

Hogy hová jutottunk...!

Testvérlapunk, a Linux Journal most ünnepli a századik számát. Mi ugyan csak a huszadik lapzártaát izguljuk végig, és nem tekinthetünk vissza nyolcéves múltra, mégis úgy érzem, osztozhatunk az ünnepségen. Akárhogy is nézzük, nagyon sokat változott a számítástechnika ez alatt a nyolc év alatt. Hogy mennyire így van ez, akkor tűnik szembe, amikor „rövid összefoglalót” szeretnénk készíteni erről az időszakról. Erre vállalkoztak az LJ munkatársai is, és akárhogy igyekeztek is megragadni a száz legfontosabb pillanatot (a hétoldalmi összefoglalót lásd a 20. oldalon), érezhető, hogy bizony sok mindent ki kellett hagyniuk.

Visszatekinteni?

De milyen tanulságot hordoz ez a mi számunkra? Nézzük csak... nyolc év alatt jutottak el az első „üzemeltethető” rendszer felbukkanásától (ami messze nem volt komoly munkavégzésre alkalmasnak nevezhető) a jelenlegi helyzetig. Tehát addig a pontig, hogy ma már mind nyugatról, mind keletről rendkívül sok hírt hallani, hogy az X állami szervezet vagy az Y óriáscég a GNU/Linuxra való átállás hatásait elemzi, vagy éppen nyilatkozatban fejtik ki lesújtó véleményüket a Microsoft üzletpolitikájáról. Amerika és Nyugat-Európa több területe is eljutott oda, hogy egy rendszergazda számára a Linux ismerete alapkövetelmény, nem pedig elnézendő hóbort. Nyolc év kellett „odakint” ahhoz, hogy megismerjék és megszeressék ezt a rendszert.

És idehaza?

A linuxos világ egy csöppet késve indult, és valljuk be, amolyan „magyar módra”. Az oktatásban értetlenül néztek a jövőre, a vállalatoknál tüzzel-vassal irtották a Linux-pártiakat, a továbbképzésre fordított összegekből sem jutott erre a területre (már ahol volt ilyen keret), sőt gyakorta olyan hangokat is hallani lehetett, hogy komoly szakmai döntések alkalmával határoztak a „szabad rendszerek” ellen, nyilvánvalóan a szakmai hozzá nem értésnek és a zárt rendszerek ügyes marketingpolitikájának következtében. Ha viszont a másik oldalról nézzük, a hazai szakembereknek két-három év

alatt kellett behozniuk a nyugat (legalább) négyéves előnyét, csekély külső támogatás mellett, vagy éppenséggel az árral szemben úszva. Ez pedig látványos sikerrel járt. Ma már szerencsére nincs olyan épeszű közép- vagy nagyvállalat, ahol ne használnának linuxos gépeket, sőt egyre több szervezet is büszkén hirdeti, hogy náluk a szabad rendszerek viszik el a pálmát. Azért, hogy ez megvalósulhatott, nagyon nagy köszönet illeti az egyetemi és főiskolai oktatókat, akik nem elégszenek meg a „szokásos” tananyaggal, hanem folyamatosan újítják, bővítik azt. Mert a Linux remek segédeszköz a számítástechnika oktatásához! És ma már szerencsére sokkal több annál.

Sikító Linux

Pár hónapja jutott el a rendszer oda, hogy bátran kijelenthetjük, egy hozzáértő rendszergazda fel tud készíteni egy hálózatot az irodai munkára, de akár a titkárnők számára is. Itt most nem a titkárnőket szeretném leszólítani, félreértés ne essék, csupán két dologra kívánok rámutatni: az egyik, hogy bizony sokat kellett várniuk (és dolgozniuk) ahhoz, hogy a megfelelő eszközök és programok elkészüljenek, a másik pedig, hogy a GNU/Linuxot nem arra találták ki, hogy egy titkárnő vagy egy képzetlen rendszergazda telepítse, üzemeltesse. Amennyire tudom, a Linuxot ért legtöbb lesújtó vélemény mögött egy félig (sem) képzett, a rendszert nem ismerő vagy éppen lusta rendszergazda áll. És a másik nagyon fontos terület, a magyar szakirodalom hiányossága is megoldódni látszik. Három évvel ezelőtt igen furán néztek rám a kiadóban, amikor kijelentettem, hogy a Linux hálózatok című könyvet ki szeretném adni. Szerencsére a könyvnek remek fogadtatása volt, és azóta se száma a Linuxszal és a rendszerhez kapcsolódó témákkal foglalkozó könyveknek. Ma már olyan témákról is komoly kiadványokat lehet találni, mint például a héjprogramozás, sőt az LME által készített Pingvin Füzeteknek is egyre több része készül el (lásd a lemez-mellékletet!). Röviden összefoglalva tehát úgy látom, hogy a hazai szakemberek a nehézsé-

gekre fittyet hányva elérték a céljukat, és egy-két év múlva majd minden vállalatnál boldogan bölintanak, ha a munkatárs Linux alatt kíván dolgozni. De addig is még sok dolgunk van! Itt van például nagy vesszőparipám, a magyarítás, a honosítás.

Szépen magyarul...

Bizonyára mindenki tudja, aki pár lapszámunkat átfutotta, hogy nálunk is komoly viták alakulnak ki egy-egy szó használatával kapcsolatban. Szerencsére nagyon sok kifejezés értelmezése és használata már letisztult, de mindig vannak „elsietett” fordítások, utólag észrevett ütözések. Történt például az előző számunkban, hogy a nagy sietésben a szerkesztő a „konferencia” szót mindenhol „tanácskozás”-ra cserélte le, ennek szomorú mellékhatásaként a több éve rendezett Linux Konferencia viccesen Linux-tanácskozás néven került bele az újságba. Ezért ezúton is elnézést kérünk a Linux Konferencia szervezőitől. Emellett nagy előrelépés, hogy végre üzembe helyeztük weben is elérhető szótárunkat: nem túl bonyolult programról van szó, remélhetőleg ennek ellenére könnyen és gyorsan használható, és sokaknak segítséget tud majd nyújtani a fordítások, magyarítások világában! Természetesen ez a szótár is egy folyamatosan bővülő és változó adatbázis, ezért arra kérek mindenkit, ha hibát talál benne, vagy hiányol belőle valamilyen kifejezést, írja meg nekem! Sőt, azt is örömmel fogadom, ha a szótárban lévő szavaknál ötletesebb, találhatóbb fordítást kapok. Előre is köszönöm és sok sikert kívánok!



Szy György
a Linuxvilág főszerkesztője,
a Kiskapu Kiadó vezetője.
Mindenki véleményét és
levelét örömmel várja az
alábbi levélcímen:

Szy.Gyorgy@linuxvilag.hu

Kiskapu-szótár

➔ szotar.kiskapu.hu

Programvadászat



Gnome 2

Hosszas fejlesztés és hibajavítás után megjelent végre a Gnome 2. Az X Window Systemre épülő Gnome-környezet a GNU/Linux-felületen kívül Solaris, HP-UX, BSD és Darwin (MacOS X) rendszereken is futtatható. Az új Gnome a grafikus felhasználói felületek készítéséhez használt GTK+ lib 2-es változatát használja, amely a szokásos GLib-en kívül újabb rendszerkönyvtárakat is igényel. A szövegkezelés és betűtípus-leképezés (font rendering) feladata a GTK+ 2-es változattól kezdve a *Pango* nevű könyvtárra hárul, míg az ATK a kisegítő lehetőségek (például a választható be- és kimeneti eszközök használatának) GTK-n keresztüli könyvtárszintű megvalósításáért felel.



Korongunkra felkerült a Gnome 2 forrása és Red Hat rendszerhez tartozó rpm-csomagok.

Pingvin Füzetek

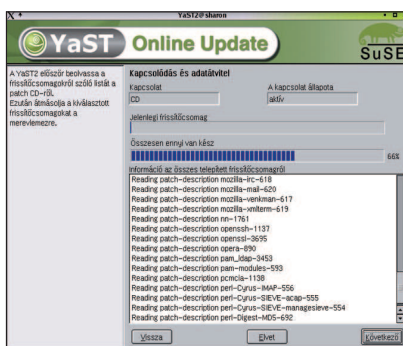
Korongunkon helyet kaptak az LME által készített eddigi Pingvin-füzetek darabjai. Ezek a következők:

- Linux-alapparancsok
- Linux-héjprogramozás
- Linux-segédprogramok
- Linux-hálózat
- Linux-karbantartás

A füzetek segítségével megbarátkozhatunk a Linux látásmódjával, alapvető parancsaival, a hálózati beállításokkal, a héjak programozásával és a rendszer karbantartásával. Ezek a számok mind-mind különálló nagy témaköröket érintenek. A kezdőket bevezetik a Unix rejtelmébe, megismertetik őket ezzel a különleges világgal; a haladók pedig kedvükre turkálhatnak rendszerük legmélyebb bugyaiban is, így veszélyeztetve azt; ám mindenképpen sokat tanulva és fejlődve fognak felállni a gép előtt. Egy jó tanács: a rendszergazdai teendőket először ne „éles” gépen gyakoroljuk!

SuSE 8.0-rendszerfrissítés

Az előző számunkból kimaradt SuSE Linux 8.0-frissítéseknek mostani koron-



gunkon adunk helyet, ebben a rövidke írásban használatának rejtelmébe szeretném olvasóinkat bevezetni. Helyezzük be a CD-t a meghajtónkba, és indítsuk el a YaST2 programot. A rendszergazdai jelszó megadása után kattintsunk a *Frissítés Patch CD-ről* felírára vagy ikonra. Ezután a program beolvassa a CD-n lévő adatokat, és megkeresi a telepített programok új vagy javított változatát. Ezt követően egy listát kapunk, amelyből kiválaszthatjuk és módosíthatjuk a YaST2 által felajánlott csomagok frissítését. A frissítés különlegessége az első helyen szereplő KDE_3.0.2-1 csomag, amelynek 0,0 MB a mérete – ettől nem kell megijedni, csak kétszer két kattintás választ el bennünket a telepítéstől, kattintgassunk hát bátran!

A csomagok melletti jelek magyarázata:

- X: az Ön által kiválasztott és a telepítéshez kijelölt frissítőcsomag.
- A YaST2 ezeket a frissítőcsomagokat letölti és telepíti. Ha egy bizonyos frissítőcsomagot mégsem akar tele-

píteni, egy dupla kattintással leveheti a listáról (vagy válassza ki a megfelelő sort, és használja az ALT-A-t).

- G: elérhető, de nem feltétlenül szükséges frissítőcsomag (frissítés egy olyan csomaghoz, ami az Ön rendszerén nincs telepítve). A frissítőcsomagot letöltheti, de a YaST2 nem fogja önműködően telepíteni.
- _: nem kiválasztott frissítőcsomag (a YaST2 se nem tölti le, se nem telepíti). Kiválasztásához kétszer kattintson a kívánt csomagra (vagy ALT-A).

Miután mindezzel végeztünk, kezdjük neki a frissítésnek! Kattintsunk a *Következő* gombra, ekkor a rendszer csak a kijelölt csomagok letöltését végzi, majd ismételtén a *Következő* gombra kattintva a csomagok ellenőrzése és telepítése történik meg. Ez a rendelkezésre álló memória nagyságától és a processzor erejétől függően bizonyon eléggé elhúzódhat. A frissítés elvégzése után jelentést kapunk a telepített csomagok, illetve a biztonsági frissítések számáról, erről bővebb tájékoztatást is kérhetünk. Legvégül kattintsunk a *Kész* gombra, és élvezzük munkánk gyümölcsét! A telepítéssel, illetve a frissítéssel kapcsolatos kérdésekre a <http://sdb.suslinux.hu> adatbázisban kaphatnak választ.

A SuSE 8.0 hivatalos frissítő-CD-t a SuSE Linux magyarországi irodája állította össze a Linuxvilág olvasói számára.

Rendszermag

Féléves fejlesztés után megjelent a 2.4.19-es megbízható rendszermag, mostani korongunkon ezt és a 2.5.30-as fejlesztői magot is közreadjuk.

Játék

Az előző korongra sajnos hibásan került fel a Tux Racer játék, ezért most ismét szerepel rajta. Sokan értékék, hogy tegyük újra közzé, mivel nagyon szeretnék kipróbálni ezt a remek játékot.



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

Új ATi Radeon kártyák jelentek meg

Az ATi Radeon sorozata új tagokkal bővült, illetve bővül a közeljövőben. Ugyan az új kártyák Radeon 9000, Radeon 9000 Pro és 9700 jelzést viselő



lapkával kerülnek forgalomba, ez a Radeon már messze nem az a Radeon, amit jó ideje ismerhetünk. A legnagyobb sorszámot kapott lapkához például 128 MB memória társul, a DVI-felülettel rendelkező megjelenítőket és 1024×768-as felbontásig tévékészülékeket is meg tudja hajtani; támogatja a 8× AGP és a 9.0-s DirectX szabványt, nyolc párhuzamosan működő leképező csővezetékekkel és négy független geometriai motorral rendelkezik, hogy csak a legfontosabbakat említsük. A Radeon 9000 kártyák a tengerentúlon már kaphatók, a 9700-as példányokra még várni kell egy hónapot.

➔ <http://www.atitech.com>

Ellenségből jóbarát?

A nyílt programok világának hívei és a Microsoft képviselői eddig nem sok jót mondtak egymásról, most mégis egyfajta közreműködés alakul ki az előbbihez tartozó Apache Software Foundation és a mamutcég között. Az érintettek közvetlenül ugyan nem dolgoznak együtt, ám a Covalent Technologies által fejlesztett Enterprise Ready Server termékben megvalósul az Apache 2.0 és a Microsoft ASP .NET fejlesztői környezetének együttélése.

Ezzel a lépéssel a Microsoft részben kiránthatja a talajt a saját IIS webkiszolgálója alól, viszont megnyerheti a fejlesztőket az ASP .NET eszközök számára. A Microsoft egy nap eltéréssel azt is bejelentette, hogy a .NET-alkalmazások az Oracle adatbázisokkal is magas szinten együtt fognak működni, így a különféle rendszerek a korábbinál sokkal jobban összebékíthetők lesznek. Azt, hogy a Microsoft valóban nyitásra szánta el magát, vagy csak rendkívül ügyes üzletpolitikát folytat, még meglátjuk.

➔ <http://apachetoday.com>

Debian GNU/Linux 3.0

A Debian Project bejelentette a 3.0-s Debian GNU/Linux-terjesztést. A korábbi változat még „csak” hatféle, a legújabb terjesztés azonban a MIPS, Intel Itanium, HP PA-RISC és S/390 géptípusok hozzáadásával immár tizenegy géptípus támogatására alkalmas, így az új rendszer a kézigépektől kezdve egészen a szuperszámítógépekig szinte bármin futtatható. Az új terjesztésnél is



elsődleges a biztonság és az adatvédelem, de többek között a KDE 2.2-es és a Gnome 1.4-es változata sem maradt ki. A fejlesztők javították a telepítőt – a terjesztés CD-ről, hálózatról és hajlékonylemezről egyaránt feltehető, illetve a korábbi terjesztések frissítése is kényelmesen elvégezhető.

Ez az első olyan Debian-változat, amely megfelel a Filesystem Hierarchy Standard (FHS) előírásainak, és – bár a megfelelő tanúsítvány beszerzése még várta magát – a Linux Standard Base (LSB) szerint fejlesztett alkalmazásokat is támogatja.

➔ <http://www.debian.org>

Oroszország nem kér a Nyugat programjaiból?

Vannak, akik máris a hidegháború folytatását emlegetik, szerintem azonban teljesen természetes önállóság-megőrzési törekvésről van szó. A nagyhatalmak ugyanis általában nem szeretnek más hatalmakra hagyatkozni, ami nemcsak az atombombák egymásra halmozásában érhető tetten, de a számítógépek területén is érvényes. Kína példáját követve most orosz részről is elkottyantották, hogy az államigazgatásban minél több nyílt – és természetesen ingyenes – operációs rendszert és alkalmazást szeretnének látni. Maga *Vlagyimir Putyin* is azt kívánja, hogy csökkenjen az országnak a nyugati cégek által készített programoktól való függése, különösen a stratégiai, nemzetbiztonsági szempontból fontos területeken. Az sem utolsó szempont, hogy újabban Oroszországban is kezdenek ódfigyelni a jogtisztaságra, ám a költségek vásárlás előtti összesítése sokszor mellbevágó eredménnyel jár.

Az orosz Védelmi Minisztériumnál már most is használnak egy Linux-alapú rendszert, amelynek biztonsága a NetWare 5.1 és Windows NT-rendszerekénél is magasabb szintű – hasonló fejlesztés egyébként az amerikai oldalon is működik.



Megjelent az Ogg Vorbis 1.0

Többéves munka eredményeképpen megjelent a szabadon hozzáférhető hang- és folyamkódoló kodek 1.0-s változata, a fejlesztéséhez az alapítványi háttérrel a Xiph Foundation bizto-



sítja. Az On2 Technologies máris bejelentette, hogy videokodekjét hamarosan Ogg Vorbis hangkódolással is hozzáférhetővé teszi. Az Ogg Vorbis hívói azt várják, hogy az új kodek átveszi az MP3 formátum vezető helyét – annak hatalmas felhasználói bázisa ellenére is. Míg az előbbi még cégek számára is ingyenes, illetve bármilyen termékbe szabadon beépíthető, addig az utóbbi a fejlesztő Fraunhofer Institute és az MPEG Consortium ellenőrzése alatt áll, amelyek minden kiadott kódolóprogram után szerzői díj beszedésére jogosultak.

➔ <http://www.xiph.org>

Liberty kontra Passport

Alig egy éve alakult meg a Liberty Alliance Project, és máris közzétette a nyílt Liberty-előírások 1.0-s változatát. A fejlesztés célja az, hogy a felhasználó egyszeri bejelentkezéssel és egyszerűsített kilépéssel szörfölhessenek egyik internetes oldaltól a másikig. Használataval lehetővé válik, hogy a különböző oldalak és kiszolgálók egyszerűbben, gyorsabban azonosítsák a látogatókat, és bonyolítsák le a tranzakciókat. A felhasználók saját választásuknak megfelelően több szolgáltatóból álló bizalmi köröket hozhatnak létre, és miután bejelentkeztek az egyikhez, újabb belépés nélkül használhatják a többi oldal



szolgáltatásait. Az első működő rendszerek felbukkanása már év végére várható, és az újabb változaton is megkezdődött a munka.

Az sem utolsó szempont, hogy így fel lehet venni a harcot a Microsoft által fejlesztett Passport-megoldással. A Liberty-tervezetben a teljesség igénye nélkül olyan cégek vesznek részt, mint a Novell, az RSA Security, az American

Express, az AOL, a HP, a Nokia, a Sony, a Vodafone vagy a Sun Microsystem; ez utóbbi a Java – C#, .NET-ügyek miatt eddig is hagyományosan „jóban” volt a Microsofttal. A Liberty részéről máris hangsúlyozzák, hogy a Liberty és a Passport közötti átjárás megoldható, és valószínűleg megfelelő programok is lesznek. A szövetség a Microsoftot is felkérte a csatlakozásra, és a cég nem is zárta ki ennek lehetőségét, bár szoros együttműködésre éppen az eddigi pengéváltások miatt valószínűleg nem érdemes számítani. Ennek fényében viszont nem egyértelmű, hogy az adatvédelmi szempontok miatt sokat támadott, a felhasználók – és sokszor a vásárló-felhasználók – szokásainak megfigyelését lehetővé tévő Passport miért volna rosszabb a Liberty megoldásánál.

➔ <http://www.projectliberty.org>

Machinima Fesztivál 2002

Filmfesztivált rendeztek az Academy Of Machinima Arts & Sciences és az nVidia Corporation szervezésében és támogatásával augusztus 17-én a texasi Mesquite városban. Az egynapos rendezvényen filmek bemutatásán kívül találkozót és kisebb kiállítást is rendeztek.



Talán létezik még olyan ember, akinek a Machinima kifejezés ismeretlen. Ez alatt az olyan digitális mozgófilmeket értik, amelyeket virtuális világban, ám a hagyományos filmekhez hasonlóan forgatnak. A művelet egyszerre hasonlít a számítógépes animációk készítéséhez, a játékok fejlesztéséhez és a filmforgatáshoz. Rövid történetváz, forgatókönyv készül hozzá, külön veszik fel a hanghatásokat és a szereplők hangját, majd rendezővel, operatőr-rendezővel felveszik a filmet, befejezésül véglegesre vágják.

A rendezvényen a díjakat 10 kategóriában osztották ki, például a legjobb hang, a legjobb vágás vagy a legjobb látvány kategóriákban. A jelentkezési határidő augusztus 9. volt, lapzártakor még nem tudni, vajon akad-e magyar alkotó a jelentkezők között, és ha igen, milyen eredménnyel szerepel.

➔ <http://www.machinima.org>

Kipróbálható a Mandrake 9.0 próbaváltozata

A nagy nyári melegben sem szünetel a munka a Mandrake háza táján. Újdonságok már nem kerülnek a terjesztésbe, jelenleg a hibák javítása folyik, amiben bárki segítséget nyújthat, ha letölti a terjesztés első próbaváltozatát, és jelenti a megtalált rendellenességeket. A várhatóan októberben megjelenő Mandrake Linux 9.0 felvonta a közelmúlt újdonságait: többek között a KDE 3.0.2, a Mozilla 1.0.0, a Gnome 2.0, az Evolution 1.08 és a GCC 3.1.1 kerül bele a DVD-n is elérhető terjesztésbe.

Nem kevésbé fontos, hogy a Mandrake pénzügyi helyzete is megszilárdulni látszik. A cég bevétele jelentősen növekedett, ezzel párhuzamosan csökkent a veszteség, a 2001/2002-es pénzügyi év harmadik negyedéve az eddigi legjobb volt a cég történetében.

➔ <http://www.mandrakesoft.com>

Az örök OpenGL

Az SGI tíz évvel ezelőtt ismerte fel, hogy a világnak egy általános 3D-s programozási felületre lesz szüksége – ekkor indult el az OpenGL 1.0 fejlesztése. Azóta a 3D-gyorsítók piaca többmilliárd dolláros üzlettel fejlődött,



ám az OpenGL fejlesztése továbbra sem állt le. Nemrég bejelentették az 1.4-es szabványt, ugyanakkor megalakult az a munkacsoport, amelynek a 2.0-s változat kidolgozása lesz a feladata. Az új szabvány megjelenése valamikor a jövő évben várható. Fejlesztését sajnálatos módon hátráltathatja, hogy a Microsoft megkaparintott néhány 3D-s képmegjelenítéssel kapcsolatos szabadalmat, és ezek után szabadalmi díjra jogosult – ez kellemtlenül érintheti a Linux- és Macintosh-felhasználókat.

Első hallásra kicsit elvetemült ötletnek tűnik, hogy a Nokia és az SGI egy mobiltelefonokra szánt, OpenGL-alapú programozási-megjelenítési felület fejlesztésére szövetkezett. A munkában a Khronos Group



is részt vesz, amely játékgépekhez, terminálokhoz és repülésirányító műszerekhez készülő különféle grafikai és médiaszabványok fejlesztésével foglalkozik. A mobilgépeken, telefonokon is használható 3D-s megjelenítési lehetőség révén teljesen új, például térinformatikai alkalmazások vagy játékok jelenhetnek majd meg.

➔ <http://www.opengl.org>



© Kiskapu Kft. Minden jog fenntartva

A D-Link bejelentette az első kétsávos vezeték nélküli útválasztót

A D-Link az első olyan cég, amely kétsávos, 2,4 és 5 GHz-es frekvenciákon egyaránt működőképes vezeték nélküli útválasztót hoz forgalomba. A készülék egyszerre szolgál IEEE 802.11a szabvány szerint működő, 54 Mb/s sebességű; IEEE 802.11b szabvány szerinti, 11 Mb/s sebességű; és IEEE 802.11g szerinti 22 Mb/s



sebességű vezeték nélküli útválasztóként, valamint négykapus, 10/100 Mb/s sebességű ethernetkapcsolóként. Segítségével könnyedén építhetők a jövőbeli fejlődést is követni, kiszolgálni képes hálózatok, illetve a meglévő eszközök mellé gond nélkül vásárolhatók újabb 5 GHz-es hálózati elemek.

A D-Link AirPro DI-764 webalapú grafikus felhasználói felülettel bír, amelyen minden beállítás sűgőval támogatottan adható meg, a későbbi visszaállításokra gondolva a számítógép merevlemezére is menthető. Az új rádiós eszköz képes tartományok, IP- és MAC-címek, valamint URL-ek – a bennük szereplő szavak alapján történő – kizárására, támogatja a virtuális magánhálózatokat, a hálózati címfordítást, beépített tűzfallal rendelkezik, valamint 256 bites WEP-titkosítással védi a hálózat sértetlenségét.
 ↪ <http://www.dlink.com>

eZ80 Webserver-i

A ZiLOG beharangozta eZ80 proceszorcsaládjának legújabb tagját, az eZ80 Webserver-i lapkát. Az új termékbe minden olyat beépítettek, amely egy kisebb webkiszolgáló üzemeltetéséhez szükséges lehet: az 50 MHz-es proceszorhoz 16 MB memória, teljes TCP/IP-

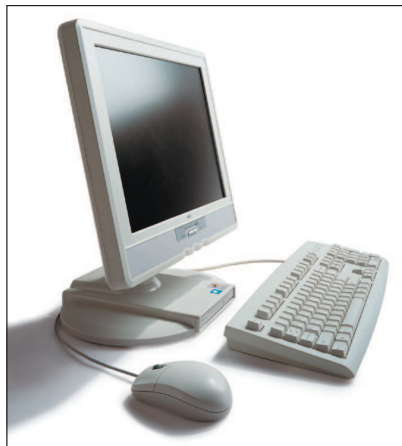


protokollkészlet, infravörös és ethernet-csatoló jár. Nem is akármilyen áron, 1000 darabot vásárolva a lapka ára nagyjából 1500 forint darabonként, de egy kicsivel magasabb díjcsomag ellenében akár egyetlen példány is megrendelhető Interneten keresztül. Az új lapka – a jó öreg Z80-as alapjaira

építkezve – elsősorban olyan készülékekben kaphat helyet, amelyeket fejlesztőik weben keresztül felügyelhetővé szeretnének tenni – ilyenek például az ipari mérőműszerek, a vezérlések, a biztonsági és az épületfelügyeleti berendezések.
 ↪ <http://www.zilog.com>

Zöld NEC gép Transmeta szívvel

Az NEC takarékos PowerMate eco számítógépébe a Transmeta TM5800 jelzésű processzora kerül, jelentette be a lapkagyártó cég. A Transmeta lapkái



energiatakarékosságuk miatt eddig elsősorban a hordozható gépekben jutottak szerephez, így a bejelentés fontos lehet a folyamatosan értékesítési – és ezzel párhuzamosan fennmaradási – gondokkal küzdő cég számára. Ugyan Japánban egy ideje már kapható PowerMate asztali gép, nem áll messze a hordozható eszközöktől, hiszen belső szerveit 15 hüvelykes TFT-kijelzőjének talpába építették be, külön házzal nem rendelkezik. A 900 MHz-es Transmeta proceszor csak kismértékben melegszik, így hűtőventilátorra sincs szüksége, ezért a gép nemcsak kevés áramot fogyaszt, de csendes is. Ki gondolná például, hogy a világ fejlettebb részein a kórházi gépek ventilátorai által felkavart porra és a vele együtt utazó kórokozókra is ügyelnek?

A PowerMate TFT-kijelzője a hagyományos képernyőkkel ellentétben nem tartalmaz ólmot, így elhasználódás után sem szennyezi vele a környezetet. Hasonlóan ólommentesek az alaplap forrasztásai is. A készülék háza az NEC szabadalmazott, NuCycle nevű, teljes mértékben újrahasznosítható műanyagjából készül, amely különleges adalékolásának köszönhetően adott esetben a tüzek terjedését is környezetszennyezés nélkül gátolja.
 ↪ <http://www.necomp.com>

Tessék, itt Miki egér beszél!

A Motorola és a Disney közösen próbálják elbűvölni a 6–12 éves korosztályt új rádiókkal és vezeték nélküli telefonokkal. A készülékek megjelenését és csengőhangját népszerű Disney-karakterek, mint Miki egér és Buzz Lightyear (Toy Story) ihlették.



A 2,4 GHz-es frekvencián üzemelő készülékek az Egyesült Államokban 60 dolláros áron már a nyár végén kaphatók lesznek, a világ egyéb részein viszont csak jövőre jelennek meg.
 ↪ <http://www.motorola.com>

Levelezni csak szépen!

Két alkalmazottjának bűcsút intett, további 150 dolgozót pedig felfüggesztett a HP Angliában és Írországban. Az alkalmazottak bűne az volt, hogy a cég erőforrásait nem a belső szabályzatnak megfelelő levelezésre használták. A nagyobb cégek egyre komolyabban ügyelnek arra, hogy munkatársaik valóban munkavégzésre, és ne viccek, szexképek, rosszabb esetben törvénytörtő anyagok terjesztésére használják hálózatukat – a HP lépése is ennek a törekvésnek a jele.

Startra kész a PCI-X 2.0

Több mint húsz gyártó várta izgatottan a PCI Special Interest Group által fejlesztett PCI-X 2.0 szabvány végleges változatának a megjelenését. Az elsősorban kiszolgálókban alkalmazott megoldás a jól ismert PCI-busz kiterjesztése, ám az asztali gépeknél jóval nagyobb adatátviteli sebességet tesz lehetővé. A PCI-X 2.0 kétféle változatban létezik, amelyek 266 és 533 MHz-es órajellel üzemelnek. Az előbbi 2,1, az utóbbi pedig 4,3 GB/s sebességű átvitelt tesz lehetővé, miközben az adatok épségét ECC-kóddal védi.
 ↪ <http://www.pcisig.com>



Medgyesi Zoltán

(mzx@axelero.hu) a BMGE 24 éves informatika szakos hallgatója. Szabadidejét legszívesebben a barát-nőjével tölti. Szeret autózni és bográcsban főzni. A Linuxot hat éve ismeri, de még nem volt lelkiereje, hogy áttérjen rá. A Linuxvilág magazin hírszerkesztője.

Cégvilág

Oracle-tervek:**a Red Hat és a Dell adja a hátteret**

Az adatbázis-rendszeréről világszerte híres cég (legutóbb júniusban döntöttek sebességrekordot az Oracle 9i 2. változatával) is folyamatosan figyeli a piacot. Bár valószínűleg sosem jut el a GNU-rendszerek árszintjére, de a Microsoft árait az Oracle is sokallja. Valószínűleg a nagyvállalatok között lesz nagy visszhangja a július 10-én bejelentett új csoportmunka-támogató rendszerének, melyet egyértelműen a Microsoft hasonló termékének versenytársaként helyez el a piacon.

A legújabb hír pedig, hogy az Oracle, a Dell és a Red Hat összefogott, hogy kifejlesszenek egy „mindent kibíró” linuxos megoldást. A megoldáshoz a vasat a Dell PowerEdge kiszolgálói jelentik, Dell/EMC és PowerVault tárolórendszerekkel, a programok oldaláról az Oracle9i Database Release 2 tartja a frontot, Real Application támogatással, a Red Hat Linux pedig az Advanced Server változatát igyekszik megfelelően hozzáhangolni az igényekhez. Valószínűleg a linuxos fejlesztők körében tapasztalható rendkívüli érdeklődésnek köszönhető e lépés is (a fejlesztők több, mint félmillió linuxos terméket tölthettek le másfél év alatt az Oracle honlapjáról). A cég egyébként komoly terveket dédelget a „feltörhetetlen és eltörhetetlen” linuxos telepekekkel kapcsolatban, a fejlesztők a honlapról teljes fejlesztői tájékoztató csomagot tölthetnek le.

**Ingyenes GPRS-hozzáférés
3 hónapig a Vodafone hálózatán**

A Vodafone 2002. július 15-én indította el GPRS-szolgáltatását teljes magyarországi hálózatán, amely három hónapig – azaz október 15-ig – ingyenes (bejegyzési, havi- és forgalmi díj nélküli), korlátlan hozzáférést tesz lehetővé. A cég átalánydíjas internetszolgáltatást is kínál előfizetőinek, ennek ára bruttó 3125 forint havonta, így akik megfelelő készülékkel rendelkeznek, a Vodafone hálózatán kedvező áron internetezhetnek. A GPRS a hálózat terheltségétől függően akár 20–40 kbit/s adatátviteli sebességre is képes, így jól használható hálózati hozzáférést jelent olyan területeken, ahol semmilyen széles sávú hozzáférési lehetőség nem áll rendelkezésre – esetleg jobb minőségű telefonvonal sem. Akinek a telefonja nem rendelkezik GPRS-képességekkel, az a cég egyik akciós csomagjában bruttó 19 900 forintért beszerezheti a Sony-Ericsson R600-as, GPRS-re alkalmas mobiltelefonját.

➔ <http://www.vodafone.hu>

NASA

A NASA mostantól a világ egyik legnagyobb szuperszámítógépe segítségével vizsgálja a Föld éghajlatát és jelzi előre a vulkánkitöréseket, a globális felmelegedést és más hasonló természeti jelenségek időjárására gyakorolt hatását. A Maryland állambeli Greenbelt város Goddard űrepülési központjában működő NASA-központ HP Alpha-

Server SC45 szuperszámítógépét a NASA mérnökei a kaliforniai El Segundóból irányított Computer Sciences Corporation munkatársaival közösen helyezték üzembe és próbálták ki. Az új gép birtokában a NASA kutatói nagyobb pontossággal tudják előrejelezni a különféle éghajlatváltozások következményeit. A CSC munkatársai az NCCS telephelyén egy 512 processzoros HP SC45 szuperszámítógépet helyeztek üzembe. Az új rendszer – amely a világ húsz leggyorsabb szuperszámítógépének egyike – több mint kétszeresére növeli a központ jelenlegi teljesítőképességét.

➔ <http://www.csc.com>

➔ <http://www.hp.hu>

**Leépítés az IBM
székesfehérvári üzemében**

Az IBM 2002. július 29-én jelezte, hogy a merevlemezek gyártásáról ismert székesfehérvári üzemében munkaerő-csökkentést fog végrehajtani. A leépítést a termékek iránti kereslet visszaesésével indokolták. Az elbocsátandók egy részének egyébként is lejár a munkaszerződése, másoktól viszont közös megegyezés alapján vesz búcsút a cég. Az IBM merevlemez-üzletágát nemrég a Hitachi vette meg, azt azonban tagadták, hogy ennek bármi köze lenne a leépítéshez.

➔ <http://www.ibm.hu>

Rádiós EnterNet-szolgáltatás

Az EnterNet vezeték nélküli szolgáltatást indít, amely telefondíj nélkül kínál internetkapcsolatot az ország számos pontján. A technológia pont-multipont rendszerű, és a bérelt vonalhoz hasonló minőséget biztosít az előfizetők számára. Az új szolgáltatás használatba vételi ideje előreláthatóan két hét lesz.

A vezeték nélküli internetelés augusztus 10-től Baja, Balatonboglár, Balatonfüred, Kalocsa, Kaposvár, Keszthely, Pápa, Salgótarján és Szentes városokban érhető el, és rövid időn belül további 45 város bekötését tervezik. Jelenleg az EnterNet által kiépített háttér 4 km-es sugárú körben biztosít internet-hozzáférést, igény esetén azonban a szolgáltatási terület tovább növelhető. A hálózat a vezeték nélküli hálózatokra kidolgozott, nemzetközileg legelterjedtebb 802.11b szabványnak megfelelően a 2,4 GHz-es szabad frekvencián működik, és jelenleg 11 Mb/s adatátviteli sebességet biztosít.

A rendszer a hamarosan piacra kerülő bővítőmodul segítségével az 5 GHz-es frekvenciasávban akár 54 Mb/s sebességgel is képes lesz működni.

➔ <http://www.enternet.hu>

Medgyesi Zoltán

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

Linux a tanteremben – 2. rész

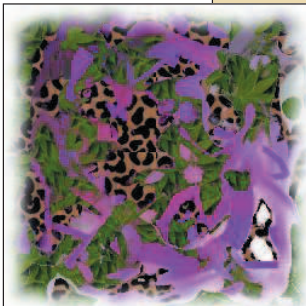
Amennyiben a jövő szempontjából értékesnek tartjuk a Linuxot, a jövő nemzedékét kell felkarolnunk, hogy elsajátítsák a Unix-gondolkodásmódot, és megtanulják kezelni a Unix-rendszereket.



„A tanár úr szerint ez a Linux egy idő múlva jobban elterjed majd, és azért tanuljuk?” Jellegzetes kérdés, amit bárki feltehet, és nincs biztos válasz. A kérdést az egyik kilencedikes tanítványom tette föl, miközben a vonaton zötykölődtünk az osztálykirándulásból hazafelé jövet. Ez egyszerre kételkedés és bizakodás. Kételkedés, mert a gyerekek körében szinte csak a Windows ismert mint egyetlen és megdönthetetlen játékközlő; legtöbbjük az iskolát kivéve még sosem látott Linuxot. Bizakodás, mert elszántságomat és határozottságomat látva meggyőződhetek róla, hogy „márpedig itt a Linuxot komolyan fogjuk venni, és meg fogjuk tanulni”, azaz furcsa tanárunk furcsa hobbjára remélhetőleg értékálló befektetés lesz.



Mondtam, bízom benne, hogy a Linux el fog terjedni, azután szépen lassan rátértem „széles út, keskeny út” elméletemre, vagyis hogy a tömegnyomás nem törheti le a szakmát, és ha a kevesek képviselik az értékálló tudást, akkor a kevesekkel kell tartani. Hogy a világ már csak ilyen, és az élet minden területén sokszor előfordul, hogy az árral szemben kell úszni; hogy a látható dolgok vakító csillogása sokszor elfedi a láthatatlan dolgok nagyszerűségét, mélységét.



Itt van például a Linux-konzol. Milyen nagyszerű, hogy rövidebb-hosszabb parancssorokkal képeket alakíthatunk át egyik formátumból a másikba, ide-oda kattintgatások nélkül:

```
for i in *.png; do convert $i
  jpeg:'basename $i .png'.jpg;
done
```

Ez csak egy lehetőség, amit például egy olyan órán mutathatunk meg, amikor weboldalt készítünk a gyerekekkel. „Az egyszerű szép.” Természetesen ahhoz, hogy mindez működjön, sok-sok programozó összehangolt munkája igényeltetik. A Linux látásmódjának megértéséhez önzetlenségre, önfeláldozásra és a csapatmunka élményének megélésére van szükség. Emellett az is számít, hogy a szabad program nem fekete doboz: tanulmányozható, hogy „mitől megy”, de az sem utolsó szempont, hogy a BSA-tól sem kell többé tartanunk. Ezek mind komoly érvek, amelyek esetleg évek hosszú sora alatt érnek meg – nem csak – a gyerekekben. A legfontosabbnak mégis a pozitív közösségi élményt tartom, ezért tanári munkám során rendszeresen olyan feladatokat igyekszem adni a gyerekeknek, amelyek megoldása során

bizonyos feladatokat közösen kell elvégezniük. A legegyszerűbb példa a programozói projektmunka, ahol a tervezést, a kivitelezést és a munka például bemutatóként történő leírását egyszerre több gyerekre rá lehet bízni. Sajnos, nem minden gyereket sikerül fellelkesíteni. Egy-egy tanuló szívesen veszi és örül, ha elmegyek a lakásukra Linuxot telepíteni, vagy egyszerűen „megnézni”, mi baja a számítógépüknek. Mások attól is elzárkóznak, hogy az iskolában a délutáni szabad gépidőben a közelgő dolgozatra gyakoroljanak, még akkor is, ha segítséget ajánlok fel nekik. Ennek oka lehet a tanulók túlterheltsége, a kamaszkor, és sokszor az is, hogy a tanárokat a társadalom csak lassan kezdi ismét elismerni. A Linuxot, mivel a diák nem találkozik vele lépten-nyomon, csak nagyon sok további idő ráfordításával tudja megtanulni. Olyan ez, mint bármilyen más tudomány: egyesek önállóan, a témába beleszeretve kíváncsiságból tanulnak, másokat meg kell győzni, hogy például az egyetemen több lesz a Linux, mint a Windows, és lépéselőnybe kerül, ha már a középiskolában megismeri. Örvendetes, hogy egyre több magyar nyelvű linuxos segédanyag létezik, bár az oktatás területén még sok a lefedetlen terület. Egy bizonyos szinten túl elengedhetetlen az angol nyelv ismerete, amely követelményt az idegen nyelvek oktatásában akár kedvező módon is fel lehet használni.

Linux a tanórán

A 2001–2002-es évben a 7. és a 9. osztályban, matematika tagozaton tanítottam Linuxot az informatika tanóra keretein belül, a Windows mellett. Kollégáim csak Windowst tanítanak, és nekem is az a meggyőződésem, hogy „mindkét” operációs rendszert be kell mutatni, de olyan módon, hogy elsősorban szemléletmódot, és ne adott programokat tanítsunk.

A legjobb lenne, ha egy kivetítő segítségével a gyerekek számára azonnal látható módon be tudnánk mutatni, hogy „mikor hova kell kattintani”. Így jobban együtt bírnánk dolgozni. Ekkora beruházásról azonban egyelőre álmodni sem merünk. Marad az módszer, hogy néhány alapvető fogást közösen, egyszerre haladva nézünk meg, majd a teremben ülő 13–16 gyereknek megadunk feladatokat adok (12 gép van a teremben), és egyenként segítünk, ha elakadnak. Egy másik lehetőség, hogy önálló feldolgozásra 10–20 pontból álló feladatsort adok ki, amelyben pontosan le van írva, mit hogyan kell tenni, mit kell beírni stb.

A tanterv szerint a cél elsősorban irodai programok tanítása, azaz a szövegszerkesztés, a táblázat- és adatbázis-kezelés van napirenden. Emellett bizonyos szintű algoritmikus gondolkodásra is meg kell tanítanunk a diákokat. Mindehhez olyan alapra van szükség, ami a számítógép működésének megértését is feltételezi, így a szokványos 7. és 9. osztályos anyag a gép-program fogalmával, az operációs rendszer alapjaival kezdődik. Sok középiskolában jellemzően DOS-t tanítanak, ami a tanulók szerint is idejétmúlt. Korszerűbb, ha a DOS (és a grafikus Windows) mellett a Unix-alapparancsokat is





oktatjuk, elmondva, hogy a DOS a Unixból alakult ki, voltaképpen annak egy lebutított változata. Több párhuzamot is lehet vonni a DOS és a Linux rendszertöltő folyamata között, talán a Linuxé annyival izgalmasabb is, hogy „látjuk”, mikor mi történik. Kár, hogy a Windowsnál ugyanez egyre kevésbé tanítható, mivel a Microsoft nem köti a felhasználó orrára, hogy a rendszert milyen lépésekben éleszti fel.

Programok

Az elmúlt tanévben még egyik osztályban sem tanítottam StarOffice-t, hanem a Microsoft Word–Excel párost mutattam be; sok tanuló elég magas szinten ismeri már ezeket a programokat, különösen a Wordöt. A következő tanévben a StarOffice (vagy OpenOffice) mellett a LyX és a TeX bemutatását tervezem, melyek a matematikai osztályokban különösen motiválók lehetnek, például matematikai versenyfeladatok megoldásainak gépeléséhez.

Az idén a 7. osztályban 2–3 hónapon keresztül tanítottam a C nyelvet, miután komoly szavazást tartottunk a gyerekekkel arról, hogy Pascalt vagy C-t szeretnék-e tanulni. Egy egyetemi kollégám ugyanis nehezményezte, hogy miért Logót, illetve Pascalt tanítunk a lány osztályában, miért nem C-t, amikor ez utóbbinak sokkal több gyakorlati haszna van. Korábban magam is írtam egy magyar nyelvű, csak az alapvető szolgáltatásokat tartalmazó Logót, ami először csak DOS alatt futott, de egy éve SVGAlib-támogatással már Linuxon is elérhető (☞ <http://logonyelv.sf.net>); a Comenius Logóval (az iskolákban szinte kivétel nélkül ezt tanítják) azonban nem veszi fel a versenyt. Az első félévi logózás után tehát a C nyelvre tértünk át.

A C a kicsik számára nehéznek bizonyult, még akkor is, ha kifejezetten jó képességű, matematika tagozatos osztályról volt szó. Sok apró szabályt kell megtanulni, és ezekhez nem lehet elég sok gyakorlati példát mutatni. Néhányan azonban előnyre tettek szert a tanórák anyagából, és a délutáni C-szakkörökön néhányuk komolyabban is hozzá tudott szólni az elhangzottakhoz. A *listán* látható programkódot két tanórán keresztül elemeztük, végül a tanulók többsége jó dolgot írt ebből a témakörből is. Sikerült tehát a 7. osztályban megjelenő titkosítási tantervi részt „becsempészni” a programozás témakörébe, ezáltal is elmélyítve a tudásukat. Az ügyesebb gyerekek ezután a titkosító algoritmust tetszés szerint módosíthatták, például az ábécét megfordítva, vagy az eredeti három betű helyett más eltolási értéket is választhattak.

A 9. osztályban, szintén matematika tagozaton Pascalt tanítottam. Ez utóbbi osztályt második éve oktatom, tavaly Pascalban kezdtünk. A gépeken még csak a DOS-os DJGPP volt fenn, ami sajnos nem megbízható, emiatt a C-vel nem is számoltam. Kilencedikben több a lány, mint hetedikben, és a tapasztalat azt mutatja, hogy a lányoknak nehezebb a számítástechnika, mint a fiúknak; így a felsőbb osztályban sokszor könnyítő lépésekre

kényszerültem. Emiatt várhatóan jövőre is a Pascalnál maradok. Nagy örömmre azonban Linux alatt sikerült összekapcsolnom a PostgreSQL adatbázis-kezelőt és a Free Pascalt, így az adatbázis-kezelést teljes egészében PostgreSQL segítségével taníthattam. A pc10-es kiszolgálón kísérleti jelleggel már hosszabb ideje fut egy PostgreSQL/PHP-alapú érdemjegy-nyilvántartó program, és gyerekek egyik összefoglaló feladata az volt, hogy úgy módosítsák a jegeiket lekérdező Pascal programot

```

Julius Caesar titkosítási módszere a tanórán
/* Julius Caesar titkos tÆsi m dszere */

#define MAXHOSSZ 100

main()
{
    char sz[MAXHOSSZ];
    int h,i;
    char b;
    printf("A titkos tand sz veg
    ↪ (csak kisbetűk, Őkezetek ");
    printf("nŀlk l, max. %d karakter):\n",
    ↪ MAXHOSSZ);
    gets(sz);
    printf("A k dolt sz veg:\n");
    h=strlen(sz);
    for (i=0;i<h;i++)
    {
        b=sz[i]+3;
        if (b>'z') b-=26;
        printf("%c",b);
    }
    printf("\n");
}

```

(☞ <ftp://pc10.radnoti-szeged.sulinet.hu/home/kovacs/Linux/PostgreSQL/tanulo.pas>), hogy azt kényelmes legyen kezelni. Önálló adatbázisokat is készítettünk néhány táblával, de a szűkös idő miatt leginkább a SELECT utasítások logikáját tanultuk. Végül „gyárlátogatást” szerveztem programozói irodánkba, ahol barátaimmal már három éve dolgozunk egy szintén PostgreSQL-alapú információs rendszeren.

A PostgreSQL Windowszal is összekapcsolható. A ☞ <http://odbc.postgresql.org> címről is letölthető meghajtóprogram segítségével könnyedén köthetjük össze a Microsoft Office-t és a PostgreSQL-t. A gyerekek akár a születésnap bulijukra szóló meghívókat is létrehozhatták a Microsoft Word körlevél szolgáltatásával, mivel a pc10-en az egész osztálynév sor önműködően rendelkezésre állt. Egy kollégám rendszerint MS Query segítségével mutatja be az SQL-t; az általában tanított Accesst én is túl bonyolultnak tartom, így felhasználtam a Queryt, és ezen keresztül küldtünk lekérdezéseket az SQL-kiszolgálónak.


```

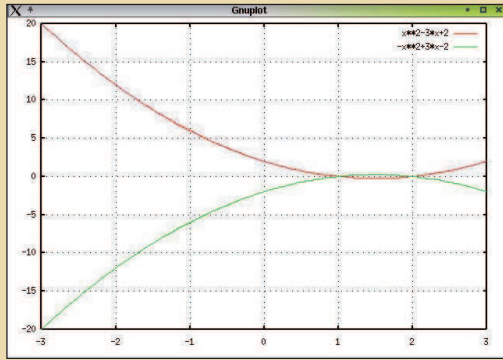
MuPAD: Session Window
File Edit Tools View Help
>> solve((b^2-1)*x=b+1,x);
piecewise( { { b + 1 } if b <> -1 and b <> 1, { } if b = -1, { } if b = 1 }
>> (b^2-1)/(b+1);
      2
      b - 1
      b + 1
>> factor(%);
      b - 1
>> 1/2+1/3+1/4+1/5+1/6;
      29/20
>> expand((3*a+2/5*b)^3);
      3      3      2      2
      27 a + 36 a b + 54 a b
      125      25      5
  
```

```

Konzole
Fájl Munkafolyamatok Beállítások Segítség
zoli@linux:~$ bc
bc 1.06
Copyright 1991-1994, 1997, 1998, 2000 Free Software Foundation, Inc.
This is free software with ABSOLUTELY NO WARRANTY.
For details type `warranty'.
999999^99
999910048508431547643044779651436913965513174063350846119423153483
9921730061928165918468524559784711035127374580003158618417094286484
49965581720563036528559778975449288816486108390262378384273170489214
85367321034857493359075615201815015132094051997884254142894157580919
83816485632238659142795360853044625780935641227630791936321664512
9874551508251856672085825591031369132966364883009949658744782958530
04301750491439343372946474838492976455165511693806251925282518949259
7754483209164432338348725827672754454989876614553527282347744431521
3103842347081552314823562415648995149800898999999
  
```

```

Konzole
Fájl Munkafolyamatok Beállítások Segítség
zoli@linux:~$ gnuplot
GNUPLOT
Linux version 3.7
patchlevel 1
last modified Fri Oct 22 18:00:00 BST 1999
Copyright (C) 1986 - 1993, 1998, 1999
Thomas Williams, Colin Kelley and many others
Type 'help' to access the on-line reference manual
The gnuplot FAQ is available from
<http://www.ucc.ie/gnuplot/gnuplot-faq.html>
Send comments and requests for help to <info-gnuplot@dartmouth.edu>
Send bugs, suggestions and nods to <bug-gnuplot@dartmouth.edu>
Terminal type set to 'x11'
gnuplot> set grid
gnuplot> plot [-3:3] x**2-3*x+2, -x**2+3*x-2
gnuplot>
  
```



➔ <http://www.mupad.de> címről igazából térítésmentesen tölthető le – igaz, a forráskód nem hozzáférhető). A gyerekek hamar rájöttek, hogy a számítógépet hogyan lehet gyorsan „paffra vágni”: a 999999^{999999} hatványozással a program hamar lemerevedik, és csak számol, számol... Ezt a feladványt egyébként a bc-nek is feladhatjuk, garantáltan el fogja vinni a processzort. Számelméleti jellegű feladatok megoldására a bc sokkal jobb, mint a hagyományos (rendszerint grafikus) kalkulátorok, mivel nem normálalakban adja meg a számolási eredményt.

Szakmai ártalom, hogy két további matematikai programot mindig megmutatok a gyerekeknek: a gnuplot függvényábrázolót és a XaoS fraktálrajzolót. A fenti négy program egy-egy jellemző képernyője látható az itt szereplő képeken. A Linux egyébként nem csak informatika órán használható. Mivel másik szakom a matematika, leginkább a matematikai programokat ismerem. De vannak nagyszerű földrajzi, kémiai, fizikai és idegen nyelvi programok is, amelyek egyik legjobb gyűjtőhelye a

➔ <http://schoolforge.net>. Hasznos lenne, ha a magyar számítástechnika-, illetve más szakos tanárok összefognának, hogy a szabadon rendelkezésre álló programokat széles körben megismertessék egymással és a tanulóikkal is. Csak kívánni lehet, hogy – miként a téma az elmúlt hónapokban az érdeklődés középpontjába került – az emelkedő pedagógusbérekkel és a csökkenő kötelező óraszámmal a tanárok alkotóbb, minőségibb munkát tudjanak majd végezni, és ez a hazai iskolarendszerbeli, számítógéppel segített oktatásra is kedvezően hasson majd. Sorozatunk következő, egyben utolsó részében a Linux-számítástechnika szakkörök egy lehetséges megközelítéséről lesz szó.



Kovács Zoltán
(kovzol@math.u-szeged.hu)
tanársegéd a Szegedi Tudományegyetem Bolyai Intézetében az Analízis Tanszéken, matematikát és számítástechnikát tanít óraadóként a szegedi Radnóti Miklós Kísérleti Gimnáziumban. Matematikai tárgyú programok oktatási felhasználásával foglalkozik. Hobbija a Linux programozása és a fraktálok matematikájának kutatása.

Mind a hetedik, mind a kilencedik osztályban nagy sikere volt a Gimp programnak. Az 10. oldalon szereplő képeken látható, milyen ügyesen és alkotó módon tudták a gyerekek használni ezt a nagyszerű grafikus programot az első negyedórás tanári segítség után, különösen a lányok. Később Windows alatt is telepítettük a gyerekekkel a Gimpet, amit többen hazavitték maguknak CD-n (Windows alatt lényegesen megbízhatóbb, de erre a Windowson futó változatot tartalmazó weboldal is figyelmeztet: „Ne csodálkozzunk, ha a Gimp lefagy. Kereskedelmi programok is gyakran fagnak le Windows alatt.”). Matematika tagozatos osztály lévén a MuPAD algebrai rendszernek is nagy sikere volt (ez ugyan nem szabad program, de része a SuSE 7.3-asnak, és a németországi

Linux-index

1. Ennyi országban fontolgatják reklám-kampány vagy mozgalom indítását, hogy megpróbálják elérni a szabad programok kormányzati használatát: **9**
2. Az ALSA (Advanced Linux Sound Architecture) által támogatott hangkártyák és lapkakészletek száma: **94**
3. Azoknak a kártyáknak a száma, amelyekhez van leírás, de még nincs ALSA-meghajtóprogram: **20**
4. Azoknak a hangkártyáknak a száma, amelyek leírását a gyártók nem hajlandók az ALSA Project rendelkezésére bocsátani: **4**
5. Ennyi karakterből áll az a Perl szabályos kifejezés, amely bármilyen érvényes URL címre illeszkedik: **7579**
6. A Linux 1.0 köszönetnyilvánításában szereplő nevek száma: **80**
7. A Linux 1.0 tömörített mérete: **1,2 MB** (hozzájárulónként **15,4 KB**)
8. A Linux 2.4.18 köszönetnyilvánításában szereplő nevek száma: **411**
9. A Linux 2.4.18 tömörített mérete: **28,8 MB** (hozzájárulónként **71,5 KB**)
10. Az „e-mail marketing” üzletág mérete milliárd dollárban: **1**
11. 2006-ra várhatóan ennyi levélszemét érkezik évente minden egyes postaládába: **1500**
12. Az év folyamán ennyi levélszemét gyűlt össze egy nem használt earthlinkes postaládában 2001. augusztus óta: **1200**
13. Ennyi levélszemét gyűlt össze ugyanabban a postaládában 2002. április 18. és május 13. között: **1124**
14. Ennyi levélszemét érkezett a Linux Journal egyik szerkesztőjének postaládájába 2002. május 1-jén: **197**

Források

- 1.: <http://www.lugcos.org.ar/serv/mirrors/proposicion/doc/referencias/#ref.#1>
- 2.: <http://linuxmafia.com/~rick/linux-info/tape-backup>
- 3-4.: <http://www.alsa-project.org/soundcards.php3>
- 5.: <http://www.foad.org/~abigail/Perl/ur12.html>
- 6-9.: <http://kernel.org>
- 10.: San Jose Mercury News
- 11.: Jupiter Media Metrix <http://www.jmm.com>
- 12-14.: a Linux Journal főszerkesztője

Linux Journal 2002. augusztus, 100. szám

Ők mondták

Visszatérés

„Tényleg azt hitte, hogy azt akarjuk, betartsák azokat a törvényeket?” – kérdezte Dr. Ferris. „Éppen ellenkezőleg: az a célunk, hogy megszegjék. Jobb, ha tudja, nem egy cserkészcsapattal áll szemben... mi hatalmat akarunk és komolyan gondoljuk... ártatlan emberek fölött nem lehet uralkodni. Bármelyik kormány hatalma kizárólag abban áll, hogy lecsaphat a bűnözőkre. Amikor pedig nincs elég bűnöző, gondoskodni kell róla, hogy legyen. Annyi dolgot kell bűnnek nyilvánítani, hogy egyszerűen lehetetlenné váljon a törvényeket betartva élni. Ugyan mire jók a törvénytisztelő állampolgárok? Mi haszna van belőlük bárkinek is? Ha viszont olyan törvényt hozunk, amiket nem lehet sem betartani, sem betartatni, de még csak tárgyilagosan értelmezni sem – a nemzet menten törvénytisztelővé válik, és dől a lé a büntetésekből. Nos, így működik a rendszer, Mr. Reardon, erre megy ki a játék, és ha végre megérti, sokkal könnyebben kijövünk majd egymással.”
(Ayn Rand, *Atlas Shrugged*)

Az Internet nyilvánvalóan lényegi szerepet játszik minden e-üzletben, holott pusztán egy az adatátvitel megvalósítására szolgáló, közönséges protokollhalmaz.
(Sybase-hirdetés)

Ha manapság a vállalaton belül egy elejétől a végéig titkosított levél fordul meg, nagy valószínűséggel arra utal, hogy valamelyik alkalmazott kábítószer-kereskedő vagy pornóképeket küldözget.
(Greg Olson elnök és alapítótag, *Sendmail.com*)

Amilyen rossz a mobiltelefonos szolgáltatások minősége, olyan éberen ügyelnek rá a szolgáltatók, nehogy kicsússzon a kezükből a szolgáltatás minőségének felügyelete. Ezért nem akarják megérteni a 802.11 előnyeit. Az ő környezetükben semmilyen víruszerű dolog nem lehet.
(Dave Sifry)

Amennyiben Ön saját honlapjára egy ránk mutató hivatkozást kíván elhelyezni, a hivatkozás kizárólag webhelyünk főoldalára mutathat, tehát nem más oldalára, sem valamelyik altartományra.
(Dallas Morning News, *A szolgáltatás feltételei – a bejegyzés oldalán*)

Ne hagyd magadat becsapni!

A BPDG szabvány célja nem a „kalózkodás” megállítása. Hollywood így próbálja visszanyerni befolyásának egy részét afelett, hogy mit tehetünk és mit nem a televízióval. A szabvány a VCP-dzsinnt akarja visszagyömöszölni a palackba, mely által Hollywood letérheti az új technológiákat, mielőtt azok új lehetőségekhez juttatnák a fogyasztókat.
(Fred von Lohmann)

Hadd jegyezzem meg, hogy a láncolt listák, a „kivág és másol” típusú hulladékgyűjtés és a TABULATOR billentyű szintén szabadalmazva vannak. Ha mindig gondosan ellenőriznénk a programok szabadalmait, soha semmit nem írnánk.
(Martin Pool az rsync gyakori kérdéseiben)

Linux Journal 2002. július, 99. szám;

Linux Journal 2002. augusztus, 100. szám

Linux Tábor 2002

Az idén – immár másodszor – ismét megrendezték a nyári Linux-tábort.

A tavalyi tábor sikerein felbuzdulva beszélgetőtársam, **Czakó Krisztián (Slapic)** az idén is megszervezte a Linux-tábort. A Budapesttől közel kétszáz kilométerre található Szerencsi Középiskolai Kollégium parkjának



árnyas fái alatt beszélgetünk.

Gibizer Tibor (a továbbiakban **Gibzo**): A hazai linuxos élet nagy eseményei között tartjuk számon az általad szervezett nyári tábort. Az elsónél egy kicsit talán meglepődünk, és mélyen magunkba nézve azon tűnődünk, vajon honnan van benned ennyi energia és elszántság, és hányan vannak, akik önköltségi alapon vállalnák egy ilyen komoly szervezést igénylő tábor lebonyolítását?

Czakó Krisztián: Azt sajnos nem tudom megmondani, hogy

hányan vállalnának ilyesmit. Nekem ez nem jelent külön terhet, hiszen lényegében egész évben ingyenes linuxos oktatást tartok. A Linux-tábor talán csak azért kap nagyobb figyelmet, mert egyrészt valóban önköltségi alapon szerveződik, tehát csak annyit fizetnek a résztvevők, amibe szállás, illetve az ellátás kerül; de azt is hozzáteszem, hogy hála és köszönet ezért **Csider Andor** kollégiumvezetőnek, mert ő is önköltségi alapon számolt nekünk mindent. Így sikerült a hihetetlenül alacsony árat elérnünk, ami nagyjából húszezer forintot tett ki egy hétre, azaz egy váltásra. Természetesen ez helyenként csökkent, hiszen a Linux-felhasználók Magyarországi Egyesületének (LME) tagjai kedvezményt kaptak, illetve szállás, és étkezés nélkül is lehetett jelentkezni, ami akár 15 000 forinttal csökkenthette a részvételi díjat, és akkor még nem számoltam bele, hogy a 6–14 éves korosztály féláron vehetett részt, a hat évnél fiatalabb csemeték után pedig nem is kellett fizetni. Ők a családi vakációzás szépségeit élvezték, nem a Linux rejtelmét.

Másrészt talán ez az egyetlen lehetőség, hogy kikapcso-

lódás közben, a nyár örömeiben lubickolva tanulhassanak Linuxot az érdeklődők.

A szervezésről talán még annyit érdemes elmondanom, hogy az idén a tavalyihoz képest megkétszereződött a résztvevők száma, mondhatnám azt is, hogy mindenki hozott magával még egy embert. Komolyra fordítva a szót, a gyakorlatban lényegében csak annyit kellett tenni, hogy eldöntöttük, közhírré tettük, és máris jöttek az emberek. Az előadókát sem kellett különösebben csalogatni, hiszen számukra teljesen ingyenes volt az ott-tartózkodás, és cserében csak arról kellett beszélniük, amiről amúgy is imádnak.

