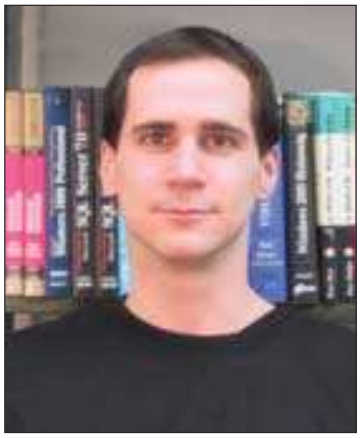
Beköszöntő



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

Elkezdődött az új év. Pontosabban hamarosan elkezdődik, ugyanis a lap nyomdába adása még a 2003-as év végén történt. Az új évet pedig igyekszünk az elmúlt év tapasztalatai fényében, újításokkal fűszerezve, érdekes olvasmányokkal köszönteni.

Sokan kérték, hogy a Linuxszal most ismerkedők számára több cikket hozzunk le. Ennek fényében a Dobbantóban több sorozatunkat is folytatjuk – mind a SUSE, mind a Debian rendszerekkel ismerkedők találnak érdekes cikkeket. Természetesen ez nem jelenti azt, hogy csak kezdő cikkekkel foglalkozunk az új évben! Ebben a számban elsősorban a rendszergazdák találnak csemegéket: IPv6-os DNS-kiszolgálók, rendszermagmodulokat kezelő keretrendszer, kiszolgálók telepítése, műsorszórás és így tovább.

Azért ne aggódjanak a programozás hívei sem, hiszen folytatjuk a QT-sorozatot, foglalkozunk a héjprogramozással, belekukucskálunk az új rendszermag API-jai közé is. Azért mindenképpen fontos, hogy kiemeljek egy cikket: a Scribusról szólót. Nagyon nagy rést tölt be (reményeim szerint hamarosan) ez a program. A tördelőprogramok világában egyszerűen nem volt használható linuxos termék (ezt a kijelentést annak tudatában tettem, hogy a TeX és a Lyx segítségével sokan könyveket is készítenek). Ha tényleg kiforrottá válik ez a program, immár a teljes DTP-folyamatot meg lehet gyorsan és hatékonyan oldani Linux alatt.

Jó olvasást kívánok tehát mindenkinek, és ha a kedves Olvasónak akad egy kis ideje, kérem, ragadjon billentyűzetet, és írja meg, melyik cikk tetszett neki!

Programvadászat

Gentoo Linux-csomagtelepítés

Most, hogy a múlt hónapban megjelentetett Gentoo Linux második korongját is közreadjuk – ami nem a programok forrásait tartalmazza, mint azt az elmúlt számban írtam, hanem előre fordított csomagokat –, leírom hogyan is lehet ezeket a csomagokat telepíteni.
Akinek még nincs telepítve a Gentoo Linux, viszont szeretné a gépén látni, az lapozzon a 24. oldalra, és a leírás segítségével telepítse fel. Ezután lapozzon ismét ide, és telepítse fel a kívánt programokat.
Az előre lefordított csomagokat az `emerge -k csomagnév` paranccsal telepíthetjük. Ahhoz, hogy a második korongon lévő csomagokat is használni tudjuk, először is fűzzük be a korongot:

```
mount /dev/cdrom /mnt/cdrom
```

ezután másoljuk be a csomagokat a megfelelő könyvtárba:

```
cp -a /mnt/cdrom/packages/*
➥/usr/portage/packages/*.
```

Grafikus felületre vágyunk? A következő parancs kielégíti ezt a vágyunkat:

```
emerge -k xfree
```

Ha ezzel elkészültünk és már nagyon hiányzik a KDE-felület, adjuk ki az

```
emerge -k kde
```

parancsot, és hamarosan használatba vehetjük ezt a memóriazabáló, ámde annál csinosabb grafikus munkakörnyezetet.
Gentoo Linuxunk kiválóan alkalmas mindenféle alkalmazás futtatására,


1.

mint azt *1. képünk* is mutatja. A csomagok között megtalálhatunk olyan programokat, mint az xmms, xcdroast, OpenOffice.org, Mozilla, KDE, Gnome és az Evolution.

Scribus

A Scribus egy Linux alatt is használható DTP-program Bővebben a 18. oldalon kezdődő cikkünkben olvashattok róla. A korongon helyet kapott a program forrása: az 1.0.1-es megbízható és a 1.1.3-as fejlesztői változat. A magyar

2.

elválasztási szótár és honosítás, az angol leírás, és a ***python-scripting*** bővítmény (plugin) szintén helyet kapott a CD-ROM ***Magazin/Scribus*** könyvtárában.

Rendszermag

Megjelent a 2.4.23-as megbízható rendszermag, illetve a 2.6.0-test11 is. Mindkettő felkerült a korongra, a ***Rendszermag*** könyvtárba.

Böngészők

Az általam mostanában használt böngésző a Mozilla alapjaira épülő Mozilla Firebird, ami egy igen jól használható és testreszabható, gyors böngésző. Ahhoz, hogy ebből mindig a legújabbat tudjam használni, nem Debian-csomagból telepítem, hanem a ***tar.gz*** fájlokat használom a frissítéshez és a telepítéshez egyaránt. A ***/opt*** könyvtárba csomagolom ki, ezután létrehozok egy közvetett hivatkozást (symbolic link) a ***/usr/local/bin*** könyvtárba, hogy bárki minden további nélkül el tudja indítani, ne kelljen keresgélnie a rendszerben a parancsfájl után. Egyszerű a program frissítése is: egyszerűen csak letörlöm a régit, és a helyébe másolom az újat. A felhasználók beállításai megmaradnak, mert ezzel a módszerrel őket sem bántjuk.
Nagyon hasznos és kedves tulajdonsága ennek a böngészőnek, hogy a reklámcsíkokat (banner) kitilthatjuk a „látómezőnkből". A nemkívánatos reklámcsíkon jobb egérkattintással előbukkanó menüben válasszuk a ***Block Image from***

3.

kiszolgálónév pontot. Arra azonban ügyeljünk, hogyha a kiszolgálónév megegyezik az általunk böngészni kívánt kiszolgáló nevével, akkor bajban leszünk, ha az adott kiszolgálóról kitiltjuk a képeket: a weboldalhoz tartozó képek sem jelennek majd meg, ami ugye számos bonyodalom és nehézkes böngészés forrása lehet.
A másik böngésző szintén ennek a családnak a tagja, sőt ez az őse a Firebirdnek is: a Mozilla 1.6a változatát adjuk közre. Mint a neve is mutatja, ez még igencsak fejlesztői változat, úgyhogy csínján bánjon vele, aki próbálgatja – mindenesetre érdemes egy próbát tenni vele.
A harmadik böngésző a méretével és a gyorsaságával szerzett magának hírnevet az informatika és a web világában: ez az Opera. Ennek 7.23-as változata kerül fel a korongunkra a magyarításával egyetemben. Érdemes a ***Bongeszok/Opera*** könyvtárat tüzetesen átnézni, mivel szinte mindenki megtalálhatja magának az előre elkészített és a rendszeréhez, a csomagkezelőjéhez a legmegfelelőbb csomagot.
Hibátlan telepítést és jó kísérletezést!

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

Vállalati Zaurus

Minden jel arra mutat, hogy a Sharp hamarosan – várhatóan 2004 első negyedében – piacra dobja SL-6000 jelzésű Zaurus modelljét. Az elsősorban vállalati piacra szánt, megjelenésében is módosított gépbe 400 MHz-es Intel XScale processzor és várhatóan összesen 128 MB memória kerül. Az SL-6000 megőrzi elődjei rejtett billentyűzetét, ugyanakkor azoknál nagyobb méretű és a külső fényviszonyoktól függetlenül jobban látható, 640×480 képpont felbontású kijelzője lesz, képes lesz 802.11b szabvány szerinti és Bluetooth alapú hálózati kapcsolatok létesítésére, valamint nagyobb kapacitású akkumulátort kap. Ára egyelőre ismeretlen.
➲ http://www.zaurus.com

Szemétgyűjtés IBM módra

Az IBM bejelentette Asset Recovery Solutions szolgáltatásának a bővítését. A szolgáltatás a kis- és középvállalkozásoknak környezetkímélő módon segít megszabadulni használaton kívüli számítástechnikai eszközeitől. Az IBM – nem IBM-eszközökre is kiterjedő – szolgáltatása valójában több szolgáltatást is magába foglal. A cég az elavult eszközöket készpénzért vásárolja fel, ha azok eladhatónak bizonyulnak. A túl öreg eszközöket kedvezményes áron veszik át és szállítják el, majd a környezetvédelmi előírásoknak megfelelő módon gondoskodnak további sorsukról, az adathordozóktól való biztonságos szabadulást pedig az adatfelülírás-szolgáltatás teszi lehetővé.
Az elavult számítástechnikai eszközök tárolása, kezelése világszerte egyre nagyobb gondot jelent – anyagi szempontból a vállalatoknak, egyébként pedig a környezetvédőknek, hiszen a nyomtatott áramkörök, tintapatronok, egyéb alkatrészek többek között különféle nehézfémeket tartalmaznak, amelyek az emberi szervezetbe kerülve rákkeltő hatásúak.
Az IBM hazánkra sajnos nem terjesztette ki szolgáltatását – vagy ha igen, elég jól titkolja –, de a helyzet szerencsére így sem teljesen reménytelen.
A HP Takeback program keretein belül a HP tintapatronokon már környezettudatos módon adhatunk túl, az egyéb elektronikai hulladékokat pedig a – például Budapesten minden kerületben megtalálható – lakossági hulladékudvarokban rakhatjuk le.
➲ http://www-1.ibm.com/financing/dispose

Bye-bye, vonalkód!

A Xybernaut és az IBSS a jövőben közösen kínál RFID (rádiófrekvenciás azonosító) rendszereket. A Xybernaut hordható gépeivel és köréjük épülő alkalmazásaival, míg az IBSS Synapse nevű elosztott, valós idejű alkalmazás-keretrendszerével járul hozzá az együttműködéshez. A két cég szállítmányozási, raktározási területre kínál RFID alapú adatgyűjtő és nyilvántartó megoldásokat.
Az RFID alapú azonosítás – a jó öreg vonalkód rovására – a közeljövőben várhatóan robbanásszerűen fog elterjedni. Az apró elektronikus címkék a termékekbe építve vagy a csomagokra ragasztva gyors, emberi beavatkozás nélküli, és nem mellesleg olcsó azonosítást tesznek lehetővé. Megjelenésük a kiskereskedelemben lesz a leglátványosabb, hiszen az üzletekben egyszerűbbé és gyorsabbá válik a számlázás, sőt előfordulhat, hogy a jövőben pénztárakra sem lesz szükség, elég lesz a bevásárlókocsit eltolni a megfelelő leolvasókapu előtt, a vásárlás összegét számítógép vonja majd le az ügyfél számlájáról.
Az RFID-címkék ugyanakkor számos jogvédelmi szervezet ellenérzését kiváltották. Segítségükkel ugyanis könnyedén nyomon lehet követni nemcsak a termékek útját, de a vásárlási szokásokat is, akár személyre lebontva. Az adatvédelmi kérdések megoldása – amellett, hogy az újdonság üzleti bevezetését aligha fogja hátráltatni – a jövő gondja, az ötletek között önmegsemmisítő és törölhető címkék egyaránt szerepelnek. Akit az adatvédelmi elvek kevésbé zavarnak, az jó üzletet csinálhat, ha rááll a témára: amerikai becslések szerint egy-egy elosztóközpontnak az új megoldásra történő átállítása több tízmillió forintos beruházást jelent, és az üzletek felszerelése is hasonló nagyságrendű költséggel jár.
➲ http://www.rfidjournal.com

Neked szemét, nekem jó

A Telus International figyelemre méltó tanulmányt tett közzé a levélszemét-üzletágról. Az írásból érdekes adatok derülnek ki, illetve kiváló összefoglaló munka több részterület – honnan veszik a levélszemetet küldők a címeket, milyen védekezési lehetőségek vannak – tekintetében. Mint a tanulmány rávilágít, míg a levélszemét a címzetteknek bosszúság, addig a küldőknek hatalmas üzleti lehetőség. Ellentétben ugyanis a hagyományos, papírra nyomtatott levélszeméttel, az elektronikus levelek előállítása, továbbítása gyakorlatilag költség nélkül oldható meg, ugyanakkor hatalmas tömegeket lehet vele könnyen és gyorsan elérni. Beszédes adat, hogy jelenleg a világ elektronikus levélforgalmának 40 százaléka szemét, és ez az arány folyamatosan növekszik. Nem véletlenül: szemetelni a látszat ellenére érdemes. Egy olyan CD, amely hetvenmillió elektronikus címet tartalmaz, mindössze 150 dollárért megszerezhető. Igaz, hogy ötven ténylegesen vásárlóvá váló címzettre egymillió boszszús számítógép-használó jut, ám ha utánaszámolunk, 3500 vásárló megszerzése mindössze 150 dollárba került, ami kiváló költségaránynak számít.
A szemetelést segíti, hogy a jogi szabályozás hézagos, és a határokon átívelő tevékenység ellen nem lehet hatékonyan fellépni. A küldő haszna pedig a legkevésbé sem vigasztalja azokat a szolgáltatókat, akik értékes erőforrásokat kénytelenek a szemétáradat kezelésére fordítani. A levélszemét miatti műszaki bővítések minden évben több milliárd dollárt emésztenek fel az internetszolgáltatók oldalán – igaz, a berendezések gyártói ezt valószínűleg a legkevésbé sem bánják. Mindenki jól fontolja meg tehát, melyik oldalra áll!
➲ http://www.telusinternational.com/services/whitepaper.html

Kapcsold a hatodik sebességet!

Az Acer ez év elejétől hivatalos szállítóként hordozható és asztali számítógépekkel, kiszolgálókkal és monitorokkal látta el a Ferrarit. Az együttműködés jegyében Ferrari 3000 néven a cég a kínálatát egy különleges noteszgéppel bővítette. Az új noteszgép a sportautókhoz méltóan „tüzes” megjelenésű, kiépítését áttekintve pedig kétség sem fűződhet kiváló teljesítményéhez: AMD Athlon XP-M 2000+ és 2500+ közötti processzorok, 15"-os kijelző 1400×1050 képpont felbontással, 512 MB memória, amely – a noteszgépek világában egészen szokatlan módon – legfeljebb 2 GB-ig bővíthető, ATi Moblity Radeon 9200 VGA-vezérlő 128 MB memóriával, 60 GB-os merevlemez, DVD-meghajtó, kapuk és csatlakozások sokasága, 3 kg alatti tömeg, háromórás akkumulátoros üzemidő.

A Ferrari 3000 hazánkban is megvásárolható, ára kb. 480 000 Ft+áfa.
➲ http://www.acer.com

Elmúlik egy korszak

A Phoenix Technologies bemutatta következő generációs BIOS-ainak elgondolását és fejlesztési útitervét.

Az immár nem is IOS-nak vagy gépszintű kódnak (firmware), hanem Core System Software-nek (CSS), vagyis központi rendszerprogramnak nevezett program teljesen újfajta szolgáltatásokkal járulna hozzá megbízhatóbb, könnyebben felügyelhető és szerteágazó kapcsolatokat nyújtó számítógépek építéséhez, a tenyérgépektől kezdve egészen a kiszolgálókig.

A mai BIOS-ok még az ősrégi IBM-szabványokon alapulnak – a kor követelményeinek már jó ideje nem felelnek meg. A Phoenix d-NA (a Device-Networked Architecture rövidítése) néven kínálja majd termékét, amely valójában építőelemek, programblokkok gyűjteménye lesz. A fejlesztők négy célt tűztek ki maguk elé: megbízhatóság, felügyelhetőség, kapcsolatteremtés és használhatóság.

A megbízhatóságot digitálisan aláírt operációs rendszerrel kívánják elérni, illetve olyan titkosítási szolgáltatásokat kívánnak megvalósítani, amelyekkel a biztonsági rendszer a munkaállomásoknál alapozható meg.

A Phoenix d-NA-re épülő gépek önfelügyelő, önjavító és önhitelesítő képességekkel fognak rendelkezni, továbbá magasabb szintű energiakezelésre és hibajóslásra lesznek képesek.

A CSS TCP/IP és XML alapú kapcsolatteremtésre mind az operációs rendszer, mind a hálózat felé képes lesz. Erre elsősorban rácshálózatok, fürtök építésekor lehet szükség, de – a megbízhatóságra visszautalva – a távfelügyelet is alapszolgáltatásnak számít majd.

A használhatóság a különféle felhasználói csoportok egyedi igényeinek jobb kielégítését szolgálja majd. A Core Managed Environment (CME) segítségével a rendszerépítők védett – víruskereső, rendszer-helyreállító – alkalmazásokat telepíthetnek majd a gépekre.

A Phoenix szándéka szerint a CSS nem egyetlen zárt termék lesz, hanem általános BIOS-helyettesítő, amely nem egy gyártó tulajdona lesz, hanem bárki elkészítheti majd a saját megvalósítását. A hordozható gépekhez és tábla-PC-khez szánt változat már megjelent, a többi kiadása jövő év első negyedévére várható.

➲ http://www.phoenix.com

Szárnyra kapva

A Seiko Epson Corporation mérnökei a névadásban kevésbé ötletesek, mint a robotépítésben: apró repülő robotjukat, amely a világ legkisebb ilyen szerkezete, egyszerűen mikro-repülőrobotnak keresztelték el. Az Epson immár tíz éve mutat be hasonló szerkezeteket, természetesen mindig az adott kor műszaki színvonalának a csúcsát képviselve. Mostani apróságuk – amellyel ezúttal is

azt kívánják szemléltetni, hogy az Epson a mikromechanika területén élenjár – két ellentétes irányban forgó rotorral emelkedik a levegőbe, ezeket eredetileg karórákba fejlesztett aprócska motorok hajtják. A felszabadult röpködést egyelőre akadályozza a tápellátás nehézsége – bár már léteznek néhány grammos elemek, amelyek elegendő energiát szolgáltatnak a robotnak, az Epson emberei elégedetlenek velük, ezért nagyobb teljesítményű akkumulátorokat keresnek. A cél, hogy a mikrorepülő erejét ne a saját akkumulátorának cipelése kösse le, hanem néhány gramm erejéig hasznos terhet is tudjon magával vinni. Addig is marad a tápvezeték, amely pórázként köti a gazdájához.

➲ http://www.epson.com

Terjeszkedő Westel WLAN

A Westel a Ferihegyi repülőtér 2B terminálján egy éve – tesztjelleggel már üzemelő rendszerére alapozva – Magyarországon elsőként indított kereskedelmi WLAN-szolgáltatást. A szolgáltatást a december 1-jei indulással egyidőben a reptér 2A termináljára és a Westel valamennyi üzletére is kiterjesztették.

A WLAN-szolgáltatást előfizetéses és Domino kártyával rendelkező ügyfelek egyaránt igénybe vehetik, az előbbiek megadott vásárlási keretük, az utóbbiak pedig kártyaegyenlegük erejéig használhatják a 15 perces egységekben számlázott hálózati csatlakozást. A Westel a szolgáltatást a későbbiekben számos további forgalmas helyen is elérhetővé fogja tenni.

➲ http://www.westel900.net

WineX 3.2

A TransGaming Technologies elkészült a WineX 3.2-es változatával, amellyel a legújabb windowsos játékok – Max Payne 2, Warcraft III, The Frozen Throne, Homeworld 2 stb. – is futtathatók Linux alapú rendszereken. A WineX 3.2 a hangkeltés, a grafikai megjelenítés és a sebesség terén is számos fejlesztés nyomát hordozza magán, amelyek eredményeképpen a korábban már futó játékok is élvezetesebben játszhatók; sok játék esetében például jelentősen csökkent a futtatáshoz szükséges memória mérete. Az előfizetők rpm, deb, ls, tgz formában tölthetik le az új kiadást.

➲ http://www.transgaming.com.

Power Mac G5

Power Mac G5 sorozatát újabb, kétprocesszoros modellekkel bővítette az Apple. A cég 64 bites, akár 1 GHz-es előoldali buszsebességgel üzemelő processzorai bátran felveszik a harcot az Intel és az AMD által jelenleg a munkaállomásokhoz kínált lapkákkal – és nem csak a sebesség terén. Ha kettő 1,8 GHz-es G5 processzort, 512 MB memóriát, 160 GB-os merevlemezt, GeForce FX 5200 Ultra VGA-kártyát és DVD-R-meghajtót tartalmazó összeállítást választunk, mindössze 2500 dollárt, vagyis nagyjából 550 ezer forintot kell leszurkolnunk, ami egy Apple-féle iparművészeti remekért – mondjuk úgy – nem pénz.

Zeneszolgáltatás a kisszobából

A Slim Devices a Squeezebox nevű zenelejátszó készülékkel bővítette kínálatát. A Squeezebox ethernet alapú vagy vezeték nélküli hálózati kapcsolaton keresztül fogadja a személyi számítógépről adatfolyamként továbbított zenei anyagokat, majd például a hifitoronynak, az erősítőnek adja tovább őket. A készülék kiszolgálását a SlimServer nevű, nyílt forrású – Linuxra, Macintoshra és Windowsra egyaránt elérhető – alkalmazás végzi. A kiszolgáló akár több Squeezeboxot is tud kezelni, ezek egymástól független zeneszolgáltatásra és összehangolt működésre is rávehetők. Az ötletes készülékre tulajdonképpen nem mondhatjuk, hogy forradalmian új lehetőségeket kínál, viszont lehetővé teszi, hogy a tulajdonosa elszakadjon a számítógépétől, és lakása távolabbi részeiben is kényelmesen, jó minőségben érhesse el a gépen tárolt zenéket.

➲ http://www.slimdevices.com

Leírások minden mennyiségben

Az egyébként linuxos oktatási és tanácsadási területeken mozgó Lineox – Lineox LIFF néven – újszerű, teljességre törekvő dokumentációs terméket adott ki.

A LIFF a gyártó állítása szerint számos korszerű megoldást egyesít, így próbál a világ legjobb linuxos dokumentációs megoldásává válni. A LIFF közel 900 MB-nyi leírást tartalmaz, a gyűjteménybe a Linux Documentation Project összes könyve és HOGYAN-ja bekerült, ahogy a hamarosan megjelenő Lineox Enterprise Linux leírása, a súgó (man) oldalak HTML-formátumba alakítva és a legújabb Red Hat- és Mandrake-kézikönyvek is.
A hatalmas szövegmennyiséget a szerkesztők indexelték és kiterjedt keresési lehetőségekkel látták el. Maga a LIFF webkiszolgáló alá telepíthető, illetve a Lineox webhelyén interneten is kipróbálható. Akinek megtetszett, a Lineox webhelyéről CD-lenyomatként töltheti le vagy CD-re írva vásárolhatja meg.
➲ http://www.lineox.com

A HD DVD nyert

Hosszas huzavona eredményeként a DVD Forum tagjai végre elfogadták a következő generációs DVD-formátumot.

Az előterjesztés eredetileg a csak olvasható és írható lemezekre egyaránt kiterjedt, ám ez alkalommal csak az előbbit fogadták el a bizottság tagjai. A HD DVD lemezek ibolyaszínű lézert fognak használni, a jelenlegi DVD-lemezekéhez hasonló szerkezetet használnak majd, a kapacitásuk pedig 15–20 GB lesz oldalanként.
A HD DVD formátum a Toshiba és az NEC fejlesztése, velük szemben gyártók egy másik csoportja áll, akik a Blu-ray formátumot támogatják. A Blu-ray kék lézerrel dolgozik, és nem működik együtt a jelenlegi DVD-formátummal, viszont olyan szereplők támogatják, mint a Sony, a Philips, a Samsung és a Matsushita. A mostani döntés az írható lemezek sorsát még nem rendezte véglegesen, és nyilván a Blu-ray fejlesztésébe ölt összegekről sem akar senki szó nélkül lemondani, így komoly esély mutatkozik arra, hogy a felhasználóknak – akár az írható DVD-lemezek esetében, ahol pluszos és mínuszos jelzésű, egymással együttműködni nem képes megoldások jelentek meg – a jövőben is kétféle szabvánnyal kell majd megküzdeniük.
➲ http://www.dvdforum.com

HP-tévé jövőre

Ha hinni lehet a híreknek, a HP jövőre nagyképernyős LCD-tévékészülékekkel bővíti kínálatát. Mivel a számítástechnikai eszközök egyre inkább szórakoztató jelleget (is) öltenek – gondoljunk csak a digitális fényképezésre, a családi filmek rögzítésére és DVD-re írására, vagy akár a HP media center PC termékére –, ezzel a lépéssel logikusan egészül ki a cég meglévő szórakoztató-termékcsaládja. A HP egyébként nem egyedüliként, sőt nem is elsőként döntött így, a Dell és az inkább az Egyesült Államokban ismert Gateway már kínál a saját neve alatt tévékészülékeket.

A méret a lényeg

Az Intel elkészült első, 65 nanométeres eljárással gyártott lapkájával, egy 4 Mbites SRAM egységgel. A cég a processzorgyártás terén még csak most tér át a 90 nanométeres megoldásra, a 65 nanométeres lapkák megjelenése 2005-re várható, viszont a bejelentés bizonyára megnyugtatja azokat, akik a lapkagyártás jövője és a Moore-törvény érvényben maradása miatt aggódtak.
Az Intel fejlesztői a méretcsökkenés miatt jelentkező nehézségek elhárítására újabb és újabb anyagokat és eljárásokat kénytelenek bevetni, ennek ellenére bizakodók: 2007-re már a 45 nanométeres gyártás beindítását tervezgetik.
Mindeközben elemzők szerint az Intel többrétegű műanyag alapú memóriáinak a fejlesztése is előrehaladott állapotba kerülhetett, a cég ugyanis műanyag memóriákkal foglalkozó csapatát programfejlesztőkkel bővítette. Hozzáértők szerint ez annak jele, hogy a műanyag lapkák hamarosan gyártásba kerülhetnek; a programozókra most az a feladat vár, hogy az új megoldások előnyeit kihasználni és szemléltetni képes alkalmazásokat készítsenek. Ha a műanyagok alkalmazása ezen a területen tömegesen beindul, akkor a várakozások szerint az elektronikus készülékekbe beépített memóriák mérete nagyságrendnyivel fog növekedni, és a szilícium alapú memóriákkal eddig megvalósíthatatlan megoldások is elérhetővé válnak.
➲ http://www.intel.com

Medgyesi Zoltán
(mz@rettesoft.hu)
A Linuxvilág hírszerkesztője. Szabadidejét legszívesebben a barátnőjével tölti, szeret autózni és bográcsban főzni.

Linux-index

1. Annak a büntetésnek a mértéke euróban, amit München városa szabott ki a The SCO Groupra, amiért megalapozatlan vádakkal illette a Linuxot: **10 000**
2. Azon PC Club boltoknak a száma, ahol a számítógépekre előre feltelepítik a LindowsOS-t: **51**
3. Azoknak az éveknek a száma, amióta az ibiblio.org Linux-változatokkal és -kiszolgálóval dolgozik: **12**
4. Az angol weben böngészőknek az a százaléka, akik inkább feladnák a mobiltelefonjukat, mint az internetet: **40**
5. A Linux-kiszolgálók piacának becsült forgalma dollármilliárdban 2002-ben: **2**
6. A Linux-kiszolgálók piacának becsült forgalma dollármilliárdban 2007-ben: **15**
7. A Linuxszal telepített PC-k eladásának Indiára becsült százalékos adata 2004 márciusáig: **10**
8. India részesedése a tengerentúli IT-szolgáltatások piacából: **60**
9. A tengerentúli IT-szolgáltatások piacának becsült éves adata dollármilliárdban: **16**
10. A személyi videófelvevők (például a TiVo) *Rupert Murdoch* által jósolt ára egy éven belül: **0**
11. Echostar (DishTV) által megadott PVR-előfizetési díj, amennyiben az ügyfél bizonyos programcsomagokra két évre fizet elő: **0**
12. *Rupert Murdoch* hírügynöksége átlagos éves bevételének (EBIT) százalékos növekedése: **20**
13. A B2B tranzakciók növekedése milliárd dollárban világszerte 1999-ben: **145**
14. A B2B tranzakciók várható növekedése milliárd dollárban világszerte 2004-ben: **7,29**
15. Az Egyesült Államok részvétele a B2B tranzakciókban 1999-ben: **39**
16. Az Egyesült Államok várható részvétele a B2B tranzakciókban 2004-ben: **39**
17. A megkérdezett programozók százaléka, akik várhatóan Linux-alkalmazásokat is írnak jövőre: **59**
18. A megkérdezett programozóknak az a százaléka, akik szerint az SCO kontra IBM per befolyásolja majd a terveiket: **6**
19. A Microsoft által tervezett Linux-kiszolgálófelhasználás százalékos növekedése 2003 júliusa és 2004 júniusa között: **24**
20. A Microsoft által tervezett Windows-kiszolgálófelhasználás százalékos növekedése ugyanerre az időszakra nézve: **9,5**

Források
1.: Heise Online
2.: Lindows.com
3.: ➲ http://www.ibiblio.org
4.: the inquirer
5–6.: Giga Information Group, via Veritas
7.: Red Hat
8–9.: Gartner Group
10–12.: Reuters
13–16.: Gartner Group
17–18.: Evans Data Corp.
19–20.: Jupiter Research

Linux Journal 2003. december, 116. szám

Ők mondták

Biztos beszívtak.
(Linus Torvalds a The SCO Group azon állításaira hivatkozva, amelyek szerint a Linux többmillió, a Unixból kimásolt sort tartalmaz)

A Linux nem termék. A Linux inkább a programösszetevők gyűjteménye, amelyeket világszerte önállóan alkotott meg több ezer ember, és amelynek minden eleme sajátos ütemezés szerint változik és fejlődik. Ha a Linuxról úgy beszélünk, mint termékről, azzal „befagyasztunk" egy alapjában véve dinamikus dolgot, és lezárunk olyasvalamit, ami egy nyitott folyamat. Ezt nem tehetjük meg, csak egy fontos elem elvesztése árán. Nem, a Linux nem egy termék – a Linux egy folyamat.
(Ian Murdock, ➲ http://news.com.com/2010-1071_3-5057321.html)

Amikor ötödikes voltam, parázs vita tört ki a helyi újságban a zsebszámológépekről. Az egyik fél szerint a számológépek jó készülékek voltak, amelyek használatát meg kellene tanítani a diákoknak. A másik fél a számológépet mankóként tartotta számon, amelynek alkalmazása csak beszennyezi az ifjak könnyen befolyásolható gondolkodásmódját, és amelyek megjelenése a világvége közeledtének biztos jele.

Mind az optimisták, mind a pesszimisták bíztak abban, hogy a véleményük számít. „Eljön majd az idő – gondolták –, amikor valaki megkérdezi tőlem, hogy mit is tegyünk ezekkel a számológépekkel." Velem és az ötödikes haverjaimmal ellentétben a felnőttek nem értették meg, hogy a számológépeket többé nem lehet eltörölni.
(Clay Shirky, ➲ http://www.corante.com/many/20030701.shtml#46771)

Változás

Az Informatikai és Hírközlési Minisztérium által 2003. május 15-én meghirdetett, „Vállalatirányítási rendszerek közepes méretű vállalkozásoknak" című, folyamatos beadású pályázatot (a pályázat jele: IHM-ITP-7) – a pályázati keretösszeg kimerülése miatt, a Pályázati Útmutató VIII. 6. pontja alapján – a minisztérium 2003. december 6-ával leállította. Az eddigi nyertesek listája a következő címen érhető el:
➲ *http://www.ihm.gov.hu/palyazatok/palyazatok_20030926_1.html*

Bemutatták a SuSE Linux 9.0-t

November 18-án egy kellemes ebéd keretében mutatták be a SuSE Linux Professional 9.0-s magyar változatát. Mi természetesen ott jártunk, de ez nem volt nekünk elég: szemügyre is vettük az új darabot.

Fél évvel az előző, 8.2-es változat után piacra került a legújabb, 9.0-val jelölt, elsősorban otthoni felhasználásra ajánlott SuSE Professional. Az eseménynek helyt adó étterem egyik különtermében körülbelül ötven mindenre elszánt érdeklődő gyűlt össze, hogy megismerhesse az új „csodafegyvert" – így hirdeti ugyanis legújabb Linuxát a SuSE.

A helyfoglalás után, de még az ebéd előtt *Dr. Szentiványi Gábor*, a SuSE magyarországi ügyvezetője előadásával megnyitotta a sajtótájékoztatót, amelynek legelső témája természetesen a Novell üzleti lépése volt, amelynek során felvásárolta a németországi SuSE Linux AG anyavállalatot. Az előadó természetesen mindenkit megnyugtatott, hogy a termékek ettől még nem tűnnek el, minden marad a régiben, hiszen csak tulajdonosváltás történt. A SuSE Professional ugyanúgy kapható és ingyenesen letölthető marad, valamint a termékcsalád többi tagja sem sínyli meg a változásokat.

Ezek után elérkeztünk az előadás fő témájához, magához a 9.0-s változathoz. Az előadó röviden összefoglalta a tudnivalókat, majd két fő újításra hívta fel a figyelmet: az egyik az NTFS lemezrészek átméretezhetősége, a másik a YaST kiszolgálómoduljainak a bővülése. A vetélytárs Windows operációs rendszerek eddig akadályt állítottak a telepítőnek, ugyanis bár a SuSE telepítés közben átméretezte a FAT-lemezrészeket, a Windows 2000 és XP NTFS lemezrészeivel meggyűlt a baja. Mivel a legtöbb Linux-változat úgy próbál előbbre jutni, hogy fokozatos átállást sugallva feltelepíthető a gépen lévő operációs rendszer mellé, a fent említett újítás valóban mérföldkő lehet a felhasználótérítés hosszú és rögös útján.

A másik újdonság a felhasználóbarát testreszabás tekintetében fontos. Számos új kiszolgálóprogram ugyanis beállítási lehetőséget kapott a YaST *Hálózati szolgáltatások* moduljában, amelynek segítségével bárki egyszerűen állíthat be DNS, DHCP, FTP, Samba vagy épp webkiszolgálókat.

Ezek mellett jelentős újításnak számít még az OpenOffice-csomagon végrehajtott változtatás, amelyből a legújabb, 1.1-es változat került a terjesztésbe, s szó esett arról is, hogy a SuSE az első olyan otthoni használatra szánt operációs rendszer, amely támogatja az AMD Athlon 64 új, 64 bites technológiáját.

A sajtótájékoztatón elhangzott még az is, hogy a SuSE-termékekre ezentúl 180 napos terméktámogatás jár, amit természetesen minden vásárló igénybe is vehet, ugyanúgy, ahogyan eddig is megtehette, ám az előző változatokra a nyilatkozatok és a doboz tanulsága szerint „hivatalosan" csak 90 nap járt, valójában azonban ez 180 nap, azaz egy félév volt. Ennyi ugyanis egy-egy változat életciklusa SuSE-éknél.

Az előadás további részében egy másik célcsoportnak szánt termék bemutatása következett: a Standard Server 8.0-s képességeiről esett pár szó, amit a cég elsősorban kisvállalkozások számára kínál a vetélytársaknál jóval kedvezőbb (ámde még így is borsos) áron. Ennek a terméknek az életciklusa sokkal hosszabb, mint az otthoni használatra szánt Professionalé, és a SuSE ötéves programtámogatási garanciát is vállal rá. Természetesen ehhez is jár a terméktámogatás és a vaskos kézikönyv, amely a termék dobozában található.

Az előadás végeztével az érdeklődők kérdésekkel bombázhatták Szentiványi Gábort, aki kötelezettségeinek eleget téve még az ebéd mellett is folytatta a kíváncsi hallgatóság tudásszomjának kielégítését.

Az ebéd végeztével, távozás előtt minden résztvevőt megajándékoztak egy SuSE 9.0-val, hogy valóban meggyőződhessenek az elmondottak igazságtartalmáról. Mivel én magam is hitetlennek számítok, a népmesei közhelyet követve utánajártam annak, hogy miféle újdonságok találhatók a frissen piacra dobott termékben.

Lássuk, mi van a dobozban!

Először vegyük szemügyre a dobozt: az előző változathoz képest meglepő a fordulat, most ugyanis teljes egészében magyar nyelvű a doboz felirata, nemcsak egy utólagosan ráragasztott matrica tanúskodik a változat honosításáról, továbbá a doboz kinézete is megújult kissé. Kinyitva a korongcsomag mellett megtaláljuk a két vaskos felhasználói kézikönyvet, amelyek nemcsak új külsőbe burkolóztak, de a tartalmuk is jócskán megnövekedett. Továbbhaladva a lemezek szempontjából elmondható, hogy ezúttal nem két, hanem egyetlen kétoldalas DVD került az öt CD mellé, amelyet nem sikerült úgy kivennem a papírborítékból, hogy az ujjlenyomataimmal ne maszatoltam volna össze mind a két oldalát. (Természetesen a korongtok és a lemezek borítója is új ruhába bújt.)

A telepítés szinte pontosan ugyanúgy zajlik, mint az előző változatoknál: minden egyszerű és követhető; nekünk, felhasználóknak még annyit sem kell tenni, mint egy windowsos telepítés során.

A CD-k beadagolása után a rendszer elindul, és mi azonnal használatba vehetjük. Az alapértelmezett grafikus felület még mindig a KDE, a „Startmenüt" azonban egy kissé átszervezték, a panel mellett sincs már ott a megszokott SuSE munkamenü, de egyáltalán nem is hiányzik. A legtöbb dolog maradt a régiben, égbekiáltó változások nem érzékelhetők. Általában elmondható,

hogy a kinézet néhány ponton megváltozott: pár helyen új ikonokat találunk, és a rendszer minden összetevőjének a változatszáma emelkedett néhány pontnyit. Akadnak természetesen új elemek: a multimédia területén megpróbáltak erősíteni, van már RealPlayerünk is, és ki tudja még hány dolog, amit csak akkor vennénk észre, ha a sok ezer csomagot mind végigbogarásznánk. Erről jut eszembe – a csomagok: változtattak az alapcsomagok összetételén, úgy próbálták meg átalakítani, hogy a legtöbb felhasználó igényét kielégítsék, tehát ne kelljen gyakran új összetevőket telepíteni a rendszerben. Ennek egyik ékes példája az, hogy az önműködő telepítés végre magától felrakta a Midnight Commandert. Mivel ezt én utólag eddig is mindig felpakoltam, s nem is nagyon ismerek olyat, aki ne használná, ez a lépés valóban hasznosnak bizonyult, főleg, ha a többi csomagösszetétel esetén is ilyen irányba mozdult el a terjesztés.
A rendszer egyébként a 2.4.21-es rendszermag javított (optimalized) változatát használja, amely a 2.6-os változat néhány újítását is tartalmazza. Maga a 2.6-os rendszermag legújabb próbaváltozata is felkerült, de ennek használata azért nem olyan egyszerű, és a 9.0-hoz tartozó jegyzetben benne is van, hogy a 2.6-os próbaváltozatú rendszermaggal futó SuSE-hez nem is jár az a bizonyos 180 napos terméktámogatás.
Talán ennek köszönhetően nem valósult meg a teljes körű NTFS-támogatottság sem. Ugyan az NFTS lemezrész telepítés közben átméretezhető, de írni még mindig nem tudjuk Linux alól, csak olvasni. Amennyiben tehát olyan lemezrészeket szeretnénk, amelyeket mindkét operációs rendszer alól el akarunk érni, azokon a lemezrészeken továbbra is célszerű FAT-ot használni.

A YaST

A legtöbb változás magában a YaST rendszerösszetevőben tapasztalható, amely a SuSE terjesztéskezelő programja, azaz SuSE Linuxok lelke. A használatát azért nem kell újra megtanulni, ugyanis nem annyira változott, sokkal inkább bővült. Ami módosulásnak nevezhető, az talán az, hogy nem YaST2-nek hívják, hanem polgári nevén csak YaST-nak.
Lássuk, mivel is bővült rendszerünk lelke! Leginkább a hálózati szolgáltatások terén tapasztalhatunk gyarapodást: az otthoni használatra is alkalmas szolgáltatásokra kapunk beállítómodulokat. Lehetőségünk nyílik például az otthoni hálózatunk üzemeltetését megkönnyítő DHCP grafikus panelen történő beállítására, házi FTP-kiszolgálót szabhatunk testre, és immár a windowsos hálózatra történő csatlakozás sem jelenthet gondot, hiszen ezekre a feladatokra is találhatók megoldások a YaST moduljai között. Ezeken túlmenően létezik néhány olyan szolgáltatás, amelyet az otthoni fejlesztők is igénybe vehetnek. Ilyen lehetőség az Apache 2 webkiszolgáló beállítása, amely azonban nem nyújt teljes lefedettséget, csak az alapvető beállítások végrehajtására van lehetőségünk. Ezért ez a szolgáltatás talán azok számára lehet hasznos, akik például a képeiket a weben meg szeretnék osztani az ismerőseikkel.
A HTTP-kiszolgálónál valamivel több beállítást áll módunkban megtenni a DNS-kiszolgáló testreszabása

közben, amely a jól ismert linuxos Bind névkiszolgálóra vonatkozik.
Ezenkívül még egy hasznos, általam egyébként is használt szolgáltatás beállítására is lehetőség nyílik: igénybe vehetjük az NTP-t (Network Time Protocol), amely lényegében annyit jelent, hogy a gép minden indításakor és azután meghatározott időközönként lekérdez egy pontos, atomórához igazított időkiszolgálót, s a mi óránkat beállítja a kívánt értékre, így mindig biztosak lehetünk abban, hogy az pontosan jár.
Ezekről a beállítómodulokról általában elmondható, hogy egyelőre csak nagyon korlátozott számú jellemzőt képesek átállítani, az viszont mindenképp biztató, hogy egyáltalán léteznek. Feltételezhető, hogy ez a jövőben csak javulni fog, s a beállítható szolgáltatások listája is gyarapodik majd.

Végkövetkeztetés

Összefoglalva azt tudom mondani, hogy ismét egy kicsit felhasználóbarátabb Linux került a piacra, ami mindenképpen hasznos a Linux terjedésére nézve. A mostani változat ugyan nem sokban tér el az előzőtől, de összefogottabb lett, s talán előnyt tudunk kovácsolni a meglehetősen rövid életciklus tényének az ismeretéből, hiszen megállapíthatjuk, hogy a változat a legfrissebb programokat tartalmazza, s ezáltal védettebbek vagyunk a támadások, vírusok ellen – és az új változatok sok esetben a mi kényelmünket szolgálják.
Mindenkinek azt javaslom, hogy próbálja ki! A későbbiek folyamán az „Átállás Linuxra lépésről lépésre" című sorozatunkban is foglalkozunk majd az új vívmányok az adott területen történő bemutatásával.

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

A 2003. év Linux Journal Szerkesztőségi díjasai

A Szerkesztőségi díjak átadása előtt minden cikkírónktól és szerkesztőnktől javaslatot kérünk a díjazottakra. Ebben az évben ismerős nevekkel és kezdő linuxosokkal egyaránt találkozhatunk a győztesek között.

Kiszolgálógép

Sputnik AP 120

Vezeték nélküli hálózatot akarsz adni vevőidnek vagy a szomszédnak? Vagy saját gépet építettél NoCatAuth-tal, és hitelesítetted valamelyik drága szolgáltatóval, vagy kockáztattál és… a Sputnik egy csodás Linux-megoldást hozott létre, olyat, amelynek révén használható és viszonylag olcsó hozzáférési ponthoz juthatunk.
➲ http://www.sputnik.com/products/ap120.html

Biztonsági eszköz

Netfilter/iptables

Mick Bauer, a biztonsági rovat szerkesztője a következőket írta: „a 2.4-es Linux-rendszermag csomagszűrő kódja – bár nem 2003-as újdonság –, jól megállja a helyét: az üzleti megoldások szintjére hozta fel a Linux-tűzfal kezelését. Rugalmas és értelmes, hatékony kapcsolatállapot-követési képességgel.” Mick rámutat még, hogy a jól ismert Netfilter a gépen egy újabb tűzfalvédelmi rétegként használható, akkor is, ha amúgy más tűzfalat is birtoklunk.
➲ http://www.kernel.org

Kiszolgáló

Newisys

Michael Baxter „remekül megtervezettnek” nevezte ezt a kétprocesszoros Opteron 1U kiszolgálót, mivel a 64 bites Opteron áttörte a x86 memória korlátait – a visszafelé történő együttműködés megőrzése mellett. A Newisys alapú kiszolgálók napjaink linuxos kiszolgálópiacának népszerű szereplője számos linuxos kiszolgálókkal foglalkozó cég között. Ja, és majdnem olyan gyorsan indítják a GNU Emacsot, mint a legtöbb ember gépe a Vimet.
➲ http://2100 www.newisys.com/products/2100.html

Munkaállomás

Dell Precision

Glenn Stone ítészünk a kétprocesszoros Xeon asztali rendszert úgy határozta meg, mint „komoly gép komoly munkához”, és csodálattal adózott a 320 MB/s sebességű SCSI RAID alrendszer teljesítményének, valamint a Dellnek a helyszíni szolgáltatásokhoz való hozzáállásáért.
➲ http://www.dell.com/precision

Webböngésző vagy ügyfél

Mozilla 1.4

Többrétegű weblapok, amelyeken fülek nyithatók; az előugró (pop-up) hirdetéseket tartalmazó ablakok tiltása, kulcsszavak tárolása – ha más böngészőt használva találkozunk velük, egyszerűen elavultnak, korlátozottan vagy borzalmasan kezelhetőnek tűnnek. A Konqueror is jó, de a Mozilla jelenleg ezt is lekörözi; azok számára készült, akik a webes megjelenítést a maguk szája íze szerint szeretnék kialakítani.
➲ http://www.mozilla.org

Grafika

Jahshaka

Nem, ez nem különdíj a „csak ne Gimp legyen, mert ők mindig nyernek” versenyben! *Greg Kroah-Hartman* hívta fel a figyelmünket erre a vadiúj, fejlesztés alatt álló videoszerkesztő alkalmazásra, és nem tudtuk megállni, hogy ne készítsünk hozzá egy teljes oktatóanyagot. A vágáson túlmenően a Jahshaka animálást, hatásokat (effect), karakterkészítőt és fájlmegosztást is kínál.
➲ http://www.jahshaka.com

Kommunikációs eszköz

Gaim

Marcel Gagné állítja: „Korábban kigúnyoltam az azonnali üzenetváltást (instant messaging – IM), de az elmúlt néhány hónapban felfedeztem, hogy milyen csodálatosan hasznos eszköz is ez. Néha semmi sem helyettesíti a valós idejű, folyamatos beszélgetést, ha műszaki nehézségeket akarunk megoldani.” Nos, a Gaim egy ilyen üzenőprogram azoknak, akinek a barátai nem szeretik az IM rendszereket. Lapzártakor a Gaim által támogatott rendszerek: AOL Instant Messenger, ICQ, MSN Messenger, Yahoo, IRC, Jabber, Gadu-Gadu és a Zephyr.
➲ http://gaim.sourceforge.net

Fejlesztőeszköz

Perl 5.8.0

Reuven M. Lerner minden hónapban tudósít bennünket egy új webfejlesztő eszközről, de most térjünk vissza a jó öreg Perlhez. Az új változat kapcsán ezt írja: „a legfontosabb, hogy támogatja a többszálúságot és az Unicode-ot – két olyan dolgot, ami segíthet előremozdítani a Perlt az elkövetkezendő években.” Szeretjük böngészni a Comprehensive Perl Archive Networköt (CPAN) olyan modulokért, amelyek egy rövid parancsfájlt képessé tesznek arra, hogy pontosan azt tegyek, amit szeretnénk.
➲ http://www.perl.org

Adatbázis-kezelő

PostgreSQL

Reuven emellett PostgreSQL-rajongó is, azé a rendszeré, amely oldalainkon is sokszor szerepelt már:

„A PostgreSQL csapata bebizonyította, hogy igenis létre lehet hozni adatbázis-kezelőt a MySQL árát nem meghaladva és a felügyelet kezelhetőségének képességével, ugyanakkor az Oracle tulajdonságkészletével. Versenytársa csak a Firebird és a SAP DB lehet, valószínűleg mindkettőre komoly felfutás vár 2003–2004-ben". Az új 7.3-as változat alapértelmezésben tartalmaz táblafüggvényeket (olyan függvények, amelyek több sort adnak vissza), sémákat, előkészített lekérdezéseket, valamint Unicode-támogatást, továbbfejlesztett naplózást és zárolás nélküli adatbázis-tisztítást. Marcel rámutat, hogy „amire szükség van, az egy pofonegyszerű adatbázisrendszer az új felhasználóknak, akik talán csak egy karácsonyi üdvözlőlapot vagy efféle akarnak készíteni. Csak, mert tudunk egy mindentudó relációs SQL adatbázis-kezelőt csinálni, ráadásul ingyen, az még nem jeleni azt, hogy a felhasználók is mindig ezt akarják."
➲ http://www.postgresql.org

Irodai program

OpenOffice.org

Marcel az OpenOffice.org-ot ugyancsak ajánlja – ismét őt idézzük: „szinte teljes körűen támogatja a Microsoft Office-dokumentumokat". Valószínűleg mindenkinek tetszeni fog a szövegszerkesztő, de az oldal rajzolóprogramot és bemutató készítő csomagot is tartalmaz.
➲ http://www.openoffice.org

Az év terméke

SGI Altix 3000

Emlékszel a világ legnagyobb egymagos Linux-gépére a 2003. februári szám címlapjáról? Nemrég hallottuk, hogy *Stephen Hawking* professzor cambridge-i kutatócsoportja szintén vett egyet. *Steve Neuner* az SGI-től azt írta 2003 februárjában: „ha megnézed a rendszermagbővítések listáját… az voltaképpen meglepően rövid, ami jól mutatja, hogy a Linux megbízható alapokra lett tervezve. Ami még nagyszerűbb, hogy nagy részük – egyéb változtatások mellett – már megjelent a 2.5-ös fejlesztői rendszermagban." Gratulálunk az SGI-nek a Linux terjesztéséért.
➲ http://www.sgi.com/servers/altix

Weblap

Linux Weekly News

Mindenki az úgynevezett metanewst használja – ez kiválogatja a legjobb cikkeket a híroldalakról –, de a Linux Weekly News remekül kiszűri a híráradatból a fontos Linux-híreket. Ezen túlmenően egészen különböző témákról is sajátos tartalommal szolgál, például a biztonsági riasztásokról vagy a rendszermag-piszkálgatásról – bármiről, amivel kihúzod a Linux Journal megjelenéséig.
➲ http://lwn.net

Játék

Frozen Bubble

Gyerünk, emberek, nincs itt semmi látnivaló, irány vissza dolgozni! Lehet, hogy ez lesz a következő Tetris… Tőlünk ugyan nem gyakran lehetett hallani róla, de bárki is írta át Microsoft Windowsra és Mac OS-re, köszönet érte, mert így most már nem csak a Linux-felhasználók termelékenységi mutatója nullázódik le. Az új 1.0-s változatban száz – igen, száz! – pálya és egy pályaszerkesztő is található.

➲ http://www.frozen-bubble.org

Rendszerbeállító és -karbantartó program

Webmin

Marcel szerint a Webmin „olyan csodálatos kis erőforrás-igényű eszköz, amely elfedi az operációs rendszerek közötti különbséget", emellett dicséri a beépített SSH-alkalmazást is. A Webmin-modulok az elképzelhető összes kiszolgálóprogramot kezelni tudják, az Apache-től a hangpostán át a WU-FTP-ig. A külső modulok megkönnyítik az ISP-k dolgát az üzenetek és a virtuális webfelügyelet ügyfeleikhez való hozzárendelésében.

➲ http://www.webmin.com

Mobil készülék

Lindows Mobile PC

Végre egy hordozható gép (notebook) előretelepített Linuxszal! *Doc Searls* örülne egy ilyen VIA processzor alapú, 256 MB RAM-mal szerelt és minden elképzelhető kütyüt tartalmazó gépnek, ha el tudná venni hatéves kisfiától.
➲ http://www.lindows.com/lindows_feature_preinstall.php

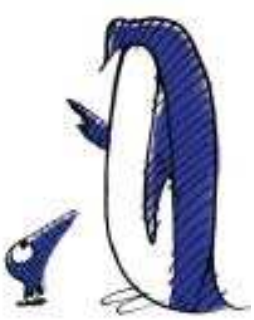
Könyv

Daniel P. Bovet és Marco Cesati
Understanding the Linux Kernel, 2. kiadás

Ez a könyv jó, ha kéznél van, ha megakadunk a Kernel Kornerben, vagy ha saját rendszermagot akarunk fordítani, és tudni szeretnénk, hogyan működik. Most nem áruljuk el, hogy kik voltak azok a szerkesztők, akik a saját könyvükre szavaztak.
➲ http://www.oreilly.com/catalog/linuxkernel2

Linux Journal 2003. augusztus, 112. szám

Most nem áruljuk el, hogy kik voltak azok a szerkesztők, akik a saját könyvükre szavaztak.



A Linux Journal honlapján számtalan gond megoldásához találhattok további segítséget. A Sunsite tüköroldalait, a gyakori kérdéseket és az egyéb útmutatásokat a ➲www.linuxjournal.com honlapon olvashatjátok el. A rovatban közzétett válaszokat Linux-szakértők kis csapata készítette el. További kérdéseiteket szívesen fogadják (angol nyelven) a ➲www.linuxjournal.com/lj-issues/techsup.html címen, ahol csak egy kérdőívet kell kitöltenetek, de a bts@ssc.com címre levelet is írhattok. A levél tárgyában szerepeljen a „BTS" kulcsszó.

A hónap szakmai tanácsai

A rendszeridő beállítása a hálózaton keresztül

Telepítettem Red Hat 8 alatt az `ntpd` démont, amely fut ugyan, de látszólag nem csinál semmit. A naplóból kiderül, hogy elindult, de pár perc múlva a következő üzenet jelenik meg:

```
kernel time discipline status change 41
```

Ezek után semmi nem kerül már a naplóállományba. Mit jelent ez az üzenet? Hogyan lehet megállapítani, hogy az `ntp` frissíti-e a rendszer óráját? Ha nem frissíti, hogyan lehet rávenni a helyes működésre?

Lance Hill, lancemhillnospam@hotmail.com

Az `ntpd` (Network Time Protocol Daemon) működésének ellenőrzése az `ntptrace localhost` parancs segítségével lehetséges. Ebből kiderül ki, hogy a rendszer és az időkiszolgáló, ami a rendszernek szolgáltatja az időt honnan „kapja ezt". Légy udvarias, ne állítsd be az `ntpd`-t egy nagy forgalmat bonyolító nyilvános időkiszolgálóra. A helyi internetszolgáltató vagy a helyi Linux-felhasználói csoport adhat hozzáférést egy közeli időkiszolgálóhoz.

Don Marti, dmarti@ssc.com

A következő dolgokat érdemes ellenőrizni. Először is az `ntpd` nem próbál meg egy óránál nagyobb időkülönbséget kiegyenlíteni, ezért az óra első beállítására használd az `ntpdate` parancsot. Másodszor ellenőrizd, hogy nem állítottad-e be az `ntpd`-t csoportos szórásos vagy adatszórásos (multicast/broadcast) ügyfélüzemmódra, hacsak nincs időkiszolgáló a helyi hálózatban. Harmadszor, ha újabb `ntp` változatot használsz, ellenőrizd, hogy a `restrict` beállítás minden használt időkiszolgálóra meg legyen adva, mert az újabb változatok jobban ügyelnek a biztonságra. Végül használd az `ntpq` eszközt és annak `assoc` parancsát az `ntpd` helyes működésének az ellenőrzéséhez.

Chad Robinson, crobinson@rfgonline.com

Nyomtatás egy alkalmazásból

Windows alá írt alkalmazásokat próbálok meg Linux alá átvinni (Red Hat 7.3 és 9), és meg kellene találnom a jelenlegi nyomtatási módszer helyettesítőjét. Jelenleg így megy a nyomtatás:

1. Az alkalmazás megnyitja az LPT8-at, és mint egy fájlba ír bele. Az LPT8 egy Windows-eszköz, nem valódi kapu. Az adatokat (sima ASCII-szöveg) a program kiírja az eszközre. Amikor vége a kiírásnak, lezárja az eszközt.
2. Az LTP8-at elkapja egy Novell Netware figyelőprogram, azaz bármit írunk az LPT8-ra, az a Novell Netware nyomtatási sorában köt ki.
3. A Novell Netware kiszolgáló elküldi az adatokat a vonalkódnyomtatónak. A nyomtató a hálózatra van kötve, és egy kis AXIS nyomtatókiszolgálót használ, amely sok protokollt ismert, például a Novell NPRINTER protokollját és az `lpd`-t. Tudni szeretném, hogy meg lehet-e tartani ezt az egyszerű működést, azaz megnyitni egy fájlt vagy eszközt, írni bele, majd lezárni. Minden másnak a háttérben kell történnie. Az világos, hogy lehetne külső programot hívni: létrehozok egy ideiglenes fájlt az adatokkal, azután átadom a fájlt például az `lpr` programnak. Ez egy kicsit bonyolultabb, de működőképes, ha az első módszer nem volna lehetséges. Esetleg létre lehetne hozni egy ideiglenes fájlt, és figyelni az ideiglenes könyvtárat, hogy van-e benne fájl. A baj csak az, hogy a felhasználó azonnali választ vár. A Novell Netware kiszolgálót Linuxra fogjuk cserélni.
   A Windows-gépek Sambát fognak használni a nyomtatáshoz. Még nem döntöttem el, hogy melyik nyomtatási rendszert válasszam, talán a CUPS lesz az.
   A programok Delphi/Kylix Object Pascal nyelven íródtak. Tudtok segíteni, hol kezdjek keresgélni?

