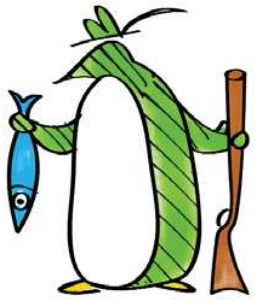
Programvadászat



Vectorlinux

Előző számunk CD mellékletén a KNOPPIX v3.4-et tettük közkinccsé, ami egy Debian alapú linux-rendszer. Most a Vectorlinuxon a sor, amely egy Slackware alapú, szintén CD-ről indítható rendszer, sokkal kisebb helyen elfér, és sokkal kevesebb programot tartalmaz. Mindezek ellenére egy nagyon gyorsan beállítható, és nagyon kényelmesen használható rendszerről van szó. Nélkülöz minden felesleges csicsát, és lényegretörő felépítéssel rendelkezik, nem csoda hát, hogy hamar megkedveltem.

Rendszerindítás

Helyezzük be a korongot a meghajtóba. Ha a rendszerünk cd-ről való indításhoz van beállítva, akkor az ISOLINUX felirat megjelenése után azonnal el is indul a Vectorlinux. Ezután az első kérdés a billentyűzet kiosztás beállítására vonatkozik, majd affelől érdeklődik a program, hogy vajon az egerünknek van-e kereke? A monitor felbontásának a beállítása is gyerekjáték, végül pedig a hálózat beállítása következik:

- adjuk meg a gép nevét
- a domain nevet
- majd válasszuk ki a kívánt IP cím beállítási módot (statikus, DHCP), ha a statikust választjuk, akkor még egy pár kérdésre válaszolnunk kell.

A rendszer megmutatja a felismert és beállított eszközök listáját, majd itt entert ütve ténylegesen elindul a rendszer.

Bejelentkezés

A bennünket fogadó képernyő ékes angolsággal közöl néhány fontos információt, nevezetesen azt, hogy a felhasználónév az `root`, a jelszó `vector`, a `vasm` paranccsal a beállítómenübe juthatunk, ahol beállíthajuk a fel nem ismert eszközeinket és a `startx` elindítja a grafikus felületet.

A grafikus felület

Miután testreszabtuk a rendszert, jöhet a móka a grafikus felülettel. Adjuk ki a `startx` parancsot, és hamarosan egy menüben találjuk magunkat, válasszuk ki melyik ablakkezelőt akarjuk használni és máris elindult teljes pompájában a Vectorlinuxunk. Én, mint megszállott XFCE rajongó, természetesen ezt választottam (a képen az IceWM felület látható).

vasm

A fontosabb menüpontok:

- AUTOSET: hangkártya, egér, X, hálózatikártya, cdrom automatikus beállítása
- XSETUP: az X beállítása
- PKGTOOL: Slackware csomagkezelő
- ADMIN: felhasználók hozzáadása/eltávolítása

A programok

Programok terén igazán nem lehet panasz, bekerült a négy nagy kedvencem a terjesztésbe, az első a FireFox (FireBird) böngésző (sajnos csak a 0.6-os változat), mely kicsi, kevés erőforrást igényel, mégis fantasztikusan testreszabható és hihetetlenül gyors. A másik kedvencem az Abiword, szintén az erőforrástakarékossága miatt. Az OpenOffice.org kicsit elnyomja ezt és az ehhez hasonló programokat, pedig igazán megérdemlik a figyelmet, mindent tud, amit valaha is elvárhatunk egy szövegszerkesztőtől. A harmadik a Gaim csevegőprogram, ezen keresztül azonnal kapcsolatba léphetünk az ismerőseinkkel, így nélkülözhetetlen számomra, és az összes erre a célra készített program közül ez vált be a legjobban. A negyedik pedig mi más lehetne mint a Gftp, amely egy nagyszerű ftp ügyfélprogram. Természetesen ez nem a programok teljes listája, csak az általam nagyrabecsült és mindennapos használatban lévőké. Aki kóborol egy kicsit a menükben, sok más egyebet is találhat.

A többiek

A Vectorlinux méretének köszönhetően sok más egyéb is felkerülhetett a mellékletre, az egyik fontos és valószínüleg sokak által nagyrabecsült ➲ http://www.jatekport.linuxuser.hu tükre, ahol rengeteg játék linux alatt történő telepítése, beállítása, esetleges trükkök gyűjteménye olvasható. Aki linux alatt játszani szeretne annak kötelező olvasmányok az itt található leírások.

Friss programok

A legfrisebb Opera, Mozilla, FireFox is helyet kapott, csakúgy mint a rendszermag, MySQL, KOffice.

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

Diétás fogyasztás

Az AMD új tagokkal egészítette ki alacsony fogyasztású processzorainak családját. A hétköznapi programok futtatására használható, ám könnyű és kis méretű készülékekbe szánt AMD Geode NX processzorok mellé egy újabb termékcsalád, az AMD Geode GX processzorok sora társul. A Geode NX processzorok 1 GHz fölötti órajelekkel működnek, energiafogyasztásuk 10 W körül mozog, vagyis a nagy teljesítmény és az alacsony fogyasztás előnyeit próbálják ötvözni. A Geode GX lapkák még ezt is túl szárnyalják, mivel 500 MHz körüli órajellel futnak, tehát kisebb teljesítményt nyújtanak, viszont akár 1 W energiával is beérik. Mindkét lapkatípus akár aktív hűtés nélkül is használható. Elsősorban vékony ügyfelekben, felső kategóriás nyomtatókban és hálózati berendezésekben találkozhatunk majd velük. A Geode név korábban a National Semiconductor termékeit fedte. A céget az AMD múlt év augusztusában vásárolta fel. A Geode processzorok piaci szegmensében erős a verseny, hiszen a Transmeta, a VIA és az Intel egyaránt rendelkeznek ilyen területekre szánt lapkákkal.

➲ http:\\www.amd.com

Lenyúlták?

A Cisco részéről megerősítették annak a két fájlnak az eredetiségét, amelyek egy orosz biztonsági témájú weboldalon tűntek fel. Rejtély, hogy az állítólag összesen 800 MB-ot kitevő Cisco IOS kód hogyan és honnan szivárgott ki, de sem a Cisco alkalmazottait, sem üzleti partnereit nem vádolják a szándékos károkozással. Főleg, hogy a kárt ebben az esetben elég nehéz lenne megállapítani. Hiába rendelkezik valaki a kóddal, megfelelő céleszközök hiányában valószínűleg lefordítani sem tudja. Biztonsági kockázatot is csak akkor jelent a kód nyilvánosságra kerülése, ha sikerül kiaknázható hiányosságot találni benne – márpedig ilyet a nemrég ellopott Windows forráskódban csak egyet találtak, annak ellenére hogy a Windows kódja valószínűleg nagyobb érdeklődésre tartott számot.

➲ http:\\ www.cisco.com

McAfee LinuxShield

A Network Associates bemutatta kifejezetten linuxos kiszolgálókra készült McAfee LinuxShield víruskereső termékét. A McAfee víruskereső motorra épülő LinuxShield az ilyen programoktól megszokott és elvárható módon önműködően frissíti vírusadatbázisát, képes az archivált fájlok átvizsgálására és a gyanús állományok heurisztikus felismerésére. Igaz, hogy a Linuxot elég kevés vírus vagy féreg fenyegeti, a Network Associates szerint azonban egyrészt jobb félni, mint megijedni, másrészt vegyes környezetben fontos lehet a linuxos kiszolgálók védelme a windowsos gépekről kiinduló támadásoktól. Fontos továbbá a windowsos gépeket fenyegető kártevők minél gyorsabb elfogása – akár már a linuxos kiszolgálókon. A Windowsok számára veszélyes programok – mielőtt tényleges céljukat elérnék – különféle nem windowsos hálózati készülékeken és kiszolgálókon haladhatnak keresztül, nem egy esetben elektronikus levélben terjednek, logikus lépés tehát, hogy már ezeknél a védelmi vonalaknál megakadályozzuk továbbjutásukat.

A LinuxShield hivatalosan a Red Hat és SuSE terjesztéseket támogatja, erőforrás-igényei viszonylag szerénynek mondhatók.

➲ http:\\www.networkassociates.com

Ezt ne

A Red Hat programozója, Ingo Molnár közzétette az első változatát annak a foltnak, amely az NX (No eXecute, nem végrehajtható) jelző támogatását valósítja meg a Linux rendszermag 2.6-os sorozatának tagjaiban. Az NX jelző, amelyet Enhanced Virus Protection névvel szélesebb körben elsőként az AMD tett elérhetővé, és amelyet azóta a Via, a Transmeta és az Intel is támogat, a memóriabeli adatszegmensek tartalmának programkódként való futtatását tiltja a processzor számára, és ezzel hathatós védelmet biztosít a puffertúlcsordulásos támadások ellen. Mivel az NX jelző használata csak nagyon kevés program futását zavarja, támogatása a későbbi rendszermagokban valószínűleg alapértelmezett lesz.

Opteronok négyesével

A Tyan K8QS Pro jelzéssel bemutatta első olyan alaplapját, amely négy darab AMD Opteron processzor fogadására is képes.

A megjelenésében is tiszteletet parancsoló alaplap a négy 800-as sorozatú processzor mellett 16 foglalatában összesen 32 GB memóriát képes fogadni, kettő darab Gigabit Ethernet csatlakozóval, egy ATi Rage VGA-vezérlővel, egy hagyományos, 32 bites és két 64 bites PCI foglalattal, továbbá kettő darab 64 bites PCI-X foglalattal rendelkezik; kiegészítőként SATA RAID és SCSI-vezérlő kérhető hozzá. A cég az alaplapot elsősorban nagy számítási teljesítményt igénylő alkalmazásokhoz, nem kiszolgálókhoz szánja; például, ha hinni lehet a híreknek, a kínai Dawning ilyen alaplapokkal fogja felépíteni TC4000A fürtjét, amelybe 640 csomópont és 2560 Opteron processzor kerül majd. Elkészülte után a fürt számítási teljesítménye alapján a világ 10 legerősebb számítógépe közé fog tartozni – ilyen referenciát alighanem bármelyik cég szívesen fogadna.
➲ http:\\www.tyan.com

CrossOver Office – töretlenül

A CodeWeavers bejelentette a CrossOver Office 3.0-s változatát. Az elsősorban irodai alkalmazások futtatására alkalmas keretprogram a Lotus Notes 6.51, a Microsoft Outlook XP és a Microsoft Project támogatásával bővült. Lényeges változás, hogy a program immár kétféle változatban is létezik, a Professional kiadást cégeknek szánják, a Standard kiadás viszont kedvező, 40 dolláros, vagyis körülbelül 9000 forintos árával az otthoni felhasználókat célozza meg. Az olcsóbb változathoz hat havi változatkövetés és támogatás jár, ez a telepítésre és korlátozott mértékben a használatra vonatkozik, míg a drágább kiadáshoz mindez 12 hónapon át és magasabb szinten jár. Akik már megvásárolták a CrossOver Officet, azok a Professional változathoz járó szolgáltatásokat jogosultak igénybe venni. Ezzel egy időben mint önálló termék megszűnt a CrossOver Plugin, mely a továbbiakban a CrossOver Office termékek beépített összetevője lesz.
➲ http:\\www.codeweavers.com

Felforgató Sun

A Sun Microsystems a jövőben elsősorban előfizetés alapján, rugalmasan igénybe vehető szolgáltatásokra és termékekre kívánja alapozni tevékenységét, derül ki a cég terveiből. A Sun bevételeinek jelenleg nagyjából 20 százaléka származik rendszeresen befolyó összegekből, 80 százalék pedig egyszeri értékesítésből. A jövőben ennek az aránynak pontosan a fordítottját szeretnék produkálni, amivel párhuzamosan az értékesítéssel összefüggő költségek csökkenésére is számítanak, így a cég jóval nagyobb nyereséget könyvelhet majd el. A változások egyik jele a Sun Preventative Services szolgáltatáscsomag kialakítása, amely – adatközpontok üzemeltetéséhez – mintegy 100 már eddig is létezett szolgáltatást fog össze egy csokorba.
Nem kevésbé formabontó a Sun azon ötlete, mely szerint néhány éven belül a vas ingyenessé válthat, ha a felhasználó előfizet bizonyos szolgáltatásokra. A Sun stratégái azonban nem álltak meg itt a formabontó ötletek tekintetében: feltett szándékuk, hogy néhány éven belül maga a gép is ingyenessé válhat, feltéve, hogy a felhasználó előfizet bizonyos szolgáltatásokra. Mivel alaptétel, hogy ingyen semmi sincs, legfeljebb más formában fizetjük meg az árát, az ilyen felvetéseket érdemes fenntartással fogadni, ám azt sem szabad figyelmen kívül hagyni, hogy a cég megtette az első lépést, Solaris operációs rendszerrel telepített, belépő szintű Opteron kiszolgálót ad azoknak, akik három éven keresztül 1499 dolláros előfizetési díjat fizetnek. Hogy a meglepetések sora folytatódjon, a Sun megerősítette, hogy legalább részlegesen nyílt forrásúvá fogja tenni Solaris operációs rendszerét.
A nyitással kapcsolatosan fontosabb részleteket még nem tudni, így egyelőre rejtély, hogy pontosan milyen feltételeket fog szabni a Sun, illetve mely részek és mikor válnak szabadon hozzáférhetővé. A Sun részéről nagy horderejű döntés a Solaris szabaddá tétele, hiszen az operációs rendszer a cég egyik legfontosabb szellemi vagyontárgya – igaz, más Sun-programok is váltak már szabaddá, elég csak az OpenOffice irodai csomagra gondolni. Egyelőre bizonytalan, hogy a Solaris rendszermagjának esetleges megnyitása milyen hatásokkal lehet a Linux fejlesztésére, az viszont sejthető, hogy az operációs rendszer fenntartása egyre kevésbé érheti meg a Sunnak, éppen azért, mert az olcsó x86 alapú gépeken futó linuxos kiszolgálók elvonták a vásárlókat.
➲ http:\\www.sun.com

Beágyazott szomszédság

A francia Jaluna SA bejelentette Jaluna/OSware termékének elérhetőségét. Magát a terméket leginkább egyfajta beágyazott keretrendszernek lehetne nevezni, mely azonos eszközön egyszerre több működési környezet, operációs rendszer használatát és futtatását is lehetővé teszi. Hasonló megoldásokkal a nagygépeknél is találkozhatunk, ám a célcsoportot ezúttal a hálózati infrastrukturális elemek, vezeték nélküli eszközök, IP alapú set-top-boxok és egyéb távközlési berendezések gyártói alkotják. Ezeknél az eszközöknél egyre gyakrabban merül fel a Linux futtatásának igénye, ám a régebbi valós idejű operációs rendszerek szolgáltatásairól, valamint az ezek alá írt alkalmazásokról sem lehet egyik pillanatról a másikra lemondani.
A több processzor párhuzamos kezelésére is képes Jaluna/OSware segítségével logikai partíciókra oszthatók a gép erőforrásai, így egyszerre több operációs rendszer is futhat egymás mellett. Mivel így külön processzorra, költségnövelő megoldásokra nincs szükség, az eszközök árának emelése nélkül is kellő gyorsasággal vezethetők be a legújabb megoldások.
A Jaluna/OSware az egyes rendszereket egymástól függetlenül kezeli, a megzavarodott alkalmazások és operációs rendszerek önmagukban is újraindíthatók, továbbá frissítésük is zavartalanul elvégezhető. A keretprogram x86, PPC és ARM9 alapú gépekre érhető el.
➲ http:\\www.jaluna.com

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

Új termékek

HP Compaq – új, üzleti számítógépcsalád

A közelmúltban a HP az asztali gépek egy új családját mutatta be, amellyel elsősorban az üzleti célú vevőkört célozza meg. Kiépítettségük alapján a gépek három kategóriába sorolhatók (Compaq 2000-es, 5000-es és 7000-es sorozat).

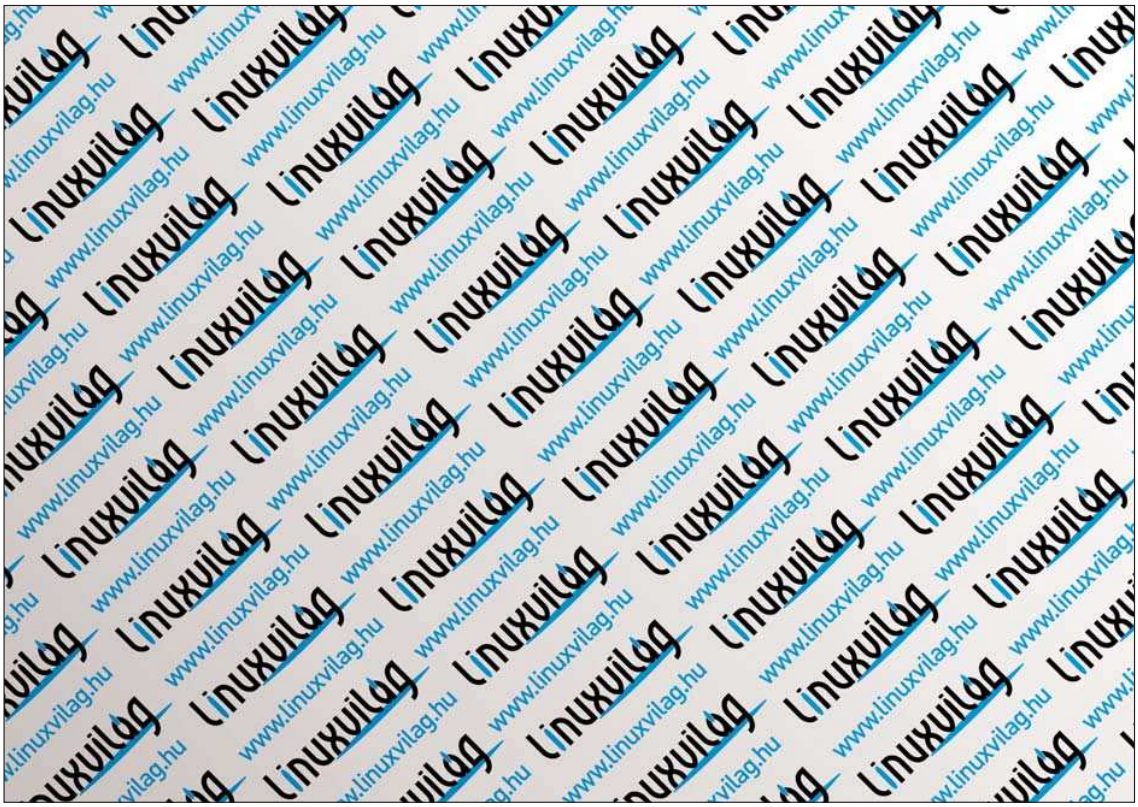
A dx2000-es sorozat tagjaiban négy DIMM foglalattal, illetve nyolc USB 2.0-ás csatlakozóval felszerelt alaplap található. Előbbiek egy- és kétcsatornás üzemmódban is képesek működni. A lapokba legfeljebb 3.0 Ghz-es Pentium 4 vagy Celeron processzor illeszthető, és természetesen a HyperThreading funkciót is támogatják. Mindehhez legfeljebb 80 Gb-os merevlemezt, 1 GB DDR SDRAM-ot illetve különböző optikai meghajtókat kérhetünk. A dc5000 sorozat tagjai mikrotoronyba vannak beépítve. Akárcsak az előbbi sorozat elemei, ezek a házak is csavarhúzó nélkül szerelhetők. A gépek az előbbi kiépítésnél említettel azonos CPU mellett legfeljebb 4 GB DDR SDRAM-ot és 160 GB-os merevlemezt tartalmazhatnak. A legnagyobb teljesítményt és a legváltozatosabb kiépítést természetesen a 7000-es sorozat kínálja. Valamennyi gépen Mandrake Linux operációs rendszer fut.

Kapcsolat: Hewlett-Packard Company, 3000 Hanover Street, Palo Alto, California 94304, 800-752-0900,
➲ http:\\www.hp.com/products/desktops

Gforge Collaborative Station kiszolgálók

A GForge csoport teljesen beállított GForge kiszolgálókat kínál. Ez egy olyan csoportos szoftverfejlesztést támogató környezet, amely lehetővé teszi, hogy kifejezetten nagy méretű fejlesztési projekteket egyetlen közös felületen keresztül lehessen kezelni illetve nyomon követni. A GForge RackMount I és II, illetve a GForge Collaboration Station I és II négy olyan kiszolgálótípus, amelyeket egyszerűen a dobozból kivéve és bekapcsolva azonnal használatba vehetünk. A gépeken egy Debian terjesztés, valamint a GForge saját szoftvere fut, amelyet a cég eleve a megrendelő igényeihez igazítva szállít. A vásárlónak természetesen utólag is lehetősége van a rendszerek testreszabására. A gépekhez egy év garancia és változatkövetés jár.

Kapcsolat: The GForge Group, 512 129th Street, Urbandale, Iowa 50323, 408-907-2744,
➲ http:\\www.gforgegroup.com



Mi újság a rendszermag fejlesztése körül?

A FAT fájlrendszer karbantartása továbbra is folyik, *Frodo Looijaard* nemrég néhány furcsa, a könyvtárindexet szokatlan karakterekkel leíró FAT-megvalósítás linuxos támogatását is megoldotta. A FAT karbantartásával az az egyik baj, hogy rengeteg különböző megvalósítás Linux alatti támogatásáról kellene gondoskodni, ami a véges erőforrások miatt gyakorlatilag kivitelezhetetlen. A másik jelentős probléma az, hogy még maga az MS-DOS-os változat sem mindig a hivatalos leírásnak megfelelően viselkedik, márpedig ilyenkor nem marad más, mint a *H. Peter Anvin* által megfogalmazott WWDD (What Would DOS Do – mit tenne a DOS) szabály alkalmazása. A szabad programok világának újra és újra felvetődő kérdése, hogy meddig érdemes követni a szabványokat, és mikor kell elfordulni tőlük, annak érdekében, hogy valami hatékonyabb fejlesztési irányba fordulhassunk.

Michael A. Halcrow egy titkosított fájlrendszeren kezdett el dolgozni. Elképzelése szerint a legjobb megoldás az lesz, ha teljesen átlátszó titkosítási réteget hoz létre a gyökér fájlrendszer felett, a fájlokat pedig titkosítottként vagy titkosítatlanként jelöli meg. A titkosító kulcsok nem könyvtárakhoz vagy blokk eszközökhöz, hanem fájlokhoz fognak tartozni, de egy-egy könyvtárat is meg lehet majd jelölni titkosítottként, ilyenkor a benne tárolt fájlok alapértelmezett beállításként titkosítottak lesznek. A kulcsokat és az egyéb metaadatokat kiterjesztett jellemzőkben kívánja tárolni. A normál fájladatokat, mint a méret és a vonatkozó engedélyek, szintén titkosítani szeretné, persze csak ha sikerül ezt megvalósítania. A törölt fájlokat gondosan le fogja radírozni a lemezről, ezzel elvéve a helyreállítás lehetőségét. Mindezek mellett számos további érdekes szolgáltatás megvalósítására is számíthatunk.

Vojtech Pavlik készített egy, a bemeneti illesztőprogramokkal kapcsolatosan felmerülő gondokról szóló GYK-t. Olyan kérdéseket tárgyal, mint például a bemeneti illesztőprogramok listájának lekérése, az egér vagy a billentyűzet rendellenes működésének megszüntetése stb. Ez az első alkalom, hogy mindezekről a bajokról valamilyen írás született, mióta – a 2.5-ös sorozat fejlesztésekor – újraírták a bemeneti réteget. Sok embernek jelentett örömet, hogy kérdéseikre végre egy központi gyűjteményben találhatja meg a válaszokat.

Mióta a 2.5-ös fejlesztői fában oly sok kavarás és kavarodás közepette újraírták az IDE illesztőprogramot, a külső modulként való betölthetőségének megvalósítására irányuló munkákat kicsit félretették, amíg a kód fennmaradó része kellően üzembiztos állapotba nem kerül. *Alan Cox* maga is elutasította a különféle fejlesztőktől származó, az IDE modulként való betölthetőségét célozó foltokat. *Witold Krecicki* nemrég tett próbálkozása nem tűnt rossznak, de úgy látszik, *Bartlomiej Zolnierkiewicz* foltja a közeljövőben nagyobb eséllyel kerül be a 2.6-os fába. A 2.5-ös időkben ő volt az, aki keményen dolgozott az IDE illesztőprogram fejlesztésén, láthatóan a modul foltjára is elég sok időt fordított, és remélhetőleg minden felmerült kérdést megfelelően sikerült kezelnie.

Kiderült, hogy a MAINTAINERS fájlban hibajelentések küldésére alkalmasként felsorolt levelezési listák némelyikére a felhasználóknak először fel kell iratkozniuk, ha visszajelzést akarnak küldeni. A levelezési listák világában természetesen nem szokatlan ez a megoldás, ám a Linux rendszermag fejlesztői között volt egy megállapodás, hogy a MAINTAINERS fájlban hibajelentések küldésére szolgálóként megadott listák bárkinek a levelét fogadják. Ez volna ugyanis a legjobb módja annak, hogy a hétköznapi felhasználók jelezhessék az általuk talált hibákat. A feliratkozás kötelezővé tételének egyik fő oka a levélszemét elburjánzása, illetve a fejlesztői levelezési listáktól való távol tartásának szándéka volt. Igaz, hogy ezt a gondot sokféle módszerrel meg lehetett volna oldani, ám arról is döntés született, hogy a MAINTAINERS fájlban szereplő listák egy része nem lesz teljesen nyitott, amíg létezik mellette egy másik lista, ahova a hibajelentések elküldhetők.

A Linux rendszermag újabb, Cooperative Linuxnak nevezett átültetése jelent meg. A User-Mode Linuxhoz (UML) hasonlóan a Cooperative Linux nem meghatározott vastípusra készült, hanem meglévő rendszer felett teszi lehetővé a Linux futtatását. A Cooperative Linux Windows 2000 és Windows XP alatt biztosan működik. Készítői remélik, segítségével sok más operációs rendszert futtató érdeklődő próbálhatja ki a Linuxot úgy, hogy meglévő rendszerét nem kell Linux-telepítésre cserélnie. A Cooperative Linuxból adott gépen akár több példány is futtatható egyszerre, és hamarosan Linux feletti futtatását is meg fogják oldani.

Zack Brown

Linux Journal 2004. május, 121. szám

Linux-index

1. A megkérdezett vállalatok ekkora százaléka fogja növelni Linux-vonatkozású kiadásait a következő évben: 60
2. A megkérdezettek ekkora százaléka fogja 6-10 százalékkal növelni Linux-vonatkozású kiadásait: 17
3. A megkérdezettek ekkora százaléka fogja 10 százaléknál nagyobb mértékben növelni Linux-vonatkozású kiadásait az elmúlt évhez képest: 43
4. A megkérdezettek ekkora százalékának legnagyobb Linux-vonatkozású kiadása az operációs rendszer újabb változatának telepítésével lesz kapcsolatos: 35
5. A megkérdezettek ekkora százaléka állította, hogy legnagyobb Linux-vonatkozású kiadása a kiszolgáló gépek bővítése lesz: 30,5
6. A megkérdezettek ekkora százaléka vallotta, hogy legnagyobb Linux-vonatkozású kiadása az adatközpontok áttelepítése lesz: 29,5
7. Az UAL Loyalty Services ennyi százalékkal kívánja növelni a gépbeszerzésre fordított összeget: 10
8. A megkérdezettek ekkora százaléka nyilatkozott úgy, hogy a linuxos fejlesztések mellett az adatközpontok áttelepítése élvez kiemelt figyelmet a 2004. évi tervezetek között: 30
9. A fentiek ekkora százaléka nyilatkozott úgy, hogy az áttérésre legfeljebb 99 ezer dollárt fog költeni: 13
10. A fentiek ekkora százaléka véli úgy, hogy százezer és félmillió dollár közötti összeget fog költeni: 52,7
11. Ennyi hónapig tart az informatikai szakembereknek egy belső alkalmazás teljes bevezetése: 9
12. Egy francia tanulmány szerint ennyi PC esetében várható a Microsoft Windows lecserélése Linuxra: 17 ezer
13. A fenti tanulmányban szereplő alkalmazások száma: 600
14. A fenti tanulmányban szereplő kiszolgálók száma: 400
15. Ennyi rendőrségi számítógépet állítottak át Linuxra Windowsról a németországi Alsó-Szászországban: 11 ezer
16. Az európai számítógépek ennyi százalékán fut jelenleg Linux: 15
17. A megkérdezettek ekkora százaléka adta a „Várok" választ az eWeek felmérésében, amikor azt firtatták, vajon hajlandóak-e 2006-ig várni a Microsoft Longhorn operációs rendszerének megjelenésére: 35
18. A megkérdezettek ekkora százaléka adta a „Linuxra váltok" választ az eWeek felmérésében, amikor azt firtatták, vajon hajlandó-e 2006-ig várni a Microsoft Longhorn operációs rendszerének megjelenésére: 40
19. A Red Hat a Linux-terjesztések közötti népszerűségi listán elfoglalt helye a Netcraft felmérése szerint: 1
20. A leggyorsabb növekedést felmutató Linux-terjesztések között a Debian helye a Netcraft felmérése szerint: 1

Források

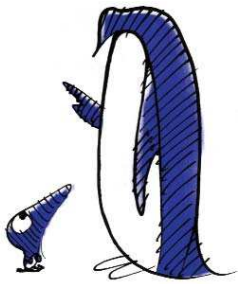
1–10: SearchEnterpriseLinux.com
11: IDC
12–16: ZDnet
17, 18: eWeek
19, 20: Netcraft

Ők mondták

Hogyan írsz az emberek számára hasznos programokat, ha nem tudod, hogy milyen szolgáltatásokat és hogyan használnak? Válaszolj két kérdésre: A nyílt forrású fejlesztésekkel kapcsolatosan beérkező hibajelentések hány százaléka származik pénzügyi szolgáltató cégek alkalmazottaitól? Mely iparágak mutatják fel a legjobb foltközzététel/futó példányok száma arányt?
Robert Lefkowitznak az OSCON rendezvényen elhangzott beszédéből

Részt venni valamiben és utánozni valamit – a kettő közötti határvonal bizony erősen elmosódott.
Scott Adams
➲ www.photodude.com/index.shtml

A javítás egyik módja „a diszkriminációs hipertér dimenzióinak bővítése" (nem, ezt a mondatot nem én mondtam).
Bill Yerazunis
➲ CRM114 & Mailfilter HOGYAN



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

Lassú mentések

Samba kiszolgálónkat (Red Hat 7.3, PII) egy új gépre tettem át (Red Hat 9, P4). Egy `cron` feladattal, `smbtar` használatával naponta mentést készítek a megosztásokról. Az up2date segítségével az összes foltot telepítettem. A hiba: a biztonsági mentés parancsfájlja sokkal lassabban fut le az új gépen, mint a régin. Mi lehet ennek az oka?
Suttó Zoltán, sutto.zoltan@rutinsoft.hu

Elsőként az ethernetkártya illesztőprogramjára gondolnék. Ellenőrizd, hogy a legújabbat és a legjobbat használod-e. Jómagam az ethernetkártyák önműködő sebességegyeztetésének működésével kapcsolatosan is tapasztaltam gondokat. Ellenőrizd, hogy 100 Mbps sebességen, teljes kétirányú módban üzemel-e a kártya.
Christopher Wingert, cwingert@qualcomm.com

Ha elemezni akarod a hibát, kezdd azzal, hogy az `smbtar` parancsfájlt úgy futtatod, hogy a nyomkövetést előzőleg bekapcsolod. Ezt a `bash -x` kapcsolójával teheted meg; az `smbtar` ugyanis egy héjprogram. Utána csak figyelned kell, hogy mely parancsok végrehajtása tart sokáig. Ennél határozottabban is indíthatsz, ha készítesz egy másolatot a parancsfájlról, majd ezt viszonylagos és abszolút időbélyegeket készítő hívásokkal szórod tele a külső parancsok hívásai között. Az időbélyegeket profilfájlba írhatod, de a naplózó parancs segítségével egyszerűen elküldheted a rendszernaplóba is. A következőkhöz hasonló héjkifejezéseket kell használnod:

```
KEZDO_IDOPONT="$(date +%s)";
VISZONYLAGOS_IDO="$KEZDO_IDOPONT"
VISZONYLAGOS_IDO="$(( $(date +%s) -
➥$VISZONYLAGOS_IDO))"
```

Így megkaphatod a pontos időt, vagyis az 1970 óta eltelt másodpercek számát. A parancsfájl teljes futási ideje tehát az aktuális időpontból kivonva a `$KEZDO_IDOPONT` értéke, amit a parancsfájl első sorában adtunk meg.
Gondolj arra is, hogy a beállításokban bekövetkezett változások miatt bosszantó késleltetések léphetnek fel a hálózati névszolgáltatások használatában. Előfordulhat például, hogy a régi */etc/hosts* fájl révén bizonyos visszirányú DNS-lekérdezések működnek, de az új telepítéssel ezek elvesztek; az is lehet, hogy a régi */etc/nsswitch.conf* csak helyi fájlokat vizsgált, az új viszont valamilyen módon NIS, LDAP vagy winbind (MS Windows tartományi) forrásokat is lekérdez. A winbind csak a 7.3-mas utáni Red Hat rendszerekbe került be, lehet, hogy ez a bűnös.
Ha a lehető legnagyobb teljesítményt szeretnéd kihozni egy gépből, akkor először méréseket kell végezned (profilkészítés), meg kell találnod a szűk keresztmetszeteket (elemzés), majd meg kell szüntetned ezeket, ha lehetséges (finomhangolás). A szűk keresztmetszetek kiiktatása során általában az adott környezetben szükségtelenül elvégzett feladatok után is kutakodunk; ilyenek például a hálózati címtárszolgáltatások felé indított, helyi fájlok használatával is kiváltható lekérdezések.
Sokszor teljesen más szemlélettel kell újragondolni az adott feladat elvégzését. A te esetedben erősen elgondolkodnék azon, hogy az `smbtar`-t használjam-e a Samba megosztások mentésére. Megtehetnéd például, hogy `rsync` segítségével a megfelelő, vagyis a megosztott könyvtárakat egyetlen nagyméretű lemezre szinkronizálod, arra a gépre, amelyben a szalagos meghajtó is található. Ezután az egészet felmásolhatod szalagra.
Jim Dennis, jimd@starshine.org

Lehet, hogy az új gép nem tudja kellő sebességgel továbbítani az adatokat a merevlemez felé. Feltételezem, hogy IDE meghajtókat használsz. Néhány Linux-terjesztés alapértelmezett telepítésnél nem feltétlenül engedélyezi a DMA-t, ezért telepítés után külön kell bekapcsolni. A meghajtót a `hdparm` segítségével tudod ellenőrizni és kipróbálni. Esetemben a rendszer a */dev/hda* meghajtóról fut:

```
[root@hamtop ~]# hdparm /dev/hda

/dev/hda:
multcount    = 16 (bekapcsolva)
IO_support   =  0 (default 16-bit)
unmaskirq    =  0 (kikapcsolva)
using_dma    =  1 (bekapcsolva)
keepsettings =  0 (kikapcsolva)
readonly     =  0 (kikapcsolva)
readahead    =  8 (bekapcsolva)
geometry     = 3648/255/63, sectors =
➥58605120, start = 0
```

A `using_dma` sort kell ellenőrizned. Ha nulla szerepel benne, akkor lehet, hogy megvan a hiba oka. Próbáld kiadni a `hdparm -d1 /dev/hdX` parancsot, ahol X a meghajtód jele. Aztán próbáld ki:

```
[root@hamtop ~]# hdparm -tT /dev/hda

/dev/hda:
Timing buffer-cache reads:   128 MB in
➥0.82 seconds = 156.10 MB/sec
Timing buffered disk reads:  64 MB in 2.68
➥seconds = 23.88 MB/sec
```

A mérés alapján a pufferelt írások sebességének jelentősen emelkednie kell ahhoz képest, amit ugyanennek a próbának a DMA engedélyezése nélkül való futtatásával kaptál. A DMA engedélyezése után gondosan vizsgáld meg a meghajtó működését, mielőtt rábíznád a rendszeredet; ritkán előfordulhat, hogy a régebbi meghajtók rosszul működnek vele. Ha ezzel sikerült elhárítani a hibát, olvass utána, hogy

a saját terjesztésednél hogyan tudod az engedélyezést a rendszerindítás alatt elvégezni. Red Hat alatt ezt a beállítást a ***/etc/sysconfig/harddisks*** fájl tartalmazza.
Timothy Hamlin, thamlin@zeus.nmt.edu

Hogyan állítható helyre a rendszermag .config fájlja?

A gépemen 2.4.22-esre frissítettem a rendszermagot, de indításkor továbbra is felkínálja a választás lehetőségét a 2.4.20-8 és a 2.4.22 között. A baj az, hogy nincs meg a 2.4.20-8 rendszermagomhoz tartozó ***.config*** fájl. Szeretném megtudni, van-e olyan parancs, amivel újra elő tudnám állítani?
Jan Nicolas Myklebust,
jan-nicolas.myklebust@cnes.fr

Ha ez az alapértelmezett Red Hat rendszermag, akkor bontsd ki a rendszermag forráscsomagját, majd a ***/usr/src/linux-2.4/configs*** könyvtárból másold ki a ***.config*** fájlt.
Christopher Wingert, cwingert@qualcomm.com

A 2.4.x és korábbi rendszermagoknál nincs a ***.config*** fájl előállítására használható parancs. Az újabb, 2.6-os rendszermagok esetében egy fordítási idejű kapcsolóval lehet ezt megtenni.
Jim Dennis, jimd@starshine.org

bash, előzmények nélkül

A 2004. februári szakmai tanácsok között szerepelt egy kérdés a tévedésből beírt adatoknak a bash előzmények közül történő elrejtésével kapcsolatban. Ha kijelentkezés helyett a `kill -9 $$` paranccsal kilövöd a saját bash folyamatodat, akkor nem fogja a lemezre írni az előzményeket.
Jack Coates, jack@monkeynoodle.org

Üres területen nem lehet lemezrészt létrehozni

Red Hat 9 rendszerem jelenlegi lemezrészei a következők:
hda1 20 GB Windows
hda2 7 GB Linux /
hda3 12 GB Linux /usr
swap 1 GB

A GNU parted segítségével a hda1 méretét 8 GB-ra csökkentettem, amivel kaptam 12 GB üres területet. Ezen a használaton kívüli 12 GB-on egy új linuxos lemezrészt szeretnék létrehozni. Gondom az, hogy a parted `mkpart` parancsa egyszerűen közli, nem tud lemezrészt létrehozni, az fdisk n parancs pedig azt mondja, először töröljek le egy lemezrészt, mielőtt újat hoznék létre.
Hiroshi Iwatani, HGA03630@nifty.ne.jp

Úgy tűnik, mintha már lenne négy elsődleges lemezrészed, márpedig ennél több nem lehet. Törölnöd kell valamelyik lemezrészt, majd logikai lemezrészt kell a helyére tenned, amelyben számos további lemezrészt hozhatsz létre.
Én a helyedben kikapcsolnám a csereterületet, törölném a lemezrészét, létrehoznék egy az összes üres részre kiterjedő logikai lemezrészt, hozzáadnék egy új csereterületet, lefuttatnám az `mkswap` parancsot, hozzáadnék egy új adattároló lemezrészt, megformáznám és végül bekapcsolnám a csereterületet. A ***/etc/fstab*** fájlt is át kell írnod úgy, hogy tartalma tükrözze a csereterületre vonatkozó változásokat, valamint az új adattároló lemezrésznek is szerepelnie kell benne.
Christopher Wingert,
cwingert@qualcomm.com

Gyors hálózati kapcsolat keresztkötésű kábellel

Keresztkötésű ethernetkábellel hogyan tudok adatokat másolni két gép között, ha mindkettőn Debian sarge fut, illetve ha az egyiken sarge, a másikon pedig Microsoft Windows található?
Zelei Ákos, azelei@freemail.hu

Egyszerűen adj mindkét gépnek tetszőleges, ugyanabba a hálózatba eső IP-címet. (Javaslom, hogy az RFC 1918 dokumentum szerint e célokra fenntartott címtartományokat használd, ilyen például a 192.168.x.*. Ha ezt választod, akkor például az egyik gép címe lehet 192.168.1.1, a másiké pedig 192.168.1.2). Ha jól választod meg a címeket (vagy követed a példámat), az alhálózati maszk és a szórási cím értékét az alapértelmezetten is hagyhatod. Ezután már meg kell tudnod pingelni egymásról a gépeket. Ha ez sikerült, akkor az összeköttetés felett bármilyen TCP/IP protokollt használhatsz. Az elérésekhez használhatod az IP-címeket, de a ***/etc/hosts*** fájlba is készíthetsz új bejegyzéseket. A kapcsolaton keresztül `rsync`-et, `scp`-t vagy bármilyen, neked tetsző protokollt futtathatsz.
A windowsos rendszeren ugyanez a helyzet: adj meg egy állandó IP-címet, majd használd a rendszer natív fájlmegosztását (ekkor Sambára lesz szükséged a Debian GNU/Linux rendszeren), vagy telepítsd fel a Cygwin for MS Windows csomagot, és futtass például `rsync`-et SSH felett.
Jim Dennis, jimd@starshine.org

Ha nem akarsz Samba kiszolgálót telepíteni a linuxos gépre, akkor tedd fel a `putty` nevű programot a windowsosra (➲ http:\\www.chiark.greenend.org.uk/~sgtatham/putty). Ha a windowsos gépen már be van állítva a fájlmegosztás, akkor futtasd az `smbclient` programot a linuxosról.
Don Marti, dmarti@ssc.com

Linux Journal 2004. május, 121. szám

Linux a gyakorlatban

A „csináld magad informatika”: hogyan teremti meg a Linux és a nyílt forráskód a „csináld magad” elvét az információs technológiában?

Aki a hagyományos IT médiát követi, az lemarad a kisebb és nagyobb vállalatok körében egyaránt érezhető új irányvonalakról szóló hírekről: előtérbe kerül a belső erőforrásokra való hagyatkozás és fokozódik a Linux fejlesztésében való részvétel. Az ***Ernie Ball***, a ***Morgan Stanley*** és a ***Ticketmaster*** vezetői beszélnek a „csináld magad” elv kialakulásáról.

Kétségtelen, hogy a Linux és a nyílt forráskód forradalmasítja a kisebb és nagyobb vállalatok informatikával kapcsolatos elveit. De hogyan? Hiszen ha a szakterület lapjait olvassuk, összejövetelekre járunk és elemzőket hallgatunk, a szlogen amit hallunk vagy olvasunk mindig és mindenhol ugyanaz: a hatalom a vállalatok kezében van.

A ***Computerworld.com*** tavaly novemberének egy hetét elemző rovatában 62 cikk közül 52 (82%) vagy valamelyik cégről szól (például: *„A Red Hat és a Fedora”*), vagy pedig a címében tartalmaz utalást egy vállalatra (*„Az IBM szerint az Egyesült Államoknak fejlesztésekre van szüksége a munkahelyek megtartása érdekében”*). A valóság azonban nem ilyen egyszerű. Az úgynevezett „cégek” és az úgynevezett „vásárlók” különféle fejlesztői közösségeket képeznek. A vállalatok és a vásárlók eszközöket fejlesztenek saját maguk vagy mások számára, illetve eladásra.

Az elmúlt év folyamán a ***Linux Journal*** megbízásából azt tanulmányoztam, hogy mi is történik valójában az IT piacon, és azt is vizsgáltam, hogy milyen szerepet játszik ezekben a változásokbana Linux és a nyílt forráskód. Első jelentésem a 2003. júliusi Linux Journalban jelent meg (➲ http://www.linuxjournal.com/article/6585), címe „Hogyan teszi a Linux bölcsebbé a vállalatokat?” Jelen írás a témában készült második jelentés, és az informatikában végbemenő változásokkal foglalkozik. Hosszú kutatásom során azt tapasztaltam ugyanis, hogy egyre inkább növekszik a személyes és közösségi fejlesztés népszerűsége és megbízhatósága, továbbá hogy egyre kézzelfoghatóbbá válik az ezt lehetővé tevő szabadság és bizalom. Nyugodtan kijelenthetjük tehát, hogy a csináld magad mozgalom mindenhol előretört, népszerűsége rohamosan növekszik, s ezt az új megközelítést nevezem én „csináld magad IT-nek” (CSMIT).

Phil Moore, a Morgan and Stanley vállalati alkalmazásinfrastruktúrájának igazgatója erről a következőképpen vélekedik:

A nyílt forráskód nem tett mást, mint lejjebb helyezte azt a képzeletbeli lécet, amely fölött az „csináld magad IT” lehetséges. A gyártók által szállított, előregyártott építőelemekből nem valósítható meg minden. Azt híresztelik magukról, hogy kész házat kapunk, de ez valójában nem igaz. Csöveket, foglalatokat és más, az összeszereléshez szükséges alkotóelemeket kapunk. A valóságban ez úgy néz ki, hogy ha vállalati rendszert építünk, akkor az IT műhely szakértőinek tudására is szükség van ahhoz, hogy az egész összeálljon. Sok tapasztalat kell ahhoz, hogy egy ilyen rendszert összeállítsunk, hiszen nem egyszerűen megvalósítható dologról van szó. De ha a vállalatok elhitették velünk, hogy minden olyan pofonegyszerű, mint a .Net reklámokban vagy bármely, ígéretektől hemzsegő nagy reklámkampányban, akkor baj van.

Mindig szükség van egy bizonyos mennyiségű „csináld magad” késztetésre. A tanácsadók besétálnak, elvégzik a munkát és az ajtóból még visszaszólnak: „Hívjatok fél év múlva és jövök frissíteni” – ez a kép igen messze áll a valóságtól. Egy vállalat folyamatosan változik. Hogy a házunk falai most jó állapotban vannak, nem azt jelenti, hogy hamarosan ne lenne szükség újrafesteni őket. Bár a CSMIT azt is jelenti, hogy csökkentjük a vállalatoktól való függésünket, azért senki nem mondja azt, hogy a cégek rosszak vagy nem játszanak fontos szerepet a piacon. Mindössze arról van szó, hogy a piac már nem hozzájuk tartozik.

Az a helyzet, hogy a kínálat és kereslet között új egyensúly jött létre, akár a gyártó, a vásárló és a fejlesztői közösségek között is. Úgy tűnik, a szoftverüzlet egyre nő.

Ebben az írásban a csináld magad mozgalom főbb tényezőit tekintjük át: mit jelent a vezető szerep, mi a világháló szerepe, mi a bátorság jutalma. Beszélünk a költséghatékonyság mindenek felett álló elvéről, a tehetség értékeléséről, a jelenlegi állapotról, a titkolózásról és új nézőpontokról.

Melyik vezért kövessük?

Természetes, hogy az ember vezetők után kutat, akikre figyelhet, akiket követhet. De mi a helyzet azokkal a fejlesztésekkel, melyeknek nincs igazi vezetőjük, ahol a jövőt érintő döntések a közösségen belül, kölcsönös meggyőződésből és az ezt kifejező tettekből alakulnak ki?

Pontosan ez zajlik a Linux, az ingyenes programok és

1. táblázat *A Ticketmaster előtt álló lehetőségek*

Terméknév	Ár nyílt forrású beszerzés esetén	Ár fizetős programok esetén
400 db PC	$1.000.000	$1.000.000
Operációs rendszer	$0	$600.000
Webkiszolgáló	$0	$120.000
Adatbázisrendszer	$0	$240.000
Összesen	$1.000.000	$1.960.000

a nyílt forráskód és a szabad szoftver mozgalom körül.
A Linux egy fejlesztés, nem pedig vállalat. Nem része semmilyen vállalati struktúrának. Akár egy kőzetlemezt, a Linuxot is a kohéziós erő tartja össze, nem pedig egy szervezet rácsai. Az ingyenes programok és a nyílt forráskód értékek és fejlesztési módszerek rendszerei. Ha csupán társulásokként vagy egyfajta árufajtaként tekintünk rájuk, akkor pont amellett megyünk el, amiről ez az egész szól.
A világháló nemcsak azokat a dolgokat biztosítja, melyeket ma már alapvető szolgáltatásnak tekintünk, hanem minden és mindenki számára lehetővé teszi, hogy bekapcsolódhasson, tanuljon és bölcsebbé váljon. Amiről sokat beszélnek, az elnyomja a mélyben rejlő elemeket, akkor is, ha ez utóbbiaknak még van üzleti vagy más értéke.
Craig McLane, a Ticketmaster technikai igazgató-helyettese így fogalmaz:

Számomra az Internetben az a legnagyszerűbb, hogy mérsékli az időeltérésből és távolságból eredő súrlódásokat. A céltudatos és tehetséges emberek egyszerűen egymásra találhatnak, kapcsolattartásuk akadálymentessé válik, és sokkal könnyebben, hatékonyabban tudnak együttműködni, mint azelőtt bármikor.
A nyílt forrás közössége valóban világméretű, és ez rengeteget számít nekünk. Nemzetközi vállalat vagyunk és terjeszkedésünk során szeretnénk eljutni azokhoz az okos emberekhez, akik még elérhetők és viszonylag szoros kapcsolatban állnak egymással.
Elképesztő erejű az a felismerés, hogy e közösség életében bárki részt vehet, nagyszerű tetteket vihet véghez és gondozhatja az innováció eme bölcsőjét.

A hálózatba kötött informatikai világ

Ma már egyetlen üzleti vagy állami szervezet sem létezhet az Internet, ezáltal a nyílt forráskódú programok nélkül. Egyre több helyütt felismerik azt is, hogy a mozgalom értékeire is szükségük van, nemcsak az abban résztvevők tehetségére és programjaikra. Csupán az elmúlt évben kezdtek ráébredni a kormányok világszerte, hogy az állami informatikát nyílt, titkoktól mentes rendszerekre kell építeni, hiszen átláthatóságot, megbízhatóságot és egyszerű fejleszthetőséget keresnek.
Nézzük például ***Sterling Ballt***, az ***Ernie Ball*** világhírű gitárhúrkészítő cég vezetőjét és névadóját. Termékeik hangját bármikor hallhatjuk, ha bekapcsoljuk a rádiót. Ami ***Sterling Ballt*** a nyílt forráskód forradalmárává teszi, az nem az informatikai tudása, hanem a függetlensége – kimondhatjuk: a merészsége. Ez akkor vált egyértelművé, amikor a BSA (a szoftverrendőrség) 2000-ben kalózprogramokat talált az ***Ernie Ball*** néhány számítógépén. A BSA a honlapján még mindig találunk a razziával foglalkozó dicsekvő hangú írást. ***Sterling Ball*** a 2003. augusztusi ***LinuxWorld*** kiállításon folytatott beszélgetésünkben a következőképpen ismertette a helyzetet:

„Egy elbocsátott alkalmazott szeretett volna bosszút állni valamiért, és úgy gondolta, hogy erre az a legmegfelelőbb módszer, ha kihívja a BSA-t. Meg is tette, engem pedig szövetségi perbe fogtak. Semmiféle figyelmeztetést, értesítést nem kaptam erről, az egyik péntek délelőtt egyszer csak betoppantak, mindenkit a gépek leállítására szólítottak fel és a gépekhez nem nyúlhattunk. Fegyveres rendőrbírók is jelen voltak. Alkalmazottaink csak rémülten ültek és ilyesmiket suttogtak, hogy „Akkor most mi lesz? A cég bűnt követett el? És ha igen, akkor mi bűnrészesek vagyunk?" Aztán kiadták a sajtójelentéseket... Érdekes egybeesés, hogy ezeket a közleményeket mindig csak olyankor adják ki, amikor bezárnak egy vállalatot, vagy valami hasonló történik.
San Luis Obispóban mi vagyunk a legnagyobb vállalat. Nagy hal vagyunk egy kis pocsolyában, ha úgy jobban tetszik. A közlemény főcíme így szólt: „Az Ernie Ballnál házkutatást tartottakaz illetékes szervek az ott használatban levő kalózmásolatok miatt", majd így folytatja: „A vállalat vezetői nem hajlandóak nyilatkozni." Nos, ez természetesen nem igaz. Otthon ültem. És olyat sohasem mondok, hogy nem nyilatkozom. Ez feltehetőleg nekik is a tudomásukra jutott, mert végül felhívtak, és megkérdeztek az ügyről. Amikor ez megtörtént, én csak annyit mondtam: Medvére vadásztak, de csak egy mókust találtak."

Az ***Ernie Ball*** bíróságon végezte és büntetést fizetett, de a történetnek ezzel még nem lett vége.

„A legrosszabb az volt, amikor a Microsoft másolatot készített a szóban forgó újságcikkből és minden nyilvántartott Microsoft felhasználónak elküldte. A cikk aljára pedig az alábbi kis kiegészítést biggyesztették: „Ne járjon úgy, mint az ***Ernie Ball***, amely egy nagyszerű vállalat, és most azt is megtanulta, hogy milyen nehéz megfelelni. Hívjon minket bizalommal. Ingyenes felmérést végzünk számítógépes hálózatán található szoftverekről, és 20% árengedményt biztosítunk termékeinkből."
Azért azt nem szabad elfelejteni, hogy mi még az incidenms előtt letöltöttük a BSA auditáló programját és nem működött. Ez az, amit félelemre alapozott marketingnek neveznek, amibe itt-ott besegít a mindenkori kormányzat is."

Sterling Ball nem lett dühös, és nem a bosszúálláson törte a fejét, hanem inkább kiszállt:

„Mindenki azt hitte, hogy megőrültem. Az egész IT részleg például meg volt győződve arról, hogy mindenkit kirúgok. Azt mondtam, nem, mert én az informatikához csak olyan jelmondatok szintjén értek, hogy „Nem lesz sikeres a válla-

lat, amíg nincs irodai csomag minden gépen." Hé, azért beszélgethetünk itt a LinuxWorldön, mert lecseréltem a szövegszerkesztőmet. A vállalatunknál most már mindenki nyílt forrású programok vegyesfelvágottját használja. Senki nem mutatta meg, hogyan kell csinálni. Magunktól kellet rájönnünk."
Ernie Ball kiszolgálóin ma már Red Hat Linux fut, a munkaállomások GNOME-ot használó egyszerű Sun grafikus terminálok, melyek az alkalmazásokat egy Linux kiszolgálón futtatják. A vállalat levelezőrendszere a Ximian Evolution, irodai csomagja pedig az OpenOffice.org.
A tanulság itt nem technológiáról szól. Hanem a függetlenségről, a feddhetetlenségről és arról, hogy képesek vagyunk-e szabadulni a reklámok hatása alól. A lecke itt az egyén, a szervezet és egy olyan piac lelkéről szól, melynek résztvevői még mindig azt hiszik, hogy az irányítás a nagyvállalatok kezében van, hiába bizonyítja ennek ellenkezőjét az Internet és a nyílt forráskód sikere.

