

Beköszöntő



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

Zenél, csillog, milyen kis aranyos!

Lehet, hogy ebben a hónapban költözik a család többi tagja is a Linux elé? E havi témánk ugyanis a multimédia, a zenebona, a csilivili filmek világa. És nem gondolhatja egyetlen szomszéd sem, hogy a Mátrix harmadik részét halkán is lehet nézni, ugye? Vagy hogy Gandalf éppen csak a hallásküszöböt súrolva

üvölt, amikor gumitestű kínai harcművészeket megszegyenítő pörgés közepette lendületből levág néhány orkot? Ugye, nem? Mert ki kell próbálni a sok lejátszót, sőt el kell készítenünk a házi DVD-lemezünket is, és mindezt nézőközönség előtt kell bemutatunk! Inkább jöjjön át, szomszéd úr, nézze meg Ön is, hogy tavaly nyáron milyen idétlenül nézett ki egy szál fürdőnaciban! Azért gondolunk ebben a hónapban is a fejlődésre! Tovább ismerkedhetünk a SuSE és a Debian lelkivilágával. A házi atomreaktorok pártolói számára is akad egy érdekes cikk: az AMD64 világa kerül terítékre a 16. oldalon. Egy asztal alatt tartható igen drága légkondicionáló szerkezetet elemez ki *Glenn Stone*. Gondoljunk csak bele, milyen lehetőségek nyílnak ki az ember előtt, ha fél másodperc alatt annyi feladatot végeztethet el a lábvízforralójával, amennyit tízegy-néhány éve egy hónap alatt sem tudtak kiszámolni! Az új rendszerhez rendkívül nagy reményeket fűznek. Vajon beváltja-e a próbagép ezeket a reményeket? *Jó olvasást kívánok mindenkinek!*

Programvadászat

DVD-s programhegyek

Vezérfonalunk jellegéből adódóan sokféle, a DVD formátum kezeléséhez szükséges programot pakoltunk fel a korongra. Ezek legtöbbszörrel lejátszani lehet, de akad itt menükészítő, átkódoló, kereső is (bár igaz, hogy ezt a készítője CD-hez szánta, de mivel adatokról van szó, adat-DVD-n is képes keresni).

MPlayer

Ennek a méltán népszerű, magyar fejlesztésű médialejátszó programnak az 1.0pre3-as változata került fel a korongra. A telepítése szerencsére már nem kíván meg hatalmas szakértelmet – ha minden fejlesztői eszköz a rendelkezésünkre áll, akkor egyszerűen az alábbi lépésekkel telepíthetjük. Csomagoljuk ki a *Magazin/A_DVD_es_a_linux/MPlayer-1.0pre3.tar.bz2* fájlt, lépünk be abba könyvtárba, ahová kicsomagoltuk, majd adjuk ki a `./configure` parancsot. Figyeljük meg a parancs kimenetét – ha minden rendben lezajlott, valami ilyesmit kell látnunk:

```
Config files successfully
↳ generated by ./configure !
```

Érdeemes azonban a teljes kimenetet átnézni, mivel kellemetlen meglepetésként érhet bennünket, ha nem tudjuk, hogy bizonyos fájlformátumokat miért nem tudunk lejátszani. Ha grafikus felületet is szeretnénk, akkor futtassuk a `./configure --enable-gui` parancsot. Ezután szűrhetünk-bőrizhetünk kedvünkre.

Xine

A Xine szintén ugyanolyan egyszerűen telepíthető, mint az MPlayer, viszont ennél választhatunk a különféle grafikus programok közül.

A lépések ugyanazok – kicsomagolás:

```
./configure
make
make install
```

Először a *xine-lib-1-rc3a.tar.gz* fájllal tegyük ezt meg, utána pedig a grafikus felületek valamelyikével (*gxine-0.3.3.tar.gz*, *xine-ui-0.9.23.tar.gz*).

Ogle

Ez a program nem olyan nagy tudású, mint az előző kettő, de hasznos, ha csak DVD-t szeretnénk lejátszani. A telepítésével egy kicsit elbajlódttam, de nem volt vészes. Olyan csomagok hiányoztak, mint a *liba52*, a *libdvcss*, a *libjpeg*, az *a52dec*, a *libmad*, a *libxml2*. Ezek közül a legjobban a *libmad* telepítésével gyűlt meg a bajom, igaz, csak a kapkodás és figyelmetlenség miatt. Mindegyikből telepítsük a `-dev` csomagot is, ezek szükségesek a fordításhoz. Ha kivágtuk magunkat a csomag függőségi erdejéből, akkor már semmi nem állhat utunkba a DVD-mozik élvezetében. A már *Marcel Gagné*-tól is megszokott (és az előbbieken énáltalam is bemutatott) ötlépéses telepítési folyamat után használatba is vehetjük ezt a programot.

VLC

Talán ez a program a legnagyobb falat, s tudását nézve sem nevezhető pehelysúlyúnak. Mindenfélté lejátszhatunk vele, ami valahol – merevlemezen, az éterben, DVD-n stb. – megtalálható.

Krystal Drop

Ha valaki úgy érzi, hogy itt az ideje a kikapcsolódásnak és a Frozen Bubble-t már meguntta vagy számtalanszor végigjátszotta, az most megpróbál-



kozhat a Krystal Droppal. Mind rpm, mind deb csomagokban is megtalálható a korongon, így ennek telepítése nem olyan „bonyolult”, mint a lejátszóké.

Elég egy egyszerű `rpm -i` vagy `dpkg -i` parancs, és máris játszhatunk, ha kiadjuk a `drop` parancsot.

Rendszermag

Tessék megismerkedni a 2.6-os rendszermagsorozattal – immáron megbízható, és akár éles kiszolgálókra is használható minőségű. Azonban – mint a 2.4-es sorozatnál is volt – nem valószínű, hogy a mostani kiszolgálóján mindenki hanyatt-homlok lecserezné a rendszermagot. Mindenesetre érdemes egy kicsit foglalkozni a beállításokkal és a fordítással, mivel a jövő egyértelműen ezen a sorozaton keresztül vezet!

A 2.4-es rendszermag fontos hibajavításait tartalmazza a 2.4.24-es mag a Rendszermag/2.4-es könyvtárban.

Condor

Ha valaki a 38. oldalon kezdődő cikkben leírtak alapján megkívánja és a rendelkezésére áll néhány szabad kapacitással rendelkező számítógép, az bátran fogjon hozzá a saját „szupergépének” megépítéséhez a Condor segítségével.

Gentoo

Sajnos a Gentoo Linux első korongjára hibásan került fel néhány csomag, így ezek hibátlan változatait most adjuk közre.



Csontos Gyula

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

Office? – az idén csak Open

A Microsoft minden igyekezete ellenére az izraeli kormányzat tartja magát ahhoz a döntéséhez, hogy ebben az évben nem vásárol új alkalmazásokat a redmondi cégtől. Az izraeli kormány a költségek lefaragása miatt döntött így, de arról nem szólnak, hogy jövőre is tartják-e magukat ehhez a takarékoság miatt választott megoldáshoz. Ha gonoszkodni akarunk, még arra is gondolhatunk, hogy az izraeli állam minden számítógépén a legújabb Windows és a legújabb Office fut, így 2004-ben újabb változatok híján nincs is mit megvenniük. A hosszú távon kétes kimenetű vita a hírek szerint azon tört ki, hogy nem volt biztos, hogy a miniszteriumoknak lesz-e módjuk arra, hogy a Microsoft Office-csomagból csak a számukra szükséges összetevőket vásárolják meg. A Microsoft saját véleménye szerint mindig is biztosította ezt a lehetőséget, a kormányzat képviselői viszont megmakacsolták magukat. Izraelben a Microsoft-képviselő éves forgalmának 3–4 százalékát teszik ki a kormányzati vásárlások, így túl nagy érvágás semmiképpen nem éri a céget, főleg ha – ismét legyünk rosszindulatúak – az elmaradt vásárlásokat jövőre pótolják. Az viszont tény, hogy az izraeliek a Sun és az IBM közreműködésével elkészítik az OpenOffice héber változatát, így ha más nem is, a nyílt irodai csomag mindenképpen nyer az ügyön.

Kiszolgálócska olcsón

Az ICP Electronics újabb taggal bővítette az előtelepített linuxos kiszolgáló klubját. A gép



előlapját szemlélve különféle szórakoztató elektronikai készülékek juthatnak eszünkbe, a hátlapjára pillantva azonban már sejthetjük, itt valami hálózatos dologról van szó. A cég állítása szerint sok – jellemzően kis-méretű – vállalkozásnak gondot okoz egy elektronikus levelezésre használható kiszolgáló összeállítása és üzemeltetése, így készen kapható megoldásukkal őket célozták meg. A WMS-2208R jelzésű készülék a POP3, az SMTP és az IMAP protokollt támogatja, illetve webes levelezési

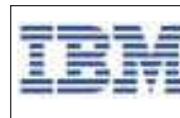
lehetőséget is kínál használóinak. Emellett segíti őket a kéréstelen levelek kiszűrésében, távollétet jelző üzenetek küldését teszi lehetővé, a nemkívánatos küldők kiszűrését, címtárak létesítését, továbbá egyéb felügyeleti szolgáltatásokkal segíti a levelezés lebonyolítását. Mindemellett hálózati kapcsolóként, forgalomirányítóként, csomagszűrő tűzfalként és nyomtatókiszolgálóként is használható. Maga a dobozka egy 300 MHz órajelű National Semiconductor GX1-es processzort és 128 MB memóriát tartalmaz, s az adatok tárolására legfeljebb két IDE felületű merevlemez képes használni. A rendszer betöltését 32 MB flashmemóriából végzi – ennek a megoldásnak az az előnye, hogy a készülék alapbeállításait egyetlen gombnyomással vissza lehet tölteni. Az adatbiztonságot szolgálja, hogy a géphe egy második merevlemez is szerelve RAID-tömböt is létrehozhatunk, illetve távoli gépre is készíthetünk ütemezett biztonsági mentéseket. ☞ <http://www.ia.ieiworld.com>

Átvették őket

A Progeny Linux bejelentette, hogy díjfizetés ellenében igénybe vehető támogatási szolgáltatást indít korábbi Red Hat Linux-előfizetők részére. A Red Hat idén január 1-jével beszüntette régebbi terjesztéseinek a támogatását, ezzel egy időben indult meg a Progeny szolgáltatása a 7.2-es, 7.3-mas és 8.0-s terjesztésekre vonatkozóan. A 9-es terjesztés támogatása május 1-jével fejeződik be a Red Hat részéről, ekkor azonban bővít a Progeny is, és az ilyen terjesztések használóit is átveszi. Az előfizetők a Progeny Platform Services nevű megoldása révén értesülhetnek a rendszerüket érintő biztonsági hibákról, illetve jutnak hozzá a megfelelő frissítésekhez is. A Platform Services gondoskodik az alkalmazások csomagolásáról, a függőségek és az ütközések figyeléséről, kezeléséről, és a debianos apt-vel (Advanced Package Tool) együttműködve biztonságos módon telepíti a biztonsági javításokat és a programfrissítéseket. Az előfizetés díja gépenként és havonta öt dollár (körülbelül 1100 forint), havi 2500 dollár fejében pedig tetszőleges számú géppel igénybe vehető a szolgáltatás. ☞ <http://www.progeny.com>

Nekik 3415

2003-ban az IBM összesen 3415 szabadalom jogát nyerte el az Egyesült Államokban, amivel megdöntötte az egy év alatt beadott szabadalmak mennyiségi világrekordját. Az IBM mögött a második helyezett több mint 1400 szabadalommal maradt le, vagyis a Nagy Kék e tekintetben lóhosszal vezet vetélytársai előtt. Az IBM 11 éve vezető az újítások terén, ez alatt több mint 25 000 szabadalmat jegyeztetett be, olyan világcégeket utasítva maga mögé, mint a HP, a Dell, a Microsoft, a Sun, az Oracle vagy az Intel. A tavalyi év fontosabb IBM-es felfedezései közül néhány: önjavító számítógépek; önműködő üzleti folytonossági és helyreállító rendszerek; weblapok előtöltése hordozható eszközökön a felhasználó választását megjósoló eljárással; gyártásoptimalizálási módszerek; hitelkártyák illetéktelen használatát megakadályozó megoldások. ☞ <http://www.ibm.com>



Jön az E28

Az E28 nevű kínai cég hazájából továbblépve Hongkongban is megkezdte linuxos intelligens telefonjainak a forgalmazását. Számunkra természetesen nem ez az érdekes fejlemény, hanem az, hogy a cég terjeszkedik, és tervei között Európa és az Egyesült Államok meghódítása is szerepel – ennek érdekében már meg is kezdték a tárgyalásokat a leendő üzletfelekkel. A cég E2800 jelzésű telefonját 130 ezer forint körüli összegért vesztegetik, első sorban – mint az árból is érzékelhető – az üzleti vásárlókat célozva meg.

A telefon személyi adatkezelő szolgáltatásokat nyújt, képes a kézírásfelismerésre, érintőképernyővel és kamerával rendelkezik, illetve SD kártyával memóriája is bővíthető. Operációs rendszere 2.4-es rendszer-magra épülő Linux, ami két ARM9-es processzoron fut, 32 MB flash és 32 MB RAM memóriával gazdálkodik. A készülék kijelzője 240×320 képpont és 4096 szín megjelenítésére képes, kamerája 110 ezer képpontot rögzít. A 900/1800 MHz-es GSM hálózatokon használható telefon GPRS, USB és infravörös kapcsolat létesítésére képes, adatkezelő alkalmazásai pedig Outlook megfelelők. A kínai piac azért nem lehet túl sovány falat, ha a Samsung is hasonló terméket jelentet meg rajta. A világcég SCH-i591 jelzésű készüléke ugyancsak Linuxot futtat, a kijelzője is versenytársáéhoz hasonló jellemzőkkel bír, de a belsejében több – kétszer 64 MB – memóriát és egy 400 MHz-es Intel XScale processzort találunk. A személyi adatkezelő alkalmazásokkal, médialejátszót, fotóalbumot, szövegszerkesztőt, táblázatkezelőt, azonnali üzenetküldő programot tartalmazó gép nem titkoltan a windowsos intelligens telefonok babérjaira tör, tudását nézve nem is esélytelenül.

➔ <http://www.e2800.com.cn>

➔ <http://www.samsung.com.cn>

Arcfelismerés az MPEG-7 szabványban

A Moving Picture Experts Group (MPEG) Committee az NEC és a Samsung közös előterjesztése alapján úgy határozott, hogy a várhatóan tavasszal megjelenő MPEG-7 szabvány arcfelismerő megoldást is tartalmazni fog. Az MPEG-7 különféle tartalomlekerdező módszerek leírását tartalmazza majd, ezek közé tartozik az arcfelismerés (advanced face recognition, AFR) is. Az AFR lehetővé teszi, hogy nagy sebességgel, a korábbiaknál pontosabban keressük elő például egy filmből az adott személyt tartalmazó jeleneteket. Alkalmazási területként nemcsak a legkézenfekvőbb szolgáltatás, a mozifilmekben való keresés merülhet fel, de a házi videózás és a biztonsági kamerás rendszerek is teljesen újfajta szolgáltatásokat kínálhatnak majd.

➔ <http://www.iso.org>

Digitális élménytábla

A Pepper bemutatta Pepper Pad névre keresztelt készülékének első mintapéldányát. A cég nem fog ilyen készülékeket forgalmazni, ők csak Pepper Keeper Software Suite név alatt a rajta futó programcsomagot adják, de a kezdeményezés így is érdekes. A táblaszerű eszközön MontaVista Linux



fut, rajta előtelepítve azonnali üzenetküldő program, böngésző, fényképalbum, médiagyűjtemény és egyéb hálózati és kapcsolattartó alkalmazások találhatók.

A Pepper Pad és a Pepper Keeper Software Suite együtt alkotják a Pepper Keeper Wi-Fi Suite csomagot, ezt a forgalmazók OEM kiserelésben vehetik meg, illetve jogot vásárolhatnak a gyártására. A Pepper termékével első sorban a hálózati szolgáltatásokat nyújtó és a digitális tartalmat szolgál-

tató vállalatokat célozza meg, akik gyorsan, költséges fejlesztések nélkül juttathatják ügyfeleiket könnyű hozzáféréshez.

Az első ránézésre inkább játékszernek tűnő tábla figyelemre méltó adottságokkal rendelkezik. 400 MHz-es Intel XScale processzora még nem újdonság, a hozzá társított 256 MB memória ellenben már szokatlan, az 5 GB-nyi tárhelyet kínáló 1,8"-os PCMCIA meghajtóról, a 20 bites sztereo hangrendszerrel és az ujjlenyomat-olvasóról már nem is beszélve. A gép Bluetooth és IEEE 802.11b csatolóval rendelkezik, képernyője 600×800-as felbontást tud, tömege viszont komoly teljesítménye ellenére is alig haladja meg az 1 kg-ot.

➔ <http://www.pepper.com>

Mindent egy helyről – Novell

A Novell ez év januárjában véglegesítette a SuSE Linux felvásárlását. A felvásárlást, amelynek során a Novell készpénzben 210 millió dollárt



fizet a SuSE Linuxért, még ősszel jelentették be, azonban a művelet befejezésére és a versenyhatóságok jóváhagyására várni kellett. A SuSE Linux a továbbiakban a Novell egyik üzleti egysége lesz, saját értékesítési és marketingcsapattal.

A Novell linuxos termékvonala ezzel teljessé vált, a korábbi felvásárlások nyomán asztali és kiszolgáló operációs rendszert, grafikus felületet, adatbázist egyaránt tud kínálni ügyfeleinek, mindezekhez háttérként világméretű tanácsadói, támogatási és képzési hálózatot biztosít.

A felvásárlás befejezésével egy időben a Novell kártérítési- biztosítási jellegű szolgáltatást is indít ügyfelei számára. A szolgáltatás lényege az, hogy a Novell a SuSE Linux összetevőinek jogtisztaságát esetleg vitató személyekkel, szervezetekkel szemben „tartja a hátát” ügyfele helyett. A jogtisztaság megőrzését szolgálja az is, hogy a Novell megállapodott a unixos jogokat birtokló SCO-val a jogvédett programrészek használatáról.

➔ <http://www.novell.com>

➔ <http://www.suse.com>

Megújuló Magnia-gépek

A Toshiba a továbbiakban Astaro Security Linuxszal szállítja Magnia SG25, SG30 és Z310m kiszolgálóit.



A kis- és közép-vállalkozásoknak szánt készülé-

kekre eddig a 7-es Red Hat Linux egy, a Toshiba által módosított változatát telepítették, ezt váltja fel a gyakorlatilag változások nélküli Astaro Linux. A frissített Magniak többek között képesek a levélszemét felismerésére, Radius és LDAP-támogatást nyújtanak, lehetővé teszik a webes forgalom cím-alapú szűrését, állapot-alapú tűzfalat futtatnak, védelmet kínálva a legfontosabb támadási módszerek ellen. Támogatják a VPN-eket: VPN-átvitelük legnagyobb sebessége 115 Mbit másodpercenként, míg tűzfaluk ugyanennyi idő alatt 735 Mbit adatot képes átvizsgálni. Ami változatlan, az a kiszolgálók webes beállító és felügyeleti felülete, amely minimális hozzáértéssel rendelkező felhasználók számára is könnyen kezelhetővé teszi őket. Az Astaro Linux 4-es változata 30 napig ingyenesen próbálható ki.

☞ <http://www.astaro.com>

☞ <http://www.toshiba.com>

Sun: programozunk együtt!

A Sun Microsystems együttműködésre kérte fel az IBM-t és a Crayt egy új számítógépes nyelv megalkotására. Az új nyelv célja az lenne, hogy növelje a tudományos és műszaki számításokra készített, s mint a leendő fejlesztők alapján sejthető, inkább nagygépeken futó alkalmazások teljesítményét és termelékenységét. A Sun a Java megalkotásával jókora tapasztalatra tett szert a géptípustól független, alacsony szintű megoldások fejlesztésében, s most ezt a tudását szeretné továbbörökíteni egy újabb területen is. A létrejövő, és a Sun szándékai szerint nyílt megoldás valamilyen módon a meglévő programozási nyelveket is támogatná, igazi tudása azonban az új, egyelőre még nem létező nyelv használatával mutatkozna meg. Az, hogy az IBM és a Cray mennyire lesznek vevők az ötletre, még bizonytalan, hiszen a nagy teljesítményű gépek építését célzó az amerikai védelmi kutatások keretein belül mindkét cégnek megvannak a maga tervei saját programozási nyelvek és operációs rendszerek fejlesztésére.

Intel HDA

Az Intel korábban Azalia kódnév alatt fejlesztgetett hangmegoldása végleges nevet nyert: Intel High Definition Audio. Jól esne egy kicsit köszörülni a nyelvemet azon, hogy csak egy koc-kafejű mérnök cserélheti le a dallamos Azalia nevet egy ilyen semmitmondó megnevezésre, de ez már legyen az ő bajuk, foglalkozunk inkább az új megoldás jellemzőivel. Létrejöttének célja, hogy a szórakoztatóelektronikai készülékeknek megszokott színvonalú hangkezelő alrendszerrel és egyéb szolgáltatásokkal – modem, adatkapcsolatok – lássa el nemcsak a személyi számítógépeket, de a hordozható eszközöket és magukat a szórakoztatóelektronikai berendezéseket is. Az Intel HDA szerkezete a bővíthetőséget szolgálja, így nemcsak zene- és DVD-lejátszáshoz kínál majd kiváló hangrendszert, de hangtovábbító és egyéb kapcsolattartási alkalmazásokhoz is. Az Intel HDA 192 kHz-es mintavételezésre, 32 bites felbontásra és többcsatornás hangkezelésre lesz képes. Jelenleg a 0.9-es változatszámot viseli, végleges változata ez év közepére várható, ezt követően ingyenesen lesz hozzáférhető, illetve bekerül a szintén az idei év első felében megjelenő Grantsdale lapkakészletbe is.

☞ <http://www.intel.com>

ATI és nVidia illesztőprogramok

1.0-5328 jelzéssel új linuxos illesztőprogramot adott ki grafikus kártyáihoz az nVidia. Az illesztőprogram a legújabb GeForce FX és a Quadro FX alapú kártyákat is támogatja, illetve javított és új grafikai-megjelenítési lehetőségeket kínál.

Ugyancsak kisebb javításokat tartalmaz az ATI legújabb, 3.7-es változatú illesztőprogramja, amelyet az Xfree86 4.1.0, 4.2.0 és 4.3.0 változatához tettek elérhetővé.

☞ <http://www.nvidia.com/object/linux.html>

☞ <http://www.ati.com/support/driver.html>



Medgyesi Zoltán

(mz@rettesoft.hu)

A Linuxvilág hírszerkesztője. Szabadidejét legszívesebben a barátnőjével tölti, szeret autózni és bográcsban főzni.



Ez elméletileg valóban nem sérti a GPL szerződést, végső soron mégis olyan helyzetet teremthet, hogy egy GPL jogú fejlesztést (például a rendszermagot) nem lehet egy másik GPL jogú projektben felhasználni.

© Kiskapu Kft. Minden jog fenntartva

Újdonságok a rendszermagfejlesztésben

Az Intel néhány új eseményfigyelő folttal (notification patches) állt elő, amelyek a már létező folyamatindítási leíró kapcsok mellett a folyamatok befejezésekor is nyújtanak leíró kapcsokat (profiling hook). Az Intel műve sokat segíthet az olyan programok továbbfejlesztésekor, mint az Oprofile, Perfmon és a VTune. Bár az ilyen kapcsok felhasználása könnyen hozzásegítheti a forgalmazókat, hogy a GPL szabadalmi megkötéseit megkerüljék, úgy tűnik, hogy az Intel alkotása mentes ettől a gondtól. *Frederic Rossi* továbbra is dolgozik saját asynchronous event mechanism (AEM) nevű, független eseményjelző kódján. Az AEM segítségével általános célú, a felhasználói térben futó programok is bejegyezhetik magukat a rendszereseményekre és megadhatják a saját eseménykezelő készletüket. A folt fő célja, hogy a rendszermagból gyorsan lehessen adatot visszaadni a felhasználói térbe. Bár az Intel munkája és az AEM közvetlenül nem kapcsolódik egymáshoz, érdekes adalékkal szolgálnak a Linux-fejlesztés irányáról: egyre jobban növekszik az egyes szintek közötti párbeszéd.

A DSI Development csapat által fejlesztett **digsig** nevű önálló modul segítségével még a futtatás előtt ellenőrizhető a futtatható állományokba ágyazott digitális aláírás. Amennyiben a fájl megváltozott, a **digsig** megakadályozza az indítását. Számos alkalmazásnak előnyére válhat ez, például a férgek elleni védelemben vagy a digitális jogok kezelésekor. A Trusted Computing tanúsítvány módszerével ellentétben a **digsig** bizonyítási folyamata mindvégig a géptulajdonos irányítása alatt marad. Nem fordulhat elő például, hogy egy távoli rendszer a gépünkön futó **digsig** modulra támaszkodva ellenőrizze, hogy a távoli rendszer által elfogadott program megadott változatát futtatjuk-e.

Manapság minden AMD Athlon processzorban van egy olyan hiba, amely az utasítás-előlekérésen (prefetch) keresztül bizonyos körülmények közt bizonyos címtartományokon képes memóriakezelési hibákat okozni. Ez nagyon hasonlít a hírhedt f00f Pentium hibához, tekintve, hogy mind a kettő alkatrészhiba, amelyet az operációs rendszernek kell kijavítani. A jó hír az, hogy ez a hiba a legtöbb esetben nem okoz igazán nagy károkat. Ennek a rendellenességnek az előfordulása nagyon ritka, valószínűleg ez az oka annak is, hogy ennyi ideig rejtve tudott maradni. *Andi Kleen* elkészítette a linuxos javítást a 2.4-es és 2.6-os változatokhoz, az AMD fejlesztői pedig jelezték, hogy megoldása együttműködik majd a jövőbeni Athlon nemzedékekkel, amelyekben ezt a hibát már kijavították. Ugyanakkor *Andi* szerint néhány Athlon-változatban, például a K7-ben, nemigen lehet kijavítani ezt a hibát, így aztán az új rendszereken még igen sokáig

szükség lehet erre a hibakerülő megoldásra.

Az SGI egy GPL meghajtót adott ki Altix soros konzoljához, azonban a szerzői jogi megjegyzésben felhívja rá a figyelmet, hogy az SGI a jogot a kódban használt valamennyi szabadalomra fenntartja. Bár a Szabad Szoftver Alapítvány (Free Software Foundation) jelezte az SGI felé, hogy ez elméletileg valóban nem sérti a GPL szerződést, végső soron mégis olyan helyzetet teremthet, hogy egy GPL jogú fejlesztést (például a rendszermagot) nem lehet egy másik GPL jogú projektben felhasználni. Ezek után felmerül a kérdés, hogy ilyen feltételek mellett a kódot egyáltalán be szabad-e engedni a rendszermagba? *Alan Cox* úgy döntött, hogy mind az SGI, mind az FSF oldalán a végére jár a dolognak, és megbizonyosodik afelől, hogy az FSF szerint ez valóban nem tekinthető a GPL szerződés megsértésének, illetve hogy hajlandó-e az SGI módosítani az álláspontját.

Matthew Wilcox elkezdte a rendszermagfejlécek két csoportba sorolását: vannak olyanok, amelyeket láthatnak a felhasználói térben futó források, és olyanok is, amelyeket nem. Állandóan gondot okoz, ha a felhasználói alkalmazások rossz rendszermagfejlécekre támaszkodnak. A rendszermag növekedésével és változásával párhuzamosan a fejlécek is változhatnak, s emiatt a felhasználói programok megsérülhetnek. A baj orvoslásához szigorú elkülönítést kell bevezetni. *Matthew* munkája éppen ezt könnyíti meg. Műve másokat is megihletett – többek között *Erik Andersent* –, akik a 2.6-os előkészítéseként megkezdték az alkalmazott elnevezési szabályok felülvizsgálatát.

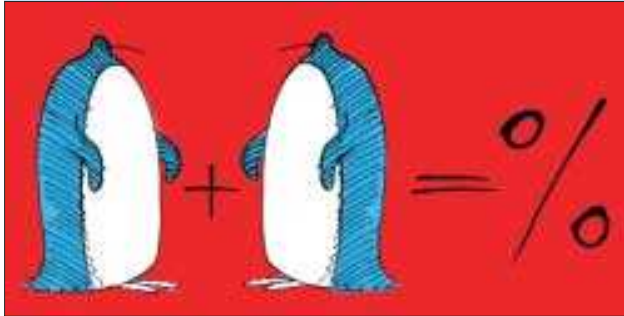
Megkezdődött a rendszermag kódjának *arch/* könyvtárhoz hasonló, de újfajta rendszerezése. Ez az új rendszer azonban géptípusonkénti szétválasztás helyett fordítóprogramonként különíti el a forrásokat. Amióta az inter „ecc” (a GCC után) a második olyan fordítóprogrammá vált, amelyik sikeresen fel tudja építeni a Linux-rendszermagot, apró hibák kezdtek jelentkezni, így a fejlesztők olyan foltokat kezdtek el küldözgetni, amelyek az egyik vagy a másik fordítóprogram sajátosságaira alapoznak. A nemrégiben lezajlott vita után, amely a GCC által támogatott, de az ecc számára érthetetlen beépített assembly nyelv körül zajlott, *Linus Torvalds* végül úgy döntött, hogy az egyes fordítóprogramokhoz tartozó foltok tárolására fordítónként külön könyvtárakat kell létrehozni. *Sam Ravnborg* és *David Mosberger*, úgy tűnik, elsőként döntött az új rendszer támogatása mellett, és csomagjaik beküldésével megkezdték az átállást.

Zack Brown

Linux Journal 2003. január, 117. szám

Linux-index

1. A Linux 2.6-os rendszermag-változtatások száma a 2002 közepe és 2003 szeptember közepe alatti fejlesztési időszakban: **12 479**
2. Az összeolvasztások (merge) száma a rendszermagban, nem beleszámítva az apply-by-proxy foltokat: **2386**



3. Az összeolvasztások becsült értéke napra lebontva: **9**
4. A rendszermag fejlesztésén dolgozók száma: **554**
5. A változtatásoknak a rendszermag fejlesztésén dolgozók legjobb tíz százalékára eső százalékos aránya: **78**
6. A változtatásoknak a rendszermag fejlesztésén dolgozók legjobb húsz százalékára eső százalékos aránya: **90**
7. A Linux százalékos részesedése a kereskedelmi kiszolgáló és operációs rendszerek (SOE) piacán 2002-ben: **23,1**
8. A Microsoft százalékos részesedése a fizetős SOE piacon: **55,1**
9. Az új szabadalmak száma millióban az SOE környezetben 2002-ben: **5,7**
10. A Linux részesedése az ügyféloldali operációs rendszerek (COE) kategóriájában 2002-ben: **2,8**
11. A Macintosh részesedése a COE kategóriájában 2002-ben: **2,9**
12. Egyéb operációs rendszerek részesedése a COE kategóriájában 2002-ben: **0,5**
13. A Microsoft részesedése a COE kategóriájában 2002-ben: **93,8**
14. A Linuxon és a Windowson kívül ennyi operációs rendszer dicsekedhet piaci növekedéssel 2002-ben: **0**

Források

1–6.: *Linus Torvalds*

7–14.: International Data Corp.
(Nemzetközi Adatszolgáltató Vállalat)

Linux Journal 2004. január, 117. szám

Neumann Kupa

A 2004 februárjáig tartó Neumann-emlékévhöz kapcsolódó program-sorozat keretében kerül sor az Informatikai és Hírközlési Minisztérium által támogatott Neumann Kupa



Internetes Sakkbajnokságra, amelynek célja az internet lehetőségeinek megismertetésén túl a világszínvonalú magyar sakkozás népszerűsítése. 2004. január 14-én kezdetét is vette a Neumann Kupa. A verseny különlegessége, hogy a megmérettetés során sakkprogramok használata is engedélyezett. Így ember és gép együtt szállhat csatába egymás ellen. Ezzel a rendezvénnyel a 100 éve született magyar tudós–matematikusra, Neumann Jánosra emlékezünk.

A nevezés ingyenes. A 2004. február 11-ig tartó bajnokságon korra és nemre való tekintet nélkül bárki részt vehet, aki regisztrált játékosként a gamer oldalon, valamint kitöltötte a versenyzői adatlapot. Díjazásban kizárólag az részesülhet, aki a következő három ismérv valamelyikének megfelel:

- magyar állampolgárság,
- határon túli magyar, amit „magyarigazolvánnyal” igazolni tud,
- a játékos neve mellett a FIDE-listán „HUN” megjelölés található.

A verseny díjai a következőképpen festenek:

- I. díj: 200 000 Ft
- II. díj: 100 000 Ft
- III–IV. díj: 50 000 Ft
- V–VIII. díj: 25 000 Ft

A verseny 2–4. helyezettje érdmédíjazásban részesül, míg a verseny győztese a Neumann Kupát nyeri el.

- ➔ <http://www.chess.hu/hu/index.php3>
- ➔ <http://www.gamerteam.hu/neumannkupa/index.html>

Szemezgettünk sajtóhíreinkből

A Debreceni Egyetem Angol-Amerikai Intézetének Oracle iLearning alapú távoktatási programja 2003-ban első helyezést ért el az Európai Bizottság és az Oktatási Minisztérium által létrehozott Európai Nyelvi Díj pályázaton.

A 2003-ban lezárult pályázati fordulón számos igen színvonalas idegen nyelvi program közül négy került díjazásra. Az élen végző Debreceni Egyetem Angol-Amerikai Intézete után a második helyet a Nyíregyházi Főiskola Eötvös József Gyakorló Általános Iskola és Gimnázium, míg a harmadik helyet a Nyelviskolák Szakmai Egyesülete és a Discimus Alapítványi Közgazdasági Szakközépiskola – Discimus Üzleti Iskola szerezte meg. Az elmúlt öt évben Európa-szerte közel száz intézmény vehette át a nyelvtanításban elért kiemelkedő eredményeiért járó díjat.

A díjat az Európai Bizottság és a magyar Oktatási Minisztérium alapította 1998-ban az Európai Unió tagállamainak kezdeményezésére. Célja, hogy hivatalos elismerésben részesítse az idegen nyelvek tanítása terén megvalósuló, példaértékűnek minősülő és innovatív programokat az oktatás és képzés bármely szintjén.

A díjnyertes program hátterét, az Oracle iLearning távoktatási keretrendszerét a Debreceni Egyetem Angol-Amerikai Intézete 2002 őszén vezette be, a kapcsolatfelvételtől számítva csupán négy hét alatt. Az új rendszer tartalmában és formájában újszerű képzési forma, amelynek bevezetésében lényeges szempont volt, hogy több tekintetben is költséghatékony megoldásról van szó, amely például kiváltotta a fénymásolt tananyagok postázását, támogatja a hallgatói csoportok együttműködését és a csoportmunkában tanító oktatók koordinálását, valamint segítségével az oktatás áttekinthetőbbé és hatékonyabban nyomon követhetővé vált.

Az Oracle iLearning rendszere arra épül, hogy az Oracle E-Business Suite alkalmazásrendszerének különböző moduljai kapcsolatban állnak egymással, így az összes szükséges adat birtokában egyszerűsödik az oktatás folyamata. Az oktatási anyagok online módon érhetőek el, az elvégzett tanfolyamok és az ezekkel járó adatok önműködően bejegyzésre kerülnek, a tanfolyamok a jelentkezők számától függően alakíthatók és időzíthetők, a tesztek összeállítását pedig adatbank segíti. Az alkalmazással egyetlen egység vagy akár számos, földrajzilag elkülönülő telephellyel rendelkező szervezet önálló oktatási rendszere is kialakítható. Az Oracle iLearning elterjedt ipari szabványoknak megfelelő alkalmazása a kívánt anyagokat képes több tartalomszolgáltató anyagaiból önműködően összeválogatni és a megadott szempontok alapján összeállítani.

➔ <http://www.oracle.com>

A mi Európánk

December vége óta érhető el a világhálón „A mi Európánk” című projektgyűjtemény. A módszertani leírás mellett 28 projekt részletes feldolgozása is helyet kapott az oldalon.

„A mi Európánk” című pályázat célja az volt, hogy a közoktatás 5–12. évfolyamán a tanárok egy olyan elektronikus könyvhöz jussanak hozzá, amely gyakorlati, azonnal felhasználható segítséget nyújt az Európai Unióhoz kapcsolódó témák önálló vagy kollégákkal történő közös feldolgozásához. A tavaly májusban meghirdetett pályázatra 107 pedagógus, szakmai műhely jelentkezett, közülük harminc került kiválasztásra, ezek közül végül 28 került be a gyűjteménybe. A tanulók életkorának megfelelő feladatok játékosak, változatosak: kvíz, keresztrejtvény, makettépítés, vaktérképek és EU-activity egyaránt helyet kap közöttük. Valamennyi projektleírásnak központi eleme a képességek és készségek fejlesztése. Az elektronikus tanári kézikönyv alkalmazói-nak, továbbgondolónak, fejlesztőinek tájékozódását részletes irodalom- és hivatkozásjegyzék segíti.

➔ <http://www.om.hu/amieuropank>

Széles sávú internet kistelepüléseknek

Az Informatikai és Hírközlési Minisztérium (IHM) „Széles sávú Internet infrastruktúra kiépítésének és a szolgáltatás beindításának támogatása Magyarország üzletileg kevésbé vonzó településein” című pályázatának (a pályázat jele: IHM-HHÁT-2) a keretösszegét az eredetileg tervezett 600-ról 900 millió forintra emelte fel. A támogatást elnyert 35 pályázó – köztük 15 kábeltelvíziós társaság és 16 vezeték nélküli szolgáltató – több mint 400 kis lélekszámú településen összesen csaknem 2 milliárd forint értékű projekt megvalósítására kap lehetőséget. A nyertesek listája a minisztérium honlapján tekinthető meg.

➔ <http://www.om.hu/>

Vodafone Live! játékok itthonról is

A Sun Microsystems bejelentette, hogy a Vodafone a java.com-ot - a Java technológia fogyasztói webhelyét - választotta Vodafone live! Java alapú játékok bemutatásához. Így egyetlen központi helyen érhetőek el játékok száza: megtalálható például a Tomb Raider: Osiris Codex, a Tetris és az Urban Freestyle Soccer is.

A java.com-ról közvetlen hivatkozások nyílnak a Vodafone live! szolgáltatást kínáló 14 országra (Ausztrália, Egyiptom, Németország, Görögország, Magyarország, Írország, Olaszország, Japán, Hollandia, Új-Zéland, Portugália, Spanyolország, Svédország és az Egyesült Királyság) felé. A jelenlegi előfizetőket egyszerű útmutatások segítik a játékok letöltésében. Az együttműködés a mobil Java-alkalmazások fejlesztőit is segíti, hiszen termékeik nagyobb nyilvánosságot kapnak.

MTE-kerekasztal az online vásárról

A fenti címmel rendezett kerekasztal-beszélgetést a Magyar Tartalomszolgáltatók Egyesülete 2003. december 16-án a Független Médiaközpontban, amelyen egy, a hazai elektronikus kereskedelem kínálati oldalát felmérő kutatás eredményeit is ismertették. A kutatást egyébként az MTE és a RE:CALL Tanácsadó Kft. az Informatikai és Hírközlési Minisztérium megbízásából közösen készítette.

Karácsonyi vásárlás a karosszékéből

Az elkészült tanulmány 289 bevásárlókosaras rendszerű, a karácsonyi ajándékozás során szóba kerülő árukat forgalmazó e-bolt kínálatát és szolgáltatását vizsgálta meg részletesen. A kutatásban megvizsgált boltok összesen 1,5 millió terméket kínálnak húsz különböző, a karácsonyi ajándékozáshoz köthető termékkörben. A kínálat igen színes, a vásárlók érdeklődési körüknek megfelelően vásárolhattak például makettet, horgászbobot, szivart vagy akár függőágyat is. A legtöbb hazai e-bolt (14%) számítástechnikai eszközöket kínál, 13–13 százalék forgalmaz könyveket, illetve zenei és filmes kiadványokat. Nagyjából 5–5 százalékkal képviseltetik magukat a szórakoztató elektronikai, a szabadidős, az irodai cikkek, az élelmiszerek (szeszesital, édesség) és a ruházati termékek forgalmazói. A legnépesebb termékkategória a könyvek (768 ezer termék), második a zenei CD-k és filmek csoportja (485 ezer termék), míg harmadik helyen a számítástechnikai eszközök szerepelnek (48 ezer termék).

A kutatás időszaka alatt az üzletek 15 százaléka nyújtott állandó jelleggel árkedvezményt termékeire, amelynek jellemző mértéke hat százalék volt. Az állandó kedvezmény mellett időszakos árkedvezményt kínált a boltok közel egyharmada, átlagosan kilenc százalékos mértékben. Az e-boltokban a postai szállítást a leggyakoribb (78%), futárszolgálat (29%) és bolti kiszállítás (22%) ritkábban vehető igénybe. A termékről általában alapos tájékoztatást kaphatunk, három boltból kettő sűgővel könnyíti meg a vásárlást; a garanciális feltételekről és az adatvédelemről ugyanakkor csak a boltok 42 százalékában olvashatunk. A marketingcélú adatgyűjtés nem jellemző (7%), de a vásárlástól való elállás jogáról csak a boltok 29 százalékánál olvashatunk. Az e-boltok ügyfélszolgálatára 91 százalékban elérhető elektronikus levélen keresztül, és általában már néhány órán belül választ ad a feltett kérdésekre. Az üzletek 87 százaléka működtet telefonos ügyfélszolgálatot, ugyanakkor csak az e-boltok 54 százalékánál szerepel a cégnév, a postai cím, a telefonszám és az elektronikus levél cím mindegyike.

A teljes tanulmány szövege a http://www.recall.hu/download/karacsonyi_vasarlas_a_karosszekbol.pdf címen érhető el.

A következtetések levonása

A tanulmány bemutatását egy beszélgetés követte a jelenlévő hallgatóság buzgó részvételével, ennek során kiderült, hogy szinte minden nagyobb online kereskedő képviseltette magát. A vita kezdeteként a meghívott szakemberek elmondta el a véleményét a kutatás eredményei és személyes tapasztalatai alapján. Megállapították, hogy a magyarországi jogi szabályozás ezen a területen jó, bár vannak még megoldásra váró kérdések, hiányosságok. Egyetértés volt abban is, hogy az eredmény mind az ár, mind a termékkínálat, mind pedig a szolgáltatási színvonal területén mindenképpen kedvező. Az azonban nem túl jó, hogy csekély érdeklődés mutatkozik az online kereskedelem iránt a fogyasztók részéről. Magyarországon a felhasználók inkább csak tájékoztatói céllal használják az online boltokat, majd a kellő adatok megszerzése után elmennek, s az árut megveszik a boltban. Ez nem csak magyar sajátosság, s nem is a felhasználói bizalmatlanságban kell keresni a bizalmatlanság okát: az emberekkel inkább meg kell ismertetni, meg kell szeretetni ezt a fajta vásárlási formát, vonzó példát kell látniuk, s előfordulhat, hogy a szemléletváltás idővel megjön. A résztvevők szerint ehhez kormányzati segítségre, az iparág összefogására, s a tartalomszolgáltatókkal történő együttműködésre lenne szükség.

Szó esett többek között egy olyan ötletéről, hogy az online árusított termékek áfáját időszakosan engedje el a kormány, s azért, hogy a nem online kereskedők ne kerüljenek hátrányba, digitális társaiknak a kedvezményért cserébe ingyenes házhoz szállítást kellene biztosítaniuk.

Az IHM képviselője egyébként biztosított bennünket arról, hogy mind törvényi úton, mind pedig egyéb lehetőségek megragadásával segíteni kívánják az e-kereskedelem előrehaladását.

Kapcsolódó címek

Magyarországi Tartalomszolgáltatók Egyesülete
<http://www.mte.hu>
 RE:CALL Tanácsadó Kft. <http://www.recall.hu>
 Informatikai és Hírközlési Minisztérium
<http://www.ihm.gov.hu>



Komáromi Zoltán

(komi@kiskapu.hu)
 23 éves, a BME hallgatója, mellette PHP-programozóként dolgozik. Kedvenc területe a multimédia.



Az elektronikus kereskedelem helyzete Magyarországon



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

☛ www.linuxjournal.com honlapon olvashatók 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.

© Kiskapu Kft. Minden jog fenntartva

A hónap szakmai tanácsai

Csere-lemezzés javítása

Red Hat 9-et használok egy HP Brio gépen (Pentium III, 20 GB merevlemez és 192 MB RAM). Amikor először telepítettem a Linuxot, a merevlemez primary slave-ként működött; később ezen változtattam, ekkor primary master lett. Indítólemez használok a Linux-rendszer indítására, de amikor ezzel próbálkozom, hibaüzenetet kapok, hogy a csere-lemezzés (swap partition) nem került használatba. A mag fel tudja ismerni, hogy megváltozott a gyökér (root) és az indítás (boot) helye, de ez nem sikerült neki a csere-lemez esetében. Hogyan tudom dinamikusan megváltoztatni a csere-lemezzés helyét? Más szóval milyen értéket kell átadnom a magnak indítás előtt, hogy megtalálja a csere-lemezzést? Ezenkívül használhatom ugyanezt az indítólemezt más lemezzéskiosztással rendelkező rendszerek indítására a Windows indítólemezéhez hasonlóan? *Aman Hardikar*, cybergeek2k@rediffmail.com

Ezt egyszerűen tisztázni lehet. A Linux a csere-lemezzést a primary slave eszközön keresi, aminek a neve `/dev/hdbX`, az X pedig a lemezzés száma. Ehhez képest a csere-lemezzés most az primary master lemezen található, aminek a neve `/dev/hdaX`. Ebből következik, hogy a Linux nem fogja megtalálni, ezért nem is tudja használatba venni. A hiba elhárításához módosítani kell a `/etc/fstab` fájlt, hogy a cserelemezre vonatkozó bejegyzés a megfelelő eszközre mutasson. Ez valahogy így nézhet ki:

```
/dev/hdb2 swap swap defaults 0 0
Változtassuk meg erre:
```

```
/dev/hda2 swap swap defaults 0 0
```

A példában a csere-lemezzés a 2. sorszámú a lemezen. Újraindítva a rendszert már jól kell működnie. Újraindítás helyett kézzel is használatba lehet venni a csere-lemezzést a `swapon -a -e` paranccsal, természetesen miután a `/etc/fstab` fájlban már elvégeztük a fenti módosítást.

Felipe Barousse Boué, fbarousse@piensa.com

Serial ATA-támogatás?

SuSE 8.2 rendszert szeretnék használni, és arra lennék kíváncsi, vajon telepíthetem-e Serial ATA merevlemezre?

Daniel Gustafsson, gustafsson_danie@hotmail.com

Mint mindig, az első kérdés az, hogy a BIOS támogatja-e a rendszerindítást Serial ATA-csatolórról? Ha nem, akkor még lehet telepíteni, csak más vezérlőn lévő eszközről kell a rendszert indítani.

Jim Dennis, jimd@starshine.org

A SuSE alkatrész-adatbázisában (☛ <http://hardwaredb.suse.de>) utána lehet nézni, hogy az adott eszközt támogatja-e a SuSE. Más

Linux-terjesztéseknek szintén lehet hasonló együttműködési listájuk. Ám mégha támogat is valamit műszakilag a terjesztésben lévő rendszermag, az a legbiztosabb, ha a terjesztés eszköz-együttműködési listájában (hardware compatibility lists) utánanézzünk a dolognak, még mielőtt megvásárolnánk valamilyen alkatrészt. A listán lévő eszközöket a rendszer nagyobb valószínűséggel önműködően tudja telepíteni, és a terjesztés biztosabban tartalmaz olyan korszerű programeszközöket, amelyek támogatják a listán lévő alkatrészeket.

Don Marti, dmarti@ssc.com

Belső webhely elérése Squid segítségével

A feleségem számítógépe egy belső hálózathoz kapcsolódik, és az internetet proxyn keresztül éri el. Ez a proxy csak a www-t engedi át. Egy közös projekten dolgozunk, és szükség lenne arra, hogy olyan belső gépeket érjen el, amelyek egy másik proxyn keresztül látszanak csak. Ez a helyzet: feleségem gépe@proxy1@internet@proxy2@belső webhelyek. A proxy2 felett teljes hatalommal rendelkezem (Debian és Squid fut rajta). Hogyan lehet ezt megcsinálni?

Mauro A. Cremonini, mac@foodsci.unibo.it

A Squid proxy használata esetén a `cache_peer` és más beállítások lehetővé teszik, hogy a proxykiszolgálókat egymás után fűzzük. További tipp és trükkök a Squid GYK-ben olvashatók a ☛ <http://www.squid-cache.org/Doc/FAQ/FAQ.html> címen.

Chad Robinson, crobinson@rfgonline.com

Ha nem futtatsz webkiszolgálót a proxy2 80-as kapuján, akkor futtathatod az `rinetd`-t, hogy az továbbítsa a kapcsolatokat egy, a belső hálózaton elhelyezkedő rendszerre

(☛ <http://www.boutell.com/rinetd>). A biztonság szempontjából viszont az lenne a legjobb, ha megkérné a feleséged hálózati rendszergazdáját, hogy nyissa meg a tűzfalat a kimenő SSH-kapcsolatok számára. Ezután hozz létre egy felhasználói fiókot a feleségednek a proxy2-n, hogy alagutazni tudjon a következő paranccsal:

```
ssh -N -L 8080:internal:80 proxy2
```

Az `ssh` parancssori kapcsolóinak jelentése: `-N`, ne indítson héjat; `-L`, a helyi kapu továbbítása; `8080`, erre a kapura kell csatlakozni a gépen (localhost); `internal`, a gép neve, amelyekre a kapcsolatot át kell irányítani; `80`, erre a kapura kell továbbítani a kapcsolatot; végül a `proxy2` az a gép, amellyel az SSH-kapcsolatot kell létesíteni.

Ezután egyszerűen a `localhost:8080` címre kell beállítani a böngészőt.

Don Marti, dmarti@ssc.com

Linux Journal 2004. január, 117. szám

Új termékek

NemeSys DAW64

A RackSaver és az AMD együttműködésével készült el az új NemeSys 64 bites digitális audió munkaállomás (Digital Audio Workstation DAW64), ami az AMD 64 bites Opteron processzorára épül. A DAW64, amelyet elsősorban digitális hangfeldolgozásra, szerkesztésre és masteringre terveztek, egy 4U fiókos (rack mountable) kiépítésű rendszer duál Opteron processzorral, 4 GB memóriával, négy 36 GB-os Raptor SATA merevlemezzel, egy 3ware RAID vezérlővel, és egy nVidia Quadro 4 380XGL grafikus kártyával. A 64 bites névtér a felhasználók számára lehetővé teszi, hogy a virtuális hangszereket és a zenei könyvtárakat a fizikai memóriában közvetlenül elérjék, feleslegessé téve a tartalom több munkaállomásra való szétosztását.

➔ <http://www.racksaver.com>

Stratus/PMC

A Stratus/PCM kétfejes grafikus vezérlő PCM kártya, VME és CompactPCI beágyazott rendszerhez. A lelke egy 128 bites



grafikus modul videórögzítési (capture) lehetőséggel kiegészítve. A Stratus/PCM két megjelenítő csatornája két VGA vagy két LVDS kimenetet, illetve egy DVI kimenetet tud meghajtani. A Silicon Motion SM731 grafikus gyorsító egy kijelzős módban 1600×1200 képpontos 24 bit színmélységű, duál üzemmódban 1024×768 képpontos 24 bites színmélységű felbontásra képes. A kártyán található még egy 33/66 MHz-es, 32 bites PCI csatoló, 16 MB SDRAM, PAL/NTSC/SECAM normájú

kompozit és S-Videó bemenet, valamint programozható VGA BIOS EEPROM. A kártya VxWorks, Xfree86 és Microsoft Windows rendszereket támogat.