Gus Wirth, gwirth@sciti.com

Ez vagy könnyű, vagy nehéz, attól függően, hogy mit szeretnél. A legegyszerűbb megoldás az általad említett megoldás az ideiglenes fájllal. Használhatod a `tmpnam()` függvényt egyedi fájlnevek létrehozására, hogy a folyamatod nehogy más folyamatokkal ütközzön. Ez a módszer megvalósítja a fent említett fájlmegnyitás, adatkiírás, fájllezárás folyamatot, és ehhez nem kell külső program. Ha a programod úgy lett volna megtervezve, hogy eleve ideiglenes fájlt használ, és egy beállítható paranccsal küldted volna el azt a nyomtatónak, akkor hordozható lett volna. A Windows alatt a `copy` paranccsal küldhetted volna el a fájlt az LTP8-ra, míg a Linux alatt az `lpr` parancsot használhattad volna ugyanerre. A program rugalmasabb lett volna.

Chad Robinson, crobinson@rfgonline.com

Noteszgép futtatása ACPI nélkül

Red Hat 9-et telepítettem a Dell Inspiron 1100 típusú noteszgépemre. Az energiaellátás kezelése nem települt (ez a gép ACPI-t használ, nem APM-et). Pár weboldal sikeres telepítésről számol be – az ACPI kivételével. Ragaszkodom a Red Hat megbízható kiadásához, mert a gépen a mindennapi munkám ellátására használom (napi 6–8 óra tanítás). Nem szoktam kísérletezni a munkagépemmel. Mivel jár, ha nem telepítem az ACPI támogatását? Kockáztatom a processzor egészségét, vagy meglehetek nélküle (egy darabig legalábbis)? Segítenétek, hogy hol találom a Linux alatti ACPI-támogatásról a leírást?

Erik Hemdal, ehemdal@townsip.com

Nézd meg a ➲ http://www.linux-laptop.net webhelyet, ahol az emberek leírják azokat a nehézségeiket, amelyeket a Linux noteszgépekre való telepítésével kapcsolatban tapasztaltak.

Usman S. Ansari, uansari@yahoo.com

Új termékek

Az Atmel elérhető beágyazott Linux alatt is

Az Atmel Corporation bejelentette, hogy Linux alatt is elérhető az AT91RM9200 mikrokontroller és az ARM9 alapú ASIC-ek. A linuxos átirat tartalmazza a 2.4.19-es rendszermagot, az ARM Linux csapat által karbantartott ARM9 Linuxot, a rendszermagfoltokat a fejlett megszakításvezérlőhöz és a rendszeridőzítőhöz és az AT91RM9200-hez való illesztőprogramot. A leggyakrabban használt parancsok egy kisméretű RAM-lemezen foglalnak helyet. A teljes fejlesztőkészlet tartalmazza az átiratot, az U-Boot betöltőprogramot és a GNU eszközkészletet forrásban és bináris formátumban.
➲ http://www.atmel.com

QVISION 2.0

A QVISION 2.0 egy valós idejű, nézet alapú hálózati megfigyelőeszköz, amely minden hálózati eseményt képes megjeleníteni. A QVISION megfigyeli az internetkapcsolaton és a belső hálózaton átmenő hálózati forgalmat, és felfedezi a szokatlan mintázatokat. A QVISION 2.0 újdonsága többek közt a viselkedés nézet, a hálózat megismerése nem aláíráson alapuló módszerrel, a házirendet érintő és biztonsági gondok valós idejű kezelése az események tulajdonságaihoz kötött szabályokkal, az okos riasztás és az előre megadott eseményekre elindított válasz. Ezenkívül kiválasztható, hogy mely alkalmazások milyen forgalmát milyen mélységben kívánjuk menteni.
➲ http://www.q1labs.com

Raptor 74GB SATA-meghajtók

Megjelent a piacon a Western Digital új, soros ATA (SATA) sínre csatlakozó, 10 000-es fordulatszámú WD Raptor merevlemezeivel. Az új Raptor 74 GB tárterülettel rendelkezik, ismeri az Ultra/150 parancssorbaállítást (CQ) és a RAFF-ot (Rotary Accelerometer Feed Forward). A Raptor-meghajtók minősége kielégíti a vállalati igényeket: 1,2 millió óra az átlagos idő két meghibásodás között, 10 000 fordulatot tesz meg percenként, az átlagok keresési ideje 4,5 ezredmásodperc, és öt év garancia van rá. Az Ultra/150 CQ összerűsíti a számítógép és a merevlemez közötti adatforgalmat, így nő az átviteli sebesség. A RAFF rezgő környezetben is fenntartja a merevlemez teljesítményét. Ráadásul a Raptor 74 GB-os meghajtó folyadékcsapágyas, ezért halkabb és megbízhatóbb.
➲ http://www.westerndigital.com

A Metrowerks beágyazott BSP-je a Motorolához

A Metrowerks új kártyatámogató csomagja (BSP-je) segíti a fejlesztőket a Motorola PowerQUICC III processzorra írt beágyazott alkalmazásoknak a létrehozásában. A Metrowerks BSP-t rögzített alkatrész-összeállítással és beállításokkal próbálták ki. Tartalmazza a PowerQUICC III processzorhoz való beágyazott Linux-rendszermagot, a GNU hibakeresőt és eszközkészletet, futtatási környezetet, illesztőprogramok és programkönyvtárak teljes készletét. A BSP a hálózati és távközlési piacon lehet sikeres. A BSP együttműködik a Metrowerks Platform Creation Suite csomaggal, amely a fejlesztőket segíti a Linux beágyazott rendszerekre való testreszabásában.
➲ http://www.metrowerks.com

Linux for Everyone 3.0 Cognitióval

A CPUBuilders kiadta Linux for Everyone v 3.0-t, amelyben megtalálható a Cognitio nevű új vezérlőpult. A termék a Sam's Clubon és a CPUBuilders webhelyén keresztül szerezhető be. A Cognitio vezérlőpult lényege, hogy segítségével egyszerűen kezelhetőek a Linux alatt az alkatrészekkel és programokkal kapcsolatos feladatok. A felhasználó azonnal működő eszközöket, CD-írót, merevlemezt, lapolvasót és USB-tárat adhat hozzá. A Linux for Everyone 3.0-s változatával felszerelt rendszerekben újdonság még az USB 2.0, a gyorsabb processzorok, a DDR memória és az USB 6-in-1 kártyaolvasó. A 3.0-s rendszerek programjai egy kattintással frissíthetők, vannak benne eszközök a hálózati és internetkapcsolat megfigyelésére és a hálózati fájlmegosztásra, és az új Linux-felhasználók számára dokumentáció is létezik.
➲ http://www.cpubuilders.com

Glacier Itanium 2 rendszerek

Az Aspen Systems bejelentette két új, Itanium 2 processzor köré épülő számítógépét, amelyet a nagyteljesítményű számítások (HPC) piacára szán. Az Aspen Systems Glacier Dual Itanium 2-es rendszer 1,4 GHz-es processzort és 1,5 MB L3 gyorsítótárat tartalmaz. A rendszer az Intel SR870BH2 kiszolgálófelületen alapul, és 2U magas, szekrénybe szerelhető kivitelben gyártják. A második rendszer a kisfeszültségű Itanium 2 processzoron alapul, 2U és 4U magas kivitelben érhető el. Ennek a processzora 1,0 GHz-es órajellel és 1,5 MB L3 gyorsítótárral rendelkezik. Mindkét rendszer alacsony fogyasztású, ami csökkenti a méretezés költségeit.

➲ http://www.aspsys.com

Zend Studio 3.0

Megjelent a Zend Studio 3.0 IDE. A csomagot a kulcsfontosságú – PHP, JavaScript és HTML alapú – webalkalmazások fejlesztőinek szánják. A Zend Studio 3.0-s támogatja a PHP 5-öt, nagyobb teljesítményt ígér, és fejlettebbek a projektkezelő eszközei. A 3.0-s változatban új hibakereső eszközök jelentek meg, például egy távoli hibakereső, amely egy hibakereső URL-en keresztül lehetővé teszi az alkalmazások hibáinak megkeresését a böngészőn keresztül. Ezenkívül a 3.0-ban megtalálható a Code Profiler, amely megtalálja az alkalmazás szűk keresztmetszeteit, amik miatt lelassul az egész projekt. A Code Analyzer a statikus forrást elemzi, és ebben fedezi fel az esetleges hibákat. Újdonság még a továbbfejlesztett XML-beépítés, a fejlett kódkiegészítés, a testreszabható billentyűzetkiosztás, a teljes változatkezelés és a beépített FTP.
➲ http://www.zend.com

Linux Journal 2003, 116. szám

Bemutatkozik a Scribus

A Scribus teljes értékű asztali kiadványszerkesztő alkalmazás, amely a szövegszerkesztőkkel ellentétben a gépelési modell helyett a bemásolásmodellen alapszik.

A kérdés, úgy tűnik, örök: megfelelőnek tekinthető-e már a Linux az asztali gépekre? Ebben a vitában most újabb komoly igenlő érvet szolgáltat a Scribus 1.0-s jelenlegi kiadása. Gyakori aggály szokott lenni, hogy „nem tudom használni a Linuxot az asztali gépemen, mert nincsenek X-alkalmazások." A Scribus hatékonyan tölti ki a hiányzó űrt, a grafikus WYSIWYG (amit látunk, azt kapjuk) kiadványszerkesztő helyét. A Linux-felhasználók és Unix-használó unokatestvéreik mostantól egy módosítható, felhasználóbarát asztali kiadványszerkesztő alkalmazást vehetnek birtokba, amely újszerű megközelítéssel kezeli a Linuxot mint asztali kiadványszerkesztőt (DTP, vagyis Desktop Publishing). Ezenkívül a Scribus a hagyományos DTP-rendszereken túlmutató képességekkel is rendelkezik: képes PDF-állományokat (Portable Document Format, azaz hordozható dokumentumformátum), webes űrlapokat és interaktív PDF-dokumentumokat készíteni. Ezáltal az egyre csiszoltabbá váló lényeges támogatókönyvtáraknak (nevezetesen a freetype2, Ghostscript és CUPS rendszereknek) hála, a linuxos asztali kiadványkészítés lehetősége végre valósággá vált. A Scribus ügyesen egészíti ki a Linux- és a Unix-rendszerek néhány DTP-vonatkozású hiányosságát a PDF formátumban történő mentés széles körű használatával, illetve bizonyos mértékben a PDF-beolvasás támogatásával.
A PDF formátum, amely az Adobe tulajdona, de ellenszolgáltatás nélkül használhatják más fejlesztők is, rugalmas és megbízható formátumot nyújt, továbbá alkalmazások széles kínálatát támogatja. A PDF igen hasonló a PostScript nyelvhez és az Adobe kiválóan dokumentálta: a nyers PDF 1.5 referenciakézikönyv csekély 1000+ oldalra rúg, ráadásul a PDF-olvasó szinte minden korszerű rendszeren elérhető.

DTP alkalmazások vagy szövegszerkesztők

A legelemibb szinten való dokumentumkészítés a Scribus esetében kimerül az általános iskolában oktatott vágd és másold módszerben. Ha a rajzeszközt használjuk, úgy viselkedik, mint egy festővászon; ha képekkel dolgozunk, olyan, mintha másolólapot (pasteboard) használnánk. Szövegszerkesztővel a leginkább olyan munkálkodni, mintha egy nagyon okos írógépen tevékenykednénk. Talán úgy a legkönnyebb elképzelni a Scribust, mint egy a nagyszerű PDF-motor köré épített csinos kezelőfelületet. Ez a motor ugyanis nagymértékben leegyszerűsíti a nyomdakész, magas felbontású PDF-állományok és a teljesen interaktív PDF-fájlok készítését. A PDF-állományok létrehozásának egyik legnagyobb kihívása – különösen a nyomdakész PDF-ek esetében –, hogy ismerni kell jó néhányat az Adobe Acrobat Distiller száz desztillálóeszköze közül.
A Scribus egyértelműen nem szövegszerkesztő, sokkal inkább a kiadványszerkesztő (page layout) programok családjába tartozik. A jól ismert DTP-alkalmazások közé sorolható az Adobe PageMaker programja (az eredeti változat) vagy az InDesign és a Quark XPress. A Scribus azért különleges, mert GPL engedélyű, valamint mert egyetlen más, ilyen profi szintű képességekkel rendelkező DTP sem fut Linux alatt. A Scribus rendszert átvitték BSD-k, HP-UX és Solaris alá is. A Cygwin segítségével a Scribus kísérleti változata már Microsoft Windows 2000 alatt is fut, sőt a Mac OS X-es változat is készül már a GPL engedélyű Qt for OS X segítségével.
A kiadványszerkesztő alkalmazások abban különböznek a szövegszerkesztőktől, hogy a szöveg, a képek és a grafikák elhelyezésével sokkal nagyobb fokú pontossággal foglalkoznak, és a tervezőnek lényegesen több mindenbe lehet beleszólása, mint a korszerű szövegszerkesztők esetében, ahol ez nem oldható meg, s néha nem is kívánatos. A szövegszerkesztők a szavak „gyártásában" jeleskednek, de kizárólag a kiadványszerkesztő programok képesek a szöveget és a képeket, valamint az egyéb művészi munkákat egyszerűen és nagy pontossággal összeszerkeszteni. A Scribus például elég sok bitnyi olyan kódot tartalmaz, amely a PostScript-kimenet javításával foglalkozik, vezérli a színreprodukálást, szöveget tartalmazó képrétegeket készít és a nagyfelbontású művészi munkákat is kezelni képes.
A kiadványszerkesztő alkalmazásokban a szöveg is egy keretbe zárt objektum, akárcsak a képek vagy az alakzatok (dobozoknak – szövegdoboz, képdoboz – is nevezzük). Ez lehetővé teszi a szöveg pontos elhelyezését és folytonosságának megteremtését az oldalon. A Scribus segítségével könnyedén hozhatunk létre különleges hatásokat, például tükrözést (flipping) vagy valamilyen szögben futó szöveget, netán önműködő folytonosságú többoszlopos szöveget. A rétegeket vagy az objektumokat egymás fölé helyezhetjük (ezt maszkolásnak is nevezik), ilyen módon látványos képi hatásokat készíthetünk. A Scribus különleges tipográfiai eszközökkel rendelkezik, amelyekkel az egyes betűk kinézetét, illetve elhelyezkedését módosíthatjuk a szavakon belül (szaknyelven ezt alávágásnak – kerningnek – nevezik). Minden egyes objektum elhelyezését századhüvelyk (inch) vagy századmilliméter pontossággal tudjuk befolyásolni. Lássuk a Scribus néhány fontosabb képességét:

- Csúcstechnológiát képviselő, nyomdai előkészítésben is használatos, az ISO PDF/X-3 szabványnak megfelelő magas minőségű nyomdakész PDF-állományok előállítása. Ez a program az egyetlen, amelyik ezt közvetlenül támogatja.
- Teljes parancsfájlokkal ellátott, interaktív PDF-dokumentumok készítése, amely külső hivatkozásokat is tartalmazhat webhelyekre vagy bemutató PDF-fájlokra à la MS PowerPoint vagy OpenOffice Impress. Létrehozhatunk számított mezőket is, valamint a felhasználó által kitöltött űrlap adatait továbbíthatjuk a honlapunkra.
- Parancsfájlvezérlés a Python nyelv segítségével. A legtöbb komoly DTP alkalmazás rendelkezik valamilyen parancsnyelvvel. A Python rendkívül hatékony és a felülethez illeszkedő parancsnyelvvel ruházza fel a Scribus rendszerét.

A legtöbb más DTP alkalmazás saját parancsnyelvet vagy az AppleScriptet használja, ezek azonban nem hordozhatók. Ez a képesség lehetővé teszi azt is, hogy egy harmadik fél által készített Python-modulokat, például képkezelő modulokat futtassunk, vagy más Python-alkalmazásokat, mondjuk a PySolt indítsuk el.

- Teljes körűen támogatja a nyomtatásban használt CMYK (cyan, magenta, yellow és black, azaz cián, bíbor, sárga és fekete) színkezelést, a színrebontást, valamint a direkt színek (spot colors) használatát és behozatalát EPS (Encapsulated PostScript) fájlokon keresztül.
- A freetype2 segítségével támogatja az Unicode-szövegeket és betűkészleteket, és a jobbról balra író nyelveket, mint amilyen az arab és a héber. Jelen pillanatban a Scribus 19 nyelven jelent meg, a legutóbb cseh, orosz és indonéz változatban (természetesen létezik magyar változat is, a SuSE Linux és az UHULinux tartalmazza).

A Scribus letöltése és telepítése

A Scribus telepítése magától értetődő. Választhatjuk a forrásból történő fordítást, avagy letölthetjük a Scribus valamelyik harmadik fél által készített csomagját. A részleteket és a webhelyeket a Scribus leírásában találjuk. A forrás lefordítása a szokásos:

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

Nem árt, ha beszerezzük a terjesztésünkön elérhető legfrissebb változatot a Trolltech Qt, ***freetype2***, ***libtiff***, ***libpng*** könyvtárakból, valamint ghostscript rendszerből. A Scribus 1.0-s programját Qt 3.1.2-hoz fejlesztették, írásunk születésekor ugyanis ez a legfrissebb változat. Amennyiben a színkezelést is használni szeretnénk, szükségünk lesz a ***littlecms*** könyvtárakra és néhány ICC sablonra. A Scribus használatához továbbá kelleni fog a ***libtiff***, a ***libjpeg*** és a ***libpng*** könyvtár. A legnagyobb gond, amivel a fordítás során találkoztunk, az az, hogy a `QTDIR` környezeti változót nem vagy helytelenül állítottuk be. A leírás részletesen ismerteti a fordítás menetét.

A Scribus felhasználói felülete láthatóan DTP-alkalmazásra van kihegyezve: a segédvonalakat a vonalzókból húzhatjuk elő, a mértékegység egyetlen kattintással állítható. A státuszsorban rengeteg adatot, illetve az eszközök és objektumok pontos méreteit láthatjuk. Bár a Scribus bármely ablakkezelő alatt működik, KDE alatt olyan különleges képességei is akadnak, mint húzd és ejtsd az asztalon, valamint KDE-stílusú bővítményeket használ.

Beszélgetés K. és V. Franz Schmidttel, a Scribus fejlesztőivel

K.: *Miért láttál neki a Scribusnak?*

V.: Szükségem volt egy programra Linux alatt, ami képes menüket és kártyákat készíteni anyósom kis bavariai hotelének; csakhogy egyáltalán nem találtam a Mac gépemen futó DTP alkalmazáshoz hasonló semmilyen programot. Eredetileg kizárólag magamnak készítettem Python nyelven, míg végül egy barátom javaslatára kitettem a webre. Nagyon meglepett a válasz.

K.: *Hogyan jutott eszedbe a Scribus név?*

V.: Gondolkoztam olyasféle néven is, mint az „Open Page” (nyílt oldalak), de számomra nem volt elég egyedi. A Scribus a latin nyelvből származik: a római hivatásos írnokokat nevezték így, ahogyan mi is használjuk az angolban a scribes szót. Számos nyelvben van jelentése.

K.: *Miért a Qt-t választottad?*

V.: Amikor eldöntöttem, hogy átállok C++ nyelvre, a Qt volt az egyetlen, teljesen dokumentált C++-készlet. A Scribus volt az első C++-projektem, sőt még ma is az. A Python kiváló volt néhány elképzelés ellenőrzésére, pár szolgáltatáshoz azonban lassúnak bizonyult.

K.: *Kikből áll a Scribus csapata?*

V.: Nos, jelenleg belőlem és *Paul Johnson*-ból, aki a Scribus 0.8-as változata óta tag. Az anoncvssel kezdett, segített a kód átnézésében és rengeteg egyéb dologgal is ő foglalkozik, például a felhasználók támogatásával és a levelezőlistával. *Peter Linnell* korábban csatlakozott a kipróbálás és a dokumentálás területére. Néhány értékelhető adomány egyéb felhasználóktól is érkezett. Levelezőlistánk igen pezsgő, és kaptam már néhány kedves elektronikus levelet olyan felhasználóktól, akik értékelték a gyors választ. Akadnak olyan Scribus-használók is, akik nagyon aktívak a levelezőlistán, és ez segít, hogy több időt fordítsak a kódolásra. Valószínűleg egy kicsit kilógunk a nyílt forrású fejlesztések sorából, hiszen valamennyien a 30-as, 40-es éveinkben járunk. Valamennyiünknek állandó munkája és családja van. De a DTP-k esetében különösen jól jön, ha az ember rendelkezik némi műszaki tapasztalattal és előismeretekkel.

A DTP és a szövegszerkesztők képességeinek összehasonlítása

Képesség	DTP	Szövegszerkesztő
Tervezés	másolótábla (Pasteboard)	intelligens írógép
Automatizálás	parancsfájlok	makrók
Színtámogatás	CMYK és RGB	kizárólag RGB
Rajzeszközök	valós PostScript-vektor	metafájlok vagy alacsony felbontású bittérképek
Négyszínnyomás	színkezeléssel	nincs
Pontosság	magas – maximum 3600 dpi	képernyőfelbontástól függő
PDF	beépített kimenettámogatás	harmadik fél vezérlőin keresztül (OpenOffice 1.1 rendelkezik beépített PDF-kimenettel)

A színkeverés titkos receptje

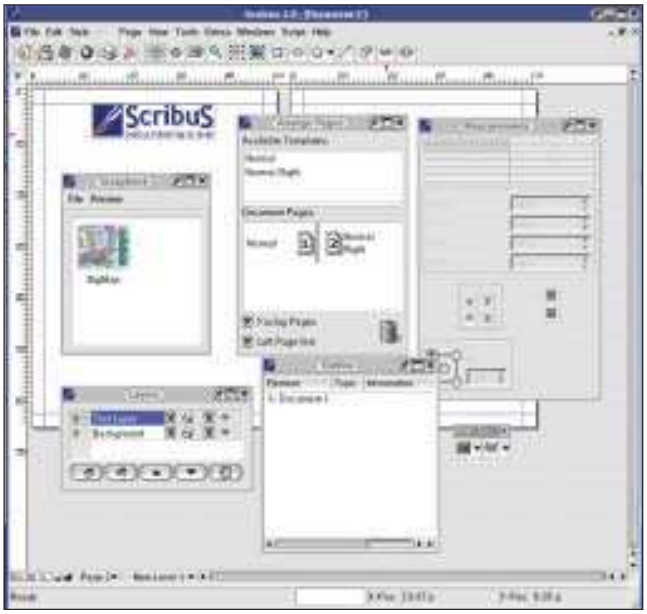
A Scribus egyik legfejlettebb lehetősége, hogy képes használni a littlecms színkezelő könyvtárakat. A LGPL engedély alatt kiadott littlecms csiszolt és könnyen módosítható csomag, amellyel sokféle színekkel kapcsolatos műveletet végezhetünk. Az eszközök végfelhasználója nagyra értékeli a nagy pontosságot és a folyamatos fejlődést.

A kereskedelmi színkezelő módszerek egészen mostanáig szigorúan őrzött titkok voltak. Hogyan tudja mindezt véghezvinni a littlecms? Először is léteznek szín- és színkeverő ISO-szabványok, amelyek nyíltak és az ICC kiadásában a ➲ http://color.org oldalon megtekinthetők. Másodszor, erősen leegyszerűsítve: az egyik színtérből a másikba való áttérést 3D-keresőtáblák végzik. A titkos recept egy, a színkezelő modul által használt algoritmus, amely a különböző színterek közti eltéréseket korrigálja. A feladatban az az igazi kihívás, hogy a színeket úgy kell az egyik színtérből a másikba átvinni, hogy mindeközben a színskála-átalakítás miatti színvesztést (gamut) a legkisebbre csökkentsük. Ez abból adódik, hogy a CMYK színtér általában kisebb színértékkel vagy tartománnyal dolgozik, mint amit az adott eszközön meg lehet jeleníteni. Például csodálatos zöld színeket lehet készíteni egy RGB monitoron, amiket igen nehéz visszaadni a nyomtatott kiadvány CMYK színeivel. A vastagság, fényesség és a papír festékbefogadó képessége egyaránt hatással van a nyomtató értékeire. A littlecms legfrissebb változatai tartalmaznak egy Black Point Compensation (fekete pont kiegyenlítés) nevű lehetőséget, amely lényegében egy újabb trükk, amivel a színeket úgy módosíthatjuk, hogy a képernyőt és a lapolvasót a lehető legközelebb vigyük a végső nyomtatási célon megjelenő képhez.

A littlecms nemcsak a cmm (színkezelő modul) modult tartalmazza, hanem további különféle színkezelésre szánt eszközöket is. Néhány parancssoros eszköz lehetővé teszi ICC profájlok elhelyezését és beállítását (tagging) képfájlokban, míg az eszközök másik csoportját monitorunk vagy lapolvasónk ICC profájljának előállításához használhatjuk fel. Elsőként el kell készítenünk a monitorunkhoz tartozó helyes ICC profájlt, amennyiben használható színkezeléssel szeretnénk kiegészíteni a Scribust. Tesztjeim során a littlecms monitor profájl jó eredményt mutatott, mint vizuális profájler (visual profiler). Az élvonalbeli DTP-k esetében a profájlkészítést és beállítást különleges felszereléssel és programokkal végzik, ami akár több ezer dollárba is kerülhet.

Mikor használjak TeX-et és mikor nyúljak a Scribushoz?

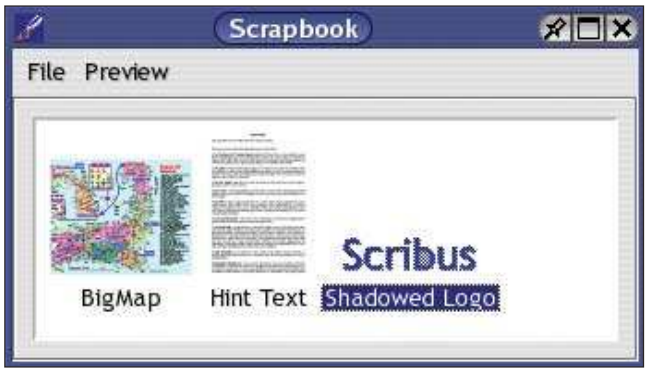
A Unix, illetve a Linux világában évekig a TeX-et és leszármazottait tekintették DTP-nek – nem is ok nélkül, hiszen a hosszú műszaki dokumentumok kiadásában, tudományos, matematikai és más karakterközpontú dokumentumok készítésében a TeX igen kiváló. A TeX nyomdakész állományok készítésére is képes – egész könyveket készítettek TeX alatt. Ugyanakkor, bár lehetőség nyílik képek és művészi alkotások beillesztésére is, a TeX nem éppen intuitív és nem is hatékony, ha erősen grafikus beállítottságú dokumentumokról van szó. A DTP lényegében az egyik legjobb felhasználási módja a WYSIWYG megközelítésnek. A TeX módszere erősen eltér a WYSIWYG DTP-k által alkalmazottól. Ez olyasmi, mint egy festményt HTML-kóddal leírni – nem lehetetlen, de nem is könnyű. A linuxos világban szinte egészen biztosan szükség és hely is van mindkettőre.



1. kép Scribus-eszközök – a Scribusban könnyen használható eszközkészleteket találunk, amelyek szolgáltatásait tooltipek ismertetik. Az óramutató járásával megegyező irányban, felülről: *1.* A Page eszközkészlet kezeli a mesteroldalakat és az oldalakat. Új oldal felvétele egyszerűen húzd és ejtsd módszerrel történik. *2.* A Measurements eszközkészlet a gyakran használt szerkesztőeszközöket tartalmazza. *3.* A PDF-eszközkészlettel különböző eszközöket (toolbars) húzhatunk be a dokumentumba. *4.* Az Outline a dokumentum összes objektumának fadiagramja, segítségével összetett dokumentumokban is eligazodhatunk. *5.* A Layers segítségével objektumokat halmozhatunk egymásra, fantasztikus hatásokat hozva létre. *6.* A Scrapbook (vázlattömb) a gyakran használt elemeket tartalmazza

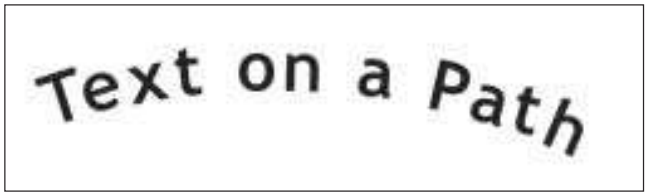
Mesterművünk létrehozása

A szövegszerkesztőkkel szemben itt nem szokás egyszerűen új üres dokumentumot nyitni és elkezdeni gépelni. Először is készítenünk kell egy szövegdobozt aztán vagy gépelni kezdünk, vagy betöltünk valamilyen szöveget. A magam részéről inkább szövegszerkesztőt használok, szerkesztek, ellenőrzöm a helyesírást, majd mentek és beolvastatom a szöveget. Hasonlóképpen a Gimpet sem nyithatjuk meg csak úgy, hogy aztán másolunk és beillesztünk. Először készítenünk kell egy képdobozt, majd a képet fel kell tennünk az oldalra. A képállományok igen nagyok is lehetnek, ezért – a gyorsabb működés érdekében – egy előnézeti kép jelenik meg a dokumentumban, egy láthatatlan hivatkozással az eredeti képállományra. Ezek a hivatkozások csak minden újbóli megnyitáskor frissülnek (munka közben nem, ha például Gimppel módosítjuk az eredeti fájlt), ez egyben garancia arra is, hogy a dokumentumunkban annak megnyitása után mindig a legfrissebb képváltozatot találjuk. Az általánosan használt objektumok nyomon követését jelentősen megkönnyíti a fényképalbum.



2. kép Felhasználóbarát – a Scrapbook segítségével könnyedén újrahasznosíthatjuk objektumainkat az oldalak és dokumentumok között. Vázlattömbjeinket külön is menthetjük, valamint a projektekhez vagy dokumentumokhoz csatolhatjuk. Itt egy teljes oldalt elfoglaló EPS-t, egy teljes oldalt kitöltő szöveget és a logót mutatjuk be. Szinte bármilyen objektum menthető, ha az oldalról a jobb gombbal a vázlattömbre húzzuk

3. kép Fejlett PDF-kimenet: képes módosítani a betűkészletek körvonalait és az objektumokon belül éles, magas felbontású, rétegekkel ellátott átmeneteket készíteni, valamint kezeli az átlátszóságot is. Az ilyesfajta kimenet fejlett Level 3 PostScript vagy PDF 1.4 műveleteket igényel, amilyeneket ez idáig kizárólag pénzes DTP-alkalmazásokban láthattunk (a pillanatfelvétel a Scribus 1.1.0 változatról készült)



4. kép A Scribus egyik legkönnyebben használható eszköze a tetszőleges görbére írható szövegkészítés. Ezáltal a felhasználó szövegei alaktalan objektumokat is követhetnek, amit a szöveg díszítésére, vagy például egy diagram útjainak az elnevezésére használhatunk fel

Kiadványunk elkészítésének megkönnyítése érdekében a Scribus néhány hasznos és felhasználóbarát rajzolóeszközt tartalmaz: különféle stílusú vonalakat, poligonokat, Bézier-görbéket rajzolhatunk, a megrajzolt objektumokat és átmeneteket színezhetjük és árnyalhatjuk (blending) – ezzel a módszerrel létrehozhatjuk az objektumok árnyékait, vagy kiemelhetjük térbeliségüket. Ezenkívül a Scribus képes a betűtípusok átalakítására is az illusztráló és rajzolóprogramok (CorelDraw, Adobe Illustrator stb.) által is alkalmazott módszerrel: a betűket PostScript-görbékre vagy -körvonalakra bontja, amelyeket aztán tetszésünk szerint csavarhatunk, méretezhetünk, hajlíthatunk és nyújthatunk. Lehetőségünk nyílik ezeket színátmenetekkel vagy képekkel kitölteni, illetve a szöveget görbékhez rendelhetjük. A Scribus képes a szöveget és a képeket SVG formátumban menteni, az SVG-fájlokban pedig minden képet PNG-állománnyá alakít.

Felhasználói testreszabhatóság

A DTP-felhasználók hozzászoktak a gyakori gyorsbillentyű-használathoz – ez alól a Scribus sem kivétel. A függvényeket, menüelemeket felhasználói gyorsbillentyűkkel érhetjük el, és jelentős mennyiségű, a jobb gombbal előhívható, helyérzékeny (context) menüt használhatunk. Betűkészlet-elérési utakat vehetünk fel vagy törölhetünk, illetve betűhelyettesítést adhatunk meg menet közben. De betűkészleteink telepítésekor óvatosan járjunk el, s ne felejtsük el frissíteni a ***fonts.dir*** és ***fonts.scale*** állományokat. A Scribus elég „nyűgös”, ha a betűkészletek elérési útja hibás és számos kódot tartalmaz, amely megpróbálja kiszimatolni a telepített betűkészleteink pontos adatait. Az új felhasználók általános hibája, hogy azt hiszik: a Scribus nem találja meg az általuk felrakott friss betűkészleteket. A Scribus makacsul elutasít minden betűkészletet, ami nincs száz százalékig rendben. Ennek okát a dokumentáció részletesen taglalja. A betűkészletek ugyanis bizonyos értelemben olyan apró programok, amelyek természetesen hibásak is lehetnek.

Szövegkezelés

A Scribus egyaránt képes ASCII szöveges állományok és a vágólap szövegének az átvételére, ezenkívül lehetőségünk nyílik beállítani a szöveg kódolását is. Így használhatunk Unicode-szöveget, de akár cirill vagy latin kódolásokat is. Ha szükséges, a szöveg irányát is megváltoztathatjuk a jobbról balra író nyelvek (arab vagy héber) esetében. A következő változatokban várhatóan újabb beviteli szűrők fognak megjelenni.
A különleges karakterkészlet-kezelő szolgáltatások lehetőséget adnak a szöveg forgatására, tükrözésére és átméretezésére. A Scribus több nyelven is képes a szavakat helyesen és önműködően elválasztani, illetve akár az alávágási (kerning) értékeket is megváltoztathatjuk. Ha egy adott karakterkészlethez AFM (ASCII font metrics) állományt veszünk fel, a Scribus ezt fogja önműködően használni a betűk közötti helyek beállításához. A Scribus továbbá egyaránt képes a Type 1, valamint a TrueType betűtípusok kezelésére.

Képkezelés

A Scribus ismeri a legtöbb általánosan használt képformátumot (PNG, TIFF, JPEG). Bármely képet átméretezhetjük, elforgathatjuk, tükrözhetjük vagy rétegezhetjük. Amennyiben a képben található beágyazott ICC színsablon, a Scribus be tudja olvasni a tagokat és a színértékeket, és a ***littlecms*** színkezelőben tárolja őket. Könnyedén beolvashatunk akár

EPS-állományokat, vagy akár egy PDF első oldalát. Az EPS- és PDF-objektumok könnyebb elhelyezhetősége érdekében a Scribus önműködően egy alacsony felbontású előnézetet állít elő. Az EPS-állományoknak egyébként alapesetben nincsen előnézetük.

A PDF- és PostScript-kimenet

A Scribus rendelkezik az egyik legjobb PDF-mentő motorral az egész bolygón – legalábbis ezt állítja a szlogenje. Véleményem szerint ez azonban csak a második az Adobe InDesignja után. A PDF-mentő motor használata nemcsak könnyű, de kiváló minőségű PDF-fájlokat menthetünk vele, feltéve, hogy jó minőségű képeket és betűkészleteket választunk munkánkhoz. A PDF-kezelőben az a zseniális, hogy elrejti a zavaró beállításokat, ugyanakkor a profi felhasználóknak lehetőséget biztosít a teljes beállításra, mint például a PDF/X-3. Kipróbálásához azonos kép- és betűkészlet felhasználásával, nagy méretű és felbontású, négyszínes CMYK PDF-állományokat készítettem a Scribusszal és más élvonalbeli DTP-alkalmazások segítségével. A különbség észrevehetetlen volt. A színek, a betűk, a képek és a szövegek pontosan ugyanúgy jelentek meg, mint bármelyik másik DTP-s alkalmazásban készített PDF-állományban. Teljes mértékben támogatja a PDF 1.4 képességeit, a 128-bites titkosítást, és a PDF-en belüli átlátszóságot. A Scribus képes jó minőségű EPS és RAW PostScript-állományokat is menteni. (A PostScript a nyomdai előkészítés esetében a legfontosabb formátum.)

A Gimp és a Scribus

A Gimp használatóságát a nyomdai előkészítés és a DTP világában részben éppen a CMYK-támogatás hiánya okozza. Egyszerűen szólva: a Gimp RGB rendszerben vagy szürkeskálával képes dolgozni. A nyomtatási világ CMYK rendszert használ – a négyszínnyomásban használt négy fő festék színét. Ha beleolvasunk a Gimp levelezőlistájába, gyaníthatjuk, hogy talán a Gimp 2.0-s környékén lesz már ilyen lehetőség, hiszen elég régóta fennálló kérésről van szó. Elég sokat vitatkoztak CMYK-színekkel kapcsolatos jogvédelmi kérdésekről is. Számomra ez teljes képtelenségnek hangzik. Mint a Scribus és a *littlecms* könyvtárak bizonyítják, maguk a CMYK-színek semmiféle szabadalom alá nem esnek. Amíg nem használunk védett eljárásokat, tiszták vagyunk. Szerencsére létezik néhány megoldás, amelyekkel Gimpünket a CMYK világában is használhatjuk:

1. A Scribus az RGB színeket önműködően átalakítja CMYK színekké, amennyiben a PDF exportálásakor a nyomtatási beállításokat nyomtatásra (printing) állítjuk.
2. *Alistair Robinson*, aki korábban a Scribushoz is küldött kódot, készített egy egyszerű, de hatékony módszert, amellyel CMYK TIFF-állományokat menthetünk a Gimppel, az RGB csatornák ügyes szétválasztásával és a szürke csatornát a CMYK-színekhez felhasználva. A Separate Gimp-bővítmény a ➲ http://www.blackfiveservices.co.uk/separate.shtml címről tölthető le.

Amennyiben Linux alatt CMYK-támogatást szeretnénk, két további lehetőségünk is van: a Corel PhotoPaint 9 még mindig letölthető az FTP-ről. Ingyenes, mint az ingyen sör, de támogatás nem jár hozzá és az újabb terjesztéseket elég nehéz telepíteni. A Caldera Graphics által kínált Cameleo termékből egyaránt beszerezhetjük az ingyenes (light) vagy az üzleti változatot, amely lapolvasó és képátalakító eszközöket is tartalmaz. Mindkettő rendelkezik ICC színkezelő támogatással. A Scribus leírásában további részletes ismertetést találunk.

Az XML ereje

A Scribus dokumentum- és beállításfájljai XML alapúak, teljesen nyíltak és dokumentáltak. Ez nagymértékben megkönnyíti az új lehetőségek hozzáadását, szövegalapú természetük pedig hibatűrővé teszi őket. A DTP-fájlformátumok az egész PC-s világ legbonyolultabb formátumai közé tartoznak. A fájlok meghibásodása gyakran okoz gondot, a DTP-állományok a hálózaton is megbízhatatlanná válhatnak. Úgy találtam, hogy a Scribus által alkalmazott XML formátum szinte teljesen összeomlásbiztos. Még a Scribus fejlesztői, CVS-változatával is csak nagyritkán vesztettem el állományt, a hibásakat pedig egy egyszerű szövegszerkesztővel megjavíthattam. A PageMaker és a Quark-felhasználók már régóta használják a gyakori mentéseket és néhány Novell-trükköt sérült fájljaik helyreállítására.

A Scribus a nyomdavilágban

A Scribus projekt indulásakor hozott döntés, amelynek szellemében a projekt elsősorban a PDF formátumra összpontosít, igen előrelátónak bizonyult. Sok évnyi tévelygés után, a nyomdai világ végül a PDF-et is elfogadja fájlcsereformátumként. (A Linux Journal PDF-ként küldi a nyomdába a

Ügyeljünk a DPI-kre!

A kezdők gyakori hibája a képfelbontás helytelen megválasztása. A weben a legtöbb kép 72–96 dpi (dots per inch) felbontású. Amikor DTP-állományokat készítünk, gyakorta sokkal nagyobb felbontásra van szükségünk. Általában minimum 200 dpi-re, de gyakran akár 300 dpi-ig is elmehetünk. Például ez a magazin optimálisnak mondható 300 dpi-vel készült. Vannak állományaim, amelyeket a Gimpből nem kevesebb mint 1200 dpi körüli értékkel mentettem, kiváló eredménnyel.

KAPCSOLÓDÓ CÍMEK

Linux DTP-hivatkozások
➲ http://www.atlantictechsolutions.com/scribusdocs/sclinks.html
Littlecms
➲ http://www.littlecms.com
Scribus Documentation On-Line
➲ http://www.atlantictechsolutions.com/scribusdocs
➲ http://home.comcast.net/~scribusdocs
Scribus-honlap
➲ http://web2.altmuehlnet.de/fschmid
➲ http://scribus.planetmirror.com
Scribus-levelezőlista
➲ http://nashi.altmuehlnet.de/mailman/listinfo/scribus

lapszámokat, mi a Linuxvilágnál PS-ben küldjük, természetesen PDF kíséretében – szerk.) Nem szokatlan, hogy a pályázati anyagokat PDF-állományban kérik a pályázóktól. Akadnak magazinok, amelyek teljes egészükben PDF-ben készülnek, és végül a kész kiadványt ismét PDF formátumban adják ki. Miért? Mert a PDF-ben a hibás betűkészleteket vagy a nem megfelelő betűtípust értelmezhetetlen karakterekkel jeleníti meg. Továbbá ma már kiforrott előtesztelő eszközök léteznek, amelyekkel ellenőrizhetjük a PDF nyomdakészségét. A PDF/X ISO szabvány megjelenése szintén nagyot lendített a PDF elfogadottságán.

Azt mondják, a Scribus soha nem is készült „Quark-gyilkosnak". A Scribus alkotóinak célkitűzése: használható eszközt adni a Linux- és Unix-felhasználók kezébe, ami egészen mostanáig kizárólag a Windows és Mac operációs rendszereken elérhető, drága kereskedelmi alkalmazások kiváltsága volt.

A DTP-felhasználók és vállalkozások kicsit hagyománytiszteletlőbbek, ha az alkalmazáscserék és fejlesztések kerülnek szóba. A megbízhatóság kényes kérdés, hiszen egy hibásan kifutó kiadvány milliókba is kerülhet. Ráadásul ezeket az alkalmazásokat hónapokba, néha inkább évekbe telik elsajátítani. Az átállás a termelékenység rovására mehet, mégha egyazon alkalmazás újabb változatát használjuk is.

A PDF/X-3-nak Scribus alatti támogatása újabb lépés afelé, hogy a Scribus-állományokat a nyomdai világ elfogadja.

A legújabb raszterkép-feldolgozók (RIP-ek) támogatják a PDF/X-3 formátumot, amely egészen az utolsó pillanatig megőrzi az RGB (vörös, zöld, kék) képeket és az ICC sablonokat: amikor az állományokat nyomólemezekké alakítják.

Összegzés

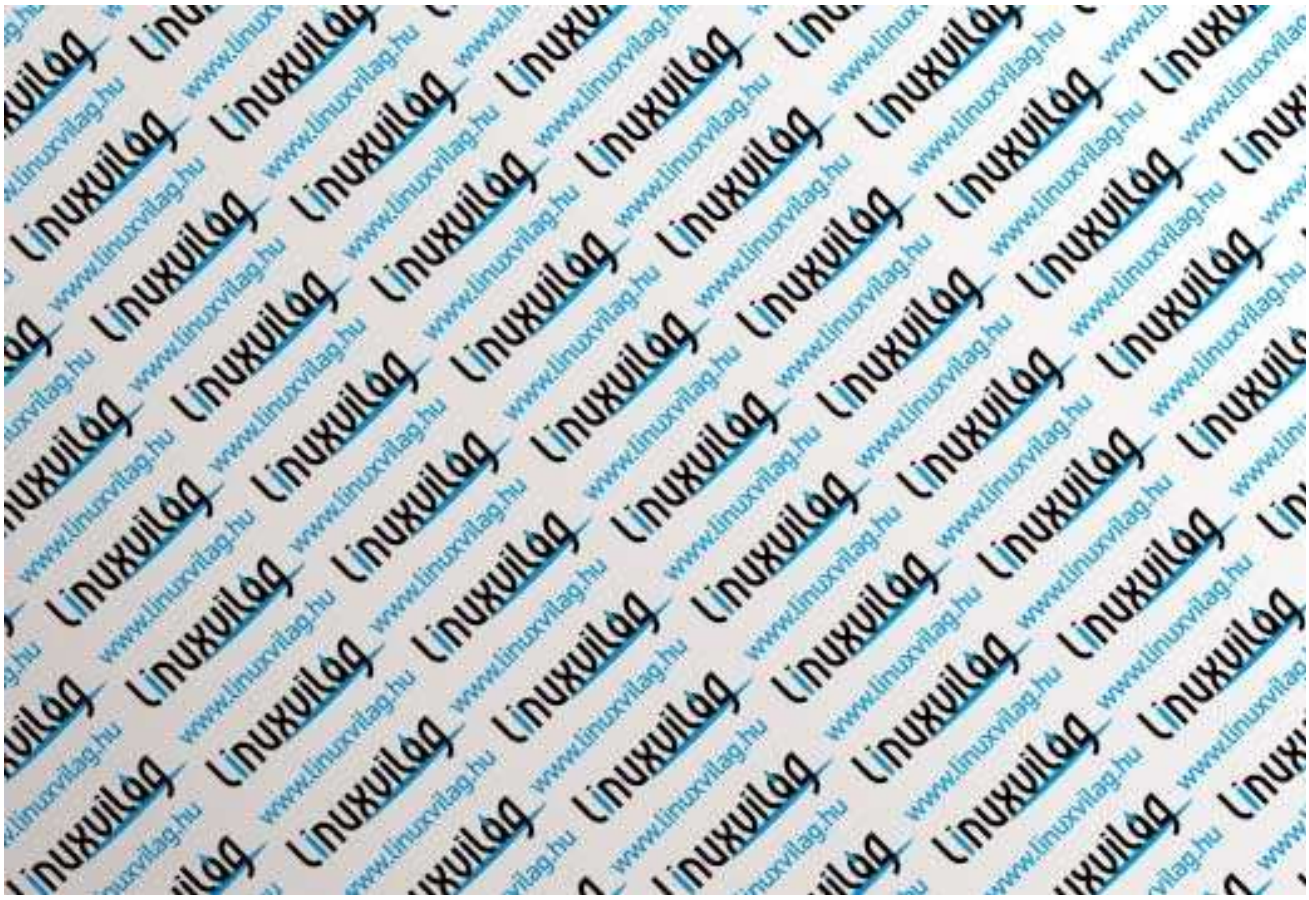
Tekintettel arra, hogy kezdő linuxos vagyok és a programozáshoz sem értek, a dokumentáció átolvasásának elvállalása számomra az egyik legértékesebb tapasztalat volt a számítógépes világban. Nekem, aki a Windows/Mac/Novell világból érkeztem, ez adott sokkal kifinomultabb áttekintést a Linux/Unix módszerekről. A seregnyi kiváló, nyílt forrású projekt fejlesztői szívesen látnak némi segítő szándékkal készült dokumentációt és tesztet. Mi, nem programozók így adhatunk nekik vissza valamit, cserébe azért a rengeteg programért, amit olyan szabadon élvezhetünk.

A DTP Linux alatti története meglehetősen rövid. 2000-ben az Adobe nyilvánosan próbálta ki a ***Framemaker*** próbaváltozatát, amely csak bizonyos Unix-fajtákon futott – aztán eltűnt. Rövid idő múlva egy Chilliware nevű cég ajánlotta fel Ice Sculptor nevű DTP-alkalmazását. A cég nem sokkal a kiadás után bezárta kapuit. Minthogy a DTP bizonyos szempontból hiányzó (niche) alkalmazáskategória, a Scribus új utakat nyit a Linux-környezetben.

Linux Journal 2003. november, 115. szám

Peter Linnell

IT-tanácsadó és a New England-i Atlantic Tech Solutions főnöke, szakterülete a hálózatok, a nyomdai előkészítés és a DTP-k. Saját magát „Windows-szökevénynek" titulálja, és a Scribus Project az első olyan nyílt forrású projekt, amelyben részt vesz. Keményen dolgozik a Red Hat Engineers' certification megszerezésén.



Gentoo Linux-telepítés: „végigjátszás”

A Gentoo Linux első korongját előző számunkban már kézhez vehették kedves olvasóink, most a második lemezt adjuk közre. A kíváncsiak, türelmetlenek és egyben hozzáértők már bizonyára kipróbálták, mire is képes a Gentoo Live CD – lehet, hogy már a gépükön csücsül, és ezt használják eddig megszokott Linuxuk helyett. Amennyiben még nem volt idejük, bátorságuk, kedvük hozzáfogni a nem grafikus sallangokkal teletűzdelt telepítési folyamatnak, ennek az útmutatónak a segítségével remélhetőleg mindenki számára elérhető közelségbe kerül egy olyan Linux telepítése, amit nem mindennapi emberek készítenek nem mindennapi embereknek…

A rendszerkövetelmények

Általánosságban elmondható a Gentoo Linuxról is, hogy 486+ processzor, 64 MB memória és 64 MB csereterület (swap) szükséges a telepítéséhez. Ez azonban tényleg annyira minimum, hogy csak a mazochisták próbálkozzanak meg vele, mivel egy ilyen gépen szörnyen lassú lesz a telepítés, de maga a rendszer is. A CD-n közreadott változat i686-os gépeken működik.

A rendszerindítás

Az első korongot a CD-ROM meghajtójába helyezve elindíthatjuk a Gentoo Linuxot – biztosan van olyan olvasóink között, aki ezt már meg is tette.
A rendszerindítást a különböző jellemzőkkel, illetve a rendszermagváltozat kiválasztásával nagymértékben testre tudjuk szabni; *1.* és *2. táblázatunkban* ezeket láthatjuk felsorolva.
Ha nem csinálunk semmit, csak elindítjuk a gépünket a CD-ről, akkor a `gentoo` rendszermag fog elindulni, ami az alapértelmezett választási lehetőség, és a legtöbb rendszer tulajdonosának ez a választás tökéletesen meg is felel. Ha esetleg csak egy sötét képernyőt látnánk, akkor indítsuk újra a gépen a `nofb` maggal – ekkor a rendszermag kerettár-támogatás (framebuffer) nélkül indul majd, ami régi vagy különleges grafikus eszköz esetén jelenthet „menekülési” utat. Az `smp` rendszermag többprocesszoros rendszerekhez kínál remek támogatást, az `acpi` mag pedig rendszerünk energiafelhasználásának takarékossági támogatásában lehet a segítségünkre. Végül a `memtest` egy memóriatesztelő program, ami akkor segít, ha memóriamoduljaink megfelelőségéről szeretnénk megbizonyosodni.
A rendszermagok finomhangolására a 2. táblázatban találhatunk különféle megoldásokat. Ezek közül az, ami számomra szükségesnek bizonyult, a `doscsi` kapcsoló volt, mivel egy régebbi gépen nem volt hajlandó felismerni az SCSI kártyát; illetve használtam még a `nofb` rendszermagot egy régi S3-as VGA kártyához is.
A tényleges folyamat: berakjuk az első CD-t a meghajtóba (a BIOS-ban remélhetőleg be van állítva a CD-ről történő rendszerindítás).
Ha lenyomjuk az F2 billentyűt, akkor a 2. táblázatban leírtakat olvashatjuk angolul, ugyanez a helyzet az F3-as billentyűvel, ennek a lenyomásával a 1. táblázatban leírtakhoz juthatunk hozzá, szintén angol nyelven.

1. táblázat

Rendszermagok	leírás
`gentoo`	A „normál” `gentoo`-rendszermag (alapértelmezett).
`nofb`	Kikapcsolt kerettár (framebuffer) mód.
`smp`	`smp` rendszermag kikapcsolt kerettármód.
`acpi`	Engedélyezi az `acpi`-t, és betölti az `acpi` modulokat.
`memtest`	A memóriatesztelő program indul el.

2. táblázat

Indítási kapcsolók	Leírás
`doataraid`	IDE RAID-vezérlők betöltése.
`dofirewire`	Firewire-vezérlők betöltése.
`dokeymap`	Billentyűzetkiosztás-választó.
`dopcmcia`	PCMCIA-vezérlők betöltése.
`doscsi`	SCSI eszközök keresése (néhány hálózati kártya nem működik vele).
`noapm`	Letiltja az `apm` modult.
`nodetect`	A hwsetup/kudzu és a hotplug programok nem futnak le a rendszerindulásnál.
`nodhcp`	Nincs önműködő DHCP-kérés indulásnál nohotplug nincs hotplug támogatás (pl. USB).
`noraid`	Letiltja az `evms` modult.
`nousb`	Nincs USB-támogatás.
`ide=nodma`	DMA tiltása, néhány nem teljesen szabványos IDE eszköznél használjuk.
`cdcache`	A teljes rendszert a memóriába tölti, ez kb. 40 MB memóriát foglal el, a segítségével lehetőség nyílik a CD-k cseréjére.

3. táblázat

Lemezrész	Méret	Típus	Lehetséges eszköz
Indítás (boot), tartalmazza a rendszermagot és a rendszerindítási adatokat.	32 MB	`ext2`, illetve `ext3` a javasolt fájlrendszertípus, ha ReiserFS-t akarunk használni, akkor a `-o notail` kapcsolóval kell befűzni (mount) a rendszerbe. Ha `ext3` vagy ReiserFS mellett döntünk, akkor számolnunk kell a naplóállomány méretével is, így inkább 64 MB-ot adjunk neki.	/dev/hda1
Csereterület (swap) (most már 2 GB a mérethatár a régebbi 128 MB-tal szemben).	memóriánk méretének az 1–2×-esét válasszuk	Linux swap	/dev/hda2
A / lemezrész a */usr*, */home*, */etc* stb. könyvtárakat tartalmazza.	>=1,5 GB	ReiserFS, `ext3` ajánlott, `ext2` elfogadható	/dev/hda3


2.

Nyomjunk ENTER-t, ha minél előbb bele szeretnénk kezdeni a rendszer felfedezésébe. Ha valami az első indulásnál nem jól sikerült, akkor a megadott beállításokat kombinálva próbáljuk meg a rendszert szóra bírni. Én személy szerint csak a fentebb említett SCSI- és kerettárhibába ütköztem bele.
Miután a rendszer felállt, az *1. képen* látható képernyő fogad bennünket.
Mint látható, rendszergazdaként vagyunk bejelentkezve, jelszavunk nincsen, hiszen úgysem tudnánk tárolni a CD-n. Beállíthatunk ugyan jelszót, de az minden újraindításkor elvész, ha CD-ről indítjuk a rendszert.

A merevlemez felosztása

A Gentoo Linux leírásában az `fdisk` programot ajánlják erre a feladatra, én azonban jobban szeretem a `cfdisk`-et, mivel jóval felhasználóbarátabb és gyorsabban használható (igaz, néha csak az `fdisk` segíthet). Én egy 2 GB-os merevlemezre telepítettem a Gentoót, ennek a felosztása látható a *2. képen*.
Készítettem egy megközelítőleg 50 MB-os indítólemezrészt, egy 256 MB-os csereterületet (swap), a maradékot pedig megkapta a rendszer maga. Természetesen ez csak útmutatás, ettől nyugodtan el lehet térni, bár az indítórészt nem érdemes nagyobbra hagyni.
Az elsődleges IDE merevlemez a */dev/hda*, az első SCSI merevlemez pedig a */dev/sda*, így a `cfdisk /dev/a merevlemezünk eszközneve` paranccsal tudjuk a felosztását elkezdeni. A lemezrészek méretére nézve néhány irányelvet tartalmaz a *3. táblázat*.

