldp.org/HOWTO/Text-Terminal-HOWTO.html>). Nem minden nullmodem kábelnek ez a bekötése, ezért ez a részlet nagyon fontos. Ezenkívül az AT-parancskészlet sem olyan nagy tudású, mint a szabványos parancskészlet, és valamennyire el is tér tőle. Továbbá ez a modem sokkal lassabban továbbítja az adatokat, mint amennyire akár csak el tudnánk képzelni.

Az mgetty és a PPP beállítása

A Conakryben levő rádiós gépet, aminek Congo (valamint radiohub) a neve, PPP-kiszolgálóként állítottuk be, ez áll készen a helyi irodák rádióin keresztül betárcsázós kapcsolatainak a fogadására. Miként a megszokott telefonos betárcsázás beállításánál, itt is az mgetty-t használjuk a soros vonal figyelésére, a modem beállítására (initialization) és arra, hogy várja a bejövő hívásokat, válaszoljon a „csengetésre”, valamint a PPP-démont elindítására.

A 9002 viszonylag jól együttműködik az mgetty-vel. Első lépésként a modemet, a gyári alapbeállításokra állítjuk be, utána biztonság kedvéért hibátűrő módba kapcsoljuk.



2. ábra A rádiós hálózat felépítése

© Kiskapu Kft. Minden jog fenntartva

A qmail összetevői

	átjáró/tűzfal	levélelosztó	rádiós elosztó	helyi gép
qmail-smtpd	igen	igen	igen	igen
qmail-qmqpc	igen			
qmail-qmqpd		igen		
qmail-qmtpd			igen	igen
qmail-pop3d		igen		igen
fastforward		igen		igen
serialmail			igen	igen
daemontools	igen	igen	igen	igen
ucspi-tcpi	igen	igen	igen	igen
djbdns	igen	igen		igen
qturn			igen	

Kikapcsolunk minden a parancsokra adandó helyi és távoli visszajelzést (E0 L0 R0); figyelmen kívül hagyjuk a hordozójelet (X0); alkatrész szintű használunk (&K3); kikapcsoljuk az önműködő választ (S0=0 &A=0); és beállítjuk az állomás címét (&I=nnnn), ahol az nnnn olyan, mint egy telefonszám, vagyis egy egyedi azonosító, melyet a többi rádió akkor hív, ha el akarja érni ezt az adott állomást.

Mi történik azután, hogy az `mgetty` válaszol a „csejengésre”, és a másik oldalról megkapja a kapcsolat létrejöttét jelentő jelzést? Ezt az `mgetty login.conf` állománya határozza meg. A betárcsázós rendszereknél gyakori, hogy `mgetty`-vel figyelgetik a bejövő PPP-csomagokat, majd önműködően indítják el a PPP-démont, a kapcsolat felépítése során rendszerint a CHAP-ot használva hitelesítésre. Ezt a beállítást a `/AutoPPP/` karaktersorozattal kezdődő sor kezeli.

Tapasztalataink szerint azonban a rádiós kapcsolat hosszú késleltetési ideje nem teszi lehetővé, hogy a `/AutoPPP/` beállítás használható legyen. Helyette alkalmazott megoldásunk egyeseknek talán megröbbentő: teljes mértékben eltekintünk a hagyományos azonosítástól! Gépeinken a bejövő kapcsolat kapcsolatbeállító állománya által átadott bejelentkezési nevet használjuk közvetlenül a PPP-démon elindítására. Amikor az `mgetty` összeegyezteti a bejelentkezési nevet a `login.conf` állomány első mezőjében található egyik bejegyzéssel, mint amilyen például a `Pklogin`, akkor lefuttatja a negyedik mezőben található programot, például a

```
/usr/local/sbin/pppd.login.kenya
```

Lényegében tehát a távoli rendszer bejelentkezési neve szolgál a hozzáférési jogosultság ellenőrzésére. Megjegyzem, hogy a `Pklogin` név valójában nem létezik a rendszerünkön, és abban is biztos lehetsz, hogy nem árultam el, milyen parancsokat használunk valójában (azt is figyelembe kell venni, hogy még a kapcsolat felépítése előtt alapos emberi azonosítási rendszert használunk, amikor a rádiósok szóban megegyeznek, hogy a kapcsolat elindítása előtt lezárják a szóban forgó csatornát). Amikor az `mgetty` megkapja a `login.conf` állományban található bejelentkezési nevet, átadja az ellenőrzést a megfelelő indítóparancsnak, mint amilyen például a `pppd.login.kenya` állomány. Ez aztán elindítja a PPP-démont, felhasználva az adott távoli gépre szabott beállításokat tartalmazó állományt, ilyen például az `options.kenya`.

Ha meg szeretnéd magadat kímélni az izzadsággal, könnyekkel és más érzelmi megpróbáltatásokkal járó keserves próbálgatástól, akkor figyelj oda a `thelcp-restart` és `ipcp-restart` beállításokra. Ezek a jellemzők adják meg másodper-

cekben mérve azt az időt, amennyit a `pppd` várni fog a kapcsolat egy-egy szakaszában a válasza, mielőtt újra próbálkozna. Ezek a jellemzők alapbeállítás szerint három másodpercre vannak beállítva, ami a szokványos telekommunikációs módszerek használata esetén több mint elég szokott lenni.

Rádió használata esetén viszont, ha nem hosszabbítjuk meg ezeket az újraindítási beállításokat, kibogozhatatlan hibahalmazzal találjuk szembe magunkat. Ugyanis az alábbi folyamat játszódik le: a `pppd` elindulásakor mindegyik fél elkezd a másikkal az egyeztetési folyamatot, hogy a kapcsolat minden egyes jellemzőjében megegyezzenek. E kezdeti egyeztetés közben, ha a kapcsolat egyik fele nem kap visszajelzést a túlsó oldalról az újrakezdésre a meghatározott időn belül, a `pppd` megismétli az adást. A közbeni időben viszont a távoli fél megkapta az eredeti adást és visszaküldi a választ. A helyi gép megkapja ezt a választ az első kérdésére, és azt gondolja, hogy ez a második kérésére küldött válasz, és így továbblép a kapcsolatfelvétel második lépésére. De ekkor a helyi kiszolgáló a második elküldött kérdésére érkező választ kapja meg, ami nem az a válasz, amire számít, ezért aztán a folyamat megoldhatatlan káoszba torkollik.

Az `lcp-restart` és `ipcp-restart` jellemzők meghosszabbításával elegendő időnyire késleltethetjük az adások megismétléséig eltelt időt, hogy mindkét fél megkapja a szükséges választ. Bőséges, 16 másodperces késleltetést állítottunk be, és azóta nem is volt több gondunk vele.

Fordítsuk most figyelmünket a vidéki állomásokra: a helyi irodákban levő távoli gépek mindegyike úgy van beállítva, hogy felhívja a Conakryben levő rádiós kiszolgálót. Hosszas tesztelésünk, kudaraink és megpróbáltatásaink után boldog és csodálatos nap volt, amikor végre sikerült létrehozni első kapcsolatunkat a guineai légtérben láthatatlanul terjedő rádióhullámokon keresztül. A Kenya nevű gépről még SSH-kapcsolatot is létrehoztunk a Congo géppel, egyszerűen azért, hogy kifejezzük kitörő örömünket, amiért sikerült „beszélni” a Conakryben ülő rádiósokkal, akik terminálképernyőjüket figyelték: „Üdvözlöt Kissiből!”

Miután végre sikerült megtalálni a megfelelő beállításokat, azt tapasztaltuk, hogy a PPP-kapcsolat, noha lassú, megbízhatóan felépíthető, és a nap minden szakában üzembiztos is marad, még olyan csatornákon is, amelyek különben zajosak és légköri zavaroktól terheltek voltak. Természetesen mindegyik rádiót és antennát a lehető legjobb teljesítményre kellett beállítani. De ha egyszer sikerült létrehozni a kapcsolatot, megnyugtató a felismerés, hogy a rádiós modemek képeknek annak fenntartására, még akkor is, amikor a körülmények közel sem kedvezőek. Ugyanis a körülményeknek valahogyan mindig megvan az a tulajdonságuk, hogy egyáltalán nem kedvezőek.

qmail a bozótban

D. J. Bernstein, a `qmail` szerzője jó néhány különleges célra szolgáló olyan eszközt és alkalmazást is kifejlesztett. Ezek tökéletesen megfelelnek a rádióval történő elektronikus levelezés céljainak, közülük központi helyen áll, hogy a `qmail` egy olyan QMTP-kiszolgálót is tartalmaz, mely Bernstein saját fejlesztésű, Quick Mail Transport protokollját valósítja meg. A QMTP egy kiegészítő programcsomag, mely a lassú kapcsolatokon keresztül történő levéltovábbításra lett kifejlesztve. A második ábrán a `qmail` a referenciahálózaton látható, ami öt gépen fut: a Coyah nevű központi levélkiszolgálón, a Congo nevű rádiós elosztón, és a három helyi irodában levő gépeken. Amint a `qmail` úgy tekint, hogy az üzenetet elküldte, várhatunk a Kissidougonban levő Kenya nevű géppel felépítendő

következő PPP-kapcsolatra. Ekkor használhatjuk a serialmail csomagot, amivel a QMTP segítségével (a körülményekhez képest) az összes, a `/var/qmail/qturn/kissidouyou/.QMAIL.PPP/` levelesládában összegyűjtött levelet átlőjük a kapcsolaton.

A helyi rádiós levélkiszolgálók mindegyike irányítás nélkül fut, és a rádió kezelője az asztali gépről egy egyszerű telnet parancssori felületen keresztül ellenőrzi. Az alapfelület négy parancsból áll, melyeket rendszerint az alábbi sorrendben futtatunk:

```
ppp.start
mail.get
mail.send
ppp.stop
```

Ezek a parancsok egyszerű héjprogramok, amelyek mindegyike elvégzi saját feladatát, és közben szerény mértékű visszajelzést is ad a kezelőnek, hogy mi is történik közben. Ezeket a feladatokat összegyűjthetnénk egyetlen parancsba is, például a `mail.run-ba`, de biztosítani akarjuk, hogy a rádió kezelőjének módja legyen megtartani ellenőrzését a rádió használata felett, figyelembe véve a szóbeli kapcsolat által támasztott követelményeket.

Nem áll szándékunkban a leveleket egyszerre küldeni és lekérdezni, először az egyiket, azután a másikat hajtuk végre. Ez is egy olyan beállítás, amit a rövidhullámú rádiós kapcsolat soványka, félduplex sávszélessége tesz szükségessé. A hálózati forgalom szempontjából olyan ez, mint az egysávos mellékutca – ha a forgalom nem kicsi, egy hosszú parkoló lesz belőle.

Afrikai algoritmus

Mint már említettük, a rövidhullámú rádiós kapcsolat csak lassú kapcsolatra alkalmas. Ennek ellenére jelentős mennyiségű levelet sikerült küldenünk a segítségével. Egy átlagos napon 300 üzenet utazik a levegő hullámain Conakry és a helyi irodák között, irodánként két-három rövid kapcsolatfelvétel során. És ahogy ez az internetes alkalmazásoknál megszokott, itt is evés közben jön meg az étvág.

A rádiós levél lényegéből fakadó korlátok figyelembe vételével igyekszünk a lehető legnyitottabb felhasználási irányelveket fenntartani. Például a személyzet tagjai szabadon használhatják a rádiós levelezést személyes célokra barátaikkal és családtagjaikkal a világ bármely részén, és az egyes felhasználók által elküldhető üzenetek számát sem korlátozzuk. A szabályzatban csak az a korlátozás szerepel, hogy megkérjük a felhasználókat, ne iratkozzanak fel levelezőlistákra.

Annak megakadályozására, hogy a nagy csatolt állományok a rádiós kapcsolatokat órákra eltorlaszolják, mint például a nagy dokumentumok vagy a grafikus állományok, a rádiókhoz csatlakoztatott qmail kiszolgálók mindegyikénél (tehát a Conakryben levő rádiós elosztó és a helyi irodák kiszolgálói esetén) 8000 karakterre korlátoztuk az üzenetek méretét. Ez elég egy négyoldalas szöveg megírására. Minden, ami belefér a 8000 karakteres korlátba csatolt állományként vagy akár összetömörítve, az mehet.

A rendszer nagyon megbízhatónak bizonyul. A Conakryben jellemző időszakos áramszünetek ellenére egy generátort és akkumulátorokat használva mindent megteszünk, hogy Coyah nevű levélelosztónk állandóan működjön. Ezeknek az intézkedéseknek köszönhetően ez a gép első üzembe helyezése óta hibátlanul működik, és e sorok leírásának idején már három hónapja folyamatosan üzemel, újraindítás nélkül.

Mindennek a megbízhatóságának viszont semmi haszna sem volna, ha a rendszert nem tudnánk hosszú időn át működtetni.

Két hónappal az első helyi rádiós kiszolgáló telepítése előtt az IRC érdeklődő és e feladatra alkalmasnak ítélt munkatársainak részvételével megalakítottuk a Hálózati felhasználók/Unix-csoportját (IRC-NU/UG). Ez a csoport rendszeres és lelkes összejöveteleket tart, hogy megtanulja a Linux, illetve a Unix használatát, valamint fejlessze a tagok hálózatfelügyeleti jártasságát. A csoportnak most már számos, éles helyzetben működő (többnyire használt alkatrészekből épített) rendszer van, amelyen dolgozni és játszodozni lehet. Az általunk telepített Linux-kiszolgálókon is fut egy egész sor további kiszolgáló és szolgáltatás, mint például DHCP, DNS, NATD, Apache, FTP, Samba és PostgreSQL. Az IRC-NU/UG olyan emberek közössége, akik fenn fogják tartani a hálózatot, és tovább fogják terjeszteni az eljövendő években.

E projekt hasznos tapasztalatainak köszönhetően a világ bármely részén könnyedén megismételhető. Iskolák, minisztériumok és más kormányzati szervek rövidhullámú rádió segítségével könnyen kiépíthetnek távoli hálózati szolgáltatásokat olyan területeken, ahol máskülönben nem lehet hozzáférni, és ezt a lehető legkevesebb költséggel tehetik meg. Ha egyszer sikerült telepíteni a rendszert, a karbantartás már gyerekjáték, és könnyen átalakítható más jellegű TCP/IP-kapcsolatokhoz is. Magának a levelezőrendszernek a karbantartása mindössze a felhasználói fiókok szokványos hozzáadását és törlését, valamint a `/etc/aliases` állomány naprakész állapotban tartását igényli.

Jelenleg mintegy ötven asztali gép és 150 fős személyzet számára nyújtunk levelezést a Guinea egész területén található négy irodánkban. A teljes, nagy kiterjedésű hálózat kiszolgálásához csupán egyetlen nyilvános IP-címre van csak szükség, melynek teljes internetszolgáltatói költsége havi 150 amerikai dollár. Az a legjobb, hogy a rendszer szabványos hálózati és internetes eljárások felhasználására épül az egész szervezeten (és egész Guineán) belül, és mindenhez szabadon felhasználható programokat használunk. Ennek nemcsak az az előnye, hogy elveti a hálózati háttér magvait ott is, ahol most még szinte elérhetetlen az Internet, de segít azoknak az alapvető ismereteknek és képességeknek a megszerzésében is, amelyek lényegbevágóak ahhoz, hogy a jövőben Afrika is csatlakozzon a hálózatokhoz.

Linux Journal 2002. november, 103. szám



Wayne Marshall

(guinix@yahoo.com) jelenleg Guineában él. Unix-programozó és műszaki tanácsadó, aki sohasem akart levelezéssel foglalkozni. Feleségével szeretnek utazni és élvezik az afrikai tájakat.

Kapcsolódó címek

- ➔ <http://www.linuxjournal/article/6299>
- Dan Bernstein qmailje és egyéb eszközei *Dave Sill* Élet a qmaillel ➔ <http://www.lifewithqmail.org>
- ➔ <http://www.nrg4u.com>
- ➔ <http://www.bushmail.co.za>
- ➔ <http://www.maflink.org>
- Codan ➔ <http://www.codan.com.au>
- ➔ <http://www.theirc.org>

A Zululandi Egyetem internetes tárkorlátrendszere

Webhasználatot felügyelő tárkorlátrendszer megvalósítása Squid címátírányítással.

AZululandi Egyetem körülbelül 6000 hallgatót számláló, „történelmileg hátrányos helyzetű” egyetem Dél-Afrika észak-keleti partjainál. A hátrányos örökség még a faji elkülönítés megszüntetése után is érezteti hatását, tanulóink főként feketék és hátrányos anyagi helyzetűek. Következésképpen költségvetésünk meglehetősen szűkre szabott, így mindennapi kihívás, hogy ebből a kevésből minél többet hozzunk ki. Ilyen helyzetben nyilvánvalóan a Linux a megfelelő választás az internetes szolgáltatások megvalósítására – mi gyakorlatilag mindenre ezt használjuk: levél-, www-kiszolgálók, DNS, DHCP, HTTP-proxik, tűzfalak és telefonos bejelentkezés. A vezetőségnek nagyon tetszik a Linux ingyenessége, ám még ennél is nyomósabb okunk van arra, hogy Linuxot használjunk – az internetes szolgáltatások kivitelezésére a Linux a legmegfelelőbb rendszer.

Az internet-hozzáférés költségei

2000 elején a tanárok és tanulóink számára a legnagyobb anyagi kihívást az internet-hozzáférés biztosítása jelentette. Az internetes adatforgalom költségei sokkal magasabbak Dél-Afrikában, mint az Egyesült Államokban. A 128 Kb-es hozzáférésünk mintegy 5000 amerikai dollárba került havonta. Ezt a sávszélességet már napközben teljesen leterhelte az a 400 oktató, akinek internet-hozzáférése volt, ezen felül a hozzáférést még a forgalmas hallgatói laborokban lévő 350 munkaállomás számára is meg kellett teremtenünk. Nagyobb sávszélességre volt szükségünk, de a költségvetés nem tette lehetővé, hogy a jelenlegi sávszélességet megkétszerezzük vagy megháromszorozzuk. Az internethasználatot figyelő és korlátozó rendszer hiányában a vezetőség nem szívesen egyezett bele az internet-hozzáférésre fordított összeg növelésébe. Vonakodtak attól, hogy nagy összegeket fizessenek ki havonta, amikor azt sem tudták, hogy kik és milyen céllal használják a Hálózatot.

Tárkorlát és hitelezés

A gondot az okozta, hogy – költségvetési keretünkön belül maradván – hogyan tudnánk elfogadható minőségű www-szolgáltatást biztosítani. Mivel az internet-hozzáférést nem szabályoztuk, az adatátviteli csatornák túlterhelése volt az egyetlen faktor, ami az igényeknek határt szabott, és ez olyan alacsony minőségű szolgáltatáshoz vezet, amit az internethasználók kemény magja éppen csak elvisel. Be kellett vezetnünk valamilyen internethasználati „költséget” annak érdekében, hogy szabályozhassuk az igényeket. A rendszer kézzelfoghatóvá tenné felhasználóinknak, hogy miibe is kerül az egyetemnek az internet-hozzáférés biztosítása. A hallgatók számára a költség előre fizetendő. Mivel a tandíjadóságok komoly nehézséget jelentenek a dél-afrikai hátrányos helyzetű egyetemek esetében, nem akartuk a helyzetet még internetes számlákkal is rontani. Ezért egy az internethasználatot szabályozó tárkorlátrendszer kidolgozása vált szükségessé. A sávszélességkorlátok azóta fellazultak, mivel tavaly új hálózati megoldást vezettek be a felsőoktatási intézményekben.

Sávszélességünk 768 Kb-es lett (384 Kb-es CIR a nemzetközi forgalomban), mialatt a havi költségek nagyjából ugyanazon a szinten maradtak. Azonban továbbra is meg voltunk győződve arról, hogy a tárkorlátrendszerre szükségünk van, ha a felhasználóknak megfelelő minőségű szolgáltatást akarunk nyújtani.

Squid és címátírányítás

Az NLANR népszerű Squid HTTP-proxy programját használjuk www-szolgáltatásainkhoz. Egy tűzfal gondoskodik arról, hogy az oktatók és a diákok csak a proxyn keresztül férhessenek hozzá a Webhez, és a proxykat úgy állítottuk be, hogy a www-használat csak felhasználói név és jelszó megadása után lehetséges. Így a Squid által létrehozott *access.log* állomány valamennyi felhasználó www-használatának bejegyzéseit tartalmazni fogja. Az *access.log* állományban lévő adatok alapján egy adott felhasználó www-használatának nyomon követése egyszerű feladat.

A Squid képes külső címátírányító programokkal együttműködni. A címátírányító beállítás után minden címkérelmet, amit a proxy megkap, az átirányító program alapértelmezett bemenetére kerül. A program kétféle választ adhat: vagy megkéri a proxyt, hogy töltsen le a kért URL mögötti erőforrást, vagy a kérést egy másik címre irányítja át. Ez a működésmód jellemző az olyan tartalomszűrő programokra is, mint a SquidGuard, ebben az esetben a címátírányító minden egyes címkérelmet összeveti egy olyan adatbázissal, amely a kiszűrendő címmintákat tartalmazza. Minden olyan kérelem, amely egy nem engedélyezett URL-re irányul, átirányítódik egy oldalra, ami közli a felhasználóval, hogy a kért oldalhoz a hozzáférés nem engedélyezett.

Rájöttünk, hogy a címátírányítót arra is felhasználhatjuk, hogy a felhasználói tárkorlátrendszert hatályba léptessük. A címátírányító ellenőrzi a felhasználó tárkorlátját, és amennyiben az engedélyezett keretet túllépte, egy olyan oldalra irányítja őt át, ami figyelmezteti erre. Bár elméletben egyszerűen hangzik, mindezeknek műveleteknek nagyon gyorsan kell lezajlaniuk, különben a proxykiszolgáló teljesítménye nagymértékben csökkenhet. A mi rendszerünkben a címkiszolgáló általában két ezredmásodpercen belül válaszol, amennyiben a kért URL engedélyezett.

A QUORUM kiszolgáló

A QUORUM (QUOta-based Resource Usage Manager, azaz tárkorlátalapú erőforráshasználat-mérő) egy kiszolgáló, egyszerű üzenet-API kétfajta üzenet típus számára: feljegyzés (egy adott felhasználó, illetve munkamenet számlázása egy adott tevékenységért) és lekérdezés (megkérdezzük a kiszolgálót, hogy egy adott felhasználó vagy munkamenet nem lépte-e túl a keretét).

A QUORUM kiszolgáló kezeli a proxy által lehívott összes cím feljegyzését, figyeli a felhasználók hitelkeretének alakulását, valamint követi a felhasználói munkafolyamatokat, amelyek egy meghatározott időtartamú folyamatos www-használatnak felelnek meg. A kiszolgáló a felhasználói és

munkamenet-feljegyzéseket, valamint a hitelkeret-követéseket egy gyorsítótárban tartja. Adott időközönként a tár a megváltozott adatai kiíródnak egy adatbázisba. A kiszolgáló egy webalapú felületen jeleníti meg a felhasználható keretet, a felhasználói munkafolyamatokat, valamint az egyéb felületi és hibakereséshez szükséges adatokat.

A kiszolgáló egy korai változatú C-ben íródott, és különböző CGI-programokon keresztül biztosította a webalapú felületet. Felhagytunk azonban ezzel a megközelítéssel, amikor találkoztunk a Java servletekkel. A jelenlegi QUORUM kiszolgáló Java servletek és JSP-k gyűjteménye, ami JDBC-n keresztül valósítja meg az adatbázissal való kapcsolódást.

A kiszolgálót a Caucho-féle Resin Servlet/JSP konténer alatt futtatjuk. Adatbázis-kezelőnek a MySQL-t választottuk, amelyhez a Resin beépített JDBC-osztályokat és adatbázis-kapcsolatsoport létrehozást biztosít. A Java servletek természetes választásnak tűnnek, amikor olyan alkalmazást akarunk készíteni, mint a QUORUM. A kiszolgáló számos, a munkafolyamatok számára szükséges állandó megosztott adatszerkezetet, munkafolyamatonkénti feljegyzéseket, valamint felhasználói feljegyzéseket és hitelkeret-feljegyzéseket tartalmaz. Mivel a servletek egyszerű Java-osztályok, amelyek egy www-kiszolgálót megvalósító konténer részei, ezeket az állandó objektumokat egyszerű a kiszolgáló részeként létrehozni, csakúgy, mint a háttérben futó szálakat, amelyek a tétlen munkafolyamatokat szűrik ki, és a megváltozott adatokat adott időközönként kiírják az adatbázisba.

Hogyan működik?

Amikor a felhasználó címet kér le, a böngésző elküldi az URL-re vonatkozó kérést a Squid proxinak. A Squid ezt a kérést továbbküldi az URL-átírányítónak. A címátírányító a querySsn üzenettel megkérdezi a QUORUM kiszolgálót, hogy a szóban forgó felhasználóhoz tartozik-e pillanatnyilag QUORUM-folyamat, és hogy a felhasználó rendelkezik-e még elegendő hitelkerettel. Ha minden rendben van, a címátírányító egy üres sort küld a Squidnek, amellyel megkéri, hogy hívja le az adott címet.

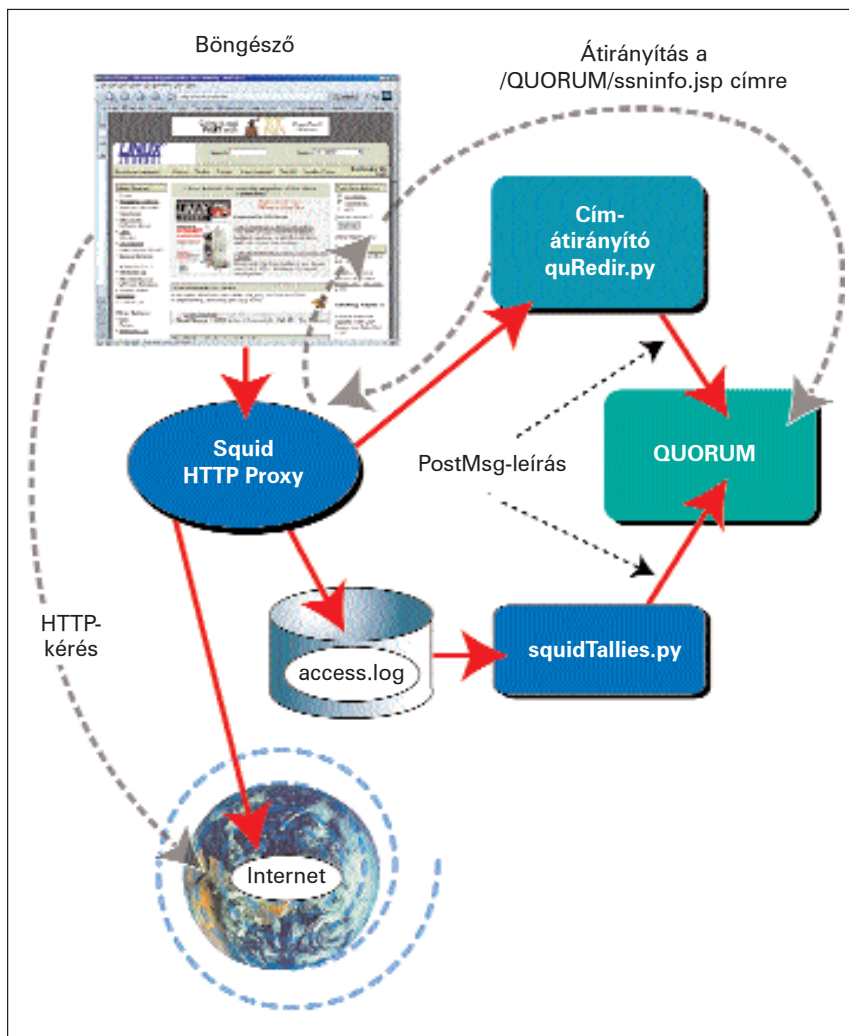
Ha a felhasználónak nincsen munkafolyamata vagy elfogyott a hitelkerete, a böngésző a QUORUM kiszolgálóban lévő *ssnInfo.jsp*-re lesz átirányítva, ami megjeleníti a felhasználó adatait, a még meglévő hitelkeretet és megkéri, hogy a *beginSsn.jsp*-re mutató hivatkozásra kattintva indítson el egy QUORUM-munkafolyamatot. Ha a felhasználónak még van hitelkerete, a JSP egy QUORUM-munkafolyamatot nyit meg a számára.

Valójában az történik, hogy a címátírányító átirányítási üzenetet küld egy servletnek, amely egy újabb átirányítási üzenetet küld az *ssnInfo.jsp*-re. A címátírányító a felhasználó nevét és IP-címét a kérés jellemzőjében adja át a servletnek:

```
/QUORUM/servlet/Redirect?ssn_id=uname@ipaddr
```

A servlet az *ssn_id* változót a felhasználó böngészőjéhez tartozó munkafolyamatba menti. Ez a többi JSP számára lehetővé teszi, hogy egy HTTP-munkamenetváltozót használjanak a felhasználóval és a munkamenettel kapcsolatos adatok lekérésére.

A webhasználat könyvelése a Squid *access.log* állományában lévő adaton alapul. Miután a Squid lekért egy URL-t, hozzáfűz egy sort az *access.log* állományhoz, ami tartalmazza a lekért



A webhasználat tárkorlátainak megvalósítása

címet, az erőforrás méretét bajtban, a lekérdező felhasználói azonosítóját, és annak a munkaállomásnak az IP-címét, amelytől a kérés származik. A *squidTallies.py* nevű Python-parancs-állomány beolvassa az *access.log* utolsó sorát, és létrehoz egy *tallySsnItem* nevű kérésüzenetet, amit a QUORUM kiszolgáló kap meg. A QUORUM feldolgozza ezeket az üzeneteket, és feljegyzi a felhasználói adatokat és munkafolyamatokat, valamint az összes tevékeny felhasználó hitelkeretét. A hitelkeret, a használat könyvelése és a felhasználói munkafolyamatok feljegyzései egy MySQL-adatbázisba íródnak. A QUORUM kiszolgáló egyik háttérben futó szála megadott időközönként átnézi a gyorsítótárban lévő munkamenet-bejegyzéseket, a felhasználó- és hitelkeretadatokat, és minden

© Kiskapu Kft. Minden jog fenntartva

megváltozott adatot kiír az adatbázisba. Ez lehetővé teszi, hogy a kiszolgáló mindig viszonylag friss adatokat tároljon az adatbázisban, anélkül, hogy túlterhelné az adatbázist azzal, hogy minden egyes címkéréskor többszörös frissítést hajt végre.

A PostMsg servlet

A QUORUM kérés üzenetei egyszerű szöveges üzenetek, amelyeket egy újsor karakter zár. Például a

```
ref000001 querySsn ssn_id=soren@1.2.3.4
↵ccode=11000
```

üzenet egy kérésüzenet, ami megkérdezi a kiszolgálót, hogy az 1.2.3.4-es IP-címen lévő soren nevű felhasználóhoz pillanatnyilag tartozik-e QUORUM-munkafolyamat és a 11000 hitelkódhoz tartozik-e felhasználható hitelkeret. Valamennyi kérésüzenet első mezője egy felhasználói referencia, amelyet a válaszüzenetben visszaküldünk az ügyfélnek, így téve lehetővé, hogy a többszálú ügyfelek összekapcsolják a válasz- és a kérésüzeneteket.

Mivel egy J2EE-alkalmazásról van szó, egy olyan üzenetváltással van dolgunk, mint például a HTTP-n keresztüli SOAP. Fontolgattuk, hogy valamennyi kérést független HTTP GET üzenetként kódolnánk a servlet számára, és aztán ezekből jönnének létre a különböző QUORUM üzenet API-k. Ez a megoldás jobb illeszkedést eredményezett volna a servlet keretrendszerhez, de a próbák azt mutatták, hogy a HTTP overhead az alkalmazásunkhoz túl nagy. Csúcsidőben a hallgatói laborokból 20–30 címkérés érkezik másodpercenként. Mivel mindegyik címkérés két QUORUM-kérésüzenetet eredményez (egy querySsn és egy tallySsnItem üzenetet), a kiszolgálónak másodpercenként 60 kérést kell kezelnie. Miután egyetlen QUORUM kiszolgálót használunk a tanári és diák proxyk kezelésére, legalább 100–150 kérelmet kell kezelnünk másodpercenként. Ezt a teljesítményt nem érhettük volna el, ha a QUORUM-kérésüzenetek megvalósítására független HTTP-üzeneteket használtunk volna, még akkor sem, ha oda-vissza üzenetekkel dolgoztunk volna egyetlen állandó HTTP-kapcsolaton keresztül. Ehelyett egyetlen servlet, a *PostMsg* kezeli le az ügyfelektől érkező összes QUORUM-kérésüzenetet. Egy ügyfélalkalmazás létrehoz egy HTTP POST-kérést, majd a POST-kérés TCP-kapcsolatát használva QUORUM-kérésüzeneteket küld, mintha egy állomány vagy más adat feltöltését végezné el a kérésben. A *PostMsg* servlet soronként egy kérésüzenetet olvas be a `ServletInputStream`-ből, a `readLine()` tagfüggvény segítségével. Feldolgozza a kérésüzeneteket, és válaszüzeneteket küld vissza az ügyfélnek a `ServletOutputStream`-en keresztül. A válasz elküldése után meghívja a `ServletHTTPResponse` objektum `flushBuffer()` tagfüggvényét, ami biztosítja, hogy a válasz tényleg el lesz küldve az ügyfélnek. Ezzel a módszerrel egyetlen QUORUM kiszolgáló több ezer kérést képes kezelni másodpercenként, egyetlen ügyfélkapcsolaton keresztül. Két apró gond azonban van ezzel a módszerrel. Az első, hogy a servlet-konténer minden olyan HTTP-kapcsolatot lezár, ami egy megadott időtartamon, általában 30 másodpercen keresztül nem működik, vagyis az ügyféllel megszakítja a kapcsolatot, ha az bizonyos idő elteltével nem küld újabb kéréseket. Az ügyfélnek ilyenkor újra kell kapcsolódnia, és el kell küldenie egy újabb HTTP POST-üzenetet, mert csak ezután lesz képes QUORUM-kérésüzeneteket küldésére. Írtunk egy Python-osztályt, ami egy Unix-csővezeték hoz létre a kérések elküldésére és a válaszok olvasására. Kérésre az osztály létrehozza

a HTTP-kapcsolatot, és elküldi a HTTP POST-kérésüzenetet. A kérésre való kapcsolódás előnye, hogy az üzenetküldések folyamata nagyméretűvé válik – a nem működő kapcsolatok nem maradnak fenn sokáig, és nem lép fel az a hiba, hogy az ügyfelek beragadnak, mivel azt hiszik, hogy a kapcsolat még él, holott a kiszolgáló már lezárta őket.



A másik felmerülő gond, hogy a HTTP/1.1 megköveteli, hogy a *Content-length* fejléccet a POST-üzenetek mind a kérés-, mind a válaszüzenetekben tartalmazzák. Míg a Tomcat ezt nem vette a figyelembe, a Resin igen, és minden olyan kapcsolatot lezár, ahol az ügyfél vagy a kiszolgáló által küldött bajtök túllépték a *Content-length* értékét. Ezt a csapdát úgy kerültük meg, hogy a *Content-length* fejléccben igen nagy értéket állítottunk be, és megbizonyosodtunk arról, hogy az ügyfél még jóval azelőtt lezárja a kapcsolatot, mielőtt a küldött bajtök a megadott értéket túllépnék.

A címátírányító teljesítménye a proxyk szempontjából kényes, hiszen minden címkérelemnek várakoznia kell, amíg a címátírányító válaszol. A QUORUM címátírányítója a legutóbbi munkafolyamatok querySsn válaszait egy gyorsítótárban tárolja. Ennek eredményeként az átírányító – ha egy munkafolyamat már létrejött és van hozzá elég hitelkeret – a címkérések alkalmával nagyon gyorsan fog működni, hiszen válaszait a gyorsítótárban lévő legfrissebb üzenetek alapján hozza létre. Ebben az esetben a választ általában két ezredmásodpercen belül vagy még ennél is rövidebb idő alatt elküldi. Még ha a válasz benne is van a gyorsítótárban, az átírányító mindig küld egy querySsn kérést a QUORUM kiszolgálónak. A válaszüzenet frissíti a gyorsítóban lévő válaszokat, így a tár mindig friss adatokat tárol a felhasználói folyamatok állapotáról és a hitelkeretről.

Ha a szükséges válasz nincs a gyorsítótárban, vagy ha a válasz kedvezőtlen (a felhasználónak elfogyott a hitelkerete), az átírányító egy querySsn kérést küld el, és egészen addig nem küld választ a proxyknak, amíg a QUORUM kiszolgálótól nem érkezett válasz. Ebben az esetben az átírányító mintegy 20–50 ezredmásodperc múlva küldi el a választát.

A Squidet úgy is beállíthatjuk, hogy a címátírányítót több példányban indítsa el, ezáltal téve lehetővé az átírányítási kérelmek párhuzamos feldolgozását. Mivel a QUORUM címátírányító több szálon fut, a Squid számos címkérelmet küldhet el egyszerre. A Squid valójában egy kisméretű beillesztett program példányait szólítja meg, amelyek egy Unix alapú hálózati kapcsolaton keresztül a címátírányítási kérelmeket a címátírányító folya-

matnak küldik tovább. Így a címátírányító valamennyi kéréshez ugyanazokat a tárolt querySsn válaszokat használhatja.

Munkába állítás és felügyelet

A felhasználói fiókok egyszerű kezelése nagy fontosságú a rendszer sikeres működtetéséhez. A QUORUM-ban ezt úgy valósítottuk meg, hogy a Web eléréséhez az összes diáknak felhasználói tárkorlással kell rendelkeznie. A QUORUM kiszolgáló az oktatók általi használatot is feljegyzi, de rájuk egyelőre még nem terjesztettük ki a kvótarendszert (az oktatói használat feljegyzése nagyon értékesnek bizonyul a hallgatók által ellopott oktatói fiókok használatának nyomon követésére).

Az adott tanórán történő internethasználat esetén az összes hallgató hitelkeretét módosítjuk egy adott mértékben, a hallgatói bejegyzések adatbázisa alapján. Ezt minden félév elején megteesszük azoknak az óráknak az esetében, ahol az oktatók internethasználati kérelemmel állnak elő.

A múltbéli tapasztalatok alapján mindig vannak olyan diákok, akik késve jegyeztetik be magukat, vagy valamilyen okból nem kerültek be a bejegyzettek adatbázisába, vagy egyszerűen csak nagyobb felhasználói tárkorlatra van szükségük. Ezért néhány tanszék számára további felhasználói fiókokat hoztunk létre. Az arra feljogosított oktatók szükség szerint további hiteleket irányíthatnak át ezekről a tanszéki fiókokról az egyes hallgatók fiókjaira. A mi tanszékünk így egy súlyos felügyeleti tehertől szabadul meg, a többi tanszék számára viszont belátást enged a felhasználói fiókok kezelésének terheibe.

Azoknak a hallgatóknak a részére, akik a kiosztott tárkorlalon felül további felhasználói hitelt kérnek, egy előre fizetett számlázási rendszert vezetünk be. Ez a rendszer nem véletlenül hasonlít a diákjaink körében nagyon népszerű kártyás mobiltelefonok számlázási rendszeréhez. A diákok az egyetemi könyvesboltban vásárolhatják meg a kártyát, ami egy titkos hozzáférési kódot tartalmaz. A számla bekapcsolható, ha a tanuló a kódot beírja egy webürlapba, és elküldi. A beírt szám MD5 hashben tárolódik, ami összehasonlításra kerül egy adatbázistáblában tárolt hash-értékekkel, és ha a hash megegyezik valamelyik tárolt értékkel, a hallgató felhasználói fiókjára terhelődik a kártyán lévő hitelkeret. Kezdeti tárgyalásokat folytattunk az egyetemünk területén vásárlási pontokat felállító cégekkel, hogy kiderítsük, árulhatjuk-e az internetes tárkorlátkártyáinkat olyan rendszerben, mint amelyet manapság a pénztáraknál a mobiltelefonok kártyáinak árulásánál használnak.