A költséghatékonyság késztető ereje

A szabadulás a felébredéshez hasonlít, és ez nem csak az ***Ernie Ball*** méretű vállalatokkal történik meg. Vegyük például a ***Ticketmastert***. ***Craig McLane*** a LinuxWorldön így nyilatkozott:

„Tíz országban 8000 ügyfelet szolgálunk ki. 2002-ben 95 millió jegyet adtunk el olyan csatornákon keresztül, melyek együttesen több mint négymilliárd USA dollár forgalmat tesznek ki. Ezzel bekerültünk a 25 legnagyobb internetes kereskedőcég közé. Valójában a második helyen állunk, a Dell és az Amazon között. Tehát üzletelünk, de ezt olyan emberek nevében is tesszük, akik megbíztak bennünket saját üzletük irányításával. 3500 kirendeltségünk, 19 telefonos ügyfélszolgálati központunk és ott van a Ticketmaster.com weboldal, mely önmagában a forgalom 50%-át jelenti. Jegypénztár-rendszereket is szállítunk. Ha vásároltál már jegyet pénztárban, az szintén Ticketmaster rendszer, a sajátunkkal megegyező elemekből felépítve. A „Termék és Technológia" nevű osztályunk képezi a Ticketmaster, mint vállalat sarokkövét. 250 ember foglalkozik kizárólag ezzel a 2000 főt alkalmazó vállalatban.
Megoldásokat és rendszereket kínálunk, de azt a 8000 ügyfelet is ki kell szolgálnunk. Sok esetben az üzletünk jellege (a szokványostól eltérő, nagy kilengéseket mutató forgalom, illetve más kereskedelmi hálózatoktól erősen különböző, egyéni rendszereink) miatt nem létezik olyan készen kapható megoldás, mely kielégítené igényeinket.
Valójában az elsők között vagyunk, akik alkalmazás-szolgáltatást nyújtanak: a szolgáltatást több ezer ügyfélre terjesztettük ki, megírtuk a kódot, helyet biztosítunk a rendszernek, ügyféltámogatást nyújtunk és eladott egységenként díjat szedünk be.
Nagy teljesítményű rendszerekre van szükségünk, melyek igen sajátos feladatokat hajtanak végre. A nyílt forráskódú programok ezt lehetővé teszik, ráadásul jobban működnek és feleannyiba kerülnek, mint a hagyományos üzleti rendszerek.
Az elmúlt másfél év során ezért alakítottuk át a➲ Ticketmaster.com oldalt úgy, hogy az kizárólag nyílt forrású eljárásokat használjon."

McLane ekkor elővette az *1. táblázatban* látható kis ábrát, mely a vállalat költségterheit mutatja nyílt forráskódú, illetve szabadalmazott rendszerek használata esetén. Így folytatja:

„Valójában minden a licenszköltségekről szól. Ugyanazt a gépet kell megvennünk, ugyanolyan kiépítésben, azonos szállítótól. De mivel nyílt forráskódú programokat használunk, nem kell licenszdíjakat fizetnünk. Láthatjuk, hogy minden befektetett egy dolláron 50 centet takarítunk meg és cserébe hasonló vagy jobb teljesítményt kapunk. Ráadásul a fejlesztői közösségek felől érkező támogatás általában többet ér, mint a programokat áruló cégek ügyfélszolgálata. A tőkefejlesztés és hardver tekintetében a weboldal néhány millió dollárba kerül, tehát az egész forgalom töredékéről beszélünk. A Wall Street vagy a CFO nézőpontjából nem változtatja meg a vállalatot, nekünk azonban számít, mivel a megtakarított pénzt további okos emberek foglalkoztatására tudjuk fordítani. És ez az, ami igazán fontos."

A tehetség értékelése

Craig MacLane számára a nyílt forrású programokkal foglalkozók értéke egyszerre nyilvánul meg a csoport, illetve az egyén erejében. Ezt egy T. S. Eliott idézettel magyarázza: „Nincs költő, nincs olyan művész, aki pusztán maga hordozza önmaga jelentését." És hozzáteszi:
Erőteljes eszközöket adunk embereink kezébe és egyre több, a megoldások keresése iránt érdeklődőt vonunk be, akik képesek önálló munkavégzésre, de fontos számukra a csapathoz tartozás is.
A ***Stas Beckmanhoz*** hasonló emberek igen fontos területtel foglalkoznak: a mod_perl 2.0-val. Az ő munkája a közösséget segíti, így minket is. Adatbázis-kapcsolatokkal is dolgozik... Tehát olyan szakterületeken is élen járunk, melyeknek eredményei széles körben hasznosításra kerülhetnek.
Geoffrey Young is nálunk kezd hamarosan. Olyan feladatokkal foglalkozik majd, melyeket önállóan is elvégezhet, és a közösség is hasznot hajthat belőle, emellett persze megkönnyíti a mi dolgunkat is.
Csapataink rendelkeznek a saját eszközeik felett. Kifogásokkal sem állhatnak elő, nem mondhatják, hogy a program gyártója nem tud választ adni a problémára vagy nem lehet telefonon elérni.
Mindenki tudja, hogy a közösség a közönség mindig ott van, és a forráshoz is hozzáférhetsz. Minden előttünk van. A nyílt forráskód, illetve a sokat tudó és segítőkész közösség tehát egyfajta felelősségbiztosítás is egyben.
Motivációban sincs hiány. Ha az emberek választhatják meg, hogy mivel szeretnének dolgozni, munkájuk azonnal értelmet kap számukra. Vállvetve dolgoznak az általunk nap mint nap használt eszközök fejlesztőivel. Ezt sehol máshol nem kapod meg."

Íme tehát egy újabb vállalat, mely okosabb lett, és pénzt takarított meg azáltal, hogy saját okos embereihez és az őket segítő közösségekhez alkalmazkodott.
Mi a különbség a mai és a 10-20 évvel ezelőtti helyzet között? ***McLane*** szerint „az Internet ma megtöbbszöröz az egyén képességeit." És hozzáteszi: „Ezt súlyos hiba volna figyelmen kívül hagyni."

Hol állunk most?

Nap mint nap olvashatunk arról, hogy az informatikai főnökök mindenhol odavannak a Linuxért. ***Ken Harris***, a ***Gap Corp***. divatcég informatikai vezetője a minap nyilatkozta, hogy bármi jöhet, „aminek csak egy kicsi köze van a Linuxhoz." ***Emea Harris*** a ***Lehman Brotherstől*** azt nyilatkozta, hogy „a Linux köré építkezést nagyon komolyan vesszük."

Amikor a csináld magad informatikáról beszéltem a 2003. júliusában tartott O'Reilly Nyílt Forrás Közgyűlésen, a beszélgetésen résztvevők (körülbelül 100 ember) többsége nagyvállalatokól érkezett régi vágású IT szakember volt. Köztük ült ***Phil Moore*** is. Az egyik hallgató azt mondta, hogy néhány vállalat úgy érzi, a nyílt forrású fejlesztések révén előnyre tesz szert a versenyben és éppen ezért nem akarnak ezekről nyilvánosan beszélni: „Nem akarják, hogy versenytársaik megtudják, hogyan lehetne gyorsabb és olcsóbb megoldásokat találni."

Elhallgatott történetek

Vajon mi minden nem került még nyilvánosságra? Felsorolok néhányat, a sorrend nem fontos:

- A Debianról nyilatkozta egy technológus, kinek cége a Red Hattal és a SuSE-vel tart fenn társult viszonyt: „Sokkal több helyen látni Debiant, mint amennyiről a különféle jelentések beszámolnak. A Red Hat és a SuSE lehet, hogy többet ad el, és így a felméréseket az első helyeken szerepelnek. A tényleges megvalósításokat tekintve azonban a Debian hatalmas népszerűségnek örvend."
- Oktatás: az elmúlt 30 év során az olyan vállalatok, mint a Digital, az Apple, az IBM vagy a Microsoft sikerre vitték azokat a tanfolyamokat, melyekkel a diákokat magukhoz láncolhatják. Most ezek a diákok a rengeteg szabad szoftver és a futtatásukra képes régi vagy olcsó gép közül választhatnak. ***Doug Kaye*** webszolgáltatás-tanácsadó és szakíró szerint „a középiskolákban és a főiskolákban ma már minden a szabad forráskódról és a LAMP-ról szól."
- A biztos mellékkereset ereje: számos informatikus arról számolt be nekem, hogy a nagyvállalatokkal ápolt jó kapcsolatok mindig is igen fontosak voltak, azok is maradnak mindörökre és kész. „Kapsz egy csomó ingyenes dolgot. Jegyeket sporteseményekre, vacsorameghívásokat, konferenciameghívókat. Sokan emiatt imádják ezt a vonalat."
- A kis szolgáltatók megnövekedett esélyei: az egyik nagyvállalati informatikus a következőket mondta nekem: „Az informatikában érzékelhető csináld magad elv lejjebb viszi a piacra való belépéshez szükséges küszöböt, így kis vállalatok is bejuthatnak a körbe. Harminc éve a nagyvállalatok a nagy megoldásokért más nagyvállalatokhoz fordultak. Ez mára megváltozott. Létrejöttek azok a 15, 20, 40 főt foglalkoztató kis cégek, melyek odamennek és tökéletes rendszereket valósítanak meg hatalmas vállalatok számára."

Phil Moore hozzáteszi:

„Amikor a partnert kereső vállalat egy nyílt forrású eszközöket szállító kis cégre adja a voksot, a nagyvállalatokkal történő egyezkedés minden lidércnyomása kikerülhető. Ha a nagy cégtől csupán kis változtatást kérünk, gyakran az is nagy összegekbe kerül. Az általam látott legkisebb értékű nagyvállalati szerződés is félmillió dollár környékén mozgott. Csupán a szerződéssel kapcsolatos tárgyalásokhoz két hatalmas jogi osztály szükséges. Én pedig körülnéztem és egy hasonló méretű változtatást nyílt forrású alapokra építve négy számjegyű összegből megoldottam. Ez egy aranybánya, mely óriási mértékben könnyít a költségvetési terheken és a folyamat jóval kevesebb időt vesz igénybe.

Kitekintés

Nagyvállalatok nélkül nem lennének magazinjaink, kiállításaink, csak hogy két kedvencemet említsem. A piac ökológiája szempontjából fontos, hogy a nagy cégek erőltessék saját termékeiket és meséljék saját történeteiket. A gond eddig az volt (sőt, most is az), hogy a piac eseményei és ezek tálalási módja között túl nagy a távolság.

Úgy vélem, hogy ez az érintett iparágak fiatal korára vezethető vissza. A számítástechnika 50 éves, a szoftveripar fele ennyi. A mindent megváltoztató Internet csupán 9 évvel ezelőtt kért helyet magának az üzleti világban.

Most a piac keresleti oldalának kell nyíltabban megfogalmaznia elvárásait, a megfelelő szinten lévő embereknek pedig nagyobb bátorságra van szükségük mindezek elmondásához. Olyan kiadványokra is szükség van, amelyek örömmel fogadják és közvetítik e véleményeket, ehhez pedig segítő szerzők, szerkesztők és elemzők kellenek.

Christopher Lyndon barátom (➲ blogs.law.harvard.edu/lyndon), a New York Times volt tudósítója és az NPR „Connections" című műsorának házigazdája szerint ami az iparágban történik – a csináld magad mozgalom –, tisztán emersoni ihletésű.

Ő hívta fel a figyelmem az író Önbizalom című esszéjének (eredeti címe: Self-Reliance, elolvasható a ➲ www.emersoncentral.com/ selfreliance.htm oldalon) egy részletére:

„Hinni saját gondolatodban, elhinni, hogy ami neked igaz, az mindenki számára igaz – ez maga a tehetség. Mondd ki, amit titkon hiszel, és általános érvényűvé válik, hiszen a legbelső a kellő időben a legkülsővé válik. [...] Előbb vagy utóbb minden ember ráeszmél, hogy az irigység tudatlanság, a színjáték öngyilkosság, hogy jóban, rosszban csak önmagára számíthat, hogy bár a világegyetem telis-tele van jóval, a táplálékot adó kukorica egyetlen szeme sem lehet az övé, ha nem hullatja verejtékét a földdarabon, melyet neki művelni adtak. A benne rejlő erő természete új, s rajta kívül senki más nem tudja, mire képes, és ő maga sem, amíg nem próbálta."

Mi megtesszük, amit nekünk kell. És mint eddig, úgy továbbra is az Olvasók segítségére is támaszkodunk.

Linux Journal 2004. február, 118. szám

Doc Searls (doc@ssc.com) a Linux Journal vezető szerkesztője. Havi rendszerességgel megjelenő cikksorozata a Linux for Suits, kéthetente olvasható hírlevelének címe SuitWatch.

Demokrácia 2.0

Két amerikai elnökválasztási kampány is kavarodást idézett elő amikor a Világhálót mint kétirányú médiumot használta. Az újítókat ezúttal ugyan maga alá gyűrte a Nagy Televízió, de a nyílt forráskódú program készen áll a következő menetre.

Két évvel ezelőtt nyáron ***Lawrence Lessing*** beszédet tartott, melyben arra buzdította a fejlesztőket, hogy vállaljanak politikai szerepet, és harcoljanak azokkal az kezdeményezésekkel, amelyek a szabad és nyitott hálózatot szeretnék korlátozni. Felszerelni rá egy olyan csővezetékrendszert, melynek minden csatlakozóját és minden elágazását az ipari termelők digitális jogainak irányítására és védelmére létrehozott mechanizmusok szabályozzák.
Lessing nem fogta vissza magát:
„Két éve magyarázok nektek! És még mindig nem tettünk semmit. Egy csomó energiát öltünk honlapok, blogok, és Slashdot-történetek létrehozásába. [De] egyáltalán nem foglalkoztunk azzal, hogy Washingtonnak ezt az elképzelését megváltoztassuk.
De ha most nem tesztek valamit, akkor ez az általatok létrehozott szabadság, melyet egész életetekben kódoltatok, el fog tűnni. Vagy azok veszik el, akik bennetek fenyegetést látnak és ezért a szabadalmaknak nevezett jogi szabályozást hívják segítségül, vagy pedig azok, akik kihasználják az újítások feletti ellenőrzésnek azt a példátlan lehetőségét, amit a szerzői jogi törvény ad a kezükbe. E törvényi változások olyan világot fognak eredményezni, amelyben a szabadságotoknak már nem lesz helye. És ha nem tudtok harcolni a szabadságotokért, nem is érdemlitek azt meg.
És ti nem tettetek semmit."
Larrynek igaza volt. A RIAA-val, az MPAA-val, a nagy kiadók és műsorszóró vállalatok lobbijaival és magával a Kongresszussal folytatott harcban a „jók" újra és újra vereséget szenvedtek.
Jegyezzük meg ezt az időpontot és szaladjunk előre 17 hónapot, 2003 decemberéig. Az elsődleges elnöki bemutatkozások megkezdődtek és az egyik demokrata jelölt nem az lehengerlő fellépésének köszönheti jó helyzetét, hanem az Internetnek, és néhány olyan eszköznek, amelyek a Hálón futnak.
Egy Christopher Lydonnal folytatott interjújában Larry erről a következőket mondta:
„Elérkeztünk oda, hogy az emberek felismerték, hogy a kultúra nem olyasvalami, amivel etetni kell őket. Mint a szovjet birodalom fennállásának vége felé, amikor az emberek rájöttek, hogy részt vehetnek a kultúra létrehozásában és elosztásában. A műszaki technikák segítségével nekünk is lehetőségünk nyílik erre. A probléma éppen az, hogy a törvény ettől megfoszt minket. A Creative Commons feladata tehát az, hogy a törvényt valamilyen módon eltávolítsa az útból, hogy az emberi alkotóerő számára felhasználható legyen a műszaki technika e rendkívüli lehetősége… Célunk az, hogy megkönnyítsük az emberek számára annak a felismerését, hogy ez a kultúra számukra is hozzáférhető.
A „mi" ebben az esetben nem csak a Creative Commons-t jelenti. Ez valami új és mégis ismerős dolog: a szabad kultúra mozgalma. „Ahogy Richard M. Stallman életet adott a szabad szoftver mozgalmának", mondta Larry, „azt hiszem kimondhatjuk, hogy mi vagyunk a szabad szoftver a kultúráért mozgalma." És Richardhoz hasonlóan Larry sem késlekedik a határok meghúzásával:
„Van egy fontos különbség a szabad szoftver mozgalma és a nyílt forráskódú szoftver mozgalma között, mégpedig az, hogy az előbbinek elsődleges célja a szabadság. Nem jobb programokat ígér, és nem egy jobb üzleti modellt. A szabadságot ígéri. És azt hiszem a szabad kultúra mozgalmának is ez a lényege. Arról szól, hogy megadja az embereknek a szabadságot a saját kultúrájuk létrehozására és művelésére.
Mint a szabad szoftvereknél, itt is lényeges az eszköz.
A szabad kultúra eszközének „egyik legfontosabb példája", Larry szerint a webnapló:
A szabad kultúra jelenti az átmenetet a műsorszórás kultúrája és az alkotó kultúra között – az előbbi az, ahol kevesek szólnak sokakhoz, az utóbbi pedig, amelyben sok ember szól sok emberhez. Látszólag mindig is ez volt az Internet lényege. De a blogok megjelenése az első alkalom, amikor a politikai eszmékkel összefüggésben történik mindez, erre átültetve és kiterjesztve, ahogy más emberek beszámolnak erről. A politikai kampányok szavajárását használva: jobb állampolgárokká váltak. Elkötelezett polgárokká. Az elmúlt 150 évben nem született olyan új találmány, amely több elkötelezett polgárt eredményezett volna.
Larry mindezt 2003 végén mondta, amikor ***Howard Dean*** felbukkant a demokraták első számú elnökjelöltjeként. A 2003-as naptári évtől kezdődően Dean Internetre épülő kampánya a semmiből indult, rekordmennyiségű pénzt gyűjtött össze, rekord számú embert mozgósítva és esélyessé téve a jelöltjét a vezető helyre mind a szavazatok, mind a pénz tekintetében. Amikor mindennek vége lett és ***John Kerry*** szerezte meg

a Demokrata párti jelöltséget, a vezető újságok előre kiszámíthatóan hasonlították Dean kampányát a dot-com lufihoz. ***Joe Trippi***, Dean kampányigazgatója fogalmazta meg, amit mindenki tudott, ha valóban követte a kampányt vagy részt vett benne: „Ez nem egy dot-com bukás volt. ***Howard Dean*** kampánya egy dot-com csoda volt. Vizsgáljuk csak meg ezt a jelenséget. Ez a fickó az elmúlt év január 31-én indult el hét emberrel, 157.000 dollárral a zsebében, és mindössze 432 ismert támogatóval… Egy csillag volt… Hogyan történhetett meg mindez? Igazi csoda, hogy Howard Dean ilyen kezdés után eljutott 45 millió dollárig, amely több, mint amennyit a demokraták történetében valaha összegyűjtöttek. Ez nem az ő érdeme, nem is az enyém. A tiétek."

Trippi szerint a politika többé nem olyasvalami, amit a média közvetít, nem egy lóverseny, ahova csak a profik léphetnek be. Ez annak a televíziós korszaknak a vége, mondja Trippi, ami a Kennedy-Nixon vitával kezdődött 1960-ban: Öt-tíz évig tartott, amíg világossá vált, hogy ez volt az a pillanat, amikor a televízió elkezdett mindent befolyásolni az amerikai politikában. Amit senki nem látott előre az az, hogy ezzel kezdetét veszi a pénzért való futás, a versengés egy olyan egyirányú kommunikációs csatorna birtoklásáért, amely lényegében kiiktatta az amerikai választókat a folyamatból. Többé már nem az amerikai átlagemberről volt szó, hanem arról, hogy „Hogyan találjak egy pénzes pasit aki ír nekem egy 2000 dolláros csekket, és hogyan vegyek a pénzen egy televíziót magamnak?"

Joe Trippi mindezt 2004 február 11-én a Digital Democracy (Digitális Demokrácia) tiltakozó előadássorozat vitaindító előadásában mondta, amely az O'Reilly Emerging Technology Conference-t nyitotta meg. Ezt az eseményt én javasoltam az O'Reilly embereinek még 2003 júliusában, a cég Open Source Convention nevű rendezvényének végén. Szintén az én ötletem volt meghívni Joe Trippit, hogy adja meg az alaphangot. Az ötlet akkor merült fel bennem, amikor az ➲ http://www.xpertweb.com alapítója, ***Britt Blaser*** és egy energikus Dean-szavazó utazásra vitt el Dean kampányának főhadiszállására a Vermont állambeli Burlingtonba. Tisztán elektronikus formában voltam jelen. Az arcom Britt laptopján volt, a hangomat a laptop hangszórói jelentették, a szemem szerepét pedig a laptop fedelére szerelt kamera töltötte be. A testem Kaliforniában maradt. Ebben a test nélküli formában ismertem meg Joe-t. Britt fel s alá sétálgatott a laptoppal, amelyet úgy tartott mint egy előétellel megrakott tálcát, az előcsarnokban belebotlottunk Joe-ba. Miután egy sziával üdvözöltem Joe-t, megkérdeztem tőle, hogy hajlandó lenne-e a februári rendezvény nyitóelőadását megtartani. Meglepetésemre igent mondott.

Hosszú és furcsa utazás

A történetbe több mint egy évvel korábban keveredtem bele, 2002 június 26-án, az ➲ http://www.eThePeople.org, vagyis „nyilvános fórum az új demokratikus párbeszédért" New York-i nyitórendezvényén. Én voltam az első számú előadó, akit azért hívtak meg, mert kitaláltam „a piac nem más, mint párbeszédek hálózata" szólást, amely a The Cluetrain Manifesto (magyarul: ***Cluetrain, A hagyományos üzletmenet végnapjai, 2001 Typotex – a ford.***) alaptézise volt, s amely az eThePeople céljának meghatározását is elősegítette. Íme a gondolatmenet:

A demokrácia egy beszélgetés. Olyan párbeszéd, amelyben részt vesznek politikusok, az érdekelt csoportok, pártok, újságírók, lobbisták, kérdezőbiztosok, levélírók, kampánytámogatók, tiltakozók és szavazók. Egy folyamatos alkudozás arról, hogy az általunk vallott prioritásokat és értékeket hogyan kellene a köz javára fordítanunk.

Az eThePeople a honlapját a Linuxra építette. Ugyanígy tett ***Scott Heiferman***, annak a MeetUp-nak az alapítója, amely mindössze 12 nappal korábban indult útjára. Scott ott volt a partin és beszélt nekem a MeetUp Linuxos alapjairól. A MeetUp odáig jutott, hogy a demokrácia legfontosabb üzleti kellékévé vált – a kocsma mellett. Csak ***Howard Dean*** miatt több mint 165.000 ember csatlakozott a MeetUp-hoz.

Britt olyan gyakran hívott a Dean-kampány vermonti főhadiszállásáról, hogy már úgy éreztem, minden munkatársát személyesen ismerem. Noha néhányan már korábbról ismertek engem, mivel a kampány tudatosan merített a Cluetrain Manifesto-ban a piaci hálózatokról elmondottakból és az elméletet alkalmazta a szavazókra.

A Cluetrain-ben az áll, hogy „a piacok egyre értelmesebbekké válnak, mégpedig gyorsabban, mint a legtöbb vállalat."A Dean kampány megfogalmazása: „A hálózatba kapcsolódó polgárok gyorsabban válnak értelmessé, mint a legtöbb kampány."

Több mint 600.000 ember csatlakozott a kampányhoz, és jelentős százalékuk kint volt az utcákon, keményen járta az országot a jelölt érdekében, kellékeket gyártott, találkozókat és gyűléseket tartott, s mindezt kevesebb felső irányítással, mint az ország történetének kampányaiban bármikor. Több önkéntestől hallottam, hogy a kampány leginkább egy elszabadult vonatra hasonlított.

Tettem egy utazást ezen a vonaton a múlt nyáron, amikor „A Háló megmentése" címmel az egyik írásom megjelent a Linux Journal honlapján. Két nappal később a BlogForAmerica oldalon, Dean Weblogján megjelent ***Matt Gross*** levele „A Háló és a politika megmentése ***Doc Searls*** szerint, avagy vegyük fel a kesztyűt". A levél célja az volt, hogy több pénzt gyűjtsön össze, mint a Dick Cheney alelnök által tervezett összeg a következő hétfőn tartandó 2000 dollár/tányéros vacsorán Dél-Karolinában.

A szöveg legfontosabb bekezdése:

Arra gondoltunk, hogy 28-án, hétfőn, felvesszük a kesztyűt. Úgy gondoljuk, hogy ti meg tudjátok mutatni Dick Cheney-nek, hogy van más módja is annak, hogy 250.000 dollárt összegyűjtsünk egy nap alatt – mégpedig olyan emberek segítségével, akik magukénak érzik ezt a demokráciát. És nem azok segítségével, akik maguknak akarják megvásárolni azt. Mi a véleményetek? Vegyük fel végre a kesztyűt? Minek nevezzük az eseményt?

A hozzászólások száma, mint mindig, több százra rúgott. Szombat reggelre 82.260,28 dollár gyűlt össze, és hétfő előtt egy órával már meghaladta a 250.000 dollárt. A végső összeg (bár mondták, hogy tovább emelkedett) 508.540,31 dollár volt, amit 9601 ember gyűjtött össze – ez átlagosan 52,97 dollárt jelent fejenként.

Ahogy Britt és mások is magyarázták később, a rohamban nem a kampány azon dolgozói pörgették fel leginkább az eseményeket, akik a blogra a részeredményeket feltették, hanem azok az emberek, akik a blog megjegyzés-rovatában

szerepeltek. Egy megjegyzésrovat indítása a kampány blogján felért azzal, mint amikor artézi kutakat fúrunk a támogatók ezreinek zsebébe.
A kampánylátogatásom közvetlenül a 2004-es iowai jelölőgyűlés előtti napokra esett. A főhadiszálláson minden maximális fordulatszámon pörgött és úgy tűnt, hogy az ügy meg sem áll az előválasztásokon elért nyeréssorozatok előtt. Egy nappal azután, hogy visszatértem New Yorkba (a LinuxWorld kiállításra), Dean elvesztette az iowai előválasztásokat, csak harmadik helyezést ért el. Azon az éjszakán hallatta a híres „kiáltását", amit az ABC News „néhány nap alatt 700-szor" játszott le. A többi már történelem.
Vagy mégsem. Érdekes módon a DeanForAmerica kommentárjai tovább folytatódtak. Voltak, akik a madridi bombatámadással kapcsolatban fejezték ki a szimpátiájukat, megjelentek javaslatok a Kerry-kabinet tagjai számára, gyakorlati ötletek a kampányban megmaradt energiák további hasznosítására, mialatt a kampányban részt nem vevők közül kevesen tűntek úgy, hogy tudomást szereznének erről. Néhány bejegyzés magától Howard Dean-től származott – ez ritkán fordult elő korábban, amikor a jelölt elnöki reményei még nem foszlottak szerte.

Alkalmazkodik a nemzet

Mikor ezt írom, éppen azon bánkódom, hogy nem jutottam el Arkansasba, hogy lássam amint ***Cameron Barett***, ***Tony Steidler-Dennison*** és más hackerek a ***Wesley Clark*** kampányban tevékenykednek. Cam az egyike az eredeti bloggereknek, Tony-t pedig a Linux Journaltól ismerem, ahol rendszeresen jelennek meg cikkei.
Cam és Tony az egész kampányt nyílt forráskódra építette, összeügyeskedtek egy csomó eszközt amelyek most visszatértek az ökoszisztémába a szerzőjükkel együtt. A Clark-kampány befejezése után Cam a Kerry kampányban folytatta a munkáját, Tony pedig visszatért a korábbi szemlélődő szerepéhez. Amikor megkérdeztem Tony-t, hogy mi történt a csapata által a Clark-kampányhoz kifejlesztett eszközökkel, így válaszolt: „Maguk az eszközök egy módszert biztosítottak a Clark-kampányhoz, különösen arra vonatkozóan, hogy minél rövidebb idő alatt kapcsolhassunk a megfelelő sebességre. Arra nem lett volna időnk, hogy a támogató eszközök összekapcsolt rendszerét ilyen gyorsan vagy ilyen költséghatékonyan létrehozzuk szabadalmazott eszközökkel.
Ezek voltak az első olyan kampányok, melyekben a világhálót valóban kétirányú kommunikációs csatornaként használták. Úgy gondolom, hogy Cam Clark Community Network (Clark közösségi hálózat) alkalmazása volt a kampányokban felhasznált leghatékonyabb eszköz a támogatói közösségek létrehozására. A 2000-es kampány az Internetet használta arra, hogy az üzeneteket a eljuttassa a tényleges és potenciális támogatókhoz. Ez még egyirányú kommunikáció volt.
Dean és Clark kampányai voltak az elsők a támogatói hangok bekapcsolása terén, és ezzel megteremtették a lehetőségét annak, hogy a jelöltek felé lehessen visszajelzéseket tenni. Más szavakkal: a kampánycsapatok ezzel kiemelték a kampányokat a pusztán adománygyűjtő szerepükből.
Elsőként említhetem meg, hogy a Clark-kampány tényleg odafigyelt a közösség együttes hangjára. A nyílt forrású eszközök pedig lehetővé tették, hogy létrehozzunk egy olyan közösséget, amely valóban hallatja a hangját a kampányban.
A Clark-kampány legfontosabb öröksége valószínűleg a Clark TechCorps. Azok az eszközök, amelyeket a támogató közösségek szervezésére hoztunk létre bárki számára elérhetőek a nyílt forráskódú felhasználói szerződés szerinti használatra és továbbfejlesztésre. Még mindig egészen nagy az aktivitás a TechCorp honlapján, mivel a téma iránt érdeklődő fejlesztők tovább dolgoznak az eszközökön. Majd meglátjuk, hogy meddig jutnak.
Sokan töltötték a 2003-as évet kódok buherálásával, mert úgy gondolták, hogy ez majd megváltoztatja a politikát -és igazuk lett. Ők a hatalmi eszközök területének nyílt forráskódú vállalkozói. Mint minden igazi vállalkozó, ők is olyan művészek, akik azért alkotnak, mert képtelenek nem alkotni.
Joe Trippi a MeatUp-ot a Dean-kampány gyilkos alkalmazásának nevezte. „Csináltunk egy kalapácsot", mondja ***Scott Heiferman*** „és ők építettek vele egy házat".
Zephir Teachout, akinek a Dean-kampányban való részvételét csak ***Joe Trippi*** szárnyalta felül, annyit mondott, a MeatUp-ról, hogy az azért ilyen jó eszköz, mert amit létrehozott, az nem csak a hálózaton létezik. A találkozások fizikailag is létrejönnek. A hálózaton kívül. ***Scott Heiferman*** elmondta nekem, hogy a MeatUp legfontosabb szervezeti kapcsolatai azokban a helyi éttermekben és kávézókban jöttek létre, ahol a tagok egymással találkoztak.
Vagyis a MeatUp sikere részben annak tudható be, hogy legalább egy web-fejlesztési alapelvet megszegett: nem hagyta túlburjánzani a szolgáltatásait és nem engedte, hogy a fejlesztési ambíciók elszakadjanak az elsődleges céltól. És ezeket a szolgáltatásokat mindvégig egyszerűnek tartotta meg, ahogy az már hagyományosan lenni szokott a nagyszerű és jól használható nyílt forrású programoknál.

Vizsgáljuk meg a forrásokat

Joe Trippi gazdag pályafutása során a Progeny-nél is dolgozott kiküldetésben, ahol munkatársa volt ***Ian Murdock***, a Debian társszerzője. Nem véletlen egybeesés, hogy ***Joe Dean*** kampányát „nyílt forráskódú kampány"-nak nevezte és azt mondta: „Az egész olyan, mint a Linux. Minél több ember csatlakozik, annál valószínűbb, hogy egy jobb dolgot hozunk létre."
Phil Windley a Utah-i CIO volt munkatársa szerint elérkeztünk annak a korszaknak a végére, amelyben a kormányzást megbénítja az ideológiák és ideológusok szemben álló tábora: a 60-as 70-es évek üzletellenes baloldaliak és a 80-as 90-es évek államellenes jobbosai. „Egyik csoport sem értette meg, hogy a polgárok nagy része nem vevő az ideológiákra. Valójában a legtöbb ember mindössze annyit vár az államtól, hogy az utakat megjavítsa." Annak a véleményének is hangot adott, hogy milyen izgalmas volt látni azt, hogy mekkora lehetőség az átlagpolgárt bevonásában és mi történik, ha hagyják részt venni a kormányzati gépezetben. Már számtalan NGO (Non-Governmental Organizations, nem hivatalos szervezetek) hozott a kormányzat számára megoldást valamilyen problémára ahelyett, hogy az ellenkezőjének a bekövetkeztére várt volna.
Módomban állt minderről személyesen is meggyőződni, amikor egy barátom megkért, hogy vegyek részt az egyik helyi csoport és a területi közigazgatás vezetőinek találkozóján. A helyi csoportnak volt egy elgondolása a területen belüli sávszélesség növelésére vonatkozóan, és ezen a talál-

kozón adták elő a konkrét javaslataikat. Meglepetésemre a választott hivatalnokokkal egy helyi kérdésről folytatott termékeny eszmecsere közepén találtam magam – ez számomra új élményt jelentett.
Amit a találkozó után vendéglátóm mondott, attól majdnem hanyatt estem, mert ilyesmit még nem hallottam azelőtt: „Nem a vezetéssel van baj. Az embereknek kell megoldásokkal előállni a helyi vezetőség felé. A közigazgatás epedve várja a válaszokat. Hozz néhányat és menni fog a dolog. Soha nem tapasztaltam, hogy a hivatalok packáztak volna velem. Minden szinten érdeklődést és együttműködési hajlamot találtam, amikor megoldásokkal álltam elő és nem csak problémákkal.
Sok tanácstalan ember a vezetéstől várja, hogy megoldja a gondjait. Ez a rendszer valódi működésének figyelmen kívül hagyásáról tanúskodik. Ha konstruktív vagy, részt vehetsz a rendszer működtetésében. A hivatalnokok minden kérdésben segítségért kiáltoznak. Ha segítünk nekik, akkor együtt dolgozhatunk velük."
„A demokrácia az első nyílt forráskódú alkalmazás" mondja ***Phil Windley***. Részletezve a kérdést arról elmélkedik, hogy az olyan nyílt forráskóddal kapcsolatos értékek, mint a kód nyilvánossá tétele, a szakértői bírálat (peer review), az önálló kezdeményezés, a javított kód állandóan ismétlődő beemelése az igazán soha be nem fejezett termékbe hosszú távon a demokráciának is lényeges elemei. Hisz abban, hogy a nyílt forráskód értékei és gyakorlata segít a demokráciának abban, hogy eljusson az ideálokig, amelyek évezredek óta beteljesületlenül maradtak.

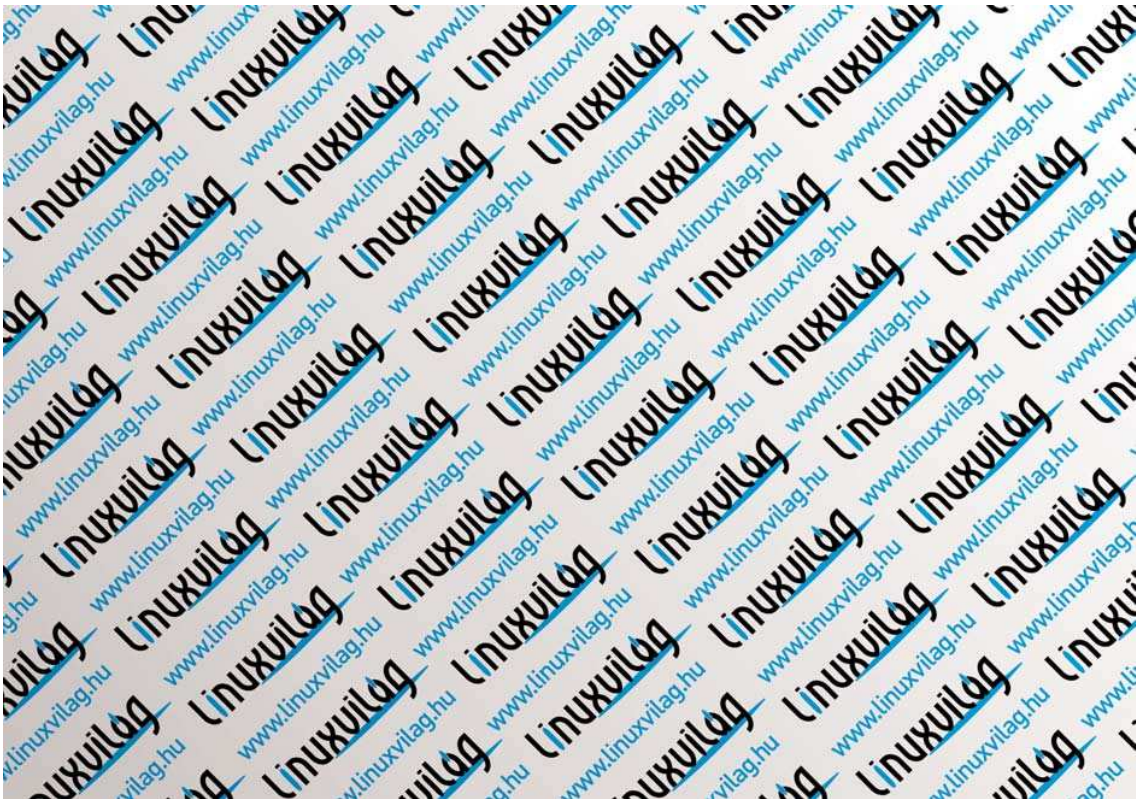
Más szóval: lehetséges, hogy eljuthatunk a demokráciának ahhoz az állapotához, amelyet mindig egy ideális állapotnak tartottunk – melyben a választott vezetők nem csak képviselik a polgárokat, hanem örömmel veszik bárkinek a segítő hozzájárulását.
Néhány érdekes tény a területi vezetőkkel folytatott értekezletről:

1. A tanácskozás minden résztvevője technikai szakember volt.
2. A megbeszélés minden közalkalmazottja úgy hitte, hogy az Internet sávszélességének növelése értelmetlen.
3. Az volt csak természetes a vezetésnek, hogy a szakemberektől várták a segítséget.

A közigazgatás helyi és regionális szintjén az Internet-párti szakemberek egyre inkább egy a szaktudásukra és megoldásaikra éhes keresleti piacon találják magukat. Legalábbis már látunk néhány megnyíló lehetőséget. Lehet, hogy hamarosan még többel találkozhatunk. Zephyr Teachout szerint több mint száz olyan lehetséges jelölt dolgozik most a közigazgatásban, akit megihletett a Dean-kampány. Marc Andreessen szerint „minden fontos fejlődési irányvonal a technikai szakemberekkel kezdődik". Ezeknek az irányoknak viszont nem kell technikaiaknak lenniük. Természetesen politikaiak is lehetnek.

Linux Journal 2004. június, 122. szám

Doc Searls



„Nem tudni, vajon legális marad-e a szabad szoftver”

Beszélgetés Richard M. Stallmannel.

Nemrégiben hazánkban járt a szabad szoftver szülőatyja (és prófétája), a Linux alapjának számító GNU készítője, Richard M. Stallman. A Linuxvilág ez alkalomból beszélgetett vele a jelenlegi állapotokról, a szoftverszabadalmakról és a jövőbeli kilátásokról.

LinuxVilág

– *A szabad szoftvereket alapvetően programozók készítik programozóknak. A világ azonban nem csak programozókból áll. Mit gondol, hogyan hódíthatja meg a szabad szoftver azokat a felhasználókat, akik nem tudják nélkülözni a kényelmet a funkcionalitás kedvéért? Nem érzi-e úgy, hogy ezen a területen – mármint a kényelem területén – a Windowsnak igenis előnye van?*

Richard M. Stallman

– De hát ez nem is igaz, és elég udvariatlan ilyesmit állítani. Természetesen a szabad programokat programozók készítik (hiszen minden programot programozók készítenek), és természetesen ők is fejlesztik azokat tovább. És valóban, vannak olyan darabok is, melyeket elsősorban programozóknak szánnak. De van – és hál Istennek egyre több van – olyan is, amelyet kezdő és tapasztalatlan felhasználóknak készítenek.

– *Például?*

– Hogy egy közeli példát mondjak: Olaszországban nemrégiben olyan embereknek tanították meg a GNU/Linux Gnome-jának, illetve a Microsoft Windows használatát akik soha életükben nem használtak számítógépet. Teljesen kezdők voltak minden szempontból. És ezek az emberek sokkal könnyebben tanultak bele a Gnome használatába, mint a másikba.
Állítólag kipróbálták a KDE-t is, ami kicsit nehezebbnek bizonyult a Gnome-nál, körülbelül olyan nehéznek, mint a Windows megtanulása. Tehát egyáltalán nem állítanám, hogy a szabad szoftvert a kezdők nehezebben, vagy egyáltalán nem tudják megtanulni kezelni.
Persze egy tapasztaltabb felhasználó sokkal többet tud kihozni a rendszeréből, mint egy tapasztalattal nem rendelkező, de hát ez így működik minden esetben. Mostanra azonban a GNU/Linux rendszer is elért egy olyan szintre, ahol egyáltalán nem számít, hogy valakinek nincsen számítógépes tapasztalata, mert ugyanolyan könnyen meg lehet tanulni, mint más rendszerek használatát.
Egyébként temérdek energiát pontosan ebbe fektettünk, hogy ti. a rendszer használatát könnyen el lehessen sajátítani, mert az elsődleges célunk mindig is az volt, hogy minden felhasználónak alternatívát és szabadságot nyújtsunk, ne csak a programozóknak.
Természetesen amikor néhány programozó barátommal összefogtunk, hogy készítsünk egy
Unixhoz hasonló operációs rendszert, akkor az eredmény még kevéssé barátságosra sikeredett. Parancsokat kellett begépelni, ami nagyon hatékony és gyors, de nem olyan könnyű használni. Aztán megcsináltuk a Gnome-ot, és a felhasználóbarát grafikus kezelőfelület hatalmas előrelépést jelentett.

Richard M. Stallman a szabad szoftver és a nyílt forráskód közötti különbségeket ecseteli az egyik televíziónak

– *Tehát most már teljesen versenyképes más rendszerekkel szemben?*

– Igen. És ez hosszú-hosszú évek megfeszített munkájának köszönhető, olyan emberek munkájának, akik kizárólag ezért dolgoztak, hogy a szabad szoftver mindenki számára hozzáférhető, és könnyen használható legyen. Mert mi azt szeretnénk, hogy a szabad szoftverek által garantált szabadság mindenki számára hozzáférhető legyen.

– És mégis mit lehet tenni annak érdekében, hogy egyre több felhasználó tisztában legyen a könnyű használhatósággal? Hogy mindenkihez eljusson?

– Nézzük csak. Hát az első és legfontosabb az lenne, hogy megoldjuk az olyan hardveres eszközök támogatását, melyek gyártói nem nagyon akarnak velünk együttműködni. Nagyon rá kell például hajtani a szabad Java platform megvalósítására. Jelenleg sajnos nagyon sokan élnek abban a tévhitben, hogy a Java egy platform független valami – ez sajnos egy óriási tévedés...

– De hát a Java minden operációs rendszeren fut, nem?

– A java gyakorlatilag minden operációs rendszeren fut, de sokan nem tudják, hogy ehhez az operációs rendszer egy olyan interpretert és könyvtárrendszert használ, amely a Sun tulajdona, és így szabadalommal védett és jogdíjköteles. Hogy a Java a szabad platformokon is futtatható legyen, mindezeket ki kell cserélni. Jelenleg is ezen dolgozunk, tulajdonképpen egy jelentős része már készen is van. Remélhetőleg a Mozilla következő verziójában már benne lesznek ezek a változások, és használni lehet a szabad Javát.

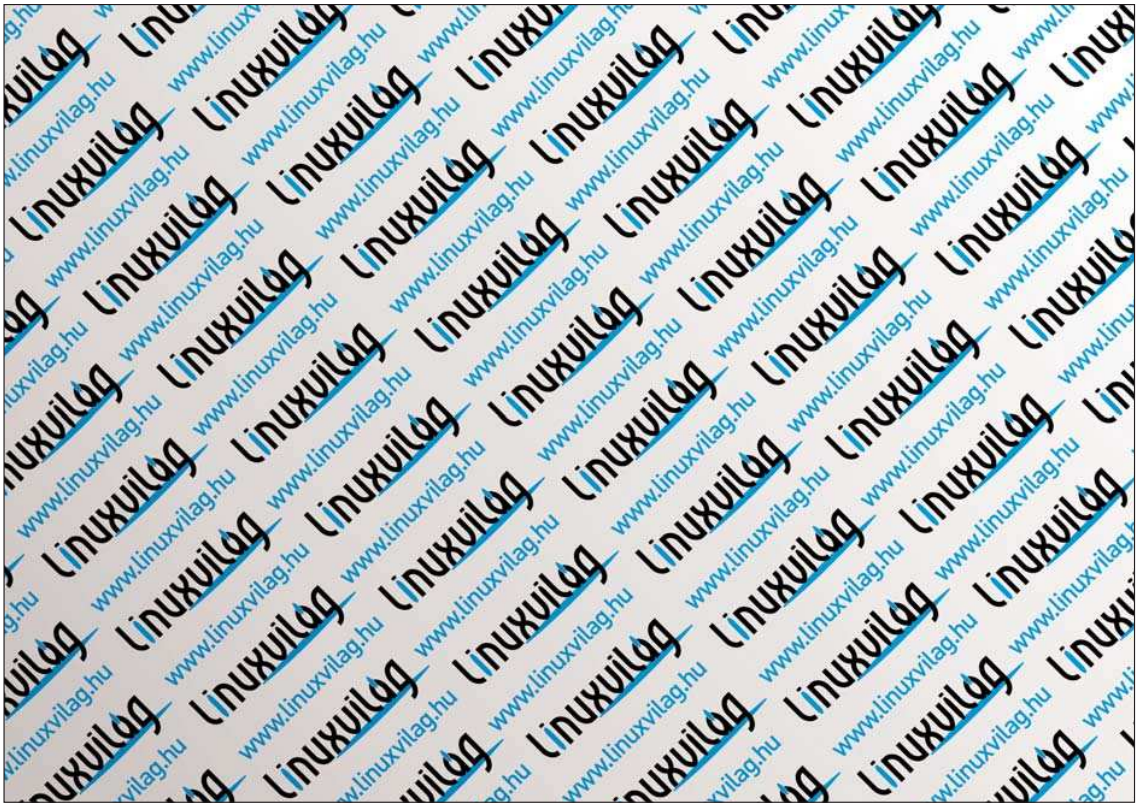
– Visszatérve az előző kérdéshez: a fogyasztói társadalomban marketing, promóció, és hirdetés nélkül nem, vagy csak nagyon nehezen lehet elérni, hogy az emberek tudomást szerezzenek a programról. Hiába van nekem a világon a legjobb termékem, ha senki sem tud róla, akkor nem fognak érdeklődni iránta. Tud-e versenyezni a szabad szoftver mozgalom a promóció tekintetében a nagyokkal? Abban, hogy megszólítsa az embereket?

– Nem hiszem, hogy ez számít. Nem értem, miért kellene ezzel foglalkoznunk. Itt a szabadság és a lekötözöttség, a lehatároltság közötti választásról van szó.

– Igen, de ha az emberek java része nem tudja, hogy választási lehetősége van, akkor nem fogja érdekelni ez a kérdéskör, legyen az mégoly magasztos és fennkölt is.

– Azért számtalan lehetőségünk van arra, hogy elérjük az embereket. Rengeteg cikk jelenik meg például, aztán ott vannak a különböző magazinok, szaklapok, mint az önöké is. Lehet, hogy az önök cikkei nem kifejezetten a szabadság kérdéséről szólnak, de alkalmanként óhatatlanul felmerül ez a kérdés, hiszen a GNU/Linux-szal kapcsolatban nem lehet elhallgatni ezt a dolgot. Ha közvetve is, de előjön. Nem számít, hogy mennyit hirdetnek a különböző cégek, mert ezek a hirdetések nem a szabad szoftver szabadságáról szólnak, és éppen ezért nem is annyira fontosak, hiszen nem terjesztik az eszmét.

– Nem érez-e lassulást a szabad szoftverek fejlődési ütemében, illetve nem tart-e attól, hogy egy idő után az emberek már csak használni akarják ezeket, fejleszteni nem, és a fejlesztések megállnak?



– Érdekes, de nem. Ha ugyanis egy program jó és sikeres, akkor annak eleve több felhasználója van, mint fejlesztője. Semmi okunk azt hinni, hogy a programozók száma lecsökken, hiszen egyre több és több programozó hagyja el az egyetemeket. És minél több felhasználója van egy szabad szoftvernek, annál több lesz a fejlesztő is. A programozók száma nem csökken, viszont a munkanélküli programozók száma egyre nő. Sokkal többen vannak, akik tudnak programozni, mint ahány állás jelenleg van ebben a szektorban. Szóval végül is van rengeteg olyan ember, akik a programozás iránti szeretetüket és elkötelezettségüket remélhetőleg így fogják majd kiélni.

– Nem lehet, hogy a szabad szoftvert jellemző demokrácia egyben kerékkötője is a hatékony fejlődésnek? Ami az előnye, egyben a hátránya is? Szinte megszámlálhatatlan Linux terjesztés van például, holott ha csak néhány, viszont sokak által támogatott összeállítás létezne, akkor hamarabb jelennének meg egyes újítások. Nem lenne hatékonyabb a kevesebb fejlesztés, és így a kisebb elaprózódás?

– A szabad szoftver és a szabadság pont arról szól, hogy az embereknek nem kell ugyanazt használniuk. Szerintem ez egy óriási félreértés, hogy a demokrácia és a szabadság csak akkor hatékony, ha mindenki egyesült erővel csinál valamit, egyként menetel a cél felé egy totálisan koordinált tömegben. Ez nagyon távol van a realitástól. Ráadásul nem szabad elfelejteni, hogy ezek a kollektív dolgok, hogy megpróbálunk mindenkit irányítani, és arra kényszeríteni, hogy egy irányba meneteljenek, ez gyakorlatilag önkényuralom. Ez nemcsak hogy gonosz dolog, de nem is alkalmas arra, hogy új dolgokat teremtsünk. A nem szabad szoftver kicsit a volt Szovjetúnióra hasonlít. A felhasználók nem dönthetnek semmiről, az egyetlen, amit tehetnek, hogy elfogadják a fejük felett, róluk hozott döntéseket – és a döntéshozók esetünkben a program fejlesztői.

– Hogyan értékeli, hogy egyre több országban térnek át a kormányzati szférában a szabad szoftverre?

– Örülök, és ennek számos oka van. Azt vallom, hogy minden felhasználónak szabadságában kéne álljon, hogy másokkal együttműködhessen. Legyen szó akár egyénekről, vállalatokról, vagy szervezetekről, vagy kormányzati szervekről. Mindenkinek jár a szabadság, amit a szabad szoftver ad meg. Egy kormányzati szervnek ráadásul hatalmas felelőssége van abban, hogy teljes mértékben uralja a számítógépeit, és az irányítás ne kerülhessen magánkezekbe. Tehát ha egy kormányzati szerv nem szabad szoftvert használ, akkor gyakorlatilag nem teljesíti a kötelességét, ami a felelősséget illeti.
De nem is a kormányzatok ebben a kérdésben a legfontosabbak, hanem az iskolák. Nekik kéne elsősorban szabad szoftvert használniuk, kivétel nélkül. Mert egy iskola azt is tanítja, hogyan bánjunk a másik emberrel. És ezzel az együttműködésre való hajlamot és a szabadságra való igényt is segíti kifejlődni a gyerekben. De van itt még más is. Amikor a gyerekek némelyike elkezd érdeklődni a programozás iránt (és néhányan mindig elkezdenek érdeklődni, efelől biztosak lehetünk), akkor előbb utóbb meg akarja tanulni, hogy azok a programok, amiket használnak, hogyan is működnek valójában. És amikor a diák odamegy a tanárhoz, és egy hagyományos szoftverről megkérdezi, hogy ez hogyan is működik, akkor a tanár csak annyit tud mondani: Ezt én nem tudhatom, és te sem tudhatod. Ez titok, és errefelé senki nem tudhatja. De ha szabad szoftverről van szó, akkor a tanár mondhatja például ezt is: nézd, én értem ezt egy bizonyos fokig, és el is tudom mondani. De ha többet akarsz róla tudni, akkor itt a forráskód, nézd meg, és egészen pontosan tudni fogod, hogyan működik a program.

– Hogyan látja a szabad szoftver jövőjét a jelenlegi tendenciák fényében? Sikeres lehet-e valaha ez az üzleti modell?

– Hát az első és legfontosabb, ami ezt befolyásolja, az az, hogy vajon legális marad-e a szabad szoftver. Az előadáson az imént szó esett a szoftverszabadalmak bevezetéséről, arról, hogy ez milyen mértékben gátolja meg a szabad szoftver fejlődését és térhódítását. Ezért jelent nekünk olyan sokat a szabadalmak engedélyezésének megakadályozása. Ha ugyanis a legtöbb ötletet szabadalom védi, melyek különféle monopóliumok tulajdonában vannak (***A Microsoft például éppen a napokban védte le a dupla kattintás ötletét – a szerk.***), akkor nem nagyon fogunk tudni hatékony programokat készíteni, vagy a meglévőket fejleszteni. Éppen azért harcolunk most az EU-ban, hogy megakadályozzuk ezt a katasztrófát.
Szerencsére nem vagyunk egyedül. Még a nem szabad szoftverek területén dolgozó fejlesztők közül is sokan vannak, akik felismerték ennek a szabadalmazási szisztémának a kártékony voltát. De például sokat tehetnek az önök olvasói is ennek érdekében, és ahogyan most a dolgok állnak, minden segítség elkel. Ha valaki csatlakozni kíván ehhez a mozgalomhoz, akkor látogasson el a ➲ http://www.ffii.org oldalra, ahol elég sok mindent megtudhat.

– Milyennek látszanak az esélyek?

– Nem tudom. Ha elég sokan csatlakoznak hozzánk, akkor nyerünk, ha nem csatlakoznak, akkor veszíteni fogunk.

– Úgy hallottuk, hogy volt némi nézeteltérés ön és a Magyar Szabad Szoftver Alapítvány között…ők ugyanis nagy erőket fektetnek a GNU/Linux magyarra fordításába, ön szerint pedig nem a fordítással, hanem a fejlesztéssel kellene foglalkozni…

– Már nem emlékszem pontosan, de azt tudom, hogy mit mondtam volna akkor is és most is egy ilyen kérdésre. Úgy gondolom, hogy egyáltalán semmi baj nincsen a rendszer nyelvének lefordításával, de az erőket a fejlesztésre kell koncentrálni. Minden tőlünk telhetőt meg kell tenni azért, hogy a legtöbb tehetséges fejlesztőt összegyűjtsük, és azok szabad programokat csináljanak. Mert emellett mindig lesznek olyanok, akik nem értenek a programozáshoz, és akkor ők foglalkozhatnának a fordítással. Míg hogyha úgy állunk neki a dolognak, hogy nosza, fordítsunk, akkor mindenki csak fordítani fog, és senki, vagy csak kevés ember foglalkozik majd a fejlesztéssel.

–susánszky–

Az iptables rejtett értékei

Néhány érdekes iptables bővítmény segítségével nemcsak karakterláncokat vagy kaputartományokat tudunk egyeztetni iptables szabályokban, de csapdát is állíthatunk a hálózatot rossz szándékkal használóknak.