Gibzo: Szerencset nem nevezném üdülőparadicsomnak. Nekem a cukorgyár jut róla az eszembe. Miért éppen erre a településre esett a választásod?

Slapic: Valóban országosan ismert az itteni cukorgyár, de legalább ennyire ismertek a helyi borok, hiszen Szerencs Tokajhegyalja talán legfestőibb részén található. Ezért természetesen a részvételi díjban szerepelt egy ingyenes pincelátogatás, ahol igazi fahordókból kóstolgathattuk a hegy levét. Tehát miért Szerencs? Ennek több oka is van. Lenyűgöző az a csend és nyugalom, ami az itteni életet jellemzi. Itt sikerült **Csider Andor** személyében megtalálni azt az embert, aki ennyire olcsón ilyen tökéletes körülményeket tudott biztosítani. Nézz körül! A kollégium épülete olyan, mint egy szálloda, egész nap ingyenes internetezési lehetőség, 3–4 ágyas szobák, nappali tévével, napi háromszori étkezés, ami igazán házias – és itt nem csak az ízekre, de a mennyiségre is gondolok. Alig 300 méterre van a gimnázium, ahol három Pentium II–III gépekkel szerelt terem áll a rendelkezésünkre, amelyek egyenként 15–20 fő befogadására alkalmasak. Van foci-, tenisz-, teke- és kosárlabdapálya, bérelhetők kerékpárok, vagy a közeli strandok is meglátogathatók.

Gibzo: Nézzük a száraz tényeket. Hányan jöttek el, miként folyt az oktatás?

Slapic: Összesen 90 fő vett részt a táborban, csoportonként öt-öt oktatóval. Úgy érzem, megérdemlik, hogy név szerint is megemlítsm őket: *Szalai Ferenc, Balázs Tibor, Deim Ágoston, Magosányi Árpád, Tomka Gergő, Haluska György*, és természetesen én is igyekeztem kivenni a részemet az oktatásból. Tudom, ez nem tíz fő, csak hét, de ennek az az oka, hogy voltak olyanok, akik mind a két hetet végigdolgozták. Napirendünk szerint az ébredés időpontja szabadon választott volt, azonban csak reggel 9 óráig lehetett reggelizni. 9-től 13 óráig tartottak az órák szünetekkel megtűzdelve. Utána ebédeltünk. A délutáni program a szabadon választott kategóriába tartozott, tehát a pincelátogatás sem volt kötelező, sőt az általam főzött bográcsos borjúpörkölt sem, amit itt a kollégium parkjában készítettem el. Hozzá kell tennem, hogy nem vagyok avatott szakértője a konyhaművészetnek, de az első „bétapörkölt” után mára büszkén kijelenthetem, hogy sikerült elérni a „stabil” változatot. Visszatérnék az oktatásra. Akik figyelemmel kísérték eddigi tanfolyamaimat, azok talán nem





lepdőnek meg azon, hogy most is a lehetőségekhez mérten igyekeztünk igazodni a tanulni vágyókhoz. Tehát nem használtunk szoros témaköröket, inkább arra helyeztük a hangsúlyt, hogy az érdeklődők igényeinek a lehető legmesszebbmenőkig eleget tegyünk. Három csoportban folyt az ismeretek átadása. A kezdők megismerkedhettek a Linux történetével, egy Debian Woody GNU/Linuxot telepítettek, elsajátították az alapvető parancsok használatát, majd az utolsó mozzanatok egyiként egy grafikus felületet telepítettek és állítottak be. Természetesen ide kívánczolt a Világháló elérésének és a levelezés alapjainak ismertetése is. Lényegében ez a szint volt az, ami az otthoni felhasználáshoz elengedhetetlen. A középhaladók a kezdőknek szóló anyag mélyebb ismertetésén túl közelebbi ismeretséget kötöttek a hálózatok rejtelmeivel, tehát szerepelt benne Apache-oktatás és a levelezőrendszerek beható tanulmányozása. A haladók a hálózati biztonság témakörét járták körül, valamint meglehetősen mélyenszántóan megismerhették a rendszer működését egy nem szok-

ványos csomagonkénti telepítés közben.

Gibzo: Pár szóban értékelnéd az elmúlt két hetet?

Slapic: Nos, ez nem könnyű. Sokan voltunk, a hangulat és az időjárás jó volt, rengeteget pihentünk, és hihetetlen mennyiségű tudást adtunk át. Manapság nem szívésen jósolok előre, most azonban úgy érzem, hogy lesz még folytatás.

Gibzo: Köszönjük szépen az interjút, és a Linuxvilág nevében továbbra is ilyen sikeres táborozást kívánunk!

Kapcsolódó címek

- <http://kollegium.szerencs.hu>
- <http://linuxtabor.webhome.hu>



Gibizer Tibor

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

Linux-index

1. A lemezkiadók körében végzett ellenőrzések az esetek ennyi százalékában találták úgy, hogy a művészt alulfizették: **99,99**
2. A fenti szám ennyi ezer eset vizsgálatából származik: **9**
3. Ugyanebben a csoportban a lemezkiadó ennyi esetben fizette túl a művészt: **1**
4. Ennyi millió dollárba kerül egy slágerfelvétel elkészítése: **0,5**
5. Ennyi családot kísért figyelemmel az a 17 évet átfogó tanulmány, amely a televíziózás és az erőszak közti összefüggést vizsgálja: **707**
6. A Sony hanglemezlemezadásból származó bruttó bevétele (milliárd dollár): **4,6**
7. A Sony elektronikai cikkek értékesítéséből származó bruttó bevétele (milliárd dollár): **40**
8. A hanglemezzipar haszonkulcsa 1994-ben (százalék): **30**
9. A hanglemezzipar haszonkulcsa ebben a tartományban mozog 2002-ben (százalék): **8–10**
10. Az amerikai népesség ennyi százaléka használta az Internetet 2001 szeptemberében: **54**
11. Ennyi millió amerikai használta az Internetet 2001 szeptemberében: **143**
12. Az 5–17 éves amerikai gyerekek ennyi százaléka használ számítógépet otthon és az iskolában: **90**
13. Az amerikai tizenévesek ennyi százaléka használja az Internetet: **75**
14. A Winnebago Industries ekkora összegre számított a Microsoft Exchange frissítésének költségeként (ezer dollár): **150**
15. A Winnebago ennyit fizetett linuxos levelezőrendszeréért, mely összeg nagy részét a vállalat IBM nagygépeinek frissítése tette ki (dollár): **26**
16. Az IBM által 2001 negyedik negyedévében kiszállított összes nagyszámítógépes MIPS ennyi százalékát ültették át Linuxra: **11**
17. Az antarktisi Larsen B jégpárkány kiterjedése 2001 végén (négyzetkilométer): **3,250**
18. A Larsen B jégpárkány súlya 2001 végén (milliárd tonna): **500**
19. Átlagos hőmérsékletemelkedés a Larsen B jégpárkány környezetében az elmúlt ötven év alatt (Celsius): **2,5**
20. Ennyi éven át állt szilárdan a Larsen B jégpárkány: **1 800**
21. A Larsen B jégpárkány 2002-ben ennyi hónap alatt esett szét a felmelegedéstől: **2**

Források

- 1–3.: *Simon Renshaw*, a Dixie Chicks ügyvezetője a New York Timesnak adott interjúban
- 4–6.: New York Times
- 7–10.: US Department of Commerce, Economics and Statistics Administration
- 11–13.: US Department of Commerce (az USA kereskedelmi minisztériuma)
- 14–16.: eWeek
- 17–21.: BBC

Linux Journal 2002. június, 98. szám



lepődnek meg azon, hogy most is a lehetőségekhez mérten igyekeztünk igazodni a tanulni vágyókhoz. Tehát nem használtunk szoros témaköröket, inkább arra helyeztük a hangsúlyt, hogy az érdeklődők igényeinek a lehető legmesszebbmenőkig eleget tegyünk. Három csoportban folyt az ismeretek átadása. A kezdők megismerkedhettek a Linux történetével, egy Debian Woody GNU/Linuxot telepítettek, elsajátították az alapvető parancsok használatát, majd az utolsó mozzanatok egyikeként egy grafikus felületet telepítettek és állítottak be. Természetesen ide kívánczolt a Világháló elérésének és a levelezés alapjainak ismertetése is. Lényegében ez a szint volt az, ami az otthoni felhasználáshoz elengedhetetlen. A középhaladók a kezdőknek szóló anyag mélyebb ismertetésén túl közelebbi ismeretséget kötöttek a hálózatok rejtelmével, tehát szerepelt benne Apache-oktatás és a levelezőrendszerek beható tanulmányozása. A haladók a hálózati biztonság témakörét járták körül, valamint meglehetősen mélyenszántóan megismerhették a rendszer működését egy nem szok-

ványos csomagonkénti telepítés közben.

Gibzo: Pár szóban értékelnéd az elmúlt két hetet?

Slapic: Nos, ez nem könnyű. Sokan voltunk, a hangulat és az időjárás jó volt, rengeteget pihentünk, és hihetetlen mennyiségű tudást adtunk át. Manapság nem szívesen jósolok előre, most azonban úgy érzem, hogy lesz még folytatás.

Gibzo: Köszönjük szépen az interjút, és a Linuxvilág nevében továbbra is ilyen sikeres táborozást kívánunk!

Kapcsolódó címek

- ➔ <http://kollegium.szerencs.hu>
- ➔ <http://linuxtabor.webhome.hu>



Gibizer Tibor

(gibzo@linuxmania.hu)

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

Linux-index

1. A lemezkiadók körében végzett ellenőrzések az esetek ennyi százalékában találták úgy, hogy a művészt alulfizették: **99,99**
2. A fenti szám ennyi ezer eset vizsgálatából származik: **9**
3. Ugyanebben a csoportban a lemezkiadó ennyi esetben fizette túl a művészt: **1**
4. Ennyi millió dollárba kerül egy slágerfelvétel elkészítése: **0,5**
5. Ennyi családot kísért figyelemmel az a 17 évet átfogó tanulmány, amely a televíziózás és az erőszak közti összefüggést vizsgálja: **707**
6. A Sony hanglemezlemezadásból származó bruttó bevétele (milliárd dollár): **4,6**
7. A Sony elektronikai cikkek értékesítéséből származó bruttó bevétele (milliárd dollár): **40**
8. A hanglemezzipar haszonkulcsa 1994-ben (százalék): **30**
9. A hanglemezzipar haszonkulcsa ebben a tartományban mozog 2002-ben (százalék): **8–10**
10. Az amerikai népesség ennyi százaléka használta az Internetet 2001 szeptemberében: **54**
11. Ennyi millió amerikai használta az Internetet 2001 szeptemberében: **143**
12. Az 5–17 éves amerikai gyerekek ennyi százaléka használ számítógépet otthon és az iskolában: **90**
13. Az amerikai tizenévesek ennyi százaléka használja az Internetet: **75**
14. A Winnebago Industries ekkora összegre számított a Microsoft Exchange frissítésének költségeként (ezer dollár): **150**
15. A Winnebago ennyit fizetett linuxos levelezőrendszeréért, mely összeg nagy részét a vállalat IBM nagygépeinek frissítése tette ki (dollár): **26**
16. Az IBM által 2001 negyedik negyedévében kiszállított összes nagyszámítógépes MIPS ennyi százalékát ültették át Linuxra: **11**
17. Az antarktiszi Larsen B jégpárkány kiterjedése 2001 végén (négyzetkilométer): **3,250**
18. A Larsen B jégpárkány súlya 2001 végén (milliárd tonna): **500**
19. Átlagos hőmérsékletemelkedés a Larsen B jégpárkány környezetében az elmúlt ötven év alatt (Celsius): **2,5**
20. Ennyi éven át állt szilárdan a Larsen B jégpárkány: **1 800**
21. A Larsen B jégpárkány 2002-ben ennyi hónap alatt esett szét a felmelegedéstől: **2**

Források

- 1–3.: *Simon Renshaw*, a Dixie Chicks ügyvezetője a New York Timesnak adott interjúban
- 4–6.: New York Times
- 7–10.: US Department of Commerce, Economics and Statistics Administration
- 11–13.: US Department of Commerce (az USA kereskedelmi minisztériuma)
- 14–16.: eWeek
- 17–21.: BBC

Linux Journal 2002. június, 98. szám

Jogos – jogtalan – 2. rész

Sorozatunkat folytatva ezúttal a jogok és jogosultságok környékén vizsgálódunk. Elemezni fogjuk, hogy szakmai szempontból nézve mit jelent a törvény szövegében szereplő „jogosulatlan belépés”, valamint „a jogosultság kereteit megsértve bent marad” fogalma.

Btk. 300/C. §

- (1) Aki számítástechnikai rendszerbe a számítástechnikai rendszer védelmét szolgáló intézkedés megsértésével vagy kijátszásával jogosulatlanul belép, vagy a belépési jogosultsága kereteit túllépve, illetőleg azt megsértve bent marad, vétséget követ el, és egy évig terjedő szabadságvesztéssel, közérdekű munkával vagy pénzbüntetéssel büntetendő.

1.

VI. fejezet Adatvédelem

Tv. 30. §

- (1) A jegyző, a 6. § (1) bekezdésében meghatározott időpontig a rendőrkapitányság vezetője, a megyei, fővárosi közigazgatási hivatal vezetője, valamint a Hivatal vezetője a polgárok személyes adatai védelméért való felelősségének körében köteles olyan technikai, szervezési intézkedéseket tenni, ellenőrzési rendszert kialakítani, és adatvédelmi szabályzatot kiadni, amely biztosítja az adatvédelmi követelmények teljesülését.
- (2) Az adatkezelés törvényességének ellenőrzésére a belügyminiszter a Hivatalnál, a megyei, fővárosi közigazgatási hivatal vezetője – az illetékességi területére kiterjedően – a megyei, fővárosi közigazgatási hivatalnál adatvédelmi felelőst nevez ki.

Tv. 31. §

- (1) A nyilvántartás szervei (6. §) kötelesek adatszolgáltatási nyilvántartást vezetni.
- (2) A nyilvántartás tartalmazza
- az adatkezelő nyilvántartási azonosítóját;
 - az adatszolgáltatás idejét;
 - az adatszolgáltatás célját és jogalapját;
 - az adatszolgáltatást igénylő polgár, jogi személy vagy jogi személyiséggel nem rendelkező szervezet, illetve megbízottja vagy képviselője nevét;
 - a szolgáltatott adatok körének megnevezését.
- (3) A polgár az adatszolgáltatási nyilvántartásból jogosult megismerni, hogy mely adatszolgáltatások alanya volt. Ezt a jogosultságot a 24. §-ban felsorolt szervek részére teljesített adatszolgáltatás tekintetében külön törvény korlátozhatja vagy kizárhatja.
- (4) Az adatszolgáltatási nyilvántartást öt évig meg kell őrizni.
- (5) A jegyző nyilvántartást vezet azokról a köztisztviselőkről, akik az eljárás során jogosultak az egyes okmánynyilvántartások adataihoz hozzáférni.
- (6) Az (5) bekezdésben meghatározott nyilvántartás tartalmazza:
- az érintett köztisztviselő természetes személyazonosító adatait;
 - okmánynyilvántartásonként elkülönítve a hozzáférési jogosultságot, a hozzáférési jogosultság kezdő időpontját és időtartamát, valamint a hozzáférési jogosultság módosításának, visszavonásának okát és időpontját.
- (7) A jegyző az érintett köztisztviselő nevét, nyilvántartási kódszámát és a (6) bekezdés b) pontjában meghatározott adatokat legkésőbb a jogosultság kezdő időpontját megelőző munkanapon közli a Hivatallal.
- (8) A nyilvántartás adatait a hozzáférési jogosultság megszűnésétől számított 5 évig kell megőrizni.

2.

E témakör vizsgálatánál kettős – szakmai és jogi – szempontokat kell szem előtt tartanunk; ezek együttesen rendelkeznek a jogokról és a jogosultságokról. Három fő fogalmat szükséges egymástól megkülönböztetnünk:

- *Az információgazda* az adatok felett valamilyen korlátozott szintű vagy teljes jogosultsággal rendelkezik, vagyis rögzítésére, módosítására, törlésére, megismerésére jogosult. Tevékenységének része az információ felhasználása is.
- *Az információ jogosultja* kettős jelentésű: egyrészt arra utal, akitől az információ származik, aki azt keletkezteti (létrehozza a forrást); illetve arra, aki az információmegismerésre jogosított.
- A harmadik a *rendszergazda*, aki az adatok kezelését, megőrzését, hozzáférhetőségének biztosítását technikai szempontból végzi, aki az információgazdák és az információra jogosultak számára számítástechnikai, informatikai vagy egyszerűen dokumentációs-raktározási szolgáltatást biztosít. Tevékenységének tárgya az információ tárolása, védelme, előkeresése.

A jogosultak e három köre technikai és jogi szempontból lényegében elkülönültek tekinthető.

Jogi meghatározottság esetén, tehát amikor valamilyen jogszabály (törvény vagy rendelet) rendelkezik, azaz mind információgazdát, mind információra jogosultakat meghatároz(hat), de rendszergazdát kötelező eljárásokat, jogokat és jogosultságokat is. Szakmai szempontból mindennapi életünk informatikai területe azonban nem jogszabállyal szabályozott, azaz a határvonalak elmosódottak vagy nincsenek. Ezek sokkal több veszélyt jelentenek a gyakorló informatikusok számára. Egy cég fejlődése során elérkezik az a pont, amikor az addig „lazán” kezelt adatbiztonság egyszerre csak komoly büntetőjogi felelősséggel terhelt helyzetet eredményezhet.

A fentebb említett, kicsit száraznak tűnő fogalmakat – az alábbi példával szeretnénk kézzelfoghatóbbá tenni. Az 1992 LXVI. törvény „A polgárok személyi adatainak és lakcímének nyilvántartásáról” címen rendelkezik az adatvédelemről, illetve a jogosultságokról, amelyek e tárgyban a közigazgatást, a polgárokat illetik. Ezt részletesebben a törvény a VI. fejezet „Adatvédelem” részében tárgyalja (lásd a 2. ábrát).

Ennek a törvénynek a 31. paragrafusa informatikusi szemmel nézve semmi mást nem jelent, mint hogy kijelöli az információgazdát, e szövegkörnyezetben az „adatkezelőt”, valamint meghatározza az adatkezelőn keresztül történő információforgalom ellenőrzését, vagyis a „naplózást” a következőképpen rögzítik (lásd a 3. ábrát).

A következő, azaz a (3) bekezdés szerint az információgazda megfelelő feltételek fennállása esetén az információra jogosultaknak információt szolgáltat. Itt a törvény megnevezi a „polgárt” mint az információ forrását, aki jogosult a róla nyilvántartott adatok megismerésére, és jogosult megismerni, hogy mely adatszolgáltatások alanya volt.



Tv. 31. §

- (1) A nyilvántartás szervei (6. §) kötelesek adatszolgáltatási nyilvántartást vezetni.
- (2) A nyilvántartás tartalmazza
 - a) az adatkezelő nyilvántartási azonosítóját;
 - b) az adatszolgáltatás idejét;
 - c) az adatszolgáltatás célját és jogalapját;
 - d) az adatszolgáltatást igénylő polgár, jogi személy vagy jogi személyiséggel nem rendelkező szervezet, illetve megbízottja vagy képviselője nevét;
 - e) a szolgáltatott adatok körének megnevezését.

3.

- (3) A polgár az adatszolgáltatási nyilvántartásból jogosult megismerni, hogy mely adatszolgáltatások alanya volt.

Az (5) bekezdés megszabja a más információra jogosultak körének kijelölési szabályait, tehát azok körét, akik valamilyen oknál fogva az „okmánynyilvántartások adataihoz hozzáférni” jogosultak.

- (5) A jegyző nyilvántartást vezet azokról a köztisztviselőkről, akik az eljárás során jogosultak az egyes okmánynyilvántartások adataihoz hozzáférni.

A továbbiakban a (6) bekezdés b) pontja időkorlátokat állít a jogosultak adat-hozzáférhetőségéhez, mely szerint a hozzáférési jogosultságnak kezdő és vég időpontja van. A végző időpont lehet időtartammal kijelölt, de visszavonásig terjedő is.

- (6) Az (5) bekezdésben meghatározott nyilvántartás tartalmazza:
 - a) az érintett köztisztviselő természetes személyazonosító adatait;
 - b) okmánynyilvántartásonként elkülönítve a hozzáférési jogosultságot, a hozzáférési jogosultság kezdő időpontját és időtartamát, valamint a hozzáférési jogosultság módosításának, visszavonásának okát és időpontját.

A törvény a végrehajtási rendeletben rendelkezik a „rendszergazda” feladatairól, tehát a technikai oldalról (lásd a 4. ábrán).

Ennyi példa után nézzük meg a Büntető törvénykönyv 300 C § (1) „Aki számítástechnikai rendszerbe a számítástechnikai rendszer védelmét szolgáló intézkedés megsértésével vagy kijátszásával jogosulatlanul belép” részének értelmét. Ha tehát valaki szándékosan – vagyis nem a saját nevében és jelszavával – jut be egy számára nem jogosított adatállományba, megvalósítja a 300 § C (1) bekezdés szerinti vétséget.

Hasonlóan kell értelmezni minden olyan eljárást, amellyel valaki a számítástechnikai rendszer védelmét szolgáló intézkedés megsértésével vagy kijátszásával jogosulatlanul lép be. A szándékos jogosulatlan belépés maga a tényállás lényegi eleme!

Vhr. 42. §

- (1) A nyilvántartás szerveinél az adatvédelemért felelős vezető, valamint a számítástechnikai rendszerek üzemeltetője gondoskodik az adatállomány fizikai megsemmisülés elleni védelméről, a rendszerben alkalmazott adatkezelési eljárások és az adatállományok biztonságáról, valamint az illetéktelen hozzáférés elleni védelemről.
- (2) A nyilvántartás kezelője, illetve üzemeltetője
 - a) gondoskodik a fokozott tűzvédelemről, valamint lehetőség szerint folyamatos áramforrást alkalmaz;
 - b) nyilvántartja a mágneses adathordozókat, és azokat a biztonsági előírásoknak megfelelően kezeli;
 - c) rendszeresen menti az adatállományt, és a másolatot olyan helyen tárolja, ahol biztonságosan megőrizhető és hozzáférhető. A helyi állomány biztonsági másolatául a területi, továbbá a központi számítógépes adatbázis szolgál;
 - d) gondoskodik a vírusvédelemről.

Vhr. 44. §

- (1) A nyilvántartó szerv, az üzemeltető, illetve az adatszolgáltatást közvetlenül (online) igénybe vevő szerv dolgozói csak meghatározott jelszó és azonosító használatával férhetnek hozzá az adatállományhoz, amelyek egyidejűleg meghatározzák az adathozzáférési jogosultság mértékét is.

4.

A „jogosultság kereteit megsértve bentmaradás” esetében a jogosultság időkorlátját valaki szándékosan lépi túl, és ezáltal valósítja meg a törvényi tényállást. Hogyan tehetjük meg? Ha például a rendszergazda a jogosultságának lejártával a jogosult hozzáférést nem törli, ezért neki lehetősége marad az (immár) illetéktelen hozzáférésre. Amennyiben illetéktelen hozzáférést szándékosan használja, az büntetendő. Érdekes, vitára okot adó gondolat a fenti 300 § C (1) értelmezése olyan esetekben, amikor az informatikai munka egy cég életében nem jogszabályokon alapul, ráadásul szervezetlen is (lássuk be, ez igen gyakori), tehát a felelősségi határok elmosódottak. Vajon ekkor hogyan alakul a büntetőjogi megközelítés? Erre a következő alkalommal térünk vissza.



Granek István

szervezőmérnök, kajak-kenu szakedző, és immár Linux-rajongó is. 1998 óta helyi hálózatok tervezésével és kiszolgálóépítéssel foglalkozik, valamint a JPTE harmadéves joghallgatója. Szeret síelni, biciklizni, de úszni is.

CD-író program karakteres felületen

Beszélgetőtársam Várkonyi Balázs, a „cdw” karakteres CD-író program alkotója. Munkatársunk a szerzőt a program születéséről és a felmerülő nehézségekről kérdezte.

Gibizer Tibor (a továbbiakban **Gibzo**): *Ma már egyre többen programoznak Linux alatt. Te mikor ismerkedtél meg ezzel a csodálatos rendszerrel?*

Várkonyi Balázs (a továbbiakban **vbali**): *Megközelítőleg hat éve találkoztam először a Linuxszal az egyik ismerősömmel. Akkor már sokat hallottam róla, és mindenki, aki*

ismerte, dicsérte. Nekem akkoriban még nem nagyon nyerte meg a tetszésemet, mivel túlságosan fapadosnak találtam, egy csomó dolgot nem értettem, amivel egy kezdő linuxos általában találkozhat.

Azóta seregnyt váltottott a Linux is meg én is. Megközelítőleg fél év serény SuSE-használat után rá kellett jönnöm, hogy az akkor még kezdőknek kiválóan alkalmas SuSE csak megvillantja a Linux-ban rejlő lehetőségeket, azonban az érdemi munkához

Gibzo: *Sokan végigjártuk ezt az utat. Maradjunk továbbra is a múltnál. Mióta programozol, és milyen nyelvet használsz?*

vbali: *1992-ben egy aprócska asztali gépen ismerkedtem meg a Basic csodáival. Akkor még az Enterprise 128-as gépe jelentette számomra a számítástechnika elérhető csúcását. Miután meglett az első PC-m, áttértem Pascalra. Egy éve dolgozom C-ben. Mindig szerettem volna C-ben programozni, de kicsit félttem is tőle. Ezt a lépést fontos volt megtennem, mivel szerettem volna átlépni a Pascal korlátait, és a nyílt forrású programozás – mely már régen megnyerte a tetszésemet – a C nyelv ismerete nélkül nem az igazi. Talán fontos megjegyezni, hogy ezeket a nyelveket mind könyvből tanultam, tanulom. A kezdeti nehézségek után nagyon megszerettem a nyelvet, és mostanra szerencsére már kezdek kicsit „C-sen” gondolkodni.*

Gibzo: *Akkor ebből egyenesen következnek az általad készített programok C nyelven íródnak?*

vbali: *Lassan elmondhatom magamról, hogy igen, bár vannak régebbi próbálkozásaim, amelyek csak egyszerű parancsfájlok, azokra most inkább nem is pazarolnék időt, mert nem túl jól sikerültek.*

Gibzo: *Szerencsére ennek ellenére sem rejtetted el őket a nyilvánosság elől, hiszen a honlapodon megtalálhatók.*

vbali: *Valóban kitétem őket, azonban ezeknek használati értékük nincs, inkább esettanulmányok, vagy nevezhetjük őket apró ötleteknek is, amit már csak azért is érdemes megnézni, hogy lássuk, milyen jó dolgokat lehet a héjprogramozással elérni.*

Gibzo: *Milyen programokkal dolgozol?*

vbali: *Ha programozom, akkor általában a Midnight Commander (mc) beépített szövegszerkesztőjét, az mcedit-et használok. Nagyon szeretem és ugyancsak*

megszoktam.

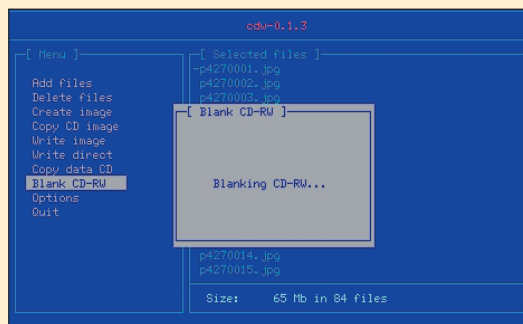
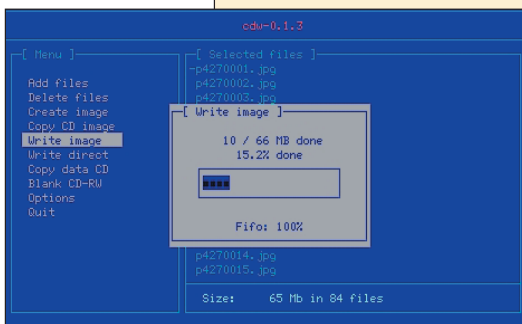
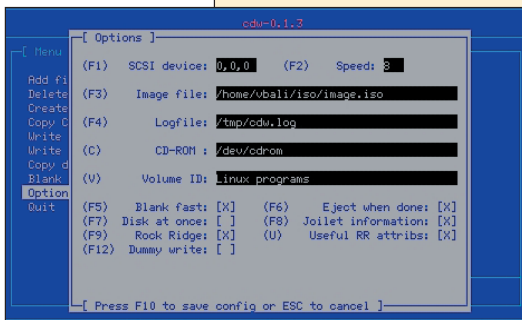
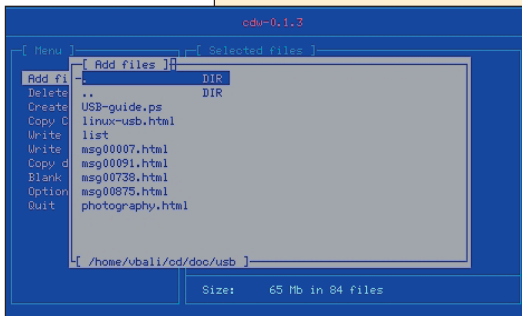
Ha Gtk-t vagy Gnome-ot programozok, az Anjutát használok, amely szintén nagyon jól használható program. Itt ki is merül a munkámhoz használt programok sora, azonban természetesen

valami olyasmit kell keresnem, ami lényegesen közelebb áll a Linux eredeti látásmódjához. Az sem mellékes, hogy nem kell évente kétszer-háromszor komoly összeget kiadnom valamiért, amit szabadon és ingyen is meg tudok szerezni, ráadásul ez a megoldás tényleg azt és úgy teszi, ahogy én szeretném. Nem húzom tovább az időt, hiszen a lényeg az, hogy Debianra (Woody) váltottam, amit lassan egy éve használok. Előtte sok mindent kipróbáltam, Red Hat, Slackware, de a Debian GNU/Linux az, ami a legjobban megnyerte a tetszésemet.

én is használok olyan általános dolgokat, mint az OpenOffice.org (hála és köszönet a magyarítóknak), Gimp, XMMS – bár ezek nem közvetlenül a munkámhoz szükségesek.

Gibzo: *Miért kezdted el CD-író programot készíteni?*

vbali: *Miután sikerült Debian alatt beüzemelni az írómat, elkezdtem próbálgatni az íróprogramokat. Mivel jobban szeretek karakteres felületen dolgozni, főleg olyanokat kerestem, amelyek grafikus felület nélkül is használha-*





tók. A cdrecord önmagában nagyon jó program, mivel azonban kellőképpen lusta vagyok hozzá, hogy mindig kézzel pötyögjem be az adatokat, kezelőfelületet szerettem volna találni hozzá. Van is néhány a „píacon”, de sajnos nem találtam olyat, amit igazán használhatónak nevezhetnék, így elkezdtem írni egyet.

Gibzo: *Jelenleg hol tart a fejlesztés?*

vbali: Gózerővel készül a 0.1.4-es változat, amelyben rengeteg újítás lesz az előző változatokhoz képest. Szerencsére pár srác segítő kezét nyújtott, és közvetlen vagy közvetett módon beszállt a fejlesztésbe. A 0.1.3-as változat debianos csomagja már elkészült és az uhus csomagon is dolgoznak már. Az új honlapot is készítik, mert a jelenlegi enyhén szólva elmarad a mai kívánalmaktól. Ennek csupán az az oka, hogy a honlapkészítéshez nem túlságosan értek, éppen ezért bízom a dolgot olyanra, akinek jobban megy az ilyesmi.

Gibzo: *Mire lenne szükség a továbbhaladáshoz?*

vbali: Legfőképpen több időre. Főállásban egy irodában dolgozom, mellette végzem a főiskolát, ezért elég túlterhelt vagyok. Amikor időm engedi, azért programozatok. Éppen ezért programozókra nagyon nagy szükség volna. Ha többen tudnánk írni a programot, sokkal gyorsabban haladnánk és jobban el tudnánk készíteni az egészet. Már az is nagy segítség lenne, ha valaki átnézné az eddigi munkámat, és egy kicsit „kitakarítaná” a soraimat.

Gibzo: *Említetted, hogy már elkészült a debianos csomag, és az Uhu is folyamatban van. Ebből arra következtetek,*

hogy azt szeretnéd, ha a munkád szerves részévé válna néhány összeállításnak. Jól érzem?

vbali: Természetesen igen. Meggyőződésem, hogy egy programra soha nem mondhatjuk, hogy kész, mivel mindig akad rajta csiszolnivaló, de ha végre eljut a megbízható változatig, biztosan nagyon boldog leszek. Nem titkolt vágyam, hogy egy olyan CD-író programot készítssek, mely a hang- és adatírás mellett a hang-minta-vételezést is tudja, tehát a wav és MP3 formátumot oda-vissza kezeli, MP3-tagszerkesztéssel rendelkezik és sorolhatnám.

Gibzo: *Jelenleg hányan dolgoztok a csoportban?*

vbali: Túl kevesen. 3–4 fő dolgozik rendszeresen a csapatban, de sajnos egyedül vagyok programozó.

A tagok listája megtalálható a

➔ <http://sourceforge.net/projects/cdw> címen.

Gibzo: *Köszönöm a beszélgetést, és jó munkát, sok sikert kívánok a „cdw”-hez.*

Érdeklődő olvasóink Várkonyi Balázsnak

a vbali@westel900.net címre írhatnak.

➔ <http://vbali.freeweb.hu>



Gibizer Tibor

(gibzo@linuxmania.hu)

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

A HP üzenete az alkatrészgyártóknak: áruljatok másnak NDA-t!

Bruce Perens, a Hewlett-Packard Linux-cárja szerint „nagyon jó” esély mutatkozik arra, hogy az újonnan létrejött, legnagyobb Linux-kereskedő, az összeolvadt HP-Compaq a Linuxszal együttműködő alkatrészeket részesítse előnyben. Az új irányelv vállalatszerete megkövetelné, hogy „a tervezés során elsősorban olyan eszközöket vegyenek figyelembe, amelyekhez rendelkezésre áll az illesztőfelület nyilvános leírása”.

A HP tervezőmérnökei számára ez annyit jelent, hogy két lehetséges alkatrész közül azt kell választani, amelyikhez van nyílt forrású meghajtóprogram, vagy amelyikhez programozási kézikönyvet is ad a forgalmazó. NDA-val kísért alkatrész mellett csak akkor dönthetnek, ha nincs más megoldás, vagy a másik megoldás megfizethetetlenül drága lenne. Perens nem latolgatta, hogy a HP mennyivel fogja kevesebbre becsülni az NDA-ba burkolt elemeket. A kérdés úgy is feltehető, hogy a HP mennyivel hajlandó többet fizetni a nyilvános leírással rendelkező eszközökért? Perens azt tanácsolja a tervezőknek, hogy „használják a fejüket”. „Grafikus lapkákkal foglalkoztam” – árulja el. Az ATI alacsony szintű grafikus alkatrészei viszonylag jók, a magas szintűek, azonban véleménye szerint „mind vacakok”. Az új

irányelv eredményeképpen a HP a korábbi NDA-ba bújtatottak helyett már nyílt leírású modemplakákat használ. Az összeolvadást megelőzően a HP-n belül „vállalati irányvonalként” elfogadták a leírt alkatrészeket javasoló rendszabályt, amely a legjobb úton haladt afelé, hogy vállalati hagyománnyá váljon. Az új, egyesült vállalati vezetéssel azonban Perensnek újból jóvá kell hagyatnia. „Az egyesítés sok dolgot megakasztott, és a (compaqos) vezetők tudomására kell hoznom, hogy erre szükségünk van” – jelentette ki. Abból, hogy a HP laptopok és PDA-k Linuxszal is használhatók lesznek, még nem következik, hogy a cég a Linuxot ezeken a gépeken hivatalosan is támogatni fogja. Nem tervezik linuxos laptopok eladását, és nem akarják hasznosítani Jim Getty munkáját, aki azzal foglalkozott, hogyan futtatható Linux és az X a (korábban Compaq, most már HP) iPAQ gépeken. „Főleg linuxos kiszolgálókkal dolgozom” – mondja Perens. Hogyan érinti az összeolvadást a Linuxot a HP-nál és a Compaq-nál? „Hacsak valami szörnyűség nem történik, biztosan a javára válik majd.”

Don Marti

Linux Journal 2002. augusztus, 100. szám



tók. A cdrecord önmagában nagyon jó program, mivel azonban kellőképpen lusta vagyok hozzá, hogy mindig kézzel pötyögjem be az adatokat, kezelőfelületet szerettem volna találni hozzá. Van is néhány a „piacon”, de sajnos nem találtam olyat, amit igazán használhatónak nevezhetnék, így elkezdtem írni egyet.

Gibzo: *Jelenleg hol tart a fejlesztés?*

vbali: Gózerővel készül a 0.1.4-es változat, amelyben rengeteg újítás lesz az előző változatokhoz képest. Szerencsére pár srác segítő kezét nyújtott, és közvetlen vagy közvetett módon beszállt a fejlesztésbe. A 0.1.3-as változat debianos csomagja már elkészült és az uhus csomagon is dolgoznak már. Az új honlapot is készítik, mert a jelenlegi enyhén szólva elmarad a mai kívánalmaktól. Ennek csupán az az oka, hogy a honlapkészítéshez nem túlságosan értek, éppen ezért bízom a dolgot olyanra, akinek jobban megy az ilyesmi.

Gibzo: *Mire lenne szükség a továbbhaladáshoz?*

vbali: Legfőképpen több időre. Főállásban egy irodában dolgozom, mellette végzem a főiskolát, ezért elég túlterhelt vagyok. Amikor időm engedi, azért programozatok. Éppen ezért programozókra nagyon nagy szükség volna. Ha többen tudnánk írni a programot, sokkal gyorsabban haladnánk és jobban el tudnánk készíteni az egészet. Már az is nagy segítség lenne, ha valaki átnézné az eddigi munkámat, és egy kicsit „kitakarítaná” a soraimat.

Gibzo: *Említetted, hogy már elkészült a debianos csomag, és az Uhu is folyamatban van. Ebből arra következtetek,*

hogy azt szeretnéd, ha a munkád szerves részévé válna néhány összeállításnak. Jól érzem?

vbali: Természetesen igen. Meggyőződésem, hogy egy programra soha nem mondhatjuk, hogy kész, mivel mindig akad rajta csiszolnivaló, de ha végre eljut a megbízható változatig, biztosan nagyon boldog leszek. Nem titkolt vágyam, hogy egy olyan CD-író programot készítssek, mely a hang- és adatírás mellett a hang-minta-vételezést is tudja, tehát a wav és MP3 formátumot oda-vissza kezeli, MP3-tagszerkesztéssel rendelkezik és sorolhatnám.

Gibzo: *Jelenleg hányan dolgoztok a csoportban?*

vbali: Túl kevesen. 3–4 fő dolgozik rendszeresen a csapatban, de sajnos egyedül vagyok programozó.

A tagok listája megtalálható a

➔ <http://sourceforge.net/projects/cdw> címen.

Gibzo: *Köszönöm a beszélgetést, és jó munkát, sok sikert kívánok a „cdw”-hez.*

Érdeklődő olvasóink Várkonyi Balázsnak

a vbali@westel900.net címre írhatnak.

➔ <http://vbali.freeweb.hu>



Gibizer Tibor

(gibzo@linuxmania.hu)

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

A HP üzenete az alkatrészgyártóknak: áruljatok másnak NDA-t!

Bruce Perens, a Hewlett-Packard Linux-cárja szerint „nagyon jó” esély mutatkozik arra, hogy az újonnan létrejött, legnagyobb Linux-kereskedő, az összeolvadt HP-Compaq a Linuxszal együttműködő alkatrészeket részesítse előnyben. Az új irányelv vállalatszerete megkövetelné, hogy „a tervezés során elsősorban olyan eszközöket vegyenek figyelembe, amelyekhez rendelkezésre áll az illesztőfelület nyilvános leírása”.

A HP tervezőmérnökei számára ez annyit jelent, hogy két lehetséges alkatrész közül azt kell választani, amelyikhez van nyílt forrású meghajtóprogram, vagy amelyikhez programozási kézikönyvet is ad a forgalmazó. NDA-val kísért alkatrész mellett csak akkor dönthetnek, ha nincs más megoldás, vagy a másik megoldás megfizethetetlenül drága lenne. Perens nem latolgatta, hogy a HP mennyivel fogja kevesebbre becsülni az NDA-ba burkolt elemeket. A kérdés úgy is feltehető, hogy a HP mennyivel hajlandó többet fizetni a nyilvános leírással rendelkező eszközökért? Perens azt tanácsolja a tervezőknek, hogy „használják a fejüket”. „Grafikus lapkákkal foglalkoztam” – árulja el. Az ATI alacsony szintű grafikus alkatrészei viszonylag jók, a magas szintűek, azonban véleménye szerint „mind vacakok”. Az új

irányelv eredményeképpen a HP a korábbi NDA-ba bújtatottak helyett már nyílt leírású modemplakákat használ. Az összeolvadást megelőzően a HP-n belül „vállalati irányvonalként” elfogadták a leírt alkatrészeket javasoló rendszabályt, amely a legjobb úton haladt afelé, hogy vállalati hagyománnyá váljon. Az új, egyesült vállalati vezetéssel azonban Perensnek újból jóvá kell hagyatnia. „Az egyesítés sok dolgot megakasztott, és a (compaqos) vezetők tudomására kell hoznom, hogy erre szükségünk van” – jelentette ki. Abból, hogy a HP laptopok és PDA-k Linuxszal is használhatók lesznek, még nem következik, hogy a cég a Linuxot ezeken a gépeken hivatalosan is támogatni fogja. Nem tervezik linuxos laptopok eladását, és nem akarják hasznosítani Jim Getty munkáját, aki azzal foglalkozott, hogyan futtatható Linux és az X a (korábban Compaq, most már HP) iPAQ gépeken. „Főleg linuxos kiszolgálókkal dolgozom” – mondja Perens. Hogyan érinti az összeolvadást a Linuxot a HP-nál és a Compaq-nál? „Hacsak valami szörnyűség nem történik, biztosan a javára válik majd.”

Don Marti

Linux Journal 2002. augusztus, 100. szám

A Linux 100 pillanata

A Linux Journal 100. megjelenése alkalmából testvér-lapunk bemutatta a Linux történetének 100 legjelentősebb eseményét. Amint azt az időszakon is megfigyelhetjük, a Linux Journal első kiadása egybeesett a Linux 1.0 megjelenésével. Azóta a magazin sorsa nagymértékben követi a Linux életvonalát. Lássuk, mi is történt velük! „A megjelenés óta nyolc, érdekes eseményekkel teli, viharos év telt el. A történések közül mindössze százat kiválasztani egyáltalán nem volt könnyű feladat, és biztos, hogy lesznek olyan olvasóink, akik hamar meg tudnak említeni olyan eseményt, amelyet kiválaszthatunk volna, de nem tettük. Mindenesetre választásunkkal megpróbáltuk hűen visszaadni a Linux történelmét jellemző hullámvasútlejelet. Szeretnénk hangsúlyozni, milyen lekötelezve érezzük magunkat *Rebecca Sobol*-nak és *Jonathan Corbet*-nek a Linux Weekly Newstől pontos és szíves történelmi szerkesztésükért, illetve amiért lehetővé tették nekünk a honlapjukon bemutatott időszak adatainak felhasználását.”

Linux-időszak

1991. augusztus

„Üdv minden Minix-felhasználónak odakinn! Készíték egy (ingyenes) operációs rendszert (csak hobbi, nem lesz olyan nagy és profi, mint a GNU) a 386- (486) AT-kló-nokhoz. Április óta kotyvasztom, és már kezd elkészülni. Szeretnék visszajelzéseket hallani, mit szerettek, illetve nem szerettek a Minixben, mivel az én operációs rendszerem némileg hasonlít rá (többek között azonos a fájlrendszer fizikai kiosztása – gyakorlati okokból). Mostanában ültettem át a bash (1.08) és gcc (1.40) programokat, és úgy tűnik, működnek a dolgok. Ez azt is jelenti, hogy pár hónapon belül valami használható fogok kapni, és kíváncsi lennék, milyen képességeket szeretnének az emberek. Minden javaslatot szívesen veszek, azt viszont nem ígérem, hogy meg is csinálom őket :) – *Linus* (torvalds@kruuna.helsinki.fi)

Ui.: Igen! Nincs benne Minix-kód és többszálú fs-sel (fájlrendszerrel) rendelkezik. Nem hordozható (portable) (a 386 feladatlátszást használja stb.), és lehet, hogy soha nem is fog az AT-merevlemezeken kívül bármi más támogatni, minthogy nekem csak ez van :-).”

1991. szeptember

Megjelenik a Linux 0.01-es változata, és felkerül a hálózatra.

1992. április

Ari Lemmke javasolja, majd be is indítja az első linuxos hírcsoportot comp.os.linux címen.

1992. október

Peter MacDonald bejelenti az SLS-t, az első önálló Linux-telepítést. A javasolt szabad lemezterület legkevesebb 10 MB.



1993. június

Patrick Volkerding Slackware-je lesz az első önálló kereskedelmi terjesztés, és hamar igen

nagy népszerűsége tesz szert a Linux-közösségen belül.

1993. augusztus

Matt Welsh Linux Installation and Getting Started ver-sion 1 című művének kiadása. Ez az első linuxos könyv.

1994. március

Megjelenik a Linux Journal első száma. A szám egy beszélgetést tartalmaz *Linus Torvalds*-szal, illetve

cikkeket *Phil Hughes*, *Robert „Bob” Young*, *Michael K. Johnson*, *Arnold Robbins*, *Matt Welsh*, *Ian A. Murdock*, *Frank B. Brokken*, *K. Kubat*, *Micahel Kraehe* és *Bernie Thompson* tollából. A bemutatkozó kiadás hirdetői közt találjuk az Algorithms Inc., az Amtec Engineering, a Basmark, a Fintronic (amelyet később VA Research, illetve VA

Linux Systems néven ismerünk), az Infomagic, a Prime Time Freeware, a Promox, a Signum Support, az SSC, a Trans Ameritech, a USENIX, a Windsor Tech és a Yggdrasil cégeket. Megjelenik a Linux 1.0.

1994. június

A New Orleans-i tanácskozás alatt *Jon „maddog” Hall* meggyőzi Linust, hogy a Linuxot ültesse át a DEC 64-bites Alpha gépeinek processzorára. Kevesebb mint két hét múlva maddog a DEC-et is meggyőzi, hogy támogassák a projektet – azon nyomban küldtek Linus-nak egy Alpha-munkaállomást.

„A Digital (DEC) és a Linux-közösség együttese volt az első igazán sikeres próbálkozás, ahol az öltönyösök és a Linux-megszállottak (geek) együtt tudtak dolgozni” – vélte maddog.

Jon „maddog” Hall megalapítja a Linux International nonprofit ismeretterjesztő szervezetet. A Linux International jó úton halad afelé, hogy a Linux sikerének egyik főbb részesévé váljon azáltal, hogy a Linux operációs rendszer megismertetésében, használatában és terjesztésében cégeket és másokat is segít.

1994. augusztus

Linux-védjegyvíta: legyen-e védjegy a Linux? *William R. Della Croce, Jr.* 1994. augusztus 15-én kérelmezi a „Linux” védjegyet, amit szeptemberben be is jegyeznek. Della Crocénak nincsen ismert köze a Linux-közös-séghez azt leszámítva, hogy leveleket küld a jelentősebb Linux-cégeknek, pénzt követelve tőlük a „Linux” védjegy használatáért. 1996-ban a bíróság Della Croce ellen hoz ítéletet. Az ügy felperesei közt találjuk *Linus Torvalds*-ot; a Specialized Systems Consultants, Inc. (a Linux Journal kiadója); a Yggdrasil Computing, Inc.; a Linux International; és a WorkGroup Solutions (más néven LinuxMall) cégeket. A felperesek győze-





delmeskednek, és 1997-ben kihirdetik, hogy a védjegy minden felperes és Linux-felhasználó nevében Linus Torvaldsot illeti meg.

1994. szeptember

A Linux első feltűnése a sajtó fő áramában. A Wired magazin megjelenteti *Seth Rosenthal* Kernel Kid című cikkét, amelyben a következőt írja: „Vajon Linus Finnország *Bill Gates*-évé szeretne válni? Valószínűleg nem. Ahogyan saját maga jellemezte önmagát: „semmiképpen sem mintatanuló”, és végezni sem siet, amióta „a Linux olyan sok időt elvesz a tanulástól, és egyébként is szeretem azt a munkát, amit az egyetemen végzek. Ez tart ébren.” *Randolph Bentson* a Linux Journal közönségének bejelenti a világ első, gyártó által támogatott linuxos eszközmeghajtóját. A Cyclades egy több kapuval szerelt soros kártyát (multiport serial card) küldött neki, hogy cserébe fejlesszen hozzá linuxos meghajtót.

1994. december

Egy nagyobb bemutató (tradeshaw) és tanácskozás hívja fel a figyelmet a Linuxra. Az Open Systems World felállít egy Linux-követőt, amelynek a Linux Journal ad otthont. A két napos szemináriumon *Eric Youngdale*, *Donald Becker*, *Dirk Hohndel*, *Phil Hughes*, *Michael K. Johnson* és *David Wexelblat* is felszólal.

1995. április

Lezajlik a Linux Expo, az első Linux-központú bemutató (tradeshaw) és tanácskozássorozat ebédekkel, köszönetnyilvánítással a North Carolina State University munkatársainak, köztük kiemelten *Donnie Barnes*-nek. Az előadók közt találjuk például *Marc Ewing-et*, *Rik Faith-et* és *Michael K. Johnson-t*. A Linux Expo sikeres lesz: a legnépszerűbb és leglátogatottabb éves linuxos eseményé válik az elkövetkező néhány év során (három év múltán a Red Hat veszi át a szervezést és válik a fő támogatóvá). A kiállítás belépti díja és a tanácskozásra szóló jegy ára 4 dollár volt.

1997. január

Az első „Linux-vírus” felfedezése. A vírus neve Bliss, tulajdonképpen bármilyen Unix-szerű operációs rendszeren működik és rendelkezik egy – „bliss-uninfect-files-please” (Bliss, kérlek, fertőtlenítsd a fájlokat) parancssori kapcsolóval. *Alan Cox* rámutat, hogy a Bliss „nem tudja kijátszani a biztonsági rendszert; azon alapul, hogy az előjogokkal bíró emberek valami butaságot tesznek”, és emlékezteti a felhasználókat, hogy kizárólag digitálisan aláírt programokat telepítsenek megbízható gépekről, illetve a telepítés előtt ellenőrizzék az aláírásokat.

„Linux alá valószínűleg könnyebb vírust írni, hiszen nyílt forráskódú, azaz a kódja elérhető. Ezért aztán egyre több Linux-vírust fogunk látni, ahogy az OS egyre gyakoribbá és népszerűbbé válik” – vágyakozó McAfee-s gondolatok.

1998. január

A *Jonathan Corbet* és *Elizabeth Coolbaugh* alapította Linux Weekly News megkezdte tevékenységét. Legelső kiadásuk január 22-én csak egy apró utalás arra nézvést, hogy az LWN mivé szeretne válni.

A Netscape bejelenti, hogy böngészőjük forrását nyílt forrású szabadalom alatt fogják kiadni. Csaknem teljesen biztosan ez marad az év egyik legfontosabb eseménye – ugyanis sok szemet felnyitott, megmutatva, hogy milyen lehetőségeket teremthet a Linux és a szabad programok. A Red Hat Advanced Development Labs (RHAD) alapításának éve. Azóta már az egyik olyan nélkülözhetetlen helyé vált, ahol az embereknek azért fizetnek, hogy szabad programokat fejlesszenek, és lényeges összetevőket készítsenek a GNOME Projecthez. Az RHAD olyan fejlesztőket tudott megnyerni, mint „Rasterman” (igaz, csak egy rövid időre) és *Federico Mena-Quintero*.

1998. február

Megjelenik a Cobalt Cube és – nagy teljesítményének, alacsony árának és ügyes formatervezésének köszönhetően – szinte azon nyomban a kereskedelmi sajtó kedvencévé válik. A Cobalt Linux terveit nem más készítette, mint *David Miller*, akitől igen sok jó dolog származik a Linux-rendszerben.



A Linux-felhasználók közössége megnyeri az InfoWorld technikai támogatás díját (Technical Support Award), a Red Hat 5.0 pedig az Operációs rendszer díjat (Operating System Award). A technikai támogatás díja volt az, ami igazán felnyitotta az emberek szemét – eddig mindenki azt mondogatta, hogy a Linuxnak egyáltalán nincs támogatása. Ez lett tehát az ellenzők „nincs is támogatása” érvének a vége.

Eric Raymond és barátai elkezdik használni a „nyílt forráskódú” (open source) kifejezést. Védjegyért folyamodnak, és felállítják az <http://opensource.org> honlapot. Ilyen módon kezdődnek az első formai erőfeszítések a Linux testületi felhasználására.

1998. március

Ralph Nader fogyasztóvédelmi ügyvéd arra kéri a nagy PC-gyártókat (Dell, Gateway, Micron stb.), hogy ne Microsoft-rendszerekkel, hanem többek között Linuxszal telepített gépeket is árúsítsanak.

1998. április

Az US National Public Radio hírének témája a Linux, jelezve első megjelenését a nem technikai sajtó főáramában is.

O'Reilly megtartja a „legelső” Szabad Szoftver Csúcst (Free Software Summit), ahol *Larry Wall*, *Brian Behlendorf*, *Linus Torvalds*, *Guido van Rossum*, *Eric Allman*, *Phil Zimmermann*, *Eric Raymond* és *Paul Vixie* szerepel.

1998. május

A Google keresőmotor megjelenése. Nemcsak az egyik legjobb keresőmotor, de Linuxon alapul és Linuxszal





foglalkozó keresőlapokat mutat be. Kezdenek a nagy adatbázisok is megérkezni. A Computer Associates bejelenti Linux-támogatását az Ingres rendszerhez, az Ardent Software pedig az O2 object database-hoz.

1998. június

„Számos terméknek, ami ingyenes, hűséges követői vannak, még ha csak kevesen is. Soha nem hallottam, hogy egy vásárló a Linuxról tett volna említést nekem.”
– Bill Gates, *PC Week*, 1998. június 25.

„... ezek az operációs rendszerek az elkövetkező három évben nem fognak széles körben elterjedni a fő üzleti alkalmazások piacán, és nem lesz széles körű, külső fejlesztők (third-party) által nyújtott támogatásuk.”
A Gartner Group szerint a szabad programoknak nem sok reményük van.

A Datapro tanulmány eredménye szerint minden rendszer között a Linux rendelkezik a legnagyobb felhasználói elégedettségi mutatóval; és arra is rámutat, hogy a Linux az egyetlen rendszer a Microsoft Windows NT után, amely a piaci részesedését növeli.

Az IBM bejelenti, hogy terjeszteni és támogatni fogja az Apache webkiszolgálót, miután megegyezett az Apache-csapattal.

1998. július

Az asztalháború elmérgesedik, a KDE- és a Gnome-hívók sértéseket vágnak egymás fejébe. Linus közbeavatkozik, és kijelenti, hogy felőle mehet a KDE. Ekkor jön ki a KDE 1.0-s változata.



A K Desktop Environment első kiadása

népszerű, mindazok aggodalmait ellenére, akik nem szeretik a Qt könyvtár szabadalmát.

Az Informix csendben programokat ad ki Linux alá. Mindeközben az Oracle egy bölcs PR-fogással kiüti az Informix céget, és elsőként jelenti be Linux-barátságát, azt sugallva, hogy hamarosan támogatni fogja a Linuxot. Az Oracle ígéretei szerint 1998 végére már elérhetők lesznek a próbaváltozatok, amelyek hónapokkal a határidő előtt elkészülnek. Ez volt – úgy tűnik – a Linux hosszútávú felhasználhatóságának tűzkeresztsége; ugyanis mind az Informix, mind az Oracle bejelentése igen nagy port vert fel.

Az Informix tulajdonképpen az Oracle-bejelentés utáni pillanatokban hozza nyilvánosságra Linux-támogatását. Később a Sybase szintén kijelenti, hogy támogatni kívánja a Linuxot. Linus feltűnik a Forbes magazin címlapján. A lap hosszú történetet közöl a Linuxról, az igen kedvező hangvételű írás sokak figyelmét felhívja a rendszerre, olyanokét is, akik addig soha nem is hallottak róla. A Linux közismert fogalomná válik.



1998. szeptember

Whitinger és Dwight Johnson beindítja a LinuxToday.com honlapot. A honlap, amelyet később az

Internet.com megszerez, kétségtelenül minden idők legolvasottabb és látogatottabb Linux portálja. Steve Ballmer a Microsofttól bevallja, hogy „aggódnak” a szabad programok miatt, és azt javasolja, hogy a Windows NT forráskódjának egy részét esetleg tegyék elérhetővé a fejlesztők számára. Ugyanebben a hónapban a Microsoft a Linuxot fenyegető versenytársként kezdi jegyezni az éves SEC (US Securities and Exchange Commission) irataiban. Számosan úgy gondolják, hogy valódi céljuk a közelgő antitrösztper befolyásolása.

1998. október

„Jelen pillanatban azonban a redmondi (Washington) cég, úgy tűnik, hálás a Linux felemelkedéséért, ugyanis felismerték benne azt a lehetőséget, amellyel bizonyíthatják, hogy a Windows mégsem monopólium. Ez azonban csak most jutott eszükbe, amikor a cég október 15-én kezdődő antitröszt tárgyalása előtt áll. Ez a rövid távú cél. Hosszú távon a Linux és más nyílt forrású programok még sok fájdalmat okozhatnak Mr. Gatesnek.”
– *The Economist*, 1998. október 3.

Az Intel és a Netscape (és két befektető cég) bejelentik kisebbségi részvételüket a Red Hat Software-ben. A pénzt az „enterprise support division” létrehozására használják fel a Red Haton belül. Az esemény hihetetlen hatást és nagymennyiségű irást vált ki a sajtóból, mivel ebben látják a Linux befogadását a komoly üzleti világba. A Corel bejelenti, hogy a WordPerfect 8 for Linux „személyes használatra” ingyenesen letölthető. Egyben bejelentik partneri viszonyukat a Red Hattal, miszerint Linuxot futtatnak a Netwinderhez.

1998. december

Az IDC jelentése szerint 1998-ban a Linux-szállítások több mint 200 százalékkal, piaci részesedése pedig 150 százalékkal nőtt. A Linuxnak 17 százalékos piaci részesedése van, és növekedési üteme minden más, a piacon résztvevő operációs rendszert messze maga mögött hagy.

1999. január

„A Microsoft Corp. óriási hírveréssel közli majd a világgal, ha a Windows 2000 végül megjelenik. A Linuxot alkotó Linus Torvalds a Linux következő nemzedékét jelentő 2.2-es változat megszületését egy egyszerű megjegyzéssel jelezte a Linux-kernel levelezőlistán.”
– Steven J. Vaughan-Nichols, Sm@rt Reseller
Megjelenik a Samba 2.0., amely visszafejtett Microsoft tartományvezérlőprotokoll-megvalósítást tartalmaz, így téve lehetővé, hogy a Linux-kiszolgálók a windowsos hálózatokban is teljes értékű szolgáltatást nyújtsanak. A Hewlett-Packard és a Compaq bejelenti, hogy Linux-alapú rendszerek eladását tervezik. Később a Dell szintén bejelenti, hogy tervbe vette a Linux-telepítésű rendszerek árusítását. Az SGI is bizonyítja hajlandóságát, amikor nyilvánosságra hozza, hogyan lehet Linuxot telepíteni a rendszerükre.
A Loki Entertainment Software bejelenti, hogy a Civilization: Call to Power Linux alatt is megjelenik.



1999. február

A Linux- és BSD-felhasználók a „Windows Refund Day” (a Windows-visszatérítés napja) alkalmából egyesülnek. Meglátogatják a Microsoftot, remélve, hogy fel nem használt Windows-engedélyüket visszavihetik, amelyet akkor voltak kénytelenek megvásárolni, amikor operációs rendszerrel együtt árusított számítógépet vásároltak.

1999. március

„Ahogyan az orosz forradalmárt kiretusálták a fotóról, úgy hagyják ki a történetírásból. *Stallman* volt a Free



Software-elképzelés és a GNU/Linux operációs rendszer kezdeményezője. Ezt azonban a LinuxWorld (Expo) írásaiból nem tudhatjuk meg. Linus Torvaldsnak jutott az összes tinta.”

– *Leander Kahney, Wired magazine, 1999. március*

Az első LinuxWorld Konferenciát és Expót Kaliforniában, San Joséban tartják. Erre az első nagy linuxos üzleti „bemutatóra”, amely a világ figyelmét felhívta a Linux megérkezésére, állítólag 12 000 ember jelentkezett.

Bemutakozott a Linux Magazine, további versengést hozva a Linux-sajtó világába. Később egyéb magazinok tűnnek fel és el: az Open, a Journal of Linux Technology (JOLT), illetve a Maximum Linux.

A VA Research egymillió dollárért megvásárolja a Linux.com tartományt és bejelenti, hogy Linux-portált kíván létrehozni. A Microsoft állítólagos ajánlata a tartományra meghiúsul.

1999. április

„...képzeld el, milyen érzés látni, hogy egy eszményi projekt megakad és hatástalan, csak azért, mert az emberek nem adják meg azt a megbecsülést, amiért megdolgoztál. Ha olyan idealista vagy, mint én, ez az egész évtizededet elronthatja.”



– *Richard Stallman a GNU/Linuxról*

Al Gore elnöki kampányának honlapjáról azt

állítják, hogy nyílt forráskódú. Később a hírt visszavonják, de a honlapon továbbra is a következők szerepelnek:

„A Gore 2000 Volunteer Source Code Projectet a nyílt forrás haladó szellemében hoztuk létre; a

➔ <http://www.algore2000.com> nyílt honlap”.

A HP bejelenti 24/7 támogatási szolgáltatását a Caldera-, Turbolinux-, Red Hat- és SuSE-változatokhoz. Egyben az OpenMailt is kiadják Linux alá.

A Linux FreeS/WAN projekt kiad egy ingyenes IP Sec-megvalósítást, lehetővé téve, hogy a Linux ma már ipari szabványnak számító VPN-átjáróként működjön.