A kártyás rendszer fontos eleme annak a célkitűzésünknek, hogy felhasználóinknak rugalmas internetszolgáltatást nyújthassunk, pénzzel támogathassuk az oktatási célú internethasználatot, és fedezhessük az egyéb használati költségeket. Jelen pillanatban ez a rendszer még csak próbaüzemben működik. Tanulóink részéről erős nyomás érezhető a rendszer minél előbbi bevezetésére, mi azonban meglehetősen óvatosak vagyunk éles bevetésével kapcsolatban, hiszen ez egy olyan szolgáltatás lesz, amelyért a hallgatók ténylegesen fizetni fognak. Fontosnak tartjuk egy jól megalapozott, világosan átlátható rendszer kidolgozását, hogy a hallgatók pontosan tudják majd, mire is adnak ki pénzt.

Tervek a jövőre nézvést

A QUORUM-ot már kezdetektől fogva olyan alkalmazásnak szántuk, ami túlmutat a webhasználat könyvelésén és a tárkorlátok kezelésén. Bármely szolgáltatás, ami egy naplóállományba írja a használattal kapcsolatos adatokat, feldolgozható egy olyan kis program által, ami egy feljegyzési kérelmet küld a QUORUM kiszolgálónak. A közeljövőben bevezetjük a levélküldés könyvelését, így a nagyméretű csatlományokat

tartalmazó levelek küldőjének hitelkeretét megterhelhetjük. A levelezéssel való visszaélés (egy egyszerű AVI-film elküldése a haveroknak) visszatérő gond nálunk. A jövőben felhasználóink szembesülnek azzal, hogy ha nagyméretű állományokat küldenek levélben, elképzeltető, hogy ezért a tettükért fizetniük kell majd.

A használatot a QUORUM-ban költségkódokban tároljuk. Jelenleg a www-használat nemzetközi és belföldi forgalomra van lebontva, amelyek költsége különbözik, valamint a Squid gyorsítótárából teljesíthető kérelmekre, amelyeknek egyáltalán nincs költségük. A költségek felbontása hierarchikus: az internethasználat www-, illetve levélhozzáférésre bontható le, és ezek is további csoportokra oszthatók. A következőkben azt tervezzük, hogy költségkódokat vezetünk be a hálózati nyomtatásért, a telefonos bejelentkezésért és a tűzfalon keresztül közvetlen TCP/IP-forgalomért. A költségkódok hierarchikus rendszere lehetővé teszi, hogy az összes szolgáltatáshoz egyetlen tárkorlátot használjunk, vagy egy adott tárkorlát hitelkerete a költségkódrendszer csupán egy meghatározott részére vonatkozzon. Például elhatározhatjuk, hogy egy adott tanfolyam hallgatóinak tárkorlátja csak az internetelésre vonatkozzon, továbbá az előre kifizetett kártyákat hálózati nyomtatásra is lehessen használni.

Internetes sávcsélességünket megpróbáljuk úgy hatékonyan kihasználni, hogy felhasználóinkat bölcs használatra biztatjuk. Használatfigyelő rendszerünk azoknak kedvez, akik a nagyméretű állományokat dél-afrikai tükörkiszolgálókról töltik le, szemben azokkal, akik a tengerentúlról teszik ugyanezt – a belföldi sávcsélesség kevesebbe kerül, mint a nemzetközi. Ehhez hasonlóan kedvezményeket kínálunk az esti órákban, és arra biztatjuk felhasználóinkat, hogy a nagy állományokat ebben a napszakban töltsék le, hiszen ilyenkor kevésbé terheltek a vonalak.

Érdekes ötletnek ígérkezik egy üzenő-, illetve faliújságszerver beépítése a QUORUM-ba, amolyan helyi hálózatos azonnali üzenetküldést megvalósítandó rendszerét.

Azt már most is látjuk, hogy melyik felhasználó netezik az adott pillanatban. A QUORUM-ot úgy is kibővíthetjük, hogy üzenetet küldhessünk egy felhasználónak vagy a felhasználók egy csoportjának. Amikor valamelyik címzett címkérelmet küld el, a QUORUM átirányítja őt egy olyan oldalra, ami a faliújságot jeleníti meg.

Összegzés

Rugalmas könyvelő- és tárkorlátkezelő rendszert fejlesztettünk ki, ami együttműködik meglévő Linux és OSS alapú internetes szolgáltatásainkkal. Ez a rendszer fontos szerepet játszik abban, hogy egyetemünkön jól kidolgozott internetes szolgáltatást kínáljunk, hogy a segítségével ellenőrizhessük az internethasználatot, és hasznos adatokkal lát el minket a szolgáltatásokkal kapcsolatban felmerülő igényekről is.

Linux Journal 2002. november, 103. szám



Soren Aalto

(soren@pan.uzulu.ac.za) a zululandi egyetem Hálózati szolgáltatások osztályán dolgozik, ahol munkaidejében rendszerfelügyelettel, webalkalmazások fejlesztésével és mások

Linuxra való áttérítésével foglalkozik. Felesége előadó az egyetemen. Két gyermekük, három macskájuk, egy kutyájuk és változó számú trópusi haluk van.



Grafikus felület programozása QT-ban (2. rész)

Folytassuk grafikus felülettel rendelkező programunk fejlesztését! Kibővítjük szolgáltatásainak a körét, valamint több nyelv támogatására készítjük fel.

Cikkünk 2002 októberében megjelent első részében a Qt Designerrel létrehoztunk egy aprócska szövegszerkesztő programot. A most következő oldalakon a program működésbeli hiányosságait fogjuk pótolni, majd a `make` és a `g++`, `C++` fejlesztőeszközök segítségével felkészítjük a többnyelvű használatra. A fordítási munkához vonzó munkakörnyezetként a Qt Linguistet fogjuk használni. A Qt-alkalmazásunkhoz szükséges `Makefile` elkészítéséhez a `qmake` segédprogramot használjuk, ami egyre inkább kiszorítja a lassan elavulttá váló `tmake`-et. Az említett `tmake` a Qt régebbi változataiban használatos, Perlben írt eszköz volt. Nézzük, hol tartunk! Noha szövegszerkesztőnk grafikus felülete már képes a megnyitott ablakok bezárására és az alkalmazásból való kilépésre, még mindig nélkülözniük kell például a párbeszédablakot, amelyben a felhasználó dönthet az adatok mentése vagy a korábban mentett állomány megtartása mellett. Az **Új**, **Mentés** és **Mentés másként** menüpontok egyelőre még nem érhetőek el, a program viszont támogatja a másolási, kivágási és beillesztési műveleteket, valamint a műveletek visszavonását és megismétlését. Ezenkívül képes dőlt, aláhúzott és félkövér betűjellemzőket is használni, sőt ezek tetszőleges kombinációira is.

A **Súgó** menüben a **Névjegy** csak akkor válik teljes mértékben működőképesé, ha a grafikus felületet befordítjuk a `C++` programba. A még hiányzó szolgáltatásokat a Qt Designer saját szerkesztőjével is elkészíthetjük. Minthogy a Qt Designerben nem lehet egyszerűen a teljes projektet lefordítani, talán könnyebb, ha mégis a kedvenc szerkesztőnket használjuk.

Szerkesztőnk összeépítése

Mivel mostanra elkészültünk a felhasználói felület leírását tartalmazó `.ui` állománnyal és a megírt forráskódot tartalmazó `ui.h` állománnyal, nézzük, mi hiányzik még!

Legelőször az `uic` eszközzel (user interface compiler) az XML-ből `C++`-kódot kell készítenünk, de szerencsére a `qmake` leveszi vállunkról ezt a terhet. Egyszerűen hívjuk meg a `qmake` fordítót a Designer által létrehozott projektfájllal:

```
qmake -o Makefile lj-article.pro
```

A `Makefile` az `.ui` nevű alkönyvtárral együtt készül el, amely az `ljeditor.h` és az `ljeditor.ui.h` állományokból létrehozott `C++`-forráskódok tárolására hivatott.

Ha ez nem történik meg, ellenőrizzük a `QMAKEPATH` változó beállítását: a használandó Qt-környezet `mkspecs` alkönyvtárára kell mutatnia:

```
QMAKEPATH=$QTDIR/mkspecs/linux-g++
export QMAKEPATH
```

Nézzük meg azt is, hogy a `$QTDIR/bin` könyvtár szerepel-e a keresési útvonalban. A `C++`-állományok létrehozása az alábbi módon történik:

```
make .ui/ljeditor.h
make .ui/ljeditor.cpp
```

Ha hiba jelentkezne, ellenőrizzük a `QTDIR`, `PATH` és `LD_LIBRARY_PATH` változók értékeit. Az elsőnek a Qt-környezet `lib` és `include` könyvtárainak szülőkönyvtárára szükséges mutatnia. A `uic`, `qmake` és Designer programok könyvtárának szerepelnie kell a `PATH` változóban, míg a `$QTDIR/lib` szükséges az összeépítő (linker) számára. Arra figyeljünk, hogy a két új fájlba közvetlenül felvitt változások elvesznek, ha az `ui`-fájlt a Designerben szerkesztjük, és újabb fordítási kör is szükséges lesz. Ezért inkább származtassunk egy új osztályt az `ljedit`-ből, és abba készítsük el a kívánt változásokat. Az `uic` két kapcsolója, a `-subdecl osztaly_nev` és a `-subimpl osztaly_nev` segítségével létrehozhatjuk az új forráskódvázat. Az alábbi paranccsal

```
uic -o editor.h -subdecl Editor
  .ui/ljeditor.h ljeditor.ui
```

hozzájutunk az `editor.h` állományhoz, ami tartalmazza az új Editor-gyermekosztályhoz szükséges függvény-prototípusokat. A következő parancs pedig létrehozza az `editor.cpp` állományt a szükséges megvalósítás vázával (figyeljük meg, hogy kapcsolóként a fejállományt is megadtam):

```
uic -o editor.cpp -subimpl Editor
  editor.h ljeditor.ui
```

Ezt a két új állományt hozzá kell adnunk a projekthez, vagyis az `lj-article.pro` állományhoz, az alábbi két sor segítségével:

```
HEADERS += editor.h
SOURCES += editor.cpp
```

Ugyanezt a Qt Designerben is megtehetjük, ha megnyitjuk a projektfájlt, majd a **Project**, **Add File** menüpontot használva kiválasztjuk a két fájlt. Figyeljünk rá, hogy beillesztéskor az állományok típusát (`C++`) is megadjuk!

Az `uic`-eszköz által létrehozott kód tartalmazni fogja a szülőosztályban levő összes függvény vázát. Ezért valóban helyénvaló, ha töröljük azokat a függvényt meghatározásokat és `-vázakat`, amelyeket a részosztályban már nem kívánunk újra létrehozni. Az `editor.h` állományból távolítsuk el ezeket a sorokat:

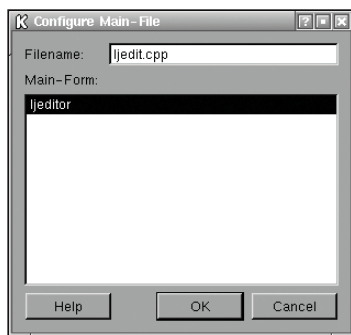
```
void fileExit();
void helpAbout();
void fileClose();
```

Ezután az `editor.cpp` állományból is töröljük a hozzájuk tartozó kódvázakat, amelyek valahogy így néznek ki:

```
void Editor::fileExit()
```

```
{
    qWarning( "Editor::fileExit() "
             "not yet implemented!" );
}
```

Természetesen a teljes programhoz még szükségünk van a `main()` függvényre. Ez megírható akár közvetlenül forráskódban, de a Qt Designer tud egy kicsit segíteni. Válasszuk a *File*,



1. kép A `main()` főfüggvény

New, C++ Main file menüpontot, majd töltsük ki a megjelenő ablakot. A következő párbeszédablakban (lásd 1. kép) adhatunk nevet a főprogramnak (én az *ljedit.cpp* nevet választottam) és a főelemnek (main widget). A Designer itt nem kínálja fel az Editor-alsztályt, így nem sok választásunk van, döntsünk az *ljeditor* mellett.

Ennek a névnek a kivá-

lasztása azt jelenti, hogy a létrehozott kódot módosítani kell.

Az *ljeditor.h* helyett az *editor.h*-t kell használnjuk, és amikor új objektumot kell létrehozni, nem az *ljeditor*-ből, hanem az *Editor*-ből kell példányosítanunk. A változtatások után *ljedit.cpp* programunk a következőképpen fog kinézni:

```
#include <qapplication.h>
#include "editor.h"

int main( int argc, char ** argv )
{
    QApplication a( argc, argv );
    Editor *w = new Editor;
    w->show();
    a.connect( &a, SIGNAL( lastWindowClosed() ),
              &a, SLOT( quit() ) );
    return a.exec();
}
```

Amint az minden Qt `main()` függvényben megszokott, létrehozunk egy `QApplication` objektumot, átadjuk neki parametereket (`argv`), majd létrehozunk a `w` nevű `Editor`-elemet. Ezt követően megmutatjuk a világnak, és az `a.exec()` függvénnyel belépünk a tényleges programba.

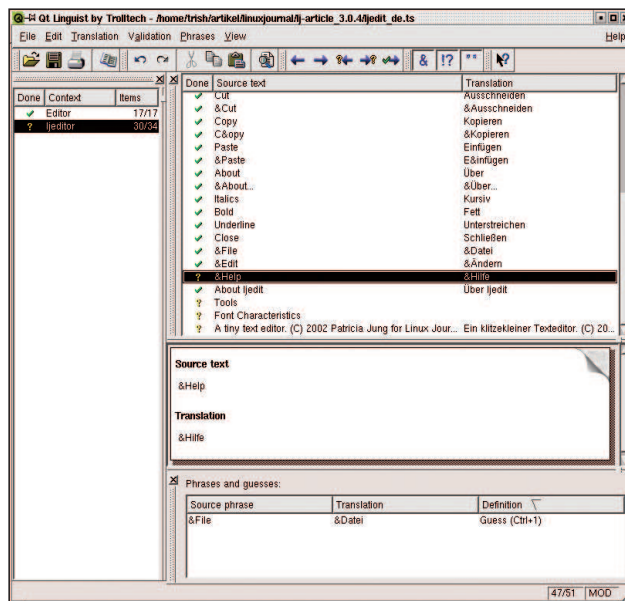
Talán magad is észrevetted, hogy az alkalmazás főelemét kijelölő `a.setMainWidget(w);` sor hiányzik – erre később még kitérek. Mivel főelem nélkül az alkalmazás nem képes kilépni, az alkalmazás `lastWindowClosed()` jelét annak `quit()` foglalatába kell kötni.

A `make`, majd a projektkönyvtárban kiadott `./lj-article` indítóparancs hatására már egy futásra kész, de még nem teljes értékű szövegszerkesztőt láthatunk.

Ha a projektállomány alapnevezésétől eltérő nevet szeretnénk választani, az *lj-article.pro* projektállományt egészítsük ki a `TARGET=ljedit` sorral.

Jöjjön a kódolás!

Alapvetően nem teszünk sokkal többet, minthogy az *editor.cpp*-ben a `qWarning "Not yet implemented"` kiírását valamilyen hasznosabb tevékenységgel helyettesítjük.



2. kép A fordítók barátja: a Qt Linguist

Amennyiben a Designer szerkesztője megfelel a céljainknak, a különböző alosztályok miatt nem különösebben kell aggodnunk, de a fordítás és hibakeresés fásasztóvá válhat, ezért én nem ezt a megoldást javaslom.

Az 1. lista az *editor.h* tartalmát mutatja be, a második lista pedig kivonat az *editor.cpp* programból. (Valamennyi lista letölthető a <http://www.linuxvilag.hu/qt> oldalról, 4722.tar.gz néven.)

A maradék négy foglalat mellett (amelyeket még nem készítettünk el) újrajrjuk a `closeEvent()` függvényt, itt kérdezzük rá a felhasználótól, hogy tényleg be kívánja-e zárni az ablakot.

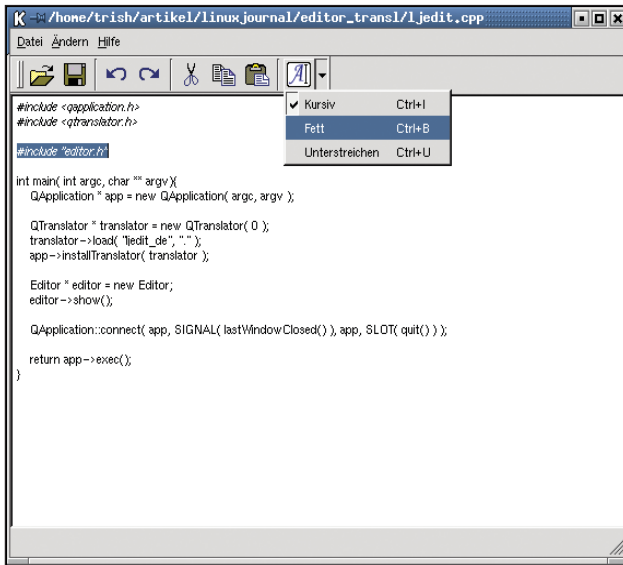
Az egyértelműség érdekében az idevágó párbeszédablak önálló függvényben, a `saveAndContinue()`-ban kapott helyet.

Az eddig említett változókon kívül két további változóra is szükségünk lesz: a `fileName`-re, ami az éppen szerkesztés alatt álló állomány nevét tartalmazza, és az `editFieldre`, a `QTextEdit`-elem másolatának tárolására. A szerkesztő részét képező (az objektumközpontú programozásnak megfelelő) létrehozó (constructor) egyetlen feladata, hogy ezeknek a változóknak kezdőértéket adjon.

Egy másik könnyű feladat a `fileNew()` foglalat megvalósítása. Ez a függvény új szerkesztőablakot hoz létre, majd meg is jeleníti. Ez az oka annak, hogy az első szerkesztőablakot nem tesszük meg mindjárt az alkalmazás főelemévé, hiszen ennek bezárása az alkalmazás összes többi ablakának bezárását is magával vonná.

De vajon mi játszódik le akkor, amikor a felhasználó bezárja az egyik ablakot vagy éppen magát az alkalmazást? A kibővített `closeEvent()` az átadott értékekkel együtt meghívja a `saveAndContinue()` függvényt: voltaképpen itt kérdezzük rá a felhasználótól arra, hogy tényleg ki akar-e lépni (159. sor). Amint az üzeneteknél szokás, itt is `tr()` függvényen keresztül adjuk át a szöveget, így segítjük a program más nyelvekre történő fordítását.

Ha a `saveAndContinue()` függvény beleegyező választ ad vissza, a folyamat folytatódik, egyébként az eseményt eldobjuk. Ha a szövegszerkesztő tartalmának a felhasználó már állománynevet is adott, vagy legalábbis már gépelt szöveget a `QTextEdit` elembe, helyes azt feltételezni, hogy a munkát meg szeretné tartani. A `saveAndContinue()` ekkor az álló-



3. kép Az ljedit.cpp (nyelvi lehetőségek kihasználása – német)

mánynevet az ablak címsorában feltüntetve üzenetablakot jelenít meg, és a mentés felől érdeklődik. Itt három gomb közül kell választani: **Yes**, **No** és **Cancel**.

A `fileName` kiírását a `%1` helyőrző segítségével oldjuk meg, ez azért fontos, mert más nyelveknél a szórend változása miatt esetleg az üzeneten belül máshova kerül a fájlnev. Ha a felhasználó a **Yes** feliratú gombot nyomta meg (181. sor), akkor a szerkesztő tartalmának mentése a megadott név alatt történik meg. Abban az esetben, ha eddig még nem kaptunk állománynevet, a `fileSaveAs()` függvény a mentés megkezdése előtt rákérdez a kívánt értékre. Ha viszont a válasz a **No** gomb megnyomása volt, akkor nem mentünk. A felhasználó minderről 2000 ezredmásodpercenként az állapotsorban fog értesítést kapni (187. sor).

A **Cancel** gomb hatására egy üzenet jelenik meg az állapotsorban, a `saveAndContinue()` függvény pedig a „nem, ne folytasd” üzenetet adja vissza. Ha az állománynév sincs megadva és a szerkesztői ablak is üres, az üzenetablak meg sem jelenik. A visszatérési érték ebben az esetben **Igaz** (TRUE) lesz: „mindenképp folytasd” (201. sor).

A `saveAndContinue()` függvény hívása a `fileOpen()` foglalatból is történhet. Ha szerkesztőablak már tartalmaz szöveget, vagy az állománynév meghatározása már korábban megtörtént, a felhasználónak lehetősége van e tartalmat lemezre menteni, mielőtt ugyanebben az ablakban egy másik állományt megnyitna.

A világraszóló program

A szöveges tartalmak `tr()`-be ágyazásával lehetőségünk nyílik a programot más nyelvek használatára is felkészíteni. Ez a lépés a `main()` függvény csekély módosítását és a tényleges fordítási munka elvégzését igényli. Ez utóbbi könnyedén elvégezhető a Qt Linguist használatával (2. kép).

Mielőtt azonban saját magad nekilátnál vagy más szakfordító(k) nekifognának a program más nyelvekre való átültetéséhez, először is magukat a lefordítandó szövegrészeket kell beszerezniük. Emiatt újabb változóval – a `TRANSLATIONS`-szel – kell a `*.pro` állományt bővíteni:

```
TRANSLATIONS = ljedit_de.ts ljedit_no.ts
```

Az iménti példa azt mutatja be, hogy ennek az alkalmazásnak a német, illetve a norvég fordítása fog megtörténni. Fontos megjegyezni, hogy az adott nyelvre jellemző fordításokat az állománynév végén szereplő ország rövidítésekkel („`de`”, illetve „`no`”) jelöljük. Amint kijelöltük a kívánt nyelveket az alkalmazásunk számára, az

```
lupdate lj-article.pro
```

parancs a nyelvészprogram számára létrehozza az XML-állományokat. A fordító most egymás után lefordítja a megadott karakterláncokat, és a munka előrehaladtával a sárga kérdőjelek zöld pipákra változnak.

Ha a bal oldalon lévő panelen látható összes osztály mellett megjelenik a zöld pipa – a fordítással készen vagyunk. A hivatásos fordítók először egy kifejezéspárból álló szótárt fordítanak le, mielőtt a program tényleges fordítását elkezdenék. A Linguist jelenleg nem támogatja a kifejezésszótárak használatát, sem teljes kifejezések fordításának bemásolását (például egy külső fordítóprogramból), így egy nagyobb program fordítási feladata rémisztően hathat a gyakorlatlan fordítókra. A programban viszont hasznos minőségellenőrzéssel kapcsolatos szolgáltatásokat találunk: a **Validation**, **Accelerators** menüpont (ha engedélyezzük) biztosítja, hogy az „&” jellel megadott gyorsbillentyűk a fordításokban is szerepeljenek. A **Validation**, **Ending punctuation** pedig biztosítja, hogy a fordítás azonos írásjellel végződjön, mint az eredeti szöveg. Amint a legutolsó lefordított kifejezés mentése is megtörtént, a **File**, **Release...** menü a fordításokat `*.qm`-kiterjesztésű, a program által használható bináris állománnyá alakítja. Ha több fordítást tartalmazó `*.ts` fájl szeretnénk terjeszteni, a

```
lrelease lj-article.pro
```

parancs segít. Amennyiben program módosítás még ezt követően is történik, az `lupdate` program a projektbe felvett `*.ts` állományokba átviszi a változtatásokat, az `lrelease` pedig frissíti a bináris változatot.

A 3. lista a már nyelvi környezetet is megfelelően használni tudó `main()` függvényt mutatja be. A program újdonsága a két beillesztett állomány: a `qtranslator.h` és a `qtextcodec.h`. A használt nyelv a rendszer nyelvi beállításától függően (a `LANG` környezeti változóban szerepel) kerül kiválasztásra (13. sor). Ha például a `LANG` változónak a `de` vagy `de_DE` értéket adjuk, az alkalmazás a `/local/lib` könyvtárban az `ljedit_de` vagy `ljedit_de.qm` nevű állományokat fog keresni. Amennyiben ilyeneket mégsem talál, a program az eredeti nyelvi változatot fogja használni. Sajnos nincs egyszerű megoldás a több könyvtárra kiterjedő keresésre, sem pedig a programba épített könyvtárnevek módosítására. Ha a `QTranslator` megtalál egy fordításállományt, akkor betölti azt, és onnantól kezdve a fordításokat használja (15. sor). Az `ljedit` német változatát a 3. képen láthatjuk.

Linux Journal 2002. október, 102. szám



Patricia Jung

(trish@trish.de) rendszergazda, szakíró és szerkesztő egy személyben, és ilyen minőségében örömmel tölti el, hogy élhet előjogaival és kizárólag Unix-, illetve Linux-rendszerekkel foglalkozhat.

Séta a lemezes egységek körül

Folytassuk a beviteli-kiviteli eszközökkel való ismerkedést! Az előző részben a terminálokról szóltunk, most a lemezes egységek kerülnek sorra.

A lemezes egységeket a merev-, illetve hajlékonylemez meghajtókkal az élen vesszük sorra. Most csupán a lemezegység és az operációs rendszer viszonyáról lesz szó, a fájlrendszerek egy későbbi rész témáját képezik, mivel az a Unix világában sokkal összetettebb, mint ahogy elsőre hinnénk. Talán akad még olyan olvasó, akiben halványan feldereng e sorozat kettővel ezelőtti része, amikor is a beviteli-kiviteli eszközök és az operációs rendszer viszonyáról próbáltunk általános képet felvázolni. Kísérletet tettünk ezek csoportosítására is. Megállapítottuk, hogy ugyan roppant sok, látszólag egymástól teljesen különböző B-K eszköz van forgalomban, az operációs rendszer szemszögéből nézve azonban alapvetően két osztályba sorolhatjuk őket, mégpedig a karakteres és a blokkalapú B-K eszközök csoportjába.

Az előző hónapban egy jellemzően karakteres eszközt vettünk alaposan szemügyre, mégpedig a terminálokat. A lemezek – amelyek mostani értekezésünk tárgyai – azonban kivétel nélkül blokk eszközök, tehát kezelésük teljesen eltérő szemléletet kíván meg az operációs rendszerektől, mint például a terminálok esetében.

De mit is értünk a „blokkeszköz” kifejezés alatt? Az ilyen típusú egység mindig adott méretű adatsomagokkal, azaz blokkokkal dolgozik. Például a merevlemezről nem olvashatunk be csupán egyetlen bájtot, hanem az egész blokkot be kell olvasnunk (amelynek mérete általában 512 bájttól és 32 kilobájttól közé esik) a memóriába, majd ott megkeresni a számunkra érdekes részeket. Ugyanez a helyzet az írás esetében is.

A blokkeszközökre jellemző az is, hogy a blokkok úgy mondhatóan véletlenül elérhetőek. Ez azt jelenti, hogy bármelyik blokk külön címmel bír, és azokat egymástól teljesen függetlenül olvashatjuk, illetve írhatjuk újra.

(Bonyolítsuk a helyzetet azzal, hogy rámutatunk, a való életben mennyire nem használható ez a csoportosítás. Elmélkedjünk el egy kicsit a szalagos meghajtók esetén. Általában ezek is blokkokban tárolják az adatot, tehát blokkeszközök. Ám egy blokk megkereséséhez a meghajtónak az egész szalagot az elejére kell csévélnie (tekernie), majd egyenként végig kell mennie az összes blokkon, amíg el nem éri „úti célját”. Ez borzalmasan lassú művelet, ami már eleve kizárja, hogy valaki véletlenül elérhető eszközként használhassa őket. Arról nem is beszélve, hogy egy meghatározott blokk újrainírása általában nem lehetséges.)

Ez a csoportosítás ugyan nem tökéletes, azonban mindig jó támpontot ad egy operációs rendszer B-K eszközkészletének megtervezéséhez, habár sokféle típusú lemez létezik, amelyeket egy operációs rendszernek tudnia kell kezelni. A Unixokban a ramlemezeket, a hajlékony- és merevlemez-meghajtókat, valamint a CD-ROM és SCSI egységeket általában különálló meghajtó program kezeli. Még mielőtt megnéznénk őket külön-külön, ejtsünk pár szót magáról az eszközkészlet felépítéséről. Itt is az a jól bevált módszer érvényesül, hogy minél kevesebb rész tudjon bármi kézzelfogható magáról az eszközről. Ezért egy jól megtervezett operációs rendszerben a következő kiala-

kítás a legcélszerűbb: legyen egy erősen eszközfüggő rész, ami ugyan nagyon „buta”, mert csak a legalapvetőbb szolgáltatások elvégzésére alkalmas, viszont mindent tud az adott eszköz programozásáról. A felette elhelyezkedő rész elől azonban már rejtve van az eszköz silány világa, csak az eszközfüggő rész segítségével érheti el az adott egységet. Ide azok az eljárások kerülnek, amelyek az összes eszköztípus esetében azonosak. A Linux- és a hozzá hasonló folyamatstruktúrált operációs



rendszerekben az eszközkészlet is külön folyamatok, mélyen a rendszermagban futnak, így számukra egyrészt a gép egyes részei elérhetőek, másrészt közös memóriaterületen osztoznak. Ez utóbbi azért célravezető, mert az eszközfüggetlen részek bármilyen típusú eszköz esetében szinte ugyanazokat az eljárásokat használják, így elég csak egyszer betölteni őket a memóriába. Az eszközfüggő részek egymástól teljesen eltérőek, ám méretük aránylag nem számottevő.

Az eszközfüggő rész tehát meglehetősen „buta”, csupán a legalapvetőbb műveletek elvégzésére képes. Mit is értünk alapvető művelet alatt? Például egy megadott blokk beolvasását, írását. Egy merevlemez-kezelő az állományokról vagy a könyvtárakról semmit sem tud, ami rendjén is van, hiszen erről gondoskodik majd a felsőbb területeken (régiónokban) lakó fájlrendszerkezelő. Az is igaz azonban, hogy a fájlrendszerkezelő számára a lemez felépítése rejtett, tehát csak logikai blokkcímmel képes dolgozni, míg az eszközfüggő rész kizárólag „fizikaiakkal”. Az eszközfüggetlen részek a lemezen található blokkokat lineáris sorozatnak tekintik, ami például a merevlemez esetében nem felel meg a valóságnak, mivel az cilinderekre és pályavonalakra van osztva.

A Unixokban az eszközfüggő rész általában hat utasítást ismer, ezek pedig: OPEN, CLOSE, READ, WRITE, IOCTL, SCATTERED_IO. Az OPEN az adott eszköz rendelkezésre állását ellenőrzi, a CLOSE pedig a még az átmeneti tárbán

lévő adatok azonnali kiírását biztosítja.

Az előző részben már találkozhattunk az átmeneti tár fogalmával. Amikor az eszközfüggő részt megkérjük, hogy írjon ki egy blokkot, lehet, hogy a lemezmeghajtó éppen valami mással van elfoglalva, vagy hatékonysági megfontolásokból (lásd később) először egy ideiglenes tárba, az átmeneti tárba rakja azt. Ezután visszaküld egy üzenetet, hogy a feladatot sikerült végrehajtania, habár maga a kiírás még nem történt meg, de majd arra is sort kerít unalmas perceiben (ezért is veszélyesek a váratlan áramszünetek, vagy amikor a rendszer rendeltetés-szerű leállítása helyett egyszerűen csak „kirúgjuk” gépünket az áramellátásból, mert azok az adatok, amelyek csak az átmeneti tárban vannak, még nem íródtak ki, és a gép áram nélkül maradásakor örökre elvesznek).

Az IOCTL utasítások eredetileg az eszköz valamely művelet-jellemzőjének megváltoztatására hívatottak. A leggyakrabban emlegetett példa a soros kapu (bár az nem blokkeszköz) átviteli sebességének és párosságának (ami a hibaellenőrzésben játszik szerepet) a megváltoztatása. Az IOCTL műveletekre blokkeszközök esetén viszonylag ritkán van szükség. A merevlemezeken esetében például a lemezfelosztási (partition table) tábla megváltoztatása tekinthető ilyen műveletnek.

A SCATTERED_IO lehetőséget biztosít arra, hogy egyszerre több blokk írását, illetve olvasását is elvégezhessük (ellentétben a READ-del és a WRITE-tal, ahol egy utasítással csak egyetlen blokkhoz férhetünk hozzá). Ennek értelme szintén a hatékonyság növelésében rejlik, mivel a lemezkezelőnek lehetősége van dönteni, hogy a kért blokkokat például milyen sorrendben olvassa be.

Az eszközfüggetlen rész is közösleges folyamat (noha a rendszer szintjén fut), de csak az eszközfüggő részekkel tart kapcsolatot, az eszközzel közvetlenül nem. Egyik legfontosabb feladata a fájlrendszerrel érkező utasítások végrehajtása, illetve olyan formára hozása, amelyet a nála alacsonyabb szinten lévő részek is megértenek. Ezek után térjünk is rá magukra a lemezekre. Először az úgynevezett ramlemezről említenénk, ami nem más, mint amikor a memóriának egy szeletét elkülönítjük, és úgy használjuk, mintha hagyományos lemez lenne. Erre a szolgáltatásra leginkább a merevlemez nélküli számítógépek-nél, illetve a rendszer telepítésekor lehet szükségünk.

A ramlemez-meghajtó működése nem nevezhető bonyolultnak. Két utasítást ismer: a blokk írását és a blokk olvasását. Mindkét esetben annyi csupán a dolga, hogy kiszámolja az adott blokk helyét a fizikai memóriában (a ramlemez kezdeti címéhez hozzáadja azt), majd elvégzi az adatok mozgását. A Unixban a ramlemezek négy fajtája lehetséges fel. Az első a */dev/ram* eszközfájlhoz rendelt. Ez tulajdonképpen a hagyományos ramlemez, aminek a méretét és kezdőpontját szabadon meghatározhatjuk. A */dev/null* eszköz is biztosan ismerős lehet a tapasztaltabb felhasználók számára, de azt kevesen gondolnák, hogy ennek a kezelése is a ramlemez-meghajtóra hárul. Természetesen az is igaz, hogy nincs túl sok gond vele, mivel az ide irányított adatokat egyszerűen csak el kell dobni. Ez az eszköz akkor tehet jó szolgálatokat, amikor például egy parancsnak nem vagyunk kíváncsiak a végeredményére. Ilyenkor annyi a feladatunk, hogy a kimenetét a */dev/null*-ba kell irányítanunk.

Van még két, a ramlemezekkel szorosan összefüggő eszköz, a */dev/mem* és a */dev/kmem*. Ezek használatáról a felhasználói szinten dolgozó alkalmazások nem is álmodhatnak, mivel ezek magának a rendszer memóriájának a közvetlen elérését teszik lehetővé. Az első az egész fizikai memóriához való hozzáférést teszi lehetővé (a 0. bajttól kezdve, ahol az ütemező

tárgyalásakor bemutatott megszakításvektor-tábla kezdődik). A */dev/kmem* nullpontja a rendszer mag adatmemóriájának kezdete. Érdekes megjegyeznünk, hogy a */dev/mem* lefedi a */dev/kmem*-et, azaz a */dev/mem*-en keresztül is elérhetjük a rendszer mag adatmemóriáját, feltéve, ha pontosan tudjuk, hol is kezdődik.

Ikón	Eszköz	Típus	Méret	Csatlakoztatási	Szabad	Foglalt %	Kihasznátság
	/dev/cdrom	iso9660	N/A	/mnt/cdrom	0 B	N/A	
	/dev/fd0	auto	N/A	/mnt/floppy	0 B	N/A	
	/dev/hda1	vfat	2,0 GB	/mnt/c	286,7	86,0%	
	/dev/hda5	vfat	3,9 GB	/mnt/d	1,4	44,5%	
	/dev/hda6	vfat	3,9 GB	/mnt/e	1,4	84,8%	
	/dev/hda7	vfat	3,9 GB	/mnt/f	1,5	62,1%	
	/dev/hda9	?	5,1 GB	/	1,1	77,7%	
	LABEL=/ ext3	N/A	N/A	/	0 B	N/A	

A */dev/ram*, a */dev/mem* és a */dev/kmem* eszközközkezelői meg- egyeznek, csak míg az első egy, a felhasználó által meghatá- rozható memóriatartományra vonatkozik, addig a másik kettő magára a fizikai memóriára. Természetesen az utóbbihoz a közösleges felhasználók nem férhetnek hozzá, mivel ezzel könnyedén kárt lehet okozni, de az ügyesebbek fel is törhetik a rendszert – leginkább a különböző hibakereső programok számára tehet jó szolgálatokat.

De térjünk is rá a valódi fizikai lemezekre, azaz a merev- és hajlékonylemez-egységekre. Még mielőtt mélyebben belemer- rülnénk az operációs rendszer lemezkezelésének rejtelmibe, nem árt, ha szólunk pár szót az eszközök felépítéséről is. Maga a lemezegység cilinderekbe szervezett, amelyek pályavonalakból állnak, a pályavonalak pedig szektorokból. Blokk- eszközközről lévén szó, minden szektor ugyanannyi bajttól áll, és a pályavonalankénti szektorok száma is megegyezik. Ez elgon- dolkodható, mivel a lemez peremén lévő pályavonalak nyilván hosszabbak, mint a bentebb lévőek, tehát több szektor is elfér rajtuk. Az IDE-eszközök esetében bizony több szektor található a külsőbb területeken, mint a belsőkben, ám erről az operációs rendszer nem tud semmit, mivel az előbbi kezeléséről maga az IDE-vezérlő gondoskodik.

A ramlemez-meghajtóval szemben a valódi lemezeknél saj- nos számolnunk kell azzal is, hogy egy művelet elvégzése bizony időbe telik. Míg a memóriában való írás, illetve az onnan történő olvasás gyakorlatilag azonnal, egy órajel alatt végbe- megy, addig a lemezmeghajtó esetében a fejlet először a meg- felelő cylinder fölé kell „állítani” (positioning), majd a lemezt úgy kell forgatnunk, hogy a megfelelő szektorok kerüljenek a fej alá. A tényleges beolvasás, illetve írás csak ez után kezdődhet meg.

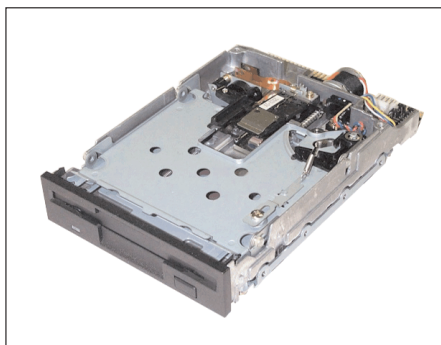
A lemezműveletek tehát borzalmasan lassúak. Szerencsére nem maga a tényleges írás, illetve olvasás veszi ez az idő nagy részét, hanem a fej célba állítása. Ezért ha az operációs rend- szer arra törekszik, hogy a beérkező lemezműveletek végre- hajtásánál egyfajta sorrendet állítson fel közöttük, mégpedig úgy, hogy a fejlet lehető legkisebb úton kelljen mozgatni, akkor lényeges hatékonyságbeli növekedést érhetünk el. Ennek fényében nézzük meg, miként is működhet egy valamirevaló operációs rendszer lemezkezelő programja. Egy lemezműveletre való felkérés bármikor érkezhet, akkor is, amikor a merevlemez épp javában dolgozik az előző művelet- ten. Ezért először gondoskodnia kell a kérések tárolásáról. Ezt egy egyszerű láncolt lista formában teszi meg, vagyis az

összes új elvégzendő műveletet hozzáfűzi a lista végéhez, az elvégzettek pedig kitérli onnan.

A lemezkezelés hatékonyságának kulcsa az, hogy a lemezkezelő milyen elv alapján hajtja végre a listában szereplő utasításokat. Könnyű belátni, hogy nem lenne túlzottan célravezető, ha a kéréseken beérkezésük sorrendje alapján kezdene el dolgozni, mivel lehet, hogy egymástól teljesen különböző cilindreken lévő szektorokkal kéne foglalkoznia, ami ebben az esetben azzal járna, hogy a fejnek kilométereket kéne odavissza rohagálnia.

Ezért e nehézség megoldására az egyik legismertebb módszer az úgynevezett liftes algoritmus, ami – mint neve is utal rá – sok szempontból hasonlít a magasabb épületekben fellelhető felvonószervezetek programjára. Csak itt a lift az olvasófej, az emeletek a cilinderek, a beérkező kérések pedig az egy-egy emeletről érkező hívásoknak felelnek meg.

A liftek általában a következő algoritmus követik: miközben fel-le száguldoznak, a kérések a felhasználóktól (illetve a házlakóitól) véletlenszerűen érkeznek a különböző emeletekről.