Az iptables használatával akár néhány perc alatt is hatékony tűzfalat építhetünk, ráadásul a költségek a kereskedelmi termékek árának töredékét teszik ki. Az alapszintű iptables tűzfalak csomagszűrést végeznek, vagyis a hálózati forgalmat csomagonként vizsgálják, illetve a csomagok kezelésével kapcsolatos döntéseiket különkülön hozzák meg minden csomagra. Egyszerű beállításokkal megadott csomagok eldobását és továbbengedését érhetjük el. Annak meghatározása, hogy az adott csomagra éppen melyik házirendet kell alkalmazni, általában a cél IP-címének és kapuszámának, illetve az utazás irányának alapján történik. Az iptables állapotadatokat is képes használni, tehát képes megalapozottabb döntéseket hozni annak a kapcsolatnak az állapotát figyelembe véve, amelyhez a csomag tartozik. Egyszerű, mégis hatékony az a tűzfal, amely például tiltja a bejövő TCP/IP csatlakozási csomagokat és az Internet felől kezdeményezett, UDP alapú adatcseréket, miközben engedélyezi a kimenőket. Ez a megoldás szabad hozzáférést biztosít a felhasználóknak a külvilághoz, ám a nemkívánatos bejövő próbálkozásoktól megvédi őt. Természetesen egy ilyen rendszer önmagában még túl egyszerű lenne, igazi használhatóságához további szűrésekre is szükség van, de nagyvonalakban erről van szó. Az iptables természetesen ennél az egyszerű csomagszűrésnél sokkal többre képes. Ezek azok a rejtett kincsek, amelyekkel meg szeretném ismertetni a kedves olvasókat. Mivel a program szolgáltatásainak és beállításainak ismertetéséhez egy vastagabb könyv terjedelme is kevés lenne, most csak arra szorítkozom, hogy rövid tájékoztatást adjak ezekről.

Ismerkedés a POM-mal

A Netfilter két összetevőcsoportból áll, az egyik a rendszermag-térben a másik a felhasználói térben fut. A felhasználói térben futó programcsoportba az iptables és a hozzá kapcsolódó segédprogramok, könyvtárak, kézikönyvek és parancsfájlok tartoznak. A rendszermag térben futnak a rendszermagfoltok és számos további kiegészítő modul. Egy avatatlan felhasználó számára rémisztő lehet a foltok telepítése egy olyan nagy és összetett rendszerre, mint amilyen a Linux rendszermag, az út ráadásul buktatókkal és zsákutcákkal teli.

Egy hibás, vagy az összetevők együttműködését akadályozó folt hatására könnyen előfordulhat, hogy a rendszermagot nem lehet lefordítani, vagy – ami még rosszabb – nem lehet vele elindítani a rendszert A Netfilter csapata éppen ezen nehézségektől próbál megkímélni bennünket, ezért készítették el önműködő útitársunkat ***Patch-o-matic***, röviden POM névvel. A POM foltok gyűjteménye, melyhez tartozik egy, azok telepítésére való, kezdők számára is könnyen használható parancsfájl. Ezen foltok némelyike alapvető, ezekre minden iptables/Netfilter telepítésnél szükség van. Mások kiegészítő jellegűek, ezek pedig érdekes szolgáltatásokat nyújthatnak. Ezek tehát azok a rejtett kincsek, melyeket a POM leírása csak a „hol haszontalan, hol remekül használható” bővítményekként emleget.
A POM futtatása roppant egyszerű. Le kell tölteni a legújabb Patch-o-matic tar archívumot az ➲ ftp.netfilter.org /pub/patch-o-matic könyvtárból, majd – root jogosultságokkal – futtatni kell az alábbi parancsot. Ügyeljünk arra, hogy a `KERNEL_DIR`-ben megadott érték a rendszermag forrását tartalmazó könyvtár nevét pontosan tartalmazza:

```
KERNEL_DIR=/usr/src/linux-2.4 ./runme extra
```

Innen kezdve a telepítés interaktív és többé-kevésbé közérthető módon történik.

Karakterlánc-darabkák

A POM talán legelterjedtebb darabja a ***string*** modul. Segítségével a csomagok adatrészének tartalmát lehet összevetni karakterláncokkal. A modul számtalan célra használható, de beállításakor némi figyelemre is szükség van. Tegyük fel például, hogy meg kívánjuk akadályozni az ELF futtatható állományok letöltését. E célból beállíthatunk egy szűrőt, amely az Internet felé néző hálózati csatolón beérkező forgalom TCP/IP alapú, 80-as forráskapuról érkező részét vizsgálja. Ha tudjuk, hogy az ELF fájlok a 7f hexadecimális karakterpárral kezdődnek, amit az ELF karakterlánc követ, akkor karakterláncok egyeztetésével megtalálhatjuk az ilyen részleteket. Nem ASCII karaktereket a csővezeték szimbóluma közé zárva illeszthetünk be a karakterláncba, mint például |7F|ELF. Feltéve, hogy az internet felé az `eth0` csatoló néz, a szükséges parancs a következő:

```
iptables -A FORWARD -i eth0 -p tcp -sport 80 \
   -m string –string ´|7F|ELF´ -j DROP
```

A hexadecimális karakterek beágyazásának lehetősége az iptables 1.2.8-as változatával jelent meg. Ha korábbi változatot használunk, akkor valamilyen trükkhöz kell folyamodnunk. Például a

```
--string "`dd if=/bin/ls bs=4 count=1
➥2>/dev/null`"
```

paranccsal a */bin/ls* program első négy karakterét vehetjük, és mivel ez is egy ELF állomány, pontosan a szükséges karakterláncot fogjuk kapni. Példánkat tovább finomíthatjuk, ha azt mondjuk, hogy a 192.168.0.5 című kiszolgálóról származó tartalomban megbízunk, így az ő esetében nem akarjuk alkalmazni a szűrést. Ezt könnyedén megtehetjük, ha fordított egyeztetés segítségével ellenőrizzük az IP-címet, valahogy így:

```
iptables -A FORWARD -i eth0 -p tcp ! \
   -s 192.168.0.5 --sport 80 -m string \
   --string ´|7F|ELF´ -j DROP
```

Sajnos a példánk több sebből is vérzik, és ezek ráirányítják a figyelmet a karakterlánc-egyeztető modul hiányosságaira. Először is, ezzel a szűréssel a megadott karakterláncot az adatsorban bárhol megtaláljuk, nemcsak a fájlok elején. A szabály hatására tehát nem kívánt egyezések is fellépnek, és olyan csomagokat is eldobunk, amelyeket nem kellene. Másodszor, ha a keresett karakterlánc két egymást követő csomagba oszlik el, akkor nem fogjuk felismerni. A modul csak akkor dolgozik eredményesen, ha a teljes karakterláncot megtalálja az adott csomagban.
Szép és jó tehát a `string` modul, de csak alapszintű szolgáltatásokat nyújt. Nem teszi lehetővé, hogy megkülönböztessük a kis- és nagybetűket, vagy meghatározzuk a keresett karakterlánc elhelyezkedését, ráadásul az adatfolyamon belül több csomagba szétosztott karakterláncok felismerésére sem képes. Nem mondhatjuk tehát, hogy ne lenne hová továbbfejleszteni a modult.

Kevesebb szabály: szebb az élet

Az `mport` bővítmény lehetővé teszi, hogy egyetlen szabállyal több kaput vagy kaputartományt is megadjunk. Alapesetben az iptables parancsokban csak egy-egy kapu vagy a szomszédos kapukat felölelő kaputartomány adható meg. Az mport telepítése után bonyolultabb szabályokat is megadhatunk: egyetlen paranccsal is megoldhatjuk például az X-terminálok használatának, a webezésnek és a levelezésnek az engedélyezését:

```
iptables -A INPUT -p tcp -m mport \
   --dports 80,110,21,6000:6003 -j ACCEPT
```

Az mport nélkül mindehhez négy külön parancsra lett volna szükség:

```
iptables -A INPUT -p tcp --dports 80 -j ACCEPT
iptables -A INPUT -p tcp --dports 110 -j ACCEPT
iptables -A INPUT -p tcp --dports 21 -j ACCEPT
iptables -A INPUT -p tcp --dports 6000:6003 \
       -j ACCEPT
```

Négy helyett egyetlen szabály – akár a rendszer teljesítményének javulását is eredményezheti, hiszen a csomagok feldolgozása kevesebb munkával jár. A szabályfájlok karbantartása is egyszerűbb, hiszen az azonos jellegű csomagkezelést igénylő szolgáltatásokat csoportosítani tudjuk. Mint néhányan talán már kitalálták, az `mport` név a "multiple ports", azaz a "többszörös kapu" szókapcsolat rövidítése.

Időalapú szabályok

A `time modul` segítségével a szabályok érvényességét időszakok és napok alapján tudjuk korlátozni. Megtehetjük például, hogy a személyes weboldalak elérhetőségét az ebédidőre korlátozzuk, vagy a szokásos karbantartások időtartamára a webes forgalmat egy tartalék kiszolgálóra irányítjuk.
Az alábbi szabállyal minden péntek reggel 4:00 és 6:30 között elérhetetlenné tehetjük a webszolgáltatást, például a rendszer karbantartása miatt:

```
iptables -A INPUT -p tcp -d 80 -m time \
   --timestart 04:00 --timestop 06:30 --days Fri \
   --syn -j REJECT
```

Megjegyezném, hogy a `-timestart`, `-timestop` és a `-days` kapcsolók egyikét sem szabad elhagyni. Ha tehát olyan szabályt akarunk írni, amely napoktól független, akkor mind a hét napot fel kell sorolnunk, mivel a kapcsolót nem hagyhatjuk el.

Csapdában

Az iptables TARPIT összetevőjét amolyan hálózati légypapírnak is nevezhetnénk. Az alapelv: aki túl közel jön, az nem egykönnyen távozik. Aki elég merész ahhoz, hogy TCP/IP kapcsolatot nyisson egy légypapírkapura, az csak nagyon nehezen tudja azt lezárni, így rendszerének erőforrásait sem tudja felszabadítani, és újra felhasználni.
A csapdaállítást az iptables úgy végzi, hogy először is fogadja a bejövő TCP/IP kapcsolatot, ezt követően pedig nulla méretű ablakra vált. Ilyenkor a támadó nem tud több adatot küldeni. Nagyjából olyan a hatás, mintha CTRL-S billentyűkombinációt nyomnánk egy terminálon.
A támadónak a kapcsolat lezárására irányuló próbálkozásait figyelmen kívül hagyjuk, így a kapcsolat aktív marad, és jellemzően csak 12-24 perc elteltével bomlik fel. A módszer a támadó erőforrásait felemészti, a légypapírt futtató Linux kiszolgáló vagy tűzfal erőforrásait viszont nem. A légypapírra a következő paranccsal irányíthatunk át csomagokat:

```
iptables -A INPUT -p tcp -m tcp -dport 80 -j TARPIT
```

A `conntrack` és a `TARPIT` azonos rendszeren való használata erősen ellenjavallt, ugyanis minden beragadt kapcsolat fogyasztja a conntrack erőforrásait.
Nagyon jó módszer a támadók megzavarására, ha a netbios kapukat hagyjuk válaszolni a kapupásztázásokra, és ezzel linuxos rendszerünket Windowst futtató gépnek láttatjuk.

Az ezekre befutó próbálkozásokat csak át kell irányítanunk a légypapírra. A hatás lenyűgöző, a támadó hosszasan fog vacakolni azzal, hogy a biztonsági résen keresztül hozzáférést próbáljon nyerni rendszerünkhöz.
A hibás működésűnek tűnő távoli gép, és a hosszúra nyúló időtúllépések jó eséllyel az őrületbe fogják kergetni. Ha ezt a hatást kívánjuk elérni, a következő szabályt kell használnunk:

```
iptables -A INPUT -p tcp -m tcp -m mport \
   --dports 135,139,1025 -j TARPIT
```

Egy másik lehetőség, ha minden kaput TARPIT-játékszerré avatunk, kivéve az eredetileg is használni kívántakat. Ilyenkor a kívülállók úgy látják, mintha minden kapu nyitva lenne, és rengeteg időt pazarolnak a bejutásra. Ráadásul ilyesfajta beállításokkal megakadályozhatjuk a *tcpdump*ot a kiszolgálón futó operációs rendszer helyes felismerésében. Az alábbi szabályokkal a webes és a levelezési forgalmat engedélyezzük, a többit a légypapírra pöcköljük:

```
iptables -A INPUT -p tcp -m tcp --dport 80 -j ACCEPT
iptables -A INPUT -p tcp -m tcp --dport 25 -j ACCEPT
iptables -A INPUT -p tcp -m tcp -j TARPIT
```

Megemlíteném, hogy a ➲ www.spinics.net/lists/netfilter/msg17583.html címen egy érdekes, valóban megtörtént esetről olvashatunk, ahol a TARPIT és a string húzott ki a bajból egy rendszergazdát.

Véletlenül

A random modul nem tesz mást, mint véletlenszerűen mutat egyezést a csomagal. Működését úgy szabályozhatjuk, hogy megadjuk az egyezés valószínűségét, ennek értéke 0 és 100 százalék közé eshet. A modul alkalmas például meghibásodott kapcsolat vagy kiszolgáló szimulálására, illetve a terhelés elosztására több, tükrözött kiszolgáló között. Az alábbi példa szerint a webes forgalmat három kiszolgáló között osztjuk el. Az első szabály a kapcsolatok egyharmadát a 192.168.0.100 címen üzemelő kiszolgáló felé irányítja. A második egyharmad a 192.168.0.101, a maradék pedig 192.168.0.102 címre jut:

```
iptables -t nat -A PREROUTING -i eth0 -p tcp \
   --dport 80 --syn -m random --average 33 \
   -j DNAT --to-destination 192.168.0.100:80

iptables -t nat -A PREROUTING -i eth0 -p tcp \
   --dport 80 --syn -m random --average 50 \
   -j DNAT --to-destination 192.168.0.101:80

iptables -t nat -A PREROUTING -i eth0 -p tcp \
   --dport 80 --syn -j DNAT \
   --to-destination 192.168.0.102:80
```

Többet és többet

Némi kutakodással számolatlan további kincset találhatunk – és ezek rengeteg örömet fognak szerezni. Itt csak néhányat említhettem meg, s ez csak elenyésző töredéke a kincses ládikó tartalmának. Ha tudni akarjuk, mi minden áll rendelkezésünkre, futtassuk a ***runme*** parancsfájlt, és böngésszük át a foltok leírásait. Csak néhány kiragadott példa a gyűjteményből:

- Kapcsolatkövetés RSH, MMS (médiafolyamok), PPTP, Quake, RPC és Talk forgalomhoz
- A beállítások és állapotadatok /proc fájlrendszeren keresztüli elérésének kiterjesztett támogatása
- Az IPv6 szolgáltatásainak kibővített támogatása
- Az IP-csomagokban található beállítások, élettartamérték stb. módosítása
- A NAT-olt kapcsolatok kifinomultabb kezelése
- Sávszélesség-használatra vonatkozó korlátok és kvóták kezelése
- A futó operációs rendszer lekérdezését meghiúsító, és a kapupásztázást felismerő megoldások
- Kapcsolatok megjelölése és a jelölések ellenőrzése

Tudástárak

A POM-ban szereplő foltok leírásai nem kerülnek be az ***iptables man*** oldalai közé, ezért azokat máshol kell keresnünk. A bővítmények használatához szükséges alapvető írásmódot az iptables beépített súgójával jeleníthetjük meg. Ha például kiadjuk az `iptables -m random -help` parancsot, akkor egyrészt a szokásos súgószöveget látjuk, másrészt ennek végén a random modul kapcsolói is felbukkannak. Hasonló eljárással a többi modulról is nyerhetünk alapszintű tájékoztatást.
Az egyes modulok leírásait a Patch-o-matic könyvtárszerkezetének megfelelő pontjain is megtaláljuk. Például a random modul leírása a ***base/random.patch.help*** fájlban szerepel. Hasonló fájlok tartoznak a többi folthoz is.
Végül érdemes ellátogatni a Netfilter webhelyére, a ➲ www.netfilter.org/patch-o-matic címre, itt minden POM folt részletes ismertetését megtaláljuk.

Új iptables modulok telepítése

Hogy a POM modult hogyan adjuk hozzá a rendszermaghoz, illetve hogyan vegyük használatba az iptables eszközöket, a ➲ www.lowth.com/howto/add-iptables-modules.php oldalon olvashatunk.

Összefoglalás

Láttuk, hogy a linuxos Netfilter számos nagyszerű szolgáltatással segít bennünket rendkívül hatékony tűzfalat építeni, ám a legtöbb Linux-terjesztés a szükséges kiegészítések jelentős részét alapesetben nem tartalmazza.
A Patch-o-matic gyűjtemény segítségével a rendszergazdák megfoltozhatják a rendszermagot, és kibővíthetik tűzfaluk alapszolgáltatásainak körét.

Linux Journal 2004. április, 120. szám

Chris Lowth az Intercai Mondiale (www.intercai.co.uk), egy angol távközlési, informatikai és üzleti tanácsadó cég munkatársa. Biztonsági programokat és hálózatfelügyeleti (OSS) megoldásokat tervez, néha megpróbálkozik a gitározással is. Chris a chris@lowth.com címen érhető el.

Fürtözött eszközök kimerítő tesztelése

Gondos hardvertesztelési terv kidolgozásával rengeteg időt, pénzt és géptermi kábelezést takaríthatunk meg.

Üzembiztos hardver nélkül még a legjobb program is elszáll. A hibás eszközök támogatásával kapcsolatos gondok és költségek akár egy egész szervezet erőforrásait leköthetik, késleltethetik a munkafolyamatokat és minden érintettet az idegbaj szélére sodorhatnak.
A Stanford Linear Accelerator Centerben (SLAC) kidolgoztunk egy tesztelési módszert, amellyel szervizcsoportunk, a SLAC Computer Services (SCS) kiszűrheti a hibás eszközöket, miközben a legjobb alkatrészeket a lehető legjobb áron tudja megvásárolni. Cikkemben számítógép-tesztelési módszereinket ismertetem, továbbá szólok néhány szót az elfogadási határértékekről is. Az itt leírt módszerek abban segítenek minket, hogy a korábbiaknál üzembiztosabb és könnyebben támogatható hardvereszközöket válasszunk ki. Írásom alapjául egy tényleges eszközbeszerzés szolgált, de a módszer főbb vonalaiban bármilyen termelési környezetbe szánt számítógéprendszer megvásárlás előtti vizsgálatára alkalmas.

A rendszerkövetelmények meghatározása

Egy növekvő fürtözött rendszer egységességét meglehetősen nehéz megőrizni, mivel az elérhető eszközök köre gyakran változik. Sajnos emiatt a rendszer bonyolultsága növekszik, üzembiztossága csökken, és a szoftveres oldal támogatása is több erőforrást köt le. Az új eszközök üzembe helyezésével újfajta hardverhibákra is fény derülhet. A változások korlátozására és rendszereink hatékony kezelésére az SCS számos eszközt és előírást fejlesztett ki, melyek segítségével az új eszközök könnyebben illeszthetők be felügyeleti és adatfeldolgozó keretrendszerünkbe. A szolgáltatások körét a lehető legszűkebbre vettük, persze úgy, hogy a felügyeleti infrastruktúra megfelelően működhessen, és kódunk is valóban érvényes eredményt szolgáltasson. Követelménylistánk a következő volt:

- Egy egység (1U) magas ház, 19"-os állványhoz méretezve
- Legalább két darab 1 GHz vagy magasabb órajelű Intel Pentium III processzor
- Legalább 1 GB ECC memória két processzoronként
- 100 MB-es Ethernet csatoló PXE-támogatással a hálózati csatoló és a BIOS oldaláról
- Soros konzol támogatása BIOS-szintű hozzáféréssel
- Egy darab 9 GB-os vagy nagyobb merevlemez, 7200-as vagy magasabb fordulatszámmal
- Minden gépnek FCC- és UL-megfelelőnek kell lennie.

A követelménylista összeállítása az eszközértékelés egyik első lépése volt. Talán meglepő, de a feltétlenül szükséges szolgáltatások kiválasztása és a „jó lenne, ha tudná” jellegűek elhagyása eleve komoly feladat volt. A leegyszerűsített követelménylistával csökkentettük a szállítók körét, és szükségtelen szolgáltatások által gerjesztett bonyodalmakat. A lista alapján 11 szállítót hívtunk meg ajánlattételre és termékteszte. Mivel nemcsak egy modellel lehetett pályázni, összesen 13 gépet teszteltünk.

A tesztelés kezdete

A pályázaton való részvétel alapfeltétele egyrészt a követelménylista teljesítése volt, másrészt három gép küldése kipróbálásra. A gépeket 90 napra kértük el, mert így elég időt kaptunk a gondos vizsgálódásra. Minden értékelésnél három géppel dolgoztunk, kettő állványra szerelve működött, a harmadikat pedig egy asztalon próbálgattuk. A későbbiek során az asztalra helyezett rendszeren tápellátási és hűtési teszteket is végeztünk. A másik két gépet pontosan úgy szereltük fel az állványra, ahogy a fürtözött gépeinket is, de az éles gépfürtbe nem vettük be őket. Néhány gép egyedi fizikai és szerelési jellemzőkkel rendelkezett, ezeket nem tudtuk használni. Munkacsoportunk weboldalán minden hibát feltüntettünk, és megpróbáltuk felkérni a gyártókat ezek kijavítására. Így a szállítók együttműködésre való hajlandóságát is ki tudtuk próbálni. Volt, aki egyszerűen kicserélte a hibás gépet, míg mások gondosan, velünk együttműködve kivizsgálták a hibát. Volt, aki úgy vélte, hogy a hiba valójában nem is hiba, egy vagy két szállító pedig egyáltalán nem foglalkozott velünk. A tesztelési időszak alatt minden gépen hardverellenőrzőt és – legalább 30 napon keresztül – bejáró parancsfájlt futtattunk.

Fizikai vizsgálatok

Az asztalon boncolgatott gépeket vizsgáltuk az összeépítés minősége, a fizikai tervezés, a hozzáférhetőség, a tápegység minősége és a hűtőrendszer szempontjából. Találtunk összevissza hajtogatott, összenyomkodott adatkábeleket, amelyek a légáram útját több helyen is elzárták, a házak meghajlottak, a hozzáférést csavarok tömkelege gátolta. Sok esetben

már a tervezés is átgondolatlan volt, volt olyan gép például, amelynek főkapcsolóját a hátuljára rakták. Figyeltünk a légáramlás és a hűtés minőségére, a méretre, a súlyra és a rendszer elrendezésére is. Például a kiegészítő meghajtó elhelyezésére szolgáló bővítőhely meglétét jó pontnak vettük. A légáramlás az erősen melegedő x86 processzoroknál rendkívül fontos, különösen a szűkös gépházaknál. A gépek közt volt olyan, melyben a légáram útját elzárták, így a légmozgás elenyésző volt, vagy egyáltalán nem is volt. Ugyancsak ide tartozik a ház merevsége, a lemezek éleinek lekerekítése, összeépítésének és kábelezésének minősége. Ezekre a másodlagosnak tűnő dolgokra azért figyeltünk, mert nekünk ezeket a gépeket évekig kell használnunk, ezért a legkisebb hiba is sok-sok kínlódás és költség okozója lehet. A szerszám nélküli szerelhetőség minden állványra szánt gépnél alapvető kellene hogy legyen. Ha a szervizköltségeket alacsonyan akarjuk tartani, biztosítani kell az alkatrészek gyors és könnyű cserélhetőségét. Az alkatrészeknek lehetőleg úgy kell elhelyezkedniük, hogy cseréjükhöz másik alkatrészt ne kelljen kivenni, és ne kelljen csavarozgatni. A tápegység minőségét egy Dranetz-BMI tápminőség-elemzővel vizsgáltuk. A tápegység korrekciós jellemzőit sokszor feltüntetik a gépekhez mellékelt leírásban, de a mért és az ígért adatok gyakran jelentősen eltértek egymástól. Volt például olyan tápegység, amelyhez a gyártó 0,96-os korrekciós tényezőt tüntetett fel, míg a mért adat 0,49 volt. 512 géppel felszorozva mindenki elképzelheti az eredményt. A gépeket tétlen állapotban és erős terhelés alatt is megvizsgáltuk. A minőségi szórás egészen elképesztő volt; végső döntésünkben egyértelműen ez játszotta a legnagyobb szerepet.

Tesztelés programokkal

A bejáratás, során a gyártók megpróbálják leterhelni a rendszert és megtalálni a hibás összetevőket. Több bejáró program is létezik, ezek egyike ➲ http:\\sourceforge.net/projects/va-ctcs címen elérhető Cerberus Test Control System. Ez tesztek és módosítható burkoló-parancsfájlok gyűjteménye, eredetileg a VA Linux rendszergyártásához tervezték. A Cerberus bejáratáshoz kiváló, de mi saját tesztprogramokat is készítettünk. Ezekkel sokkal sikeresebben tudtuk összeomlasztani a gépeket, mint egy általánosabb eszközzel tudtuk volna. A leírást gondosan tanulmányozzuk át, és vigyázzunk, mert a bejáró programok kárt is okozhatnak a számítógépben. A Cerberus futtatásánál fel kell vállalnunk ezt a kockázatot. Nyilván egy fürt építésénél a megbízhatóság terén cseppnyi engedmény sem tehető, és a bejáró parancsfájlok pontosan ilyen szintű próbákhoz használhatók. Amikor a gépeket vizsgáltuk, az éles rendszerben tapasztalt hibák alapján csapatunk két tagja foglalkozott saját bejáró programok megírásával. Míg a teljesítménytesztek kifinomult módszerekkel mérik a gépek teljesítményét, a bejáró parancsfájlok egészen egyszerűek. A rendszert erős terhelésnek tesszük ki, és meglátjuk, hogy bírja-e vagy sem. Ha a összeomlik, akkor nem felel meg.

Teljesítmény

Teljesítménytesztekhez programok garmadája érhető el. A legjobb természetesen az, ha a termelési kódot futtatjuk a kiszemelt gépeken, akár már a bejáratási fázisban. Sajnos erre nem mindig van mód, ekkor az átlagos teljesítménytesztek is kiválóan megfelelnek. A normál teljesítménytesztek a különböző gépek teljesítményének összehasonlítását is lehetővé teszik. Számos különböző tesztprogramot gyűjtöttünk össze, ezekhez írtunk egy burkolót – így jött létre a HEPIX-Comp (High Energy Physics—Compute) csomag. Kényelmes eszköz, de önmagában nem használható teljesítménytesztként, inkább a gép különféle jellemzőinek mérésére alkalmas. A HEPIX-Comp – többek között – a következő programokat foglalja magába: Bonnie++, IOZone, Netpipe, Linpack, NFS Connectathon, Streams. Az adott rendszeren futtatandó termelési kód jellegzetességeinek ismerete nélkül a teljesítménytesztek eredményét nem lehet érdemben értékelni. Ha például az alkalmazás erősen támaszkodik a hálózatra, akkor a hálózati sávszélesség és a késleltetés fontosabb, mint mondjuk a gyors előoldali sín. A gépek különféle adottságainak mérésére kiváló programok léteznek. A Streams például a memória be- és kiviteli alrendszer átviteli sebességét méri, amely elsősorban a több szintre szervezett memóriával rendelkező gépeknél fontos mutató. Sok gyártó a legjobb képet próbálja festeni, amikor termékének teljesítményét közli. A soros írások sebességét például be- és kiviteli teljesítményként megadni sokkal előnyösebb, mint a kis adatmennyiségeket véletlenszerűen kiíró műveletek sebességét feltüntetni. Ha egységesített tesztkörnyezettel rendelkezünk, amely az éles gépeken is üzemelő Linux-telepítés alatt fut, akkor a mérési eredmények összehasonlítása sem okozhat gondot. Az egységesített tesztrendszer legfontosabb előnye az egyenlő bánásmód. Természetesen így nem feltétlenül lehet kimutatni, hogy melyik gép teljesítménye a legjobb, de nem is biztos, hogy ez a cél.

Ami nem a műszakiak feladata volt

A szerződési feltételek, garancia és a határidők szintén fontos szerepet játszanak az új rendszerek munkába állításában. A garanciális feltételek alapvetően befolyásolja a rendszerek hosszú távú költségeit. Nem mellékes a szállító cég pénzügyi helyzete sem. Hiába ugyanis a kedvező garanciális vállalás, ha a cég nem teljesíti azt. Nem szabad megfeledkezni az átvételi feltételekről sem, bár ezekre sokszor csak akkor gondolnak, amikor már késő. Ezekben a feltételekben kell megfogalmazni, hogy a gépek telepítését mikor tekintjük befejezettnek, és mikor vagyunk hajlandóak átvenni a rendszert. Megeshet, a már kiszállított rendszer hibáira az éles üzem megkezdésekor derül fény. Az ilyen esetekre a vevőnek jogot kell biztosítani arra, hogy a hiba javítását vagy a gépek elszállítását kérje. A szállító részéről viszont jogos elvárás a hardveres és a szoftveres hibák elkülönítése. Sokszor a szállítónak a vevővel együttműködve kell megállapítania a hiba jellegét, ennek költségét be kell építeni a rendszer árába.

A végeredmény

Az ismertetett eljárás sikerét igazolja, az eredményeképp kiválasztott rendszert lényegesen könnyebb és olcsóbb volt üzemeltetni, mint azon gépeinket, amelyek vásárlásakor ateszelés elmaradt.

Linux Journal 2003. november, 115. szám

John Goebel

Az SQL-Ledger használata üzleti célokra

Ha pusztán azért használsz zárt forrású operációs rendszert, mert kizárólag azon tudod futtatni a számlázóprogramot, akkor elszalasztod az esélyét annak, hogy egy rugalmasabb, jobban támogatott rendszered legyen.

Valamikor az 1990-es évek vége felé megpróbáltam egy jó számlázóprogramot keresni Linux alá. Lehangoló volt az eredmény, egyik program sem tűnt igazán használhatónak, és soknál még a reményt sem éreztem arra, hogy valaha is azzá válna. Aztán nagyjából három és fél évvel ezelőtt belebotlottam a ***Dieter Simader*** által írt SQL-Ledgerbe (SL). Megvallom őszintén, első látásra úgy tűnt, hogy az SQL-Ledger komoly vállalkozásnak nem való. Nem volt benne pénztártámogatás, nem tudott bérkifizetéseket kezelni és számos további szolgáltatást sem ismert. Mivel azonban könnyen lehetett telepíteni, rugalmas volt, illetve egyéb tényezők miatt mégis úgy döntöttem, hogy kipróbálom.

Elkezdtem IT tanácsadó cégemben, a Pananix SA-ban az SQL-Ledgert használni – és több mint megfelelőnek bizonyult! Tudtam vevőket és eladókat megadni, számlákat és rendeléseket készíteni, alapszintű jelentéseket nyomtathattam, többek között főkönyvi kivonatot, eredménykimutatást és mérleget. Az SL a maga kezdetleges módján még a szolgáltatások és a termékek nyilvántartását is támogatta, tudott raktárlistát és szolgáltatáslistát készíteni.

Mivel jelenleg spanyol nyelvű országban élek, sejtettem, hogy lesznek kisebb gondok. A spanyol számviteli kifejezéseket nem nagyon ismerem, ám a jogszabályok előírják, hogy a program felületének spanyol nyelvűnek kell lennie, valamint a bizonylatoknak és a jelentéseknek spanyolul kell készülniük, már csak a helyiek és az adóellenőrök miatt is. Szerencsére az SL-t úgy készítették, hogy használója kiválaszthatja az általa kívánt nyelvet. Vagyis míg én mindent angolul látok, addig a könyvelőm és az adóhatóság emberei spanyol felületet kapnak – ami már tökéletesen legális. Ha egy kívánt nyelv nem szerepel a programban, a hozzáadás könnyedén megoldható. Ez a lehetőség a többi, általam megvizsgált számlázóprogramból vagy kimaradt, vagy túl gyengén valósították meg; ez lett az egyik fő oka annak, hogy az SL mellett döntöttem.

Telepítés

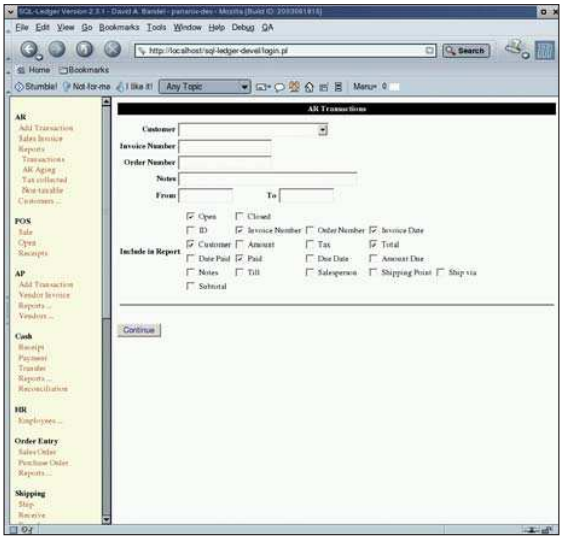
A telepítés már első találkozásunkkor is könnyű volt, és azóta csak tovább egyszerűsödött. A legnagyobb probléma a legtöbb felhasználó számára az lehet, hogy az adatbáziskezeléshez szükség van néhány Perl-modulra. További nehézséget jelenthet a PostgreSQL beállítása, ám a leírás és a GYIK minden lépést részletesen ismertet. Amíg képesek vagyunk betűről betűre követni őket, addig kezdőként sem lehet semmilyen gondunk. Igazi hibák csak akkor jelentkeznek, amikor a kedves felhasználók elkezdenek variálni, és szándékosan vagy szándékolatlanul módosítják a jogosultságokat, akár magában az adatbázisban is.

1. ábra Az SQL-Ledger induló oldala

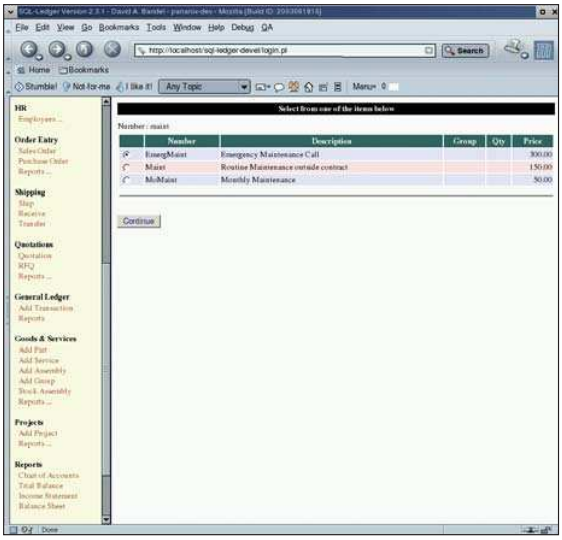
Ha a Perl és valamelyik támogatott adatbázis-kezelő felkerült a gépre, a többi már gyerekjáték. A MySQL egyelőre nem támogatott, és nem is lesz az, míg az SQL-92 szabvány bizonyos, jelenleg hiányzó részeit meg nem valósítják benne. A tervezés és a próbák során PostgreSQL-t használtak, de az Oracle és a DB2 is megfelelő.

A webkiszolgáló oldalán is meg kell adni néhány beállítást, mielőtt az SL-t elérhetnénk, de ehhez csak be kell másolni néhány sort a ***httpd.conf*** fájlba, majd újra kell indítani a szolgáltatást. Ha az SL-t a DocumentRoot könyvtárba telepítjük, akkor akár el is hagyhatjuk ezt a lépést, feltéve, hogy a `DocumentRoot` alatt jogunk van CGI parancsfájlokat futtatni. Ha a jelentéseket nyomtatás helyett PDF fájlok formájában szeretnénk a lemezre menteni, akkor ellenőrizzük, hogy fel van-e telepítve gépünkre a LaTeX.

Az SL telepítésének legjobb módja a szerző által összeállított ***setup.pl*** fájl futtatása, ez minden szükséges tennivalót elvégez. Így a legkisebb az esélye annak, hogy bármit is elrontanánk, és a parancsfájl az SL-t is frissíti. A szerző nemrég olyan kódrészleteket is írt, amelyekkel ellenőrizhető az adatbázis, és ezzel biztosítható, hogy annak tartalma és



2. ábra A vevő tranzakciók lekérdezésének képernyője



3. ábra Választólista

a frissítéskor telepítendő programváltozat együttes használatra alkalmasak legyenek. Ha a program ilyen eltérést talál, akkor önműködően frissíti az adatbázis szerkezetét is, amiért, úgy vélem, külön dicséret illeti a szerzőt.

Beállítások és biztonság

Aki követi írásaimat, jól tudja, hogy a biztonságnak mindig kiemelt figyelmet szentelek. Bizonyára minden könyvelő örömmel veszi tudomásul, hogy az SL alatt a biztonsági beállítások felhasználónként adhatók meg. Megoldható például, hogy az egyik felhasználó csak a bejövő, míg a másik csak a kimenő számlákat lássa.
Az SL a legtöbb esetben beállítható úgy, hogy megfeleljen az általánosan elfogadott számlázási elveknek (GAAP).
A legtöbb országban külön GAAP-változat létezik, ám az alapelvek nagyon hasonlók. Az SL-ben megoldható, hogy például a felhasználók ne léphessenek vissza és ne vonhassák vissza a tranzakciókat, hanem egy másik, ellentétes előjelű tranzakciót legyenek kénytelenek elvégezni. Az időbeli szakaszokat le is zárhatjuk, így azokban visszamenőleg semmit nem lehet módosítani.
Mindezt a felügyeleti programrészből végezhetjük el. Itt történik továbbá a felhasználók hozzáadása, illetve a rendszer viselkedésének alapvető szabályozása is. Lényegében mi magunk dönthetjük el, hogy mennyire biztonságos vagy nyitott rendszert szeretnénk összeállítani.

Az SQL-Ledger futtatása

Az SL-ben minden a számlatükör körül forog. Telepítéskor megadhatjuk, hogy kívánunk-e ilyet betölteni. Ennek későbbi módosítása nem fog nehézséget okozni, és akár újat is létrehozhatunk. A legtöbb vállalkozásnál valószínűleg szükség lesz kisebb változtatások elvégzésére.
Az SL adókezelő alrendszere rendkívül rugalmas, gyakorlatilag bármilyen adórendszerhez hozzáigazítható, ha a megfelelő táblákat a megfelelő százalékkulcsokkal egymáshoz kapcsoljuk. Mivel mindezt nagyon könnyen megtehetjük, az SL kiváló választás olyan helyeken, ahol az adózási szabályokból fakadóan két vagy három különböző adó is terheli az értékesítéseket. Az alapértelmezett számlatükörben – pusztán a szemléltetés kedvéért – három adó szerepel.
Az adótáblák többszörös hivatkozásokkal kapcsolódnak a fogyasztókhoz, a szállítókhoz, az árukhoz és a szolgáltatásokhoz, és egyezések keresése alapján történik annak eldöntése, hogy egy meghatározott adónemet alkalmazni kell-e. A vevők és a szállítók kezelése egymástól függetlenül történik, ráadásul a vevőkhöz és a szállítói termékekhez, szolgáltatásokhoz tartozó adóbeállításokkal rendkívül rugalmas adószámítási rendszert alakíthatunk ki. Megadhatunk negatív adókulcsot is, hogy kezelni tudjuk a munkáltatói oldal által fizetett adókat és az adóztatott adókat. Az itteni adórendszer például megköveteli, hogy az adóztatott tételekhez csatolt szolgáltatásokra is adót számítsunk fel, holott a szolgáltatásokra egyébként nem számítunk adót. Meg kellett tehát adnom egy adóztatott és egy másik, adómentes szolgáltatáshívást.
Így az ügyfelek nem veszik észre, hogy időnként van adó egy szolgáltatáson, máskor pedig nincs. Mivel az adóztatott szolgáltatás mindig hardvereladással párosult, egyszerűen megadtam egy telepítési csomagot, amely a hardvert és az adóztatott szolgáltatást is tartalmazza, és így az egész csomagra vonatkozik az adófizetési kötelezettség. Egyetlen más linuxos számlázóprogramot sem láttam, amely ilyen mértékű rugalmasságot biztosított volna.
Az SL bármilyen webböngészővel ellátott gépről hozzáférhető, akár grafikusan, akár szövegesen. Ehhez az kell, hogy az SL-t futtató kiszolgálót képesek legyünk HTTP-n vagy HTTPS-en keresztül elérni. Ha az emberek túlnyomó részéhez hasonlóan grafikus böngészőt használunk, akkor bejelentkezés után két keretet láthatunk. A bal oldali a több kategóriára osztott menüpontokat tartalmazza, a jobb oldali pedig a fő képernyő, itt adhatunk meg adatokat.
Nekem minden frissítés után az első dolgom az, hogy belépek a ***bin/mozilla*** könyvtárba, és a ***menu.pl*** átírásával kiszé-

Amikor a nyílt forrás működni kezd

Az SQL-Ledger szerzője Dieter Simader. A munkát négy évvel ezelőtt kezdte. A program azóta egyre népszerűbb lett, most már teljes időben dolgozik rajta. Programját nyílt forrással, GPL szerződés hatálya alatt terjeszti, másoktól nem kér díjat a használat jogáért, amivel fő bevételi forrásáról mondott le.
Vajon hogyan tud megélni egy ingyenes, szabad programból? Ha ellátogatunk az SL webhelyére, láthatjuk, hogy Dieter számos támogatási szolgáltatst kínál, mindegyiket igen méltányos ron. Programja kiváló minőségű, ajánlatai között egyaránt találunk adatátalakítási szolgáltatást, igénybevétel szerinti fizetési lehetőséget és a fejlesztői változatra vonatkozó támogatási szolgáltatást - arra az esetre, ha különleges szolgáltatásokat várnánk a programtól. Ugyancsak térítés ellenében kínálja a program kézikönyvét, melyben nemcsak a használat módját, de a képernyők egy vagy több felhasználó számára, az SL API-jának segítségével történő testreszabását is ismerteti.
Ezek az értéknövelt szolgáltatások bőven megérik árukat, minden vállalkozás számára hasznosak lehetnek. Persze egy vállalkozás felfogadhat mást is, aki átvállalja Dieter munkáját, és támogatást nyújt, egyedi sablonokat készít, adatokat emel be, de kevesen végzik el mindezt olyan jól, mint ő.
Magam is tanácsadó vagyok, az ügyfeleim általában velem akarnak dolgozni, és nem egy ismeretlennel a világ másik feléről. Ezt a kényelmet hajlandóak is megfizetni nekem. Ráteszek tehát néhány százalékot Dieter áraira, megfizetem az ő munkáját, és mindketten jól járunk. Ő munkához jut, programjának támogatásáért pedig pénzt kap. Én mérgelődési jutalékot kapok, ugyanis ügyfeleim engem fognak idegesíteni, ha a program működése nem felel meg tökéletesen az elvárásaiknak. Ezzel a módszerrel nagyon jól be lehet törni olyan piacokra is, amelyek nyelvét a szerző nem beszéli, vagyis a jutalék akár fordítási díjként is felfogható. Az SQL-Ledger kiváló bizonyíték arra, hogy a nyílt forrásra épülő üzleti modell igenis működőképes.

lesítem a menüsort. Az eredeti az én ízlésemnek ugyanis kicsit keskeny, 35 képponttal bővítve sokkal kellemesebb látványt nyújt. Ha szöveges böngészővel közelítünk, például a kereteket nem kezelő Lynxszel, akkor a menü elemei az oldal alján jelennek meg.
A megjelenő kategóriák köre a felügyeleti oldalon megadott beállításoktól függ. A menüből szimpla menüpontokat, de akár teljes kategóriákat is el lehet távolítani. Előfordulhat tehát, hogy valaki – a beállításoktól és a futtatott változattól függően – az itt szereplőknél gyérebb menüvel találkozik. Az 1. ábrán látható ernyőképen tisztán látszik, hogy a 2.3.1-es változatról készült, mely egy fejlesztői változat. Valamivel több szolgáltatás található, mint az üzembiztos verzióban, ám a nem üzembiztosként való megjelölés figyelmeztet arra, hogy nem esett át olyan gondos ellenőrzéseken, mint a párja. A fontosabb menüpontok: Vevők (AR), Nyugta (POS), Szállítók (AP), Pénzmozgások (Cash), a Személyzeti nyilvántartás (HR), Rendelések (Order Entry), Szállítások (Shipping), Árajánlatok (Quotations), Főkönyvi könyvelés (General Ledger), Áruk és szolgáltatások (Goods & Services), Munkaszám (Projects) és Jelentések (Reports). Van néhány olyan menüpont, amelynek neve után három pont látható, ezekben almenüpontokat találhatunk.
A Törzsadatok… (System...) pontra kattintva például az újabb változatokban egy hosszú listába jutunk.
Gyorsan nézzünk bele a Vevő / Jelentések (AR / Reports) menüpontba! A Jelentések (Reports) pontra kattintva kibővül a menülista. Ha ezután a Tranzakciók (Transactions) pontot választjuk, egy újabb képernyőre jutunk, ahol kiválaszthatjuk, hogy mely tranzakciókat szeretnénk látni, illetve milyen adatokra vagyunk kíváncsiak. (Lásd a 2. *ábrát*.)
Vegyük észre a 2. ábrán is szereplő, az ablak tetején látható Vevő (Customer) mezőt. Sok helyen találkozhatunk vele, akár legördülő lista, akár külön ablakban megjelenő választólista formájában. A legjobb azonban az, hogy megjelenési módját minden felhasználó külön állíthatja be. Ha tehát valaki szereti a legördülő listákat, és nem zavarja, hogy akár 16 oldalon keresztül kell görgetnie, mire eléri a kívánt nevet, írjon jó nagy számot az erre a célra szolgáló mezőbe. Aki rövidebb, könnyebben kezelhető listákat szeretne látni, az adjon meg kisebb értéket: addig látható ugyanis legördülő lista a képernyőn, amíg a benne szereplő elemek száma meg nem haladja a megadott számot. E felett elég, ha beírjuk a keresett név első néhány betűjét, frissítjük az ablak tartalmát, és máris megjelenik egy a 3. ábrán láthatóhoz hasonló, rövidke választólista.

Az SQL-Ledger használata üzleti célokra

Mint minden más nagy tudású számlázóprogramnál, itt is ismerni kell magát a programot ahhoz, hogy könyvelési feladatokra tudjuk használni.
Egy két esettől eltekintve a program használata mindeddig hiba mentes volt. A vevők listája például határozottan barátságos, és kiválóan alkalmas az e-mail címek nyilvántartására. A havi számlákat elektronikus levélben szoktam kiküldeni, az SL segítségével ez a feladat sokkal könnyebbé vált. Miután postáztunk egy számlát, gyakorlatilag egyetlen kattintással előhívhatjuk és elküldhetjük elektronikus levélben is.
A szállítói rendelések feladása is gyerekjáték, ezeket szintén elküldhetjük elektronikus levélben. Amikor rendeléseink beérkeznek, további néhány mozdulattal frissíthetjük a raktárkészletet és elkészíthetjük a szükséges bizonylatokat. Még egy gyors látogatás a Pénzmozgások/Kifizetések részben, és végeztünk is.
Ha több vállalkozást irányítunk, akkor eltérő adatbázisneveket választva tetszőleges számú példányt futtathatunk. Természetesen külön felhasználónevekre is szükség lesz, minden vállalkozáshoz egy, és ezt szerintem a legegyszerűbben felhasználó/vállalkozásnév kombinációkkal oldhatjuk meg. Akinek ez nem tetszik, válasszon kényelmesebb megoldást. Mivel a legújabb SL már pénztárkezelésre is képes, vonalkódolvasó birtokában a cikkszámokat is egyszerű eljárással vihetjük be. A programban a Frissítés (UPDATE) gomb kiemelt helyet kapott, így, miután a cikkszámot vonalkódról beolvastuk, a többi adatot már könnyedén csatolni tudjuk.

További mutatványok?

Az SL már most is rendkívül sok szolgáltatást nyújt, különösen a néhány évvel ezelőtti önmagához képest. Készítője azonban rendkívül ígéretes tennivalólistát tart fenn. Úgy vélem, mindazok alapján, amit az SL bizonyított, a bérlistakezelő rendszer bármely másikkal képes lesz felvenni a versenyt.
Az SL tennivalólistáján olyan elemek szerepelnek, mint:

- Költségvetések: Összehasonlítások az aktuális költségekkel, valamint a költséghatárok betartatása.
- Gyártás: Kész és gyártás alatt lévő termékek raktárkészlete, anyag és emberi erőforrás-tervezés.
- Tételek átrendezése: Tömegáruk átcsomagolása kisebb kiszerelésbe, tárolt tranzakciók és egyedi jelentések.
- Kötegelt űrlap-előállítás: Számlák, rendelések és egyéb bizonylatok kötegelt nyomtatása.
- Pénzügyi jelentések: További összehasonlítási lehetőségek, például hónap-hónap viszonylatban.

Akit érdekelnek a jövő újdonságai, fussa át az SL honlapjának „What's Ahead" oldalát.

Támogatás

Az SL-hez számos formában kaphatunk támogatást.
A program honlapjáról több felhasználói listát is elérhetünk, ezeken különféle nyelveken folyik az eszmecsere. Jelenleg hat felhasználói lista létezik, a szerzőt az angol nyelvűn lelhetjük fel. Alkalmankénti hozzászólásaival elsősorban a félreértéseket oszlatja el.

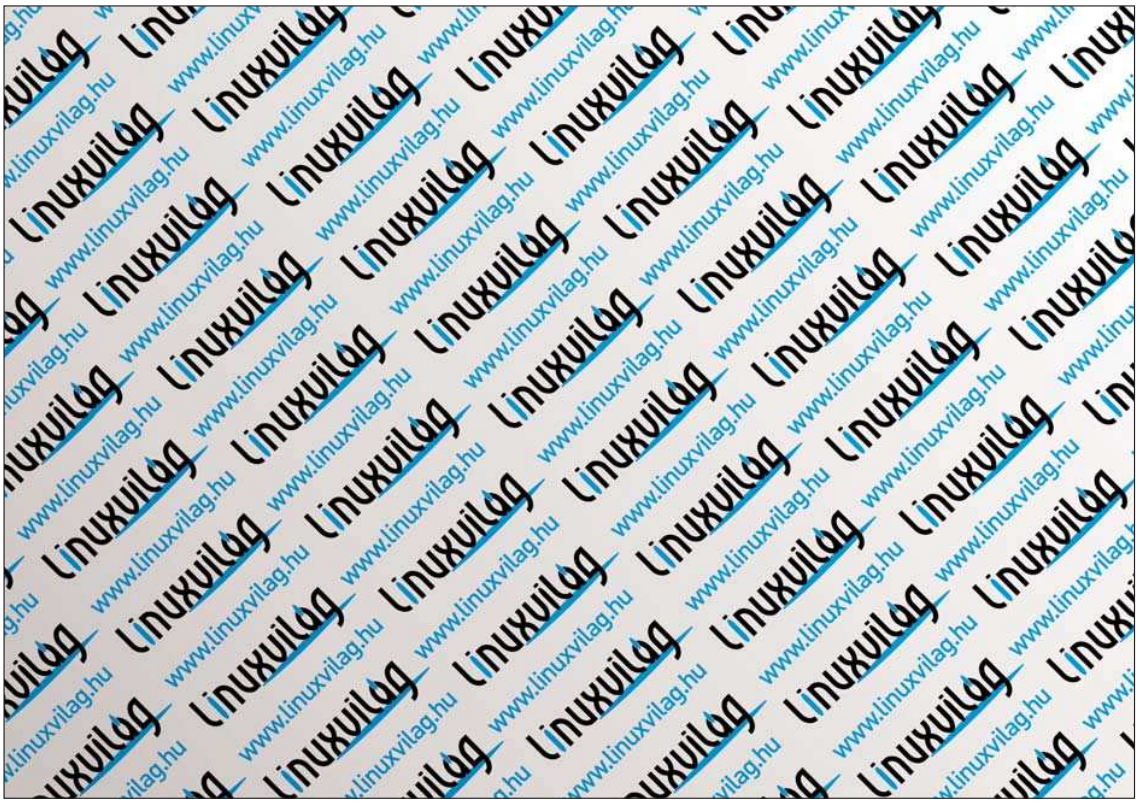
Fizetős támogatást ugyancsak kaphatunk, rendkívül kedvező áron. Több különböző támogatási forma létezik, a legtöbbhöz jár az SQL-Ledger kézikönyvének egy példánya.
Ha támogatás igénybe vétele nélkül akarjuk használni a programot, azt is nyugodtan megtehetjük, azonban, mint minden GPL programnál, tudomásul kell vennünk, hogy az érmének két oldala van. A fizetett támogatással garanciát kapunk a biztos működésre. A szolgáltatás keretein belül lehetőség van arra is, hogy adatok beemelését kérjük régebbi, tabulátorokkal tagolt szöveges fájlba írni képes számlázóprogramunkból.

Összefoglalás

Másik számlázóprogramra áttérni mindig kínszenvedés, de az SL-re való váltást érdemes megfontolni. Az ár nem lehet akadály, a program pedig számos fizetős társával bátran felveheti a versenyt.

Linux Journal 2004. május, 121. szám

David A. Bandel (david@pananix.com)
Linux/UNIX/hálózati (vezetékes és vezeték nélküli) tanácsadó Panamában, aki a telekommunikáció szinte minden területébe belekóstolt már. Szerzőként vagy társszerzőként három linuxos témájú könyv megszületésében segédkezett, két nagyfrekvenciás rádiós Sailmail állomást tart fenn, továbbá önkéntes alapon segíti a Linux Professional Institute munkáját. Amikor nem dolgozik, a Costa Rica tengerpartjának közelében fekvő farmján pihen.



OOo dokumentumok kezelése a Ruby segítségével

Miért kellene várnunk egy olyan operációs rendszerre, amely lehetővé teszi, hogy irodai dokumentumainkat az általunk fejlesztett alkalmazásokban használjuk fel? Dolgozzunk OpenOffice.org XML-alapú dokumentumainkkal a Ruby segítségével.

Nagyot fejlődött az utóbbi időben a szövegfeldolgozásra, táblázatkezelésre és bemutatók készítésére használható ***OpenOffice.org*** (OOo) nevű sokoldalú irodai alkalmazáscsomag. Az OOo forráskódja és fájlformátumai nevéhez méltóan teljesen nyilvánosak. Ami hatalmas előny azok számára, akik dokumentumaikat a létrehozó alkalmazások nélkül szeretnék módosítani.