„Egyszerűen az a tény, hogy egy hivatalos SEC-dokumentum tartalmazza a GPL-szöveget, már épp eléggé bizonyítja, hogy a program üzleti szabályai erősen újraíródnak.”

– *Andrew Leonard, Salon*

1999. május

„Ez a két kicsiny szó: „nyílt forrás”, mágikus jelentést nyert, akár csak a portál 1998-ban (nálunk kicsit később – a ford.), vagy a push 1997-ben (ami a hazai viszonyok

között nem tudott meggyökeresedni – a ford.). Csak ezt kell suttogni, és minden a miénk lehet: a média figyelmé, a fogyasztók érdeklődése és természetesen a kockázati befektetők bizalma”.

– *Andrew Leonard, Wired*

1999. augusztus

Az első Intel IA-64 Merced lapka. Bár az Intel több operációs rendszernek is adott szimulátorokat, a Linux az egyetlen OS, amelyen az új kiépítés már első napjaiban is működik. A Register főcíme: „Megjelent a Merced lapka: a Linux fut, az NT nem”.

Az SGI bejelenti Linux-alapú kiszolgálórendszerét, az 1400L-t. Az SGI egyben közli partneri viszonyát a Red Hattal, és komolyan beszáll a rendszermag fejlesztésébe. Megjelennek a Red Hat induló részvényei (IPO); az utolsó pillanatban végrehajtott átárazás segít megnehezíteni azoknak az embereknek a dolgát, akik a közösség felajánlásaiból vették ki a részüket. A részvényárak azonnal ötven dollárra emelkednek, ami ez idő tájt még magasnak tűnik.

„Immár sokadszor valaki utat mutatott a paradicsomba, és kiépítette a parkolót Linux-programozók ezrei számára, akik létrehozták az utopisztikus nyílt forráskódú – szabad segítség a szabadon használható operációs rendszer megalkotásához – mozgalmat. A Red Hat kezdő részvényeinek sorsa volt a biztos jele annak, hogy a Wall Street átvágta az új Linux-sétány szalagját.”

– *The Industry Standard*

A Motorola linuxos nyilatkozatokat tesz a linuxos beagyzottrendszer-protokollról, a támogatási és oktatási szolgáltatásokról, valamint a Lineóval alakuló partneri viszonyáról.

A Sun megszerzi a StarDivisiont; bejelenti tervét, miszerint a StarOffice-t a Sun Community Source License védelme alatt adná ki, illetve elkészíti az irodai csomag webre felkészített változatát.

1999. szeptember

„A New Jersey-beli, burlingtoni székhelyű Burlington Coat Factory Warehouse Corp. körülbelül egymillió dollárt költ 1250 Linuxszal telepített Dell-számítógép vásárlására, a Red Hatnek azonban egy centet sem fizet a támogatásért” – mondja *Michael Prince* információs vezető.

„Feltételezem, hogy a Red Hat üzleti modellje van, akinek hasznos, viszont számunkra nincs sok értelme”.

– *Daniel Lyons, Forbes, 1999. május 31.*

Szeptemberben a Burlington végül mégiscsak vásárolt támogatást a Red Hattól.

Az első nagy Linux-részvényroham. Az Applix részvényei mennyiségileg több mint a kétszeresére nőnek, elérve a közel 27 millió részvényt – a tényleges piacon lévő kilencmillió részvény háromszorosát.

A SCO Észak-Európában kiadott füzetében a Linuxot becsmerli:

„A Linux jelen pillanatban sokkal inkább tekinthető az IT-tanulók játékszerének, mint komoly operációs





rendszernek, amelyre egy üzlet működését, biztonságát és jövőjét lehet alapozni. Mivel a Linux alapvetően mindenkinek szabad, ez azt jelenti, hogy nincsen olyan ember, illetve cég, akit felelősségre lehetne vonni, ha valami nem jól működik, ráadásul nem lehet előrejelezni, hogy a Linux melyik úton fog továbbfejlődni.”

A Red Hat részvényei elérik a 135 dollár/részvény árat, ami ez idő tájt hihetetlenül magasnak tűnik.

1999. október

A Sun Microsystems nyilvánosságra hozza, hogy a Solaris forrását a Sun Community Source License védelme alatt fogja kibocsátani. A tényleges kiadást azonban bírálatok érik:

„A Linuxra irányuló törekvésként a Sun elhatározta, hogy szerdán bejelenti: az új Solaris 8 operációs rendszer forráskódja *nyílt* lesz. A Webster szótára igen sok meghatározást kínál erre a szóra, többek között a „nem lezárt, rögzített vagy bezárt” is szerepel benne. Ha a Sun-nyilatkozat mélyére ásunk, azt találjuk majd, hogy a felajánlásuk nem igazán találkozik a szótár meghatározásával, ezt a Nyílt Forráskód mozgalomra hagyja.”

– Lawrence Aragon, *Redherring.com*, 2000. január 26.

1999. november

„... ha egy dologgal kellene jellemezni a Linux-felhasználókat, ők a cselekvők, nem a siránkozók.”

– Andy Patrizio, *Wired*

A Red Hat részvényben, csaknem 700 millióért megvásárolja a Cygnust. A Red Hat további beszerzéseiről is felroppenek pletykák, és úgy tűnik, nincs megállás.

1999. december

A VA Linux Systems két átárazás után kilép a piacra (eredeti értéke 11–13 dollár/részvény volt). A végleges IPO-ár 30 dollár/részvény; ez szinte azonnal 300 dollárra szökik, majd záráskor 250 dollár körül végez. Ezzel felállítja a NASDAQ történetének legmagasabb IPO-emelkedés csúcseredményét.

„Húha! Még emlékszem arra, amikor a nagy kérdés az volt: hogyan fogunk ebből pénzt csinálni?!”

– Eric Raymond

2000. január

A VA Linux Systems bemutatja a SourceForge-t (igaz, a honlap maga már 1999 novemberétől elkészült és működött). A SourceForge GPL engedély alatt egyben működésének kódját is elérhetővé tette. Az év végére



a SourceForge 12 000 projektnek ad otthont és 92 000 bejegyzett fejlesztője van.

A Kínai Népköztársaságban megjelenik a Red Flag Linux 1.0.

A Transmeta megtöri hosszan tartó hallgatását, és bemutatja a világnak, mire készül: a Crusoe lapkára.

A Linux Professional Institute bejelenti az első profi Linux-hitelesítő vizsgalehetőség létrejöttét.

A Linuxot sztároló sajtó önti magából az olyan cégek híradásait, amelyek a Linux részvények sikerét próbálják

meglovagolni. A Vitamins.com például a következőt írja: „A Vitamins.com újfent kitért a hasonló internetes egészségügyi honlapok tömegéből, hiszen az első között van, akik a vezető fejlesztő és nyílt forráskódú megoldásokat nyújtó Red Hat által készített Linux operációs rendszert használják.”

2000. február

A legfrissebb IDC-jelentések azt sugallják, hogy a kiszolgáló operációs rendszerek 1999-es eladásának 25 százalékával a Linux most a „kiszolgálókon alkalmazott második legnépszerűbb operációs rendszer”. Az első a Windows NT 38 százalékkal, harmadik helyen a Netware áll 19 százalékkal. Az IDC korábban azt jelezte, hogy a Linux 2002 vagy 2003 táján tornázhatja fel magát a második helyre. A forradalom az előrejelzésnél jóval előbbre jár. A VA Linux Systems egy magas szintű vásárlás keretében megszerzi az Andover.net-et, ahol az Andover-részvények 0,425 VA részvényt érnek, azaz az áruk megközelítőleg 50 dollár/részvény. Az Andover.net a tulajdonosa többek közt a népszerű Slashdot.org és a Freshmeat.org lapoknak.

A LinuxMall.com, illetve a Frank Kaspar and Associates szintén az egyesülést tervezteti. A LinuxMall.com csaknem a kezdetektől a Linux kiskereskedelmi értékesítésének vezetője volt, a Kaspar az egyik legnagyobb hálózat. A Red Hat immár sorozatban a negyedik alkalommal nyeri el az InfoWorld Az év terméke (Product of the Year) díját.

2000. március

„A nyílt kód törvénye kimondja, hogy egyetlen alkotó sem kaphat végtelen hatalmat egy nyílt kód felett. Még a királyok sem nyerhetnek teljhatalmat. Ha például Linus Torvalds, a Linux-rendszermag atyja olyan módon próbálná félrekormányozni a GNU/Linuxot, amit a közösség többi tagja elutasítana, akkor ők egyszerűen kiszedhetnék a sértőnek talált részeket, és elindulhatnának egy



külön úton. Ez a fenyegetés megköti a királyok kezét

– csak arrafelé vezethetnek,

ahová tudják, hogy az emberek követni fogják őket.”

– Lawrence Lessignek a *The American Prospect*ben megjelent „Befektetés, szabályozás és az Internet” című írásából.

Megjelent a LILO új változata, amely képes túllépni az 1024-sávú rendszerindítási határt, ami már évek óta hátráltatta a PC-rendszereket.

A legfrissebb Netcraft-felmérés szerint a Web csaknem hatvan százalékán Apache fut.

Március 21-én, némi késlekedés után, a Caldera Systems



nyilvánossá válik. A részvény, amelyet

14 dollár/részvény áron ajánlottak,

26 dolláron kezd és 29,44-en zár. Így 110 százalékos nyereséget könyvelhet el már az első napon.

„A Caldera egyetlen olyan céget sem ismer, amely nyereséges üzletet hozott volna létre részben vagy egészében nyílt forráskódú programok alapján.”

– Caldera SEC-iratok


slackware
l i n u x

A Walnut Creek (a Slackware anyavállalata) és a BSDi beharangozza egyesülését. A Yahoo! saját tőkés (equity investment) beruházásba kezd az új cégnél. A Motorola Computer Group közlése HA Linux-terjesztésének megszületését. Ez a terjesztés azokat a telekommunikációs alkalmazásokat célozza meg, amelyeknek igen magas rendelkezésreállásra van szükségük; ide értendők olyan tulajdonságok, mint a menet közbeni cserélhetőség (hot-swap capability), illetve az i386- és PowerPC-megoldásokon való elérhetőség. Megjelenik az Embedded Linux Consortium (Beágyazott Linux Konzorcium). Célja „a hatalmas beágyazott piacon elmélyíteni, kiszélesíteni és felgyorsítani a Linux-megoldások terjedését”. Első vezetője *Rick Lehrbaum*, ugyanaz az ember, aki többek közt a LinuxDevices.com és a DesktopLinux.com honlapok mögött áll.

Az Ericsson beharangozza Screen Phone HS210 termékét, amely tulajdonképpen Linux-alapú érintőképernyős telefon, amelyet levelezésre, böngészésre és egyéb feladatokra lehet használni.

Az Ericsson és az Opera Software bejelenti, hogy az Ericsson Linux-alapú HS210 Screen Phone-ja az Opera böngészőt fogja alkalmazni.

2000. április

A kód beszédesnek ítéltetik. 2000. április 4-én az Egyesült Államok feljebbviteli bírósága kihirdette döntését Peter Junger az Exportfelügyeleti szabályozás (Export Administration Regulations) kérdéséről, amely megakadályozta őt, hogy titkosító algoritmusokat tartalmazó példakódokat továbbítsa az Interneten. A döntés legfontosabb része: „Mivel a számítógépes forráskód a számítógépprogramozással kapcsolatos információk és ötletek cseréjének kifejező formája, döntésünk értelmében az Első kiegészítés (First Amendment) rá is vonatkozik.”

Andy Tanenbaum BSD felhasználási szerződés alatt kiadja a Minix operációs rendszert. Ha a Minix a kezdetektől nyílt forráskódú lett volna, a Linux talán meg sem születik.

2000. május

A SuSE kibocsátja az első, IBM S/390 nagygépeket támogató Linux-terjesztést. „Közel 140 terjesztő cég létezik szerte a világon. Úgy gondoljuk, hogy az irányadó öt kivételével valamennyit felvásárolják, kiszorítják a piacról vagy jelentéktelenségre kárhoztatják. A piacon osztozó vezető cégek jelenleg jól körvonalazható földrajzi határokhoz kötődnek. A Red Hat rendelkezik a legnagyobb piaci jelenléttel, és az észak-amerikai piacon is irányító szerepet játszik; Európában a SuSE vezet, Ázsiában a Turbolinux, Dél-Amerikában pedig a Conectiva.”
 – *Keith Bachman*, a *WR Hambrecht* elemzője a *The Red Herring* előrejelzéséről

MySQL

2000. június
 Egyes üzleti megfontolások felgyorsítják a MySQL GPL-alapú szerződés alatti kiadását. Immár két ingyenesen felhasználható adatbázis-

kezelő létezik, melyeket a Linux és a Free Software közösségek széles körben használnak: a PostgreSQL és a MySQL, mivel a Debian Free Software irányelveknek és a Nyílt Forráskód irányelveknek egyaránt megfelelnek. Továbbá a Progress Software NuSphere néven kizárólag a MySQL támogatását célzó új céget alapít.

2000. július

„Az NDA-khoz (titoktartási nyilatkozatokhoz) kötött üzleti világban a Debian nyitott könyv. A feladatnyilatkozatok (mission statements) világában a Debian társadalmi megegyezésen alapul. Amikor az üzleti terjesztők azon ügyeskednek, vajon hány üzleti programot tudnak belepréselni egyetlen Linux-termékbe, a Debian továbbra is a programok szabadságának védőbástyája marad – élő bizonyítékeként a tökéletesen működő és használható operációs rendszernek, minden üzleti kód nélkül meg.”
 – *Evan Leibovitch*, *ZDNet*

A Sun nyilvánosságra hozza, hogy a StarOffice GPL alatt jelenik meg. A kódot újraszerkesztik, beépítik a Bonobo és GTK-rendszerbe, és újrahasznosítható összetevőként adják ki. A StarOffice-t szintén átírják, hogy képes legyen néhány nyílt, XML-alapú fájlformátumot használni. Az Oracle Linux-alapú internetes eszköze megjelenik a polcon.

Napvilágot lát *Larry Ellison* – régóta tartó, a világ számára nem Microsoft-rendszereket nyújtani kívánó személyes hadjáratának – legújabb eredménye, a „New Internet Computer” (NIC). Azokat az embereket célozza meg, akik csak a Hálózatot kívánják elérni. A termék tulajdonképpen egy (monitor nélküli) 199 dolláros X-terminál.

Megérkeznek az első jelentések, miszerint a Caldera esetleg megveszi az SCO-t. Később, 2000-ben a Caldera és a SCO bejelentik a Caldera International kialakítását, amely a Caldera létező tevékenységéből és a három SCO-részleg közül kettőből alakulna.

Ted Ts'o az új 2.4 állapotlista kezelőjeként (status list maintainer) lép elő. Ezt a munkát *Alan Cox* végezte, míg ki nem jelentette, hogy ideje volt „valaki mást keresni a fenntartására”. *Ted Ts'o* végül választott Linus többszöri kérésére, és elfogadta az állapotlista-kezelői helyet.

2000. augusztus

A HP, az Intel, az IBM és a NEC bejelentik a nyíltforráskód-fejlesztő labor (Open Source Development Lab) megalakulását, amely a teljesítményméréshez és kipróbáláshoz számos eszközt elérhetővé tesz a Linux-fejlesztők számára.

2000. szeptember

„Mocsok vagyok. Fogalmam sincs, miért gondolják másként az emberek. Pedig úgy gondolják. Az emberek azt képzik, hogy rendes ember vagyok, pedig a valóság az, hogy áskálódó, megalkuvó mocsok vagyok, akít nem érdekel, ha belegázol mások lelkébe, és nem hatnak meg az elveszett munkaórák, csakis akkor, ha olyasmit eredményeznek, amit jobb rendszernek tartok.”
 – *Linus Torvalds*, *amint éppen megpróbálja megváltoztatni a róla kialakult képet*





Lejár az RSA-szabadalom, lehetővé téve az üzleti programok nélküli biztonságos webes vásárlásokat.

A Trolltech GPL alá helyezi a Qt könyvtárat, egyszer és mindenkorra pontot téve a régóta húzódó, kínos szabadalom körötti civakodás végére.

CueCat-fiaszkó. A Digital Convergence megpróbálja lecsuktatni azokat a programozókat, akik Linux-vezérlőt írtak a CueCat vonalkódolvasóhoz. A cég nagy számban terjesztett ilyen ingyenes vonalkódolvasókat, azt várva, hogy az emberek kényszerűségből az üzleti programmal és honlappal használják majd őket. A fenyegetés miatt a vezérlőt egy rövid ideig valamivel nehezebb volt megtalálni a Neten, amit a cég győzelemként értékelt, és gyorsan továbblépett.

2000. október

A Microsoft egy európai hirdetésben azt állítja, hogy a pingvinek hamar mutációnak indulnak – a hirdetés rövid idő alatt nagy népszerűsége tesz szert.

2000. december

„Döbbenet fedeztem fel, hogy a Linux telepítése milyen egyszerű. Miért? Nos, a világ megváltozott. Többé már nem kell a telepítés előtt megértenünk mindent a Linux-szal kapcsolatban, letöltenünk azt a rengeteg kódrészletet, ami a teljes értékű rendszer futtatásához szükséges, majd még a „BSW” szabvány előtti ősrégi csomagolástechnikával összeraknunk őket. Az olyan cégek, mint a Red Hat és a Corel, minden nélkülözhetetlen programot betesznek a termékbe, és a gond (többnyire) a múlté.”

– *John Schwartz, Washington Post*

Az IBM bejelenti, hogy egymilliárd dollárt szándékozik 2001-ben a Linuxba fektetni.

2001. január

Január 4-én megérkezik a várva várt 2.4.0-s rendszermag. Az US Nemzetbiztonsági Ügynökség (National Security Agency – NSA) GPL alatt kiadja a SELinuxot. A SELinux a hagyományos Unix-szerű jogosultságrendszeren felül egy további biztonsági réteget is nyújt.

2001. március

San Joséban (Kalifornia állam) megtartják a Linux 2.5 rendszermag-fejlesztői találkozóját; ez talán a világtörténelem legteljesebb Linux-magfejlesztői összejövetele.

2001. április

Az IBM-nek néhány városban „Peace, Love and Linux” (Béke, szeretet és Linux) feliratú graffitijeivel gondoljai támadnak.

„A Slackware mindig is pénzt hozott a házhoz (melyik másik üzleti alkalmazágyártó mondhatja el ugyanezt?), csakhogy a BSDi-vel végül a süllyedő hajón végezzük.”

– *Patrick Volkerding*

2001. május

A Japánban árusított Sony Playstation Linux-készlet nyolc perc alatt elfogy, annak ellenére, hogy az áru-készletet megkésztették.

2001. június

A Sharp beharangozza Lineo Embedix rendszerén alapuló linuxos tenyérgepének (PDA) küszöbön álló megjelenését. Az alkatrésziparban tevékenykedő VA Linux Systems inkább a SourceForge-ra szeretne összpontosítani. Később a VA teljes egészében kiejti nevéből a „Linux” szót, és VA Software Corporation néven indul újra. A szerda délután megjelent sajtóhírekben *Larry M. Augustin*, a VA Linux vezérigazgatója (CEO) a stratégia-váltást logikus lépésnek nevezte

„Megkülönböztető erőnk mindig is a programszaktudásunkban volt – mutatott rá Augustin.”

– *Wired*

Ezek szerint csak mi gondoltuk azt, hogy a VA alkatrészgyártó cég volt.

2001. július

Szabadítsuk ki Dmitryt! *Dmitry Sklyarov*-ot letartóztatják Las Vegasban, miután az Adobe panaszt tesz



az Advanced eBook Processor miatt.

A következő hónapban elítélik a DMCA (digitális szerzői jogi törvény) megsértéséért és összeesküvésért: a legnagyobb kiszabható büntetés felső határa 25 évig terjedő szabadságvesztés.

Dmitry védelme a DMCA alkotmányosért voltának elemzésén, a szólásszabadsággal kapcsolatos és jogszolgáltatási kérdéseket feszegető tételeken alapul. Az év hátralevő részében a vádakot feltételelesen ejtik – az egy év jó magaviselet és az ElcomSoft tárgyalások bizonyítékai alapján.

„Bár az Adobe visszavonta a Dmitry Sklyarov ellen felhozott vádakot, tiszteletben tartjuk az esküdszék és a szövetségi kormány döntését, miszerint az ElcomSoft céget pereljük be mint törvénszegő jogi személyt. Az Adobe szeretne teljes mértékben együttműködni a kormánnyal, ahogyan azt a törvény megköveteli.”

– *Az Adobe álláspontja*

2001. november

A Sharp Electronics Corporation nekikezd a Zaurus tenyérgep különleges Linux fejlesztői előkiadásának, hogy a szabad programok fejlesztőit erre az új felületre csábítsa.

2002. február

Az Avaya, a korábbi PBX és a Lucent vállalkozási rendszerrészlege nyilvánosságra hozza a Linux-alapú PBX rendszerek megjelenését.

„Akadnak néhányan – magamat is közéjük sorolom –, akik úgy gondolják, jó dolog visszatérni a Földre. Semmi rossz sincs abban, ha az ember pénzt szeretne keresni, és a Linux nem sokat nyer vele, ha egy bizonytalan alapítvány hátán a földhözragadt világ fölé emelkedik.”

– mondja *Evan Leibovitch* a Linux rohanó világáról.

Linux Journal 2002. augusztus, 100. szám

Tudod-e?

1. A képen *Richard M. Stallman* látható, amint az

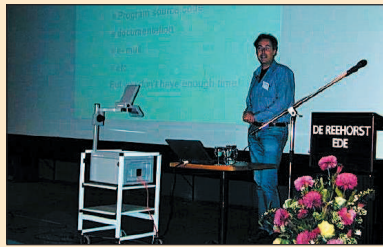


1999-es LinuxWorld Expón a kaliforniai San Joséban átveszi az IDG Open Source Computing díjat. A kérdés: ki az a két kisgyerek a színpadon?

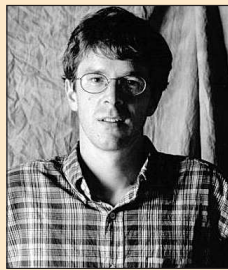
2. Ki az a híres kitalált személy, aki a képen látható Metacortex nevű cégnél dolgozik?



5. Nézd meg jól a képet! Ha elég ügyes vagy, ki tudod találni, ki is szerepel rajta.



6. Másoddiplomás hallgatóként írt egy kísérleti programot, mely saját magát terjesztette.



A programot az MIT-ből indította útjára, hogy ne derüljön ki, hogy Cornellben fejlesztette. Az évkönyvek a programot alkotója után nevezték el.

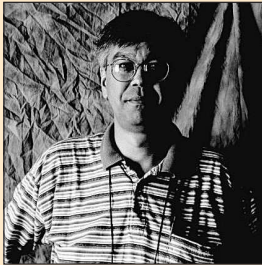
Melyik programról van szó, és ki az alkotó?

3. E híres személy neve matematikai eredetű.



Sokszor említik együtt a BugTraqkel, az átmeneti tár túlcsondulásáról írt könyvére pedig klasszikusként hivatkoznak. Hogy hívják?

4. A Hollandiában található Groningeni Egyetemen dokto-

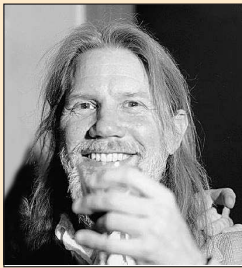


rált. Tizenkét évet töltött az Eindhoveni Egyetem Számítástudományi tanszékén, ahol rendszermérnökként dolgozott. Most az IBM Thomas J. Watson kutatóközpontjában tevékenykedik, és ha alkalma kínálkozna rá, mindent „beburkolna”. Nevezzük meg őt!

7. Valódi neve *Peiter Zatko*, de más elnevezéseken is ismert. Azt beszéljük, hogy a 2000. februári, Clintonnal való találkozására előtt egyetlen képet sem lehetett róla találni.



8. *David Kahn* (The Codebreakers) a reneszánsz óta azt tartotta a legforradalmibb



gondolatnak, amin a kérdéses személy dolgozott. A munka, amiben egy stanfordi tanárral együtt tevékenykedett, megjelent az IEEE Transaction on Information Theoryben, és így kezdődik: „Forradalmi változások előtt állunk a titkosírás területén”. Ki ez a személy?

9. Minden időkegyik legkedveltebb játéka, a PacMannek eredetileg más volt a neve. Mi volt az, és miért hívták úgy?



- Válaszok**
1. A díjat Linus Torvalds adta át és ő hozta magával a gyermekeit.
 2. Neo (teljes nevén Thomas Anderson) a Mátixből.
 3. *Elias Levy*, más néven *Alph One*. BugTraq-felügyelő (volt), és ő a szerzője a „Smashing the Stack for Fun and Profit” című írásnak.
 4. *Wietse Venema*.
 5. *Bram Moolenaar*, a Vim kifejlesztője.
 6. *Robert T. Morris*, aki a Morris féregről ismert.
 7. *Mudge of L0pht*, pontosabban @stake.
 8. *Whitfield Diffie*, (a „Diffie-Hellman” párból). A korszerű nyilvános kulcsú titkosíráshoz tartozó egyik alapítójaként ismert. Ő az NSA az állítja, ők már évekkel korábban ismerték a nyilvános kulcsú titkosítást.
 9. A PacMan név a japán „paku-paku” szlengetéjezésből származik, annyit tesz: enni. A játékot Japánban eredetileg PuckMannek hívták, de az amerikai hangzású) megváltoztatatták a névet, ugyanis a Namco attól tartott, hogy így az amerikai szlók nem veszik meg a gyerekeknek.

Sumith Dhar –
Linux Journal
2002. augusztus,
100. szám

A hónap szakmai tanácsai



Nincs állandó IP-cím, de SMTP-t szeretnék használni

Lehetséges internetes levélkiszolgálót működtetni betárcsázós kapcsolaton keresztül? Tudom, hogy a ppp0 IP-címe minden kapcsolódás alkalmával más és más. Az MX-bejegyzésnek a levélkiszolgálóra kell mutatnia, és a szolgáltatómnál a kiszolgáló IP-címét is meg kell adni, bár az a következő kapcsolódáskor nem lesz érvényes. Helyben kell DNS-t futtatnom, vagy használhatom a szolgáltató DNS-ét? Nemrég olvastam *Marcel Gagné* írását az otthoni és kirodai levélkiszolgálókról, mégis úgy érzem, nem állt össze teljesen a kép. *Daryl E. Murray*, daryl@Planet4us.net

Betárcsázós kapcsolaton keresztül is lehetséges SMTP-kiszolgálót futtatni, és úgy állítani be a DNS-t, hogy minden IP-címváltáskor frissüljön, de hidd el, nem ezt akarod. Ha nem lehet UUCP-t használni, ami a te esetben helyes megoldás lenne a levelek átírányítására, használj Fetchmailt a levelek letöltésére (az UUCP elég régi és a rendszergazdák már nem nagyon ismerik, valószínűleg internetszolgáltatód sem támogatja). Ha több postafiók leveleit kell letöltened, használj szolgáltatód spoolját, ahol minden neked szóló levél egy levelesládában van; ezt szedd le a Fetchmaillel, és válogasd szét az *Envelope-To*: mező alapján vagy aszerint, amit a szolgáltatód az eredeti *Envelope-To* tárolására használ. *Marc Merlin*, marc@merlins.org

A Fetchmailt csak akkor tudod leveleid letöltésére használni, ha szolgáltatód minden levélnél az *Envelope-To*: fejlécsort használja. Olvasd el az erről szóló figyelmeztetést a <http://www.tuxedo.org/~esr/fetchmail/fetchmail-man.html#25> címen.

Ha betárcsázós kapcsolathoz állandó IP-cím jár, és szolgáltatód hajlandó a leveleidet ideiglenesen tárolni, a kapcsolat felépülése után az SMTP *ETRN* parancsát kell kiadnod. A Sendmailben létezik hozzá egy segédprogram. *Don Marti*, dmarti@ssc.com

Az újonnan telepített Linux nem indul

Megpróbáltam telepíteni a Red Hat 6.2-t az egyik gépemre. A telepítés rendben lezajlott. Eltávolítottam a telepítőlemezt és újraindítottam a gépet. Ezután folyton azt a hibáüzenetet kapom, hogy nem rendszerlemez van a meghajtóban, és cseréljem ki. *Tim Dreas*, timdreas_@hotmail.com

Úgy tűnik, hogy sem a LILO, sem a GRUB rendszerbetöltő program nem települt megfelelően. Ha a telepítés során készítettél vészindító lemezt (természetesen készítettél, nem igaz?), akkor annak segítségével elindíthatod a gépet. Amikor a Linux parancsértelmező beviteli sora (#) megjelenik, írd be a `lilo -v` parancsot, amely a rendszerbetöltőt megpróbálja a merevlemezre írni. *Felipe E. Barousse Boué*, fbarousse@piensa.com

Az elveszett DDS-3 szalag fosztogatói

HP9000-es gépen, Linux-kiszolgálón (Red Hat 7.2-t – 2.4.9.31SMP rendszermag – futtat) egy HP-UX 11 alatt írt DDS-3 szalagot próbálok beolvasni. A Linux-kiszolgálóban a `/dev/st0` eszközön egy Sony SCSI szalagos egység van. A szalag egy nagy Oracle adatbázis export-állományait tartalmazza. A HP gép többé nem érhető el, feladatomban Linuxon újraépíteni az adatbázist. Olvasásnál akár `tar -tvf /dev/st0-t`, akár `tar -xvf /dev/st0-t` használok, mindig hibáüzenetet kapok: *Input/Output error at the beginning of the tape, error is not recoverable, exiting now*. Miért? *Adrian Manship*, adrian.manship@skynet.be

Két szalagos meghajtóval kapcsolatos dolgot kell ellenőrizned. Először is a DDS-meghajtók a vas szintjén támogatják a tömörítést, talán ez be volt állítva a HP-UX-nál, de a Linuxnál nem. Ha ez a beállítás nem egyezik, gondok lehetnek a szalag beolvasásával.

Másodsor, bár a `tar` eléggé elterjedt alkalmazás, talán mégsem `tar`-ral készítették a szalagot. A szalag létrehozására használt Oracle program talán a saját belső formátumában írja és olvassa a mentéseket. Az állomány egy kis darabjából állapítsd meg, hogy tényleg `tar`-ról van-e szó. Add ki a

```
dd if=/dev/st0 of=/tmp/tape bs=512
count=16 parancsot – a bs után meghajtódnak megfelelő blokkméretet írva, és akkora count értéket használva, amely a gyors betekintéshez már elegendő számú blokkot eredményez. A példa 8 Kb-et olvas be a szalagról. Lehet, hogy maga a szalag hibásodott meg. Ha ez a helyzet, használj hexaszerkesztőt vagy olyan szövegszerkesztőt, amelyik az ASCII karaktertábla felső felében elhelyezkedő elemektől sem akad ki, és vizsgálj meg a szalag tartalmát. Az általános formátum szerint minden állománybejegyzés az állomány nevével kezdődik, ezt követi a fejléc mindenféle adattal (például méret és jogosultságok), majd maga az állomány következik. Chad Robinson, crobison@rfgonline.com
```

A legegyszerűbb az lenne, ha vennél, kölcsönkérnél, kerítenél egy ugyanolyan gépet, és azon állítanád vissza az adatokat.

Christopher Wingert, cwingert@qualcomm.com

Egykor én is DDS-szalagokat használtam HP-UX alatt, és a legtöbb esetben a `cpio` segédprogram segítségével készítettem mentéseket. Ne feledd, hogy a `cpio` bonyolult alkalmazás sok kapcsolóval, ezért olvasd el a súgóoldalát (`man cpio`), és játssz el kicsit a beállításokkal! Légy óvatos, és véletlenül se írd felül a szalag tartalmát; a legjobb, ha fizikailag írásvédetté teszed. A következő címen egy bevezető jellegű `cpio` leírást találsz:

http://www.lns.cornell.edu/public/COMP/info/cpio/cpio_2.html.

Felipe E. Barousse Boué, fbarousse@piensa.com

Linux Journal 2002. augusztus, 100. szám

A Linux Journal honlapján számtalan gond megoldáshoz találhattok további segítséget. A Sunsite tüköroldalait, a gyakran feltett kérdéseket és az egyéb útmutatásokat a www.linuxjournal.com honlapon olvashatjátok el.

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

További kérdéseiteket szívesen fogadják (angol nyelven) a

www.linuxjournal.com/lj-issues/techsup.html címen, ahol csak egy kérdőívet kell kitöltenetek, de a bts@ssc.com címre levelet is írhattok. A levél tárgyában szerepeljen a „BTS” kulcsszó.

Új termékek

Linux USB 2.0 meghajtó

A Cypress Semiconductor kiadta az SL811 HS beágyazott közvezérlő-jéhez való USB meghajtóprogramot, ennek segítségével a fejlesztők beágyazott alkalmazásait – például a mobiltelefonokat, a PDA-kat és a hálózati eszközöket – USB-képességekkel ruházzák fel. Mivel az SL811HS vezérlő memóriába leképezett eszköz, a meghajtó lehetővé teszi, hogy a Linux-gazdaverem az USB-támogatást anélkül nyújtsa, hogy tudatában lenne a nem OHCI/UHCI-típusú vezérlő jelenlétének. Az SL811HS gazdavezérlő két szerepet ellátó kapuval rendelkezik, amely USB-gazdaként és USB-eszközként is működhet, továbbá támogatja a gyors és az alacsony sebességű USB-eszközöket. A meghajtó-program forráskódja a következő címről tölthető le:

➔ <http://www.cypress.com/press/linux>.

Adatok: [Cypress Semiconductor Corporation](http://www.cypress.com),

➔ <http://www.cypress.com>

Tetra Platform

Az Equator Technologies Tetra elnevezésű gépe kis helyigényű, könnyen bővíthető, így alapul szolgálhat egyéb eszközök, mint például értelmes



IP-kamerák, digitális képmagnók vagy IP-alapú internetes videotovábbító készülékek gyors fejlesztéséhez. A választható elemek között megtalálható a Tetra processzorkártya, egy nyílt eszközcsatló felület, amely további eszközök csatlakoztatását teszi lehetővé. A minden processzorkártyán megtalálható közös alkatrészek, például a BSP lapkacsatló, a memóriarendszer és a hálózati csatló elférnek a 2,75 hüvelyk × 4 hüvelyk méretű eszközön. A termék 64 MB SDRAM-mal és 4 MB flash-memóriával rendelkezik, továbbá analóg hangkimenetet, S/P-DIF hang ki- és bemenetet és 10/100Base-T ethernetet tartalmaz. Támogatja a Linux- és a VxWorks-rendszereket.

Adatok: [Equator Technologies, Inc.](http://www.equator.com),
e-mail: sales@equator.com,
➔ <http://www.equator.com>

BladeRack RS-1100V

A RackSaver mai ajánlata a kiszolgálók piacán: a BladeRack (szabadalmi oltalom alatt). A BladeRack akár 66 RS-1100RV RackBlade-et tartalmazhat, amelyek darabonként legfeljebb két 2,4 GHz-es Xeon processzornak adhatnak helyet. Ez egy 210 cm magasságú hordozható házban összesen 132 processzort jelent. AMD-alapú processzorok használata is lehetséges. A BladeRack 66 GB/s DDR ECC PC-2100 Corsair memóriát tartalmazhat Supermicro's P4DPR-6GM+ alaplapon. Mindezeket felül a BladeRack könnyen méretezhető és elérhető, a ventilátorok, hálózati alegységek és a kiszolgálók maguk is menet közben cserélhetők. A mindenhol kapható, szabványos elemeket használó BladeRack a számításgényes kutatásokat végző központokat és egyetemeket, valamint a 3D-s alkalmazásokat, és más nagy mennyiségű adatkezeléssel járó feladatokat futtató felhasználókat célozza meg.

Adatok: [RackSaver, Inc.](http://www.racksaver.com),

e-mail: inquiries@racksaver.com,

➔ <http://www.racksaver.com>

Xcelerix Embedded IMDB

A Xcelerix bejelentette a cég hivatalos megalakulását és a korábban ERDB néven ismert Xcelerix In-Memory Database megjelenését. A Xcelerix IMDB rendszere előre meghatározott lekérdezések alapján összeköti a megfelelő adatbázis táblákat, és bináris fát használ az indexeléshez. Ez lehetővé teszi a szűk keresztmetszetek felszámolását, így a rendszer másodpercenként akár egymillió tranzakció kezelésére képes (SQL mérési eredmény). A Xcelerix IMDB a memóriában tárolt adatok 20:1 arányú tömörítése a Xcelerix IMDB-t alkalmassá teszik beágyazott, válaszügyre érzékeny vagy valós idejű rendszerekben való felhasználásra. A termék több 32- és 64-bites rendszerre elérhető. A Xcelerix IMDB az SQL, a TCP/IP, az IEEE 802.3 és a POSIX mellett lemezalapú adatbázisrendszerekkel is egyesíthető.

Adatok: [Xcelerix Corporation](http://www.xcelerix.com),

➔ <http://www.xcelerix.com>

Eclipse Database Center

A Linux NetworX bejelentette az Eclipse Database Clustert. A termék az Oracle9i Real Application Cluster rendszeren alapul. A felhasználásra kész rendszer méretezhető, megbízható és magas elérhetőséget biztosít. Alkalmos adattárház üzemeltetésére, online tranzakciókezelésre és adatkezelő alkalmazások számára. A kiszolgálótelep több Xeon processzorra épül, Dot Hill adattárolókat használ, a csomópontok közti összeköttetés nagysebességű Gigabit Fiber Channelen valósul meg. Az Eclipse egybeépül az ICE telepkezelő eszközökkel, így távolról nyomon követhető a rendszer áramellátásának állapota, mérhető és lekérdezhető a hőmérséklete, valamint valós időben az egész rendszer figyelemmel kísérhető.

Adatok: [Linux NetworX](http://www.linuxnetworx.com),

➔ <http://www.linuxnetworx.com>

Cobalt RaQ 550

A Sun Cobalt RaQ 550 egy egység magas, szekrénybe szerelhető kiszolgálókészülék, amelyen az előre telepített programok és eszközök sokasága azonnal lehetővé teszi az e-mail, webkiszolgáló vagy más internetes szolgáltatás azonnali üzembe helyezését. Az 1,26 GHz-es processzorra és akár 2 GB memóriára épülő, 80 GB RAID 0-ra és 1-re képes merevlemezrel rendelkező készüléket olyan kis- és közepes méretű vállalkozások számára tervezték, amelyek internetes szolgáltatásairól saját maguk szeretnének gondoskodni. Az előre telepített programok között megtalálható az Apache webkiszolgáló, az Apache Tomcat, a Sendmail, a Bind DNS-kiszolgáló, a JSP, az SNMP-ügynök, és a Legato NetWorker, valamint Knox Arkeia mentőügylek.

Adatok: [Sun Microsystems, Inc.](http://www.sun.com/cobalt),

➔ <http://www.sun.com/cobalt>



Linux Journal 2002. augusztus

Köznép-Hollywood 2:0

A szórakoztatóipar és a technológia küzdelmének kezdete messzire nyúlik vissza. 1984-ben *Jack Valenti*, az MPAA elnöke így nyilatkozott: „A videomagnó olyan az amerikai filmgyártók és az amerikai közönség számára, mint a bostoni fojtogató a házban egyedül tartózkodó nőnek.” A jelenlegi bizalmatlanság gyökerét azonban egészen 1908-ig vissza lehet vezetni, amikor is a zenei kiadók azt állították, hogy a gépzongorák tekercei megsértik a zenei szerzői jogokat. A pert ugyan elvesztették a Legfelsőbb Bíróságon, de az ipar rábírta a Kongresszust, hogy vezessék be az úgynevezett mechanikus engedélyt, amely megszabott szabadalmi díj ellenében engedélyezte a megjelentetett zeneművek másolását.

1931-ben néhány zeneszerző pert indított, mert szerintük az egyik szálloda rádióállomása megsértette szerzői jogukat, amikor lejátszotta az általuk írt zenét.

A zeneszerzők maradtak alul. Az 1960-as években a könyvkiadók elvesztették a fénymásolókkal szemben folytatott

pert, de a Legfelsőbb Bíróság döntése lehetővé tette a kiadott művek

„szerzői jogilag megengedett felhasználását”.

A hatvanas évek végén és a hetvenes évek elején a műsorszórók a Community Antenna Television (CATV) kábeltelevíziót támadták (a moziipar szintén a pereskedők között volt, és a színházi alkalmazottakat is rávették, hogy „Harcolj a fizetős tévé ellen!” feliratú kítűzőt viseljének). A Legfelsőbb Bíróság döntése a két ügyben megnyitotta az utat a mai kábeltelevíziózás elért. A fentiekből látható, hogy ez a háború már igen csak régóta tart.

Öröm az örömben, hogy a technológia két újabb győzelmet aratott azok felett, akik irányítani akarják: az egyiket a Creative Commons elindítása jelenti, a másikat pedig az, hogy az amerikai Kongresszusi Könyvtár vezetője visszautasította a Copyright Arbitration Royalty Panel (CARP) javaslatát, amely szigorú feltételeket és magas jogdíjat szabott volna ki az internetes rádióállomásokra. A Creative Commons (<http://www.creativecommons.org>) szervezet az O'Reilly Emerging Technologies tanácskozásán alakult meg a kaliforniai Santa Clara városában, május 16-án, több hónapon át tartó, csendes előkészületek eredményeképpen. Vezetője, *Lawrence Lessig* a stanfordi jogi egyetem professzora, aki két nagy jelentőségű mű (Code and Other Laws of Cyberspace, The Future of Ideas) szerzője. Mindkét írásában elmarasztalja a szórakoztatóipart, amiért folyamatosan a saját érdekeik szerint próbálják alakítani a szerzői jogi törvényt, és az Internet kialakult szokásai helyett egy szigorúan szabályozott rendszert akarnak bevezetni a „tartalmak” ter-

jesztésének korlátozása céljából.

A szervezet a Stanford Egyetemen található, ahol szoros együttműködik az egyetem internetes és társadalmi központjával (Center for Internet and Society). Ezenkívül „nagylelkű támogatást” kap a Center for the Public Domain szervezettől. Ez utóbbi korábban a Red Hat Center nevet viselte, és a Duke Egyetem ad neki ott-hont. Elnöke *Bob Young*, a Red Hat alapítója.

A Creative Commons nem lobbizik a szórakoztatóipar ellen, hanem kézzelfogható megoldásokat dolgoz ki a szellemi termékek eredeti forrása, azaz a művészek számára. A tanácskozáson tartott beszédében Lessig a következőket mondta:

„A törvényben említett tartalom az enyém, tehát nekem kellene megszabni, hogy milyen feltételek mellett teszem elérhetővé. A szerző szándékait a tartalom természetét illetően a számítógép által olvasható formában kell megadni. Nem kellene megosztani a világot a felügyelet és az elérhetőség hívei között. Azoknak is megoldást kell találni, akik – egyéni szinten – mindkettőt pártolják.” A Creative Commons készülődése közben komoly veszély fenyegette az internetes rádiózást, amely több ezer állomást számlál (sokan közülük ügyesen Linuxot és más nyílt forrású programokat használnak). A 2002 februárjában benyújtott CARP-javaslat nyomán az adók többsége valószínűleg csődbe ment volna. Amennyiben elfogadták volna, a CARP 2002. május 22-től vált volna kötelező érvényűvé. Május 21-én azonban a Kongresszusi Könyvtár vezetője kiadta „a testület határozatát elutasító rendeletet”, amely egy hónap múlva véglegessé vált. A rádióállomások és az őket támogató csoportok, például a SavelInternetRadio.org megkönnyebbülten lélegeztek fel.

Ezzel az ügy azonban még távolról sincs megoldva. A CARP a digitális szerzői jogi törvény (Digital Millennium Copyright Act) származéka, amely változatlanul érvényben van. Ennél is fontosabb, hogy a Commons alapötletét sem értik kellőképpen – különösen kormánykörökben nem, ahol a szórakoztatóipar továbbra is hatalmas befolyással bír.

Ismét Larry Lessiget idézem: „Csak akkor juttathatjuk érvényre ennek a területnek a fontosságát, ha a hétköznapi emberek is megértik. Ez pedig mindaddig nem következik be, amíg a szakértők el nem magyarázzák a politikusoknak, hogy a technológiába beépített értékek milyen jelentősek a szabadság és az alkotóerő szempontjából.”

Linux Journal 2002. augusztus, 100. szám



Doc Searls

(doc@ssc.com) a Linux Journal szerkesztője és a Cluetrain Manifesto társszerzője.

Méltányos használat

A szerzői jogi törvény (17 U.S.C. § 107) méltányos használatról szóló elvének meghatározása nehezebb, mint gondolnánk.

Időről időre olyan esetekkel találkozom, amikor valaki a *méltányos használat* (fair use) elvére hivatkozva sérti meg valaki másnak a szerzői jogait. A „méltányos” szó hétköznapi értelemben sajnos olyan jelentésekkel is bír, amelyek különböznek a „méltányos használat” kifejezés jogi értelmétől.

A törvény nem rendelkezik arról, hogy azokat a felhasználói engedélyeket, amelyeket visszataszítónak találunk vagy erkölcsileg elítélünk, figyelmen kívül hagyassunk, amennyiben ez méltányos. Egy bíróság a következőképpen írta le a méltányos használat elvének alkotmányos alapját:

A méltányos használat kiváltságának alapvető igazolása abban az alkotmányos szándékban fedezhető föl, amely a szerzői jogi védelmet egyáltalán lehetővé teszi, tudniillik „a tudomány és a hasznos művészetek haladásának támogatására” (USA Alkotmány, I. Cikk, 8§/8)... ennek a célnak a megvalósítása érdekében „a szerzői jog megsértésének ügyében eljáró bíróságoknak időnként a szerzői jog birtokosának legmagasabb anyagi megtérüléshez fűződő érdekét alá kell rendelnie annak a magasabb rendű közérdeknek, ami a tudományok, a művészet és az ipar előremozdításához fűződik” – Rosemont Enters. kontra Random House, Inc., 366 F.2d 303 (2d Cir. 1966), cert. denied, 385 U.S. 1009 (1976).

E szerint a megfogalmazás szerint a méltányos használat *kiváltság*, nem csupán védekezés. Azt a jogunkat, hogy szerzői joggal védett műveket bizonyos célokra felhasználjunk, az alkotmánynak ugyanaz a mondata biztosítja, amely egy szerző számára lehetővé teszi, hogy művét szerzői jogi védelem alá helyezze.

A méltányos használat elvét a kongresszus törvénybe iktatta (Copyright Act, 17 U.S.C. § 107), hogy világossá tegye, mely felhasználási módok tekinthetők méltányosnak. A törvény a következő felsorolást tartalmazza:

„olyan célokra, mint a műbírálát, a magyarázat, a híradás, az oktatás (beleértve tantermi használatra több másolat készítését), a tudományos célok vagy a kutatás”.

Amikor az „olyan célokra, mint” kifejezést látjuk egy jogszabályban, vitára készülünk fel. A törvényhozók nyilvánvalóan nem sorolhatták föl az összes lehetséges célt, tehát a kérdés az, hogy milyen további célok felelhetnek meg a mintának? A kongresszus négy szempontot adott meg annak megállapítására, hogy a felhasználás az adott esetben méltányosnak tekinthető-e:

1. A felhasználás jellege és célja, ideértve azt is, hogy a felhasználás üzleti jellegű vagy nonprofit oktatási célú. A híradás, a tudományos kutatás és az oktatás például az előnyben részesített méltányos felhasználás körébe tartozik. Az üzleti célú felhasználás nem támogatott.
2. A szerzői jog által védett mű jellege. A törvény általában szükségesebbnek tartja a tényeken alapuló

művek, mint a szépirodalom terjesztését. Amennyiben a nyelvi megfogalmazás másolását a mögöttes tényanyag terjesztése érdekében engedélyezni kell, a másolás könnyebben igazolható.

3. A felhasznált rész mennyisége és fontossága a mű egészéhez viszonyítva. A méltányos használat kevésbé helytálló, ha egész műveket vagy legértékesebb részeit másoljuk.
4. A használat hatása a mű feltételezhető piacára vagy értékére. Ha a mű másolása megakadályozza a szerzői jog tulajdonosát abban, hogy haszonra tegyen szert belőle, még abban az esetben is, ha csupán a lehetséges haszonról van szó, a másolás kevésbé igazolható. Ez a szempont a méltányos használat oldalára billentheti a mérleget olyan esetekben, amikor a műnek nincsenek más ismert példányai, amikor a tulajdonos személye nem azonosítható, vagy amikor nincs olyan valós piac, amely a mű értékesítését lehetővé tenné.

Önmagában e szempontok egyike sem meghatározó, a bíróságoknak „a különböző érdekek figyelmes mérlegelésével” kell elemeznie őket (Financial Information, Inc. kontra Moody's Inv. Serv., 751 F.2d 501 (2d Cir. 1984)). Ehhez hasonlóan, amikor valakinek meg akarjuk sérteni a szerzői jogait, az „érdekek figyelmes mérlegelését” magunknak is el kell végeznünk. Fontoljuk meg, hogy célunk elég hasonló-e a törvényben felsorolt célok egyikéhez, és vegyük sorra a szempontokat, megvizsgálva, hogy az adott szempont kedvez-e nekünk. Ha nem győződünk meg róla, hogy átjutottunk a méltányos használat rostáján, ne sértsük meg a szerzői jogokat. Amennyiben nyílt forrású programokat használunk, mint amilyen a Linux, a méltányos használat kérdése általában nem merül föl. A nyílt forrású programok terjesztési engedélye mindenkinek mindenhol mindenféle elképzelhető célra biztosítja a program másolásának, módosításának, terjesztésének (eladásának vagy ingyenes továbbadásának) a jogát, valamint a hozzáférést a programkódhoz, amely mindezt lehetővé teszi. A szerzői jog birtokosa mindenkit felhatalmaz, így nem szükséges a méltányos használat elvére hivatkoznunk. Megint egy szempont, amely szerint a szabad és nyílt kódú programok jobbak a jogdíjas programoknál. Jogdíjas program esetében győződjünk meg róla, hogy jó érveink vannak a felhasználás méltányosságára mellett, ha bármi olyat teszünk, amit a felhasználói engedély nem tesz lehetővé. A szabad és nyílt forrású programok esetében élvezzük a teljes körű méltányos felhasználási jogot.

Linux Journal 2002. augusztus, 100. szám



Lawrence Rosen

(www.rosenlav.com) magángyakorlatot folytató jogász. A Nyílt Forrás Kezdeményezés ügyvezető igazgatója és jogtanácsosa (www.opensource.org).



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

A 100. szám

A dotcomkorszak előtt nem létezett a „Linux-cég” fogalma; később viszont nehéz volt olyan nagy számítástechnikai céget találni, amelyik ne használt volna Linuxot, vagy a kínálatában ne szerepeltek volna Linux-alapú termékek.

Manapság a Wall Street Journal leghangsúlyosabb szava a Linux. Holtversenyben áll a Unixszal, Windowszal, a Checkbooksszal, a Rocksszal és a DB2-vel. Az újság piaccal foglalkozó részében mind együtt szerepelnek az első olyan oldalon, amely a reklámoknak van fenntartva. Nem túlzok, ha azt állítom, hogy a Linux sikertermék lett a rendszerszintű megoldásokat szállító cégek berkeiben, és a folyamat itt nem ér véget. Egy évvel ezelőtt a Linux csak a webkiszolgálók területén volt sikeres, ám most terjeszkedik. A HP szerint legutóbbi eladási mutatóik azt jelzik, hogy a Linux növekvő jelenléttel bír a pénzügyi szolgáltatások piacán. A Windows még mindig uralkodó helyzetben van, de vajon meddig? Néhány éve még maga a kérdés is nevenséges lett volna! Bár a Microsoft háborút látszik hirdetni a Linux és a nyílt forrású programok ellen, a fegyver visszafelé sül el. A cég állítólag megpróbálta meggyőzni az Egyesült Államok Nemzetvédelmi Minisztériumát, hogy a nyílt forrású programok mind a biztonságra, mind pedig a cég szellemi tulajdonára veszélyt jelentenek. Eközben a Védelmi Minisztérium a MITRE Társasággal jelentést készíttetett, amelyben 249 olyan esetet határoznak meg, amikor nyílt forrású eszközöket lehet használni. Ilyen többek között a Hírszerzési Osztály webportálja, vagy az USA hadseregében és légierijében használt hálózati biztonsági programok. Ahol a Linux mint operációs rendszer nem jöhet számításba, új ajtókat nyit más ingyenes és nyílt Unix-változatoknak, például a BSD-nek, amelyet beépítettek a Darwinba, az Apple OS X nyílt forrású kezdeményezésébe. Ezek az események nagymértékben növelték a világ Linux-barát PC-inek és kiszolgálóinak számát.

1993-ban *Phil Hughes* felvett engem egy levelezőlistára, amely egy ingyenes programokkal foglalkozó magazin lehetőségeit derítette fel. Miután egy ideig csak ötleteket ütköztettünk, Phil hirtelen bejelentette, hogy saját kis cége, az SSC útjára indít egy lapot, ami egy 21 éves finn srác szellemi gyermekével, a Linuxszal foglalkozik. Azt hittem, elment az esze. De Phil ösztönösen ráérez olyan dolgokra, amelyeket mások nem is látnak. Néhány évvel később olyan KDE alatt futó irodai alkalmazásokat mutatott nekem, amelyek kísértetiesen hasonlítottak arra a látványra, amely Windows alatt fogadja az embert – azzal a különbséggel, hogy ingyenes volt és nyitott. Annyira elbűvölt, hogy kaptam az ajánlatán és csatlakoztam a magazinhoz. Ez 1999 elején történt, akkortájt a tőkebefektetések még nagy lendülettel folytak, és a Linuxnak a befektetőkre gyakorolt erőteljes hatása a Viagrához volt mérhető. Néhány hónapig nem volt hiány Linux-milliárdosokban.

Nemrégiben nagytakarítást végeztem üzleti kártyáim között, de mielőtt kidobtam volna őket, helyzetképet készítettem. Érdekes történet bontakozott ki:

- Néhány cég felhagyott az üzleti tevékenységgel. Ebben a cipőben járt a Linux Laptops, a Rebel.com, a Loki, az Eazel és OpenSales, amely az Xelerate nevet vette fel, mielőtt megszűnt.

- További cégek, például a Kerbango és a Cobalt nagyobb cégek tulajdonába kerültek – ezeknél a 3Com és a Sun volt a felvásárló.
- A Linux kikerült mások tevékenységének középpontjából. A VA Linux cég a nevét VA Software-re változtatta, pedig egykor olyannyira összefonódott ezzel az L betűs szóval, hogy hétszámjegyű összeget fizetett a Linux.com tartománynévéért.
- A Corel, miután megpróbált a Linux köré a felhasználókra épülő üzletet teremteni, megszabadult a Linuxtól és egyéb nehezekektől, hogy a felszínen tudjon maradni.
- A Linuxcare még él, bár az alapítói nélkül, akik a Sputnik névvel kerültek be a hírek közé – ez egy új cég a mobil vezeték nélküli hálózatok piacán.
- Más cégek – például a Boxx (nem azonos a texasi Boxx Technologies-zel), ami Linux/Windows hibrid hordozható gépet készített (a prototípusig eljutott) – egyszerűen eltűntek.
- Két, egymással kapcsolatban álló piaci kísérlet, a CoSource és a SourceExchange érdeklődés hiányában csendben megszűnt.
- Természetesen sikertörténetek is léteznek. A Linux-alapú TiVo a nagy érdeklődést kiváltó digitális videokamera-üzlet rokon értelmű kifejezése lett. A Borland, megszabadulva a felejthető Inprise névtől, jó üzletet csinált többfelületen futó linuxos fejlesztő eszközzel, a Kylixszal.
- A SuSE ugyan csökkentette működési területeit, Európában továbbra is vezető a különböző Linux-változatok terén, és világviszonylatban is erős. A Caldera elég jól tartja magát, csakúgy, mint a Turbolinux. A nagy győztes egy olyan cég, amelytől nincs üzleti kártyám: a Red Hat, mely továbbra is büszkén lengeti a pingvines zászlót, és a messze legnépszerűbb Linux-változat készítője.

Mi itt, a Linux Journalnál együtt vészeltük át a nehéz időköt a talpon maradó linuxos cégekkel. Ami tovább lökött minket, az a Linux folyamatos menetelése volt afelé, amit *Linus Torvalds* félig viccesen „világuralomnak” nevezett.

A Linuxnak végre sikerült, amire a Unix-rajongók évtizedek óta vágytak: az operációs rendszer elterjedté vált. Ma az egyetlen komoly jövővel rendelkező Unix-rendszer ingyenes, nyitott és készen áll a fejlesztésre – bárki beugorhat és segíthet. Ezt nevezem én sikertörténetnek.

Linux Journal 2002. augusztus, 100. szám



Doc Searls

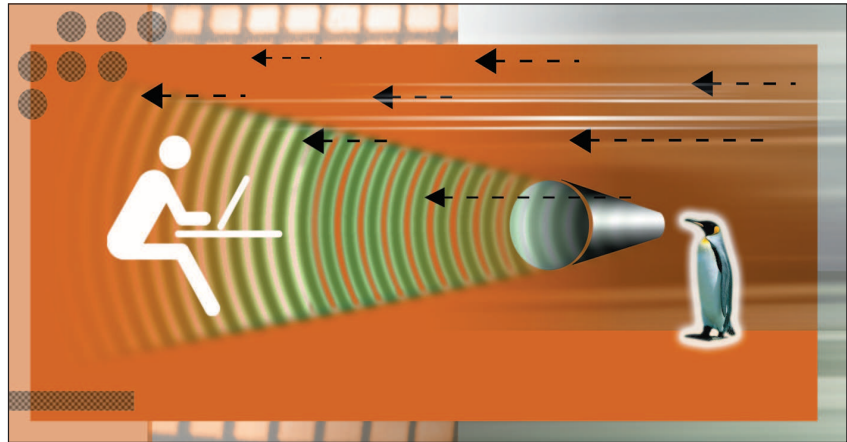
(doc@ssc.com) a Linux Journal szerkesztője és a Cluetrain Manifesto társszerzője.

Halló, beágyazottrendszer-fejlesztők!

Vegyétek meg ezt az újságot! Lapunkban most már egy olyan rész is megtalálható, amely a beágyazott rendszerek Linux alatti fejlesztésével foglalkozik, és részletes tájékoztatást tartalmaz az eszközvezérlők kódolásáról és más olyan témákról, amelyeket az asztali rendszerek fejlesztői is hasznosítani tudnak. Akiket elsősorban a beágyazott rendszerek fejlesztése érdekel, talán felteszik a kérdést, hogy miért kezdjék el most olvasni a Linux Journalt, amikor eleve tudják, hogy tele lesz nem beágyazott rendszerekről szóló témákkal is. És ugyan miért érdekelnék a hagyományos Linux-programok fejlesztőit a beágyazott rendszerekről szóló cikkek? Az, ami őket érdekli, amúgy sem fog egy linuxos órán futni, nem igaz?

A rövid válasz az, hogy ebben a hónapban indul útjára *Greg Kroah-Hartman* Beágyazott rendszerek című rovata az eszközvezérlőkről. Greg foglalkozik a Linux USB és PCI Hot Plug karbantartásával és támogatásával, és már régebben is nagyszerű cikkeket írt a Linux Journalba és az Embedded Linux Journalba. A Beágyazott rendszerek rovat kéthavonta jelenik meg, ami azt jelenti, hogy átlagosan havonta nyújtunk segítséget, és félhavonta hozunk le olyan rendszermagányosságokat, amelyeket sehol máshol nem találhattok meg. Most, miután már elküldted az előfizetési lapodat, eláruljuk a hosszú választ is, vagyis azt, hogy a Linux és általában a szabad programok természetüknél fogva beágyazhatóak. A beágyazható projektek eleve olyan programokon indulnak fejlődésnek, amelyek más rendszerre szabadon átvihetők, testreszabhatók és árusíthatók. Ha nem hiszed, figyelj meg a sokat dicsért, kifejezetten „média operációs rendszernek” szánt BeOS lenyűgöző sikerét (ah!), mely a legnagyobb csendben tűnt el a süllyesztőben, míg a valódi médiaeszközök – például a TiVo vagy a Moxi – Linuxra tértek át.

Amikor Gregnek az új Cypress USB lapkáról, az SL811HS-ről beszéltem, valamint arról, hogy a vezérlőjét a Cypress GPL alatt adja ki, az első reakciója az volt, hogy örül neki, hogy a Cypress érti „a csízót”. És valóban,



nagyon hasznos eszköznek is tűnik. Mesterként és szolgaként (master-slave) is képes működni, ami azt jelenti, hogy a gépedhez egy PDA-t csatlakoztathatsz, amellyel összhangba hozhatod az állományaidat, majd a PDA-hoz USB-eszközöket csatlakoztathatsz, például egeret vagy billentyűzetet. Próbáld csak ki! Ti, akik asztali gépként vagy kiszolgálóként működő Linuxra fejlesztetek, miért kényszerítitek állandó üldögélésre a felhasználókat, és arra, hogy kéztőcsatorna-szindrómától szenvedjenek, amikor kényelmes beágyazott eszközöket használhatnának ott is, ahol a valódi munka (vagy a szórakozás) zajlik? Vagy ami még jobb, miért nem hagyjuk ki a felhasználót teljesen az unalmas részekből, és engedjük, hogy a gép foglalkozzon a való világgal? Programjaitok hatékonyabb munkát végezzenek „beágyazott” rendszerekben, mint a kiszolgálószobában vagy az irodában. Ugyanez vonatkozik a web-lapokra is. Próbálgátok csak ki egy Zauruson még ma!

De térjünk vissza egy kicsit arra, mi is a csízó. A beágyazott Linux nem csupán egy befőliázott csomag, amelyre rá van írva, hogy „beágyazott”. Ez egy szabványos, általános felület, széles körű meghajtógyűjteménnyel, ami napról napra bővül (köszönjük a Cypressnek is), és amelyet olyan felhasználói szerződéssel adnak ki, ami engedi, hogy elvégezzük a munkát. A szokványos Linuxhoz hasonlóan a beágyazott rendszerek is jó programok a csízót értő emberek számára.