A lift azonban mindig csak egy irányba mozog, amíg ugyanabba az irányba van kérés. Az irányváltogatás csak akkor következik be, amikor az eredeti irányban lévő kérések már elfogytak. Sebességnövekedést érhetünk

el úgynevezett pályavonalankénti raktározással is. Mint már említettük, a legnagyobb időt a fej cilinderek közötti mozgatása veszi el, ehhez képest a szektorok beolvasása, illetve átmozgatása a memóriába szinte pillanatok műve. Sőt mi több, amikor a lemezt elforgatjuk a fej alatt, hogy a megfelelő szektort kiolvashassuk, a fejnek alkalma nyílik a pályavonalon található többi szektor beolvasására is. Ezért az a bevált módszer, hogy az adott pályavonalon található összes szektort beolvassuk. (A jelenlegi vezérlők többsége ezt már saját maga is megcsinálja, a beolvasott adatokat pedig a saját belső átmeneti tárába helyezik, ahonnan majd később átkerülhetnek a memóriába). Az utolsó dolog, amire a lemezkezelőnek figyelnie kell, a hibakezelés. A ramlemezek esetében kizárólag programozási hibáról beszélhetünk, például egy nem létező blokkot szeretnénk elérni. Ebben az esetben sok mindent nem lehet tenni, ki kell javítani a lemezkezelő programot. A valódi lemezeknél azonban már külső tényezők is közrejátszhatnak, ha például porszemcse kerül a fej és a lemez közé, vagy csütörtököt mond a vezérlőelektronika.

Egy hiba észlelésénél mindig el kell döntenie, mit tegyünk a kialakult helyzettel. Vannak olyan hibák, amelyek csak valamely ideiglenes ok miatt állnak fent. Ilyen lehet a lemezfelületre került porszemcse. Ilyenkor általában az „idő majd mindent elrendez” elve érvényesül, azaz az adott művelet egy későbbi megismétlésével a hiba megszűnik.

De nem mindig. Ha az adott blokkon fizikai sérülés keletkezett, hiába olvassuk a végtelenségig, nem jutunk előbbre. Ilyenkor a „bad sector” esete forog fenn, tehát tartózkodnunk kell a további használatáról. Régebbi lemezek esetén ez a feladat a lemezkezelőre hárult. Neki kellett számon tartania azt egy kü-

lönleges állományban, hogy mely blokkok hibásak, és azokat foglaltnak tekintette.

A mai merevlemezek vezérlői már nagyon okosak, saját maguk tartják nyilván a hibás blokkokat. A lemezeken található néhány úgynevezett tartalék pályavonal. Ha a vezérlő talál egy hibás blokkot, akkor az egész pályavonalat egy tartalékkal helyettesíti. Tehát az összes, a hibás szektort tartalmazó pályavonalra vonatkozó kérés valójában a kijelölt tartalék pályavonalon fog végbemenni. Mivel erről maga a lemezegység találaton elektronika gondoskodik, az operációs rendszer erről semmit sem tud, sőt, még azt sem, hogy a lemez valahol hibás blokkot tartalmaz. Ez kényelmi szempontból valóban hasznos, ám könnyen felborulhat például a liftes algoritmus által nyújtott sebességnövekedés (mondjuk abban az esetben, amikor a hibás blokk jó messze esik a tartalék pályavonalakat tartalmazó cilinderektől).

Időnként úgynevezett keresési hibák is előfordulnak, ilyenkor a fej nem a megfelelő cylinder fölé kerül. A fejet mozgató kis motort is a vezérlő irányítja, ami pontosan tudja, hogy a fej épp hol tartózkodik. Amikor utasítást kap egy blokk kiolvasására, áramimpulzusokat küld a motornak, hogy lökdösse arrébb a fejet ennyi cylinderrel és ebben az irányban. A merevlemezek esetében a vezérlő ezt észlelni tudja, és kijavítja a hibát, tehát a merevlemez-kezelő megint mentesül a munkától. A hajlékonylemez-kezelőnek rosszabb sors jutott: ő nem számíthat ilyen jellegű segítségre a hajlékonylemez-vezérlőtől.

Igazából a hajlékonylemez vezérlője sokkal fejletlenebb a merevlemezekenél, ami azzal a furcsa következménnyel jár, hogy hiába egyszerűbb felépítésű a hajlékonylemez, a kezelőprogramja sokkal bonyolultabb, mint a merevlemezé. Ha keresési hiba lép fel, akkor ahelyett, hogy a vezérlő önállóan megoldaná, a lemezkezelőnek fel kell szólítania a lemez meghajtót, hogy állítsa be újra a fejet, azaz tolja oldalra, amíg csak bírja, és a pillanatnyi cilindert nyilvántartó változóját állítsa 0-ra. Ezután ismét kísérletet lehet tenni a szükséges cylinder megkeresésére. Akadnak más olyan gondok is, amelyek a merevlemezek kezelésénél nem kell számítanunk, de erről inkább a következő részben szólnánk részletesebben.

A legkellemetlenebb hiba, ami előfordulhat, a vezérlőprogram nem megfelelő működése. Tulajdonképpen ez is egy közönséges program, tehát előfordulhatnak benne hibák. Például bizonyos szerencsétlen véletlenek közrejátszásából fakadóan elképzelhető, hogy végtelen ciklusba kerül vagy hasonló hiba lép fel. Ilyen esetben sincs minden veszve, mert megoldható a vezérlőprogram újraindítása, de elképzelhető, hogy ez sem segít. Ekkor jön a már jól ismert magpánik (kernel panic) felirat. Szerencsére ilyesmi nagyon ritkán fordul elő.

A továbbiakban folytatni fogjuk a lemezekkel való ismerkedést, még gyakorlatiasabb megközelítésben. A következő alkalommal kicsit részletesebben bemutatjuk a DMA-t, ejtünk pár szót az LBA-ról, továbbá a lemezkezelők működését elemezzük egy picit. Ha ezzel megvagyunk, egy egészen egyedi bevitelkiviteli eszköz következik, az óra, ami fontosabb szerepet játszik az operációs rendszer életében, mint gondolnánk.

Garzó András

(garzoand@interware.hu) körülbelül három éve foglalkozik Linux- és más Unix-rendszerekkel. Legjobban az operációs rendszerek lelkivilága érdekli, de nyitott egyéniség. Kedvenc étele a palacsinta, és van egy Richard nevű macskája. Minden észrevételt, megjegyzést, levelet szívesen fogad.

Muzsikáló Linux

Sok olyan elvetemült él a földön, aki Linuxát nemcsak lázas munkára, de például kikapcsolódásra is szeretné használni. Sorozatunk elkövetkezendő néhány részében a Linux könnyedebb arcát szeretnénk bemutatni.

E hónapban többek között zenét fogunk hallgatni kedvenc operációs rendszerünk segítségével, méghozzá MP3-akat, mivel ma ez az egyik legelterjedtebb audioformátum. Természetesen szó lesz még másról is, például arról, hogy miként alakíthatjuk át zenei CD-nk hanganyagát MP3-ba. Az MP3 formátum népszerűségét minden bizonnyal annak köszönheti, hogy bármilyen hanganyagot képes kis méretben és viszonylag jó minőségben tárolni. Míg a „hagyományos” WAV formátumban tárolt 74 perc (egy korongnyi) hosszúságú zene körülbelül 640 MB-ot „fogyaszt”, addig mindez MP3-ban mentve 100 MB körül van (amelynek a minősége megközelíti a CD-két). Ma már szinte az összes operációs rendszer alá elburjánzottak az MP3-formátumot is támogató lejátszók, kódolók stb. Magától értetődően kedvenc operációs rendszerünk sem lehet kivétel: a Linux ezen a téren is bőséges kínálatot tud felmutatni.

Nyilván az MP3-hallgatás előfeltételeihez tartozik az is, hogy legyen pár darab (de legalább egy) MP3-unk a merevlemezén. Ezért elsőként hanglemezeinken található hanganyag MP3-ba való átalakításában mélyülünk el egy kicsit. Ez kétféle művelet: először a CD-n lévő hanganyagot kell rögzítenünk a merevlemezre wav-formátumban, ezután egy MP3-tömörítő segítségével az egészet MP3-ba alakítani. Mindkét művelet elvégzéséhez külön alkalmazást használunk.

Az MP3-kódolók közül kettőt emelnénk ki: a LAME-t és a gogót. A LAME-t (Lame Aint an MP3 Encoder) eredetileg egy jól használható, megbízható és gyors kódoló, szinte minden felületre megtalálható. Linux alá csak forrásban érhető el (☞ <http://sourceforge.net/projects/lame/>), ám a fordítás művelete nem ördögösség. A letöltött fájlt először csomagoljuk ki a `tar -xvzf lame-3.92.tar.gz` parancs segítségével. Lépjünk be a létrejött könyvtárba, majd adjuk ki a `./configure ; make` utasítást. Ha

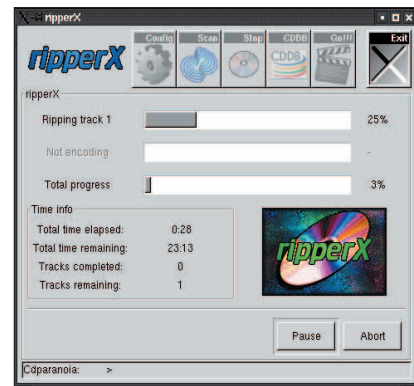
a fordítás közben zajlott, rendszergazdaként jelentkezünk be, és ugyanebből a könyvtárból a `make install` parancs segítségével telepíthetjük a lefordított összetevőket.

A gogo (☞ http://homepage1.nifty.com/herumi/gogo_e.html) egy másik, de a LAME kódjára épülő MP3-tömörítő. Itt nem annyira az átláthatóságon volt a hangsúly (mint a LAME-nél, amelyet eredetileg oktatási célokra kezdtek el fejleszteni), hanem a gyorsaságon és a hatékonyságon. A gogo is több felületen használható, a Linuxon kívül még Windows és OS/2 alatt is élvezhetjük a szolgáltatásait. Ez is csak forrásban érhető el, a fordítás az előzőhöz hasonló módon zajlik, azt leszámítva, hogy itt egy NASM nevű csomagra is szükség van. Ha a terjesztésünk ezt esetleg nem tartalmazná, letölthetjük a

☞ <http://sourceforge.net/projects/nasm/> címről, akár RPM-ben is, így ennek a fordításával már nem kell vesződnünk. A WAV-rögzítők közül leginkább a `cdparanoia`-t tudom ajánlani, mivel ez a legtöbb terjesztésben is fellelhető. Ha a miénkben mégsem találunk, letölthetjük a

☞ <http://www.xiph.org/paranoia/> címről. A `cdparanoia`-val, a LAME-mel és a gogóval nagyon sok grafikus alkalmazás képes együttműködni. Ilyen például a RipperX is (☞ <http://sourceforge.net/projects/ripperx/>), ami azon kívül, hogy leszedi a hanglemezen tárolt zenét, a lame, illetve a gogo könyvtári segítségével az MP3-ba való kódolást is azonnal el tudja végezni. Ennek köszönhetően a felhasználónak csak a megfelelő gombot kell megnyomnia, majd egy rövid kérésünömet tartania – a RipperX mindent magától elvégz. Hasznos szolgáltatása még, hogy képes kapcsolatot tartani az Interneten lévő úgynevezett CDDb adatbázisokkal, ha tehát egy olyan lemezt húzunk le (grab), amelyik szerepel ebben az adatbázisban, nem kell vesződnünk az album és a számok címeinek megadásával.

A RipperX elérhető forrásban, de a binárisban is, RPM-ben. A fordítás itt



sem jelenthet gondot, de ha az utóbbit szedjük le, a telepítéshez rendszergazdaként bejelentkezve csak az `rpm -i csomagnev` utasítást kell kiadnunk. A program a `ripperX` parancssal indul. Elképzeltető, hogy azon nyomban valamiféle hibaüzenet fogad minket, ami tudomásunkra hozza, hogy nem képes hozzáférni a megfelelő eszközhöz. Ilyenkor az a megoldás, hogy rendszergazdaként indítjuk az alkalmazást, de az is megoldás, ha a `cdparanoia`-n bekapcsoljuk SetUID bitet. Ez azt jelenti, hogy a tulajdonos (azaz a rendszergazda) jogosultságaival fog futni, amit rendszergazdaként belépve a `chmod +s /usr/bin/cdparanoia` utasítással tehetünk meg. Most már gond nélkül el kell indulnia a RipperX-nek. A kezelése sem jelenthet gondot, de azért nem árt – még mielőtt munkába fognánk – a következő beállításokat elvégezni: nyomjuk meg a **Config** gombot, majd kapcsoljuk be a **Make mp3 from existing Wav file**-t. A **Wav** fülre kattintva állítsuk be a **Ripper plugin**-nál a `cdparanoia`-t. Az **MP3** fülnél pedig be-

állíthatjuk, hogy melyik tömörítőt használja, illetve milyen minőségűek legyenek az elkészítendő MP3-ak. A többi beállítást érdemes alapértelmezetten hagyni, azonban nem árt ellenőrizni, hogy minden megfelel-e az igényeinknek. Ha mindezzel készen vagyunk, helyezük be a CD-t, és nyomjuk meg a *Scan* gombot – ennek hatására a számok beolvasásra kerülnek. Adjuk meg az előadó

(a rendszermagfordításról és a modulokról részletesen írtunk a *Linuxvilág* 2001. decemberi számában, a 45. oldalon). Az ALSA esetében is szükséges egy alapszintű rendszermag-támogatás, de az OSS-modulokra nincs szükség. Elég tehát a *Sound card support*-ot befördítanunk. Az ALSA-t legegyszerűbben hivatalos webhelyükről, a <http://www.alsa-project.org/> oldalról tölthetjük le.



és az album nevét, majd a számok közötti mezőkben a darabok címeit is. A *Go!* gomb megnyomásával kezdhetjük a munkát. A kész MP3-akat a RipperX által az előadó és az album nevéből létrehozott könyvtár alatt találjuk majd meg. Érdemes pár szót szólni a hangkártya „dalra fakasztásáról” is, ami szintén elengedhetetlen teendő az MP3-ak hallgatásához. A jelenlegi terjesztések telepítői általában felismerik hangkeltő eszközünket, és nincs is szükség további állítgatásokra. Ez azonban sajnos nem mindig ilyen egyszerű.

A Linux-rendszermag a hangkártyák támogatásához általában az úgynevezett OSS-t (Open Sound System) használja, ami más Unix-rendszerek alá is létezik, viszont elég sok galibát okozhat és kevés hangkártyát támogat. Szerencsére van egy testvére: segítségül hívhatjuk az ALSA-t. Ez tulajdonképpen egy olyan meghajtócsomag, ami számos hangszóközöt támogat.

Ha az OSS mellett döntünk, a rendszermag fordításakor tegyük modulba a megfelelő támogatást, majd a fordítás után töltsük be a *morprobe* paranccsal

Innen két dologra lesz szükségünk: az *alsa-driver* és az *alsa-lib* csomagokra. Az *alsa-utils* sok hasznos segédprogramot tartalmaz, tehát ezt is célszerű felrakni. A fordításról itt is nekünk kell gondoskodnunk. Azt javaslom, hogy először a *lib*-et fordítsuk le, utána a meghajtót. A fordítás itt is hasonló módon zajlik, csak a meghajtó esetén a fordítás után még ki kell adnunk a `./snddevices` parancsot. Ha mégis gond adódna, az ALSA leírását, illetve a *Sound-HOWTO*-t (<http://www.linux.org/docs/ldp/howto/Sound-HOWTO/index.html>) érdemes tanulmányozni.

Egy MP3 meghallgatásához csupán egy lejátszóra van szükségünk. Nézzük, milyen lehetőségeink vannak! Először mindenképp az *mpg123*-at illik megemlíteni, amit a konzol szerelmesei nagy örömmel használnak. Kezelőfelülete nincs, csupán csatlókkal kell megadni a lejátszandó fájl nevét, és márt is adhatjuk magunkat a zene élvezetének. A program gyors, hordozható, a legtöbb Unix-rendszer alatt fellelhető, és akár egy 486-oson is képes lejátszani

– igaz, csak monóban vagy csökkentett hangminőségben (22 KHz-ben). Ez a program egyébként szinte az összes terjesztésben megtalálható. Akit nem elégít ki az *mpg123*, annak nyugodt szívvel ajánlhatjuk az *xmms*-t, ami már egy ízig-veéig grafikus alkalmazás. Ez szolgáltatásaiban, kinézetében és kezelésében is sokban hasonlít a windowsos világban rendkívül népszerű *WinAmphoz* (sőt, a régi neve – *X11AMP* – is windowsos rokonára emlékeztet). Ez az alkalmazás is része a legtöbb változatnak.

Érdemes még szót ejtenünk a *FreeAMP*-ről is, amely az utóbbi időben különböző jogi nehézségek miatt névváltoztatásra kényszerült, így lett belőle *Zinf*. Ez az egyébként Windows alatt is elérhető program sokak szerint a Linuxon fellelhető legszebben szóló MP3-lejátszó.

A *Zinf*-et a legegyszerűbben a hivatalos honlapjáról letöltve (<http://www.zinf.org/>) szerezhetjük be. *Red Hat* RPM-ben is elérhető, de ha gondoljuk, mi magunk is lefordíthatjuk. Az RPM-ből való telepítés a szokásos módon zajlik: `rpm -i csomagnev.rpm`. Fontos megjegyezni (bár a honlapon is fel van tüntetve), hogy az alkalmazásnak még egy *MusicBrainz* nevű csomagra is szüksége van, ami szintén ugyanonnan tölthető le (RPM-ben és forrásban egyaránt). A fordításhoz itt is szükségünk van a *NASM*-re.

Végül, de nem utolsó sorban feltétlenül meg kell említenünk az *AlsaPlayer*-t (<http://www.alsaplayer.org/>), ami néhány figyelemre méltó tulajdonsággal bír. Például több szálban fut, így a több processzor nyújtotta előnyöket ki tudja használni. Ezenkívül a lejátszás sebességét is beállíthatjuk (mind a gyorsabb, mind a lassabb irányba), aminek leginkább a DJ-palánták fognak örülni. A legközelebbi alkalommal már játszani is fogunk a pingvinünkkel. Tájékozatlan körökben valamiért az a tévhit kering, hogy a Linux egyáltalán nem alkalmas játékok futtatására. Meg fogjuk mutatni, hogy ez mennyire nincs így! Ezenkívül szót ejtünk még az *OpenGL*-ről is.

Garzó András

(garzoand@interware.hu) körülbelül három éve foglalkozik Linux- és más Unix-rendszerekkel. Legjobban az operációs rendszerek lelkivilága érdekl, de nyitott egyéniség. Kedvenc étele a palacsinta, és van egy Richard nevű macskája. Minden észrevételt, megjegyzést, levelet szívesen fogad.

CD-írás lépésről lépésre

A CD-író üzembe helyezése Linux alatt nem tartozott a legegyszerűbb feladatok közé – egészen mostanáig.

Kezdetben volt a hajlékonylemez. Olcsóság, ugyanakkor kis tárcapacitás jellemezte, továbbá kinosan kellett ügyelni a közlekedési eszköz megválasztására. Egy villamos, trolis, vagy – magasságos égi! – egy metró végzetes hatással lehetett féltve őrzött adatainkra. Nem jelentett megnyugvást az olyan látszatselekvés sem, mint az antisztatikus zacskóba vagy alufóliába történő csomagolás. Azután jött a CD (Compact Disc), és egy csapásra megváltozott minden. Eleinte drága volt a nyersanyag, az író úgyszintén. Mára viszont az árak olyan mélyre zuhantak, hogy a legtakarékosabb megoldást jelenti az adattárolás területén. Előnyeit a végtelenségig sorolhatnánk: könnyű, egyszerűen tárolható, biztonságos, nagy tárcapacitással bír és így tovább. Írásomban lépésről lépésre vezetlek végig egy CD-író üzembe helyezésének megpróbáltatásain egy Debian Woody rendszeren. Egyetlen feltétel, hogy tudj kezelni a csomagkezelőt, legyen az `dselect`, `aptitude` vagy bármi más. Amennyiben más Linux-terjesztés híve vagy, akkor is érdemes elolvasnod az írást, hiszen máshol is hasonló módon kell eljárnod.

1. lépés: A rendszermag újrafordítása

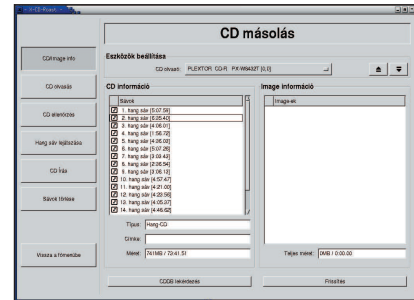
Ne ijedj meg! Ha SCSI típusú CD-író van, és már tudsz vele olvasni, ezt a lépést kihagyhatod. IDE csatolójú írók esetén sem bonyolult, mindössze egy apró módosítást fogunk ejteni a rendszermag beállításában. Először is telepítsd a `kernel-source-2.4.18` nevű csomagot, ha ilyen nincs, a lehető legfrissebbet. A `kernel-package` telepítését is felajánlja, amelynek segítségével az elkészült rendszermagból csomagot készíthetsz, így a saját változatodat mások már sokkal könnyebben telepíthetik. Erre azonban szinte biztos, hogy nem lesz szükséged. Telepítsd viszont a `libncurses5-dev`, illetve a `bzip2` csomagot, ha még nincs fenn. Mindezek után, még mindig rendszergazdaként, add ki az alábbi parancsokat. Először belépsz abba a könyvtárba, ahol a rendszermag forrása található, kicsomagolod, majd a linuxos szokásoknak

hódolva létrehozod egy közvetett hivatkozást a kicsomagolt forrásra `/usr/src/linux` néven:

```
# cd /usr/src
# bzipcat kernel-source-
  2.4.18.tar.bz2 | tar xv
# ln -s kernel-source-2.4.18
  linux
```

```
# cd linux
# cp /boot/config-2.4.18-
  bf2.4 .config
# make menuconfig
```

Érdeemes átmásolni a Debian alapbeállításait, így neked csak finomhangolni kell a rendszermagot, nem pedig a semiből felépíteni – ezt végzi el az utolsó előtti sor. Ha ilyen állományt nem találsz, keress egy a beállítandó rendszermag változatszámához legközelebb állót. Az utolsó parancsot követően egy menüvezérelt felülethez jutsz. Most jön egész eddigi munkánk lényege. Alapjában véve kizárólag SCSI-s íróval lehet CD-t írni. Az IDE-s írók munkára bírásához egy kis trükk szükséges. Nem a megszokott `ide-cd` modult kell használni, hanem az `ide-scsi`-t. Ez egy általános SCSI-protokollt emuláló eszközillesztő. Segítségével az IDE-s CD-író SCSI-eszközként látszik a rendszerben. Azért fordítjuk újra a rendszermagot, mert a Debian Woodyban alpból az `ide-cd` nem modul, hanem bele van vésvé a rendszermagba. Modulát tesz-szük, így meghagyjuk magunknak a választás lehetőségét a hagyományos IDE és a SCSI-t emuláló meghajtó között. Az `ATA/IDE/MFM/RLL support/IDE, ATA and ATAPI Block devices` menüpont alatt található az `Include IDE/ATAPI CDROM support`-ot. Ha a mellette álló, két egymás felé fordított relációs jel között nincs semmi vagy * jel szerepel, a kurzorral lépkedj erre a pontra, és nyomd meg az M betűt. Ugyanezt végezd el a `SCSI emulation support`-nál is. Ezek után az `SCSI support / SCSI CD-ROM support`-ot is add hozzá modulként rendszered leendő magjához. Ha már rendszermagot fordítasz, némi



teljesítménynövekedés elérése érdekében érdemes a processzorodhoz hangolni; ezt a főmenüből a *Processor type and features* menüpontba lépéssel teheted meg. Végezetül lépj ki a programból, és add ki a következő parancsokat:

```
# make dep && make modules &&
  make modules_install &&
  make bzImage
# cp arch/i386/boot/bzImage
  /boot/vmlinuz-2.4.18-cd
```

Az első sor függőségeket ellenőriz, lefordítja a modulokat, telepíti őket és egy lépésben elkészíti a rendszermagot. A második az elkészült remekművet a megfelelő helyre másolja be. A `/etc/lilo.conf`-ot a következő sorokkal kell kiegészítened:

```
image=/boot/vmlinuz-2.4.18-cd
label=Linux-cd
```

Továbbá keresd meg `default=` kezdetű sort, és változtasd meg az alábbi módon:

```
default=Linux-cd
```

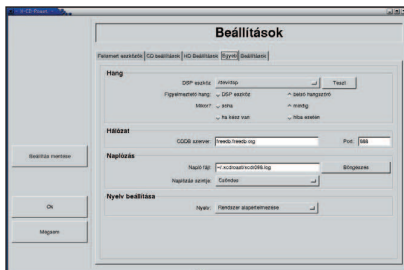
Végül futtasd a LILO-t:

```
# lilo
```

2. lépés: discover, újraindítás, fstab

Telepítsd a `discover`-t. Minden függőségnek tegyél eleget, vagyis az összes felajánlott csomagot telepítsd fel. Egy olyan remek eszközfelismerő programról van szó, ami a felismerés után az eszköz használatba vételéhez a megfelelő modult egyből be is tölti. ATAPI

eszköz esetén az `ide-scsi` modult is mindenképpen megpróbálja betölteni. Ha viszont az `ide-cd` nem modul, hanem a rendszermag része, az `ide-scsi` hiába töltődik be, a rendszer már IDE-eszközként látja a CD-író. Ezért kellett a rendszermagban az `ide-cd`-t modullá tenni. Az első két lépés elvégzése után indítsd



újra a számítógépet. Ha minden jól ment, rendszerbetöltésnél a *Loading Linux-cd...* feliratot kell látnod. Továbbá az `init` elindulása után egy *Detecting hardware...* üzenettel kell találkoznod, majd az `ide-scsi` önműködő betöltése következik. Jelentkezz be, és ellenőrizd, hogy a `/dev/cdrom` hivatkozás mire mutat:

```
$ file /dev/cdrom
/dev/cdrom: symbolic link to
↳ /dev/cdrom0
$ file /dev/cdrom0
/dev/cdrom0: symbolic link to
↳ /dev/scd0
```

Ha nyomozásod eredményeként a fent látható `scd0`-ig jutsz el, már sínen vagy. Ellenkező esetben még most derítsd ki a baj forrását, enélkül ugyanis nem tudsz továbblépni. Ha sikerült, nézzünk egy kicsit körül a rendszerben, hogy lássuk, mi változott. Észreveheted, hogy a `/cdrom` könyvtár mellett létrejött egy `/cdrom0`, és a `/cdrom` sem üres, hanem egy hivatkozás van benne a `/cdrom0-ra/cdrom/cdrom0` néven. Mindez azt jelenti, hogy a befűzési pont megváltozott. Ezek után nem a `/cdrom` alá kell fűzni a CD-ket, hanem a `/cdrom0` alá. Ez azért van így, mert egy SCSI-s rendszernél – mivel ott az IDE két

eszközével szemben egy csatornára akár hét eszközt is rá lehet kötni – számítanak arra, hogy több CD-olvasód, illetve -íród van. Az `ide-scsi` ezt tökéletesen utánozza, ezért minden CD-meghajtónak létrehoz egy befűzési pontot a gyökérkönyvtárban, illetve közvetett hivatkozásokat a `/cdrom`-ban. Ennek megfelelően módosítanod kell a `/etc/fstab` állományt az a sorát, amelyik a CD-olvasóra vonatkozik. A második elemet egyszerűen írd át `/cdrom0-ra`, majd mentsd.

3. lépés: az xcdroast telepítése

Már csak egy feladatunk maradt: kiválasztani egy megbízható íróprogramot, és belekezdeni a CD-írásba. A beállítás már eddig sem volt magától értetődőnek nevezhető, ezért most nem töltöm meg a `cdrecord` milliónyi kapcsolójának magyarázatával a magazint. Ugyanakkor az is igaz, hogy a `cdrecord` a legszélesebb körben használt és legelismerettebb programok közé tartozik. A csúnya parancssori kapcsolók eltakarása végett most egy grafikus előtétet (frontend) fogok bemutatni.

Az `xcdroast` nagyon könnyen használható grafikus CD-író alkalmazás, ami mindazokat a híres-hírhedt parancsokat felhasználja, amelyek a Linux alatti CD-írás őskoráig nyúlnak vissza. Ezek között említhetjük a `cdrecord`-ot, az `mkisofs`, vagy a `cdada2wav`-ot. Telepítés közben a `debconf` megkérdezi, hogy SUID bittel akarod-e telepíteni, vagyis azt szeretné-e, hogy ne csak a rendszergazda használhassa az `xcdrecord`-ot. Én úgy gondolom, hogy minél kevesebbszer kell rendszergazdaként bejelentkezni, annál jobb. Továbbá egy ilyen jellegű programot SUID-dá tenni nem olyan égető biztonsági gond, ami miatt ne tudnék aludni – tehát igennel válaszoltam a kérdésre. Habár a fentiek értelmében egyszerű felhasználóként szeretném használni a programot, a fő beállítóállományt legelőször a rendszergazdának kell elindítania és létrehoznia a fő beállítóállományt. Ezért kivételesen rendszergazdaként kell elindítani az `X-et`, majd az `xcdroast`-ot. Magyar locale esetén a CD-író a *Beállítások* gombra kattintva a *Felismert eszközök* fülön látható. Ha mégis üres a lista, de az `ide-scsi` helyesen betöltődött, akkor sajnos CD-íród azon kevesek közé tartozik, amit a `cdrecord` nem támogat. Létezik ugyan egy Orange Book (Narancsszínű könyv) nevű szabvány. Amennyiben ennek a nemzetközileg is elismert szabványnak az íród megfelel, működnie

kell a `cdrecord`-dal. Ha a CD-író az előbbi értelemben nem szabványos, próbálj valamilyen szélesebb körben használt típust beszerezni. A *CD-beállítások* fül meglehetősen egyértelmű. Arra azonban felhívnom a figyelmet, hogy többek között a CD-író sebességénél nem szabad takarékoskodni. Itt állíthatod be, hogy milyen sebességet nem léphet át az író: ha gyorsabbra állítod, mint amilyen, a CD-t a lehető leggyorsabban megpróbálja elkészíteni. Ha viszont a lassabbat választod, számos erőforrást kihasználatlanul hagyasz. A *HD-beállítások* alatt legalább egy, lemezlenyomatok tárolására alkalmas könyvtárat kell megnevezned, vagyis legalább annyi helytel írhatóknak kell lennie, amennyi egy lenyomathoz elegendő (megközelítőleg 700 MB). Ez azért szükséges, mert ha még van is egy másik olvasód egy CD lemásolásához, az új CD-t nem érdemes közvetlenül az olvasó által szolgáltatott adatokkal írni, mert az olvasó lassúsága miatt „kiapadhat” az író gyorsára, és elrontja a lemezt. Mivel a merevlemez sokkal gyorsabb, ennek elkerülése végett a másolandó CD-ről lenyomatot szoktak készíteni, majd a lenyomatot írják ki. Az *Egyéb* fül alatt a *Hang*, *CDDDB*, a *Naplózás* és a *Nyelvi beállítások* található. Ezenkívül a *Beállítások* és végül *Felhasználók* olyan további panelek, amelyek önmagukért beszélnek. Ha végeztél a *Beállítás mentése* felirátú gombbal, az *Ok*-t kell megnyomnod. Ezt követően újra be kell lépned egyszerű felhasználóként, és már sütheted is a CD-ket! Végül, de nem utolsó sorban mindenki figyelmét fel szeretném hívni arra, hogy a törvénytelen másolatok készítéséből fakadó következményektől a Pingvin sem tud megvédeni. Ezért minden korong másolása előtt érdemes ellenőrizni a kapcsolódó szerzői jogokat taglaló felhasználási szerződést, ennek hiányában a szerzői jogi törvényt.

E cikkre a *Free Document Licence* vonatkozik: <http://www.gnu.hu/fdl.html>



Fülöp Balázs

(xut@freemail.hu)

17 éves, imádja a Túró Rudit, a Debian Linuxot és a teheneket. Az ELTE Radnóti Miklós Gyakorlóiskola tanulója immár ötödik éve. Kedvenc írója Slawomir Mroček. Leginkább a számítógépes hálózatok biztonsága érdekli.

A GNU Fordítógyűjtemény (2. rész)

Ebben a részben a GCC kapcsolóival és használatuk rejtelseivel ismerkedhetünk meg.

A mennyiben már sok függvényt írtunk meg, és azok eléggé általánosak ahhoz, hogy más programokban is felhasználjuk őket, célszerű könyvtárakat kialakítani belőlük. A programfejlesztésben a könyvtár (library) szó újra felhasználható, lefordított programrészeket jelent, amelyeket egy állományba gyűjtünk össze.

Gyakorlásképpen készítsünk könyvtárat sorozatunk előző részében használt *quaternion.c* forrásfájlból, ami egyetlenegy saját függvényt tartalmaz, a `printQuaternion()` nevűt, ez hívja meg a standard C könyvtár `printf()` függvényét. Első lépésben az ismert módon tárgykódot kell készítenünk a forráskódból, majd azt az `ar` program segítségével archív állománnyá kell alakítanunk:

```
> gcc -c quaternion.c -o liblinux.o
> ar rcs liblinux.a liblinux.o
```

Az `ar` program utáni `rcs` kapcsolók jelentése a következő:

- `r`: beszurja a felsorolt fajlokat az archiv allomanyba, torolve az ott mar jelenlevő, azonos nevűeket (az `r` rövidítés a `replace`, azaz helyettesít szóból származik).
- `c`: létrehozza az archív fájlt, ha még nem létezik (a `c` a `create` szó első betűjére utal).
- `s`: tárgykód fájlindexet fűz az archív állományhoz vagy frissíti már létező fájl esetében.

A GNU `ar` program utáni első fájlnev a létrehozandó archív neve lesz, ezután kell felsorolni azokat a fajlokat, amelyeket az archívban akarunk tárolni. Az archív fajlok szerkezete olyan, hogy több tárgykódfájlt képesek összefogni, miközben az egyes fajlok és azok tartalma számunkra elérhető marad. Az archívban lévő fajlokat tagoknak (members) hívjuk, és az `ar` programmal módosíthatjuk őket. Az archív könyvtárfájlok kiterjesztése alapértelmezetten `.a`.

Ne felejtjük el, hogy új könyvtárak létrehozásakor fejállományokat is kell készítenünk, és azokat elérhetővé kell tennünk a fordító számára!

Ha most a *liblinuxvilag.a* könyvtárat bemásoljuk egy munkamappába a a *41-es CD Magazin/GCC/lib/static/01* mappájában található fájlokkal együtt, akkor lefordíthatóvá válik picurka, de nagy tudású programunk:

```
> gcc linuxvilag.c -o linuxvilag -I. -L.
↳ -llinux
```

A `-L` kapcsoló a könyvtár helyét mutatja, ami a *01* példamappában a munkamappa, ezt egy pont jelzi. Mivel egy mappában több más programkönyvtár is lehet, a GCC fordítónak meg kell mondanunk, hogy programunkat melyikkel akarjuk összekapcsolni. Ehhez a `-l` kapcsolót használjuk, ami után a használni kívánt könyvtár neve következik.

Az előző CD-melléklet *Magazin/GCC/lib/static/02* mappájában ugyanezek a fajlok vannak elhelyezve, de itt két almappát is kialakítottam, ahová a fejállományt és a `linuxvilag` könyvtárat tettem. Ebből az okból kifolyólag most a következő fordítási utasítást kell kiadnunk:

```
> gcc linuxvilag.c -o linuxvilag -I./header
↳ -L./library -llinuxvilag
```

A fordítás után a `linuxvilag` program futtatható lesz. Míg a *02* almappában a pillanatnyi munkakönyvtárhoz viszonyított elérési utat adtam meg, a *03* almappában teljesen relatív útmegadást alkalmaztam:

```
> gcc linuxvilag.c -o linuxvilag -Iheader
↳ -Llibrary -llinux
```

Természetesen abszolút elérési út megadása is lehetséges lett volna:

```
> gcc linuxvilag.c -o linuxvilag
↳ -I/home/linuxvilag/gcc/lib/static/04/header
↳ -L/home/linuxvilag/gcc/lib/static/04/library
↳ -llinux
```

Munkánk végeztével elégedetten nyugtázzhatjuk, hogy a `linuxvilag` nevű statikus könyvtárral összekapcsolt programunk lefut.

Osztott könyvtárak készítése

A statikus könyvtárak (static libraries) újra felhasználhatók, de minden egyes újonnan megírt és lefordított programhoz ismételtlen hozzá kell kapcsolnunk őket. A dinamikus vagy más néven osztott könyvtárak (shared libraries) viszont csak egyszer kerülnek be a memóriába, és ott egyszerre több program is el tudja érni őket.

A statikus könyvtárakat fordításidőben kapcsoljuk a programokhoz, az osztott könyvtárakat pedig futásidőben.

Ha osztott könyvtárat akarunk készíteni, akkor először a forráskódból kell tárgykódot készítenünk az ismert módon, de most használnunk kell a `-fPIC` kapcsolót, hogy a fordító helyzettől független kódot (Position Independent Code) hozzon létre, ami azt jelenti, hogy a behívott programrészlet bármilyen memóriacímre képes betöltődni:

```
> gcc -fPIC -c quaternion.c -o liblinux.o
```

A Unix-hagyományok szerint a könyvtárak nevei a *lib* rövidítéssel kezdődnek, ezért nem kell teljes névvel hivatkozni rájuk. Például a fenti *liblinux.a* statikus könyvtárra elegendő volt a `-llinux` névvel hivatkozni. A megosztott könyvtárak nevei szintén a `library` szóból származó `lib` szócskával kezdődnek, de a név után egy pont és a `so` kiterjesztés következik. Ezt a formát `so`-névnek (soname) hívják, például a *vga* könyvtár esetében ez *libvga.so*. A `so` név után a fő- és az alváltozat-szám következik, majd a kiadás azonosítója.

Miután létrehoztuk a *liblinux.o* tárgykódfájlt, a következő paranccsal el kell készítenünk a megosztott könyvtárat:

```
> gcc -shared -Wl,-soname,liblinux.so
↳ -o liblinux.so.1.0.0 liblinux.o -lc
```

A következő kapcsolókat láthatjuk a parancs után:

- `-shared`: olyan tárgykódot hoz létre, ami később futásidőben is hozzákapcsolható más programokhoz.
- `-Wl`: az összekapcsoló program (linker) számára ad át kapcsolókat. A vesszők teszik lehetővé, hogy egyszerre többet is megadjunk.
- `-soname`: a fentebb említett `so` névre utal.
- `-liblinux.so`: az adott könyvtár névleges `so` neve.
- `-liblinux.so.1.0.0`: a teljes `so` név a fő- és alváltozatszámmal, valamint a kiadást jelölő számmal együttesen.

- `-liblinux.o`: az a tárgykódfájl, amit korábban a `-fPIC` kapcsolóval fordítottunk le.
- `-lc`: a gépünkben lévő C könyvtárhoz kapcsolódunk, hogy lehetővé tegyük egy olyan C-változat használatát, ami miatt a program futása megbízhatatlanná válna. Ha ezt a kapcsolót használjuk, egy inkompatibilis C könyvtár esetén le sem fogjuk tudni fordítani a programot (lásd még az `info ldconfig` sűgőoldalt).

Ezután létre kell hoznunk a közvetett hivatkozásokat:

```
ln -s liblinux.so.1.0.0 liblinuxvilag.so.1
```

```
ln -s liblinux.so.1.0.0 liblinuxvilag.so
```

Ezzel el is készítettünk megosztott könyvtárunkat, amire elegendő a `linuxvilag` szóval hivatkozni. Az utolsó előtti lépésben lefordítjuk programunkat:

```
> gcc linux.c -o linuxvilag -lheader
```

```
↳ -Llibrary -llinuxvilag
```

Ha nem vagyunk biztosak abban, hogy programunk valóban egy osztott könyvtárhoz kapcsolódott-e, használjuk az `ldd` programot, például:

```
> ldd static
```

Az `ldd` a Library Dependency Display szavak rövidítése, amit könyvtárfüggőség-kijelzőként fordíthatnánk magyarra. Ez az eszköz megmutatja nekünk, hogy egy adott program milyen megosztott könyvtárakkal van összekapcsolva.