OOo Extract

Először akkor figyeltem fel az OpenOffice.org dokumentumokban rejlő lehetőségekre, amikor ***Daniel Carrera*** bemutatta az ***OOoExtract*** nevű programját. Ez egy Ruby alkalmazás, melynek segítségével a parancssorból kereshetünk szövegrészleteket ***OOoWriter*** dokumentumokban. A honlapja szerint az OooExtract segítségével a szövegben és a stílusok között is kereshetünk, akár helyettesítő karakterekkel is, emellett a program képes logikai műveletekből felépített keresések futtatására is. A program minden olyan operációs rendszeren fut, melyhez létezik Ruby fordító.
A Rubyról már esett szó a Linuxvilágban. Aki nem ismeri, annak legyen elég annyi, hogy leginkább a Perl és a Smalltalk közti átmenetről van szó, némi Lisp és Python beütéssel. Erősen objektum-alapú, formanyelve tiszta és intuitív. Szerzője, ***Yukirhiro „Matz" Matsumoto***, az első alfaváltozatot 1994-ben adta ki.
Az OOoExtract kipróbálásához töltsük le a programot. Jelenleg az alkalmazást egyetlen futtatható fájlként, vagy tarcsomagként tölthetjük le, melyben a szükséges könyvtárak külön fájlokban találhatók. Telepítés után létrehozhatunk egy ***Writer*** dokumentumot és abban kereséseket hajhatunk végre. Ha kéznél van az OOo, akkor indítsuk el és írjunk be valami rövid szöveget, például:

```
Ez egy példa
Két sorból áll
```

Mentsük a fájlt ***pelda1.sxw*** néven ugyanabba a könyvtárba, ahová az OOoExtractot telepítettük és indítsuk el az OOoExtractot a parancssorból:

```
./ooo_extract.rb –text példa pelda1.sxw
Ez egy példa
```

A program ekkor a ***pelda1.sxw*** fájlban keresi a „példa" szót. Ez valójában egy egyszerű kifejezés. Bonyolultabbakat is használhatunk, például az alábbit, mely a három karakterből álló szavakat keresi:

```
./ooo_extract.rb –text "\s\w\w\w\s"  sample1.sxw
Két sorból áll
```

Ez így nagyszerű, de az OOoExtract egyik legnagyszerűbb tulajdonsága mégis az, hogy a metaadatok, azaz a dokumentumban lévő szöveggel kapcsolatos adatok között is kereshetünk. Tegyük fel, hogy a példaként felhozott Writer dokumentumot az alábbi sorral egészítjük ki:
Itt már a formázáson is változtattunk
A szöveg begépelése után jelöljük ki a „már" szót, stílusát állítsuk át a Footer (lapalji jegyzet) stílusra és futtassuk le az alábbi keresést:

```
./ooo_extract.rb –style="Footer"  pelda1.sxw
Itt már a formázáson is változtattunk
```

Az OOoExtracttel tehát nemcsak a szöveg tartalmában kereshetünk, de a megadott formázási stílushoz tartozó szövegrészleteket is előbányászhatjuk. Így az OOoExtract akár a tartalomra és a jelentésre épülő kereséseket is végezhet, ezáltal a Writer dokumentumból egyszerű adatbázist hozva létre.
A programot egyszerre több fájlon is lefuttathatjuk, ha a fájlnévben helyettesítő karaktereket használunk. Tegyük fel, hogy Writer fájlokban recepteket tárolunk. Ha a szövegeket egyéni stílusokkal készítettük el, akkor például megkereshetjük, hogy melyik receptben szerepel az alma mint hozzávaló:

```
./ooo_extract.rb –text="alma" –style="Hozzávaló"
recipes/*.sxw
AlmasSalsa.sxw: 2 közepes piros alma
AlmasRetes.sxw: 4 csésze hámozott és szeletelt alma
```

Az SXW fájlformátum

Hogyan képes az OOoExtract erre a varázslatra? A titok a fájlformátumban rejlik. Bár minden Writer fájl kiterjesztése .sxw, a file nevű UNIX parancsot alkalmazva kiderül, hogy zip fájlról van szó:

```
$ file pelda1.sxw
pelda1.sxw: Zip archive data, at least v2.0 to
➥extract

És mi található benne? Lássuk:
$ unzip -l pelda1.sxw
Archive:  pelda1.sxw
  Length     Date   Time    Name
 --------  -------- -----
       30  11-26-03 01:40   mimetype
     2328  11-26-03 01:40   content.xml
     8358  11-26-03 01:40   styles.xml
     1159  11-26-03 01:40   meta.xml
     7021  11-26-03 01:40   settings.xml
      752  11-26-03 01:40   META-INF/manifest.xml
 --------                   ----
    19648                   6 files
```

Az OOo XML formátumában minden tartalom és metaadat egyszerű szövegként kerül tárolásra, nem kell aggódnunk a titokzatos bináris kódolás vagy az átláthatatlan szerkezetek miatt. Mivel az adatok XML formában vannak tárolva, az OOo fájljait sok más programmal is beolvashatjuk. Az egyszerű szövegként való tárolás természetesen azt is jelenti, hogy a fájllal kapcsolatos adatok magában a fájlban tárolódnak, egyszerűen olvasható formában. Természetesen az OpenOffice.org munkatársai nem hagynak minket magunkra, mert a dokumentációban a fájlformátum részletes leírása is megtalálható. A OpenOffice.org XML 1.0-és felhasználói kézikönyve egy 571 oldalas PDF fájl. Bevallom, hogy nem olvastam el az egészet, de kétlem, hogy a legapróbb részlet is hiányozna belőle.
Példánk kedvéért csupán néhány alapvető szabályt tekintünk át, melyek segítségével felfedhetjük az OOoExtract működését, illetve a fájlformátumot is megismerhetjük.
A példadokumentum kicsomagolása után a ***content.xml*** fájlt töltsük be egy szövegszerkesztőbe. Ekkor észrevehetjük, hogy a fájl nem túl áttekinthetően van formázva. Futtassunk a fájlon egy XML formázó segédprogramot (például tidy-t), amely sorkihagyásokat és behúzásokat helyez el a szövegben.
A fájl egy XML meghatározással kezdődik, melyet egy DOCTYPE hivatkozás követ, közvetlenül utána pedig a gyökérelem áll, az `office:document-content`. A kezdő tag számos XML névtér tulajdonságot tartalmaz, melyekkel nem kell foglalkoznunk, de áttekintve őket már lehet némi elképzelésünk az OOo dokumentumokban található tartalom felépítéséről.
Közvetlenül a gyökérelemben a szkriptek, betűtípus-meghatározások és stílusok gyerekelemeit találjuk. Mivel példánk egyszerű dokumentum, itt nem sok adatot találunk. Számunkra most az `office:body` elem tartalma fontos. Még ebben a fontos fájlban is viszonylag kevés elem fejezi ki bizonyos alkotóelemek, például táblázatok és ábrák hiányát vagy meglétét. A teljes dokumentum hozzáférhető a CD melléklet Magazin/OOo könyvtárban.
A dokumentum tényleges tartalma text:p elemekben található:

```
<text:p text:style-name="Standard">Ez egy
➥példa</text:p>
<text:p text:style-name="Standard">Két sorból
➥áll</text:p>
<text:p text:style-name="Footer"> Itt már
➥a formázáson is változtattunk</text:p>
```

Ha véletlenül nem ismernénk az XML formanyelvét, a fenti sorok a text (szöveg) névtérben meghatározott p (bekezdés) elemek. Az előtag és a kettőspont segítségével hivatkozunk a dokumentum tetején meghatározott névtér URI-ra, így elkerülhetjük a más XML szóhasználatokhoz meghatározott p elemekkel való ütközést. Példánkban ezt egy teljes elemnévként tekintjük.
A példa csak három bekezdést tartalmazott, tehát három p elemre számíthatunk. Mindegyik rendelkezik egy text:stílusnév tulajdonsággal, mely a bekezdés szövegére vontakozó stílust jelöli meg. Az OOoExtract ezen tulajdonság segítségével találja meg a keresett stílussal írt szövegrészleteket.
Vajon mi lehet a Footer (lapalji jegyzet) stílusban?
A stílusmeghatározást nem a content.xml fájl tartalmazza, s ez az elkülönítés jó dolog. Mennyivel kényelmetlenebb lenne, ha egy egyszerű név helyett minden esetben a stílus teljes leírása (betűméret, szín stb.) állna itt. Így elvesztenénk a tartalom alapján történő keresés lehetőségét, s az adatokkal csak a leképezés szintjén foglalkozhatnánk. Aki kíváncsi a Footer meghatározására, az kukkantson bele a styles.xml fájlba. Ebből kiderül, hogy a Footer a Standard stílusból épül fel, az alapstílus némi módosításával.

Zipből REXML

Valóban nagyszerű, hogy az OpenOffice.org tömörített XML fájlokat használ, de mi következik azután, hogy a csomagot kibontottuk? Szerencsére a Ruby 1.8 tartalmaz egy remek XML olvasót. A REXML egy, az XML 1.0-nak megfelelő olvasó, mely a Rubyra emlékeztető API-ja mellett az XPath és a SAX2 teljes megvalósítását is magában foglalja. Fejlesztője ***Sean Chittenden***, aki bevallása szerint azért írta, mert akkoriban csak a Rubyban csupán két módszer létezett az XML fájlok olvasására. Az egyik a natív C olvasóhoz való kapcsolódás, melynek hátránya a hordozhatóság elvesztése. A másik, hogy tisztán csak a Rubyt használjuk, Sean szerint azonban a Ruby ehhez nem rendelkezett megfelelő API-val. Sean jól ismerte a különféle Java-alapú XML olvasókat, azonban ezek a W3C-féle DOM-hoz vagy a közösségi SAX-hoz igazodnak, s ez nem tesztett neki. Az Electric XML feljesztői egy olyan, a Java szókincsére épülő API-t tettek elérhetővé, melyhez a Java programozók hamar hozzászokhatnak.
Ugyanez az elv állt az REXML (Ruby Electric XML) API fejlesztése mögött is . Nem meglepő módon a REXML API az eredeti Java-stílustól fokozatosan eljutott a Ruby-szerű felépítéshez, így a fejlesztők a Rubyból ismert formanyelv és programozási fogások (például a blokkok és beépített eszközök) használatával képesek XML fájlokat kezelni.

A REXML API

A REXML faolvasó segítségével egyszerűen betölthetjük az XML dokumentumokat:

```
require "rexml/document"
file = File.new( "xml_fájl.xml" )
```

```
doc = REXML::Document.new file
vagy:
require "rexml/document"
my_xml_string = "<sample>
   <text>Ez a mi REXML dokumentumunk</text>
   </sample>"
doc = REXML::Document.new my_xml_string
```

A Document konstruktor karakterláncot vagy I/O objektumot fogad. A REXML felismeri, hogy melyik esetről van szó és eszerint cselekszik. Ha megvan a kérdéses dokumentum, az elemek megtalálásához a Ruby tömbjeit és az `each` kulcsszót XPath választóval kell használnunk:

```
my_xpath = "sample/text"
doc.elements.each( my_xpath ){
    |el| puts el.text }
```

A fenti példában az each metódus az XPath választóra illeszkedő összes elemen végighalad. Minden egyes meghívásnál egy kódblokk (a kapcsos zárójelek közötti rész) kerül meghívásra. Az el változó az éppen vizsgált elemet jelenti, tehát a példa az XPath-ra illeszkedő elemekhez tartozó szöveget jeleníti meg.

XPath

Writer példaszövegünk és a hozzá tartozó XML fájl meglehetősen egyszerű, így a megfelelő stílushoz tartozó elemek megtalálása sem túl bonyolult feladat. Egy ilyen egyszerű példa leginkább a jelen cikkhez hasonló szövegekhez használható, de a valóságban kevéssé valószínű, hogy ilyennel találkozunk: lehet, hogy a leíró rendszert csupán részleteiben ismerjük. Az ilyen fájlokban való keresés már nagyobb kihívást jelent, de az XPath minden gondunkat megoldja. Az XPath a W3C javaslata az XML dokumentumok részeinek eléréséhez. Segítségével egy útvonalmeghatározást hozhatunk létre, mely a helymeghatározást az elemek, tulajdonságok neve, tartalma, illetve relatív vagy abszolút helyzete alapján végzi el. Összetett XML dokumentum esetén például az alábbi sorral egy XPath kifejezést hozhatunk létre, mely az összes olyan `text:p` elemet megtalálja, mely az `office:body` elem közvetlen leszármazottja:

```
*/office:body/text:p
```

XPath nyelven a kezdő * karakter azt jelenti, hogy kövessünk az XML dokumentumban minden olyan `text:p` elemekhez vezető útvonalat, ahol a `text:p` elem az `office:body` elem gyermeke. REXML-ben pedig az alábbi XPath segítségével válogathatunk az illeszkedő elemek között:

```
xml.each_element( */office:body/text:p" ) do |el|
   # Az el változóval csinálunk valamit, például
   # valamilyen tartalom vagy stílustulajdonság
után kutatunk
end
```

A fenti példában a do és az end között egy blokk szerepel. Névtelen függvényre hasonlít, mely a csoport minden egyes eleménél (jelen esetben az XPath-ra illeszkedő elemeknél) meghívásra kerül. Az elem pedig paraméterként kerül átadásra, ezt jelzi a do után álló két függőleges vonal is. Lényegében így működik az OOoExtract, de a számos parancssori kapcsoló működéséről még többet olvashatunk a program honlapján.

A cél egy általános OOo API

Az OOoExtract megismerése után rájöttem, hogy a Rubyhoz egy általánosabb célú OOo objektumra van szükségem. Az OOoExtract mögött álló alapelvek nem csak az adatok olvasását teszik lehetővé, hanem segítségükkel például az adatbáziseszközökből ismert és kedvelt CRUD műveletek létrehozása, frissítése és törlése is megvalósítható. E célból jött létre az OOo4R fejlesztés a RubyForge-on, a Ruby szoftveres CVS gyűjteményében. A cél az adatok és metaadatok egyszerű elérése, az XPath rugalmas, észrevétlen használata és egy API létrehozása a közhelynek számító műveletek elvégzésére. Helyhiány most csak a metaadatok elérésével foglalkozunk részletesebben, hiszen itt tulajdonképpen arról van szó, hogy a Ruby dinamikus üzenetkezelésének segítségével keresünk az elemek tartalmában.
Korábban láthattuk, hogy egy OOo dokumentum több XML fájlt tartalmaz, melyek egy zip fájlban kerülnek tárolásra. A ***content.xml*** fájllal már foglalkoztunk, egy másik pedig a ***meta.xml*** névre hallgat. Ez utóbbi magával a dokumentummal kapcsolatos adatokat tartalmaz, például a címét, a létrehozás időpontját és a szavak számát. A gyökérelem az ***office:document-meta***, ezen belül helyezkedik el az `office:meta` elem, mely az értékes adatokat tartalmazó származékelemek szülője:

```
<meta:initial-creator>James Britt
</meta:initial-creator>
<meta:creation-date>2003-11-25T17:36:31
</meta:creation-date>
<dc:creator>James Britt</dc:creator>
<dc:date>2003-11-25T18:40:59</dc:date>
<dc:language>en-US</dc:language>
<meta:editing-cycles>13</meta:editing-cycles>
```

A teljes metaadatfájl megtalálható a CD melléklet Magazin/OOo könyvtárában.
A fő Document osztály mellett az OOo4R a metaadatokat magában foglaló meta osztályt is meghatároz. Egy meta osztály egy REXML dokumentum használatával tárolja a ***meta.xml*** fájl tartalmát. Egy meta objektum jórészt tulajdonságok gyűjteménye. Általában arra van szükségünk, hogy lekérdezünk bizonyos adatokat, például a szerző nevét, vagy beállítunk egy új értéket, például új címet adunk meg. Minden tulajdonsághoz két metódus szükséges, de használhatunk dinamikus metódusmeghívást is. Ez utóbbi azt jelenti, hogy elfogjuk az elérési kérelmeket, találunk rájuk illeszkedő tulajdonságneveket és végrehajtjuk a kért műveletet vagy megszakítást kezdeményezünk.
Az alábbi kódpélda az OOo-ban használt Dublin Core metaadat-elemekkel foglalkozik. A Dublin Core Metaadat Kezdeményezés (Dublin Core Metadata Initiative) a metaadatokkal kapcsolatos szabványalkotás nyílt fóruma. Dublin Core elemeket gyakran találunk RSS adatokban és néhány XHTML dokumentumban is előfordulnak. Az

OpenOffice.org XML fájljaiban lévő elemekhez hasonlóan ezek is rendelkeznek névtér-előtaggal. Ezeket az előtagokat nem kell megjegyeznünk, a teljes elemnevet hozzárendelhetjük valami egyszerűbben megjegyezhető névhez.
A Meta osztály meghatározása egy hash létrehozásával kezdődik, mely a valódi elemnevekhez barátságosabb neveket, illetve a metaadatok alap XPath-jét tartalmazó osztálykonstanst rendel. Az osztály alkotófüggvénye (konstruktora) egyszerűen REXML dokumentumot készít az XML forrásból:

```
module OOo
  class Meta

  NAME_MAP = {
   ´description´ => ´dc:description´,
   ´subject´     => ´dc:subject´,
   ´creator´     => ´dc:creator´,
   ´author ´     => ´dc:creator´,
   ´date´        => ´dc:date´,
   ´language´    => ´dc:language´,
   ´title´       => ´dc:title´
  }
    XPATH_BASE  = “*/office:meta”

    def initialize( src )
      @doc = REXML::Document.new(  src.to_s )
    end
```

A `method_missing` nevű metódust újra meghatározhatjuk úgy, hogy minden Ruby osztály számára elérhető legyen. Ilyenkor nem megszakítás jön létre, hanem a metódus megnézi, hogy az objektumhoz küldött üzenet illeszkedik-e a metadaat valamely elemére:

```
def method_missing( name, *args )
  n = name.to_s
  if is_assignment? n
    el = map_for_assignment n
    xpath = ”#{XPATH_BASE}/#{el}”
    assign( xpath, *args)
  else
    el = Meta.map_name n
    xpath = ”#{XPATH_BASE}/#{el}”
    find( xpath  )
  end
end
```

A `method_missing` első paramétere egy szimbólumobjektum, tehát a kód a helyettesítő karakterláncot olvassa be. Az `is_assignment` metódus ellenőrzi, hogy a név egyenlőségjelre végződik-e. Ha igen, akkor hozzárendelésról van szó és ekkor a `map_for_assigment` a metaadat neve után álló minden jelet töröl, a barátságosabb nevet pedig a Dublin Core elemnévhez kapcsolja. Az `assign` a REXML dokumentumban frissíti a megfelelő elemet:

```
def assign( xpath, val )
  node = @doc.elements.to_a( xpath )[0]
  node.text = val
end
```

Ha nem hozzárendelésről van szó, a kód megpróbálkozik a metaadatok beolvasásával. Akárcsak előzőleg, a név itt hozzárendelésre kerül, de most a kód a `find`-ot hívja meg:

```
def find( xpath )
 begin
  return @doc.elements.to_a( xpath.to_s )[0].text
 rescue Exception
  raise OOo::OOoException.new(
      “Error with xpath ´#{xpath}´: #{$!}”, $@ )
 end
end

# Kihagyott segítőmetódusok...

 end
end
```

E módszerrel a többi metaadat-elemet is elérhetjük, bár van néhány különleges eset, ahol a metaadatokat egy csapat származékelem tartalmazza. A módosított OOo dokumentumok mentéséhez a Ruby beépített Zip osztályának segítségével frissítsük a zip fájl tartalmát és írjuk vissza a lemezre.

Összefoglalás

Mivel az OpenOffice.org fájlformátum egy teljesen dokumentált XML formátumot használ, OOo fájlokat az OOo nélkül is létrehozhatunk és módosíthatunk. A Ruby beépített dinamikus XML kezelése tökéletesen alkalmassá teszi a programot az OOo dokumentumok kezelésére.

Linux Journal 2004. március, 119. szám

James Britt az arizonai Scottsdale-ben működő, szoftverfejlesztéssel és tervezéssel foglalkozó vállalat, a Neurogami vezetője. Részt vett egy XML-ről szóló, a Wrox Press által kiadott könyv megírásában, emellett számtalan szoftverfejlesztéssel foglalkozó cikk származik tőle, valamint a Rubyról és XML-ről előadást tartott a texasi Austinban rendezett 3. Nemzetközi Ruby Konferencián.
A jamesgb@neurogami.com címen érhető el.

FORRÁSOK

Dublin Core: dublincore.org
OOo_extract:
www.math.umd.edu/~dcarrera/openoffice/tools/ooo_extract.html
OOo formátumok:
xml.coverpages.org/starOfficeXML.html
OpenOffice.org XML: xml.openoffice.org
Ruby:
linux.oreillynet.com/pub/a/linux/2001/11/29/ruby.html
RubyForge: www.rubyforge.org
“Gondolkodj XML-ben: az Open Office fájlformátuma”:
www-106.ibm.com/developerworks/xml/library/x-think15
XPath: www.w3.org/TR/xpath

Biztonságos PHP a gyakorlatban

Ahhoz, hogy meg tudjuk védeni PHP alkalmazásainkat, az első és legfontosabb lépés a biztonsági veszélyek felismerése és megértése.

Az elmúlt két évben a PHP magfejlesztői rendkívüli munkát végeztek: a PHP felhasználói közösséget ellátták egy olyan hatékony technológiával, amely figyelemre méltóan jól teljesít változatos környezetben. Ahogy a webalkalmazások egyre népszerűbbek lettek, a webfejlesztőknek muszáj volt szembenézni a lehetséges biztonsági rések növekvő számával, melyek komolyan veszélyeztethetik a munkájukat. Sok útmutató, könyv és cikk látott napvilágot, ahogy az új technikákat fejlesztették. Ezek az újonnan felmerülő veszélyek azonban mégsem kapnak annyi figyelmet, amennyit megérdemelnek.

Jelen cikk azoknak a profi és nyílt-forráskódú PHP fejlesztőknek szól, akik magas szintű biztonságot szeretnének nyújtani felhasználóiknak illetve ügyfeleiknek. Az írásnak nem célja, hogy a programozók minden egyes kérdésére választ adjon. Inkább abban próbál segíteni, hogy az alkalmazás lehetséges problémái már a tervezési folyamat során kiderüljenek. Hosszú távon ugyanis ez teszi lehetővé nekünk, PHP fejlesztőknek, hogy az új biztonsági fenyegetésekre megfelelően reagáljunk.

Számos cikk megjelent már a biztonságos PHP fejlesztésről, s valamennyi nagyjából azonos témákat tárgyal. A következőkben ugyan gyorsan átvesszük a fontosabb alapfogalmakat, de feltételezem, hogy a legtöbben már jól ismerik ezket, ezért nem fogok sok időt tölteni velük.

register_globals

A PHP nyelvben használhatjuk a `register_globals` beállítási változót, melynek engedélyezésével az alkalmazás valamennyi változója globálissá válik. Ez annyit tesz, hogy a Webkiszolgálónak eljuttatott POST, GET, süti és session változók ugyanabba a kalapba kerülnek, ami fejlesztői szemmel nézve meglehetősen kényelmes módja a változók kezelésének.

Tervezési szempontból e beállítás engedélyezése valószínűleg alkalmazásunk egészének biztonsági szintjére kedvezőtlen hatással lesz, hiszen a felhasználók közvetlen eléréssel rendelkeznek majd az alkalmazásunkban használt valamennyi változóhoz. A PHP-t ma már alapértelmezés szerint kikapcsolt `register_globals` változóval kapjuk, és azt javaslom, a biztonság kedvéért hagyjuk is így. Ez alól csak az jelenthet kivételt, ha gépünkön régről ott maradt alkalmazások futnak, amelyek megkövetelik e változó engedélyezését.

Cross-site scripting

A népszerű Cross-site scripting, azaz XSS technika segítségével a felhasználó átveheti az irányítást a megjelenés, a tartalom, vagy akár a Web alkalmazás teljes biztonsági rendszere felett. Nem a PHP az egyetlen nyelv, amely sebezhető e módszerrel, hiszen nem nyelvi hibáról van szó, hanem sokkal inkább a webalkalmazás tervezésének bizonyos hiányosságairól.

A Cross-site scripting sok formában létezik. Az egyik népszerű megoldás, amikor HTML vagy JavaScript kódot helyezünk az űrlapok mezőibe, így kényszerítve az alkalmazást valami olyan megjelenítésére, amit egyébként nem tenne. Ez az eljárás ékesen bizonyítja, mennyire fontos, hogy minden bemenetet megszűrjünk érkezzen az a felhasználótól, másik honlaptól, vagy adatbázisból. A `htmlentities()` PHP függvény általában jó megoldás az ilyesfajta támadások elhárítására.

GET változók

A legtöbb Webalkalmazás számára nagyon fontos, hogy olyan URL-t tudjunk nyújtani a felhasználóinknak, melyet később felhasználhatnak ha vissza szeretnének térni oda ahol éppen állnak. Fejlesztőként azonban nem árt ha meg tudjuk állapítani, mely információkhoz tud a felhasználó hozzáférni a lehetséges módok bármelyikével.

A lekérdezés szöveg módosításával a felhasználó módosítani tudja az alkalmazásunkban használt változók tartalmát. Az ilyesfajta próbálkozások megakadályozása már bonyolultabb annál, mintsem hogy egyszerű bemenetszűréssel megoldható legyen, de ez a követendő irány. Az ilyen támadások ellen talán a legjobb módszer, ha alkalmazásunkhoz hibatűrő adatfolyam-kezelést és jól felépített hibakezelő rendszert készítünk.

SQL beszúrás

Ennek a webalkalmazások ellen indított rosszindulatú támadásnak olyan pusztító következményei lehetnek, melyek messze túlszárnyalva az egyéb támadási formák (például a cross-site scripting) által okozott károkat, hiszen ezzel a módszerrel akár a teljes adatbázisunkat elveszíthetjük.

Az SQL beszúrás elve nagyon egyszerű. A legtöbb Webalkalmazás sütikből, POST és GET változókból várja a bemenő paramétereket. Ezeket gyakran használjuk SQL lekérde-

1. lista PHP alatti SQL lekérdezés készítése POST változók alapján

```
<?php
$query = “SELECT id, name FROM `records` LIMIT “
         . $_POST[´NUM´];
$result = $db->select($query);
?>
```

2. lista Rosszindulatú űrlap, amit SQL beszúrásos támadáshoz használhatnak fel.

```
<form action=”example.com/form.php”
method=”POST”>
<input type=”text” name=”NUM”
       value=”5; DELETE FROM `records`”>
<input type=”submit”>
</form>
```

3. lista Egyszerű „Ártalmas SQL parancsok” szűrő

```
<?php
function filter_sql($input) {
    $reg = “(delete)|(update)|(union)|(insert)”;
    return(eregi_replace($reg, “”, $input));
}
?>
```

zések paramétereként, ezzel biztosítva a felhasználónak dinamikus tartalmat. Amennyiben a felhasználónak van valami elképzelése arról, hogy hogyan épülhet fel az adatbázisunk, elvileg képes lehet úgy módosítani a paramétereinket, hogy azzal SQL parancsokat adjon ki a lekérdezésünkön belül.
Nézzünk egy gyors példát. Az alkalmazásunk POST módszerrel vár adatokat egy űrlaptól. A célunk legyen az adatbázis x bejegyzésének megjelenítése, ahol a felhasználó tetszése szerint maga állíthatja be x értékét. Ezért az űrlapunk tartalmaz egy NUM mezőt, amely aztán a parancsfájlunkhoz továbbítja a beírt értéket. Az *1. lista* mutatja be a folyamatot. Ebben az esetben a felhasználó hamisíthat egy HTML űrlapot, amely a táblánkat teljesen kiürítő, gondosan megtervezett adatot küld el.
Ha a felhasználó úgy dönt, hogy a *2. listában* bemutatott űrlaphoz hasonlót használ, az eredmény a következőképpen néz majd ki:

```
SELECT id, name FROM `records` LIMIT 5;
DELETE FROM `records`
```

Nyilvánvalón igen egyszerű kivédeni az ilyen támadásfajtákat, mégis úgy vettem észre, rengeteg alkalmazás egyáltalán nem rendelkezik a megfelelő védelemmel.

Jelen esetben ugyan az SQL beszúrás ellen meglehetősen jó védelmet nyújtana ha az `intval()` függvénnyel egész számmá alakítanánk a NUM értékét, de fontos látnunk, hogy a fejlesztők nem gondolhatnak végig minden egyes változót minden egyes SQL lekérdezésben. Ezért a legjobb, ha alkalmazásunkban automatizáljuk ezt a műveletet. Minthogy a modern web alapú alkalmazások egyre gyakrabban használnak magmodult vagy valamilyen központi kapcsolótábla rendszert, az ilyen megoldások alkalmazás szintű megvalósítása is leegyszerűsödik. Az alkalmazásunkban használható automatizált (streamlined) képességekkel a cikk későbbi részében foglalkozunk. Addig is felsorolunk néhány ötletet, ami segíthet felépíteni a saját megoldásunkat:

1. Az SQL parancsokhoz használjunk szkifeket (szabályos kifejezéseket): ez a módszer akkor az igazi, ha nem várunk szöveges adatokat a felhasználótól, de az SQL kulcsszavak kiszűrésével általában jó eredményt érhetünk el (*3. lista*).
2. Használjunk Ellenőrző kódokat (assertion): Ezzel a megoldással később foglalkozunk majd.
3. Védett szövegek: amennyiben várhatóan nem kapunk bináris adatokat, a bemenet biztonsága érdekében érdemes védeni a szövegeket (escaping). A fenti példánkban ugyan a szövegvédelem nem sokat segített volna, de sok SQL beszúrásos támadás alapul az SQL lekérdezés megszakításán és új lekérdezés beszúrásán. Az ilyen támadásokat hatékonyan megelőzhetjük például a `mysql_escape_string()` függvénnyel .

Titkosítás

Az adatbázis kiszolgálókban és más tárhelyeken gyakran tárolunk sebezhető adatokat. Ilyen esetekben nekünk, fejlesztőknek, rendkívül fontos, hogy legyen valamilyen módszerünk az adatok biztonságos tárolására és igény szerinti könnyű előkereshetőségére.
A PHP tartalmaz egy bővítményt, melynek segítségével kihasználhatjuk az Mcrypt Könyvtár (***mcrypt.sf.net***) képességeit adataink titkosítására illetve későbbi visszakódolására. A Mcrypt PHP bővítmény dokumentációja a ➲ http:\\www.php.net/mcrypt lapon olvasható, amit alkalmazás előtt nem árt gondosan áttanulmányozni.
A bővítmény lenyűgöző mennyiségű algoritmust ismer, kezeli többek között a triple-DES-t, a Blowfisht, a Twofisht és a Two-Wayt. Hacsak nem értünk a titkosításokhoz, a Mcrypt bővítmény alkalmazása nem teljesen magától értetődő; a számtalan blokkalgoritmus és titkosítási módszer elsőre elég zavaró lehet. A *4. listában* megtekinthetjük, milyen módszereket kínál a Mcrypt bővítmény valamint hogyan használhatjuk azokat.

Ellenőrző kódok (assertions)

Ez a funkció lehetővé teszi a PHP fejlesztőnek, hogy az alkalmazást hibakezeléssel lássa el, és megőrizze az adatok integritását. Ez nem kifejezetten biztonsági jellegű funkció, ráadásul a PHP-ben több élvonalbeli nyelvnél is megtalálható (C, Python), miért is hozom fel akkor? Dióhéjban: azért, mert ha ügyfeleinknek vagy felhasználóinknak biztonságos alkalmazásokat szeretnénk nyújtani, az első lépés a hibakezelés.

4. lista Tipikus példa a Mcrypt bővítmény használatára

```
<?php
/* készítsünk véletlenszerű kulcsot
   de tartsuk kéznél, hiszen később
   ezt használjuk a visszakódoláshoz
*/
$key = "AOQKJLCLIGAKJHSD
        <NKLXASLUIHJKHAS
        OIUDSgfuyJKLBLKU";

$string = $_POST[´password´];

/* Először is meg kell nyitnunk a Mcrypt titkosí-
tó modulját*/
$mod = mcrypt_module_open
(´blowfish´,´´,´ecb´,´´);

/* Ezután létre kell hoznunk egy Initialization
   Vector-t
   a méret és a forrás alapján.
   A forrásunk tetszőleges lehet, illetve hasz
   nálhatunk előre definiált állandókat.
   A vektor méretének megadása a felhasznált
   modultól függ*/

$iv_size = mcrypt_enc_get_iv_size($mod);

/* Az alaphelyzetbe állító vektor a példánkban a
   /dev/random forrásból származó $size karakte
   rein alapszik  */

$iv =
mcrypt_create_iv($iv_size,MCRYPT_DEV_RANDOM);

/* A következő lépésként ellenőrizzük, hogy
   a kulcsunk nem túl nagy-e
   és csonkoljuk ha szükséges*/
$max_key_size = mcrypt_enc_get_key_size($mod);
$key = substr($key,0,$max_key_size);

/* Alaphelyzetbe állítjuk a titkosító algoritmust
   a mcrypt_generic_init segítségével*/
mcrypt_generic_init ($mod,$key,$iv);

/* Az adatainkat ez után már titkosíthatjuk a
   mcrypt_generic függvénnyel. A függvény
   visszatérési értéke a titkosított adat lesz*/
$encrypted = mcrypt_generic($mod,$string);

/* A Mcrypt használatának befejezése után
   fel kell szabadítanunk a folyamatban
   felhasznált vermeket*/
mcrypt_generic_deinit ($mod);

/* Végül, le kell zárnunk a felhasznált titkosító
   modult*/
mcrypt_module_close ($mod);

/* Most nézzük, hogyan kódoljuk vissza az adatot:
*/
$padded = // lásd a következő sort
mcrypt_decrypt(´blowfish´,$key,$encrypted,´ecb´,$
iv);

/* Most a visszakódolt szövegünk nullával tagolt,
   ezért el kell távolítanunk a felesleges \0-kat
*/
$plain = str_replace("\0","",$padded);
echo "Titkosított szöveg: $encrypted<br>";
echo "Visszakódolt szöveg: $plain<br>";
?>
```

A Assertions PHP alatt két függvény, az `assert_options()` és az `assert()`, segítségével valósítható meg. Az elsőt alkalmazásunk alaphelyzetbe állításakor, illetve beállítás állományában hívjuk meg, a másodikat bárhol meghívhatjuk a kódban, ahol a bemenet helyességét kell kikényszerítenünk. Az *5. lista* azt mutatja be miképpen készíthetünk hibakezelő rendszert, ami egyszerű jelentést készít, ha az ellenőrzés sikertelen.
A „PHPUnit Project" egy teljes körű elemtesztelő készlet, amely ingyenesen hozzáférhető a PHP fejlesztők számára, és pontosan arra alkalmas amit mi most összeraktunk.
A honlapot a ➲ http:\\www.phpunit.sf.net címen találjuk.

Adatfolyam

Amennyiben több különféle webprojekten is dolgoztunk, az új fejlesztéseinkhez valószínűleg van már valamiféle közös váz, amit használni kezdtünk, netán kifejlesztettük saját változatunkat. Az adatkezelés központosítására sokféle módszer létezik és a projektünket leíró követelmények függvényében bizonyos modellek nyilván jobban megfelelnek mint mások. A következő néhány bekezdésben bemutatok egy egyszerű tervezési sablont amely a fejlesztőknek üzleti célú projektekhez is elegendő rugalmasságot és méretezhetőséget biztosít.
Első lépésként ki kell dolgoznunk egy olyan módszert, amellyel minden bemenő adatunkat központosíthatjuk, és egységes szűrőfelületen küldhetjük keresztül. Ezáltal a további módosításokat egyszerűen moduláris jelleggel végezhetjük majd. Példánkban következő fájlstruktúrát használjuk:

- */**index.php***: az egyetlen állomány a gyökérben.
- */**lib***: könyvtárakm amelyeket .htaccess véd.
- */**lib**/**config.inc.php***: beállítás állomány.
- */**tpl***: sablonok, szintén a .htaccess védelme alatt.
- */**doc***: projektek és API dokumentációk.
- */**images***: képek.
- */**classes***: osztályok, .htaccess védelemmel.

Amint az a *2. ábrán* látható, alkalmazásunk magját az ***index.php*** állomány alkotja, melynek közvetlen elérése van

5. lista Hibajelentés Assertion segítségével

```
<?php

/* A teljes alkalmazásunkban ki-be
   kapcsolhatjuk az
   assertion képességet az ASSERT_ACTIVE változó
   1 vagy 0
   értékre állításával.
*/
assert_options(ASSERT_ACTIVE,1);

/* Azt szeretnénk, ha az alkalmazásunk kilépne
   ha az assertion sikertelen. (Legalábbis ebben
   a példában)
*/
assert_options(ASSERT_BAIL,1);

/* Példánkban magunk végezzük a hibajelentést
   ezért kikapcsoljuk az alapértelmezett
   figyelmeztetéseket.
*/
assert_options(ASSERT_WARNING,0);

/* Az sikertelen assertion ellenőrzéskor
   meghívandó
   saját függvényünk neve „display_error" lesz.
*/
assert_options(ASSERT_CALLBACK, "display_error");

$email = strtolower($_POST[´email´]);
$parts = array();

// elkészítjük a szabályos kifejezésünket
$regex = "^([.\´a-z0-9]+)@([.\´a-z0-9]+)$";

/* Helyes formátum ellenőrzése, és az email
   cím ellenőrzése egy időben történik.
   Figyeljük meg a különleges formátumot.
   Minden idézőjelek közt található, a hibát
   pedig
   megjegyzésbe tettük. Ezt a hibát később,
   szabályos
   kifejezéssel fogjuk kigyűjteni.
*/
assert("ereg(\$regex, \$email, \$parts); /*
        hibás e-mail cím: $email */");

/* Ez a rész nem hajtódik végre, ha az
   assertation
   sikertelen így biztonságosan továbbléphetünk
*/
$username = $parts[1];
echo "Üdv otthon, " . $username;

// Most következik az ASSERT_CALLBACK függvényünk
function display_error($file, $line, $error) {

     // Ez a rész gyűjti ki a hibaüzenetet
     $regex = "(.*)/\* (.*)\*/";
     $parts = array();
     ereg($regex, $error, $parts);
     $msg = $parts[2];

     // csinos kimenetet készíthetünk
     echo "
     <table bgcolor=\"#bbbbee\">
     <tr><td colspan=´2´ align=´center´>
     <b>Hibajelentést</b>
     </td></tr>
     <tr><td>Állomány:</td><td>$file</td></tr>
     <tr><td>Sor:</td><td>$line</td></tr>
     <tr><td>Üzenet:</td><td>$msg</td></tr>
     ";

}

?>
```

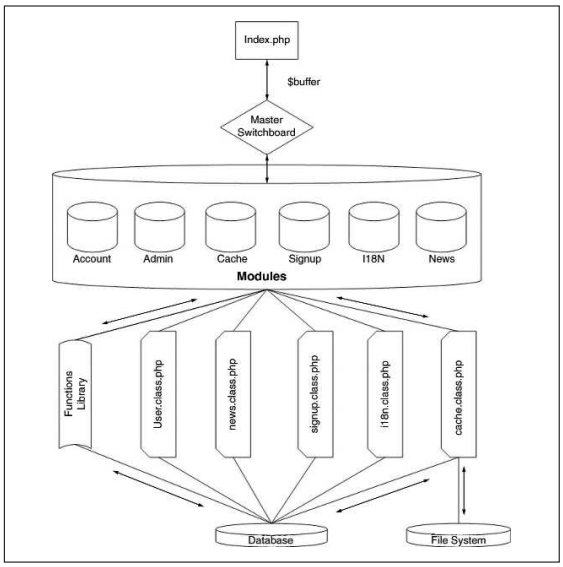
a `template`, `class` vagy `configuration` állományok bármelyikéhez; a felhasználó azonban soha nem érheti el ezeket az állományokat.
Menjünk végig lépésről lépésre a 2. ábrán látható vázlaton. Vegyük példaként azt a folyamatot, ahogy a felhasználó bejelentkezik a rendszerbe.

1. A felhasználó paraméter nélkül kéri le az ***index.php*** állományt. Az index létrehoz egy vermet, majd továbbadja a kapcsolótáblának, amely meghívja az alapértelmezett modult. A modul sablon segítségével megjeleníti az alkalmazás alapértelmezett bejelentkező képernyőjét.
2. A felhasználó kitölti, majd elküldi az azonosítási űrlapot. Az űrlap kimenetét valami ilyesmi címre irányítja: `?module=account&action=login`. A kapcsolótábla meghívja a login függvényt, amely nem más mint a user osztály elérési felülete. A függvény létrehoz egy példányt a user osztályból. Ez az objektum szolgál csatolófelületként a felhasználónk és az adatbázis között, valamint ez végzi el a lekérdezést.
3. Az adatbázis visszaküldi az adatokat az objektumnak, majd az objektumból a modulhoz kerül, amely beállítja a megfelelő session változókat, meghívja a megfelelő sablont, és segítségével módosítja a vermet. Végül a válaszüzenetet elküldi az index-nek.

A bemutatott modell adatfolyam-kezelése első látásra talán kicsit bonyolultnak tűnhet, valójában azonban meglehetősen egyszerű. A felhasználói bemenet gyorsan a megfelelő modulhoz kerül, a hibakezelést a kapcsolótábla szintjén végezzük. Az egyéb bemenetek – mint például az adatbázis-adatok és fájlrendszer-elérések – szűrését a nekik megfelelő osztályok végzik. Minden osztály egy

Error Report
File: /var/www/localhost/htdocs/assert1.php
Line: 38
Message: Invalid email address: test@

1. ábra Az 5. lista által készített mintaüzenet



2. ábra Alkalmazás mag

különleges vázosztályt bővít amely biztosítja a megfelelő bemenetszűrési képességeket, így egyik osztálynak sem kell ilyesmi miatt aggódnia.
A modell igen hatékony, hiszen méretezhető és hibatűrő szerkezetre épül, de ne feledjük, hogy más, érdekes modellek is léteznek. Vethetünk például egy pillantást a Phrame Project weblapjára (➲ http:\\www.phrame.sf.net) , amely az MVC rendszeren (➲ http:\\www.ootips.org/mvc-pattern.html) alapuló Model2 megközelítést alkalmazza.

Biztonsági mód

A PHP biztonsági módja (Safe mode) olyasmi amit mindenképpen érdemes megtanulni minden PHP fejlesztőnek és rendszergazdának. A biztonsági mód olyan beállítások gyűjteménye, melyek segítségével a rendszergazda befolyásolni tudja a PHP-értelmező működését és biztonsági alapelveket határozhat meg. Rendszergazdai szemmel nézve ez annyit jelent, hogy meg kell tanulnunk, hogyan kell helyesen beállítanunk a rendszert úgy, hogy közben ne tegyük lehetetlenné fejlesztőinknek az alkalmazásaik elindítását a kiszolgálónkon. Fejlesztői szemmel nézve azt kell megtanulnunk, milyen hatással lehet egy adott képesség az alkalmazásunkra, ha történetesen be van kapcsolva.
A `safe_mode` beállítások bekapcsolása indokolt lehet, ha PHP alkalmazásokat futtató megosztott kiszolgálókat üzemeltetünk, és a kiszolgálókat használó PHP fejlesztőkben nem bízhatunk meg. A `safe_mode` beindítása ***php.ini*** állományunkban hatékonyan meggátol minden fájlrendszer elérést, kivéve, ha a fájl tulajdonosának UID azonosítója nem azonos a futó parancsfájl UID értékével. A PHP `safe_mode` alatt is lehetőséget ad e funkció átállítására, ehhez a `safe_mode_gid` kapcsolót kell engedélyeznünk. Ilyenkor, a PHP az állományok GID értékét vizsgálja az UID érték helyett.
Jó gyakorlat, ha felhasználóinknak nem engedélyezzük a rendszer bármely bináris állományának végrehajtását; ilyenkor jöhet szóba a `safe_mode_exec_dir`. Ennek a felbecsülhetetlen értékű funkciónak a segítségével utasíthatjuk a PHP-t, hogy csak akkor hajtson végére bármilyen programot az `exec()` vagy bármely más függvényen keresztül, ha a program megtalálható a `safe_mode_exec_dir` által megjelölt könyvtárban. (Ilyen például az */usr/local/php/bin*).
Miután megismertük a PHP `safe_mode` engedélyezésekor érvénybe lépő korlátozásokat, elkezdhetünk olyan programokat fejleszteni, amelyek nem robbannak le, ha ilyen beállítással rendelkező gépen kell futniuk. Sok ISP használja a `safe_mode` beállításait. A legfontosabb szempontok a következők:

- A fájlműveleteket, legyenek azok írási vagy olvasási műveletek, próbáljuk meg az alkalmazásunkhoz adott állományokra korlátozni.
- Ne használjuk ki, hogy valamilyen külső program már telepítve van vagy végrehajtható, ha a projektünk nem csak a saját kiszolgálóinkon fog futni.

A rendszergazdák ezen felül más hatékony eszközöket is használhatnak rendszer teljes körű biztonságának növelésére. Ide tartozik a `disable_functions` amely megakadályozza, hogy bizonyos függvényeket meghívjunk, valamint az olyan függvények mint az `open_basedir`, amely minden fájlműveletet egy megadott könyvtárra korlátoz.

A PHP dokumentáló csapat által készített irodalom részletesen foglalkozik a témával. Ezen túl készült olyan írás is, amely a `safe_mode`, a hozzá kapcsolódó függvények valamint azok következményeinek részletei taglalja.

Linux Journal 2004. április, 120. szám

Xavier Spriet az utóbbi négy év során PHP alatt fejlesztett programokat. A eliquidMEDIA International vezető fejlesztője. Xavier xavier@wuug.org címen érhető el.

FORRÁSOK

Mcrypt bővítmény: php.net/mcrypt
Mcrypt Project: mcrypt.sf.net
Az MVC minta: ootips.org/mvc-pattern.html
PHP Dokumentáció: php.net/manual/en
PHP Biztonság: php.net/manual/en/security.index.php
A PHPUnit Projekt: phpunit.sf.net
A Phrame Projekt: phrame.sf.net
Safe Mode: php.net/manual/en/features.safe-mode.php

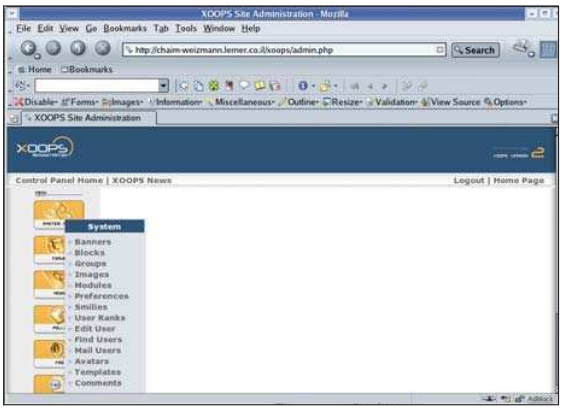
XOOPS

Készítsünk közösségi weblapokat cikksorral, megjegyzéssel, felhasználói blogokkal és egyebekkel felszerelve.

Az elmúlt néhány hónapban egy olyan tartalomkezelő programmal (CMS) ismerkedhettünk meg, amely megkönnyíti az utóbbi időben roppant népszerű blogok készítését. Hogy mi is pontosan a blog, arra jelenleg nem létezik egyértelmű és szigorú meghatározás, de a legtöbb ember valószínűleg egyetért abban, hogy egy olyan formáról van szó, melyben rövid cikkekből áll a tartalom, ahol a legfrissebb jelenik meg legelőször.
Létezik azonban egy nagyobb, kicsit eltérő típus is, amit jobb híján talán Slashdot-stílusnak nevezhetnénk. Ezeket a blogokat ténylegesen kis csoportok szerkesztik, ám van e mellett egy rengeteg tagból álló közösség is, amely részt vesz a napi munkában. Utóbbiak sok szempontból talán közelebb állnak a fórumokhoz és a hirdetőtáblákhoz mint a blogokhoz, de van elég sok naplószerű jellemzőjük is. Miután ugyanis a lap látogatója elolvasta a vezérfonalat adó cikket, elkészítheti saját weblogját, valamint elolvashatja a többi felhasználó bejegyzéseit.
Néhány csomag – közülük is elsősorban a Slash, amely a Slashdot alapjául szolgál, vagy a use.perl.org – kifejezetten a nyilvános és privát weblogok alapötlete köré épült, de más csomagok is felnőttek a feladathoz. Az egyik ilyen csomag az XOOPS, mely név valószínűleg az Extended Object Oriented Portal System (Kiterjesztett objektum-központú portál rendszer) rövidítéséből származik. Az XOOPS a PHPNuke és PostNuke családba tartozó rendszerek oldalhajtása.
Ebben a hónapban az XOOPS rendszerrel ismerkedünk meg, különös figyelmet szentelve a WeBLog (kiejtve: Víbílog) modulnak, amelyet nyilvános és privát blogok létrehozására és felügyeletére terveztek. Vannak ugyan fenntartásaim az XOOPS-al kapcsolatban, bizonyos közösségek számára azonban kiváló eszköz lehet. Különösen két területen nehéz jobbat találni: rendkívül egyszerű telepíteni és karbantartani, továbbá minden felhasználó saját weblogot kaphat.

Történelem

Az XOOPS kiváló példája a GNU General Public Licence hatékonyságának, a nyílt fejlesztési közösségnek, valamint annak, hogy nem feltétlenül káros minden szétválás a nyílt forrású fejlesztésekben. Az XOOPS története a PHP nyelven és MySQL adatbázis háttérrel készülő PHPNuke programmal kezdődött. A PHPNuke akkoriban a PHP nyelven készülő, ThatWare nevezetű Slash klónon alapult, de utóbbinak már felhagytak a fejlesztésével.



1. ábra Az XOOPS honlapgazda oldala, ahol kiválaszthatjuk a lapunkon megjelenő modulokat

Ami ez után történt, annak megítélése – finoman szólva – elég ellentmondásos, és jól példázza, hogy mi történhet akkor, amikor egy nyílt forrású fejlesztés eldurvul: a PHPNuke számos fejlesztője úgy döntött, elszakadnak a projekttől és saját fejlesztésüket PostNuke-nak nevezték el. Manapság a PHPNuke és a PostNuke egyaránt létezik, amivel a felhasználói és fejlesztői közösségek közti kapcsolat szenvedett némi csorbát.
Nos, úgy tűnik, hogy a XOOPS a PHPNuke objektum-központú átirata. Az XOOPS fejlesztők úgy gondolják, hogy a rendszert objektum alapon újraírva megnövekszik a biztonság, és könnyebbé válik a fejlesztés és a karbantartás. A jó hír az, hogy a fent felsorolt rendszerek mindegyike elérhető a GNU General Public License alatt.
Az XOOPS-ot szemmel láthatólag kisebb fejlesztőcsapat készíti, mint a PHPNuke-ot vagy PostNuke-ot. Mégis, a beépített és mások által készített modulok száma igen magas, valamint objektum-központú lehetőségeinek köszönhetően nagyon könnyen készíthetünk saját modulokat. Tulajdonképpen a modulok kis száma ellenére lenyűgöző, mennyire egyszerűen telepíthetőek. Ez azt jelenti, hogy inkább

XOOPS-t használjunk a többi Nuke-szerűség helyett? Nos, ez legalább annyira függ a meggyőződéstől és a tervezett felhasználástól, mint magától a technológiától, bár önmagában a tény, hogy az XOOPS objektumokat használ, tapasztalataim szerint az élvonalba emeli, legalábbis elméletben.

Letöltés és telepítés

Miután áttekintettük a XOOPS és rokonainak őstörténetét, állítsuk be saját XOOPS kiszolgálónkat. Az XOOPS futtatásához szükségünk lesz egy MySQL kiszolgálóra – személy szerint a köztudottan elavult 3.23.58 verziót használom – valamint egy Apache kiszolgálóra (1-es vagy 2-es változat) telepített PHP támogatással.

Az XOOPS hosszú ideje az egyik legfigyelemreméltóbb és legegyszerűbben telepíthető program amivel találkoztam. A kódot az XOOPS weblapról töltöttem le (lásd a hálózati források szakaszt) majd az ideiglenes könyvtárban megnyitottam a ***.tar.gz*** állományt. A teljes HTML alkönyvtárat felmásoltam az Apache tartalom gyökér egyik könyvtárába: ***/usr/local/apache/html/xoops***. Ezt követően az XOOPS telepítőt a ➲ /xoops URL alól érhettem el.

A teljesen web-alapú telepítő lassan de biztososan vezet minket végig a folyamaton. Talán a leginkább elbizonytalanító rész az a lap, ahol az új adatbázist és táblákat létrehozzuk és a jogosult MySQL felhasználó nevét és jelszavát kell megadnunk. Ezen kívül a telepítő lehetne kicsit szószátyárabb a hibaüzenetek tekintetében, tekintve, hogy nem mindenki tapasztalt web/adatbázis programozó.

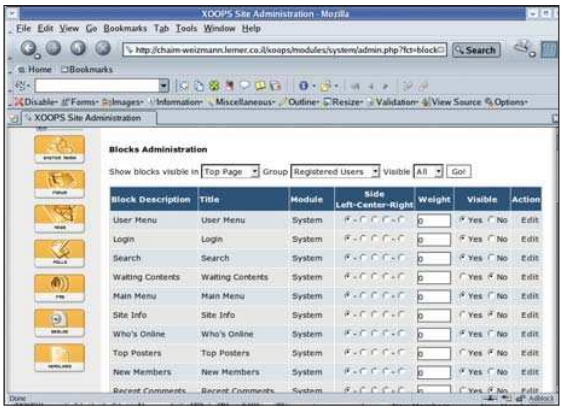
Azt kell mondjam, a telepítés nagyon egyszerű és gördülékeny volt. Visszatértem a rendszerem ➲ /xoops URL-jére és máris egy egyszerű bemutatkozó képernyővel találkoztam, amely belépésre invitált. Beléptem a lap adminisztrátoraként, és bár a főlap elég üres volt, gyorsan tovább tudtam lépni a képernyő bal felső részén található rendszergazdai menüre (*1. ábra*).

Az XOOPS kezelése eléggé magától értetődő, ha rájövünk, hogy XOOPS alatt minden modulnak számít. A legtöbb XOOPS gazda első dolga tehát további modulok feltelepítése. Lépjünk a kezelői (administration) menübe és válasszuk a Modules pontot. A képernyő tetején találjuk a már feltelepített modulokat, míg a lap alján sorakoznak azok, amelyek elérhetők de nincsenek telepítve.

Utóbbiak telepítéshez kattintsunk a modul neve melletti ikonra. Választásunk jóváhagyását követően megtekinthetjük milyen táblák jöttek létre a MySQL adatbázisunkban, milyen új elérési jogosultságok kerültek a rendszerbe valamint azt is, hogy a modul telepítése sikeres volt-e.

Tegyük fel, hogy így történt, és visszatérhetünk a Module Administration lapra, ahol láthatjuk, hogy a frissen feltelepített modul a képernyő aljáról átkerült a tetejére .

Az XOOPS oldal alapértelmezetten gyakorlatilag üres. Ha egyéb elemeket szeretnénk a honlapunkhoz adni, akkor át kell lépnünk a blokkok (Blocks) karbantartói lapjára. A News (hírek) modul három blokkot tartalmaz – ***bigstory***, ***topics*** és ***top***. A fő blokk-karbantartó menüben a lapgazda beállíthatja mely blokkok legyenek láthatóak. A dolog igazi erejét azonban az egyes blokkokhoz tartozó szerkesztés (Edit) menüben találjuk, ahol megadhatjuk, hogy a képernyőn hol jelenjen meg az adott blokk – oldalt, középen vagy éppen teljesen máshol.



2. ábra Készítsünk egyedi lapkinézetet a blokkok helyzetének megadásával a Blocks Administration menüben

Modulok

Minden XOOPS modul jól meghatározott és egységes könyvtárakba rendezett HTML, PHP és SQL részekből áll. Az osztálydefiníciókat mindig a ***class*** könyvtárban, az adatbázis definíciókat az ***SQL***-, a blokkokat a ***blocks***, míg a képeket az ***images*** könyvtárban találjuk. A beállítás és verzió információk mindig a ***xoops_version.php*** állományban, a modul gyökerében lelhetők fel.

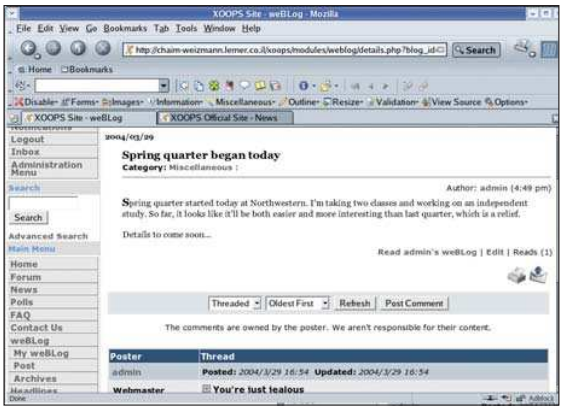
Az új modul telepítése mindössze annyiból áll, hogy letöltés után a könyvtárát áthelyezzük ***modules*** könyvtárba. Tehát ha letöltöttük a ***foo*** nevű modult, és a XOOPS könyvtárunk a ***/usr/local/apache/html/xoops***, egyszerűen csak másoljuk a ***foo*** könyvtárat a ***/usr/local/apache/html/xoops/modules*** helyre. Legközelebb, amikor a Module Administration lapra lépünk, a modult már ott találjuk, telepítésre és beállításra készen.