➔ <http://www.peritek.com>

Panasas

ActiveScale tárolótelep

Az ActiveScale tárolótelep (Storage Cluster), amely az ActiveScale File Systemből, StorageBlades és DirectorBlades tárolókból épül fel 4U méretben –



integrált gigabit ethernetkapcsolóval, objektumalapú felépítést használ, amelyben az adatokat nagy virtuális objektumokként kezeli. Az ActiveScale File System a fájlokat adatobjektumokká alakítja, majd az adattevékenységet dinamikusan elosztja a StorageBlades tárolók között. Ez a kialakítás párhuzamos adatutakat tesz lehetővé a kiszolgálók és a StorageBlades tárolók között, megszüntetve a szűk keresztmetszetet a teljesítményben és kapacitásban. Az alkalmazások és a felhasználói adatok a StorageBlades-ekben tárolódnak, a fájlrendszer aktivitását pedig a DirectorBlades irányítja. Ez utóbbi emellett virtuális adatobjektumokat készít a StorageBlades tárolók hatáskörében, így a rendszer egy névtér alatt látható.

➔ <http://www.panasas.com>

Aqua Data Studio 3.5

Az Aqua Data Studio 3.5 egy olyan adatbázis-lekérdező és -felügyelő eszköz, amely a fejlesztők számára SQL-parancsfájlok létrehozását, szerkesztését és végrehajtását, valamint az adatbázis-szerkezet böngészését és módosítását teszi lehetővé. A program összevont adatbázis-

környezetet nyújt önálló, egységes csatolófelülettel az összes fő relációs adatbázishoz. Ez biztosítja az adatbázis kezelőjének vagy a fejlesztőnek, hogy egy alkalmazásból egyszerre több feladatot is meg tudjon oldani. Az Aqua Data Studio támogatja többek között az Oracle 8i/9i, IBM DB2, Informix Dynamic Server, Sybase Adaptive Server, Microsoft SQL Server, MySQL és PostgreSQL adatbázis-felületeket. A 3.5-ös változat új tulajdonságai között található a teljes sémakibontás, az adatbázis-objektumok SQL-parancsfállal való ellátása és a kibővített Query Analyzer.

➔ <http://www.aquafold.com>

Xybernaut ServicePoint

A Xybernaut bejelentette a ServicePoint megjelenését, ami egy hordozható minikiszolgáló mobil alkalmazáskiszolgáló felületként beállítva és szállítva. A ServicePoint mobil vagy akár



viselhető minikiszolgálóként is működhet, ami az adat- és alkalmazáskiszolgálást közvetlenül juttatja el különböző eszközökhöz és felhasználókhöz, vezeték nélküli vagy vezetékes hálózaton. A ServicePoint eszköz egy Mobile Celeron Ultra Low Voltage Memory and Storage processzort, 256 MB SDRAM-ot, 5 vagy 10 GB belső merevlemezt, választható külső meghajtócsatlakozást, CompactFlash és USB kaput, 8 MB SDRAM videómemóriát és beépített, teljesen kétirányú (full duplex) sztereó hangkártyát tartalmaz, mindegyike Red Hat Linux Professional Servert telepítenek.

➔ <http://www.xybernaut.com>

Linux Journal 2004, 117. szám

Az év linuxos erőműve

Az AMD64 papíron lenyűgöző – de vajon valóban tudunk-e 64 bites processzorokkal minden eddiginél gyorsabb gépet építeni úgy, hogy az összes illesztőprogramunk is működjön?

Kipróbáltuk: összedobáltuk a legújabb processzort, a legújabb megjelenítő-, hangkeltő- és tárolóeszközöket, majd kíváncsian lestük, hogy mekkora füst száll fel a kupacból. Tavalyi linuxos erőművünk – *Don Marti* főszerkesztő döntése alapján – egy multimédiás munkaállomás lett. Először egy kétprocesszoros Xeon gépre gondoltam, a korábbi munkahelyemen ugyanis dolgoztam már ilyenekkel, csakhogy felhívtam *Trey Harris* a Monarch Computer Systemstől, hogy megérkeztek az első kétprocesszoros Opteron alaplapok. Nyeltem egy nagyot. Az egyetlen olyan hivatalosan is kiadott operációs rendszer, amely Opteronon fut, a Linux. Leveleket váltottunk, variáltunk, végül sikerült összedobnunk egy olyan gépet, amely írásom születésekor ugyan két hibával is küszködött, de a megjelenés időpontjára minden bizonnyal tökéletessé vált. Alaplapként egy Arima HDAMB típusú ATX-es munkaállomás-alaplapot választottunk AMD 8111/8151 Rhapsody lapkészlettel. Az alaplap négy DDR333 Registered ECC DIMM modul (mi két darab 1 GB-os Corsair CM74SD1024RLP-2700s modult használtunk), egy AGP 8x kártyát és öt darab 32 bites PCI-kártyát képes befogadni. Hátlapján négy USB-aljzatot, párhuzamos és soros kaput, az egér és a billentyűzet számára pedig PS/2-es csatlakozókat találunk, továbbá ide kerültek a Broadcom Gigabit ethernetcsatló és a Realtek ALC650 hangrendszer kivezetései is. A próbagépben nem volt FireWire csatló és SATA kapu sem, ám bár kiegészítő jelleggel mindkettő rákerülhetett volna az alaplapra. A két 940-es aljzatba AMD Opteron 246-os processzorok kerültek 2 GHz-es órajellel és Thermaltake AI724 hűtőkkel.

A megjelenítést egy nVidia Quadro FX 1000 kártya végezte – ez egy munkaállomásokba szánt, két monitor vezérlésére képes kártya, amelyre azért esett a választásom, mert az nVidia még 2002 decemberében bejelentette, hogy támogatja az AMD64 processzorokkal való használatát. (Az ATI a nyomdába adás időpontjában még nem nyújtott támogatást saját hasonló kategóriájú termékéhez, a Fire GL X1-hez.) A hangkeltést a Creative Labs Sound Blaster Audigy 2 kártyája végezte, amely 24 bites felbontásra és 192 kHz-es mintavételezésre képes, THX minősítéssel és 6.1-es analóg és Dolby Digital EX digitális kimenettel rendelkezik. A kártyán egy IEEE 1394 (FireWire) csatló is található.

Az IDE-csatolókra egy JLMS XJ-HD166S 16x DVD/CD-ROM-meghajtó és egy LITE-ON LTR-52327S 52/32/52x CD-RW-meghajtó került. DVD-író azért nem szereltünk a gépbe, mert a DVD+RW/DVD-RW/DVD-RAM formátumok közötti kavargás, illetve a

linuxos illesztőprogramok beszerzése még nincs tökéletesen megoldva. A Linux és a nyílt szabványok – mint az IDE – szépsége éppen abban rejlik, hogy mindenki kiválaszthatja a neki tetsző készüléket, majd a beszerelés után számíthat arra, hogy a szerzeménye működni is fog.

Ha az IDE-csatolók kapcsán nem említettem őket, akkor vajon mi újság a merevlemezekkel? Nos, az Opteronok mellett ezek lettek a gép leginkább figyelemreméltó részei. Nagyon érdekelt, vajon hogyan muzsikál a 3ware új Serial ATA RAID-kártyája, az Escalade 8500-4. Egy négycsatornás, félmagas, 64 bites PCI-kártyáról van szó, amely 32 bites foglalatban is gond nélkül használható. Mire ez a cikk megjelenik, az Escalade 8506s típus, a 8500s utóda is elérhető lesz. A 3ware termékadási útiterve szerint a második nemzedékbeli vezérlő 25 százalékkal lesz gyorsabb a 8500s típusnál. Én a 8500-assal tudtam dolgozni, a teljesítménymutatókat is ennek megfelelően kell kezelni.

A 8500-4 kártyán négy SATA-150 kapu van, ezek az áramkör hátulso élére kerültek, vagyis a meghajtók felé néznek. A kiváló elrendezésnek és annak köszönhetően, hogy a SATA-kábelek jóval vékonyabbak és rövidebbek elődeiknél, a meghajtók lényegesen jobban hűthetők. Erre szükség is volt: a négy 36 GB-os Western Digital Raptor WD360GD Serial ATA egység ugyanis tízezres fordulatszámra pörgött. Nem véletlen, hogy a nagyobb kapacitású, de kisebb fordulatszámú meghajtók helyett ezeket választottuk: nem nagy tárterületre, hanem jókora sebességre vágytunk. A meghajtók – a szabványok közötti átállást segitendő – normál Molex tápcsatlakozóval és újfajta SATA tápcsatlakozóval is rendelkeznek. Az egyik merevlemez a hajlékonylemezes meghajtó alá került, a többi



három függőlegesen a ház aljába, a hűtőventilátorok elé. Ha már itt tartunk, ejtsünk néhány szót a házról is. Az összes alkatrész – egy ENERMAX 465P-VE-24P típusú, 460 wattos tápegység társaságában – egy egyedi, csendes Lian-Li PC-6270 házba került. A házhoz Lian-Li módra mindenféle tetszetős kiegészítő jár, például szinte minden részét csavarhúzó nélkül is tudjuk szerelni. Előoldalán egy levehető ajtó mögött négy 5,25-ös és három 3,5-ös meghajtóhely található, belül további egy vízszintes és öt függőleges meghajtóhely áll rendelkezésünkre. A házban az alaplap rögzítésére szolgáló lapja kihúzható, valamint rendelkezik egy szűrővel ellátott légbemlővel is, a merevlemezek hűtő két ventilátor itt szívja be a levegőt. A meleg



levegő eltávolítását két másik ventilátor végzi a ház hátoldalán. Az alsó ötmeghajtós, függőleges meghajtóhely egy átalakítóval vízszintes, négymeghajtós helyé változtatható, egy apró, ragasztóval rögzített kapocs pedig a SATA-kábeleket tartja távol a levegő útjától. A ház elején két USB-kapukapcsoló található, ezek akkor is hozzáférhetők, ha az ajtó be van zárva. Köszönhetően a ház felső és oldalsó lemezeire helyezett vastag szivacsrétegeknek, illetve az ajtó köré illesztett gumitömítésnek, a gép viszonylag csendesen üzemelt.

A grafikus illesztőprogramok és a nyílt forrás

A 3dfx felvásárlása óta minden nagyobb grafikai cég – ATI, nVidia és Matrox – zárt forrásúvá tette 3D-gyorsítást is kínáló illesztőprogramját. Hozzáállításuk több szempontból is – úgymond – kellemetlen. Először is, a kérdéses cégtől függ, hogy mikorra fejleszti ki az illesztőprogramokat, illetve mikor javítja ki a hibáikat. Ha úgy döntenek, hogy az univerzum összes operációs rendszeréhez előbb adnak ki illesztőprogramot, minket pedig két évig váratnak, esetleg tudomást sem vesznek a Linuxról, akkor így döntöttek, nem tudunk mit tenni. Mivel manapság nem egyszerű belépni a grafikus piacra, nem hiszem, hogy túl sokat tehetnének ez ellen. Mégha ezek a cégek úgy is döntenek, hogy támogatnak minket, rengeteg dologhoz, például a próbakódokhoz továbbra sem lesz hozzáférésünk. Ellenpélda az Audigy 2, aminek csak lerántottam a kódját CVS-ből, és már működött is.

A programozásért rajongók hozzátehetik, hogy egy kód kiadásával nemcsak az azt próbálgatók száma nő meg nagyságrendekkel, de tanulás, szórakozás céljából is bele lehet nézni, ahogy azt sokan szívesen megtennék az újabb grafikus illesztőprogramokkal. Végül rámutatnék, hogy azért sincs sok értelme titokban tartani a kódot, mert a kártya nélkül úgysem használható semmire. Sajnos nem ismerem az illesztőprogramok zártan tartásának hivatalos okait – nem volt időm, hogy utánuk nyomozzak; de szívesen vitáznék róluk nyíltan. Úgy vélem, a Linux-közösség és a grafikai termékek fejlesztői számára egyaránt előnyösebb volna a nyílt, barátságos viszony, amely mindannyiunkat hozzásegítene ahhoz, hogy a legjobb illesztőprogramokkal, a legjobb grafikus kártyákkal dolgozhassunk a legjobb operációs rendszer alatt. Meggyőződésem, hogy az a cég, amelyik elsőként nyit ebbe az irányba, gazdaságilag is előnyt szerez, a Linuxot használók ugyanis eszközvásárlásaik során is egyre inkább ragaszkodnak elveikhez.

© Kiskapu Kft. Minden jog fenntartva

nVidia: zártan, de nem elzárkózva

Jeff Brown az nVidiától pont akkor csörgött rám, amikor írásom utolsó betűit pötyögtem be. Még frissek voltak az illesztőprogramjukkval szerzett élményeim. Gyorsan lecsaptam, és megkérdeztem tőle, az nVidia miért nem teszi nyílt forrásúvá az illesztőprogramjait, hiszen így én is gyorsabban megoldhattam volna a felmerült gondokat. Azt válaszolta, hogy az illesztőprogram – amelynek magja a Windowstól kezdve a Linuxon keresztül egészen FreeBSD-ig és az OS X-ig azonos – körülbelül a felét képviseli az nVidia szellemi tulajdonának, míg a másik fél maga a grafikai processzor. Összehasonlításképpen: a hálózati kártyáknál az Intel szellemi tulajdona túlnyomórészt magán a kártyán található, így az illesztőprogramot nyugodt szívvel helyezhették a GPL szerződés hatálya alá. Az nVidia – mintegy kárpótlásul – széles körű támogatást és visszajelzési lehetőséget nyújt. Brown elmondása szerint például *Andy Mecham*, az nVidia egyik mérnöke a munkaidejének felét azzal tölti, hogy az nnews.com linuxos fórumán segítséget nyújt a felhasználóknak. Ezt magam is megerősíthetem: miközben a fórumon

keresgéltem, feltűnően gyakran találkoztam Andy nevével. Brown hozzátette, a linuxos oldal nagy felhasználói, például a Visual Effects Society tagjai – akik közül egynek sikerült megjegyeznie a nevét, és ez a Disney volt – elégedettek az nVidia által nyújtott támogatással. Ezek a vállalatok kereskedelmi szintű támogatással ellátott gépeket akarnak használni, hasonlókat az erőműhöz, természetesen nem hétköznapi feladatokra.

Az, hogy a Monarch segítségével még én mint egyszerű felhasználó is felvehettem a kapcsolatot egy olyan mérnökkel, aki anélkül is eredményesen oldotta meg gondomat, hogy hozzáfért volna a gépemhez, önmagában is bizonyítja, hogy az nVidia magas szintű támogatást ad. Ugyan nem hiszem, hogy valaha is GPL alatti illesztőprogramot fogunk kapni bármelyik nVidia kártyához is, úgy vélem, megkapjuk a második legjobb dolgot, amit remélhetünk: egy, a jövőt tisztán látó céget, amely – ha nem is adja ki titkos receptjét – mégis elkötelezetten segít bennünket.

A Tiobench kimenete (rövidítve) – soros olvasás

Azonosító	Fájl-méret	Blokk-méret	Mennyiség szála	Átlagos átvitel	(CPU%)	Késleltetés
Athlon	1792	4096	1	81,60	29,29%	0,048
Opteron	4096	4096	1	59,30	14,86%	0,066
Athlon	1792	4096	2	58,07	30,61%	0,132
Opteron	4096	4096	2	62,18	13,46%	0,125
Athlon	1792	4096	4	54,37	61,00%	0,285
Opteron	4096	4096	4	59,19	14,59%	0,260
Athlon	1792	4096	8	55,29	64,72%	0,542
Opteron	4096	4096	8	48,44	13,17%	0,625

A Tiobench kimenete (rövidítve) – véletlenszerű olvasás

Azonosító	Méret	Fájl-méret	Blokk-szála	Mennyiség átvitel	(CPU%)	Késleltetés
Athlon	1792	4096	1	1,32	0,975%	2,952
Opteron	4096	4096	1	1,18	0,151%	3,296
Athlon	1792	4096	2	2,29	1,374%	3,354
Opteron	4096	4096	2	1,93	0,740%	4,017
Athlon	1792	4096	4	3,18	2,859%	4,639
Opteron	4096	4096	4	2,76	1,236%	5,431
Athlon	1792	4096	8	3,70	2,221%	7,264
Opteron	4096	4096	8	2,96	2,085%	9,860

A Tiobench kimenete (rövidítve) – soros írás

Azonosító	Fájl-méret	Blokk-méret	Mennyiség szála	Átlagos átvitel	(CPU%)	Késleltetés
Athlon	1792	4096	1	20,65	11,17%	0,126
Opteron	4096	4096	1	28,15	10,78%	0,101
Athlon	1792	4096	2	22,15	26,81%	0,228
Opteron	4096	4096	2	22,69	14,23%	0,292
Athlon	1792	4096	4	22,97	29,67%	0,472
Opteron	4096	4096	4	20,04	14,44%	0,714
Athlon	1792	4096	8	21,87	27,93%	0,856
Opteron	4096	4096	8	13,42	11,03%	1,978

A gépet most már kiveséztük, foglalkozunk egy kicsit az operációs rendszerrel is. A SuSE Linux Enterprise Server 8 (SLES 8) mellett döntöttem, ugyanis nagyon tetszettek a SuSE 32 bites rendszerei. Igaz, hogy a SLES 8 nem egy csillogó-villogó, mindenkit elbűvölő rendszer, hibátlan működése és könnyű használhatósága mégis meggyőző volt. Alapszintű Gnome- és KDE-környezetet tartalmaz, valamint rengeteg fejlesztőeszközt, így kiváló alapként szolgál profi alkalmazások fejlesztéséhez és használatához. Ha mindez nem lett volna elég, a YaST2, a SuSE grafikus felületű telepítője engem végleg meggyőzött. A YaST2 a

nyomtató beállításától kezdve a felhasználók hozzáadásán keresztül a tűzfal finomhangolásáig mindenre jó, a szükséges teendőket gyorsan, letisztult eljárásokkal, különösebb meglepetések nélkül végezhetjük el. Az egyetlen gondom az volt, hogy nem ALSA hangkártyát képtelen voltam vele üzembe helyezni, de erről majd később.

Számos terjesztésnél felmerül, hogyha nem x86 alapú gépre végezzük a telepítést, akkor a csomagok régiek, és a legkülönbözőbb dolgok működésében tapasztalunk rendellenességeket. A SLES 8 2.4.19-es rendszermagot, KDE 3.0-t, Samba 2.2-t és XFree86 4.2-t tartalmaz. Nem éppen a legújabb, de megteszi. Nem kevésbé fontos, hogy az alkotóelemek jól együttműködnek, és – a hangtól eltekintve – semmi olyan hibát nem találtam, amit ne tudtam volna pillanatok alatt elhárítani.

Mint már utaltam rá, a hangkártya beüzemelése érdekes kaland volt. Az Audigy 2 viszonylag új termék, és már előre sejtettem, hogy komolyan meg kell küzdenem vele. Nem csalódtam: a YaST2 ALSA-beállítója ugyan szépen felismerte a kártyát, hang azonban nem volt. Megpróbálkoztam a SourceForge-on található emu10k1 Project illesztőprogramjával is.

A `/etc/modules.conf` fájlt kézzel átírva sikerült elérnem a legújabb OSS illesztőprogram betöltését, csakhogy a gép továbbra is némaságba burkolózott. Mielőtt végleg feladtam volna és jelentettem volna a hibát, rákerestem a SourceForge hibaadatbázisában. Megtudtam, hogy a CVS-ben található változat másnál működik. De vajon 64 bites gépen is így tesz majd? A `make` parancsfájlokkal folytatott küzdelem után hamarosan kiderült, hogy a CVS-ben talált változat ütközik az alaplap Realtek lapkával, ezért ezt az utóbbi bejegyzését töröltem a `modules.conf` fájlból, majd újraindítottam a gépet. Kicsivel később, amikor a hangszórókból felzendült Theodorakis „Óda Zeuszhoz” című műve, a győzelem jóleső érzésével dőlhettem hátra.

Ugyan magas szintű műszaki próbákra sem időm, sem berendezéseim nem voltak, zeneileg viszonylag kiművelt személynek tartva magam úgy gondoltam, kipróbálom, mit tud az Audigy 2 – vajon megéri-e az árát? Feleségem gépe és az enyém egymás mellett állnak, ugyanolyan

hangszórókkal felszerelve. Kihúztam a gépéből a hangszóró kábelét, átdugtam az újonnan épített erőműbe, majd CD-ről elindítottam Arnaud „Bulger’s Dream” című művét. Mindkét próbánál XMMS-t használtam, kikapcsolt hangszínszabályzóval. Nos, be kell ismernem, hallás alapján az Audigy 2 messze tisztábban és szélesebb tartományban szóló hangot ad ki magából, mint Esoniq 5880-as kártyám. A zene mintha étellel telibb lett volna, és erősebb hangerőn sem kerültek bele torzítások. Biztosan léteznek olyan kártyák, amelyek még az Audigy 2-nél is szebb hangot keltenek életre, de egy átlagos földi halandónak ez is kiválóan megfelel.

A Tiobench kimenete (rövidítve) – véletlenszerű írás

Azonosító	Fájl-méret	Blokk-méret	Mennyiség szál	Átlagos átvitel	(CPU%)	Késleltetés
Athlon	1792	4096	1	0,60	0,234%	0,014
Opteron	4096	4096	1	0,47	0,121%	0,009
Athlon	1792	4096	2	0,59	0,479%	0,028
Opteron	4096	4096	2	0,50	0,159%	0,011
Athlon	1792	4096	4	0,64	0,542%	0,029
Opteron	4096	4096	4	0,49	0,155%	0,012
Athlon	1792	4096	8	0,68	0,558%	0,036
Opteron	4096	4096	8	0,50	0,192%	0,013

És a lényegét se felejtsem ki: 64 bites gépben is működik. A gép összeállítása során egy másik nehéz kérdéssel is szembesültem, de a nyomdába adás előtt ezt is sikerrel hártottam el: ez pedig a videokártya hibája volt. Aki 2003 augusztusában olvasta a témában a weben megjelent cikket (☛ <http://www.linuxjournal.com/article/6922>), az talán emlékszik rá, hogy az ATI Fire GL X1 kártyája azért maradt ki a gépből, mert csak 32 bites rendszerekben volt használható. Az ATI részéről semmilyen tájékoztatást nem sikerült szerezni a 64 bites rendszerek támogatásával kapcsolatban. Igaz, az ATI képviselője – miután elolvasta írásomat – legalább megkeresett, és megkérdezte tőlem, hogy milyen kártyát használtam és miért cseréltem le. Amikor először próbálkoztam meg az nVidia Quadro FX 1000 kártya illesztőprogramjának futtatásával, az X-kiszolgáló száz százalékban leterhelte a processzort, és gyakorlatilag használhatatlanná vált. Tette mindezt egy már hat hónapja kiadott illesztőprogrammal. A Monarch segítségével közvetlen kapcsolatba léptem az nVidiával; **Mark Visconti**, az nVidia mérnöke átnézte a gépet, azt javasolta, hogy frissítem a BIOS-t, mert amivel dolgozni próbáltam, egy korábbi próbaváltozatnak tűnt. Engedelmesen letöltöttem a BIOS és a memória programozására szolgáló alkalmazást – nem működött. Szerencsére még korábbról volt egy kapcsolat az Arimánál, a kártya gyártójánál. Kiderült, hogy a **PHLASH.EXE**-nek valamiféle rejtélyes kapcsolókat kell megadni, csak így lehet rávenni a programozásra. Természetesen mindeközben a jó öreg DOS-t sem lehet megkerülni. Ezt követően a Visconti által javasolt BIOS-beállításokat adtam meg. Fáradozásom nem volt haszontalan, a Linux elindítása után a startx parancs hatására felbukkant az X áhított képernyője. Azt sikerült ellenőriznem, hogy a kiszolgáló valóban 3D-módban futott, de teljesítméymérésre már nem maradt időm. Figyelembe véve az nVidia kártyájával a 32 bites gépen végzett mérések eredményét, 50–60 képkocka/másodperc eredményt kaptunk volna, ha nem jobbat. Ha sikerül összehozni a mérést, az eredményeket weboldalunkon közzé fogjuk tenni. A gép igazi tudása azonban még meg sem mutatkozott. A Chromium videoteszt mellett lefuttattuk a Linux Journal Benchmark Bundle csomagot is (bonnie++ , postgres-test, tiobench és rendszermag-fordítás), és az eredmények

lenyűgözőek voltak. Az alaprendszermag lefordítása mindkét processzor használatával mindössze 1 perc és 35 másodpercig tartott. Csak összehasonlításképpen: a saját 1,1 GHz órajelű Duron processzorom ugyanezzel a feladattal 6 perc 50 másodperc alatt végez. A nyers gigahertzekeket tekintve is azonnal megállapíthatjuk, hogy az Opteron és 3ware alapú gép 15 százalékkal nagyobb teljesítményt kínál. Az Opteron processzor beépített memóriavezérlővel bír, amely az északi híd közreműködése nélkül képes 64 bites, teljes kétirányú csatornát biztosítani a memória kezelésére. A nálam járt gépet – 32 bites módban – egyetlen DIMM modul is lehetett használni, ami ugyan sebességsökkenéshez vezet, ám hibakereséskor hasznos lehet.

A Tiobench ugyancsak érdekes eredményeket adott, ezek az első táblázatban láthatók. Összehasonlítottam, hogy a 3ware–Western Digital párosítás mit tud Opteron és Athlon 2800 processzorral. Bár a számokat nem lehet egyben összehasonlítani, az Opteron alapú gépen ugyanis kihasználtam a 64 bites, nagyobb fájlkezelés is lehetővé tévő címzési lehetőséget, ám a kapott értékek önmagukért beszélnek. A véletlenszerű írásnál és olvasásnál az erőmű láthatóan lemaradt, de a késleltetésekkel újra csökkent a hátránya. Az erőmű a több mint kétszer nagyobb méretű fájl ellenére is messze gyorsabb volt az Athlonnál. Szerintem a soros olvasás és írás végén látható teljesítménycsökkenés a 3ware kártya átmeneti tárának kimerüléséből fakad, hiszen eddig a pontig szép egyenes a görbe. Amikor áttekintettük a a weben megjelent cikkemhez érkezett hozzászólásokat linuxos erőműről, úgy döntöttünk, hogy a gép zajszintjét is megmérjük. A ház előlapjától 25 cm-re 50,5 dBA, a felhasználó helyén (60 cm-re a ház tetejétől) 50,0 dBA, a ház hátlapjától 25 cm-re pedig 60,0 dBA zajt mértünk. Nem tagadom, ezek elég nagy számok, például a már említett Dell gépnél rendre 47, 45 és 55 dBA volt a kapott érték. Valószínűleg a Thermaltake hűtők voltak a bűnösök, mert amikor AMD hűtőkkel próbáltuk a gépet, jóval csendesebben üzemelt – igaz, jobban melegedett is. Az erősebb hűtőkkel – hőmérsékletét tekintve – üzembiztosabb volt az erőmű, ezért az előnyért a magasabb zajszinttel kellett fizetni.

Köszönetnyilvánítás

A szerző szeretne köszönetet mondani a 3ware, a Western Digital, az nVidia, az Arima és a SuSE cégeknek támogatásukért, továbbá Monarch Computer Systemsnek, amiért segítettek összeállítani a gépet.

Linux Journal 2003. december, 116. szám



Glenn Stone

Rendszergazda, újságíró, Red Hat Certified Engineer és Linux-rajongó. 1999 óta épít számítógépeket szórakozásból és a munkájából fakadóan. Boldogan tölti napjait az Egyesült Államok nyugati partvidékén.



DVD-lejátszók

A négy legjobb linuxos DVD-lejátszóalkalmazás tudását, használatának kényelmét és teljesítményét hasonlítjuk össze, majd néhány tanácsot is adunk a zökkenőmentes lejátszás érdekében.

Szeretem a filmeket. Szinte minden filmet megnézek, műfajtól, származási országtól és nyelvtől függetlenül. Múlt karácsonyra Ivy és én kaptunk a gyermekeiktől egy DVD-lejátszót, és azóta elkötelezett híve lettem a formátumnak. A kép élesebb, a hang tisztább, és maga a formátum is fontos kényelmi szolgáltatásokat nyújt, mint például a feliratozás, a jelenetek vagy a nyelvek kiválasztása. Az egyetlen baj az, hogy a készülék az ő házában van, és nem az enyémben. Nekem még tévém sincs, tehát ha akarnék egy DVD-lejátszót, akkor tévét is kellene vennem, erre pedig nem szeretném rászánni magam. Van viszont egy szép 19"-os monitorom, amit egy remek videokártya vezérel. Magától értetődő volt tehát, hogy beszerezek egy DVD-meghajtót a gépembe, és DVD-lejátszásra alkalmassá teszem a rendszert. Ebben a cikkben elmesélem, hogyan jutottam túl mindezen, illetve milyen tapasztalatokat szereztem a különféle alkalmazásokkal.

A számítógép

Miközben a gépemet megpróbáltam DVD-lejátszásra alkalmassá tenni, rengeteg ismeretet szereztem a vas tekintetében. Gépem most alkalmas a folyamatos filmlejátszásra, de a legnagyobb teljesítményt csak némi „matatás” árán tudtam kihozni belőle. A továbbiakban egyrészt szót ejtek a kiépítéséről, másrészt egy korszerűbb gép összeállítására is megpróbálok javaslatot tenni. A vas ezen a területen elég sokat számít, ezért első lépésben mindenkinek meg kell róla győződnie, hogy a gépe elég erős-e ahhoz, hogy elfogadható minőségben játssza le a DVD-lemezeket.

A valóság

DVD-meghajtóm egy olcsó darab, egy barátomtól vettem, akinél feleslegesen hanyódott. A `dmesg` a következő adatokat adta róla:

```
hdd: LITEON DVD-ROM LTD163D,
↳ATAPI CD/DVD-ROM drive
```

Ha valaki meg szeretné tudni, hogy milyen meghajtó van a gépében, akkor a `dmesg | grep DVD` parancsot kell kiadnia. A Linux remek DVD-támogatással bír, a „Linux Hardware Compatibility HOWTO” állítása szerint gyakor-

latilag minden ATAPI és SCSI felületű DVD-ROM és DVD-RW egység támogatása megoldott. Ha nem tudjuk, hogy melyik márkát válasszuk vagy valamelyik típussal kapcsolatban bővebb tájékoztatásra vágyunk, a Google segítségével biztosan megtalálunk minden adatot. A gépben egy 800 MHz-es AMD Duron processzor, két 15 GB-os merevlemez és 512 MB RAM volt. A monitor egy teljesen átlagos 19"-os monitor, amelyre a képet egy 64 MB memóriával ellátott nVidia GeForce2 videokártya rajzolta. A hangkeltésről egy Sound Blaster Live! Value kártya gondoskodott, ehhez egy digitális Yamaha DMP7 keverő, egy 100 wattos QSC erősítő és egy pár Yorkville Sound YS-10 stúdióhangfal csatlakozott. A rendszerem úgy volt lefordítva, hogy kis késleltetéseket biztosítson, és a merevlemezek is a legnagyobb átviteli sebességre voltak állítva.

Ami jól jön

Érdemes ennél gyorsabb, legalább 1 GHz órajelű processzort beszerezni. Ugyan hallottam olyat, hogy kifogástalanul ment a DVD-lejátszás 600 MHz-es, sőt lassabb gépeken is, ám az 1 GHz-nél kisebb órajelű gépeknél gondosabb finomhangolás szükséges. Ha a filmeket széles vásznú módban szeretnénk látni, akkor legalább 19"-os monitorra lesz szükségünk. A grafikus lapkakészletnek támogatnia kell az XFree86 Xv bővítményt, a videokártyának pedig legalább 16 MB memóriával kell rendelkeznie. Sztereó vagy 5.1-es hang megszólaltatásához a Creative Labs Sound Blaster Live! vagy Audigy 2 hangkártyája kiválóan megfelel.

Még egy gondolat a megfelelő hangrendszeréről. Sok DVD 5.1-es térhangzást és egyéb hangzásbeli érdekességeket kínál, ám ezek Linux alatti támogatása nem mindig megoldott. Egy olcsóbb 5.1-es hangszórókészlet alkalmi használatra kiváló, de ha komolyabb hangminőségre vágyunk, akkor hangszórókból is komolyabbakat kell vennünk. Írásomnak a Linux Journal webhelyén megjelent változata (☞ <http://www.linuxjournal.com/article/7174.html>) források (Resources) részében szerepel egy hivatkozás a Tom's Hardware Guide oldalra, ahol bővebb tájékoztatást is találni a 4.1-es és 5.1-es, személyi számítógépekhez szánt hangrendszerekről.

Beállítási kérdések

Próbálkozásaimhoz egy erősen átalakított 7.2-es Red Hat-terjesztést használtam. Az itt leírt módszerek és eljárások – esetleg kisebb módosításokkal – valószínűleg bármelyik elterjedtebb terjesztés alatt használhatók. Egy alapbeállításokkal telepített Linux nem feltétlenül nyújt kifogástalan DVD-s teljesítményt, ezért az alábbiakban megemlítenék néhány a webről és a különféle lejátszók leírásából kima-zsolázott tanácsot. Elsősorban négy dologra kell figyelni: a rendszermag, az X és a DVD-meghajtó finomhangolására, illetve a meghajtó befűzési pontjának a helyességére.

A rendszermag

DVD-lejátszás terén a legjobb eredményt csak a rendszermag néhány beállításának módosításával érhetjük el. Erősen javaslom *Andrew Morton* feldolgozási szüneteket rövidítő foltjainak a használatát, illetve *Robert Love* időosztásos (preemptive) foltját is érdemes feltenni. A két folt telepítését követően rendkívül kicsi, 3 msec alatti átlagos késleltetésekkel fog dolgozni a rendszer. Nálam jelenleg a 2.4.18-as rendszermag fut, de Andrew és Robert foltjai nagyon sok változathoz elérhető (lásd a *Kapcsolódó címeket*). Ugyancsak érdemes ellenőrizni, hogy a rendszermagban engedélyezve van-e az RTC és az MTRR támogatása; ezek a *Character devices* és a *Processor type and features* beállítás csoportban található. Az RTC a számítógép valós idejű órájához biztosít hozzáférést. A rendszermag beállítási leírásából kiderül, hogy minden PC rendelkezik ilyen órával, csak a használata az alapértelmezett rendszermag-beállításokban nincs engedélyezve. Igaz, nincs feltétlenül szükség rá, de a pontosabb időzítések miatt sok linuxos hang- és mozgókép-alkalmazás – például az MPlayer – használja ezt az órát, ezért az elérését biztosító programrészt érdemes vagy belefordítani a rendszermagba, vagy legalább dinamikusan betölthető modulként elérhetővé tenni. A leírásból tudtam meg azt is, hogy az MTRR (memory type range registers, memóriatípus-tartományregiszterek) engedélyezésével elérhető alrendszer a processzor részéről lehetővé teszi az egyes memóriatartományokhoz való hozzáférés szabályozását. Az írásegyesítés (write-combining) engedélyezése megengedi, hogy a PCI vagy AGP buszon továbbítandó írási utasítások elküldés előtt egyetlen nagyobb blokkba egyesüljenek. Ezzel a képadatok írásának sebessége a 2,5-szeresére vagy még nagyobb mértékben növekedhet. Az „ATA/IDE/MFM/RLL Support” részben javaslom az általános PCI bus-master DMA-támogatásának, illetve a PCI DMA alapértelmezett használatának engedélyezését. Az utolsó lépés a gép hangkártyájának beállítása. Mivel én az ALSA hangrendszert használom, egyszerűen csak engedélyeznem kellett a hangkártya támogatását a rendszermag beállításai között. Az ALSA-illesztőprogramok felhasználói területen futnak betölthető rendszermagmodulként és a rendszergazda telepítheti őket. Az ALSA helyett más



modulba fordított illesztőprogramot is lehet használni, de én úgy látom, hogy az ALSA illesztőprogramok a legjobbak. Sőt a 2.5.x sorozat után lényegében az ALSA lesz a Linux-rendszermagok hangrendszere.

Az X tényező

A próbákat az XFree86 4.1.0-s változatával kezdtem. Nagyjából minden működni látszott, de amikor bezártam a lejátszót, az X-kiszolgáló fogta magát és összeomlott, én pedig a parancssorban találtam magam. Miután frissítettem az XFree86 4.3.0-s változatra és telepítettem nVidia GeForce2 kártyám legújabb illesztőprogramját (1.0-4363), a hiba megszűnt. Mivel az XFree86 4.3.0-s egy csomó, a korábbi változatokban talált hiba javítását tartalmazza, mindenkinek azt javaslom, hogy telepítse a gépére, függetlenül attól, hogy melyik lejátszóprogramot választja. Aki pedig nVidia kártyát használ, próbálja meg a legújabb illesztőprogramot feltenni. Mint korábban utaltam rá, az XFree86 rendelkezik egy videokimeneti illesztőprogrammal, ez az xv; a képkocka-átmenetítár (frame buffer) eszközhöz azonban más illesztőprogramok is léteznek, mint például az SDL (Simple DirectMedia Layer) és az X11/Xshm. A lejátszás teljesítménye a kiválasztott illesztőprogramtól is függ. Az xv általában tökéletesen megfelel, de a leginkább tetsző lejátszóprogram kiválasztása után mindenki nyugodtan kísérletezzék ki, hogy melyik illesztőprogrammal tudja a legszebb kimenetet elérni. Jómagam a Blackbox ablakkezelőt használom. Szeretem, ha valami egyszerű és gyors, márpedig a Blackbox pontosan ilyen. Tisztában kell lennünk azzal, hogy a DVD-lejátszás minőségét az ablakkezelő és a munkakörnyezet is befolyásolja – ha magasak az igényeink, esetleg nagyobb teljesítményű eszközökre lehet szükségünk.

Pörgessük fel a lemezt!

Sokaknak valószínűleg újdonság, hogy egy DVD- vagy CD-ROM-meghajtót ugyanúgy hangolni lehet, ahogyan egy merevlemez. A hdparm segédprogram képes a legnagyobb hatékonyságúra állítani a meghajtót. Ehhez rendszergazdaként a következő kapcsolókkal kell futtatni:

```
hdparm -c1 -d1 -a8 -u1 /dev/hdd
```

A parancsban a -c1 a 32 bites be- és kivitelt, a -d1 pedig a DMA-használatot engedélyezi, a -a8 a fájlrendszer előleolvasási értékét adja meg, a -u1 pedig a meghajtó megszakításkimaskolási jelzőjét állítja be. A /dev/hdd kapcsolónak a megfelelő meghajtóra kell mutatnia. A fenti átadott értékek az én meghajtómnál megfeleltek, azt azonban senkinek nem javaslom, hogy vakon, a hdparm

A linuxos DVD-lejátszóprogramok összehasonlítása

Lejátszó	Változat	Felirat	Menü-támogatás	Véletlenszerű léptetés	Vezérlés billentyűzetről	Grafikus felület*	Processzor-terhelés**	Felhasználói szerződés
MPlayer	0.90	Igen	Nem	Igen	Igen	Elhagyható	40%–50%	GPL
Ogle	0.9.1	Igen	Igen	Igen	Igen	Elhagyható	20%–40%	GPL
VideoLAN Client	0.5.3	Igen	Igen	Igen	Igen	Alapértelmezett	20%–30%	GPL
xine	1-beta12 (lib) 0.9.21 (ui)	Igen	Igen	Igen	Igen	Alapértelmezett	20%–40%	GPL

*Azt adja meg, hogy a grafikus felület az alapértelmezett módon lefordított alkalmazásnak a része-e vagy sem.

A megadott értékek a gkre11m segítségével a **Származás fejeadás című DVD lejátszása közben mért alsó és felső terhelési határértékek. A rendszer terheléséhez hozzájárult az XMMS, öt munkaasztal futása (Blackbox ablakkezelő alatt) és egy aktív DSL-kapcsolat, amelyet Netscape vagy Opera böngészővel használtam.

leírásának átnézése nélkül alkalmazza őket. A DVD-meghajtó csak olvasni tudja a lemezt, így a fájlrendszer megsérülése itt természetesen nem fordulhat elő, az viszont igen, hogy helytelen beállításokkal lerontjuk a meghajtó teljesítményét. Felfigyeltem arra is, hogy a Red Hat 8.0 terjesztést használók nehézségeket tapasztaltak a DVD-meghajtó DMA-elérésének engedélyezésekor. Aki ezt a terjesztést használja, az alábbi sorral bővítsa a `/etc/modules.conf` fájlt:

```
options ide-cd dma=1
```

A /dev/dvd

A későbbiekben említett alkalmazások kivétel nélkül a `/dev/dvd` befűzési pont alatt keresik a meghajtót. Ugyan mindegyik programnak meg lehet adni egyedi elérési helyet is, könnyebb az életünk, ha létrehozunk a `/dev/dvd` befűzési pontot. A meghajtó általában a `/dev/cdrom` úton érhető el, tehát egy, a `/dev/cdrom` könyvtárra mutató hivatkozást kell létrehozunk `/dev/dvd` névvel. Ehhez rendszergazdaként az alábbi parancsot kell kiadnunk:

```
ln -sf /dev/cdrom /dev/dvd
```

Ha több CD/DVD-meghajtóval rendelkezünk, akkor a megfelelő sorszámú meghajtóra kell hivatkoznunk – nálam ez a `/dev/cdrom1` volt, a `/dev/cdrom` ugyanis a CD-íróm.

A próbalemezek

A programokat különféle lemezekkel próbáltam ki, természetesen mindegyik jogtiszt, boltban vásárolt példány volt. A közeli könyvtárból DVD-lemezeket is lehet kölcsönözni, ezek jelentős része nem a legjobb állapotban van, mégis gond nélkül sikerült lejátszani őket, kivéve Bruce Lee „Kínai kapcsolat” című filmjét, ami rettenetesen le volt amortizálva. Még egy olyan ki tudja honnan származó, pocsék állapotú lemezt is sikerült lejátsznom, amit különálló DVD-lejátszónk még csak fel sem ismert. Örömmel

jelenthetem ki, hogy az összes feliratokkal, nyelvvalasztással, fejezetvalasztással és bőrökkel (skin) kapcsolatos szolgáltatás kifogástalanul működött.

Linuxos DVD-lejátszók

Belepiszkáltunk a rendszermagba, telepítettük a DVD-meghajtót, itt az ideje tehát, hogy negyvenedik alkalommal is megnézzük mondjuk a „Shrek”-et. Ehhez már csak egy lejátszóprogram szükséges, és – minő szerencse – Linux alá kiváló DVD-lejátszóalkalmazások léteznek. Az alábbiakban a négy legnépszerűbbel foglalkozunk bővebben, ez az MPlayer, az Ogle, a VideoLAN Client és a Xine. További lejátszókat és DVD-s vonatkozású bővítményeket a Freshmeat webhelyén lehet találni.

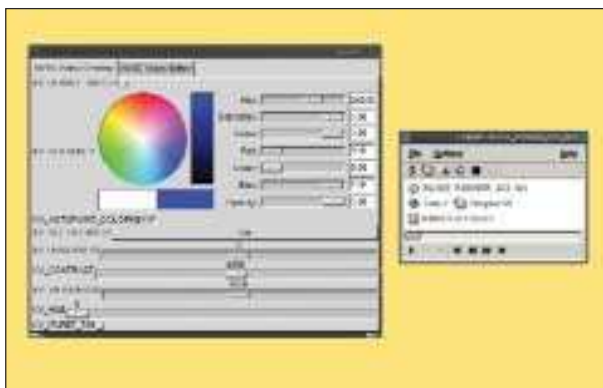
Várakozások

Különálló DVD-lejátszóm már van, így a programoktól elvártam, hogy támogassák a készüléken megszokott szolgáltatásokat. A kipróbált alkalmazások legalább a következő lehetőségeket kivétel nélkül biztosították: normál lejátszási műveletek (indítás, leállítás, szünet, gyorskeresés előre és hátra), jelenetvalasztás, feliratozás, hangbeállítások és a DVD-menük megjelenítése. Ha a programból megvalósított DVD-lejátszás mellett döntök, akkor, azt hiszem, jogos elvárás, hogy grafikus módból könnyen lehessen teljes képernyőre váltani, legyen mód a véletlenszerű ugrásokra a filmen belül, a lejátszási műveletek billentyűzetről történő vezérlésére és bőrök használatára.

A lejátszóprogramok lényegében megfeleltek ezeknek az elvárásoknak. A programváltozatok, használati feltételek és processzor-terhelések összehasonlítása *táblázatunkban* látható. Tulajdonképpen mindegyik alkalmazás remekül teljesített, legjobbat nem is tudnék kiválasztani közülük. Érdemes mindegyiket kipróbálni, majd mindenki használja azt, amelyik a legjobban megtetszett neki. Az alkalmazások közül az Ogle viszonylag egyszerű, ez csak DVD-lejátszásra képes, míg a többiek nagyjából egy súlycsoportba



1. kép Az MPlayer Blue Skin bőrrrel

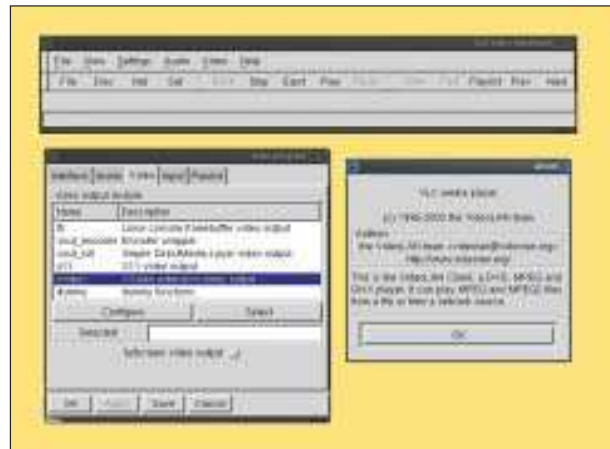


2. kép Az Ogle a Goggles grafikus felülettel és a gxvattr segédprogrammal

tartoznak. A programok lefordításának módját minden esetben megadom – aki a telepítés könnyebb módját szeretné választani, az az egyes alkalmazások weblapján talál RPM és egyéb formátumú előfordított csomagokat.

Win32 kodekek és a DeCSS

A lejátszóprogramok szolgáltatásaik jelentős részét Windows és Mac OS alá készült videokodekek (tömörítő és kibontó programkönyvtárak) segítségével valósítják meg. Kicsit pontosabban: az MPlayer és a Xine csak bizonyos kodekek segítségével tudja lejátszani a Microsoft ASF/WMV és az Apple QuickTime MOV formátumú fájlokat. Ezek a kodekek általában nem található meg a lejátszóprogramok forrását vagy futtatható változatát tartalmazó csomagban, de könnyen beszerezhetők, célszerűen valamelyik kodekgyűjtemény tagjaiként. A legtöbb esetben a normál csomagokkal is boldogulhatunk, de aki fanatikus mozirajongó, az nyugodtan telepítse akár az összes létező kodeket. Az, hogy ezeknek a kodekeknek a használata jogilag mennyire tiszta, az bizony kérdéses,



3. kép A VideoLAN Client Controls and Preferences párbeszédablaka

de mivel szabadon letölthetők az internetről, mindenkinek azt javaslom, hogy gyorsan gyűjtse is be őket – lehet, hogy később már nem teheti meg ilyen egyszerűen. A mindenki által ismert DeCSS a DVD-lemezek CSS-sel (Content Scrambling System) titkosított tartalmának dekódolását végzi. Az eredeti DeCSS program csak lefordított változatban volt elérhető, de hamarosan visszafejtették, majd a forrást közzétették az interneten. Természetesen a CSS visszafejtése miatt csúnya jogi csatározások kezdődtek; akit érdekelnek a részletek, az a <http://cyber.law.harvard.edu/openlaw/DVD> oldalon olvashat róluk. Az itt említett lejátszók egyikének sincs szüksége a DeCSS-re, ugyanis a futási idejű CSS-dekódolásra a nyílt forrású libdvdcss-t alkalmazzák. A lejátszóprogramok régiózárolás nélküli meghajtóval is használhatók.

MPlayer

Az MPlayer az itt említettek közül kétségtelül a legnagyobb tudású lejátszóprogram (lásd még: Linuxvilág 2002. március, 28. oldal, MPlayer for Linux). Határozottan több mint egyszerű DVD-lejátszó, az általa kezelt mozgókép- és hangformátumokat még felsorolni is nehéz lenne. Nagyjából kijelenthetjük, hogy ahány videofájl csak létezik, azt az MPlayer lejátszsa.

Az MPlayer DVD-lejátszás tekintetében megfelel a fenti elvárásoknak, illetve olyan további lehetőségeket is kínál – hangszín- és képkiegyenlítés –, amelyeket jómagam, miután megszoktam őket, már nehezen tudnék nélkülözni. A képjellemzők közül csúszkák segítségével a fényerőt, a kontrasztot, a színárnyalatot és a telítettséget szabályozhatjuk. A hang- és képkiegyenlítő egyaránt valós időben, rövid válaszidőkkel használható. Szintén fontos szolgáltatás a képarány módosításának a lehetősége, amelynek a segítségével beállíthatjuk, hogy a kép a képernyő mekkora területét foglalja el. DVD-lemezeknél én ezt a lehetőséget nem nagyon használtam, mivel a filmek általában széles képernyős vagy normál formátumban (sokszor mindkettőben) kerülnek rá a korongokra. Az MPlayer előre beállított normál, kétszeres és teljes képernyős megjelenítési módot is felkínál.

Az MPlayer hiába lopta be magát a szívembe, muszáj megemlítenem, hogy a lefordítása nem gyereksjáték. Mielőtt

hozzákezdénénk a fordításhoz, bogarásszuk végig a milliönyi beállítási lehetőséget (az eredményt a `./configure --help` paranccsal tekinthetjük meg). Csak példaként: az MPlayer grafikus felülete alapbeállítás szerint nincs engedélyezve, ezt kézzel kell megtennünk. A DVD-menüket a program a `libdvdnav` segítségével jeleníti meg, de az MPlayer 0.90-es leírása szerint a menütámogatás pillanatnyilag nem működik. Szerencsére a mellékelt leírás minden részletre kiterjedő, így a fordításról is részletes tájékoztatást nyújt. Mindenképpen mélyedjünk el benne, mielőtt panaszkodni kezdenénk az MPlayer levelezőlistáján. Az MPlayer kapcsán érdemes még kitérni arra, hogy fejlesztői a GCC 2.96-os bizonyos változatait hibásnak tartják, valamint az nVidia grafikus kártyák használatát sem javasolják. Álláspontjuk az, hogyha valaki a kifogásolt összetevők valamelyikét használja, akkor nem nyújtanak neki támogatást. Ez elég kellemetlen volt, mivel én nVidia kártyával rendelkezem, és a GCC 2.96-os változata van fent a gépemem. Mindettől függetlenül az MPlayert gond nélkül fordítottam le és használom, s tökéletesen elégedett vagyok vele. Ha valaki nem boldogul az MPlayer lefordításával, a Google segítségével biztosan talál megoldást. A sportszerűség kedvéért hozzá kell tennem, hogy az MPlayer fejlesztői és felhasználói közössége rendkívül segítőkész.

Ogle

Az Ogle szigorúan csak DVD-lejátszó – annak viszont nagyszerű! Az Ogle volt az első olyan lejátszóprogram, amelyik támogatta a DVD-menüket, illetve az egyéb apróságokat, mint a könyvjelzők, a léptetés, a többszörös hang, a digitális SPDIF hangkimenet és a kép adott területének kivágása és kinagyítása. Könyvjelző szolgáltatása egyedülálló; könnyű is rászokni a használatára. A filmet bármikor leállíthatjuk, a pillanatnyi helyet a könyvjelzővel megjelölhetjük, majd a segítségével később bármikor folytathatjuk a lejátszást. Első hallásra talán nem tűnik túl izgalmas dolognak, de hasznos apróság. A Goggles grafikus felülettel kiegészítve az Ogle arra is képes, hogy indítás után azonnal megkezdje a meghajtóban található lemez lejátszását.