A lemezrészek formázása

A Gentoo Linux támogatja az `ext2`, `ext3`, XFS, JFS és ReiserFS fájlrendszereket. Ezek közül mindenki kiválaszthatja, hogy melyiket akarja használni; természetesen érdemes naplózott fájlrendszert használni, ezzel is bebiztosítva magunkat az adatvesztés ellen.
A különböző parancsok különböző fájlrendszereket hoznak létre az adott eszközön:

ext2: `mke2fs /dev/hda1`
swap: `mkswap /dev/hda2`
ReiserFS: `mkreiserfs /dev/hda3`

Mint látható, az előbb felosztott merevlemezen az indítólemezrészt `ext2`-re, a csereterületet `swap`-ra, a / lemezrészt pedig ReiserFS-re formáztuk. Ha `ext3`-ra akarjuk valamelyiket formázni, akkor használjuk a `mke2fs -j /dev/hda3` parancsot, ugyanez a parancs, ha `xfs`-t szeretnénk:

```
mkfs.xfs /dev/hda3
```

A lemezrészek életrekeltése

Először keltsük életre a csereterületet (swap) a `swapon /dev/hda2` paranccsal. Ha most kiadjuk a `top` vagy `free` parancsot, a visszakapott adatok között virtuális memóriánknak is meg kell jelennie. Ezután a / (root) lemezrészt fűzzük be (mount) a */mnt/gentoo* könyvtárba:

```
mount /dev/hda3 /mnt/gentoo
```

hozzunk létre egy *boot* nevű könyvtárat:
`mkdir /mnt/gentoo/boot`

és fűzzük be ide az indítólemezrészt:
`mount /dev/hda1 /mnt/gentoo/boot`

A telepítés

Mi most a legegyszerűbb telepítési folyamatot mutatjuk be, nem fordítunk csomagokat, hanem az előre elkészített csomagokból fogjuk a rendszert összeállítani. Ehhez szükségünk lesz a ***stage3-i686-20030910.tar.bz2*** csomagra, ezt a korongon a ***stages*** könyvtárban találjuk. Lépjünk be a */mnt/gentoo* könyvtárba, és adjuk ki a `tar -xvjpf /mnt/cdrom/stages/stage3-i686-20030910.tar.bz2` parancsot. Ennek következményeképpen ez a közel 140 MB-os tarlabda

kicsomagolódik a merevlemezre. Ezután a ***snapshot*** könyvtárban lévő ***portage-20030911.tar.bz2*** fájl kicsomagolása következik a ***/mnt/gentoo/usr*** könyvtárba, ezt a feladatot nagyon remekül megoldja a `tar -xvjf /mnt/cdrom/snapshots/portage-20030911.tar.bz2 -C /mnt/gentoo/usr` utasítás. Most már csak két parancsnyira vagyunk a csomagok másolgatásának a végétől:

```
cp -R /mnt/cdrom/distfiles
➥/mnt/gentoo/usr/portage/distfiles
cp -a /mnt/cdrom/packages
➥/mnt/gentoo/usr/portage/packages
```

Chroot

Ezennel elérkeztünk a telepítés legizgalmasabb részéhez: bemegyünk a „kaptárba”.
Ahhoz, hogy `chroot` környezetben tudjunk dolgozni, a következő dolgokra lesz szükségünk:

- először is felélesztjük a ***/proc*** fájlrendszert:
  `mount -t proc proc /mnt/gentoo/proc`
- átmásoljuk a ***resolv.conf*** fájlt az új rendszerre:
  `cp /etc/resolv.conf /mnt/gentoo/etc/resolv.conf`
- chrootolunk az új rendszerbe:
  `chroot /mnt/gentoo /bin/bash`
- frissítjük a környezetünket:
  `env-update`
- végül beolvastatjuk a profájlt:
  `source /etc/profile`

Alapvető beállítások

Az időzónát az `ln -sf /usr/share/zoneinfo/Europe/Budapest /etc/localtime` paranccsal állíthajuk be.
Ezután bármit csinálunk, legalább a pontos időt nyomon tudjuk követni. Létre kell hoznunk a ***/etc/fstab*** fájlt is, ami a rendszerben található eszközök csatolási pontjait adja meg a rendszer indulásakor. A fentebbi példáknál maradva, a ***/dev/hda*** merevlemezt és a fentebbi felosztást feltételezve a *4. táblázatban* látható állapot a megfelelő.

A rendszermag telepítése

Adjuk ki az `emerge -k sys-kernel/gentoo-sources` parancsot a rendszermagforrás feltelepítéséhez, majd a rendszermag önműködő létrehozásához használjuk a `genkernel` eszközt. Természetesen nem ez az egyedüli útja rendszermagunk elkészítésének, de hely hiányában most csak a legegyszerűbb és leggyorsabb változatot vesszük szemügyre. Ahhoz, hogy a `genkernel`-t használni tudjuk, először telepítenünk kell:

```
emerge -k genkernel
```

(biztosan feltűnt már a kedves olvasóknak, hogy egy csomag telepítését az `emerge` paranccsal tehetjük meg).
A `genkernel` futtatása után egy rendszermagot, modulokat és egy `initrd`-t (initial root disk) kapunk.

Fontos programok telepítése

Rendszerünk egészséges életéért szükségünk lehet néhány olyan programra, ami átsegít a nehéz napokon. Ilyen program például a `syslog`, ennek telepítése és élesítése az `emerge -k app-admin/sysklogd; rc-update add sysklogd default` parancsokkal lehetséges. Szükség lehet még fájlrendszer-karbantartó és segédprogramokra: `emerge -k reiserfsprogs`.

4. táblázat

/dev/hda1	/boot	ext2	noauto,noatime	1 2
/dev/hda3	/	reiserfs	noatime	0 1
/dev/hda2	none	swap	sw	0 0
/dev/cdroms/cdrom0	/mnt/cdrom	iso9660	noauto,ro,user	0 0
none	/proc	proc	defaults	0 0

A rendszerindító program telepítése és beállítása

Telepítsük a GRUB-ot – én azért részesítem előnyben ezt a programot a LILO-val szemben, mert ezzel a laptopom is kiválóan működik, tehát elég sok időt töltöttem el vele ahhoz, hogy megszokjam és megszeressem. Adjuk ki a `emerge -k grub`, majd a `grub` parancsot. Most a GRUB parancsértelmezőjében találjuk magunkat, itt az előző merevlemez kiosztásnál maradva a következő parancsokkal telepíthetjük a rendszerindító programot.
A rendszerindítási lemezrész, ahová a GRUB fájljai is kerülnek:

```
root (hd0,0)
```

A merevlemez rendszerindító lemezrészébe (MBR) írjuk be magunkat: `setup (hd0)`, majd kilépünk a `quit`-tel.
Most hozzunk létre egy ***/boot/grub/menu.lst*** fájlt a következő tartalommal:

```
default 0
timeout 30
splashimage=(hd0,0)/boot/grub/splash.xpm.gz

title=Gentoo Linux (genkernel)
root (hd0,0)
kernel (hd0,0)/boot/kernel-KV root=/dev/hda3
initrd (hd0,0)/boot/initrd-KV
```

Ha eddig mindennel elkészültünk, rendszerünk remélhetőleg használható állapotba került. Akkor most adjuk ki a következő parancsokat:

```
etc-update
exit
cd /
umount /mnt/gentoo/boot
umount /mnt/gentoo/proc
umount /mnt/gentoo
reboot
```

Ezután távolítsuk el a korongot a meghajtóból, és lássuk, mi is történik.
A Gentoo Linux csomagtelepítési módjáról a 6. oldalon Programvadászat című írásunkban olvashatnak az érdeklődők.

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

A barátságos keselyű

A számítógéptelepek és a Grid egyre nagyobb jelentőséggel bír az informatika világában. A Condor ennek az egyik megvalósítása.

Elsősorban arra szeretném felhívni a figyelmet, hogy mi a projekt célja és kik tudják a legjobban hasznosítani, illetve mire lehet és mire nem ajánlatos használni.
A Condor különálló számítógépek erőforrásainak vezérlésére szolgál, elsősorban nagy számítás- és kis I/O-igényű feladatok feldolgozására. Támogatja a feladatok sorba állítását (queue), ütemezését, fontosság- (priority) megadását, az erőforrás-figyelést és -vezérlést. Egyaránt lehet alkalmazni kinevezett telepek (dedicated clusters), de más célra is használt számítógépek erőforrásainak az összekapcsolására.
Nem alkalmas rövid, gyors lefutású, illetve nagy I/O-igényű programok futtatására. Olyan kinevezett telep esetén, amely közös hálózati fájlrendszert használ, a második feltétel nem olyan szoros megkötés – alkalmatlan például a webkiszolgáló feladatainak több gépre történő szétosztására.
Elsődleges célja, hogy nagy számításigényű feladatok futását irányítsa eltérő és gyakran változó erőforrásokon. Azt nem teszi lehetővé, hogy egy folyamatot valamely erőforráson azonnal elindítson – több perc is eltelhet a tényleges futtatásig –, de ha nagyszámú erőforrás áll rendelkezésre, amin számos olyan feladatot (job) kell lefuttatni, amelyek kellően párhuzamosíthatók, ennek a vezérlését viszonylag jól elvégzi. Nem is biztos, hogy megtalálja a leghatékonyabb sorrendet és erőforrás-kiosztást, de általában közelít hozzá. A szakirodalomban ezt összefoglalóan nagy feldolgozási képességű környezetnek (High-Throughput Computing) nevezik.
Az ötlet onnan származott, hogy az egyetemi tanszékeken viszonylag sok gép található, kihasználtságuk azonban csekély, ugyanakkor általában egész nap üzemelnek és hálózatba vannak kötve. Ezt az „elpocsékolt” számítási teljesítmény akarták hasznosítani. Például valaki le szeretne futtatni néhány feladatot (job), ami a saját számítógépén akár hónapokig is eltartana. Mindez a tanszék vagy az egyetem gépein pár nap alatt elvégezhető, ha Condorral össze vannak kapcsolva. Egy feladatot csak el kell indítani, és amennyiben a Condor talál hozzá megfelelő erőforrást, azonnal oda is költözteti, majd amikor a program lefutott, értesíti a tulajdonost.
Másik jelentős felhasználási területe a Grid – egy világméretű telep létrehozása, amelyben az erőforrások teljesen eltérőek és minduntalan változnak.
A Condor az utóbbi pár évben a Grid kapcsán egyre inkább előtérbe került. Bár a projekt még akkor indult (1988), amikor a mai „Grid” fogalom nem is alakult ki teljesen, mégis nagyon alkalmas eszköz a Grid megvalósítására, ugyanis mindent tud, ami a Gridhez szükséges: képes teljesen eltérő, állandóan változó környezetben erőforrásokat nyilvántartani és kiosztani. Egyetlen hiányossága a felügyelet (monitoring) területén tapasztalható, de ezt például a magyar fejlesztésű (MTA SZTAKI) P-GRADE kiválóan kiegészíti.

Járulékos előnyök

Lehetőség nyílik arra, hogy a Condor a feladatok futtatására csak a szabad gépidőt használja fel. Az is meg van oldva, hogy-ha valaki hosszú idő után tér vissza a saját számítógépéhez, amin más feladat (job) fut, akkor amint egyetlen billentyűt lenyom, a Condor önműködően más gépre költözteti az ott futó programot.
Lehetőség van Glide-in módszer alkalmazására, amelynek az az előnye, hogy olyan gépek is használhatók Condor-feladatok (Condor-job) futtatására, amelyekre nem telepítettek Condort.
A ClassAd módszer segítségével megadható, hogy milyen típusú erőforrások szükségesek a feladat futtatásához: milyen a kiépítés (architect), a gépek száma, a gép processzorainak a gyorsasága, a processzorok száma és a memória mérete.

Használat

A telepítésről a hivatalos weblapon egy csomó leírás található (➲ http://www.cs.wisc.edu/condor/). Lépésről lépésre végigvezet bennünket az egyes szakaszokon. Itt erre nem térek ki, csak néhány lényeges pontra hívom fel a figyelmet.
Az első fontos tényező: a tűzfalak beállításai. A Condor nagyszámú különböző kaput (port) használ a folyamatok költöztetésére és a kapcsolattartásra, ezért az összekapcsolt gépek között legalább az 1024 feletti kapuk használatát engedélyezni kell. Ez tanszékek esetében általában nem gond, mert egyetlen külső tűzfallal bírnak és a belső hálózaton szinte minden megengedett.
Nehézség támadhat azonban az egyetemek, az egymástól távol lévő szervezetek összekapcsolásakor. E gondokra is született néhány megoldás, ilyen például a Condornak a Globusszal történő kombinálása.
A Globus – a Condorhoz hasonlóan – folyamatok más gépen való futtatását teszi lehetővé, de csak alacsonyabb szintű szolgáltatásokat nyújt. Biztonságos, titkosított csatornákat tartalmaz a folyamatok költöztetésére, a költöztetés viszont nem önműködő. A Globus esetén valóban le kell írni, hogy melyik folyamatot hol szeretnénk futtatni és hogy a folyamatok hogyan tartsanak kapcsolatot egymással. Ez nagymértékben megnehezítheti a programozók, illetve a kutatók munkáját, ezért kifejlesztették a Condor-G változatot, amely a Globust a folyamatirányítás képességével bővíti ki.
A másik megoldás, hogy a különálló Condor-telepeket (Condor-Pool) VPN-nel kapcsoljuk össze. Ilyen megoldást alkalmaz a már korábban említett KlaszterGrid-projekt is.
Használata egyszerű: össze kell állítani a feladatot, le kell írni, hogy milyen erőforrások szükségesek, és a Condor nyújtotta programkönyvtárral újra kell építeni a programot. Ez a folyamat függhet attól, hogy a feladatot milyen Condor-univerzumban futtatjuk. (Általános univerzum esetén a fenti lépéseket kell végrehajtani.)
A Condor nagyszerű választás lehet kutatóintézetek, egyetemek számára, ahol nincs pénz szuperszámítógépek beszerzésére, így minimális ráfordítással is viszonylag nagy számítási teljesítményt lehet összefogni.

Kolcza Péter (kpeter@sysconfig.hu)

Dinamikus rendszermagmodul-támogató keretrendszer

Kezeld a modulokat egy egyszerű háttérrendszerrel a rendszermagtól elkülönülten, és hozz ki többet a csomagkezelődből!

A forráskód önmagában is csodás dolog; a rendszermagfába foglalt modulforrás még csodásabb. Ennek a forrásnak a támogatása és támogatottsága az egyik legfontosabb tényező. Napjainkban a Linux robbanásszerűen terjed a vállalatoknál, és az a lehetőség, hogy egyetlen telefonhívással segítséget tudjunk kérni, nélkülözhetetlen. A Linux fejlesztését meghatározó kereteket soha nem látott mértékben formálják át a vállalatok elvárásai. A közösség részéről sokszor kétkedő és aggódó hangok hallatszanak emiatt, ám ha mindenki jól végzi a feladatát, a fejlesztések előnyeit bárki élvezheti.
A dinamikus rendszermagmodul-támogatási (DKMS) keretrendszer nagyszerű példa minderre. A DKMS segítségével – a rendszert a Dell Computer Corporation számára készítették, a céget a javításoknak az ügyfelekhez ellenőrzött módon való eljuttatásában segíti – az illesztőprogramok fejlesztése, a tesztek és az ellenőrzések elvégzése is felgyorsítható, és ez a lehetőség a teljes közösség számára adott.
A DKMS keretrendszer alapjában véve egy kettős, a rendszermagfától különálló fa, amely a modulforrásokat és a lefordított, futtatható modulokat tartalmazza. A kettősség lehetővé teszi a moduloknak a rendszermagtól való szétválasztását, amely hasznos lehetőség a linuxos megoldásokat szállítók és telepítők számára. A rendszer nagyszerűsége abban rejlik, hogy – szabályozott és támogatható módon – lehetővé teszi az illesztőprogramok beillesztését a meglévő rendszermagokba. Így a szolgáltatók és ügyfeleik az adott rendszermag változatától függetlenül átadhatják–átvehetik a javításokat. Ha egy illesztőprogramhoz megjelenik egy javítás, a DKMS áthidaló megoldásként lehetővé teszi annak terjesztését, amíg bele nem kerül a rendszermagba.
Ha tovább vizsgálódunk, a DKMS a felhasználók számára további lehetőségeket is kínál. Forrásból fordítani, telepíteni, forrás-RPM-ekkel küzdeni: ez az, amire a legtöbben a legkevésbé sem vágynak. A valóságban számos olyan Linux-felhasználó van, aki kevés tapasztalattal rendelkezik, és egyszerűbb megoldásokat szeretne. A DKMS segít megoldani ezeket kérdéseket, mivel egyetlen futtatható állománnyal oldja meg a modulok fordítását, telepítését és eltávolítását. A modulok új rendszermagokon való üzembe helyezését is megkönnyíti, hiszen a telepítendő modulok beállításai kizárólag egy korábban már működött rendszermagon is alapulhatnak. Termelési környezetben ez nagy könnyebbséget jelent a rendszergazdák számára, hiszen ezentúl nem kell választaniuk az előre megadott programkészletek, valamint az újabb rendszermagok biztonsági szolgáltatásai között.
A DKMS a fejlesztők és a tapasztalt Linux-rajongók számára is tartogat meglepetéseket. A modulok és a rendszermagkettőzés által való (nem teljes) szétválasztásának már említett lehetősége nagyszerű környezet létrehozását teszi lehetővé illesztőprogramok fejlesztéséhez. Nem kell különféle rendszermagokhoz újra és újra javításokat készíteni, ehelyett a javításokat széles körben lehet terjeszteni és éles helyzetben ki lehet próbálni. A próbák felgyorsításával a teljes fejlesztési folyamat is lerövidül. A különféle rendszermagkiadások léte a továbbiakban nem jelenthet akadályt a modulkódok széles körű terjesztése előtt, így alaposabban kipróbált kód jöhet létre, amit hamarabb be lehet építeni a rendszermagba – máris mindenki jól járt, fejlesztők és felhasználók egyaránt.
A DKMS használatával – a rendszermagfüggő programok előállításának egyszerűsödésével – könnyebbé válik a fejlesztők élete. Korábban például a Dell elsősorban rendszermagtól függő, előre lefordított RPM-ek formájában terjesztette a moduljait. Ahogy újabb és újabb rendszermagok jelentek meg, gyakran egyhangú, végtelennek tűnő munkába merültünk, amikor ezekhez újra kellett fordítani a futtatható fájlokat – ezt valószínűleg egyetlen fejlesztő sem kívánja magának. A Dell mégis ragaszkodott ehhez az eljáráshoz, mert ez kívánta meg a legkevesebb munkát és hozzáértést azon felhasználóktól, akiknek telepíteniük kellett a modulokat. A DKMS egyszerre könnyen használható és segít csökkenteni a fejlesztői oldalra háruló munka mennyiségét. A DKMS használatakor mindössze a modul forráskódjának kell a felhasználó gépén rendelkezésre állnia. Bármilyen rendszermagot futtat is a felhasználó, a DKMS elvégzi a modul fordítását és telepítését, mindenkit megkímélve a magváltozatok követése kapcsán előforduló bosszúságoktól.

A DKMS használata

Ha sikerült felkeltenem az érdeklődést a DKMS iránt – márpedig ez volt a szándékom –, próbáljunk meg egy kicsit elmélyülni a használatában. Ha rá szeretnénk venni a DKMS-t valamilyen modul kezelésére, akkor a modul forrásának rendelkezésre kell állnia a helyi gép */usr/src/(modul)-(modul-változat)/* könyvtárában. Szükség van egy ***dkms.conf*** állományra, ebben megfelelően formázott utasításokkal kell előírni a DKMS-nek többek közt azt, hogy hová telepítse a modult, illetve hogyan fordítsa le. A ***dkms.conf*** állomány formátumáról később még szólni fogok. Ha ez a két követelmény teljesül, és a DKMS telepítve van a rendszerre, meg is kezdhetjük a használatát, ha a kívánt modul/modulváltozat ággal kibővítjük a DKMS-fát. Lássunk egy példaparancsot a hozzáadás elvégzésére:

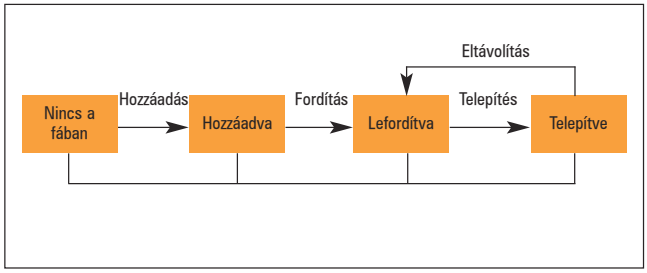
```
dkms add -m qla2x00 -v v6.04.00
```

A parancs lefutásakor a ***qla2x00/v6.04.00*** könyvtárral bővül a meglévő */var/dkms* fa. A parancs kiadása után létrejön a */var/dkms/qla2x00/v6.04.00/* könyvtár, egy közvetett hivatkozás a */var/dkms/qla2x00/v6.04.00/source* könyvtárról a */usr/src/qla2x00-v6.04.00/* könyvtárra, valamint az eredeti ***dkms.conf*** állományról készül egy másolat */var/dkms/qla2x00/v6.04.00/dkms.conf* név alatt.
A hozzáadás után készen állunk a modul lefordítására. A `dkms build` parancsának működéséhez a megfelelő rendszermagforrásnak a */lib/module/kernel-version/build* közvetett hivatkozással elérhetőnek kell lennie. A modul lefordításához

használt `make` parancsot a ***dkms.conf*** beállítófájl határozza meg. Az előbbi példát folytatva:

```
dkms build -m qla2x00 -v v6.04.00
➥-k 2.4.20-8smp
```

A paranccsal lefordítjuk a modult, de nem telepítjük. A `build` parancs egy `kernel-version` átadott értéket is vár, ha ezt elhagyjuk, akkor az éppen futó rendszermag változatszámát veszi figyelembe. Mint sejthető, az éppen futótól eltérő rendszermaghoz is könnyedén készíthetünk modulokat. Ezt a lehetőséget egy rendszermag-előkészítő programrész biztosítja, amely a modulok létrehozása előtt jut szerephez. A minden részletre kiterjedő előkészítés során lefut egy `make mrproper` parancs, a megfelelő rendszermag ***.config*** fájlja a rendszermag forrásának könyvtárába másolódik, lefut a `make oldconfig`, és végül a `make dep`. Ezek a lépések garantálják, hogy a létrejövő modul a megfelelő rendszermagszimbólumokkal épüljön fel. Az alapértelmezett beállítás szerint a DKMS a ***/lib/modules/kernel-version/build/configs/*** könyvtárban keresi a rendszermag ***.config*** fájlját, és a beállításfájlok kezelésekor a Red Hat névadási szokásait követi. Ha a rendszermag ***.config*** fájlja nem ebben a könyvtárban található, akkor a helyét a `build` parancs kiadásakor a `--config` kapcsolóval kell megadni a DKMS-nek.



Modulok állapota DKMS alatt

A fordítás végrehajtása során – eddigi példánknál maradva – létrejön a ***/var/dkms/qla2x00/v6.04.00/2.4.20-8smp/*** könyvtár, illetve ezen belül egy ***log*** és egy ***module*** alkönyvtár. A ***log*** könyvtárba a modul fordításának naplója, a ***module*** könyvtárba pedig a lefordított ***.o*** állományok kerülnek.
A fordítás befejezése után a modult azonnal telepíteni lehet ahhoz a rendszermaghoz, amelyhez készült. A telepítés során a rendszer a ***/lib/modules/*** fa megfelelő helyére másolja a modul lefordított, futtatható állományát, ahogy azt a ***dkms.conf*** fájlban előírtuk. Ha ezen a helyen már létezik ilyen nevű modul, azt a DKMS menti a saját fájába mint eredeti modult, így később, az újabb modul eltávolításakor a régebbi visszamásolható a helyére. Példa a telepítés elvégzésére:

```
dkms install -m qla2x00 -v v6.04.00 -k
2.4.20-8smp
```

A parancs futása során létrejön egy ***/var/dkms/qla2x00/v6.04.00/kernel-2.4.20-8smp*** – ***/var/dkms/qla2x00/v6.04.00/2.4.20-8smp*** közvetett hivatkozás. A DKMS ilyen hivatkozások segítségével tartja nyilván, hogy mely rendszermaghoz melyik illesztőprogram-változat van telepítve. Mint említettem, ha azonos névvel már létezik modul, a DKMS ennek másolatát a saját fájába, a ***/var/dkms/module-name/original_module/*** könyvtárba menti.

Esetünkben a modul a ***/var/dkms/qla2x00/original_module/2.4.20-8smp/*** könyvtárba kerül.
A DKMS-féle életciklus a modulok eltávolításával, a fából való kivételével ér véget. Az eltávolítás során a rendszer törli a korábban telepített modult, és – ha van ilyen – az eredetit helyezi vissza. Ha a DKMS-fán belül egy modulnak több változata is megtalálható, akkor az eltávolítás során a DKMS nem tesz kísérletet arra, hogy kitalálja, melyik korábbi változatot szeretnénk újra használatba venni. Ehelyett, ha rendelkezünk valós `original_module` mentéssel az eredeti DKMS-telepítésből, akkor ez kerül vissza a rendszermagba. A modul minden egyéb változata ugyanúgy marad. Példa az eltávolításra:

```
dkms uninstall -m qla2x00 -v v6.04.00
➥-k 2.4.20-8smp
```

Ha a rendszermag változatát meghatározó átadott értéket elhagyjuk, a program az éppen futó rendszermagot veszi figyelembe. Hasonló célokra használható a `remove` parancs is. A `remove` és az `uninstall` hasonló egymáshoz: a `remove` is végrehajtja azokat a lépéseket, amelyeket az `uninstall` elvégez. Ha az eltávolított modulváltozat az adott változat utolsó példánya a rendszeren lévő összes rendszermaghoz, akkor a `remove` parancs használatakor az eltávolítási folyamat elvégzése után a rendszer az adott modul összes nyomát eltünteti a DKMS-fából. Másként fogalmazva: amikor az `uninstall` parancs lefut, a modulok „lefordított” állapotban maradnak. Ezzel szemben a `remove` parancs használatakor az `add` parancsig kell visszalépnünk, ha újra használni akarjuk az adott modult a DKMS-sel. Lássunk két példát a `remove` használatára:

```
dkms remove -m qla2x00 -v v6.04.00
➥-k 2.4.20-8smp
dkms remove -m qla2x00 -v v6.04.00 --all
```

Az első `remove` parancs a modul eltávolítását végzi el.
Ha ez a modulváltozat semmilyen más rendszermaghoz nincs telepítve, akkor a program nyomtalanul eltünteti a DKMS-fából. Ha például a `qla2x00/v6.04.00` a `2.4.20-8bigmem` rendszermaghoz is telepítve van, akkor az első `remove` parancs érintetlenül hagyja a DKMS-fában. Nem így a példa második parancsa. Ez a `qla2x00/v6.04.00` modul összes változatát minden rendszermagból eltávolítja, majd a `qla2x00/v6.04.00` összes hivatkozását teljesen eltünteti a DKMS-fából. A `remove` paranccsal tehát megtisztítható a DKMS-fa.

További DKMS-parancsok

A DKMS jól használható állapotlekérdező paranccsal rendelkezik, amely a fában éppen megtalálható tartalomról ad tájékoztatást. Ha kapcsolók nélkül használjuk, akkor az összes elérhető adatot visszaadja. Értelemszerűen a kapott adatok jellege a lekérdező parancs futtatásakor megadott kapcsolóktól függ. Minden bejegyzéshez vagy hozzáadott (`added`), vagy lefordított (`built`), vagy telepített (`installed`) állapot tartozhat. Ha az eredeti modult is mentette a program, akkor erről is kapunk visszajelzést. Néhány példa az állapot lekérdezésére:

```
dkms status
dkms status -m qla2x00
dkms status -m qla2x00 -v v6.04.00
```

```
dkms status -k 2.4.20-8smp
dkms status -m qla2x00 -v v6.04.00
➥-k 2.4.20-8smp
```

A DKMS egy másik fontos szolgáltatása a `match` parancs által érhető el. A `match` parancs megvizsgálja a DKMS által adott rendszermaghoz telepített modulok beállításait, és egy másik rendszermagra alkalmazza őket. A `match` lefutása után ugyanazok a modulváltozatok, amelyeket egy rendszermaggal már használunk, egy másik rendszermaghoz is telepíthetők. Erre elsősorban akkor lehet szükség, ha újabb rendszermagra frissítünk, ám meg szeretnénk tartani meglévő DKMS moduljainkat. Például:

```
dkms match --templatekernel 2.4.20-8smp
➥-k 2.4.20-9smp
```

A `--templatekernel` az a rendszermag, amelynek alapján a beállításokat meghatározzuk. A `-k` rendszermag az, amelyre a beállításokat alkalmazzuk.
A DKMS rendszerfelügyeleti célokra az `mktarball` és az `ldtarball` parancsot is támogatja. Ezekkel a parancsokkal a felhasználó ***.tar***-állományokat hozhat létre és nyithat meg, valamint illeszthet be a DKMS-fába, például arra az esetre, ha a DKMS segítségével több hasonló rendszerbe kell valamit telepítenie. Így a rendszergazda egyetlen számítógépen készítheti el az összes modult. Nincs szükség arra, hogy ugyanazt a modult minden gépen külön lefordítsuk, ehelyett a már lefordított futtatható állományt kell csak beilleszteni az adott géphez tartozó DKMS-fába. Az `mktarball` paranccsal adott modulváltozat forrásából lehet ***.tar***-állományt készíteni. Így menthető minden olyan magváltozat DKMS-fája, amely a megadott modulváltozat alapján fordított modullal rendelkezik. Példa:

```
dkms mktarball -m qla2x00 -v v6.04.00
➥-k 2.4.20-8smp,2.4.20-8
```

A `-k` átadott értéktől függően az `mktarball` csak a megadott rendszermagok futtatható állományait archiválja. Ha ilyen értéket nem adunk át, akkor az adott modulváltozat összes lefordított futtatható állományát összecsomagolja.
Az `ldtarball` használatakor a DKMS egyszerűen kibontja az `mktarball` segítségével létrehozott archív állományt, és a benne találtakat beilleszti a futtató rendszer DKMS-fájába. Ennél az eljárásnál minden modul lefordított állapotban marad, majd a `dkms install` paranccsal telepíteni lehet a futtatható modulokat a ***/lib/modules*** fába. Rendes körülmények között az `ldtarball` nem írja felül az adott rendszer DKMS-fájában már meglévő állományokat. Ha mégis erre van szükség, a `--force` kapcsolóval írhatjuk elő a fa tartalmának az archív állományéval történő felülírását. Példa az `ldtarball` használatára:

```
dkms ldtarball --config
➥qla2x00-v6.04.00-kernel2.4.20-8smp.tar.gz
```

A DKMS parancsai közül utolsóként az `mkdriverdisk`-et említeném meg. Mint neve is utal rá, az `mkdriverdisk` kikeresi a megfelelő forrásokat a DKMS-fából, majd egy illesztőprogramot tartalmazó hajlékonylemez-lenyomatfájlt készít, amit frissített illesztőprogramok terjesztésére lehet használni. Példa az `mkdriverdisk` használatára:

Példa a .spec fájlra

```
%define module qla2x00

Summary: Qlogic HBA module
Name: %module_dkms
Version: v6.04.00
Release: 1
Vendor: Qlogic Corporation
Copyright: GPL
Packager: Gary Lerhaupt
➥<gary_lerhaupt@dell.com>
Group: System Environment/Base
BuildArch: noarch
Requires: dkms gcc bash sed
Source0: qla2x00src-%version.tgz
Source1: dkms.conf
BuildRoot: %{_tmppath}/%{name}-%{version}-
➥%{release}-root/


%description
This package contains Qlogic´s qla2x00
➥HBA module meant
➥for the DKMS framework.


%prep
rm -rf qla2x00src-%version
mkdir qla2x00src-%version
cd qla2x00src-%version
tar xvzf
➥$RPM_SOURCE_DIR/qla2x00src-%version.tgz


%install
if [ "$RPM_BUILD_ROOT" != "/" ]; then
      rm -rf $RPM_BUILD_ROOT
fi
```

A lista folytatását lásd a következő oldalon.

```
dkms mkdriverdisk -d redhat -m qla2x00
➥-v v6.04.00 -k 2.4.20-8BOOT
```

Jelenleg egyedül a Red Hat illesztőprogram-lemezformátumának a támogatása érhető el, ám a közösség segítségével a többi terjesztés követelményei és formátumai is teljesíthetők, támogathatók. A DKMS által a Red Hat illesztőprogram-lemezek készítéséhez igényelt külön fájlokról és formátumukról a ➲ http://people.redhat.com/dledford weboldalon lehet bővebb tájékoztatást találni. Ezeket a fájlokat a modulforrások könyvtárában kell elhelyezni.

A dkms.conf beállítófájl formátuma

A DKMS-csomagokkal foglalkozók számára a ***dkms.conf*** beállítófájl az egyetlen olyan külső forrás, amelyre a forrás ***.tar***-állományok DKMS-re való késszítéséhez szükség van. A beállítófájl valójában héjváltozók listáját tartalmazza, a DKMS ezek alapján végzi munkáját a csomaggal. Példaként

A lista folytatása az előző oldalról.

```
mkdir -p
➥$RPM_BUILD_ROOT/usr/src/%module-%version/
install -m 644 $RPM_SOURCE_DIR/dkms.conf
➥$RPM_BUILD_ROOT/usr/src/%module-%version
install -m 644 qla2x00src-%version/*
➥$RPM_BUILD_ROOT/usr/src/%module-%version


%clean
if [ "$RPM_BUILD_ROOT" != "/" ]; then
        rm -rf $RPM_BUILD_ROOT
fi


%files

%defattr(0644,root,root)

%attr(0755,root,root)
➥/usr/src/%module-%version/

%pre

%post
/sbin/dkms add -m %module -v %version
/sbin/dkms build -m %module -v %version
/sbin/dkms install -m %module -v %version
exit 0


%preun
/sbin/dkms remove -m %module
➥-v %version --all
exit 0
```

egy részlet a `qla2x00/v6.04.00` ***dkms.conf*** fájlból:

```
MAKE="make all
➥INCLUDEDIR=/lib/modules/$kernelver/
➥build/include"
MAKE_smp="make SMP=1 all
➥INCLUDEDIR=/lib/modules/$kernelver/
➥build/include"
LOCATION="/kernel/drivers/addon/qla2200"
REMAKE_INITRD="yes"
MODULE_NAME="qla2200.o:qla2200_6x.o
➥qla2300.o:qla2300_6x.o"
CLEAN="make clean"
MODULES_CONF_ALIAS_TYPE=
➥"scsi_hostadapter"MODULES_CONF0=
➥"options scsi_mod
➥scsi_allow_ghost_devices=1"
```

Mint látható, a héjváltozókat csupa nagybetűvel kell írni. Ez alól az egyik kivétel a `MAKE_` utasítás. A DKMS az általános `MAKE=` parancsot használja a modulok lefordításához. Ha azonban létezik egy `MAKE_kernel-regexp-text` parancs, és a `MAKE_` utáni rész (karakterlánc-részletként) egyezést mutat azzal a rendszermaggal, amelyhez a modul készül, akkor ezt a másodlagos `make` parancsot használja. A fenti példában kiderül, hogy a DKMS a `MAKE_smp` utasítás használatára is képes, ha `smp` rendszermaghoz kell modult fordítani. Hasonló `PATCH_` parancsok is léteznek. Ha az alulhúzás utáni szöveg egyezést mutat azzal a rendszermaggal, amelyhez a modul készül, akkor elsőként ez a folt kerül be a modul forrásába. Így a fejlesztőknek csak egyetlen forrás ***.tar***-állományt kell terjeszteniük, amelyhez egy ***dkms.conf*** fájl és akár több folt tartozik. Természetesen szükség szerint különféle foltokat is be lehet építeni a forrásba, így biztosítva, hogy minden modul minden rendszermaggal megfelelően együttműködjön. Érdemes megjegyezni, hogy a ***dkms.conf*** `$kernelver` változót is tartalmazhat, amelyet a fordítás alatt a program annak a rendszermagnak a változatszámára cserél le, amihez a modul készül. Ez különösen fontos, hiszen így mindig megfelelő ***include*** könyvtárakra lehet hivatkozni, amikor az éppen futótól eltérő rendszermaghoz fordítunk modult.

A DKMS és az RPM együttes használata

A DKMS és az RPM remekül együtt tud működni. A hibátlan működéshez mindössze olyan RPM-et kell készíteni, amely elvégzi a forrás telepítését. A forrásokat általában forrás RPM-ek segítségével telepítjük, ám ezek nem feltétlenül működnek együtt a DKMS-sel, így a forrás telepítésén túl másra nem nagyon alkalmasak. Jobban járunk, ha a forrás ***.tar***-fájl egy RPM-be kerül, így el lehet helyezni a megfelelő ***/usr/src/(modul)-(modulváltozat)/*** könyvtárba, és a megfelelő DMKS-parancsokat is meg lehet hívni. A `%post` és a `%preun` alapvetően DKMS-parancs. *Listánkon* jó példa látható egy ***.spec*** fájlra.

A jövő

Mivel a DKMS nagyon új keretrendszer, számos dologgal kell még bővíteni, másokat ki kell venni belőle, esetleg újra kell írni – ahogy a közösség határoz. A DKMS-tervezet előrehaladásával kapcsolatosan a ➲ http://www.freshmeat.net/projects/dkms oldalon találhatók naprakész tudnivalók. Kérdéseit, észrevételeit bárki felteheti vagy elküldheti, ha csatlakozik a DKMS-devel levelezési listához, amely a ➲ lists.us.dell.com/mailman/listinfo címről érhető el.
A Linux Journal 2002. decemberi számában ***Linus Torvaldstól*** a következő idézet jelent meg:
„Úgy látom, minden kereskedelmi szereplőnek megvan a maga ütemterve, és bár ezek sokszor ütköznek egymással, végül egy olyan állapot felé terelik az egész rendszert, amely mindannyiunk számára biztosítja a kívánt szolgáltatásokat.”
Linuxos termékek kereskedelmi értékesítőjeként a Dellnek alapvető érdeke, hogy megfelelő megoldást találjon erre a modul-, illetve rendszermagkérdésre, amivel nemcsak saját ügyfeleinek nyújthat magasabb szintű szolgáltatást, de a közösséget is segíti. Ezeket a szempontokat a DKMS tervezése során sem veszítették szem elől.

Linux Journal 2003. szeptember, 113. szám

Gary Lerhaupt (gary_lerhaupt@dell.com)
Programmérnök a Dell Linux Development csoportjánál. Részt vett a Dell Red Hat Linux alapú Oracle9i Real Application Clusters (RAC) tervezetében is. Gary az Ohio Egyetemen szerezte meg informatikusi diplomáját.

Régi ismerősök látogatása

A 2.5-ös változat szerint tárgyalt rendszermag API-k közül néhányon kisebb módosításokat hajtottak végre, mielőtt bekerültek volna a 2.6-os rendszermagba. Ezekről a módosított változatokról szeretnék szólni néhány szót.

Ez témában első cikkemet több mint egy évvel ezelőtt írtam, és mivel a Linux fejlesztése gyors ütemben folyik, számos korábban leírt dolog mára elavultnak számít. Ez alkalommal a korábban ismertetett rendszermag API-k módosításait szeretném áttekinteni.

A tty változásai

A tty réteg az egyik legbiztosabb működésű, sokat bizonyított rendszermag-API. A hivatkozások megfelelő számlálásának és a zárolási lehetőségnek a hiánya, valamint a tty-eszközök kiosztásának furcsa módja mind a réteg elöregedésére vezethető vissza. Szerencsére ***Al Viro*** nemrég kipucolta a régről maradt ócskaságokat a 2.5-ös rendszermagsorozatban látott tty-rétegből. Mindeközben természetesen számos dolog megváltozott, és az új tty-illesztőprogramok írásakor erre tekintettel kell lenni.

A Linuxvilág 2002. szeptemberi és novemberi számában (illetve a Linux Journal 2002. augusztusi és októberi számában; a cikkek elérhetők a ➲ www.linuxjournal.com/article/5896 és a ➲ www.linuxjournal.com/article/6226 címen) a tty-rétegről, illetve a `struct tty_driver` adatszerkezet szolgáltatások visszahívóival történő feltöltéséről írtam. Azóta `struct tty_operations` névvel létrejött egy új adatszerkezet, ez tartalmazza az összes szolgáltatás visszahívóját. A `struct tty_driver` továbbra is tartalmazza a régi függvénymutatókat, így ezek átmásolására készült egy új függvény, a `tty_set_operations`. A kettősség hamarosan remélhetőleg megszűnik. A `struct tty_driver` adatszerkezetből jó néhány változó kikerült: a `table`, a `termios`, a `termios_locked` és a `refcount` mezőnek búcsút inthetünk. A tty-réteg most már az összes zárolási és hivatkozásszámlálási igényt kielégíti, megszabadítva az egyes tty-illesztőprogramokat e feladatok kezelésének terhétől. A `magic` és a `num` változót most már nem kell közvetlenül a tty-illesztőprogramnak beállítania. A változóértékek beállítását új függvény, az `alloc_tty_driver` végzi, amit minden tty-illesztőprogramnak meg kell hívnia, ugyanis ez végzi az illesztőprogramok számára a helyfoglalást. Az adott illesztőprogram által támogatott különféle tty-eszközök számát átadott értékként szükséges tudatni ezzel a függvénnyel. A tiny tty illesztőprogram, amelyet korábban vettünk elő példaként, az alábbiak szerint hozza létre a `struct tty_driver` adatszerkezetet:

```
/* a tty illesztőprogram helyfoglalása */
tiny_tty_driver = alloc_tty_driver(TINY_TTY_MINORS);
```

Korábban a tty-illesztőprogram nevének megválasztása is gondokat okozott, ugyanis a rendszermag `devfs` függvénye túlterhelte a `name` mezőt. ***Christoph Hellwig*** munkájának hála, aki a `struct tty_driver` adatszerkezetben létrehozott egy új változót, a `devfs_name`-t. Most már a `name` mező egyszerű, rövid nevet is kaphat, amelyet a tty ***proc*** fájlokban fogunk viszontlátni. A `devfs_name` értékét arra a névre kell állítani, amelyet a `devfs` használ az illesztőprogram eszközcsomópontjának létrehozásakor. A 2.5-ös rendszermagsorozatban a `MOD_INC_USE_COUNT` és a `MOD_DEC_USE_COUNT` makrókat túlságosan csúnyának minősítették, és használatuk a rendszermagban nagyrészt meg is szűnt. Ahhoz, hogy ez megtörténhessen, a modulhivatkozások számlálása az eredeti hívásokhoz képest eggyel magasabb szintre került. Most már lehetséges, hogy a rendszermag az előtt növelje egy-egy modul hivatkozási számlálóját, hogy beugrana magába a modulba. Hasonlóan, amikor a rendszermag végzett a modul használatával, a számláló értékét önműködően csökkenteni tudja.

A modulokra vonatkozó módosítás a tty-rétegben történt, így egyetlen tty-illesztőprogramnak sem kell tartalmaznia a `MOD_*`

2. lista Példa egy „tiocmget” függvényre

```
int
tiny_tiocmget(struct tty_struct *tty,
              struct file *file)
{
    struct tiny_private *tp = tty->private;
    unsigned int msr = tp->msr;
    unsigned int mcr = tp->mcr;
    unsigned int result = 0;

    result = ((mcr & MCR_DTR)     ? TIOCM_DTR: 0)
                                    /* DTR beállítva */
           | ((mcr & MCR_RTS) ? TIOCM_RTS: 0)
                                    /* RTS beállítva */
           | ((msr & MSR_CTS) ? TIOCM_CTS: 0)
                                    /* CTS beállítva */
           | ((msr & MSR_CD)  ? TIOCM_CAR: 0)
                                /* vivőérzékelés beállítva */
           | ((msr & MSR_RI)  ? TIOCM_RI:  0)
                                /* csengetésjelző beállítva */
           | ((msr & MSR_DSR) ? TIOCM_DSR: 0);
                                    /* DSR beállítva */

    return result;
```

3. lista Példa a tiocmset függvényre

```
int
tiny_tiocmset(struct tty_struct *tty,
              struct file *file,
              unsigned int set, unsigned
              ➥int clear)
{
    struct tiny_private *tp = tty->private;

    if (set & TIOCM_RTS)
        mcr |= MCR_RTS;
    if (set & TIOCM_DTR)
        mcr |= MCR_RTS;
    if (set & TIOCM_LOOP)
        mcr |= MCR_LOOPBACK;

    if (clear & TIOCM_RTS)
        mcr &= ~MCR_RTS;
    if (clear & TIOCM_DTR)
        mcr &= ~MCR_RTS;
    if (clear & TIOCM_LOOP)
        mcr &= ~MCR_LOOPBACK;

/* az új MCR érték beállítása az eszközön */
    tp->mcr = mcr;
    return 0;
}
```

makrókat. Egy `owner` változó ugyanakkor bekerült a `struct tty_driver` adatszerkezetbe, feladata a tty-illesztőprogramot birtokló modul megadása. Az alábbi sor ennek a változónak a helyes megadását szemlélteti:

```
tiny_tty_driver->owner = THIS_MODULE;
```

A tty-mag ebből tudja meg, hogy melyik modul tartozik ehhez a tty-illesztőprogramhoz.
Az 1. listán látható (55. CD Magazin/Api könyvtára), hogy a változások következtében miként módosul a tty-illesztőprogramok helyes kezdeti értékadásának és bejegyzésének módja. A tty-kihívó (callout) eszközöket teljesen eltávolították a rendszermagból. Azok a tty-illesztőprogramok, amelyek a 2.5-ös rendszermagfában megvoltak, és kihívásokat alkalmaztak, a módosításnak megfelelően átalakultak.

Kevesebb ioctl

A tty-adatszerkezet változásaival párhuzamosan néhány tty `ioctl` is kikerült a rendszerből, ezek a következők: `TIOCMGET`, `TIOCMBIS`, `TIOCMBIC` és `TIOCMSET`. Helyüket két új függvényvisszahívó vette át: a `tiocmget` és a `tiocmset`, ezeket a `struct tty_operations` adatszerkezethez adták hozzá. A függvények megadása a következő:

```
int (*tiocmget)(struct tty_struct *tty,
                struct file *file);
int (*tiocmset)(struct tty_struct *tty,
                struct file *file,
                unsigned int set,
                unsigned int clear);
```

A `tiocmget` függvény meghívására akkor kerül sor, amikor a tty-mag vagy a felhasználó tudni szeretné az adott tty-kapu éppen érvényes vonali beállításait. Működése majdnem pontosan olyan, mint a régi `TIOCMGET` `ioctl` hívásé. A vonalállapotot különböző `MSR_*` értékek adják meg, ezeket – a korábbi `ioctl` által az illesztőprogramtól elvárt felhasználói területre való másolás helyett – közvetlenül a függvényhívás adja vissza (lásd a *2. listát*).
A `tiocmset` függvényre akkor van szükség, amikor a tty-mag vagy a felhasználó be szeretné állítani, illetve törölni szeretné a különféle vonali beállítások valamelyikét. Ez a függvény egymaga helyettesíti a `TIOCMBIS`, a `TIOCMBIC` és a `TIOCMSET` `ioctl` hívásokat. A függvény `set` és `clear` változóival szabható meg, hogy melyik vonali beállítást szeretnénk megadni, illetve melyiket akarjuk törölni. Ugyanazt a vonali beállítást nem lehet egyszerre beállítani és törölni is, így a változók feldolgozásának sorrendje érdektelen (lásd a *3. listát*).
A tty-mag módosításai kismértékben az usbserial-magot is érintették. A `tiocmget` és a `tiocmset` függvény újdonságként jelent meg a `struct usb_serial_device_type` adatszerkezetben. Az ezekre a függvényekre irányuló tty-hívásokat a rendszer továbbadja az alacsonyabb szinten futó `usbserial` illesztőprogramoknak – feltéve, hogy az adott `usbserial` illesztőprogram támogatja ezeket a visszahívásokat.

Köszönetnyilvánítás

Szeretném kifejezni hálámat ***Al Viro***, ***Christoph Hellwig*** és ***Russell King*** uraknak, amiért végre nekiláttak a tty-réteg megtisztításának. Odaadásuk révén a tty-réteg méltó része lesz a rendszermagnak. Módosításaik fontos szerepet játszanak a tty-illesztőprogramfelület egyszerűbbé tételében, és az ő révükön az illesztőprogramok készítői a jövőben a rendszermagot érintő párbeszédek kezelése helyett inkább az eszközök jellegzetességeire összpontosíthatnak.

A cikkhez kapcsolódó listák megtalálhatóak az 55. CD Magazin/Api könyvtárában.

Linux Journal 2003. október, 114. szám

Greg Kroah-Hartman (greg@kroah.com)
Jelenleg a Linux-rendszermag különféle illesztőprogram-alrendszereiért felelős. Az IBM-nél dolgozik és a Linux-rendszermaggal kapcsolatos kérdésekkel foglalkozik.

Építsünk IPv6 DNS-kiszolgálót Linuxon!

Útmutató IPv6 DNS-névfeloldást nyújtó kiszolgáló készítéséhez, valamint néhány hasznos IPv6-alkalmazási példa.

Az IPv6 az Internet Engineering Task Force (IETF) által tervezett következő nemzedékbeli protokoll, amelynek célja az IPv4, azaz a jelenlegi Internet Protokoll leváltása. Az IPv4 figyelemreméltóan rugalmasnak bizonyult, ugyanakkor a tervezésekor nem vettek figyelembe néhány, manapság egyre fontosabbá váló tényezőt, például a nagy címhelyet, a mobilitást, a biztonságot, az önműködő beállítási lehetőségeket és a szolgáltatási minőséget. E nehézségek feloldására az IETF egy új protokollkészletet, valamint az IPv6 nevű szabványt dolgozta ki, amely több, az IPv4 frissítését célzó megoldást és javasolt elképzelést magában foglal. Ennek eredményképpen az IPv6 az IPv4 több hibáját is megoldja, és számos fejlesztést és új képességet vezet be, amelyek a jövő internetellátásáról gondoskodnak majd.

Az IPv6 várhatóan fokozatosan váltja majd fel az IPv4 rendszert, miközben az átmeneti időszak alatt egymás mellett működik mind a kettő. A kiszolgálók kettős rendszerűek lesznek: egyszerre támogatják az IPv4 és IPv6 protokollt.

Ebben a cikkben az IPv6 névfeloldási rendszerével ismerkedhetünk meg közelebbről, remélve, hogy sikerül olyan műszaki segédletet nyújtanunk, amelynek alapján bárki elkészítheti a saját, IPv6 névfeloldást végző IPv6 alapú linuxos DNS-kiszolgálóját a BIND 9.x legfrissebb változatával.

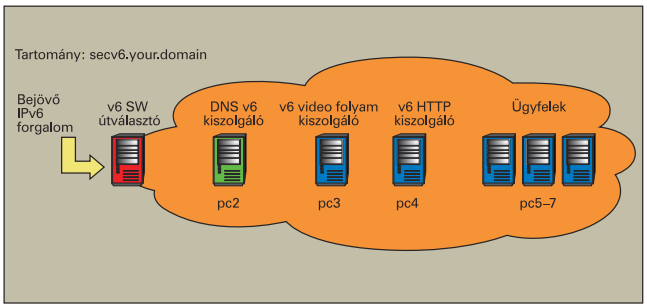
Általános hálózati áttekintés

Ebben a részben egy mintahálózatot mutatunk be (lásd az *ábrát*), amelyben különféle IPv6-kiszolgálók vesznek részt. A rendszerben a következő csomópontokat találjuk:

- Az útvonalválasztó kiszolgáló (pc1) programalapú IPv6-útvonalválasztó kiszolgálóként működik. Ez a gép nyújtja az összes IPv6-csomópontnak az útvonalválasztási szolgáltatást.
- DNS IPv6-kiszolgáló (pc2) biztosítja az IPv6-névfeloldást.
- Két alkalmazáskiszolgáló: az egyik videofolyamokat (pc3), a másik Apache alapú webkiszolgálót (pc4) üzemeltet.
- Ügyfélgépek (pc5-7), amelyeket a kipróbáláshoz használunk.

IPv6-névfeloldás

A tartománynevek az internet jelentéssel bíró, könnyen megjegyezhető „becenevei”, amelyeket a fizikai címekhez rendelünk. A tartománynévrendszer (DNS) lesz az, amelyik ezeket a neveket megkeresi és az internetprotokoll címeivé alakítja át őket. Minthogy a név/IP összefüggések központi tárolása nem lenne célszerű, a tartománynevek és IP-címek listája az egész interneten szétszórva, hatáskörön alapuló hierarchiába rendezve található. Általában elérésünk szolgáltatójának közeli földrajzi környezetében található valahol egy DNS-kiszolgáló; ez a DNS-kiszolgáló fogja hozzárendelni a DNS-kérelmeket címekhez, avagy továbbítja őket az internet más kiszolgálóihoz. Az IPv6 DNS-kérelmek esetében az IPv6-címek megadására egyaránt használhatjuk az A6 és az AAAA írásmódot (syntax).



Hálózatiszerkezet-minta

Az AAAA erőforrásrekordot (amelyet négyes A bejegyzésnek neveznek) adott hosszúságú adatként adjuk meg. Az AAAA segítségével könnyen megadhatjuk az IPv6-névfeloldáshoz tartozó DNS-bejegyzéseket a következő, lényegében az IPv4 „A” bejegyzéseivel azonos módon:

```
$ORIGIN X.EXAMPLE.
N    AAAA 2345:00C1:CA11:0001:1234:5678:9ABC:DEF0
N    AAAA 2345:00D2:DA11:0001:1234:5678:9ABC:DEF0
N    AAAA 2345:000E:EB22:0001:1234:5678:9ABC:DEF0
```

Az A6 erőforrásbejegyzést változó hosszúságú adatként adjuk meg. Az A6-os rendszerben egyszerre több DNS-rekorddal is megadhatunk valamilyen IPv6-címet. Íme az RFC 2874 szabványból kiollózott példa:

```
$ORIGIN X.EXAMPLE.
N    A6 64 ::1234:5678:9ABC:DEF0 SUBNET-1.IP6
SUBNET-1.IP6  A6 48 0:0:0:1::  IP6
IP6  A6 48 0::0        SUBSCRIBER-X.IP6.A.NET.
IP6  A6 48 0::0        SUBSCRIBER-X.IP6.B.NET.

SUBSCRIBER-X.IP6.A.NET. A6 40 0:0:0011::
                          ➥A.NET.IP6.C.NET.
SUBSCRIBER-X.IP6.A.NET. A6 40 0:0:0011::
                          ➥A.NET.IP6.D.NET.
SUBSCRIBER-X.IP6.B.NET. A6 40 0:0:0022:: B-
                          NET.IP6.E.NET.
A.NET.IP6.C.NET. A6 28 0:0001:CA00::
                          ➥C.NET.ALPHA-TLA.ORG.
A.NET.IP6.D.NET. A6 28 0:0002:DA00::
                          ➥D.NET.ALPHA-TLA.ORG.
B-NET.IP6.E.NET. A6 32 0:0:EB00::
                          ➥E.NET.ALPHA-TLA.ORG.
C.NET.ALPHA-TLA.ORG. A6 0 2345:00C0::
D.NET.ALPHA-TLA.ORG. A6 0 2345:00D0::
E.NET.ALPHA-TLA.ORG. A6 0 2345:000E::
```

Ha a fenti kódot AAAA bejegyzésekké alakítanánk át, a következő sorokat kapnánk (lásd a *36. oldalon*):

1. lista A /etc/named.conf

```
options {
directory "/var/named";

// csak a névkiszolgáló beállításait
// gyorstárazzuk
zone "." IN {
type hint;
file "named.ca";
};

// itt adjuk meg a visszacsatolt (loopback)
// névkeresést
zone "localhost" IN {
type master;
file "master/localhost.zone";
allow-update { none; };
};

// itt adjuk meg a visszacsatolt
// névvisszakeresést
zone "0.0.127.in-addr.arpa" IN {
type master;
file "master/localhost.rev";
allow-update { none; };
};

// Itt adjuk meg a secv6
// tartománynév-keresést
// A biztonságos (aláírt) zónafájl neve
// secv6.your.domain.signed
// A szabványos zónafájl a secv6.your.domain
// lesz
zone "secv6.your.domain" IN {
type master;
file "master/secv6.your.domain.signed";
// file "master/secv6.your.domain";
};

// Itt adjuk meg a secv6 tartomány fordított
// névkeresését (AAAA)
zone "secv6.int" IN {
type master;
file "master/secv6.int";
};

// Itt adjuk meg a secv6 tartomány fordított
// névkeresését (A6)
zone "secv6.arpa" IN {
type master;
file "master/secv6.rev";
};

// A titkos kulcsot megnyírtuk, hogy elférjen
key "key" {
algorithm hmac-md5;
secret
"HxbmAnSO0quVxcxBDjmAmjrmhgDUVFcFNcfmHC";
};
```

2. lista A /var/named/master/secv6.your.domain

```
$TTL 86400
$ORIGIN secv6.your.domain.
@ IN SOA secv6.your.domain.
hostmaster.your.domain. (
2002011442 ; Serial number (yyyymmdd-num)
3H ; Refresh
15M ; Retry
1W ; Expire
1D ) ; Minimum
IN MX 10 noah.your.domain.
IN NS ns.secv6.your.domain.
$ORIGIN secv6.your.domain.
ns 1D IN AAAA fec0::1:250:b7ff:fe14:35d0
1D IN A6 0 fec0::1:250:b7ff:fe14:35d0
secv6.your.domain. 1D IN AAAA
fec0::1:250:b7ff:fe14:35d0 1D IN A6 0
fec0::1:250:b7ff:fe14:35d0
pc2 1D IN AAAA fec0::1:250:b7ff:fe14:35d0  1D
IN A6 0
fec0::1:250:b7ff:fe14:35d0
pc3 1D IN A6 0 fec0::1:250:b9ff:fe00:131
1D IN AAAA
fec0::1:250:b9ff:fe00:131
pc6 1D IN A6 0 fec0::1:250:b7ff:fe14:3617
1D IN AAAA
fec0::1:250:b7ff:fe14:3617
pc4 1D IN A6 0 fec0::1:250:b7ff:fe14:35c4
1D IN AAAA
fec0::1:250:b7ff:fe14:35c4
pc5 1D IN A6 0 fec0::1:250:b7ff:fe14:361b
1D IN AAAA
fec0::1:250:b7ff:fe14:361b
pc7 1D IN A6 0 fec0::1:250:b7ff:fe14:365a
1D IN AAAA
fec0::1:250:b7ff:fe14:365a
pc1 1D IN A6 0 fec0::1:250:b9ff:fe00:12e
1D IN AAAA
fec0::1:250:b9ff:fe00:12e
pc1 1D IN A6 0 fec0:0:0:1::1 1D IN AAAA
fec0:0:0:1::1
$INCLUDE
"/var/named/master/Ksecv6.your.domain.+003+27
034.key"
```

```
$ORIGIN X.EXAMPLE.
N    AAAA 2345:00C1:CA11:0001:1234:5678:9ABC:DEF0
N    AAAA 2345:00D2:DA11:0001:1234:5678:9ABC:DEF0
N    AAAA 2345:000E:EB22:0001:1234:5678:9ABC:DEF0
```

Miután az IPv6-névfeloldást beállítottuk, a tartománynév-rendszert (DNSSEC) felvehetjük a DNS-kiszolgálónkra. A DNSSEC három külön szolgáltatást is nyújt: kulcselosztást, adatforrás-azonosítást és -átvitelt, valamint kérelemazonosítást.
A DNSSEC teljes leírását az RFC 2535 szabványban olvashatjuk.