1. A beágyazott rendszereknek nincs szükségük rendszergazdákra.
2. Az alkalmazás már fel van rájuk telepítve. Nem kell további programokat telepíteni, hogy képes legyen a feladata elvégzésére.
3. Gyorsan válaszol. Sok beágyazott rendszernél alapkövetelmény a valós idejűség, de még ha nem is az, akkor sem kényszerítenek rá, hogy olyan ostobaságokra kelljen várnod, mint az fdisk.
4. Közvetlenül a való világgal áll kapcsolatban, nem pusztán a hálózattal vagy a kéztőcsatornát pusztító billentyűzettel és egerrel.
5. Nem lehet vele vitatkozni! Nem érez sajnálatot, bánatot vagy félelmet!
6. És soha, de soha nem áll le, amíg csak meg nem halsz!

Várjunk csak, az utolsó két pont a Terminátorra vonatkozik, és az összes beágyazott rendszerre nem áll. De a lényeg érthető – a beágyazott rendszerek már most olyanok, amilyenek a programok valóni akarnak, ha felnőnek, és a „hagyományos” Linux ugyanebbe az irányba tart.

Linux Journal 2002. augusztus, 100. szám



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

A tty-réteg

Az eszközmeghajtók fejlesztéséről szóló sorozatunk első részében azzal foglalkozunk, hogyan kezeli a rendszermag a rendszerkonzolt és a soros kapukat.

Köszöntöm az olvasót a „Driving Me Nuts” című új rovatban. Ebben a rovatban a Linux-rendszermag különböző eszközmeghajtó-alrendszereit vizsgáljuk, megpróbáljuk átlátni az általuk és az eszközmeghajtókon keresztül biztosított különböző felületek széles skáláját. Ha valaki azt szeretné, hogy egy adott alrendszert közelebbről megvizsgáljunk, kérem, küldjön levelet.

Számos nagyon jó referencia létezik a Linux-rendszermag és a Linux-eszközmeghajtók programozásának témakörében (lásd a *Kapcsolódó címeket*). Ez a rovat feltételezi, hogy legalább futólág ismerjük ezeket, vagy elérhető közelségben vannak olvasóink számára.

Vágjunk neki az útnak: először vizsgáljuk meg a rendszermag tty-rétegét. Ezt a réteget minden Linux-felhasználó igénybe veszi, amikor parancssorba ír, vagy a soros kapun keresztül lép kapcsolatba valamilyen eszközzel.

Minden tty-meghajtó számára létre kell hozni egy `struct tty_driver`-t, amely leírja saját magát, és a tty-réteg segítségével bejegyzi a felépítését. A `struct tty_driver` az `include/linux/tty_driver.h` állományban kerül megadásra. Az 1. lista mutatja, hogyan épül fel ez a szerkezet a 2.4.18 rendszermagváltozattól kezdve. A szerkezet mérete igen tekintélyes, úgyhogy próbáljuk meg kisebb részekre bontani. A `magic` mezőnek minden esetben a `TTY_DRIVER_MAGIC` értéket kell felvennie. Ezt a tty-réteg annak ellenőrzésére használja, hogy valóban tty-meghajtóról van szó. A `driver_name` és `name` mezők az eszközmeghajtó leírására használatosak. A `driver_name` mező tartalmát, ami a `/proc/tty/drivers` fájlban is megjelenik, valamilyen kifejező értékre kell beállítani. A `name` mező annak megadására szolgál, hogy milyen `/dev` vagy `devfs` név szolgál alapul a meghajtónkhoz. Például a rendszermag soros meghajtója a `driver_name` mező értékét `serial-ra`, a `name` mezőét pedig `ttys-re` állítja, ha a `devfs` nincs engedélyezve és `ttys/%d-re`, ha engedélyezve van. Amennyiben a `devfs` engedélyezett, a mezőt használja az eszközmeghajtó számára történő új eszközcsoport létrehozására. A `name` mező `%d` része az eszköz mellékszámával (minor number) kerül feltöltésre, amikor a tty-alrendszer számára megtörténik a bejegyzése.

A `name_base` mezőre csak akkor van szükség, ha az eszköz számozása nem a 0 mellékszámval kezdődik. Csaknem minden eszközmeghajtónál a 0 értéket kell beállítani. A `major`, `minor_start` és `num` mezők írják le, hogy a tty-rétegben eszközmeghajtóknhoz milyen fő-, illetve mellékszámok vannak rendelkezésre. A `major` mező tartalmának az eszközmeghajtóhoz rendelt főszám (major number) kell lennie. Amennyiben új eszközmeghajtót hozunk létre, olvassuk el a *Documentation/devices.txt* leírást arra vonatkozóan, hogyan biztosíthatunk eszközmeghajtónk számára új főszámot. Annak is hasznos a fájl elolvasása, aki arra kíváncsi, hogy az egyes eszközmeghajtók milyen fő- és mellékszám párokat használnak. A `minor_start` mező az eszköz által használt első mellékszám megadására szolgál. Ha eszközmeghajtónkhoz egy teljes főszám hozzá van

rendelve, annak értéke 0 kell legyen. A `num` mező értéke mutatja meg, hogy hány különböző mellékszámot rendeltünk az eszközmeghajtóhoz.

Vagyis amennyiben eszközmeghajtónk a 188-as főszámmal rendelkezik, az alábbi értékeket kell beállítani:

```
major: 188
      minor_start: 0
      num: 255
```

A `type` és `subtype` mezők írják le, hogy eszközmeghajtónk milyen típusú tty-meghajtóként szerepel a tty-réteg számára.

A `type` mező az alábbi értékeket veheti fel:

- `TTY_DRIVER_TYPE_SYSTEM`: a tty-alrendszer belső használatára van fenntartva annak közlésére, hogy belső tty-eszközmeghajtóról van szó. Ha ezt az értéket veszi fel, akkor a `subtype` értéke `SYSTEM_TYPE_TTY`, `SYSTEM_TYPE_CONSOLE`, `SYSTEM_TYPE_SYSCONS` vagy `SYSTEM_TYPE_SYSPTMX` lehet. Ezt a típust normál tty-eszközmeghajtó nem használhatja.
 - `TTY_DRIVER_TYPE_CONSOLE`: csak a konzolmeghajtó használhatja, másra ne használjuk!
 - `TTY_DRIVER_TYPE_SERIAL`: bármilyen soros eszköz meghajtója használhatja. Ez esetben a `subtype` értéke `SERIAL_TYPE_NORMAL` vagy `SERIAL_TYPE_CALLOUT`, a meghajtó típusától függően. A `type` mezőnek ez az egyik leggyakrabban használt beállítása.
 - `TTY_DRIVER_TYPE_PTY`: a pseudo terminal felület (pty) által használt beállítás. Ha ezt az értéket használjuk, a `subtype` a `PTY_TYPE_MASTER` vagy `PTY_TYPE_SLAVE` értéket veheti fel.
- Az `init_termios` mező a kezdeti időértékek (sorbeállítások, sebességek) beállítására szolgál az eszköz első létrehozásakor. A `flags` mező az eszközmeghajtó igényeitől függően az alábbi bitértékek eredőjét veszi fel:
- `TTY_DRIVER_INSTALLED`: a bit beállításakor az eszközmeghajtó nem tudja saját magát bejegyezni a tty-réteggel, úgyhogy ne használjuk.
 - `TTY_DRIVER_RESET_TERMIOS`: a bitet magasra állítva a tty-réteg nullázza az időértékeket, amikor az utolsó folyamat lezárja az eszközt. Konzol- és pty-meghajtók esetén hasznos.
 - `TTY_DRIVER_REAL_RAW`: e bit bekapcsolása azt jelzi, hogy az eszközmeghajtó biztosítja a párosságadat (parity) vagy a megszakításjelző karakterek jelentését a sorvezérlő felé, amennyiben a sorvezérlőt erre nem utasítjuk. Általában minden eszközmeghajtónál beállítjuk, ez lehetővé teszi a sorvezérlő hatékonyságának növelését.
 - `TTY_DRIVER_NO_DEVFS`: a bit magas értékénél a `tty_register_driver()` hívása nem hoz létre `devfs` bejegyzést. Bármilyen eszközmeghajtó esetén hasznos lehet, ha az eszköz fizikai jelenlététől függően a fő eszközt dinamikusan akarjuk létrehozni, illetve megsemmisíteni. Példák olyan meghajtókra, amelyek használják a bitet: USB soros meghajtók, USB modemmeghajtó és az USB Bluetooth tty-meghajtó.

1. lista A struct tty_driver

```

struct tty_driver {
    int magic;
    const char *driver_name;
    const char *name;
    int name_base;
    short major;
    short minor_start;
    short num;
    short type;
    short subtype;
    struct termios init_termios;
    int flags;
    int *refcount;
    struct proc_dir_entry *proc_entry;
    struct tty_driver *other;

    /*
     * Pointer to the tty data structures
     */
    struct tty_struct **table;
    struct termios **termios;
    struct termios **termios_locked;
    void *driver_state;

    /*
     * Interface routines from the upper tty
     * layer to the tty driver.
     */
    int (*open)(struct tty_struct *
        ↪tty, struct file * filp);
    void (*close)(struct tty_struct *
        ↪tty, struct file * filp);
    int (*write)(struct tty_struct * tty,
        ↪int from_user, const unsigned char
        ↪*buf, int count);
    void (*put_char)(struct tty_struct *tty,
        ↪unsigned char ch);
    void (*flush_chars)(struct tty_struct *tty);
    int (*write_room)(struct tty_struct *tty);
    int (*chars_in_buffer)(struct tty_struct
        ↪*tty);
    int (*ioctl)(struct tty_struct *tty,
        ↪struct file * file, unsigned int cmd,
        ↪unsigned long arg);
    void (*set_termios)(struct tty_struct
        ↪*tty, struct termios * old);
    void (*throttle)(struct tty_struct * tty);
    void (*unthrottle)(struct tty_struct * tty);
    void (*stop)(struct tty_struct *tty);
    void (*start)(struct tty_struct *tty);
    void (*hangup)(struct tty_struct *tty);
    void (*break_ctl)(struct tty_struct *tty,
        ↪int state);
    void (*flush_buffer)(struct tty_struct *tty);
    void (*set_ldisc)(struct tty_struct *tty);
    void (*wait_until_sent)(struct tty_struct
        ↪*tty, int timeout);
    void (*send_xchar)(struct tty_struct *tty,
        ↪char ch);
    int (*read_proc)(char *page, char **start,
        ↪off_t off, int count, int *eof, void *data);
    int (*write_proc)(struct file *file,
        ↪const char *buffer, unsigned long count,
        ↪void *data);

    /*
     * linked list pointers
     */
    struct tty_driver *next;
    struct tty_driver *prev;
};

```

A *refcount* mező egy a tty-meghajtóban lévő egészre mutató pointer típusú változó; a tty-réteg használja a meghajtó hivatkozásainak kezelésére, a tty-meghajtónak nem szabad használnia.

A *proc_entry* mező beállítását nem közvetlenül a tty-meghajtó végzi. Ha a tty-meghajtó *write_proc* vagy *read_proc* függvényt valósít meg, ez a mező tartalmazza a létrejövő meghajtó *proc_entry* mezőjét. A másik mezőt csak a pty-meghajtó használja, tty-meghajtók nem alkalmazhatják.

Van még néhány különböző tty-szerkezetre mutató pointer típusú változónk. Az egyik a *table* mező, ami egy *tty_struct*-mutatókból álló tömbre mutat. A *termios* és a *termios_locked* mezők egy *struct termios* mutatókból álló tömbre mutatnak. E tömbök mindegyikének annyi eleme van, amennyit a feljebb említett *minor* mezőben beállítottunk. Ezeket a tty-réteg használja a különböző mellékszámú eszközök megfelelő kezeléséhez, a tty-meghajtónak közvetlenül nem szabad módosítania. A *driver_state* mezőt csak a pty-meghajtó használja, más tty-meghajtónak nem szabad módosítania.

A *tty_driver* szerkezet a különböző függvénymutatók hosszú listáját tartalmazza. Ezeket a tty-réteg használja a tty-meghajtó meghívására, amikor valamilyen műveletet akar végrehajtani. Egy tty-meghajtónak nem kell az összes függvénymutatót megadni, ám amennyiben ezt a függvényt a tty-meghajtó végre-

hajtja, néhányat közülük kötelező jelleggel meg kell adnia. Az *open* függvényt a tty-réteg akkor hívja, amikor az *open* (2) azon az eszközcsomóponton meghívása kerül, amihez tty-meghajtónk rendelve van. A tty-réteg egy fájlmutatóval, valamint egy olyan mutatóval végzi a függvényhívást, amely az eszközhöz rendelt *tty_struct* szerkezetre mutat. Ezt a mezőt a tty-meghajtónak a helyes működéshez mindenképpen be kell állítania a helyes működéshez (ellenkező esetben a felhasználó az *open* (2) hívásakor -ENODEV üzenetet kap). A *close* függvényt a tty-réteg akkor hívja, amikor *release* (2) hívás érkezik a korábban az *open* (2) által létrehozott fájlmutatóra. Ez azt jelenti, hogy az eszközt le kell zárni. A *write* függvény hívását a tty-réteg akkor kezdeményezi, amikor a tty-eszköz felé adatot kell továbbítani. Az adat érkezik a felhasználó, illetve a rendszermag felől (a *from_user* bit beállításra kerül, ha a felhasználó felől érkezik). A függvénynek az eszközre kivitt karakterek számával kell visszatérnie. Ezt a függvényt a tty-meghajtónak kötelezően meg kell adnia. A *put_char* függvényt a tty-réteg akkor hívja, amikor az eszközre egyetlen karaktert kell továbbítani. Amennyiben az eszközön nincs hely a karakter számára, figyelmen kívül hagyható. Amennyiben egy tty-meghajtó nem adja meg ezt a függvényt, a tty-réteg a *write* függvényt hívja, ha egyetlen karaktert akar küldeni.

2. lista Egy minimál tty-meghajtó

```

#define TINY_TTY_MAJOR 240 /* experimental range */
#define TINY_TTY_MINORS 255 /* use the whole
↳major up */
static int tty_refcount;
static struct tty_struct
↳*tiny_tty[TINY_TTY_MINORS];
static struct termios
↳*tiny_termios[TINY_TTY_MINORS];
static struct termios
↳*tiny_termios_locked[TINY_TTY_MINORS];

static struct tty_driver tiny_tty_driver {
    magic: TTY_DRIVER_MAGIC,
    driver_name: "tiny_tty",
#ifdef CONFIG_DEVFS_FS
    name: "tts/ttty%d",
#else
    name: "ttty",
#endif
    major: TINY_TTY_MAJOR,
    num: TINY_TTY_MINORS,
    type: TTY_DRIVER_TYPE_SERIAL,
    subtype: SERIAL_TYPE_NORMAL,
    flags: TTY_DRIVER_REAL_RAW |
          TTY_DRIVER_NO_DEVFS,

    refcount: &tiny_refcount,
    table: tiny_tty,
    termios: tiny_termios,

    termios_locked: tiny_termios_locked,
    open: tiny_open,
    close: tiny_close,
    write: tiny_write,
    write_room: tiny_write_room,
};

static int __init tiny_init (void)
{
    /* register the tty driver */
    tiny_tty_driver.init_termios =
↳tty_std_termios;
    tiny_tty_driver.init_termios.c_cflag =
↳B9600 | CS8 | CREAD | HUPCL | CLOCAL;
    if (tty_register_driver (&tiny_tty_driver)) {
        printk (KERN_ERR "failed to register
↳tiny tty driver");
        return -1;
    }
    return 0;
}

static void __exit tiny_exit (void)
{
    tty_unregister_driver (&tiny_tty_driver);
}

module_init (tiny_init);
module_exit (tiny_exit);

```

A `flush_chars` függvényt a tty-réteg bizonyos számú karakter `put_char` függvénnel történt kiírása után hívja. A tty-meghajtónak az eszközt a visszaradott adatok azonnali továbbítására kell felszólítania.

A `write_room` függvényt a tty-réteg akkor használja, ha azt szeretné lekérdezni, hogy a tty-meghajtó még hány felhasználható hellyel rendelkezik az írási tárban. Ez a szám aszerint változhat, ahogy a karakterek kiürülnek az átmeneti tárból.

A `chars_in_buffer` függvény annak lekérdezésére szolgál, hogy hány karakter várakozik még mindig az írási átmeneti tárból való továbbításra.

Az `ioctl` függvényt a tty-réteg akkor hívja, amikor az eszköz-csomópontra `ioctl(2)` hívás érkezik. Ez a tty-meghajtó számára eszközfüggő `ioctl` függvények végrehajtását teszi lehetővé. Amennyiben az `ioctl`-kérést a meghajtó nem támogatja, egy `ENOIOCTLCMD` üzenettel kell visszatérnie. Ennek köszönhetően a tty-réteg képes arra, hogy ha lehet, az általános `ioctl`-változatot valósítsa meg.

A `set_termios` függvényt a tty-réteg akkor hívja meg, amikor az eszköz termiosbeállításai megváltoznak. A tty-meghajtónak ekkor meg kell változtatnia az eszköz fizikai beállításait a termiosszerkezet különböző mezőinek értékeitől függően. A tty-meghajtónak képesnek kell lennie kezelni azt az esetet, amikor függvény hívásakor egy korábbi változó esetleg `NULL` értékkel bír.

A `throttle` és `unthrottle` függvények a tty-réteg bemeneti tárolójának túlterhelését segítenek kezelni. A `throttle` függvény akkor használatos, amikor a tty-réteg bemeneti tárolója kezd megtelni. A tty-meghajtónak meg kell próbálnia jeleznie az eszköznek, hogy ne küldjön neki több karaktert. Az

`unthrottle` függvény akkor hívódik, amikor a bemeneti tár kiürült, és a tty-réteg újabb karakterek elfogadására kész. A tty-meghajtónak jeleznie kell az eszköznek, hogy az adatok fogadásának nincs akadálya.

A `stop` és `start` függvények sok tekintetben hasonlítanak a `throttle` és `unthrottle` függvényhez, de azt is jelzik, hogy a tty-meghajtónak az eszközhöz történő adatküldést félbe kell szakítania, és később folytatja a műveletet.

A `hangup` függvény azt jelzi a tty-meghajtónak, hogy a tty-eszköz tevékenységét fel kell függesztenie.

A `break_ctrl` függvény hívására akkor kerül sor, amikor a tty-meghajtónak az RS-232 kapu `BREAK` állapotát kell ki-bekapcsolnia. Ha az állapot `-1`, be kell kapcsolnia, ha `0`, akkor a `BREAK` állapot kikapcsolása szükséges. Amennyiben ezt a függvényt a tty-meghajtó végrehajtja, a tty-réteg kezeli a `TCSBRK`, `TCSBRKP`, `TIOCSBRK` és `TIOCCBRK` `ioctl`-eket. Ellenkező esetben ezek a tty-meghajtó `ioctl` függvénye számára továbbítódnak.

A `flush_buffer` függvény akkor kerül hívásra, ha a tty-meghajtónak az írási átmeneti tárból minden adatot ki kell ürítenie. Ez azt is jelenti, hogy a visszaradott adatok elvesznek, és nem továbbítja őket az eszközhöz.

A `set_ldisc` függvényt a tty-réteg akkor hívja, amikor megváltoztatta a tty-meghajtó sorszerkezetét. Ez a függvény általában többször nem is használatos, ezért nem is kell beállítani.

A `wait_until_sent` függvényt a tty-réteg akkor hívja, amikor a tty-meghajtó írási tárában lévő összes adatot ki akarja küldeni az eszközre. A függvény a befejezésig nem térhet vissza, ennek eléréséhez az alvó üzemmód is megengedett.

A `send_xchar` függvény a nagyobb fontosságú `XON` és `XOFF`

karakterek tty-eszköze való küldését végzi.

A `read_proc` és `write_proc` függvények akkor használatosak, amikor a meghajtó egy `proc/tty/driver/<name>` bejegyzést akar megvalósítani, ekkor a `<name>` a feljebb leírt `name` mező nevét veszi fel. Ha ezek közül valamelyik függvény be van állítva, akkor a bejegyzés létrejön és bármilyen `read(2)` vagy `write(2)` hívás az alkalmas függvényhez továbbítódik. Végül a `next` és `prev` mezőket a tty-réteg az összes különböző tty-meghajtó összefűzésére használja, a tty-meghajtónak nem szabad hozzányúlnia.

Nincs read függvény?

Az egyik dolog, ami szemet szúrhat a fenti függvénylistában, a `read` függvény megvalósításának hiánya. A tty-réteg egy átmeneti tárral rendelkezik, amely adatokat küld a felhasználó számára, amikor `read(2)`-hívás érkezik valamelyik tty-eszköz csomópontja felől. Azt a tárat a tty-meghajtónak kell feltöltenie, ha adat érkezik az eszköztől. Mivel a tty-adat nem mindig akkor áll rendelkezésre, amikor a felhasználó akarja, ennek a modellnek a használata szükséges. Ily módon a tty-réteg minden kapott adatot átmenetileg tárol, az egyedi tty-meghajtónak pedig nem kell aggodniuk a forgalom elakadása miatt, amíg az adat feltűnik a tty-vonalon.

A Tiny tty Driver

Most hogy túl vagyunk a különböző mezőkön, valójában mik azok, amikre egy alapvető tty-meghajtó felkészítéséhez és futtatásához szükségünk van? A 2. listán látható példa a lehető legkisebb tty-meghajtót tartalmazza. Ha a bemutatott lépéseket elvégeztük, hozzuk létre a `tiny_open`, `tiny_close`, `tiny_write` és `tiny_write_room` függvényeket, és már készen is vagyunk egy parányi tty-meghajtó létrehozásával. Ennek megvalósítására látható egy példa a 3. listán (39. CD Magazin/tty könyvtár). Ez a tty-meghajtó egy időzítőt hoz létre, ami – egy igazi eszközt utánozva – minden második másodpercen adatot helyez el a tty-rétegen. Továbbá megfelelően kezeli az eszköz szerkezetének zárolását, amennyiben SMP-gépen fut.

Az adatok áramlása

Amikor a tty-meghajtó `open` függvénye hívásra kerül, a meghajtótól azt várjuk, hogy adatokat mentsen a `tty_struct` változóba, amit átadunk neki. Ennek célja, hogy az eszközmeghajtó tisztában legyen vele, melyik eszközre hivatkozunk, amikor a későbbi `close`, `write` vagy más függvényeket hívjuk. Ha ezt nem tesszük, a meghajtó a `MINOR(tty->device)` függvény segítségével adja át az eszköz mellékszámát. Ha rápillantunk a `tiny_open` függvényre, látható, hogy a tty-meghajtóban a `tiny_serial` szerkezet kerül mentésre. Ez lehetővé teszi a `tiny_write`, `tiny_write_room` és `tiny_close` függvények számára lehetővé teszi, hogy visszakeressék a `tiny_serial` szerkezetet, és megfelelően változtassák meg. A tty-meghajtó `open` és `close` függvényei ugyanarra az eszközre többször is hívhatók – ahogyan a különböző felhasználók és programok kapcsolódnak az eszközhöz. Ez megengedné, hogy az egyik folyamat adatokat írjon, míg egy másik olvas. Hogy minden helyesen folyék le, meg kell számolnunk, hogy a kapu hányszor volt kinyitva és becsukva. Amikor először nyitjuk meg a kaput, elvégezhetjük a szükséges eszköz-előkészítést és memóriefoglalást. Az utolsó kapuzáráskor pedig megfelelően állíthatjuk le az eszközt és szabadíthatjuk fel a memóriát. Példaként figyeljük meg a `tiny_open()` és `tiny_close()` függvényeket, ahol látható, hogyan lehet megszámlálni a kapu nyitásait. Amikor az eszköz felől adat érkezik, szükség van rá, hogy a tty-

eszköz átmeneti fliptárában helyezük el. Ezt a következő kóddal is megvalósíthatjuk:

```
for (i = 0; i < data_size; ++i) {
    if (tty->flip.count >=
        TTY_FLIPBUF_SIZE)
        tty_flip_buffer_push(tty);
    tty_insert_flip_char(tty, data[i], 0);
}
tty_flip_buffer_push(tty);
```

Ebben a példában biztosított, hogy ne történjen túlsordulás, miközben adatokat helyezünk el a tárban. Azoknak a meghajtónak az esetében, amelyek nagyon nagy sebességgel képesek az adatok fogadására, a `tty->low_latency` kapcsolót kell beállítani, ennek hatására az utolsó `tty_flip_buffer_push()` hívás azonnal végrehajtódik. A példában a `tty_timer` függvény egy egybájtos adatot helyez el a tty fliptárában, majd az időzítőt a következő hívásra várva visszaállítja. Ezzel egy eszköztől érkező lassú karakterfolyamot utánoz.

Eszközre történő adat küldésekor a `write` függvény hívódik meg. Fontos, hogy a függvény ellenőrizi a `from_user` jelzőt, így elkerüli, hogy a felhasználói adatok véletlenül közvetlenül a rendszermag területére legyenek másolva. A `write` függvény számára engedélyezett a rövid írási visszatérés. Ez annyit tesz, hogy az eszköz nem tudott minden kért adatot elküldeni.

A `write(2)` függvényt hívó felhasználói program feladata – a visszatérési érték alapján – annak eldöntése, hogy az összes adat tényleg el lett-e küldve. Sokkal könnyebb ezt az ellenőrzést a felhasználói területen lefolytatni, mint egy meghajtóprogramnak addig várakoznia, míg az adatok kiküldésre kerülhetnek.

A tty-felület változásai

A tty-réteg a 2.0, 2.2 és 2.4 rendszermagokkal igen megbízható, és ez idő alatt nagyon kevésbé változott. Így a 2.0-s változathoz írt meghajtó a 2.4-es maggal szinte változtatás nélkül működni fog. A 2.5-ös rendszermagsorozatban a tty-réteg átírásra van kijelölve, így ez a cikk is tartalmazhat olyan részeket, amelyek már nem érvényesek. Amennyiben kétség merülne fel, forduljunk annak a rendszermagnak az `include/tty_driver.h` fájljához, amelyikre a meghajtót fejleszteni szeretnénk. Továbbá vessünk egy pillantást a `driver/char` könyvtárban található bármelyik tty-meghajtóra olyan függvények megvalósításainak példáiért, amelyek e cikkben nem szerepelnek.

Összegzés

Áttekintettük a tty-réteg alapjait, feltárva a 2.4-es rendszermag `tty_driver`-szerkezetének összes mezőjét, kiemelve, hogy melyekre van szükség egy eszközmeghajtó megvalósításához. A 3. listán látható `tiny_tty.c` meghajtó (39. CD Magazin/tty könyvtár) jó példa egy kis méretű, sikeresen működtethető tty-eszközmeghajtóra. A kód a jövőbeli saját tty-eszközmeghajtóink létrehozásához szabadon felhasználható.

Kapcsolódó címek a 39. CD Magazin/tty könyvtárában találhatóak.

Linux Journal 2002. augusztus, 100. szám



Greg Kroah-Hartman

(greg@kroah.com) jelenleg a Linux USB és a PCI Hot Plug rendszermagfelelőse.

Az IBM-nél dolgozik, ahol számos, a Linux rendszermagjával kapcsolatos kérdéssel foglalkozik.

Beágyazott rendszerek ízlés szerint

A Virtex-II Pro FPGA és a MontaVista Linux együttese rendkívül rugalmas megoldást kínál a tervezők számára.

Hányszor fordult már elő, hogy a követelmények meghatározása után döbbsenél rá, hogy egyszerűen nincs olyan mikroprocesszor, ami megfelelne elvárásaidnak és kellő számú csatolt eszközt lenne képes kezelni? Hányszor kellett szembesülnöd azzal, hogy a alkalmasabb processzorra még nincs átültetve a Linux? Mennyit kellett szenvedned, hogy a tervezéshez és a hibakereséshez az összes szükséges eszköz rendelkezésre álljon? Hány óra munkád ment el ezeknek a buktatóknak a megkerülésével? Nyugi, ne húzd fel magadat! A Virtex-II Pro FPGA családhoz tartozó eszközökkel, a MontaVista Linux operációs rendszerével és fejlesztői környezettel rendszeredet a saját elképzeléseid szerint, rengeteg idő és fáradság megtakarításával alakíthatod ki.

A Virtex-II Pro FPGA család akár négy PowerPC 405 típusú processzort, 16 darab 3,125 Gb/s sebességű Rocket I/O soros adóvevőt és 3,8 Mb blokk-RAM-ot (BRAM) is képes kezelni, valamint egy négymillió kapus programozható rendszerlapka is a rendelkezésedre áll. A nagy tudású eszközök alkalmazások széles köréhez használhatók fel, és kellő rugalmasságot biztosítanak a rendszertervezők számára. A teljes rendszer egyetlen lapkából is állhat, tetszőleges be- és kiviteli eszközök csatlakoztathatók hozzá, például használhatunk öt UART-ot, egy PCI-sínt és akár több Gigabit ethernetkaput is – mindet a központi lapkán elhelyezett processzorok vezérlik.

A Xilinx és az IBM együttműködése révén valósul meg a PPC405 szabvány támogatása, amely egy rendkívül üzembiztos sínrendszert jelent, valamint az egyes összetevők egymás közötti kapcsolatát is meghatározza. A Xilinx igen sokféle eszközt kínál a Virtex-II Pro FPGA-hoz, amelyek meghatározott szolgáltatáskészlettel rendelkező modulok, és közvetlenül csatlakoznak az IBM CoreConnect sínrendszere szerint működő közös sínre (lásd az *ábrán*).

A CoreConnect számos szolgáltatással segíti a processzor és az FPGA-összetevő összekapcsolását, így állnak össze teljes berendezéssé. A legfontosabb CoreConnect-szolgáltatások: alkalmazás-központú rendszersínek, amelyek között megtalálható a helyi processzorsín (PLB), a lapkán elhelyezett, a külső eszközök használatát lehetővé tévő sín (OPB), valamint az eszközezőregisztráló regisztersín (DCR). A modulok révén sokféle be- és kiviteli módszert és külső memóriavezérlőt is használhatunk.

A be- és kiviteli eszközök között említhetem meg a 16450-es és 16550-es UART-, az IIC- és SPI-vezérlőket. A memóriavezérlők között SRAM/FLASH, SDRAM, DDR SDRAM és ZBT eszközök kezelésére alkalmasak egyaránt található. Modulként természetesen további rendszerszintű be- és kiviteli felületeket is elérhetünk: PCI-vezérlőt, 10/100 Mb és Gb sebességű vezérlőket, valamint a System ACE beállítófelületet is. Minden modulhoz eszközzillesztő program is tartozik, ha a használatához szükség van rá. Az eszközzillesztőket forráskódként bocsátják rendelkezésre.

A sokoldalú programtervező eszközöket, amelyek segítségével a felhasználók kiaknázzhatják a processzorok, a modulok és

általában véve az FPGA képességeit, részletesebben is ismertetem. A vas és a programok közötti magas szintű párbeszéd teszi egyedivé a Virtex-II Pro FPGA-t, hiszen így válik lehetővé a teljesen testreszabott beágyazott megoldások egyetlen lapkán történő fejlesztése. A vas és a rajta futó alkalmazások együttes tervezéséhez kiváló programtervező és beágyazott hibakereső alkalmazások állnak rendelkezésre. Mielőtt azonban elmerülnénk az együttes tervezéssel kapcsolatos részletekben, ismerjük meg egy kicsit jobban a Virtex-II Pro FPGA lelkivilágát!

A PowerPC 405 processzorok 300 MHz-es vagy nagyobb órajellel futnak, 16 KB adat és 16 KB utasítás-gyorsítótárral rendelkeznek. A helyi processzor- és külső eszközöket csatlakoztató sín 100 MHz-es vagy nagyobb órajellel üzemel.

A Virtex-II Pro FPGA egy fontos jellemzője a lapkán elhelyezett memória (OCM). Az OCM elérési jellemzőit tekintve hasonlóan viselkedik, mint a gyorsítótár, ám kezelését a felhasználó végzi. A Virtex-II Pro FPGA OCM-ként kétkapus BRAM-ot tartalmaz, amely rendkívül nagy sebességű adattovábbítási

A Virtex-II Pro Platform FPGA-k az alábbi protokollokat és sebességeket támogatják

| Protokoll | Sebesség |
|----------------|------------|
| 3GIO | 2,5 Gb/s |
| Serial ATA | 1,5 Gb/s |
| InfiniBand | 2,5 Gb/s |
| Gb ethernet | 1,25 Gb/s |
| 10 GE (XAUI) | 3,125 Gb/s |
| Serial RapidIO | 1,25 Gb/s |
| Fibre Channel | 1,06 Gb/s |

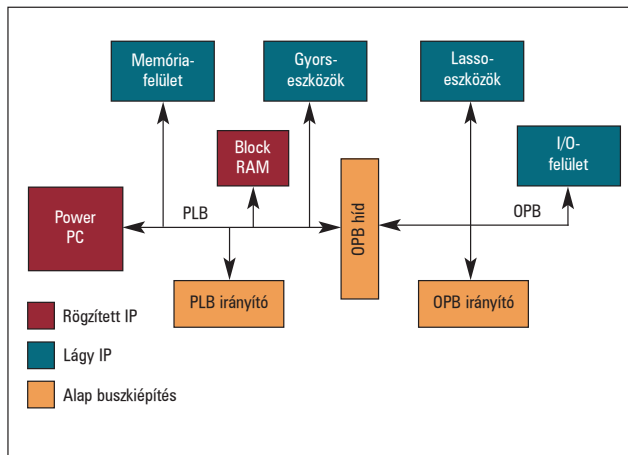
eszközként szolgál a külső eszközök és a processzor, illetve átmeneti tárként a processzorok között.

A 3,125 Gb/s sebességre képes Rocket I/O-soros adóvevők számos különböző adattovábbítási szabványt támogatnak, amelyek felsorolása *táblázatunkban* látható. Az olyan fontos szolgáltatásokat, mint az átmeneti tárcák biztosítását, a 8B/10B kódolást és dekódolást, a CRC-számítást és -ellenőrzést a lapkák valósítják meg, így az FPGA-összetevő mellett semmi sem foglalja a helyet.

A BRAM-ok a processzor által kezelt memóriákként használhatók, és a CoreConnect sínre vagy OCM-ként csatlakoztathatók. Az OCM jobb teljesítményt és közvetlen hozzáférést kínál a processzor részéről, a BRAM-ot a helyi processzorsínre kötve az összetevők DMA-átvitelét is végezhetnek, amivel nem terhelik a processzort. A tervező feladata tehát eldönteni, hogy mi módon lehet a BRAM-ot a legjobban kihasználni.

Az FPGA-összetevő a Xilinx moduljaival, illetve saját tervezésű modulokkal egészülhet ki. Egyes modulok egymással is folytatnak párbeszédet, mások viszont csak párhuzamosan tudnak dolgozni.

Mivel az FPGA-k tetszés szerint átállíthatók, a csatlakozóeszközök száma és típusa a rendszer működése közben is módosítható. Ilyen rugalmas rendszerhez megfelelő operációs rendszerre is szükség van, illetve olyan eszközökre, amelyek segít-



A CoreConnect segítségével csatlakozó modulok



A Virtex-II Pro ML300 felülnézetből

ségével a készülékek és a programok tervezői beágyazott alkalmazásait fejleszthetik és hibajavítással láthatják el. Az alkalmazástól függően a felhasználók általában a kereskedelmi, készen megvásárolható valós idejű operációs rendszerek közül választanak, vagy beágyazott Linuxot telepítenek. A Linux világból a Xilinx a Virtex-II Pro FPGA családhoz rendkívüli sokoldalúsága miatt a MontaVista Linuxot választotta, mint a céljainak leginkább megfelelő beágyazott operációs rendszert. A Linux alapállapotban is rengeteg eszközt támogat. A Xilinx és a MontaVista Software réteges eszközülesztő szerkezet fejlesztése mellett döntött. Ez egy alacsony szintű, operációs rendszertől független rétegből áll, amely közvetlenül a vasat kezeli; valamint egy operációs rendszertől függő illesztési rétegből, ami az operációs rendszer és az alacsony szintű eszközülesztők közé fészkel be magát. A Xilinx készíti az alacsony szintű eszközülesztőket, amelyek így a lehető legjobban kihasználják a modulok képességeit. A MontaVista Software a linuxos illesztési réteget készíti el,

bevetve az eszközülesztők átlátszó beépítésével kapcsolatos tapasztalatait. Mindkét réteg nyílt forrású tárházba kerül és GPL szerződéssel érhető el. Az eszközülesztők a rendszer-magba fordítva és betölthető modulként egyaránt használhatók. Meghagyva a modulok betöltésének és eltávolításának lehetőségét olyan készülékek építhetők, amelyek a rendszer futása közben is átállíthatók. Futó operációs rendszer mellett teljesen megszokott dolog egy-egy USB-s készüléket vagy PC-kártyát kicserélni, ám magán a lapkán található eszközt leváltani, és menet közben betölteni a hozzá tartozó illesztőprogramot – ez azért egészen új dolog.

A Linux-átültetés elsőként a Xilinx Virtex-II Pro ML300 felülethez (lásd a képen) készült el, és az áramkörön található modulok és eszközök nagy részét támogatja. Mivel az ML300 sokoldalú, jól kidolgozott áramkör, az átültetést a felhasználók saját eszközeikhez minden bizonnyal könnyedén hozzá tudják majd igazítani. A MontaVista Software magas szintű támogatást nyújt a hasonló tervezetekhez.

Lássuk, milyen eszközök állnak az eszköz- és programmérnökök rendelkezésére a rendszerek fejlesztéséhez, indításához és hibáik kijavításához. A Xilinx által biztosított System Generator for Processors, a GDB/XMD, a System ACE és a ChipScope Pro teljes eszközkészletet jelent bármely vassal, alkalmazásokkal és rendszerrel kapcsolatos tervezéshez.

A System Generator for Processors (SGP) segítségével a Virtex-II Pro FPGA alapján fejlesztett eszköz mind a vas, mind a programok oldaláról végleges egésszé állítható össze. Felhasználóbarát párbeszédablakokban lehet megadni a rendszer beállításait, például a csatlakozóeszközök alapcímzeit, a használni kívánt megszakításokat vagy a telepített memória nagyságát. Az SGP elkészíti az eszköz felépítését leíró állományokat, amelyek alapján az FPGA legyártható, illetve a szimulációk lefutathatók. Egy beállítóállományt is készít, amely a Linux-rendszermag és a szükséges illesztőprogram-modulok fordításához használható. *Listánk* egy ilyen beállítóállomány egy részletét ismerteti, ebben a felhasználó által megadott, a rendszer megszakításvezérlőjének működését befolyásoló beállítások találhatóak.

Az SGP segítségével a tervezők különböző beállításokkal és variációkban próbálhatják ki születőben lévő beágyazott rendszerüket. A beállítások módosításával mind a vas, mind a programok a pillanatnyi követelményeknek megfelelően finomhangolhatók. Csak azok a szolgáltatások kerülnek be a késztermékbe, amelyekre valóban szükség van, a feleslegesek a tervezés során eltávolíthatók. Az eszközök alapértelmezett előbeállításokkal is rendelkeznek. Mindennek eredménye az, hogy a vas és a programok kevesebb helyet igényelnek, jobb teljesítményt nyújtanak és egyszerűsödik – illetve egyes esetekben teljesen elmarad – a rendszerindítási eljárás is. Az SGP és a hozzá tartozó eszközök nyílt felületet használnak, így bárki bővítheti őket saját eszközeinek szolgáltatásaival vagy eszközülesztőivel. Az XMD egy hibakereső kiszolgáló, a vizsgált rendszeren található GDB-vel az On-Chip Debug (OCD) protokoll segítségével tartja fenn a kapcsolatot; és a célrendszert a Virtex-II Pro FPGA-ban található processzor JTAG kapuján keresztül vezérli. Az XMD egyszerre több GDB kiszolgálására is képes, így egyidejűleg több processzoron is hibakeresést végezhetünk. Hogy pontosabb legyek: a Virtex-II Pro FPGA-n belül található négy processzor mindegyikén akár egyazon időben is végezhetünk hibakeresést. Linux alatt a GDB parancssorból vagy a többféle grafikus felület valamelyikével futtatható. Alapértelmezettként Insight-felületet támogat, de DDD és Emacs is társulhat hozzá.

© Kiskapu Kft. Minden jog fenntartva

```

                A beállítófájl eszközülesztők beállítására szolgáló részlete

#ifdef XPARAMETERS_H      /* a k r k r s beillesztések megelőzésére */
#define XPARAMETERS_H      /* védelmi makr kat használva */

/*****
 *
 * Megszak tés-vezérlő (Intc) meghatározások.
 * DeviceID kezdőértéke 0
 *
#define XPAR_INTC_NUM_INSTANCES      1      /* A példányok száma */
#define XPAR_INTC_NUM_INSTANCES      1      /* Device ID példa */
#define XPAR_INTC_0_MAX_ID            31     /* jelenleg csak 32
                                                megszak tés használhat */
                                                /* az őrvényes tartomány 0-31 */
#define XPAR_INTC_0_REG_BASE          0xD0000FC0
                                                /* Regiszter alap cím */
#define XPAR_INTC_0_UARTNS550_0_VEC_ID 31
#define XPAR_INTC_0_IIC_0_VEC_ID      29
#define XPAR_INTC_0_EMAC_0_VEC_ID     28

#define XPAR_INTC_MAX_ID              31     /* az őrvényes tartomány
                                                0-31 */

#endif      /* vége a védelmi makr nak */

```

hasznos lehet, ha ugyanazokat a jeleket több ILA is vizsgálja. Volt olyan eset, hogy két ILA-egységet csatlakoztattunk a PLB cím- és adatvonalaira, hogy egy memóriakezeléssel kapcsolatos hibát vizsgáljunk. Az egyik ILA a PPC405-re csatlakozva vizsgálta a PLB-jeleket. Tudtuk, hogy a hiba fellépésekor a processzor hozzáfér a memóriához. A másik ILA a DDR memóriavezérlőre csatlakozva figyelte a PLB jeleit. A két ILA által a cím- és adatvonalakról szállított adatok alapján megtaláltuk a hibát, és sikeresen kiküszöböltük. Az eszköz belső világának elérhetősége és a síneken folyó átvitelek megfigyelhetősége különösen Linux-alapú, MMU-t is használó rendszer-

A GDB-t a PPC405 rendszerek támogatásával bővítették, illetve a target ocd parancs is képes a Virtex-II Pro FPGA összes szolgáltatásának kihasználására. Ennek köszönhetően a PPC405 összes regisztere, a gyorsítótárak és a TLB bejegyzései, illetve az OCM tartalma nemcsak figyelemmel kísérhető és módosítható, hanem a processzor memóriaterébe is bemásolható.

A beágyazott rendszerek hibáinak felderítése hagyományosan nehéz feladat – a vas és a programok működését egyszerre kell figyelemmel kísérni. Külső, a folyamatokba bele nem avatkozó hibakereső programmal, például a GDB és XMD párossal a munka sokkal egyszerűbb. A PPC405 emellett vasból megvalósított töréspontokat is tud kezelni, így kivételek esetén a processzor működése befagyasztható. A végrehajtott adatátvitelek és a memória-hozzáférések figyelemmel követése különösen nehezzé válik, ha a processzorok és a hozzájuk csatlakozó eszközök egyetlen lapkára kerülnek. Ekkor ugyanis minden fontos jel a lapkán belül utazik, és gyakorlatilag nem nagyon lehet hozzájuk férni. A Virtex-II Pro FPGA esetén nincsenek ilyen korlátozások, mivel a külső eszközök „puha” vasként épülnek be a rendszerbe. A megfelelő eszközzel minden jel látható és elérhető.

A ChipScope Pro egy a vasba beépített logikai elemző. Egyrészt a logikai elemzőből áll, amely a vizsgálórendszeren fut, másrészt a magába az eszközebe fordított vagy beépített ravaszokból és adategységekből. A beépített logikai elemzőegységek (ILA) tetszőleges számú belső jel vizsgálatára állíthatók be, és a felhasználó által megadott feltételek teljesülésekor vagy a processzorsín-átvitelek hatására cselekednek. Egyezre több ILA-egység is működő lehet. Egyes esetekben

nél hasznos. Ekkor a programok oldaláról nézve ugyanaz a fizikai memóriablokk több képzetes címtérre is leképezhető, a vas oldaláról nézve viszont minden cím fizikai.

A ChipScope Pro, a GDB és az XMD együtt alkalmazva rendkívül mély betekintést tesz lehetővé a rendszer belső életébe. A programok közös, az FPGA JTAG kapujára csatlakozó adatkábelt használnak. Az alkalmazások barátságos egymás mellett élése jelentősen megkönnyíti a hibakeresési környezet kialakítását.

A rendszerindítás folyamata roppant fontos a beágyazott készülék bekapcsolásakor. Néhány lépésben megtörténik a processzor, a memóriarendszer és az átviteli eszközök indítása. A Virtex-II Pro FPGA esetében a rendszerindítás két lépésben zajlik. Egyrészt megtörténik az FPGA beállítása, másrészt elindul a processzor. Az FPGA-t a számos lehetséges módszer egyikével kell beállítani a szükséges feladatokhoz. A cikk során még visszatérünk az egyik jól használható eljárásra. Gyakran az igényekhez igazított eszközzel folyik a rendszer és a processzor indítása, a Linux-rendszer mag betöltése a memóriába, majd a vezérlés átadása a rendszer mag belépési pontjára. A Virtex-II Pro FPGA a hagyományos rendszerindítást is támogatja, amelynél az elsődleges rendszertöltő külső ROM-ban, vagy a belső BRAM-ban található. Az utóbbi esetben nincs is szükség külső ROM-ra, ilyenkor az FPGA beállítását végző bitfolyam a rendszerbetöltőt is tartalmazza. Közvetlenül azután, hogy az FPGA beállítása megtörtént, a processzor kikerül az alapállapotból, a BRAM-ból megkezdja az utasítások olvasását, és lefuttatja az elsődleges rendszerindítót.

Egy – különösen MontaVista Linuxsal együtt – remekül hasz-

nálható megoldás a System ACE segítségével indítja a rendszert. A System ACE külső kísérőlapka a Virtex-II Pro FPGA-hoz, és ROM nélkül is lehetővé teszi a rendszer indítását. Két fő feladata van: az első az FPGA, a processzor és a processzorsínen található egyéb eszközöknek a JTAG kapun keresztül történő beállításával végzett rendszerindítás, amelyet Compact Flash-kártya vagy Microdrive segítségével végez el. A második, hogy az általa használt tárolóeszköz a Linux által is elérhető fájlrendszerként jelenik meg.

A Microdrive egy FAT12 vagy FAT16 és egy Linux-részt tartalmaz. A Linux-rendszer a System ACE támogatására van beállítva, fordításkor a System ACE különleges fájlformátumára alakítják, összefűzik az FPGA-t beállító bitfolyammal, és a FAT-fájlrendszeren helyezik el. A készülék bekapcsolásakor a System ACE a beállítóállományt beolvassa a FAT-fájlrendszerből, beállítja az FPGA-t és elindítja a rendszermagot. Az indítási folyamat alatt a Linux-rendszer a Microdrive linuxos részét saját könyvtárként fűzi be. A System ACE használatának ugyancsak fontos előnye, hogy a processzor alapállapotához nem szükséges külön memória, a FAT-részen több különböző indítási beállítás is tárolható, és ezek normál fájlműveletekkel is módosíthatók.

A System ACE a JTAG láncot veszi igénybe, akár a processzor hibakereső kapuját használó külső hibakeresők. A programkód, az adatok és szükség szerint a Linux-rendszer a Microdrive által használt ramlemez betöltése a memóriába a JTAG láncon és a processzorsínen keresztül történik. A processzor által elérhető egyéb összetevők beállítása is hasonló módon, a rendszermag betöltése előtt végezhető el. A folyamat végén a PPC405 programszámlálóját a Linux-rendszer kezdő címére állítjuk, és megkezdjük az operációs rendszer futtatását.

A System ACE egy kapcsolót tartalmaz, amely a nyolc működő beállítás egyikét választja ki. A betöltendő beállítást programból is lehet módosítani. A futó Linux-rendszer kiválasztja az új beállításokat, alapállapotba hozza a gépet, amely immár az új beállításokkal indul el, akár teljesen más külső eszközöket használva.

A FAT-fájlrendszer révén a System ACE által használt állományt a rendszer működése közben is lehet frissíteni – így lényegében használat közben a teljes eszköz javítható, frissíthető.

A Virtex-II Pro fejlesztői készletével szinte élmény egy-egy új rendszert tervezni. A készlet segítségével még annak megépítése előtt szimulálható a beágyazott rendszer működése, illetve a hibakeresési szakasz is újabb elvonatkoztatási szinttel gazdagodik. A készülék minden egyes összetevője önmagában is vizsgálható. A teljes készülék összeállítás után a vas és az alkalmazások együttes működése is utánozható, így a beágyazott rendszer kifogástalansága ellenőrizhető.

A valódi eszközben felfedezett hibák a szimulációs szakaszba visszavezethetők és pontosan visszakövethetők. A GDB/XMD oly módon is beállítható, hogy HDL szimulátorhoz kapcsolódjon, következképpen a tervező lépésről lépésre követheti a programok végrehajtását, figyelheti a síneken folyó átviteleket, valamint az eszközök állapotváltozásait. A Virtex-II Pro FPGA és a MontaVista Linux együttese százszázalékos, kiváló megoldást kínál számos különböző alkalmazási területre. A soros, több gigabit sebességű Rocket I/O adóvevőknek köszönhetően távközlési területen is jól használható, például bázisállomásokon, ahol a nagy sávszélesség mellett hatalmas számítási teljesítményre is szükség van. Ugyanezeket az adóvevőket több eszköz között hátoldali átkötésként is lehet használni. Az elérhető külső eszközök a négy processzorral társulva kiváló választással teszik adatfeldolgozási és grafikus

feladatokhoz, akár munkaállomásként is.

A processzorok az FPGA-lapba építése több jövőbeli fejlesztéshez és érdekes elgondoláshoz szolgál alapul. Rendszertervezési szempontból egyetlen közös processzorsínen egyetlen gépben több processzor rákapcsolható. A kifinomultabb rendszerek a sín bedugulását kapcsolókkal előzik meg, és növelik a teljesítményt. Az FPGA jellegénél fogva a rendszertervezők először az egyszerűbb megoldást választják, majd áttérhetnek a bonyolultabbra. Teljesen mindegy, hogyan döntenek, a Linux hűsége társuk marad. Mivel a jelzők és mutatók kétkapus BRAM-okkal könnyedén megvalósíthatók, az erőforrások kezelése és a megosztott memóriakezelés különösebb nehézségek nélkül megoldható. A vas és az alkalmazások szoros együttműködése jelentősen növeli a rendszer teljesítményét. A vas oldaláról a gépek gyorsak, képesek a párhuzamosított működésre, a programok pedig rendkívül rugalmasak. A Linux számos olyan rendszerhívást használ, amelyek vasból valósíthatók meg. A rendszertervező számára csak az jelenthet kihívást, hogy meg kell találnia azokat a függvényeket, amelyeket érdemes a vasra terhelni, ezzel is gyorsabb, kifinomultabb rendszert építve. A még bonyolultabb gépeknél a Linux mellékprocesszorok igény szerinti használatára is képes. Ilyenkor részben átállítja az FPGA-t, így az éppen futtatott alkalmazás igényeinek megfelelő eszközök elérhetővé válnak. Míg az egyik alkalmazás nagyszámú FFT-átalakítást végez, a másik például adott részeket keres az adatfolyamokban. Amikor az időzítő az egyik alkalmazástól a vezérlést átadja a másiknak, a megfelelő kiegészítő modul is lecsereéli. Az időzítő a korábbi tapasztalatok alapján dönti el, hogy érdemes-e kihasználni a vas gyorsítását, vagy a megfelelő programból megvalósított függvényeket kell alkalmazni. A Virtex-II Pro FPGA és a MontaVista Linux a megfelelő rendszertervező, hibakereső és beállító eszközökkel együtt rendkívül rugalmas és kiváló teljesítményű eszközkészletet jelentenek. Az elkészülő eszköz tervezése a követelmények, és nem az eszközök és programok korlátjai szerint történik, a rendszer részei magas fokon egybeépíthetők, miközben nem kell lemondani a megfigyelhetőségről, a moduloknak és a hozzájuk kapható illesztőprogramoknak köszönhetően csökken a piacrajutás ideje, végül a mérnökök alkotóereje új teret nyer a készülékek és a rajtuk futó alkalmazások együttes tervezésében.

Linux Journal 2002. augusztus, 100. szám



Michael Baxter

kilencéves kora óta dolgozik a számítógépiparban, amióta magával ragadta az 1969-os 2001-vízió, az Úrodüsszeia. Tapasztalt számítógép-, rendszer-, áramkör- és FPGA-tervező. Tíz USA-szabadalmat jegyez számítógépek és áramkörök tervezésével kapcsolatban, valamint további ötnek a társhelfelalója. Szeret túrázni, érdekli az amatőr rádiózás és a Lisp nyelven való programozás.



Peter Ryser

rendszertervező mérnök a Xilinx, Inc. cégnél. Több, a Virtex-II Próval kapcsolatos beágyazott alkalmazástervezetért felelős. A peter.ryser@xilinx.com címen érhető el.

GNU Bayonne, a kapcsolatteremtő

A Bayonne Projekt bármely gyártó kártyájával képes telefonos kapcsolatot létrehozni.

Három évvel ezelőtt rá kellett jönnöm, hogy komoly hiány tapasztalható a szabad programok terén. Igaz ugyan, hogy a szabad programok elterjedtek, és a vállalkozói területen szinte minden részt be is töltöttek, de egyikük sem próbálta megoldani a távközlésben felmerült gondokat.

Pedig a kapcsolattartás nemcsak hogy minden üzletnek része, de egyben (gyakorta észre sem vett) része a felhasználói élménynek is. Ugyanakkor a nyilvános telefonos hálózaton keresztül működő telefonos szolgáltatások létrehozásához szükséges alkatrészek egyre szélesebb körben terjednek el a piaci PC gépek és operációs rendszerek körében, ideértve természetesen a GNU/Linuxot is.

Hogy szabad programmal célozhassuk meg ezt a piacrészt, néhányan úgy döntöttünk, egy keretrendszert hozunk létre, amelyben leírjuk, milyennek kellene lenniük ezeknek a szolgáltatásoknak; az asztali felhasználók és alkalmazásfejlesztők igényeitől kezdve egészen a legnagyobb távközlési cégek által támogatott elvárásaiig. A projekt később GNUCOMM néven vált ismertté, amikor hivatalosan is a GNU projekt munkacsoportjává vált.

Az egyik olyan terület, amit meg szerettünk volna határozni, a telefonos alkalmazáskiszolgáló volt. Egy ilyen kiszolgáló a könnyen telepíthető új telefonos alkalmazáskiszolgálások egyszerű létrehozását tenné lehetővé. Ezek olyan alkalmazások lennének, amelyeket kifejezetten arra fejlesztenek ki, hogy a kiszolgálót a hagyományos telefonvonalon, elérő hús-vér emberekkel tartásuk a kapcsolatot, akik az alkalmazással hang, illetve a telefon nyomógombjainak segítségével érintkezhetnek. Jellegetesen ilyen alkalmazások a hangpostafiók-rendszerek vagy az előre fizetett (prepaid) hívási megoldások. Minden ilyen rendszer meglehetősen összetett, valamint a PC-rendszer és a nyilvános telefonhálózat közötti kapcsolat megteremtése néha programozott rendszereket vagy különleges számítógépes telefonalkatrészeket igényel. Ez lehet olyan eszköz amely analóg telefonvonalon képes beszélni a hívóval,

de akár olyan berendezés is, amely többkapus hangvezérlést nyújt ISDN vagy T1 digitális hangáramkörön keresztül, amit a nagyobb vállalatok közvetlenül a helyi távközlési szolgáltató központjából kaphatnak meg. Figyelembe véve, hogy az ilyen rendszerek a múltban általában igen költségesnek számítottak, soha nem voltak nyíltak, és gyakran nehezen programozhatónak bizonyultak, úgy gondoltam, ezeket a gondokat egyszerre fogom megoldani a jelenleg leglátogatottabb nyílt programrendszer, a GNU/Linux alá írt kiszolgálóprogram létrehozásával. Amikor nekikezdünk a projektnek, még igen kevés cég kínált GNU/Linux alatt is használható alkatrészeket, így azt használtuk, ami elérhető volt. Még manapság is minden telefonos kártya teljesen különböző, és többnyire saját API-t melkélnek hozzá. Mivel sem az eszközök, sem az API-k nem szabványosítottak, a legtöbb ember, aki telefonos alkalmazásokat készít, csak egyetlen gyártó kártyacsoportjához fejleszt, és ezt kizárólag a gyártó által nyújtott API-n keresztül teszi. Ez a gyakorlat egyben azt jelenti, hogy a számítógépes távközlési üzletben minden gyártónak igen széles termékcsalád-kínálatot kell készítenie, mivel a felkínált termékcsalád részeit nem egykönnyen lehet más termékekkel helyettesíteni. Ezek a tényezők igencsak megnehezítik, hogy új telefonos kártyagyártó lépjen a színre, ugyanakkor megkönnyíti a korlátozott számban jelenlévő gyártók dolgát, hogy komolyabb változtatás nélkül megtarthassák piacukat. Természetesen nem állíthatjuk, hogy egyáltalán nem történtek próbálkozások szabványosított API kialakítására. Végül is ott van a ECTF (European Community Telework/Telematics Forum). Lévéni üzleti gyártói ipari konzorciuma, már elő kellett volna állniuk egy, bizottságokon keresztül vizsgált összetett szabványkészlettel, és olyan javaslatokkal, amelyek azt határoznák meg, hogyan fejlesszék és támogassák a gyártók a számítógépes telefonos megoldásokat. Továbbá mindezt olyan módon kellene megtenniük, amely növeli a különleges tudás iránti igényt, emelve ezzel a

számítógépi telefonos piacon jelenlévő tagjaikra nehezedő nyomás mértékét. A másik népszerű szervezet az ITU (International Telecommunications Union) névre hallgat, amely leginkább arról ismert, hogy kinevezéseit gyakran nemzeti kormányzatok adják. Az USA-ban például a kinevezéseket az állami részleg adja, politikai alapokon, ahelyett, hogy a legjobb és legragyogóbb elmék közt válogatnának.

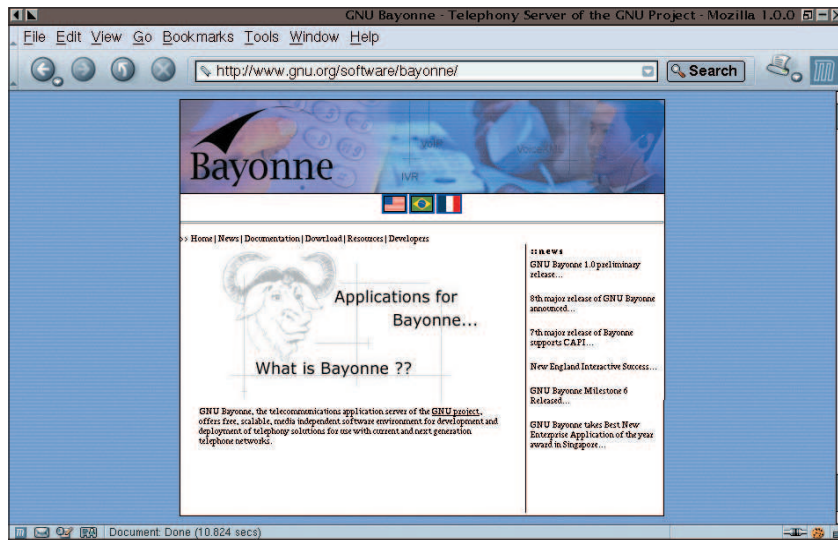
A mi célunk nemcsak az volt, hogy szabad programként telefonos kiszolgálót hozzunk létre, de egyúttal a telefonos alkalmazáskiszolgálásokat is olyan könnyen és egyszerűen kezelhetővé szerettük volna tenni, amilyen egy weblap készítése és karbantartása. Ugyanakkor a telefonos meghajtókat és API-kat egy olyan szintig akartuk elvonatkoztatni, hogy ezek az alkalmazáskiszolgálások fejlesztésénél már egyáltalán ne számítsanak, és láthatatlanok legyenek. Ha ez sikerülne, az azt jelentené, hogy bárki tetszés szerint lecserélheti az eszközeit, ahelyett, hogy egyetlen gyártó által nyújtott termékekhez lenne kötve.

Kezdetben volt az ACS

Mivel a kiszolgálóban mindent alacsony szinten akartunk elvonatkoztatni, az első szükséges dolog egy C++ nyelven íródott osztálykészlet volt. Több okból is a C++-t szerettem volna használni. Először is, a meghajtott csatolófelületek létrehozásához az osztálybeágyazások felhasználása tűnt természetesebbnek, figyelembe véve elvont természetüket. Másodsor, úgy láttam, sokkal gyorsabban tudok hibamentes C++-kódot írni, mint hibamentes C-kódot. Tulajdonképpen ez lett az első nagyméretű C++-projekt. Ugyancsak könnyű megmagyarázni, miért nem a már meglévő keretrendszereket használtuk. Tudtuk, hogy szükségünk lesz szálalásra, foglalatok (socket) támogatására és néhány egyéb elemre. Egyetlenegy létező keretrendszer sem tudta az összes szükséges dolgot biztosítani, kivéve néhányat, amelyek azonban jóval nagyobbak és összetettebbek voltak, mint amire

nekünk szükségünk volt. Például szeretnünk volna egy telefonoskiszolgáló-vázat. Ez idő tájt a legjobban használható keretrendszer a ACE (Adaptive Communication Environment) volt, amely általában néhány MB-tal növelte meg a futtatható könyvtár méretét. Mivel célunk az volt, hogy akár 8–12 MB

De milyen formát ölthet egy ilyen parancsnyelv? Sok parancsnyelv egyes értelmező példányhoz elkülönített végrehajtást tételez fel (szálanként vagy folyamatonként) minden ami sajnos használhatatlanná teszi őket a céljainkra. Sok nyelv feltételezi, hogy a kifejezés értelmezésének ideje nem határozható



memóriával rendelkező gépeken is futtatni tudjuk a programot, ez elfogadhatatlan hátránynak bizonyult. A GNU Common C++ (eredetileg APE) azért jött létre, hogy egyszerűen értelmezhető és hordozható osztálymeghatározásokat nyújtson a szálak, a foglatatok (socket), a jelzők (semaphore), a kivételek és egyéb elemek részére. Az APE azóta megnőtt: mostanra már számos fejlesztés forrásává vált azon felül, hogy a GNU része.