Az Ogle önmagában a parancssorból használható. Lefordítását beépített grafikus felülettel is végezhetjük, de külső fejlesztőknek az Ogle honlapján elérhetővé tett felületei közül is válogathatunk. Én szeretem, ha egy egyszerű kis kezelőpanel kéznél van, és a Goggles felület is tetszik, de akinek más az ízlése, az Ogle-t billentyűzetről is könnyen kezelheti. Az utóbbi mellett szól, hogy a Goggles grafikus felület használatához a FOX eszközkészletre is szükség van, ami számos Linux-terjesztésből hiányzik. Az Ogle honlapjáról könnyen eljutunk a Goggles honlapjára, ahol tájékozódhatunk a FOX eszközkészlet beszerzésének a módjáról.

Az Ogle leírása alapján bukkantam rá az `xvattr` segédprogramra is. Az Ogle és az itt említett lejátszók alapértelmezett kimeneti illesztőprogramja az `xv`, ennek beállításait a lejátszóprogramokban található beállítások segítségével módosíthatjuk. Az `xvattr` ezzel szemben egy különálló program, amely lekérdezi a videokártya `xv`-vonatkozású jellemzőit, és lehetővé teszi parancssorból – vagy az `gvxvattr` nevű grafikus felületen keresztül történő – mó-



4. kép A Xine az alapértelmezett grafikus felülettel

dosításukat. Nekem igencsak jó szolgálatot tett, amikor az nVidia kártyámon alapbeállítás szerint engedélyezett kettős puffereles miatt – amit az `xvattr` segítségével ki tudtam kapcsolni – másodpercenként túl kevés képkockát jelenített meg a gép. Aki többet szeretne megtudni videokártyájának `xv`-képeiről, az `xvattr` segítségével eredményesen vizsgálódhat.

VideoLAN Client

A VideoLAN Client (VLC) egy több géptípuson is futtatható, nagysebességű hálózatokon videó- és hangtovábbításra képes kiszolgáló alapú megoldás egy része. A VideoLAN – egyébként kiváló – leírása szerint a tervezetnek része a VideoLAN Server (VLS) is, amely MPEG-1, MPEG-2 és MPEG-4 fájlok, DVD-k, digitális műholdas csatornák, digitális földi sugárzású tévécsatornák és élő videóképek egyes vagy többes címmel történő hálózati sugárzására képes. A tervezet másik tagja a VLC, amely MPEG-1, MPEG-2 és MPEG-4 fájlok, illetve DVD-k egyes vagy többes címmel való sugárzására ugyancsak használható kiszolgálóként. Ügyfélként MPEG-folyamok fogadására, dekódolására és megjelenítésére alkalmas, több operációs rendszer alatt is.

DVD-lejátszóként a VLC teljesítménye semmiben nem marad el a többi programétól. A VLC okozta viszont a legkisebb processzorterhelést, így lett a legjobb választás a hálózati üzemeltetésű vagy kisebb teljesítménnyel bíró gépek számára. A VLC grafikus felülete semmi izgalmat nem tartogat, de kényelmesen, jól használható. Aki nem akar százféle formátumot lejátszani vagy gépének a jellege az említetteknek felel meg, annak a VLC telepítését javaslom.

Xine

Ha az MPlayer és a Xine két versenylő volna egy futamban, akkor fej-fej mellett futnának be. A Xine fordítása jóval kisebb kihívást jelent, és fejlesztői a fordítóprogramra és a videokártyára sem olyan finnyások. A csomag minden szükséges könyvtárat tartalmaz, és alapbeállításokkal

fordítva grafikus felületet is kapunk. Alapszolgáltatásként támogatja a DVD-menüket, számos videoformátumot képes lejátszani, képkiegénylítője pedig az MPlayeréhez hasonló. A Xine grafikus felülete elsőre kicsit furcsának hat, de elrendezése átgondolt, használata pedig könnyű. A Xine teljesítménye röviden és tömören: kiváló. Amikor először mértem meg az általa okozott processzorterhelést, kicsit csalódottan szemléltem az eredményt, aztán hamar rájöttem, hogy egy, a videokimeneti illesztőprogramok próbálgatásakor megadott és elfeledett beállítás okozza a program gyenge teljesítményét. Korábban ugyanis az xshm illesztőprogram használatára állítottam be a Xine-t, de a későbbiek során figyelmen kívül hagytam, hogy ez a beállítás érvényben maradt. Magával az illesztőprogrammal egyébként semmi gond nem volt, egyszerűen csak arra figyeltem fel, hogy a Xine sokkal nagyobb számítási teljesítményt kötött le, mint a többi alkalmazás. Amikor kimeneti illesztőprogramként újra az xv-t jelöltem ki, a processzorterhelés látványosan visszaesett, én pedig nagyobb önmérsékletet próbáltam tanúsítani a Xine alapbeállításainak módosításában. A xine-check segédprogram futtatásával mindenki lemérheti, hogy a Xine vajon mekkora teljesítményt ígér a gépén.

Az egyetlen szolgáltatás, amit a Xine felületéből hiányoltam, a véletlenszerű fejezetválasztás. A jelenetek között ugyan lehet előre és hátra lépkedni, de nincs olyan grafikus ábrázolású jelenetfa, mint az Ogle vagy az MPlayer esetében. A DVD-menüből azonban gond nélkül tudunk jelenetet választani, ilyen módon amíg a lemez menüje ezt lehetővé teszi, addig a Xine sem áll utunkba. A Xine kapcsán még egy kifogás merülhet fel, mégpedig az, hogy forrásfája az összes szükséges könyvtárat tartalmazza. Akadnak néhányan, akiknek ez nem tetszik, de szerintem roppant kényelmes megoldás. Így nem kell feltérni az egész webet, hogy a csomag teljessé váljon – az eredeti összeállítás minden szükséges elemet tartalmaz, kivéve a Win32 és az Apple kodekeket.

A Xine alapértelmezett grafikus felülete könnyen használható, áttekinthető, de ha valaki nem tud megbékélni vele, akkor a Xine honlapján további felületeket is talál. Néme-lyik ezek közül kimondottan szép, de például olyanra is akad példa, amely a videokimenetet karakteres ASCII-képernyőként jeleníti meg.

A többi lejátszó rajongói táborához hasonlóan a Xine-közösség is több nagy forgalmú levelezőlistát tart fenn. Ha a kiváló leírás ellenére valamilyen kérdésünk megválaszolatlan maradt, a fejlesztők és felhasználók közössége minden bizonnyal meg fogja adni a választ.

Másolás és hardveres dekódolás

Nem szeretném újraírni a <http://bunkus.org> mindenre kiterjedő és kiváló útmutatóját, inkább csak összefoglalnám, hogy mi kell egy DVD-lemez lemásolásához. Vegyünk egy nagyméretű merevlemezt, telepítsük a mencoder vagy a transcode csomagot, illetve a hozzá tartozó dvd:rip grafikus felületet, majd kövessük a <http://bunkus.org> oldalon leírtakat. A DVD-másolás során számtalan beállítást változtathatunk meg, ezért ha jó eredményt akarunk elérni, tervezzünk előre. A <http://bunkus.org> oldalon a szerző DVD-lemezenként

legalább 10 GB szabad hely fenntartását javasolja.

A másolás még gyors processzor birtokában is több órán át fog tartani.

Nekem nincs DVD-íróm, így e tekintetben nem tudok tanácsokat adni, hacsak azt nem, hogy **Jörg Schilling** kiváló programját, a cdrecord-ot aligha tudjuk majd nélkülözni. Ahogy azonban a DVD-k másolásával, úgy az írással kapcsolatban is számos, a weben megjelent cikk elérhetősége szerepel írásom webes változatának források részében (<http://www.linuxjournal.com/article/7174.html>), ezekben részletesen olvashatunk a műveletről.

Miközben cikkemet írtam, a Linux Audio Users levelezőlista tagjaitól megkérdeztem, hogy melyik DVD-lejátszóprogramot használják. Az MPlayer és a Xine egyértelműen vezetnek, de az egyik tag megkérdezte, hogy vajon hardveres DVD-dekóderekkel is foglalkozom-e. Ezekkel semmi tapasztalatom nincs, de az olvasóktól minden ilyen témájú visszajelzést örömmel veszek.

Leírások és támogatás

Valószínűleg az itt szereplő lejátszóprogramok egyikének a fejlesztője sem akarta, hogy szemrehányást kapjon programjának hiányos leírása miatt. Az MPlayer, a VideoLAN Client és a Xine leírása kifejezetten bőszéges útmutatást nyújt a fejlesztőknek és a felhasználóknak egyaránt. Az Ogle leírása nem ennyire kimerítő, de ez a legkisebb tudású program, és sűgőoldalból is igazán kiváló jár hozzá. A jelentős forgalmat lebonyolító levelezőlisták és archívumaik révén bármelyik lejátszóprogramhoz kiterjedt támogatást kaphatunk.

Az utolsó képkocka

Az itt említett lejátszóprogramok mindegyike kifinomult, komoly múltra visszatekintő alkalmazás. A Linux kiváló multimédiás rendszerré fejlődött, hang- és videólejátszáshoz egyaránt a legjobb választások egyike. Minden olvasót szeretnék az írásomban szereplő programok kipróbálására biztatni és várom visszajelzéseiket. Mindehhez jó szórakozást kívánok, de mindenki legyen tudatában annak, hogy egy kicsit dolgoznia is kell majd, mire beüzemelje az alkalmazásokat.

Köszönetnyilvánítás

Szeretnék köszönetet mondani mindazon fejlesztőknek, akik keményen dolgoznak annak érdekében, hogy a Linux-felhasználók világszerte gondtalanul élvezhessék a DVD-zés örömeit. **Siggi Langauf** és **Bill Fink** sok segítséget nyújtottak a Xine megismeréséhez, és értékes bírálatot adtak írásomra vonatkozóan. Az esetleges hibákért az olvasók szíves elnézését kérem, minden helyreigazítást szívesen fogadok.

Linux Journal 2004. január, 117. szám



Dave Phillips

Zenész, tanár és író. Az Ohio állambeli Findlayben él. 1995-ben ismerkedett meg a Linuxszal, azóta tagja a linuxos zenei közösségnek. A Kiskapu Kiadó gondozásában megjelent Linux Zene és hang című könyv (ISBN: 963 9301 31 0) szerzője.



DVD-összeállítás

Készítsünk házi videófelvételeinkhez olyan sokrétű menürendszert, amelyet a néző egy DVD-lejátszón használni tud.

A DVD authoring költséges multság. A mindentudó, profi alkalmazások több ezer dollárba kerülnek, míg az olcsóbb termékek – mint például az Apple iDVD programja – csak korlátozott szolgáltatásokat kínálnak. Egy új nyílt forrású próbálkozás, a dvdauthor az olcsó, de profi szintű DVD authoring lehetőségét nyújtja Linux alatt. Bár még mindig nem támogatja a teljes DVD-szabványt, a fejlesztés gyors léptekkel halad előre és minden friss kiadás új tulajdonságokat tartalmaz. Írásunkban ismertetjük, hogyan lehet viszonylag összetett DVD-alkalmazást, fotóalbumot készíteni a dvdauthor-ral és egy szilárd alapokon nyugvó, nyílt forrású eszköztárral, az mjpegtools-zal kiegészítve. Bemutatjuk a dvdauthor által jelenleg támogatott szolgáltatásokat és azt, hogy miként alkalmazzunk nyílt forrású eszközöket asztali DVD-lejátszón használható DVD-R-lemezek készítésére.

Hogyan működik a DVD (gyorstalpaló)?

Egy DVD egy vagy több videóanyag-készletet (Video Title Set – VTS) tartalmaz, amelyek MPEG-2-es adatfolyam formájában tartalmazzák a képi adatokat. Egy lemezen legfeljebb 99 VTS lehet és minden készlet legfeljebb 99 fejezetre osztható tovább, lehetővé téve, hogy a DVD-lejátszóval a videojelfolyamon belül egy meghatározott pontra ugorjunk. Minden VTS-ben 8 különböző hangszáv és 32 feliratsáv lehet, amelyeket a néző szabadon váltogathat. Menürendszer is adható a készlethez, ami megkönnyíti a felhasználónak a felirat- és hangszávválasztást. Egy választható felső szintű menü, amelyet *video manager menu*-nek (VMGM) nevezünk, szolgál a különböző videóanyag-készletek közötti navigálásra. Egy VTS tartalmazza a fő filmet, egy másik a filmről készített dokumentumfilmet, a VMGM pedig lehetővé teszi a néző számára, hogy kiválasszassa, melyiket is szeretné megnézni.

A DVD formátum nem szüntette meg a különbséget a két fő videóformátum, az NTSC (elsődlegesen az Egyesült Államokban használatos) és a PAL (Európában és Japánban elterjedt) között. Az írásunkban foglaltak a PAL környezetre vonatkoznak, ezért a képkocka (frame) adatok és a felbontás adatok PAL rendszerűek, de külön felhívjuk a figyelmet a különbségekre, és az NTSC formátumhoz is kínálunk megfelelő beállításokat.

A DVD-szabvány fejlett tulajdonságokat is tartalmaz,



1. kép A főmenü képe

például a régió kódok használatát, egyazon műsor több nézőpontból való megtekinthetőségét, valamint egyszerű számítások elvégzésének lehetőségét a DVD-lejátszók beépített regisztereinek felhasználásával. Ezekről a tulajdonságokról nem sokat tudok, ezért nem is térek ki rájuk ebben a cikkben – a dvdauthor levelezőlistájáról könnyen további útmutatáshoz lehet jutni.

Tervezés

Mielőtt hanyatt-homlok belevágnánk a menükészítésbe, a feliratozásba és az összeállításba, érdemes papírral, ceruzával vázlatot készíteni a DVD leendő felépítéséről. A kereskedelmi DVD-készítő programok grafikus kezelőfelületet kínálnak a DVD-szerkezet létrehozására, de ilyen eszköz még nem áll rendelkezésére annak, aki Linux alatt szeretne DVD-t készíteni. Ahogy nemsokára meglátjuk, a parancssoros eszközöknek számos beállítási lehetősége van, így igen hasznos, ha nem fejben tartjuk, hanem egy papírra vázoljuk fel, hogy mit is szeretnénk. Most egy fotóalbumot készítek DVD-re, amelyen az észak-carolinai egyetemen (Chapel Hill) folytatott tanulmányaim alatt tavaly készített képeket fogom tárolni. Az egyszerűség kedvéért csak hat kép lesz minden kategóriában. Papíron megterveztem, hogy a fő menü (a VMGM) hat nyomógom-

bot fog tartalmazni, valamint egy rejtett hivatkozást, amellyel extra képekhez lehet jutni (csakúgy, mint a „gyári” DVD-k rejtett „húsvéti tojás” tulajdonságaival), ezenkívül egy zeneszám fog szólni a háttérben. A négy normál gomb négy menüpontot takar. Az egyes csoportok menürendszerre két további menüt, egy hangsávot, kiválasztható előnézeti képeket, valamint egy nyomógombot tartalmaz, amivel a következő adag előnézeti képre lehet lépni, ezenkívül lesz egy gomb az összes kép levetítésére, valamint egy másik a főmenübe történő visszalépésre. Hogy ne bonyolítsuk tovább a dolgot, a vetítés alatt ugyanaz a zene szól, mint a csoportmenü háttérben. A vetítés befejezése után a lejátszás a csoportmenübe tér vissza. A titkos hivatkozás egy rövid bemutatót indít el menü nélkül, de ennek két zenei sávja van, amelyek között lejátszás közben váltani lehet.



2. kép A főmenü maszkjának elkészítése

Hogy elkerüljem a keveredést, a fájlok elhelyezésére létrehoztam egy könyvtárszerkezetet. A képkönyvtár fogja tartalmazni a kész DVD-t, míg a nyers fotók a *photos/setN* könyvtárba, a videófájlok pedig a *titleN* könyvtárba kerülnek:

dvd

- title1
- title2
- title3
- title4
- title5
- mainmenu
- photos
 - set1
 - set2
 - set3
 - set4
 - set5
- image

A képek feldolgozása

A bemutatók elkészítéséhez a képeket videóklippé kell alakítani. Az *mjpegtools* csomagban található egy *jpeg2yuv* nevű eszköz, amellyel a JPEG-fájlokat YUV (ez a videó-

zásban használt színmodell) formájúra alakítja. Ezekből aztán az *mpeg2enc* MPEG-2 videófájl készítő. Az átalakítás megkezdése előtt azonban meg kell róla győződnünk, hogy a képek a DVD-nek megfelelően vannak-e megformálva. A képeket ajánlatos kézzel átalakítani, hogy megfeleljenek a PAL vagy az NTSC formátum képméretének (720×576 vagy 720×480 képpont), még mielőtt rájuk engedjük a *jpeg2yuv*-ot. Az *mjpegtools* csomag tartalmaz egy *yuvscal*er nevű eszközt, amely a bemenő YUV-adatot önműködően megfelelő felbontásúra méretezi át, de ez megváltoztathatja a kép oldalarányát, és így torzítást okozhat. A képek átalakításához az alábbi parancsot lehet használni:

```
Jpeg2yuv -n 125 -I p -f 25 -j picture1.jpg |
mpeg2enc -f8 -o slide.mpg
```

A *-n* kapcsoló utasítja a *jpeg2yuv*-ot, hogy hány képkockát készítsen – ebben az esetben 125-öt (ez 5 s, 25 kép/s sebességnél); NTSC esetében 25 helyett 29,97-tel lehet számolni. A *-I* kapcsoló állítja be a progresszív mód használatát (így nem vész el a kép egy része), a *-f* pedig a kívánt képsebességet határozza meg. Az *mpeg2enc -f* kapcsolójával lehet megadni, hogy DVD MPEG-2 videóadatfolyamot készítsen. Ezt az eljárást a DVD-re kerülő minden egyes képen meg kell ismételni, ezért a futtatásához érdemes egy gyors Perl- vagy héjprogramot készíteni.

Zene hozzáadása

Ha megvannak az egyes képek videóklipjei, a következő lépés a háttérzene hozzáadása, amely az összes klip lejátszása alatt szólni fog. A DVD-szabvány lehetővé teszi PCM, AC3 vagy MPEG-2 hangsáv használatát, változatos bitsebességek mellett. A fotóalbumhoz MPEG-2-t választottam – úgy döntöttem, hogy nincs szükség az AC3 vagy a PCM jobb minőségére. Az *mjpegtools* csomag tartalmaz egy *mp2enc* programot, amely a WAV-fájlból MPEG-2 fájl készítő. Egy másik eszköz, az *mp1ex* pedig egymásba fűzi az MPEG videóadatfolyamot és az egy vagy több hangfájlt. Ahhoz, hogy többhangsávú fájlt készítsünk, az *mjpegtools* CVS-változatára van szükség. Először fűzzük össze az egybetartozó videófájlokat a *cat* használatával, mivel az *mp1ex*-nek a tapasztalatok szerint gondjai vannak a több bemeneti fájl kezelésével.

```
cat *mpg > video.mpg
```

A *title5* könyvtárban (az 5. résznek két hangsávja lesz, ez a rejtett anyag) az *mp1ex*-et az alábbi módon futtattam:

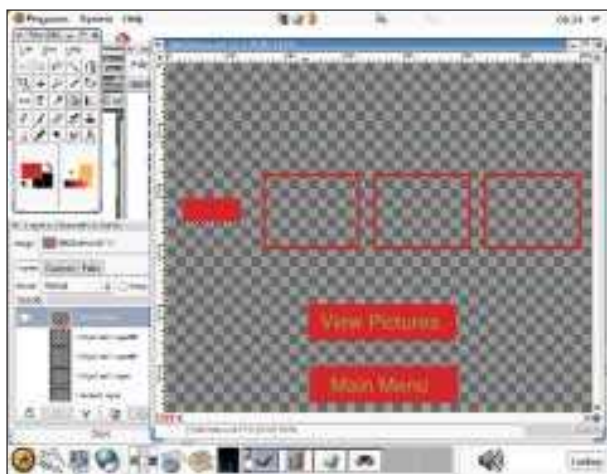
```
mp1ex -f8 -o video%d.mpg fotók/5adag/video mpg
↳ audio1.mp2 audio2.mp2
```

Ez a pillanatnyi könyvtárban egy sorozatfájlt hoz létre *video1.mpg*, *video2.mpg*, ..., *videoN.mpg* elnevezéssel, amelyekbe be van fűzve a két hangsáv, az *audio1.mp2* és az *audio2.mp2*. Az *mpeg2enc*-hez hasonlóan az *mp1ex -f8* kapcsolója teszi biztossá, hogy az új MPEG-fájl DVD-megfelelő legyen.

Egy dolog miatt kell aggódnunk, mégpedig, hogy az *mp1ex* a teljes hangsávot befűzi. Ha a hang hosszabb, mint a videóklipek összideje, az utolsó klip megnyúlik, de csak annyira,



3. kép Egy almenü betekintő képekkel



4. kép Az almenü maszkja

hogy képpel pótolja ki a hiányzó időt. Hogy ezt elkerüljük, vagy a képkocka sebességet kell módosítani a jpeg2enc használatakor, vagy szerkesszük megfelelő méretűre a szóban forgó hangfájlt.

Menük készítése

A DVD-menük készítésének eljárása a diák készítéséhez hasonló. A Gimpet használva készítsünk egy új képet 720×567 képpont (PAL) méretben, 75 dpi vízszintes és 80 dpi függőleges felbontással (NTSC esetében ezek 81 és 72 legyenek). Adjunk hozzá egy alfacsatornát, majd készítsük el a kívánt menüképet (lásd a főmenü alapképét az 1. képen). Amikor ez kész, adjunk hozzá egy újabb réteget és jelöljük ki a gombok helyét, ahogy a második képen is látható. Ezen a maszkon legfeljebb négy szín használható – én a pirossal jelöltem ki a gombok területét, és fehérrel írtam fel a szöveget, hogy akkor is látszódjon, amikor a gomb ki van választva. Ha a menükép már olyan, amilyennek szeretnénk, a háttérréteget JPEG-fájlként, a maszkot pedig indexelt PNG formában exportáljuk. Győződjünk meg róla, hogy a színek számát négyre állítottuk, illetve háromra, ha az átlátszó hátteret színeként mentjük. Ezeket

a lépéseket minden egyes menüképnél el kell végezni (további példaként lásd a 3. és 4. képet). Ahogy az előbb is, az MPEG-fájl létrehozására használjuk a jpeg2yuv és mpeg2enc programokat, majd az mplex segítségével fűzzük egybe a hangsávval. Bizonyos okokból a menünek egy hangfájltra van szüksége ahhoz, hogy működjön, így ha csendes menüt szeretnénk, egy rövid csendfájlt kell alkalmazni. A DVD-n a nyomógombok feliratkéiben vannak megvalósítva, ezért a dvdauthor submux eszközét használjuk arra, hogy a menü egy nyomógombját meghatározzuk. A program a feliratokat egy szöveges fájlból olvassa be, és egybefűzi őket a videojelfolyammal. A submux fájl formátuma a következőképpen fest:

```
<fájlnev><start_idő><vége_idő>
<x_eltolás><y_eltolás><48 bites számok>
```

Lehet több bejegyzés is a .sub fájlban, de csak egy sorra van szükség a menükhöz:

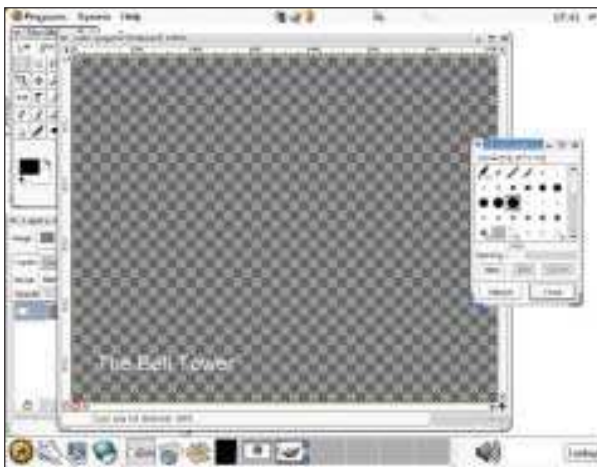
```
Mainmask.png 00:00:00.00 00:00:00.00 0 0 0 1 0 1
```

Az indítás (start) és a leállítás idejét (end times) nullára állítva utasítjuk a DVD-lejátszót, hogy a feliratot folyamatosan a képernyőn tartsa. A 0 1 0 1 sorozat kapcsolja a kiválasztott gombot pirosra, amennyiben a dvdauthor által használt alapértelmezett palettát alkalmazzuk. A négy különböző szám (0–255 között) vezérli az átlátszóság mértékét az indexelt PNG színeiben. Lehet kísérletezni is, hogy a különböző értékek milyen eredményeket hoznak létre, de a példában megadott értékek használható eredményt adnak. Elkészülvén a felirattájjal, futtassuk le a submux-ot, hogy a feliratokat a videojelfolyamhoz adjuk:

```
Submux menu.sub < video.mpg > menu.mpg
```

Feliratok

Ahogy azt már említettem, egy DVD akár 32 különböző feliratot is tartalmazhat egy VTS-ben. Bár ezeket felirattal hivatkoznak, valójában átfedő képfájlokról van szó, nem szövegről, vagyis ezeket más célokra is fel lehet használni. Jó példa erre a Mátrix DVD-n lévő „Kövessd a



5. kép Egy feliratkép

fehér nyulat” lehetőség. Feliratot készíteni nem bonyolult feladat, gyakorlatilag ugyanazzal a művelettel készül, mint a menü, csak háttérrel együtt. Az 5. képen egy feliratképre látható példa, ami egy fényképnek ad címet. Ennek a feliratnak a submux leírása – ami két másodpercig jeleníti meg a szöveget – a következő:

```
Sub1.png 00:00:00.00 00:00:02.00 0 0 0 255 0 255
```

Példa-DVD-nken különálló, rövid MPEG-2-klippekkel dolgozunk, ezért mindegyiknek saját submux leírása van. Hosszabb klipek esetében a submux fájl általában több bejegyzést tartalmaz.

Rakjunk össze mindent!

Elkészítettük a videófájlokat, feliratot kaptak, kész a menü – nincs más dolgunk, mint mindezekből a dvdauthor felhasználásával összeállítani a készterméket. A dvdauthor-nak két üzemmódja van, egy a normál feliratokhoz és egy a VMGM menü létrehozásához. A VMGM a menüön kívül nem tartalmazhat más videóadatokat. Kezdjük az első címmel:

```
dvdauthor -o tmp -m -P
↳ -b 239x397-489x457,subtitle32+vts1
↳ -b 239x500-489x560,vmgm1
↳ -b 27x223-127x263,subtitle32+vtsm.2
↳ -b 165x184-325x305,subtitle32+vts1.1
↳ -b 352x184-512x305,subtitle32+vts1.2
↳ -b 539x184-699x305,subtitle32+vts1.3
```

```
title2/titlemenu1.mpg -m -P
↳ -b 239x397-489x457,subtitle32+vts1
↳ -b 239x500-489x560,subtitle32+vmgm1
↳ -b 27x223-127x263,subtitle32+vtsm.1
↳ -b 165x184-325x305,subtitle32+vts1.4
↳ -b 352x184-512x305,subtitle32+vts1.5
↳ -b 539x184-699x305,subtitle32+vts1.6
```

```
title2/titlemenu2.mpg -t -P
```

```
title2/v1.mpg title2/v2.mpg title2/v3.mpg
↳ title2/v4.mpg title2/v5.mpg title2/v6.mpg
↳ -i post=3Dvtsm
```

A fenti kód két menüt határoz meg, mindegyiken hat gombbal, összesen hat videóklippel (a dvdauthor a klipek közötti átmenetet fejezetpontként határozza meg) és egy post-video utasítással, amelynek hatására a DVD-lejátszó visszatér a *title* menühöz. A koordinátarendszer ugyanaz, mint a Gimpben, így könnyen meg lehet állítani a használni kívánt régiókat. Egy gomb változatos akciókra képes (lásd a dvdauthor -help-et a lehetőségek kilistázásához). A fenti példában az első gomb beállítja a nullás hangsávot és kiválasztja a szükséges feliratfolyamot (a 0-s folyam érdekes módon a 32-es, az 1-es folyam a 33-as és így tovább), és lejátsza a hozzárendelt videófájlt. A második gomb a lejátszót visszaküldi a VMGM menübe, a harmadik pedig a következő címmenüre visz. A többi három gomb a videófolyamon belül a megadott fejezetre

ugrik, attól függően, hogy melyik betekintő képet választjuk ki. A -o kapcsoló a kimeneti könyvtárat adja meg, ami a mi esetünkben az *image/* alkönyvtár. A program növekményes módon működik, tehát ha ugyanazt a parancsot kétszer adjuk ki, akkor egy új címkészletet hoz létre, nem az eredetit frissíti.

A főmenü hőzelőszőr a dvdauthor-nak meg kell adnunk a -T kapcsolót, ebből fogja tudni, hogy el kell készítenie a szükséges VMGM-adatot. Ezután a már elkészített VTS-fájlokat kell a megfelelő gombokhoz rendelnünk:

```
dvdauthor -o tmp -T -m -b 497x89-693x136,vtsm1
↳ -b 497x138-693x187,vtsm2
↳ -b 497x189-693x239,vtsm3
↳ -b 497x240-693x289,vtsm4
↳ -b 426x405-490x474,vts5
↳ -P mainmenu/mainmenu.mpg
```

Kipróbálás és sütés

Nem árt, ha a lemezre írás előtt kipróbáljuk az elkészített DVD-lemezképet. Ezt egyszerűen megtehetjük a Xine használatával, csak annyi a dolgunk, hogy a Xine-nak tulajdonságként (argumentum) a lemezkép helyét adjuk



6. kép A DVD iBookon futtatva, a rejtett hivatkozást kiválasztva



7. kép Az egyik bemutató futtatása az iBookomon

meg, valahogy így:

```
xine dvd: /út_a_dvd_gyokerhez/VIDEO_TS/
```

és a Xine úgy fog működni, mintha lemezzel végezné a lejátszást. Mivel a dvdauthor növekményes működésű, akár minden egyes cím elkészültekor ellenőriztjük a Xine-nal, hogy a terveink szerint működik-e.

Ha a DVD-vel elégedettek vagyunk, következhet a lemezre írás – én a cdrecord-provdv alkalmazást használok ilyesmire. A művelet hasonló, mint a cdrecord esetében, vagyis először az mkisofs segítségével el kell készítenünk az ISO lemezképet:

```
mkisofs -o <kimeneti_fájlnev>
↳ -dvd-video <út_a_DVD_gyökehez>
```

Az így elkészült ISO-lemezképet az alábbi paranccsal tudjuk kisütni:

```
cdrecord.provdv dev=3D0,0,0
↳ -pad -dao <út_a_DVD_lemezképhez>
```

A 6. és 7. képen a kész DVD látható a saját iBook gépemem futtatva.

Összegzés

Bár a dvdauthor-nak nincs olyan könnyen használható kezelőfelülete, mint egyes profi alkalmazásoknak,

ugyanazokat a lehetőségeket nyújtja a DVD-készítéshez, mint amiket Hollywood is használ, mindössze a DVD-R nyersanyag áráért. Remélem, írással sikerült betekintést engednem a DVD-összeállítás alapjaiba és néhány ötlettel tudtam szolgálni saját alkalmazások készítéséhez.

Jó alkotást kívánok!

Linux Journal 2004. január, 116. szám



Ian Pointer (ian@snappishproductions.com)
Számítógépes tudományból diplomázott az Egyesült Királyságban, jelenleg nem áll alkalmazásban. Túl sok DVD-je van otthon és most még többet tervez készíteni.

KAPCSOLÓDÓ CÍMEK

cdrecord.provdv ➔ <http://www.fokus.gmd.de/research/cc/gclone/employees/joerg.schilling/private/cdrecord.html>

dvdauthor ➔ <http://dvdauthor.sf.net>

dvdrttools ➔ <http://www.nongnu.org/dvdrttools>

The GIMP ➔ <http://www.gimp.org>

mjpegtools ➔ <http://mjpegtools.sf.net>

Xine ➔ <http://www.xinehq.de>





Felületfüggetlen CD-tárgymutató

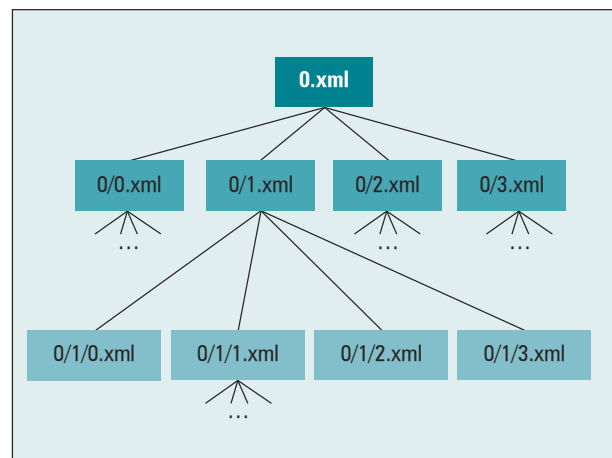
Jó lenne a CD-ROM-ok tartalmát egy olyan keresővel kereshetővé tenni, amelyik állandó indexfájlok használatával bármelyik böngészőben képes futni. A JavaScript és az XML ezt lehetővé teszi.

Nemrég az egyik ügyfelem számára CD-katalógust készítettem. Az volt a feladatom, hogy kulcsszavakra lehessen rákeresni. Már meglévő megoldások után kutattam, de sajnos csak olyanokat találtam, amelyek egy adott zárt kódú operációs rendszer alatt futottak, telepíteni kellett őket a felhasználó számítógépére, és minden példány után használati díjat kellett fizetni. Az ilyen telepítési feltételek korlátozóak és hosszú távon nagyon sokba kerülnek. Továbbá nem minden CD-felhasználó futtatja azt a bizonyos zárt kódú operációs rendszert, így a lehetséges felhasználók táborát mesterségesen csökkentenénk. Miközben a feladat megoldásán rágódtam, felfigyeltem a Linux Journal Archive CD-re. Rájöttem, hogyha valaki már megoldotta ezt a feladatot, az biztosan szerepel az LJ Archive CD-n. Képzeltethetek a csalódásomat, amikor kiderült, hogy ugyan az LJ Archive CD-nek van tárgymutatója, de nem kereshető. Nekem kell megoldani a feladatot, magamnak kell rábukkannom a megoldásra is. Szerencsére rátaláltam a jsFindre.

Felhasználási feltételek

A jsFind terjesztésre és felhasználási feltételeire már viszonylag korán gondoltam. A korai változatokat megmutattam munkatársaimnak, akik azt javasolták, hogy kövessem a szokásos módszert: előbb védessem le a kódot és utána próbáljam eladni. Ez azt jelentette volna, hogy a jsFind ugyanazt a gyakorlatot követi, mint versenytársai. Szívesebben foglalkozom kódolással, mint eladással, és amúgy sem hiszem, hogy a terméknek túl nagy piaca lenne. Inkább legyenek olyan CD-ROM-ok, amelyeknek a tartalmában az összes böngészővel bármelyik operációs rendszer alatt lehet keresni.

A GPL (GNU Public License) sokkal inkább megfelel a céljaimnak. Ha a jsFind szabadon terjeszthető, akkor a saját érdemei miatt fog elterjedni, és a közösség további fejlesztésekkel fog hozzájárulni a kódhoz. A zárt kódú rendszerek egyik célja, hogy a felhasználót bezárják az adott program börtönébe. Ha például egy CD-ROM-hoz mellékeltem keresőt csak egyetlen böngészővel és egy adott operációs rendszer alatt lehet használni, akkor a felhasználó kénytelen azt a bizonyos operációs rendszert



A jsFind B-fát hoz létre, amelyben minden csomópont egy XML-fájl

futtatni. A CD-ROM előállítójának muszáj lesz mindig megvennie az adott operációs rendszerhez való fejlesztőeszközöket, hogy naprakész maradjon. A folyamat oda vezet, hogy a gyártók és a felhasználók nem tudnak mást használni, csak azt a bizonyos zárt kódú operációs rendszert. A GPL feltételei szerint kiadott jsFind megtöri ezt az ördögi kört.

A megvalósítás

A jsFind kulcsszókereső-motor egy kis, ötszáz soros JavaScript-program. Minden böngésző, amelyik támogatja a DOM Level 3 JavaScript-bővítményeket, képes betölteni az XML-fájlokat. A Mozilla, a Netscape és a Microsoft Internet Explorer jelenlegi változatai mind támogatják ezeket a kiterjesztéseket, és a Konqueror következő változata is valószínűleg fogja. A tárgymutató XML-fájlokban tárolódik, a JavaScript pedig hatékonyan keres bennük, és előállítja az eredményt. A keresés eredményeit az őt lekérő weboldalra szintén JavaScript postázza vissza.

A jsFind tekintetbe veszi, hogy a CD-ROM tartalma állandó – a webbel vagy más változó adathalmazzal ellentétben a CD-ROM tartalma a legyártása után nem fog megváltozni. A SWISH-E jobban használható dinamikus indexelésre, különösen, ha valaki olyan szerencsés, hogy külön



A keresési példa eredménye

kiszolgálót állít be a kulcsszókeresésre. A jsFind ezért csak azt feltételezi, hogy adott egy JavaScriptet ismerő webböngésző és egy pár böngészhető fájl – ez jelentős megszorítást jelent a lehetséges megoldásokra nézvést. A legtöbb indexelő algoritmus a beszúrás, a frissítés, a törlés és kiválasztás ideje közötti egyensúly megteremtésén fáradozik. Mivel a CD-ROM megváltoztathatatlan, soha nem kerül sor törlésre vagy frissítésre. A beszúrásra a CD megírása előtt kerül sor és az időigénye érdektelen. A felhasználó szemszögéből a kiválasztás ideje lesz a legfontosabb. A kis tárhelyigény is követelmény, hiszen egy átlagos CD-re 700 MB-nál több anyag nem fér fel.

A feltételek figyelembevételével az indexelő algoritmusokat ismét megvizsgálva érdekes eredményre jutunk: a leggyakrabban B-fákat és hasító táblákat (hash) használnak. Azért választottam a B-fákat, mert a fájlrendszer a fájlokat faszervezetbe rendezve tárolja, és ezt felhasználhatjuk a B-fa szerkezetének a tárolására is, helyet takarítva meg ezáltal. Ezenkívül elemezni lehet a kulcs-hivatkozás párokat és kiegyensúlyozott B-fát lehet létrehozni. Az XML-fájlok szerkezete a lehető legegyszerűbb, a helyfoglalás további csökkentésének céljából egybetűs elemeket használtam.

A B-fák leírása

A B-fa elnevezésű adatszerkezetet gyakran használják adatbázisok indexeléséhez és tárolóeljárásokhoz. Hatékonyan, kis időigénnyel kereshető, és a tárolást, illetve a visszakeresést tömbösen oldották meg, ez pedig jól együttműködik a mostani vasakkal. A B-fa csúcsokból áll, amelyeknek rendezett valakijük van. Minden kulcs egy hozzárendelt adathalmazra hivatkozik. Ha a kért kulcs a rendezésben két kulcs közé esik, akkor hivatkozást kapunk vissza egy másik kulcsokból álló csúcsra. A kiegyensúlyozott B-fa olyan, amelynél a keresésnél betöltendő csúcsok legnagyobb száma kicsi.

A jsFind B-fát hoz létre, amelyben az XML-fájlok jelentik a fa csúcsait és a fájlrendszer könyvtárai a hivatkozások egy másik csúcsalmazra. Ez lehetővé teszi, hogy a B-fa szerkezetének egy részét a fájlrendszer valósítsa meg.

1. lista Exportálás XML-be

```
$ swish-e -f mystuff.index -T INDEX_XML >
  ↳ mystuff.xml
```

Ennek az XML-nek a szerkezete ilyen:

```
<index>
  <word>
    <name>itt_egy_kulcsszó</name>
    <path freq="11" title="Valami okosság">
      /cdrom/blah.html
    </path>
    <path freq="10" title="További menő dolgok">
      /cdrom/blah2.html
    </path>
  </word>
  <word>
    ...
  </index>
```

2. lista A Mystuff.keys fájl kulcsszavai

```
$ tail mystuff.keys
you 134910
for 138811
i 149471
in 168657
is 179815
of 252424
and 273283
a 299319
to 349069
the 646262
```

Ha minden XML-fájl ugyanabban a könyvtárban helyezkedik el, a fájl megnyitási idő hosszú lenne, ezért a hatékonyság érdekében a fájlrendszerben szükség van az alkönyvtárak használatára.

Rövid kezelési leírás

A végfelhasználóknak nem kell ezzel törődniük. Nekik elég a keresett szót beírniuk a weblapon, és a jsFind visszaadja a kulcsszavakat tartalmazó oldalakra mutató hivatkozásokat. Nem kell telepíteni, nem kell vacakolni a beállítással, minden megy, mint a karikacsapás.

Ha tartalomfejlesztő vagy, a te életed egy kicsit nehezebb. A jsFind eszközkészlet megkísérli megkönnyíteni a munkádat. A kezdéshez elég a Perl és sok gépidő, hogy elkészítsd a tárgymutatót. Valószínűleg az összes böngészőt érdemes beszerezned, hogy összevetethes az eredményeket. Egyetlen példa `makefile` található a jsFind-terjesztésben, de több lépcsőben kell az egyéni igényeknek megfelelően testreszabni.

Első lépésben meg kell szerezni a kulcsszavak és a hivatko-

3. lista Tárgymutató B-fába rendezett XML-fájlja alakítása

```
$ mkindex.pl mystuff-filtered.xml 25
blocksize: 20
keycount: 101958
depth: 4
blockcount: 5098
maximum keys: 194480
fill ratio: 0.524259563965446
bottom fill: 92698
working: 11%
```

zások adathalmazát, méghozzá XML formátumban. A SWISH-E program saját magam foltozta változatát használok a tárgymutató kinyeréséhez és létrehozásához, majd az eredményeket a jsFind Perl-programjai számára emészthető XML formátumban exportálok. Feltéve, hogy a SWISH-E által gyártott tárgymutató a *mystuff.index* fájlban van, az 1. listában látható parancs végzi el az XML-be történő exportálást.

Az XML-fájl a kulcsszó alapján van rendezve. A kapott adathalmaz valószínűleg még túl nagy, mert a SWISH-E nem foglalkozik a gyakori szavak kiszűrésével. Az eredményt két Perl-programmal szűrhetjük: az *occurrences.pl* és a *filter.pl* segítségével. Az *occurrences.pl* elkészíti a kulcsszavak listáját, és meghatározza, hogy hányszor fordulnak elő a tárgymutatóban:

```
$ occurrences.pl mystuff.xml | sort -n -k 2 >
mystuff.keys
```

A *mystuff.keys* fájl minden sorában egy kulcsszó található, amit előfordulásainak a száma követ (2. lista).

Ezen a ponton véget is ér a kizárandó kulcsszavak listájának hallatlanul bonyolult elkészítése. A kulcsfájlban szerepeljenek azok a szavak, amelyeket a végső tárgymutatóból ki szeretnél hagyni. A saját fájl szerkesztésénél még jobb ötlet, ha a háromszáz leggyakoribb angol szó listáját megszerzed a <http://www.zingman.com/commonWords.html> címről.

Ezután futtasd a szűrőt. A csomagban mellékelt *filter.pl* nevű Perl-program elkészíti az eredményhalmazt. Jelenleg arra lett beállítva, hogy minden egybetűs tárgymutatókulcsot kiszűrjön (kivéve a C betűt). A két számjeggyel kezdődő kulcsokat (azaz a 3com és társai rendben vannak), valamint mindent mást, ami a kivétel-fájlban meg van adva, ugyancsak kiszűr:

```
$ filter.pl mystuff.xml mystuff.keys
->mystuff-filtered.xml
```

Ez a lépés meglehetősen hosszú ideig tart. Ellenőrizd, hogy a végső fájl elfér-e a rendelkezésre álló tárhelyen. A végső tárgymutató mérete nagyjából 75 százaléka lesz a szűrt tárgymutatónak. Ha ez túl nagy, a méretét egy nagyobb kulcsszó-kivételista megadásával leszoríthatod.

A következő nagy lépés magának a tárgymutatónak a létrehozása. Létezik egy olyan parancsfájl, amelyik a tárgymu-

tatót B-fába rendezett XML-fájlokká alakítja (3. lista). A következő megfontolandó feladat a paraméterezés. A *blockcount* értéke azt jelenti, hogy hány csúcsot kell létrehozni a B-fában. Minden csúcs egy kulcsfájl, egy adatfájl és egy könyvtárat jelent. Ha a fájl és könyvtárak száma túl nagy, a *blocksize* növelésével be lehet szabályozni. Ha a *blocksize* túl nagy, a keresés lelassul. A *bottom fill* értékkel egyensúlyban lehet tartani. Ha ennyi kulcsot teszünk az alsó sorba, az alsó sor lezáródik és nem lehet több csúcsot létrehozni. Így egy kiegyensúlyozott fát kapunk.

Ha minden jól megy, a pillanatnyi könyvtárban három fájl jön létre: a *0.xml*, a *_0.xml* és a *0* könyvtár – ezek a tárgymutató fájljai. A következő lépésben az eredményt a mellékelt példa segítségével illeszd be a saját HTML/JavaScript-oldaladba. Az eredményeket a mellékelt programrész kapja meg, és ezt kell visszaküldeni a pillanatnyi weboldalra. A példa JavaScriptet használ dinamikus HTML előállítására.

Összegzés

A jsFind sokféleképpen fejleszthető tovább, és bizonyára ezek a fejlesztések meg is fognak valósulni, amikor a nyílt forrás hívei elkezdik használni a programot. Ilyen lehetőség lenne például a képarcívumok bélyegképekkel való keresése, a többlapos eredménykijelzés vagy a böngészőmegfelelőség-ellenőrzés. Ezek a mostani kódból kiindulva mind megvalósíthatók. A Linux Journal 2002-es Archive CD-jén már a jsFind keresőmotor működik. Ha CD-ROM-okat állítasz össze, a zárt kódú operációs rendszerekre elérhető megoldások helyett használd inkább a jsFindot. Olcsóbb lesz és növeli a lehetséges felhasználók számát. A végfelhasználók jobban örülnek, ha semmit nem kell telepíteniük a gépükre, és nem kell egy másik operációs rendszert elindítaniuk, ha keresni akarnak egy CD tartalmában. Ezt a programot valószínűleg másra is fel lehet használni, ezért hát tekintsük nyílt forrású szerszámosládánk újabb elemének.

Linux Journal 2003. december, 116. szám



Shawn P. Garbett

Tanácsadó több mint 15 év tapasztalattal a háta mögött a Unix-rendszerek és azok orvosi alkalmazásainak a területén. Kedveli a dzsesszt és a fordultatos történeteket.

KAPCSOLÓDÓ CÍMEK

GNU <http://www.gnu.org>

Josh Rabinowitz „Hogyan indexeljük?”

Linuxvilág 2003. szeptember

<http://www.linuxvilag.hu>

A jsFind honlapja

<http://www.elucidsoft.net/projects/jsfind>

Hangkezelés a Pd-vel

A Pd segítségével akár egy visszhangszűrő vagy valamilyen más hangeszköz vizuális létrehozására is képessé válunk.

A Pure Data (Pd) valós idejű vizuális programozói környezet hang- és egyéb multimédia-alkalmazások létrehozására. Segítségével foltokat (patch) hozhatunk létre, amelyek a hang- és képadatokon valamilyen műveletet hajtanak végre. Ezeket a foltokat a program vizuálisan ábrázolja: rajzolásal adjuk meg, hogy a jelek adatai merre haladjanak és mi történjen velük. A folyamat hasonlít a különböző részekből felépülő analóg szintetizátorok programozásához, és jól illeszkedik a hang- és videóalkalmazások programozásához. A jelek a rendszerbe befelé és abból kifelé haladnak, és a rendszeren való áthaladás közben megfelelő módon módosíthatjuk őket. Terjedelmi okokból ez a cikk csak a Pd kiforrottabb hangképességeit mutatja be. Ha később a képfeldolgozó bővítményeket (plugin) is ki szeretnénk próbálni, látni fogjuk, hogy az elvek nagyon hasonlóak a hangfeldolgozó részben megismertekhez.

Hangjelek és üzenetek

A Pd az adatok két alaptípusát különbözteti meg: az üzeneteket és a hangjeleket. Az üzenetek a MIDI hangkimenethez hasonlóan csak szórványosan jönnek létre, számokat vagy karakterláncokat tartalmazhatnak, és például olyan üzenetek átadására használatosak, mint az „állítsd a kimenet erősítését x értékre”. A hangjelek állandóan jelen vannak, a hang átvittele a DSP (Digital Signal Processor) minden egyes bekapcsolásakor megtörténik. A Pd belsejében a hangok 32 bites lebegőpontos számok formájában tárolódnak, ami azt jelenti, hogy a hagyományos analóg vagy digitális

hangfeldolgozással ellentétben a Pd jelei szinte bármilyen amplitúdóval bírhatnak. A feldolgozás során létrehozhatunk egy nagyon halk hangot az egyik szakaszban, amit egy másikban minőségromlás nélkül erősíthetünk fel. Természetesen, amikor a jeleket valamilyen eszközre engedjük, az értéknek a használható, -1 és 1 közé eső tartományba kell esnie, különben levágja (clipp) a hanghullám tetejét.

Az üzeneteket és hangjeleket az alább ismertetésre kerülő különféle dobozok (boxes) segítségével módosíthatjuk – ezeket összeillesztve az összekapcsolt dobozokat foltnak (patch) hívjuk.

A dobozok – dolgozzunk is egy kicsit!

Minden feladatot a dobozok végeznek el. A Pd a dobozok négy fő típusát különbözteti meg: objektum, üzenet, grafikus felület (GUI) és megjegyzés. Ezek a dobozok műveleteket hajtanak végre az üzeneteken és a hangon, lehetővé téve a felhasználói bemenetek megadását és az elvégzett tevékenységek feljegyzését. Az objektumdobozok további két típusra oszlanak: vezérlőobjektumokra és hullámvonalobjektumokra. A vezérlőobjektumok az üzenetekkel foglalkoznak, ezért a tevékenységük nem folyamatos. A hullámvonalobjektumok a hangjelekkel dolgoznak, emiatt állandóan működnek.

Az üzenetdobozok a tartalmukat a kimeneti kapujukra irányítják, amikor a felhasználó rájuk kattint vagy

	1. ábra Egy objektumdoboz
	2. ábra Egy üzenetdoboz
	3. ábra Egy GUI-doboz
	4. ábra Egy megjegyzésdoboz

amikor üzenet érkezik a bemenetükre. A GUI-dobozok egyszerűen csak hivatkoznak olyan dobozokra, amelyekkel kapcsolatba léphetünk, ilyen a bal oldalon lévő számdoboz is. Végül a megjegyzések teszik lehetővé szövegek bevitelét a foltba, más hatásuk nincs.

A Pd elindítása és az első foltunk létrehozása

Feltéve, hogy már túl vagyunk a Pd fordításán és telepítésén, meg kellene próbálnunk elindítani. Először is bizonyosodjunk meg róla, hogy beállítottuk a Pd futtatható fájljának setuid bitjét, és hogy a fájl tulajdonosa a rendszergazda. Bár biztonsági kockázatot jelenthet, ez feltétlenül szükséges a valós idejű ütemezéshez, ha a rendszergazdán kívül más felhasználóként is futtatni szeretnénk a Pd-t. Ha ezt nem tesszük és nem aktiváljuk a valós idejű ütemezést, akkor kattanásokat és pukkanásokat fogunk hallani minden olyan esetben, amikor egy másik folyamat – akár az X-kiszolgáló – bármilyen tevékenységgel próbálkozik.