Az egyik fontos dolog amit a XOOPS fejlesztők szem előtt tartottak, az adatbázis továbbfejlesztése. A XOOPS modulok PHP és HTML részeinek fejlesztési irányát nem nehéz elképzelni. Egy PHP-t és SQL-t egyaránt alkalmazó modul frissítése ugyan már kicsit keményebb dió lehet, de úgy néz ki, a fejlesztők ezt sikerrel megoldották, és tiszteletre méltó munkát végeztek. A XOOPS modul SQL könyvtárában nem csak a MySQL definíciókat tartalmazó ***mysql.sql*** állományt találhatjuk meg, hanem az ALTER TABLE parancsokat tartalmazó frissítő állományt is, amely a régi verzióról az újra változtatja adatbázisunkat. Mivel a modulok táblái a rendszer többi részétől elszigetelve léteznek, a többitől teljesen függetlenül, és szinte bármilyen sorrendben frissíthetjük az XOOPS moduljait.

WeBLog

Miután ráéreztünk a modulokkal való munkára, hozzálathatunk ***Hiroyuki Sakai*** WeBLog moduljának telepítéséhez. Most végre olyan jellemzőkkel láthatjuk el programunkat, amelyeket eredetileg kerestünk: egy közösség által is használható webnapló rendszert építünk.

A WeBLog telepítéséhez töltsük le az állományt a főoldalról (lásd a forrásokat). Akárcsak a legtöbb XOOPS modul esetében, most is ki kell csomagolnunk a `tar_zxf` paranccsal, majd a legfelső szintű weblog könyvtárat átmásolnunk



3. ábra Siker! A beállított lapon megfigyelhetjük a felhasználók bejegyzéseit és a hozzájuk fűzött megjegyzéseket

vagy átmozgatnunk az XOOPS modulkönyvtárába. Az én rendszeremen a WeBLog a */usr/local/apache/html/xoops/modules/weblog* könyvtárba került.

A modul telepítéséhez, egyszerűen lépjünk a Module Administration lapra és kattintsunk az install (telepítés) pontra. Különféle weblog típusokat hozhatunk itt létre és állíthatunk be. A Module Administration lapon baloldalt megjelenik egy új napló-ikon, mellyel beállíthatjuk a jogosultságokat, a kategóriákat, valamint elvégezhetünk néhány alapvető adatbázis karbantartási feladatot.

Miután engedélyeztük a weblogok létrehozását, térjünk vissza a lapunk nyitóoldalára. Itt a képernyő bal oldalán már látnunk kell a WeBLog menüpontot a rendszer valamennyi legfrissebb weblog elemének gyűjteményével együtt. Ha a modult most futtatjuk első ízben, akkor ez a lap többnyire üres lesz. Saját weblogunkat a My WeBLog pontra kattintva hozhatjuk létre, illetve szerkeszthetjük. Küldeményünket gépeljük a HTML űrlap megfelelő részébe, legalább 75 karakter hosszúságban, hacsak a *Preferences Administration* lapon meg nem változtattuk ezt az értéket. Miután bevittük az új bejegyzésünket, a WeBLog menüelemre visszatérve meg is tekinthetjük azt. Ez a közös képernyő az összes egyéni naplót egyesíti, egyúttal azt is lehetővé teszi, hogy a felhasználók nevére kattintva, csak a kiválasztott blogot tekintsük meg.

A WeBLog számos egyéb hasznos lehetőséget is tartalmaz még, az RSS hírügynökségtől kezdve, a kereshető és böngészhető archívumon át a küldeményekhez fűzhető megjegyzésekig. Azt is kérhetjük, hogy az új küldemények felkerülésekor kapjunk figyelmeztetést. Az ilyen figyelmeztetések a főmenüben jelennek meg, amikor belépünk az oldalra.

Használjuk az XOOPS-t?

Amennyiben blogot használó emberek hálózati közösségét szeretnénk létrehozni, az XOOPS nem rossz választás. Aktív fejlesztőcsapat támogatja, a WeBLog modul elég stabilnak és jól karbantartottnak tűnik. Továbbá a technológia egyszerű és magától értetődő ahhoz, hogy a web/adatbázis tapasztalattal rendelkezők maguk módosíthassák, állíthassák be saját rendszerüket.

Az XOOPS vizsgálgatása közben azonban nem bírtam megállni, hogy össze ne hasonlítsam azt az OpenACS-el, ezzel a szintén közösség centrikus rendszerrel, amely ugyanúgy rendelkezik weblog modullal. Az érdekelt, hogy vajon van-e bármilyen ok, amiért valaki inkább az XOOPS rendszerét választaná az OpenACS helyett, amely technikailag kétségtelenül messze fejlettebb.

Bármennyire is szeretem az OpenACS-et, a válasz egyértelműen igen. Az OpenACS kiváló rendszer ha egy keményvonalas web/adatbázis-guru van a közelben, és ha root jogosultságunk van a kiszolgálónkhoz, továbbá, ha az OpenACS eszközkészletet saját igényeinkhez tudjuk igazítani. Sokaknak azonban nincs megfelelő embere vagy pénze egy ilyen magasszintű rendszer használatához; ők a PHP-t és MySQL-t futtató megosztott kiszolgálókon is gyorsan és könnyen létrehozható megoldásokat keresik. Nekik az XOOPS nagyon jó választás. Ugyan sok minden hiányzik belőle, de előnyei is számosak: bárhova feltelepíthető, a PHP/MySQL programozók hatalmas közössége támogatja és minimális ismerkedés után a nem-programozók által is karbantartható. Sok dolog hiányzik belőle, amit egy kifinomultabb rendszertől elvárnék. Például nincs Unicode támogatása, nincs lehetőség egyetlen csomag több példányban való telepítésére, tartalom-változatkezelésre és hierarchikus csoportok létrehozására. De arra amire tervezték nagyon is jól használható.

Úgy érzem nehéz lenne a XOOPS-t a hagyományos értelemben vett tartalomkezelők közé sorolni, akármit is állítsanak a csomag karbantartói. Igaz, valóban lehetővé teszi, hogy új küldeményeket postázzunk és azokat a hely gazdája megjelenés előtt engedélyezze. Azonban az XOOPS rendszert inkább egy könnyen használható közösségi felületként képzelhetjük el, melynek segítségével egyszerűen hozhatunk létre weblogokat és egyéb szabványos tartalmakat. Sok embernek ez több is, mint elég. Az, hogy ilyen egyszerűen hozhatnak létre egy kifinomult honlapot, sokkal nagyobb súllyal esik latba mint az egyéb elvárások..

Összefoglalás

Az XOOPS egyértelműen nem való mindenkinek–a világnézet, az objektum-központú kód és az említett funkciók hiánya miatt kétszer is meggondoltam, egyáltalán ajánljam-e. Csakhogy azt is láttam, mennyi időt és energiát emészt fel az indulás olyan alkalmazásokkal mint a Zope vagy az OpenACS – és ez a csere nem mindig kifizetődő. Ha kiválóan értünk a PHP és MySQL eszközökhöz, és jó weblog képességekkel felvértezett közösségi oldalt szeretnénk, nem árt elgondolkodni az XOOPS használatán.

A cikkhez tartozó források a CD-n a Magazin/XOOPS könyvtárban találhatóak.

Linux Journal 2004. június, 122. 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.

Linuxos kiszolgálót mindenkinek! (7. rész)

A SuSE Linux, mint kiszolgáló, kisvállalati és otthoni környezetben.

A kedves olvasó a sorozat előző részeiben már találkozhatott több a Suse Linux segítségével létrehozott internetes szolgáltatás megvalósításával. Mostani cikkemben bemutatom, miként csinálhatunk kiszolgálót a legnépszerűbb internetes szolgáltatáshoz, a webhez. Telepíteni fogunk egy webkiszolgálót, hozzá egy adatbáziskezelőt, valamint egy szerver oldali szkriptelő környezetet. Ez a hármas együtt olyan környezetet alkot, amivel bátran hozzákezdhetünk weboldalak fejlesztéséhez is.

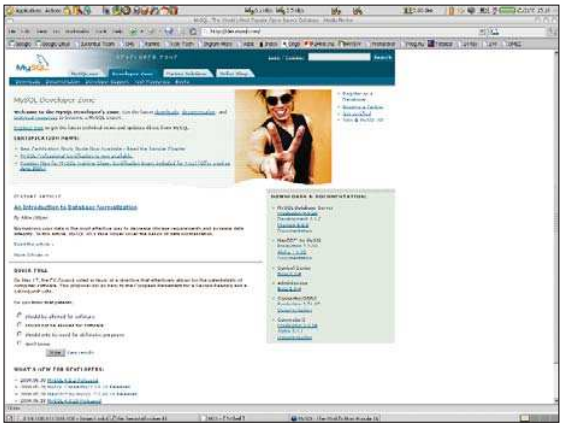
A Webkiszolgáló

Linuxos környezetben a webkiszolgáló egyet jelent az Apache kiszolgálóval. Ennek története olyan sikertörténet, amelyet eddig kevés termék tudott utána csinálni. Megjelenése óta folyamatosan nő a felhasználók köre és részesedése a webkiszolgálók területén. 2004 februárjában a webkiszolgálók közel 70%-a Apache-ot használt. Ez önmagában is figyelemre méltó eredmény, hát még ha megnézzük, hogy kik a vetélytársai. Olyan, kicsinek nem mondható cégek, mint a Microsoft az IIS-el, vagy a Sun a SunONE projektjével. Ezek - kis túlzással - jelenleg a fasorban sincsenek. Nem járunk messze a valóságtól, ha a „kis indián" sikerét egészen egyszerűen elsöprőnek nevezzük. (Helyzetjelentés 2004. májusából a Netcraft.com-on:
➲ http://news.netcraft.com/archives/web_server_survey.html)
Ha fentiekhez még azt is hozzávesszük, hogy az Apache egy ingyenesen használható, fantasztikusan sokrétű szolgáltatásokat tartalmazó szoftver, akkor azt hiszem, nem kérdés, hogy mitől olyan sikeres. Használjunk hát mi is Apache-ot! Oké, használjunk Apache-ot. De melyiket? Van ugyanis 1.3.x és 2.0.x. Ráadásul a válasz nem is olyan egyszerű, mint amilyennek első pillantásra tűnhet. Ha ugyanis jobban megvizsgáljuk a változatokat, hamar kiderül, hogy mindkettőnek megvannak az előnyei és hátrányai egyaránt.
A 2.0 ugyan újabb és ez sokaknak rengeteget számít, de az 1.3-as verzió a stabilitásával és kiforrottságával olyan alternatívát nyújt, amelyet kritikus környezetben egészen egyszerűen nem szabad figyelmen kívül hagyni.
A Suse 8.2, 9.0 és a legfrissebb 9.1-es verziója is tartalmazza mindkét verziót, sőt a YaST a 9.0-tól konfigurációs felületet is nyújt, de sajnos csak az Apache2-höz és annak is csak az alapvetőbb beállításaihoz. Akinek viszont elegendőek ezek a funkciók, azok nyugodtan használják ki az Apache 2.0 és a YaST nyújtotta szolgáltatásokat. Én most az 1.3-as verzióval fogok foglalkozni, így ez a cikk nem szigorúan csak

Suse webkiszolgálók beállításához szolgál útmutatóul. Viszont a 2.0-ás verziót választók is kapnak némi útmutatást, hiszen az alapvető beállítások nagyjából megegyeznek.

A telepítés

Webszervernek alapesetben telepítsük az `apache` csomagot, a php szkriptelő nyelv telepítéséhez pedig a `mod_php4`, illetve a `mod_php4-core` csomagokat.
Ha ezzel megvagyunk, akkor leraktuk az alapjait egy dinamikus adatokat szolgáltatni képes webes rendszernek. Amennyiben szeretnénk – és miért ne szeretnénk – adatainkat adatbázisban tárolni, telepítsünk egy adatbáziskezelőt is.
Az adatbáziskezelők esetében jóval nagyobb a választék, mint a webkiszolgálók terén. Sokuk fizetős, de több ingyenesen használható rendszer közül is választhatunk. A fizetős rendszerek közül talán a legnépszerűbbek az Oracle és MSSQL. Ezek ugyan nem olcsók, de szolgáltatásaikban egy bizonyos szinten felül olyan pluszt nyújtanak, amelynek nélkülözése, vagy saját erőből történő megvalósítása az adott projekt szempontjából esetleg kivitelezhetetlen. Ha azonban ezekre a szolgáltatásokra nincs szükségünk, akkor nyugodt szívvel használjuk valamelyik ingyenes rendszert. Linux alatt a két legnépszerűbb a MySQL és a PostgreSQL. Mindkettő nagy tudású, nagy teherbírású adatbázis-kezelő, de természetesen itt is vannak szolgáltatásbeli különbségek. Mi most a MySQL telepítésével fogunk foglalkozni, de a PostgreSQL telepítése teljesen hasonló ehhez.



A MySQL telepítése, meglepő módon, a `mysql` csomag telepítésével hajtható végre. Érdemes a MySQL-hez feltelepíteni a `phpmyadmin` nevű csomagot, mert ez egy remekül használható grafikus felületet nyújt a MySQL kezeléséhez, amelyet kár volna kihagyni.

Az Apache beállítása

Az Apache beállításait Suse Linux környezetben az `/etc/httpd` könyvtárban végezhetjük el. Ebben a könyvtárban több különböző konfigurációs állomány is található, így most ezeket fogjuk szépen sorba venni.

Telepítéskor szerencsére a Suse elég jól összeállít egy olyan konfigurációt, amely kiindulási alapként szolgálhat. Ha az */etc/httpd* könyvtárba belépünk, akkor találunk további alkönyvtárakat, mint például a `modules`, illetve az `ssl.*` könyvtárak.

A Suse a modules könyvtárban tárolja a telepített modulokhoz tartozó beállítási részleteket, amelyeket a *httpd.conf* állomány feldolgozása közben majd használni fogunk.

Az ssl.* könyvtárak a HTTPS protokoll használatához szükséges tanúsítványok tárolására szolgának, így a HTTPS-t használó oldalainkhoz készített tanúsítványokat itt célszerű elhelyezni.

Mielőtt rátérnénk a *httpd.conf* állomány elemzésére, vessünk egy pillantást az */etc/sysconfig/apache* állományra, ugyanis a Suse-tól megszokott módon itt a program több paramétere beállítható.

Szerencsére mind a sysconfig állományok, mind az apache konfigurációs fájljai elég bőbeszédűen kommentáltak, a megértésükkel valószínűleg nem lesz különösebb probléma, így helyszűke miatt most csak a fontosabb és érdekesebb beállításokat nézzük meg.

A *sysconfig* könyvtárban található *apache* állomány első beállítása rögtön az, hogy akarjuk-e egyáltalán ezt az állományt az Apache konfigurálásához használni. Ha az `ENABLE_SUSECONFIG_APACHE` változót yes értékkel láttuk el, akkor érdemes továbbmenni az állományon, ha nem, akkor minden egyes konfigurálást személyesen kell elvégezni a *httpd.conf* állományban.

A `HTTPD_SEC_SERVERADMIN` változó egy sztringet vár értéknek, mégpedig a rendszergazda e-mail címét. Ezt azért hasznos megadni, mert a rendszer hibás működését észlelve a felhasználók ezen a címen tudják a rendszergazdának jelezni a tapasztalataikat.

A `HTTPD_SEC_PUBLIC_HTML` változó értékét yes-re állítva lehetővé tesszük, hogy a rendszer felhasználói a saját home könyvtárukban található *public_html* könyvtárban elhelyezett weblapokat megtekinthessék a ➲ http://szerver_neve/~user_neve cím alatt, mégpedig az */etc/https/suse_public_html.conf* állományban leírtak szerint.

A php szkriptnyelv használatához a `HTTPD_SEC_MOD_PHP` változó értékét állítsuk yes-re. Innentől kezdve az Apache induláskor betölti a php modult az */etc/httpd/modules* könyvtárból.

A fentiekhez hasonlóan az SSL kiterjesztést a http protokollhoz a `HTTPD_SEC_MOD_SSL` változó értékének yes-be billentésével tölthetjük be.

És akkor most térjünk rá a *httpd.conf* állományra.

Amint azt már említettem, az apache ebben az állományban tárolja a beállításait, így ennek turkálásával jóformán minden beállítást el tudunk végezni.

A *httpd.conf* állomány szekciókra van szabdalva és minden szekciót bőséges magyarázattal láttak el. Az első szekcióban megadhatjuk a kiszolgáló globális beállításait. Elsőnek a `ServerType` változót kell értékkel ellátnunk és mivel mi egy otthoni, vagy kisebb vállalati környezetben szeretnénk a kiszolgálót üzemeltetni, a `standalone` alapbeállítás a megfelelő számunkra.

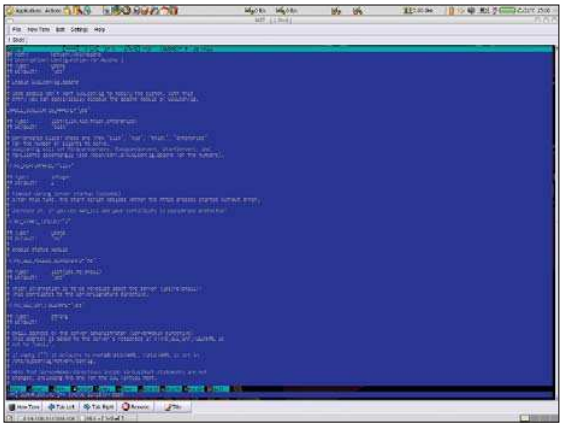
A második beállításként a `ServerRoot` változó értékének meg kell adnunk, hogy a Suse fáljrendszerén belül melyik könyvtárban található a webkiszolgáló alapértelmezett könyvtára. Ez alapesetben az */srv/www*, de természetesen áthelyezhető egy tetszőleges könyvtárba.

Következőkben *lock* és *pid* állományok elérhetőségét adhatjuk meg, ezeket azonban célszerű az alapértelmezett értékükön hagyni.

A következő négy beállítás a ügyfelek kiszolgálóval való kapcsolattartására vonatkozik. A `Timeout` változó értéke adja meg, hogy a szerver mennyi idő után küldjön időtúllépés üzenetet a böngészőnek. Ezt az értéket érdemes a kapcsolatunk sebességéhez és a szerver terheltségéhez igazítani. Az alapértelmezett 300 másodperc tapasztalatunk szerint jó kiindulóérték, csak akkor változtassuk, ha nagyon muszáj.

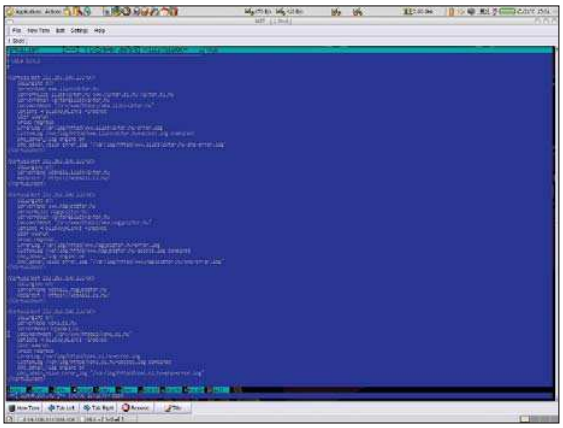
A `KeepAlive` kapcsoló értékének On-ra állítása nagyon jó ötlet, mert ezzel ki tudjuk használni, hogy ne csak egy kérés fusson le egy TCP kapcsolat alatt, így rengeteg időt ta-

karíthatunk meg az egyes TCP kacsolatok külön felépítésének kivédésével. Adott esetben ez óriási sebesség-növekedést jelenthet. Ezzel együtt állítsuk be a `MaxKeepAliveRequests`-nél, hogy hány kérést kívánunk kiszolgálni egy TCP kapcsolat alkalmával. Tanácsos ezt a számot jó nagyra állítani, hogy minél kevesebbszer kelljen a TCP kapcsolatot felépíteni.
Ha 0-ra állítjuk, akkor korlátlan számú kérés kiszolgálható egy TCP kapcsolaton belül, de jó ha tudjuk, hogy ez nem mindig jó ötlet.
A `KeepAliveTimeout` változó értékében megadhatjuk, hogy két kérés között maximum hány másodperc telhet el, így védekezhetünk az ellen, hogy sok, esetleg halott kapcsolatot tartsunk fenn.
Az első szakaszban található továbbá a `MaxClients` változó, ahol be tudjuk állítani, hogy a kiszolgáló egyszerre hány klienst kezelhet maximum. Ez biztonsági szempontból fontos lehet, ugyanis ha a kiszolgáló teljesítményéhez mérten állítjuk be, akkor egészen hatékonyan ki tudjuk védeni a webszervert érő DoS támadásokat.
A `Listen` és `BindAddress` változók kitöltésével meg tudjuk határozni, hogy a kiszolgáló milyen címeken és milyen portokon fogadjon kéréseket. Ez globális beállítás, ami nem feltétlenül szerencsés, ha több oldalt is ki szeretnénk szolgálni, ezért inkább használjunk `VirtualHost`-okat, amelyekről mindjárt szó lesz.
A következőkben a `LoadModule` és `AddModule` résznél a rendszer által használt modulok betöltésére kerül sor, amiket csak akkor módosítsunk, ha tudjuk, hogy mit csinálunk, mert elég nagy galiba lehet a végeredmény. Másfelől pedig figyeljük oda arra, hogy a Suse a ***sysconfig*** könyvtárban beállítottak szerint itt automatikusan betölti például az SSL és PHP modulokat, ha ezt beállítottuk.
Következik a második szekció, az alap kiszolgáló beállítása. Ez egy nagyon hosszú beállítási csomag, ezért nem is vállalkozom most arra, hogy részletesen ismertessem, mert azzal az egész újságot meg lehetne tölteni. Ezért megnézünk pár alapbeállítást, majd létrehozunk néhány különböző virtuális kiszolgálót.
Első beállításunk a `Port` változó megadása legyen. Itt arról van szó, hogy a kiszolgáló alapesetben melyik porton figyelje a beérkező kéréseket. Ez alapértelmezésként természetesen a 80-as és ettől eltérni csak extrém esetekben szokás, mert előfordulhat, hogy így nem találnak meg minket.
Következő beállításnak az `<IfDefine SSL>` jelölések között megadhatjuk a `Listen` változónak, hogy amennyiben az SSL modul be van töltve, akkor ne csak a 80-as, de a HTTPS, azaz a 443-as kaput is figyelje.
A `User` és `Group` változók azt jelölik, hogy a szerver könyvtárában elhelyezett adatok kinek a tulajdonában kell hogy legyenek, azaz milyen jogosultságok szükségesek a file-ok olvasásához, az oldal megjelenítéséhez. A Suse a telepítéskor a `wwwrun` beépített felhasználót és a `nogroup` csoportot használja. Tehát az */srv/www/htdocs* könyvtárban elhelyezett ***index.html*** és egyéb állományokra olyan jogot kell adni, hogy ez a felhasználó lássa. Ha ezt nem tesszük meg, egy gyönyörű „az oldal megnyitására nincs joga” üzenetet fogunk kapni a böngészőtől.
Nézzük a következő változót. ***ServerAdmin*** és egy e-mail címet vár tőlünk. Ha most kicsit elgondolkozunk azon, hogy



mintha ezt már egyszer beállítottuk volna, akkor jó úton járunk, ugyanis valóban megtettük. A `sysconfig`-ok között már beállítottuk az alapértelmezett e-mail címet, így a rendszer töltésekor a Suse ezt nekünk be kell, hogy helyettesítse. Az adminisztrátori e-mail cím beállítása ékes példája annak, hogy a Suse miként képes arra, hogy felülírja nekünk az Apache beállításait, ha nem figyelünk oda. Amennyiben nagyobb mértékű módosításokat hajtunk végra a ***httpd.conf***-ban, akkor esetleg megfontolandó, hogy kikapcsoljuk a sysconfig-ban az Apache támogatást, így teljes uralmunk lehet az Apache beállításai felett.
Következő beállításunk a szerver nevének megadása. Vigyázzunk azonban, mert amennyiben a beállítás létezik az */etc/sysconfig/network/config* állomány `FQHOSTNAME` változójánál, akkor ez az érték is felülírásra kerül. Így ezt az értéket is inkább az előbb megadott helyen állítsuk.
A `DocumentRoot` változónak azt a könyvtárat kell megadnunk, ahol az alapértelmezett weboldalt tárolni szeretnénk. Ez Suse-éknél a */srv/www/htdocs* könyvtár.
A következő rész a rendszer biztonsági beállításairól szól. Ez egyfelől egy nagyon hosszú, másfelől egy elég összetett beállítási csoport, így ezt most ezen cikk keretei között nem tárgyaljuk. De akit igazán érdekel egy biztonságos webkiszolgáló üzemeltetése az ne röstelljen dokumentációt olvasni!
És akkor most nézzünk pár gyakorlati példát virtuális kiszolgálók létrehozására. Ezek azért jók nekünk, mert segítségükkel minden egyes oldalhoz külön-külön meg tudjuk adni a beállításokat, és így elkerülhetjük az ütközéseket. Ezeket a bejegyzéseket általában a ***httpd.conf*** állomány végén szokták elhelyezni, de még elegánsabb, ha egy külön fájlban és a ***httpd.conf*** végére csak egy `include` paranccsal hozzáfűzzük.
Egy virtuális kiszolgáló definiálása a következőképpen néz ki:

```
<VirtualHost 123.123.123.123:80>
SSLEngine off
ServerName www.tartomany.hu
ServerAdmin webmaster@tartomany.hu
DocumentRoot "/srv/www/htdocs/www.tartomany.hu"
Options +FollowSymLinks +Indexes
User wwwrun
Group nogroup
```



```
ErrorLog /var/log/httpd/www.tartomany.hu-error.log
CustomLog /var/log/httpd/www.tartomany.hu-
access.log combined
php_admin_flag engine on
php_admin_value error_log
"/var/log/httpd/www.tartomany.hu-php-error.log"
</VirtualHost>
```

Az első sorban megadjuk az IP címet, amin a kiszolgáló figyeli az adott virtuális kiszolgálóra érkező kéréseket. Amennyiben az IP cím helyére *-ot teszünk, akkor minden IP-re érkező kérést megpróbál feldolgozni. Az IP cím után található a port-szám, amin ül a kiszolgáló.
A második sorban beállítjuk, hogy ne használjunk SSL kiterjesztést. Amennyiben itt ***on*** értéket írunk, akkor a kiszolgáló port-számát állítsuk a 443-ra, mivel alapesetben a HTTPS-nek ez a száma.
A harmadik sorban megadjuk azt a címet, amit a felhasználó a böngészőbe írva eljut hozzánk. A negyedik sor ismét egy adminisztrátori e-mail címet rejt.
A `DocumentRoot` az oldal gyökérkönyvtárára mutat, míg az Options-nél az állítottuk be, hogy a könyvtárban található közvetett hivatkozások kerüljenek feldolgozásra, illetve a +Indexes azt jelenti, hogy amennyiben a webszerver nem talál egy ***index.html***, ***index.htm***, ***index.shtml***, ***index.php***, ***index.php4***, ***index.php3***, ***index.php***, vagy ***index.cgi*** (esetleg egyéb előre beállított típusú) állományt a megadott könyvtárban, akkor html formázással kiteszi az adott könyvtár tartalmát. Ez egyfelől egy jópofa funkció, hiszen így egy lekérdezhető fájlszervert csinálhatunk a weben, ugyanakkor bosszantó is lehet, ha olyan állományokat tárolunk, amelyeket nem szeretnénk direktben megmutatni.
A `User` és `Group`- változók szerepe a korábban elmondottak alapján remélem egyértelmű.
Az `ErrorLog` és `CustomLog` változóknak megadhatjuk, hogy az adott virtuális kiszolgáló hová naplózzon, így lehetőségünk van kiszolgálónként külön naplóállományt készíteni, ami felügyeleti szempontból nagyon praktikus.
A `php_admin_flag engine` on beállításával engedélyezzük, hogy az oldalhoz tartozó állományokban található php szkripteket a php feldolgozza és azt mint kimenetet jelenítse meg számunkra. Ha php-t szeretnénk használni, akkor ezt mindenképpen kapcsoljuk be.

Utolsóként pedig azt állítottuk be, hogy a php feldolgozás közben keletkező hibákat hol jelezzük.
Nos ennyit röviden egy virtuális kiszolgáló beállításáról. Apache-al kapcsolatban nagyon jó dokumentációt találunk a ➲ http:\\www.apache.org oldalon, érdemes olvasgatni.

A PHP beállítása

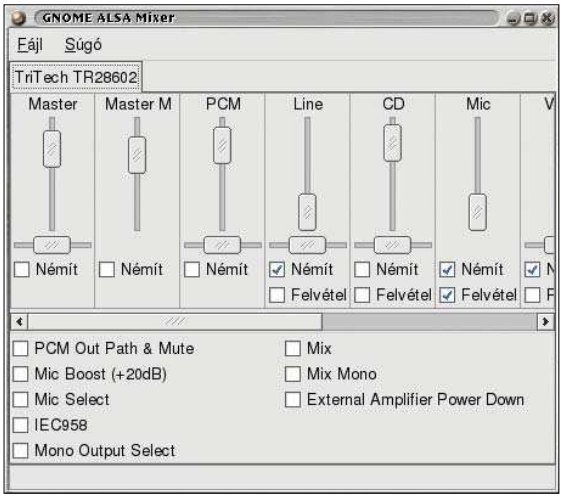
A PHP feldolgozó beállításait az ***/etc/php.ini*** állományban végezhetjük el. Ez is egy jól kommentált konfigurációs fájl és szintén nagyon sok beállítást tartalmaz. A beállítások egy jelentős részéhez nem kell nagyon hozzányúlnunk, de nézzünk pár hasznos megoldást.
A két legfontosabb rész most számunkra a `Data Handling` és a `File Uploads` lesz.
Előbbi felelős a php-n átfolyó adatok kezeléséért, utóbbi pedig a feltöltések beállításait tartalmazza.
A `Data Handling` alatt található a `post_max_size` változó, amely azt mutatja, hogy a webszerver által küldött POST adat mérete legfeljebb mekkora lehet. Amennyiben a weboldalunkon olyan űrlapokkal dolgozunk, melyek adott esetben nagyobb adatot is tartalmazhatnak (például képek), akkor érdemes megbecsülni ezt az értéket és esetleg az alapértelmezett 8 Mb-nál nagyobb értéket megadni.
A másik, számunkra nagyon fontos változó a `default_charset`, amivel be tudjuk állítani az alapértelmezett karakterkészletet. Ezt a magyar betűk helyes megjelenítésének érdekében ajánlatos `iso-8859-2` értékre állítani.
A `Data Handling` részben érdemes még egy változóra, nevezetesen a `register_globals`-ra odafigyelni. Ha ezt a funkciót bekapcsoljuk, akkor képesek leszünk a rendszerben globális változókat kezelni, ami egyfelől egy segítség a programozásban, másfelől viszont támadhatóvá tesszük a rendszert. Ennek a funkciónak a bekapcsolását gondoljuk át!
A `File Uploads` rész akkor lehet számunkra fontos, ha olyan oldalakat szeretnénk üzemeltetni, amelyeken keresztül állományokat lehet feltölteni a kiszolgálóra. Először is engedélyeznünk kell a feltöltést a `file_uploads` változó On értékre állításával, majd az `upload_tpm_dir` változónak megadhatjuk, hogy melyik könyvtárat használja az ideiglenes állományok tárolására. Az `upload_max_filesize` változó pedig azt a fájlméretet mutatja, amely maximálistan feltölthető. Ez alapértelmezésként 2 Mb.
Ha a fenti beállításokkal végeztünk, akkor olyan rendszerhez juthatunk, amely alap webkiszolgálóként megállja a helyét. Természetesen itt is, mint mindenhol, igaz az, hogy nincs két egyforma igény, így a saját igények kielégítése már nem ennek a cikknek a keretei közé tartozik.
Használjuk bátran a hálón lévő doksikat, olvassuk őket figyelmesen és tanulmányozzuk rendszeresen, mert így viszonylag rövid idő alatt szert tehetünk a feladatok elvégzéséhez szükséges tudásra. Mindenkinek eredményes próbálkozást kívánok.

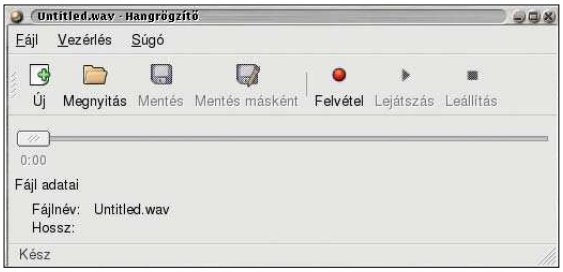
Illés Viktor (viktor@ei.hu)
23 éves, a BME műszaki informatikus szakának hallgatója, mellette weblapokkal, linuxos és windowsos rendszerekkel foglalkozik. Szabadidejét legszívesebben a szabadban tölti, teniszezik és kerékpározik.

A Magyar Linux: UHU

Sokáig érte az a vád a Linuxot, hogy multimédiában nagyon gyenge. Nos ez az utóbbi 1-2 évben jelentősen megváltozott. Ha van is a Linuxnak némi hátránya e téren, az folyamatosan csökken. És ez igaz az UHU-Linuxra is.

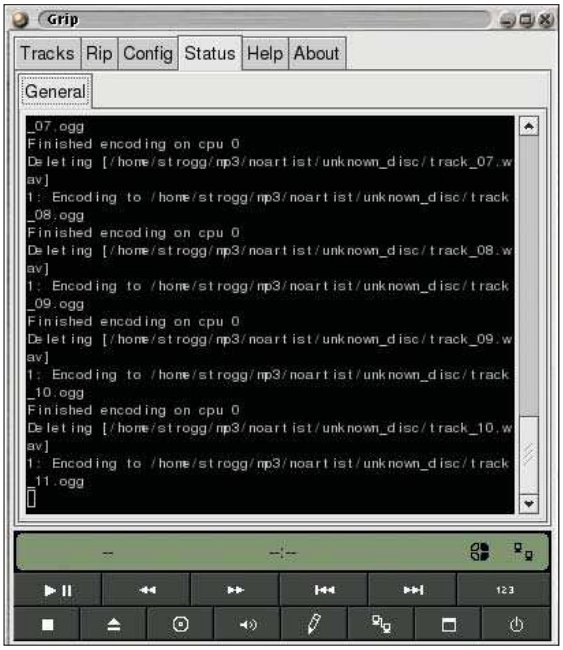
Minden operációs rendszer tartozéka a mixer. Ez egy mini keverőpult, ahol számos audio-sáv hangerejét megváltoztathatjuk. A komolyabb darabokon ezek mellett pedig még a hangszín, pár effekt, és a 3D-s hangok szabályozását, megváltoztatását is lehetővé teszik. Azonban ebből is több féle létezhet. Alapvetően két programmal találkozhatnak a felhasználók, az egyik a Kmix, másik pedig a Gnome-AlsaMixer. Előbbi a KDE, utóbbi a Gnome felhasználói felülettel együtt telepszik gépünkre. Mindkét program képes beépülni az adott ablakkezelő tálcájába, így könnyen elérhető és hasznos társ válhat belőle. Személyes meglátásom, hogy a Gnome alá fejlesztett ilyen programok mintha kiforrottabbak lennének.
A Gnome-Alsa-mixer több beállítási lehetőséget ad, mint a KDE-s Kmix, ráadásul finomabban lehet vele dolgozni. A Kmix mindössze az adott sávok hangerejének szabályozására használható, többet nem tud. A Gnome hangrögzítője is stabilabbnak bizonyult a Krecnél.
Ezt a saját tapasztalatomra alapozom: a Compaq Armada M700 laptopon, ESS-Maestro 2 hangkártyával katasztrofális minőséget produkált a Krec, amihez képest a Gnome hangrögzítője kifejezett felüdülés volt. Azért persze senki ne várjon tőle csodát, hiszen egyik sem komolyabb felhasználásra készült.
Lejátszás terén még jobban áll az UHU. Az xmms manapság már szinte nélkülözhetetlen minden Linux összeállítás esetén, így itt, az UHU-ban is megtalálható.
Ez az a lejátszó, amelyet a Linux winamp -jának is nevezneknek, hisz majdnem ugyanazt tudja. Sőt, bizonyos bővítmények megléte esetén videót is lejátszik, (akárcsak a másik rendszerbeli rokona) és sok más funkció is megvalósítható vele. (rengeteg formátum, vizuális, és audio effektek, stb...)
CD lejátszásra a gnome CD player éppúgy alkalmas, mint a KDE alatti KsCD. Utóbbinak előnye, hogy képes beépülni a tálcára, és onnan bármikor, kattintásra elindítani a Cd lejátszást. A sort még hosszasan lehetne folytatni, a multimédiás lehetőségek mérlegelése nem azonban itt kezdődik, hisz ezek alapfunkciók. Nézzük tovább...





Audió tömörítés, rippelés

Hosszas kísérletezgetés után úgy láttam, hogy a legjobb CD rippelő program linuxra a Grip. Kényelmes használni, a sebességével sincs gond, és kellemesen, sokféle módon állítható be, mégsem túl bonyolult a kezelése. Számos hasonló program válik nehézkessé azért, mert a készítők mindent bele akarnak zsúfolni, és a végeredmény egy programnak látszó állatorvosi ló lesz. Ezért a kezelés nehézkessé válik, és aki pusztán kedvenc dalait szeretné mp3-ban tudni, annak mindenképpen kényelmetlen is. Nos a Grip nem esik ebbe a hibába. Jó érzékkel ismeri fel, hogy hol a határ, mikor kezd a funkcionalitás a kezelhetőség rovására menni. De lássuk mit is tud pontosan? A program telepítése az apt-get -el történik, igencsak gyorsan, ami elsősorban annak köszönhető, hogy nem túl nagy méretű (337 KB) programról



van szó. A telepítés után vagy konzolból indítjuk a „grip" parancssal, vagy pedig menüből. Célszerű a program indítása előtt behelyezni a CD-t (ez nem kötelező, csak kényelmesebb), ugyanis az induláskor az első fül amit meglátunk, a „Tracks". Ha behelyeztük a CD-t, nyomban láthatjuk is a sávokat.

Itt akár ki is jelölhetjük azonnal, mit szeretnénk leszedni. A következő fül, a „Rip". A program, nagyon felhasználóbarát módon, többféle lehetőséget ajánl fel, szó szerint gombnyomásra. ***Rip+encode, rip only***, ezek megszakítása, vagy a DDJ Scan. Ugyanitt egyébként a CD szektorai szerint is megadhatjuk, hogy mit is szedjen le a korongról. Illetve, ha már elindítottuk a rippelést, a kiválasztott funkcióknak megfelelően láthatjuk, hogy hol is tart a program.

A harmadik a „Config" fül. Valószínűleg itt fogunk a legtöbbet időzni. Olyan sokféle beállítást ajánl fel, amely több mint elegendő egy profi válogatás elkészítéséhez. Megadhatjuk a CD eszközleírójának a helyét, a külső rippelő programot (`cdparanoia`, `cdda2wav`, `other`), a kiválasztott program „parancssori" paraméterit (pontosabban finomíthatjuk, mert előre ki van töltve), és megadhatjuk az encoder típusát is (`Baldeenc`, `Lame`, `mp3encode`, `gogo`, `oggenc`). Én általában az ogg-ot javaslom, mivel ekkor az ogg-vorbis nyílt formátumú kódoló .ogg kiterjesztésű fájlokat hoz létre. Ez valamivel jobb minőséget, és kisebb fájlméretet produkál mint az mp3, ráadásul teljesen nyílt formátum, mentes mindenféle licensz-problémától. (amely mint tudjuk egyre komolyabban sújtja az mp3-at, ezért terjedhet az .ogg olyan gyorsan.) Természetesen mondanom sem kell talán, hogy az UHU-ban lévő xmms alapból le tudja játszani az .ogg fájlokat. A többi beállítási lehetőség már az internet felé mutat: A CDDB adatbázisok elérését, és a proxy-t állíthatjuk be, ha szükségét látjuk. Egy profi válogatáshoz valóban elkél.

Az utolsó hasznos fő fülecske, a „Status". Itt követhetjük nyomon, hogy mit is csinál éppen a program, amikor dolgozik.

Mindez nem hangzik túl soknak, de valljuk be, az esetek 95%-ban nincs is szükségünk ennél többre. A program alján található vezérlőgombok pedig nevetségesen egyszerűvé teszik a kezelését. Véleményem szerint ez az egyik legjobb program, amelyet mp3, vagy ogg gyűjtemény létrehozására érdemes használni. Sebességére igazán nem lehet panasz, hisz a mai processzorok teljesítménye mellett teljesen rutinfeladat egy CD leszedése, és tömörítése. Nálam ez a művelet közel 20 percig tart egy Duron 800 Mhz-es gépen, és 700 MB méretű CD esetén. Mindeközben több példányban fut az OpenOffice.org, mail kliens, irc kliens, vagy éppen egy kisebb játék.

Hangszerkesztés UHU-n

Már linuxon sem ördöngősség a hang-adatok szerkesztése. Az Audacity nagyon népszerű, és igen jó program,1.2.1-es, meglehetősen stabil verzióját érdemes használni, amely letölthető a ➲ http://audacity.sourceforge.net címről. De létezik .uhu csomagban is, a telepítés ez esetben szintén az `apt-get`-el történik.

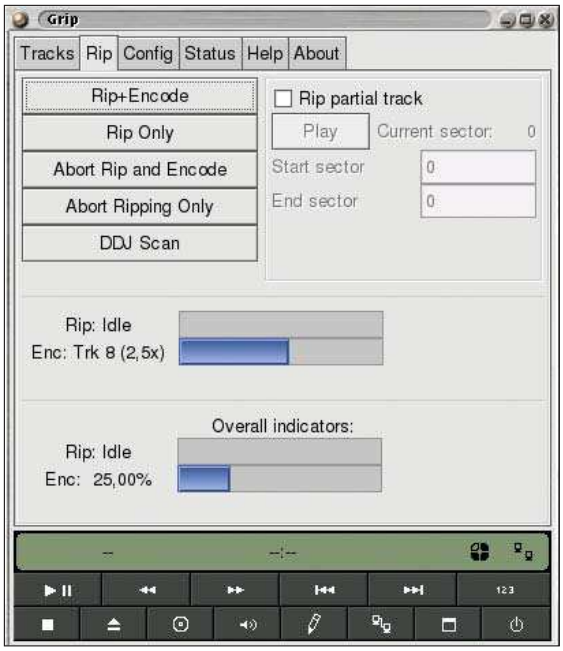
Célszerű először a wxWidgets készletet telepíteni, mivel erre feltétlen szüksége van. A program indulásakor kellemes, átlátható, és határozottan ízléses külsejű felület fogad minket, gyakorlatilag azonnal munkára készen.

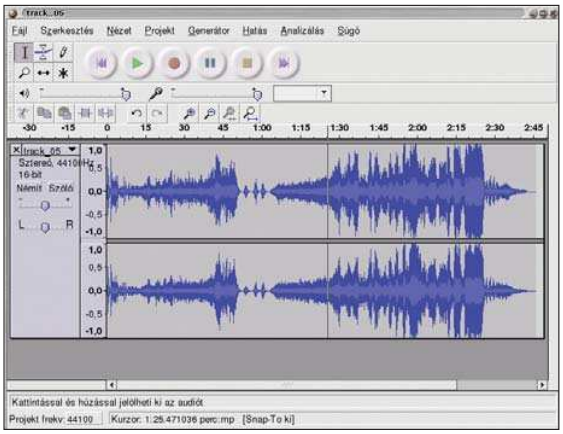
A beállításokra mindenképpen vessünk egy pillantást, hisz ha esetleg több hangkártyánk van, érdemes beállítani őket. Nem árt továbbá vetni egy-egy pillantást minden menüpontra, mert nem mindig helyes, amit ő annak gondol.

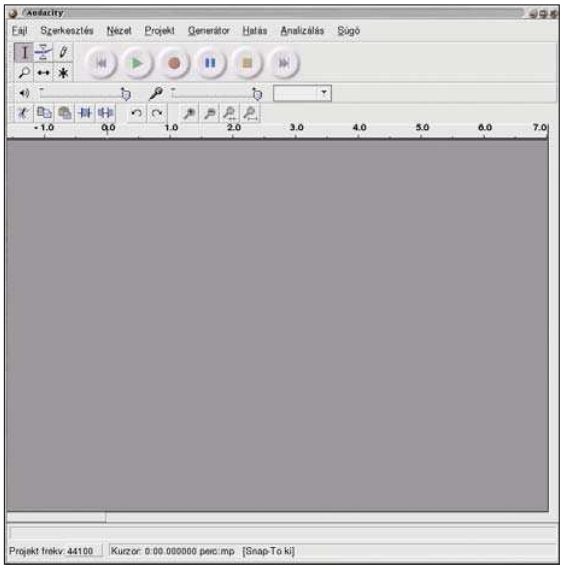
Ha úgy látjuk, hogy minden rendben, akkor akár munkához is láthatunk. Erre többféle lehetőségünk is van, egész pontosan három. A program ugyanis képes digitalizálni, külső fájlt megnyitni (ogg,mp3,wav), vagy projeckt-et nyitni. A digitalizálásnál aprónak tűnő, ám a kényelmet nagyban szolgáló megoldás, hogy bárhonnan képes fogadni az adatot, tehát a hangkártya bármely bementi pontjáról tudunk felvenni.

Ez természetesen akkor is igaz, ha bakelit lemezes lemezjátszónkat csatlakoztatjuk, vagy éppen egy külső CD lejátszót. A felvétel közben azonnal látható és hallható az eredmény, sőt korrekciókra is lehetőségünk nyílik. Persze ennek akkor még nem sok értelme van, hisz úgy is átfutunk a teljes anyagon.

Több hasznos, és egyszerű módja is van a felvételeink javításának. A normalizálás akkor jön jól, ha a hangerőt szeretnénk egy részen valamivel visszavenni, és máshol érintetlenül hagyni. Ekkor egérrel kijelöljük a kívánt részt, majd kiválasztjuk a ***Hatás->Normalizálás*** menüpontot.







Válasszuk a „Normalize amplitude to -3 dB” pontot, és készen is vagyunk. A másik fontos, (és valószínűleg nagyon gyakran használt) lehetőség, a zajszűrés. Ez egy rendkívül sokoldalúan hasznosítható modul, és igen jól működik.
Nem kell mást tennünk, mint kikeresni azt a részt, ahol kizárólag a zaj hallatszik. (általában a számok legeleje az, amikor még nem szól a zene, de a felvétel már megy.)
Ezt a részt kijelöljük, majd ***Generátor-> Csend*** menüpontot használva létrehozunk egy ugyanolyan hosszúságú részt, de most már a zaj nélkül. Ezután kijelöljük a teljes anyagot (a zajmentest is), majd a ***Hatás->Zajszűrő*** opcióval készítsünk zajprofilt. Ebből fogja tudni pontosan a program, hogy mit is kellene eltüntetnie. Ismét ***Hatás-> zajszűrés***, és válasszuk ki a z előnézet pontot. Hallgassuk meg, hogy milyen is lesz a végleges. Csak ezután kattintsunk a „zaj eltávolítása gombra”. Fontos, az előnézet használata, hisz ha nem vagyunk elégedettek az eredménnyel, a preview feletti csúszkán állítva még javíthatunk rajta.
A tempó- és sebességmodosítás akkor jön jól, ha például magnóról vettünk fel, amely lassaban csévél. Esetleg saját dallamokkal kísérletezünk, és kíváncsiak vagyunk arra, hogyan is hallatszik mindez gyorsabban. Ekkor használjuk a ***„Hatás-> Tempó”*** és a ***„Hatás->Sebesség”*** pontokat. Ha kizárólag a tempót szeretnénk módosítani, okos dolog, ha a ***„Hatás-> A tempó módosítása”*** pontot választjuk, ugyanis ekkor csak a tempó módosul, ellentétben az előzővel, ahol a hangmagasság is.
Ezzel mindössze a legfontosabb pár funkcióját ismertettem az audacity-nek. Nagyon profi még a visszhang effektje, a fel- és lekeverések, a sávok felcserélése, a mély kiemelése, és a hangszínnel végezhető műveletek is.
Nagyon kényelmes, hogy gond nélkül kezel közvetlenül is mp3, illetve .ogg tömörített állományokat, amivel sok időt takaríthatunk meg. (Figyelem! mp3 esetén külön be kell állítani, hogy melyik kódolóprogramot szeretnénk használni.)
Az audacity részletes bemutatása természetesen lehetetlen vállalkozás lenne, annyi mindent tud. Rendkívül impresszív, stabil, és gyors kis program, a nagyok tudásával felvértezve.
Elmondható róla, hogy csak annyit tud, amennyire szükség van, de azt nagyon. Nincsen több tonna gomb, és beállítópanel, kezelése kényelmes, határozottan „kézreáll” minden feladatban amire csak használni szeretnénk. Olykor (ilyen téren mindenképpen) tanulhatnának tőle a kereskedelemben kapható „nagyok”.
Mindenkinek ajánlom figyelmébe ezt a programot.
Külön köszönet Mucsina Norbert Jánosnak, aki bevezetett az audió feldolgozás rejtelmeibe, és sokat segített az audacity használata során.

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.

MPlayer – határtalan képességek

Ebben az írásban azt nézzük meg, miként érett igazi nagysággá a világ legnépszerűbb médialejátszója!

Bizonyára a tisztelt olvasók is emlékeznek a nagyjából két éve megjelent Mplayer-ről szóló cikkre, melyben egy szenvedésektől nem teljesen mentes folyamat bemutatásával megkíséreltem egy egyszerű telepítési és használati útmutató létrehozását és elterjesztését. A bonyolult telepítési leírás ellenére is rengeteg pozitív visszajelzést kaptam ami egyértelműen jelezte számomra, hogy igen fontos és sokakat érdeklő témáról van itt szó. Azóta eltelt nem kevés idő, és a programmal kapcsolatban természetesen sok minden megváltozott, nem csoda hát, ha az a bizonyos leírás sem állja már meg a helyét teljes mértékben. Úgy vélem, nem engedhetjük megmagunknak azt a mérhetetlen luxust, hogy a linux világában az egyik alappillérnek számító MPlayer-ről ne legyen a médiában használható leírás. Épp ezért szeretnék most ezen a helyzeten változtatni, s ismertetni újra a program alapvető működését, telepítésének módját, az azóta történt változtatásoknak megfelelően.

Ha valaki még nem hallott volna róla…

Az MPlayer egy Linux alatt működő, alapjaiban magyar fejlesztésű videolejátszó, amely azonban sok más rendszer alatt, például Unixon és akár a nem x86 processzorokon is fut. Alapvetően parancssoros lejátszónak indult, ebben rejlik igazi ereje is, ám az utóbbi változatokban már grafikusfelület használatára is lehetőség nyílik. Szinte az összes videofájltípust képes lejátszani. Ismeri a legtöbb MPEG-, VOB-, AVI-, VIVO-, ASF/WMV-, QT/MOV-, FLI-fájlt, amelyek mellé jó néhány beépített XAnim és Win32 kodeket sorakoztat fel. Nem találtam olyan hibákkal teli, sérült indexszel rendelkező fájlt, amelyet ne tudott volna mindenféle fennakadások nélkül megjeleníteni, és ez mindenképpen egy roppant előnye a programnak, hiszen igen sok bosszúságtól mentheti meg az egyszeri felhasználót.
VideoCD-vel, SVCD-vel és DVD-vel szintén könnyedén megbírkózik. A program másik figyelemre méltó tulajdonsága a megjelenítési módok széles választéka. Támogatja az X11-, Xv-, DGA-, OpenGL-, SVGAlib-, fbdev-, aalib-, DirectFb-módokat, amelyeket akár SDL-en vagy GGI-n keresztül is el tud érni – mindeközben a réteg nyújtotta lehetőségeket is kihasználja. Ezen felül támogatja az MPEG-kártyákkal történő dekódolást és megjelenítést, a tv-tunerről történő lejátszást. Támogatja az OSD-t is, melynek segítségével a filmekre simított, árnyékolt feliratokat helyezhetünk. Ismer jó néhány feliratformátumot, köztük egy saját fejlesztésűt is.
Nagyon fontos újítása az volt, hogy ki tudja használni a videokártyákba épített hardveres gyorsítást (az X-felületen keresztül), ami gyakorlatilag azt jelenti, hogy a képet át tudjuk méretezni úgy, hogy ez ne emésszen több processzoridőt, s közben a kimeneti kép is szép maradjon.

Egy kis áttekintés

Hogy mi is változott a ***Linuxvilág Magazin 14. számában*** leírt 0.6-os változat óta? Alapvetően az alkalmazás teljes egészében sokkal érettebb lett, kiforrottabb és letisztultabb. Ezt nem csak az összefogott, átgondolt, jól felépített szolgáltatások rendszere mutatja, de a telepítés is sokkal egyszerűbb lett. Mielőtt azonban túlzottan örülni kezdenénk, el kell mondanom, hogy a ritkán használt, különleges szolgáltatások igénybevételéhez, ritka fájlformátumok lejátszásához továbbra is „hegesztenünk" kell. Noha az esetek túlnyomó többségében erre nem lesz szükség, mégis érdemes tudni, hogy mire képes a lejátszónk, minek köszönheti már-már legendásnak mondható hírnevét.
Konkrétabban közelítve a kérdést: megváltozott a lejátszási kapcsolók rendszere. Bevezették a ***protokoll://cím*** stílusú állomány-meghatározást, átalakították az elérhető szolgáltatások (szűrők, stb.) megadásának módját. Aztán: új dvd-lejátszó kód került a forrásba, rengeteg új fájlformátumhoz született támogatás, beszédesebb lett a telepítő, s ki tudja még mi minden változott, lett hatékonyabb belül, a kódban. Alapvetően egyébként sokkal kevesebbet kell vesződnünk a telepítéssel, és a használattal, de lássuk most azt, hogy mennyi az annyi, tehát mit is jelent valójában ez a kicsi vesződés.

Hozzávalók

Az MPlayer leírásában fel van sorolva minden olyan csomag, azok verziói, amikre szükségünk lehet. Ezek manapság egy átlagos linuxos gépen alapértelmezetten telepítve vannak, ezért részletes ismertetésüktől most eltekintek, és csak nagy vonalakban szólok a legfontosabbakról:
`binutils`: a processzorban található magas szintű utasítások (MMX, 3DNow!) eléréséért felelős.

`gcc`: A C fordító, amelynek változatszáma kiemelten fontos tényező. Messze nem mindegy ugyanis, hogy melyiket használjuk. A fejlesztők a 2.95.3-2.95.4-es változatokat, vagy a 3.3 felettieket ajánlják, mert ezekkel működik a kód, a többi változat tele van programhibával.
`XFree86`: maga az X-felület, de ez ugye általában be van lőve a gépeken. Ahhoz azonban, hogy kihasználhassuk a kiszolgáló nyújtotta hardveres lehetőségeket, a fordítás előtt tegyük fel a fejlesztői csomagot. Ez Debian alatt az ***xlibs-dev***, Redhat/SuSE terjesztéseknél pedig az ***Xfree86-devel*** nevet viseli.
`make`: a rendszerfüggetlen fordítási és telepítési szkriptek futtatásáért felelős.
Ezen kívül szükség lesz még az MPlayer forrására, amely a ➲ http://www1.mplayerhq.hu/MPlayer/releases/MPlayer-1.0pre4.tar.bz2 címről tölthető le.