Maguknak a szolgáltatásoknak a készítése közben ráébredtünk, hogy új módszerre van szükségünk a telefonos alkalmazások létrehozásához – mégpedig olyanra, amely a folyamatot egy átlagos rendszergazda számára is érthetővé teszi. Az egyszerűség kedvéért egy általános parancsnyelv használata mellett döntöttünk, amely később GNU ccScript néven vált ismertté. Parancsfájlok írásával és hangminták rögzítésével gyakorlatilag bárki anélkül hozzájárulhat a telefonos alkalmazáskiszolgálások létrehozásához, hogy különleges ismeretekre vagy átfogó rálátásra lenne szüksége; vagy olyan hihetetlenül összetett API-kra, mint amilyeneket a ECTF támogat. Mivel az alul elhelyezkedő telefonos vas láthatatlan, és csak elvont formában érhető el az alkalmazás parancsfájlnyelvből, a kártyacsatládtól való függőség hátránya is megszűnt.

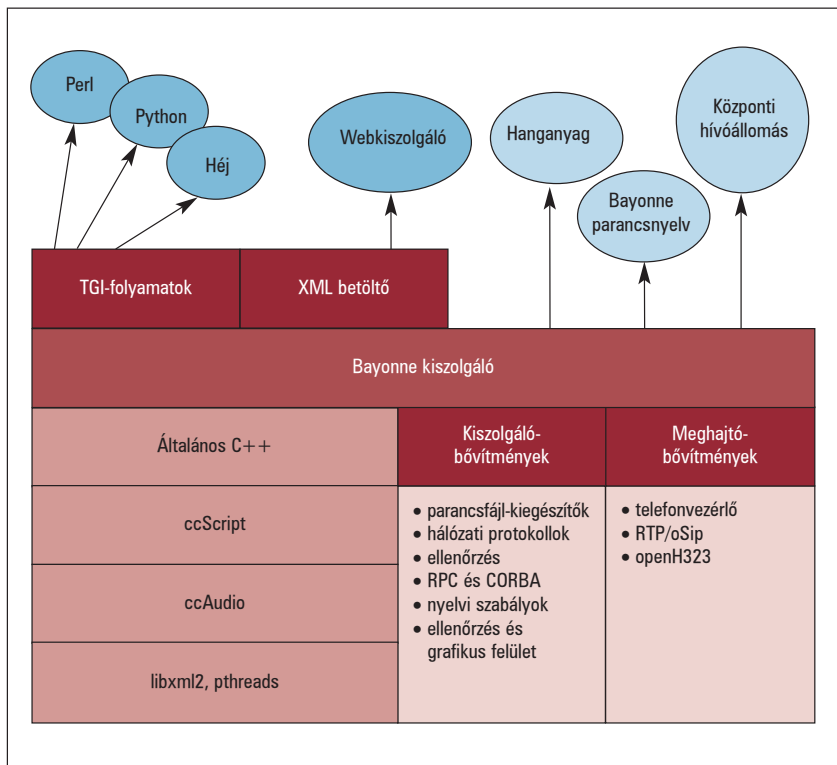
meg. Egy kifejezés ugyanis például önhívó (rekurzív) függvényeket vagy akár egész alprogramokat meghívhat. Mi azonban – ahogy már korábban említettük – nem szeretnünk volna minden egyes értelmező példányhoz egy-egy külön végrehajtási folyamatot rendelni, és azt sem akartuk, hogy az összes példány válaszoljon a telefonos vezérlő legelső eseményhívására, amint az állapota megváltozik, így aztán a már létező, általánosan használt megoldások (Tcl, Perl, Guile stb.) egyike sem jöhetett közvetlenül szóba. Ehelyett első kiszolgálónkhoz inkább egy új, nem blokkosított, determinisztikus parancsnyelvet készítettünk. Parancsnyelvünk több szempontból is egyedülállónak volt nevezhető. Először is lépésenként végrehajtott volt és nem blokkosított. Az utasítások vagy azonnal végrehajtottak és visszatértek, vagy a végrehajtó segítségével végrehajtásukat egy későbbi időpontra időzítették. Így lehetővé vált, hogy egyetlen szál több értelmezőpéldányt hívjon meg és kezeljen. A telefonos kiszolgáló akár egy idejű hívások százait kell kezelje egy nagy terheltségű távközlési eszközön. Mivel nekünk nincs szükségünk a kiszolgálón futó hagyományos szálak százaira, szerény CPU-terhelést kapunk. A másik dolog, amiben parancsnyelvünk egyedülálló, a memóriába töltött

parancsfájlok kezelése. Hogy a parancsfájlok töltése során fellépő késlekedést vagy akadályoztatást megelőzzük, a memóriában az összes parancsfájl egy virtuális gép (Virtual Machine azaz VM) memóriaszerkezetébe töltődik be, és itt értelmeződik. Ha meg akarjuk változtatni a parancsfájlt, egy teljesen új VM-példány jön létre, amely az új parancsfájlt tartalmazza. A jelenleg futó hívások a régi VM-ben futnak tovább, az új hívók számára azonban már az új VM lesz elérhető. Amikor az utolsó régi hívás is lezárult, a teljes régi VM törlődik. Így még akkor is elérhetjük a százszázalékos rendelkezésre állást, ha a szolgáltatások időközben módosulnak. Végül, mivel C++ parancsnyelv-rendszert készítettünk, a parancsértelmező közvetlen osztálykiegészítését is lehetővé tettük, hogy a későbbiekben új parancsképessegekkel lehessen bővíteni. Így bárki készíthet a nyelvből egy az adott alkalmazáshoz, vagy (ha szükséges) az adott telefonos vezérlőhöz levezetett nyelvváltozatot, egyszerűen csak a hagyományos C++-osztálykiterjesztést kell felhasználnia. A kiegészítés az eredeti nyelvből levezethető.

Bár a kiszolgáló parancsnyelv teljes telefonos szolgáltatások létrehozását is támogatja, nem általános célú programozási nyelvnek terveztük, és külső programkönyvtárakkal sem lehet egybeépíteni, ahogyan a hagyományos nyelveket. A nem blokkosítottág előfeltétele, hogy minden a kiszolgálóhoz készülő kiegészítés magas szinten testreszabott kell legyen. Ehelyett mi egy általános célú parancsfájlkészítő módszert szeretnünk volna, ami adatbázisokkal és más rendszererőforrásokkal is képes kapcsolatot tartani. Ezért ACS (Adjunct Communication Server) Projektünk létrehozásakor lényegében ugyanazt a modellt választottuk, mint amit a webkiszolgálók esetében alkalmaznak. Kiszolgálónk TGI-modellje nagyon hasonló a webkiszolgálók CGI-rendszerének működéséhez. A TGI külön folyamatként indul el, majd az adatokat környezeti változókon keresztül juttatja a telefonhívóhoz. Azért használunk környezeti változókat parancssori paraméterátadás helyett, mert így megelőzhetjük az adatok utáni kémkedést (hiszen azok lényeges dolgokat is tartalmazhatnak, például hitelkártyaszámokat), amelyek egyébként egy egyszerű ps paranccsal kiolvashatóak lennének. A TGI-folyamat szabványos kimeneten keresztül kapcsolódik a kiszolgálóhoz, ahol a TGI-alkalmazás által készített

kimenetet kiszolgáló parancsok meghívására használjuk fel. Ezek a parancsok különböző feladatokat láthatnak el: visszatérő értékeket állíthatnak be, amelyeket például egy adatbázis-keresés eredményeképpen kapunk, vagy új folyamatokat hívhatnak meg, ilyen lehet például a kifelé menő tárcsázás. Ahelyett, hogy minden egyes egyidejű hívásnak

beépíteni. Mint már korábban megjegyeztük, az egyetlen követelmény az, hogy a rendszergazda kiszolgálóoldali parancsfájlokat legyen képes készíteni, hangfelvételeket tudjon lejátszani és rögzíteni, illetve legyen valamennyi gyakorlata valamilyen általános eszköz (például Perl) használatában. Kiszolgálónk egy jellemző alkalmazása



külön átjárót biztosítanánk, minden TGI-átjáróhoz egy-egy folyamatkészletet tartunk fenn, így azokat korlátozott erőforrásoknak tekinthetjük. Feltételezzük, hogy az átjáró végrehajtási ideje a teljes hívási időnek csak kis százalékát jelenti, így a gyors TGI-índítás érdekében hatékony megoldást jelent egy kis számú folyamatkészlet állandó készenlétben tartása. Ez a módszer segít elkerülni a torlódást, például ha az összes hívó éppen egyszerre érkezik a TGI-hez. Ezekkel az alapvető eszközökkel végül lehetővé vált a hangválaszokat adó (voice response) alkalmazások elkészítése. Amikor a rendszer végre működni kezdett, első telefonos kiszolgálónk már üzleti környezetben futott az Open Source Telecom és más cégek gépein. Ez a széles körű felhasználás más vállalkozási-gazdasági tényezők mellett részben annak volt köszönhető, hogy annyira könnyen lehet új alkalmazás-szolgáltatásokat készíteni, illetve telefonos alkalmazásokat a kiszolgáló alá

valahogy úgy nézhet ki, miként azt az *ábrán* láthatjuk (elérhető az www.linuxvilag.hu/Bayonne címen), ez nevezetesen a *playrec* parancsfájl. A parancsfájl bemutatja a jelenlegi parancsnyelv különböző jellegzetességeit, az érvényességi köröket, az eseményvezérlést, amelyeket (nevesített parancsfájl-hivatkozások alatt futva) létrehozzák az interaktív telefonos alkalmazás feldolgozásához szükséges logikai láncot. A 2. listában (ami a www.linuxvilag.hu/Bayonne címen érhető el), a Perl felhasználására mutatunk be egy példát a *TGI.pm* modul és a *tgi-getdval.pl* Perl parancsfájl segítségével.

Hogyan lett az ACS-ből GNU Bayonne?

Mint korábban már említettük, ezeket a célokat az első ilyen telefonos alkalmazáskiszolgálóval már nagyjából két évvel ezelőtt elértük, ezt Adjunct Communication Servernek, röviden ACS-nek neveztük. Sajnos az ACS névgondokkal küszködött, és számos olyan levelet

kaptam különböző emberektől, akik arra mutattak rá, hogy az ACS nevet már több másik projekt is használja, például az AI's Circuit Simulator. Ez természetesen baj volt.

Ugyanakkor az ACS szerkezetének korlátai is kezdtek megmutatkozni. Először is azon az elképzelésen alapult, hogy a kiszolgálót közvetlenül a telefonos kártyához kell kapcsolni, valahogy úgy, ahogyan az XFree86 3 köti az X-kiszolgálót az adott videokártya-családhhoz. Ez azt jelenti, hogy minden egyes kártyacsaládnak külön kiszolgálót kell fordítani, ezáltal sok kód szükségtelenül többszörözött meg szerepelni. Úgy döntöttem, hogy a teljes kiszolgálómagot az alapoktól kezdve újraírom, és ezt néhány héten belül be is tudtam fejezni. Az első dolog, amit megterveztem, a bővítmények (plugin) támogatása volt, egy kicsit más elképzelés alapján, ahogyan azt a legtöbb ember korábban csinálta.

A bővítmény általában egy kis objektumfájl, amelyet egy ismert szimbólum vagy szerkezet alapján dinamikusán töltünk be, és azután a betöltött fájl átvizsgálásával könnyedén megtalálhatjuk. Így a modult az *dlopen* utasítással bárki megnyithatja, majd a *dlsym* segítségével megkeresheti az adott szimbólumot, hogy aztán a modulon belül függvényeket hívhasson meg.

Én egy másik megoldást alkalmaztam: az új kiszolgálót úgy alakítottam ki, hogy a saját szimbólumait exportálja. A kiszolgáló egy létrehozókkal (constructor) ellátott alapsztályhalmazzal rendelkezik, amely a rendszerleíró szerkezetet készíti fel a használatra. A betölthető modulokat C++ nyelven, származtatott osztályként lehet megírni, melyek alapsztálya a kiszolgálóban van megadva, és ezek levezetett osztályainak statikus objektumokat tehet elérhetővé. Amikor a modult az *dlopen*-nel betöltjük, a statikus objektumok létrehozói (constructor) önműködően meghívódnak, az alapsztályból a kiszolgálólenyomatra (server image) mutató hivatkozások pedig önműködően feloldásra kerülnek (resolved). A kiszolgálólenyomatban tárolt alapsztály a létrehozó által önműködően meghívódik és bejegyzi a modulobjektumot. Így aztán az *dlopen* nemcsak, hogy betölti a modult, de egyben a használatra való felkészítést is elvégzi – mindezt egyetlen műveletben. Továbbá néhány dolog, ami korábban az ACS része volt, most külön csomagba került. Ekkor vált a GNU ccScript és a

GNU ccAudio külön osztálykönyvtárrá, minthogy ezek jelképezték az ACS-ben korábban is megtalálható igen hasznos parancsnyelvmotort és a hangfeldolgozó szolgáltatásokat. Különösen az iránt érdeklődtünk, hogyan tudnánk a parancsnyelvet más kiszolgálókban használni, amely a GNUCOMM része lehetne.

A GNU ccAudio bizonyítottan hasznos, általános célú hangfeldolgozó könyvtár. Felhasználható egyszerű és duplafrekvenciás hangok előkészítésére, amelyeket később a memóriából lejátszhatunk, továbbá több bemeneti fájlból csomagolt, állandó hosszúságú keretből (fixed-length frames) álló hangfelvételt képes összeállítani (a végükön szünettel), ahogyan azt a legtöbb DSP (digital signal processor; digitális jelfeldolgozó) bemenete megköveteli. Ez a képesség kiemeli a többi hangfeldolgozó könyvtár sorából, amelyek ilyen mutatóvonalakra rendszerint nem képesek. A legjobb az lenne, ha a GNU ccAudio-t sikerülne teljes értékű, általános célú hangfeldolgozó keretrendszerrel fejleszteni, amely egyben kiszolgálóalapú DSP-szerű feldolgozásra is alkalmas.

Megvolt tehát az új kiszolgálónk, mindössze a neve hiányzott. Mivel valami egyedi és mások által valószínűleg még nem használt nevet szerettünk volna, úgy döntöttünk, nem használunk újabb rövidítést. Helyette, mivel a kiszolgáló tulajdonképpen híd a számítógép és a telefonos világ közt, logikusan egy hídmetaforát választottunk. De melyik hídra essen a választásunk?

Kézenfekvő lenne a Brooklyn híd. Csak hogy ez meglehetősen gyakran használt név, és negatív mellékzöngéi vannak, így nem tűnt jó választásnak. Ugyanígy a Golden Gate név is meglehetősen felkapott, és többnyire az IBM Java kezdeményezésével kapcsolják össze. Szóba került a Tacoma Narrows is mint lehetőség, de figyelembe véve, hogy önmegsemmisítéséről volt híres, úgy gondoltuk, ezt inkább ejtjük, meghagyjuk a washingtoni kereskedelmi gyártóknak. Létezik viszont egy híd nem messze innen, New Jersey-től, a Bayonne. Valószínűleg nem nagyon hallott róla senki, tehát a neve is kevésbé használt.

A ma és a holnap

2002 nyara a GNU Bayonne 1.0-s változatának megjelenését jelzi. Jelenleg a GNU Bayonne nemcsak, hogy része a GNU Projektnek, de csomagját több GNU/Linux terjesztés alaprészeként is megtalálhatjuk, így a GNU/Debian és Mandrake-terjesztésekben is. Minthogy

a célunk a telefonos alkalmazásszolgáltatások elérhetőségének minél teljesebb körben történő elterjesztése a szabad program fejlesztői közt, ez igen kellemes előrelépést jelent.

A GNU Bayonne-t máris széles körben használják szerte a világon. A felhasználók köre az orosz üzleti távközlési szolgáltatóktól az USA állami és szövetségi kormányzati ügynökségeiig terjed, és számos olyan vállalkozást találhatunk köztük, amelyek különleges hangszolgáltatásokat biztosítani tudó egybekötött webhelyeket vagy hangüzenet-szolgáltatáshoz hasonló vállalkozási alkalmazásokat kerestek.

A GNU Bayonne nem önmagában létezik, hanem egy sokkal nagyobb metaprojekt, a GNUCOMM része. A GNUCOMM célja a jelenlegi és a következő nemzedékbeli telefonos hálózatok részére telefonos szolgáltatás nyújtása, szabadon felhasználható (free license) program segítségével. Ezeket a szolgáltatásokat a következőképpen határozhatjuk meg

1. olyan szolgáltatások, amelyek az asztali felhasználókkal tartják a kapcsolatot, ilyenek például a telefonszámok tárcsázására képes címtárak és a telefonprogram-alkalmazások (soft phone applications);
2. olyan szolgáltatások, amelyek telefonkapcsolatokat valósítanak meg, ilyen például az IPSwitch GNU softswitch projekt és a GNU oSIP proxykiszolgáló;
3. olyan szolgáltatások, amelyek átjáróként működnek a jelenlegi és a következő nemzedékbeli telefonos hálózatok közt, ilyen a troll és a tűzfalal védett telefonos hálózatok proxykiszolgálói, például az Ogre;
4. valós idejű adatbázis-adatforgalmazó rendszerek, mint például a preViking Infotel és a BayonneDB;
5. hangalapú alkalmazásszolgáltatások, mint amilyeneket a GNU Bayonne-n keresztül is megvalósíthatunk.

Még a GNU Bayonne 1.0 befejezése előtt, 2001 vége felé már elkezdődött a GNU Bayonne örökösének fejlesztése is. Ez az örökös számos olyan szerkezeti választást megpróbál leegyszerűsíteni, amelyek a fejlesztések kezdeti szakaszában kerültek a projektbe, remélve, hogy az új megoldások által a GNU Bayonne-t könnyebb lesz majd átdolgozni és egyszerűsíteni. A terv kiválasztása és a kezdeti tervezés nagy része 2001 végén két nap alatt ment végbe, mialatt a londoni találkozáson összejöttem a preViking telefonos kiszolgálót tervező emberekkel. Néhány ilyen változtatás indokolta a

preViking projekt közvetlen bevonását a GNU Bayonne fejlesztésébe.

Az egyik legnagyobb kihívást a jelenlegi GNU Bayonne kiszolgáló fejlesztésében a telefonos kártyák moduljainak elkészítése jelenti. Ezek gyakran minden egyes meghajtóhoz komoly fejlesztést követelnek meg, és a kód is sokszor szerepel többszörözötten. A GNU Bayonne 2 ezt a kihívást úgy oldja meg, hogy a vezérléshoz használt állapotgépet (state machine) a kiszolgálómagba helyezi át, majd C++-osztálymeghatározásokon keresztül teljes mértékben elfedi. Ezáltal a meghajtók egyszerűbbé válnak, ugyanakkor lehetővé teszi számunkra, hogy egyetlen kódalapból többféle kiszolgálót hozzunk létre.

A GNU Bayonne 2 másik újdonsága a távközlési cégeknek készített linuxos megoldások sokkal közvetlenebb támogatása. Azaz elődjével szemben ez az új megoldás folyamatosan élő kiszolgáló esetén is képes a szolgáltatásból kapukat kivenni és visszahelyezni, ami lehetővé teszi, hogy a kártyákat menet közben telepítsük (hot-plug) vagy cseréljük (hot-swap). A távközlési osztályú kiszolgálókon a rendszermag figyelmeztetést küld a változás tényéről, az alkalmazásszolgáltatások pedig figyelhetik és válaszolhatnak ezekre az eseményekre. A GNU Bayonne 2-t úgy terveztük, hogy vegye figyelembe ezt a „értesítő” elvet az általa irányított erőforrások kezelése során.

Végül a GNU Bayonne 2 a kezdetektől úgy készül, hogy több módon is kihasználja az XML nyújtotta előnyöket. Beállításnyelvként saját XML-nyelvjárást használ, webszolgáltatásként is működik, amely egyszerre képes az éppen futó GNU Bayonne pillanatnyi állapotát XML formában megadni, illetve az XMLRPC-t támogatni. Ez illeszkedik ahhoz az elképzelésünkhöz, amely szerint a telefonos kiszolgálókat webkiszolgálókkal építhetjük egybe, s amely jól mutatja, hogyan képzeljük el a projekt további útját.

Linux Journal 2002. augusztus, 100. szám



David Sugar

több mint 20 éve fejleszt szabad programokat. A GNU Project számos csomagjának elsődleges szerzője, köztük a GNU Bayonne-é is. Ő az Open Source Telecom alapítója és a DotGNU-t kormányzó bizottság elnöke.

Ahol pingvin még sosem járt

Tux megérkezett az Északi-sarkra, így a beágyazott Linux hódítása valóban világméretűvé válik.

Siker-e a beágyazott Linux? A megjelenő újabb és újabb eszközök láttán azt kell mondanunk, igen. Ebben a hónapban megtudjuk, hogy Tux hogyan hódította meg a sarkvidéket. Lehet, hogy a Mikulást kereste? Mindenféle aprósággal is foglalkozunk, amelyek ugyancsak beágyazott Linuxot futtatnak.

A Mikulás nyomában

Az Amerikai Óceán- és Légkörkutató Intézet (NOAA) eljuttatta az első webkamerát az Északi-sarkra – a készülék beágyazott Linuxot futtat. Telepítésére 2002. április 28-án került sor. A gép naponta négy képet rögzít, amelyek a NOAA honlapján tekinthetők meg (☞ http://www.arctic.noaa.gov/gallery_np.html).

A webkamera képein követni lehet az Északi-sark hőretegének és időjárásának alakulását, valamint a NOAA Csendes-óceáni Környezeti Laboratórium kihelyezett műszereinek állapotát – állítja **James Overland**, a NOAA Északi-sark Tervezetének vezetője.

Bár a kamera arra is képes, hogy másodpercenként egy képkockányi frissítéssel mozgóképet továbbítson, a NOAA az idő túlnyomó részében kikapcsolva tartja, hogy takarékoskodjon a napelemekkel töltött akkumulátor energiájával. A kamera minden nap négyszer felébred, az Iridium-hálózat alacsony pályán keringő műholdjainak segítségével felhívja a NOAA kiszolgálóját, majd 2400 baud sebességgel, PPP-n keresztül továbbítja a legújabb képeket. A kamera egy NetCam-típusú készülék,



1. kép A NetCam az Északi-sarkon



2. kép A StarDot Technologies NetCam kamerája

ami a StarDot Technologies, egy húszfős kaliforniai cég terméke. A StarDot 1996-ban készítette első webkameráját, majd 1997-ben egy egyedi kameravezérlő lapkát fejlesztettek ki, amivel több mint 65 ezer kamerát adtak el.

A NetCam beágyazott számítógépén μ CLinux (☞ <http://www.uclinux.org>) fut, lelke egy 54 MHz órajelű Motorola ColdFire processzor 8 MB RAM-mal és 2 MB flashmemóriával. Két RS-232-es soros kapuval rendelkezik, egy 10 Mb/s sebességű ethernetcsatlóval, I2C soros busszal, valamint négy digitális ki- és bemeneti csatlakozóval. Az eszköz Boa (☞ <http://www.boa.org>) alapú webkiszolgálót futtat, így webböngészővel a világ bármely pontjáról elérhető, feltéve, hogy ethernet-, modemes vagy vezeték nélküli internetkapcsolattal látjuk el. ☞ <http://www.stardot-tech.com>

Mindenféle finomság beágyazott Linuxszal snom 100 VoIP Phone

A snom VoIP telefon különféle nyílt telefonszabványokat támogat: SIP, H.323/H.450 és Asterisk, illetve ismeri a HTTP-, a TAPI- és a LDAP-protokollokat is. A cég szerint a snom 100 segítségével létesített IP-alapú kapcsolatok hangminősége semmivel sem marad el a hagyományos ISDN telefonokétól. Emellett a 128×64 képpontos grafikus LCD-n megjelenő webböngésző jellegű felület segítségével a hívások indítása, a távoli felügyelet és a beállítások módosítása is kényelmesen elvégezhető. Az IP-alapú

kapcsolatok kezelése mellett a készülék hagyományos telefonos szolgáltatásokkal is bír: hívástartásra, a hívások várakoztatására, hívásátadásra, hívásátírányításra, a hívó fél azonosítására is alkalmas és így tovább. A snom 100 belsejében egy 50 MHz-es Motorola MPC855T PowerQUICC Integrated Communications processzor és 16 MB RAM található. Saját fejlesztésű



3. kép snom 100 VoIP Phone

beágyazott operációs rendszere a 2.4.18-es Linux-rendszermagra épül, a snom fejlesztői saját illesztőprogramjaikat és könyvtáraikat a grafikus LCD-hez házon belül készítették, illetve külön beágyazott HTTP-kiszolgálót írtak.

☞ http://www.snomag.de/snom100_en.htm

hippo Internet Phone

A hippo Internet Phone ugyanúgy néz ki, mint egy teljesen átlagos asztali telefon, ám a telefonhívásokat ethernet-alapú hálózati kapcsolaton és normál telefonvonalon – PPP-protokollon, a felhasználó internetkapcsolatát használva – egyaránt el tudja intézni. A beszélgetések nem hagyományos analóg jeleként, hanem digitálisan, az Interneten keresztül továbbítódnak. A kapcsolat túlsó végén a hívást egyaránt fogadhatja egy másik internetes telefon – akár PC-alapú, akár célkészülék –, de egy normál, hagyományos telefonkészülék, ide értve a mobiltelefonokat is. Használatával tekintélyes mértékben csökkenthető a telefonszámla, hiszen nem kell megfizetni a távolsági hívások árát. A hippo Internet Phone egy 4×20 karakteres



Hippio Internet Phone telefon

LCD-vel, 12 gombos telefonos billentyűzettel, hat szolgáltatásgombbal, valamint egy kézibeszélővel bír,

amely a hagyományos telefonoknál megszokott hangjelzéseket is képes szimulálni. A hippo Internet Phone telefonjának belsejében egy 48 MHz-es Motorola MPC850/823 PowerPC-típusú, a teljes rendszert egyetlen lapkán hordozó beágyazott számítógép található, amely 16 MB DRAM-mal gazdálkodik, és a MontaVista Hard Hat Linuxán alapuló operációs rendszert futtat.
 ↪ <http://www.hippoinc.com>

SONICblue Rio Central

Kiváló hangminőségű, az otthoni hifitornyokba illeszkedő egység, amely 40 GB-os merevlemezén 650 CD-nyi zene tárolására képes. Mondhatni egyszerű, akár egy CD-lejátszó, ám okos, mint egy PC – mármint egy Linuxot futtató PC. A készülék önálló hangrendszerként is üzemeltethető, és jelentősen megkönnyíti a zenezámok tárolását – „csak be kell helyezni a lemezeket, a többit elvégzi magától” –, illetve szinte



5. kép SONICblue Rio Central

végtelen számú, ízlés szerint összeállított lejátszási listát hozhatunk vele létre. Nem szabad elfeledkezni a nagy méretű kijelzőről, az ötletes kezelői felületről, illetve azokról a fejlett keresési lehetőségekről, amelyek segítségével pillanatok alatt rátalálhatunk a kívánt zenére. A készülék arra is képes, hogy korábbi szokásaink alapján ajánlatokat állítson össze. Belül egy 206 MHz órajelű Intel StrongARM processzor gazdálkodik 16 MB memóriával, operációs rendszere pedig Debian/ARM-alapú Linux. A Rio Central összetettebb hálózat központjaként is szolgálhat, így több Rio Receiver vékony ügyfél kiszolgálására képes – ezek egyébként szintén beágyazott Linuxot futtatnak – HomePNA vagy

ethernethálózaton keresztül. USB-kapcsolaton át Rio hordozható MP3-lejátszóra is áttölthetünk róla számokat. A készülék külső USB-ethernetcsatló segítségével kezelni tudja a széles sávú kapcsolatokat, az ilyen kapcsolattal nem rendelkező felhasználókra gondolva beépített 56 Kb/s sebességű modemmel is, illetve 10 Mb/s sebességű HomePNA-csatlakozással rendelkezik.
 ↪ <http://www.sonicblue.com>

Cyclades Device Server

A Cyclades TS100 nagy teljesítményű, ugyanakkor kis méretű készülék, amelyet különféle soros eszközök TCP/IP-hálózatra való csatlakoztatására használhatunk. Jellemző alkalmazási területei: az ipari automatizálás, a sávon kívüli felügyelet, a gépesített értékesítés, illetve az egyéb soros felülettel rendelkező eszközök hálózatba kötése. A készülék



Cyclades TS100

10/100 Mb sebességű ethernet és RS-232/RS-485 soros kapukkal rendelkezik, így használatával hagyományos, soros felülettel rendelkező gépek és műszerek is széles sávú TCP/IP-hálózatra kapcsolhatók. Versenytársaival ellentétben – amelyek saját fejlesztésű programokon alapulnak – a TS100-on beágyazott Linux és egyéb nyílt alkalmazások futnak, így testreszabása is könnyen megoldható. Noha alig nagyobb, mint egy csomag kártya – 7,1×8,6×3 cm méretű –, a TS100 két mikroprocesszort is tartalmaz, ugyanis belsejébe a Motorola MPC855T jelzésű „PowerQUICC Integrated Communications Processor” lapkája került. Az MPC855T kétmagú, a teljes rendszert egyetlen lapkán tartalmazó processzor, amely egyrészt egy 50 MHz órajelű PowerPC processzort, valamint egy külön RISC-motort foglal magában, az utóbbi az adatátvitellel kapcsolatos feladatokat vállalja át. 16 MB SDRAM és egy 4 MB-os flash-lemez található benne, bekapcsoláskor a belső programnak a memóriába történő kibontása az utóbbiból történik. A készülék beágyazott Linux-rendszere a 2.2.14-es rendszermagon alapul, de fut rajta néhány nyílt forrású segédprogram is, mint a GoAhead webkiszolgáló, amellyel a készülék webalapú beállítását és felügyeletét oldhatjuk meg, továbbá

Portslave, OpenSSH 3.1, crontab, BusyBox, net-tools, rsyncm és egyebek. A Cyclades a MontaVista Hard Hat Linux 1.2-es változatával kezdett el dolgozni, a rendszer átalakításait maguk végezték.
 ↪ <http://www.cyclades.com>

Linksys Wireless Presentation Gateway

Érdekes, beágyazott Linuxot futtató készülék, amelynek segítségével a vezeték nélküli hálózati kapcsolattal rendelkező mobilgépek tulajdonosai bemu-



Linksys Wireless Presentation Gateway

tatókat vagy egyéb adatokat jeleníthetnek meg. Ezt VGA-szabvány szerinti eszközön, például multimédia-kivetítőn, képernyőn vagy LCD-n tehetik meg, méghozzá úgy, hogy nem kell

a megjelenítő eszközhöz csatlakoztatni az egyes gépeket fizikailag nem kell a megjelenítő eszközhöz csatlakoztatni. A WPG11 lehetővé teszi, hogy a felhasználók a megjelenítő vezérlését pillanatok alatt átadják egymásnak. Minden felhasználó saját, egyedi kóddal rendelkezik, amellyel elérheti és vezérelheti az eszközt. Mivel a kábelezésre és a telepítésre nem kell külön időt szánni, a vezeték nélküli kapcsolattal rendelkező felhasználók könnyedén, a kódokat megadva adogathatják körbe a megjelenítő használati jogát. Így például egy tárgyalás résztvevői valós időben, mondandójuk ismertetése közben azonnal vizuális háttéranyagot szolgáltathatnak a többieknek.

↪ <http://www.linksys.com>

Egyre többen vannak...

Ha érdekelnek a beágyazott Linuxot futtató újdonságok, látogass el a LinuxDevices.com „Embedded Linux Cool Devices Quick Reference Guide” részlegébe, amely a
 ↪ <http://www.linuxdevices.com/articles/AT4936596231.html> címen található.

Linux Journal 2002. augusztus, 100. szám



Rick Lehrbaum
 (rick@linuxdevices.com) hozta létre a LinuxDevices.com „beágyazott Linuxok portálját”, amely nemrég tagja lett a ZDNet Linux Resource Centernek. Rick 1979 óta foglalkozik beágyazott rendszerek fejlesztésével.

Csomagtelepítési trükkök

Miután a felhasználók feltelepítették életük első Linuxát, előbb-utóbb eljutnak arra a szintre, hogy új alkalmazással szeretnék rendszerüket gyarapítani. Most a csomagtelepítés rejtelseibe avatjuk be olvasóinkat.

Egy terjesztés nemcsak magát az operációs rendszert (tehát a rendszermagot, azaz a rendszer lelkét jelentő programot), hanem több ezer hozzávaló alkalmazást, programozási könyvtárakat (libraries), illetve egyéb bővítményeket is tartalmaz. Ezeket a könnyebb telepíthetőség kedvéért csomagokba (packages) szervezzük.

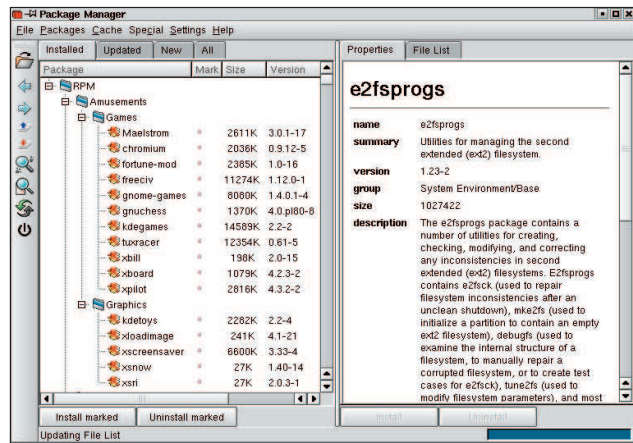
Réges-régen, az első Linux-változatok megjelenésekor ezek a csomagok pusztán tömörített állományok voltak. A telepítéskor meg kellett mondani, hogy melyeket szeretnénk felpakolni, és a rendszer szépen egyesével kicsomagolta őket a merevlemezünkre. Az idő múlásával ez a rendszer nem volt hatékony, ezért megszülettek az úgynevezett *csomagkezelők* (package managers). A csomagkezelők adatbázissal bírnak, amelyben a rendszerünkre feltelepített összes csomag szerepel. Ebben az adatbázisban nemcsak a csomagok nevei, hanem a hozzájuk tartozó állományok is fel vannak tüntetve, így egy csomag eltávolítása nem okozhat gondot. Ha új csomagot telepítünk a rendszerünkre, természetesen az is bekerül az adatbázisba.

A csomagkezelők nemcsak egy csomag telepítését, illetve eltávolítását könnyítik meg, hanem a csomagok frissítését is leegyszerűsítik.

Egy csomag egyébként több egyszerű tömörített állománynál. Minden csomag tudja magáról azt is, hogy milyen további csomagokra van szükség ahhoz, hogy az adott alkalmazás vagy bővítmény gond nélkül működjön. Az ablakkezelőknek például szükségük van a grafikus rendszerre, a KDE alá írt alkalmazásoknak pedig a Qt-könyvtárakra. Ezeket hívjuk a csomag *függőségeinek* (dependencies).

Új csomag telepítésekor a csomagkezelő mindig ellenőrzi a csomag függőségét, azaz azt, hogy az akadálytalan működéshez megvannak-e az elengedhetetlen csomagok. Ha nincsenek, hibaüzenetet kapunk, amely kiírja, hogy milyen további csomagokat kell felraknunk. Általában van rá a mód, hogy a csomagkezelőt rávegyük, hagyja figyelmen kívül a függőségi gondokat, és mindenképpen telepítse az adott csomagot. Ebben az esetben azonban számolnunk kell azzal, hogy az adott alkalmazás egyáltalán nem fog elindulni.

Egy csomag azt is tudja önmagáról, hogy melyek azok a csomagok, amelyekkel ütközik. Ez azt jelenti, hogy egy rendszerben bizonyos csomagok nem férnek meg egymással: vagy azért, mert akadályozhatják egymás működését, vagy mert mind a ketten ugyanazt a feladatot látják el, amelyet csak egyvalaki végezhet el. Efféle összeütközések felhasználói alkalmazásoknál ritkán fordulnak elő (hiszen semmi sem zárja ki például azt, hogy kétféle szövegszerkesztővel is rendelkezünk), hasonló gondok leginkább programozási könyvtáraknál szoktak jelentkezni. Ilyen esetekben a felhasználónak el kell döntenie, hogy a két csomag közül melyiket szeretné használni. A csomagok két parancsfájl tartalmaznak: egy telepítőt és egy eltávolítót. Az első a csomag a telepítés, a második pedig az eltávolítás után fut le önműködően. Ezeknek a parancsfájlok-



A Kpackage segítségével könnyen kezelhetjük csomagjainkat

nak a feladata általában bizonyos beállítások megváltoztatása vagy különböző könyvtárak és egyéb állományok létrehozása, amelyek az alkalmazás gördülékeny használatához szükségesek. Napjainkban a két legelterjedtebb csomagkezelő a Red Hat RPM-je, illetve a Debian által fejlesztett DPKG. A Linux-változatok többsége főleg ezek valamelyikét használja.

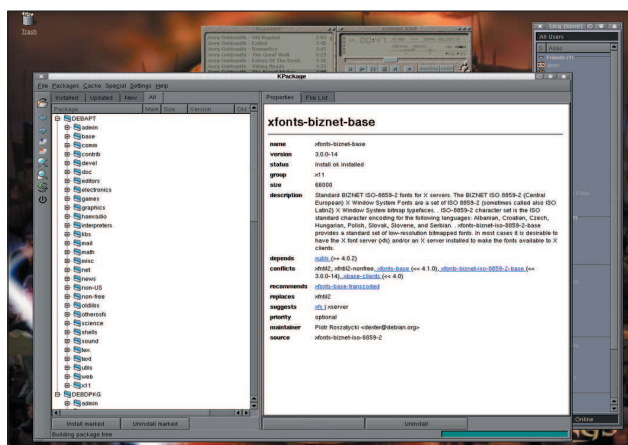
A SuSE és a Mandrake, illetve a kezdőbb felhasználókat megcélzó terjesztések elsősorban az RPM-et támogatják. Rengeteg RPM-et támogató grafikus csomagkezelő napvilágot látott, amelyek leegyszerűsítik a csomagok telepítését, illetve törlését. Sőt egyes terjesztések saját fejlesztésű csomagkezelővel is rendelkeznek, amely megkönnyíti a telepítő CD-n lévő csomagok böngészését, felpakolását.

Mi most ezekkel nem kívánunk foglalkozni, hiszen „ahány terjesztés, annyi szokás”. Inkább azt nézzük meg, hogy akkor mi a teendő, ha egy Internetről letöltött vagy a telepítő CD-n lévő RPM-csomagot szeretnénk rendszerünkre telepíteni az rpm parancs segítségével.

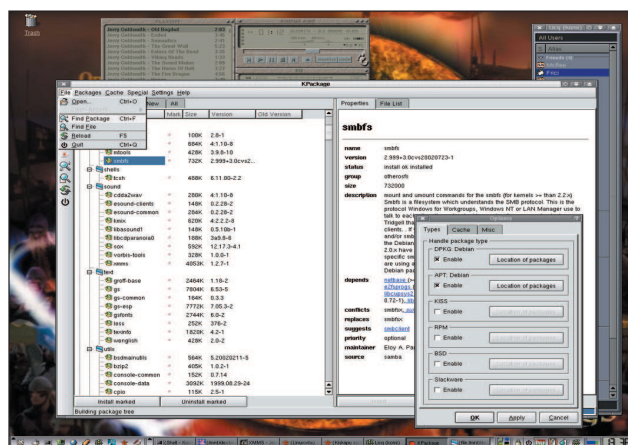
Az RPM-csomagokat roppant egyszerű felismerni, ugyanis .rpm kiterjesztésűek. Mind a Debian-, mind az RPM-csomagok állománynevei három részből épülnek fel: az első rész a csomag neve, a második a változatszáma, a harmadik pedig azt mondja meg, hogy az adott csomag melyik processzorcsaládra lett lefordítva (bár ez nem minden esetben van feltüntetve).

Egy RPM-csomagot a következő módon telepíthetünk: rpm -i csomagnév-vel vagy rpm --install csomagnév-vel. Ha nem akarjuk felpakolni, csupán az épségét szeretnénk ellenőrizni, a -V kapcsolót használjuk! Ha az adott csomag rendszerünkön már fent van, és újabb változatúra kívánjuk lecserélni, akkor a -U vagy a --upgrade használható. Egy csomag rpm -e csomagnév utasítással távolítható el.

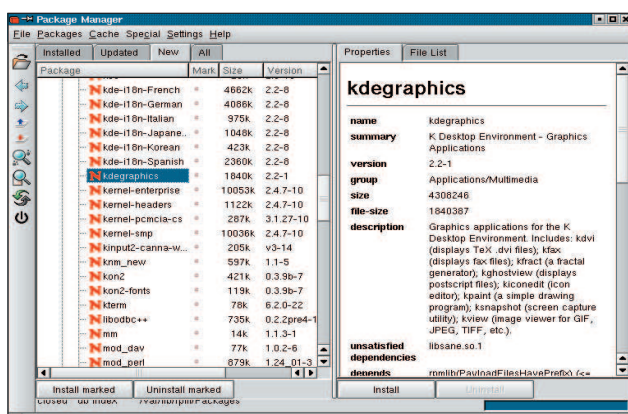
A Debian-csomagok kiterjesztése .deb. A csomagkezelést a dpkg utasítással tudjuk végrehajtani. A csomagtelepítés ennél a



A csomagkategoriók listája



Az egyes csomagfajták kezelésének engedélyezése



A Kpackage a már rendszerünkön lévő csomagok kezelését teszi egyszerűbbé

változatnál is a `-i` kapcsoló használatával zajlik. A már telepített csomagokat a `-r`-rel törölhetjük le (2002. március, 43. oldal: Csomagkezelés Debian/GNU alatt, Mihez kezdünk, ha már áll a rendszer, de valamit telepíteni, frissíteni szeretnénk? – Szy György; 2002. augusztus, 44. oldal: Csomagkezelés GNU/Debian alatt (2. rész); Az aptitude további lehetőségei – Szy György) Érdemes pár szót szólnunk az Alien nevű programról, amelynek segítségével RPM-csomagból Debian-csomagot készíthetünk, és fordítva. Az Alien minden Linux-változatban fellelhető. Ha egy csomagból Debian-csomagot szeretnénk készíteni, a `--to-deb`, ha RPM-csomagot, akkor pedig a `--to-rpm` kapcsolókat kell használnunk. Az Alien `tar.gz` csomagok készítésére is képes, ebben az esetben a `--to-tgz` kapcsolót kell segítségül hívunk. Ha például Debian-csomagot szeretnénk átalakítani RPM-mé, a következő utasítást kell kiadnunk: `alien --to-rpm csomagnév.deb`. Ha már a Debian csomagkezelését érintettük, illik pár szót szólni az APT-ről is. Ez is a `dpkg-re` épül, és a csomagok „beszerzését”, illetve frissítését könnyíti meg. Csak egy parancs kiadásába kerül, és már le is szedi az Internetről a kívánt csomag legfrissebb változatát. De ez még nem minden, egy utasítás segítségével akár a rendszerünkön lévő összes csomagot is frissíthetjük. Az APT egyébként nagyon rugalmas rendszer, mert teljesen mindegy, hogy a csomagok „beszerzési helye” az Interneten lévő egyik tükörzés-e, esetleg a gépbe helyezett Debian CD-n található. Ezeket a forrásokat a `/etc/apt/sources.list` állományban kell sorolnunk. Az APT is rendelkezik adatbázissal, amelyben a telepíthető csomagok nevét függőségükkel együtt tárolja.

Ha egy csomagot szeretnénk telepíteni, az adatbázisból megkeresi a nevét, ellenőrzi a függőséget, letölti (vagy átmásolja a CD-ről), majd meghívja a `dpkg-t`, amely a telepítést intézi. Ezt az adatbázist az `apt-get update` paranccsal frissíthetjük, illetve ha új CD-t szeretnénk az adatbázishoz adni, az `apt-cdrom add` parancsot használjuk. Nagyon fontos, hogyha új forrást tüntetünk fel a `sources.list` állományban, ezt az utasítást akkor is ki kell adnunk. Egy csomag telepítését az `apt-get install csomagnév` módon végezhethetjük. Ha a rendszerünkön lévő csomagokat az adott forráson megtalálható frissebb változatokra szeretnénk frissíteni, az `apt-get upgrade` parancsot kell használnunk. Az Internetet böngészve bármikor ráakadhatunk egy-egy hasznosnak tűnő linuxos alkalmazásra. Ezek sajnos nem mindig érhetőek el Debian- vagy Red Hat-csomagban, általában csak a forrását közlik, `tar.gz` típusú állományokban. Ebben az esetben kénytelenek vagyunk lefordítani, majd saját kezűleg telepíteni. De ez csak első hallásra tűnik ördögös feladatnak. Első dolgunk a kicsomagolás, ami a `tar -xvzf állomány.tar.gz` utasítás kiadásával lehetséges. Lépünk be a létrejött könyvtárba! Az ilyen csomagoknál általában létezik egy `INSTALL` nevű állomány, amelyben az üzembe helyezés menete pontosan le van írva. Mi most az általában használatos utat mutatjuk be, de elképzelhető, hogy egyes csomagoknál a telepítés menete eltérő. Első lépésként adjuk ki a `./configure` utasítást! Ilyenkor a program ellenőrzi, hogy minden a fordításhoz és működéshez szükséges dolog megtalálható-e a rendszeren (például C-fordító). Ha a `configure` gond nélkül lefutott, kezdődhet a fordítás. Ha nem, feladatunk annyival súlyosbodik, hogy nekünk kell kitalálni, milyen csomagok hiányoznak, bár azok a mellékelte leírásban általában fel vannak tüntetve. A fordításhoz semmiféle programozási előismeret nem szükséges, csak a `make` parancsot kell kiadnunk. Ezután rendszergazdaként jelentkezünk be, és térjünk vissza ugyanebbe a könyvtárba, majd a programot a `make install` utasítás segítségével telepítjük fel. Az így telepített programokat csak kézzel tudjuk eltávolítani!

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

© Kiskapu Kft. Minden jog fenntartva

Versenyhelyzet és holtpont nélkül

Sorozatunk előző részéből már kiderült, hogy az operációs rendszer összes szolgáltatása a folyamatkezelésen alapul. Az előző részben végig a folyamatokat és a velük szorosan összefüggő fogalmakat tárgyaltuk, korántsem értünk azonban a témakör végére.

Tessék, még mindig a folyamatoknál tartunk! Nehezen tudunk elszakadni ettől a témától, pedig hány más izgalmas dolgról nem esett még szó, például a fájl- és memóriakezelésről, vagy a beviteli és kiviteli eszközök vezérléséről. Az előző részben azt tárgyaltuk, hogy az operációs rendszerek legalapvetőbb szolgáltatásokat ellátó részei is – például a memóriakezelő vagy az eszközmeghajtók – egy-egy folyamatként, a felhasználói alkalmazásokkal párhuzamosan futnak (igaz, alacsonyabb, gépközelibb szinten). A folyamatkezelés tehát a legfontosabb, ha nem is a legérdekesebb elem az operációs rendszerek életében.

A feladatkezelés tárgyalásakor még nem érintettük az úgynevezett *szálakat*, illetve nem esett szó a folyamatok közötti kapcsolattartás megvalósításáról. Ez utóbbi nagyon fontos, az erőforrás-kezelés ugyanis teljes egészében erre a rendszerre fog épülni.

A szálak

Tegyük fel, hogy barátunktól megkaptuk egy weboldal címét, ahol topmodellek fürdőruha-divatbemutatójáról készült fényképek tömkelege található. A férfi felhasználók többsége az ilyenekre felkapja a fejét, és rögtön be is „izzítja” kedvenc böngészőprogramját.

Az ehhez hasonló oldalak általában rengeteg kis képet tartalmaznak, nem is beszélve az elhelyezett reklámok sokaságáról. Elméletben egy weboldal betöltése a következőképpen zajlana: a böngésző először megkapja a HTML-forrást, amelyben többek között le van írva az oldalon megjelenítendő képek elérési helye. Ezután sorban egymás után letöltjük a képeket a saját gépünkre, majd megjelenítjük az oldalt.

A valóságban ez a módszer nem hatékony. Ahhoz, hogy egy képet letölthessünk a kiszolgálóról, először kapcsolódnunk kell, meg kell várni, amíg a kép teljes egészében lejön, majd bontanunk kell a kapcsolatot. Kis képekről lévén szó a kapcsolatok létrehozása és lebontása csaknem annyi időbe telik, mint maga az adatátvitel, tehát rengeteg idő kárba vész. Több száz kép esetében annyira idegölő lehet ez a sok üresjárat, hogy fürdőruha topmodellek ide vagy oda, a felhasználónak elmegey a kedve az egésztől.

A legjobb megoldás az lenne, ha a letöltéseket *párhuzamosíthatnánk*. Az ilyen és ehhez hasonló nehézségek feloldására – amikor egy folyamaton belül is további párhuzamosításra van szükség – találták ki az úgynevezett *pehelysúlyú folyamatokat*, közismertebb nevükön a *vezérlési szálakat*.

Egy vezérlési szálát úgy is elképzelhetünk, mint utasításokból álló sorozatot. Az első helyen van az első végrehajtandó utasítás, a másodikon a második és így tovább. Minden szálhoz tartozik egy *utasításszámláló*, amely megmondja, hogy éppen hol tartunk az utasítások végrehajtásában.

Nem meglepő, hogy minden folyamatnak legalább egy vezérlési szállal és egy utasításszámlálással rendelkeznie kell. A kor-

szzerű operációs rendszerek azonban azt is megengedik, hogy egy folyamat további vezérlési szálakat hozzon létre.

A szálak sok tekintetben hasonlítanak a folyamatokhoz. Itt is megtalálható a három alapvető állapot (futó, futásra kész, illetve blokkolt). A tárolásukról egy úgynevezett *száltáblázatban* kell gondoskodnunk, amely többek között az utasításszámlálókat tartalmazza, amelyek azt mondják meg, hogy éppen hol függesztettük fel a szálak futását.

A szálak azonban nem folyamatok a folyamatban. Míg ugyanis a folyamatok külön címtartománnyal (memóriaszelettel) rendelkeznek, amelyekbe más folyamatok nem „túrhatnak bele”, addig a szálak közösen osztoznak folyamatuk címtartományán. A szálak megvalósítására két út is kínálkozik. Az első, az operációs rendszer a rendszermag szintjén támogatja, hogy egy folyamat több vezérlési szállal rendelkezzen. A második, a rendszer egyáltalán nem törődik a vezérlési szálakkal, a megvalósítás a felhasználóra van bízva (azaz maga a folyamat fog a saját vezérlési szálainak kezeléséről gondoskodni). Vajon melyik megoldás a nyerő?

Aki úgy gondolja, hogy az első variáció a jobb, azaz a szálak kezelését az operációs rendszerre kell bízni, igaza van. Tegyük fel, hogy van egy folyamatunk két vezérlési szállal, és a szálak kezelése a felhasználói területen történik. Semmi gond sem lesz egészen addig, amíg az egyik szál valami oknál fogva nem blokkolódik (például azért, mert visszajelzésre vár egy eszköztől). Ebben az esetben az ütemező az egész folyamatot blokkolni fogja. Ha azonban az ütemező tudna a szálak létezéséről, akkor egy szál blokkolása esetén az adott folyamat szálai közül kiválasztana egy futásra kész szálat. Ha nincs ilyen, egy másik folyamatban keres futásra kész szálat.

Az sem téved viszont, aki úgy gondolja, hogy a szálak felügyeletének a felhasználó hatáskörébe kell tartoznia. A sebesség szempontjából sokkal hatékonyabb, ha a szálak közötti kapcsolattartást felhasználói szinten végezzük, és nem a rendszer-magon keresztül. Mivel tökéletes megoldás nincs, az operációs rendszerek mind a kétféle módszert egyszerre használják. Sajnos, korántsem ez az egyetlen bonyodalom a szálak kezelésének a kérdésében. Így nehézséget okoz, hogy a folyamat szálai közös adatterületen osztoznak. A legklasszikusabb példa egy rendszerhívás végrehajtása. Az első szál meghív egy rendszerhívást. A Linux esetében ennek sikeréről (vagy kudarcáról) az *errno* nevű globális változó ad felvilágosítást. Mielőtt azonban ellenőrizni tudnánk az *errno* értékét, szálváltás történik, és a második szál is végrehajt egy rendszerhívást, ennek hatására pedig az *errno* értéke megváltozik. Amikor a vezérlés visszakerül az első szála, az már hamis adatokkal fog dolgozni. Ez viszont csak a jéghegy csúcsa. Mi történik akkor, ha az egyik szál látja, hogy nincs elég memória, ezért újabb blokkokat kezd lefoglalni, miközben egy szálváltás következtében a másik szál ugyanígy cselekszik?

A szálak bevezetése nehezen leküzdhető feladatok elé állítja az operációs rendszerek fejlesztőit. Valószínű, hogyha egy meglévő rendszerbe szeretnénk beültetni a vezérlési szálakat, azt nem úszhatjuk meg a rendszer alapjainak újratervezése nélkül.

A folyamatok kapcsolattartása

Az előző részben alaposan kiveséztük a folyamatkezelés egyik legfontosabb elemét, az ütemezést. Most egy másik, nem kevésbé elhanyagolható részt tekintünk át.

A *folyamatok közötti kapcsolattartás* (InterProcess Communication, röviden IPC) megvalósítása is a rendszermagra hárul. Ha felidézünk magunkban az operációs rendszerek elvi felépítését bemutató táblázatot, akkor az IPC-t is valahol a legalsó réteg bugyraiban kell keresnünk. Az ütemező-megszakítás-kezelő réteg tehát újabb feladatot kapott: a folyamatok közötti kapcsolattartás megvalósítását és ellenőrzését.

Az IPC témakörébe nem csak az tartozik, hogy két folyamat miképp tud üzenetet küldeni egymásnak. Az IPC-nek olyan feladatokat is meg kell oldania, mint a versenyhelyzetek feloldása, illetve a folyamatok közötti összehangolás. Hogy ezek pontosan mit jelentenek, az alábbiakból mindjárt ki is derül.

Üzenetküldés

A Linux és Unix-rendszerek a folyamatok közötti kapcsolattartást úgynevezett *üzenetküldéses rendszerrel* valósítják meg. Ehhez mindössze két könyvtári eljárásra van szükség: egyre, amellyel üzenetet továbbíthatunk egy másik folyamatnak; és egy másikra, amellyel fogadhatjuk a nekünk küldött üzeneteket. Ezeket az eljárásokat a folyamatok bármikor meghívhatják. Az üzenetküldéses rendszer előnyeit elsősorban az osztott rendszereknél élvezhetjük. Sorozatunk bevezető részében említettük: az a jól felépített operációs rendszer, amely átültethető akár több gépből összeállított osztott (telepekre) rendszerekre átültethető anélkül, hogy az egészet az alapjaitól fogva újra kellene terveznünk. Az üzenetküldés ebből a szempontból nagyon hatékony, mivel a folyamatok szemszögéből teljesen mellékes, hogy a kapcsolattartásban résztvevő másik folyamat a telep egy másik gépén fut-e vagy sem. Egy gép esetében azonban sokkal gyorsabb megoldásokat is találhatnánk az üzenetküldésnél.

De miként valósítsuk meg az üzenetküldést? A két legkézenfekvőbb megoldás a randevú és a levelesláda. Az első a következőképpen működik: ha egy folyamat üzenetet küld egy másik folyamatnak, akkor egészen addig blokkolódik, amíg azt a másik folyamat nem fogadja. Fordítva is így van. Ha egy folyamat meghívja az „üzenet fogadása” eljárást, mindaddig blokkol, amíg az üzenete meg nem érkezik. Ez a megoldás roppant egyszerű, mivel a rendszernek az üzenetet csak át kell másolnia az egyik folyamattól a másikhoz. Ez a módszer azonban megköveteli, hogy a kapcsolattartásban résztvevő két folyamatnak a futását egymáshoz kell igazítanunk.

A levelesláda ennél sokkal rugalmasabb. Minden folyamatnak fenntartunk egy memóriarekeszt, amely meghatározott számú üzenet ideiglenes tárolására képes. Ha az adott folyamatnak üzenete érkezik, a rendszer a levelesládába teszi, és egészen addig ott is tartja, amíg a folyamat ki nem szedi onnan.

A Unix-rendszerek a levelesláda-módszer egyik egyedi formáját használják, mégpedig a *csővezetékeket* (a pipe-okat). A csővezeték olyan levelesláda, amely az üzeneteket nem egymástól elválasztva, külön, hanem egyben, egy fájlként tárolja. A levelesláda üritésekor a folyamat az összes üzenetet egy darabban fogja megkapni. A folyamatoknak kell gondoskodniuk arról, hogy a fogadó el tudja különíteni egymástól a beérkezett

üzeneteket. Például úgy, hogy mindegyiket egy egyedi karakterrel zárja, vagy megegyezünk az üzenetek egy állandó méretében.

Versenyhelyzetek

Két vagy több folyamat olykor arra kényszerül, hogy közös tárterületen dolgozzon. Nem kell messzire mennünk egy hétköznapi példáért, csak a legközelebbi 12 termes multiplex moziba, ahol a helyfoglalásokat számítógépes adatbázisban tartják nyilván.

Tegyük fel, hogy minden pénztárhoz tartozik egy terminál, amelyeken a foglalásokat be lehet táplálni. A terminálok az



osztott adatbázist tartalmazó számítógéppel össze vannak kapcsolva. A rendszer minden terminálhoz külön folyamatot rendel, tehát a helyfoglalásokat az összes pénztárból egyszerre végezhetjük.

Képzeliük el, hogy egy időben két pénztárban is ugyanazt a helyet szeretnék lefoglalni (például azért, mert a teremben már csak egy hely maradt, és az utolsó pillanatban érkező két vendég ugyanazért a helyért verseng). Nézzük meg lépésenként, hogy mi történik a rendszerben!

Az első terminált kezelő folyamat az adatbázisból ellenőrzi, hogy a lefoglalni kívánt hely valóban szabad-e még. Mivel az, buzgón nekilát lefoglalni, azaz frissíteni az adatbázis megfelelő rekordját. Igen ám, de éppen ebben a pillanatban történik egy óramegszakítás, és az ütemező a másik terminál folyamatának adja át a vezérlést. Ámde ott is áll egy vendég, aki ugyanarra a helyre áhítozik. Ez a folyamat is ellenőrzi tehát, hogy szabad-e az adott hely (tudjuk, hogy szabad, mert a másik folyamat az adatbázis frissítését még nem tudta befejezni), majd lefoglalja azt. Ezután a vezérlés visszakerül az első folyamatra, amely mit sem tud arról, hogy közben valaki a „háta mögött” már megkapta a kérdéses helyet, így ez is lefoglalja ugyanazt. Ilyenkor úgynevezett *versenyhelyzet* alakul ki. Ennek lényege, hogy két (ritkább esetben több) folyamat egy időben ugyanahhoz az egymás között megosztott adathoz szeretne hozzáférni. Ha nem figyelünk arra, hogy abban a pillanatban valaki más is dolgozik az adott adattal, akkor a művelet végeredménye attól függ, hogy melyik folyamat mikor és hogyan futott. Sohasem jósolhatjuk meg tehát, hogy mi lesz egy létrejött versenyhelyzet eredménye.

Minden olyan rendszerben, ahol két vagy több folyamat megosztott adatokkal dolgozik, számolnunk kell a versenyhelyzetek kialakulásával. Vitatkozhatunk arról, hogy az előbb említett mozi példánál mekkora a valószínűsége egy versenyhelyzet kialakulásának. De a számítástechnika nem valószínűség-szá-

mítás. Másrészt pedig egy olyan rendszerben, ahol egy adatbázist egyszerre több százan írnak és olvasnak, és az adatok akár egy másodpercen belül többször is frissülhetnek, a versenyhelyzetek mindennaposá válhatnak. Ha egy rendszerben lehetőség van a versenyhelyzet kialakulására, de ez nem egészséges helyzet, lehetőleg el kell kerülni.

Egy folyamat „élete során” sok mindent olyasmit is tesz, ami nem fenyeget versenyhelyzettel. A belső vagy a saját memória-tartományán végzett számítások nem járnak ezzel a veszéllyel. Amikor azonban egy megosztott memóriarészhez vagy állományhoz nyúl, egyből megváltozik a helyzet. Ezért a folyamatok tevékenységének ezt a részét a folyamat *saját kényes (kritikus) területének* nevezzük.

A versenyhelyzetek elkerülésére a kézenfekvő megoldás a kölcsönös kizárás elve: meg kell tiltanunk, hogy egy időben egynél több folyamat is ugyanahhoz a megosztott dologhoz nyúlhasson. Másképpen szólva el kell érniünk, hogy egyszerre csak egy folyamat „tartózkodhasson” a saját kényes területén. Ennek megvalósítására az egyik legkézenfekvőbb megoldás az, ha valamilyen úton-módon az ütemezést a kényes feladatok idejére felfüggesztjük. Ahogyan sorozatunk előző részében is említettük, az egész ütemezés a megszakításrendszerre épül. Tehát a megszakításokat csak le kell tiltanunk egy erre megfelelő processzorutasítással. A kényes rész után pedig csak „vissza kell kapcsolni” azokat. Na de rendelkezhet egy közös folyamat ekkora hatalommal a számítógépünk felett? Mi történne, ha a saját kényes területén végtelen ciklusba kerülne? – lefagyna ez egész szekció. Ez nyilvánvalóan nem jó megoldás, tehát valami másra lesz szükség.

Próbálkozhatnánk esetleg egy olyan globális változó bevezetésével, amelynek megmondja, van-e valamilyen folyamat a saját kényes területén. A gyakorlatban ez úgy zajlana, hogyha egy folyamat be szeretne lépni a kritikus területre, a változó értékét ellenőriznie kell. Ha az 0, akkor belép, és beállítja 1-re; ha eredetileg nem 0 volt az értéke, akkor folyamatosan ellenőrzi, hogy mikor lesz az értéke ismét 0.