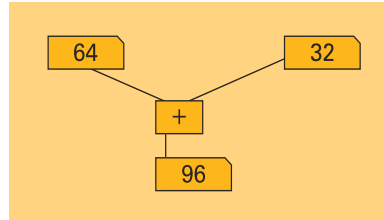


1. kép A főablak

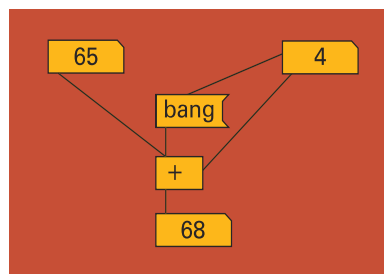
A Pd futtatásánál használjuk a `-rt` kapcsolót és emellett azokat a beállításokat, amelyeknek még szükségét érezzük. Javasolom a `-verbose` kapcsolót is, mivel a Pd amúgy nem túl bőbeszédű, és ezzel a beállítással hasznos ismeretekhez juthatunk. Ezután egy ahhoz hasonló ablakot kell látnunk, amelyet az 1. kép is mutat. Az **IN** és **OUT** feliratú dobozok a be- és kimenet hangszint csúcserőértékeit mérik, és a **peak meters** (csúcscintmérés) jelölőnégyzetre kattintva engedélyezhetők. Ha valamelyik túllépi a megengedett szintet, a rendszer levágja, és a megfelelő CLIP-doboz piros színű lesz. A **DIO errors** gomb abban az esetben kezd villogni, ha a bemenetben vagy kimenetben valamilyen összehangolási hiba lép fel. A gombra kattintva a legutóbbi hibák listáját jeleníthetjük meg. A **compute audio** (hang számítása) jelölőnégyzet a hangfeldolgozást kapcsolja be, illetve ki. Először is hozzunk létre egy új, üres foltot, amelyen dolgozhatunk (**File>New**) – ebből egy olyan egyszerű foltot fogunk készíteni, amelyik a „Szia Világ!” szöveget írja ki a szabványos kimenetre. Szükségünk van tehát egy üzenetdobozra, amelyben a „Szia Világ!” szöveget tároljuk, és egy objektumdobozra a nyomtatás végrehajtásához. Mindkettő a **Put** (elhelyezés) menüből hozható létre. Gyorsbillentyűk használatára is lehetőségünk nyílik: a **CTRL-1** az objektumdobozt, a **CTRL-2** pedig az üzenetdobozt helyezi el. Amikor ezzel készen vagyunk, kattintsunk a dobozra és gépeljük be a megfelelő szöveget, majd kapcsoljuk az üzenetdoboz alján lévő kimenetet az objektumdoboz tetején látható bemenethez. Foltunknak a 5. ábrán láthatóhoz hasonlóan kell festenie. A **CTRL-E** billentyűkombinációval lépünk ki a szerkesztőmódból. Próbáljunk rákattintani a „Szia Világ!” üzenetdobozra; ha minden rendben van,



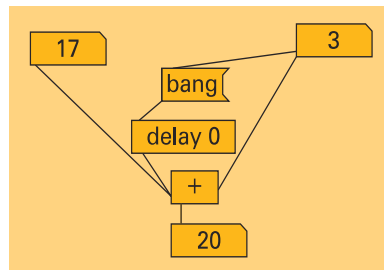
5. ábra A klasszikus „Szia Világ!” példa



6. ábra Két szám összeadása



7. ábra A forró és hideg bemenet kezelésének naiv módszere



8. ábra

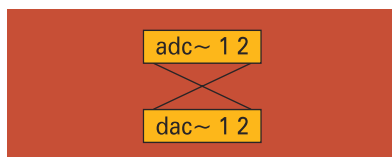
A forró és hideg bemenetek helyes kezelése

akkor az üzenetnek meg kell jelennie abban a terminálablakban, ahonnan a Pd-t elindítottuk.

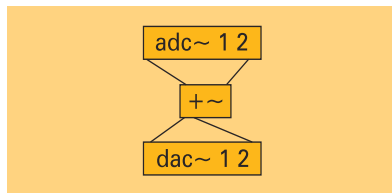
Az üzenetek használata

Most próbáljunk ki egy bonyolultabb példát: készítsünk egy olyan foltot, amely két számot ad össze és megjeleníti az eredményt. Ehhez hozzunk létre egy olyan foltot, amelyet a 6. ábrán látunk, két számdobozzal a tetején, középen egy objektumdobozzal, és lent ismét egy számdobozzal. Ezután lépünk ki a szerkesztésből (ismét **CTRL-E**) és próbáljuk megváltoztatni a fenti számokat. Ezt rákattintással és az

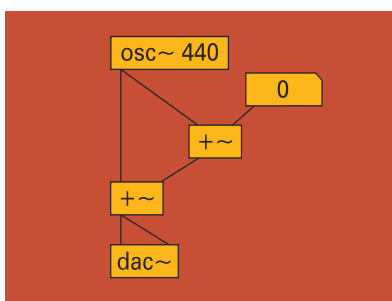
új érték beírásával, vagy kattintással és az egér húzásával végezhetjük el: felfelé mozogva növekszik az érték, lefelé pedig csökken. Valószínűleg észrevettük, hogy a bal oldali szám megváltozásakor a lenti eredmény azonnal megváltozik, a jobb oldali érték megváltoztatásakor azonban nem történik semmi. Vajon mi lehet ennek az oka? A Pd csaknem minden objektumára igaz, hogy a bal szélén lévő bemeneti csatlakozót forrón tartja, azaz ennek bármilyen változása azonnal megjelenik a kimeneten. A többi bemenet hideg, értékük megváltozása nem okoz változást a kimeneten. Az új érték mindaddig egyszerűen csak tárolódik, amíg a forró bemenet ki nem váltja a számítás elvégzését – ekkor kerül az új érték felhasználásra. Hogyan érhetnénk el, hogy a hideg bemenet változása is megváltoztassa a kimenetet? Az egyik módszer egy üzenetdoboz beszurása. A 7. ábrán a jobb oldali számdoboz kimenetét egy üzenetdoboz bemenetére kötöttem, majd ennek az üzenetdoboznak a kimenetét az alatta lévő összeadó objektumdoboz forró bemeneti csatlakozójára. Amikor egy üzenetdoboz bemenetére bármilyen üzenet érkezik, a benne tárolt tartalmat új üzenetként a kimenetére küldi ki. Így amikor a jobb oldali számdoboz tartalma megváltozik, az egy üzenetet küld a bang-üzenetdoboznak, amely egy bang-üzenetet (bang message) küld az összeadó objektumdobozra. A bang-üzenetek jelentése az, hogy „Csinálj valamit!”, vagyis ha egy objektumdoboz ilyen üzenetet kap a forró bemenetére, azonnal végrehajtja a kért számítást. Most ezt a viselkedést arra használjuk fel, hogy az összeadó objektumdobozunk olyan módon működjön, mintha két forró bemeneti csatlakozóval rendelkezne. De várjunk csak! Ez nem azt jelenti, hogy a bang-üzenetnek a szám után kell megérkeznie? Ha nem így történik, a folt nem fog működni, igaz? Nos, igen: attól függően, hogy a kapcsolatokat milyen sorrendben hoztuk létre, tapasztalhattuk, hogy nem működik. És csakugyan, a 7. ábrán látható számok összeadása e miatt a hiba miatt nem pontos, vagyis szükségünk van valamilyen módszerre, amellyel elérhetjük, hogy a bang-üzenet a szám után érkezzen meg. Ennek az az



9. ábra A jobb és bal oldal felcserélése egy sztereójelben



10. ábra A sztereójel monóvá alakítása



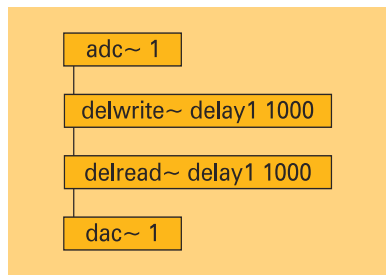
11. ábra Két jel kioltása inverz jel használatával

egyszerű módja, hogy egy késleltetést iktatunk be, mint az a 8. ábrán is látható. Érdekes módon a 0 késleltetés is működik. Az üzenet egy DSP-ciklusnyi késleltetést szenved, ilyen módon biztosított, hogy a bang-üzenet érkezik be másodikként.

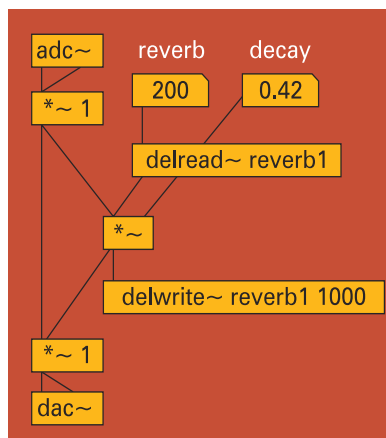
Hangfeldolgozás

A hangfeldolgozás legalapvetőbb függvényei a bemenet és a kimenet. Az `adc~` hullámvonal-objektum jelképezi az analóg–digitális átalakítót, amely az első feladatot hajtja végre, a `dac~` hullámvonal-objektum pedig a digitális–analóg átalakítót, ennek szerepe a kimenet előállítása. Alapértelmezésben mindkettő az első két csatornán működik. Ha ezt meg szeretnénk változtatni – például olyan többcsatornás hangkártyánk van, mint amilyen a Hasmmerfall HDSP –, értékként megadhatjuk a csatornák számait, ezáltal a megfelelő csatornák hozzárendelődnek megfelelő bemeneti és kimeneti csatlakozóikhoz. A 9. ábrán látható egy egyszerű példa, amelyben a sztereóbemenet csatornái

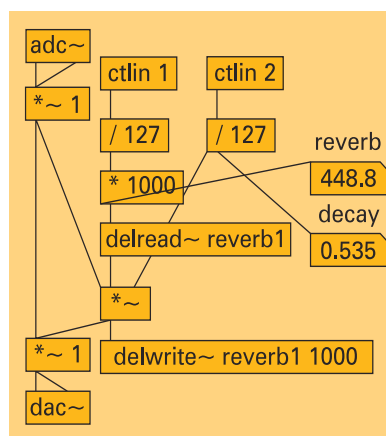
felcserélődnek és a kimenetre vannak irányítva. Mivel a sztereóbemenet az alapértelmezett, szükségtelen megadni a csatornák számozását, de szemléltetés céljából most feltüntettem. A hangok adatai egy bizonyos mintavételezéssel érkező számsorozatok, így ezeken alkalmazhatók a



12. ábra Egy késleltetési vonal



13. ábra Egy egyszerű visszhangszűrő



14. ábra

A MIDI használata egy folt vezérlésére

matematikai műveletjelek (operator). Például a 10. ábrán látható módon a sztereójelből monó hangot állítunk elő azáltal, hogy a jobb és bal csatornát

összeadjuk. Egy másik hasznos műveletjel a szorzás (`*~`), amit erősítőként alkalmazhatunk. Ne feledjük azonban, hogyha a hangot valamilyen eszközzel bocsátjuk, a jeleket levágja, amennyiben az értékük nem esik -1 és 1 közé.

A 11. ábrán egy összetettebb példát szemléltet. Érdekes lehet lehalkítani a berendezésünket, mert az eredmény elég hangos. A fenti `osc~` objektum egy szinuszel-generátor, ami ebben az esetben 440 Hz-en fut. Az előállított jel két részre válik: a bal fele közvetlenül az összeadóra jut, míg a jobb oldali rész először egy szorzáson megy keresztül.

Most próbáljuk meg a -1 értéket beírni a számdobozba – a hang elhallgat. Vajon mi lehet az oka? Ha még vissza tudunk emlékezni a középiskolában a hullámok fizikájáról tanultakra, eszünkbe juthat, hogy minden hullám kioltatható egy másikkal. Ebben az esetben a -1 az eredeti jel tökéletes inverzét állítja elő, így ahol az eredeti jel értéke 1, az inverzé -1. Ha ezt a két jelet összeadjuk, nullát kapunk, ami a csendnek felel meg. Kipróbálhatjuk azt is, hogy a SHIFT billentyűt lenyomva tartjuk és az egérkurzort a számdobozon húzzuk; a számok elég lassan változnak ahhoz, hogy halljuk elhallkultni és megszűnni a hangot, ahogy elérjük a -1-et.

Egy egyszerű visszhangszűrő létrehozása

Építsük össze ezeket az alkotóelemeket és próbáljunk meg kihozni belőlük valami használható dolgot: készítsünk egy egyszerű visszhangszűrőt (Reverb Filter) a fenti módszerek használatával, hozzávéve még egy újat, a késleltetővonal (Delay Line). Ha már van némi tapasztalatunk a hanghatások (sound effects) tervezésében, valószínűleg tudjuk, mi az a késleltetővonal, ha nincs, képzeljük egy hangok tárolására alkalmas átmeneti tárnak. A hang, amely a késleltetővonal bemeneti csatlakozójára kerül, egy meghatározott ideig tárolódik, mielőtt megjelenne a kimeneten. Az analóg korszak minden bonyolult módszerében fellelhető volt a késleltetés valamilyen megvalósítása, ezek némelyike különleges rugókat és szalaghurkokat használt.

Szerencsénkre mi már digitális jelekkel foglalkozunk, itt a késleltetővonal létrehozásához nincs szükség többre, mint egy FIFO tárolóra. A visszhanghatás egy olyan természetes jelenség, ami a hangnak a környezetében lévő felületekről történő visszaverődésének következménye. A nagyobb távolság miatt a visszaverődött hang egy kicsivel az eredeti hang után szólal meg. Ezek a visszaverődött hangok saját maguk is visszaverődnek. A legtöbb környezet rendelkezik valamilyen fokú visszhanghatással, sőt ennek a hatásnak egy pontosan szabályozott mértéke a koncerttermek igen meghatározó tulajdonsága. A nem túl erős akusztikával rendelkező termekben a visszhanghatás hiánya vagy csekély volta miatt az az érzésünk, mintha a hangot egy nem körülhatárolt térbe eregetnénk, mintha az nem verődne vissza. Ennek a szűrőnek a lelke a késleltetés. A 12. ábrán a hangbemenet 1 másodperces (1000 milliszekundum) késleltetését létrehozó késleltetővonalat látunk. A Pd-ben a késleltetővonalak névvel rendelkeznek, ami annyit jelent, hogy használatukhoz két különböző objektum szükséges: egy író és egy olvasó. Az író, a `delwrite~`, két értékkel bír: az első a késleltetővonal neve, a második pedig a használni kívánt legnagyobb késleltetés ideje ezredmásodpercekben megadva. Egy késleltetővonal csak egy íróval rendelkezhet. A másik objektum az olvasó, a `delread~`, amelynek szintén két értéke van: a késleltetővonal neve és a kívánt késleltetés mértéke. Az íróval ellentétben tetszőleges számú olvasónk lehet tetszőleges késleltetésekkel. Létezik az olvasó egy további típusa, a változó késleltetésű objektum (`vd~`), amely egy hangjel hatására változtatja a késleltetés mértékét, de ez már túlmutat cikkem keretein. Összegezve: a visszhangszűrőnek szüksége van egy közvetlen jeltömbre és egy vagy több késleltetett jelre, a visszaverődésekre. A visszaverődések rekurzívak, tehát a rendszerben valamilyen visszacsatolásnak is lennie kell. Végül, bár a bemeneteink és kimeneteink sztereójelet kezelnek, helytakarékoságból most csak egy egyszerű monó visszhangszűrőt valószínűsítünk meg. Ehhez össze kell keverni a bejövő jobb és bal csatornát, majd a

másik felén kettéosztani. A 13. ábrán láthatunk egy ilyen szűrőt. Ha a bemenettől követjük a jel útját, az első dolog, amit észre kell vennünk, a `*~ 1` objektum. Ennek az a célja, hogy egy kicsit áttekinthetőbbé tegye a foltot. Ez a jelet eggyel szorozza, ami önmagában felesleges, viszont bemenetet ad mind a jobb, mind a bal csatorna számára, és egyetlen csatornává keveri össze a jeleket. A folt alján még egy hasonló elrendezést láthatunk. A visszhang jele ezután egy újabb szorzón megy keresztül, amely csökkenti a visszaverődés szintjét, mielőtt egy újabb visszhangciklusba kerülne. A `delwrite~` és a `delread~` képezik a visszaverődési ciklust. Végül a szorzó kimenete a `dac~`-ra kerül vissza.

MIDI-vezérlés hozzáadása

Először is győződjünk meg arról, hogy a használni kívánt MIDI eszköz különösen megfelelően működik, ha az Evolution UC-16-hoz hasonló vezérlőeszközzel van szó. Egy egyszerű módszer annak ellenőrzésére, hogy a Pd fogadja-e a MIDI-ről érkező jeleket: válasszuk ki a bármelyik ablak jobb felső sarkában megtalálható **Help** (súgó) menü **Test Audio and MIDI** (a hang és MIDI kipróbálása) menüpontját. Nyissuk meg, és állítsunk elő néhány MIDI-eseményt. Ha a vezérlőegységünk működik és vezérlőjeleket ad ki, akkor – ellentétben a megjegyzéseseményekkel – az ablak alján látható **ctlin** (control in, vezérlőjelbemenet) objektumok alatti számoknak változniuk kell. Előfordulhat, hogy ehhez játszaniuk kell egy kicsit a Pd `-midi` parancssori kapcsolójával, és ne felejtjük elolvasni a `-verbose` kapcsolóval indított Pd elindulásakor megjelenő hibaüzeneteket. Ha mindez működik, próbáljuk ki a 14. ábrán látható foltot, amely a 13. ábrán látható visszhangra hasonlít, azzal a különbséggel, hogy a visszaverődést és a késleltetést most a MIDI és nem a GUI vezérli. Ez a váltás a **ctlin** objektum segítségével valósítható meg, amelynek ebben a példában egyetlen értéke van: a használni kívánt vezérlő száma. Kimenete a vezérlőről érkező szerkezet nélküli érték. Ha UC-16-ot vagy ahhoz hasonló vezérlőt használunk, ez az érték 0 és 127 közé eshet. A többi objektumnak egy kis matematikai

átalakítást kell végeznie ezen az értéken ahhoz, hogy a kívánt tartományba essen. Visszhang esetén ez a tartomány 0-tól 1000-ig terjed, ezért először 127-tel osztunk, hogy 0 és 1 közé eső számot kapjunk, majd ezt szorozzuk 1000-rel, hogy a kívánt értékhez jussunk. A csillapítás számára a 0 és 1 közé eső érték megfelelő, ezért már csak osztanunk kell. Jó gyakorlat jelent vizuális visszajelzést adni a hatás beállításának értékeiről, ezért ez a folt mind a visszaverődési, mind pedig a csillapítási értékekről a megfelelően jelölt számdobozokba küld egy-egy másolatot. Meg is volnánk: van egy teljesen számítógépesített, MIDI-vel vezérelt hanghatásunk, amelyet csupán néhány Pd-objektum összekapcsolásával hoztunk létre. És ez csak a kezdet a lehetőségeknek. Van egy MIDI billentyűzetünk? Használhatjuk vezérlőként és írhatunk egy olyan foltot, amelyik meg is szólaltatja a hangokat; vagy egy olyan foltot is létrehozhatunk, amely a MIDI billentyűzetet vezérli. Akár teljes, újraprogramozható hanghatásdobozt (effects box) is létrehozhatunk. Nem kell megelégednünk a már kész Pd-objektumokkal sem, írhatunk saját objektumokat C nyelven vagy akár magában a Pd-ben. Ha a Linux és a Pd a társunk, akkor az egyedüli korlát a saját tudásunk és a processzorunk sebessége.

Linux Journal 2003, 116. szám



Peter Todd 14 éves kora óta használja a Linuxot. Másodállásban egy kis Linux alapú hangstúdió vezető technikus. Amikor éppen nem dolgozik, akkor a Wexford Főiskolát látogatja, ahol jelenleg kerámia- és tervezőgrafika szakon tanul.

KAPCSOLÓDÓ CÍMEK

A cikk minden példája elérhető a Linux Journal FTP-oldalán az <ftp://ftp.ssc.com/pub/lj/listings/issue116/7062.tgz> címen. A reverb, delay rejtelmeiről a <http://www.emilrulez.hu/delay.html> oldalon olvashatunk.



DVD-átkódolás Linux-metaszámítással

A Condor mint nagy teljesítményű DVD-átkódoló rendszer Linuxhoz.

A videó- és hangkódolás terén napjainkban bekövetkezett fejlődésnek köszönhetően az MPEG-2 formátum vált a digitális videóközvetítések (digital video broadcasting, DVB) és a DVD tárolóeszköz alapjává. Eszközök széles palettája támogatja ezt a formátumot. Az MPEG-2-mozifájlok mérete általában 3–6 GB körül mozog, ami megfelelő a DVD-k esetében, de túl sok a CD-R hordozókhoz. Hasonlóképpen a jó minőségű MPEG-2-videók jól működnek a DVB-S vagy DVB-T hálózatokon, nemigen használhatók azonban az IEEE 802.11b vagy a házi HomePlug-átvitelben. Az ilyen feladatok megoldására kezdtek el kidolgozni a továbbfejlesztett tömörítési módszereket, amelynek eredményeképpen végül az MPEG-4 szabványossá vált. Az MPEG-4 formátum képes 700 MB alá csökkenteni a filmek méretét, miközben viszonylag elfogadható képminőséget ad. Mivel igen sok multimédiás tartalom DVD-hordozón, MPEG-2-állományokban érhető csak el, át kell őket kódolnunk, hogy MPEG-4 megfelelőikhez hozzájuthassunk. Ebben a cikkben a linuxos keretrendszeren alapuló, nagy teljesítményű DVD-átkódolásra képes Condor metaszámítási (metacomputing) rendszert mutatjuk be. Bár kötött számú géphez létezik néhány hálózati párhuzamos átkódoló eszköz, minket most nem érdekelnek a párhuzamos átkódoláson alapuló metaszámítási rendszerek. A metaszámítás olyan szerkezetet jelent, ahol a fizikai kinézetet elrejtjük és helyette egy egyszerűsített virtuális gépet jelentünk meg. A példa kedvéért: az általunk használt Condor eszköz az elérhető gépek idejéből „lop el” egy szeletet, amikor azt sem felhasználó, sem elsőbbséget élvező folyamat nem használja.

Háttér

Az olcsó DVD-lejátszók megjelenésének köszönhetően virágzott a DVD-filmpiac, és ebben nem kis szerepe volt annak, hogy a DVD tárolóeszközként sokkal megbízhatóbb, mint a VHS-kazetták. A DVD-rögzítési piac ugyanakkor még gyerekcipőben jár. Mivel a CD-R technológia már jó ideje létezik, és a CD-R lemezek sokkal olcsóbbak, mint a DVD-lemezek, az otthoni felhasználók kidolgozták, hogy miképpen lehet a DVD-filmeket CD-ken tárolni, megközelítőleg azonos minőségben. Az ilyesfajta megoldás a mostanában megjelent kódolási módszerek megjelenésével vált lehetővé; ezek az MPEG-4 szabványon alapulnak, és nagyon jó tömörítési aránnyal dolgoznak. A DVD-lemezek

1. táblázat A tesztgépek

név	processzor	MHz	memória	Kflops	Míps
gigabájt	Intel Pentium 4	1700	256 DDR	528 205	1388
kilobbájt	Intel Pentium 4	1700	256 DDR	624 242	1355
nazgul	Intel Celeron	433	192	152 593	491
titan	Intel Pentium II	350	320	67 987	398
brio	Intel Pentium II	350	192	72 281	398

tartalmának CD méretűre kódolása azonban a legtöbb asztali PC számára még mindig túlságosan számításigényes. A párhuzamosítás ígéretes megoldásnak tűnik a DVD-átkódolás felgyorsításához. A legnyilvánvalóbb megközelítés a kézi párhuzamosítás, azaz amikor kézzel daraboljuk az állományt, majd különböző gépeken átkódoljuk a darabokat, végül az eredményt újraegyesítjük. A kézi párhuzamosítás azoknak lehet megfelelő, akik a teljes folyamatot nyomon szeretnék követni. Ugyanakkor, ha nagy teljesítményű, „küld el és felejtst el” DVD-átkódolást szeretnénk megvalósítani, érdemes kihasználni a metaszámítás előnyeit. A folyamatok párhuzamosításához a folyamatot elemi feladatokra kell bontani, majd ütemezni őket, végül begyűjteni az eredményt. Ennek megfelelően szükségünk lesz egy erőforrás-kezelő eszközre. A legalapvetőbb metaszámítási és párhuzamosítási szolgáltatásokat megtaláljuk a Condor és a Globus programokban. Mi a Condort választottuk, mivel nem bonyolítja a dolgokat, könnyen telepíthető és beállítható, valamint kiválóan működik Linux alatt és nem kell hozzá külön fűrtöt építeni. A Condor számításigényes feladatokra kihegyezett munkaterhelés-kezelő rendszer. Akárcsak más teljes értékű parancsfájlrendszerekben, a Condorban is megtaláljuk a munkafolyamat-rendezési lehetőséget, az időzítési szabályokat, az elsőbbségi sémákat és az erőforrás-ellenőrzést. A felhasználók elküldik soros vagy párhuzamos munkáikat a Condornak, az pedig elhelyezi őket a sorban. A rendszabály alapján eldönti, hogy mikor és hol kell lefuttatni a kódot, figyeli a fejlődésüket, végül természetesen értesíti a felhasználót a munkafolyamat befejezéséről. Miközben a hagyományos munkafolyamat-kezelő rendsze-

2. táblázat Számítási eredmények
(t = idő, fps = képkocka/másodperc)

elosztás	gép	videó- kódolás (t)	film- összefűzés (t)	összes idő	fps
apró darabok	g ----	1:51:14	0: 01:32	2:03:52	19,6
	gk---	0:58:25	0: 02:57	1:09:00	35,1
	gk-b	0:53:53	0: 03: 32	1:04:31	37,6
	g-ntb	1:13:04	0: 03:54	1:24:55	28,6
	gkntb	0:47:33	0: 04: 07	0:58:55	41,2
mester- munkás	g ----	1:49:11	0: 01:31	1:59:23	20,3
	gk---	0:56:23	0: 03: 03	1:06:10	36,7
	gk-b	0:51:06	0: 02:37	1:00:06	40,3
	g-ntb	1:08:00	0: 03: 00	1:17:38	31,2
	gkntb	0:42:55	0: 03: 58	0:52:35	46,1

3. táblázat Összehasonlító eredmények,
State-of-the-Art átkódoló alkalmazások

	összes idő	fps	AVI méret (MB)
prototípus, mester-munkás (gkntb)	0:52:35	46,1	739,2
Prototípus, apró-darabok (gkntb)	0:58:55	41,2	742,8
prototípus (egy processzor, g)	1:59:23	20,3	739,1
Mencoder 0.90 Linux (g)	2:01:02	20,0	731,1
FlaskMpeg 0.78.39 Windows (g)	1:53:12	21,4	724,1

rek képességeihez hasonló tulajdonságokkal is rendelkezik, a Condor különleges felépítése miatt olyan helyzetekben is sikeres lehet, amelyekben a hagyományos rendszerek kudarcot vallanak. A Condor egyedi szerkezete lehetővé teszi, hogy befogja az egyébként unatkozó asztali gépek szabad processzoridejét. Például beállíthatjuk a Condort úgy, hogy csak akkor használja az asztali gépeket, amikor az egér és a billentyűzet használaton kívül áll. A Condor érzékeli, ha egy gép többé már nem érhető el (például billentyűütemést tapasztal), és létrehoz egy átlátszó ellenőrzőpontot, majd a feladatot áttelepíti egy másik, egyébként üzemben kívüli gépre. A Condor képes valamennyi I/O kérelmet átlátszó módon visszairányítani az elküldő gépre, ennek eredményeképpen bármely közösség zökkenőmentesen képes egyesíteni számítási erejét. A metaszámításra képes üzleti átkódoló alkalmazások jelenlegi hiánya valószínűleg azzal magyarázható, hogy a metaszámítás erősen kötődik a Unix tudományos közösségéhez.

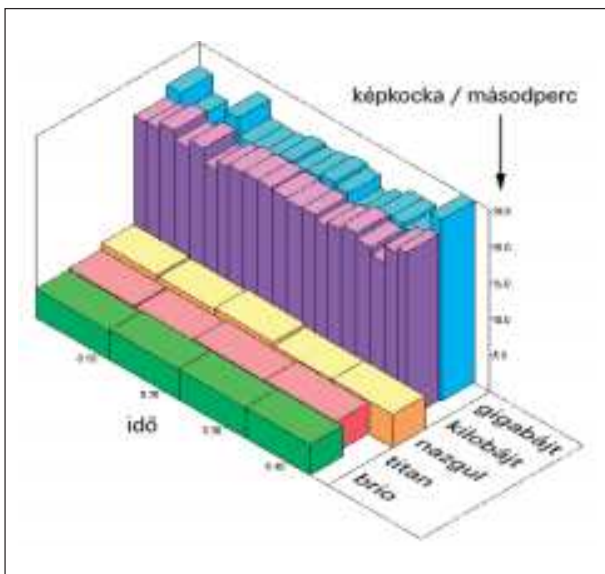
A szórakoztatóprogram-tervezők viszont mindig is a metaszámítás-ellenes windowsos világnak szentelték teljes figyelmüket. Például a linuxos átkódolófejlesztések szempontjából kulcsfontosságú a legfrissebb DivX kodek (a cikk születésének idején ez a v.5.0.5-ös), ami még mindig nem fut hibátlanul a Pentium 4-es linuxos gépeken. Az előző kiadás a v.5.0.1alpha volt, ezt a megbízhatatlanra sikerült változatot az előző évben adták ki. Ez a példa jól mutatja, hogy milyen gondokkal kell megküzdeni, amikor metaszámításokat támogató linuxos rendszereken szórakoztató alkalmazásokat próbálunk megvalósítani. Bár rengeteg átkódoló alkalmazás létezik, itt csak szerintünk legérdekesebb hármat mutatjuk be:

- **FlaskMpeg:** ez az egyik első átkódoló rendszer – jelenleg az egyik legnépszerűbb ilyen rendszer a windowsos világban. Nem támogatja a párhuzamosítást.
- **Mencoder:** a Linux-alkalmazások élvonalába tartozó DVD-átkódoló program. Hatékonyasága (kimenet a bemenethez aránya) általánosságban valamivel alulmarad a FlaskMpeg teljesítményéhez képest. Akárcsak az előző program, ez sem támogatja a párhuzamosítást.
- **Dvd::rip:** magas színvonalú linuxos átkódoló program, amely egy másik programon, a Transcode-on alapul. Eredménye nagyjából a Mencoderéhez mérhető (csak sokkal lassabb – a ford.). A Dvd::rip támogatja a párhuzamosítást – ennek a beállítása azonban nem túl egyszerű, mivel az átkódolásban résztvevő valamennyi számítógépen külön be kell állítanunk. A beállítások statikusak és nem válaszolnak a környezet változásaira (ez fontos különbség a mi Condor-központú rendszerünkhöz képest). A Dvd::rip nem engedélyezi a hang-sávokat, amelyeket műszaki okok miatt kénytelenek vagyunk sorosan kódolni, amire a Dvd::rip fel is hívja a figyelmünket, bár meg nem oldja (erről bővebben a Dvd::rip honlapján olvashatunk). Mindazonáltal ez csak kisebbfajta gond, hiszen az átkódolás összeidejében alapvetően a videójel tömörítési ideje a mérvadó, függetlenül attól, hogy a hangátkódolás párhuzamosan vagy sorosan történik-e.

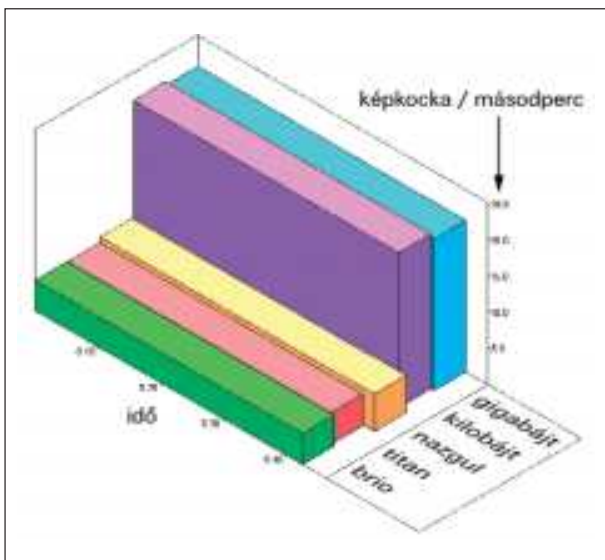
DVD-felosztás

A DVD az ISO/IEC 11172 (MPEG-1) és ISO/IEC 13818 (MPEG-2) szabványok alkészletén alapul. A DVD-film három részre osztható: a videoobjektum- (VOB) állományokra, amelyek legnagyobb mérete darabonként legfeljebb 1 GB lehet, a nyalábolásos (multiplexing) video- és hangállományforrásokra.

Háromfajta MPEG-2-filmkocka (frame) létezik: I (Intra, belső), P (Predictive, jósló) és a B (Bidirectionally-predictive, kétirányban jósló). Az I képkockák teljes képkockákat jelölnek, a P és B filmkockák csak az előző, illetve következő képkocka közti különbségeket tárolják. Nyilvánvalóan tűnik, hogy a videójel vágását valamelyik I filmkocka elején kell elvégezni. Ez majdnem pontosan így is van, de azért nem egészen. Figyelembe kell venni néhány egyéb jellemzőt is, például a képsűrűséget (framerate) és a méretet. Ezek az adatok a sorfejléc (Sequence Header) részét képezik – ezért aztán olyan csomagot kell kiválasztani vágási pontnak, amelyik tartalmaz sorfejléct. Szerencsére minden



1. ábra Az egyes gépek átviteli sebessége, apró darabok módszere



2. ábra Az egyes gépek átvitele sebessége, mester–munkás módszer

egy I képkocka előtt van ilyen sorfejléc. Egy másik fontos kérdés a P és B filmkockák miatt szükséges képkocka-újrarendezés. Minden I filmkocka után következhetnek B képkockák, amelyek P képkockáktól függenek, ezek viszont az I képkocka előtt jönnek. Amennyiben a képanyagot éppen ez előtt az I képkocka előtt vágjuk el, a képtárolás egységességét (konzisztenciáját) nem tudjuk fenntartani. Az egyik megoldás szerint a B képkockákat az előző adagba helyezzük át, ennek eredményképpen a kép előfeldolgozása valamivel bonyolultabb lesz. Nyilvánvaló, hogy nem érdemes minden arra alkalmas helyen szétszedni a filmet, hiszen így darabkák mérete nagyon kicsi lenne. Általában körülbelül 300 KB adat található két egymást követő I képkocka közt, bár ez a hossz erősen függ a képsűrűségtől, a képmérettől és egyéb jellemzőktől.

Terheléselosztás

Projektünkben két alapvető terheléselosztó módszert vizsgáltunk. Az első a Small-Chunks nevű módszer, amelynek során a DVD-filmet adott méretű apró darabokra vágjuk fel. A Condor minden elérhető számítógépnek kioszt egy darabot. Amikor a gép befejezte a darab átkódolását, kér egy másikat – ez a folyamat addig ismétlődik, amíg a kiszolgálón még maradt feldolgozatlan darab. A másik mester–munkás nevű megközelítésben a terheléselosztás a mesterprocesszor által kiadott megosztásokon alapul – a többi résztvevő számítógép értelemszerűen munkás lesz. Ezt a stratégiát gyakran alkalmazzák nagy teljesítményű feldolgozásoknál. Ebben a projektben az egyes gépekhez tartozó darabok méretét a tanulószakasz alapján osztjuk ki, mint azt a következő részben részletesen is ismertetjük. Hangsúlyozzuk, hogy szándékosan nem foglalkozunk a felhasználói felület vagy a gépek hibalehetőségeivel. Amennyiben ezeket is számításba vesszük, a projektünkben használt egyszerű mester–munkás (master–slave) megoldás teljesítménye nagymértékben csökkenne. Az általunk ismertetett két megoldás bemutatócélokat szolgál, hiszen különleges esetek, egyrészt tiszta mester–munkás kiépítésüknek köszönhetően, másrészt az apró darabok feldolgozása miatt. Kíváncsiak voltunk, hogy az alkalmazás a tömörítési idő és a tömörített fájl méret tekintetében mindkét véglet esetében hasonlóképpen viselkedik-e. A cikkünkben közölt eredmények azt sugallják, hogy a kis darabokon alapuló megoldás egyszerűsége miatt előnyösebb lehet (nincs szüksége tanítószakaszra), illetve természetesebben illeszkedik a Condor gépelérhetőség-kezelő rendszeréhez.

Mester–munkás tanulószakasz

A mester–munkás rendszerező tagja számára biztosítanunk kell a szükséges adatokat; ehhez az összes gépet előre ki kell értékelnünk. A kiértékelést a tanítószakasz folyamán véghezvük, ahol az egyes gépek képkocka/másodperc teljesítményértékét becsüljük meg. Prototípusunkban ez a tanítószakasz különféle apró videósorozatokat átkódolásával történik, ennek alapján becsüljük meg az egyes gépek képkocka/másodperc értékét. Ezeket az eredményeket használjuk fel az adatméretek kiszámításához, amely a gép becsült teljesítményével arányos. Eseményi esetben ezzel a megközelítéssel csökkenthetjük a DVD-átkódoláshoz használt időt, hiszen valamennyi számítógép egy időben fejezi be a munkát.

A próbarendszer kiosztása

Tesztrendszerünkben egy átlagos heterogén számítási környezetet szimuláltunk, olyan gépeket is felhasználva, amelyek felhasználhatósági élettartamuk végét járták. Öt számítógépet kapcsolunk össze (lásd az 1. táblázatot), amelyeket számítási teljesítményük alapján három csoportba osztottunk. Az első csoportba két gép tartozott („gigabajt” és „kilobajt”), a második csoportba egyetlen gép került („nazgul”), végül két gép alkotta a harmadik, legkisebb teljesítményű csoportot („titan” és „brio”). Valamennyi gép 100 Mb/s átvitelű ethernethálózatra csatlakozott és Red Hat Linux 8.0-s operációs rendszert használt. Minden gép NIS-kiszolgálón keresztül megvalósított azonos felhasználói téren és azonos fájlrendszeren („gigabajt” NFS-

kiszolgálóján) osztozott. Végül a Condor 6.4.7-es változatát telepítettük, amelynek központi vezérlője „gigabájt” lett. A Condort úgy állítottuk be, hogy a felhasználói tevékenységtől függetlenül tartsa meg a feladatait a megfelelő processzoron. Így a kipróbálás során bemutatott eredmények, mint fentebb említettük, a lehető legjobb esetet mutatják. A DVD-DivX párhuzamos átkódoló programot a következő könyvtárak támogatták:

- libmpeg2 0.3.1: DVD MPEG-2-folyam-demultiplexing és -visszakódolás.
- liba52 0.7.5-cvs: DVD AC3-audio-visszakódolás.
- DivX 5.0.1alpha: MPEG-4-videókódolás.
- lame 3.93.1: MP3-audiókódolás.

Párhuzamos videóátkódolás

Miután a videó darabolásával elkészültünk, a videóadathalmokat átadjuk a Condor átkódoló munkafolyamatainak. Ezek a folyamatok a Condor Vanilla univerzumában futnak, mivel a DivX könyvtárát dinamikusan töltik be. Az átkódolásához szükséges adathalmazokat minden átkódoló közvetlenül a forrás-VOB-ból olvassa. A kimenő adatokat ugyanabba a könyvtárba helyezik. Az olvasási, illetve írási műveletek képtől képig alapon működnek: a kódoló beolvas egy képet, átkódolja azt, majd kiírja az eredményt. Ezzel a stratégiával jobb eredményeket érhetünk el, mintha egész adatsoportokat (chunks) küldenénk a munkásokhoz és azoknak helyi fájlrendszerén dolgoznánk fel, majd a teljes átkódolt adathalmazrészeket a kiszolgálógépre visszaküldve összeállítanánk. Valamennyi gép NFS-t használt a bemenő VOB-állományok és a kimeneti könyvtár eléréséhez. A párhuzamos átkódolási szakasz után az átkódolt eredmény több független fájlba kerül, amelyeket a mestergépen egymáshoz fűzve kapjuk meg a kész DivX-filmet. Az eredményeket a 2. táblázat mutatja be. Kétfajta terhelésselosztási módszert alkalmaztunk, a kis darabokra vágást és a mester–munkás megközelítést. A próbához felhasznált mozi, a „Mindent anyámról”, körülbelül 1 óra és 37 perc hosszú, eredeti mérete 2,94 GB. A 2. táblázat „gép” oszlopában rövidítés a próbagépek nevének első betűje, például a gigabyte-ra g-vel a titánra t-vel hivatkozunk. A - (mínuszjel) azt jelenti, hogy az adott gépet nem használtuk abban a tesztben. A kis darabokon alapuló megoldásban használt darabméret 60 MB volt. A videó-előfeldolgozásra fordított időt nem adtuk meg, mivel minden esetben elhanyagolható mennyiség volt. A 2. tábla alapján levonhatunk néhány következtetést. Először is az fps oszlop alapján a terhelésselosztás jobb lesz, mind a mester–munkás megközelítés, mind a kis darabokra vágás esetében. A különbség kicsi ugyan, de a felhasznált gépek számának növekedésével párhuzamosan növekszik. Általában véve a párhuzamosítás növeli az átkódolás teljesítményét, ami azonnal nyilvánvalóvá válik, amint egy újabb hatékony gépet helyezünk üzembe (lásd a [g----] és a [gk---] közötti különbséget). Az alacsony teljesítményű gépek beillesztésével csekély növekedés érhető el (lásd a [gk---] vagy [gk--b] és a [gk--b] vagy [gkntb] eseteket). Mindazonáltal az alacsony-kategóriás gépek együttes teljesítménye már figyelemre méltó, különösen, ha egyetlen erős géphez viszonyítva vizsgáljuk a növekedést (lásd a [g----] és a [g-ntb] különbségét).

A prototípus további tulajdonságainak vizsgálatához két népszerű kódoló eszközt, a Mencoder és a FlaskMpeget hasonlítottuk össze. A 3. táblázatban ezeket az eredményeket foglaltuk össze. Prototípusunk egyprocesszoros változatának eredménye valahol a FlaskMpeg és Mencoder teljesítménye között helyezkedik el. Az eredményfájl méret tekintetében prototípusunk a legrosszabb esetben (apró darabok módszere) mindössze 2,6 százalékkal nagyobb DivX-mozit készített, mint a FlaskMpeg kimenete. A apró darabok módszerével kapott valós tömörítési arány (24,67%) és a FlaskMpeg teljesítménye (24,05%) közel azonos – ha a feldolgozás sebességét figyelembe vesszük, a különbség nem jelentős. Érdeemes megjegyezni, hogy a FlaskMpeg a DivX v. 5.0.5 Pro kodeket használja, ami Linux alatt a cikk születése idején még nem volt elérhető. Ezért aztán amint a linuxos változat megjelenik, a tömörítési teljesítmény különbsége tovább csökkenhet. Végül az 1. és a 2. kép a prototípusrendszerek gépeinek terhelésselosztási teljesítményét mutatja be az apró darabok és a mester–munkás megoldás esetén. A számítógépek nem pontosan egy időben fejezik be a munkát; ez várható is, hiszen a kis darabok terhelésselosztása csak megközelítőleges. Továbbá a mester–munkás megoldásban a munkadarabok méretét a tanulmány alapján határozzuk meg, ami jellemző, de nem teljesen pontos.

Összegzés

Cikkünkben a Linux-alapokra épülő, nagy teljesítményű DVD-átkódoló rendszert, a Condort mutattuk be. Eredményeink alapján elmondható, hogy a metaszámítás alapú párhuzamos átkódolás érdeklődésre tarthat számot, és figyelemreméltó növekedést tapasztalhatunk a ma létező egyprocesszoros Windows alapú eszközökhöz képest. Saját eredményeink szerint a mester–munkás módszer jobb eredményt ad, mint a kis daraboké, de a különbség nem jelentős, gyakorlatilag jelentéktelennek tekinthető.

Linux Journal 2003. december, 116. szám

Francisco J. Gonzalez-Castano

A spanyolországi Vigo egyetem Departamento de Ingenieria Telematica részlegén a TC-1 Information Technology csoport vezetője.

Rafael Asorey Cacheda

Kutató a vigói egyetem TC-1 Information Technology csoportjánál.

Rafael P. Martinez-Alvarez

Telekommunikációs mérnökként dolgozik a spanyolországi vigói egyetem TC-1 Information Technology csoportjánál.

Eduardo Comesana-Seijo

1976-ban született a spanyolországi Vigóban. Az itteni egyetem TC-1 Information Technology csoportjának a kutatója volt. Jelenleg a Comunitel Global SA (a spanyol Telco) munkatársa.

Javier Vales-Alonso

A spanyolországi Cartagena főiskola információs technológia és kommunikáció karán tanít.

Az Amd önbefűző használata

Vajon az Amd önbefűző (automounter) segítségével hogyan biztosíthatunk egységes, könnyen felügyelhető, bárhol elérhető hozzáférést az összes fájlkiszolgálónkhoz?

A rendszergazdák manapság nagyméretű, számos NFS-kiszolgálót és -ügyfelet magukban foglaló telephelyet felügyelnek. A felhasználók általában igénylik, hogy bármelyik állomásra be tudjanak jelentkezni, s közben akármelyik távoli kiszolgálóról el tudják érni ugyanazokat a fájlokat. E hozzáférés biztosítására kézenfekvő, ám barbár megoldás az összes fájlkiszolgálónak kézi befűzése az összes ügyfélgépen. Természetesen egy ilyen rendszert lehetetlen felügyelni és bármelyik kiszolgáló elérhetetlenné válása az összes ügyfélgép leállását okozhatja. Emellett a felhasználóknak tudniuk kellene, hogy az egyes kiszolgálókról milyen név használatával érhetik el az állományokat.

Mindezekon a gondokon túlléphetünk, ha önbefűző használata mellett döntünk. Az önbefűző beállításakor a fájlkiszolgálók összes adatát megadjuk, így a rendszergazdának csak az önbefűző beállításfájlját kell karbantartania, azt is csak egyetlen helyen. Az önbefűző arra is képes, hogy a távoli fájlkiszolgálók elérését igény szerint biztosítsa, vagyis a befűzés akkor történik meg, amikor a felhasználó először próbálja meg használni az adott kiszolgálóra mutató elérési utat. Így az ügyfélgépeken mindig kizárólag a használatban lévő és elérhető kiszolgálók vannak befüzve, s ezáltal a leállások valószínűsége is csökken. Végül az önbefűző egységes névteret kínál az erőforrások számára. A `/src/kernel` elérési út például sokkal egyszerűbben kezelhető, mint a `kiszolgáló1:/n/raid/1/ksrc/v2.4/21preX`, illetve alkalmas a valódi elérési út elrejtésére is.

A legtöbb kereskedelmi Linux-terjesztés tartalmaz önbefűzőt. Ezek a programok azonban csak egy-egy rendszeren működnek, egyedi beállításokat használnak vagy csak korlátozott szolgáltatáskészletet adnak. Az Amd önbefűző, más néven Berkeley önbefűző fejlesztésekor a hordozhatóság és a gazdag szolgáltatásválaszték megteremtése volt a cél. Számos Unix-rendszeren módosítások nélkül futtatható, a többi megoldás szolgáltatásainak gyakorlatilag mindegyikét képes nyújtani, illetve az általa kínált számos lehetőség miatt még a legkülönlegesebb elvárásokat támaztó rendszergazda igényeinek is képes megfelelni. Ha különféle Unix-rendszereket kell felügyelnünk és szeretnénk megkönnyíteni az életünket, akkor érdemes az Amd mellett döntenünk. A továbbiakban az Amd működését fogom ismertetni,

valamint példákon keresztül mutatom be a képességeit. Elsősorban vállalati rendszergazdák számára szeretnék segítséget nyújtani, de remélem, a fájlrendszerek módosításában élvezetet leelő más érdeklődők is örömmel fogják olvasni soraimat.

Előfeltétel a Linux, az NFS és a fájlrendszerek legalább alapvető ismerete. Az Amd önbefűző képességeit néhány oldalon természetesen nem lehet ismertetni, ehhez egy egész könyvre lenne szükség. Azt azonban célul tűzhetem ki, hogy általános gyakorlati helyzetekre utalva először egyszerűbb, majd egyre bonyolultabb példákkal keltsem fel az érintettek érdeklődését.

Az Amd működése

Az Amd egy felhasználói szinten futó démon, amely egy önbefűzési pontnak nevezett könyvtárhoz csatlakoztatja magát. Amikor a felhasználók ezt az önbefűzési könyvtárat megpróbálják elérni, az Amd a rendszermagtól megkapja az ezirányú kéréseket. Az Amd biztosítja a felhasználó által igényelt erőforrás elérhetőségét, valamint válaszol a rendszermag kérésére. A rendszermag ezt követően a megfelelő állapotkódot adja vissza a felhasználónak, aki – micsoda varázslat – máris látja a kért elérési úton található fájlokat. Lássunk egy példát! Tegyük fel, hogy az Amd elindult és csatlakozott a `/src` könyvtárhoz. A felhasználó kiadja az `ls -l /src/kernel` parancsot. A rendszermag tudja, hogy a `/src` könyvtárra vonatkozóan az `amd` démon az érvényes fájlkiszolgáló, így felfüggeszti a felhasználó `ls` folyamatát és üzenetet küld az Amd-nek, amelyben a `/src` önbefűzési pont alatt található `kernel` név feloldására kéri. Amikor az Amd elindul, minden önbefűzési ponthoz betölti az önbefűzési térképet (automounter map). Az önbefűzési térképek egyszerű állományokból, NIS/NIS+, LDAP, N/DBM és egyéb forrásokból olvashatók be. Az Amd a `/src` ponthoz kulcs-érték párok listáját olvassa be. A kulcs azt a nevet jelképezi, amelyet az Amd-nek az önbefűzőt könyvtáron belül kell átadnia, az érték pedig az Amd által a megnevezett kulcs elérésének biztosításához igényelt adatokat tartalmazza. A térkép egy eleme például a következő lehet:

```
kernel type:=nfs;rhost:=kiszugalol;
  ↪ rfs:=/n/raid/1/ksrc/v2.4/21prex
```

Példánkban a `kernel` a kulcs – a hozzá tartozó értéktől szóköz választja el. Az érték három változó értékadásából áll, amelyeket pontosvessző különít el egymástól. A két sort a visszaperjelnek (backslash) köszönhetően a rendszer egyként kezeli. A változók értékének megadása – Pascal stílusban – a `:=` formulával történik. A fenti példában a `type` (típus) változó tartalma alapján megállapítható, hogy a térképelem NFS-befűzésre utasít. Az `rhost` változó a távoli NFS-kiszolgáló nevét tartalmazza, míg az `rfs` változó a távoli állomáson kívülről is elérhetővé tett (exportált) könyvtár elérési útját adja meg.

Térjünk vissza a felfüggesztett `ls` folyamathoz. Amikor az Amd a `kernel` névre vonatkozó keresési kérést kapja a rendszermagtól, megvizsgálja a térkép tartalmát, és megtalálja az ehhez a névhez tartozó elemet. Az Amd befűzi a távoli *kiszolgáló1* állomás `/n/raid1/l/ksrc/v2.4/21preX` könyvtárát, majd átadja az NFS-könyvtár típusáról, helyéről a rendszermagnak azokat az adatokat, amelyek alapján az folytatni tudja az `ls` parancs futtatását. Az `ls` folyamat befejezi a `/src/kernel` könyvtár tartalmának listázását, miközben a legcsekélyebb mértékben sem szerez tudomást az Amd és a rendszermag között lejártszódot párbeszédről, főleg nem a `/src/kernel` könyvtár fájljainak tényleges helyéről. Az Amd ellenőrzi azt is, hogy az egyes önbefűzött pontokat mikor használták utoljára, és a szükségteleneket önműködően leválasztja. Ezzel biztosítható, hogy a rendszer csak a valóban szükséges, gyakran használt elemeket fűzze be. Elébe mennék annak a kérdésnek, hogy vajon a leválasztáshoz szükséges időtartamot meg lehet-e változtatni: igen, meg lehet, sőt több tucatnyi egyéb beállításba is belenyúlhatunk. Most bizonyára mindenki azt gondolja, hogy az Amd tisztességes beállítása napokig tart, pedig nem így van – a legtöbb beállításnak az alapértéke is gondos finomhangolás eredményeképpen állt elő.