Indulhat a móka

Ha a fenti csomagok megvannak, akkor elkezdhetjük az „Mplayer-light" változat telepítését, amely azonban a legtöbb esetben tökéletesen elég is lesz. A későbbiekben látni fogjuk, miként okosítható tovább a lejátszó, de most azt szeretném bemutatni, hogy a program beüzemelése valóban ilyen egyszerű. Magam is csak egy kicsivel „okosabb" változatot használok, s tökéletesen elég minden feladatra.

1. lépés: csomagoljuk ki a letöltött állományt a ***/usr/src*** könyvtárba.
2. lépés: rendszergazdaként lépjünk be a ***/usr/src/MPlayer-1.0pre4*** könyvtárba, s adjuk ki a `./configure` parancsot. Ha ez rendben lefutott, a program utasít bennünket, hogy adjuk ki a `make` parancsot. A fordítás befejeztével (5-10 perc) ismét kiírja, hogy mi legyen a következő lépés. Adjuk ki a `make install` parancsot.

Ezennel az alkalmazásunk futtatásra kész, képesek vagyunk vele DVD-k, VCD-k, fájlok és weben lévő állományok lejátszására.

OSD

Innen már csak egy aprócska lépés annak megoldása, hogy a program lejátszás közben képes legyen az egyes műveletek nyugtázását és némi hasznos információt (lejátszás állapota, idő kijelzése) kiírni a képernyőre. Megjegyzendő, hogy a feliratok lejátszásához is ugyanez az OSD (On-Screen Display) használatos.
Mi kell tennünk? Töltsük le a ➲ http://www1.mplayerhq.hu/MPlayer/releases/fonts/font-arial-iso-8859-2.tar.bz2 állományt. Ebben négy betűméretben, négy különböző könyvtárban találjuk az MPlayer által használni kívánt betűtípusokat. Csomagoljuk ki valahová az állományt, válasszuk ki belőle a nekünk tetsző betűméretet (én a legnagyobbat szoktam), majd az adott alkönyvtár tartalmát (a könyvtár nélkül!) másoljuk át a ***/usr/local/share/mplayer/font*** könyvtárba. Indítsuk el az alkalmazást egy tetszőleges állománnyal: `mplayer <fájl neve>`, s nyomjuk meg például a numerikus billentyűzeten a „/" vagy a „*" gombot. Ezzel szabályozhatjuk a hangerőt, miközben egy skála (a TV-knél megszokott módon)

mutatja, hogy milyen hangos épp a műsor. Ezen kívül természetesen számtalan egyéb dolgot is kiír a program. Az „o" betű nyomkodásával például váltogathatjuk, hogy milyen információk jelenjenek meg a képernyőn, de erről majd később.

Hogyan tovább?

Tegyük fel, hogy szeretnénk néhány különleges funkciót is igénybe venni. Jó lenne például ***asf*** vagy más fájlokat lejátszani, szeretnénk, ha a filmhez tartozó feliratok a széles vásznú változatoknál a holttérben jelennének meg, stb. Mint az írásunk címéből is kiderül, ezen a területen sincs leküzdhetetlen akadály. Kezdetnek nézzük, hogyan is játsszuk le a ritka windowsos formátumokat.
Először is vessünk egy pillantást arra, hogy hogyan működnek az olyan lejátszók, mint amilyen az MPlayer is.
A különleges, tömörített formában tárolt filmeket a beolvasás után szét kell választani képre és hangra (demultiplexing). Ha ez megvan, akkor az egyes adatfolyamokból ki kell nyerni újra a képet és a hangot. Ennek elérése érdekében a lejátszók ún. kodekeket (kóder-dekóder) hívnak segítségül, melyek az adott tárolási formátumokra jellemzőek. Ezek olyan jószágok, amelynek az egyik oldalán odaadjuk a tömörített adatfolyamot, s a másik oldalán kijön a megjeleníthető kép vagy a hallgatható hang. A dolog természetesen visszafelé is működik, azaz az ellenkező úton a kodek tömöríteni is tud. Ezek után egy nehéz feladat keretében összehangolják a hangot és a képet, majd átadják az adatokat egy alacsonyabb szintre, hogy az a képernyőn megjelenhessen. Ragadjunk le egy kicsit azoknál a bizonyos kodekeknél. Nem kell ahhoz nagy fantázia, hogy lássuk, ezek bizony függetlenek az alkalmazástól. Ha tehát ismerjük a tömörítésre jellemző kodeket, akkor annak megléte lehetővé teszi számunkra, hogy az ilyen felépítésű lejátszók képesek legyenek az adott formátum megjelenítésére. Ésszerű lehetőség volna tehát segítségül hívni a ritka formátumokat lejátszani tudó kodekeket. Ezt nem is nehéz megoldani. Létezik ugyanis egy olyan kodekgyűjtemény, amely átalakított eredetileg Windows alá készített DLL-eket tartalmaz, s szolgáltatásait az MPlayer értelmezni tudja.
A gyűjteményt a ➲ http://www1.mplayerhq.hu/MPlayer/releases/codecs/essential-20040427.tar.bz2 állomány letöltése után érhetjük el. Tömörítsük ki a fájl tartalmát a ***/usr/lib/win32*** könyvtárba, még mielőtt az MPlayert lefordítanánk.

Abban az esetben, ha ezen már túl vagyunk, újra kell fordítani. Ehhez lépjünk be a forrás könyvtárába ***(/usr/src/MPlayer-1.0pre4)***, s adjuk ki a `./configure`, `make`, `make install` parancsokat ebben a sorrendben. A művelet végeztével szinte az összes létező fájlformátum lejátszására lehetőségünk nyílik.
Ezen túlmenően lehetőségünk van egyéb ismeretlen win32-es, valamint más, „alternatív" kodekek használatára is. Ennek igénybevételéhez javaslom a dokumentáció részletes elolvasását

Feliratok kontra SDL

A hatékonyabb és kényelmesebb lejátszás érdekében ajánlatos telepítenünk az SDL-t (Simple DirectMedia Layer). Ez egy olyan réteg a Linux X-felületében, amely annak kép- és hangkezelő részére közvetlenül ráépülve, egységes felületet biztosít és értéknövelt szolgáltatásokat nyújt az őt használó programok számára.
Előfordulhat, hogy bizonyos kompatibilitási hiányosságok következtében csak ezen keresztül érhetjük el, hogy lejátszónk helyesen működjön, de megvallom őszintén, én még ilyen helyzettel nem találkoztam.
Nekünk azonban most nem is ezért fontos ez a réteg. Ha ugyanis SDL-t használunk, teljes képernyős módban nem csak akkora lesz a filmünk, mint amilyen széles a vászon, hanem a lejátszó az egész képernyővel gazdálkodhat, ami által képes lesz a feliratot fizikailag is a képernyő aljához igazítani. Tehát szélesvásznú filmjeink esetében a feliratok nem lógnak bele a filmbe. Ráadásul a „c" betű nyomogatásával (teljes képernyős módban) változtathatunk az X-felület beállítási fájljában megadott felbontások között, és beállíthatjuk, hogy milyen fizikai felbontásban szeretnénk megnézni az adott filmet. Ez akkor lehet hasznos, ha például a kivetítőnk nem képes nagy felbontás megjelenítésére, vagy épp a filmünk kis felbontása miatt nagyon elhomályosodik a kép az erős felnagyítás következtében.
Az SDL a Linux terjesztés része, általában ***libsdl*** néven érhető el a csomaglistában, s fontos, hogy ebből is szükségünk van a fejlesztői (devel) változatra. Mielőtt tehát lefordítanánk, vagy újrafordítanánk az MPlayert, telepítsük az SDL könyvtárat, annak fejlesztői változatával együtt.
Újrafodrítás után a `-vo:sdl` kapcsolóval vehetjük igénybe a szolgáltatást.

GUI

Nos első pillantásra talán a grafikus felület változott a legtöbbet. Már nem kezdetleges, sőt, egészen jól használható, ezért is írom le, hogy mit kell tennük azért, hogy használni tudjuk grafikus felületünket.

Még valami: a használat során soha ne feledjük, hogy az elsődleges felület mindig is a parancssor marad, tehát a grafikus felületről csak a legfontosabb funkciókat érhetjük el.
A telepítéshez szükségünk lesz a ***libpng-dev*** és ***libgtk-dev*** csomagokra (egyes terjesztéseknél ezek a nevek eltérőek lehetnek). Ha ezt telepítettük, adjuk ki az MPlayer forrás könyvtárában a `./configure -enable-gui` kapcsolót.
Ezek után a `make` majd a `make install` parancsokat.
A fordítás végeztével azonban még nem működik a felület, ehhez ugyanis le kell tölteni egy kinézetet a ➲ http://www.mplayerhq.hu/homepage/design7/dload-hu.html címről, majd kitömöríteni a ***/usr/local/share/mplayer/Skin/default*** könyvtárba úgy, hogy ebben a könyvtárban már a fájlok legyenek benne, és ne további alkönyvtárak. Alapértelmezetten ugyanis a grafikus felület ebben a könyvtárban keresi a kinézetet.(Ha más könyvtárba raktuk, nem a default-ba, akkor a `-skin <könyvtár neve>` kapcsolóval választhatjuk ki indításkor). Az alkalmazás egyébként a `gmplayer` parancs kiadásával indul. Ha a program elindult, kinézettől függően láthatunk mindenféle érdekes kapcsolókat. Ezek közül leginkább a beállító-panel gombjára szeretném felhívni a figyelmet. Erre kattintva ugyanis grafikus felületről állíthatjuk be a főbb paramétereket, amelyek azonban a film újraindításáig nem lépnek érvényre.
A grafikus felület praktikus, jól használható, bár magam részéről továbbra is a parancssoros felhasználást javaslom.

A használatról

Ha már itt tartunk, lássuk kissé részletesebben a fentiekben telepített alkalmazás használatához szükséges kapcsolókat.
Az alapfelállás szerint a használat során az alábbi felépítésű parancsot kell kiadni: `mplayer <fájl(ok) neve (URL)> [kapcsolók]`. Ha egyszerre több fájlt, több bemenetet adunk meg, akkor azok lejátszási listaként működnek majd, azaz egymás után következnek a megadott elemek, van lehetőségünk véletlenszerű lejátszásra, stb. Ezen túl lehetőségünk van lejátszási listák létrehozására is.
A kapcsolók közül vannak egyszerűek és összetettek. Az egyszerű kapcsolók vagy érték nélküliek (jelzőbitek), vagy egyetlen értéksort használnak, összetett társaik pedig egy témakört összefogva több értékcsoport megadását várják az előre meghatározott formátumban (az értékcsoporton belül is több érték szerepelhet). Ilyen például a videoszűrőket jelképező kapcsoló. Mivel ez így egy kicsit homályosan hangzik, lássuk, hogy néz ki ez a gyakorlatban.

Egyszerű kapcsolók

`-vo <kimenet neve>`: ezzel a kapcsolóval határozhatjuk meg, hogy az mplayer által ismert számtalan kimenet közül

melyikre szeretnénk a képanyagot küldeni. A `-vo help` kapcsoló kilistázza a lehetséges értékeket. Az alapértelmezett eszköz az xv. Mint fentebb olvashatták, így vehető igénybe az SDL által nyújtott szolgáltatás is.

`-ao <kimenet neve>`: Ugyanazt tudja, mint a `-vo` kapcsoló, de az itt megadott jellemzők a hangkimenetre vonatkoznak

`-vc <kodek1 neve, kodek2 neve,>`: Mint tudjuk, az MPlayer egyszerre rengeteg kodeket kezel, s ezek közül a `codecs.conf` fájljában meghatározottak szerint választ. Előfordul, hogy mi nem az általa választottat szeretnénk, hanem egy másikat. Ezek közül sorrendben haladva választja ki az első alkalmasat.

`-ac <kodek1,kodek2>`: A fentiekkel egyenértékű, de ez jelen esetben a hang adatfolyamra vonatkozik.

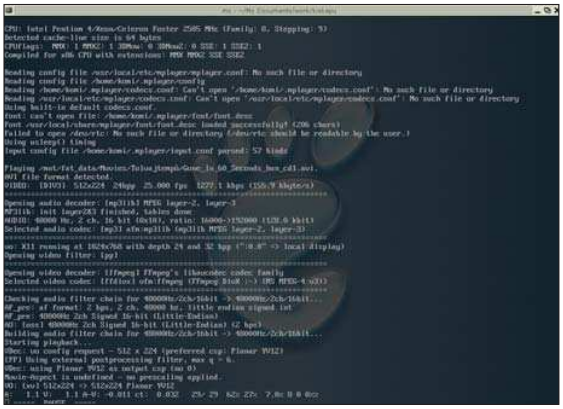
`-aspect <vízszintes:függőleges képarány>`: Ha nem megfelelő a képarány, a kapcsoló segítségével megváltoztathatjuk. (pl. `-aspect 4:3`)

`-playlist <fájl neve>`: Mint említettem, lehetőségünk van előre gyártott lejátszási listák használatára. Erre való ez a bizonyos kapcsoló. Ilyenkor bemenetet nem kötelező megadni, elég ezt az egyetlen kapcsolót használnunk. A program támogatja a Winamp ill. ASX formátumú lejátszási listákat, valamint egy sajátot is. Ez utóbbi rendkívül egyszerű: a fájl minden sora egyetlen lejátszandó elemet tartalmaz, ami akkor lehet hasznos, ha sok fájlunk van, amelyek egymás után következnek. Ekkor célszerű őket egy fájlba tenni, így a későbbiek folyamán nem kell vacakolnunk a parancssoros használat során. Legegyszerűbb, ha egy fájllista kimenetét beleirányítjuk ilyenkor egy fájlba, az alábbi módon:
`ls *.vob >play.list`

`-fs`: Ezt a kapcsolót hozzáfűzve a parancshoz az MPlayer teljes képernyős módban kezdi el lejátszani a filmet. Ennek akkor lehet előnye, ha például azt szeretnénk, hogy SDL kimenetet használva, teljes képernyős módra váltva ne váltson a program felbontást. Ilyenkor ha teljes képernyős módban indulunk, az X alapértelmezett felbontását fogja használni az SDL kimenet is. Ha ezt gyakran használjuk, célszerű a később bemutatott beállítási fájlban rögzíteni, mint alapállapotot. Ilyen esetekben azonban az fordulhat gyakran elő, hogy szeretnénk, ha időnként nem teljes képernyőben indulna. Meglepő módon ehhez a `-nofs` kapcsoló használatára van szükségünk. Általában igaz az MPlayerre, hogy minden jelzőbit jellegű kapcsolónak van ellentettje, amelyet az angol nem (no) tagadószóval kiegészített kapcsoló jelöl. (pl.: `-lirc --nolirc`; `-shuffle --noshuffle`, stb.)

Összetett kapcsolók

`-vf <szűrő1[=jellemző1:kapcsoló2],szűrő2,szűrő3>`
Ez a kapcsoló a képanyagra vonatkoztatott szűrőket foglalja össze, egy videoszűrő réteget alkotva. Ilyen szűrőből természetesen többet is megadhatunk, ezért ezeket egymás után fűzve az MPlayer az elejéről a vége felé haladva sorban alkalmazza a kitömörített képkockákra. Ezek a változások természetesen csak az adott lejátszásra érvényesek, nem érintik



a forrásanyagot. Az összetett kapcsolók használatával megszűntek az itt-ott kiadható, mindenféle nevű kapcsolók, a videoszűrőket csak ezen a lehetőségen keresztül tudjuk igénybe venni. Nézzünk néhány gyakran használt darabot:

`pp`: ez a szűrő a képek utófeldolgozását végzi. Sok esetben a kép zajos, mákos. Ez a szűrő gyakorlatilag egy képjavító algoritmus, amely a keletkező képeket javítja, elmossa. Nem keverendő össze a hardveres átméretezés során történő elmosási folyamattal. Ha nem adunk meg semmit, csak annyit, hogy `-vf pp`, akkor a processzoridő függvényében automatikusan az általa ítélt legjobb minőségű képszűrést alkalmazza.

`crop[=szélesség,magasság,x_pozíció,y_pozíció]`:
ez a lehetőség kivág egy területet a képből, és csak azt jeleníti meg. Mérete: szélesség x magasság, s az így keletkezett ablak bal felső sarkát az x_pozíció:y_pozícióra helyezi a képen. Ennek segítségével levágható a filmek szélén oly gyakran előforduló fekete keret.

`cropdetect`: mivel a fenti értékeket nehéz pontosan kiszámítani, ez a kapcsoló képkockánként kiírja a szabványos kimenetre, hogy szerinte mik a felesleges részek. A gyakorlatban ez úgy működik, hogy a kapcsoló figyeli, meddig fekete a kép széle, éppen ezért olyan képkockához kell tekerni, aminél maga a hasznos képanyag széle nem tartalmaz feketét, különben ezt is leszedi a program.

`flip`: megfordítja a képet fejjel lefelé.

`mirror`: a kép tükörképét jeleníti megadhatunk.

`-af <szűrő1[=jellemző1:kapcsoló2],szűrő2,szűrő3>`
Ez a kapcsoló a `-vf-hez` teljesen hasonlatos, csak épp a hanganyagra vonatkozik

Néhány érdekesebb hangszűrő

`volume[=erősítés]`: a kapcsoló segítségével változtathatjuk az hangsáv hangerejét, ezáltal segíthetünk a túl halk filmeken. (A dvd-k eredeti hangjai általában igen halkan szólnak.) Az erősítés értéke deciBell-ben értendő és -200-tól +40-ig terjedhet. Persze akármekkorát nem erősíthetünk rajta, hisz ez a jel torzulásával jár.

`resample[=mintavételi_frekvencia]`: a kapcsoló megadásával megváltoztathatjuk a lejátszott hanganyag mintavételezési frekvenciáját. Ez a lehetőség olyan esetekben lehet leginkább hasznos, amikor a filmünk 48KHz-es hangját a mi 44KHz-es hangkártyánk nehezen szólaltatja meg.

Az érthetőség kedvéért lássunk egy összetett példát: játszunk le egy merevlemezen található fájlt úgy, hogy tegyünk a képre elmosást, vágjuk le a szélét, hangosítsuk fel kicsit, és mindeközben használjunk SDL kimenetet, valamint mpeg12 kodeket. A szükséges parancs így fog kinézni:

```
mplayer test.mpeg -vc mpeg12 -vf crop=200:150:5:5,pp
➥ -af volume=10 -vo x11
```

DVD lejátszáshoz például a lemez behelyezése után a következő utasításra van szükségünk:

```
mplayer dvd://1 -alang hu -slang hu
```

Ez lejátssza a fő adatfolyamot (magát a filmet), magyar nyelven, magyar felirattal. Menü kezelésére sajnos nincs lehetőség. VCD lejátszása ugyanígy történik, csak ott `vcd://`-el kezdődik a lejátszandó parancs. Gyakori hiba szokott lenni, hogy a lejátszóprogram nem találja a DVD eszközt, s ezáltal nem tudja lejátszani a korongot. Ennek az az oka, hogy az MPlayer (vagyis az a könyvtár, amelyet a dvd-k lejátszásához használ) közvetlenül az eszközhöz hozzáférve végzi a dolgát, nem fájlszinten (ennek következtében nem is kell befűznünk (mount) a fájlrendszerbe). Az MPlayer számára az alapértelmezett eszköz a /dev/dvd. Ha ez létezik a rendszerünkben, nincs is szükségünk, hogy bármit csináljunk, a fenti parancs valóban működik. Ha azonban nincs /dev/dvd eszközünk, akkor két dolgot tehetünk. Egyik, hogy a -dvd-device kapcsolóval megadjuk neki, hogy melyik eszközhöz kell hozzáférnie, hogy elérje a DVD-korongot, a másik megoldás pedig az, hogy egy szimbolikus linket hozunk létre az adott eszközre. A leggyakrabban a rendszerben fellelhető CD-ROM/DVD meghajtókat a linux a */dev/cdrom* eszközzel azonosítja. Erre kell tehát mutasson az a bizonyos link, s így a DVD lejátszás nem ütközik többé akadályokba. Hasonló a helyzet a VCD/SVCD-k lejátszása terén is, ám itt a /dev/cdrom az alapértelmezett eszköz. Ha nekünk nem itt van a CD meghajtónk, vagy épp több is van, és mi most a másik meghatóban lévő lemezt akarjuk lejátszani, akkor a -cdrom-device kapcsolóval segíthetünk a problémán.

Továbbmenve: kedvenc autósműsorom, a TotalCar epizódjait a netről is el lehet érni és le lehet játszani. Mindössze azt kell megmondanunk, hogy melyik távoli fájlra van szükségünk:

```
mplayer http://dl.totalcar.hu/tc235_1.wmv
```

A fentieken kívül természetesen számtalan kapcsoló található még a programban, ezek tanulmányozásához a legjobb út a mellékelt magyar nyelvű használati utasítás részletes átolvasása.

Lehetőségek menet közben

Az Mplayer talán egyik legnagyobb vívmánya, hogy menet közben a billentyűzetről az összes vezérlő funkció elérhető. Hangosítás, halkítás, fényerő, kontraszt, pozicionálás háromféle ugrásnagyságban, stb. Ezen túl az OSD-nek köszönhetően természetesen mindegyik műveletről nyugtát is ad, tehát kiírja a képre, hogy mi történt az adott billentyű megnyomásának hatására.
Az alapértelmezett billentyűkódok:

```
balra, jobbra: előre-hátra teker pár másodpercet
PgUp,PgDown: előre-hátra teker 10 percet
+,-: késlelteti, vagy sietteti a hangot
100ms-onként
q: kilép
f: teljes képernyős mód
o: OSD állapotának változtatása
/,* halkítás, hangosítás
1,2; 3,4; 5,6; 7,8: hardveres gyorsítást
használva ezekkel a billentyűpárokkal állítható
a kontraszt, a fényerő, a színárnyalat illetve
a telítettség
```

Ezek a beállítások egyébként a */usr/local/etc/mplayer/ input.conf* fájlban módosíthatók, testreszabhatók. Az eredeti fájl egyébként a forráskönytár *etc* alkönyvtárában található. S ha már itt tartunk, tegyünk egy kis kitérőt.
Az Mplayer nagyjából pontosan úgy működik, mint bármelyik más linuxos alkalmazás. Van globális beállítása, amely a */usr/local/* könyvtáron belül helyezkedik el szétszórva, s van felhasználófüggő, amely a *sajátkönyvtár .mplayer* könyvtárán belül található. Ide helyezhetünk betűtípusokat, kinézeteket, *input.conf* fájlt, s alapértelmezetten létrehozza a telepítés a *.config* fájlt, amelyben `kapcsol=érték` formában megadhatjuk azokat a parancssori kapcsolókat, amelyeket mindig használunk. Tehát ha azt szeretnénk, hogy például mindig teljes képernyőn induljon a program, akkor írjuk bele az alábbi sort: `fs=1`. (Az értékkel nem rendelkező, ún. bináris kapcsolókat 0 és 1 számokkal engedélyezhetjük illetve tilthatjuk)

Összegzés

A lejátszó lehetőségeit bemutatni egy ekkora cikkben szinte teljességgel lehetetlen. A fent célja mindössze annyi volt, hogy a kedves olvasó kedvet kapjon a dolgokban való elmélyedéshez. Azonban ha ez a kedv mégsem jönne meg, az itt leírtak alapján akkor is hatékonyan és kényelmesen tudjuk használni a programot. Aki erre vágyik, annak tényleg elég a cikk elején leírtakat elvégezni, s rögtön ugorhat is iromány végére, ahol a használatról esik szó. Remélem, mindenki kellőképp eligazodik majd a röviden leírt módszerek között, s megtalálja számítását az MPlayer használatával.

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

Számítógépes hálózatok (4.rész)

Adatkapcsolati réteg, keretezés, forgalomirányítás és szabályozás

Belekezdünk az adatkapcsolati réteggel történő mélyebb ismerkedésbe. Lesz szó keretekről, forgalomszabályozásról de még hibajavításról is.

Immár hetedik alkalommal jelentkezik számítógép-hálózatokat boncolgató sorozatunk, amelyet a gyakorlati jelentőséggel nem igazán bíró, ám az elmélet megértéséhez sokat segítő OSI modellre építünk. Idáig sok témát érintettünk már, de részletesen csak azzal a fizikai réteggel foglalkoztunk, amely lehetővé teszi, hogy két „szomszédos" számítógép információkat cseréljen. Ezek az információk bitsorozatok, amelyeket a hálózati réteg vagy elektromágneses hullámok vagy pedig a vezeték bizonyos fizikai jellemzőinek megváltoztatásával továbbít. Miután a hálózati réteg „ráhelyezte" a bitsorozatot az átviteli csatornára, feladata végetér. Ezzel azonban csak félmunkát végzünk, a „szomszédos" számítógépek közötti adatcsere még sincs megoldva. A fizikai réteg ugyanis semmilyen módon nem garantálja azt, hogy a neki átadott bitsorozat sértetlenül megérkezzék (már ha egyáltalán megérkezik) a célhoz. Az átviteli csatornák pedig nem tökéletesek, könnyen előfordulhat, hogy egy-egy bit elveszik útközben, de az is, hogy több érkezik meg, mint amennyit elküldtünk. Ha esetleg a megérkezett bitek számban megegyeznek az elküldöttekkel, akkor pedig értékeikben különbözhetnek.

A hálózati réteg – amely már azon fáradozik, hogy a hálózatban ne csak az egymással közvetlenül összekapcsolt gépek kommunikálhassanak, hanem a hálózat bármely pontjáról bármely pontjára tudjunk csomagot küldeni – helyes működésének alapvető előfeltétele, hogy az adatátvitel többnyire hibamentes legyen, illetve hogy ha mégis hiba történik, akkor az érzékelhető legyen. Ezért van szükség az adatkapcsolati rétegre, amelynek legfontosabb feladata, hogy felismerje az átviteli közeg tökéletlenségéből adódó hibákat, amelyeket ezután lehetőség szerint megpróbál kijavítani.

Keretezés

Az imént említett feladatra a legcélszerűbb megoldás az, ha a bitfolyamot az adatkapcsolati réteg különálló részekre, úgynevezett keretekre(frames) tördeli, és ezekhez hozzácsatol egy úgynevezett ellenőrző összeget (checksum). Az ellenőrző összeg alapján a legtöbb esetben ki lehet szűrni az átvitelkor keletkezett hibákat. A keret megérkezésekor ezt a számot ismét kiszámolja a rendszer, és ha az nem egyezik meg az eredetivel, akkor átviteli hiba történt, amit jelezni kell a küldő felé.

Ez első hallásra nem tűnik valami nagy kunsztnak, de mindjárt kiderül, hogy egyáltalán nem valami triviális feladat. A dolgot ugyanis megnehezíti, hogy a célállomásnak el kell tudnia különíteni egymástól a beérkező kereteket, azaz tudnia kell, hogy egy keret mikor ér véget.

Erre a problémára a legegyszerűbb megoldás az lehet, hogy amikor a forrás elküldte az első keretet, egy meghatározott ideig beszünteti az adást, azaz szünetet tart. Mikor ez a kis idő lejár, elindítja a következő keretet. A hálózatokról azonban tudnunk kell, hogy minden olyan feladat, ahol az időzítés szerepet játszik, a legtöbb esetben kivitelezhetetlen. Ezek a szünetek ugyanis megnyúlhatnak, esetleg lerövidülhetnek, szélsőséges esetekben el is tűnhetnek. A fogadó így képtelen lesz teljes bizonyossággal felismerni a kerethatárokat, ezért más megoldást kell kitalálni.

Egyik alternatíva lehet a keretszámlálás néven elhíresült módszer, amikor minden keret a keretben található karakterek számával kezdődik. Ha például a keret első karaktere egy 8-as, akkor a fogadó tudni fogja, hogy a nyolcadik karakter után érkező bitek már egy új kerethez tartoznak. Ezzel a megoldással az a probléma, hogy a kommunikációban részt vevő két fél nagyon könnyen kieshet a szinkronból. Vegyük például azt az esetet, amikor egy átviteli hiba megrongálja a keret első karakterét, és így a célhoz nem egy 8-as, hanem egy 12-es érkezik. Ennek következtében a következő tizenkét karaktert ehhez a kerethez sorolja, miközben az utolsó négy karakter már egy másik keretnek a része. Persze ez a hiba egyből kibukik az ellenőrző szám alapján elvégzett hibakereséskor. Ez azonban nem sokat számít, a fogadó ugyanis ekkor sem fogja tudni, hol kezdődött a következő keret. Az sem segít, ha a forrás megismétli a hibás keretet, hiszen a cél nem tudja, hány karaktert lépjen vissza ahhoz, hogy a kicserélendő rossz keret valódi kezdetére érkezzen. Ez a módszer sem igazán nyerő tehát.

A keretszámlálás legnagyobb buktatója, hogy nem nyújt lehetőséget a szinkron újbóli felvételére, azaz arra, hogy a vevő valamilyen módon ismét képben legyen a kerethatárokat illetően.

A következő ismertetendő módszer erre már lehetőséget nyújt. Az alapötlet, hogy a keretek kezdetét illetve végét egy-egy speciális karaktersorozat jelölje. A keret kezdetét jelölje például a DLE STX nevű (DLE: ***Date Link Escape***, STX: ***Start TeXt***), a végét pedig a DLE ETX (ETX: ***End TeXt***) nevű két karakterből álló sorozat. Ha mondjuk egy átviteli hiba folytán valamelyik kerethatárt jelölő karakter megsérül, ak-

kor sem menthetetlen a helyzet. A cél ismét szinkronba kerülhet a forrással, ha figyeli a DLE STX és DLE ETX karaktereket.
Ez a módszer csak addig működőképes, amíg nem próbálunk bináris adatokat küldeni (például programkódot, képet, videót, lebegőpontos számot, stb.). Ezek az adatok ugyanis könnyen tartalmazhatnak DLE karaktert, amely speciális jelentéssel bír a célállomás számára, és így megzavarhatja a kommunikációt. A megoldás az, hogy valamiképpen jelölni kell, hogy az adott DLE karakter csak egy, az adatban szereplő karakter, vagy valóban a kerethatár kezdetét jelzi. A leggyakoribb megoldás ilyenkor az, hogy az adatkapcsolati protokoll a szövegben „véletlenül" előforduló DLE karaktereket mindig megduplázza (kétszer küldi el). Ez a módszer már ígéretesnek tűnik, csak az a baj, hogy szorosan összefügg a 8 bites ASCII karakterkódolással. Igaz, hogy ez egy nagyon elterjedt kódolás (főleg a PC-s világban), azonban mégsem várhatjuk el, hogy minden számítógép 8 biten, ezen belül az ASCII kódolás segítségével képezze le a karaktereket. Ha azt szeretnénk, hogy hálózatunknak mindenféle színű és szagú számítógépek is tagjai lehessenek, akkor egy olyan keretező algoritmusra van szükség, amely független a karakterek méretétől és kódolásától.
A megoldás hasonló lesz az előbbihez, azzal a különbséggel, hogy a keretek kezdetét nem karakterek, hanem egy speciális bitminta jelzi, mégpedig a 01111110. Ha az adatban találunk öt egymás követő egyest, akkor az adatkapcsolati réteg automatikusan egy 0-t szúr be, így ismét egyértelműen meghatározhatóak lesznek a kerethatárok.
Ez már egy jól használható keretezési eljárás, a gyakorlatban azonban mindig a keretszámlálást alkalmazzák valamilyen másik módszerrel együtt. Ezért a vevő egy keretet csak akkor fogad el érvényesnek, ha egyrészt a hibaellenőrző kód alapján helyes, másrészt például a határolójelek is a helyükön vannak.

Nyugtázás

A keretezésnek köszönhetően a vevő értesülhet arról, hogy történt-e átviteli hiba, vagy sem. A forrás azonban semmit sem tud az általa elküldött keretek sorsáról. Ez persze csak akkor jelent gondot, ha összeköttetés alapú szolgálatot kell teljesíteni.
Hogy a kommunikáció biztonságos legyen, kell valamiféle lehetőség arra, hogy a vevő értesíthesse a forrást az általa fogadott adatok állapotáról. Például a forrásnak vissza kell küldeni egy nyugtát, amelyben megerősíti, hogy az adatok rendben megérkeztek, vagy épp azt kéri, hogy ismételje meg az adást.
Vannak bizonyos szerencsétlen körülmények, amelyekkel számolnunk kell. Például mi a helyzet akkor, ha egy keret teljes egészében elveszik? (Ilyen előfordulhat, például amikor nagyon zajos a vonal). A keretek elvesztése rendkívül kellemetlen dolog, mert a forrás várja a nyugtát, a vevő azonban nem is tud arról, hogy neki keretet küldtek, amelyre egy nyugtával kéne válaszolnia. Ezt a problémát persze könnyen kiköszöbülhetjük úgy, hogy a forrás minden kerethez egy időzítőt is rendel. Amikor a keret küldésétől számítva eltelik egy bizonyos idő, és nem érkezik nyugta, akkor valószínűsíthető, hogy a keret nem érkezett meg a célhoz. Ilyenkor a forrás felhagy a várakozással, és megismétli a kérdéses keretet.
Egy keret megismétlése azonban nem veszélytelen dolog, ugyanis fennáll az eshetősége, hogy a vevő kétszer kapja meg ugyanazt a keretet, így kétszer továbbítja azt a hálózati réteg felé. Ezért szükség van keretazonosítók bevezetésére is, például sorszámokra, amelyekkel meg tudjuk számozni az elküldött kereteket.
Az időzítők és a keretek sorszámozásával az adatkapcsolati réteg biztosítani tudja, hogy minden pontosan egyszer (nem kétszer és nem is nullaszor) érkezzen meg a fogadóhoz. A gyakorlatban ez egy kicsit bonyolultabban zajlik a fent leírtaknál. A következő részben részletesen is megnézzük majd, milyen további problémákkal szembesülhetünk a való életben.

Forgalomszabályozás

Egy keretet elveszteni nem csak a zajos átviteli csatorna miatt lehet. Előfordulhat olyan eset is, hogy a küldő sokkal gyorsabban küldi a kereteket, mint ahogy azt a vevő kényelmesen fogadni tudná. Például azért, mert a vevő eleve lassabb, vagy éppen szörnyen leterhelt. Ha ez a helyzet, akkor a forrás teljesen elárasztja a célállomást, amely előbb-utóbb nem lesz képes megfelelően fogadni a kereteket, és egészen biztos, hogy egyet-kettőt el fog veszíteni. Nagyon szomorú, ha ilyesmi megtörténik, ezért szükség van valamiféle védelmi mechanizmusra, amely megakadályozza az ilyen helyzetek kialakulását.
A megoldás kulcsa a vevő visszafogásában rejlik, azaz nem szabad hagyni, hogy az gyorsabban adjon, mint ahogyan azt a célállomás venni képes. Ez az úgynevezett forgalomszabályozás (flow control). Sokféle forgalomszabályozási módszer ismert (a következő részekben részletesen is megnézünk néhányat), de a legtöbb ugyanazt az alapelvet követi: addig nem küldik el a következő keretet, amíg a vevő valamilyen módon engedélyt nem ad rá. Ez nem feltétlenül jelenti azt, hogy a vevőnek minden egyes keret után szólni kell, hogy jöhet a következő. Lehet ugyanis, hogy a kapcsolat felépítésekor megmondja, hogy most például x darab keretet máris lehet küldeni, de utána addig semmit, amíg nem jelez vissza. A keret küldését persze ezen kívül más, az adott protokolltól függő szabályok is befolyásolják. A továbbiakban erről még szólunk részletesebben is.

Hibaészlelés és javítás

A biztonságos kommunikációhoz önmagában nem elég, hogy minden keret pontosan egyszer érkezik meg, illetve hogy a forrás nem küld gyorsabban, mint ahogy a vevő fogadni képes. Legalább ennyire fontos, hogy pontosan az érkezzen meg, amit átküldünk, azaz a bitek értéke ne változzon.
Hogy mikor és mennyi átviteli hiba keletkezik, abban szerepet játszik az is, hogy milyen technikával továbbítjuk a biteket. Például a rádióadók esetében a hibák inkább csoportosan fordulnak elő. Más rendszereknél pedig csak egy-egy bit romlik el, viszont időben egymástól függetlenül.
Mindkét esetnek van jó és rossz oldala is. Ha a hibák csoportosan jönnek, az abból a szempontból jó, hogy nem lesz minden elküldött keretünkben hiba, így mondjuk átlagosan csak minden 100. keretet kell újraküldeni. Hátránya azonban, hogy ha hiba van, akkor az valószínűleg elég sok bitet érint a keretben ahhoz, hogy nehéz legyen kijavítani, esetleg magát a hibát felfedezni.

Akárhogyan is legyen, a fellépő hibákat kezelni, sőt javítani kell. A hibák kezelésére két alapvető módszer létezik. Az első esetben annyi, úgynevezett rendundáns információt fűzünk az adatbitekhez, amennyiből a forrás fel tudja fedezni az esetleges hibát, és ki is tudja következtetni, hogy mi lehetett az eredeti szöveg. Ehhez úgynevezett hibajavító kódok lehetnek a segítségünkre. A másik stratégia szerint csak annyi redundáns bitet fűzzünk az adathoz, amennyiből meg tudjuk állapítani, hogy történt-e hiba, vagy sem. Ehhez hibajelző kódokat kell alkalmaznunk. (Fontos, hogy ebben az esetben csak a hiba meglétére tudunk következtetni, arra nem, hogy melyik bit hibás).

Hogy megérthessük, miként működnek a hibajavító és hibajelző kódok, fontos definiálni azt, hogy „matematikailag" mit is nevezünk átviteli hibának.

A Hamming-távolság

A keretben szereplő adatbitek számát jelöljük m-el, a hozzáfűzött redundánsbitek számát pedig r-el. Az n = r + m hosszú adat- és redundáns bitekből álló bitsorozatot kódszónak nevezzük.

Két kódszót úgy tudunk összehasonlítani egymással, hogy bitenként végigmegyünk rajtuk, és minden azonos helyen lévő bit között elvégzünk egy kizáró vagy (XOR) műveletet. Ha a művelet végeredménye 1, akkor a két bit különböző. Azt, hogy a két kódszó összehasonlításakor hányszor kapunk 1-est, azaz hány eltérő bit van, a két kódszó ***Hamming távolságának*** nevezzük. Ha más szemszögből közelítjük meg a dolgot, akkor akár azt is mondhatnánk, hogy két kódszó Hamming távolsága azt határozza meg, hogy hány darab egybites hiba kell ahhoz, hogy az egyik kódszóból megkapjuk a másikat.

Mivel az adat a bitek bármilyen „kombinációjából" állhat, összesen 2^m-ediken féle úgymond legális adatbit sorozat szerepelhet egy keretben. A redundáns bitek számítási módjából következik, hogy nem fordulhat elő mind a 2^r darab, azaz a lehetséges kódszavak száma nem éri el a 2^n-t. Ha tudjuk a redundáns bitek kiszámításának módját, akkor könnyedén elkészíthetjük a legális kódszavak listáját. Ha megvan a lista, akkor abból kikereshetjük azt a két kódszót, amely a legkisebb Hamming távolsággal rendelkezik. Ez az érték lesz a kód Hamming távolsága.

Hogy egy kód hibajavító, vagy csak hibajelző, azt a Hamming távolsága határozza meg. Ha például fel szeretnénk ismerni d bithibát, akkor legalább egy d + 1 Hamming távolságú kódot kell alkalmaznunk, hiszen egy ilyen kódban hiába fordul elő d bithiba, akkor sem tudunk megkapni egy másik érvényes kódszót.

A hibajavító kódnak azonban nagyobb Hamming távolsággal kell rendelkeznie, pontos értéke (2d + 1), ha ki is szeretnénk javítani legaláb d darab bithibát. Az ekkora távolságú Hamming kódoknál az érvényes kódszavak egymástól olyan messze vannak, hogy d darab bit megváltoztatása sem viheti egy másik legális kódszóba.

Ezt csak elmesélni bonyolult, valójában nagyon egyszerű dologról van szó. Nézzünk meg egy-egy példát a hibajelző, illetve a hibajavító kódra. A legegyszerűbb hibajelző kód a paritásbit, amely azt mondja meg, hogy az adattagban lévő 1-esek száma páros, vagy páratlan. (Ebben az esetben a redundáns információ maga a paritásbit). Ha egy bithiba történik, akkor megváltozik a paritás, így azt ezzel a módszerrel észlelni tudjuk. A paritás bites kód Hamming távolsága 2, mivel egy bit hibát észlelhetünk. (Például két bithibára ezzel a módszerrel nem derülne fény).

Nézzünk egy példát a hibajavító kódra is. Példánk alanya egy olyan kód, amely a következő legális kódszavakat fogadja el: 0000000000, 0000011111, 1111100000, 1111111111. Ennek a kódnak a Hamming távolsága minden bizonnyal 5, mivel kétbitnyi hibát tud kijavítani. Ha például a 0000000111 kódszó érkezik, akkor ebből ki tudjuk következtetni, hogy az eredeti üzenet a 0000011111. Ha azonban három vagy több bithiba keletkezik, például a csupa nulla helyett szintén a 0000000111-t kapjuk, akkor nem a helyes kódszót állítjuk vissza.

A gyakorlatban is használt hibajavító kódok közül a Hamming kódot emelnénk ki. Működésének ismertetése túlmutat e sorozat keretein, annyit jegyeznénk meg róla csupán, hogy egybites hibák kijavítására szolgál, de némi trükközés segítségével képessé tehetjük csoportos hibák javítására is. A hibajavító kódokkal a való életben ritkán találkozhatunk. Ennek az az oka, hogy sokkal gazdaságtalanabb, mintha csak hibajelző kódot használnánk, és ha szükséges, újra küldenénk a hibás keretet. A hibajavító kódoknak egyirányú kommunikáció esetén van jelentősége, akkor ugyanis a vevő nem tud visszaszólni a forrásnak.

Hibajelző kódok

A hibajelző kódok közül már volt szó a paritásbites kódról, amelynek nagy hátránya, hogy csak páratlan számú hibáknál bukik ki a probléma, azaz egy hiba jelzésének az esélye 50%. Ez nem egy jó arány. Sokat segíthet a dolgon, ha az elküldendő blokkot egy n X m -es mátrixnak tekintjük, és minden oszlophoz külön kiszámoljuk a paritásbitet. Ezután a mátrixhoz csatoljuk utolsó sorként a paritásbiteket, majd sorfolytonosan továbbítjuk az egész mátrixot.

Ez már egy némileg elfogadható megoldás, a gyakorlatban mégis inkább a polinomkódot, vagy ismertebb nevén a CRC kódot választják. A CRC kód működésének ismertetése is túlmutat e sorozat keretein, csupán annyit jegyeznénk meg róla, hogy nagyon pici a lehetősége a hibák rejtvemaradásának. A CRC16 például 16 bites ellenőrző összeggel dolgozik, így felismer minden egy- és kétbites hibát, továbbá minden páratlan számú bitet érintő hibát és minden 16, vagy annál kevesebb bitnyi csoportos hibát. (A csoportos hiba alatt azt értjük, hogy nem feltétlenül rossz az összes bit, de az első és az utolsó biztosan az).

A CRC-16 a 17 bites csoportos hibák 99,997%-át is felismeri. A CRC kicsit bonyolult algoritmus (főleg, hogy sok polinomosztást kell végrehajtanunk), mégis könnyű készíteni olyan áramkört, amely képes kiszámolni az ellenőrző összeget. Ezért általában hardveresen végzi a CRC-n alapuló ellenőrzést.

A következő részben tovább folytatjuk az adatkapcsolati réteggel való ismerkedést, méghozzá elemi adatkapcsolati protokollokat veszünk majd szemügyre.

Garzó András
garzo@interware.hu

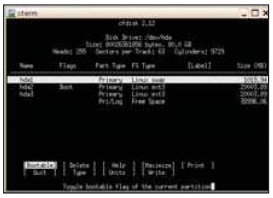
Debian otthonra (8. rész)

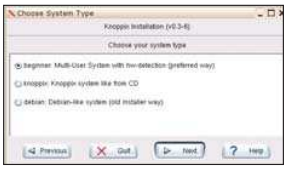
Lentről fel, vagy fentről le?

Hogy miért került az alcímbe az egyik kedvenc Holló Színház-előadásomból idézett mondat? A darabban a világhírű, már-már tökéletes Besenyő Pista bácsi bemutatja a világnak Boborjánt, aki a nagyközönség előtt beszél urduul. Boborján nyilvánvalóan nem a hagyományos úton tanulta meg ezt a világnyelvet, mégis használta. Nos, ebben a tekintetben akár vonhatunk is egy perverz párhuzamot a rendszergazdák és Boborján világa között. Kétféle rendszergazda van ugyanis: az első, aki hosszú-hosszú ideig szötymög a gépekkel, majd szép lassan egyre komolyabb feladatokat is el tud végeztetni velük, a másik pedig, akihez benyit a főnök, és azt mondja: „Mostantól Linux rendszergazda vagy és holnapra telepítesz egy fájl- és nyomtatókiszolgálót!”. Ez utóbbi pislog, mint Boborján. Mégis, valahogy meg kell felelni a kihívásnak.
A sorozat eddigi részeiben területenként és elemenként megismerkedtünk a Debian alapjaival, elméleti (valamint, esetenként gyakorlati) tapasztalatokat gyűjtve. A cikkeket végigolvasva remélhetőleg kialakult egy átfogó kép arról, hogy hol, mit kell beállítani, felépíteni. Így tehát néhány hét gyakorlás után akár mi magunk is összerakhatunk egy remek kiszolgálót, egyesével végigjárva a rejtett buktatókat és végignyálazva a leírásokat (amennyiben használhatom ezt a szófordulatot a README fájlok olvasására). De a lusta rendszergazdajelölt gyakran elábrándozik arról, „milyen szép is volna, ha ezt a bosszantó telepítgetést valaki elvégezte volna helyettem”. Remek gondolat! Nem kell egyesével feltelepítgetni a csomagokat, beállítgatni a szolgáltatásokat, kiválogatni a szükséges csomagokat... egyszóval minden eleve működne. A baj csak az, hogy ezek az „előre elkészített” rendszerek többnyire rpm-alapúak, és mi, hithű .deb-, nyílt és szabadforrású-rajongók, ragaszkodunk a kereskedelem-mentességhez, nem kívánunk frissítéseket vásárolni, és így tovább.
Megoldás természetesen van, sőt, nem is messze! Több Debian-alapú terjesztést készítettek, amelyet a terjesztés karbantartója ellátott minden jóval. Az egyik legjobb ilyen rendszer az Linuxvilág előző számának lemezmellékletén is megtalálható Knoppix. Keressük elő gyorsan az előző szám mellékletét és szimatoljunk körbe egy kicsit, mi is a helyzet ezzel a rendszerrel!

Knoppix – az első találkozás

A Knoppix 3.4 egy úgynevezett élő lemez (vagy mentőrendszer), ami annyit jelent, hogy a telepítőlemezről vígan elindul és három fő célra használhatjuk. Az első a helyreállítás: a program összeomlott rendszerek felett feléled a merevlemez használata nélkül, majd vígan körbenézhetünk róla, mi is volt a baj. A második cél a Linux kipróbálása: bármelyik gépen futtatható és nagyon sok olyan programot próbálhatunk ki, mely az áttérés esetén fontos lehet a felhasználó számára. A harmadik pedig, hogy kényelmesen telepíthessünk a gépre egy teljesértékű rendszert. Azt viszont nem árt tudni, hogy a rendszert nem magyarok állították össze, és ezért – noha törekedtek a lehető legtöbb nyelv támogatására – sajnos a magyar nyelv támogatása bizony elég hiányosnak nevezhető.





A telepítés első lépései

Az elmúlt napokban feladatom volt, hogy több munkagépre telepítsek Linuxot. A gépek mindössze annyiban hasonlítanak egymásra, hogy valamilyen Intel Pentium processzorra épült rendszerek voltak. Mivel ezek elsősorban irodai alkalmazásra készülnek, kell rajtuk lennie egy irodai programcsomagnak, kell tudniuk nyomtatni, levelezni, fájlmegosztásokat használni – egyáltalán: részt kell venniük a hálózati munkában. Szerencsés helyzetben voltam, hiszen a hálózat már létezett, így a központi kiszolgálóval nem kellett sokat szenvedni (biztosítva volt a fájlmegosztás samba-kiszolgálóval, egy DNS-kiszolgáló, volt egy DHCP-kiszolgáló is, valamint a levelezés egy IMAP-kapcsolaton keresztül működik).
Akárhogyan is, az embernek nincs kedve egyesével az összes gépet telepíteni. Hogy ezt a feladatot megoldjuk, arra több lehetőség is kínálkozik. Az első, hogy az egyik gépen feltelepítünk egy teljes rendszert, majd másolgatjuk a többire. Ez bizony eltarthat egy-két napig. Utána több lehetőségünk van, készíthetünk akár egy „telepítőlemezt”, ha elég kicsi a rendszerünk: a Linux képes arra, hogy a feltelepített teljes rendszert felmásolja egy korongra, majd erről tudjuk „visszamásolni” a rendszert a többi gépre. A második lehetőségünk ennek egy változata, amikor is a telepített csomagok

listáját nyerjük ki, majd a többi gépen is szép sorjában végigtelepítgetjük azokat. Ennél a változatnál készüljünk fel arra, hogy ötvennyolcszor kell kiválasztanunk az alapbeállításokat, megadnunk az időzónát és így tovább. A harmadik, általam is választott megoldás pedig egy élő rendszer használata majd testreszabása. Ennél a változatnál egy előre elkészített terjesztést „a lehető legegyszerűbb" módon telepítünk, majd elvégezzük a hátralevő testreszabást.
A feladatot, mint mondtam, a Knoppixszal és egy kis testreszabással oldottam meg. Szerencsére a Knoppix által használt önműködő gépfelismerő és önbeállító héjprogramok remekül működnek és csak itt-ott kell belenyúlnom a folyamatba. A negyedik telepítési kísérlet után pedig már könnyedén el tudtam készíteni egy saját héjprogramot, mely az egész testreszabási folyamatot elvégzi helyettem. Nézzük most meg, hogyan lehet könnyen és gyorsan egy teljesértékű irodai rendszert felépíteni a Knoppixból.

A telepítés menete

A telepítés első fontos lépése a lemezrészek létrehozása. A Knoppix telepítéséhez egy darab, legalább 2,5 GB méretű lemezrész kell, és javasolt egy csereterület használata is. Ha még nem hoztuk létre a lemezrészeket, akkor rendszergazdaként a `cfdisk` segítségével tudjuk ezt megtenni. Nálam tehát két lemezrészre volt szükség:

```
/dev/hda1      Primary Linux swap (82) 1020 MB
/dev/hda2 Boot Primary Linux      (83) 3060 MB
```

Ez után, biztos ami biztos, indítsuk újra a gépet, hogy az automatikusan felismert lemezrészek adatai frissüljenek (például a */etc/fstab* fájlban). Ahogy tapasztaltam, ez az újraindítás főleg akkor fontos, ha a merevlemezen előtte már léteztek `hda1` és `hda2` lemezrészek, de más adatokkal.
A második indítás után a lemezrészeket mi magunk is formázhatjuk (`mkswap /dev/hda1`; `mkfs.ext3 /dev/hda2`), majd indítsuk el a `knoppix-installer` programot. Ez a telepítőprogram könnyen megválaszolható kérdéseket tesz fel nekünk (részemről a Beginner telepítési módot javaslom, a Knoppix remekül előtelepített rendszer, így az alapbeállítások szinte mindig jók). Szinte csak a gyökérrészt (a „/" lemezrészt) és a `lilo` telepítési célját kell megadjuk (ez utóbbi, ha a Knoppix egyedüli rendszer lesz a merevlemezen, nyugodtan legyen a `mbr`), majd jelszót választanunk a rendszergazda és a saját felhasználónk számára és már indul is a telepítés. Tíz perces másolás után a telepítőprogram felkészíti a gépet az önálló indulásra, majd egy rendszerindító hajlékonylemezt is készíthetünk vele. Ezek után már egy élő rendszerünk van, ami önálló működésre képes. De ami a legjobb része a dolognak: eddig nem volt szükség hálózati kapcsolatra, letöltésekre sem!