Ezt a folyamatos ellenőrzést *tevékeny várakozásnak* nevezzük, ugyanis a folyamat semmi érdemlegeset nem tesz, mégis zabálja a processzoridőt. Ám ennek a megoldásnak is akad hátulütője: egy kis szerencsétlenség következtében itt is kialakulhatnak versenyhelyzetek. Egy rosszul jött óramegszakítás következtében előfordulhat, hogy egyszerre két folyamat is bekerül a saját kritikus területére. Például az egyik folyamat ellenőrizte, hogy a változó értéke 0, és át akarja állítani 1-re, azonban közben történik egy váltás, és egy másik folyamat belép a saját kritikus területére. Miután a vezérlés visszakerül az előző folyamatra, az nem fogja ismét ellenőrizni a belépés lehetőségét, így egy időben két folyamat dolgozhat a megosztott adatokkal.

Bizonyos szempontból jobb megoldást jelent az úgynevezett *szigorú változtatás*, amikor is a kritikus területre való belépés jogát a folyamatok egymásnak „adogatják”, tehát az A folyamat egészen addig nem léphet be, amíg a B folyamat ki nem lépett, viszont ezután a B folyamat sem léphet vissza mindaddig, amíg az A folyamat be nem fejezte a saját kényes területét. A szigorú változtatás megvéd minket a versenyhelyzetek kialakulásától, csak hogy olyankor nem hatékony, ha a két folyamat sebessége különböző, mivel a lassabb a gyorsabb folyamatot hátráltatni fogja.

Sokféle más megoldást is kitaláltak a tevékeny várakozásra alapozva, amellyel szigorú változtatás nélkül is elkerülhetjük a versenyhelyzeteket. Az alapkérdés azonban még mindig fennáll: a tevékeny várakozás közben sok processzoridő megy

kárba, azonkívül könnyen el lehet képzelni olyan egyedi helyzetet, ahol ez a fajta megoldás nem alkalmazható.

Erre találták ki az *altatás-ébredés rendszerét*. Egy folyamat, ha nem tud belépni a kényes területre, blokkolja magát. Az ébredésről majd a másik folyamat fog gondoskodni, miután kilépett a kritikus területéről.

Vannak azonban olyan esetek, amikor ez sem használható eredményesen (például az úgynevezett összekötött tároló megvalósításához). Erre találták ki a különböző jelzőket (semaphore), figyelőket (monitor) és még sok minden más. A feladat és megoldási módjainak ismertetése azonban meghaladja e cikk kereteit, de a kíváncsibb olvasók az Interneten rendkívül sok anyagot találhatnak erről és az IPC más kérdéseiről (az összekötött tároló, közismertebb nevén a gyártó-fogyasztó kérdésére a legjobb megoldást az üzenetküldés alkalmazása jelenti).

A lényeg az, hogy olyan megoldást találni, amely minden helyzetben megvéd minket a versenyhelyzetektől, nagyon nehéz. De szerencsére sok esetben az üzenetküldés is felhasználható erre a célra.

Filozófusok és holtpontok

Bizonyos esetekben nemcsak a versenyhelyzetek elhárítására kell figyelmet fordítanunk, hanem arra is, hogy bizonyos eseménysorozatok ne következhesse be. Ezt *összehangolásnak* nevezzük. Nézzünk erre egy ma már klasszikussá vált példát, az étkező filozófusok kérdését. Öt filozófus egy kerek asztal körül ül, előttük egy-egy tál spagetti, a tányérok között egy-egy villa. A filozófusok élete meglehetősen sivár, mivel csak gondolkodásból és evésből áll.

Ha egy filozófus sokáig gondolkodik, előbb-utóbb megéhezik, és ennie kell. Igen ám, de a spagetti annyira csúsós, hogy egy villával képtelenség megfogni, ezért a sikeres étkezéshez legalább két villára van szükség. A filozófus csak a tányérja mellett villákat próbálhatja megszerezni, és egyszerre csak egyet.

A feladat egy jó algoritmus megírása az étkezésre, amely soha sem akad el.

Ez nem is olyan egyszerű feladat. Nézzük például legegyszerűbb megoldást, amikor a filozófus először a bal, majd a jobb villát próbálja meg megszerezni. De mi történik, ha az összes filozófus egyszerre éhez meg, és egy időben kaparintja meg a tőle balra lévő villát? Ebben az esetben egyik filozófusnak sem lesz esélye arra, hogy a jobb oldalt megszerezze, és kialakul az úgynevezett *holtpont*.

A holtpontok és a versenyhelyzetek elkerülése is az IPC témakörébe tartozik, még ha ez nem is jár mindig közvetlen adatátvitellel. Nem jó, ha egy rendszerben előfordulhatnak holtpontok és versenyhelyzetek fordulhatnak elő.

Az étkező filozófusokkal tulajdonképpen ugyanaz a gond, mint a korlátozott számú beviteli-kiviteli eszközök vezérlésével, ha a villákat úgy fogjuk fel, mint egy-egy ilyen eszközt. A holtpontok tehát a következő részben is visszaköszönnék, amikor is a különböző eszközök kezelésével foglalkozunk, továbbá néhány olyan megoldást mutatunk be, amelyek segítségével elkerülhetjük a holtpontok kialakulását.

Garzó András

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



Egy kis térinformatika

A TNT programcsomag bemutatása.

Néhány éve térinformatikával foglalkoztam, és akkoriban hívták fel a figyelmemet egy programcsomagra, ami az oktatásban is jól használható. A MicroImages (☞ <http://www.microimages.com>) cég által forgalmazott csomagban TNT néven egy teljes térinformatikai rendszer rejlik. A próbaváltozat próbaadatokkal az ftp-kiszolgálójukról letölthető. Most ennek rövid bemutatására vállalkozom.

Maga a program több felületen is elérhető. A Unix az alapfelület, hiszen a windowsos változat is a TWM ablakkezelőn keresztül tudja megjeleníteni a program különböző ablakait. Ezen belül a rendszer több Unix-változatot támogat, például Sun SPARCstationt, a HP 9000 sorozatú 700 munkaállomást, a Silicon Graphics munkaállomást, az IBM RS/6000-t (Power PC alapú processzor), és a DEC Alpha AXP sorozatot (Digital Unix). Működéséhez a következő követelmények teljesítése szükséges: megjelenítéshez legkevesebb 256 színű képernyő, körülbelül 400 MB merevlemez-terület, minimum 16 MB memória ajánlatos (természetesen ezt az értéket a feldolgozandó adatok mennyiségével arányosan növelni kell).

A PCK közül az Intel 486-alapú vagy AMD, Cyrix CPU matematikai co-processzorral, Pentium vagy ennél újabb ajánlott (Windows NT esetében DEC Alpha AXP processzor is lehetséges). Legkevesebb 16 MB memóriával kell rendelkezniünk, a képernyő-megjelenítésnek legkevesebb 256 színűnek és 640×480-as felbontásúnak kell lennie (ajánlott az 1024×768-as felbontás használata). A rendszer csak PC-n hardverkulcsos, ezért egy párhuzamos vagy soros kapu is szükséges. Egy egér használata is igényeltetik, mert enélkül a rendszert nem lehet használni. Windows 95, 98 vagy NT esetében elkél a CD-ROM-meghajtó, és megközelítőleg 300 MB merevlemez-terület. Minimum kiépítés Macintosh gépeknél a Power Macintosh 60× vagy G3 processzor. A többi feltétel megegyezik az előbb felsoroltakkal. A Linuxszal szemben támasztott követelmények azonosak a PC-nél ismertekkel, erről és a telepítésről a későbbiekben még szó lesz.

Néhány szóban arról, mit is jelent a térinformatika. Feladata általánosságban a térbeli, helyhez köthető adatok ábrázolása, megjelenítése, valamint ezen adatok kiértékelése. Eszköztárába nagyon sok adatgyűjtési módszer, eszköz beletartozik. Néhány ezek közül a teljesség igénye nélkül:

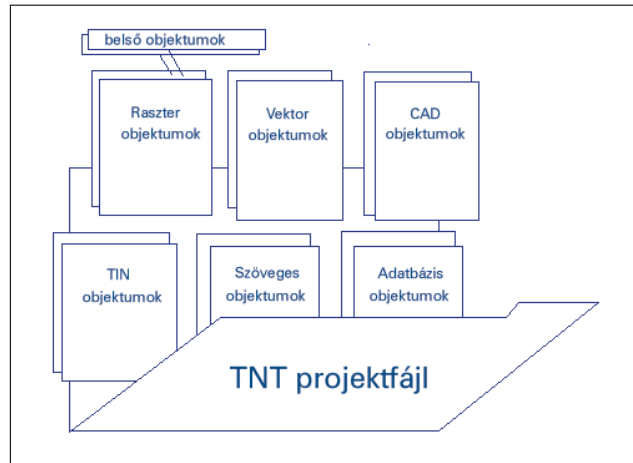
- földi adatgyűjtő eszközök,
- földmérő műszerek, eljárások,
- légifelvételek,
- űrfelvételek.

Az így összegyűjtött adatokat az informatika fejlődésének köszönhetően digitalizálják, és úgynevezett *térképi adatbázisokban* tárolják. Ezekből az adatokból szakértő kezekben nagyszámú információ nyerhető, így számos szakterületen felhasználhatjuk – ilyen például a honvédség, a térképészet, a mezőgazdaság, a bányászat és a geológia.

A késztermékek és az alapadatok igen sokfélék lehetnek. Ezek esetében csak azok magyarázatával foglalkozom, amelyekkel

a TNT-programok is meg tudnak birkózni (az alábbiakban a TNT-programleírást használtam forrásként.)

A TNT-programok adatszerkezetének megértéséhez érdemes megtekinteni a projektfájl felépítését (lásd az ábrát).



A TNT-objektumok belső felépítése

Itt – annak ellenére, hogy a vektoros és raszteres képeket meg tudja jeleníteni és szerkeszteni is lehet – nem mondhatjuk el, hogy a program vegyes adatszerkezetet használ. Ezt tapasztalataim alapján állítom, mert amennyire átláttam a programcsomagot, a vektoros és raszteres képeket külön alprogram tette szerkeszthetővé, és igazából csak a megjelenítésre alkalmas harmadik program volt az, ahol mindkettőt együtt lehetett használni.

A projektfájl elemei

• Raszteres objektumok

A raszteres objektum jellemzője, hogy kétdimenziós: sorokból és oszlopokból épül fel. A legkisebb egység, amit képes megjeleníteni és tárolni tud, a pixel, ami többféle formájú lehet, az esetek nagy részében négyzet alakú (gondoljunk csak az általunk használt raszteres képfarmátumokra, például .jpg, .gif, .tif stb.). Egy raszteres képre jellemző pixeleinek nagysága és tájolása (a derékszögű koordináta-rendszerrel való elfordulásuk). A raszteres adatszerkezetben a pixelek helyzetén kívül fontos adat a pixelhez rendelt és általában megjelölt érték. Ennek az értéknek a nagyságát általában a pixelhez rendelt színnek milyenségével érzékelhetjük. Gondoljunk például egy légifelvételre, ahol a hétköznapi életben ismert fényképek hatásának megfelelően a létrejövő képen az erdők zöldek, a tavak kékek stb. Ha ugyanezt a területet infrakamerával fényképezzük, az adatrögzítőről beolvasott képek színei jócskán eltérhetnek az általunk megszokottaktól (esetleg az erdő mélyvörös színben jelenik meg). Ezen raszterkép mérete általában jóval nagyobb a hasonló tartalmú vektoros

CAD-kép méreténél, hiszen itt jóval több adatot kell tárolni (a vonal esetében ellentétben a vektoros adatszerkezettel nemcsak a két végpontot, hanem a vonal összes köztes pontjának koordinátáit rögzíteni kell.)

A TNT-programok a megjelenítési lehetőségeknek megfelelően a szükséges átalakítások után a különböző színmélységű képeket meg tudják jeleníteni, függetlenül attól, hogy valójában milyen színképzési eljárással lettek tárolva, illetve különböző fedvényeket, hisztogramokat stb. is létre tudunk belőlük hozni.

A raszter által kezelt színmélységek a következők lehetnek: 1 bit (binary), 4 bit, 8 bit, 16 bit, 32 bit vagy 64 bit. Képes 128-bitos rasztert is kezelni, de csak különleges feltételek mellett. A TNT professional által támogatott legnagyobb raszterkép mérete 2 000 000 000×2 000 000 000, pixelenként 1-től 128-bitos adattartalommal. Annak szemléltetésére, hogy ez milyen nagy érték, a leírás Dél-Afrikát említi (amelynek területe 1200×1500 km), amiről a 10m×10 m-es felbontású SPOT műhold felvételei csak 120 000×150 000 pixelrel használnak fel. Dél-Afrika esetében ennek a pixelméretnek egy milliméternél kisebbnek kellene lennie ahhoz, hogy a program legnagyobb teljesítőképességét kihasználhassuk. A szabadon felhasználható TNTlite esetében a legnagyobb méret „csak” 314 368 pixel (1024×256, 512×512, 640×480 pixel).

- **Vektoros objektumok**

A TNT-programok adatfelépítése vektoros és CAD-adatokat különböztet meg, alott mindkettő a vektoros adatszerkezethez tartozik. A következőkben a TNT-leírás általi megkülönböztetést mutatjuk be.

A vektorelemek és objektumok három alapvető elemből épülnek fel: pontokból, vonalakból és poligonokból. A pontok a pontszerű egyetlen mérési ponttal helyhez köthető elemeket ábrázolják, például a hétköznapi életből vett mérési pontok a villanyoszlopok. A vonalak két végponttal határolható egyenesek, mint például egy kerítés. A poligon tulajdonképpen egyenes vonalakkal felépülő vonalsor. Ezeket használhatjuk mondjuk egy határvonal megrajzolására. Érdeemes megemlíteni, hogy sok esetben az ívek, a körívek is ilyen poligonokként kerülnek tárolásra, mivel az ív mért pontjai azok, amelyek meghatározzák a formáját, és az csupán egy adat, hogy a programnak a megjelenítéskor nem egyenesekkel, hanem ívekkel kell őket összekötnie (AutoCAD esetében ezt jól lehet látni). Fontos megemlíteni a vektoros objektumok egyik fontos jellemzőjét, hogy belőlük úgynevezett *topológiai hálózat* hozható létre. Ennek a hálózatnak többek között az feladata, hogy az egyes földterületek egyetlen egységes rajzelemből, egy zárt poligonból álljanak össze. Gondoljunk csak földhivatalokban meglévő ingatlan-nyilvántartásra. Csak a földterülethez kapcsolatos van értelme a tulajdonos adatainak társításának, a határvonal egyes elemeihez külön-külön nem sok értelme lenne. A vektoros adatszerkezet szinte sugallja, hogy réteg-megjelenítésre is szükségünk van. A rétegek feladata az egyes, valamilyen közös tulajdonság szempontjából azonos vektorelemek kezelhetőségének megvalósítása (például a víznek, az erdős területeknek egy-egy saját egyedi megjelenítési formával kell rendelkezniük).

A TNT által CAD-objektumnak hívott rajzelemek több vektoros elemből épülnek fel. A TNT-programok többek közt a következő elemek rajzolására képesek: pont, vonal, téglalap, poligon, kör, húr, ék, szöveg.

A TNT Professional esetében a projektfájlban tárolható

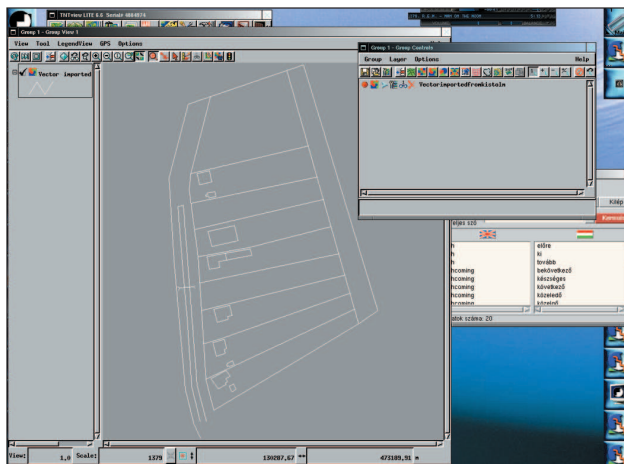
legnagyobb CAD-objektum 2 000 000 000 rajzelemet tartalmazhat. A szabadon letölthető TNTlite esetében a CAD-objektumok száma legfeljebb 500 rajzelem lehet.

Egyéb adatok, késztermékek

Ebben a részben azokat az adattároló, adatmegjelenítő eszközöket próbálom bemutatni, amelyek az elemzéseknek a végeredményei, de akár az alapjai is lehetnek.

- **TIN-objektumok**

A TIN vagy Triangulated Irregular Network jelentése szabálytalan háromszöghálózat. A TIN-objektumok négyféle rajzelemből épülhetnek fel: pontokból, élekből, háromszögekből és héjből. Azok számára, akik nem igazán tudják, mi is ez, álljon itt egy kis magyarázat: a domborzatnak vannak olyan elemei, amelyek adatait valamilyen adatgyűjtő eszközzel rögzítve könnyedén létre tudunk hozni egy domborzatmodellt. Leegyszerűsítve: ha a pontokat csak a legközelebbi lévő szomszédaival kötjük össze, a terepfelülethez nagyon hasonló háromszögekből álló térbeli felület hozható létre. Példaként lássuk a TIN-modellt mutató ábrát (lásd a 2. képet).



1. kép Egy ITR-ben készített térképészlet .dxf állománya beolvasás után

A TIN-objektumok az elemeknél annyival többek, hogy itt a pontokat csomópontokként tároljuk, a vonalakat pedig poligonokként.

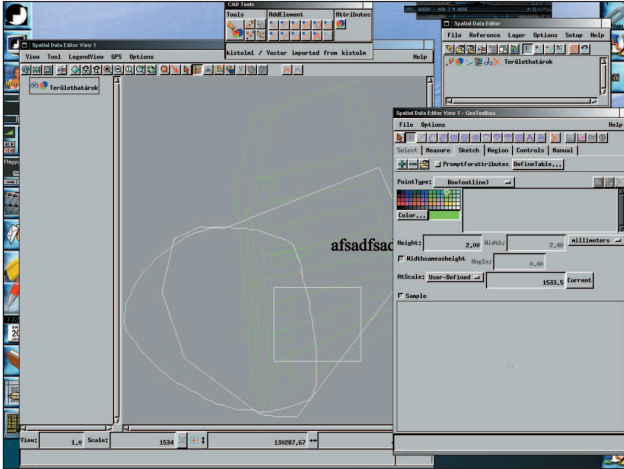
A TNT professional esetében az egyszerű TIN-objektumok legnagyobb száma a projektfájlban belül 2 000 000 000 háromszög, 2 000 000 000 él és 2 000 000 000 pont. A TNTlite esetében ez csak 1500 pont.

- **Szöveges objektumok**

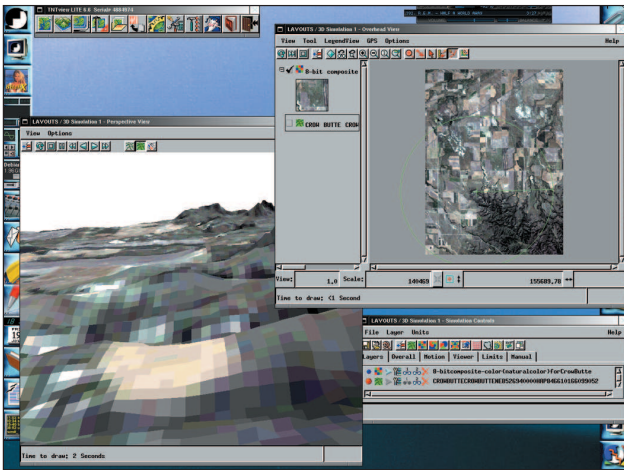
Ide tartoznak a különféle feliratok és a leíró adatok (attribútumadatok) egy része. A felhasznált rajzok – legyenek azok vektoros, raszteres, a képi elemek XY (esetleg Z mint magasság) adatain túl egy-egy egyéb leíró jellemzőt is hozzájuk csatol. Ezeket az adatokat a rendszer általában nem a képi állományon belül, hanem egy külön adatbázisban tárolja. Az adatok tartalmazhatják az objektum azonosítóját, a rajzelemek stílusát stb. Például egy földterület esetében a helyrajzi számot egy külön adatbázisban tároljuk, ami az adatbázisban az adott földterület azonosítójaként szerepel.

- **Adatbázis-objektumok**

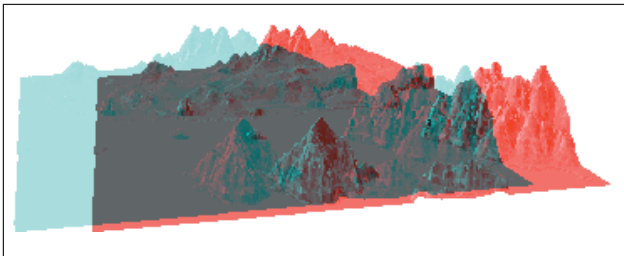
Az adatbázis-objektumok az előbbi leíró adatokhoz kapcsolódnak, hiszen sok esetben azokat valamilyen adatbázisban



2. kép Az előbbi állomány szerkesztés közben

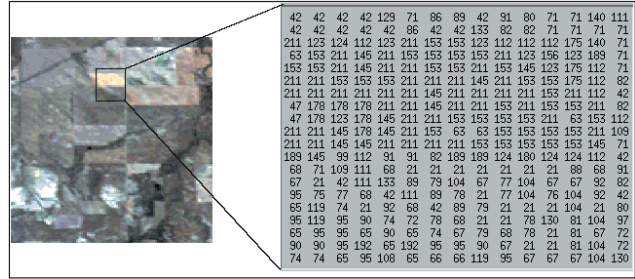


3. kép A bemutató állományok között találtam ezt a 3D-s rácstra alakított műholdképet



4. kép Talán mindenki emlékszik a piros-kék lencsés 3D-s képet adó szemüvegre

tároljuk. A képi adatokhoz kapcsolódó adatbázisadatokat az egyes raszter-, vektor-, CAD- vagy TIN-objektumokhoz rendelhetjük hozzá. Kétféle adatkapcsolat hozható létre. A külső adattárolás esetében a projekt fájlhoz egy külső fájlt csatolunk, és abban tároljuk az adatokat. Ezen belül ismét két lehetőség van. Az első esetben az adatok egy külső adatbázisfájlban vannak tárolva, így többek közt ASCII szöveges, DBASE, INFO adatbázis, Mapinfo leírófájl RBASE formátumban. Ebben az esetben az adatbázis sorainak adatai közvetlenül a hozzájuk kapcsolódó sorokhoz, rajzelemhez kapcsolódnak. A másik eset, amit talán jobban lehet alkalmazni – hiszen a csatolható adatok mennyiségi



5. kép Példa a raszterobjektumokra

korlátja nem olyan erős –, az ODBC használata. Ebben az esetben több DBMS-program adattábláihoz kapcsolódhatunk, például dBase-hez és Oracle-hez.

A TNT Professional esetén a külső adatbázis mérete akkora lehet, amekkorát a rendszer elbírná. Például a dBASE IV fájlak kisebbnek kell lenniük kétmilliárd bajtnál, ami több mint egymilliárd rekordot tesz ki. A TNTlite esetében ez a szám nagyságrendekkel kisebb, csupán 1500 rekord. A belső adatbázis esetén az adatbázis a TNT projekt fájlján belül kerül tárolásra. Ekkor természetesen külső, már meglévő adatbázisokból is importálhatunk adatokat, így dBASE III+ és IV, INFO, R:BASE adatbázisokból, ODBC-ből, Tydac SPANS leíró, illetve vesszővel határolt szöveges fájljokból. E feladatok ellátásért többek közt a TNTmips nevű programrész felelős.

A programról

Sajnos a programcsomag bemutatásakor erősen válogatnom kell, hiszen a program rengeteg olyan szolgáltatással rendelkezik, amit csak egy térképész vagy térinformatikus tud értékelni. Emiatt inkább csak a megemlítés szintjén sorolnám azokat az adatfeldolgozó lehetőségeket, amelyeket a program a magáénak tudhat:

- GPS- (Global Positioning System) adatok beolvasásának a lehetősége.
- a CartoScriptTM, a TNT térképészeti parancsnyelv lehetővé teszi a szimbólumok és egyéb rajzelemek egyszerű és gyors kezelését, például a vonalak típusának együttes módosítását.
- Jól testreszabható topológiaépítő szolgáltatás.
- 3D-s domborzatmodellek, TIN-modellek.
- Tematikus térképkészítési lehetőség, például éves csapadék-térkép megynkénti részletességgel.
- Rasztertranszformációs lehetőség: a lapolvasóval beolvasott kép földrajzi koordinátákhoz való igazítása.
- Felületmodellek készítése, elemzése, vízgyűjtőterület meghatározása.
- Raszterképmozaik, képelemzési lehetőségek, hisztogramok készítése.
- Lekérdezőépítő a csatolt adatbázisokhoz.
- Meglehetősen sok vetületi és mértékegységrendszert támogat.
- Mozgóképek, film is létrehozható a segítségével, mintha a terület felett repülnék, vagy egy dombtetőről néznénk körül.

A program telepítése

Mint említettem, én csak a próbaváltozatot töltöttem le. Telepítése meglehetősen egyszerű feladat volt: a letöltött tar.gz fájlok kicsomagoltam a saját könyvtáramba. A program könyvtárán belül nem találtam telepítőprogramot, hanem csak a már lefordított futtatható fájlokat. Futtatáskor a programok nálam

a *libstdc++-so.2.8* fájlt keresték. Mivel ennek jelenleg a *libstdc++-so.3*, *libstdc++-so.4* változatai voltak fenn, fel kellett telepítenem egy korábbi változatot, ami együttműködött a keresett változattal. A *libstdc++-2.9-glibc2.1.deb* által tartalmazott *libstdc++-2-libc6.1-1-2.9.0.so* fájl bizonyult megfelelőnek. Ezek után a program mindenféle hiba nélkül működött.

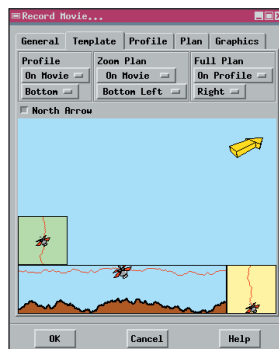
A program részei, feladataik

A telepítési könyvtárban több futtatható fájl van, némelyikük egy másik futásához szükséges, a többség azonban saját önálló grafikus ablakot indít el. Most lássuk őket feladatuk rövid leírásával!

- *tntview*: talán ez a program rendelkezik a legtöbb szolgáltatással. Ebben az ablakban új 2D-s, illetve 3D-s projektfájl-adatsortot tudunk létrehozni, új nézeti ablakot tudunk létrehozni, valamelyik meglévő projektről másolatot tudunk készíteni, több projektet is meg bírunk nyitni. Itt található a külső adatbeolvasási lehetőség, a parancsfájl-építő, az eszköztár-beállító és a szolgáltatásbeállító ablak, valamint a súgóeszközök indítója is.
- *tntdisp*: az előbbi program egyszerűsített változatát jeleníti meg.
- *tntmips*: a beolvasott adatok raszteres és vektoros kiértékelése ebben a programban végezhető el. A *Display* menüben rengeteg átalakítást segítő menüpontot találunk. Ajánlott a *Display/GettingStarted* pont menüinek megtekintése.
- *tntmenu*: az előbbi szolgáltatás menüpontjait jeleníti meg.
- *tntatlas* (a startatlással azonos): a rajzok megjelenítését, az atlasfájlok (előre meghatározott nézetű képek) megtekintését, a különféle nézetek előállítását segíti.
- *sdedit*: a vektoros és a CAD-állományok szerkesztését teszi lehetővé.
- *georef*: az RGB raszteres képfájlok átalakítását segíti. Számos átalakítást ismer.
- *featmap*: a raszteres képfájlok szerkesztését teszi lehetővé, és a területek osztályozását segíti.

A próba tapasztalatai

A programcsomag néhány szolgáltatását kipróbáltam. Kíváncsi voltam a sebességre, az átláthatóságra, a kezelőeszközök elérhetőségére, egyáltalán: arra, hogy mennyire felhasználóbarát. Az első próbálkozás során lakóhelyem egy nagyobb területének beolvasására voltam kíváncsi. Természetesen a teljes beolvasás lehetetlen volt, mivel a letölthető programváltozat adatmennyiségét a beolvasandó terület jócskán meghaladta, a program azonban a betöltési folyamatot ilyenkor is végrehajtja, és csak ennek végeztével jelzi a határoltságot. A beolvasandó adatokat még régebben az ITR-ből (interaktív térképszerkesztő rendszer, magyar fejlesztésű DOS-os program) *.DXF* állományba mentettem. Az állomány mérete közel 14 MB volt, a tartalmazott pontok száma csaknem 400 ezer (ez csak a pontok száma az összekötő vonalak, feliratok, jelkulcsok nélkül). A beolvasást egy 600-ról 900-ra felpörgetett Celeron processzoros, 256 MB memóriával rendelkező gépen, Debian SID operációs rendszerrel végeztem. A külső vektoros adat beolvasásáról tudnunk kell, hogy ez a program, ha úgy van beállítva, önműködően megpróbálja létrehozni a topológiát. A térkép többek közt tartalmazta az épületeket, a melléképületeket, a földterületeket, az utakat és a vízfolyásokat is. Az adatok beolvasása, átalakítása nem teljes processzorkihasználtság mellett 4,5 órát vett igénybe, mivel időnként az OpenOffice



6. kép Készítsünk a 3D-s domborzatmodell alapján filmet!

doljunk csak bele abba, hogy egy adott földterületből és a rajta átmenő útból három területi egység jön létre, attól függetlenül, hogy ezek külön rétegben helyezkednek el.

Egy másik település pontjait és vonalait úgy töröltem, hogy az elemek száma kisebb legyen a próbaváltozat által maximálisan használhatóknál. Az adatok beolvasása egy másodperc alatt elkészült (lásd, a 4. képet).

Ezek után az így elkészült állományt az *sedit* programmal beolvasítottam, és a rajzszerkesztési lehetőségekkel próbálkoztam. Szerintem viszonylag rövid idő alatt meg lehet szokni a kezelőfelület sajátosságait. Egyetlen furcsasággal találkoztam: a billentyűzet többek között a BACKSPACE, DELETE billentyűkre és a számbillentyűzetre nem válaszol. Ez valószínűleg a többi Unix-változatról történő átprogramozás eredménye (törléskor a DELETE használata helyett a szöveget az egérrel jelöltem ki, és a CTRL+C billentyűkombinációval töröltem.) Az ehhez kapcsolódó képen jól látható a rajzeszközök beállítására szolgáló GeoToolbox (jobb oldalt lent), illetve a rétegkezelő ablak (jobb oldalt fent). Ez utóbbiban sorrendben a réteg elemeinek kiválaszthatóságát, beállításait, részletes adatait, a rétegműveleteket, a láthatóságot és a törlését lehetett kiválasztani. A CAD *Tools* ablak (középen fent) – mint látható – eléggé sokféle új rajzelem hozzáadását és beállítását teszi lehetővé.

A programhoz letöltöttem az ugyancsak elérhető próbaállományokat is. Az egyik ezek közül a 3DSIM könyvtár, amiben egy 3D-s domborzatmodell volt látható az alapjául szolgáló légifelvétellel együtt. A domborzatmodell érdekessége többek között az volt, hogy „körülnézhettem benne”, vagyis a képen bal oldalon látható domborzatmodellel a kamerát körbe tudtam forgatni.

Összegzés

Összességében a program jól kezelhető, rengeteg adatkezelési lehetőséget tartalmaz és jól leírt. Ezért mindenkinek a figyelmébe ajánlom, egy kipróbálás erejéig, és térinformatika-oktatási, -tanulási célokra és a gyengített lehetőségekkel rendelkező változat is mindenképpen jól használható.



Tóth Béla (tothb1@freemail.hu)

Nős, két gyermek büszke atyja. Dolgozott földmérőként, majd térinformatikus szakmérnöki képesítést szerzett. Egyaránt otthonosan mozog a CAD és a térinformatikai programokban, valamint a DOS- és Windows-alkalmazásokban.

Legkedveltebb elfoglaltsága már két és fél éve a Linux.

Legyen saját IRC-botod!

Egy kis Perl-tudás, kevés hálózati ismeret és egy csipetnyi RFC-olvasgatás a biztos receptje egy egyéni IRC-bot készítésének.

Az IRC (Internet Relay Chat) egy olyan TCP/IP-alapú protokoll, amely hálózaton keresztül valós idejű párbeszédet folytatását teszi lehetővé. Ha két ügyfél csevegni szeretne egymással, mindketten csatlakoznak egy-egy kiszolgálóhoz (nem feltétlenül ugyanahhoz). Legalább az egyiküknek tudnia kell a másik azonosítóját (becenévét, azaz nickjét), hogy a beszélgetés megkezdődhessen. Az első ügyfél üzenete elküldéséhez azt a kiszolgálót használja, amelyikhez kapcsolódott. Amikor a kiszolgáló megkapja a kézbesítendő üzenetet, felállít egy útvonalat, amelyen kiszolgálóról kiszolgálóra terjed az üzenet, amíg csak el nem éri azt, amelyikhez a másik ügyfél kapcsolódott. Így ő megkapja az üzenetet, és ugyanezzel a módszerrel válaszolhat rá. Habár az IRC egyszerű szöveges alapú protokoll, a biztonság érdekében több dologra is ügyel. Egyrészt az ügyfél csak akkor kapja meg az üzenetet, ha a címzettek között van (ugyanazt az üzenetet több becenévre is el lehet küldeni), másrészt egy kiszolgáló egy adott ügyfél megtalálásához csak egyetlenegy másik kiszolgálónak adhatja tovább az üzenetet. Ez jó, hiszen valamiféle biztonságot nyújt, ha egy IRC-hálózaton a kiszolgálók közül remélhetőleg nem az összes kapja meg az üzenetedet. Ez a felépítés ugyanakkor gátja is a rendszernek. A csatornákon, ahol nemegyszer 50–60 ügyfél is eszmét cserél, gyakran tapasztalható, hogy nem minden ügyfélhez jut el időben az üzenet, és a többiek úgy látják, hogy már megtárgyalt kérdésekről beszél (lag). Az is előfordul, hogy ha a két (földrajzi értelemben vett) hely közti összeköttetést biztosító kiszolgáló leáll, az IRC-hálózat két félre szakad. Legyenek az ügyfelek ugyanazon a csatornán, mindenki csak azt látja, aki vele azonos oldalon van (split). Ilyenkor időbe telik, amíg új útvonalat találva ismét elérhetik egymást.

Mi az a bot?

Egy IRC-bot ahhoz hasonló ügyfél, amelyet csevegésre is használasz. A különbség, hogy a botot valamilyen sokszor ismétlődő feladatra tartják, és nem szükséges előtte embernek ülnie. A `#linux.hu` csatornán létezik egy Rozsdas nevű bot, ami a két kérdőjellel kezdődő sorokat keresi, és ha ilyet lát, a kérdőjeleket követő kifejezésre valamilyen leírást ad. Akad olyan csatorna, ahol egy botnak nincs más dolga, mint feltennie egy kérdést, majd a válaszra várni. Ha valaki a megadott időn belül helyesen válaszol, egy pontot kap. Ezután – vagy az idő lejártával – a bot felteszi a következő kérdést. Egy játék megadott számú kérdésből áll, és a végén az győz, akinek a legtöbb pontja van. Ezekhez hasonló és ezer másik feladat képzelhető el egy bothoz. Természetesen én egy egyszerűbbet írtam, ami adott időközönként felolvas egy az állományból véletlenszerűen kiválasztott sort. Gyakorlásnak megfelelő, viszont hasznavehetetlen. Ezért arra kérek mindenkit, ezt a botot ne küldje fel egyik kiszolgálóra sem! Egyrészt egyes kiszolgálók kifejezetten megtiltják a botok jelenlétét, másrészt nincs olyan csatorna, ahol szívesen látnának egy ilyen botot. A legjobb, ha feltelepítesz egy saját IRC-kiszolgálót, és azon próbálgatsz.

Így gyorsabb a fejlesztés, és biztosan senkinek nem okozol kellemetlenséget. Egy kiszolgáló telepítése és beállítása túlmutat e cikk keretein, ezért kérek, vedd a fáradságot, és válassz egyet a terjesztésedhez megfelelők közül. A Debian használók az csomag telepítésével az ircd két változatához is hozzájuthatnak. Ők ircd néven az Undernet hivatalos kiszolgálóját érhetik el, míg a dancer egy kisebb hálózatokra tervezett változat.

A protokollról nagyvonalakban

Az IRC-hálózaton mind a kiszolgálók, mind az ügyfelek üzenetekkel tartják a kapcsolatot. Egy üzenet a következőképpen fest:

```
[ ':' <elitag> <SZ K Z> ] <parancs>
↳ <param0terek> <crlf>
```

Az előtag (prefix) használata nem kötelező. Az előtag jelenlétét a kettőspont (:) mutatja. A kettőspont és az előtag között nem lehet szóköz. Az előtag az üzenet feladójának azonosítását szolgálja, ha elmarad, az IP-csomag küldője lesz a feladó. Elsősorban kiszolgálók használják; ha ügyfél használja, és nem önmagát tünteti fel feladóként, az első kiszolgáló, amelyikhez eljut az üzenet, visszajelzés nélkül megsemmisíti azt. Én sem használtam sehol az előtagot.

A parancs szabványos IRC-parancs. Habár a parancsnevekben az IRC-alkalmazások elvileg nem különböztetik meg a kis- és nagybetűket, érdemes csupa nagybetűvel írunk.

A kapcsolók a parancstól függenek. Nem tartalmazhatnak szóközt, legfeljebb az utolsó, ebben az esetben kettőspontot kell tennünk elé.

A sort sortörés-kocsivissza zárja, de ha jobban tetszik, a legtöbb nyelvben ez is megteszi: `\n\r`.

Mielőtt ügyfél csatlakozhatna egy csatornához, először be kell jegyeznie magát a kiszolgálónál. Ez két paranccsal történik: USER és NICK.

```
USER <felhasználó név> <m d> <használatlan>
↳ <val di_név>
```

Az első kapcsoló a felhasználói név. A linuxos IRC-ügyfelek legtöbbször annak a felhasználónak a nevét küldik el, akinek a nevében a program fut. Te mint programozó akármit megadhatsz. Arra figyelj, hogy bizonyos felhasználóneveket egyes csatornákon szűrnek. A `#linux.hu`-ra például root felhasználónévvel nem lehet felmenni (illetve fel lehet, de egy bot kidob).

A mód egy szám, ami az ügyfél különleges tulajdonságait írja le. Itt lehet többek között azt beállítani, hogy láthatatlan legyél. Ez azt jelenti, hogy a csatorna tagjainak listájában nem jelensz meg.

A harmadik kapcsoló jelenleg használaton kívül van, ezért csillag (*) karaktert szokás helyette írni.

A valódi név egy tetszőleges karakterlánc. Ide többnyire

a felhasználói névhez tartozó teljes nevet szokták ide írni, ám ez változó. Mivel szóközt is tartalmazhat, kettősponttal kell kezdődnie.

NICK <becenőv>

Ezzel a kapcsolóval határozhatod meg a becenevedet. Bejegyzés után bármelyik kiszolgáló küldhet a botnak egy ping-et. Az IRC-kiszolgálók ugyanis nem ICMP-üzenetekkel ellenőrzik, hogy egy ügyfél fent van-e még a hálózaton, hanem belső ping-gel. Ha nem készíted fel a botot, hogy ezekre válaszoljon, a kiszolgáló bontani fogja a kapcsolatot. A ping parancsot kapcsolóként egy azonosító követi. Az erre válaszoló ügyfélnek egy pong parancsot kell küldenie, kapcsolóként ugyanazzal az azonosítóval.

A PRIVMSG parancsot használhatod, ha hozzá akarsz szólni a beszélgetéshez, és te is ilyeneket fogsz kapni, ha valaki veled beszélget.

PRIVMSG <c mzett> <sz veg>

Ha a címzett egy becenév, akkor magánbeszélgetés indul közted és a címzett között. Ha kettős kereszttel (#) kezdődik, egy csatornát jelöl, és a csatorna összes tagja megkapja az üzenetet. Ha dollárjellel (\$) indul, a dollárjel után álló kiszolgálóhoz csatlakozó összes ügyfél olvasni fogja a szöveget. Helyettesítő karakterek mindhárom esetben használhatók (*, ?). Egy megszorítás létezik, miszerint a kettős kereszttel, illetve dollárjellel kezdődő nevekben legalább egy pontnak (.) szerepelnie kell, és az utolsó pontot nem követheti helyettesítő karakter. Ezt azért találták ki, hogy például a \$*-ra ne lehessen üzenetet küldeni, ezt ugyanis a világ összes IRC-hálózatot használója megkapná.

A szöveg természetesen tartalmazhat szóközt, így kettősponttal kell kezdődnie.

Ha további részletekre vagy kíváncsi a protokollal kapcsolatban, vagy más parancsok is érdekelnek, ajánlom figyelmedbe a kapcsolódó RFC-eket: RFC1459 és RFC2812 (lásd a 39. CD Magazin/Bot könyvtárban).

Egy perlbeli megvalósítás

Ezek után lássunk egy példát! Az én kis IRC-botom nem csinál mást, csak csatlakozik a megadott csatornára, és bizonyos időközönként felolvas egy véletlenszerűen kiválasztott sort egy állományból. Az Interneten ezeket a botokat nagyon idegesítőnek tartják, és végleg meg tudod vele utáltatni magadat. Mindenki elismeri a programozói teljesítményt, azonban nem győzöm elégszer hangsúlyozni: ne használd ezt a botot nyilvános IRC-kiszolgálón! Ha nagyon tetszik, írd egy olyat, amelyik hasznos. És feltétlenül kérdezd meg a kiszolgáló rendszergazdáját, hogy felteheted-e! Tudomásom szerint az *irc.extra.hu* egyáltalán nem engedélyezi a botok használatát a kiszolgálón. Az *elte.irc.hu* napi üzenetéből (motd, message of the day) az derül ki, hogy két hétig elviselik a botokat, utána mindenkit nyomtatékosan megkérnek, hogy távolítsa el kis kedvencét.

A *sote.irc.hu*-ra lehet botot vinni, de erről egy e-mailben (*ircadm@sote.irc.hu*) értesíteni kell a kiszolgáló üzemeltetőjét. (A listát lásd a 39. CD Magazin/Bot könyvtárban).

Mivel feltételezem, hogy programoztál már Perlben, csak nagy vonalakban magyarázom el a program működését. A \$\
különleges változót a Perl minden kimenő adat végéhez hozzáfűzi. Így a print utasítások végére nem kell mindig a kötelező \
\
-t odaírni. A \$| szintén a kimenő adatokra

vonatkozik: nem használ átmeneti tárat az íráshoz, minden adatot azonnal kiír.

A \$nick a becenevet, a \$channel a csatorna nevét, a \$wait pedig egy számot tartalmaz, amely másodpercekben fejezi ki azt az időt, amennyinek két bementett idézet között legalább el kell telnie.

A helyi géphez a 6667-es kapun, TCP/IP-protokollon keresztül kapcsolódik. Ha nem sikerült, a program leáll.

Láthatóan semmiféle különleges módot nem adok a botnak, és a teljes neve Zebulon Master.

A @stuff tömböt több dologra is használni fogom (lásd később). Amennyiben a \$joined 0, még nem kapcsolódott, ha 1, az előbbi iterációban kapcsolódott, amennyiben 2, már rég fent van a csatornán. A \$last tárolja az utolsó kiíratás dátumát az 1970. január 1-je óta eltelt másodpercek számában. A \$i általános ciklusváltozó.

A while ciklus minden lefutásakor beolvass egy sort a \$irc foglalatról.

Az első, hogy ellenőrzi, megpingelték-e. Ha igen, a ping kapcsolóként megadott azonosítóval egy pong-ot küld. Ha nem ping volt, és épp most csatlakozott a csatornához, egy üdvözlő szöveget küld. Ha valaki megszólította, és azt mondta, „aludj”, kilép.

Ha letelt a megadott idő, a megadott állományból kiválaszt egy sort és elküldi.

Amennyiben még nem csatlakozott, megteszi.

A ciklusból történő kilépés után a Quit paranccsal búcsút mond a kiszolgálónak, és ha nem szakította volna meg a kapcsolatot, akkor a bot megteszi.

Bothibák

Ez egy nagyon kezdetleges bot, és több gond is akad vele. Nem kezeli az olyan váratlan eseményeket, ha a becenév foglalt, vagy nem engedik be a csatornára (miért is ne engednék). Továbbá a saját kitzűött célját sem látja el tisztességesen. Nem 12 másodpercenként olvas fel egy sort az állományból. Ha senki sem szólal meg a csatornán, és nem kap üzenetet, nem fog beszélni. Ez azért van, mert a Perl <> műveletjelét használtam olvasásra. Ez a megadott leíróról egy sort olvas be. Mivel a leíró nyitva van (a foglalat nem zárult le), nincs állományvége (EOF), és addig nem tér vissza, amíg üzenetet nem kap. Valójában az olvasást a sysread() függvényvel kellene megoldani, ahogy más Perlben írt botok is teszik, de nem akartam túlbonyolítani a példát. Léteznek bővítmények az IRC-protokollhoz, mint például a CTCP (Client To Client Protocol). Ezzel a kiszolgálót megkerülve lehet adatot kérni egy ügyféltől. Egy ügyfél a másiktól a használt ügyfél típusát, a rendszeridőt és más hasonló adatot kérdezhet le. Ha a mi botunk kap egy ilyen üzenetet, nem válaszol rá. Én azért szeretem, mert az enyém.

Végezetül

Nagyon izgalmas IRC-botot írni. Remélem, te is örömet lelsz benne, és a tudásodat okosan használod. Sok szerencsét az IRC határtalan világához!



Fülöp Balázs

(xut@freemail.hu) 17 éves, imádja a Túró Rudit, a Debian Linuxot és a teheneket. Az ELTE Radnóti Miklós Gyakorlóiskola tanulója immár ötödik éve. Kedvenc írója Slawomir Mrodek. Leginkább a számítógépes hálózatok biztonsága érdekli.

Szüreteljünk almát!

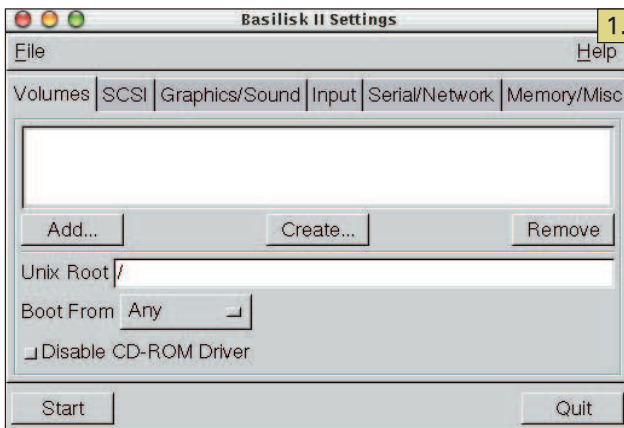
Sorozatunk mostani részében az Apple Mac OS futtatásához szükséges környezet beállításával kapcsolatban szerzett ismereteimet kísérlem meg olvasóinknak átadni.

Mindenekelőtt szerezzük be a nélkülözhetetlen össze-
tevéket! Szükség lesz a Basilisk II forrására, egy
eredeti Macintosh-gép ROM-jára, az Apple webol-
daláról letölthető Mac OS 7.5.3-as, illetve a 7.5.5-ös Mac OS-
frissítésre (ezeket az Apple ingyenesen elérhetővé, letölthetővé
tette), egy indító lemezlenyomatra, néhány segédprogramra
– és már indulhat is a móka.

Én azt szeretem, ha egy program vagy egy kísérlet mindig
külön munkakönyvtárral bír, így a múlt hónapban elkezdett
EMU könyvtárban létrehoztam egy *MacOS* alkönyvtárat,
innenről ebben dolgozom. Először a Basilisk II csomagot
készítem elő a fordításra az alábbi parancs segítségével:

```
tar -xvzf BasiliskII_src_31052001.tar.gz
```

feltételezve, hogy a fentebb említett fájl az *EMU/MacOS*
könyvtárban található. Ekkor létrejön egy *BasiliskII-0.9*
könyvtár, a *BasiliskII-0.9/src/Unix* könyvtárba belépve rögtön
nekiállhatunk a program fordításának: adjuk ki a
. /configure, majd a make és a make install parancsokat.
Alapértelmezettként a program a /usr/local alkönyvtáraiba
telepíti magát, ehhez azonban írási joggal kell rendelkezünk,
a legegyszerűbben a su parancs segítségével szerezhetünk
rendszergazdai jogokat. Természetesen más könyvtárat is meg-
adhatunk. Most már futtathatjuk is a programot, adjuk ki a
BasiliskII parancsot. Ekkor egy kényelmesen kezelhető
grafikus beállítófelületet kapunk, ahol a program működését
mindentre kiterjedően állíthatjuk be (1. kép).



Vegyük sorra a menüpontokat!

Volumes: azokat a lemezrészeket állíthatjuk be, amelyeket
használni szeretnénk; az *Add...* gombra kattintva már meglévő
lemezlenyomatokat adhatunk a rendszerhez, a *Create...*
gombbal új lemezlenyomat(ok)at hozhatunk létre, a *Remove*
gomb használatával pedig eltávolíthatjuk a kijelölt lemezle-
nyomatot. A *Unix Root* sorában adhatjuk meg, hogy melyik
könyvtárat szeretnénk az emulátorból a unix nevű lemezzel

A ROM-fájl

Mint már szoltam róla, az emulátor futtatásához egy eredeti Mac
ROM szükséges, amihez egy a saját tulajdonunkban lévő gép ROM-
jának a kinyerésével juthatunk hozzá. Felhívnam mindenkinek a
figyelmét, hogy bármilyen más forrásból származó ROM használata
törvénytelen!

elérni (lásd később); végezetül a rendszerindítás eszközét
állíthatjuk be – ez lehet bármi (Any) vagy CD-ROM, illetve
engedélyezhetjük vagy tilthatjuk a CD-ROM használatát.
Graphics/Sound: itt adhatjuk meg a grafikus felület jellegét, ami
lehet teljes képernyős (*Fullscreen*) vagy ablakban futó (*Window*);
beállíthatjuk az ablak másodpercenkénti frissítését, az ablak
méretét, illetve a hang engedélyezését. Egészen különleges
képernyőméreteket is megadhatunk, például 800×384-et.
Serial/Network: e helyütt a soros kapukat és a hálózati eszközt
állíthatjuk be (lásd később).

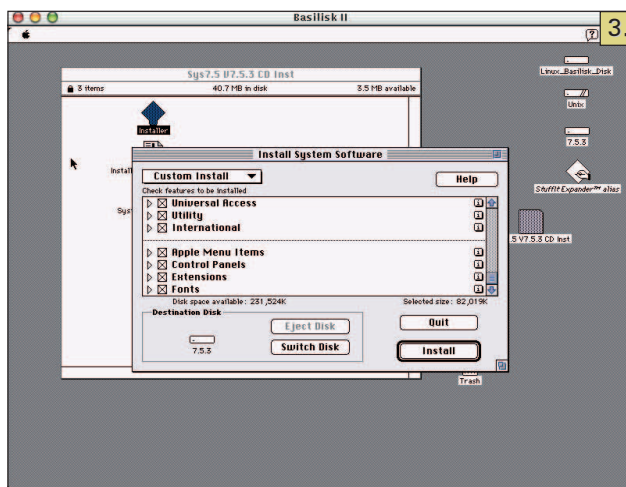
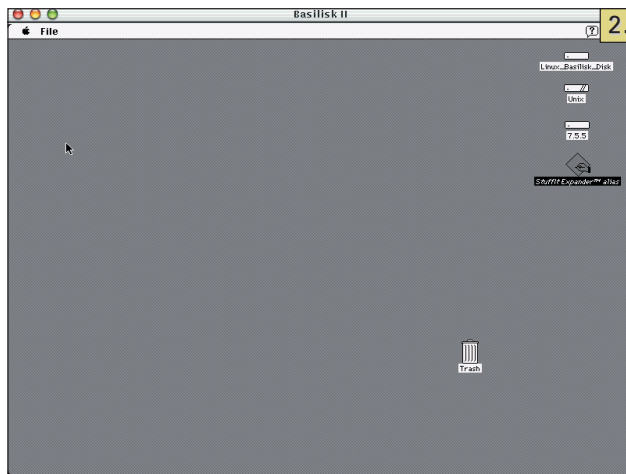
Memory/Misc: a *Volumes* után ez a legfontosabb beállításokat
tartalmazó menüpont, az emulált gép fizikai memóriaméretét,
a Macintosh-gép fajtáját, illetve a ROM-fájl elérési útvonalát
adhatjuk meg vele (a ROM-fájl nélkül a programot egyáltalán
nem tudjuk használni).

Ha ezzel készen vagyunk, kísérletet tehetünk a Mac bekap-
csolására: kattintsunk a *Start* gombra. Most csak egy szürkés
háttérrel kapunk középen egy lemezzel, aminek a kellős közepén
egy kérdőjel villog – ez bizony azt jelenti, hogy hiányzik
a rendszer indulásához szükséges operációs rendszer.

Az operációs rendszer telepítése

Ehhez a lépéshez egy „rendszerindító” lemezre lesz szükség.
Én az Interneten találtam egy ilyen lemezt, és ezt használtam.
Hozzunk létre egy lemezlenyomatot a rendszernek, nevezzük
mondjuk *7.5.3.dsk*-nak. Legalább egy 150–250 MB-os lemez
létrehozását javaslom, mivel saját tapasztalataim alapján a jó
öreg programokkal ezt is hamar meg lehet tölteni. Adjuk hozzá
az indító lemezlenyomatot is; a memóriaméret 16–32 MB
között eszményi választásnak tűnik, ezek után kattintsunk
a *Start* gombra. Most már látnunk kell az operációs rendszer
betöltődését. A rendszer betöltése közben láthatjuk, hogy az
általunk létrehozott lemezlenyomatot is felismeri, azonban
nem tudja birtokba venni, mivel nincs előkészítve a haszná-
latra. Ezt tegyük meg, kattintsunk az *Initialize* gombra, amikor
is figyelmeztető üzenetet kapunk, miszerint a lemez tartalma
elvész, ha a műveletet tovább folytatjuk. Ennek ellenére kat-
tintsunk bátran a *Continue* gombra.

Ezután a 2. képen látható képernyőhöz hasonlólt kell látnunk.
Macintoshunk feltámadt! Kattintsunk a *Stuffit Expander* ikonra,
majd a *File/Preferences...* menüponttal csalogassuk elő a csoma-
golóprogram beállítófelületét. Itt a *Destination* ikonra kattintva
állíthatjuk be azt a kívánt könyvtárat, ahová a kicsomagolt
anyagok kerülnek. Nálam ez a 7.5.3-as lemezen létrehozott



Install könyvtár lett. Ezek után a *Unix* nevű lemezrészre kattintva keressük meg a 7.5.3-as Mac OS telepítőfájljait, szám szerint 19-et, és mindet kijelölve dobjuk rá a *Stuffit Expander* ikonra – ekkor az összes fájl kicsomagolódik. A megadott könyvtárban kattintsunk a *System 7.5.3 01of19.smi* fájlra, a felhasználói szerződést fogadjuk el. Az ellenőrzés után létrejön egy *Sys7.5 V7.5.3 CD Inst* ikon a munkaasztalon. Bőszén kattintsunk rá, így megkapjuk az *Installer* programot. Erre is kattintanunk kell kettőt (akinek a kattintóizmjai esetleg elfáradtak, az pihenhet egy kicsit), ekkor már ténylegesen a 7.5.3-as rendszert telepítjük! Választhatjuk az egyszerű (Easy), vagy az egyéni (Custom) telepítést.

A teljes telepítés megközelítőleg 82 MB helyet foglal el. Ha végeztünk vele, lépünk ki belőle, és indítuk újra a rendszert. Távolsítsuk el az indító lemezlenyomatot, és élvezzük a sok kattintgatás eredményét: a Mac OS 7.5.3-as rendszert a PC-n! Nálam az egyetlen hibát valami a/rose bővítmény jelentette, de miután eltávolítottam a rendszer könyvtárából, egy teljesen egészséges rendszert kaptam.

Hálózati beállítások

Ahhoz, hogy a mostani Mac-gépből a hálózatot is lássuk, először a *sheep-net* rendszermagmodult le kell fordítani és be kell tölteni a memóriába. Nekem a Basilisk II-vel egy csomagban érkező modul nem sikerült lefordítanom, így hosszas keresgélés után ráakadtam egy Mandrake-csomagra, amivel azután sikerrel jártam. A modul betöltése a rendszergazdaként kiadott

`insmod sheep_net -up .o` paranccsal lehetséges. Ezután használhatjuk az *eth0* hálózati kaput, amelynek beállításait a Mac OS elindulása után a *System Folder/Control Panels/ TCP/IP* ikonokon keresztül érhetjük el.

Frissítés 7.5.5-re

Az alapterjesztés nem tartalmazza a *Stuffit Expander*-t, ezért töltsük le a <http://www.stuffit.com/expander/download.html> címről, és telepítsük. Ám ahhoz, hogy telepíteni tudjuk, a kicsomagoláshoz egy előző változatra lesz szükség (kicsit a 22-csapdájára emlékeztető helyzet...). Az első indításkor használt lemezből kinyerhetjük ezt a programot. Ezután másoljuk be a *System 7.5.5 update.hqx* fájlt az emulátor meghajtójára, majd ismét kattintgassunk, amíg csak a fájl ki nem csomagolódik. A kicsomagolásra megadott könyvtárban ismét egy *Installer* programot kell keresni, erre kattintva a telepítő frissíti a rendszerösszetevőket.

A telepítés befejezése után virtuális gépünket indítsuk újra! Élvezzük a frissített rendszert! Csak akkor, ha elindul – én az induláskor fellépő hibát úgy tudtam orvosolni, hogy a következő indításkor a SHIFT billentyűt lenyomva tartottam, ezáltal a rendszer semmilyen kiegészítést nem töltött be. A *Control Panels/Memory* ikonján kattintva egy beállítófelületet kapunk, ahol a *Virtual Memory*-t kapcsoljuk ki. Ezután már tökéletesen működött minden.

A következő részben a különféle programok lelőhelyeit és telepítésüket vesszük szemügyre.



Csontos Gyula

(Csontos.Gyula@linuxvilag.hu)

a Linuxvilág szakmai és CD-szerkesztője.

Szabadidejében szívesen mászik hegyet és kerékpározik.

Kapcsolódó címek

Ahol mindenképpen érdemes böklátszni:

- ➔ <http://www.uni-mainz.de/~bauec002/B2Main.html>
- ➔ <http://www.info.apple.com/support/downloads.html>
- ➔ <http://www.ultimatemac.com/>
- ➔ <http://lowendmac.com/mac2win/02/0501.html>
- ➔ http://sourceforge.net/forum/forum.php?forum_id=6132
- ➔ http://download.info.apple.com/Apple_Support_Area/Apple_Software_Updates/English-North_American/Macintosh/System/Older_System/System_7.5_Version_7.5.3/
- ➔ <http://www.apple.com/>
- ➔ <http://www.vineyard.net/vni/faq/config/mactcp.shtml>
- ➔ <http://toastytech.com/guis/mac755.html>
- ➔ <http://www.seanet.com/help/dial/mac/opentransportppp.shtml>
- ➔ <http://www.peak.org/ipswitch/mactcp.html>
- ➔ http://libweb.sonoma.edu/research/instructions/mac7_5_3.html
- ➔ <http://geeks.free.fr/macOS.htm>
- ➔ <http://www.welovemacs.com/macOS7-5-5.html>
- ➔ <http://www.speakeasy.net/low/main.php?page=macOS758>
- ➔ http://www.its.unimelb.edu.au/InfoSheets/InfoSheet_135.html

PHP – mnoGoSearch

Azaz hogyan házassítsuk az izmos keresőmotort a szeretett PHP-val?

Sokszor felmerülő igény egy kiterjedtebb honlappal szemben, hogy a tartalma kereshető legyen. A feladat elég nyögvenyelős lehet, amennyiben a webhely sok, különféle működésű modulból épül fel. Minden egyes modulra saját keresőt építeni, majd egy rendszerbe foglalni őket, igen nagy falat lehet. Érdemes elgondolkodni, nem volna-e jó általánosítani valahogy a feladatot, és az összes egyedi esetet egységes rendszerbe foglalni. Hosszú távon mindenképpen kifizetődő az egyszerű nagyobb munkát bevállalni. Egy jól összerakott keresőrendszert feleltetni meg a későbbi igényeknek már töredékét teszi ki az amúgy szükséges munkaóráknak. Amikor az ember azon gondolkodik, hogy valamiféle általános megoldás lenne az igazi, az sem árt, ha körülszimatol egy kicsit a Világhálón, hátha a több tízezer nyílt forrású megoldás között talál valami, a célnak megfelelőt.

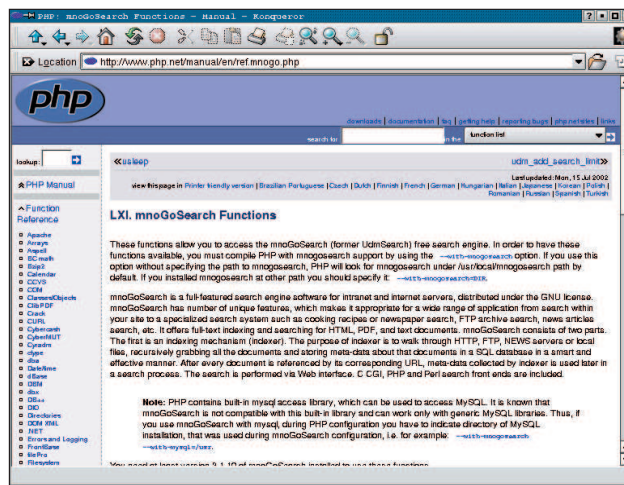
A mnoGoSearch – ismerkedés

Jómagam a fenti gondolatmenetet végigvezetve jutottam el a mnoGoSearch keresőmegoldásához, még messzire sem kellett érte mennem. A PHP kézikönyvét böngészgetve akadtam rá – mondhatni véletlenül – a mnoGoSearch nevű kiegészítésre. Kis utánajárás után kiderült, hogy a mnoGoSearch maga egy igen kiterjedt, nyílt forrású keresőmotor. Saját, C-ben írt CGI-felülete is jól használható és rugalmas. Emellett Perlhez és PHP-hez is létezik illesztés. Ez a támogatás a PHP hivatalos terjesztésének részét képezi.