Az Amd indítási beállító fájlja

Az Amd beállításfájlja általában a `/etc/amd.conf`. A beállítások megadásának formája hasonló az `smb.conf` beállításfájlnál megismerthez; nézzünk meg egy példát:

```
[global]
log_file = /var/log/amd
debug_options = all,noreaddir

[/net]
map_type = file
map_name = /etc/amd.net
mount_type = nfs

[/home]
map_type = nis
map_name = amd.users
mount_type = autofs
```

A fenti `amd.conf` fájl elsőként az átfogó beállításokat adja meg, ezek minden önbefűzött könyvtárra érvényesek. Minden beállítás egy egyszerű kulcs=érték pár. Az első átfogó beállítás (`log_file`) egy naplófájl helyét és nevét adja meg, az Amd ide írja a hibaüzeneteket és a használati naplót. A második átfogó beállítás (`debug_options`) a

programot a könyvtárolvasási műveletek kivételével az összes lépés bőbeszédű naplózására bírja rá. A következő részekben két önbefűzött könyvtárat adtunk meg. Az egyik a `/net` könyvtár, ennek elemeit az Amd a `/etc/amd.net` fájlból olvassa be. A másik a `/home`, amelynek elemei a helyi NIS (YP) kiszolgálóról származnak.

A `mount_type` átadott értékhez némi magyarázatot szeretnék fűzni: az Amd alapesetben a rendszermag felé felhasználói szinten futó NFSv2/UDP-kiszolgálóként látszik. Amikor tehát a rendszermagnak tájékoztatnia kell az Amd-t arról, hogy egy felhasználó valamelyik elemet – például a `/src/kernel` könyvtárat – próbálja elérni, a rendszermag RPC-üzeneteket küld az Amd-nek, pontosan olyan módon kódolva az `NFS_LOOKUP` kérést, mintha egy távoli NFS-kiszolgálóval próbálná meg felvenni a kapcsolatot. Az egyetlen különbség ekkor az, hogy az Amd egy felhasználói szinten futó folyamat és nem rendszermagszintű NFS-kiszolgáló; valamint az Amd a helyi gépen fut, vagyis a rendszermagnak az NFS RPC-keket a 127.0.0.1 címre kell küldenie. Felhasználói szintű NFS-kiszolgálóként az Amd hordozható és minden Unix alapú gépen ugyanúgy működik. A felhasználói szintű NFS-kiszolgálók üzemeltetése ugyanakkor nagyszámú környezetváltást és rendszermagon belüli adatmozgatást okoz, ami a teljesítmény romlásához vezet. Tovább rontana a helyzetet, ha az Amd-folyamat váratlanul összeomlana (ami nem történhet meg, hiszen a kódunk száz százalékban hibamentes) – ekkor az adott állomáson önbefűzött könyvtárat elérő folyamatok mindegyikét magával rántaná, s ezután általában a rendszer újraindítása válna szükségessé.

Körülbelül egy évtizede a Sun Microsystems fejlesztői felismerték az önbefűzésnek ezeket a hátrányait, és készítették egy különleges, a rendszermagon belül futó, az önbefűzést segítő fájlrendszert: ez lett az AutoFS. Az AutoFS szinte minden olyan szolgáltatást biztosít, amelyre egy önbefűzőnek a rendszermagon belül – ahol a műveletek megbízhatóbban és gyorsabban elvégezhetők – szüksége lehet. Az AutoFS gyakran egy felhasználói szintű önbefűzővel működik együtt, amelynek a feladata bizonyos térképkeresési és -értelmezési műveletekre korlátozódik. Az Amd – mint a fenti `amd.conf` fájlból is kiderül – elég rugalmas ahhoz, hogy felhasználói szintű NFS-kiszolgálóként és AutoFS-megfelelő önbefűzőként egyszerre üzemeljen. Nekünk mindössze annyi a dolgunk, hogy a megfelelő értéket adjuk a `mount_type` beállításnak. De miért is nem használjuk csak az AutoFS-t? Az AutoFS sajnos nem minden operációs rendszeren érhető el, amelyeken pedig az (Linux, Solaris és számos egyéb), ott eltérő működésű és viselkedésű megvalósításaival találkozhatunk. Ez az oka annak, hogy nem minden rendszergazda kedveli az AutoFS-t. Szerencsére itt van az Amd, ami mindenkinek megadja a választás szabadságát.

A felhasználók kezdőkönyvtárai

A felhasználók kezdőkönyvtárai gyakorlatilag minden nagyobb rendszerben több fájlkiszolgálóra el vannak osztva. A felhasználók azonban nem nagyon szeretik azt, ha a kezdőkönyvtárunk például a `/u/munkatarsak/fajlok1/janos` elérési út alól egy új fájlkiszolgáló üzembe állítását és az adatok átmozgatását követően a `/u/ujfajlok3/janos` könyvtár

alá költözik. Sokkal szerencsésebb, ha minden kezdőkönyvtárhoz egységes névadási eljárást vezetünk be, így a `/home/janos` mindig az adott személy kezdőkönyvtárának a pillanatnyi helyére fog mutatni. A rendszergazda ekkor nyugodtan átmozgathatja a felhasználó kezdőkönyvtárát az új, nagyobb kapacitású fájlkiszolgálóra; mindössze a felhasználó `amd.users` térképében kell módosítania a `janos` elemet. Nézzünk egy példát egy egyszerű `amd.users` térképre – ez három különböző kiszolgálóról három felhasználói kezdőkönyvtár befűzésére utasítja a rendszert:

```
#megjegyzés: amd.home térkép
/defaults type:=nfs
janos rhost:=kiszalgalol;rfs:=munkalemez/janos
istvan rhost:=raid3;rfs:=ujlemez/istvan
daniel rhost:=raid3;rfs:=ujlemez/daniel
```

Példánk egy különleges `/defaults` bejegyzéssel kezdődik, amely egy, a térkép elemeire egységesen érvényes értéket ad meg. Ebben az esetben a térkép minden befűzése NFS típusú lesz. A következő három sor a felhasználó nevét, a távoli állomást és a felhasználó kezdőkönyvtárának az elérhetővé tételéhez befűzendő lemezzel nevével tartalmazza. Az egyes felhasználók kezdőkönyvtárának elérési útja – például `/home/istvan` – hosszú időn keresztül azonos maradhat, mégis megoldható, hogy például István kezdőkönyvtárának a valódi helye – természetesen az ő munkájának megzavarása nélkül – akár gyakrabban is megváltozzon.

Miként a Perl, úgy az Amd is számos módszert kínál ugyanannak a célnak az elérésére, és időnként valamelyik jobb a többinél. A fenti térkép több okból sem a legjobb választás, lássunk tehát néhány tanácsot az Amd-térképek jobbá tételére! Először tekintsük át, hogy mi történik, amikor a `/home/daniel` könyvtár próbáljuk meg elérni a raid3 gépről: az Amd az NFS-kiszolgálóként üzemelő raid3 gépről NFS-ügyfélként megkísérli befűzni a raid3-at. Végigbukfencezni a teljes hálózati protokollkészletet, felvállalni az NFS protokoll használata miatti többletterhelést, csak azért, hogy elérjünk egy helyi könyvtárat? Nem tűnik hatékony megoldásnak. Az Amd éppen ezért egy másik típusú befűzést is kezel: a `link` típust, amely közvetett hivatkozással (symbolic link) használható. Dániel térképelemét tehát a következőképpen lehet átírni:

```
dan -rhost:=raid3;rfs:=ujlemez/daniel
➔ host!=${rhost};type:=nfs
➔ host=${rhost};type:=link
```

Az átírt térképelem az Amd térképeinek újabb szolgáltatásaira irányítja rá a figyelmünket. Azzal kezdeném, hogy a visszaperjelek előtt szöközők vannak. Az Amd a visszaperjelek utáni szöközőket figyelmen kívül hagyja, de az előttük lévők nem. Dániel térképeleme tehát három szöközőkkel elválasztott részre osztható, amelyeket helyszíneknek nevezünk. Az első helyszín a kötőjellel kezdődik és a térképelem alapértelmezéseit adja meg, felülbírálva a `/defaults` részben megadott beállításokat. A második és a harmadik helyszín megadása egy kiválasztóval kezdődik. Az Amd térképkiválasztói dinamikus változók, amelyeknek az értékeit az Amd akár futási időben is vizsgálni képes és össze

tudja hasonlítani. Amint az el is várható tőle, az Amd tucatnyi kiválasztó használatát támogatja. Dániel térképelemét helyszínenként értékeli ki, egészen addig haladva, amíg valamelyik kiválasztó vizsgálata igaz értékhez nem vezet. Ekkor az itt megadott helyszínt fűzi be. A példában tehát az Amd először azt vizsgálja meg, hogy nem egyezik-e meg a futtató állomás neve az előre megadott rhost névvel. A raid3-tól eltérő nevű állomásokon tehát NFS-típusú befűzést végez, magán a raid3 állomáson ellenben a gyorsabb és egyszerűbb, közvetett hivatkozás alapú módszert alkalmazza.

Az `amd.home` térkép még egy okból tekinthető rossz hatékonyságúnak: a `/ujlemez/istvan` és a `/ujlemez/daniel` befűzése ugyanarról az NFS-kiszolgálóról történik, holott ezek nagy valószínűséggel ugyanannak a kívülről is elérhetővé tett fájlrendszernek az alkönyvtárai. Ez az eljárás lassú, pazarolja a rendszer mag erőforrásait. Szerencsésebb, ha a rendszer ugyanazt az `rfs` értéket használja, de elérési útként a pillanatnyi befűzési pontok alkönyvtárait adja vissza. (A visszaadott elérési utakhoz az alhivatkozások hozzáfűzése önműködően megtörténik.)

```
/defaults type:=nfs;sublink=${key}
istvan rhost:=raid3;rfs:=ujlemez
daniel rhost:=raid3;rfs:=ujlemez
```

/net térképek

Az önbefűző térképeket gyakran használják arra, hogy tetszőleges állomás tetszőleges fájlrendszerének a befűzését lehetővé tegyék; sokszor ezt hálózattérképnek nevezük. A térkép az összes fájlkiszolgáló elérésére átfogó és egységes módot kínál:

```
/defaults fs:=a/${rhost}/root/${rfs}
* rhost=${key};type:=host;rfs:=/
```

A példa rövid, de annál több újdonságot mutat. Először az `fs` változó alapértelmezett értékét adjuk meg – ez egyedileg azonosítja a befűzendő távoli NFS-kiszolgálót és fájlrendszert. Mindezt azért tehetjük meg, mert az `fs` változó azt a helyi gépen lévő elérési utat adja meg, ahova az Amd a távoli állomások fájlrendszereit fűzi be. Az ütközések elkerülése érdekében ennek az elérési útnak egyedinek kell lennie.

Második lépésként a tényleges térképelem kulcsát adjuk meg. Ez egy csillag, vagyis helyettesítő elemről van szó, amely bármilyen értékkel egyezik és a `key` változó értékét a `/net`-en belül keresett névre állítja. Ez a helyettesítő `key` érték lesz a távoli állomás neve (`rhost`). A következő művelet egy különleges, `host` típusú Amd befűzési elem megadása, az `rfs` változó értékével pedig az adott távoli állomás összes kívülről elérhető fájlrendszerének befűzését kérjük (kezdő elem a `/`). Az Amd `host` befűzési típusa úgy működik, hogy a távoli állomás `rpc.mountd` démonától lekéri a kívülről elérhető fájlrendszerek listáját. Ezt követően sorra befűzi őket. Tegyük fel, hogy a `proba` állomás két fájlrendszert tesz kívülről elérhetővé, ezek a `/homes` és a `/proj/X11`. Ha belépünk a `/net/proba` könyvtárba, az Amd befűzi a `proba:/homes` könyvtárat a `/a/proba/root/home` könyvtár alá, a `proba:/proj/X11` könyvtárat pedig a `/a/proba/root/proj/X11`

könyvtár alá. Ha a `/net/proba` könyvtárban kiadunk egy `ls` parancsot, akkor mindkét befűzött könyvtárat látni fogjuk.

Egy térkép mind felett

Nagyméretű telephelyeken, ahol több alcsoport is található (ezek számtalanszor részben saját felügyelettel rendelkeznek), a befűzési lehetőségek sokszor attól függenek, hogy éppen hol vagyunk. Előfordulhat, hogy a hatékony üzemeltetés érdekében a különböző gépekhez készült futtatható fájlokat különböző alhálózatokra terjesztjük ki. Amd-térképek központi gyűjteményét előállítva az összes helyi igényt ki lehet elégíteni:

```
/defaults type:=link
lbin in_network(eng);fs:=/local/${arch}/bin
↳ in_network(10.0.1.0/24);fs:=/x/beta/bin
```

A példatérkép az `in_network` kiválasztó függvényt alkalmazza. A kiválasztó függvények az igaz, illetve hamis értékeket a rendszer pillanatnyi állapota és a számokra átadott értékek alapján állítják elő. A kiválasztó megvizsgálja, hogy a pillanatnyi állomásnév része-e az eng hálózatnak; ennek megadása általában a `/etc/networks` fájlban található. Ha igen, az Amd az `arch` változó értékül a pillanatnyi géptípust adja. Így az `lbin` elem IA-32 rendszereken a `/local/i386/bin` könyvtárra, SPARC alapú rendszereken a `/local/sparc/bin` könyvtárra stb. fog mutatni. A térkép következő helyszíne szemlélteti, hogy az `in_network` kiválasztó a hálózat-hálózati maszk párok vizsgálatára is képes. A kiválasztó tulajdonképpen nemcsak különféle formátumú adatok vizsgálatára képes, de a helyi állomás bármely működő hálózati csatlózával képes az egyezések keresésére. Ezzel a lehetőséggel élve többlaki állomásokon is hatékony hálózathasználatot valósíthatunk meg.

ISO-9660 képek

Sokan szeretik ISO-9660 CD-ROM képekben (images) tárolni adataikat, csak hogy tartalmuk eléréséhez vagy CD-ROM-ra kell írni őket, vagy befűzésükhöz egy különleges hurok-illesztőprogramot (loop driver) kell használni. A `cdfs` befűzési típus alkalmas az ISO-9660 CD-k kezelésére. Ha a `dev` átadott értékben fájlnevet adunk meg és a `loop` befűzési módot választjuk, az Amd közvetlenül is képes

befűzni az ISO-képeket. Ezt követően úgy érhetjük el őket, hogy nem kell az összes ISO-képből kimásolnunk a fájlokat:

```
/defaults type:=cdfs;opts:=loop
shrike1 dev:=/iso/rh9/shrike-disc1.iso
shrike2 dev:=/iso/rh9/shrike-disc2.iso
shrike3 dev:=/iso/rh9/shrike-disc3.iso
```

Csináldmagad-térképek

Minden említésre érdemes eszköznek kellő bővíthetőséget kell kínálnia a váratlan helyzetek és igények kezelésére. Ahhoz, hogy az Amd befűzhessen valamilyen fájlrendszert, ismernie kell annak kezelési módját. A program befűzési típust használva egyedi befűzési és leválasztási módokat adhatunk meg, ha az operációs rendszer képes az adott fájlrendszer kezelésére, de az Amd esetleg nem:

```
r2 type:=program;dev:=/dev/sda1;
↳ mount:="/sbin/dohans dohans ${dev} ${fs}";
↳ unmount:="/sbin/undohans undohans ${fs}"
```

A fenti példában az előre megadott `dev` értéket és az önműködően meghatározott értékkel rendelkező `fs` értéket alkalmazzuk a `dohans` nevű héjparancsfájl futtatására. Ez végzi a `/dev/sda1` ReiserFS-ként történő befűzéséhez szükséges műveleteket.

Összegzés

Az Amd önbefűző nagyteljesítményű, sokoldalú eszköz, amely számos különböző környezetben alkalmazható. A rendszergazdák kellő gondossággal számos hasznos szolgáltatást kínálhatnak a felhasználóknak, miközben csökkenthetik a felügyeleti tennivalók mennyiségét. Az Amd az `Am-utils` csomag része, amely a legtöbb Linux-terjesztéshez előfordított állapotban is elérhető (☞ <http://www.am-utils.org>).

Linuxvilág 2003. október, 114. szám



Erez Zadok (ezk@cs.stonybrook.edu)

A Stony Brook Egyetem informatikai karán fájlrendszerekkel és operációs rendszerekkel kapcsolatos kutatásokat vezet. A *Linux NFS and Automounter Administration* (Sybex, 2001) című könyv szerzője.



Újdonságok a Blenderben

Akik igazán érdeklődnek a Blender lehetőségei és újdonságai iránt, bizonyára észrevették, hogy az elmúlt két hónapban a programnak újabb változatai jelentek meg.

Az egyik újítás a felhasználói felület kezelésében jelentkezik, most már ugyanis tetszőlegesen beállíthatjuk, hogy a felület milyen elemei milyen felosztásban foglalják el a munkaterületeket. A Blender háromféle ilyen előre beállított felosztást tartalmaz, amelyeket a nézeteken történő jobb egérgombos kattintással hívhatunk elő. Beállíthatjuk, hogy függőlegesen vagy vízszintesen helyezkedjenek-e el az egyes vezérlőelemek, vagy a *Free* menüpont kiválasztásával tetszőleges helyre vonszolhatjuk őket. Amennyiben az adott feladathoz nincs szükség a vezérlőelemekre, a fejlécükön található kis háromszögre kattintva be is csukhatjuk őket. Ilyen elrendezéseket láthatunk az 1. képen.

Új típusú mozgítás, forgatás és átméretezés

Korábban is alkalmazhattunk olyan módosítást, ami csak az egyik tengely mentén érvényesült. Most ennek továbbfejlesztett változataként a mozgítás, forgatás vagy átméretezés során alkalmazhatjuk az X, az Y vagy a Z billentyűket, amelyekkel a műveletet az adott világ tengelyére korlátozhatunk. Amennyiben másodsor is nyomjuk az adott billentyűt, a művelet a tárgy helyi koordinátatengelye mentén kerül végrehajtásra. A harmadik lenyomás alkalmával a Blender visszatér a korlátozás nélküli módba. Amikor ilyen korlátozás van érvényben, az adott tengelyt a Blender meg is jeleníti, így adva visszajelzést a kiválasztott módról. Észrevehetjük, hogy korábban a tükrözést is hasonló

módon végezhetjük el. Természetesen ez az újítás nem akadályozza a tükrözés végrehajtását, de újabban a pontszerkesztő módban kell majd kijelönnünk a tükrözni kívánt pontokat, s ezután az M billentyű hatására megjelenő menüből választhatjuk ki a tükrözés tengelyét.

Ezúttal sokkal egyszerűbben végezhetünk pontos átméretezést, eltolást és forgatást is. Amikor a megfelelő műveletet kiválasztjuk, egyszerűen beírhatjuk vagy az N billentyű lenyomása után megadhatjuk a kívánt számértékeket. Amikor ilyen módon viszünk be adatokat, a nézetablak fejlécében láthatjuk az adott értékeket, és amelyek érték körül a kisebb, illetve a nagyobb jelet látjuk, azt az értéket fogjuk módosítani. A TAB billentyűvel léphetünk a következő összetevőre, az ENTER-rel elfogadhatjuk a változtatásokat, míg az ESC-pel elvethetjük a változtatásokat.

A tárgyakat alkotó pontok kiválasztására is találhatunk egyszerűbb megoldást. Amikor pontszerkesztő módban vagyunk, a SHIFT-R billentyűvel egy egész pontsort kiválaszthatunk a tárgy pontjai közül. Azt, hogy mely pontok kerülnek majd kiválasztásra, a Blender sárga színnel fogja jelölni. Az előbbi kiválasztási mód ismételt alkalmazásával a már kijelölt pontokhoz hozzáadódik az újabb választás eredménye, míg az ESC billentyűvel elvethetjük az újabb pontok kijelölését. Amennyiben kiválasztottunk egy ilyen pontsort, a CTRL-R billentyűk lenyomásával újabb pontsor hozzáadásával azonnal feloszthatjuk a háromszögeket.



1. kép Átrendezett felület



2. kép A metszet

Néhány próbált javítás

A Lattice módosító most már metaball objektumok átalakítására is alkalmas. Sőt az újabb változatokban új metaball objektumokat találhatunk, ilyen például a síkfelület, a cső, a kocka és a torzított gömb. Ezek segítségével könnyebben készíthetünk például vízbe eső vízcseppeket vagy más hasonló animációt. A következő újítás, amiről említett kell tennem, a tárgyak vágására használható „kés” (*Knife*) eszköz. Amikor pontszerkesztő módban vagyunk, nyomjuk meg a SHIFT-K billentyűket – ekkor a felbukkanó menüből kiválaszthatjuk a vágás módját. Az *Exact* módszerrel pontosan a megrajzolt körvonal és a már meglévő élek met-

széspontjában keletkeznek új pontok, míg a *Midpoint* pontot választva azoknak a háromszögeknek a középpontjában keletkeznek az új pontok, amelyeknek az éleit a kijelölő vonallal átvágjuk. Figyelembe kell venni azt a tényt, hogy a kés eszköz aktiválása után már nincs lehetőségünk új pontokat kijelölni a tárgyon, és az eszköz csak a korábban már kiválasztott háromszögekre alkalmazható. További megkötés, hogy az eszköz hatása csak a felhasználóhoz, vagyis a képernyőhöz közelebbi pontokra és háromszögekre érvényes. Mindazonáltal hasznos segédeszközt építettek a Blenderbe a programozók, hiszen



3. kép Az elkészült forgástest

például ennek segítségével készíthetünk egy emberfejen szájníylást vagy rajzolhatjuk meg a szegződör körvonalát, majd a keletkező új pontokat a megfelelő helyre helyezhetjük át. Így nem kell élenként elvégeznünk a felosztást, ha új pontokat akarunk a meglévő tárgy egyes területein létrehozni, és nem kell minden élt a korábban használt *Subdivi* de művelettel felosztani.

Visszavonás

Ugyancsak hasznos újdonság a pontszerkesztő módban a művelet-visszavonási lehetőség. Amikor valamelyik tárgynak a pontjait elmozdítjuk, forgatjuk vagy más módon változtatjuk, a változásokat az U billentyűvel visszavonhatjuk. Sajnos a jelenlegi változatban ez a lehetőség még nincs teljesen kiforrvá, ha ugyanis két pontszerkesztés közben egy másik tárgyat is változtatunk, a visszavonás lehetősége már nem áll rendelkezésünkre. Hogy mégis hasznos segítségként alkalmazhassuk ezt az eszközt, célszerű egy-egy tárgyat teljesen befejezni, tetszőleges számban használva a visszavonást, és csak ezután fogni bele másik tárgy szerkesztésébe. A SHIFT-U billentyűvel visszatérhe-

tünk a visszavonás előtti állapothoz. Amikor egy tárgyat szerkesztünk, nemcsak a legutolsó változtatásokat vonhatjuk vissza, hanem az ALT-U hatására megjelenő menüből a megfelelő mélységű visszavonást választhatjuk ki. Tekintettel arra, hogy a visszavonás lehetősége nagyon memóriagényes, a legnagyobb mélységet állíthatjuk be, ha a nézetablak fejlécén kiválasztjuk a nagy „I”-betűt formázó ikont, a megjelenő új nézetben pedig az *Edit methods* gombot aktiváljuk. Itt egy beállító csúszkát találhatunk, amivel 0 és 64 között szabályozhatjuk a visszavonás mélységét – alapértelmezésben ez az érték 32.

Pontszerkesztő mód

M	tükrözés menüből
SHIFT-R	a szegmensek kiválasztása
Ctrl-R	a kiválasztott szegmensek felosztása
U	visszavonás
SHIFT-U	nincs visszavonás
ALT-U	tetszőleges mélységű visszavonás

Normál mód

N	számérték megadása
---	--------------------

A fenti táblázatban az újonnan tanult billentyűkombinációkat foglalom össze.

Hogyan készíthetünk forgástesteket a Blenderben?

Most pedig azt mutatom be, hogy miként készíthetünk forgástesteket. Elsőként hozzunk létre egy Bezier curve objektumot, majd váltsunk át pontszerkesztő módba, ahol alakítsuk ki a kívánt keresztmetszetet, ha szükséges, a CTRL billentyű melletti egérekattintással adjunk új pontokat a görbéhez. Helyezzük el a vezérlőpontokat úgy, hogy végeredményként a görbe alakja megfeleljen az elképzeltéinknek. Ezután lépünk ki a pontszerkesztő módból, és az ALT-C billentyűvel görbénket alakítsuk át szerkeszthető tárggyá. A pontszerkesztő módba visszaváltva jelöljük ki a tárgy minden pontját, és az E használatával a pontokat vízszintesen húzzuk ki. Ennek a kihúzásnak a mértéke fogja meghatározni a forgás-

test falvastagságát – ha ilyen nem szeretnénk megadni, akkor ezt a lépést hagyjuk ki. Ezután maradjunk pontszerkesztő módban, és a grafikus kurzort állítsuk oda, ahol majd a testünk középpontja lesz. A Blender ugyanis a kurzor körül fogja a kijelölt pontokat elforgatni. Miután meghatároztuk a középpontot, válasszuk ki azt a nézetet, ami az elkészült tárgy felülnézete lesz. Az F9-cel válasszuk ki a szerkesztő nézetet, és itt, a *Mesh Tools* vezérlők között láthatjuk a *Spin* gombot. Ez alatt szerepel, hogy a forgatás során a Blender hány fokkal forgatja el a keresztmetszetet; a mellette látható *Steps* érték pedig azt adja meg, hogy ezt a szögelfordulást hány lépésben kell elérni. Itt célszerű 360 fokot beállítani, míg a lépésszám az olvasóra van bízva. Természetesen nem szükséges nagy lépésszámot megadni, 16-os érték mellett már szép eredményt érhetünk el, kisebb értékek megadásával viszont érdekes „forgástesteket” hozhatunk létre, például négyszög alapú testet is készíthetünk. Miután a szükséges értékeket beállítottuk, kattintsunk a *Spin* gombra, és adjuk meg, hogy milyen nézetben szeretnénk a műveletet végrehajtani. Amennyiben a metszetet oldalnézetben készítettük el, a felülnézet lesz megfelelő számunkra, ezért is váltottunk már korábban a készülő tárgy felülnézetébe. Tehát kattintsunk a kiválasztott nézetre és szemléljük meg tevékenykedésünk eredményét. Ha nem vagyunk elégedettek a látvánnyal, az U billentyűvel még visszatérhetünk a kezdeti metszethez, és azt átalakítva újabb tárgyat hozhatunk létre. Kiegészítésként ennyi lenne a hozzáfűznivalóm. Remélem, az elmúlt hosszú, csendes időszakban mindenki az érdeklődésének megfelelő mértékben elmélyedt már a program használatában. További kellemes időtöltést kívánva búcsúzom.



Fábíán Zoltán

(dzooli@freemail.hu)

26 éves, jelenleg oktatóként dolgozik, szabadidejében szívesen foglalkozik Blenderrel, programozással

és elektronikai tervezéssel. Szereti a természetet, túrázást, úszást és a kellemes baráti társaságot.

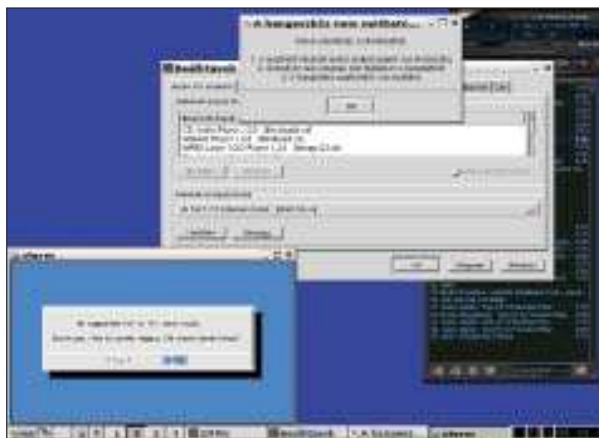
Debian otthonra (3. rész)

Daloljon, őrmester!

Köszöntök minden álmatlanságban szenvedőt! A korábbi cikkek alapján, úgy gondolom, mindenki sejti, hogy hosszabb kihagyás után érdemes frissíteni (ez a Sarge esetében négy-öt napot jelent), egyúttal ismételten érdemes egy erősebb netkapcsolatért fohászzkodni. Igen, sajnos vannak olyan területei kis hazánknak, ahol ez – úgy tűnik – tényleg az égiek hatáskörébe tartozik. Ne is menjek nagyon messzire: a szerkesztőség pont egy remek kis helyen van, a nyolcadik kerület sűrűjében, négy sarokra a Blaha Lujza tértől, a Köztársaság térrel szemben. És rendes (értsd: modernnél erősebb) netkapcsolat nincs a házban. Kész vicc, hogy a szemközti épületben még tudnak kábelmodemezni, az út túloldalán viszont ez sajnos „technikailag nem megoldható”. A másik szolgáltató véleménye szerint meg „errefele nincs igény”. Nnna, kipuffogtam magam. Szóval frissítsünk! És ha már a frissítéseket ilyen gyakran végezzük, még egy apróságra érdemes figyelni. A rendszer először letölti a csomagokat, majd telepíti őket, végül a letöltött csomagokat otthagyja a lemezen. Mivel nem ritka, hogy egy Sarge alatt egy hét után frissítve száz megát töltünk le, érdemes a régi csomagokat letörölni. Az apt a csomagokat a `/var/cache/apt/archives` könyvtárban tartja, innen bátran és gátlástalanul törölhetjük az összeset. Vigyázz, a `/var/cache/apt` alatt két `.bin` kiterjesztésű fájl is található, azokat hagyd meg. Egyébként egy esetben érdekesek lehetnek a `archives` alatti csomagok: ha próbálgatni szeretnéd a `dpkg` használatát.

A `dpkg` valójában az `apt`-rendszer mögött megbúvó csomagkezelő program. Ő végzi az egyes csomagok telepítését, leszedését. Megadod neki a fájl nevét egy `-i` kapcsolóval, és vígan telepíti. A hátulütője, hogy nem kezd el mindenféle „függőségek miatt szükséges” csomagot letölteni neked, előnye viszont, hogy fájlokkal tud dolgozni, nem kell neki forráslista.

Nézzünk egy példát! Jelöltünk vígan letölti az új Opera böngészőt a `http://www.opera.com/download` címről (természetesen a legújabb elérhető változat sosem „stable”, de hiszen veszélyesen kell élni!), majd kiadja a `dpkg -i ./opera_7.23-20031119.2-shared-qt_en_i386.deb` utasítást. Erre a rendszer vagy feltelepíti, vagy nem. Ilyenek ezek a számítógépek, binárisan gondolkodnak. Ha a függőségek miatt nem sikerül a letöltött programot telepíteni, a leggyakoribb hiba, hogy a program egy, a rendszerről hiányzó könyvtárra támaszkodna. Esetünkben ez a QT3 többszálú változata lehet. Két lehetőségünk van: vagy valahogy



felimádkozom a `libqt3-mt`-t a gépre, vagy beszerzek egy olyan opera-csomagot, amibe ezt a könyvtárat is belefördítették. Aki nagyon ráér, valamint további tapasztaltpontokra van szüksége, hogy rendszergazda típusú kalandozófigurája szintet lépjen, nyugodtan kísérletezzen a `libqt3-mt` telepítésével, kényelmesebb olvasóim pedig eleve a másfél megával nagyobb `opera-static` csomagot töltsék le.

De jajaj, máris hogy előrerohantunk! Böngészés? És ráadásul Operával? Hiszen még hangja sincs a mi kis drágaságunknak! Na jó, lehet, hogy volt neki, de néhány frissítés után tuti, hogy eltűnt valahol. Így van ez, Gollam a költői szépségű barlangjában ücsörögve csak szeretgette őt, de ő elhagyta Gollamot.

Hangerőt ide!

Bepillantást nyerhetünk a nyílt fejlesztések és a korlátlan szabadság egy újabb csodájába, az egymás mellett és helyett működő hangkezelő alrendszerek világába. A történet még azokra az időkre nyúlik vissza, amikor a hangkártya kuriózum volt. A hangkártyagyártók nem szívesen adtak ki adatokat a kártyájukról, elsősorban üzleti megfontolásokból, így alakulhatott ki, hogy a különböző hangkártyákhoz eltérő rendszerek születtek. Azután ezek a rendszerek egyre több kártyát ismertek fel az adott alrendszer fejlesztőinek kedvétől (és az ingyen kapott próbakártyáktól) függően. A feladatot tovább nehezítette, hogy több szinten folyamatosan változott a rendszer. Nem elég, hogy némelyik ablakkezelő környezet is saját hangkezelő alrendszer

fejlesztett, „Majd én rendet teremtek itt!” felkiáltással, de mélyebb szinten is komoly változások történtek, például az eszközök nevei terén. A jelenlegi, végkifejletéhez közeledő helyzet már valamennyire letisztult, mégis egy becsületes, magára valamit is adó hangkezelő program vagy felvállalja, hogy több alrendszerrel támogató, vagy közli, hogy neki ez vagy ez kell, és kész.

És mik is vannak? Mélyebb, gépközeli szinten az első egységesítő rendszer az OSS (Open Sound System) volt, amely mögött egy kereskedelmi cég, a 4Front áll (☞ <http://www.opensound.com>). A cég ennek fényében az egész fejlesztést két vonalon viszi: a „nyitott”, valamint a „fizetős” vonalon. Annak ellenére, hogy sok része fizetős, a rendszer szabadon használható része (OSS/Free) jól



működik és közkedvelt. Érdekeség, hogy a szinte mindenki által használt `xmms` lejátszót is a 4Front fejleszti (nyílt forrású fejlesztés). A másik nagy csapat ALSA névre hallgat (☞ <http://www.alsa-project.org>). Ez a vonal teljesen nyitott, szintén remekül használható, valamint teljes mértékben igyekszik nyújtani az OSS/Free szolgáltatásait. Véleményem szerint ehhez a rendszerhez lényegesen könnyebben használható programokat kapunk.

Emellett még a feleslegesen bonyolulttá és csicsássá összehátolt ablakkezelők hangrendszereit is pillantsuk meg. A KDE2 szerves része, ahogy a honlap mondja, „alapvető eleme” (☞ <http://www.arts-project.org>) az aRts (Analog Realtime Synthesizer), ami csudaügyes-csudaszép, és csudára nem volt hajlandó együttműködni a legtöbb programmal. Ennek megvolt az oka is, hiszen ki kellett alakítani egy olyan keretet, amelyre minden multimédia-program egységesen építhető. Nagy kihívás volt például, hogy több program miként használjon egyszerre egy hangkártyát. Erre született meg egyébként a másik csilivili környezetben az Esound csomag is, amelyet az már ott is ASD igyekszik lecserélni (ha nem volna még elég). Egyébként mindegyik honlapján megtaláljuk a „Minek egy újabb hangdémon?” című oldalt, az ASD esetén ez a nyitóoldal is kikerült (☞ <http://asd.sourceforge.net>).

Ha valaki még további okosságokkal kívánja tömni a fejét ezen a téren, szívből ajánlom a – szerintem – legértelmesebb oldalt, a ☞ <http://google.com> használatát. És nem csak úgy össze-vissza keresgetni ám! Tessék besétálni a Google-

könyvtár megfelelő oldalára (☞ http://directory.google.com/Top/Computers/Software/Operating_Systems/Linux/Hardware_Support/Sound/?tc=1)!

Ezek után nézzük a gyakorlatot! Alulról építkezve több lehetőségünk van, de ha komolyan gondoljuk, akkor mindenképpen magiszintű támogatásra van szükségünk. Szerencsére az előfordított magokban a legtöbbször mind az OSS, mind az ALSA támogatva van. Fontos viszont megjegyezni, hogy az ALSA moduljait külön csomagként érhetjük el (`alsa-modules-2.4.sorszám`). Hogy egy adott rendszer most éppen hogyan áll, könnyen kiderítheted, ha megnézed, hogy milyen modulokat használ, illetve hogy egyáltalán milyen modulok érhetőek el.

A rendszermag által használt modulokat az `lsmod` listázza ki vagy a `modconf` programban nézheted meg. A `kernel/drivers/sound` ágba mindenképpen szükséged lesz a `sound` és a `soundcore` modulokra. Előfordulhat, hogy hangkártyád a kettő rendszer közül csak az egyik támogatja. Ilyenkor a döntés – miszerint is OSS vagy ALSA – nem rajtunk áll. Ha a gépedben például ALI 5451-es vagy 5455-ös hangkártya van (számos laptop ezt tartalmazza), akkor máris döntésre jutottál, ugyanis az 5455-öshöz a `kernel/drivers/sound` alatt találsz támogatást, az 5451-eshez viszont az `alsa` alatt.

Ha mind a két rendszer támogatja a kártyát, és választani lehet, én az ALSA-ra szavazok, már csak azért is, mert ennek felélesztése és üzemeltetése egy kicsit trükkösebb, így jobban elősegíti a linuxos társadalom összekovácsolódását. Azonkívül sok kényelmes program érhető el, ilyen például az `alsacnf`, amelyik megtalálja az összes, az ALSA által támogatott hangkártyát, majd üzembe is helyezi őket. A hangrendszerek kapcsán azonban szintén gyakran előtérbe kerül a hit kérdése, mégpedig a rendszergazának a számítógépebe vetett hite. Mert mit is várunk el a rendszertől? Elvárjuk például, hogyha egyszer megszólalt, utána is szóljon. Nos, ez nem így van ám. Mert ugye előfordulhat, hogy csak a rendszergazda tud zenét hallgatni (a felhasználókat hozzá kell adni a `audio` csoporthoz), újraindítás után csend vala (Elmenttetted a beállításokat? Nem kapcsolod ki a hangot a szabályozóban?), az ALSA nem akarja támogatni az OSS-alkalmas programokat (bekapcsoltad az OSS-réteget a `/etc/default/alsa` fájlban?), vagy éppen rendszermagot frissítettél (az új rendszermaghoz letöltötted az `alsa-modules-t`? Létezik egyáltalán hozzá?). Röviden összefoglalva tehát, az ember ki van szolgáltatva az elemek tombolásának.

Ha valamilyen célgépet készítesz, például külön rendszert építesz a kedvenc játékodhoz, akkor előfordulhat, hogy magad akarod lefordítani a rendszermagot és az ALSA-rendszert is. Ehhez segítséget a ☞ http://www.sonic.net/~rknop/linux/debian_alsa.html című oldalon találsz.

Nos, a hangkezelés valóban nagy falat, remélem, ebben a cikkben sikerült megismertetni az alapokkal és képes vagy a saját gépedből is hangot kicsikarni! És ha a Linux és a zene közös világában komolyabban el szeretnél mélyedni, ajánlom *Dave Phillips* Linux Zene és hang című könyvét. Kellems újratelepítéseket kívánok!

Szy György



A DVD és a Linux

A napokban a szekrényemben ráakadtam egy csomó régebbi számítástechnikai szaklapra. Nem tudtam ellenállni a kísértésnek, és egy kicsit nosztalgizni kezdtem.

Az egyik tekintélyes lap 2000. évi valamelyik számát lapozgatva ráakadtam egy háromoldalas, Linuxról szóló cikkre. Az írás azt méltatta, hogy milyen vonzó a Linux és mennyire más, mint a többi rendszer, valamint kitért néhány alapfogalomra is. A cikk végén a következőt olvastam: aki Linuxot használ, annak egyelőre le kell mondania az USB és a DVD nyújtotta élményekről. Felsőhajtottam – valóban, akkoriban ez még így volt.

Élénken emlékszem, hogy valamikor mekkora újdonság volt a DVD, és a Linux csakugyan nem támogatta. Ez azonban gyökeresen megváltozott: a DVD gyakorlatilag ugyanolyan természetes lett Linuxon, mint a CD, viszont nem annyira egyértelmű a telepítése és a használatba vétele. Jelenleg DVD-n jellemzően filmeket találhatunk, bár a kapacitása következtében nagyobb programoknak vagy operációs rendszereknek is megfelelő tároló eszköz. A szaküzletekben pedig manapság már mindenféle árkategóriában kaphatunk meghajtókat, akár olyat is, amely egyúttal CD-író vagy CD-újraíró. Ezek a kombómeghajtók is megfizethetők, gyakorlatilag két jobb minőségű játék árértékért már beszerezhetők.

Én egy LG GCC sorozatú DVD-CDRW-meghajtóval rendelkezem, amelyet játszani könnyedséggel lehet működtésre bírni Linux alatt. A meghajtót a gépbe másodlagos szolgálként (secondary slave) raktam be, így teljesen független lett a merevlemezről és nem fogja vissza.

Ezzel az eszköz telepítését gyakorlatilag el is intéztük. Linux alatti működéséről viszont érdemes megjegyezni egy nagyon fontos dolgot: a legtöbb Linux-változat a CD-írókhoz hasonlóan a DVD-meghajtókhoz is SCSI emulációt használ. Tehát hiába vagyunk biztosak abban, hogy a meghajtó IDE-csatolású, Linux alatt nagy rá az esély, hogy SCSI-meghajtóként kell keresnünk, amennyiben a meghajtó kombó (azaz CD-író is); akkor pedig teljesen biztos, hogy kizárólag SCSI-ként érhetjük el (ha csak DVD-olvasó az eszközünk, akkor IDE eszközként is remekül működik).

A telepítés

Nem túl valószínű, hogy a megvásárolt DVD- vagy DVD+R/RW-meghajtónkat nem támogatja a Linux, viszont



1. kép Az MPlayer kényelmes és gyors, ráadásul magyarul beszél

érdemes rá kitérnem, hátha mégis így alakul. A Linuxnak önműködően fel kell ismernie, hogy milyen meghajtónk van, amiben a meghajtó gépi kódja (firmware) segíti. Ekkor tölti be a rendszer a SCSI emulációs modult. Ha nem ismeri fel, illetve felismeri, de mégsem tölti be a megfelelő modult, akkor ezt sajnálatos módon nekünk kell megtennünk. Szerencsére ez nem nagy kitérő az eredeti telepítési folyamat során: mindössze annyi a feladatunk, hogy a LILO vagy a Grub rendszerindító menüjét a feladatnak megfelelően átszerkesszük. Itt tudjuk ugyanis beállítani a rendszer a megfelelő kapcsolóit, vagyis azt, hogyan induljon el a gép.

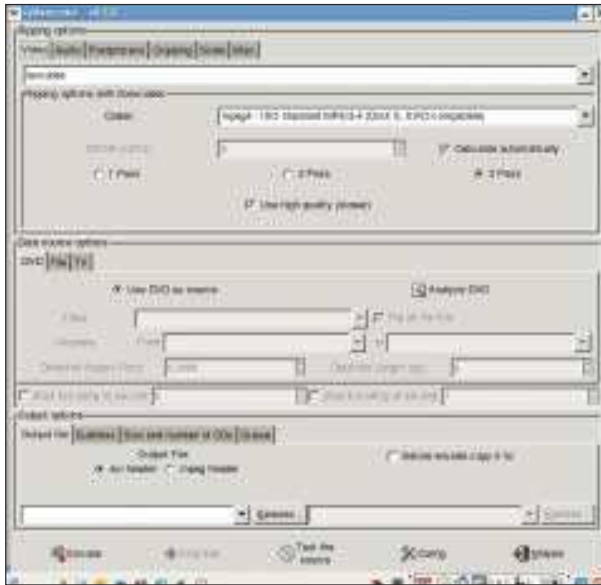
Ha a meghajtót a Secondary Slave kábelre helyeztük (ügyeljünk a meghajtó beállításaira), a Linux számára a `/dev/hdd` eszközként jelentkezik. Ekkor az indítómenü menüpontja alatti értékekhez hozzá kell tenni a következőt:

```
/dev/hdx=ide-scsi
```

A fenti példából kiindulva:

```
/dev/hdd=ide-scsi
```

Ez akkor választható beállítás, ha a meghajtó csupán DVD – kötelező azonban, ha az eszköz kombó, ha ugyanis kombómeghajtó esetén a fentiek a rendszer önműködően



2. kép A GMEncoder egy grafikus felület a Mencoderhez



3. kép A Xine remekül kezeli a videófájlokat

nem teszi meg, és mi sem segítünk neki, DVD-khez használhatjuk ugyan, de CD-íróként nem fog tudni működni. Ha az indítófolyamat (boot) során látjuk, hogy betölti az ide-scsi modult, vagy (SuSE esetén) az indító menüpontnál ezt tapasztaljuk, akkor biztosak lehetünk benne, hogy ez ügyben nincs teendőnk. Mindössze azt fontos észben tartani, hogy az eszköz további beállítása során ne a `/dev/hdX`, hanem a `/dev/sdX` eszközre hivatkozzunk (a `/dev/sd..` az SCSI-meghajtók eszközeirőjüket).

Ha mindez megvan, és a Linux betöltődött, célszerű ellenőrizni néhány értéket, vagyis azt, hogy létrejött-e minden eszközeirő. Forduljunk a konzolhoz és kérdezzük is meg. Linux-változattól függően kombómeghajtó esetén a következő eszközeirők jöhetnek létre:

```
ls -l /dev/cdrecorder (SuSE)
```

```
ls -l /dev/cdwriter (egyéb terjesztések)
ls -l /dev/cdr (UHU Linux)
```

Ha valamelyik megjelenik, akkor minden rendben. Ha viszont egyik sem, létre kell hozni a megfelelőt. Ellenőrizzük, hogy magát az eszközt felismerte-e:

```
ls -l /dev/sdX
```

(Például SuSE esetén az X a szalagkábelhez tartozó betűjel.)

```
ls -l /dev/cdroms/cdrom0 (UHU Linuxban így érzük el.)
```

Az UHU Linux esetén annyival bonyolódik a helyzet, hogy a meghajtót egy elég hosszú névvel érhetjük el. Nálam a Secondary Slave kábel elhelyezkedő kombómeghajtó a következő útvonalon található:

```
/dev/scsi/host0/bus0/target0/lun0/cd
```

Ebben az esetben maga a cd az eszközeirő, a többi pedig könyvtár. Ez az eszközeirő mindenképpen létrejön, ha maga a gép felismerte a meghajtót, tehát fizikailag működik. UHU Linux esetén ennek közvetett hivatkozása (symbolic link) a `/dev/cdroms/cdrom0`.

Ha mégsem jönne létre önműködően (ez nagyon ritka), akkor nekünk kell létrehoznunk a hivatkozást, ehhez rendszergazdaként kell bejelentkezni. Először használjuk a `su` parancsot, sikeres bejelentkezés után pedig hozzuk létre a közvetett hivatkozást.

```
ln -s /dev/scsi/host0/bus0/target0/lun0/cd
    /dev/cdroms/cdrom0
```

(Ha más a CD-meghajtó, akkor értelemszerűen `cdrom1` lesz a hivatkozás neve.)

Más Linux-terjesztésekben elég, ha a `/dev/sdX` eszközeirő létrejött, mert ebből kiindulva már egyszerű a dolgunk. Mindez kizárólag a CD-író vagy a DVD-CD-író kombók esetén fontos, egyszerű DVD-meghajtó esetén nem. Ez az egyszerűbb eset, mert ekkor valószínűleg azonnal létrejött a `/dev/dvd` eszközeirő is, és mint említettem, a SCSI emuláció is teljesen felesleges, tehát ha a rendszer `/dev/hdX`-ként látja, az teljesen megfelelő.

Ha mégsem jött volna létre a `/dev/dvd`, ezt is készítsük el:

```
ln -s /dev/hdX /dev/dvd
```

Ezzel telepítettük a DVD-meghajtót. Ha viszont kombó, akkor csak most kezdtük el. Ebben az esetben, ha létre is jön az eszközeirő, az valószínűleg a CD-íróra érvényes. Linux-terjesztéstől függően az alábbiak jöhetnek létre, ellenőrizzük is őket:

```
ls -l /dev/cdr (UHU Linux)
ls -l /dev/cdrw (UHU Linux, CD-RW-meghajtó esetén)
ls -l /dev/cdrecorder (SuSE, Mandrake)
ls -l /dev/cdwriter (szintén SuSE vagy Mandrake)
```

Szinte biztos, hogy kombó esetében a `/dev/dvd` nem fog



4. kép Az UHU Linuxban alapértelmezetten a Dvd:rip települ

létrejönni – ekkor el kell készítenünk:

```
ln -s /dev/hdX /dev/dvd
```

(SuSE, Mandrake stb. DVD esetén szinte mindegy, melyiket használjuk, ha létezik a */dev/hdX* vagy a */dev/sdX*.)

```
ln -s /dev/sdX /dev/dvd
```

(Ha ezúttal is az SCSI emulációt szeretnénk használni.) UHU Linux esetén, csak ha már létrejött (létrehoztuk) a */dev/cdroms/cdromX* (0, 1, 2...n) eszközeleírót, akkor adjuk ki a következő parancsot:

```
ln -s /dev/cdroms/cdromX /dev/dvd
```

Figyelem, ha kizárólag ez lesz az egyetlen CD-olvasásra is alkalmas meghajtónk, ne felejtjük el kiadni az `ln -s /dev/cdroms/cdromX /dev/cdrom` (UHU Linux), illetve a `ln -s /dev/hdX (sdX) /dev/cdrom` (SuSE, Mandrake) parancsot, mivel ha nem jött létre, akkor nem fogunk tudni sem videoCD-t nézni, sem audio-CD-t hallgatni.

Még egy fontos dolog azért akadhat: ha egy film megtekintése közben a lejátszás akadozna, valószínűleg nincsen beállítva a közvetlen memóriaelérés (DMA, vagyis direct memory access). Ekkor annyit a teendők, hogy a terminálban rendszergazdaként jelentkezünk be, és a `hdparm` parancsot használjuk:

```
hdparm -d1 -k1 /dev/...
```

(A `-d1`-gyel bekapcsoljuk a DMA-t, a `k1`-gyel utasítjuk, hogy ezt „jegyezze meg”. A */dev/...* pedig maga az eszköz.)

Lejátszók és kódolás

Ha eddig eljutottunk, a telepítést be is fejeztük, és nincs más hátra, mint a lemezt behelyezni a meghajtóba, és kiválasztani egy DVD-lejátszóra is alkalmas programot. Több ilyen apró programot is megemlíthetünk, kettő azonban kiemelkedik a mezőnyből: az MPlayer és a Xine. Mindkettő ingyenesen beszerezhető, sőt az összes Linux-terjesztés tartalmazza valamelyiket. Az UHU és a SuSE az MPlayert, a Mandrake pedig a Xine-t. Figyelem: a SuSE által telepített MPlayer nem alkalmas a DVD-nézésre, mivel nem tartalmazza a DeCSS csomagot, illetve számos kodeket – készítői teszik mindezt azért, mert tartanak a DeCSS tisztázatlan jogállása miatt. Ezért a SuSE esetén mindenképpen újra kell fordítani az MPlayert, hiszen hiába telepítenénk külön a DeCSS-t, ezt nem tudnánk a lejátszó „tudomására” hozni. (Az MPlayer fordításának leírása a Linuxvilág 2002. márciusi számában található meg.) Ha UHU Linux alól szeretnénk használni, akkor viszont tényleg semmi dolgunk nincs, mivel az UHU-val települő MPlayer ezt alából tudja. Sőt mindkét lejátszót telepíthetjük, hisz a Xine-nak is létezik *.uhu* csomagja. Telepítése a legegyszerűbben az `apt-get`-tel történhet. Rendszergazdaként konzolon keresztül jelentkezünk be és adjuk ki az alábbi parancsokat:

```
apt-get install xine-lib
```

(A Xine függvénytárainak és kodekjeinek a csomagja.)

```
apt-get install xine-gui
```

(Maga a lejátszó és grafikus felülete.)

Ezek után már tényleg nincs más dolgunk, mint hátradőlni, valamint lemezt helyezve a meghajtóba megnézni a kedvenc filmünket.