IPv6-támogatás a rendszermagban és a hálózati bináris állományokban

Az IPv6-tal együttműködő BIND-változatunk telepítése előtt elengedhetetlen, hogy a rendszermagban, valamint az IPv6-ot támogató rendszer hálózati bináris állományaiban engedélyezzük az IPv6-támogatást. Ezt a témakört a Linuxvilág 2002. decemberi számában megjelent korábbi cikkünkben már lefedtük (IPv6-támogatás linuxos hálózati csomópontokon), (➲ http://www.linuxvilag.hu/cikkek/26.html#1519). A cikkben megadott útmutatókat követve könnyen felkészülhetünk a legfrissebb, IPv6-támogatással ellátott BIND-változat telepítésére.

BIND és IPv6-támogatás

A BIND legfrissebb változata az Internet Software Consortium honlapján a ➲ http://www.isc.org/products/BIND/BIND9.html címen érhető el. A BIND 9-es változatában szinte a teljes BIND-szerkezetet újraírták. Rengeteg fontos új képesség és fejlesztés mutatkozott be, közülük a számunkra a leglényegesebb az IPv6 támogatása. A BIND 9.x lehetővé teszi, hogy a kiszolgáló IPv6-kapcsolatokon keresztül válaszoljon a DNS-lekérdezésekre, ismeri az IPv6-erőforrásbejegyzéseket (A6, DNAME és a többi), valamint támogatja a bitlánc alapú címkéket. Továbbá BIND 9.x alatt a kísérleti IPv6-megoldáskönyvtárat is használhatjuk. Számos egyéb képességet is kiaknázhatunk, róluk a BIND honlapján olvashatunk bővebben.

A BIND 9.x telepítése

A cikk születésének pillanatában a legutóbbi üzembiztos kiadás a BIND 9.2.1. Telepítési és beállítási útmutatónk ezen a változaton alapul. A BIND telepítéséhez először is töltsük le a legfrissebb BIND-változatot a */usr/src* könyvtárba, majd csomagoljuk ki a következő parancsokkal:

```
% tar -xzf bind-9.2.1.tar.gz
% cd bind-9.2.1
```

Bár az IPv6-támogatás a BIND esetében alapértelmezett, fordításkor külön meg kell adnunk. Továbbá, minthogy ki szeretnénk használni a DNSSEC előnyeit, a BIND-et crypto-támogatással kell fordítanunk. Az OpenSSL 0.9.5a vagy frissebb változat telepítése megfelel a célnak. A szükséges kapcsolókkal ellátott beállítás-parancsfájl futtatása a következőképpen néz ki:

```
% ./configure -enable-IPv6 -with-openssl
```

Végül fordítsuk le és rendszergazdaként telepítsük a csomagot:

```
% make && make install
```

Alapértelmezés szerint a BIND 9 állományai a fájlrendszeren elszórva találhatók. A beállításfájlok a */etc/named.conf*-ba kerülnek; a `named` nevezésű futtatható állomány a */usr/local/sbin* könyvtárba, valamennyi további érintett beállításfájl pedig a */var/named* könyvtárba kerül.

3. lista A /var/named/master/localhost.zone

```
// A localhost.zone a helyi kapcsolatokat
// kezeli a loopback csatolófelületen
// keresztül
$TTL 86400
$ORIGIN localhost.
@ 1D IN SOA @ root (
42 ; serial (d. adams)
3H ; refresh
15M ; retry
1W ; expire
1D ) ; minimum
1D IN NS @
1D IN A 127.0.0.1
```

4. lista A /var/named/master/localhost.rev

```
// a localhost.rev határozza meg a loopback
// csatolófelület fordított DNS kezelését
$TTL 86400
$ORIGIN 0.0.127.in-addr.arpa.
@ IN SOA 0.0.127.in-addr.arpa.
hostmaster.secv6.your.domain. (
42 ; Serial number (d. adams)
3H ; Refresh
15M ; Retry
1W ; Expire
1D ) ; Minimum
NS ns.secv6.your.domain.
MX 10 noah.ip6.your.domain.
PTR localhost.
```

Az IPv6 DNS és a DNSSEC beállítása

A DNS-lekérdezéseket több különféle módon is fel lehet oldani. Például a DNS-kiszolgáló használhatja a gyorstárat, vagy kapcsolatba léphet egy másik DNS-kiszolgálóval az ügyfél nevében, hogy teljesen kiértékelhesse a nevet. Amikor a DNS-kiszolgáló megkap egy lekérdezést, először is ellenőrzi, hogy képes-e saját hatáskörben válaszolni a kiszolgálón megadott helyi zóna beállításmezői alapján. Amennyiben a kérelmezett név megegyezik a helyi zónaadatokkal, a kiszolgáló saját hatáskörben válaszol, ezt az adatot használva a kért név feloldásához. A teljes DNS-lekérdezési folyamatban négy DNS-zóna létezik:

1. ***Elsődleges*** (Master): a kiszolgáló rendelkezik a zónaadat elsődleges másolatával és saját hatáskörben válaszol rá.
2. ***Másodlagos*** (Slave): a másodlagos zóna az elsődleges zóna másolata. Minden másodlagos zónának létezik egy listája az elsődlegesekről, akiktől lekérheti zónamásolatainak a frissítéseit. A másodlagos úgy is beállítható, hogy a lemezen másolatot tartson a zónáról, felgyorsítva az indulást. Egyetlen elsődleges kiszolgáló a terhelés megosztása végett tetszőleges számú másodlagos kiszolgálóval rendelkezhet.
3. ***Csonka*** (Stub): a csonka zóna nagyon hasonlít a másodlagos zónára, a viselkedése is hasonló, azonban a teljes zóna

5. lista A /var/named/master/secv6.rev

```
// A secv6.rev adja meg a secv6 fordított
// tartománykeresését A6 formátumban
$TTL 86400
$ORIGIN secv6.arpa.
@ IN SOA secv6.arpa.
hostmaster.secv6.your.domain. (
2002011442 ; Serial number (yyyymmdd-num)
3H ; Refresh
15M ; Retry
1W ; Expire
1D ) ; Minimum
NS ns.secv6.your.domain.
MX 10 noah.your.domain.
; fec0:0:0:1::/64
$ORIGIN \[xfec0000000000001/64].secv6.arpa.
\[x0250b7fffe1435d0/64]
➥1D IN PTR pc2.secv6.your.domain.
\[x0250b9fffe000131/64]
➥1D IN PTR pc3.secv6.your.domain.
\[x0250b7fffe143617/64]
➥1D IN PTR pc6.secv6.your.domain.
\[x0250b7fffe1435c4/64]
➥1D IN PTR pc4.secv6.your.domain.
\[x0250b7fffe14361b/64]
➥1D IN PTR pc5.secv6.your.domain.
\[x0250b7fffe14365a/64]
➥1D IN PTR pc7.secv6.your.domain.
\[x0250b9fffe00012e/64]
➥1D IN PTR pc1.secv6.your.domain.
```

6. lista A /var/named/master/secv6.int

```
// A secv6.int adja meg a secv6 fordított
// tartománykeresését AAA formátumban
$TTL 86400
$ORIGIN secv6.int.
@ IN SOA secv6.int.
hostmaster.secv6.your.domain. (
2002011442 ; Serial number (yyyymmdd-num)
3H ; Refresh
15M ; Retry
1W ; Expire
1D ) ; Minimum
NS ns.secv6.your.domain.
MX 10 noah.your.domain.
; fec0:0:0:1::/64
$ORIGIN
1.0.0.0.0.0.0.0.0.0.0.0.c.e.f.secv6.int.
0.d.5.3.4.1.e.f.f.f.7.b.0.5.2.0
➥IN PTR pc2.secv6.your.domain.
e.2.1.0.0.0.e.f.f.f.9.b.0.5.2.0
➥IN PTR pc1.secv6.your.domain.
1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0
➥IN PTR pc1.secv6.your.domain.
1.3.1.0.0.0.e.f.f.f.9.b.0.5.2.0
➥IN PTR pc3.secv6.your.domain.
7.1.6.3.4.1.e.f.f.f.7.b.0.5.2.0
➥IN PTR pc6.secv6.your.domain.
4.c.5.3.4.1.e.f.f.f.7.b.0.5.2.0
➥IN PTR pc4.secv6.your.domain.
b.1.6.3.4.1.e.f.f.f.7.b.0.5.2.0
➥IN PTR pc5.secv6.your.domain.
```


helyett kizárólag az elsődleges zóna DNS-bejegyzéseit tartja nyilván. A csonka zónák tartják nyilván, hogy mely DNS-kiszolgálók felelősek a szervezetért. Közvetlenül csatlakoznak a gyökér DNS-kiszolgálóhoz, így megállapíthatják, hogy melyik tartományért melyik kiszolgáló felelős.

4. ***Továbbító*** (Forward): a továbbítózóna a zónából érkező minden lekérést egy másik kiszolgálóra irányít. Mint ilyen tulajdonképpen gyorstárazó DNS-kiszolgálóként működik a hálózaton, illetve képes internetes DNS-szolgáltatásokat nyújtani egy olyan tűzfal mögötti hálózat számára, amely korlátozza a külső DNS-lekérdezések elérését. Természetesen a DNS-kiszolgálónak ilyenkor el kell tudnia érni az internet DNS-szolgáltatásait. Ez a helyzet nagyon hasonló a teljes körű továbbítás (global forwarding) képességhez, itt azonban a továbbítókat zónánként választhatjuk ki.

Példahálózatunkra (lásd az *ábrán*) vetítve a fentieket: a tartományunkban létre kell hoznunk egy elsődleges kiszolgálót: ***secv6.your.domain***. Az *1. lista* azt mutatja be, hogyan kell festenie a ***/etc/named.conf*** beállításainak. (A titkos kulcsot megnyirbáltuk, hogy beleférjen egy sorba.)
A következő lépés a tartományunkat leíró beállításfájlok létrehozása. Megjegyezzük, hogy eddig még semmiféle IPv6-specialitást nem csináltunk. A DNSSEC esetében a ***/var/named/master/secv6.your.domain.signed*** nevezetű fájl a DNS-kiszolgáló zónakulcsával aláírt tartományfájl. Ez igen fontos a DNSSEC alkalmazásakor, hiszen így az ügyfelek valamennyi további DNS-kérelmet hitelesíteni tudnak. A DNS-kiszolgáló zónakulcsa nem egyezik meg a beállításfájlban található kulccsal; a zónakulcs készítésének részleteit a cikk későbbi részében ismerhetjük meg.
A következő szerkesztendő állomány a ***/var/named/master/secv6.your.domain*** lesz. Példánkban (*2. lista*) az AAAA és az A6 formátumot egyaránt használtuk. A fájl végén olvasható `$INCLUDE` meghatározás beilleszti a zónakulcs nyilvános részét. A kulcs titkos részét tartsuk elzárva. Sajátkulcsunk végén a „private” szó olvasható, míg a nyílt kulcsot a „key” szó zárja. Amennyiben kérdések merülnek fel a DNSSEC kulcsokkal és jogosultságaikkal kapcsolatban, olvassuk el a BIND-kézikönyvet. A 2. listában egy jellegzetes IPv6 DNS-tartomány-beállítást mutatunk be a ***secv6.your.domain*** példán keresztül.
A ***/var/named/master*** beállításfájlban olvasható „Hostmaster” lényegében a rendszergazda elektronikus levélcíme, ahol az első pont helyére a kukac (@) jel kerül, a formai megkötések miatt.
Továbbá a 2. lista elején található „IN SOA” szerkezet első száma hagyományosan egy sorozatszám, mégpedig ÉÉÉÉHHNNXX alakban, ahol az XX egy, minden egyes DNS-zóna frissülésekor folyamatosan növekvő szám.
Most nézzük meg, miképpen készíthetünk zónakulcsot! Ennél a lépésnél lényeges lesz, melyik könyvtárban dolgozunk, mivel a kulcsok itt tárolódnak. A kulcsok tárolására a ***/var/named/master*** könyvtárat javasoljuk. A következő parancs 768 bites DSA-kulcsot készít a zónánkhoz:

```
% dnssec-keygen -a DSA -b 768 -n ZONE
➥secv6.your.domain
```

7. lista Az ügyfélgépek /etc/resolv.conf állománya

```
# A secv6 tartomány beindításához indítsuk
# el a namedet a pc2 gépen és használjuk
# ezt az állományt
search secv6.your.domain
nameserver fec0::1:250:b7ff:fe14:35d0
```

8. lista A6 DNS Query

```
pc2% dig 0.0.0.0 secv6.your.domain a6
; <<>> DiG 9.1.0 <<>> 0.0.0.0
➥secv6.your.domain A6
[...]
;secv6.your.domain.  IN A6
;; ANSWER SECTION:
secv6.your.domain.  86400 IN A6 0
➥fec0::1:250:b7ff:fe14:35d0
;; AUTHORITY SECTION:
secv6.your.domain.  86400 IN NS
➥ns.secv6.your.domain.
;; ADDITIONAL SECTION:
ns.secv6.your.domain.  86400 IN A6 0
➥fec0::1:250:b7ff:fe14:35d0
ns.secv6.your.domain.  86400 IN AAAA
➥fec0::1:250:b7ff:fe14:35d0
```

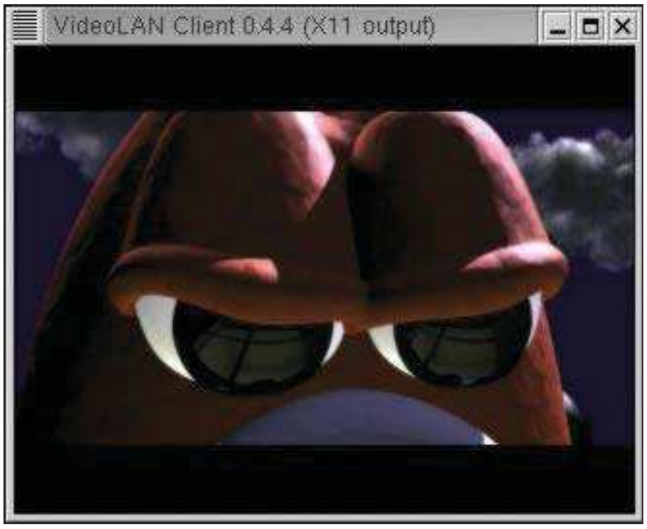
9. lista AAAA DNS Query

```
pc2% dig 0.0.0.0 secv6.your.domain aaaa
; <<>> DiG 9.1.0 <<>> 0.0.0.0
➥secv6.your.domain AAAA
[...]
;secv6.your.domain.  IN AAAA
;; ANSWER SECTION:
secv6.your.domain.  86400 IN AAAA
➥fec0::1:250:b7ff:fe14:35d0
;; AUTHORITY SECTION:
secv6.your.domain.  86400 IN NS
➥ns.secv6.your.domain.
;; ADDITIONAL SECTION:
ns.secv6.your.domain.  86400 IN A6 0
➥fec0::1:250:b7ff:fe14:35d0
ns.secv6.your.domain.  86400 IN AAAA
➥fec0::1:250:b7ff:fe14:35d0
```

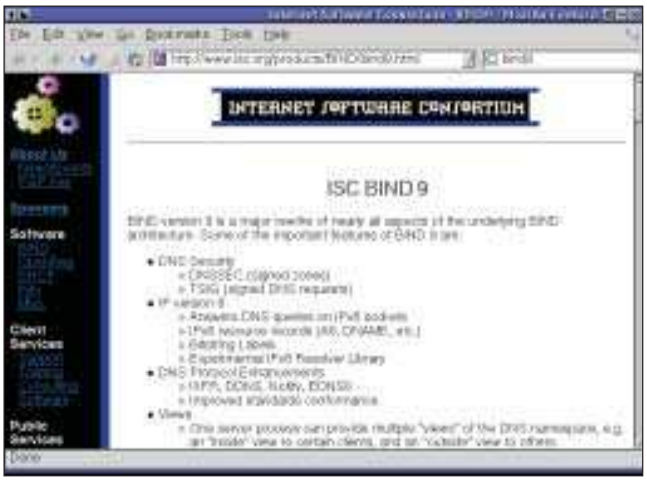
Alapértelmezés szerint az aláírásokhoz valamennyi saját kulcscsal rendelkező zónakulcsot fel kell használni. A kulcsnak vagy a munkakönyvtárban kell lennie vagy a zónafájlba kell beilleszteni. A következő parancs aláírja a ***secv6.your.domain*** zónát, feltételezve, hogy a ***/var/named/master/secv6.your.domain*** fájlban található:

```
% dnssec-signzone -o secv6.your.domain
➥secv6.your.domain
```

A létrejövő kimeneti fájl neve: ***/var/named/master/secv6.your.domain.signed***. Erre a fájlra hivatkozhatunk zónabemeneti állományként a ***/etc/named.conf*** beállításfájlban. További beállításfájlok még a ***localhost.zone*** (*3. lista*), ***localhost.rev*** (*4. lista*), ***secv6.rev*** (*5. lista*) és a ***secv6.int*** (*6. lista*). A két visszafele kereső zónafájl (***secv6.rev*** és ***secv6.int***) között az a különbség, hogy az egyikben az A6 karaktersorozatokat adhatjuk meg (amelyeket a ***secv6.rev***-ben nem kell megfordítani), míg a ***secv6.int***-ben felsorolhatjuk az AAAA formátumú bejegyzéseket. Például a `ping6` kizárólag a ***secv6.int*** tartományba kerülhet, minthogy nem támogatja az A6 formátumot.



1. kép Az IPv6-videominta kimenete



2. kép Az Internet Software Consortium BiND 9 honlapja

A DNS-démon elindítása

Miután elkészültünk a telepítéssel és a beállításokkal, készen állunk a DNS-démon elindítására a pc2 gépen. A `named` alapértelmezés szerint a ***/etc/named.conf*** állományt használja, de ha kívánjuk, a `-c` kapcsoló segítségével más beállításokat is megadhatunk. Figyelembe véve, hogy hová telepítettük a démont, gépeljük be a következőket:

```
pc2% /usr/local/sbin/named
```

Még egy további lépés szükséges IPv6-hálózatunk gépein: frissítsük a ***/etc/resolv.conf*** (*7. lista*) állományokat úgy, hogy tartalmazzák a DNS-kiszolgáló IP-címét. Fontos, hogy az

IP-címet adjuk meg és ne a DNS-kiszolgáló tartománynevét, hiszen ez az az állomány, ahol a rendszer a DNS-címét keresi majd. Más szavakkal, ha itt a DNS-kiszolgáló tartománynevét adjuk meg, honnan fogja tudni a rendszer, hogy milyen szám tartozik a DNS tartománynevéhez?

Kipróbálás

A beállítások kipróbálásához igen egyszerű módszereket alkalmaztunk. Első lépésben ellenőrizzük, hogy a DNS-kiszolgálón engedélyezettek-e az A6-os címek, majd megvizsgáljuk, hogy a DNS-kiszolgáló értelmezi-e az AAAA-címeket. A tesztet a pc2-es gépen hajtottuk végre. Itt most mindössze egyetlen lehetőséget adunk meg – egyébként a lista túlságosan hosszú lenne. Példánkban a dig nevű DNS-kereső eszközzel hajtunk végre keresést a ***secv6*** tartományban, A6 formátumban (*8. lista*). Ezt követően az AAAA formátumot is kipróbáljuk (*9. lista*). Egyik esetben sem adtunk meg keresendő címet, így a 0.0.0.0-t használjuk. Másodszor próbaképpen SSH kapcsolatokat hozunk létre, először IPv6-címet, majd az IPv6-gépneveket használva.

Mintakiszolgáló IPv6-alapokon

IPv6-hálózatunkban két alkalmazáskiszolgálót használtunk: az Apache-t mint webkiszolgálót, illetve a VideoLant a videofolyamokhoz. Mivel az IPv6-névfeloldást videofolyamok olvasása közben is ki szeretnénk próbálni, egy felhasználó a pc5-ös gépen rákapcsolódik a pc3 videofolyamkezelő kiszolgálójára. A videokiszolgáló a pc3 gépen található (fec0::1:250:b7ff:fe14:5768), míg a videoügyfél a pc5 gépen (fec0::1:250:b7ff:fe50:7c). Ha a `tcpdump` segítségével körbeszimatolunk a hálózati adatok között a pc5-ön, néhány csomagot elfoghatunk a videofolyamból. Alább bemutatjuk a nyomkövetés egy részletét:

```
% tcpdump ip6     # csak az IPv6-forgalmat
➥követjük, rootként vagy setuid rootként
➥kell futtatni
[nyissz...]
02:09:26.716040
➥fec0::1:250:b7ff:fe14:5768.32769
➥> fec0::1:250:b7ff:fe50:7c.1234: udp 1316
02:09:26.735805
➥fec0::1:250:b7ff:fe14:5768.32769
➥> fec0::1:250:b7ff:fe50:7c.1234:
➥udp 1316
02:09:26.735971
➥fec0::1:250:b7ff:fe14:5768.32769
➥> fec0::1:250:b7ff:fe50:7c.1234:
➥udp 1316
02:09:26.736082
➥fec0::1:250:b7ff:fe14:5768.32769
➥> fec0::1:250:b7ff:fe50:7c.1234: udp 1316
02:09:26.755810
➥fec0::1:250:b7ff:fe14:5768.32769
➥> fec0::1:250:b7ff:fe50:7c.1234: udp 1316
02:09:26.755935
➥fec0::1:250:b7ff:fe14:5768.32769
➥> fec0::1:250:b7ff:fe50:7c.1234: udp 1316
02:09:26.775787
➥fec0::1:250:b7ff:fe14:5768.32769
➥> fec0::1:250:b7ff:fe50:7c.1234: udp 1316
```

A videó helyesen jelenik meg az a Linux X kiszolgáló X11 kimenetén; a *1. kép* mutatja be a folyam elfogott adatait.

Összegzés

Az IPv6 lassan valósággá válik. A következő néhány évben kiszolgálóinkon képesnek kell lennünk az IPv4 és az IPv6 rendszert is kezelni, még mielőtt a teljes IPv6-átállás megtörténne. A teljes IPv6-átálláshoz különböző mozaikdarabkákat kell összeraknunk, amelyek közül az egyik létfontosságú elem éppen az IPv6-tal együttműködő BIND-megoldás lesz.

Köszönetnyilvánítás

Köszönet az Ericsson Research Corporate részlegnek, amiért jóváhagyták e cikk megjelenését; az Open Systems Labnek munkánk támogatásáért a Linuxon és IPv6-on keresztül; valamint köszönet ***Simon Jubinville***-nek az Open Systems Labtól értékes észrevételeiért.

Linux Journal 2003. október, 114. szám

David Gordon (gordd00@dmi.usherb.ca)
A Québecben lévő Sherbrook Egyetem Számítástechnikai tanszékének hallgatója, épp most készül megszerezni a diplomáját. Az Ericsson Research Labjében segédfelügyelőként dolgozik.

Ibrahim Haddad (Ibrahim.Haddad@Ericsson.com)
Kutató a montreali Ericsson Corporate Kutatórészlegénél, ahol harmadik nemzedékbeli vezeték nélküli IP-hálózatokon dolgozik.

KAPCSOLÓDÓ CÍMEK

BIND-kézikönyv
➲ http://www.crt.se/dnssec/bind9/Bv9ARM.html
BIND ➲ http://www.isc.org/products/BIND/BIND9.html
Az AAAA és az A6 formátum összehasonlítása
➲ http://www.ietf.org/proceedings/02mar/I-D/draft-ietf-dnsext-aaaa-a6-01.txt
DNSSEC ➲ http://www.ietf.org/rfc/rfc2535.txt
DNSSEC és IPv6 A6
➲ ftp.rfc-editor.org/in-notes/rfc3226.txt
DNSSEC hitelesítés-szolgáltató
➲ ftp.rfc-editor.org/in-notes/rfc3008.txt
IPv6 HOWTO
➲ http://www.bieringer.de/linux/IPv6/IPv6-HOWTO/IPv6-HOWTO.html
Linuxos IPv6-megoldások
➲ http://www.linuxjournal.com/article/5468
IPv6 támogatás DNS-hez
➲ http://www.ietf.org/rfc/rfc2874.txt
IP Version 6 címzési rendszer
➲ http://www.rfc-editor.org/rfc2373.txt
Linux-rendszermag ➲ http://www.kernel.org
Ibrahim Haddad és *Marc Blanchet*: Supporting IPv6 on a Linux Server Node, Linux Journal 2002. augusztus
➲ http://www.linuxjournal.com/article/4763

Egységes és objektumközpontú adatbázis-kezelés (2. rész)

A folytatásában a bemutatjuk, hogyan kezelhetünk Qt rendszer alatt SQL sormutatót, és készíthetünk grafikus felületet programoknak a Qt Designer segítségével.

Egy további adatkezelő osztály a `QSqlCursor`, amit közvetlen adatkezeléshez is használhatunk, de a fejlesztők fő célja az volt, hogy a grafikus elemek adatforrásának a szerepét is betölthesse. Ebben a pontban csak az adatkezelő vonatkozásokat tekintjük át, így példaprogramjaink továbbra is karakteres képernyőn fogják megjeleníteni a kimenetüket.
Ez az osztály lehetővé teszi egy táblára vagy nézetre (view) épülő SQL `select` parancs megalkotását és végrehajtását, amelynek az SQL sormutatónak nevezett eredménytábla lesz az eredménye. A sormutató sorain haladva adatmódosító utasításokat is kiadhatunk. A gyors megértés érdekében lássunk egy egyszerű példát! A feladat legyen az, hogy az `ujnevek` tábla azon sorait listázzuk ki a képernyőn, ahol a `honap=1` feltétel teljesül (lásd még az 55. CD Magazin/Qt/sql_3.cpp könyvtárában).

```
1. //
2. // sql_3.cpp
3. //
4. #include <iostream>
5. #include <qapplication.h>
6. #include <qsqldatabase.h>
7. #include <qsqlcursor.h>
8.
9. using std::cout;
10.
11.//--- A program indulási pontja ---
12.int main(int argc, char **argv)
13.{
14. QApplication app(argc, argv, false);
15. QSqlDatabase *pg = QSqlDatabase::addDatabase
    ➥( "QPSQL7", "PG_CS_ADATOK" );
16. if ( !pg ) { cout <<
    ➥"Hiba az illesztőprogram betöltésekor!";
    ➥return (1); }
17.
18. pg->setDatabaseName("cs_adatok");
19. pg->setUserName("postgres");
20. pg->setPassword("111111");
21. pg->setHostName("localhost");
22. pg->setPort( 5432 );
23.
24. if ( !pg->open() )
25. {
26. cout << "Hiba: az adatbázis nem nyitható
    ➥meg!";
27. return (1);
28. }
29.
30. QSqlCursor cur("ujnevek", true, pg);
31. cur.select("honap='1'");
32.
33. while ( cur.next() )
34. {
35. cout << "\n"
    ➥<< cur.value( "honap" ).toString() << "\t";
36. cout
    ➥<< cur.value( "nap" ).toString() << "\t";
37. cout << cur.value( "nevnap" ).toString();
38. }
39.
40. pg->close();
41. return 0;
42.}
```

Az 1–29. sor ismert fogalmakat tartalmaz. A 30. sorban egy `cur` nevű `QSqlCursor` típusú objektumot határoztunk meg. A létrehozó első értékében megmondtuk, hogy a végrehajtandó `select` az `ujnevek` táblára épüljön. A második, `true` érték azt kéri, hogy önműködően jöjjön létre a sorokat és a mezőket elérhetővé tevő `QSqlRecord` és `QSqlField` objektumegyüttes; a harmadik érték tisztázza azt, hogy melyik adatbázis-kapcsolaton szeretnénk dolgozni. A 31. sorban történik a `select` parancs véglegesítése és végrehajtása.
A `select()` tagfüggvénynek több alakja is van, így megadhatunk szűrőket (esetünkben `honap='1'`), sorrendeket, illetve mindkettőt egyszerre. A fentiek alapján létrehozott, ténylegesen végrehajtott SQL-parancs a következő lesz:

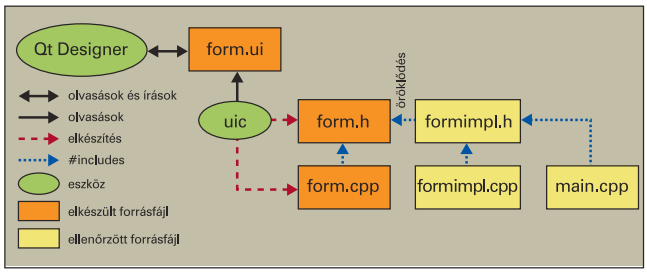
```
select * from ujnevek where honap='1';
```

A 33–38. sorban egy ciklusban kiírjuk a lekérdezett adatokat. A `cur.value` egy `QSqlRecord` objektumra mutat, amelyen keresztül az oszlopok nevét (használhatnánk indexet is) használva könnyen hozzáférhetünk adatainkhoz.
A fenti program által kiírt lista nem biztos, hogy a napokat nagyság szerinti sorrendben hozza le. Amennyiben ezt a feladatot is meg szeretnénk oldani, akkor a `cur.select()` hívást a következőképpen át kell fogalmazni:

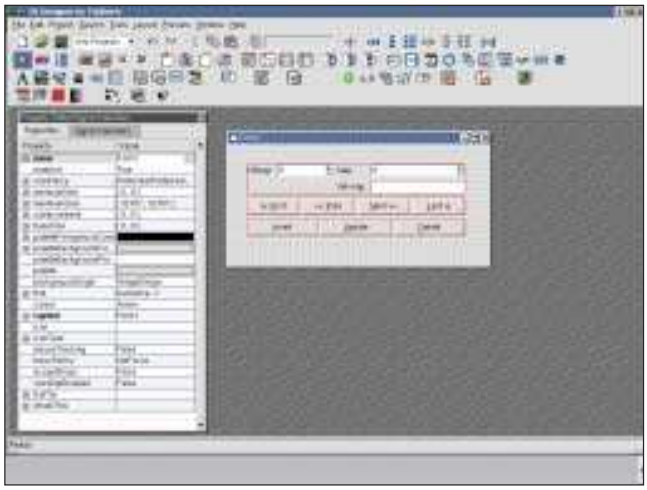
```
...
QSqlIndex idIndex;
QStringList ind = QStringList() << "nap";
idIndex = cur.index( ind );
cur.select("honap='1'", idIndex);
...
```

Látható, hogy a szűrőfeltételt a `select()` első értékében meghagytuk. A második értéket egy `QSqlIndex` objektum segítségével lehet megadni, ennek létrehozását a sormutatóobjektum az `index()` tagfüggvénnyel, a fenti módon támogatja. A létrehozott és végrehajtott SQL-parancs ez esetben a következő lesz:

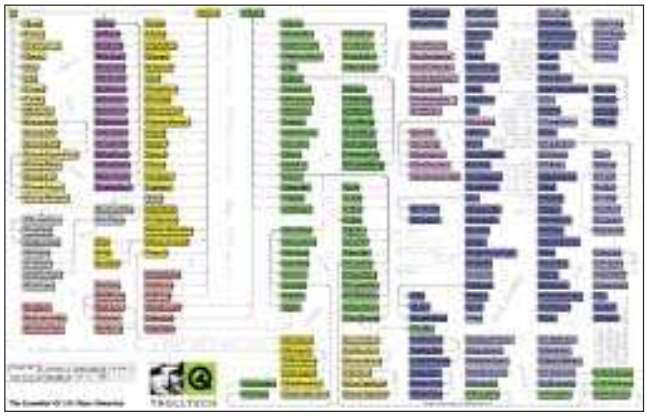
```
select * from ujnevek where honap='1' order
by nap;
```



A C++-forráskód létrehozása



A Qt Designer



A Qt osztálydiagram

Most már tudjuk, hogyan hozhatunk létre egy SQL sormutatót – a következőkben annak módját ismerjük meg, hogyan lehet a sormutató használatával adatmódosító műveleteket végrehajtani. Már az elején felhívjuk a figyelmet a `QSqlCursor::setPrimaryIndex()` tagfüggvényre, ami arról ad tájékoztatást, hogyan lehet egy táblában vagy nézetben egy sort egyértelműen azonosítani. Ezt a hívást az `update` és `delete` műveletek igénylik, az `insert` enélkül is hibátlanul működik. Amennyiben a táblának van elsődleges kulcsa és ez csak egyetlen mezőből áll, ez a hívás elhagyható. Emlékezzünk vissza, hogy egy sormutatónaknak mindig van egy pillanatnyi sora, ami egyébként `QSqlRecord` típusú. Ez a sor egy átmenetitár-területen van létrehozva, aminek a címét a `QSqlCursor` `primeInsert()`, `primeUpdate()` vagy `primeDelete()` tagfüggvényével le is kérhetjük, attól függően, hogy mit szeretnénk csinálni.
A `QSqlCursor` osztály kalkulált mezőket is tartalmazhat. Nézzük meg az alábbi programrészletet, ami az `update` műveletet mutatja be; a program az `i.Imrel.névnap` értéket `irImre(én)le` értékre cseréli ki.

```
1. ...
2. QSqlCursor cur("ujnevek", true, pg);
3. QSqlIndex idIndex;
4. QStringList ind = QStringList()
                ➥<< "honap" << "nap";
5. idIndex = cur.index( ind );
6. // A következő _ sor fontos a del()
   // és az update() működéséhez
7. cur.setPrimaryIndex( idIndex );
8. cur.select( "nevnap='Imre'" );
9. cur.next();
10.QSqlRecord *rec = cur.primeUpdate();
   // Csak lekérjük
11. rec->setValue("nevnap", "Imre (én)");
12. cout << "\nRekordszám: " << cur.update();
13....
```

A fenti kódrészlet a szokásos adatbázis-kapcsolódással és az alkalmazásobjektum létrehozásával indul, amit az 1. sorban `io...` io-tal jeleztünk. Az SQL `update` művelet szempontjából érdekes kódrészletet a 2–12. sorban emeltük ki. A 2. sorban létrehoztunk egy `cur` sormutatóobjektumot az `ujnevek` táblára. A 3–5. sor egy olyan `QSqlIndex` objektum (melynek neve `idIndex`) létrehozását mutatja, amiben az `ujnevek` tábla `honap` és `nap` oszlopai az indexelés szempontjai. A `cur` objektum `index()` tagfüggvénye szolgáltatja az indexelőobjektumunk létrehozását, amit ezután a 7. sorban a `setPrimaryIndex()` hívásban használni tudunk, így megteremtettük az `update` művelet használhatóságát. A 8. sor lefuttatja az SQL `select` lekérdezést a `nevnap='Imre'` szűrés mellett. A 10. sorban a `rec` rekordmutatót az `update` átmeneti tár memóriacímre állítjuk, így bármelyik is legyen a sormutató pillanatnyi sora, a `rec` is eléri azt mivel `QSqlRecord` osztálybeli objektum. A `rec` mutatón keresztül könnyen módosítható a sormutató pillanatnyi sora. A `setValue()` tagfüggvény első értéke az oszlopnév, míg a második tulajdonság (argumentum) a beállítandó új érték. A 12. sor lényegi része a `cur.update()` hívás, ami egy SQL `update` parancs kiadását jelenti, illetve visszaadja a sikeresen módosított sorok számát.
Az SQL sormutató használatával végrehajtható törlés művelet is egyszerű. Ebben az esetben a következő kódsorokat kell használnunk a megfelelő helyeken:

```
QSqlRecord *rec = cur.primeDelete();
// Csak lekérjük
cout << "\nRekordszám: " << cur.del();
```

A beszúrás műveletet a következő kódminta szemlélteti:

```
QSqlRecord *rec = cur.primeInsert();
// Csak lekérjük
rec->setValue("honap", 11);
rec->setValue("nap", 5);
rec->setValue("nevnap", "Imre" );
cout << "\nRekordszám: " << cur.insert();
```

Hibakezelés

Az SQL alapú hibák természetesen egy-egy adatbázis-kapcsolat viszonylatában merülnek fel, így a hibás állapotot is egy `QSqlDatabase` objektumon keresztül tudjuk lekérdezni, amire a `QSqlError` típusú objektumot visszaadó `lastError()` tagfüggvény szolgál. Például a következő kódsor a képernyőre az utolsó SQL-műveletben felmerült hiba típusát (a `pg` az adatbázis-kapcsolatot megtestesítő objektum) írja ki:

```
cout << "\nA hiba = " <<
➥(pg->lastError()).type();
```

Adatbázis-kezelő grafikus programok készítése

Eddig kizárólag olyan programokkal foglalkoztunk, amelyekben csak a szöveges képernyőt és a Qt SQL-modul illesztő- és adatkezelő részeit használtuk. Most áttekintjük a Qt SQL rendszer harmadik nagy részterületét, az adat(bázis)függő elemek (widgets) használatát és a grafikus programok készítésének a módját. Most csak egyszerű példákat mutatunk be, nem feltételezzük, hogy az olvasó ismeri a Qt grafikakészítés módszereit: jel-, illetve foglalat- (signal/slot) szerkezeteket, akciókat, eseményeket és a grafikus vezérlőket.
Nézzünk meg egy kicsi, de teljes mintapéldát, ami a nevek táblánk táblázatszerű kezelését mutatja meg (lásd még: 55. CD Magazin/Qt/tablazat_1.cpp), :

```
1. //
2. // tablazat_1.cpp
3. //
4. #include <iostream>
5. #include <qapplication.h>
6. #include <qsqldatabase.h>
7. #include <qsqlcursor.h>
8. #include <qdatatable.h>
9.
10.using std::cout;
11.
12.int main(int argc, char **argv)
13.{
14. QApplication app(argc, argv, true);
15. QSqlDatabase *pg =
        ➥QSqlDatabase::addDatabase
        ➥( "QPSQL7", "PG_CS_ADATOK" );
16. if ( !pg ) { cout << "Hiba a driver
        ➥betöltésekor!"; return (1); }
17.
18. pg->setDatabaseName("cs_adatok");
19. pg->setUserName("postgres");
20. pg->setPassword("111111");
21. pg->setHostName("localhost");
22. pg->setPort( 5432 );
23.
24. if ( !pg->open() )
25. { cout << "Hiba: az adatbázis nem
              ➥nyitható meg!";
26. return (1);
27. }
28.
29. QSqlCursor kurzor("ujnevek", true, pg);
30. kurzor.select("");
31.
32. QDataTable *t = new QDataTable(
        ➥&kurzor, false );
33.
34. t->addColumn("honap", "Hónap");
35. t->addColumn("nap", "Nap");
36. t->addColumn("nevnap", "Névnap");
37.
38. app.setMainWidget( t );
39.
40. t->refresh();
41. t->show();
42.
43. return app.exec();
44.}
```

Előzetesen érdemes megjegyezni, hogy a Qt-elemek mindegyike tároló- (container) és vezérlőelem egyaránt lehet, ami a veterán Windows-programozók számára talán még meglepő is. Bármelyik elem betöltheti a fő vezérlő szerepét is. A `tablazat_1.cpp` programban a `QDataTable` összetevőt alkalmaztuk, egyben az az alkalmazás fő vezérlője is. A 14. sorban meghatározott `app` alkalmazás az objektum harmadik értéke `true`, ami lehetővé teszi a grafikus elemek használatát. Ez újdonság az eddigiekhez képest. A 29–30. sorokban hoztuk létre azt a `kurzor` objektumot, ami egyben a grafikus tábla vezérlő adatforrása is lesz. A 32. sorban hozzuk létre a `t` mutatót, ami egy `QDataTable` objektumra mutat. A létrehozó első értéke a `kurzor` objektumunk címe. A második, `false` értékű azt jelenti, hogy mi akarjuk meghatározni a táblázat oszlopait. Amennyiben itt `true` értéket adunk, az oszlopok önműködően a sormutató oszlopaiból lesznek képezve (***auto populate*** üzemmód). Ebben az esetben a 34–36. sorra nem volna szükség. Az `addColumn()` tagfüggvények az oszlopokat határozzák meg, amelyekben az értékek jelentése ebben a sorrendben: egy sormutató oszlopváltozó, az oszlop fejléc címkéje. A 38. sorban táblázatunkat beállítjuk az alkalmazás fő vezérlőjének. A 40. sor `refresh()` tagfüggvénye felolvassa a kurzor sorait (hatékony, mert csak annyit, amennyi a megjelenítéshez szükséges) és ki is listázza azt. A 41. sor a vezérlőt megjeleníti a képernyőn.
A táblázatos kezelést megvalósító programunk futási képét az 55. CD Magazin/Qt könyvtárban lévő ***tablazat.jpg*** kép mutatja.

Űrlap alapú programok készítése

A Qt rendszerben a Delphi/C++ Builder-űrlapokhoz hasonló bonyolultságú űrlapokból álló alkalmazásokat is készíthetünk. Már most célszerű megemlíteni a Qt Designer nevű kiváló eszközt, amivel grafikusan készíthetjük el az űrlapokat, illetve az eseménykezelők és az események összekötését. A Qt Designer a Java nyelv elgondolásához hasonló – fejlett önműködő elemelrendezés-kezelővel (layout manager) rendelkezik. Az adatbázis alapú űrlapok készítésének áttekintéséhez nézzük meg a ***form1.cpp*** program forráskódját.

```
1. //
2. // form1.cpp
3. //
4. #include <qapplication.h>
5. #include <qdialog.h>
6. #include <qlabel.h>
7. #include <qlineedit.h>
8. #include <qsqlform.h>
9. #include <qsqldatabase.h>
```

```
10.#include <qsqlcursor.h>
11.#include <qdatatable.h>
12.#include <qlayout.h>
13.
14.////////////////////////////////
15.class TMyForm : public QDialog
16.{
17. public:
18. TMyForm();
19.
20. QSqlDatabase *pg;
21. QSqlCursor kurzor;
22.
23. QLineEdit *lNev;
24. QLabel *lHonap;
25. QLabel *lNap;
26. QGridLayout *grid;
27.};
28.
29.TMyForm::TMyForm()
30.{
31. pg = QSqlDatabase::addDatabase( "QPSQL7",
        ➥"PG_CS_ADATOK" );
32. // Kapcsolódás az adatbázishoz
33. pg->setDatabaseName("cs_adatok");
34. pg->setUserName("postgres");
35. pg->setPassword("111111");
36. pg->setHostName("localhost");
37. pg->setPort( 5432 );
38. pg->open();
39.
40. QSqlCursor kurzor("ujnevek", true, pg);
41. kurzor.select("honap='1'");
42. kurzor.next();
43.
44. lNev = new QLineEdit(this);
45. lHonap = new QLabel(this);
46. lNap = new QLabel(this);
47.
48. grid = new QGridLayout(this);
49. grid->addWidget( lNev, 0, 0 );
50. grid->addWidget( lHonap, 1, 0 );
51. grid->addWidget( lNap, 1, 1 );
52. grid->activate();
53.
54. QSqlForm aform(this);
55. aform.setRecord( kurzor.primeUpdate() );
56. aform.insert(lNev, "nevnap");
57. aform.insert(lHonap, "honap");
58. aform.insert(lNap, "nap");
59. aform.readFields();
60.}
61.
62.//--- A program indulási pontja ---
63.int main(int argc, char **argv)
64.{
65. QApplication app(argc, argv, true);
66.
67. TMyForm *f = new TMyForm();
68. f->show();
69. app.setMainWidget(f);
70. return app.exec();
71.}
```

A 15. sorban egy `TMyForm` osztályt vezetünk be, ami a `QDialog` utóda. Az osztály 15–27. sorában lévő megadásából (declaration) látható, hogy ugyanazokat a grafikus elemeket (címke, editor) használjuk, mint amikor nem használunk mögöttük adatforrásokat. A ***form1.cpp*** programban az osztályunk létrehozója minden lépést elvégez ahhoz, hogy az űrlapunk adatfüggő űrlap legyen. A 40–42. sor létrehozza az űrlap adatforrását. A 44–46. sor három új grafikus vezérlőt létesít, amiket a 48–52. sorban meghatározott `grid` nevű elrendezés-kezelő önműködően felpakol az űrlapunkra.
Nekünk csak azt kell megmondanunk, hogy a képzeletbeli rácscellák közül melyik vezérlőt hol (sor, oszloppozíció) szeretnénk látni. Végül az 54–59. sor az, ami űrlapunkból adatbázis-függő űrlapot varázsol. A feladatot egy `QSqlForm` típusú `aform` nevű objektum végzi el, aminek a tulajdonosa a `this`, azaz a mi későbbiekben létrehozandó `TMyForm` (a neve `(*f)` lesz és a `main()` függvényben látható) objektumunk lesz. Az 55. sor jegyzi be, hogy az adatforrás rekordjait honnan kell majd venni. Az 56–58. sor világosítja fel `aform` objektumunkat arról, hogy egy adott elemhez melyik sormutató-oszlopot szeretnénk társítani, majd az 59. sorban elvégezzük az elemek tényleges adatfeltöltését. A 63–71. sorban megírt `main()` főfüggvény működése során létrehozzuk a `TMyForm` objektumunkat, ami egyben a program fővezérlője is lesz.
A ***form1.cpp*** futtatható változatát az 55. CD Magazin/Qt könyvtárban lévő form.jpg képen láthatjuk.
Űrlapunk jelenleg csak annyit tud, hogy a sormutató első sorát megjeleníti egy űrlapos űrlapjában (Fruzsina, 1, 1 névnap, hó, nap). A navigálás, az adatmódosítás és az adatbázisba való mentés még nincs megoldva. A név mezőt már most is egy `QLineEdit` vezérlő tartalmazza, ezért a megjelenített első rekordnak ezt a mezőjét módosíthatjuk. De hogyan érvényesíthető ez a módosítás az adatbázisban? Erre szolgál a `QSqlForm::writeFields()` tagfüggvény. Az adatbázisban való módosítást tehát a következő kódrészlet mutatja be:

```
... változtatások a form vezérlői segítségével
aform.writeFields();
kurzor.update();
...
```

Ezt a kódrészt valakinek meg kell hívnia, azaz jellemzően egy eseménykezelőben kívánatos szerepeltetni. A `TMyForm` osztályt ki kell egészíteni egy `PUBLIC SLOT` szerepű, `mentes()` nevű tagfüggvénnyel, ami ezt az eseménykezelést valósítja meg. Ez a tagfüggvény így egészítené ki az osztályunkat:

```
class TMyForm : public QDialog
{
public:
TMyForm();
public slots:
void mentes();
...
}
```

A `public slots` címkét a `Qt moc` (Meta Object Compiler) program tünteti el, illetve szabványos C++-programot hoz létre belőle.
Ezek után feltehetünk egy `QPushButton` vezérlőt az űrlapra, amelynek a megvalósítása vázlatosan így nézne ki:

```
...
QPushButton *btnMentes = new
QPushButton("&Mentés", this);
...
grid->addWidget(btnMentes, 2, 2);
// Elhelyezzük az űrlapon
...
connect(btnMentes, SIGNAL(clicked()),
➥this, SLOT(mentes()));
...
```

A `btnMentes` gomb létrehozása után azt a már előzőleg létrehozott `grid` elrendezés-kezelő objektumunk segítségével feldobjuk az űrlapunkra. A `connect()` nagyon fontos függvény, mert ez köti össze az eseményt keltő objektum jelzését (signal) az eseményt kezelő objektum(ok) eseménykezelőivel, azaz a foglalattal. A `connect()` függvény értékeinek értelmezése:

1. A jelzést küldő objektum címe.
2. A jelzés (ez csak egy olyan tagfüggvényfej, amit soha nem kell megvalósítani, de meg kell határoznunk az értékeit, ha vannak; a `click()` jelzés érték nélküli).
3. Az eseményt kezelő objektum címe.
4. Az eseményt kezelő függvény, amit foglalatnak nevezünk (foglalat, mert ide csatlakoztathatjuk az eseménykezelést).

A ***form1.cpp*** programnak tehát azt a változatát is elkészíthetjük, amikor az adatbázisba mentő kód egy gomb megnyomása utá lefut.
A navigálás is hasonlóan oldható meg. Be kell vezetnünk a navigálógombokat (vagy más erre használható vezérlőt), amik mögé olyan eseménykezelőket (foglalatokat) kell írnunk, amik meghívják a `kurzor.next()...` tagfüggvényeket.

A Qt Designer használata

A gyakorlatban a komolyabb kiépítettségű űrlap alapú programjainkat általában nem soronként írjuk meg, így a Qt rendszerben is megtalálható a Qt Designer nevű grafikus fejlesztői környezet. A Designer az adatbázisfüggő űrlapok készítését is lehetővé teszi (*1. kép*).
Ez az eszköz a grafikai tervezés eredményét (űrlapok és eseménykezelők) egy ****.ui*** kiterjesztésű felhasználói kezelőfelület leíró XML-fájlban tárolja. Az ui fájlokból valódi C++-forrásprogram hozható létre, amelynek elkészítési folyamatát ábránk mutatja. A létrehozott C++-forrásprogram az űrlap osztályát megvalósító kódot is tartalmazza, amit az alkalmazás más helyein természetesen kedvünk szerint használhatunk, de a létrehozott forrásfájlokat kézzel ne változtassuk meg, illetve szükség esetén használjuk az öröklés lehetőségét. A Qt Designer szorosan össze van építve a KDevelop fejlesztői környezettel is, a két eszköz együttes használata nagyon termékeny.
A Designerben látható fenti űrlap egy `QDataBrowser` osztály alapú űrlap, ami a már ismert `QDataTable` osztály űrlap alapú testvére. Ez az osztály fogja tartalmazni a már ismertetett `QSqlForm` objektumot, így nekünk nem kell egy külön `TMyForm` szerepű osztályt gyártanunk. A fenti űrlapot a Designer segítségével a következő lépésekben hoztuk létre:

1. Egy teszt nevű új projekt létrehozása.
2. A ***Tools/Database/DataBrowser*** menüpont mögötti varázsló futtatása (létrehozza az űrlapot).
3. Mentés után a következő fájlok találhatók a lemezen: ***teszt.pro*** (ez a projektfájl), ***ab_qt_form1.ui*** (a megtervezett űrlap egy XML formátumban tárolt leírása), ***ab_qt_form1.h***.

A ***main.cpp*** főprogramot a következő minta alapján lehet létrehozni, illetve a Designerben ezt a fájlt is hozzá kell adni a ***teszt.pro*** projektünkhöz.

```
1. #include <qapplication.h>
2. #include <qsqldatabase.h>
3. #include
    ➥"/home/inyiri/cpp/qt_sql/.ui/ab_qt_form1.h"
4.
5. int main( int argc, char ** argv )
6. {
7. QApplication a( argc, argv );
8.
9. QSqlDatabase *pg =
   ➥QSqlDatabase::addDatabase( "QPSQL7" );
10. pg->setDatabaseName("cs_adatok");
11. pg->setUserName("postgres");
12. pg->setPassword("111111");
13. pg->setHostName("localhost");
14. pg->open();
15.
16. Form1 *w = new Form1;
17. w->show();
18. a.connect( &a, SIGNAL( lastWindowClosed()
    ➥), &a, SLOT( quit() ) );
19. return a.exec();
20.}
```

A 9–14. sorból láthatjuk, hogy az adatbázishoz történő kapcsolódást nekünk kell beprogramozni. A 16. sorban található `Form1` osztály a Designerben lett meghatározva. Természetesen nincs semmilyen korlát arra vonatkozóan, hogy a projektünk hány űrlapot tartalmazhat.
Ebből a négy fájlból a következő lépésekkel állítható össze egy futtatható program:

1. `qmake teszt.pro`
   A Qt `qmake` segédprogramja a Designer-projektfájl alapján létrehoz egy szabványos `Makefile`-t. Ez az eljárás a felületfüggetlen programozást is lehetővé teszi.
2. `make`
   A cikk példáit és a fordítást támogató `Makefile`-okat a magazin 55. CD-jének Magazin/Qt könyvtárában találhatjuk meg.

Mindenkinek kellemes programozást kívánok!

Irodalomjegyzék

1. A Qt csomaghoz adott leírások
➲ http://doc.trolltech.com
➲ http://www.trolltech.com
2. Linuxvilág 2002. október és 2002. december: Qt-programozás
3. Joe Celko: SQL felsőfokon
Kiskapu Kiadó (ISBN: 963 9301 20 5)

Nyíri Imre (inyiri@mol.hu)
Jelenleg a MOL Rt.-nél dolgozik. Informatikai vállalkozásában az Internet, a Linux, valamint a Java-programozás gyakorlati hasznosításával foglalkozik. Örök szerelme a C++ maradt.

Mindenre van megoldás!

Avagy pazarlás lenne egy régi gépet csak azért kidobni, mert nem tegnapi rajta a dátum.

Az elmúlt két hétben egy meglehetősen régi géppel dolgoztam. Gyakorlatilag egy rossznak kikiáltott, „halálra ítélt” gépről van szó, amelynek a képességei mai szemmel nézve igen szerénynek tűnnek.
Egészen pontosan egy Pentium 75 MHz-es processzorral szerelt, 16 MB EDO memóriát tartalmazó gépről van szó, aminek a háttértára 859 MB kapacitású, 5,25"-os méretű, igen koros merevlemez. Ezenkívül a képért egy S3 Trio felel. A gép olyan szinten állt, hogy gyakorlatilag ujjnyi volt benne por, és a CD-meghajtója sem akart semmit beolvasni. Egyetlen „használhatónak” ítélt eszköze a monitor volt, ami talán még ért is valamennyit. Egy igazi számítógéphívő számára viszont nincs rosszabb érzés, mint csak úgy kidobni egy számítógépet – én is önkéntelenül a megmentésén kezdtem el fáradozni. Ennek a folyamatát fogom most leírni.

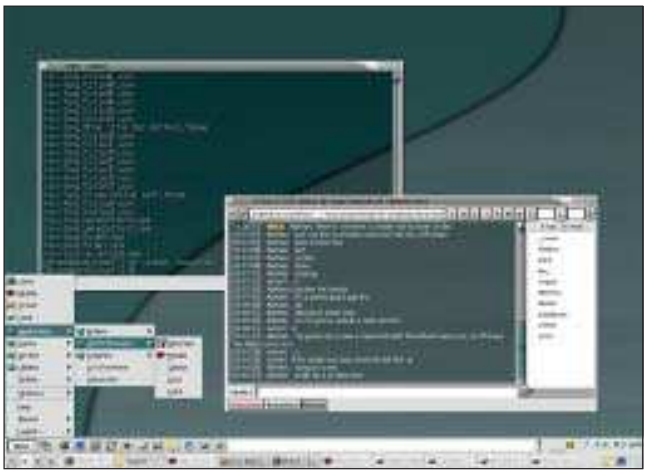
Öreg gép nem vén gép

Az első feladat a gép szétszedése volt (közvetlenül a bekapcsolás után), hogy megnézhessem, mi újság odabent. A bekapcsolás után a használhatatlan CD-meghajtóval szembesültem.
A gép takarításért kiáltott, ugyanis az ujjnyi por sem az elektronikának, sem a merevlemeznek nem tesz jót. Azonnal nekifogtam, és gyakorlatilag darabokra kaptam a gépet. A takarítást követő összerakás során a rossz CD-meghajtót már vissza sem tettem, hanem finoman áthelyeztem a kukába, és feltúrtam a saját „leselejtezett eszközök” dobozomat, majd beletettem a meglelt, nem kevésbé öreg, de működőképes CD-meghajtómat. Ekkor a gép már teljesen használhatónak tűnt, pontosabban minden alkatrész működött. Ezután a hozzávaló programok keresése következett. Mivel más rendszer szóba sem jöhetett, és bő választékom van Linuxokból és általában a nyílt forrású rendszerekből, nem volt gond utánanézni, vajon mi is futna emberi sebességgel ezen a kielégítőnek nem nevezhető gépen. Több rendszert is kiválasztottam, amelyekkel érdemesnek tűnt kísérletezgetni. A feladatnak ebben a szakaszában kell eldöntenünk, hogy pontosan mire is szeretnénk használni egy ilyen kiépítésű gépet. Igazából két fő „feladatkör” jöhet számításba: vagy kiszolgálóként, vagy ügyfélként üzemeltethetjük. Ha kiszolgáló lesz, akkor egyszerűen csak védje a belső hálózatot vagy legyen rajta FTP-kiszolgáló is? Ennek a megoldásnak nagy előnye, hogyha egyszerűen csak a védelem és az internetmegosztás a feladata, akkor igazából elegendő hozzá a jelenlegi kiépítés, főként, hogy így nincs szükség az X-felületre. Telepíteni ugyan lehet (hátha mégis kell), de ez nem jelenti azt, hogy akkor is futnia kell, ha végképp semmi szükség nincsen rá (általában erre lehet számítani egy kiszolgáló esetében.)
Ha azonban ügyfélként működtetjük, felmerülnek bizonyos nehézségek. Grafikus felület például ilyenkor mégiscsak szükséges rá, ami szemmel látható sebességcsökkenéssel jár, még akkor is, ha egy jóval kevesebb erőforrást használó ablakkezelőt állítunk be, mint a KDE. Talán mondanom sem kell, hogy ezen a kiépítésen a KDE és a Gnome alapból száműzve lesz, mivel a rendszer komolyan lelassulna tőlük. További szempont, hogy a processzor sebessége mennyi terhelés mellett mutat fel megbízható, emberi sebességet; ezenkívül – mint a továbbiakban látni fogjuk – fontos a memória mérete is. Ezek tehát mind-mind lényeges összetevők.

Mit is válasszak?