Ungarische Knoppix

Nézzük meg, hogy ez a telepítés miben nem elégíti ki igényeinket. Ahogy írtam, a Knoppix saját héjprogramjaival „körbenéz", majd legjobb tudása szerint építi fel a rendszert. Sajnos ez a „legjobb tudás" nem tartalmazza a magyar beállításokat. Van egy-két beállítás, amit mi magunk kell hogy megtegyünk. Ezek közül több (remélem) bekerül majd az új Knoppixba, de addig is: magad uram, ha szolgád nincsen! A beállítások legtöbbjét megtehetjük már most, mielőtt a telepített rendszert a merevlemezről indítjuk, vagy – ez a kényelmesebb megoldás – újraindítás után. Ha újraindítás nélkül szeretnénk dolgozni a rendszeren, akkor be kell fűznünk a lemezrészt, és minden leírt változtatást ezen belül kell elvégeznünk (tehát amikor */etc/X11/XF86Config-4* fájlról beszélek, akkor az valójában a */mnt/hda2/etc/X11/XF86Config-4* fájlt jelenti):

```
mount /mnt/hda2
cd /mnt/hda2
```

Keressük meg a telepített rendszerben az alapbeállításokat létrehozó héjprogramot, és mondjuk meg neki, hogy magyarul szeretnénk beszélni. Ehhez a */etc/init.d/knoppix-autoconfig* fájlt kell szerkesztenünk. Az alaptelepítésben sajnos nem szerepel nagy kedvencem, a `nano`, de az `mcedit` is remekül használható. A héjprogramban találunk egy csomó beállítást a némettől a kínain keresztül a törökig, na, ide hozzunk létre egy magyar részt, vagy az egész `case` részt törölve írjuk be a következőket:

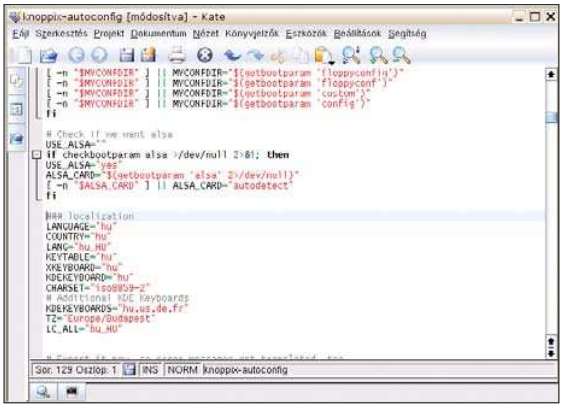
```
LANGUAGE="hu"
COUNTRY="hu"
LANG="hu_HU"
KEYTABLE="hu"
XKEYBOARD="hu"
KDEKEYBOARD="hu"
CHARSET="iso8859-2"
# Additional KDE Keyboards
KDEKEYBOARDS="hu,us,de,fr"
TZ="Europe/Budapest"
LC_ALL="hu_HU"
```

A másik hasonló változtatás az X billentyűzetkiosztása. Ehhez az új rendszer */etc/X11/XF86Config-4* fájljában kell az előző cikkemben is leírt változást megejteni. Ezt a fájlt egyébként az */usr/sbin/mkxf86config* fájl hozza létre, az XKEYBOARD környezeti változó értéke alapján, a változásokat itt is eszközölhetjük. Ahogy az előző cikkemben írtam, ezt a fájlt kezelhetjük a `dpkg-reconfigure xserver-xfree86` paranccsal, de akkor sajnos a Knoppix ügyes felismerőprogramja által kinyomozott monitorbeállítások elvesznek. A másik lehetőségünk tehát, hogy a beállítási fájlban lecseréljük az amerikai kiosztást magyarra. Ezt többféleképpen is megtehetjük, részemről egy egyszerű `sed` parancsot használok (ehhez hasonlóval az említett `mkxf86config` fájlt is könnyedén testre lehet szabni):

```
mv /etc/X11/XF86Config-4 /etc/X11/XF86Config-4-
backup
cat /etc/X11/XF86Config-4-backup | sed
s/\"XkbLayout\"\ \"us\"/\"XkbLayout\"\ \"hu\"/ >
/etc/X11/XF86Config-4
```

A rengeteg védőkarakter (a „\" jelek a parancsban) azért kell, hogy az idézőjelek és a szóközök a héjprogramot ne zavarják meg. Mivel hamarosan saját csomagjainkat is fel akarjuk telepíteni, meg kell adnunk a megfelelő csomagforrásokat a */etc/apt/sources.list* fájlban:

```
deb ftp://ftp.hu.debian.org/debian sarge main
contrib non-free
deb ftp://ftp.hu.debian.org/debian-non-US
sarge/non-US main contrib non-free
```



A többi sort nyugodtan megjegyzésbe tehetjük (minden „#” karakterrel kezdődő sor megjegyzésnek számít). Ezek után, ha még nem a merevlemezen futó rendszer él, indítsuk újra a gépünket. Az új rendszerben először is be kell állítanunk a hálózatot (`netcardconfig`), és már telepítgethetünk is:

```
apt-get update
apt-get install aptitude
aptitude
```

A rendszeren alapesetben rendkívül sok csomag található. Ha például irodai munkaállomásnak szeretnénk használni a gépet, akkor mind a játékok, mind a tördelő és grafikai programok, de még a php, a squid, a mysql, valamint a WLan-támogató csomagok is feleslegesek. Van több olyan oldal is, ahol az emberek közzéteszik kedvenc „ezeket-töröld-de-gyorsan” listáikat. Ilyen például a ➲ http://www.weblog.nohair.net/archives/000437.html címen található leírás második fele is. Egyszer érdemes végigfutni az aptitude listáján, de szerencsére a már telepített csomagok nem zavarnak sok vizet, így csak akkor érdemes nekiállni törölgetni, ha helyet szeretnénk megspórolni, vagy ha nem szeretnénk, hogy a titkárnő Frozen bubble-függő legyen. Erre a programra azonban még visszatérek egy kicsit később.
Lényegesebb, hogy telepítsük a nekünk fontos csomagokat. Keressük meg az alábbiakat, és telepítsük őket (a hozzájuk szükségesekkel együtt): `icu-locales`, `kde-i18n-hu`, `libi18n-charset-perl`, `liblocale-codes-perl`, `localeconf`, `mozilla-locale-hu`. A lista természetesen a gép feladatától függően változhat. Ha valaki például elsősorban irodai feladatokra használja a rendszert, érdemes az eredetileg felrakott ***openoffice.org*** helyett (ami egy német és angol nyelvet támogató barkácsolt csomag) a legújabbat telepíteni, lehetőleg magyar nyelvi támogatással együtt.
Ez eleve több csomagot jelent (az `openoffice.org` kezdetű csomagok közül a `-bin`, `-debian-files`, `l10n-hu` végűek is), illetve a flash bővítmény (`flashplugin-nonfree`) és az `msttcorefonts` is jól jön. Ez utóbbi csomag letölti a hálózatról a Microsoft által közzétett betűtípusokat (Arial, Verdana stb.), melyek akkor jönnek jól, ha valamelyik üzletfelünk Word dokumentumot küld át és szeretnénk pontosan úgy látni a fájlt, ahogy azt ő készítette. Érdemes a `localepurge` csomagot is felrakni, ez a program csomagtelepítések után magától lefut és sok helyet takaríthatunk meg, ha töröltetjük vele a mindenféle nyelv támogatását.
Új gépről lévén szó, érdemes az időzónát és a pontos időt is beállítani, valamint az alaplap órájába is elmenteni:

```
tzconfig
date --set pontosidő
hwclock --systohc
```

Ha a testreszabáshoz héjprogramot készítünk, akkor bosszantó lehet, hogy a `tzconfig` mindig megkérdezi, hogy meg akarjuk-e változtatni a beállítást. Majd meg kell adnunk a beállítást. Ehhez a héjprogramokban használható „beágyazott dokumentum” megoldást is használhatjuk. A lényege, hogy megadhatunk egy végjelet (itt `EOF`), és a héjprogram értelmezésekor a `bash` úgy kezeli a parancs és a végjel közötti részt, mintha az egy külső fájl volna. Az alábbi rész hatásával azonos volna, ha készítenénk egy külön fájlt, majd a `cat` paranccsal a `tzconfig` bemenetére irányítanánk annak tartalmát:

```
cat <<EOF | tzconfig
y
8
Budapest
EOF
```

Ha konzolon is szoktunk dolgozni és szeretnénk megtalálni a magyar ékezeteket, akkor a `dpkg-reconfigure console-common` segítségével választhatjuk ki, hogy a rendszer milyen billentyűzetkiosztást használjon alapértelmezettként. Ha ezek után újra elindítjuk az `aptitude` programot, akkor egy `g` után kiderül, hogy rengeteg „upgradable” azaz frissíthető állapotú csomag van a gépen. Ezeket az „Upgradable packages” soron állva egy `U` megnyomásával egyetlen gombnyomással megjelölhetjük frissítendőnek. Ha ezt megtesszük, akkor a rendszer elég sok anyagot le kíván majd tölteni.
Ha sok gépen kell ugyanígy telepítsünk, egyszerűbb, ha egy helyi tükröt, vagy egy `apt-proxy`-t használunk. Van egy egyszerűbb (és valljuk be, nem annyira szép) megoldás is. Az első gépen elvégezzük a letöltéseket, majd amikor a többi gépet telepítjük, egyszerűen átmásoljuk a már kész gép */var/cache/apt/archives* könyvtárának tartalmát a célgép egy adott könyvtárába és ott kiadjuk a `dpkg -i ./*deb` parancsot. Ezzel telepítjük az összes, a másik gépre letöltött csomagot. Remélhetőleg nem töltöttünk le olyan csomagot, amit később mégiscsak töröltünk.
Természetesen a törlések között leírthatunk a rendszerről sok felesleges dolgot is: játékokat, régi X-kiszolgáló programokat, fejlesztőeszközöket, stb. Ki kell viszont emelnünk a `frozen-bubble-data` csomagot, a drágaság ugyanis nem hagyja magát csak úgy törölni. Panaszkodik, hogy nem találja az eltávolító héjprogramját (helytakarékosság miatt ez nincs a rendszeren), de ha frissítjük ezt a csomagot, és utána akarjuk törölni, akkor is mindenáron törölni akarja a */usr/games/frozen-bubble.wav* fájlt, ami bizony nincs ott. A megoldás pofonegyszerű: másoljunk neki oda egyet, had örüljön: `cp /usr/share/apps/kget/sounds/added.wav /usr/games/frozen-bubble.wav`.
Még egy érdekes mozzanatot hangsúlyoz a legtöbb Knoppix-testreszabó oldal: Ez pedig az elárvult csomagok törlése. Egy csomag akkor árvul el, ha az összes olyan cso-

magot töröljük a rendszerről, amelyiknek szüksége van erre a csomagra. Lényegében tehát „már nem használt” csomagokról van szó. Ennek ellenére óva intek mindenkit az ellenőrzés nélküli csomagtörléstől, mindig legyünk körültekintőek. Ellenőrzési célzattal kitehetjük a csomagneveket egy fájlba, majd a fájlt végignézve nyugodtabb lehet a lelkünk a csomagok törlése kapcsán. Vagy egyszerűen csak hagyjuk a rendszeren ezeket a csomagokat, ha nem zavar minket az a tíz-tizenöt megabájtnyi foglalt terület. A törlésre az alábbi parancsot használom:

```
deborphan > ~/torolni.txt
mcedit ~/torolni.txt
apt-get remove `cat ~/torolni.txt`
```

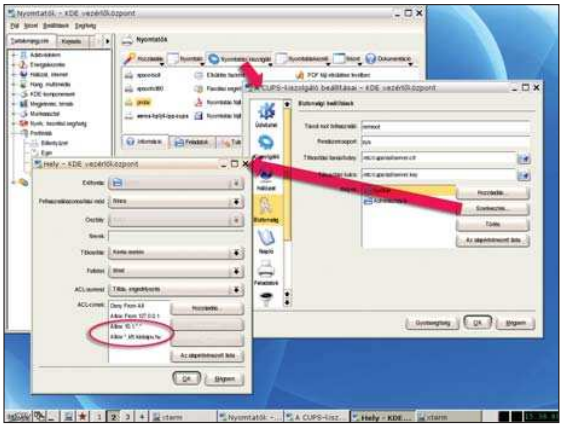
Némi időt rászánva pillanatok alatt összedobhatunk egy héjprogramot, amely elvégzi a fenti feladatokat, és igény szerinti telepítést kapunk eredményül.

És megint CUPS

Külön szólnom kell a nyomtatásról. Alapesetben a cups jár a géphez, és ahogy az az előző cikkemben írtam, szerintem ez jó választás. Bár már többször érintettük a nyomtatás témakörét, mégis érdemes újra elővenni, ugyanis ez az egyik leggyakrabban előkerülő mumus a linuxos irodai rendszereknél. Ahogy írtam, több nyomtatórendszer is létezik Linuxhoz és a leggyorsabban úgy tudunk káoszt előidézni, hogy felrakjuk mondjuk egyszerre az `lpd`-t és a `cups`-ot is. Tehát figyeljünk arra, hogy a gépeken csak a `cups` legyen jelen.
A CUPS egy okos és sokmindent önműködően megoldó rendszer, de hogy használni tudjuk, be kell állítanunk a kiszolgálók és az ügyfelek oldalán is. Nézzük először a kiszolgálók beállítását. Először is meg kell oldani, hogy a kiszolgáló mindig magától elinduljon (kézzel a `/etc/init.d/cupsys start` paranccsal indíthatjuk). Ehhez használjuk az előző cikkben említett `ksysv` programot, vagy mi magunk is létrehozhatjuk a szükséges hivatkozásokat, minden futási szinthez külön-külön, például:

```
cd /etc/rc5.d
ln -s ../init.d/cupsys S60cupsys
```

A CUPS beállításához nem a szokásos böngészőből megnyitott ➲ http://localhost:631 oldalról elérhető webes beállítót javaslom, hanem a KDE-hez tartozó vezérlőközpont nyomtató-beállító alkalmazását. Ebben a programban egyrészt könnyen tudjuk telepíteni a nyomtatót, másrészt a kiszolgáló beállításait is könnyen meg tudjuk adni. A program jobb alsó sarkában ellenőrizhetjük, hogy melyik nyomtatási rendszert akarja éppen használni. Ezt állítsuk CUPS-ra! Anélkül, hogy mélyebben belemennénk, egy egyszerű irodai környezetben egyszerűen beállíthatjuk a nyomtatókat. Nem kell mást tennünk, csak a nyomtatót felvenni helyi nyomtatónak azon a gépen, melyhez csatlakozik. Ugyanezen a gépen engedélyezzük azokat a bejövő gépeket, ahonnan nyomtatni szeretnénk: ehhez a Nyomtatási kiszolgáló menügombból a kiszolgáló beállításait választva, a megjelenő ablakban pedig a Biztonság alatt a ***Gyökér*** elemet szerkesztve adhatunk hozzá engedélyezett, illetve tiltott címeket (lásd a fenti képet). Lényegében itt adjuk meg, hogy melyik géprő fogad el a rendszer nyomtatást. Ha itt nem engedélyezzük a nyomtatást, akkor könnyen találkozhatunk a kö-



vetkező hibával: az ügyfél látja a nyomtatót, tud is rá nyomtatást küldeni, de a nyomtatás a helyi sorban várakozik, nem jut el a kiszolgálóig. Hogy milyen nyomtatókat lát egy gép, azt az `lpstat -a` paranccsal nézhetjük meg.
Az ügyfélgépek beállítására két lehetőségünk van. Az első, hogy az irodában van egy-két megbízható nyomtatókiszolgáló (például ha csak két nyomtató van, melyek egy központi géphez, mondjuk a ***prnsrv*** nevűhöz vannak kötve), akkor ezt a kiszolgálót használhatja az ügyfélgép, mint nyomtatáskiszolgáló. Ennek beállításához az ügyfélgépen a ***/etc/cups/client.conf*** fájlba írjuk be az alábbi sort (gépnév helyett megadhatunk IP-címet is):

```
ServerName prnsrv.cegnevem.hu
```

Az „önműködőbb” megoldás, hogy minden fontosabb gépre cups-kiszolgálót telepítünk, ami folyamatosan gyűjti a tallózólistáját, így mindig az éppen elérhető nyomtatókat felkínálva. Ez a megoldás hibatűrőbb, mint az egy központi kiszolgáló, de könnyebben okoz zavarokat. Előfordulhat például, hogy a nyomtatót átdugja a titkárnő a másik gépbe, hiszen „annak a gépnek is van ugyanolyan nyílása”, és még a nyomtató sem tűnik el (a CUPS számára a nyomtató tovább létezik, csak éppen nem érhető el). Így az előbb leírt hibajelenséghez hasonló helyzetbe jutunk. Még egy ügyes tulajdonsága van a CUPS rendszernek: ha több gépre telepítünk ugyanolyan nevű nyomtatót, akkor az ügyfél azt a nyomtatót használja, amelyik éppen ráér.
Természetesen nem kell tallózólistákból dolgozni, telepíthetünk „rejtett” nyomtatókat is, olyanokat, amelyek a böngészőben (tallózólistában) nem szerepelnek, ezek telepítéséhez azonban ismerni kell a nyomtató teljes elérési útját és típusát.

Ennyi az egész?

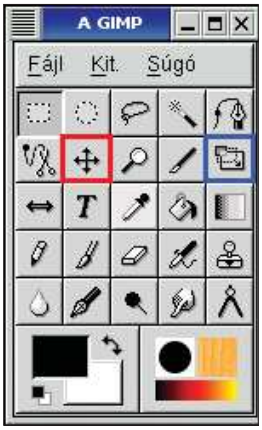
Igen, készen vagyunk, egy teljes irodai hálózatot feltelepítettünk, szerkesztőprogramokkal, nyomtatással, tokkal-vonóval. Ha tényleg hálózatban akarjuk használni a gépet, akkor még meg kell említenem, hogy érdemes egy névkiszolgálót is telepíteni, valamint egy DHCP-kiszolgálót. Ez utóbbival ugyanis megtakaríthatjuk az ügyfélgépek egyenkénti beállítását.
Kellemes időtöltést kívánok a Linux irodai használatához!

Szy György

A GIMP és a színek (3. rész)

Harmadik részéhez érkezett a Gimp nevű programot bemutató sorozatunk. Az írás segítségével tovább ismerkedhetünk a rendszer rejtett értékeivel, és megtanulhatunk egy-két igen hasznos trükköt is.

Ebben a hónapban tovább folytathatjuk a GIMP csodálatos világával való ismerkedést, amely még nekem is szolgált meglepetéssel. E sorozat írása közben ugyanis még ennyi idő után is fedezek fel újdonságokat a programban. A másik, szintén ide kívánkozó jó hír, hogy nemrégiben jutottam hozzá a GIMP 2.0-ás változatához és örömmel láttam, hogy a scheme parancsnyelv mellett már Python nyelven is írhatjuk a kiegészítő vagy munkánkat megkönnyítő segédprogramokat. Ezekkel bővebben is megismerkedünk hamarosan, egyelőre legyen annyi elegendő, hogy mostantól nem kell új programnyelvet megtanulnunk, és mégis kihasználhatjuk a GIMP önműködő képfeldolgozásra való képességét.



1. kép A GIMP eszköztára

Nos, e könnyed bevezető után bizonyára mindenki ráhangolódott a tanulásra, ismerkedjünk hát meg néhány hasznos menüponttal. Bizonyára mindenkivel előfordult már, hogy egy képről el kellett távolítania a felesleges, üres részeket. Például valaki egy kicsit nagyobbra jelölte a lapolvasó aktív területét, és emiatt olyan fehér területek kerültek a beolvasott képre, amelyeknek nincsen sem információtartalmuk, sem egyéb jelentőségük. Ilyen esetekben lehet segítségünkre az ***Automatikus levágás (Autocrop)*** és a ***Zealous Crop*** menüpont, amelyek megpróbálják ezeket a területeket megkeresni, és a képet méretre vágják. Tapasztalatom szerint a második megoldás, vagyis a ***Zealous Crop*** jobb végeredményt ad, ám lassabban működik, ami egy gyorsaságot igénylő munka esetén igen hátrányos.

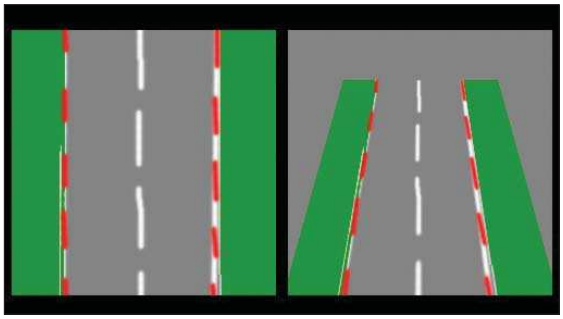
Ezeket a kivágásokat a jobb egérgombbal a képre kattintva, a menüben érhetjük el, a ***Kép->Átalakítás (Image->Transforms)*** pontokon keresztül.

Itt találhatunk még egy nagyon hasznos menüpontot, ami a ***Guillotine*** néven található meg. A GIMP lehetőséget ad arra, hogy függőleges és vízszintes segédvonalakat helyezzünk el a képen. Ennek használatához nyomjuk meg valamelyik vonalzó fölött a bal egérgombot, és húzzuk rá a képre a segédvonalat. A felső vonalzóról lehet vízszintes segédvonalat a képre helyezni, míg a bal oldaliról függőlegeset.

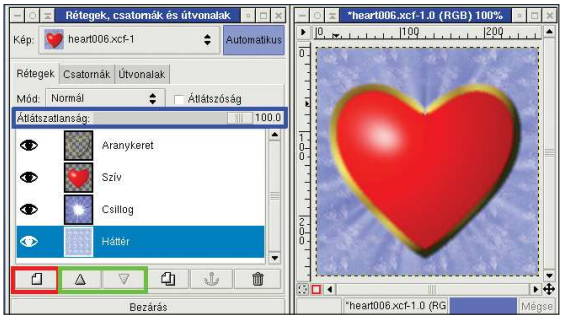
Amiért mindez érdekes számunkra, az az imént említett menüpont. Ha ezt kiválasztjuk, akkor a GIMP a megadott segédvonalak mentén feldarabolja a képet. Ennek gyakorlati hasznát remélhetőleg nem kell részletezni ezért csak megemlítek párat: használhatjuk arra, hogy a honlapunkon a nagyobb képeket feldaraboljuk a táblázatos megjelenítéshez, vagy arra, hogy amikor játékunkat készítjük, akkor a különféle figurákat egy képről több kisebbre helyezzük el.

Tulajdonképpen nem is kell megvárnunk, amíg eltűnik ez a menü, máris folytathatjuk a különféle átalakítások megismerését. Az ***Eltolás (Offset)*** segítségével a képet eltolhatjuk megadott mértékben. Ha a választó párbeszédablakban kiválasztjuk a ***Körbekerít (Wrap around)*** kapcsolót, akkor a kép kilógó részei az ellenkező oldalon visszatérnek, ha viszont ezt nem kapcsoljuk be, akkor eltűnnek.

Természetesen a képet el is forgathatjuk, mégpedig előre megadatott kilencven fokos lépésekben a ***Forgatás (Rotate)*** menüben található forgatási értékek kiválasztásával, vagy az eszköztár segítségével. Az eszköztárban az 1-es képen kékkel keretezett gombra kattintva az egérmutató megváltozik és egy forgatásra utaló jelként jelenik meg. Ha ilyenkor a képre kattintunk, megjelenik egy négyzetrács, közepén egy fekete pöttyel. A fekete pont adja meg a forgatás középpontját és az egérgomb lenyomása közben mozgatva az egeret megadhatjuk az elforgatás szögét. Ugyanezzel az eszközzel lehetőségünk van átméretezésre, torzításra és perspektíva létrehozására is. Egy ilyen perspektivikus átalakítás eredménye látható a 2-es képen. A különféle átalakításokat úgy érhetjük el, hogy az eszköz gombjára kettőt kattintva a felbukkanó (3-as képen látható) párbeszédablakban kiválasztjuk a megfelelő átalakítást. Itt állíthatjuk be azt is, hogy az átalakítás során megjelenjen-e a segítséget nyújtó négyzetháló, és hogy az eredményt szeretnénk-e simítani. Ez utóbbi művelet főként nagyításkor (***Scaling***) lehet érdekes. Ilyenkor nem egyszerűen többször jelenik meg a képpont, de a szomszédos képpontok között átmenetek is keletkeznek.

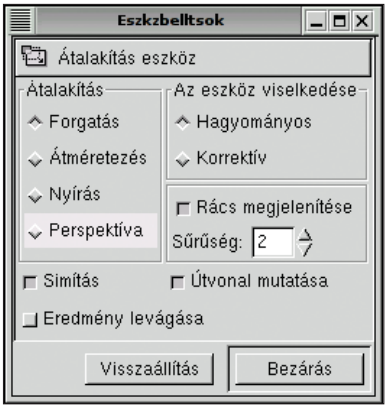


2. kép Perspektívikus átalakítás



4. kép A réteg-kezelő

Más lehetőségünk is van az eltolásra és az átméretezésre. Az 1-es képen piros kerettel jelölt eszközzel tudjuk a képet vagy annak valamely rétegét eltolni, míg átméretezni a menüben az ***Kép->Kép átméretezése (Image->Scale) Image*** menüpontokat kiválasztva lehetséges. Miután az említett eszközzel eltoltuk a képet, célszerű a ***Rétegek->Réteg kép méretűvé (Layer->Layer to imagesize)*** menüpontok kiválasztásával a réteg méretét a képmérethez igazítani. Az előbbi menüben találjuk meg a rajzvászon átméretezésére szolgáló menüpontot is, mégpedig a ***Kép->Rajzvászon mérete (Image->Canvas) size*** pontok kiválasztásával. Ez a szolgáltatás akkor lehet hasznos, ha valamilyen kép mellé szeretnénk elhelyezni egy másikat, vagy hozzárajzolnánk más formákat. Így készült például a 2-es képen látható átalakított útdarabka is. Maradjunk még egy kis ideig a ***Kép (Image)*** menü tárgyalásánál. Itt találhatunk ugyanis két nagyon fontos eszközt. Az egyik a kép színmélységének változtatására szolgál, ezt a Mode almenüben találhatjuk meg. Itt alakíthatjuk át a képeinket szürkeárnyalatosra, vagy korlátozott színekből összeálló képpé. Fontossága abban rejlik, hogy a GIMP a legtöbb szűrést és a bonyolultabb műveleteket csak RGB színekkel rendelkező képeken tudja elvégezni. Tehát ha nem ilyen a feldolgozandó kép – például a GIF formátumú képek maximálisan 256 színt használhatnak –, akkor először át kell alakítanunk. Ez az átalakítás akkor is jól jöhet, amikor olyan megjelenítő-eszközünk van, ami nem képes sok szín leképezésére. Ha ilyen eszközökön (régebbi mobilok, játékkonzolok) szeretnénk megjeleníteni a képeket, akkor felhasználás előtt a át kell azokat alakítani.



3. kép Átalakítás beállításai

A másik fontos menüpont tulajdonképpen kéttő darab menüpont. Az egyik a ***Összevonás (Compose)***, míg a másik a ***Felbontás (Decompose)***. Ez utóbbival tudjuk az elkészült képeket színrétegekre bontani, vagyis a nyomdai előkészítéshez alkalmas formátumra alakítani. Itt választhatunk RGB, HSV, CMYK, CMY színtereket, vagy ha a képnek van átlátszóság csatornája (Alpha channel) akkor szintén ezzel a művelettel juthatunk hozzá. A ***Összevonás (Compose)*** menüpont alkalmas az előbbi művelet megfordítására. A megadott képeket színcsatornaként értelmezve összeállítja az eredeti képet. A képhez az ***Alpha*** menü segítségével adhatunk átlátszóságot szabályozó színcsatornát. Sajnos, tapasztalatom szerint, ezt csak úgy tudjuk szerkeszteni, hogy a képet szétválasztjuk csatornáira, majd az alfa csatorna szerkesztése után újra összeállítjuk.

Egy nagyon hasznos lehetőséget szeretnék még bemutatni a programmal kapcsolatban, ami nélkül ma már egy modern pixelgrafikai program szinte elképzelhetetlen. A GIMP lehetőséget biztosít a felhasználónak arra, hogy a teljes grafikát egymás felett elhelyezkedő fóliákból állítsa össze. Ezeket a fóliákat a szaknyelv rétegeknek (Layer) nevezi. A rétegek használatának szintén sok felhasználási lehetősége adódhat, gondoljunk csak arra, hogy egy képből kivágott részletet szeretnénk egy másikon elhelyezni. Ilyenkor a kivágott képrészletet külön rétegen helyezzük el, majd a helyzetének pontos beállítása után hozzákapcsolhatjuk az alatta lévő réteghez. A GIMP-ben a rétegek kezelésére szolgáló párbeszédablakot a CTRL-L billentyűkkel érhetjük el. A párbeszédablak legnagyobb részét a rétegek megjelenítése foglalja el. Minden egyes tételnek külön nevet adhatunk, ha kettőt kattintunk rá. Az első (legalsó) réteg a háttér, ennek az a különleges tulajdonsága, hogy nem adhatunk hozzá a megjelenést szabályozó maszkot és nem helyezhetjük más rétegek fölé. Háttér rétege minden képnek van, azonban a háttérnek is megadható átlátszóság, aminek akkor láthatjuk hasznát, ha a honlapunkon átlátszó PNG vagy GIF képeket használunk. A 4-es képen látható egy kép és a rétegeket megjelenítő párbeszédablak, amelyhez több réteget használtam. A rétegeket úgy tehetjük aktívvá, hogy a nevükre kattintunk az egérrel. Így a szűrők és a rajzoló műveletek is a kiválasztott rétegen érvényesülnek. Minden réteg neve mellett megjelenik egy előnézeti kép és egy szem, már ha a réteg látható. A képen piros kerettel jelzett gomb segítségével adhatunk új rétegeket a képhez, ezek átlátszóságát pedig a kék kerettel jelölt csúszkával állíthatjuk. A zölddel keretezett gombok alkalmasak a kijelölt réteg sorrendjének megváltoztatására. Ezek mellett találhatjuk a következőket: réteg kétszerezése, réteg csatolása az alatta lévőhöz és a réteg törlésére szolgáló szemetes. A rétegek csatolására szolgáló gomb olyan esetekben

alkalmazható, amikor valamilyen más elemet illesztettünk a képre vagy arról kiválasztottunk egy részletet.
Még mielőtt valakinek a használat során kellemetlenséget okozna a nem megfelelő fájlformátum kiválasztása, el kell mondanom, hogy a GIMP a rétegeket csak a saját formátumában mentve képes megőrizni. Ez a formátum az XCF, és amikor a képet mentjük, elegendő a kiterjesztésként megadni, nem kell feltétlenül kiválasztanunk a legördülő listából.
Most már tudjuk, hogy az egyes rétegeknek megadhatunk átlátszóságot, itt az ideje hát, hogy megismerkedjünk a további lehetőségekkel is. Minden rétegnek megadható, hogy milyen kapcsolatban legyen az alatta lévőkkel, azaz milyen műveletet végezzen a program a rétegek képpontjai között. Itt csak néhány lehetőséget mutatnék be, hiszen minden műveletet minden kombinációban megmutatni gyakorlatilag lehetetlen ilyen szűkre szabott keretek között. Az adott műveletet az átlátszóságot meghatározó csúszka felett található legördülő listában választhatjuk ki. Ezek közül a ***Kivonás (Subtract)*** azt jelenti, hogy az adott réteg képpontértékeit a program kivonja az alatta lévő képpontokból, az ***Összegzés (Addition)*** pedig a képpontok között végez összeadást. Érdekes még a ***Érték (Value)***, amelynek alkalmazásakor csak az adott réteg világosság-értékeit veszi figyelembe a program a végső kép meghatározásakor. Mindezen műveletek kombinálása legyen az olvasó feladata, ugyanis valószínűleg szórakoztató és izgalmas kihívást talál benne minden a kreativitást kedvelő felhasználó.
És akkor most térjünk át a rétegek menüpontjaira. Ha egy rétegre a jobb egérgombbal kattintunk, akkor egy menüt láthatunk, amiben fontos műveleteket végezhetünk az adott réteggel. Az első öt menüpontot nem kell külön magyaráznom, mert hatásukban megegyeznek a párbeszédablak alsó részén található gombokkal. A következő a ***Réteg határvonal mérete (Layer boundary size)***, amellyel meghatározhatjuk, hogy az adott rétegnek mekkora részét vegye figyelembe a program a műveletek végzésekor, míg a ***Réteg kép méretűre (Layer to Imagesize)*** arra való, hogy a kijelölt réteg méretét a képmérethez igazítsa. A következő a ***Réteg átméretezése (Scale Layer)*** ami megegyezik a kép átméretezésével, csak ebben az esetben kizárólag a kijelölt rétegre vonatkozik. Természetesen a rétegeket egyesíthetjük is, mégpedig a következő menüpontok segítségével. A ***Látható rétegek összefésülése (Merge visible layers)*** az éppen látható rétegeket egyesíti. A réteg láthatósága az előtte lévő „szem” ikon alapján állapítható meg. Ez az egyesítési művelet akkor lehet hasznos, amikor úgy gondoljuk, hogy a kép egy részével már elégedettek vagyunk, például már pontosan a helyén van a beillesztett képdarabka. Ekkor egyesítjük a hozzá tartozó réteggel és a továbbiakban nem kell fognunk a fejünket, ha egy szűrőt mondjuk csak az összetartozó rétegek egyikére alkalmaztunk.
A ***Összefésülés lefelé (Merge Down)*** menüpont hasonlóan működik az előbbihez, de itt csak közvetlenül a kijelölt réteg és az alatta lévő egyesítése történik. A ***Egy réteggé lapítás (Flatten Image)*** menüpont kiválasztásával érhetjük el a kép minden rétegének egyesítését. Ha valami olyan formában mentjük a képet, ami nem támogatja a rétegeket, akkor a GIMP maga is felkínálja, hogy egyesíti a rétegeket, és ilyenkor célszerű ezt engedélyezni, hiszen ha ezt nem tesszük meg, akkor csak az éppen kiválasztott réteget fogja menteni a program. Az ezután következő három menüpont a rétegek maszkolására használatos. A rétegmaszkot úgy kell elképzelni, hogy az ezen található világosságértékek határozzák meg, hogy a rétegnek ez a része mennyire látható. Ha ez az érték 0 (fekete), akkor a hozzá tartozó képpont átlátszó, ha 255-ös (fehér), akkor a képpont teljesen látható. Természetesen nem csak ez a két érték használható. Tehát a maszkot először is létre kell hozni. Ezt a ***Rétegmaszk hozzáadása (Add Layer Mask)*** menüpont választásával tehetjük meg, majd meg kell adnunk, hogy kezdetben a maszk milyen legyen. Lehet teljesen átlátszó, teljesen átlátszatlan és használhatjuk a réteg meglévő alfa csatornáját is. A maszkról is készül előnézeti kép, amit a létrehozás után láthatunk is a rétegkezelő párbeszédablakban. Amikor a maszkkal elkészültünk, a réteghez véglegesen hozzárendelhetjük a ***Rétegmaszk érvényesítése (Apply layer mask)*** menüpont kiválasztásával. Azt azonban tudnunk kell, hogy a maszkot hogyan szerkeszthetjük. A réteg kiválasztása után kattintsunk a hozzá tartozó maszk előnézeti képére. Ezt követően a képszerkesztő műveletek mindaddig a maszkra vonatkoznak, amíg a képet nem választjuk ki újra. Mielőtt még ezt a végleges hozzárendelést elvégeznénk, lehetőségünk van a rétegmaszk törlésére is a ***Rétegmaszk törlése (Delete Layer Mask)*** menüpont választásával. Az utolsó menüpont, amivel a maszkot kezelhetjük, egy igen hasznos művelet. Ez a ***Maszk kijelöléssé alakítása (Mask to Selection)*** amivel kiválaszthatjuk a maszknak megfelelő formát.
Most pedig néhány ötletet szeretnék adni, hogy megjöjjön a kedves olvasó kísérletező kedve. Tegyük fel, hogy egy almát kell elhelyeznünk egy már meglévő tálba. Az almát ekkor egy másik képről kijelöljük, majd a tálat tartalmazó képre illesztjük. Elhelyezzük a tálban, de sajnos azt kell látnunk, hogy a tál széle nem takarja el az almát. A kép olyan, mintha az alma a tál előtt lenne az asztalon. Ekkor segíthet például a maszkolás. Az almát külön rétegre helyezzük, a takart részt pedig maszkkal takarjuk el. Ezután már csak az árnyékokat kell egy kicsit átrajzolni, a tálban az alma árnyékát megfesteni és nagy vonalakban készen is van a tál feltöltése. Természetesen ezt a feladatot megoldhattuk volna úgy is, hogy a felesleges részeket egyszerűen letöröljük, ekkor azonban az almának az örökkévalóságig ugyanabban a magasságban kellene maradnia, nem helyezhetnénk feljebb, mert akkor nem lenne teljes a kép. A fenti módszerrel például kedvenc portrénkat is keretbe helyezhetjük, ahogyan az 5-ös képen látható, most azonban búcsúzom kedves olvasóimtól, remélhetőleg mindenkinek hasznos időtöltést jelent a program használata és ha segítségre van szükség, az eddigi ismert e-mail címemen várom a kérdéseket.

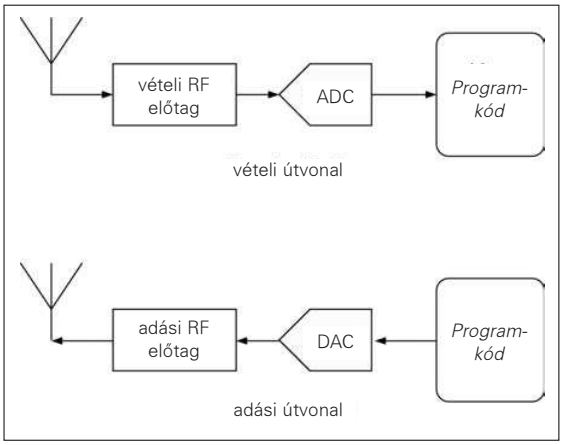
5. kép Portré

Fábián Zoltán

GNU Radio: A rádiófrekvenciás világ felfedezésének eszközei

A program alapú rádiók célja az, hogy a programkódot és az antennát a lehető legközelebbi kapcsolatba hozzák egymással. A GNU Radio mindenkinek biztosítja azokat az eszközöket, amelyek segítségével csatlakozhat a gyors processzoroknak köszönhető távközlési forradalomhoz.

A programrádió lényegében az antenna és a programkód viszonyának lehető szorosabbá tételét jelenti. Gyakorlatilag minden, a rádiókészülékkel kapcsolatos problémát programozási problémává alakít. Ennek a rádiónak a legfontosabb ismertetőjele, hogy egy program határozza meg az elküldött hullámformákat, és ugyancsak program végzi a fogadott hullámok visszaalakítását. Ez a működésmód szöges ellentétben áll a megszokott rádiókéval, amelyekben a jelfeldolgozást vagy digitális lapkákkal kiegészített vagy tisztán analóg áramkörök végzik. A GNU Radio egy szabadon hozzáférhető programozási eszközkészlet az említett programrádiók készítéséhez.
A programrádió valóságos forradalmat jelent a rádiók tervezésében, az így létrejövő készülékek üzem közben is változhatnak, korábban elképzelhetetlen lehetőségeket kínálva használójuknak. A programrádiók természetesen képesek mindarra, amire hagyományos társaik, de minket most sokkal inkább a programok használatából fakadó rugalmasság érdekel. A ma még igencsak korlátozott képességű, mindig csak egy adott feladatra használható készülékek helyett a közeli jövőben többfunkciós távközlési eszközöket fogunk használni. Képzeljünk el egy olyan készüléket, amely mobiltelefonként lehetővé teszi GPRS alapú kapcsolatok létesítését, de a 802.11 Wi-Fi és a 802.16 WiMax szabvánnyal valamint a műholdas összeköttetésekkel is megbirkózik – hogy a jövőbeli új megoldásokról most ne is beszéljünk.
A legérdekesebb mindebben, hogy központok nélküli távközlési rendszereket lehet majd építeni. Ha megnézzük a meglévő rendszereket, azt látjuk, hogy túlnyomó részük infrastruktúra alapú. A jelenlegi rádió- és TV-adások egyirányú csatornákon jutnak el hozzánk, szigorúan szabályozzák őket, és az általuk szolgáltatott tartalmat számos szervezet ellenőrzi. A mobiltelefonok ugyan kényelmessé teszik életünket, ám az általuk támogatott szolgáltatások köre a szolgáltató a vállalat érdekeit szolgálja, és nem a mienket.
A központosított rendszerek megnehezítik az újdonságok bevezetését. Ha ezt nehezen tudjuk elképzelni, akkor gondoljunk csak az internetre, és máris helyben vagyunk.
A mobiltelefonnak nem kellene tehetetlenül csüngenie a szolgáltató sokszor korlátozottan elérhető, korlátozott szolgáltatásokat nyújtó hálózatán, hanem igazán okos eszközzé válhatna. A hálózatot maguk a felhasználók tulajdonában lévő készülékek alkotnák. A készülékek egymás között hálót hoznának létre, egyeztetnék a háttérszolgáltatásokat; és minden újszerű megoldásra, szolgáltatásra és alkalmazásra nyitottak lennének.



1. ábra Általános programrádió blokkdiagramja

Blokkdiagram

Az *1. ábrán* egy általános programrádió blokkdiagramja látható. Ha meg akarjuk érteni a programrész működését, először a szerkezet jellemzőit kell gyorsan áttekintenünk. Ha megvizsgáljuk az 1. ábra vételi útvonalát, akkor egy antennát, egy rejtélyes RF előtagot, egy analóg-digitális átalakítót (ADC) és egy programkódot láthatunk. Az analóg-digitális átalakító köti össze a folytonos analóg jelek és a programból is kezelhető, diszkrét digitális minták világát.
Az ADC-k két fő jellemzője a mintavétel gyakorisága és a dinamikatartomány. A mintavételi gyakoriság azt mutatja meg, hogy az ADC másodpercenként hányszor méri meg

1. kódrészlet Hello Világ! (tárcsahang megszólaltatása)

```
#!/usr/bin/env python

from GnuRadio import *

def build_graph ():
    sampling_freq = 32000
    ampl = 8192

    fg = gr_FlowGraph ()

    src0 = GrSigSourceS (
        sampling_freq, GR_SIN_WAVE, 350, ampl)

    src1 = GrSigSourceS (
        sampling_freq, GR_SIN_WAVE, 440, ampl)

    sink = GrAudioSinkS ()

    fg.connect (src0, sink)
    fg.connect (src1, sink)

    return fg

if __name__ == '__main__':
    fg = build_graph ()
    fg.start ()         # szál(ak) létrehozása
➥fork hívásokkal és visszatérés
    raw_input ('A kilépéshez nyomd le az Enter
➥gombot: ')
    fg.stop ()
```

az analóg jelet. A dinamikatartomány a legkisebb és a legnagyobb megkülönböztethető jel közötti tartományt jelenti, amely az ADC digitális kimenetének és az átalakító kialakításának függvénye. Egy nyolcbites átalakító például legfeljebb 256 jelszintet képes ábrázolni, míg egy 16 bites átalakító már 65536-ot. Általában elmondhatjuk, hogy az átalakító fizikai kialakításától és árától függ, hogy mekkora kompromisszumot kell kötnünk a nagyobb mintavételi gyakoriság vagy a szélesebb dinamikatartomány tekintetében.

Mielőtt elmerülnénk a programozási kérdésekben, nem árt némi elméleti alapozást. 1927-ben a svéd fizikus és villamosmérnök ***Harry Nyquist*** fogalmazta meg azt a szabályt, amely szerint analóg-digitális átalakításnál az ADC mintavételi frekvenciájának legalább kétszer akkorának kell lennie, mint a mintavételezett jel sávszélességének. Ellenkező esetben ugyanis a jel nem állítható hibátlanul helyre, azaz jeltorzulás történik. Hasonló jelenségnek lehetünk tanúi, amikor a régi westernfilmekben úgy látjuk, mintha a lovaskocsik kerekei hátrafelé forognának. Ilyenkor a kamera mintavételi gyakorisága nem volt elég nagy ahhoz, hogy egyértelműen rögzíteni tudja a küllők állását.

Tegyük fel, hogy a jelek felülről korlátosak, vagyis a minket érdeklő sávszélesség 0 és `fmax` közötti. Ilyenkor a Nyquist-szabály szerint a mintavételi frekvencia értéke legalább 2 * `fmax` kell legyen. Igen ám, de ha az ADC-nk 20 MHz-en fut, akkor hogyan fogjuk hallgatni a 92,1 MHz-en sugárzott rádióadást? Pontosan ezért van szükség az RF előtagra.

A vevőoldali RF előtag a bemenetére adott frekvenciákat a kimeneten alacsonyabb tartományba tolva adja ki. Az RF például használható arra is, hogy a 90-100 MHz tartományba eső jeleket a 0-10 MHz tartományba tegyük át.

A legtöbb esetben elég az is, ha az RF előtagot egyfajta fekete dobozként kezeljük, amelynek vezérlő bemenetén csupán a megváltoztatandó frekvenciatartomány közepét kell megadnunk. Tényleges példaként említhetném azt a kábelmodemes vevőegységet, amelyet mi is sikerrel alkalmaztunk, és amely az 50-800 MHz tartomány egy 6 MHz-es szeletét teszi át az 5,75 MHz-es közepű tartományba. A kimeneti tartomány középső frekvenciáját középfrekvenciának (intermediate frequency, röviden IF) nevezzük.

Ha a végletekig egyszerűségre törekszünk, az RF előtagot akár el is hagyhatjuk. Van olyan, akinek a GNU Radio próbálgatása közben sikerült AM és rövidhullámú adásokat hallgatnia úgy, hogy egy száz méteres kábelt közvetlenül egy 20 millió minta/másodperc mintavételű ADC-hez csatlakoztatott.

Lássuk a programokat!

A GNU Radio tartalmaz egyrészt elemi jelfeldolgozó függvényeket, másrészt ezek egységes alkalmazássá gyúrásához kiegészítőket. A programozó a rádiót úgy állítja össze, mintha egy gráfot rajzolna meg. A gráf pontjai az elemi jelfeldolgozó függvények, az élek pedig a köztük folyó adatáramlást jelképezik. A jelfeldolgozó függvények C++ nyelven készültek. A függvények lényegében végtelen, a bemenetük felől a kimenetük felé haladó folyamokat alakítanak át. A függvények fontos jellemzője bemeneteik és kimeneteik száma, illetve az ezeken keresztülhaladó adatok típusa. Leginkább kisméretű egész (short), lebegőpontos (float) és komplex mennyiségeket használunk.

Bizonyos jelfeldolgozók csak kimenettel vagy csak bemenettel rendelkeznek, ezek adatforrásként és -nyelőként viselkednek. A források közt találunk olyat, amely fájlból vagy ADC-ről olvas, a nyelők pedig képesek fájlba, digitális-analóg átalakítóra (DAC) vagy grafikus kijelzőre írni.

A GNU Radio körülbelül száz elemi jelfeldolgozó függvényt tartalmaz, de újak készítése sem ördöngösség.

A gráfok alapvetően C++ nyelven állíthatók össze és futtathatók, de működő egység összeállítása Python alatt a legegyszerűbb. Az *1. kódrészlet* a „Hello Világ” GNU Radio-féle változata. Két szinuszhullámot állít elő, majd a hangkártyára küldi őket, az egyiket a bal, a másikat a jobb csatornára.

Első lépésünk egy áramlási gráf (flow graph) létrehozása, ebbe kerülnek majd az elemi jelfeldolgozó függvények és a közöttük fennálló kapcsolatok. A két szinuszhullám a `GrSigSourceS` hívások révén jön létre. Az S utótag jelzi, hogy a forrás `short` kimeneteket ad. Az egyik szinuszhullám 350, a másik 440 Hz-es frekvenciájú. Ha együtt szólaltatjuk meg őket, mintha az Amerikában megszokott tárcsahangot hallanánk.

A `GrAudioSinkS` olyan nyelő (sink), amely bemenetét (általában egy vagy két `short` típusokat tartalmazó folyam) a hangkártyára írja ki. A három elemi jelfeldolgozó függvényt az áramlási gráf `connect` függvényével kapcsoljuk össze.

Amikor készen áll a gráf, elindítjuk. A `start` hívás hatására egy vagy több szál jön létre (`fork` hívásokkal), ezekben futnak a gráf által megadott számítások; a vezérlés pedig azonnal visszakerül a hívóhoz. Ilyenkor egyszerűen csak várjunk, amíg a felhasználó le nem nyomja valamelyik billentyűt.

Teljes értékű FM vevő

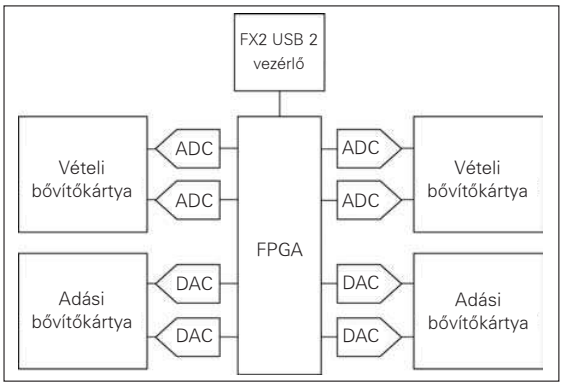
A 2. *kódrészlet* egy leegyszerűsített, de működő FM vevő összeállítását szemlélteti. Magába foglalja az RF vezérlését és a szükséges jelfeldolgozókat is. Ebben az esetben egy kábelmodemes vevőegységből épített RF előtag használatáról és egy 20 millió minta/másodperces analóg-digitális átalakító használatával számoltunk. Ahogy a Hello Világ!-os példánál, itt is, létrehozzuk a gráfot, összekötjük az elemi jelfeldolgozókat, majd kiadjuk az indulási parancsot. Forrásunk ebben az esetben a nagysebességű ADC, a `GrHighSpeedADC`. Ezt a `GrFreqXlatingFIRfilterSCF` követi, ami egy véges impulzusválaszú (FIR) szűrő. Ez választja ki a kívánt FM állomást, majd alakítja alapsávúra (0Hz, DC) a jelet. A 20 millió minta/másodperces átalakító és a kábelmodemes vevőegység használatával nagyjából a teljes tartomány egy 6 MHz-es szeletét vesszük. Ebbe a szeletbe tíz vagy akár több FM állomás is eshet, az általunk kívántat a `GrFreqXlatingFIRfilterSCF` segítségével választhatjuk ki. Most azt az állomást választjuk, amely az RF előtag IF értékének közepére esik (5,75MHz). A szűrő kimenete komplex minták folyama, 160 ezer minta/másodperc gyakorisággal. A komplex alapsávi jelet a `GrQuadratureDemodCF`-fel etetjük meg, ez végzi a tényleges FM demodulációt. A `GrQuadratureDemodCF` a szomszédos komplex minták szögeinek kivonásával végzi munkáját, lényegében differenciálva a frekvenciát. A `GrQuadratureDemodCF` kimenete tartalmazza a bal plusz jobb mono hangjelet, a 19 kHz-es elválasztójelet, a 38 KHz-es közepű bal mínusz jobb sztereojelet, illetve az e fölötti alhordozókat. Végső lépésként a jelet keresztülküldjük egy aluláteresztő szűrőn, majd megtizedeljük, miközben csak a bal plusz jobb hangsávot őrizzük meg. A végfok a hangkártya, 32000 minta/másodperc mintavételi gyakorisággal. A jelfeldolgozással kapcsolatosan a GNU Radio Wikiben lehet további tudnivalókat találni.

Grafikus felületek

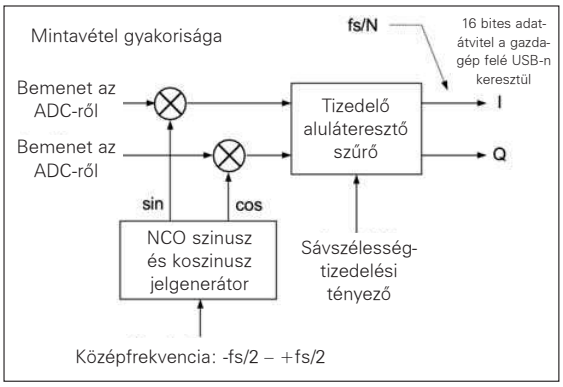
A GNU Radio alkalmazások grafikus felületei Python alatt készülnek, a kezelőfelületek tetszőleges, Python alól elérhető eszközkészlettel összeállíthatók. A géptípusok közötti hordozhatóság elősegítése miatt mi a wxPython használatát javasoljuk. A valós idejű C++ folyamatgráf és a Python-világ közötti kapcsolattartáshoz a GNU Radio megfelelő folyamatközti adattovábbításra alkalmas elemi eljárásokat biztosít.

Gépkövetelmények

A GNU Radio csaknem hardverfüggetlen. Napjaink sok gigahertzes, egyciklusú lebegőpontos egységgel felszerelt szuperskalár processzorai révén még asztali számítógépekkel is elég komoly digitális jelfeldolgozást lehet végezni. Egy 2 GHz órajelű Pentium vagy Athlon processzor 2 milliárd lebegőpontos FIR mintát tud kiértékelni másodpercenként. Majdnem kizárólag programokra alapozva olyan távközlési rendszereket építhetünk, amilyenekre néhány évvel ezelőtt még csak nem is gondolhattunk.



2. ábra Általános programrádió-periféria, USRP



3. ábra Digitális lefelé transzponáló elem

A számítási igény természetesen attól függ, hogy pontosan mit akarunk csinálni, de egy 1-2 GHz-es gép 256 MB memóriával elég kell legyen. Természetesen az analóg világ és a számítógép kapcsolatát is meg kell valahogy teremtenünk. Ha nem akarunk túl sokat költeni, egy jobb minőségű, 96 kHz-es, 24 bites hangkártya megteszi, de végső esetben gépünk beépített hangkártyáját is használhatjuk. Bármelyik megoldást is választjuk, viszonylag keskeny sávban fogunk tudni jelfeldolgozást végezni, és valamilyen keskenysávú RF előtagot is be kell szereznünk.
Egy másik megoldás készen kapható, nagysebességű, PCI foglalatú analóg-digitális átalakító kártya beszerzése. Ezek akár 20 millió mintát is képesek előállítani másodpercenként, ám áruk egy teljes számítógépével vetekszik. Az ilyen nagysebességű kártyákhoz kábelmodemes vevőegységet érdemes RF előtagként illeszteni.

A Universal Software Radio Peripheral

A legjobb megoldás a Universal Software Radio Peripheral (általános programrádió-periféria, USRP). A 2. *ábrán* az USRP blokkdiagramja szerepel. Az USRP Matt Ettus agyszüleménye, lényegében egy rendkívül rugalmas, USB csatolós készülék, amely képes a rádiófrekvenciás világhoz kapcsolni számítógépünket. Az USRP egy kisméretű alaplapot tartalmaz, amely legfeljebb négy darab 12 bites 64 millió minta/másodperces ADC-t, négy 14 bites, 128 millió minta/másodperces DAC-t, egy egymillió kapus, programozha-

2. kódrészlet Egyszerű FM vevő

```
#!/usr/bin/env python

# egyszerű FM vevő

from GnuRadio import *

#
# gr_FlowGraph visszaadása
#
def build_graph (IF_freq):
    input_rate = 20e6

    CFIR_decimate = 125
    RFIR_decimate = 5
    fm_demod_gain = 2200

    quad_rate = input_rate / CFIR_decimate
    audio_rate = quad_rate / RFIR_decimate

    volume = 1.0

    src = GrHighSpeedADCSourceS (input_rate)

    # FIR szűrőminták számítása a csatornaválasz-
    # táshoz
    channel_coeffs = \
      gr_firdes.low_pass (
        1.0,          # erősítés
        input_rate,   # mintavételi gyakoriság
        250e3,        # aluláteresztés vágási
                      # frekvenciája
        8*100e3,      # átvitt sáv szélessége
        gr_firdes.WIN_HAMMING)

    # bemenet: short; kimenet: complex
    chan_filter =
      GrFreqXlatingFIRfilterSCF (CFIR_decimate,
            channel_coeffs, IF_freq)

    # bemenet: complex; kimenet: float
    fm_demod =
      GrQuadratureDemodCF (volume *
                           fm_demod_gain)

    # FIR szűrőminták számítása a hangszűrőhöz
    width_of_transition_band = audio_rate / 32
    audio_coeffs =
      gr_firdes.low_pass (
        1.0,          # erősítés
        quad_rate,    # mintavételi gyakoriság
        audio_rate/2 - width_of_transition_band,
        width_of_transition_band,
        gr_firdes.WIN_HAMMING)

    # bemenet: float; kimenet: short
    audio_filter =
      GrFIRfilterFSF (RFIR_decimate,
                      audio_coeffs)

    final_sink = GrAudioSinkS ()

    fg = gr_FlowGraph ()

    fg.connect (src, chan_filter)
    fg.connect (chan_filter, fm_demod)
    fg.connect (fm_demod, audio_filter)
    fg.connect (audio_filter, final_sink)

    return fg

if __name__ == '__main__':

    # csatlakozás az RF előtaghoz
    rf_front_end = microtune_eval_board ()
    if not rf_front_end.board_present_p ():
        raise IOError, 'RF előtag nem található'

    # az erősítés és a rádióállomás frekvenciájá-
    # nak beállítása
    rf_front_end.set_AGC (300)
    rf_front_end.set_RF_freq (100.1e6)

    IF_freq = rf_front_end.get_output_freq ()
    fg = build_graph (IF_freq)
    fg.start ()     # szál(ak) létrehozása
                    # fork hívásokkal és
                    # visszatérés
    raw_input ('A kilépéshez nyomd le az Enter
              gombot: ')
    fg.stop ()
```

tó logikai tömböt (FPGA) és egy programozható USB 2.0 vezérlőt tartalmaz. Egy teljes kiépítésű USRP alaplap négy bővítőkártya használatát teszi lehetővé: kettő vevő- és kettő adókártyáét. Az RF előtagok a bővítőkártyákon találhatók. A különféle frekvenciasávok kezelésére számos bővítőkártya létezik. Amatőr rádiósok számára a kis energiájú bővítőkártyák a legmegfelelőbbek, ezek a 440 MHz-es és az 1,24 GHz-es sávban adnak-vesznek. Létezik csak vételre használható, kábelmodemes vevőegységre épített bővítőkártya is, amely az 50 MHz – 800 MHz tartományt kezeli. A bővítőkártyákat úgy tervezték, hogy egyénileg, kézzel is könnyen összeállíthatók, ezzel is segítik az érdeklődőket a kísérletezésben.