MPlayer

Ha az MPlayert választjuk, akkor egy kényelmes és gyors, ráadásul teljesen magyar nyelvű és készítésű lejátszóval élvezhetjük filmünket, amely nagyon jól kezeli a külső felirattájakat, mindemellett könnyű kezelni és jól méretezhető. A leírása is magyar nyelvű, és teljes értékű lejátszóként „kulcsra” készen kapjuk az UHU Linuxszal. Arról nem is beszélve, hogy az MPlayer egy Enkóder programmal (Mencoder) is rendelkezik, amelynek segítségével a filmet képesek vagyunk leszedni a DVD-ről a merevlemezre, szinte bármilyen formátumban. Ehhez készült a `gmencoder` grafikus felület, így nem kell parancssorból ügyeskedni, hanem elegendő az egérrel kattintgatni. Szintén elérhető *.uhu* csomagban, és az `apt-get`-tel telepíthető is. Maga a program kiválóan kiaknázza a mencoder lehetőségeit, szinte mindent be lehet állítani, s a leszedett filmet a kiválasztott kodekkel akár azonnal képes egy vagy több CD-nyi méretűre tömöríteni, hogy a kész fájlok aztán azonnal fel is írja.

Xine

A másik program, amiről beszéltünk, a Xine. A Xine nem magyar fejlesztésű, viszont szintén ragyogó kis program. Remekül kezeli a videófájlokot, viszont nem méretezhető

olyan jól, mint az MPlayer. Ismeri a kattintásra történő képlőpást a filmekből, és akik az asztali lejátszók kényelmére áhítoznak, azok számára bizonyára öröm, hogy a Xine a DVD-menüt is kezeli. Így tényleg teljes a DVD-élmény, kiváltképpen, ha megfelelő hangrendszerrel rendelkezünk hozzá. A Xine szintén nyílt forrású fejlesztés, így a forráskódja is elérhető, valamint ugyanúgy fordítható, mint az MPlayer. Saját tapasztalatom szerint azonban az MPlayer-t valamivel könnyebb fordítani, mint a Xine-t. Természetesen ez a lejátszó is elérhető *.rpm*, *.deb*, *.uhu* csomagban.

Akinek viszont vonzóbb a külső program a DVD merevlemezre való kódolásához, annak jól jöhet a Dvd::rip. Ez is szinte az összes rendszer alá elérhető, és az UHU-ban alapértelmezettként települ.

Ez a program olyan kényelmes a GMencoderhez képest, hogy a CD beállításainál azt is meghatározhatjuk, hogy milyen CD-t szeretnénk írni a tömörített anyagból. A beállítások között szerepel az iso9660 (a fájl egyszerűen CD-n), a videoCD, de a SuperVideo CD lehetőség is. Az utóbbiak előnye, hogy esetükben az asztali DVD-lejátszón is megnézhető az írt CD. Minden másban szinte teljesen megegyezik a gMencoderrel, ugyanúgy beállítható a fájlnev, hogy hány CD-re szeretnénk rátenni a filmet, adhatunk hozzá külső feliratfájlt, és az elkészült fájlokat szintén azonnal képes CD-re felírni. Leszedhető a teljes film vagy csak egy részlet, de megadhatjuk neki azt is, hogy csak az X-edik perctől az Y-edik percig írja ki. Egyetlen hátránya van a GMencoderrel szemben: a Dvd::rip először a teljes

DVD anyagát nyers adatként leszedi a merevlemezre, és csak utána kezdi el végrehajtani a kiadott feladatot. Ez pedig csak olyan esetben működhet, ha sok szabad lemezterülettel rendelkezünk.

A rendszerigény

A DVD-filmek megtekintése nem igényel hatalmas erőforrást, egy 500 MHz-es processzorral szerelt gépen teljesen élvezhető. Az MPlayer készítői szerint azonban 300 MHz-es processzor alatt ne is kísérletezzünk, és ez igaz is, ha azt veszem alapul, hogy egy ilyen gépen elindul ugyan a film, de nem beszélhetünk élményről. A kényelmes filmnézéshez az 500 MHz-es processzor melegen ajánlott, ha pedig le is szeretnénk szedni az anyagot a fent említett két program valamelyikével, akkor tényleg a 800 MHz-es az alap. Természetesen a memóriaméret is fontos ilyenkor, mivel a tömörítés lényegesen több erőforrást igényel, mint a megtekintés. Ilyen rendszerfeltételek mellett elmondhatjuk, hogy manapság a DVD mindenki számára elérhető lett, immáron felületfüggetlenül.

Kellemes szórakozást a DVD-k világában!



Dancsok „strogg” Zoltán (strogg@mail.tvnet.hu)

Jelenleg technikai szerkesztőként dolgozik a BME-OMIKK-nál, ahol oktat is. Emellett egyetemi képzésben vesz részt, programozó matematikus szakon. Négy éve foglalkozik Linuxszal. Szabadidejében operációs rendszereket gyűjt és weblapot vezet.



Héjprogramozás Linux alatt (10. rész)

Héjprogramozásról szóló cikksorozatunk utolsó részében a kifinomultabb programozási fogásokból és a különleges megoldásokból mutatunk be néhányat.

Számos olyan Unix-parancs, illetve segédprogram létezik, amelyik képes a rekurzióra, vagyis arra, hogy egy bizonyos műveletet ne csak egy adott könyvtár fájljain, hanem egy teljes könyvtárszerkezeten hajtson végre. A legegyszerűbb példa erre talán a mindenki által ismert „statáriális” törlés, az `rm -rf *` parancs, amely a pillanatnyi könyvtártól kezdődően mindent eltüntet. Némi képzelőerővel egy héjprogramot is felruházhathatunk efféle „teljes hatókörű áttekintőképességgel”. Tegyük fel, hogy egy nagyobb programot fejlesztünk, amelynek a kódja több tíz könyvtárban szétszórt (pontosabban rendszerezett...) több száz forrásfájlból áll. Ebben a rengetegben megtalálni valamit – például egy függvény meghatározását – néha egyáltalán nem egyszerű feladat. Nagy hasznunkra lehet a `grep` parancs.

Milyen jó is lenne, ha volna egy „rekurzív” `grep` parancsunk, ami erre is képes lenne (igaz, ami igaz, van ilyen program, a `grep -r` illetve az `rgrep` parancsok pontosan ezt teszik, de a feladat szépsége miatt ezt mégsem hagyhatjuk ki). Nosza, írjuk meg! A legtöbben most valószínűleg többoldalas forráslistára számítanak, pedig a megoldás az összes szokásos körítéssel együtt mindössze ennyi:

```
1: #!/bin/sh
2: # Önhívó grep parancs
3:
4: PROGRAMNEV=`basename $0`
5:
6: if [ $# -ne 1 ]
7: then
8:     echo "Használat: $PROGRAMNEV
9:         ↪ <keresendo minta>"
10:    exit 1
11: fi
12: find . -type f -exec grep "$1" {} \;
```

Tizenkét sor, amiből az első tizeneggyel kapcsolatban valószínűleg már semmit sem kell elmagyaráznunk. A tizenkettedik sor viszont határozottan érdekes. Itt a kereséshez a valamenyinyi Unix-rendszeren megtalálható `find` segédprogramot használjuk. Ez nemcsak a feltételnek megadott fájlok kikeresésére képes, hanem arra is, hogy egy adott műveletet végrehajtsunk

rajtuk. Ez utóbbit a `-exec` kapcsoló után adhatjuk meg. Az egyszerűség kedvéért nézzük végig egyenként a 12. sor elemeit! A `find` utáni `.` (pont) azt jelenti, hogy a keresést a pillanatnyi könyvtárban kezdjük. A `-type f` segítségével határoztuk meg, hogy fájlokat keresünk, vagyis a könyvtárak „kimaradnak a játékból”. Az `exec` után látható `grep` pontosan az, aminek látszik: a jól ismert `grep` parancs, amit most a `find` minden olyan fájlra végre fog hajtani, ami a keresési feltételeknek megfelel. A találati listát a `{}` (az egyszeres kapcsos zárójelkettős) jelöli, vagyis a `find` ezt fogja helyettesíteni a megtalált fájlok listájával. A végrehajtandó parancssor végét a pontosvessző jelzi, amit azért kellett levédenünk, mert a héj szintén értelmezi.

Néhány célszerű kiegészítés

Ez a program valóban úgy működik, mint a klasszikus `grep` parancs, vagyis a teljes könyvtárszerkezeten végighaladva minden fájlból megjeleníti azokat a sorokat, amelyekben szerepel a keresett szó vagy kifejezés. Példánknál maradvá, ha csak az a célunk, hogy a forrásfájlokból kikeresünk egy változó alapértelmezett értékét vagy megállapítsuk egy függvény értékeinek helyes sorrendjét, akkor ez a megoldás tökéletes. Hogyha azonban arra vagyunk kíváncsiak, hogy hol volt találat, akkor ezzel a kimenettel semmire sem megyünk. Helyette nyilván arra volna szükségünk, hogy a találati sorok helyett a megfelelő fájl neve jelenjen meg. Szerencsére megint nem kell messzire mennünk a megoldásért. Amennyiben átolvassuk a `grep` parancs súgóoldalát, láthatjuk, hogy – számos egyéb mellett – létezik egy `-l` kapcsolója is, amely a szokványos működés helyett éppen azt teszi, amire szükségünk van: fájlnevet jelenít meg a tényleges találat helyett (ez természetesen igaz a `rgrep` és `grep -r` parancsokra is). Nem kell tehát mást tennünk, mint kiegészíteni a 12. sort ezzel a kapcsolóval:

```
find . -type f -exec grep -l "$1" {} \;
```

Megeshet azután, hogy a keresést csak adott típusú fájlokra, például egyedül a fejlécállományokra vagy csak a forrásszövegekre akarjuk korlátozni. Ilyenkor egy második parancssori kapcsolóban (`$2`) megadhatjuk a fájl típusúra jellemző kiterjesztést, és nem típus, hanem név alapján végezhetjük a keresést, valahogy így:

```
find . -name '.*'$2 -exec grep -l "$1" {} \;
```

Itt a parancssorban valóban csak a keresett fájlok kiterjesztését szükséges megadnunk, a "*" előtagot már maga a program illeszti hozzá az utolsó sorban. Természetesen ha nem „belső használatra” terveztük a programot, akkor ez sokak számára zavaró lesz, hiszen a gyanútlan felhasználók önkéntelenül ki fogják írni a teljes alakot. Ez a forma azonban esetünkben már csak azért sem megfelelő, mert ha ezt a csillagot is tartalmazó kapcsolót nem zárják kettős idézőjelek közé, akkor a héj is értelmezi azt, és a csillag helyére be fogja helyettesíteni az adott könyvtár listáját.

Nagy könyvtárszerkezetek kezelése

Az imént bemutatott keresőprogramok két műveletet hajtanak végre: előállítják a megfelelő típusú fájlok neveinek a listáját, majd átadják azt a grep parancsnak. Amíg a lista nem teljes, addig a találatokat a programot futtató héj rakatározza egy belső átmeneti tárolóban. Bár ez a láthatatlan közbelső művelet normális esetben semmiféle gondot nem jelent, egy általános célú program megvalósításakor mégis jobb, ha nem építünk a felhasználó előrelátására.

És mivel minden átmeneti tár befogadóképessége véges, megtörténhet, hogy a felhasználónk olyan mennyiségű fájlban akar keresést végezni, amelyek listája már túlsordulást okoz. Ha ez bekövetkezik, akkor rendkívül érdekes helyzet áll elő. A rendszer minden próbálkozásunkra „A paraméterek listája túl hosszú” (*Argument list too long*) üzenettel válaszol, és a megadott műveletet azokon a fájlokon sem hajtja végre, amelyeknek a neveit még képes volt tárolni. Ha például sikerült létrehozunk egy olyan könyvtárat, amelyben a fájlok listájának hossza meghaladta ezt a mágikus határt, akkor még az amúgy mindenható `rm *` parancs sem működik.

Az ilyen helyzetek elkerülésére találták ki az `xargs` parancsot, amellyel úgy hívhatunk meg egy másik programot, hogy a parancssori kapcsolóit (a szükséges kapcsolókat és fájlneveket) az `xargs` szabványos bemenetére küldjük. Az `xargs` ebből és a saját parancssori kapcsolóként megadott másik program nevéből összeállít egy parancssort, és végre is hajtja azt.

A közérthetőség kedvéért lássunk egy egyszerű példát. Az `ls -l *.txt` parancs az `xargs` közbeiktatásával a következőképpen hajtható végre:

```
echo -l *.txt | xargs ls
```

Ennek így semmi értelme sem volna, ha az `xargs` nem nyújtana valami többletet. Ez a bizonyos többletszolgáltatás pedig az, hogy az `xargs` az értéklistát tetszőleges hosszúságú darabokra tudja bontani, és ezzel elkerüli a korábban említett átmeneti tár túlsordulását.

Az alapértelmezett darabolási hossz eleve úgy van beállítva, hogy idomuljon a kérdéses rendszer igényeihez. Ha tehát semmi más célunk nincs, csak hogy a héjprogramunkat különlegesen nagy bemenetek kezelésére is fel akarjuk készíteni, akkor nem kell egyebet tennünk, mint a feldolgozási sor megfelelő pontjára beilleszteni az `xargs` segédprogramot. Esetünkben ezt a program lelkét jelentő 12. sorban kell megtennünk, a következőképpen:

```
find . -type f print | xargs grep -l "$1"
```

vagy

```
find . -name '.*'$2 print | xargs grep -l "$1"
```

Figyeljük meg, hogy a `grep` futtatását most nem a `find`, hanem az `xargs` végzi. A `find` feladata csupán a fájlok listájának előállítását és kiküldését a szabványos kimenetre.

Közvetett változóhasználat

Ha egy héjváltozó tartalmához akarunk hozzáférni, nem kell egyebet tennünk, mint a neve elé egy `$` (dollárjelet) írni, és a rendszer rögtön tudja, mit akarunk. De mi történik akkor, ha a kérdéses változó neve egy másik változóban található, például azért, mert egy feldolgozási művelet sor választotta ki több lehetséges jelölt közül? A helyzet ilyenkor nagyjából olyan, mintha egy magas szintű programnyelvben mutatót használnánk, amellyel közvetett módon férhetünk hozzá a változó tartalmához.

Egy héjprogramban az efféle fogások kivitelezésére több megoldás is kínálkozik.

```
1: #!/bin/sh
2: # Közvetett változóhasználat
3: változo=13
4: változo_neve="változo"
5: parancs="echo "$$változo_neve ; eval "$parancs"
6: eval `echo "echo "$$változo_neve`
7: eval echo \$$változo_neve
8: echo ${!változo_neve}
```

Ha a Bash parancsértelmezőt használjuk és lefuttatjuk ezt a programot, négyszer fog megjelenni a 13-as szám. Más héj esetében a negyedik megoldásra (8. sor) hibaüzenetet kapunk, mivel ez a Bash egy különleges szolgáltatását használja.

A 5. sorban a változó nevét tároló változó felhasználásával egy karakterláncban egy parancssort állítunk össze, majd az `eval` segítségével le is futtatjuk.

A 6. sorban tulajdonképpen ugyanezt a megoldást használjuk, de parancsbehelyettesítés segítségével az egész szerkezetet egyetlen sorra sűrítjük össze.

A 7. sor mutatja talán a legkifinomultabb megoldást: egyszerűen egy literális `$` (dollárjelet) illesztünk be a változó nevét tartalmazó változó tartalma elé, és így adjuk át az `eval` parancsnak.

Végül a 8. sor egy különleges, csak a Bash héjnál működő megoldást mutat. A Bash eleve tartalmaz egy programozási szerkezetet a közvetett változóhasználatra, vagyis nála az alapfelszerelés része az, ami az egyszerű héjknál inkább csak ügyes trükkökkel érhető el.



Buki András (buki.andras@insilico.hu)

Körülbelül kilenc éve dolgozik Linux operációs rendszerrel. Állandó szerzőtársa Prof. Dr. H. V. Kuksinak, akivel a Duna vagy a Tisza partján szoktak az élet és a tudomány viselt dolgairól töprengeni.

Hogyan térjünk át Linuxra lépésről lépésre? (4. rész)

Tegyük, amit a többiek: webezzünk, levelezzünk kedvünkre!

Most, hogy rendszerünk hátterével többé-kevésbé tisztában vagyunk, itt az ideje, hogy valami vidámabb, látványosabb dologba fogjunk.

Nézzük meg, találjuk ki, hogy miként vehetjük igénybe az internet két leggyakoribb szolgáltatását: a levelezést és a böngészést. Gondolom, már senki sem lepődik meg, ha azt mondom, hogy igen nagy választékban találunk Linux alatt olyan programokat, amelyek segítségével mindez lehetséges. Írásunkban ezek közül a programok közül szeretnék a teljesség igénye nélkül bemutatni néhányat, s egy-egy kézzelfogható példán keresztül szemléltetni, hogyan is bírhatjuk működésre őket olyan módon, hogy minden igényt kielégítsenek.

A web és a Linux

Akár Lolkáé és Bolkáé vagy Tomé és Jerryé, úgy fonódik össze a web és a Linux fogalma is szinte a kezdetektől fogva. Rendszerünk ugyanis hálózatkezelésben mindig rendkívül erős volt, a HTTP protokoll megjelenésével egyidőben jelentek meg az első böngészők is. Manapság már ott tartunk, hogy hirtelenjében is öt-hat lényegesen különböző böngészőtípust tudok felsorolni, amelyek elérhetők Linux alá. Ezek mindegyikét használják is, ugyanis ki-ki a saját meggyőződésének engedelmessévé a számára legkedvesebb böngészőre esküszik, s alkalmazza azt minden helyzetben. Akadnak természetesen olyanok, amelyek valamiért kiemelkednek a mezőnyből: vagy azért, mert gyorsak, vagy mert pontosak (esetleg mindkettő). Jelen cikkben a két legkiemelkedőbb böngészőt szeretném bemutatni, amelyek népszerűségüknek köszönhetően az összes terjesztésben alapértelmezetten fellelhetők: a Mozilla és a Konqueror nevű programról van szó. Az előbbi a Netscape böngészőkkel mutat szoros rokonságot, de ledolgozta magáról annak kedvezőtlen vonásait, az utóbbi a KDE grafikus munkakörnyezet elterjedésének köszönheti sikerét, ugyanis annak szerves részét alkotja. Sorozatunk előző részében már említettem, hogy a Windowsban hasonlatos szemléletet követve a munkakörnyezetet jól összekalapálták egy fájl- és webböngészővel, így a mi rendszerünkön is alapértelmezetten megtalálható. Kezdjük is mindjárt ezzel az ismerkedéssel!

A Konqueror

Indítsuk el, és lássunk csodát: ez bizony pontosan ugyanúgy néz ki és működik, mint az összes többi böngésző,



1. kép Az Evolution levelezőprogram felhasználói felülete

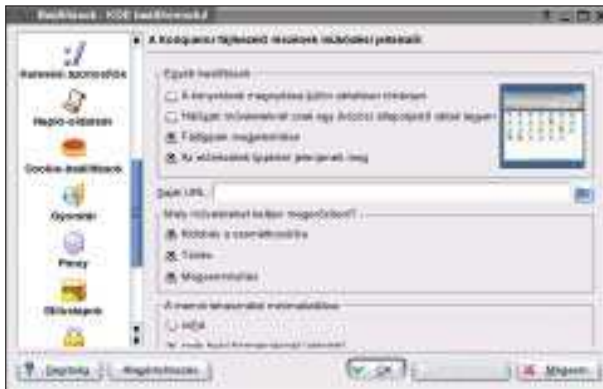


2. kép Az Evolution első indításakor megjelenő varázsló

ezért az ezzel kapcsolatos részletekre nem térnek ki bővebben. Böngésszünk vele bátran, semmiféle különleges beállítás nem szükséges hozzá. Ha az internetkapcsolatunkat telepítéskor (vagy azóta) jól beállítottuk, a világon semmi dolgunk, csak a kívánt URL-t szükséges beírni. Akkor adódhatnak bajaink, ha proxyt keresztül kapcsolódunk:



3. kép A Konqueror webböngésző



4. kép A Konqueror beállításai

ekkor természetesen be kell állítani a programot, hogy a kívánt átjárót használva jelenítse meg a tartalmat. (A proxy keresztül történő kapcsolódás egyébként a magánhasználat során igen ritka. Általában egy-egy szerkezeti egységen belül szokták alkalmazni a forgalom kézbe tartására, a letöltés gyorsítására.) A változtatásokat a **Beállítások** menü **Konqueror beállításai** menüpontjára kattintva előpattanó ablakban tehetjük meg. A megjelenő panelen az összes beállítási lehetőség elérhető, de mi most csak a proxy beállításaira összpontosítsunk. A bal oldali menüben találunk is egy ilyen nevű lehetőséget, itt választhatjuk ki, hogy a böngészőnk önműködően vegye-e észre a proxy jelenlétét, megadhatunk előre elkészített leírófájlt, vagy épp kézzel is beállíthatjuk a proxykiszolgálóra vonatkozó adatokat. Ha ez megvolna, akkor működő internetkapcsolat mellett most már aztán semmi sem állhat programunk útjába. Itt azonban még ne torpanjunk meg, ha már úgymint a beállításokról esett szó, nézzünk szét egy kicsit ebben az ablakban.

Az első néhány beállítási pont a menüben a Konquerorra mint fájlkezelőre vonatkozik. Beállíthatjuk a fájlkezelés jellemzőit, a kinézetet, hogy melyik fájl előnézetét engedélyezzük, és kezelhetjük a rendszerhez tartozó társításokat is. Ezek azt mondják meg, hogy egy adott típusú fájl milyen rendszerösszetevővel kell használnunk. A továb-

biakban testreszabhatjuk a megjelenő betűk fajtáját, méretét, a gyorsítótárat, a sűtiket, a stíluslapokat és még sok-sok egyéb apróságot is. Láthatjuk, hogy mennyivel bővebb a beállítások köre, mint más operációs rendszerek böngészőinél. Bevallom, hogy számos lehetőséget még én magam sem próbáltam ki, azonban volt már olyan hasznos kis kapcsoló, amelynek a hiánya megnehezítette volna a dolgomat. Ezért jó ez az egész, mert a fejlesztések az utóbbi időben egyre inkább abba az irányba mozdultak el, hogy az alapbeállítások minél tökéletesebbek legyenek, így nem kell vacakolnunk velük, de ha akarjuk, akkor lehetőségünk nyílik a változtatásra. Az alapbeállítások természetesen nem tökéletesek, ugyanis néhány képességre nekünk kell megtanítani a böngészőnket (szerencsére a hajlandóságot a fejlesztők beleépítették). Ilyen például a Java-kisalkalmazások (applet) futtatása vagy bizonyos változatoknál a flashanimációk lejátszása. A SuSE 9-ben ezek a kérdések meg vannak oldva, alapértelmezetten a lejátszóprogram és a Java2 futtatókörnyezet is telepítve van. Ha azonban mi egy régebbi változatot (esetleg egy másik terjesztést) használunk, vagy a későbbiek során éppen a legújabb lejátszó programot szeretnénk használni, az alábbi módon telepíthetjük a flashlejátszót.

Flashlejátszó telepítése

Látogassunk el a <http://www.macromedia.com/shockwave/download/alternates/> címre, s töltsük le a Mozilla böngészővel együttműködő linuxos flashlejátszó bővítményt (plugin). Ha ezzel megvolnánk, csomagoljuk ki a letöltött fájlt, és helyezzük el a benne található **libflashplayer.so** állományt a saját könyvtárunk **.netscape/plugins** könyvtárában. Ezután indítsuk újra böngészőt, majd a **Beállítások** ablakban a **Bővítmőmodulok** panelen kattintsunk az **Új Netscape bővítmőmodulok keresése** gombra. Ha ellenőrizni szeretnénk a munkánkat, akkor a **Bővítmőmodulok** fülön keressük meg az előbb bemásolt fájlt, s közvetlenül alatta láthatjuk, hogy a Konqueror miként ismert fel az adott bővítményt. Ezzel egyébként mást is előrevetítettünk, a böngészőnket ugyanis ezen a helyen okosíthatjuk ki a különböző modulok használatával kapcsolatban. Alapértelmezetten találhatunk itt néhányat, például a **.pdf**-fájlok megjelenítéséhez használható **ppdf** bővítményt, vagy éppen a **RealMedia**-fájlok lejátszásához szükséges modult. A különleges fájlformátumok előállítói a legtöbb esetben igyekeznek a megfelelő bővítmőmodulokat is biztosítani, így ha valaki azzal a nehézséggel kerül szembe, hogy egy honlap valamelyik tartalma valamilyen okból nem jelenik meg, kezdésként megpróbálhatja az adott bővítmény telepítésével orvosolni a hibát.

Java-kisalkalmazások futtatása

A Java-kisalkalmazások valójában weboldalakra beágyazott különleges Java-programok, ezért a működésükhöz szükségük van a Java futtatókörnyezetre. Mielőtt tehát ilyesmire adnánk a fejünket, szükségünk lesz a Java Runtime Environment nevű futtatókörnyezetre, amely tartalmazza a Java virtuális gépet, amelyen az alkalmazás futhat, ezen túlmenően pedig egy Netscape-bővítményt, de erről majd később lesz szó.

Az előkészületeket kezdjük azzal, hogy a sorozat második



5. kép A Mozilla webböngésző



6. kép A Mozilla beállításai

részében leírtaknak megfelelően a YaST csomagkezelő moduljának segítségével telepítjük a *java2-jre* nevű csomagot, ami a */usr/lib/SunJava2-1.4.2* könyvtárba fog kerülni. A könyvtár neve természetesen a birtokunkban lévő változatban szereplő JRE csomag változatszámától függően módosulhat, ám a helye a tapasztalatok szerint továbbra is a */usr/lib* könyvtár marad.

Ez után a lépés után már csak egy dolgot kell tennünk: be kell állítanunk a Konquerorban a Java virtuális gép elérési útját. Ezt a már jól ismert beállítóablakban tehetjük meg, a *Java/JavaScript* beállítási pont alatt. A *Java program elérési útja* szövegmezőbe a */usr/lib/ sunjava2-1.4.2/bin/java* szöveget írjuk, majd nyomunk *OK*-t. Előfordulhat, hogy újra kell indítani a KDE-t, hogy a változtatások érvénybe lépjenek, de utána bátran látogassunk el egy olyan oldalra, ahol Java-kisalkalmazás szerepel, hogy kipróbáljuk új lehetőségeinket. Javasolom a <http://www.iit.bme.hu/~cseb/Education/Bounce.htm> oldal megtekintését, ahol egy kis rugósinga-kisalkalmazás várja a látogatókat. Sok mindenre nem használható, de ha fut, biztosak lehetünk benne, hogy mindent jól állítottunk be. Az eljárás egyébként minden terjesztésnél ugyanez. Telepíteni kell a JRE-t, majd a *java* nevű bináris elérési útját meg kell adni a Konqueror beállításainál. Ne aggódjunk akkor sem, ha az adott terjesztésben nem található meg a JRE csomag, ez ugyanis a <http://java.sun.com> címről egy önkicsomagoló állomány formájában letölthető, amit aztán oda másolunk, ahová szeretnénk. A lényeg csupán annyi, hogy mindig tudjuk, hová telepítettük a programot, hogy utána a virtuális gép elérési útját megfelelően be tudjuk állítani.

A SuSE 9-et használóknak megint szerencsésük van, ugyanis nekik nem kell végigjárni ezeket a lépéseket, a telepítés során a YaST mindent beállít. Ha azonban 8.2-es változatot vagy annál régebbit használunk, meg kell tennünk a fenti lépéseket, azzal a plusz körültekintéssel, hogy megfigyeljük, hogy milyen nevű könyvtárba települ a *java2-jre* csomag.

Akik a fentieket olvasva elszörnyülködtek, azoknak elmondanám, hogy új böngészők használata esetén sem



7. kép A Mozilla-Mail első indításakor megjelenő varázsló

a JRE, sem a flashlejátszó telepítését nem kell megismételni, a többi böngésző is ezt fogja használni, legfeljebb egy kicsit másképp. Hogy mennyire másképp, ezt most mindjárt meg is nézzük.

A Mozilla webböngésző

Eredetét tekintve a Netscape nyílt forráskódú változata, amely jelenleg már az 1.5-ös változatnál tart. Sikerét talán annak köszönheti, hogy bár nem túl gyors, de pontosan jeleníti meg az elemeket, és a jelenleg elérhető legjobb linuxos böngésző. A HTML-oldalak nézegetésén kívül azonban másra is lehetőségünk nyílik, a program ugyanis fejlett levelezőprogramot, naptárat és egy HTML-szerkesztőt (a Netscape Composer utódját) is tartalmaz. A SuSE különböző változataiban alapértelmezetten nincs telepítve, ezt nekünk kell megtenni. Telepítsük hát a *mozilla* nevű csomagot a YaST segítségével, majd az *ALT+F2* billentyűkombináció lenyomásával a KDE futtatási ablakában adjuk ki a *mozilla* parancsot, és a böngésző pár másodpercen belül elindul.

Mik azok a közvetett hivatkozások?

A linuxos fájlrendszerek egyik sajátossága a jogosultságok kezelésén túl az is, hogy lehetőségünk nyílik hivatkozások létrehozására, amelyek mindig valamilyen állományra mutatnak. Windowsos megfelelője talán a parancsikon lehetne, de ez ebben a rendszerben sokkal általánosabban és sokkal átláthatóbban működik. Alapvetően két fajtája van: a lágy (soft) és a kemény (hard) hivatkozás; a kettő között az az alapvető különbség, hogy míg a kemény hivatkozás gyakorlatilag a fizikai adatállomány egy újabb megjelenése, addig a lágy hivatkozás csak egy mutató az eredeti fájlra. A különbség akkor látszik, ha mondjuk, megpróbáljuk letörölni az eredeti fájlt (amire a hivatkozás mutat). Lágy hivatkozás esetében a rá mutató hivatkozás érvénytelenné válik, nem érhető el többé, az állomány elveszett, ám kemény hivatkozás esetében észre sem vesszük, hogy hiányzik, mert a rá mutató hivatkozás úgy él tovább, mintha az eredeti fájl volna. (Kemény hivatkozást egyébként csak fájlokra készíthetünk, könyvtárakra nem.)

Ezeket a bizonyos lágy hivatkozásokat nevezzük közvetett hivatkozásoknak.

Indítás után megintcsak azt látjuk, hogy pontosan ugyanolyan, mint a többiek, és ugyanúgy is kell használni. Beállítási lehetőségeit tekintve sincs túl sok különbség a Konquerorhoz képest. Jó alapbeállításokkal rendelkezik, s itt is minden kis apróságot meg tudunk változtatni, a gyorsítótár beállításától kezdve a legkisebb betűméretig. Jelen böngésző esetében is ugyanaz a két kényes pont adódik: a flashlejátszás és Java-futtatás. Ha azonban végigcsináltuk az előző lépéseket, itt már gyakorlatilag alig lesz dolgunk. Kezdjük is mindjárt a flash lejátszásával. Ha bemásoltuk azt a bizonyos *libflashplayer.so* nevű fájlt a saját könyvtárunk *.netscape/plugins* könyvtárába, akkor indításkor a Mozilla felismeri a telepített bővítményt, amiről úgy győződhetünk meg, hogy a hivatkozási sorba egy különleges címet írunk: `about:plugins`. Ennek hatására egy jegyzéket kapunk a telepített és felismert bővítményekről. Ha egy sincs telepítve, egy *Default plugin* nevű bővítmény mindig akad: ez egy olyan alapértelmezett mindenese, amire a megfelelő működés érdekében van szükség. Itt kellene felfedeznünk egy Shockwave Flash nevű modult, s ha ez megtörtént, bátran kipróbálhatjuk egy flashreklámot (banner) tartalmazó oldallal.

A Mozilla és a Java-kisalkalmazások

Ha már telepítettük a JRE-t, csupán egy-két apróságot kell még megtennünk. A Mozillánál nem a virtuális gép elérési útját kell megadnunk, hanem azt a bizonyos bővítményt kell telepítenünk, amelyet az elején már egyszer említettem. A telepítés csak annyiból áll, hogy egy közvetett hivatkozást (symbolic link) hozunk létre a JRE-ben található Netscape bővítményre, amely a saját könyvtárunk *.netscape/plugins* könyvtárába kerül. Ehhez a következő parancsot kell kiadnunk:

```
ln -s /usr/lib/SunJava2-
```

```
1.4.2/jre/plugin/i386/ns610/libjavaplugin_oji.so
```

```
~/netscape/plugins/libjavaplugin_oji.so
```

Itt is fontos, hogyha nem pontosan ott szerepel az adott Netscape-bővítmény (más változat, más terjesztés), akkor írjuk át az elérési útvonalat, s vigyázzunk arra is, hogy a céldoldalon a saját könyvtárunkban létezzen a *.netscape/plugins* könyvtár.

Ha megoldottuk a hivatkozás elkészítését, újra kellene indítanunk a böngészőt, s utána a már emlegetett `about:plugins` beírásával ellenőrizzük le, hogy megtalálta-e a bővítményt, s tudunk-e már Java kisalkalmazásokat futtatni. Ha mindent jól csináltunk, akkor a bővítmények listájában rengeteg új bejegyzést találunk, amelyet az újonnan telepített modul okoz. Ezek után látogassunk el ismét a <http://www.iit.bme.hu/~cseb/Education/Bounce.htm> oldalra, hogy megnézzük, fut-e a rugós inga.

Itt ismét megjegyezném, hogy a SuSE 9-et használók tábora ismét fellélegezhet: szintúgy semmit sem kell tenniük, a telepítés után a Java futtatókörnyezet alapértelmezetten be van állítva.

Mi legyen a levelekkel?

Ennyi böngészgetés után a következő teendőnkre ugorhatunk: telepítsük, majd állítsuk be a szükséges programokat ahhoz, hogy levelezni tudjunk. A levelezés még öregebb sport az interneten, mint a web, ebből kifolyólag ez is meglehetősen elterjedt. Rendszerünk is számos olyan összetevővel rendelkezik, amelyekkel valóban játszva olvashatunk leveleket – akár szöveges környezetben is. Az előbbi vonalnál maradván itt is két elterjedt levelezőprogramot említenék: az egyik a Ximian által fejlesztett Evolution nevű Outlook-másolat, amely jóval több, mint egy egyszerű levelezőprogram – összevont naptár- és címjegyzékkezelő is egyben. Alapértelmezetten az Evolution sem része a rendszernek, így a YaST segítségével telepítsük az *evolution* nevű csomagot. A csomagnak rengeteg függősége van, mindet engedélyezzük, hogy a programunk működőképes legyen.

Az Evolution

Telepítés után a *KDE menü/Internet/Levelezés/Ximian Evolution* menüpontjának aktiválásával futtassuk a programot. Bár a levelező egész bonyolultnak tűnik első látásra (egy csomó szolgáltatása van), nem nehezebb beállítani, mint egy windowsos Outlook-ügyfelet, sőt szinte pontosan ugyanúgy kell. Ezt nem azért mondom, hogy kedvet csináljak hozzá, hanem azért, hogy bevezessem: a program ugyanolyan varázslóval rendelkezik, mint redmondí társa, s ezt az első indítás során fel is kínálja – a segítségével kényelmesen beállíthatjuk a postafiókunkat.

Az Evolution varázslója

Az előugró varázslóban lépünk egyet előre, majd adjuk meg a nevünket, elektronikus levélcímünket, ha szükségesnek látjuk, töltsük ki a többi mezőt is, majd haladjunk ismét tovább.

A kiszolgáló típusának állítsuk be a megfelelőt. Itt jegyzem

meg, hogy a legelterjedtebb a POP protokoll használata – aki otthonról használja a vállalati vagy a szolgáltatója által rendelkezésére bocsátott címet, vagy épp a Freemailről tölti le a leveleit, annak ezt kell használnia. Miután kiválasztottuk, adjuk meg a kiszolgáló címét és a bejelentkezési nevünket, majd haladjunk tovább.

A következő képernyőn a levélfogadásra vonatkozó jellemzőket állíthatjuk be: milyen időközönként ellenőrizze a postaládát, letöltés után hagyjon-e másolatot, tiltsa-e a POP3-kiterjesztéseket (alapértelmezetten nem kell hozzányúlnunk).

Ha innen is továbbmegyünk, kimenő leveleink küldésének a jellemzőit rögzíthetjük. Használjuk az SMTP protokollt, adjuk meg a szolgáltató által biztosított kimenő levélkiszolgálót, majd innen is lépünk tovább.

Most már nevet adhatunk a beállított kapcsolatnak, hogy a későbbiekben ennek alapján találjuk meg, ha módosítani szeretnénk. Ha ez az első postafiók, amit létrehozunk, hagyjuk, legyen ez az alapértelmezett.

Most következik az időzóna meghatározása: keressük meg a listában Budapestet, s ugorjunk az utolsó oldalra, majd az **Alkalmaz** gombbal mentsük a beállításokat.

A felhasználói felület

Ha már láttunk Outlookot, nem fogunk nagyon meglepődni, hogy ismeretlen felületre jutottunk, ez ugyanis a megszólalásig hasonlít rá. Feltételezem, hogy mindenki tud ilyen alkalmazással levelet írni, ezért a sajátosságok bemutatásától eltekintenek. Ami érdekes lehet, az az, hogy hol találjuk a beállításokat. Az **Eszközök** menü **beállítások** menüpontjára kattintva egy testreszabási lehetőségekkel gazdagon megtűzdelt ablakot nyitunk meg. A mindennapos használathoz természetesen itt sem kell (nem is ajánlott) semmit átállítani, de érdekes lehet megismernedni a számtalan lehetőséggel.

A Mozilla Mail

Fentebb már volt szó róla, hogy a Mozilla nem csupán böngésző: a levelezésünket is kezelhetjük vele. Az újabb Mozilla-csomagok esetében ez már külön is van bontva az alapcsomagtól, ezért nekünk is külön kell telepíteni a YaST segítségével a **mozilla-mail** nevű csomagot. A telepítés után a **KDE menü/Internet/Levelezés/Mozilla Mail** menüpontra kattintva elindul az alkalmazás. Postafiókunk eléréséhez szintúgy be kell állítani a programot, s ezúttal is egy jó szándékú varázsló siet a segítségünkre.

A Mozilla Mail varázslója

Először is maradjunk a postafiók létrehozásánál, majd a következő panelen az Evolution programnál megismertekhez hasonlóan adjuk meg a nevünket és az elektronikus levélcímünket, s haladjunk tovább.

A kiszolgálók megadása következik: adjuk meg a kimenő és a bejövő levélkiszolgálókat. A bejövő gép legyen az, ahová a belépésünk szól, a kimenő pedig legyen az az SMTP-kiszolgáló, amit az Evolution testreszabásánál is megadtunk.

A következőkben adjuk meg a kiszolgálóra érvényes bejelentkezési nevünket, s ezek után a későbbi azonosítás céljából postafiók-beállításunkat ismét nevezzük el.

Beállításainkat a befejező képernyő **Finish** gombjára kattintva menthetjük.

A Mozilla Mail felhasználói felülete

Ha az előbb az Evolutiont a windowsos Otulookhoz hasonlítottam, akkor a Mozilla Mail a windowsos Outlook Express kiköpött mása. A fenti két alkalmazásból is látszik, hogy Linuxunk mennyire hasonlatos más operációs rendszerekhez, és ha erről az oldalról nézzük, valóban nincs mitől félnie annak, aki még csak most ismerkedik a környezettel, hiszen jelentősen nem tér el az általa eddig megszokottól. Külsőségeiben semmiképp, legfeljebb szemléletében, és talán ez az, amivel többet is nyújt, mint a többiek.

Még egy kis ráadás

A levelezős rész bevezetőjében már szoltam arról, hogy leveleinket itt akár szövegesen is olvashatjuk. Létezik ugyanis számtalan terminálmódú levelezőprogram is, melyek közül a legismertebb a Washingtoni Egyetem által fejlesztett Pine nevű alkalmazás (PINE=Personal Internet News and E-mail client). Ez egy beépített szövegszerkesztővel rendelkező levelezőprogram, amely a legtöbb használatos protokollt ismeri, s abban rejlik igazi előnye, hogy terminálról bejelentkezve is lehetővé teszi a hozzáférést postaládánkhoz. Alapértelmezetten a pillanatnyi felhasználó helyi postafiókját keresi, s nyitja meg, ebből már kitalálható, hogy általában a levélkiszolgáló oldalán szokták futtatni, oda csatlakozva a bizonyos terminálprogrammal – ezáltal leveleinket távolról, központi helyen kezelhetjük.

Végszó

A fentebb felsorolt programok természetesen csak a legismertebbek, mindnek a bemutatása területi okokból elmaradt, ám ha valaki kedvet érez, nyugodtan kipróbálhat még párat. Böngészők közül az Operát tudom ajánlani, vagy a feltörekvőben lévő Gnome2 grafikus környezethez készült Nautilust, amely egyben fájlböngésző is. Ha azonban valaki levelezőkből szeretne másikat, nyugodtan nézze meg a KMail nevű KDE-s alkalmazást – ez, bár egyszerűbb működésű, mint az itt bemutatottak, talán ebből fakadóan lényegesen gyorsabb is, mint társaik. Az pedig teljesen szabadon megválasztható, hogy ki miből melyiket szeretné használni. Tárgyilagosan nézve is a böngészők közül mindegyik nagyjából ugyanazt tudja, s a levelezőprogramok levelezőrése is hasonló, bár ott a nyújtott szolgáltatások terén már van némi eltérés. Ennek ellenére az emberek sokszor inkább a meggyőződésükre – úgy is mondhatnám, hogy a szívükre – hallgatnak; tegye hát ezt a kedves olvasó is, s használja a számára legvonzóbb programokat, ha már ilyen egyszerű választani a kínálatkozó lehetőségek között.



Komáromi Zoltán (komi@kiskapu.hu)

23 éves, a BME hallgatója, mellette PHP-programozóként dolgozik. Kedvenc területe a multimédia.

Hálózatok (2. rész)

Sorozatunk e részében szó lesz az entitásokról, a csatolófelületekről, a protokollokról, a hivatkozási modellekről és sok minden másról.

Mint azt az előző részből megtudtuk, a rétegeknek az a feladatuk, hogy valamiféle feladatot végezzenek el a közvetlenül felettük álló réteg számára. Ezt úgyis mondhatjuk, hogy az n . réteg szolgáltatást nyújt az $n+1$. réteg számára.

Felmerülhet a kérdés, hogy miként nyújthat egy réteg szolgáltatást? Valójában nem maga a réteg az, amely szolgáltatást nyújt, hiszen a réteg nem kézzelfogható, inkább csak amolyan elméleti dolog. Minden réteghez tartoznak azonban olyan elemek, amelyek „fizikai” értelemben véve is munkát végeznek. Ez lehet például egy folyamat, de lehet egy eszközön elhelyezett nyomtatott áramkör is – ezek az elemek az adott réteg entitásai (entitites).

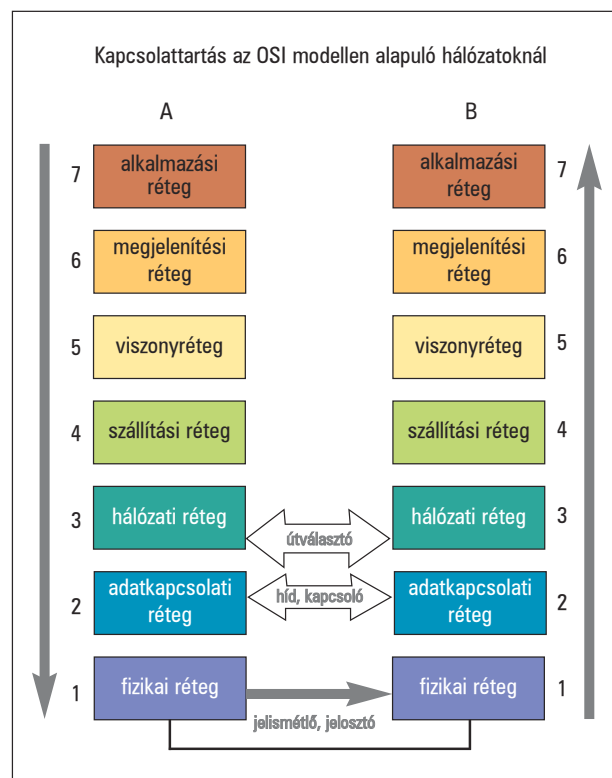
Átfogalmazva tehát azt is mondhatjuk, hogy egy adott réteg entitásai szolgáltatásokat végeznek a közvetlenül felettük lévő réteg számára. Ha két entitás ugyanabban a rétegben, ám különböző gépen helyezkedik el, akkor ők úgynevezett társentitások (peer entitied).

Az odáig rendben van, hogy a rétegek entitásai képesek bizonyos feladatokat elvégezni, ám a felső rétegeknek valamiképpen el kell tudniuk érni őket, ezért szolgáltatás-elérési pontokra (Service Access Points, SAP) van szükség, amelyeken keresztül az alsó rétegek szolgáltatásai elérhetőek. Fontos az is, hogy ezek a pontok egyértelműen megnevezhetők legyenek, ezért mindegyiknek egy egyedi címmel kell rendelkeznie.

A SAP csak első hallásra tűnhet titokzatosnak, valójában egyszerű dologról van szó. Vegyünk például egy szinte minden háztartásban megtalálható hálózatot: a vonalas telefonhálózatot. Ennél vajon mi lehet a szolgáltatási pont? Természetesen nem más, mint az a falba szerelt háromlábú csatlakozó, amihez a telefonkészülékünket kapcsoljuk. Ezek a SAP-ok egyedi címmel is rendelkeznek, amiket telefonszámoknak nevezünk.

Csomagok

Azt már tudjuk, hogy két szomszédos réteget az úgynevezett csatolófelület (interface) köt össze, de az még rejtve maradt számunkra, hogy miként is működik a rétegek közötti adatcsere. Ahhoz, hogy egy csatolófelületen adatokat adjunk át, szükség van bizonyos előzetes megállapodásokra. A rétegek közötti csatolófelületek általában a következőképpen működnek: a felső réteg entitása az alsó réteg szolgáltatás-elérési pontján keresztül az alsó réteg entitásá-



1. ábra Kapcsolattartás az OSI modellen alapuló hálózatoknál

nak egy úgynevezett csatolófelület-adategységet (Interface Data Unit, röviden IDU) ad át. Az IDU két részből áll: a vezérlőadatból és egy szolgáltatási adategységből (Service Data Unit, azaz SDU). Az SDU tulajdonképpen maga az adat, amit a hálózaton – egész pontosan a társentitásnak – el szeretnénk küldeni. A vezérlőadat azonban nem része ennek az üzenetnek, hanem magáról az átküldendő adatról árul el bizonyos dolgokat, amelyekre az alsó rétegnek szüksége lehet (vagy legalábbis megkönnyíti a munkáját). Vezérlőadat lehet például az SDU mérete.

Az SDU továbbításának módja az alsó rétegre van bízva. Ha túl nagy méretű, akkor az alsó réteg akár fel is darabolhatja, majd fejléccel ellátva darabonként küldheti tovább. A fejléc vezérlőadatokból áll és az SDU elejére illesztik. Az ilyen fejléccel ellátott darabkákat protokoll-

adategységeknek (Protocol Data Units, röviden PDU) nevezzük. Fontos, hogy a PDU-k fejléce csak a társentítá-sokat érdekli, a felsőbb rétegekhez már nem jutnak el. Az interneten áramló csomagok is egyfajta PDU-k.

A rétegek szolgáltatásai

Az alsó réteg a felső réteg szempontjából mindig arról gondoskodik, hogy az általa küldött üzenetet a társentítésnek átjuttassa. Erre azonban sokféle út kínálkozik, az egész attól függ, hogy milyen jellegű a kapcsolat. Most röviden átbeszéljük ezeket a szolgáltatástípusokat – érdemes nagyon figyelni, mivel a későbbiekben sokat fogunk rájuk hivatkozni.

Mind közül az első a kapcsolatköz-pontú szolgáltatás (connection oriented service); legfontosabb tulajdonsága, hogy az átvitel meg-kezdése előtt fel kell építeni a kap-csolatot. Miután ez létrejött, az adatok sorban továbbíthatók (ame-lyek a továbbítási sorrendben is érkeznek meg), majd ha minden kész, akkor ugyanazzal a lendület-tel le is kell bontanunk a kapcsola-tot. Ez olyasmi, mint a telefonálás: először fel kell hívni az illetőt, és miután mindent megbeszélünk vele, le kell raknunk a kagylót. Ennek az elvnek gyökeres ellen-téte a kapcsolat nélküli szolgáltatás (connectionless service). Ekkor nem kell a kapcsolat felállításával bajlódni, az adatcsomagokat egyszerűen csak a megfelelő címre kell küldenünk. Az ilyen jellegű szolgáltatásoknál általában min-den üzenet más útvonalon halad, és sohasem lehet teljes bizonyos-sággal megbecsülni az átviteli időt. Sőt olyan eset is előfordulhat,

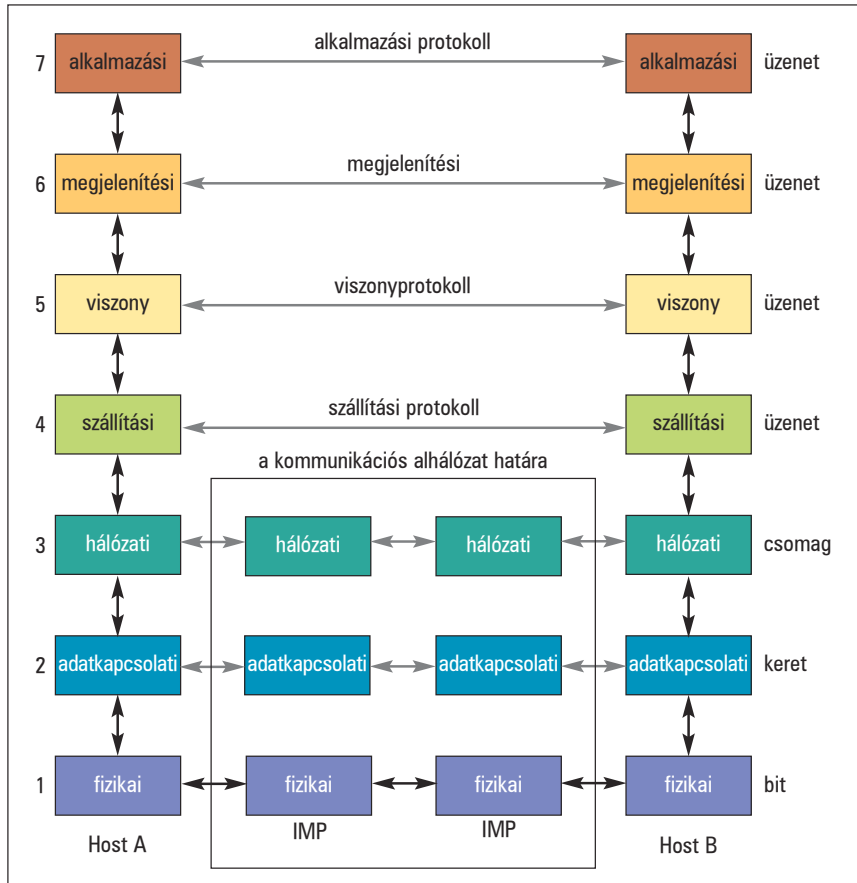
hogy a másodiknak küldött csomag előbb ér oda, mint az első. Jellemzően ilyen kapcsolat az elektronikus levél vagy a hagyományos levelezés.

A szolgáltatásokat azonban más jellemzők alapján is csoportosíthatjuk, például a minőségük alapján. Egyes szolgál-tások teljesen megbízhatóak, azaz minden elküldött adat sértetlenül és biztosan megérkezik. Akadnak azonban olyan szolgáltatások is, amelyeknek nem éppen a megbízhatóság az erősségük. Például könnyen előfordulhat, hogy bizonyos bájtok elvesznek, illetve sérülten érkeznek.