A fentiek fényében elkezdtem gondolkodni – segítségül hívtam a kávéfőzőt is –, hogy mindezeken felül mit érdemes még figyelembe vennem, esetleg létfontosságú észben tartanom…
Egy ilyen összetett kérdéskör esetén mindig az legyen az első lépés, hogy pontosan meghatározzuk a célfelhasználási területet. Ez nekem sem sikerült elsőre, ugyanis úgy gondoltam, hogy talán egy otthoni kis kiszolgálónak jó, ha ADSL-kapcsolatot lehet vele felépíteni, és van benne `iptables`. Mivel ezt szinte mindegyik terjesztés tudja, úgy terveztem, hogy továbbviszem a dolgot, és talán egy kicsit többet is ki lehet hozni belőle. Ezért alapfelhasználásaként kiszolgáló lesz, de ha elbír egy grafikus felületet egy Abiworddel, akkor talán ügyfélként is működhet a hálózat kiszolgálása és védelme mellett.
Ekkor jönnek azok a nagyon fontos szempontok, amiket érdemes alaposan végig gondolni, mégpedig az eszközök és a programok kölcsönös összefüggésének pontos vizsgálata. Arra gondolok, hogy a jó programválasztás tényleg csodákra képes, ha szempontként figyelembe vesszük, hogy az adott programnak milyen a teljesítménye, az erőforrásigénye a gyártó által megadott eszközökön.
Ez egy kicsit nyilván csalóka, ugyanis egy Linux-változat majd' a végletekig méretezhető (a személyes rekordom egy 486-os 4 MB memóriával, amelyen a grafikus felület nem evilági módon tekeri a lemezt és nagyon lassú is, de karakteresen üzemel). Ez nem jelenti azt, hogy hozza a jó sebességet, illetve kiszolgálóként a biztonságot. Természetesen a Linux méretezhetőségének egyik alapeleme, hogy a rendszermagot az adott kiépítéshez lehet fordítani, de nem hiszem, hogy bármelyikünk szívesen kivárná, amíg egy 75 MHz-es gépen lefordul a rendszermag.
Tekintsük át a szempontokat:

1. A programnak nagyjából mennyi erőforrásra lesz szüksége?
2. Az első kérdésre adott válaszhoz adjuk hozzá annak még legalább az ötven százalékát, bár én a 66 százalékával szeretem megtoldani. Ekkor (ha a gép tudáshatárának a környékén maradunk) szinte biztosak lehetünk egy üzembiztos, elviselhető sebességű működésben.
3. Milyen programok fognak futni ezen a rendszeren?
   Ez legalább olyan fontos, mint az első két kérdés, ugyanis nem tartom túl valószínűnek, hogy ezen a gépen valakinek sikerül úgy elindítania az OpenOffice.org rendszert, hogy azt lehessen rá mondani, hogy használható a sebessége. Ez egyúttal visszautal a gép felhasználási területeinek kérdésére, mivel így jóval kisebb lesz az esélyünk, hogy irodai ügyfélként tudjuk használni. Ez nem jelenti azt, hogy a gép nem jó mondjuk kezdőknek Linuxot tanítani, vagy egyszerű szövegek szerkesztésére, esetleg egyszerű levelezésre, internetezésre.
4. Nagyon fontos, kiszolgáló esetében pedig egyenesen elengedhetetlen tudnunk, hogy mégis milyen idős a rendszer?
   Ez nagyon fontos kérdés, ugyanis minél frissebb egy



rendszer, annál több hiba van benne kijavítva, s annál kisebb az esélyünk arra, hogy egy ismert biztonsági rés veszélyeztesse a hálózatunkat, illetve a gépünket. Feltétlenül nézzük meg, hogy milyen változatszámú a rendszermag, az X-felület, a gcc és hasonlók. (Nekem jelen esetben – az S3 videokártya miatt – nagyon jól jön a 3-as X.)

Ez látszólag csak pár apróság, azonban nagyon meg tudjuk könnyíteni az életünket, ha alaposan végiggondoljuk őket. Nekem tucatszám vannak itthon Linux-változataim, ezek fényében mégis kevésnek éreztem a számosságukat.
Több terjesztést válogattam ki – sajnos a nem kifejezetten kisebb gépekre tervezett terjesztések szinte kivétel nélkül harakirit követtek el, amikor meglátták, hogy mekkora a gép.
A legtöbbjük már az első indulás során sem akart eljutni a telepítőig, ugyanis kevésnek találták a memóriát. Nem egyszer fordult elő, hogy maga a memórialemez (ramdisk), amiben el szeretne terpeszkedni, ugyanakkora volt, mint maga a rendszermemória. *Táblázatba* szedtem, hogy hány Linux-terjesztéssel kísérleteztem, és mi lett az eredménye.
Mint a táblázatból látható, a manapság forgalomban lévő terjesztések nem ilyen kis gépre lettek kitalálva. Ennek ellenére találtam egy tökéletes gyöngyszemet: a Vector Linux 3.x többkevesebb sikerrel mindennek megfelelt. Elvben elindul egy 486-os gépen, de ezt mégsem javasolnám, viszont a vizsgált gép esetében tökéletes.

Mandrake 9.0	Egy ilyen kis gépen már nem működik.
Debian 3.0r1	Éppen nem volt nálam a Debianom.
Caldera OpenLinux	Jó lenne, de túl koros.
RedHat 9.0	Egy ilyen kis gépen nem működik.
IcePack Linux 2.0.1	Még neki is kicsi a gép.
UHU linux 1.0	Egy ilyen kis gépen nem működik.
Vector Linux	Minden feltételnek megfelel.
SuSE 6.x/7.x	Működik, de túl öreg.
SuSE 8.x	Egy ilyen kis gépen nem működik.

Heuréka!

A Vector Linux már 2.4.x-es rendszermagot tartalmaz, valamint 2.95-ös gcc-t, és mindez 16 megabájt memóriára van belőve (optimalized). 350 megabájt merevlemez-terület szükséges neki, és tudja az ADSL kapcsolatot, az `iptables`-e elég fejlett, és reményre ad okot, hogy az X még 3-as (ez kezeli még az S3-as kártyákat), alapértelmezett ablakkezelője pedig az `iceWM`.
Ekkor derült ki a gép további hibája: hiába tartalmazza a BIOS a CD-ROM-ról a rendszerindítást, mégsem képes végrehajtani. Mást nem lehetett tenni, elkészítettem a telepítő CD-n megtalálható két hajlékonylemez lemezképe alapján a Vector Linux indító- és rootlemezét. Erről indítva a gépet hozzá is láthatunk a telepítéshez. Ekkor egy régi VGA képernyő fogad bennünket. Sajnos egy kicsit együgyű a telepítő, tehát nem enged szabadon garázdálkodni, például feltétlenül ragaszkodik a merevlemez ellenőrzéséhez.
Azoknak, akik egyébként nem szoktak hozzá a régi telepítőkhöz (én már el is szoktam tőlük) mindenképpen javaslom, hogy ne egyedül álljanak neki. Viszont miután ezen a kis nehézségen túljutottunk, szinte azonnal felmászik az alaprendszer. Bámulatos módon valóban 350 MB merevlemez-területre van szüksége. Érdekes viszont, hogy az alaprendszer telepítése és az MBR-telepítés (LILO-t használ és telepít) után teszi csak fel az X-felületet, és ezután a rendszermagot. A telepítés befejeztével egyébként egy teljesen használható rendszert kapunk, ami megfelelhet az alapvető igényeknek. A grafikus felületet meglepő módon nemcsak a jól ismert XF86Configgal, hanem a SuSE 6.x/7.x-ből ismert `sax`-szal is be lehet állítani (messze jobbnak és szebbnek tartom, mint az XF86Configot).
Ha mindez megvan, akkor érdekes módon egy emberi sebességű, de teljesen jól futó rendszert kapunk. Alap és ablakkezelője az IceWM, de van WindowMaker is. A XDM az alapértelmezett beléptetőfelület, de csak miután a */etc/init.d* könyvtár parancsfájljaiban engedélyezzük, ugyanis az induláskor a karakteres felület indul el, és csak rendszergazdaként lehet elindítani a grafikus felületet (`startx` parancs).
Az `IceWM` témája vicces módon a Windows XP-t idézi, de számos egyéb témát is le lehet hozzá tölteni. Én az utóbbit követtem el.
Ekkor jön még csak a neheze, ugyanis még semmi hasznosra nem használható a gép, mindössze működik. A kikapcsolás után beleszereltem egy rtl8139-es és egy rtl8029-es hálózati kártyát. A gépet bekapcsolva örömmel láttam, hogy az indulás során betölti a kártyákhoz a megfelelő modulokat. Egyébként szándékosan Realtek kártyákat raktam bele, mivel tudtommal ezek szinte minden rendszerrel együttműködnek.

Irány a hálózat!

A Vector Linux alatt a telepítőt a `vasm`-mel lehet meghívni. Ez azonban már csak alapszintű rendszerfelügyeletre használható

– természetesen minket ezúttal a hálózat érdekel. Állítsuk be az eth0 és eth1-es kártyákat. A Vector nekem ugyan nem ajánlotta fel a pppoe kapcsolódási módot (ADSL kapcsolathoz nélkülözhetetlen), ennek ellenére megadtam neki a DHCP-t, illetve a DNS-t, és a bejelentkezéshez szükséges adatokat. Az eth1-nek viszont érdemes IP-t adni, hisz ő fogja kiszolgálni a belső hálózatot. Meglepődésemre – habár sehol sem találtam a pppoe megadásának a helyét – könnyedén működött az ADSL. Ahhoz, hogy a belső hálózatról is el lehessen érni az internetet, igazán nem sok dolog kell. A Vector Linux alatt ismerjük az eth1 IP-címét és a netmaszk (255.255.255.0) is adja magát. A másik gép az, aminél vigyázni kell az adatok megadásával.
Az ügyfélgépben a hálózati kártya IP-címe lehet egyazon tartományban, mint a kiszolgálóé, de ez nem javasolt. Ha lehet, válasszunk másik tartományból IP-címet, adjuk meg a kiszolgáló esetében is használt két DNS-t, illetve a netmaszk is azonos. Egy dologra kell csak ügyelni, ez pedig az átjáró (gateway). Az ügyfélgép számára az átjáró a Vector linuxos kiszolgálógép eth1-es kártya IP-címe; és arra is érdemes odafigyelni, hogy a legtöbb Linux számára nem az az eth0, amelyiket először a gépbe szereltük, hanem mindig a legfölső sínben elhelyezkedő hálózati kártya. (Sajnos akadnak kivételek, de a legtöbb Linux-terjesztés ez alapján sorolja be a kártyákat.)
Ha ez is megvan, akkor gyakorlatilag semmi más dolgunk nincsen, mint arra utasítani az iptables-t, hogy ossza meg az internetet. Nyissunk meg egy szövegszerkesztőt. Ez bármilyen lehet, akár a legegyszerűbb is, és a következő utasításokat írjuk bele:

```
iptables -A FORWARD -i eth0 -j ACCEPT
iptables -t nat -A POSTROUTING -o ppp0
➥-j MASQUERADE
echo 1 > /proc/sys/net/ipv4/ip_forward
```

Az első sornál az eth0 adatait fogadjuk, és engedélyezzük a továbbításukat. A második sorban, a belső hálózatot kiengedjük a ppp0-n keresztül a világba, az útirányító táblázatban kapjanak helyet a "ppp0" adsl interface adatai (figyelem, az ADSL hálózati csatolófelülete mindig a ppp0, nem keverendő össze az eth0-val, ami a fizikailag létező hálózati kártya eszközleírója!), illetve ezt maszkolja is. Ennek az lesz a hatása, hogy a kiszolgáló „mögött” dolgozó gép minden hálózati adatforgalmát úgy fogják látni kívülről, mintha a kiszolgálógép csinálná. Tehát kimaszkoltuk, ezáltal eltüntettük az ügyfélgépet. Ezt ebben a formában célszerű egy parancsfájlban elhelyezni, és a szöveges fájlnak ne adjunk kiterjesztést, mert felesleges. Mindezek után semmi akadálya az ügyfélgépről az internet elérésének. Célszerű, ha a kipróbálásnál a kapcsolat felépítésének minden szakaszában folyamatosan ellenőrizzük magunkat. Először a kártyákat pingeljük, azután célszerű az egyik gépről a másikat pingelni. Legvégül pedig, amikor úgy gondoljuk, hogy mindent jól állítottunk be, az ügyfélgépről pingeljünk webcímet, hogy lássuk, valóban jó-e a kapcsolat.
A további beállítások igazából már csak finomítások. Így például a mentett parancsfájlra lehet egy indító parancsfájlban hivatkozni (a */etc/init.d/*-ben találhatóak), így minden indításkor elindul a megosztás, illetve a véletlen „leszakadásra” is lehet parancsfájlt írni, vagy akár a fenti három sort a végtelenségig finomítani. Eme alaplépések után egy működő hálózaton szinte már mindent meg tudunk tenni, habár erre a kis géphre adatbázis-kezelő rendszert nem tanácsos feltenni.
Természetesen célszerű ellenőrizni, hogy éppen milyen nyitott kapuk vannak a kiszolgálón (nmap parancs), és ennek megfelelően szükséges tovább finomítani a beállításokat. Az iptables beállításait mindenképpen érdemes testreszabni, ugyanis nem árt, ha a belső hálózat védelmére is figyelmet fordítunk.

Végszó

A fentebb leírtak jól tükrözik, hogy kevéske képzelőerővel és egy kis erőfeszítéssel mindenféle bonyolult trükközés nélkül sikeresen beállítottunk egy olyan számítógépet kis hálózati kiszolgálónak, amit rajtunk kívül talán mindenki kidobott volna. Elmondható, hogy hiába mindössze 75 MHz-es a processzor és 16 megabájt a memória, ebből semmit nem látunk a hálózat használata során. A fent említett gép a leírt operációs rendszerrel alkalmas volt arra, hogy három gépet szolgáljon ki és védjen. Az én véleményem az, hogy egy 3–5 gépes otthoni hálózatot jó sebességgel, teljesen megfelelően ki tud szolgálni, és mivel régi gép, a fogyasztása is kicsi, tehát a családi költségvetés sem érzi meg folyamatos használatát.
Jó munkát és kellemes időtöltést kívánok a Linuxszal!

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.

KAPCSOLÓDÓ CÍM

Vector Linux ➲ http://www.vectorlinux.com

Internetes rádióállomást mindenkinek! (2. rész)

A SHOUTcast webes felülete és a kiszolgáló különleges beállításai.

Sorozatunk előző részében eljutottunk odáig, hogy az alapvető tulajdonságok beállítása után működésre bírtuk a kiszolgálót, elláttuk egy bemenő adatfolyamforrással, s ezek után kapcsolódva a kiszolgálóhoz hallhatóvá vált a számítógépünkből kiáramló zenefolyam. Akár azt is mondhatnánk, hogy készen vagyunk, ám ebben az esetben jogosan illethetnének bennünket azzal váddal, hogy a kihívások terén hamar megelégszünk az elért eredményekkel. Annak érdekében, hogy e vádakat elkerüljük, s emellett még egy valóban minden tekintetben tökéletes összeállításhoz juthassunk, ebben a cikkben néhány szót szeretnék szólni adatfolyam-kiszolgálónk különleges testreszabási lehetőségeiről, valamint a különleges szolgáltatásként nyújtott webes felület kezeléséről.

A mindenre odafigyelő, százszemű SHOUTcast

Indiánnyelven talán valami ehhez hasonló névvel illethetnénk kiszolgálónkat, ugyanis ha valaki már belenézett a beállításfájlba az előző cikkben írt utasítások alapján, akkor láthatta, hogy mennyi mindent be lehet benne állítani a hallgatóság lélekszámától kezdve a nem kívánatos ügyfelek tiltási idejének lejártáig. Ezek a beállítások az otthoni vidám rádiózgatás eléréséhez természetesen nem fontosak, az alapértelmezett értékek tökéletesen megfelelnek, ám ha valaki az előző rész nyomán próbálná meg megvalósítani az ott említett kollégiumi rádióállomást, annak bizony szüksége lehet néhány különleges érték átállítására, hogy a rendeltetésszerű szolgáltatást senki meg ne zavarja, és a hallgatóság a lehetőségekhez mérten minél magasabb szintű műsorélményt tudhasson magáénak. A beállítási lehetőségeket a könnyebb kézben tarthatóság érdekében célszerű kategóriákba sorolni, ahogyan ezt a program írói meg is tették, és a beállításfájlban közreadták. A főbb csoportok az alábbiak: alapvető beállítások, naplózási tulajdonságok, hálózati beállítás, kiszolgálóbeállítások, hozzáférések kezelése, összevont többes beállítások, egyéb trükkök.

Alapvető beállítások

E csoport beállítási lehetőségeit előző írásunkban már végignéztük, ide tartoznak ugyanis azok a tulajdonságok, amelyek megadása feltétlenül szükséges ahhoz, hogy egyáltalán futtatni tudjuk a kiszolgálónkat. Mint ahogyan azt is leírtuk, hogy itt található a csatlakozási jelszó, a kiszolgálókapu és a hallgatóság legnagyobb létszámának a meghatározása.

Naplózási tulajdonságok

Nagy rendszerek esetén bizony feltétlenül szükségünk lehet a működés időben változó részleteire is. Terjedelmes hallgatóság esetében ugyanis igen fontos, hogy ki mennyi sávot használ, mekkora a vonalunk terheltsége, s egyáltalán: szeretnénk tudni, hogyan állunk a hallgatósággal, mert ennek függvényében megváltoztathatjuk a műsor tartalmát. Mindezek mellett erőforrást takaríthatunk meg, ha például egy háttérben futó kiszolgálónál kikapcsoljuk az amúgy sem látható konzolra történő naplózást. E nehézségek kezelését oldhatjuk meg a csoport különböző beállításaival. Nézzünk meg ezekből párat!

LogFile: itt állíthatjuk be a naplófájlunk helyét és nevét. Ha egyáltalán nem szeretnénk a naplózást, az adatokat irányítsuk a */dev/null*-ba.
RealTime: amennyiben ennek az értéke `1`, a kiszolgáló konzolján minden másodpercben frissül a SHOUTcast pillanatnyi állapota, ahogyan azt már az első indítás során is megfigyeltük.

```
<11/24/03@10:39:23> [sleeping] 0 listeners
➥(0 unique)
```

Screenlog: ennek az értéknek a nullára állításával kapcsolhatjuk ki a konzolra történő naplózást.
W3CEnable: bekapcsolható vele a W3C (www compliant) naplózás, amelynek segítségével a webkiszolgálóknál szokásos naplót kaphatjuk eredményül. Ennek az az előnye, hogy igen részletes kimutatást kaphatunk akár minden egyes ügyfélről. Rögzíti az összes hallgató által lejátszott összes számhoz tartozó adatokat az elküldött adatmennyiséggel együtt. Ez természetesen önmagában egy kezelhetetlen adatmassza lenne, amellyel egyáltalán nem kerülnénk közelebb az általunk áhított adatokhoz. Éppen ezért ez a napló olyan formátumú, hogy a HTTP-kiszolgálók naplójához illeszkedik, ezáltal lehetőségünk nyílik az ott használt statisztikakészítő programok használatára, amelynek segítségével így a naplófájlokat webes, grafikonokkal teli formában nézhetjük meg. A tárgyalt naplófájl nevét a `W3CLog` értékeként állíthatjuk be.

Hálózati beállítások

E beállítások megváltoztatására többnyire akkor lehet szükségünk, ha egyszerre több kiszolgálót is üzemeltetünk, vagy kiszolgálónkat épp továbbító üzemmódban használjuk. A részletekért vizsgáljuk meg inkább az egyes testreszabási lehetőségeket!

SrcIP: itt adhatjuk meg, hogy melyik IP-címről kapcsolódhat az adatfolyam a kiszolgálónkhoz. Alapértelmezett értéke: `ANY`. Ez azt jelenti, hogy bárhonnan kapcsolódva sugározható adatfolyam a kiszolgálónkon keresztül. Ezzel csak az a gond, hogy így bárki, aki ismeri a jelszót, kapcsolódhat hozzánk. Ezt úgy akadályozhatjuk meg a legjobban, hogy megnevezzük azt az egy gépet, amelyet engedélyezni szeretnénk. A legtöbb esetben ez mi magunk vagyunk; ekkor állítsuk ezt az értéket 127.0.0.1-re, amely a helyi visszacsatolást címezi, így csak a saját masinánkról tudunk adatfolyamot szolgáltatni a kiszolgálónknak.
DestIP: ez az az IP-cím, amelyen a kiszolgáló figyeli a bejövő kapcsolatokat. Ha csak egyetlen IP-címünk van, a beállítási lehetőségnek nincs túl sok jelentősége. Akkor válhat fontossá,

ha a gépünkben több hálózati eszköz is helyet kap, s esetleg több kiszolgálót is futtatunk egy időben, így az egyes kiszolgálópéldányok esetében meg kell adnunk, hogy melyikhez melyik IP-cím tartozik. Alapértelmezett értéke szintén `ANY`, ne is változtassunk ezen, ha hagyományos összeállítás áll rendelkezésünkre.
`RelayPort, RelayServer`: ezek a beállítások a beállításfájlban megjegyzésbe vannak téve. Csak akkor szabad használni,

1. kép A VIP ügyfelek listája – akik mindig csatlakozhatnak

2. kép A kitiltottak listája

ha a programunk továbbító üzemmódban dolgozik (megfelel egy hagyományos átjátszóállomásnak). A beállított értékeknek megfelelően ügyfélként csatlakozik a célszámítógéphez, és az onnét letöltött adatfolyamot továbbítja a saját hallgatói számára.

Kiszolgálóbeállítások

Az itt található beállítások akár az „egyéb” kategóriát is erősíthetnék, hiszen a segítségükkel a kiszolgáló által nyújtott szolgáltatás egyes minőségi jellemzőit állíthatjuk be. Ilyen például az, hogy adatfolyam-kiesés esetén mennyi idő elteltével rúgja ki a kapcsolódott hallgatókat, vagy az, hogy milyen átmeneti zeneszámot játsszon be erre az időre.

`AdminPassword`: segítségével a webes ügyfél használatát szigoríthatjuk. Abban az esetben ugyanis, amikor ez az érték nincs megjegyzésbe téve (comment), a webes felületre történő hagyományos bejelentkezés „felhasználói” módba fokoz vissza bennünket. Mint emlékezhetünk rá, `admin` felhasználóként a kiszolgáló alapbeállításainál megadott jelszóval jelentkeztünk be, s ha csak ez az érték lett a beállításfájlban megadva, kapásból rendszergazdaként azonosít bennünket a rendszer. Ellenkező esetben az ott megadott jelszóval történő azonosítás csupán nézegetőmódot tesz lehetővé a webes felületen: nem tudunk felhasználókat kirúgni, a tiltólistát szerkeszteni stb. Ha mégis ezt szeretnénk, akkor az `AdminPassword` értéknél megadott jelszót kell beírni. Ezzel néhány kiváltságos személy számára lehetővé tehetjük, hogy mélyebben betekinthessenek a rádióállomás életébe, ugyanakkor megakadályozhatjuk, hogy belekontárkodjanak a mi kis „életművünkbe”.
`AutoDumpUsers`: amennyiben a változó értéke `1`, a kiszolgáló a kapcsolódott hallgatókat önműködően leválasztja, ha a bemeneti adatfolyam valamilyen okból megszakad.
`AutoDumpSourceTime`: annak a tétlen időnek a másodpercben mért értéke, amennyi alatt a kiszolgáló nem választja le a bemeneti adatfolyamot, ha valamiért elakad (nem jön bemenő jel a kiszolgálónak).
`IntroFile`: annak az MP3-fájlnak az elérési útja, amelyet a kiszolgáló önműködően lejátszik, amikor egy hallgató csatlakozik hozzá. Ilyen lehet például a rádióállomás szignálja. Ha az érték nincs beállítva, akkor csatlakozáskor természetesen nem kapunk egy ilyen kis bejátszást.
`BackupFile`: annak az MP3-fájlnak az elérési útja, amit a kiszolgáló önműködően elindít, és folyamatosan játszik, ha megszakad a bemeneti adatfolyam. Ehhez természetesen az `AutoDumpUsers` értéket `0`-ra kell állítani. Fontos még, hogy e vészfájl értékeinek meg kell egyeznie az adatfolyam-kiszolgáló által küldött MP3-folyam értékeivel (`Samplerate/Chanells`).
`TitleFormat`: segítségével megadható, hogy a csatlakozó hallgató lejátszóprogramjában az általunk kívánt kiírás jelenjen meg címként.
`URLFormat`: hasonló a `TitleFormat` beállításhoz, de itt az ügyfélhez küldött weboldal címét állíthatjuk be, ahol a hallgató a rádióállomással kapcsolatos adatokat találja.
`PublicServer`: ennek segítségével oldhatjuk meg, hogy a rádióállomásunk ne legyen nyilvános (public), ezáltal nem is érhető el a SHOUTcast weboldalán a rádióállomások listájában. Három értéket vehet fel: `always`, `never`, `defalut`. Állítsuk be a `never` értéket, ha azt szeretnénk, hogy a kiszolgálónk címe ne legyen nyilvános. Abban az esetben is használjuk ezt, ha nem rendelkezünk internetkapcsolattal, és a programunk folyamatosan dobálja a hibaüzenetet, miszerint nem tud csatlakozni az ➲ http://yp.shoutcast.com címre, amelynél a rádiónkat bejelegyezhetné a „központi telefonkönyvbe”.
Ezen az ➲ http://yp.shoutcast.com címen elérhető szolgáltatás gyűjti ugyanis a világon futó rádióállomások címeit, hogy azonnal közzé is tegye őket a weben, így egy mindig időszerű állomáslistát mutat az odalátogató rádiózni vágyóknak.
A változó alapértelmezett értéke `default`, ami azt jelenti, hogy az adatfolyam-kiszolgálónál beállított értékek befolyásolják, hogy nyilvános-e vagy sem. Az `always` és `never` értékek ezt felülírják: vagy mindig, vagy soha nem küldik el a nyilvános adatokat a katalóguskiszolgálónak.
`AllowRelay, AllowPublicRelay`: mindkét beállítási lehetőség arra vonatkozik, hogy megengedjük-e másoknak, hogy továbbítsák a mi adatfolyamunkat, azzal a különbséggel, hogy az `AllowPublicRelay` lehetőség csak a nyilvános listán történő szerepeltetést tiltja le, azt nem, hogy az általunk sugárzott adatfolyamot mások továbbíthassák.
`MetaInterval`: ez egy érdekes szabályozási lehetőség. Beállíthatjuk, hogy mennyi adatmennyiség után küldje újra az úgynevezett metaadatokat, azaz milyen időközönként jusson hozzá a hallgató lejátszóprogramja az adatfolyam egyéb tulajdonságaihoz (a címéhez stb.). A leírásban azt mondják, hogy ezt nem érdemes piszkálnunk, de talán mégis érdekes lehet, hogy még ilyen apróságokat is kézzel beállíthatunk.

Hozzáférések szabályozása

Itt található néhány olyan szabály, amelyeket kényes esetben alkalmaz a kiszolgáló.

ListenerTimer: egy hallgató által megszakítás nélkül történő legnagyobb kapcsolódási idő percben kifejezve. Amennyiben ezt túllépi, a kiszolgáló önműködően leválasztja az adásról.
RipFile: annak a fájlnak a neve, amiben azon ügyfelek IP-címe található, akik számára a csatlakozás mindig engedélyezve van. Különösen akkor kerül előtérbe, ha a megengedett hallgatók száma eléri a határt, s valaki utánuk szeretne kapcsolódni. Ha benne van a `RipFile`-ban, akkor a kiszolgáló kirúgja a legrégebben kapcsolódott felhasználót, s annak helyére kerül a most csatlakozni kívánó VIP-hallgató. Ez abban az esetben lehet hasznos, ha néhány továbbító üzemmódban dolgozó kiszolgálónak szeretnénk csatlakozást biztosítani, de a sávszélességünk csak korlátos számú csatlakozást engedélyez. A fájl tartalma egyébként webről kezelhető, kézzel inkább nyúljunk hozzá!
RipOnly: ha ennek a változónak az értéke ***Yes***, úgy csak azok az ügyfelek kapcsolódhatnak, akik az előbb említett `RipFile`-ban benne vannak, számukra ugyanis helyeket tart fenn a kiszolgáló. Talán akkor hasznos, ha egy tisztán továbbítókiszolgálóknak szánt adatforrás vagyunk, és azt szeretnénk, hogy minden esetben csak egy meghatározott ügyfélkör csatlakozhasson hozzánk.

Összevont többes beállítások

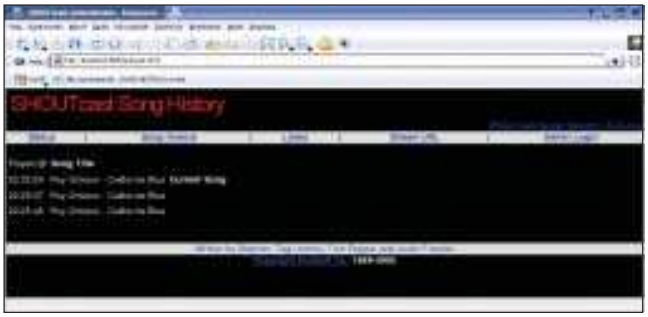
A beállítási csoport használata akkor lehet indokolt, ha az adott gépen több kiszolgáló is fut, s ezek beállításai nagymértékben megegyeznek, leszámítva egy-két egyedi értéket. Ez esetben munkás lenne minden egyes kiszolgálót mindig beállítani, erre a gondra kínál megoldást a következő két kapcsoló.
Unique: egy változót határozhatunk meg vele, amit utána minden olyan beállítási lehetőségnél felhasználhatunk, amely egy fájlra mutat. Miután megadtuk (declarated) a változót, az egyes elemeknél a $, dollárkarakterrel helyettesíthetjük az értékét. Mivel ez első olvasatra meglehetősen zavarosnak tűnhet, álljon itt egy példa. Megadjuk az `Unique` értékét a beállításfájl legelején: `elso_kiszolgalo`. Ezután a változékony fájlneveket (például a naplófájl neve) helyettesítsük $ dollárkarakterrel, valahogy ilyen módon:

```
Log=/usr/local/shoutcast/$.log
```

Ekkor az adott könyvtár ***elso_kiszolgalo.log*** nevű állományába történik a lapozás. Ha ekkor a kiszolgáló beállításait egy másik SHOUTcast-példányhoz is átmásoljuk, elegendő csupán az elején az `Unique` változó értékét átírni, s további módosításokra nem lesz szükségünk.

Include: az itt megadott fájl tartalmát a kiszolgáló mint beállításfájlt beolvassa, és az ott található összes értéket felhasználja (a programozási nyelveknél is megszokott `include` művelet mintájára). Ha a későbbiekben az `Include` érték után újra meghatározunk egy adott értéket, amely a beállítási fájlban is szerepel, akkor az felülírja az ott található értéket, és a dolog természetesen fordítva működik, ha az `Include` érték a beállítás után van csak megadva.
Ezzel a módszerrel az is elérhető, hogy a közös adatokat egyetlen nagy közös beállításfájlban tároljuk, s ezt a kiszolgálóra jellemző beállításfájl elején beolvastatjuk, majd a hiányzó értékeket az adott fájl végén pontosítjuk, esetleg egy már meglévő értéket felülírunk.

3. kép A SHOUTcast-kiszolgáló webfelületének nyitóoldala



4. kép A legutóbbi pár lejátszott szám adatai

Egyéb trükkös beállítások

Innen csak egy érdekes lehetőségre szeretném felhívni a figyelmet. Lehetőség nyílik a processzorok számának a beállítására is. Alapesetben természetesen felismeri, hogy hány CPU van a gépben, és az egyes kapcsolatokhoz tartozó szálakat egyenletesen elosztja a processzorok között. Ha azonban azt szeretnénk, hogy a kiszolgáló ne az összes processzort használja, akkor megadhatjuk, hogy hány darab között ossza szét a terhelést, s ez a lehetőség még akkor is hasznos lehet, ha valamiért nem jól ismerné fel a SHOUTcast, hogy hány processzorunk van, de ez már szinte elméleti szintű beállítás.
A fentebb részletezett rengeteg lehetőség természetesen nem az összes, az talán egy fél újságba sem fért volna bele. Azokat próbáltam kiválasztani, amelyek egyrészt fontosak, másrészt rávilágítanak a program különleges képességeire. Ezenkívül kimaradt még néhány apróság, ami érdekes és fontos is, de a webfelületről kezelhető, tehát az ott leírtakban mindenképp tárgyalni fogjuk. Ezek után lássuk a kiszolgáló által nyújtott, böngészőből távolról is elérhető webes felületet!

A SHOUTcast webes lehetőségei

Arról már a legutóbbi cikkben is esett szó, hogy mivel úgyis egy HTTP-kiszolgáló a rádióállomásunk alapja, a fejlesztők ellátták egy pluszszolgáltatással: egy internetböngésző segítségével is kapcsolódhatunk hozzá. Ennek az az előnye, hogy a hallgatóságot mindenféle földi jóval kényeztethetjük, ami a rendelkezésükre álló adathalmazt illeti, s ezen felül megkönnyíthetjük futó kiszolgálónk karbantartását, felügyeletét egy ilyen grafikusnak mondható felületen keresztül. A felület a ➲ ***http://kiszolgálónév:kiszolgálókapu*** címen érhető el. A tájé-

koztatóoldalhoz semmilyen bejelentkezés nem szükséges, ezt bármelyik hallgató megtekintheti.

Az állapot menüpont

Az oldal letöltése után kiszolgálónk állapota látszik. Megnézhetjük a sugárzott adatfolyam tulajdonságait, a címét, az adatfolyamhoz kapcsolódó weboldal címét, a hallgatók számát, az épp most játszott szám címét és sok egyebet is.

A lejátszott számok listája

A következő menüpont kiválasztásával (***song history***) az utolsó néhány zeneszám címét és előadóját nézhetjük meg időben csökkenő sorrendben, a kezdési időponttal feltüntetve. Befolyásolni tudjuk, hogy összesen hány tétel jelenjen meg. A kiszolgáló beállításai között keressük meg a ***ShowLastSongs*** bejegyzést, és az értékénél állítsuk be, hogy visszamenőleg hány címet írjon ki az ide látogatóknak.

Hallgass ide!

A menüben eggyel balra található ***listen*** menüpont arra szolgál, hogy rákattintva XMMS vagy Winamp ügyfelünkben megnyissa az adott rádióállomás URL-jét. Ez csupán egy kényelmi szolgáltatás, hogy az URL hozzáadásánál ne kézzel kelljen a lejátszóprogramból begépelgetni a címet.

Nézzük, mit kínál a mostani adás!

Ha tovább lépünk a menüben, a ***Stream URL*** menüpontra látogatva megtekinthetjük az adatfolyam-kiszolgáló által beállított URL-t, amely az adatfolyammal kapcsolatos hallgatói adatokat tartalmazza.

Admin login

Ide kattintva egy bejelentkező képernyő pattan fel, ahol azonosítanunk kell magunkat. Jelentkezzünk be ***admin*** felhasználóként, majd írjuk be a jelszót. Ha beállítottuk a korábban tárgyalt `AdminPassword` értéket, akkor azt írjuk ide, ellenkező esetben a beállítási fájl elején használt `Password` mező értékét használjuk. Ha mindent jól gépeltünk be, a felügyeleti felületen találjuk magunkat.

A rendszergazdai honlap

Hallgatók

Miután elénk tárult a bejelentkezés utáni képernyő egyrészt címük szerint láthatjuk a csatlakozott hallgatók listáját, és fel van még tüntetve a kapcsolódási időtartam, az, hogy hányszor ürült ki hálózati hiba miatt az átmeneti táruk. Másrészt módunkban áll kirúgni, kitiltani őket, kitiltani a teljes alhálózatot, ahonnan csatlakozhatnak vagy épp felvehetjük őket a VIP-hallgatók listájára. Ennek hatására bekerülnek a már emlegetett `RipFile`-ba, így minden esetben csatlakozni tudnak. A lista alatt ott található a hallgatók által is elérhető nyitóoldal tartalma az épp sugárzott adatfolyam tulajdonságaival.

A napló utolsó néhány bejegyzése

A menü második eleme a ***tail logfile*** menüpont, amely arra szolgál, hogy megjelenítse a naplófájl utolsó bejegyzéseit, s ezáltal nem az egészet kell böngésznünk.

A teljes naplófájl

Ha mégis a teljes állományra szükségünk volna, a ***view logfile*** menüpont segítségével megnézhetjük. Ez akkor lehet érdekes, ha kíváncsiak vagyunk a működés részleteire vagy valamilyen rendellenesség okát keressük.

Tiltólista

Már beszéltem róla, hogy lehetőségünk van kirúgni a felhasználókat, ha valamilyen okból (lehetőleg nem személyes) kifolyólag megsértenek egy-egy működési szabályzatot, vagy nem rendeltetésszerűen veszik igénybe a szolgáltatást. Ha az ilyen hallgatókat ártó szándék vezérli, az bizony vissza fog térni a következő kattintással, mi meg rúghatjuk ki újra. Ennek elkerülése végett az ilyen szemtelen felhasználókat tiltólistára tehetjük, az ott szereplő IP-címről csatlakozó hallgatót ugyanis a rendszer nem fogja beengedni. Ennek neve ***ban list***, ezt a menüpontot választva kezdetben egy üres listát kapunk, alatta két űrlappal. A felső használatával lehetőségünk van egy IP-cím megadására, amely a tiltás után már nem kapcsolódhat újra, ám ha jelenleg is aktív, nem rúgja ki a rendszer, így amíg kapcsolata meg nem szakad, továbbra is a hallgatónk marad. Az alsó űrlap segítségével egész alhálózatokat tilthatunk ki, ha egy ügyfél mondjuk DHCP segítségével folyton újracsatlakozva változtatgatná az IP-címét.
A tiltólista minden bejegyzése mellett található egy ***unban*** hivatkozás, ennek segítségével távolíthatjuk el az illetőt erről a tiltólistáról.

Foglalt helyek listája

A beállítási lehetőségeknél már megemlítettük a foglalt helyek jelentőségét, s azt is szóba hoztam, hogy a lista webes felületről szerkeszthető. A ***reserve ip list*** hivatkozásra kattintva egy, az előzőhöz hasonló listát kapunk, ahol az alsó űrlap segítségével adhatunk hozzá újabb elemeket, s a felsorolt fenntartott IP-címek a mellettük található ***clear entry*** hivatkozással távolíthatók el a listából.

Kijelentkezés

Érdemes igénybe venni az utolsó menüpontot abban az esetben, ha nyilvános helyről látogattuk meg az oldalt, mivel egészen addig hozzáférünk a rendszergazdai módhoz, amíg az összes böngészőpéldányt be nem zártuk.

Zárszó

Azt hiszem, ezzel be is fejeződött ennek a valójában egyszerűnek mondható, kis 200 kilobájt körüli méretű programocskának a bemutatása. Talán egy kicsit „velős” így egyszerre ez a sok beállítási lehetőség, én azonban mégis örülni szoktam nekik, ugyanis így mindent könnyedén testreszabhatok, nem érzem megkötve a kezem.
A SHOUTcast kiszolgáló egyébként is a jól megírt programok közé tartozik, mivel ha nem akarjuk, nem kell agyonállítgatnunk: az alapértelmezett beállításokkal, ha átlagos feladatokra használjuk, az esetek többségében helyesen működik. Elképzelhetjük, hogy a beállítási lehetőségek kombinálásával működési módok ezreit állíthatjuk elő, s az is látható, hogy igazán komoly, akár több tíz vagy száz gépből álló rádióállomás-rendszer is kiépíthető mindenféle kiegészítők használata nélkül. Az egyetlen dolgunk, hogy kellően átgondoljuk, s jól felépítsük a hálózatunkat, a többit elvégzi a SHOUTcast.

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

Debian otthonra (2. rész)

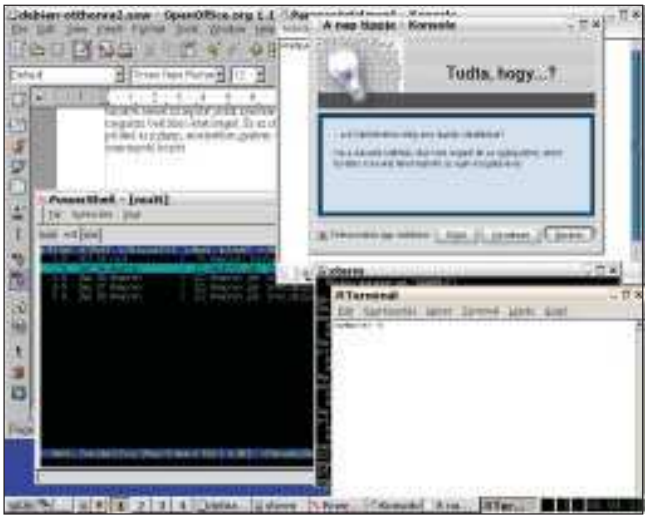
Végezzünk el még néhány alapvető beállítást!

Nos, gondolom, hogy azok közül, akik velem tartottak az elmúlt hónapban, kevesen bírták ki, hogy ne telepítgessenek mindenféle csomagokat a gépükre, és nagy valószínűséggel eljutottak idegeik vagy csomagfüggőségeik teljes összekeveredéséhez. Aki megélte már azt a pillanatot, amikor úgy dönt, hogy nincs mit veszítsen, letöröl minden szükségtelen csomagot, majd tiszta lappal kezd, és ezzel a felkiáltással el is röppent az egész éjszaka, annak gratulálok, a legjobb úton halad! Hiszen a filozófiai rendszerekből tudjuk, hogy az elméleti tudás rendkívül fontos, de fabatkát sem ér, ha azt nem támasztja alá gyakorlat. Illetve mintha olvastam volna valami ilyet egy szexuálterapeuta írásai között is…

De ne rohanjunk előre. Ha az előző cikk óta nem frissítetted a rendszeredet, akkor most mindenképpen érdemes megtenned. Egyszerű a történet: `apt-get update`, ezzel frissíted az elérhető csomagok listáját, majd add ki az `apt-get upgrade` utasítást, hogy le is töltsd azokat, amelyek a rendszer szerint neked kellenek. Az `upgrade` parancsra nagy valószínűséggel kapsz egy hosszú listát a frissítendő csomagokról („…will be upgraded”), rosszabb esetben egy hasonlóan hosszút azokról, amelyeket a rendszer nem frissít magától („…have been kept back”). A visszatartott csomagokat is frissíteni akarjuk, ezért aztán, miután jól végigfutott az `apt-get`, indítsd el az Aptitude-ot, és vele is végeztesd el a frissítést (`u`, majd kétszer `g`). Amivel többet végez az Aptitude (nagy vonalakban), az az, hogy a „menet közben szükségessé vált” csomagokat magától felrakja. Ezeket az első `g` utáni képernyőn zöld háttérrel sorolja fel a program. Mindkét frissítés a már ismert módon halad: a program letöltögeti a csomagokat, kibontja, majd egyesével telepíti őket.

Egy gyors megjegyzés. Előfordul, hogy csak a fontos csomagokat szeretnéd frissíteni, és jelenleg épp nem vagy vevő rá, hogy amúgy sem túl remek netkapcsolatodat azzal fáraszd, hogy az ***openoffice.org-bin*** csomag 1.1.0-s változatáról átállj az 1.1.0-2-re, mindezt röpke negyvenegynéhány mega letöltéséért cserébe. Ha ez a helyzet, az Aptitude-ban az első `g` után keresd ki azt a csomagot, amit nem szeretnél letölteni, majd nyomj rajta egy egyenlőségjelet (`=`). De ha már így előreszaladtunk, és tényleg van a gépen OpenOffice.org, felhívom a figyelmeteket a már említett, binárist telepítő csomaghoz fűzött megjegyzésre: ha frissíted a program által használt valamelyik könyvtárat (xlibs, libc6, libstdc++5, libgcc1, libfreetype6, libart-2.0-2, libfontconfig1, libxaw7), utána futtasd le a `/usr/sbin/oooprelink` programot.

De ismét csak ne rohanjunk úgy előre! Először is, van-e egyáltalán grafikus felület a gépen? Az X beállítását már többször megrágtuk az újságban (Linuxvilág 2003. április: XF86Config-4, 2002. június: Grafikus meghajtók Linux alatt, 2002. október: Egy másik környezet). Ma már a rendszer szerencsére nagyon jó eséllyel kitalálja nekünk, hogy milyen képességekkel rendelkező videokártyánk van, és a fontosabb alapbeállításokat el is végzi. Gond akkor adódik, ha abban a szerencsés helyzetben vagy, hogy egy újabb videokártyával bírsz, például egy új laptopba épített ATI Radeon IGP 340M (kis Compaqomhoz ilyen tartozik).



1. kép Terminálablakok garmadája

Ehhez a kártyához a rendszer telepítésekor nem találtam használható meghajtót, az X pedig nagy bátran a Vesa meghajtót ajánlja. Szerencsére nem CRT-vel szerelt laptopom van, valamint a legvadabb játék, amit mostanában játszom, a webszerkesztő program, így a 60 Hz-es frissítés nekem elfogadható.

Jó, jó, elismerem, ez egy eléggé „savanyú a szőlő” megoldás. Az igazat megvallva az Xi Graphics Inc. május közepén közhírré tette, hogy kereskedelmi termékként forgalomba hozott egy remek meghajtót ehhez a videolapkához (➲ http://www.xig.com/Pages/PrReleases/PRMay03ATI-IGP-340M.pdf), a baj csak az, hogy nem szívesen fizet érte az ember. Na, azért nem olyan rossz a helyzet, a happypenguin.org oldalon (➲ http://www.happypenguin.org/forums/viewtopic.php?t=835) például valaki a 320M-essel szerelt gépével 480 fps-t mért 1024×768-ban (Warcraft3). Ehhez csupán újra kellett fordítsa az X-et és a rendszermagot a megfelelő foltokkal. Ehhem, a kiegészítések fejlesztését a bugzillán követik (➲ http://bugs.xfree86.org/show_bug.cgi?id=314), már a 4.4.0-s X-hez is készült, a baj csak az, hogy Debian alatt – véleményem szerint – az X házi fordítása és telepítése pont annyira egyszerű, mint felprogramozni egy kávéfőzőt titkárnői feladatok elvégzésére. Tehát, ha valakinek tényleg rengeteg ideje van, és egy játékra kihegyezett gépet akar készíteni, a legjobb, ha egy külön telepítést készít. Ezen a külön rendszeren aztán saját magot is fordíthatsz, csak a szükséges elemeket használva, és nem is kell felrakni rá mindenféle egyéb csomagot, így nem lesz gond a frissítésekkel sem.

De térjünk vissza a cikk eredeti vezérfonalához. Tehát van egy grafikus felületünk, ami rossz esetben csak Vesa módban (még rosszabb esetben FrameBuffer vezérlővel) működik. A grafikus felület önmagában nem elég, szükséged lesz ablakkezelőre is. Ahogy azt egy korábbi cikkben kiveséztem (Linuxvilág 2002. október: Egy másik környezet), az ember több irányba is elin-

dulhat: a Gnome zászlóshajójaként törhet át az Északi-tenger jégtáblákkal tarkított habjain, a három szent betű, a K, a D és az E alatt számolgathatja a fogaskerekeket, vagy választhat egy gyors és üzembiztos, ellenben fapados ablakkezelőt. Mivel most a rendszer alapjait építjük fel, mindenképpen azt ajánlom, hogy amíg gyakori újraindítások várhatók, a harmadik lehetőséget válaszd. Javaslatom tehát az IceWM. Tudom, hogy számos más, remek ablakkezelő is van, és tényleg érdemes kipróbálni őket, de idővel úgyis körbenézel a kínálatban, ha olyan igényed támad, amire nem találsz itt megoldást.

2. kép Locales

Ha már fut a grafikus felület (akár egy bejelentkezéskezelővel, mint például a `kdm`, a `gdm` vagy az `xdm`, akár közvetlenül a parancssorból indítva), a következő lépés, hogy gyorsan használható környezetet alakítsunk ki. Figyelem, a most következő leírást megrögzött angolul beszélők, ékezeteket nem ismerők és egyetemi kollégiumban lakók nyugodtan kihagyhatják! Tehát a „hogyan varázsoljuk magyarrá a rendszert?” kérdésről van szó. Ugyancsak egy korábbi cikkben már részletesen végigvittük a folyamatot (Linuxvilág 2002. október: Debian Woody 3.0), a lényeg: első körben érdemes telepíteni a `locales` csomagot (`apt-get install locales`), majd beállítani, hogy milyen környezetekre készüljön fel a rendszer (négy igazán érdekes van számunkra, ezek a következők: en_US ISO-8859-1, en_US.UTF-8 UTF-8, hu_HU ISO-8859-2, hu_HU.UTF-8 UTF-8 locales). Ezek után telepíts a számodra érdekesebb csomagokat Aptitude-ból (érdemes a listát szűkíteni – `l` – például a `~d8859|~dhungar` feltétellel).

Mindenképpen szánj időt a csomagok megismerésére, ez egyszer elkerülhetetlen. Később úgyis ismerősként köszönnek majd vissza, de addig legalább egyszer érdemes végigböngészgetni, hogy milyen csomagok lehetnek érdekesek, az izgalmasabb csomagoknál esetleg a függőségeket is megnézheted.

Ha már van grafikus felületünk, gyorsan ismerkedjünk meg a legfontosabb és leggyakrabban használt programmal: a grafikus terminállal. Újabb néhány perc kötelező foglalkozásként telepíts a gépedre legalább négy terminálprogramot, és próbáld is ki őket alaposan: az első az `xterm`. Előnye, hogy szinte mindenhol megtalálod, gyors és megbízható. A második az `rxvt`, azok kedvence, akik a mikroméretű programokra buknak.

Az „okos” terminálprogramok közül kettőt érdemes ismerni: a `konsole`-t, ami a KDE alapértelmezettje, valamint a vetélytárs `gnome-terminal`-t. Mindkettő jól használható, mindkettő támogatja a többlapos üzemmódot (több terminált nyithatsz egy ablakban, mindegyiket egy füllel jelzi a program), mindkettőhöz hozzá kell szokni. Tudását és teljesítményét tekintve remek középutat jelent szerintem a `powershell`, mindenképpen érdemes kipróbálni, megnézni a beállítási lehetőségeit (lásd az *1. képet*). És az utolsó kategória a különcöknek szól: a többnyelvű terminálok, például az `mlterm`, amelyekben gyakran egy gombnyomásra lehet váltani a nyelvek (vagy akár az írásirányok) között.

Utolsó „felkészítő” tevékenységként egy karakteres szerkesztőprogramra lesz szükségünk. Olyanra, amit tudunk majd terminálból használni. Ebből van talán a legtöbbféle Linux alatt. Ha egy programozójelölt be akarja írni a nevét a történelemkönyvbe, valami maradandót kell alkotnia. Sokan igyekeznek ezt egy huszonnyolcezredik szerkesztőprogram megírásával megtenni. Ezek közül a programok közül is érdemes – legalább alapszinten – megismerni néhányat, hiszen bármikor előfordulhat, hogy nem a saját gépünkön kell valamilyen beállítást elvégezni, és éppen nincs netkapcsolat, hogy kiadhassuk az `apt-get install joe` parancsot. Nézzük csak, mik azok, amelyekre mindenképpen érdemes rászagolni? (A „rászagolás” alatt azt értem, hogy próbából hozz létre egy fájlt a kérdéses programmal, amely legalább két bekezdés hosszú; legalább az egyik sor legyen több mint 80 karakter, és utána még el is tudd menteni azt. Haladóknak a keresés-csere szolgáltatás kipróbálása is kötelező!)

Az alaptelepítésben mindenképpen találunk valamilyen `vi`-leszármazottat (lásd még bővebben a 78. oldalon). Erről a rossznyelvek szerint kettő dolgot kell tudni: az egyik, hogy a fanatikusok kéjesen magyarázzák meg bárkinek, ha kell, napjában ötvenszer, hogy ezt nem „ví”-nek, hanem „víjáj”-nak kell ejteni, a másik pedig, hogy a `:q!` kombinációval tudunk kilépni belőle a változások mentése nélkül.

Hasonlóan ősi vonalat képviselnek az Emacs család tagjai, bár itt már a kiejtésben is vita van az „emacs”-osok és az „imeksz”-esek tábora között. Valójában mindkét szerkesztő rendkívül jól és hatékonyan használható, de a használatukhoz nagy gyakorlat szükséges. Grafikus felületen alapprogramnak számít még az `xedit`, amit sok program használ külső forrásmegjelenítőként. Bár nem terminálon futó program, érdemes legalább egyszer próbára tenni.

A gyarló emberek számára találták ki a felhasználóbarát szerkesztőket, amelyek legalább annyi infót adnak, hogyan tudhatunk meg többet a használatukról. Némelyik még a kiadható parancsok listáját is megmutatja nekünk. Ide tartozik a (gondolom, az olvasók által is jól ismert) Midnight Commander szerkesztője, az `mcedit` (Linuxvilág, 2001. június–július: Szerkesztők háborúja (3. rész)) vagy a régi WordStar-parancsokat használó `joe` (Linuxvilág, 2001. április: Szerkesztők háborúja (1. rész)) is. Ezek könnyen használható és emberi léptékű programok, és nagy valószínűséggel a legtöbb rendszeren jelen vannak. Ugyanebbe a kategóriába tartozik a `pine` levelezőprogram szerkesztője, a `pico`(Linuxvilág, 2001. május: Szerkesztők háborúja (2. rész)), ami külön programként is remekül használható. Olyannyira kedvelt kis program, hogy a rendszerfelügyelettel kapcsolatos legújabb „de facto” szerkesztő, a `nano` is ez alapján készült. Ha tehát szerkesztőt kell ajánlanom, akkor a `nano` az, hiszen egyrészt könnyű megtanulni használni, másrészt a mérete és megbízhatósága miatt szinte minden alaprendszernek és helyreállító rendszernek a részét képezi.

A következő részben egy cseppet körbenézünk a hangok terén, majd elkezdünk böngészni!

Szy György (Szy.Gyorgy@linuxvilag.hu)
A Linuxvilág főszerkesztője,
a Kiskapu Kiadó vezetője.
Mindenki levelét örömmel várja.

Héjprogramozás Linux alatt (9. rész)

Zárolás, várakozás és többpéldányos futás.

Sorozatunk előző részében bemutattuk, hogyan oldható meg biztonságosan az átmeneti fájlok kezelése akkor is, ha a programunkból egyszerre több példány fut. Ugyanakkor vannak olyan helyzetek, amikor a többpéldányos működést kifejezetten tiltani akarjuk, vagyis zárolást kell végrehajtanunk. A zárolás lényege, hogy a héjprogram tényleges működésének megkezdése előtt meggyőződik róla, hogy fut-e már belőle egy másik példány. Ha igen, akkor vagy azonnal kilép, vagy megvárja, amíg a másik program futása befejeződik. Ha nem talál másik példányt, akkor valamilyen egyezményes módon jelzi, hogy ő már fut, és végrehajtja a megadott műveleteket. Természetesen arról is gondoskodnunk kell, hogy kilépéskor a zárolójelet eltávolítsuk, ellenkező esetben soha többé nem tud egyetlen ilyen program sem lefutni.

A zárolófájl kezelése

A zárolást a legegyszerűbben úgy oldhatjuk meg, ha egy meghatározott helyen létrehozunk egy megadott nevű fájlt, ami most a jelző szerepét fogja betölteni. Ha a fájl létezik, akkor már biztosan fut egy példány a programból, ha nem, akkor szabad a pálya. A módszer működésének természetesen alapfeltétele, hogy a fájlt kilépéskor mindenképpen töröljük, még akkor is, ha a program nem szabályos módon állt le. Mint azt a sorozat előző részéből már tudjuk, éppen erre szolgál a `trap` parancs.
Lássunk egy egyszerű példát! Tegyük fel, hogy héjprogramunknak egy beállításfájlt kell módosítania. Ha a programot többen is futtathatják, fennáll annak a veszélye, hogy egyszerre kísérlik meg módosítani. Ennek pedig egészen furcsa következményei lehetnek, hiszen részben vagy egészben felülírhatják egymás beállításait vagy teljesen összezavarhatják a fájl szerkezetét. Ebben a helyzetben tehát mindenképpen zárolás szükséges, amivel határozottan megtiltjuk a beállítóprogram többpéldányos működését.
Lássuk a program legegyszerűbb megvalósítását az *1. listában*!
A beállításokat tároló fájl (***proba.conf***) most az egyszerűség kedvéért a pillanatnyi könyvtárban található, a zárolás jelzésére szolgáló bejegyzést pedig a */tmp* könyvtárban fogjuk elhelyezni. (Természetesen bármilyen más könyvtár is – amelyhez mindazoknak írási jogosultságuk van, akik a programot futtathatják – megfelel a célnak.)
A zárolófájl neve most `pidlock` lesz, vagyis az átmeneti fájlokkal ellentétben semmilyen egyedi azonosítót nem tartalmaz. Ugyanakkor bizonyos később említendő okok miatt zárolás esetén is célszerű jelezni, hogy kicsoda – vagyis melyik folyamatazonosítóval rendelkező programpéldány – hozta létre. Ezt hagyományosan úgy oldják meg, hogy a folyamatazonosítót magában a zárolófájlban helyezik el.
Éppen ez látható példaprogramunk 14. sorában. A 9. sorban megvizsgáljuk, hogy létezik-e a zárolófájl. Ha igen, akkor a program egyszerűen leáll, egy üzenetben jelezve ennek az okát. Ha nem, akkor létrehozza a zárolást, majd belekezd a tényleges műveletek végrehajtásába. Ez esetünkben mindössze annyi lesz, hogy a `read` paranccsal egyetlen sornyi szöveget bekérünk, amit aztán hozzáfűzünk a beállításokat tartalmazó fájlhoz.

```
 1: #!/bin/sh
 2: # Példa zárolásra
 3:
 4: LOCKFILE="/tmp/pidlock"
 5: CONFIG_FILE="proba.conf"
 6:
 7: trap 'rm $LOCKFILE; exit 2' 1 2 3 15
 8:
 9: if [ -f $LOCKFILE ]
10: then
11:   echo "Már fut egy példány
          ➥a programból!"
12:   exit 1
13: else
14:   echo $$ > $LOCKFILE
15: fi
16:
17: echo -n "Új szöveg: "
18: read sor
19: echo $sor >> $CONFIG_FILE
20:
21: rm $LOCKFILE
```

```
 1: if [ -f $LOCKFILE ]
 2: then
 3:   echo "Már fut egy példány
          ➥a programból!"
 4:   echo "Várakozunk a kilépésére..."
 5:   while [ -f $LOCKFILE ]
 6:   do
 7:     sleep 2
 8:   done
 9: fi
```

Kilépés előtt (21. sor) a zárolást töröljük – utat engedve a többi példánynak. Ha menet közben bármilyen galiba történne, és a program nem lenne képes eljutni a 21. sorig, akkor a 7. sorban megadott műveletsor lép életbe. Itt a `trap` parancs elfogja az összes lényegesebb, a program leállásával kapcsolatos jelet, törli a zárolófájlt, majd engedelmeskedve a jelzésnek egy `exit` paranccsal kilép.