Ha a `static` nevű, futtatható állományhoz (lásd az *executable* almappát az előző CD-mellékleten) a statikus `linuxvilag` könyvtárat kapcsoltuk, az `ldd` program a következőt írja ki a standard kimenetre:

```
libc.so.6 => /lib/libc.so.6 (0x4002c000)
```

```
/lib/ld-linux.so.2 => /lib/ld-linux.so.2
```

```
(0x40000000)
```

Ez azt jelenti, hogy programunk a standard C könyvtár mellett még az `ld.so` betöltőprogramot is használja.

A GCC kapcsolói

Esély sincs arra, hogy ebben a rövid írásban felsoroljam a GCC összes kapcsolóját, de néhány fontosabbat megemlítenék. Ezek a kód egyszerűsítésével, a hibakereséssel és a GCC üzeneteivel kapcsolatosak. Aki részletesebb ismertetésre vágyik, az nézze meg az `info gcc` sűgőlapokat.

• Egyszerűsítés

Az optimalizálás növeli a forráskód lefordításához szükséges időt.

-O – a fordító megpróbálja csökkenteni a tárgykód méretét és gyorsítani a lefordítandó programok végrehajtási idejét.

-O0 – nincs semmiféle egyszerűsítés.

-O1 – ugyanaz, mint a -O kapcsoló.

-O2 – magában foglalja az összes O1 szintű egyszerűsítést, amit még olyan utasításokkal egészít ki, amik növelik a processzor munkájának a hatékonyságát. Az O2 szintű egyszerűsítés általában kellő hatékonyságú.

-O3 – magában foglalja az összes O2 szintű egyszerűsítést, de ciklusbontással további gyorsítást próbál elérni, azaz ha egy ciklusnak a lépésszáma már a fordítás idején ismert, akkor a fordító megszünteti a ciklust, és minden egyes lépést egymás után következő programrészé alakít át, megnövelve így a program méretét, de felgyorsítva annak futását.

-Os – a program méretének csökkentése az elsődleges cél. Magába foglal minden olyan O2 szintű egyszerűsítést, ami nem növeli nagymértékben a program méretét.

• Hibakeresés

A hibakeresést megkönnyítő adatok kerülnek bele a tárgy-

kódállományokba, ami lehetővé teszi a nyomkövetést, de nagymértékben megnöveli a bináris állományok méretét. Ez a növekedés akár nagyságrendi is lehet. Logikusnak tűnhet, hogy könyvtárainkat hibakeresés nélkül fordítsuk le, hiszen így a méretük kisebb lesz. De ha arra gondolunk, hogy más fejlesztők is használni fogják azokat, már nem feltétlenül olyan jó ötlet a hibakeresés lehetetlenné tétele.

Itt is több szint létezik:

-g – az alapértelmezett szint.

-g1 – a legalacsonyabb szint. A legkevesebb hibakereséshez felhasználható adatot tartalmazza.

-g2 – megegyezik a -g kapcsolóval, tehát ez az alapértelmezett szint. Szimbólumtáblákat épít bele a tárgykódállományba, bent hagyja a sorok számozását és további adatot kapunk a helyi és külső változókról.

-g3 – tartalmazza a g2 szint minden hibakeresést segítő adatát a makró-meghatározásokkal együtt.

-ggdb – olyan adatokat épít be a tárgykódállományba, amiket a GNU hibakereső is tud használni. Ez a program a GDB, amiről egy későbbi írásomban még részletesen szólni kívánok. Ennek is három szintje van, például a -ggdb3 alakot is beírhatjuk.

Általában nem célszerű a hibakereső adatok beépítése az egyszerűsítéssel együtt, annak ellenére, hogy a GCC ezt megengedi. Nem várt hibák léphetnek fel, mivel az egyszerűsítés következtében eltűnhetnek olyan változók, amikről biztosan tudjuk, hogy előzőleg megadtuk őket; programunk vezérlése futás közben olyan helyre ugorhat, amire egyáltalán nem számítnak; vannak olyan függvények, amelyekre várakozásainkkal ellentétben nem kerül rá a vezérlés, mert azok eredményét állandónak tekinti az egyszerűsítő program; más függvények pedig egészen máshová kerülnek, mint ahová mi eredetileg megírtuk őket, mert kikerültek a ciklusokból. Általános szabálynak tekinthetjük, hogy elsőként mindig a hibákat keressük meg, és csak azután egyszerűsítjük a tárgykódállományt, miután úgy gondoljuk, hogy az hibátlan.

• Hibák és üzenetek

A GCC megkülönbözteti az olyan hibákat, amelyek valóban végzetes hibák és lefordíthatatlanná teszik a programunkat, és az olyan hibákat, amelyek valamilyen programozói odafigyelésből erednek, mint például egy változó megadása, de fel nem használása. Ezzel a téma körrel kapcsolatban két kapcsolót említek meg, amelyeknek a feladatát a -W után következő szó határozza meg:

-Wall – minden üzenetet kiír fordítás közben.

-Werror – minden üzenetet hibaként értelmez.

A GCC Fordítógyűjtemény

A fenti példákban C-forráskódot alakítottunk futtatható állománnyá a GCC segítségével. Az utolsó példában azonban már egy C++-fájlt fordítottunk le a következő utasítással (lásd az előző CD-melléklet *gcc/code/example09* mappáját):

```
gcc quaternion.cc linuxvilag.cc -o linuxvilag
```

A gcc minden nehézség nélkül lefordítja a C++, Java, Fortran, Pascal programokat.



Szaló István

(ratiosoft@freemail.hu) tanár, immár több mint másfél évtizede foglalkozik programozással, de csak a Java és a Linux megismerése után tudta meg, hogy mi is az igazi programozás. Több írása megjelent már a hazai számítástechnikai lapokban.

Netbankozzunk Linux alól!

Senki sem szeret sorban állni – aki mégis, jelentkezzen!
Nahát, miattatok van még mindig sorakozó a bankban!

A kezdő operátor is hamar megtanulja, hogyan lehet egyetlen mozdulattal a nyomtatási sorban a saját nyomtatnivalóját a többieké előtt érvényesíteni. Ismerek egy volt operátort, aki egyenesen sportot űzött a felhasználók bosszantásából, és egy nagy hivatalban egyenesen ilyen sorokat hozott létre. Egész egyszerűen ha egy felhasználó nyomtatott valamit egy adott nyomtatóra, azt egy egyszerű átírányítással átvezényelte egy másik nyomtatóra, és várt. Az eredmény mindig fejtelenség lett. Azt sajnos nem tudom, hogy a magyar bankokban sok ilyen vicces kedvű operátor dolgozik-e, az mindenesetre biztos, hogy sor szinte minden bankban létezik. Azt is mondhatnánk, hogy a sor a bankok állandó együtthatója. A hétfői nap a legválságosabb szinte mindig, bár a többi napon sem bonthatunk pezsgőt, főleg ha megnézzük átlagos nyitva tartásukat: munkaidő 9–12-ig van, majd rövid ebéd-szünet követi 13.30-ig, azután mindent beleadnak egészen a késő éji 16 óráig. Egy átlagos embernek tehát más megoldást kell keresnie. A választható megoldások között már jó régen feltűnt a telebankos ügyintézés lehetősége is, amelyet szinte minden hazai bank ingyen és bérmentve kínál, sőt arra is rájöttek, hogyha a kedves ügyfél nem a bankban zsidbad, hanem telefonon intézkedik, az a banknak is jó. Ezért a telefonos ügyintézés díjai általában olcsóbbak, mintha egy fiókban tevékenykednénk. Én magam is számtalanszor használtam ezt a lehetőséget, de rengeteg gyenge pontot találtam (azaz rést a pajzson).

1. Mint mindenki, én is esténként intéztem banki ügyeimet, ezért rengeteget kellett várni az ügykezelő jelentkezésére – hiába, a sor itt is csak sor.
2. Egy átlagos felhasználó semmilyen úton nem tudja elérni, hogy a kapcsolattartás titkosított adatsatornán történjen a bank és közte. A legtöbbben a munkahelyükön telebankoznak, ami még veszélyesebb, mert a legtöbb telefonközpont rögzíti a beütött számok listáját. Egy olyan banknál, ami

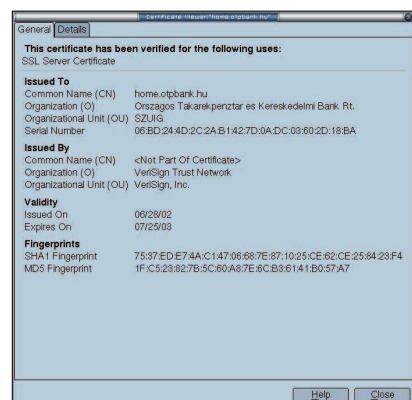
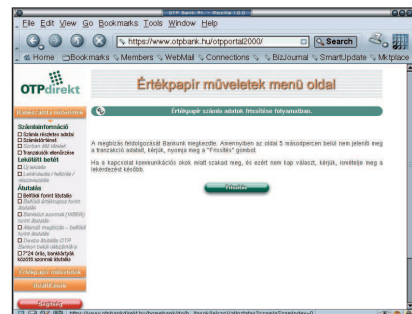
minden alkalommal megköveteli a teljes jelszó- és a felhasználói azonosító bepötyögését, végzetes is lehet számunkra, hiszen a telefonközpontot üzemeltető személy pontosan tisztában lesz az adatainkkal. A legtöbb banknál tört részű PIN-kód (személyi azonosító) bekérésével próbálkoznak. Ez jó ötlet, ha a PIN legalább tízjegyű és egyszerre legkevesebb négyet bekérünk, azonban a legtöbb bank négyjegyű telefonos PIN-t használ, és többnyire egyszerre kell beütünk őket. Kivételesen az OTP Bank, ahol ugyanakkor tőlem már egy éve szinte mindig ugyanazt a három számot kéri be a hétjegyű kódból.

A sok-sok hátrány mellett van egy nagyon-nagy előnye a telefonos ügyintézésnek: a beszélgetést kivétel nélkül mindig felveszik. Ha galiba adódna, jó esetben bizonyítani tudjuk az igazunkat. Nézzük, hogyan is kerül a képbe a Linux!
Szinte mindegyik bank kínál már internetes felületen elérhető ügyintézési megoldást. Sajnos nem volt lehetőségem végigpróbálni az összes ismertebb bank szolgáltatását, de az általam két legnépszerűbbnek ítélt cég – a CIB Bank és az OTP Bank – szolgáltatásainak utánajártam.
Még mindig akad olyan pénzintézet, amelyik egyáltalán nem veszi tekintetbe, hogy az ablakos operációs rendszeren kívül még számos szabad és fizetős operációs rendszer is létezik, és hogy az előbbieket csak *exe*-ként futtatható DOS-os vagy ablakos bankolást tesznek lehetővé. A fenti bankok viszont remek lehetőséget biztosítanak az internetes számlaműveletek elvégzésére. Mivel nincs új a nap alatt, mindkét intézmény megközelítőleg ugyanazokat a lehetőségeket kínálja.

OTP Bank

Ez idáig kőkeményen ragaszkodtak egy javás programhoz, amely Linux alatt változatos sikerrel futott vagy fagyott be (október végén elindítottak egy nem Java alapú böngészőből használható

változatot is). A befagyás főleg akkor okozhat szívdobogást, amikor az APEH-nek utalunk, és nem tudjuk eldönteni, hogy az átutalás elment-e vagy sem. Természetesen lekérdezésre lehetőség van, de ha hétköznapokon 17 és 18 óra



között utal át az ember, valamikor 17:30 tájkán egyszer csak eljön a rendszer karbantartásának ideje (soha nem ugyanakkor), és előfordulhat, hogy bár az átutalás átment, a Java-alkalmazás kifagy, és mire újra le tudná kérdezni az ember, a bank már zárva van (virtuálisan is). Így csak a következő munka-

Rendszermag-bővölési útmutató

Greg elmagyarázza, hogyan kell használni a BitKeepert vagy a patch és diff programokat együtt.

A Linux-magfejlesztés számos vonatkozásban eltér a hagyományos programfejlesztési folyamatoktól. Amikor a rendszermag adott részletén dolgozunk, több területre is figyelmet kell fordítanunk:

1. Figyelnünk kell arra, hogyan fogadják a rendszermag más részei az általunk az adott területen éppen végzett módosításokat.
2. Ne felejtjük el folyamatosan érvényesíteni a változtatásokat a gyors ütemű rendszermagfejlesztés célterületén.
3. Az általunk vagy a mások által végzett módosítások nyomán fellépő összeférhetlenséget fel kell oldani.
4. Képesnek kell lenni rá, hogy saját módosításainkat mások számára könnyen használható formában tegyük közzé.

Sok éven keresztül foglalkoztam a soros kapuhoz kapcsolódó USB-meghajtók fejlesztésével és karbantartásával, végül az USB-kódok karbantartását a rendszermagba helyeztem át. Eme cikkben a korábbi munkám során használt eszközök működését fogom elmagyarázni, de az újabb eszközökről is szólni fogok, amelyek segítettek a rendszermagfejlesztéssel lépést tartani. Ráadásul ezek tették lehetővé, hogy mindezt kisebb erőfeszítéssel tegyem.

A patch és a diff program használata

A rendszermaggal kapcsolatos teendők során az egyik leggyakoribb módszer a patch és diff programok használatát foglalja magában. Ezt használhatjuk a magfejlesztéshez valamilyen másfajta forráskódkezelő rendszer helyett. Az egyik módszer lényege abban áll, hogy egyidejűleg két fát használ: az egyik a „tiszta” fa, a másik pedig a „munka” fa. A tiszta fa a Linux-változattal együtt kapott rendszermag, míg a másik a tiszta fával megegyező rendszermagváltozatot alapul, és már a módosításainkat is tartalmazza. Itt érkezik el a pillanat, amikor a patch és diff programokkal kivonatot készíthetünk módosításainkról, hogy ezeket majd át tudjuk vinni a rendszermag frissebb változatába.

A példa kedvéért induljunk el a 2.4.18-as rendszermagtól, amelyet a <http://www.kernel.org/pub/linux/kernel/v2.4/linux-2.4.18.tar.gz> címről letöltöttünk a munkakönyvtárunkba:

```
$ ls
linux-2.4.18.tar.gz
Végezzük el a rendszermag kicsomagolását, azután a linux néven létrejövő könyvtárnak adjunk valami beszédesebb nevet:
$ tar -zxf linux-2.4.18.tar.gz
$ mv linux linux-2.4.18
$ ls
linux-2.4.18 linux-2.4.18.tar.gz
```

Most készítsünk másolatot az adott rendszermagváltozatról, és lássuk el valamilyen más névvel:

```
cp -R /usr/src/linux-2.4.18
/usr/src/linux-2.4.18-greg
$ ls
linux-2.4.18 linux-2.4.18-greg
↳ linux-2.4.18.tar.gz
```

A fejlesztési munkát most már nyugodtan végezhetjük a *linux-2.4.18-greg* könyvtárban, és érintetlenül fogjuk hagyni az eredeti rendszermagkönyvtárat. Amint befejeztük a munkát, a javítófoltot a többi érintett munkatársnak el kell küldenünk. A *Documentation/SubmittingPatches* elnevezésű állományban szereplő leírásból megtudhatjuk, hogy a magfejlesztők milyen formában szeretik a foltokat megkapni vagy éppen küldeni. Amellett ez az állomány ismerteti a *dontdiff* állomány használatát is, amely pontosan ilyen javítófoltok létrehozásában van a segítségünkre. A *dontdiff* az állományoknak azon körét határozza meg, amelyet szeretnénk, ha a diff program nem venne figyelembe.

Folt létrehozásához a következő parancs használható:

```
$ diff -Naur -X dontdiff \
linux-2.4.18 linux-2.4.18-greg/ > az_en_foltom
Ez a parancs létrehozza az az_en_foltom nevű állományt, amelynek tartalma a 2.4.18-as maghoz képest módosított rendszermag. Ez a folt a többi érintett fél számára már levélben továbbítható.
```

Friss rendszermagváltozatok

Amint frissebb rendszermag jelenik meg, a működő rendszerben levő módosításokat át kell vinni az új rendszerbe, és a létrehozott foltot meg kell próbálni egy tiszta rendszermagra alkalmazni. A folyamat két lépésből áll:

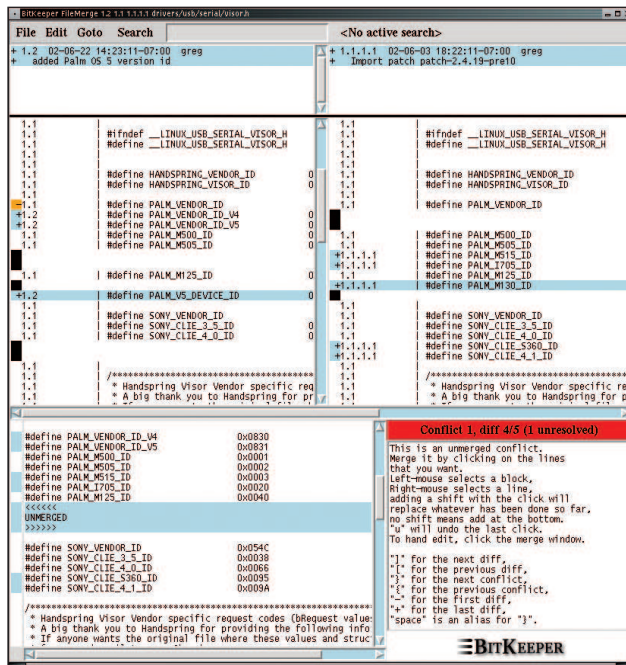
1. Hozzuk létre a javítófoltot, ahogyan azt az előző példában már láttuk.
2. A Kernel.org hivatalos javítófoltját felhasználva növeljük egygel a rendszermag változatszámát:

```
$ cd linux-2.4.18
$ patch -p1 < ../patch-2.4.19
$ cd ..
$ mv linux-2.4.18 linux-2.4.19
```
3. A javítófolt eltávolításával együtt növeljük a munkakönyvtár változatszámát, majd érvényesítjük a frissítést:

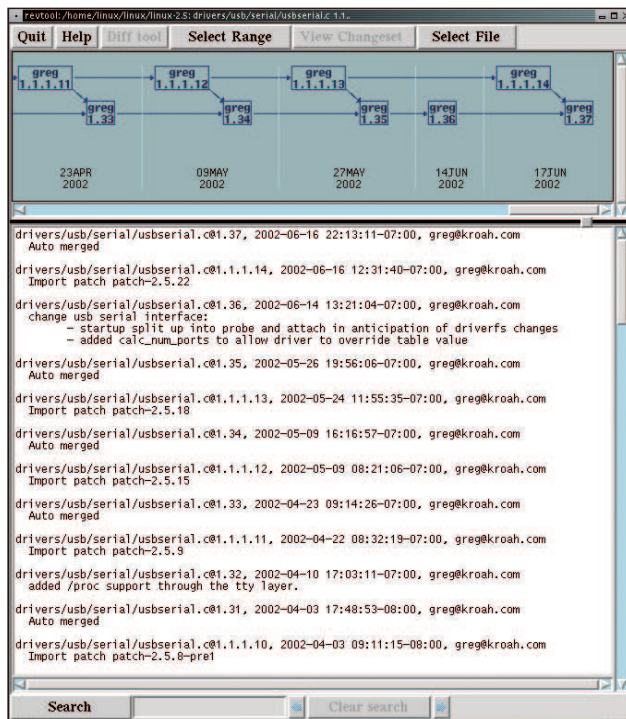
```
$ cd linux-2.4.18-greg
$ patch -p1 -R < ../my_patch
$ patch -p1 < ../patch-2.4.19
$ cd ..
$ mv linux-2.4.18-greg linux-2.4.19-greg
```
4. A hivatalos frissítés után kíséreljük meg saját javítófoltunk telepítését:

```
$ cd linux-2.4.19-greg
$ patch -p1 < ../my_patch
```

Amennyiben a folt telepítése nem zajlik hibátlanul, oldjuk meg a felmerült összeférhetlenséget. A folt ezekről a *.rej* és *.orig* kiterjesztésű állományok révén fog tájékoztatást adni, amelyeket összehasonlíthatunk, és a hibákat kedvenc szövegszerkesztőnk segítségével, kézzel javíthatjuk. Ez az összefésülési folyamat lehet a legnehezebb szakasz, ha a forráskódjában magunk is végeztünk olyan módosításokat, amelyek érintik a korábban mások által megváltoztatott területeket. Amennyiben a fent leírt fejlesztési folyamat mellett döntesz,



1. kép Egy jellemző összevágatási nehézség



2. kép A BitKeeper nyomon követi a módosításokat

javaslom, a foltkészítő eszközöket szerezd be a

☞ <http://cyberelk.net/tim/patchutils> webhelyről. Ezek a programok ezernyi módon könnyítik meg a szöveges javítófoltok kezelését, és a magfejlesztőknek már sok fáradtságos munkaórát takarítottak meg.

A forráskódkezelés

A patch és diff programok használatával végzett magfejlesztés általában véve meglehetősen jól működik. Egy idő után azonban az emberek többsége belefárad és elkezdi olyan más

munkamódszer után kutatni, ami nem jár annyi foltözéssel meg összevágatással.

Néhány éve fedeztem fel a BitKeepert – amely a <http://www.bitkeeper.com> címen érhető el –, s azóta ezt használom magfejlesztésre. Eredetileg lehetővé tette számomra, hogy nyomon kövessem a rendszermagfán végbemenő külső módosításokat, és hogy a saját módosításaimat minden különösebb erőfeszítés nélkül közzétegyem. Minthogy maga **Linus Torvalds** és **Marcello Tosatti** szintén a BitKeepert használja magfejlesztésre, számomra nem jelent nehéz feladatot, hogy a fő rendszermagfába szánt foltjaimat elküldjem nekik. A BitKeepert magfejlesztő eszközként meglehetősen sokan visszásnak tartják, tekintettel a BitKeeper engedélyezési stratégiájára. A használatba vétel előtt olvasd át a felhasználási szerződést és dönts el, valóban használni kívánod-e. A fejlesztőeszközzel és néhány paranccsal való megismerkedés érdekében végiglépkedhetsz a BitMover honlapján.

A BitKeeperrel végzett munka során rendszermagodat Linus vagy Marcello rendszermagjára építheted, vagy a különböző változatok felhasználásával akár a saját változatodat is létrehozhatod.

Mindazonáltal a BitKeeperrel egy saját rendszerfa építését javaslom, hacsak nem a Linusnak vagy Marcellónak szánt javítófoltok küldését fontolgatod. Ily módon a különböző magfejlesztők által készített változások nem fognak maguk alá temetni és a teendőidre tudsz összpontosítani.

A két fa

A BitKeeperrel végzett magfejlesztéshez ugyancsak két ismerős fát kell létrehozni (ezt a továbbiakban inkább rendszerraktárnak – repository – nevezném): a „tiszta” fát és a „munka” fát. A kiindulásképpen használt tiszta Bitkeeper-rendszerraktárhoz a saját munkakönyvtárban a Linux-változathoz kapott rendszermagot használod:

```
$ ls
linux-2.4.18.tar.gz
Uncompress this kernel:
$ tar -zxf linux-2.4.18.tar.gz
$ ls
linux linux-2.4.18.tar.gz
```

Ezek után hozd létre a linux-2.4 nevű BitKeeper-feladatot:

```
$ bk setup linux-2.4
```

A BitKeeper most néhány kérdést fog feltenni, azután szerkesztésre megnyit egy állományt – a feladat leírását ebben lehet megadni. Ehhez használhatod a kedvenc szövegszerkesztőt, majd a fájlt mentsd lemezre.

Egy **linux-2.4** nevű könyvtár fog létrejönni, ide fogja elhelyezni a feladatot. Az eredeti rendszermagváltozatot hozzuk át egy új rendszerraktárba:

```
$ ls
linux linux-2.4 linux-2.4.18.tar.gz
```

```
$ bk import -tplain linux linux-2.4
```

Ez a művelet el fog tartani egy darabig. Amint a BitKeeper végzett az fájlok átvételével, javaslok ennek a pontnak a rendszermag változatszámával történő megjelölését. Ez később a különböző rendszermagváltozatok fellelését meg fogja könnyíteni:

```
$ cd linux-2.4
$ bk tag LINUX_2.4.18
```

Készíts hasonmást erről a rendszerraktárról, vagyis a tiszta rendszerfáról, azaz klónozd, hogy később módosítani tudd:

```
$ bk clone linux-2.4 greg-2.4
```

A rendszermagon végzett összes munkát a **greg-2.4** nevű

könyvtárban fogjuk elvégezni. A bk-klónnál használhatjuk a -l lehetőséget, amely nagy lemezrészt használ fel, cserébe a meta-adatállományokhoz tartozó befűzések létrehozása következtében gyorsabban fog futni. Amint megtörténik egy állomány frissítése, a régi befűzést a BitKeeper megszünteti, és ahol csak szükséges, újat hoz létre. Ha nagyszámú különböző rendszerraktárt hozol létre ugyanazon a lemezen, ajánlott ennek a lehetőségnek a használata.

Amint befejeztük a változásokészletek létrehozását, beépítve a fejlesztési folyamat során végzett összes módosítást – erről a BitKeeper oktató oldalai részletesen szólnak –, létre szeretnénk hozni a változtatásainkat felsorakoztató foltot. Ez egy a *greg-2.4* könyvtárból kiadott egyszerű paranccsal tehető meg:

```
$ bk export -tpatch -rLINUX_2.4.18..+ -h \
> ../az_en_foltom
```

Ez a parancs létre fog hozni a jelzett rendszermagtól – LINUX_2.4.18 – a legfrissebb változásokészletekig terjedő foltot, amelyet mentsünk lemeze *az_en_foltom* néven. Ez a folt már bármilyen címre továbbítható levélben, pontosan ugyanúgy, ahogy a diff által készített folt. Biztosan észre fogod venni, hogy ennek az elkészítése sokkal rövidebb ideig tartott, mint a folt létrehozása a patch és diff programok használatát feltételező előző módszerrel.

Új rendszermagváltozatok

Amint megjelenik egy új rendszermagváltozat működő rendszerből, a módosításokat nyilván át szeretnéd vinni az új rendszermagba. Itt mutatkozik meg a BitKeeper igazi ereje a hagyományos patch-diff-módszerrel szemben. Először térjünk vissza az eredeti tiszta fához, és hozzuk át az új foltot:

```
$ ls
greg-2.4 linux-2.4 patch-2.4.19
```

```
$ cd linux-2.4
$ bk import -tpatch -SLINUX_2.4.19
➔ ../patch-2.4.19
```

Ha a BitKeeper egyes állományokat újonnan létrehozottként vagy töröltként jelölt meg, azok valójában még csak át lettek nevezve vagy át lettek helyezve a fában. Ekkor a BitKeeper egy grafikus felülettel ellátott eszközt bocsát a rendelkezésedre, amellyel kiderítheted, hogy tulajdonképpen mely állományok jöttek most létre, melyek kaptak új elnevezést, és mik kerültek törlésre.

Most pedig térjünk vissza a munkarendszerraktárhoz, és hozzuk át bele a módosításokat:

```
$ cd ../greg-2.4
$ bk pull
```

A BitKeeper a 2.4.18-as és 2.4.19-es változatú rendszermagok módosításait összeválogatja a munkarendszerraktárba. Abban az esetben, ha a saját módosításaid valamilyen összefésülési nehézségbe ütköznek az új rendszermagban, a program tájékoztatni fog róla, és meg fogja kérdezni, hogy mit kívánsz tenni. Az ehhez hasonló összeütközések megoldásához a nagy segítséget jelentő háromutas összeválogató eszköz használatát javaslom. Ez az eszköz megmutatja az eredeti fájlt a saját magad által végzett változtatásokkal, illetve a folt vagy mások által véghez vitt módosításokkal. Ezt követően lehetővé teszi, hogy eldöntsd, melyik változtatást szeretnéd elfogadni, esetleg a fájlt kézzel szerkeszteni, netán mindkét módosítást érvényesíteni. A 1. kép egy a fő rendszermagban végzett változtatással összeütközésbe került állományt szemléltet, amelynek módosítását jómagam hajtottam végre. Amint végeztél az összeférhetlenségek feloldásával – ami

ugye nem volt sokkal könnyebb, mint a .rej kiterjesztésű állományok átvizsgálása –, máris folytathatod a munkát a frissített rendszermagon. Hadd mondjam el ismét, hogy az összes általunk végzett módosítást a *greg-2.4* könyvtárban kiadott alábbi paranccsal egyetlen foltba lehet foglalni:

```
$ bk export -tpatch -rLINUX_2.4.19..+ -h \
> ../my_patch
```

A BitKeeper további előnyei

A BitKeeper lehetővé teszi az adott fájlban végzett összes módosítás időrendbeli, egyszerű áttekintését. Látható például, hogy a módosítást az egyik rendszermagfolt végezte-e vagy maga a fejlesztő. A 2. képen látható példában a rendszerraktárban tárolt *drivers/usb/serial/usbserial.c* állomány időbeli változásait kísérhetjük figyelemmel. Ezzel az eszközzel megfigyelhetjük, milyen változások zajlottak le, de még azt is, hogy egy-egy változatban mely programsorok lettek módosítva. A magfejlesztéshez használt BitKeeper egyik roppant nagy előnye, hogy igen nagy teljesítményű változatkezelő rendszer, és lehetővé teszi, hogy ugyanazon a programkódrészletben több más fejlesztővel együtt egyidejűleg dolgozz. Lehetővé teheted, mások számára, hogy munkafádból kivonatot készítsenek, vagy a munkafád tárolására akár önálló kiszolgálógépet is telepíthetsz. A hasznos példák végett keresd fel a BitKeeper oktatóoldalait, amelyekből megtudhatod, hogyan kell telepíteni, és milyen módon kell a fejlesztési ciklust működtetni.

Összefoglalás

A Linux-magfejlesztés két különböző útját mutattam be: az egyik a patch és diff programok használatát jelenti, a másik a BitKeeperét. Személy szerint úgy találom, hogy a BitKeeperrel több időt tudtam a tényleges fejlesztési munkára fordítani, és kevesebbet kellett a módosítások bogarászásával töltenem. Továbbá nem veszítettem el a fonalat a 2.2, 2.4 és 2.5 jelű rendszermagokba épített linuxos USB- és gyors csatlakoztatású (Hot Plug) PCI egységek meghajtóprogramjaival kapcsolatban sem.

Linux Journal 2002. szeptember, 101. szám



Greg Kroah-Hartman

jelenleg a Linux-rendszerek USB és gyors csatlakoztatású (PCI Hot Plug) egységeinek rendszermagba épített meghajtóprogramjainak fejlesztője. Az IBM-nél dolgozik, és rendszer-maggal kapcsolatos kérdésekkel foglalkozik.

A greg@kroah.com címen lehet vele kapcsolatba lépni.

Kapcsolódó címek

Val Henson és Jeff Garzik „BitKeeper for Kernel Developers” (BitKeeper rendszermagfejlesztők számára), 2002, Ottawa Linux Symposium Proceedings.

A BitKeeper oktatóoldalai a Világhálón

➔ <http://www.bitkeeper.com/Test.html>

A BK-műveletről pingvinstílusban a 2.5 rendszermagban, a *Documentation/BK-usage/bk-kernel-howto.txt* állományban olvashatsz.

Külső változások kezelése a BitKeeper használatával

➔ <http://www.bitkeeper.com/tracking.ps>

Biztonság a helyi hálózaton: IP Tables

Mick azt fejtegeti, hogyan kell az IP Tableshez kapcsolódó tulajdonosegyeztetési kiterjesztést (Owner Match Support) használni ahhoz, hogy megakadályozzuk a felhasználók hálózati protokollokkal kapcsolatos visszaéléseit.

A legtöbbször talán határozottan tűzfaleszközként gondol az IP Tablesre, amellyel a külső támadókat sarokba lehet szorítani. De vajon tudta-e mindenki, hogy ezzel a helyi felhasználók tevékenységének is határt lehet szabni? A kísérleti tulajdonosegyeztetés új lehetőségeket teremt az IP Tables számára, amelyekkel meg lehet akadályozni, hogy a helyi felhasználók mások hálózati folyamatain keresztül csomagokat küldjenek.

A példa kedvéért tételezzük fel, hogy a rendszergazdai azonosító (root) egyik cron által ütemezett munkája az Stunnel programot használja arra, hogy egy távoli *rsync*-folyamat számára állományokat küldjön. Amíg az alagút nyitva van, ezt bármelyik helyi hálózaton levő felhasználó használhatja a távoli *rsync*-kiszolgálóhoz való hozzáférésre. Az IP Tables a segítségére siethet az említetthez hasonló hálózati élőkódés megelőzésében. Írásunk pontosan azt fogja bemutatni, hogyan is teszi ezt.

A kérdés

Az alagút-segédprogramok a rendelkezésre álló biztonsági eszközök egyik legfontosabb új csoportját alkotják. Lehetővé teszik a számunkra az olyan megbízhatatlan szolgáltatások burkolását, mint amilyen a Telnet, az IMAP és a POP3 – titkosított látszólagos alagutakban átláthatóan és hatékonyan. Már hosszasan írtam ezeken az oldalakon a Secure Shellről (SSH) és annak foglaltatottabbító képességeiről; az Stunnel és az SSL Wrap is ehhez hasonló, szintén erre a célra szolgáló ingyenes eszköz a Linux-rendszerben.

Vajon mi történik, ha egy ilyen alagutat telepítasz egy többfelhasználós rendszerbe? Mi lesz képes megállítani a meghatalmazással nem rendelkező helyi hálózaton levő felhasználókat abban, hogy a saját forgalmukat átküldjék ezen az alagúton? Mostanáig gyakorlatilag semmi sem tudta ezt megakadályozni. Mint hogy az alagút-segédprogramok többsége úgy működik, hogy új figyelőkaput hoz létre – például: `localhost:992` – az alagút innenső oldala számára, rendszerint csak a kiszolgálóalkalmazástól függ az alagút túloldalán, hogy hitelesíti-e felhasználókat. Tegyük fel, hogy az Stunnelt használom SSL-alagút felállítására a saját *cruller* nevű rendszeremtől a távoli *strudel* rendszerig, amelyen Telnetet szeretnék futtatni. Hagyjuk figyelmen kívül, hogy ez a fajta tevékenység SSH-val egyszerűbb lenne; most valamilyen oknál fogva nem szeretném helyileg telepíteni az SSH-t. A távoli gépen, ami az `inetd`-n keresztül a TCP 23-as kapuján át már futtatja a Telnet démon, az Stunnelt démon módban futtatom a következő paranccsal:

```
stunnel -d 992 -r localhost:23
➤ -p /etc/stunnel/strudel.pem
```

A helyi gépen az Stunnelt ügyfélmódban fogom futtatni, és a helyi gép TCP-protokolljának 992-es kapuján fog figyelni, viszont a kapcsolatot az *strudelen* lévő TCP 992-es kapujára az alábbi paranccsal továbbítja:

```
stunnel -c -d 992 -r strudel:992
```

Amennyiben még sohasem használtad az Stunnelt, és ezek a parancsok semmit sem mondanak neked, még ne kezdj el aggódni. Ebből annyi lényeges, hogy ebben a példában a *cruller*ről az *strudelre* a Telnettel való bejelentkezéshez a *crulleren* az alábbi parancsot kell használnom:

```
telnet localhost 992
```

Ezen a ponton az *strudel* kérni fogja a felhasználói azonosítót és a jelszavamat, de bejelentkezési hitelesítéseimet az Stunnel a Telnet-től eltérően titkosítani fogja, és nem a megszokott szöveggé lesznek továbbítva. A távoli gépen működő Stunnel-folyamat a csomagokat vissza fogja fejteni, és át fogja őket adni a Telnet-folyamatoknak. Ez a folyamat a hitelesítési szakaszban nemcsak egyszer fog lezajlani, hanem a teljes Telnet-művelet során végig ismétlődik. Az Stunnel a teljes tranzakció során mindkét fél számára közvetítőként dolgozik, mégpedig olyan közvetítőként, amelynek nincs tudomása az alagútba juttatott adatok mibenlétéről és nem is foglalkozik velük, feltéve, hogy azok TCP alapúak.

Ez eddig szép és jó: rendelkezem titkosítással, ami az Stunnel nélkül nem létezne, és szerény mértékű hitelesítéssel – magának a Telnetnek a természetéből adódóan. Most már csak az okoz fejtörést, hogy bármelyik *crulleren* levő felhasználó képes a helyi gép TCP 992-es figyelő kapujára telnetelni, és bejelentkezni az *strudelre*. Az is aggaszthat, hogy valaki máris megpróbálja a *strudelre* való belépéshez használt jelszavamat kitalálni. Kezdjük ott, hogy vajon hogyan lehet ezeket az embereket megállítani abban, hogy mindenféle csomagot küldözgessenek keresztül az alagúton?

Íme a válasz: az IP Tables programmal és annak tulajdonosegyeztetési kiterjesztésével.

Az eszköz

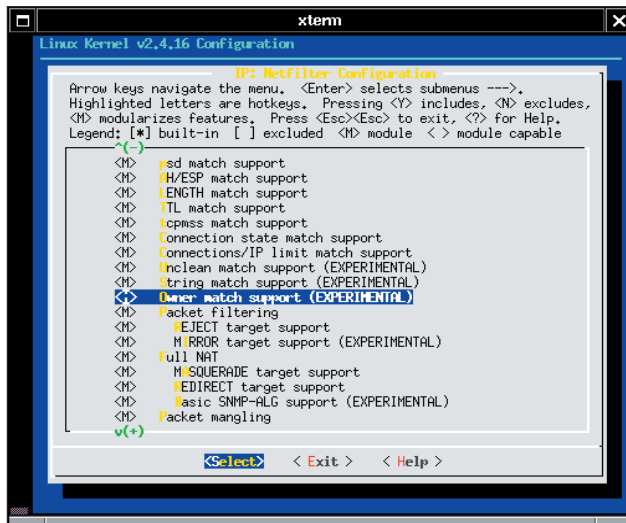
Az IP Tables tulajdonosegyeztetési kiterjesztése négy ellenőrzési szemponttal egészíti ki az `iptables` parancsot:

```
--uid-owner UID: egyeztetési a csomagokat, amelyek az adott felhasználói azonosítóval (UID) jöttek létre
--gid-owner GID: egyeztetési a csomagokat, amelyek az adott csoportazonosítóval (GID) jöttek létre
--pid-owner PID: egyeztetési a csomagokat, amelyek az adott folyamatazonosítóval (PID) jöttek létre
--sid-owner SID: egyeztetési a csomagokat, amelyek az adott szezonazonosítóval (SID) jöttek létre
```

A négy közül mostani céljaink szempontjából az első két adat fontos. Nem biztos, hogy a tulajdonosegyeztetési kiterjesztés szerepel a Linux-változatodban működő rendszermagban. Ha az IP Tables fejlesztőcsoportja nem is feltétlenül, de a Linux-rendszermag fejlesztőcsapata kísérleti szolgáltatásnak tekinti, ezért ennek rendszermagba történő fordítását magadnak kell elvégezned. A program forráskódja ezzel szemben a 2.4-es rendszermag kódalapjának része, és emiatt bármilyen friss Linux-változat rendszermagjának forrásprogramcsomag-

jával (2.4.x) könnyen kivitelezhető.

A rendszermag újrafordításakor több jellemzőt közvetlenül be kell állítani. Először is a *Kódérettségi szint* lehetőségénél válassz a *Prompt for development and/or incomplete code/drivers* (Fejlesztés alatt álló rendszer és/vagy befejezetlen programkód/meghajtók). Ezután a hálózati protokolloknál az egyénileg kiválasztott hálózati protokollokon és szolgáltatásokon kívül ellenőrizd, hogy a *Network Packet Filtering* lehetőség ki legyen választva. Ez be fogja kapcsolni az IP-részcsoportot: a Netfilter beállítását, amelyet az 1. képen tekinthetünk meg. Ezeket a lehetőségeket – csillaggal való kijelöléssel – befördíthatjuk a



1. kép Rendszermag fordítása tulajdonosegyeztetési kiterjesztéssel

rendszermagba, vagy modulokká is fordíthatjuk őket (az M betűvel). A legtöbb ember modulként fordítja őket, egy időben ritkán van rájuk szükség. Az a modul, amelyek a jelen pillanatban a legfontosabb számunkra, az Owner Match Support (tulajdonosegyeztetési kiterjesztés).