Né becsljük le azonban az utóbbi típusba tartozó szolgál-atokat sem! Egy megbízható kapcsolathoz ugyanis elenged-hetetlen, hogy a vevő fél valamiféleképpen ellenőrizni tudja a kapott üzenetek sértetlenségét; ezenkívül gondos-kodni kell a hibajavításról is. Egyes kódolási eljárások segít-ségével bizonyos hibákat a vevő önállóan is ki tud javítani, de az is előfordulhat, hogy meg kell kérnie a küldőt: ismét-lje meg az üzenetet. (A hibajavító kódokról egy későbbi

részben bővebben is szót ejtünk.) Ez azonban még mindig kevés. Ha teljesen bizonyosak akarunk lenni, akkor a vevő félnek minden kézhez kapott üzenet után egy nyugtát kell visszaküldenie. Miután a nyugta visszaérkezett, akkor a küldő biztos lehet benne, hogy az üzenet célba ért, és küldeni lehet a következő adatblokkot.

Könnyű belátni, hogy az ilyen, nyugtázáson alapuló kap-csolat jelentősen növeli az adatforgalmat. Előfordul azon-ban az is, hogy az ebből adódó késleltetés nagyobb gondok



2. ábra A rétegek ugyanazok

forrása lehet, mintha egy-egy bájtt hibásan érkezne meg (vagy esetleg meg se érkezne). A legjobb példa erre a háló-zaton keresztüli hang- és videójeleátvitel. Ha minden elkül-dött csomag után várnánk a nyugtát, a hang, illetve a kép szaggatott lenne, ami az élvezhetőség rovására menne. Ha azonban néhány képpont hibásan érkezik, az nem annyira zavaró (már ha egyáltalán feltűnik), mintha az egész filmet szaggatottan kellene végignéznünk.

Egy állomány átküldésénél azonban más a helyzet: ott egy bitet sem szabad tévednünk, a legkisebb átviteli hiba is végzetes lehet. Ebben az esetben nem tarthatjuk fényűzés-nek a nyugtázást, mivel ebben az esetben a sebesség kér-dése csak másodlagos.

Az ilyen megbízható kapcsolatalapú szolgáltatásokat két altípusra bonthatjuk tovább: az egyik ezek közül az üzenet-sorozat. Itt létezik egy előre megbeszél-t blokkméret – az üzenetet a küldő ilyen méretű csomagokra darabolja, majd egymás után elküldi őket. Ha ez a méret például 1 KB,

akkor egy 2 KB-os üzenet két darabban fog megérkezni. A másik fajta a bájtfolyam – ez olyasmi, mintha kinyitnánk egy csapatot. Két elküldött adatmennyiség között nem létezik semmiféle határ. Ez a megoldás a küldő szempontjából biztosan kényelmesebb, a vevő azonban képtelen lesz megállapítani, hogy a kézhez kapott 2 KB az két 1 KB-os vagy egy 1 KB-os üzenet. Bájtfolyamat jellemzően a terminálok használnak. A kapcsolat nélküli szolgáltatások között is van megbízható és kevésbé megbízható. Az utóbbit datagram szolgáltatásnak nevezzük, megbízhatóbb formáját pedig nyugtázott datagram szolgáltatásnak (acknowledged datagram service). Ezt akkor érdemes használni, amikor csak valami apró méretű dolgot szeretnénk átküldeni. Így ugyanis a kapcsolatot nem kell felépítenünk és lebontanunk, ami már önmagában is tovább tartana, mint maga az üzenet elküldése. Létezik még egy szolgáltatás, amelyről illik pár szót ejteni. Ez nem más, mint a kérés–válasz, amelyet az ügyfél–kiszolgáló modellben használnak. Ekkor a küldő egy kérdést küld el, amelyre a visszajelzés maga lesz a válasz, például ilyen, ha egy adatbázis-kiszolgálótól lekérjük egy meghatározott rekord tartalmát.

Alacsonyszintű szolgáltatás

Már majdnem mindent tudunk a hálózatok elvi működéséről, egyedül az alacsonyszintű szolgáltatásról nem szóltunk. Hogy ez pontosan mit takar, azt nagyon nehéz megfogalmazni (legalábbis csak hihetetlenül bonyolult lehet), pedig egyszerű dologról van szó. Az alacsonyszintű szolgáltatások nem mások, mint elemi műveletek, amelyeket a szolgáltatás végre tud hajtani és a felhasználó (vagy a felsőbb réteg) meg tud hívni. E műveletek segítségével hajtathatunk végre bizonyos feladatokat a szolgáltatással, illetve kaphatunk adatokat a társentitás állapotáról. Nézzünk egy példát! Ha mondjuk az a feladat, hogy építsünk ki egy kapcsolatot, akkor először a `CONNECT.request` műveletet kell meghívni. Ennek hatására a szolgáltatás egy kapcsolatteremtési és kérelmet küld a címzett számára. Amint az üzenet megérkezik, a vevő entitása az alatta lévő szolgáltatástól egy `CONNECT.indication` jelzést kap. Ekkor az entitás a `CONNECT.response` primitív segítségével elfogadhatja a kapcsolódási kérelmet, de akár vissza is utasíthatja. Ha mégis belemegy, akkor a küldő entitás a `CONNECT.confirm` jelzés segítségével értesülhet a kapcsolat létrejöttéről. Egyes primitívek értékekkel is bírhatnak, például a `CONNECT.request`-nek értéként meg kell adni a célállomást.

A protokoll

A protokoll és a szolgáltatás között szoros kapcsolat van, ám sokan úgy gondolják, hogy ez a két fogalom egy és ugyanaz. Mivel ez az elképzelés sajnos téves, fontos rávilágítanunk a protokollok és a szolgáltatások közötti különbségekre. Először is foglaljuk össze, hogy mi is az a szolgáltatás: ha nagyon pontosak szeretnénk lenni, akkor a szolgáltatást egy halmaznak tekintjük, amelynek az elemei a primitívek – ezek a felsőbb rétegek számára nyújtanak bizonyos szolgáltatásokat. Fontos, hogy a szolgáltatás semmit nem árul el a megvalósításról, azaz hogy miként hajtja végre ezeket a feladatokat; csupán egy (elemi) művelethalmaz, amelynek elemeit, a felsőbb rétegek hívhatják meg. A protokoll azonban nem a műveletek, hanem a szabályok

halmaza. Ezek a szabályok pontosan megmondják, hogy a csomagok milyenek legyenek, hogyan nézzen ki a fejléc stb. A protokoll a szolgáltatások megvalósításának eszköze. Fontos azonban, hogy az entitásoknak módjukban áll a protokollokat megválasztani, feltéve, hogy ezáltal nem módosulnak az alacsonyszintű szolgáltatások. Láthatjuk tehát, hogy a szolgáltatások és a protokoll fogalma élesen elkülöníthető. Voltak olyan hálózatok, amelyek nem tettek efféle megkülönböztetést. Az ilyen hálózatokon a kapcsolattartás valahogy úgy folyt, hogy a folyamat saját maga állította össze a csomagot egy memóriaterületre, majd meghívta a `Send packet` (vagy valami hasonló nevű) rendszerhívást, amelynek az értéke az adott memóriaterület címe volt. Ez azért volt rossz, mert a felhasználónak ismernie kellett az adott protokollt. Ezért, ha meg akarták változtatni a protokollokat, akkor az összes hálózati alkalmazást módosítani kellett.

Hivatkozási modellek – az OSI modell

A hivatkozási modellek nem mások, mint kézzelfogható példák arra, hogy milyen rétegekből állhatnak egyes hálózatok. Mi kettővel ismerkedünk meg: a mostani részben az OSI-val, a következőben a TCP/IP hivatkozási modellel. Az OSI modellt az iskolákban nagyon szeretik oktatni (és számonkérni), ám gyakorlati haszna nem igazán van. A TCP/IP-nek azonban annál inkább, mivel az egész internet erre épül.

Fontos az elején megjegyeznünk, hogy a TCP/IP és az OSI modell távolról sem tökéletes. Mindenesetre mindkét modellel megismerkedünk, viszont sorozatunkat az OSI modell szerint fogjuk felépíteni. Ennek okaira inkább később derítenénk fényt, most legyen elég annyi, hogy a TCP/IP-ben a rétegek sokkal összetettebb feladatokat látnak el, így az elméleti működést könnyebb vázolni, ha az OSI szerint haladunk. A továbbiakban tehát az OSI rétegeivel ismerkedünk meg.

Az OSI modell történetéről csak annyit, hogy a Nemzetközi Szabványügyi Szervezet (ISO) ajánlására tett egyféle lehetséges megoldás, amelyet első ízben sikerült nemzetközileg is elfogadtatni.

Az OSI modellnek hét rétege van. Most gyorsan átfutjuk, hogy mely rétegek milyen feladatok elvégzésére hivatottak. A legalsó a fizikai réteg, amelynek a feladata a bitek továbbítása az egyik géptől a másikhoz. Itt inkább műszaki nehézségeket kell leküzdeni, például garantálni, hogy pontosan az érkezik meg, amit átküldtünk.

Az adatkapcsolati réteg legfontosabb feladata a fizikai réteg munkájának az ellenőrzése, azaz hogy a legalsó réteg által fel nem ismert hibákat kiszűrje. Gond abból adódik, hogy a fizikai réteg csak bitfolyamokkal dolgozik, a belső adattartalommal nem foglalkozik, s így az esetlegesen megjelenő vonalzajokból fakadó hibákat nehéz kiszűrni. Az adatkapcsolati réteg ezért az elküldendő adatot úgynevezett adateretekre (data frames) darabolja fel. A keretek mérete általában legfeljebb pár kilobájt; a réteg a kereteket sorrendben továbbítja. Mivel a fizikai réteg a kerethatárokkal nem foglalkozik, az adatkapcsolati szintnek valahogy el kell tudnia különíteni az egyes rétegeket egymástól; ezért minden réteget egy különleges bitsorozat zár. Előfordulhat olyan eset is, amikor az adattag véletlenül tartalmazza ezt a bit-

sorozatot. Ilyenkor a vevő fél nem lesz képes helyesen összeállítani a kereteket, ami komoly bonyodalmakhoz vezethet. Emiatt arról is gondoskodni kell, hogy a vevő nehogy kerethatáráként fogja fel az adattagban megbúvó keretvége-jelzéseket.

Amikor egy keret megérkezik a vevő félhez, annak egy nyugtázó keret (acknowledgment frame) kell visszaküldenie. Ha a küldő fél nem kap ilyet, akkor nagy a valószínűsége annak, hogy a küldött adat elveszett, és a kérdéses keretet ismét el kell küldeni. Az élet azonban sohasem egyszerű, főleg ha maga a nyugtázó keret eltűnik. Ekkor a küldő nem értesül a keret megérkezéséről, és ismét elküldi, azaz a vevő kétszer kapja meg ugyanazt a keretet (keretkötés). Egy másik komoly gond forrása lehet az is, ha az adó gyorsabban ad, mint ahogy a vevő képes lenne feldolgozni. Ezt az esetet csak úgy lehet elkerülni, ha a réteg tartalmaz valamiféle forgalomkezelő módszert. Ilyen lehet például, hogy az adó megkérdezi a vevőt, hogy az éppen mekkora méretű szabad átmeneti tárral rendelkezik.

A következő szint a hálózati réteg, amelynek első számú feladata a csomagok célba juttatása. Ennek elengedhetetlen eleme az útvonalválasztás, azaz minden csomag számára ki kell jelölni egy utat a forrás- és a célállomás között. Az útvonalválasztásnak két fajtája van. Létezik a statikus, amikor a hálózatba bele vannak „égetve” az útvonalak, de lehet dinamikus is. Ebben az esetben az útvonalválasztás a különböző csatornák terheltségétől függően kerül kiválasztásra. A csomagok célba juttatásához az útvonalválasztáson kívül még egy bonyolult nehézséget le kell küzdeni: a hálózatok közötti együttműködés hiányát. A hálózatok ugyanis különböző csomagmérettel és protokollokkal dolgozhatnak. Ha a cél tehát egy másik hálózat, akkor gondoskodni kell a csomagok helyes átalakításáról is.

A szállítási réteg egyfajta összeköttetés a felhasználó és a hálózati eszköz között; lényegében ez határozza meg, hogy a felhasználók számára az adott hálózat milyen szolgáltatásokat képes nyújtani. Ez és a felette lévő rétegek mindegyike azonban más jelentős dolgokban is különbözik az alsó háromtól.

Ennek a szintnek a feladata hivatalosan csak annyi, hogy a fentről kapott elküldendő adatokat feldarabolja, elküldje a célállomásnak, ami hibátlanul megkapja őket. Az adatkapcsolati réteggel szemben itt az a különbség, hogy már nem kell különböző műszaki nehézségekkel foglalkoznunk. A gépszint már rejtve marad számunkra. Ezért a szállítási réteg úgymond végpontok közötti kapcsolatot hoz létre. Ez abban különbözik az előbbiektől, hogy azok mindig csak a hálózatban lévő legközelebbi szomszédal tartották a kapcsolatot. Itt azonban már nem foglalkozunk azzal a ténnyel, hogy egy-egy csomag több csomóponton keresztül jut el a célállomásig.

Mit jelent ez? A hálózatokban mindig két program tart kapcsolatot egymással: a forrásállomás programja kapcsolatot kezdeményez a célállomás valamely hasonló programjával. Miután a kapcsolat létrejött, a két program olyan módon tart kapcsolatot egymással (csomagok és vezérlőjelek segítségével), mintha a két gép közvetlen kapcsolatban állna. A hálózati alkalmazások számára rejtve marad az alkatrészes megvalósítás, és teljesen mindegy az is, hogy a csomagnak a célig hány csomóponton keresztül kellett

átküzdenie magát. Az alsó három réteg ezzel szemben mindig csak a szomszédos állomásokkal foglalkozik. Amikor egy felsőbb réteg kapcsolatba akar lépni valakivel a hálózaton, akkor a szállítási réteg úgynevezett hálózati összeköttetést hoz létre. Minden egyes kapcsolatnak külön összeköttetése van. Sőt még azt is lehet, hogy a szállítási réteg a gyorsabb átvitel érdekében további összeköttetéseket hoz létre a célállomással, ezáltal lehetővé teszi, hogy az adatokat az egyes csatornák között megossza. Mivel a számítógépek manapság többfeladatos operációs rendszereket futtatnak, a gépek egyszerre több kapcsolatot is fenntarthatnak. Ehhez azonban az is szükséges, hogy az egyik gépen futó program megmondhassa, hogy a másik gép melyik programjával akar beszélni. Hogy a szállítási réteg miként birkózik meg ezekkel a feladatokkal, arról a későbbiekben kimerítő részletességgel fogunk szólni. A viszonyréteg tulajdonképpen egyfajta irányító szerepet tölt be: egyrészt irányítania kell a kapcsolatot, mivel bizonyos esetekben nem adhat egyszerre mind a két fél, s a viszonyrétegnek kell meghatároznia, hogy mikor ki adhat. Gyakran adódik olyan helyzet is, amikor különböző kényesnek számító műveleteket nem hajthat végre egy időben mindkét fél, ezért a viszonyrétegnek egy vezérlőre kell gondoskodnia. A vezérlő egy időben mindig csak az egyik félnél lehet, és csak akkor hajthat végre bizonyos műveleteket, ha a vezérlő éppen nála van. Másrészt a viszonyrétegnek az összehangolás is a feladatai közé tartozik: a sikeres kapcsolathoz elengedhetetlen, hogy a két fél egymással összhangban legyen. Az összehangolás lehetővé teszi, hogy a kapcsolat megszakadásakor se kelljen mindent előlről kezdeni. Ha például egy fájlátvitel megszakad, akkor a kapcsolat újrafelépítésével az átvitelt onnan folytathassuk, ahol az félbemaradt.

A megjelenítési réteg a felhasználók életét hivatott megkönnyíteni. Célja, hogy a minden alkalommal (vagy legalábbis gyakran) elvégzendő feladatokat a felhasználók helyett megvalósítsa; ilyesmi például az alsó rétegek számára továbbítandó adatok átalakítása. Erre azért van szükség, mert a felhasználói programok elég bonyolult adatszerkezetekkel dolgoznak, s ezek egy az egyben nem vihetők rá a kommunikációs csatornára. A másik nem elhanyagolható dolog, hogy minden rendszer más és más kódot használ a karaktérsorozat megjelenítésére – az ebből adódó különbségeket is le kell küzdeni.

Az utolsó és egyben legfelső réteg az alkalmazási réteg. Valódi feladata az, hogy a különböző operációs rendszerek együttműködési képtelenségéből adódó nehézségeket kiküszöbölje. Gondoljunk csak a fájlátvitelre: minden operációs rendszer másként tárolja az állományokat. Egy hálózaton ugyanakkor elvárható, hogy például egy linuxos gépről egy VMS rendszerre is küldhessünk állományokat. Az alkalmazási réteghez még számos más protokoll tartozik, amelyek például a felhasználók közötti levelezést és egyéb különlegesebb feladatokat látnak el.

Gondok az OSI modellel

Sorozatunkat ugyan az OSI modell rétegeire fogjuk építeni, mindazonáltal az OSI modellel csupán annyi gond van, hogy gyakorlati alkalmazása szinte lehetetlen. Valójában még senkinek sem ment az OSI modell teljesen sikeres

megvalósítása. Ennek okai a következők:

- Először is maga a módszer rossz. Gond van a rétegek elrendezésével. A viszony és a megjelenítési réteg szinte teljesen üres (bizonyos hálózatokban el is hagyják), az adatkapcsolati és a hálózati rétegnek viszont túl sok a feladata. Ezeket talán jobb lett volna további rétegekre bontani. Az OSI egyébként eredetileg öt rétegből (a viszony- és a megjelenítési réteg nélkül) állt. A rétegek számának hétre történő módosítása az IBM-től való félelemnek tudható be. Az IBM is készített egy saját hétretegű hivatkozási modellt, amelynek a neve nemes egyszerűséggel System Network Architecture (SNA) volt. Akkoriban az IBM eléggé befolyásos cég volt, és sokan tartottak tőle, hogy erőfölényét kihasználva mindenkire rákényszeríti az SNA-t. Mivel ez az IBM saját szabványa volt, azt bármikor kedve szerint változtatgathatta volna. Ezért volt szükség az OSI-ra mint egy olyan – szintén hétretegű – modellre, amelyre egy nemzetközi bizottság felügyel.
- A másik gond az OSI-val, hogy rendkívül bonyolult és számos érthetetlen módosítás került bele az évek során: ilyenek voltak például az alsó rétegekbe beágyazott hibajavítási és címzési szolgáltatások. Sokan rámutattak, hogyha ezeket a szolgáltatásokat a felsőbb rétegek végzik, akkor nagyobb hatékonyság érhető el. Bizonyos szolgáltatásokat pedig egyáltalán nem tartalmaz a modell, például a titkosítást vagy a hálózatrányítást.
- A legeslegsúlyosabb gond azonban az OSI mögötti szemléletmód, amelynek alapján az egészet kitalálták. A gond

a következő: kapcsolatot a `CONNECT.request` primitívvel lehet létrehozni. Ehhez egy rendszerhívásnak (vagy egy osztott könyvtárban elhelyezett függvénynek) kell tartoznia, amelynek segítségével egy folyamat egy másik gépen futó folyamattal kapcsolatba tud lépni. A modell szerint ilyenkor a `CONNECT.indication` nevű primitív jelez a vevőfolyamatnak, hogy valaki kapcsolatba akar lépni vele. Ezt megszakítás segítségével érhetjük el. A hálózati alkalmazásokat azonban magas szintű programozási nyelvekben írják, és ott rendkívül kellemetlen dolog a megszakításkezelés. Sokkal jobb megoldás, ha létezik mondjuk egy `receive()` nevű rendszerhívás, amelyik a folyamatot egészen addig blokkolja, amíg valaki kapcsolatot nem kezdeményez vele. Ezzel teljes egészében kiválthatnánk a `CONNECT.indication` primitívet.

Amiért mégsem így van, az annak tudható be, hogy az OSI modellt tervezésekor a kommunikáció elvét követték. Ez valami olyasmi, mint a telefon. Amikor keresnek minket, a telefon csörögni kezd. Amikor egy számítógépen egy folyamatot keresünk, akkor az nem képes csörögni, legfeljebb megszakításokat idézhet elő. Ez a szemlélet azonban gyökeresen ellentétben áll a korszerű programozási elvekkel. Volt más dolog is, ami hozzájárul az OSI modell bukásához: megjelent a Berkeley Unix nevű operációs rendszer, amely egy ingyenes és viszonylag jól működő, az OSI-hoz hasonló hivatkozásimodell-megvalósítást tartalmazott. Ez a modell pedig a TCP/IP volt. A következő részben innentől folytatjuk.

Garzó András (garzo@elte.hu)



Linuxos kiszolgálót mindenkinek! (3. rész)

A SuSE Linux mint kiszolgáló – kisvállalati és otthoni környezetben.

Sorozatunk előző részében eljutottunk oda, hogy felhasználóinkat felvettük, csoportokba rendeztük őket, elkészítettük a rendszer alapvető biztonsági beállításait, valamint a rendszer védelme érdekében tűzfalat készítettünk. A most következő részben helyet alakítunk ki az állományok tárolására és elérhetőséget adunk a kiszolgálónkon, valamint bemutatjuk a DHCP-kiszolgálók működését, előnyeit egy vállalati rendszer működésében, valamint beállításukat a SuSE YaST modulján keresztül. Megnézünk több különböző fájlkiszolgálót, amelyek különböző operációs rendszerű ügyfelekkel tudnak együttműködni.

Fájlkiszolgálók: a Samba

Amennyiben kiszolgálónk tárhelyet szeretne biztosítani windowsos ügyfelek számára, valamint meg akarjuk oldani a windowsos ügyfelek felhasználóinak központi karbantartását, a samba csomag telepítésére lesz szükségünk. Ezt a YaST-ban a már megismert módon tudjuk elvégezni, a *Szoftver* menüponton belül a *Szoftver telepítése és eltávolítása* menüpont alatt. Mindenképp telepítsük fel a samba és a yast2-samba-server csomagot, mert ezekre szükségünk lesz a kiszolgáló kialakításánál. Amennyiben az ügyféloldalon is a Sambán keresztül szeretnénk elérni a hálózati adatokat, a samba-client csomagra, illetve a YaST-ból való telepítéshez a yast2-samba-client csomagra lesz szükségünk. Ha végeztünk a telepítéssel és a YaST modulokat is telepítettük, a *Hálózati szolgáltatások* menüpont alatt megjelenik a *Samba szerver* és a *Samba kliens* menüpont, rajtuk keresztül tudjuk majd a modulokat indítani. Ne feledjük, ahhoz, hogy a kiszolgáló a gép indításakor el is induljon, a YaST *Rendszer* menüpontja alatt található *Futásiszint-szerkesztő* modulban be kell állítani a Samba kiszolgálóhoz és ügyfélhez tartozó megfelelő futtatási szintet.

A beállítás megkezdéséhez nyissuk meg a *Hálózati szolgáltatások* menüpont alatt található *Samba szerver* modult.

A megjelenő ablakban először engedélyezni kell a Samba kiszolgálót, majd ki kell választanunk a megosztás, illetve a kiszolgáló típusát. Három lehetőség van: a PDC (Primary Domain Controller – elsődleges tartományvezérlő), BDC (Backup Domain Controller – tartalék tartományvezérlő) és az egyszerű fájl- és nyomtatómegosztás.

Ha windowsos munkaállomásaink felhasználóit központilag szeretnénk kezelni, válasszuk az *Elsődleges tartomány-*

Néhány szó a windowsos tartományvezérlőkről

A Microsoft WindowsNT Server különböző változatai által használt tartományszerkezet alapvetően olyan módon épül fel, hogy létezik egy elsődleges tartományvezérlő (PDC), ezen a gépen létre kell hozni a felhasználókat, a csoportokat, illetve a hozzájuk tartozó házirendeket, majd a felhasználó a munkaállomásról a tartományba bejelentkezve a központilag kialakított beállításokat fogja használni. Amennyiben szükséges, a rendszerben további tartományvezérlők, az úgynevezett tartalék tartományvezérlők (BDC) helyezhetőek el. A tartalékvezérlők feladata, hogy az elsődleges kiszolgáló meghibásodása esetén átvegyék a felhasználók kiszolgálását, így a BDC-k felhasználói adatbázisa teljesen megegyezik a PDC adatbázisával, amit folyamatos egyeztetéssel érnek el.

A Windows 2000 Server, valamint a Windows Server 2003 különböző változatai már egy más szerkezetet használnak, amit Active Directorynek neveznek. Ez egy faszerkezetű szervezés, ami szorosan kapcsolódik a DNS felépítéséhez. A SuSE 9.0-ban lévő 2.8-as kiadású Samba az Active Directoryt nem támogatja, ez csak a 3.0-s változattól érhető el, így most a régebbi rendszerű tartománykiszolgálókkal foglalkozunk.

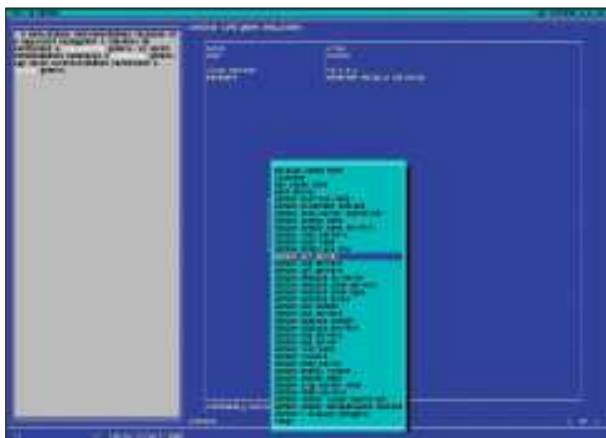
vezérlő menüpontot, ha viszont csak egyszerű fájl- és nyomtatómegosztást szeretnénk készíteni, akkor elegendő választás a *File and printer sharing* menüpont. Linuxos tartalék tartományvezérlő létrehozása nem javasolt, mivel gondot okozhat más Windows NT kiszolgálókkal való összekapcsoláskor.

Amennyiben a megfelelő kiszolgálótípust kiválasztottuk, meg kell adnunk a tartományt vagy a munkacsoportot, amelybe a kiszolgálót el szeretnénk helyezni, valamint megadhatunk egy szöveges leírást, ami alapján a felhasználóink azonosítani tudják majd a kiszolgálót.

A *Jelszóellenőrzés részletei* gombra kattintva megadhatjuk, hogy a jelszavakat milyen formában kívánjuk tárolni. Lehetőség nyílik helyi adatbázisban való tárolásra, ekkor a Samba a */etc/samba* könyvtárban létrehoz egy *smbpasswd* állományt, amelyben tárolni fogja a Samba-felhasználókat, illetve a hozzájuk tartozó jelszavakat – természetesen

kódolt formában. Lehetőségünk van LDAP-kiszolgálón tárolni a felhasználói adatokat, ilyen módon több különböző operációs rendszeren alapuló vagy egyes hálózati rendszer is egyetlen felhasználói adatbázisból dolgozhat. Az LDAP használatához meg kell adnunk az LDAP-kiszolgáló elérhetőségét, a keresési alap DN-t (Search Base Distinguished Name), amely az adatbázisban való kereséshez kiindulási pontként szolgál, valamint egy felügyeleti DN-t (Administred Distinguished Name), amelyre akkor lesz szükségünk, ha új felhasználókat és csoportokat kívánunk létrehozni.

Következő lépésben kijelölhetjük a megosztandó könyvtárak és nyomtatók listáját. Ezt első lépésben úgy tehetjük meg, hogy a *Saját könyvtárak megosztása* és a *Nyomtatók*



megosztása négyzeteket bejelöljük, és az utóbbinál a megosztandó nyomtatókat kiválasztjuk a nyomtatók listájából. A *Saját könyvtárak megosztása* arra vonatkozik, hogy a felhasználókhöz tartozó saját könyvtárak önműködően megosztásra kerüljenek-e, vagy minden felhasználónál külön döntünk ebben a kérdésben. A *Szakértői* gomb megnyomása után lehetőségünk van könyvtárak egyedi megosztására, a megosztások törlésére, elérhetőségük szabályozására. A *Hozzáadás* gombra kattintva meg kell adnunk a megosztás nevét, a leírását, valamint ki kell választanunk a megosztandó könyvtárat. Miután a megosztás létrehozásával végeztünk, a megfelelő megosztást kiválasztva szerkeszthetjük is. Itt olyan értékeket tudunk megadni, mint a könyvtár olvashatósága, a létrehozott állományok jogosultsági szintje, a könyvtár böngészhetősége stb. Alaposan nézzük át ezeket a bejegyzéseket, mert egy rossz beállítással biztonsági lyukat hozhatunk létre!

Ha végeztünk a könyvtárak megosztásának beállításával, térjünk vissza a YaST-ba, és ezután máris kipróbálhatjuk az elkészített könyvtárakat.

A Samba csomag beállításait a `/etc/samba/smb.conf` állományban találjuk, amelyben a beállítások kézzel is módosíthatóak. A módosítások létrehozása után a `/etc/init.d/samba` parancsfájlt kell futtatni a `restart` vagy `reload` kapcsolóval. Bár a Samba meglehetősen jól állítható a YaST-on keresztül, ettől függetlenül haladó felhasználóknak ajánlott a <http://www.samba.org> oldalon található magyar nyelvű leírás átolvasása, mert a Samba lényegesen több szolgálta-

tást, beállítási lehetőséget kínál, mint ami a YaST-on keresztül elérhető.

A <http://www.samba.org> oldalon megtalálható továbbá a Samba jelenleg legfrissebb 3.0-s változata is, így vállalkozó kedvű olvasóink akár ezt is megpróbálhatják működtetni.

Tipp: fájlkiszolgáló telepítésekor figyeljünk arra, hogy a megfelelő kapukat az internet felé tiltsuk le, hogy ne tudjon bárki hozzáférni az állományainkhoz.

Fájlkiszolgálók: az NFS, a hálózati állományrendszer

Linuxos munkaállomások használata esetén a Samba-ügyfélcsoomag segítségével ugyan lehetőségünk van az előbb tárgyalt Samba kiszolgáló használatára, de ez nem célravezető, mivel az SMB állományrendszeren tárolt adatok nem tartják meg a linuxos állományokra és könyvtárakra összeállítható jogosultsági beállításokat. Az NFS, azaz hálózati állományrendszer (Network File System) éppen erre kínál megoldást. Ez egy olyan hálózati állományrendszer, amely kezelni képes a Unix/Linux típusú állományjogosultságokat, így egy megfelelően kialakított hálózati rendszer esetén a felhasználók úgy látják az állományokat, mintha azok a saját gépükön lennének. Az NFS-kiszolgáló beállítására is ad eszközt a YaST, mégpedig az *NFS-Szerver* és *NFS-Kliens* modulokat, ezek szintén a *Hálózati szolgáltatások* menüpont alatt találhatók. Az NFS kiszolgálómodul indításakor a telepítő rákérdez, hogy az NFS-kiszolgáló indításra kerüljön-e, itt természetesen ezt a lehetőséget válasszuk.

A megjelenő *Könyvtárak kiajánlása* ablakban két külön listát találunk, az egyik a megosztott könyvtárak listája, a másik a könyvtárakhoz hozzáférő gépek és jogosultságaik gyűjteménye. A könyvtárak listájának bővítéséhez a *Könyvtár hozzáadása* gombot használhatjuk. Ekkor meg kell adnunk a megosztandó könyvtár pontos helyét a fájlrendszeren. Az ügyfelek listájának bővítésére a *Gép hozzáadása* gombot használjuk. A megjelenő ablakban meg kell adnunk a gép nevét különböző formátumokban, például az egyes gépeket, a hálózati csoportokat, a helyettesítő neveket (wildcards) vagy az IP-címeket, tartományokat. A *Beállítási lehetőségek* között megadhatjuk az elérés módját – mint csak olvasható, olvasható és írható –, a memóriában lévő adatok kiszolgálóra való visszaírásának módját, illetve az elérési jogosultságokat. Bővebb leírás az `exports` súgó állományban található.

Amennyiben az NFS-megosztásokat megfelelően használjuk, lehetőségünk nyílik arra is, hogy bármelyik gép elé is üljenek le a felhasználók hálózati rendszerünkben, ugyanazt vagy legalábbis nagyjából ugyanazt a felületet lássák. Így megoldható, hogy a felhasználók bármelyik gépre be tudjanak jelentkezni azáltal, hogy a felhasználói adatbázis csak egy kiszolgálón van tárolva; könyvtáraik ugyanazon az elérési úton legyenek megtalálhatóak, felhasználói beállításai változatlanok legyenek, függetlenül attól, hogy melyik gép előtt ülnek.

NFS fájlrendszereket a `/etc/fstab` állományban való regisztráció útján tudunk befűzni. A bejegyzés teljesen hasonló egy helyi fájlrendszer befűzéséhez. Természetesen a YaST erre is kínál nekünk felhasználóbarát megoldást: ha elindítjuk a *Hálózati szolgáltatások* menüpont

NFS-ügyfélmodulját, akkor a megjelenő ablakban felvehetjük a befűzni kívánt NFS állományrendszerek listáját. A **Hozzáadás** gombra kattintva megjelenik a csatolni kívánt rendszer jellemzőit tartalmazó ablak. Meg kell adnunk a kiszolgáló nevét (gépnév vagy IP-cím), a távoli könyvtárat, a helyi befűzési pontot – ami lehetőség szerint legyen ugyanott, ahol a távoli befűzési pont van –, valamint a `/etc/fstab` fájlbejegyzéseiben is megtalálható beállítási lehetőségeket, de ezt alapesetben hagyjuk az előre beállított default s értéken.

Összefoglalva, a Samba kiszolgálót olyan esetekben érdemes használni, amikor windowsos ügyfeleket szeretnénk kiszolgálni, az NFS-t pedig olyan összeállításban, amikor a munkaállomások is Linuxot futtatnak. Mindkét rendszerrel megoldható, hogy a felhasználókat egy közös adatbázisban tartjuk, beállításait pedig az úgynevezett barangoló profilok (roaming profile-ok) tárolják, így a rendszerbe kapcsolt bármelyik gépről a jól megszokott felhasználói felületükkel dolgozhatnak.

Dinamikus hálózati beállítás: a DHCP

A DHCP (Dynamic Host Configuration Protocol – dinamikus gazdagép-beállító protokoll) egy olyan protokoll, amelynek segítségével a hálózatba kötött gépek hálózati beállításait egy központi adatbázisból tudjuk vezérelni. Így a gépek IP-címe, neve és egyéb más jellemzői egy központi kiosztás részét képezik, azaz ezeket a beállításokat nem szükséges minden gépen egyesével elvégeznünk.

A címkiosztásnak két módja van: az egyik, amikor az eszközhöz állandó hálózati jellemzőket rendelünk, a másik pedig, amikor az eszköz a bejelentkezésekor a legelső szabad címet kapja. Az állandó hozzárendeléshez az szükséges, hogy minden hálózati eszközhöz egy egyedi úgynevezett MAC-cím (Media Access Control address – média elérésvezérlő cím) tartozzon – ennek alapján tudjuk beállítani a DHCP-kiszolgálónkban az adott eszközhöz tartozó hálózati jellemzőket. Ilyen eszköz lehet egy hálózati kártya, lehet egy modem, de akár egy megfelelő mobiltelefon is. A rendszer úgy működik, hogy az ügyfelek egy előre beállított időtartamra lekérlik kiszolgálótól a számukra kiosztott hálózati értékeket, majd önműködően beállítják saját magukra. A bérleti idő lejártá után az ügyfél újabb kérést küld, amire válaszul a kiszolgáló elküldi a friss értékeket. Így akár azt is megoldhatjuk, hogy menet közben, az ügyfelek módosítása nélkül cseréljük le a rendszerben a hálózati beállítást.

A SuSE 9.0-s változata rendelkezik mind a DHCP ügyfél-, mind a DHCP kiszolgálócsomagokkal, sőt a hozzájuk tartozó beállításokat a már jól ismert YaST felületen keresztül is elvégezhetjük.

Először telepítenünk kell a `dhcp`, a `dhcp-server` csomagokat. Amennyiben ügyfélként kívánunk csatlakozni DHCP-kiszolgálóhoz, természetesen a megfelelő ügyfélcsomagokat is telepíteni kell. A YaST modul a `yast2-dhcp-server` csomag telepítésével tudjuk elérhetővé tenni.

Ha a modul telepítettük, a **Hálózati szolgáltatások** menüpont alatt megjelenik a **DHCP szerver** menü, ezt elindítva kezdhetjük meg a kiszolgáló beállítását. A megjelenő **DHCP-szerverbeállítások** ablak első két jellemzője a kiszolgáló engedélyezése és a tűzfal beállítása DHCP használá-

tára. Mindkettő esetében válasszuk az **igen** értéket.

Az **Alhálózatok**-nál megadhatjuk a kiszolgált hálózati tartományokat IP-címmel és **Hálózati maszk**-kal leírva. Amennyiben a hálózati munkaállomásokhoz a MAC-cím alapján állandó hálózati értékeket szeretnénk rendelni, ezt a **Gépek** menüpont alatt tehetjük meg. A megjelenő ablakban látható a már felvett gépek listája, illetve a **Hozzáadás** gombbal új gépeket vehetünk fel. A **host**-nál meg kell adnunk a gép azonosítására szolgáló nevet, a **fixed-address**-nél pedig a kívánt IP-címet, a **hardware**-nél pedig viszont az eszköz egyedi azonosítójának típusát és azonosítószámát adhatjuk meg. A típus az esetek többségében az ethernet, de elképzelhető, hogy olyan hálózatunk van, ami token ring hálózati rétegre épül. Ennek eldöntéséhez a hálózat alapvető ismerete szükséges. Ha a típus kiválasztottuk, akkor meg kell adni a MAC-címet, ami hat részből áll – mindegyik rész 0 és 255 közötti értéket vehet fel, mindezt hexadecimálisan leírva és : (kettősponttal) elválasztva. Ezek az alapvető jellemzőkön kívül a **Hozzáadás** gombra történő kattintással minden egyes munkaállomáshoz továbbiak is meghatározhatók. Megadhatunk többek között tartomány- (domain) és netbiosnevet, DNS-kiszolgálókat, SMTP-kiszolgálókat és egyéb olyan jellemzőket, amikre a hálózati kapcsolattartás során szükség lehet. Ezeket a jellemzőket rendelhetjük egyes munkaállomásokhoz, azoknak egy csoportjához, vagy akár teljes hálózatokhoz is, s mindezt csak a megfelelő helyen szükséges beállítani. Ha ezeket a szolgáltatásokat a munkaállomások beállításánál adjuk meg, akkor ez csak az adott munkaállomásra fog vonatkozni, ha az alhálózati beállításoknál, akkor az adott alhálózatra, ha pedig a kiszolgáló beállításainál, akkor az összes olyan gépre, amelyik ettől a kiszolgálótól kapja a hálózati beállításokat.

Ha végeztünk a beállításokkal, a **Kész** gombra kattintva véglegesíthetjük őket.

A YaST-ban elvégzett beállítások a `/etc/dhcpd.conf` állományban is megtehető – itt kézzel tudjuk szerkeszteni a bejegyzéseket. Miután végeztünk a fájl szerkesztésével, a `/etc/init.d/dhcpd` restart paranccsal tudjuk a DHCP-kiszolgálót újraindítani.

Itt is fontos ellenőrizni, hogy a kiszolgálóhoz tartozó megfelelő futási szintek be vannak-e állítva, különben a gép újraindítása után előfordulhat, hogy DHCP-kiszolgálónk nem indul el.

Írásomban röviden megpróbáltam bemutatni, hogyan lehet fájl-kiszolgálót telepíteni Linux alatt, illetve azt, hogy miként tudjuk a hálózatunkhoz csatlakozó gépeket önműködően beállítani, anélkül, hogy ehhez minden géppel külön-külön foglalkoznunk kéne.

Az interneten rengeteg különböző szintű és mélységű leírás található a két fenti témában is, így javaslom, hogy keressük meg és használjuk is őket a rendszer beállítása során.



Illés Viktor (viktor@ei.hu)

23 éves, a BME műszaki informatikus szakának hallgatója, mellette weblapokkal, linuxos és windowsos rendszerekkel foglalkozik. Szabadidejét legszívesebben a szabadban tölti, teniszezik és kerékpározik.

Bevezetés a szabályos kifejezések használatába (2. rész)

Az előző lapszámban áttekintettük a szabályos kifejezések legalapvetőbb elemeit és láttunk néhány példát az alkalmazásukra is. Ebben a cikkben a kifinomultabb keresési és szövegfeldolgozási módszerekről, lehetőségekről esik szó.

Azt már tudjuk, hogy a szabályos kifejezések közönséges karakterekből (például „alma”), karakterhalmazokból ([a-z] vagy [0-9]), illetve néhány különleges jelentéssel bíró jelből (., , \$, ^) állnak össze.

Ezeket túl van az alkalmazható jelkészletnek egy olyan része is, amelynek az elemei a már ismert jelek jelentését módosítják, illetve finomítják.

Csillagok, csillagok

Talán a leggyakrabban használt jelentésmódosító jel a * (csillag), vagyis a többszöröző karakter. Ez a közvetlenül előtte álló karakter tetszőleges számú előfordulását írja elő, a nullaszorosát is beleértve. Figyeljünk a megfogalmazásra, ugyanis ez az utóbbi kitétel nagyon fontos.

Az „a” mint szabályos kifejezés pontosan egyetlen „a” betűre illeszkedik. Tegyük fel, hogy olyan szövegrészeket akarunk azonosítani, amelyek tetszőlegesen sok „a” betűből állnak, de kizárólag ilyen betűkből. A kezdőknek ilyenkor szinte kivétel nélkül az a* tűnik a helyes megoldásnak, pedig ez óriási tévedés. Mivel ez a kifejezés nulla és tetszőlegesen sok „a”-ra is illeszkedik, bármi megfelel neki, hiszen bármely szövegrészre igaz, hogy vagy van benne „a” betű, vagy nincs. Ezzel a feltétellel szűrve tehát egy szöveget annak az összes sora megfelel a keresési feltételnek. Soha ne feledkezzünk meg tehát a * (csillag) nullaszoros illeszkedést is megengedő viselkedéséről, mert ha figyelmetlenek vagyunk, esetleg teljesen értelmetlenné tehetünk egy amúgy jól megtervezett szabályos kifejezést. Esetünkben a helyes megoldás az aa* szabályos kifejezés használata, amelyben a * hatása csak a második „a”-ra terjed ki. Így a keresendő mintában legalább egy „a”-nak mindenképpen szerepelnie kell. Ugyanezt jelenti az a+\ kifejezés is, amely egy újabb jelentésmódosító karaktert, a + (pluszjelet) tartalmazza. A + jelentése azonos a csillagéval, azzal az apró eltéréssel, hogy legalább egyszeres illeszkedést ír elő. Az előtte látható \ (perjel) karakterre azért van



szükség, mert sem ez, sem az összes többi jelentésmódosító karakter nem része a szabályos kifejezések alapkészletének. Ha tehát nem betű szerinti értelemben akarjuk őket használni (esetünkben nem egy pluszjelet akarunk kerestetni), akkor ezt a megfelelő programnak jeleznünk kell. Röviden tehát a \ kapcsolja be az illető karakter különleges jelentését.

A csillagot – hasonlóan minden egyéb jelentésmódosító jelhez – nemcsak egyetlen karakterre, hanem karakterhalmazra is alkalmazhatjuk. Ha például egy szövegből ki akarjuk válogatni az olyan sorokat, amelyekben egyessel kezdődő szám is van, akkor a

```
cat szöveg.txt | grep " 1[0-9]*"
```

parancsot használhatjuk. Figyeljük meg, hogy az egyes előtt egy szóköz is van. Ha ezt nem vettük volna bele a szabályos kifejezésbe, akkor az minden olyan számra is illeszkedne, amelyben bárhol legalább egy egyes előfordul. Ez pedig nem felel meg az előzetesen kitűzött feltételnek. Ha jobban belegondolunk, a megoldás még így sem tökéletes. Ha ugyanis egy egyessel kezdődő szám a sor elejére kerül, akkor az előtt nem lesz szóköz, vagyis hiába felel meg elvileg a feltételnek, a fenti szabályos kifejezés nem illeszkedik rá. Ismét csábító a gondolat, hogy a

```
cat szöveg.txt | grep " *1[0-9]*"
```

szabályos kifejezéssel próbálkozzunk, de ismét nincs szerencsénk. A " *" kifejezésrészlet az előbb elmondottaknak megfelelően „nulla vagy több szóközt” jelöl, ami első közelítésben rendben is volna, csak hogy a szabályos kifejezések logikája szerint ennek a feltételnek egy tetsző-

leges, egyetlen szóközt sem tartalmazó karakterlánc is megfelel. Ha tehát egy sorban szerepel például a "312" karakterlánc (amely nem egyessel kezdődő szám), akkor a `grep` ezt is megfelelőnek fogja találni. Szerinte ugyanis van benne egyes (1), amit egy van több számjegy követ (" [0-9]*"), előtte pedig – szerinte – nulla vagy több szóköz van (" *"), hiszen a 3 pontosan nulla szóközt tartalmaz. A helyes megoldást egy kicsit később tudjuk csak megvalósítani.

Csoportok

Szabályos kifejezésekből kerek zárójelek segítségével csoportokat is alkothatunk. A programok ezeket a csoportokat egy egységként kezelik, és a jelentésmódosító jelek is egyben vonatkoznak rájuk. Ha például olyan sorokat keresünk egy szövegben, amelyekben az „ab” betűpáros legalább egyszer előfordul, akkor a

```
cat szoveg.txt | grep "\(ab\)+"
```

parancs segítségével szűrhetjük ki őket. Figyeljük meg, hogy az összes jelentésmódosító jel előtt használunk kellett a `\` (perjel) karaktert, jelezve, hogy nem kerek zárójelet, illetve pluszjeleket keresünk a szövegben, hanem a logikai szerkezet részeként alkalmazzuk őket. A csoportosításnak köszönhetően a `+` jelentésmódosító itt az „ab” karakterpárra vonatkozik, és nem csupán a „b” betűre.

Logikai VAGY kapcsolat

Ha két szabályos kifejezést a `|` (cső) karakterrel választunk el egymástól, azt jelezzük, hogy közülük bármelyik illeszkedése megfelelő. Kicsit rövidebben fogalmazva ilyenkor logikai VAGY kapcsolatot létesítünk közöttük. Számos esetben jól jöhet ez a lehetőség. Ha például egy keresésnél egy szó bizonyos ragozott alakjai is megfelelőek, akkor nem kell külön-külön minden egyes alakra elvégezni a keresést, hanem a ragokat egyetlen önálló csoportként megjelölve logikai VAGY kapcsolattal köthetjük őket a szótóhoz. Ez így meglehetősen száraz volt, tehát nézzünk egy példát. Ha az „asztal” szó egyes ragozott alakjait akarjuk kikeresni, akkor a következőképpen járhatunk el:

```
cat szoveg.txt | grep "asztal\(on\|hoz\|nak\|ra\) "
```

Mivel a ragok egyetlen csoportot alkotnak (kerek zárójelek), a csoport elemei között pedig logikai VAGY kapcsolat áll fenn (`|`), közülük bármelyik megfelel az illesztésnél, de egyszerre mindig csak egy.

Hasonló módszert alkalmazhatunk a korábban félbehagyott feladat megoldására is. Ott azt kell megoldanunk, hogy az egyes számjegy előtt vagy szóköz legyen, vagy az egész szerkezet a sor elején kezdődjön. Emlékezzünk rá, hogy a „sorkezdett” a szabályos kifejezésekben önálló elem (`^`), így akár egy csoport eleme is lehet:

```
cat szoveg.txt | grep "^(^| )1[0-9]*"
```

Igaz ugyan, hogy a sok `\` miatt ez a kifejezés már meglehetősen cifrának tűnik, de pontosan úgy működik, ahogy azt szeretnénk. Az egyes előtt egyetlen csoport van, amelynek

az egyik eleme egyetlen szóköz, a másik a sorkezdett, a kettőt pedig logikai VAGY köti össze. Így az illesztésnél bármelyikük megfelelő, de mindig csak az egyik. Érdekes még megjegyezni, hogy a szabályos kifejezések-nél a szó hagyományos értelmében vett logikai ÉS kapcsolat nem létezik, hiszen két logikailag különböző tartalmú szabályos kifejezés egyszerre nem illeszkedhet egy szövegrészre.

Az illeszkedések számának finomhangolása

Az ismétlődő részek számának pontos megadására a `*` (csillag) és a `+` (pluszjel) alkalmatlan, hiszen az első csak nulla vagy több, a második pedig egy vagy több illeszkedést ír elő, ami a pontosságtól meglehetősen távol esik. Ugyanakkor létezik a szabályos kifejezések jelkészletének olyan eleme is, amellyel egészen pontosan határozhatjuk meg, hogy egy karakter vagy csoport hányszor forduljon elő a keresett szövegrészben. Erre való a kapcsos zárójelek, amelyekkel három különböző illeszkedési korlátot fogalmazhatunk meg.

Ha a nyitó és záró kapcsos zárójelek között csak egyetlen számot adunk meg, akkor pontosan ennyiszere illeszkedést írunk elő. Az

```
"a\{5\}"
```

kifejezés tehát pontosan öt „a” betűre illeszkedik.

Ha a zárójelek között megadott számot egy vessző is követi, az azt jelenti, hogy legalább ennyiszere illeszkedést várunk, vagyis az illeszkedések számát csak alulról kívánjuk korlátozni. Végül, ha két számot adunk meg vesszővel elválasztva, azzal alsó és felső korlátot egyaránt meghatározunk. Mindez egyébként azt jelenti, hogy „\{0, \}” jelentése megegyezik a `*` műveletjel (operator) jelentésével, míg „\{1, \}” ugyanúgy működik, mint a `+` – csak sokkal hosszabb.

Nézzünk megint néhány példát! A befejezetlen mondatok végét jelző három pontot a következő szabályos kifejezéssel azonosíthatjuk:

```
cat szoveg.txt | grep "\.\{3\}"
```

Ha négy vagy több pontot keresünk (ilyenek például az űrlapokon a kitöltendő mezők), azt a következőképpen tehetjük meg:

```
cat szoveg.txt | grep "\.\{4,\}"
```

A legalább három, de legfeljebb nyolcjegyű számokat a következő módszerrel válogathatjuk ki:

```
cat szoveg.txt | grep "[0-9]\{3,8\} "
```

(Figyeljük meg, hogy a kifejezés elején és végén is van egy-egy szóköz.)

Néhány példa a gyakorlatból

A sorozat előző részében is bemutatunk már néhány olyan megoldást, amelyekkel bizonyos gépelési hibák javíthatók. Ott még viszonylag keveset tudtunk a szabályos

kifejezésekről, most viszont csaknem a teljes fegyverkészlet a rendelkezésünkre áll. Lássunk először két új feladatot, majd visszatérhetünk a korábban említettekhez.

Gyakori gépelési hiba, hogy a szavak közé egyetlenél több szóköz kerül, vagy lemarad a mondat végéről a pont. Írjunk olyan egyszerű programokat, illetve feldolgozási sorokat, amelyek képesek ezeket a hibákat azonosítani, valamint ki is javítják őket.