Kinek a zárolása?

Figyeljük meg, hogy a 0-s jel (`EXIT`), vagyis a szabályos kilépés nem szerepel az elfogott jelek sorában. Ennek két oka van: egyrészt programunkban nem szerepel olyan közbenső kilépési művelet, amely megelőzi a zárolás létrehozását. A program végén bekövetkező szabályos kilépésnél ugyanakkor mi magunk gondoskodunk a zárolás megszüntetéséről.
Mégha lenne is közbenső kilépés, az esetünkben akkor sem meg-

felelő megoldás, ha a fenti egyszerű `trap` paranccsal kezeltetjük az `EXIT` jelet, mivel így az esetlegesen elindított második programpéldány kilépéskor törölné az első által létrehozott zárolást. Ha mindenképpen egyben akarjuk kezelni a zárolás törlését, akkor az efféle galibák elkerülésére a törlés előtt meg kell győződnünk róla, hogy programunk a maga által létrehozott zárolófájlt törli és nem valaki másét. Éppen erre szolgál a fájlban tárolt folyamatazonosító. Ha ugyanis ez nem azonos a törlést

```
 1: if [ -f $LOCKFILE ]
 2: then
 3: # Hamis zárolás vizsgálata
 4:    lockpid=`cat $LOCKFILE`
 5:    if [ `ps ax | grep "^ $lockpid" |
       ➥wc -l` -eq 0 ]
 6:    then
 7:      echo "Hamis zárolás!"
 8:      rm $LOCKFILE
 9:      echo $$ > $LOCKFILE
10:   else
11:      echo "Már fut egy példány
            ➥a programból!"
12:      exit 1
13:   fi
14: else
15:   echo $$ > $LOCKFILE
16: fi
```

kezdeményező programéval, akkor biztosan nem az ő jelzéséről van szó, tehát nem törölheti. Mindez azt jelenti, hogy a 7. sorban található műveletsort a következőre kell lecserélnünk:

```
trap ' exit ' 1 2 3 15
trap ' if [ `cat $LOCKFILE` -eq "$$" ];
➥then rm $LOCKFILE ; exit ; fi' EXIT
```

Mindenekelőtt ennél a módszernél a 21. sor szerepeltetése nemcsak fölösleges, hanem kifejezetten hibás is, hiszen most a szabályos kilépéssel kapcsolatos törlést is a második `trap` utáni kódra bíztuk. Az első `trap` segítségével mindössze annyit teszünk, hogy a „nem szabályos” leállással (például a Ctrl+C megnyomása) kapcsolatos jeleket szabályos leállássá változtatjuk úgy, hogy hatásukra egy közönséges `exit` parancsot hajtunk végre. Ezt a következő, csak az `EXIT` jelet kezelő `trap` ismét elfogja, és a szokásos módon kezeli.
Az olvasó most nyilván azt kérdezi, hogy miért nem vonjuk össze a két `trap`-et, elvégre mindkettő pontosan ugyanazt a műveletsort hajtja végre. Látszólag tehát semmi mást nem kellene tennünk, mint a második után az elsőnél megadott jeleket is felsorolni, az elsőt pedig egyszerűen kihagyni:

```
trap ' if [ `cat $LOCKFILE` -eq "$$" ];
➥then rm $LOCKFILE ; exit ; fi' EXIT 1 2 3 15
```

Nosza, próbáljuk ki! Cseréljük le a két sort a fenti egyre, indítsuk el a programot, majd szöveg helyett vagy némi gépelés után nyomjuk meg a Ctrl+C billentyűket. Hibaüzenetet kapunk, amely szerint a zárolást jelző fájl nem létezik. (Hova lett?!)
A `trap` utáni utasítássorban kénytelenek vagyunk magunk is végrehajtani egy `exit` parancsot, különben a program soha nem áll le. (Az előző részből tudhatjuk, hogy halhatatlan program nincs. A `kill -KILL` parancs mindenkire jótékony hatással van.) Ha a `trap` nem `EXIT` jelet fog el, akkor maga mindenképpen kivált egy ilyent, amit aztán boldogan el is fog, hiszen szerepel a listájában. (Kétszer ugyanazt a jelet természetesen nem fogja el.)
A bűntény tehát a következőképpen történt. Amikor leütöttük a Ctrl+C billentyűket, a keletkező 2-es jelet a `trap` elfogta, és ennek eredményeképpen törölte a zárolófájlt. Ezután az `exit` paranccsal maga is kiváltott egy 0-s jelet, amit ismét elfogott. Törölni viszont már nincs mit, tehát hibaüzenetet kapunk, majd a program valóban leáll.
Akármelyik (helyes) megoldást választjuk is, példaprogramunk működéséről úgy győződhetünk meg, hogy két terminálablakban próbálunk meg belőle futtatni egy-egy példányt.

Várakozás zárolás feloldására

Az előbbi példában, ha a program zárolást érzékelt, egyszerűen leállt. Előfordulnak azonban olyan helyzetek, amikor célszerűbb megvárni, amíg a másik példány lefut és törli a zárolást. Ehhez az előbbi megvalósítás 9–15. sorát a *2. listában* található kódrészletre kell cserélnünk.
A `sleep` parancs semmi egyebet nem tesz, mint várakozik. Hogy meddig, azt az utána szereplő érték adja meg másodpercekben. (A `sleep` valójában nem végez pontos időzítést. Ez egy többfeladatos, többfelhasználós operációs rendszerben eleve nem egyszerű feladat.)
Esetünkben az 5. sorban induló ciklus két másodpercenként ellenőrzi, hogy a zárolófájl még létezik-e, és csak akkor áll le, ha ez a feltétel már nem teljesül. Természetesen más ellenőrzési szakaszt is megadhatunk, illetve egy számlálót is használhatunk, és bizonyos számú ellenőrzés után feltétel nélkül leállíthatjuk a programot.

Hamis zárolás törlése

Bonyolultabb programok esetében előfordulhat olyan helyzet, amikor valamilyen oknál fogva nem kezelhetünk minden leállító jelet, és így a zárolófájl a folyamat leállása után is hátramaradhat. Ha ilyenkor a fenti egyszerű megoldást használjuk a zárolás vizsgálatára, programunk a végtelenségig várakozni fog, hiszen semmilyen módon nem érzékeli, hogy a zárolást létrehozó program már nem fut, így esélye sincs annak feloldására.
A megoldás ismét a zárolófájlban tárolt folyamatazonosítóra épül. Ha listát kérünk valamennyi futó folyamatról (`ps ax`), és azok között nem szerepel a zárolás tulajdonosa, akkor nyilvánvalóan hamis zárolásról van szó, amit törölni kell. Mindez azt jelenti, hogy az eredeti programunk 9–15. sorát a *3. listában* található kódrészletre kell cserélni.
A 4. sorban kiolvassuk a zárolófájlban tárolt folyamatazonosítót; az ötödikben a teljes folyamatlistában megszámoljuk azokat a sorokat, amelyek elején ez a szám szerepel. (A `ps` a folyamatazonosító elé egy szóközt illeszt be!) Ha nincs ilyen, akkor a kérdéses folyamat már nem fut, tehát az általa létrehozott zárolást törölni kell, ugyanakkor létre kell hozni helyette az újat (9. sor). Programunk működését a legegyszerűbben úgy próbálhatjuk ki, ha egy „segédprogramot” írunk hozzá, ami egy hamis zárolást hoz létre a saját folyamatazonosítójával, majd azonnal leáll.

Büki 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? (3. rész)

A fájlkezelés alapjai, avagy nyissuk fel a motorháztetőt!

Sorozatunknak ebben a részében egy igazi mumushoz érkeztünk el. Talán egyik területet sem fedi nagyobb homály, mint a Linux fájlkezelését. A dolog már-már érthetetlen, hiszen a történet nagyjából harminc évvel ezelőtt kezdődött. Ekkor jelentek meg ugyanis az első Unixok, amelyek már akkor is ugyanezt az elvet követték, amelyet a Linux is szinte teljes mértékben átvett és a kezdetektől fogva alkalmaz. Jelen cikkben ezt a fájlkezelési rendszert szeretném bemutatni legalább olyan részletességgel, hogy kényelmesen meg tudjuk oldani a sorozat további részei során felmerülő, fájlokkal kapcsolatos feladatokat. Mivel a témakör meglehetősen átfogó, ki szeretném ragadni belőle az alapvető részeket, amelyeken keresztül bemutatható a unixos állományelgondolás – ennek segítségével viharmentes vizek felé kormányozgathatjuk rendszerünket. Ezenkívül szükségünk lesz egy-két fájlműveletet végző eljárásra annak érdekében, hogy élőben láthassuk, mi is a valós eredménye annak, amiről itt értekezünk. Kicsit tömörebben szólva: olyan kérdésekről lesz szó, amiket az eddigi használat során a SuSE grafikus felülete és maga a telepítő többnyire elrejtett a szemünk elől, ám ahhoz, hogy továbbhaladhassunk a mindennapi használat felé, feltétlenül szükségünk van a válaszokra.

Hová lett a C meghajtóm?

…kérdezgetik ijedten a frissen Linuxot telepített felhasználók. Már ***Einstein*** is megmondta annak idején, hogy az anyag nem vész el, legfeljebb átalakul. A mondás tökéletesen igaz a mi helyzetünkre is, a titok nyitja a unixos könyvtárszerkezet-kezelésben található. Linux alatt ugyanis a könyvtárszerkezetet megtestesítő faszerkezet hagyományosan egygyökerű. Ha jobban szemügyre vesszük eddig használt rendszerünket, megfigyelhetjük, hogy ott is ugyanilyen könyvtárszerkezettel dolgoztunk, ám a kiindulópont mindig egy meghajtó volt, így tehát az ott használt fának annyi gyökere volt, ahány meghajtóval a rendszer rendelkezett. Mostani összeállításunk esetében azonban a rendszer határoz meg számunkra egy gyökeret (ez a / könyvtár), s minden későbbiekben használt hálózati vagy helyi meghajtó könyvtárszerkezete csak ez alatt a gyökér alatt fakadhat. Mindezt úgy oldjuk meg, hogy az egyes meghajtókat, amiket eddig használtunk, befűzzük (mount) az új fájlrendszer egy meghatározott pontja alá, s ezek után tartalmát úgy érhetjük el, hogy belemegyünk abba a bizonyos könyvtárba, ahová az adott meghajtót előzetesen befűztük. Belelépve a könyvtárba észre sem vesszük, hogy egy másik meghajtóra léptünk volna át, olyan, mintha csak az adott könyvtár tartalmát böngésznénk. A művelet tehát a felhasználó számára teljesen átlátszó, a működés részleteit a rendszer teljesen elrejti, így ha nem ismerjük, hogy ki mikor mit hova fűzött be, meg sem tudjuk állapítani, hogy milyen fájlrendszeren tartózkodunk éppen. Nem látjuk, hogy helyi lemezrészről vagy hálózati meghajtóról van-e szó, hajlékonylemez, PenDrive vagy CD-ROM-e az, amit olvasunk pillanatnyilag olvasunk, s azt sem tudjuk azonnal, hogy az adott eszközön milyen típusú fájlrendszer van (természetesen a csatlakoztatásnál olyan nevekkel hivatkozunk a tartalomra, amelyekből azonosítható, hogy melyik eszközt használjuk). Ez az egységesen kezelt felépítmény az elmúlt jó pár évben már bizonyította hatékonyságát, ám az kétségtelen, hogy először hozzá kell szokni, s csak azután élvezhetjük igazán a használatát.

Meghajtók befűzése a rendszerbe

Az eddigi használat során ilyesmire nem volt szükségünk, hiszen a rendszer anélkül végezte el helyettünk ezt a műveletet, hogy észrevettük volna. Legelőször a telepítés során találkoztunk vele, ahol a lemezrészek kialakítását végeztük. Az egyes lemezrészek mellett ugyanis találtunk egy olyan jellemzőt, hogy ***Csatolási pont***, amely a rendszer egy könyvtárára mutatott. Alapértelmezés szerint a windowsos lemezrészek esetében a /Windows/C, D, E könyvtárakba fűzte be a rendszer a régi C, D, E stb. meghajtókat. A telepítés után a munkaasztalon találtuk a kötetcímke után elnevezett meghajtóikonokat, amelyekre kattintva megtekinthettük a tartalmukat. Biztos vagyok abban is, hogy mindenki rájött arra, hogy a CD-meghajtó ikonjára kattintva elérte annak tartalmát, ám amikor ki akarta venni a CD-t, az semmiképp sem adta magát. Ennek az az oka, hogy a rendszerbe behelyezett minden lemezt vagy hajlékonylemezt ugyanúgy be kell fűzni, mielőtt használatba vennénk, s eltávolításuk előtt le kell választani (unmount) őket a fájlrendszerről. A CD-t is csak azután vehetjük ki a meghajtóból, miután leválasztottuk. Ezt úgy tehetjük meg (mint ahogy sokan meg is tették), hogy a CD ikonra a jobb gombbal kattintva a menüből kiválasztjuk a ***Lecsatlakoztatás*** menüpontot. Előfordulhat, hogy hibaüzenetet kapunk, ami a legtöbb esetben azért következik be, mert a lemez valamelyik fájlját használjuk: épp lejátsszunk egy MP3-at a CD-ről, vagy a böngészőben nyitva van egy ablak, amely a CD tartalmára mutat. Zárjuk be azokat az alkalmazásokat, amelyek valamilyen módon használják a fájlokat, majd próbálkozzunk újra. Látni fogjuk, hogy a művelet ezúttal hibaüzenet nélkül lezajlik. Létezik néhány megoldás arra nézve is, hogy ne kelljen minden esetben ki-befűzögetni a CD-ket, lemezeket, ezekről a sorozat későbbi cikkeiben esik majd szó, most azonban kanyarodjunk vissza egy kicsit a befűzéshez.

Megjegyzés

A hajlékonylemezt leválasztás nélkül is eltávolíthatjuk, legfeljebb néhány hibaüzenetet kapunk utána, ám akad egy nagyobb baj is: elképzelhető, hogy a ráírt adatok nem lesznek rajta a lemezen. Ennek az az oka, hogy a fájlok a lemezre írás előtt egy átmeneti tárba kerülnek, hogy utána csoportosítva gyorsabban írhassuk a lemezre, s ha leválasztás nélkül vesszük ki a lemezt a meghajtóból, elképzelhető, hogy még nem történt meg a kötegelt lemezreírás. Ha viszont kiadjuk az `unmount` parancsot, biztosak lehetünk benne, hogy a rendszer az átmeneti tárban található adatokat is ráírja, így nem ér minket adatvesztés.

Új merevlemezem van, mit tegyek vele?

Tegyük fel, hogy a gépben található merevlemezfiókba (mobilrack) új meghajtót helyeztünk, s látni szeretnénk a tartalmát. Sokat nem kell aggódnunk, hiszen a rendszer indításakor a SuSE eszközfelismerő programja felkínálja a lehetőséget, hogy az újonnan behelyezett lemezt beállítsuk. Ha igent mondunk, a rendszer elindítja a lemezrészkezelő programot, ahol új lemezrészeket hozhatunk létre, s a ***Csatlakoztatási pont*** mezőben megadhatjuk, hogy fájlrendszerünk mely pontjához szeretnénk csatlakoztatni az új lemezrészt. Meglévő lemezrész esetében természetesen nem kell újat létrehozni. Ha például le szeretnénk másolni a hordozható merevlemezen található adatokat, amelyet épp most hoztunk az ismerősünk gépéből, az elinduló lemezrészkezelő programban csak adjuk meg a csatlakoztatási pontot a felismert lemezrész mellett, s a művelet befejezése után azonnal egy, a többihez hasonló meghajtóikont kapunk az asztalon. Ezek után a lemezrész elérhető.

Egy kis háttéradat

A lemezkezelő valójában egy előlap (frontend) valamilyen lemezkezelő programhoz, és a lemezek befűzését végző `mount` rendszerhez. A meghajtóbejegyzések a */etc/fstab* beállításfájlban találhatók, ebbe bárki belenézhet. Rendszerindításkor a benne található bejegyzéseken végigfut a `mount` parancs, és amelyiknél engedélyezve van az önműködő befűzés, azt csatlakoztatja. A `mount` parancs természetesen kézzel is futtatható, a használata egyszerűsítve valahogy így néz ki:

```
mount -t <fájlrendszer típusa> <eszközkezelő fájl neve> <csatlakoztatási pont>
```

Az így befűzött meghajtók azonban a következő rendszerindításkor nem csatlakoznak önműködően, ehhez a bejegyzést rögzíteni kell a */etc/fstab* fájlban.

A könyvtárszerkezet

Nem árt, ha tudjuk, hogy mit hol találunk. Ilyen megfontolásból szeretném a főbb könyvtárak rendeltetését bemutatni – természetesen a teljesség igénye nélkül.

/bin: itt találhatók az alaprendszer indítható állományai a hétköznapokban leggyakrabban használt parancsok (például: `date`, `chmod`, `chown`).

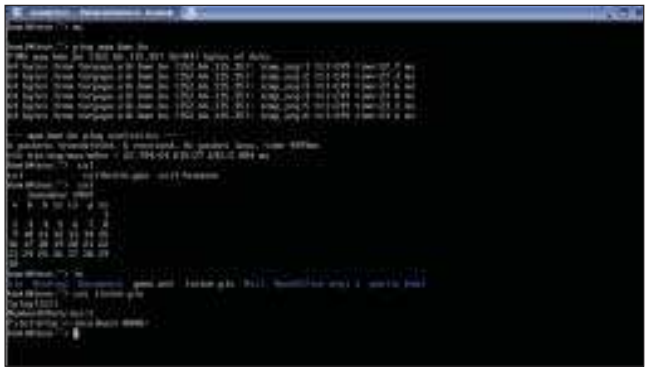
/boot: ebben a könyvtárban vannak a rendszerindításhoz nélkülözhetetlen állományok, köztük maga a rendszermag – a rendszertöltő állományai, beállításai is itt lelhetők fel.

/dev: ez a könyvtár csupa különleges fájlt tartalmaz, itt vannak ugyanis az egyes eszközöket képviselő fájlok, a *dev* könyvtáron belül kategóriákba rendezve; ha valamelyik alkalmazás meg akar szólítani egy eszközt, akkor valójában az eszköz kezelőállományához fordul, amelyeket egytől egyig itt talál meg.

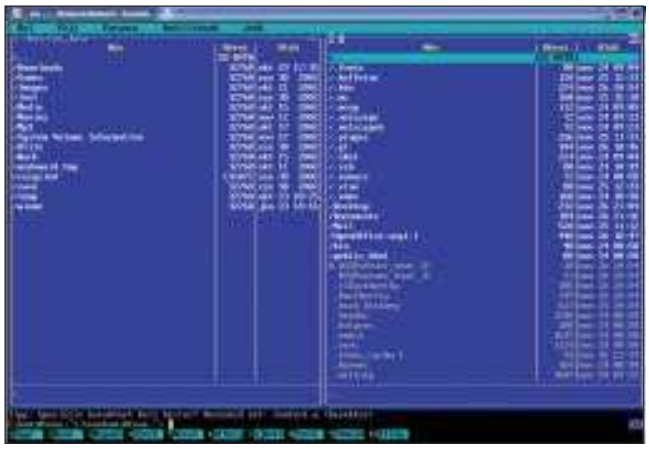
/etc: szinte kivétel nélkül itt helyezkednek el a rendszerközeli alkalmazások beállításfájljai; a legtöbb estben, amikor valamit testre kell szabnunk, ebben a könyvtárban kell keresnünk az átállítási lehetőségeket.

/home: s különböző felhasználók saját könyvtárai találhatók itt, mindenkihez tartozik egy; ezen a könyvtáron belül minden felhasználó azt csinál, amit csak akar – fájlokat hozhat létre, törölheti őket stb.

/lib: más alkalmazások számára különböző szolgáltatásokat kínáló állományokat találhatunk itt; a segítségükkel elérhető,



1. kép Konsole – A KDE X-terminálja



2. kép Így néz ki a Midnight Commander

hogy ugyanazokat az előre megírt eljárásokat több alkalmazás is használja; nagyon hasonlóak a Windowsnál megszokott DLL-állományokhoz

/proc: ez egy látszólagos fájlrendszer; gépünk alkatrészeiről találunk itt mindenféle adatot, a legtöbb alkalmazás is innen olvassa ki a gép tulajdonságait; mindegyik itt található fájl egy szöveges állomány, amely minden rendszerindítás során újra létrejön.

/tmp: a rendszer működése során keletkező átmeneti állományok tárolására szolgál, így rendet lehet tartani az egyes alkalmazások háza táján, mivel mindenki egy közös helyre szemetel, amit a rendszer hatékonyan karbantarthat.

/usr: itt találhatók a felhasználói alkalmazások állományai; ha egy általános célú csomagot telepítünk, annak a célterülete szinte minden esetben e könyvtáron belülre mutat.

Jogosultságok

Ezennel a fájlrendszerrel tisztában vagyunk annyira, amennyire nekünk, kezdő felhasználóknak szükségünk van, ám most egy másik fontos tényezőt kell szemügyre vennünk, amely azért fontos, hogy kellő hozzáértéssel hozzunk létre fájlokat, mozgassuk, másoljuk őket stb.

Mint már említettem, a Linux többfelhasználós rendszer, ami azt jelenti, hogy egy időben egyszerre többen is használhatják – akár a helyi gépen, akár valamilyen terminálon vagy hálózati kapcsolaton keresztül. Ahhoz azonban, hogy akár a véletlen, akár a szándékos károkozást vagy adatlopást megakadályozzuk, gondoskodni kell arról, hogy mindenki csak olyan állományokhoz férjen hozzá, amelyek a rendeltetésszerű használathoz szükségesek. Egy kicsit

bővebben ez azt jelenti, hogy egy átlagos felhasználónak nem kell látnia mások állományait, neki ahhoz semmi köze, és a rendszer működéséhez szükséges állományokat sem szabad megváltoztatnia! E feladatok megoldására létezik az úgynevezett jogosultságrendszer, amely szinte egyidős a fent vázolt fájlrendszerrel, fájlszerkezettel, és egyébként szorosan kapcsolódik hozzá.

Nézzük, hogy is van ez!

A rendszer összes állományának – beleértve a befűzött meghajtókon találhatókat is – vannak bizonyos jellemző tulajdonságai. Itt természetesen nem a méretre és a formátumra gondolok, hanem azokra a pluszadatokra, amelyek azt az ismeretet hordozzák, hogy egy fájllal egy adott felhasználó milyen műveleteket végezhet. Ezek közül számunkra most három oktális szám válik fontossá. E három szám alapján ismeri a rendszer a felhasználók jogosultságait az adott fájlra, s különböztet meg három felhasználói réteget: a tulajdonost, a csoportot és mindenki mást. Mindhárom rétegre vonatkoznak bizonyos jogosultságok (mindegyik esetében ugyanazok közül lehet választani), s a hozzáférni kívánó felhasználó alapján dől el, hogy ki mit csinálhat, kire mi vonatkozik.
Ez egy kissé zavaros, ugye? Ne aggódjunk, hamarosan mindenki megvilágosodik!

Az a mágikus oktális szám

Ez a nyolc variáció alkotja a Unix-rendszerek állományain végezhető műveletek halmazát. A dolog talán a kettes számrendszer használatával érthető meg legjobban. Minden helyi értéknek egy művelet felel azokon a helyeken, ahol a művelet engedélyezett, a helyi értéken képviselt értékek összeadódnak. Alapvetően háromféle jogosultság van: olvasás (r), írás (w) és futtatás/böngészés (x). Ez az utóbbi jogosultság fájlok esetében futtatásra vonatkozik, könyvtárak esetében pedig arra, hogy az oda belépő felhasználó lekérheti-e a könyvtár tartalmának a listáját. Ezeket egymás mellé írva azt kapjuk, hogy: `rwx`. Ez maga az oktális szám emberi formátumban való megtestesülése. Az első betű helyi érték szerint 4-et, a második 2-t, a harmadik 1-et jelent. Ezek után, ha egy állományra az adott jogosultságcsoportból olvasási és írási jog van, azt egy 6-os szám jelképezi, ha csak olvasható, azt egy 4-es. Ilyen számból három van, s ez mondja meg az egyes csoportokhoz kötött jogosultságokat, amiket aztán a rendszer értelmez.
Biztosan felmerült már néhány emberben, hogy mit jelent a futtathatóság. Ezekben a rendszerekben nem a kiterjesztés alapján dől el, hogy egy adott állomány indítható-e, hanem ez is a fájl egy tulajdonsága.
Példaképpen hozzunk létre egy fájlt, majd mindenkinek adjunk hozzá olvasási jogot, magunknak pedig írási/olvasásit. Ehhez indítsunk el egy parancssort (például a Konsole a KDE menü mellett a panelen), majd adjuk ki a következő parancsokat:

```
cd ~
touch probafajl
chmod 644 probafajl
```

Ez a bizonyos `chmod` parancs szolgál arra, hogy a segítségével megváltoztassuk egy fájl jogosultságait. Legegyszerűbb használatában a folyamatosan emlegetett három oktális szám egymás után írásával adhatunk újabb jogokat a fájlnak, könyvtárnak.

Haladjunk az elejéről, s nézzük a tulajdonos jogait! Minden állománynak, könyvtárnak van egy tulajdonosa, aki egy létező felhasználó a rendszerben (a tulajdonos általában az, aki az adott fájlt létrehozta). Az első oktális szám rá vonatkozik. A fájl tulajdonosa a többi jogosultságtól függetlenül azt tehet az állománnyal, ami itt engedélyezve van.
A következő szám a csoport jogaira vonatkozik. Minden fájlhoz tartozik ugyanis egy felhasználói csoport, mint ahogy minden egyes felhasználó is benne van egy neki megfeleltetett csoportban (például a felhasználók csoportjába tartozik Józsi). Ha egy olyan felhasználó akar hozzáférni a fájlhoz, aki az adott csoportban benne van, az az oktális számnak megfelelő műveleteket végezheti el a fájllal.
Végül az utolsó szám az egyéb, avagy „kívülálló” kategóriára vonatkozik. Mindenki, aki nem tagja a csoportnak és nem is tulajdonosa az állománynak, az itt lefektetett jogokat birtokolja. Igen ám, de ezek közül egy jogosultság csakis akkor érvényes, ha vagy a tulajdonosi, vagy a csoportra vonatkozó ugyanezen jog engedélyezett.
Ezeket a jogosultságokat csakis a rendszergazda (root), vagy az adott fájl/könyvtár tulajdonosa változtathatja meg – ha ez nem így volna, az egész védelem nem érne semmit, hiszen bárki felhatalmazhatná magát egy fájl használatára.
Ide kapcsolódik még az a kérdés is, hogy hogyan adható át egy fájl tulajdonjoga egy másik felhasználónak vagy hogyan változtatható meg a fájlhoz rendelt csoport? A válasz felemás, ugyanis a tulajdonjog senkire sem ruházható át, a hozzárendelt csoportot viszont a fájl tulajdonosa megváltoztathatja. Ezek a szabályok nem vonatkoznak azonban a renszergazdára: ő bárkinek átadhatja bárki fájljait és bármelyik fájl csoportját megváltoztathatja.
Minden ilyen változtatás a `chown` paranccsal történik, amelynek használata a következő:

```
chown: <felhasználó>:<csoport>
➥<fájl vagy könyvtár>
```

Egy kis könnyítés…

Ez bizonyára soknak tűnt így elsőre, de azzal szeretném megnyugtatni a kedves felhasználókat, hogy ezt azért nem kell állandóan használnunk, de egy „Hozzáférés megtagadva” üzenet esetén azonnal tudjuk a hiba okát, s orvosolni tudjuk azt – rendszerismeretünkhöz elengedhetetlen ezeknek a tényeknek az ismerete. Azért nem kell mindent kézzel beállítani, az állományoknál a grafikus felület is kínál számunkra beállítási lehetőségeket. Kattintsunk rá valamelyik meghajtóikonra, majd válasszunk egy tetszőleges fájlt vagy könyvtárat, s kattintsunk rá az egér jobb gombjával, majd válasszuk a legalsó ***Tulajdonságok*** menüpontot. Eredményül egy panelt kapunk, amelyen a második fül a jogosultságokra vonatkozik. Itt a fent tárgyaltakat állíthatjuk be igen kényelmes módon, feltéve, ha van hozzá jogunk. Emellett van néhány parancssoros eszköz, amelyek nemcsak a jogosultságkezelést könnyítik meg, de a fájlműveleteket is segítik úgy általában. Ilyen eszköz a közismert Midnight Commander, amelynek ismertetése előtt azonban egy kis kitérőt kell tennünk.

A parancssor használata

A Unix-rendszerekről általában elmondhatók, hogy parancssorban rendkívül erősek, tehát igen hatékonyan lehet használni a rendszerünket szöveges utasításokon keresztül is. Ha jobban belegondolunk, tulajdonképpen a legtöbb grafikus szolgáltatás is ezekre a parancssori elemekre épül, azzal a különbséggel,

3. kép Jogosultságok megtekintése a Konqueror segítségével

hogy ott fizikailag nem mi gépeljük be a parancsokat.
Az egész folyamatot tekintve a dolog úgy zajlik, hogy bejelentkezéskor mindenkihez hozzárendelődik egy héj (shell), s utána ezen a héjon belül tevékenykedhetünk, kihasználva a rendszerünk nyújtotta szolgáltatásokat. (Ez a fajta héj-hozzárendelődés természetesen a távoli felhasználás során is érvényes.) A grafikus munkakörnyezet ezt ugyan elrejti előlünk, de a háttérben ott van, olyannyira, hogy egy parancssoros ablak formájában akár elő is hozhatjuk. Ebből egyébként tetszőleges számút futtathatunk egymás mellett, mindehhez csak az kell, hogy valahonnan (a bejelentkezéskor kapott héjból) el tudjuk indítani őket. Számunkra az alapértelmezett program, amely biztosítja a parancssort, nem más, mint a KDE egyik alkalmazása, a Konsole, amelyet a panelen a KDE menü mellett találunk. Elindítva egy a windowsos parancssornak megfelelő linuxos parancsértelmezőt kapunk, ám ennek az erejével, tudásával össze sem hasonlítható redmondi társa (ennek többnyire hagyomány- és szemléletbeli okai vannak). Annak értekében, hogy mindez hihető legyen, sorozatunk egy későbbi részében egy teljes cikket fogunk áldozni a linuxos parancssor, azon belül is a bash nevű parancsértelmező program bemutatására.
Egy valamirevaló linuxos szakember élete nagy részét ilyen parancsértelmezők között tölti, s legfeljebb a weblapokat nézegeti grafikusan – nem árt tehát, ha fokozatosan hozzászokunk (eleinte csak a gondolathoz) a parancssorok jelenlétéhez, aktív használatához.

A Midnight Commander

Ez egy klasszikus, meglehetősen idős, karakteres módú fájlkezelő alkalmazás, nevéből kikövetkeztethetően a DOS-os Norton Commanderhez, a DOS Navigatorhoz vagy éppen a windowsos Total Commanderhez hasonló feladatot lát el. Tisztában vagyok vele, hogy nincs az az ember, aki ne használna ilyen osztott paneles fájlkezelő megoldást. Én Linux alá ezt a programot javaslom, hiszen kis memóriaigénye és karakteres módú megjelenése miatt bárhol alkalmazható. Mindenekelőtt telepítsük az alkalmazást, amelyhez indítsuk el a YaST vezérlőközpontot, s ott a ***Szoftver*** kategóriából válasszuk a ***Szoftver telepítése és eltávolítása*** lehetőséget (mint azt az előző cikk csomagkezelésről szóló részében megtanultuk). Itt a csomagcsoportokon belül a ***File Utilities*** kategóriában található ***mc*** csomagra lesz szükségünk. Ha valamilyen okból kifolyólag nem találnánk meg, válasszuk a keresés lehetőséget, s keressünk rá az ***mc*** kifejezésre. A telepítés után nincs is szükségünk másra, mint elindítani egy parancssort, s kiadni az `mc` parancsot.
Indítás után elénk tárul a már jól megszokott felépítésű fájlpanel-képernyő, így már talán otthonosabban érezhetjük magunkat a rendszerünkben. A részletes megismerést az olvasóra bízom, itt most csak néhány témába vágó szolgáltatását szeretném kiemelni. Először is itt van ugye a fájlműveletekhez nyújtott támogatás (másolás, törlés, mozgatás stb.), másodszor pedig a segítségével viszonylag átlátható formában állíthatjuk be a fájlok jogosultságait.
Tegyünk is egy próbát, s keressük meg az előbbi példában létrehozott ***probafajl*** nevű állományunkat, álljunk rá a kurzorral, majd nyomjuk le a Ctrl+x billentyűkombinációt, ezután nyomjuk le a c billentyűt. Azonnal láthatjuk a jogosultságkezelő ablakot, ott pedig a jól megismert három kategóriát: ***tulajdonos***, ***csoport***, ***mások***. Mindegyik mellett ott az emlegetett ír, olvas, futtat/belép jogosultság, a mi dolgunk csak annyi, hogy a szóközbillentyű segítségével beikszeljük azt a jogot, amire szükségünk van, így nem kell binárisan vagy épp oktálisan számolgatnunk a beállítandó jogosultságnak megfelelő kódot.
Ha ezzel megvolnánk, maradjuk továbbra is a megadott állományon, s most nyomjuk le a Ctrl+x kombinációt, majd az o billentyűt. Ekkor a tulajdonlást kezelő ablak jelenik meg, ahol tulajdonosként a listából kiválasztva egy másik csoportot rendelhetünk a fájlhoz, rendszergazdaként pedig ugyanezen a módon a fájl tulajdonosát is megváltoztathatjuk.

Összegzés

Most végre elmondhatjuk magunkról, hogy valamelyest konyítunk a linuxos fájlkezeléshez, s ha nem is ismerünk minden apró trükköt, a mindennapi feladatok során már egészen jól elboldogulunk ezekkel az ismeretekkel. Nem kell azonban minden bitet részletesen ismernünk, a gyakorlatban az itt elmondottakra kezdőként olyan helyzetekben lehet szükségünk, ha mondjuk kézzel, s nem csomagból telepítünk egy alkalmazást, s esetleg olyan könyvtárba kell azt másolni, ahová csak a rendszergazdának van írási joga. Ebben az esetben egyetlen teendőnk van: be kell jelentkezni rendszergazdaként, s úgy elvégezni az előírt műveletet. Hasonló gond léphet fel bizonyos fájlok létrehozásánál törlésénél, vagy egyéb olyan alkalmazásoknál, amik egy olyan fájlt szeretnének használni, amelyre nincs is olvasási joguk. Ekkor, ha ismerjük a fájlt, egy huszárvágással megoldhatjuk a problémákat, s idővel a parancssor egyre fokozódó használatával eljutunk oda, amikor már mindenféle külső segítség nélkül, csupán az addig összegyűjtött ismeretanyag felhasználásával képesek leszünk minden nehézséget elhárítani.
Kellemes gyakorolgatást!

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

Linuxos kiszolgálót mindenkinek! (2. rész)

A SuSE Linux mint kiszolgáló – kisvállalati és otthoni környezetben.

Az elmúlt alkalommal eljutottunk oda, hogy – elsősorban kiszolgálói feladatok ellátására – feltelepítettünk egy SuSE Linuxot. Most ezt a kiszolgálót fogjuk finomhangolni: felhasználókat, felhasználói csoportokat készítünk, beállítjuk a kiszolgáló biztonsági szintjét, felügyelhetővé tesszük a gépet és tűzfalat készítünk.

Felhasználók és felhasználói csoportok

Felhasználók létrehozásakor ökölszabályként alkalmazandó, hogy a felhasználókat a működési területüknek megfelelő felhasználói csoportokba tegyük, a felhasználói csoportokat pedig úgy hozzuk létre, hogy jól elkülöníthetőek legyenek egymástól. Minden felhasználónak csak a működéséhez, a munkája elvégzéséhez szükséges jogosultságokat adjuk meg. Felhasználókat vagy a `useradd` paranccsal vehetünk fel, vagy a YaST rendszergazdai felületen keresztül. A `useradd` parancsról bővebb leírást kaphatunk a megfelelő súgóoldal megnyitásával. A YaST-on keresztül egy olyan felülethez juthatunk, ami az összes felhasználót és felhasználói csoportot megmutatja nekünk, így grafikusan is jó áttekintést kapunk a felhasználókról. A YaST-on belül a ***Biztonság és felhasználók*** menüpontban találjuk a felhasználók felügyeletével foglalkozó részt. A ***Csoportok létrehozása és módosítása*** menüpontot kiválasztva megjelenik a ***Felhasználók és csoportok adminisztrációja*** ablak, ezen belül pedig egy lista a már létező felhasználói csoportokról. Az ablak alján található ***Szűrő beállítása*** és ***Egyedi szűrő*** gombbal szűrési feltételeket adhatunk meg, így téve még átláthatóbbá a listát. A ***Hozzáadás*** gombbal új csoportot hozhatunk létre. Ekkor meg kell adnunk egy csoportnevet, egy egyedi csoportazonosító számot, megadhatunk továbbá jelszót a csoporthoz, és rögtön ki is jelölhetjük, hogy mely felhasználókat társítsuk a csoporthoz.

Felhasználók hozzáadásához a ***Felhasználók létrehozása és módosítása*** menüpontot válasszuk. Ekkor az előbbihez hasonló ablak jelenik meg, ahol szintén meghatározhatunk szűrőket és a ***Hozzáadás*** gombbal új felhasználót adhatunk a rendszerhez. A megjelenő ablakban meg kell adnunk a felhasználó teljes nevét, az azonosítóját, illetve a jelszavát. A ***Jelszóbeállítások*** gombra kattintva a jelszó lejáratával kapcsolatos adatokat adhatjuk meg, úgymint figyelmeztetés, az érvényesség időtartama, a lejárat pontos időpontja. A ***Részletek*** gombra kattintva beállíthatjuk a felhasználó egyedi azonosítószámát, saját könyvtárának a helyét – amit egyébként a felhasználói csoportoknak megfelelően érdemes elhelyezni, és ez alapján csoportosítani a felhasználókat –, megadhatunk kiegészítő felhasználói adatokat, a bejelentkezési héjat (shellt), alapértelmezett és további csoportokat. Héjnak érdemes a felhasználó szükségleteinek megfelelő héjat megadni. Így amennyiben felhasználónk csak levelezést vagy megosztott könyvtárakat használ, úgy héjnak a */**bin**/**false***-t adjuk, így megakadályozzuk, hogy ez a felhasználó a kiszolgálóra akár helyileg, akár távoli eléréssel (SSH, Telnet stb.) közvetlenül bejelentkezzen. Amennyiben a felhasználó munkája során szükség van konzolra, ráadásul párhuzamosan több felületet is használnánk, úgy használjuk

a */**usr**/**bin**/**screen*** héjat, amit a ***screen*** csomag telepítése után érhetünk el. Ha ezzel végeztünk, kattintsunk a ***Következő*** gombra, majd fogadjuk el az eddig elkészített beállításainkat.

Biztonsági beállítások

A YaST ***Biztonság és felhasználók*** menüpontjának biztonsági beállításai közül először foglalkozzunk a ***Biztonsági beállítások***kal. Ezt a modult indítva három előre meghatározott biztonsági összeállítás közül választhatunk, vagy létrehozhatjuk a sajátunkat. Beállíthatjuk, hogy a felhasználói jelszavak létrehozásakor a rendszer egy szótárral hasonlítsa össze a megadott jelszót, és ha megtalálja, akkor figyelmeztesse a felhasználót, hogy egy ilyen jelszót esetleg könnyen ki lehet találni. A ***Jelszó elfogadhatóságának ellenőrzése*** lehetőség pedig biztosítja, hogy a felhasználónak megfelelően bonyolult jelszót gépeljen ahhoz, hogy a rendszer elfogadja. Megadhatjuk továbbá a jelszó titkosításának eljárását, a legfeljebb figyelembe vett karakterek számát, a legkisebb jelszóhosszt, valamint a jelszó lejárata előtti figyelmeztetést napokban.

Tűzfalak

Rendszerünk védelme szempontjából nagyon fontos szerepet kap egy jól beállított tűzfal. Tűzfalunkra hárul ugyanis az a feladat, hogy elválassza a biztonságos belső hálózatot a nem biztonságos külső hálózattól, például az internettől.

A tűzfalaknak többféle összeállítása létezhet az alkalmazott hálózat topológiájától, szolgáltatásától, biztonsági követelményeitől függően. A megfelelő összeállítás kiválasztása mind a rendszer biztonsága, mind a rendszerre fordított anyagiak szempontjából kényes kérdés. Mielőtt nekiállunk tűzfalat készíteni, érdemes kockázatelemzést készíteni, hogy a megfelelő biztonságot garantálni tudjuk, de azért ne szaladjon el velünk a ló.

A legegyszerűbb eset, amikor egy kiszolgáló védelméről kell csak gondoskodni. Ekkor elég felmérnünk azt, hogy a kiszolgálón milyen szolgáltatások fognak futni, és a tűzfalat ehhez

mérten tudjuk beállítani. Ilyenkor elég a szolgáltatások által igényelt kapuk kinyitása, természetesen a kapuk forgalmának monitorozása mellett.
Kicsit bonyolultabb a helyzet, ha olyan hálózatunk van, ami egy adott gépen (hálózati átjárón) keresztül éri el mondjuk az internetet. Ekkor a tűzfalunkat úgy kell beállítani, hogy az védelmet nyújtson az átjárónak és a mögötte lévő hálózatnak is úgy, hogy a hálózati ügyfelek munkáját ne akadályozza. Különös esetben elképzelhető, hogy a hálózati átjárót a belső hálózat oldaláról is védeni kell – ez csak tovább bonyolítja a dolgunkat. A SuSE Linuxba épített tűzfalszolgáltatás akár háromrétegű tűzfalszerkezetet is támogat. Így lehetőségünk van a belső hálózat–DMZ–külső hálózat felépítés kialakítására. Belső hálózatunk értelemszerűen a biztonságos helyi hálózat; a külső hálózat a nem biztonságos hálózati szakasz, ahonnan a támadások várhatóak; s a DMZ pedig nem más, mint a szabad zóna

Frissítés SuSE 9.0-ra

Magyarországon is bemutatták a SuSE Linux 9.0-s kiadását, amely a fejlesztők szerint az utóbbi idők egyik legkiforrottabb Linux-változata. Sok olyan szolgáltatást kínál, ami miatt egy kiszolgálót is érdemes lehet frissíteni az új kiadásra. Ilyen szolgáltatás például a YaST-ba épített Samba, DHCP, DNS vagy Apache modul, hogy csak párat említsek. Természetesen a frissítés szükségességét mindenkinek saját magának kell megítélnie, de én úgy döntöttem, hogy frissítek, és a cikksorozatban a továbbiakban a SuSE 9.0-s kiadás nyújtotta lehetőségeket is bemutatom.
A frissítés maga nem túl bonyolult dolog, de azért körültekintést igényel. A telepítés a megszokott módon kezdődik: először behelyezzük a CD-t vagy a DVD-t a meghajtóba, és a gép indítása után kiválasztjuk a *Telepítés* menüpontot. A telepítő a megszokott módon először a telepítés nyelvét állítja be, itt válasszuk a magyart. A következő lépésben a telepítő rákérdez a telepítés típusára. Amennyiben már meglévő SuSE Linux-rendszerünk van, úgy válasszuk a Meglévő rendszer frissítése lehetőséget, s ha új rendszert szeretnénk telepíteni, akkor természetesen az *Új telepítés* menüpont a megfelelő választás.
Miután a telepítés menetét az előző cikkemben már leírtam, és ez a 9.0-s kiadásnál sem különbözik lényegesen, most a frissítés menetével foglalkozom.
Miután a telepítő ellenőrizte a meglévő rendszert és a csomagadatbázist frissítette az új változatra, megjelenik a *Telepítési beállítások* ablak. Itt az új rendszer telepítéséhez hasonlóan megadhatjuk a telepítendő rendszer nyelvét, billentyűzetkiosztását, de amire mi most a figyelmünket összpontosítjuk, az a *Frissítési mód* és a *Biztonsági mentés* menüpontok.
A *Frissítési mód* menüpont alatt adhatjuk meg, hogy egy előre összeállított csomaglista alapján kívánjuk-e frissíteni a rendszert, vagy a meglévő csomagjainkat kívánjuk frissíteni. Az előbbi előnye, hogy felteszi a legfrissebb, legújabb szolgáltatásokat, az utóbbié, hogy nem változtat a meglévő csomag-összeállításon. Én az utóbbit javaslom, mert az új csomagokat később is tudjuk telepíteni, viszont ezzel a beállítással biztosított, hogy a rendszer frissítés után is a már összeállított kiépítésnek megfelelően fog működni. Fontos még odafigyelni az ablak alján lévő *Karbantartás nélküli csomagok törlése* jelölőnégyzetre. Ezt javasolm kikapcsolni, mert ha nem tesszük meg, az olyan csomagokat, amelyekről a rendszer úgy ítéli meg, hogy nem fogjuk használni, egyszerűen letörli, ez pedig meglepetéseket okozhat a későbbiekben.
A *Biztonsági mentések* menüpont alatt lehetőségünk van a régi rendszerről készítendő mentések beállítására. Mindenképpen jelöljük be a biztonsági mentés készítését a módosítandó állományokról és a teljes */etc/sysconfig* könyvtárról. A mentés később a */var/adm/backup* könyvtárban lesz található, ahonnan akár törölni is lehet.
Ha ezekkel a beállításokkal végeztünk, indíthatjuk a frissítést.
A SuSE saját bevallása szerint a 7.3, 8.0, 8.1 és 8.2 rendszerekről történő frissítés zökkenőmentes lesz. Ettől függetlenül készítsünk az adatainkról, beállításokról biztonsági másolatot, és ahogy mondani szokták: mindenki a saját felelősségére csinálja!
Ha a telepítő végzett a csomagok telepítésével, akkor hátravan még az internetkapcsolatunk kipróbálása és a frissített csomagok letöltése az internetről. Frissítéshez válasszuk a *suselinux.hu* kiszolgálót, és ha gondoljuk, akkor bízzunk meg a rendszer önműködő csomagtelepítésében.
Ezzel végeztünk is, feltettük a legfrissebb SuSE Linuxot, egy ajánlott újraindítás után dolgozhatunk is tovább.

(demilitarized zone), vagy elsődleges hálózat (perimeter network), ami egy köztes, elkülönített réteg a biztonságos belső hálózat és a külső hálózat között vagy mellett. Itt szokták elhelyezni az olyan gépeket, amelyek például az internet felől elérhető szolgáltatásokat nyújtanak. Mivel a szabad zónát mindkét oldalról tűzfal határolja, az esetlegesen a szabad zónában megtámadott gépről a belső hálózaton keresztül nem tudnak további adatokat szerezni – sikeres támadás esetén a támadó nem jut rögtön hozzáféréshez a belső hálózathoz. Ez a kiépítés a legbiztonságosabb megoldás, de elég drága, hiszen további eszközök beszerzését teszi szükségessé, illetve a kiépítése bonyolultabb, mint egy belső hálózat–külső hálózat felépítésű rendszeré.
Otthoni, kisvállalati felhasználási körben – megfelelő biztonsági kockázatelemzést követően – dönthetünk úgy, hogy ez az utóbbi felépítés megfelelő biztonságot nyújt számunkra. Ebben az esetben a belső hálózat–külső hálózat kiépítésére a YaST megfelelő eszközöket nyújt.
A YaST ***Biztonság és felhasználók*** menüpontja alatt található a ***Tűzfal*** modul, amelynek az indításával kezdhetjük meg a tűzfalunk beállítását. Az első lépésben meg kell adnunk a kiszolgáló külső és belső hálózatához tartozó hálózati csatoló nevét. Figyeljünk rá oda, hogyha ADSL kapcsolatot használunk, akkor nem a hálózati kártyát kell megadnunk, hanem az adott eszközhöz tartozó PPP-csatolót (általában a `ppp0`-t).
Következő lépésben megadhatjuk, hogy milyen szolgáltatásokat akarunk elérhetővé tenni a kiszolgálón, így például a HTTP, SMTP, egyéb levelezési protokollok, SSH. Érdemes arra odafigyelni, hogy a HTTP, SMTP, POP3, IMAP, `telnet` protokollok a teljes adatforgalmat titkosítás nélkül bonyolítják le, így ha valaki egy arra alkalmas helyen lehallgatja a hálózatunkat, akkor felhasználói nevekhez, jelszavakhoz is hozzájuthat. Ezért ahol lehet, érdemes az SSL-es szolgáltatásokat használni. Ezeknek a beállításához majd tanúsítványokat kell létrehozni, amivel a protokoll a titkosítást fogja végezni, de erről még szólunk a későbbiekben.
A ***Szakértő*** gombra kattintva további kapukat adhatunk meg elérésre, nevük a */etc/services* állományban találhatók meg.
Következő lépésben négy fontos beállítást végezhetünk el.
Az első az ***Útvonalkövetés engedélyezése***, amely lehetővé teszi, hogy a távoli gépről úgynevezett „ICMP time to live exceeded” csomagokat küldjünk a gépnek. Ez egyfelől hasznos, mert a `ping` parancs segítségével megállapíthatjuk, hogy a gép pillanatnyilag elérhető-e, ugyanakkor ez támadási felületet ad a szolgáltatásmegtagadásos (Denial of Service, DoS) támadásokhoz. DoS támadás alkalmával például ICMP csomagokkal árasztják el a kiszolgálót, ami – mivel a nagy terhelés miatt nem fogja tudni kiszolgálni a csomagokat – elérhetetlenné válik. Ezt a kockázatot mindenképpen figyelembe kell vennünk.
A második beállítási lehetőség a ***Forgalomtovábbítás és álcázás*** (NAT – Network Address Translation), amire akkor lesz szükségünk, ha a tűzfal mögött elhelyezett hálózatnak saját IP-tartományt akarunk adni, és az ügyfelek számára elérhetővé kívánjuk tenni a külső hálózatot. Ekkor a belső hálózati gépekről indított forgalom úgy fog látszani, mintha azt a kiszolgáló indította volna. Ez arra jó, hogy elrejtsük, hogy a kiszolgáló mögött helyi hálózatot hoztunk létre, valamint ennek segítségével takarékoskodni tudunk a rendelkezésünkre álló nyilvános IP-címekkel. Amennyiben otthoni hálózatot szeretnénk üzemeltetni, ennek a lehetőségnek használata szükséges például egy ADSL kapcsolat megosztására. (Figyelem, a szolgáltatók az ***Előfizetői szerződés*** keretében az ADSL és a kábeltévés internetelérések megosztását korlátozhatják, sőt akár tilthatják is!)
A harmadik beállítási lehetőség a ***Minden futó szolgáltatás védelme*** lehetőség. Ha ezt bekapcsoljuk, úgy az engedélyezett szolgáltatásokon kívül minden más szolgáltatáshoz érkező kérés vissza lesz utasítva – hasznos például DoS támadások ellen.
A negyedik pedig a már említett ***Védelem a belső hálózattal szemben***. Ekkor a belső hálózatról is csak a kijelölt szolgáltatások lesznek elérhetők.
Negyedik lépésben lehetőségünk van beállítani, hogy a tűzfal futása alatt milyen biztonsági események legyenek a helyi naplóba bejegyezve. Alapesetben a kényes csomagok kerülnek naplózásra, mind az elfogadott, mind az eldobott csomagok. Mivel ezek a bejegyzések a többi rendszerbejegyzéssel együtt a */var/log/messages* naplóállományba kerülnek, ez a naplózás eléggé meg fogja növelni a napló méretét, amely így elérheti

Kockázatelemzés adatbiztonsági szempontból

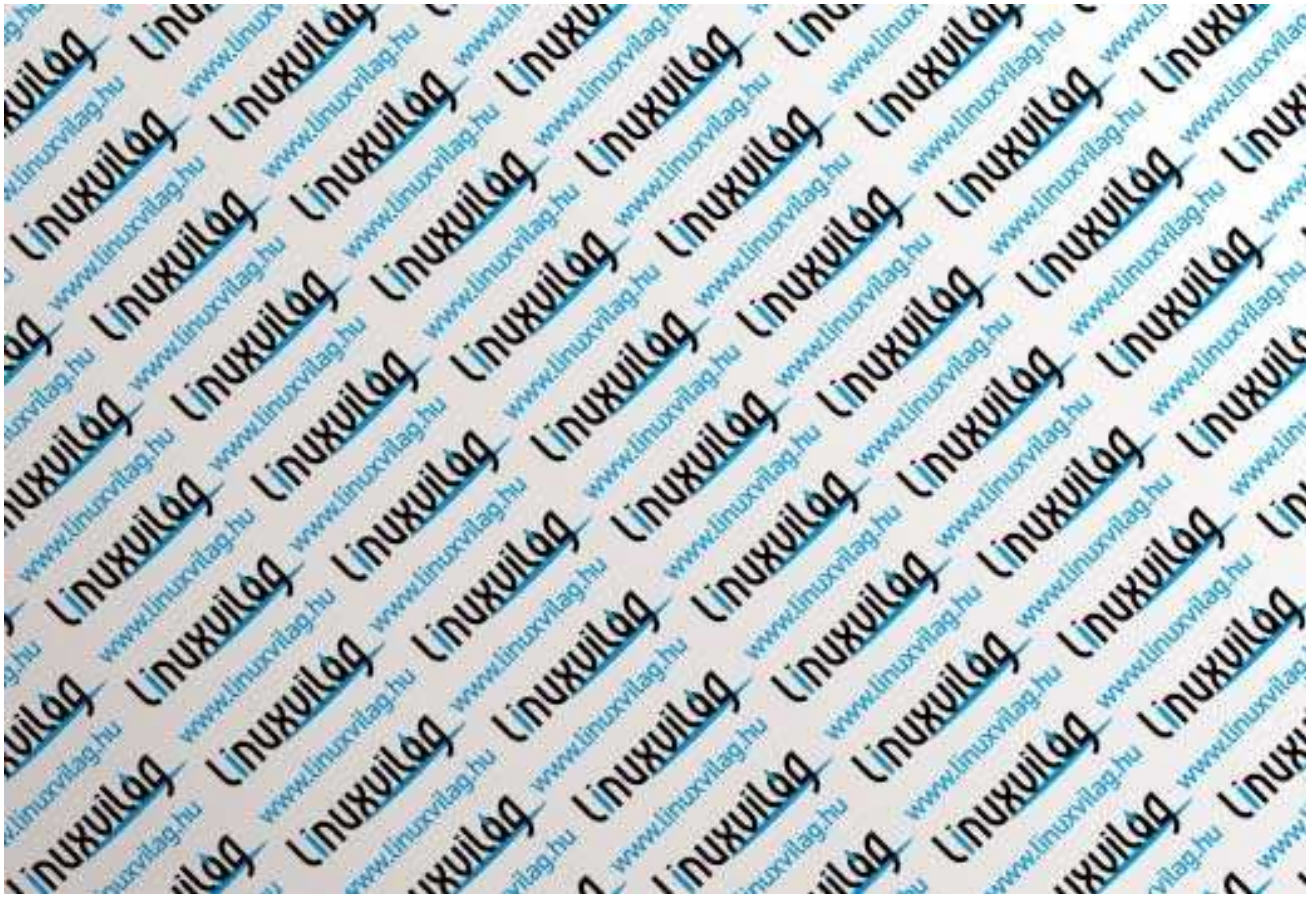
A biztonság az egyik legfontosabb dolog napjainkban – erre érdemes költeni, mert adataink elvesztése óriási gondokat idézhet elő, és ez ma már nem csak üzleti környezetben van így. Otthoni felhasználókat is fájdalmasan érinthet, mondjuk a családi fényképek elveszítése, amiket az elmúlt években az újonnan vásárolt digitális kamerával készítettek, vagy éppen egy egyetemi házi feladat, munkahelyi leírás eltűnése. Éppen ezért érdemes odafigyelni a biztonságra, költeni rá. És itt szokott felmerülni a kérdés, hogy mennyit is áldozzunk a biztonságra? A kevés adott esetben olyan, mintha semmit nem tettünk volna, a sok pedig felesleges kiadásként jelentkezik. A kulcsszó: felmérés és tervezés.
Megtehetjük, hogy az adatainkat többszörözött lemezrendszeren tároljuk, ahogy azt az előző cikkemben be is mutattam, megtehetjük, hogy rendszeres mentéseket készítünk, amit aztán a számítógép mellett tartunk. Mit ér mindez egy tűz alkalmával? Semmit. Elveszik a gép, megsemmisül a mentés.
Beállíthatunk akármilyen jó tűzfalat, készíthetünk szigorú felhasználói beállításokat – mindez semmit nem ér, ha a kiszolgáló fizikailag hozzáférhető és el lehet vinni a merevlemezeket.
Meg kell tehát teremteni a fizikai és logikai védelmet, az adatok biztonságos tárolásának feltételeit, és fel kell készülnünk egy esetleges rendszer-helyreállításra. Ehhez készíthetünk vésztervet, meghatározhatjuk, hogy ilyen esetben kinek mi a feladata. Amire pedig semmiképpen nem lehet felkészülni, olyan esetekre köthetünk biztosítást. Ezzel ismét eljutottunk oda, hogy határt kell szabnunk, mire és mennyit akarunk költeni. Körültekintés és alapos tervezés, ez a megfizethető biztonság kulcsa.

a napi 3–4 MB-ot is. Hibakeresési céllal bekapcsolhatjuk, hogy minden egyes csomag naplózásra kerüljön, de ez tényleg csak rövid idejű használatra ajánlott, mert így akár több 10 MB-os naplóállomány is előállhat. Ha minden beállítással végeztünk, akkor elindíthatjuk tűzfalunkat, így ezek után már nagyobb biztonságban érezhetjük magunkat.
Tűzfalunk összetettebb beállítása érdekében vessünk egy pillantást a ***Rendszer*** menüpont ***/etc/sysconfig*** szerkesztőmoduljára. A bal oldali fában a ***/network/firewall/SuSEfirewall2*** alatt találhatóak a SuSE-tűzfal további beállítási lehetőségei. Ezek módosítása csak haladó felhasználóknak ajánlott, mert csúnyán elszúrhatjuk vele a tűzfal beállításait. Ezekre a beállításokra egyszerű irodai, otthoni kiszolgálók esetén ritkán van szükség. Az egyik ilyen beállítási terület például az UDP-csomagok forgalmának a szabályozása, amelyre például DNS-kiszolgáló üzemeltetésekor vagy IPSEC titkosított csatorna létrehozásakor lehet szükség. Itt tudunk beállítani kaputovábbítást (port forward) a belső hálózat felé, és itt tudjuk beállítani a már említett három rétegű tűzfalszerkezetet is.