A Linux-rendszermag fordításáról, telepítéséről és moduljainak hátralevő részéről másutt nagyon jó leírást olvashatunk, nevezetesen a rendszermag forráskódjának README állományában. Amint végeztél a rendszermag fordításával, telepítésével, és a gépet is újraindítottad, máris használhatod vadonatúj tulajdonosegyeztetési modulodat, ami az `ipt_owner` nevet fogja kapni. A modul betöltéséhez használd a `modprobe` parancsot:

```
modprobe ipt_owner
```

A gyakorlatban az IP Tables-szabályokat talán a `/etc/init.d/inetd` héjprogramból szeretnéd indítani. Ha az IP Tablest valóban így kívánod használni, a fenti `modprobe` parancsot szűrd be a héjprogram elejére, pontosabban bármilyen tulajdonosegyeztetést felhasználó IP Tables-parancs elé. Fontos, hogy a tulajdonosegyeztetési kiterjesztést sem a Bastille-Linux önműködő tűzfalbeállítás, sem a SuSE Linux SuSE tűzfal-héjprogramja jelentősebb módosítás nélkül nem támogatja. Ez nem is meglepő, hiszen ezeket és más egyszerű Internet csomagszűrő eszközöket elsődlegesen alacsony szintű internetes védelmi célokra szánták, semmint a felhasználói jogok haladó szintű szabályozására. Az utóbbihoz az IP Tables-szabályokat magadnak kell megalkotnod.

Ezek után térjünk vissza a *cruller* és az *Stunnel* ügyfélprogramhoz. Tétélezzük fel, hogy a *cruller* rendszermagját az `ipt_owner` modullal fordították, a modult a `modprobe` utasítással töltötték be; minthogy az IP Tables egyelőre még nincs

beállítva, semmilyen csomagszűrés nem történik.

Tegyük fel azt is, hogy az *Stunnel* csatlakozón keresztüli Telnet-használatot kizárólag a rendszergazdára szeretnénk korlátozni, ahogyan azt a cikk elején elterveztük. Talán még nem felejtetted el, hogy a *crulleren* az *Stunnel* alagútprogram számára a 992-es TCP-kapun figyelést állítottunk be, ami a csomagokat az *strudel* ugyanilyen TCP-kapujára tükösíti és továbbítja. Amennyiben a *cruller* nem tűzfalnak lett beállítva, a kimeneti lánc (OUTPUT) számára az „alapértelmezés szerinti elfogadás” alapvelvel dolgozhatunk. Tűzfalakon viszont a láncoknak általában a „mindent” vagy a „mindent visszautasít” alapvel szerint kell működniük. Az egygépes kiszolgálóknál – vagyis amelyekbe csupán egy kártya van szerelve – a bástyakiszolgálókon a kimenő forgalmat illetően néha az engedékenyebb hozzáállás is megengedett. Amennyiben a *crulleren* éppen ez a helyzet, a megkívánt korlátozáshoz egyetlen szűrési szabály használata is elegendő:

```
iptables -A OUTPUT -p tcp --dport 992
  -d localhost -m owner ! --uid-owner root
  -j REJECT
```

Boncolgassuk ezt a parancsot mezőnként:

- A OUTPUT: ez jelzi az IP Tables számára, hogy az OUTPUT lánc végére új szabályt szeretnénk fűzni.

-p tcp: arra utasítja az IP Tablest, hogy csak TCP protokollú csomagokat fogadjon, és töltsd be az IP Tables TCP-lehetőségeit. --dport 992: ez a TCP-re jellemző lehetőség megparancsolja az IP Tablesnek, hogy csak a 992-es kapura irányuló TCP-csomagot fogadja.

-d localhost: azt parancsolja az IP Tablesnek, hogy csak a localhostnak küldött csomagokat fogadja el.

-m owner: előírja az IP Tables számára a tulajdonosegyeztetési kiterjesztés betöltését.

! --uid-owner root: megtiltja az IP Tables számára, hogy a rendszergazdához tartozó folyamatok által létrehozott csomagokon kívül más csomagot is elfogadjon.

-j REJECT: az IP Tablestől azt követeli meg, hogy az ebben a sorban megadott összes szempontnak megfelelő csomagot utasítsa vissza.

Röviden: ez a szabály az IP Tablesen keresztül megmondja a rendszermagnak, hogy a saját gép 992-es TCP-kapujához érkező csomagokat ejtse el, hacsak nem a rendszergazda valamelyik folyamata küldte azokat.

Képzljük most el, hogy a *cruller* házirendje inkább az elővigyázatos „elvetés”-hez áll közelebb, semmint az „elfogadás”-hoz. Az alapértelmezés szerinti elvetés a legtöbb IP Tables-telepítésnél előnyben részesítendő. A legkevésbé előjog alapelve az egyik legfontosabb fogalom az adatbiztonság-technikában: ami nincs félreérthetetlenül engedélyezve, az tilos!

Nekünk azonban most egy hosszabb kimeneti láncra van szükségünk. Egy üres láncból kiindulva: meg kell mondanunk az IP Tablesnek, hogy a már elfogadott szezonokhoz tartozó csomagokat továbbítsa:

```
iptables -I OUTPUT 1 -m state --state
  ESTABLISHED,NEW -j ACCEPT
```

A -state egyeztetési kiterjesztés az IP Tablest állapotellenőrző képességekkel ruházza fel, amelyek révén az IP Tables a csomagokat képes az érvényes szezonok és adatfolyamok viszonylatában kiválogatni. A szolgáltatás – önmagáért való értelmén és szépségén túl – nagymértékben csökkenti az egy-egy művelethez meghatározandó szabályok számát. Az állapotkövetés nélkül egy helyett két szabályra lenne szükség, például egy kifelé irányuló Telnet-műveletnél egyre a „bemenő” és egyre a „kimenő” láncban. Ezért is jó lenne a fenti szabályt szinte

mindig, bármilyen olyan lánc felett alkalmazni, amelynek működési alapleve a „csomagok elejtése (DROP)”.

Ezután az Stunnel számára lehetővé kell tennünk az *strudel*hez történő kapcsolódást:

```
stunnel -A OUTPUT -p tcp -dport 992
-d strudel -j ACCEPT
```

Ez a parancs a „kimenő” lánc végére új szabályt fűz, ami a kimenő kapcsolatoknak megengedi, hogy a TCP 992-es kapuján keresztül a *strudel*hez kapcsolódjanak.

Végül kiadunk egy parancsot, amely az „alapértelmezés szerinti elfogadás”-hoz hasonlít, azzal a különbséggel, hogy egyrészt ez a célgépnél a visszautasítás helyett az elfogadást célozza (ACCEPT), másrészt a `--uid-owner` lehetőségénél a tagadó szerepet betöltő felkiáltójel is hiányzik:

```
iptables -A OUTPUT -p tcp --dport 992 -d
localhost -m owner --uid-owner root -j ACCEPT
```

Vegyünk még egy példát! Az *.rsync* nagy teljesítményű állományátviteli segédprogram, ami képes az állományok közti eltérések alapján dolgozni. Képes egy távoli gépen lévő állományt a helyi gépen szereplővel összehasonlítani, így csak az eltérő részleteket kell letölteni. Az *.rsync*-et az SSH-val együttműködve használhatjuk, és igen, kitaláltad, az Stunnel programmal együtt!

Induljunk ki abból, hogy a *cruelleren* működik egy cron által ütemezett alkalmazás, amely az *rsync*kel a *strudelen* levő *stuff.txt* állománynak és a helyi gépen tárolt másolatának az összehasonlítását végzi. Tételezzük fel azt is, hogy az említett állomány érzékeny adatokat tartalmaz, ezért az állománytörvényszabályozás céljára az Stunnelt használjuk. Kizárólag a helyi rendszergazdáknak – akik mindnyájan a „wheel” csoport tagjai – kell tudniuk szabályozni a héjprogram működését vagy használni az alagutat.

Az *strudelen* az *rsync* démon módban működik, és úgy lett beállítva, hogy egy modult, egy virtuális kötetet megosztva használjon, ennek neve: *attic*. Feltéve, hogy a */etc/rsyncd.conf* helyesen lett beállítva, az *rsync*et démon üzemmódban futtató parancs igen egyszerű:

```
rsync --daemon
```

Az Stunnel a TCP-n ugyancsak rendelkezik a 273-as figyelő kapuval, ami az *rsync*-folyamathoz titkosítja és továbbítja a forgalmat, ez viszont a TCP 873-as kapuján figyel. Az *strudelen* az Stunnelt ilyen módon futtató parancs a következő:

```
stunnel -d 273 -r localhost:873
```

```
➔ -p /etc/stunnel/strudel.pem
```

A *cruelleren* a megfelelő ügyfél-üzemmódú Stunnel-felügyelőkaput az alábbi módon kellene megszólítani:

```
stunnel -c -d 273 -r strudel:273
```

Minden rendben: beüzemeltünk egy alagutat, amivel a *cruelleren* levő TCP 273-as kapuján keresztül küldött csomagok titkosítva lesznek, és az *strudel* TCP-jének 273-as kapujához lesznek továbbítva, ahol visszafejtésre kerülnek, és az *strudel* helyi *rsync*-folyamatához a TCP 873-as kapuján át továbbítódnak. Ha a *crueller*-en lévő közönséges felhasználó, az akadályt jelentő IP Tables-szabályok távollétében, most megpróbálná használni az alagutat, erőfeszítése sikerrel járna:

```
rsync --port=273 -v localhost::attic/stuff.txt
stuff.txt
```

```
wrote 508 bytes read 575 bytes 2166.00 bytes/sec
total size is 48188 speedup is 44.49
```

Ha azonban a *cruelleren* érvényesítjük azt az IP Tables-szabályt, amely az *rsync*-alagút használatát a „wheel” csoport tagjaira korlátozza:

```
iptables -A OUTPUT -p tcp -d localhost
```

```
➔ --dport 272 -m owner ! --gid-owner wheel
➔ -j REJECT
```

A közönséges felhasználónak az a kísérlete, hogy elcsenje a *stuff.txt* állományt, most már kudarcba fog fulladni:

```
rsync --port=273 -v localhost::attic/stuff.txt
rsync: failed to connect to localhost:
```

```
Connection refused
```

```
rsync error: error in socket IO (code 10)
at clientserver.c(97)
```

Amennyiben viszont a *wheel* csoportba tartozó *admin7* próbál meg a kiszolgálógéphez kapcsolódni, sikerül csatlakoznia:

```
rsync --port=272 -v localhost::chumly/stuff.txt
stuff.txt
```

```
wrote 508 bytes read 575 bytes 2166.00 bytes/sec
total size is 48188 speedup is 44.49
```

Remélhetőleg feltűnt, hogy ez „alapértelmezés szerinti elfogadás”-t feltételez. Ha viszont a „kimenet” a „mindent elvet” alapelv alapján működik, egy olyan szabályra lenne szükségünk a kimeneti láncban, ami lehetővé tenné a kimenő kapcsolatot a *strudel* TCP-jének 273-as kapujával. A „kimenet” láncnak ugyancsak egy megengedő létező, illetve kapcsolódó szezoniszabállyal kell kezdődnie. Mivel mindkét szabály erősen emlékeztet az előző példánkban szereplőkre, mellőzni fogom a bemutatásukat.

A tulajdonosegyeztetési kiterjesztéshez és az Stunnelhez kapcsolódó vegyes megjegyzések

Mint látható, a `--uid-owner` és a `--gid-owner` használata meglehetősen egyszerű. Még nem említettem, de a példában már bemutattam, hogy mindkét módszer megengedi nevek és számszerű azonosítók használatát.

A másik fontos téma, amivel most foglalkoztam, a TCP-burkolóhoz hasonló hozzáférés-szabályozás. Bármilyen TCP-burkolót használó rendszer – vagy olyan rendszer, amelynek Stunnel binárisát a *libwrapper*-támogatással fordították – */etc/host.allow* állományát ki kell egészíteni megfelelő bejegyzésekkel, amelyek azon az adott gépen az Stunnelt képesessé teszik feladata ellátására: akár ügyfélmódban, akár démonként. Ez tulajdonképpen jó dolog; ahelyett, hogy újabb akadályt látnál benne az Stunnel számára, inkább úgy gondold rá, mint a biztonságot jelképező hagyma egyik újabb rétegére. Végül rád hagyom, kedves olvasóm, a `--pid-owner`, `--sid-owner` lehetőségekkel való szöszmötölést. Adnék azonban egy tanácsot. Sok démon szülő-PID-jeit az indulás során meghatározott helyre teszi, nevezetesen a */var/run/ssh.pid*-be. Az IP Tables indulása során beolvasott ehhez hasonló PID-állomány segít beazonosítani az egyes folyamatoktól származó csomagokat. Szerencse fel!

Linux Journal 2002. augusztus, 100. szám



Mick Bauer

(mick@visi.com) hálózati biztonsági tanácsadó az Upstream Solutions Inc.-nél Minneapolisban (Minnesota). Mick a szerzője a hamarosan megjelenő új O'Reilly-könyvnek, amelynek címe „Building Secure Servers With Linux”.

Kapcsolódó címek

A Netfilter weblapja ➔ <http://netfilter.samba.org>
Az *rsync* weblapja ➔ <http://rsync.samba.org>

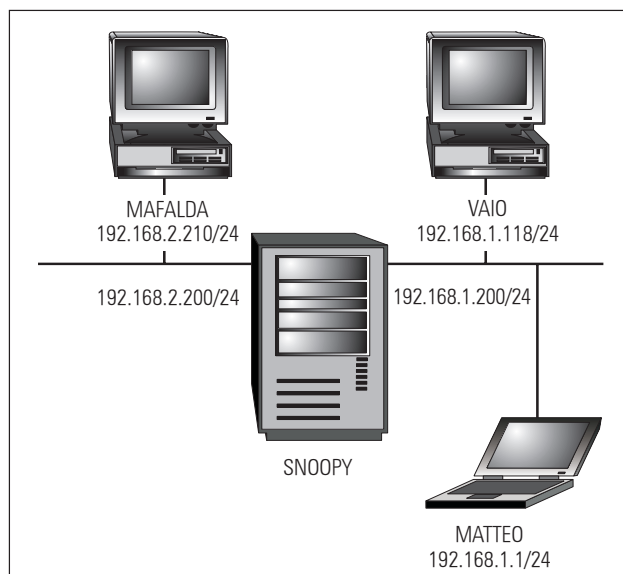
Csoportszórásos útválasztókód a rendszermagban

Tegyük lehetővé, hogy egyetlen csomag több címre is megérkezhesen, és takarítsunk meg jókora sáv szélességet! – ezt ígéri a csoportszórás (IP multicasting). Lássuk, vajon hogyan kezeli a Linux!

Ezek az oldalak arról fogok beszélni, miképpen kezeli a Linux-rendszermag a csoportszórásos (multicast) forgalmat, illetve hogyan lehet a rendszermagkód egyszerű foltozásával belenyúlni. Igaz, meglehetősen egyedi témát vetünk fel, az itt elhangzó dolgok azonban bárki számára izgalmasak lehetnek, akit érdekel a csoportszórásos útválasztás. Ha figyelni, esetleg módosítani szeretnének valamilyen létező csoportszórásos protokollt, hasznos segítséget nyújthatnak az alábbiak. A Milánói Egyetemen egy új, CAMP (Call Admission Multicast Protocol) névre keresztelt protokollt fejlesztünk, ami néhány fontos döntéséhez a csoportszórásos rendszermagkód által nyújtott adatokat használja fel. Képesnek kell lennünk tehát a fontos eseményekről (például a JOIN- és LEAVE-kérésekről) szóló értesítések fogadására. Mint azt valószínűleg már többen ismerik, a Linux-rendszermag csoportszórásos útválasztóként is képes működni, támogatva mind az 1-es, mind a 2-es PIM (Protocol Independent Multicast, <http://netweb.usc.edu/pim>) protokollt. Valamennyi MFC (Multicast Forwarding Cache) frissítési műveletet egy a rendszermaggal kapcsolatot tartó, felhasználói módban futó külső folyamat végzi el. Ebben a cikkben azt fogjuk vizsgálni, hogyan használja fel a rendszermag a felhasználó módú démon által küldött üzeneteket az MFC frissítésére. Rövid bemutatás után részletesebben is ismertetjük a saját megoldásunkat. *Ábránkon* a kipróbáláshoz használt alhálózatunk szerkezetét mutatjuk be. Mint látható, a Snoopy a PIMd (2.1.0-alpha 29.9 változatát) a Linux 2.4.18-as rendszermagváltozat felett futtatva működik csoportszórásos útválasztóként. A rendszermag minden csoportszórással kapcsolatos kódja két fájlban található meg: `ipmr.c` (`net/ipv4/ipmr.c`) és `mroute.h` (`include/linux/mroute.h`). Mielőtt belenézni a kódba, két másik fájl is meg kell említenünk: a `/proc/net/ip_mr_vif-et` és a `/proc/net/ip_mr_cache-t`. Ahogy hamarosan látni fogjuk, ez a két fájl különösen fontos a csoportszórásos útválasztó pillanatnyi állapotának felmérése során. Az `ip_mr_vif` fájl felsorolja a csoportszórásos műveletekbe bevont valamennyi virtuális csatolófelületet, míg az `ip_mr_cache` az MFC pontos állapotáról ad felvilágosítást.

Példaképpen elkezdünk csoportszórásos forgalmat küldeni a Vaioról a 224.225.0.1 címre. A csomagokat a Snoopy fogadja, de nem küldi tovább az eth1-re, hiszen még nem érkezett JOIN-jel a Mafaldáról. Most nézzük azt az állapotot, amikor a Mafalda kiad egy JOIN-t a 224.225.0.1 címre. Az 1. és 2. táblázatban láthatjuk a két proc fájl tartalmát (a Snoopy gépen).

Az `ip_mr_vif` szerint két csatoló (eth0 és eth1) vesz részt csoportszórásos útválasztásunkban, míg a harmadikként felbukkanó csatoló (pimreg) a csoportszórás-kezelő által bejegyzett virtuális eszköz. A 2. táblázatban a PIMd által felépített útvonalat láthatjuk. Az eth0 (Iif: 0) csatolón a 192.168.1.118 címmel azonos forrásból (Origin: 7601A8C0) érkező és 224.225.0.1 (Group: 0100E1E0) címre szánt



A próbahálózat szerkezete

csomagok az eth1 (Oifs: 1: – a második számot egyelőre hagyjuk figyelmen kívül) csatolóra irányítódnak át. Most a Mafaldáról adjunk ki egy LEAVE üzenetet. Néhány másodperc múltán a `/proc/net/ip_mr_cache` tartalma megváltozik. Az eredményt a 3. táblázatban találjuk.

Továbbra is ugyanaz maradt a csatolónk és a csoportjelzésünk, a bemenő csatoló (input interface) viszont már -1-re változott, kimenő csatolónk pedig egyáltalán nincs. Ez azért van így, mert a csomagok továbbra is érkeznek az eth0-n, csak éppen miután a többi csatolón senki nem szándékozik megkapni őket, valamennyit el kell vetni. Amikor a csomagokat ilyen okból vetjük el, mindig új bejegyzés készül a feloldatlan címekhez szánt különleges sorban. A feloldatlan cím létét jelzi, hogy a bemenő csatoló értéke -1 lesz, illetve nem fogunk kimenő csatolót találni.

A feloldatlan című csomagok ilyenfajta sorolásának megvan a maga oka: az IGMP-csomag fogadása és a megfelelő MFC-bejegyzés létrehozása ugyanis igen sok időt vesz igénybe (próbahálózatunkon is közel 2–3 másodpercet, egymással kapcsolatban álló, csoportszórásos útválasztókat tartalmazó nagyobb hálózatokon akár 20–30 másodpercet). Amikor előző példánkban a Vaio csoportszórásos csomagokat kezdett el továbbítani, igen nagy esély volt rá, hogy a Mafalda JOIN üzenetét a Snoopy gép még nem kezelte, és a csomagok továbbításához szükséges megfelelő MFC-bejegyzés sem készült még el. Ezért, hogy a Vaio által küldött csomagokat ne veszítsük el addig, amíg a JOIN-kérelmek végrehajtódik és az új MFC-bejegyzés elkészül, egy különleges gyorsárba helyezzzük őket. Amint

1. táblázat A /proc/net/ip_mr_vif tartalma

Interface	BytesIn	PktsIn	BytesOut	PktsOut	Flags	Local	Remote
0 eth0	129090	1655	1872	24	00000	C801A8C0	00000000
1 eth1	1872	24	129090	1655	00000	C802A8C0	00000000
2 pimreg	0	0	0	0	00004	C801A8C0	00000000

az MFC-bejegyzés elkészül, a sort felszabadítjuk, és a benne várakozó csomagokat a megfelelő célnak továbbítjuk. Nyilvánvaló, hogy teljesítmény- és memóriakorlátok miatt ez a sor nem nőhet túlságosan nagyra. Ezt egy nagyon egyszerű időzítő szolgáltatás hozzáadása oldja meg, ami időről időre törli a feloladatlan bejegyzéseket (`ipmr_expire_process()`). Vizsgáljuk meg a folyamatban felhasznált adatszerkezeteket (lásd az 1. listán).

A `vif_device` a valós hálózati csatolóhoz rendelt virtuális eszköz. A `dev` mező a valós csatolóeszközt (hardware interface) képviselő `net_device` szerkezetre irányított mutató. Számunkra azonban az `mfc_cache` szerkezete még érdekesebb. Mezői önmagukért beszélnek, és a 2., illetve a 3. táblázatban közölt adatoknak felelnek meg.

A `ipmr.c`-ben használt három fő változó a következő:

```
/* Eszközök */
static struct vif_device vif_table[MAXVIFS];

/* További gyorsítók */
static struct mfc_cache
↳ *mfc_cache_array[MFC_LINES];

/* Feloladatlan bejegyzések sora */
static struct mfc_cache *mfc_unres_queue;
```

A `vif_table` nem más, mint a PIMd által készített virtuális csatolókat tartalmazó tömb; az `mfc_cache_array` az MFC-t képviseli; az `mfc_unres_queue` a fent leírt feloladatlan bejegyzések listája. A csoportszórás-kezelő kód elemzése előtt talán érdemes pár szót ejteni az `mfc_cache` szerkezet egyik alkotóeleméről: a TTL tömbről. A tömb minden egyes értéke közvetlenül a `vif_table`-hez van rendelve. Tulajdonképpen minden egyes csatolófelülethez rendelt összes csoportszórásos címhez tartozik egy-egy bájttértek, ami a TTL-határt adja meg. Mint azt később látni fogjuk, ezt az értéket hasonlítjuk majd minden IP-csomag TTL mezőjéhez, ha a csomag továbbításáról kell majd döntenünk.

Most, hogy már láttuk a csoportszórásos útválasztás adatainak alapszerkezetét, elkezdhetjük vizsgálni, miképpen kezeli őket a rendszermag. Az összes szolgáltatást egyetlenegy fájlban találjuk meg: az `ipmr.c`-ben. Ne feledjük, hogy az itt található kód magát az útvonalválasztó kódot nem tartalmazza. Az `ipmr.c`-ben található függvényeket a csoportszórásos útvonalválasztó démon (jelen esetben a PIMd) fogja felhasználni a fenti adatszerkezetek kezeléséhez. Egyszerűen szólva, amikor a PIMd úgy dönt, hogy itt az ideje felvenni vagy törölni egy adott útvonalat, csak küld egy üzenetet a rendszermagnak, amelyben megadja,

hogyan kell végrehajtania. Hogy ezt megtehesse, a PIMd-nek képesnek kell lennie IGMP csomagok fogadására – ezeket a rendszermag adja át a felhasználótérnek. A PIMd két különböző módon tartja a kapcsolatot a rendszermaggal: `ioctl`-ek útján, illetve `setsockopt()` rendszerhívásokkal. A `vif` és `mfc` táblákat egyaránt

`setsockopt()` rendszerhívásokkal éri el.

Hogy jobban megérthessük működésének a lényegét, nézzünk meg a PIMd kódjának néhány részletét. Különös figyelmet érdemelnek a rendszermaggal társalgó függvények, amelyeket a PIMd terjesztés `kern.c` fájljában találjuk meg. A `k_chg_mfc()` függvény felelős az új MFC-bejegyzések felvételéért, illetve a már létező elemek módosításáért, míg az elemek törlését a `k_del_mfc()` végzi. Hogy a rendszermagnak megmondhassuk, hogyan kell a csoportszórásos csomagokat továbbítani, néhány, az `mfc_cache` szerkezetekben felsoroltakhoz hasonló adatot át kell adnunk a felhasználói démonból. Lényeges, hogy ezt az adatot be kell csomagolnunk az `mfccctl`-ként meghatározott új szerkezetbe (2. lista).

A példánkban használt mezőneveknek maguktól értetődőnek kell lenniük. Mégis fontos kiemelni az `mfc_ttls` tömb szerepét. Mint azt korábban elmondtuk, ez az érték jelzi a TTL-határt; a felhasználó-térbeli démon azonban ezt a mennyiséget egy kicsit más módon kezeli. A `k_chg_mfc()` függvénynek kell a rendszermagnak megmondania, hogy melyik csatolófelületeken kell továbbítani a csoportszórásos csomagot. Ehhez viszont a kimenő felületek listáját kell átadni – ezt a szerepet a `mfcc_ttls` tölti be. Az elvet az alábbi kódrészlet mutatja be:

```
for (vifi = 0, v = uvifs;
     vifi < numvifs; vifi++, v++)
{
    if (VIFM_ISSET(vifi, oifs))
        mc.mfcc_ttls[vifi] =
↳ v->uv_threshold;
    else
        smc.mfcc_ttls[vifi] = 0;
}
```

Amennyiben a csatolófelület éppen egy csoportszórásos cím kimeneti csatolófelülete, a hozzá tartozó TTL határ valós értéket vesz fel; egyébként az értéke nulla lesz. A nulla értéket

2. táblázat A /proc/net/ip_mr_cache tartalma

Group	Origin	lif	Pkts	Bytes	Wrong Oifs
0100E1E0	7601A8C0	0	755	58890	0 1:1

3. táblázat A /proc/net/ip_mr_cache tartalma a LEAVE-üzenetet követően

Group	Origin	lif	Pkts	Bytes	Wrong Oifs
0100E1E0	7601A8C0	-1	4	0	0

1. lista A csoportszórásos útvonalválasztó kódban használt „vif_device” és „mfc_cache” szerkezetek

```

struct vif_device
{
    /* A használt eszköz */
    struct net_device      *dev;

    /* Kimutatások */
    unsigned long         bytes_in,bytes_out;
    unsigned long         pkt_in,pkt_out;

    /* Forgalmirányítás (NI) */
    unsigned long         rate_limit;

    /* TTL határ */
    unsigned char         threshold;

    /* vezérlési zászló */
    unsigned short        flags;

    /* Címek (remote a tunnelekhez)*/
    __u32                  local,remote;

    /* fizikai csatlakozás fellet-index */
    int                    link;
};

struct mfc_cache
{
    /* a gyorsítótárhoz tartozó bejegyzés */
    struct mfc_cache *next;

    /* a bejegyzés melyik csoportba tartozik */
    __u32 mfc_mcastgrp;

    /* a csomag forrása */
    __u32 mfc_origin;

    /* forrás csatlakozás */
    vifi_t mfc_parent;

    /* jelzők */
    int mfc_flags;

    union {
        struct {
            unsigned long expires;
            /* Feloldatlan puffer */
            struct sk_buff_head unresolved;
        } unres;
        struct {
            unsigned long last_assert;
            int minvif;
            int maxvif;
            unsigned long bytes;
            unsigned long pkt;
            unsigned long wrong_if;
            /* TTL határ */
            unsigned char ttls[MAXVIFS];
        } res;
    } mfc_un;
};

```

2. lista A PIMd-ben használt mfccctl

```

struct mfccctl
{
    /* mcast eredete */
    struct in_addr mfcc_origin;

    /* A kördőses csoport */
    struct in_addr mfcc_mcastgrp;

    /* hova érkezik */
    vifi_t mfcc_parent;

    /* hova megy */
    unsigned char mfcc_ttls[MAXVIFS];

    /* csomagszámlálás a forrás csoporthoz */
    unsigned int mfcc_pkt_cnt;
    unsigned int mfcc_byte_cnt;
    unsigned int mfcc_wrong_if;
    int mfcc_expire;
};

```

a rendszermag az adott csoporthoz tartozó, nem kimenő csatlakozás felületnek fogja értelmezni, következésképpen az mfc_cache szerkezet megfelelő bájttal a 255-ös (decimális)

értékre állítja be, és a csomagot nem továbbítja.

Most pedig nézzük meg, mit is tesz a rendszermag, amikor egy új MFC-bejegyzés beillesztését kérő üzenetet kap. A kérés az ipmr_mfc_add() függvény kezeli. A rendszermag ellenőrzi az MFC jelenlegi bejegyzéseit, hátha csak egyszerű frissítésről van szó. Ha talál megfelelő elemet, az új TTL értékeket a már meglévő mfc_cache szerkezetbe másolja, illetve egyúttal a minvif és maxvif értékeket is frissíti. Ezek az értékek egy adott üzenetszóró címhez tartozó összes kimenő csatlakozás index-értékeinek két határértékét tartalmazzák. Ezt a munkát az ipmr_update_thresholds() végzi. Az érthetőség kedvéért a 3. listában ezt a függvényt is bemutatjuk, mivel így jobban látható a minval és maxval mezők jelentése.

De térjünk vissza az ipmr_mfc_add() függvényünkhöz! Tételezzünk fel egy olyan esetet, ahol nem találunk létező MFC-bejegyzést. Ez esetben új szerkezetet kell lefoglalni, majd beszúrni az MFC-táblába. A művelet befejezése után a rendszermagnak már csak egy dolgot kell elvégeznie: továbbítani kell az összes olyan, az mfc_unres_queue sorban várakozó feloldatlan csoportszórásos csomagot, amelynek a frissen beillesztett csomópont lett volna a célja. Ha talál ilyen csomagot, eltávolítja és továbbítja az új csatlakozás felületre.

A másik feladat, amit végre kell hajtani: az MFC törlése. Ez eléggé magától értetődő – az adatszerkezetek alapján véve ugyanazok, mint amiket eddig láttunk. A gyorsítótárbejegyzés eltávolításához a felhasználó módú démon a k_del_mfc() függvényt hívja meg, a rendszermagmód-kezelő (handler) pedig meghívja az ipmr_mfc_delete()-t. Ez a függvény

3. lista Így frissíti a meglévő MFC-bejegyzéseket a rendszermag

```
static void
ipmr_update_thresholds(struct mfc_cache *cache,
                      unsigned char *ttls)
{
    int vifi;

    cache->mfc_un.res.minvif = MAXVIFS;
    cache->mfc_un.res.maxvif = 0;
    memset(cache->mfc_un.res.ttls, 255, MAXVIFS);

    for (vifi=0; vifi<maxvif; vifi++) {
        if (VIF_EXISTS(vifi) &&
            ttls[vifi] && ttls[vifi] < 255) {
            cache->mfc_un.res.ttls[vifi] =
                ttls[vifi];
            if (cache->mfc_un.res.minvif > vifi)
                cache->mfc_un.res.minvif = vifi;
            if (cache->mfc_un.res.maxvif <= vifi)
                cache->mfc_un.res.maxvif = vifi + 1;
        }
    }
}
```

egyszerűen törli a megadott bejegyzést az MFC-ből. Most már tudjuk, hol kerülnek be, hogyan törődnek és módosulnak az MFC-bejegyzések, itt az ideje, hogy a csoportszórásos útválasztókód eltérítésével (hooking) is elkezdjünk foglalkozni. A függvényeltérítés (hooking) meglehetősen egyszerű eljárás. Tulajdonképpen arról van szó, hogyha a korábban bemutatott módon egy függvényt illesztünk a kód középebe, a rendszer a rendszermagmodulban megadott külső függvényre vált. Ezáltal a modulban megadott függvényt visszahívott függvénynek (callback) tekinthetjük, ami minden esetben meghívódik, valahányszor MFC-bejegyzés kerül be, módosul vagy törődik. Eltérítő megoldásunk kivitelezését a 2.4.x rendszermagokban eleve megtalálható, jól ismert eltérítő megoldásra, a Netfilterre alapoztuk. Ha esetleg valaki még sose használta, annyit mindenképpen meg kell említeni, hogy népszerű programról van szó, amelyet csomagszűrési feladatok ellátására építettek a rendszermagba. Amit a Netfilter a csomagokkal végez, megtehető tetszőleges adatszerkezetekkel is. A mi esetünkben ez az `mfc_caches` lesz. A visszahívott függvény mintapéldánya a következő:

```
typedef void nf_nfy_msg(
    unsigned int hook,
    unsigned int msgno,
    const struct net_device
    *dev,
    void* moreData);
```

A hook jelen esetben a tartományt jelöli (ez az érték minden esetben a `PF_INET` értéket veszi fel); a `msgno` az üzenet számát jelöli (a rendszermag által végrehajtott művelet, például `mfc_cache`-bejegyzés hozzáadása); a `dev` a műveletben résztvevő bármelyik pillanatnyi `net_device` értékét felveheti; végül a `moreData` egy általános adatszerkezetet célzó, `void*` típusú mutató. Esetünkben ez az adatszerkezet az `mfc_cache` lesz.

Most, hogy már ismerjük a visszahívás formátumát, nézzük meg, miképpen hívja meg a rendszermag. Ez lényegében igen egyszerűen történik; a következő sorokat a rendszermagkódba helyezve az adott művelethez bármilyen bejegyzett eltérítő függvényt meghívhatunk:

```
#ifdef NF_EVT_HOOK && NF_MCAST_HOOK
NF_NFY_HOOK(
    PF_INET,
    NF_NFY_IP_MCAST_MSG,
    IPMR_MFC_ADD,
    NULL,
    (void*) c);
#endif*
```

Az eltérítő függvény szerkezetének részletes ismertetése sajnos meghaladja cikkünk kereteit. Az olvasó tisztább képet alkothat magáról a kódról, ha megtekinti az általunk módosított rendszermagfájlokat (amik megtalálhatók a <http://www.linuxvilag.hu/multi> oldalon), vagy az eredeti Netfilter-megoldást.

Most, hogy végre teljes képet kaphattunk a linuxos csoportszórásos útvonalválasztás megvalósításáról, illetve pár dolgot megjegyeztünk az eltérítő függvények megoldása kapcsán, érdemes pár szót szólni a próbáink során felmerült néhány gondra is.

Térjünk vissza próbahálózatunkhoz, és képzeljük el a következő felállást: a Snoopy csoportszórásos forgalmat küld ki, miközben Mafalda és Vaio egyaránt JOIN-kérelmet továbbít. Azt váránk, hogy új csoportszórásos útvonal jön létre az `eth0` és `eth1` kimenő csatolókkal. Sajnos nem ez fog történni. Ha vetünk egy pillantást a `/proc/net/ip_mr_cache`-re, csak egyetlen útvonalat látunk a Mafaldára, ugyanakkor a csoportszórásos forgalom mégis zavartalanul eléri a Mafaldát és Vaiót egyaránt. Íme az ok: a Snoopyról kimenő csomagok forráscímként a `192.168.1.200`-as számot használják. Ezért aztán amikor az `eth0`-n küld adatot kifelé, a Snoopy úgy fog viselkedni, mintha a csoportszórásos forgalmat a helyi hálózaton bonyolítaná le. Ez azt jelenti, hogy Snoopy még azelőtt elkezd csomagokat küldeni az `eth0`-n, mielőtt a Vaio kiadná a JOIN-kérelmet, hiszen a rendszermag nem képes kihasználni a PIM démon azon képességét, amivel a más gépekről érkező IGMP-csomagokat is értelmezni tudja. Hogy a többi munkaálomlás is megkapja a csoportszórásos forgalmat, egyszerűen kiküldi a csomagokat. Ebből következik, hogy az ugyanezen a hálózati szelvényen található bármely más gép – csoportszórásos hálózati kártyába épített szűrőjét bekapcsolva – a kívánt adatot leszedheti a drótról. Ugyanilyen okokból a csoportszórásos JOIN- és LEAVE-kérelmek sem irányíthatók az elsődleges csatolóra, mivel azon a csatolón a rendszermag nem végez pontos csoportszórásos útvonalválasztást.

Linux Journal 2002. november, 103. szám



Matteo Pelati

(matteo@dolce.it) tanulmányai mellett kutatói segédmunkatárs a milánói egyetemen. Elsődleges érdeklődési körébe tartoznak a hálózati szabványok, az operációs rendszerek és az utazás.

IPv6-támogatás linuxos hálózati csomópontokon

Ez a cikk a Linux-kiszolgálók IPv6-beállításához és IPv6 alapú internetcsatlakoztatásukhoz nyújt technikai segítséget.

Az IPv4, az IP-protokoll jelenlegi változata bebizonyította, hogy hatékony, könnyen megvalósítható, széles körben használható, és kiállta a jelenkori Internet méretének próbáját, ahol nagyrészt még mindig IPv4 rendszert használnak, holott már közel húszéves módszer. Az IPv4 meglelt korának ellenére figyelemre méltóan rugalmas volt, de azért manapság már kezdenek gondok jelentkezni. Az IPv4 eredeti tervénél nem vettek figyelembe néhány, manapság már igen lényegesnek számító szempontot. Ezek közé tartozik például a nagy címtartomány kezelése, amely megoldást jelent a címbeszűkülés kérdésre, a hordozhatóság, a biztonság, az önműködő beállítási megoldások nyújtása, illetve a szolgáltatás minősége.

Hogy megoldást találjon e gondokra, az Internet Engineering Task Force (IETF) kifejlesztett egy protokoll- és szabványkészletet, amelyet IP version 6 (azaz IPv6) néven ismerünk, s amely már tartalmazza az IPv4 továbbfejlesztéséhez szükséges elképzeléseket és ajánlott módszereket. Az IPv6 képességei közé tartozik az új fejlécformátum, a nagyobb (128-bites) címtartomány, a hatékony és hierarchikus címzési és útkeresési háttér, az állapotfüggetlen (stateless) és állapotfüggő (statefull) címbiztonság, beépített biztonság, jobb hordozhatóságtámogatás, illetve egy új protokoll a szomszédos csomópontok kapcsolattartásához.

Az IPv6 esetében az állapotfüggetlenség és -függőség jelentése a következő:

- Az állapotfüggő önműködő beállítási módszer esetén befűzött csomóponti számítógép az összes szükséges adatot (cím, beállítási adatok, kapcsolók stb.) egy kiszolgálótól (annak adatbázisából) kapja meg. Az adatok szigorú ellenőrzés alatt vannak, s a vezérlést például a DHCPv6 protokoll végezheti.
- Az állapotfüggetlen önműködő beállítás esetén nem szükséges az előbbi feltétel. A befűzött számítógép a befűzéséhez szükséges adatokat a hálózatból nyeri ki (üzenetszórás segítségével). Az IPv6-címek csak egy meghatározott ideig érvényesek, azt követően más csomópontnak lesznek kiosztva. Látható, hogy az IPv6 nem csupán az IPv4 gondjait küszöböli ki, hanem sok újdonságot is bevezet. Az IPv6 várhatóan fokozatosan váltja fel az IPv4 protokollt, s a váltás során jó pár évig egymás mellett léteznek majd.

Linuxos IPv6-megvalósítások

Linux alá két fő IPv6 fejlesztési vonal létezik: a Linux-rendszermag részeként érkező megvalósítások, illetve az USAGI (UniverSAl playGround for IPv6, magyarul IPv6 egyetemleges játszótér) megoldások. Az USAGI Projekt azon munkálkodik, hogy termelési minőségű IPv6 protokollt állítson elő Linux alá, szorosan együttműködve a WIDE, KAME és TAHI projektekkel. Különböző szervezetekből érkező jelentkezők készítik, amelyek a IPv6-protokollverem létrehozásával járulnak hozzá a Linux és a IPv6-közösség munkájához. Jelenleg több különböző fejlesztés is folyik különféle terjesztési csoportokban; a USAGI ezeket a csoportokat igyekszik egyesíteni, hogy végül

minden Linux-terjesztésbe egyetlen, egységes IPv6-megvalósítás kerüljön.