Az USRP rugalmassága egyrészt a két programozható elemből, másrészt a gazdagépen futó könyvtárral folytatott párbeszédből fakad. Ha ízelítőt akarunk az USRP képességeiből, elég megvizsgálnunk indításának folyamatát. Maga az USRP nem rendelkezik ROM-ba írt belső programmal, csu-

pán néhány bájtnyi gyártó- (VID) és termékazonosítóval (PID), valamint változatszámmal. Amikor az USRP-t először csatlakoztatjuk az USB-kapura, a gazdagépen futó könyvtár egy beállítások nélküli készüléket lát – a beállítások hiányát a VID, a PID és a változatszám kiolvasásával észleli.
A könyvtár első lépésként a 8051 kódot tölti le, ez határozza meg az USB-vezérlő viselkedését. Amikor ez a kód elkezd futni, az USRP leválik az USB-sínről, majd újracsatlakozik hozzá. Az újracsatlakozás után a gazdagép már egy másik eszközt lát, a VID, a PID és a változatszám ugyanis megváltozik. A most már meglévő és futó belső program megadja az USB végpontokat, felületeket és parancskezelőket. Az USB-vezérlőnek kiadható parancsok egyike az FPGA feltöltése. A könyvtárkód, miután érzékelte az USRP új eszközként való újracsatlakozását, a beállítási folyamat következő fázisára lép, és áttölti az FPGA beállítására szolgáló bitfolyamot.
Az FPGA-k általános célú lapkák, működésüket a beléjük írt beállító bitfolyam határozza meg. A bitfolyamot leginkább objektumkódként, a kívánt működés magas szintű leírásának fordítási kimeneteként képzelhetjük el. Esetünkben a kívánt működést a Verilog hardverleíró nyelven fogalmaztuk meg. Ez a kód is forrásként, a GNU Radio többi részéhez hasonlóan GNU GPL alatt érhető el.

Mi történik az FPGA-ban?

Az FPGA egy kisméretű, erősen párhuzamos számítógépként fogható fel, aminek működését az adott feladat alapján mi tervezhetjük meg. Programozása igényel némi tapasztalatot, ugyanis ha elrontjuk, akkor véglegesen használhatatlanná tehetjük a lapkát. Éppen ezért tettünk közzé egy általános beállítást, amely alkalmazások széles köréhez használható.
Egy jó USB-gazdavezérlővel az USRP folyamatos 32 MB/másodperces átvitelre képes az USB-sínen keresztül. Az USB-sín váltakozó kétirányú átvitelre használható, a 32 MB/másodperc kapacitást szükség szerint oszthatjuk el a vételi és az adási irány között. A normál beállítás lehetővé teszi, hogy a vételi irányban kiválasszuk a digitalizált spektrum minket érdeklő részét vagy részeit, majd ezeket szükség szerint alapsávú jellé alakítsuk vagy tizedeljük. Mindez egyenértékű azzal, ami az RF előtagnál történik, leszámítva, hogy ebben az esetben digitalizált mintákkal dolgozunk. Azt a kódrészletet, amely ezt a feladatot látja el, digitális lefelé transzponáló elemnek nevezzük. (*3. ábra*) Mindezt digitális tartományban végezzük el, aminek az előnye, hogy azonnal meg tudjuk változtatni a középfrekvenciát, ami elsősorban frekvenciaugrásos szórt spektrumú rendszereknél hasznos. Adási irányban ennek pontosan az ellenkezője történik. Az FPGA több digitális lefelé és felfelé transzponáló elemet tartalmaz, ezek szükség szerint azonos és eltérő ADC-khez is csatlakoztathatók. A mögöttük álló elméletet itt nem áll módunkban kifejteni, a GNU Radio Wikiben azonban minden szükséges tudnivaló megtalálható.

GNU Radio alkalmazások

A már említett példákon túl a GNU Radio részeként teljes értékű HDTV adóhoz és vevőhöz, spektrumelemzőhöz, oszcilloszkóphoz, párhuzamos többcsatornás vevőhöz és persze modulátorok és demodulátorok folyton bővülő gyűjteményéhez is hozzájutunk.

A jelenleg tervezés vagy megvalósítás alatt álló rendszerek közül néhány példa:

- A TiVoval egyenértékű rádiós megoldás, több csatorna egyidejű rögzítésének lehetőségével
- Passzív radarrendszer, amely a TV-adásokat használja jelforrásként.
  Akik antennán keresztül nézik a tévéadásokat, csak gondoljanak arra, hogy ha repülőgép halad el felettük, ugrálni kezd a kép.
- Rádiócsillagászat
- TETRA adó-vevő
- Digitális világrádió
- Program alapú GPS
- Elosztott érzékelőhálózatok
- Spektrumhasználat elosztott mérése
- Amatőr rádió adó-vevők
- Alkalmi hálós topológiájú hálózatok
- RFID érzékelő/olvasó
- Több bemenetű több kimenetű (MIMO) jelfeldolgozás

Politikai kérdések

A szabadon elérhető, rádiók építésére alkalmas programok nem mindenkinek tetszenek. Az Amerikai Egyesült Államokban például az Amerikai Mozgókép Szövetség ellenállásába futottunk bele, ők a Broadcast Flag (másolásvédelmi kiegészítő jel a digitális adásokban) alkalmazásával próbálják korlátozni a földi sugárzású digitális tévéadásokhoz épített vevőkészülékek körét.
Az Amerikai Szövetségi Kommunikációs Bizottság egy előzetes állásfoglalást adott ki az kognitív képességekkel rendelkező rádiós megoldásokra és a program alapú rádiókra vonatkozóan.
Az állásfoglalásban számos problémát vetnek fel, felmerül többek közt a nagysebességű digitális-analóg átalakítók értékesítésének korlátozása, az engedély nélküli programok távol tartása a program alapú rádiózáshoz szükséges vasaktól digitális aláírások vagy hasonló eszközök segítségével és új korlátozások bevezetése az amatőrök számára gyártott rádiókra vonatkozóan.

Összegzés

A program alapú rádiózás érdekes terület, a GNU Radio minden eszközt rendelkezésünkre bocsát, hogy megkezdhessük felfedezését.
Aki komoly tudást akar szerezni a program alapú rádiózásban, annak több szakterületet is meg kell ismernie. Mi csak annyit ígérhetünk, hogy mindent megteszünk az első lépések megkönnyítése érdekében.

Linux Journal 2004. június, 122. szám

Eric Blossom a GNU Radio Project alapítója. Mielőtt program alapú rádiózással kezdett volna foglalkozni, évekig a biztonságos telefonszolgáltatások területén dolgozott. Amikor éppen nem program alapú rádiót bütyköl, valószínűleg jógázással vagy ju-jitsuval tölti az idejét. Az eb@comsec.com címen érhető el.

„Szerkeszd te is a Linuxvilágot!” – eredmények

Tavasszal kérdőív kitöltésére kértük olvasóinkat honlapunkon, amelyet nagy örömünkre sokan kitöltöttek. A vélemények több kérdésben meglehetősen megosztottak voltak, de így is rengeteg hasznos információval szolgáltak.

Szerepeltek benne kérdések a válaszolók demográfiai adataival (nem, lakhely, foglalkozás stb.), a linuxhoz fűződő viszonyukkal és felhasználói szokásaikkal kapcsolatban.
Minket azonban a kérdések másik csoportjára adott válaszok érdekeltek igazán – azok, amelyekből pontosabban feltérképezhetjük olvasóink igényeit. A kérdőív segítségével azonban nagy mennyiségű és objektíven értékelhető információhoz jutottunk. Összesen 510-en töltötték ki a kérdőívet, ami már értékelhető mintával szolgál ahhoz, hogy képviselje olvasóink véleményét.
Terjedelmi korlátok miatt itt most csupán egy rövidebb áttekintéssel szolgálunk, amely ismerteti a felmérés eredményét. Ha valaki kíváncsi a teljes anyagra, elolvashatja honlapunkon.

Általános adatok

Olvasóink túlnyomó része férfi, ám nők is olvasnak minket (1.5%)?. Az átlag életkor 29 év, azonban szinte minden korosztály képviselteti magát: legfiatalabb olvasónk 13, a legidősebb 67 éves. Lakhelyüket tekintve az olvasók nagyobb része vidéki településen él (58%).

Iskolai végzettség és foglalkozás

A válaszadók 56% felsőfokú végzettségű, vezető beosztásban 10% dolgozik. A legtöbben szellemi foglalkozásúak (57%), a fizikai dolgozók aránya csekély (3%). Viszonylag magas a diákok (22%) és a vállalkozók (16.5%) aránya.
Szakmájukat tekintve a rendszergazdák, programozók és rendszermérnökök vannak a legtöbben (összesen 40,5%).
A munkahelyre legjellemzőbb ágazatok a számítástechnika (47%), az oktatás (14%), és a kereskedelem (8%). A számítástechnikán belül a programfejlesztés (25.5%) és az oktatás (15.8%) van a legnagyobb arányban.

Linux a cégeknél

Biztató a Linux elterjedtsége a válaszadók munkaadó cégeinél: túlnyomó részüknél (86.7%) alkalmazzák a Linuxot valamilyen formában, többségüknél szerver és kliens gépeken is (46.4%).
Olvasóink nagyobb része Linuxos ismereteit tekintve a „haladó” kategóriába sorolható, de a kezdők is elég magas arányban vannak. Reméljük, hogy az az 1.1% aki „csak hallott” róla, hamarosan kipróbálja, és ebben nekünk is van némi szerepünk.

Milyen operációs rendszert használsz?

Furcsa lett volna, ha itt nem a Linux kapott volna elsöprő többséget (96%), ám érdekes, hogy a Windows változatait együttvéve megközelítőleg ugyanannyian használják (92%). Ebben nincs semmi különös, hiszen gyakori, hogy valaki mindkettőt használja. Előfordulhat persze, hogy nincs is más választása a felhasználónak – például ha csak a Windows-on elérhető programra van szüksége adott feladathoz, vagy a munkahelyén csak ez elérhető. Ugyanakkor van még egy idetartozó csoport – az olvasói levelekből kiderül, hogy sokan még nem próbálták ki, de nagyon érdekli őket a Linux és fontolgatják a váltást.
A BSD és a UNIX rendszerek is aránylag jó helyezést kaptak – együttesen 26%-ot. A Linux terjesztések közül az egyértelmű kedvenc a Debian, a dobogó két másik helyezettje az UHU és a SuSE. Sokan megjegyezték, hogy szívesen próbálgatnak, váltogatnak több változatot is.

Hogyan szerzed linuxos tudásod?

A válaszok alapján egyértelműen az Internet a legnépszerűbb forrás (81.7%), de sokan használnak könyveket is (63.3%) linuxos tudásuk elmélyítésére. Emellett valószínű, hogy a többség hasznosnak tartja a linuxos levelezőlistákat is: több mint kétharmaduk már feliratkozott valamelyikre.
Az újság tartalmát és küllemét firtató kérdésekre adott válaszok aránya alapján a többség olyan igényes kivitelű, színvonalas tartalmú lapnak tartja a Linuxvilágot, amely a kezdő és a haladó felhasználók számára is hasznos.

Kedvenc cikksorozat, rovat, szerző

A cikksorozatok, a rovatok és a szerzők értékelése között nem mutatkozott különösen nagy eltérés – mindegyik jó átlaggal végzett, csupán egy-kettő kapott kiemelkedő pontszámot.
A sorozatok közül ***Büki András*** Héjprogramozás című cikkei kapták a legmagasabb pontszámot, utána ***Mick Bauer*** OpenSSH-ról szóló és ***Szy György*** Debianos sorozata következett. A legnépszerűbb rovatnak a Szaktekintély, a Vezér-

fonal és a Kovácsműhely bizonyult. Az olvasók kedvenc külföldi szerzői: ***Marcel Gagné***, ***Mick Bauer*** és ***Hans Reiser***. A hazai szerzők közül ***Borbély Zoltán***, ***Büki András*** és ***Csontos Gyula*** kapta a legtöbb szavazatot.

Magyarítás

Ez kiemelten fontos kérdés volt számunkra, mivel az évek során számos kritika érte a szerkesztőséget a magyarított kifejezésekkel kapcsolatban. A válaszokból kiderült, hogy sokan egyetértenek azon törekvésünkkel, hogy egységes szóhasználatot alakítsunk ki. Összességében meglehetősen megosztotta a válaszadók táborát ez a téma: 53%-uk kifejezetten pozitívan értékelte a kérdést (bár ebből 25% jelezte, hogy néha zavarja a magyarítás), 22%-uk a magyarítás ellen szavazott (ebből 13% képviselte a radikális tábort, akik a semmit nem fordítanának le), és 25% vélte úgy, hogy a kompromisszum volna az üdvözítő megoldás, vagyis minden szakkifejezés mellett az eredeti feltüntetése.

Cd meléklet

Ha volna egy olyan CD-ROM, ami a Linuxvilág eddig megjelent számok teljes archívumát tartalmazza (2000-2003), megvennéd-e és mennyit volnál hajlandó áldozni érte?
A válaszok alapján arra következtethetünk, hogy érdemes lenne egy archív CD-t összeállítani, ugyanis az olvasók közel 80%-a megrendelné, 1000 illetve 2000 Ft-ért.

Ár

Mi a véleményed a Linuxvilág áráról?
Ebben a kérdésben is jelentősen megoszlottak a vélemények: a válaszolók többsége (56%) úgy vélte, hogy elfogadható az ára a tartalmát és minőségét tekintetbe véve. További 13%-uk szerint is megéri az árát, de nem engedheti meg magának. 22% túl drágának tartotta.

Mi az ami nem tetszik, mit változtatnál? (szabad-szöveges válaszok)
Erre a kérdésre a kérdőívet kitöltők jelentős hányada válaszolt (35%) és közülük sokan részletesen is kifejtették véleményüket. Az ilyen „szabad-szöveges" válaszok kiértékelése ugyan jóval nagyobb munkát igényel és nehéz számokban kifejezni az eredményét, ezekből is számos értékes véleményt, ötletet és javaslatot nyertünk.
Itt természetesen lehetetlen részletesen bemutatni a válaszokat, ezért egy rövid összefoglalót adunk arról, hogy melyek voltak a leggyakrabban felvetett témák.
Örömmel olvastuk, hogy igen sokan elégedettek a lappal – erre utalt a rövid, de pozitív „ilyen nem jut eszembe", „minden klassz, csak így tovább" típusú válaszok magas aránya.
A magyarítás itt is előkelő helyen szerepelt, volt aki kimerítően fejtegette mi zavarja ezzel kapcsolatban, volt aki a zárójeles megoldást javasolta, és volt, aki ékes tömörséggel csupán annyit írt: „a magyarítás monnyon le". Azért volt ellenvélemény is: „Engem a magyarítások nyertek meg a lap vásárlójának. Köszönöm!"
Többen megjegyezték, hogy szerintük némely cikkek „felületesek" és sokkal több gyakorlatias megközelítésű cikket szeretnének látni a lapban. Megközelítőleg ennyien jelezték, hogy a CD-melléklet felesleges vagy tartalma ritkán hasznos számukra.
Néhányan szeretnék, ha vastagabb lenne a lap, több cikk lenne programozás és hardver témában.

Kiemelten fontos kérdések

A magyarítás problémája

Nos, talán ez az a téma, ami a leginkább felkorbácsolta az olvasói kedélyeket. A levelekből, fórumok hozzászólásaiból és a kérdőív válaszaiból is kitűnik, hogy vannak, akik nem értenek egyet lapunk magyarítási törekvéseivel. Sőt, többen úgy érzik, hogy erőszakosan magyarítunk minden idegen szót. Az angol szakkifejezések magyarítása meglehetősen ellentmondásokkal teli feladat, amely a szerkesztőségünknek is gyakran fejtörést okoz.
Sokan azt hihetik, hogy ez az egész magyarítás egy hóbort csupán, amivel bosszantani akarjuk a hozzáértő olvasókat, valójában csak arra törekszünk, hogy kialakítsunk egy egységes és következetes nyelvezetet. Természetesen elkerülhetetlen, hogy eközben néhol hibát vétünk és erőltetett eredmények születnek, de úgy véljük ez vele jár és idővel a félresikerült kifejezések kiszóródnak, felváltják helyüket jobbak. Nem lehet "félig" fordítani (persze lehet, de szerintünk az rosszabb). Ha az úgymond "haladó" kifejezéseket egyáltalán nem fordítanánk, akkor a következetesség elvét elfelejthetnénk és igen felemás szöveg keletkezne. Így a szakemberek számára viszont gyakran zavaróak a magyarított kifejezések. Annak érdekében, hogy enyhítsünk ezen a gondon, eddig is tettünk lépéseket: bizonyos szavak mellett feltüntettük zárójelben az eredeti angolt is, és ezért indította el a főszerkesztőnk, Szy Gyuri a Szótár rovatot.
Összefoglalva: bár a válaszadók jelentős részének tetszik a lap nyelvezete, az eddigi/jelenlegi Linuxvilágban megjelenő magyarítással sok olvasónk valamilyen mértékben elégedetlen. Nyilvánvalóan nem létezik olyan változat, ami minden olvasónknak tetszik, de mi úgy véljük a következő megoldás elfogadható a többség számára és a mi sem kényszerülünk elveink feladására: a jövőben minden olyan kifejezés mellett, amelynek nincs széles körben elfogadott magyar megfelelője (illetve ha az egyértelműség úgy kívánja), zárójelben, dőlt betűvel feltüntetjük az angol eredeti szót.
A különösen problémás kifejezések listáját pedig megjelentetjük a magazin honlapján, ahol lehet javaslatokat beküldeni, illetve ezeket értékelni.
Nagy segítséget jelent, ha a kritika mellett több építő javaslatot is kapunk a szavak magyar megfelelőjére.

CD-melléklet és a lap ára

A kérdőív és a honlapon közzétett szavazás eredménye is megerősíti, hogy sokan vannak, akik a CD-mellékletet jelenlegi formájában feleslegesnek tartják, vagy használnák, de gyakran nem olyan programokat tartalmaz, amely hasznos számukra. Bár többen kényelmes dolognak tartják, hogy CD-ről is telepíthetik a programokat, a többség inkább az Internetet választja.
Emellett a lap árát is viszonylag sokan drágállották. Igaz jelentős részük úgy véli, hogy a tartalmát és színvonalát tekintve megéri az árát.
Noha a többi számítástechnikai laphoz nincs értelme hasonlítani. Erre most nem térnék ki bővebben, csupán néhány tényt említenék: a magazin nem egy általános in-

formatikai lap, hanem egy szűkebb réteghez – a Linux iránt érdeklődőkhöz – szól, ebből adódóan alacsony a példányszáma és kevesebb hirdetőt vonz. Igaz, hogy a hazai piacon drágább lapok közé tartozik. Ugyanakkor nem titok, hogy igen szoros költségvetéssel és csekély létszámú szerkesztőséggel működik. Mindebből következik, hogy az árát nem áll módunkban csökkenteni, hacsak nem módosítunk jelenlegi formáján. Sokat törtük a fejünket, hogy mi lehetne az üdvözítő megoldás. Egyelőre nem született végső döntés csak egy tervezet állítottunk össze, aminek lényege a következő: A fentiek alapján kézenfekvő megoldásnak tűnik a CD-melléklet átalakítása, ezért terveink szerint az a jövőben a következő formában működne. Az újságárusoknál CD-melléklet nélkül lenne kapható a Linuxvilág, alacsonyabb áron. Nem szűnne meg, csak különválasztanánk a laptól – lényegében egy független szolgáltatásként működne.

Akinek fontos a CD-melléklet, továbbra is hozzájuthatna: megrendelhetné a lapban vagy a honlapon elérhető megrendelőlappal. Sőt, több lehetőség között is választhatna, mivel nem csupán egy CD-t állítanánk össze. Ezeket a „CD-csomagokat" folyamatosan a vevők igényeihez igazítanánk (pl. szavazni lehet majd, hogy milyen CD-k legyenek elérhetők).

Az árcsökkenés miatt a jelenlegi előfizetőket a következő módon kárpótolnánk: előfizetésük időtartama meghosszabbodna, a fél éveseké egy, az egyéveseké két számmal.

Mint említettük, ez csupán egy tervezet, amelynek megvalósításához alaposabban ki kell dolgoznunk a részleteket. Ezért a végső döntés előtt egy újabb, kifejezetten erre a kérdésre vonatkozó, rövid felmérést végzünk. A honlapon elérhető lesz egy szavazás, amiben olvasóink jelezhetik, hogy egyetértenek-e a fenti tervezettel, illetve szerintük mi volna a jobb megoldás. Az olvasók véleménye döntő fontosságú ebben a kérdésben, ezért kérünk mindenkit, hogy adja le voksát honlapunkon az új CD-mellékletről szóló szavazásban!

Amin változtatni szeretnénk – összefoglaló

Tartalom folyamatos szerkesztése a visszajelzések alapján

- A többség szerint felhasználói szintet tekintve megfelelő a cikkek aránya, de a „haladó" tartalomra némileg nagyobb az igény – ezt figyelembe vesszük a jövőben.
- A honlapon feltüntetjük egy listában, hogy milyen témákra van különösen igény .
  - Javasolhatnak más témákat is, ha érdemesnek tartjuk, felvenni a listába.
  - A listában szereplő témákra folyamatosan lehet szavazni.
- A cikkek/szerzők/rovatok minősítésére lehetőséget adunk a honlapon is:
  - Az éppen aktuális szám cikkeit lehet pontszámmal és hozzászólással értékelni.
  - Ennek eredménye alapján toplistát adunk közre.

Honlap

- Sokan látogatják, töltenek le cikkeket és hasznosnak találják ezért nyár végéig elérhetővé tesszük a 2003 decemberig megjelent összes cikket.
- Gyakrabban frissítjük a honlap tartalmát.
- Saját fórumot nyitunk minden hazai szerzőnknek, ahol az olvasók megvitathatják a cikkeket.

Archív CD megjelentetése

- 2000-2003 évfolyamok összes cikkét tartalmazza PDF formátumban.
- Az ára 1500 Ft lesz, a honlapon keresztül lehet megrendelni.
- Az előfizetők ingyen kapnak egy példányt.

Magyarítással kapcsolatos módosítások

- Az adott cikkben, ahol először jelenik meg, minden kifejezés mellett zárójelben, dőlt betűvel feltüntetjük az angol eredeti szót.
- „Mini-szószedet" jelenik meg külön „dobozban" az adott oldalakon.
- Olvasóink is segíthetnek a magyarításban értékeléssel és javaslatokkal:
  - A különösen problémás kifejezéseket feltüntetjük a honlapon.
  - Beküldhetnek változatokat, amiket feltüntetünk mellettük és ezeket pontszámmal lehet értékelni.
  - Időnként a pontozás alapján felülvizsgáljuk, hogy érdemes-e lecserélni az általunk használt szót.

Amit tervezünk

CD-melléklet és az ár változására vonatkozó tervezet

- Az újságosoknál kapható Linuxvilág nem tartalmazna CD-t.
- Külön lehetne megrendelni a honlapon, faxon vagy levélben.
- Több féle tartalmú CD-t lehetne rendelni, amelyek tartalmát az olvasók aktuális igényihez igazítjuk – ezekről szavazni lehetne a honlapon .
- A magazin cikkeihez tartozó anyagok (kódok, beállítófájlok, programok, stb.) letölthetők lennének a honlapunkon.
- A lap bolti ára kb. 1700 Ft-ra csökkenne.
- Az előfizetési díj félévre is csökkenne, jelenlegi előfizetőink előfizetése meghosszabbodna: féléves előfizetőké egy, az egyéves előfizetőké két számmal.

Nyertesek

A kérdőívet kitöltők közötti sorsolás eredménye

Linuxvilág pólót kapnak ajándékba:
Tibor Tótok – Budapest
Horváth Attila – Budapest
Pék János Dániel – Nyíregyháza
Debre Zoltán – Budapest
Szabó János – Budapest

Tux plüsspingvint kapnak ajándékba:
Hlács Zoltán – Tárnok
Kocsis Attila – Hajdúböszörmény
Biller Bálint László – Budapest
Kiss Márton – Budapest
Zsédely Balázs – Győr

Őrületes ez a demokrácia!

Manapság rendkívül népszerűek a honlapokon találhat olvasói szavazások. Próbáljunk ki mi is egy egyszerű csoomagot, és bővítsük ki honlapunkat egy szavazóládával.

François, ne mondd, hogy semmi közöd ehhez, úgysem hiszem el. Ezek a számokat mintha valaki megpiszkálta volna, aminek következtében erősen feléd húznak. Általában nem kérdőjelezem meg a tisztességedet, mon ami, de hát az étterem még csak most nyitott és a vendégeink sem érkeztek meg.
Vajon mivel tudod akkor megmagyarázni a rengeteg begyűjtött szavazatot? Quoi?
Igen, François, látom. Látom, hogy a vendégek már megérkeztek, de néhány perce még nem voltak itt. Most már mindegy, nyomás a boros pincébe, de immédiatement. Hozz egy 2001 Sonoma County Zinfandel-t.
De nehogy azt hidd, hogy ezzel el van intézve a dolog. Erről még később beszélünk. Vite!
Üdvözlet mindenkinek Chez Marcel portáján, az ízletes Linux falatok otthonában, a világ legfinomabb borospincéjében, valamint a működő elektronikus demokrácia világában – Linux módra, persze.
Amíg hűséges pincérem felhozza a bort, had szóljak pár szót a ma esti menüről, amely remélhetőleg elnyeri majd az én kedves vendégeim tetszését.
Noha kétségtelen, hogy az elektronikus demokrácia még csak gyerekcipőben jár, de az is biztos, hogy tett már egy két reménykedésre okot adó lépést. Lehetőség van arra például, hogy Linux rendszerünkkel és egy Apache Webkiszolgálóval felvértezve megismerkedjünk a hálózati szavazás rejtelmeivel.
Oh, François, visszatértél. Kérlek tölts a vendégeknek, míg én elmesélem első kísérletünket az elektronikus demokrácia világában, és felvetek egy kérdést, mely oly kedves a te szívednek. Demokrácia.
Mes amis, kedves barátaim, biztosan láttak már különféle hálózati szavazóeszközökkel ellátott weblapokat, melyek a komolytól az egészen mókásig terjedő kérdésekkel nyaggatnak bennünket.
Kis erőfeszítéssel saját weblapunkat is felszerelhetjük ilyesmivel.
Étlapunk első fogása ***Michael*** és ***Erin Spiceland My Voting Script***-je (lásd a hálózati források részt). Használatához nem is kell mást tennünk, mint, hogy a cgi-bin könyvtárunkban kitömörítjük a tarlabdát (tarball). Nálam ez a */usr/local/apache/cgi-bin* könyvtár:

1. ábra Egyszerű My Voting Script szavazóláda, amelyen éppen kedvenc pincérem fizetésemelése felől szondáztatom a közvéleményt. Persze nem csak eldöntendő kérdéseket tehetünk fel.

```
cd /usr/local/apache/cgi-bin
tar -xzvf vote-2.1.tar.gz
```

Ezáltal létrehoztuk a vote nevű könyvtárat. Most lépjünk be a könyvtárba és nyissuk meg a ***sitevariables.pl*** nevű állományt. Több paramétert is beállíthatunk itt, többek közt felhasználónak megjelenített szavazólap színeit. Ne siessünk, mert érdemes egy kis időt szánnunk erre a részre, hogy fedezhessük, miféle trükkökre van itt lehetőség. A legérdekesebb rész persze mindenki számára a kérdések és lehetséges válaszok megadása:

```
$question = ´Kapjon François fizetésemelést?´;
@answers = (´Igen´,´Nem´);
```

A kérdésnek nem kell szigorúan igen vagy nem kérdésnek lennie. Ilyesmit is felvehetünk: `@answers = ('Igen','Nem', 'Talán');`. Nos, mielőtt lefuttatnánk ezt a François fizetéséről szóló szavazást, nézzünk meg néhány fontos paramétert:

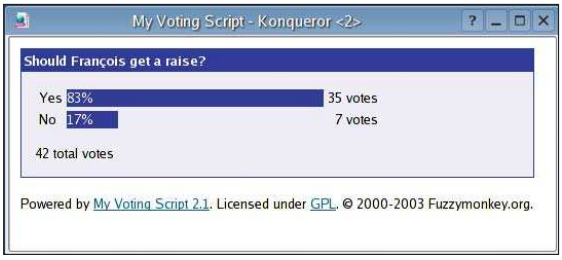
```
$lockhours = 0;
$url = ´http://yourserver/cgi-bin/vote/index.cgi´;
```

A $lockhours paraméterrel adhatjuk meg, hogy egy adott IP cím mennyi ideig várakozzon mielőtt újabb szavazatot

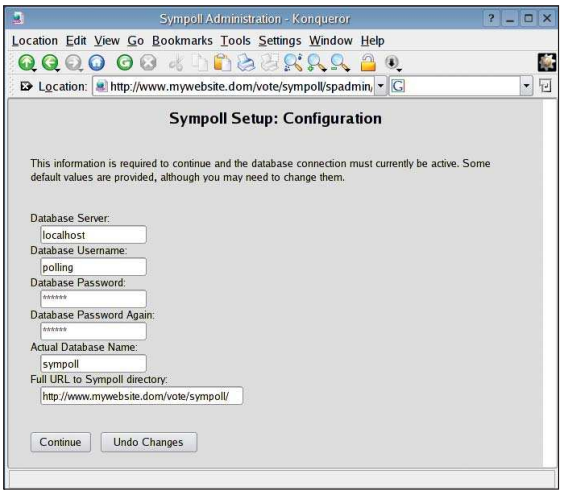
fogadnánk el róla. Ha nullára állítjuk, mint az fent látható, akkor ugyanazon IP címről tetszőleges mennyiségű szavazatot lehet leadni.
Most már csak ellenőriznünk kell, hogy a kiszolgálónk címét és a parancsfájlra mutató útvonalat helyesen állítottuk-e be. Az új szavazóládánk kipróbáláshoz irányítsuk böngészőnket az ***index.cgi*** állományra: ➲ http://yourserver/cgi-bin/vote/index.cgi.
A szavazó kérdésnek most meg kell jelennie az ablakban , ahogy azt az *1. ábrán* láthatjuk is.
A szavazat elküldéséhez egyszerűen csak kattintsunk valamelyik válaszra, jelen esetben az Igen-re vagy a Nem-re. Miután szavaztunk, a pillanatnyi eredményt tartalmazó felugró ablak jelenik meg. A szavazatok százalékos arányát színes oszlopok mutatják, grafikusan is megjelenítve az eredményt. Amint azt a *2. ábrán* láthatjuk, hűséges pincéremnek lenyűgöző mértékű támogatást sikerült begyűjtenie.
A script elég egyszerű, és egy időben csak egy kérdést tud kezelni. Ha ennél valamivel nagyobb tudású dolgot keresünk, nézzük meg ***David S. Raeman Sympoll*** programját. Mint a neve is mutatja, egy egyszerű szavazórendszerről van szó, bár ez esetben az egyszerűség inkább a beállítási módra vonatkozik, és nem a használhatságra.
A telepítés mindössze néhány másodpercet vett igénybe, utána minden egyszerűen működött. Ez az apró szavazócsomag lehetővé teszi egyszerre több szavazás indítását is, melyekben szavazatonként egynél több választásunk is lehet, aminek köszönhetően a lehetséges válaszok meglehetősen összetettek is lehetnek.
A szavazatokat és a hozzá tartozó információkat MySQL adatbázis fájlban tárolja a rendszer. A programhoz szükségünk lesz egy bekapcsolt PHP támogatással futó Apache kiszolgálóra, valamint természetesen egy futó MySQL rendszerre.
Tömörítsük ki a ***Sympoll*** forrását (lásd a forrásokat) a webkiszolgálónk dokumentum könyvtárának megfelelő alkönyvtárába. Nálam például a dokumentum gyökér a */usr/local/apache/htdocs*, ennek megfelelően a terjesztés forrását ebbe a könyvtárba tömörítettem ki a `tar -xzvf sympoll-1.5.tar.gz` paranccsal. Miután a könyvtár helyére került, a `chmod 660 config.php` parancs végrehajtásával biztosítsuk, hogy a ***config.php*** állományt módosíthatja az apache kiszolgáló. Mielőtt azonban belépnénk a böngészőbe, érdemes megjegyezni, hogy a szerző szerint érdemes az admin könyvtár nevét valami másra változtatni, hogy nehezebb legyen rátalálni.
Ezután a következő parancs segítségével létrehoztam a ***sympoll*** nevű MySQL adatbázist (bár ez a név tetszőleges lehet):

```
mysqladmin -u root -p create sympoll
```

Nem szerettem volna ha a root felhasználó kezel egy hálózati elérést – ez ugyanis nem valami biztonságos megoldás –, éppen ezért létrehoztam a polling nevű felhasználót. Ehhez először is beléptem a MySQL parancssorába a `mysql -u root -p` utasítással. Ezt követően a ***mysql> prompt***-ban a következő sorral létrehoztam a Sympoll tábláit kezelni tudó felhasználót:



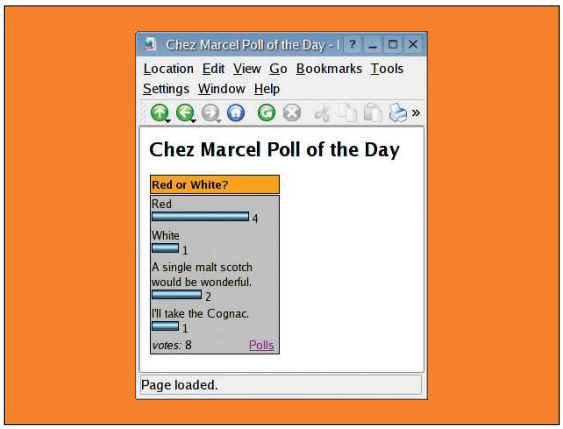
2. ábra François-nak van oka örülni: kezdetben a szavazás ígéretesen indul, a vendégek több mint négyötöde szerint érdemelne emelést.



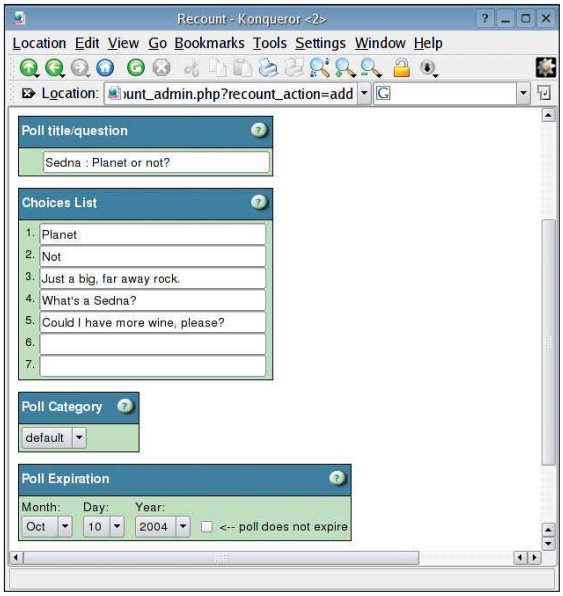
3. ábra A Sympoll beállítása, nos, egyszerű: barátságos és könnyen kezelhető rendszergazdai felület, melynek kezelését megtanulni igazán nem valami nagy ördöngösség.

```
mysql> grant all privileges on sympoll.*
  -> to polling@localhost identified by ´secr3t´;
```

Persze ügyeltünk rá, hogy eltérő felhasználó nevet és jelszót használjunk, ugye? Nem szeretnénk ha éttermünk minden látogatója ilyen könnyen kitalálható jelszóval kerülne szembe. A következő lépés, az új adatbázisunk tábláinak elkészítése, amit a rendszergazdai felületen keresztül tehetünk meg. Böngészőnket irányítsuk a kiszolgálónk ***Sympoll***-t tartalmazó könyvtárára:➲ http://yourhost/sympoll/admin (*3. ábra*).
A rendszergazdai felület barátságos és lépésről lépésre végigvezet bennünket mindenen. Először meg kell adnunk a MySQL felhasználónevet és jelszót. A rendszerbe négy új tábla kerül be (***sympoll _data***, ***sympoll_list***, ***sympoll_auth*** és a ***sympoll_iplog***). A táblák létrehozását jóvá kell hagynunk. Kattintsunk a ***Continue*** gombra mígnem a ***super administrator configuration*** lapra nem kerülünk. Ez lesz majd a Sympoll rendszergazda, tehát az a felhasználó aki a többi rendszergazdát és a szervezett szavazásokat létrehozhatja. Gépeljünk be egy felhasználónevet és jelszót, érvényesítsük a jelszóválasztásunkat, és kész is vagyunk. Máris a bejelentkező lapon találjuk magunkat, gyakoroljuk hát egy picit a demokráciát. A főmenüvel rendkívül egyszerű dolgozni. Felvehetünk,



4. ábra A Vörösbor jól áll a szavazás kezdetén. Amint az látható, ekkorra már Roger is, és Gar is jelezte, hogy milyen italokat tartana feltétlenül szükségeshez egy ilyen vacsorához.



5. ábra Új Recount szavazóláda készítése. Benjamin D. Jones keze munkáját dicséri a program, amely egyaránt használhat MySQL és PotgreSQL adatbáziskezelőt.

módosíthatunk és törölhetünk létező szavazásokat, felvehetünk vagy eltávolíthatunk admin felhasználókat, megváltoztathatjuk a megjelenítési paramétereket és naplókat, továbbá néhány egyéb, egyszerűen átírható opciót. Maguk a szavazások sem statikusak. Képzeljünk el egy szavazást ahol fehér vagy vörös bor közt választhatunk. Aztán jön Roger a kettes asztaltól, aki szeretné a konyakot is felvetetni a listába, míg a tizenhatos asztalnál ülő Gar szerint elfogadhatatlan, hogy egyetlen single malt whisky sem szerepel a lehetőségek közt. Semmi gond. Végül is felvehetünk új elemeket és folytathatjuk a szavazást.
Miután felvettünk egy szavazási kategóriát, valami ilyesfajta kódot kell látnunk:

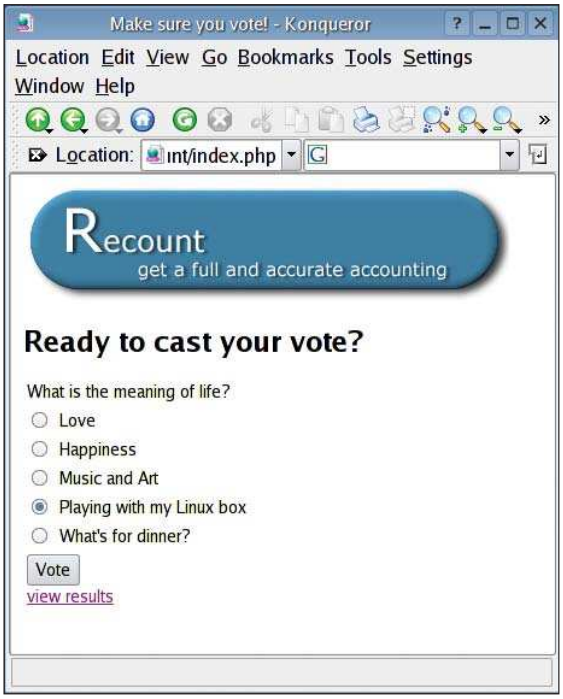
```
<?php require
 ´/usr/local/apache/htdocs/sympoll/booth.php´;
 display_booth(2); ?>
```

Ezt bármilyen PHP Weblapba beilleszthetjük és már készen is vagyunk. A *4. ábrán* megtekinthetjük, milyen eredményre számítsunk.
Érdemes egy kicsit megismerkednünk a kiegészítő beállítási opciókkal. Az alapértelmezett színek és a szavazólap megjelenése ízlésünkhöz igazítható.
A mai menü utolsó fogása Benjamin D. Jones kezétől született, és szerintem nincs is olyan, aki a program nevéből ne hallana ki bizonyos politikai mellékzöngéket.
Az a program neveugyanis, hogy Recount, ami annyit tesz:újraszámlálás. Nem is téved nagyot, aki a politikára asszociál a név hallatán, ugyanis a programot kétségtelenül a 2000. évi Egyesült államokbeli elnökválasztások ihlették.
A Recount rendkívül rugalmas szavazási lehetőséget nyújtó PHP alkalmazás, úgy a megjelenítés, mint az eredménykészítés szempontjából. Egyaránt használhat MySQL vagy PostgreSQL adatbáziskezelőt.
Ha ki szeretnénk próbálni ezt az ínyncséget, akkor először is töltsük le a legfrissebb Recount készletet a honlapjáról, ezt követően pedig tömörítsük ki a webkiszolgálónk könyvtárába (általában közvetelenül a gyökér alá).
Mint említettem, a Recount egyaránt támogatja a PostgreSQL és MySQL adatbázisokat, de egyenlőre maradok a MySQL ismertetésénél. Kezdetnek hozzunk létre egy ***Recount MySQL*** adatbázist ugyanazzal a módszerrel amit fentebb ***Sympoll*** esetében is használtunk: `mysqladmin -u root -p create recount`. Akár az előző példánkban, most is a korábban létrehozott polling MySQL felhasználót fogjuk használni, ami azt jelenti, hogy a polling felhasználónak meg kell adnunk a jogot az adatbázissal való munkához. Ezt úgy tehetjük meg, hogy csatlakozunk a MySQL-hez a `mysql -u root -p` utasítással, majd a ***mysql>*** promptból begépeljük a következőket:

```
mysql> grant all privileges on recount.*
    -> to polling@localhost identified by ´secr3t´;
```

A táblákat ezt követően a Recount forrás könyvtárából lefuttatott paranccsal hoztam létre:
`mysql -u polling -p recount < sql/mysql.recount.sql`
A Recount terjesztésben találunk egy ***config.recount.php-dist*** nevű állományt. Másoljuk át ezt ***config.recount.php*** névre majd nyissuk meg a kedvenc szerkesztőnkkel. Nem valószínű hogy sok problémánk lesz az állománnyal hiszen kiválóan dokumentált. Mindazonáltal a leginkább a következő sorok érdekelhetnek bennünket:

```
define(´RECOUNT_HOME_PATH´,
       ´/usr/local/apache/htdocs/recount´);
define(´RECOUNT_DB_TYPE´,
       ´mysql´);        /* “mysql” or “pgsql” */
define(´RECOUNT_DB_HOST´,
       ´localhost´);      /* db host */
define(´RECOUNT_DB_USER´, ´polling´);
define(´RECOUNT_DB_PASS´, ´secr3t´);
define(´RECOUNT_DB_NAME´, ´recount´);
```



6. ábra Mi az élet értelme? A Recount szavazóláda működés közben. Nem fogják kitalálni, hogy a kedves vendégek hogyan szavaztak erről az igen fontos kérdésről.

Egy kicsit lejjebb írjuk be a Recount rendszergazda felhasználó nevét és jelszavát. Ez nem a mi felhasználó nevünk és nem is a MySQL felhasználónév hanem az a felhasználó és jelszó páros, amit a szavazások karbantartására használunk majd:

```
define(´RECOUNT_ADMIN_USER´, ´admin_name´);
define(´RECOUNT_ADMIN_PASS´, ´admin_password´);
```

A beállítás fájl rendberakása után a jogosultságok megváltoztatásával ne feledjük el megfelelően védelmet biztosítani:

```
chmod 600 config.recount.php
```

Ennyi volt a teljes beállítás. Új szavazás létrehozásához állítsuk böngészőnket a ➲ http://kiszolgálónk/recount/recount_admin.php címre majd lépjünk be a korábban megadott rendszergazdai felhasználónévvel és jelszóval. Ne feledjük hogy ez a felhasználó nem azonos a MySQL felhasználóval amely a ***Recount*** adatbázisát frissítheti. Új szavazóláda létrehozásakor (*5. ábra*) megadhatjuk a szokásos, többféle válasszal ellátott kérdést, valamint a szavazás lejárati dátumát is. Amennyiben már vannak korábbi szavazóládáink, szerkeszthetjük vagy törölhetjük is azokat. Az adatok begépelése után az új szavazásunk kap egy azonosítót.
A friss szavazóládák kipróbálásához a csomagban találunk egy ***recount_testing.php*** nevű, használatra kész lapot. Ha böngészőnket erre a címre irányítjuk bármilyen szavazóláda ID-t megadhatunk. A szavazás és az eredmény egyaránt megjelenik alapon. Az adatott szavazóládához tartozó szavazólapot magunk is könnyen elkészíthetjük. vessünk egy pillantást a következő egyszerű HTML/PHP lapra:

```
<html>
<head><title>Make sure you vote!</title></head>
<body>
<p><img src="images/logo.png">
<h1>Ready to cast your vote?</h1>
<?php
      require(´config.recount.php´);
      $poll = new RecountPoll(2);
      print($poll->basicForm());
?>
</body></html>
```

A szavazóláda azonosítóját a fenti `$poll = new RecountPoll(2)` sorban találjuk. Ebben a példában , a szavazóláda azonosító ((a zárójelek közt) éppen 2. Ezt aztán tetszés szerint kidíszíthetjük, vagy saját honlapunkba illeszthetjük a PHP kódot. Az eredmények megtekintéséhez egy másik lapra lesz szükségünk, ahol pedig a következő PHP kódot használjuk az alapértelmezett HTML tegek között.

```
<?php
      require("config.recount.php");
      require("results.recount.php");
      ?>
```

Ennyi az egész. A 6. ábrán működés közben is megtekinthetjük a Recount szavazóládát. mint említettem nagyon rugalmas csomagról van szó. A hozzá adott dokumentációt is érdemes végigböngészni néhány további tippért.
Hiába, úgy tűnik a záróra gyorsan közeledik. Vessünk még gey utolsó pillantást a François fizetésemelését firtató szavazás állására. Mon Dieu, te jó ég, 17,972 szavazat érkezett, amely a fizetésed emelését javasolja, és mindösszesen 26 ellene? Nincs is ennyi ember az étteremben, mon ami. Természetesen azonnal újraszámlálást követelek.
Amíg azonban újraindítjuk a szavazást ebben a fizetésemelési ügyben, légy oly kedves és tölts még egy kicsit vendégeink poharába.
Teljesen biztos vagyok ugyanis benne, hogy az újratöltés kérdése olyan valami, amelynél szinte bizonyosra vehetjük az egyöntető és lelkes igen szavazatot. A következő alkalomig pedig, mes amis, kedves barátaim, igyunk egymás egészségére.

A vôtre santé!
Bon appétit!

Linux Journal 2003. november, 115. szám

Marcel Gagné (mggagne@salmar.com)
az Ontario állambeli Mississaugában él.
Ő a szerzője a Válts Linuxra! Búcsú a kékhaláltól című előkészületben lévő könyvnek. Első könyve, a nagy tetszést aratott Linux-rendszerfelügyelet (Kiskapu Kiadó).

Siemens Mobiltelefon Testreszabása Linuxon

Hosszú idő után végre a Siemens mobiltelefonokhoz is elérhető Linux alá olyan program, amelynek segítségével könnyedén és kényelmesen bütykölhetjük telefonunkat.

Sokáig semmi olyan program nem volt elérhető linuxra, amely képes lett volna segítséget nyújtani a telefonok testreszabásához. Aztán megjelent a gnokii, amely egy kis segédprogram Nokia típusú telefonokhoz. Egyéb márkák tulajdonosainak azonban továbbra is más rendszer alatt kellet elvégeznie bizonyos feladatokat. Nos, mintha lassan kezdene megtörni ez ügyben is a jég. Ha rákeresünk a interneten a „Siemens+Linux" kifejezésekre, akkor általában a Siemens cég linuxos kapcsolatával foglakozó honlapokkal találkozunk. Ha azonban elég kitartóan keresünk, ráakadhatunk a következő oldalra:
➲ http://mirror01.users.i.com.ua/~dmitry_z/siefs/
Ez a picike fejlesztői weblap kifejezetten a Siemens-telefonok Linux alatti használatával foglalkozik. A honlap szerint az alábbi típusok támogatottak, és teszteltek: Siemens S45, ME45, SL45, S55, M55, MC60.
A honlap megjegyzi, hogy a C55, az M50, az MT50 típusú készülékek a soros porton bírhatók szóra, valamint, hogy nem valószínű, hogy gond lesz az újabb típusokkal, mint például az SL55, C60, stb.
Először mindenképpen be kell szereznünk egy adatkábelt. Általában a telefonunkhoz kétféle kábel kapható, USB, és soros. Nos, nekem semmiféle tapasztalatom nincsen az USB kábellel, ráadásul a fent említett honlap is a sorost tanácsolja egyes típusoknál. Elvi akadályát ugyan nem látom az USB-nek, de mint említettem, semmiféle tapasztalatom nincs vele. Viszont a sorossal nem nagyon lehet melléfogni. Az eljárás másik akadálya lehet az automount. Itt elméletben lehetnek gondok, de erre esély főleg USB kábel használatánál van. Ha így állna a helyzet, akkor töröljük le az `/etc/sysconfig/automount` fájlt, és indítsuk újra a gépet. Ha szeretnénk visszakapcsolni, akkor hozzuk létre újra. Ne aggódjunk, ez egy 0 byte méretű fájl, csak létre kell hozni, és újraindítani a gépet (UHU-Linux 1.1.1 alatt). A teszt kedvéért visszakapcsoltam, de nálam nem okozott galibát soros port használatakor.
Ha ezzel megvagyunk, és a telefon össze van kötve a géppel, akkor semmi akadálya nincsen a szoftveres feltételek megteremtésének. Töltsük le a fent említett honlapról a ***siefs-0.2.tar.gz*** fájlt, illetve a lap alján linkelt ***fuse 1.1.tar.gz***-t. Csomagoljuk ki őket!

```
Tar -xvzf fuse-1.1.tar.gz
tar -xvzf siefs-0.2.tar.gz
```

Ha ez megvan, akkor lépjünk be a „fuse" mappájába, és sorrendben adjuk ki a következő parancsokat!

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

Ezek után lépjünk be a ***siefs*** könyvtárába, és adjuk ki ugyanezeket a parancsokat. Ezzel le vannak fordítva a fuse, és siefs meghajtók, és telepítettük is őket.
Most már nincs más dolgunk, mint becsatolni a telefont. Hozzunk ehez létre egy könyvtárat, ahová bekötjük.

```
mkdir /mnt/siemens
```

Utána a létrehozott könyvtárba csatoljuk a soros porta kötött telefont (a soros port (Com1) eszközleírója a ***/dev/ttyS0***, vagy ***(Com2) /dev/ttyS1***)

```
mount -t siefs /dev/ttyS0 /mnt/siemens
```

Ekkor – ha mindent jól csináltunk – az ***/mnt/siemens*** mappában megláthatjuk a telefon belső könyvtárait. A navigáláshoz nyugodtan használjuk a Midnight Commandert. Ha befejeztük a munkát, akkor ne feledjük leválasztani a telefont az `umount /mnt/siemens` paranccsal!

Hangok, képek feltöltése!

Ha megnézzük pl: az A60 „lemezét", akkor látunk benne egy „Pictures" és egy „Sounds" mappát is. Ezekben tárolja a telefon a képeket, és csengőhangokat.
Olvassuk el feltétlenül ,hogy milyen formátumú képeket szeret a készülék, ezeket ugyanis mindenképpen érdemes a megfelelő méretre kicsinyíteni. A készülék leírásában nézzük meg, hogy mekkora felbontású a képernyő, mert maximálisan olyan méretű lehet a kép is. Az biztos, hogy a 32 bites színmélységű képeket nem szereti egyik sem, bár előfordulhat, hogy elfogadja legfeljebb nem fogja tudni megjeleníteni ezt a minőséget, és a képben levő adattöbblet feleslegesen foglalja az amúgy elég kicsi memóriát. Ha készen van a kép, akkor - ugyanúgy mint a hangoknál, csak üssünk „F5"-öt.
Így tudjuk könnyedén teljesen testre szabni a telefonunkat Linux alól. További sok sikert, és jó munkát.

Dancsok Zoltán

Jump&Bump: Tasli Hapsi Linuxra

E hónapban is folytatjuk az apróbb, ám nem kevéssé szórakoztató játékok feltérképezését. A Jump&Bump egy rendkívül mulatságos játék, amely leginkább a klasszikusabb Commodore, és Atari időket idézi.

Nosztalgikus érzéseket kelthet az emberben a Jump&Bump, mely egy rendkívül szellemes, szórakoztató kis játék, amely a *jumpbump* paranccsal, vagy a menüből indítható. Először egy beállító felületet kapunk, ahol a későbbi játék főbb paramétereit lehet megszabni. Itt állítható be, hogy melyik pályát választjuk, (amíg nem telepítünk többet, addig mindössze egyetlen pálya áll a rendelkezésünkre), teljes

képernyőn fusson-e, letiltható továbbá a hang, vagy tükrözhető a pálya. Ha beállítottuk amit akartunk, kattintsunk a start gombra. Ekkor választhatjuk ki a játékosokat. Összesen négy nyuszi közül választhatunk, melyeknek irányítása a billentyűzet különböző részeiről történik. A kiválasztott játékosokkal a képernyőn látható fatörzs bal oldaláról ugorjunk át a jobb oldalára. A játék indítása úgy történik, hogy valamelyik nyuszi kiugrál a képernyőről a jobb oldalon.
A játékban a feladat pedig a következő: közel kell férkőzni a másik nyuszihoz, és leütni. Ezt automatikusan elvégzi a nyuszink, ha elég közel kerültünk a másikhoz, esetleg ráugrunk, el is neveztem a nyulamat „Tasli Hapsi" -nak.

Mint fentebb említettem, a játék pontokra megy, felső korlát nélkül. Tehát a játékosok maguk döntik el, hogy meddig szeretnének ugrándozni egy adott pályán, mielőtt kilépnének belőle. Természetesen a kilépéskor megtekinthető a játékosok listája, az elért pontjaikkal együtt.

A játékélmény

Konkrétan, és tömören összefoglalva: frenetikus. Nagyon jól mulatok, amikor játszom vele. Rendkívül ötletesnek, és aranyosnak találtam. A hangok rendkívül sokat dobnak a hangulaton, és nagyon mulatságossá teszik a játékot. Az irányítása nagyon egyszerű, percek alatt elsajátítható. A pályák sokfélesége pedig gondoskodik róla, hogy ne lehessen egykönnyen megunni.
Az élmény fokozásának igen jó módja, ha teljes képernyős módban játszunk. Igaz, a grafika ekkor nem az igazi, de ezzel korántsem azt mondom, hogy baj lenne vele. Sőt. Ablakban határozottan jól néz ki, de a hangulatot teljes képernyőn teszi teljessé.

Gépigény, és telepítés

A játék a ➲ http://jumpbump.mine.nu címen érhető el. Jó hír, hogy számos rendszerre megcsinálták többek között BeOS, Linux, BSD, Dos, Windows, MacOS X-re. Az említett címen találhatóak a pályák is, valamint az iménti rendszerek közül jó párhoz a csomag, természetesen a CVS is.
A telepítés egyik legkényelmesebb módja, ha elérhető csomagban úgy, hogy apt-get -el le tudjuk szedni. UHU-Linux tulajdonosok mindenképpen élhetnek ezzel lehetőséggel. Sőt, csomagban elérhető rengeteg pálya is, több mint ötven darab. Mérete tömörítve 41 MB, és szintén az apt-get-el telepíthető. A telepített játék gépigénye viszont „lemérhetetlen", legalábbis számomra. Teljes képernyős módban, és minden zenével együtt sem volt hajlandó döccenni egyet sem egy 650 P3 -as laptopon, se celeron 466-on, se P2 233 -on.

Dancsok Zoltán