Naplóállományok kezelése

A SuSE 9.0 a rendszer futása alatt keletkező naplóeseményeket a */var/log/* könyvtárban tárolja. A könyvtárban további mappák is találhatók, amelyekben bizonyos telepített szolgáltatások külön naplóállományokat hoznak létre, például az Apache webkiszolgáló a ***http*** könyvtárat, a Samba fájl- és nyomtatókiszolgáló a ***samba*** könyvtárat. Rendszerünk biztonságos üzemeltetéséhez elengedhetetlen, hogy a naplóállományokat figyelemmel kísérjük, hiszen az üzemzavar vagy a támadási kísérlet ezekből derül ki egyértelműen. A */var/log* könyvtárban találhatók a ***messages*** és a ***mail*** állományok, ahová a rendszer és a levelező az elkészült naplóbejegyzéseket gyűjti.
Ezeknek az állományoknak az állandó szemmel tartása fontos, ugyanakkor elég kényelmetlen dolog. A SuSE Linuxban erre létezik egy `logdigest` nevű csomag, ami minden éjszaka kigyűjti az aznapi naplóbejegyzéseket és egy megadott elektronikus címre juttatja el őket. A `logdigest` csomag egyik legfőbb erénye, hogy a naplózott események megjelenítését szabályos kifejezésekkel saját magunk csoportosíthatjuk, így előtérbe helyezhetünk olyan bejegyzéseket, amelyek fontosak számunkra. A */etc/logdigest* könyvtárban találhatók az ***alarming*** és az ***ignore*** állományok. Az előbbibe tegyük az olyan kifejezéseket, amelyek fontos, riasztásértékű adatokkal szolgálnak, például a rendszermaghibára utaló bejegyzéseket. Az utóbbi állományba tegyük az olyan bejegyzéseket, amelyeket fontosnak tartunk menteni, de adott esetben nem akarunk minden nap átböngészni. A ***config*** állomány a `logdigest` beállítására szolgál. Itt adhatjuk meg, hogy mely naplóállományok kerüljenek feldolgozásra, mely felhasználó kapja meg a kiküldött elektronikus levelet, illetve, hogy egyéb rendszeradatok feldolgozásra kerüljenek-e.
Ha idáig eljutottunk, akkor van egy működő, alapjában véve biztonságosnak tekinthető kiszolgálónk.

Illés Viktor (viktor@ei.hu)
23 éves, a BME műszaki informatikus szakának hallgatója, mellette weblapokkal, linxos és windowsos rendszerekkel foglalkozik. Szabadidejét legszívesebben a szabadban tölti, teniszezik és kerékpározik.



Antik kinézet a Gimp segítségével

Próbáljuk ki ezeket a trükköket, és adjunk a képeinknek klasszikus, kortalan megjelenést!

A digitális fényképezőgépek egyik sokat reklámozott előnye a hagyományos darabokkal szemben – legalább is alacsony ISO értékeknél – a képek szemcsézettségének a hiánya. A fényképezés és papírképkészítés hagyományos feldolgozásának velejárója a szemcsézettség, ami különböző tényezők eredménye, de a legfőbb ok magában a film felépítésében (különböző rétegeiben) keresendő, ami nem más, mint fényérzékeny részecskék emulziója a celluloidon. Minél nagyobbak ezek a részecskék, a film annál érzékenyebb a fényre. Ez határozza meg a film ISO-besorolását. A nagyobb érzékenységnek ára van: minél nagyobbak a részecskék, annál könnyebben jelennek meg szemcsézettségként a kidolgozott filmen. Néhány nagyérzékenységű fekete-fehér film, amilyen például a Kodak T-Max 3200, különösen szemcsés – ezeket azok a fotósok részesítik előnyben, akik szeretik képeiken a szemcsézett megjelenést.

A digitális fotózásban az érzékelő zajaként kapjuk meg ennek a szemcsézettségnek a hasonmását. Digitális képeimen rendszerint csökkenteni akarom a zajt, de néha olyan homokszerű, szemcsézett filmhatást keltő látványt szeretnék elérni, amit a művészmozikban, az utcai fényképezésnél vagy a régi fotókon látni. Ebben a cikkben egy hatékony trükköt mutatok arra, hogy egyébként szemcsézetlen digitális fotóinkon hogyan utánozzuk ezt a szemcsézettséget.

A másik klasszikus fényképészeti fogás, amelyik a digitális képfeldolgozásban is helyet kapott, a szépiaszínezés. A szépiaszínezést eredetileg arra fejlesztették ki, hogy a korai fekete-fehér, ezüstalapú papírképek élettartamát növeljék. A folyamat hatására a képen lévő ezüstszemcsék ezüstszulfiddá alakulnak át, ami stabilabb az ezüstnél – így legalább lassul a kémiai károsodás feltartóztathatatlan folyamata. A XIX. század végén és a XX. elején az akkori, valószínűleg piszkosfehér színű fotópapírok gyenge minősége miatt ez kellemesebb látványt is nyújtott, mint az egyszerű fekete-fehér szín.

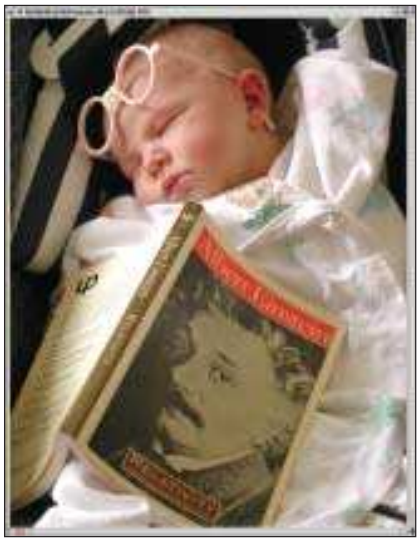
A hagyományos szépiaszínező folyamat során a kidolgozott képet óvatosan rázogatták egy fehérítő oldatban, míg az ezüst egy része vagy az összes a fémes állapotból át nem alakult. Az öblítés után a képet szépia-színezőfürdőbe áztatták, amíg csak a kifehérített ezüst el nem érte a kívánt árnyalatot. Végül a felesleges színező eltávolítása céljából a képet ismét lemosták, majd megszárították. A digitális képkezelésre való áttérés révén és az archív hatású színezőanyagokkal készülő festékek használatával a szépiaszínezés ma már szinte kizárólag esztétikai célokat szolgál.

Egy szépiafotó az időtlenség érzését árasztja magából. Egy olyan hatékony képfeldolgozó programmal, mint a Gimp és a megfelelő fotóminőségű tintasugaras nyomtatóval kitűnő minőségű szépiaszínű fotókat állíthatunk elő anélkül, hogy bűzös, mérgező vegyszerekbe kellene nyúlkálnunk.

A cikkben annak a vignette-hatásnak az elkészítési módját is meg fogom mutatni, amely az arcképfényképezés másik népszerű hozománya. A vignette-hatás kifejezést ebben az esetben nem a népszerű „fokozatosan elhalványuló, ovális képkivágás” értelemben használom, hanem olyan különleges lencsét vagy lencseárnyékolót értek alatta, amivel a kép keretének sarkai felé haladva fokozatos fényelhalványulás érhető el.

A film szemcsézettségének utánzása

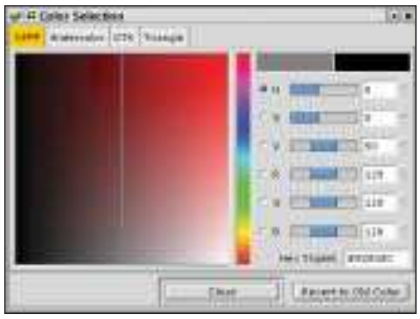
A Gimp menüinek nagy része a kép ablakában végrehajtott jobb egérkattintással érhető el. A következőkben ezt a műveletet JK-val (Jobb Klikk) rövidítem. Ha egy alkalmazandó Gimp-tevékenységet szeretnék leírni, zárójelek közé tett menüpontsorozatot vagy billentyűkombinációkat olvashatunk majd. Például egy kép megnyitásakor a (***JK>File>Open***) formát használom, ami annyit jelent, hogy a jobb egérgombbal kattintunk a kép ablakán, kiválasztjuk a ***File*** menüpontot, majd a megjelenő menüből az ***Open***-t. Ha célszerűbbnek tűnik a billentyűkombinációs megoldás, listát közlök azokról a gombokról, amiket meg kell nyomnunk. Például a kép másolására szolgáló (CTRL-C) jelentése: nyomjuk le és tartsuk nyomva a CONTROL billentyűt, s ezután nyomjuk meg a C-t.



1. kép A fotó tele lehetőségekkel

2. kép RGB-ből szürkeárnyalatosba és vissza RGB-be az átalakítás után

A kezdéshez feltételezem, hogy rendelkezünk egy, az *1. képen* láthatóhoz hasonlóan művészi lehetőségeket hordozó fényképpel, amit betöltöttünk a Gimpbe. Ha a hatást hozzám hasonlóan feketefehérben szeretnénk alkalmazni, akkor az első lépés az, hogy a színes képünket megfosztjuk a színeitől. Az átalakítás egy egyszerű módja a szürkeárnyalatos (grayscale) módba való átváltás (***JK>Image>Mode>Grayscale***), amely

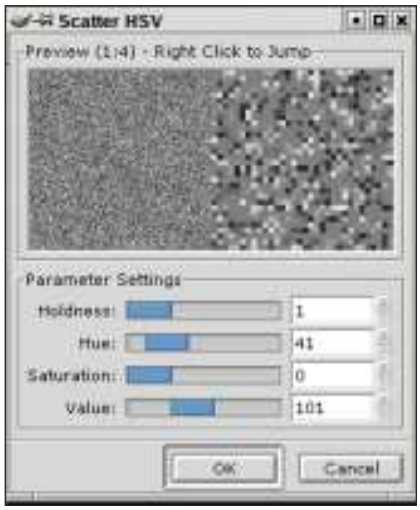


3. kép A színválasztó ablak

4. kép A Gimp eszköztára az előtér színének megváltoztatása után

5. kép A rétegablak a FilmGrain réteg hozzáadása után

a képek nagy részénél elfogadható eredményt ad. Néhány képnél másfajta megközelítéssel jobb eredményt érhetünk el. A rugalmasság végső határát a csatornakeverő (***Channel Mixer***) bővítmény egyéni, ***Monochrome*** (monokróm) beállításával érhetjük el



6. kép A Zaj/Elosztás (Noise/Scatter) HSV szűrő

(***JK>Filters>Colors>Channel Mixer***).
A fekete-fehérré történő átalakításról a ➲ http://mmmaybe.gimp.org/tutorials/Color2BW címen olvashatunk tájékoztatót. Ha megvan a megfelelően kinéző fekete-fehér képünk, továbbléphetünk. Ha a szabványos szürkeárnyalatos módba való átalakítást használtuk, ahogy én is tettem, a képet vissza kell alakítanunk RGB módba (***JK>Image>Mode>RGB***). A GIMP eszközdobozán kattintsunk az előtérszín mintájára, hogy megnyíljon a színkiválasztó párbeszédablak. A *3. képen* látható módon állítsuk be a piros, zöld és kék színeket: piros = 128, zöld = 128, kék = 128, és kattintsunk az ***OK*** gombra. Ez a kiválasztás az előtérszínt semleges szürkére változtatja az eszközdobozon, ahogy az a *4. képen* is látható. Most hívjuk elő a rétegek párbeszédablakát (CTRL-L), és kattintsunk az új réteg létrehozását szolgáló gombra, ami az *5. kép* alján bekarikázva látható. Adjuk neki a ***Film Grain*** nevet, és válasszuk ki az előtér színnel való kitöltésének a lehetőségét. Ha az ***OK***-ra kattintunk, akkor semmi mást nem fogunk látni, mint összefüggő szürke színt a képablakban, mivel a most létrehozott réteg eltakarja az alatta lévő kép rétegét. Most a réteg keverési eljárását (blending mode) változtassuk meg ***Overlay*** (lefedő) módra (az 5. kép a felső részén bekarikázva látható), ekkor újra láthatóvá válik a fénykép. A keverési mód a réteg tulajdonsága, amely azt határozza meg, hogy az adott réteg az összetett kép előállítása során milyen módszert használva keveredik az alatta lévő rétegekkel. Az ***Overlay*** (lefedő) mód használatakor a semleges szürkénél világosabb pontok világosítják a képet, az annál sötétebbek pedig sötétítik. A létrehozott réteg most teljes egészében semleges szürke, így a háttérben lévő képen semmilyen változás nem látható. A rétegek párbeszédablakán a ***Film Grain*** réteg kijelölt állapota mellett hívjuk elő a ***Scatter HSV*** (HSV-szóró) szűrőt (a képen jobb kattintással és a ***Filters>Noise>Scatter HSV*** kiválasztásával). Ez a szűrő zajmintázatot ad a semleges szürke réteghez, ami az alatta lévő képhez hozzáadódva szemcsézettnek tűnő látványt hoz létre. Íme a szűrő jellemzőinek egy nem túlságosan technikai jellegű leírása a szemcsézettség létrehozására:

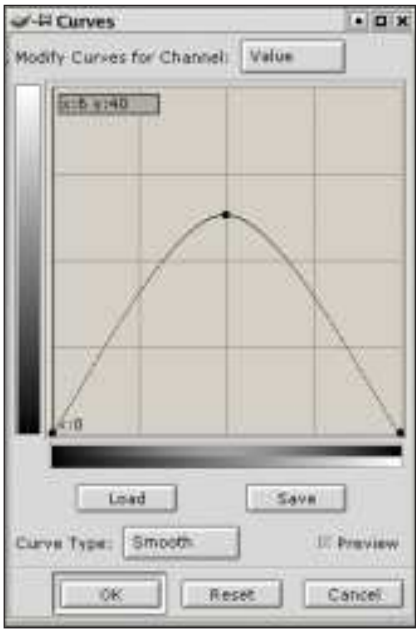
- ***Value*** (fényességérték): erre a jellemzőre úgy tekintsünk, mint a felület szemcsézettségének a vezérlőjére. Az értéket növelve növekszik a szemcsék kontrasztja és az alkalmazott zaj, ezáltal a szimulált szemcsék nagyobbaknak és sötétebbnek tűnnek.
- ***Hue*** (színárnyalat): ha a ***Saturation*** (telítettség) értéke nulla, akkor ennek a jellemzőnek csak a (véletlen) mintázatra van egy kis hatása. Próbálgassuk, amíg a tetszésünknek megfelelő mintázat ki nem alakul.
- ***Saturation*** (telítettség): ha nem akarunk színes szemcséket (például egy színes képen), akkor ezt 0 értékre kell állítanunk. Ha színes szemcséket akarunk, akkor ezzel a ***Hue*** beállításához hasonlóan kívánságunk szerint állíthatjuk be a színt.
- ***Holdness***: a szemcsésség intenzitásának és tagoltságának a finombeállítására szolgáló jellemző. Ezt a többi vezérlő beállítása után használjuk. Minél nagyobb az értéke, a szemcsézettség annál finomabb és kevésbé feltűnő.

Játsszunk addig ezekkel a jellemzőkkel, amíg az előnézeti képen egy érdekes szemcsemintázatot nem kapunk (ahogy a *6. képen* is látható), majd kattintsunk az ***OK***-ra. Amikor a szűrő befejezi a munkáját, a képünket már a szemcsézettséggel együtt kell látnunk. Ha nem tetszik a látvány, vonjuk vissza a műveletet (CTRL-Z) és más beállításokkal újra fogadjuk el (SHIFT-ALT-F). Valószínűleg nem érdemes túl sokat bíbelődni ezen a ponton, mert a következő lépések finomítják a szemcséket és tompítják a hatást.
Bár ennél a lépésnél a megfelelő eredmény birtokában akár meg is állhatnánk, a továbbfejlesztésre is akad lehetőségünk. Ha megvizsgálunk egy hagyományos, szemcsés fekete-fehér fényképet, látható, hogy a szemcsék a szín-

7. kép Rétegmaszk hozzáadása

tartomány közepén jobban láthatóak, és sokkal kevésbé tűnnek fel a sötét és a legvilágosabb részeken. Ehhez a keverési módot (***blending***) kell módosítanunk, hogy a háttérréteg világosabb és sötétebb részei kevésbé részesüljenek a szemcsézés hatásából. Amennyiben olvasták korábbi oktatóírásomat a rétegekkel és rétegmaszkokkal kapcsolatban (Fényképek kontrasztjának javítása a Gimp segítségével; Linuxvilág 2003. júliusi szám, talán nem felejtették el, hogy a rétegek keverését a rétegmaszkkal befolyásolhatjuk. Minél világosabb egy képpont a rétegmaszkon, annál kevésbé átlátszó a felette lévő réteg megfelelő pontja, és minél kevésbé átlátszó ez a képpont, annál erősebb a hatása, amikor az alatta lévő réteg pontjával egyesítésre kerül. A kívánt hatás eléréséhez szükséges fogás a háttérkép félig invertált másolatának előállítása a rétegmaszkon. Olvassunk csak tovább, mindjárt kiderül, mire gondolok.

A rétegek párbeszédablakán a jobb gombbal kattintsunk a ***Film Grain*** rétegen, és válasszuk az ***Add Layer Mask*** (rétegmaszk hozzáadása) menüpontot. Az ***Add Mask Options*** (maszk beállításai) párbeszédablakban válasszuk a ***White*** (fehér) színt, ami a teljes átlátszatlanságot jelenti, és kattintsunk az ***OK***-ra. Most kattintsunk a háttér rétegére (***Background layer***). Az egérrel fókuszáljunk a kép ablakára, jelöljünk ki mindent és másoljuk a vágólapra (ennek a leggyorsabb módja: CTRL-A, majd egy CTRL-C). Térjünk vissza a rétegek párbeszédablakához, és a ***Film Grain*** rétegben kattintsunk a rétegmaszk ikonjára; ez a réteg kicsinyített képe melletti fehér téglalap, ami a *7. képen* látható bekarikázva. Ezután az egérrel fókuszáljunk vissza a kép ablakára, és hajtsuk végre a beillesztést (CTRL-V). A rétegablakon a



8. kép A Curves ablak

9. kép Fekete-fehér változat mesterséges szemcsékkel

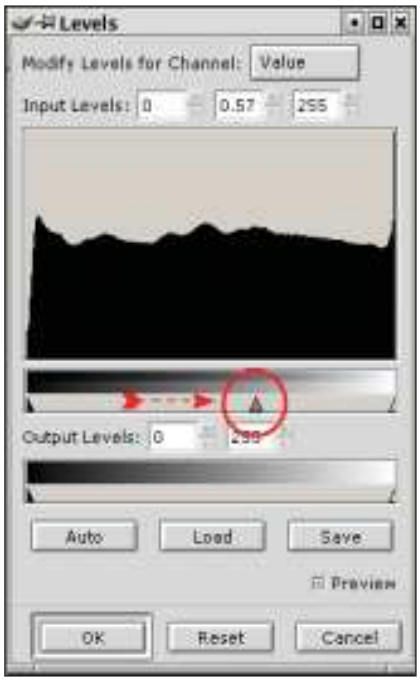
10. kép A rétegablak a Szépia réteg hozzáadása után

11. kép Szemcsés fénykép szépiaszínezéssel

12. kép A rétegablak

beillesztett kép rétegmaszkhoz történő rögzítése céljából kattintsunk a horgonygombra. Ezután a lépés után a rétegek ablakának a 7. képhez hasonló kinézetet kell mutatnia.

A rétegmaszkot a most végrehajtott beillesztés után hagyjuk kiválasztva, és a kép ablakán végrehajtott jobb kattintással, majd az ***Image>Colors>Curves*** menüpont kiválasztásával hívjuk elő a ***Curves*** (grafikonok) párbeszédablakot. Kattintsunk az egyenes grafikon közepére egy vezérlőpont hozzáadásához. Ezután a *8. képen* mutatott módon húzzuk le a jobb oldali (fenti) végpontot a jobb alsó sarokba. Most hajtjuk végre a rétegmaszk félinvertálását minden nagy intenzitású rész sötétté változtatásával, így a kép legvilágosabb részei a középtónusú részek lesznek. Esetleg a középső pontot érdemes egy picit feljebb húzni, ahogy én is tettem, hogy jobban kiemeljük a közepes világosságú pontok fényességét – eközben figyeljük a kép ablakát, amelyen látható ennek a kiemelésnek a hatása. Amikor mind-



13. kép A vignette finomadagolása

14. kép Az elkészült kép

ezzel elkészültünk, kattintsunk az *OK*-ra – látható különbséget kell tapasztalnunk a szemcsék megjelenésében az árnyékos és fényes részeken. Összességében némileg enyhült a szemcsésség hatása, ez az oka annak, hogy az előző lépésben nem akartuk túl finomra beállítani a látványt.
Ha tisztábban szeretnénk látni a rétegmaszk szerepét, tartsuk lenyomva a Ctrl billentyűt, majd a rétegek ablakában kattintsunk a rétegmaszk ikonjára. Egy kis piros körvonalnak kell az ikon körül megjelennie, és a kép ablakában megjelenik a rétegmaszk hatása nélküli képösszetétel. Újabb kattintás az ikonon a Ctrl billentyű nyomva tartása mellett, s ezzel újra bekapcsoljuk a rétegmaszkot. Utolsó lépésként fontoljuk meg az elmosódás (Gaussian Blur) alkalmazását a szemcsék enyhe kisimítására. Kiválasztásához a rétegek ablakában kattintsunk a szemcsés ikonra, majd a kép ablakán végezzünk egy ***JK>Filters>Blur>IIR Gaussian Blur*** kiválasztást. Legyünk óvatosak, mert ha az elkenés sugaraként nem egy nagyon kicsi értéket választunk, akkor hatástalanítjuk a korábban hozzáadott szemcsés hatást, és az eredmény egy zajos, de nem szemcsés kép lesz. Az 1 és 3 közötti értéknek megfelelőnek kell lennie, amennyiben a ***Scatter HSV*** szűrőben a ***Value*** értékére elég magas számot állítottunk be. Ha a szemcsékkel enélkül is elégedettek vagyunk, akkor ezt a lépést kihagyatjuk. A *9. képen* elért eredményt a *6. képen* látható beállításokkal és 1 képpontos Gauss-elmosási sugárral kaptam.

Szépiaszínezés

Most, hogy már van egy művészi hatású fekete-fehér fotónk, adjunk hozzá még egy kis szépiaszínezést, hogy még klasszikusabb és időtlenebb megjelenést érjünk el. A rétegek párbeszédablakában legyen kiválasztva a ***Film Grain*** réteg, majd az ablak alján kattintsunk a réteg megkétszerezésére: ezzel a teljes réteg lemásolódik, beleértve a rétegmaszkot is. Kattintsunk kétszer a ***Film Grain*** réteg másolatára, és nevezzük át ***Sepia Tone***-ra. Változtassuk az új réteg keverési módját (***blend mode***) ***Color*** (szín) értékre; ezzel a ***Sepia Tone*** réteg színfokozat (***hue***) és telítettség (***saturation***) értékei az alatta lévő rétegek fényerősségével fognak rendelkezni. Végül magának a rétegnek a kijelöléséhez a ***Sepia Tone*** rétegen kattintsunk a réteg kicsinyített ikonjára. A színválasztó párbeszédablak megjelenítéséhez a Gimp fő eszközdobozában kattintsunk az előtérszín mintájára, ahogy korábban is tettük, és állítsuk be a színt (piros = 162, zöld = 138, kék = 101), majd kattintsunk az ***OK***-ra.
A színmintának barnás árnyalatúvá kell válnia. Ha egyszer elsajátítjuk ezt a technikát, más színekkel is kísérletezhetünk, de ez jó kiindulási alap. Válasszuk ki a kitöltő eszközt (***Fill***) a Gimp eszköztárának festőcsoportjából, és fókuszáljunk az egérrel a kép ablakára. Válasszunk ki mindent (Ctrl-A), majd az új színnel való feltöltéshez kattintsunk egyszer az ablakra. A réteg párbeszédablaknak a *10. képhez* hasonló képet kell mutatnia, a képnek pedig egy általános szépiaárnyalatot kell felvennie, ahogy a *11. képen* is látható.
Nem véletlen, hogy a ***Sepia Tone*** rétegen is ugyanazt a rétegmaszkot használjuk, mint a ***Film*** Grain rétegen. A hagyományos vegyszeres, sötétkamrás eljárás használatakor is az figyelhető meg, hogy a szépiaárnyalat a közepes tónusokban jelenik meg leginkább, a sötétebb és világosabb részek kevésbé barnának tűnnek. Ennek a szépiaszínezésre gyakorolt hatása kipróbálásához ugyanazt a Ctrl-kattintás fogást használhatjuk, amit korábban is. Ha a rétegmaszk használata nélküli erősebb szépiaárnyalat jobban tetszik, egyszerűen kattintsunk a jobb gombbal a rétegen, és válasszuk a rétegmaszk törlését (***Delete Layer Mask***).

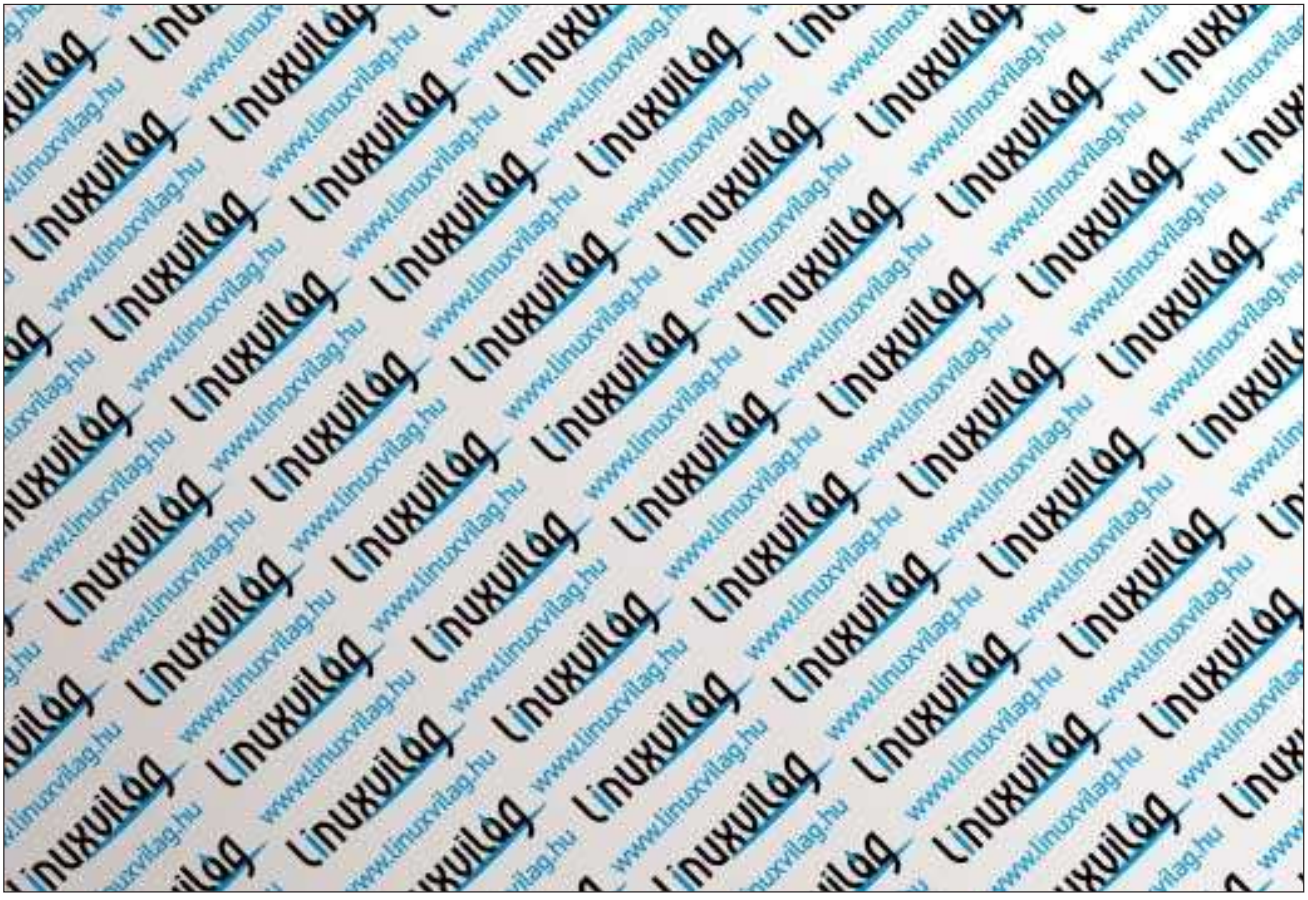
A vignette-hatás

Itt az ideje, hogy a végső hatás hozzáadásával igazán kitűnővé tegyük a képet. Egy vignette-hatást fogunk a képhez adni, ami a pontfényhatás egy változata, amely a szemlélő tekintetét azzal vonzza a kép fő témájára, hogy attól távolodva a kép sugárirányban enyhén elsötétül. Ha megfelelő finomsággal alkalmazzuk ezt a hatást, a nézőnek nem is szabad észrevennie a beavatkozást. Ez az eljárás különösen portrék esetén hatásos.

A rétegek párbeszédablakában válasszuk ki a háttérréteget (***Background layer***), és kattintsunk az ablak alján lévő kettőzőgombra. Kattintsunk kétszer a másolaton, és nevezzük át ***Vignette***-re. Jobb kattintás a ***Vignette*** rétegen, és adjunk hozzá egy rétegmaszkot fehér színnel (teljesen átlátszatlan).
A Gimp eszköztárán az alapértelmezett fekete előtérszín visszaállításához kattintsunk a fekete-fehér színminta kicsinyített képére. Kattintsunk kétszer a ***Blend (Gradient)*** (keverék) eszközre a kiválasztásához, és nyissuk meg a ***Tool Options*** (eszközbeállítások) párbeszédablakot. Itt a gradiens típusának válasszuk a ***Radial***-t. Térjünk vissza a kép ablakához, és kattintsunk annak a területnek a közepére, amelyre a vignette-hatást irányítani szeretnénk, ezután húzzuk kifelé a sarok felé és engedjük el. Ebben az esetben én a gyermek állára kattintottam és a jobb felső sarok felé húztam. Az egyetlen változás, amit látni fogunk, a rétegmaszk ikonján megjelenő sugárirányú átmenet, ahogy az a *12. képen* látható.
Térjünk vissza a réteg ablakon a ***Vignette*** réteghez, és kattintsunk a réteg ikonjára, hogy az legyen a rétegmaszk helyett kiválasztva. Most menjünk a kép ablakára, és hívjuk elő a ***Levels*** (szintek) párbeszédablakot (***JK>Image>Colors>Levels***). Mozgassuk a középső (szürke) csúszkát egy kicsit jobbra, majd engedjük el, ahogy a *13. képen* látható. Ellenőrizzük a hatást a kép ablakán, és addig állítgassuk a csúszkát, az eredménnyel amíg elégedettek nem leszünk, ekkor kattintsunk az ***OK***-ra. A *14. képen* látható a végső kép; a vignette-hatás a tekintetet a gyermek arcára vonzza és kellemes sötét kontrasztot ad a képnek. Ha egy más fényhatás létrehozásához az átmenet újrarajzolására lenne szükség, egyszerűen válasszuk ki a ***Vignette*** rétegmaszkját, kattintsunk a ***Blend*** eszközre, és próbáljuk újra a képen történő beállítást: az új átmenet fel fogja váltani a régit.
Itt az ideje, hogy hátradőljünk, és megszemléljük alkotásunkat. A réteg láthatóságának be- és kikapcsolásához próbáljunk a különböző rétegek mellett látható szemre kattintani, ezzel egyszerűen tanulmányozhatjuk a leírt különböző hatások látványát. Ha a kép valamelyik hatás nélkül jobban tetszik, a réteg kiválasztásával és a szemetesre való kattintással törölhetjük az adott réteget. Ha a fényképen további szerkesztéseket szeretnénk végrehajtani, talán érdemes a képet most más néven menteni vagy megkettőzni (CTRL-DBMS), vaéamint a rétegeket egyesíteni (***JK>Layers>Flatten Image***).
Jó ötlet ellenőrzési pontokat iktatni az eredménybe, hogy egy műveletet újra lehessen kezdeni, ha valami félresikerülne. Kísérletezhetünk a színfokozat és telítettség megváltoztatásával, módosíthatjuk a kontraszt szintjeit és görbéit, vagy más szerkesztést hajthatunk végre az egyesített változaton.
Remélem, hogy ezzel a cikkel sikerült éreztetnem a rétegekben rejlő hatalmas lehetőségeket. Bár a képfeldolgozás rétegszerű szemlélete több memóriát igényel a géptől, a rugalmasságban ez bőven megtérül – vagyis tegyünk még egy kis memóriát a linuxos gépünkbe, és alkossunk minél szebb antik mesterműveket a rétegek segítségével!

Linux Journal 2003. október, 114. szám

Eric Jeschke
(eric@redskiesatnight.com)
Az Indiana Egyetemen számítástechnikából szerzett PhD fokozatot. Hawaii-on él a feleségével és gyermekeivel.



Bevezetés a szabályos kifejezések használatába (1. rész)

A Linuxot úgy is használhatjuk, mint bármelyik más grafikus operációs rendszert. Aki azonban „érezni akarja az erőt”, annak meg kell tanulnia varázsolni a kódokkal!

Számtalanszor olvashattuk már azt az állítást, amely szerint a Unix operációs rendszert programozók írták programozóknak. Hogy ez mennyire igaz, azt a Unix- és a Linux-rendszerek számos tulajdonsága és viselkedésformája is alátámasztja. Az „igazi” linuxos például komoly munka közben nem az egérrel kattogtat, hanem a közönséges földi halandók számára igencsak meglepő dolgokat irkál a parancssorba. Néha annyira furcsákat, hogy egyes esztéták szerint azokat látván az ember önkéntelenül és „elzöldült képpel kapott a légibetegség esetén használatos zacskó után”. (Az idézet ***David Gelernter*** „Ami működik, az csodálatos” című könyvéből származik, amely alcíme szerint a technika esztétikájáról szól.)

Lássuk csak, mi zavarhatja meg ennyire ezeknek az ártatlan embereknek az egyensúlyérzékét. Az első furcsaság számukra valószínűleg az lehet, hogy Linux alatt a feladatok megoldására szolgáló – néha valóban kissé ördögi – gépezeteket maguknak kell apró építőelemekből összeállítaniuk. Ezt hívjuk héjprogramozásnak, amiről egy egész cikksorozatot olvashat az, aki végiglapozza a Linuxvilág előző néhány számát. Abból az egyszerű felismerésből kiindulva, hogy az emberek többsége szeret legózni, nem valószínű, hogy a fent említett gondot egyedül a héjprogramozás okozná – kell itt lennie még valaminek.

A másik „nagy zavarkeltő” a szabályos kifejezések használata lehet. Ezekkel gyakorlatilag tetszőlegesen bonyolult, a keresés-sel, helyettesítéssel vagy más ezekhez hasonló szövegfeldolgozási műveletekkel kapcsolatos utasítás, algoritmus tömören – általában egyetlen sorban – leírható. Az általánosságnak azonban ára van: egy szabályos kifejezésekből felépített, logikai értelemben „gyönyörű” feldolgozási utasítás a kívülállók és a kezdő felhasználók számára rémálom. Az ember az első pillanatban általában azt se tudja, melyik végén kezdje el olvasni.

A nehézségeket csak fokozza, hogy a szabályos kifejezésekre támaszkodó segédprogramok (itt elsősorban a `grep`, `sed` és `awk` nevű eszközökről van szó) dokumentációja tartalmazza ugyan a megfelelő útmutatásokat, de egyáltalán nem olyan stílusban, ami igyekezne a kezdőket átsegíteni a dolog nehezén. Ezek a leírások inkább csak azok számára nyújtanak lexikonszerű segítséget, akik a dolog lényegével már tisztában vannak, vagyis legalább nagyjából tudják, hogy mit keresnek.

Az írásainkban bemutatott példák ezzel szemben fokozatosan nehezednek, és kifejezetten a gyakorlatra összpontosítva igyekeznek bevezetni az olvasót a szabályos kifejezések használatába.

A szabályos kifejezések kézzelfogható elemei

Gyakorlatias szempontból közelítve meg a kérdést a szabályos kifejezések arra szolgálnak, hogy a segítségükkel egy szöveg bizonyos tulajdonságokkal rendelkező részeit azonosíthassuk. Az azonosítás célja lehet keresés (`grep`) vagy valamilyen feldolgozás (`sed` és `awk`).

Tekintve, hogy egy szöveg legkisebb építőeleme egyetlen karakter, a legegyszerűbb szabályos kifejezés is egyetlen karakterből áll, ami – nem túl meglepő módon – önmagára illeszkedik. Amennyiben tehát végrehajtjuk a

```
cat szöveg.txt | grep "a"
```

műveletet, akkor a keresésre szolgáló és szabályos kifejezésekkel vezérelhető `grep` parancs a szövegnek minden olyan sorát megjeleníti, amelyben bárhol legalább egy „a” betű szerepel.

A következő egyszerű szabály szerint ha két szabályos kifejezést egymás mellé írunk, akkor ismét szabályos kifejezést kapunk. Ennek szellemében a

```
cat szöveg.txt | grep "ablak"
```

parancs az összes olyan sort megjeleníti, amelyben legalább egyszer szerepel az „ablak” szó. Ezen a ponton fontos megjegyezni, hogy a szabályos kifejezés illeszkedésének nem feltétele az, hogy a keresett szó magában álljon, vagyis az „ablak” illeszkedik az „ablaktörlő” és a „vakablak” szavakat tartalmazó sorokra is.

A szabályos kifejezések elvont elemei

Előfordulnak olyan esetek, amikor nem ennyire egyértelműen egyetlen szó vagy karakterlánc formájában írható le az a dolog, amit keresünk. Például előfordulhat, hogy kizárólag „a” betűvel kezdődő, ötbetűs szavakat akarunk keresni. Ez első látásra ugyan egyáltalán nem tűnik bonyolultnak, de mint hamarosan látni fogjuk, arra kiválóan alkalmas, hogy rajta keresztül bemutassuk a jól használható szabályos kifejezések megfogalmazásához szükséges gondolatmenetet.

Először is minden olyan sajátságot fel kell sorolnunk, ami a keresendő objektumot egyértelműen azonosítja. Kis „a” betűvel kezdődő pontosan ötbetűs szavakat keresünk, vagyis olyan karakterláncokat, amelyek első eleme az „a” betű, amit pontosan négy másik, amúgy tetszőleges karakter követ.

A szabályos kifejezésekben az „egy darab akármi” jele a pont (.). Ez tehát pontosan egy, de tetszőleges karakterre illeszkedik, ami esetünkben azt jelenti, hogy az „a betűvel kezdődő ötbetűs szó” fogalma a

```
"a...."
```

szabályos kifejezéssel írható le. Vagy mégsem? Hozzunk létre egy szövegfájlt a következő tartalommal:

```
Az almák édesek.
A vadalma fanyar.
```

És most nézzük, melyik sorra illeszkedik a fenti szabályos kifejezés:

```
cat szöveg.txt | grep "a...."
```

Mindkét sor megjelent a `grep` kimenetén, vagyis mindkettőre illeszkedett a logikai minta – pedig a másodikban sem ötbetűs,

A szabályos kifejezések alapelemei

Elem	Leírás.
karakter	A szabályos kifejezések legkisebb építőeleme. Bármely karakter önmagára illeszkedik.
\karakter	Olyan karakter, amely a szabályos kifejezésekben absztrakt jelentéssel is rendelkezik (pl.: ^ , $, [ ), de az adott helyen literális értelemben kívánjuk használni.
[lista]	Egyetlen karakter, amely a megadott lista bármely eleme lehet.
[^ lista]	Egyetlen karakter, amely a listában felsoroltakon kívül bármi lehet.
[a-z]	Bármilyen kisbetű (kivéve a magyar ékezetes karaktereket).
[A-Z]	Bármilyen nagybetű (kivéve a magyar ékezetes karaktereket).
[a-zA-Z]	Bármilyen betű (kivéve a magyar ékezetes karaktereket).
[0-9]	Bármilyen számjegy.
^	Sorkezdet.
$	Sorvég.

sem „a” betűvel kezdődő szó nincs. Ugyanakkor a „vadalma” szóban két olyan részlet is van („adalm” és „alma”), amely „a” betűvel kezdődik és azt négy karakter követi. Világos tehát, hogy nem elég egyértelműen fogalmaztuk meg a keresési feltételt.
Ha egy szó valóban pontosan ötbetűs, akkor előtte és utána is szóköznek kell állnia. Az előbbinél tehát sokkal alkalmasabb a

```
cat szöveg.txt | grep " a.... "
```

szabályos kifejezés; ez már valóban csak a megfelelő sort jeleníti meg a kimeneten.

További bonyodalmak

Eredeti célunknál maradva finomítsuk tovább egy kicsit a keresés szempontjait! Például meg lehet-e azt oldani, hogyha egy ötbetűs, „a” betűvel kezdődő szó a mondat elején található, és ezért nagybetűvel kezdődik, akkor is rátaláljon a `grep`? Vagy mi történjen akkor, ha egy ilyen szó a mondat végére kerül, és így nem szóköz, hanem pont van utána?
Mindezek kipróbálására az előbb használt példaszöveget egészítsük ki három újabb sorral:

```
Az almák édesek.
A vadalma fanyar.
A kép címe: Almák és körték.
És megint almák.
Langyos a sör.
```

Ha az illesztendő minta egy pontján vagylagosan többféle lehetőséget akarunk megengedni, szögletes zárójelek között kell felsorolnunk őket. Az `"[abcd]"` szabályos kifejezés tehát az ábécé első négy betűje közül bármelyikre illeszkedik, de továbbra is csupán egyetlen karakternek felel meg, tehát vagy egyetlen „a” betűnek, vagy egyetlen „b” betűnek és így tovább. Ha az összes kisbetűre így akarunk hivatkozni, azt az `"[a-z]"` formájú kifejezéssel tehetjük meg. Természetesen az összes nagybetűre is hivatkozhatunk ilyen módon, az `"[a-zA-Z]"` szabályos kifejezés pedig valamennyi betűnek megfelel.
Ennek fényében ha az ötbetűs szó elején a kis és nagy „a” betűt is meg akarjuk engedni, a végén pedig a szóközt és a mondatvégi pontot is, akkor a következő szabályos kifejezést szükséges használnunk:

```
cat szöveg.txt | grep " [Aa]....[ \.]"
```

A kifejezés végén a pont előtt egy \ is szerepel. Erre azért van szükség, mert ennek a karakternek a szabályos kifejezésekben különleges jelentése van (lásd fent), és valahogyan jeleznünk kell, hogy itt betű szerinti értelemben (literálisan) használjuk, vagyis tényleg pontot keresünk.
Most már minden nagyon szép és jó, csak egy gond van még: langyos a sör…
Ez a mondat is megjelent a kimeneten, pedig nincs benne öt betűből álló, „a” betűvel kezdődő szó. Vagy mégis? Nos, ami azt illeti, van benne egy olyan rész, hogy „ a sör. ”. Ez pont öt karakter, amit egy szóköz előz meg és egy pont követ. És mivel ráadásul az „a” betű az első, illeszkedett rá a megadott szabályos kifejezés. Igaz ugyan, hogy a közbenső betűk között van egy szóköz is, de azt nem is kötöttük ki eddig semmilyen formában, hogy az ilyen „ötbetűs” karakterláncokat nem szeretjük. Oldjuk meg tehát ezt is:

```
cat szöveg.txt | grep " [Aa][^0-9 ][^0-9 ]
➥[^0-9 ][^0-9 ][ \.]"
```

(Khm... Hova is tettem azt a zacskót?)
A `[^0-9 ]` felsorolás természetesen a számjegyeket és a szóközt jelenti. A nyitó szögletes zárójelet közvetlenül követő ^ karakter ugyanakkor a tagadás jele, vagyis ez a kifejezés mindenre illeszkedik, ami nem szám és nem szóköz.
Az olvasó most nyilván azt kérdezi, hogy minek kellett ezt ilyen faramuci módon megfogalmazni. Miért nem írtuk egyszerűen azt, hogy `"[a-zA-Z]"`? Nos, van itt egy kis gond. Az a helyzet, hogy a Unix világában a magyar ékezetes betűk nem számítanak betűnek, vagyis nem elemei az imént említett halmaznak. Viszont szerencsére nem is számok, így az általunk alkalmazott körmönfont megfogalmazás éppen megfelelő.
A következő kérdés az lesz, hogy feltétlenül le kell-e írni az illető halmazt négyszer egymás után, ha négy ilyen karaktert keresünk? És mi van, ha nem négyet, hanem százhuszonhetet akarunk azonosítani? Természetesen létezik egyszerűbb írásmód, de erről egy kicsit később lesz szó.

További elemek

A szabályos kifejezések jelkészletében létezik egy-egy elem a sor elejének és végének azonosítására is. A sorkezdetet a ^, a sorvéget a $ karakter jelöli. Ha tehát egy szabályos kifejezés elé odaírjuk a ^ jelet, azzal kikötjük, hogy az illesztést mindenképpen a sor elején akarjuk elvégezni. Hasonlóan a $ a sor végén illeszt. (Igen, egy kissé zavaró, hogy a ^ két különböző dolgot is jelent, de ennek nyilván történeti okai vannak.)
Fontos hangsúlyozni, hogy a sorkezdet- és sorvégjeleknek önmagukban is van értelme, bár karaktereket tartalmazó felsorolásokban azért nem szerepelhetnek. Így a

```
cat szöveg.txt | grep "^$" | wc -l
```

feldolgozási sor például megszámolja, hogy a bemenő szövegben hány teljesen üres sor van, a

```
cat szöveg.txt | grep -v "^$"
```

pedig az ilyen sorokat kiszűri a bemenetből, és csak a tömörített szöveget jeleníti meg. (A `grep -v` kapcsolója a tagadás jele, vagyis a megadott szabályos kifejezésnek nem megfelelő sorok fognak megjelenni a kimeneten.)
Az imént érintőlegesen említettem ugyan, de azért nem árt még egyszer hangsúlyozni, hogyha egy keresendő karakter szabályos kifejezés részeként is előfordulhat, akkor `\` karakternek kell megelőznie, jelezvén, hogy literálisan, vagyis betű szerinti értelmében használjuk. Ha tehát például a befejezetlen mondatot jelző három egymás követő pontot akarunk keresni egy szövegben, akkor a

```
cat szöveg.txt | grep "\.\.\."
```

kifejezést kell használnunk. Ha lefelejtjük a három `\` karaktert, akkor a `grep` három tetszőleges karaktert fog keresni, és így minden olyan sort megjelenít, amelyik legalább három karaktert tartalmaz.

Néhány példa

A szabályos kifejezésekről eddig tanultakat táblázatban foglaltam össze. Zárásként nézzünk meg néhány olyan gyakorlatias példát, amelyeket már ennyi ismeret birtokában is képesek vagyunk megoldani.
Ha sokat gépelünk, néha megesik, hogy elfelejtjük kitenni a mondat végére a pontot. Ráadásul az ilyen hibát utólag is elég nehéz észrevenni. Ugyanakkor a következő egészen egyszerű szabályos kifejezés segítségével kiszűrhetők azok a sorok, amelyekben ezt a hibát valószínűleg elkövettük:

```
cat szöveg.txt | grep "[a-z] [A-Z]"
```

Láthatóan azt használjuk ki, hogyha a mondat végéről hiányzik a pont, akkor egy kisbetűvel végződő szót egy szóköz után rögtön egy nagybetűs követ. A módszer ugyan nem tökéletes, hiszen ha tulajdonnév szerepel egy mondatban, akkor is lesz benne ilyen részlet. Ugyanakkor a valóban hibás sorok biztosan fönnakadnak a rostán. (Ha azt akarjuk, hogy ékezetes betűkkel is működjön a szűrés, akkor a korábban említett trükköt kell alkalmaznunk.)
Hasonlóan gyakori hiba a „z” és „y” felcserélése, ami „ay” és „egz” karakterláncok felbukkanását vonja maga után. Ezeket a következő egyszerű `sed` programmal cserélhetjük „magyar megfelelőjükre”:

```
cat szöveg.txt |\
sed "s/ ay / az /g
     s/ Ay / Az /g
     s/ egz / egy /g
     s/ Egz / Egy /g"
```

A csere miatt most a kis és a nagy kezdőbetűs alakok összevonására nem használhattuk a `" [Aa]y "`, illetve a `" [Ee]gz "` szabályos kifejezéseket. Bár már így is négy különböző cserét írtunk elő, a módszer mégsem tökéletes, mert nem fogja érzékelni a sor elején, illetve a sor végén található hibásan írt alakokat. (Ezek előtt, illetve mögött nem lesz meg az előírt szóköz.)
A teljes megoldás tehát – legalábbis jelenlegi ismereteink birtokában – tizenkét cseréből állna.
Végül nézzünk egy olyan példát, amely kigyűjti egy szövegből az elektronikus levélcímeket. Ennek a legegyszerűbb megvalósítása az lenne, ha egyszerűen a `grep` segítségével a @ karakterre végeznénk keresést. Ugyanakkor ez a megoldás eléggé spártai kinézetű kimenetet eredményezne, hiszen a `grep` találat esetén csak egész sorokat tud megjeleníteni. Ha tehát a címek nem külön sorokban szerepelnek, akkor az utómunkálatokat nekünk magunknak kellene elvégezni.
Használjuk ki tehát, hogy az `awk` soron belül is képes vizsgálódni, és ugyanúgy ismeri a szabályos kifejezéseket, mint a `grep`. Ha feltételezhető, hogy a feldolgozandó szövegben nemcsak elektronikus levélcímekben szerepel a @ karakter, akkor egy kifinomultabb keresési feltételt is használhatunk: egy levelezési címben a @ előtt és után is betűnek kell lennie. Mindezek fényében a következő egysorost használhatjuk a válogatás elvégzésére:

```
cat szöveg.txt | awk '/@/ {for(i=1;i<=NF;i++)
if($i~["[a-zA-Z]@[a-zA-Z]") print $i }'
```

A feldolgozási program előtt két / jel között megadott szabályos kifejezés a sorok címzésére szolgál. A program tehát eleve csak azokat a sorokat fogja vizsgálni, amelyekben – egyelőre bármilyen formában – szerepel a @ karakter. Az ilyen sorokon mezőnként végigmegy, és megvizsgálja, hogy melyikre illeszkedik a `"[a-zA-Z]@[a-zA-Z]"` szabályos kifejezés. (Az `awk` nyelvben az illeszkedik jele a ~ karakter.) Csak azokat a mezőket jeleníti meg, amelyeknél ez a feltétel teljesül.
Hasonlóan érdekes feladat egy szövegből a pénzösszegek kigyűjtése, például könyvelési célzattal. Az árakat arról lehet felismerni, hogy egy számot egy, a „Ft” karakterpárost is tartalmazó szó követ. Ez utóbbi ragozott alak is lehet (például „500 Ft-ba került”), illetve eshet a mondat végére is. Lássuk a válogatást végző programot:

```
#!/bin/sh
cat $1 | awk \
'/Ft/  {
  elozo=$1
  for(i=2;i<=NF;i++)
  {
    if($i~"Ft" && elozo~"[0-9]") print elozo, "Ft"
    elozo=$i
  }
}'
```

Tekintettel a feldolgozás összetettségére, ezt a példát már célszerűbb volt a fent látható módon héjprogram formájában megvalósítani. Ismét csak azokkal a sorokkal foglalkozunk, amelyekben legalább egyszer előfordul valahol a „Ft” karakterkettős. A feldolgozás lényege az, hogy a sor mezőit páronként vizsgáljuk. Ha a pillanatnyi mezőben felismerjük a „Ft” részletet (`$i~"Ft"`), az előző mezőben pedig van számjegy (`elozo~"[0-9]"`), akkor kiíratjuk az előző mezőt. A módszer egyetlen hibája, hogy ha az ár éppen egy sor végére, a „Ft” jel pedig a következő sor elejére kerül, akkor nem működik.
Némi további programozással természetesen ez is megoldható.

Büki 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.

Elektronikuslevél-beépítés

Tartsuk ébren weboldalunk felhasználóinak az érdeklődését: figyelmeztessük őket elektronikus levélben a számukra érdekes hírekre és vitákra!

Több mint húsz éve használom a számítógépemet az embertársaimmal való kapcsolattartásra. Kezdetben csak alkalmanként írogattam a helyi hirdetőtáblára, mindez mára személyes és szakmai életem elválaszthatatlan részévé vált. Ebben a hónapban – miközben folytatom a kicsomagolást új chicagói otthonomban, és küzdök a felmerülő hibákkal, amelyek megnehezítik, hogy folytassam a Bricolage-ról szóló cikksorozatomat – a jelenkor néhány, az elektronikus levelekkel kapcsolatos kérdésébe pillantok bele. Írásomban a web-, illetve a levelezésegyesítésről, illetve a levelezés és az adatbázis egyesítéséről, sőt még az SMTP-szintű levélszemét elleni harcról is szó esik.

A web és az elektronikus levél egyesítése

Ma már valóban nehéz olyan nagy és korszerű weblapot találni, amelyen nincs megoldva felhasználói visszajelzések és jelentkezések kérdése. A kérdés csak az, hogy miként illeszthetünk be egy fórumot a weblapunkra? A válasz legtöbb esetben attól függ, hogy milyen oldalt használunk.
Amennyiben nagyobb eszközkészletet alkalmazunk, például OpenACS-t, Zope-ot vagy PHPNuke-ot, legalább egy könnyen telepíthető csomagot fogunk találni a témában. Ilyenkor a felület egyrészt összhangban lesz a statikus oldalakkal és az egyéb alkalmazásokkal, valamint ezek az eszközkészletek oldalunk más részeivel azonos felhasználói és engedélyrendszert használnak a fórumok elérésére. Más szavakkal, nem szükséges külön oldalgazdát és fórumgazdát kijelölnünk; bárki, aki rendszerfelügyeleti jogosultságokkal rendelkezik az oldalon, minden további beállítás nélkül képes lesz a fórumot és bármilyen más alkalmazást futtatni.
Másik megoldásként a kiszolgálónkon telepített kiszolgálóoldali technológiákat felhasználva felrakhatunk valamilyen önálló webfórumcsomagot. Ha például PHP-rendszerünk van, tegyük fel a ***Phorum*** csomagot. Ez ugyan nem fog teljes mértékben beépülni oldalunk egyéb részeibe, viszont igen hatékony és megbízható csomag, ami MySQL és PostgreSQL alatt egyaránt jól működik. Az alapul szolgáló technológiák szerint rengeteg lehetőség között választhatunk, a PHP és JSP nyelvektől kezdve egészen jó öreg Perlben írt CGI-programokig. Természetesen kereskedelmi terméket is választhatunk, megvásárolhatunk egy olyan terméket, mint például a ***WebCrossing***. Néhány ügyfélnél használtam már a ***WebCrossing***-ot, és meg kell mondanom, annak ellenére, hogy ez a csomag jóval több lehetőséget ajánl, mint a ***Phorum***-hoz hasonló nyílt forrású csomagok, a különbségek a legtöbb esetben nem olyan nagyok, hogy megérje pénzt költeni rá, nem beszélve arról, hogy ehhez egy olyan csomagot kellene megtanulni, amelyet nem lehet sem módosítani, sem fejleszteni.
Végül létrehozhatjuk a saját fórumcsomagunkat is, ahogy azt néhány évvel ezelőtt magam is megtettem e hasábokon. A webalapú fórumok kialakítására képes programok megírása nem is olyan nagyon bonyolult, de a fejlesztésre és hibakeresésre szánt időt és erőfeszítéseket sokkal érdemesebb egy létező csomag megtanulására fordítani. Akinek azonban van egy kis tapasztalata a web és az adatbázisok témában, az tudja, hogy egy webfórumcsomag létrehozása nemigen áll másból, mint hogy összepakol néhány táblát (felhasználók, üzenetek és a hírcsoportok), majd megadja a jogosultságot az embereknek, hogy bejegyzéseket (küldemények) helyezhessenek el az egyes hírcsoportokban.
Ezek a rendszerek több mint elégségesek egy kis vagy közepes méretű weboldal számára; de ha időközben hatalmasra duzzadna is a rendszerünk, az elküldött üzenetek ezreivel és felhasználók százaival birkózva is valószínűtlen, hogy komoly hatása lenne ezekre a rendszerekre. Ugyanis valamennyi rendszer relációs adatbázisokat használ az adattárolásra, márpedig még a legkisebb és legegyszerűbb korszerű adatbázisrendszer is képes napi néhány ezer tranzakció kezelésére.
Egyre gyakrabban fordul elő azonban, hogy az egyszerű webalapú fórum már nem bizonyul elegendőnek. Bár az emberek általában hajlandók végignézni a webalapú fórumokat, nemigen térnek vissza nap mint nap, hogy kövessék a vita fonalát. Hiszen míg az elektronikus levél aktív (push) médium, amelyben az adatokat elküldik nekünk, a fórumok passzívak (pull), ahol az új üzenetek a lekérésünkre várnak.
Az aktív és passzív felosztás nem új találmány; akik még emlékeznek közülünk a web előtti internet korára, jól tudják, hogy ugyanezen felosztás alapján különböztettük meg a levelezőlistákat és az Usenet hírcsoportokat. A megoldás végül az lett, hogy néhány mail-Usenet átjárót kellett készíteni, amelyek közül sokat még ma is használnak. A GNU Emacs legfrissebb hibáiról egyaránt tájékozódhatunk, ha feliratkozunk a bug-gnu-emacs levelezőlistára, vagy ha elolvassuk a ➲ gnu.emacs.bug hírcsoportot a Useneten. A két dolog egyenértékű, s egy átjáróprogramon alapul, amely felesleges ismétlések nélkül viszi át az üzeneteket egyik rendszerről a másikra.
Létezik ilyen rendszer a webalapú fórumokhoz is? A válasz egyértelmű igen. Végtére is nem túl nehéz elérni egy web-, illetve adatbázisrendszeren, hogy új üzenetek érkezésekor bizonyos híranyagokat egy adott elektronikus levélcímlistára küldjön el. Kifinomult figyelmeztetőrendszer felépítése sem sokkal nehezebb, létezik például egy ***Forums*** OpenACS-csomag, amely lehetővé teszi, hogy a felhasználók az adott fórumokra vagy fórumon belüli hírcsoportokra feliratkozzanak, majd havi, napi, illetve azonnali frissítést kérjenek maguknak. Más szavakkal a Forum program képes elektronikuslevél-tömörítvényeket (digest) készíteni, pontosan úgy, ahogy azt a listaprogramok teszik.