E cikkhez a Linux-rendszermag <http://kernel.org>-ról letöltött 2.4.5 változatát használtuk, jelenleg azonban a 2.4.19-es rendszermag a legfrissebb. Először is megmutatjuk, hogyan lehet IPv6-támogatással lefordítani a rendszermagot, majd megnézzük, miképpen lehet az alapvető hálózati programokat rábírní az IPv6-támogatására, végül azt is megtudhatjuk, hogyan kapcsolhatjuk rá IPv6-támogatású kiszolgálónkat a IPv6 Internetre, a <http://www.freenet6.net> projekt szolgáltatását használva.

IPv6-támogatás a Linux-rendszermagban

Első lépésünk a Linux-rendszermag letöltése a

<http://kernel.org>-ról, majd az állomány kicsomagolása:

```
tar -xzf linux-2.4.19.tar.gz
```

Ezáltal létrejött egy *linux* nevű könyvtárunk. Mozgassuk ezt a könyvtárat a */usr/src* könyvtár alá *linux-2.4.19* néven, jelezve a rendszermag változatszámát. Ezután készítenünk kell egy hivatkozást a 2.4.19 forráskönyvtárra:

```
ln -s /usr/src/linux-2.4.19 /usr/src/linux
```

Ha ezzel megvagyunk, az új rendszermagban be kell állítanunk az IPv6-támogatást:

```
cd /usr/src/linux
make xconfig
```

vagy karakteres felületen használhatjuk a `make menuconfig` parancsot.

A rendszermag-beállítás során ehhez két lehetőséget kell bekapcsolnunk. Először is lépünk a *Code Maturity Level* menüpontra, és kapcsoljuk be a *development/incomplete code/drivers* pontot:

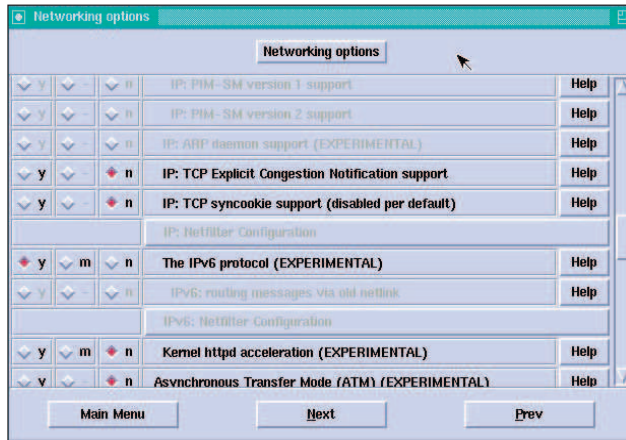
```
"Prompt for development and/or incomplete
code/drivers" YES
```

Majd menjünk a *Networking Options* menübe, itt most be tudjuk kapcsolni az IPv6-protokollt:

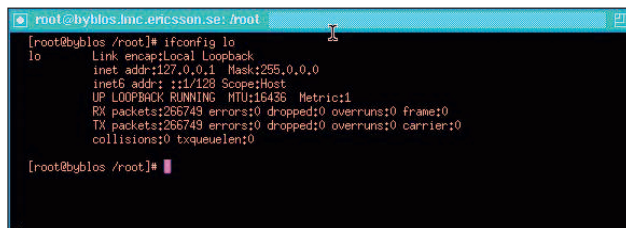
```
IPv6 Protocol (EXPERIMENTAL) YES
```

Rendszermagszinten mindössze ennyi beállításra volt szükségünk. Ezután beállításainkat menthetjük, majd a *Save and Exit* gombra kattintva kiléphetünk. Ezáltal a */usr/src/linux* könyvtárban létrejön egy *.config* nevű állomány, amely tulajdonképpen a rendszermag beállításfájlja. Most már készen állunk a rendszermag lefordítására, ehhez a következő lépéseket kell megtennünk:

```
make clean; make dep; make bzImage
```



1. kép A Linux-rendszermag IPv6-beállítás



2. kép A hurokeszköz

Az eredmény egy új rendszermagpéldány lesz a `/usr/src/linux/arch/i386/boot/` könyvtár alatt. Ha a további képességeket esetleg modulként állítottunk be, a modulokat is le kell fordítani, illetve telepíteni:

```
make modules; make modules_install
```

Ezek után az új, IPv6-támogatással rendelkező rendszermagunkat át kell másolnunk a `/boot` könyvtár alá:

```
cp /usr/src/linux/arch/i386/boot/bzImage
  ↪ /boot/bzImage.ipv6
```

és frissítenünk kell `System.map` fájlunkat:

```
cp /usr/src/linux/System.map
  ↪ /boot/System.map-2.4.5-ipv6
ln -fs /boot/System.map-2.4.5-ipv6
  ↪ /boot/System.map
```

Egyetlen lépés maradt csak hátra: frissítenünk kell a `/etc/lilo.conf` fájlt, hozzáadva az új IPv6-támogatású rendszermagbejegyzést. Nyissuk meg a `/etc/lilo.conf` fájlt, és adjuk hozzá a következő bejegyzést:

```
image=/boot/bzImage.ipv6
label=linux_ipv6
root=/dev/hda1 # ezt cserölj k ki a saját
               # lemezrösz nkre
read-only
```

Végül frissítsük a LILO-beállításokat a következő paranccsal:

```
/sbin/lilo
```

Észítsük a `linux_ipv6` bejegyzés rendszerindításkor bekerül LILO-beállításaink közé. Most már készen állunk rá, hogy újraindítsuk a kiszolgálót. Amikor a LILO feljön, válasszuk a `linux_ipv6` indítást (a LILO nem feltétlenül a menüvel jön fel, ilyenkor egy CTRL vagy ALT lenyomása szükséges a LILO felirat megjelenésakor). Et voilà! Rendszerünk rendszermag-szintű IPv6-támogatással indul. Ellenőrzésképpen gépeljük be az `ifconfig` parancsot. Hurokeszköz-beállításunkban az IPv6 típusú `::1` helyi címet kell mutatnia (lásd a 2. képet).

IPv6-programok és -eszközök

Ha a rendszermag már támogatja az IPv6-ot, feltelepíthetünk néhány eszközt beállításaink kipróbálására, illetve a más rendszerekkel történő IPv6-átvitel kiépítésére. Először is szükségünk lesz az IPv6-támogatással bíró alaphálózati eszközökre, hogy beállíthassuk a csatolófelületet, illetve néhány más olyan IP-eszközre, mint a `ping6`, `telnet6` stb., végül képesnek kell lennünk néhány alapvető IPv6-próba végrehajtására.

Ezeket az eszközöket három csomagban érhetjük el: `net-tools`, `iputils` és `NetKit`-eszközök. Az első kettő feltétlenül szükséges, a `NetKit` elhagyható. Továbbá, ha az esetleg felmerülő lehetséges IPv6 hálózati hibákat fel szeretnénk tudni deríteni, szükségünk lesz az IPv6-támogatású `tcpdump` és `libpcap` csomagokra is. A következő részekben a fent említett csomagok telepítésével foglalkozunk.

net-tools

A `net-tools` a linuxos hálózatkezelés irányítására szánt programcsomag, amelyben olyan programokat találunk, mint az `arp`, a `hostname` (`domainname`, `nsdomainname`, `nisdomainname`), az `ifconfig`, az `ipmaddr`, az `iptunnel`, a `netstat`, a `rarp`, a `route` és a `plipconfig`. Ezt a csomagot a <http://www.tazenda.demon.co.uk/phil/net-tools> címen lelhetjük fel.

A csomag telepítéséhez először is töltsük le a csomagokat közvetlenül a `/usr/src` könyvtárba, majd gépeljük be egyenként a következő parancsokat:

```
cd /usr/src
tar xIvf net-tools-1.60.tar.bz2
cd net-tools-1.60
./configure.sh config.in
```

A beállítás során néhány kérdésre kell válaszolnunk, hogy beállíthassuk a `net-tools`-t. Azt javasolom, a következő kérdésekre válaszoljunk igennel (y):

```
INET6 (IPv6) protocol family (HAVE_AF_INET6)
  ↪ [n] y
SIT (IPv6-in-IPv4) support (HAVE_HWSIT)
  ↪ [n] y
Build iptunnel and ipmaddr (HAVE_IP_TOOLS)
  ↪ [n] y
```

Ezek után le kell fordítanunk és telepítenünk kell a `net-tools` csomagot:

```
make
make install
```

A programok a `/sbin` és `/bin` könyvtárakba kerülnek, és máris használhatjuk őket, feltéve, hogy a gépet az IPv6-támogatású rendszermaggal már újraindítottuk.

iputils

Ez a csomag a következő eszközöket tartalmazza: ping, ping6, traceroute6, rdisc, clockdiff, tftpd, tracepath, tracepath6 és arping. Elképzelhető, hogy a rendszerünkön telepített IP-eszközök nem támogatják az IPv6-ot. Az első lépés annak vizsgálata, hogy jelenlegi változatunk vajon Ipv6-támogatású-e. Gépeljük be a következő utasítást a parancssorba:

```
rpm -q --qf "%{NAME}-%{VERSION}\n" iputils
```

(Debian alatt a fentebbi programok külön net-tools, traceroute és netkit-ping nevű csomagokban vannak). Ha a kimenet iputils-20000121 vagy ennél frissebb, akkor nem kell a legújabb iputils csomagra frissítenünk. Ha nem ez a helyzet, az eszközök feltelepítéséhez követnünk kell a következő lépéseket. Először is töltsük le a csomagot a <http://ftp.inr.ac.ru/ip-routing> címről. Mi az iputils-ss001110.tar.gz csomagot használtuk. Ezután csomagoljuk ki a */usr/src* könyvtárba:

```
tar -xzf iputils-ss001110.tar.gz
```

Végül fordítsuk le az iputils-t a make paranccsal. A csomag nem támogatja a make install utasítást, ezért aztán az elkészült programokat tetszés szerinti helyre tehetjük. Természetesen nem árt, ha meggyőződünk róla, hogy elérési útvonalunkban az eszközök régi változata nem ütközik-e az újabbakkal. Mentsük tehát az új programokat a */usr/local/iputils/bin/* könyvtárba. A csomag minden egyes eszközhöz egy-egy súgóoldalt is szolgáltat. Ezeket olyan könyvtárba kell helyezni, ahol a man parancs megtalálja őket; ha kíváncsiak vagyunk, milyen könyvtárak szerepelnek a man elérési útjában, gépeljük be a manpath parancsot.

NetKit-eszközök

Ezekre az alapvető eszközökre lesz szükségünk új IPv6-hálózatunk működtetéséhez és kipróbálásához. A NetKit a következő eszközöket tartalmazza: ping, finger, telnet, rwho és az ezeknek megfelelő démonok. Ezek igen hasznos eszközök, hiszen IPv6-támogatással is le tudjuk őket fordítani. A csomagot a következő címről tölthetjük le: <http://freshmeat.net/projects/netkit>. Mi a nkit-0.5.1.tar.gz változatot próbáltuk ki. A következőkben azt mutatjuk be, milyen lépéseket kell megtenni az eszközök Linux-kiszolgálóra való telepítéséhez. Először is töltsük le a legfrissebb NetKit-csomagot a fenn megadott honlapról. Azután a letöltött fájlt helyezük a */usr/src* könyvtárba. Bontsuk ki a csomagot a következő paranccsal:

```
tar -xzf nkit-0.5.1.tar.gz
```

Ezután adjuk ki a ./configure utasítást. Fordítsunk a make clean és make parancsokkal, majd a programokat másoljuk a */usr/local/bin* könyvtárba:

```
cp telnet/telnet /usr/local/bin/telnet6
cp telnetd/in.telnetd
  => /usr/local/sbin/in.telnetd6
cp finger/finger /usr/local/bin/finger6
cp ping/ping /usr/local/bin/ping6
cp fingerd/in.fingerd
  => /usr/local/sbin/in.fingerd6
```

Mostantól rendelkezünk az alapvető szolgáltatásokkal, így például helyi IPv6 visszacsatolt (loopback) eszközünkön kipróbálhatjuk a ping6 parancsot. Figyelem, a Red Hat 7.x-felhasználóknak a NetKit-csomagot foltozniuk kell! A folt a <http://ftp.bieringer.de/pub/linux/IPv6/netkit> címen érhető el.

Választható eszközök

Van néhány választható eszköz is, amit feltelepíthetünk a rendszerünkre, ha képességeit IPv6-protokollra is ki szeretnénk terjeszteni. Ebben a cikkben csak három ilyen csomagot említünk meg: libpcap, tcpdump és xinetd.

A libpcap és a tcpdump

Ha meg szeretnénk érteni, mi történik csomagszinten az IPv6-hálózatunkban, a libpcap és tcpdump csomagokat is fel kell készítenünk az IPv6-ra. A libpcap egy olyan rendszerfüggetlen felület felhasználói szintű csomagok elfogására, ami tulajdonképpen hordozható keretrendszer nyújt az alacsony szintű hálózatfigyeléshez. A másik a tcpdump nevű eszköz, ami hálózatfigyelési és adatgyűjtési feladatokat lát el. Amennyiben ilyen szolgáltatásokat szeretnénk, előbb le kell töltenünk a legfrissebb változatokat, majd telepítenünk szükséges őket. Az általunk kipróbált változatok a tcpdump 3.6.2 és a libpcap 0.6.2 voltak. Először is töltsük le a csomagokat a <http://www.tcpdump.org> oldalról, majd másoljuk a */usr/src*-be. Csomagoljuk ki a megfelelő parancsokkal:

```
tar -xzf libpcap-0.6.2.tar.gz
tar -xzf tcpdump-3.6.2.tar.gz
```

A kicsomagolást követően két könyvtárat kapunk, csomagonként egyet-egyet. Ezután mind a két csomag esetében ugyanazokat a lépéseket kell megtennünk; igaz, elsőként a libpcap csomagon kell végrehajtanunk és csak azután a tcpdump-on. Először is futtassuk le a beállító parancsfájlt IPv6-támogatás engedélyezéssel:

```
./configure --enable-ipv6
```

Majd fordítsuk le a csomagot a make clean és make parancsokkal. Végül a make install paranccsal telepítsük a programokat.

Következő lépésként módosítanunk kell az elérési utunkat, hogy az IPv6-támogatású programokat is elérhessük. Esetleg a */etc/profile* fájlt is átszerkeszthetjük, hogy a PATH környezeti változó tartalmazza a */local/sbin* és */usr/local/bin* könyvtárakat, majd a változtatások érvényesítéséhez töltsük újra a */etc/profile* fájlt:

```
source /etc/profile
```

A xinetd IPv6-támogatással

Ha telnet6 alatt is el szeretnénk érni a gépünket, a xinetd-t is le kell fordítanunk inet6-támogatással. Alap esetben a feltelepített démon nem képes IPv6-címeket kezelni. Ezért aztán kénytelenek vagyunk xinetd-re frissíteni. A xinetd legfrissebb változatát a <http://synack.net/xinetd> címen találjuk. A mi összeállításunkban xinetd-2.1.8.8p3.tar.gz szerepelt. Töltsük le a xinetd-2.1.8.8p3.tar.gz (vagy a legújabb) csomagot a */usr/src* könyvtárba, majd csomagoljuk ki:

```
tar -xzf xinetd-2.1.8.8p3.tar.gz
```

Futtassuk a beállító parancsfájlt:

```
./configure --with-inet6 --prefix=/usr/local/bin
```

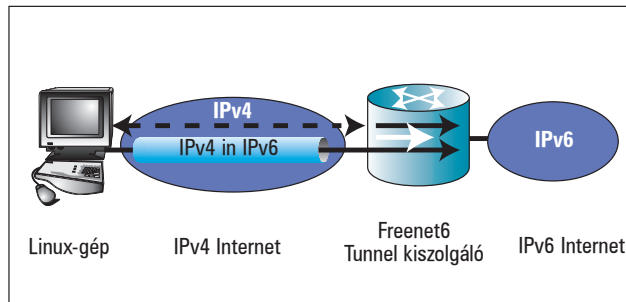
A `--prefix=/usr/local/bin` kapcsoló azt jelzi, hogy a létrejövő programok a `/usr/local/bin` könyvtár alá kerülnek. Ezután a fordítás és a telepítés következik:

```
make clean
make
make install
```

A következő lépés a régi `inetd.conf` átalakítása az új beállításfájl formátumára:

```
/usr/sbin/xconv.pl < /etc/inetd.conf >
  => /etc/xinetd.conf
```

ahol a `/usr/sbin` a `xinetd` program könyvtárára mutató elérési út. Zárójelben jegyzem meg, nem árt meggyőződni róla, hogy a `xconv.pl` parancsfájl első sora valóban a helyes Perl programra mutat-e, illetve végre tudja-e hajtani egyáltalán. Ezután néhány igazán apró módosítást kell csak végezni a `/etc/xinetd.conf` állományban, jelezve, hogy `telnet6d`-t és `tfptp6d`-t használunk a szokásos IPv4 Telnet- és TFTP-



1. ábra Freenet6-szerkezet egyetlen gazdagéppel

démonok helyett. A fentiek elvégzése után rendszerünk Telnet és FTP szolgáltatása IPv6 rendszer alá került.

IPv6-alkalmazások

Számos alkalmazás létezik, ami támogatja az IPv6-ot. Ennek ellenére mi most csak egyetlen kiszolgálóalkalmazást említünk meg: az Apache webkiszolgálót. Az Apache az Internet legkedveltebb webkiszolgálója (forrás: <http://netcraft.com>). A legfrissebb kiadások már tartalmazzák az IPv6-támogatást, s ez alkalmassá teszi arra, hogy kipróbálhassuk rajta IPv6-beállításainkat. Miután letöltöttük és telepítettük az Apache webkiszolgáló legfrissebb változatát, IPv6 alatt is szolgáltatathatunk honlapokat. A kényelem kedvéért a `/etc/hosts` fájlt átszerkeszthetjük, hogy tartalmazza a következő sort:

```
:::1 ip6-localhost ip6-localhost
```

Ezek után a `:::1` használata helyett az `ip6-localhost` nevet is használhatjuk. Ne feledjük el ellenőrizni a `/etc/protocols`-t. Ha az alább említett bejegyzések itt nem találhatók meg, be kell őket szúrunk az IPv6-protokolltámogatási részhez:

```
ipv6 41 IPv6 # IPv6
ipv6-route 43 IPv6-Route # etvÆlasztÆsi fejlœc
# az IPv6-hoz
ipv6-frag 44 IPv6-Frag # t redetzetsœgi fejlœc
# az IPv6-hoz
ipv6-crypt 50 IPv6-Crypt # titoktartÆsi fejlœc
# az IPv6-hoz
ipv6-auth 51 IPv6-Auth # azonos tÆsi fejlœc
# az IPv6-hoz
ipv6-icmp 58 IPv6-ICMP icmpv6 icmp6M
# ICMP az IPv6-hoz
ipv6-nonxt 59 IPv6-NoNxt # nincs k vetkezi
# fejlœc az IPv6-hoz
ipv6-opts 60 IPv6-Opts # CœlbeÆll tÆsok
# az IPv6-hoz
```

Csatlakozás az IPv6 Internethez

Még 1996-ban, amikor az első IPv6 IETF szabvány elkészült, nagy érdeklődés mutatkozott egy IPv6-gerinc létrehozása iránt. Az 1996-os IETF-Montreal alatt megszületett a 6bone (IPv6 gerinc). Ez a kipróbálásra szánt (de még mindig érvényes) `3ffe::/16` tartományt használja. Kezdetben a gerinc nagy része a jelenlegi IPv4 Interneten, alagutakon keresztül működött. Ezáltal egy virtuális IPv6-hálózat jött létre az IPv4 Interneten keresztül. Manapság a 6bone egyaránt tartalmaz saját és alagutakon futó kapcsolatokat. A 6bone kipróbálásra készült, így aztán nincs is szolgáltatásszintű egyezmény az egyes szervezetek közt, természetesen ez nem azt jelenti, hogy megbízhatósági vagy hitelességi gondokkal küszködne. Minden, a `3ffe::/16` tartományon keresztülhaladó forgalom korlátozás nélkül hiteles.

Akár a 6bone-hoz, akár az IPv6 Internethez szeretnénk kapcsolódni, szükségünk lesz (akárcsak IPv4 alatt) egy szolgáltatóra. Ha közvetlenül illet nem találunk, vagy ha a jelenlegi ezt a szolgáltatást nem nyújtja, a legegyszerűbb megoldás csövezeteket készíteni egy olyan szolgáltatóhoz vagy géphez, ami hajlandó továbbító szolgáltatást felajánlani nekünk.

1999 júliusában a három helyi nyilvántartó, az amerikai ARIN, az európai és afrikai RIPE és az ázsiai illetőségű APNIC elkezdett szabályos, `2001::/16` tartományban kezdődő nem próbacímeket kiadni a szolgáltatóknak. Azok a gépek, amelyek ebben a tartományban kaptak címet, alkotják ma a termelési IPv6 Internetet.

Akárcsak az Internet korai napjaiban, elindult egy projekt, aminek a célja az volt, hogy segítse az embereket elindulni az IPv6 világában, felajánlva nekik az ingyenes és önműködővé varázsolt csövezeték-szolgáltatást, ami bármilyen személyt vagy szervezetet az IPv6 Internethez tud kapcsolni. A Viagénie tanácsadó cég által vezetett projekt neve Freenet6.net, ami ingyenesen és jelentkezéses, illetve „run-on-a-best-effort” alapon működik. A szolgáltatás igen népszerű a közösségen belül, mivel könnyű és gyors elérést nyújt az Internethez. A Freenet6-ot a tunnel broker (RFC 3053) után mintázták, ahol a csomópont és a tunnel broker között IPv6-over-IPv4 alagút jön létre. A Freenet6 ennek a módszernek egy továbbfejlesztett változata, ahol a csomópont a kiszolgálóval való egyeztetéshez Tunnel Setup Protocolt (TSP) használ. Az ügyfélcso mpont lehet gazdagép vagy útvonalválasztó is. A Freenet6 TSP-kiszolgálója nemcsak csatornákat, hanem nagy címtartományokat is nyújthat bármilyen felhasználói szolgáltatásnak. A kiadott címtér /48 alakú, amely (16 biten) 65 536 alhálózatot tartalmazhat, melyek mindegyike 264 csomóponttal rendelkezhet (64 bit). Ez jóval több, mint a teljes mai Internet! Ez a címtér

© Kiskapu Kft. Minden jog fenntartva

a felhasználóhoz van rendelve, így az ügyfél gép IPv4-címének változása esetén is megmarad. A felhasználók és szervezetek számára óriási szabadságot nyújt, hogy címek millióit rendelhetik a kiszolgálókhoz és a szolgáltatókhoz – ezt egyáltalán nem volt könnyű megvalósítani az IPv4 alatti NAT-tal.

Az IPv6-over-IPv4 csatorna felépítéséhez mindkét oldali IPv4 és a másik oldalon IPv6 cím-beállításokat igényel. Amikor valamelyik oldal megváltoztatja az IPv4-címét, ennek megfelelően a csatorna mindkét oldalán meg kell változtatni a beállításokat. Ez különösen akkor nehézkes, amikor az IPv4-csomópont betárcsázásos alapon működik, vagy gyakran változtat címet. A Freenet6 szolgáltatásban megvalósított TSP beállítható úgy, hogy figyeljen ezekre a dolgokra.

Valahányszor a csatorna ügyfele megváltoztatja az IPv4-címét, például rendszerindításkor a DHCP-szolgáltatás segítségével, a TSP-ügyfél elküldi a módosított és hitelesített adatot a kiszolgálónak, így a csatorna továbbra is működőképes marad. A Freenet6 ügyfélszolgáltatás által támogatott rendszerek: Linux, FreeBSD, OpenBSD, NetBSD, Windows, Solaris és Cisco. Az 1. ábra a Freenet6 alapszerkezetét mutatja be.

Az IPv6 telepítését követően Linux alatt a következő lépéseket kell megtennünk, hogy használhassuk a Freenet6 szolgáltatást. Először is lépünk fel a <http://www.freenet6.net> oldalra, és jegyezzük be magunknak egy felhasználói nevet. Töltsük le a linuxos TSP-ügyfelet! Kövessük a fordításra és telepítésre vonatkozó utasításokat. Ezután állítsuk be a mellékelt *tspc.conf* fájlt. Írjuk be a felhasználói nevet és a jelszót, végül indítsuk el a TSPC-ügyfelet:

```
tspc -vf tspc.conf
```

A *tspc* ügyfélparancsot berakhatjuk a rendszerindító folyamatba is, így a csatorna minden egyes rendszerindítás után önműködően újra felépül, még ha az IPv4-címünk meg is változott.

A Freenet6 egyaránt adhat nekünk egyetlen IPv6-címet (ha géphez kérjük), vagy egy teljes /48 tartományt (amennyiben útválasztóhoz kérjük). A Freenet6 úgy állítja be a Linuxot, hogy megfeleljen a feladatnak.

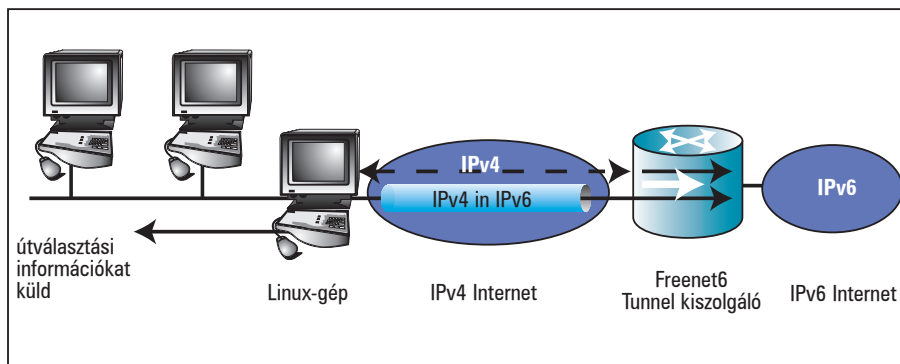
Útvonalválasztó esetében /48 formájú címet kapunk, ahol az útvonalválasztónk első alhálózatának beállítása útvonalválasztó üzenetszórásos típusú lesz. Ez azt jelenti, hogy az ezen az alhálózaton található gépek megkapják az előtagot (prefix), és önműködően beállíthatják magukat, ahogyan azt a 2. ábrán láthatjuk.

Az IPv6 Internethez sokféle módon kapcsolódhatunk. A Freenet6 a TSP-protokollal együtt könnyű, IPv6-csatornán alapuló kapcsolatot nyújt nekünk, állandó címtérrel, ahol ha IPv4-címünk meg is változik, IPv6-címünk és a kapcsolatunk változatlan marad.

Összefoglalás

Tevékenységünk részeként az Ericsson Research Canada Open Architecture Research részlegénél néhány IPv6-tal kapcsolatos projektet futtatunk, többek között foglalkoztunk telecom-fokozatú kiszolgáló-csomópontjaink IPv6-támogatásával, alkalmazás-kiszolgálók IPv6 alapú működésének megvalósításával, illetve kutatási projekteket hozunk létre különféle IPv6-területeken.

Az egyik ilyen érdekes művelet a jelenleg elérhető Linux IPv6-megvalósítások kipróbálása volt, illetve annak eldöntése, hogy melyik változatot érdemes Linux-gépeinkre átültetni. A javaslatokat IPv6-megvalósítások karakterisztikája alapján tettük,



2. ábra Freenet6-szerkezet útválasztóval és több géppel

ideértve a fejlesztés ütemét, a szabványoknak való megfelelést, illetve a teljesítményüket más megvalósításokhoz képest.

Az eredményeket az ottawai Linux Symposiumon mutattuk be Kanadában (http://www.linuxsymposium.org/2002/view_txt.php?text=abstract&talk=93).

2002-ben tovább folytatjuk IPv6-kutatásainkat Linux-telepeinken. Jelenleg erőfeszítéseinket IPv6 alapú linuxos SIP- és SCTP-megvalósításokra összpontosítjuk, illetve szeretnénk egy IPv6 teljesítménymérő környezetet felállítani, amivel meghatározhatjuk az egyes IPv6 alatt futó gépek és alkalmazások teljesítményét és méretezhetőségét.

Végezetül rámutatnék arra, hogy az IPv6 legfőbb ereje a címekben rejlik. Bárki kaphat egy akkora tartományt, mint a teljes jelenlegi IPv4-címtér. A felhasználók ezáltal tökéletes szabadságot kapnak a kiszolgálók és szolgáltatások telepítése terén, anélkül, hogy NAT-gondokkal vagy a korlátozott címtérrel kellene küszködniük. Üdvözöllek az internetes szolgáltatások telepítésének szabad világában!

Köszönetnyilvánítás

Szeretnénk köszönetet mondani az Ericsson Open Architecture Researchnek e cikk megjelenésének támogatásáért, a Canarie, Inc.-nek (<http://www.canarie.ca>) a Freenet6 Projekt támogatásáért, illetve David Gordon-nak (David.Gordon@Ericsson.ca) azért a segítségért, amit a ECUR Lab IPv6 csomópontjainak beállításában nyújtott, illetve a cikk szerkesztéséért és átnézéséért.

Linux Journal 2002. augusztus, 100. szám



Ibrahim Haddad
(Ibrahim.Haddad@Ericsson.com) jelenleg kutatóként dolgozik a montreali Ericsson Corporate Research Unit részlegénél Kanadában.



Marc Blanchet
Viágénie-ben dolgozik egy IP, IPv6 módszerekre és hálózati biztonságra szakosodott tanácsadó cégnél. 1995 óta foglalkozik az IPv6-tal és számos IETF dokumentumot írt már a témában.

Pózoljunk Linux alatt! (2. rész)

Emulátorokról szóló sorozatunk előző részében életre keltettünk egy Palm-gépet. Most részletesebben is megismerkedhetünk a programok kínálatával és beszerzésével.

Mint minden népszerű dolognak, a Palm alapú gépeknek is népes internetes tábora van, ebből következően számtalan ezzel foglalkozó weboldala található a Neten. Nagyon sok oldalt átböngészve az alábbi hivatkozásokat illesztettem könyvjelzőim közé:

- ➔ <http://www.tealpoint.com>
- ➔ <http://pda.tucows.com/palm.html>
- ➔ <http://www.cesinc.com/>
- ➔ <http://www.palmgear.com/software/index.cfm>

A programok

A programok tárháza meglehetősen nagy, akad kereskedelmi termék és szabadon felhasználható, ingyenes, illetve időkorlátos stb. program is. Néha feltört vagy törvénytelenül terjesztett program is felbukkan, ezek használatától óva intenek minden-

kit! A fentebbi címek közül válogatva szinte bármilyen feladatra alkalmas és bármilyen típusú programot (ingyenes, fizetős...) megtalálhatunk.

Az első programtípus, amit egy ilyen kis gépen én nagyon kedvelek, a játék. Nézzük, mik közül választhatunk!

Aki szeret sakkozni, annak a PocketChess (tm) Deluxe 2.5 programot ajánlanám. Mindent tud, amit egy sakkprogramnak tudnia kell: játszhatunk a gép ellen, vagy akár hűsvér ellenfelet is választhatunk.

A lövöldözni vágyóknak a Galax v2.2b programot javaslom, ez a Linux alatt is elérhető xgalaga program „kicsinyített változata”.

Nagyon kellemes program a Liberty, aminek segítségével a Game Boyra készült játékokat futtathatjuk PalmOS alatt. Lehet, hogy ezt sokan egy kicsit túlzásnak gondolják, vagyis hogy az emulált Palm-gépekben még egy másik rendszert is emuláljunk, de ha valaki tüzetesen ki akarja próbálni ezeket a kis gépeket, erre is lehetősége nyílna.

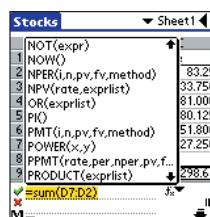
Pasziánsz a javából! Aki a munkahelyi gépe előtt eltöltött kinosan lassan vándorzó órák alatt „véletlenül” függőségi viszonyba került a pasziánsz programmal, az itt is megtalálhatja kedvencét. Így már nincs elvesztegetett idő: ha beszerez egy Palm-gépet, akkor akár buszon, akár villamoson, vonaton, de még a karácsonyi nagy beszerzési láz hosszasan kígyózó sorai-

ban is hódolhat szenvedélyének a Patience 2.5.2 programmal. A második programtípus, amit valószínűleg sokan használnak: a különféle irodai alkalmazások, szövegszerkesztők, táblázat-



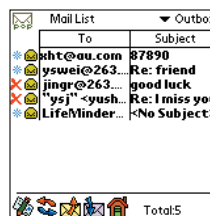
kezelők (már amennyire egy ilyen kis gépen ezek a fogalmak értelmezhetők, ugyanis billentyűzet nélkül szöveget szerkeszteni olyan, mint kormány nélküli autót vezetni).

Ezeknél az irodai alkalmazásoknál nagyon fontos, hogy együttműködjenek a Microsoft Office-szal, hiszen a felhasználók zöme még mindig ezt használja. Ennek a kitételnek a talán legjobb megfelelő program a QuickOffice, illetve ennek nagytestvére, a QuickOffice Pro. Mindkét

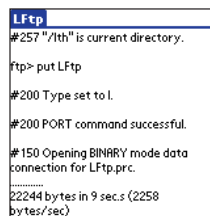


program kereskedelmi termék, az előbbi 39,95 dollárt kóstál, a Pro-változat pedig 49,95 dollárt. Miként a képeken is látható, egy méretes képernyő itt sem árt. ➔ <http://www.cesinc.com> oldalról tölthető le a próbaváltozat.

A harmadik nagy programcsoport, amit körbevizslattam, az internetes alkalmazások (web-, levelező- és FTP-programok).



Böngészni szintén rémálomszerű egy ilyen gépen, néha azonban elkerülhetetlen. Képzeljük el, hogy éppen hegyet mászunk, amikor egyszer csak kíváncsiak leszünk a világ tőzsdéinek pillanatnyi állapotára! Ilyenkor a hegy tetején elővárászoljuk zsebünkéből a kis Palm-gépecskét és a mobiltelefonunkat, remélve, hogy lesz térerő, és rákapcsolódhatunk az Internetre. Majd néhány órát eltöltünk az olvashatatlan oldalak böngészésével. Nos, ha ennyire önkínzóak lennénk, használjuk a PocketLink 2.50b böngészőprogramot.



Levelezőügyfélnek én az AceMail 1.3.5 programot választanám, mivel egyszerű grafikus kezelőfelülettel bír, ezért könnyű használni.

Az FTP protokollt karakteres felületen használóknak pedig az Lftp 1.7.6 program maga lesz a kánaán.

Jó szórakozást!



Csontos Gyula

(Csontos.Gyula@linuxvilag.hu) a Linuxvilág szakmai és CD-szerkesztője. Szabadidejében szívesen mászik hegyet és kerékpározik.

Nálad mi van a falban?

Alex elmeséli, hogyan derítheted ki mindezt néhány drótdarabbal és egy linuxos számítógép hangkártyával.

Szögkeresőt szinte fillérékért lehet venni, kinek jutna hát az eszébe, hogy egy linuxos gépet használjon a falban lévő vezetékek, szögek, csavarok vagy egyébek felderítésére?

A CanDetect Project egy összetettebb munka melléktermékeként jött létre, ahol jóval nehezebb, egy olcsó kis kütyүнél többet igénylő feladatot kellett megoldani. A CanDetect a Corroding Aircraft Non Destructive Evaluation using software Tools and an Eddy Current Tester (Repülőgépek korróziójának károsítás nélküli mérése programok és örvényárammérő műszer segítségével) rövidítése, a tervezet célja pedig az, hogy a repülőgép-szerelőket olcsó, a gépeken fellépő korrózió vizsgálására alkalmas eszközzel lássa el. A CanDetect révén megtekarítható a drága célgépek, külső erősítők, modulátorok és tápegységek ára, a szerelők mégis felderíthetik az apró hibákat a festés vagy más fém alkatrészek alatt is.

A berendezés egy bármely Linux alapú gépen futtatható programból, valamint egy érzékelőből áll, ami a `/dev/dsp` eszközzel közvetlenül csatlakozik – magyarul a hangkártyának megfelelő karaktereszközhöz vezet. Az Embedded Linux Journal NIC (New Internet Computer) versenyére készült egy rendszerindításra is alkalmas CD-ROM lenyomat, ami a célprogramok futtatására használható.

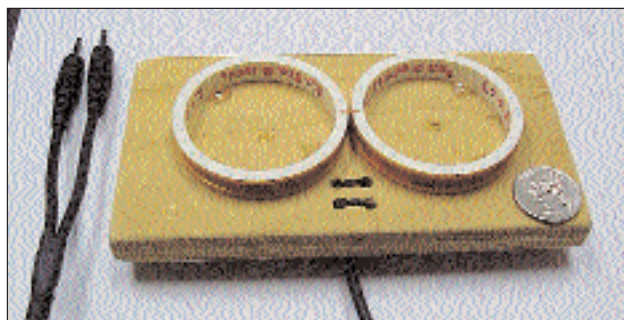
Az érzékelő

Ahol gyakran van földrengés, például Kaliforniában, ott nem biztonságos téglából építkezni. Sokkal célszerűbb erős és könnyű, alumíniumból vagy fából készülő, függőleges tartóelemekből építeni (kb. 40 cm térközzel), a vázra pedig vékony, sima farostlemez, gipsz vagy valamilyen hasonló, csupán önmagát megtartani képes anyag kerül. Földrengéskor viszont az ilyen fal felborítja a mellé állított bútorokat, hacsak nincsenek biztonságosan hozzárögzítve. Amikor kisebb-nagyobb bútorokat erősítünk a falhoz, a csavarokat mindig a belső vázba kell becsavarni, mert csak ez képes elviselni a terhelést. A falakban futnak az elektromos vezetékek és a csövek is, ezeket viszont – különösen hosszabb csavarokkal – ajánlatos elkerülni.

Az 1. képen látható érzékelőt munkatársam, **Brian Whitecotton** készítette. Az ő munkája szebb, mint az enyém. Az érzékelő két tekercsből áll, amelyeknek a közepe 6 centire van egymástól.

A tekercseket a 2. képen látható módon egy falaphoz ragasztottuk. Mindkét tekercs átmérője 5,58 cm, és 80 menetnyi szigetelt drótból állnak. A két tekercs sorba van kötve – az áram az ellenkező irányban halad bennük körbe – a bal és jobb kimenő vezetékek között. A tekercsek közti felezőpont a mono mikrofonbemenetre csatlakozik. A kimenet és a bemenet földje közös, amint az 1. ábrán is látható. Az itt szereplő elrendezés ellenállása 6,2 Ω, induktivitása 0,8 mH, tényleges impedanciája pedig 50 W 10 KHz-en.

Az érzékelő építésekor ügyelni kell arra, hogy az összeköttetések egyezzenek, a két tekercs pedig ugyanolyan legyen. Bármennyire is kézenfekvő lenne, az érzékelő építésekor más fém alkatrészeket nem szabad használni, mivel az érzékelő



1. kép 10 KHz-es frekvenciához és 2 cm-es mélységhez tervezett induktív tekercspar

pont a falban lévő fémek felderítésére készül. Egy ilyen érzékelővel egyébként nem lehet repülőgépeket vizsgálni.

A fal felől vett jelek

A fizika iránt kevésbé érdeklődő Linux-felhasználók időnként megriadnak a tervezetünk mögött álló fizikai törvényektől, holott erre igazából semmi okuk nincs. Ha mást nem is, annyit érdemes megjegyezni, hogy a pontszerű források, például a szegecses kerek jelet adnak, a hosszúkás formájúak pedig inkább pontszerűt.

Ahogy a falat a kívülről láthatatlan szegecses után kutatva átvizsgáljuk, a program pozitív vagy negatív változást jelez, ha a szegecs valamelyik tekercs középpontjába esik (lásd a 2. ábrát).

Ha nagyobb falfelületet kell átvizsgálni, a kapott jel jellegéből megállapítható, hogy az egymás mellett húzott tekercsek melyikével találunk például egy szegecseset. Ha a szegecs pontos helyét is meg akarjuk határozni, a két tekercs egymás mögött halad, így követve, hogy a szegecs mikor kerül a két tekercs közé, ekkor ugyanis nulla jelet kapunk. Itt néhány milliméteres pontossággal dolgozhatunk. Az elektromos kábelek és a fém szerkezeti elemek, amint az az 3. ábrán is megfigyelhető, jóval erőteljesebb jelet adnak.