A többszörös szóközök a

```
" \+"
```

szabályos kifejezéssel azonosíthatók. Ezek egyszerűen cserélését a legegyszerűbben egy sed programmal oldhatjuk meg, valahogy így:

```
cat szoveg.txt | sed "s/ \+/g"
```

Hasonló, de egy kicsit körmönfontabb megoldást igényel a mondatvégi pontok pótlása. Ha a szövegben nincsenek tulajdonnevek, akkor a pont hiányát arról tudjuk felismerni, hogy egy kisbetűt egy szóköz, majd egy nagybetű követ. Ennek a helyzetnek a következő szabályos kifejezés felel meg:

```
"[a-z] [A-Z]"
```

Az igazi gondot az jelenti, hogy itt a cserét nem tudjuk olyan egyszerűen megfogalmazni, mint az előbb, hiszen a fenti kifejezésre illeszkedő szövegrészt nem egy pontra kell cserélni, hanem magukra a megtalált karakterekre és az őket elválasztó pontra. Szerencsére a sed-nek van memória-feladatköre. Ha a kereséshez használt kifejezésben valamit kerek zárójelek közé zárunk, akkor az így kialakított csoportok tartalmára azok sorszámával hivatkozhatunk. A megoldás tehát a következőképpen fest:

```
cat szoveg.txt | sed "s/([a-z]) \([A-Z])\1\2/g"
```

A cserélőutasítás második felében szereplő \1 most azt a részt jelenti, amire [a-z], a \2 pedig azt, amelyikre [A-Z] illeszkedett. A többi \ annak jelzésére szolgált, hogy az illető karaktert vezérlésre és nem mintaként kívánjuk használni.

Egz és ay

Cikksorozatunk első darabjában csak részben, illetve csak rendkívül körülményesen voltunk képesek elhárítani azt a nehézséget, amikor gépelés közben felcseréljük a „z” és „y” betűket. Azt már akkor is tudtuk, hogy a megfelelő eszköz a sed, amelynek keresési mintaként a megfelelő szabályos kifejezést megadva az önműködően végigolvassa a szöveget, és elvégzi a szükséges helyettesítéseket.

A gondot tulajdonképpen az okozta, hogy az elgépelés – egyszerűsége ellenére – egészen változatos helyeken bukkanhat fel. A közönséges előfordulási helyzeteken túl előfordulhat például, hogy egy mondat „Ay” karakterpárossal kezdődik, vagy „egz.”-re végződik. Ilyenkor sajnos nem igaz, hogy a hibásan gépelt rész előtt és után is szóköz van, vagyis nehezzé válik a cserélendő karakterlánc behatárolása.

Ugyanakkor most is megfogalmazhatunk olyan általános szabályokat, amelyeket aztán szabályos kifejezéssé alakíthatunk. Lássuk először az „egz” esetét. Ez kezdődhet kis vagy nagy „e” betűvel, amit szóköz vagy sorkezdet előz meg. Az „e” betűt biztosan egy „g” majd egy „z” követi. Végül vagy szóköz, vagy pont, vagy vessző következik. Hasonló a helyzet az „ay” karakterpárossal is. Ennyi csupán az eltérés, hogy itt a cserét éppen fordítva kell elvégezni, vagyis nem „z”-t kell „y”-ra cserélni, hanem fordítva. Ennyi elmélkedés után vegyünk egy nagy levegőt (a gyengébb idegzetűek inkább üljenek le), és lássuk a megoldást.

```
1: #!/bin/sh
2: # z-y csere: általános megoldás
3:
4: cat $1 | sed 's/(\ |^)\(A|a\)y\(\
↳\|\. \| , \| )\1\2z\3/g
5: s/(\ |^)\(Eg|eg\)z\(\ |\. \| , \| )\1\2y\3/g'
```

Ez első látásra meglepő. Másodikra mulatságos vagy rémisztő – a felhasználó lelkivilágától függően. Akármis is érez azonban most a nyájas olvasó, vizsgálja meg először a fenti főmedvényt abból a szempontból, hogy van-e benne olyan rész, amit nem ismer. Elvileg nem lehet ilyen. Jelenlegi ismereteink birtokában ezt már képesek vagyunk egyedül is összehozni.

Azért mielőtt a következő cikkhez lapoznánk, pusztán a biztonság kedvéért elemezzessük egy kissé művünket. Az valószínűleg azonnal nyilvánvaló, hogy itt két cseréről van szó: az első az „ay” a második az „egz” típusú elgépelésekkel foglalkozik. A keresési minta a következővel indul: \(\ |^). Némi bogarászás után azonnal látszik, hogy ez két egyszerű szabályos kifejezés (egy szóköz és egy sorkezd) logikai VAGY kapcsolattal összeláncolva, és kerek zárójelekkel csoporttá tömörítve. A helyettesítésnél az erre illeszkedő szövegrész lesz \1.

Aztán egy hasonló csoport következik: \(\A|a\). Ez tehát egy kis vagy nagy „a” betű, amire a helyettesítési programban a \2 szimbólummal hivatkozunk. Most jön az elgévelt karakter („y”), majd egy újabb logikai VAGY-okkal összetartott csoport: \(\ |\. \| , \|).

A cserét előíró program a csoportokra illeszkedő tartalmakat változatlanul hagyja, csupán az elgépel „y”-t cseréli „z”-re (“\1\2z\3”). Az utasítás után szereplő „g” azt jelenti, hogy ezt a műveletet a minta valamennyi illeszkedési helyén el kell végezni. Az 5. sorban egy újabb, az előbbihez teljesen hasonló minta, illetve program következik, amely mindenütt az „egz” jellegű hibákat javítja ki.

Ennyi logikai szépség és matematikai báj után mit is kívánhatnék annak, aki átrágtá magát ezen és az előző cikken? Haladjon bátran előre a Linux rögös útjain, és soha többé ne féljen a szabályos kifejezésektől!



Búki András (buki.andras@insilico.hu)

Körülbelül kilenc éve dolgozik Linux operációs rendszerrel. Állandó szerzőtársa Prof. Dr. H. V. Kuksinak, akivel a Duna vagy a Tisza partján szoktak az élet és a tudomány viselt dolgairól töprengeni.

Kiadás Bricolage-alapokon

Működtessük weboldalunkat a profi híroldalak modellje alapján: a Bricolage segítségével írásokat hagyhatunk jóvá, és nyomon követhetjük, hogy ki mit szerkeszt.



A ki a hálózaton híroldal vagy magazint üzemeltet, az valószínűleg hallott már a tartalomkezelő rendszerekről (CMS). A CMS program leegyszerűsíti a webhely számtalan oldalának a kezelését, ugyanis segít a felhasználók, a csoportok, a jogosultságok, a szerkesztési felelősség és az egész honlapon közös sablonok rendszerezésében. Minél több oldal van a webhelyünkön, minél több ember dolgozik a rajta, annál valószínűbb, hogy sokat nyerünk egy CMS telepítésével.

Sok cég csapott le a kínálókozó lehetőségre és kínál CMS programokat, amelyek ígéreteik szerint szinte minden szolgáltatást készen nyújtanak – a webtartalom megírását kivéve. (Bár egy-két oldal írásainak a minőségét elnézve az ember néha egészen elbizonytalanodik.) Igaz, beletelt némi időbe, de mára számos nyílt forrású CMS csomag is elérhető, amelyek a legösszetettebb honlapokkal szemben támasztott igényeket is kielégítik. Az egyik leghatékonyabb és legnépszerűbb ilyen csomag a Bricolage, amely az Apache kiszolgáló `mod_perl` modulja és a PostgreSQL relációs adatbázis-kezelő, valamint a `HTML::Mason` sablonrendszer erejét ötvözi egyetlen elegáns csomagban, lásd a <http://sourceforge.net/projects/bricolage> címen. Sorozatunk korábbi részeiben megismerhettük a Mason rendszer néhány alapbeállítását és a bizonyos események bekövetkeztekor induló riasztásokat. Egészen mostanáig azonban elhanyagoltuk minden CMS legfontosabb képességét, nevezetesen a tartalomnak a végleges

weboldalon történő megjelentetését. Ebben a hónapban a cikkek útját a megszületésüktől egészen a kiadásukig követjük végig, azt, ahogyan végighaladnak a Bricolage kiadási útvonalán.

A webhely létrehozása

Mindenekelőtt létre kell hoznunk a weboldalt, amelyen aztán a kiadandó tartalom megjelenhet. A gépemem készítettem egy virtuális webhelyet `output.lerner.co.il` néven, külön könyvtárral a hiba- és elérési naplónak. Azután a `listán` látható VirtualHost-utasításokat vettem fel az Apache beállításfájlba.

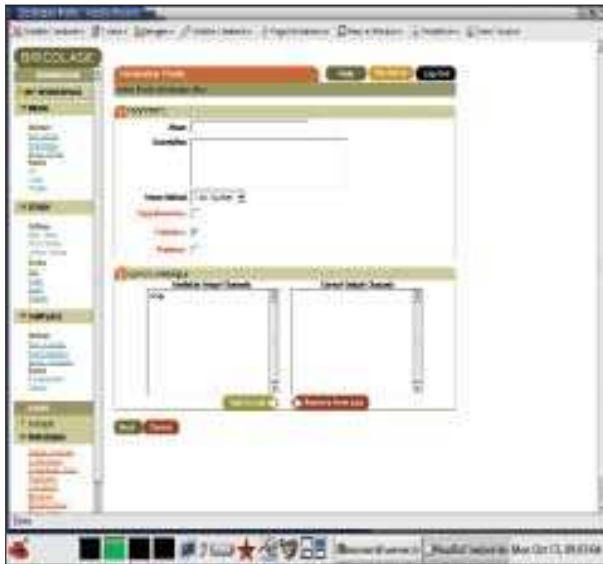
Mostantól, ha a kiszolgálóm az `output.lerner.co.il` címre érkező kérelmet fogad, az `output.lerner.co.il/www` könyvtárban fog keresni a fő dokumentumgyökér helyett, amelyet egyébként általában a `/usr/local/apache/htdocs` könyvtárban találunk.

Mielőtt a Bricolage-dzsel cikkeket tehetnénk közzé a weben, meg kell mutatnunk a CMS-nek, hogy hol találja az új fájlokat. A Bricolage rendszerében ezt a `Distributions` fejléc alatt található `Destinations` menüpontban tehetjük meg. Egy időben több kimeneti célt is készíthetünk, így egyetlen Bricolage-példánnyal több weboldalt is kezelhetünk. Ilyen eset könnyen előállhat olyan kiadónál, amelynek a csapata egyszerre több különböző újságon is dolgozik. A kimeneti célok lehetnek helyi vagy valamilyen FTP-vel elérhető távoli fájlrendszeren.

Készítsünk új kiadási célt a `New Destination` pontra

VirtualHost-utasítások az Apache-beállításfájlban

```
<virtualHost 69.55.225.93>
  ServerName output.lerner.co.il
  ServerAdmin reuven@lerner.co.il
  DocumentRoot /usr/local/apache/v-sites/output.lerner.co.il/www
  CustomLog /usr/local/apache/v-sites/output.lerner.co.il/logs/access-log combined
  CustomLog /usr/local/apache/v-sites/output.lerner.co.il/logs/referer-log referer
  ErrorLog /usr/local/apache/v-sites/output.lerner.co.il/logs/error-log
</virtualHost>
```



1. kép Új cél beállítása



2. kép A Edit Desken ellenőrizhetjük a szerkesztésre kész történeteket

való kattintással. A legtöbb kis lapnak elegendő egyetlen csatorna, ami lehetővé teszi, hogy valamennyi dokumentumot a webhely *DocumentRoot* könyvtárába helyezzünk el a lemezen. Ezt követően be kell állítanunk még néhány dolgot:

1. A weboldal és a Bricolage rendszer azonos rendszeren található-e vagy a fájlokat FTP segítségével távoli rendszerre szükséges másolni.
2. Az ehhez tartozó kimeneti csatornát személyesen elérhetővé (private) kell tennünk. Feltételezzük, hogy a Bricolage kiszolgáló és a végleges webkiszolgáló azonos számítógépen található – nagyobb méretű környezetben, különösen ott, ahol a teljesítmény lényeges, bölcs dolog a kettőt elválasztani egymástól. Az első cél képernyőjét az 1. képen láthatjuk.

Bár a Bricolage új kimeneti csatornák meghatározását is lehetővé teszi, egyelőre tartuk meg az eredeti webcsatornát. A kimeneti csatornák és a célok közötti kü-

lönbség elsősorban meglehetősen zavaró lehet. A kimeneti csatornákat képzeljük el logikai, míg a célokat fizikai célpontként, és rögtön látni fogjuk, hogy rengeteg különféle kombinációban ötvözhetjük a kettőt. Egyszerre több kimeneti csatornánk is lehet, amelyek egyetlen célra irányulnak, vagy egyetlen kimeneti csatornánk, amely több célra irányul.

Az új célunkkal kapcsolatos alapvető adatok megadása után meg kell adnunk legalább egy műveletet (move), majd meg kell határoznunk a legfontosabb részt: a kiszolgálószakaszt. A kiszolgálószakaszban adhatjuk meg, hogy melyik kiszolgáló tartalmazza a fájlokat. Mivel korábban már jeleztük, hogy a fájlokat az FTP használata helyett csak másolni fogjuk, elegendő a cél elérési útját megadnunk – ennek azonosnak kell lennie az Apache-ban megadott virtuális gép *DocumentRoot* bejegyzésével, s ezután innen érhetjük el a fájlokat.

Írás és kiadás

Miután létrehoztuk a kimeneti csatornát, visszamegyünk, feltesszük a riportersapkánkat és létrehozunk egy új cikket. (Ne felejtsük el, hogy ezeket a menüket egyébként a magazin csapatának különféle emberei vagy csoporttagjai használhatják.) Kattintsunk a *Story* menüben a *New Story* menüpontra, adjuk meg legalább a címet és a kategóriát (/), ha csak korábban nem határoztunk meg egyéb kategóriát), valamint a fontosságot (*normal*). Ezenkívül beállíthatjuk a kiadási dátumot, amely nemcsak a cikk megjelenésének dátumát adja meg a weben, hanem azt az URL-t is, amellyel azután elérhetjük. A Salon.com régebbi olvasói valószínűleg megfigyelték már, hogy a címek a kiadás dátumát is tartalmazzák; így azután nem meglepő, hogy a Bricolage, amely a Salonnál fejlesztett CMS-ből alakult ki, követi ezt a hagyományt.

Miután ezeket az alapvető adatokat megadtuk, kattintsunk a *Create* gombra. Itt lehetőségünk lesz elkészíteni a cikk szövegét, új bekezdéseket felvenni (egy időben egyet, vagy a nagy *Edit* gomb használatával üres sorokkal elválasztva). Amikor befejeztük a szerkesztést, az oldal alján található *Check In* gomb lenyomásával átküldhetjük az *Edit Desk*-re.

Sajnos a webalapú felületek természetéből adódóan, illetve a Bricolage nyújtotta rugalmasság következtében a Bricolage felülete gyakran egy kicsit szövevényesnek tűnhet, és számos különféle küldőgombbal vagyunk kénytelenek dolgozni. Szerencsére a gombok felirata egyértelmű – esetleg előfordulhat, hogy a *Edit Desk*-re továbbítás helyett mentjük a történetünket, de véletlenségből biztosan nem fogjuk törölni. Azt mondják, az új Bricolage-felhasználóknak érdemes elolvasniuk a gombok feliratát, hogy ne is feltételezzék, hogy minden *Submit* gomb ugyanúgy működik.

A cikket tehát elküldtük a *Edit Desk*-re, úgyhogy ismét lecseréljük a kalapunkat, és úgy teszünk, mintha mi lennénk a szerkesztők, akiknek a cikkeket el kell bírálniuk. Kattintsunk a *Desks* alfejléc alatti *Edit* menüpontra a *Stories* menüben, és máris olvashatjuk a *Edit Desk*-re jelenleg elküldött valamennyi cikket (2. kép). A már kiadott írások címsorában P betűt találunk. A sorból kiemelt (checked out) írások (ezek azok, amelyeket szerkesztői változtatásra



3. kép A kiadásra váró történetek átnézése a Publish Desken

jelölünk ki) átszerkeszthetők. Mi azonban most nem szeretnénk átírni a történetet, hiszen tudjuk, hogy a szerző olyan ember, aki mindig befejezi a mondatait és módszeresen ellenőrzi a munkáját. Ezért tovább is küldjük a **Publish Desk**-re, ahol a cikkeket utoljára átnézik, végül kikerülnek a webre. Ehhez a **Move To** menü **Publish** pontját választjuk, majd a **Move Assets**-re kattintunk.

Most már átléphetünk a kiadói munkafelületre, ahol az elemeket utoljára átnézve kihelyezhetjük őket a webre. Figyeljük meg, hogy a Bricolage beépített támogatással rendelkezik az olvasószerkesztéshez és a jogi felülvizsgálat kezeléséhez. Amennyiben szervezetünk felépítése az alapértelmezettnél összetettebb, további munkafelületeket is készíthetünk.

A **Publish Desk** sok tekintetben hasonló a többihez: megjegyzéseket fűzhetünk a történetekhez, megtekinthetjük a történet szerkesztői útját, vagy bepillanthatunk a naplőbejegyzésekbe. Az írásokat a **Publish Desk**-en is szerkeszthetjük: emeljük ki (check out) őket, majd kattintsunk az **Edit** hivatkozásra. A **Publish Desk** legfontosabb tulajdonsága, hogy itt egy (nem meglepően **Publish** névre hallgató) további jelölőnégyzetet és egy **Publish Checked** feliratú gombot is találunk (3. kép).

Amint rákattintunk **Publish Checked** gombra, a Bricolage megkérdezi, hogy mikor szeretnénk az írást megjelentetni. Ez nem feltétlenül egyezik meg az írás kiadási dátumával. Elképzelhető, hogy egy február 1-jei dátummal ellátott művet már január 31-én elérhetővé szeretnénk tenni; de éppúgy lehetséges, hogy csupán február 7-én szeretnénk közzétenni. A **Publish Assets** gombra kattintva megindul a folyamat, a Bricolage pedig majd jelenteni fogja, ha közreadta a kérdéses cikket (vagy történetet). Ekkor egészen biztosak lehetünk benne, hogy a dokumentumunk felkerült a korábban megadott címre – jelen esetben az **index.html** néven a **DocumentRoot** könyvtárba.

Kezelési kérdések

Ha egy nyomtatott újság vagy magazin valamilyen cikket megjelentet, többé nincs lehetőség a kiadott változat javítására. A weben azonban nem kell mást tennünk, mint hogy átszerkesztjük a kérdéses dokumentumot, és máris mindenki láthatja a változást. E technológiai lehetőséggel

kapcsolatos újságírói etikai kérdéseket sutba dobva a Bricolage a cikkek könnyű megváltoztatását és weboldalra való újraküldését teszi lehetővé.

A már kiadott cikkek megváltoztatásához válasszuk az **Active Stories** menüpontot, amely felhossa a jelenleg elérhető történeteket. Kiválaszthatjuk a kiemelő cikket és a pillanatnyi felhasználó számára szerkeszthetővé tehetjük. (A Bricolage változatkezelő rendszere biztosítja, hogy egy adott dokumentumot egy időben csak egyetlen felhasználó szerkeszthesse.) Azután átszerkesztjük a fájlt és elküldhetjük a **Edit Desk**-re, pontosan úgy, ahogy az eredeti cikkel is tettük. Szintén, akár csak az előbb, az **Edit Desk**-ről kell az írást továbbküldeni a **Publish Desk**-re, ahonnan aztán kikerülhet a webre.

Azok számára, akik ez idáig csak kis weboldalaknál dolgoztak, az ilyesmi talán egy kicsit bonyolultnak tűnhet. Végteára is nem éppen az az egyik vonzereje a webnek, hogy a dokumentum szerkesztése során a változtatás azonnal láthatóvá is válik? Tény, hogyha weboldalunkhoz tartalomkezelő rendszert használunk, bizonyos engedményeket kell tennünk. Többé már nem módosíthatjuk egyszerűen a fájlokat a merevlemezen; kénytelenek vagyunk bejelentkezni, kiemelni a fájlokat, elvégezni a módosításokat, visszahelyezni őket, és elküldeni a **Publish Desk**-re, ahol a változtatásainkat esetleg el is utasíthatják. Azonban bármilyen összetett és bürokratikus is ez a módszer, még mindig sokkal jobb, mintha számtalan ember harcolna a fájlok birtoklásáért, különböző emberek szerkesztenék ugyanazt az állományt – néha eltérő módon – egy élesben működő honlapon. A CMS-t éppen arra tervezték, hogy lelassítson minket, ahogy a biztonsági öv is némiképp gátolja a mozgásunkat. Mindkét esetben az a megkötések célja, hogy ne üssük meg magunkat. Ráadásul a bürokrácia még némileg a hasznunkra is válhat, hiszen a dokumentum valamennyi előző változata elérhető. Ha valamelyik fontos történet felét véletlenségből letöröltük volna, könnyedén visszaterethetünk a korábbi változathoz, hiszen a Bricolage tárolta az adatbázisában.

Összegzés

A Bricolage teljes körű, lenyűgöző képességekkel rendelkező, kifinomult, nyílt forráskódú kezelőrendszer. Mint azt ebben a hónapban láthattuk, Bricolage alapú weboldalt létrehozni és az Apache beállításfájlból néhány egyszerű utasítással meghatározott kiadási csatornákon keresztül írásokat közreadni nem is olyan nehéz. A következő részben bricolage-es vizsgálódásaink végéhez érünk és megnézzük, hogyan formálhatjuk igényeink szerint a kimeneti sablonokat, és az egyszerű fekete-fehér kiadványt miként varázsolhatjuk egy kicsit izgalmasabbá.

Linux Journal 2004. január, 117. szám



Reuven M. Lerner (☞ <http://www.lerner.co.il/atf>)

Nyílt forrású programokra, valamint web- és adatbázis-alkalmazásokra szakosodott tanácsadó.

Könyve, a Core Perl, 2002 januárjában jelent meg a Prentice Hall gondozásában. Reuven a feleségével és lányaival nemrég költözött Chicagóba.

Méretezhetőség: egyszerűtől az összetettig

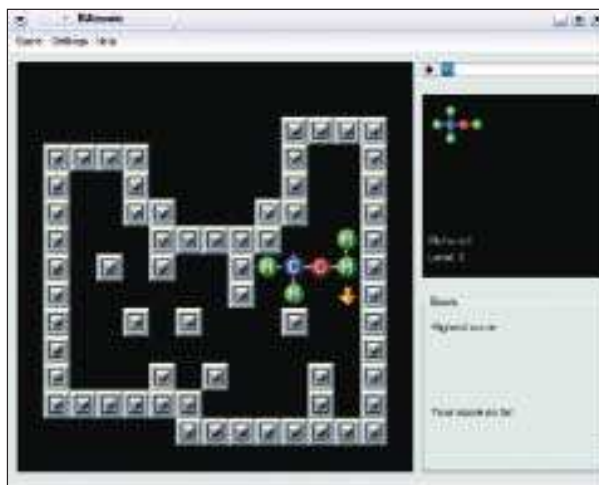
Ha már nem hoz minket lázba egy újabb lövöldözős vagy kártyajáték, szórakoztassuk magunkat virtuális alakzatokkal vagy kóstoljuk meg szakácsunk jóízű molekuláit!

Ez egy metilalkohol-molekula, François, miért kérdezed? Nem, mon ami, lehet, hogy alkohol, de egyáltalán nem hasonlít arra az alkoholra, ami a legnemesebb folyadékban, a borban található. Amaz etilalkohol lenne, és amint láthatod, annak a molekuláris szerkezete egészen más. Azt kérdezed, miért foglalkozom ezzel? Azért, mert mostani témám a rendszermag méretezhetősége. Még mindig nem érted az összefüggést? A rendszermag méretezhetősége a fejlesztés egyik magával ragadó területe, François. Ez áll a Linux jövőjének homlokterében, és az ezen munkálkodó programozók lenyűgöző dolgokat hoznak létre. Úgy gondoltam, hogy mivel a vendégeink megérkezésekor egy kis bort fogunk kóstolgatni, talán megengedhető a méretezhetőség egy másik fajtájának a vizsgálata. François, most meg miért nem figyelsz? Á, megérkeztek a vendégeink! Isten hozott titeket, mes amis, Chez Marcelnél, a kitűnő Linux-konyha, a különleges borok és az alapelemektől induló építkezés otthonában! François-val épp a mostani témánkról beszélgettünk. Foglaljatok helyet, az asztalok terítve, hűsleges pincéjük pedig szalad a pincébe, hogy felhozza nekünk a bort. Úgy gondolom, a 2001-es Châteauneuf-du-Pape a maga gazdag gyümölcsös ízvilágával és összetett aromájával éppen megfelelő lesz – a déli szárnyban találd meg, az olvasószoza bejárata mellett.

KAtomic

Amíg a borra várunk, hadd ismertesem a mai menü első fogását. Mivel egy kis bort fogunk kóstolgatni, úgy érzem, hogy a mai fogásokkal érdemes egy másfajta méretezhetőséget vizsgálnunk. Ahogy néhány programsor is alapvetően befolyásolhatja a rendszermag bizonyos szolgáltatásainak a teljesítményét, úgy néhány eltévedt atom is komoly hatást gyakorolhat a molekula tulajdonságaira. Gondoljunk például az etilalkohol és metilalkohol közötti különbségre. A legtöbbünk már évek óta nem foglalkozik a kémiával, hacsak a főzést nem soroljuk ide, non? A KDE programcsomag részeként találunk egy KAtomic nevű programot, amely *Andreas Wüst* munkáját dicséri (1. kép). A *Games* (Játékok) menüpont alatt találjuk meg, a többi stratégiai vagy fejtörő játék között. Ha játszottunk már a Sokobannal, némi hasonlóságot

vehetünk észre. A KAtomic alapelve egyszerű: hozzunk létre valamilyen összetettebb dolgot. Kezdetben van egy molekulánk a jobb oldali panelen, amelynek az alkotóelemei szét vannak szórva. A TAB billentyű lenyomásával



1. kép

Molekulák felépítése atomról atomra a KAtomic segítségével



2. kép A Construo élvezetessé

varázsolja a lehetetlen formák létrehozását

(vagy az egérrel kattintva) kijelölünk egy atomot, amelyet a kurzorbillentyűk valamelyikével hozhatunk mozgásba. Ezután az atom mindaddig mozog, amíg valamilyen akadályba – a falba vagy egy másik atomba – nem ütközik. Ezzel a módszerrel az egyszerűből összetett dolgot hozhatunk létre, emellett nagyszerű lehetőség arra is, hogy valamivel elüssük a szabadidőnket.

Construo

Amilyen egyszerűen használjuk a programkódot arra, hogy egyre összetettebb programokat hozzunk létre, úgy haladunk a molekuláktól a látható szerkezetek felé. A ma esti menünk következő fogását egy kicsit nehezebb bemutatnom, és nem tagadom, hogy első látásra voltak vele kapcsolatban bizonyos kételyeim. A neve Construo, és őszintén állíthatom, hogy hihetetlenül vonzó kis programról van szó.

Ingo Ruhnke programja tulajdonképpen egy kétdimenziós tervezőprogram, amelynek révén az általunk megadott pontokon egymáshoz kapcsolt pálcák és rugók segítségével objektumokat hozhatunk létre a képernyőn, amelyeket ezután „futtathatunk” – ami annyit jelent, hogy érvényesülni hagyjuk rajtuk a valós fizikai erőket. A cél stabil alakzat létrehozása, ami sokkal nehezebb, mint első hallásra gondolnánk. Ha az alkotásunk inogni kezd és végül a földre zuhan, a szerkezetet újra kell gondolnunk és módosítanunk kell. A Construo honlapjának egyik megjegyzésében ragaszkodnak hozzá, hogy a program pillanatnyi állapotában nem egy játékról van szó, de én ezzel nem értek egyet. Ennek a gondolkodtatva szórakoztató programnak a beszerzéséhez a <http://www.nongnu.org/construo> weboldalra kell ellátogatnunk.

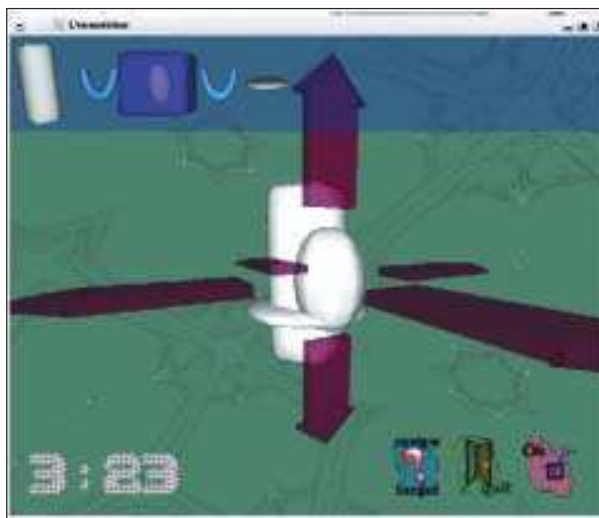
A Construo lefordításához egyszerű kicsomagoló-, beállító- és fordítóműveletekre van szükség. Igen, mes amis, ez a hagyományos ötlépes kicsomagoló-lefordító eljárás:

```
tar -xzvf construo-0.2.2.tar.gz
cd construo-0.2.2
./configure
make
su -c "make install"
```

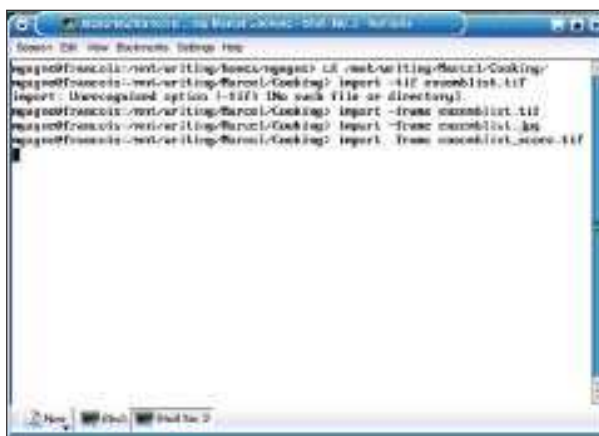
A `make install` parancs futtatása a `/usr/local/games` könyvtárba telepíti a programot. Ebben a könyvtárban a program két változatát találjuk meg: `construo.x11` és `construo.glut` néven. A SuSE 8.2 változatát használó noteszgépen, amelyen a Construo-t próbálgattam, az X11 változat képe élesebbnek, a szöveg tisztábbnak tűnt (2. kép).

A program első indításakor tiszta lappal indulunk. A program ablakának felső részén néhány billentyűparancs látható. A jobb és bal oldalon vezérlőgombokat találhatunk. Ha kilépéskor félbehagyunk egy munkát, a Construo a következő indításakor visszatér a befejezetlen tervre. Ha François-hoz hasonlóan tartunk a teljesen tiszta lappal való kezdéstől, kattintsunk a **Load** (betöltés) gombra, és keressük meg az **Examples** (példák) könyvtárat, ahol olyan előre-gyártott szerkezeteket találhatunk, amelyek módosításával elég magabiztossá válhatunk ahhoz, hogy teljesen előlről belevágjunk a tervezésbe.

A képernyőn bal egérr kattintással hozhatunk létre új



3. kép Összetett formák létrehozása egyszerű elemekből az Ensemblist CSG-jében



4. kép A Play gomb megnyomásával belépünk az Ensemblist különös virtuális világába

pontot, ezután mozogunk a kurzorral a másik végpont (vagy csatlakozási pont) helyére, és itt kattintsunk újra a bal gombbal. Egy idő után úgy fogjuk érezni, hogy az alkotásunk elég stabil, ekkor a **Run** (futtatás) gombra kattintva hozhatjuk mozgásba a fizika törvényeit és figyelhetjük meg, hogy ezek hatására miképpen viselkedik a szerkezet. Első próbálkozásunk eredménye minden bizonnyal meghajlik, csavarodik, inog, majd végül ledől – amit nagy élvezet lehet végignézni. A Construo lehetővé teszi az objektumok mozgásba hozását is (nézzük meg a kosárlabdás-példát), akadályok (ütközők) elhelyezését és egyéb módszerek használatát, amely felszabadítja a bennünk lévő az alkotószellemet.

A most épp lerombolt állapotban lévő alkotásunk módosításához és megerősítéséhez kattintsunk az **Undo** (visszavonás) gombra, mire az újra visszanyeri korábbi pompáját. Ha beleununk az alkotásba és készen állunk arra, hogy egy újba kezdjünk, a látvány törléséhez nyomjuk meg a C billentyűt, és egy új szerkezettel törjünk még nagyobb magasságokba.

Már az is elég érdekes, amikor szilárd alapra építkezünk,

de próbáljuk ki ugyanezt három dimenzióban nulla gravitáció mellett is! Ezt az Ensemblist nevű programmal tehetjük meg, amely a 2003-as párizsi játékprogramozó kiállításra íródott. Ez is egyike azoknak az egyszerű gondolatoknak, amelyek végül sokkal nagyszerűbbnek bizonyulnak. Mindössze csak a képernyőn bemutatott alakzatot kell egyszerű geometriai formák használatával megalkotni. Az összeszereléshez egyszerű programozói és matematikai alapelemeket: a logikai függvényeket, a halmazuniót és a metszetképzést használhatjuk.

Talán elfelejtettem megemlíteni, hogy az alakzatok háromdimenziósak, és egy összekapcsolt virtuális térben lebegve bármelyik tengely körül elforgathatóak. Ezt az alkotók rixed és dom CSG, vagyis constructive solid geometry (szerkezeti térgeometria) kifejezéssel illetik (3. kép).

Ensemblist

Az Ensemblist forráskód formájában tölthető le a <http://www.nongnu.org/ensemblist> címről. Miután a `tar -xvzf ensemblist.tgz` paranccsal kicsomagoltuk és a `cd ensemblist` utasítással átváltottunk a forrás könyvtárára, egyszerűen a `make` parancs kiadásával végezhetjük el a fordítást. Ennek befejeztével adjuk ki a `su -c 'make install'` utasítást. A program futtatásához az `ensemblist` szót kell begépelnünk a parancssorba – a program alapesetben a `/usr/local/games` könyvtárba települ. Az ezután következő dolgok elsősre egy kicsit furcsán hatnak. A program ellátogat az Ensemblist weboldalára, s ezután egy folyamatábra felett lebegve találjuk magunkat (4. kép), amelyen a felhasználói nevünket és a pillanatnyi pontszámunkat látjuk (ami ekkor 1). A bevezető kép alatt két lehetőség közül választhatunk: **OK** és **Quit** (kilépés). Az egér segítségével az egérmutató fel és lefelé történő mozgatásával közelíthetjük és távolíthatjuk a látványt. Kattintsunk az **OK**-ra: ezzel a választással juthatunk el a játék beállításaihoz, ahol a szabad játék (free play) és a verseny (campaign) közül választhatunk – az utóbbi előre meghatározott szinteken vezet keresztül minket. Valószínűleg a versennyel szeretnétek majd kezdeni. És esetleg François-t is meg szeretnétek kérni arra, hogy az erőgyűtéshez töltsé újra a borospoharaitokat. A játék indulásakor kijelölt objektumunk egy fraktálfalakal határolt gömb alakú térben jelenik meg. Kattintsunk a jobb alsó sarokban lévő **Play** ikonra, és már indul is a játék. A bal alsó sarokban egy számláló méri az eltelt időt. Az alapelemek, amikből a végső formát kell összeállítanunk, geometriai alapformák, amelyek a bal felső részen láthatóak. Ezeket az unió műveleti jel (operator) köti össze. Rájuk kattintva az unió (U alakú jel), a metszet (fordított U) és a különbség (mínuszjel) közül választhatunk. A jobb alsó sarokban új ikonok jelennek meg, amelyekkel az eredeti célformát újra megnézhetjük, kilép-

hetünk, vagy alkotásunkat befejezettnek nyilváníthatjuk. Ennyi az egész. Mozgassuk a formákat a középső területre vagy onnan ki, a szabályoknak megfelelően. Most arra is rájöhettünk, hogy az egyszerű is milyen bonyolult lehet. Az Ensemblist nagyon furcsa játék és ez a furcsasága okozza, hogy képtelenek leszünk abbahagyni. A későbbi játékindításoknál használhatjuk a `--no-net` kapcsolót, mivel a szintek már a gépünkre letöltött állapotban vannak. Mielőtt a következő alkalommal a **Play** gombra kattintnánk, vegyük szemügyre az **Editor** (szerkesztő) feliratú gombot. Igen, ha elég erőt érzünk hozzá, létrehozhatjuk a saját szintjeinket is, hozzájárulva ezzel az Ensemblist-univerzum építéséhez.

Mon Dieu! Úgy tűnik, megint sikerült egy estének a végére érniünk. Ötletet nyerve azokból az összetett csodákból, amelyeket egy egyszerű elvekre épülő programmal létrehozhatunk, van esetleg olyan köztetek, aki a tehetségét egy olyan szerkezet létrehozásában is kamatoztatni tudná, amellyel időt nyerhetünk? Az idő kiterjesztését tekinthetjük úgy is, mint a méretezhetőség egy új szintjét. Arra mindenesetre van még időnk, hogy még egy pohár bort elfogyassunk. Lazuljatok el, fejezzétek be, amin éppen dolgoztok, François pedig kitölti nektek az utolsó pohárkával. Ha már túlságosan elpilledtetek, örülni fogtok neki, hogy az a szint, ameddig a borotokat emelnetek kell, a szátoknál van. Tréfálok, mes amis, élvezzétek csak az italt. Egészségetekre a következő vizontlátásig! A votre santé! **Bon appétit!**

Linux Journal 2004. január, 117. szám



Marcel Gagné (mggagne@salmar.com)

Mississaguában, Ontario államban él.

Ő a szerzője a Kiskapu kiadásában tavaly szeptemberben megjelent Linux-rendszerfelügyelet (ISBN 96-9301-40) című könyvnek.

KAPCSOLÓDÓ CÍMEK

Construo

➔ <http://www.nongnu.org/construo>

Ensemblist

➔ <http://www.nongnu.org/ensemblist>

KAtomic

➔ <http://games.kde.org>

Marcel borlapja

➔ <http://www.marcelgagne.com/wine.html>





Conflict Freespace: The Great War

A Volition.inc az egyik cég a játégyártók közül, amelyik lépéseket tett a Linux felé, és lehetővé tette, hogy alkotásai a linuxosoknak is élményt nyújtsanak.



Acég történetének legnagyobb sikere a Descent Freespace, egy kétrészes sorozat, amelynek még a küldetéslemezei is az eladási listák élvonalában szerepeltek. És most úgy döntöttek, hogy a linuxos fejlesztők számára is elérhetővé teszik a forrást, hogy a játékot átültetve mi is hatalmas sebességgel repülhessünk egy csatahajó felett, és az ellenséges vadászokkal élethalálharcot vívva győzelemre segítsük az emberiséget. A történet szerint – amelyet egy nagyon ütős intró vezet be – az emberek és a vasudani faj tizennégy éve háborúban áll egymással. Ekkor kerülünk be kezdő pilótaként a földi erők kötelékébe. A három (egyébként mellőzhető) kiképzőpálya után a Galatea csatahajóra jutunk, itt fogunk szolgálni. Hamarosan azonban feltűnik egy harmadik faj is, a shivanok, akik hatalmas erővel törnek be a szektorokba és mérnek csapást az emberi és a vasudani erőkre. A háború folyamatosan bontakozik ki. Több küldetés könnyűnek tűnik, de folyamatosan nehezebbé válnak: az elején még csak aszteroidamezőben vadászunk le a vasudani kötelékeket vagy egy kis felderítő hajót védünk, esetleg rakományt kell elfoglalnunk, vagy egyszerűen csak egy űrcsatában kell foglyul ejtenünk egy áruló tisztünket. A végére azonban a háború kiszélesedik a három faj között, így számtalanszor fogunk kereskedelmi útvonalat védeni, rombolókat, cirkálókat kísérni, hatalmas űtközetekben harcolni, és még számtalan izgalmas feladat vár ránk. A küldetések helyenként igencsak megemelik az adrenalin-szin-



tet, és bizony, ez a játék azon ritka programok közé tartozik, amelyeknél valóban benne érezzük magunkat a történetben, tehát nem csak váltakozó bitfolyam az alaplap vezetőkei között.

Telepítés

A Linux alá történő telepítéshez szükségünk lesz a játék lemezeire (2 CD) és a telepítőállományra. Az utóbbit a <http://www.icculus.org/~ravage/freespace> weboldaltól lehet letölteni (az 56 CD. Magazin/Jatek könyvtárban is megtalálható). (Jó tudni, hogy a <http://www.icculus.org/~ravage> oldalról több más nagysikerű játék telepítője is letölthető. További érdekességek a <http://www.icculus.org/~talyor/freespace> weboldalon találhatóak. A játégyártó hivatalos oldala pedig a <http://www.volition-inc.com/fs/> címen érhető el.

Ha letöltöttük a telepítőt, csak el kell indítanunk, úgy, hogy a játék első korongja benne legyen CD-meghajtóban. A telepítője teljesen kulturált: grafikus és hihetetlenül egyszerű felrakni vele a játékot. Egyetlen hibát találtam benne,

a filmek és egyes hangadatok között. A telepítés során ugyan felajánlja a fájlok felrakását, de ez a Linux-terjesztéstől függően eléggé kiszámíthatatlan (szerintem olykor még a csillagállás is befolyásolja), és vagy nem teszi fel a filmeket, vagy bekéri a második korongot is. A telepítés során tehát ezt a lehetőséget érdemes bekapcsolni, és kézi ellenőrzést végezni.

Aki egyszer már végigjátszotta a játékot, az tudja, hogy igazán az első CD filmjei a jók és az outro, a többi hanyagolható. Tehát bátran másoljuk fel a filmeket tartalmazó könyvtárat a játék gyökérkönyvtárába. Olykor az is előfordult, hogy egyes hangfájlcsomagokat is kihagy. Mindenképpen azt javasolom, hogy folytassuk le a telepítést, aztán a CD-ről kézzel másoljunk fel minden nagy adatállományt. Természetesen azt, ami már fent van, nem érdemes felülírni. Ha ezzel is végeztünk, kezdődhet a játék!

Itt a 3D, hol a 3D?

A játék használ 3D-s megjelenítést, de nem kötelezően. A windowsos

változattal ellentétben itt nem kell beállítani a *Setup* menüben, hogy van-e 3D-s kártyánk, önműködően érzékeli azt, és annak megfelelően indul el. Ezt nekem rendesen megcsinálta egy Woodoo3 és egy GeForce2 esetében is. A rendszerigény viszont érdekes: a hivatalos adatok szerint 3D-kártya nélkül Pentium 166 MHz kell neki, 32 MB memória



kíséretében. 3D-kártya megléte esetén viszont elég egy Pentium 100 MHz. Nos, ezt cáfolnám, tapasztalatom szerint célszerű, ha van egy Pentium 233 MHz-esünk, de a 32 MB memória valóban elég

neki. (Természetesen a több: jobb.) A játék támogatja a botkormányhasználatot, de mint tudjuk, ezzel Linux alatt gondok adódhatnak; egész pontosan elég nehéz beállítani őket, a drágább, programozható darabokat pedig egyenesen reménytelen. De semmi komoly gond nincs, a játék jól játszható egerrel, amihez akár a billentyűzetet is segítségül



hívhatjuk. Egyébként érdemes elidőzni az *Options* menüben, mert a játék és a billentyűzet beállítása is bámulatosan testreszabható.

Fokozható? Igen!

Talán nem vagyok egyedül: ha eljutok a játék közepéig, teljesen a hatása alá kerülök. A játékot végigjátszása után sem érdemes letörölni. Igaz ugyan, hogy összesen egyetlen kampányt tartalmaz, amely 35 küldetésből áll, és kizárólag az emberek oldalán szállhatunk harcba, viszont számtalan bővítmény létezik hozzá.

A játék ugyanis tartalmazza a FRED pályaszerkesztőt, amely csak a „másik” rendszerben működik jól, viszont a Wine segítségével Linux alatt is életre kelthető, és kisebb küldetéseket végre lehet vele hajtani. Továbbá a fent említett weboldalról letölthető a küldetéslemez, a Silent Treath telepítője, de rengeteg a „Fun site” is az interneten. Innen tucatszám hozzá tudunk pályához, küldetéshez, sőt akár teljes kampányhoz is jutni. Csak az ilyen oldalak adataival megtölthető egy

CD. További jó hír, hogy a folytatás, tehát a Freespace2 telepítője is készen áll, vagyis játszható Linuxon. Ebben az esetben azonban érdemes a rendszerigényt újrafogalmaznom, ugyanis itt már a Pentium 300 MHz és a 64–128 MB memória is elkél.

Természetesen jobb eredmény érhető el, ha a játékot futtató gép tudja az MMX-et. 3DNow- és MMX2-támogatást azonban ne várjuk el tőle, ugyanis a játék megjelenésekor ilyen rendszerek még nem léteztek, és ezen a javítások sem változtattak. A javasolt gépigényt viszont olyan szempontok alapján jelöltem meg, ahol a játék több mint kielégítően, szinte teljesen hibátlanul és a legkisebb élvezhető 800×600-as felbontásban fut. Hozzáteszem: nyugodt lélekkel használhatjuk az 1024×768-as felbontást is, a sebességben nem fogunk komoly változást tapasztalni. Érdekes kitérni arra is, hogy az eredeti játék alapértelmezettként egy „setup” menüvel indult. Itt ezzel nem fogunk találkozni: a program mindent, amire a játékhoz szüksége van, önműködően próbál megtalálni, ezek kézi beállítására nekünk nincs is lehetőségünk. A saját tapasztalatom azonban az, hogy ez az eljárás legfeljebb szokatlan lesz azoknak, akik eddig más rendszeren játszottak, viszont ebben az esetben semmi hátránya sincsen, hogy ez az indítómenü nem létezik. Ismét felhívnam viszont a figyelmet, hogy a teljes játék bámulatos mértékben beállítható, ami igaz csak előnyös lehet. Ugyanis számos küldetésben olyan helyzetekbe fogunk kerülni, amikor szinte zongorázni kell a billentyűzeten, és még csak ekkor számít az, hogy mennyire „kézenfekvő” egy-egy parancs vagy manőver, esetleg valamilyen műszer megtekintése.

További jó játékot és kellemes időtöltést kívánok ezzel a kiváló programmal!



Dancsok „strogg” Zoltán

(strogg@mail.tvnet.hu)

Jelenleg technikai szerkesztőként dolgozik a BME-OMIKK-nál, ahol oktat is. Emellett egyetemi képzésben vesz részt, programozó matematikus szakon. Négy éve foglalkozik Linuxszal. Szabadidejében operációs rendszereket gyűjt és weblapot vezet.

© Kiskapu Kft. Minden jog fenntartva

Krystal Drop

A Krystal Drop egy olyan, a Tetrishez hasonló játék, ahol számos „kristályt” kell egy sorba rendezned. Itt azonban a sorok függőlegesek, és a játékosnak



az azonos színű kristályokat kell a függőlegesen elrendezett „kristályok” köré rendeznie. Bonyolultabb a dolog, mint amilyennek hangzik. Egy menet körülbelül egy percre tart, de lehet, hogy neked kellene többet gyakorolnod. A futtatáshoz szükséges: libSDL, libpthread, libSDL_image-1.2, libSDL_ttf-2.0, libGL, libSDL_mixer-1.2, libxml2, libz, libstdc++, libm, libgcc_s, glibc, libX11, libXext, libdl, libjpeg, libpng, libfontconfig, libvorbisfile, libvorbis, libogg és libmpeg. [☛ http://krystaldrop.sourceforge.net](http://krystaldrop.sourceforge.net)

Cacti

A Cacti hasonlít az MRTG-hez, de könnyebben beállítható és felügyelhető, és alapértelmezés szerint nem csak a hálózati forgalom ábrázolására használható. Egyes Cacti-felhasználók még további mentek: ábrázolni tudják a webkiszolgáló-találatokat, a névfeloldási (Bind) statisztikákat, továbbá az



időjárást és még ki tudja, mit. A Cacti honlapja mindent elmagyaráz. Néhány rövidke parancsfájl segítségével te is biztosan tudod majd használni bármilyen adat megjelenítésére. Előfeltételei: rrdtool, MySQL, webkiszolgáló, PHP SNMP-vel és MySQL-támogatással, snmpd, Perl, illetve egy grafikus webböngésző. [☛ http://www.raxnet.net/products/cacti](http://www.raxnet.net/products/cacti)

w3m

Három évvel ezelőtt számos jó programot mutattam be, de szerintem a szövegalapú böngésző w3m magasan a többi felett végzett. Mivel a rendszerek nagy többsége ma kereteket (frame) és képeket használ, szövegalapú böngészővel netezni frusztráló élmény. Bár lassú hivatkozások esetében vagy X nélküli rendszereken jó, mégsem ez az eszményi. Mindezek ellenére a w3m jól kezeli a kereteket, sőt többet is mutat a képernyőn. Emellett az egyéni beállításoktól függetlenül néhány képet is látni enged. A régebbi Lynxet már rég lecseréltem a w3m-re. Futtatásához szükséges: libgc, libpthread, libm, libnsl, libgpm, libncurses, libssl, libcrypto, glibc, libstdc++, libdl és libgcc_s. [☛ http://w3m.sourceforge.net](http://w3m.sourceforge.net)

Simple Document Management System

Számos vállalatnak, de néha magán-személyeknek is szükségük van dokumentumarchiválási rendszerre. Ez a dokumentumkezelő rendszer MySQL segítségével valósítja meg a tárolást, és különböző hozzáférési szintet biztosít annak, akinek kell. Az egyes dokumentumokhoz való hozzáférés megtagadható, vagy be lehet állítani csak olvasásra, módosításra vagy mindenhatóra (ez szinte bármit megenged). Működéséhez MySQL, PHP, webkiszolgáló és webböngésző szükséges. [☛ http://sdms.cafego.net](http://sdms.cafego.net)

mlter-sender

A mlter-sender a Sendmail mlter bővítménye, amely az üzenet fogadása közben csatlakozik a küldőhöz. Ha a küldő a feladó állítólagos elsődleges levélkiszolgálóján nem található, az üzenet szinte biztosan levélszemét, és a program elutasítja. Ennek a bővítménynek köszönhetően jelentős mértékben csökkent nálam a levélszemét mennyisége, és noha a „fegyverkezési verseny” nem ért véget, még mindig felvehetek néhány mlter-bővítményt. Jelenleg mlter-sendert és SpamAssassint futtatok a mlteren keresztül. A program működéséhez Berkeley DB, Sendmail, mlter, libsntert és glibc szükséges. [☛ http://www.snert.com/Software/mlter-sender](http://www.snert.com/Software/mlter-sender)

Atanks

Ez a játék, illetve különböző változatai a legősibb számítógépes játékok közé tartoznak. Emlékszem egy ehhez hasonlóra, ahol két tüzéségi ágyú köpdösött egymásra nagyméretű golyókat. De még ennél is régebben, amikor lyuggatott szegélyű papírra nyomtattam és egy 75 baudos modemmel csatlakoztam a hálózatra,



volt egy játék, amelyben a gravitációs erő és a szél erősség megadása után kellett megpróbálnom kilőni a gép ágyúját, még mielőtt az lőtt volna ki engem. Bár fényévekkel az imént leírt kezdetleges játékok után készült, az Atanks ugyanezen az elven működik, csak jobb fegyverekkel és pajzsokkal. Futtatásához liballeg, libpthread, libXext, libX11, libdl, libstdc++, libm, libgcc_s és glibc szükséges. [☛ http://atanks.sourceforge.net](http://atanks.sourceforge.net)

Asymptopia Memóriaajáték

Akarsz számítógépezés közben tanulni? Ez a program előre meghatározott időintervallumként kérdéseket tesz fel neked, hogy próbára tehesd a tudásodat, miközben letöltésre vársz vagy az interneten böngészsz. Új kérdések készítéséhez a LaTeX-re lesz szükséged, de segítségül szolgálhatnak hozzá a példakérdések. Futtatásához szükséges: libXm, libXpm, libXext, libXt, libSM, libICE, libX11, libasymptopia, libm, libstdc++, glibc, libgcc_s, libXp, libdl, libtiff, libpng12, libjpeg, libz, illetve LaTeX. [☛ http://www.asymptopia.com](http://www.asymptopia.com)



David A. Bandel
(dbandel@pananix.com)
Jelenleg Panamában él, Linux- és Unix-tanácsadással foglalkozik.

Társzerzője a Que

Special Edition: Using Caldera OpenLinux című könyvnek.