Maga a mnoGoSearch két jól elkülöníthető részre bontható. Az egyik végzi a weboldalak, adatbázistáblák indexelését a megadott beállítások alapján. Ezt időszakosan (tipp: cron) futtatva egy szorgalmas keresőrobotot állíthatunk munkába. Magának a keresőfelületnek a megvalósításában lehetőségünk nyílik dönten, hogy a mnoGoSearch sajátját használjuk-e fel vagy írunk egyet. Ha a kereső saját megoldását alkalmazzuk, a keresőoldalt sablonfájl segítségével faraghatjuk egyedi arculatúra. Ha a PHP-illesztésen keresztül vesszük fel a kapcsolatot, teljesen egyedi felületeket hozhatunk létre. Ennek a programozói feladatnak a bonyolultsága nem haladja meg egy egyszerű adatbázis-lekérdezés szintjét.

A mnoGoSearch nagy erőssége, hogy külső szűrőket is képes alkalmazni, amelyek segítségével akár .pdf vagy egyéb különlegesebb formátumú állományok szövegének indexelésére is sor kerülhet. Ehhez egy, az adott formátumú anyagot szöveges állománnyá alakító programot kell beszerezni. Ha a program megvan, a keresőt kell beállítani, hogy az adott típusú állományokat ezen a szűrőn átengedve lásson hozzá a tartalom átvizsgálásához. A mnoGoSearch jó pár formátummal és kapcsolódási lehetőséggel önmagában is elboldogul. Beépített képességei a következők:

- egyszerű text/html, text/plain oldalakat indexel (még szép);
- dinamikusan létrehozott tartalmú oldalakat a legcifrabb URL-en keresztül is is elér;
- titkosított (HTTPS) kapcsolatokat is kezel;
- ha kell, jelszavasán védett (HTTP-azonosítás) oldalak is feldolgozhatók vele;



1. kép <http://hu.php.net/mnogo>

- FTP-helyek végignyálazása sem okoz neki gondot;
- adatbázis-kiszolgálók tábláinak indexelésével is elboldogul, bár itt vannak megkötések.

A keresőmotor maga is SQL-alapú, adatbázisban tárolja az indexelés során összegyűjtött adatokat. Rengeteg adatbázis-kiszolgáló-típussal képes együttműködni. Ha indexelésük céljából közvetlen adattáblákban is turkálni szeretnénk, egy erős megszorítással találkozunk: csak ugyanabban az adatbázisban képes feldolgozni a mnoGoSearch-adattáblákat, amelyekben a saját indexelésre fenntartott táblákat is tárolja. Ez alól az egyedüli kivétel az, ha háttéradatbázisként MySQL-t használunk.

A mnoGoSearch telepítése

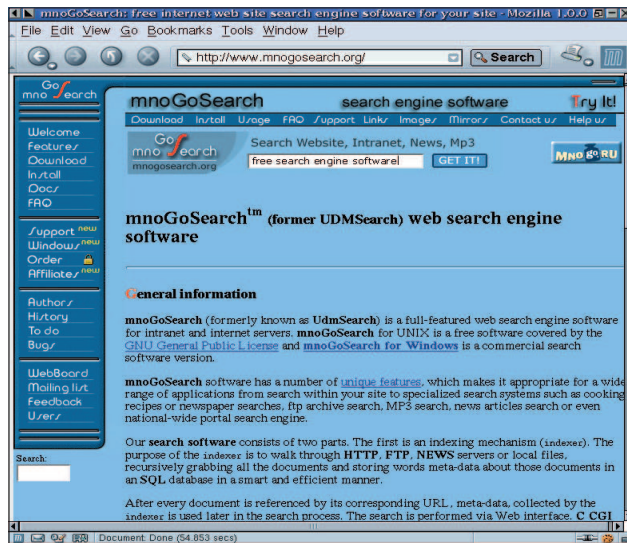
Először is szükség lesz egy friss forrásra. Ehhez a mnoGoSearch honlapját kell felkeresni, azon belül is a következő oldalt: <http://www.mnogosearch.org/download.html>. A cikk írásakor a legutolsó üzembiztos kiadás mellett döntöttem, ami a 3.1.20 változátszámmal bír. Ennek telepítését Debian Woody környezetben próbáltam ki, de egyes apró eltéréseket kivéve ez a folyamat valószínűleg más GNU/Linux-rendszereken is követhető. A bemutatott telepítés MySQL jelenlétét feltételezi, ugyanis ezt választottam háttéradatbázisnak. Ha megvan a *mnogosearch-3.1.20.tar.gz* állomány (vagy az éppen legfrissebb), a szokásos módon lehet nekiesni.

Kicsomagolás, belépés a könyvtárba, ismerkedés a `./configure --help` kiadása által kapott lehetőségekkel. Eztán a `configure` parancsfájl megfelelő beállításokkal való futtatása következik, végül `make`, `make install`.

Ez a hagyományos telepítési módszer, ám érdemes észrevenni, hogy a fejlesztők egy kellemes Perlben írt kis kérdezőprogramot mellékeltek, ami végül a válaszoknak megfelelően hívja meg a `configure` héj-parancsfájl. Lássuk, miféle beállításokat hajthatunk végre!

Ha a fent említett `install.pl` futtatása mellett döntünk, az első kérdése arra fog vonatkozni, hogy mi legyen a telepítés célkönyvtára – ez alapesetben a `/usr/local/mnogosearch` könyvtár. Ha nincs jobb ötlet, így is maradhat, a PHP is itt fog majd kutakodni utána.

Ha a `configure` közvetlen meghívását választjuk a fordítás előtti beállítások végrehajtására, e fenti adat a `--prefix=<könyvtár>` kapcsolóval adható meg. Ha meg



2. kép A mnoGoSearch hivatalos honlapja

vagyunk elégedve az alapértelmezett könyvtárral (lásd az előző bekezdést), az adat megadása el is hagyható. Ha másképp nem rendelkezünk, a most megadott könyvtár alá kerül minden mnoGoSearch-csel kapcsolatos adat, a beállításoktól egészen a naplóállományokig. Nálam ez így maradt. Kézi `configure` híváskor ez esetben semmit nem kell megadni, az `install.pl` futtatásánál is elég ENTER-t ütni az alapértelmezett `no` válaszra. Ha ez a felállítás mégsem tetszene, egy `yes` után az összes alkönyvtár elhelyezkedése sorra megadható. Parancssoros beállítás esetén a szükséges kapcsolók a `configure` súgóijából ismerhetők meg.

Ha a könyvtárak megadásán túltettük magunkat, következik a háttéradatbázis megadása. A beépített adatbázis-támogatásra nem lesz szükség, hiszen a gépen van telepített MySQL.

Az `install.pl` ezt magától képes megtalálni, tegye is csak. Válaszoljunk `yes`-t arra a kérdésre, hogy maga keresse-e meg a lehetséges adatbázis-csatlakozásokat. Ha nem találna a megfelelő könyvtárat, magunknak kell megadnunk. Először az adatbáziskiszolgáló típusát, majd azt a könyvtárat kell megadni, ami alatt a MySQL fejlécfájlok és függvénykönyvtárak találhatóak. A MySQL-támogatással való felruházáshoz kézi vezérlés esetén a `--with-mysql=<könyvtár>` megadására lesz szükség.

Az ezt követő kérdésekre mindenütt hagyhatjuk az alapértelmezett választ, ezek a beállítások céljainkhoz javarészt jók lesznek. Ha a kézi `configure` meghívást választottuk, és semmilyen egyedi beállítást nem akarunk alkalmazni, nem kell további kapcsolókkal bajlódniuk.

Vagy így, vagy úgy, de eljutottunk oda, hogy a `configure` parancsfájl sikeresen lefutott. Ekkor már a hagyományokat követve jöhet a szokásos `make`, majd a `make install`. Ezáltal a mnoGoSearch alaptelepítése a helyére került.

További tennivalók és beállítások

Bizony, a keresőrendszer telepítése még csak az első lépés volt. A feltett csupasz rendszert életre kell keltetni, be kell állítani. Először is a mnoGoSearch számára létre kell hozni a MySQL háttéradatbázist:

```
mysqladmin -uroot -p<jelsz> create
↳ mnogosearch
```

Ezáltal létrejött az üres adatbázis, amelyben majd a mnoGoSearch a szótárát, az oldaladatbázisát fogja tárolni. Ehhez azonban még nincsenek meg a szükséges adattáblák. Ezek létrehozásához a mnoGoSearch forráskönyvtárban, a `create/<adatbázisszerver_neve>` könyvtárban találunk segédanyagot, jelen esetben a `create/mysql` alatt. Ezt a szöveges állományt egy az egyben ráengedhetjük a MySQL-re, amely a kívánt adattáblákat létrehozza:

```
mysql -u<felhasználó> -p<jelszava> mnogosearch
↳ < create.txt
```

Az adatbázis megvan, a táblák rendben, de akad még tennivaló a ház körül. Ha másért nem is, de próbaképpen érdemes beüzemelni a mnoGoSearch saját CGI-felületét. Ehhez a mnoGoSearch telepítési könyvtárának `bin` alkönyvtárban található `search.cgi` programcskát kell átmásolni az Apache `cgi-bin` könyvtárba. Emellett `search.htm` néven másoljuk le a mnoGoSearch `/etc` könyvtárban levő `search.htm-dist` nevű sablonállományt is. Ez azért szükséges, mert ezen arculatmeghatározó állomány nélkül a keresőfelület meg sem mozdul, ráadásul innen veszi a beállításait. Ekkor a keresőfelületnek a `www.kiszolgálóneve.hu/cgi-bin/search.cgi` címen működnie kell, de természetesen nem lehet keresni, hiszen még nincs semmilyen adathalmaz. Ráadásul egy kis beállítgatásra is szükség lesz, hiszen a keresőprogram még arról sem tud, merre keresse a háttéradatbázisát. A keresőfelület beállításához nem, de az indexelő robot beállításához szükséges beállításokért felelős állományra szükség lesz. Ez a fájl a sablonfájllal egy könyvtárban található, `indexer.conf-dist` néven. Ebből `indexer.conf` elnevezéssel szükség lesz egy másolatra, amelyben a mnoGoSearch indexelő motorjának beállításait tudjuk megadni. Furcsának tűnhet, hogy külön beállítóállomány szükségeltetik az indexelést végző programrésznek, külön a keresőfelületnek. Jobban belegondolva viszont belátható, hogy ez a két modul akár két teljesen más gépen is elhelyezkedhet, ráadásul ugyanahhoz az adatbázishoz csak az indexelő részre kell olyan adatbázis-kapcsolat, amelynek a mnoGoSearch adatbázisába írni kell tudnia. A `search.cgi` teljesen mellőzhető, hiszen e cikk végcélja, hogy bemutassa, hogyan lehet PHP-ban mnoGoSearch-háttérrel dolgozó keresőfelületeket kialakítani. Mivel amúgy sem árt a PHP és a mnoGoSearch házasítása előtt kipróbálni, hogy a keresőt hajlandó-e működni, és mert a beállítások orozslánrészét úgyis az `indexer.conf`-ban kell megadni, érdemes ezt a kis pluszmunkát elvégezni. Lássuk hát, mit és hogyan kell „belőnünk”!

Be kell állítani többek között, hogy melyik kiszolgálón található az adatbáziskiszolgáló, és milyen felhasználónév-jelszó párossal lehet hozzáférni. Kezdjük az `indexer.conf`-fal! A `DBAddr` címke után az adatbázis-hozzáférés adatait címfórmátumban kell megadnunk. Szokatlan elgondolásnak tűnhet, hogy egy ilyen erőforrást is ebben a formában adjunk meg, ám mivel ez egy az `ftp://` vagy `http://` címmel képviselt adatforrással teljesen egyenértékű háttérerőforrás jelöl meg, e megadási

mód abszolút jogos és célravezető:

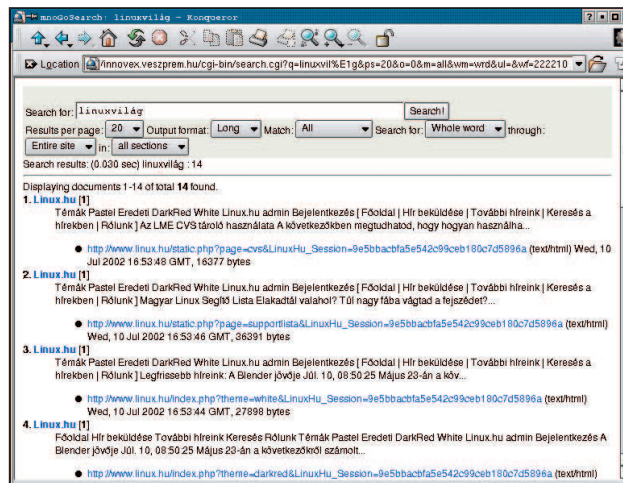
```
DBAddr mysql://felhasznalo:jelszo@localhost/
mnogosearch/
```

A cím elején a `mysql://jelzi`, hogy `mysql` „protokollt” szükséges használni. Ezután a `mnogosearch` adatbázishoz férő MySQL-felhasználó neve és jelszava következik, kettősponttal elválasztva. A `@` jelet követően lehet megadni az adatbázis-kiszolgáló gazdagép teljes nevét. Mivel ez nálam ugyanaz a gép, elegendő volt a `localhost`. Itt ha a `mysql` nem a szabványos kapun figyel, egy kettősponttal elválasztva a kapu száma is megadható. Az ezt követő perjel után adható meg az adatbázis neve, ami pár bekezdéssel ezelőtt `mnogosearch` néven lett létrehozva. Ugyancsak ezt az adatsort kell a `search.htm` sablonállományban megadni, hogy a keresőfelület is tudja, hol kutakodjon. A megadás módja teljesen megegyező, itt is a `DBAddr` címke a felelős az adatbázis-kapcsolat meghatározásáért. Ezzel mind a `search.cgi`-t, mind az indexelő programot működőképessé varázsoltuk. A keresőfelület már bármely szóra hibajelzés nélkül képes kijelenteni, hogy nincs rá találat az adatbázisában.

A használhatóság céljából érdemes egy kis figyelmet szentelni a `LocalCharset` címkének, ugyanis itt adható meg, hogy milyen kódolás szerint értelmezzen, valamint hogy milyen karakterek számítnak értékelhető betűnek. E beállítás hiányában az indexelő motor a számára nem kezelhető karakterek mentén a szavakat szétvája, és a két szórészletet tárolja indexelt szóként. Ez magyar szövegnél annyit jelentene, hogy a hosszú ékezetes karaktert tartalmazó szavaink biztosan több szóként tárolódnának. Példának okáért a „következők” szó helyett a `mnoGoSearch` a „következ” és az „ek” szavakra lelne. Állítsuk tehát a `LocalCharset` címke mögötti értéket az `iso-8859-2` értékre. Ezt mind az `indexer.conf`, mind a `search.htm` állományban meg kell tennünk.

A `search.html`-ben megadandó beállításoknak ezzel a végére is értünk. Az egyelőre szóba sem került, miként fogja az indexelő program feldolgozni a kívánt oldalakat, ráadásul azt sem tudja, melyek az indexelhető (azaz a kereshetőnek szánt) webcímek és tartományok. Ahhoz, hogy a honlapokat feldolgozó program tényleges munkát végezhesen, meg kell neki adnunk azokat a belépési pontokat, ahonnan a tartalom indexelését el kell kezdenie. Szintén megadandó, hogy a lapban talált hivatkozások közül melyeket kövesse és melyeket ne. Erre való az `indexer.conf` `Server` címkéje. Ezáltal adhatók meg az olyan belépési pontok címei, melyek mentén a weblap feldolgozása megindulhat. A lap címének megadása mellett azt is közölni kell, hogy az esetleges hálóhivatkozások milyen szinten legyenek lekövetve. Ennek négy különféle módja létezik:

- `page`: semmiféle hivatkozást nem hajlandó követni, csakis ezt az oldalt indexeli. Ritkán van rá szükség.
- `path`: ez esetben szorgalmas kis pókunk a megadott címen található oldalban csak az olyan hivatkozásokon fog végigszaladni, amelyek a megadott címmel egy könyvtárba vagy az alá mutatnak. Mondhatni, az eme elérési útvonal alatti területet fogja végigzongorázni. Alapértelmezésben, ha külön nem adjuk meg, az indexelő program ezt a fajta hivatkozáskövetési módot fogja alkalmazni.
- `site`: annyiban különbözik az előzőtől, hogy a teljes webhelyet hajlandó végigkövetni, még akkor is, ha a megadott belépési pont annak csak egy alkönyvtára. Csak akkor megy ki innen, ha a kijutáshoz szükséges hivatkozást talál.



3. kép A CGI-felület munkára fogva

Azaz ha a `http://www.valami.hu/pillanatnyi` címről indulva hivatkozást talál a `www.valami.hu`-ra vagy annak bármely alkönyvtárára, vígan követni fogja. Más webhelyekre mutató címeket viszont továbbra sem követ.

- `world`: ezt a típust választva indexelő programunkat arra utasítjuk, hogy minden fellelt hivatkozást gátlástalanul kövessen. Figyelem, ezzel a típussal érdemes nagyon óvatosan bánni, mert hamar elérhetővé lesz gépünk tárhelyének határait, hiszen egy jó kezdőoldalról pók barátunk a fél Internetet végig próbálja majd nyalazni!

Az elméleti alapok után jöjjön egy élő példa! Jomagam pókomat a `www.linux.hu` címre engedtem rá, a következő sorral:

```
Server site http://www.linux.hu/
```

Ezzel a sorral pókunk számára megjelöltünk egy belépési pontot, ahol megkezdheti a munkáját. Derék munkásunk ügyesen végig is nyalazza a megtalált hivatkozásokat, csakhogy kizárólag a `www.linux.hu` címen belül maradva. Próbaszerencse, együk meg a pudingot. Az indexelés elindításához a `mnogosearch` `sbin` könyvtárban az `indexer` nevű futtatható állományt kell elindítani. A pók elstartol, és meg sem áll az utolsó oldalig. Amit talál, nem túl sok, mert a `linux.hu` főbb alegységei különböző altartományokban helyezkednek el: `lists.linux.hu`, `devel.linux.hu` stb. Ezeket az indexelő – teljesen jogosan – nem követi. De mi módon tehetnének meg, hogy ezeket is kövesse? Újabb `Server` parancsokkal? Azok nem erre valók, hiszen tartománymegjelölésen túl fő feladatuk a belépési pontok megjelölése. Olyan címtartományokat kell megadni, ahová a keresőpók még beteheti a lábát, s ez a `Realm` címkével lehetséges. Ahhoz, hogy programunk a `linux.hu`-t, és összes altartományát indexelje, a következő sorra is szükség lesz:

```
Realm http://*.linux.hu/
```

Ez utasítja az indexelőt, hogy bátran kövessen minden olyan hivatkozást, ami megfelel a fenti szűrésnek. A `Realm` alapesetben nem szabályos kifejezésekkel szűr, ehelyett kevésbé erőforrás-igényes megoldással él: a `*` (csillag) jelent akárhány akármilyen karaktert, a `?` (kérdőjel) pedig egy tetszőleges karaktert. A legtöbb címszűrést ennyivel el is lehet intézni, néha azonban a szabályos kifejezésekre is szükség lehet. A fenti sort behelyettesíthetjük a következővel:

```
Realm Regex ^http://.*\.linux.hu/
```

Ez pontosan ugyanazt a szűrést fogja elvégezni, mint az eggyel feljebbi. Szabályos kifejezést ennél sokkal összetettebb feladatokra érdemes alkalmazni. Ha ezen beállításokkal élve indítanánk el az „indexer”-t, kis hivatkozáskövető robotunk már jelentős mennyiségű oldalon menne keresztül, de ne pazaroljuk a *linux.hu* erőforrásait. Ha egy mód van rá, az indexelő robot figyelmét inkább egy saját weblapra irányítsuk!

A feldolgozás végeztével, de akár közben is, böngészőnkkel ismét meglátogathatjuk a *search.cgi*-t, és lőn csoda: ha jó szót írunk be, előbukkan a találati lista. Hurrá, pezsgőbontás, tűzijáték! A keresőmotor telepítése és élesztése sikeresen lezajlott. Korántsem lehet elégedetten hátradőlni a karosszékben, hiszen egészen idáig egy mukkot sem ejtettem arról, miként fog a mnoGoSearch és a PHP együttműködni.

Ahhoz, hogy a hőn áhított együttműködés létrejöjjön, a PHP-t természetesen a mnoGoSearch-támogatással kiegészítve újra kell fordítani. Ezért akár az egész PHP is újrafordítható, de vélhetőleg a kiszolgálón futó PHP-programcskánk jelentéktelen hányada fogja a PHP-nak ezt a modulját dolgoztatni. És biza, ha befordítjuk, minden PHP-fájl futtatásakor ott lesz a memóriában, amiből pedig sosem elég. Fontos még, hogy gépünkön a *libz.so* függvénykönyvtár telepítve legyen, és a PHP is *-with-zlib* beállítással legyen lefordítva. A *zlib* Debianon az *apt-get install libz* parancssal ott is volt a gépen, ezek után a PHP újrafordítása is könnyedén lezajlott. Ha már újrafordítás, a *--with-mnogosearch* kapcsolót is be lehetne illeszteni, de a fenti takarékosági megfontolásból nem tettem meg. Kinek hogy tetszik: a PHP újrafordítható, ha az eredeti beállítások mellé a *mnogosearch* beillesztését is felvesszük. Az újrafordítás megkezdése előtt ne feledjünk el egy *make clean*-t is kiadni, és a *config.cache* állomány törlése is hasznos. A *mnogosearch*-kiegészítés külön lefordítása sem ördögösség, ehhez mindössze a PHP forráskönyvtárából a *./configure --with-mnogosearch=shared* parancsra, majd egy *make-re* lesz szükség. Ezzel a művelettel a forráskönyvtár *modules* alkönyvtárban egy *mnogosearch.so* nevű állományt kapunk vissza, ezt kell beraknunk a PHP kiterjesztő moduljait tartalmazó könyvtárba. E könyvtár elhelyezkedése egy *phpinfo()*-ből *extension_dir* néven megtudható, de a *php.ini*-ben természetesen meg is adható. A lényeg a végeredmény, a *mnogosearch.so* az *extension_dir*-nek megfelelő könyvtárba kell kerüljön. Innentől amelyik PHP-parancsfájlban szükségünk lesz a mnoGoSearch-kezelő függvényekre, be kell szűrni egy

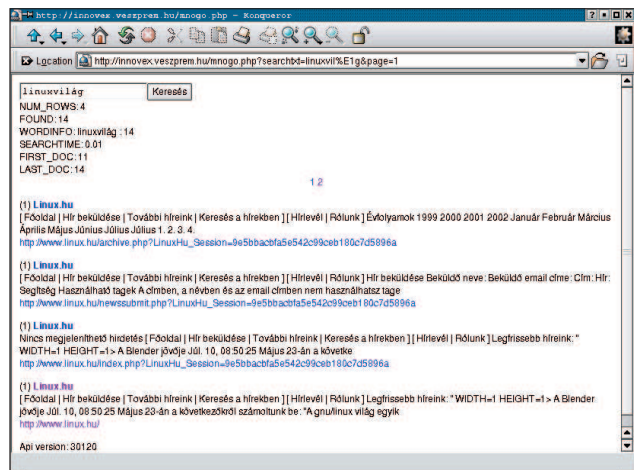
```
dl( mnogosearch.so );
```

sort az elejére, és rendszerünkbe máris behúztuk a megfelelő kiterjesztést.

Együtt a PHP-val

Most, hogy minden együtt van, minden működik, lássuk, mi módon tudjuk kisajtolni a kereséseket a mnoGoSearch adatbázisából a PHP erre szakosodott, immár telepített utasításain keresztül. Ahhoz, hogy a mnoGoSearch keresőjében lekérdezéseket hajthassunk végre, szükség lesz egy kapcsolódásra. Ez a modul is a szokásos utat követi, ahogyan azt a fájl- vagy az adatbázis-kezelésben megszokhattuk. Először le kell foglalnunk egy adott típusú erőforrást, hogy aztán ennek az azonosítójával hivatkozva végezhessünk műveleteket. Az első lépés mindig az *udm_alloc_agent()* függvény meghívása lesz, az utolsó szó joga pedig minden esetben az *udm_free_agent()*-et illeti.

E második függvénnyel tudjuk elvégezni munkánk végén a takarítómunkát – némi hasznos memória visszaadásával a rendszernek. Ennek egyetlen értéke az azonosító, amit a kapcsolat létrehozásakor az *udm_alloc_agent()*-től kapunk. A függvény egyetlen kötelező értékében a kapcsolódási adatokat kell megadnunk, melyek megadási formája teljes mértékben megegyezik a mnoGoSearch beállítási fájljaiban használatos DBAAdrr-féle címformával. Érdemes megemlíteni, hogy az itt megadott adatbázisfelhasználó-jelszó páros nem kerül kiértékelésre, csak ha az *udm_find()* függvény meghívásával érdemi keresést kezdeményezünk. Ezért ne csodálkozzunk, ha az esetleges csatlakozási hiba nem itt, az erőforrás foglalásakor keletkezik, hanem pár utasítással később. Nagyon fontos, ezért külön kiemelő, hogy az adatbázis-hozzáférést megadó címet egy / (perjel) zárjuk, enélkül ugyanis a kapcsolat nem fog létrejönni.



4. kép Keresés, immár PHP-val

Az előbb említett *udm_find()* függvény által tudunk kereséseket végrehajtani. Két értéket vár, az elsőben a keresés kezdeményezése előtt lefoglalt erőforrás azonosítóját kell megadni, másodikként magát a keresőkérést. Egy keresés megadásakor a keresendő szavak szóközzel elválasztott listáján felül egyéb trükkökkel is élhetünk: a különféle szavakat zárójelekkel csoportosíthatjuk, köztük logikai műveleteket is végrehajthatunk. Az & (és jel) a logikai és-t, míg a | (csőjel) a logikai vagyot hivatott jelképezni. Megjegyzem, a sima szavak felsorolása is olyan, mintha logikai és jellel választottam volna el őket egymástól. Azaz a *linux php* keresőkulcs ugyanazt az eredményt fogja adni, mint a *linux&php*. Ugyanakkor a *linux|php* olyan találatokat hoz, amelyekben vagy a *linux* vagy a *php* szó megtalálható. A kizárt szavak megjelölésére a keresőkben általában megszokott kizáró - (mínuszjel) helyett itt a ~-t (tildejel) állították hadrendbe. Az ilyen jellel előjelzett szavakat tartalmazó dokumentumok e lehetőség kihasználásával nem kerülnek a találati listába. Megjegyzem, önmagában egy *~php* keresési kulcs megadása nem értelmezhető, nem fogja az összes olyan találatot kiadni, amelyben nem szerepel a *php* szó. Ez csak arra való, hogy a meglévő találati listából kizárja a nem megfelelőket. Az alábbi tehát működik: *linux ~php*, ami által olyan dokumentumok kerülnek a találati listára, melyek Linux témába vágóak, ámde amelyekben szó sem esik a *php*-ről. Ha végül sikerült keresést kezdeményeznünk, eredményezés egy azonosítón keresztül férhetünk hozzá. Ezt az azonosítót az *udm_find()* adja meg számunkra, vissza-térési értéként.

A keresőoldal

```

<?
define('_PERPAGE',10);

if (!extension_loaded('mnogosearch.so')) {
    dl('mnogosearch.so');
}

echo '
<form action="'.PHP_SELF.'">
<input type="text" name="searchtxt"
↳ value="'.searchtxt.'">
<input type="submit" name="start_search"
↳ value="Keresés">
</form>';

$agentid = udm_alloc_agent
↳ ('mysql://n?v:jelsz @localhost/mnogosearch/');
udm_set_agent_param
↳ ($agentid,UDM_PARAM_CHARSET,'iso-8859-2');
udm_set_agent_param
↳ ($agentid,UDM_PARAM_PAGE_SIZE,_PERPAGE);
udm_set_agent_param
↳ ($agentid,UDM_PARAM_PAGE_NUM,$page);

$sres = udm_find($agentid,$searchtxt);

$num_rows = udm_get_res_param
↳ ($sres, UDM_PARAM_NUM_ROWS);
$num_found = udm_get_res_param
↳ ($sres, UDM_PARAM_FOUND);

echo 'NUM_ROWS: '.$num_rows.'  
';
echo 'FOUND: '.$num_found.'  
';
echo 'WORDINFO: '.udm_get_res_param
↳ ($sres, UDM_PARAM_WORDINFO).'';
echo 'SEARCHTIME: '.udm_get_res_param
↳ ($sres, UDM_PARAM_SEARCHTIME).'';
echo 'FIRST_DOC: '.udm_get_res_param
↳ ($sres, UDM_PARAM_FIRST_DOC).'';

echo 'LAST_DOC: '.udm_get_res_param
↳ ($sres, UDM_PARAM_LAST_DOC).'';

$maxpage = ceil($num_found / _PERPAGE);
$page = floor($page);
$page = ($page > $maxpage) ? $maxpage:$page;

echo '<center>';
for ($i=0; $i<$maxpage; $i++) {
    echo '
<a href="'.PHP_SELF.'?searchtxt=
↳ urlencode($searchtxt).'&page='.(($i)).'">
↳ '.(($i+1)).'</a>';
}
echo '</center>';

for ($i=0; $i<$num_rows; $i++) {

    $hit_url = udm_get_res_field
↳ ($sres,$i, UDM_FIELD_URL);
    $hit_title = udm_get_res_field
↳ ($sres,$i, UDM_FIELD_TITLE);
    $hit_text = udm_get_res_field
↳ ($sres,$i, UDM_FIELD_TEXT);
    $hit_rating = udm_get_res_field
↳ ($sres,$i, UDM_FIELD_RATING);

    echo '
<p>
('.$hit_rating.')
<a href="'.hit_url.'"><b>'.hit_title.'
↳ </b></a><br>'.hit_text.'

```

Miféle adatok nyerhetők ki ebből az azonosítóból? Az `udm_get_res_param()` a keresés eredményére általánosan jellemző adatokat adja vissza, míg az `udm_get_res_field()` segítségével a találatok adatainak kinyerése lehetséges. Kezdjük az általános adatokkal! Ezek megszerzéséhez első értéként a keresési eredményt megjelölő azonosítót kell megadni, tehát azt, amit az `udm_find()` kiadásakor kaptunk, másodikkal pedig azt kell megadnunk, hogy mely adatra van szükségünk. Ezek az adatok a következők:

- `UDM_PARAM_NUM_ROWS`: a listázásra kerülő találatok száma.
- `UDM_PARAM_FOUND`: az összes találatok száma. Mivel a találatok megadható méretű oldalakra bontódnak szét, az eggyel ezelőtti adat nem feltétlenül egyezik meg ezzel, ami valóban megadja, hány találat sikeredett.
- `UDM_PARAM_WORDINFO`: szöveges formában adja vissza,

hogy az egyes szavakra külön-külön hány találat érkezett.

- `UDM_PARAM_SEARCHTIME`: századmásodperc pontossággal a keresési idő.
- `UDM_PARAM_FIRSTDOC`: az éppen listázandó oldal első találatának sorszáma.
- `UDM_PARAM_LASTDOC`: az éppen listázandó oldal utolsó találatának sorszáma.

Lapok? Bizony-bizony, a `mnoGoSearch` modul a találatokat lapokra osztja. Ha másképp nem rendelkezünk, az első oldalt fogjuk visszakapni, és egy oldalon húsz találat megjelenítésére kapunk módot. Ezen természetesen lehet változtatni, csupán az `udm_set_agent_param()` függvényt kell szorgalmasan hívogatni. Pár példa a teljesség igénye nélkül:

```
// 10 találat legyen 1 oldalon;
udm_set_agent_param($aid,UDM_PARAM_PAGE_SIZE,10);
```

```
// Ugrás a harmadik oldalra
// (az első oldal száma a 0)
udm_set_agent_param($aid,UDM_PARAM_PAGE_NUM,2);
// Ennek megadása nagyon fontos, mert enélkül
// a mnoGoSearch hibásan keresi a megadott
// űkezetes szavakat!
// Meg kell egyeznie az indexer.conf-ban
// megadott LocalCharset űrtűkkel.
udm_set_agent_param($aid,UDM_PARAM_CHARSET,
'iso-8859-2');
```

A fentiekén túl ezen a függvényen keresztül rengeteg beállítást lehet megadni, de helyszűke miatt erre most nincs lehetőség, viszont a <http://hu.php.net/mnogo> címen a teljes lista elérhető. Térjünk inkább rá, hogyan is lehet a találatokat kipróbálni, hogy azután guszpusos formában tálalhassuk a nyájas látogató elé. Erre a korábban már említett `udm_get_res_field()` hivatott. Első értéként a keresési eredménylistára hivatkozó azonosítót kell megadni, másodiként pedig a találat oldalon belüli sorszámát. Figyelem, a számozás nulláról indul! A harmadikkal megmondjuk, melyik adatra van szükségünk az adott találati sorral kapcsolatban, lehetséges értékei közül a fontosabbak:

- UDM_FIELD_URL: a talált oldal webcíme.
- UDM_FIELD_TITLE: a meglelt weboldal `<title>` `</title>` címkéi közt megadott oldal elnevezés.
- UDM_FIELD_DESC: a meglelt oldal leírása. Ezt a mnoGoSearch az oldal Description-típusú metaeleméből nyeri ki.

- UDM_FIELD_TEXT: az oldal szövegének első pár sora. A fenti Description hiányában érdemes kijelteni, hogy a keresést kezdeményező némi rálátással bírhatson, ez-e a neki megfelelő oldal.

Végszó

Nagyjából ennyi azoknak a PHP-utasításoknak a köre, amelyek alkalmazása elengedhetetlen, ha ezen a módon akarunk kereséseket végrehajtani. Összefoglalóként lássuk mindezt egyben is, egy igen egyszerű keresőoldal példáján keresztül, melyet a 66. oldalon tanulmányozhatnak részletesebben.



Heilig (Cece) Szabolcs

(cece@php.net) Veszprémben él. Hobbija, kedvenc időtöltése és munkája is a programozás. Szabadidejében hajlamos kerékpárra pattanni, vagy baráti társaságban szerepjátékokkal foglalatzkodni.

Kapcsolódó címek

- A keresőmotor hivatalos lapja
 ➔ <http://www.mnogosearch.org>
 A mnoGoSearch PHP-modul leírása
 ➔ <http://hu.php.net/mnogo>



Adatbázisok egységesített elérése PHP-val

Írásunkban két olyan módszert tárgyalunk, amelyekkel egyszerűbben férhetünk hozzá adatbázisainkhoz: a PEAR::DB-t és az ADODB-t.

A mennyiben foglalkoztál már PHP-val, a PEAR név biztosan ismerősen cseng. Magyarul körtét jelent, de PHP-s körökben hallatán inkább az e névvel fémjelzett alkalmazáscsomagra gondol az ember. Tapasztaltabb felhasználóknak az ADODB név is ismerősen csenghet. Cikkünkben a PEAR::DB és az ADODB nevű egységesített adatbázis-elérési rétegekkel fogunk megismerkedni. Mindkét programcsomag egy egyszerűnek tűnő, de eddig megoldatlan gondot orvosol. Tegyük fel, hogy az egyik délután írtál egy néhány oldalas PHP-kódot, amellyel a cég MySQL-adatbázisában kell valamilyen műveletet elvégezned. Tételezzük fel azt is, hogy nap mint nap ilyen egyszerűbb kódokkal foglalkozsz. Hónapok, esetleg évek során egész szép mennyiségű kód gyűlik össze, benne kisebb-nagyobb programokkal, amelyekben csak az a közös, hogy egy adatbázis-kiszolgálón végeznek különböző műveleteket. A programok száma egyre nő, a cég ügyfélköre is folyamatosan gyarapodik, egy szép napon pedig – amikor már nem tudod tovább állítgatni az adatbázis-kiszolgálót – rádöbbsz, hogy a jelenlegi rendszer teherbírása már nem elégséges. A cég vezetősége úgy dönt, hogy ezt elhárítandó egy Oracle adatbázis lenne a megfelelő. És innentől kezdve csőstől jönnek a nehézségek: az adatbázis megváltoztatásával az összes eddigi alkalmazás alól – mondhatni – kihúzod a szőnyeget. Minden egy csapásra működés-képtelenné válik.

Ha nem voltál elég előrelátó, és ilyen helyzetben találsz magad, az elkövetkező hónapok lehet, hogy a korábbi programok újraírásával fognak telni. Ha ellenben erre az eshetőségre már a legelején is gondoltál, előfordulhat, hogy mindössze egyetlen fájlban kell egy változó értékét módosítanod. Amennyiben előretekintő vagy, a fenti módosítással a cég – és egyben az adatbázis – fejlődése akadálytalanul folyhat tovább, és neked nem kell mást tenned, mint amit eddig: az adatbázisból értékeket kihalászni, megjeleníteni és olykor módosítani. Ha viszont balszerencsésnek bizonyulsz, nincs más választásod, mint ettől a pillanattól kezdve elszakadni az, eddig használt adatbázisodtól, és egy adatbázis-független kód írásába kezdeni, szabadidődben pedig a korábban felhalmozódott kódtömeg rendbetételével foglalkozkodni. Mostanra valószínűleg mindenki számára világossá vált, miért is kell a PHP alapértelmezett adatbázis-felületét úgy ahogy van elfelejteni. Lássuk hát a megoldást!

A PEAR

A PEAR egy nyílt forráskódra épülő PHP-programkönyvtár, mely PHP- és C-bővítményeket egyaránt tartalmaz. Hasonló a Perl CPAN-jához. Az teszi mégis különlegessé, hogy a PEAR-ben található bővítményeknek egy sor szabálynak meg kell felelniük. Ezek a szabályok ugyanúgy vonatkoznak a kódolási stílusra, a kód tagoltságára, a bekezdések elhelyezkedésére és a behúzások nagyságára, mint az alkalmazások egységesített programozói felületére, valamint a megjegyzések elhelyezkedésére és szerkezetére. Ezeknek a felté-

1. lista A MySQL-adatbázis létrehozása (db.sql)

```
CREATE DATABASE konyvesbolt;
USE konyvesbolt;

CREATE TABLE konyvek (
    sorszam smallint NOT NULL auto_increment,
    darabszam tinyint,
    konyvcim varchar(20),
    PRIMARY KEY (sorszam)
);

INSERT INTO konyvek (darabszam, konyvcim)
VALUES(10, "AkiØrt a harang sz l");
INSERT INTO konyvek (darabszam, konyvcim)
VALUES(15, "Egri csillagok");
INSERT INTO konyvek (darabszam, konyvcim)
VALUES(20, "Zabhegyezi");
INSERT INTO konyvek (darabszam, konyvcim)
VALUES(8, "A PÆl utcai fiæk");
```

teleknek mind a programkönyvtárhoz kapcsolódó C-programoknak, mind a PHP-programoknak meg kell felelniük. Egy alkalmazás csak akkor lehet a programcsomag része, ha minden elvárásnak eleget tesz.

A PEAR telepítéséhez először a csomagkarbantartó programot kell letöltenünk és telepítenünk a következő paranccsal:

```
lynx -source http://pear.php.net/go-pear | php4
```

Megeshet, hogy a PHP4 rendszerünkön a php paranccsal érhető el, ez esetben a következőt kell beírunk:

```
lynx -source http://pear.php.net/go-pear | php
```

Mindkettőhöz az szükségeltetik, hogy rendszerünkön a PHP4 CGI-csomagja elérhető legyen, akkor is, ha a PHP4-et nem CGI-ként, hanem Apache-bővítményként futtatjuk. A csomag neve *php4-cgi*, ami Debian rendszereken az `apt-get install php4-cgi` paranccsal egyszerűen telepíthető, más rendszerek esetén pedig az adott Linux-terjesztéshez adott CD-k valamelyikén találjuk meg. A parancs futtatása után fogadjuk el a felajánlott beállításokat, és kérjük az alapértelmezett csomagok telepítését. Ha rendszerünkön nem a PHP4 CGI-változatát használjuk, a */etc/php4/cgi/php.ini* megváltoztatása nem szükséges, ellenben a */etc/php4/apache/php.ini* fájlban az `include_path` nevű változót módosítanunk kell, hogy tartalmazza a PEAR könyvtárát. Ha a kérdéses sor eddig így nézett ki:

```
include_path = ". "
```

2. lista A beállításokat tartalmazó fájl (config.inc.php)

```

1. <?php
2.
3. $dbhost = 'localhost';
4. $dbuser = 'fulesmacko';
5. $dbpass = 'mezesbodon';
6. $dbname = 'konyvesbolt';
7.
8. ?>

```

3. lista A MySQL közvetlenül (mysql.php)

```

1. <?php
2.
3. include('config.inc.php');
4. // ...
5. $db = mysql_connect($dbhost, $dbuser,
6. $dbpass);
7. // ...
8. mysql_select_db($dbname);
9. $sql = "SELECT * FROM konyvek";
10. $res = mysql_query($sql);
11. while ($row = mysql_fetch_row($res))
12.     printf("%s (%ddb)<br>\n", $row[2],
13. $row[1]);
14. printf("K nyvek száma: %d<br>\n",
15. mysql_num_rows($res));
16. mysql_close($db);
17. ?>

```

akkor erre kell cserélnünk:

```
include_path = "/usr/share/pear:."
```

A telepítést követően rendszerünkön rendelkezésre áll a /usr/bin/pear nevű futtatható fájl, amellyel a PEAR bővítményeit kezelhetjük. A rendelkezésre álló bővítmények listája ➔ <http://pear.php.net/packages.php> címen érhető el, de kipróbálhatjuk az alábbi a parancsot is:

```
pear list-remote-packages
```

A listából kiválasztott elemet a

```
pear install
➔http://pear.php.net/get/csomag_neve
```

parancsral telepíthetjük fel, vagy ha a szükséges csomagot már letöltöttük, a következő is elég lesz:

```
pear install csomag.tgz
```

A Debian-változat legújabb terjesztéseiben a PEAR, illetve az alapértelmezett csomagok az apt-get install libphp-pear parancsral is telepíthetők.

4. lista A PEAR::DB-példa (pear-db.php)

```

1. <?php
2.
3. include('config.inc.php');
4. require_once('DB.php');
5. $db = DB::connect
6. ("mysql://$dbuser:$dbpass@$dbhost/$dbname");
7. // ...
8. // erre itt nincs sz kszög (select_db)
9. $sql = "SELECT * FROM konyvek";
10. $res = $db->query($sql);
11. while ($row = $res->fetchRow())
12.     printf("%s (%ddb)<br>\n", $row[2],
13. $row[1]);
14. printf("K nyvek száma: %d<br>\n",
15. $res->numRows());
16. $db->disconnect();
17. ?>

```

5. lista Példák adatbázis-azonosítókra

```

dbt pus://felhno:v:jelsz @protokoll+gopno:v:
➔555//usr/dbno:v.db
dbt pus://felhno:v:jelsz @gopno:v/dbno:v
dbt pus://felhno:v:jelsz @gopno:v
dbt pus://felhno:v @gopno:v
dbt pus://gopno:v/dbno:v
dbt pus://gopno:v
dbt pus (vÉltozat)
dbt pus
dbt pus://felhno:v:jelsz @tcp(localhost:555)/
➔dbno:v
dbt pus://felhno:v:jelsz @unix
➔(/var/run/dbsocket)/dbno:v

```

Előkészületek

Mielőtt belevágnánk programjaink és elveink megreformálásába, nézzük csak meg, hogyan dolgoznánk, ha adatbázisainkat a PHP felületén keresztül közvetlenül kezelnénk. Ebben az esetben az első dolgunk az, hogy meghatározzuk, milyen adatbázist szeretnénk használni, mert ettől függ a kód többi része. Mi most a MySQL-t vesszük alapul, tekintve, hogy ez az egyik legnépszerűbb adatbázisrendszer (PostgreSQL-rajongóknak: csak a PostgreSQL után!). Hozzunk tehát létre a MySQL-ben egy új adatbázist, és abban egy példatáblát (lásd 1. lista). Ha ezzel megvagyunk, készítsünk egy egyszerű eljárást, amellyel adatbázisunk tartalmát kiíratjuk (2. és 3. lista). Ha ez is kész, továbbléphetünk, és megtudhatjuk, hogyan zajlik mindez a függetlenített adatbázis-elérési módozatokkal.

Adatbázis-kezelés PEAR::DB módra

A 4. listán a MySQL-es példa látható kissé átalakítva. Az óriási különbség a két programkód között az, hogy míg az első esetben a kód mondjuk Oracle-re történő átültetése legalább 5–10 percünkbe kerülne, addig ebben az esetben nem tart tovább néhány másodpercnél. Nem is beszélve arról, hogy az

6. lista A PEAR::DB által támogatott adatbázistípusok

```
mysql -> MySQL
pgsql -> PostgreSQL
ibase -> InterBase
msql -> Mini SQL
mssql -> Microsoft SQL Server
oci8 -> Oracle 7/8/8i
odbc -> ODBC (Open Database Connectivity)
sybase -> SyBase
ifx -> Informix
fbsql -> FrontBase
```

Oracle PHP-s felületének leírását is át kellene rágnunk, mielőtt belekezdünk az új kód megírásába. Itt mutatkozik meg igazán, micsoda kényelmet jelent egy egységesített felület használata.

Vegyük szépen sorra, mit jelentenek az egyes sorok! Először is a 3. sorban beillesztjük az adatbázissal kapcsolatos beállításokat tartalmazó fájlt – ez egyértelmű. A 4. sorban esetleg a `require_once()` tűnhet furcsának. A `_once` módosító arra hivatott, hogy ha a programunk egy más részén már beszúrtuk a kérdéses fájlt, programunkat esetlegesen megzavarva ne szűrődjék be újra. Az `include()` és a `require()` között egyébként csak annyi a különbség, hogy az `include()` csak figyelmeztet, ha nincs meg a keresett fájl, viszont a `require()`, ha nem találja a fájlt, a program futását le is állítja. Az első lényeges különbségek az 5–7. sorban mutatkoznak meg. Míg a PHP alapfelületével szükséges volt külön kijelölni az adatbázist, PEAR::DB esetében egy különleges DSN-t, vagyis adatbázis-azonosítót kell megadnunk. A DSN-nek csak a neve ijesztő, használata jóval egyszerűbb, mint ahogyan azt az 5. lista egyértelműen megmutatja. A 6. lista a PEAR::DB által támogatott adatbázisokhoz tartozó kulcsszavakat tartalmazza, amelyeket a DSN-ek megadásánál vehetünk igénybe.

A 8. és 9. sorban sem találhatunk lényeges újdonságokat: megadjuk a lekérdezést, majd a `$db` objektumon keresztül futtatjuk. A 10. sor már tartalmaz néhány újdonságot, itt tűnik fel a `fetchRow()` tagfüggvény is. Ennek a feladata, hogy az adatbázistól sorban lekérdezze az SQL-utasításra válaszul érkezett mezőket. A 13. sorban található `numRows()` tagfüggvény a lekérdezés által visszaadott sorok számát jeleníti meg, ezt követően pedig a `disconnect()` tagfüggvényvel leválsunk az adatbázisról.

Az adatbázis módosítása az INSERT, UPDATE és DELETE utasításokkal ugyanúgy működik, ahogy korábban megszoktuk, csak ebbe az új környezetbe van beleültetve.

A `fetchRow()` függvényről még tudnunk kell valamit. Sokkal barátságosabbá tehetjük programunkat, hogyha számok helyett az adatbázis egyes mezőire a nevükkel hivatkozunk. Ennek beállítására szolgál a `fetchRow()` tagfüggvény első és egyben el is hagyható kapcsolója, amely alapértelmezésben a `DB_FETCHMODE_ORDERED` értéket veszi fel. Ha azonban az eddigiekkel ellentétben a `DB_FETCHMODE_ASSOC` vagy `DB_FETCHMODE_OBJECT` értékekkel hívjuk meg, a mezőkre a nevükkel is hivatkozhatunk: az első esetben egy nevesített tömb segítségével, az utóbbinál pedig objektumhivatkozáson keresztül. Például:

```
$row = $res->fetchRow(DB_FETCHMODE_ASSOC);
```

7. lista A `prepare()`, az `execute()` és a `quote()` (pear-db-2.php)

```
1. $ujkonyvek = array("AAAA", "BBBB", "CCCC");
2.
3. $sql = "INSERT INTO konyvek (darabszam,
   konyvcim) VALUES(10, ?)";
4. $query = $db->prepare($sql);
5.
6. foreach ($ujkonyvek as $ujkonyv)
7.     $res = $db->execute($query,
   $db->quote($ujkonyv));
8.
9. $ujkonyvek2 = array();
10. $ujkonyvek2[] = array(12, "Szõp
   remõnyek");
11. $ujkonyvek2[] = array(21, "Bæcsæ a
   fegyverektõl");
12.
13. $sql = "INSERT INTO konyvek (darabszam,
   konyvcim) VALUES(?, ?)";
14. $query = $db->prepare($sql);
15.
16. foreach ($ujkonyvek2 as $ujkonyv2)
17.     $res = $db->execute($query,
   DB->quote($ujkonyv2));
```

8. lista Az ADODB-példa (adodb.php)

```
1. <?php
2.
3. include('config.inc.php');
4. require_once('adodb.inc.php');
5. $db = NewADOConnection("mysql");
6. $db->Connect($dbhost, $dbuser, $dbpass,
   $dbname);
7. // erre itt nincs sz kszög (select_db)
8. $sql = "SELECT * FROM konyvek";
9. $res = $db->Execute($sql);
10. for (; !$res->EOF; $res->MoveNext())
11.     printf("%s (%ddb)<br>\n",
   $res->fields[2], $res->fields[1]);
12.
13. printf("K nyvek száma: %d<br>\n",
   $res->RecordCount());
14. $db->Close();
15.
16. ?>
```

```
print($row[ konyvcim ]);
```

vagy:

```
$row = $res->fetchRow(DB_FETCHMODE_OBJECT);
print($row->konyvcim);
```

Ha a `fetchRow()`-nak nem akarjuk minduntalan megadni az általunk előnyben részesített hivatkozási mód nevét, a kö-

vetkező paranccsal alapértelmezetté tehetjük:

```
$db->setFetchMode(DB_FETCHMODE_ASSOC);
```

Hibakezelés

A `PEAR::DB::isError()` statikus – vagyis nem objektumhoz kötött – tagfüggvényével ellenőrizhető, hogy az elvégzett művelet során történt-e valamilyen hiba:

```
$db = DB::connect($DSN);

if (DB::isError($db))
    die($db->getMessage());
```

A hiba leírásával a `getMessage()` tagfüggvény tér vissza. Az adatbázis-lekérdezésnél előforduló hibákat az alábbi módon ellenőrizhetjük:

```
$res = $db->query($sql);

if ($db->isError($res))
    die(DB::errorMessage($res));
```

A `PEAR::DB`-nek egy önműködő szerkezete is létezik a hibák kezelésére. Erről bővebben a <http://pear.php.net/manual/en/> címen olvashatsz, a `DB::setErrorHandler()` tagfüggvényénél.

További lehetőségek

Amennyiben több hasonló módosítást szeretnénk végrehajtani egy adatbázison, jó szolgálatot tehet nekünk a `prepare()` és az `execute()` tagfüggvény. Mindkettő ismerős lehet, hiszen a Perl DBI programozása során már találkozhattunk velük. Vess egy pillantást a [7. listára!](#)

A `prepare()` tagfüggvénnyel egy lekérdezést többszöri használatra lehet felkészíteni, a lekérdezésben változó értékeket pedig az `execute()` tagfüggvény egymás után következő meghívásai során adjuk meg. Az `execute()` első értéke a `prepare()` által visszaadott `$query` változó, melyet az értéket tartalmazó változó vagy – több érték esetén – az értékeket tartalmazó tömb követ.

A `quote()` függvény szerepe nagyon fontos: a segítségével biztosíthatjuk ugyanis, hogy az adatbázisba kerülő értékek mindig az adott adatbázisnak megfelelően kódolódjanak. Az „ez egy 'A' betű” karakterláncban az aposztróft a MySQL esetén egy visszafelé tört nonallal védjük le („ez egy 'A' betű”), míg MS Accessben vagy Sybase-ben egy másik aposztróffal („ez egy ''A'' betű”). Egy jól megtervezett adatbázis-egységesítési rendszerben az ilyesmikre is gondolni kell!

A PEAR::DB hátulütői

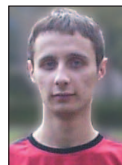
A `PEAR::DB` egyik legnagyobb hibája minden bizonnyal az, hogy még igencsak fejlesztés alatt áll, következésképpen néhány dolog nem működik benne, vagy nem úgy működik, ahogy az ember elvárna. A `PEAR::DB`-ből egyelőre az adatbázisrendszerek kezeléséhez elengedhetetlen dátumkezelés egységesítését megvalósító függvények is hiányoznak. Erre feltétlenül szükség van ahhoz, hogy valódi adatbázis-független kódot írjunk, így remélhetőleg a közeljövőben ennek a megalkotására is sor kerül. Addig pedig – ha a `PEAR::DB` mellett döntünk – saját magunk alkotta függvényekkel vagyunk kénytelenek elvégeztetni ezt a feladatot.

Bemutatkozik az ADODB

Az ADODB ugyanazt a célt szolgálja, mint a `PEAR DB` nevű csomagja. Működési elve és programozói felülete is hasonló, a különbség csupán annyi, hogy az ADODB jóval kiforrottabb kódra épül, és számtalanul többen használják, mint a `PEAR::DB`-t. Az is az ADODB mellett szól, hogy jelenleg sokkal több adatbázisrendszert támogat, mint a `PEAR::DB`. A PHP MySQL-ről átalakított példaprogramunk ADODB-s változata a [8. listán](#) látható. Az elgondolás itt is ugyanaz, mint a `PEAR::DB` esetében, és a két felülethez tartozó kód – a kezelést illetően – lényegében ugyanaz, eltekintve a megvalósítás részleteiben rejlő különbségektől. Az első különbség az, hogy az ADODB esetében a csatlakozás két sorba tagolódik. A második szembevetőbb: a `for`-t használjuk az eddig megszokott `while` helyett. Ez azért szükséges, mert az ADODB az egyes sorok lekérdezése után a sormutatót nem önműködően lépteti tovább, azt magunknak kell a `MoveNext()` tagfüggvény meghívásával megtenni.

Összegzés

Az ADODB és a `PEAR::DB` két jól átgondolt, adatbázis-kezelést egységesítő rendszer, melyek használata minden komoly fejlesztő és cég számára mindenképpen indokolt. Jelen pillanatban az ADODB használata fejlettségének köszönhetően elterjedtebb, ugyanakkor a `PEAR::DB` a PHP `PEAR` projektjének a része – idővel valószínűleg ez fog fölénybe kerülni. Természetesen csak akkor, ha az ADODB-t mind az adatbázisok támogatottságában, mind a leírás használhatóságában utoléri. Az ADODB-hez nagyon jó leírás tartozik, míg a `PEAR`-hez a szép és ügyes `PHPDoc`-ra épülő – ez azonban egyelőre sajnos még nagyon hézagos: teljesen hiányoznak belőle a példaprogramok, a tagfüggvények leírása sokszor egy-egy mondatra korlátozódik, nem is beszélve arról, hogy a dolgok a valóságban gyakorta nem úgy működnek, mint ahogyan elképzelték. Bármelyik megvalósítás mellett döntünk is, végeredményben csak jól járhatunk.



Gludovátz Gábor

(ggabor@sopron.hu) egy soproni cég Linux-rendszerekkel foglalkozó rendszergazdája. Kedvenc időtöltése a programozás, és a Linux lelkivilágában való kutakodás. Ha ideje engedi, szívesen hódol szenvedélyének és

bringáján a környező erdőket járja. Honlapja a <http://www.sopron.hu/~ggabor/> címen érhető el.

Kapcsolódó címek

PEAR <http://pear.php.net>
 ADODB <http://php.weblogs.com/ADODB/>
 PEAR-leírás [http://pear.php.net/manual/en/Különféle PEAR::DB-oktatók címei](http://pear.php.net/manual/en/Kulönféle_PEAR::DB-oktatók_címei)
http://vulcanonet.com/soft/?pack=pear_tut
<http://www.phpbuilder.com/columns/allan20010115.php3?page=1>
http://evolt.org/article/Abstract_PHP_s_database_code_with_PEAR_DB/17/21927/index.html
http://www.devshed.com/Server_Side/PHP/DBAbstraction/page1.html

Együttélés

Az elmúlt hónapban sokat játszottam egy dobozos SuSE Linux 8.0-s Professional kiadással. Telepítettem, próbálgattam, hangolgattam. Amit egy új rendszerrel általában tenni szokás, mind megtettem vele, sőt talán egy kicsivel többet is.

Megtapasztalhattam mindent, amit egy egyszerű felhasználó valaha ki akarna próbálni. Ezeket a benyomásaimat fogom most kedves olvasóinkal megosztani.

A telepítés

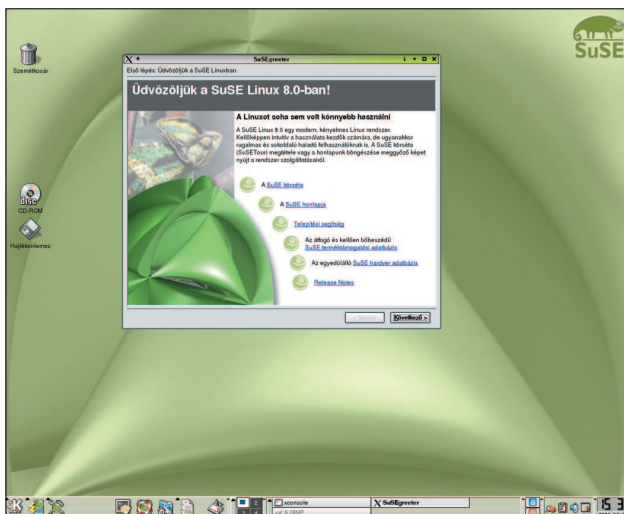
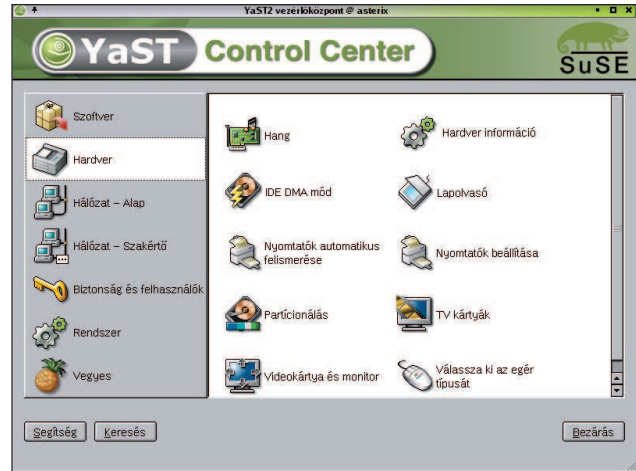
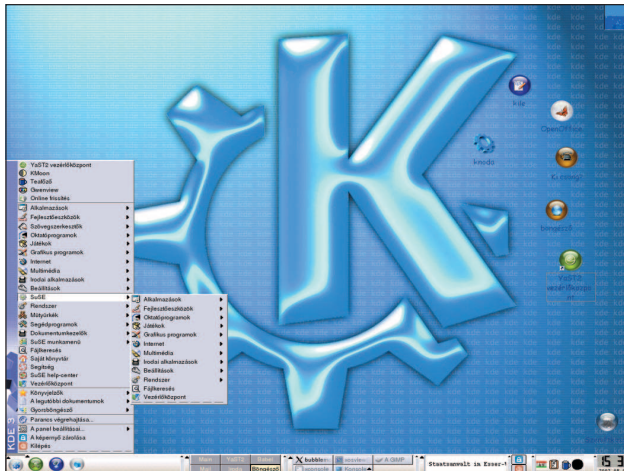
Az első korong behelyezése után a rendszer telepítése a gép elindulásakor elkezdődik. Én a legegyszerűbb telepítési módot választottam, mivel a legtöbben valószínűleg ezt választják. Viszonylag minden rendben zajlott, de a merevlemez felosztásánál már egy négy, illetve egy tíz gigabájtos lemezrészlettel rendelkezem – ezek közül a tíz gigabájtoson tartom az adataimat, amiket egy esetleges újratelepítés után is használni szeretnék. A telepítő azonban csak a nagyobbat jelölte meg mint telepítési célpontot – formázással együtt –, a kisebbik pedig mintha nem is létezne a rendszeremen, tudtomra sem adta, hogy oda is telepíthetném, vagy hogy egyáltalán létezik ilyen. Miután megértettem a telepítővel, hogy ez így nincs rendben, elkezdődhetett a tényleges telepítés. Az előre kiválasztott csomagokat telepíttem, nem változtattam semmit az összetevőkön, így a merevlemez elfoglalt hely 1,7 GB lett (ezt kissé sokallottam, de miután láttam a programok mennyiségét, no meg a KDE 3 sem a pehelysúlyú csomagok közé tartozik, kissé megnyugodtam). A rendszer laptopomban szinte az összes általános alkatrészt hibátlanul felismerte, amit azonban hiányoltam, az az infravörös kapu, mivel így az Internetet nem tudtam rögtön mobiltelefonnal használni. Végző lépésként a rendszerindító program beállítása és telepítése zajlik. Sajnos itt nem adatott meg a választás lehetősége a LILO-GRUB párosból, mindenképpen a LILO-val kellett megbarátkoznom. Mivel a laptop, amire telepíttem, számomra nem nyilvánvaló okok miatt képtelen a LILO használatára, mindenképpen örültem volna a választási lehetőségnek. Indítólemezt készítettem egy hajlékonylemezre, ezzel élesztve fel a rendszert. Ezután telepíttem és állítottam be a GRUB-ot. Laptopom immár teljes értékű rendszerrel bír. Az előre meghatározott csomagok között azonban nem szerepelt olyan program, mint a Gimp vagy a Mozilla. Számomra mindkét program nagyon fontos, annak ellenére, hogy a KDE 3-ban a Konqueror nagyon sokat fejlődött az internetböngészés terén. Én akkor is a Mozillát szeretem használni, mivel ebben szinte mindent egy helyen tudok elvégezni, a Gimpről pedig azt hiszem, nem kell sokat írnom, a hiánya azonnal érezhető volt! A másik furcsaság, hogy két igen nagyméretű irodai csomag is felkerült a merevlemezemre: az egyik a StarOffice 5.2 (megközelítőleg 200 MB), a másik az OpenOffice.org 641-es magyarított változata (nagyjából ugyancsak 200 MB). Ha ebből csak az egyiket telepítettem volna, a rendszer 1,5 GB-on is elfért volna. Tudom, ezeket kézzel is beállíthattam volna, hogy települjenek; illetve azt is, hogy ne tegyék, de úgy vélem, egy kezdő linuxos még azt sem tudja, mi fán terem a csomagok kiválasztása.