A jelalak alapján lehet megkülönböztetni egymástól a pontszerű (például egy szegecs feje) és a hosszúkás (például valamilyen vezeték) forrásokat. A pontszerű forrás csaknem ugyanolyan jelet ad, bárhol is van a tekercs hatókörén belül, ennél

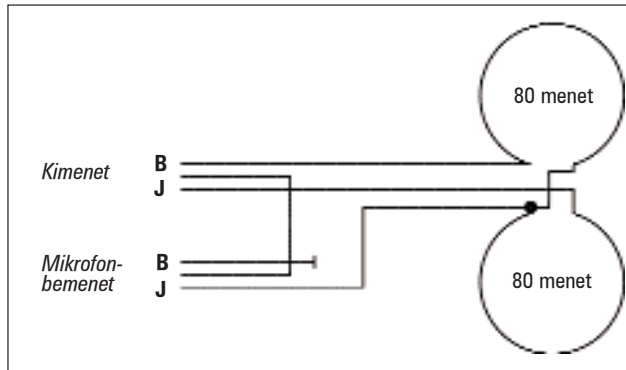


2a

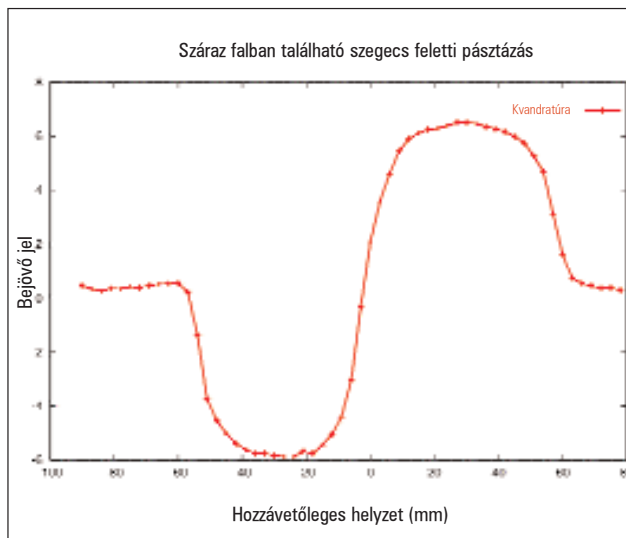


2b

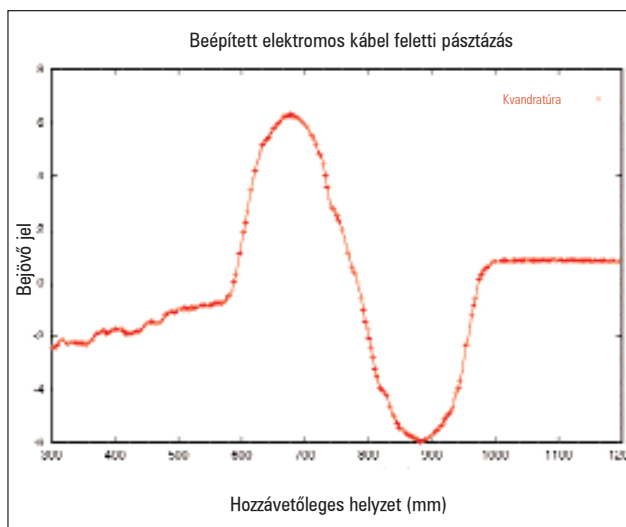
2a, 2b. kép Segítség az 1. ábrán látható érzékelő összeállításához



1. ábra Az 1. képen látható érzékelő kapcsolási rajza

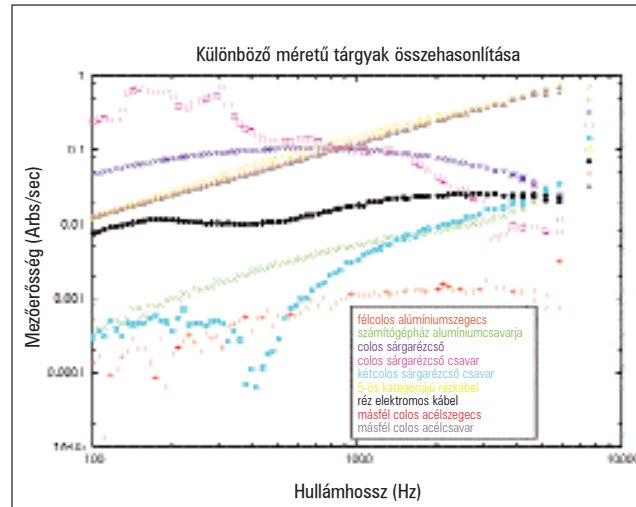


2. ábra Prostar hordozható gép, 1 KHz-es lineáris letapogatás egy száras falban található szegecs felett

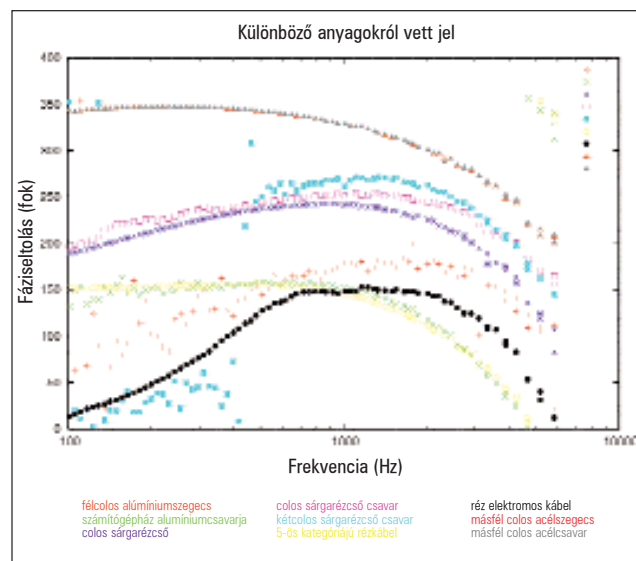


3. ábra Prostar hordozható gép, 1 KHz-es lineáris letapogatás egy 110 voltos elektromos vezeték felett

tehát a jel laposabb jellegű. A hosszúkás forrás jele attól függően változik, hogy mekkora szakasza esik a tekercs hatókörébe, és akkor a legerősebb, ha a forrás pontosan a tekercs közepén



4. ábra A jel mérete utal a tárgy nagyságára. Az alumínium kisebb jelet ad, mint az acél



5. ábra A jel fázisa elárulja, milyen fajtájú fémtárgyat találtunk. A vas és az acél grafikonja magasabb, a réz és az alumíniumé alacsonyabb, a sárgarézt és a hasonló fémeké pedig nagyjából középre kerül. 1 KHz környékén a fázisok jól láthatóan három csoportba tömörülnek

halad keresztül. Ennek jele tehát hegyes, kiugró jellegű. A fémtárgyakat ugyan egy hasonló olcsó keresővel is meg lehet találni, de ha az alapvető helymeghatározáson kívül többre van szükség, finomabb műszert kell beszerezni. Az olcsóbb eszközök arra sem képesek, hogy megkülönböztessék egymástól például az épület vázának elemeit, a vízvezetékeket és az elektromos vezetékeket.

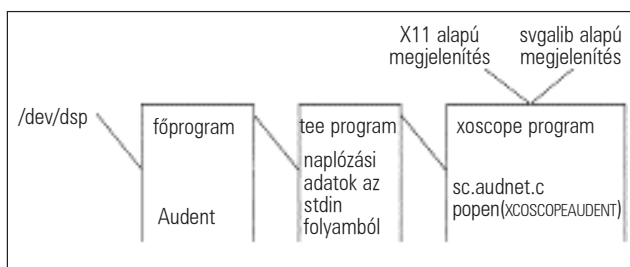
A drágább és nagyobb tudású műszerek apró kijelzőn keresztül jelentik meg a mérések eredményét. Programalapú műszerünk viszont képes kihasználni a nagyobb képernyő adta lehetőségeket, illetve a mérés finomításával a különféle tárgytípusokat is azonosítja. Ha sikerült megtalálni egy tárgyat a falban, a jel változásai alapján állapítható meg, hogy pontosan mire bukkantunk (lásd a 4. ábrát). A jel méretéből megtudható, hogy az adott tárgy mekkora (5. ábra). A jelek hullámhosszváltással módosulnak, ez alapján megbecsülhető a tárgy alakja.

© Kiskapu Kft. Minden jog fenntartva

A számítógépes oldal

A CanDetect Project szerény követelményeket támaszt a gépezemben, nincs különösebb gépigénye. Ha éppen windowsos gépet sikerült kölcsönözni valahonnan, elég egy hajlékonylemezzel vagy CD-ROM-ról elindítani a rendszert. Mi egy olcsó, CD-ről történő indításra alkalmas New Internet Computer (NIC ➔ <http://www.thinknic.com>) gépet használtunk az Embedded Linux Journal versenyén.

Ahhoz, hogy a ki- és bemenő hullámformákat egyszerre lehessen kezelni, a hangkártyának (illetve az alaplapra épített hanglapkának) képesnek kell lennie a teljes kétirányú működésre. Sok régebbi hangkártya és még több jelenlegi hordozható számítógép gyakran csak úgy tud teljes kétirányú módban működni, hogy közben csökkenti a mintavételi felbontást illetve gyakoriságot. A kártyának 16 bites módban kell üzemelnie. A mérések során jobb elkerülni a régebbi vagy rossz hangú kártyákat. A mikrofonbemenetről még a teljes hangerőt beállítva sem szabad hallható sustorgásnak jönnie.



6. ábra A programrészek kapcsolatát szemléltető blokkdiagram

Ha nagyszámú gép közül választhatsz, a Linux Terminal Server Project (LTSP ➔ <http://www.ltsp.org>) segítségével a mérőprogramot inetd alól induló szolgáltatásként is futtathatod.

Miután minden hálózati kártyához elkészítetted a megfelelő EtherBoot (➔ <http://www.etherboot.org>) hajlékonylemezt, egyetlen este alatt több száz gépet is végigpróbálhatsz. A hajlékonylemezzel az összes gépet hálózatról indítsd el, X-terminálról jelentkezz be távoli munkaállomásodra, majd futtass le egy parancsfájlt. A parancsfájlnak fel kell dolgoznia a DISPLAY változót, meg kell állapítania az IP-címet, csatlakoznia kell a számítógépen futó inetd szolgáltatáshoz, le kell futtatnia a tesztet, majd naplóznia kell, és meg kell jelenítenie az eredményt.

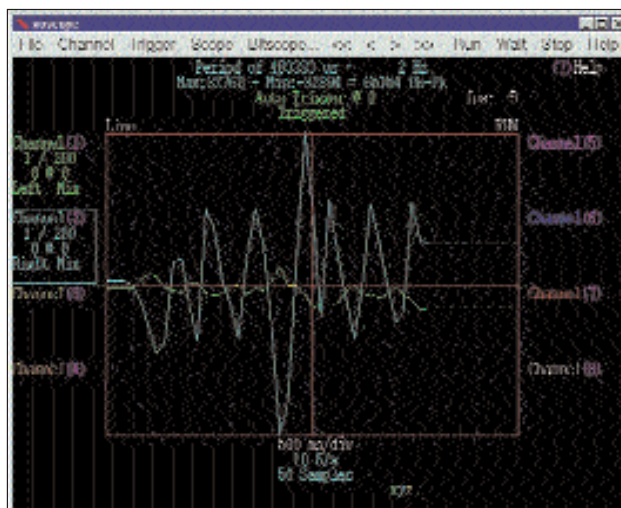
A programok

AudNet fejlesztésünk feladata a hangkártya működtetése. A folyamat megjelenítését az xoscope egy átszabott változata végzi (lásd a 6. ábrát).

Az AudNet feladatának lényege az, hogy a hangkártyán keresztül folyamatosan lejátszik valamit, és a bejövő jeleket is fogadja. A kimenő és bejövő csatornák összhangban tartása nagyon fontos. Általában két bemenő és két kimenő elektromos csatorna van a hangkártyákon, és mindegyik csatornához egy hullámforma tartozik. Minden hullámformaérték a teljes -1...+1 tartományból származhat.

Számos különböző típusú mérési kérés lehet tetszőleges időpontban átadni, amelyek teljesítése a lehető leghamarabb megtörténik. A könyvtár végighalad a feladatok várakozási során, amíg a végére nem ér – már ha a végére ér valaha. Ha a feladatlista kiürül, az utolsó feladatot addig ismételteti, amíg újat nem kap. Minden kérdésben szerepel, hogy a hozzá tartozó hullámformát hány alkalommal kell megismételni.

Mivel a lejátszandó adatokat legalább egy másodperccel a hang-



3. kép Képernyőkép az xoscope ablakáról adatbázis-készítés közben

várakozási sorba kell helyezni, a felvett adatok pedig picivel a tényleges felvétel után jelennek meg a sorban, a könyvtár figyelemmel követi, hogy mennyi adat található a rendszermagnak eme átmeneti táraiban. Az éppen kiküldött kérés gyakran eltér attól, amelynek az adatait a bemenő csatornákon begyűjtjük. Mivel sok elektromágneses rendszer csak kevésbé hajlamos rezonálni, a feladatban azt is meg kell adni, hogy a hullámformát hányszor kell lejátszani, hogy az érzékelő és a minta kellő valószínűséggel nyújtson megbízható eredményt. Ezekre az ugróciklusokra csak akkor van szükség, ha a könyvtár az egyik feladatról egy másik kérésre vált át, feladat ismétlésekor szükségtelenek.

Egyszerű hullámhosszméréshez a példaprogram (main) mindössze két átadott értéket vár: a hullámhossz értékét Hz-ben, és a kimeneti fáziseltolást fokban. A „main” két tiszta, a megadotthoz a lehető legközelebb eső hullámhosszú, a két hangszóró csatornán ellenkező előjellel megjelenő szinuszhullám előállítására utasítja az AudNetet. Ezután az ismétlések számát úgy állítja be, hogy másodpercenként tíz mérési eredményt kapjon.

xoscope – grafikus megjelenítés

Az eredmények megjelenítésére az xoscope módosított változatát használjuk. Egy oszcilloszkóp túlterheltsége esetén nem tudja kirajzolni a bejövő jeleket, ezért elveti őket – az eredeti xoscope is így tesz. A CanDetect számára ez elfogadhatatlan, mivel itt hosszan figyelemmel kell követni egy lassan változó jelet; az xoscope módosításával elhárítottuk ezt a gondot. A Linux-hangkártya illesztőprogramot AudNet alapúra cseréltük le. A NIC-en bármely fájlrendszer-hozzáférés megakadhat, amíg a CD-ROM-meghajtó felpörgésére kell várni, és bármelyik képernyőrajzolásnak addig várakoznia kell, amíg a processzor a gyorsítás nélküli grafikus lapkakészlettel dolgozik. Emiatt az xoscope gyakran több másodpercig nem hívta meg az illesztőprogramunkat. Az időnként öt másodpercet is elérő késleltetés miatt a rendszermag átmeneti tárja betelt, és a mérés kiesett a szinkronból.

Az xoscope másik eszközüllője az XOSCOPEAUDNET környezeti változót olvassa ki, amitől egy 10 Hz mintavételi idejű adatfolyam átadására képes bemenetet várunk. A környezeti változó életre hívása az xoscope indítása előtt történik meg, így a főprogram minden esetben a kívánt beállításokkal indul. Mivel a „main” külön programként fut, fájlokat nem ér el,

a képernyőt nem használja, futását semmi sem akaszthatja meg. Az illesztőprogram arányosítja a lebegőpontos értékeket, így a képernyő egy osztása 1/100 000 részét jelképezi a hangkártya teljes bemeneti tartományának. A teljes tartományból 32 százalékkal kilépő jelek az *xoscope* belső átmeneti tárainak korlátozott dinamikatarományára miatt elvesznek.

Az *xoscope* CVS-en keresztül elérhető módosításait *src* modulunkban folyamatosan követjük. Az *xoscope.patch* fájlt még a beállítások megadása előtt be kell illeszteni. A folt új illesztőprogramokat tartalmaz, illetve néhány alapbeállítást is módosít. A beállító parancsfájl új parancssori kapcsolója, a `--without-vga` letiltja az *svgalib* keresését – a lehetőség különösen akkor hasznos, ha *svgalib*-bel rendelkező gépen végezzük a fordítást, de a program futtatását ilyenekkel nem rendelkező gépen tervezzük.

A bináris állomány összeállítása

Ha az alábbiakat követve létrehozol egy rendszerindításra alkalmas hajlékonylemezt, könnyedén elvégezheted a próbamérést:

```
$ mkdir candetect.sf.net
$ cd candetect.sf.net
$ cvs -d :pserver:anonymous@cvs.candetect.sf.net :
  ↪ /cvsroot/candetect login
$ cvs -d :pserver:anonymous@cvs.candetect.sf.net :
  ↪ /cvsroot/candetect co .
$ make /dev/fd0
```

Így – ha netán módosítani szeretnéd – a teljes forráskód a rendelkezésedre áll. A tervezet felső szintű *Makefile* állománya

- lefordítja a könyvtárat és a főprogramot;
- letölti az *xoscope* egy különleges változatát CVS-ről;
- bedolgozza a foltot, majd nem X11 alapú használatra állítja be;
- gondoskodik az *oscope* lefordításáról (ez az *svgalib*-hez tartozó bináris);
- letölt és kibont egy 2.4-es sorozatú Linux-rendszermagot;
- többféle hangkártya használatára állítja be, majd lefordítja;
- *BusyBox*ot is tartalmazó fájlrendszert állít össze;
- összegyűrja az egészet, a rendszerindításhoz pedig *syslinux*-ot ad hozzá;
- az eredményt átmásolja egy hajlékonylemezre.

A folyamat kéri a rendszergazdai jelszót, ha rendszergazdai jogosultságokkal kell futnia; a megjelenő parancssort mindig érdemes átolvasni. Megjegyezzük, hogy a *CanDetect* modulokat tartalmazó könyvtár mellett egy *xoscope* nevű is létrejön (így elkerülhető a keveredés a CVS tartalmával). Debian alapú rendszeren a *Makefile* állományok felismerik, ha valamelyik segédprogram hiányzik, és megpróbálkoznak a telepítésével. Egyéb terjesztéseken először a függőségeket kell feloldanod, vagy módosítanod kell a *Makefile* fájlokat.

A Linux előnyei és a további kihívások

A rendszermag több géptípuson is futtatható. A fejlesztést asztali és laboratóriumi munkaállomásokon végeztük, célunk egy hordozható és beágyazott számítógépeken is futtatható alkalmazás készítése volt; ezek ugyanazt a rendszermag API-t használják. Az illesztőprogramok több eszköz meghajtására is képesek, alapállapotban akár nyolc hangkártyát is kezelni tudnak. Egyetlen számítógéppel különböző hangkártyákat használva egyszerre több mérés is végezhető, amelyek ugyanazt a kód API-t használják.

A hangbemenet elérése eszközön keresztül történik,

ami ugyanúgy fájlként kezelhető, mint például a hálózati kapcsolatok. A hangkártyák vezérléséhez nincs szükség különleges hívásokra. Ez különösen olyan egyszerű gépeknél jelent fontos fegyvertényt, amelyeknél a kijelző mérete vagy a processzor teljesítménye a programok részéről némi alkalmazkodást kíván.

A roncsolásmentes mérőeszközök fejlesztésénél a munka leginkább időigényes része a helyes működés igazolása. Az algoritmusok üzembiztonságát csak úgy lehet bizonyítani, hogy a méréseket és a jelfeldolgozást különféle mintákon, különféle operációs rendszerek alatt sorra elvégezzük. Az *Open Sound System* a Linux-rendszermag más szolgáltatásaival egybefogva alkalmazásunkhoz a legkülönbözőbb számítógépeken biztosít általános futtatási környezetet. Így – mivel a forráskódban nem történtek változások – a program ellenőrzésének eredményei hitelesek lesznek.

A rendszermag átmeneti tárai miatt jelentkező késleltetések révén az algoritmusok alkalmazási köre viszonylag szűk, és mivel időnként különösen nagy értékek mérésére is szükség van, a repülésnél történő használatuk is korlátozott. A jelenlegi rendszermag-fejlesztések éppen ezeknek a korlátoknak a lebontását célozzák – ám a falvizsgálatot ez már nem érinti.

További feladatok

A *CanDetect* készen áll annak igazolására, hogy a repülésben is használható. Az eredmények megjelenítése továbbra is roppant nagy kihívást jelent. A többcsatornás hangkártyákat is szeretnénk támogatni, és ugyanezt az alkalmazást más jellegű mérések elvégzésére is alkalmassá kívánjuk tenni, ezek a következők:

- az alkatrészek ellenállása, teljesítmőképessége, induktivitása méréséhez;
- a páratartalom, a nedvesség és vegyi anyagok méréséhez;
- a tekercskapcsolások mágneses helyzetérzékeléséhez;
- az örvényáramokhoz való roncsolásmentes vizsgálatoknál;
- Hall-mágneses ellenállás iránytűként való használatához;
- a különféle anyagok zajcsillapítási mutatóinak mérésére;
- a pilóták beszélgetéseinek felvételére, mobil helyzetérzékelésre;
- a szonárokhoz és hasonló visszaverődésen alapuló célkereső módszerekhez;
- a mechanikai nyúlás- és alakzatváltozás-mérőknél;
- sugárrezonátorok esetében, például MEMS (mikroelektromechanikus) érzékelőkben.

Köszönetnyilvánítás

A NIC gépet az *Embedded Linux Journal* 2001-es pályázatának döntős résztvevőjeként kaptuk. Győztes pályaművünkről bővebben az *Embedded Linux Journal* (☞ http://embedded.linuxjournal.com/advertising/press/nic_winners.php) 2001. májusi-júniusi számában, illetve a ☞ <http://candetect.sourceforge.net> oldalon lehet olvasni. Az indukciós érzékelőt *Brian Whitecotton* készítette, a fénykép *Keith Ostrom* munkája.

Linux Journal 2002. október, 102. szám



Alex Perry

(alex.perry@ieee.org) PhD-fokozatot szerzett Cambridge-ben, az IEEE egyik vezető munkatársa. Hivatalos pilóta, repülésirányító, és repülésbiztonsági tanácsadó San Diegóban (Kalifornia). A nyílt forrású FlightGear

(☞ <http://www.flightgear.org>) repülőszimulátor egyik fejlesztője.

OpenACS-csomagok

Webalapú alkalmazások csomagjainak telepítése, megosztása és eltávolítása az APM segítségével.

A programfejlesztés egyik alapötlete szerint a nagyobb feladatokat kisebb modulokra kell bontani. A modularizálás egyszerűbbé teszi a rendszer egyedi igényekhez történő igazítását, hiszen megoldhatóvá válik a kívánt modulok beillesztése és a szükségtelenek eltávolítása. A modulok használatával a munka is könnyebben megosztható több programozó között. Egy gyors pillantás az Interneten jelenleg ingyenesen elérhető Linux-, Apache-, Perl- és Python-modulokra azonnal rávilágít a módszer jelentőségére.

Az OpenACS 4 (Open Architecture Community System) rendszert a múlt hónapban vizsgáltunk meg. Ez a hálózati közösségek létrehozására tervezett eszközkészlet legutóbbi változatai során megrázó változásokon ment keresztül. Talán mind között a legfontosabb eltérést a szolgáltatások modulokba osztása jelenti. Ezeket a modulokat az OpenACS-világban csomagoknak (packages) nevezik. Mivel minden csomag önálló, és mivel az összes csomagot bármely URL-hez csatlakoztathatjuk, az OpenACS 4 a rugalmas közösségi weblapok létrehozását minden eddiginél egyszerűbbé tette.

Ebben a hónapban elkezdjük az ismerkedést az OpenACS-csomagokkal, megnézzük, hogyan lehet őket telepíteni és használni (a cikkben feltételezzük, hogy korábban már feltelepítettük a PostgreSQL, AOLserver és OpenACS-mag programokat, ahogyan azt a Kovácsműhely előző két részében bemutattuk). Mivel a legtöbb OpenACS-hely nem próbál meg mindent nulláról felépíteni, hanem a beépített alkalmazásokkal érkező szolgáltatásokat használja, a csomagok telepítése olyasmiről, amit az alaprendszer feltelepítése után minden OpenACS-rendszergazdának végre kell tudnia hajtani.

Web/Adatbázis Csomagok

Figyeljünk meg a következő egyszerű, Perlben írt CGI-programot:

```
#!/usr/bin/perl

use strict;
use warnings;
use CGI;

my $query = new CGI;
print $query->header();
print $query->start_html(-title => "Testing");
print "<p>This is some text</p>\n";
print $query->end_html();
```

Helyezzük a fenti programot *test.pl* néven webkiszolgálónk CGI könyvtárába! A végrehajtás eredményét mindenki megtekintheti a <http://www.lerner.co.il/cgi-bin/test.pl> címen. Ha azt szeretném, hogy a programom több különböző néven is elérhető legyen, lemásolhatom; az újonnan kiválasztott nevet az URL is tükrözni fogja.

A dolgok némileg bonyolódnak, ha a kiszolgálóoldali alkalmazásom egyetlen program helyett több CGI-kódot is tartalmaz.

Amennyiben egy ilyen alkalmazásról szeretnék a rendszeremen másolatot készíteni, az összes programfájl le kell másolnom. A legtöbb esetben megkönnyíti a dolgomat, ha a fájlokat külön könyvtárba helyezem, majd ahányszor valahol máshol is le akarom futtatni, a könyvtárat a teljes tartalmával együtt lemásolom.

Az ilyen másolatok készítése azonban felvet néhány szinkronizációs problémát: ha kiküszöbölök valamilyen hibát a program egyik példányán, ugyanazt a változtatást a kód minden másolatában el kell végezni. A nehézségek egy részét ugyan a CVS használatával megoldhatom, de a helyzet úgy is megoldható, hogy csak egyetlen változatot tartok meg a fájlrendszeremen. Ezután a webkiszolgálót (legyen az Apache vagy AOLserver) úgy állítom be, hogy egy vagy több URL-en keresztül is elérhessem a programot (esetleg közvetett hivatkozást hozok létre, ez talán a legegyszerűbb – a fordító). Ezek után gondoljuk meg, mi történik, ha az alkalmazás relációs adatbázisokkal is dolgozik. Az alkalmazás telepítése már korántsem egyszerű fájlmásolásból vagy HTTP-kiszolgáló beállításból áll. Immár meg kell tudnunk oldani az alkalmazás-másolatok között fellépő esetleges ütközéseket és zavarokat, például azt, hogy a */foo/bboard* fórum adatbázisszinten ne keveredjen össze a */bar/bboard* másolattal. Amikor az alkalmazást eltávolítjuk a rendszerről, az általa használt adatbázis-táblákat is el kell tudnunk távolítani.

OpenACS alatt a gond megoldását az APM (ArsDigita Package Manager) használata jelenti. Az APM-et eredetileg a ma már nem működő ArsDigita nevű tanácsadó cég fejlesztette ki, amelyik az OpenACS elődjét is megalkotta. Az ACS kizárólag Oracle adatbázis-kiszolgáló alatt működött, az OpenACS azonban Oracle és PostgreSQL alatt egyaránt.

Az APM több, adatbázis-kezelőt használó, kiszolgálóoldali alkalmazásokra jellemző gondot is megold. Többek között változatkezelést végez, táblakészítő és -eltávolító parancsfájlokat tartalmaz és adatbázis-független. Az APM-et úgy tervezték, hogy az alkalmazás minden egyes példányja saját független beállításváltozókkal rendelkezessen, illetve egy vagy több URL-hez is hozzárendelhető legyen.

Fájlrendszerkiosztás

Az APM tulajdonképpen nem más, mint egy *.apm* kiterjesztésű *.tar.gz* fájl. Az állomány névformátuma általában: *csomagnév-0.5d.apm* formátumú, ahol a csomagnév a csomaghoz tartozó egyedi névvel helyettesítendő. Ez a példacsomag a 0.5-ös fejlesztői változatot tartalmazza. Ha a csomagot megnyitjuk a `tar -zxvf fájlnév` paranccsal, feltáru előtünk a szabványos fájl- és könyvtárszerkezet:

- *packagename.info*: a csomag tartalmát leíró XML-állomány. Ez a fájl – amit az OpenACS APM-alkalmazás általában önműködően készít el – mutatja meg az OpenACS-nek, hogy mely fájlok vannak a csomaghoz rendelve, illetve milyen beállítási értékeket érhet el a felhasználó. Azt is



megmutatja, hogy az alkalmazás singleton-e (azaz nyújt-e szolgáltatásokat a rendszer többi részének), vagy hagyományos alkalmazás (vagyis egy bizonyos URL alól futtatható).

- Az SQL könyvtár: itt találjuk a táblalétrehozó (és táblatörlő) parancsfájlokat. Eredetileg, amikor az ACS még csak az Oracle-t támogatta, ez a könyvtár mindössze két fájlt tartalmazott, *csomagnév-create.sql* és *csomagnév-drop.sql* néven. Az APM-telepítő a csomag telepítésekor a *create* (létrehozó) parancsfájlt, a csomag eltávolításakor pedig a *drop* (törlő) parancsfájlt futtatja (a *create* parancsfájl gyakran INSERT-eket is használ; ilyenkor a későbbi felhasználáshoz a táblákat szabványadatokkal tölti fel).
- A *tcl* könyvtár függvény meghatározásokat tartalmazó *Tcl*-fájlokat tárol. Ezek a függvények indításkor betöltődnek az AOLserverbe, ami sebességnövekedést jelent az OpenACS rendszer egyéb helyein használt *.tcl*- (vagy *.adp*-) lapokhoz képest.
- Jelenleg – mivel az OpenACS az Oracle mellett a PostgreSQL-t is támogatja – ez a könyvtárszerkezet módosult kissé. Az *sql* könyvtárban egy *oracle* és egy *postgresql* alkönyvtárat találunk, amelyek azonos szolgáltatású táblalétrehozó és -törlő parancsfájlokat tartalmaznak. Minden feltelepített OpenACS rendszer tárolja, hogy éppen milyen adatbázis-kezelőt támogat (ezt az adatot az AOLserver *nsd.tcl* beállításfájljának egyik változójában találjuk meg), és ennek megfelelően választja ki a legmegfelelőbb parancsfájlt.
- A *www* könyvtárban találjuk azokat a dolgokat, amelyek logikusan a webalkalmazás részét képezik. Ide helyezzük *.tcl*- és *.adp*-lapjainkat, illetve az összes grafikát, és az alkalmazáshoz tartozó egyéb kiegészítő állományokat is. Az OpenACS lekérdezőstovábbítója (query dispatcher) – ami lehetővé teszi, hogy a kiszolgálóoldali programok több adatbáziskiszolgálót is támogassanak – *.xml* kiterjesztésű XML-fájlokkal működik; ezek szintén a *www* könyvtárba kerülnek.
- Az OpenACS sablonrendszerének működési sajátosságai folytán nem ritka, hogy egyetlen honlap három fájlt is használ: egy *.tcl*-fájlt a változók beállításához, egy *.xml*-fájlt, ami az adatbázis táblasorainak SQL-lekérdezését teszi lehetővé, és egy *.adp*-fájlt, amely az adat HTML-átalakításáért felelős.

Az APM-ek ezeken felül számos egyéb fájlt is tartalmazhatnak, például adatbázis-továbbfejlesztő és összefésülő parancsfájlokat (azok számára, akik a csomag előző változatából szeretnének adatokat átmenteni), működési próbákat (a csomag hibátlan működésének az ellenőrzésére), karbantartói eszközöket (a *www/admin* alatt) és HTML formátumú csomagleírást (a *www/doc* alatt).

Csomagok betöltése és telepítése

Az APM-csomag használatba vételének első lépése a betöltés, ami általában annyit jelent, hogy ugyanabba a fájlrendszerbe másoljuk, ahol OpenACS rendszerünk is megtalálható. Amennyiben OpenACS rendszerünk a */web/atf/* könyvtárban helyezkedik el, minden csomagot a */web/atf/packages* könyvtárba kell helyezni (ezért aztán minden egyes csomagnak egyedi névre lesz szüksége; sok OpenACS-fejlesztő követi az Emacs-stílusú csomagelnevezési hagyományt, ahol a csomag nevét a fejlesztő vagy az ügyfél neve előzi meg – ez segít elkerülni az egyes csomagok közötti ütközést). A csomag teljes könyvtárszerkezetét másoljuk a */web/atf/packages*-be, és győződjünk meg róla, hogy a fájlok és könyvtárak olvashatók

(és írhatók) azon a felhasználói néven, amely alatt az AOLserver üzemel.

Az APM-csomagok telepítésének könnyebb és megbízhatóbb módja, ha a csomagkezelőben a *load packages* (csomagok betöltése) hivatkozásra kattintunk. Az OpenACS rá fog kérdezni az APM-hez rendelendő URL-re vagy a csomagokat tartalmazó könyvtárnévre. Az OpenACS ezt követően megkeresi az összes ott található *.apm*-fájlt, kicsomagolja és betölti őket a rendszerbe.



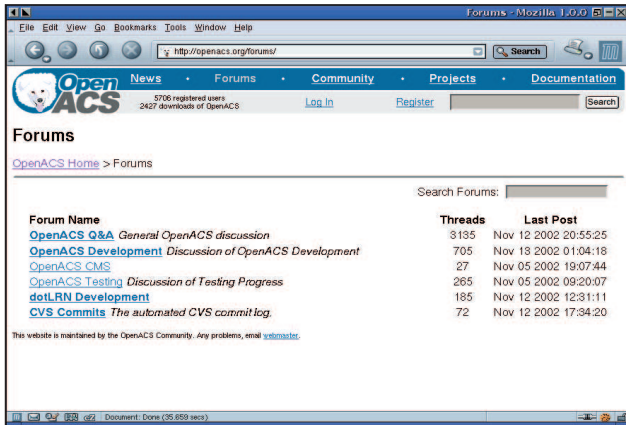
Miután a csomag betöltődött a fájlrendszerbe, fel kell telepítenünk a hozzá tartozó adatmodellt, és be kell jegyeznünk a rendszerbe. Ezt a weblapú csomagkezelő felhasználásával tehetjük meg, amit az OpenACS rendszereken általában a */acs-admin/apm* URL alatt találhatunk meg. A csomagkezelőhöz általában csak a hely karbantartója férhet hozzá.

A csomagkezelő oldala a rendszerbe töltött összes csomagot bemutatja, illetve jelzi, melyek vannak telepítve, melyiket helyettesítette egy frissebb változat, vagy melyek várnak még telepítésre. A lap tetején fellelhető menü segítségével a rendszer csomagjainak különféle alhalmazait is leválogathatjuk, például kérhetjük, hogy csak azokat mutassa, amelyek fejlesztéséért és kezeléséért személyesen felelünk.

A csomagkezelő a legfontosabb eszköz, amellyel a csomagok létrehozását, módosítását, frissítését és telepítését végezzük:

- Telepíthetjük a csomaghoz tartozó adatmodellt, és bejegyezhetjük a csomagot az OpenACS-be. Az OpenACS következő indításakor a csomag *tcl* könyvtárban található fájlok az AOLserver memóriájába töltődnek. Csomagunk ezt követően az OpenACS honlaptérképén keresztül csatolható URL-hez, ahogy azt hamarosan láthatjuk.
- A csomagkezelővel új OpenACS-csomagokat is készíthetünk. Az új OpenACS-csomag előállításának első lépése, hogy a csomagkezelővel egy új könyvtárat és *.info* fájlt hozunk létre.
- Megvizsgálhatunk bármilyen, korábban a rendszerbe töltött fájlt, és lekérhetjük a csomaghoz rendelt kapcsolók, fájlok vagy bármilyen egyéb adat listáját.
- A csomagot módosíthatjuk is. Megváltoztathatjuk a kapcsolóit, fájljait, illetve a csomaghoz tartozó egyéb adatokat.

A csomag telepítéséhez kattintsunk az *install packages* hivatkozásra a lap alján. A csomagkezelő előbb új csomagokat keresve végignézi a csomagkönyvtárat, majd kiválaszthatjuk a telepí-



teni kívánt csomagot (ha első ízben telepítjük az OpenACS-t, egyetlen csomag sincs még telepítve, ezért itt igen hosszú listát találunk). Minden csomag egy vagy több másik csomagtól függhet. Ha olyan csomagot próbálunk feltelepíteni, amelynek az előkövetelményei még nincsenek fenn, a csomagkezelő a folytatás előtt megerősítést kér.

A telepítöben megadhatjuk, hogy csak a csomag adatmodelljét szeretnénk-e feltenni, vagy a csomag használatát is engedélyezzük a webhelyen. A magam részéről én minden felrakott csomagot engedélyezek, de biztos vagyok benne, hogy van olyan helyzet, amikor ezt nem szeretnénk megtenni. Miután ellenőriztük a megfelelő dobozokat, kattintsunk a *Next* gombra, ami az adatmodellt fogja telepíteni. Ezek után az AOLserver-t újra kell indítanunk, hiszen sok modul függ az AOLserver indításakor betöltődő *tbl* könyvtáraktól. A csomagok addig nem működnek, amíg a kiszolgálót újra nem indítjuk.

Csomag befűzése

Az AOLserver újraindítását követően ismét lépünk a csomagkezelőbe, és nézzük meg az engedélyezett csomagok listáját. Az összes betöltött csomagnak látszania kell a listában. Ezen a helyen a csomagokat tetszés szerint módosíthatjuk, újabb elemeket tölthetünk be, illetve a betöltött csomagot tényleges honlappá alakíthatjuk. Ezt a befűzésnek (mounting) nevezett folyamatot az OpenACS helytérképén végezhetjük el. Az OpenACS karbantartó rendszerének egyik legzavaróbb tulajdonsága, hogy a helytérkép nem a teljes helyhez tartozó felügleleti lapon található, hanem a fő webhely felületére került. Más szavakkal a csomagokat a */acs-admin/apm* alatt tudjuk kezelni, ellenben a térképet a */admin/site-map* alatt érhetjük el. Ennek is megvan az oka, de inkább megtéveszti az embereket, semmint segít rajtuk.

A hely térképe mutatja meg az OpenACS-nek, hogy milyen URL-t kell az alkalmazáshoz csatolnia, például ha a *bboard* OpenACS-csomagot a */forum* URL vagy a */bboard* URL alá szeretnénk elérni. Néhány esetben ténylegesen jól jöhet, ha mindkét helyen elérhetjük. A térképen mindezt egérgattintásokkal oldhatjuk meg.

Amikor a csomagot első ízben szeretnénk valamilyen URL-hez rendelni, a / elérési út jobb oldalán található *New subfolder* befűzést kell használnunk. Meg kell adnunk, hogy melyik URL alá szeretnénk az új alkalmazást elhelyezni. Ahhoz például, hogy a *bboard* csomag a */forum* alá legyen elérhető, a forum szót kell megadnunk (a bevezető perjel nélkül).

Amennyiben ennyivel megelégszünk, az új alkönyvtár semmivel sem lesz több egy mappánál, amibe újabb mappákat,

illetve állandó dokumentumokat helyezhetünk. Ezért a mappához tartozó *New application* befűzésre kell kattintanunk, ahol a feltelepített alkalmazáscsomagok közül a megfelelőt kiválasztva azt emberi fogyasztásra alkalmas névvel láthatjuk el. Ez a név fog megjelenni a fejlécekben és a címsorokban. Előfordulhat ugyanis, hogy a csomagot a */bboard* URL mögé szeretnénk helyezni, de névként inkább az érthetőbb *Vitaforum* nevet adjuk neki. A címet nem lehet egyszerűen megváltoztatni, ezért válasszunk megfontoltan!

A *new application* befűzés egy új csomagpéldányt hoz létre, majd azt csatlakoztatja az általunk választott URL-hez. Ha még egy nevet szeretnénk az alkalmazáshoz rendelni, készítsünk egy újabb alkönyvtárat, és használjuk a *mount* hivatkozást. Eltartott egy darabig, mire rájöttem, hogy a *mount* létező alkalmazáspéldányok csatlakoztatására is képes, a *new application* pedig egy teljesen új alkalmazáspéldányt hoz létre. Így már értelmet nyer az alkalmazásleoldás (unmounting) lehetősége is (az elérési út jobb oldalán található *unmount* hivatkozással érhetjük el), ami a kiválasztott alkalmazást nem törli, csak elérhetetlenné teszi. Az adott alkalmazáspéldány végleges törléséhez a térképen az *unmounted application* hivatkozásra kell kattintanunk, majd a *delete link*-re kell – közvetlenül az *unmounted application* után – rábóknunk.