Levél küldése

Ha viszont azt szeretnénk, hogy a jelentkezők saját elektronikus levelezőprogramjukkal is üzeneteket tudjanak küldeni

a listára, a dolgok bonyolultabbá kezdenek válni, ugyanis figyelembe kell vennünk néhány tényezőt:

- ***Biztonság és engedélyek*** – bárki küldhet üzeneteket és válaszolhat a fórumra vagy csak a tagok tehetik meg? Ha a fórum elérése valamilyen szinten kötött, nyilván kell tartanunk az engedélyezett elektronikus levélcímeket. Minthogy közismerten nem túl nagy művészet összekalapálni egy hamis elektronikus levélfejlécet, nem igazán lehet biztonsággal megállapítani, hogy a küldemény valóban a feliratkozótól jött-e, esetleg éppen egy féreg vagy vírus üzenget a feliratkozó nevében.
- ***MIME*** – a legnépszerűbb elektronikus levelezőprogramok (különösen a Microsoft Outlook) alapértelmezés szerint egy vagy több csatolmánnyal küldik a leveleket. A Forum programnak elég intelligensnek kell lennie, hogy kezelni tudja az ilyen típusú leveleket, kiszedve a csatolmányt és a HTML-részeket.
- ***Méret*** – amennyiben a levélfórum-átjáró nem elég okos ahhoz, hogy kigyomlálja a különösen nagy küldeményeket, akkor egy hatalmas küldemény postázásával bárki szolgáltatmegtagadás- (Denial-of-Service) támadást intézhet a weboldalunk ellen. A programnak elég okosnak kell lennie ahhoz is, hogy az ilyen leveleket kiszűrje, és az oldal gazdájának lehetővé tegye a felhasználói levélméretek korlátozását.
- ***Témák*** – a legtöbb fórumprogram az együvé tartozó küldeményeket a küldemény címe alapján együtt kezeli, vagy megjegyzi, hogy melyik levél melyik levélre volt válasz. Igaz, ez nem túl egyszerű, de a nyomon követés a web- és elektronikus levél egyesítése esetében is megoldható, bár mindkét médium más típusú rendszert alkalmaz az ilyesféle dolgok nyilvántartására.

Számos megoldást láttam az ilyen és hasonló nehézségek kezelésre, de mind a mai napig nem találtam olyat, amelyik valamennyi kívánalmamat teljesítette volna.
A ***Phorum*** egy ***phorummail*** nevű fájlt ad nekünk, amit parancssorból kell meghívnunk, feltételezhetően a ***.forward*** vagy ***.qmail*** állományból, vagy a ***mail alias*** meghatározásokat tartalmazó fájlból. A rendszergazdának készítenie kell egy álnevet (például ***apartments-forum***), majd a ***.forward*** fájlt a phorummailre állítva átadja neki a kötelező `FORUM_ID` és az elhagyható `PATH_TO_FORUM` értékeket. Amint ezzel megvagyunk, bárki, aki levelet ír az ***apartments-forum@saját.webhely*** címre, egyben a fórumunkra is írhat. Nyilvánvaló módon a phorummail helyes működéséhez az engedélyeket megfelelően kell beállítani.
A gond csak az, hogy a küldeményekkel kapcsolatban nincs különösebb biztonság; igaz ugyan, hogyha a fórum moderált, a levélben érkezett üzenetek addig elfogadatlannak lesznek bejelölve, amíg a moderátor meg nem nézi őket. A témakezelést többféle ügyes módszerrel is elintézhetjük, úgy tűnik viszont, hogy nemigen megoldott a MIME- vagy a levélbomba-támadások kezelése. Más szóval, a Phorum levélfórum elvárásainknak megfelelően kezeli a leveleket, de ennél többet nemigen nyújt.
Az OpenACS fórumprogramja már kifinomultabb, `qmail` alapú rendszert nyújt, amelyben minden kimenő figyelmeztető üzenet egyedi azonosítóval rendelkezik. Ez azt jelenti, hogy az elektronikus levélüzenetre küldött válasz a kérdéses témához kerül. Igazság szerint az, hogy az OpenACS az elektronikus levélcímeket használja bejelentkező névként, eléggé gyatra biztonságot ad, de továbbra is tény, hogy nagyon nehéz megállítani az üzenetek hamisítását.

Levelezőlisták és adatbázisok

Az általam ajánlható legjobb megoldás, ha mindent fejreállítva egy már létező elektronikus levéllista alapján hozunk létre magunknak webalapú fórumot. A `qmail` szerzője, ***Dan Bernstein*** által készített `ezmlm` például rendelkezik egy `ezmlm-idx` nevű bővítménykészlettel, amely többek közt lehetővé teszi, hogy a feliratkozók listáját MySQL- vagy PostgreSQL-adatbázisban tároljuk. A lista beállításakor a rendszergazda egyúttal néhány adatbázistáblát is létrehoz, majd az `ezmlm`-ben beállítja őket.
Így egy jó webfejlesztő létrehozhat egy elektronikus levéllistát, majd azt tükrözheti a weben. Bármi, ami a webről érkezik, úgy néz majd ki, mintha a pillanatnyilag bejelentkező felhasználótól érkezne, akinek feltételezhetőleg be kell jelentkeznie a webalapú fórumalkalmazásba. Bármilyen adat, amely a pillanatnyi felhasználótól érkezik, ugyanazon a szokásos ellenőrzésen megy keresztül, amit az `ezmlm` egyébként alkalmaz.
Egyelőre úgy tűnik, hogy a `mailman` levelezőlista, amely minden MTA-val képes együttműködni (például `qmail`, Sendmail, Postfix és Exim) és folyamatos, lenyűgöző fejlesztés alatt áll, semmiféle olyan megoldással nem rendelkezik, amellyel a felhasználóit relációs adatbázisba menthetné. Ugyanakkor viszonylag biztonságos, könnyen használható Berkeley DB fájlokban tárolja őket, így webalapú fórumcsomagunk innen kiolvashatja őket.
Ha a fórumokat levelezőlistaként használjuk, azért felmerülhet néhány gond: többek között a témacsoportok kérdése, amely – mint fentebb említettük – egészen másképp kezelhető az elektronikus levél és a web esetében. Vegyük ehhez hozzá azt a tényt, hogy a legtöbb fórum a felhasználóinak lehetővé teszi, hogy kiemeléseket és csatolmányokat használjanak, sőt akár szerkeszthessék is a saját küldeményeiket – és máris nyilvánvalóvá válik, hogy ez a házasság a két médium között nem lesz éppen súrlódásmentes, mégha létre is tudjuk hozni.
Mindazonáltal a szolgáltatások sok olyan ember számára hasznosnak bizonyulhatnak, akik hajlandóak lemondani a cifra dolgokról. Ehelyett ugyanis a fórumuk egyre növekvő számú jelentkezőivel vannak elfoglalva, és a felhasználóiknak megadják a lehetőséget, hogy eldöntsék, milyen formában szeretnének jelentkezni ezekre a fórumokra.

qpsmtpd

Ahogy nyilván önök közül is sokan, jómagam is nap mint nap szenvedek a kéretlen levelek áradatától. A nyílt forráskódú ***SpamAssassin*** képes kategorizálni a bejövő leveleket, így a levélszemét (spam) ellen folytatott küzdelmeimben kiváló szövetségesnek bizonyult. Amennyiben futtatunk már valaha levelezőlistát, kétségtelenül felfedeztük, hogy a levélférgek nemigen válogatnak: éppoly előszeretettel küldik magukat a listákra, mint az egyéneknek.
Bár a rendszerfelügyelet és az SMTP-kiszolgálók nem igazán illeszkednek szorosan cikkünk témájához, úgy éreztem, muszáj néhány dicsérő sort szentelnem ***Ask Bjoern Hansen*** nyílt forrású SMTP-kiszolgálójának, a `qpsmtpd`-nek.
A `qpsmtpd` eredetileg `qmail` mellett működött, jelenleg már számos MTA-val képes együttműködni, ideértve a Sendmailt és a Postfixet.
De miért használnánk `qpsmtpd`-t az alapértelmezett `qmail-smtpd` helyett? A Perl-guruk számára egyszerű az indok a váltásra, hiszen `qpsmtpd` Perlben készült. De ha ennél egy kicsit kevésbé vagyunk nyelvőrültek, akkor is számos jó dolgot találhatunk benne. Ugyanis a `qpsmtpd` az SMTP-k levél-

küldő eljárásait több állomásra osztja fel, mi pedig valamennyi ilyen állomáshoz saját szolgáltatásokat illeszthetünk be.
A qpsmtpd saját lapjáról, a ➲ http://www.develooper.com-ról (igen, két o betű áll egymás után) töltöttem le, majd a telepítési útmutatót követve húsz perc múlva már futott is. Ne feledjük, hogy a qpsmtpd teljes értékű SMTP-kiszolgáló, azaz nem fog futni, ha más SMTP-kiszolgáló ellenőrzi a 25-ös kaput. Amennyiben az SMTP-kiszolgáló rendszerindításkor való indulását és későbbi futását a daemontools program segítségével végezzük, érdemes ellenőrizni, nem maradt-e véletlenül egy régi SMTP-kiszolgálóra mutató hivatkozás a */**service*** könyvtárban. Ilyen esetben ugyanis a legközelebbi újraindítás után két versengő SMTP-démonunk lesz.
A qpsmtpd kulcselemei a ***plugins*** alkönyvtárban található bővítmények. A soronként egy bővítményt tartalmazó ***config/plugins*** állomány segítségével magunk is beilleszthetünk saját bővítményeket. Például a saját ***config/plugins*** állományom egyik részlete így néz ki:

```
# quit_fortune

check_earlytalker
count_unrecognized_commands 4

require_resolvable_fromhost
```

Más szóval, megjegyzésbe tettem a quit_fortune bővítményt, viszont aktiváltam a check_earlytalker, count_unrecognized_commands és require_resolvable_fromhost bővítményeket.
A count_unrecognized_commands értékként egyetlen számot vár, amit itt meg is adtunk.
Ha meg szeretnénk nézni ezeket a bővítményeket, esetleg sajátot szeretnénk elhelyezni, lépjünk be ***plugins*** könyvtárba. Minden bővítmény tartalmaz egy regisztráló alprogramot, amellyel a qpsmtpd valamelyik állomásához csatlakozik, valamint egy másik alprogramot, amelyet azután az állomás meghívhat. Például a require_resolvable_fromhost bővítmény a következőképpen kezdődik:

```
use Net::DNS qw(mx);

sub register {
  my ($self, $qp) = @_;
  $self->register_hook("mail",
        ➥"mail_handler");
}
```

Azaz a regisztráló alprogram értesíti a qpsmtpd-t, hogy amint az SMTP-ügyfél meghívja a mail parancsot, a mail_handler alprogramot is meg kell hívni. Ez az alprogram pedig a következőképpen működik:

```
sub mail_handler {
  my ($self, $transaction, $sender) = @_;

  $sender->format ne "<>"
    and $self->qp->config
        ("require_resolvable_fromhost")
    and !check_dns($sender->host)
    and return (DENYSOFT,
        ($sender->host
         ? "Could not resolve ". $sender->host
         : "FQDN required in the envelope
           ➥sender"));
  return DECLINED;
}
```

Ha már volt dolgunk mod_perl alatti webfejlesztéssel, akár ismerősnek is tűnhet. A mail_handler egyrészt visszaadhat DECLINED értéket, ami azt jelenti, hogy minden simán ment, és a levél keresztülmehet. Küldhet DENYSOFT értéket is, ami a felhasználónak azt üzeni, hogy inkább később próbálkozzon. Ezt azért tesszük, mert nem szeretnénk azonnal elutasítani a leveleket csak azért, mert a DNS-kiszolgáló leállt – kizárólag a kéretlen levélküldőket szeretnénk megbüntetni, illetve azokat, akiknek nem kellene közvetlenül elküldeniük a levelet. Küldhetünk továbbá DENY üzenetet, amely véglegesen elutasítja az üzenetet.
Annak ellenére, hogy nemigen kapunk hozzá túlságosan jó leírást, a qpsmtpd letöltését követően néhány órával már meg tudtam írni az első működőképes bővítményemet, és biztos vagyok benne, hogy a többi felhasználó is hasonló tapasztalatokat fog szerezni. Minthogy a qpsmtpd Perlben íródott, természetesen gyorsan és könnyen elérhetjük a Perl-programok által hozzáférhető összes lehetőséget, valamint a fejlesztést leegyszerűsítő a CPAN-modulokat.
Több szinthez is csatolhatunk bővítményeket, többek közt a helo, ehlo, connect, sőt akár a rcpt szinthez is, amelyek mindegyikén különféle teszteket végezhetünk el. Még ***SpamAssassin***-bővítmény is létezik qpsmtpd-hez, amely a híres levélszemétszűrő programot hívja meg, mielőtt a levél megérkezne a levelesládánkba.
Körülbelül egy hónapja használok qpsmtpd-t, és a levelesládámba vándorló kéretlen levelek száma örvendetesen megcsappant, mégha néhányat át is enged a ***SpamAssassin***. Ha a saját gépünkön dolgozunk, jó szívvel ajánlom a qpsmtpd-t. Kiváló példa arra nézve, hogyan írjunk tetszőleges bővítményeket kezelő programot, ráadásul csak azt a levelet kapjuk meg, amelyet tényleg meg kell kapnunk.

Összegzés

A levelezés létfontosságú része az internetnek, mint azt kétségtelenül mindenki tudja, aki ezeket a sorokat olvassa. Az internet azonban folyamatosan terjeszkedik, így a levelezésre több irányból is nyomás nehezedik. Ebben a hónapban a webalapú fórumokkal ismerkedhettünk meg, illetve megnéztük, miképpen lehet csatlakoztatni őket a levelezőlistákhoz. Röviden bepillantottunk a qpsmtpd, egy választható, különlegesen bővíthetőre és átállíthatóra tervezett SMTP-kiszolgáló világába. Remélem, a következő hónapban visszatérhetünk a Bricolage és a többi nyílt forrású tartalomkezelő rendszer birodalmába, nagy hangsúlyt fektetve arra, hogyan illeszthetünk be már létező weboldalainkba valamilyen CMS-t.

A Kapcsolódó címek megtalálhatóak az 55. CD Magazin/Elektronikuslevel könyvtárában.

Linux Journal 2003. december, 116. 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 feleségével és lányaival nemrég költözött Chicagóba.

Dobj be még egy ötcentest!

Konyhafőnökünk ízletes válogatást készít sokoldalú zenelejátszó programokból. Próbáljuk ki mindet a „My Heart Will Go On”-nal!

Hogy érted azt, hogy még soha nem hallottad ezt a dalt, François? Ez ***Céline Dion*** egyik klasszikussá vált száma, mon ami. Mit mondasz? Úgy gondolod, hogy az éneklésemet a zuhanyozás idejére kellene korlátoznom? Azt hiszem, François, megfeledkezel arról, hogy ki ennek az étteremnek a konyhafőnöke és a tulajdonosa… Non, mon ami, csupán csak a tájékoztatás végett említettem, és inkább szemet hunyok epés megjegyzésed fölött. Nézz csak ide! Mindegyik programunk képes rá, hogy a segítségével lejátszási listát állítsunk össze kedvenc dalainkból a vendégek számára.

Ami a vendégeinket illeti, már itt is vannak. Isten hozott titeket, mes amis, Chez Marcelnél, a kitűnő Linux-konyha, a különleges borok és a kiváló könnyűzene otthonában – a mostani témám ugyanis a szórakoztatás. Azt kérdezitek, hogy miért a zenei szórakoztatás? Életem egy régebbi időszakában azzal kerestem egy kis zsebpénzt magamnak, hogy a szüleim barátainak a gyerekeire felügyeltem. A barátok egyike zenegépek javításával foglalkozott, így a házban mindig ott volt egy igazi zenegép, tele 45-ös fordulatú slágerekkel. A gép időről időre cserélődött, de a pénzadagoló mechanika kiiktatásával mindig hozzájuthattam az engem érdeklő zenékhez. Ez volt ám a remek szórakozás!

Mint olvasóink közül is bizonyára sokak, én is számos dalt tartok a merevlemezemen Ogg Vorbis és MP3 formátumban. Ennek jó oka van, mert ugyan elég tekintélyes CD-gyűjteménnyel rendelkezem, de ha a noteszgépemmel szeretném lejátszani őket, akkor a lemezeket is magammal kellene hordoznom, ami nem túl kényelmes megoldás. Ezért ezekről a CD-kről beolvasom a dalokat és a lehető legkisebb helyen, nevezetesen a noteszgépemben tárolom őket. A dobozokban tárolt CD-k eltűnnek a lemezem virtuális terében, de sajnos a rendszer még messze van a tökéletestől. Most ezeket a dalokat számos nagyméretű könyvtárban tárolom, amik kevéssé vagy egyáltalán nem szervezettek. Ha valamit le szeretnék játszani, előtte keresgélnem kell. De most végre búcsút vehetünk a keresgéléstől, mes amis! Mai menünkön néhány nagyszerű nyílt forráskódú zenelejátszó program szerepel, amelyek közül bármelyiket a Linux-rendszerünkre feltelepítve egy jól elrendezett, kereshető és élvezettel használható dalgyűjteményhez juthatunk – éppen olyanhoz, mint fiatalságom gépei voltak, csak éppen sokkal kisebb méretben.

François! Te meg mit lábatlankodsz még mindig itt? A pincébe, immédiatement! Úgy gondolom, ma valami könnyűre van szükségünk, ami friss és izgalmas. A 2001-es Riesling Auslese Gold Cap Mosel-Saar-Ruwer Wehlener Németországból bizonyára megfelel az érzékszervek legfontosabbika, az ízlelés elsőrangú szórakoztatásához. Vite, François!

Amíg François visszaér a borral, kezdjük is el a linuxos zenelejátszó programok terén tett körsétánkat. Ha KDE munkaasztal felhasználói vagyunk (sőt akkor is, ha esetleg nem), érdemes egy pillantást vetnünk ***Scott Wheeler*** JuK programjára (*1. kép*). A kiejtésnél a jukebox (zenegép) szóra gondoljunk,

1. kép A JuK felkerült a KDE 3.2 tervezett részeit tartalmazó jelöltlistára

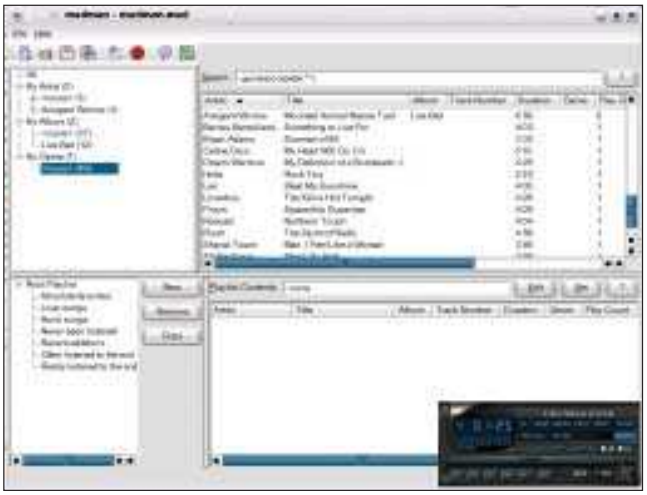
és máris tudni fogjuk, miről van szó. Lévén KDE-alkalmazás, a program egy tálcaikonnal szépen beleillik a KDE-környezetbe, amivel könnyen eltüntethetjük a programot a látóterünkből. Támogatást kínál az Ogg Vorbis- és MP3-fájlokhoz, a gyűjtemény- és lejátszási listák létrehozásához, a címkeszerkesztéshez és még sok egyébhez.

A JuK a KDE 3.2 részeként kerül majd kiadásra és érdemes arra, hogy bármelyik jövőbeli KDE-telepítés része legyen. Addig is szerezhetünk a programból egy példányt a ➲ http://www.slackorama.net/cgi-bin/content.pl?juk címről. A JuK fordítása a már megszokott öt lépésből áll:

```
tar -xzvf juk-1.95.tar.gz
cd juk-1.95
./configure
make
su -c "make install"
```

Amikor a `juk` paranccsal elindítjuk a programot, egy egyszerű kéttáblás felülettel találjuk szemben magunkat egy oldalsávval a bal oldalon és egy nagy munka-, illetve megjelenítő ablakkal a jobb oldalon, amelyek mindegyike átméretezhető. A szokásos menü és a gyors elérést nyújtó ikonok felül helyezkednek el. Az oldalsáv egyetlen, ***Collection List*** (gyűjteménylista) feliratú ikonnal rendelkezik. Ha a JuK-hoz zeneszámokat kívánunk hozzáadni, az nem áll többől, mint a könyvtármegnyitó ikonra való kattintásból, egy könyvtár kiválasztásából és a kívánt dalok kijelöléséből. Kattintsunk az ***OK*** gombra, és a dalok megjelennek az alapértelmezett gyűjteménylistában. Mivel a rendezettség a cél, újabb lejátszási listákat hozhatunk létre és ezekre ráhúzhatjuk a kiválasztott zeneszámokat. A címek megjelennek az új listában, de benne maradnak a főgyűjteményben is.

2. kép Az a drága kis Rhythmbox



3. kép Nem kell bolondnak lennünk ahhoz, hogy a Madmant használjuk

A fő zeneszámablakban valamelyik oszlopra rákattintva az album címe, az előadó, a dátum vagy egyéb azonosító szerint rendezhetjük a tételeket. Bárki, aki rendelkezik már a számítógépén valamilyen zenegyűjteménnyel, tudja, hogy az adatcímkékben (tagokban) tárolt adatok nem minden esetben helyesek. Ennek a gondnak a kezelésére a JuK hatékony címkeszerkesztőt is biztosít. Azt szeretem a leginkább ebben a szolgáltatásban, hogy a dalcímen jobb egérgombbal kattintva a sorban is szerkeszthetjük a tartalmat, de az alaposabb címkeszerkesztőt is előhívhatjuk. Tegyük fel, hogy 12 zeneszámunk van ugyanarról az albumról, és az album dátumát is rögzíteni szeretnénk. Ekkor a JuK szerkesztője lehetővé teszi, hogy több címet is kijelöljünk és ezek kiválasztott mezőit egyszerre megváltoztassuk – ez a módszer nagyon célszerű és sok időt takaríthatunk meg a révén.

A Gnome oldaláról engedtessék meg bemutatnom ***Colin Walters*** Rhythmbox nevű programját (*2. kép*). A Rhythmbox egy újabb takaros zenelejátszó program. Kereshetünk vele zeneszámokat, különböző tulajdonság alapján rendezhetünk, lejátszási listákat hozhatunk létre és a többi. A zenekönyvtár

4. kép A Madman akkor is előkeresi nekünk a dalt, ha nem emlékszünk már a pontos írásmódjára

tallózónézete két vagy háromoszlopos nézetre állítható be, megjelenítve az előadót, a műfajt és az album címét. Kattintsunk bármelyikre, és a Rhythmbox leszűkíti a kiválasztást. Egy címkeszerkesztő is része a programnak, de – jelenlegi állapotában legalábbis – nincs mód az adat sorban szerkesztésére. Beépített internetrádió-támogatással rendelkezik, egy vevőkártyával ugyanarról a felületről élvezhetjük a kedvenc adóinkat. A Rhythmbox letöltésére a ➲ http://www.rhythmbox.org honlap nyújt lehetőséget.

A program a `gstreamer` csomag segítségével játssza le a zeneszámokat, ezért ennek megléte fontos előfeltétel. A Rhythmbox forrásból való fordítása nem túl bonyolult dolog, de én egy kicsit felhúztam magam, mire az összes szükséges Gnome fejlesztői csomagot megkerestem, amelyek között megtalálható a `bonobo`, a GTK+, a `pango`, a `gnomeui`, az `atk`, a `gconf` stb. Mindezt azért említem meg, mert jó párszor vissza kellett térnem olyan csomagokért, amelyek a fordításkor hiányoztak. Akik a gyors megoldásokat kedvelik, azok számára bináris csomagok is elérhetők. Ha jó RPM-lelőhelyet keresünk, próbáljuk meg az ➲ http://rpm.pbone.net címet. Visszatérve a fordításra: ismét egy ötlépéses kicsomagoló–fordító folyamatról van szó, tehát nagy meglepetések nem érhetnek bennünket:

```
tar -xzvf rhythmbox-0.5.3.tar.gz
cd rhythmbox-0.5.3
./configure
make
su -c "make install"
```

A Rhythmbox első indításakor egy kis kérdés–válasz részen kell túljutnunk, amelyben meg kell adnunk, hogy a rendszerünkön hol tároljuk a zenéket. Megadhatjuk az elérési útvonalat, de teljesen ki is hagyhatjuk ezt a lépést. Amennyiben kihagyjuk, kézzel később is hozzáadhatjuk a zeneszámokat. A Rhythmbox minden tétel címkéjéből beolvassa a dal címét, az előadót és a többi adatot, majd a rendezést önműködően végrehajtja. A címkeszerkesztő segítségével nulla és öt csillag között rangsorolhatjuk a zeneszámokat, és ezt az osztályozást később a kedvenc dalaink kiválasztására használhatjuk fel. Mai menünk utolsó fogása „a zenekezelő, amely mindent elintéz” szlogennel értékeli önmagát, s ez az önreklám felkelti az ember figyelmét. De ne feledkezzünk meg az italunkról sem, kortyintsunk egyet ebből a finom borból! A program, amely mindent elintéz, ***Andreas Klöckner*** Madman nevű alkotása (*3. kép*), ami a ➲ http://madman.sourceforge.net/index.php oldalon érhető el.

Ha a program forrásának lefordítása mellett döntünk, akkor régi barátunkhoz, az ötlépéses kicsomagoló–fordító eljáráshoz kell visszatérnünk. Az írás idején kiadott csomag a ***madman-0.91.1.tar.gz***.

Természetesen a Madman alapvető szolgáltatásai nagyjából megegyeznek a JuK és Rhythmbox által nyújtott zenegyűjte-

mény-elrendező és a kiválasztott dalokat lejátszó felület képességeivel. Pedig a Madman valóban különbözik – még az érzékelése is más. Keresőképességei például elképesztően rugalmasak. Tegyük fel, hogy egy ***Céline Dion*** által előadott zeneszámot szeretnénk meghallgatni, de nem tudjuk, hogyan írják a Celine szót. Egyszerűen írjuk be, hogy selin, vagy valamit, amiről úgy gondoljuk, hogy jó lehet, és a Madman felkutatja számunkra a dalt (*4. kép*). Mais oui, természetesen tudom, hogyan kell írni.

Várjunk csak, van itt más is. A Madman főablakának az alsó részén egy újabb kitűnő Madman-szolgáltatást találhatunk. A program önműködően nyomon követi, hogy mely dalokat hallgattuk végig és hányszor. Így egyszerűen a zenehallgatás révén is létrejön kedvenceink listája. Arra is lehetőségünk nyílik, hogy egy dal címén a jobb egérgombbal kattintva, azt az ötcsillagos rendszert használva besoroljuk valamelyik minőségi kategóriába.

A Madman indításakor rákérdez, hogy mely könyvtárakban tároljuk a zeneszámainkat. Egy egyszerű hozzáadó párbeszédablakról van szó, amellyel annyi helyet adunk meg, amennyit akarunk. Gyűjteményünk összes zenéje elérhető előadó, műfaj vagy az album címe szerint. Lehetőség nyílik egy dal felkutatására a Madman korábban említett szuperrugalmas keresőmotorjával is. Amikor találtunk egy kedvünkre való dalt, jelöljük ki és nyomjuk meg az F9 billentyűt; az F10 billentyűvel pedig a következőt tudjuk kiválasztani. Dönthetünk a teljesen véletlenszerű kiválasztás mellett is.

Míg a most vizsgált programok mindegyike közvetlenül a felületről játszotta le a zenét – a lejátszószerkezetet elrejtve a háttérben –, a Madman az XMMS-t indítja el a lejátszáshoz.

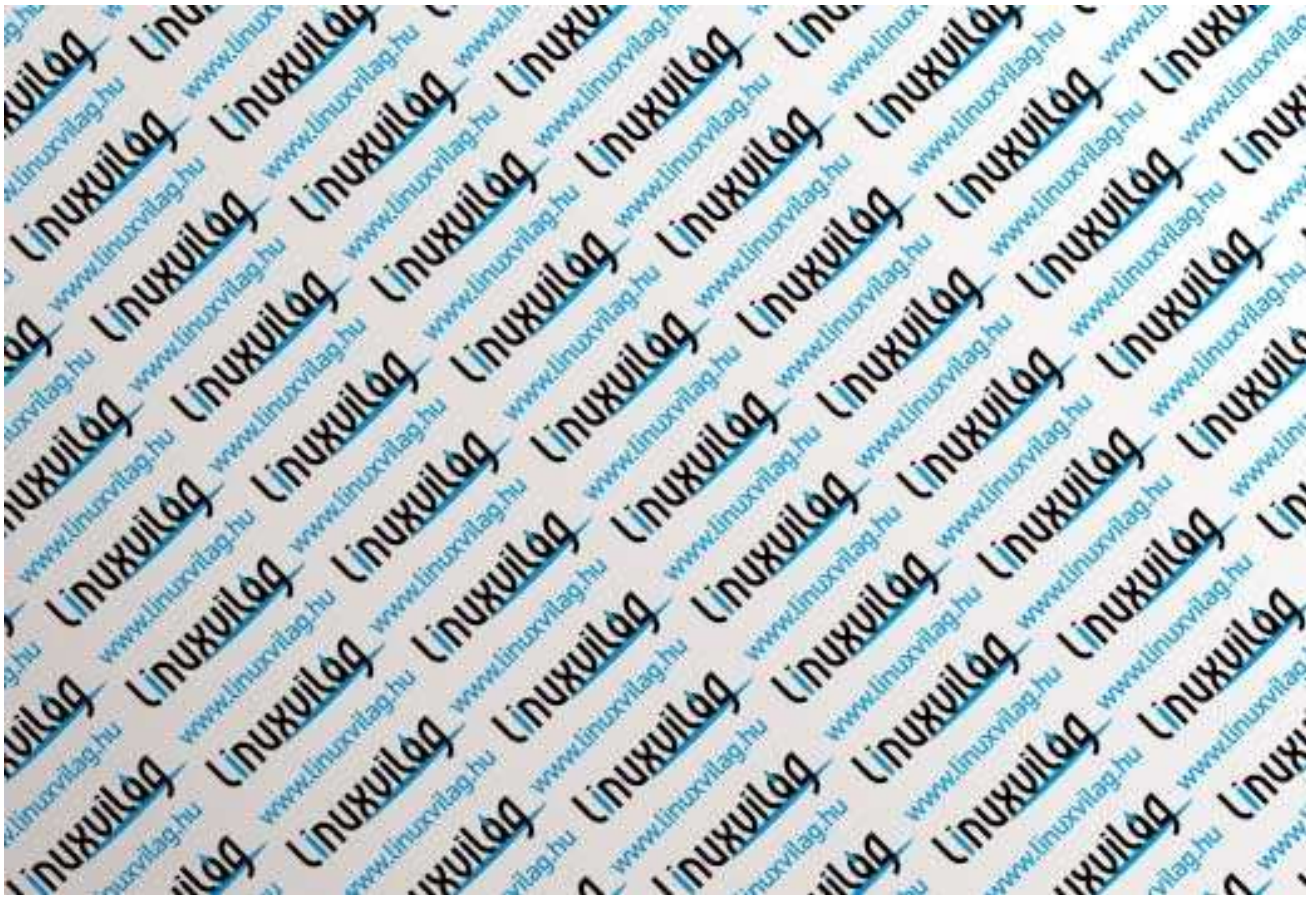
A beállítások párbeszédablakán van egy lehetőség, amely más lejátszóprogram kiválasztását teszi lehetővé, de pillanatnyilag az XMMS tűnik az egyetlen alkalmas lejátszónak.

Egy csomó szeretnivaló akad a Madmanben. Amikor CD-lemezeket olvasok be és alakítok át, néha azt tapasztalom, hogy az adat nem került át megfelelően a címkékbe. Eredményül egy üres felsorolást kapok. Semmi gond: nyomjuk le az F9-et a lejátszásához, ellenőrizzük az XMMS kijelzőjén a címet, és egy lassú kétszeres kattintással nyissuk meg javításra a Madman kijelzőjén.

Mit mondtál, François? Á, köszönöm! Sajnálom, mes amis, de úgy tűnik, sikerült egy újabb estét eltöltenünk. Hogy elszállt az idő! Mindazonáltal nem akarjuk, hogy innen, Chez Marcel-től a vendégek anélkül menjenek el, hogy még egy pohárka bort kényelmesen elfogyasztanak, ezért megkérem François-t, hogy töltsön még egyszer utoljára. Mivel mindenkinek van az asztalán egy linuxos gép, nyugodtan feltehetitek a fejhallgatót és mialatt a bort kortyoljátok, meghallgathattok egy búcsúdalt. Viszontlátásra a következő alkalmig! Egészségetekre, mes amis! A vôtre santé! Bon appétit!

Linux Journal 2003. december, 116. 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 (jelenleg is egy könyvön dolgozik).



Bevezetés a vi használatába

Még akkor is érdemes megismerkedni a mindenütt jelenlévő vi használatának alapjaival, ha munkád zömében másfajta szövegszerkesztőt használsz.

A Linuxszal vagy Unixszal ma ismerkedők többsége már tudja, hogyan kell használni egy grafikus WYSIWYG (amit látsz, azt kapod) szövegszerkesztőt, amely nagymértékben támaszkodik az egér, az ikonok és a legördülő menük használatára. A vi-jal, a Unix/Linux szerkesztőjével való első találkozás alkalmával az nehézkesnek tűnhet, és nem látszik valami hatékonynak sem, de ennek éppen az ellenkezője az igaz.
Ahhoz, hogy a parancsok beviteléhez a teljes billentyűzet rendelkezésre álljon, a vi-nak egy parancs- és egy beszúró üzemmódja van.
Az üzemmódok először zavaróak lehetnek. A legtöbb szerkesztőnél bármi, amit begépelsz, bekerül a dokumentumba, a szerkesztés az egér, esetleg gyorsbillentyűk segítségével történik. A vi-nál parancsmódban a teljes billentyűzet a szerkesztőparancsok bevitelére szolgál. Csak akkor kerül bele szöveg a dokumentumba, ha kiadod a beszúrás parancsot.
A legtöbb új vi-felhasználó csak néhány alapvető parancsot tanul meg: kurzorbillentyűk, `i` a beszúrás, `x` egy karakter törlése, `dd` egy sor törlése, `:wq` mentés és bezárás, esetleg még egy-két más parancsot is megismernek. Hamar megunják, hogy minduntalan ugyanazokra a billentyűkre kell támaszkodni, ugyanakkor félnek attól, hogy hatalmas parancslistákat tanuljanak meg. Ahelyett, hogy egyszerűen egy bemagolandó felsorolást készítenénk, most megpróbáljuk felfedezni a mögöttes szerkezetet, könnyen megjegyezhetővé téve ezáltal a parancsokat, és megpróbáljuk elérni, hogy gyorsan kiemelt vi-felhasználóvá lépjünk elő. A parancsok néhány alapvető kategóriája: szövegbeszúró, kurzormozgató, blokkszerkesztő, kettőspont (ex) parancsok, beállítások és egyéb parancsok.

Szöveg beszúrása

Az első parancs, amit az ember megtanul a vi-ban, az az `i`. Az `i` a szövegszerkesztőt beszúró módba teszi, ahol is minden, amit begépelsz, bekerül a dokumentumba, és egészen addig ott marad, amíg meg nem nyomod az Esc billentyűt. Az `i` azonban nem az egyetlen módja szöveg beszúrásának.
A kurzor mögé vagy a sor végére úgy adhatunk szöveget, hogy az `a` parancsot használjuk, vagy a sor végére ugrunk, és az `A` paranccsal fűzzük hozzá a szöveget. Hasonlóképpen az `I` a sor elejére szúr be szöveget. Ezeknél, valamint a többi beszúróparancsnál a parancs üzemmódba való visszatéréshez mindig nyomd le az Esc-t.
Ha a pillanatnyi sor mögé újabb sort szeretnél beszúrni, és oda szeretnéd tenni a szöveget, használd az `o`-t, vagy a pillanatnyi sor elé való beszúráshoz az `O`-t.
A pillanatnyi karakter cseréjéhez nyomd meg az `s`-t, vagy ha sok karaktert szeretnél cserélni, az `R`-t, a grafikus szövegszerkesztők felülíró üzemmódjához hasonlóan.
Csaknem az összes parancsnál használhatunk számelőtagot, amely általában az adott mennyiségben ismétli meg a parancsot. Például írd be, hogy ***3iHurrá***, majd nyomd le az Enter és Esc billentyűket. Ez 3 sor „Hurrá”-t tesz be az átmeneti tárba. Az `s` parancs különbözik ettől; itt a szám azt mutatja meg, hogy hány karakter kerül lecserélésre: az `5s` öt karaktert cserél le azzal, amit beírsz.
Beszúró üzemmódban néhány vezérlőbillentyűnek különleges jelentése van. A kurzorbillentyűket a mozgáshoz lehet használni, de ne feledjük, hogy még mindig beszúrómódban vagyunk. Ugyanígy a Ctrl-T-t és a Ctrl-D-t a sor elején a szöveg behúzására lehet használni. A Ctrl-T a behúzás következő szintjére mozgatja a szöveget, a Ctrl-D pedig vissza. A legjobb eredmény eléréséhez ezeket az ***autoindent*** és a ***shiftwidth*** beállításokkal együtt kell alkalmazni (lásd lejjebb, a beállítások résznél).

A kurzor mozgatása

A grafikus szövegszerkesztőknél elsősorban egérkattintással mozgunk. Néhány vi-klón támogatja az egér használatát, de a vi és klónjai mind olyan kurzormozgató parancsokon osztoznak egymással, amelyekkel nagyon gyorsan lehet mozogni. A legegyszerűbbek a kurzorbillentyűk vagy az azoknak megfelelő betűk: `h` (balra), j (le), `k` (fel) és `l` (jobbra). Ha azonban csak ezeket alkalmazod, elszalasztod a vi-ban rejlő erő teljes kiaknázását.
A sorban úgy is mozoghatsz, hogy a jobb vagy bal nyilakat (vagy az l-t és h-t) folyamatosan nyomva tartod, de létezik ennél egyszerűbb mód is. A sor elejére való ugráshoz a `^`, a sor végére való ugráshoz pedig a `$` parancs használatos. Ha egyszerre egy szót szeretnél ugrani, használd `w`-t, a `W`-t pedig, ha még tovább akarsz menni. Ha vissza szeretnél ugrani, akkor a `b` vagy a `B` használatos. Az `e`, vagy `E` betűket is lehet a pillanatnyi szó végére való ugrásnál alkalmazni.
A kisbetűs `w`, `b` és `e` a szó alfanumerikus karaktereinek sorozatát jelenti; a nagybetűsek szóközt használnak a szavak elválasztásához.
A pillanatnyi sorban való mozgásnak egy másik módja az, ha egy adott karakterre ugrunk. Például, ha a kurzort a következő `s`-hez szeretnénk mozgatni a sorban, gépeljük be az `fs`-t. Használjuk az `F`-et az előző betű megtalálásához: `Fs`. Vagy használjuk a `ts`-t, hogy pontosan az `s` elé kerüljünk vagy a `Ts`-t, ha a másik irányba szeretnénk menni. Az `s` helyére bármilyen más karaktert is beírhatunk, amit meg szeretnénk keresni. Ha meg szeretnénk ismételni a keresést, a , (vessző) karaktert írjuk be.
Ha a kurzort a képernyő tetejére, közepére, vagy aljára szeretnénk mozgatni, használjuk a `H`, `M` vagy `L` betűket (nagybetű). A `H`-nál vagy `L`-nél a számelőtag azt mondja meg, hogy hány sort szeretnénk lépni az oldal aljától vagy tetejétől nézve; az `M`-nél a számnak nincs hatása.
Mondatonként való előre- és hátraugráshoz használjuk a `(` vagy `)` zárójeleket; a bekezdésenkéntihez pedig a `{` és `}` kapcsos zárójeleket. Ha sok zárójellel rendelkező forráskódot szerkesztesz, a `%` paranccsal próbálkozz: helyezd a kurzort a következő karakterek valamelyikére: `(`, `)`, `[`, `]`, `{` vagy `}`, majd üsd le a `%` jelet; ekkor

a megfelelő zárójelhez fogsz ugrani.
A mozgáshoz gyakran a / paranccsal való keresés a legjobb módszer. Például gépeljük be a /hello parancsot, majd nyomjuk le az ENTER-t, ha a következő „hello”-ra szeretnél ugrani a dokumentumban. A „hello” helyett bármilyen szabályos kifejezést használhatsz. Az n parancs megismétli a keresést; az N az ellentétes irányba teszi ugyanezt.
A ? parancsot a fájlban való visszafelé kereséshez használhatod. A keresés kis- és nagybetűérzékeny; az ignorecase beállítást a cikk egy későbbi részében fogom elmagyarázni.
A legtöbb kurzormozgató parancsnál számelőtagot lehet használni a parancsok ismétléséhez. Például az 5w öt szóval fog előre mozgatni, a 2n nem a következő találathoz visz, hanem az azt követőre, az 5fs az ötödik s-hez fog mozgatni a kurzor jobb oldalán.

Blokkszerkesztő parancsok

A grafikus szerkesztőkben egy szövegblokk módosításához először jelöld ki a szöveget, kattints rá és húzd át az egeret a szöveg felett, majd kattints egy ikonra vagy menüelemre, vagy üss le egy gyorsbillentyűt. Bár a vi nem alkalmazza az egeret, az eljárás meglepően hasonló. Mint az előbb láttuk, a vi-nak széles kurzormozgató parancskészlete van. Ezek közül bármelyik tulajdonságként használható a blokkszerkesztő parancsokhoz (például a dw töröl egy szót). Ahelyett, hogy a vi-ban a szövegblokk meghatározásához először egérrel kijelölnénk a szöveget, majd egy műveletet választanánk, először írjuk be a blokkszerkesztő parancsot (d), majd a kurzormozgató parancsot (w). Gyorsbillentyűként a blokkszerkesztő parancsot kétszer is beírhatjuk, hogy az egész jelenlegi sorra hatással legyen, például a dd kitörli a pillanatnyi sort. Mint láthattuk, a d parancs a törlés. Amikor szöveget törlünk, a szöveg megjegyzésre kerül, a grafikus szerkesztők kivág parancsához hasonlóan. A szöveg beillesztéséhez (a vi szóhasználatában: tenni, „put”) a p-t írjuk be a kurzor mögé való beszúráshoz (vagy nagy P-t, ha a kurzor elé szeretnénk beszúrni). Ez megoldja a kivág–beilleszt parancsokat, de mi a helyzet a másolással? A vi-ban ezt kirántásnak („yank”) hívják, és a y parancsot használják. Ez ugyanolyan, mint a d parancs, azzal a különbséggel, hogy nem töröl semmit: az yy megismétli a pillanatnyi sort, a yw a pillanatnyi szót és így tovább.
Nagy mennyiségű szöveg cseréjénél először kitörölhetjük a szöveget, majd az i-vel beszúrhatjuk az újat, de helyette használhatjuk a c (módosít, change) parancsot, és máris beszúró módban vagyunk és beírhatjuk a módosító szöveget; ha befejezted, üsd le az ESC billentyűt. Használhatod kurzormozgató paranccsal együtt (mint például: cw), vagy az egész sort módosíthatod a cc paranccsal.
A programozók imádják a > és < parancsokat, amelyek egy szöveget tolnak el jobbra vagy balra. Egy blokknyi kód eltolásához helyezd a kurzort a { , vagy } karakterekre a blokk elején vagy a végén, és ha jobbra szeretnéd eltolni a szöveget, írd be a >% parancsot. Bármilyen mozgatóparancsot használhatsz a % helyett, az egész sor eltolásához pedig használd a >> parancsot. A < jel segít, ha balra szeretnéd tolni a szöveget.
Lásd a CTRL-T és a CTRL-D billentyűket (miközben beszúró üzemmódban vagy) és a shiftwidth beállítást lejjebb.
Végül a ! parancsot kell alkalmazni a szöveg külső program általi szűréséhez. Írd be a futtatandó program nevét, és a szerkesztő az állomány kijelölt részét szabványos bemenetként adja át; a program kimenetét pedig beilleszti a dokumentum megfelelő helyére. Például a !! a pillanatnyi sort szűri; a !} egy bekezdést szűr. Különösen hasznos parancs a Unix/Linux-szűrő fmt, amely újraformálja a bekezdéseket. Az állomány módosítása nélküli külső program futtatásához nézd meg a :! parancsot a „kettőspont (ex) parancsok” részben. Bármely számelőtag a parancs kurzormozgató részének kerül átadásra, akár elsőnek adjuk meg, akár a szerkesztő- és kurzormozgató parancs belsejében. Például mind a 3dw, mind pedig a d3w három szót töröl; a 3yy és a y3y parancs három sort másol.

A kettőspont (ex) parancsok

A vi egy parancscsoportot örökölt elődjétől, az ex-től, amit az úgynevezett kettőspont üzemmódon keresztül lehet elérni. Gyakori példa a :wq (mentés és kilépés) és a :s/X/Y/ (helyettesítsd X-et Y-nal). Írd be a : (kettőspontot), amelyet egy ex parancs követ, majd nyomd le az ENTER-t. Miután a parancs befejezte működését, a vi parancsmódba tér vissza.
Számos változata létezik a :w és :q parancsoknak. A fájl mentéséhez, majd a munka folytatásához használjuk a :w parancsot. A vi-ből a :q segítségével lehet kilépni, de csak akkor, ha a munka mentve lett; a minden feltétel nélküli kilépéshez használjuk a :q! parancsot (ez azonban veszélyes!). Próbáljuk meg a :w! parancsot a csak olvasható mód felülbírálásához, ha a :q hibát jelez. Sokan használják a :wq! mentés és kilépés parancsot. A vim-ben vagy az nvi-ben ez ugyanaz, mintha azt írnánk be, hogy :w! majd azt, hogy :q, ha nem volt hiba mentés közben. De a vi néhány más változata ezt úgy kezeli, hogy :w és :q!, ami azt jelenti, hogyha bármilyen hiba történt mentés közben, a munka elveszik. Emiatt jobb szokás a :wq-t használni.
A csere parancs rendkívül hatékony. Legegyszerűbb formájában a :s/X/Y/ az X előfordulását Y-ra cseréli le a pillanatnyi sorban. A pillanatnyi sorban az összes előfordulás cseréjéhez adjuk hozzá a g (global) betűt a parancs végéhez, például :s/foo/bar/g. Rakjuk hozzá a c-t a parancs végéhez, ha meg kívánjuk erősíteni a változtatásokat, például :s/foo/bar/gc. Több sorban való cseréhez két sorszámot írjunk előre, vesszővel elválasztva őket egymástól. Így például a :10,20s/foo/bar/g a „foo” összes előfordulását „bar”-ra cseréli le a 10. és 20. sor között. Az első sorhoz az 1-et, az utolsóhoz pedig a $ jelet kell alkalmazni; az összes sor cseréjéhez viszont a következőt: :1,$s/foo/bar/g (az 1,$, helyett használhatjuk a % jelet, például :%s/foo/bar/g).
Egy másik hasznos parancs a :g/X/. Ez azt teszi lehetővé, hogy bármilyen ex parancsot futtathass az összes X-et tartalmazó sorban. Például az összes olyan sor megjelenítéséhez a képernyőn, amely tartalmazza a „hello” szót, használd a :g/hello/p parancsot (a :p egy ex parancs egy sor képernyőn való megjelenítéséhez). Ezt össze lehet még kapcsolni a :s/// paranccsal. Cseréljük le „foo”-t „bar”-ra azokban a sorokban, amelyek tartalmazzák a „hello”-t: :g/hello/s/foo/bar/g. Mind a :s/X/Y/, mind pedig a :g/X/ parancsokban az X kifejezés egy szabályos kifejezés, nem pedig egyszerűen egy statikus karaktersorozat. Ennek kiküszöbölésére a nomagic beállítást állítsuk kikapcsolt állapotba (lásd lejjebb). A szabályos kifejezések megismeréséhez ajánlom ***Büki András*** Bevezetés a szabályos kifejezések használatába című írását a 69. oldalon, illetve ***Jeoffrey Friedl*** Mastering Regular Expressions című könyvét (O'Reilly & Associates, 2. kiadás, 2002).
Külső program futtatásához használd a :! parancsot. A :!ls megjeleníti a pillanatnyi könyvtár tartalmát. Kényel-

mi szolgáltatás, hogy a parancsban a százalékjel (%) helyettesíti az éppen szerkesztett fájl nevét, azaz a `:!chmod +x %` parancs futtathatóvá teszi a fájlt. A `:!!` parancs begépelése megismétli az előző `:!` parancsot. A parancs-üzemmódbeli `!` paranccsal ellentétben ez nem módosítja a szöveget; a program kimenete, ha egyáltalán van, egyszerűen csak megjelenik a képernyőn.

Beállítások

A vi-ban az `ex` `:set` parancs segítségével be lehet állítani a szerkesztő viselkedését. Például a keresőparancsoknál a kis- és nagybetű alapesetben megkülönböztetett; ennek megváltoztatásához használjuk a `:set ignorecase` vagy a `:set ic` lehetőséget. A legtöbb beállításnak van egy-két karakteres rövidítéssel ellátott változata.

Kétféle beállítás lehetséges: a kétértékű (igaz/hamis) beállítások, és azok, amelyek értéket vesznek fel. Egy kétértékű beállítást a „no” előtaggal lehet kikapcsolni; például a `:set noignorecase` vagy `:set noic` kikapcsolja a kis- és nagybetűfigyelést.

Az értéket felvevő beállítások egyenlőségjelet alkalmaznak, amely mögött a beállítandó érték szerepel. Például a `shift` parancs alapértelmezett szóközértéke (>, <, ^T és ^D) nyolc szóköz (egy tab karakter), a legtöbb programozó azonban két vagy négy szóközt részesít előnyben. A `:set shiftwidth=4` (vagy a `:set sw=4`) parancsot kell használni ahhoz, hogy az összes behúzásparancs négy szóközzel rendelkezzen.

Íme néhány egyéb hasznos beállítás:

- ***wrapmargin=N***
  Bekapcsolja a sortörést; a szám azt határozza meg, hogy hány karakterrel a sor vége előtt kell sortörést alkalmazni, például `:set wm=8`.
- ***az autoindent (kétértékű)***
  Minden sort az azt megelőző értékkel húzza be, például `:set ai`.
- ***magic (kétértékű, alapértelmezett igen, nincs rövidítése)***
  Ez szabályozza a szabályos kifejezések viselkedését a keresőparancsokban, például `:set nomagic`.

Több beállítás egyszeri alkalmazásához a parancsokat eggyé össze lehet kapcsolni, mint például `:set noic wm=8 sw=4 nomagic ai`.

A későbbi szerkesztési tevékenységhez a beállításokat menteni lehet, ekkor be kell rakni őket a *.exrc* fájlba. Mikor a vi-t futtatod, az állományban fellelhető összes `ex` (kettőspont üzemmód) parancs végrehajtásra kerül. Hagyd figyelmen kívül a : kettőspontot, egyszerűen írd be a `set` parancsokat.

Egyéb parancsok

A grafikus szerkesztők gördítősávja azt mondja meg, hol tartózkodsz a szövegben; s rá lehet kattintani, ha a fájl egyes részeit szeretnéd látni. Mivel a vi-nak nincs gördítősávja, az ennek megfelelő billentyűzetparancsokat használjuk.

A CTRL-G a fájl állapotát a pillanatnyi sor számával mutatja meg. A fájl egy meghatározott sorára az `XG` paranccsal lehet eljutni. A `G` önállóan a fájl végére ugrik, vagy az `XG` helyett használhatjuk a `:X`-et.

A CTRL-F egy oldallal előremozgatja a képernyőt, a CTRL-B egy oldallal vissza. Ha félképernyőnként szeretnénk mozogni, a CTRL-D (le) és a CTRL-U (fel) parancsokat kell használnunk.

Néhány egyéb parancs:

- Az `x` parancs letörli a pillanatnyi karaktert (a nagybetűs X a bal oldali karaktert törli le). Számelőtaggal több karaktert is letörölhetünk. Az ezzel a paranccsal törölt karakterek ugyanúgy kerülnek kivágásra, mint a `d` parancs esetében.
- A `~` a pillanatnyi karaktert nagybetűről kisbetűre alakítja vagy fordítva.
- Egy hiba visszavonása az `u` parancscsal történik. A `vim`-ben többszintű visszavonás működik. Az `u` parancs megismétlése a sorban minden egyes lépést visszavon; a CTRL-R billentyű leütése pedig visszateszi őket (redo). A vi más változatainak csak egyszintű visszavonása van; az `u` újbóli kiadása egyszerűen visszavonja az előző lépést, visszaállítva a változtatásokat.
- A nagybetűs `U` minden változtatást visszavon, amit eddig a pillanatnyi sorban tettünk; de miután kiléptél az adott sorból, a szolgáltatás már nem működik többé.
- Írd be a . (pont) parancsot az utolsó szerkesztőparancs megismétléséhez.
- Az utolsó `:s///` parancs megismétléséhez gépeljük be a `&` parancsot (ez azonban csak a pillanatnyi sorban viszi véghez a módosításokat).

További olvasnivaló

A `vim` szövegszerkesztőnek van egy jól használható oktatómodulja, amellyel a parancsok nagy részét gyakorolni lehet. Írd be a `vimtutor` parancsot a parancssorba, vagy add ki a `:help tutor` parancsot a `vim`-en belül.

Linux Journal 2003. 114. szám

William R. Ward (wrw@bayview.com)
1987 óta használja a vim+et. Tanár, író és programozó, feleségével, Hollyval a kaliforniai Mountain View-ban él. Cégük, a Bay View Consulting Services (➲ http://www.bayview.com) Perl, Linux/Unix és az ezekhez kapcsolódó témák oktatásával foglalkozik.

A vi története

A vi szövegszerkesztőt (a név a vizuális szó rövidítése) eredetileg *Bill Joy* írta 1976-ban a kaliforniai Berkeley Egyetemen. A program kezdettől fogva szerepelt a Unix BSD-változatában (Berkeley Standard Distribution). Más Unix-változatok, többek között a Linux is, az évek folyamán átvette a programot. A vi szövegszerkesztő az `ex` leszármazottja, ami viszont az `ed`-en alapult. Ezeket a régebbi szövegszerkesztőket arra tervezték, hogy a távírón használják őket, és egyszerre csak egy sor szöveget voltak képesek megjeleníteni – a vi volt az első unixos teljes képernyőt kihasználó szövegszerkesztő.

Az eredeti vi-forráskód a szerzői jogok miatt nem volt nyilvános (egészen néhány hónappal ezelőttig), de sok klónját elkészítették. Az egyik, amelyik a legtöbb Linux-terjesztésben szerepel, a `vim` (továbbfejlesztett vi – vi improved). A jelenlegi szabványos BSD szövegszerkesztő az nvi (új vi – new vi). További vi-szövegszerkesztő például az elvis, a vile és a stevie. Ezek tartalmazzák a vi szolgáltatásait és a saját kiegészítéseiket.

KAPCSOLÓDÓ CÍMEK

A vi szövegszerkesztő és klónjai
➲ http://www.math.fu-berlin.de/~guckes/vi/index.php3
A vi-kedvelők honlapja ➲ http://www.thomer.com/vi/vi.html