Használat

A KDE magyar felületének köszönhetően frissen telepített operációs rendszerét bárki azonnal használatba veheti, a YAST felülete szintén anyanyelvünkön szól hozzánk, így a kisebb rendszerbeállításokat könnyedén elláthatjuk. A grafikus bejelentkezés könnyedén és zökkenőmentesen működik, bejelentkezés előtt kiválaszthatjuk, hogy milyen ablakkezelőt szeretnénk indítani a meglévők közül. Valószínűleg a KDE a legnépszerűbb, és ezt az ablakkezelőt készítették fel arra, hogy a legjobban együttműködjön a SuSE rendszerrel. Az AfterStep ablakkezelőt is kipróbáltam, de sajnos hiába foglal kevés memóriát, ha a telepített alkalmazások nem jelennek meg a menüjében, csak a gyárilag beállított értékek szerepeltek. Ha rendszer-karbantartási feladatot szeretnénk végrehajtani, a YaST ikonra kattintva előugrik egy rendszergazdai jelszót kérő ablak. Bár megadhatjuk, hogy jegyezze meg a jelszót, erről mégis mindenkit lebeszélnek, mivel nem biztonságos. Ezután a YaST-ban szinte mindent beállíthatunk. Egyik fő erőssége a csomagok frissítése (szerintem) – frissíthetünk az Internetről, CD-ről, de merevlemezről is (a CD-ről történő frissítés leírását a programvadászat oldalon találod meg). Nagyon kellemes a magyar nyelv használata, mivel aki rendelkezik némi számítógépes vénával, de nem beszél sem a német, sem az angol nyelvet, az is hamar képes ráérezni a rendszer előnyeire. A csomagok leírásának magyartása azonban még nem az igazi, ezért böngészésük közepette elkél egy kis angoltudás.

A SiS VGA-meghajtójának zártsága miatt a grafikus felület sajnos nem sikerült tökéletesen beállítanom, mivel hiányzott a 3D-támogatás. Ezt csak az Interneten való keresgélés után, a SiS oldaláról letöltött meghajtó és néhány leírás elolvasását követően sikerült üzembe helyezni, az irodai munkához azonban szükségtelen a 3D-támogatás. Nyomatatási rendszernek a CUPS-rendszert (Common Unix Printing System) választottam, mivel ez könnyedén felismerte a hálózatban fellelhető megosztott nyomtatókat, és elsőre mindent jól állított be, így kézzel semmihez sem kellett hozzányúlnom. Ez különösen azoknak a felhasználóknak jelent segítséget, akik nem szeretnének elmélyedni a hálózati beállításban, nem kívánnak minden egyes hálózati megosztásról pontos adatokat szerezni, mégis minden előnyét ki szeretnék aknázni. Nagyon érdekesnek találtam, hogy Samba-megosztásokat csak a rendszergazda fűzhet be a könyvtárszerkezetbe, ami biztonsági szempontból jó, de ha egy felhasználó egy távoli könyvtárat szeretne használni, azt bizony csak a rendszergazda jelszavának ismeretében teheti meg (vagy szól a rendszergazdának). Mivel laptopra telepíttem, a rendszer energiatakarékosságára is kíváncsi voltam. Eddig nem volt rossz tapasztalatom ezzel a lehetőséggel, viszont a rendszer a számtalan gépindítás közül háromszor annyira takarékoskodott



az energiával, hogy a processzorhűtő ventilátort elfelejtette bekapcsolni, ami nyáron különösen hamar lefagyáshoz vezet, ezért ennek a kezelését inkább a BIOS-ra bíztam.

Összegzés

A dobozos változat megvásárlásával alapján véve egy leírásokkal megfelelően ellátott rendszert kapunk kézhez. A több mint nyolcszázoldalas magyar nyelvű leírás végigvezet a telepítésen, a beállításon, a rendszer finomhangolásán és az ablakkezelők (KDE, Gnome) megismerésén (a leírásban ezt grafikus munkakörnyezetnek hívják). Megismerkedhetünk az alapvető programok használatával, így az OpenOffice.orggal, a Windowsból már biztosan ismerős pdf-olvasóval, az Adobe Acrobat Readerrel, a KDE és a Gnome böngészőivel, a KMail levelezőprogrammal, a KOnCD CD-író programmal, a Gimp képszerkesztő programmal, a Kooka lapolvasó felülettel, a hangkezeléssel és a tévé, videó, rádió és webkamera kérdéskörével is. A harmadik és egyben legvaskosabb könyv a rendszer lelkéhez közelebb álló tudományt csurgatja az ember fejébe, a főbb fejezetcímek (idézve a könyvből):

- Telepítés YaST2-vel
- Különleges telepítési módok
- Rendszerfrissítés és csomagkezelés
- YaST2 – beállítási eszközök
- Rendszerbetöltés és rendszerbetöltő programok
- Az X Window rendszer
- A grafikus felület beállítása

- A botkormány beállítása
- Hangkezelés az ALSA-val
- Nyomtatás
- A rendszermag és paraméterei
- Rendszerjellemzők
- A SuSE Linux rendszerindítási elgondolás
- Linux a hálózatban
- A Squid proxy szerver
- Hálózati biztonság
- A GNU GPL

Mindez közel ötszáz oldalon – az oldalszámból sejteni lehet, hogy kellő mennyiségű adatot tartalmaz a rendszer megismeréséhez, karbantartásához, és mindezt úgy teszi, hogy a dokumentáció teljességgel a feltelepített rendszerhez íródott. Így sokkal kevesebb kinnal tanulhatjuk meg mindazt, amit egy általános Linux-könyv megvételénél bizony szinte minden kezdő pingvinista megtapasztalhat. Mindehhez a

☞ <http://sdb.suselinux.hu> weboldalon található ingyenesen használható adatbázis tartozik, ahol választ kaphatunk a kérdésekre, illetve a termék mellé egy 90 napos ingyenes telepítési segítség is jár.



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

Kapcsolódó címek

- A hivatalos magyar oldal ☞ <http://www.suselinux.hu>
- A hivatalos német oldal ☞ <http://www.suse.de>
- A hivatalos angol oldal ☞ <http://www.suse.de/en/>
- Alkatrész-adatbázis ☞ <http://hardwaredb.suse.de/>
- Terméktámogatási adatbázis ☞ <http://sdb.suselinux.hu/hu/sdb/html/>
- Programkarbantartás ☞ <http://support.suse.de/psdb/>
- Biztonsági bejelentések ☞ <http://www.suselinux.hu/biztonsag/>

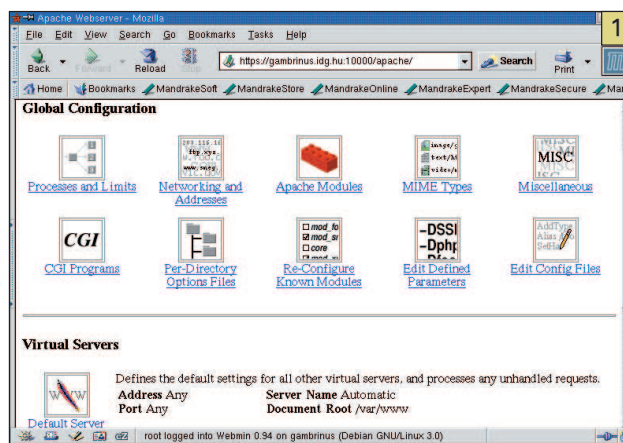
Webkiszolgálók harca

A cikkben a Linux alatt is futó webkiszolgálókat veszem górcső alá, ezek közül is azokat, amelyek szabadon elérhetők. Lássuk, miből gazdálkodhatunk, és melyik kiszolgálónak mi az erénye, illetve a hátránya.

Rögtön tisztázzuk, hogy bizonyos alkalmazások nemcsak webkiszolgálóként, hanem alkalmazáskiszolgálóként is működnek, esetleg saját beépített webfejlesztési eszközeik vannak. Az ismerkedést kezdjük a jó öreg Apache-csal, és tekintettel a népszerűségére, egy kicsit többet foglalkozunk vele.

Apache

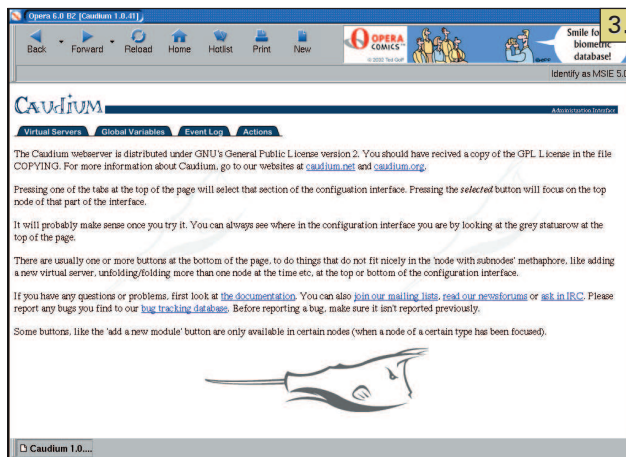
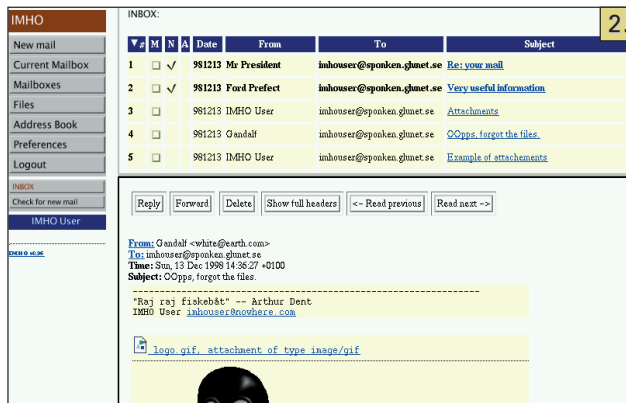
Az Apache a világ legtöbbet használt webkiszolgálója, az összes kiszolgálóprogram körülbelül 55–60 százalékát ez a program teszi ki. Jelenleg a 2.0-s sorozat a legújabb, ami jelentős megújuláson ment keresztül az előző 1.3-as sorozathoz képest. Teherbírása és bővíthetősége legendás. Bővíthetőségét modulalapú felépítésének köszönheti, és él is ezzel a lehetőséggel. A program fordításakor mi dönthetjük el, hogy mi kerüljön magába a webkiszolgáló szerepét ellátó alkalmazásba, és mi modulba. Ha úgy kívánjuk, az összes modult belefördíthatjuk a programba, ennek azonban semmi értelme, mert csak lassabban fog elindulni, és egy csomó olyan modul lesz benne, ami felesleges. Milyen újdonságokkal rendelkezik, illetve hogyan is működik a 2.0-s Apache? Nos, az előző sorozathoz képest előrelépés, hogy már nem folyamat- (process), hanem szálalapú (thread) futtatásra is lehetőség nyílik. Miért előnyös ez? A weblapok lekérésekor legalább egy új folyamatnak el kellett indulnia, ilyenkor természetesen újra be kellett tölteni a demont a memóriába, következésképpen pazarolta az erőforrásokat, és lassabb volt az indulás. Ezzel ellentétben a szálak közös memóriaterületet használnak, és egy folyamaton belül több kiszolgálódémon is elindulhat. Ekkor tehát az Apache kevesebb memóriát használ, és gyorsabban szolgálja ki a kéréseket. A szálak programozásánál ugyanakkor körültekintően kell eljárni, mivel súlyos gondok adódhatnak, ha elrontja az ember, és ezt nem mindegyik felület támogatja megfelelő módon. Érdemes hát ezt a módszert használni a webkiszolgálónál. A fejlesztők viszont a választás lehetőségét meghagyják a rendszergazdáknak, hogy a megbízhatóságot és a kipróbált módszert választják-e – hiszen az Apache ebben is nagyon jól teljesített –, esetleg megkockáztatják a még kevesebb ideje használatos kódot, és valóban nagyobb teljesítményt érnek el. Hogyan lehetséges ez? Az 1.3-as ág fejlesztésénél a fejlesztők a POSIX-szabványokon alapuló megoldást használták, ahol pedig ezt megvalósító réteg nem áll rendelkezésre az operációs rendszerben, ott egy POSIX-emulációs réteget húztak fel a rendszerre. Ez néhol kevésbé megbízhatóvá és lassabbá tette a rendszert. A 2.0-s ág a hálózati kapcsolatok kezeléséhez és a fájlrendszer eléréséhez már a futtató operációs rendszer által használt natív rendszerhívásokat használ. A natív rendszerhívások lehetőségét a fent említett MPM-ek (Multi-processing Modules) és az APR (Apache Portable Runtime) biztosítja. Az MPM-ek teszik lehetővé azt is, hogy a rendszergazda kiválaszthassa, milyen módon használja a kiszolgálót – a régi, folyamatalapú, vagy az új, kevert, több folyamatot és több



szálakat használó, vagy egy harmadik, úgynevezett Perchild modult használva. A Worker MPM az a modul, ami a kevert megoldást alkalmazza. Elindul egy folyamat, és az ügyfelek egy bizonyos kérés számig ezen belül szolgálják ki a rendszert. Ha a kérések túllépeik ezt a számot, egy újabb folyamat indul, és kéréseket azon belül lehet ismét a meghatározott számig kiszolgálni. Mind a folyamatok, mind a folyamatok belüli szálak legnagyobb mérete megadható. A Preforking MPM a régi Apache-megközelítést használja, a már fent leírt egy démon folyamatmegközelítést. Itt a kiszolgálható ügyfelek legnagyobb számát az engedélyezett folyamatok száma határozza meg. Az erőforrások tekintetében a drágább indulásért és a nagyobb igényért cserébe nagyobb megbízhatóságot kapunk. A Perchild MPM pedig egy olyan új, szintén kevert megoldást alkalmazó rendszer, amelynél folyamatonként beállítható, hogy milyen felhasználói azonosító alatt működjön. Ezzel a felhasználókat teljesen el tudjuk határolni egymástól. Ez a modul nem minden felületen érhető el, és a fejlesztők is felhívják rá a figyelmet, hogy bár már üzembiztos, még fejlesztés alatt álló modulról van szó.

Az Apache felügyeleti eszközökkel is jól el van látva, amelyek közül a Webmin – mint nyílt forrású megoldás – külön kiemelhető (1. kép).

Az Apache felhasználási szerződése mind az üzleti felhasználást, mind az olyan megoldások fejlesztését lehetővé teszi, amelyek már nem lesznek nyílt forrásúak. A világon a legismertebb cég, amely Apache-csal foglalkozik és kereskedelmi tevékenységet folytat, a Covalent. Nemrég olyan megállapodást kötöttek a Microsofttal, amelynek értelmében az ASPNet-alapú alkalmazások Apache alatt is futtathatók lesznek. A kiadott leírás csak a Windows operációs rendszerről szól, de ne feledjük, hogy egy modulal már most is lehetőség nyílik .asp kiterjesztésű weblapokat futtatni Apache alatt, bár ez tényleg nem rendelkezik a .NET lehetőségeivel. Ez az Apache legnagyobb erőssége:



a modulokon keresztül nemcsak az igényeinknek megfelelően finomhangolható, de gyakorlatilag nincs olyan parancsnyelv, amit ne támogatna. Az Apache projekt azonban ennél sokkal szélesebb körű: szerepel benne Java-alapú webkiszolgáló, illetve alkalmazáskiszolgáló, XML-RPC- és SOAP-megvalósítás is – ezekről egy másik számban bővebben írok majd.

Boa

Ha valakinek egyszerű, gyors, CGI-eket is futtatni tudó webkiszolgálóra lenne szüksége, a Boa a számára megfelelő kiszolgáló. Futása közben csak akkor indít magából újabb példányokat, ha CGI-t kell futtatnia, máskülönben a már futó binárison belül oldja meg a kért oldalak kiszolgálását. A már futó folyamat önmagán belül többszörözi a HTTP-csatornákat. Sebessége is lenyűgöző: a tesztek alapján egy 300 MHz-es gépen is több ezer kapcsolatot képes kezelni! Fő célja a sebesség, és nem utolsósorban a biztonság. Fejlesztése már 1991-ben elkezdődött, ezenkívül beállítófájla nagyon hasonlít az Apache-éhoz. Ha ennyi mindent tud egy kis bináris, akkor miért nem ismert és népszerű? Például azért, mert nem támogatja a kiszolgálóoldali parancsnyelveket, például a PHP-t. A legkritikább esetben gyorsabb vagy könnyebb C-alapú CGI-ben írni meg egy dinamikus weboldalt, mint PHP-ban, nem meglepő tehát, hogy a rendszergazdák legtöbbször más megoldásokhoz fordulnak.

Roxen

A Roxen Webchallenger a második legnépszerűbb nyílt forrású program a világon. Megbízhatóságáról sok mindent elárul, hogy a <http://www.bsd.hu> is ezt használja (az élet furcsa fintora: a Roxen GPL felhasználási szerződésű, a fejlesztők

ugyanis nem szeretik a BSD felhasználási szerződést). Különlegessége, hogy nagy részét a Pike nevű parancsnyelvben írták – erre sok más tekintetben is támaszkodik –, és folyamatok helyett a kezdetektől fogva szálakat használt. Ami számos rendszergazda szemében vonzóvá teheti, az a weben keresztüli felügyelhetőség. Ez egyben legnagyobb hátránya is: amikor először nézünk bele a beállítófájlba, valószínűleg még a lélegzetünk is eláll, és nem kis időt vesz igénybe, mire elsajátítjuk a felület kezelését. Az XML ugyanis nagyon szép lehetőségeket kínál az adatok megjelenítésére, de ha egy beállítófájlban ebben tárolnak, és azt nekünk kézzel kell szerkeszteni, nos... Én inkább maradok az Apache szemléleténél – egyszerű beállítófájl, ha valaki nem ismeri, grafikus beállítóprogramot kap. Nos, a Roxennél meg sem próbálnék kézzel beállítani a kiszolgálót. A Pike nyelven keresztül kitűnően és gyorsan lehet olyan weboldalakat összeütni, amelyek *.rxml* kiterjesztéssel rendelkeznek. Figyelem! Csak a Roxen Webchallenger tartozik GPL felhasználási szerződés alá, és a Roxen honlapján (<http://www.roxen.com>) szinte csak a Roxen CMS-ről esik szó – ez teljes fejlesztési és vezérlőfelületet ad az oldalhoz, de kereskedelmi termék. A letöltéseket a Download menüpont alól tehetjük meg, a legutolsó változat a 2.2-es. Érdekessége, hogy egy hozzá fejlesztett webmail tartozik mellé, ami a legjobbak egyike, az IMHO (2. kép).

Caudium

A Caudium a Roxen 1.3-as sorozatából indult ki mint külön fejlesztési ág, kihasználva a GPL adta lehetőségeket. A történet valamikor még az 1.3-as Roxen idejében kezdődött, amikor is a fejlesztők úgy gondolták, hogy nem raknak vissza minden foltot (patch). Elég sok külső és néhány belső – értsd: céges – fejlesztő úgy gondolta, hogy külön utakra lépnek, és nem követik a cég elképzeléseit és jövőbeni terveit. Így született meg a Caudium. Sok mindenben hasonlít a Roxenre, abban is, hogy saját webmail programmal rendelkezik, amelynek neve CAMAS. Ez is külön fejlesztési ág, az IMHO-ból jött létre a Caudiumhoz igazítva. További megegyező tulajdonságuk, hogy csak webfelületen keresztül érdemes beállítani őket (3. kép). Ha telepítjük a támogatást, PHP-oldalaink is működni fognak alatta.

Zope

A Linuxvilágban már sorozat is megjelent róla, így sokaknak ismerős lehet. Beépített webkiszolgálóval rendelkezik, a fő hangsúly azonban mégiscsak a tartalomszolgáltatáson van, amit rendkívül magas szintű nyelven lát el. Már készülő hozzá a sablonértelmező, amelynek segítségével a kiszolgált oldalak küllemének cseréje könnyedén megoldhatóvá válik, és a megjelenítés sokkal jobban elválasztható lesz magától az alkalmazástól. Ehhez részletesebben lapozzuk fel a Linuxvilág régebbi számain (15–19. szám).



Deim Ágoston (ago@lsc.hu)

Kedveli a sört, szereti a futást és imádja Szabó Lőrinc verseit. Nem hisz vakon egyik rendszerben sem. Vonzódik a BSD-hez is. Tagja az LME-nek és a MBE-nek. Mottója: a gép nem lehet fontosabb az embernél.

- ➔ <http://httpd.apache.org>; ➔ <http://www.boa.org>
- ➔ <http://www.roxen.com>; ➔ <http://www.caudium.org>
- ➔ <http://www.lysator.liu.se/~stewa/IMHO/>

Miért Linux?

Reuven visszatér az alapokhoz, és áttekintést ad a Linux üzleti világban való fejlődéséről.

Bontsunk pezsgőt! A Linux Journal ebben a hónapban jelenik meg századszorra. Ennek öröme egy kicsit félretereszem szokásos nyílt forrású webes-adatbázisos témáimat és csatlakozom az ünneplőkhöz.

Számos okuk van a Linux felhasználóinak (és általában a nyílt forrású programok híveinek) az öröme. A csúcstechnológiára épülő gazdaság ugyan visszaesett, a nyílt forrású programok fejlesztése azonban továbbra is nagy lépésekkel halad előre. Amikor a Linux Journal első száma megjelent, még kevés ember hallott a finn egyetemista által írott rendszermagról, amely komoly szerepet játszott a Linux és a szabad programok sikerében.

Manapság sokan tudnak a Linuxról, még ha nem is értik meg, hogy mi az és mire jó. Sok ügyfelem tudja, hogy a nyílt forrású megoldásokat részesítem előnyben, és mindig kíváncsiak arra, hogy miért van ez így, és ami még fontosabb, miért érdekük nekik is, hogy ilyen megoldást válasszanak. Vállalva annak a kockázatát, hogy a már hívó emberek között tartsak hittérítést, ebben a hónapban áttekintem azokat az okokat, amiért a Linux kiváló választás a kiszolgálóoldali alkalmazások futtatásánál. Remélem, néhány itt leírt ötlet a jövőben segíteni fog a szabad programokon alapuló megoldások népszerűsítésében munkatársaid és ügyfeleid körében.

Ár és megbízhatóság

A megszállott szakembereket a módszerek és az eszközök érdeklik, amelyek új tudáshoz és új távlatokhoz vezetnek. A való világban viszont az embereket az érdekli, hogy munkájukat a lehető leggyorsabban és legolcsóbban végezzék el. A program csak eszköz a cél eléréséhez, nem maga a cél. Emiatt úgy vettem észre, hogy a legjobban úgy adható el az embereknek a nyílt forrású program, ha azt mondjuk: kevesebbe kerül és többet tud. Egyik tényező sem elég önmagában – könnyen találhatunk drága, jó minőségű programokat, valamint telepítésre sem méltó, ócska és olcsó programokat. A fogyasztók, és így az ügyfeleim is, mindig szívesen szereznek többet kevesebbért, és emiatt a szabad programok tetszenek nekik.

Amikor az ügyfélnek vázolom a megoldást, elkezdem elmagyarázni, hogy olyasmit ajánlok, amit ők lehetetlennek hittek: a szabad és olcsó programok pontosan azt teszik, amit elvárnak tőlük, ráadásul le sem fagynak. Amikor elmondom egy Windows-felhasználónak, hogy még sohasem láttam Linuxot lefagyni hat év és több tucat rendszer után, meglepődnek és hitetlenkednek. Ha elmondom azt is, hogy ez a program szabadon letölthető az Internetről, még nehezebben hiszik el. Ügyfeleim gyakran érdeklődnek, hogy ki támogatja a programot, mi lesz, ha valami elromlik. Megkönnyebbülve veszik tudomásul, hogy nemcsak én ajánlom fel a szükséges támogatást, hanem más is képes erre, ha az én munkámmal nem lennének elégedettek. Ez természetesen szöges ellentétben áll számos tanácsadó cég programtelepítésekkel kapcsolatos viselkedésével és megszorításaival. A nyílt forrású megközelítés

ezért felhasználóbarátabb, mint a hagyományos programfejlesztési modellek: csökkenti a költségeket és erősíti a versenyt.

Természetesen nem minden szabad program jó minőségű, és nem minden tanácsadó tudja, hogy mit kell tenni. A közösségi fejlesztés kitűnő eredményeket hozhat, de ez nem azt jelenti, hogy minden Interneten megjelenő program biztonságos és megbízható. Valójában teljesen világos, hogy sok programot, beleértve egy-két népszerűt is, egyáltalán nem próbáltak ki a kiadás előtt. Az ilyen programok az egész nyílt forrású közösséget lejáratták, és több kárt okoznak, mint amennyi hasznot hajtanak. Évente többször előfordul, hogy felhív az ügyfél, hogy ki kellene javítani egy programot, amelyet letöltöttek, és első ránézésre működött is, de végül kiderült, hogy nem biztonságos, nem megbízható, és tele van hibákkal.

A hibák javítása

Még ha úgy is találd, hogy a kiszolgálód hibás és nem biztonságos nyílt forrású alkalmazásoktól függ, nincs minden veszve. A szabad programok természetéből következően adott a lehetőség, hogy igényeid szerint módosítsd vagy javítsd, ha valami nem tetszik. A megosztott forrású programok felhasználói szerződéseinek kitalálói nem értik a lényegét, amikor megengedik a forráskód megtekintését, csakhogy nem engedik meg a módosítását vagy a javítását. Ha veszek egy házat vagy egy autót, akkor tökéletesíthetem és meg is javíthatom, ha akarom; miért lenne ez másképp a programokkal?

Az igaz, hogy a megosztott forráskód lehetővé teszi a kód tanulmányozását, így a megbízhatósági és a biztonsági hibák gyorsabban előkerülnek és kijavíthatók. A forráskód olvasásának lehetősége azonban közel sem olyan fontos, mint a továbbfejlesztés lehetősége. Ráadásul a továbbfejlesztett változat visszajuttatása a közösségi változatba azt eredményezi, hogy mindenki hasznosítja az egyén javítását, és további fejlesztésre nyílik lehetőség. Ezért a közösségi munkában való részvétel minden nyílt forrású program felhasználójának érdeke; ez nem egyszerűen csak szép dolog.

Mivel általában kiforrott eszközöket használok – például Linuxot, Apache-ot, Perl-t és Python-t –, viszonylag ritka, hogy hibát találjak a letöltött programokban. Évente többször is előfordul azonban, hogy az általam használt programokban valami hiányosságot fedezek fel. Mivel hozzáférék a forráskódhoz, a lehető leggyorsabban meg tudom változtatni, és ez azt is jelenti, hogy ezzel a gonddal másoknak már nem kell megbirkózniuk.

Érdekes, hogy ez az érv még ma is használható, tekintve, hogy a nyomtatómeghajtókkal kapcsolatos hasonló gondok indították arra *Richard Stallman*-t, hogy megalapítsa a Free Software Foundation-t, amely a GNU-projektet döntő szerepet játszott a GNU/Linux és a szabad program sikerében. Az is csodálatos, milyen hamar hozzászoktunk, hogy a forráskód



elérhető, és a számítógépes rendszer minden pontját tetszés szerint megvizsgálhatjuk vagy megváltoztathatjuk. Hasonlóképpen a Linux-terjesztések is egyre inkább teljes rendszerré válnak. Nemrég kezdtem el dolgozni egy projekten, amelynek Solarison kell futnia, és rögtön rájöttem, hogy mennyivel gazdagabb és jobban felépített egy átlagos Linux-terjesztés, mint például a Solaris alapkiépítésben. Fél napig is eltart, amíg letöltöm és telepítem az összes szükséges programot, mint a gcc, a Perl, a Python stb. Az elmúlt években hozzászoktam, hogy minden környezetemben levő gépen ott van a gcc, ezért úgy éreztem magamat, mint akit visszadobtak a Unix sötét középkorába.

Nincsenek titkok

A mérnökök nem tudnak titkot tartani, erre *Scott Adams* is rámutatott a Dilbert-képregényekben. Valóban, az egyik dolog, amit nagyon vonzóknak találók a nyílt forrású programokban, hogy nincsenek titkok. Az ügyfelek azért bérelnek fel engem, mert időt akarnak megtakarítani maguknak, vagy mert nem értenek egy bizonyos területhez, de nem azért, mert rákényszerülnek. Ebből az okból kifolyólag munkámat az ügyvédi vagy könyvelői munkához hasonlítom tekintem, hiszen tanácsot és megoldásokat ajánlok fel nyilvánosan elérhető tudásanyag alapján.

A „nincs titok” eszmeisége jól működik az ügyfeleimnél, még azoknál is, akiket egyáltalán nem érdekel a programok működése. Tudják, hogy kérdezhetnek, és én a legjobb tudásom szerint válaszolok, nem kell marketingrizsával, kötelező frissítésekkel és kétértelmű beszéddel fárasztanom őket. Műszaki érdeklődésű ügyfeleim élvezik, hogy ha akarnak, belenyúlhatnak a kódba és elolvashatják a leírást; csak azért nem tudnak annyit, amennyit én tudok, mert nincs hozzá idejük és elég tapasztalatuk.

A nem hasznosuló ügyfelek sok tekintetben tökéletes célpontjai a nyílt forrású programoknak, és sokszor fellelkesednek, ha ilyen eszközöket használhatnak. Különösen az oktatási intézményeknek tetszik a tudás megosztásának és a közösségi munkának az eszméje. Hathatós érv, ha elmondjuk nekik, hogy pénzt takaríthatnak meg, és a hasonló gondolkozású emberek közösségébe tartozhatnak. Ráadásul a nem hasznosuló szervezetek nem érzik fontosnak, hogy a változtatásokat a cégen belül tartsák, így könnyebben részt vehetnek a nyílt forrású közösség munkájában.

Jó minőségű eszközök

Amikor cikkeket kezdtem írni a Linux Journalba, a legtöbb kiszolgálóoldali alkalmazás kézzel írott CGI-program volt. Rengeteg webhely használ még ma is ilyen programokat. De ahogy a web egyre bonyolultabbá vált, az emberek olyan eszközöket kezdtek igényelni, amelyekkel könnyen és gyorsan lehet jó minőségű és méretezhető webalkalmazásokat készíteni. Nem túl meglepő módon seregnyi zárt forrású kereskedelmi program jelent meg, hogy kielégítse ezeket az igényeket. Az már viszont felháborító, hogy mennyi pénzt kérnek el ezekért a programokért – ráadásul kötelező szerződést kötni velük, hogy az ő tanácsadóikat bérelj fel a testreszabáshoz. Szerencsére a nyílt forrású világ válaszolt a kihívásra. Számos nyílt forrású eszköztár áll rendelkezésre bonyolult kiszolgálóoldali alkalmazások készítéséhez. A Zope, mint az elmúlt hónapokban láttuk, egy rendkívüli (bár bonyolult) alkalmazás-kiszolgáló, amelynek segítségével adatbázisokhoz és más adatforrásokhoz kapcsolódni képes webalkalmazásokat készíthetünk. A következő hónapban megnézzük az OpenACS-t,

amellyel az online közösségi rendszerek létrehozása és módosítása válik egyszerűbbé. Továbbá az Apache Software Foundation által támogatott `mod_perl`, Mason és számos Javához és XML-hez kapcsolódó eszköz egyre inkább azt a kedvezőtlen folyamatot gerjeszti, hogy a megfelelő eszköz kiválasztása a telepítéssel és a használatával összemérhető nehézségű feladattá válik.

Bármily csodálatosak is legyenek ezek az eszközök, ne feledjük, hogy nem mindenkit lehet a mi oldalunkra állítani. A legdurvább élményem ezzel kapcsolatban az, hogy tavaly egy leendő ügyfél nem engem választott egy egyszerű tartalomkezelő rendszer megvalósítására, amely a termékeit jelenítette volna meg a weben. Megtudtam, hogy az ajánlatom 800 000 dollárral volt kevesebb, mint a legközelebb álló versenytársamé. Mivel azonban én nyílt forráskódra építettem, a versenytárs neve viszont ismert volt tartalomszolgáltatás világában, vesztettem (az ügyfél azóta sorozatos elbocsátásokra kényszerült, negyedéves eredményei egyre romlottak, és weboldalát még mindig kézzel rakják össze, így legalább némi erkölcsi győzelmet elkönnyelhetek).

Azt is meg kell jegyeznünk, hogy a nyílt forrású világ nem minden résztvevője tartja be a közösségnek tett ígéreteit. A nyílt forrás számos támogatója meglepődött és csalódott, amikor a Lutris tavaly „bezárta” az addig nyílt forrású Enhydra Enterprise Java alkalmazáskiszolgáló kódját. Szerencsére vannak helyettesítő termékek, nemcsak a GPL-es JBoss alkalmazáskiszolgáló tett szert óriási népszerűsége az elmúlt év során, hanem nemrég a Sun is bejelentette, hogy a nem üzleti, nyílt forrású J2EE-megvalósítások is hivatalos tanúsítványt kaphatnak a következő hónapokban. Ez csökkentheti néhány üzleti szereplő idegenkedését a nyílt forrású programokkal szemben.

Még ha érnek is kudarcok, ne hagyj magadat becsapni: az IBM, a HP és még a Sun is elismeri ma már, hogy a Linuxot és a nyílt forrású programokat komolyan kell venni, mert sokoldalúak és megbízhatóak. A „világuralom” ideje még nem jött el, de a márkanévet már elismerik, a pénzügyi tényezők és az akadémiai és üzleti világból érkező elismerés segít minket a továbblépésben.

Boldog születésnapot!

Írásomat azzal fejezem be, hogy az újság létrejöttét lehetővé tevő keményen dolgozó munkatársakat elismerés és nagyrabecsülés illeti. A Linux Journalt egy évvel azelőtt kezdtem olvasni, hogy az első cikkem megjelent benne, és azóta is minden hónapban olvasom. A cikkek a világ legalkotóbb programozóinak sokféle érdeklődési területét mutatják be. A tény, hogy a Linux Journal a 100. számához érkezett, minden kétség nélkül azt jelenti, hogy a Linuxnak és a nyílt forrású programoknak helyük van ezen a világon. Remélem a 200. számba is írhatok egy hasonló cikket!

Linux Journal 2002. augusztus, 100. szám



Reuven M. Lerner

(reuven@lerner.co.il) kisebb webes és internetes módszerekkel foglalkozó tanácsadó cég tulajdonosa és vezetője. A cikk megjelenésének időpontjában valószínűleg már végleg elkészült Core Perl című könyvével, melyet idén jelentet meg a Prentice-Hall. Az ATF honlapon érhető el (☞ <http://www.lerner.co.il/atf/>).

Zendítsünk rá és ünnepeljünk!

Marcel ezúttal két médialejátszót ismertet, és azt is bemutatja, hogyan lehet rájuk különböző bőrköket húzni.

Öröm téged újra látni, François, mon ami! Bizom benne, hogy becsülettel gondját viselted az étteremnek, mialatt én fergeteges európai körutamon vettem részt.

Köszönöm kérdésedet, valóban nagyszerű volt az utazás. Oui, az ételek is kitűnőek voltak, de sajnos egyik étterem sem dicsekedhetett hozzád hasonló pincérrel.

Láttam a zene és a fények városait: Londont, Salzburgot, Rómát, Firenzét és még sok más rendkívüli helyet. És a sorból természetesen nem hiányozhatott a legcsillogóbb város, Párizs sem. Ez valahogy illik is ahhoz az ünnepi alkalomhoz, amelyhez a mai menüt kell összeállítanom: a Linux Journal századik, ünnepi kiadásához. Mais oui, François, mivel is lehetne jobban ünnepelni, mint zenével és fényekkel? Gyorsan, még mielőtt a vendégeink megérkeznek, szaladj le a pincébe, és hozd fel, kérlek, az 1989-es Pessac Léognant.

De hiszen már itt is vannak a vendégek! François, siess! Foglaltok helyet, mes amis, helyezétek magatokat kényelembe. Linux-konyhánk mai menüje egy különleges alkalomhoz kötődik. A Linux Journal megérte a 100. számát, ami már jó évjárnak számít. Vessünk egy pillantást a menü első fogására, erre az XMMS-nek nevezett programocskára. Alapjait tekintve szabályos médialejátszóról van szó, az XMMS mégis több egy egyszerű zenelejátszónál. Közelebről megvizsgálva látványos fénybemutatóval is megörvendeztet minket. Támogatja az Ogg Vorbis, MP3 és WAV formátumokat, és megfelelő bővítményekkel RealAudio-fájlok lejátszására is használható.

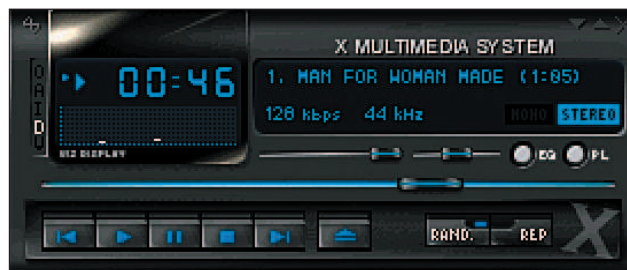
Nem kell túl messzire mennünk érte, hiszen minden elterjedtebb Linux-rendszercsomag tartalmazza. Ha a pillanatnyi telepítéseknek nem része, próbálkozzatok a rendszercsomag CD-ivel, ha pedig minden mással kudarcot vallottatok, a <http://www.xmms.org> címről tölthetitek le a legújabb és legjobb forráskódot.

Az XMMS fordítása egyszerű, ismerős lépéseken keresztül valósítható meg. A jelenlegi változat a GTK 1.2.2 vagy későbbi változatára épül. Szükségek lehet az OpenGL vagy Mesa könyvtárakra, amennyiben az OpenGL-bővítményeket akarjátok használni (erről később bővebben is szó lesz). A forráskódból kiindulva az alábbi lépéseket kell követni:

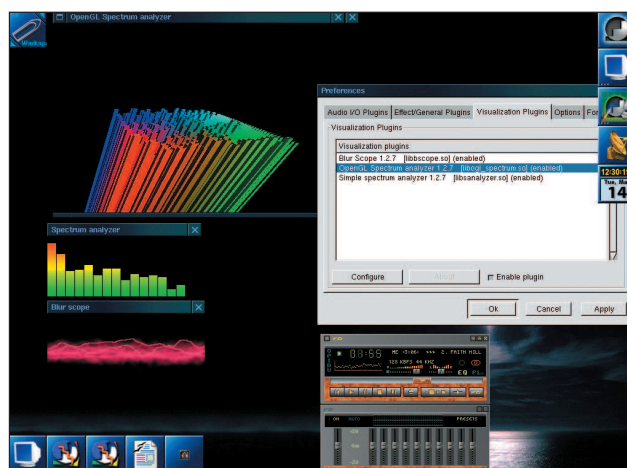
```
tar -xzf xmms-1.2.7.tar.gz
cd xmms-1.2.7
./configure
make
su -c "make install"
```

A program elindításához az `xmms &` parancsot kell begépelnünk, majd ENTER-t ütnünk. Ha ez az első alkalom, amikor az XMMS-t futtatjuk, valami olyasmit fogunk látni, mint az erősítő az asztali hifiberendezésünkön (lásd az 1. képet).

Ez még csak a program első egysége. Nézzük az erősítő jobb oldalán található gombokat. Az egyiknek EQ (equalizer, azaz hangszín-szabályozó), a másiknak PL (playlist – a lejátszandó zeneszámok listája) a felirata. A gombokra kattintva beren-

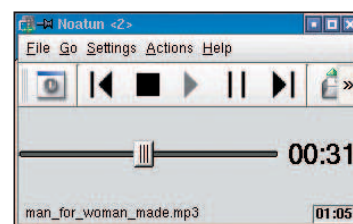


1. kép Az XMMS alapértelmezett képe



2. kép Az XMMS fényjátéka

dezésünk további két egysége lép működésbe. Mivel a három egység egymástól függetlenül mozgatható, előfordulhat, hogy helyzetük beállításával több időt vacakolunk el, mint amennyit szeretnénk. Ez a nehézség egyszerűen megoldható, elég csak a jobb



3. kép A KDE Media Player (alapértelmezett bőr)

egérgombbal kattintani az erősítő egységen, majd a megjelenő menüből kiválasztani az *Options* (beállítás) menüpont *Easy Move* (egyszerű mozgás) beállítást. A CTRL-E gyorsbillentyű-kombinációt is használhatjátok ugyanerre.

Az XMMS széles körűen támogatja a bemenetet, a kimenetet és a megjelenést kezelő bővítményeket. Ehhez a *Preferences* (környezeti beállítás) menüponton keresztül juthatunk el (vagy a CTRL-P gyorsbillentyűvel). Ennek hatására egy új ablak jelenik meg, amely számos futásidejű beállítást kínáló fület tartalmaz. Itt találjuk meg az audio be- és kimenet szabályozásának eszközeit is a különleges hatásos és a megjelenést bővít-

ményei mellett. Ha az XMMS első elindításakor gondok adódnának a hanggal, első utatok ide vezessen. Nyissátok meg az audiooldalt, és ellenőrizzék a kimenet bővítményének beállításait. Nekem az XMMS fordításakor itt az OSS-meghajtót kellett beállítanom.

Még rengeteg időt tölthetnék a különböző beállítások ismeretetésével, ehelyett azonban inkább rátok bízom a kipróbáláskat. A megjelenítés bővítményeinél kezdődik az igazi látványosság. Találni fogtok olyan eszközöket, mint a *Spectrum Analyzer* (hangképelemző), illetve a *Blur Scope*. Korábban már említettem az *OpenGL Spectrum Analyzer*-t, egy másik remek bővítményt, ami a zenéhez illő háromdimenziós színes látványt biztosítja – ezt akár teljes képernyőn is futtathatjátok, elég csak hátradőlni a fotelben, és élvezni az előadást.

Az XMMS általam egyik legkedveltebb vonása (létezik egy pár) a különböző bőrök (skin) használatának lehetősége. Ezekkel az XMMS kinézetét állíthatjátok át az eredeti fekete fémes kinézetéről valami hagyományosabbra, cseresznyefa színűre vagy csiszolt alumínium felületűre. A CTRL-S gyorsbillentyűvel hívható elő a *Skin Browser* (bőrböngésző), amely a jobb kattintásra megjelenő menüből is kiválasztható. Ha épp most fejeztétek be a program telepítését, előfordulhat, hogy a lista egyetlen bejegyzést sem tartalmaz. Ebben az esetben be kell szereznetek néhány bőrt. Ehhez térjete vissza az XMMS honlapjára és válasszátok ki a *Skins* menüpontot. Biztosíthatlak benneteket, nem fogjátok egyhamar megenni a program kinézetét. Az itt fellelhető számtalan bőr között megtaláljátok a fent említett cseresznyefa hatásút is.

Hogy miképpen telepíthetitek ezeket a bőröket? A honlapon mindegyiket `tar.gz` formátumban találjátok. Töltsétek le, amelyik tetszik, és mentsetek a `$HOME/.xmms/Skins` könyvtárba. Nem kell kicsomagolni a fájlt, elég csak az említett helyre másolni. Ha kiderülne, hogy mégis tévedtem, esetleg a <http://www.xmms.org> oldalon lévő választékot is unni kezdenétek, örülni fogtok, ha elmondom, hogy az XMMS a Winamp bőreit is támogatja. Tegyetek látogatást a <http://www.winamp.com> oldalon, egy ideig biztosan nem fogtok unatkozni. Csak kattintsatok a menü *Skins* feliratára – és jó szórakozást! A *második képen* a hagyományos cseresznyefa-felületű XMMS rendszeremet láthatjátok különböző látványos bővítmények ablakainak a társaságában.

Még mindig a bőroknél maradva, egy kicsit elbúcsúzom az XMMS-től, mivel a következő program is remekül kezeli ezeket a bővítményeket. Ha KDE-t használtok, van egy remek kis programotok a gépen, a Media Player. A program indítása a K-menüből, a *Műtímedia* majd a *KDE Media Player* menüpontot választva vagy a `noatun &` parancs begépelésével történhet. A gond az, hogy első indításkor valószínűleg a 3. képen lévő meglehetősen unalmas formában fog megjelenni. Azért ne keseredjete el, ez még csak az alapértelmezett *Excellent* nevű bőr. Kattintsatok a *Beállítások*, majd a *Noatun beállításai...* menüpontra. Az előugró menü *Bővítmőmodulok* pontját választva egy fülekkel ellátott menübe juttok. Az *Csatlakozási felületek* felirat alatt a lejáratos négyféle típusa állítható be. Ezek közül a *K-Jofol* és a *Kaiman* rendelkezik a bőr cserélésének lehetőségével. Mindkét esetben további bőrök tölthetők le a KDE-Look weboldalról (<http://www.kde-look.org>). Ez a hely akkor is nagyon hasznos, ha a KDE-környezet megjelenésével kapcsolatos egyéb anyagokat – asztali témákat, ikonkészleteket – szeretnétek letölteni.

Kezdjétek az *Excellent* felület kikapcsolásával, és válasszátok a *K-Jofol* felületet. Egy megjelenő menüsor mutatja a változókat, a *K-Jofol*-bőrök beállítása is kiválasztható. Ugyanez történne

a Kaiman-felület esetén is. Erre a menüpontra kattintva jobb oldalon egy legördülő lista jelenik meg a telepített bőrök előnézeti képével. A Noatun bőreit a `$HOME/.kde/share/apps/noatun/skins` könyvtár tartalmazza.

Új bőrök hozzáadásához erre a könyvtárra kell váltani, kicsomagolni a <http://www.kde-look.org> címről letöltött bőröket, és már készen is vagyunk. Egy buktatója mégis van a dolognak: az XMMS-től eltérően a bőröknek png formátumban kell lenniük, és muszáj őket kicsomagolni a bőrök alkönyvtárába. Más honnan letöltött bőrök is használhatók, de a jpg formátumban lévő képeket előtte png formátumúra át kell alakítani. Szerencsére ez pofonegyszerű művelet, amelyet a `convert` paranccsal végezhetünk el (a parancs az ImageMagick csomag része):

```
convert image.jpg image.png
```

A fenti parancs csak a KDE 2.x esetén működik, azonban KDE 3.x-et használata esetén egy kicsit eltérően működik a dolog. A `$HOME/.kde/share/apps/noatun/skins` könyvtárban további két alkönyvtárat fogtok találni *winamp* és *kaiman* néven. Ha még nem léteznek, hozzátok létre. Ezután a *winamp* könyvtárba csomagolhattok ki minden arra érdemesnek tartott XMMS-bőrt. Példaként térjünk vissza a cseresznyefa-kinézethez:

```
cd $HOME/.kde/share/apps/noatun/skins/winamp
tar -xzf /path_to/cherrywood.tar.gz
```

Mostantól, ha a *Noatun beállításai* menüpontra kattintotok, ezt az új bőrt látnotok kell a Winskin-listán a felhasználói beállítások menüpontja alatt.

Az XMMS-ről szólva említettem a programhoz található remek bőrök széles választékát. Ez a KDE Media Playerre és a Noatunra is igaz. Kedvenceim közül való a Tyler és a Blur Scope. Próbáljátok ki néhányat közülük, de azért legyetek óvatossal! A Madness nevű tényleg az, aminek hívják, csak akkor próbáljátok ki, ha egy kis bolondságra vágytok – én mindenesetre előre figyelmeztettelek benneteket.

Tekerjétek fel a hangerő-szabályozót, mes amis! François (akinek éppen ebben a számban sikerült egy kis szerephez jutnia, lásd a 25. oldalt) teletölti a poharaitokat. Ma este a pince nyitva áll, mindenki kedvére fogyaszthatja, amit kíván. Szóljon a zene, táncoljanak a fények! Ünnepeljünk!

Viszontlátásra a következő hónapban! A votre santé! Bon appétit!

Linux Journal 2002. augusztus, 100. szám



Marcel Gagné (mggagne@salmar.com) Mississaugaban (Ontario, Kanada) él, a Salmar Consulting Inc. rendszerépítéssel és hálózati tanácsadással foglalkozó cég elnöke. Pilóta és sci-fi író egy személyben. A Világhálón elérhető honlapján sok hasznos dolgot találhatunk. <http://www.salmar.com/marcel/>

Kapcsolódó címek

ImageMagick <http://www.imagemagick.org>
 A KDE honlapja (Noatun) <http://www.kde.org>
 Marcel borlapja <http://www.marcelgagne.com/nfwine.html>
 Az XMMS honlapja <http://www.xmms.org>

Visszaélések az Interneten

Segíts a levélszemét megállításában,
készíts tesztek a diákjaidnak,
és természetesen figyelj a csomagokat.

Néhány hónapja egy érdekes telefonbeszélgetés részese voltam. Nagyjából így folyt le (helyszűke miatt rövidítve közlöm):

– Szevasz, David, van egy kis gond az egyik ügyfelem levélkiszolgálójával (Caldera eServer 2.3). Két évvel ezelőtt telepítettem, azóta remekül ment, de nemrég volt egy kis áramszünet, ami miatt leállt. Most a gép elindul ugyan, de az ethernet-csatolókat nem kelnek életre.

– Hm, indítsd el őket kézzel, aztán kiderül, miért nem indulnak el bekapcsoláskor. Az `fsck` nem talál semmit?

– Nem, semmi. Próbáltam futtatni az `ifconfig`-ot, de nem állítja be a csatolókat.

– Futtasd az `lsmod`-ot, talán nincsenek felrakva az illesztő-programok. Az `lsmod` után próbálkozz újra az `ifconfig`-gal.

– A modulok telepítve vannak, az `ifconfig` `segfault`-tal száll el.

– `Segfault`? – Ekkor mintha félreverték volna egy harangot a fejemben. – Az `ifconfig` RPM-et cseréld le, lehet, hogy a gép leállásakor megsérült – az `ifconfig` ugyanis nem az a fajta program, ami csak hipp-hopp elszállogat – gondoltam.

– A csomagot a `--force` beállítással tudod lecserélni.

– Úgy látom, nem lehet lecserélni az `ifconfig`-ot, még a `--force` is hatástalan.

– Akkor futtasd le az `lsattr ifconfig` parancsot, és mondd, mit látsz.

– Egy „i” betű van a neve mellett. Eközben az Űrsekerekből ismerős vészjelző fények és hangok zsongtak a fejemben. – Akkor figyelj ide, futtasd a `locate ifconfig` parancsot.

– A következőt látom: `/sbin/ifconfig; /dev/sdgl.azgub/backup/ifconfig, /usr/man/man8/ifconfig.8.gz`.

– Remek, találtunk egy rejtett támadócsomagot (rootkit) a `/dev` könyvtárban. Betörték az ügyfeled gépére. Egyébként a kiszolgáló felrakása óta telepítettél valamilyen biztonsági foltot? – Válasz nem érkezett, de feltételeztem, hogy nem. Egy hónap telt el azóta, hogy ajánlatot küldtem neki, ami szerint megoldanám a biztonsági gondjait, és néhány egyéb szolgáltatás mellett telepítenék neki egy tűzfalat is. A behatoló azóta is szabadon garázdálkodik a gépen, az ethernet-kártyák lehallgató üzemmódban vannak, a tisztelt ügyfél pedig valószínűleg könnyen múlnak tartja a biztonsági gondokat; hát igen, megváltoztatta a jelszavát, elővigyázatos volt – szerintem ez is több a semminél. Hogy ki tört be? Miért? Hogy a derekasan aláaknázott gépet felhasználták-e más rendszerek feltörésére, DDOS-támadásokhoz zombiként, ne adj’ isten mindkét célra? Néhány embert egyszerűen el kellene tiltani az Internettől. Vajon egyedül van? Aligha. Kiszolgálóimat naponta megpróbálják feltörni, a sávszélességemet vírusok, önműködő támadások és egyéb vackok emésztik fel. Én pedig fizetek érte. Szerintem egy jó kis törvényre lenne szükségünk...



Smtprc

Ellenőrizd, hátha a saját hálózatról is küldözget valaki levélszemétet! Ezzel a kiegészítővel minden bizonnyal megoldható az e téren jelentkező gondok. A futtatásához `libpthread` és `glibc` szükséges.

➔ <http://sourceforge.net/projects/smtprc>

GTK-Agenda

Kiváló kezdeményezés egy tetszetős, GTK-alapú naptár készítésére. PostgreSQL-adatbázisban tárol neveket, telefonszámokat és elektronikus levélcímeket. Jelenleg sajnos csak spanyol nyelven érhető el, ám a feliratok és egyéb szövegek módosítása nem lehet túl bonyolult, így remélhetőleg hamarosan más nyelveken is elérhető lesz. Az alkalmazásból leveleket is lehet küldeni, ha rendelkezünk futó SMTP-démonnal. Ha további elvárásaink vannak az adatbázissal szemben, a bővítése – úgy tűnik – könnyen megoldható. A futtatáshoz szükséges: `libgtk`, `libgdk`, `libgmodule`, `libglib`, `libdl`, `libXext`, `libX11`, `libm`, `libpq`, `libssl`, `libcrypto`, `libcrypt`, `libresolv`, `libnsl` és `glibc`.

➔ <http://pbrufal.kleenix.org/proyectos.shtml>

x86info

Ha nem elégedsz meg a `/proc/cpuinfo` segítségével megszereshető adatokkal, erre van szükséged. Közvetlenül a processzor regisztereiből olvas, így annyi adatot szolgáltat, amennyit a legtöbbünk soha nem fog megérteni és sohasem fog használni. A futtatásához `glibc` szükséges.

➔ <http://sourceforge.net/projects/x86info>

ILIAS

Ha az Interneten keresztül szeretnél feladatokat és vizsgalásokat kiosztani a diákoknak, megtalálod a megfelelő eszközt. A hallgatók adatai, a vizsgaeredmények és a további tudnivalók egy MySQL-adatbázisba kerülnek. A futtatáshoz szükséges: Apache, PHP és MySQL, MySQL kiszolgáló, GD, `zlib`, `freetype`, `libjpeg`, `ImageMagick`, `zip` és `unzip`.

➔ <http://www.ilias.uni-koeln.de/ios/index-e.html>

pktstat

Rengeteg olyan alkalmazás érhető el, ami a hálózati csomagokat figyeli, ez azonban egy kicsit más. Ez ugyanis a csomagok sávszélesség-használatát figyeli. Apró gyöngyszem, a segítségével pillanatok alatt észreveszed, ha valamelyik felhasználód mondjuk Kazaaval foglalja le a sávszélesség 99,7 százalékát. Futtatásához `libm` és `glibc` szükséges.

➔ <http://www.itee.uq.edu.au/~leonard/personal/software/#pktstat>

DNSMan

Webes alkalmazás, talán a legegyszerűbb módja annak, hogy a BIND zónafájljait karbantartsd, szerintem még a webmin BIND moduljánál is jobb. A követelmények között kevés dolog szere-

pel, viszont a DNS-es rendszeren webkiszolgálót kell üzemeltetned. Fejlesztését továbbra is figyelemmel fogom kísérni, ugyanis a készítője jó pár érdekes ötletet vonultatott fel a tennivalók listájában. A futtatáshoz szükséges: webkiszolgáló (Apache), amely képes CGI-parancsfájlok futtatására, továbbá Perl, BIND 8 vagy 9.
 ➔ <http://www.xsta.cc/dnsman>

ntop

A három évvel ezelőtti emlékek felidézésekor erősen ingadoztam két nagyszerű program között. Az ntop és a sticker-book voltak versenyben, az utóbbi remek fejlesztés gyerekeknek – jómagam nagyon szeretem, de végül mégis az ntop nyert. Három év alatt sokat fejlődött: egyszerű ncurses segédprogramból HTTP- vagy HTTPS-protokoll felett használható, nagy tudású webes ügyfélle nőtte ki magát, amely gdgraph segítségével grafikus megjelenítésre is alkalmas, ha úgy kívánjuk. Az ntop kevésbé hasonlít korábbi önmagára, használata is sokkal könnyebb lett. Ha top jellegű hálózati programra van szükség, ne keresgélj túl sokat. A futtatáshoz szükséges: libmysqclclient, libcrypt, libm, libssl, libpthread, libresolv, libnsl, libld, libgdbm, libz és glibc.
 ➔ <http://www.ntop.org>

Nessus

Három évvel ezelőtt írtam még a Tedről, egy kiváló RTF szövegszerkesztőről, a Nessusról, egy biztonságellenőrző programról és az Nmapról, amely hálózatpásztázó. Nehéz volt, mégis a Nessust választottam közülük.

Na jó, csaltam is egy kicsit. A Nessus ugyanis az Nmapot is használja. A Nessus talán a legteljesebb és legnagyobb tudású biztonság-felülvizsgáló eszköz, amit csak meg lehet szerezni – ráadásul ingyenes is. Ha a fejlesztői kiadást használod, akkor az összes biztonsági hiányosságról áttekintést kapsz, és megteheted a szükséges intézkedéseket. Ha te felelsz a hálózat biztonságért, nélküle neki se állj semminek. A futtatáshoz szükségesek: libX11, libXext, libXi, glibc, libld, libgdk, libglib, libgmp2, libgtk, libm, libnsl és libresolv.

➔ <http://www.nessus.org>

xlog

Rádióamatőrök számára kiváló naplózó eszköz a megismert emberek elérhetőségének nyilvántartására. Több naplót is vezethetsz, a sávokat pedig a *Preferences* (Tulajdonságok) menüben törölheted vagy adhatod hozzá a listához. A dátumot már a program írja be, a kapcsolat létrehozásakor pedig elég rákattintani a *Time* (Idő) gombra, és az időpont is megjelenik. Ezután elég megadni a *Hívó/válaszoló állomás-t*, bepötyögni a megjegyzéseket, kiválasztani a frekvenciát, és az *Add* (Hozzáadás) gombra kattintani. A naplóban később – többek között – keresni is lehet. A felület párját ritkító mértékben felhasználóbarát, a naplókkal akár egy kívülálló is dolgozni tud. Nekem is menni fog. A futtatáshoz szükségesek: libgtk, libgdk, libgmodule, libglib, libld, libXext, libX11, libm és glibc.

➔ <http://people.debian.org/~pa3aba/xlog.html>

axelq

Az Axel segédprogramban nem alakítható ki olyan letöltési sor, amellyel a letöltéseket későbbre állíthatnánk. A kiegészítés segítségével ezt is megtehetjük, és a megadott lista tagjait egy későbbi időpontban az Axel segítségével tölthetjük le. Ha megtetszik az Axel, érdemes letölteni. A futtatáshoz `/bin/sh` szükséges.

➔ <http://electron.its.tudelft.nl/~hemmin98/axelq.html>

integrit

Ez az érdekes kis program a Tripwire és az AIDE mellé sorolja magát. Újabb eszköz a rendszer megfigyelésére, és egész jól működik. Könnyen használható, és beállítható a legfontosabb állományok és könyvtárak változásainak követésére. Az integrit statikusan van lefordítva, így a futtatásához nem kell külső fájl.
 ➔ <http://integrit.sourceforge.net>

GRPN

Ebben a hónapban eléggé megkavarodtam, amikor a három évvel ezelőtti felhozatalból válogattam. A Keystone-t eladták a WhitePajamasnak, fejlesztését leállították, számos más alkalmazás pedig kevés fejlődésen ment keresztül, már ha egyáltalán dolgoztak rajta. Sok más programmal ellentétben nem lett ugyan a kedvencem, mégis a GRPN-t választottam. Valószínűleg nem sokan emlékeznek a fordított lengyel ábrázolásra, azonban elég sok számításhoz használtuk, és ha jól emlékszem, szinte az összes tudományos számításhoz előkerült. Ha szükséged van a lengyel módszerre, vagy egyszerűen csak megtetszett, akkor ez a számológép neked készült. Kezeli az általános matematikai műveleteket, valamint az exponenciális, logaritmus és trigonometrikus függvényeket. A futtatáshoz szükséges: libgtk, libgdk, libmodule, libglib, libld, libXext, libX11, libm és glibc.

➔ <http://lashwhip.com/grpn.html>

TuxTyping

Gyermekeknek készült gépelésoktató program. A betűk lehalló halakkal érkeznek, amelyeket Tuxnak, a pingvinnek el kell kapnia. Ha valami bonyolultabbra vágyunk, rövidebb, túlnyomórészt hárombetűs szavakat is kérhetünk, a feladat nem változik. A grafika és a remek játszhatóság kiváló szórakozást nyújt a gyerekeknek, és eközben még gépelni is megtanulnak. A futtatáshoz szükséges: libSDL, libSDL_image, libSDL_mixer, libm, libld, libartsc, libpthread, libX11, libjpeg, libpng, libz, libtiff, libvorbisfile, libvorbis, libogg, libmpeg, glibc.

➔ <http://www.geekcomix.com/dm/tuxtype>

di

A di segédprogram (disk information – lemezinformációk) rengeteg hasznos adatot árul el a merevlemezéről. Ezek jelentős részét a df is elárulja, ám nem mindet. Az utóbbi időben például a ReiserFS és az ext3 fájlrendszer egyaránt bekerültek a rendszermagba (hogy az LVM-et már ne is említjük), és sokszor nem árt tudni, milyen fájlrendszerrel van dolgunk. A df ezt nem árulja el, a di ellenben azonnal megmutatja. A kimenet formátuma is letisztultabb, mint a df esetében, különösen a hosszú devfs elnevezésekkel rendelkező rendszerek esetében. A futtatásához glibc szükséges.

➔ <http://www.gentoo.com/di>

Ennyit erre a hónapra.

Linux Journal 2002. augusztus, 100. szám.



David A. Bandel

(dbandel@pananix.com) jelenleg Panamában él, Linux- és Unix-tanácsadással foglalkozik. Társszerzője a Que Special Edition: Using Caldera OpenLinux című könyvnek.