Az alkalmazások minden egyes példánya saját jogosultságrendszerrel és kapcsolókkal bír. A kapcsolók használatát különösen hasznos résznek találtam az OpenACS-ben, hiszen így annak ellenére, hogy az egyszer létrehozott alkalmazást több helyen használom, minden egyes példányhoz saját beállításokat rendelhetek. A térkép megfelelő hivatkozásait követve megtekinthetjük, illetve megváltoztathatjuk a csomagpéldányhoz rendelt kapcsolókat.

Összegzés

A *.apm*-fájl formájában terjesztett és CVS-ben az APM-alkalmazással karbantartott OpenACS-csomagok web- és adatbázis-alapú alkalmazások készítését és terjesztését teszik lehetővé. Importálás után az APM-csomag több példányban is felhasználható, egyedenként külön jogosultságokkal és eltérő kapcsolókkal. Mint azt a következő hónapban látni fogjuk, az APM-et saját web-, illetve adatbázis-alkalmazás csomagjaink készítésénél is felhasználhatjuk, ezáltal könnyedén terjeszthetjük társalkotóink és a közösség más tagjai között.

Linux Journal 2002. november, 103. szám



Reuven M. Lerner

(reuven@lerner.co.il) egy kisebb webes és internetes módszerekkel foglalkozó tanácsadó cég tulajdonosa és vezetője. Az ATF honlapon érhető el (☞ <http://www.lerner.co.il/atf/>).

Kapcsolódó címek

Az OpenACS-ről bővebben a

☞ <http://www.openacs.org>-on olvashatunk. A weblap leghasznosabb része valószínűleg a

☞ <http://www.openacs.org/bboard> címen található fórum, ahol vezető OpenACS-fejlesztők beszélgetnek egymással, illetve vitatják meg az eszközkészlet jövőjét.

Kóstoljunk bele a tisztán linuxos irodába!

Néhány egyszerű, elegáns és gazdaságos módja annak, hogy munkaállomásainkat Linuxra állítsuk át.

François, csodálatos látványt nyújt ma az étterem! Menj, kérlek, még egyszer körbe, hogy minden munkaállomás rendelkezze-e a szükséges eszközökkel – szeretném, ha minden tökéletesen működne.

Bonsoir, mes amis, isten hozott Chez Marcelnél, a kitűnő Linux-konyha és a csodálatos borok otthonában! Foglaljatok helyet, és helyezétek magatokat kényelembe! François azonnal hozza a bort. Siess! Az 1999-es Chateauf-neuf-du-Pape illeni fog az alkalomhoz.

Azok számára, akik egy ideje látogatják már éttermünket, biztos örömet okoz az a tény, hogy számos vállalat (és magánszemély) keres más választási lehetőséget jelenlegi asztali operációs rendszere helyett. A Linux a kézenfekvő megoldás, és az átálláshoz számos viszonylag fájdalommentes lehetőséget kínál. Sajnos az újabb Linux-rendszer-csomagok a KDE vagy Gnome futtatásával meglehetősen nagy erőforrásigényt támasztanak, sok vállalat pedig éppen a szerényebb PC-ken is működő rendszerek lehetőségét keresi. Szerencsére a Linux lehetőséget ad rá, hogy a munkaállomásokra csak a legszükségesebb összetevők kerüljenek, a munka nagy része pedig a nagy teljesítményű kiszolgálót terhelje.

A telepítésnek ezt a módját nevezik *vékony ügyfél modellnek*. Hogy pontosan milyen „vékony” lehet ez az ügyfél, az a választott megközelítéstől függ. Ha kitűnő alapanyagok széles választéka áll rendelkezésünkre, elvárható, hogy sokféleképpen lehessen elkészíteni a menüt. A friss homár megpárolva, vajöntettel és fokhagymával tálalva nyilvánvalóan kitűnő, de nem ez az egyetlen elkészítési módja. Ugyanez igaz arra, amit Linux-konyháinkban készítünk, ezúttal a vékony ügyfél megvalósítására. Az egyik megközelítés még a számítógép újratelepítését sem igényli.

A vékony ügyfél megvalósításának talán legegyszerűbb módja, ha a munkagép nem futtat ablakkezelőt, hanem azt a kiszolgálóról kapja. Nézzük ennek megoldását:

```
startx /usr/X11R6/bin/xterm
```

A kapott eredmény messze nem mondható érdekesnek. Egy dísztelen, öreg X-felületet kapunk, amiben semmi nem fut, csak az X-terminál, semmi ablakkezelés vagy ilyesmi. Mit lehet ezzel így kezdeni? Egy dolgot biztosan, az OpenSSH segítségével csatlakozni a kiszolgálónkhoz, az alábbi módon:

```
ssh -C -X -l user_name server_name
```

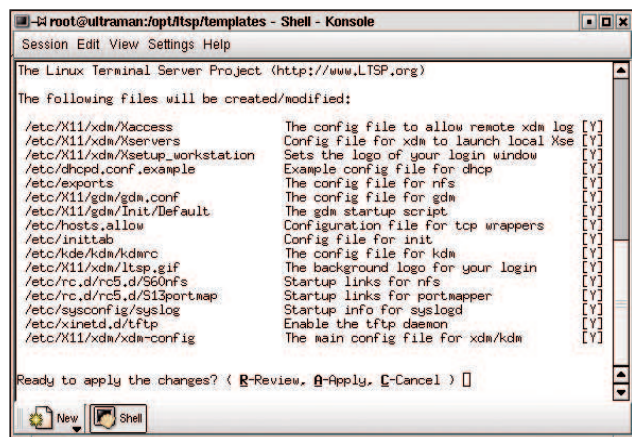
A -C kapcsoló elindítja a biztonságos adatfolyam tömörítését, míg a -X lehetővé teszi az X-kérések továbbítását. Amint a parancssorban találjuk magunkat, már indíthatjuk is az asztalunkat a távoli gépen a startkde vagy wmaker vagy más általunk kedvelt asztal indítóparancsával. Hirtelen egy olyan grafikus asztali környezetben találjuk magunkat, ami a távoli kiszolgálón fut.

Megjegyzendő, hogy a kiszolgáló `/etc/ssh/sshd_config` fájljában

az X11-továbbítást engedélyezni kell, mielőtt a kiszolgáló az ügyfél kéréseit fogadná. A fájl megfelelő kapcsolójának beállítása az alábbi módon történik:

```
X11Forwarding yes
```

Ha meg kell változtatnunk a kapcsolót, az sshd újraindítása is szükséges lesz, amit az `service sshd restart` parancsral végezhetünk el. A távoli munkavégzés ez a módja remekül működik, és még akkor is igen hasznos, ha már futtatunk egy grafikus Linux-felületet. Leginkább akkor vesszük hasznát, ha nélküle a számítógépteremig kellene kutyagolnunk, ami természetesen az épület másik végében található.



Az LTSP beállítása

Ha a saját felületeket is futtatni szeretnétek a kiszolgálón lévő mellett, a CTRL-ALT-F2 (vagy F3, F4, és így tovább) billentyű-kombinációval kapcsoljatok át valamelyik karakteres konzolra, és indítsátok el egy második X-folyamatot, valahogy így:

```
startx /usr/X11R6/bin/xterm -- :1
```

A -- :1 beállítás (figyeljünk a kettős kötőjelre!) egy második grafikus felületet indít el – egy újabb unalmas xterm ablakkal és egy újabb unalmas háttérrel. Innen az ssh-t használhatjuk a kiszolgálóhoz való csatlakozáshoz, és bármilyen tetszőleges ablakkezelőt elindíthatunk. A grafikus felületek között a CTRL+ALT+F7 CTRL+ALT+F8 lomkiszolgálókkal váltogatunk. A beállítás hátrányai között kell megemlítenem, hogy az a szó szoros értelmében nem tekinthető „vékonynak”, és az OpenSSH-adatfolyam a tömörítés mellett is lassúnak tűnhet. Egy X-folyamat futtatásának (az ügyfélgép minél kisebb terhelése mellett) legjobb módja az XDMCP (X Display Manager Protocol, azaz X-megjelenítéskezelő protokoll) használata. Indításához az ügyfélgép Linux-parancssorába egyszerűen a következő sort gépeljétek be:

```
X -query kiszolgalo_neve
```

És csodák csodája: egy grafikus bejelentkezőképernyőt láthatok, éppen olyat, mintha a kiszolgálón futna. Óh, mes amis, mint azt bizonyára sejtitek, ennek az egyszerű receptnek az elkészítéséhez szükség van egy kis előkészületre, még mielőtt a parancsot futtathatnánk – de nagyon egyszerű. Mialatt François újratölti poharaitokat, el is magyarázom. Ahhoz, hogy a kérést küldő gép számára grafikus bejelentkezőképernyő jelenjen meg, a `/etc/X11/xdm/Xaccess` fájlban engedélyezni kell az alábbi sorokat:

```
*                               #bErmelyik gØp
                               #kaphat bejelentkezi
                               #ablakot
*   CHOOSER BROADCAST          #bErmelyik indirekt gØp
                               #kaphat vElaszt t
```

Továbbá szükség van az `xdm` (vagy a `kdm`, vagy a `gdm`) futására, ennek egyik gyors megoldása, ha a kiszolgálót 5-ös futás-szintre kapcsoljuk. Akik közületek grafikus bejelentkező ablakkal indítják a gépet, valószínűleg átugorhatják a következő néhány sort. Ahhoz, hogy 5-ös futási szinten indítsunk szolgáltatásokat, a `/etc/inittab` fájl `initdefault` kapcsolóját 5-ösre kell állítanunk:

```
id:5:initdefault:
```

Ha a gép nem grafikus felülettel indul, a fenti 5 helyett 3-nak kell állnia. Még egy utolsó beállítást kell végrehajtani ebben a `/etc/inittab` fájlban. Nézzük meg a fájl végét, ahol valami ehhez hasonlót kell találnunk:

```
# Az xdm futtatEsa 5- s futEszszinten
# Az xdm gy egy œj k l n szolgEltatEsz lesz
x:5:respawn:/etc/X11/prefdm -nodaemon
```

Jelentkezzünk ki és indítsuk újra az X-kiszolgálót, hogy az új beállítások életbe lépjenek. Ezután a munkaállomásról futtasdátok az `X -query ip_c m` (természetesen a kiszolgáló IP-címe) parancsot. Ha XDM-et futtatok, a dolognak működni kell. Természetesen sokan közületek a KDE-ben lévő `kdm` bejelentkezéskezőt használjátok, és csak egy üres szürke képernyőt láttok egy keresztel a közepén (bár a keresztet mindenfelé mozgatni lehet, ez nem túl érdekes). Ennek oka, hogy a KDM elérése az XDMCP-n keresztül egy további kis trükköt igényel.

A KDE a `kdmrc` fájlt használja. Az én rendszeremen ez a `/etc/kde/kdm` könyvtárban található. Keressétek meg az alábbi sorokat, és győződjétek meg róla, hogy az `[Xdmcp]` rész `Enable` kapcsolója `true`-ra van-e állítva:

```
[Xdmcp]
# Figyeljen-e a KDM az XDMCP kØrØsekre:
Enable=true
```

Mialatt François kihozza a vajas Brie sajtot a bor mellé, hadd szóljak néhány szót a legvékonyabb ügyfélgépekről, a merevlemez nélküli munkaállomásokról. Ezt a megközelítést különböző beállítások támogatják, ezek közül ma egyre fogok összpontosítani. Mivel ez a Linux munkahelyi bevezetésének egy igen fontos eszköze, ennek a kitűnő Linux-konyhának a tulajdonosaként kötelességemnek érzem, hogy más projektek közül

is kiemeljek egy maroknyit, amelyek figyelemre tarthatnak számot. Még záróra előtt megosztok veletek néhányat ezek közül, de most beszéljünk ezeknek a munkaállomásoknak az indításáról.

Az ethernetkártyátok megfelelő meghajtóprogramjával ellátott indítólemezt az Etherboot weboldalon (<http://etherboot.sourceforge.net>) található eszközök segítségével készíthettek. A forrás lefordítása egyszerűen a tar-csomagok kibontásából, az `src` könyvtárra való váltásból és a `make` parancs futtatásából áll. Egy még egyszerűbb módszer is rendelkezésünkre áll a ROM-o-matic.net (<http://rom-o-matic.net>) tagjainak köszönhetően. Ők foglalkoznak egy indító ROM-gyűjtemény karbantartásával, amelyet az Etherboot projekt alapján készítenek. Válasszuk ki a kívánt kiadást (a legfrissebb általában a legjobb), a kártyánk típusát (3c509, tulip stb.) és kattintsunk a **Get ROM** gombra. A letöltött fájl lemezre tudjuk írni:

```
cat eb-5.0.6-yournic.lzdisk > /dev/fd0
```

Voilà, a lemezünk kész! Tegyük be a munkaállomás számítógépbe, és indítsuk újra. A rendszer indítási folyamata tart egy kis szünetet, hogy lehetőséget adjon a helyi vagy hálózatos indítás kiválasztására – a hálózatos az alapértelmezett. A gond csak az, hogy az ügyfélgépnek indítás után nincs mihez csatlakoznia. Ez az a pont, ahol a Linux Terminal Server Project (<http://www.ltsp.org>) szerephez jut. A projekt lényege, hogy a vékony Linux-ügyfél beállítását és használatát a lehető legegyszerűbbé teszi.

Most tegyük félre egy kicsit a munkaállomást, és néhány percig összpontosítsunk a kiszolgálóra. Szükségünk lesz néhány további dologra, kezdve a DHCP-kiszolgálóval (Linux-rendszerünk `dhcp` csomagja), egy TFTP-kiszolgálóra (a legvalószínűbb, hogy `tftp-server` néven találjuk meg a telepítőlemezen), továbbá NFS-kiszolgálón csomagjainak a telepítésére. Mindegyiknek ott kell lennie a telepítő lemezen, úgyhogy nem kell túl messzire mennünk értük.

A TFTP esetén egy bejegyzés jön létre a `/etc/xinet.d` könyvtárban a szolgáltatás számára. Bár a telepítő létrehozta ezt a bejegyzést, alapértelmezésben sajnos le is tiltja. Győződjünk meg arról, hogy a szolgáltatás (`etc/xinet.d/tftp`) tartalmazza-e az alábbi sort:

```
disable = no
```

Indítsuk újra a `xinetd`-t a TFTP-démon működésbe hozásához. Ha még nem tettük meg, ez jó alkalom magunknak az LTSP-csomagok letöltésére. Az oldalon minden csomag tar és gzip formátumban is megtalálható, de előre fordított RPM-csomagokat is letölthetnek a Red Hat, SuSE, Mandrake stb. felhasználói. A Debian használók is megtalálják a maguk előre csomagolt DEB-fájljait. Ha a tar formátumot választjuk, egyszerűen bontsuk ki őket, és minden csomag esetén futtassuk le az `install.sh` parancsállományt. A csomagokról szólva, íme, ezekre lesz szükség a meghatározott sorrendben: `ltsp_core`, `ltsp_kernel`, `ltsp_x_core` és `ltsp_x_fonts`. Van még néhány helyi alkalmazáscsomag, ami szintén szükséges a futtatáshoz, például a Netscape, de ezekkel most ne foglalkozzunk. Ez az a pont, ahol a beállítás során kezdtek egy kicsit érdekesebbé válni a dolgok. Három különböző gépre telepítettem fel az LTSP-t, és úgy tapasztaltam, hogy a sorrend igen fontos. Az LTSP beállításához már futó `dhcpd`-re volt szükségem, de a démon elindításához érvényes `/etc/dhcpd.conf` fájljal kellett

rendelkezni. Mivel az autoconfig nem működött, ez a fájl nem volt a gépen, ezért úgy döntöttem, hogy kézzel hozok létre egy saját *dhcpd.conf* fájlt – ez nem is olyan bonyolult. Vessetek egy pillantást az alábbi beállításra! Mintának megfelelő, természetesen mindig szem előtt tartva, hogy a tartomány (és a root-path, valamint a kiszolgáló IP-címe) különbözni fog az enyémtől:

```
subnet 192.168.22.0 netmask 255.255.255.0 {
    option domain-name "yourdomain.dom";
    option root-path
    ↪ "192.168.22.10:/opt/ltsp/i386";
    range dynamic-bootp 192.168.22.128
    ↪ 192.168.22.254;
    default-lease-time 21600;
    max-lease-time 43200;
}
host baroque {
    filename "/lts/vmlinuz-2.4.18-ltsp-1";
    hardware ethernet 00:50:FX:5B:XX:56;
    fixed-address 192.168.22.5;
    option host-name "baroque";
}
```

Vegyétek észre, hogy legalább egy számítógépet létrehoztam. A fájlnev megfelel a legfrissebb LTSP-kiadásnak, de az ethernetkártya eszköz- vagy MAC-címének (amellett, hogy itt csak kitalált érték szerepel) meg kell felelnie az ügyfélgép adatainak. A *dhcpd* a *service dhcpd start* parancsral indítható. Miután DEB-, RPM- vagy tar-csomagjainkat telepítettük, és elindítottuk a különböző szolgáltatásokat, ideje köszönetet mondanunk az LTSP tagjainak, mert a többi munka nagy részét már elvégezték helyettünk. Nekünk már csak a rendelkezésünkre álló előkészítő parancsfájlt kell lefuttatnunk:

```
cd /opt/ltsp/templates
./ltsp_initialize
```

A parancs beírása után figyelmeztetést kapunk, hogy a parancsfájl során fontos rendszerállományok fognak megváltozni, mégpedig a */etc/dhcpd.conf*, */etc/exports* és néhány egyéb fájl (lásd a *képet*).

Mivel az NFS új exportokkal frissült, és a DHCP beállításai is megváltoztak, mindkét démont újra kell indítani:

```
service nfs restart
service dhcpd restart
```

Csaknem készen is vagyunk, mes amis. Másodperceken belül rendelkezni fogunk egy élvonalbeli, mégis szerény, linuxos vékonyügyfél-telepítéssel. Mivel az LTSP minden programja a */opt/ltsp* könyvtárban van, néhány beállítás kedvéért egy kis kirándulást kell tennünk ide. Kezdjük a */opt/ltsp/i386/etc/lts.conf* fájl szerkesztésével. Nézzük az alábbi sort!

```
SERVER = 192.168.22.10
```

Ide annak a gépnek az IP-címét kell beírunk, amelyiket kiszolgálóként használjuk. Ezt az *lts.conf* fájlt érdemes közelebről is megismerni. Ebben lehetőség nyílik bizonyos beállítások megváltoztatására abban az esetben, ha a munkaállomások valamelyike nem működne az alapbeállításokkal. Íme egy beállítás, amit *baroque* nevű munkaállomás számára készítettem (egy

öreg soros egérrel kellett működésre bírnom):

```
XSERVER = vesa
X_MOUSE_PROTOCOL = "Microsoft"
X_MOUSE_DEVICE = "/dev/ttyS0"
X_MOUSE_BUTTONS = 3
LOCAL_APPS = N
SWAPFILE_SIZE = 64m
RUNLEVEL = 5
```

Amikor egyre újabb és újabb munkaállomásokat adunk a rendszerhez (és üzembe helyezzük), ne felejtsük el, hogy ezeket hozzá kell adnunk a */etc/dhcpd.conf* fájlhoz. Tehát a *dhcpd* szolgáltatást nem kell újraindítani a változtatások után. Amikor túlvagyunk ezen a kis kellemetlenségen, tegyük be a hajlékonylemezt a meghajtóba, indítsuk újra a munkaállomást, és néhány másodpercen belül látnunk kell a folyamatot, ahogy a rendszer-mag-nyomattájl a TFTP-n keresztül a munkaállomásra töltődik. Ezután gépindításkor lefutó sorokat láthatjuk, majd egy önműködő X-beállítás követi. Ha az X a folyamatnak ezen a pontján esetleg megáll, a munkaállomás bejegyzését az *lts.conf* fájlban átállíthatjuk 3-as futási szintre. Ekkor a gép indítása után szöveges módba jutunk.

A záróra vésszesen közeleg, mes amis, de megígértem, hogy megemlítek még néhány hasonló projektet. Figyelmetekre lehet érdemes a Solucorp's X terminal eszközkészlet (↪ <http://www.solucorp.qc.ca>), amelyet a Linuxconfról híres *Jacques Gélinas* vezet. Ne hagyjátok ki *Diego Torres Milano* weboldalát se a ↪ <http://pxes.sourceforge.net> címen, ahol a PXES vékonyügyfélcsoomagról szerezhetek további adatokat. Köszönöm, mes amis, hogy Chez Marcellel töltöttétek ezt az estét. Remélem, jó benyomást gyakorolt rátok a „vékony” Linux, és az üzleti életben hasznosítani tudjátok az itt látottakat. Az egyszerűsített felügyeletben rejlő költségmegtakarítás ezt a megoldást tökéletes választássá teszi a mai költségérzékeny világban. Frissítsük a kiszolgálót, és ezzel frissítettünk minden programot; készítsünk biztonsági másolatot a kiszolgálóról, és ezáltal mentettük az összes munkaállomást – mi lehetne ennél egyszerűbb? François, kérlek, utoljára még töltsd tele vendégeink poharát. Vizontlátásra a következő hónapig! A votre santé! Bon appétit!

Linux Journal 2002. november, 103. szám



Marcel Gagné

Mississaguában, Ontario államban él. Ő a szerzője a Kiskapu kiadásában szeptemberben megjelent Linux-rendszerfelügyelet (ISBN 963-9301-40) című könyvnek (jelenleg is egy könyvön dolgozik). A mggagne@salmar.com címen érhető el.

Kapcsolódó címek

Etherboot ↪ <http://etherboot.sourceforge.net>
 Linux Terminal Server Project ↪ <http://www.ltsp.org>
 Linux X Terminal Toolkit ↪ <http://www.solucorp.qc.ca>
 Marcel's Wine Page
 ↪ <http://www.marcelgagne.com/wine.html>
 PXES Linux Thin Client ↪ <http://pxes.sourceforge.net>
 ROM-o-matic ↪ <http://rom-o-matic.net>

Ébresztő, Amerika, elmegy a hajó!

A nyitott gondolkodású embereknek és a kísérletező kedvnek hála a Linux Panamában is terjedőben van.

Amikor a Linux sikertörténeiről esik szó, sokaknak valószínűleg csak a DreamWorks vagy a többi nagy cég jut eszébe. Természetesen ezek is nagyszerű piacok a Linux számára, de nem az egyetlenek. Igazán látványos fejlődést az Egyesült Államok határain kívül, a harmadik világ országaiban láthatunk, például Panamában, ahol én is élek. Amikor ideköltöztem, szinte senkit nem ismertem, aki akárcsak hallott volna a Linuxról, nemhogy ki is próbálta volna. Szerencsére a helyzet gyorsan változik. Az indulás dögös volt, de minden nap újabb eredményeket hoz: itt egy tűzfal, ott egy levélkiszolgáló – és szépen haladunk előre. Nem kis részben az olyan terjesztéseknek köszönhetően, amelyek új meghajtó megvásárlása vagy a régi operációs rendszer letörlése nélkül is lehetővé teszi a Linux kipróbálását – ilyen például a Knoppix (☞ <http://www.knopper.de>) – eljutottunk oda, hogy a Linux lassan meghódítja az üzleteket és az asztali gépeket. Négy és hét éves lányaim be tudnak jelentkezni Linux alá, tudnak vele webezni és játszani, és nem azon török a fejüket, hogy létezik-e másik operációs rendszer is. A ☞ <http://www.Barbie.com> és a ☞ <http://www.CartoonNetwork.com/PowerPuffGirls> oldalakat szeretik a legjobban, a Galaga és a Gcompris pedig hálózat nélkül is elszórakoztatja őket. Amerikának nem szabad feladnia az előnyét, hiszen még a gyerekek is meg tudják szeretni a Linuxot. Ha tehát a feltörekvő piacokra gondolunk, gondolkodjunk világméretben, és gondoljunk a fiatalokra.

Genmenu

Ennek a Bash parancsfájlnak a segítségével, egy menüt készíthetünk, amivel a Blackbox, a Fluxbox, az Enlightenment vagy a WindowMaker közül választhatunk. Mivel a beállításokat a \$HOME könyvtárban tárolja, nem csak a rendszergazda használhatja. Javasolom, először készíts biztonsági mentést, mivel a Genmenu az eredeti fájlokat derekasán átszabja – valószínű, hogy írásom megjelenésének idejére a hibát már kijavították. Ha járatos vagy a Bash-programozásban, úgy is átírhatod, hogy az általad kedvelt alkalmazásokat is megkeresse, vagy hogy bizonyosakat ne keressen meg. A futtatásához Bash szükséges. ☞ <http://projects.gtk.mine.nu/genmenu>

HTun

Ha olyan tűzfalon akarsz keresztüljutni, amelyen csak egy vagy két kapu van nyitva, a túlóldali rendszerekhez viszont különböző kapukon kell csatlakoznod, a HTun lehet a megoldás. A HTun futtatásához szükséged lesz egy internetkapcsolattal rendelkező gépre (például az otthoni gépedre), amelyhez VPN-kapcsolatot létesítesz, majd az otthoni gépedet ugródeszkaként használhatod olyan rendszerek eléréséhez, amelyek a tűzfal mögül nem láthatók. Természetesen némi védelemről gondoskodni kell, hogy mások ne élhessenek vissza az otthoni géped által nyújtott lehetőséggel, ám ettől függetlenül nagyszerű eszköz a túl szigorú tűzfalak megkerülésére. A futtatásához libpthread és glibc szükséges. ☞ <http://htun.runlinux.net>

Shared Recording Server

Az erez nyers hangkezelő rendszerekhez készült. Képes csatlakozni a hangeszközhöz, és könnyebben kezelhető felületet biztosít hozzá. Futása közben a 6130-as kapun más program – akár több is – csatlakozhat hozzá, és átvehetik a hangkártyáról érkező hangokat. A futtatásához szükséges: libargh (a csomag tartalmazza), libstdc++ , libm és glibc. ☞ <http://oktober.stc.cx/source/erec.html>

Simple Expense Manager

Annyira könnyen kezelhető kiadás-nyilvántartó program, hogy már csak ezért is érdemes kipróbálni. Eddig a könyvelőprogramommal tartottam nyilván a kiadásaimat, de ezt sokszor fárasztó nap mint nap megtenni. Igaz, a pénzmozgásokat továbbra is be kell vinnem a könyvelőprogramba, de megtehetem a hét vagy a hónap végén is, összesítve, így nem kell naponta megküzdenem a feladattal. A futtatásához Perl és az Apache::Htpasswd Perl-modul szükséges. ☞ <http://www.angelfire.com/tn3/petbath>

ProBIND

Mindig örülök, ha egy eszköz megkönnyíti – például – a DNS-felügyeletet. A ProBIND roppant hasznos kiegészítő, bár ha több felhasználód is van, akik csak a saját DNS-kiszolgálójukat frissíthetik, akkor biztonság tekintetében nem tökéletes. Ha csak egy ember frissít több zónát, vagy mindegyik felhasználó megbízható, kiváló választás. Másik hátránya, hogy a használatához szükség van a PHP CGI-modulra, különálló értelmezőként. Egy szó mint száz: ha egy csomó DNS-zónát kell pátyolgatnod, próbáld ki. A futtatásához szükséges: MySQL, PHP MySQL-támogatással (modulként és különálló értelmezőként is), Apache, Perl, Net::DNS Perl modul és OpenSSH (elhagyható). ☞ <http://probind.sourceforge.net>

graphical Process Statistics

A három évvel ezelőtti szemlézett programok között szerepelt a DHCPXD, amely továbbra is kiváló DHCP-ügyfélnek számít, illetve a gPS. Rengeteg kitűnő program van, amelyek minden szükséges adatot a rendelkezésedre bocsátanak a gépeden futó folyamatokról, a ps és a pstree parancssori eszközöket pedig mindenki ismeri. Ha viszont másra is kíváncsi vagy, akkor ezeket újra le kell futtatnod, a gPS azonban folyamatosan frissül. A más rendszereken futó folyamatokat is figyelemmel kísérheted, ha a csomagban található távoli gPS-lekérdezőt futtatod rajtuk. Futtatásához szükséges: libgtk, libgdk, libgmodule, libglib, libdl, libXi, libXext, libX11, libpthread, libstdc++ , libm és glibc. ☞ <http://www.gps.seul.org>

Linux Journal 2002. november, 103. szám

David A. Bandel

A biztonság – életstílus

Rövid, egyszerű recept a biztonság megteremtéséhez, biztonsági mentések készítéséhez, a rendszer figyeléséhez és játékokhoz.

Akik rendszeresen olvassák a rovatomat, tudják, hogy a biztonság a mániám. Ez alkalommal is a rendszer- és hálózati biztonságról olvashattok, de most a David-féle szentírást szeretném megismertetni veletek. Nem engedhetem meg magamnak azt a fényűzést, hogy egy nyugodt irodában üldögélve elmélkedjek a biztonságról, tőlem azt várják az ügyfeleim, hogy a lehető legmagasabb fokú biztonságot teremtsen meg a számukra, és mindezt úgy, hogy közben ne kelljen egy rakás pénzt elkölteniük. A receptem roppant egyszerű és rövid, de azt előrevetném, hogy aki jóvágású öltönyössé szeretne válni, szerintem foglalkozzon más területtel.

Először is állíts le minden nélkülözhető hálózati szolgáltatást (kezdve az `inetd`-vel, ha semmi olyan nem fut, amit ez indítana). Az ellenőrzésre megfelel a `netstat -tupan` parancs. Készíts minél szigorúbb `/etc/hosts.allow` szabályokat (az ellenőrzést a `tcpdchk` és a `tcpdmatch` segítségével végezheted el), majd a Netfilter szűrőabláját körültekintően használva tilts le mindent, amit nem akarsz kifejezetten engedélyezni.

Győződj meg arról, hogy a hálózati szolgáltatásokat nyújtó programok (Apache, Sendmail, `wu-ftpd`, `sshd` stb.) lehetőleg újabb változatát használod. A naplófájlokat minden nap nézd át, és keress bennük rendellenességeket (a megfelelő programokkal a szokásos bejegyzéseket kiszűrheted). A naplók jó esetben egy rendkívül biztonságos központi kiszolgálókra kerülnek.

Ellenőrizd, hogy a felhasználók biztonságos jelszavakat használnak-e. Ügyelj arra, hogy a nyilvánosság a tűzfalon keresztül csak a neki szánt hálózati részeket érje el, az érzékeny adatokat forgalmazó belső hálózathoz az Internetről ne lehessen hozzáférni. Használj VPN-t (FreeS/WAN, OpenSSH), és lehetőleg minden hálózaton keresztül továbbított adatot titkosíts (FreeS/WAN, OpenSSH, GnuPG).

Ne tulajdoníts túl nagy jelentőséget a dolgoknak, ha gond nélkül túlléphetnek rajtuk. A felhasználóid újra és újra megfognak lepni azzal, hogy mennyire nem értenek bizonyos dolgokhoz, és milyen sokat tapogatóznak a sötétben, amit te akár támadásnak is vélhetsz. Pedig nem az.

A recept segíteni fog, ám semmit nem ér, ha kifelejtjük az egyik nélkülözhetetlen összetevőt: a biztonság életstílus, nem pedig programok vagy tiltások halmaza. Mutass példát, és segíts másoknak munkájuk során a biztonságos eljárások kialakításában. Ők lehetnek szemed fényei, de legszörnyűbb rémálmaid szereplői is. Ha a fentiekhez vallásos elvakultsággal ragaszkodsz, nem fogsz betörés áldozatává válni. Ha mégis baj történne, húzd le a hálózatot, derítsd ki, hogy a támadó hogyan jött be (és mi volt a célja), majd hozd rendbe a rendszert, újra nézz körül javítások után, zárd be az ajtót, és lehetőleg torlaszold el az utat, ahol a betörő bejutott, majd irány a hálózat. A hatóságokat is lehet értesíteni, de ne ez legyen az első dolog, hacsak a nyomozásra és a vádemelésre fordított fáradság kifizetődőnek nem bizonyul.



hdup

Rendszeres biztonsági mentéseket kell készítened egy gépről? Mi lenne, ha mindent átmásolnál egy másik gépre? Esetleg titkosítani is lehetne az anyagot, nem? Más nem is kell. Csak add meg a kívánt műveletet – teljes vagy növekményes mentést is végezhetesz – egy beállítófájlban, majd hívd meg a programot havonta, hetente vagy naponta. A futtatásához szükséges: `glibc`, `bash`, `openssl` és `mccrypt` (elhagyható).

➔ <http://www.miek.nl/projects/hdup/hdup.shtml>

Nebula Cards

Akar valaki zsupgázni? Haverok és gépi játékosok ellen egyaránt játszhat. A különféle, négy játékkal folyó játékok – fekete macska, `bridzs` – megírása sem lehet ördögösség. A felület egy Java-kisalkalmazásnak köszönhetően webes. Futtatásához Java szükséges.

➔ <http://nebulacards.sourceforge.net>

User-Friendly IPTables Firewall

Ha gondot okoz az IP Tables szabályok elkészítése, próbáld meg az UIF segítségével. Ugyan a szabályok ellenőrzésére az UIF még nem képes, de legalább segít az elkészítésükben. A UIF beállítóállománya jóval egyszerűbb, mint maguk az IP Tables szabályok, tehát először írd meg a beállítófájlt, majd az UIF eme jóval érthetőbb bemenet alapján elkészíti a szabályokat. A program az összeköttetés alapú kapcsolatokat is kezeli. Futtatásához szükséges: Perl, `NetAddr::IP` és `Net::LDAP` Perl modulok, valamint értelemszerűen IP Tables.

➔ <http://lug.mfh-iserlohn.de/uif>

Linux Monitor

Remek segédprogram, ha a rendszer létfontosságú jellemzőit akarod szemmel tartani. Ha kezd megtelni valamelyik lemez, vagy nem fut valamelyik szolgáltatás, `syslog`-bejegyzést készít. A Linux Monitor más programot is el tud indítani, neked csak a kívánt időközök kell megadnod, a `linux_mon` mindent lát. Különösen akkor használható hatékonyan, ha nagyszámú gép küldi a jelentéseket egy központi naplónak. Futtatásához `libcrypto`, `libdl` és `glibc` szükséges.

➔ <http://sourceforge.net/projects/linux-mon>

Linux Journal 2002. október, 102. szám



David A. Bandel

(dbandel@pananix.com) jelenleg Panamában él, Linux- és Unix-tanácsadással foglalkozik. Társ szerzője a *Que Special Edition: Using Caldera OpenLinux* című könyvnek.



Java alapú webszolgáltatások

Végre megérkezett az első magyar nyelvű könyv a webszolgáltatásokról!

E könyvet tulajdonképpen a megjelenése előtt elég jól ismertem, ugyanis a most kiadott kötet angol nyelvű eredetijét (Building Web Services with Java – SAMS Publishing, 2001) már volt szerencsém elolvasni. A magyar kiadás végigkövetése meglepően könnyűnek bizonyult, pedig az eredeti kiadvány – nyelvezete, angol-sága folytán – nem tartozik az olvasmányosabb munkák közé.

A könyv a webszolgáltatások (eredeti nevén Web Services) 2001-es állapotáról ad részletes képet. Egészen friss munkának tekinthető, és ószre nem is lehetne ennél újabb szakkönyvet a magyar piacra dobni. A webszolgáltatások szabványai azonban hétről hétre, hónapról hónapra módosulnak. E fiatal, elképesztő mértékben fejlődő programfejlesztési területéről már 2002 vége felé is lehetne olyan könyvet írni, ami a szóban forgó könyv tartalmát több helyen megváltoztatná. Például a szolgáltatásleírás nevű verem esetében a szolgáltatás-összehangolás elnevezésű réteg szabványaira az IBM és a Microsoft időközben már közösen kidolgozta az egységes Business Process Execution nyelvet, felváltva ezzel a WSFL-t és az Xlangot. Létrejött a WS-I (Web Services Interoperability Organization) nevű szervezet is, hogy végre a teljes szabványosítás felé vezesse a ma még de facto szabványnak tekinthető, elsősorban az IBM-nél és a Microsoft.NET-nél alkalmazott megoldásokat. A Java oldaláról pedig mérföldkönek tekinthető a Sun Java WSDP (Web Services Developer Pack) nevű, néhány hónapja napvilágot látott nagy-szerű API-csomagja.

Ettől függetlenül mélyreható ismereteket szerezhetünk a webszolgáltatások üzenetkezelő, leíró és feltáró szolgáltatásairól. Sok igazán szemléletes és a kellő helyen felbukkanó ábrát találhatunk, főleg a „Webszolgáltatások feltárása” című fejezet adatszerkezeti ábrái jelenthetnek hatékony segítséget a UDDI megértésében. A könyv legnagyobb erőssége az alapossága. A fejezetek igényesek, a magyarázatok világosak – írjuk szakértelméről tanúskodnak. Másik erőssége, hogy az üzleti alkalmazások,

az üzleti logika oldaláról mutatja be a webszolgáltatások területét. Nem kell unalmas, szikár fejezeteket olvasnunk a SOAP-ügyfelek és -kiszolgálók működéséről, a szerzők a SOAP-ot rögtön e-üzleti környezetbe helyezik, példákkal megtűzdelve (lásd például „A SOAP használata e-üzleti alkalmazásokban” című fejezetet).



Adatok

Cím: Java alapú webszolgáltatások

Szerzők: Steve Graham, Simeon Simeonov, Toufic Boubez, Doug Davis, Glen Daniels, Yuichi Nakamura, Ryo Neyama

Kiadó: Kiskapu Kft.

ISBN: 963 9301 04 3

Ár: 7840 Ft

Az XML-ről ugyanakkor egy kicsit túlsokat és feleslegesen értekeznek. Gondoljunk arra, hogy a magyar könyvpiacon is elég sok XML-lel foglalkozó kiadvány található, és ezt a könyvet nem elsősorban az XML megismerése miatt fogjuk bújni. A SOAP legújabb megvalósításának, az Axisnak viszont több hely is juthatott volna.

Nagyon izgalmas a jövőről szóló utolsó fejezet, amelyben a könyv betekintést nyújt a szoftverügygynökök, a különböző szerkezetek és a webszolgáltatások összekapcsolhatóságának gondolatvilágába. A nagyszerű technológiai anyag egy

kicsit elnyomja az egyébként értékes javás fejlesztői anyagot. Hiányzik egy webszolgáltatás teljes, egy helyen történő bemutatása a kódolástól a bejegyzésen át az alkalmazásig. Kár, hogy egy adott téma taglalásakor a SkatesTown nevű cég nagyon jó példának tekinthető szolgáltatásának csupán részletei bukannak fel. Kiváló kódrészekkel szemlétett azonban a UDDI4J programozási felületről és a dinamikus keresésről szóló rész a könyv vége felé.

A magyar kiadvány nyelvezetéről, az angol szaknyelv magyarra ültetésének nehézségeiről is érdemes néhány szót ejteni. Rögtön bele lehetne kötni a *webszolgáltatások* fogalmába. A magyar kifejezés arra utalhat, hogy itt a web általános szolgáltatásairól van szó, ami a web megjelenésével közel egyidős lehet. Ám nem ez a szó valódi értelmezése, a Web Services kifejezés egy programtechnológia vagy webmodell összefoglaló neve, amellyel webes környezetben távoli eljárás-hívások által adott, jellemzően üzleti alkalmazásokat kezelhetünk, fejleszthetünk és köztük egyfajta kapcsolattartást biztosíthatunk. Bár több kisebb-nagyobb hibát találtam még, azért ne gondoljuk, hogy mindez a fordítás, illetve szakmai nivó összképét lényegesen befolyásolja. Mint az írás elején említettem, egy igencsak jól olvasható művel van dolgunk. Az olyan kifejezések, mint *computing as a utility* (számítógép mint eszköz), a *service discovery* (szolgáltatás feltárása), a *binding* (összekapcsolás) fordítása telitalálat.

A könyv webfejlesztőknek, Java-programozóknak, tanulóknak és szakembereknek, valamint minden, az Internet háttere iránt érdeklődő olvasó számára egyaránt ajánlható.

Gottdank Tibor

(gottdank@ilab.sztaki.hu) informatikus mérnök, kezdetben egy pesti kórház egyik rendszergazdája volt, majd Berlinben Java-fejlesztőként dolgozott. Jelenleg a webszolgáltatások elméletével és gyakorlatával foglalkozik az MTA-SZTAKI-nál, valamint Java alapú fejlesztői projekteken vesz részt.